

HỌC VIỆN HÀNG KHÔNG VIỆT NAM  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO ĐỀ TÀI**  
**QUẢN LÝ KÝ TÚC XÁ**

HỌC KỲ 1 – NĂM HỌC: 2023 - 2024

MÔN HỌC: LẬP TRÌNH WINDOWS

MÃ LỚP HỌC PHẦN: 010100085301

*GVHD: Hồ Ngọc Thanh*

Nhóm sinh viên thực hiện: Nhóm 10

<i>Họ tên</i>	<i>Mã sinh viên</i>
1. Lê Cao Tấn Lộc	2154810086
2. Nguyễn Thị Thúy Hà	2154810061
3. Trịnh Vinh Qui	2154810110
4. Phan Tường Bảo Trâm	2154810077

TPHCM, tháng 11 năm 2023

## This image shows a full page of primary-ruled paper. It features multiple sets of horizontal dashed lines spaced evenly down the page, providing a guide for handwriting practice. The background is white, and there are no margins or other markings present.

**Chữ ký của giảng viên**

## **LỜI CẢM ƠN**

Xin chân thành cảm ơn Học viện Hàng không Việt Nam và khoa Công nghệ thông tin đã đưa môn Lập trình Windows cũng như tạo điều kiện và hỗ trợ suốt quá trình học tập của chúng em. Đồng thời nhóm em cũng không quên gửi lời cảm ơn đến thầy Hồ Ngọc Thanh đã đồng hành và tận tình dẫn dắt, truyền dạy những kiến thức bổ ích, những kinh nghiệm quý báu cho chúng em trong quá trình nghiên cứu đề tài này; đây chắc chắn là những kiến thức quan trọng, là hành trang để nhóm em có thể vững bước sau này. Nhờ những kiến thức bổ ích đó cùng với sự dẫn dắt và chỉ bảo tận tình của thầy đã giúp đề tài của nhóm hoàn thành thuận lợi. Đồng thời, cảm ơn những thành viên trong nhóm đã nhiệt tình trao đổi, đóng góp ý kiến giúp cho bài báo cáo hoàn thành đúng thời gian quy định. Tuy đã có nhiều cố gắng nhưng do kiến thức khá sâu rộng và thời gian nghiên cứu có hạn nên đề tài của nhóm em sẽ không tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý của các thầy để chủ đề này có điều kiện hoàn thiện hơn. Sau cùng, nhóm em kính chúc thầy luôn dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ sau.

## MỤC LỤC

CHƯƠNG I. GIỚI THIỆU .....	1
1. Mở đầu.....	1
2. Lý do chọn đề tài .....	1
3. Ý nghĩa của đề tài .....	1
4. Phạm vi nghiên cứu .....	2
5. Cấu trúc của đề tài .....	2
CHƯƠNG II. CÔNG NGHỆ ỨNG DỤNG .....	3
1. Hệ quản trị cơ sở dữ liệu Microsoft SQL Server .....	3
2. Công cụ Microsoft Visual Studio .....	4
3. Entity Framework (EF) Core.....	4
4. LINQ .....	5
CHƯƠNG III. SẢN PHẨM ĐỒ ÁN.....	7
1. Thiết kế của các bảng .....	7
2. Sơ đồ liên kết của các bảng .....	8
3. Các thiết kế form .....	8
<b>3.1. Form đăng nhập:</b> .....	8
<b>3.2. Form chính (QLKTX):</b> .....	12
<b>3.3. Hệ thống quản lý:</b> .....	12
<b>3.4. Tìm kiếm:</b> .....	15
<b>3.5. Báo cáo thống kê:</b> .....	16
4. Các đoạn code chính yếu để minh họa về 3 form cơ bản .....	18
5. Ưu điểm và hạn chế.....	26
CHƯƠNG IV. KẾT LUẬN.....	27
TÀI LIỆU THAM KHẢO .....	28

# CHƯƠNG I. GIỚI THIỆU

## 1. Mở đầu

Trong thời đại ngày nay, công nghệ thông tin là một công cụ kỹ thuật hiện đại phát triển mạnh mẽ và việc ứng dụng vào các lĩnh vực như kinh tế, giáo dục, thương mại... là không thể không nhắc đến. Nhận thấy được sức mạnh to lớn này, cùng với những lợi ích mà tin học hóa mang lại cho đời sống, góp phần thành công và hội nhập không ngừng của con người và công việc. Trong thời kỳ bùng nổ của công nghệ thông tin, việc áp dụng kiến thức về công nghệ thông tin nói chung và môn lập trình Windows nói riêng là một việc cần thiết và hữu ích để nâng cao khả năng tối ưu và tiện lợi của nó. Sự hiệu quả càng thể hiện rõ ràng hơn khi áp dụng những kiến thức đã học của bộ môn Lập trình Windows vào việc quản lý ký túc xá – một đề tài thiết thực và là môi trường để nhóm chúng em có cơ hội một lần nữa được ôn tập và vận dụng kiến thức của môn học. Ứng dụng quản lý ký túc xá của nhằm đơn giản hóa và tối ưu hóa quá trình quản lý cơ sở lưu trú, từ việc đăng ký, thanh toán cho đến theo dõi thông tin sinh viên và trạng thái phòng. Nhóm chúng em hướng đến việc cung cấp một nền tảng linh hoạt, dễ sử dụng và bảo mật để hỗ trợ các cơ sở ký túc xá trong việc nâng cao trải nghiệm cư trú của sinh viên và quản lý hiệu quả nguồn lực.

## 2. Lý do chọn đề tài

Trong môi trường đô thị ngày càng phát triển, quản lý ký túc xá trở thành một thách thức quan trọng, đặc biệt là với sự gia tăng nhanh chóng của sinh viên như Việt Nam hiện nay. Để đáp ứng nhu cầu ngày càng tăng về chỗ ở, nếu không ứng dụng công nghệ thông tin, việc quản lý ký túc xá sinh viên của trường sẽ rất khó khăn. Vì vậy, để thực hiện quản lý chặt chẽ và hiệu quả ký túc xá sinh viên, cần xây dựng hệ thống công nghệ thông tin để có thể đáp ứng nhu cầu tiện lợi, nhanh chóng và an toàn.

Ngoài ra đề tài còn giúp nhóm em có cơ hội được củng cố kiến thức đã học, hiểu rõ hơn về quá trình phát triển, nâng cao kỹ năng lập trình và sáng tạo, áp dụng kiến thức đã học vào thực tiễn và được trải nghiệm cách quản lý một mô hình.

Chính vì lý do đó, chúng em chọn đề tài: “Quản lý ký túc xá sinh viên” cho bài báo cáo cuối kì bộ môn Lập trình Windows.

## 3. Ý nghĩa của đề tài

Đề tài "Quản lý ký túc xá sinh viên" mang lại nhiều ý nghĩa quan trọng, đặc biệt là khi sự phát triển đô thị và nhu cầu ngày càng cao về chỗ ở.

- Tối Ưu Hóa Quản Lý: Hệ thống quản lý giúp tối ưu hóa quá trình quản lý kí túc xá bằng cách tự động hóa nhiều công việc, từ đăng ký, thanh toán cho đến theo dõi thông tin sinh viên và trạng thái phòng. Điều này giúp giảm thiểu công sức và thời gian quản trị, tăng cường hiệu suất công việc.
- Tiện Lợi Cho Sinh Viên: Sinh viên có thể trải nghiệm quy trình đăng ký và thanh toán một cách thuận tiện từ xa, giúp họ tiết kiệm thời gian. Điều này mang lại trải nghiệm người dùng tích cực và nâng cao sự hài lòng của sinh viên đối với dịch vụ cung cấp.
- Hiện Đại Hóa Cơ Sở Lưu Trú: Sự áp dụng công nghệ thông tin vào quản lý kí túc xá là một bước quan trọng trong việc hiện đại hóa cơ sở lưu trú. Điều này giúp các cơ sở này cải thiện hình ảnh, thu hút đối tượng người ở và nâng cao độ cạnh tranh trên thị trường.

#### **4. Phạm vi nghiên cứu**

Đề tài tập trung vào việc xây dựng Hệ thống quản lý ký túc xá dành cho sinh viên. Đồng thời tìm hiểu các môi trường và công nghệ được vận dụng vào trong bài tiểu luận.

#### **5. Cấu trúc của đề tài**

Phần tiếp theo của đề tài sẽ giới thiệu về lý thuyết và công nghệ liên quan đến lập trình. Sau đó, chúng em sẽ trình bày quá trình thiết kế và triển khai đề tài, cùng với các thử nghiệm và đánh giá hiệu suất. Cuối cùng, chúng em sẽ tổng kết thành quả và trình bày những hướng phát triển tiếp theo.

Chúng em rất mong rằng đề tài này sẽ mang lại những kiến thức bổ ích và trải nghiệm thú vị, đồng thời đóng góp vào việc hiểu rõ hơn về lĩnh vực lập trình và ứng dụng CNTT trong nhiều lĩnh vực khác.

## CHƯƠNG II. CÔNG NGHỆ ỨNG DỤNG

### 1. Hệ quản trị cơ sở dữ liệu Microsoft SQL Server

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Là một máy chủ cơ sở dữ liệu, nó là một sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo tiêu chuẩn RDBMS (Relational Database Management System) theo yêu cầu của các ứng dụng phần mềm khác. Có thể chạy trên cùng một máy tính hoặc trên một máy tính khác trên mạng (bao gồm cả Internet).

Microsoft SQL Server được xây dựng một cách tối ưu để có thể hoạt động trên hệ cơ sở dữ liệu rất lớn, lên đến hàng Terabyte. Microsoft SQL Server cung cấp cho người dùng đầy đủ các công cụ, phục vụ việc quản lý dữ liệu từ giao diện GUI đến ngôn ngữ truy vấn SQL. Điểm mạnh của Microsoft SQL Server là có thể kết hợp được với nhiều nền tảng phổ biến như ASP.NET, C# để xây dựng Winform hoặc là nó cũng có thể hoạt động độc lập.

#### - Chức năng chính:

- Quản lý dữ liệu: Microsoft SQL Server giúp lưu trữ, truy xuất và quản lý dữ liệu trong một môi trường an toàn và hiệu quả.
- Quản lý bảo mật: Cung cấp các tính năng bảo mật như xác thực và phân quyền để đảm bảo an toàn thông tin.

#### - Ngôn ngữ truy vấn:

Microsoft SQL Server sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để thực hiện các thao tác truy vấn dữ liệu.

#### - Hiệu suất và tối ưu hóa:

Microsoft SQL Server cung cấp các công cụ để tối ưu hóa hiệu suất cơ sở dữ liệu, bao gồm các chỉ mục, thống kê và quy tắc.

#### - Dịch vụ tích hợp:

Tích hợp với nhiều dịch vụ khác nhau như Reporting Services, Analysis Services, Integration Services để hỗ trợ việc báo cáo, phân tích và tích hợp dữ liệu.

#### - Hỗ trợ đa nền tảng:

Microsoft SQL Server có phiên bản chính thức cho nền tảng Windows và phiên bản Linux, mở rộng khả năng triển khai trên nhiều hệ điều hành.

## **2. Công cụ Microsoft Visual Studio**

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã" nhiều người sử dụng nhất thế giới", được dùng để lập trình C++ và C# là chính. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

- **Chức năng chính:**

Visual Studio cung cấp một môi trường phát triển tích hợp cho việc xây dựng, kiểm thử, và triển khai ứng dụng trên nền tảng Windows và nền tảng web.

- **Ngôn ngữ lập trình:**

Hỗ trợ nhiều ngôn ngữ lập trình như C#, Visual Basic, F#, C++, Python, JavaScript, và nhiều ngôn ngữ khác.

- **Phát triển ứng dụng đa nền tảng:**

Visual Studio hỗ trợ phát triển ứng dụng cho nhiều nền tảng như Windows, Web, iOS, Android và Linux.

- **Thiết kế giao diện:**

Cung cấp các công cụ thiết kế giao diện đồ họa cho ứng dụng Windows và ứng dụng web.

- **Tích hợp Git:**

Visual Studio tích hợp sẵn với Git để quản lý mã nguồn và phiên bản.

## **3. Entity Framework (EF) Core**

Entity Framework (EF) Core là ORM (Object-Relational Mapper) Framework để truy cập dữ liệu trong .NET Core. Nó được phát hành cùng với .NET Core và là Phiên bản mở rộng, nhẹ, mã nguồn mở và đa nền tảng của công nghệ truy cập dữ liệu Entity Framework. Nó hoạt động trên nhiều hệ điều hành như Windows, Mac và Linux.

- **ORM (Object-Relational Mapping):**

EF Core giúp ánh xạ các đối tượng trong mã nguồn .NET vào cơ sở dữ liệu, và ngược lại, mà không cần phải viết các truy vấn SQL trực tiếp.

- **Cross-platform và Open Source:**



EF Core là một framework đa nền tảng, có thể chạy trên nền tảng Windows, Linux và macOS. Nó cũng là mã nguồn mở, cho phép cộng đồng đóng góp và sửa lỗi.

- **Hỗ trợ nhiều loại cơ sở dữ liệu:**

EF Core hỗ trợ nhiều loại cơ sở dữ liệu như SQL Server, MySQL, PostgreSQL, SQLite và nhiều loại cơ sở dữ liệu khác.

- **Code First và Database First:**

Hỗ trợ cả phương pháp Code First (tạo các đối tượng trước, sau đó tạo cơ sở dữ liệu từ chúng) và Database First (bắt đầu từ cơ sở dữ liệu và tạo các đối tượng từ đó).

- **Linq to Entities:**

Sử dụng Language Integrated Query (LINQ) để thực hiện các truy vấn dữ liệu dễ đọc và dễ hiểu, được tích hợp chặt chẽ với ngôn ngữ lập trình C#.

- **Migrations:**

EF Core cung cấp một cơ chế "migrations" để quản lý sự thay đổi cơ sở dữ liệu một cách dễ dàng khi mã nguồn của ứng dụng thay đổi.

#### 4. LINQ

LINQ (Language Integrated Query) là một tính năng của ngôn ngữ lập trình C# được Microsoft giới thiệu từ phiên bản .NET Framework 3.5 trở lên. LINQ cung cấp một cách tiện lợi và mạnh mẽ để thực hiện các truy vấn dữ liệu trong mã nguồn .NET, bao gồm cả các truy vấn trên các cấu trúc dữ liệu như mảng, danh sách, tập hợp, và cơ sở dữ liệu. Dưới đây là một số thông tin về LINQ:

- **Language Integrated Query (LINQ):** LINQ là một phần của C# và .NET, cho phép lập trình viên sử dụng cú pháp query để truy vấn dữ liệu từ các nguồn khác nhau.

- **Phong cách lập trình bậc cao:** LINQ hỗ trợ phong cách lập trình bậc cao, giúp giảm bớt số lượng mã nguồn và làm tăng tính rõ ràng và dễ đọc.

- **Thực hiện trên nhiều nguồn dữ liệu:** LINQ có thể được sử dụng để thực hiện truy vấn trên một loạt các nguồn dữ liệu, bao gồm cả mảng, danh sách, tập hợp, cơ sở dữ liệu và các nguồn dữ liệu khác.

- **LINQ to Objects:** Cho phép truy vấn trực tiếp trên các cấu trúc dữ liệu trong mã nguồn như mảng, danh sách, và tập hợp.

- **LINQ to SQL:** LINQ to SQL cho phép tương tác với cơ sở dữ liệu SQL Server bằng cách sử dụng LINQ.

- **LINQ to XML:** LINQ to XML giúp thực hiện các truy vấn trên tài liệu XML.

- **LINQ to Entities:** Liên quan đến Entity Framework, LINQ to Entities cho phép thực hiện các truy vấn trên các đối tượng được ánh xạ từ cơ sở dữ liệu

### CHƯƠNG III. SẢN PHẨM ĐỒ ÁN

#### 1. Thiết kế của các bảng

##### - Bảng Contracts

	Column Name	Data Type	Allow Nulls
▶🔑	ContractID	int	<input type="checkbox"/>
	StartDate	datetime2(7)	<input type="checkbox"/>
	EndDate	datetime2(7)	<input type="checkbox"/>
	StudentID	bigint	<input type="checkbox"/>
	RoomID	int	<input type="checkbox"/>
	Price	decimal(18, 2)	<input type="checkbox"/>

##### - Bảng Staffs

	Column Name	Data Type	Allow Nulls
▶🔑	StaffID	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input type="checkbox"/>
	Address	nvarchar(MAX)	<input type="checkbox"/>
	Phone	nvarchar(MAX)	<input type="checkbox"/>
	Username	nvarchar(MAX)	<input type="checkbox"/>
	Password	nvarchar(MAX)	<input type="checkbox"/>
	Role	int	<input type="checkbox"/>
			<input type="checkbox"/>

##### - Bảng Students

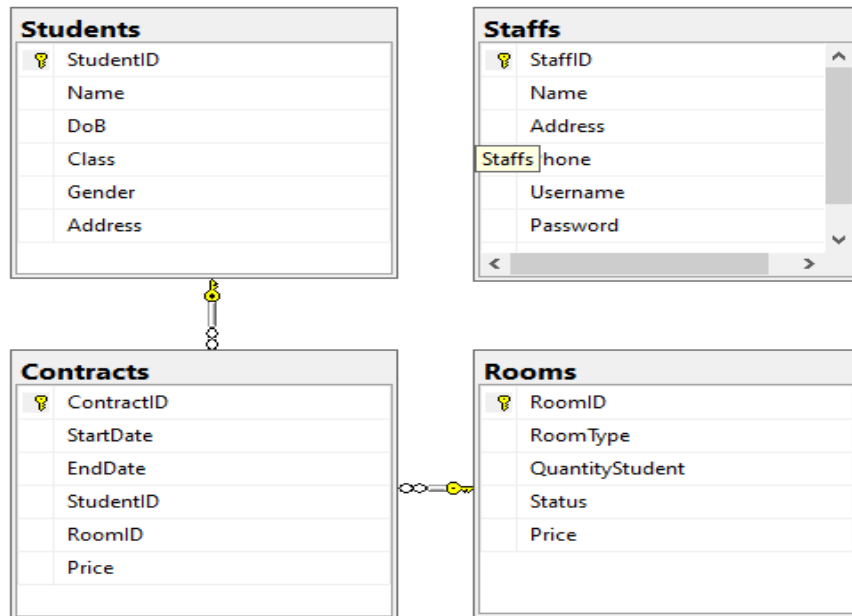
	Column Name	Data Type	Allow Nulls
▶🔑	StudentID	bigint	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input type="checkbox"/>
	DoB	datetime2(7)	<input type="checkbox"/>
	Class	nvarchar(MAX)	<input type="checkbox"/>
	Gender	nvarchar(MAX)	<input type="checkbox"/>
	Address	nvarchar(MAX)	<input type="checkbox"/>

##### - Bảng Rooms

	Column Name	Data Type	Allow Nulls
▶🔑	RoomID	int	<input type="checkbox"/>
	RoomType	nvarchar(MAX)	<input type="checkbox"/>
	QuantityStudent	int	<input type="checkbox"/>
	Status	nvarchar(MAX)	<input type="checkbox"/>
	Price	decimal(18, 2)	<input type="checkbox"/>

## 2. Sơ đồ liên kết của các bảng

Sơ đồ liên kết các bảng:



## 3. Các thiết kế form

### 3.1. Form đăng nhập:

Form được khởi tạo trong sự kiện load của form chính QLKTX, khi form chính load, form login sẽ hiển thị lên trước, nếu đăng nhập thành công, form chính sẽ nhận kết quả trả về người dùng đăng nhập từ form login, sau đó form login sẽ đóng lại và hiển thị form chính. Nếu đăng nhập không thành công, sẽ thông báo đăng nhập lại, nếu đóng form, cả form login và form chính sẽ đóng lại.





```

public partial class fLogin : Form
{
    public fLogin()
    {
        InitializeComponent();
    }
    private void btCancel_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
    private void btLogin_Click(object sender, EventArgs e)
    {
        if (txtName.Text == "")
        {
            toolTip1.Show("Bạn phải nhập tên người dùng?", txtName, 0, 0, 1000);
            txtName.Select();
        }
        else if (txtPassword.Text == "")
        {
            toolTip1.Show("Hãy nhập mật khẩu?", txtPassword, 0, 0, 1000);
            txtPassword.Select();
        }
        else
        {
            try
            {
                using (var db = new EFCore())
                {
                    Utility.staff = db.Staffs.SingleOrDefault(e => e.Username ==
txtName.Text && e.Password == txtPassword.Text);
                    if (Utility.staff != null) //Nếu đăng nhập thành công
                        DialogResult = DialogResult.OK; //đóng form
                    else
                        MessageBox.Show("Sai tên người dùng hoặc mật khẩu");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Lỗi"); ;
            }
        }
    }
}

```

### ***Trong đó:***

Hàm btCancel\_click sẽ thực hiện thoát khi nút cancel được click vào, hàm btLogin\_click sẽ thực hiện kiểm tra tài khoản và mật khẩu có để trống hoặc có tồn tại trong hệ thống không, nếu mọi bước kiểm tra được hoàn tất sẽ tiến hành sử dụng tài khoản mật khẩu để đăng nhập.

Nếu sai mật khẩu sẽ xuất hiện thông báo ra màn hình “Sai mật khẩu, vui lòng thử lại” cho đến khi mật khẩu được nhập đúng. Nếu cả tài khoản và mật khẩu được nhập đúng sẽ tiến hành đặt thuộc tính DialogResult là OK để form QLKTX kiểm tra và lấy tên của nhân viên đăng nhập và thông báo xin chào ở form QLKTX.

### **- Phân quyền**

Ta sẽ cấp quyền cho người dùng thông qua Role.

Role có các giá trị và quyền sau

1 – toàn quyền, bao gồm: xem/ xóa/ sửa/ nhập dữ liệu và xem báo cáo

2 – nhập liệu, tìm kiếm và xem báo cáo

3 – xem báo cáo

Giả sử hệ thống và phân quyền tương ứng như sau:

Role	Phân quyền	Các form có thể truy cập	Thêm, xóa, sửa, nhập dữ liệu	Xem báo cáo	Tài khoản & mật khẩu
1	Toàn quyền	Tất cả các form	Tất cả, sửa tổng tiền	Có	admin <u>admin</u>
2	Một phần	fNewRoom fNewStaff fNewStudent	Nhập dữ liệu, tìm kiếm, xem dữ liệu	Có	
3	Xem báo cáo			Có	

#### - Code phân quyền

```
fLogin f = new fLogin();

if (f.ShowDialog() != DialogResult.OK)
    Application.Exit();
else
{
    f.Close();
    if (Utility.staff != null)
    {
        // Chào người dùng đăng nhập
        hello.Text += " " + Utility.staff.Name;
        // PHÂN QUYỀN USER THEO ROLE
        // LEVEL 1
        if (Utility.staff.Role == 1)
        {
            // Toàn quyền xem/xóa/sửa/nhập dữ liệu
        }
        // LEVEL 2
        if (Utility.staff.Role == 2)
        {
            //Nhập liệu
            // Không sửa xóa
            dataGridView1.ReadOnly = true;
            dataGridView1.CellContentClick -= dataGridView1_CellContentClick;
            dataGridView2.ReadOnly = true;
            dataGridView2.CellContentClick -= dataGridView2_CellContentClick;
            dataGridView4.ReadOnly = true;
            dataGridView4.CellContentClick -= dataGridView4_CellContentClick;

            dataGridView3.ReadOnly = true;
            dataGridView3.CellContentClick -= dataGridView3_CellContentClick;
        }
        // LEVEL 3
        if (Utility.staff.Role == 3)
        {
            // Báo cáo
        }
    }
}
```

```

        tabControl1.Visible = false;
        tabControl2.Visible = false;
    }
}
}

```

### 3.2. Form chính (QLKTX):

Form chính là form quản lý và là form điều hướng tất cả các form khác. Trong form chính có 4 menu chính về hệ thống quản lý, tìm kiếm, báo cáo và hỗ trợ.

StudentID	Name	Birthday	Class	Gender	Address	Sửa	Xóa
2154810000	Nguyễn Văn Đạt	12/05/2003	21ĐHTT01	Nam	Hà Tĩnh		
2154810003	Trần Thị Hương	09/15/2002	21ĐHTT02	Nữ	Quảng Bình		
2154810004	Phạm Văn An	07/03/2003	22ĐHTT01	Nam	Hải Phòng		
2154810005	Lê Thị Lan	12/20/2001	22ĐHTT02	Nữ	Hà Nội		
2154810006	Hoàng Văn Bình	04/08/2002	22ĐHTT03	Nam	Đà Nẵng		
2154810007	Nguyễn Thị Thu	02/11/2003	22ĐHTT04	Nữ	Quảng Ngãi		
2154810008	Trần Văn Tuấn	06/29/2002	22ĐHTT05	Nam	Hồ Chí Minh		
2154810009	Lê Thị Mai	10/17/2003	22ĐHTT06	Nữ	Hải Dương		
2154810010	Nguyễn Văn Hoàng	03/05/2002	22ĐHTT01	Nam	Nghệ An		
2154810011	Phạm Thị Ngọc	11/23/2001	23ĐHTT01	Nữ	Hà Nam		
2154810012	Trần Văn Quang	07/14/2003	23ĐHTT01	Nam	Bắc Ninh		
2154810013	Đỗ Thị Hà	09/01/2002	23ĐHTT02	Nữ	Bình Dương		
2154810014	Nguyễn Văn Hoa	05/18/2003	23ĐHTT03	Nam	Quảng Nam		
2154810015	Lê Thị Ngân	08/26/2002	23ĐHTT04	Nữ	Thừa Thiên Huế		
2154810016	Phan Văn Đức	12/09/2001	23ĐHTT05	Nam	Cần Thơ		
2154810017	Bùi Thị Phương	01/04/2003	23ĐHTT06	Nữ	Kiên Giang		

### 3.3. Hệ thống quản lý:

Bao gồm các tabcontrol như sinh viên, phòng, hợp đồng, nhân viên để hiển thị thông tin về danh sách các bảng trên. Ngoài chức năng hiển thị, hệ thống quản lý còn có khả năng thêm mới và sửa thông qua các form mới (4 form trong đó 2 form thêm mới và 2 form sửa). Trong đó, form hợp đồng có thể thêm xóa sửa ngay trong form chính mà không cần phải mở thêm bất kỳ form nào khác.

RoomID	RoomType	Quantity	Status	Price	Sửa	Xóa
1	Nam	6	Còn trống	500000.00		
2	Nam	6	Còn trống	500000.00		
3	Nam	8	Còn trống	400000.00		
4	Nam	8	Còn trống	400000.00		
5	Nữ	6	Còn trống	500000.00		
6	Nữ	6	Còn trống	500000.00		
7	Nữ	8	Còn trống	400000.00		
8	Nữ	8	Còn trống	400000.00		
11	Nam	4	Còn trống	800000.00		
12	Nữ	10	Trống	300000.00		
13	Nam	6	Còn trống	500000.00		



**QUẢN LÝ KÍ TÚC XÁ** Xin chào Lê Cao Tấn Lộc

Hệ thống quản lý

Sinh viên Phòng Hợp đồng Nhân viên

**QUẢN LÝ HỢP ĐỒNG**

Mã hợp đồng  Ngày bắt đầu 11/20/2023 01:34:36

Mã SV  Ngày kết thúc 11/05/2024 12:00:00

Mã phòng

	ContractID	StartDate	EndDate	StudentID	RoomID	Price	Sửa	Xóa
▶	1146	11/17/2024 19:31	11/05/2025	2154810000	2	500000.00		
	1143	07/30/2023 09:00	08/30/2024 09:00	2154810017	5	500000.00		
	1142	07/30/2023 09:00	08/30/2024 09:00	2154810015	5	500000.00		
	1141	07/30/2023 09:00	08/30/2024 09:00	2154810013	8	400000.00		
	1140	07/30/2023 09:00	08/30/2024 09:00	2154810011	7	400000.00		
	1139	07/30/2023 09:00	08/30/2024 09:00	2154810009	6	500000.00		
	1138	07/30/2023 09:00	08/30/2024 09:00	2154810007	5	500000.00		
	1137	07/30/2023 09:00	08/30/2024 09:00	2154810005	5	500000.00		

**QUẢN LÝ KÍ TÚC XÁ** Xin chào Lê Cao Tấn Lộc

Hệ thống quản lý

Sinh viên Phòng Hợp đồng Nhân viên

**QUẢN LÝ NHÂN VIÊN**

	Sửa	Xóa	StaffID	Name	Address	Phone
▶			1	Lê Cao Tấn Lộc	Hà Nội	0123456781
			2	Nguyễn Thị Thủy Hà	Hồ Chí Minh	0123456782
			3	Trịnh Vĩnh Quý	Hải Phòng	0123456783
			4	Phan Tường Bảo Trâm	Bình Dương	0123456784
			5	Đặng Hoàng Mai Uyên	Bình Phước	0123456785

- Một số form thực hiện chức năng thêm, sửa cho các bảng:

+) **Form thêm sinh viên (fNewStudent):** Form này sử dụng để thêm mới sinh viên dựa trên các trường dữ liệu, tuy chưa có sự kiện validating cho các trường dữ liệu nhưng ở mỗi trường dữ liệu đều có kiểm tra trước khi lưu và thêm dữ liệu vào cơ sở dữ liệu.

Thêm mới sinh viên

**THÊM SINH VIÊN MỚI**

Tên sinh viên

Ngày sinh 11/18/2023  Giới tính Nam ☐ Nữ ☐

Lớp

Quê quán

+) **Form sửa sinh viên (fEditStudent):** Ở form sửa sinh viên, các trường dữ liệu đã được validating trước khi lưu dữ liệu được lưu vào cơ sở dữ liệu, khi click vào nút sửa của mỗi hàng dữ liệu, dữ liệu của hàng đó sẽ được tải lên form sửa.

**+) Form thêm phòng (fNewRoom):** Ở form này, các trường dữ liệu cũng đã được kiểm tra lỗi đầy đủ trước khi thêm phòng vào cơ sở dữ liệu, các tùy chọn của form được sử dụng combobox đã tải dữ liệu từ database lên.

**+) Form sửa phòng (fEditRoom):** Form sửa phòng cũng tương tự như form thêm phòng, các trường dữ liệu được tải lên khi nhấn sửa, các trường này cũng được kiểm tra dữ liệu trước khi lưu và có cài đặt sự kiện validating cho các trường dữ liệu đầy đủ, dữ liệu được tải lên combobox để tùy chọn dễ dàng hơn.

### 3.4. Tìm kiếm:

- Form tìm kiếm sinh viên được tìm kiếm dựa trên mã sinh viên

QLKT

QUẢN LÝ KÍ TÚC XÁ Xin chào Lê Cao Tấn Lộc

Hệ thống quản lí

Tìm kiếm

Báo cáo

Trợ giúp

Sinh viên Hợp đồng

Tìm kiếm hợp đồng

Mã SV

Tìm

ContractID	StartDate	EndDate	StudentID	RoomID	Price
1128	07/30/2023 09:00	08/30/2024 09:00	2154810000	1	500000.00
1129	07/30/2023 09:00	08/30/2024 09:00	2154810004	2	500000.00
1130	07/30/2023 09:00	08/30/2024 09:00	2154810006	3	400000.00
1131	07/30/2023 09:00	08/30/2024 09:00	2154810008	4	400000.00
1132	07/30/2023 09:00	08/30/2024 09:00	2154810010	1	500000.00
1133	07/30/2023 09:00	08/30/2024 09:00	2154810012	2	500000.00
1134	07/30/2023 09:00	08/30/2024 09:00	2154810014	3	400000.00
1135	07/30/2023 09:00	08/30/2024 09:00	2154810016	4	400000.00
1136	07/30/2023 09:00	08/30/2024 09:00	2154810003	5	500000.00
1137	07/30/2023 09:00	08/30/2024 09:00	2154810005	5	500000.00
1138	07/30/2023 09:00	08/30/2024 09:00	2154810007	5	500000.00
1139	07/30/2023 09:00	08/30/2024 09:00	2154810009	6	500000.00
1140	07/30/2023 09:00	08/30/2024 09:00	2154810011	7	400000.00
1141	07/30/2023 09:00	08/30/2024 09:00	2154810013	8	400000.00
1142	07/30/2023 09:00	08/30/2024 09:00	2154810015	5	500000.00
1143	07/30/2023 09:00	08/30/2024 09:00	2154810017	5	500000.00
1146	11/17/2024 19:31	11/05/2025	2154810000	2	500000.00

- Form hợp đồng được tìm kiếm dựa trên mã hợp đồng

QLKT

QUẢN LÝ KÍ TÚC XÁ Xin chào Lê Cao Tấn Lộc

Hệ thống quản lí

Tìm kiếm

Báo cáo

Trợ giúp

Sinh viên Hợp đồng

Tìm kiếm hợp đồng

2154810000

Tìm

ContractID	StartDate	EndDate	StudentID	RoomID	Price
1128	07/30/2023 09:00	08/30/2024 09:00	2154810000	1	500000.00
1146	11/17/2024 19:31	11/05/2025	2154810000	2	500000.00

3.5. Báo cáo thống kê:

- Report dạng bảng đơn giản

QLKTX

QUẢN LÝ KÍ TÚC XÁ

Xin chào Lê Cao Tấn Lộc

Hệ thống quản lí

Tìm kiếm

Báo cáo

Trợ giúp

Sinh viên x Hợp đồng Phòng Hợp đồng Phiếu thu tiền

1 of 1 100% Find Next

18/11/2023 02:35:56

Mã phòng	Ngày bắt đầu	Ngày kết thúc	Mã số sinh viên	Số phòng	Giá phòng
1128	30/07/2023	30/08/2024	2154810000	1	500000
1129	30/07/2023	30/08/2024	2154810004	2	500000
1130	30/07/2023	30/08/2024	2154810006	3	400000
1131	30/07/2023	30/08/2024	2154810008	4	400000
1132	30/07/2023	30/08/2024	2154810010	1	500000
1133	30/07/2023	30/08/2024	2154810012	2	500000
1134	30/07/2023	30/08/2024	2154810014	3	400000
1135	30/07/2023	30/08/2024	2154810016	4	400000
1136	30/07/2023	30/08/2024	2154810003	5	500000
1137	30/07/2023	30/08/2024	2154810005	5	500000
1138	30/07/2023	30/08/2024	2154810007	5	500000
1139	30/07/2023	30/08/2024	2154810009	6	500000
1140	30/07/2023	30/08/2024	2154810011	7	400000
1141	30/07/2023	30/08/2024	2154810013	8	400000
1142	30/07/2023	30/08/2024	2154810015	5	500000
1143	30/07/2023	30/08/2024	2154810017	5	500000
1146	17/11/2024	05/11/2025	2154810000	2	500000

- Report dạng bảng nhóm

QLKTX

QUẢN LÝ KÍ TÚC XÁ

Xin chào Lê Cao Tấn Lộc

Hệ thống quản lí

Tìm kiếm

Báo cáo

Trợ giúp

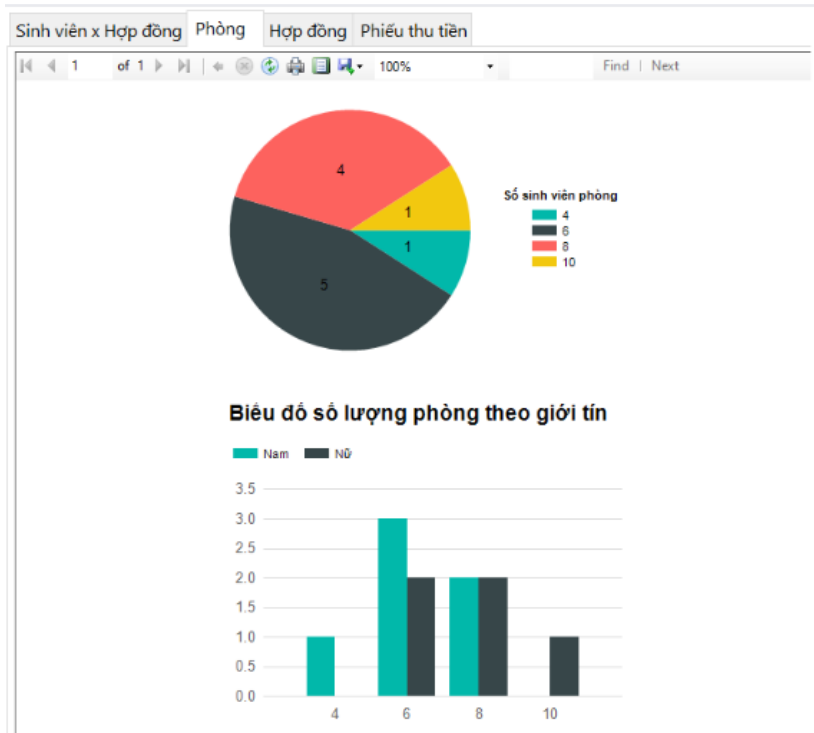
Sinh viên x Hợp đồng Phòng Hợp đồng Phiếu thu tiền

1 of 2 100% Find Next

18/11/2023 02:35:54

MSSV	Tên sinh viên	Giới tính	Phòng	Giá phòng
2154810000	Nguyễn Văn Đạt	Nam	1	500000.00
2154810004	Phạm Văn An	Nam	2	500000.00
2154810006	Hoàng Văn Bình	Nam	3	400000.00
2154810008	Trần Văn Tuấn	Nam	4	400000.00
2154810010	Nguyễn Văn Hoàng	Nam	1	500000.00
2154810012	Trần Văn Quang	Nam	2	500000.00
2154810014	Nguyễn Văn Hoa	Nam	3	400000.00
2154810016	Phan Văn Đức	Nam	4	400000.00
2154810003	Trần Thị Hương	Nữ	5	500000.00
2154810005	Lê Thị Lan	Nữ	5	500000.00
2154810007	Nguyễn Thị Thu	Nữ	5	500000.00
2154810009	Lê Thị Mai	Nữ	6	500000.00
2154810011	Phạm Thị Ngọc	Nữ	7	400000.00
2154810013	Đỗ Thị Hà	Nữ	8	400000.00
2154810015	Lê Thị Ngân	Nữ	5	500000.00

- Report dạng đồ thị



- Report dạng phiếu

QLKTX

**QUẢN LÝ KÍ TÚC XÁ** Xin chào Lê Cao Tấn Lộc

Hệ thống quản lý

Sinh viên x Hợp đồng Phòng Hợp đồng Phiếu thu tiền

2154810000 In phiếu

Tìm kiếm

Sinh viên x Hợp đồng Phòng Hợp đồng Phiếu thu tiền

HỌC VIỆN HÀNG KHÔNG VIỆT NAM  
KÝ TÚC XÁ SINH VIÊN HỌC VIỆN

Thời gian: 01:32:13 18/11/2023

**PHIẾU TÍNH TIỀN**

Tên sinh viên: Nguyễn Văn Đạt MSSV: 4309620000

Lớp: Giới tính: Nam

Số phòng	Loại phòng	Price
1	Nam	500000.00
2	Nam	500000.00

Cảm ơn bạn đã sử dụng dịch vụ của chúng tôi!

## 4. Các đoạn code chính yếu để minh họa về 3 form cơ bản

### Form quản lý chính QLKTX – sinh viên

```
public partial class QLKTX : Form
{
    public QLKTX()
    {
        InitializeComponent();
    }
    private void QLKTX_Load(object sender, EventArgs e)
    {
        using (var db = new EFCore())
        {
            dataGridView1.DataSource = db.Students.Select(x => new { x.StudentID,
            x.Name, x.DoB, x.Class, x.Gender, x.Address }).ToList();
            dataGridView6.DataSource = db.Students.Select(x => new { x.StudentID,
            x.Name, x.DoB, x.Class, x.Gender, x.Address }).ToList();
            dataGridView2.DataSource = db.Rooms.Select(x => new { x.RoomID,
            x.RoomType, x.QuantityStudent, x.Status, x.Price }).ToList();
            dataGridView3.DataSource = db.Contracts.Select(x => new {
            x.ContractID, x.StartDate, x.EndDate, x.StudentID, x.RoomID, x.Price }).ToList();
            dataGridView9.DataSource = db.Contracts.Select(x => new {
            x.ContractID, x.StartDate, x.EndDate, x.StudentID, x.RoomID, x.Price }).ToList();
            dataGridView4.DataSource = db.Staffs.Select(x => new { x.StaffID,
            x.Name, x.Address, x.Phone }).ToList();
        }
        // Login
        fLogin f = new fLogin();
        if (f.ShowDialog() != DialogResult.OK)
            Application.Exit();
    else
    {
        f.Close();
        if (Utility.staff != null)
        {
            // Chào người dùng đăng nhập
            hello.Text += " " + Utility.staff.Name;
            // PHÂN QUYỀN USER THEO ROLE
            // LEVEL 1
            if (Utility.staff.Role == 1)
            {
                // Toàn quyền xem/xóa/sửa/nhập dữ liệu
            }

            // LEVEL 2
            if (Utility.staff.Role == 2)
            {
                //Nhập liệu
                // Không sửa xóa
                dataGridView1.ReadOnly = true;
                dataGridView1.CellContentClick -= dataGridView1_CellContentClick;
                dataGridView2.ReadOnly = true;
                dataGridView2.CellContentClick -= dataGridView2_CellContentClick;
                dataGridView4.ReadOnly = true;
                dataGridView4.CellContentClick -= dataGridView4_CellContentClick;
                dataGridView3.ReadOnly = true;
                dataGridView3.CellContentClick -= dataGridView3_CellContentClick;
            }
            // LEVEL 3
            if (Utility.staff.Role == 3)
            {
                // Báo cáo
            }
        }
    }
}
```

```

        tabControl1.Visible = false;
        tabControl2.Visible = false;
    }
}

}

private void QLKTX_Activated(object sender, EventArgs e)
{
    using (var db = new EFCore())
    {
        dataGridView1.DataSource = db.Students.Select(x => new { x.StudentID,
x.Name, x.DoB, x.Class, x.Gender, x.Address }).ToList();
        dataGridView2.DataSource = db.Rooms.Select(x => new { x.RoomID, x.RoomType,
x.QuantityStudent, x.Status, x.Price }).ToList();
        dataGridView3.DataSource = db.Contracts.OrderByDescending(x =>
x.ContractID).Select(x => new { x.ContractID, x.StartDate, x.EndDate, x.StudentID,
x.RoomID, x.Price }).ToList();
        dataGridView4.DataSource = db.Staffs.Select(x => new { x.StaffID, x.Name,
x.Address, x.Phone }).ToList();
    }
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView1.Columns[e.ColumnIndex].Name.Equals("Xóa"))
    {
        try
        {
            long StudentID =
Convert.ToInt64(dataGridView1.Rows[e.RowIndex].Cells["StudentID"].Value);
            using (var db = new EFCore())
            {
                Student student =
db.Students.Single(c => c.StudentID == StudentID);
                if (MessageBox.Show("Bạn muốn xóa khách hàng " + student.Name,
"Xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
== DialogResult.Yes)
                {
                    db.Students.Remove(student);
                    Contract contract = db.Contracts.Single(c => c.StudentID ==
StudentID);
                    db.Contracts.Remove(contract);
                    db.SaveChanges();
                    QLKTX_Activated(sender, e);
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi, chưa xóa được? Error: " + ex.Message);
        }
    }
    if (dataGridView1.Columns[e.ColumnIndex].Name.Equals("Sửa"))
    {
        try
        {
            if (Utility.IsOpeningForm("fEditStudent")) ;
            fEditStudent f = new
fEditStudent((long)dataGridView1.Rows[e.RowIndex].Cells["StudentID"].Value);
            f.Show();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi, chưa sửa được? Error: " + ex.Message);
        }
    }
}

```

```

    }
}
private void dataGridView2_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView2.Columns[e.ColumnIndex].Name.Equals("Delete"))
    {
        try
        {
            int RoomID =
Convert.ToInt32(dataGridView2.Rows[e.RowIndex].Cells["RoomID"].Value);
            using (var db = new EFCore())
            {
                Room room =
db.Rooms.Single(c => c.RoomID == RoomID);
                if (MessageBox.Show("Bạn muốn xóa phòng " + room.RoomID,
"Xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
== DialogResult.Yes)
                {
                    db.Rooms.Remove(room);
                    db.SaveChanges();
                    QLKTX_Activated(sender, e);
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi, chưa xóa được? Error: " + ex.Message);
        }
    }
    if (dataGridView2.Columns[e.ColumnIndex].Name.Equals("Edit"))
    {
        try
        {
            if (Utility.IsOpeningForm("fEditRoom")) ;
            fEditRoom f = new
fEditRoom((int)dataGridView2.Rows[e.RowIndex].Cells["RoomID"].Value);
            f.Show();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi, chưa sửa được? Error: " + ex.Message);
        }
    }
}
private void dataGridView4_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView4.Columns[e.ColumnIndex].Name.Equals("Xoa"))
    {
        try
        {
            string staffID =
dataGridView4.Rows[e.RowIndex].Cells["MaNhanVien"].Value.ToString();
            if (MessageBox.Show("Bạn muốn xóa nhân viên " +
dataGridView4.Rows[e.RowIndex].Cells["TenNhanVien"].Value,
"Xóa",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                if (true)
                {
                    MessageBox.Show("Xóa dữ liệu thành công?");
                    QLKTX_Activated(sender, e);
                }
            }
        }
    }
}

```



```

        }
    }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Lỗi, chưa xóa được? Error: " + ex.Message);
    }
}
if (dataGridView4.Columns[e.ColumnIndex].Name.Equals("Sua"))
{
    try
    {
        if (Utility.IsOpeningForm("fEditStaff")) ;
        fEditStaff f = new
fEditStaff((string)dataGridView4.Rows[e.RowIndex].Cells["MaNhanVien"].Value);
        f.Show();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Lỗi, chưa sửa được? Error: " + ex.Message);
    }
}

}

private void dataGridView3_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView3.Columns[e.ColumnIndex].Name.Equals("contractRemove"))
    {
        try
        {
            int ContractID =
Convert.ToInt32(dataGridView3.Rows[e.RowIndex].Cells["ID"].Value);
            using (var db = new EFCore())
            {
                Contract contract =
                db.Contracts.Single(c => c.ContractID == ContractID);
                if (MessageBox.Show("Bạn muốn xóa hợp đồng " + contract.ContractID,
                "Xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
                == DialogResult.Yes)
                {
                    db.Contracts.Remove(contract);
                    db.SaveChanges();
                    QLKTX_Activated(sender, e);
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi, chưa xóa được? Error: " + ex.Message);
        }
    }
    if (dataGridView3.Columns[e.ColumnIndex].Name.Equals("contractEdit"))
    {
        addContract.Enabled = false;
        try
        {
            int ContractID =
Convert.ToInt32(dataGridView3.Rows[e.RowIndex].Cells["ID"].Value);
            using (var db = new EFCore())
            {
                Contract contract =
                db.Contracts.Single(c => c.ContractID == ContractID);
                contractID.Text = contract.ContractID.ToString();
                contractStudentID.Text = contract.StudentID.ToString();
            }
        }
    }
}

```

```

        contractRoomID.Text = contract.RoomID.ToString();
        dateTimePickerStart.Text = contract.StartDate.ToString();
        dateTimePickerEnd.Text = contract.EndDate.ToString();
    }
}
catch (Exception ex)
{
    MessageBox.Show("Lỗi, chưa sửa được? Error: " + ex.Message);
}
}

private void addContract_Click(object sender, EventArgs e)
{
    if (string.IsNullOrWhiteSpace(contractRoomID.Text))
    {
        tooltip1.Show("Hãy nhập mã phòng?", contractRoomID, 0, 0, 1000);
        return;
    }
    if (string.IsNullOrWhiteSpace(contractStudentID.Text))
    {
        tooltip1.Show("Hãy nhập mã sinh viên?", contractStudentID, 0, 0, 1000);
        return;
    }
    // DateTimePickerStart | DateTimePickerEnd
    // Truy vấn dữ liệu
    using (var db = new EFCore())
    {
        long studentID = long.Parse(contractStudentID.Text);
        int roomID = int.Parse(contractRoomID.Text);
        Student student = db.Students.Single(s => s.StudentID == studentID);
        Room room = db.Rooms.Single(r => r.RoomID == roomID);
        // nam không ở phòng nữ và ngược lại
        if (!student.Gender.Equals(room.RoomType)) MessageBox.Show("Sinh viên " +
student.Gender + " không thể ở phòng " + room.RoomType);
        bool StudentExists = db.Contracts.Any(c => c.StudentID == studentID);
        // hợp đồng có tồn tại
        if (StudentExists)
        {
            Contract latestContract = db.Contracts.Where(c => c.StudentID ==
studentID && c.EndDate < dateTimePickerStart.Value)
                .OrderByDescending(c => c.EndDate)
                .FirstOrDefault();
            if (latestContract == null)
            {
                MessageBox.Show("Ngày kết thúc của hợp đồng cũ phải < hơn ngày bắt
đầu của hợp đồng mới");
            }
            else if (dateTimePickerStart.Value > dateTimePickerEnd.Value)
            {
                MessageBox.Show("Ngày bắt đầu của hợp đồng phải bé hơn ngày kết
thúc");
            }
            else
            {
                // Tiếp tục với quá trình lưu thông tin hợp đồng mới
                Contract contract = new Contract();
                contract.StudentID = long.Parse(contractStudentID.Text);
                contract.RoomID = int.Parse(contractRoomID.Text);
                contract.StartDate = dateTimePickerStart.Value;
                contract.EndDate = dateTimePickerEnd.Value;
                contract.Price = room.Price;
                db.Contracts.Add(contract);
            }
        }
    }
}

```

```

        db.SaveChanges();
        QLKTX_Activated(sender, e);
        toolTip1.Show("Thêm hợp đồng thành công!", addContract);
    }
}

}

private void saveContract_Click(object sender, EventArgs e)
{
    if (string.IsNullOrWhiteSpace(contractRoomID.Text))
    {
        toolTip1.Show("Hãy nhập mã phòng?", contractRoomID, 0, 0, 1000);
        return;
    }
    if (string.IsNullOrWhiteSpace(contractStudentID.Text))
    {
        toolTip1.Show("Hãy nhập mã sinh viên?", contractStudentID, 0, 0, 1000);
        return;
    }
    if (dateTimePickerStart.Value < DateTime.Now)
    {
        MessageBox.Show("Ngày không hợp lệ. Vui lòng chọn một ngày trong tương
lai.");
        dateTimePickerStart.Value = DateTime.Now;
        return;
    }

    using (EFCore db = new EFCore())
    {
        try
        {
            long studentID = long.Parse(contractStudentID.Text);
            int rooomID = int.Parse(contractRoomID.Text);
            Student student = db.Students.Single(s => s.StudentID == studentID);
            Room room = db.Rooms.Single(r => r.RoomID == rooomID);
            // nam không ở phòng nữ và ngược lại
            if (!student.Gender.Equals(room.RoomType))
            {
                MessageBox.Show("Sinh viên " + student.Gender + " không thể ở phòng
" + room.RoomType);
                return;
            }
            // Lấy hợp đồng cần cập nhật từ cơ sở dữ liệu
            Contract contractToUpdate = db.Contracts.Single(c => c.ContractID ==
int.Parse(contractID.Text));

            // Kiểm tra xem hợp đồng có tồn tại không
            if (contractToUpdate != null)
            {
                // Cập nhật thông tin hợp đồng
                contractToUpdate.StartDate = dateTimePickerStart.Value;
                contractToUpdate.EndDate = dateTimePickerEnd.Value;
                contractToUpdate.StudentID = studentID;
                contractToUpdate.RoomID = rooomID;

                // Lưu thông tin vào cơ sở dữ liệu
                db.SaveChanges();

                toolTip1.Show("Cập nhật hợp đồng thành công!", saveContract, 0, 0,
1000);
                QLKTX_Activated(sender, e);
            }
            else

```

```

        {
            MessageBox.Show("Không tìm thấy hợp đồng để cập nhật.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Đã xảy ra lỗi khi cập nhật hợp đồng: " + ex.Message);
    }
}

}

Form nhập mới – Sinh viên:
public partial class fNewStudent : Form
{
    Student student;
    public fNewStudent()
    {
        InitializeComponent();
    }
    private void btClose_Click(object sender, EventArgs e)
    {
        Close();
    }

    private void fNewStudent_FormClosing(object sender, FormClosingEventArgs e)
    {
        e.Cancel = false;
    }

    private void btSave_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(studentName.Text))
        {
            toolTip1.Show("Hãy nhập tên sinh viên phòng?", studentName, 0, 0,
1000);
            return;
        }
        if (string.IsNullOrEmpty(studentDoB.Text))
        {
            toolTip1.Show("Hãy nhập ngày tháng năm sinh?", studentDoB, 0, 0, 1000);
            return;
        }
        if (string.IsNullOrEmpty(studentClass.Text))
        {
            toolTip1.Show("Hãy nhập lớp học?", studentClass, 0, 0, 1000);
            return;
        }
        if (string.IsNullOrEmpty(studentAddress.Text))
        {
            toolTip1.Show("Hãy nhập địa chỉ?", studentAddress, 0, 0, 1000);
            return;
        }
        try
        {
            student = new Student();
            student.Name = studentName.Text;
            student.DoB = studentDoB.Value.Date;
            student.Class = studentClass.Text;
            student.Gender = femaleCheck.Checked ? "Nam" : "Nữ";
            student.Address = studentAddress.Text;
            using (var db = new EFCore())
            {
                db.Students.Add(student);
                db.SaveChanges();
            }
        }
    }
}

```

```

    }
    studentName.ResetText();
    studentDoB.ResetText();
    studentClass.ResetText();
    studentAddress.ResetText();
    femaleCheck.CheckState = CheckState.Unchecked;
    maleCheck.CheckState = CheckState.Unchecked;
    toolTip1.Show("Lưu thành công!", btSave, 0, 0, 1000);
}
catch (Exception ex)
{
    toolTip1.Show("Lưu thất bại? Error: " + ex.Message, btSave, 0, 0,
1000);
}
studentName.Focus();
}
}
}

```

### Form sửa – Sinh viên:

```

public partial class fEditStudent : Form
{
    long ID;
    Student student;
    EFCore db = new EFCore();
    public fEditStudent(long ID)
    {
        InitializeComponent();
        this.ID = ID;
    }
    private void fEditStudent_Load(object sender, EventArgs e)
    {
        student = db.Students.Single(p => p.StudentID == ID);
        Text += " - Mã SV " + student.StudentID.ToString();
        studentName.Text = student.Name;
        studentAddress.Text = student.Address;
        studentClass.Text = student.Class;
        studentDoB.Text = student.DoB.ToString();
        if (student.Gender.Equals("Nam")) maleCheck.Checked = true;
        else femaleCheck.Checked = true;
    }

    private void fEditStudent_FormClosing(object sender, FormClosingEventArgs e)
    {
        e.Cancel = false;
    }
    private void btSave_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrWhiteSpace(studentName.Text))
        {
            toolTip1.Show("Hãy nhập tên sinh viên phòng?", studentName, 0, 0,
1000);
            return;
        }
        if (string.IsNullOrWhiteSpace(studentDoB.Text))
        {
            toolTip1.Show("Hãy nhập ngày tháng năm sinh?", studentDoB, 0, 0, 1000);
            return;
        }
        if (string.IsNullOrWhiteSpace(studentClass.Text))
        {
            toolTip1.Show("Hãy nhập lớp học?", studentClass, 0, 0, 1000);
            return;
        }
    }
}

```

```

if (string.IsNullOrEmpty(studentAddress.Text))
{
    toolTip1.Show("Hãy nhập địa chỉ?", studentAddress, 0, 0, 1000);
    return;
}
try
{
    student.Name = studentName.Text;
    student.DoB = studentDoB.Value.Date;
    student.Class = studentClass.Text;
    if ((femaleCheck.CheckState == CheckState.Checked &&
maleCheck.CheckState == CheckState.Checked) || (femaleCheck.CheckState ==
CheckState.Unchecked && maleCheck.CheckState == CheckState.Unchecked))
student.Gender = "Khác";
    else if (femaleCheck.CheckState == CheckState.Checked) student.Gender
= "Nữ";
    else if (maleCheck.CheckState == CheckState.Checked) student.Gender =
"Nam";
    student.Address = studentAddress.Text;
    db.SaveChanges();
    toolTip1.Show("Lưu thành công.", btSave, 0, 0, 1000);
}
catch (Exception ex)
{
    toolTip1.Show("Lưu thất bại? Error: " + ex.Message, btSave, 0, 0,
1000);
}
studentName.Focus();
}
private void btClose_Click_1(object sender, EventArgs e)
{
    Close();
}
}

```

## 5. Ưu điểm và hạn chế

- **Ưu điểm:** Trong đồ án của chúng em có những ưu điểm như sau
  - Giao diện đẹp mắt, dễ sử dụng thao tác, dễ nhìn thấy.
  - Truy vấn nhanh, upload dữ liệu trực tiếp.
  - Truy vấn tìm kiếm linh hoạt, tạo bảng đa dạng.
  - Các hàm và các báo cáo đã được viết sẵn, dễ dàng sử dụng lại bằng việc điền các tham số mà không phải mất công cài đặt.
  - Phân quyền theo vai trò nhân viên.
- **Nhược điểm:**
  - Chưa có validating cho tất cả các trường dữ liệu.
  - Chưa sử dụng linh hoạt các hàm trong báo cáo để tạo ra nhiều trường dữ liệu quan trọng và cần thiết.

## **CHƯƠNG IV. KẾT LUẬN**

Đồ án quản lý ký túc xá của chúng em không chỉ là một bài thực hành mà còn là một dự án thực tế có thể giải quyết các vấn đề thực tế trong quản lý ký túc xá. Việc này không chỉ tiết kiệm công sức và thời gian mà còn giúp tối ưu hóa nguồn lực và tiền bạc.

Ứng dụng của đề tài đã góp phần vào việc giải quyết các thách thức trong quản lý ký túc xá thông qua việc ứng dụng kiến thức đã học. Điều này có thể bao gồm việc theo dõi thông tin sinh viên, quản lý phòng ở, cũng như tối ưu hóa quá trình đặt phòng và thanh toán.

Đồ án không chỉ là một bước học lý thuyết mà còn là một trải nghiệm thực tế, giúp chúng em hiểu rõ hơn về sự quan trọng của việc áp dụng kiến thức vào thực tế để giải quyết vấn đề. Chúng em hy vọng rằng đồ án của chúng ta sẽ đóng góp vào sự phát triển của quản lý ký túc xá và mang lại giá trị thực tế cho cộng đồng.

## TÀI LIỆU THAM KHẢO

- Wikipedia - Microsoft SQL Server

[https://vi.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://vi.wikipedia.org/wiki/Microsoft_SQL_Server)

- Wikipedia Microsoft Visual Studio

[https://vi.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://vi.wikipedia.org/wiki/Microsoft_Visual_Studio)

- Giao diện Guna

<https://www.nuget.org/packages/Guna.UI2.WinForms/>

- Bài giảng của thầy Hồ Ngọc Thanh

- Entity Framework Core tutorial

<https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

- Linq learning tutorial

<https://www.tutorialsteacher.com/linq>