

TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN
VIỆN CÔNG NGHỆ THÔNG TIN KINH TẾ

----o0o----



CHUYÊN ĐỀ THỰC TẬP TỐT NGHIỆP

Đề tài:

**XÂY DỰNG WEBSITE QUẢN LÝ DANH MỤC ĐẦU TƯ
VÀ CHI TIÊU CÁ NHÂN DLYM**

<i>Giảng viên hướng dẫn</i>	: ThS. Phạm Thảo
<i>Họ và tên</i>	: Trần Quang Khải
<i>Mã sinh viên</i>	: 11146120
<i>Lớp</i>	: Công nghệ thông tin K56A

HÀ NỘI, 2018

MỤC LỤC

LỜI NÓI ĐẦU	1
CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI.....	3
1.1. Giới thiệu chung.....	3
1.2. Yêu cầu đặt ra.....	3
1.3. Mục tiêu của đề tài	4
1.4. Phạm vi đề tài.....	5
Chương II: GIỚI THIỆU VỀ CÁC CÔNG NGHỆ SỬ DỤNG.....	6
2.1. Công nghệ Web.....	6
2.1.1. ASP.NET MVC	6
2.1.2. Ngôn ngữ lập trình C#	13
2.2. Công nghệ xử lý giao diện	14
2.2.1. HTML (Hypertext Markup Language)	14
2.2.2. CSS (Cascading Style Sheets)	15
2.2.3. JavaScript.....	15
2.3. Một số Third – Party	16
Chương III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	19
3.1. Mô tả hệ thống	19
3.1.1. Sơ đồ kiến trúc hệ thống	19
3.1.2. Sơ đồ chức năng của hệ thống	20
3.1.3. Chức năng đối với khách hàng.	20
3.1.4. Chức năng đối với quản trị Website	22
3.2. Phân tích hệ thống về mặt chức năng.....	23
3.2.1. Biểu đồ luồng dữ liệu (BLD) mức khung cảnh	23
3.2.2. Biểu đồ luồng dữ liệu (BLD) mức đỉnh.....	24
3.2.3. Biểu đồ luồng dữ liệu (BLD) mức dưới đỉnh.....	25
3.3. Phân tích hệ thống về cơ sở dữ liệu	27
3.3.1. Xác định các thực thể.....	27
3.3.2. Sơ đồ thực thể liên kết	32
3.3.3. Mô hình quan hệ giữa các thực thể.....	36
3.3.4. Danh sách các bảng dữ liệu	36
3.3.5. Mô tả chi tiết các bảng.....	37
3.4. Giới thiệu một số sơ đồ thuật toán sử dụng trong Website.....	45
3.4.1. Sơ đồ thuật toán đăng nhập.....	45
3.4.2. Sơ đồ thuật toán chức năng Kế hoạch chi tiêu.....	45

3.4.3. Sơ đồ thuật toán chức năng Chi tiêu dự định.....	47
Chương IV: GIỚI THIỆU MỘT SỐ GIAO DIỆN CHÍNH CỦA WEBSITE.....	49
4.1. Giới thiệu một số giao diện chính của Website	49
4.1.1. Giao diện trang chủ khách vãng lai.....	49
KẾT LUẬN	52
TÀI LIỆU THAM KHẢO.....	53

LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn sâu sắc tới các thầy cô giáo trong bộ môn Công nghệ thông tin kinh tế trường Đại học Kinh Tế Quốc Dân đã tận tình giảng dạy và chỉ bảo, truyền đạt cho em những kiến thức bổ ích về chuyên ngành công nghệ thông tin. Giúp em nâng cao được hiểu biết và kỹ năng nghề nghiệp cho tương lai trong thời gian học tập vừa qua.

Em xin được gửi lời cảm ơn đến Thầy giáo ThS. Phạm Thảo đã tận tình giúp đỡ, hướng dẫn và chỉ dạy cho em trong suốt quá trình hoàn thành báo cáo thực tập chuyên đề. Bên cạnh những hiểu biết và kinh nghiệm còn quá ít của bản thân em, chắc chắn không thể tránh khỏi những thiếu sót khi thực hiện báo cáo của mình, kính mong thầy cô thông cảm. Sự góp ý của thầy cô sẽ là kinh nghiệm quý báu cho công việc của em sau này.

Kính chúc các thầy cô luôn khỏe mạnh, đạt được nhiều thành công trong sự nghiệp trồng người của mình.

Hà Nội, ngày 20 tháng 5 năm 2018

Sinh viên

Trần Quang Khải

LỜI NÓI ĐẦU

Cuộc cách mạng công nghệ 4.0 đang phủ sóng mạnh mẽ trên mọi lĩnh vực của đời sống. Áp dụng các thành tựu đột phá trong công nghệ thông tin để nâng cao chất lượng sản phẩm, tối ưu hóa sản xuất và phục vụ nhu cầu đời sống. Ứng dụng công nghệ thông tin được xem là một trong yếu tố quan trọng và cần thiết trong mọi hoạt động, tạo nên bước đột phá mạnh mẽ. Nó được xem là phát minh vĩ đại của loài người.

Áp dụng công nghệ vào các lĩnh vực đời sống đã không còn xa lạ đối với chúng ta. Nhờ đó, con người có thể dễ dàng kiểm soát các hoạt động, đạt được hiệu quả làm việc cao nhất. Đặc biệt là đối với các hoạt động quản lý chi tiêu và đầu tư, việc nắm bắt và kiểm soát là cực kỳ quan trọng. Nhu cầu được sử dụng một ứng dụng cho phép ghi chép, lưu trữ và thống kê các hoạt động chi tiêu và đầu tư là rất cần thiết trong bối cảnh các ứng dụng trên thị trường không đủ đáp ứng được các yêu cầu phức tạp của con người. Từ đó, em đã thực hiện đề tài “Xây dựng Website quản lý danh mục đầu tư và chi tiêu cá nhân” với mong muốn có thể đáp ứng được tốt nhất nhu cầu quản lý chi tiêu và đầu tư, định hướng chi tiêu và đầu tư cho khách hàng.

Nội dung chính của báo cáo bao gồm:

Chương I: Tổng quan về đề tài

Phát biểu đề tài, mục tiêu và mục đích hướng tới của đề tài.

Chương II: Giới thiệu về các công nghệ sử dụng

Giới thiệu các công nghệ được sử dụng để thực hiện triển khai đề tài.

Chương III: Phân tích thiết kế hệ thống

Áp dụng phương pháp phân tích và thiết kế hệ thống. Áp dụng phương pháp phân tích để đưa ra các sơ đồ phân tích thiết kế hệ thống.

Áp dụng phương pháp phân tích và thiết kế cơ sở dữ liệu. Thiết kế mô hình thực thể liên kết, phân tích chi tiết các bảng dữ liệu.

Một số sơ đồ thuật toán chính được sử dụng trong hệ thống.

Chương IV: Giới thiệu một số giao diện chính của website

Giới thiệu một số giao diện chính hoạt động trong website.

Kết luận

Phân tích, đánh giá các ưu điểm, nhược điểm và hướng phát triển của chương trình.

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu chung

Quản lý tài chính cá nhân cũng được coi là một bước để làm giàu. Một trong những bí quyết làm giàu của các đại gia, tỷ phú là họ biết cách quản lý tài chính tốt. Nhưng làm sao để có thể quản lý tài chính một cách tốt nhất? Bạn yên tâm là đã có ứng dụng công nghệ thông tin giúp bạn quản lý tài chính một cách trực quan và dễ dàng nhất. Không cần phải bỏ ra hàng giờ đồng hồ để thống kê các khoản thu chi và đầu tư, làm việc với một đống các con số lộn xộn nữa, mọi việc lưu trữ, tính toán và thống kê đều đều được xử lý trên hệ thống. Sự ra đời của Website Quản lý danh mục đầu tư và chi tiêu cá nhân DLYM sẽ làm thay đổi thói quen chi tiêu của bạn, giúp bạn có cái nhìn trực quan về tình hình tài chính cũng như cùng bạn tối đa hóa tài chính, khắc phục những hạn chế mà các ứng dụng quản lý tài chính cá nhân trên thị trường còn chưa hoàn thiện.

Hệ thống Quản lý danh mục đầu tư và chi tiêu cá nhân DLYM là một mô hình dịch vụ giúp cho từng cá nhân có thể quản lý các hoạt động chi tiêu dùng hàng ngày, thu nhập hàng tháng, các khoản đầu tư tài chính, cùng với đó thiết lập các kế hoạch chi tiêu, các khoản chi tiêu cố định. DLYM giúp khách hàng có cái nhìn tổng quan về tình hình tài chính cá nhân qua các thống kê, báo cáo trực quan theo các khoảng thời gian và theo từng nhóm danh mục. Từ đó, giúp khách hàng tối ưu hóa các khoản chi tiêu và đầu tư tài chính.

Hệ thống DLYM được thực hiện bằng ngôn ngữ lập trình C# dựa trên ASP.NET MVC cùng hệ quản trị cơ sở dữ liệu SQL Server.

1.2. Yêu cầu đặt ra

Bài toán đặt ra ở đây là xây dựng một Website quản lý các hoạt động tài chính (chi tiêu, thu nhập và đầu tư) giúp cho từng khách hàng có thể dễ dàng lưu trữ, xử lý, thống kê các hoạt động tài chính và định hướng chi tiêu, đầu tư. Do đó, các yêu cầu đặt ra:

- Lưu trữ các thông tin về hoạt động chi tiêu, thu nhập, đầu tư
- Lập định mức kế hoạch chi tiêu
- Lập hóa đơn

- Thông tin về các danh mục đầu tư
- Báo cáo, thống kê cho khách hàng
- Thống kê cho nhà quản trị

Ngoài chức năng trên, hệ thống còn thường xuyên gửi thông tin cảnh báo cho khách hàng qua email và thiết bị di động.

1.3. Mục tiêu của đề tài

Đề tài thực hiện nghiên cứu, xây dựng và phát triển một website quản lý danh mục đầu tư và chi tiêu cá nhân. Đối với từng đối tượng sử dụng, website có những mục đích khác nhau. Hệ thống cho phép từng khách hàng quản lý dữ liệu tài chính cá nhân, cho phép quản trị viên quản lý khách hàng, các danh mục hệ thống.

- Đối với đối tượng sử dụng là khách hàng

Với đối tượng là khách hàng – những người sử dụng website để lưu trữ, quản lý chi tiêu, thu nhập và đầu tư, website cần đảm bảo việc lưu trữ dữ liệu khách hàng là dễ dàng và bảo mật. Đưa ra cái nhìn tổng quan về tình hình tài chính của khách hàng theo nhiều tiêu chí.

Cung cấp cho khách hàng chức năng lập kế hoạch chi tiêu, định mức cho các khoản chi tiêu theo từng khoảng thời gian cụ thể. Lên lịch trình cho các khoản chi tiêu theo tần suất, thông báo tới khách hàng khi đến hạn chi tiêu.

Thông tin đến khách hàng về thị trường đầu tư mà khách hàng quan tâm. Theo dõi giá trị khoản đầu tư tại mọi thời điểm. Thông báo đến cho khách hàng khi thị trường có biến động.

Thống kê, báo cáo cho khách hàng về tình hình chi tiêu và đầu tư. Tỷ lệ các loại chi tiêu trong cấu thành nên tổng chi tiêu. Thống kê đa tiêu chí, giúp khách hàng dễ dàng nhận định về tình hình tài chính cá nhân.

- Đối với đối tượng sử dụng là quản trị viên

Quản trị viên là người quản lý và kiểm soát các hoạt động của hệ thống. Khi xây dựng website cần chú ý tạo điều kiện để quản trị viên có thể quản lý và kiểm soát các danh mục một cách trực quan và đơn giản nhất. Việc thêm, sửa, xóa thông tin danh mục phải đảm bảo được tính chặt chẽ và ràng buộc dữ liệu. Để thực hiện được điều đó, cơ sở dữ liệu phải được xây dựng hợp lý, cụ thể, rõ

ràng. Ràng buộc giữa các bảng dữ liệu là điều cần thiết và cơ sở dữ liệu phải được chuẩn hóa.

Thường xuyên cập nhật mới các loại tiền tệ, danh mục đầu tư, danh mục chi tiêu. Xuất bản tin tức và đề xuất giúp gợi ý tài chính cho khách hàng.

Ngoài các chức năng trên, điều quan trọng đối với website là tính bảo mật dữ liệu khách hàng. Tất cả các dữ liệu của khách hàng trên hệ thống cần được bảo đảm tính riêng tư và an toàn tuyệt đối. Đồng thời, cần xây dựng website dễ dàng bảo trì, nâng cấp và sửa chữa khi xảy ra sự cố hay có những điều chỉnh phù hợp.

1.4. Phạm vi đề tài

Với đề tài “Xây dựng Website Quản lý danh mục đầu tư và tài chính cá nhân DLYM”, Website cần thực hiện được một số chức năng sau:

- Lưu trữ thông tin chi tiêu, thu nhập và đầu tư cá nhân.
- Lập kế hoạch định mức chi tiêu.
- Thiết lập các khoản chi tiêu thường xuyên, theo tần suất định trước.
- Thông tin về thị trường đầu tư.
- Thống kê, báo cáo tình hình tài chính.
- Quản lý các danh mục chức năng của hệ thống.
- Quản lý API lấy dữ liệu từ đơn vị thứ 3.
- Quản lý tài khoản, phân quyền chức năng cho thành viên quản trị và khách hàng.

Chương II: GIỚI THIỆU VỀ CÁC CÔNG NGHỆ SỬ DỤNG

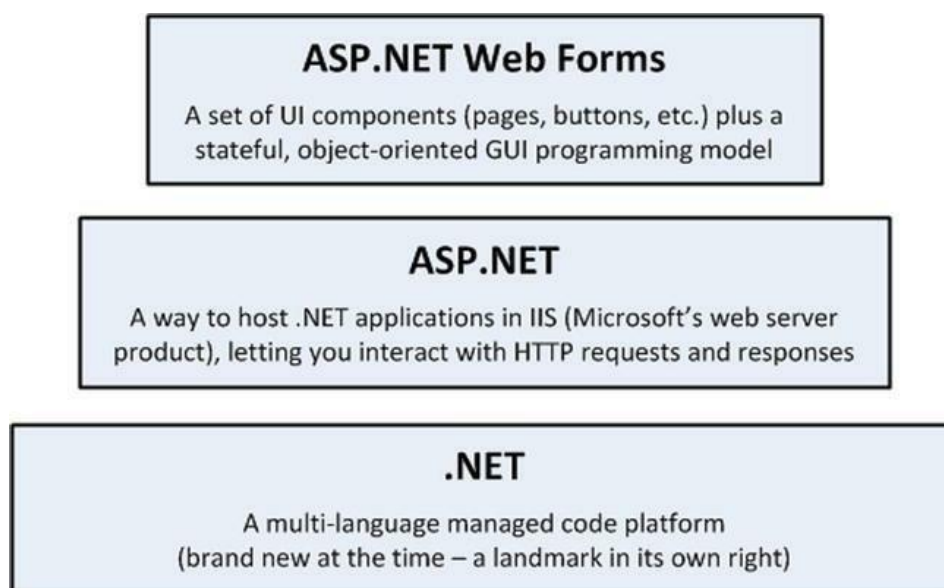
2.1. Công nghệ Web

2.1.1. ASP.NET MVC

ASP.NET MVC là một nền tảng phát triển Web của Microsoft kết hợp sự hiệu quả của mô hình cấu trúc model-view-controller (MVC), các ý tưởng mới nhất và các kỹ thuật phát triển linh hoạt, và những gì tốt nhất của nền tảng ASP.NET. Đây là một thay thế hoàn chỉnh cho các dạng Web Form ASP.NET truyền thống, đem đến nhiều ưu điểm nhưng chủ yếu dành cho các dự án phát triển Web.

2.1.1.1. ASP.NET

ASP.NET là một sự thay đổi to lớn khi được xuất hiện lần đầu tiên vào năm 2002.



Hình 2.1. Các tầng công nghệ của ASP.NET Web Forms

ASP.NET là một mô hình phát triển Web hợp nhất bao gồm các dịch vụ cần thiết để xây dựng lên một ứng dụng Web doanh nghiệp với số lượng các dòng code là nhỏ nhất. ASP.NET là một phần của .NET Framework. Các ứng dụng viết bằng ASP.NET cũng có thể truy cập các lớp trong .NET Framework. Người dùng có thể sử dụng bất kỳ ngôn ngữ nào tương thích với CLR (Common

Language Runtime) của .NET để viết chương trình, mà một số ngôn ngữ phổ biến có thể kể đến là C#, Visual Basic .NET, Jscript và J#.

ASP.NET là một nền tảng phát triển web cung cấp các mô hình lập trình, một cơ sở hạ tầng phần mềm toàn diện và đa dạng các dịch vụ để xây dựng nên các ứng dụng web cho máy tính cá nhân cũng như các thiết bị di động.

ASP.NET hoạt động trên giao thức HTTP, sử dụng các lệnh và chính sách HTTP để thiết lập một giao tiếp song song giữa trình duyệt và máy chủ. Nó được sử dụng để tạo nên các ứng dụng web tương tác, với một số lượng lớn các điều khiển như textbox, button, label để tạo nên một trang HTML một cách nhanh chóng.

ASP.NET cung cấp ba framework được sử dụng để xây dựng nên một ứng dụng web: ASP.NET Web Forms, ASP.NET MVC và ASP.NET Web Pages. Cả ba framework này đều khá ổn định, người dùng có thể tùy chọn một trong ba framework để dựng trang web của riêng mình. Mỗi framework hướng tới một loại đối tượng khác nhau. Tùy vào trải nghiệm người dùng mà lựa chọn framework phù hợp nhất với kiểu ứng dụng được tạo.

ASP.NET Web Forms là công nghệ phát triển Web đã có từ lâu đời và cho đến hiện tại, bản thân nó đã xuất hiện những hạn chế:

- + *Tầm quan trọng của View State*: Cơ chế thực cho việc duy trì trạng thái thông qua các yêu cầu (View State) hình thành một khối lượng dữ liệu được truyền phát giữa client và server. Dữ liệu này có thể lên tới hàng trăm kilobyte ngay cả trong những ứng dụng Web nhỏ, và nó di chuyển qua lại với mỗi một yêu cầu, dẫn đến làm chậm thời gian phản hồi và tăng nhu cầu băng thông của server.

- + *Chu kỳ tồn tại của trang*: Cơ chế kết nối các sự kiện bên phía client với bộ code quản lý sự kiện bên server, từng phần của chu kỳ tồn tại, có thể bất thường. Một số ít nhà phát triển đã vận dụng thành công hệ thống điều khiển theo bậc tại thời gian khởi chạy mà không gặp lỗi bên View State hay phát hiện bất cứ xử lý sự kiện gặp lỗi khi khởi chạy.

- + *Cảm giác về phân chia của các vấn đề sai*: mô hình mã code nằm phía sau của ASP.NET Web Forms cung cấp phương tiện để lấy phần code ứng dụng

ra khỏi ngôn ngữ đánh dấu trong HTML của nó và đưa vào một lớp code nền đằng sau. Kết quả cuối có thể hơi khó hiểu.

+ *Giới hạn kiểm soát trong HTML*: Các control của server biểu hiện chính nó dưới dạng HTML, nhưng không nhất thiết phải là HTML mà bạn muốn. Trong các phiên bản trước của ASP.NET, HTML đầu ra không đáp ứng tiêu chuẩn Web hay tận dụng tốt CSS (Cascading Style Sheets), các control của server được khởi tạo không lường trước và các giá trị thuộc tính ID phức tạp khó tiếp cận bằng Javascript. Những vấn đề đó được cải thiện phần lớn trong Web Form phát hành gần đây, nhưng vẫn khó khăn để có được HTML mà bạn mong muốn.

+ *Sự trừu tượng sai, lệch lạc*: Web Form cố giấu đi HTML và HTTP ở bất cứ nơi nào có thể. Khi bạn cố gắng cài đặt một hành vi tùy chỉnh, bạn thường đi xa khỏi tưởng tượng, buộc bạn phải đảo ngược cơ chế sự kiện hoặc thực hiện các hành vi khó hiểu để làm nó sinh ra mã HTML mong muốn. Thêm vào đó, tất cả khái niệm trừu tượng này có thể đóng vai trò như một rào cản gây khó chịu cho các nhà phát triển thông thạo.

+ *Khả năng nghiệm thu kém*: Các nhà phát triển Web Form không hề biết trước được rằng kiểm nghiệm tự động sẽ trở thành một thành phần thiết yếu của quá trình phát triển phần mềm. Không hề ngạc nhiên lắm, kiến trúc gắn kết chặt chẽ mà họ thiết kế không phù hợp cho các bài kiểm nghiệm chi tiết. Kiểm nghiệm tính tương thích, hoà hợp cũng biến thành một thử thách khó khăn Web Forms không phải tất cả đều kèm, xấu và Microsoft đã bỏ nhiều công sức vào việc phát triển các tiêu chuẩn, đơn giản hoá quá trình phát triển, và thêm vào đó lấy thêm các tính năng từ ASP.NET MVC.

2.1.1.2. MVC (Model – View – Controller)

MVC là một mô hình kiến trúc phân tách một ứng dụng thành ba phần chính: Model, View và Controller. Mỗi thành phần được xây dựng để xử lý một khía cạnh phát triển cụ thể của ứng dụng. MVC là một trong số những web framework tiêu chuẩn được sử dụng phổ biến nhất để tạo nên các dự án.

Ba thành phần trong mô hình MVC gồm có:

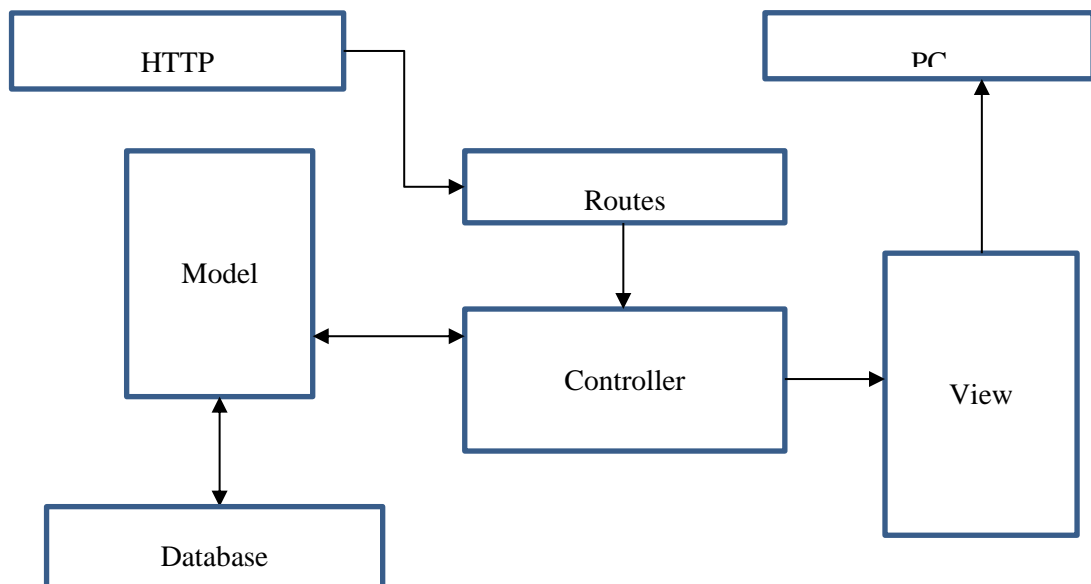
- Model: ứng với tất cả các logic liên quan đến dữ liệu. Model là đại diện cho dữ liệu được chuyển giao giữa View và Controller, liên quan đến database.

- View: được sử dụng cho tất cả các logic UI của ứng dụng, liên quan đến UI Interface.
- Controller: đóng vai trò trung gian giữa Model và View để xử lý các logic nghiệp vụ: sử dụng Model và tương tác với View để hiển thị đầu ra cuối cùng.

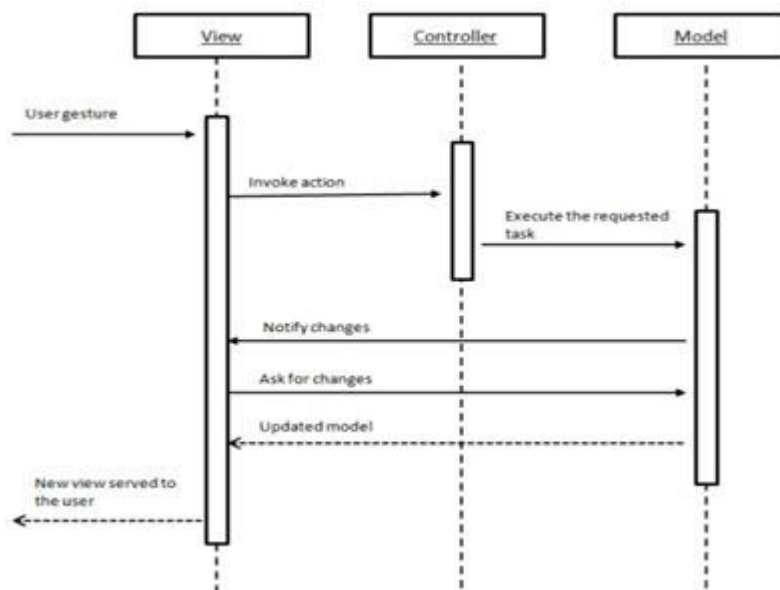
Quy trình xử lý chung của website hiện đại thường như sau:

- Người dùng thực hiện gọi yêu cầu xử lý.
- Controller nhận yêu cầu và đưa lệnh xử lý các yêu cầu đó. Các lệnh thực thi với View thì cập nhật hoặc phục vụ yêu cầu trang web, với Model thì trình diễn logic. Giả sử rằng lệnh yêu cầu có yếu tố logic.
- Model thực thi phần logic được lấy ra từ cơ sở dữ liệu và trả lại phản hồi dựa trên hướng dẫn của Controller.
- Controller truyền dữ liệu ra View, cập nhật giao diện hiển thị cho người dùng.

Bất kỳ yêu cầu nào cũng đều phải đi qua các Controller trước khi chúng được chuyển hóa thành các câu lệnh thực thi cho View hoặc Model.



Hình 2.2. Sơ đồ mô hình MVC



Hình 2.3. Sơ đồ xử lý dữ liệu trong mô hình MVC

2.1.1.3. ASP.NET MVC

Trong tháng 10/2007, Microsoft công bố một nền tảng phát triển MVC Web, được xây dựng dựa trên cốt lõi của nền tảng ASP.NET, được thiết kế rõ ràng để đối trọng trực tiếp với sự phát triển của công nghệ khác như Rails và đáp lại những chỉ trích phê bình của Web Forms. Nền tảng mới ASP.NET MVC đã khắc phục mọi hạn chế của Web Forms và đem ASP.NET ra khỏi bờ vực bị loại bỏ.

Khả năng mở rộng

MVC Framework được xây dựng như một loạt các thành phần độc lập đáp ứng giao diện .NET hay được xây dựng trên một lớp cơ sở trừu tượng. Bạn có thể dễ dàng thay thế các thành phần, như hệ thống định tuyến, cơ chế view, và controller, với từng thứ khác tùy vào thiết lập của bạn. Nói chung, MVC Framework đưa ra cho bạn 3 lựa chọn cho mỗi một thành phần:

- + Sử dụng các cài đặt mặc định của các thành phần (gần đủ cho hầu hết các ứng dụng).
- + Rút ra một lớp con của cài đặt mặc định để tinh chỉnh hành vi hoạt động của nó.

+ Thay thế hoàn toàn thành phần với thiết lập cài đặt mới của giao diện hay lớp cơ sở trừu tượng logic.

Kiểm soát chặt chẽ hơn đối với HTML và HTTP

ASP.NET MVC tạo ra tiêu chuẩn đánh dấu rõ ràng phù hợp. Phương pháp tích hợp hỗ trợ HTML của nó tạo ra chuẩn output, nhưng có một sự thay đổi đáng kể so với Web Forms. Thay vì tạo ra khoảng không cho HTML mà bạn có ít quyền kiểm soát, MVC Framework khuyến khích bạn tạo ra đánh dấu đơn giản với phong cách của CSS.

ASP.NET MVC làm việc phối hợp cùng HTTP. Bạn có quyền kiểm soát trên các yêu cầu đi qua giữa trình duyệt và server, nên bạn có thể điều chỉnh trải nghiệm của người dùng theo ý mình. AJAX được thực hiện dễ dàng, và không có bất kỳ postbacks tự động nào để can thiệp với trạng thái client.

Khả năng kiểm nghiệm

ASP.NET MVC hỗ trợ rất tốt trong việc kiểm nghiệm phần mềm. Bạn có thể tích hợp với công cụ kiểm nghiệm mã nguồn mở như Nunit và xUnit cũng như các công cụ test đi kèm trong Visual Studio. Việc kiểm thử các chức năng của phần mềm là vô cùng đơn giản.

Khả năng kiểm thử không chỉ là vấn đề của unit-testing. Ứng dụng ASP.NET MVC cũng làm việc tốt với công cụ thử nghiệm giao diện người dùng tự động hoá. Bạn có thể viết kịch bản kiểm tra mô phỏng các tương tác người dùng mà không cần đoán xem cấu trúc thành phần HTML, các lớp CSS, hay IDs mà framework sẽ tạo ra, và bạn không phải lo về sự thay đổi cấu trúc bất ngờ.

Hệ thống định tuyến mạnh mẽ

Một URL gọn rất khó để thực hiện trong nền tảng trước, nhưng ASP.NET MVC dùng một tính năng gọi là định tuyến URL để tạo ra đường dẫn URL ngắn gọn theo mặc định. Điều này cho phép bạn kiểm soát lược đồ đường dẫn URL của bạn và mối quan hệ của chúng với ứng dụng của bạn, cho bạn sự tự do để tạo

ra một mô hình đường dẫn có ý nghĩa và tiện ích cho người dùng, không cần thiết phải thoả theo một mô hình định trước.

Kế thừa những ưu điểm của nền tảng ASP.NET

ASP.NET MVC dựa trên nền tảng .NET, bạn có thể linh hoạt viết code với bất kì ngôn ngữ .NET và tiếp cận với cùng tính năng API - không chỉ nằm trong MVC mà còn thư viện lớp .NET rộng lớn và các hệ sinh thái của thư viện .NET của bên thứ ba.

Tính năng tạo sẵn của nền tảng ASP.NET – như chứng thực, thành viên, vai trò, cấu hình, và toàn cầu hoá - có thể làm giảm khối lượng code bạn cần để phát triển và duy trì bất kì ứng dụng Web nào. và những tính năng này cũng rất hiệu quả khi được dùng trong nền tảng MVC Framework vì chúng là dự án Web Form cổ điển. Các nền tảng bên dưới ASP.NET cung cấp một bộ công cụ phong phú có thể dựa trên đó để xây dựng ứng dụng Web với MVC Framework.

ASP.NET MVC là mã nguồn mở

Không giống các nền tảng phát triển Web trước đó của Microsoft, bạn được tự do download mã nguồn nguyên bản cho ASP.NET MVC, thêm vào đó được tuỳ chỉnh và biên dịch phiên bản cho riêng mình. Điều này rất có giá trị khi hướng đi debug của bạn hướng đến một thành phần hệ thống và bạn muốn được vào bên trong code của nó (xem xét những ý kiến ban đầu của nhà lập trình). Nó cũng hữu ích nếu bạn đang xây dựng một thành phần nâng cao và muốn xem thử khả năng phát triển có tồn tại, hay tìm hiểu cách mà cách thành phần đó thực sự làm việc.

2.1.1.4. ASP.NET MVC 5

Phiên bản 5 của MVC Framework là một nâng cấp tương đối nhỏ và rất nhiều thay đổi trong cách mà dự án ASP.NET được tạo ra và quản lý trong Visual Studio.

Tính năng	Mô tả
Authentication Filters- bộ lọc xác thực	Một dạng bộ lọc mới được dùng bao gồm các loại xác thực khác nhau trong cùng một controller
Filter Overrides- Quyền lọc filter	Một loại bộ lọc mới được áp dụng cho phương thức hoạt động ngăn cho các bộ lọc định nghĩa toàn cầu hay nằm trong controller khỏi bị ảnh hưởng
Attribute Routing- Thuộc tính định tuyến	Một tập hợp các thuộc tính cho phép URL định tuyến được định nghĩa trong lớp controller

ASP.NET 4.5.1, phiên bản mà MVC dựa vào đó để hình thành, đã được nâng cấp khá tốt. Thay đổi quan trọng nhất là việc thêm vào các hàm ASP.NET Identity API, thay thế cho hệ thống thành viên để quản lý thông tin xác thực, ủy quyền của người dùng.

2.1.2. Ngôn ngữ lập trình C#

C# là ngôn ngữ hướng đối tượng cho phép các nhà phát triển xây dựng đa dạng các ứng dụng hoạt động mạnh mẽ trên nền tảng .NET framework. C# có thể được sử dụng để tạo nên các ứng dụng khách Windows, các dịch vụ XML Web, các thành phần phân tán, các ứng dụng client – server, các ứng dụng cơ sở dữ liệu, ... Visual C# cung cấp một trình soạn thảo mã lệnh cải tiến, thiết kế giao diện người dùng tiện lợi, debugger tích hợp và rất nhiều các công cụ khác để dễ dàng phát triển các ứng dụng dựa trên ngôn ngữ C# và .NET framework.

Cú pháp của C# mang tính biểu đạt rõ ràng, vì vậy nó rất đơn giản và dễ học. Cú pháp của C# đơn giản hơn nhiều so với sự phức tạp của C++, mang đến những tính năng mạnh mẽ như kiểu giá trị null, enum, delegate, biểu thức lambda và truy cập bộ nhớ trực tiếp – điều không thể tìm thấy trong ngôn ngữ Java. C# hỗ trợ các kiểu và phương thức chung, các vòng lặp, ... Các truy vấn query (LINQ) tạo nên các truy vấn mạnh mẽ, hỗ trợ tối đa nhà phát triển trong việc truy cập cơ sở dữ liệu.

Là một ngôn ngữ hướng đối tượng, C# hỗ trợ các khái niệm về bao đóng, kế thừa và đa hình. Tất cả các tham số và phương thức, bao gồm cả các phương thức chính, các điểm truy cập ứng dụng, đều được đóng gói trong các định nghĩa lớp. Một lớp có thể kế thừa trực tiếp từ một lớp cha. Một phương thức có thể được nạp chồng một phương thức ảo khác trong cùng một lớp. Trong C#, một struct giống như một lớp, nó là một ngăn xếp phân loại thực hiện giao diện nhưng không hỗ trợ kế thừa.

C# hỗ trợ các phát triển các component của phần mềm thông qua một số cấu trúc ngôn ngữ: các inline XML documentation comment, truy vấn ngôn ngữ tích hợp (LINQ) cung cấp khả năng truy vấn được xây dựng trên nhiều nguồn dữ liệu khác nhau, ...

2.2. Công nghệ xử lý giao diện

2.2.1. HTML (Hypertext Markup Language)

HTML là viết tắt của cụm từ “Hypertext Markup Language” – ngôn ngữ đánh dấu siêu văn bản. HTML là ngôn ngữ được sử dụng để tạo nên các trang web. “Hypertext” đề cập đến các hyperlink và một trang HTML có thể có. “Markup language” đề cập đến cách mà các tag được sử dụng để định nghĩa nên một page layout và các element trong page.

Web đã trải qua nhiều thay đổi trong hàng thập kỷ vừa qua, tuy nhiên HTML vẫn luôn được sử dụng để phát triển các trang web. Các trang web đã được cải tiến và hơn rất nhiều so với thời kỳ sơ khai, ngày càng phức tạp hơn, trong khi đó, HTML lại trở nên đơn giản hơn rất nhiều. Với cùng một webpage, HTML5 có ít dòng lệnh hơn HTML4.01 hoặc XHTML1.0. Điều này là do HTML hiện đại phụ thuộc vào việc xếp tầng hoặc Javascript để format hầu như toàn bộ các element trong một trang.

Nội dung các thẻ cơ bản của HTML như sau:

- `<!DOCTYPE html>`: xác định loại tài liệu và phiên bản HTML.
- `<html>`: phần tử gốc của trang HTML
- `<head>`: chứa tiêu đề tài liệu mà có thể giữ các thẻ như `<title>`, `<link>`, ...
- `<title>`: chứa tiêu đề tài liệu

- `<body>`: chứa nội dung trang có thể nhìn thấy
- `<h1>`: xác định một tiêu đề lớn
- `<p>`: xác định một đoạn văn

2.2.2. CSS (Cascading Style Sheets)

CSS là viết tắt của từ “Cascading Style Sheet”. CSS được sử dụng để format layout của một trang web. Chúng có thể được sử dụng để định nghĩa các kiểu text, kích thước bảng, và các thành phần khác của trang web mà đã được định nghĩa từ trước bởi HTML.

CSS giúp cho các nhà phát triển website tạo ra một cái nhìn thống nhất trên một trang web. Thay vì phải định nghĩa kiểu cho từng bảng và từng khối trong một trang HTML, các style thường dùng được định nghĩa một lần trong file CSS. Khi style đã được định nghĩa, nó có thể được sử dụng ở bất cứ trang web nào có liên kết đến file CSS đó.

Ngoài việc hỗ trợ tối đa tạo các style text, CSS cũng giúp ích rất nhiều trong việc format các thành phần khác của web. CSS có thể được sử dụng để xác định lại các padding của table cell, style, màu của đường viền bảng. CSS giúp cho các nhà phát triển Web chính xác hơn trong việc thể hiện ra giao diện người dùng một cách đẹp mắt hơn một trang HTML thông thường. Hầu như tất cả các trang web hiện nay đều sử dụng CSS để style cho trang web của mình.

2.2.3. JavaScript

Javascript là ngôn ngữ kịch bản thường được sử dụng khi phát triển website. Ban đầu, nó được phát triển bởi Netscape như một cách để thêm các element động và có tính tương tác cho trang web. Javascript chịu ảnh hưởng của Java nhưng lại có cú pháp giống với C#, dựa trên nền tảng ECMAScript – một ngôn ngữ kịch bản được phát triển bởi Sun Microsystems.

Javascript là ngôn ngữ kịch bản phía máy khách, nghĩa là source code được xử lý ở phía trình duyệt người dùng thay vì trên web server. Các hàm Javascript có thể chạy sau khi một trang web được load mà không cần phải liên kết với server. Ví dụ một hàm Javascript có thể kiểm tra một form trên trang web trước khi nó được gửi để đảm bảo các hàm bắt buộc điền thông tin đều không để trống.

Code Javascript có thể tạo ra thông báo lỗi trước khi bất kỳ thông tin nào thực sự được gửi đến server.

Giống như các ngôn ngữ kịch bản phía máy khách, chẳng hạn như PHP và ASP, Javascript code có thể được thêm vào bất cứ đâu trong HTML webpage. Tuy nhiên, chỉ có đầu ra của server – side code mới được hiển thị ra trong HTML, trong khi Javascript code còn lại vẫn hiển thị trên source web. Nó cũng có thể được tham chiếu trong một file .JS riêng biệt, có thể được xem trên trình duyệt.

2.3. Một số Third – Party

2.3.1. Google Authenticator

Google Authenticator là phần mềm được cài đặt trên các thiết bị như điện thoại di động, máy tính bảng, máy vi tính, để cung cấp mật khẩu sử dụng một lần khi sử dụng các ứng dụng trực tuyến được phát triển bởi Google. Nó sử dụng Thuật toán mật khẩu một lần dựa trên thời gian (TOTP) và Thuật toán mật khẩu một lần dựa trên HMAC (HOTP) để xác thực người dùng truy cập vào các ứng dụng di động.

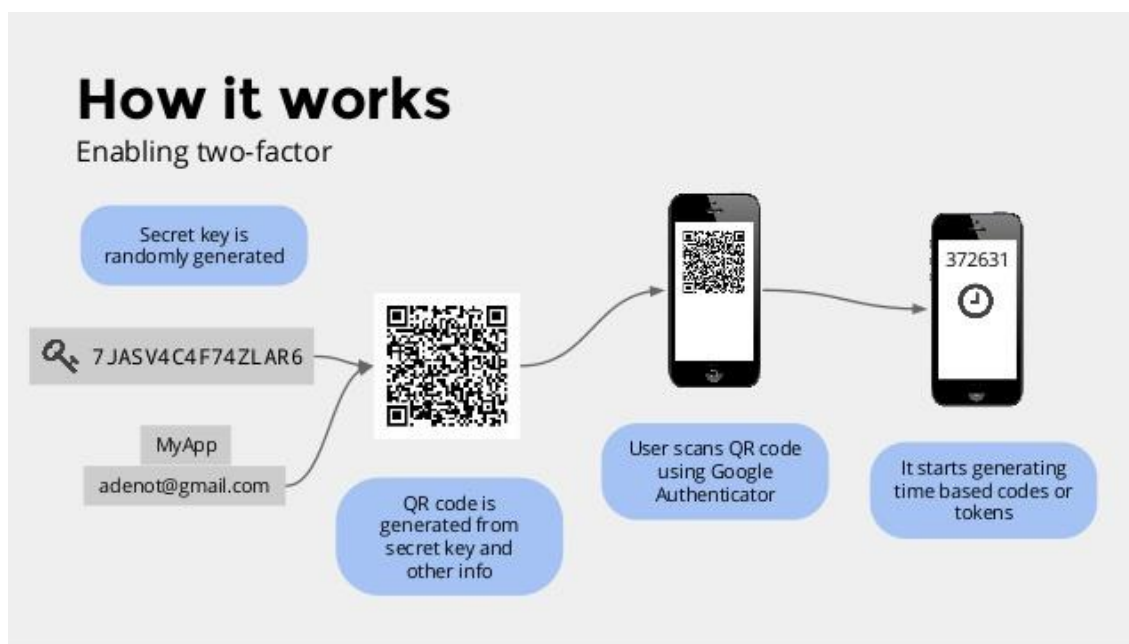
Với HOTP, máy chủ và máy khách chia sẻ một giá trị bí mật và một bộ đếm, được sử dụng để tính toán mật khẩu một lần độc lập trên cả hai mặt. Bất cứ khi nào mật khẩu được tạo và sử dụng, bộ đếm được tăng lên ở cả hai bên, cho phép máy chủ và máy khách đồng bộ.

TOTP về cơ bản sử dụng cùng một thuật toán như HOTP với một sự khác biệt lớn. Bộ đếm được sử dụng trong TOTP được thay thế bằng thời gian hiện tại. Máy khách và máy chủ vẫn được đồng bộ miễn là thời gian của hệ thống vẫn giữ nguyên. Điều này có thể được thực hiện bằng cách sử dụng giao thức Thời gian Mạng.

Khóa bí mật (cũng như bộ đếm trong trường hợp HOTP) phải được truyền đạt tới cả máy chủ và máy khách tại một thời điểm nào đó. Trong trường hợp Google Authenticator, điều này được thực hiện dưới dạng URI được mã hóa QRCode.

DLYM sử dụng Google Authenticator để xác thực người dùng truy cập hệ thống theo hai bước. Người dùng có thể tùy chọn sử dụng xác thực hai lớp hay đăng nhập chỉ sử dụng tài khoản và mật khẩu.

1. Khách hàng đăng nhập sử dụng tài khoản và mật khẩu đã đăng ký trên hệ thống. Nếu tài khoản và mật khẩu chính xác, khách hàng sẽ được chuyển sang bước xác thực thứ hai.
2. Khách hàng sử dụng ứng dụng Google Authenticator quét mã QR để nhận mã bí mật (mã bí mật được thay đổi sau 30 giây), sử dụng mã bí mật đó để xác thực tại bước hai này. Nếu mã xác thực chính xác, thì khách hàng đã đăng nhập thành công. Ngược lại, hệ thống yêu cầu khách hàng nhập lại mã bí mật.



Hình 2.4. Kiến trúc Google Authenticator

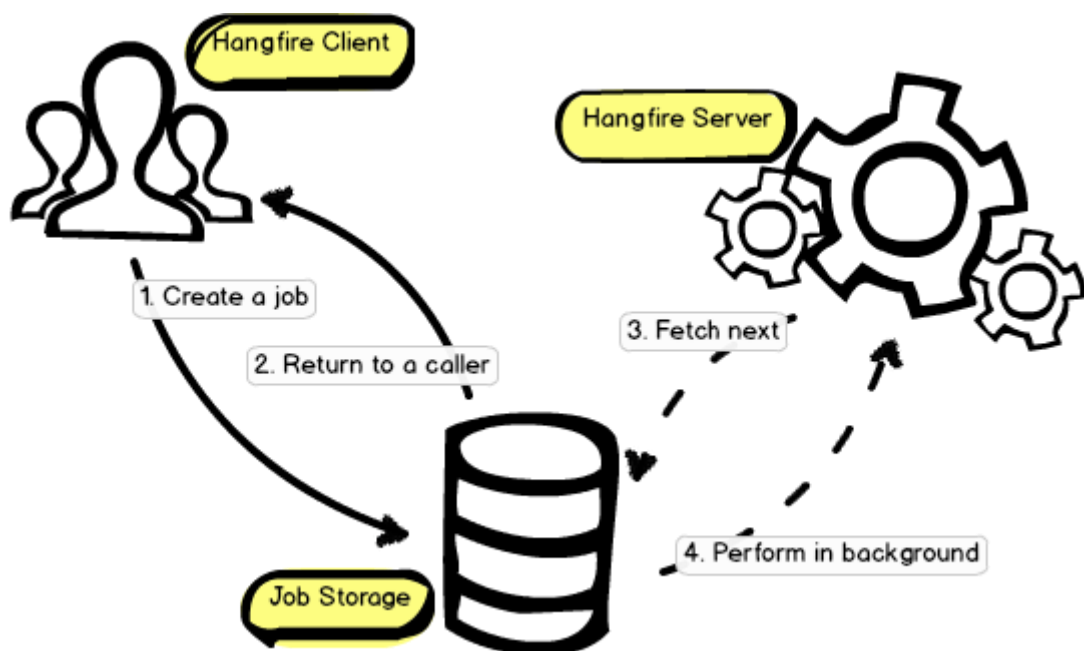
2.3.2. Hang Fire

Hang Fire là một thư viện mã nguồn mở của .NET, cho phép lập lịch biểu các công việc chạy ngầm trên hệ thống. Đây là một giải pháp để thay thế cho dịch vụ Windows Service.

Hangfire hỗ trợ tất cả các loại tác vụ nền - ngắn hạn và chạy dài, chuyên sâu về CPU và I/O, công việc thực hiện một lần và lặp lại thường xuyên mà

không cần phải khởi động lại. Các công việc mà Hang Fire có thể hỗ trợ bao gồm:

- **Fire-and-forget:** Những công việc này được thực hiện chỉ một lần và gần như ngay lập tức sau khi chạy ứng dụng.
- **Delayed:** Các công việc bị trì hoãn chỉ được thực hiện một lần, nhưng không phải ngay lập tức - chỉ sau khoảng thời gian được chỉ định.
- **Recurring:** Các công việc định kỳ được thực hiện nhiều lần theo lịch biểu CRON được chỉ định .
- **Continuations:** Tiếp tục được thực hiện khi công việc cha đã hoàn thành .
- **Batches:** Một nhóm các công việc nên được tạo ra một cách nguyên tử.
- **Batch Continuations:** Tiếp tục hàng loạt được kích hoạt sau khi tất cả các công việc nên trong một nhóm công việc cha đã hoàn thành .
- **Background Process:** Sử dụng chúng khi cần chạy các quy trình nền liên tục trong suốt thời gian tồn tại của ứng dụng của bạn.



Hình 2.5. Kiến trúc Hang Fire

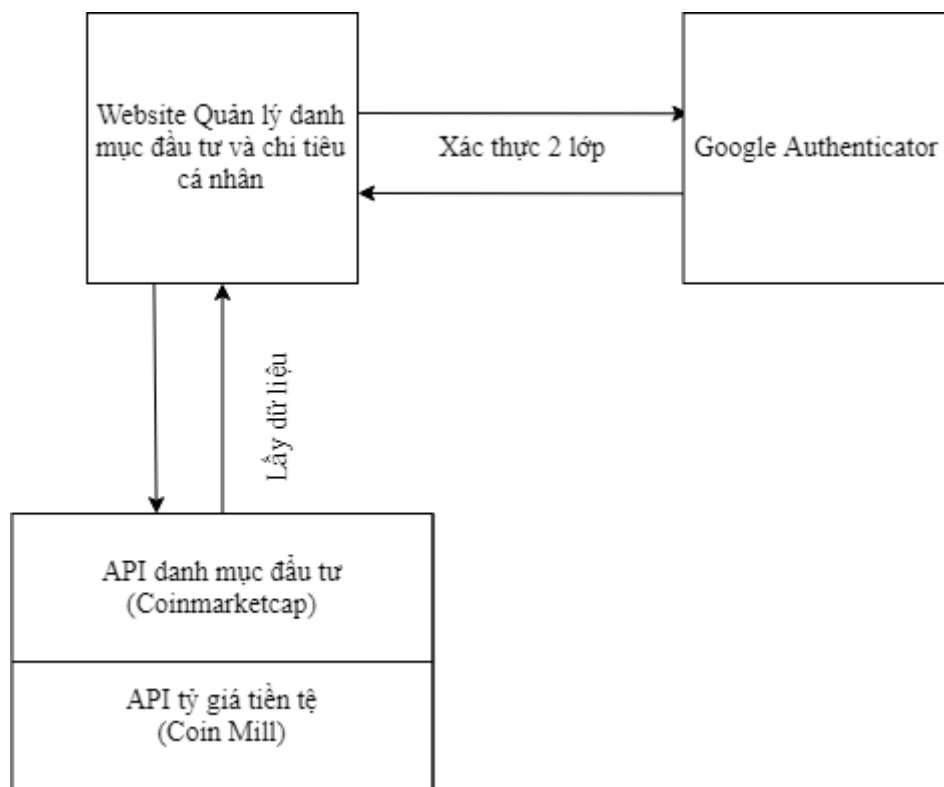
Chương III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Mô tả hệ thống

Hệ thống được thiết kế để cho hai đối tượng sử dụng:

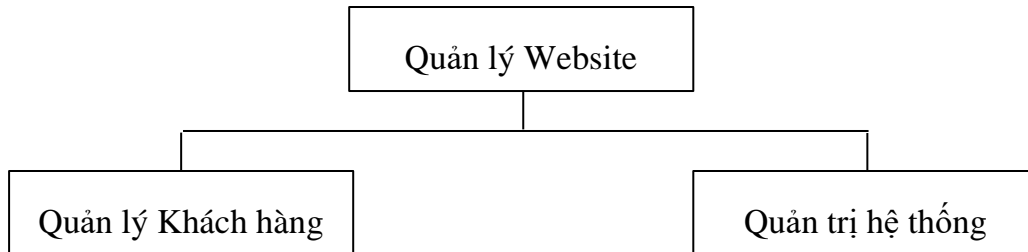
- Đối với thành viên: Khách hàng sau khi đã đăng ký tài khoản thành công và cập nhật thông tin cá nhân được sử dụng đầy đủ chức năng của hệ thống bao gồm: Tạo ví, quản lý chi tiêu và thu nhập, quản lý đầu tư, quản lý chi tiêu định kỳ, lập kế hoạch chi tiêu, thống kê báo cáo.
- Đối với quản trị viên: Sử dụng tài khoản trên trang quản trị để thực hiện các chức năng: Quản lý các danh mục, quản lý tài khoản khách hàng, quản lý phân quyền, thống kê báo cáo.

3.1.1. Sơ đồ kiến trúc hệ thống



Hình 3.1. Sơ đồ kiến trúc của hệ thống

3.1.2. Sơ đồ chức năng của hệ thống

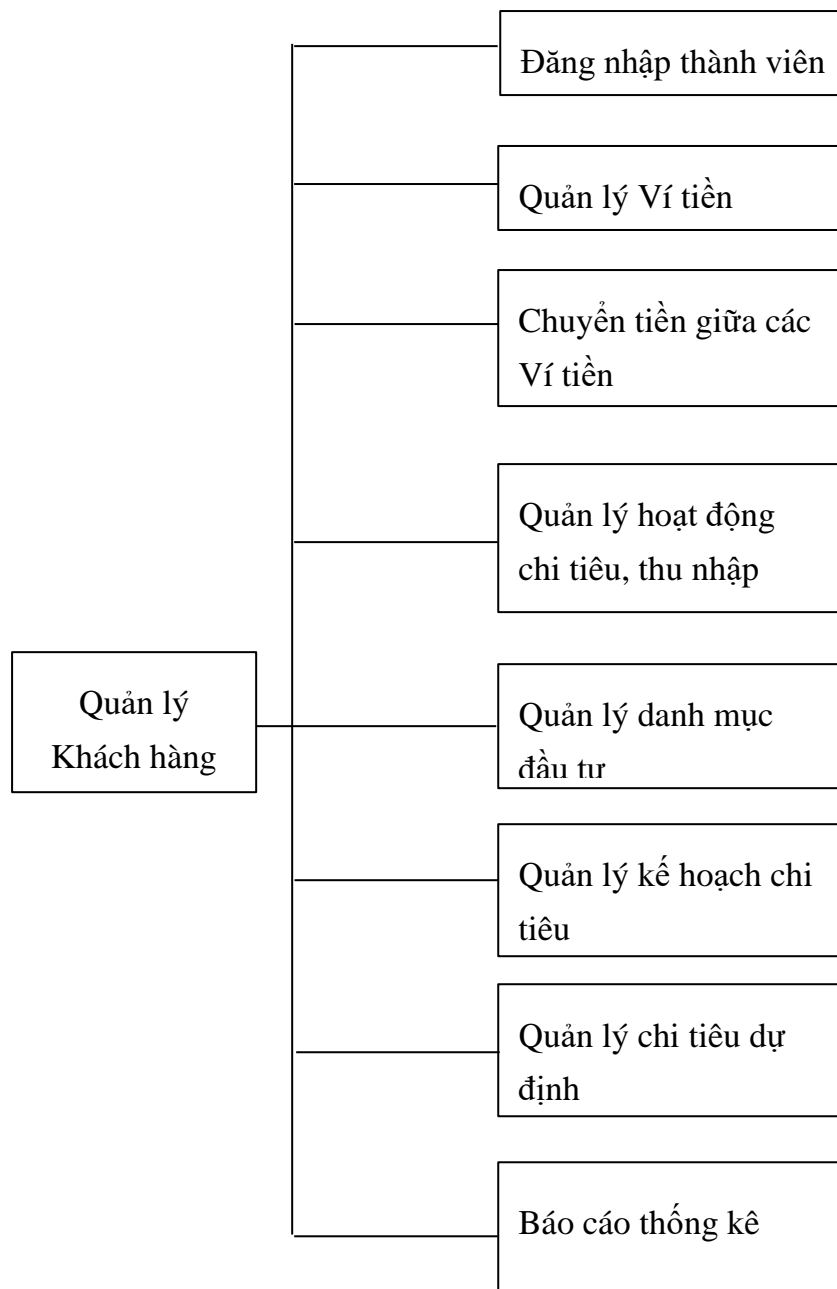


Hình 3.2. Sơ đồ chức năng của hệ thống

3.1.3. Chức năng đối với khách hàng.

- Đăng nhập thành viên
- Quản lý Ví
- Quản lý hoạt động chi tiêu, thu nhập
- Quản lý danh mục đầu tư
- Quản lý kế hoạch chi tiêu
- Quản lý chi tiêu dự định
- Báo cáo thống kê

Sơ đồ chức năng:

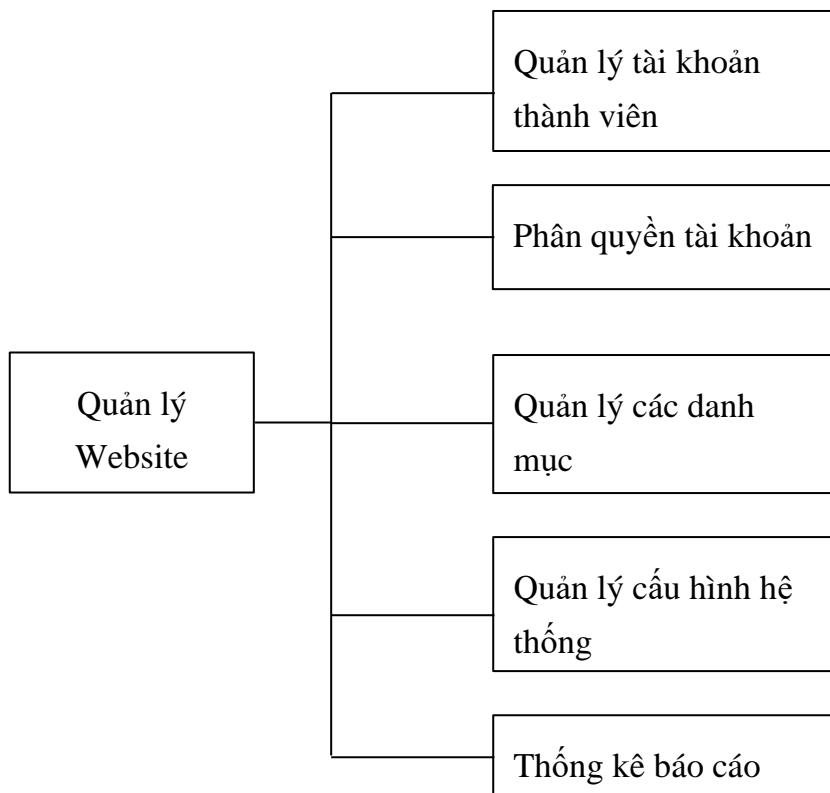


Hình 3.3. Sơ đồ chức năng đối với khách hàng

3.1.4. Chức năng đối với quản trị Website

- Quản lý tài khoản thành viên
- Quản lý các danh mục
- Quản lý cấu hình hệ thống
- Thống kê báo cáo

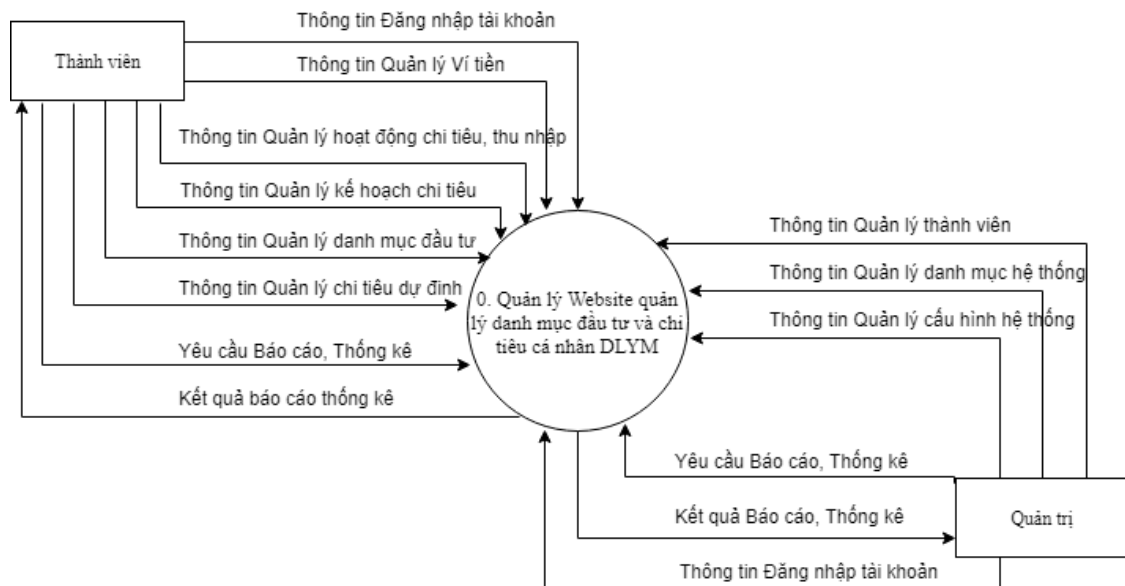
Sơ đồ chức năng:



Hình 3.4. Sơ đồ chức năng đối với quản trị Website

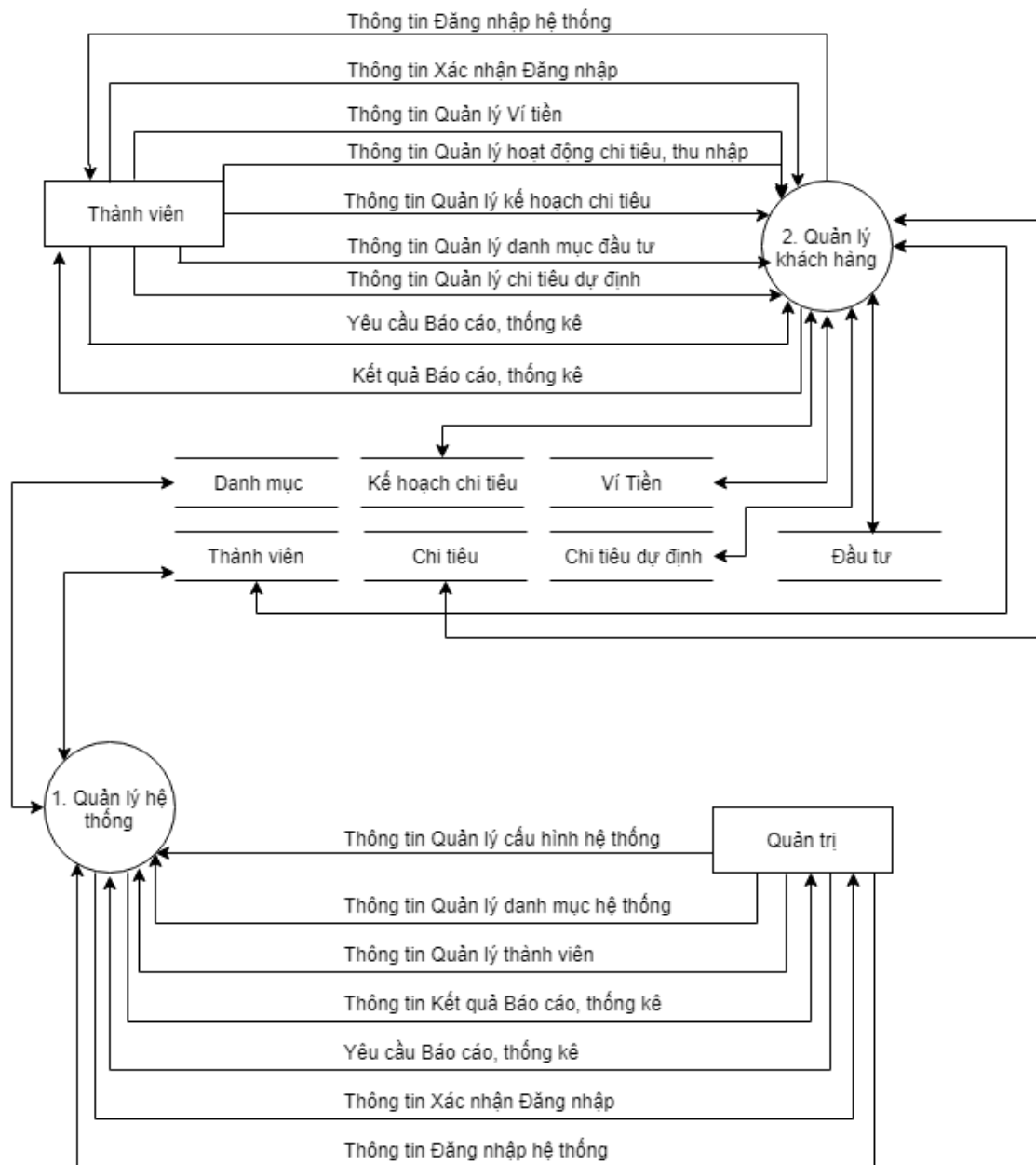
3.2. Phân tích hệ thống về mặt chức năng

3.2.1. Biểu đồ luồng dữ liệu (BLD) mức khung cảnh



Hình 3.5. Biểu đồ luồng dữ liệu (BLD) mức khung cảnh

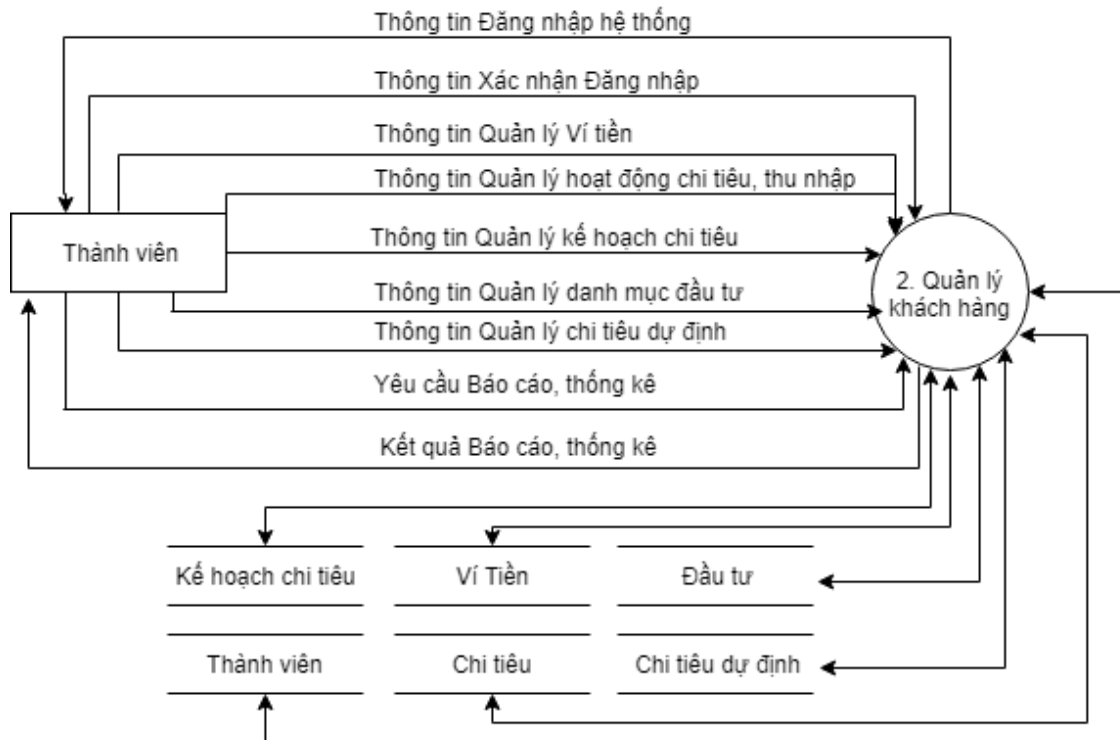
3.2.2. Biểu đồ luồng dữ liệu (BLD) mức đỉnh



Hình 3.6. Biểu đồ luồng dữ liệu (BLD) mức khung cảnh

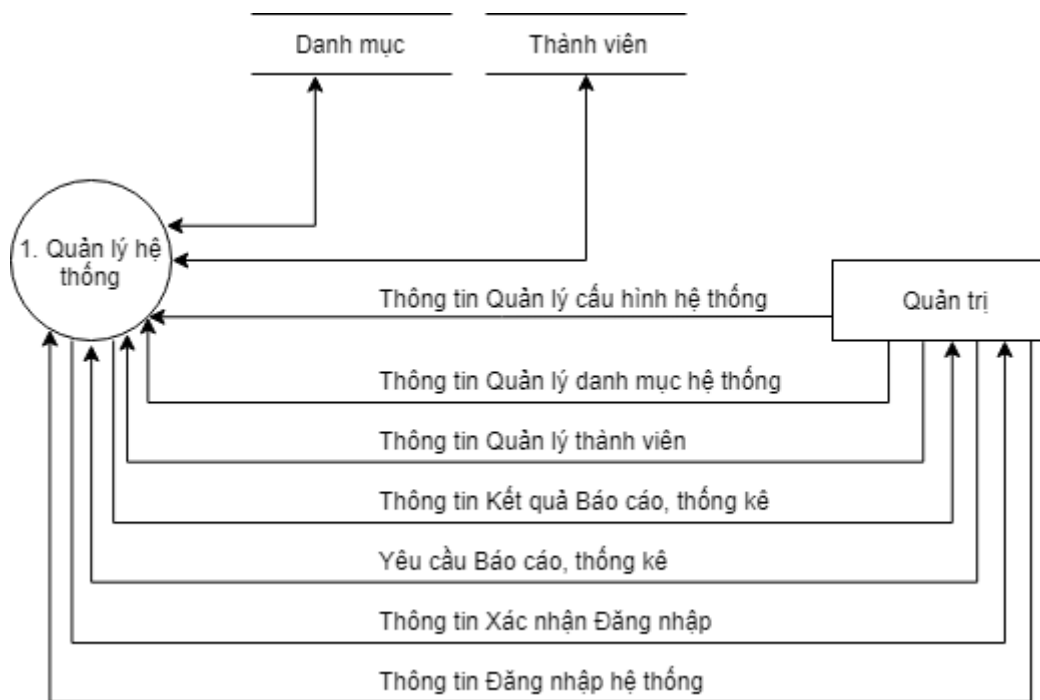
3.2.3. Biểu đồ luồng dữ liệu (BLD) mức dưới đỉnh

3.2.3.1. Đối với chức năng quản lý khách hàng



Hình 3.7. Biểu đồ luồng dữ liệu (BLD) mức dưới đỉnh
đối với chức năng quản lý khách hàng

3.2.3.2. Đối với chức năng quản lý hệ thống

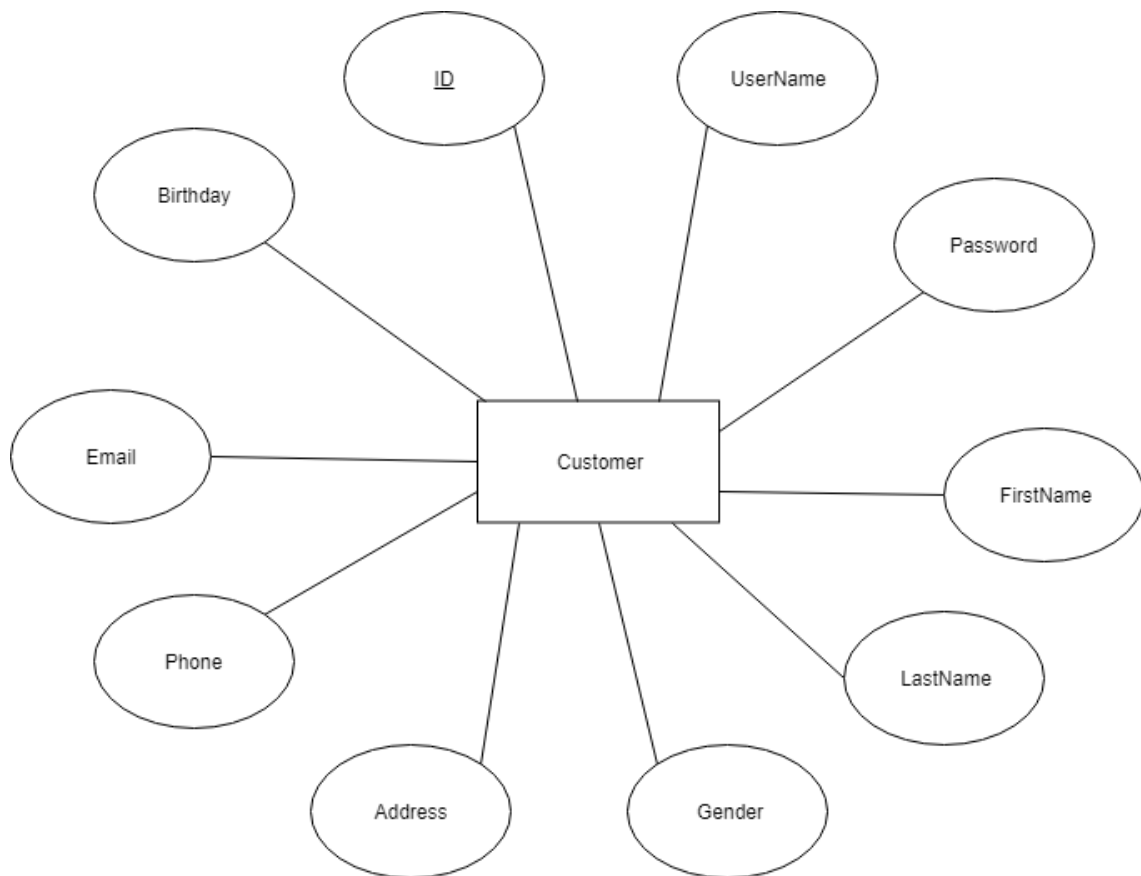


Hình 3.8. Biểu đồ luồng dữ liệu (BLD) mức dưới đỉnh
đối với chức năng quản lý hệ thống

3.3. Phân tích hệ thống về cơ sở dữ liệu

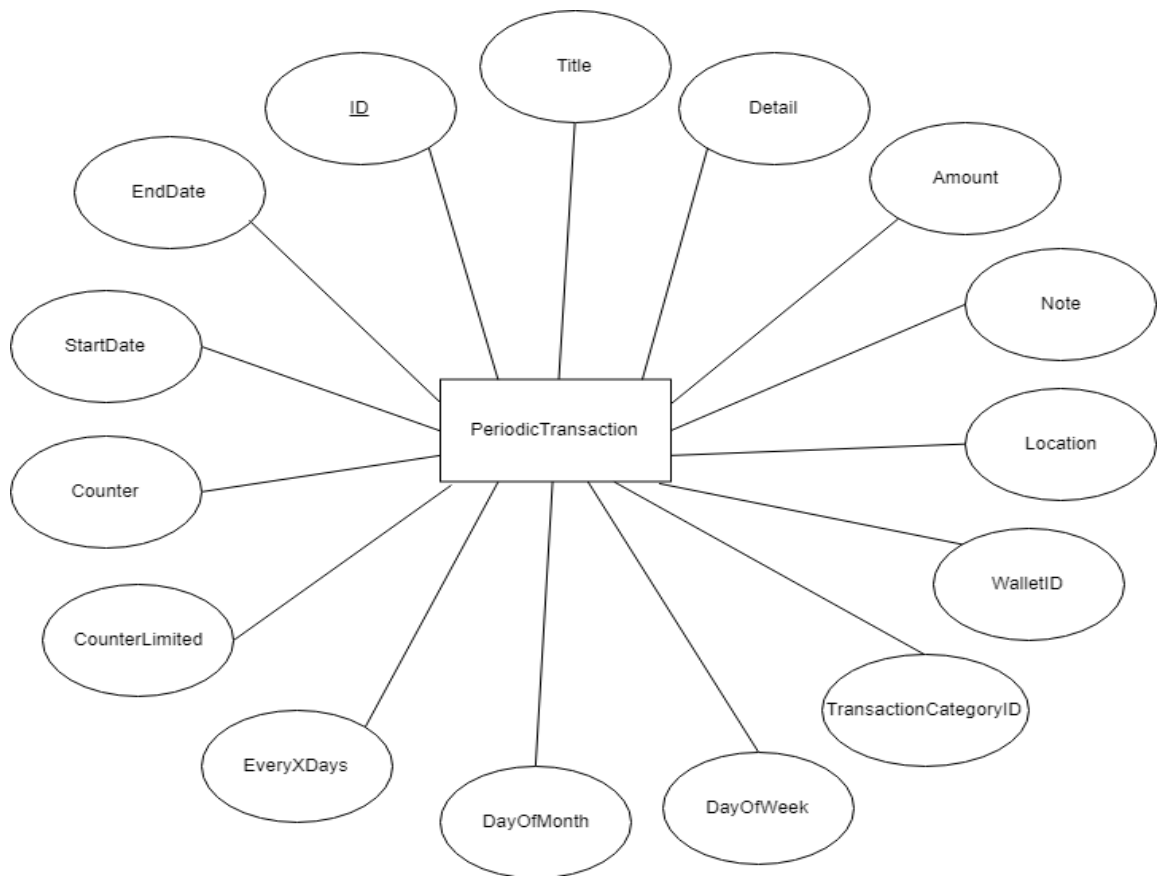
3.3.1. Xác định các thực thể

Thực thể Khách hàng(Customer):



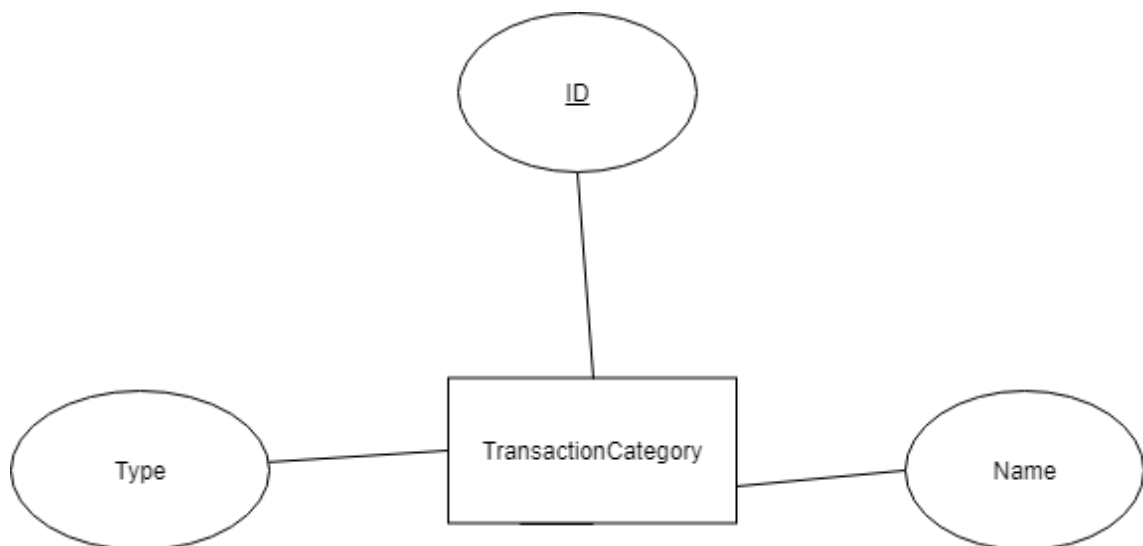
Hình 3.9. Thực thể Khách hàng(Customer)

Thực thể Chi tiêu dự định (PeriodicTransaction)



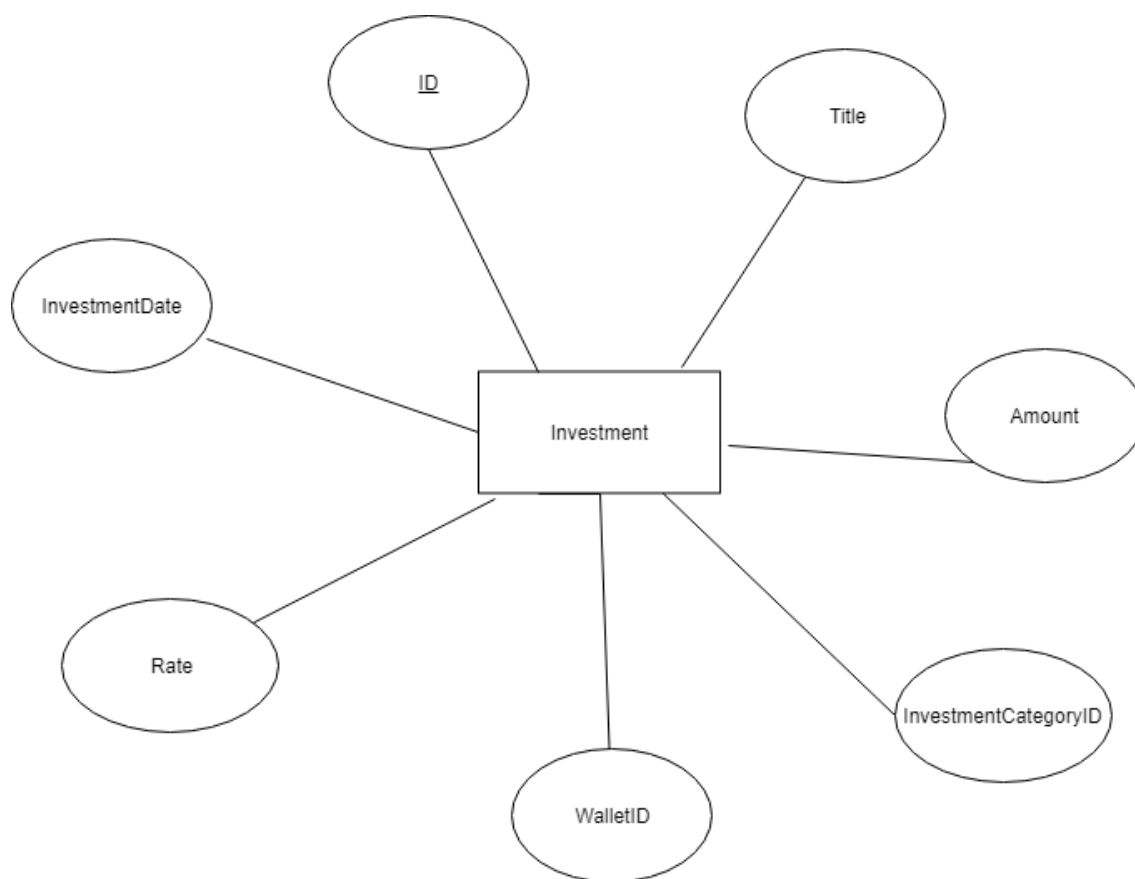
Hình 3.10. Thực thể Chi tiêu dự định (PeriodicTransaction)

Thực thể Danh mục chi tiêu, thu nhập (TransactionCategory)



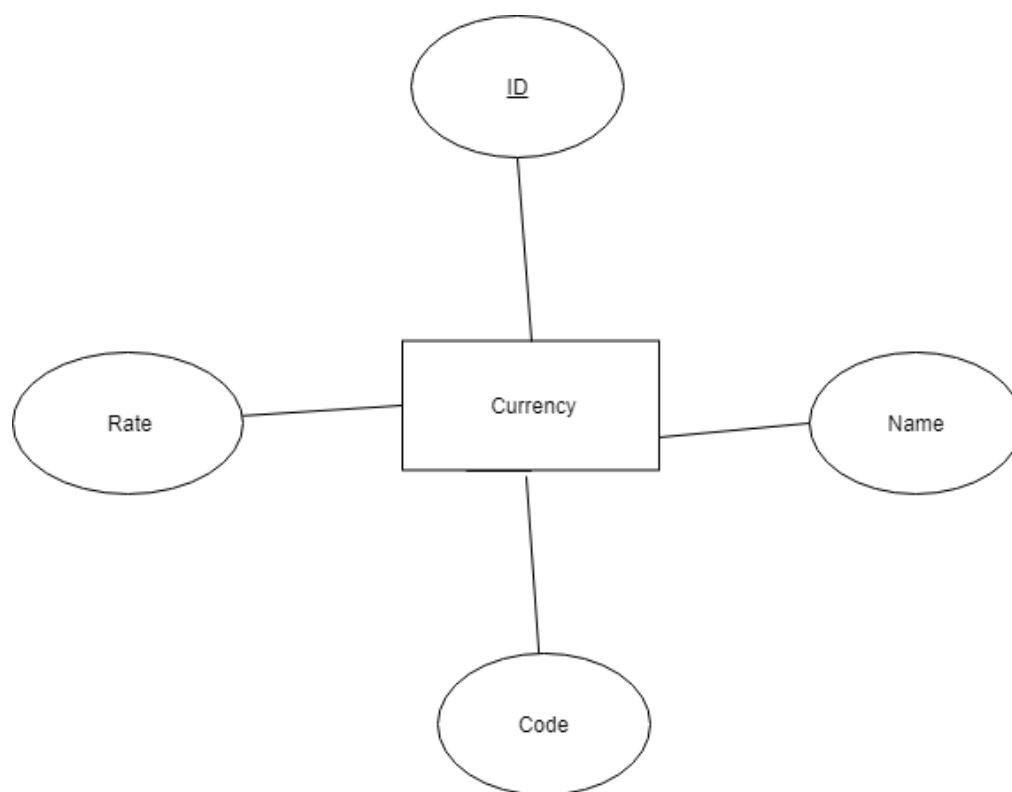
Hình 3.11. Thực thể Danh mục chi tiêu, thu nhập (TransactionCategory)

Thực thể Kế hoạch (Budget)



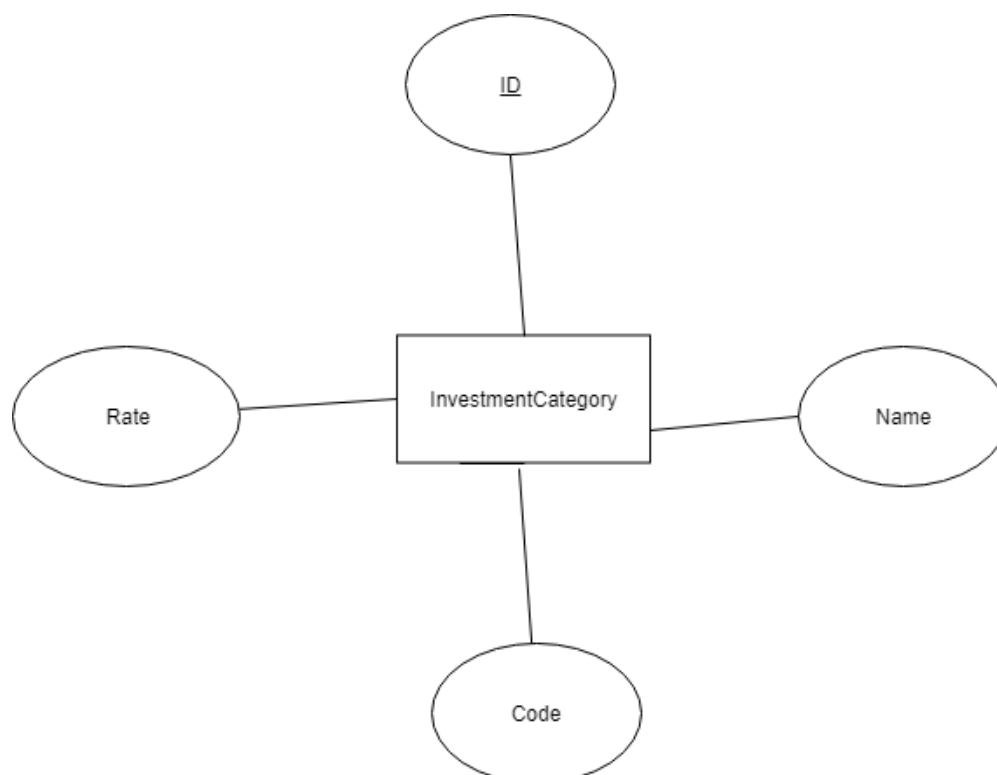
Hình 3.12. Thực thể Kế hoạch (Budget)

Thực thể Tiền tệ (Currency)



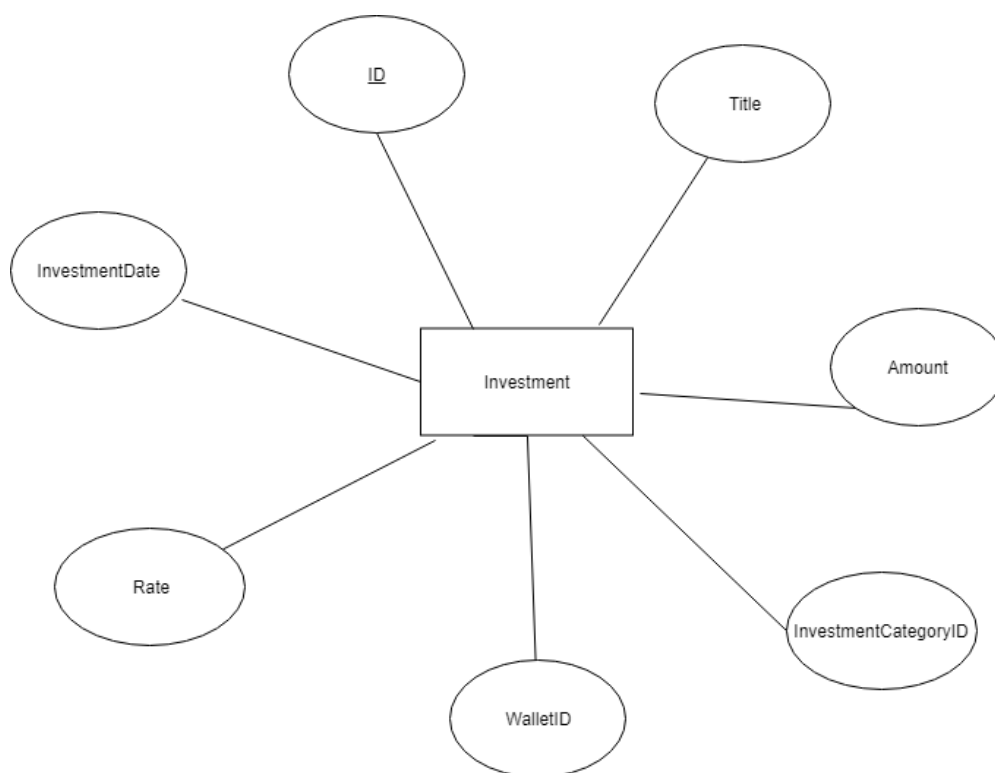
Hình 3.13. Thực thể Tiền tệ (Currency)

Thực thể Danh mục đầu tư (InvestmentCategory)



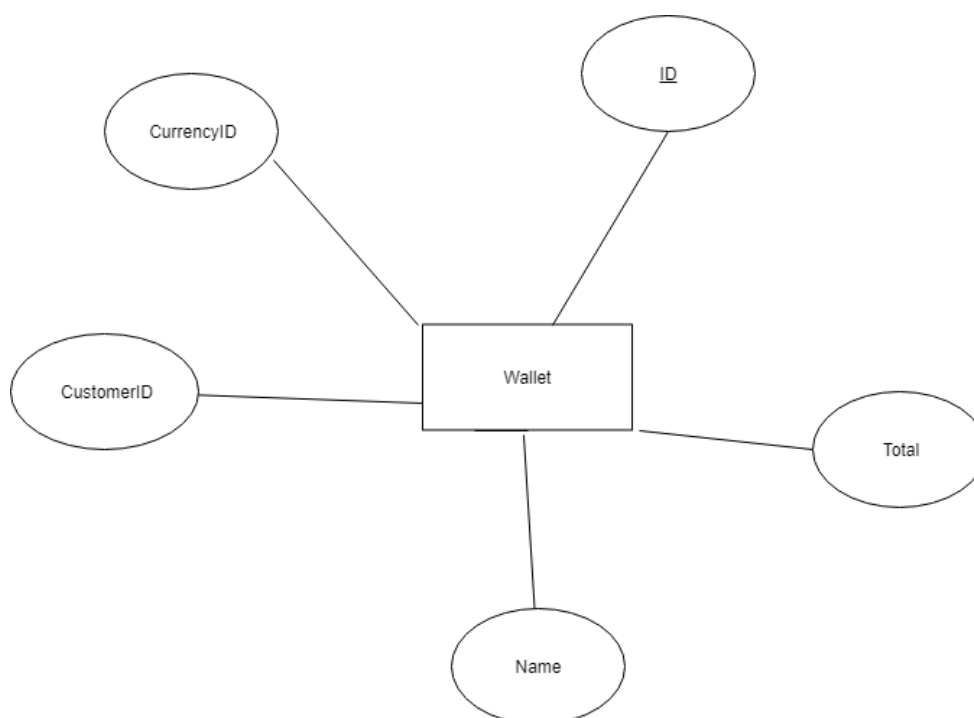
Hình 3.14. Thực thể Danh mục đầu tư (InvestmentCategory)

Thực thể Đầu tư (Investment)



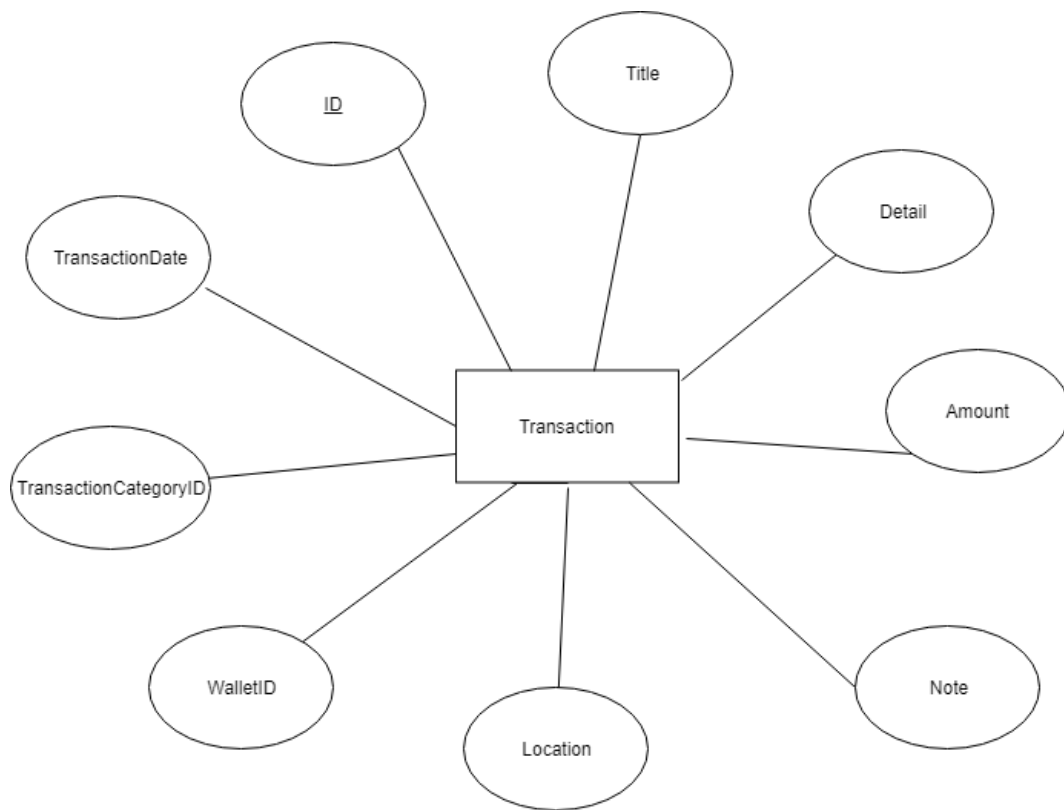
Hình 3.15. Thực thể Đầu tư (Investment)

Thực thể Ví người dùng (Wallet)



Hình 3.16. Thực thể Ví người dùng (Wallet)

Thực thể Chi tiêu, thu nhập (Transaction)

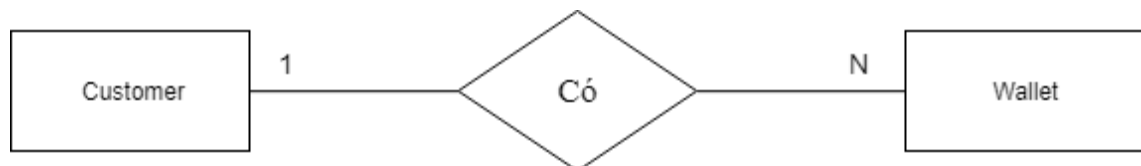


Hình 3.17. Thực thể Chi tiêu, thu nhập (Transaction)

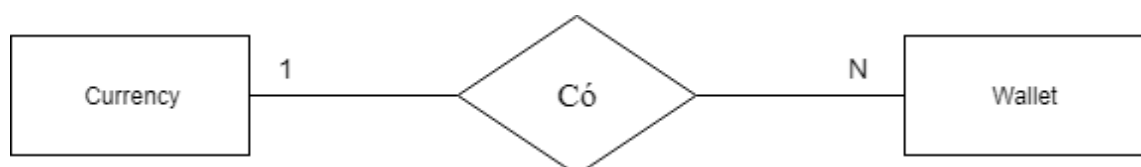
3.3.2. Sơ đồ thực thể liên kết

3.3.2.1. Xét các thực thể liên kết

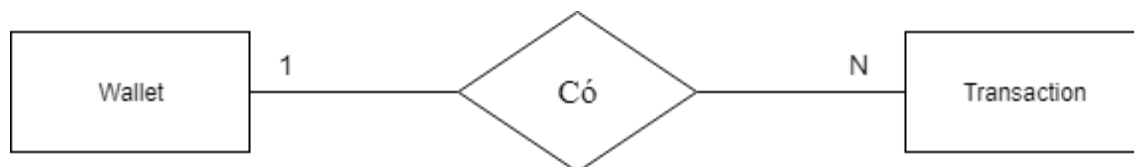
Mỗi khách hàng được phép tạo mới và sử dụng nhiều ví tiền. Do đó, liên kết giữa hai thực thể Customer và Wallet là liên kết một – nhiều.



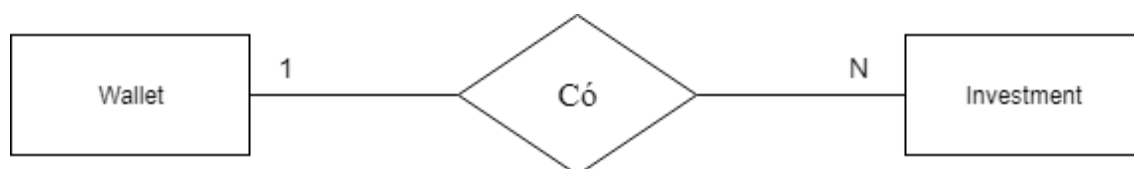
Mỗi ví tiền khách hàng thuộc một loại tiền tệ. Do đó, liên kết giữa hai thực thể Currency và Wallet là liên kết một – nhiều.



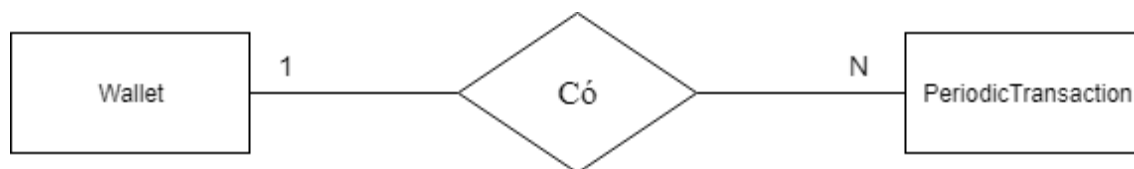
Mỗi khách hàng được phép tạo mới nhiều hoạt động chi tiêu, thu nhập ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và Transaction là liên kết một – nhiều.



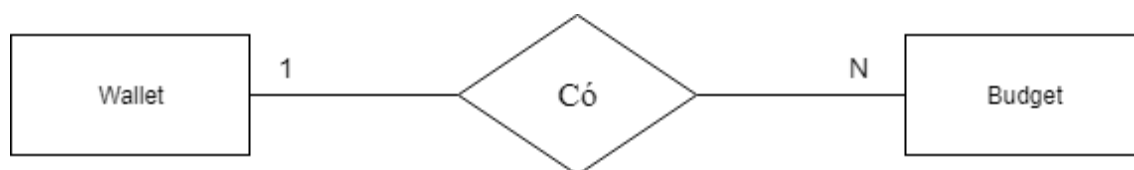
Mỗi khách hàng được phép tạo mới nhiều hoạt động đầu tư ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và Investment là liên kết một – nhiều.



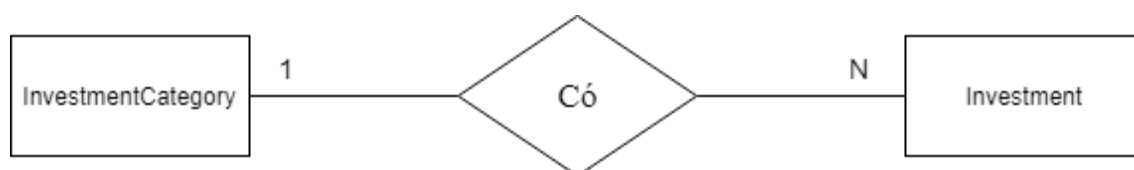
Mỗi khách hàng được phép tạo mới nhiều chi tiêu dự định ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và PeriodicTransaction là liên kết một – nhiều.



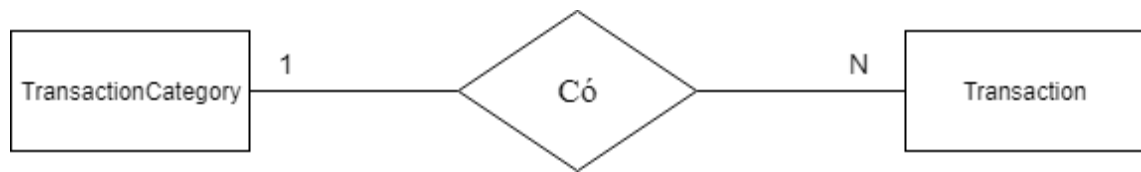
Mỗi khách hàng được phép tạo mới nhiều kế hoạch chi tiêu ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và Budget là liên kết một – nhiều.



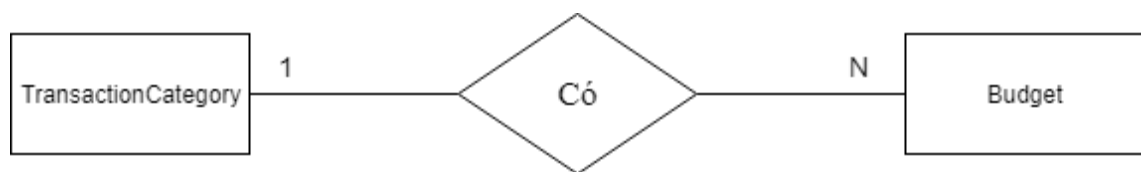
Mỗi khoản đầu tư thuộc một danh mục đầu tư. Do đó, liên kết giữa hai thực thể InvestmentCategory và Investment là liên kết một – nhiều.



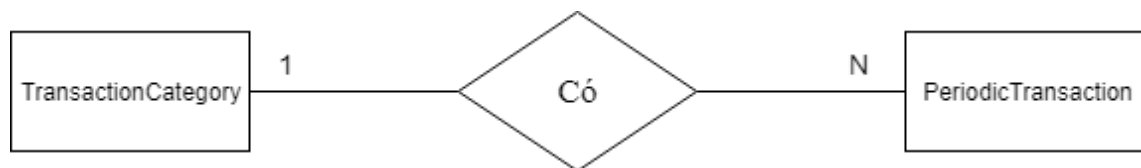
Mỗi hoạt động chi tiêu, thu nhập thuộc một danh mục chi tiêu. Do đó, liên kết giữa hai thực thể TransactionCategory và Transaction là liên kết một – nhiều.



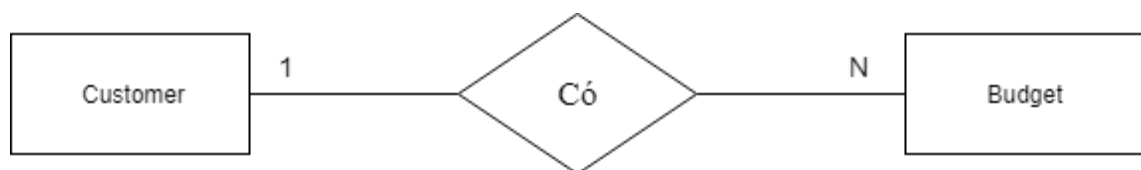
Mỗi kế hoạch chi tiêu thuộc một danh mục chi tiêu. Do đó, liên kết giữa hai thực thể TransactionCategory và Budget là liên kết một – nhiều.



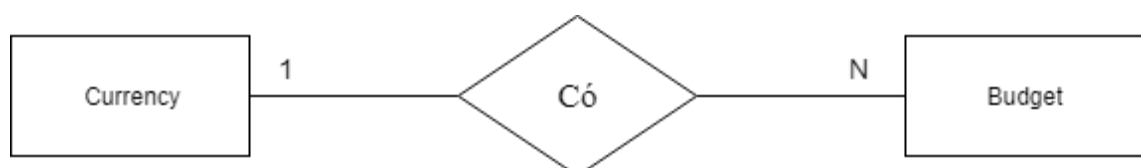
Mỗi chi tiêu dự định thuộc một danh mục chi tiêu. Do đó, liên kết giữa hai thực thể PeriodicTransaction và Transaction là liên kết một – nhiều.



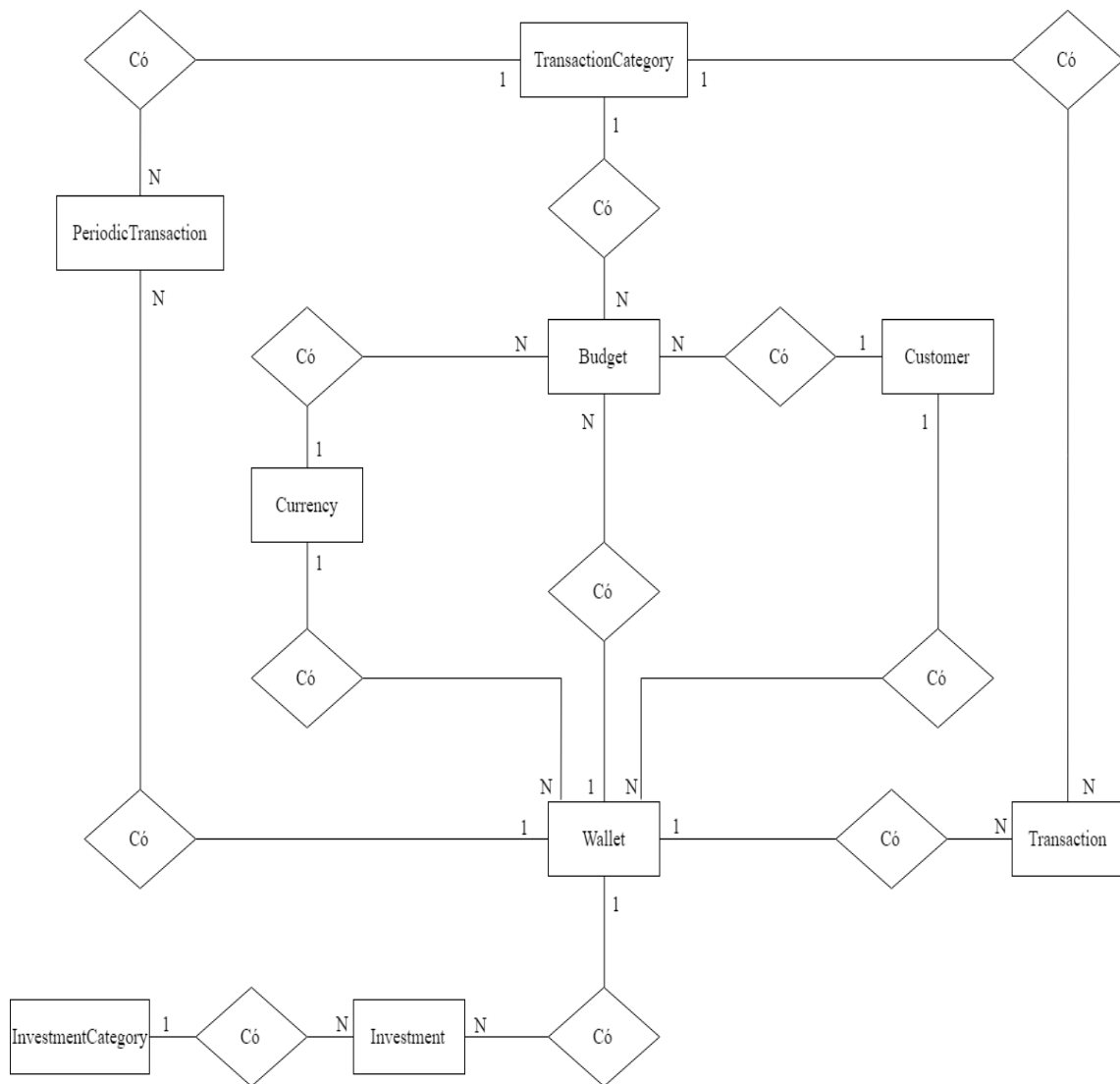
Mỗi kế hoạch chi tiêu khi khách hàng muốn lên kế hoạch cho toàn bộ các ví, kế hoạch chi tiêu của khách hàng sẽ không gáp với một ví tiền cố định mà gắn trực tiếp với khách hàng đó. Do đó, liên kết giữa hai thực thể Customer và Budget là liên kết một – nhiều.



Mỗi kế hoạch chi tiêu khi khách hàng muốn lên kế hoạch cho toàn bộ các ví, kế hoạch chi tiêu của khách hàng thuộc một loại tiền tệ. Do đó, liên kết giữa hai thực thể Currency và Budget là liên kết một – nhiều.

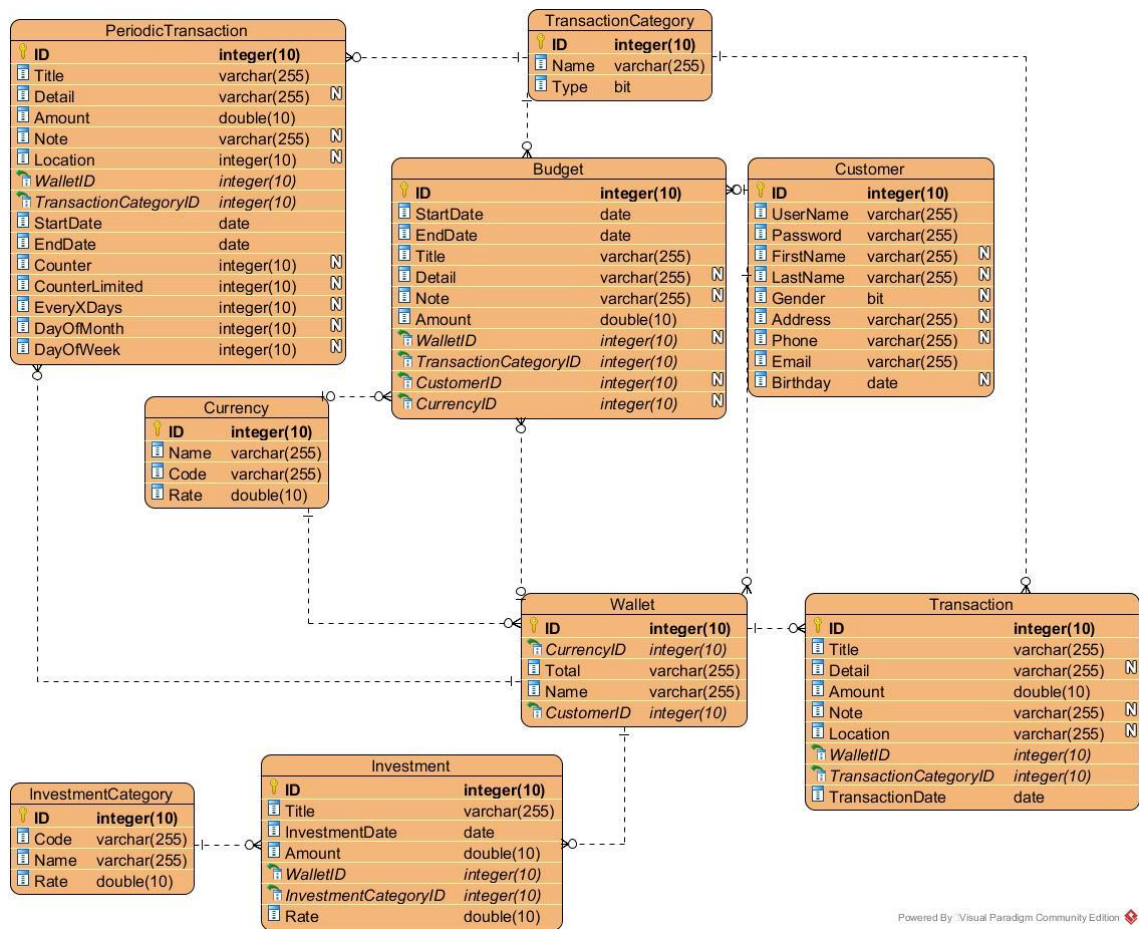


3.3.2.2. Sơ đồ thực thể liên kết



Hình 3.18. Sơ đồ thực thể liên kết

3.3.3. Mô hình quan hệ giữa các thực thể



Hình 3.19. Mô hình quan hệ giữa các thực thể

3.3.4. Danh sách các bảng dữ liệu

STT	TÊN BẢNG	DIỄN GIẢI
1	Customer	Lưu thông tin tài khoản khách hàng
2	Wallet	Lưu thông tin ví tiền của khách hàng
3	Currency	Lưu thông tin các loại tiền tệ
4	Transaction	Lưu thông tin các hoạt động chi tiêu, thu nhập

5	Investment	Lưu thông tin đầu tư
6	InvestmentCategory	Lưu thông tin danh mục đầu tư
7	Budget	Lưu thông tin kế hoạch chi tiêu
8	PeriodicTransaction	Lưu thông tin các khoản chi tiêu dự định
9	TransactionCategory	Lưu thông tin các danh mục chi tiêu và thu nhập

3.3.5. Mô tả chi tiết các bảng

Mục đích: Có cái nhìn chi tiết và cụ thể về cơ sở dữ liệu về đề tài nghiên cứu

Khóa chính (Primary Key): là một giá trị dùng để phân biệt bản ghi này với bản ghi khác. Giá trị của khóa chính trong mỗi bản ghi là duy nhất trong mỗi bảng.

Khóa ngoại (Foreign Key): là một trường hay một nhóm trường trong một bản ghi của một bảng, trỏ đến khóa của một bản ghi khác của một bảng. Thông thường, khóa ngoại trong một bảng trỏ đến khóa chính của một bảng khác. Kiểu tham chiếu này có thể liên các thông tin lại với nhau và nó là một phần quan trọng của quá trình chuẩn hóa.

Bảng Customer

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã tài khoản
UserName	Varchar	20	Not null	Tên đăng nhập
Password	Varchar	255	Not null	Mật khẩu đăng nhập

FirstName	Varchar	50	Null	Họ và tên đệm
LastName	Varchar	20	Null	Tên
Gender	Bit		Null	Giới tính
Address	Varchar	255	Null	Địa chỉ
Phone	Varchar	15	Null	Điện thoại
Email	Varchar	100	Not null	Địa chỉ Email
Birthday	Varchar	20	Null	Ngày sinh

Bảng Wallet

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã Ví
CurrencyID	Int	10	Foreign Key	Mã tiền tệ
Total	Double	16	Not null	Số dư hiện tại
Name	Varchar	100	Not null	Tên Ví
CustomerID	Int	10	Foreign Key	Mã khách hàng

Bảng Currency

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã tiền tệ
Name	Varchar	100	Not null	Tên tiền tệ
Code	Varchar	10	Not null	Mã tiền tệ trên thị trường
Rate	Double	16	Not null	Tỷ giá so với VNĐ cập nhật liên tục

Bảng Transaction

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã giao dịch
Title	Varchar	100	Not null	Mô tả
Detail	Text		Null	Chi tiết
Amount	Double	16	Not null	Lượng tiền giao dịch
Note	Varchar	255	Null	Ghi chú
Location	Varchar	255	Null	Địa điểm giao dịch
WalletID	Int	10	Foreign	Mã ví tiền

			Key	
TransactionCategoryID	Int	10	Foreign Key	Mã danh mục chi tiêu, thu nhập
TransactionDate	Varchar	20	Not null	Ngày giao dịch

Bảng Investment

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã đầu tư
Title	Varchar	100	Not null	Mô tả
InvestmentDate	Varchar	20	Not null	Ngày đầu tư
Amount	Double	16	Not null	Lượng đầu tư (đơn vị ứng với danh mục đầu tư)
WalletID	Int	10	Foreign Key	Mã Ví tiền
InvestmentCategoryID	Int	10	Foreign Key	Mã danh mục đầu tư
Rate	Double	16	Not null	Tỷ giá của danh mục đầu tư tại thời điểm đầu tư

Bảng InvestmentCategory

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã danh mục đầu tư
Name	Varchar	100	Not null	Tên danh mục đầu tư
Code	Varchar	10	Not null	Mã danh mục đầu tư trên thị trường
Rate	Double	16	Not null	Tỷ giá quy ra USD cập nhật liên tục

Bảng Budget

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã kế hoạch
Title	Varchar	100	Not null	Mô tả
Detail	Text		Null	Chi tiết
Amount	Double	16	Not null	Lượng tiền giới hạn
Note	Varchar	255	Null	Ghi chú
WalletID	Int	10	Foreign Key, Null	Mã ví tiền

TransactionCategoryID	Int	10	Foreign Key	Mã danh mục chi tiêu, thu nhập
StartDate	Varchar	20	Not null	Ngày bắt đầu thực hiện
EndDate	Varchar	20	Not null	Ngày kết thúc
CurrencyID	Int	10	Foreign Key ,Null	Mã tiền tệ khi kế hoạch áp dụng với tất cả các ví
CustomerID	Int	10	Foreign Key, Null	Mã khách hàng khi kế hoạch áp dụng với tất cả các ví

Bảng PeriodicTransaction

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã Chi tiêu dự định
Title	Varchar	100	Not null	Mô tả
Detail	Text		Null	Chi tiết
Amount	Double	16	Not null	Lượng tiền giao dịch
Note	Varchar	255	Null	Ghi chú

Location	Varchar	255	Null	Địa điểm giao dịch
WalletID	Int	10	Foreign Key	Mã ví tiền
TransactionCategoryID	Int	10	Foreign Key	Mã danh mục chi tiêu, thu nhập
StartDate	Varchar	20	Not null	Ngày bắt đầu
EndDate	Varchar	20	Not null	Ngày kết thúc
Counter	Int	10	Null	Số lần đã chi trả
CounterLimited	Int	10	Null	Số lần giới hạn
DayOfMonth	Int	10	Null	Tần suất chi trả vào ngày X hàng tháng
DayOfWeek	Int	10	Null	Tần suất chi trả vào thứ X hàng tuần
EveryXDays	Int	10	Null	Tần suất chi trả sau X Ngày

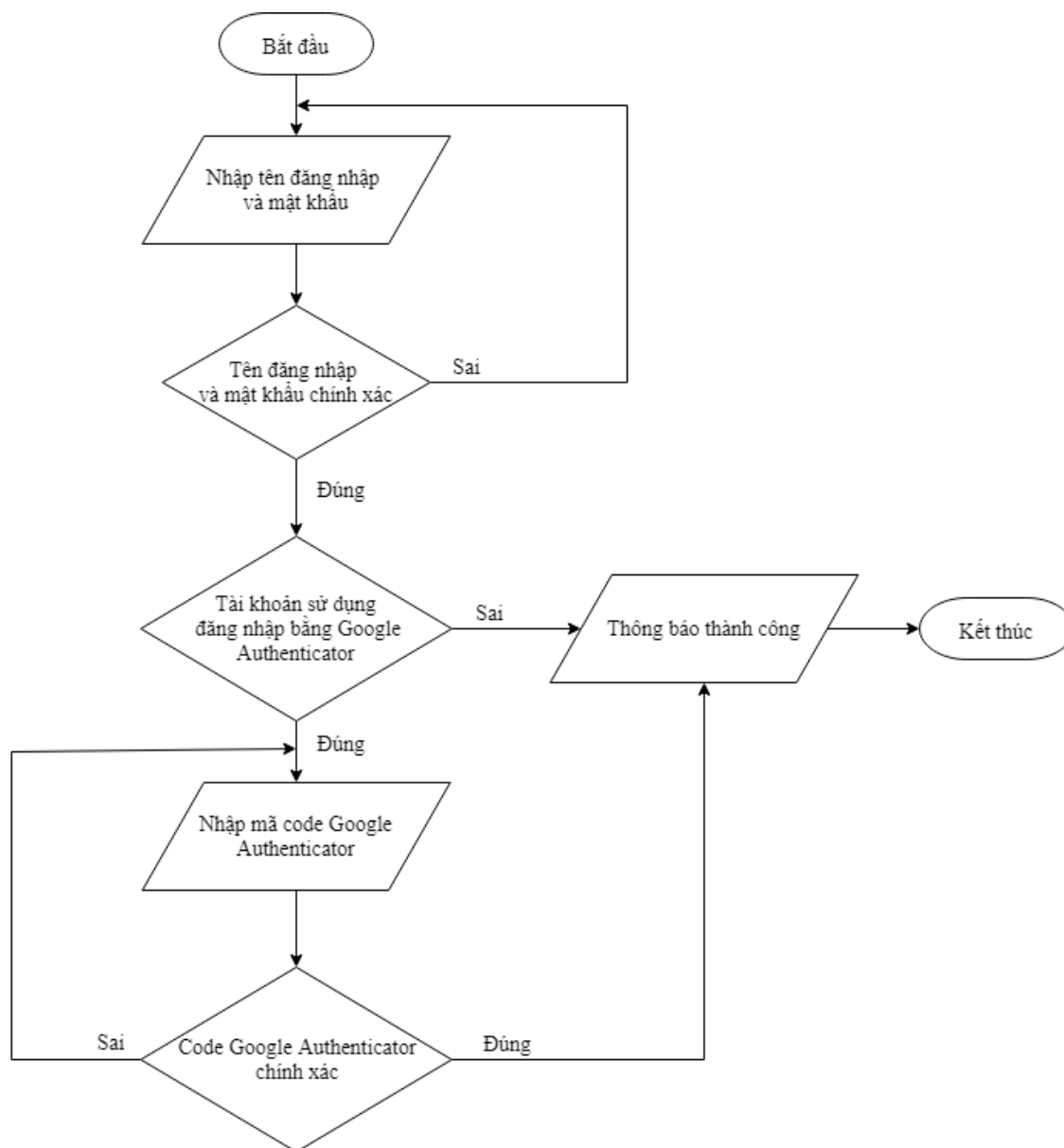
Bảng TransactionCategory

Tên trường	Kiểu dữ liệu	Kích cỡ	Tính chất	Diễn giải
ID	Int	10	Primary key	Mã danh mục chi tiêu, thu

				nhập
Name	Varchar	100	Not null	Tên danh mục chi tiêu, thu nhập
Type	Int	10	Not null	Loại danh mục: 0 – Thu nhập, 1- Chi tiêu

3.4. Giới thiệu một số sơ đồ thuật toán sử dụng trong Website

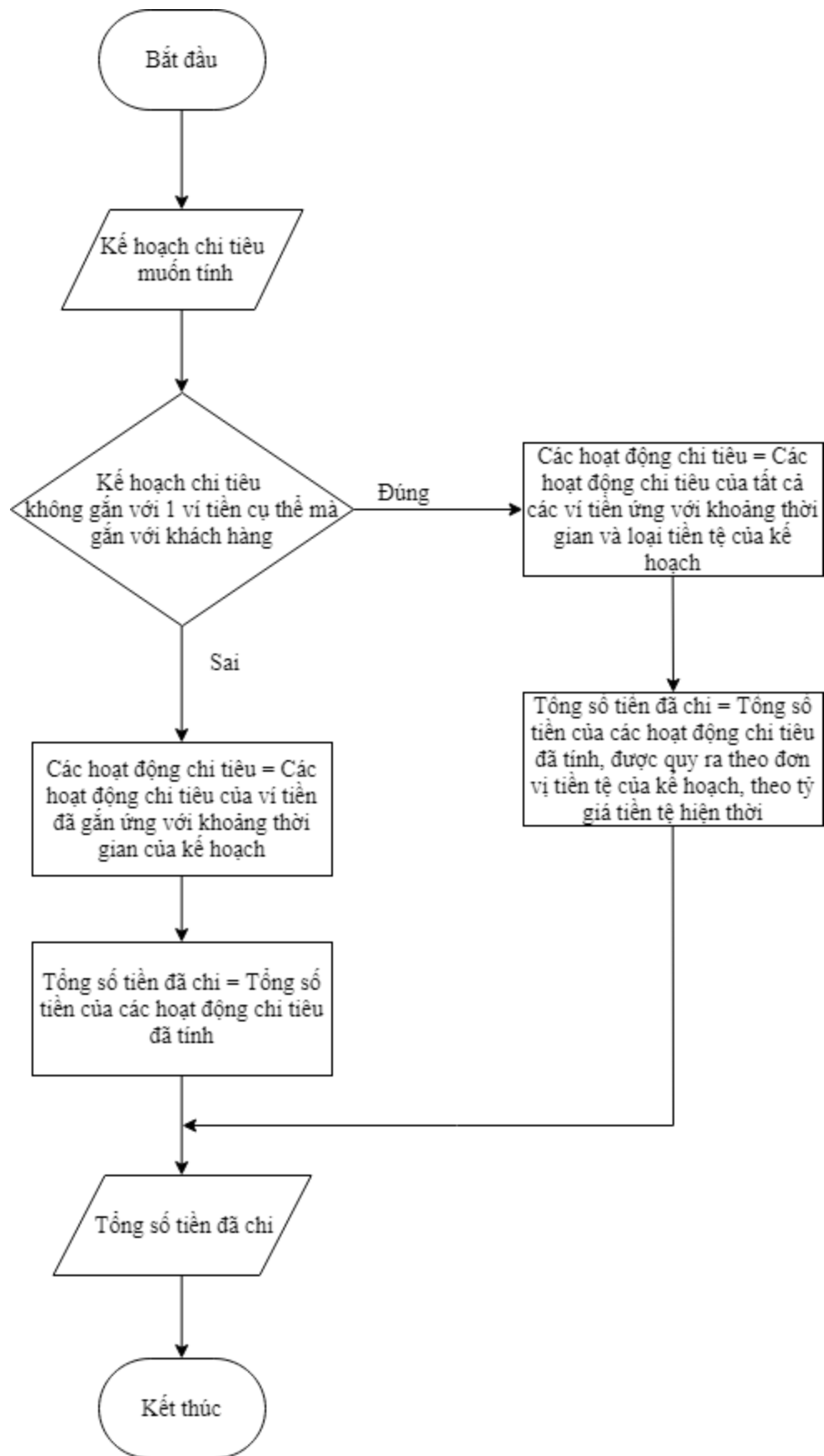
3.4.1. Sơ đồ thuật toán đăng nhập



Hình 3.20. Sơ đồ thuật toán đăng nhập

3.4.2. Sơ đồ thuật toán chức năng Kế hoạch chi tiêu

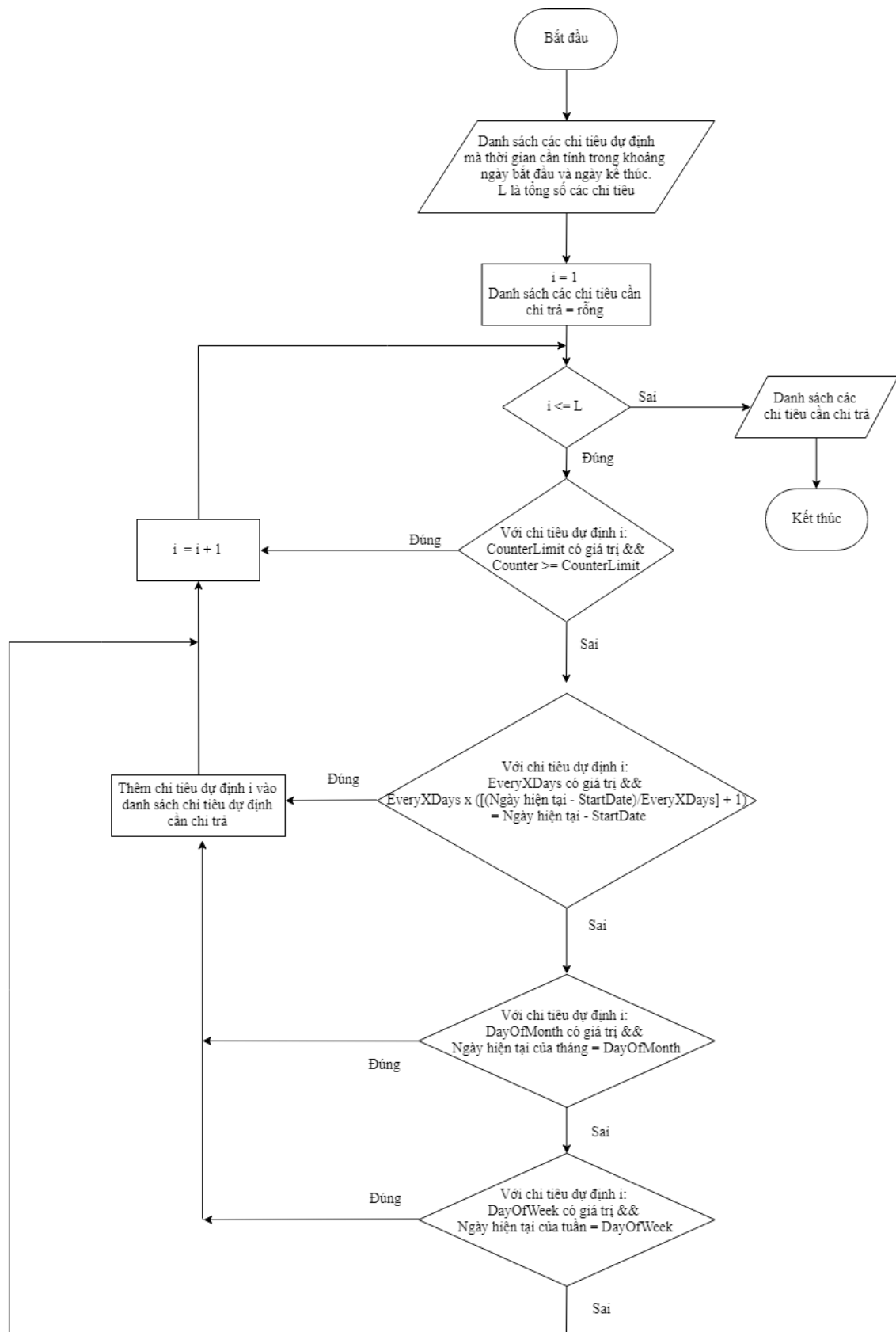
Mô tả: Cần kiểm tra lượng tiền độ thực hiện một kế hoạch chi tiêu cho trước. Đầu vào là kế hoạch muốn kiểm tra, đầu ra là tổng số tiền đã chi thuộc kế hoạch.



Hình 3.21. Sơ đồ thuật toán chức năng Kế hoạch chi tiêu

3.4.3. Sơ đồ thuật toán chức năng Chi tiêu dự định

Mô tả: Tại một thời điểm, cần kiểm tra những Chi tiêu dự định nào đã thiết lập đến thời điểm chi trả. Đầu vào là danh sách Chi tiêu dự định mà thời điểm muốn kiểm tra nằm trong ngày bắt đầu và ngày kết thúc, đầu ra là danh sách Chi tiêu dự định cần chi trả.



Hình 3.21. Sơ đồ thuật toán chức năng Chỉ tiêu dự định

Chương IV: GIỚI THIỆU MỘT SỐ GIAO DIỆN CHÍNH CỦA WEBSITE

4.1. Giới thiệu một số giao diện chính của Website

4.1.1. Giao diện trang chủ khách vãng lai





Ngân sách

Dễ dàng lập kế hoạch và quản lý các hoạt động chi tiêu.



Hóa đơn

Theo dõi các hóa đơn, nhận thông báo mọi lúc, mọi nơi.



Đầu tư

Thông tin liên tục và chính xác về các thị trường đầu tư.

DLYM, cuộc cách mạng tài chính cá nhân!

Chúng tôi có mọi thứ để giúp bạn chi tiêu hợp lý và đầu tư hiệu quả.



DỄ DÀNG THEO DÕI THU CHI

Nhanh chóng ghi chép và phân loại các khoản chi tiêu hoặc thu nhập.

Báo cáo trực quan với biểu đồ theo tuần, tháng hoặc năm.

Thêm ghi chú cũng như phân loại theo nhóm thuận tiện.

QUẢN LÝ DANH MỤC ĐẦU TƯ TIỆN LỢI

Ghi chép và phân loại các thông tin đầu tư.

Cập nhật thông tin về các danh mục chính xác và nhanh chóng.

Phân tích trực quan tình hình thị trường đầu tư.



LÊN KẾ HOẠCH CHI TIÊU VÀ TIẾT KIỆM

Lên kế hoạch chi tiêu như ăn uống, du lịch (Ngân sách).

Tự động nhắc nhở khi bạn đã tiêu gần mức hạn mức đã đề ra.

Theo dõi quá trình tiết kiệm giúp bạn có động lực tiết kiệm nhiều hơn.

DLYM, tính năng trực quan, kết quả mạnh mẽ!

DLYM sẽ thay bạn quản lý tài chính hiệu quả. Chủ động quản lý chi tiêu và đầu tư, lên kế hoạch cho tương lai. DLYM sẽ thay đổi cuộc sống của bạn.



Ngân sách hoạt động

Tạo nhiều ngân sách cùng hoạt động, theo dõi các hoạt động chi tiêu.



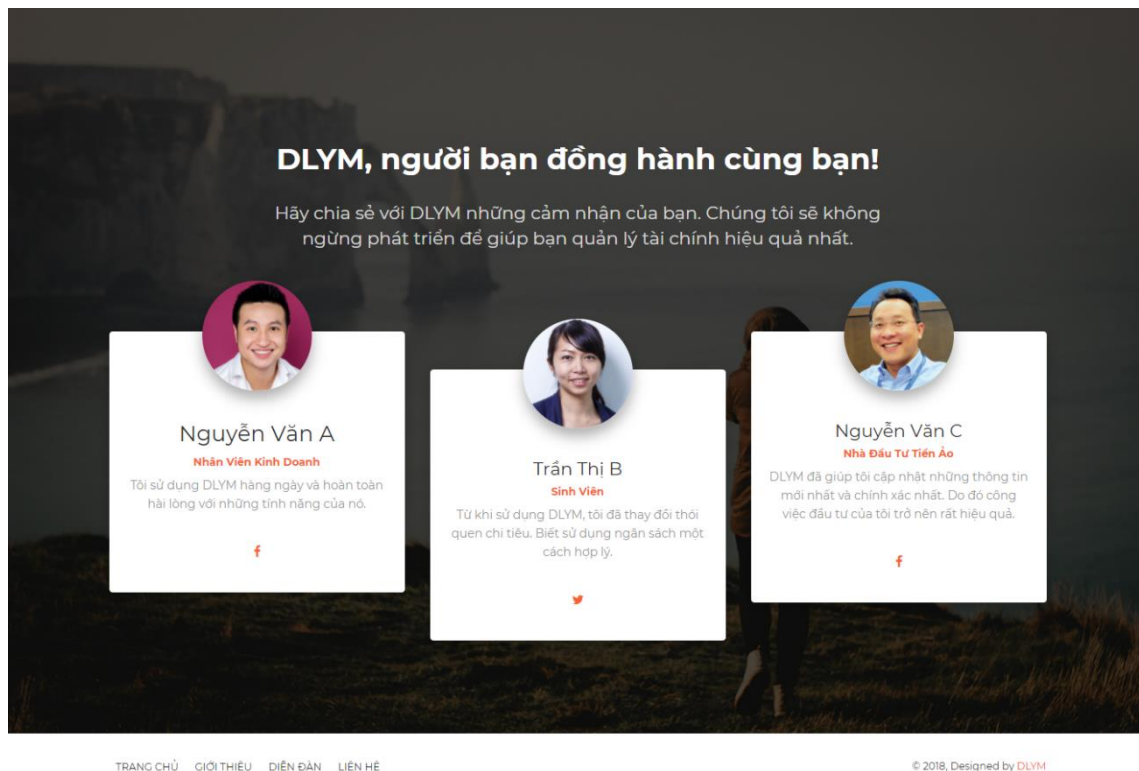
Sử dụng mọi lúc, mọi nơi

Quản lý tài chính của bạn ngay trên điện thoại hoặc máy tính bảng.



Hỗ trợ tối đa

Nhận các cảnh báo chi tiêu và lời khuyên được cá nhân hóa để tối đa hóa lợi ích.



Hình 4.1. Giao diện trang chủ khách vãng lai

KẾT LUẬN

Website quản lý danh mục đầu tư và chi tiêu cá nhân DLYM là kết quả của quá trình nghiên cứu lý thuyết và vận dụng kiến thức để xây dựng. Website đã giải quyết các vấn đề đặt ra và hoàn thiện các chức năng như:

- + Giao diện thân thiện với người dùng, tối ưu trên các thiết bị di động, ipad.
- + Khách hàng có thể quản lý chi tiêu, thu nhập và đầu tư một cách thuận tiện nhất
- + Khách hàng có thể lập kế hoạch chi tiêu và các chi tiêu dự định để tối ưu hóa các khoản chi tiêu sinh hoạt.
- + Nhà quản trị có thể quản lý các danh mục, tài khoản khách hàng, phân quyền quản lý, khách hàng.
- + Website được bảo mật dạng MD5, validation, đăng nhập 2 lớp sử dụng Google Authenticator .

Bên cạnh đó website còn có một số các hạn chế như sau:

- + Công cụ làm website vẫn chưa tối ưu nhất, chưa cập nhật mới nhất.
- + Giao tiếp với khách hàng còn hạn chế.

Trong thời gian tới em sẽ cố gắng học hỏi thêm nhiều kinh nghiệm để tích lũy cho mình nhiều kiến thức và hoàn thiện website , khắc phục những nhược điểm kể trên, up nên thông tin tên miền + hosting đi vào hoạt động, và xây dựng thêm một vài chức năng mới.

TÀI LIỆU THAM KHẢO

1. Website học lập trình trực tuyến và miễn phí

<https://www.w3schools.com/>

<http://docs.microsoft.com/en-us/aspnet/mvc>

<http://www.asp.net/mvc>

<https://tedu.com.vn>

Tài liệu Tiếng Việt

2. Nguyễn Văn Ba (2006) Phân tích thiết kế hệ thống thông tin , NXB Đại học Quốc Gia Hà Nội.
3. T.S Nguyễn Trung Tuấn, ThS Tống Minh Ngọc Cơ sở dữ liệu.
4. Slide bài giảng Phân tích thiết kế hệ thống – Trần Thị Mỹ Diệp – Đại học Kinh tế quốc dân.
5. Bài giảng ASP.NET MVC 5 – Lại Đức Chung - Trường đào tạo CNTT Quốc tế Bachkhoa - Aptech

Tài liệu Tiếng Anh

5. Programming ASP.NET MVC 5 A Problem Solution Approach – Nimit Joshi.
6. Pro ASP.Net MVC 5 – Adam Freeman – Appress.