

VIETNAM NATIONAL UNIVERSITY HOCHIMINH CITY
UNIVERSITY OF SCIENCE
FACULTY OF ELECTRONICS AND TELECOMMUNICATION
DEPARTMENT OF TELECOMMUNICATION AND NETWORK



fetel@HCMUS
KHOA ĐIỆN TỬ - VIỄN THÔNG

ĐỒ ÁN CUỐI KÌ
ĐỀ TÀI QUẢN LÝ THƯ VIỆN
THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Supervisor: Cao Tran Bao Thuong, MS.c

Group Member:

Huynh Thi Ngoc Phuc	21207195
Pham Hoai An	21207120
Tran Thien Phuc	21207077

Ho Chi Minh City, Apr 2024

MỤC LỤC

LỜI NÓI ĐẦU	3
1. TỔNG QUAN	4
1.1. Giới thiệu về vấn đề	4
1.2. Mục đích và yêu cầu	4
1.3. Quy định thực hiện	5
2. XÁC ĐỊNH VÀ MÔ HÌNH HÓA YÊU CẦU	6
3. XÂY DỰNG ỨNG DỤNG	7
3.1. Xây dựng các đối tượng	7
3.2. Xây dựng lớp quản lý (nhân viên thư viện, thủ thư và cộng tác viên)	7
3.3. Xây dựng chương trình quản lý	11
4. TESTCASE	37
4.1. Caseform “Quản Lý Sách”	37
4.2. Caseform “Quản Lý Độc Giả”	40
4.3. Caseform “Quản Lý Thẻ”	43
5. GITHUB	49
6. ĐÁNH GIÁ – NHIỆM VỤ	50

LỜI NÓI ĐẦU

Những năm gần đây, vai trò của công nghệ thông tin trong đời sống con người đã ngày càng trở nên quan trọng. Thực tế đã cho thấy, việc ứng dụng thành tựu công nghệ thông tin vào các công đoạn như sản xuất, quản lý... đã tạo nên một lợi ích to lớn, dẫn đến cuộc cách mạng số hóa bùng nổ trong thời đại hiện nay. Tuy nhiên, việc ứng dụng công nghệ không phải luôn dễ dàng, đặc biệt là với những nước đang phát triển và thiếu hụt nguồn nhân lực chất lượng trong lĩnh vực công nghệ thông tin như Việt Nam. Điều đó đặt ra yêu cầu các sản phẩm công nghệ phải dễ sử dụng, thân thiện với những người không có kiến thức công nghệ thông tin và vẫn đảm bảo tính hiệu quả, đúng đắn của nó.

Sau khi được học tập và rèn luyện dưới sự hướng dẫn của thầy Cao Trần Bảo Thương tại khoa Điện tử - Viễn Thông, trường Đại học Khoa Học Tự Nhiên – ĐHQGHCM, nhóm sinh viên chúng tôi đã được trang bị những kiến thức cơ bản để có thể phát triển một phần mềm ứng dụng thỏa mãn yêu cầu trên. Vì vậy, chúng tôi lựa chọn thực hiện đề án môn học “Phần mềm Quản lý Thư viện” để có thêm kinh nghiệm và cái nhìn sâu hơn về nhu cầu của người sử dụng phổ thông cũng như về các giai đoạn phát triển một phần mềm đáp ứng nhu cầu đó.

Chúng em xin chân thành cảm ơn thầy Cao Trần Bảo Thương đã tận tình truyền dạy những kiến thức cần thiết và hướng dẫn chúng em hoàn thành đề án này.

Trong quá trình thực hiện đề án, không thể tránh khỏi những sai sót, chúng em mong được nhận sự phản hồi góp ý của thầy và mọi người để chúng em rút kinh nghiệm và hoàn thiện hơn.

Trân Trọng Cảm ơn Thầy!

1. TỔNG QUAN

1.1. Giới thiệu về vấn đề

Đứng trước xã hội đang phát triển với tốc độ chóng mặt ngày nay, cộng với đặc thù ngành học bắt buộc phải liên tục trau dồi kiến thức về công nghệ và kỹ thuật mới, nhu cầu đọc sách của sinh viên ngày càng tăng cao, đòi hỏi lượng sách nhập về phải được đảm bảo và các đầu sách mới phải thường xuyên được cập nhật. Từ đó, khối lượng công việc của thư viện trường tăng nhiều hơn, nảy sinh yêu cầu tạo ra một phần mềm *Quản lý thư viện* đáp ứng đầy đủ các nhu cầu của người quản lý cũng như của sinh viên.

1.2. Mục đích và yêu cầu

1.2.1. Mục đích

- Giảm thiểu khối lượng công việc thủ công, tiết kiệm thời gian
- Tăng năng suất trong việc nhập liệu, lập báo cáo
- Tăng tính bảo mật và hiệu quả trong công tác quản lý

1.2.2. Đối tượng sử dụng

Phần mềm này dành cho các cán bộ quản lý, thủ thư và cộng tác viên – những người chịu trách nhiệm quản lý thông tin sách và độc giả của thư viện trường.

1.2.3. Yêu cầu

- Giao diện của chương trình được thiết kế hợp lý, rõ ràng và thuận tiện cho người sử dụng.
- Tốc độ xử lý của chương trình tốt, đáp ứng được lượng công việc lớn trong một khoảng thời gian chấp nhận được.
- Các nghiệp vụ phải được chương trình ghi nhận và xử lý đúng đắn.
- Người sử dụng có thể điều chỉnh, cập nhật thông tin và quy định.

1.3. Quy định thực hiện

Trong khuôn khổ đề án môn học này, chúng tôi chia quy trình phát triển phần mềm Quản lý thư viện thành các giai đoạn chính như sau:

- **Giai đoạn 1:** Xác định và mô hình hóa yêu cầu phần mềm.
- **Giai đoạn 2:** Phân tích và thiết kế.
 - **Giai đoạn 2.1:** Phân tích, phác thảo các thành phần của phần mềm.
 - **Giai đoạn 2.2:** Thiết kế hệ thống.
 - **Giai đoạn 2.3:** Thiết kế cơ sở dữ liệu.
 - **Giai đoạn 2.4:** Thiết kế giao diện.
- **Giai đoạn 3:** Cài đặt.
- **Giai đoạn 4:** Kiểm thử và sửa lỗi.

2. XÁC ĐỊNH VÀ MÔ HÌNH HÓA YÊU CẦU

Các yêu cầu cần thiết cho chương trình .

STT	Công việc	Loại công việc	Ghi chú
1	Lập thẻ độc giả	Lưu trữ	
2	Xóa thẻ	Lưu trữ	
3	Tra cứu độc giả	Tra cứu	
4	Xóa độc giả	Lưu trữ	
5	Tiếp nhận sách mới	Lưu trữ	
6	Tra cứu sách	Tra cứu	
7	Xóa sách	Lưu trữ	
8	Cho mượn sách	Lưu trữ	
9	Nhận trả sách	Lưu trữ	

3. XÂY DỰNG ỨNG DỤNG

3.1. Xây dựng các đối tượng

```
public class Book
{
    public string BookCode { get; set; }
    public string Title { get; set; }
    public string Author { get; set; }
    public string Category { get; set; }
    public int Quantity { get; set; }
}
```

Lớp sách được xây dựng với các thuộc tính: Mã sách, tên sách, tác giả, phân loại và số lượng.

```
public class Person
{
    public string ID { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public long Fee { get; set; }
}
```

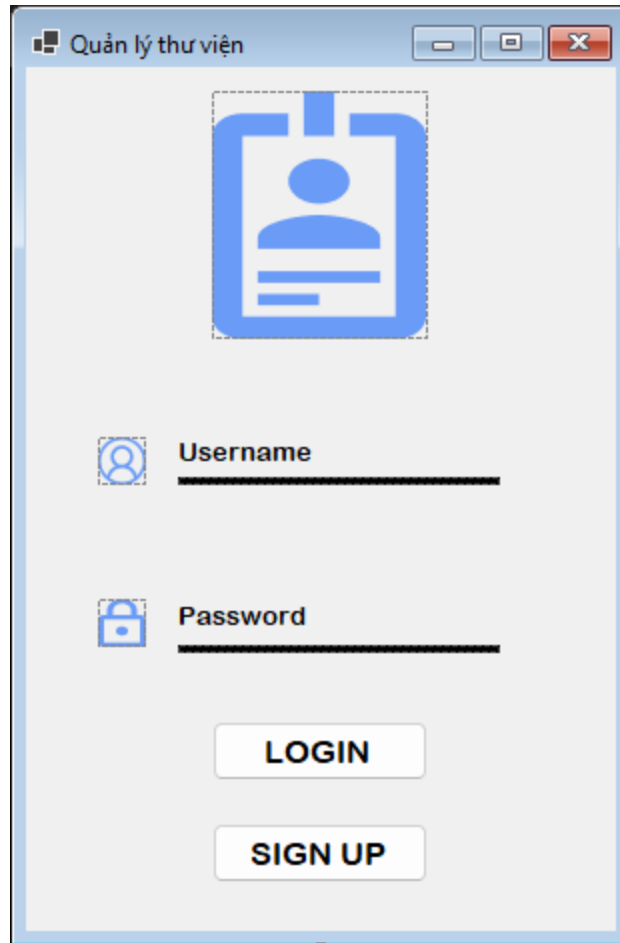
Lớp độc giả được xây dựng với các thuộc tính: Độc giả, tên độc giả, loại độc giả và phí.

```
public class Card
{
    public string ReaderID { get; set; }
    public string CardID { get; set; }
    public string CardExp { get; set; }
    public string BookBorrow { get; set; } // Change from string[] to string
}
```

Lớp thẻ thư viện được xây dựng với các thuộc tính: Mã độc giả, mã thẻ, hạn thẻ và sách đã mượn.

3.2. Xây dựng lớp quản lý (nhân viên thư viện, thủ thư và cộng tác viên)

- Lớp quản lý đăng nhập được xây dựng dựa trên lớp LoginForm của ứng dụng
- Giao diện của form đăng nhập vào ứng dụng:



- Xây dựng form đăng nhập

```
using TestWinform.Properties;
namespace TestWinform
{
    public partial class LoginForm : Form
    {
        private List<Administrator> administrators;

        public LoginForm()
        {
            InitializeComponent();
            // Khởi tạo danh sách Quản trị viên
            administrators = new List<Administrator>();

            // Thêm các mẫu Quản trị viên
            administrators.Add(new Administrator("admin", "admin123"));
            administrators.Add(new Administrator("HoaiAn2912",
"PhamHoaiAn29"));
        }

        private void txtUsername_MouseClick(object sender, MouseEventArgs e)
        {

```



```

        if (txtUsername.Text == "Username")
        {
            txtUsername.Clear();
        }
    }

private void txtPassword_MouseClick(object sender, MouseEventArgs e)
{
    if (txtPassword.Text == "Password")
    {
        txtPassword.Clear();
        txtPassword.PasswordChar = '*';
    }
}

private void button1_MouseClick(object sender, MouseEventArgs e)
{
    string username = txtUsername.Text;
    string password = txtPassword.Text;

    bool loginSuccessful = false;

    // Lặp qua danh sách các quản trị viên và kiểm tra đăng nhập
    foreach (var admin in administrators)
    {
        if (admin.Username == username && admin.Password ==
password)
        {
            loginSuccessful = true;
            break; // Thoát khỏi vòng lặp khi tìm thấy đúng người
dùng
        }
    }

    if (loginSuccessful)
    {
        MessageBox.Show("Login successful!");
        // Thực hiện các hành động sau khi đăng nhập thành công
        Form1 Mn = new Form1();
        Mn.Show();
        Visible = false;
    }
    else
    {
        MessageBox.Show("Invalid username or password!");
    }
}

private void button2_MouseClick(object sender, MouseEventArgs e)
{
    string username = txtUsername.Text;
    string password = txtPassword.Text;
    administrators.Add(new Administrator(username, password));
    MessageBox.Show("Registration successful!");
}

```

```

public class Administrator
{
    public string Username { get; set; }
    public string Password { get; set; }

    public Administrator(string username, string password)
    {
        Username = username;
        Password = password;
    }

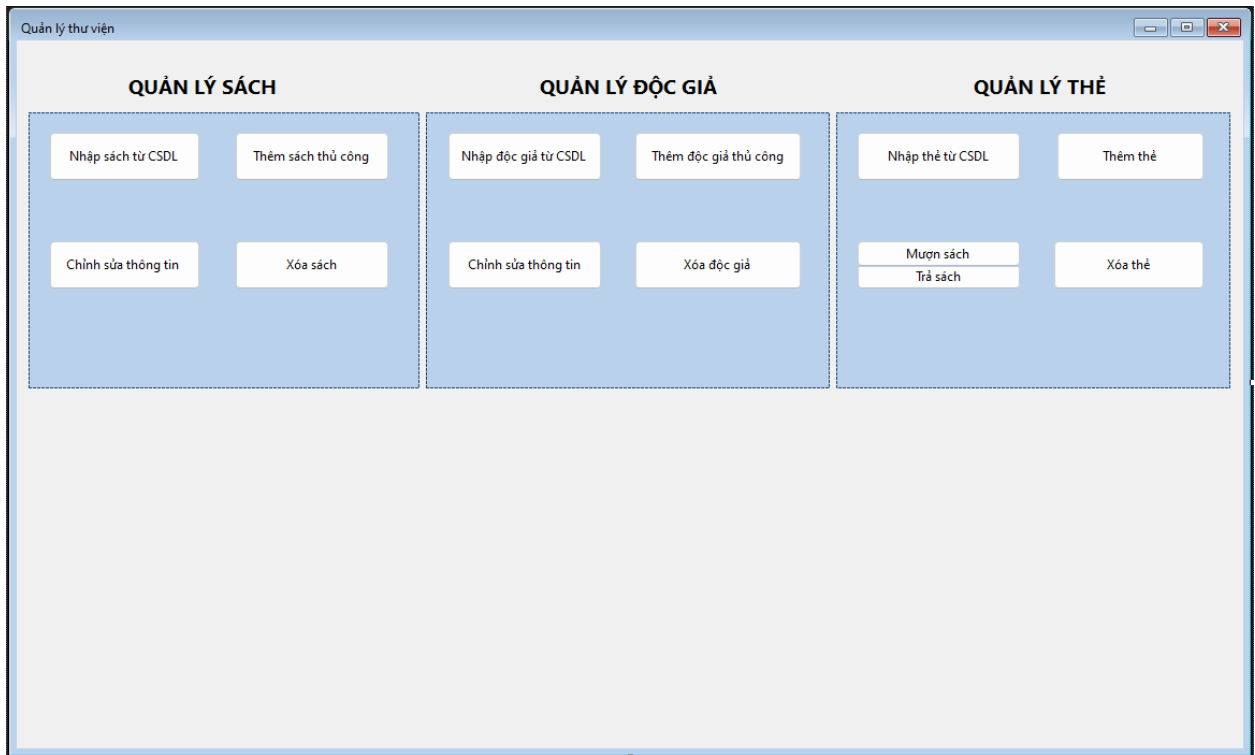
    private void button1_Click_2(object sender, EventArgs e)
    {

    }
}

```

- Bằng ngôn ngữ C# và .net sử dụng Winform chúng em tạo form với các textbox và button có các sự kiện để kích hoạt xử lý các nhiệm vụ đã được lập trình sẵn.
- Phương thức hoạt động
- Một List<Administrator> với phạm vi truy cập Private được tạo nên dùng để lưu trữ các thuộc tính của quản trị viên: Tên đăng nhập và mật khẩu
- Khi đăng nhập, lần lượt nhập tên đăng nhập và mật khẩu vào các textbox sau đó nhấn chọn button Login.
- Sau khi đăng nhập thì chương trình sẽ được đưa đến một form được dùng để quản lý chung cho tất cả các đối tượng: Sách, độc giả và thẻ thư viện

3.3. Xây dựng chương trình quản lý



Chương trình quản lý chung trên được xây dựng để quản lý cả 3 thành phần chính trong quản lý thư viện đó là: Quản lý sách, quản lý độc giả và quản lý thẻ.

Quản lý sách:

- **Nhập sách từ cơ sở dữ liệu:** Cho phép nhập các thông tin của sách từ cơ sở dữ liệu được thiết lập sẵn (ở đây nhóm chúng em sử dụng file .csv để làm cơ sở dữ liệu cho đối tượng sách). Dưới đây là các trường dữ liệu của cơ sở dữ liệu về sách: Mã sách, tên sách, tác giả, phân loại và số lượng.

B001	Book1	Author1	1	299
B002	Book2	Author2	4	997
B003	Book3	Author3	3	720
B005	Book5	Author5	4	682
B006	Book6	Author6	5	923
B007	Book7	Author7	1	753
B008	Book8	Author8	2	518
B009	Book9	Author9	2	707
B010	Book10	Author10	5	356
B004	Book4	Author4	2	1000

- Dưới đây là đoạn chương trình dùng để đọc dữ liệu của đối tượng sách từ cơ sở dữ liệu đã có sẵn:

```
private void button1_Click(object sender, EventArgs e)
{
    // Read books from CSV
    List<Book> csvBooks = ReadBooksFromCSV("Book_Data.csv");

    // Display CSV books in DataGridView
    dataGridView1.DataSource = csvBooks;
}

private List<Book> ReadBooksFromCSV(string filePath)
{
    List<Book> books = new List<Book>();

    try
    {
        using (var reader = new StreamReader(filePath))
        {
            // Read each line in the file
            string line;
            while ((line = reader.ReadLine()) != null)
            {
                // Split the line into parts
                string[] parts = line.Split(',');

                // Check if all fields are present
                if (parts.Length >= 5)
                {
                    // Create a Book object from the data
                    Book book = new Book
                    {
                        BookCode = parts[0],
                        Title = parts[1],
                        Author = parts[2],
                        Category = parts[3],
                        Quantity = int.Parse(parts[4],
CultureInfo.InvariantCulture)
                    };
                    books.Add(book);
                }
            }
        }
    }
}
```

```

        else
        {
            // Handle invalid line
            Console.WriteLine("Invalid line: " + line);
        }
    }

    //MessageBox.Show("Successfully read file: " + filePath);
}
catch (Exception ex)
{
    MessageBox.Show("Error reading CSV file: " + ex.Message);
}

return books;
}

```

- Lần lượt dữ liệu sẽ được đọc từ file “Book_Data.csv” với các trường dữ liệu đã nêu tương ứng với các parts[0] đến parts[4] trong chương trình.
 - Với các trường có dữ liệu chưa phù hợp với yêu cầu thì sẽ được chuyển về kiểu dữ liệu phù hợp với nó.
 - Thêm sách thủ công: Sách sẽ được thêm vào và cập nhật xuống cơ sở dữ liệu.
- Giao diện thêm sách được thiết lập như sau:

- Trong chức năng thêm sách vẫn đảm bảo tối đa các yêu cầu về dữ liệu cần có của một đầu sách.

- Dữ liệu được nhập vào các textbox và thực thi sau khi nhấn vào button
- Dưới đây là chương trình thực hiện nhập thông tin của sách và thực hiện cập nhật ở:

- **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;

namespace TestWinform.Properties
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        public Book GetBook()
        {
            // Retrieve data from Form2
            string bookCode = textBox1.Text;
            string title = textBox2.Text;
            string author = textBox3.Text;
            string category = textBox4.Text;
            int quantity;
            if (!int.TryParse(textBox5.Text, out quantity))
            {
                MessageBox.Show("Invalid quantity.");
                return null;
            }

            // Create a new Book object
            return new Book
            {
                BookCode = bookCode,
                Title = title,
                Author = author,
                Category = category,
                Quantity = quantity
            };
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            // Retrieve book data from Form2
            Book newBook = GetBook();
        }
    }
}
```

```

        // Check if book data is valid
        if (newBook != null)
        {
            // Create a message to display book details
            string message = $"Book added successfully:\n\n" +
                $"Book Code: {newBook.BookCode}\n" +
                $"Title: {newBook.Title}\n" +
                $"Author: {newBook.Author}\n" +
                $"Category: {newBook.Category}\n" +
                $"Quantity: {newBook.Quantity}";

            // Show MessageBox with book details
            MessageBox.Show(message, "Success", MessageBoxButtons.OK,
                MessageBoxIcon.Information);

            // Confirm the addition and close Form2
            DialogResult = DialogResult.OK;
            Close();
        }
    }

    private void textBox1_MouseClick(object sender, MouseEventArgs e)
    {
        if (textBox1.Text == "Mã sách")
        {
            textBox1.Text = null;
        }
    }

    private void textBox2_MouseClick(object sender, MouseEventArgs e)
    {
        if (textBox2.Text == "Tên sách")
        {
            textBox2.Text = null;
        }
    }

    private void textBox3_MouseClick(object sender, MouseEventArgs e)
    {
        if (textBox3.Text == "Tác giả")
        {
            textBox3.Text = null;
        }
    }

    private void textBox4_MouseClick(object sender, MouseEventArgs e)
    {
        if (textBox4.Text == "Loại sách")
        {
            textBox4.Text = null;
        }
    }

    private void textBox5_MouseClick(object sender, MouseEventArgs e)
    {
        if (textBox5.Text == "Số lượng")
        {
            textBox5.Text = null;
        }
    }

```

```

    }
}
}

```

- **Form quản lý chính:** Thực hiện gọi Form “Thêm sách” ngay khi sự kiện click chuột vào button “Thêm sách thủ công” được kích hoạt:

```

private void Book_AddNormal_Click(object sender, EventArgs e)
{
    // Show Form2 to add a new book
    using (Form2 form2 = new Form2())
    {
        if (form2.ShowDialog() == DialogResult.OK)
        {
            // Retrieve book data from Form2
            Book newBook = form2.GetBook();

            // Ensure newBook is not null
            if (newBook != null)
            {
                try
                {
                    // Append new book data to the CSV file
                    AppendBookToCSV(newBook, "Book_Data.csv");

                    // Display the updated list of books in DataGridView
                    List<Book> csvBooks = ReadBooksFromCSV("Book_Data.csv");
                    dataGridView1.DataSource = csvBooks;
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error adding book to CSV file:
{ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
    }
}

```

- Cập nhật về cơ sở dữ liệu: Dữ liệu sau khi được nhập thủ công phải được lưu lại cơ sở dữ liệu để đảm bảo tính cập nhật dữ liệu. Đoạn chương trình sau thực hiện thêm dữ liệu vào cơ sở dữ liệu sẵn có:

```

private void AppendBookToCSV(Book book, string filePath)
{
    try
    {
        // Open the CSV file in append mode
        using (var writer = new StreamWriter(filePath, true))
        {
            // Write the new book data to the end of the file
            writer.WriteLine($"{book.BookCode},{book.Title},{book.Author},{book.Category},{book.
Quantity}");
        }
    }
}

```



```

        }

        MessageBox.Show("Successfully added book to CSV file.", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        throw new Exception($"Error appending book to CSV file:
{ex.Message}");
    }
}

//=====Cập nhật giá trị về cơ sở dữ
liệu=====//
private void SaveBooksToCSV(List<Book> books)
{
    try
    {
        // Specify the path of the CSV file
        string filePath = "Book_Data.csv";

        // Create or overwrite the CSV file
        using (var writer = new StreamWriter(filePath))
        {
            // Write each book as a line in the CSV file
            foreach (Book book in books)
            {
                writer.WriteLine($"{book.BookCode},{book.Title},{book.Author},{book.Category},{book.
                Quantity}");
            }
        }

        MessageBox.Show("Successfully saved data to CSV file.", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error saving data to CSV file: {ex.Message}",
        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

- **Chỉnh sửa sách:** Các thông tin của sách được chỉnh sửa trực tiếp trên DataGridView được hiển thị trên Form quản lý chính và cập nhật về cơ sở dữ liệu. Đoạn chương trình dùng để chỉnh sửa sách như sau:

```

private void button1_Click_1(object sender, EventArgs e)
{
    // Toggle edit mode
    isEditMode = !isEditMode;

    // Enable or disable DataGridView editing based on edit mode
    dataGridView1.ReadOnly = !isEditMode;
}

```

```

        // Optionally, you can also change the appearance of the DataGridView to
        indicate edit mode
        dataGridView1.DefaultCellStyle.BackColor = isEditMode ? Color.LightGray
        : Color.White;

        if (isEditMode)
        {
            // Change button text when in edit mode
            button1.Text = "Lưu thay đổi?";
        }
        else
        {
            // Change button text when not in edit mode
            button1.Text = "Chỉnh sửa thông tin";

            // Save changes to CSV
            List<Book> books = dataGridView1.DataSource as List<Book>;
            if (books != null)
            {
                SaveBooksToCSV(books);
            }
        }
    }
}

```

- **Xóa sách:** Toàn bộ thông tin của sách sẽ được xóa ngay sau khi chọn một hàng chứa thông tin của sách đó và click chọn “Xóa sách”. Đoạn chương trình thực hiện chức năng xóa được viết như sau:

```

private void button2_Click(object sender, EventArgs e)
{
    // If in edit mode, delete the selected book
    if (isEditMode)
    {
        // Get the index of the selected row
        int selectedIndex = dataGridView1.SelectedCells[0].RowIndex;

        // Get the list of books from the DataGridView DataSource
        List<Book> books = dataGridView1.DataSource as List<Book>;

        // Ensure there is a valid list and a valid selected index
        if (books != null && selectedIndex >= 0 && selectedIndex <
books.Count)
        {
            // Remove the selected book from the list
            books.RemoveAt(selectedIndex);

            // Update the DataGridView DataSource with the updated list of
books
            dataGridView1.DataSource = null;
            dataGridView1.DataSource = books;

            // Optionally, you can refresh the DataGridView to ensure
changes are reflected immediately
            dataGridView1.Refresh();
        }
    }
}

```

```

        // Save changes to CSV
        SaveBooksToCSV(books);

        // Change button text back to "Xóa sách" after saving changes
        button2.Text = "Xóa sách";

        // Change DataGridView back to white background after saving
changes
        dataGridView1.DefaultCellStyle.BackColor = Color.White;
    }
    else
    {
        MessageBox.Show("No book selected or invalid selection.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else
{
    // Toggle edit mode
    isEditMode = !isEditMode;

    // Enable or disable DataGridView editing based on edit mode
    dataGridView1.ReadOnly = !isEditMode;

    // Optionally, you can also change the appearance of the
DataGridView to indicate edit mode
    dataGridView1.DefaultCellStyle.BackColor = isEditMode ?
Color.LightGray : Color.White;

    // Change button text to "Lưu thay đổi" when in edit mode
    button2.Text = isEditMode ? "Lưu thay đổi" : "Xóa sách";
}
}

```

Quản lý độc giả:

- Nhập độc giả từ cơ sở dữ liệu: Cho phép nhập các thông tin của sách từ cơ sở dữ liệu được thiết lập sẵn(ở đây nhóm chúng em sử dụng file .csv để làm cơ sở dữ liệu cho đối tượng độc giả). Dưới đây là các trường dữ liệu của cơ sở dữ liệu về độc giả: Mã độc giả, tên độc giả, loại độc giả và phí.

P001	Name1	Chil	1000
P002	Name2	Chil	1000
P003	Name3	Old	0
P004	Name4	Chil	1000
P008	Name8	Adult	3000
P009	Name9	Adult	3000
P010	Name10	Adult	3000
P011	Name11	Chil	1000

Dưới đây là đoạn chương trình dùng để đọc dữ liệu của đối tượng sách từ cơ sở dữ liệu đã có sẵn:

```
private void Person_AddDB_Click(object sender, EventArgs e)
{
    // Read books from CSV
    List<Person> Persons = ReadPersonFromCSV("Person.csv");

    // Display books in DataGridView
    dataGridView1.DataSource = Persons;
}

private List<Person> ReadPersonFromCSV(string filePath)
{
    List<Person> Persons = new List<Person>();

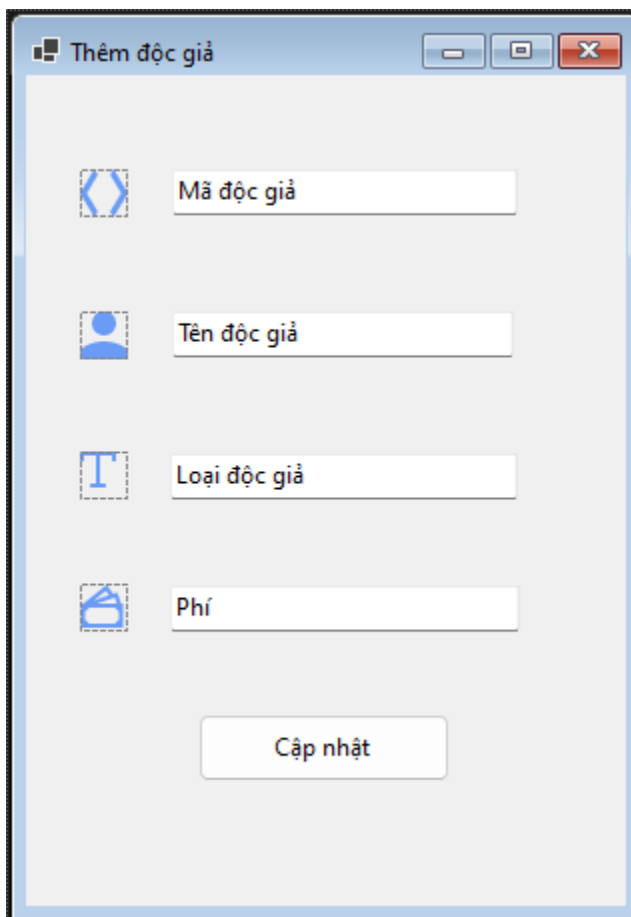
    try
    {
        using (var reader = new StreamReader(filePath))
        {
            // Read each line in the file
            string line;
            while ((line = reader.ReadLine()) != null)
            {
                // Split the line into parts
                string[] parts = line.Split(',');

                // Check if all fields are present
                if (parts.Length >= 4)
                {
                    // Create a Person object from the data
                    Person Person = new Person
                    {
                        ID = parts[0],
                        Name = parts[1],
                        Description = parts[2],
                        Fee = long.Parse(parts[3],
CultureInfo.InvariantCulture)
                    };
                    Persons.Add(Person);
                }
                else
                {
                    // Handle invalid line
                    Console.WriteLine("Invalid line: " + line);
                }
            }
        }

        //MessageBox.Show("Successfully read file: " + filePath);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error reading CSV file: " + ex.Message);
    }
}
```

```
    return Persons;  
}
```

- Lần lượt dữ liệu sẽ được đọc từ file “Person.csv” với các trường dữ liệu đã nêu tương ứng với các parts[0] đến parts[3] trong chương trình.
- Với các trường có dữ liệu chưa phù hợp với yêu cầu thì sẽ được chuyển về kiểu dữ liệu phù hợp với nó.
- Thêm độc giả thủ công: Độc giả sẽ được thêm vào và cập nhật xuống cơ sở dữ liệu. Giao diện thêm độc giả được thiết lập như sau:



- Trong chức năng thêm độc giả vẫn đảm bảo tối đa các yêu cầu về dữ liệu cần có của một đầu sách.
- Dữ liệu được nhập vào các textbox và thực thi sau khi nhấn vào button
- Dưới đây là chương trình thực hiện nhập thông tin của sách và thực hiện cập nhật ở:

- **Form “Thêm 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;

namespace TestWinform.Properties
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        public Person GetPerson()
        {
            // Retrieve data from Form2
            string personCode = textBox5.Text;
            string name = textBox6.Text;
            string description = textBox7.Text;
            long fee;
            if (!long.TryParse(textBox8.Text, out fee))
            {
                MessageBox.Show("Invalid fee.");
                return null;
            }

            // Create a new Book object
            return new Person
            {
                ID = personCode,
                Name = name,
                Description = description,
                Fee = fee,
            };
        }

        private void button2_Click(object sender, EventArgs e)
        {
            // Retrieve book data from Form2
            Person newPerson = GetPerson();

            // Check if book data is valid
            if (newPerson != null)
            {
                // Create a message to display book details
                string message = $"Reader added successfully:\n\n" +
                                $"Reader ID: {newPerson.ID}\n" +
                                $"Name: {newPerson.Name}\n" +
                                $"Description: {newPerson.Description}\n" +
```

```

        $"Fee: {newPerson.Fee}\n";

        // Show MessageBox with book details
        MessageBox.Show(message, "Success", MessageBoxButtons.OK,
        MessageBoxIcon.Information);

        // Confirm the addition and close Form3
        DialogResult = DialogResult.OK;
        Close();
    }
}

private void textBox5_MouseClick_1(object sender, MouseEventArgs e)
{
    if (textBox5.Text == "Mã độc giả")
    {
        textBox5.Text = null;
    }
}

private void textBox6_MouseClick_1(object sender, MouseEventArgs e)
{
    if (textBox6.Text == "Tên độc giả")
    {
        textBox6.Text = null;
    }
}

private void textBox7_MouseClick_1(object sender, MouseEventArgs e)
{
    if (textBox7.Text == "Loại độc giả")
    {
        textBox7.Text = null;
    }
}

private void textBox8_MouseClick_1(object sender, MouseEventArgs e)
{
    if (textBox8.Text == "Phí")
    {
        textBox8.Text = null;
    }
}
}
}

```

- **Form quản lý chính:** Thực hiện gọi Form “Thêm độc giả” ngay khi sự kiện click chuột vào button “Thêm độc giả thủ công” được kích hoạt:

```

private void button3_Click(object sender, EventArgs e)
{
    // Show Form2 to add a new book
    using (Form3 form3 = new Form3())
    {
        if (form3.ShowDialog() == DialogResult.OK)
        {
            // Retrieve book data from Form2

```

```

        Person newPerson = form3.GetPerson();

        // Ensure newBook is not null
        if (newPerson != null)
        {
            try
            {
                // Append new book data to the CSV file
                AppendPersonToCSV(newPerson, "Person.csv");

                // Display the updated list of books in DataGridView
                List<Person> csvPersons =
ReadPersonFromCSV("Person.csv");
                dataGridView1.DataSource = csvPersons;
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error adding book to CSV file:
{ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

- **Cập nhật về cơ sở dữ liệu:** Dữ liệu sau khi được nhập thủ công phải được lưu lại cơ sở dữ liệu để đảm bảo tính cập nhật dữ liệu. Đoạn chương trình sau thực hiện thêm dữ liệu vào cơ sở dữ liệu sẵn có:

```

private void AppendPersonToCSV(Person person, string filePath)
{
    try
    {
        // Open the CSV file in append mode
        using (var writer = new StreamWriter(filePath, true))
        {
            // Write the new book data to the end of the file
            writer.WriteLine($"{person.ID},{person.Name},{person.Description},{person.Fee}");
        }

        MessageBox.Show("Successfully added book to CSV file.", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        throw new Exception($"Error appending book to CSV file:
{ex.Message}");
    }
}

private void SavePersonToCSV(List<Person> Persons)
{
    try

```



```

{
    // Specify the path of the CSV file
    string filePath = "Person.csv";

    // Create or overwrite the CSV file
    using (var writer = new StreamWriter(filePath))
    {
        // Write each book as a line in the CSV file
        foreach (Person person in Persons)
        {
            writer.WriteLine($"{person.ID},{person.Name},{person.Description},{person.Fee}");
        }
    }

    MessageBox.Show("Successfully saved data to CSV file.", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (Exception ex)
{
    MessageBox.Show($"Error saving data to CSV file: {ex.Message}",
        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

- **Chỉnh sửa thông tin độc giả:** Các thông tin của độc giả được chỉnh sửa trực tiếp trên DataGridView được hiển thị trên Form quản lý chính và cập nhật về cơ sở dữ liệu. Đoạn chương trình dùng để chỉnh sửa sách như sau:

```

private void button4_Click(object sender, EventArgs e)
{
    // Toggle edit mode
    isEditMode = !isEditMode;

    // Enable or disable DataGridView editing based on edit mode
    dataGridView1.ReadOnly = !isEditMode;

    // Optionally, you can also change the appearance of the DataGridView to
    indicate edit mode
    dataGridView1.DefaultCellStyle.BackColor = isEditMode ? Color.LightGray
    : Color.White;

    if (isEditMode)
    {
        // Change button text when in edit mode
        button4.Text = "Lưu thay đổi?";
    }
    else
    {
        // Change button text when not in edit mode
        button4.Text = "Chỉnh sửa thông tin";

        // Save changes to CSV
        List<Person> Persons = dataGridView1.DataSource as List<Person>;
        if (Persons != null)

```

```

        {
            SavePersonToCSV(Persons);
        }
    }
}

```

- **Xóa độc giả:** Toàn bộ thông tin của độc giả sẽ được xóa ngay sau khi chọn một hàng chứa thông tin của sách đó và click chọn “Xóa độc giả”. Đoạn chương trình thực hiện chức năng xóa được viết như sau:

```

private void button5_Click(object sender, EventArgs e)
{
    // If in edit mode, delete the selected Person
    if (isEditMode)
    {
        // Get the index of the selected row
        int selectedIndex = dataGridView1.SelectedCells[0].RowIndex;

        // Get the list of Persons from the DataGridView DataSource
        List<Person> Persons = dataGridView1.DataSource as List<Person>;

        // Ensure there is a valid list and a valid selected index
        if (Persons != null && selectedIndex >= 0 && selectedIndex <
Persons.Count)
        {
            // Remove the selected Person from the list
            Persons.RemoveAt(selectedIndex);

            // Update the DataGridView DataSource with the updated list of
Persons
            dataGridView1.DataSource = null;
            dataGridView1.DataSource = Persons;

            // Optionally, you can refresh the DataGridView to ensure changes
are reflected immediately
            dataGridView1.Refresh();

            // Save changes to CSV
            SavePersonToCSV(Persons);

            // Change button text back to "Xóa sách" after saving changes
            button5.Text = "Xóa độc giả";

            // Change DataGridView back to white background after saving
changes
            dataGridView1.DefaultCellStyle.BackColor = Color.White;
        }
        else
        {
            MessageBox.Show("No Person selected or invalid selection.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {

```

```

        // Toggle edit mode
        isEditMode = !isEditMode;

        // Enable or disable DataGridView editing based on edit mode
        dataGridView1.ReadOnly = !isEditMode;

        // Optionally, you can also change the appearance of the DataGridView
        to indicate edit mode
        dataGridView1.DefaultCellStyle.BackColor = isEditMode ?
        Color.LightGray : Color.White;

        // Change button text to "Lưu thay đổi" when in edit mode
        button5.Text = isEditMode ? "Lưu thay đổi" : "Xóa sách";
    }
}

```

Quản lý thẻ thư viện:

- **Nhập thẻ thư viện từ cơ sở dữ liệu:** Cho phép nhập các thông tin của thẻ từ cơ sở dữ liệu được thiết lập sẵn (ở đây nhóm chúng em sử dụng file .csv để làm cơ sở dữ liệu cho đối tượng thẻ thư viện). Dưới đây là các trường dữ liệu của cơ sở dữ liệu về thẻ thư viện: Mã độc giả, mã thẻ, hạn thẻ và sách đã mượn.

P001	3167	4/25/2025	B001
P002	8748	4/26/2025	B002
P004	2971	4/28/2025	B004
P005	4092	4/29/2025	B005
P007	9991	5/1/2025	B007
P008	2329	5/2/2025	B008
P009	6258	5/3/2025	B009
P010	2653	5/4/2025	B010

Đoạn chương trình nhập thông tin thẻ thư viện từ cơ sở dữ liệu như sau:

```

private void button6_MouseClick(object sender, MouseEventArgs e)
{
    List<Card> csvCards = ReadCardFromCSV("Card.csv");

    // Display CSV books in DataGridView
    dataGridView1.DataSource = csvCards;
}

private List<Card> ReadCardFromCSV(string filePath)
{
    List<Card> cards = new List<Card>();

    try
    {
        using (var reader = new StreamReader(filePath))

```

```

    {
        // Read each line in the file
        string line;
        while ((line = reader.ReadLine()) != null)
        {
            // Split the line into parts
            string[] parts = line.Split(',');

            // Check if all fields are present
            if (parts.Length >= 4)
            {
                // Create a Book object from the data
                Card card = new Card
                {
                    ReaderID = parts[0],
                    CardID = parts[1],
                    CardExp = parts[2],
                    BookBorrow = parts[3]
                };
                cards.Add(card);
            }
            else
            {
                // Handle invalid line
                Console.WriteLine("Invalid line: " + line);
            }
        }
    }

    // MessageBox.Show("Successfully read file: " + filePath);
}
catch (Exception ex)
{
    MessageBox.Show("Error reading CSV file: " + ex.Message);
}

return cards;
}

```

- **Chức năng thêm thẻ:** Thẻ thư viện được thêm thủ công(Chỉ dành cho những đối tượng đã đăng ký độc giả) sau khi thẻ được thêm nó sẽ hiển thị lên GridViewData và được cập nhật về cơ sở dữ liệu thông qua một Form “Thêm thẻ”.
- Khi nhấn vào button thêm thẻ thì **đoạn chương trình sau thực hiện chức năng gọi form”Thêm thẻ”**:

```

private void button10_MouseClick(object sender, MouseEventArgs e)
{
    using (Form4 form4 = new Form4())
    {
        if (form4.ShowDialog() == DialogResult.OK)
        {
            // Retrieve card data from Form4
            Card newCard = form4.GetCard();

```

```

// Ensure newCard is not null
if (newCard != null)
{
    try
    {
        // Append new card data to the CSV file
        AppendCardToCSV(newCard, "Card.csv");

        // Display the updated list of cards in DataGridView
        List<Card> csvCards = ReadCardFromCSV("Card.csv");
        dataGridView1.DataSource = csvCards;
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error adding card to CSV file:
{ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}
}
}

```

- Dưới đây là chương trình của form:

The screenshot shows a Windows application window titled "Thêm thẻ" (Add Card). Inside the window, there are three input fields arranged vertically. The first field is labeled "Mã độc giả" (Reader Code) and has a magnifying glass icon to its left. The second field is labeled "Mã thẻ" (Card Code) and has a card icon to its left. The third field is labeled "Hạn thẻ (mm/dd/yyyy)" (Card Expiry Date) and has a calendar icon to its left. Below these fields is a button labeled "Cập nhật" (Update). The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace TestWinform.Properties
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        public Card GetCard()
        {
            // Retrieve data from Form2
            string personID = textBox1.Text;
            string cardID = textBox2.Text;
            string date = textBox3.Text;
            string bookBorrow = " ";

            // Create a new Book object
            return new Card
            {
                ReaderID = personID,
                CardID = cardID,
                CardExp = date,
                BookBorrow = bookBorrow,
            };
        }

        private void textBox1_MouseClick(object sender, MouseEventArgs e)
        {
            if (textBox1.Text == "Mã độc giả")
            {
                textBox1.Text = null;
            }
        }

        private void textBox2_MouseClick(object sender, MouseEventArgs e)
        {
            if (textBox2.Text == "Mã thẻ")
            {
                textBox2.Text = null;
            }
        }

        private void textBox3_MouseClick(object sender, MouseEventArgs e)
        {
            if (textBox3.Text == "Hạn thẻ (mm/dd/yyyy)")
            {
                textBox3.Text = null;
            }
        }

        private void button1_MouseClick(object sender, MouseEventArgs e)
        {
            // Retrieve book data from Form2

```

```

Card newCard = GetCard();

// Check if book data is valid
if (newCard != null)
{
    // Create a message to display book details
    string message = $"Card added successfully:\n\n" +
        $"Reader ID: {newCard.ReaderID}\n" +
        $"Card ID: {newCard.CardID}\n" +
        $"Card Exp: {newCard.CardExp}\n";

    // Show MessageBox with book details
    MessageBox.Show(message, "Success", MessageBoxButtons.OK,
        MessageBoxIcon.Information);

    // Confirm the addition and close Form3
    DialogResult = DialogResult.OK;
    Close();
}
}
}
}

```

- Trong chức năng thêm độc giả vẫn đảm bảo tối đa các yêu cầu về dữ liệu cần có của một đầu sách.
- Dữ liệu được nhập vào các textbox và thực thi sau khi nhấn vào button
- Dữ liệu được thêm và cập nhật về cơ sở dữ liệu qua hai hàm sau:
- **Hàm cập nhật về CSDL:**

```

private void SaveCardsToCSV(List<Card> cards)
{
    try
    {
        // Specify the path of the CSV file
        string filePath = "Card.csv";

        // Create or overwrite the CSV file
        using (var writer = new StreamWriter(filePath))
        {
            // Write each card as a line in the CSV file
            foreach (Card card in cards)
            {
                // Combine the BookBorrow array into a single string
                separated by semicolons
                string bookBorrowString = string.Join(";", card.BookBorrow);

                // Write the card data to the CSV file
                writer.WriteLine($"{card.ReaderID},{card.CardID},{card.CardExp},{bookBorrowString}");
            }
        }
    }
}

```

```

        MessageBox.Show("Successfully saved data to CSV file.", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error saving data to CSV file: {ex.Message}",
        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

- Hàm thêm thẻ vào CSDL:

```

private void AppendCardToCSV(Card card, string filePath)
{
    try
    {
        // Open the CSV file in append mode
        using (var writer = new StreamWriter(filePath, true))
        {
            // Write the new card data to the end of the file

            writer.WriteLine($"{card.ReaderID},{card.CardID},{card.CardExp},{card.BookBorrow}");
        }

        MessageBox.Show("Successfully added card to CSV file.", "Success",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        throw new Exception($"Error appending card to CSV file:
        {ex.Message}");
    }
}

```

Chức năng mượn sách: Sách sẽ được mượn thông qua:

- Nhập mã thẻ
- Nhập mã sách cần mượn
- Hệ thống sẽ tìm mã độc giả và mã sách qua cơ sở dữ liệu hiện có
- Khi hoàn thành tìm kiếm thì số lượng sách của mã đó sẽ -1 và trường BookBorrow của lớp Card sẽ được cập nhật thêm mã sách đã mượn
- Sau đó tất cả dữ liệu sẽ được cập nhật về CSDL hiện có của lớp Sách và lớp Thẻ thư viện.

Chức năng trả sách: Sách sẽ được trả thông qua:

- Nhập mã thẻ
- Nhập mã sách cần mượn
- Hệ thống sẽ tìm mã độc giả và mã sách qua cơ sở dữ liệu hiện có
- Khi hoàn thành tìm kiếm thì số lượng sách của mã đó sẽ +1 và trường BookBorrow của lớp Card sẽ được cập nhật bằng cách xóa mã sách đã trả.
- Sau đó tất cả dữ liệu sẽ được cập nhật về CSDL hiện có của lớp Sách và lớp Thẻ thư viện.

Dưới đây là đoạn chương trình để thực hiện mượn và trả sách.

```
//=====Mượn
sách=====//

private void button8_Click(object sender, EventArgs e)
{
    string cardID = Microsoft.VisualBasic.Interaction.InputBox("Nhập mã thẻ
mượn:", "Mã thẻ", "");
    string bookCode = Microsoft.VisualBasic.Interaction.InputBox("Nhập mã
sách cần mượn:", "Mã sách", "");

    // Load books from CSV
    List<Book> books = ReadBooksFromCSV("Book_Data.csv");

    // Load cards from CSV
    List<Card> cards = ReadCardFromCSV("Card.csv");

    // Find the book with the given bookCode
    Book borrowedBook = books.FirstOrDefault(book => book.BookCode ==
bookCode);

    // Find the card with the given cardID
    Card borrowerCard = cards.FirstOrDefault(card => card.CardID == cardID);

    if (borrowedBook != null && borrowerCard != null)
    {
        // Reduce the quantity of the borrowed book by 1
        borrowedBook.Quantity--;

        // Add the borrowed book's code to the borrower card's BookBorrow
string
        if (string.IsNullOrEmpty(borrowerCard.BookBorrow))
        {
            borrowerCard.BookBorrow = bookCode;
        }
        else
        {
            borrowerCard.BookBorrow += " " + bookCode; // Using ';' as
delimiter
    }
}
```

```

    }

    // Save changes to the CSV files
    SaveBooksToCSV(books);
    SaveCardsToCSV(cards);

    MessageBox.Show("Sách đã được mượn thành công.", "Thông báo",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Không tìm thấy thẻ hoặc sách tương ứng.", "Lỗi",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    dataGridView1.DataSource = cards;
}

//=====Trả
sách=====//
private void button7_MouseClick(object sender, MouseEventArgs e)
{
    string cardID = Microsoft.VisualBasic.Interaction.InputBox("Nhập mã thẻ
    trả:", "Mã thẻ", "");
    string bookCode = Microsoft.VisualBasic.Interaction.InputBox("Nhập mã
    sách cần trả:", "Mã sách", "");

    // Load books from CSV
    List<Book> books = ReadBooksFromCSV("Book_Data.csv");

    // Load cards from CSV
    List<Card> cards = ReadCardFromCSV("Card.csv");

    // Find the book with the given bookCode
    Book borrowedBook = books.FirstOrDefault(book => book.BookCode ==
    bookCode);

    // Find the card with the given cardID
    Card borrowerCard = cards.FirstOrDefault(card => card.CardID == cardID);

    if (borrowedBook != null && borrowerCard != null)
    {
        // Reduce the quantity of the borrowed book by 1
        borrowedBook.Quantity--;

        // Add the borrowed book's code to the borrower card's BookBorrow
        string
        if (string.IsNullOrEmpty(borrowerCard.BookBorrow))
        {
            borrowerCard.BookBorrow = bookCode;
        }
        else
        {
            borrowerCard.BookBorrow = borrowerCard.BookBorrow.Replace(" " +
            bookCode, "");
        }

        // Save changes to the CSV files
        SaveBooksToCSV(books);
    }
}

```

```

        SaveCardsToCSV(cards);

        MessageBox.Show("Sách đã được trả thành công.", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Không tìm thấy thẻ hoặc sách tương ứng.", "Lỗi",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    dataGridView1.DataSource = cards;
}

```

- **Xóa thẻ:** Chức năng cuối cùng của chương trình là xóa thông tin thẻ thư viện, chọn một hàng chứa thông tin của thẻ sau đó chọn “Xóa thẻ” thì toàn bộ thông tin của thẻ sẽ bị xóa và cập nhật lại về CSDL. Dưới đây là đoạn chương trình với chức năng xóa thẻ.

```

private void button9_Click(object sender, EventArgs e)
{
    // If in edit mode, delete the selected Card
    if (isEditMode)
    {
        // Get the index of the selected row
        int selectedIndex = dataGridView1.SelectedCells[0].RowIndex;

        // Get the list of Cards from the DataGridView DataSource
        List<Card> Cards = dataGridView1.DataSource as List<Card>;

        // Ensure there is a valid list and a valid selected index
        if (Cards != null && selectedIndex >= 0 && selectedIndex <
Cards.Count)
        {
            // Remove the selected Card from the list
            Cards.RemoveAt(selectedIndex);

            // Update the DataGridView DataSource with the updated list of
Cards
            dataGridView1.DataSource = null;
            dataGridView1.DataSource = Cards;

            // Optionally, you can refresh the DataGridView to ensure
changes are reflected immediately
            dataGridView1.Refresh();

            // Save changes to CSV
            SaveCardsToCSV(Cards);

            // Change button text back to "Xóa sách" after saving changes
            button9.Text = "Xóa thẻ";

            // Change DataGridView back to white background after saving
changes
            dataGridView1.DefaultCellStyle.BackColor = Color.White;

```

```

        }
        else
        {
            MessageBox.Show("No Card selected or invalid selection.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        // Toggle edit mode
        isEditMode = !isEditMode;

        // Enable or disable DataGridView editing based on edit mode
        dataGridView1.ReadOnly = !isEditMode;

        // Optionally, you can also change the appearance of the
DataGridView to indicate edit mode
        dataGridView1.DefaultCellStyle.BackColor = isEditMode ?
Color.LightGray : Color.White;

        // Change button text to "Lưu thay đổi" when in edit mode
        button9.Text = isEditMode ? "Lưu thay đổi" : "Xóa thẻ";
    }
}
}

```

4. TESTCASE

Xây dựng các trường hợp kiểm định cho chương trình hoạt động .

4.1. Caseform “Quản Lý Sách”

- Nhập sách từ cơ sở dữ liệu file .csv

Quản lý thư viện

QUẢN LÝ SÁCH
Nhập sách từ CSDL
Thêm sách thủ công
Chỉnh sửa thông tin
Xóa sách

QUẢN LÝ ĐỘC GIẢ
Nhập độc giả từ CSDL
Thêm độc giả thủ công
Chỉnh sửa thông tin
Xóa độc giả

QUẢN LÝ THẺ
Nhập thẻ từ CSDL
Thêm thẻ
Mượn sách
Trả sách
Xóa thẻ

	BookCode	Title	Author	Category	Quantity
▶	B001	Book1	Author1	1	299
	B002	Book2	Author2	4	997
	B003	Book3	Author3	3	720
	B005	Book5	Author5	4	682
	B006	Book6	Author6	5	923
	B007	Book7	Author7	1	753
	B008	Book8	Author8	2	518
	B009	Book9	Author9	2	707
	B010	Book10	Author10	5	356
	B004	Book4	Author4	2	1000

- Thêm sách thủ công và cập nhật trực tiếp vào file .csv :

Quản lý thư viện

QUẢN LÝ SÁCH
Nhập sách từ CSDL
Thêm sách thủ công
Chỉnh sửa thông tin
Xóa sách

QUẢN LÝ THẺ
Nhập thẻ từ CSDL
Thêm thẻ
Xóa thẻ

Thêm sách
<> B100
VLSI - 1
Tống Văn Ôn
10
100
Cập nhật

Success
Book added successfully:
Book Code: B100
Title: VLSI - 1
Author: Tống Văn Ôn
Category: 10
Quantity: 100
OK

	BookCode	Title	Author	Category	Quantity
▶	B001	Book1	Author1	1	299
	B002	Book2	Author2	4	997
	B003	Book3	Author3	3	720
	B005	Book5	Author5	4	682
	B006	Book6	Author6	5	923
	B007	Book7	Author7	1	753
	B008	Book8	Author8	2	518
	B009	Book9	Author9	2	707
	B010	Book10	Author10	5	356
	B004	Book4	Author4	2	1000

- Cập nhật thành công :

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

Success

Successfully added book to CSV file.

OK

BookCode	Book2	Author2	4	Quantity
B001				299
B002	Book2	Author2	4	997
B003	Book3	Author3	3	720
B005	Book5	Author5	4	682
B006	Book6	Author6	5	923
B007	Book7	Author7	1	753
B008	Book8	Author8	2	518
B009	Book9	Author9	2	707
B010	Book10	Author10	5	356
B004	Book4	Author4	2	1000

- Sách mới được đưa vào cơ sở dữ liệu :

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Mượn sách

Trả sách

BookCode	Title	Author	Category	Quantity
B001	Book1	Author1	1	299
B002	Book2	Author2	4	997
B003	Book3	Author3	3	720
B005	Book5	Author5	4	682
B006	Book6	Author6	5	923
B007	Book7	Author7	1	753
B008	Book8	Author8	2	518
B009	Book9	Author9	2	707
B010	Book10	Author10	5	356
B004	Book4	Author4	2	1000
B100	VLSI - 1	Tống Văn Ôn	10	100

- **Chỉnh sửa thông tin trực tiếp trên file :**

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh s

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Trả sách

Xóa thẻ

Success

Successfully saved data to CSV file.

OK

	BookCode	Title	Author	Category	Quantity
►	B100000	Book1	Author1	1	299
	B002	Book2	Author2	4	997
	B003	Book3	Author3	3	720
	B005	Book5	Author5	4	682
	B006	Book6	Author6	5	923
	B007	Book7	Author7	1	753
	B008	Book8	Author8	2	518
	B009	Book9	Author9	2	707
	B010	Book10	Author10	5	356
	B004	Book4	Author4	2	1000
	B100	VLSI - 1	Tống Văn Ôn	10	100

- **Xóa sách cập nhật trực tiếp trên file :**

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Mượn sách

Trả sách

	BookCode	Title	Author	Category	Quantity
►	B002	Book2	Author2	4	997
	B003	Book3	Author3	3	720
	B005	Book5	Author5	4	682
	B006	Book6	Author6	5	923
	B007	Book7	Author7	1	753
	B008	Book8	Author8	2	518
	B009	Book9	Author9	2	707
	B010	Book10	Author10	5	356
	B004	Book4	Author4	2	1000
	B100	VLSI - 1	Tống Văn Ôn	10	100

- **Xóa sách thành công :**

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Trả sách

Success

Successfully saved data to CSV file.

OK

BookCode	BookName	Author	Age	Quantity
B003	Book3	Author3	3	720
B005	Book5	Author5	4	682
B006	Book6	Author6	5	923
B007	Book7	Author7	1	753
B008	Book8	Author8	2	518
B009	Book9	Author9	2	707
B010	Book10	Author10	5	356
B004	Book4	Author4	2	1000
B100	VLSI - 1	Tổng Văn Ôn	10	100

4.2. Caseform “Quản Lý Độc Giả”

- Nhập độc giả từ cơ sở dữ liệu file .csv :

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Xóa thẻ

ID	Name	Description	Fee
P001	Name1	Child	1000
P002	Name2	Child	1000
P003	Name3	Old	0
P004	Name4	Child	1000
P008	Name8	Adult	3000
P009	Name9	Adult	3000
P010	Name10	Adult	3000
P011	Name11	Child	1000

- Thêm độc giả thủ công :

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL Thêm sách thủ công

Chỉnh sửa thông tin Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL Thêm độc giả thủ công

Chỉnh s...

QUẢN LÝ THẺ

Nhập thẻ từ CSDL Thêm thẻ

Xóa thẻ

ID	Name	Description	Fee
P001	Name1	Chil	1000
P002	Name2	Chil	1000
P003	Name3	Old	0
P004	Name4	Chil	1000
P008	Name8	Adult	3000
P009	Name9	Adult	3000
P010	Name10	Adult	3000
P011	Name11	Chil	1000

Thẻ...

Success

Reader added successfully:

Reader ID: P100

Name: Trần Thiên Phúc

Description: Adult

Fee: 3000

- Thêm độc giả hoàn tất :

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL Thêm sách thủ công

Chỉnh sửa thông tin Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL Thêm độc giả thủ công

Chỉnh sửa thông tin Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL Thêm thẻ

Mượn sách Trả sách Xóa thẻ

ID	Name	Description	Fee
P001	Name1	Chil	1000
P002	Name2	Chil	1000
P003	Name3	Old	0
P004	Name4	Chil	1000
P008	Name8	Adult	3000
P009	Name9	Adult	3000
P010	Name10	Adult	3000
P011	Name11	Chil	1000
P100	Trần Thiên Phúc	Adult	3000

- **Chỉnh sửa thông tin file độc giả :**

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL
Thêm sách thủ công

Chỉnh sửa thông tin
Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL
Thêm độc giả thủ công

Chỉnh sửa thông tin
Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL
Thêm thẻ

Xóa thẻ

ID	Name	Description	Fee
P200000	ABCDXYZ	ChillChill	0
P002	Name2	Chil	1000
P003	Name3	Old	0
P004	Name4	Chil	1000
P008	Name8	Adult	3000
P009	Name9	Adult	3000
P010	Name10	Adult	3000
P011	Name11	Chil	1000
P100	Trần Thiên Phúc	Adult	3000

Success

Successfully saved data to CSV file.

OK

- **Xóa độc giả :**

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL
Thêm sách thủ công

Chỉnh sửa thông tin
Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL
Thêm độc giả thủ công

Chỉnh sửa thông tin
Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL
Thêm thẻ

Mượn sách
Trả sách
Xóa thẻ

ID	Name	Description	Fee
P002	Name2	Chil	1000
P003	Name3	Old	0
P004	Name4	Chil	1000
P008	Name8	Adult	3000
P009	Name9	Adult	3000
P010	Name10	Adult	3000
P011	Name11	Chil	1000
P100	Trần Thiên Phúc	Adult	3000

- Xóa độc giả thành công :

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL
Thêm sách thủ công

Chỉnh sửa thông tin
Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL
Thêm độc giả thủ công

Chỉnh sửa thông tin
Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL
Thêm thẻ

Mượn sách
Trả sách
Xóa thẻ

ID	Name
P003	Name3
P004	Name4
P008	Name8
P009	Name9
P010	Name10
P011	Name11
P100	Trần Thiên Phúc

Success

Successfully saved data to CSV file.

OK

4.3. Caseform “ Quản Lý Thẻ”

- Nhập thẻ trực tiếp từ cơ sở dữ liệu file .csv :

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL
Thêm sách thủ công

Chỉnh sửa thông tin
Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL
Thêm độc giả thủ công

Chỉnh sửa thông tin
Xóa độc giả

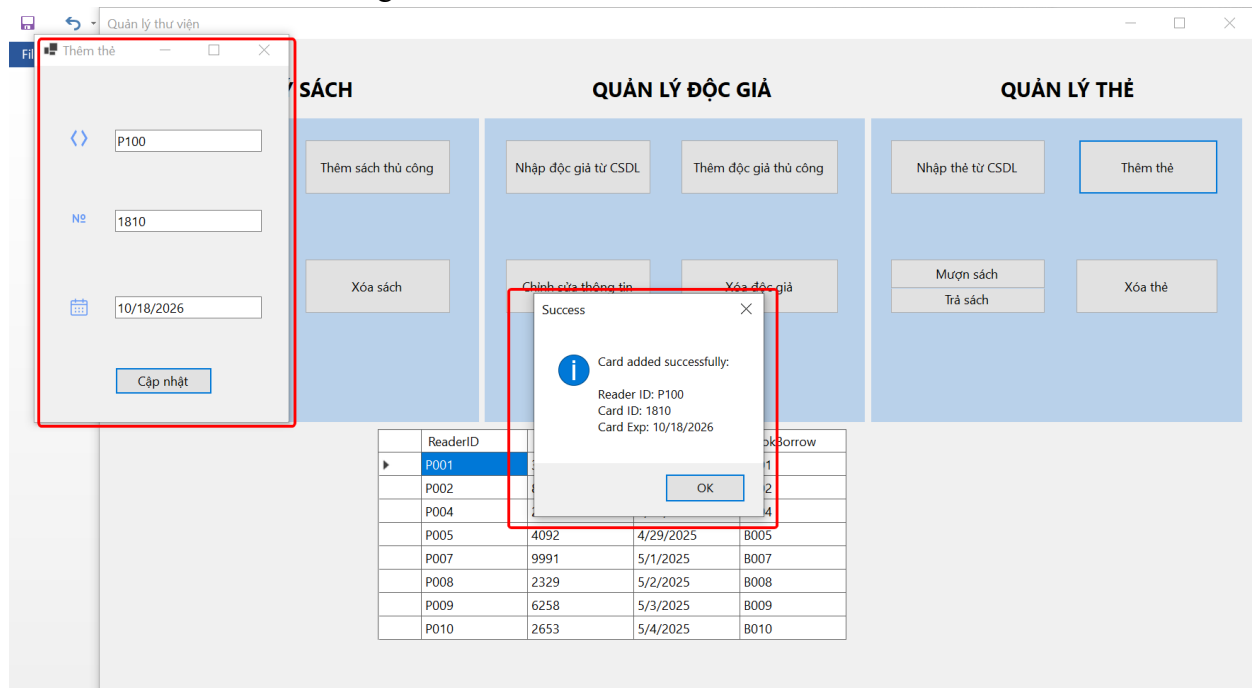
QUẢN LÝ THẺ

Nhập thẻ từ CSDL
Thêm thẻ

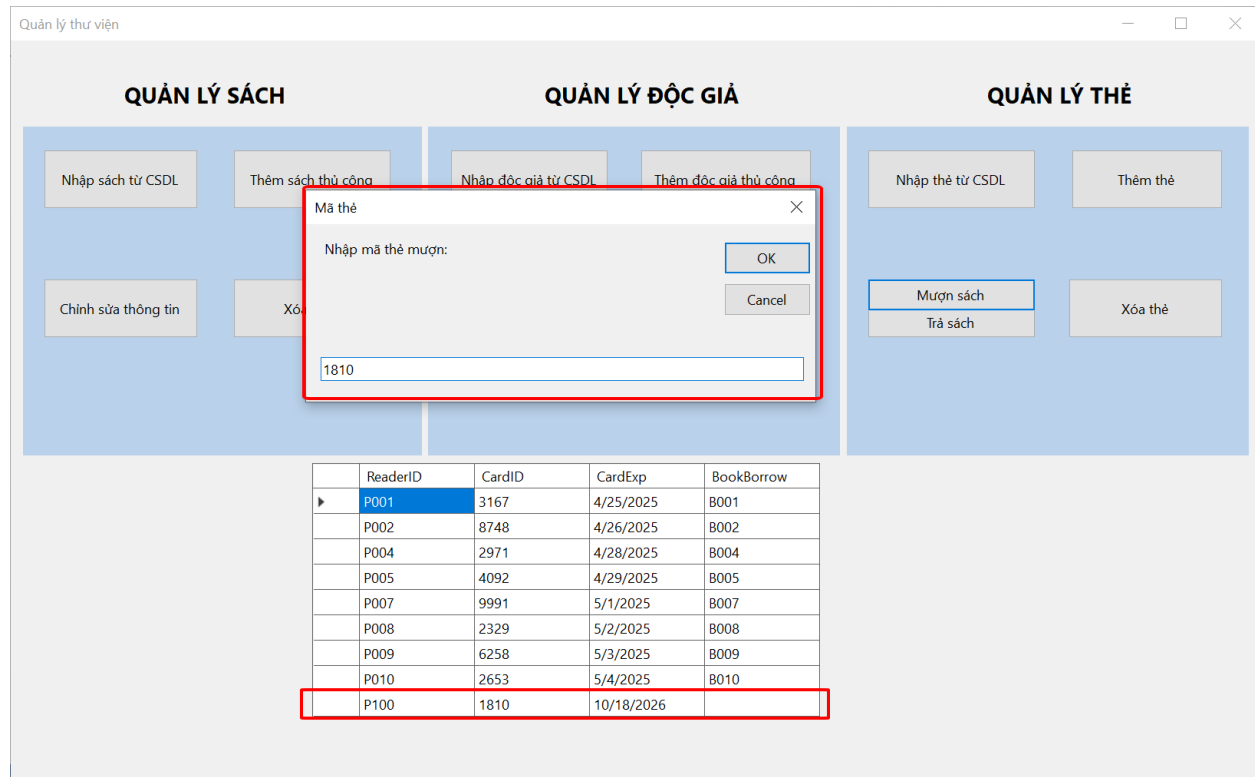
Mượn sách
Trả sách
Xóa thẻ

ReaderID	CardID	CardExp	BookBorrow
P001	3167	4/25/2025	B001
P002	8748	4/26/2025	B002
P004	2971	4/28/2025	B004
P005	4092	4/29/2025	B005
P007	9991	5/1/2025	B007
P008	2329	5/2/2025	B008
P009	6258	5/3/2025	B009
P010	2653	5/4/2025	B010

- Thêm thẻ thủ công :



- Mượn sách từ CSDL “Quản Lý Sách “ đã có :



- Nhập mã sách cần mượn:

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL Thêm sách thủ công

Chỉnh sửa thông tin Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL Thêm độc giả thủ công

QUẢN LÝ THẺ

Nhập thẻ từ CSDL Thêm thẻ

Mượn sách Trả sách Xóa thẻ

Mã sách

Nhập mã sách cần mượn:

OK Cancel

B100

ReaderID	C		
P001	3167	4/25/2025	B001
P002	8748	4/26/2025	B002
P004	2971	4/28/2025	B004
P005	4092	4/29/2025	B005
P007	9991	5/1/2025	B007
P008	2329	5/2/2025	B008
P009	6258	5/3/2025	B009
P010	2653	5/4/2025	B010
P100	1810	10/18/2026	

- Mượn sách “Thành Công” :

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL Thêm sách thủ công

Chỉnh sửa thông tin Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL Thêm độc giả thủ công

Chỉnh sửa thông tin Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL Thêm thẻ

Mượn sách Trả sách Xóa thẻ

Thông báo

Sách đã được mượn thành công.

OK

ReaderID	C		
P001	3167	4/25/2025	B001
P002	8748	4/26/2025	B002
P004	2971	4/28/2025	B004
P005	4092	4/29/2025	B005
P007	9991	5/1/2025	B007
P008	2329	5/2/2025	B008
P009	6258	5/3/2025	B009
P010	2653	5/4/2025	B010
P100	1810	10/18/2026	

- Dữ liệu được cập nhật vào file:

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Trả sách

Xóa thẻ

	ReaderID	CardID	CardExp	BookBorrow
▶	P001	3167	4/25/2025	B001
	P002	8748	4/26/2025	B002
	P004	2971	4/28/2025	B004
	P005	4092	4/29/2025	B005
	P007	9991	5/1/2025	B007
	P008	2329	5/2/2025	B008
	P009	6258	5/3/2025	B009
	P010	2653	5/4/2025	B010
	P100	1810	10/18/2026	B100

- Trả sách :

+ Nhập mã thẻ và mã sách cần trả :

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Trả sách

Xóa thẻ

Mã thẻ

Nhập mã thẻ trả:

OK

Cancel

1810

	ReaderID	CardID	CardExp	BookBorrow
▶	P001	3167	4/25/2025	B001
	P002	8748	4/26/2025	B002
	P004	2971	4/28/2025	B004
	P005	4092	4/29/2025	B005
	P007	9991	5/1/2025	B007
	P008	2329	5/2/2025	B008
	P009	6258	5/3/2025	B009
	P010	2653	5/4/2025	B010
	P100	1810	10/18/2026	B100

- Trả sách thành công:

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Trả sách

Xóa thẻ

Thông báo

Sách đã được trả thành công.

OK

	ReaderID	CardID	CardExp	BookBorrow
►	P001	3167	4/25/2025	B001
	P002	8748	4/26/2025	B002
	P004	2971	4/28/2025	B004
	P005	4092	4/29/2025	B005
	P007	9991	5/1/2025	B007
	P008	2329	5/2/2025	B008
	P009	6258	5/3/2025	B009
	P010	2653	5/4/2025	B010
	P100	1810	10/18/2026	B100

- Tự động cập nhật file:

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL

Thêm sách thủ công

Chỉnh sửa thông tin

Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL

Thêm độc giả thủ công

Chỉnh sửa thông tin

Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL

Thêm thẻ

Mượn sách

Trả sách

Xóa thẻ

	ReaderID	CardID	CardExp	BookBorrow
►	P001	3167	4/25/2025	B001
	P002	8748	4/26/2025	B002
	P004	2971	4/28/2025	B004
	P005	4092	4/29/2025	B005
	P007	9991	5/1/2025	B007
	P008	2329	5/2/2025	B008
	P009	6258	5/3/2025	B009
	P010	2653	5/4/2025	B010
	P100	1810	10/18/2026	

- Xóa thẻ trực tiếp:

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL Thêm sách thủ công

Chỉnh sửa thông tin Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL Thêm độc giả thủ công

Chỉnh sửa thông tin Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL Thêm thẻ

Mượn sách Trả sách Xóa thẻ

	ReaderID	CardID	CardExp	BookBorrow
	P001	3167	4/25/2025	B001
	P002	8748	4/26/2025	B002
	P004	2971	4/28/2025	B004
	P005	4092	4/29/2025	B005
	P007	9991	5/1/2025	B007
	P008	2329	5/2/2025	B008
	P009	6258	5/3/2025	B009
	P010	2653	5/4/2025	B010
▶	P100	1810	10/18/2026	

- Cập nhật về cơ sở dữ liệu

Quản lý thư viện

QUẢN LÝ SÁCH

Nhập sách từ CSDL Thêm sách thủ công

Chỉnh sửa thông tin Xóa sách

QUẢN LÝ ĐỘC GIẢ

Nhập độc giả từ CSDL Thêm độc giả thủ công

Chỉnh sửa thông tin Xóa độc giả

QUẢN LÝ THẺ

Nhập thẻ từ CSDL Thêm thẻ

Mượn sách Trả sách Xóa thẻ

	ReaderID	CardID	CardExp	BookBorrow
▶	P001	3167	4/25/2025	B001
	P002	8748	4/26/2025	B002
	P004	2971	4/28/2025	B004
	P005	4092	4/29/2025	B005
	P007	9991	5/1/2025	B007
	P008	2329	5/2/2025	B008
	P009	6258	5/3/2025	B009
	P010	2653	5/4/2025	B010


5. GITHUB

Commit project lên trang GitHub .

Link url : [phoai29/LibraryManagement \(github.com\)](https://github.com/phoai29/LibraryManagement)

- Sau khi hoàn thành đồ án chúng em đưa lên github để thầy có thể dễ dàng kiểm tra và đánh giá quá trình làm việc cũng như thực hiện dự án này :

README



fetel@HCMUS
KHOA ĐIỆN TỬ - VIỄN THÔNG

Phần mềm Quản lý Thư viện

Giới thiệu

- Đây là ứng dụng được sử dụng làm đồ án cuối kì cho môn Thực hành Lập Trình Hướng Đối Tượng.
- Ứng dụng được xây dựng theo mô hình 3 lớp, sử dụng Entity Framework, cài đặt giao diện bằng C# Winforms.
- Thông tin chi tiết về project có thể tìm đọc trong file Report.docx

Giảng viên hướng dẫn

- ThS. Cao Trần Bảo Thương - Khoa Điện Tử - Viễn Thông , Trường Đại Học Khoa Học Tự Nhiên - ĐHQG HCM

Sinh viên thực hiện

STT	MSSV	Họ và tên	Email
1	21207195	Huynh Thi Ngoc Phuc	21207195@hcmus.edu.vn
2	21207120	Pham Hoai An	21207120@hcmus.edu.vn
3	21207077	Tran Thien Phuc	21207077@hcmus.edu.vn

Chạy thử project

Yêu cầu để thực thi project

- Visual Studio 2022
- .NET Framework 8.0
- Database file .csv
- Siticone UI/UX Framework

Bước 1: Khởi tạo database

- Thực thi file .csv để khởi tạo database

Bước 2: Thực thi chương trình

- Mở solution trong Visual Studio
- Tải về bản Nuget nếu VS không tự động tải: Entity Framework 8.0.0, Siticone UI/UX Framework

6. ĐÁNH GIÁ – NHIỆM VỤ

Ordinal Number	Student Name	Student ID	Task	Complete	Point
01	Pham Hoai An (Team leader)	21207120	<ul style="list-style-type: none">- Login form- Quản lý thẻ và form liên quan	100%	
02	Huynh Thi Ngoc Phuc	21207195	<ul style="list-style-type: none">- Quản lý sách và form liên quan	100%	
03	Tran Thien Phuc	21207077	<ul style="list-style-type: none">- Quản lý độc giả và form liên quan	100%	