**TRƯỜNG ĐẠI HỌC QUY NHƠN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

HỌC PHẦN LẬP TRÌNH TRÊN DESKTOP

QUẢN LÝ THƯ VIỆN

**Sinh viên:** Nguyễn Thị Thanh Hương

**Lớp:** Công nghệ thông tin 44D

**Giảng viên:** Đinh Thị Mỹ Cảnh

**Quy Nhơn**, Tháng 6/Năm 2023

**MỤC LỤC**

[**CHƯƠNG 1. GIỚI THIỆU** 1](#_Toc136445223)

[**CHƯƠNG 2. THIẾT KẾ CƠ SỞ DỮ LIỆU** 2](#_Toc136445224)

[**2.1. Sơ đồ cơ sở dữ liệu** 2](#_Toc136445225)

[**2.2. Mô tả cơ sở dữ liệu** 2](#_Toc136445226)

[**CHƯƠNG 3. THIẾT KẾ GIAO DIỆN** 4](#_Toc136445227)

[**3.1. Data Provider** 4](#_Toc136445228)

[**3.2. Màn hình Login** 7](#_Toc136445229)

[**3.3. Màn hình Registration** 11](#_Toc136445234)

[**3.4. Màn hình Trang chủ** 16](#_Toc136445239)

[**3.5. Màn hình Đổi mật khẩu** 18](#_Toc136445241)

[**3.6. Màn hình Quản lý Sách** 23](#_Toc136445246)

[**3.7. Màn hình Quản lý Độc giả** 34](#_Toc136445251)

[**3.8. Màn hình Quản lý Mượn sách** 44](#_Toc136445256)

[**Chương 4. KẾT LUẬN** 55](#_Toc136445261)

# 

# **CHƯƠNG 1. GIỚI THIỆU**

Hiện nay nhu cầu, học tập nghiên cứu tại các trường học là rất lớn. Vì vậy công tác quản lý thư viện là vấn đề trọng điểm trong việc cung cấp tài liệu cho học sinh, sinh viên. Nhưng đa phần hiện nay các thư viện đều áp dụng mô hình tổ chức chưa hợp lý trong quá trình quản lý thư viện. Hệ thống thư viện nhiều nơi còn sơ sài trong vấn đề quản lý (xét về mặt cập nhật thông tin cho người dùng cũng như thuận tiện cho thủ thư trong vấn đề quản lý) khiến cho công tác mượn trả sách, cũng như công việc của thủ thư hiện nay mất thời gian và độ chính xác không được đảm bảo.

Việc độc giả gặp khó khăn trong việc làm thẻ, công tác mượn sách diễn ra một cách thủ công. Người mượn phải tự tìm tài liệu bằng cách tự tìm trong nhiều cuốn sách có sẵn trong thư viện. Dẫn đến việc độc giả không chủ động trong việc mượn sách. Ngoài ra, thủ thư phải quản lý công việc chủ yếu bằng giấy tờ khiến cho hiệu suất công việc giảm cũng như độ chính xác không được cao, thống kê thông tin rất mất thời gian.

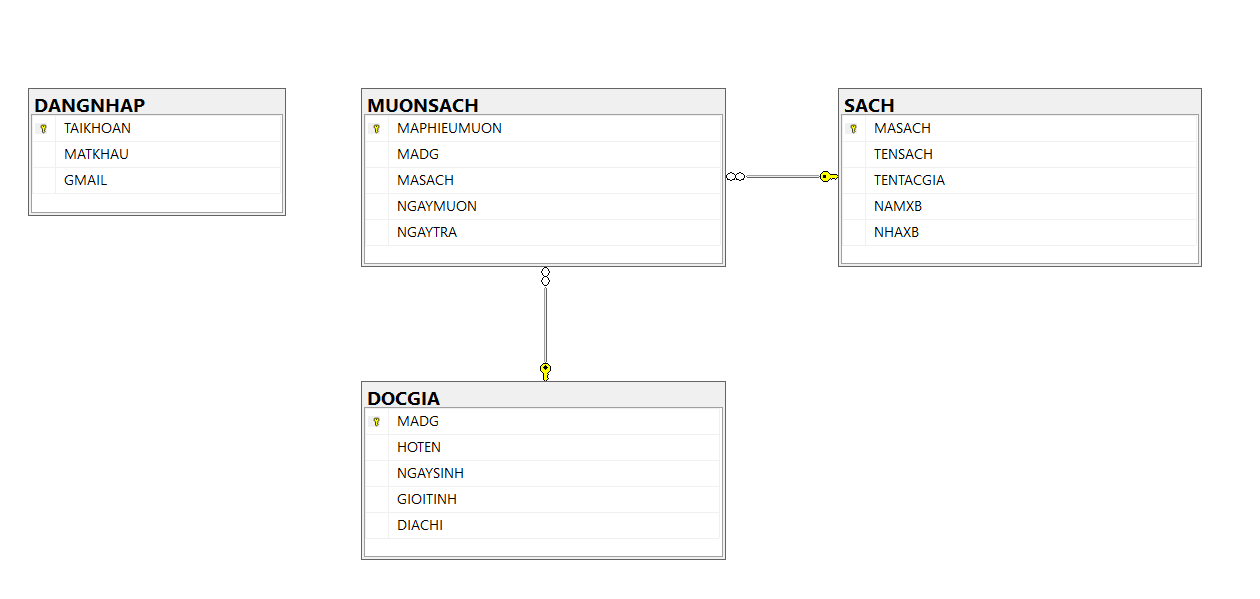
Vậy nên, thông qua học phần “Lập trình trên Desktop” em đã được học tập và trang bị kiến thức để có thể làm một sản phẩm nhỏ gồm một vài chức năng cơ bản thuận tiện trong việc Quản lý thư viện hiện nay.

Em xin chân thành cảm ơn cô Đinh Thị Mỹ Cảnh đã tận tình truyền dạy những kiến thức cần thiết và hướng dẫn em hoàn thành bài tập này này.

Trong quá trình thực hiện bài tập lớn này, không thể tránh khỏi những sai sót, em mong được nhận sự phản hồi góp ý của cô để em rút kinh nghiệm và hoàn thiện hơn.

# **CHƯƠNG 2. THIẾT KẾ CƠ SỞ DỮ LIỆU**

# **2.1. Sơ đồ cơ sở dữ liệu**



**Hình 2.1.** Diagram Quản lý thư viện.

# **2.2. Mô tả cơ sở dữ liệu**

a. Mô tả bảng DANGNHAP

Gồm: TAIKHOAN, MATKHAU, GMAIL.

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| TAIKHOAN | VARCHAR | Là khóa chính để phân biệt với các tài khoản khác, mỗi tài khoản có một tên khác nhau |
| MATKHAU | MATKHAU | Mật khẩu của tài khoản người dùng. |
| GMAIL | GMAIL | Gmail của tài khoản. |

b. Mô tả bảng MUONSACH

Gồm: MAPHIEUMUON, MADG, MASACH, NGAYMUON, NGAYTRA

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| MAPHIEUMUON | VARCHAR | Là khóa chính để phân biệt với các mã phiếu mượn khác |
| MADG | VARCHAR | Mã độc giả mượn sách |
| MASACH | VARCHAR | Mã sách mà độc giả mượn |
| NGAYMUON | DATETIME | Ngày mượn sách |
| NGAYTRA | DATETIME | Ngày trả sách |

1. Mô tả bảng SACH

Gồm: MASACH, TENSACH, TENTACGIA, NAMXB, NHAXB.

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| MASACH | VARCHAR | Là khóa chính để phân biệt với các loại sách khác, mỗi sách có một mã sách. |
| TENSACH | NVARCHAR | Tên của sách. |
| TENTACGIA | NVARCHAR | Tên của tác giả. |
| NAMXB | INT | Năm xuất bản của sách |
| NHAXB | NVARCHAR | Nhà xuất bản của sách |

1. Mô tả bảng DOCGIA

Gồm: MADG, HOTEN, NGAYSINH, GIOITINH, DIACHI

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| MADG | VARCHAR | Là khóa chính để phân biệt với các độc giả khác, mỗi độc giả có một mã riêng biệt |
| HOTEN | NVARCHAR | Tên của độc giả |
| NGAYSINH | DATETIME | Ngày sinh của độc giả |
| GIOITINH | BIT | Giới tính : 1 là Nam, 0 là Nữ |
| DIACHI | NVARCHAR | Địa chỉ nơi ở của độc giả |

# **CHƯƠNG 3. THIẾT KẾ GIAO DIỆN**

# **3.1. Data Provider**

static public class DataProvider

{

public static SqlConnection cnn;

public static SqlDataAdapter da;

public static SqlCommand cmd;

/// <summary>

/// Phương thức mở kết nối nếu có lỗi xảy ra thì nó sẽ ném lỗi và trả về cho tầng Bussiness(BUS)

/// </summary>

public static void moKetNoi()

{

cnn = new SqlConnection();

//Liên kết đến đường dẫn trong file App.Config

cnn.ConnectionString = ConfigurationManager.ConnectionStrings["Connect"].ConnectionString.ToString();

try

{

if (cnn.State == ConnectionState.Closed)

cnn.Open();

}

catch (Exception ex)

{

throw ex;

}

}

/// <summary>

/// Phương thức đóng kết nối

/// </summary>

public static void dongKetNoi()

{

try

{

if (cnn.State == ConnectionState.Open)

cnn.Close();

}

catch (Exception ex)

{

throw ex;

}

cnn.Dispose(); //Giải phóng bộ nhớ

}

/// <summary>

/// Phương thức thực hiện câu lệnh strSQL truy vấn dữ liệu và kết quả trả về là một DataTable

///Dùng cho các câu lệnh SELECT

/// </summary>

/// <param name="sql"></param>

/// <returns></returns>

public static DataTable getTable(string sql)

{

moKetNoi();

da = new SqlDataAdapter(sql, cnn);

DataTable dt = new DataTable();

da.Fill(dt);

dongKetNoi();

return dt;

}

/// <summary>

/// Phương thức cập nhật dữ liệu, thực hiện các câu lệnh Thêm, Xóa, Sửa

/// </summary>

/// <param name="sql"></param>

/// <param name="name"></param>

/// <param name="value"></param>

public static bool updateData(string sql, string[] name = null, object[] value = null)

{

moKetNoi();

cmd = new SqlCommand(sql, cnn);

//Xóa parameters của các dữ liệu trước và sau khi thực hiện vòng for

cmd.Parameters.Clear();

if (value != null)

{

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

{

cmd.Parameters.AddWithValue(name[i], value[i]);

}

}

if(cmd.ExecuteNonQuery() > 0)

{

cmd.Dispose();

dongKetNoi();

return true;

}

dongKetNoi();

return false;

}

/// <summary>

/// Kiểm tra dữ liệu dùng count(\*)

/// </summary>

/// <param name="sql"></param>

/// <returns></returns>

public static int executeScalar(string sql)

{

moKetNoi();

int i = 0;

cmd = new SqlCommand(sql, cnn);

i = (int)cmd.ExecuteScalar();

dongKetNoi();

return i;

}

/// <summary>

/// Kiểm tra đăng nhập

/// </summary>

/// <param name="sql"></param>

/// <param name="name"></param>

/// <param name="var"></param>

/// <returns></returns>

public static bool checkLogin(string sql, string name, string var)

{

string user = null;

moKetNoi();

cmd = new SqlCommand(sql, cnn);

cmd.Parameters.AddWithValue(name, var);

user = (string)cmd.ExecuteScalar();

dongKetNoi();

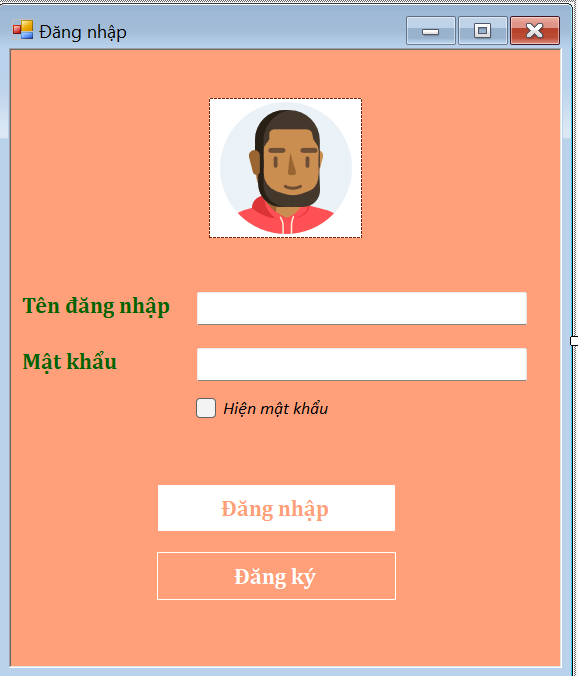
return user != null;

}

}

# **3.2. Màn hình Login**

## **3.2.1. GUI**



**Hình 3.2.1.** Màn hình đăng nhập

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Tên đăng nhập | Textbox | Người dùng nhập tên đăng nhập |
| Mật khẩu | Textbox | Nguời dùng nhập mật khẩu |
| Hiện mật khẩu | Checkbox | Hiển thị mật khẩu |
| Đăng nhập | Button | Đăng nhập khi đã điền đủ thông tin |
| Đăng ký | Button | Đăng ký khi chưa có tài khoản |

1. Câu lệnh cho nút Đăng nhập

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

{

if(txtTenDangNhap.Text != "" && txtMatKhau.Text != "")

{

string tenTaiKhoan = txtTenDangNhap.Text;

string matKhau = txtMatKhau.Text;

if (busDangNhap.checkTaiKhoan(tenTaiKhoan))

{

if (busDangNhap.checkMatKhau(matKhau))

{

MessageBox.Show("Đăng nhập thành công !");

frmMain f = new frmMain();

f.Show();

this.Hide();

}

else

{

MessageBox.Show("Mật khẩu không chính xác!");

}

}

else

{

MessageBox.Show("Tên đăng nhập không chính xác !");

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin đăng nhập !");

}

}

1. Câu lệnh cho nút Đăng Ký

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

{

frmDangKy f = new frmDangKy();

f.Show();

this.Hide();

}

1. Câu lệnh cho Checkbox Hiện mật khẩu

private void chkHienMatKhau\_CheckedChanged(object sender, EventArgse)

{

if (chkHienMatKhau.Checked)

{

txtMatKhau.PasswordChar = (char)0;

}

else

{

txtMatKhau.PasswordChar = '⁕';

}

}

## **3.2.2. DTO**

public class DTO\_DangNhap

{

private string taiKhoan;

private string matKhau;

private string gmail;

/\*========= GETTER/SETTER =========\*/

public string TaiKhoan

{

get

{

return taiKhoan;

}

set

{

taiKhoan = value;

}

}

public string MatKhau

{

get

{

return matKhau;

}

set

{

matKhau = value;

}

}

public string Gmail

{

get

{

return gmail;

}

set

{

gmail = value;

}

}

/\*========== CONSTRUCTOR ==========\*/

public DTO\_DangNhap()

{

}

public DTO\_DangNhap(string tk, string mk, string gmail = "")

{

this.taiKhoan = tk;

this.matKhau = mk;

this.gmail = gmail;

}

}

## **3.2.3. BUS**

/\*============= ĐĂNG NHẬP =============\*/

// Phương thức kiểm tra tên tài khoản

public bool checkTaiKhoan(string tentaikhoan)

{

return dalDangNhap.checkTaiKhoan(tentaikhoan);

}

//Phương thức kiểm tra mật khẩu

public bool checkMatKhau(string matkhau)

{

return dalDangNhap.checkMatKhau(matkhau);

}

## **3.2.4. DAL**

/\*============= ĐĂNG NHẬP =============\*/

//Kiểm tra tên tài khoản có trong CSDL không

public bool checkTaiKhoan(string tentaikhoan)

{

string sql = "SELECT TAIKHOAN FROM DANGNHAP WHERE TAIKHOAN = @taikhoan";

string name = "@taikhoan";

return DataProvider.checkLogin(sql, name, tentaikhoan);

}

//Kiểm tra mật khẩu có trong CSDL không

public bool checkMatKhau(string matkhau)

{

string sql = "SELECT MATKHAU FROM DANGNHAP WHERE MATKHAU = @matkhau";

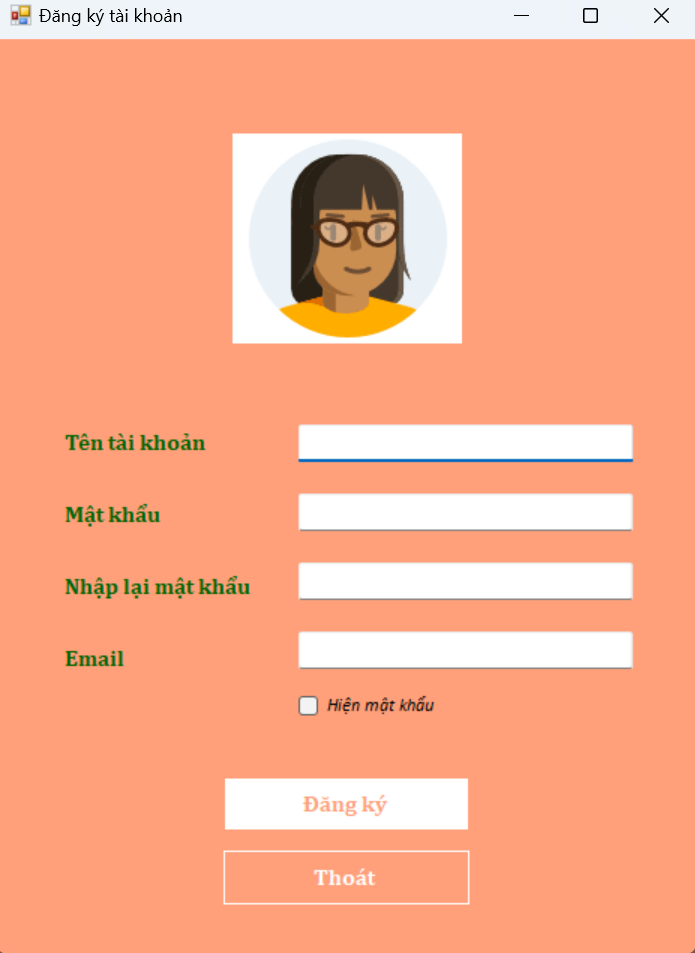
string name = "@matkhau";

return DataProvider.checkLogin(sql, name, matkhau);

}

# **3.3. Màn hình Registration**

## **3.3.1. GUI**



**Hình 3.3.1.** Màn hình Registration

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Tên tài khoản | Textbox | Người dùng nhập tên tài khoản |
| Mật khẩu | Textbox | Người dùng nhập mật khẩu |
| Nhập lại mật khẩu | Textbox | Người dùng nhập lại giống mật khẩu đã nhập ở ô trên |
| Email | Textbox | Người dùng nhập Gmail |
| Hiện mật khẩu | Checkbox | Hiển thị mật khẩu |
| Đăng ký | Button | Đăng ký khi hoàn thành các thông tin |
| Thoát | Button | Thoát form Đăng ký quay về form Đăng nhập |

1. Câu lệnh cho Checkbox Hiện mật khẩu

private void chkHienMatKhau\_CheckedChanged(object sender, EventArgs e)

{

if (chkHienMatKhau.Checked)

{

txtMatKhau.PasswordChar = (char)0;

txtNhapLaiMk.PasswordChar = (char)0;

}

else

{

txtMatKhau.PasswordChar = '⁕';

txtNhapLaiMk.PasswordChar = '⁕';

}

}

1. Câu lệnh cho nút Đăng ký

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

{

if(txtTenTaiKhoan.Text != "" && txtMatKhau.Text != "" && txtNhapLaiMk.Text != "" && txtEmail.Text != "")

{

string tenTK = txtTenTaiKhoan.Text;

string matKhau = txtMatKhau.Text;

string nhapLaiMk = txtNhapLaiMk.Text;

string gmail = txtEmail.Text;

if(busDangKy.checkTenTaiKhoan(tenTK) == 0)

{

if (matKhau.Equals(nhapLaiMk))

{

DTO\_DangKy dk = new DTO\_DangKy(tenTK, matKhau, gmail);

if (busDangKy.dangKy(dk))

{

MessageBox.Show("Đăng ký tài khoản thành công !");

frmDangNhap f = new frmDangNhap();

f.Show();

this.Hide();

}

}

else

{

MessageBox.Show("Mật khẩu không khớp. Vui lòng nhập lại mật khẩu !");

}

}

else

{

MessageBox.Show("Tên tài khoản đã được đăng ký. Vui lòng chọn tên khác");

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin !");

}

}

1. Câu lệnh cho nút Thoát

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

{

frmDangNhap f = new frmDangNhap();

f.Show();

this.Hide();

}

## **3.3.2. DTO**

public class DTO\_DangNhap

{

private string taiKhoan;

private string matKhau;

private string gmail;

/\*========= GETTER/SETTER =========\*/

public string TaiKhoan

{

get

{

return taiKhoan;

}

set

{

taiKhoan = value;

}

}

public string MatKhau

{

get

{

return matKhau;

}

set

{

matKhau = value;

}

}

public string Gmail

{

get

{

return gmail;

}

set

{

gmail = value;

}

}

/\*========== CONSTRUCTOR ==========\*/

public DTO\_DangNhap()

{

}

public DTO\_DangNhap(string tk, string mk, string gmail = "")

{

this.taiKhoan = tk;

this.matKhau = mk;

this.gmail = gmail;

}

}

## **3.3.3. BUS**

/\*============= ĐĂNG KÝ =============\*/

//Phương thức thêm thông tin một người dùng mới vào cơ sở dữ liệu

public bool dangKy(DTO\_DangNhap dk)

{

return dalDangNhap.dangKy(dk);

}

//Phương thức kiểm tra tên tài khoản đã có trong cơ sở dữ liệu hay chưa

public int checkTenTaiKhoan(string taikhoan)

{

return dalDangNhap.checkTenTaiKhoan(taikhoan);

}

## **3.3.4. DAL**

/\*============= ĐĂNG KÝ =============\*/

/// <summary>

/// Thêm người dùng mới

/// </summary>

/// <param name="dk"></param>

/// <returns></returns>

public bool dangKy(DTO\_DangNhap dk)

{

string sql = "INSERT INTO DANGNHAP VALUES (@taikhoan, @matkhau, @gmail)";

string[] name = { "@taikhoan", "@matkhau", "@gmail" };

object[] value = { dk.TaiKhoan, dk.MatKhau, dk.Gmail };

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Kiểm tra tên tài khoản đã có trong cơ sở dữ liệu hay chưa

/// </summary>

/// <param name="taikhoan"></param>

/// <returns></returns>

public int checkTenTaiKhoan(string taikhoan)

{

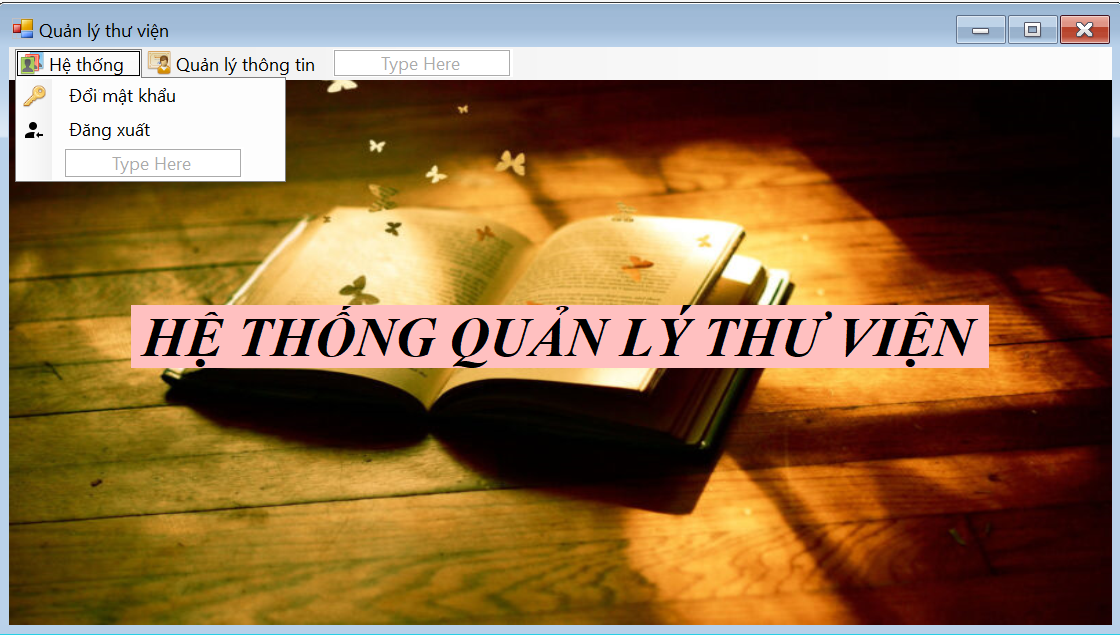
string sql = string.Format("SELECT COUNT(\*) FROM DANGNHAP WHERE TAIKHOAN='{0}'", taikhoan);

return DataProvider.executeScalar(sql);

}

# **3.4. Màn hình Trang chủ**

## **3.4.1. GUI**



**Hình 3.4.1.a.** Màn hình Trang chủ



**Hình 3.4.1.b.** Màn hình Trang chủ

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Hệ thống quản lý Thư viện | Label | Hiển thị tiêu đề cho form Trang chủ |
| Đổi mật khẩu | MenuStrip | Hiển thị form đổi mật khẩu |
| Đăng xuất | MenuStrip | Hiển thị form đăng nhập khi người dùng nhấn vào |
| Sách | MenuStrip | Hiển thị form Quản lý sách |
| Độc giả | MenuStrip | Hiển thị form Quản ký độc giả |
| Mượn trả sách | MenuStrip | Hiển thị form Quản lý mượn trả sách |

1. Câu lệnh cho toolStrip Đổi mật khẩu

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

{

frmDoiMatKhau f = new frmDoiMatKhau();

f.Show();

this.Hide();

}

1. Câu lệnh cho toolStrip Đăng xuất

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

{

frmDangNhap f = new frmDangNhap();

f.Show();

this.Hide();

}

1. Câu lệnh cho toolStrip Quản lý Sách

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

{

frmSach f = new frmSach();

f.Show();

this.Hide();

}

1. Câu lệnh cho toolStrip Quản lý độc giả

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

{

frmDocGia f = new frmDocGia();

f.Show();

this.Hide();

}

1. Câu lệnh cho toolStrip Mượn trả sách

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

{

frmMuonSach f = new frmMuonSach();

f.Show();

this.Hide();

}

# **3.5. Màn hình Đổi mật khẩu**

## **3.5.1. GUI**



**Hình 3.5.1.** Màn hình Đổi mật khẩu

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Tên đăng nhập | Textbox | Người dùng nhập tên đăng nhập. |
| Mật khẩu cũ | Textbox | Người dùng nhập mật khẩu cũ (mật khẩu đang sử dụng). |
| Mật khẩu mới | Textbox | Người dùng nhập mật khẩu mới. |
| Nhập lại mật khẩu mới | Textbox | Người dùng nhập lại mật khẩu mới. |
| Hiện mật khẩu | Checkbox | Hiển thị mật khẩu. |
| Đổi mật khẩu | Button | Đổi mật khẩu khi điền đầy đủ thông tin. |
| Thoát | Button | Trở về trang chủ. |

1. Câu lệnh cho Checkbox Hiện mật khẩu

private void chkHienMatKhau\_CheckedChanged(object sender, EventArgs e)

{

if (chkHienMatKhau.Checked)

{

txtMatKhauCu.PasswordChar = (char)0;

txtMatKhauMoi.PasswordChar = (char)0;

txtNhapLaiMkMoi.PasswordChar = (char)0;

}

else

{

txtMatKhauCu.PasswordChar = '⁕';

txtMatKhauMoi.PasswordChar = '⁕';

txtNhapLaiMkMoi.PasswordChar = '⁕';

}

}

1. Câu lệnh cho nút Đổi mật khẩu

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

{

if (txtTenDangNhap.Text != "" && txtMatKhauCu.Text != "" && txtMatKhauMoi.Text != "" && txtNhapLaiMkMoi.Text != "")

{

string tenDangNhap = txtTenDangNhap.Text;

string matKhauCu = txtMatKhauCu.Text;

string matKhauMoi = txtMatKhauMoi.Text;

DTO\_DangNhap dtoDoiMatKhau = new DTO\_DangNhap(tenDangNhap, matKhauMoi);

if (busDoiMatKhau.checkTaiKhoan(tenDangNhap))

{

if (busDoiMatKhau.checkMatKhau(matKhauCu))

{

if (txtMatKhauMoi.Text.Equals(txtNhapLaiMkMoi.Text))

{

if (busDoiMatKhau.doiMatKhau(dtoDoiMatKhau))

{

MessageBox.Show("Đổi mật khẩu thành công");

frmDangNhap f = new frmDangNhap();

f.Show();

this.Hide();

}

else

{

MessageBox.Show("Đổi mật khẩu không thành công");

}

}

else

{

MessageBox.Show("Mật khẩu không khớp. Vui lòng kiểm tra lại");

}

}

else

{

MessageBox.Show("Mật khẩu cũ không chính xác. Vui lòng kiểm tra lại");

}

}

else

{

MessageBox.Show("Tên đăng nhập không đúng");

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin");

}

}

1. Câu lệnh cho nút Thoát

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

{

frmMain f = new frmMain();

f.Show();

this.Hide();

}

## **3.5.2. DTO**

public class DTO\_DangNhap

{

private string taiKhoan;

private string matKhau;

private string gmail;

/\*========= GETTER/SETTER =========\*/

public string TaiKhoan

{

get

{

return taiKhoan;

}

set

{

taiKhoan = value;

}

}

public string MatKhau

{

get

{

return matKhau;

}

set

{

matKhau = value;

}

}

public string Gmail

{

get

{

return gmail;

}

set

{

gmail = value;

}

}

/\*========== CONSTRUCTOR ==========\*/

public DTO\_DangNhap()

{

}

public DTO\_DangNhap(string tk, string mk, string gmail = "")

{

this.taiKhoan = tk;

this.matKhau = mk;

this.gmail = gmail;

}

}

## **3.5.3. BUS**

/\*============= ĐỔI MẬT KHẨU =============\*/

//Phương thức cập nhật mật khẩu của người dùng trong cơ sở dữ liệu

public bool doiMatKhau(DTO\_DangNhap matkhau)

{

return dalDangNhap.doiMatKhau(matkhau);

}

## **3.5.4. DAL**

/\*============= ĐỔI MẬT KHẨU =============\*/

/// <summary>

/// Sửa mật khẩu

/// </summary>

/// <param name="matkhau"></param>

/// <returns></returns>

public bool doiMatKhau(DTO\_DangNhap matkhau)

{

string sql = string.Format("UPDATE DANGNHAP SET MATKHAU = @matkhau WHERE TAIKHOAN= '{0}'", matkhau.TaiKhoan);

string[] name = { "@matkhau" };

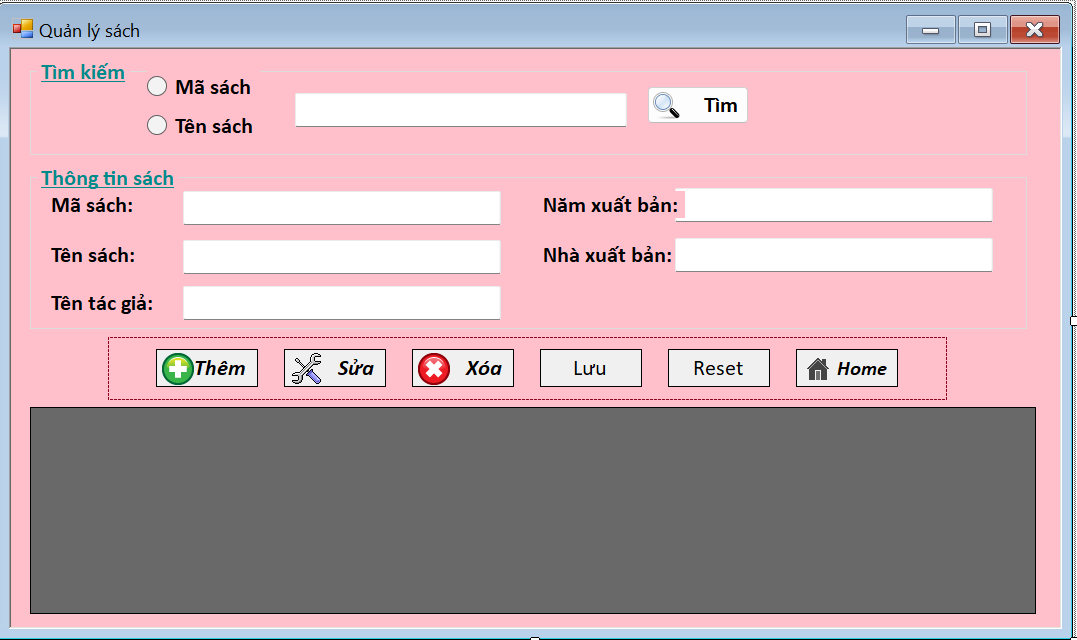
object[] value = { matkhau.MatKhau };

return DataProvider.updateData(sql, name, value);

}

# **3.6. Màn hình Quản lý Sách**

## **3.6.1. GUI**



**Hình 3.6.1.** Màn hình quản lý Sách

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Mã sách | Radio button | Tìm kiếm sách theo mã sách. |
| Tên sách | Radio button | Tìm kiếm sách theo tên sách. |
| Tìm | Button | Thực hiện chức năng tìm kiếm sách theo mã hoặc tên sách. |
| Mã sách | Textbox | Hiển thị thông tin mã sách. |
| Tên sách | Textbox | Hiển thị thông tin tên sách. |
| Tên tác giả | Textbox | Hiển thị thông tin tên tác giả. |
| Năm xuất bản | Textbox | Hiển thị thông tin năm xuất bản. |
| Nhà xuất bản | Textbox | Hiển thị thông tin nhà xuất bản. |
| Thêm | Button | Thực hiện chức năng thêm một cuốn sách mới khi người dùng nhấn vào. |
| Sửa | Button | Thực hiện chức năng sửa thông tin cuốn sách sách khi người dùng nhấn vào. |
| Xóa | Button | Thực hiện chức năng xóa sách khi người dùng nhấn vào. |
| Lưu | Button | Thực hiện chức năng lưu thông tin sách khi thêm sách mới một cuốn sách. |
| Reset | Button | Làm mới các Textbox có trên form |
| Home | Button | Quay trở về trang chủ |

1. Câu lệnh cho nút Tìm

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

{

if (txtTimKiem.Text == "")

{

MessageBox.Show("Vui lòng nhập nội dung cần tìm");

}

else

{

DTO\_Sach sach = new DTO\_Sach();

DataTable dt = new DataTable();

if (rdoMaSach.Checked)

{

sach.MaSach = txtTimKiem.Text;

dt = busSach.timKiemTheoMa(sach);

txtTimKiem.Clear();

txtTimKiem.Focus();

}

else if (rdoTenSach.Checked)

{

sach.TenSach = txtTimKiem.Text;

dt = busSach.timKiemTheoTenSach(sach);

txtTimKiem.Clear();

txtTimKiem.Focus();

}

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy sách");

}

else

{

dgvSach.DataSource = dt;

}

}

1. Câu lệnh cho nút Thêm

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

{

txtMaSach.Enabled = true;

txtMaSach.Text = "";

txtTenSach.Text = "";

txtTenTG.Text = "";

txtNamXB.Text = "";

txtNXB.Text = "";

btnSave.Enabled = true;

btnReset.Enabled = true;

btnXoa.Enabled = false;

btnSua.Text = "Hủy";

btnThem.Enabled = false;

dgvSach.Enabled = false;

dgvSach.DataSource = busSach.loadDataSach();

}

1. Câu lệnh cho nút Sửa

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

{

int i = dgvSach.CurrentCell.RowIndex;

if (i >= 0)

{

if (btnSua.Text == "Hủy")

{

txtMaSach.Enabled = false;

btnSave.Enabled = false;

btnReset.Enabled = false;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Text = "Xóa";

btnSua.Text = "Sửa";

btnThem.Enabled = true;

dgvSach.DataSource = busSach.loadDataSach();

dgvSach.Enabled = true;

}

else

{

if (txtMaSach.Text != "" && txtTenSach.Text != "" && txtTenTG.Text != "" && txtNamXB.Text != "" && txtNXB.Text != "")

{

//Lấy thông tin từ các Control

string maSach = txtMaSach.Text;

string tenSach = txtTenSach.Text;

string tenTacGia = txtTenTG.Text;

int namXB = int.Parse(txtNamXB.Text);

string nhaXB = txtNXB.Text;

DTO\_Sach sach = new DTO\_Sach(maSach, tenSach, tenTacGia, namXB, nhaXB);

if (busSach.suaSach(sach))

{

MessageBox.Show("Sửa thông tin sách thành công !");

dgvSach.DataSource = busSach.loadDataSach();

}

else

{

MessageBox.Show("Sửa thông tin sách không thành công !");

}

}

else

{

MessageBox.Show("Vui lòng chọn thông tin sách cần sửa");

}

}

}

}

1. Câu lệnh cho nút Xóa

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

{

if (btnXoa.Text == "Hủy")

{

txtMaSach.Enabled = false;

btnSave.Enabled = false;

btnReset.Enabled = false;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Text = "Xóa";

btnSua.Text = "Sửa";

btnThem.Enabled = true;

dgvSach.DataSource = busSach.loadDataSach();

dgvSach.Enabled = true;

}

else

{

if (txtMaSach.Text != "" && txtTenSach.Text != "" && txtTenTG.Text != "" && txtNamXB.Text != "" && txtNXB.Text != "")

{

string maSach = txtMaSach.Text;

string tenSach = txtTenSach.Text;

string tenTacGia = txtTenTG.Text;

int namXB = int.Parse(txtNamXB.Text);

string nhaXB = txtNXB.Text;

DTO\_Sach sach = new DTO\_Sach(maSach, tenSach, tenTacGia, namXB, nhaXB);

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

if (result == DialogResult.Yes)

{

if (busSach.xoaSach(sach))

{

MessageBox.Show("Xóa sách thành công !");

lamMoi();

dgvSach.DataSource = busSach.loadDataSach();

}

else

{

MessageBox.Show("Xóa sách không thành công !");

}

}

}

else

{

MessageBox.Show("Vui lòng chọn sách cần xóa");

}

}

}

1. Câu lệnh cho nút Lưu

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

{

if (txtMaSach.Text != "" && txtTenSach.Text != "" && txtTenTG.Text != "" && txtNamXB.Text != "" && txtNXB.Text != "")

{

string maSach = txtMaSach.Text.Trim();

if (busSach.checkMaSach(maSach) == 0)

{

string tenSach = txtTenSach.Text;

string tenTacGia = txtTenTG.Text;

int namXB = int.Parse(txtNamXB.Text);

string nhaXB = txtNXB.Text;

DTO\_Sach sach = new DTO\_Sach(maSach, tenSach, tenTacGia, namXB, nhaXB);

if (busSach.themSach(sach))

{

MessageBox.Show("Thêm sách thành công !");

dgvSach.DataSource = busSach.loadDataSach();

}

else

{

MessageBox.Show("Thêm sách không thành công !");

}

}

else

{

MessageBox.Show("Bị trùng mã sách !");

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ các thông tin");

}

}

1. Câu lệnh cho nút Reset

//Phương thức làm mới Controls

public void lamMoi()

{

txtMaSach.Clear();

txtMaSach.Focus();

txtTenSach.Clear();

txtTenTG.Clear();

txtNamXB.Clear();

txtNXB.Clear();

}

//Nút Reset thực hiện chức năng làm mới

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

{

lamMoi();

dgvSach.DataSource = busSach.loadDataSach();

}

1. Câu lệnh cho nút Home

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

{

frmMain f = new frmMain();

f.Show();

this.Hide();

}

## **3.6.2. DTO**

public class DTO\_Sach

{

private string maSach;

private string tenSach;

private string tenTacGia;

private int namXuatBan;

private string nhaXuatBan;

/\*============ GETTER/SETTER ============\*/

public string MaSach

{

get

{

return maSach;

}

set

{

maSach = value;

}

}

public string TenSach

{

get

{

return tenSach;

}

set

{

tenSach = value;

}

}

public string TenTacGia

{

get

{

return tenTacGia;

}

set

{

tenTacGia = value;

}

}

public int NamXuatBan

{

get

{

return namXuatBan;

}

set

{

namXuatBan = value;

}

}

public string NhaXuatBan

{

get

{

return nhaXuatBan;

}

set

{

nhaXuatBan = value;

}

}

/\*===== CONSTRUCTOR =====\*/

public DTO\_Sach()

{

}

public DTO\_Sach(string ms, string tensach, string tentg, int namxb, string nxb)

{

this.maSach = ms;

this.tenSach = tensach;

this.tenTacGia = tentg;

this.namXuatBan = namxb;

this.nhaXuatBan = nxb;

}

## **3.6.3. BUS**

public class BUS\_Sach

{

DAL\_Sach dalSach = new DAL\_Sach();

// Phương thức trả về bảng dữ liệu chứa thông tin của tất cả các sách trong CSDL public DataTable loadDataSach()

{

return dalSach.loadDataSach();

}

// Phương thức trả về bảng dữ liệu chứa thông tin của sách có mã trùng với mã được truyền vào.

public DataTable timKiemTheoMa(DTO\_Sach ma)

{

return dalSach.timKiemTheoMa(ma);

}

// Phương thức trả về bảng dữ liệu chứa thông tin của sách có tên sách trùng với tên được truyền vào.

public DataTable timKiemTheoTenSach(DTO\_Sach tensach)

{

return dalSach.timKiemTheoTenSach(tensach);

}

// Phương thức thêm thông tin một cuốn sách mới vào CSDL

public bool themSach(DTO\_Sach sach)

{

return dalSach.themSach(sach);

}

// Phương thức cập nhật thông tin của một cuốn sách trong CSDL

public bool suaSach(DTO\_Sach sach)

{

return dalSach.suaSach(sach);

}

// Phương thức xóa thông tin của một cuốn sách trong CSDL

public bool xoaSach(DTO\_Sach sach)

{

return dalSach.xoaSach(sach);

}

//Phương thức kiểm tra mã sách

public int checkMaSach(string ma)

{

return dalSach.checkMaSach(ma);

}

}

## **3.6.4. DAL**

public class DAL\_Sach

{

/// <summary>

/// Lấy toàn bộ sách

/// </summary>

/// <returns></returns>

public DataTable loadDataSach()

{

string sql = "SELECT \* FROM SACH";

return DataProvider.getTable(sql);

}

/// <summary>

/// Tìm kiếm tên theo mã

/// </summary>

/// <param name="ma"></param>

/// <returns></returns>

public DataTable timKiemTheoMa(DTO\_Sach ma)

{

string sql = string.Format("SELECT \* FROM SACH WHERE MASACH LIKE '%{0}%'", ma.MaSach);

return DataProvider.getTable(sql);

}

/// <summary>

/// Tìm kiếm theo tên sách

/// </summary>

/// <param name="tensach"></param>

/// <returns></returns>

public DataTable timKiemTheoTenSach(DTO\_Sach tensach)

{

string sql = string.Format("SELECT \* FROM SACH WHERE TENSACH LIKE N'%{0}%'", tensach.TenSach);

return DataProvider.getTable(sql);

}

/// <summary>

/// Thêm sách

/// </summary>

/// <param name="maSach"></param>

/// <param name="tenSach"></param>

/// <param name="tenTg"></param>

/// <param name="namXB"></param>

/// <param name="nhaXB"></param>

public bool themSach(DTO\_Sach sach)

{

string sql = string.Format("INSERT INTO SACH VALUES (@masach, @tensach, @tentacgia, @namxb, @nxb)");

string[] name = { "@masach", "tensach", "tentacgia", "@namxb", "@nxb" };

object[] value = { sach.MaSach, sach.TenSach, sach.TenTacGia, sach.NamXuatBan, sach.NhaXuatBan};

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Sửa sách

/// </summary>

/// <param name="sach"></param>

/// <returns></returns>

public bool suaSach(DTO\_Sach sach)

{

string sql = string.Format("UPDATE SACH SET TENSACH = @tensach, TENTACGIA = @tentacgia, NAMXB = @namxb, NHAXB = @nxb WHERE MASACH = '{0}'", sach.MaSach);

string[] name = { "@tensach", "@tentacgia", "@namxb", "@nxb" };

object[] value = { sach.TenSach, sach.TenTacGia, sach.NamXuatBan, sach.NhaXuatBan};

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Xóa sách

/// </summary>

/// <param name="sach"></param>

/// <returns></returns>

public bool xoaSach(DTO\_Sach sach)

{

string sql = "DELETE FROM SACH WHERE MASACH = @masach";

string[] name = { "@masach" };

object[] value = { sach.MaSach};

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Kiểm tra mã sách

/// </summary>

/// <param name="ma"></param>

/// <returns></returns>

public int checkMaSach(string ma)

{

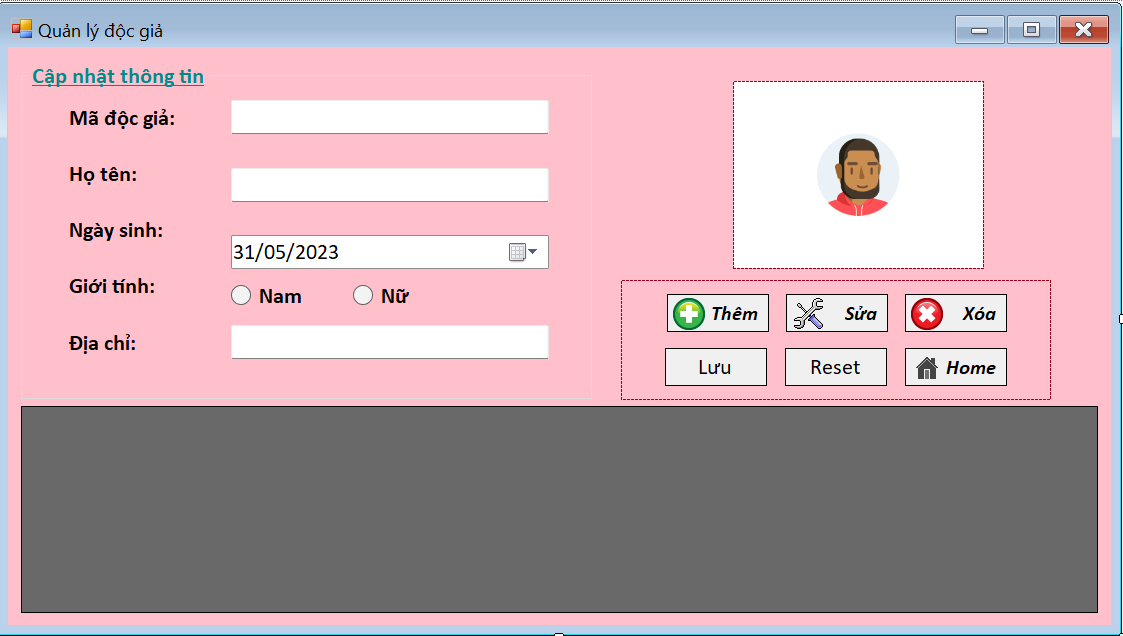
string sql = string.Format("select count(\*) from SACH where MASACH ='{0}'", ma);

return DataProvider.executeScalar(sql);

}

# **3.7. Màn hình Quản lý Độc giả**

## **3.7.1. GUI**



**Hình 3.7.1.** Màn hình quản lý Độc giả

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Mã độc giả | Textbox | Hiển thị mã độc giả |
| Họ tên | Textbox | Hiển thị họ tên độc giả |
| Ngày sinh | Textbox | Hiển thị ngày sinh của độc giả |
| Giới tính | Radio Button | Hiển thị giới tính của độc giả |
| Địa chỉ | Textbox | Hiển thị địa chỉ của độc giả |
| Thêm | Button | Thực hiện chức năng thêm một độc giả mới khi người dùng nhấn vào. |
| Sửa | Button | Thực hiện chức năng sửa độc giả khi người dùng nhấn vào. |
| Xóa | Button | Thực hiện chức năng xóa độc giả khi người dùng nhấn vào. |
| Lưu | Button | Thực hiện chức năng lưu đôc giả khi thêm mới một độc giả. |
| Reset | Button | Làm mới các Textbox có trên form |
| Home | Button | Quay về trang chủ |

1. Câu lệnh cho nút Thêm

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

{

txtMaDG.Enabled = true;

txtMaDG.Text = "";

txtHoTen.Text = "";

dtpNgaySinh.Value = DateTime.Now;

txtDiaChi.Text = "";

rdoNam.Checked = false;

rdoNu.Checked = false;

btnSave.Enabled = true;

btnReset.Enabled = true;

btnXoa.Enabled = false;

btnSua.Text = "Hủy";

btnThem.Enabled = false;

dgvDocGia.Enabled = false;

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

btnSave.Enabled = true;

}

1. Câu lệnh cho nút Sửa

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

{

int i = dgvDocGia.CurrentCell.RowIndex;

if(i >= 0)

{

if(btnSua.Text == "Hủy")

{

txtMaDG.Enabled = false;

btnSave.Enabled = false;

btnReset.Enabled = false;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Text = "Xóa";

btnSua.Text = "Sửa";

btnThem.Enabled = true;

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

dgvDocGia.Enabled = true;

}

else

{

if (txtMaDG.Text != "" && txtHoTen.Text != "" && dtpNgaySinh.Text != "" && rdoNam.Text != "" && txtDiaChi.Text != "")

{

string maDG = txtMaDG.Text;

string hoTen = txtHoTen.Text;

DateTime ngaySinh = Convert.ToDateTime(dtpNgaySinh.Text);

bool gt = rdoNam.Checked == true ? true : false;

string diaChi = txtDiaChi.Text;

//Tạo DTO

DTO\_DocGia docgia = new DTO\_DocGia(maDG, hoTen, ngaySinh, gt, diaChi);

if(ngaySinh >= DateTime.Now)

{

MessageBox.Show("Ngày sinh không hợp lệ");

}

else

{

if (busDocGia.suaDocGia(docgia))

{

MessageBox.Show("Sửa độc giả thành công");

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

}

else

{

MessageBox.Show("Sửa độc giả không thành công");

}

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin");

}

}

}

else

{

MessageBox.Show("Hãy chọn độc giả muốn sửa");

}

}

1. Câu lệnh cho nút Xóa

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

{

if (btnXoa.Text == "Hủy")

{

txtMaDG.Enabled = false;

btnSave.Enabled = false;

btnReset.Enabled = false;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Text = "Xóa";

btnSua.Text = "Sửa";

btnThem.Enabled = true;

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

dgvDocGia.Enabled = true;

}

else

{

if(txtMaDG.Text !="" && txtHoTen.Text !="" && dtpNgaySinh.Text != "" && txtDiaChi.Text != "")

{

string maDG = txtMaDG.Text;

string hoTen = txtHoTen.Text;

DateTime ngaySinh = dtpNgaySinh.Value;

bool gt = rdoNam.Checked == true ? true : false;

string diaChi = txtDiaChi.Text;

DTO\_DocGia docgia = new DTO\_DocGia(maDG, hoTen, ngaySinh, gt, diaChi);

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

if (result == DialogResult.Yes)

{

if (busDocGia.xoaDocGia(docgia))

{

MessageBox.Show("Xóa độc giả thành công");

lamMoi();

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

}

else

{

MessageBox.Show("Xóa độc giả không thành công");

}

}

}

else

{

MessageBox.Show("Vui lòng độc giả cần xóa");

}

}

}

1. Câu lệnh cho nút Lưu

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

{

if (txtMaDG.Text != "" && txtHoTen.Text != "" && dtpNgaySinh.Text != "" && rdoNam.Text != "" && txtDiaChi.Text != "")

{

string maDG = txtMaDG.Text.Trim();

string hoTen = txtHoTen.Text;

DateTime ngaySinh = dtpNgaySinh.Value;

bool gt = rdoNam.Checked == true ? true : false;

string diaChi = txtDiaChi.Text;

//Tạo DTO

DTO\_DocGia docgia = new DTO\_DocGia(maDG, hoTen, ngaySinh, gt, diaChi);

if (ngaySinh >= DateTime.Now)

{

MessageBox.Show("Ngày sinh không hợp lệ");

}

else

{

if (busDocGia.checkMaDocGia(maDG) == 0)

{

if (busDocGia.themDocGia(docgia))

{

MessageBox.Show("Thêm độc giả thành công");

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

}

else

{

MessageBox.Show("Thêm độc giả không thành công");

}

}

else

{

MessageBox.Show("Bị trùng mã độc giả");

}

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin");

}

}

1. Câu lệnh cho Nút Reset

//Phương thức làm mới controls

public void lamMoi()

{

txtMaDG.Clear();

txtHoTen.Clear();

dtpNgaySinh.Value = DateTime.Now;

rdoNam.Checked = false;

rdoNu.Checked = false;

txtDiaChi.Clear();

}

//Nút Reset thực hiện chức năng làm mới

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

{

lamMoi();

dgvDocGia.DataSource = busDocGia.loadDataDocGia();

}

1. Câu lệnh cho nút Home

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

{

frmMain f = new frmMain();

f.Show();

this.Hide();

}

## **3.7.2. DTO**

public class DTO\_DocGia

{

private string maDG;

private string hoTen;

private DateTime ngaySinh;

private bool gioiTinh;

private string diaChi;

/\*============ GETTER/SETTER ============\*/

public string MaDG

{

get

{

return maDG;

}

set

{

maDG = value;

}

}

public string HoTen

{

get

{

return hoTen;

}

set

{

hoTen = value;

}

}

public DateTime NgaySinh

{

get

{

return ngaySinh;

}

set

{

ngaySinh = value;

}

}

public bool GioiTinh

{

get

{

return gioiTinh;

}

set

{

gioiTinh = value;

}

}

public string DiaChi

{

get

{

return diaChi;

}

set

{

diaChi = value;

}

}

/\*===== CONSTRUCTOR =====\*/

public DTO\_DocGia()

{

}

public DTO\_DocGia(string madg, string ht, DateTime ngaysinh, bool gt, string dc)

{

this.maDG = madg;

this.hoTen = ht;

this.ngaySinh = ngaysinh;

this.gioiTinh = gt;

this.diaChi = dc;

}

}

## **3.7.3. BUS**

public class BUS\_DocGia

{

DAL\_DocGia dalDocGia = new DAL\_DocGia();

// Phương thức trả về bảng dữ liệu chứa thông tin của tất cả độc giả trong cơ sở dữ liệu.

public DataTable loadDataDocGia()

{

return dalDocGia.loadDataDocGia();

}

// Phương thức thêm thông tin một độc giả mới vào cơ sở dữ liệu

public bool themDocGia(DTO\_DocGia docgia)

{

return dalDocGia.themDocGia(docgia);

}

// Phương thức cập nhật thông tin độc giả trong cơ sở dữ liệu

public bool suaDocGia(DTO\_DocGia docgia)

{

return dalDocGia.suaDocGia(docgia);

}

// Phương thức xóa thông tin độc giả trong cơ sở dữ liệu

public bool xoaDocGia(DTO\_DocGia docgia)

{

return dalDocGia.xoaDocGia(docgia);

}

// Phương thức kiểm tra mã độc giả

public int checkMaDocGia(string ma)

{

return dalDocGia.checkMaDocGia(ma);

}

}

## **3.7.4. DAL**

--Tạo procedure trong SQL để lấy toàn bộ độc giả

CREATE PROCEDURE SelectAllDocGia

AS

SELECT

MADG, HOTEN, CONVERT (VARCHAR(10), NGAYSINH, 103) AS NGAYSINH, --định dạng lại ngày sinh theo kiểu dd/mm/yyyy

CASE

GIOITINH WHEN 'TRUE' THEN N'Nam'

ELSE N'Nữ'

END AS GIOITINH,

DIACHI

FROM DOCGIA

GO;

EXEC SelectAllDocGia

public class DAL\_DocGia

{

/// <summary>

/// Lấy toàn bộ thông tin độc giả

/// </summary>

/// <returns></returns>

public DataTable loadDataDocGia()

{

string sql = "SelectAllDocGia";

return DataProvider.getTable(sql);

}

/// <summary>

/// Thêm độc giả

/// </summary>

/// <param name="docgia"></param>

/// <returns></returns>

public bool themDocGia(DTO\_DocGia docgia)

{

string sql = "INSERT INTO DOCGIA VALUES (@madg, @hoten, @ngaysinh, @gt, @diachi)";

string[] name = { "@madg", "@hoten", "@ngaysinh", "@gt", "@diachi" };

object[] value = { docgia.MaDG, docgia.HoTen, docgia.NgaySinh, docgia.GioiTinh, docgia.DiaChi};

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Sửa độc giả

/// </summary>

/// <param name="docgia"></param>

/// <returns></returns>

public bool suaDocGia(DTO\_DocGia docgia)

{

string sql =string.Format("UPDATE DOCGIA SET HOTEN = @hoten, NGAYSINH = @ngaysinh, GIOITINH = @gt, DIACHI = @diachi WHERE MADG ='{0}'",docgia.MaDG);

string[] name = { "@hoten", "@ngaysinh", "@gt", "@diachi"};

object[] value = {docgia.HoTen, docgia.NgaySinh, docgia.GioiTinh, docgia.DiaChi};

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Xóa độc giả

/// </summary>

/// <param name="docgia"></param>

/// <returns></returns>

public bool xoaDocGia(DTO\_DocGia docgia)

{

string sql = "DELETE FROM DOCGIA WHERE MADG = @madg";

string[] name = { "@madg"};

object[] value = { docgia.MaDG};

return DataProvider.updateData(sql, name, value);

}

//Kiểm tra mã độc giả

public int checkMaDocGia(string ma)

{

string sql = string.Format("SELECT COUNT(\*) FROM DOCGIA WHERE MADG ='{0}'", ma);

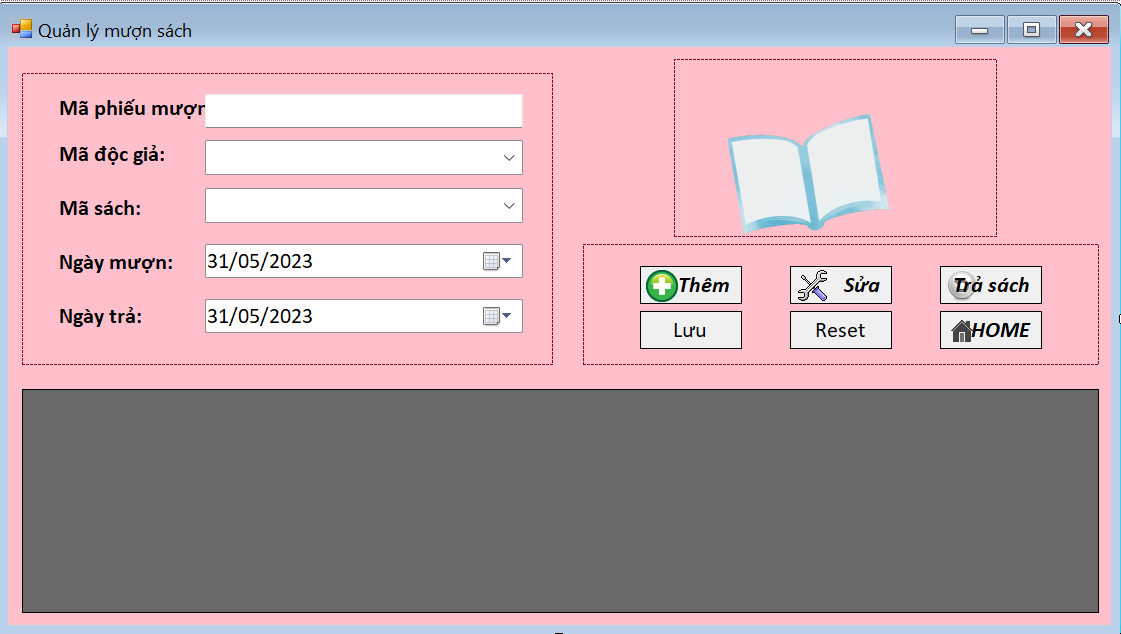
return DataProvider.executeScalar(sql);

}

}

# **3.8. Màn hình Quản lý Mượn sách**

## **3.8.1. GUI**



**Hình 3.8.1.** Màn hình quản lý Mượn sách

1. Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu** | **Mô tả** |
| Mã phiếu mượn | TextBox | Hiển thị thông tin mã phiếu mượn |
| Mã độc giả | Combobox | Hiển thị thông tin mã độc giả |
| Mã sách | Combobox | Hiển thị thông tin mã sách |
| Ngày mượn | Datetime Picker | Hiển thị ngày mượn sách |
| Ngày trả | Datetime Picker | Hiển thị thông tin ngày trả sách |
| Thêm | Button | Thực hiện chức năng thêm một thông tin mượn sách mới khi người dùng nhấn vào. |
| Sửa | Button | Thực hiện chức năng sửa thông tin mượn sách khi người dùng nhấn vào. |
| Trả sách | Button | Thực hiện chức năng xóa thông tin mượn sách khi người dùng nhấn vào. |
| Lưu | Button | Thực hiện chức năng lưu thông tin sách khi thêm sách mới một cuốn sách. |
| Reset | Button | Làm mới các Textbox có trên form |
| Home | Button | Quay về trang chủ |

1. Câu lệnh cho nút Thêm

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

{

txtMaPhieuMuon.Enabled = true;

cboMaDG.Text = "";

cboMaSach.Text = "";

dtpNgayMuon.Value = DateTime.Now;

dtpNgayTra.Value = DateTime.Now;

btnSave.Enabled = true;

btnReset.Enabled = true;

btnXoa.Enabled = false;

btnSua.Text = "Hủy";

btnThem.Enabled = false;

dgvMuonSach.Enabled = false;

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

btnSave.Enabled = true;

}

1. Câu lệnh cho nút Sửa

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

{

int i = dgvMuonSach.CurrentCell.RowIndex;

if(i >= 0)

{

if (btnSua.Text == "Hủy")

{

txtMaPhieuMuon.Enabled = false;

btnSave.Enabled = false;

btnReset.Enabled = false;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Text = "Xóa";

btnSua.Text = "Sửa";

btnThem.Enabled = true;

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

dgvMuonSach.Enabled = true;

}

else

{

if (txtMaPhieuMuon.Text != "" && cboMaDG.Text != "" && cboMaSach.Text != "" && dtpNgayMuon.Text != "" && dtpNgayTra.Text != "")

{

string maPhieuMuon = txtMaPhieuMuon.Text;

string maDG = cboMaDG.Text.Trim();

string maSach = cboMaSach.Text.Trim();

DateTime ngayMuon = dtpNgayMuon.Value;

DateTime ngayTra = dtpNgayTra.Value;

//Tạo DTO

DTO\_MuonSach muonsach = new DTO\_MuonSach(maPhieuMuon, maDG, maSach, ngayMuon, ngayTra);

//Sửa

if (ngayTra > ngayMuon)

{

if (busMuonSach.suaMuonSach(muonsach))

{

MessageBox.Show("Sửa thành công");

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

}

else

{

MessageBox.Show("Sửa không thành công");

}

}

else

{

MessageBox.Show("Ngày trả phải lớn hơn ngày mượn");

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin");

}

}

}

else

{

MessageBox.Show("Hãy chọn phiếu mượn muốn sửa");

}

}

1. Câu lệnh cho nút Trả sách

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

{

if (btnXoa.Text == "Hủy")

{

txtMaPhieuMuon.Enabled = false;

btnSave.Enabled = false;

btnReset.Enabled = false;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Text = "Xóa";

btnSua.Text = "Sửa";

btnThem.Enabled = true;

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

dgvMuonSach.Enabled = true;

}

else

{

if (txtMaPhieuMuon.Text != "" && cboMaDG.Text != "" && cboMaSach.Text != "" && dtpNgayMuon.Text != "" && dtpNgayTra.Text != "")

{

string maPhieuMuon = txtMaPhieuMuon.Text;

string maDG = cboMaDG.Text.Trim();

string maSach = cboMaSach.Text.Trim();

DateTime ngayMuon = dtpNgayMuon.Value;

DateTime ngayTra = dtpNgayTra.Value;

//Tạo DTO

DTO\_MuonSach muonsach = new DTO\_MuonSach(maPhieuMuon, maDG, maSach, ngayMuon, ngayTra);

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

if (result == DialogResult.Yes)

{

if (busMuonSach.xoaMuonSach(muonsach))

{

MessageBox.Show("Xóa thông tin mượn sách thành công");

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

lamMoi();

}

else

{

MessageBox.Show("Xóa thất bại !");

}

}

}

else

{

MessageBox.Show("Hãy chọn phiếu mượn cần xóa.");

}

}

}

1. Câu lệnh cho nút Lưu

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

{

if (txtMaPhieuMuon.Text != "" && cboMaDG.Text != "" && cboMaSach.Text != "" && dtpNgayMuon.Text != "" && dtpNgayTra.Text != "")

{

string maPhieuMuon = txtMaPhieuMuon.Text.Trim();

string maDG = cboMaDG.Text.Trim();

string maSach = cboMaSach.Text.Trim();

DateTime ngayMuon = Convert.ToDateTime(dtpNgayMuon.Value);

DateTime ngayTra = dtpNgayTra.Value;

//Tạo DTO

DTO\_MuonSach muonsach = new DTO\_MuonSach(maPhieuMuon, maDG, maSach, ngayMuon, ngayTra);

if(busMuonSach.checkMaDocGia(maDG) != -1)

{

if (busMuonSach.themMuonSach(muonsach))

{

MessageBox.Show("Thêm thành công");

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

}

else

{

MessageBox.Show("Thêm không thành công");

}

}

else

{

MessageBox.Show("Độc giả đã mượn 3 cuốn sách, không thể mượn thêm");

}

}

else

{

MessageBox.Show("Ngày trả phải lớn hơn ngày mượn");

}

}

else

{

MessageBox.Show("Mã phiếu mượn bị trùng");

}

}

else

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin");

}

}

1. Câu lệnh cho nút Reset

//Reset dữ liệu trên control

public void lamMoi()

{

txtMaPhieuMuon.Clear();

cboMaDG.SelectedIndex = 0;

cboMaSach.SelectedIndex = 0;

dtpNgayMuon.Value = DateTime.Now;

dtpNgayTra.Value = DateTime.Now;

}

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

{

lamMoi();

dgvMuonSach.DataSource = busMuonSach.loadDataMuonSach();

}

1. Câu lệnh cho nút Home

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

{

frmMain f = new frmMain();

f.Show();

this.Hide();

}

## **3.8.2. DTO**

public class DTO\_MuonSach

{

private string maPhieuMuon;

private string maDG;

private string maSach;

private DateTime ngayMuon;

private DateTime ngayTra;

/\*============ GETTER/SETTER ============\*/

public string MaPhieuMuon

{

get

{

return maPhieuMuon;

}

set

{

maPhieuMuon = value;

}

}

public string MaDG

{

get

{

return maDG;

}

set

{

maDG = value;

}

}

public string MaSach

{

get

{

return maSach;

}

set

{

maSach = value;

}

}

public DateTime NgayMuon

{

get

{

return ngayMuon;

}

set

{

ngayMuon = value;

}

}

public DateTime NgayTra

{

get

{

return ngayTra;

}

set

{

ngayTra = value;

}

}

/\*===== CONSTRUCTOR =====\*/

public DTO\_MuonSach()

{

}

public DTO\_MuonSach(string maphieumuon, string madg, string masach, DateTime ngaymuon, DateTime ngaytra)

{

this.maPhieuMuon = maphieumuon;

this.maDG = madg;

this.maSach = masach;

this.ngayMuon = ngaymuon;

this.ngayTra = ngaytra;

}

}

## **3.8.3. BUS**

public class BUS\_MuonSach

{

DAL\_MuonSach dalMuonSach = new DAL\_MuonSach();

// Phương thức trả về bảng dữ liệu chứa tất cả thông tin mượn sách trong cơ sở dữ liệu.

public DataTable loadDataMuonSach()

{

return dalMuonSach.loadDataMuonSach();

}

//Phương thức load dữ liệu mã độc giả từ cơ sở dữ liệu và trả về dưới dạng DataTable.

public DataTable loadDataMaDG()

{

return dalMuonSach.loadMaDG();

}

//Phương thức load dữ liệu mã sách từ cơ sở dữ liệu và trả về dưới dạng DataTable.

public DataTable loadDataMaSach()

{

return dalMuonSach.loadMaSach();

}

// Phương thức thêm một thông tin mượn sách mới vào cơ sở dữ liệu

public bool themMuonSach(DTO\_MuonSach muonsach)

{

return dalMuonSach.themMuonSach(muonsach);

}

// Phương thức cập nhật thông tin một cuốn sách trong cơ sở dữ liệu

public bool suaMuonSach(DTO\_MuonSach muonsach)

{

return dalMuonSach.suaMuonSach(muonsach);

}

// Phương thức xóa thông tin một cuốn sách trong cơ sở dữ liệu

public bool xoaMuonSach(DTO\_MuonSach muonsach)

{

return dalMuonSach.xoaMuonSach(muonsach);

}

// Phương thức kiểm tra mã phiếu mượn

public int checkMuonSach(string maphieumuon)

{

return dalMuonSach.checkMuonSach(maphieumuon);

}

//Kiểm tra mã độc giả đã mượn sách trong bảng MUONSACH

public int checkMaDocGia(string madocgia)

{

int count = dalMuonSach.checkMaDocGia(madocgia);

if(count >= 3)

{

return -1;

}

else

{

return count;

}

}

}

## **3.8.4. DAL**

--Tạo procedure để lấy toàn bộ thông tin mượn sách

CREATE PROCEDURE SellectAllMuonSach

AS

SELECT

MAPHIEUMUON, MADG, MASACH,

CONVERT (VARCHAR(10), NGAYMUON, 103) AS NGAYMUON,

CONVERT (VARCHAR(10), NGAYTRA, 103) AS NGAYTRA

FROM MUONSACH

GO;

EXEC SellectAllMuonSach

public class DAL\_MuonSach

{

/// <summary>

/// Lấy toàn bộ thông tin mượn sách

/// </summary>

/// <returns></returns>

public DataTable loadDataMuonSach()

{

string sql = "SellectAllMuonSach";

return DataProvider.getTable(sql);

}

//Lấy mã độc giả

public DataTable loadMaDG()

{

string sql = "SELECT \* FROM DOCGIA";

return DataProvider.getTable(sql);

}

//Lấy mã sách

public DataTable loadMaSach()

{

string sql = "SELECT \* FROM SACH";

return DataProvider.getTable(sql);

}

/// <summary>

/// Thêm thông tin mượn sách

/// </summary>

/// <param name="muonsach"></param>

/// <returns></returns>

public bool themMuonSach(DTO\_MuonSach muonsach)

{

string sql = "INSERT INTO MUONSACH VALUES (@maphieumuon, @madocgia, @masach, @ngaymuon, @ngaytra)";

string[] name = { "@maphieumuon", "@madocgia", "@masach", "@ngaymuon", "@ngaytra" };

object[] value = { muonsach.MaPhieuMuon, muonsach.MaDG, muonsach.MaSach, muonsach.NgayMuon, muonsach.NgayTra };

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Sửa thông tin mượn sách

/// </summary>

/// <param name="muonsach"></param>

/// <returns></returns>

public bool suaMuonSach(DTO\_MuonSach muonsach)

{

string sql = string.Format("UPDATE MUONSACH SET MASACH = @masach, NGAYMUON = @ngaymuon, NGAYTRA = @ngaytra WHERE MAPHIEUMUON = '{0}'", muonsach.MaPhieuMuon);

string[] name = { "@masach", "@ngaymuon", "@ngaytra" };

object[] value = { muonsach.MaSach, muonsach.NgayMuon, muonsach.NgayTra};

return DataProvider.updateData(sql, name, value);

}

/// <summary>

/// Xóa thông tin mượn sách

/// </summary>

/// <param name="muonsach"></param>

/// <returns></returns>

public bool xoaMuonSach(DTO\_MuonSach muonsach)

{

string sql = "DELETE FROM MUONSACH WHERE MAPHIEUMUON = @maphieumuon";

string[] name = { "@maphieumuon" };

object[] value = { muonsach.MaPhieuMuon };

return DataProvider.updateData(sql, name, value);

}

//Kiểm tra mã phiếu

public int checkMuonSach(string maphieumuon)

{

string sql =string.Format("SELECT COUNT(\*) FROM MUONSACH WHERE MAPHIEUMUON ='{0}'", maphieumuon);

return DataProvider.executeScalar(sql);

}

}

//Kiểm tra mã độc giả đã mượn sách trong bảng MUONSACH

public int checkMaDocGia(string madocgia)

{

string sql = string.Format("SELECT COUNT(\*) FROM MUONSACH WHERE MADG ='{0}'", madocgia);

return DataProvider.executeScalar(sql);

}

# **CHƯƠNG 4. KẾT LUẬN**

Trong quá trình thực hiện cũng như sau khi hoàn thành bài tập lớn. Những nội dung mà em đã tìm hiểu, những kiến thức mà bản thân em đạt được là:

* Về nội dung tìm hiểu:

1. Tìm hiểu mô hình, database của một hệ thống quản lý thư viện, các chức năng của một ứng dụng quản lý thư viện.
2. Kết nối cơ sở dữ liệu với ADO.NET.
3. Tìm hiểu mô hình 3 lớp ADO.NET.
4. Giao diện người dùng với Windows Forms.
5. Tìm hiểu về cách sử dụng Word cơ bản.

* Về kiến thức đạt được:

1. Hiểu về mô hình 3 lớp và cách triển khai ADO.NET trong ứng dụng Windows Forms.
2. Biết cách kết nối và thao tác với cơ sở dữ liệu sử dụng ADO.NET gồm các thao tác: truy vấn cơ sở dữ liệu, thêm, sửa, xóa.
3. Xây dựng được giao diện đồ họa cơ bản cho ứng dụng, có thêm kiến thức về các controls, các phương thức trong Windown Forms.
4. Thực thi được các thao tác cơ bản của ứng dụng Quản lý thư viện như: thêm, sửa, xóa, tìm kiếm.
5. Biết sử dụng Word cơ bản.