

**TRƯ**

**Ờ**

**NG Đ**

**Ạ**

**I H**

**Ọ**

**C KINH T**

**Ế**

**QU**

**Ố**

**C DÂN**

**VI**

**Ệ**

**N CÔNG NGH**

**Ệ**

**THÔNG TIN KINH T**

**Ế**

**----**

**o0o**

**----**

**CHUYÊN Đ**

**Ề**

**TH**

**Ự**

**C T**

**Ậ**

**P T**

**Ố**

**T NGHI**

**Ệ**

**P**

***Đ***

***ề***

***tài:***

**XÂY DỰNG GAME NHẬP VAI TERRESURVIVAL**

**TRÊN UNITY**

***Gi***

***ả***

***ng viên hư***

***ớ***

***ng d***

***ẫ***

***n***

**:**

**ThS.**

**Nguyễn Thanh Hương**

***H***

***ọ***

***và tên***

**:**

**Lù Chín Trình**

***Mã sinh viên***

**:**

**11208209**

***L***

***ớ***

***p***

**:**

**Khoa học máy tính 62**

**HÀ N**

**Ộ**

**I,**

**2024**

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc93170)

[CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI 3](#_Toc93171)

[1.1. Giới thiệu chung 3](#_Toc93172)

[1.2. Yêu cầu đặt ra 3](#_Toc93173)

[1.3. Mục tiêu của đề tài 4](#_Toc93174)

[1.4. Phạm vi đề tài 5](#_Toc93175)

[Chương II: GIỚI THIỆU VỀ CÁC CÔNG NGHỆ SỬ DỤNG 6](#_Toc93176)

[2.1. Công nghệ Web 6](#_Toc93177)

[2.1.1. ASP.NET MVC 6](#_Toc93178)

[2.1.2. Ngôn ngữ lập trình C# 13](#_Toc93179)

[2.2. Công nghệ xử lý giao diện 14](#_Toc93180)

[2.2.1. HTML (Hypertext Markup Language) 14](#_Toc93181)

[2.2.2. CSS (Cascading Style Sheets) 15](#_Toc93182)

[2.2.3. JavaScript 15](#_Toc93183)

[2.3. Một số Third – Party 16](#_Toc93184)

[Chương III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 19](#_Toc93185)

[3.1. Mô tả hệ thống 19](#_Toc93186)

[3.1.1. Sơ đồ kiến trúc hệ thống 19](#_Toc93187)

[3.1.2. Sơ đồ chức năng của hệ thống 20](#_Toc93188)

[3.1.3. Chức năng đối với khách hàng 20](#_Toc93189)

[3.1.4. Chức năng đối với quản trị Website 22](#_Toc93190)

[3.2. Phân tích hệ thống về mặt chức năng 23](#_Toc93191)

[3.2.1. Biều đồ luồng dữ liệu (BLD) mức khung cảnh 23](#_Toc93192)

[3.2.2. Biều đồ luồng dữ liệu (BLD) mức đỉnh 24](#_Toc93193)

[3.2.3. Biều đồ luồng dữ liệu (BLD) mức dưới đỉnh 25](#_Toc93194)

[3.3. Phân tích hệ thống về cơ sở dữ liệu 27](#_Toc93195)

[3.3.1. Xác định các thực thể 27](#_Toc93196)

[3.3.2. Sơ đồ thực thể liên kết 32](#_Toc93197)

[3.3.3. Mô hình quan hệ giữa các thực thể 36](#_Toc93198)

[3.3.4. Danh sách các bảng dữ liệu 36](#_Toc93199)

[3.3.5. Mô tả chi tiết các bảng 37](#_Toc93200)

[3.4. Giới thiệu một số sơ đồ thuật toán sử dụng trong Website 45](#_Toc93201)

[3.4.1. Sơ đồ thuật toán đăng nhập 45](#_Toc93202)

[3.4.2. Sơ đồ thuật toán chức năng Kế hoạch chi tiêu 45](#_Toc93203)

[3.4.3. Sơ đồ thuật toán chức năng Chi tiêu dự định 47](#_Toc93204)

[Chương IV: GIỚI THIỆU MỘT SỐ GIAO DIỆN CHÍNH CỦA WEBSITE 49](#_Toc93205)

[4.1. Giới thiệu một số giao diện chính của Website 49](#_Toc93206)

[4.1.1. Giao diện trang chủ khách vãng lai 49](#_Toc93207)

[KẾT LUẬN 52](#_Toc93208)

[TÀI LIỆU THAM KHẢO 53](#_Toc93209)

ii

**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 Cô Nguyễn Thanh Hương 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 2 năm 2024

Sinh viên

Lù Chín Trình

# LỜI NÓI ĐẦU

Trong thời đại số hóa ngày nay, trò chơi điện tử không chỉ là một hình thức giải trí mà còn là một công cụ mạnh mẽ để kể câu chuyện, khám phá thế giới và tạo ra trải nghiệm độc đáo cho người chơi. Với sự phát triển không ngừng của công nghệ, chúng ta có thể thấy sự tiến bộ đáng kinh ngạc trong việc tạo ra những trò chơi đồ họa và gameplay ngày càng ấn tượng.

Trong dự án của em, em đã đặt ra mục tiêu phát triển một trò chơi nhập vai RPG 2D trên Unity có tên là **“TerreSurvival”** với một thế giới đầy màu sắc và hấp dẫn. Em muốn mang đến cho người chơi không chỉ là một trò chơi mà còn là một cuộc phiêu lưu đầy cảm xúc và ý nghĩa.

Trò chơi của em sẽ cho phép người chơi nhập vai vào vai một nhân vật, khám phá các vùng đất đa dạng, đối đầu với những thử thách khác nhau, và tương tác với các nhân vật trong trò chơi để tiến bộ trong cuộc phiêu lưu của họ. Em hy vọng rằng trò chơi này sẽ mang lại cho người chơi những trải nghiệm đáng nhớ và động viên họ khám phá và tin tưởng vào khả năng của bản thân.

Em đã đầu tư rất nhiều công sức và tâm huyết vào dự án này, và em rất mong muốn được chia sẻ nó với mọi người. Chúng tôi tin rằng với sự đồng lòng và hỗ trợ từ cộng đồng, dự án của em sẽ trở thành một thành công lớn.

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 Game**

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

**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

**“TerreSurvival”** là một trò chơi nhập vai RPG 2D được xây dựng dựa trên Unity, nơi người chơi sẽ khám phá một thế giới đầy màu sắc và nguy hiểm. Họ có thể tự do di chuyển trên bản đồ, chiến đấu với quái vật, thu thập exp và vật phẩm để tăng cấp và nâng cấp vũ khí. Mục tiêu cuối cùng của họ là đạt được chiến thắng bằng cách đi đến điểm thoát hiểm cuối cùng.

Trong TerreSurvival, người chơi sẽ đối mặt với một loạt các thử thách và quái vật đáng sợ trong hành trình của mình. Họ có thể chiến đấu một mình hoặc hợp tác với những người chơi khác để vượt qua những tình huống khó khăn.

Hệ thống nâng cấp vũ khí và trang bị trong trò chơi cho phép người chơi tùy chỉnh nhân vật của mình theo cách họ muốn và phát triển chiến thuật riêng của mình.

Với đồ họa 2D đẹp mắt và gameplay hấp dẫn, TerreSurvival hứa hẹn mang đến cho người chơi những trải nghiệm đích thực của một cuộc phiêu lưu trong thế giới game..

# 1.2. Yêu cầu đặt ra

Yêu cầu cho trò chơi TerreSurvival bao gồm:

* **Gameplay Di chuyển và Tấn công**: Tạo hệ thống di chuyển và tấn công cho nhân vật chính.
* **Hệ Thống Vũ Khí và Chiêu Thức**: Phát triển vũ khí đa dạng và các chiêu thức tương ứng.
* **Animation Cho Hành Động của Nhân Vật Với Vũ Khí**: Tạo animation cho các hành động di chuyển và tấn công của nhân vật.
* **Hệ Thống Quái Vật**: Xây dựng các loại quái vật đa dạng và hệ thống AI để chúng tìm đến và tấn công người chơi.
* **Hệ Thống Âm Thanh và Hiệu Ứng**: Tích hợp âm thanh và hiệu ứng đặc biệt để tạo ra trải nghiệm âm nhạc và hấp dẫn cho người chơi.

# 1.3. Mục tiêu của đề tài

Mục tiêu của dự án TerreSurvival không chỉ đơn giản là tạo ra một trò chơi nhập vai RPG 2D trên nền tảng Unity, mà còn là xây dựng một thế giới ảo độc đáo và đầy kịch tính, nơi mà em có thể bước vào và trải nghiệm một cuộc phiêu lưu tuyệt vời.

Em muốn tạo ra một trò chơi mà không chỉ làm hài lòng người chơi bằng cách cung cấp cho họ một trải nghiệm giải trí thú vị, mà còn đem lại cho họ cảm giác như đang sống trong một thế giới hoàn toàn mới, với những khám phá và thử thách không ngừng.

Mục tiêu của em là xây dựng một hệ thống gameplay linh hoạt và phong phú, từ việc di chuyển trên bản đồ, tấn công quái vật, thu thập trang bị và kỹ năng, cho đến việc tiến bộ và nâng cấp nhân vật của mình. Em muốn mỗi người chơi có thể tùy chỉnh nhân vật của mình theo cách riêng, phát triển chiến thuật và phong cách chơi của mình trong cuộc phiêu lưu.

Ngoài ra, em cũng đặt ra mục tiêu xây dựng một thế giới game đa dạng và phong phú, với nhiều loại môi trường khác nhau và đủ loại quái vật để người chơi có thể khám phá và chiến đấu. Mỗi khu vực trong trò chơi đều được thiết kế cẩn thận, với đồ họa đẹp mắt và các yếu tố tạo cảm giác sống động, từ rừng sâu tới hang động huyền bí.

Bên cạnh đó, một mục tiêu quan trọng khác của em là xây dựng một hệ thống AI thông minh cho quái vật, giúp tạo ra những trận đấu hấp dẫn và đầy thách thức. Em muốn mỗi trận đấu không chỉ là một cuộc chiến đấu đơn giản, mà còn là một thử thách chiến lược và tư duy.

Cuối cùng, em muốn tạo ra một trò chơi có sức hút lâu dài và đem lại cho người chơi những trải nghiệm độc đáo và không quên. Em tin rằng TerreSurvival sẽ là một cuộc phiêu lưu đích thực và đầy cảm xúc, và em không ngừng nỗ lực để đạt được mục tiêu đó..

# 1.4. Phạm vi đề tài

Phạm vi của đề tài TerreSurvival bao gồm các yếu tố sau:

* **Gameplay và Hệ Thống Chiến Đấu**: Phát triển hệ thống di chuyển, tấn công và phòng thủ cho nhân vật chính, bao gồm cả việc xử lý va chạm và đấu tranh với quái vật. Tạo ra các kỹ năng và chiêu thức đặc biệt cho từng loại vũ khí, cùng với các hiệu ứng và animation tương ứng.
* **Thế Giới và Môi Trường**: Thiết kế và triển khai một thế giới mở rộng đa dạng với nhiều khu vực khác nhau như rừng, hang động, và thị trấn hoang tàn. Mỗi khu vực sẽ có độ khó và yếu tố nguy hiểm riêng, yêu cầu người chơi phải áp dụng chiến thuật và kỹ năng phản ứng.
* **Hệ Thống Vũ Khí và Trang Bị**: Xây dựng một loạt các loại vũ khí và trang bị đa dạng, từ vũ khí cận chiến đến vũ khí từ xa và các bộ giáp bảo vệ. Người chơi sẽ có thể thu thập và nâng cấp trang bị của họ trong suốt cuộc phiêu lưu.
* **Hệ Thống Quái Vật và AI**: Tạo ra một loạt các loại quái vật với hành vi và kỹ năng đa dạng. Phát triển hệ thống AI để điều khiển chúng, bao gồm cả khả năng tìm đường và tấn công người chơi một cách thông minh.
* Âm Thanh và Hiệu Ứng: Tích hợp âm nhạc, hiệu ứng âm thanh và hiệu ứng hình ảnh phù hợp, tạo ra một bầu không khí sống động và kích thích cho trò chơi.
* **Giao Diện Người Dùng:** Thiết kế giao diện người dùng thân thiện và dễ sử dụng, cho phép người chơi dễ dàng tương tác với trò chơi và quản lý nhân vật và trang bị của mình.
* **Tính Tái Chơi và Nâng Cấp**: Tạo ra một trải nghiệm có tính tái chơi cao, với nhiều hoạt động phụ và mục tiêu phụ để người chơi có thể thử thách bản thân. Cung cấp hệ thống nâng cấp và phát triển nhân vật liên tục để duy trì sự hấp dẫn của trò chơi.

# Chương II: GIỚI THIỆU VỀ CÁC CÔNG NGHỆ SỬ DỤNG

# 2.1. Công nghệ Game.

### 2.1.1. Unity.

Unity là một hệ thống game đa nền tảng được phát triển bởi Unity Technologies, được công bố và phát hành lần đầu vào tháng 6 năm 2005 tại Hội nghị Nhà phát triển Toàn cầu của Apple như là một công cụ phát triển trò chơi trên Mac OS X. Từ đó, hệ thống đã dần được mở rộng để hỗ trợ nhiều nền tảng khác nhau như máy tính để bàn, di động, console và thực tế ảo. Nó đặc biệt phổ biến cho việc phát triển trò chơi di động trên iOS và Android, được xem là dễ sử dụng đối với các nhà phát triển mới bắt đầu, và được ưa chuộng trong việc phát triển trò chơi độc lập.

Hệ thống này có thể được sử dụng để tạo ra các trò chơi ba chiều (3D) và hai chiều (2D), cũng như các mô phỏng tương tác và các trải nghiệm khác. Nó đã được áp dụng rất nhiều trong ngành phát triển game.

*2.1.1.1. Ưu điểm, nhược điểm của Unity.*

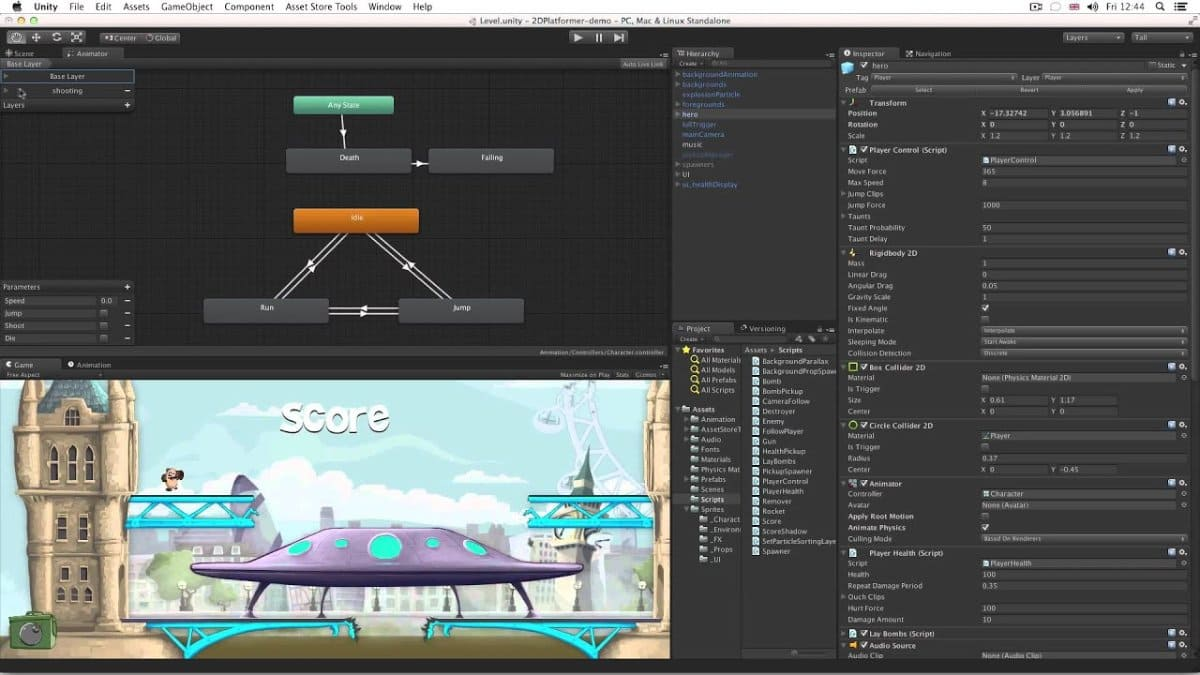
a. Ưu điểm.

Unity tạo nên một môi trường phát triển tích hợp (IDE) mạnh mẽ, giúp các nhà lập trình tạo, thử nghiệm và xuất bản các ứng dụng trò chơi. Nhưng để trở thành lựa chọn hàng đầu của đông đảo lập trình viên trên thế giới thì phần mềm này có nhiều tính năng đặc biệt như:

**Công cụ tạo trò chơi**

Tính đến năm 2018, Unity được sử dụng để tạo ra hơn 50% trò chơi mobile và 60% nội dung thực tế ảo. Và là phần mềm dùng để sản xuất các game nổi tiếng như Pokemon Go, Call of Duty: Mobile, Cuphead,…

Không chỉ được tích hợp hệ thống vật lý 2D và 3D, giúp xử lý các tính toán va chạm và động lực học trong game. Mà công cụ này sử dụng C# - ngôn ngữ lập trình quen thuộc với hầu hết developer.



Hình 1. Tạo trò chơi 2D và 3D chuyên nghiệp.

Bên cạnh đó, công cụ này cung cấp các công cụ và tài nguyên đồ họa để tạo ra các hiệu ứng đẹp mắt trong game. Đặc biệt là còn hỗ trợ đa nền tảng cho phép developer xuất trò chơi trên Windows, macOS, Linux, Android, iOS, PlayStation,… mà không cần thay đổi nhiều trong mã nguồn.

**Cộng đồng hỗ trợ và kho tài nguyên**

Unity có một cộng đồng gọi là Forum Unity, nơi hỗ trợ người dùng về cách sử dụng, cập nhật phiên bản hoặc xử lý lỗi. Nhờ có cộng đồng này mà các developer game không phải tự mình “mày mò” khi sử dụng.

Bên cạnh đó, Unity có một kho tài nguyên, gọi là “chợ plug-in”. Ở đây, bạn có thể tải về các gói như mô hình 3D, vật liệu, âm thanh, script và nhiều thứ khác, giúp tiết kiệm thời gian làm dự án.



Hình 2. Kho tài nguyên game của Unity.

b. Nhược điểm.

*2.1.1.2. MVC (Model – View – Controller)*

*2.1.1.3. ASP.NET MVC*

*2.1.1.4. ASP.NET MVC 5*

### 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 các 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 hướng 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 ngũ kịch bản được phát triển bởi Sun Microsystem.

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.

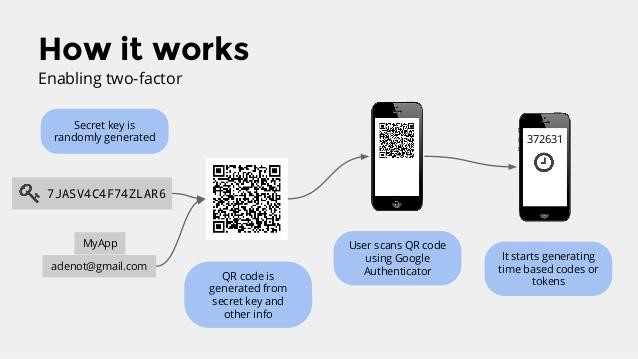
TOTP về cơ bản sử dụng cùng một thuật toán như HOTP với một sự khác biệt lớn. Bộ đếm được sử dụng trong TOTP được thay thế bằng thời gian hiện tại. Máy khách và máy chủ vẫn được đồng bộ miễn là thời gian của hệ thống vẫn giữ nguyên. Điều này có thể được thực hiện bằng cách sử dụng giao thức Thời gian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



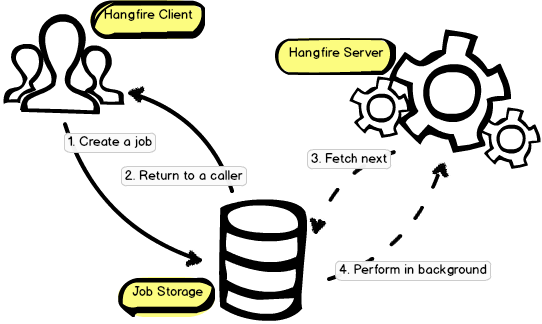
Hình 2.4. Kiến trúc Google Authenticator

##### *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 nền được tạo ra một cách nguyên tử.
* **Batch** **Continuations**: Tiếp tục hàng loạt được kích hoạt sau khi tất cả các công việc nền 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

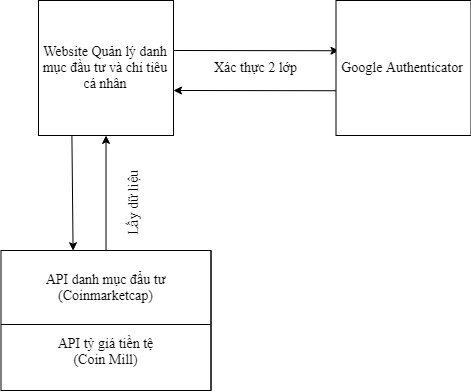
# Chương III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

# 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

Qu

ả

n lý Website

Qu

ả

n lý Khách hàng

Qu

ả

n tr

ị

h

ệ

th

ố

ng

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 đ**

**ồ**

**ch**

**ứ**

**c năng:**

Qu

ả

n lý

Khách hàng

Đăng nh

ậ

p thành viên

Qu

ả

n lý Ví ti

ề

n

Qu

ả

n lý ho

ạ

t đ

ộ

ng

chi tiêu,

thu nh

ậ

p

Qu

ả

n lý danh m

ụ

c

đ

ầ

u tư

Qu

ả

n lý k

ế

ho

ạ

ch chi

tiêu

Qu

ả

n lý chi tiêu d

ự

đ

ị

nh

Báo cáo th

ố

ng kê

Chuy

ể

n ti

ề

n gi

ữ

a các

Ví ti

ề

n

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:**

Qu

ả

n lý

Website

Qu

ả

n lý tài kho

ả

n

thành viên

Qu

ả

n lý các danh

m

ụ

c

Qu

ả

n lý c

ấ

u hình h

ệ

th

ố

ng

Th

ố

ng kê báo cáo

Phân quy

ề

n tài kho

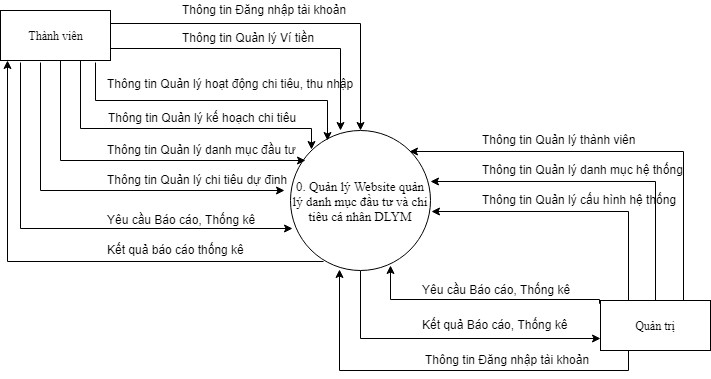
ả

n

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

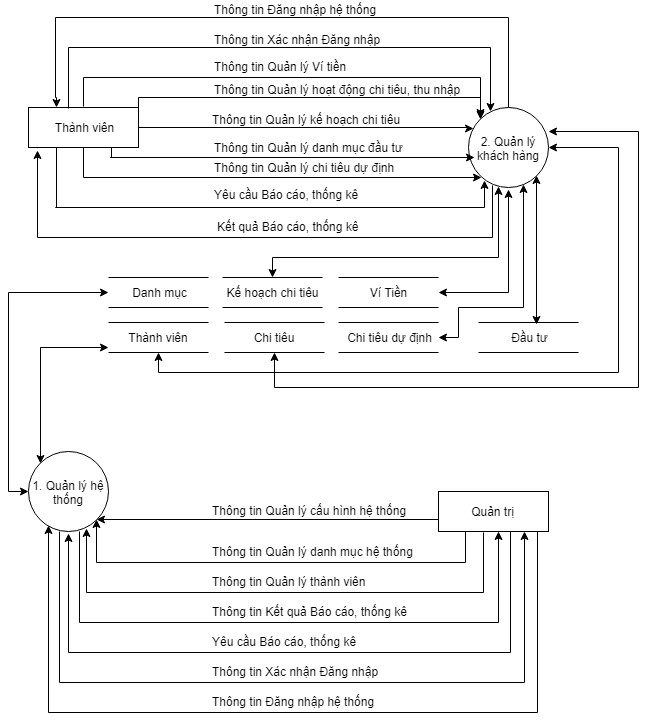
# 3.2. Phân tích hệ thống về mặt chức năng

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



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

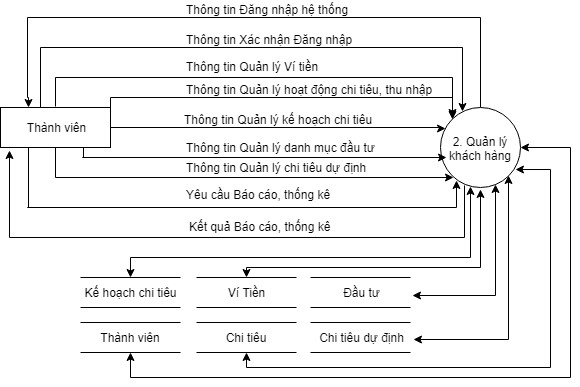
### 3.2.2. Biều đồ luồng dữ liệu (BLD) mức đỉnh



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

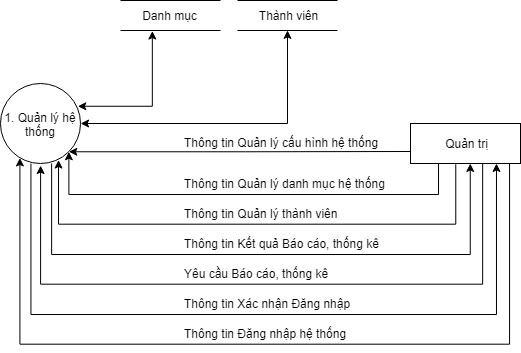
### 3.2.3. Biều đồ luồng dữ liệu (BLD) mức dưới đỉnh

3.2.3.1. Đối với chức năng quản lý khách hàng



Hình 3.7. Biều 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



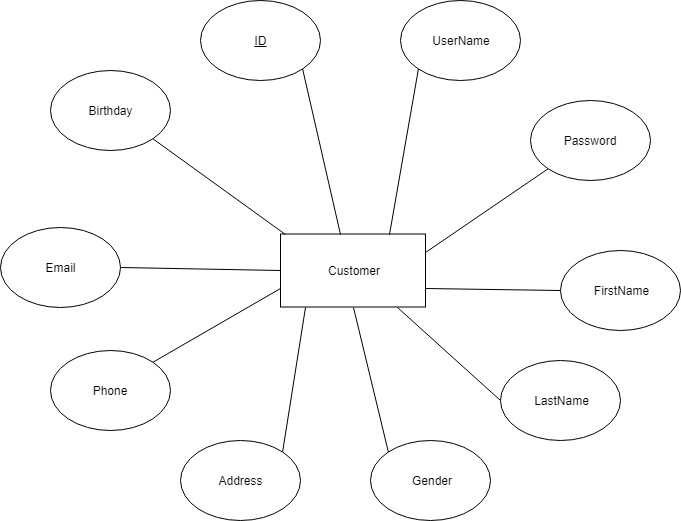
Hình 3.8. Biều đồ luồng dữ liệu (BLD) mức dưới đỉnh

đối với chức năng quản lý hệ thống

# 3.3. Phân tích hệ thống về cơ sở dữ liệu

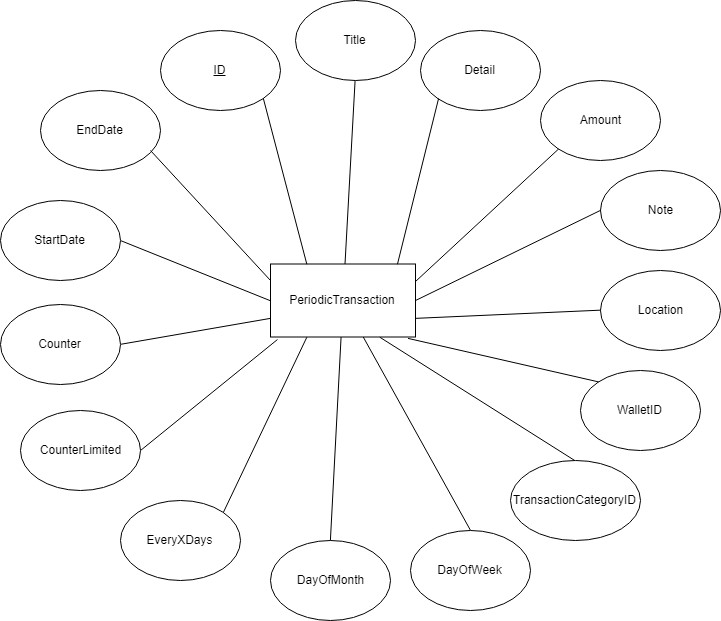
### 3.3.1. Xác định các thực thể

Thực thể Khách hàng(Customer):



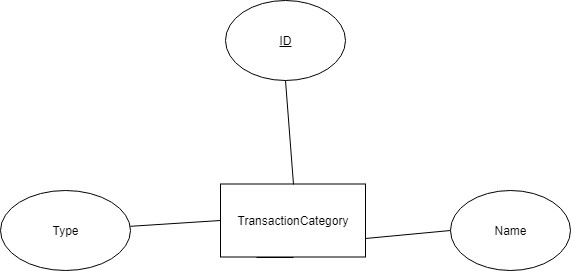
Hình 3.9. Thực thể Khách hàng(Customer)

Thực thể Chi tiêu dự định (PeriodicTransaction)



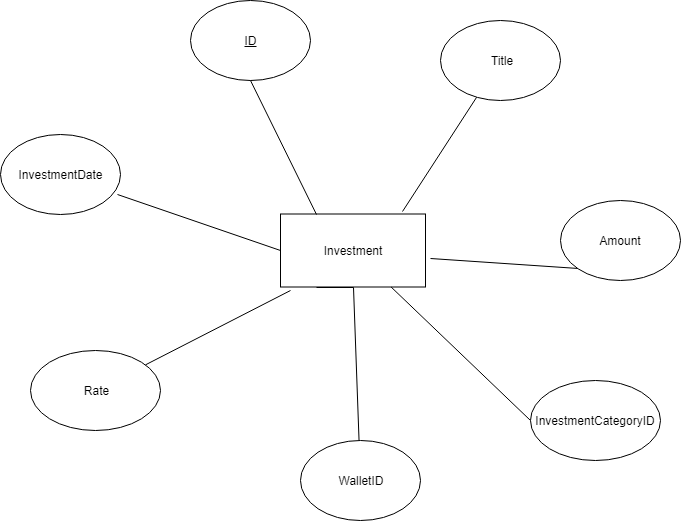
Hình 3.10. Thực thể Chi tiêu dự định (PeriodicTransaction)

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



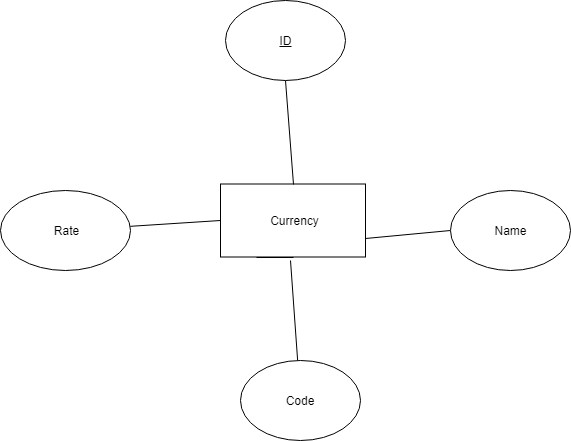
Hình 3.11. Thực thể Danh mục chi tiêu, thu nhập (TransactionCategory)

Thực thể Kế hoạch (Budget)



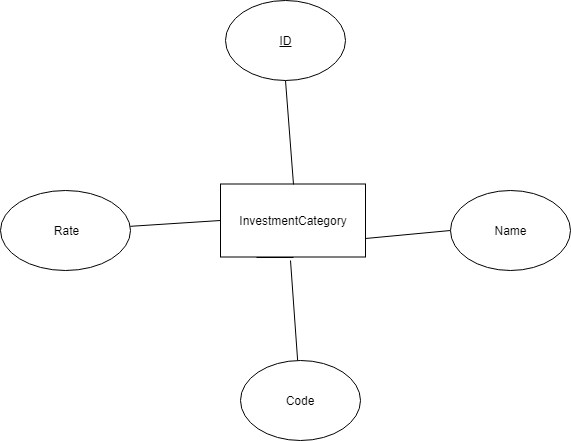
Hình 3.12. Thực thể Kế hoạch (Budget)

Thực thể Tiền tệ (Currency)



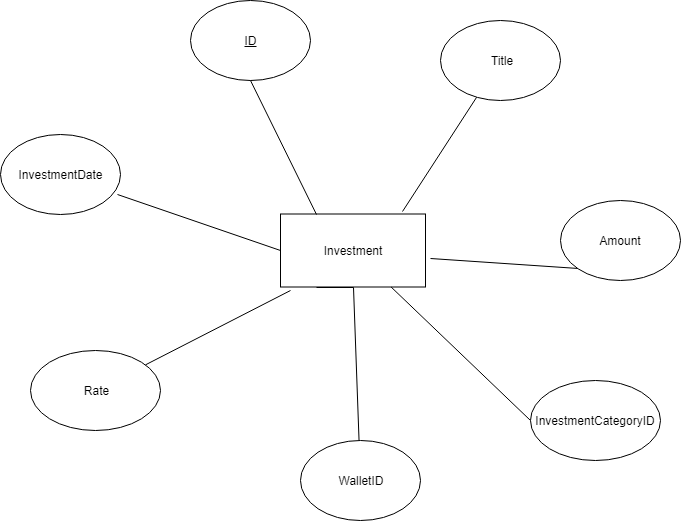
Hình 3.13. Thực thể Tiền tệ (Currency)

Thực thể Danh mục đầu tư (InvestmentCategory)



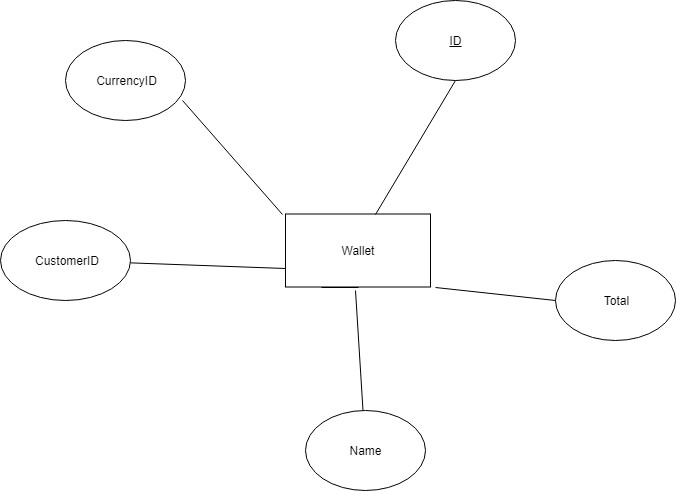
Hình 3.14. Thực thể Danh mục đầu tư (InvestmentCategory)

Thực thể Đầu tư (Investment)



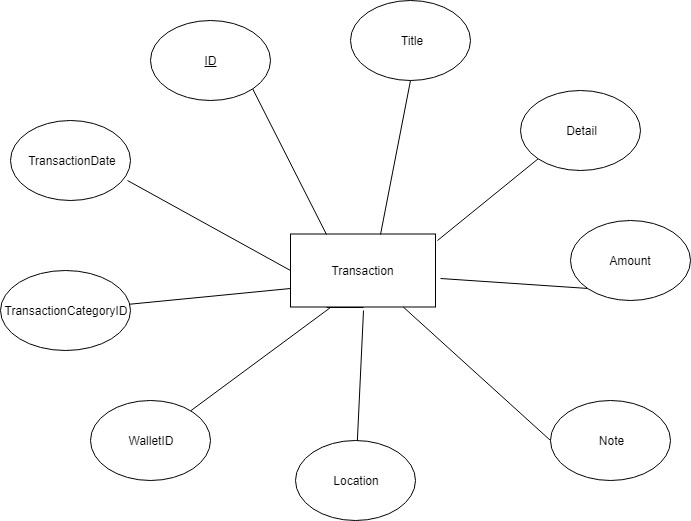
Hình 3.15. Thực thể Đầu tư (Investment)

Thực thể Ví người dùng (Wallet)



Hình 3.16. Thực thể Ví người dùng (Wallet)

Thực thể Chi tiêu, thu nhập (Transaction)

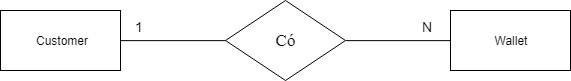


Hình 3.17. Thực thể Chi tiêu, thu nhập (Transaction)

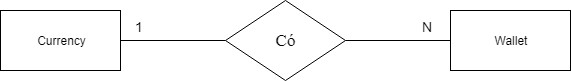
### 3.3.2. Sơ đồ thực thể liên kết

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.



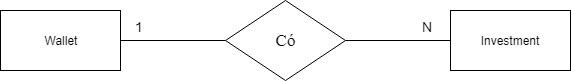
Mỗi ví tiền khách hàng thuộc một loại tiền tệ. Do đó, liên kết giữa hai thực thể Currency và Wallet là liên kết một – nhiều.



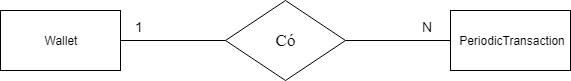
Mỗi khách hàng được phép tạo mới nhiều hoạt động chi tiêu, thu nhập ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và Transaction là liên kết một – nhiều.



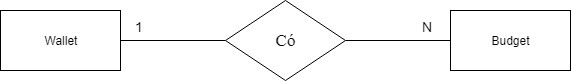
Mỗi khách hàng được phép tạo mới nhiều hoạt động đầu tư ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và Investment là liên kết một – nhiều.



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.



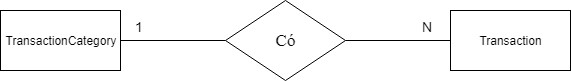
Mỗi khách hàng được phép tạo mới nhiều kế hoạch chi tiêu ứng với từng ví tiền. Do đó, liên kết giữa hai thực thể Wallet và Budget là liên kết một – nhiều.



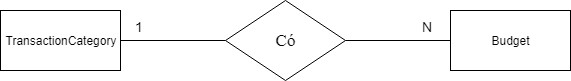
Mỗi khoản đầu tư thuộc một danh mục đầu tư. Do đó, liên kết giữa hai thực thể InvestmentCategory và Investment là liên kết một – nhiều.



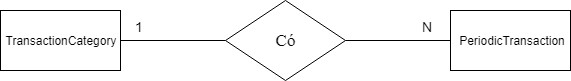
Mỗi hoạt động chi tiêu, thu nhập thuộc một danh mục chi tiêu. Do đó, liên kết giữa hai thực thể TransactionCategory và Transaction là liên kết một – nhiều.



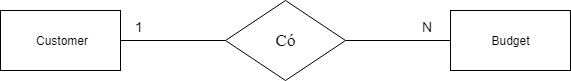
Mỗi kế hoạch chi tiêu thuộc một danh mục chi tiêu. Do đó, liên kết giữa hai thực thể TransactionCategory và Budget là liên kết một – nhiều.



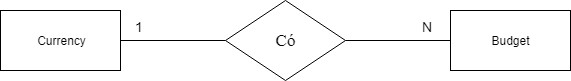
Mỗi chi tiêu dự định thuộc một danh mục chi tiêu. Do đó, liên kết giữa hai thực thể PeriodicTransaction và Transaction là liên kết một – nhiều.



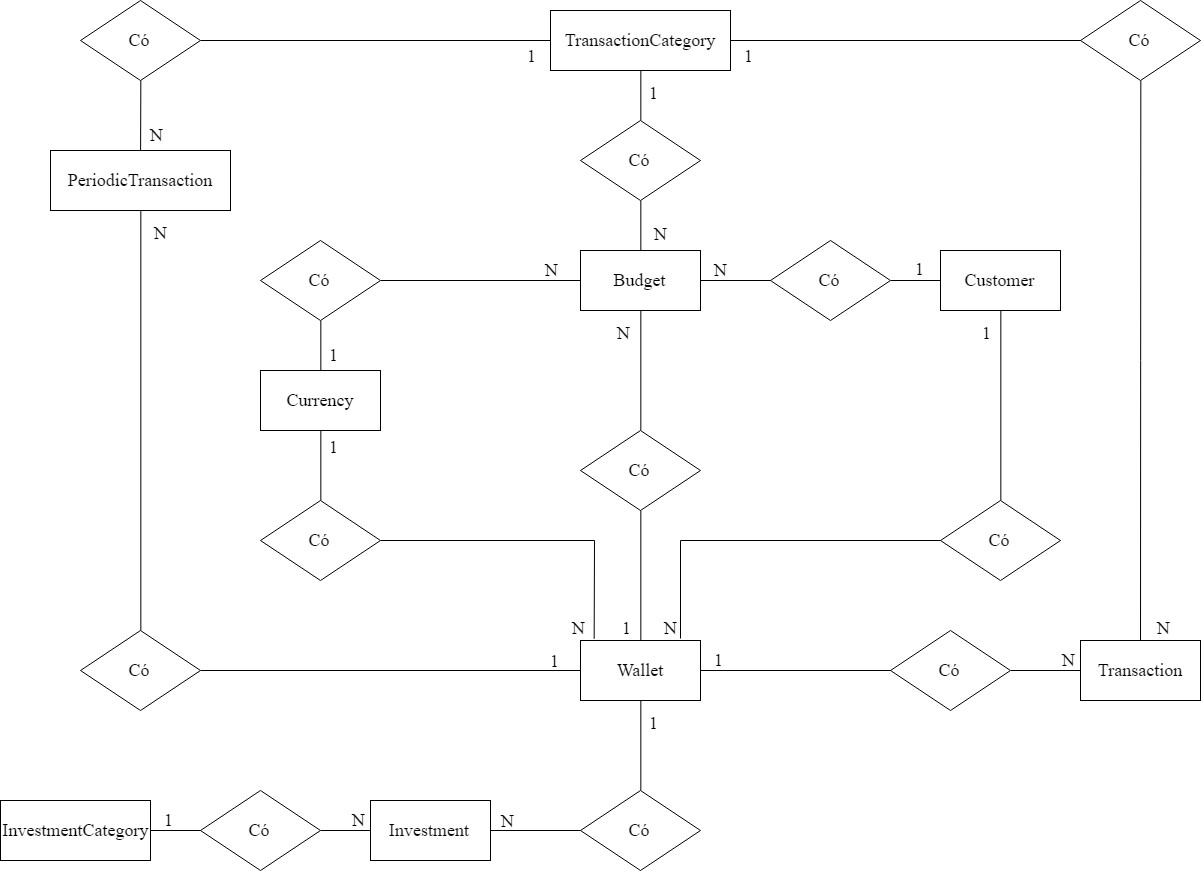
Mỗi kế hoạch chi tiêu khi khách hàng muốn lên kế hoạch cho toàn bộ các ví, kế hoạch chi tiêu của khách hàng sẽ không gắp với một ví tiền cố định mà gắn trực tiếp với khách hàng đó. Do đó, liên kết giữa hai thực thể Customer và Budget là liên kết một – nhiều.



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.

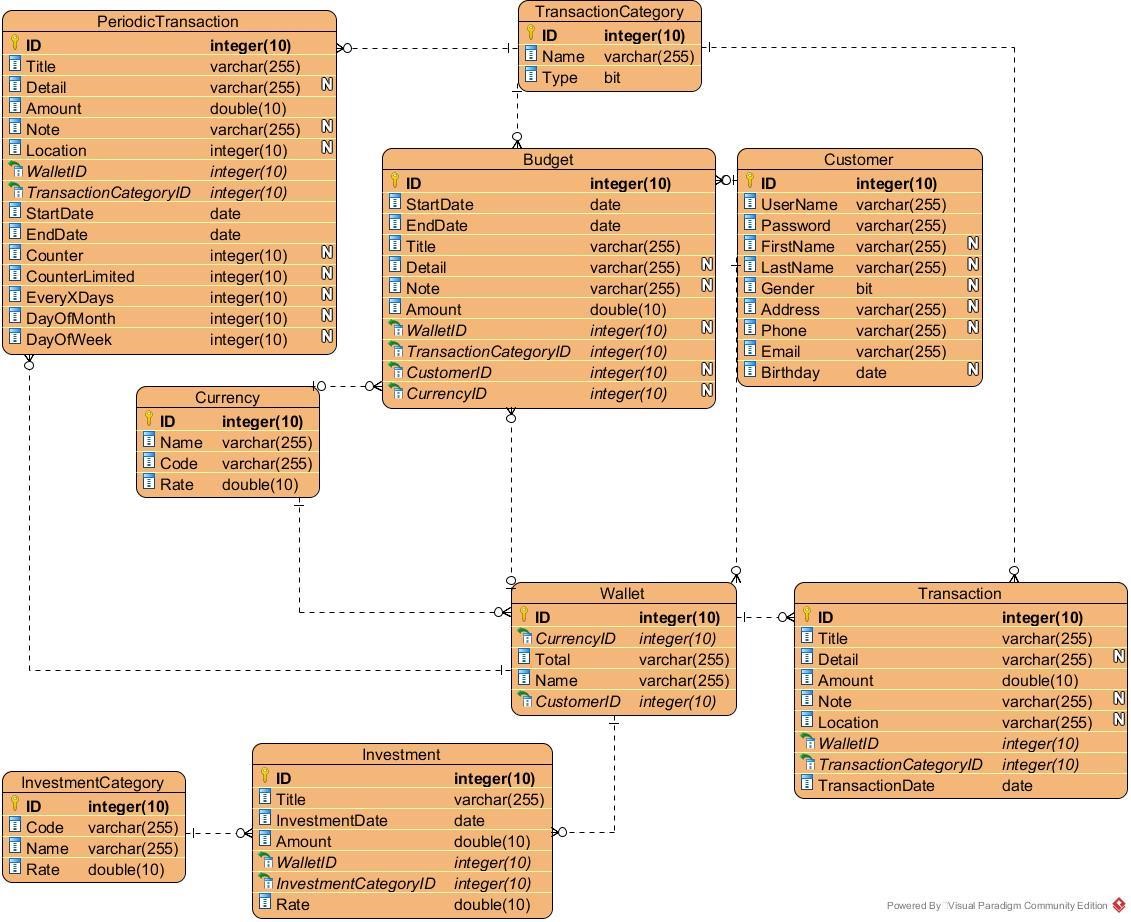


3.3.2.2. Sơ đồ thực thể liên kết



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 ngoài 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

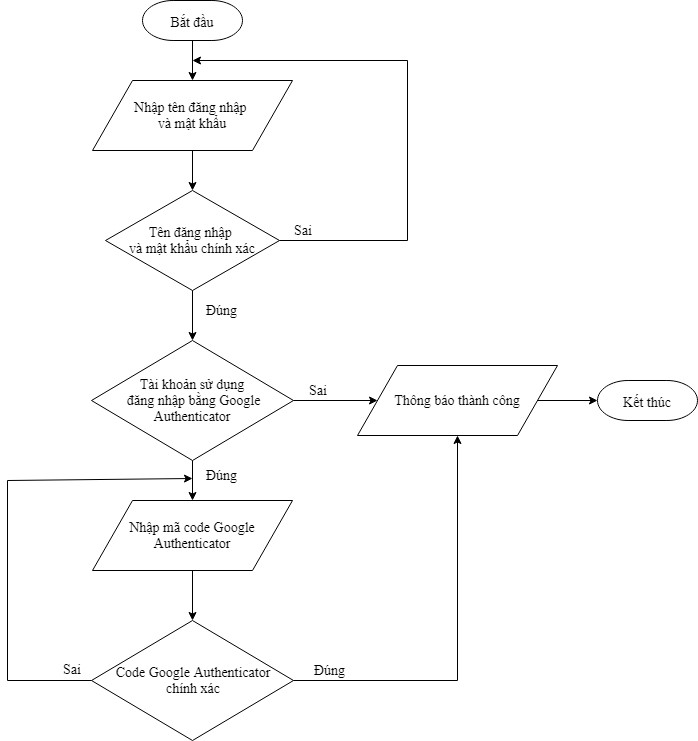
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Tính chất** | **Diễn giải** |
| **ID** | Int | 10 | Primary key | Mã Chi tiêu dự định |
| Title | Varchar | 100 | Not null | Mô tả |
| Detail | Text |  | Null | Chi tiết |
| Amount | Double | 16 | Not 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

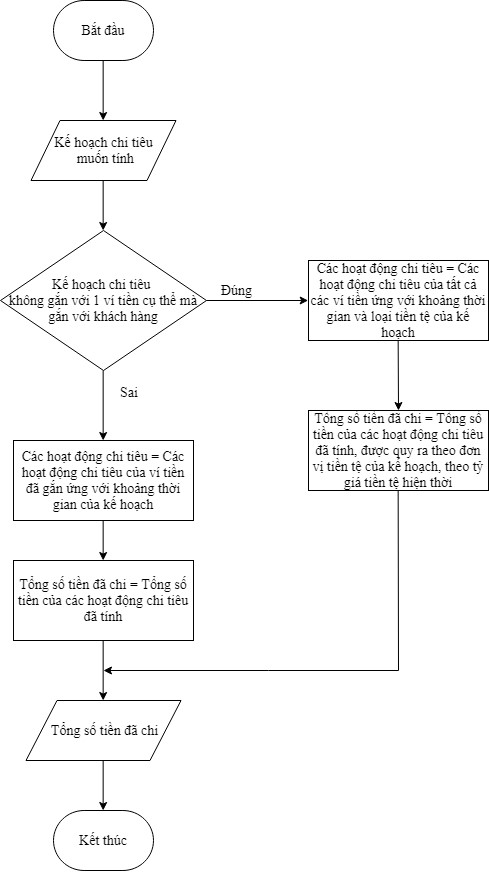
### 3.4.1. Sơ đồ thuật toán đăng nhập



Hình 3.20. Sơ đồ thuật toán đăng nhập

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

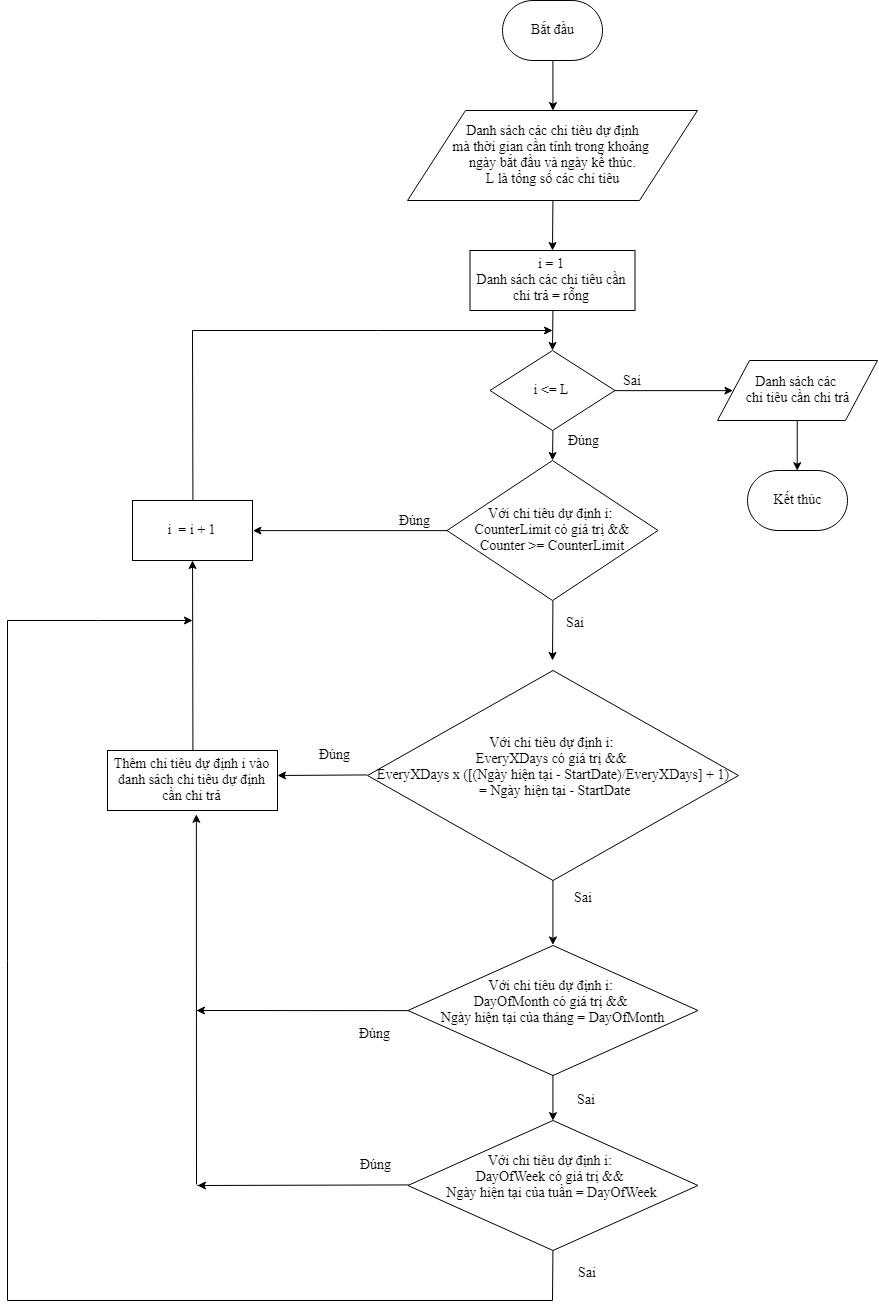
Mô tả: Cần kiểm tra lượng tiến độ thực hiện một kế hoạch chi tiêu cho trước. Đầu vào là kế hoạch muốn kiểm tra, đầu ra là tổng số tiền đã chi thuộc kế hoạch.



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 Chi tiêu dự định

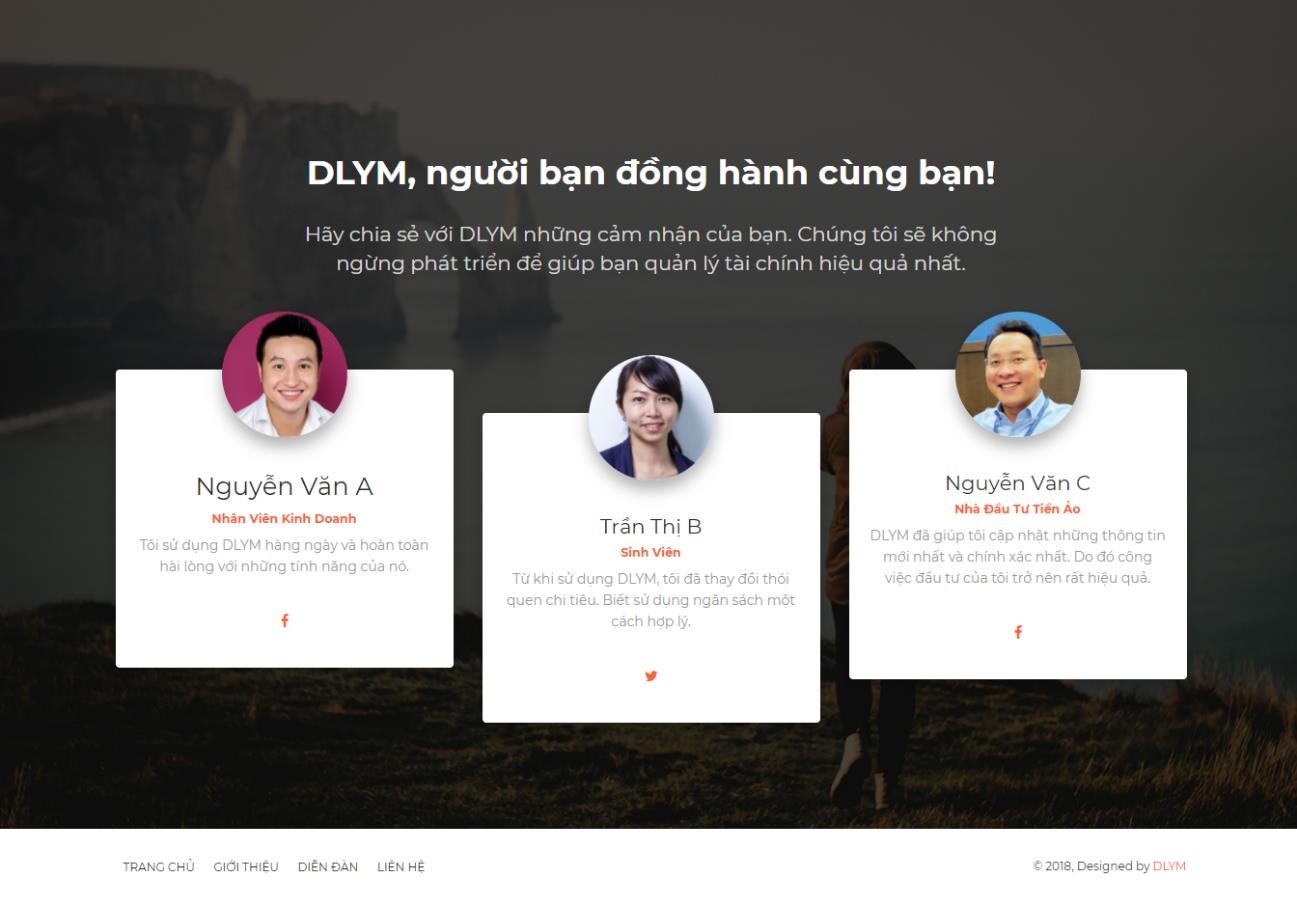
# Chương IV: GIỚI THIỆU MỘT SỐ GIAO DIỆN CHÍNH CỦA WEBSITE

# 4.1. Giới thiệu một số giao diện chính của Website

### 4.1.1. Giao diện trang chủ khách vãng lai







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. Wesite 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 mât dạng MD5, valdition, 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. Khái niệm unity trên Wekipedia:

<https://en.wikipedia.org/wiki/Unity_(game_engine)>\_**Tài liệu Tiếng Anh**

1. Tính năng nổi bật của Unity: <https://cellphones.com.vn/sforum/unity-la-gi#tinh-nang-noi-bat-cua-unity> \_**Tài liệu Tiếng Việt.**
2. T.S Nguyễn Trung Tuấn, ThS Tống Minh Ngọc Cơ sở dữ liệu.
3. Slide bài giảng Phân tích thiết kế hệ thống – Trần Thị Mỹ Diệp – Đại học Kinh tế quốc dân.
4. Bài giảng ASP.NET MVC 5 – Lại Đức Chung - Trường đào tạo CNTT Quốc tế Bachkhoa - Aptech

#### Tài liệu Tiếng Anh

1. Programming ASP.NET MVC 5 A Problem Solution Approach – Nimit

Joshi.

1. Pro ASP.Net MVC 5 – Adam Freeman – Appress.