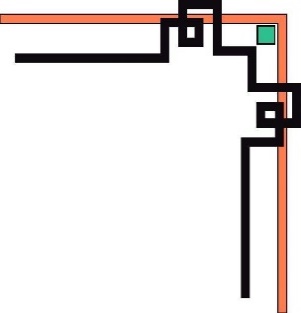
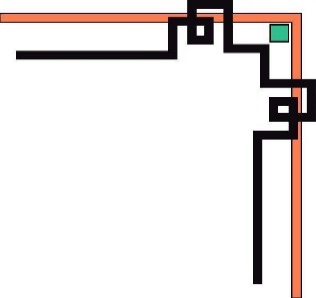
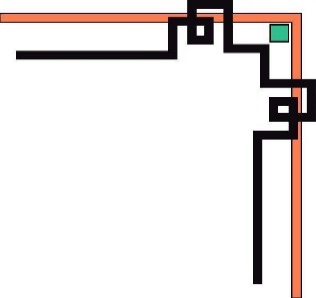
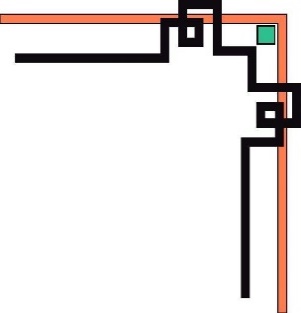
**HỌC VIỆN KỸ THUẬT QUÂN SỰ**



**Khoa Công Nghệ Thông Tin**

****

**Báo Cáo**

**CÔNG NGHỆ LẬP TRÌNH TÍCH HỢP**

|  |  |
| --- | --- |
| **Thành viên** | **Phân công** |
| 1. Vũ Mạnh Tuấn | Xây dựng giao diện ban đầu, các chức năng: tạo tài khoản, quản lý khoản thu,  hạn mức chi, dòng thời gian, báo cáo, thiết lập |
| 1. Ngô Thị Dung | Chức năng: Quản lý tiết kiệm, mục đích tiết kiệm, tiện ích |
| 1. Tăng Thị Huyền Trang | Chức năng quản lý khoản chi, giới thiệu sản phẩm, đăng nhập |
| 1. Nguyễn Đức Sơn | Chức năng quản lý vay- cho vay, quản lý người vay |

*Giáo viên hướng dẫn*: Vũ Văn Trường

Thành viên và phân công

Mục Lục

[**1. Các chức năng của phần mềm** 3](#_Toc533249716)

[1.1. Biểu đồ phân rã chức năng 3](#_Toc533249717)

[1.2. Quy trình nghiệp vụ 4](#_Toc533249718)

[**2. Quy trình làm phần mềm** 7](#_Toc533249719)

[**3. Các bước giải quyết từng chức năng** 8](#_Toc533249720)

[3.1. Chức năng đăng nhập 10](#_Toc533249721)

[3.2. Đăng ký tài khoản 12](#_Toc533249722)

[3.3 Quản lý khoản thu 17](#_Toc533249723)

[3.4 Quản lý khoản thu 19](#_Toc533249724)

[3.4 Quản lý khoản chi 22](#_Toc533249725)

[3.5 Hạn mức chi 32](#_Toc533249726)

[3.6 Dòng thời gian 35](#_Toc533249727)

[3.7 Quản lý khoản Vay-ChoVay-Người Vay 38](#_Toc533249728)

[**3**.8 Tiết kiệm 87](#_Toc533249729)

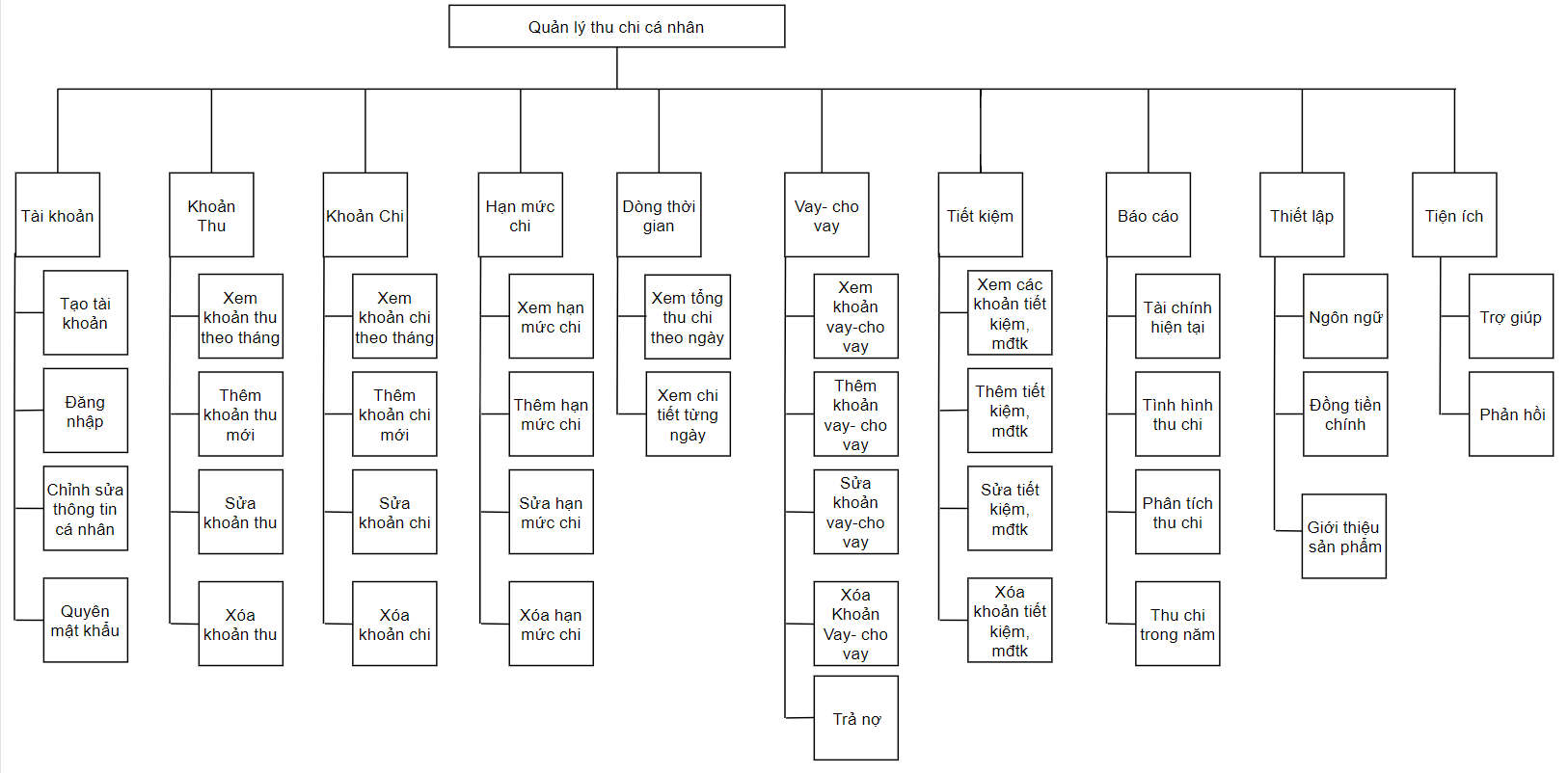
[3.9 Báo cáo 114](#_Toc533249730)

[**4. Giao diện chung của cả chương trình.** 125](#_Toc533249731)

Phần mềm Quản Lý Thu Chi Cá Nhân

**1. Các chức năng của phần mềm**

## 1.1. Biểu đồ phân rã chức năng



## 1.2. Quy trình nghiệp vụ

- Người dùng mở chương trình lên và tiến hành đăng nhập vào hệ thống nếu đã có tài khoản, nếu chưa có tài khoản người dùng sử dụng chức năng đăng ký tài khoản để tạo tài khoản cho mình, điều kiện tạo tài khoản là phải có một email hợp lệ và truy cập được, hệ thống sẽ sử dụng mail đó để xác thưc,

- Sau khi đăng nhập vào hệ thống người dùng có thể sử dụng các chức năng của phần mềm

1.2.1 Quản lý khoản thu

- Người dùng lựa chọn chức năng quản lý khoản thu trên giao diện phần mềm

- Hệ thống sẽ đưa ra các khoản thu trong tháng hiện tại, người dùng có thể xem các khoản thu của tháng khác bằng cách sử dụng chức năng xem theo tháng, người dùng chọn thời gian muốn xem và bấm xem.

- Thêm khoản thu mới: người dùng bấm vào thêm khoản thu, sau khi điển đầy đủ thông tin thì bấm thêm, hệ thống báo kết quả đã thêm hoặc chưa thêm được

- Sửa khoản thu: người dùng chọn khoản thu cần sửa, sau khi sửa thông tin xong, người dùng bấm lưu, hệ thống báo kết quả đã lưu hoặc chưa lưu được

- Xóa khoản thu: người dùng chọn xóa khoản thu đã thêm trước đó, hệ thống báo kết quả đã xóa hoặc chưa xóa.

1.2.2. Quản lý khoản chi

- Người dùng lựa chọn chức năng quản lý khoản chi trên giao diện phần mềm

- Hệ thống sẽ đưa ra các khoản chi trong tháng hiện tại, người dùng có thể xem các khoản chi của tháng khác bằng cách sử dụng chức năng xem theo tháng, người dùng chọn thời gian muốn xem và bấm xem.

- Thêm khoản chi mới: người dùng bấm vào thêm khoản chi, sau khi điển đầy đủ thông tin thì bấm thêm, hệ thống báo kết quả đã thêm hoặc chưa thêm được

- Sửa khoản chi: người dùng chọn khoản chi cần sửa, sau khi sửa thông tin xong, người dùng bấm lưu, hệ thống báo kết quả đã lưu hoặc chưa lưu được

- Xóa khoản chi: người dùng chọn xóa khoản chi đã thêm trước đó, hệ thống báo kết quả đã xóa hoặc chưa xóa.

1.2.3. Hạn mức chi

- Người dùng thiết lập hạn mức chi cho các loại khoản chi trong tháng

- Thêm hạn mức chi: người dùng chọn thêm hạn mức chi, sau khi điền đủ thông tin thì bấm lưu, lưu í trong một tháng, một loại khoản chi chỉ được thiết lập hạn mức một lần.

- Sửa hạn mức chi: người dùng chọn hạn mức cần sửa, sau khi sửa thông tin xong thì bấm lưu

- Xóa hạn mức chi: người dùng chọn hạn mức cần xóa và bấm đồng í xóa.

1.2.4 Dòng thời gian

- Chức năng này cho phép người dùng nắm được tình hình thu chi trong tháng hiện tại bằng cách đưa ra tổng thu chi theo từng ngày trong tháng.

- Người dùng bấm xem dòng thời gian

- Chọn ngày cần xem chi tiết, hệ thống trả về kết quả chi tiết các khoản thu chi

1.2.5 Vay – cho vay

- Người dùng quản lý các khoản đi vay và các khoản cho vay

- Thêm khoản vay- cho vay: người dùng lựa chọn thêm khoản vay hoặc cho vay, sau khi điền đủ thông tin thì bấm lưu

- Sửa khoản vay, cho vay: người dùng lựa chọn khoản cần sửa và bấm lưu

- Xóa khoản vay, cho vay: người dùng lựa chọn khoản cần xóa và bấm đồng í xóa

- Trả nợ: người dùng lựa chọn khoản vay, hoặc cho vay cần trả và bấm trả, sau khi điền đủ thông tin trả nợ thì chọn lưu

1.2.6 Tiết kiệm

- Người dùng quản lý các khoản tiết kiệm của mình, mỗi cá nhân có những mục đích tiết kiệm, người dùng thêm các mục đích tiết kiệm trước, và các khoản tiết kiệm thực hiện cho các mục đích tiết kiệm này.

-Thêm mục đích tiết kiệm: lựa chọn thêm mục đích tiết kiệm, điền đủ thông tin và chọn lưu

- Sửa mục đích tiết kiệm: lựa chọn sửa mục đích, sửa thông tin và chọn lưu

- Xóa mục đích tiết kiệm: lựa chọn xóa, bấm đồng í xóa

- Thêm tiết kiệm: thêm các khoản tiền tiết kiệm cho mục đích tiết kiệm

- Sửa tiết kiệm: chọn lần tiết kiệm cần sửa và lưu

-Xóa tiết kiệm: lựa chọn lần tiết kiệm cần xóa và chọn đồng í xóa

1.2.7 Báo cáo

- Tình hình thu chi: người dùng nhận được thông tin thu chi trong ngày, trong tuần, trong tháng và trong năm

- Tài chính hiện tại: người dùng nhận được thông tin về số tiền có trong ví, số tiền trong tài khoản atm, số tiền cho vay, và số tiền mình vay

- Phân tích chi tiêu: hệ thống sử dụng biểu đồ tròn để phân tích các khoản thu chi theo loại khoản thu chi của người dùng trong một tháng

- Thu chi theo năm: hệ thống sử dụng biểu đồ cột để đưa ra tổng thu chi theo tháng trong năm hiện tại, từ đó người dùng dễ dàng so sánh mực độ thu chi giữa các tháng trong một năm

1.2.8 Thiết lập

- Ngôn ngữ: người dùng lựa chọn ngôn ngữ hiển thị

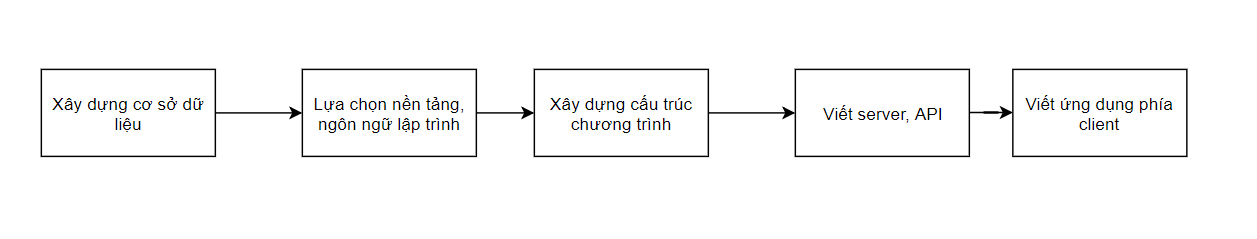
- Đồng tiền chính: lựa chọn đồng tiền sử dụng, ví dụ: việt nam đồng

1.2.9 Tiện ích

- Phản hồi: người dùng phản hồi thông tin

- Trợ giúp: xem các trợ giúp, hướng dẫn sử dụng các chức năng

# **2. Quy trình làm phần mềm**



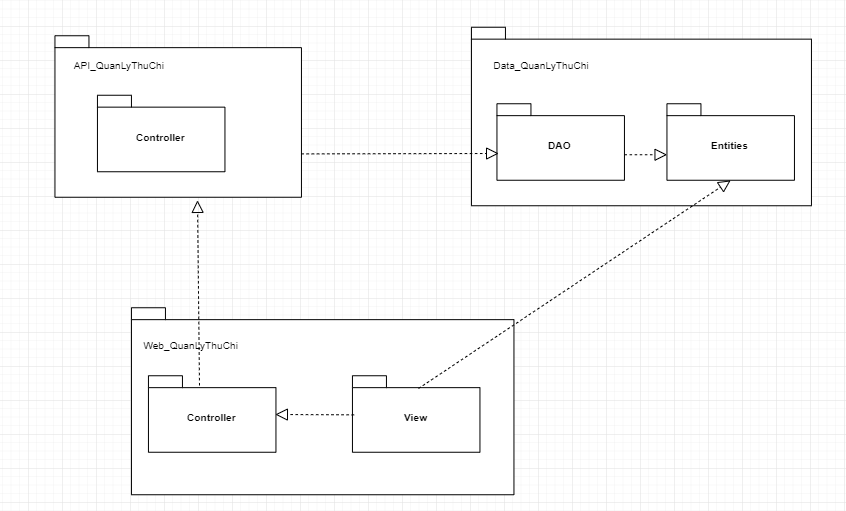
* Phân chia công việc trong nhóm: cả nhóm cùng nhau xây dựng cơ sở dữ liệu, xác định các chức năng của phần mềm, xây dựng cấu trúc chương trình
* Mỗi thành viên sẽ được phân công làm chức năng cụ thể, trong đó thì thành viên nào phụ trách chức năng nào thì phải xây dựng phía server, API và viết chức năng phía client
* Phân công cụ thể như sau:

|  |  |
| --- | --- |
| **Thành viên** | **Phân công** |
| 1. Vũ Mạnh Tuấn | Xây dựng giao diện ban đầu, các chức năng: tạo tài khoản, quản lý khoản thu,  hạn mức chi, dòng thời gian, báo cáo, thiết lập |
| 1. Ngô Thị Dung | Chức năng: Quản lý tiết kiệm, mục đích tiết kiệm, tiện ích |
| 1. Tăng Thị Huyền Trang | Chức năng quản lý khoản chi, giới thiệu sản phẩm, đăng nhập |
| 1. Nguyễn Đức Sơn | Chức năng quản lý vay- cho vay, quản lý người vay |

# **3. Các bước giải quyết từng chức năng**

- Nhóm thực hiện trên 2 nền tảng là Web MVC và window form

Mô hình chương trình



Mô tả: Hệ thống được xây dựng dựa theo mô hình ASP.NET Web API bao gồm 3 gói chính: API\_QuanLyThuChi, Data\_QuanLyThuChi và Web\_QuanLyThuChi liên kết với nhau.

- Gói API\_QuanLyThuChi gồm gói nhỏ là gói Controller

-Gói Data\_QuanLyThuChi gồm 2 gói con là gói DAO và Entities .Gói DAO chứa các hàm xử lý,thực thi,tương tác với cơ sở dữ liệu,và gọi tới các lớp trong Entities.Gói Entities gồm các lớp là lớp chứa thông tin đối tượng (Dữ liệu), chỉ có các thuộc tính và các hàm get, set.

-Gói Web\_QuanLyThuChi gồm 2 gói con:

+ Gói View gồm các lớp giao diện của hệ thống, nhận dữ liệu từ Controller và sử dụng các hàm trong lớp Entities để hiển thị dữ liệu.

+ Gói Controller gồm các lớp điều khiển có chức năng nhận yêu cầu từ người dùng và sử dụng các lớp trong gói ASP\_QuanLyThuChi và View để xử lý và trả kết quả cho người dùng.

## **3.1. Chức năng đăng nhập**

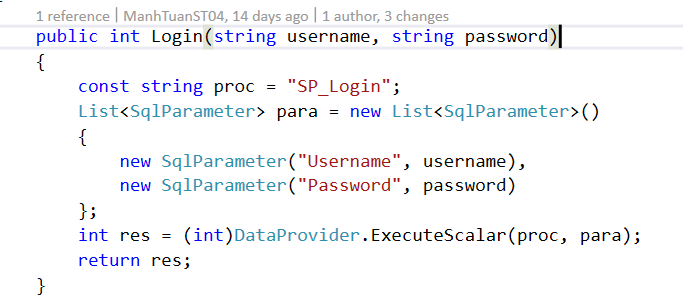
1. Ý tưởng giải quyết

- Nếu người dùng nhập khớp tên đăng nhập và mật khẩu đã đăng ký thì có thể đăng nhập thành công vào hệ thống, nếu không thì hệ thống báo lỗi và yêu cầu nhập lại

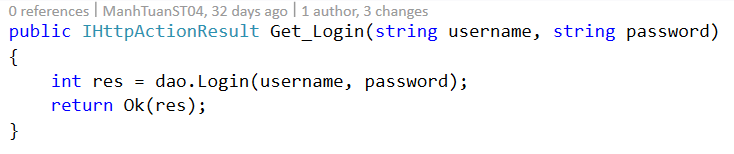
2. Viết code trên các lớp

2.1 Lớp Data:

- Hàm Login



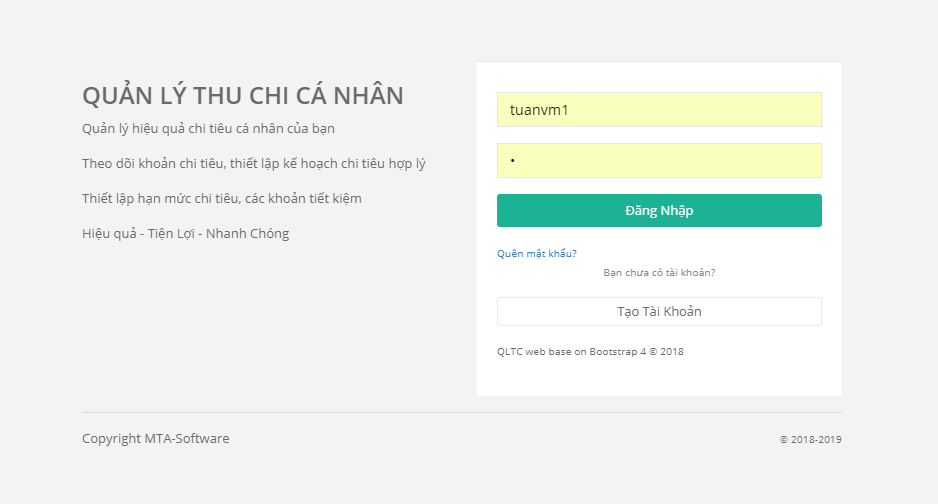
2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện



4. Kết quả chạy

- Hiện ra giao diện đăng nhập, nếu người dùng nhập khớp tên đăng nhập và mật khẩu thì sẽ vào được giao diện quản lý chính của chương trình, nếu sai thì hệ thống báo lỗi và yêu cầu nhập lại

## **3.2. Đăng ký tài khoản**

1. Ý tưởng giải quyết

- Khi người dùng đăng ký thông tin thì tiến hành kiểm tra tên đăng nhập, mail để tránh trùng với các tài khoản đã có.

- Sau đó xác thực email người dùng đã đăng ký, gửi mã xác nhận vào mail, người dùng phải kiểm tra mail, nhập mã này để hoàn tất quá trình đăng ký.

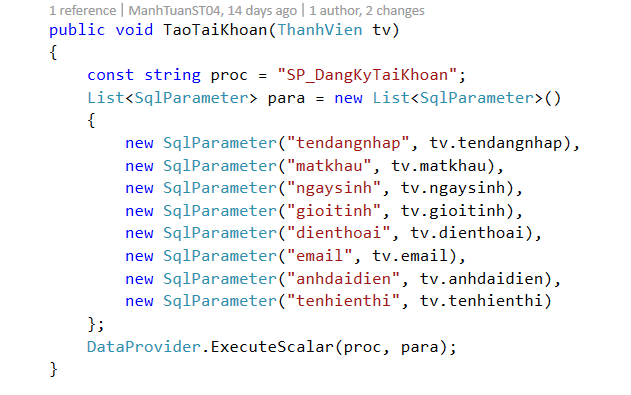
2. Viết code trên các lớp

2.1. Lớp Data

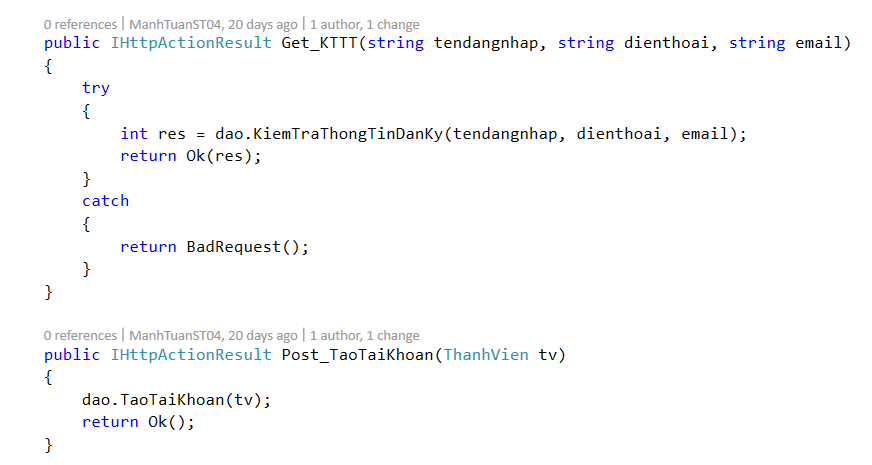
Hàm kiểm tra thông tin đăng ký



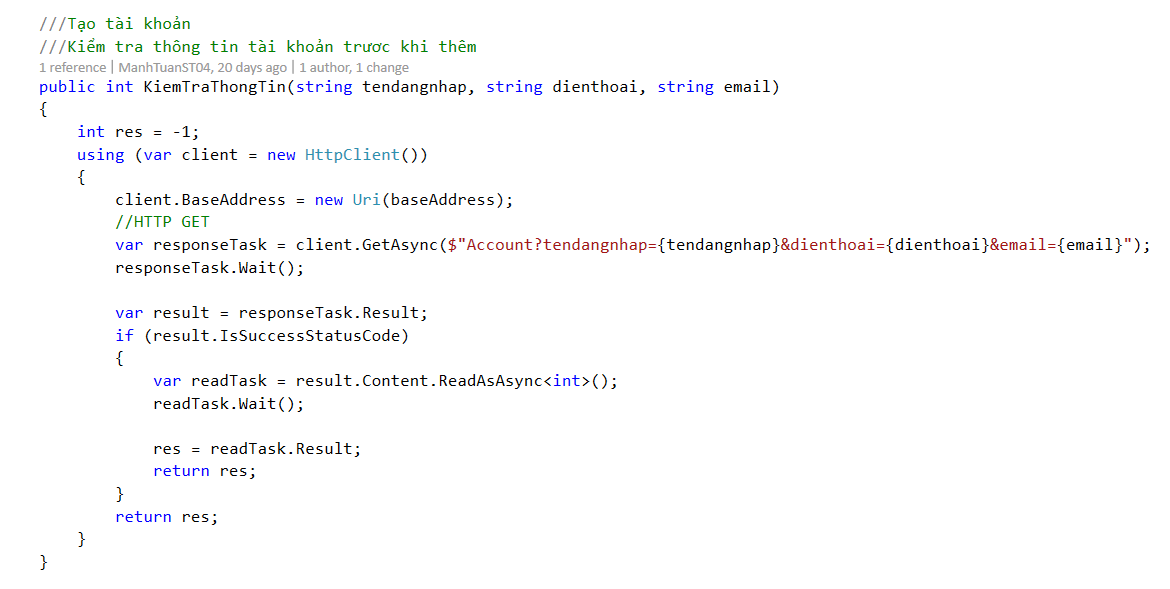
Hàm tạo tài khoản



2.2. Lớp API



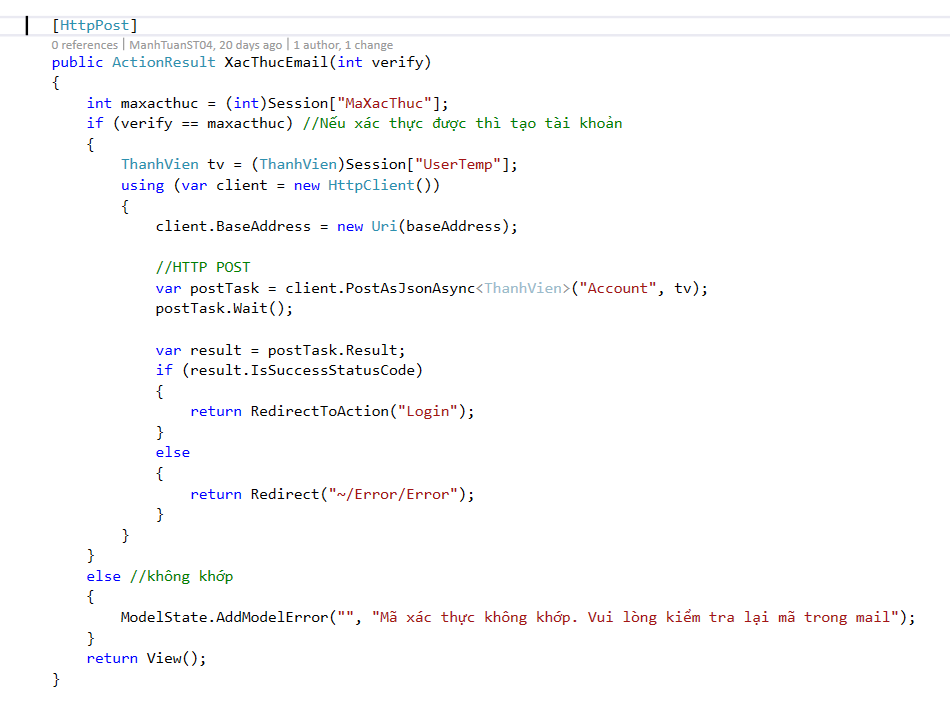
2.3. Lớp Web – Phía client



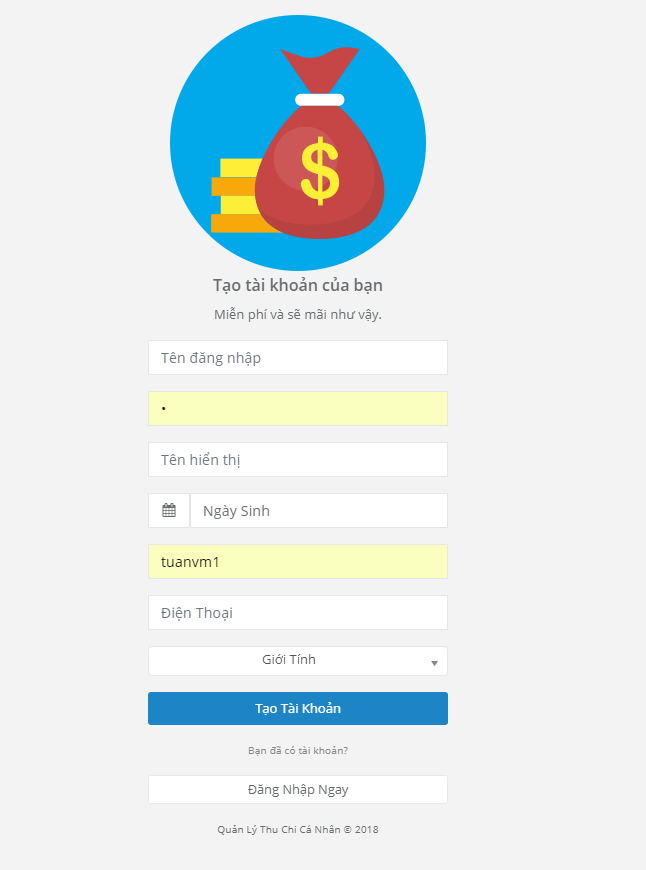
- Hàm tạo tài khoản

public ActionResult TaoTaiKhoan(string tendangnhap, string tenhienthi, string matkhau, DateTime ngaysinh, string gioitinh, string dienthoai, string email)

- Hàm gửi mail xác nhận



3. Giao diện



4. Kết quả chạy

- Hiện ra form đăng ký tài khoản

- Người dùng đăng ký, xác thực mail thành công thì sẽ tạo được tài khoản

- Nếu thông tin đăng ký trùng, hay không xác thực được mail thì không tạo được tài khoản

## **3.3 Quản lý khoản thu**

3.3.1. Thêm khoản thu

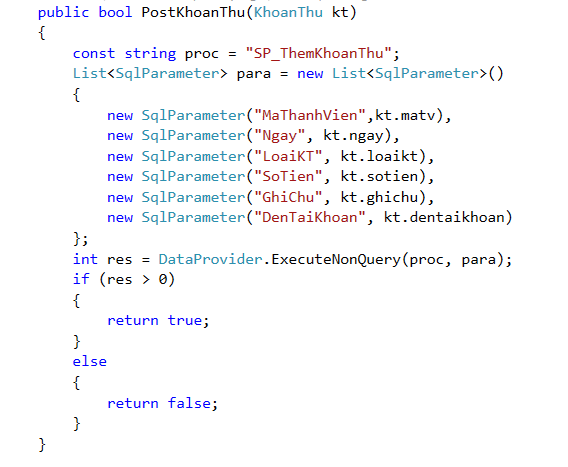
1. Ý tưởng giải quyết

- Người dùng nhập thông tin khoản thu cần thêm. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm khoản chi vào bảng khoản chi trong cơ sở dữ liệu

2. Viết code trên các lớp

2.1. Lớp Data

Hàm thêm khoản thu



2.2. Lớp API



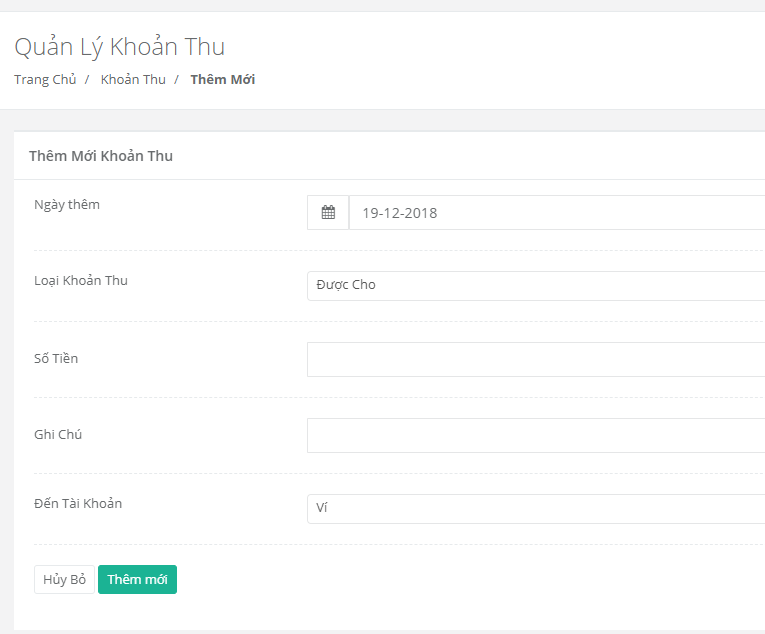
2.3. Lớp Web – Phía client

- Hàm thêm khoản thu

- Hàm load dữ liệu ra combobox tài khoản

- Hàm load dữ liệu ra combobox loại khoản thu

3. Giao diện



4. Kết quả chạy

- Hiện ra giao diện quản thêm mới khoản thu

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công khoản thu

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

## **3.4 Quản lý khoản thu**

3.3.1. Thêm khoản thu

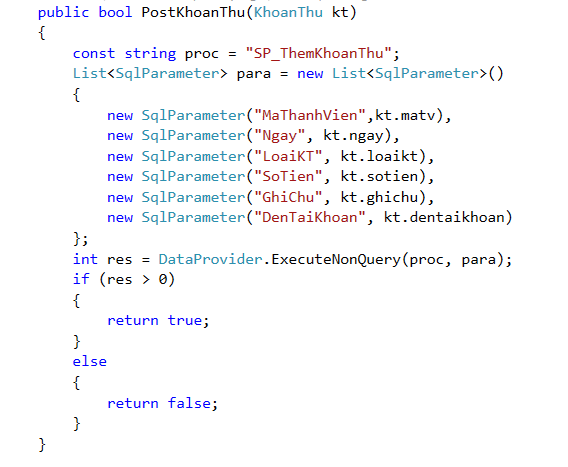
1. Ý tưởng giải quyết

- Người dùng nhập thông tin khoản thu cần thêm. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm khoản thu vào bảng khoản thu trong cơ sở dữ liệu

2. Viết code trên các lớp

2.1. Lớp Data

Hàm thêm khoản thu



2.2. Lớp API



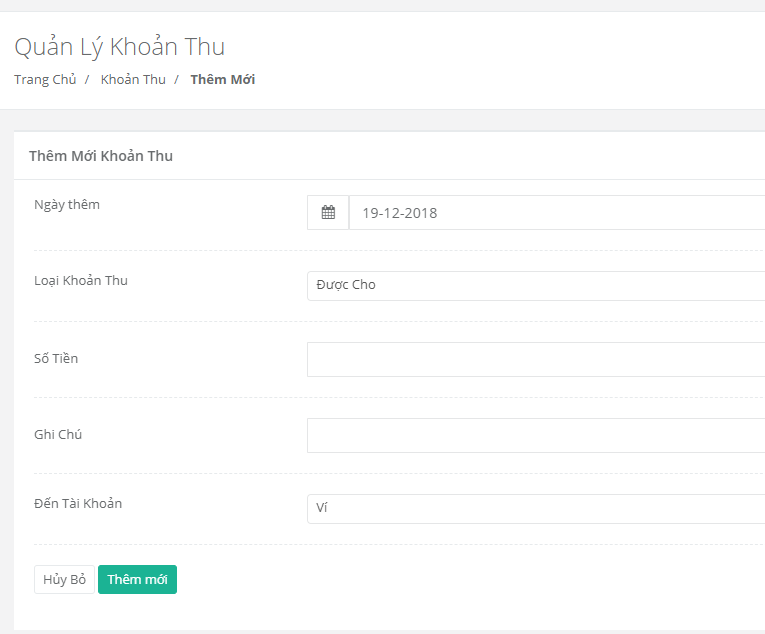
2.3. Lớp Web – Phía client

- Hàm thêm khoản thu

- Hàm load dữ liệu ra combobox tài khoản

- Hàm load dữ liệu ra combobox loại khoản thu

3. Giao diện



4. Kết quả chạy

- Hiện ra giao diện quản thêm mới khoản thu

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công khoản thu

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

## **3.4 Quản lý khoản chi**

3.4.1. Thêm khoản chi

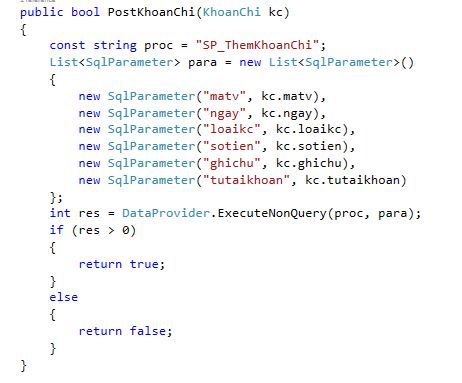
1. Ý tưởng giải quyết

- Người dùng nhập thông tin khoản chi cần thêm. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm khoản chi vào bảng khoản chi trong cơ sở dữ liệu

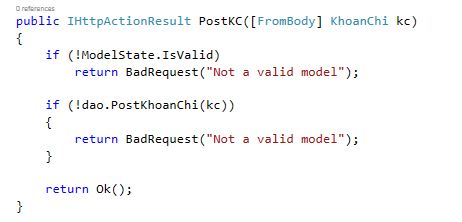
2. Viết code trên các lớp

2.1. Lớp Data

Hàm thêm khoản chi



2.2.Lớp API



2.3. Lớp Web – Phía client

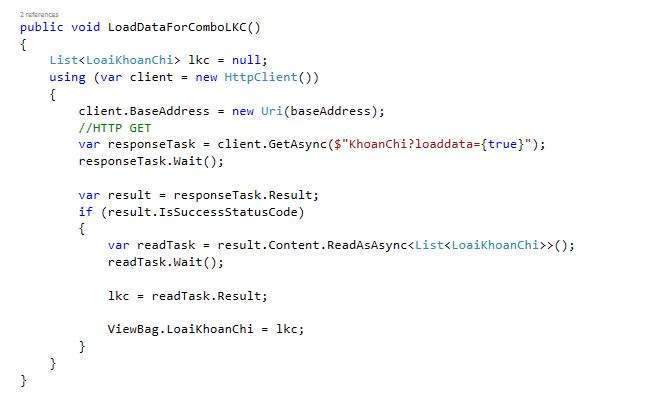
- Hàm thêm khoản chi



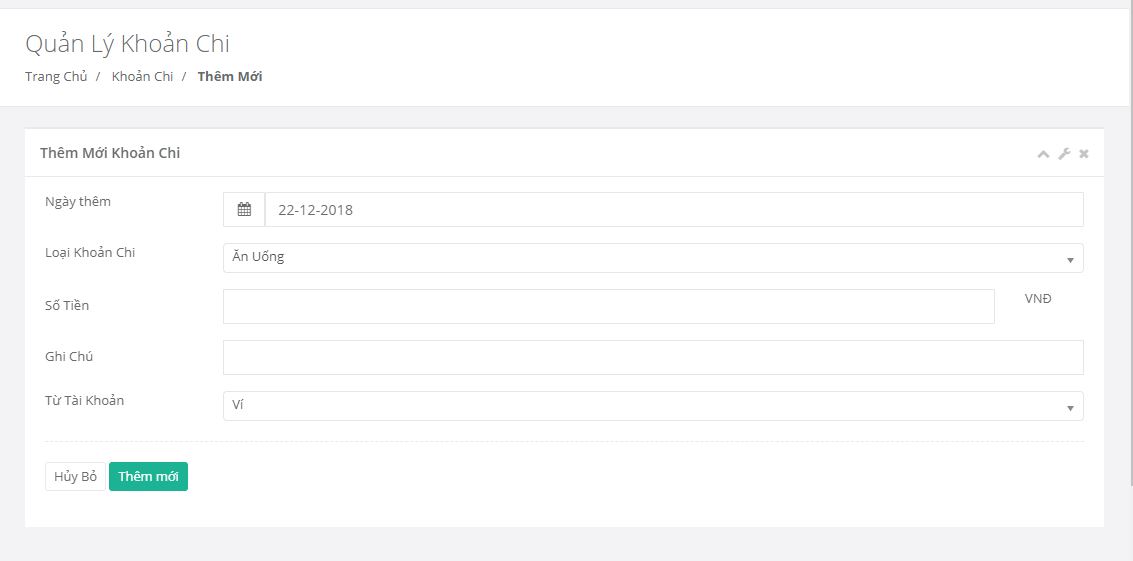
- Hàm load dữ liệu ra combobox tài khoản



- Hàm load dữ liệu ra combobox loại khoản chi



3. Giao diện



4. Kết quả chạy

- Hiện ra giao diện quản thêm mới khoản chi

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công khoản chi

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

3.4.2.Sửa khoản chi

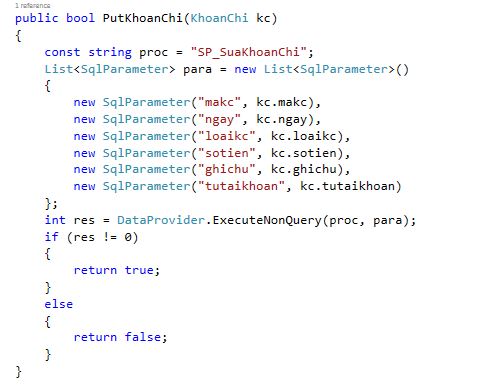
1. Ý tưởng giải quyết

- Người dùng muốn sửa thông tin đã nhập.Phía Server nhận và xử lý yêu cầu.

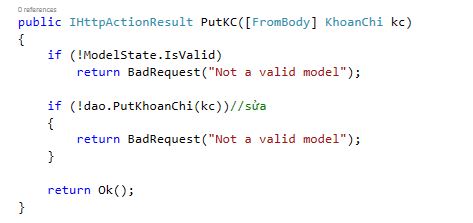
2. Viết code trên các lớp

2.1. Lớp Data

Hàm sửa khoản chi:

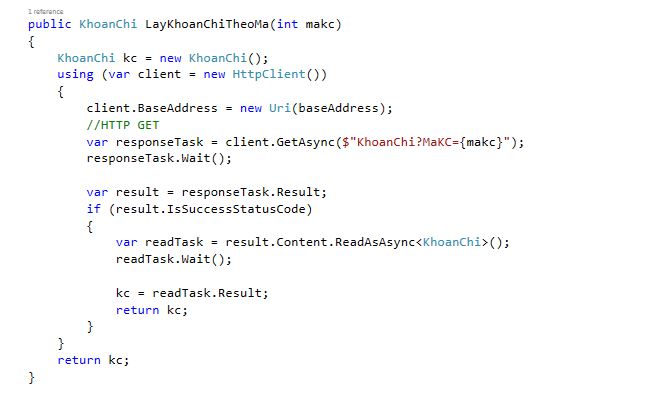


2.2.Lớp API:



2.3. Lớp Web – Phía client

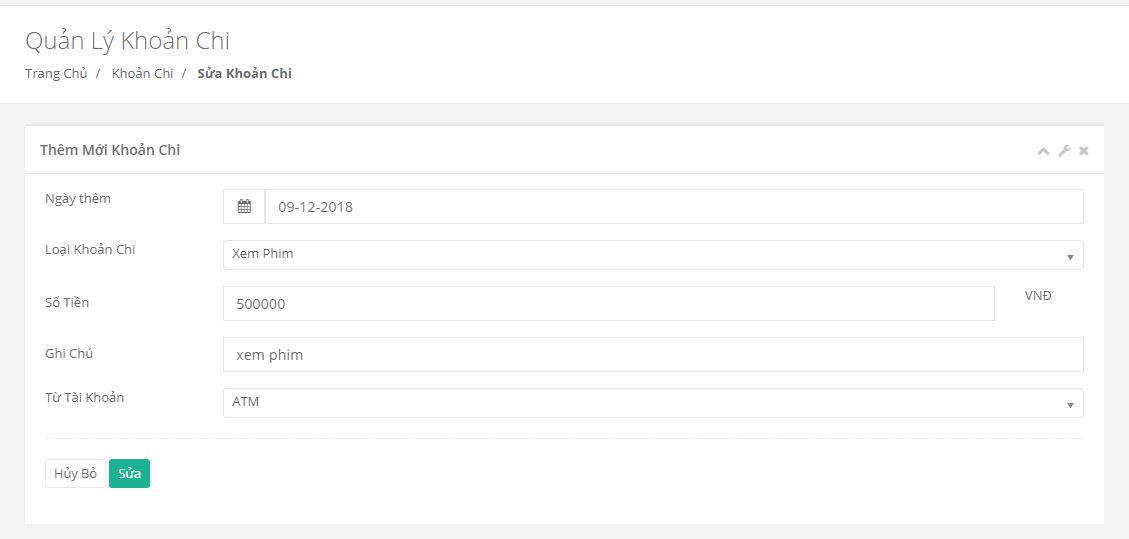
- Hàm lấy Khoản chi theo mã:



**-** Hàm sửa khoản chi:

****

3. Giao diện

****

4. Kết quả chạy

- Hiện ra giao diện quản sửa khoản chi

- Nếu thành công sẽ hiện thông báo sửa thành công

- Nếu thông tin không hợp lệ sẽ báo lỗi không sửa được.

3.4.3.Xóa khoản chi

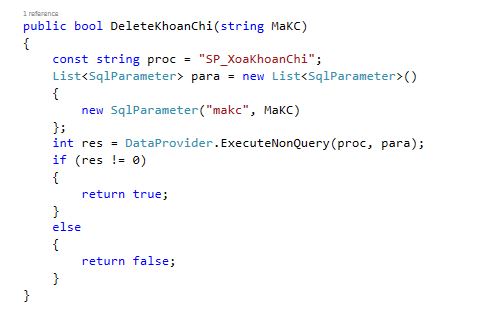
1. Ý tưởng giải quyết

- Người dùng muốn xóa khoản chi.Phía Server nhận và xử lý yêu cầu

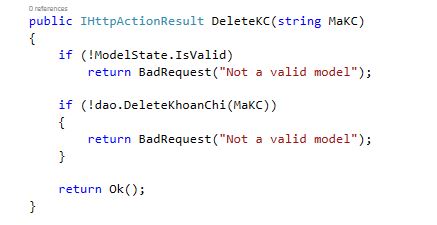
2. Viết code trên các lớp

2.1. Lớp Data

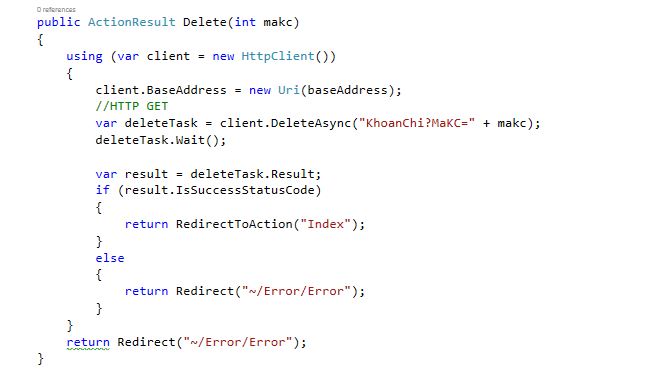
Hàm xóa khoản chi:



2.2.Lớp API



2.3. Lớp Web – Phía client



3.Giao diện



4. Kết quả chạy

- Hiện ra thông báo xóa khoản chi

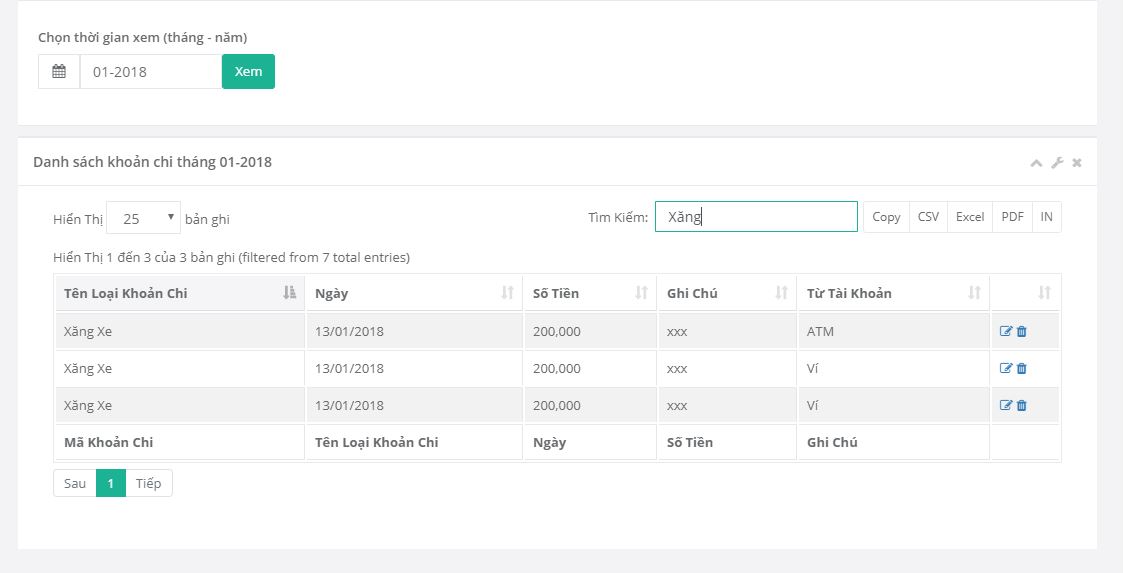
- Nếu thành công sẽ hiện thông báo xóa thành công nếu không thành công sẽ hiện thông báo lỗi xóa khoản chi.

3.4.4.Tìm kiếm khoản chi

Hàm tìm kiếm khoản chi DAO:



Giao diện



Kết quả:

Hiển thị kết quả tìm kiếm hợp lệ mà người dùng nhập vào

## **3.5 Hạn mức chi**

Thiết lập hạn mức chi

1. Ý tưởng giải quyết

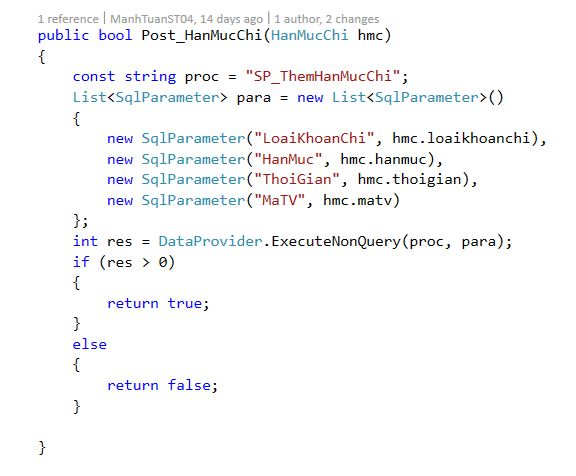
- Người dùng lựa chọn loại khoản chi cần thiết lập hạn mức trong tháng

- Server kiểm tra thông tin, ở đây kiểm tra tính hợp lệ của thông tin nhập vào, trong một tháng một loại khoản chi chỉ được thiết lập hạn mức một lần, cần có hàm kiểm tra, nếu loại khoản chi này đã được thiết lập hạn mức rồi thì thông báo về phía người dùng

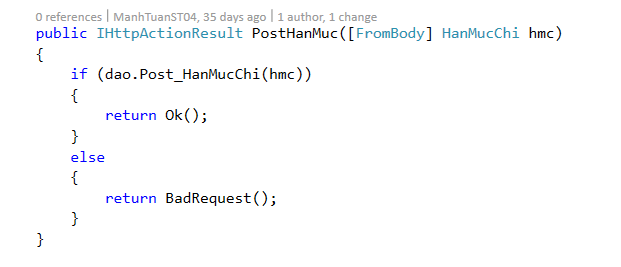
- Nếu thông tin hợp lệ thì server lưu lại vào database

2. Viết code trên các lớp

2.1. Lớp Data



2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện



4. Kết quả chạy

- Hiện ra giao diện thêm hạn mức chi

- Người dùng chọn loại khoản chi cần thiết lập hạn mức, hạn mức cho loại khoản chi đó, và tháng thiết lập hạn mức -> thông tin hợp lệ thì lưu thành công vào hệ thống

## **3.6 Dòng thời gian**

1. Ý tưởng giải quyết

- Chức năng này cho phép người dùng xem tổng thu chi theo từng ngày, từ ngày đầu tháng đến ngày hiện tại trong tháng.

- Các giải quyết là từ bảng khoản thu và bảng khoản chi, ta tính tổng theo từng ngày, với ngày bắt đầu là ngày đầu tháng, xác định dựa vào ngày hiện tại trên hệ thống. Kết quả trả ra gồm ngày, tổng thu, tổng chi.

2. Viết code trên các lớp

2.1. Lớp Data



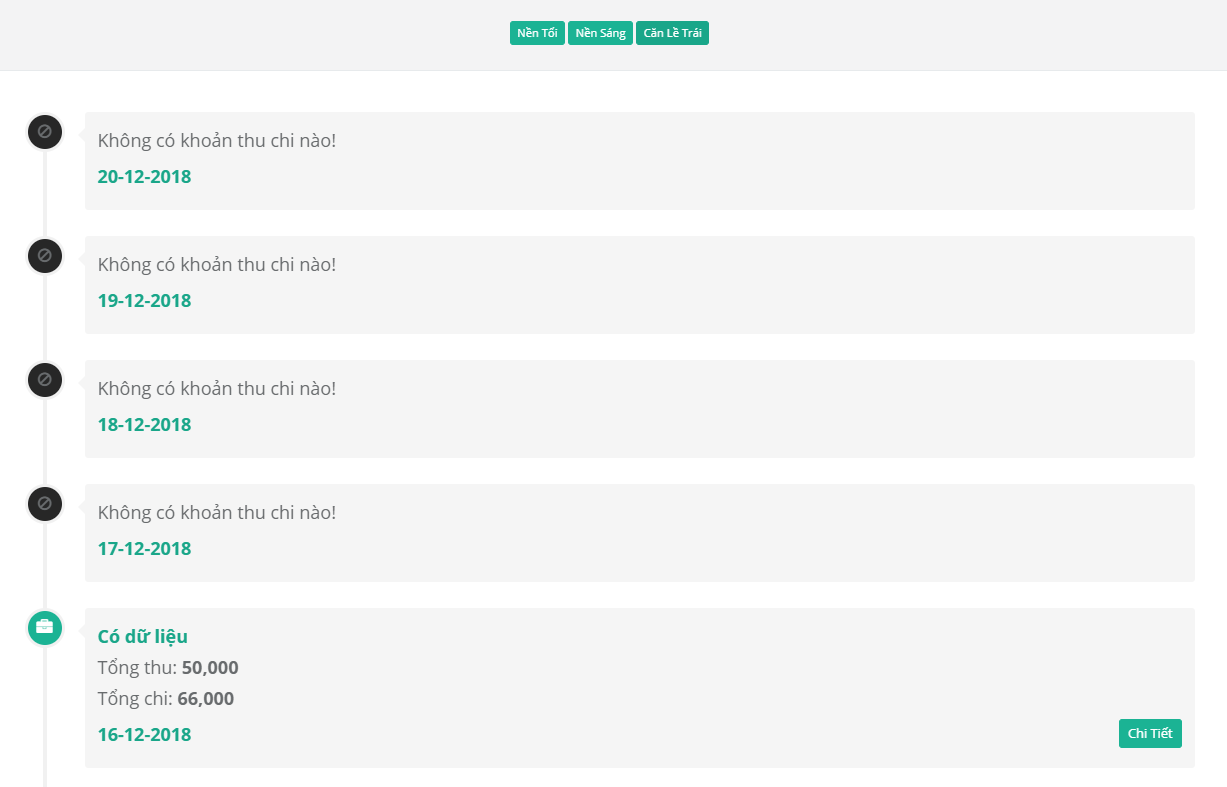
2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện



4. Kết quả chạy

**-** Hiện được ra giao diện người dùng, các ngày và tổng thu chi tương ứng

**3.7 Vay – cho vay**

## **3.7 Quản lý khoản Vay-ChoVay-Người Vay**

**3.7.1. Xem khoản vay**

**3.7.1.1.Xem khoản vay tổng thể**

1. Ý tưởng giải quyết

- Người dùng bấm xem khoản vay. Phía server nhận yêu cầu, hiển thị tất cả khoản vay trong bảng khoản Vay\_Chovay trong cơ sở dữ liệu

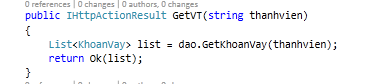
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản Vay



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản Vay

**3.7.1.2.Xem tổng khoản vay theo người cho mình vay**

1. Ý tưởng giải quyết

- Người dùng bấm xem khoản vay theo người vay. Phía server nhận yêu cầu, hiển thị tất cả khoản vay theo người vay trong cơ sở dữ liệu

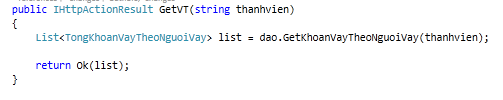
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản Vay theo người cho mình vay



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản vay theo người cho mình vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản Vay theo người cho mình vay

**3.7.1.3.Xem khoản vay chưa trả hết**

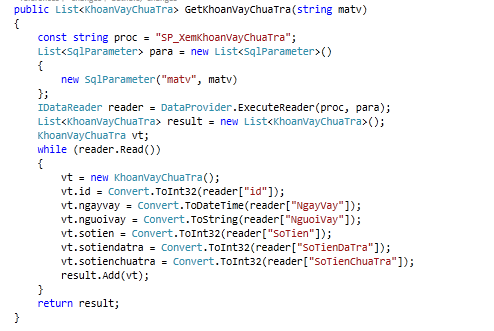
1. Ý tưởng giải quyết

- Người dùng bấm xem khoản vay chưa trả hết. Phía server nhận yêu cầu, hiển thị tất cả khoản vay chưa trả hết trong cơ sở dữ liệu

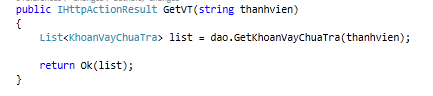
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản Vay chưa trả hết



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản vay chưa trả hết



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản Vay chưa trả hết

**3.7.1.4.Xem khoản vay đã chả hết**

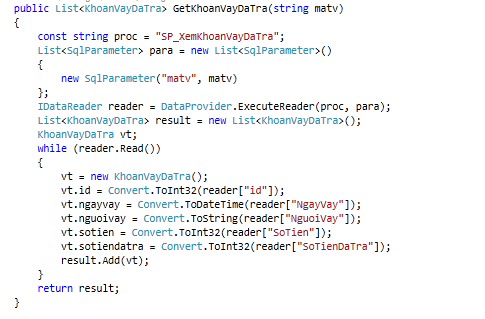
1. Ý tưởng giải quyết

- Người dùng bấm xem khoản vay đã trả hết. Phía server nhận yêu cầu, hiển thị tất cả khoản vay đã trả hết trong cơ sở dữ liệu

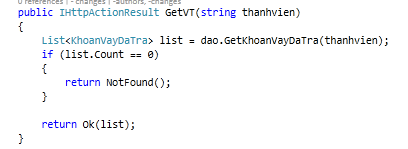
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản Vay đã chả hết



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản vay đã trả hết



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản Vay đã trả hết

**3.7.2. Thêm khoản vay**

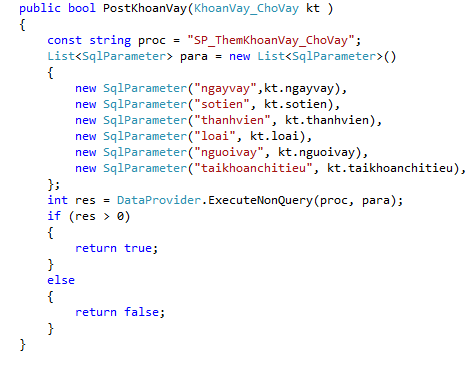
1. Ý tưởng giải quyết

- Người dùng nhập thông tin khoản vay cần thêm. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm khoản vay vào bảng khoản Vay\_Chovay trong cơ sở dữ liệu

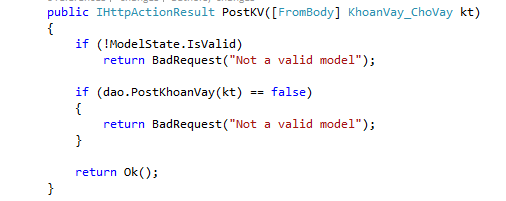
2. Viết code trên các lớp

2.1. Lớp Data

Hàm thêm khoản Vay

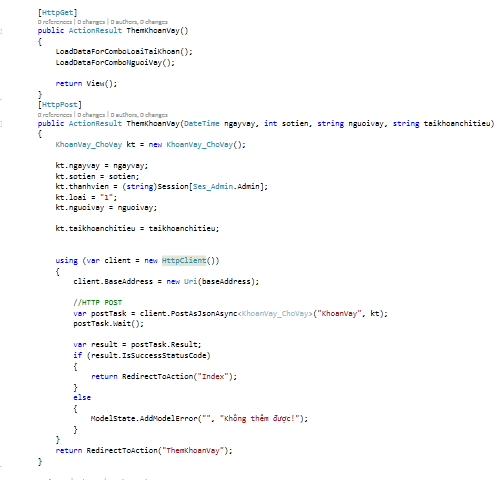


2.2. Lớp API

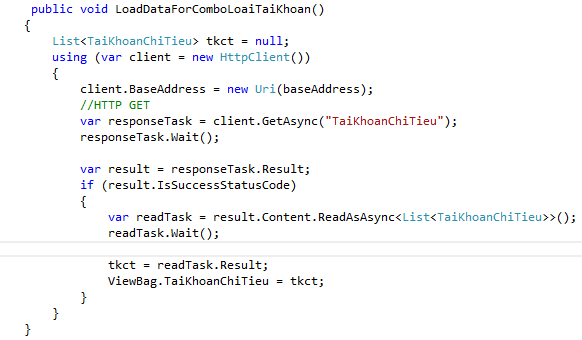


2.3. Lớp Web – Phía client

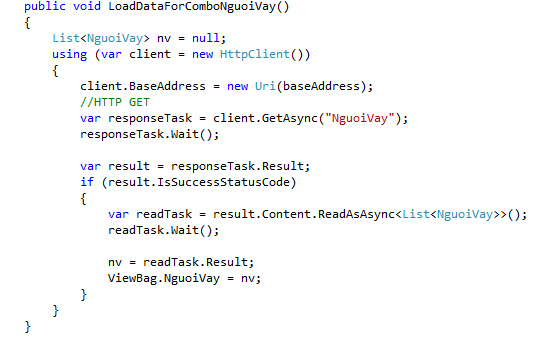
- Hàm thêm khoản vay



- Hàm load dữ liệu ra combobox loại tài khoản



- Hàm load dữ liệu ra combobox người cho vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện quản thêm mới khoản Vay

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công khoản Vay

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

**3.7.3. Sửa khoản vay**

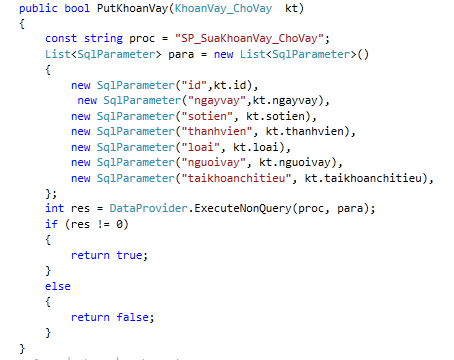
1. Ý tưởng giải quyết

- Người dùng bấm sửa khoản vay , sau đó nhập thông tin cần sửa. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành sửa khoản vay vào bảng khoản Vay\_Chovay trong cơ sở dữ liệu

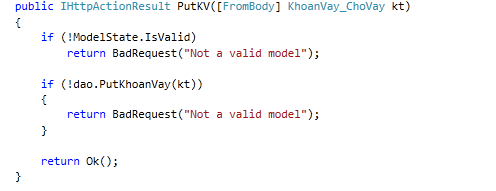
2. Viết code trên các lớp

2.1. Lớp Data

Hàm sửa khoản Vay

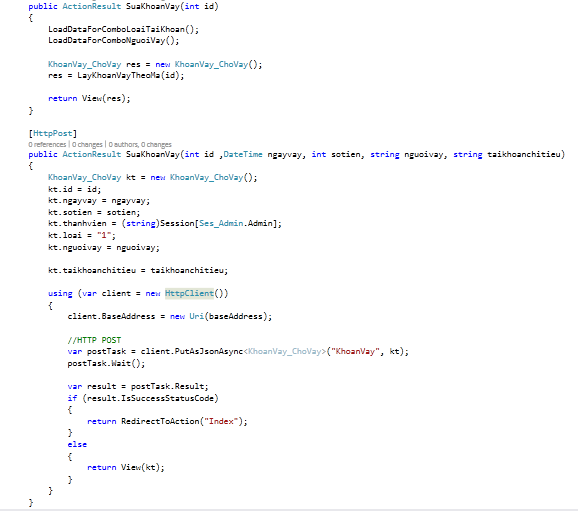


2.2. Lớp API

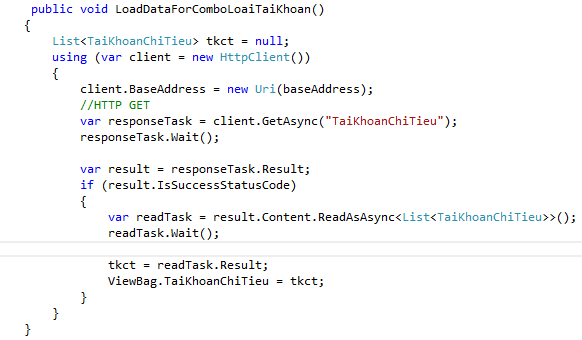


2.3. Lớp Web – Phía client

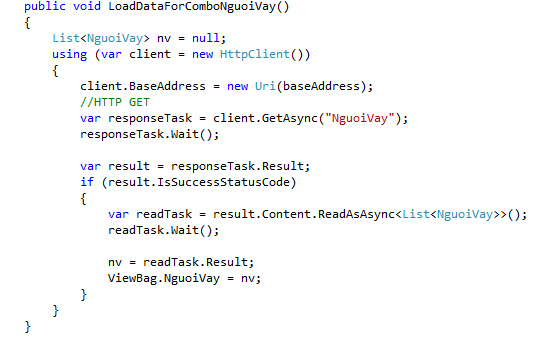
- Hàm Sửa khoản vay



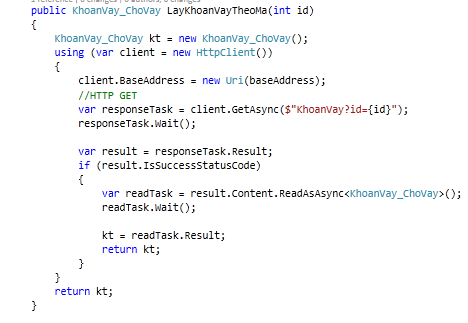
- Hàm load dữ liệu ra combobox loại tài khoản



- Hàm load dữ liệu ra combobox người cho vay



-Hàm lấy khoản vay theo mã



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện sửa khoản Vay

- Nếu người dùng nhập thông tin hợp lệ thì sửa thành công khoản Vay

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

**3.7.4. Xóa khoản vay**

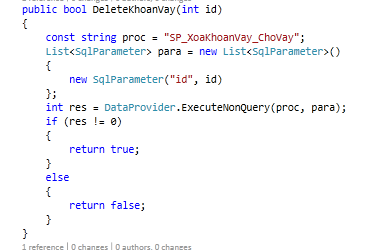
1. Ý tưởng giải quyết

- Người dùng chọn khoản vay cần xóa. Phía server nhận yêu cầu, và xác nhận xóa

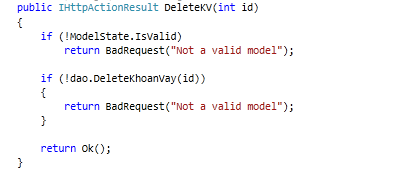
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xóa khoản vay



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm Xóa khoản vay



3. Giao diện

4. Kết quả chạy

- Hiện ra thông báo xác nhận xóa hay không

-Nếu chọn có xóa thành công hiện ra danh sách khoản vay

-Nếu chọn không hủy bỏ xóa

**3.7.5. Xem khoản cho vay**

**3.7.5.1.Xem khoản cho vay tổng thể**

1. Ý tưởng giải quyết

- Người dùng bấm xem khoản cho vay. Phía server nhận yêu cầu, hiển thị tất cả khoản cho vay trong bảng khoản Vay\_Chovay trong cơ sở dữ liệu

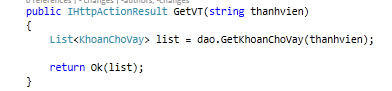
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản cho Vay

c

2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản cho vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản cho Vay

**3.7.5.2.Xem tổng khoản cho vay theo người vay**

1. Ý tưởng giải quyết

- Người dùng bấm xem khoản cho vay theo người vay. Phía server nhận yêu cầu, hiển thị tất cả khoản cho vay theo người vay mình trong cơ sở dữ liệu

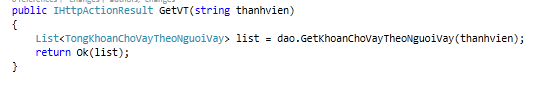
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản cho Vay theo người vay mình



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản cho vay theo người vay mình



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản cho Vay theo người vay mình

**3.7.5.3.Xem khoản cho vay chưa trả hết**

1. Ý tưởng giải quyết

- Người dùng bấm xem khoản cho vay chưa trả hết. Phía server nhận yêu cầu, hiển thị tất cả khoản cho vay chưa trả hết trong cơ sở dữ liệu

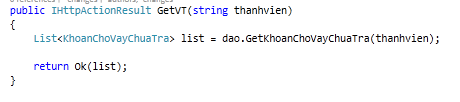
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản cho Vay chưa trả hết



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản cho vay chưa trả hết



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản cho Vay chưa trả hết

**3.7.5.4.Xem khoản cho vay đã chả hết**

1. Ý tưởng giải quyết

- Người dùng bấm xem khoản cho vay đã trả hết. Phía server nhận yêu cầu, hiển thị tất cả khoản cho vay đã trả hết trong cơ sở dữ liệu

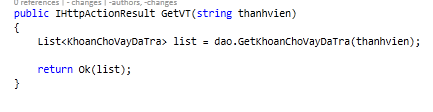
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem khoản cho Vay đã chả hết



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem khoản cho vay đã trả hết



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem khoản cho Vay đã trả hết

**3.7.6. Thêm khoản cho vay**

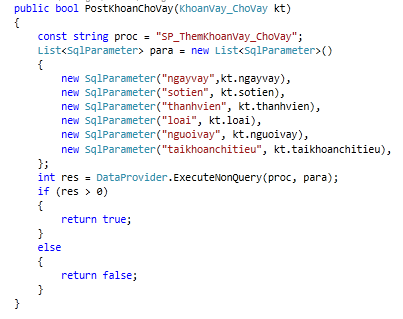
1. Ý tưởng giải quyết

- Người dùng nhập thông tin khoản cho vay cần thêm. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm khoản cho vay vào bảng khoản Vay\_Chovay trong cơ sở dữ liệu

2. Viết code trên các lớp

2.1. Lớp Data

Hàm thêm khoản cho Vay

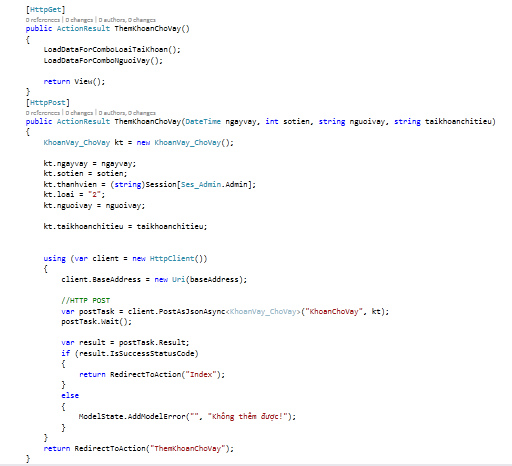


2.2. Lớp API

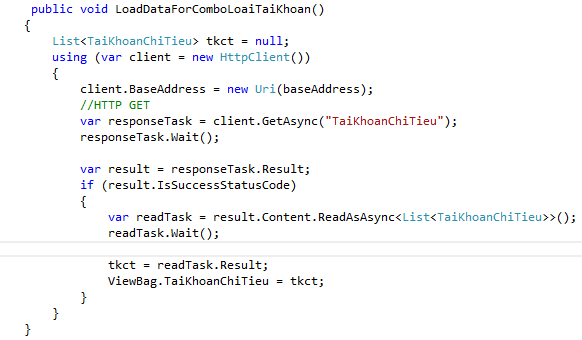


2.3. Lớp Web – Phía client

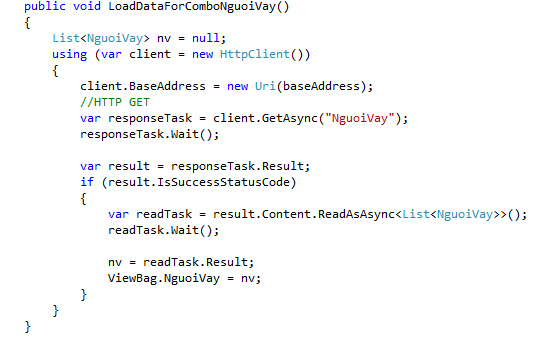
- Hàm thêm khoản cho vay



- Hàm load dữ liệu ra combobox loại tài khoản



- Hàm load dữ liệu ra combobox người vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện quản thêm mới khoản cho Vay

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công khoản cho Vay

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

**3.7.7. Sửa khoản cho vay**

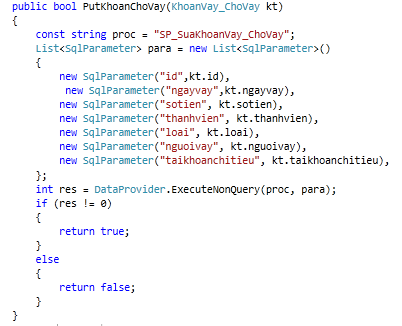
1. Ý tưởng giải quyết

- Người dùng bấm sửa khoản cho vay , sau đó nhập thông tin cần sửa. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành sửa khoản cho vay vào bảng khoản Vay\_Chovay trong cơ sở dữ liệu

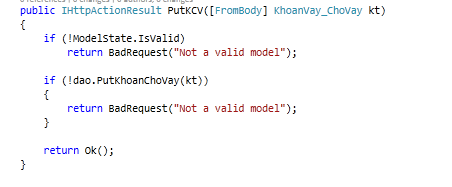
2. Viết code trên các lớp

2.1. Lớp Data

Hàm sửa khoản cho Vay



2.2. Lớp API

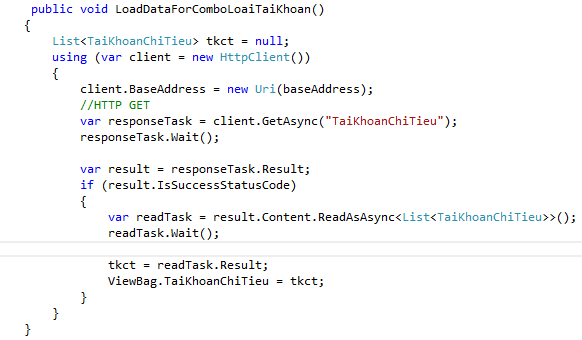


2.3. Lớp Web – Phía client

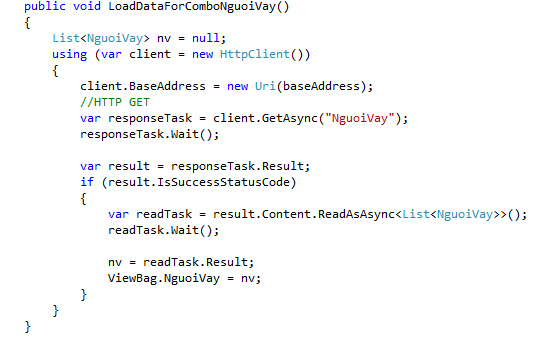
- Hàm Sửa khoản cho vay



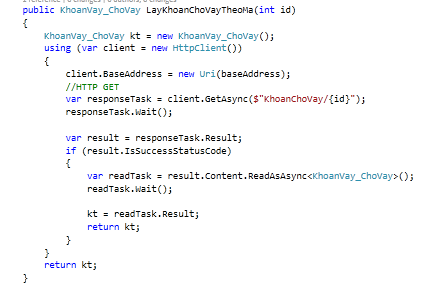
- Hàm load dữ liệu ra combobox loại tài khoản



- Hàm load dữ liệu ra combobox người cho vay



-Hàm lấy khoản cho vay theo mã



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện sửa khoản cho Vay

- Nếu người dùng nhập thông tin hợp lệ thì sửa thành công khoản cho Vay

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

**3.7.8. Xóa khoản cho vay**

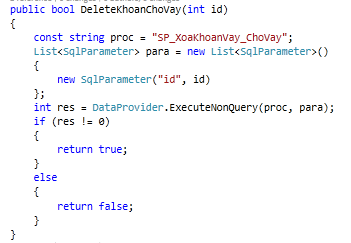
1. Ý tưởng giải quyết

- Người dùng chọn khoản cho vay cần xóa. Phía server nhận yêu cầu, và xác nhận xóa

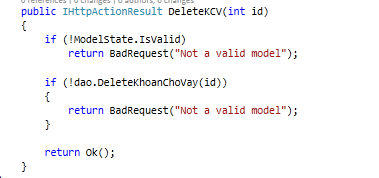
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xóa khoản cho vay

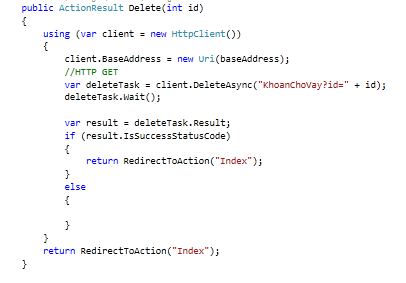


2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm Xóa khoản cho vay



3. Giao diện

4. Kết quả chạy

- Hiện ra thông báo xác nhận xóa hay không

-Nếu chọn có xóa thành công hiện ra danh sách khoản cho vay

-Nếu chọn không hủy bỏ xóa

**3.7.9. Xem người vay**

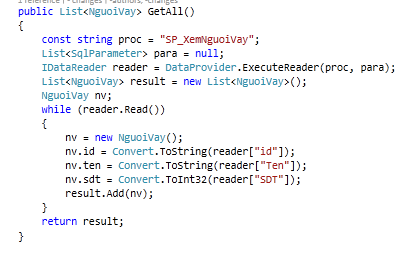
1. Ý tưởng giải quyết

- Người dùng bấm xem người vay. Phía server nhận yêu cầu, hiển thị tất cả người vay trong bảng khoản Nguoivay trong cơ sở dữ liệu

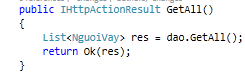
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xem người Vay



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm xem người vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện Xem người Vay

**3.7.10. Thêm người vay**

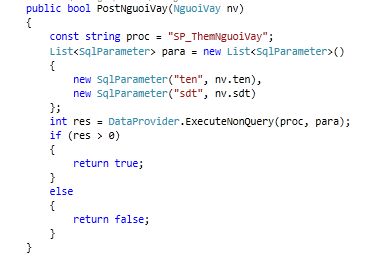
1. Ý tưởng giải quyết

- Người dùng nhập thông tin người vay cần thêm. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm người vay vào bảng người vay trong cơ sở dữ liệu

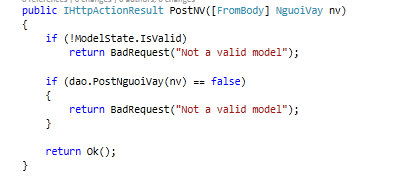
2. Viết code trên các lớp

2.1. Lớp Data

Hàm thêm người Vay



2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm thêm người vay



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện thêm mới Người Vay

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công người Vay

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

**3.7.11. Sửa người vay**

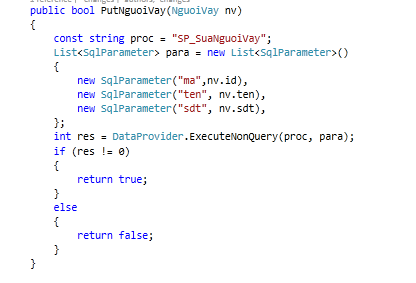
1. Ý tưởng giải quyết

- Người dùng bấm sửa người vay , sau đó nhập thông tin cần sửa. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành sửa người vay vào bảng NguoiVay trong cơ sở dữ liệu

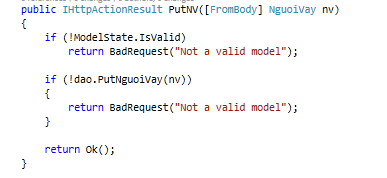
2. Viết code trên các lớp

2.1. Lớp Data

Hàm sửa Người Vay



2.2. Lớp API

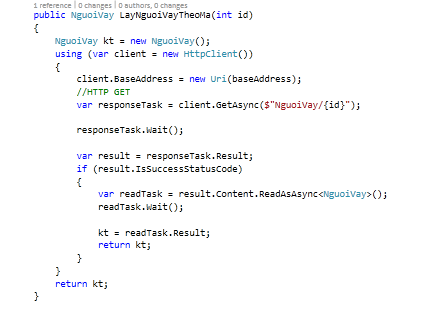


2.3. Lớp Web – Phía client

- Hàm sửa người vay



-Hàm lấy người vay theo mã



3. Giao diện

4. Kết quả chạy

- Hiện ra giao diện sửa Người Vay

- Nếu người dùng nhập thông tin hợp lệ thì sửa thành công Người Vay

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

**3.7.12. Xóa người vay**

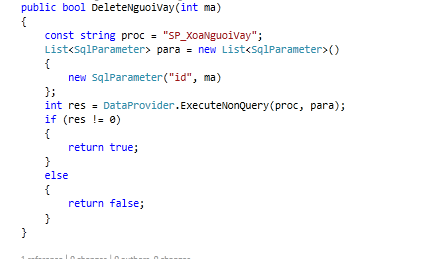
1. Ý tưởng giải quyết

- Người dùng chọn người vay cần xóa. Phía server nhận yêu cầu, và xác nhận xóa

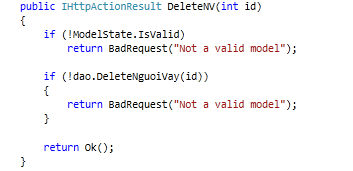
2. Viết code trên các lớp

2.1. Lớp Data

Hàm xóa người vay

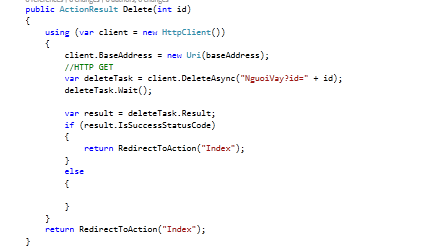


2.2. Lớp API



2.3. Lớp Web – Phía client

- Hàm Xóa người vay



3. Giao diện

4. Kết quả chạy

- Hiện ra thông báo xác nhận xóa hay không

-Nếu chọn có xóa thành công hiện ra danh sách người vay

-Nếu chọn không hủy bỏ xóa

**3.7.13. Trả nợ(Khoản vay(Khoản cho vay))**

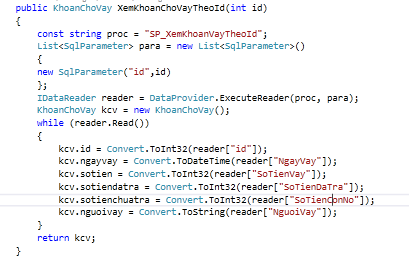
1. Ý tưởng giải quyết

- Người dùng bấm Trả nợ khoản vay(khoản cho vay). Phía server nhận yêu cầu, hiển thị thông tin khoản nợ(ngày vay,người cho vay, tổng số tiền nợ,tổng số tiền đã trả,tổng còn nợ) , Lịch sử các lần trả khoản nợ(ngày trả,số tiền trả) đó trong cơ sở dữ liệu, và Người dùng nhập thông tin cần trả nợ. Phía server nhận dữ liệu, nếu thông tin hợp lệ thì tiến hành thêm khoản trả nợ vào bảng trả nợ trong cơ sở dữ liệu

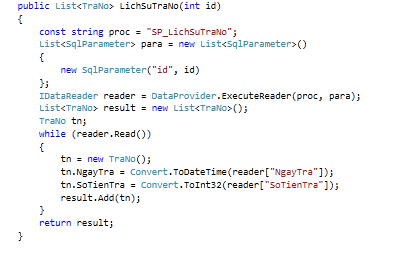
2. Viết code trên các lớp

2.1. Lớp Data

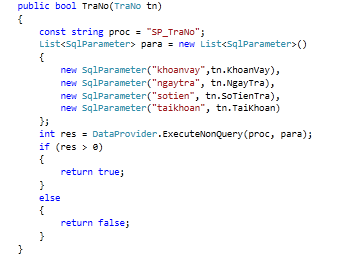
Hàm xem khoản nợ



Hàm lịch sử trả nợ

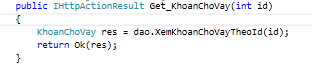


Hàm trả nợ cho khoản mình nợ

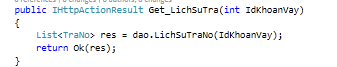


2.2. Lớp API

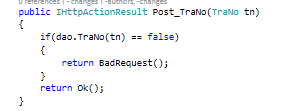
Hàm Xem khoản nợ



Hàm lịch sử trả nợ



Hàm Trả nợ

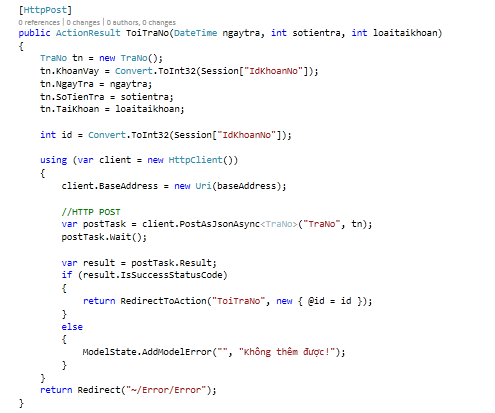


2.3. Lớp Web – Phía client

- Hàm xem khoản vay mình nợ



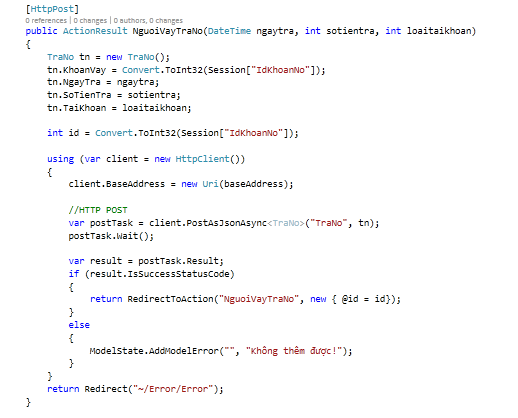
-Hàm mình trả nợ cho khoản vay



- Hàm xem khoản cho vay mà người vay đã nợ mình



-Hàm người vay trả nợ cho mình



-Hàm lịch sử trả nợ



3. Giao diện

-Người vay trả nợ

-Tôi trả nợ

4. Kết quả chạy

- Hiện ra giao diện thông tin khoản nợ + Lịch sử các lần trả nợ+Trả nợ cho khoản nợ của Khoản Vay( Khoản Cho Vay)

- Nếu người dùng nhập thông tin hợp lệ thì thêm thành công khoản trả nợ

- Nếu thông tin sai hoặc thiếu thì người dùng phải nhập lại và nhập đủ

## **3.8 Tiết kiệm**

**3.8.1. Tiết kiệm:**

1. Ý tưởng giải quyết:

- Hiển thị danh sách các khoản tiết kiệm bao gồm: ngày thêm, số tiền, mục đích

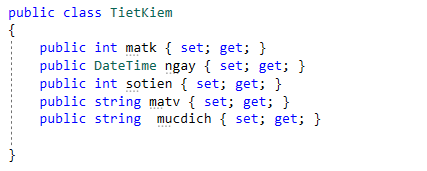
-Người dùng nhập thông tin cần thêm khoản tiết kiệm.Phía server nhận dữ liệu,nếu thông tin hợp lệ thì tiến hành thêm khoản tiết kiệm vào bảng tiết kiệm trong cơ sở dữ liệu

-Người dùng chọn nút sửa,hệ thống hiển thị form tiết kiệm.Người dùng nhập thông tin cần sửa. Phía server nhận dữ liệu,nếu thông tin hợp lệ thì tiến hành cập nhật khoản tiết kiệm vào bảng tiết kiệm trong cơ sở dữ liệu

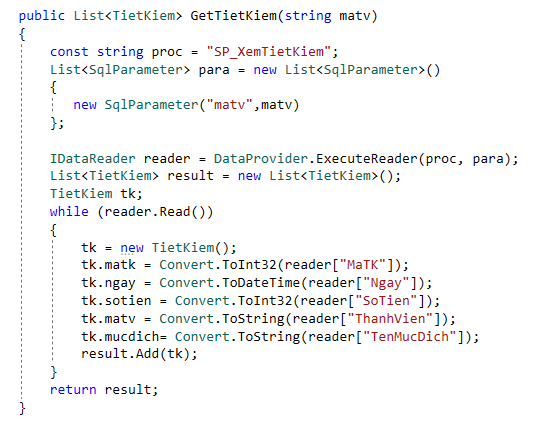
-Người dùng chọn xóa, hệ thống hiển thị cảnh báo hỏi người dùng có muốn xóa không.Người dùng nhấn có, phía server tiến hành xóa khoản tiết kiệm từ bảng tiết kiệm trong cơ sở dữ liệu

2. Viết code trên các lớp

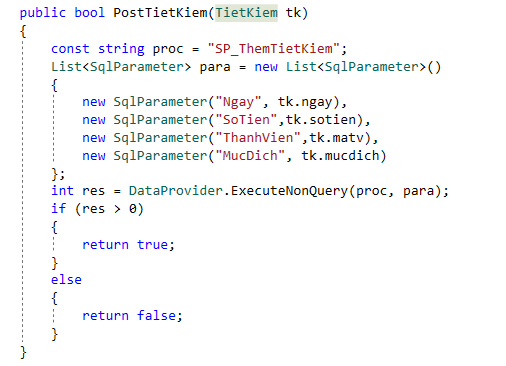
2.1. Lớp Data



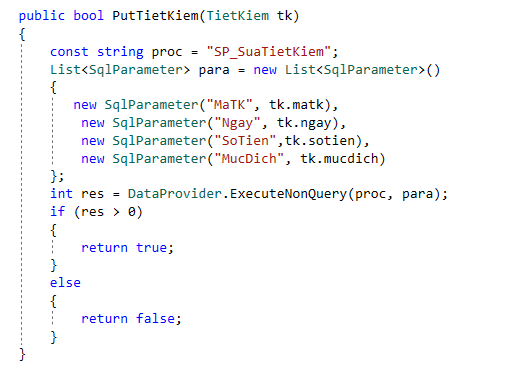
Hàm lấy ra danh sách tiết kiệm



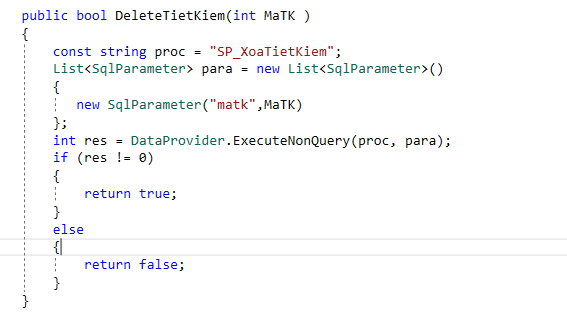
Hàm thêm tiết kiệm:



Hàm sửa tiết kiệm:

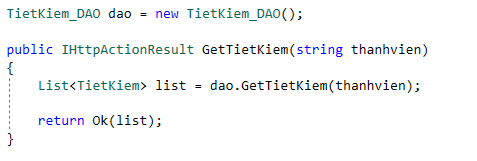


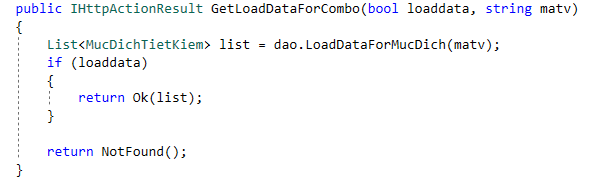
Hàm xóa tiết kiệm:



2.2. Lớp API

Hàm xem tiết kiệm

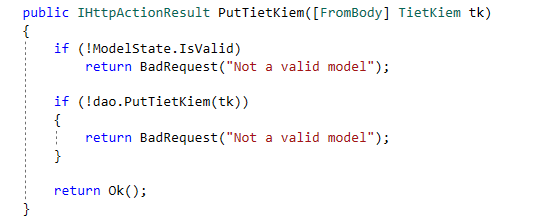




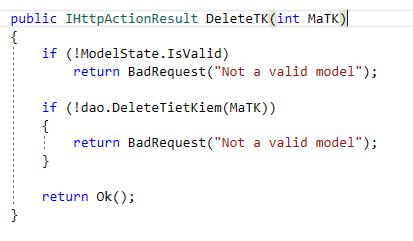
Hàm thêm tiết kiệm:



Hàm sửa tiết kiệm:



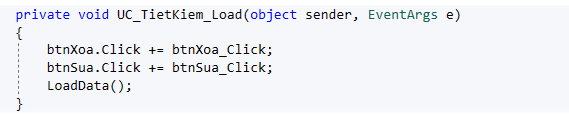
Hàm xóa tiết kiệm:



2.3. Lớp WF

- Hàm hiển thị danh sách khoản tiết kiệm:

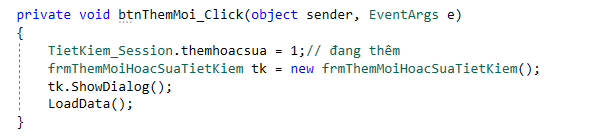


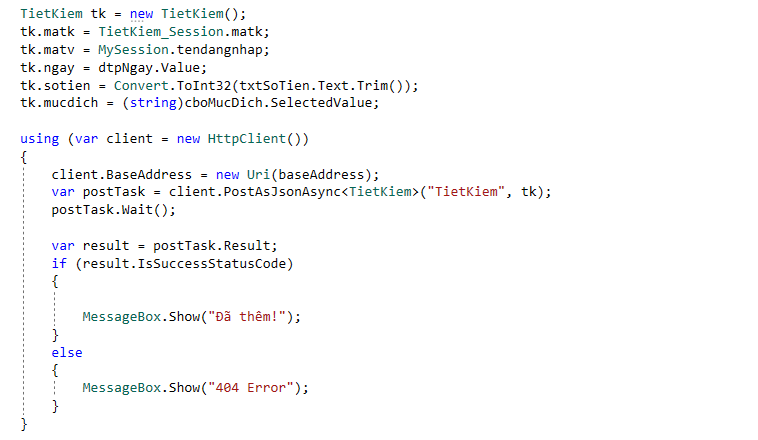


- Hàm load dữ liệu ra combobox mục đích:

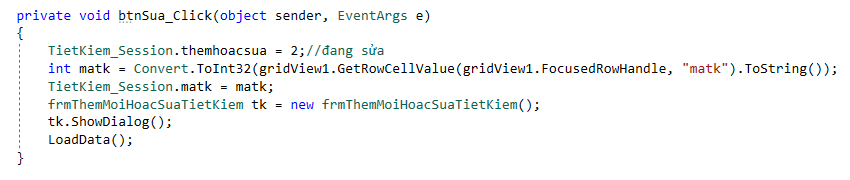


- Hàm thêm tiết kiệm:



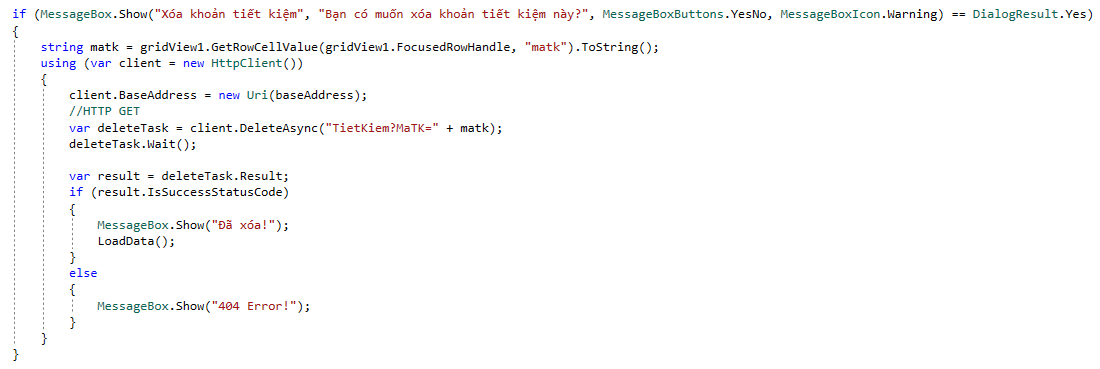


-Hàm sửa tiết kiệm:





-Hàm xóa tiết kiệm:

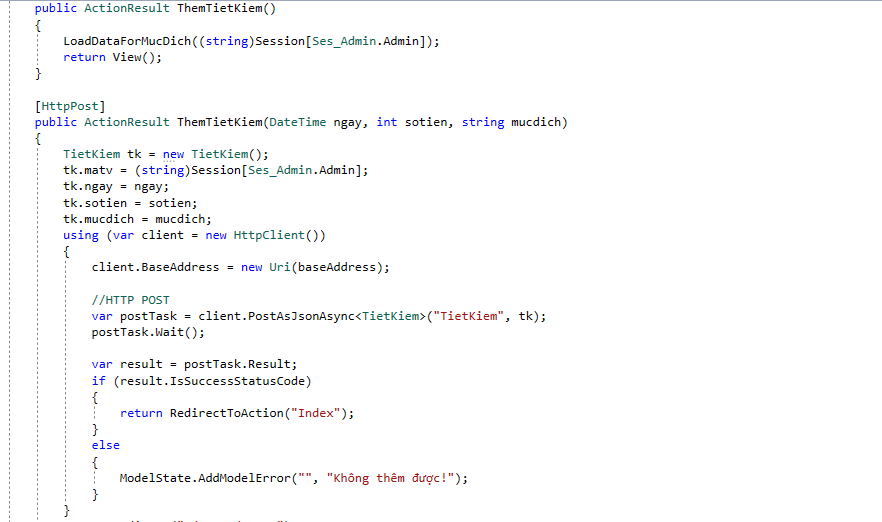


2.4.Lớp Web

-Hàm hiển thị danh sách tiết kiệm:



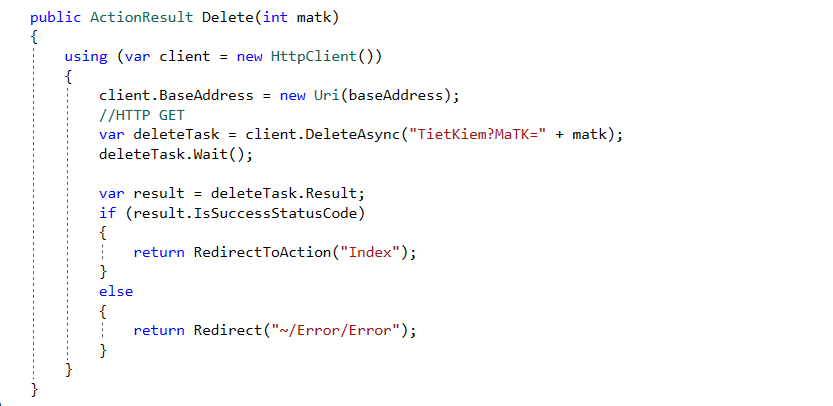
-Hàm thêm



-Hàm sửa



-Hàm xóa



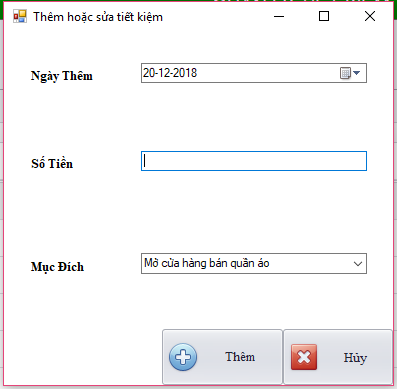
3. Giao diện

3.1.WinForm

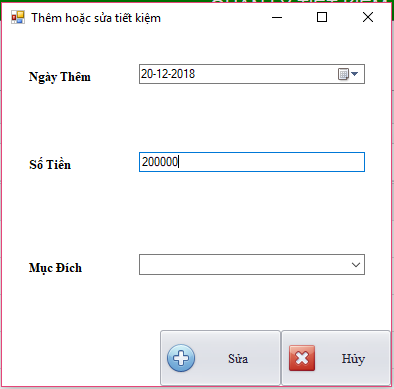
-Giao diện hiển thị danh sách tiết kiệm:



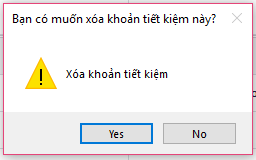
-Giao diện thêm tiết kiệm:



-Giao diện sửa tiết kiệm:



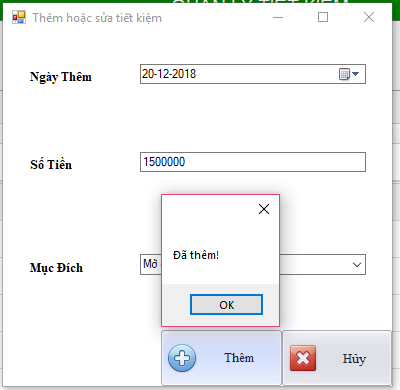
-Giao diện xóa:



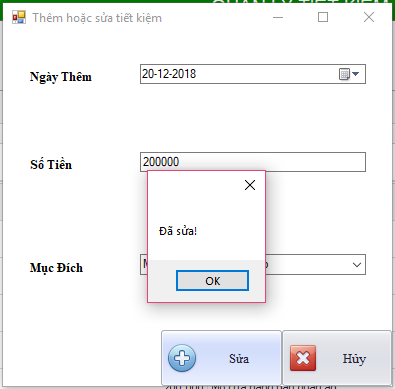
4. Kết quả chạy

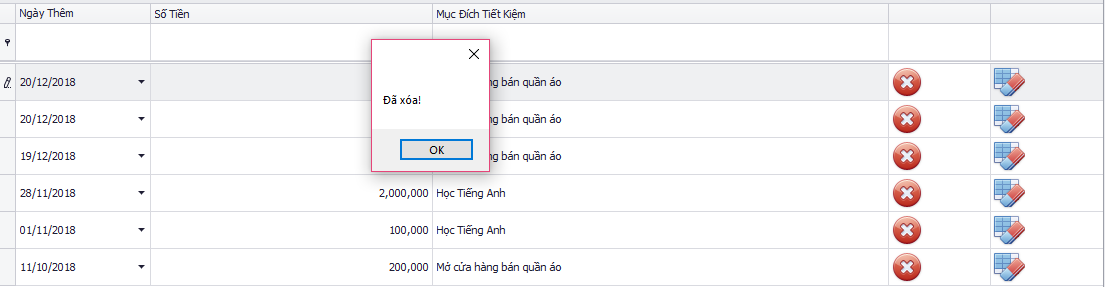
4.1. WinForm:

- Hiện ra giao diện thêm mới tiết kiệm



- Hiện ra giao diện sửa tiết kiệm



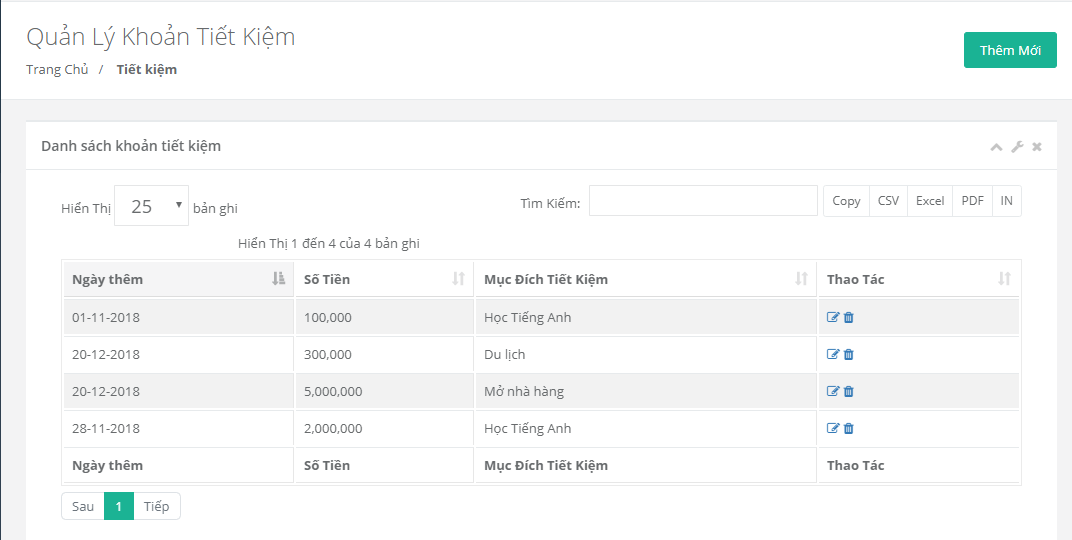
-

-Giao diện tìm kiếm:

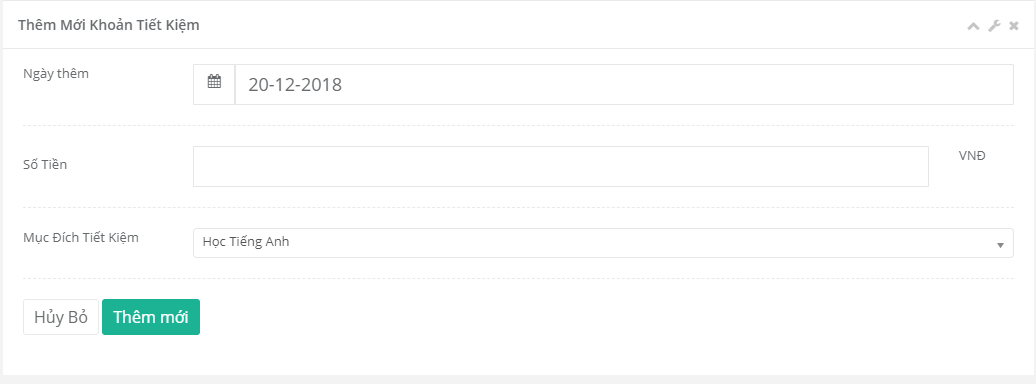


4.2. Web

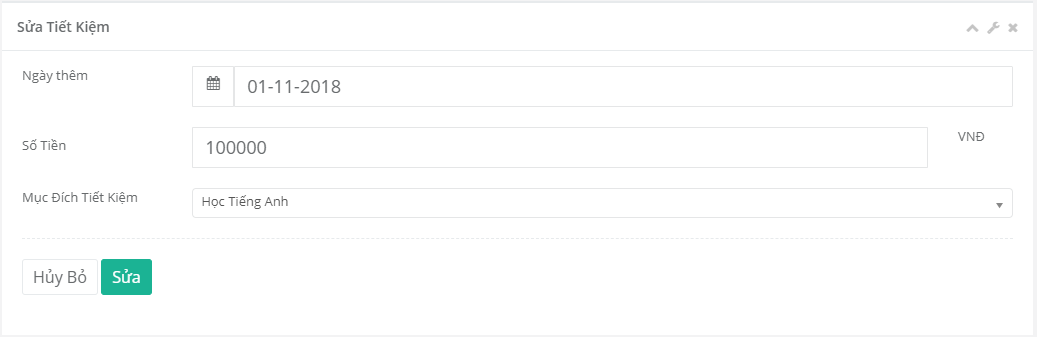
-Giao diện danh sách tiết kiệm:



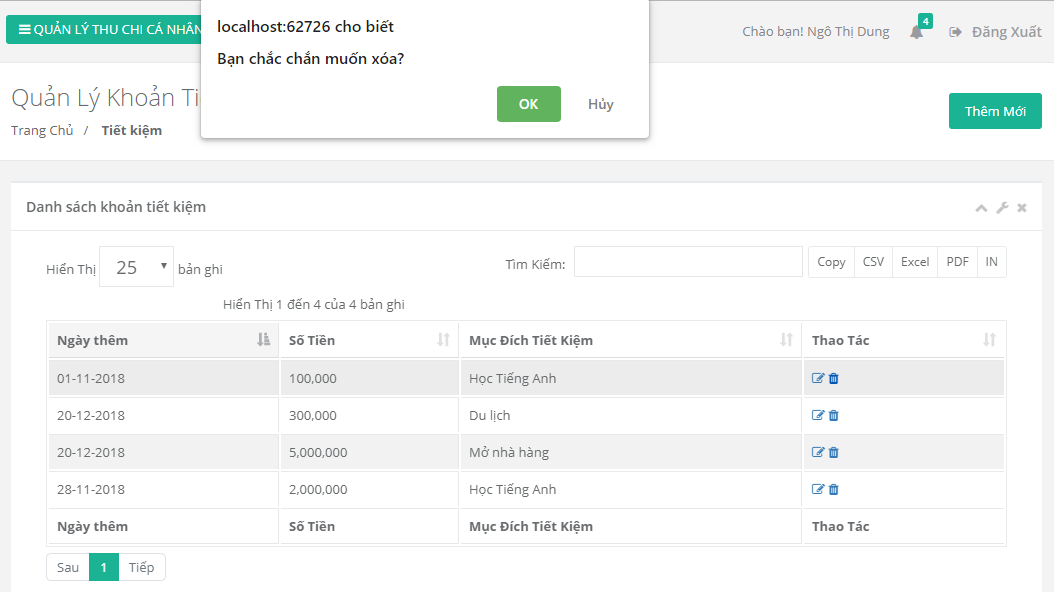
-Giao diện thêm tiết kiệm :



-Giao diện sửa:



-Giao diện xóa:



**3.8.2. Mục đích tiết kiệm:**

**1. Ý tưởng giải quyết:**

- Hiển thị danh sách các mục đích tiết kiệm bao gồm: tên mục đích, số tiền dự kiến, số tiền đã tiết kiệm, ngày bắt đầu, ngày kết thúc, trạng thái (đang tiết kiệm hay đã hoàn thành)

-Người dùng nhập thông tin cần thêm mục đích tiết kiệm.Phía server nhận dữ liệu,nếu thông tin hợp lệ thì tiến hành thêm mục đích tiết kiệm vào bảng mục đích tiết kiệm trong cơ sở dữ liệu

-Người dùng chọn nút sửa,hệ thống hiển thị form mục đích tiết kiệm.Người dùng nhập thông tin cần sửa. Phía server nhận dữ liệu,nếu thông tin hợp lệ thì tiến hành cập nhật mục đích tiết kiệm vào bảng mục đích tiết kiệm trong cơ sở dữ liệu

-Người dùng chọn xóa, hệ thống hiển thị cảnh báo hỏi người dùng có muốn xóa không.Người dùng nhấn có, phía server tiến hành xóa mục đích tiết kiệm từ bảng mục đích tiết kiệm trong cơ sở dữ liệu

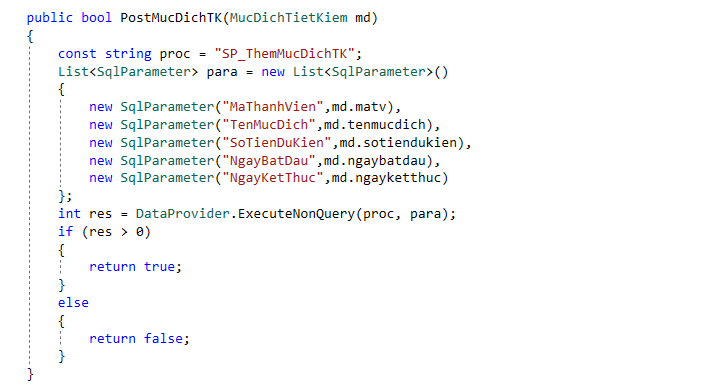
**2. Viết code trên các lớp**

2.1. Lớp Data

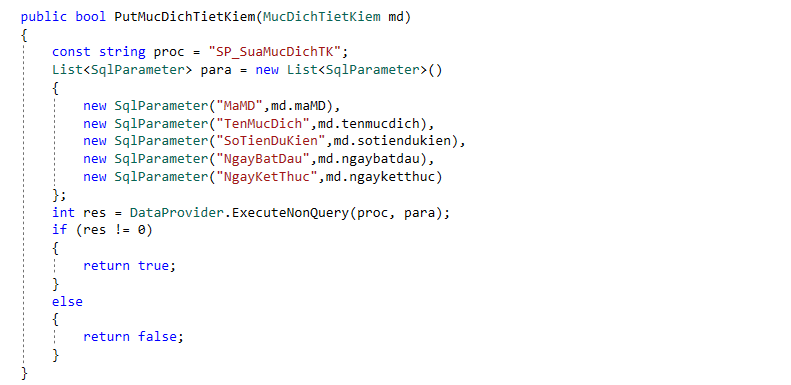
-Hàm lấy ra danh sách mục đích tiết kiệm:



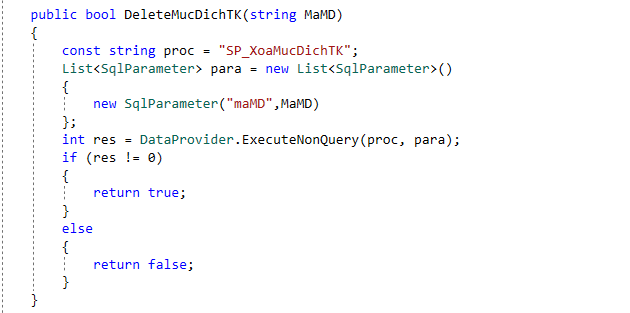
-Hàm thêm:



-Hàm sửa:

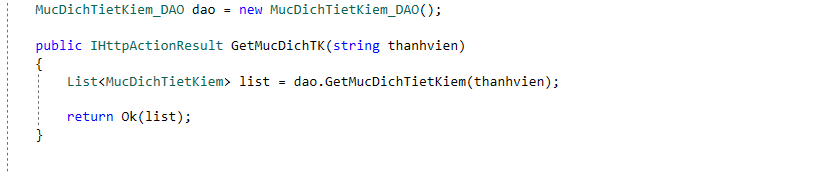


-Hàm xóa:

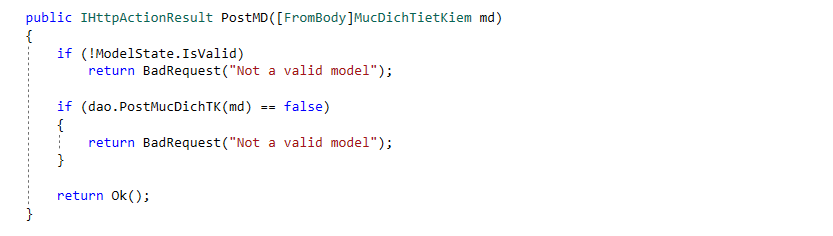


2.2.Lớp API

-Hàm lấy ra danh sách mục đích tiết kiệm:



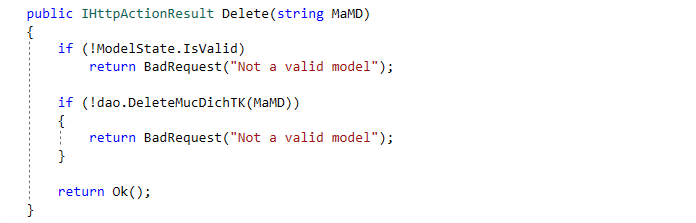
-Hàm thêm:



-Hàm sửa:



-Hàm xóa:

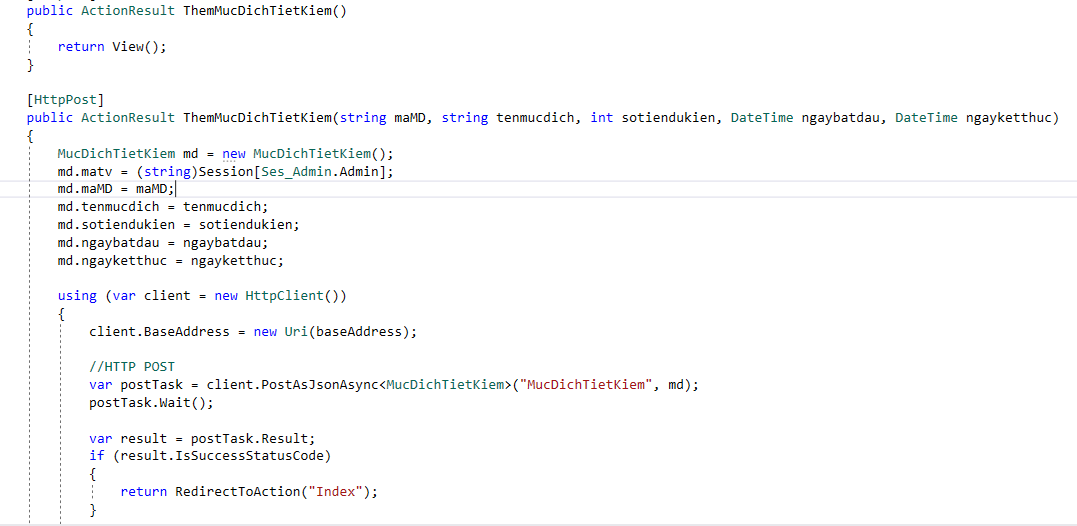


2.3.Lớp Web

-Hàm lấy ra danh sách mục đích tiết kiệm:



-Hàm thêm:



-Hàm sửa:



-Hàm xóa:



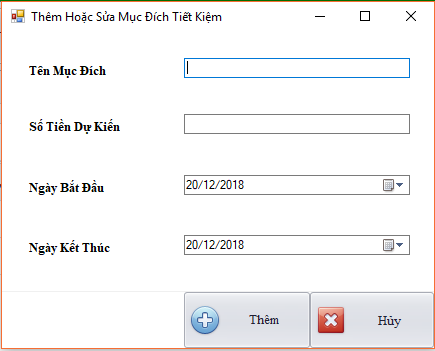
**3.Giao diện:**

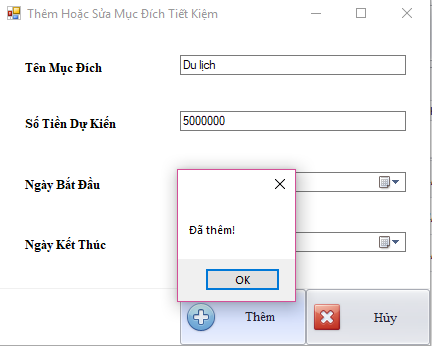
3.1.WinForm:

-Giao diện mục đích tiết kiệm:

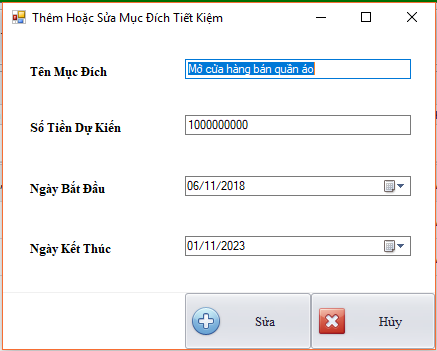


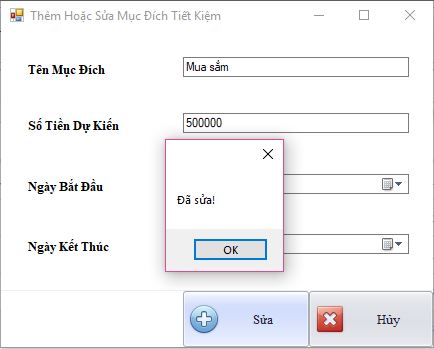
-Giao diện thêm mục đích:



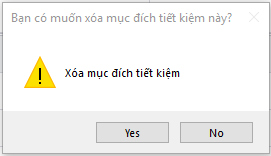


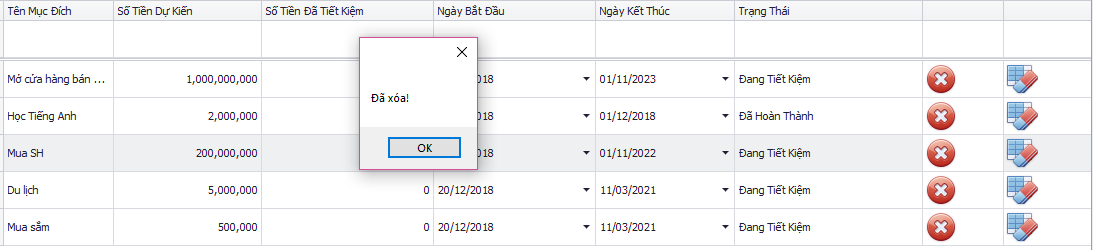
-Giao diện sửa mục đích:



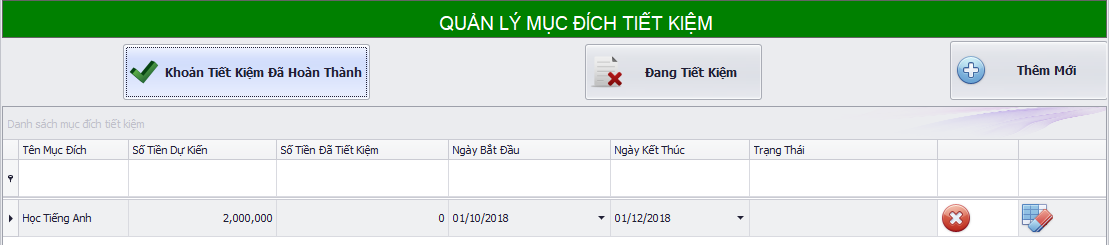


-Giao diện xóa mục đích:





Xem khoản tiết kiệm đã hoàn thành:

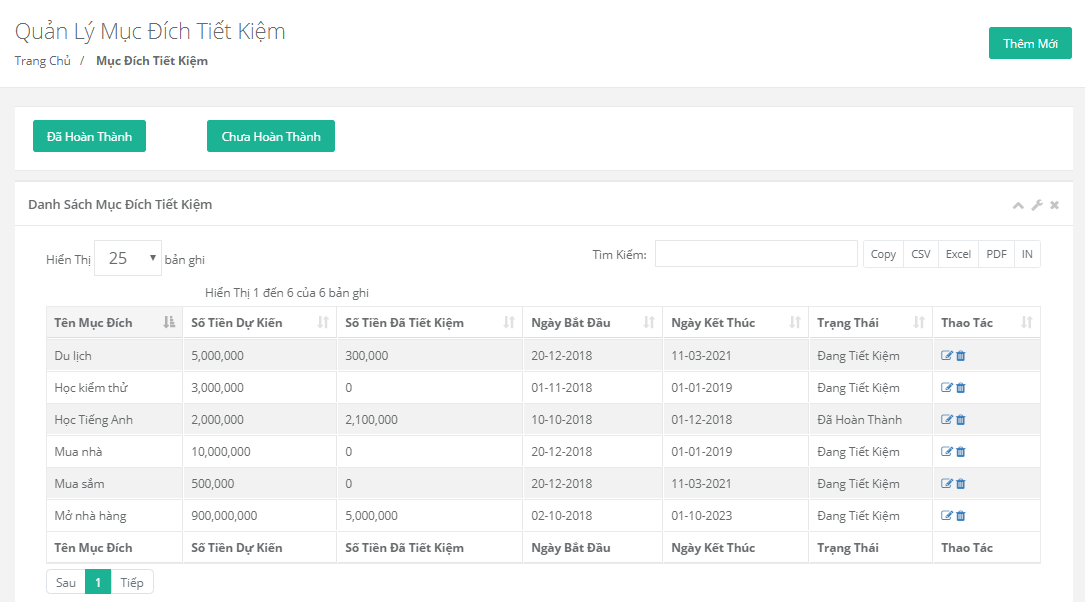


Xem khoản đang tiết kiệm:

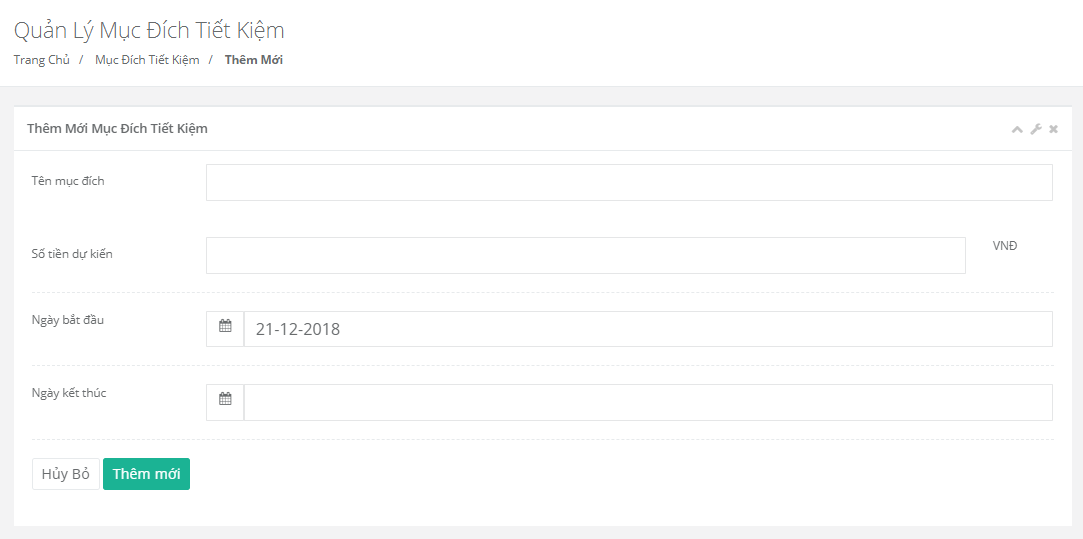


3.2. Web:

-Giao diện hiển thị danh sách mục đích tiết kiệm:



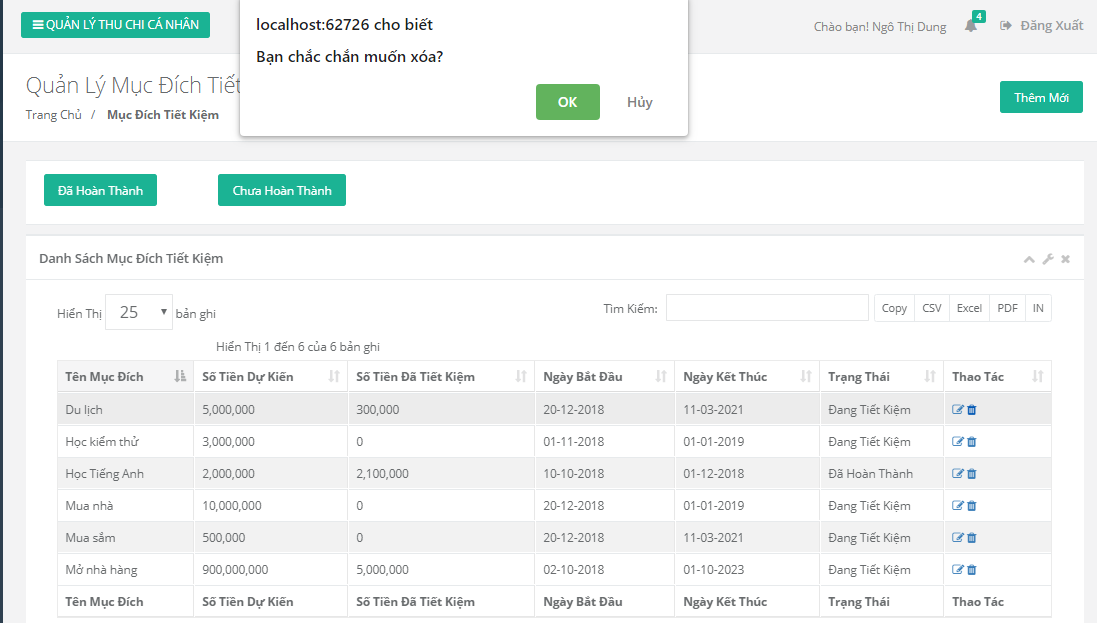
-Giao diện thêm:



-Giao diện sửa:



-Giao diện xóa:



## **3.9 Báo cáo**

3.9.1 Chức năng tài chính hiện tại

1. Ý tưởng giải quyết

- Chức năng tài chính hiện tại cho phép người dùng xem được số tiền trong ví, số tiền trong atm, số tiền cho vay, số tiền vay

- Ý tưởng giải quyết là: để tính số tiền hiện có trong ví thì ta cần truy vấn các bảng khoản thu, khoản chi, bảng vay, bảng cho vay, ta thiết lập được công thức như sau:

Đặt: tổng thu về cho vào ví = a

Tổng chi lấy từ ví = b

Tổng cho vay lấy tiền từ ví bằng c

Tổng vay rồi cho vào ví là d

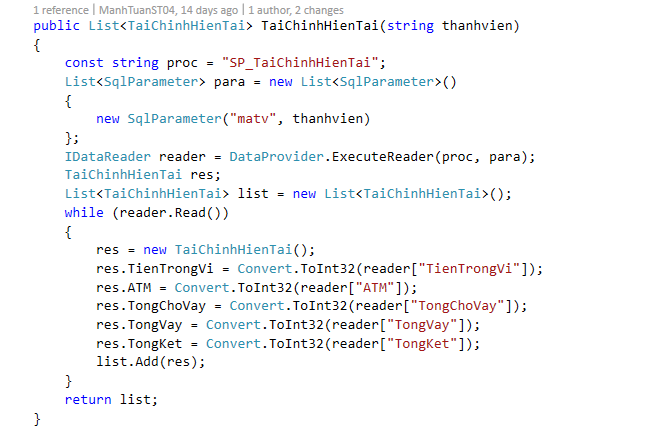
Vậy Tiền trong ví = a – b – c + d

- Tương tự ta tính được các khoản còn lại

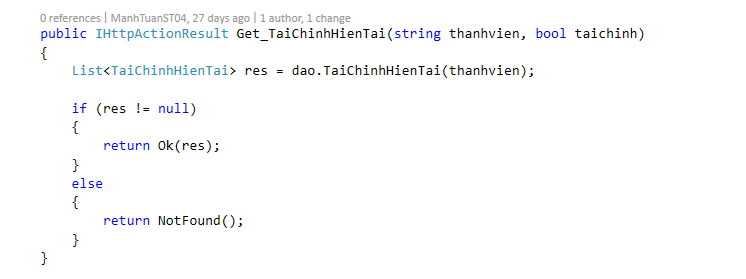
- Kết quả trả ra của thủ tục là số tiền trong ví, số tiền trong atm, tổng vay và tổng cho vay

2. Viết code trên các lớp

2.1. Lớp Data



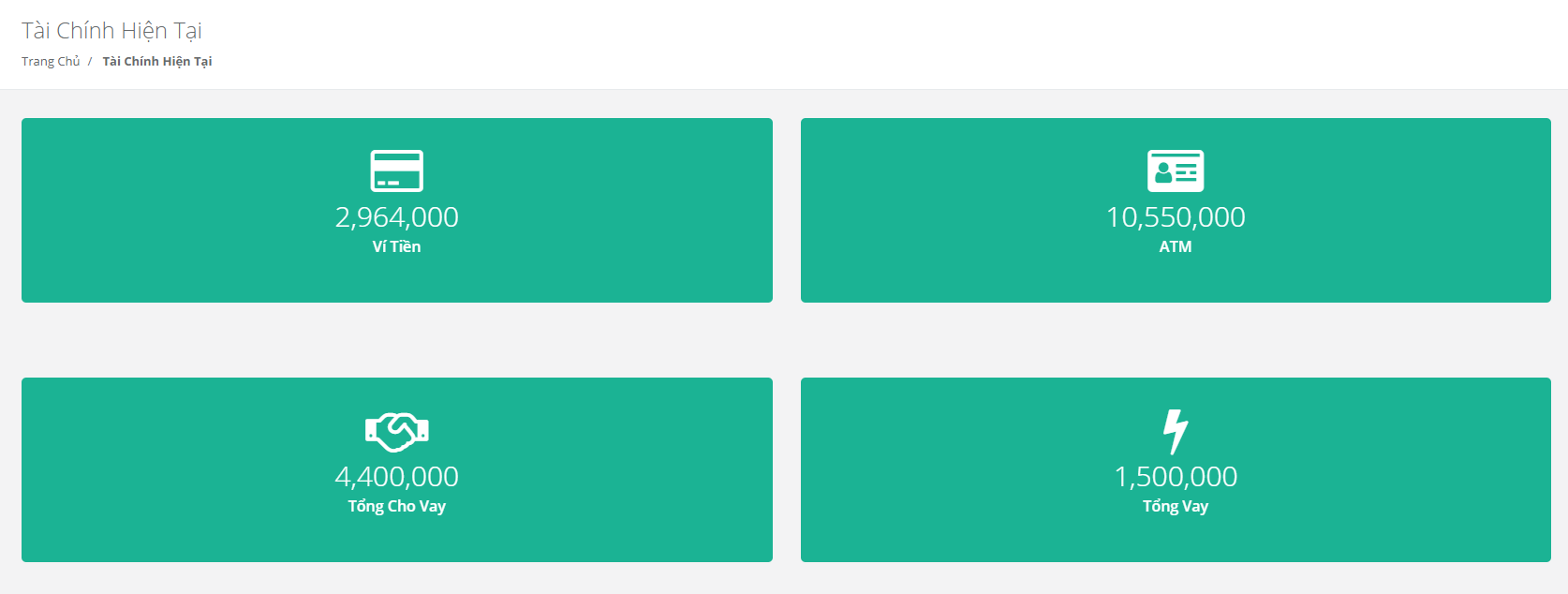
2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện



4. Kết quả chạy

- Hiện ra được số tiền trong ví, số tiền trong atm, tổng cho vay và tổng vay

3.9.2 Chức năng tình hình thu chi

1. Ý tưởng giải quyết

- Chức năng tình hình thu chi cho phép người dùng kiểm soát tổng thu chi trong ngày, tuần , tháng, năm.

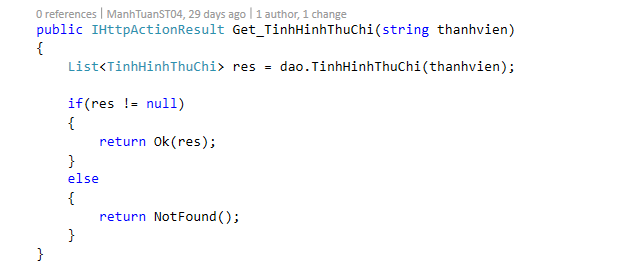
- Ý tưởng giải quyết: dựa vào bảng khoản thu và bảng khoản chi để tính ra tổng thu chi ứng với từng khoảng thời gian.

2. Viết code trên các lớp

2.1. Lớp Data



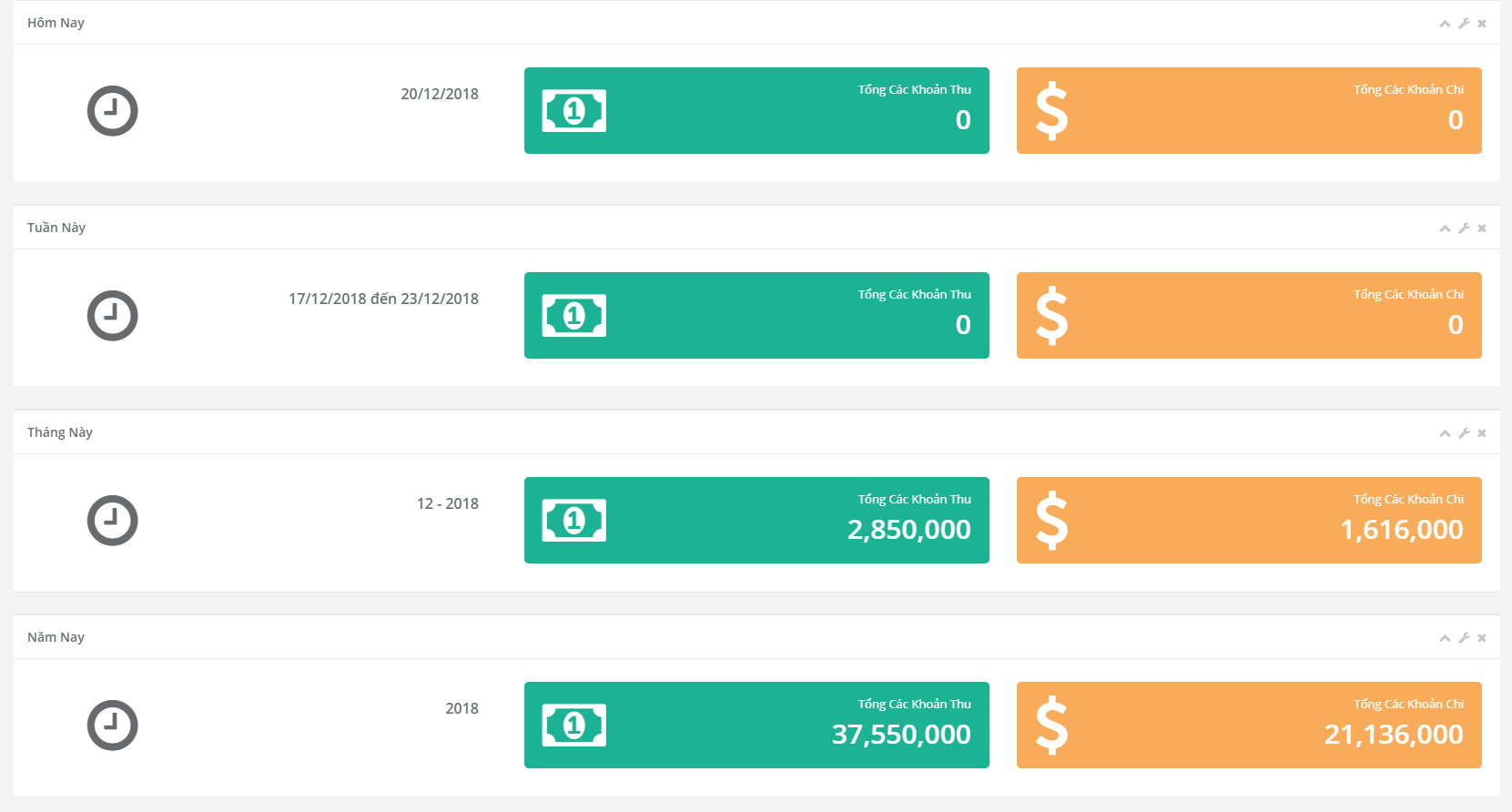
2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện



4. Kết quả chạy

- Đưa ra được tổng thu chi theo ngày, tuần, tháng, năm

3.9.3 Chức năng phân tích thu chi

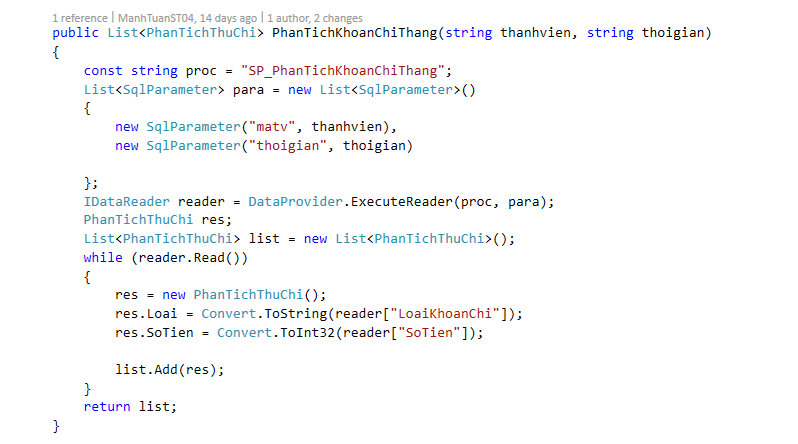
1. Ý tưởng giải quyết

- Chức năng phân tích thu chi cho phép người dùng xem tổng thu chi theo các loại khoản thu, loại khoản chi. Người dùng dễ dàng nắm được tháng này chi cho các loại khoản chi như ăn uống, mua sắm bao nhiêu…

- Ý tưởng giải quyết phân tích chi: từ bảng khoản chi, và bảng loại khoản chi, ta có thể tính được tổng chi theo từng loại khoản chi trong bảng loại khoản chi, sau đó đưa ra kết quả gồm loại khoản chi và tổng chi. Sau đó có thể tính ra phần trăm để hiện ra kết quả dạng biểu đồ tròn ở phía người dùng

2. Viết code trên các lớp

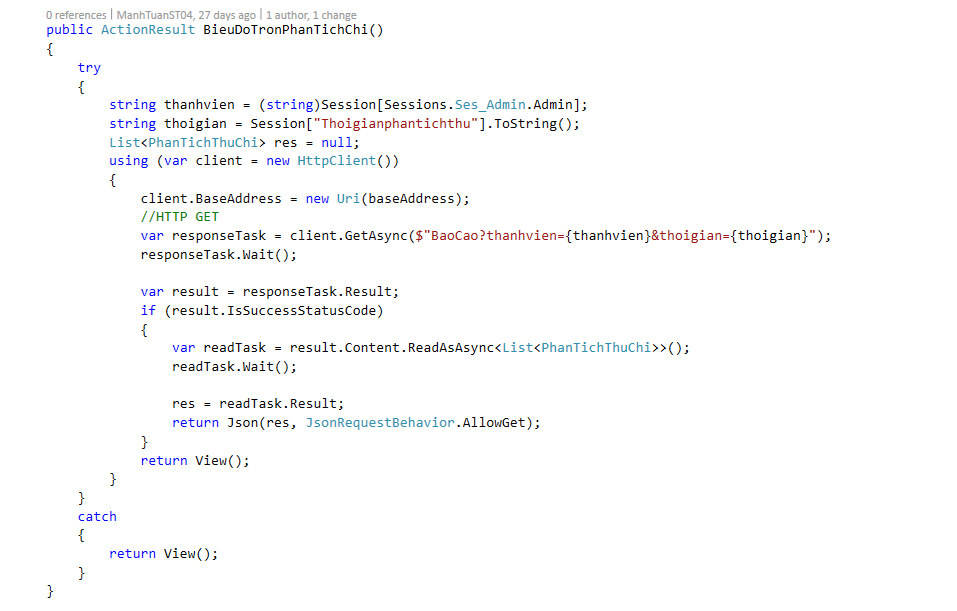
2.1. Lớp Data



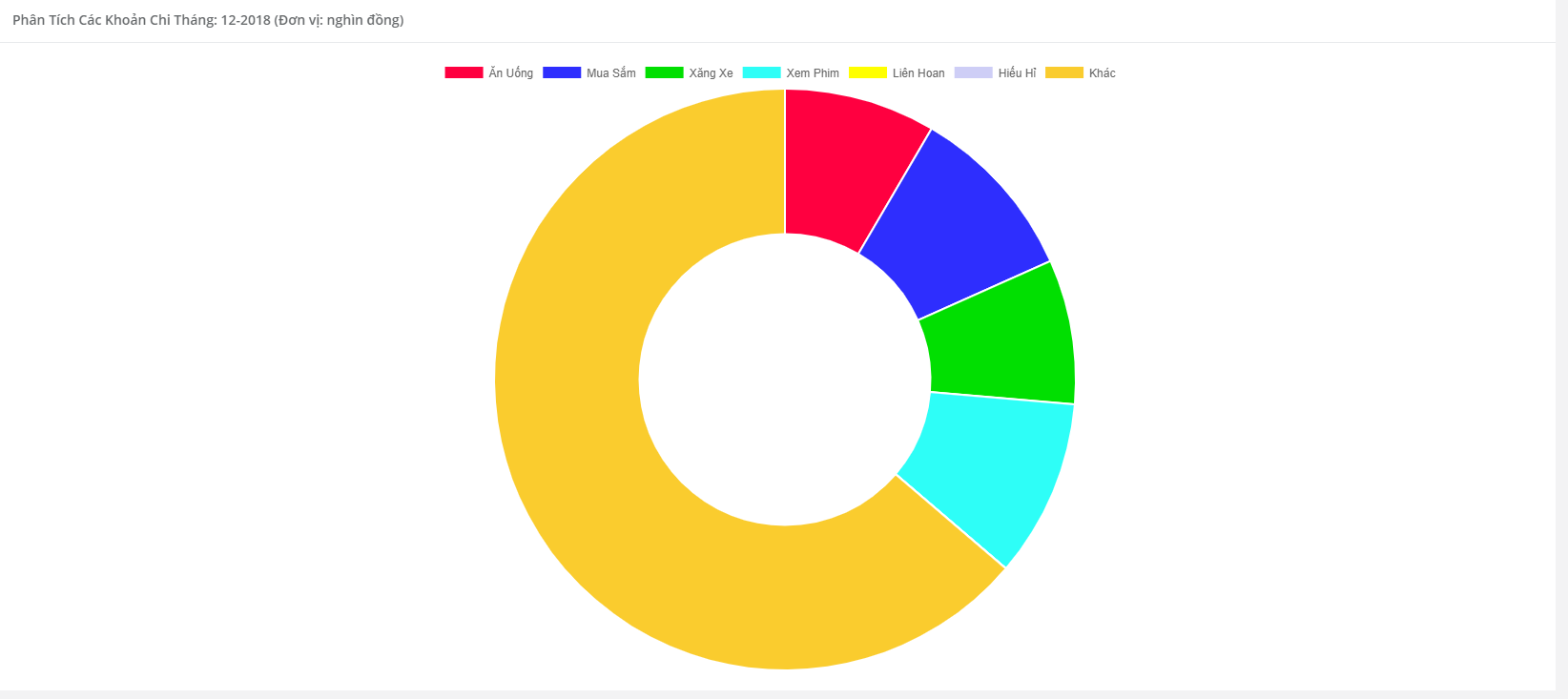
2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện



4. Kết quả chạy

- Đưa ra giao diện biểu đồ tròn, hiện các loại khoản thu và số tiền tương ứng

3.9.4 Thu chi theo năm

1. Ý tưởng giải quyết

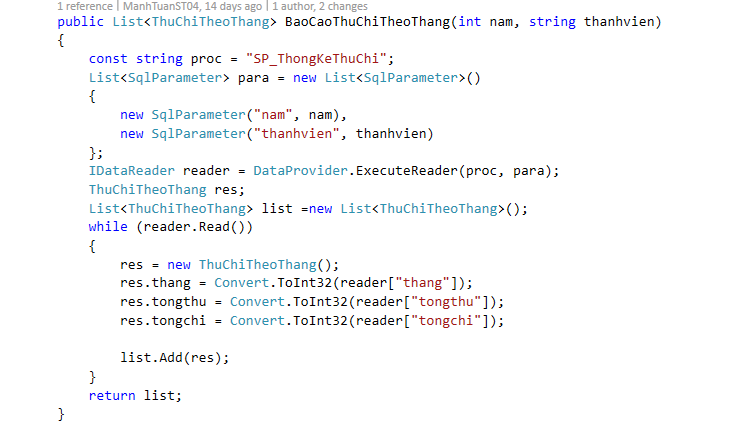
- Chức năng thu chi theo năm cho phép người dùng xem được tổng thu chi theo tháng trong năm hiện tại

- Ý tưởng giải quyết: từ bảng khoản thu và bảng khoản chi, ta có thể tính tổng thu theo từng tháng trong năm, bắt đầu từ tháng 1 đến tháng 12

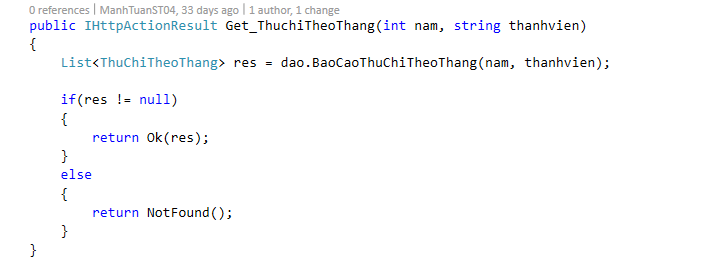
- Kết quả trả ra gồm 12 tháng trong năm và tổng thu, tổng chi từng tháng

2. Viết code trên các lớp

2.1. Lớp Data



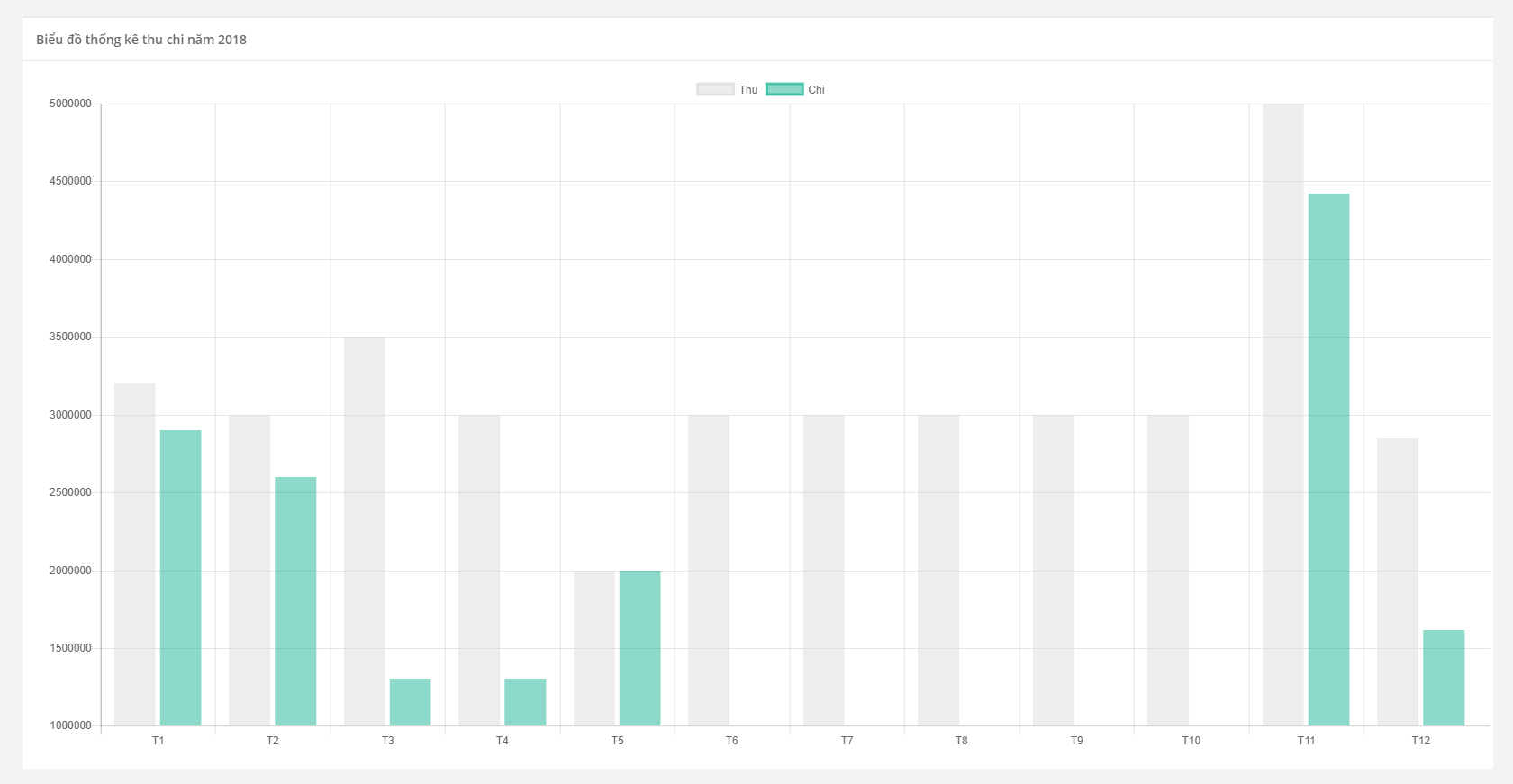
2.2. Lớp API



2.3. Lớp Web – Phía client



3. Giao diện

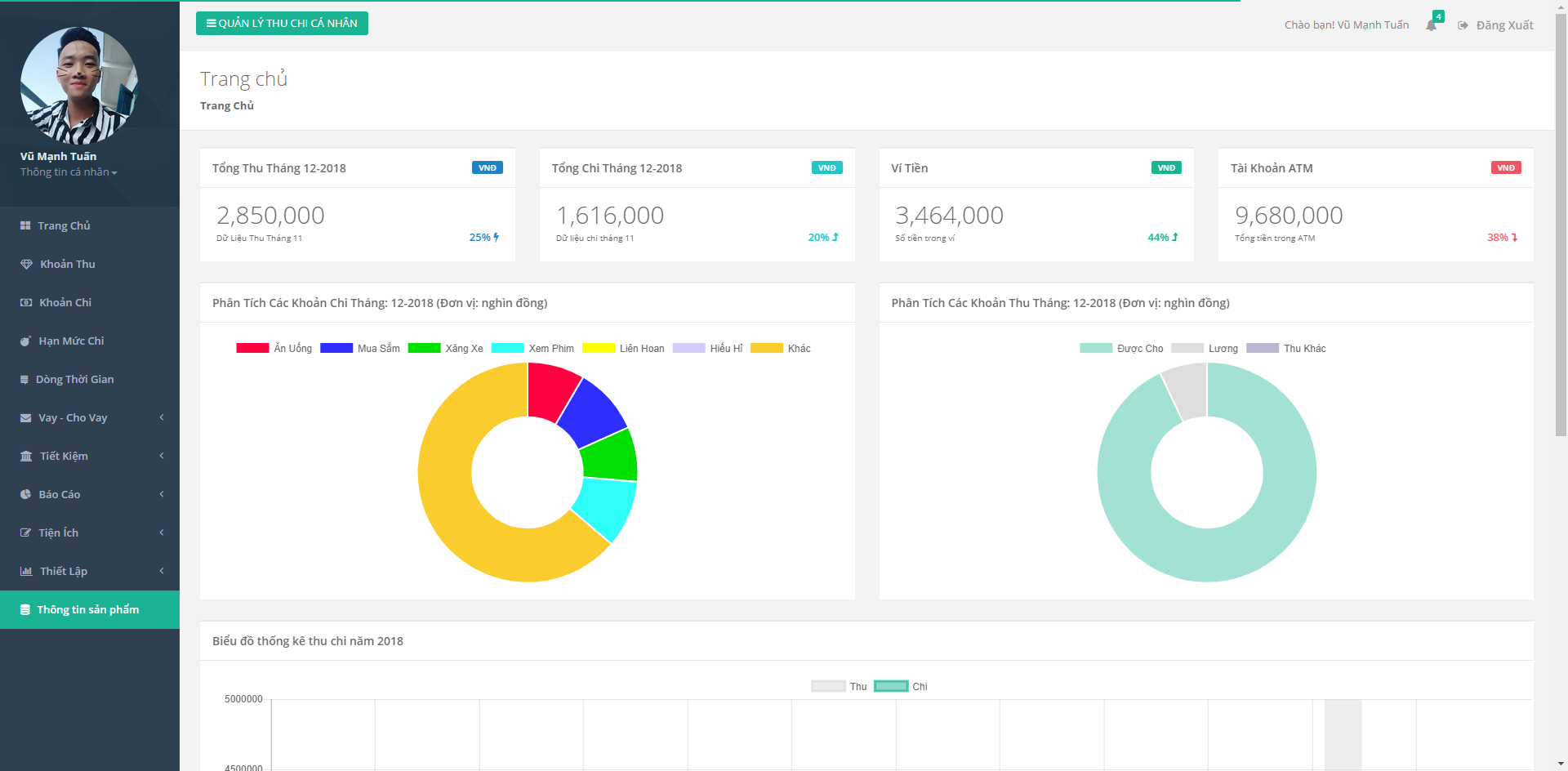


4. Kết quả chạy

- Đưa ra biểu đồ cột, hiển thị được tổng thu và tổng chi theo từng tháng

# **4. Giao diện chung của cả chương trình.**

- Nền tảng web



-Nền tảng windows form

