

# TRƯỜNG ĐẠI HỌC BÀ RỊA – VŨNG TÀU

## VIỆN CNTT - ĐIỆN - ĐIỆN TỬ



# **BARIA VUNGTAU UNIVERSITY CAP SAINT JACQUES**

## BÁO CÁO HỌC PHẦN

# THỰC TẬP CHUYÊN NGÀNH

# ĐỀ TÀI: QUẢN LÝ THU' VIỆN

Trình độ đào tạo : Đại học

## Ngành : Công nghệ thông tin

## Chuyên ngành : Công nghệ thông tin

Khóa học : 2017-2021

Lóp : DH17CT

Nhóm sinh viên : Nguyễn Thị Minh Thắm

# Nguyễn Huỳnh Hải Phương

GVHD : ThS. Lê Thị Vĩnh Thanh

BÀ RIA - VŨNG TÀU, NĂM 2019

## LỜI MỞ ĐẦU

Sách luôn giữ một vị trí quan trọng trong việc học tập và bồi dưỡng kiến thức cho học sinh đồng thời sách là tài sản – vật phẩm thông dụng nhất, là nơi lưu trữ và truyền bá kiến thức từ thế hệ này sang thế hệ khác. Nhu cầu sử dụng thư viện rất rộng rãi. Có lẽ, không có một lĩnh vực hoạt động trong xã hội lại không cần đến thư viện.

Thư viện trường 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 độc giả. Để phục vụ độc giả nhanh, gọn và chính xác, thư viện cần tin học hóa công việc quản lý danh mục sách và quản lý độc giả của mình. Chính vì số lượng sách ngày càng nhiều chúng ta không thể quản lý bằng sổ sách, để tiết kiệm thời gian cũng như quản lý thông tin sách một cách hiệu quả thì việc xây dựng phần mềm quản lý sách là vô cùng cần thiết. Với đê tài quản lí thư viện chúng ta sẽ thấy được tầm quan trọng của việc ứng dụng công nghệ thông tin trong việc quản lý, nghiên cứu và tìm kiếm thông tin thư viện.

## Mục lục

<b>LỜI MỞ ĐẦU</b>	2
<b>LỜI CẢM ƠN</b>	4
<b>I. Khảo sát</b>	6
<b>II. Phân tích yêu cầu hệ thống</b>	8
<b>III. Thiết kế hệ thống</b>	9
<b>IV. Cơ sở dữ liệu</b>	18
<b>V. Thiết kế Form</b>	20
1. Form Load	20
2. Form Chính	21
3. Form Tìm kiếm sách	22
4. Form Cập nhật Loại sách	22
5. Form Cập nhật sách	23
6. Form Độc giả	24
7. Form Tác giả	24
8. Form Mượn sách	25
9. Form Trả sách	25
<b>VI. Xử lý Form</b>	26
1. Giới thiệu về Mô hình 3-tiers.	26
2. Form Chính	29
3. Form Tìm kiếm sách	32
4. Form Cập nhật	36
5. Form Độc giả	53
6. Form Tác giả	58
7. Form Quản lý mượn/ trả sách	63
<b>VII. Kết luận</b>	70
1. Kết quả đạt được	70
2. Hạn chế	70
3. Hướng phát triển	70
<b>VIII. Tổng kết</b>	70
<b>IX. Tài liệu tham khảo</b>	70

## LỜI CẢM ƠN

----□□□□□----

Để hoàn thành báo cáo này trước tiên nhóm em xin gửi đến các quý thầy, cô giáo trường Đại Học Bà Rịa – Vũng Tàu lời cảm ơn chân thành và sâu sắc nhất.

Em xin chân thành cảm ơn tới giảng viên Lê Thị Vĩnh Thanh - giảng viên Viện CNTT - Điện - Điện Tử là người đã tận tình hướng dẫn nhóm em trong suốt quá trình hoàn thành đề tài này.

Vì kiến thức bản thân còn hạn chế, trong quá trình thực tập, hoàn thiện chuyên đề này nhóm em không tránh khỏi những sai sót, kính mong nhận được đóng góp ý kiến của quý thầy cô và các bạn.

Xin trân trọng cảm ơn!

Tp. Vũng Tàu, ngày tháng năm 2019

**Nhóm sinh viên thực hiện**

Nguyễn Thị Minh Thắm

Nguyễn Huỳnh Hải Phương

### **Nhận xét của giảng viên hướng dẫn**

## **Giảng viên xác nhận**

Kí tên

# XÂY DỰNG PHẦN MỀM QUẢN LÝ THƯ VIỆN

## I. Khảo sát

Thư viện trường Đại Học Bà Rịa – Vũng Tàu quản lý hàng trăm ngàn sách và tạp chí phục vụ cho sinh viên của trường. Sinh viên có thể mượn sách đọc tại chỗ hoặc đem về nhà. Để phục vụ độc giả nhanh, gọn và chính xác, thư viện cần tin học hóa công việc quản lý danh sách và danh mục độc giả của mình.

### GIỚI THIỆU

Thư viện Trường đại học Bà Rịa - Vũng Tàu chính thức được thành lập ngày 26/6/2009 Theo quyết định số 638/QĐ- ĐH BR-VT, trụ sở đặt tại cơ sở 2 số 01 Trương Văn Bang, P.7, Tp.Vũng Tàu.

#### 1. *Chức năng và nhiệm vụ*

##### 1.1. *Chức năng*

Thư viện trường có chức năng phục vụ hoạt động giảng dạy, học tập, đào tạo, nghiên cứu khoa học, triển khai ứng dụng tiến bộ khoa học công nghệ và quản lý của nhà trường thông qua việc sử dụng, khai thác các loại tài liệu có trong thư viện (tài liệu chép tay, in, sao chụp, khắc trên mọi chất liệu, tài liệu điện tử, mạng Internet...).

##### 1.2. *Nhiệm vụ*

a. Tham mưu giúp lãnh đạo Nhà trường xây dựng quy hoạch, kế hoạch hoạt động dài hạn và ngắn hạn của thư viện; tổ chức điều phối toàn bộ hệ thống thông tin, tư liệu, thư viện trong nhà trường;

b. Bổ sung, phát triển nguồn lực thông tin đáp ứng nhu cầu giảng dạy, học tập, nghiên cứu khoa học và chuyển giao công nghệ của nhà trường; thu nhận các tài liệu do nhà trường xuất bản, các công trình nghiên cứu khoa học đã được nghiệm thu, tài liệu hội thảo, khoá luận, luận văn thạc sĩ, luận án tiến sĩ của cán bộ, giảng viên, sinh viên, học viên, chương trình đào tạo, giáo trình, tập bài giảng và các dạng tài liệu khác của nhà trường, các án phẩm tài trợ, biếu tặng, tài liệu trao đổi giữa các thư viện;

d. Tổ chức phục vụ, hướng dẫn cho bạn đọc khai thác, tìm kiếm, sử dụng hiệu quả nguồn tài liệu và các sản phẩm dịch vụ thông tin - thư viện thông qua các hình thức phục vụ của thư viện phù hợp với quy định của pháp luật;

đ. Nghiên cứu, ứng dụng thành tựu khoa học công nghệ tiên tiến và công nghệ thông tin vào công tác thư viện;

e. Xây dựng quy hoạch, kế hoạch, tổ chức bồi dưỡng chuyên môn, nghiệp vụ, ngoại ngữ, tin học cho cán bộ thư viện để phát triển nguồn nhân lực có chất lượng nhằm nâng cao hiệu quả công tác;

g. Tổ chức, quản lý cán bộ, tài sản theo sự phân cấp của hiệu trưởng; bảo quản, kiểm kê định kỳ vốn tài liệu, cơ sở vật chất kỹ thuật và tài sản khác của thư viện; tiến hành thanh lọc ra khỏi kho các tài liệu lạc hậu, hư nát theo quy định;

h. Thực hiện báo cáo tình hình hoạt động hàng năm và báo cáo đột xuất khi có yêu cầu của các cấp có thẩm quyền.

## 2. **Cơ cấu tổ chức**

### 2.1. **Nhân sự**

- Trưởng thư viện: CN. Lê Hồng Dương.
- Chuyên viên: 01 (CN).
- Nhân viên: 01 (TC).

### 2.2. **Các bộ phận chuyên môn nghiệp vụ**

- Tổ Bổ sung, xử lý tài liệu và phục vụ bạn đọc.
- Tổ tin học.

## 3. **Cơ sở vật chất, vốn tài liệu**

### 3.1. **Cơ sở vật chất**

Tổng diện tích 287 m<sup>2</sup> bao gồm:

- Phòng đọc tổng hợp: diện tích 125 m<sup>2</sup>, sức chứa 100 chỗ ngồi.
- Kho sách: diện tích 60 m<sup>2</sup>.
- Kho sách ngoại văn: diện tích 41 m<sup>2</sup>

- Thư viện điện tử: diện tích 60 m<sup>2</sup> với 03 máy điều hòa nhiệt độ, 40 máy tính kết nối Internet cáp quang tích hợp Wifi (Fiber vnn) tốc độ cao.
- 01 Server cài đặt phần mềm quản lý Thư viện, phần mềm Thư viện số.
- 03 máy tính, 01 máy in dành cho cán bộ thư viện sử dụng.
- 02 máy tính dành cho tra cứu tài liệu đặt tại Phòng đọc tổng hợp và Kho tự chọn

- 01 máy nước uống nóng - lạnh.

### **3.2. *Vốn tài liệu***

Tổng số sách: 19.782 bản / 5.401 tên; 77 loại báo – tạp chí; hơn 600 Khóa luận tốt nghiệp, Đề tài nghiên cứu khoa học, luận văn thạc sĩ, luận án tiến sĩ; hơn 1.000 CD-ROM các loại; hơn 10.000 tài liệu điện tử toàn văn.

## **II. Phân tích yêu cầu hệ thống**

### **1. Yêu cầu thiết bị, phần mềm**

- Hỗ trợ laptop hoặc máy tính chạy Windows 7 trở lên.
- Sử dụng máy tính chip Core i3 (i5 càng tốt), RAM 2GB.
- Có cài chương trình Visual Studio.
- Được kết nối với máy in.
- Ngôn ngữ có thể thay đổi.
- Giao diện dễ sử dụng.
- Có thể bảo mật – lưu trữ thông tin người dùng.
- Nâng cấp, chỉnh sửa, cập nhật dữ liệu khi xảy ra lỗi.

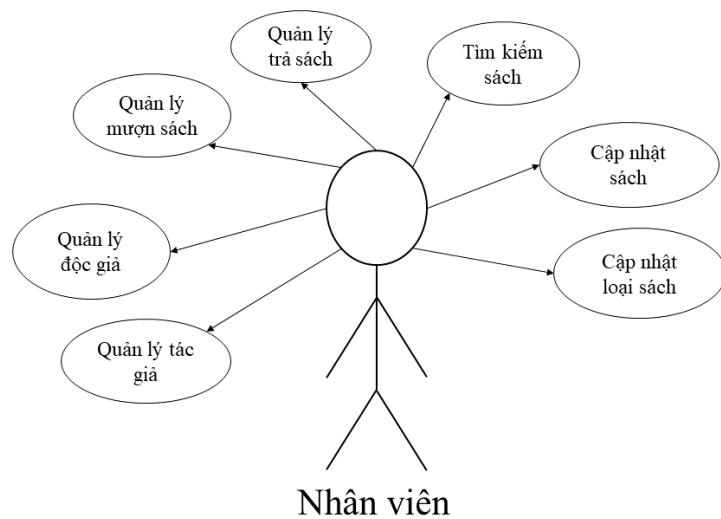
### **2. Yêu cầu người dùng**

- Có kinh nghiệm cài đặt và sử dụng các phần mềm.

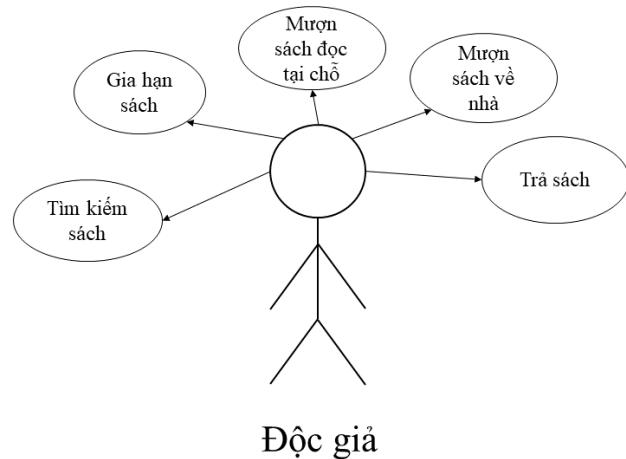
### **III. Thiết kế hệ thống**

#### **1. Xác định tác nhân**

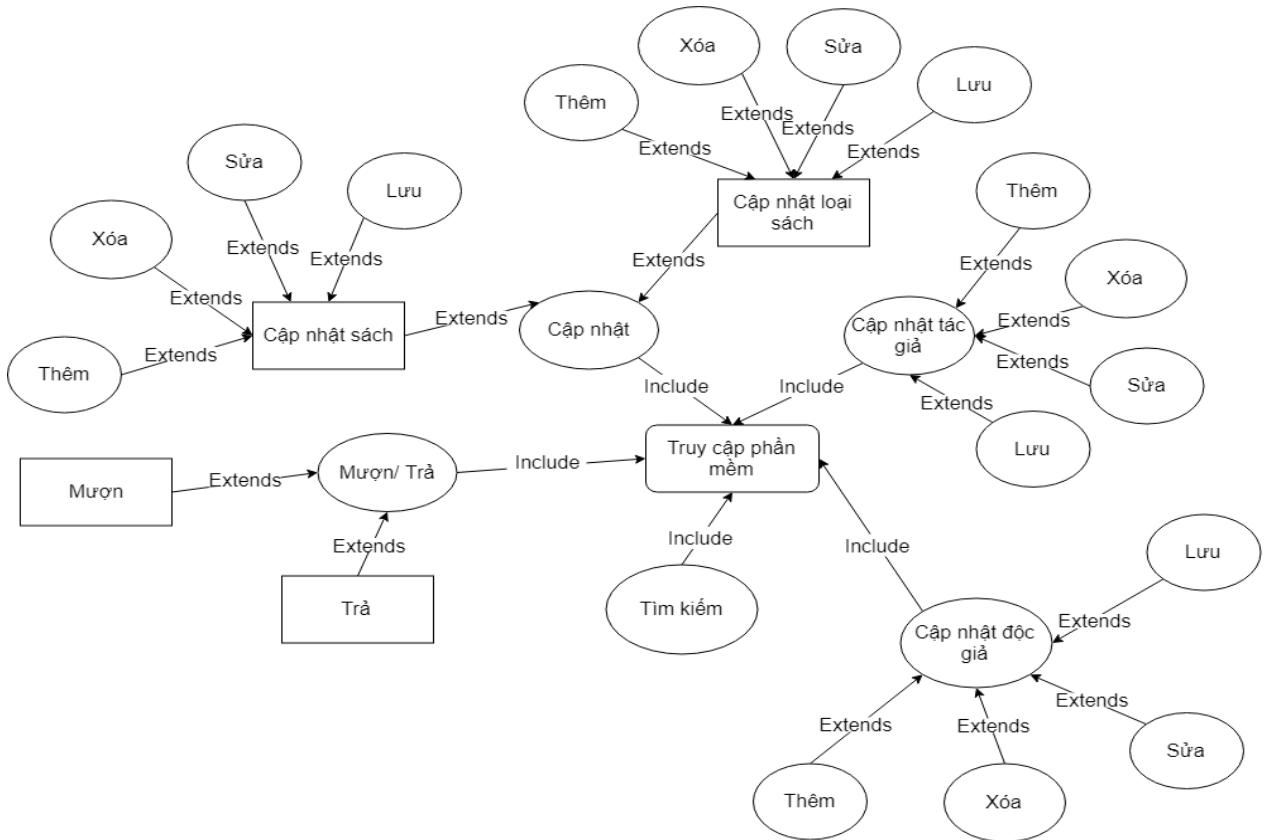
##### **1.1 Nhân viên**



##### **1.2 Độc giả**

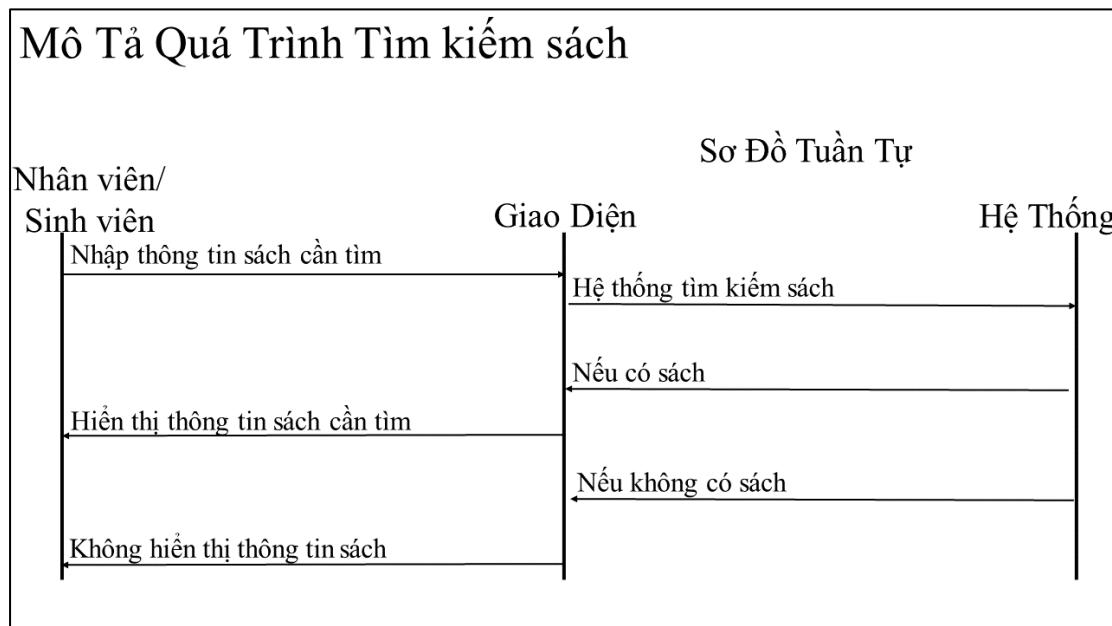


## 2. Use case tổng quát



### 3. Mô tả Use case

### **3.1. Mô tả quá trình Tìm kiếm sách**

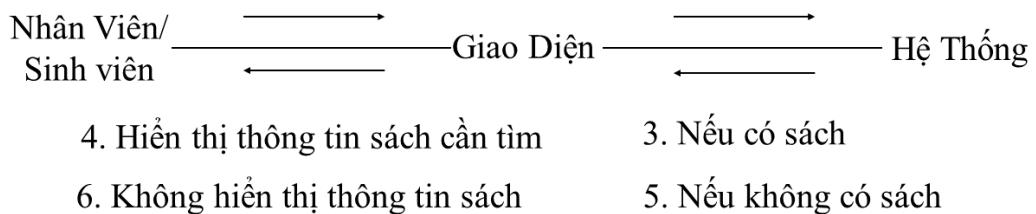


*Hình 3.1.1. Sơ đồ tuần tự mô tả quá trình Tìm kiếm sách*

## Mô Tả Quá Trình Tìm kiếm sách

\*Sơ Đồ Hợp Tác:

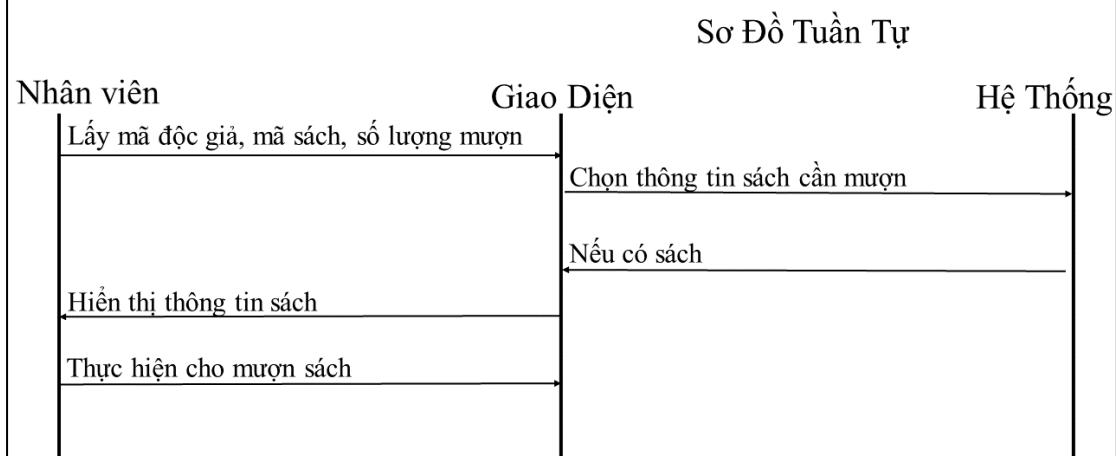
1. Nhập thông tin sách cần tìm
2. Hệ thống tìm kiếm sách



*Hình 3.1.2. Sơ đồ hợp tác mô tả quá trình Tìm kiếm sách*

### 3.2. Mô tả quá trình mượn sách

#### Mô Tả Quá Trình Mượn sách đọc tại chỗ

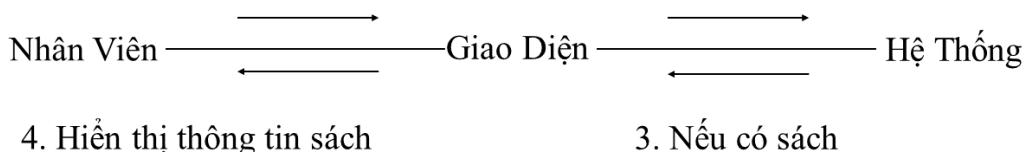


*Hình 3.2.1. Sơ đồ tuần tự mô tả quá trình Mượn sách đọc tại chỗ*

## Mô Tả Quá Trình Mượn sách đọc tại chỗ

\*Sơ Đồ Hợp Tác:

1. Lấy mã độc giả, mã sách số lượng mượn
2. Chọn thông tin sách cần mượn
5. Thực hiện cho mượn sách



Hình 3.2.2. Sơ đồ hợp tác mô tả quá trình Mượn sách đọc tại chỗ

## Mô Tả Quá Trình Mượn sách về nhà

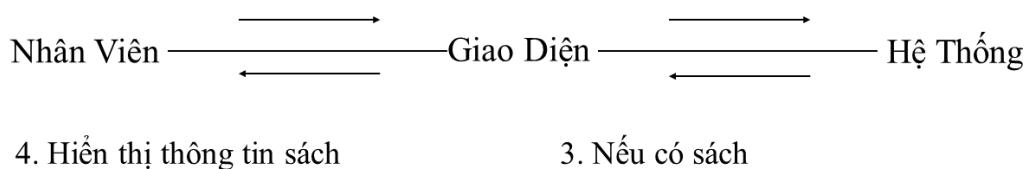


Hình 3.2.3. Sơ đồ tuần tự mô tả quá trình Mượn sách về nhà

## Mô Tả Quá Trình Mượn sách về nhà

\*Sơ Đồ Hợp Tác:

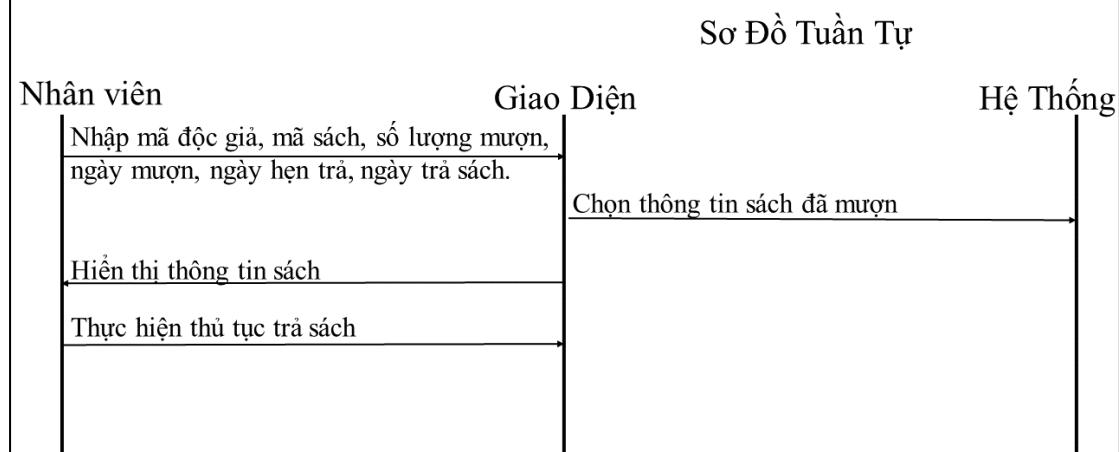
1. Lấy mã độc giả, mã sách, số lượng mượn, ngày mượn và ngày trả sách
2. Chọn thông tin sách cần mượn
5. Làm thủ tục cho mượn sách



Hình 3.2.4. Sơ đồ hợp tác mô tả quá trình Mượn sách về nhà

### 3.3. Mô tả quá trình trả sách

#### Mô Tả Quá Trình Trả sách



Hình 3.3.1. Sơ đồ tuần tự mô tả quá trình Trả sách

## Mô Tả Quá Trình Trả sách

\*Sơ Đồ Hợp Tác:

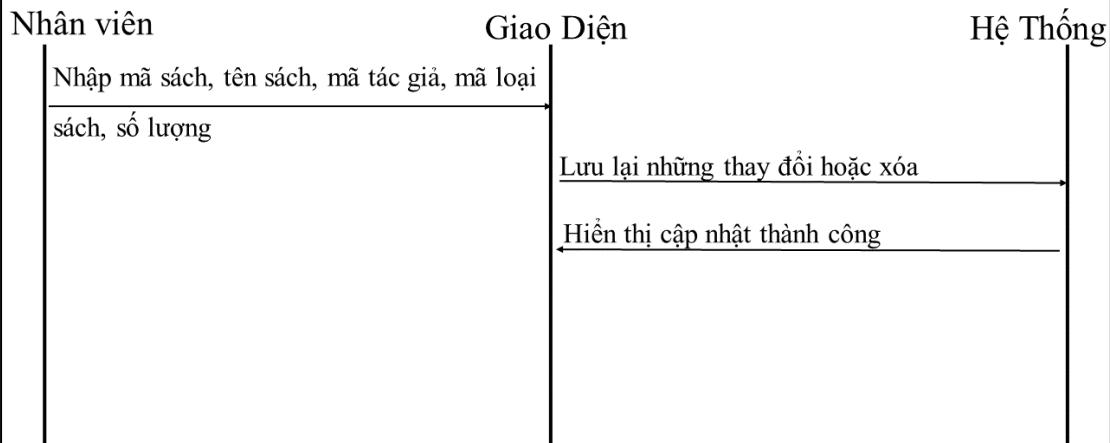
1. Nhập mã độc giả, mã sách, số lượng mượn, ngày mượn, ngày hẹn trả, ngày trả sách.
  2. Chọn thông tin sách đã mượn
  3. Hiển thị thông tin sách
  4. Thực hiện thủ tục trả sách
- Nhân Viên —————→ Giao Diện —————→ Hệ Thống  
←———— Giao Diện ←————

Hình 3.3.2. Sơ đồ hợp tác mô tả quá trình Trả sách

### 3.4. Mô tả quá trình cập nhật sách

## Mô Tả Quá Trình Cập nhật sách

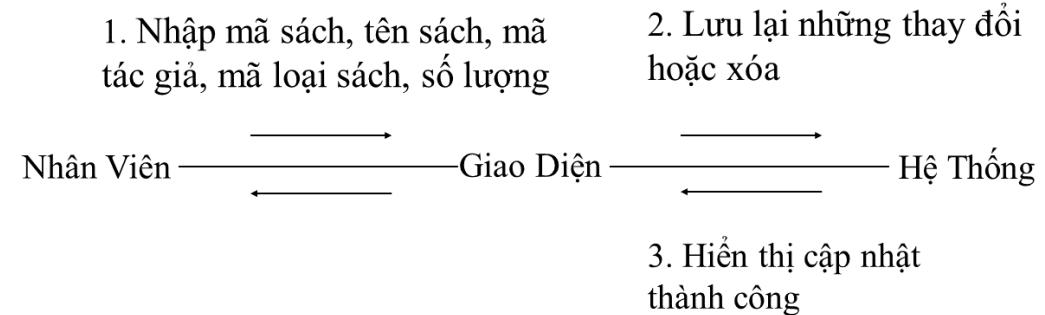
Sơ Đồ Tuần Tự



Hình 3.4.1. Sơ đồ tuần tự mô tả quá trình Cập nhật sách

## Mô Tả Quá Trình Cập nhật sách

\*Sơ Đồ Hợp Tác:

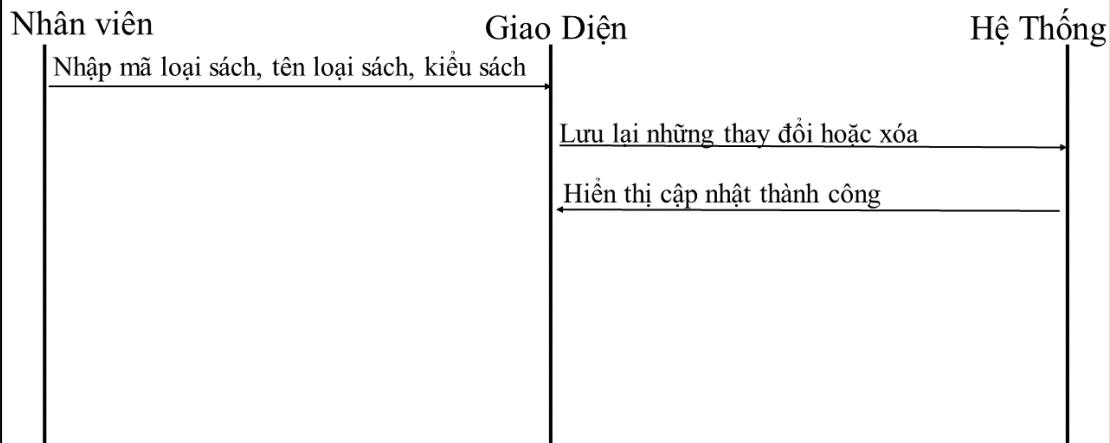


*Hình 3.4.2. Sơ đồ hợp tác mô tả quá trình Cập nhật sách*

### 3.5. Mô tả quá trình cập nhật loại sách

## Mô Tả Quá Trình Cập nhật loại sách

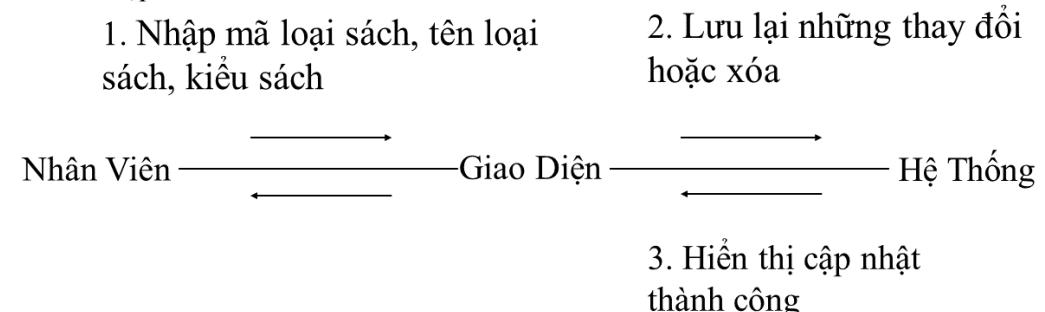
Sơ Đồ Tuần Tự



*Hình 3.5.1. Sơ đồ tuần tự mô tả quá trình Cập nhật loại sách*

## Mô Tả Quá Trình Cập nhật loại sách

\*Sơ Đồ Hợp Tác:

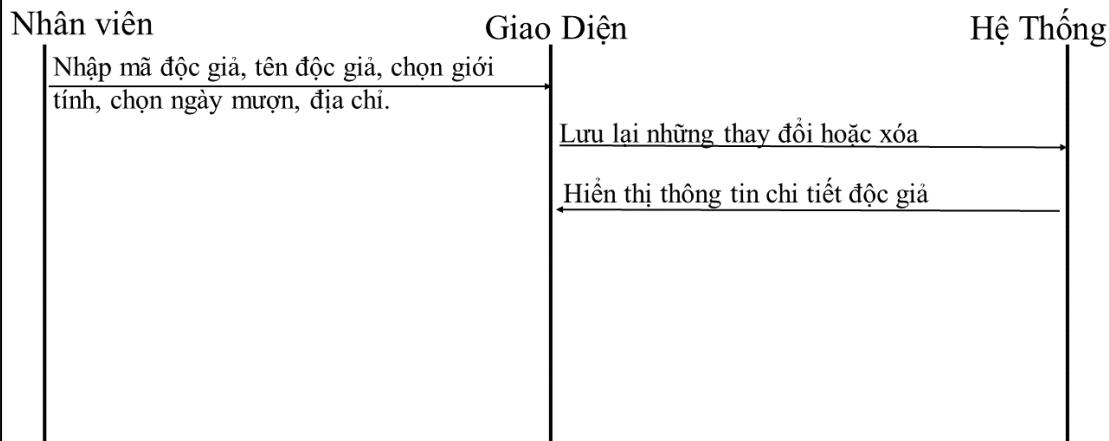


Hình 3.5.2. Sơ đồ hợp tác mô tả quá trình Cập nhật loại sách

### 3.6. Mô tả quá trình quản lý độc giả

#### Mô Tả Quá Trình Quản lý độc giả

Sơ Đồ Tuần Tự

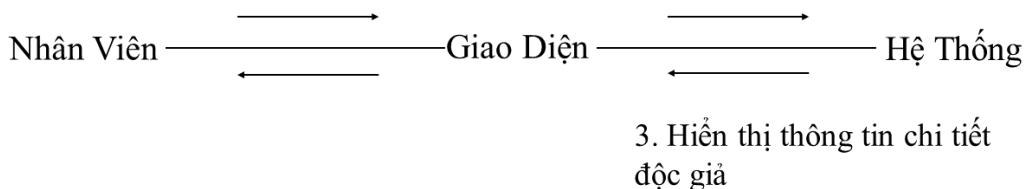


Hình 3.6.1. Sơ đồ tuần tự mô tả quá trình Quản lý độc giả

## Mô Tả Quá Trình Quản lý độc giả

\*Sơ Đồ Hợp Tác:

1. Nhập mã độc giả, tên độc giả, chọn giới tính, chọn ngày mượn, địa chỉ.
2. Lưu lại những thay đổi hoặc xóa

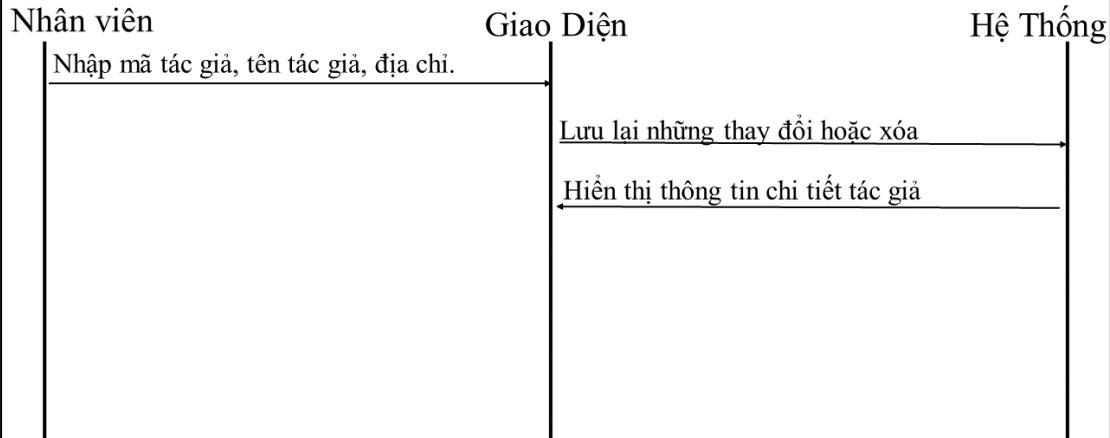


*Hình 3.6.2. Sơ đồ hợp tác mô tả quá trình Quản lý độc giả*

### 3.7. Mô tả quá trình quản lý tác giả

## Mô Tả Quá Trình Quản lý tác giả

Sơ Đồ Tuần Tự

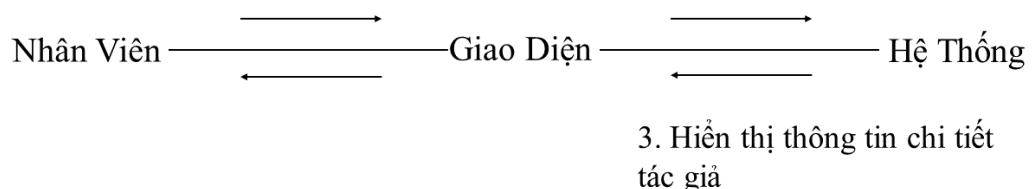


*Hình 3.7.1. Sơ đồ tuần tự mô tả quá trình Quản lý tác giả*

## Mô Tả Quá Trình Quản lý tác giả

\*Sơ Đồ Hợp Tác:

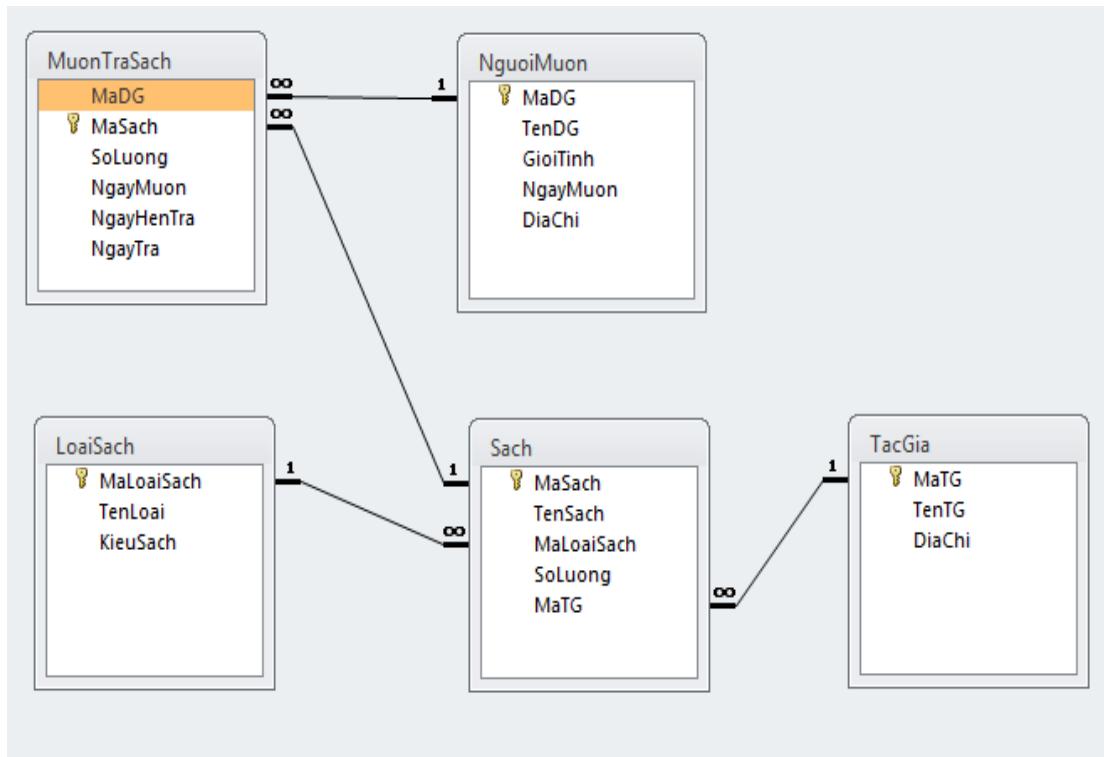
1. Nhập mã tác giả, tên tác giả, địa chỉ.
2. Lưu lại những thay đổi hoặc xóa



Hình 3.7.2. Sơ đồ hợp tác mô tả quá trình Quản lý tác giả

## IV. Cơ sở dữ liệu

### 1. Thiết kế cơ sở dữ liệu



Hình 1.1. Sơ đồ quan hệ cơ sở dữ liệu

## 2. Mô tả cơ sở dữ liệu

- Mô tả bảng Sach

Chứa thông tin của sách: MaSach, TenSach, MaLoaiSach, SoLuong, MaTG

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	MaSach	Text	Là khóa chính để phân biệt với các sách khác, mỗi sách có 1 mã sách
2	TenSach	Text	Tên của sách
3	MaLoaiSach	Text	Thuộc tính để phân biệt các loại sách(Khóa ngoại)
4	SoLuong	Number	Cho biết số lượng của mỗi cuốn sách trong danh mục sách
5	MaTG	Text	Mã tác giả của sách

- Mô tả bảng LoaiSach

Chứa thông tin của loại sách: MaLoaiSach, TenLoai, KieuLoai.

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	MaLoaiSach	Text	Thuộc tính để phân biệt các loại sách (khóa chính). Mỗi loại sách có 1 mã loại
2	TenLoai	Text	Tên của thể loại sách
3	KieuSach	Text	Kiểu sách

- Mô tả bảng NguoiMuon

Chứa thông tin của độc giả, người mượn sách: MaDG, TenDG, GioiTinh, NgayMuon, DiaChi

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	MaDG	Text	Là khóa chính để phân biệt với các độc giả khác, mỗi độc giả được cung cấp một mã riêng.
2	TenDG	Text	Tên của độc giả
3	GioiTinh	Yes/No	Giới tính độc giả
4	NgayMuon	Date time	Ngày độc giả mượn sách
5	DiaChi	Text	Địa chỉ của độc giả

- Mô tả bảng **TacGia**

Chứa thông tin của tác giả mỗi đầu sách: MaTG, TenTG, DiaChi

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaTG	Text	Là khóa chính để phân biệt với các tác giả khác, mỗi tác giả được cung cấp một mã riêng.
2	TenTG	Text	Tên của tác giả
5	DiaChi	Text	Địa chỉ của tác giả

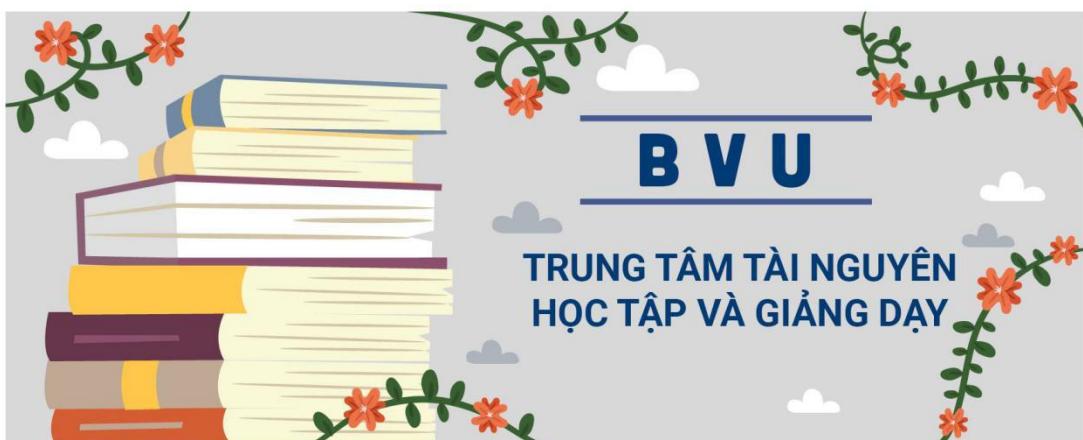
- Mô tả bảng **MuongTraSach**

Chứa thông tin của độc giả mượn sách và sách mà độc giả đó mượn, ngày mượn trả sách: MaDG, MaSach, SoLuong, NgayMuon, NgayHenTra, NgayTra.

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaDG	Text	Mã độc giả mượn sách
2	MaSach	Text	Mã sách mà độc giả mượn
5	SoLuong	Number	Số lượng sách mà độc giả mượn
6	NgayMuon	Date time	Ngày độc giả mượn sách
7	NgayHenTra	Date time	Ngày độc giả hẹn trả sách
8	NgayTra	Date time	Ngày độc giả trả sống

## V. Thiết kế Form

### 1. Form Load



## 2. Form Chính

CỬA SỔ QUẢN LÝ CHÍNH

# QUẢN LÝ THU VIỆN

**Loại Sách**

Chính Khoa Kì Kinh Sách Trí Học Năng Tế giá... Tạp Tin Chi Hoc
---

**Thông tin chi tiết**

Mã sách :	
Tên sách :	
Mã loại sách :	
Số lượng :	
Mã tác giả :	

**Danh Sách**

Mã sách	Tên sách	Mã loại sách	Số lượng	Mã Tác giả

**Công cụ**

- Cập nhật
- Quản lý độc giả
- Quản lý tác giả
- Quản lý mượn trả
- Tìm kiếm
- Thoát

- Hiển thị các loại sách.
- Hiển thị danh sách, thông tin chi tiết sách.
- Cho phép nhân viên quản lý thư viện cập nhật, quản lý độc giả, ...

### 3. Form Tìm kiếm sách

TÌM KIẾM

#### Nhập Thông Tin Cần Tìm Kiếm

Tìm Theo

Mã Sách  
 Tên Sách

**Tìm kiếm**

**Thoát**

---

#### Thông Tin Tìm Kiếm

Mã Sách :

Tên Sách :

Số Lượng :

Mã Tác Giả :

Mã Loại Sách :

- Chức năng: Cho phép người dùng nhập nội dung và hệ thống sẽ hiển thị tên sách cần tìm.

### 4. Form Cập nhật Loại sách

CẬP NHẬT SÁCH

#### Cập Nhật Loại Sách

##### Chi Tiết Loại Sách

Mã Loại Sách :  Kiểu Sách :

Tên Loại Sách :

**Thêm**   **Lưu**   **Sửa**   **Xóa**   **Thoát**

---

#### Danh Sách Loại Sách

Mã Loại Sách	Tên Loại Sách	Kiểu Sách
CT	Chính Trị	Sách
KH	Khoa Học	Báo
KN	Kĩ Năng	Sách
KT	Kinh Tế	Sách
SGK	Sách giáo khoa	Sách
TC	Tạp Chí	Báo
TH	Tin Học	Sách

- Chức năng: Cho phép nhân viên quản lý cập nhật loại sách. Có thể thêm, sửa, xóa hoặc lưu.

## 5. Form Cập nhật sách

- Chức năng: Cho phép nhân viên quản lý thư viện cập nhật sách. Có thể thêm, sửa, xóa hoặc lưu.

## 6. Form Độc giả

QUẢN LÝ ĐỘC GIẢ

**Chi Tiết Độc Giả**

Mã Độc Giả :	<input type="text"/>	Giới Tính	Ngày Mượn :	Wednesday, August 14, 2019	
Tên Độc Giả :	<input type="text"/>	<input checked="" type="radio"/> Nam	<input type="radio"/> Nữ	Địa Chỉ :	<input type="text"/>

**Thao Tác:** [Thêm] [Lưu] [Sửa] [Xóa] [Thoát]

Mã Độc Giả	Tên Độc Giả	Giới Tính	Ngày Mượn	Địa Chỉ
DG001	Nguyễn Thị Minh ...	Nữ	1/1/2019	Phú Yên
DG002	Nguyễn Huỳnh Hả...	Nữ	2/1/2019	Vũng Tàu
DG003	Lê Thành Trung	Nam	6/6/2019	Long Sơn
DG004	Nguyễn Vĩnh Lộc	Nam	2/7/2019	Vũng Tàu
DG005	Nguyễn Tùng Lâm	Nam	6/7/2019	Hạ Long
DG006	Đinh Văn Thương	Nam	2/22/2019	Đà Lạt

- Chức năng: Cho phép nhân viên quản lý thư viện ghi lại thông tin độc giả. Có thể thêm, sửa, xóa hoặc lưu thông tin độc giả.

## 7. Form Tác giả

QUẢN LÝ TÁC GIẢ

**Chi Tiết Tác Giả**

Mã Tác Giả :	<input type="text"/>	Địa Chỉ :	<input type="text"/>
Tên Tác Giả :	<input type="text"/>		

**Thao Tác:** [Thêm] [Lưu] [Sửa] [Xóa] [Thoát]

Mã Tác Giả	Tên Tác Giả	Địa Chỉ
TG001	Lê Nguyễn Hoài ...	Thành phố HCM
TG002	Nguyễn Anh Tuấn	Hà Nội
TG003	Phạm Mỹ Tiên	Thành phố HCM
TG004	Phan Gia Huy	Thái Bình
TG005	Phan Gia Bảo	Thái Bình
TG006	Phan Trúc Linh	Vũng Tàu
TG007	Phan Gia Hân	Vung Tàu
TG008	Đinh Van Thuong	Đà Lat

- Chức năng: Cho phép nhân viên quản lý thư viện ghi lại thông tin tác giả. Có thể tùy chỉnh như thêm, xóa, sửa hoặc lưu thông tin tác giả.

## 8. Form Mượn sách

Quản Lý Mượn Sách

<b>Thông Tin Sách</b>	<b>Mượn Sách</b>																				
Chọn Mã Sách : <input type="text" value="7 thói quen thành công"/>	Mã Độc Giả : <input type="text" value="DG001"/>																				
Mã Sách :	Mã Sách :																				
Mã Loại :	Số Lượng Mượn :																				
Số Lượng :	Ngày Mượn : <input type="text" value="Wednesday, August 14, 2019"/>																				
Mã Tác Giả :	Ngày Hẹn Trả : <input type="text" value="Wednesday, August 14, 2019"/>																				
<input type="button" value="Mượn Mới"/> <input type="button" value="Cho Mượn"/> <input type="button" value="Thoát"/>																					
<b>Danh sách cho mượn</b>																					
<table border="1"> <tr> <th>Mã Độc Giả</th> <th>Mã Sách</th> <th>Số Lượng</th> <th>Ngày Mượn</th> <th>Ngày Hẹn Trả</th> </tr> <tr> <td>DG001</td> <td>S001</td> <td>1</td> <td>7/3/2019</td> <td>3/14/2019</td> </tr> <tr> <td>DG002</td> <td>S002</td> <td>2</td> <td>3/4/2019</td> <td>10/3/2019</td> </tr> <tr> <td>DG002</td> <td>S005</td> <td>1</td> <td>5/4/2019</td> <td>5/15/2019</td> </tr> </table>		Mã Độc Giả	Mã Sách	Số Lượng	Ngày Mượn	Ngày Hẹn Trả	DG001	S001	1	7/3/2019	3/14/2019	DG002	S002	2	3/4/2019	10/3/2019	DG002	S005	1	5/4/2019	5/15/2019
Mã Độc Giả	Mã Sách	Số Lượng	Ngày Mượn	Ngày Hẹn Trả																	
DG001	S001	1	7/3/2019	3/14/2019																	
DG002	S002	2	3/4/2019	10/3/2019																	
DG002	S005	1	5/4/2019	5/15/2019																	

- Chức năng: Cho phép nhân viên quản lý thư viện lấy thông tin độc giả; thông tin, số lượng, mốc thời gian sách cần mượn.

## 9. Form Trả sách

Quản Lý Mượn Sách

<b>Trả Sách</b>																									
Mã Độc Giả : <input type="text" value="DG001"/> Ngày Mượn : <input type="text" value="Wednesday, August 14, 2019"/>																									
Mã Sách :	Ngày Hẹn Trả : <input type="text" value="Wednesday, August 14, 2019"/>																								
Số Lượng Mượn :	Ngày Trả : <input type="text" value="Wednesday, August 14, 2019"/>																								
<input type="button" value="Trả Sách"/> <input type="button" value="Lưu"/> <input type="button" value="Thoát"/> Trả Sách :																									
<b>Danh Sách Trả Sách</b>																									
<table border="1"> <tr> <th>Mã Độc Giả</th> <th>Mã Sách</th> <th>Số Lượng</th> <th>Ngày Mượn</th> <th>Ngày Hẹn Trả</th> <th>Ngày Trả</th> </tr> <tr> <td>DG001</td> <td>S001</td> <td>1</td> <td>7/3/2019</td> <td>3/14/2019</td> <td>10/3/2019</td> </tr> <tr> <td>DG002</td> <td>S002</td> <td>2</td> <td>3/4/2019</td> <td>10/3/2019</td> <td>9/3/2019</td> </tr> <tr> <td>DG002</td> <td>S005</td> <td>1</td> <td>5/4/2019</td> <td>5/15/2019</td> <td>5/10/2019</td> </tr> </table>		Mã Độc Giả	Mã Sách	Số Lượng	Ngày Mượn	Ngày Hẹn Trả	Ngày Trả	DG001	S001	1	7/3/2019	3/14/2019	10/3/2019	DG002	S002	2	3/4/2019	10/3/2019	9/3/2019	DG002	S005	1	5/4/2019	5/15/2019	5/10/2019
Mã Độc Giả	Mã Sách	Số Lượng	Ngày Mượn	Ngày Hẹn Trả	Ngày Trả																				
DG001	S001	1	7/3/2019	3/14/2019	10/3/2019																				
DG002	S002	2	3/4/2019	10/3/2019	9/3/2019																				
DG002	S005	1	5/4/2019	5/15/2019	5/10/2019																				

- Chức năng: Cho phép nhân viên quản lý thư viện lấy thông tin độc giả; thông tin sách, mốc giờ gian ngày mượn và ngày trả sách.

## VI. Xử lý Form

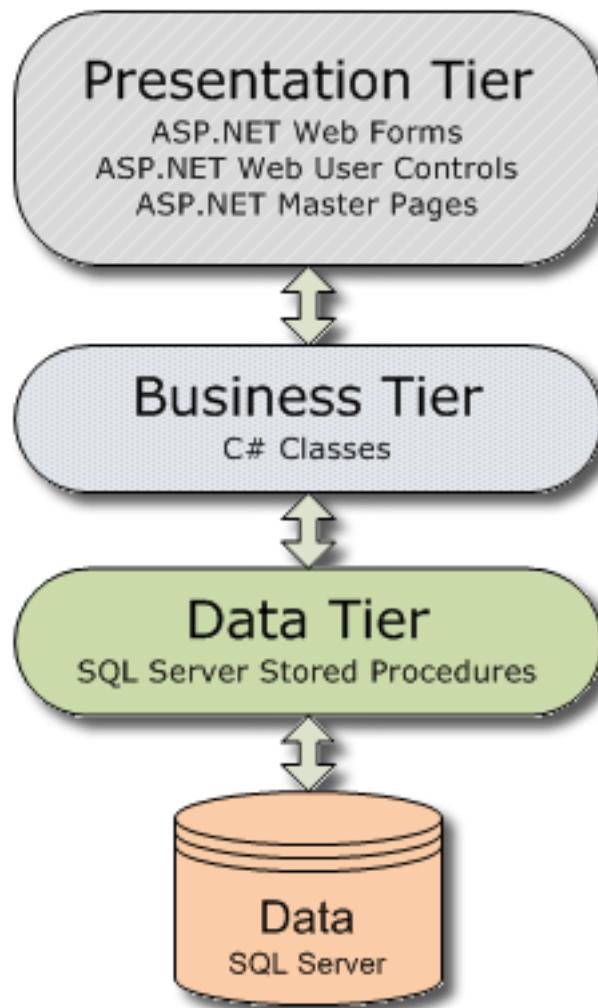
### 1. Giới thiệu về Mô hình 3-tiers.

- 3-tiers là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế.

- 3-tiers có tính vật lý (physical): là mô hình client-server (mỗi tier có thể đặt chung 1 nơi hoặc nhiều nơi, kết nối với nhau qua Web services, WCF, Remoting...).

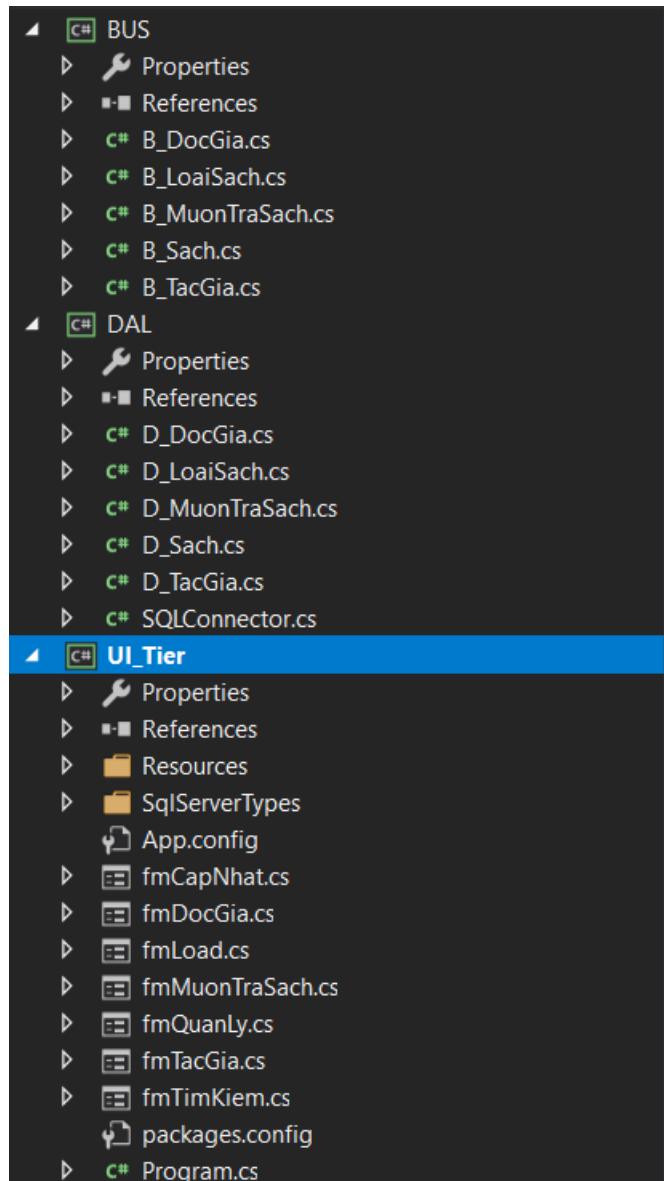
- Như vậy, ta có thể mô hình này phân tách ứng dụng ra làm 3 module riêng biệt, bao gồm:

- **Presentation tier:** bao gồm các thành phần phần xử lý giao diện Graphic User Interface (GUI). Được dùng để giao tiếp với người dùng, nhiệm vụ chính là hiển thị dữ liệu và nhận dữ liệu từ người dùng.
- **Business tier:** gồm các thành phần Business Logic Layer (BLL), Data Access Layer (DAL) và Data Transfer Object (DTO). Được dùng để cung cấp các chức năng của phần mềm
- **Data tier:** lưu trữ dữ liệu, là các hệ quản trị CSDL như MS SQL Server, Oracle, SQLite, MS Access, XML files, text files,... Dùng để lưu trữ dữ liệu, cho phép lớp Business Logic có thể tìm kiếm, trích xuất, cập nhật, ...



*Hình 1.1. Mô hình 3-tiers*

- \* **Ưu điểm:**
  - Dễ dàng mở rộng, thay đổi quy mô của hệ thống: Khi cần tải lớn, người quản trị có thể dễ dàng thêm các máy chủ vào nhóm, hoặc lấy bớt ra trong trường hợp ngược lại.
- \* **Nhược điểm:**
  - Việc truyền dữ liệu giữa các tầng sẽ chậm hơn vì phải truyền giữa các tiến trình khác nhau (IPC), dữ liệu cần phải được đóng gói → truyền đi → mở gói trước khi có thể dùng được.
  - Việc phát triển ứng dụng phức tạp hơn.
  - Cấu trúc 3 tầng tương ứng của phần mềm quản lý thư viện



## 2. Form Chính

The screenshot shows the Main Form (Form Chính) with the following components:

- Section 1 (Loại Sách):** A list of book categories: Chính Khoa, Kỹ Kinh Sách, Trí Học Năng Té giá..., Tạp Tin Chi Học. It includes two numbered arrows: **1** pointing to the list and **2** pointing to the 'Danh Sách' section.
- Section 3 (Thông tin chi tiết):** A group of input fields for book details: Mã sách, Tên sách, Mã loại sách, Số lượng, and Mã tác giả. Each field is enclosed in a red box and has an arrow labeled **3** pointing to it.
- Section 2 (Danh Sách):** A table showing a list of books with columns: Mã sách, Tên sách, Mã loại sách, Số lượng, and Mã Tác giả. This section is also enclosed in a red box and has an arrow labeled **2** pointing to it.
- Section 4 (Sidebar):** A vertical stack of buttons with icons and labels:
  - Cập nhật
  - Quản lý độc giả
  - Quản lý tác giả
  - Quản lý mượn trả
  - Tìm kiếm
  - Thoát

STT	Biên cô	Xử lý
1	Chọn	Chọn loại muốn xem thông tin
2	Hiển thị	Hiển thị thông tin danh sách các đầu sách thuộc loại sách đã chọn
3	Hiển thị	Hiển thị thông tin chi tiết
4		Chọn thao tác cần thực hiện: <ul style="list-style-type: none"> <li>Khi Click nút “Cập nhật” sẽ hiển thị form Cập nhật loại sách và Cập nhật sách.</li> <li>Khi Click nút “Quản lý độc giả” sẽ hiển thị form Quản lý độc giả.</li> <li>Khi Click nút “Quản lý tác giả” sẽ hiển thị form Quản lý tác giả.</li> <li>Khi Click nút “Quản lý mượn trả” sẽ hiển thị ra form Quản lý mượn trả.</li> </ul>

- Code của Form Chính:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
using System.Windows.Forms;
using BUS;

namespace UI_Tier
{
    public partial class fmQuanLy : Form
    {
        public fmQuanLy()
        {

            fmLoad f = new fmLoad();
            f.ShowDialog();
            InitializeComponent();
        }

        DataTable dtLoaiSach, dtSach;
        B_LoaiSach objLoaiSach = new B_LoaiSach();
        B_Sach objSach = new B_Sach();

        private void btnCapNhat_click(object sender, EventArgs e)
        {
            fmCapNhat f = new fmCapNhat();
            f.ShowDialog();
        }

        private void btnQLDocGia_click(object sender, EventArgs e)
        {
            fmDocGia f = new fmDocGia();
            f.ShowDialog();
        }

        private void btnQLTacGia_click(object sender, EventArgs e)
        {
            fmTacGia f = new fmTacGia();
            f.ShowDialog();
        }

        private void btnQlMuonTra_click(object sender, EventArgs e)
        {
            fmMuonTraSach f = new fmMuonTraSach();
            f.ShowDialog();
        }

        private void btnTimKiem_click(object sender, EventArgs e)
        {
            fmTimKiem f = new fmTimKiem();
            f.ShowDialog();
        }

        private void btnThoat_click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

```

private void QuanLy_Load(object sender, EventArgs e)
{
    NapDuLieuLoaiSach();
}

private void lvwLoaiSach_SelectedIndexChanged(object sender,
EventArgs e)
{
    lvwLoaiSach.Items.Clear();
    if (lvwLoaiSach.SelectedItems.Count > 0)
    {
        dtSach =
objSach.getTableSach_Theo_MaLoaiSach(lvwLoaiSach.SelectedItems[0].Ta
g.ToString());
        foreach (DataRow dr in dtSach.Rows)
        {
            ListViewItem li =
lvwLoaiSach.Items.Add(dr["MaSach"].ToString());
            li.Tag = dr["MaSach"].ToString();
            li.SubItems.Add(dr["TenSach"].ToString());
            li.SubItems.Add(dr["MaLoaiSach"].ToString());
            li.SubItems.Add(dr["SoLuong"].ToString());
            li.SubItems.Add(dr["MaTG"].ToString());
            li.ImageIndex = 1;
        }
    }
}

private void LvwLoaiSach_SelectedIndexChanged_1(object
sender, EventArgs e)
{
    lvwDanhSach.Items.Clear();
    if (lvwLoaiSach.SelectedItems.Count > 0)
    {
        dtSach =
objSach.getTableSach_Theo_MaLoaiSach(lvwLoaiSach.SelectedItems[0].Ta
g.ToString());
        foreach (DataRow dr in dtSach.Rows)
        {
            ListViewItem li =
lvwDanhSach.Items.Add(dr["MaSach"].ToString());
            li.Tag = dr["MaSach"].ToString();
            li.SubItems.Add(dr["TenSach"].ToString());
            li.SubItems.Add(dr["MaLoaiSach"].ToString());
            li.SubItems.Add(dr["SoLuong"].ToString());
            li.SubItems.Add(dr["MaTG"].ToString());
            li.ImageIndex = 1;
        }
    }
}

private void LvwDanhSach_SelectedIndexChanged(object sender,
EventArgs e)

```

```

{
    if (lvwDanhSach.SelectedItems.Count > 0)
    {
        foreach (ListViewItem lvw in
lvwDanhSach.SelectedItems)
        {
            ListViewItem item =
lvwDanhSach.SelectedItems[0];
            txtMaSach.Text = lvw.SubItems[0].Text;
            txtTenSach.Text = lvw.SubItems[1].Text;
            txtMaLoaiSach.Text = lvw.SubItems[2].Text;
            txtSoLuong.Text = lvw.SubItems[3].Text;
            txtMaTG.Text = lvw.SubItems[4].Text;
        }
    }
}

private void NapDuLieuLoaiSach()
{
    dtLoaiSach = objLoaiSach.getTableLoaiSach();
    foreach (DataRow dr in dtLoaiSach.Rows)
    {
        ListViewItem li =
lvwLoaiSach.Items.Add(dr["TenLoai"].ToString());
        li.Tag = dr["MaLoaiSach"].ToString();
        li.ImageIndex = 0;
    }
}
}

```

### 3. Form Tìm kiếm sách

The screenshot shows a Windows application window titled "TÌM KIẾM". The interface is divided into several sections:

- Search Input:** A text input field with a placeholder "Nhập Thông Tin Cần Tìm Kiếm" (Enter search information). It is highlighted with a red border.
- Search Button:** A button labeled "Tìm kiếm" with a magnifying glass icon. It is highlighted with a red border.
- Search Criteria Panel:** A group of radio buttons under the heading "Tim Theo" (Search by). It contains two options: "Mã Sách" and "Tên Sách". This panel is highlighted with a red border.
- Exit Button:** A button labeled "Thoát" (Exit) with a red border.
- Result Title:** A title "Thông Tin Tìm Kiếm" (Search results) with a red border.
- Result Fields:** A row of five text input fields for displaying search results. From left to right: "Mã Sách : [ ]", "Mã Tác Giả : [ ]", "Tên Sách : [ ]", "Mã Loại Sách : [ ]", and "Số Lượng : [ ]". All these fields are grouped together with a red border.

STT	Biến cód	Xử lý
1	Nhập nội dung	Nhập vào thông tin yêu cầu.
2	Chọn	Click nút “Tìm kiếm” để thực hiện tìm kiếm.
3	Chọn	Chọn cách thức tìm kiếm.
4	Hiển thị	Hiển thị thông tin vừa tìm thấy.

- Code của Form Tìm kiếm sách:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using BUS;

namespace UI_Tier
{
    public partial class fmTimKiem : Form
    {
        public fmTimKiem()
        {
            InitializeComponent();
        }
        B_Sach objSach = new B_Sach();
        private void BtnTimKiem_Click(object sender, EventArgs e)
        {
            DataTable dtMa =
objSach.getTableSach_Theo_MaSach(txtTimSach.Text.ToString());
            DataTable dtTen =
objSach.getTableSach_Theo_TenSach(txtTimSach.Text.ToString());
            int a = 0;
            if (btnTimKiem.Text == "Tìm Kiếm")
            {
                if (txtTimSach.Text == "")
                {
                    MessageBox.Show("Vui lòng nhập thông tin cần tìm",
"Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    txtTimSach.Focus();
                }
                else
                {
                    if (rdMaSach.Checked == false && rdTenSach.Checked ==
false)
                    {
                        MessageBox.Show("Vui lòng lựa chọn tìm kiếm theo mã
sách hoặc tên sách", "Thông Báo", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                    }
                    else
                    {
                        btnTimKiem.Text = "Thứ lại";
                        txtTimSach.Enabled = false;
                    }
                }
            }
        }
    }
}

```

```

        if (rdMaSach.Checked == true)
        {
            foreach (DataRow drw in dtMa.Rows)
            {
                txtMaSach.Text = drw["MaSach"].ToString();
                txtTenSach.Text =
drw["TenSach"].ToString();
                txtSoLuong.Text =
drw["SoLuong"].ToString();
                txtMaTacGia.Text = drw["MaTG"].ToString();
                txtMaLoaiSach.Text =
drw["MaloaiSach"].ToString();
                a++;
            }
        }
        else if (rdTenSach.Checked == true)
        {
            foreach (DataRow drw in dtTen.Rows)
            {
                txtMaSach.Text = drw["MaSach"].ToString();
                txtTenSach.Text =
drw["TenSach"].ToString();
                txtSoLuong.Text =
drw["SoLuong"].ToString();
                txtMaTacGia.Text = drw["MaTG"].ToString();
                txtMaLoaiSach.Text =
drw["MaloaiSach"].ToString();
                a++;
            }
        }
        if (a != 0)
        {
            lblTTTimKiem.Text = "Thông tin về sách: " +
txtTimSach.Text;
        }
        else
        {
            lblTTTimKiem.Text = "Không tìm thấy sách này
trong thông tin sách";
        }
    }
}
else
{
    btnTimKiem.Text = "Tìm Kiếm";
    txtTimSach.Enabled = true;
    txtTimSach.Text = "";
    txtTimSach.Focus();

    txtMaSach.Text = "";
    txtTenSach.Text = "";
    txtSoLuong.Text = "";
    txtMaTacGia.Text = "";
    txtMaLoaiSach.Text = "";
    lblTTTimKiem.Text = "";
}
}

```

```
    private void FmTimKiem_FormClosing(object sender,
FormClosingEventArgs e)
{
    DialogResult dl = MessageBox.Show("Bạn có muốn đóng chương
trình không?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
    if (dl == DialogResult.No)
        e.Cancel = true;
}

private void BtnThoat_Click(object sender, EventArgs e)
{
    this.Close();
}

}
```

## 4. Form Cập nhật

### a. Form Cập nhật loại sách:

**1** ← **2** ← **3** → **4**

STT	Biến cố	Xử lý
1	Nhập nội dung	Nhập vào thông tin yêu cầu.
2	Hiển thị	Hiển thị danh sách thông tin các loại sách.
3	Chọn	Chọn thao tác cần thực hiện.
4	Chọn	Chuyển qua tab Cập nhật sách.

### b. Form Cập nhật sách:

**5** ← **1** ← **2** ← **3** ← **4** → **5**

STT	Biến cố	Xử lý
1	Nhập nội dung	Nhập vào thông tin yêu cầu.
2	Hiển thị	Hiển thị danh sách thông tin các loại sách.
3	Chọn	Chọn thao tác cần thực hiện.
4	Chọn	Chuyển qua tab Cập nhật sách.
5	Chọn	Chuyển qua tab Cập Nhật Loại Sách.

<b>STT</b>	<b>Biến cố</b>	<b>Xử lý</b>
1	Nhập nội dung	Nhập vào thông tin yêu cầu.
2	Chọn	Chọn thông tin có sẵn trong danh sách.
3	Hiển thị	Hiển thị danh sách thông tin sách.
4	Chọn	Chọn thao tác cần thực hiện.
5	Chọn	Chuyển qua tab Cập nhật loại sách.

- Code của Form Cập nhật:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using BUS;

namespace UI_Tier
{
    public partial class fmCapNhat : Form
    {
        public fmCapNhat()
        {
            InitializeComponent();
        }

        DataTable dtSach, dtLoaiSach, dtTacGia;
        B_LoaiSach objLoaiSach = new B_LoaiSach();
        B_Sach objSach = new B_Sach();
        B_TacGia objTacGia = new B_TacGia();
        BindingSource bs = new BindingSource();
        public static string strCapNhat;
    }
}

```

```
private void btnLuuLoaiSach_Click(object sender, EventArgs
e)
{
    if (txtMaLoaiSach.Text == "")
    {
        MessageBox.Show("Chưa nhập mã loại sách!", "Error");
        txtMaLoaiSach.Focus();
    }
    else if (txtTenLoaiSach.Text == "")
    {
        MessageBox.Show("Chưa nhập tên loại sách!",
"Error");
        txtTenLoaiSach.Focus();
    }
    else if (txtKieuSach.Text == "")
    {
        MessageBox.Show("Chưa nhập kiểu sách!", "Error");
        txtKieuSach.Focus();
    }
    else
    {
        if (lvwDanhSachLoaiSach.SelectedItems.Count == 0)
        {
            LuuThemLoaiSach();
        }
        else
        {
            LuuSuaLoaiSach();
        }
    }
}
private void LuuThemLoaiSach()
{
```

```

try
{
    lvwDanhSachLoaiSach.Items.Clear();
    DataRow dr = dtLoaiSach.NewRow();
    dr["MaLoaiSach"] = txtMaLoaiSach.Text;
    dr["TenLoai"] = txtTenLoaiSach.Text;
    dr["KieuSach"] = txtKieuSach.Text;
    setButtonLoaiSach();
    dtLoaiSach.Rows.Add(dr);
    objLoaiSach.CapNhatTale_LoaiSach(dtLoaiSach);
    NapListViewLoaiSach();
    MessageBox.Show("Lưu mới dữ liệu thành công!");
}
catch (Exception)
{
    NapListViewLoaiSach();
    MessageBox.Show("Mã loại sách " + txtMaLoaiSach.Text
+ " đã tồn tại!\nVui lòng nhập lại mã loại sách!");
    txtMaLoaiSach.Focus();
}
}

private void LuuSuaLoaiSach()
{
    try
    {
        dtLoaiSach =
objLoaiSach.getTableLoaiSach_Theo_MaLoai(lvwDanhSachLoaiSach.Selected
dItems[0].Text);
        dtLoaiSach.Rows[0]["MaLoaiSach"] =
txtMaLoaiSach.Text;
        dtLoaiSach.Rows[0]["TenLoai"] = txtTenLoaiSach.Text;
        dtLoaiSach.Rows[0]["KieuSach"] = txtKieuSach.Text;
        objLoaiSach.CapNhatTale_LoaiSach(dtLoaiSach);
    }
}

```

```
        MessageBox.Show("Lưu sửa dữ liệu thành công!");
        NapListViewLoaiSach();
    }
    catch (Exception)
    {

        MessageBox.Show("Thất bại!");
    }
}

private void setButtonLoaiSach()
{
    btnThemLoaiSach.Text = "Thêm";
    txtMaLoaiSach.Text = "";
    txtTenLoaiSach.Text = "";
    txtKieuSach.Text = "";
    txtMaLoaiSach.Focus();
    btnXoaLoaiSach.Enabled = true;
    btnSuaLoaiSach.Enabled = true;
    lvwDanhSachLoaiSach.Enabled = true;
}

private void NapListViewLoaiSach()
{
    dtLoaiSach = objLoaiSach.getTableLoaiSach();
    lvwDanhSachLoaiSach.Items.Clear();
    foreach (DataRow dr in dtLoaiSach.Rows)
    {
        ListViewitem li =
    lvwDanhSachLoaiSach.Items.Add(dr["MaLoaiSach"].ToString());
        li.Tag = dr["MaLoaiSach"].ToString();
        li.SubItems.Add(dr["TenLoai"].ToString());
        li.SubItems.Add(dr["KieuSach"].ToString());
        li.ImageIndex = 0;
```

```
        }

    }

private void btnSuaLoaiSach_Click(object sender, EventArgs
e)
{
    if (lvwDanhSachLoaiSach.SelectedItems.Count > 0)
    {
        if (btnSuaLoaiSach.Text == "Sửa")
        {
            btnSuaLoaiSach.Text = "Hủy";
            grbChiTietLoaiSach.Enabled = true;
            txtMaLoaiSach.Enabled = false;
            txtTenLoaiSach.SelectAll();
            txtTenLoaiSach.Focus();
            btnThemLoaiSach.Enabled = false;
            btnXoaLoaiSach.Enabled = false;
        }
        else
        {
            btnSuaLoaiSach.Text = "Sửa";
            grbChiTietLoaiSach.Enabled = false;
            btnThemLoaiSach.Enabled = true;
            btnXoaLoaiSach.Enabled = true;
        }
    }
    else
    {
        MessageBox.Show("Chưa chọn dữ liệu cần sửa!");
    }
}

private void btnXoaLoaiSach_Click(object sender, EventArgs
e)
```

```
{  
    if (lvwDanhSachLoaiSach.SelectedItems.Count > 0)  
    {  
        DialogResult dl = MessageBox.Show("Bạn có chắc muốn  
xóa dữ liệu này không?", "Warning", MessageBoxButtons.YesNo,  
MessageBoxIcon.Question);  
        if (dl == DialogResult.Yes)  
        {  
            try  
            {  
                DataRow[] dr =  
dtLoaiSach.Select("MaLoaiSach='" + txtMaLoaiSach.Text + "'");  
                dr[0].Delete();  
  
objLoaiSach.CapNhatTale_LoaiSach(dtLoaiSach);  
                // clear text  
                txtMaLoaiSach.Text = "";  
                txtTenLoaiSach.Text = "";  
                txtKieuSach.Text = "";  
                setButtonLoaiSach();  
                NapListViewLoaiSach();  
                MessageBox.Show("Xóa thành công!");  
            }  
            catch (Exception ex)  
            {  
                MessageBox.Show(ex.Message);  
  
            }  
        }  
    }  
    else  
    {  
        MessageBox.Show("Bạn hãy chọn dữ liệu cần xóa!");  
    }  
}
```

```
        }

    }

private void btnThem_click(object sender, EventArgs e)
{
    if(btnThem.Text == "Thêm")
    {
        btnThem.Text = "Hủy";
        txtMaSach.Text = "";
        txtTenSach.Text = "";
        txtSoLuong.Text = "";
        grpChiTietSach.Enabled = true;
        txtMaSach.Focus();
        btnXoa.Enabled = false;
        btnSua.Enabled = false;
        //lvwDanhSach.Enabled = false;
    }
    else
    {
        grpChiTietSach.Enabled = false;
        btnThem.Text = "Thêm";
        txtMaSach.Text = "";
        txtTenSach.Text = "";
        txtSoLuong.Text = "";
        btnXoa.Enabled = true;
        btnSua.Enabled = true;
        lvwDanhSach.Enabled = true;
    }
}

private void btnLuu_click(object sender, EventArgs e)
{
```

```
if (txtMaSach.Text == "")  
{  
    MessageBox.Show("Chưa nhập mã sách!", "Error");  
    txtMaSach.Focus();  
}  
else if (txtTenSach.Text == "")  
{  
    MessageBox.Show("Chưa nhập tên sách!", "Error");  
    txtTenSach.Focus();  
}  
else if (txtSoLuong.Text == "")  
{  
    MessageBox.Show("Chưa nhập số lượng sách!",  
"Error");  
    txtSoLuong.Focus();  
}  
else  
{  
    if (lvwDanhSach.SelectedItems.Count == 0)  
    {  
        LuuThem();  
    }  
    else  
    {  
        LuuSua();  
    }  
}  
}  
private void LuuThem()  
{  
    try  
    {
```

```

        DataRow dr = dtSach.NewRow();
        dr["masach"] = txtMaSach.Text;
        dr["tensach"] = txtTenSach.Text;
        dr["maloaisach"] = cbMaLoaiSach.SelectedValue;
        dr["soluong"] = Convert.ToInt32(txtSoLuong.Text);
        dr["matg"] = cbTenTG.SelectedValue;
        dtSach.Rows.Add(dr);
        objSach.CapNhatTable_Sach(dtSach);
        btnSuaLoaiSach.Enabled = true;
        btnXoaLoaiSach.Enabled = true;
        l vwDanhSach.Enabled = true;
        btnThemLoaiSach.Text = "Thêm";
        MessageBox.Show("Lưu mới dữ liệu " + txtMaSach.Text
+ " thành công!");
        NapListViewDanhSach();
    }
    catch (Exception)
    {
        MessageBox.Show("Mã Sách " + txtMaSach.Text + " đã
tồn tại! \nVui nhập mã sách khác!");
        NapListViewDanhSach();
        txtMaSach.Focus();
    }
}
private void LuuSua()
{
    try
    {
        dtSach =
objSach.getTableSach_Theo_MaSach(lvwDanhSach.SelectedItems[0].Text);
        dtSach.Rows[0]["MaSach"] = txtMaSach.Text;
        dtSach.Rows[0]["TenSach"] = txtTenSach.Text;
    }
}

```

```
        dtSach.Rows[0]["MaLoaiSach"] =
cbMaLoaiSach.SelectedValue;

        dtSach.Rows[0]["MaTG"] = cbTenTG.SelectedValue;

        dtSach.Rows[0]["SoLuong"] =
Convert.ToInt32(txtSoLuong.Text);

        objSach.CapNhatTable_Sach(dtSach);

        MessageBox.Show("Lưu sửa dữ liệu thành công!");

        NapListViewDanhSach();

    }

    catch (Exception)
{



    MessageBox.Show("Thất bại!");

}

}

private void NapListViewDanhSach()
{

    dtSach = objSach.getTableSach();

    lvwDanhSach.Items.Clear();

    foreach (DataRow dr in dtSach.Rows)
{



        ListViewItem li =
lvwDanhSach.Items.Add(dr["MaSach"].ToString());

        li.Tag = dr["MaSach"].ToString();

        li.SubItems.Add(dr["TenSach"].ToString());

        li.SubItems.Add(dr["MaLoaiSach"].ToString());

        li.SubItems.Add(dr["SoLuong"].ToString());

        li.SubItems.Add(dr["MaTG"].ToString());

        li.ImageIndex = 0;

    }

}

private void btnSua_click(object sender, EventArgs e)
```

```
{  
    if (lvwDanhSach.SelectedItems.Count > 0)  
    {  
        if (btnSua.Text == "Sửa")  
        {  
            btnSua.Text = "Hủy";  
            grbChiTietSach.Enabled = true;  
            txtMaSach.Enabled = false;  
            txtTenSach.SelectAll();  
            txtTenSach.Focus();  
            btnThemLoaiSach.Enabled = false;  
            btnXoaLoaiSach.Enabled = false;  
        }  
        else  
        {  
            btnSua.Text = "Sửa";  
            btnThemLoaiSach.Enabled = true;  
            btnXoaLoaiSach.Enabled = true;  
            grbChiTietSach.Enabled = false;  
        }  
    }  
    else  
    {  
        MessageBox.Show("Bạn hãy chọn dữ liệu cần sửa!",  
"Thông Báo!");  
        grbChiTietSach.Enabled = false;  
    }  
}  
  
private void btnXoa_click(object sender, EventArgs e)  
{  
    if (lvwDanhSach.SelectedItems.Count > 0)
```

```
{  
    DialogResult dl = MessageBox.Show("Bạn có chắc muốn  
xóa dữ liệu này không?", "Warning", MessageBoxButtons.YesNo,  
MessageBoxIcon.Question);  
  
    if (dl == DialogResult.Yes)  
    {  
        try  
        {  
            DataRow[] dr = dtSach.Select("MaSach='"  
+txtMaSach.Text + "'");  
            dr[0].Delete();  
            objSach.CapNhatTable_Sach(dtSach);  
            // clear text  
            txtMaSach.Text = "";  
            txtTenSach.Text = "";  
            cbMaLoaiSach.Text = "";  
            txtSoLuong.Text = "";  
            cbTenTG.Text = "";  
            MessageBox.Show("Đã xóa dữ liệu!", "Thông  
Báo!");  
        }  
        catch (Exception ex)  
        {  
  
            MessageBox.Show(ex.Message);  
        }  
    }  
    else  
    {  
        MessageBox.Show("Hãy chọn dữ liệu cần xóa!", "Thông  
báo");  
    }  
}
```

```
private void btnThoat_click(object sender, EventArgs e)
{
    this.Close();
}

private void CapNhat_Load(object sender, EventArgs e)
{
    NapListViewDanhSach();
    NapListViewLoaiSach();
    NapComboBox();
    grbChiTietLoaiSach.Enabled = false;
    grbChiTietSach.Enabled = false;
}
private void NapComboBox()
{
    dtLoaiSach = objLoaiSach.getTableLoaiSach();
    cbMaLoaiSach.DataSource = dtLoaiSach;
    cbMaLoaiSach.DisplayMember = "TenLoai";
    cbMaLoaiSach.ValueMember = "MaLoaiSach";
    dtTacGia = objTacGia.getTableTacGia();
    cbTenTG.DataSource = dtTacGia;
    cbTenTG.DisplayMember = "TenTG";
    cbTenTG.ValueMember = "MaTG";
}

private void CapNhat_FormClosing(object sender,
CancelEventArgs e)
{
    DialogResult dl = MessageBox.Show("Bạn có muốn đóng
chương trình không?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
    if (dl == DialogResult.No)
```

```
        e.Cancel = true;
    }

    private void lvwDanhSachLoaiSach_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (lvwDanhSachLoaiSach.SelectedItems.Count > 0)
    {
        foreach (ListViewItem lvw in
lvwDanhSachLoaiSach.SelectedItems)
        {
            ListViewItem item = new ListViewItem();
            txtMaLoaiSach.Text = lvw.SubItems[0].Text;
            txtTenLoaiSach.Text = lvw.SubItems[1].Text;
            txtKieuSach.Text = lvw.SubItems[2].Text;
        }
    }
}

private void lvwDanhSach_SelectedIndexChanged_1(object
sender, EventArgs e)
{
    if (lvwDanhSach.SelectedItems.Count > 0)
    {
        ListViewItem item = lvwDanhSach.SelectedItems[0];
        if (item != null)
        {
            foreach (ListViewItem lvw in
lvwDanhSach.SelectedItems)
            {
                txtMaSach.Text = lvw.SubItems[0].Text;
                txtTenSach.Text = lvw.SubItems[1].Text;
                cbMaLoaiSach.SelectedValue =
lvw.SubItems[2].Text;
            }
        }
    }
}
```

```
        txtSoLuong.Text = lvw.SubItems[3].Text;
        cbTenTG.SelectedValue =
lvw.SubItems[4].Text;
    }
}

}

private void btnThoatLoaiSach_Click(object sender, EventArgs e)
{
    this.Close();
}

private void FmCapNhat_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult dl = MessageBox.Show("Bạn có muốn đóng
chương trình không?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
    if (dl == DialogResult.No)
        e.Cancel = true;
}

private void CbTenTG_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void btnThemLoaiSach_click(object sender, EventArgs e)
{
    if (btnThemLoaiSach.Text == "Thêm")
    {
```

```
    btnThemLoaiSach.Text = "Hủy";
    txtMaSach.Text = "";
    txtTenSach.Text = "";
    txtSoLuong.Text = "";
    grbChiTietLoaiSach.Enabled = true;
    txtMaSach.Focus();
    btnXoaLoaiSach.Enabled = false;
    btnSuaLoaiSach.Enabled = false;
    // lvwDanhSachLoaiSach.Enabled = false;
}
else
{
    grbChiTietLoaiSach.Enabled = false;
    btnThemLoaiSach.Text = "Thêm";
    txtMaSach.Text = "";
    txtTenSach.Text = "";
    txtSoLuong.Text = "";
    btnXoaLoaiSach.Enabled = true;
    btnSuaLoaiSach.Enabled = true;
    lvwDanhSachLoaiSach.Enabled = true;
}
}

private void txtSoLuong_KeyPress(object sender,
KeyEventArgs e)
{
    if ((e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar
>= 'a' && e.KeyChar <= 'z'))
        e.Handled = true;
}
}
```

## 5. Form Độc giả

The screenshot shows a Windows application window titled "QUẢN LÝ ĐỘC GIẢ". Inside, there's a form for "Chi Tiết Độc Giả" (Reader Detail) with fields for Reader ID, Name, Gender (radio buttons for Male and Female), Borrow Date (set to Wednesday, August 14, 2019), and Address. Below the form is a grid listing reader details with columns: Reader ID, Name, Gender, Borrow Date, and Address. The grid contains six entries. At the bottom, there are buttons for Adding, Saving, Editing, Deleting, and Exiting.

Mã Độc Giả	Tên Độc Giả	Giới Tính	Ngày Mượn	Địa Chỉ
DG001	Nguyễn Thị Minh ...	Nữ	1/1/2019	Phú Yên
DG002	Nguyễn Huỳnh Hâ...	Nữ	2/1/2019	Vũng Tàu
DG003	Lê Thành Trung	Nam	6/6/2019	Long Sơn
DG004	Nguyễn Vĩnh Lộc	Nam	2/7/2019	Vũng Tàu
DG005	Nguyễn Tùng Lâm	Nam	6/7/2019	Hạ Long
DG006	Đinh Văn Thương	Nam	2/22/2019	Đà Lạt

STT	Biến cố	Xử lý
1	Nhập nội dung	Nhập vào thông tin yêu cầu.
2	Chọn	Chọn ngày mượn.
3	Chọn	Chọn giới tính.
4	Hiển thị	Hiển thị danh sách thông tin độc giả.
5	Chọn	Chọn thao tác cần thực hiện.

- Code của Form Độc giả:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using BUS;

namespace UI_Tier
{
    public partial class fmDocGia : Form
    {
        public fmDocGia()
        {
            InitializeComponent();
        }
    }
}
```

```
DataTable dtDocGia;
B_DocGia objDocGia = new B_DocGia();
private void btnThem_click(object sender, EventArgs e)
{
    if (btnThem.Text == "Thêm")
    {
        btnThem.Text = "Hủy";
        grbChiTietDG.Enabled = true;
        txtMaDG.Text = "";
        txtTenDG.Text = "";
        dtNgayMuon.Text = "";
        txtDiaChi.Text = "";
        txtMaDG.Focus();
        btnXoa.Enabled = false;
        btnSua.Enabled = false;
        // lvwDanh SachDG.Enabled = false;
    }
    else
    {
        setButton();
    }
}
private void setButton()
{
    btnThem.Text = "Thêm";
    txtMaDG.Text = "";
    txtTenDG.Text = "";
    rdNam.Checked = true;
    dtNgayMuon.Text = "";
    txtDiaChi.Text = "";
    btnXoa.Enabled = true;
    btnSua.Enabled = true;
    lvwDanh SachDG.Enabled = true;
}
private void btnLuu_click(object sender, EventArgs e)
{
    if (txtMaDG.Text == "")
    {
        MessageBox.Show("Chưa nhập mã Độc Giả!", "Thông báo");
        txtMaDG.Focus();
    }
    else if (txtTenDG.Text == "")
    {
        MessageBox.Show("Chưa nhập Tên Độc Giả!", "Thông báo");
        txtTenDG.Focus();
    }
    else if (txtDiaChi.Text == "")
    {
        MessageBox.Show("Chưa nhập địa chỉ độc giả!", "Thông báo");
        txtDiaChi.Focus();
    }
}
```

```

        else
    {
        if (lvwDanhSachDG.SelectedItems.Count == 0)
            LuuThemDG();
        else
            LuuSuaDG();
    }
}

private void LuuThemDG()
{
    try
    {
        lvwDanhSachDG.Items.Clear();
        DataRow dr = dtDocGia.NewRow();
        dr[ "MaDG" ] = txtMaDG.Text;
        dr[ "TenDG" ] = txtTenDG.Text;
        dr[ "GioiTinh" ] = rdNam.Checked ? "Nam" : "Nữ";
        dr[ "NgayMuon" ] = dtNgayMuon.Value;
        dr[ "DiaChi" ] = txtDiaChi.Text;
        dtDocGia.Rows.Add(dr);
        objDocGia.CapNhatTable_DocGia(dtDocGia);
        setButton();
        NapListViewDSDocGia();
        MessageBox.Show("Lưu dữ liệu thành công!");
    }
    catch (Exception)
    {
        NapListViewDSDocGia();
        MessageBox.Show("Lưu dữ liệu thất bại!");
    }
}
private void NapListViewDSDocGia()
{
    dtDocGia = objDocGia.getTableDocGia();
    lvwDanhSachDG.Items.Clear();
    foreach (DataRow dr in dtDocGia.Rows)
    {
        ListViewItem li =
        lvwDanhSachDG.Items.Add(dr[ "MaDG" ].ToString());
        li.Tag = dr[ "MaDG" ].ToString();
        li.SubItems.Add(dr[ "TenDG" ].ToString());
        li.SubItems.Add(dr[ "GioiTinh" ].ToString());
        string ngaymuon =
Convert.ToDateTime(dr[ "NgayMuon" ].ToString()).ToShortDateString();
        li.SubItems.Add(ngaymuon);
        li.SubItems.Add(dr[ "DiaChi" ].ToString());
        li.ImageIndex = 0;
    }
}
private void LuuSuaDG()
{
    try
    {

```

```

        dtDocGia =
objDocGia.getTableDocGia_Theo_MaDG(lvwDanhSachDG.SelectedItems[0].Te
xt);
        dtDocGia.Rows[0][ "MaDG" ] = txtMaDG.Text;
        dtDocGia.Rows[0][ "TenDG" ] = txtTenDG.Text;
        dtDocGia.Rows[0][ "GioiTinh" ] = rdNu.Checked ? true :
false;
        dtDocGia.Rows[0][ "NgayMuon" ] = dtNgayMuon.Value;
        dtDocGia.Rows[0][ "DiaChi" ] = txtDiaChi.Text;
        objDocGia.CapNhatTable_DocGia(dtDocGia);
        MessageBox.Show("Lưu sửa dữ liệu thành công!");
        NapListViewDSDocGia();
    }
    catch (Exception)
    {
        MessageBox.Show("Thất bại!", "Error");
    }
}
private void btnXoa_click(object sender, EventArgs e)
{
    if (lvwDanhSachDG.SelectedItems.Count > 0)
    {
        DialogResult dl = MessageBox.Show("Bạn có chắc muốn
xóa dữ liệu này?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
        if (dl == DialogResult.Yes)
        {
            DataRow[] dr = dtDocGia.Select("MaDG=' " +
txtMaDG.Text + " '");
            dr[0].Delete();
            objDocGia.CapNhatTable_DocGia(dtDocGia);
            // clear text
            txtMaDG.Text = "";
            txtTenDG.Text = "";
            rdNam.Checked = true;
            dtNgayMuon.Text = "";
            txtDiaChi.Text = "";
            grbChiTietDG.Enabled = false;
            NapListViewDSDocGia();
            MessageBox.Show("Xóa dữ liệu thành công!",
"Thông báo");
        }
    }
    else
    {
        MessageBox.Show("Hãy chọn dữ liệu cần xóa!", "Thông
báo");
    }
}

private void btnThoat_click(object sender, EventArgs e)
{
    this.Close();
}

```

```

        }

    private void btnSua_Click(object sender, EventArgs e)
    {
        if (lvwDanhSachDG.SelectedItems.Count > 0)
        {
            if (btnSua.Text == "Sửa")
            {
                btnSua.Text = "Hủy";
                grbChiTietDG.Enabled = true;
                btnThem.Enabled = false;
                btnXoa.Enabled = false;
                btnLuu.Enabled = true;
                dtNgayMuon.Text = "";
                txtMaDG.Focus();
            }
            else
            {
                btnSua.Text = "Sửa";
                grbChiTietDG.Enabled = false;
                btnThem.Enabled = true;
                btnXoa.Enabled = true;
                btnLuu.Enabled = false;
                dtNgayMuon.Text = "";
            }
            else
            {
                MessageBox.Show("Bạn hãy chọn dữ liệu cần sửa!",
"Thông báo");
            }
        }
    }

    private void DocGia_FormClosing(object sender,
FormClosingEventArgs e)
{
    DialogResult dl = MessageBox.Show("Bạn có muốn đóng
chương trình không?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
    if (dl == DialogResult.No)
        e.Cancel = true;
}

private void lvwDanhSachDG_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (lvwDanhSachDG.SelectedItems.Count > 0)
    {
        ListViewItem li =
lvwDanhSachDG.SelectedItems[lvwDanhSachDG.SelectedItems.Count - 1];
        foreach (ListViewItem l vw in
lvwDanhSachDG.SelectedItems)
        {
            txtMaDG.Text = l vw.SubItems[0].Text;
        }
    }
}

```

```

        txtTenDG.Text = lvw.SubItems[1].Text;
        dtNgayMuon.Text = lvw.SubItems[3].Text;
        txtDiaChi.Text = lvw.SubItems[4].Text;
    }
}

private void DocGia_Load(object sender, EventArgs e)
{
    NapListViewDSDocGia();
    grbChiTietDG.Enabled = false;
    rdNam.Checked = true;
}
}
}

```

## 6. Form Tác giả

1 → Input fields for Mã Tác Giả, Tên Tác Giả, and Địa Chỉ.

2 → List view displaying a list of authors (Danh sách tác giả).

3 → Toolbar buttons for Thêm, Lưu, Sửa, Xóa, and Thoát.

Mã Tác Giả	Tên Tác Giả	Địa Chỉ
TG001	Lê Nguyễn Hoài ...	Thành phố HCM
TG002	Nguyễn Anh Tuấn	Hà Nội
TG003	Phạm Mỹ Tiên	Thành phố HCM
TG004	Phan Gia Huy	Thái Bình
TG005	Phan Gia Bảo	Thái Bình
TG006	Phan Trúc Linh	Vũng Tàu
TG007	Phan Gia Hân	Vũng Tàu
TG008	Đinh Văn Thuong	Đà Lạt

STT	Biến cố	Xử lý
1	Nhập nội dung	Nhập vào thông tin yêu cầu.
2	Hiển thị	Hiển thị danh sách thông tin tác giả.
3	Chọn	Chọn thao tác cần thực hiện.

- Code của Form Tác giả:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using BUS;
namespace UI_Tier
{
    public partial class fmTacGia : Form
    {
        public fmTacGia()
        {
            InitializeComponent();
        }
        DataTable dtTacGia;
        B_TacGia objTacGia = new B_TacGia();

        private void setButton()
        {
            btnThem.Text = "Thêm";
            txtMaTG.Text = "";
            txtTenTG.Text = "";
            txtDiaChi.Text = "";
            txtMaTG.Focus();
            btnXoa.Enabled = true;
            btnSua.Enabled = true;
            lvwDanh SachTG.Enabled = true;
        }

        private void LuuThemTG()
        {
            try
            {
                lvwDanh SachTG.Items.Clear();
                DataRow dr = dtTacGia.NewRow();
                dr["MaTG"] = txtMaTG.Text;
                dr["TenTG"] = txtTenTG.Text;
                dr["DiaChi"] = txtDiaChi.Text;
                dtTacGia.Rows.Add(dr);
                objTacGia.CapNhatTable_TacGia(dtTacGia);
                setButton();
                NapListView();
                MessageBox.Show("Lưu mới tác giả thành công!");
            }
            catch (Exception)
            {
                NapListView();
                MessageBox.Show("Mã tác giả " + txtMaTG.Text + " đã
tồn tại\nVui lòng nhập mã tác giả khác!");
                txtMaTG.Focus();
            }
        }
        private void LuuSuaTG()
        {
            try

```

```

        {
            dtTacGia =
objTacGia.getTableTG_Theo_MaTG(lvwDanhSachTG.SelectedItems[0].Text);
            dtTacGia.Rows[0][ "MaTG" ] = txtMaTG.Text;
            dtTacGia.Rows[0][ "TenTG" ] = txtTenTG.Text;
            dtTacGia.Rows[0][ "DiaChi" ] = txtDiaChi.Text;
            objTacGia.CapNhatTable_TacGia(dtTacGia);
            MessageBox.Show("Lưu sửa dữ liệu thành công!");
            NapListView();
        }
        catch (Exception)
        {

            MessageBox.Show("Thất bại!");
        }
    }
    private void NapListView()
    {
        dtTacGia = objTacGia.getTableTacGia();
        lvwDanhSachTG.Items.Clear();
        foreach (DataRow dr in dtTacGia.Rows)
        {
            ListViewItem li =
lvwDanhSachTG.Items.Add(dr[ "MaTG" ].ToString());
            li.Tag = dr[ "MaTG" ].ToString();
            li.SubItems.Add(dr[ "TenTG" ].ToString());
            li.SubItems.Add(dr[ "DiaChi" ].ToString());
            li.ImageIndex = 0;
        }
    }

    private void FmTacGia_Load(object sender, EventArgs e)
    {
        NapListView();
        grbChiTietTG.Enabled = false;
    }

    private void FmTacGia_FormClosing(object sender,
FormClosingEventArgs e)
    {
        DialogResult dl = MessageBox.Show("Bạn có muốn đóng
chương trình không?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
        if (dl == DialogResult.No)
            e.Cancel = true;
    }

    private void LvxDanhSachTG_SelectedIndexChanged(object
sender, EventArgs e)
    {

        if (lvwDanhSachTG.SelectedItems.Count > 0)
    }

```

```

        foreach (ListViewItem lvw in
lvwDanhSachTG.SelectedItems)
        {
            ListViewItem item = new ListViewItem();
            txtMaTG.Text = lvw.SubItems[0].Text;
            txtTenTG.Text = lvw.SubItems[1].Text;
            txtDiaChi.Text = lvw.SubItems[2].Text;
        }
    }

private void BtnThem_Click_1(object sender, EventArgs e)
{
    if (btnThem.Text == "Thêm")
    {
        btnThem.Text = "Hủy";
        txtMaTG.Text = "";
        txtTenTG.Text = "";
        txtDiaChi.Text = "";
        txtMaTG.Focus();
        grpChiTietTG.Enabled = true;
        btnXoa.Enabled = false;
        btnSua.Enabled = false;
        //lvwDanhSachTG.Enabled = false;
    }
    else
    {
        setButton();
    }
}

private void BtnLuu_Click_1(object sender, EventArgs e)
{
    if (txtMaTG.Text == "")
    {
        MessageBox.Show("Chưa nhập mã tác giả!", "Error");
        txtMaTG.Focus();
    }
    else if (txtTenTG.Text == "")
    {
        MessageBox.Show("Chưa nhập tên tác giả!", "Error");
        txtTenTG.Focus();
    }
    else if (txtDiaChi.Text == "")
    {
        MessageBox.Show("Chưa nhập địa chỉ tác giả",
"Error");
        txtDiaChi.Focus();
    }
    else
    {
        if (lvwDanhSachTG.SelectedItems.Count == 0)
LuuThemTG();
    }
}

```

```

        else
            LuuSuaTG();
    }

private void BtnSua_Click_1(object sender, EventArgs e)
{
    if (lvwDanhSachTG.SelectedItems.Count > 0)
    {
        if (btnSua.Text == "Sửa")
        {
            btnSua.Text = "Hủy";
            grbChiTietTG.Enabled = true;
            txtMaTG.Focus();
            txtMaTG.SelectAll();
            btnThem.Enabled = false;
            btnXoa.Enabled = false;
            txtMaTG.Enabled = false;
        }
        else
        {
            btnSua.Text = "Sửa";
            grbChiTietTG.Enabled = false;
            btnThem.Enabled = true;
            btnXoa.Enabled = true;
        }
    }
    else
    {
        MessageBox.Show("Bạn chưa chọn dữ liệu cần sửa!");
    }
}

private void BtnXoa_Click_1(object sender, EventArgs e)
{
    if (lvwDanhSachTG.SelectedItems.Count > 0)
    {
        DialogResult dl = MessageBox.Show("Bạn có chắc muốn  
xóa dữ liệu này không?", "Warning", MessageBoxButtons.YesNo,  
MessageBoxIcon.Question);
        if (dl == DialogResult.Yes)
        {
            try
            {
                DataRow[] dr = dtTacGia.Select("MaTG=' " +  
txtMaTG.Text + "'");
                dr[0].Delete();
                objTacGia.CapNhatTable_TacGia(dtTacGia);
                // clear text
                txtMaTG.Text = "";
                txtTenTG.Text = "";
                txtDiaChi.Text = "";
                setButton();
                NapListView();
            }
        }
    }
}

```

```

        MessageBox.Show("Xóa thành công!");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
else
{
    MessageBox.Show("Chưa chọn dữ liệu cần xóa!");
}
}

private void BtnThoat_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

## 7. Form Quản lý mượn/ trả sách

### a. Quản lý mượn sách:

Quản Lý Mượn Sách

**Thông Tin Sách**

Chọn Mã Sách: 7 thói quen thành cô

Mã Sách :

Mã Loại :

Số Lượng :

Mã Tác Giả :

**Mượn Sách**

Mã Độc Giả : DG001

Mã Sách :

Số Lượng Mượn

Ngày Mượn : Wednesday, August 14, 2019

Ngày Hẹn Trả : Wednesday, August 14, 2019

**Danh sách cho mượn**

Mã Độc Giả	Mã Sách	Số Lượng	Ngày Mượn	Ngày Hẹn Trả
DG001	S001	1	7/3/2019	3/14/2019
DG002	S002	2	3/4/2019	10/3/2019
DG002	S005	1	5/4/2019	5/15/2019

STT	Biến cố	Xử lý
1	Chọn	Chọn tên sách muốn hiển thị.
2	Hiển thị	Hiển thị thông tin sách.
3	Chọn	Chọn mã độc giả mượn sách.
4	Nhập nội dung	Nhập thông tin và số lượng sách.
5	Chọn	Chọn ngày mượn và ngày hẹn trả sách.
6	Hiển thị	Hiển thị thông tin danh sách mượn sách.

7	Chọn	Chọn thao tác cần thực hiện.
8	Chọn	Chuyển qua tab Trả sách.

### b. Form Trả sách:

STT	Biến cód	Xử lý
1	Chọn	Chọn mã độc giả.
2	Nhập nội dung	Nhập thông tin và số lượng sách đã mượn.
3	Chọn	Chọn mốc thời gian: ngày mượn, ngày trả sách.
4	Hiển thị	Hiển thị danh sách thông tin trả sách.
5	Chọn	Chọn thao tác cần thực hiện.
6	Chọn	Chuyển qua tab Quản lý mượn sách.

- Code của Form Quản lý mượn/ trả sách:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
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 BUS;

namespace UI_Tier
{
    public partial class fmMuonTraSach : Form
    
```

```

{
    public fmMuonTraSach()
    {
        InitializeComponent();
    }
    B_MuonTraSach objMuonTra = new B_MuonTraSach();
    B_Sach objSach = new B_Sach();
    B_DocGia objDocGia = new B_DocGia();
    B_LoaiSach objLoaiSach = new B_LoaiSach();
    DataTable dtMuonTra, dtSach, dtDg;
    private void btnMuonMoi_Click(object sender, EventArgs e)
    {
        cbMaDG.Text = "";
        txtMaSach.Text = "";
        dtNgayMuon.Text = "";
        dtNgayHenTra.Text = "";
        txtSLMuon.Text = "";
    }

    private void btnChoMuon_Click(object sender, EventArgs e)
    {
        if (txtMaSach.Text == "")
        {
            MessageBox.Show("Chưa nhập mã sách!");
            txtMaSach.Focus();
        }
        else if (txtSLMuon.Text == "")
        {
            MessageBox.Show("Chưa nhập số lượng sách cho
mượn!");
            txtSLMuon.Focus();
        }
        else
        {
            try
            {
                lvwDanhSach.Items.Clear();
                DataRow dr = dtMuonTra.NewRow();
                dr["MaDG"] = cbMaDG.Text;
                dr["MaSach"] = txtMaSach.Text;
                dr["SoLuong"] = txtSLMuon.Text;
                dr["NgayMuon"] = dtNgayMuon.Value;
                dr["NgayHenTra"] = dtNgayHenTra.Value;

                MessageBox.Show(cbMaDG.SelectedItem.ToString() + "\n" +
                txtMaSach.Text + "\n" + txtSLMuon.Text + "\n" + dtNgayMuon.Value + "\n" +
                dtNgayHenTra.Value);
                dtMuonTra.Rows.Add(dr);
                objMuonTra.CapNhatTable_MuonTra(dtMuonTra);
                //setButton();
                NapListviewMuonSach();
                MessageBox.Show("Đã cho mượn thành công!");
            }
            catch (Exception ex)
        }
    }
}

```

```

        {
            MessageBox.Show("Mã sách " + txtMaSach.Text + " "
không có trong kho sách!\nHãy nhập mã sách khác!");
            MessageBox.Show(ex.Message);
            NapListviewMuonSach();
            txtMaSach.Focus();
        }
    }
}

private void NapListviewMuonSach()
{
    dtMuonTra = objMuonTra.getTableMuonTraSach();
    foreach (DataRow dr in dtMuonTra.Rows)
    {
        ListViewItem li = new ListViewItem();
        li.Text = dr["MaDG"].ToString();
        li.SubItems.Add(dr["MaSach"].ToString());
        li.SubItems.Add(dr["SoLuong"].ToString());
        string ngaymuon =
Convert.ToDateTime(dr["NgayMuon"].ToString()).ToShortDateString();
        li.SubItems.Add(ngaymuon);
        string ngayhentra =
Convert.ToDateTime(dr["NgayHenTra"].ToString()).ToShortDateString();
        li.SubItems.Add(ngayhentra);
        lvwDanhSach.Items.Add(li);
    }
}

private void btnTraSach_click(object sender, EventArgs e)
{
    DateTime ngayhentra = dtNgayHenTra_TraSach.Value;
    DateTime ngaytra = dtNgayTra.Value;
    if (ngaytra <= ngayhentra)
    {
        lblTinhTrangTraSach.Text = "Đúng hạn";
    }
    else
    {
        lblTinhTrangTraSach.Text = "Quá hạn";
    }
}

private void btnLuu_click(object sender, EventArgs e)
{
    SqlConnection connection = new SqlConnection(@"Data
Source=MINHTHAM\SQLEXPRESS;Initial Catalog=QLThuVien;Integrated
Security=True");
    connection.Open();
    string updateString = "Update MuonTraSach set NgayTra='"
+ dtNgayTra.Value + "' Where MaDG=N'" + cbMaDG_TraSach.Text + "' And
MaSach=N'" + txtMaSach_TraSach.Text + "'";
    SqlCommand command = new SqlCommand(updateString,
connection);
}

```

```

        command.ExecuteNonQuery();
        NapListviewTraSach();
        MessageBox.Show("Đã trả!");
    }
    private void NapListviewTraSach()
    {
        lvwDanhSachTra.Items.Clear();
        dtMuonTra = objMuonTra.getTableMuonTraSach();
        foreach (DataRow dr in dtMuonTra.Rows)
        {
            if (dr==null)
            {
                continue;
            }
            ListViewItem li =
lvwDanhSachTra.Items.Add(dr[ "MaDG" ].ToString());
            li.Text = dr[ "MaDG" ].ToString();
            li.SubItems.Add(dr[ "MaSach" ].ToString());
            li.SubItems.Add(dr[ "SoLuong" ].ToString());
            string ngaymuon =
Convert.ToDateTime(dr[ "NgayMuon" ].ToString()).ToShortDateString();
            li.SubItems.Add(ngaymuon);
            string ngayhentra =
Convert.ToDateTime(dr[ "NgayHenTra" ].ToString()).ToShortDateString();
            li.SubItems.Add(ngayhentra);

            if (dr[ "NgayTra" ].ToString() == string.Empty)
            {
                li.SubItems.Add("");
            }
            else
            {
                string ngaytra =
Convert.ToDateTime(dr[ "NgayTra" ].ToString()).ToShortDateString();
                li.SubItems.Add(ngaytra);
            }

            //lvwDanhSachTra.Items.Add(li);
        }
    }

    private void lvwDanhSach_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (lvwDanhSachTra.SelectedItems.Count > 0)
    {
        ListViewItem li =
lvwDanhSachTra.SelectedItems[lvwDanhSachTra.SelectedItems.Count - 1];
        foreach (ListViewItem l vw in
lvwDanhSachTra.SelectedItems)
        {

```

```

        cbMaDG_TraSach.SelectedValue =
lvw.SubItems[0].Text;
        txtMaSach_TraSach.Text = lvw.SubItems[1].Text;
        txtSoLuong_TraSach.Text = lvw.SubItems[2].Text;
        dtNgayMuon_TraSach.Text = lvw.SubItems[3].Text;
        dtNgayHenTra_TraSach.Text =
lvw.SubItems[4].Text;
        dtNgayTra.Text = lvw.SubItems[5].Text;
    }
}
}

private void cbChonMaSach_SelectedIndexChanged_1(object sender, EventArgs e)
{
    string strmasach =
cbChonMaSach.SelectedValue.ToString();
    dtSach = objSach.getTableSach_Theo_MaSach(strmasach);
    foreach (DataRow drw in dtSach.Rows)
    {
        lblMaSach.Text = drw["MaSach"].ToString();
        lblMaLoai.Text = drw["MaLoaiSach"].ToString();
        lblSoLuong.Text = drw["SoLuong"].ToString();
        lblMaTG.Text = drw["MaTG"].ToString();
    }
}

private void MuonTraSach_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult dl = MessageBox.Show("Bạn có muốn đóng
chương trình không?", "Warning", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);
    if (dl == DialogResult.No)
        e.Cancel = true;
}

private void lvwDanhSach_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lvwDanhSach.SelectedItems.Count > 0)
    {
        ListViewItem li =
lvwDanhSach.SelectedItems[lvwDanhSach.SelectedItems.Count - 1];
        foreach (ListViewItem lvw in
lvwDanhSach.SelectedItems)
        {
            cbMaDG.Text = lvw.SubItems[0].Text;
            txtMaSach.Text = lvw.SubItems[1].Text;
            txtSLMuon.Text = lvw.SubItems[2].Text;
            dtNgayMuon.Text = lvw.SubItems[3].Text;
            dtNgayHenTra.Text = lvw.SubItems[4].Text;
        }
    }
}

```

```

        }

    private void MuonTraSach_Load(object sender, EventArgs e)
    {
        NapListviewMuonSach();
        NapListviewTraSach();
        LoadComboBox();
        LoadComboBoxTraSach();
    }
    private void LoadComboBox()
    {
        //dtMuonTra = objMuonTra.getTableMuonTraSach();
        dtDg = objDocGia.getTableDocGia();
        cbMaDG.DataSource = dtDg;
        cbMaDG.DisplayMember = "MaDG";
        cbMaDG.ValueMember = "MaDG";
        dtSach = objSach.getTableSach();
        cbChonMaSach.DataSource = dtSach;
        cbChonMaSach.DisplayMember = "TenSach";
        cbChonMaSach.ValueMember = "MaSach";

    }
    private void LoadComboBoxTraSach()
    {
        dtMuonTra = objMuonTra.getTableMuonTraSach();
        cbMaDG_TraSach.DataSource = dtMuonTra;
        cbMaDG_TraSach.DisplayMember = "MaDG";
        cbMaDG_TraSach.ValueMember = "MaDG";
    }

    private void CbMaDG_SelectedIndexChanged(object sender,
EventArgs e)
    {
        string strDocGia = cbMaDG.SelectedValue.ToString();
        dtDg = objDocGia.getTableDocGia_Theo_MaDG(strDocGia);
    }

    private void btnThoat_click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}

```

## VII. Kết luận

### 1. Kết quả đạt được

- Phần mềm đã được hoàn thiện với giao diện đơn giản, dễ sử dụng, phù hợp với nhu cầu của người dùng.
- Hiệu được quy trình, nghiệp vụ quản lý thư viện: quy trình cập nhật sách, quy trình cập nhật loại sách, quy trình mượn trả sách, ...

### 2. Hạn chế

- Hệ thống thực hiện chức năng tìm kiếm còn đơn giản, chưa thực hiện được các thao tác tìm kiếm phức tạp.

### 3. Hướng phát triển

- Từ hạn chế trên, cần có một hướng phát triển mới để đề tài ngày càng hoàn thiện hơn.
- Tiếp tục nghiên cứu, xây dựng để phần mềm ngày càng hoàn thiện và có thể áp dụng vào thư viện của trường.
- Hỗ trợ người dùng tìm kiếm thông tin nhanh hơn, chính xác hơn.
- Thêm chức năng tạo báo cáo sách, độc giả mượn trả.

## VIII. Tổng kết

Qua quá trình thực hiện đề tài, với sự hướng dẫn của cô Thanh và bạn bè, nhóm em đã xây dựng thành công phần mềm Quản lý thư viện với các chức năng Tìm kiếm, Cập nhật, Quản lý sách, Quản lý mượn/ trả sách. Đề tài này sẽ góp phần hỗ trợ cho việc quản lý thông tin dễ dàng hơn thay vì phải quản lý bằng thủ công.

Hi vọng sau này sẽ biết được nhiều nguồn kiến thức có sẵn, nhóm em có thể học hỏi, nghiên cứu, áp dụng và làm nên nhiều ứng dụng tiện ích hơn nữa để phục vụ cuộc sống con người.

## IX. Tài liệu tham khảo

<https://expressmagazine.net/development/1495/mo-hinh-3-layers-3-tiers-trong-lap-trinh-quan-ly-net>

