**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------🙡🕮🙣----------**

****

**BÀI TẬP LỚN**

**MÔN LẬP TRÌNH WINDOWS**

**ĐỀ TÀI: HỆ THỐNG QUẢN LÝ THƯ VIỆN**

**Giáo viên : Phạm Văn Hà**

**Sinh viên thực hiện: Phạm Quốc Cường**

**Phạm Hải Dương**

**Phạm Văn Hiểu**

**Lớp: ĐH KTPM 3 – K11**

**Hà Nội – 2018**

MỤC LỤC

[**Lời Mở Đầu** 1](#_Toc532678437)

[**CHƯƠNG 1. KHẢO SÁT BÀI TOÁN VÀ PHÂN TÍCH YÊU CẦU** 2](#_Toc532678438)

[1.1. Khảo sát yêu cầu 2](#_Toc532678439)

[1.2. Giao diện 3](#_Toc532678440)

[1.2.1. Giao diện 3](#_Toc532678441)

[1.2.2. Dữ liệu 4](#_Toc532678442)

[**CHƯƠNG 2. THIẾT KẾ HỆ THỐNG (THIẾT KẾ CẤU TRÚC, GIAO DIỆN, CSDL)** 5](#_Toc532678443)

[2.1. Thiết kế cấu trúc 5](#_Toc532678444)

[2.2. Thiết kế cơ sở dữ liệu 6](#_Toc532678445)

[2.2.1. Sơ đồ cơ sở dữ liệu 6](#_Toc532678446)

[2.2.2. Mô tả cơ sở dữ liệu 6](#_Toc532678447)

[2.3. Thiết kế giao diện 8](#_Toc532678448)

[2.3.1. Giao diện Login 8](#_Toc532678449)

[2.3.2. Giao diện quản lý sách 9](#_Toc532678450)

[2.3.3. Giao diện Độc giả 9](#_Toc532678451)

[2.3.4 Giao diện Quản Lý Phiếu 10](#_Toc532678452)

[**Chương 3. Chương trình và các kết quả đạt được** 12](#_Toc532678453)

[3.1. Form Login 12](#_Toc532678454)

[3.2. Form QuanLySach 13](#_Toc532678455)

[3.2.1. PanelQuanLySach 14](#_Toc532678456)

[3.2.2. PanelDocGia 17](#_Toc532678457)

[3.2.3. tabQuanLyPhieu 21](#_Toc532678458)

[**Tổng kết** 26](#_Toc532678459)

# Lời Mở Đầu

Cùng với sự phát triển của nền kinh tế thì giáo dục cũng được mở rộng về quy mô và chất lượng để có thể cung cấp nhân lực có trình độ , chuyên môn…phục vụ cho nền kinh tế ngày càng vững mạnh. Bộ phận thư viện là bộ phận không thể thiếu để phục vụ cho việc nghiên cứu, tìm kiếm các tài liệu học tập một cách tốt nhất cho mọi người.

Thư viện quản lý khoảng hàng trăm ngàn đầu sách và tạp chí, phục vụ cho độc giả học tập tham khảo. Độc giả 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 .... Do đó chương trình quản lí thư viện ra đời sẽ giúp cho các thủ thư đỡ mất công tìm kiếm thủ công trong mớ tài liệu đồ sộ, giúp việc quản lí sách và độc giả dễ dàng,thuận tiện hơn.

Bài tập lớn này gồm 3 chương:

Chương 1: Khảo sát bài toán và phân tích yêu cầu

Chương 2: Thiết kế hệ thống (Thiết kế cấu trúc, giao diện, CSDL)

Chương 3: Chương trình và các kết quả đạt được

Trong quá trình tìm hiểu, biên soạn, bài tập lớn này khó tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp thêm để có nhiều kinh nghiệm hơn nữa trong cách làm bài tiểu luận, bài tập lớn, bài tốt nghiệp,....

*Trân trọng cảm ơn !*

**Nhóm sinh viên thực hiện**

# CHƯƠNG 1. KHẢO SÁT BÀI TOÁN VÀ PHÂN TÍCH YÊU CẦU

* 1. **Khảo sát yêu cầu**

|  |  |
| --- | --- |
| **Câu hỏi** | **Câu trả lời của khách hàng** |
| **1.** Anh chị có thể cho chúng tôi biết mục đích anh chị muốn có phần mềm không ạ? | Thư viện chúng tôi muốn có phần mềm để dễ duyệt, quản lý các thông tin của tài liệu, độc giả, danh sách các phiếu trong thư viện, mục đính giảm thiểu các hồ sơ giấy tờ. |
| **2.** Anh chị có yêu cầu cơ bản nào về phần mềm quản lý này ko? | Tôi muốn phần mêm phải có các chức năng cơ bản như: Thêm, Sửa, Xóa và Tìm Kiếm dữ liệu cho phần Quản Lý Sách, Quản Lý Độc Giả và Quản Lý Phiếu. |
| **3.** Anh chị muốn có tài khoản để quản lý chứ? | Vâng, tôi muốn có tài khoản với quyền quản trị nội dung. |
| **4.** Anh chị có muốn thêm các thông tin thủ công không? | Có, Tôi muốn có chức năng Thêm thủ công nữa. |
| **5.** Anh chị muốn Sách gồm những thông tin cụ thể nảo? | Tôi muốn thông tin về Sách phải có đủ: Mã sách, Tên sách, Tên thể loại, Số luongj, Tên tác giả. |
| **6.** Anh chị muốn Độc Giả gồm các thông tin cụ thể nào? | - Còn phía thông tin Độc Giả thì phải có các thông tin về: Mã độc giả, Tên độc giả, Giới tính, Địa chỉ, Số điện thoại.  - Tôi muốn có thêm bảng Phiếu Mượn gồm: Mã phiếu, Mã độc giả, Mã sách, Số lượng, Ngày mượn, Ngày trả. Và một bảng Thể Loại để lưu tên các thể loại. |
| **7.** Anh chị muốn chức năng Tìm Kiếm ra sao thì có thể đáp ứng yêu cầu anh chị? | Chúng tôi muốn làm thanh tìm kiếm để khi nhập dữ liệu chúng tôi có thể truy xuất được các thông tin. |
| **8.** Anh chị có dặn dò gì hay còn yêu cầu gì nữa không ạ? | Không ạ. |

* 1. **Giao diện**
     1. **Giao diện**
  + Màn hình đăng nhập (chỉ dành cho Admin).
  + Quản Lý Sách:
* Thêm thông tin sách (thủ công) vào các mục tương ứng.
* Các chức năng thêm, sửa, xóa sách.
* Gõ vào ô tìm kiếm để tìm thông tin sách.
* Độc Giả:
* Thêm thông tin độc giả (thủ công) vào các mục tương ứng.
* Các chức năng thêm, sửa, xóa độc giả.
* Chức năng lập phiếu mượn.
* Gõ vào ô tìm kiếm để tìm thông tin độc giả.
* Quản Lý Phiếu:
* Tab Quản Lý Phiếu:
* Các chức năng: Hiện phiếu mượn, Hiện phiếu trả, Lập phiếu trả, Xóa phiếu trả.
* Gõ vào ô tìm kiếm để tìm thông tin phiếu.
* Tab Lập Phiếu Mượn:
* Điền các thông tin cần thiết để lập phiếu mượn.
* Các chưc năng thêm sách, xóa sách, lập phiếu mươn.
* Tab Lập Phiếu Trả:
* Chọn độc giả muốn lập phiếu mượn.
* Các chứa năng xóa, lập phiếu trả.
* Giới Thiệu:
* Sơ qua về phần mềm.
  + 1. **Dữ liệu**
* Thông tin về Tài khoản: Tên tài khoản, Mật khẩu.
* Thông tin về Sách:Mã sách, Tên sách, Mã thể loại, Số lượng, tên tác giả.
* Thông tin về Độc giả: Mã độc giả, Tên độc giả, Giới tính, Địa chỉ, Số điện thoại.
* Thông tin về Phiếu: Mã phiếu, Mã độc giả, Mã sách, Số lượng, Ngày mượn, Ngày trả .
* Thông tin về Thể loại: Mã thể loại, Tên thể loại.

# CHƯƠNG 2. THIẾT KẾ HỆ THỐNG (THIẾT KẾ CẤU TRÚC, GIAO DIỆN, CSDL)

* 1. **Thiết kế cấu trúc**

**DTO**

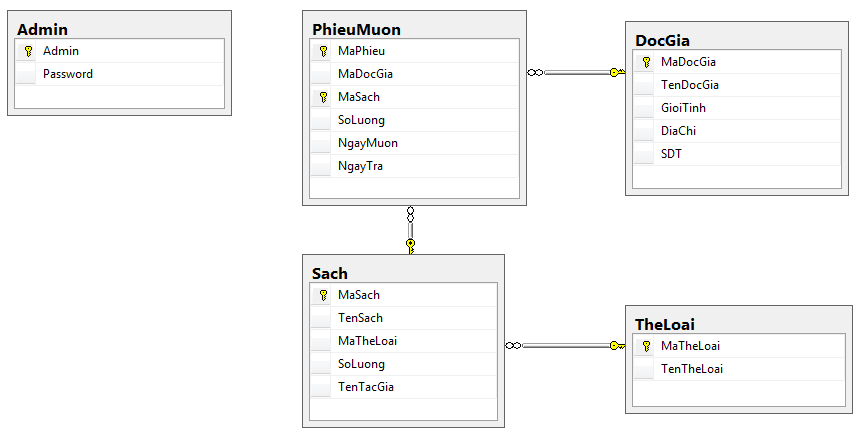
**GUI**

**BUS**

**DAO**

**DATA**

* Chương trình sử dụng mẫu thiết kế Singleton Pattern là mẫu thiết kế đảm bảo rằng một lớp chỉ có một thể hiện (instance) duy nhất và trong đó cung cấp một cổng giao tiếp chung nhất để truy cập vào lớp đó.
* Việc sử dụng Singleton Pattern đem lại các lợi ích sau:
* Quản lý việc truy cập tốt hơn vì chỉ có một thể hiện đơn nhất.
* Cho phép cải tiến lại các tác vụ (operations) và các thể hiện (representation) do pattern có thể được kế thừa và tùy biến lại thông qua một thể hiện của lớp con
* Quản lý số lượng thể hiện của một lớp, không nhất thiết chỉ có một thể hiện mà có số thể hiện xác định.
* Khả chuyển hơn so với việc dùng một lớp có thuộc tính là static, vì việc dùng lớp static chỉ có thể sử dụng một thể hiện duy nhất, còn Singleton Pattern cho phép quản lý các thể hiện tốt hơn và tùy biến theo điều kiện cụ thể.
  1. **Thiết kế cơ sở dữ liệu**
     1. **Sơ đồ cơ sở dữ liệu**

****

* + 1. **Mô tả cơ sở dữ liệu**
       1. **Mô tả bảng Sach**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | MaSach | nvarchar | Là khóa chính để phân biệt với các sách khác, mỗi sách có 1 mã sách |
| 2 | TenSach | nvarchar | Tên của sách |
| 3 | MaTheLoai | nvarchar | Thuộc tính để phân biệt các loại sách(Khóa ngoại) |
| 4 | SoLuong | int | Cho biết số lượng của mỗi cuốn sách trong danh mục sách |
| 5 | TenTacGia | nvarchar | Tên tác giả của sách |

* + - 1. **Mô tả bảng DocGia**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | MaDocGia | nvarchar | Là khóa chính để phân biệt với các độc giả khác, mỗi độc giả có 1 mã độc giả |
| 2 | TenDocGia | nvarchar | Tên của độc giả |
| 3 | GioiTinh | nvarchar | Giới tính độc giả |
| 4 | DiaChi | nvarchar | Địa chỉ của độc giả |
| 5 | SDT | nvarchar | Số điện thoại của độc giả |

* + - 1. **Mô tả bảng PhieuMuon**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | MaPhieu | nvarchar | Là khóa chính để phân biệt với các phiếu khác, mỗi phiếu có 1 mã phiếu |
| 2 | MaDocGia | nvarchar | Mã độc giả mượn sách |
| 3 | MaSach | nvarchar | Mã sách độc giả mượn |
| 4 | SoLuong | int | Số lượng sách mà độc giả mượn |
| 5 | NgayMuon | date | Ngày độc giả mượn sách |
| 6 | NgayTra | date | Ngày độc giả trả sách |

* + - 1. **Mô tả bảng TheLoai**

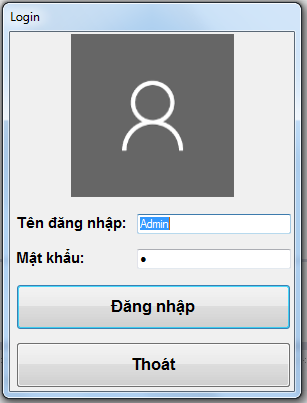
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | MaTheLoai | nvarchar | Là khóa chính để phân biệt với các thể loại khác, mỗi thể loại có 1 mã thể loại |
| 2 | TenTheLoai | nvarchar | Mã độc giả mượn sách |

* + - 1. **Mô tả bảng Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | Admin | nvarchar | Tên người quản trị |
| 2 | Password | nchar | Mật khẩu |

## **2.3. Thiết kế giao diện**

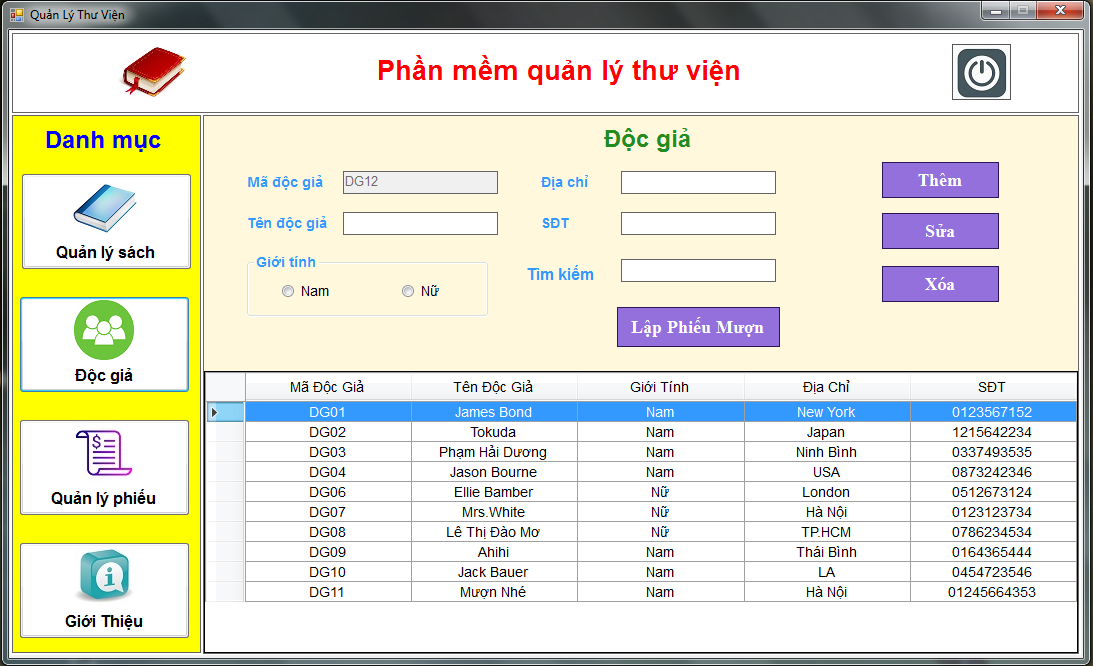
### **2.3.1. Giao diện Login**



### **2.3.2. Giao diện quản lý sách**

****

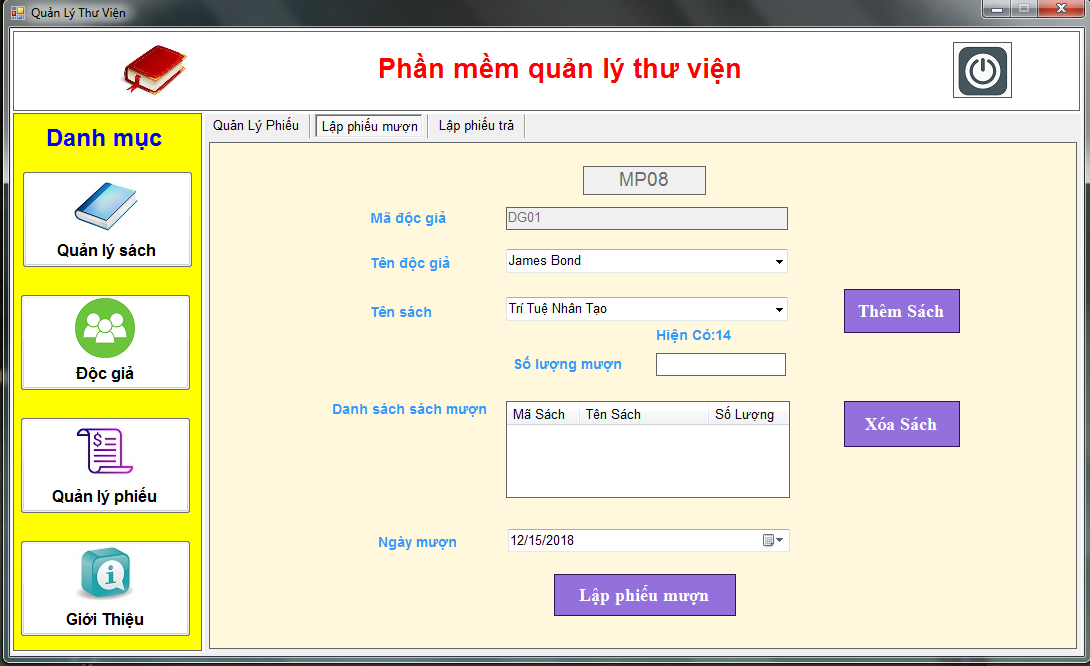
### **2.3.3. Giao diện Độc giả**



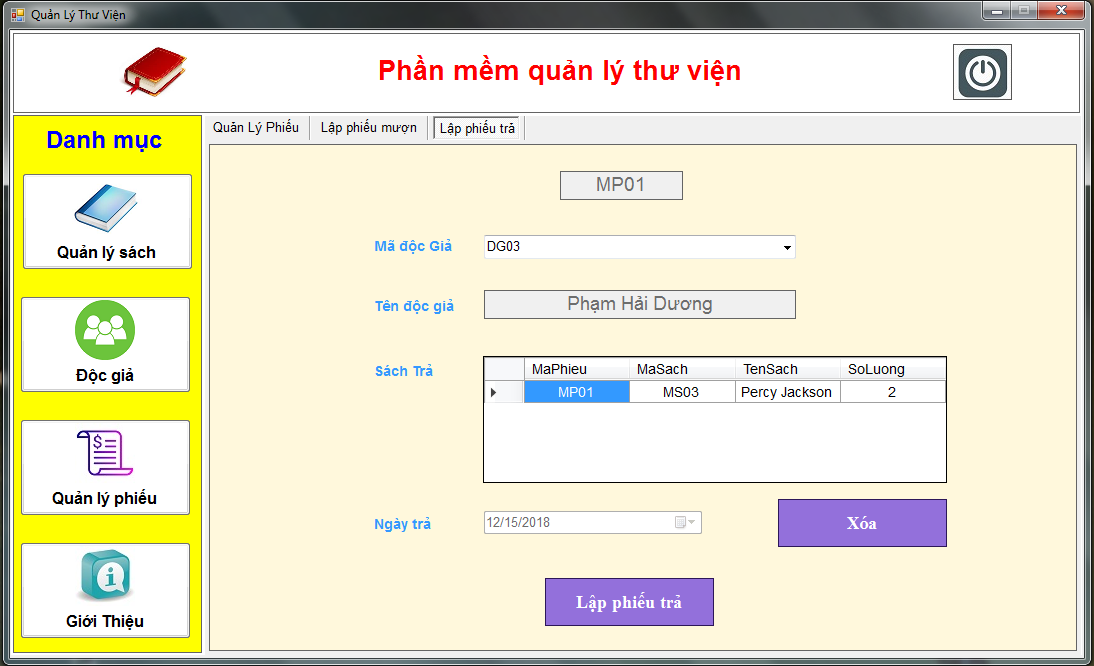
### **2.3.4** **Giao diện Quản Lý Phiếu**

****

**2.3.4.2 Giao diện lập phiếu mượn**

****

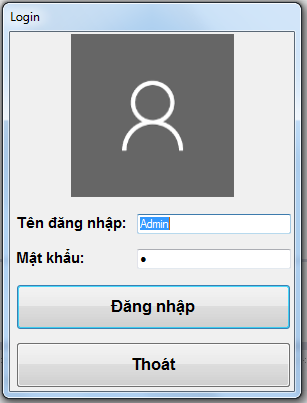
**2.3.4.3 Giao diện lập phiếu trả**

****

1

# Chương 3. Chương trình và các kết quả đạt được

* 1. **Form Login**



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

{

Account account = new Account(txtlogin.Text, txtpassword.Text);

if (AccountBUS.Instance.login(account))

{

QuanLySach f = new QuanLySach();

f.ShowDialog();

this.Close();

}

else MessageBox.Show("Sai tên tài khoản hoặc mật khẩu");

}

Hàm kiểm tra xem đã nhập đúng tên tài khoản và mật khẩu chưa. Nếu đúng **form Login** sẽ bị đóng , **form QuanLySach** được hiện lên. Ngược lại thông báo “Sai tên tài khoản hoặc mật khẩu”.

* 1. **Form QuanLySach**
* **Form QuanLySach** được thiết kế với các panel lồng nhau bao gồm:
* panelQuanlysach
* panelDocGia
* tabQuanLyPhieu
* panelAbout
* Sau khi đăng nhập thành công **Form QuanLySach** load dữ liệu lên:

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

{

this.docGiaTableAdapter1.Fill(this.\_QuanLyThuVien1\_0DataSet1.DocGia);

this.sachTableAdapter.Fill(this.\_QuanLyThuVien1\_0DataSet1.Sach); this.phieuMuonTableAdapter.Fill(this.\_QuanLyThuVien1\_0DataSet1.PhieuMuon);

dataGridView1.DataSource = PhieuMuonBUS.Instance.LoadPhieu();

dtgQuanLySach.DataSource = SachBUS.Instance.ShowSach();

dtgDocGia.DataSource = DocGiaBUS.Instance.ShowDG();

dgvPhieuMuon.DataSource = PhieuMuonBUS.Instance.LoadPhieuMuon();

//đổ dữ liệu lên combobox Thể loại sách

cmbTheLoai.DataSource = SachBUS.Instance.GetCatory();

cmbTheLoai.DisplayMember = "TenTheLoai";

cmbTheLoai.ValueMember = "MaTheLoai";

//đổ dữ liệu lên combobox Tên sách

cmbTenSachPhieu.DataSource = SachBUS.Instance.ShowSach();

cmbTenSachPhieu.DisplayMember = "TenSach";

cmbTenSachPhieu.ValueMember = "MaSach";

//do du lieu len combobox Ten Doc Gia

cmbTenDGPhieu.DataSource = DocGiaBUS.Instance.ShowDG();

cmbTenDGPhieu.DisplayMember = "TenDocGia";

cmbTenDGPhieu.ValueMember = "MaDocGia";

//đổ dữ liệu lên combobox phiếu trả

cboDGPT.DataSource = PhieuMuonBUS.Instance.GetMaDG();

cboDGPT.ValueMember = "MaDocGia";

//Tăng mã tự động

tangMa("DG",dtgDocGia,txtMaDG);

tangMa("MS", dtgQuanLySach,txtMaSach);

tangMa("MP", dataGridView1, txtmaPhieu);

txtMaSach.Enabled = false;

txtMaDG.Enabled = false;

//tô màu sach co so luong =0

for (int i = 0; i < dtgQuanLySach.RowCount; i++)

{

if (int.Parse(dtgQuanLySach.Rows[i].Cells[3].Value.ToString()) ==0)

dtgQuanLySach.Rows[i].Cells[3].Style.BackColor =Color.Red;

}

}

* Phần mềm sử dụng tăng mã sách, mã độc giả, mã phiếu tự động:

private void tangMa(string ch,DataGridView dataGridView,TextBox textBox)

{

if (dataGridView.Rows.Count == 0)

textBox.Text = ch + "00";

else

{

string str = (dataGridView.Rows[dataGridView.Rows.Count - 1].Cells[0].Value.ToString()).Remove(0, 2);

Int32 temp = Int32.Parse(str);

textBox.Text = ch + (temp + 1 < 10 ? "0" : "") + (temp + 1);

}

}

* + 1. **PanelQuanLySach**



* Sau khi đăng nhập thành công màn hình hiện lên là **panQuanLySach** hoặc cũng có thể mở **panQuanLySach** bằng việc click vào “Quản lý sách’ khi đang ở các panel khác.
* Thêm sách: Người dùng nhập dữ liệu cho sách mới và click vào nút “thêm” để thêm sách.

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

{

if (txtTenSach.TextLength == 0) MessageBox.Show("Tên sách không được để trống.");

else if (txtSoLuong.TextLength == 0) MessageBox.Show("Vui lòng nhập số lượng");

else

{

try

{

Int32 flag = 0;

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

{

string tenSach = dtgQuanLySach.Rows[i].Cells[1].Value.ToString().ToLower();

string tacgia = dtgQuanLySach.Rows[i].Cells[4].Value.ToString().ToLower();

if (txtTenSach.Text.ToLower() == tenSach && txtTacGia.Text.ToLower() == tacgia)

{

MessageBox.Show("Sách đã có rồi.");

flag = 1;

}

}

if(flag==0)

{

Sach sach = new Sach(txtMaSach.Text, txtTenSach.Text, (cmbTheLoai.SelectedValue.ToString()), int.Parse(txtSoLuong.Text), txtTacGia.Text);

SachBUS.Instance.AddBook(sach);

QuanLySach\_Load(sender, e);

tangMa("MS", dtgQuanLySach, txtMaSach);

}

}

catch (SqlException)

{

MessageBox.Show("Trùng mã Sách.");

}

catch(FormatException)

{

MessageBox.Show("Nhập sai kiểu dữ liệu.");

}

}

}

* Sửa sách: Người dùng chọn sách cần sửa, sau đó thay đổi thông tin muốn và click “Sửa”.

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

{

if (txtTenSach.TextLength == 0) MessageBox.Show("Tên sách không được để trống.");

else if (txtSoLuong.TextLength == 0) MessageBox.Show("vui lòng nhập số lượng");

else

{

try

{

Sach sach = new Sach(txtMaSach.Text, txtTenSach.Text, cmbTheLoai.SelectedValue.ToString(), int.Parse(txtSoLuong.Text), txtTacGia.Text);

SachBUS.Instance.UpdateBook(sach);

QuanLySach\_Load(sender, e);

}

catch (FormatException)

{

MessageBox.Show("Nhập sai kiểu dữ liệu.");

}

}

}

* Xóa sách: Người dùng chọn sách cần xóa , sau đó click “Xóa”.

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

{

try

{

DialogResult dlr = MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if(dlr == DialogResult.Yes)

{

string ma = txtMaSach.Text;

SachBUS.Instance.DeleteBook(ma);

QuanLySach\_Load(sender, e);

}

}

catch

{

return;

}

}

* Tìm kiếm sách: Người dùng nhập dữ liệu vào ô tìm kiếm, kết quả tìm kiếm sẽ được hiển thị sau mỗi lần nhập.

private void txtTimKiem\_TextChanged(object sender, EventArgs e)

{

string dk = txtTimKiem.Text;

dtgQuanLySach.DataSource= SachBUS.Instance.LookBook(dk);

}

public DataTable LookBook(string dk)

{

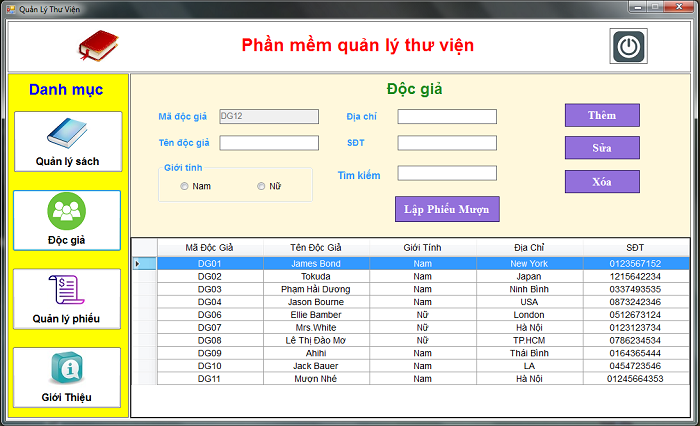
string query = "Select MaSach,TenSach,TenTheLoai,SoLuong,TenTacGia '%" + dk + "%' OR TenSach LIKE N'%" + dk + "%' OR TenTheLoai LIKE N'%" + dk + "%' OR TenTacGia LIKE N'%" + dk + "%' OR SoLuong LIKE '%"+dk+"%' ) ";

DataTable data= DataProvider.Instance.ExecuteQuery(query);

return data;

}

* + 1. **PanelDocGia**
* Sau khi click “Độc giả” panDocGia hiện lên:



* Thêm độc giả: người dùng nhập dữ liệu cho độc giả mới và click vào nút “thêm” để thêm độc giả.

private void btnThemDG\_Click(object sender, EventArgs e) //Thêm DG

{

try

{

if (txtTenDG.Text == "")

MessageBox.Show("Nhập tên độc giả.");

else if (rdoNam.Checked == false && rdoNu.Checked == false)

MessageBox.Show("Chọn giới tính.");

else if (txtDiaChi.Text == "")

MessageBox.Show("Nhập địa chỉ.");

else

{

String gioitinh;

if (rdoNam.Checked) gioitinh = "Nam";

else gioitinh = "Nữ";

Double value;

if (Double.TryParse(txtSDT.Text, out value) == false)

MessageBox.Show("Số điện thoại phải là số.");

else if (txtSDT.TextLength < 10 || txtSDT.TextLength > 11)

MessageBox.Show("Số điện thoại gồm 10 hoặc 11 số.");

else

{

Int32 flag = 0;

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

{

string tenDG = dtgDocGia.Rows[i].Cells[1].Value.ToString().ToLower();

string sdt = dtgDocGia.Rows[i].Cells[4].Value.ToString().ToLower();

if (txtTenDG.Text.ToLower() == tenDG && txtSDT.Text.ToLower() == sdt)

{

MessageBox.Show("Độc giả đã có rồi.");

flag = 1;

}

}

if(flag==0)

{

DocGia docgia = new DocGia(txtMaDG.Text, txtTenDG.Text, gioitinh, txtDiaChi.Text, txtSDT.Text);

DocGiaBUS.Instance.ThemDG(docgia);

QuanLySach\_Load(sender, e);

MessageBox.Show("Thêm thành công.");

tangMa("DG", dtgDocGia, txtMaDG);

}

}

}

}

catch (SqlException)

{

MessageBox.Show("Trùng mã độc giả.");

}

}

* Sửa độc giả: Người dùng chọn độc giả cần sửa, sau đó thay đổi thông tin muốn và click “Sửa”.

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

{

try

{

if (txtTenDG.Text == "" && rdoNam.Checked == false && rdoNu.Checked == false && txtDiaChi.Text == "" && txtSDT.Text == "")

MessageBox.Show("Chọn độc giả.");

else

{

String gioitinh;

if (rdoNam.Checked) gioitinh = "Nam";

else gioitinh = "Nữ";

Double value;

if (Double.TryParse(txtSDT.Text, out value) == false)

MessageBox.Show("Số điện thoại phải là số.");

else if (txtSDT.TextLength < 10 || txtSDT.TextLength > 11)

MessageBox.Show("Số điện thoại gồm 10 hoặc 11 số.");

else

{

DocGia docgia = new DocGia(txtMaDG.Text, txtTenDG.Text, gioitinh, txtDiaChi.Text, txtSDT.Text);

DocGiaBUS.Instance.SuaDG(docgia);

QuanLySach\_Load(sender, e);

MessageBox.Show("Sửa thành công.");

}

}

}

catch

{

return;

}

}

* Xóa độc giả : Người dùng chọn độc giả cần xóa, sau đó click “Xóa” để xóa độc giả đã chọn.

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

{

try

{

DialogResult dlr = MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dlr == DialogResult.Yes)

{

if (txtTenDG.Text == "" && rdoNam.Checked == false && rdoNu.Checked == false && txtDiaChi.Text == "" &&

txtSDT.Text == "")

MessageBox.Show("Chọn độc giả.");

else

{

DocGiaBUS.Instance.XoaDG(txtMaDG.Text);

QuanLySach\_Load(sender, e);

}

}

}

catch

{

return;

}

}

* Lập phiếu mượn: người dùng chọn độc giả cần lập phiếu,sau đó click ”Lập phiếu mượn”.

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

{

if (txtTenDG.Text == "")

MessageBox.Show("Vui lòng chọn độc giả.");

else

{

panQuanLySach.Hide();

panDocGia.Hide();

tabQuanLyPhieu.Show();

txtMaDGPhieu.Text = txtMaDG.Text;

cmbTenDGPhieu.SelectedValue = txtMaDGPhieu.Text;

tabQuanLyPhieu.SelectedTab = tabQuanLyPhieu.TabPages[1];

}

}

* Tìm kiếm độc giả: Người dùng nhập dữ liệu vào ô tìm kiếm, kết quả tìm kiếm sẽ được hiển thị sau mỗi lần nhập.

private void txtTimKiemDocGia\_TextChanged(object sender, EventArgs e)

{

String str = txtTimKiemDocGia.Text;

dtgDocGia.DataSource = DocGiaBUS.Instance.TimKiemDG(str);

}

* + 1. **tabQuanLyPhieu**
* tabQLPhieu:



* Người dùng click “Hiện phiếu mượn” bảng phiếu mượn sẽ hiện ra



* Người dùng click ”Hiện phiếu trả” bảng phiếu trả hiện lên:



* Xóa phiếu trả: người dùng chọn phiếu cần xóa sau đó click “Xóa phiếu trả” để xóa phiếu.

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

{

try

{

DialogResult dlr = MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if(dlr == DialogResult.Yes)

{

int currentIndex = dgvPhieuTra.CurrentCell.RowIndex;

string mp = dgvPhieuTra.Rows[currentIndex].Cells[0].Value.ToString();

string mdg2 = dgvPhieuTra.Rows[currentIndex].Cells[1].Value.ToString();

string ms2 = dgvPhieuTra.Rows[currentIndex].Cells[2].Value.ToString();

PhieuMuonBUS.Instance.DeletePhieu(mp, mdg2, ms2);

dgvPhieuTra.DataSource = PhieuMuonBUS.Instance.LoadPhieuTra();

}

}

catch

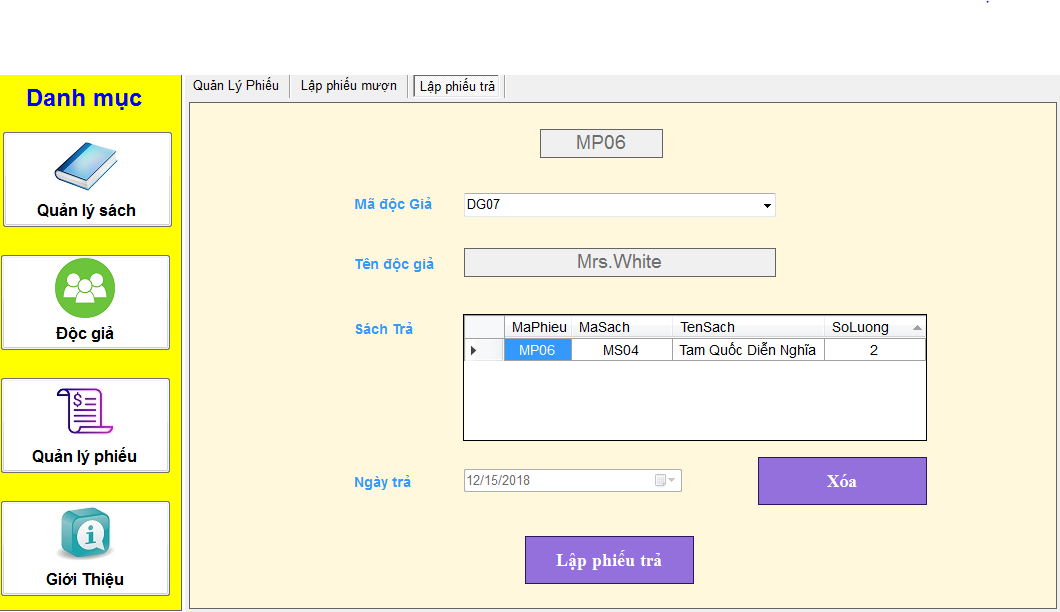
{

return;

}

}

* Lập phiếu trả:người dùng chọn phiếu mượn cần lập sau đó click ”Lập phiếu trả” ,tabLPhieu mượn được mở ra với thông tin của độc giả cần trả sách.



* Người dùng click “Lập phiếu trả ” để hoàn thành lập phiếu trả.

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

{

DateTime ngaytra = dpkNgayTra.Value;

if (cboDGPT.SelectedIndex == -1)

MessageBox.Show("Chọn độc giả cần lập phiếu.");

else

{

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

{

string sltra = dgvSachTra.Rows[i].Cells[3].Value.ToString();

string ms = dgvSachTra.Rows[i].Cells[1].Value.ToString();

PhieuMuonBUS.Instance.AddPhieuTra(ngaytra, ms, cboDGPT.Text); SachBUS.Instance.UpdateSLBook(int.Parse(SachBUS.Instance.getSLuong(ms)) + int.Parse(sltra),ms);

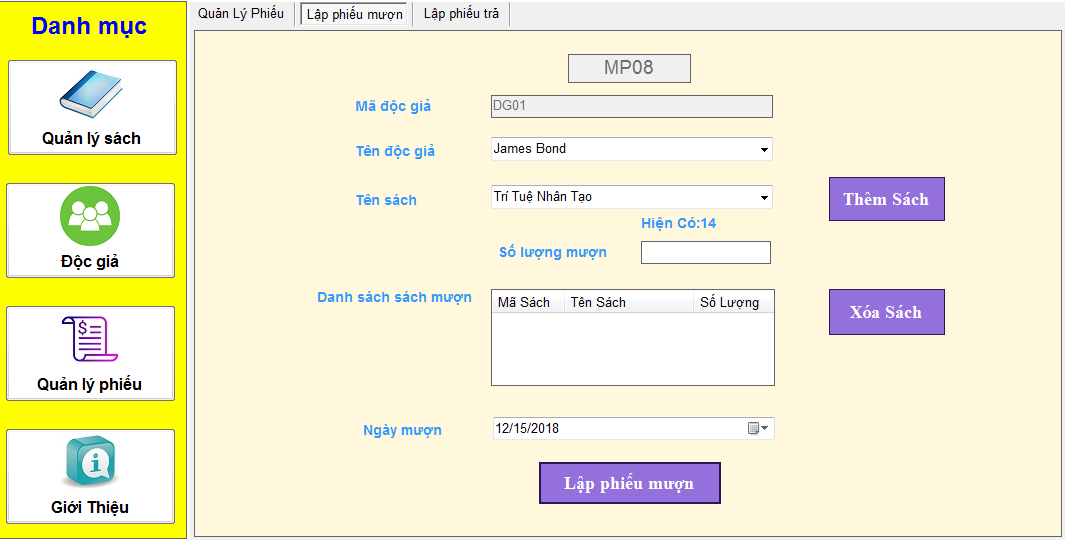
}

dgvSachTra.DataSource = PhieuMuonBUS.Instance.GetSachTra(cboDGPT.Text);

QuanLySach\_Load(sender, e);

}

* tabLPhieuMuon



* Người dùng click “Lập phiếu mượn” để hoàn thành lập phiếu.

private void btnLapPhieu\_Click\_1(object sender, EventArgs e)

{

if (txtMaDGPhieu.TextLength == 0) MessageBox.Show("Mã Độc Giả Trống.");

else if (cmbTenSachPhieu.SelectedIndex == -1) MessageBox.Show("Vui lòng chọn sách");

else if (lstSachMuon.Items.Count == 0)

{

MessageBox.Show("Vui lòng thêm sách mượn.");

}

else

{

for (int i = 0; i < lstSachMuon.Items.Count; i++)

{

PhieuMuon phieu = new PhieuMuon(txtmaPhieu.Text, txtMaDGPhieu.Text, lstSachMuon.Items[i].SubItems[0].Text, int.Parse(lstSachMuon.Items[i].SubItems[2].Text), dpkNgayMuon.Value);

PhieuMuonBUS.Instance.AddPhieuMuon(phieu);

SachBUS.Instance.UpdateSLBook(int.Parse(SachBUS.Instance.getSLuong(lstSachMuon.Ite ms[i].SubItems[0].Text)) - int.Parse(lstSachMuon.Items[i].SubItems[2].Text), lstSachMuon.Items[i].SubItems[0].Text);

}

lstSachMuon.Items.Clear();

QuanLySach\_Load(sender, e);

}

}

# *Tổng kết*

* **Ưu điểm**
* Chương trình có giao diện thân thiện , thuận tiện cho người sử dụng.
* Các chức năng như tìm kiếm , thêm , sửa , xóa… được thực hiện nhanh chóng và thuận tiện .
* Phù hợp với các ứng dụng triển khai trên diện rộng.
* Dễ dàng sử dụng và quản lý.
* **Nhược điểm**
* Do thời gian nghiên cứu có hạn và khả năng còn hạn chế nên một số vẫn còn trên ý tưởng mà chưa được thực thi.
* Chương trình có tính chuyên nghiệp chưa cao ,chức năng còn ít .
* Chưa giải quyết trọn vẹn các vẫn đề nảy sinh trong quá trình quản lý.
* **Hướng phát triển của phần mềm**
* Tìm hiểu và hoàn thiện một số chức năng còn chưa được thực hiện trong bài tập lớn.
* Hỗ trợ cơ sở dữ liệu SQL để khả năng bảo mật , tốc độ truy vấn , khả năng quản lý dữ liệu tốt hơn

*Đây là sản phẩm của các thành viên nhóm em. Nếu có gì sai sót mong thầy góp ý để sản phẩm của chúng em được hoàn thiện hơn ạ .*

***Em xin chân thành cảm ơn !***