**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

KHOA KINH TẾ VÀ QUẢN TRỊ KINH DOANH

**===o0o===**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MÔN: LẬP TRÌNH VỚI CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: PHÂN TÍCH NGHIỆP VỤ VÀ XÂY DỰNG PHẦN MỀM QUẢN LÝ CHI TIÊU**



**Giảng viên hướng dẫn: ThS. Mai Hà An**

**Sinh viên thực hiện: Bùi Văn Mạnh**

**Mã sinh viên: 1851070199**

**Lớp: K63 - HTTT**

**Hà Nội, 2021**

**MỤC LỤC**

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

LỜI CẢM ƠN

[CHƯƠNG I: TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU 1](#_Toc90859381)

[1.1. Lý do chọn đề tài 1](#_Toc90859382)

[1.2. Mục đích chọn đề tài 1](#_Toc90859383)

[1.3. Mục tiêu và phạm vi nghiên cứu 1](#_Toc90859384)

[1.3. Phương pháp nghiên cứu 1](#_Toc90859385)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 2](#_Toc90859386)

[2.1. Công nghệ C#.Net 2](#_Toc90859387)

[2.1.1. Nền tảng của .NET 2](#_Toc90859388)

[2.1.2. NET Framework 2](#_Toc90859389)

[2.1.3. Biên dịch và ngôn ngữ trung gian (MSIL) 3](#_Toc90859390)

[2.2. SQL Server 3](#_Toc90859391)

[CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 5](#_Toc90859392)

[3.1. Cơ sở dữ liệu 5](#_Toc90859393)

[3.2. Biểu đồ UC hệ thống 7](#_Toc90859394)

[3.3. Biểu đồ hoạt động 8](#_Toc90859395)

[3.4. Biểu đồ trình tự 11](#_Toc90859396)

[CHƯƠNG IV: LẬP TRÌNH VÀ KẾT QUẢ 14](#_Toc90859397)

[4.1. Các hàm chức năng 14](#_Toc90859398)

[4.1.1. RefreshNguonTien 14](#_Toc90859399)

[4.1.2. RefreshLichSuGanDay 15](#_Toc90859400)

[4.1.2. RefreshChart 16](#_Toc90859401)

[4.2. Kết quả lập trình 19](#_Toc90859402)

[4.2.1 Giao diện Tổng quan 19](#_Toc90859403)

[4.2.2. Giao diện lịch sử thu chi 20](#_Toc90859404)

[4.2.3. Giao diện thông kê báo cáo 21](#_Toc90859405)

[4.2.4. Giao diện thêm chi tiêu 22](#_Toc90859406)

[4.2.6. Giao diện thêm thu nhập 22](#_Toc90859407)

[4.2.6. Giao diện chuyển đổi nguồn tiền 23](#_Toc90859408)

[CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc90859409)

[5.1. Kết luận 24](#_Toc90859410)

[5.2. Hướng phát triển 24](#_Toc90859411)

TÀI LIỆU THAM KHẢO

**DANH MỤC BẢNG BIỂU**

[Bảng 1: Bảng khoản chi 5](#_Toc90859506)

[Bảng 2: Bảng khoản thu 5](#_Toc90859507)

[Bảng 3: Bảng lịch sử thu chi 5](#_Toc90859508)

[Bảng 4: Bảng Nguồn tiền 6](#_Toc90859509)

[Bảng 5: Bnarg Loại nhóm thu 6](#_Toc90859510)

[Bảng 6: Bảng loại nhóm chi 6](#_Toc90859511)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Kiến trúc khung ứng dụng .Net 2](file:///D:\K63HTTH\Subjects\LapTrinhCoSoDuLieu\BC\BC_LTCSDL_Edit_L4.docx#_Toc90859512)

[Hình 2: Biểu đồ UC Hệ thống 7](#_Toc90859513)

[Hình 3: Biểu đồ hoạt động thêm chi tiêu 8](#_Toc90859514)

[Hình 4: Biểu đồ hoạt động thêm thu nhập 9](#_Toc90859515)

[Hình 5: Biểu đồ hoạt động chuyển đổi nguồn tiền 10](#_Toc90859516)

[Hình 6: Biểu đồ trình tự thêm chi tiêu 11](#_Toc90859517)

[Hình 7: Biểu đồ trình tự thêm thu nhập 12](#_Toc90859518)

[Hình 8: Biểu đồ tình tự chuyển đổi nguồn tiền 13](#_Toc90859519)

[Hình 9: Giao diện tổng quan 19](#_Toc90859520)

[Hình 10: Giao diện lịch sử 20](#_Toc90859521)

[Hình 11: Giao diện thống kê báo cáo 21](#_Toc90859522)

[Hình 12: Giao diện thêm chi tiêu 22](#_Toc90859523)

[Hình 13: Giao diện thêm thu nhập 22](#_Toc90859524)

[Hình 14: Giao diện chuyển đổi nguồn tiền 23](#_Toc90859525)

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành tới các Thầy, Cô trong trường Đại học lâm nghiệp và thầy cô giáo tại bộ môn Tin Học. Đặc biệt, em xin gửi lời cảm ơn chân thành tới **Thầy** **Mai Hà An** – đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn nhóm trong suốt quá trình nghiên cứu và hoàn thành tốt môn học Lập trình với cơ sở dữ liệu trong đề tài: ***“Phân tích nghiệp vụ và xây dựng phần mềm quản lý chi tiêu”.***

Em xin chân thành cảm ơn!

*Hà Nội, ngày 11 tháng 12 năm 2021*

**Sinh viên thực hiện**

**Bùi Văn Mạnh**

# TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU

## 1.1. Lý do chọn đề tài

Việc ứng dụng công nghệ thông tin vào cuộc sống đã trở thành hoạt động không thể thiếu của con người trong thời kì công nghiệp hóa, hiện đại hóa. Hiện nay, việc quản lý chi tiêu của gia đình có thể được thực hiện một cách đơn giản, hiệu quả hơn với việc sử dụng các phần mềm tin học. Việc lưu trữ, quản lý bằng máy tính giúp giải phóng con người khỏi các công việc ghi chép và tính toán bằng giấy bút, giảm sự nhầm lẫn khi tính toán.

Xuất phát từ thực tế việc chi tiêu hàng ngày của bản thân và gia đình, em đã phân tích những vấn đề thường xảy ra trong quản lí chi tiêu và thấy được sự cần thiết của phần mềm quản lý chi tiêu để giúp quản lý tốt hơn chi tiêu của bản thân. Vì vậy, em đã lựa chọn đề tài:

***“Phân tích nghiệp vụ và xây dựng phần mềm quản lý chi tiêu”***

Để làm đồ án cuối kỳ cho môn học Lập trình với cơ sở dữ liệu do thầy **Mai Hà An** giảng dạy.

## 1.2. Mục đích chọn đề tài

* Rèn luyện kỹ năng phân tích và thiết kế hệ thống.
* Xây dựng hệ thống quản lý chi tiêu đơn giản hiệu quả.

## 1.3. Mục tiêu và phạm vi nghiên cứu

#### Mục tiêu

Ứng dụng các công nghệ đã được học xây dựng thành công hệ thống quản lý chi tiêu.

#### Phạm vi nghiên cứu

* Phạm vi công nghệ: Lưu trữ cơ sở dữ liệu trên phần mềm Microsoft SQL Server Management Studio và lập trình trên phần mềm Microsoft Visual Studio.
* Phạm vi con người: Cá nhân.

## 1.3. Phương pháp nghiên cứu

* Phương pháp kế thừa tài liệu
* Phương pháp chuyên gia

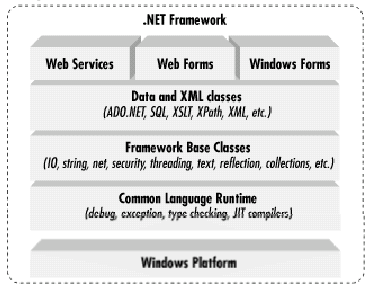
# CƠ SỞ LÝ THUYẾT

## Công nghệ C#.Net

### Nền tảng của .NET

Khi Microsoft công bố C# vào tháng 7 năm 2000, việc khánh thành nó chỉ là một phần trong số rất nhiều sự kiện mà nền tảng .Net được công công bố. Nền tảng .Net là bộ khung phát triển ứng dụng mới, nó cung cấp một giao diện lập trình ứng dụng (Application Programming Interface - API) mới mẽ cho các dịch vụ và hệ điều hành Windows, cụ thể là Windows 2000, nó cũng mang lại nhiều kỹ thuật khác nổi bật của Microsoft suốt từ những năm 90. Trong số đó có các dịch vụ COM+, công nghệ ASP, XML và thiết kế hướng đối tượng, hỗ trợ các giao thức dịch vụ web mới như SOAP, WSDL và UDDL với trọng tâm là Internet, tất cả được tích hợp trong kiến trúc DNA.

Nền tảng .NET bao gồm bốn nhóm sau:

* Một tập các ngôn ngữ, bao gồm C# và Visual Basic .Net; một tập các công cụ phát triển bao gồm Visual Studio .Net; một tập đầy đủ các thư viện phục vụ cho việc xây dựng các ứng dụng web, các dịch vụ web và các ứng dụng Windows; còn có CLR - Common Language Runtime: (ngôn ngữ thực thi dùng chung) để thực thi các đối tượng được xây dựng trên bô khung này.
* Một tập các Server Xí nghiệp .Net như SQL Server 2000. Exchange 2000, BizTalk 2000, … chúng cung cấp các chức năng cho việc lưu trữ dữ liệu quan hệ, thư điện tử, thương mại điện tử B2B, …
* Các dịch vụ web thương mại miễn phí, vừa được công bố gần đậy như là dự án Hailstorm; nhà phát triển có thể dùng các dịch vụ này để xây dựng các ứng dụng đòi hỏi tri thức về định danh người dùng…
* .NET cho các thiết bị không phải PC như điện thoại (cell phone), thiết bị game

### NET Framework

**.NET Framework** là một nền tảng lập trình và cũng là một nền tảng thực thi ứng dụng chủ yếu trên hệ điều hành [Microsoft Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows) được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft). Các chương trình được viết trên nền.NET Framework sẽ được triển khai trong môi trường [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) (ngược lại với *môi trường phần cứng*) được biết đến với tên [Common Language Runtime](https://vi.wikipedia.org/w/index.php?title=Common_Language_Runtime&action=edit&redlink=1) (CLR). Môi trường phần mềm này là một [máy ảo](https://vi.wikipedia.org/wiki/M%C3%A1y_%E1%BA%A3o) trong đó cung cấp các dịch vụ như [an ninh phần mềm](https://vi.wikipedia.org/w/index.php?title=An_ninh_ph%E1%BA%A7n_m%E1%BB%81m&action=edit&redlink=1) (*security*), [quản lý bộ nhớ](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B) (*memory management*), và các xử lý lỗi ngoại lệ (*exception handling*).

Hình 1: Kiến trúc khung ứng dụng .Net

**.NET framework** bao gồm tập các thư viện lập trình lớn, và những thư viện này hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; truy cập, kết nối [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u); [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web); các [giải thuật](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n), [cấu trúc dữ liệu](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u); giao tiếp mạng... CLR cùng với bộ thư viện này là 2 thành phần chính của.NET framework.

**.NET framework** đơn giản hóa việc viết ứng dụng bằng cách cung cấp nhiều thành phần được thiết kế sẵn, người lập trình chỉ cần học cách sử dụng và tùy theo sự sáng tạo mà gắn kết các thành phần đó lại với nhau. Nhiều công cụ được tạo ra để hỗ trợ xây dựng ứng dụng.NET, và IDE (*Integrated Developement Environment*) được phát triển và hỗ trợ bởi chính Microsoft là [Visual Studio](https://vi.wikipedia.org/wiki/Visual_Studio).

### Biên dịch và ngôn ngữ trung gian (MSIL)

**Ngôn ngữ trung gian chung**hoặc **Ngôn ngữ trung gian dùng chung** (**Common Intermediate Language - CLI**), là ngôn ngữ lập trình có thể đọc được của con người ở mức thấp nhất được xác định bởi đặc tả Cơ sở chung hạ tầng ngôn ngữ (CLI) và được [.NET Framework](https://vi.wikipedia.org/wiki/.NET_Framework) và [Mono](https://vi.wikipedia.org/wiki/Mono) sử dụng. Ngôn ngữ nhắm mục tiêu một môi trường thời gian chạy tương thích với CLI biên dịch tới CIL, được lắp ráp thành một [mã đối tượng](https://vi.wikipedia.org/wiki/M%C3%A3_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) có định dạng [bytecode](https://vi.wikipedia.org/wiki/Bytecode). CIL là một ngôn ngữ lắp ráp [hướng đối tượng](https://vi.wikipedia.org/wiki/H%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng), và hoàn toàn dựa trên stack. Bytecode của nó được dịch sang mã gốc hoặc - thường được thực hiện bởi một [máy ảo](https://vi.wikipedia.org/wiki/M%C3%A1y_%E1%BA%A3o).

**CIL** ban đầu được gọi là **Ngôn ngữ trung gian dùng chung của Microsoft (MSIL)** trong phiên bản beta của các ngôn ngữ .NET. Do tiêu chuẩn hoá C # và [cơ sở ngôn ngữ dùng chung](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_ng%C3%B4n_ng%E1%BB%AF_d%C3%B9ng_chung), bytecode được chính thức gọi là **CIL**.

## SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (**Transact-SQL)**để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

Một vài ấn bản SQL Server:

* **Enterprise** : chứa tất cả cá đặc điểm nổi bật của SQL Server, bao gồm nhân bộ máy cơ sở dữ liệu và các dịch vụ đi kèm cùng với các công cụ cho tạo và quản lý phân cụm SQL Server. Nó có thể quản lý các CSDL lớn tới 524 [petabytes](https://en.wikipedia.org/wiki/Petabyte) và đánh địa chỉ 12 [terabytes](https://en.wikipedia.org/wiki/Terabyte) bộ nhớ và hỗ trợ tới 640 bộ vi xử lý(các core của cpu)
* **Standard** : Rất thích hợp cho các công ty vừa và nhỏ vì giá thành rẻ hơn nhiều so với Enterprise Edition, nhưng lại bị giới hạn một số chức năng cao cấp (advanced features) khác, edition này có thể chạy tốt trên hệ thống lên đến 4 CPU và 2 GB RAM.
* **Developer** : Có đầy đủ các tính năng của Enterprise Edition nhưng được chế tạo đặc biệt như giới hạn số lượng người kết nối vào Server cùng một lúc…. Ðây là phiên bản sử dụng cho phát triển và kiểm tra ứng dụng. Phiên bản này phù hợp cho các cá nhân, tổ chức xây dựng và kiểm tra ứng dụng
* **Workgroup**: ấn bản SQL Server Workgroup bao gồm chức năng lõi cơ sở dữ liệu nhưng không có các dịch vụ đi kèm. Chú ý phiên bản này không còn tồn tại ở SQL Server 2012.
* **Express**: SQL Server Express dễ sử dụng và quản trị cơ sở dữ liệu đơn giản. Được tích hợp với Microsoft Visual Studio, nên dễ dàng để phát triển các ứng dụng dữ liệu, an toàn trong lưu trữ, và nhanh chóng triển khai. SQL Server Express là phiên bản miễn phí,  không giới hạn về số cơ ở dữ liệu hoặc người sử dụng, nhưng nó chỉ dùng cho 1 bộ vi xử lý với 1 GB bộ nhớ và 10 GB file cơ sở dữ liệu. SQL Server Express là lựa chọn tốt cho những người dùng chỉ cần một phiên bản SQL Server 2005 nhỏ gọn, dùng trên máy chủ có cấu hình thấp, những nhà phát triển ứng dụng không chuyên hay những người yêu thích xây dựng các ứng dụng nhỏ.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1. Cơ sở dữ liệu

* **Bảng Khoản chi (KhoanChi):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Khóa chính |
| IdNhomChi | Int | Khóa ngoại, Tên nhóm chi |
| IdNguonTien | Int | Khóa ngoại, Tên nguồn tiền |
| SoTien | Float | Số tiền |
| Date | Date | Ngày thêm khoản chi |
| Note | Nvarchar | Ghi chu |

Bảng 1: Bảng khoản chi

* **Bảng Khoản thu (KhoanThu):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Khóa chính |
| IdNhomThu | Int | Khóa ngoại, Tên nhóm thu |
| IdNguonTien | Int | Khóa ngoại, Tên nguồn tiền |
| SoTien | Float | Số tiền |
| Date | Date | Ngày thêm khoản thu |
| Note | Nvarchar | Ghi chú |

Bảng 2: Bảng khoản thu

* **Bảng Lịch sử thu chi(LichSu):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Khóa chính |
| LoaiChiTieu | Nvarchar | Loại chi tiêu |
| TenChiTieu | Nvarchar | Tên chi tiêu |
| SoTien | Float | Số tiền |
| Date | Nvarchar | Thời giản |
| Note | Int | Ghi chú |

Bảng 3: Bảng lịch sử thu chi

* **Bảng Nguồn tiền (NguonTien):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Khóa chính |
| TenNguonTien | Nvarchar | Tên nguồn tiền |
| SoTien | Float | Số tiền |

Bảng 4: Bảng Nguồn tiền

* **Bảng Loại Nhóm Thu (NhomThu):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Khóa chính |
| TenNhomThu | Nvarchar | Tên nhóm chi |

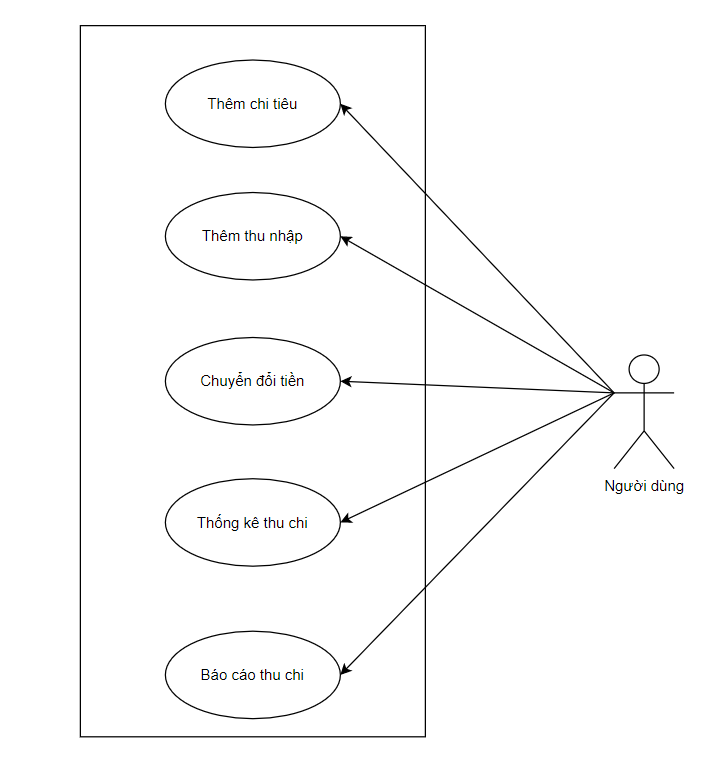
Bảng 5: Bnarg Loại nhóm thu

* **Bảng Loại Nhóm Chi (NguonChi):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Khóa chính |
| TenNhomChi | Nvarchar | Tên nhóm chi |

Bảng 6: Bảng loại nhóm chi

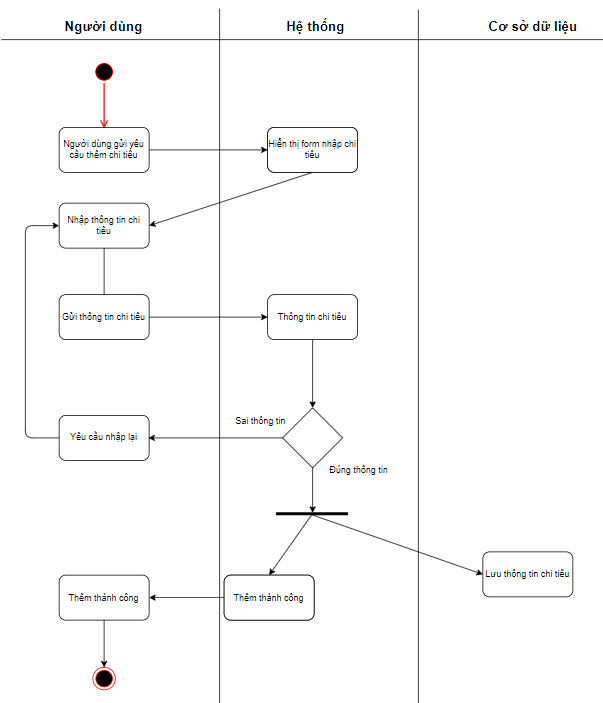
## 3.2. Biểu đồ UC hệ thống



Hình 2: Biểu đồ UC Hệ thống

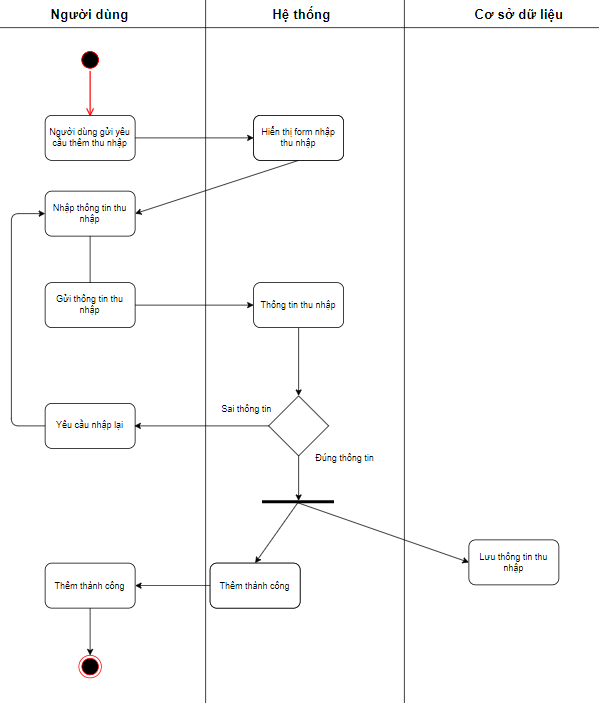
## 3.3. Biểu đồ hoạt động

* **Biểu đồ hoạt động thêm chi tiêu**



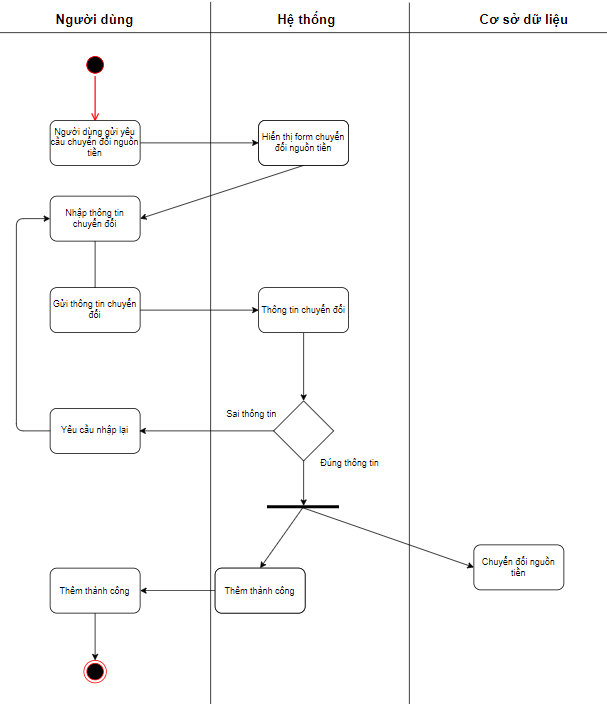
Hình 3: Biểu đồ hoạt động thêm chi tiêu

* **Biểu đồ hoạt động thêm thu nhập**



Hình 4: Biểu đồ hoạt động thêm thu nhập

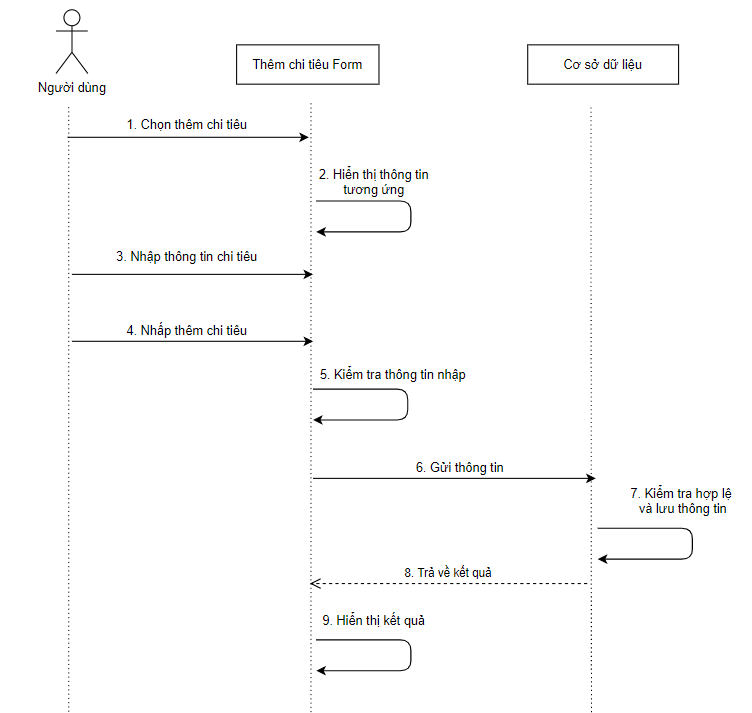
* **Biểu đồ hoạt động chuyển đổi nguồn tiền**



Hình 5: Biểu đồ hoạt động chuyển đổi nguồn tiền

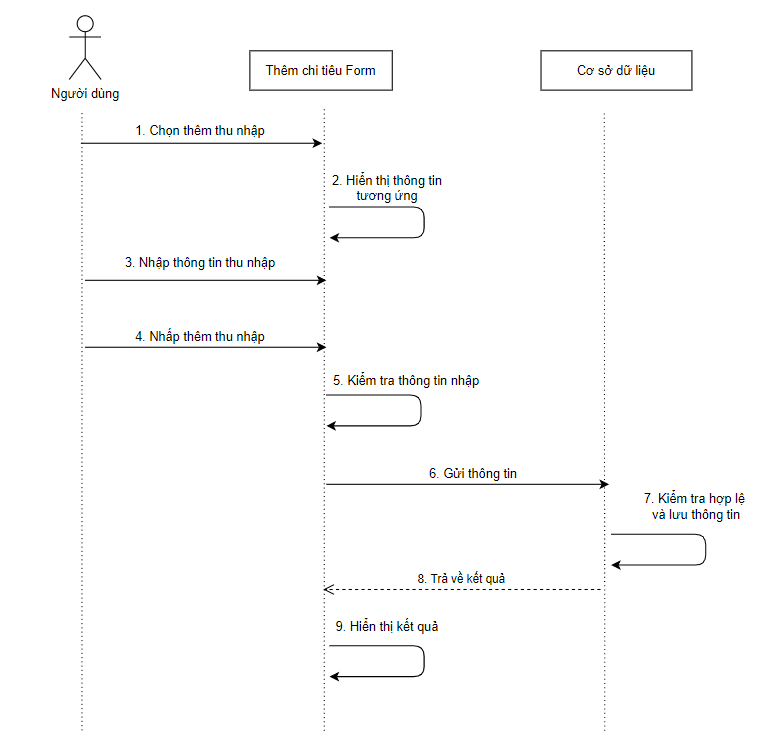
## 3.4. Biểu đồ trình tự

* **Biểu đồ trình tự thêm chi tiêu**



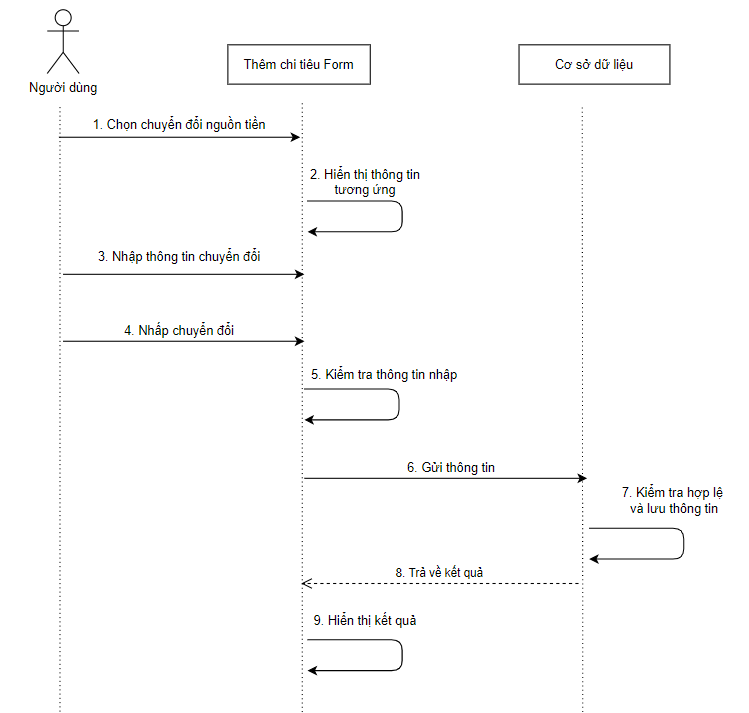
Hình 6: Biểu đồ trình tự thêm chi tiêu

* **Biểu đồ trình tự thêm thu nhập**



Hình 7: Biểu đồ trình tự thêm thu nhập

* **Biểu đồ trình tự chuyển đổi nguồn tiền**



Hình 8: Biểu đồ tình tự chuyển đổi nguồn tiền

# LẬP TRÌNH VÀ KẾT QUẢ

## 4.1. Các hàm chức năng

### 4.1.1. RefreshNguonTien

**Mô tả:** Hàm này sẽ được gọi để kiểm tra mỗi khi chi tiêu của người dùng thay đổi ảnh hưởng tới nguồn tiền, nguồn tiền trên hệ thống sẽ được cập nhật trực tiếp.

**Chức năng:** Cập nhật nguồn tiền mỗi khi thay đổi

public void RefreshNguonTIen()

{

List<Models.MNguonTien> ds = cNguonTien.SelectNguonTien();

if (ds != null && ds.Count > 0)

{

float tongtien = 0;

float formMoney = 0;

foreach (Models.MNguonTien m in ds)

{

if (m.Id == "1")

{

float currentMoney = float.Parse(m.SoTien);

if (checkisFloat(lbTienNganHang.Text.Split('đ')[0]))

{

formMoney = float.Parse(lbTienNganHang.Text.Split('đ')[0]);

}

if (currentMoney != formMoney)

{

lbTienNganHang.Text = currentMoney.ToString("N0", CultureInfo.InvariantCulture) + "đ";

RefreshLichSuGanDay();

}

}

else if (m.Id == "2")

{

float currentMoney = float.Parse(m.SoTien);

if (checkisFloat(lbTienMat.Text.Split('đ')[0]))

{

formMoney = float.Parse(lbTienMat.Text.Split('đ')[0]);

}

if (currentMoney != formMoney)

{

lbTienMat.Text = float.Parse(m.SoTien).ToString("N0", CultureInfo.InvariantCulture) + "đ";

RefreshLichSuGanDay();

}

}

tongtien += float.Parse(m.SoTien);

}

if (checkisFloat(lbTongTien.Text.Split('đ')[0]))

{

formMoney = float.Parse(lbTongTien.Text.Split('đ')[0]);

}

if (tongtien != formMoney)

{

lbTongTien.Text = tongtien.ToString("N0", CultureInfo.InvariantCulture) + "đ";

RefreshLichSuGanDay();

}

}

}

### 4.1.2. RefreshLichSuGanDay

**Mô tả:** Cập nhật những thay đổi của lịch sử chi tiêu người dùng mỗi khi người dùng thêm một chi tiêu, thu nhập, chuyển đổi nguồn tiền

**Chức năng:** Cập nhật dữ liệu

public void RefreshLichSuGanDay()

{

List<Models.MLichSu> ds = cLichSu.SelectLichSuGanDay();

if (ds != null && ds.Count > 0)

{

DataTable tb = new DataTable();

tb.Columns.Add("TT", typeof(int));

tb.Columns.Add("ID", typeof(string));

tb.Columns.Add("Date", typeof(string));

tb.Columns.Add("LoaiChiTieu", typeof(String));

tb.Columns.Add("TenChiTieu", typeof(String));

tb.Columns.Add("SoTien", typeof(String));

int tt = 0;

foreach (MLichSu m in ds)

{

tt++;

if (!(tt > 20))

{

DataRow r = tb.NewRow();

r["TT"] = tt;

r["ID"] = m.Id;

r["Date"] = DateToDDmmYYYY(m.Date.Split(' ')[0]);

r["LoaiChiTieu"] = m.LoaiChiTieu;

r["TenChiTieu"] = m.TenChiTieu;

r["SoTien"] = float.Parse(m.SoTien).ToString("N0", CultureInfo.InvariantCulture) + "đ";

tb.Rows.Add(r);

}

else

{

break;

}

}

dgv.DataSource = tb;

dgv.Columns["ID"].Visible = false;

dgv.Columns["TT"].Visible = false;

dgv.Columns[0].HeaderText = "STT";

dgv.Columns[2].HeaderText = "Ngày";

dgv.Columns[2].Width = 93;

dgv.Columns[3].HeaderText = "Nhóm";

dgv.Columns[3].Width = 117;

dgv.Columns[4].HeaderText = "Loại chi tiêu";

dgv.Columns[4].Width = 114;

dgv.Columns[5].HeaderText = "Số tiền";

}

else

dgv.DataSource = null;

}

### 4.1.2. RefreshChart

**Mô tả:** Những dữ liệu chi tiêu của người dùng khi bị thay đổi hoặc có thể là bị xóa, thêm, chuyển đổi sẽ ảnh hưởng tới dữ liệu của biểu đồ. Vậy nên hàm này thực hiện việc thay đổi cập nhật dữ liệu biểu đồ đúng với thực tế.

**Chức năng:** Cập nhật dữ liệu

public void RefreshChartByDate()

{

string datefr = DateChange(fr.Text);

string dateend = DateChange(to.Text);

float tongchi = cKhoanChi.SelectTongChiByDate(datefr, dateend);

float tongthu = cKhoanThu.SelectTongThuByDate(datefr, dateend);

// Tổng Thu Chi Pie

tongthuchi.Series.RemoveAt(0);

tongthuchi.Series.Add("TongThuChiPie");

tongthuchi.Series["TongThuChiPie"].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Pie;

//

tongthuchi.Series["TongThuChiPie"].Points.AddXY("Tổng thu", tongthu);

tongthuchi.Series["TongThuChiPie"].Points.AddXY("Tổng chi", tongchi);

tongthuchi.Series["TongThuChiPie"].IsValueShownAsLabel = true;

// Chi Tiêu Pie

chitieu.Series.RemoveAt(0);

chitieu.Series.Add("ChiTieuPie");

chitieu.Series["ChiTieuPie"].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Pie;

//

float chitieu1 = cKhoanChi.SelectTongChiByIdAndDate(1, datefr, dateend);

float chitieu2 = cKhoanChi.SelectTongChiByIdAndDate(2, datefr, dateend);

float chitieu3 = cKhoanChi.SelectTongChiByIdAndDate(3, datefr, dateend);

float chitieu4 = cKhoanChi.SelectTongChiByIdAndDate(4, datefr, dateend);

float chitieu5 = cKhoanChi.SelectTongChiByIdAndDate(5, datefr, dateend);

float chitieu6 = cKhoanChi.SelectTongChiByIdAndDate(6, datefr, dateend);

float chitieu7 = cKhoanChi.SelectTongChiByIdAndDate(7, datefr, dateend);

float chitieu8 = cKhoanChi.SelectTongChiByIdAndDate(8, datefr, dateend);

float chitieu9 = cKhoanChi.SelectTongChiByIdAndDate(9, datefr, dateend);

float chitieu10 = cKhoanChi.SelectTongChiByIdAndDate(10, datefr, dateend);

float chitieu11 = cKhoanChi.SelectTongChiByIdAndDate(11, datefr, dateend);

float chitieu12 = cKhoanChi.SelectTongChiByIdAndDate(12, datefr, dateend);

float chitieu13 = cKhoanChi.SelectTongChiByIdAndDate(13, datefr, dateend);

float chitieu14 = cKhoanChi.SelectTongChiByIdAndDate(14, datefr, dateend);

//

if (chitieu1 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Ăn uống", chitieu1);

}

if (chitieu2 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Sinh hoạt", chitieu2);

}

if (chitieu3 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Đi lại", chitieu3);

}

if (chitieu4 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Giải trí", chitieu4);

}

if (chitieu5 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Sức khỏe", chitieu5);

}

if (chitieu6 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Tích lũy", chitieu6);

}

if (chitieu7 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Nhà cửa", chitieu7);

}

if (chitieu8 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Con cái", chitieu8);

}

if (chitieu9 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Học tập", chitieu9);

}

if (chitieu10 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Xã giao", chitieu10);

}

if (chitieu11 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Kinh doanh", chitieu11);

}

if (chitieu12 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Cho vay", chitieu12);

}

if (chitieu13 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Trả nợ", chitieu13);

}

if (chitieu14 != 0)

{

chitieu.Series["ChiTieuPie"].Points.AddXY("Từ thiện", chitieu14);

}

chitieu.Series["ChiTieuPie"].IsValueShownAsLabel = true;

// Thu nhập Pie

thunhap.Series.RemoveAt(0);

thunhap.Series.Add("ThuNhapPie");

thunhap.Series["ThuNhapPie"].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Pie;

//

float thunhap1 = cKhoanThu.SelectTongThuByIdAndDate(1, datefr, dateend);

float thunhap2 = cKhoanThu.SelectTongThuByIdAndDate(2, datefr, dateend);

float thunhap3 = cKhoanThu.SelectTongThuByIdAndDate(3, datefr, dateend);

float thunhap4 = cKhoanThu.SelectTongThuByIdAndDate(4, datefr, dateend);

float thunhap5 = cKhoanThu.SelectTongThuByIdAndDate(5, datefr, dateend);

float thunhap6 = cKhoanThu.SelectTongThuByIdAndDate(6, datefr, dateend);

float thunhap7 = cKhoanThu.SelectTongThuByIdAndDate(7, datefr, dateend);

//

if (thunhap1 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Lương", thunhap1);

}

if (thunhap2 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Thưởng", thunhap2);

}

if (thunhap3 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Đi vay", thunhap3);

}

if (thunhap4 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Đầu tư", thunhap4);

}

if (thunhap5 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Kinh doanh", thunhap5);

}

if (thunhap6 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Thu hồi nợ", thunhap6);

}

if (thunhap7 != 0)

{

thunhap.Series["ThuNhapPie"].Points.AddXY("Mừng tặng", thunhap7);

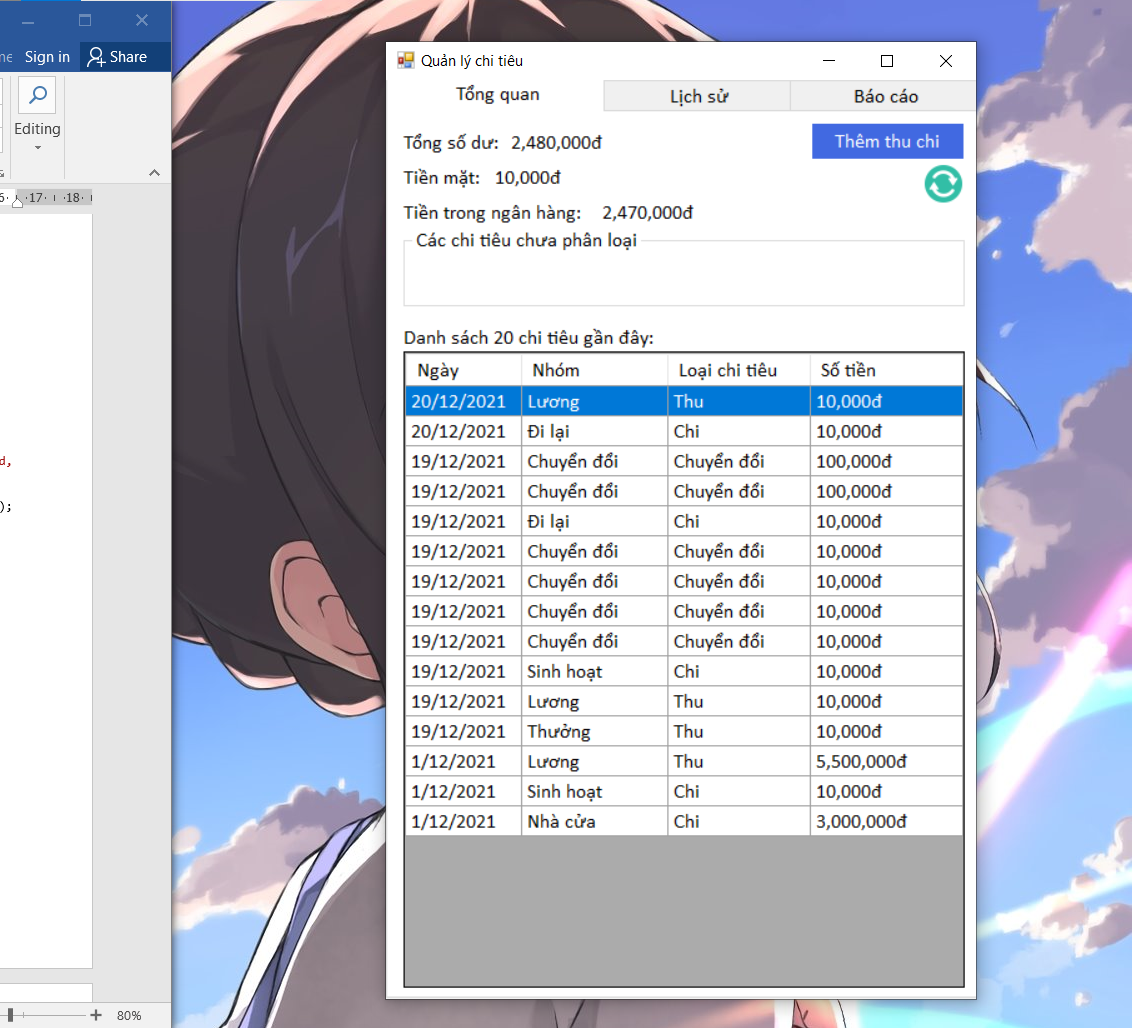
}

thunhap.Series["ThuNhapPie"].IsValueShownAsLabel = true;

}

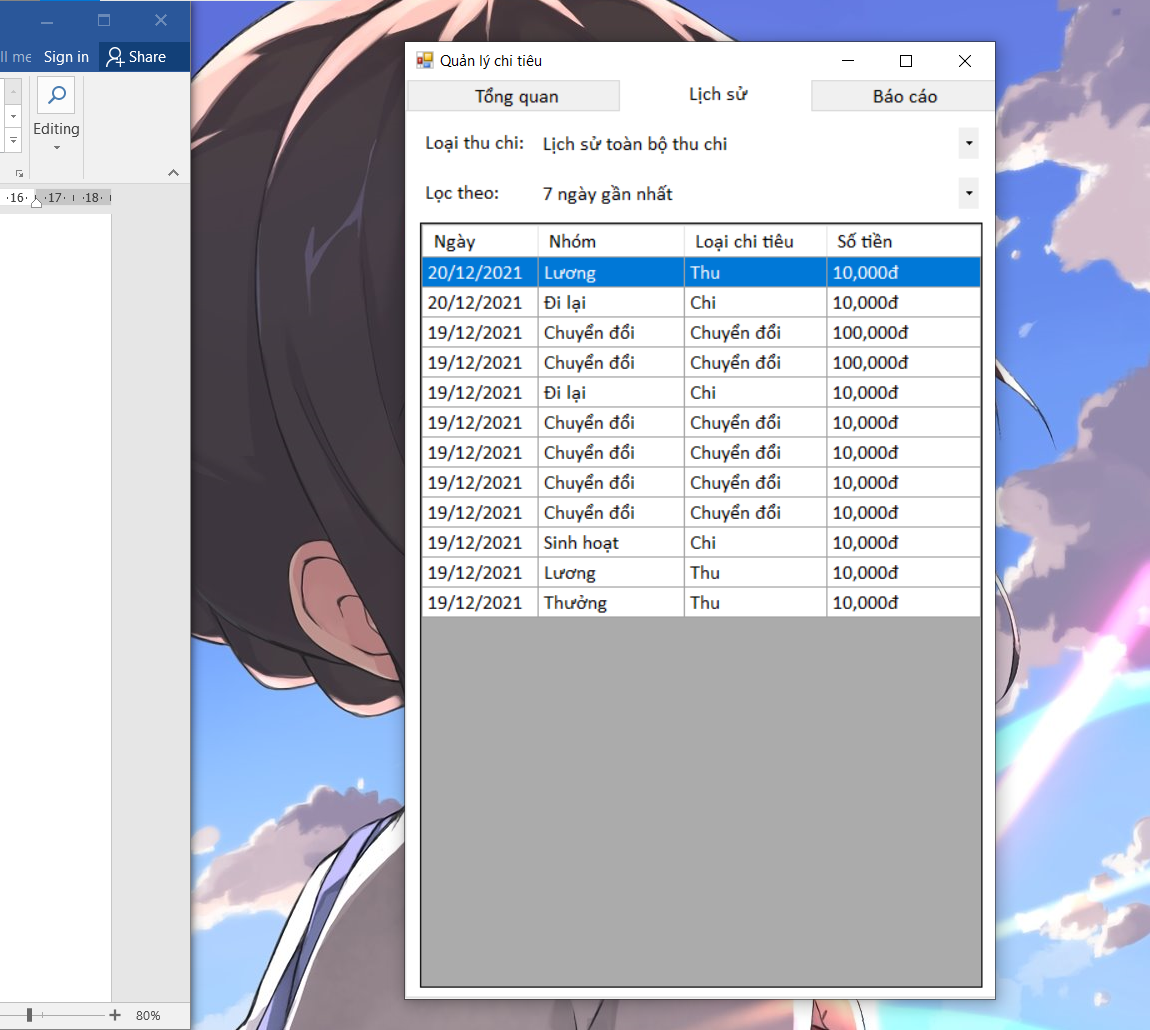
## 4.2. Kết quả lập trình

### 4.2.1 Giao diện Tổng quan



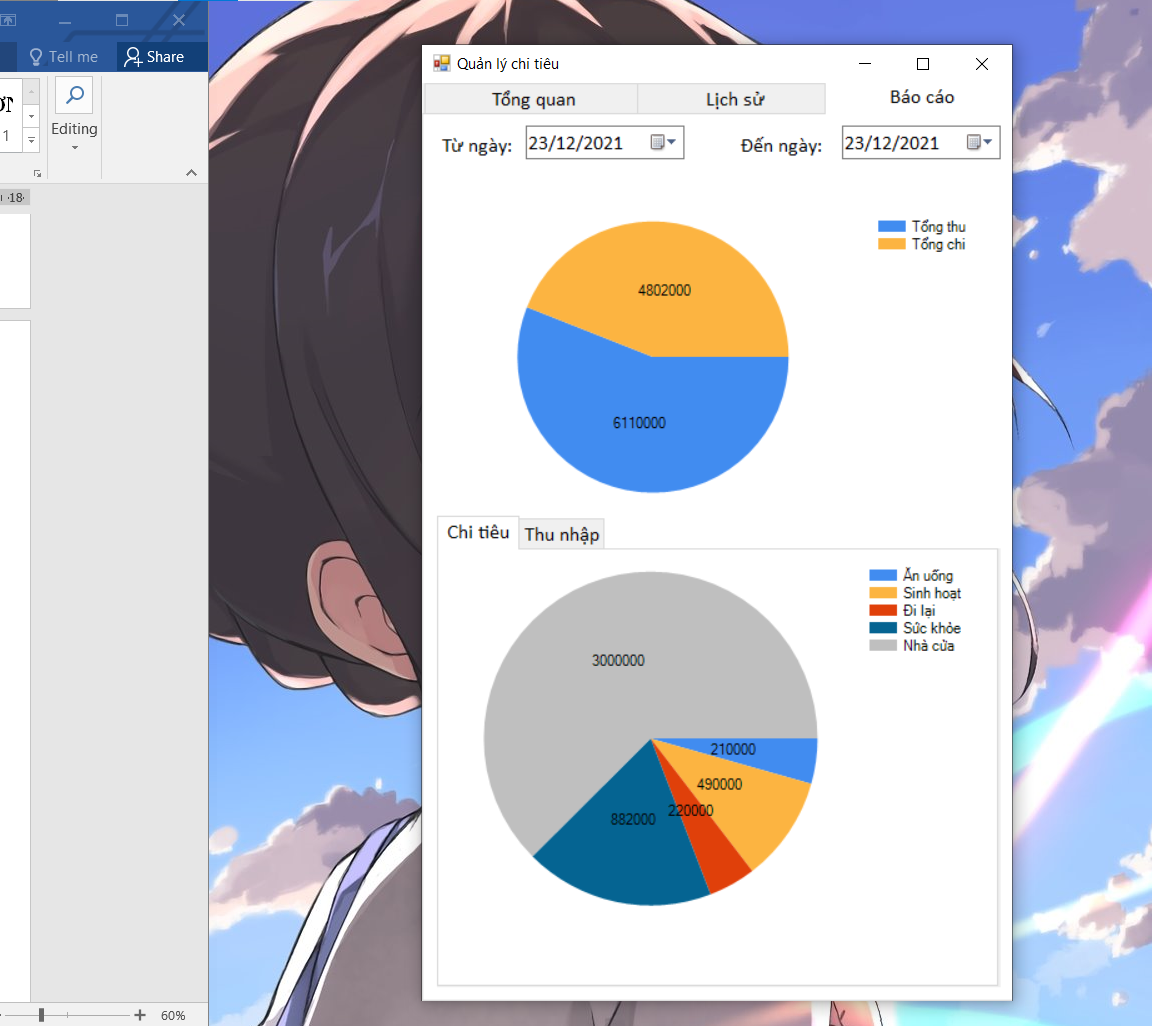
Hình 9: Giao diện tổng quan

### 4.2.2. Giao diện lịch sử thu chi



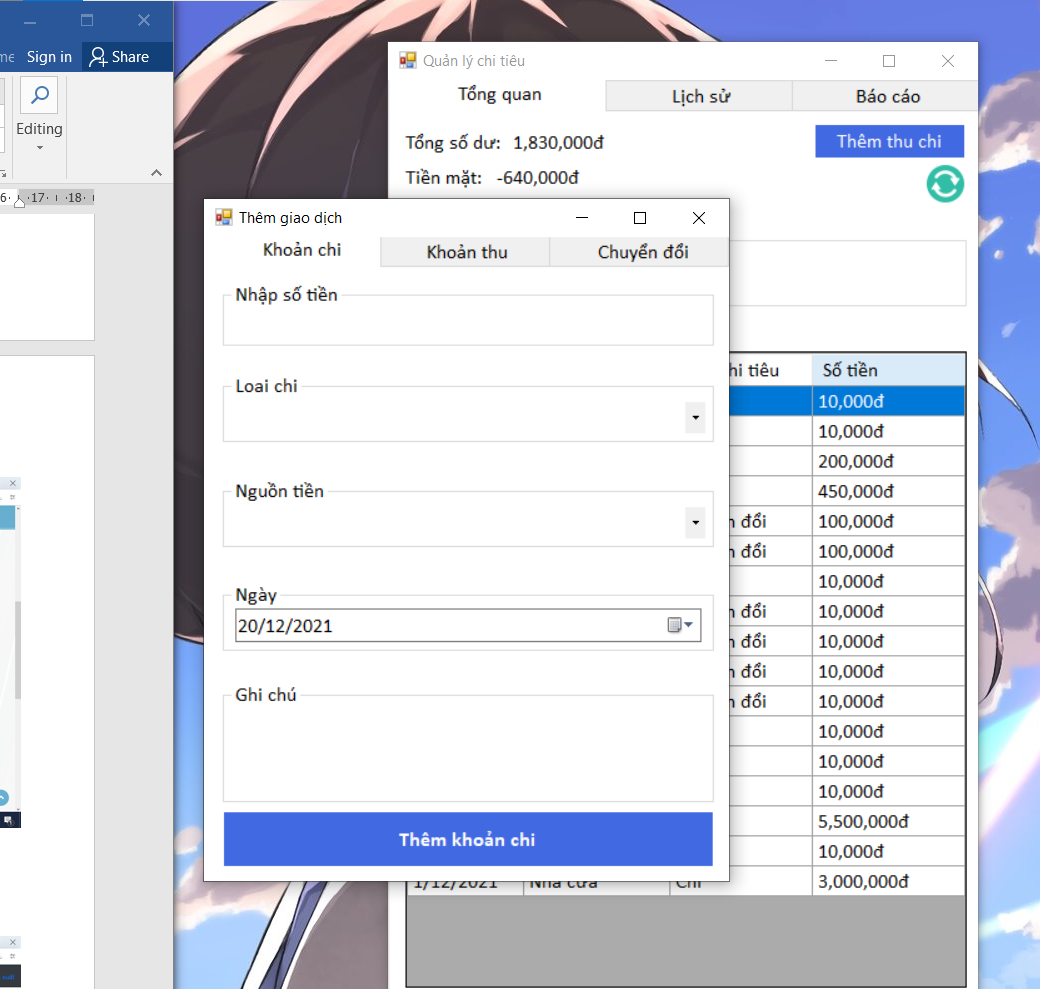
Hình 10: Giao diện lịch sử

### 4.2.3. Giao diện thông kê báo cáo



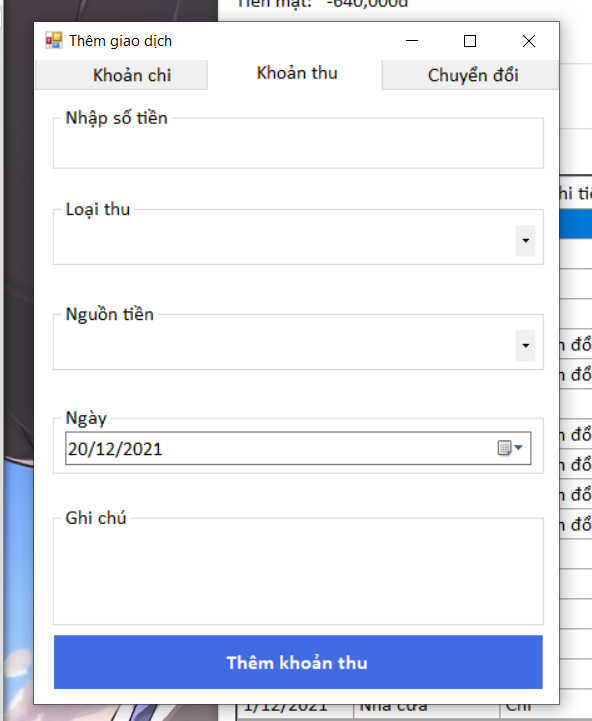
Hình 11: Giao diện thống kê báo cáo

### 4.2.4. Giao diện thêm chi tiêu

\ 

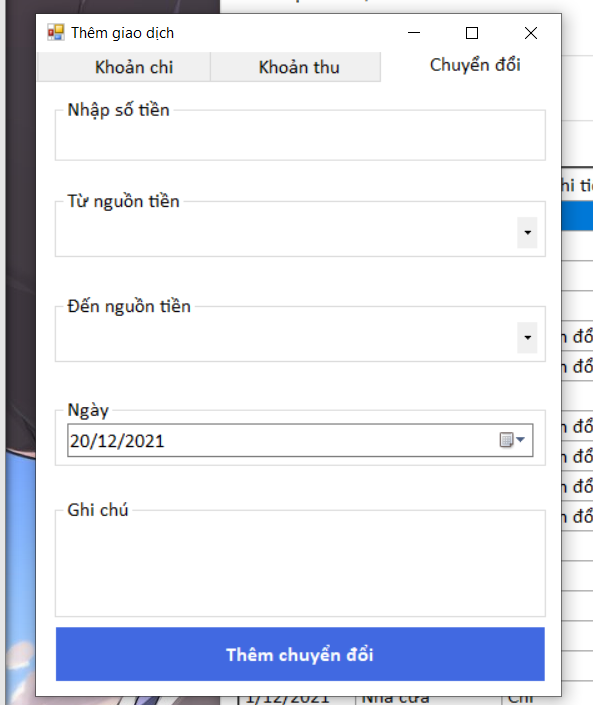
Hình 12: Giao diện thêm chi tiêu

### 4.2.6. Giao diện thêm thu nhập



Hình 13: Giao diện thêm thu nhập

### 4.2.6. Giao diện chuyển đổi nguồn tiền



Hình 14: Giao diện chuyển đổi nguồn tiền

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

Như vậy, Trong quá trình xây dựng phần mềm quản lý chi tiêu, em đã đạt được những mục tiêu sau:

* Nâng cao được lập trình cũng như phát triển hệ thống.
* Xây dựng thành công phần mềm quản lý chi tiêu đơn giản hiệu quả.

Trong quá trình xây dựng hệ thống vì chưa có nhiều kinh nghiệm nên vẫn còn chưa thực sự hoàn thiện em mong thầy có thể góp ý và chỉ ra những thứ còn thiếu sót của em. Em trân thành cảm ơn!.

## 5.2. Hướng phát triển

Những hướng phát triển trong tương lai của chúng em gồm có:

* Xây dựng hệ thống có thể đáp ứng được nhu cầu do cho nhiều người dùng hơn.
* Xây dựng hệ thống có độ bảo mật cao hơn.
* Xây dựng hệ thống có nhiều chức năng đáp ứng tính mới
* Xây dựng hệ thống trên mobile

**TÀI LIỆU THAM KHẢO**

[1]. ***.Net Framework*** (https://vi.wikipedia.org/wiki/.NET\_Framework)

[2]. ***SQL Server*** (<https://timoday.edu.vn/>)