

TRƯỜNG ĐẠI HỌC PHENIKAA
KHOA CÔNG NGHỆ THÔNG TIN

~~~~~\*~~~~~



**PHENIKAA**  
UNIVERSITY

**BÀI TẬP LỚN**  
**MÔN CÔNG NGHỆ. NET**

***ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÝ SINH VIÊN***

|                             |                       |
|-----------------------------|-----------------------|
| <i>Lớp học phần</i>         | : Công nghệ. Net 05   |
| <i>Giảng viên hướng dẫn</i> | : Th.S Phạm Ngọc Hưng |
| <i>Nhóm thực hiện</i>       | : Nhóm 5              |

## DANH SÁCH THÀNH VIÊN VÀ PHÂN CÔNG NHIỆM VỤ

| Họ và tên      | Mã sinh viên | Công việc | Thang đánh giá(10) |
|----------------|--------------|-----------|--------------------|
| Trần Đức Thịnh | 21010636     |           |                    |
| Hồ Quỳnh Công  |              |           | 3                  |
| Phạm Hồng Phúc |              |           | 3                  |
| Phạm Đức Hiếu  |              |           | 8                  |

## Table of Contents

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| Mở đầu .....                                                                           | 5  |
| 1. Lý do chọn đề tài.....                                                              | 5  |
| 2. Mục đích nghiên cứu.....                                                            | 5  |
| 3. Phạm vi nghiên cứu.....                                                             | 5  |
| Nội dung.....                                                                          | 6  |
| Chương 1: Xác định yêu cầu .....                                                       | 6  |
| 1.1. Phân tích hiện trạng và yêu cầu.....                                              | 6  |
| 1.2. Quy trình nghiệp vụ hệ thống.....                                                 | 6  |
| Chương 2: Phân tích, thiết kế hệ thống và quy trình nghiệp vụ của phần mềm .....       | 6  |
| 2.1. Phân tích hệ thống .....                                                          | 6  |
| 2.2. Thiết kế hệ thống.....                                                            | 7  |
| 2.3. Quy trình nghiệp vụ.....                                                          | 7  |
| 2.4. Chức năng và giao diện: .....                                                     | 8  |
| Chương 3: Triển khai .....                                                             | 12 |
| 3.1. Cài đặt và chuẩn bị môi trường làm việc .....                                     | 12 |
| 3.2. Tạo dự án WPF.....                                                                | 12 |
| 3.3. Thiết kế cơ sở dữ liệu .....                                                      | 12 |
| 3.4. Xử lý dữ liệu và hiển thị lên giao diện người dùng, tối ưu hoá chương trình ..... | 16 |
| 3.5. Sử dụng Git và GitHub trong quá trình phát triển phần mềm. ....                   | 17 |
| 3.6. Thử nghiệm và triển khai hệ thống.....                                            | 18 |
| Kết luận.....                                                                          | 21 |

## Mở đầu

### 1. Lý do chọn đề tài

“Quản lý sinh viên” luôn là vấn đề được nhà trường quan tâm. Trong trường đại học hay các trường khác việc quản lý sinh viên, học sinh vẫn chưa được hoàn thiện, đôi khi vẫn còn những sai sót về thông tin sinh viên. Những thông tin cũ và sai tuy đã được sửa nhưng vẫn chưa được đồng nhất ở các văn phòng khoa, văn phòng nhà trường.

Thêm vào đó chúng em muốn thể hiện đầy đủ thông tin để việc tìm kiếm, tra cứu thông tin về học sinh, sinh viên được đầy đủ hơn, nhằm tiết kiệm thời gian. Vì vậy nhóm em chọn đề tài “Quản lý sinh viên”.

### 2. Mục đích nghiên cứu

Xây dựng chương trình “Quản lý sinh viên” nhằm hỗ trợ cho công tác quản lý thông tin sinh viên một cách tốt hơn, giúp cho việc điều chỉnh và bổ sung thông tin được thực hiện một cách dễ dàng và chính xác, việc tìm kiếm thông tin sinh viên không mất quá nhiều thời gian, nâng cao hiệu quả công việc và tiết kiệm được nhiều thời gian.

### 3. Phạm vi nghiên cứu

Nghiên cứu quản lý sinh viên theo hệ thống tín chỉ, trong đó quản lý thông tin sinh viên, quản lý môn học và quản lý điểm số của sinh viên

## Nội dung

### Chương 1: Xác định yêu cầu

#### 1.1. Phân tích hiện trạng và yêu cầu

Hệ thống quản lý sinh viên còn vấp phải sự bất cập giữa các phòng và khoa sửa đổi thông tin chưa được đồng bộ hoá. Nhiều thông tin vẫn còn bị sai lệch. Ở một số nơi, việc lưu trữ thông tin còn được lưu dưới dạng sổ sách, khiến cho việc tra cứu trở nên khó khăn. Hay hiện đại hơn một chút thì được lưu trong Excel, điều này khiến cho khi sửa đổi thông tin của sinh viên ở một phòng ban thì không được đồng bộ hoá

Yêu cầu đặt ra là chương trình phải khắc phục được những hạn chế nêu trên, việc chỉnh sửa thông tin cần phải được đồng nhất, dễ dàng nhanh chóng, chỉnh sửa phải đồng nhất

#### 1.2. Quy trình nghiệp vụ hệ thống

Hệ thống phải có chức năng cập nhật, sửa, bổ sung thông tin một cách đầy đủ và chính xác, có thể tìm kiếm thông tin, lọc theo yêu cầu.

### Chương 2: Phân tích, thiết kế hệ thống và quy trình nghiệp vụ của phần mềm

Trong phần này, chúng ta sẽ phân tích và thiết kế hệ thống của phần mềm quản lý sinh viên được viết bằng Windows Presentation Foundation (WPF). Hệ thống này cho phép người dùng thêm, sửa xoá và tìm kiếm thông tin của sinh viên trong cơ sở dữ liệu.

#### 2.1. Phân tích hệ thống

Để phân tích hệ thống, chúng ta cần xác định các chức năng cần thiết để quản lý thông tin sinh viên. Chức năng cơ bản bao gồm:

- \* Thêm sinh viên mới
- \* Xem danh sách sinh viên
- \* Tìm kiếm sinh viên
- \* Sửa thông tin sinh viên
- \* Xoá sinh viên ra khỏi danh sách

Ngoài ra, chúng ta cần xác định được các loại thông cần lưu trữ cho mỗi sinh viên. Các thông tin này bao gồm:

- \* Mã sinh viên

- \* Họ tên

- \* Ngày sinh

- \* Giới tính

- \* Địa chỉ

- \* Số điện thoại

- \* Email

- \* Khoá

Chúng ta còn quản lý thông tin môn học của sinh viên và điểm số của sinh viên, có thể thêm, sửa, xoá môn học và cập nhật điểm của sinh viên.

## 2.2. Thiết kế hệ thống

Sau khi đã phân tích hệ thống, chúng ta cần thiết kế cấu trúc của hệ thống. Đầu tiên, chúng ta sẽ thiết kế giao diện sẽ bao gồm các điều khiển để người dùng có thể thực hiện lưu trữ các thông tin cần thiết đối với sinh viên

Sau đó, sẽ viết code để kết nối giao diện người dùng với cơ sở dữ liệu. Code này sẽ cho phép người dùng thực hiện các chức năng quản lý sinh viên và thao tác với cơ sở dữ liệu.

## 2.3. Quy trình nghiệp vụ

Giảng viên và sinh viên đăng nhập vào hệ thống với tài khoản được cấp, mỗi sinh viên được thêm mới thì tài khoản sử dụng cho sinh viên đó sẽ tự động được thêm vào hệ thống theo cấu trúc: tài khoản sẽ là mã sinh viên và mật khẩu sẽ là ngày sinh

của sinh viên (gồm 8 chữ số). Sau khi đăng nhập thành công vào hệ thống về phía giảng viên:

\* Về phía giảng viên: Giảng viên có thể thêm, sửa, xoá thông tin liên quan đến sinh viên; thêm, sửa xoá với thông tin môn học và quản lý thông tin đăng kí học của sinh viên, quản lý điểm học tập của sinh viên, mọi chức năng đều có thể xuất báo cáo dưới dạng excel.

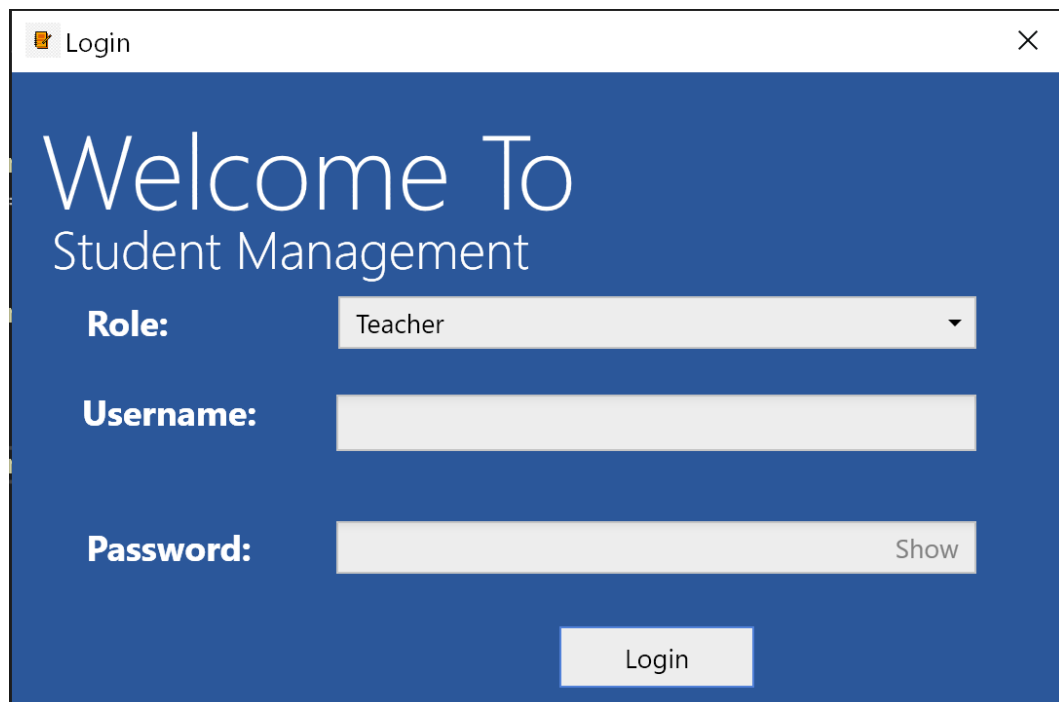
\* Về phía sinh viên: Sinh viên có thể đăng nhập, kiểm tra thông tin cá nhân của bản thân, xem điểm học tập và xem được những môn học mà mình đã đăng kí. Gửi yêu cầu cập nhật nếu thông tin cá nhân có sai sót

#### 2.4. Chức năng và giao diện:

Chức năng đầu tiên: Đăng nhập vào hệ thống

Khi mở chương trình người dùng cần cung cấp tài khoản và mật khẩu đã được cấp để có thể đăng nhập vào hệ thống và đi tiếp đến các chức năng tiếp theo. Giao diện sẽ có thành phần: Label, TextBox, PasswordBox, Button

Giao diện của chức năng





## Chức năng thứ hai: Quản lý thông tin sinh viên

Cụ thể, khi đã đăng nhập vào hệ thống thì trang đầu tiên người dùng nhìn thấy là quản lý thông tin sinh viên. Giao diện sẽ có các thành phần: Combobox, DataGrid, Label, TextBox. Người dùng có thể thực hiện việc thêm, sửa, xóa thông tin của sinh viên, tìm kiếm thông tin của sinh viên.

### Giao diện của chức năng

The screenshot shows a web application titled 'ControlPanel'. The main content area is titled 'Quản lý thông tin sinh viên' (Student Information Management). It features a search bar with the text 'Tìm kiếm:'. Below the search bar is a table with 8 columns: Mã sinh viên, Mã ngành, Họ tên, Ngày sinh, Giới tính, Số điện thoại, Email, and Khoa. The table contains 8 rows of student data. Below the table is a form for adding or updating student information. The form has fields for Mã sinh viên, Email, Tên sinh viên, Mã ngành (a dropdown menu), Số điện thoại, Khoa (a dropdown menu), Ngày sinh (a date picker), and Giới tính (a dropdown menu). At the bottom of the form are buttons for 'Add', 'Update', 'Delete', 'Clear', and 'Export'. A 'Đăng xuất' (Logout) button is located in the sidebar.

| Mã sinh viên | Mã ngành | Họ tên       | Ngày sinh            | Giới tính | Số điện thoại | Email                      | Khoa   |
|--------------|----------|--------------|----------------------|-----------|---------------|----------------------------|--------|
| sv001        | CNTT     | Trần Văn A   | 1/1/2000 12:00:00 AM | Nam       | 0987654321    | sv1@example.com            | Khoa C |
| sv0010       | CNTT     | Phạm Thị D   | 1/4/2000 12:00:00 AM | Nam       | 0987654324    | sv4@st.phenikaa-uni.edu.vn | Khoa C |
| sv00111      | CNTT     | Hoàng Văn E  | 1/5/2000 12:00:00 AM | Nam       | 0987654325    | sv5@st.phenikaa-uni.edu.vn | Khoa C |
| sv002        | CNTT     | Nguyễn Thị B | 2/1/2000 12:00:00 AM | N?        | 0987654322    | sv2@example.com            | Khoa C |
| sv003        | CNTT     | Lê Văn C     | 3/1/2000 12:00:00 AM | Nam       | 0987654323    | sv3@example.com            | Khoa C |
| sv004        | CNTT     | Phạm Thị D   | 4/1/2000 12:00:00 AM | N?        | 0987654324    | sv4@example.com            | Khoa C |
| sv005        | CNTT     | Hoàng Văn E  | 5/1/2000 12:00:00 AM | Nam       | 0987654325    | sv5@example.com            | Khoa C |

Form fields:

- Mã sinh viên:
- Email:
- Tên sinh viên:
- Mã ngành:
- Số điện thoại:
- Khoa:
- Ngày sinh:
- Giới tính:

Buttons:

Các chức năng còn lại: Quản lý thông tin môn học, Quản lý điểm, Quản lý đăng ký học

Tương tự như quản lý thông tin sinh viên các chức năng cơ bản là thêm, sửa, xóa và tạo báo cáo dưới dạng excel. Các thành phần giao diện giống như quản lý thông tin sinh viên

### Giao diện của quản lý thông tin môn học

ControlPanel

Quản lý thông tin sinh viên

Quản lý thông tin môn học

Quản lý điểm

Quản lý đăng kí

Đăng xuất

Tìm kiếm:

Mã học phần:

Tên học phần:

Tin chỉ:

Mã ngành:

Update

Delete

Add

Clear

Export

| Mã học phần | Tên học phần                   | Tin chỉ | Mã ngành |
|-------------|--------------------------------|---------|----------|
| ECO1001     | Kinh tế học đại cương          | 3       | KinhTe   |
| ECO1002     | Quản trị kinh doanh            | 4       | KinhTe   |
| ENG1001     | Tiếng Anh cơ bản               | 2       | NgoaiNgu |
| ENG1002     | Tiếng Anh giao tiếp            | 3       | NgoaiNgu |
| INT1001     | Lập trình căn bản              | 3       | CNTT     |
| INT1002     | Cấu trúc dữ liệu và giải thuật | 4       | CNTT     |

Giao diện của quản lý điểm

ControlPanel

Quản lý thông tin sinh viên

Quản lý thông tin môn học

Quản lý điểm

Quản lý đăng kí

Đăng xuất

Tim kiếm:

Lọc: <None>

| Mã sinh viên | Mã môn  | Tên môn                        | Điểm chuyên cần | Điểm giữa kì | Điểm cuối kì | Tổng kết |
|--------------|---------|--------------------------------|-----------------|--------------|--------------|----------|
| sv001        | INT1001 | Lập trình căn bản              | 9               | 8.5          | 7            | 7.8      |
| sv001        | INT1002 | Cấu trúc dữ liệu và giải thuật | 8               | 7.5          | 8            | 7.8      |
| sv002        | ECO1001 | Kinh tế học đại cương          | 8.5             | 7            | 9            | 8.15     |
| sv002        | ECO1002 | Quản trị kinh doanh            | 9               | 8            | 8.5          | 8.35     |
| sv003        | ENG1001 | Tiếng Anh cơ bản               | 8               | 7.5          | 7            | 7.3      |
| sv003        | ENG1002 | Tiếng Anh giao tiếp            | 9               | 9            | 8.5          | 8.75     |

Mã sinh viên:

Mã môn:

Tên môn:

Điểm chuyên cần:

Điểm giữa kì:

Điểm cuối kì:

Tổng kết:

Update

Export

## Giao diện của quản lý đăng kí

ControlPanel

Quản lý thông tin sinh viên

Quản lý thông tin môn học

Quản lý điểm

Quản lý đăng kí

Đăng xuất

| Mã sinh viên | Tên sinh viên | Khoá   |
|--------------|---------------|--------|
| sv001        | Trần Văn A    | Khoa C |
| sv0010       | Phạm Thị D    | Khoa C |
| sv00111      | Hoàng Văn E   | Khoa C |
| sv002        | Nguyễn Thị B  | Khoa C |
| sv003        | Lê Văn C      | Khoa C |
| sv004        | Phạm Thị D    | Khoa C |
| sv005        | Hoàng Văn E   | Khoa C |

| Mã môn học | Tên môn học           | Số tín chỉ | Mã ngành |
|------------|-----------------------|------------|----------|
| ECO1001    | Kinh tế học đại cương | 3          | KinhTế   |
| ECO1002    | Quản trị kinh doanh   | 4          | KinhTế   |

Export

Về phía sinh viên, giao diện được cấu tạo từ các Label, Button, DataGrid

Giao diện của sinh viên:

StudentRole

THÔNG TIN CƠ BẢN

Cập nhật thông tin

Mã SV

sv001

Ngày sinh

2000-01-01

Email

sv1@example.com

Mã ngành

CNTT

Họ tên

Trần Văn A

Giới tính

Nam

Số điện thoại

0987654321

Khoá

Khoa Công nghệ?  
thông tin

ĐIỂM HỌC TẬP

| MaSinhVien | MaMonHoc | TenMonHoc                      | DiemChuyenCan | DiemGiuaiKi | DiemCuoiKi | TongKet |  |
|------------|----------|--------------------------------|---------------|-------------|------------|---------|--|
| sv001      | INT1001  | Lập trình căn bản              | 9             | 8.5         | 7          | 7.8     |  |
| sv001      | INT1002  | Cấu trúc dữ liệu và giải thuật | 8             | 7.5         | 8          | 7.8     |  |

### Chương 3: Triển khai

Trong phần này chúng ta sẽ tập trung vào triển khai hệ thống được viết bằng WPF. Triển khai hệ thống bao gồm:

#### 3.1. Cài đặt và chuẩn bị môi trường làm việc

Cài đặt môi trường phát triển WPF, trong đề tài này, nhóm chúng em sử dụng bộ công cụ Visual Studio và cài đặt .NET Framework phù hợp cho ứng dụng

#### 3.2. Tạo dự án WPF

Tạo dự án dưới dạng WPF Application bằng Visual Studio. Thiết kế giao diện kéo thả bằng các thành phần đã nêu trên.

#### 3.3. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được đưa vào sử dụng là SQL Server, một trong những lý do chọn hệ quản trị cơ sở dữ liệu này vì đều được Microsoft phát triển nên rất dễ dàng để tích hợp vào với nhau trong cùng một ứng dụng WPF

Các bảng trong cơ sở dữ liệu và các ràng buộc

Bảng SinhVien: Lưu trữ các thông tin cơ bản của sinh viên

| Column Name | Data Type |
|-------------|-----------|
| MaSinhVien  |           |
| MaNganh     |           |
| TenSinhVien |           |
| NgaySinh    |           |
| GioiTinh    |           |
| SoDienThoai |           |
| Email       |           |
| Khoa        |           |

Bảng AdminAccount: Lưu trữ tài khoản, mật khẩu của giảng viên

| Column Name | Data Type |
|-------------|-----------|
| id          |           |
| username    |           |
| password    |           |

Bảng StudentAccount: Lưu trữ tài khoản của sinh viên

| Column Name | Data Type |
|-------------|-----------|
| MaSinhVien  |           |
| username    |           |
| password    |           |

Bảng Nganh: Lưu trữ thông tin về các ngành học

| Column Name | Data Type |
|-------------|-----------|
| MaNganh     |           |
| TenNganh    |           |

Bảng Diem: Lưu trữ thông tin về điểm của sinh viên

| Column Name   | Data Type |
|---------------|-----------|
| MaSinhVien    |           |
| MaMonHoc      |           |
| DiemChuyenCan |           |
| DiemGiuaKy    |           |
| DiemCuoiKy    |           |

Bảng MonHoc: Lưu trữ thông tin môn học

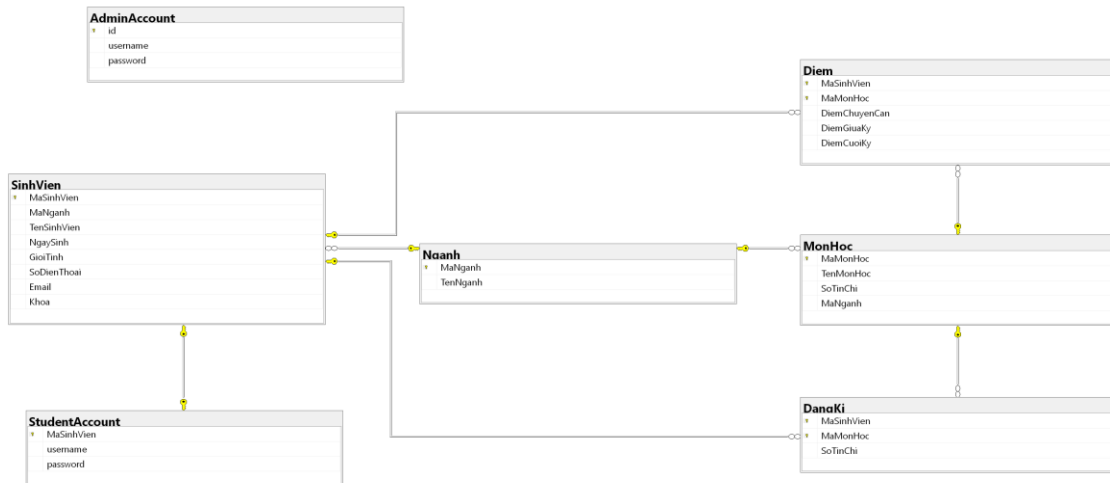
| Column Name | Data Type |
|-------------|-----------|
| MaMonHoc    |           |
| TenMonHoc   |           |
| SoTinChi    |           |
| MaNganh     |           |

Bảng DangKi: Lưu trữ thông tin đăng kí môn học của sinh viên

| Column Name | Data Type |
|-------------|-----------|
| MaSinhVien  |           |
| MaMonHoc    |           |

|          |  |
|----------|--|
| SoTinChi |  |
|----------|--|

## Sơ đồ ràng buộc của cơ sở dữ liệu



### 3.4. Xử lý dữ liệu và hiển thị lên giao diện người dùng, tối ưu hoá chương trình

Để xử lý dữ liệu và hiển thị lên giao diện người dùng trong hệ thống, bài toán được xây dựng gần như tương tự mô hình MVC (Vì trong WPF khi ta tạo một View thì sẽ có code-behind, để làm bài dễ dàng nên nhóm đã viết luôn code ở phần này).

#### 3.4.1. Xử lý dữ liệu

Thiết kế các lớp đại diện cho dữ liệu trong hệ thống (Models). Trong bài có 4 model chính là *Account* (đại diện cho dữ liệu về thông tin tài khoản), *Course* (đại diện cho dữ liệu về thông tin các môn học), *Grade* (đại diện cho dữ liệu về thông tin điểm học tập), *Student* (đại diện cho dữ liệu về thông tin sinh viên).

Sử dụng công nghệ truy cập dữ liệu ADO.NET để kết nối và tương tác với dữ liệu. ADO.NET có thể thực thi các câu lệnh truy vấn SQL bằng cách sử dụng các đối tượng SqlCommand, SqlConnection, SqlDataReader,...

#### 3.4.2. Hiển thị dữ liệu

Dữ liệu được đổ từ database vào các Models và từ đó ta thao tác với dữ liệu thông qua Models và Models sẽ cập nhật các thay đổi dữ liệu lên database

Sử dụng thành phần giao diện là DataGrid với kỹ thuật binding để hiển thị dữ liệu lên các cột tương ứng.

#### 3.4.3. Xử lý sự kiện

Xác định đúng các sự kiện từ giao diện người dùng, các sự kiện của nút nhấn và các thao tác lên DataGrid. Xử lý các ngoại lệ do người dùng gây ra, thông báo lỗi cho người dùng

#### 3.4.4. Thư viện và công cụ sử dụng

|                            |                                                                                                 |
|----------------------------|-------------------------------------------------------------------------------------------------|
| Sử dụng ngôn ngữ C#        | <a href="https://visualstudio.microsoft.com/en/vs">https://visualstudio.microsoft.com/en/vs</a> |
| Framework: WPF Application |                                                                                                 |



|                                |                                                                                                                                                                     |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AdonisUI                       | <a href="https://www.nuget.org/packages/AdonisUI/1.17.1?_src=template">https://www.nuget.org/packages/AdonisUI/1.17.1?_src=template</a>                             |
| Cơ sở dữ liệu: SQL Server 2022 |                                                                                                                                                                     |
| DocumentFormat.OpenXml         | <a href="https://www.nuget.org/packages/DocumentFormat.OpenXml/2.20.0?_src=template">https://www.nuget.org/packages/DocumentFormat.OpenXml/2.20.0?_src=template</a> |

### 3.3.5. Xử lý các ngoại lệ do chương trình gây ra

Ngoài những ngoại lệ do người dùng gây ra chúng ta cần xử lý và thông báo, thì còn có ngoại lệ của các đối tượng, phương thức được cung cấp sẵn trong WPF. Nhằm tránh crash (văng) ra khỏi chương trình làm gián đoạn công việc hiện tại và gây khó chịu trong quá trình sử dụng thì cần phải xử lý triệt để. Đặt bẫy lỗi (try\_catch) ở những phương thức, đối tượng có thể gây ra lỗi một số ví dụ như khi thực hiện truy vấn lên cơ sở dữ liệu đối tượng SqlConnection có thể văng ra một lỗi không thể kết nối đến cơ sở dữ liệu do nhà phát triển cấu hình sai phương thức kết nối. Nếu không xử lý, khi người dùng thực hiện truy vấn lên cơ sở dữ liệu sẽ gây crash chương trình.

### 3.5. Sử dụng Git và GitHub trong quá trình phát triển phần mềm.

Sử dụng Git và GitHub là một phương pháp phổ biến để quản lý mã nguồn và phiên bản trong quá trình phát triển phần mềm. Trong quá trình phát triển mỗi thành viên được giao một nhiệm vụ khác nhau, vì thế vấn đề xảy ra xung đột là điều không thể tránh khỏi. Git giúp ta giải quyết bài toán này bằng cách clone dự án về máy cá nhân rồi sau đó mỗi thành viên sẽ làm task của mình sau đó push toàn bộ phần mình hoàn thành lên GitHub. Mọi thành viên trong nhóm sẽ thấy được sự thay đổi của toàn bộ dự án. Ngoài ra, GitHub còn có quản lý yêu cầu kéo (pull request) để cho phép kiểm tra, xem xét các thay đổi trước khi chúng ta thêm (merge) vào dự án của nhóm.

Ngoài ra, khi sửa đổi code sẽ có một số vấn đề phát sinh ngẫu nhiên mà bản thân người phát triển không lường trước được thì lúc này Git sẽ cho phép chúng ta

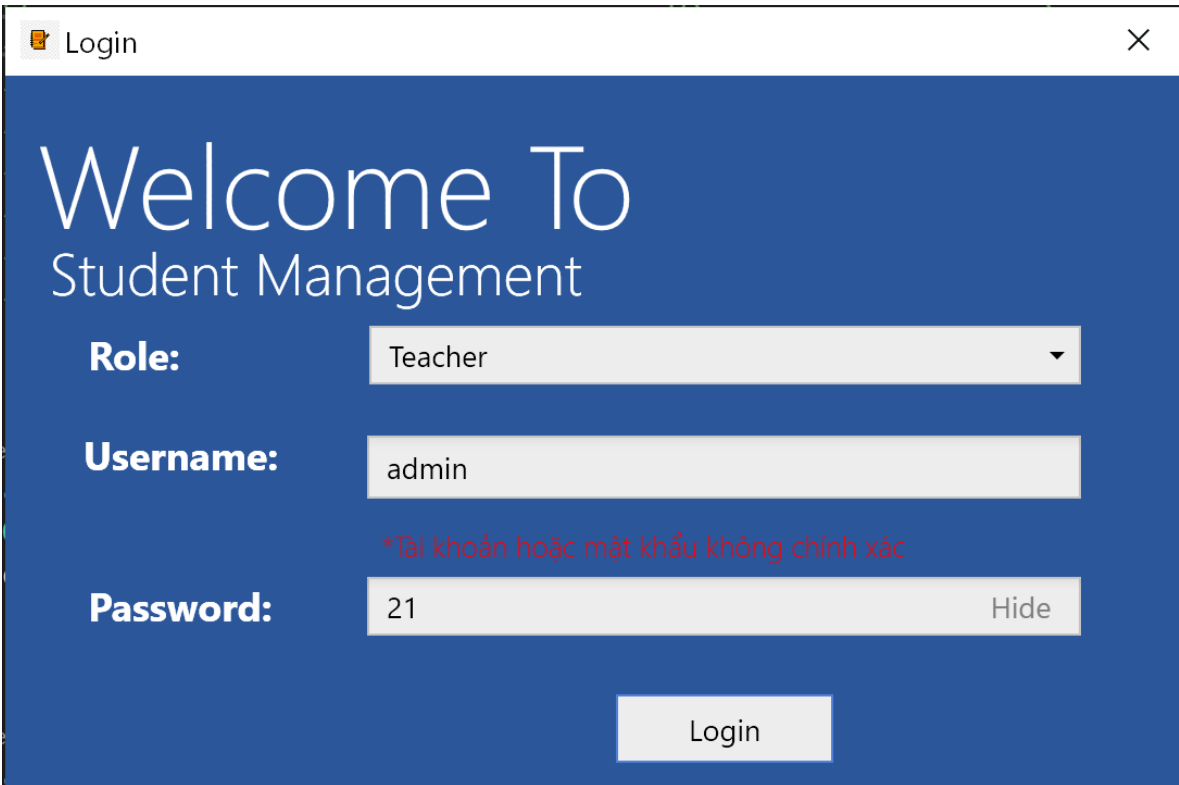
checkout (quay lại phiên bản nào đó hoạt động được), giúp chúng ta sửa chữa sai lầm gây ra.

### 3.6. Thử nghiệm và triển khai hệ thống

#### 3.6.1. Kiểm tra tính năng đăng nhập

\* Về phía Giảng viên:

Giả sử tên đăng nhập là “admin” và mật khẩu là “1”, khi người dùng nhập mật khẩu không chính xác hay chọn role không chính xác với tài khoản được cấp thì ứng dụng sẽ thông báo lỗi như hình:



The screenshot shows a web application login interface. At the top, there's a title bar with a folder icon and the text 'Login', and a close button (X). The main content area has a blue background with the text 'Welcome To Student Management' in white. Below this, there are three input fields: 'Role:' with a dropdown menu showing 'Teacher', 'Username:' with the text 'admin', and 'Password:' with the text '21'. A red error message '\*Tài khoản hoặc mật khẩu không chính xác' is displayed below the username field. A 'Login' button is located at the bottom right of the form area.

Hoặc khi người dùng nhập thiếu trường dữ liệu thì ứng dụng cũng sẽ thông báo giúp cho người dùng có thể nắm bắt được thông tin:

Login

# Welcome To Student Management

**Role:** Teacher

**Username:** admin

\*Vui lòng nhập đầy đủ thông tin

**Password:** Hide

Login

Khi thực hiện đăng nhập thành công, ứng dụng sẽ có một thông báo cụ thể khi người dùng đăng nhập thành công và tiếp đó là màn hình chính của chức năng.

Các tính năng chính của chương trình: Thêm, sửa, xóa, cập nhật và xuất báo cáo. Có thông báo lỗi khi người dùng nhập thiếu trường thông tin, hoặc sai định dạng số điện

thoại, email (@st.phenikaa-uni.edu.vn)

ControlPanel

Quản lý thông tin sinh viên

Quản lý thông tin môn học

Quản lý điểm

Quản lý đăng kí

Thông báo

Đăng xuất

Tim kiếm:

| Mã sinh viên | Mã ngành | Họ tên       | Ngày sinh            | Giới tính | Số điện thoại | Email                      | Khoa   |
|--------------|----------|--------------|----------------------|-----------|---------------|----------------------------|--------|
| sv001        | CNTT     | Trần Văn A   | 1/1/2000 12:00:00 AM | Nam       | 0987654321    | sv1@example.com            | Khoa C |
| sv0010       | CNTT     | Phạm Thị D   | 1/4/2000 12:00:00 AM | Nam       | 0987654324    | sv4@st.phenikaa-uni.edu.vn | Khoa C |
| sv00111      | CNTT     | Hoàng Văn E  | 1/5/2000 12:00:00 AM | Nam       | 0987654325    | sv5@st.phenikaa-uni.edu.vn | Khoa C |
| sv002        | CNTT     | Nguyễn Thị B | 2/1/2000 12:00:00 AM | N?        | 0987654322    | sv2@example.com            | Khoa C |
| sv003        | CNTT     | Lê Văn C     | 3/1/2000 12:00:00 AM | Nam       | 0987654323    | sv3@example.com            | Khoa C |
| sv004        | CNTT     | Phạm Thị D   | 4/1/2000 12:00:00 AM | N?        | 0987654324    | sv4@example.com            | Khoa C |
| sv005        | CNTT     | Hoàng Văn E  | 5/1/2000 12:00:00 AM | Nam       | 0987654325    | sv5@example.com            | Khoa C |

Mã sinh viên:

Tên sinh viên:

Số điện thoại:

Ngày sinh:

Email:

Mã ngành:

Khoá:

Giới tính:

Add

Update

Delete

Clear

Export

Những chức năng quản lý khác cũng được xử lý lỗi tương tự những vấn đề phát sinh do người dùng.

\* Về phía sinh viên:

Trang đăng nhập chung với đăng nhập của giảng viên, sinh viên cần chọn đúng role của mình để có thể đăng nhập vào hệ thống. Mọi thông báo lỗi đều giống như của giảng viên.

Chức năng của sinh viên chỉ có thể xem điểm học tập cá nhân và tạo báo cáo cho điểm học tập của bản thân mình.

## Kết luận

Trong quá trình phát triển, nhóm đã thực hiện được các bước như phân tích, thiết kế hệ thống và triển khai ứng dụng. Ứng dụng được công nghệ quản lý phiên bản với Git và sử dụng gần như thành thạo công nghệ ADO.NET để kết nối và tương tác với cơ sở dữ liệu. Hiểu được một số design pattern như Single-ton, MVC, MVVM. Sử dụng thành thạo các thành phần giao diện cơ bản cùng với kỹ thuật binding.

So sánh với yêu cầu đề tài và đánh giá mức độ hoàn thành:

Theo yêu cầu đề tài đặt ra, nhóm đã hoàn thành các bước chính trong phát triển ứng dụng như phân tích, thiết kế hệ thống, triển khai và xử lý dữ liệu.

Mức độ hoàn thành: Tuỳ thuộc vào cách triển khai cụ thể và yêu cầu, mức độ hoàn thành có thể khác nhau. Tuy nhiên, nhóm đã thực hiện được các bước chính và có một ứng dụng quản lý sinh viên cơ bản

Hướng phát triển và mở rộng sản phẩm:

Tăng cường tính năng: Chúng ta có thể mở rộng sản phẩm bằng cách thêm các tính năng như tạo lịch học, quản lý điểm danh, tạo báo cáo tổng kết, v.v

Giao diện người dùng tương tác tốt hơn: Nâng cấp giao diện người dùng để cung cấp trải nghiệm tốt hơn cho người dùng. Bạn có thể thêm chức năng tìm kiếm, sắp xếp, lọc dữ liệu, thêm các biểu đồ và biểu đồ thống kê để hiển thị thông tin một cách trực quan.

Quản lý lớp học và học phí: Mở rộng tính năng quản lý lớp học, gồm các thông tin như tên lớp, giảng viên, phòng học, thời khóa biểu, và học phí. Tạo khả năng quản lý việc thu, chi tiền học phí và tạo báo cáo liên quan.

Tích hợp hệ thống đăng ký môn học: Xây dựng tính năng đăng ký môn học cho sinh viên, bao gồm việc chọn môn học, xem tình trạng đăng ký, kiểm tra trùng lịch, và xác nhận đăng ký.

Bảo mật và phân quyền: Tăng cường bảo mật bằng cách xây dựng hệ thống phân quyền, giới hạn quyền truy cập dựa trên vai trò người dùng. Điều này sẽ giúp đảm bảo chỉ những người được ủy quyền mới có thể truy cập, chỉnh sửa và xóa thông tin quan trọng.

Tích hợp hệ thống thông báo: Tạo tính năng thông báo để gửi thông báo, tin nhắn hoặc email đến sinh viên và giảng viên. Điều này sẽ giúp cải thiện giao tiếp và tương tác giữa các thành viên trong hệ thống.