|  |
| --- |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **---------------------------------------** |
|  |
| **ĐỒ ÁN**  **TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH: CÔNG NGHỆ THÔNG TIN**  **HỆ THỐNG QUẢN LÝ FUND**  **(MINH BẠCH, ĐƠN GIẢN HOÁ GIẤY TỜ LIÊN QUAN ĐẾN QUỸ ĐẦU TƯ CHO CHÍNH PHỦ)** |
| **CBHD:****ThS. Nguyễn Thanh Hải** |
| **Sinh viên: Trịnh Dương Nhật**  **Mã sinh viên: 2020600976** |
|  |
| **Hà Nội – Năm 2024** |

**MỤC LỤC**

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT iii](#_Toc166824603)

[DANH MỤC BẢNG BIỂU iv](#_Toc166824604)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ v](#_Toc166824605)

[MỞ ĐẦU 1](#_Toc166824606)

[1. Lý do chọn đề tài 1](#_Toc166824607)

[2. Mục tiêu của đề tài 1](#_Toc166824608)

[3. Nội dung nghiên cứu 1](#_Toc166824609)

[4. Phạm vi đề tài 2](#_Toc166824610)

[5. Dự kiến kết quả 2](#_Toc166824611)

[6. Bố cục đề tài 2](#_Toc166824612)

[CHƯƠNG 1:TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT 3](#_Toc166824613)

[1.1 Giới thiệu về ngôn ngữ JavaScript 3](#_Toc166824614)

[1.2 Ưu điểm của JavaScript 3](#_Toc166824615)

[1.3 Nodejs Platform 4](#_Toc166824616)

[1.4 Express.js Framework 5](#_Toc166824617)

[1.4.1 Giới thiệu 5](#_Toc166824618)

[1.4.2 Lịch sử hình thành và phát triển 5](#_Toc166824619)

[1.4.3 Ưu điểm của nodejs 6](#_Toc166824620)

[1.5 HTML, CSS 7](#_Toc166824621)

[1.5.1 HTML 7](#_Toc166824622)

[1.5.2 CSS 7](#_Toc166824623)

[1.6 Tổng quan về ReactJs 7](#_Toc166824624)

[1.7 Môi trường phát triển ứng dụng 8](#_Toc166824625)

[1.7.1 PostgreSQL 8](#_Toc166824626)

[1.8 Kết luận 9](#_Toc166824627)

[CHƯƠNG 2:PHÂN TÍCH THIẾT KẾ HỆ THỐNG 10](#_Toc166824628)

[2.1 Khảo sát hệ thống 10](#_Toc166824629)

[2.1.1 Mục đích 10](#_Toc166824630)

[2.1.2 Phương pháp: 10](#_Toc166824631)

[2.1.3 Đối tượng khảo sát 10](#_Toc166824632)

[2.1.4 Kết quả khảo sát sơ bộ 10](#_Toc166824633)

[2.2 Xác định yêu cầu hệ thống 11](#_Toc166824634)

[2.2.1 Các yêu cầu chức năng: 11](#_Toc166824635)

[2.2.2 Các yêu cầu phi chức năng 11](#_Toc166824636)

[2.3 Xây dựng biểu đồ Usecase 11](#_Toc166824637)

[2.3.1 Các tác nhân 11](#_Toc166824638)

[2.3.2 Các usecase của hệ thống 11](#_Toc166824639)

[2.3.3 Biểu đồ usecase: 12](#_Toc166824640)

[2.3.4 Mô tả chi tiết các usecase 13](#_Toc166824641)

[2.4 Xây dựng cơ sở dữ liệu 42](#_Toc166824642)

[2.4.1 Biểu đồ Entity Relationship Diagram 42](#_Toc166824643)

[2.4.2 Các bảng trong cơ sở dữ liệu 43](#_Toc166824644)

[CHƯƠNG 3:KẾT QUẢ, KIỂM THỬ VÀ ĐÁNH GIÁ 46](#_Toc166824645)

[3.1 Kết quả giao diện màn hình 46](#_Toc166824646)

[3.1.1 Giao diện màn hình trang chủ 46](#_Toc166824647)

[3.1.2 Giao diện màn hình tìm kiếm công ty. 46](#_Toc166824648)

[3.1.3 Giao diện màn hình chi tiết những thông tin liên quan đến công ty. 47](#_Toc166824649)

[3.1.4 Giao diện danh sách các quỹ 48](#_Toc166824650)

[3.2 Kế hoạch kiểm thử 48](#_Toc166824651)

[3.3 Chiến lược kiểm thử 49](#_Toc166824652)

[3.4 Mô hình thử nghiệm 49](#_Toc166824653)

[3.5 Kết quả kiểm thử 49](#_Toc166824654)

[KẾT LUẬN 48](#_Toc166824655)

[TÀI LIỆU THAM KHẢO 51](#_Toc166824656)

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CSDL | Cơ sở dữ liệu |
| CSS | Cascading Style Sheets |
| GUI | Graphic User Interface |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| SEO | Search Engine Optimization |
| SGML | Standard Generalized Markup Language |
| ST | System Testing |
| UAT | User Acceptance Testing |

# DANH MỤC BẢNG BIỂU

[Bảng 2.1 Bảng các Usecase của hệ thống 11](#_Toc166824657)

[Bảng 2.2 Bảng Accounts (Người dùng) 43](#_Toc166824658)

[Bảng 2.3 Bảng GPCompanies (Bài viết) 43](#_Toc166824659)

[Bảng 2.4 Bảng Funds (Qũy) 43](#_Toc166824660)

[Bảng 2.5 Bảng DocGPs (Tài liệu cho công ty) 44](#_Toc166824661)

[Bảng 2.6 Bảng DocFunds(Tài liệu cho quỹ) 44](#_Toc166824662)

[Bảng 2.7 Bảng GPCompanyAccounts (quyền tài khoản với công ty) 44](#_Toc166824663)

[Bảng 2.8 Bảng FundAccount (Thông tin đầu tư của tài khoản đối với quỹ) 45](#_Toc166824664)

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 1.1 Top các ngôn ngữ phổ biến năm 2022 - Theo trang Codelearn tổng hợp 4](#_Toc167458163)

[Hình 2.1 Biểu đồ Usecase của hệ thống 12](#_Toc167458164)

[Hình 2.2 Biểu đồ trình tự Usecase Đăng ký 13](#_Toc167458165)

[Hình 2.3 Biểu đồ lớp chức năng Usecase Đăng ký 15](#_Toc167458166)

[Hình 2.4 Biểu đồ trình tự Usecase Đăng nhập 16](#_Toc167458167)

[Hình 2.5 Biểu đồ lớp chức năng Usecase Đăng nhập 18](#_Toc167458168)

[Hình 2.6 Biểu đồ trình tự Usecase Quên mật khẩu 19](#_Toc167458169)

[Hình 2.7 Biểu đồ các lớp Usecase Quên mật khẩu 21](#_Toc167458170)

[Hình 2.8 Biểu đồ trình tự Usecase Đổi mật khẩu 22](#_Toc167458171)

[Hình 2.9 Biểu đồ các lớp Usecase Đổi mật khẩu 24](#_Toc167458172)

[Hình 2.10 Biểu đồ trình tự Usecase Cập nhật thông tin cá nhân 25](#_Toc167458173)

[Hình 2.11 Biểu đồ các lớp Usecase Cập nhật thông tin cá nhân 27](#_Toc167458174)

[Hình 2.12 Biểu đồ trình tự Usecase tạo công ty 28](#_Toc167458175)

[Hình 2.13 Biểu đồ các lớp Usecase Tạo công ty 30](#_Toc167458176)

[Hình 2.14 Biểu đồ trình tự Usecase chính sửa thông tin công ty 31](#_Toc167458177)

[Hình 2.15 Biểu đồ các lớp Usecase chỉnh sửa thông tin công ty 32](#_Toc167458178)

[Hình 2.16 Biểu đồ trình tự Usecase thêm tài liệu cho công ty 33](#_Toc167458179)

[Hình 2.17 Biểu đồ các lớp Usecase thêm tài liệu cho công ty 34](#_Toc167458180)

[Hình 2.18 Biểu đồ trình tự Usecase tạo quỹ 35](#_Toc167458181)

[Hình 2.19 Biểu đồ các lớp Usecase tạo quỹ 37](#_Toc167458182)

[Hình 2.20 Biểu đồ trình tự Usecase chỉnh sửa quỹ 38](#_Toc167458183)

[Hình 2.21 Biểu đồ các lớp Usecase chỉnh sửa bài viết 39](#_Toc167458184)

[Hình 2.22 Biểu đồ trình tự Usecase thêm tài liệu cho quỹ 40](#_Toc167458185)

[Hình 2.23 Biểu đồ các lớp Usecase thêm tài liệu cho quỹ 41](#_Toc167458186)

[Hình 2.30 Biểu đồ Entity Relationship Diagram 42](#_Toc167458187)

[Hình 3.1 Giao diện màn hình trang chủ 46](#_Toc167458188)

[Hình 3.2 Giao diện màn hình tìm kiếm công ty. 47](#_Toc167458189)

[Hình 3.3 Giao diện thông tin chi tiết về công ty. 47](#_Toc167458190)

[Hình 3.4 Giao diện danh sách quỹ trực thuộc 48](#_Toc167458191)

# MỞ ĐẦU

## 1. Lý do chọn đề tài

Trong thế giới ngày càng phát triển và cạnh tranh, việc quản lý tài chính hiệu quả trở nên vô cùng quan trọng. Đặc biệt, lĩnh vực quản lý quỹ đang thu hút sự chú ý ngày càng lớn từ các nhà đầu tư, doanh nghiệp, và tổ chức tài chính. Đề tài "Hệ thống quản lý Fund" được lựa chọn vì đây là một lĩnh vực đầy thách thức và tiềm năng, đòi hỏi sự sáng tạo và áp dụng công nghệ hiện đại để tối ưu hóa quy trình quản lý.

Thực trạng ở Việt Nam chưa có nhiều nền tảng quản lý đáp ứng được nhu cầu những người quản lý quỹ. Nhiều người mới gặp khó khăn về quản lý về giấy tờ và biến động quỹ quá rời rạc. Việc Xây dựng một hệ thống quản lý quỹ sẽ bắt nhịp xu thế, đáp ứng được nhu cầu quản lý giấy tờ của các nhà quản lý quỹ.

Ngoài ra, việc xây dựng một hệ thống quản lý quỹ mang lại nhiều lợi ích kinh tế và xã hội. Hệ thống sẽ cải thiện chất lượng cuộc sống và đóng góp vào sự phát triển kinh tế và xã hội của Việt Nam.

Bởi những lý do trên, em đã tiến hành nghiên cứu về đề tài: ***“Hệ Thống Quản Lý Fund"*** với mong muốn xây dựng được một hệ thống đáp ứng nhu cầu đầu tư của các nhà quản lý quỹ, tổ chức tài chính nhằm giải quyết minh bạch các lịch sử giao dịch cũng như các thông tin về quỹ.

Để hoàn thành đề tài này em xin được gửi lời cảm ơn sâu sắc đến quý thầy cô trường Đại Học Công Nghiệp Hà Nội, đặc biệt là thầy ThS Nguyễn Thanh Hải đã tận tình chỉ bảo và hỗ trợ em trong quá trình thực hiện đề tài.

## 2. Mục tiêu của đề tài

* Xây dựng được Website hệ thống quản lý sử dụng ngôn ngữ Javascript.
* Rèn luyện các kiến thức và kĩ năng phát triển hệ thống trên nền tảng Javascript và PostgresQL. Tích lũy thêm nhiều kinh nghiệm cho bản thân và thái độ làm việc chuyên nghiệp.

## 3. Nội dung nghiên cứu

* Xây dựng Website hệ thống quản lý Fund:
  + *Tạo và quản lý tài khoản người dùng:* Cho phép người dùng đăng ký, đăng nhập, quản lý thông tin cá nhân và bảo mật tài khoản, dữ liệu cá nhân.
  + *Tạo và chỉnh sửa công ty:* Hệ thống cho Admin tạo công ty và cung cấp giấy tờ liên quan đến công ty.
  + *Tạo và quản lý hệ thống fund:* Cho phép tài khoản đã có công ty tạo ra quỹ và cung cấp các giao dịch thay đổi thông tin tài chính quỹ.
* Các yêu cầu khác:
* Xây dựng bản thiết kế theo đúng quy chuẩn đầy đủ nội dung.
* Có cấu trúc dữ liệu chuẩn hóa, phù hợp thực tiễn.
* Giao diện chuyên nghiệp, dễ sử dụng, phù hợp với người dùng.
* Có kế hoạch và thực hiện đánh giá kiểm thử hệ thống.

## 4. Phạm vi đề tài

* Đề tài chỉ dừng lại ở người dùng có thể tạo quỹ, cung cấp các giấy tờ liên quan, điều chỉnh tài chính của hệ thống quỹ.

## 5. Dự kiến kết quả

* Hoàn thành hệ thống quản lý fund với các chức năng đã đề ra.
* Thực hiện kiểm thử hệ thống quản lý fund.

## 6. Bố cục đề tài

* Ngoại trừ các phần Mở đầu, Kết luận và Tài liệu tham khảo báo cáo được bố cục thành 3 chương:
* *Chương 1:* Tổng quan về cơ sở lý thuyết
* *Chương 2:* Phân tích thiết kế hệ thống
* *Chương 3:* Kết quả, kiểm thử và đánh giá

# TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

## Giới thiệu về ngôn ngữ JavaScript

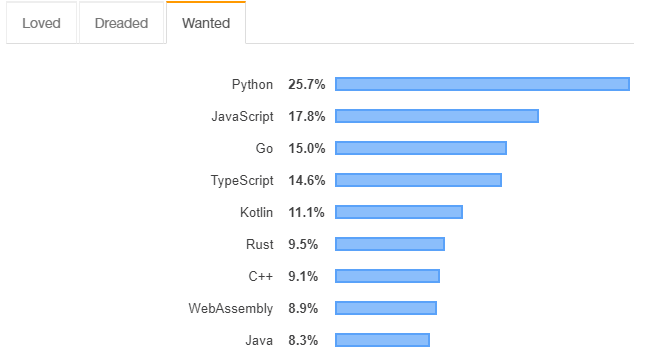
* JavaScript, hay còn được gọi là JS, là một ngôn ngữ lập trình phía máy chủ và phía trình duyệt được phát triển bởi Netscape Communications Corporation vào năm 1995. JavaScript được thiết kế để cung cấp khả năng tương tác động và tạo ra nội dung web động, đáp ứng nhu cầu của các trang web hiện đại. Ngôn ngữ này được thiết kế với cú pháp đơn giản, dễ đọc và dễ hiểu, điều này giúp cho các nhà phát triển có thể tập trung vào việc giải quyết vấn đề thay vì phải mất thời gian để hiểu cú pháp.
* Một trong những đặc điểm nổi bật của JavaScript là khả năng xử lý sự kiện và tạo ra giao diện người dùng tương tác, cho phép các trang web trở nên sống động và thân thiện với người dùng. Điều này giúp cho ứng dụng web được tối ưu hóa để đáp ứng các yêu cầu về trải nghiệm người dùng và tương tác nhanh chóng. JavaScript cũng cung cấp một số tính năng tiên tiến như ES6 (ECMAScript 2015) với các tính năng như arrow functions, template literals, và modules, giúp làm cho quá trình lập trình trở nên dễ dàng hơn và linh hoạt hơn.

## Ưu điểm của JavaScript

JavaScript, hay còn được gọi là JS, là một ngôn ngữ lập trình phía máy chủ và phía trình duyệt được phát triển từ năm 1995, đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất hiện nay. JavaScript có nhiều ưu điểm đáng chú ý, bao gồm:

* Hiệu suất cao: JavaScript đã được tối ưu hóa để cung cấp hiệu suất nhanh chóng, giúp ứng dụng web chạy mượt mà và phản hồi nhanh chóng.
* Đa luồng và song song: JavaScript hỗ trợ xử lý đa luồng thông qua Web Workers, cho phép chạy các tác vụ nặng nhọc trong nền mà không làm gián đoạn giao diện người dùng.
* Tính đơn giản và dễ đọc: JavaScript được thiết kế với nguyên tắc "Giữ nó đơn giản", giúp cho việc học và sử dụng ngôn ngữ trở nên dễ dàng hơn.
* Quản lý bộ nhớ hiệu quả: JavaScript sử dụng bộ thu gom rác tự động (garbage collector) để giải phóng bộ nhớ tự động, giảm bớt thời gian bảo trì bộ nhớ của người lập trình.
* Hỗ trợ đầy đủ cho đa nền tảng: JavaScript có thể chạy trên bất kỳ trình duyệt web nào, không cần cài đặt thêm phần mềm, làm cho việc phát triển ứng dụng trở nên dễ dàng và thuận tiện hơn.
* Quản lý lỗi tốt: JavaScript cung cấp các công cụ tiện ích để quản lý lỗi, giúp người lập trình dễ dàng xác định, sửa chữa và phát hiện lỗi trong mã nguồn.
* Khả năng tích hợp và mở rộng: JavaScript hỗ trợ tích hợp với các thư viện và công cụ bên thứ ba, như jQuery, React, Angular, và Vue.js, mở rộng khả năng và phạm vi ứng dụng.

Với tất cả các ưu điểm và tính năng của mình, JavaScript đang được sử dụng rộng rãi trong các công ty công nghệ hàng đầu trên thế giới như Google, Facebook, Amazon, Netflix, và nhiều công ty khác, giúp tạo ra các ứng dụng web và ứng dụng di động phức tạp và tương tác.



Hình 1.1 Top các ngôn ngữ phổ biến năm 2022 - Theo trang Codelearn tổng hợp

## Nodejs Platform

* Node.js là một nền tảng lập trình đa năng, cho phép các nhà phát triển xây dựng các ứng dụng web và ứng dụng di động hiệu suất cao. Node.js nổi bật với hiệu suất cao, khả năng xử lý đa luồng mạnh mẽ và tính bảo mật cao. Để giúp các nhà phát triển xây dựng ứng dụng Node.js nhanh chóng và hiệu quả hơn, đã xuất hiện nhiều framework Node.js phổ biến và được ưa chuộng.
* Một trong những framework Node.js phổ biến nhất hiện nay là Express.js. Express.js là một framework nhẹ và nhanh, được sử dụng rộng rãi để xây dựng các ứng dụng web hiệu suất cao. Nó cung cấp một loạt các tính năng hữu ích như routing, middleware, và hỗ trợ cho các phương thức HTTP. Ngoài ra, Express.js còn được tích hợp với các thư viện khác như Mongoose để xử lý cơ sở dữ liệu.
* Một framework Node.js khác cũng rất phổ biến là Koa.js. Koa.js là một framework được thiết kế để giúp các nhà phát triển xây dựng ứng dụng web hiệu suất cao với một kiến trúc đơn giản và dễ hiểu. Koa.js cung cấp một API nhẹ nhàng và linh hoạt, cho phép các nhà phát triển tùy chỉnh và mở rộng ứng dụng của họ một cách dễ dàng. Nó cũng hỗ trợ middleware, giúp quản lý các tác vụ như xử lý request, response, và xử lý lỗi một cách hiệu quả.
* Cả Express.js và Koa.js đều là những framework Node.js được ưa chuộng nhờ vào tính năng mạnh mẽ, hiệu suất cao và khả năng tùy chỉnh cao. Chúng giúp giảm thiểu thời gian phát triển và tăng cường khả năng mở rộng của ứng dụng, đồng thời cung cấp một môi trường phát triển mạnh mẽ và linh hoạt cho các nhà phát triển.

## Express.js Framework

### Giới thiệu

* Express.js là một trong những framework phổ biến nhất của Node.js, được sử dụng rộng rãi để xây dựng các ứng dụng web hiệu suất cao. Express.js được thiết kế để đơn giản hóa việc xử lý HTTP request/response, cung cấp một loạt các tính năng hữu ích giúp các nhà phát triển xây dựng ứng dụng web một cách dễ dàng hơn.
* Với Express.js, việc định tuyến (routing) trở nên rất dễ dàng, cho phép bạn định tuyến theo cú pháp đơn giản và dễ hiểu. Express.js cung cấp middleware để giúp các nhà phát triển xử lý các yêu cầu và trả về phản hồi một cách dễ dàng. Middleware có thể được sử dụng để xác thực người dùng, kiểm soát quyền truy cập, hoặc thực hiện các tác vụ khác như xử lý lỗi và logging.
* Express.js được đánh giá cao về hiệu suất, nhờ vào kiến trúc đơn giản và cấu trúc dữ liệu nhỏ gọn. Điều này làm cho nó trở thành một lựa chọn tuyệt vời cho các ứng dụng web đòi hỏi tốc độ xử lý cao. Ngoài ra, Express.js còn hỗ trợ nhiều tính năng hữu ích khác như xử lý file, bộ đệm, xác thực, bảo mật, và nhiều hơn nữa. Tất cả các tính năng này giúp các nhà phát triển xây dựng các ứng dụng web hiệu quả và tiết kiệm thời gian.
* Express.js không chỉ đơn giản là một framework, mà còn là một nền tảng mạnh mẽ cho việc phát triển ứng dụng web và ứng dụng di động. Với sự hỗ trợ của một cộng đồng phát triển lớn và phong phú, Express.js tiếp tục mở rộng và cải tiến, giúp các nhà phát triển tiếp cận và tận dụng những công nghệ mới nhất trong lĩnh vực phát triển web.

### Lịch sử hình thành và phát triển

* Express.js, một trong những framework phổ biến nhất của Node.js, cũng được phát triển bởi một nhóm lập trình viên tại Viện Công nghệ Georgia (Georgia Tech) vào năm 2010. Ban đầu, Express.js được tạo ra như một dự án để khám phá các phương pháp xử lý HTTP Request/Response trên Node.js.
* Express.js được thiết kế với mục đích đơn giản hóa việc xử lý HTTP Request/Response và cung cấp các tính năng hữu ích giúp các nhà phát triển xây dựng ứng dụng web dễ dàng hơn. Với cú pháp đơn giản và dễ hiểu, Express.js cho phép các nhà phát triển xây dựng các ứng dụng web hiệu quả và tiết kiệm thời gian.
* Hiện nay, Express.js được sử dụng rộng rãi bởi các công ty công nghệ lớn như Netflix, LinkedIn, Uber, và nhiều hơn nữa. Express.js cũng được đánh giá cao bởi cộng đồng lập trình viên Node.js vì tính đơn giản, dễ sử dụng, và hiệu suất tốt. Nó đã trở thành một nền tảng quan trọng trong phát triển ứng dụng web và ứng dụng di động, hỗ trợ một loạt các tính năng như middleware, định tuyến, và xử lý file, giúp tăng cường khả năng mở rộng và hiệu suất của các ứng dụng.

### Ưu điểm của nodejs

* Tốc độ và Hiệu suất: Node.js nổi tiếng với tốc độ xử lý nhanh chóng và hiệu suất cao, cho phép xử lý hàng nghìn kết nối đồng thời mà không gặp vấn đề về hiệu suất. Node.js sử dụng mô hình I/O không chặn (non-blocking I/O), cho phép nó xử lý nhiều yêu cầu cùng một lúc mà không cần chờ đợi kết thúc của mỗi yêu cầu trước khi chuyển sang yêu cầu tiếp theo. Điều này giúp cải thiện hiệu suất và tốc độ xử lý của các ứng dụng web và ứng dụng di động.
* Cú pháp đơn giản: Node.js sử dụng JavaScript, một ngôn ngữ lập trình phổ biến và dễ học, giúp cho việc phát triển ứng dụng trở nên dễ dàng hơn. Với cú pháp đơn giản và dễ hiểu, Node.js là một lựa chọn tuyệt vời cho cả những lập trình viên mới học lẫn những người muốn nhanh chóng xây dựng các ứng dụng web.
* Hỗ trợ Middleware: Node.js hỗ trợ việc sử dụng middleware, cho phép mở rộng và tùy chỉnh ứng dụng web một cách linh hoạt. Middleware là các hàm có thể thực hiện các tác vụ như xử lý request, response, và thực hiện các tác vụ trung gian khác trước khi request đến đến hàm xử lý chính.
* Tính linh hoạt: Node.js cho phép sử dụng các thư viện và framework bên ngoài, giúp tăng cường khả năng phát triển và mở rộng ứng dụng. Điều này tạo ra một môi trường phát triển linh hoạt, cho phép các lập trình viên chọn lựa và tích hợp các công cụ phù hợp với nhu cầu của dự án.
* Tài liệu phong phú: Node.js có một cộng đồng phát triển lớn và tài liệu phong phú, bao gồm hướng dẫn, ví dụ, và tài liệu chính thức. Tài liệu này giúp các lập trình viên dễ dàng tìm hiểu và sử dụng Node.js, từ những người mới học đến những chuyên gia muốn tìm hiểu sâu hơn về các tính năng và kỹ thuật phát triển ứng dụng.

## HTML, CSS

### HTML

* HTML là chữ viết tắt cho “HyperText Markup Language”, hay còn được gọi là "Ngôn ngữ Đánh dấu Siêu văn bản". Nó là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì.

### CSS

* CSS là chữ viết tắt cho “Cascading Style Sheets”, được sử dụng để mô tả giao diện và định dạng của một tài liệu viết bằng ngôn ngữ đánh dấu. Nó cung cấp một tính năng bổ sung cho HTML. Nó thường được sử dụng với HTML để thay đổi style của trang web và giao diện người dùng. Nó cũng có thể được sử dụng với bất kỳ loại tài liệu XML nào bao gồm cả XML đơn giản, SVG và XUL.

## Tổng quan về ReactJs

* ReactJS là một thư viện mã nguồn mở phổ biến được sử dụng để xây dựng giao diện người dùng trên web. Nó được phát triển bởi Facebook và được giới thiệu vào năm 2013. ReactJS được thiết kế để nhanh chóng, hiệu quả và có khả năng mở rộng, giúp cho việc xây dựng các ứng dụng web quy mô lớn trở nên dễ dàng hơn.
* Sau đây là một số tính năng và khái niệm quan trọng của ReactJS:
  + *Kiến trúc dựa trên Component:* ReactJS có kiến trúc dựa trên Component, trong đó các giao diện người dùng phức tạp được chia thành các component nhỏ, có thể sử dụng lại. Điều này giúp cho việc phát triển và bảo trì các ứng dụng quy mô lớn trở nên dễ dàng hơn.
  + *Virtual DOM:* ReactJS sử dụng Virtual DOM, đó là một bản sao trong bộ nhớ của DOM. Điều này cho phép ReactJS cập nhật giao diện người dùng một cách hiệu quả mà không cần phải thao tác trực tiếp trên DOM.
  + *JSX:* JSX là một phần mở rộng cú pháp cho JavaScript, cho phép các nhà phát triển viết mã HTML-like bên trong các tệp JavaScript của họ. JSX được sử dụng để định nghĩa cấu trúc và giao diện của các component trong ReactJS.
  + *Quản lý trạng thái:* ReactJS sử dụng khái niệm "trạng thái" để quản lý dữ liệu và hành vi của các component. Trạng thái là một đối tượng chứa thông tin về trạng thái hiện tại của component và có thể được cập nhật theo thời gian.
  + *Luồng dữ liệu một chiều:* ReactJS sử dụng luồng dữ liệu một chiều, điều này có nghĩa là dữ liệu chảy từ component cha xuống các component con. Điều này giúp cho việc quản lý luồng dữ liệu trong ứng dụng trở nên dễ dàng hơn và duy trì trạng thái ổn định.
  + *Redux:* Redux là một thư viện phổ biến được sử dụng để quản lý trạng thái trong các ứng dụng ReactJS. Nó cung cấp một kho chứa trung tâm để quản lý trạng thái ứng dụng, giúp cho việc quản lý trạng thái của các component trong ứng dụng trở nên dễ dàng hơn và giúp duy trì trạng thái ổn định.
  + *React Router:* React Router là một thư viện cung cấp cho nhà phát triển các công cụ để quản lý định tuyến trong ứng dụng ReactJS. Nó cho phép nhà phát triển định tuyến các URL của trang và hiển thị các component phù hợp với từng URL đó.
  + *Cộng đồng phát triển mạnh mẽ:* ReactJS có một cộng đồng phát triển mạnh mẽ và rộng lớn, với hàng ngàn thư viện và công cụ hỗ trợ được phát triển bởi các nhà phát triển khác nhau. Điều này giúp cho việc phát triển và bảo trì các ứng dụng ReactJS trở nên dễ dàng hơn.

## Môi trường phát triển ứng dụng

### PostgreSQL

* PostgreSQL, còn được biết đến với tên mã nguồn là Postgres, là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở mạnh mẽ và linh hoạt, được phát triển bởi một nhóm các lập trình viên và tổ chức không chính thức. PostgreSQL được sử dụng rộng rãi trong các ứng dụng web và là một phần quan trọng của công nghệ LAMP (Linux, Apache, MySQL, PHP/Python/Perl), cũng như trong nhiều ứng dụng doanh nghiệp và dịch vụ đám mây.
* PostgreSQL được thiết kế để xử lý các cơ sở dữ liệu lớn và có thể đáp ứng các yêu cầu về tốc độ truy xuất và xử lý dữ liệu phức tạp. PostgreSQL hỗ trợ các tính năng quản trị cơ sở dữ liệu như tạo, sửa đổi, xóa và truy xuất dữ liệu, cùng với các tính năng bảo mật mạnh mẽ và khả năng sao lưu dữ liệu. PostgreSQL nổi bật với khả năng hỗ trợ các kiểu dữ liệu phong phú, tính năng ACID (Atomicity, Consistency, Isolation, Durability) mạnh mẽ, và khả năng mở rộng tốt.
* PostgreSQL có nhiều phiên bản và cung cấp nhiều tùy chọn để tùy chỉnh và cấu hình hệ thống theo nhu cầu sử dụng của người dùng. PostgreSQL cũng hỗ trợ các ngôn ngữ lập trình phổ biến như PHP, Java, Python, và C++, cho phép người dùng tạo ra các ứng dụng tùy chỉnh và tích hợp với các ứng dụng web.
* So với MySQL, PostgreSQL nổi bật với việc hỗ trợ các tính năng phức tạp hơn như các kiểu dữ liệu phong phú, hỗ trợ cho các câu lệnh SQL chuẩn SQL, và khả năng mở rộng tốt hơn. PostgreSQL cũng được biết đến với tính năng ACID mạnh mẽ, điều này giúp đảm bảo rằng các giao dịch dữ liệu luôn được hoàn thành một cách chính xác và an toàn.
* Một số ưu điểm của PostgreSQL bao gồm:
  + *Hiệu suất cao:* Được thiết kế để xử lý các cơ sở dữ liệu lớn và có khả năng xử lý hàng nghìn truy vấn mỗi giây.
  + *Độ tin cậy cao:* Được sử dụng rộng rãi và được đánh giá cao về độ tin cậy và sẵn sàng của hệ thống.
  + *Tính linh hoạt:* Có nhiều phiên bản và tùy chọn cấu hình để phù hợp với các nhu cầu sử dụng khác nhau.
  + *Bảo mật:* Cung cấp các tính năng bảo mật để bảo vệ dữ liệu khỏi các mối đe dọa bên ngoài.
  + *Tính mở rộng:* Có khả năng mở rộng để đáp ứng các nhu cầu mở rộng của hệ thống và dữ liệu.

## Kết luận

* Trong quá trình lựa chọn công nghệ để xây dựng đề tài **Hệ thống quản lý Fund**, em đã quan tâm đến sự đơn giản, hiệu suất và khả năng mở rộng của các công nghệ. Do đó, em đã quyết định sử dụng Express.js để phát triển ứng dụng back-end. Express.js có tính năng đơn giản, hiệu suất cao và khả năng xử lý các yêu cầu đa nhiệm, phù hợp với một ứng dụng mạng xã hội có khả năng mở rộng trong tương lai.
* Đối với cơ sở dữ liệu, em đã chọn PostgreSQL vì nó là một hệ quản trị cơ sở dữ liệu phổ biến và được sử dụng rộng rãi trong các ứng dụng web. PostgreSQL có tính năng đáp ứng các yêu cầu về tốc độ truy xuất và xử lý dữ liệu, độ tin cậy cao và tính linh hoạt trong cấu hình.
* Tổng thể, sự kết hợp của Express.js, MySQL và Reactjs đã mang lại một giải pháp hiệu quả cho việc xây dựng đề tài ***Hệ thống quản lý Fund***, với độ tin cậy và khả năng phát triển tốt hơn nữa trong tương lai.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Khảo sát hệ thống

### Mục đích

# Mục đích của việc khảo sát người dùng cho đề tài Hệ thống quản lý Fund là để đánh giá và phân tích các yếu tố ảnh hưởng đến hiệu suất, tính tin cậy và khả năng mở rộng của hệ thống.

# Bằng cách này, chúng ta có thể xác định được các vấn đề và cải tiến để cải thiện hiệu suất và khả năng mở rộng của hệ thống. Ngoài ra, việc khảo sát hệ thống cũng giúp đưa ra các quyết định phù hợp về lựa chọn công nghệ, thiết kế kiến trúc và cách triển khai hệ thống để đáp ứng các yêu cầu của người dùng.

# Việc khảo sát hệ thống cũng giúp chúng ta đảm bảo tính bảo mật của hệ thống và đưa ra các giải pháp phù hợp, xây dựng ứng dụng thân thiện, gần gũi, phù hợp với người dùng. Vì vậy, việc khảo sát hệ thống là một bước quan trọng trong quá trình phát triển và xây dựng hệ thống quản lý quỹ.

### Phương pháp:

* Phỏng vấn trực tiếp

### Đối tượng khảo sát

* Là người có nhiều năm làm việc trong quản lý quỹ.

### Kết quả khảo sát sơ bộ

* Mục tiêu: Mục tiêu của việc khảo sát trong đề tài xây dựng **Hệ thống quản lý Fund** là để thu thập thông tin, ý kiến và mong muốn của người dùng. Cụ thể, mục tiêu của việc khảo sát bao gồm:
  + Hiểu rõ nhu cầu và mong muốn của người dùng đối với hệ thống quản lý cơ bản, bao gồm các tính năng, chức năng, và trải nghiệm người dùng mà họ mong đợi.
  + Biết được mức độ hài lòng của người dùng đối với hệ thống quản lý Fund, từ đó đánh giá được điểm mạnh, điểm yếu của hệ thống và cải thiện những khía cạnh chưa đạt yêu cầu.
  + Xác định các tính năng hoặc chức năng mà người dùng muốn được thêm vào hệ thống quản lý Fund để nâng cao trải nghiệm người dùng.
  + Tìm hiểu về thói quen sử dụng mạng xã hội của người dùng, từ đó có thể điều chỉnh và tối ưu hoá giao diện, trải nghiệm người dùng.

## Xác định yêu cầu hệ thống

### Các yêu cầu chức năng:

* *Đăng ký và đăng nhập:* Cho phép người dùng đăng ký tài khoản mới và đăng nhập vào hệ thống quản lý Fund.
* *Tạo và quản lý tài khoản người dùng:* Cho phép người dùng đăng ký, đăng nhập, quản lý thông tin cá nhân và bảo mật tài khoản, dữ liệu cá nhân.
* *Tạo và chỉnh sửa công ty:* Hệ thống cho Admin tạo công ty và cung cấp giấy tờ liên quan đến công ty, gán cho người quản lý công ty.
* *Tạo và quản lý hệ thống fund:* Cho phép tài khoản đã có công ty tạo ra quỹ và cung cấp các giao dịch thay đổi thông tin tài chính quỹ.
* *Cung cấp giấy tờ cho quỹ*: Cho phép người dùng cup cấp giấy tờ liên quan đến quỹ mà tài khoản đang quản lý.

### Các yêu cầu phi chức năng

* Có hiệu năng cao, thời gian phản hồi nhanh.
* Giao diện người dùng đẹp, thân thiện và dễ sử dụng.
* Có khả năng đáp ứng nhiều người truy cập.
* Dễ sử dụng đối với người dùng.
* Khả năng bảo trì, quản lý tốt.
* Tuân thủ quy định của pháp luật.

## Xây dựng biểu đồ Usecase

### Các tác nhân

* Người dùng: Là những cá nhân sử dụng và tương tác, bao gồm việc tạo tài khoản, đăng nhập, đăng ký, chỉnh sửa thông tin cá nhân, tạo quỹ, xem thông tin quỹ và cung cấp giấy tờ liên quan đến quỹ và công ty.

### Các usecase của hệ thống

Bảng 2.1 Bảng các Usecase của hệ thống

|  |  |
| --- | --- |
| **Tác nhân** | **Các ca sử dụng** |
| Admin, Người quản lý | * Đăng ký, Đăng Nhập, quên mật khẩu * Cập nhật tài khoản * Tạo vả chỉnh sửa công ty * Chỉnh sửa thông tin công ty * Xem chi tiết tài liệu |
| Admin | * Tạo công ty |
| Người quản lý | * Tạo quỹ * Chỉnh sửa và thêm giấy tờ cho quỹ |

### Biểu đồ usecase:

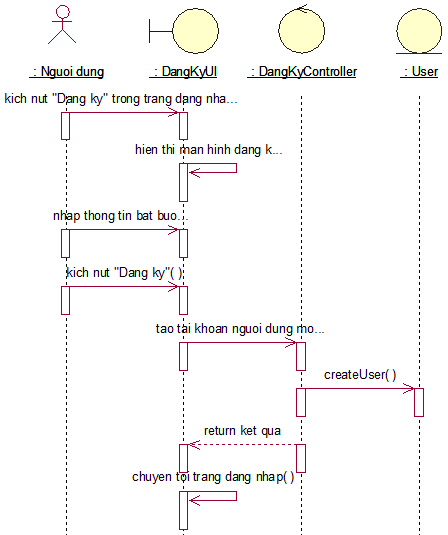


Hình 2.1 Biểu đồ Usecase của hệ thống

### Mô tả chi tiết các usecase

#### Usecase Đăng ký

##### Biểu đồ trình tự

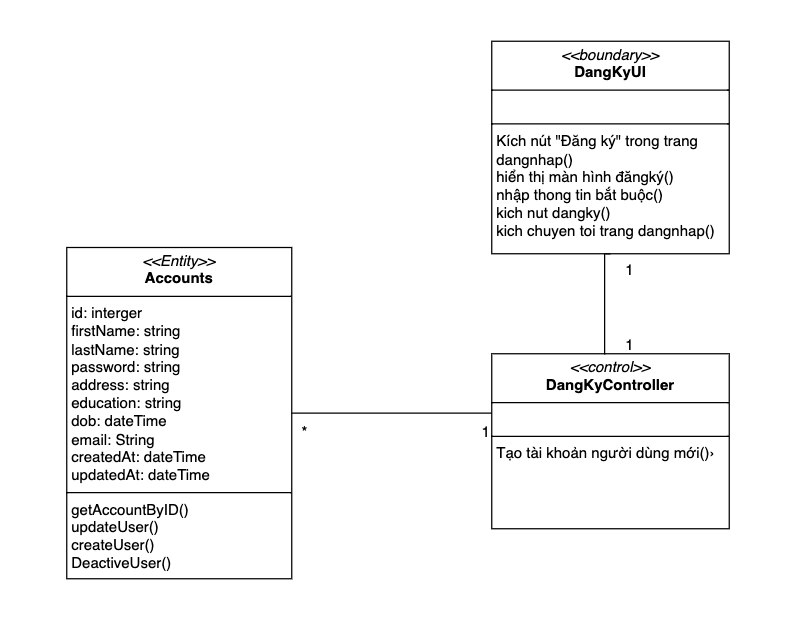


Hình 2.2 Biểu đồ trình tự Usecase Đăng ký

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng tạo tài khoản mới.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    - Usecase này bắt đầu khi người dùng kích vào nút “Đăng ký” trên thanh trong trang đăng nhập. Hệ thống sẽ yêu cầu nhập các thông tin (Họ, tên, số điện thoại, email, học vấn, mật khẩu, ngày sinh) để tạo tài khoản.
    - sau khi gửi thông tin lên hệ thông,hệ thống sẽ chuyển đến trang chủ và thông báo thành công.Hệ thống sẽ mã hoá mật khẩu và lưu thông tin của người dùng vào bảng ACCOUNTS và chuyển tới màn đăng nhập. Usecase kết thúc.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và usecase kết thúc.
    - Tại bước 2 trong luồng cơ bản khi người dùng nhập thông tin không hợp lệ. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục thao tác.
* **Các yêu cầu đặc biệt:** Người dùng cần nhập đúng các thông tin yêu cầu để tạo tài khoản.
* **Tiền điều kiện**: Không có.
* **Hậu điều kiện:** Đăng ký thành công sẽ cập nhật một bản ghi mới trong bảng Accounts.
* **Điểm mở rộng:** Không có.

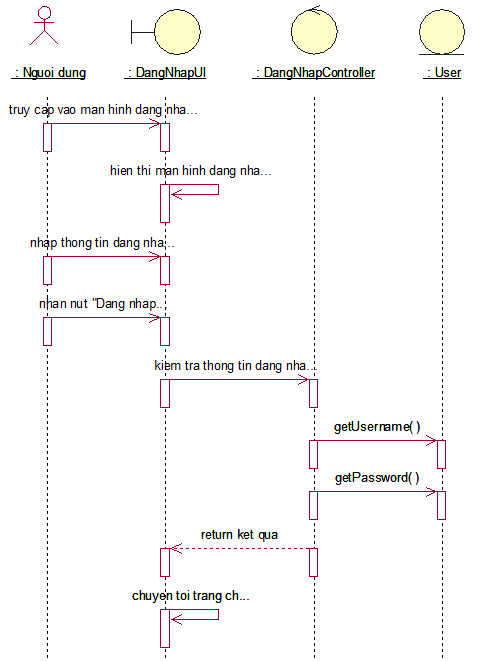
##### Biểu đồ lớp



Hình 2.3 Biểu đồ lớp chức năng Usecase Đăng ký

#### Usecase Đăng nhập

##### Biểu đồ trình tự

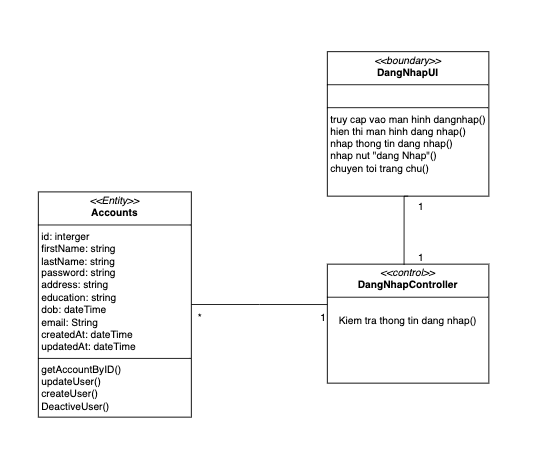


Hình 2.4 Biểu đồ trình tự Usecase Đăng nhập

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng đăng nhập vào hệ thống.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    - Usecase này bắt đầu khi người dùng ấn vào nút “Đăng nhập” trên thanh menu. Hệ thống sẽ yêu cầu người dùng nhập tên đăng nhập và mật khẩu.
    - Người dùng nhập đầy đủ thông tin yêu cầu và kích vào nút “Đăng nhập”. Hệ thống sẽ kiểm tra thông tin nhập so với thông tin trong bảng ACCOUNTS nếu có tài khoản trong bảng ACCOUNTS sẽ đăng nhập và chuyển hướng tới trang chủ.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và usecase kết thúc.
    - Tại bước 2 trong luồng cơ bản khi người dùng nhập thiếu thông tin, thông tin không hợp lệ hoặc nhập sai thông tin không tìm thấy bản ghi trong bảng ACCOUNTS. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục thao tác hoặc kết thúc thao tác.
* **Các yêu cầu đặc biệt:** Người dùng cần nhập đúng các thông tin yêu cầu để đăng nhập.
* **Tiến điều kiện:** Người dùng đã có tài khoản.
* **Hậu điều kiện:** Đăng nhập thành công người dùng được thực hiện các chức năng khác.
* **Điểm mở rộng:** Không có.

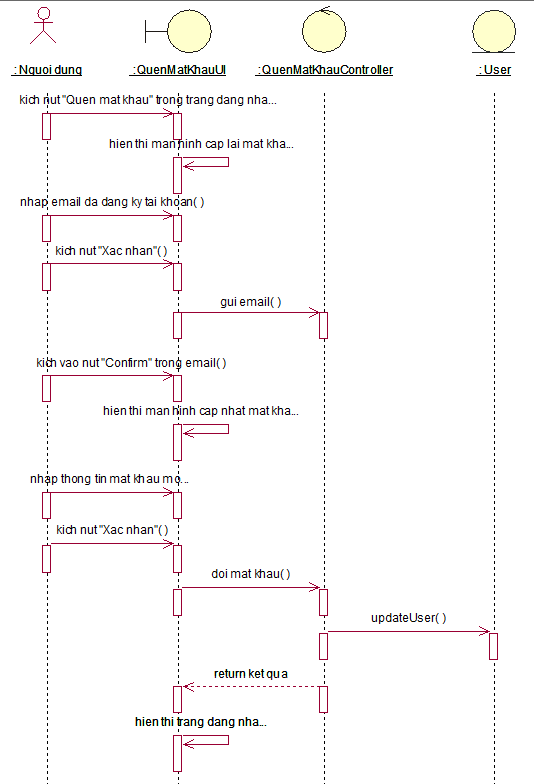
##### Biểu đồ lớp



Hình 2.5 Biểu đồ lớp chức năng Usecase Đăng nhập

#### Usecase Quên mật khẩu

##### Biểu đồ trình tự

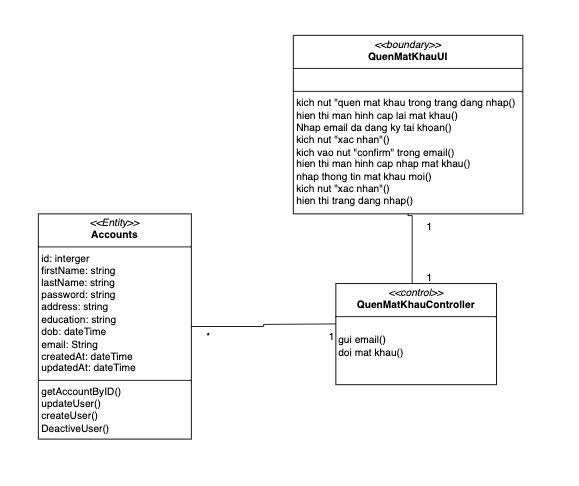


Hình 2.6 Biểu đồ trình tự Usecase Quên mật khẩu

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng lấy lại mật khẩu tài khoản.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    - Usecase này bắt đầu khi người dùng kích nút “Quên mật khẩu” trong màn hình đăng nhập. Hệ thống sẽ chuyển sang trang để người dùng nhập email đăng ký tài khoản.
    - Sau khi nhập email và kích nút “Xác nhận”. Hệ thống sẽ gửi link để đặt lại mật khẩu đến email này.
    - Sau kích link được gửi đến mail, hệ thống sẽ chuyển đến trang đặt lại mật khẩu. Sau khi điền mật khẩu mới, kích nút “Xác nhận” hệ thống sẽ tiến hành cập nhật mật khẩu mới vào bảng ACCOUNTS.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Không có.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

##### Biểu đồ lớp



Hình 2.7 Biểu đồ các lớp Usecase Quên mật khẩu

#### Usecase Đổi mật khẩu:

##### Biểu đồ trình tự

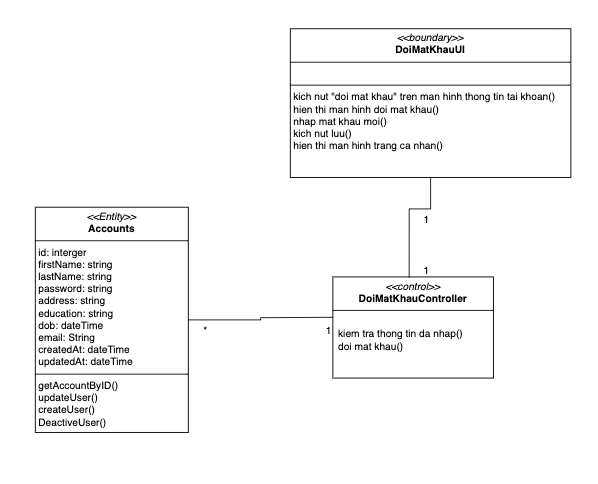


Hình 2.8 Biểu đồ trình tự Usecase Đổi mật khẩu

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng đổi mật khẩu tài khoản đăng nhập.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    - Usecase này bắt đầu khi người dùng vào trang cập nhật mật khẩu và kích vào nút “Cập nhật”. Hệ thống sẽ hiển thị yêu cầu nhập mật khẩu cũ và mật khẩu mới.
    - Người dùng kích vào nút “Lưu”. Hệ thống sẽ gửi link để xác nhận đã đổi mật khẩu đến email này. Sau khi click vào link, hệ thống sẽ so sánh mật khẩu cũ (nhập trong form) và mật khẩu đã lưu trong CSDL, nếu trùng nhau thì lưu mật khẩu mới vào trong CSDL.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
    - Tại bước 2 trong luồng cơ bản khi người dùng nhập sai mật khẩu cũ. Hệ thống sẽ hiển thị thống báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục thao tác hoặc kết thúc thao tác.
* **Các yêu cầu đặc biệt:** Người dùng cần nhập đúng mật khẩu cũ để đổi mật khẩu thành công.
* **Tiền điều kiện:** Người dùng cần đăng nhập.
* **Hậu điều kiện:** Cập nhật thành công mật khẩu của một bản ghi trong bảng ACCOUNTS.
* **Điểm mở rộng:** Không có.

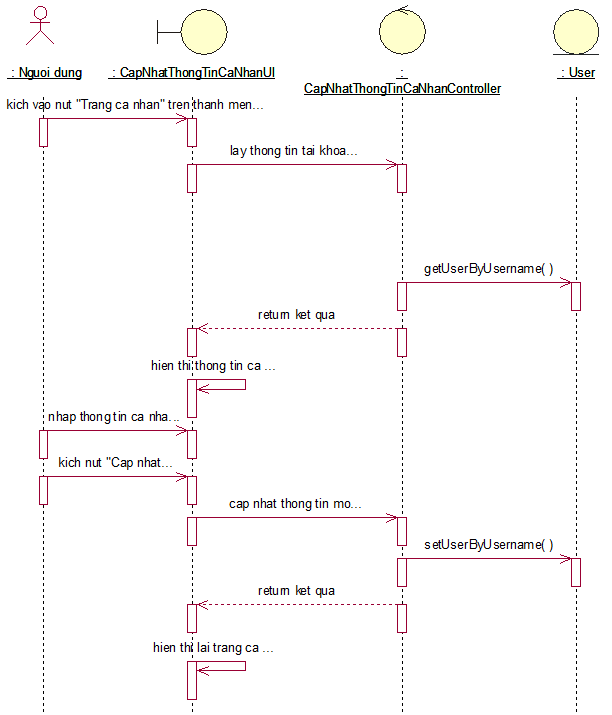
##### Biểu đồ lớp



Hình 2.9 Biểu đồ các lớp Usecase Đổi mật khẩu

#### Usecase Cập nhật thông tin

##### Biểu đồ trình tự

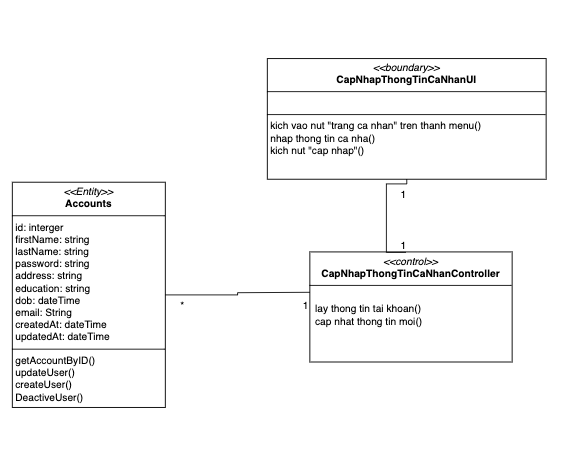


Hình 2.10 Biểu đồ trình tự Usecase Cập nhật thông tin cá nhân

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng cập nhập thông tin của tài khoản.
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu khi người dùng di chuột qua tên đăng nhập trên thanh navbar và kích vào nút “Tài khoản của tôi” trong menu. Hệ thống sẽ hiển thị thông tin cũ để tiến hành chỉnh sửa.
    - Người dùng kích vào nút “Cập nhật”. Hệ thống sẽ cập nhật thông tin của người dùng trong bảng **ACCOUNTS** và hiện thông báo cập nhật thành công.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
    - Tại bước 2 trong luồng cơ bản khi người dùng nhập thông tin không hợp lệ. Hệ thống sẽ hiển thị thống báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục thao tác hoặc kết thúc thao tác.
* **Các yêu cầu đặc biệt:** Không.
* **Tiền điều kiện:** Người dùng cần đăng nhập.
* **Hậu điều kiện:** Cập nhật thành công thông tin một bản ghi trong bảng **ACCOUNTS**.
* **Điểm mở rộng:** Không có.

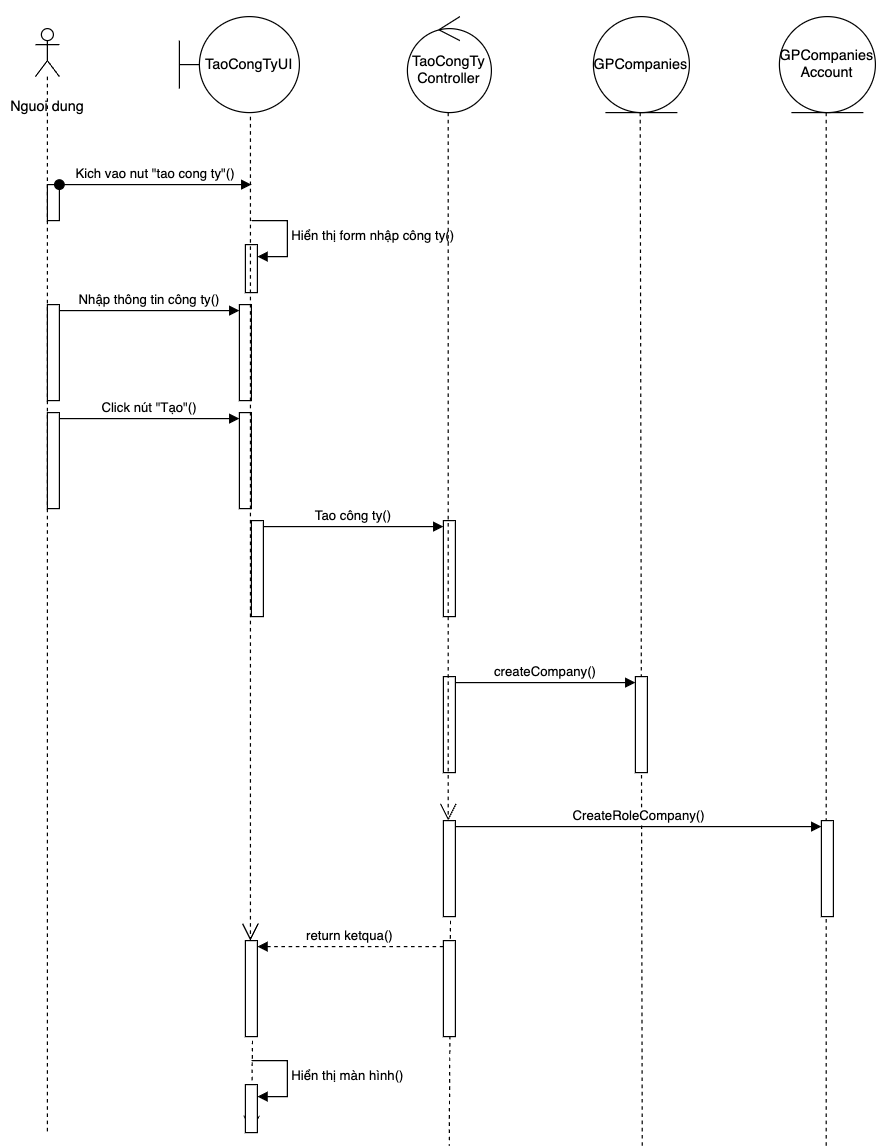
##### Biểu đồ lớp



Hình 2.11 Biểu đồ các lớp Usecase Cập nhật thông tin cá nhân

#### Usecase tạo công ty

##### Biểu đồ trình tự

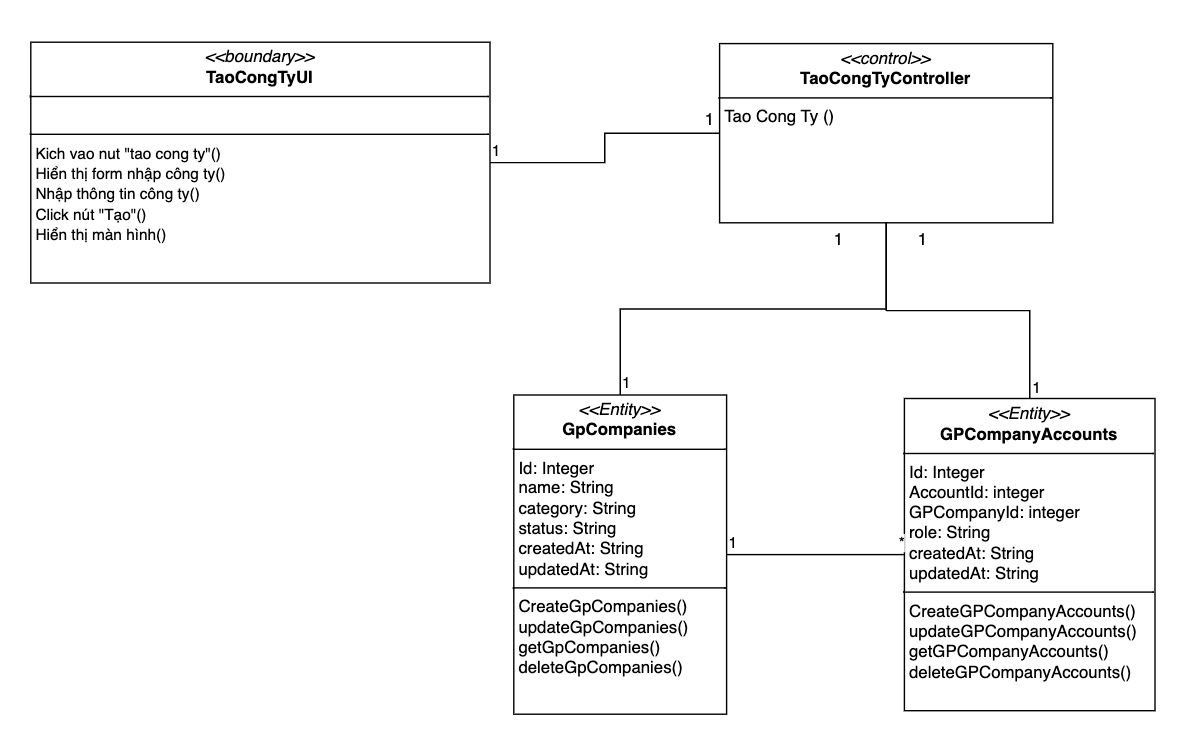


Hình 2.12 Biểu đồ trình tự Usecase tạo công ty

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng tạo công ty.
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu khi người dùng nhấn vào nút “Tạo công ty” trong màn danh sách công ty. Hệ thống sẽ hiển thị cửa sổ tạo công ty mới.
    - Người dùng cần nhập các thông tin.
    - Người dùng kích vào nút “Tạo”. Hệ thống sẽ cập nhật thông tin của bài viết trong bảng GPCompanies và GPCompaniesAccounts.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
    - Tại bước 2 trong luồng cơ bản khi người dùng nhập thông tin không hợp lệ. Hệ thống sẽ hiển thị thống báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục thao tác hoặc kết thúc thao tác.
* **Các yêu cầu đặc biệt:** Không.
* **Tiền điều kiện:** Người dùng cần đăng nhập với tài khoản admin
* **Hậu điều kiện:** Cập nhật thành công thông tin một bản ghi trong các bảng GPCompanies, GPCompaniesAccounts.
* **Điểm mở rộng:** Không có.

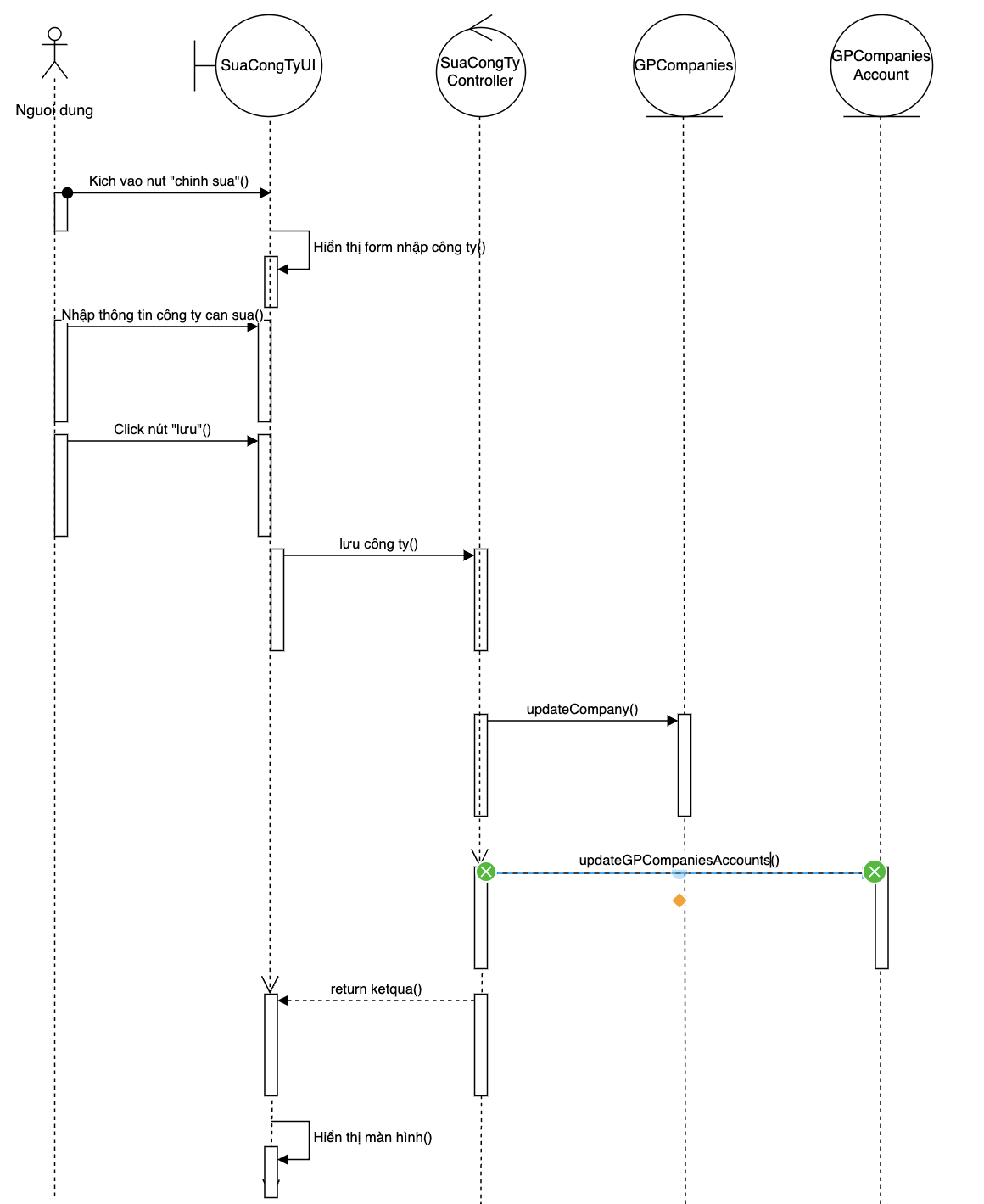
##### Biểu đồ lớp



Hình 2.13 Biểu đồ các lớp Usecase Tạo công ty

#### Usecase Chỉnh sửa thông tin công ty

##### Biểu đồ trình tự

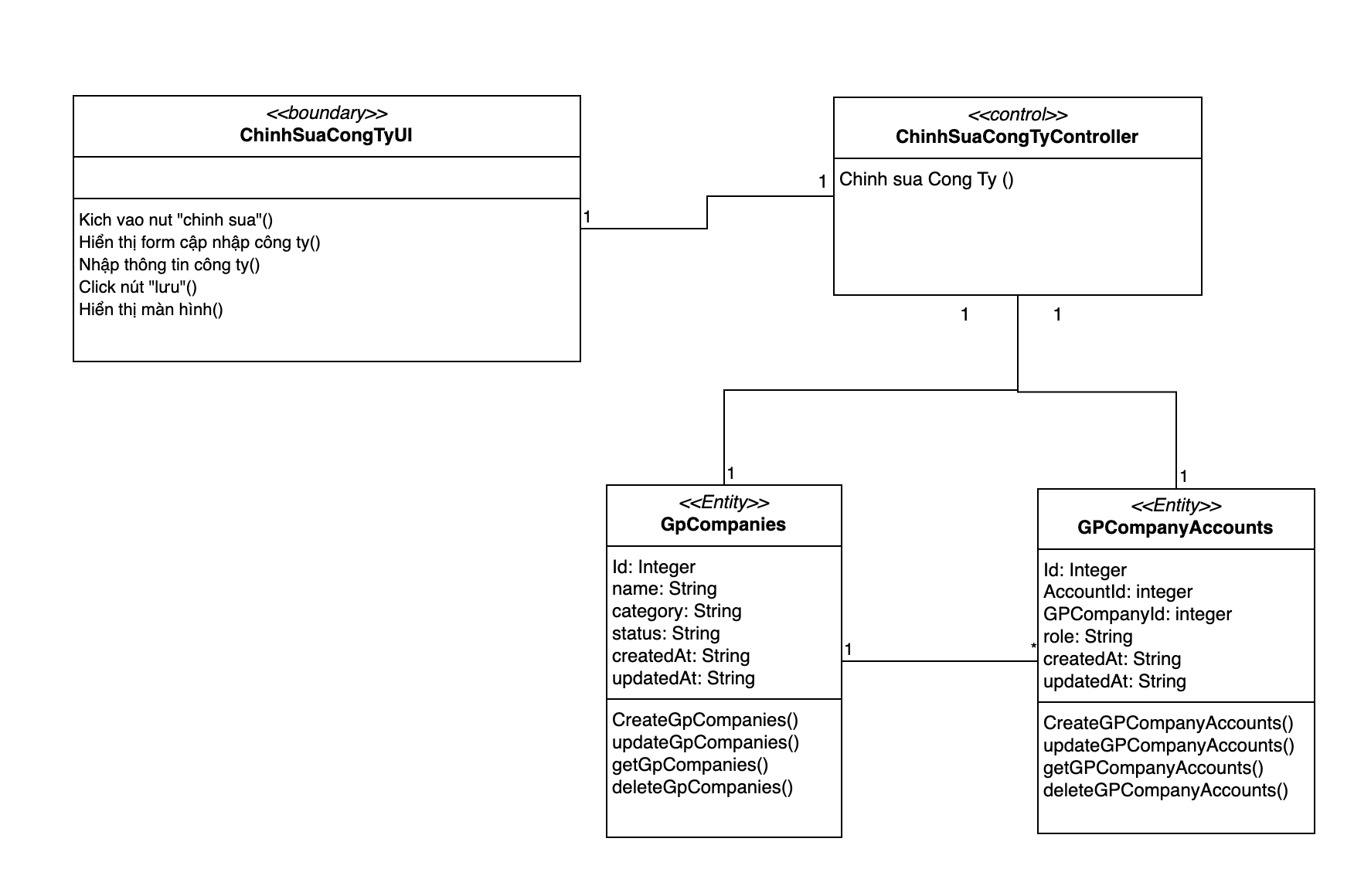


Hình 2.14 Biểu đồ trình tự Usecase chính sửa thông tin công ty

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng chỉnh sửa thông tin công ty
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu khi người dùng kích vào nút “chỉnh sửa” ở cuối dòng bảng hiển thị danh sách công ty, sau đó nhập thông tin cần chỉ sửa vào cửa sổ đang hiển thị rồi ấn “lưu"..
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Người dùng cần đăng nhập.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

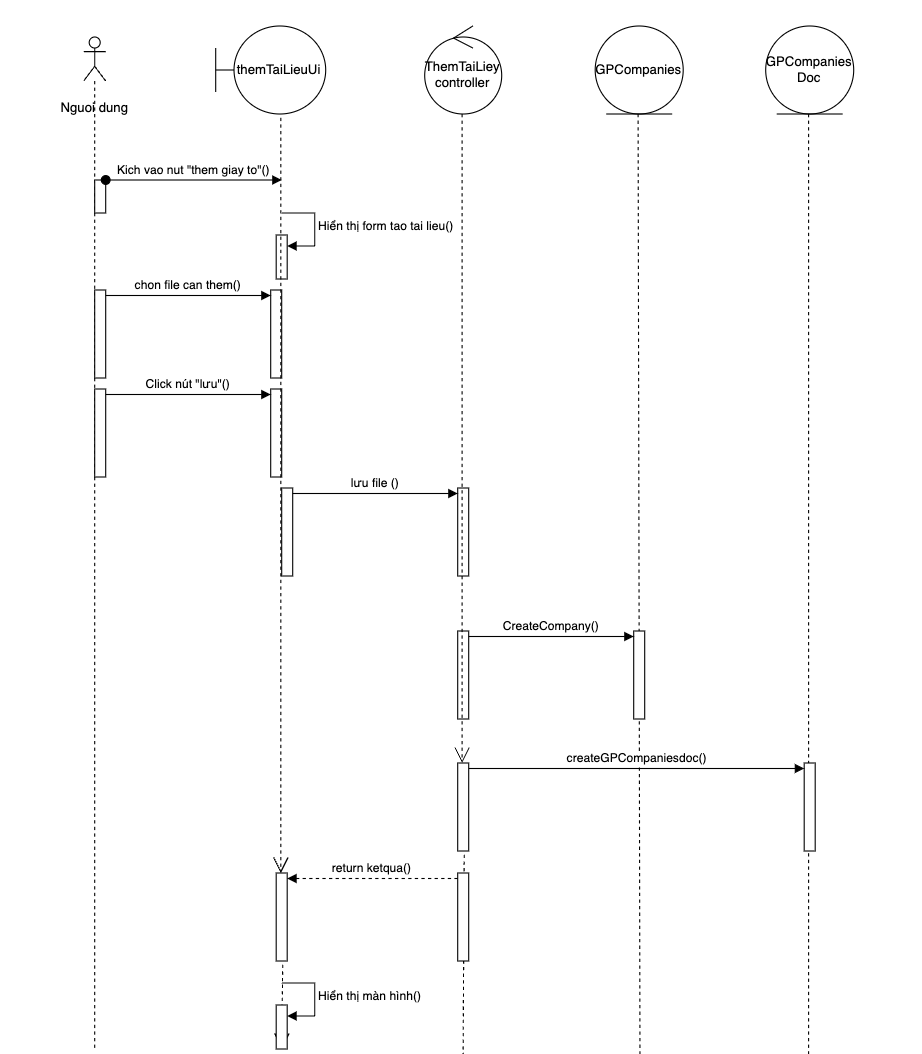
##### Biểu đồ lớp



Hình 2.15 Biểu đồ các lớp Usecase chỉnh sửa thông tin công ty

#### Usecase Thêm tài liệu cho công ty

##### Biểu đồ trình tự

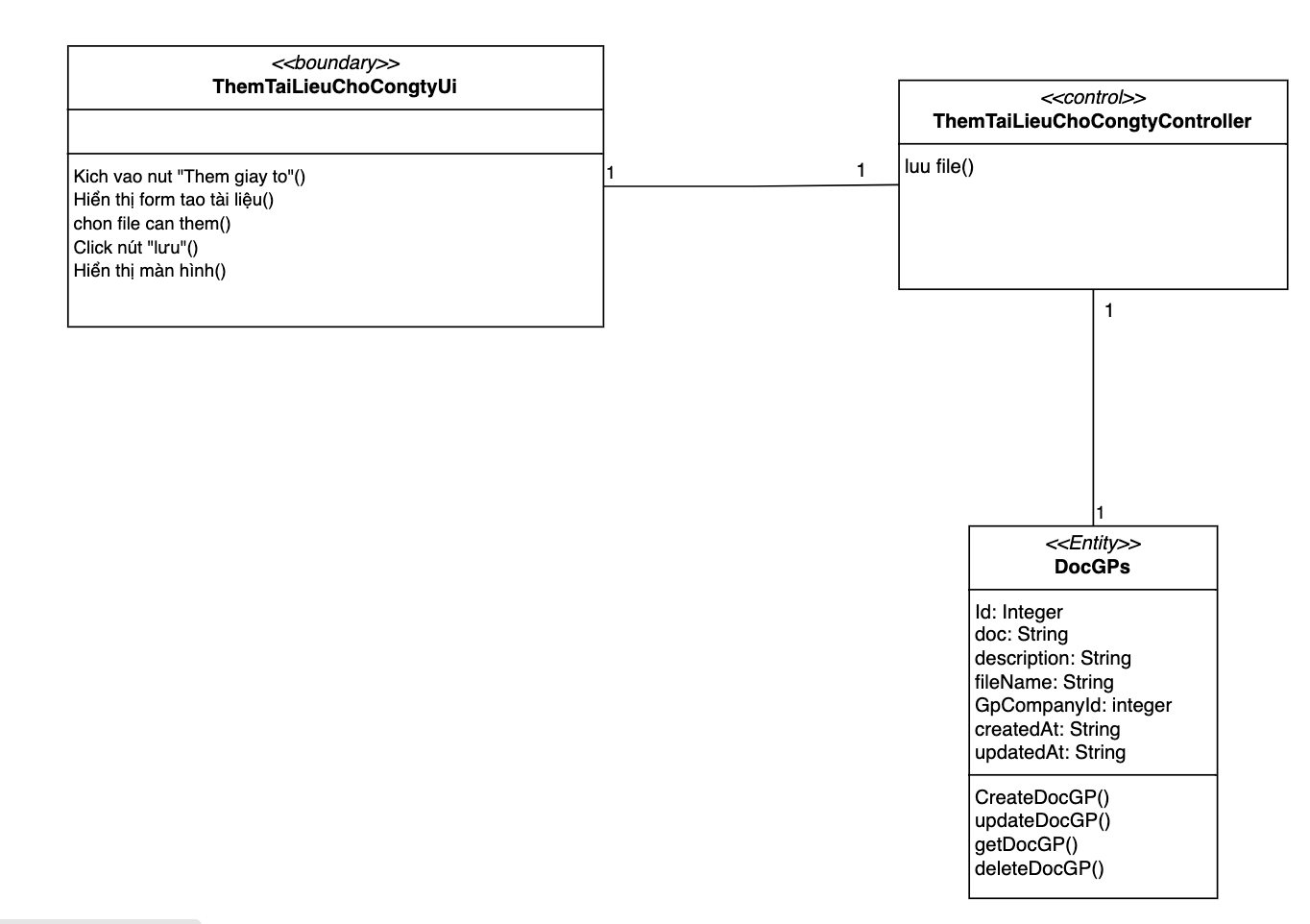


Hình 2.16 Biểu đồ trình tự Usecase thêm tài liệu cho công ty

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng thêm tài liệu cho công ty.
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu khi người dùng kích vào nút “Cung cấp tài liệu trong menu” trong menu của bài viết. Sau đó, người dùng cập nhật lại nội dung bài viết và ấn vào nút “Lưu”. Hệ thống sẽ cập nhật thông tin của bài viết sau đó lưu vào bảng POSTS và hiển thị lại trên màn hình.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Người dùng cần đăng nhập.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

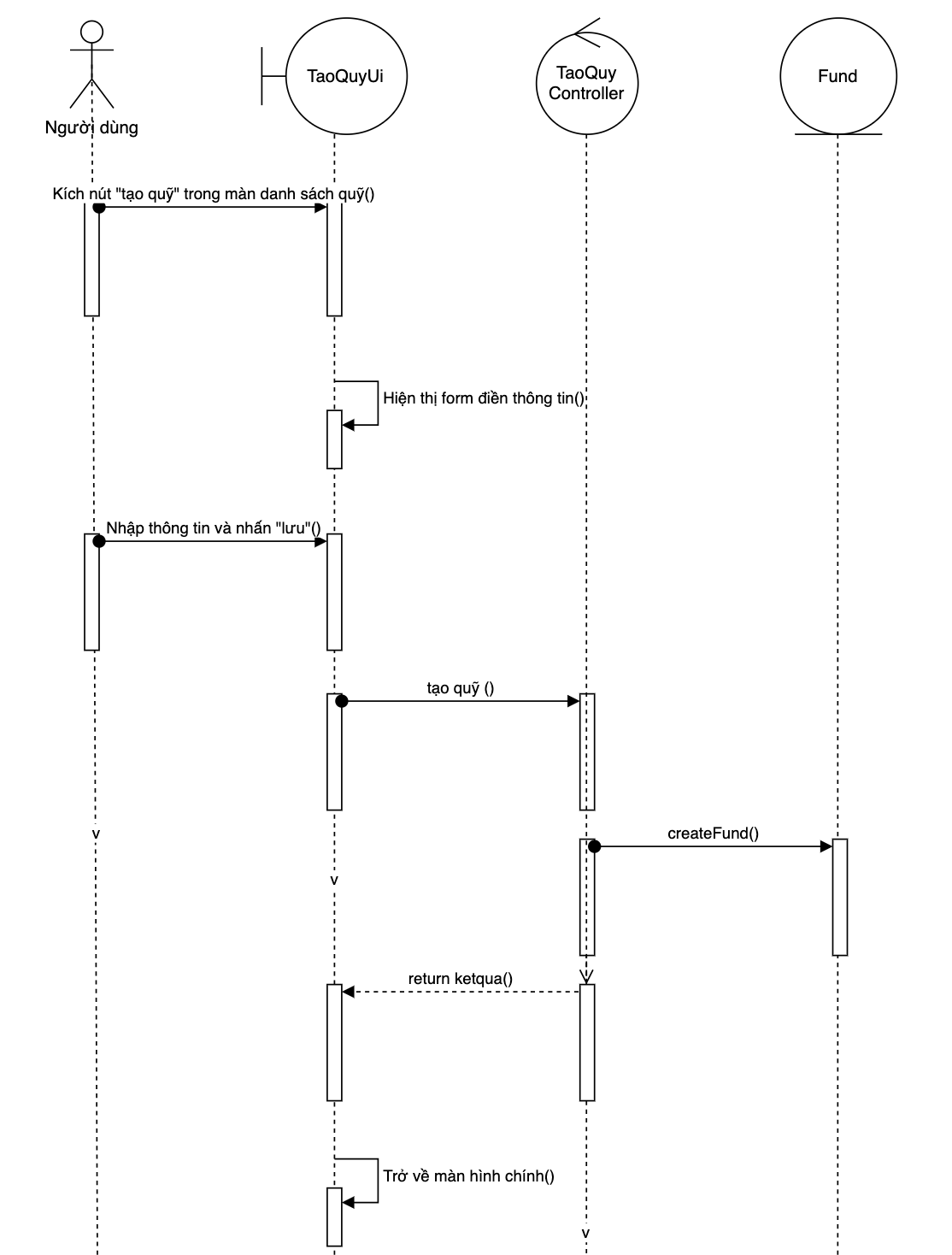
##### Biểu đồ lớp



Hình 2.17 Biểu đồ các lớp Usecase thêm tài liệu cho công ty

#### Usecase Tạo quỹ

##### Biểu đồ trình tự

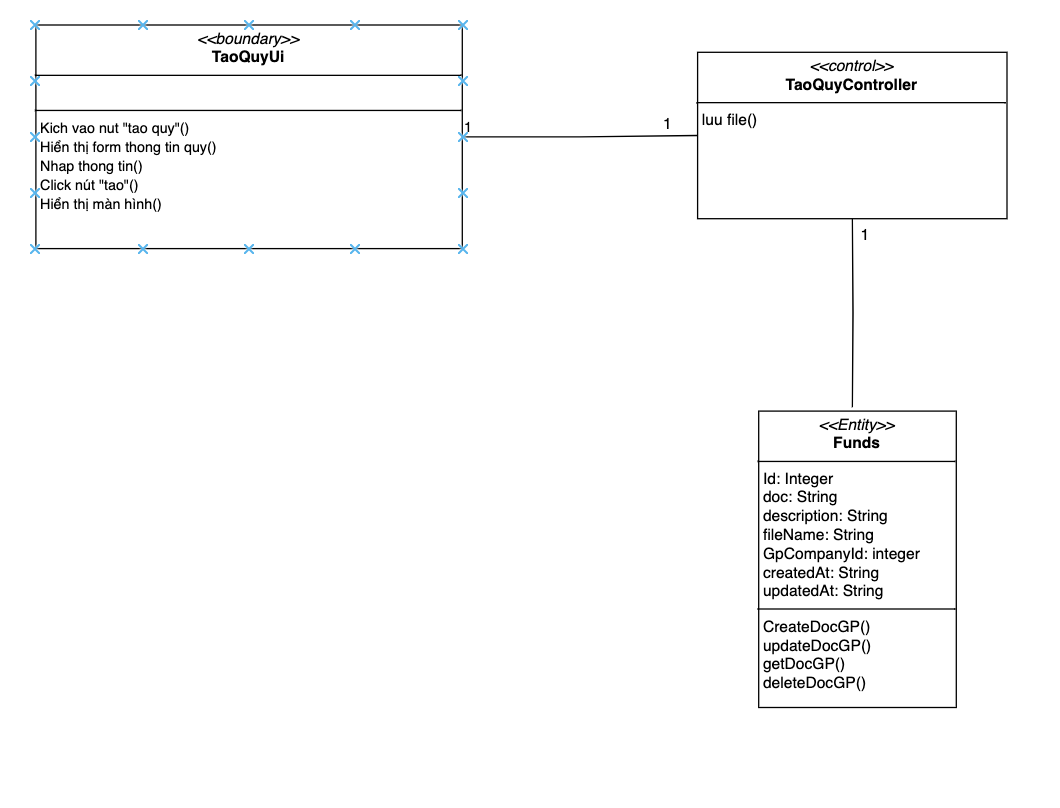


Hình 2.18 Biểu đồ trình tự Usecase tạo quỹ

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người tạo quỹ.
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu sau khi người dùng kích nút “tạo quỹ" trong màn hiển thị danh sách quỹ. sẽ hiển thị một form để người dùng nhập thông tin quỹ vào. Người dùng nhấn nút “tạo” thì hệ thông sẽ tạo một quỹ trên.
    - Usecase kết thúc.
* **Luồng rẽ nhánh:**
  + - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:**
  + Người dùng cần đăng nhập.
  + Người dùng cần quản lý ít nhất một công ty.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

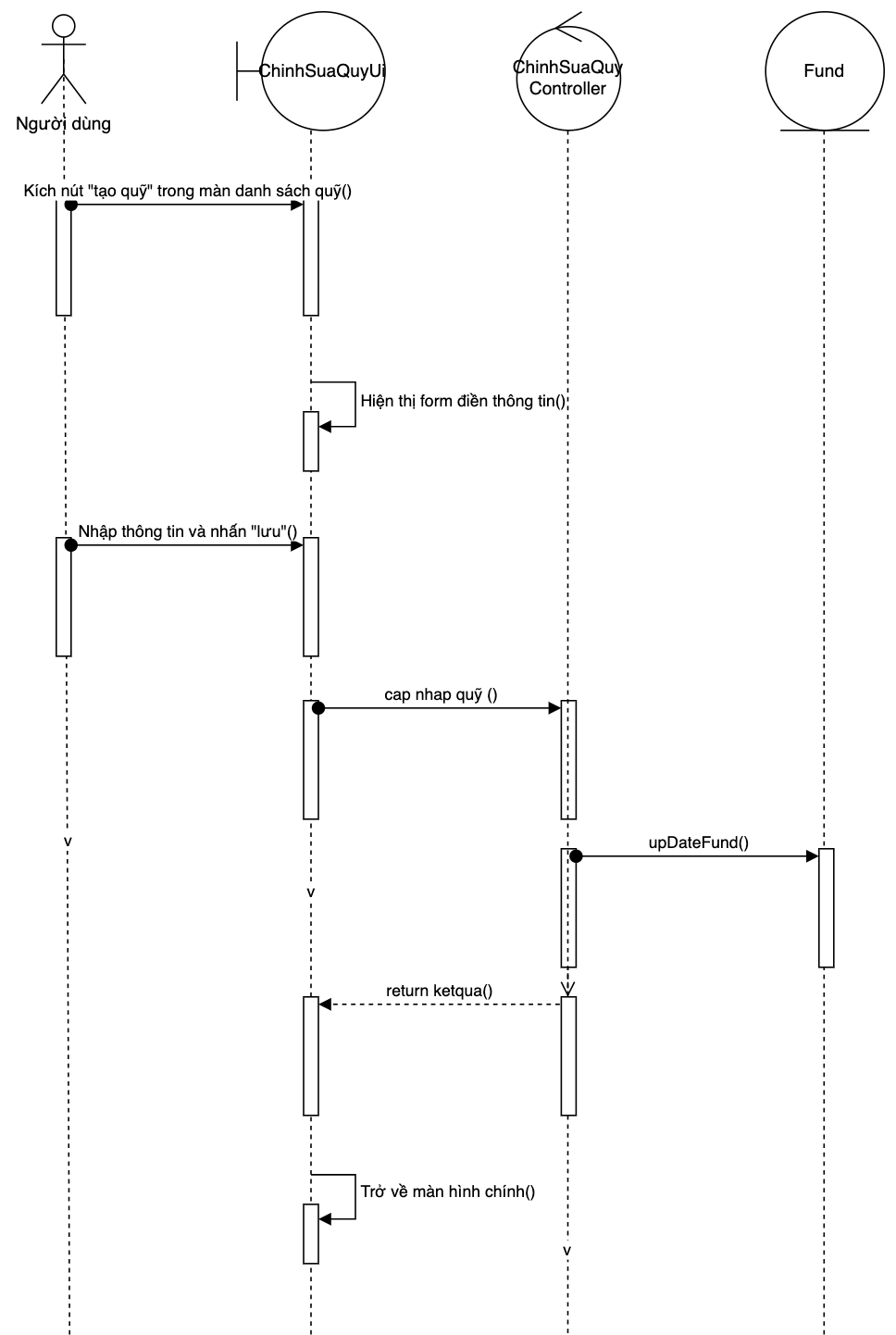
##### Biểu đồ lớp



Hình 2.19 Biểu đồ các lớp Usecase tạo quỹ

#### Usecase Chỉnh sửa quỹ

##### Biểu đồ trình tự



Hình 2.20 Biểu đồ trình tự Usecase chỉnh sửa quỹ

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase cho người quản lý vào thay đổi thông tin.
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu khi người nhấn chuột vào “Tạo quỹ", Hệ thống hiển thị form để nhập các thông tin cần thiết.
    - Sau khi thông tin và ấn nút gửi, Hệ thống sẽ thêm thông tin vào bảng Funds.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** **:** Người dùng phải đăng nhập với tài khoản có quyền quản lý quỹ đó mới được phép chỉnh sửa.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

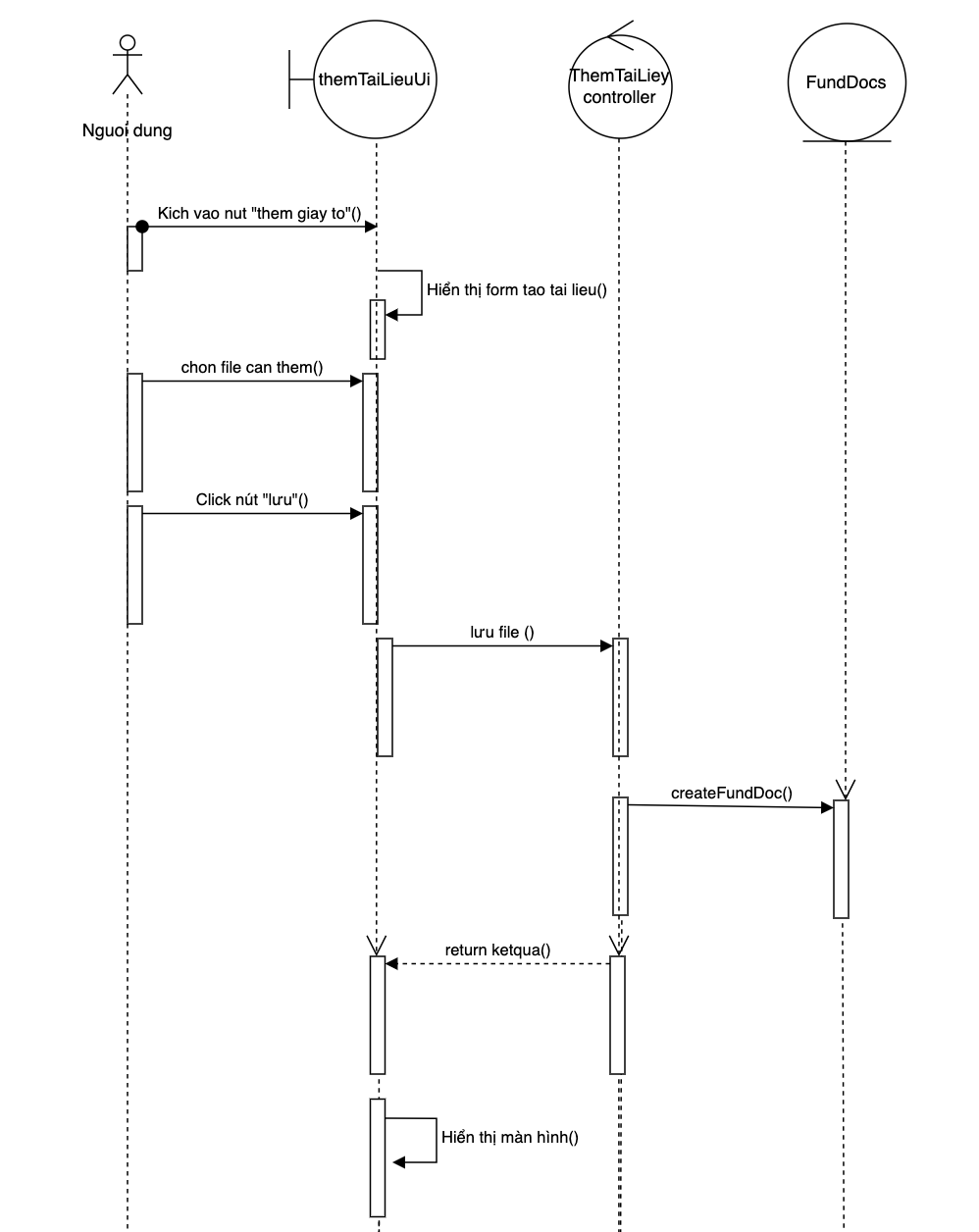
##### Biểu đồ lớp



Hình 2.21 Biểu đồ các lớp Usecase chỉnh sửa bài viết

#### Usecase thêm tài liệu cho quỹ

##### Biểu đồ trình tự

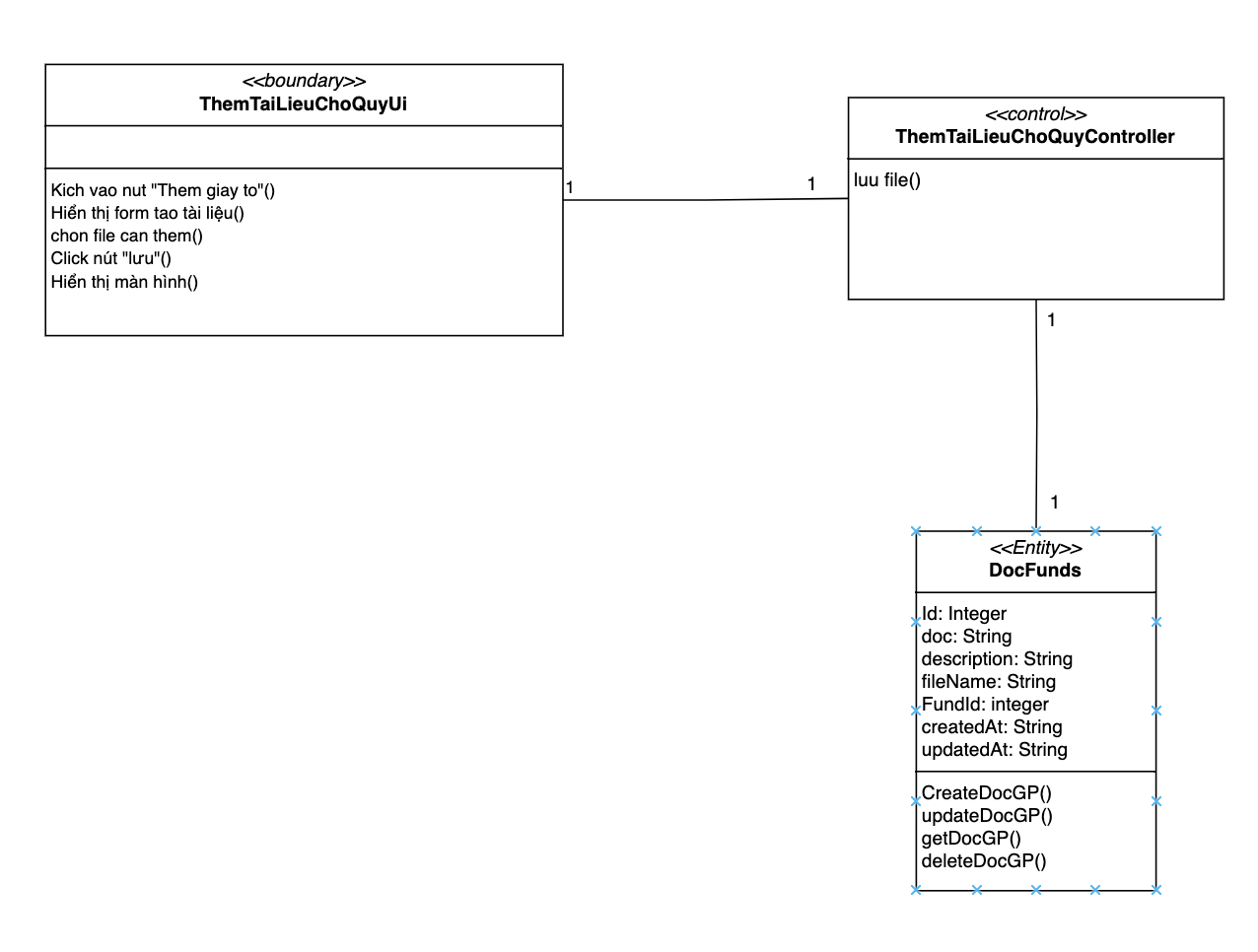


Hình 2.22 Biểu đồ trình tự Usecase thêm tài liệu cho quỹ

##### Đặc tả chi tiết

* **Mô tả vắn tắt:** Usecase này cho phép người dùng thêm tài liệu cho một quỹ.
* **Luồng sự kiện:**
  + **Luồng cơ bản:** 
    - Usecase này bắt đầu khi người dùng kích vào nút “Thêm tài liệu” trong trang quản lý Fund. Hệ thống sẽ hiển thị một form để người dùng thả file vào.
    - Sau khi thả file, hệ thống sẽ lưu file ở một bên thứ 3, và các thông tin còn lại sẽ được lưu vào bản FundDocs.
    - Usecase kết thúc.
  + **Luồng rẽ nhánh:**
    - Tại bất cứ thời điểm nào trong quá trình thực hiện usecase nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và usecase kết thúc.
* **Tiền điều kiện:** Người dùng cần đăng nhập với tài khoản là người quản lý fund.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

##### Biểu đồ lớp



Hình 2.23 Biểu đồ các lớp Usecase thêm tài liệu cho quỹ

## Xây dựng cơ sở dữ liệu

### Biểu đồ Entity Relationship Diagram

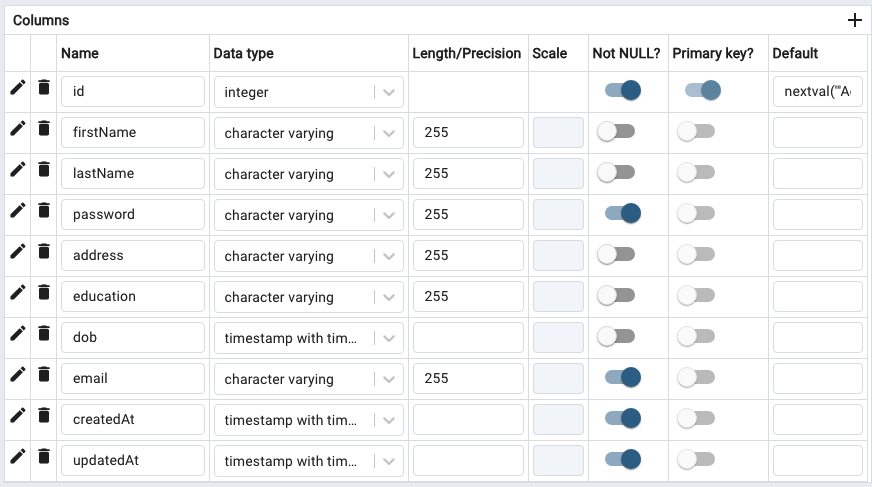
**

Hình 2.30 Biểu đồ Entity Relationship Diagram

### Các bảng trong cơ sở dữ liệu

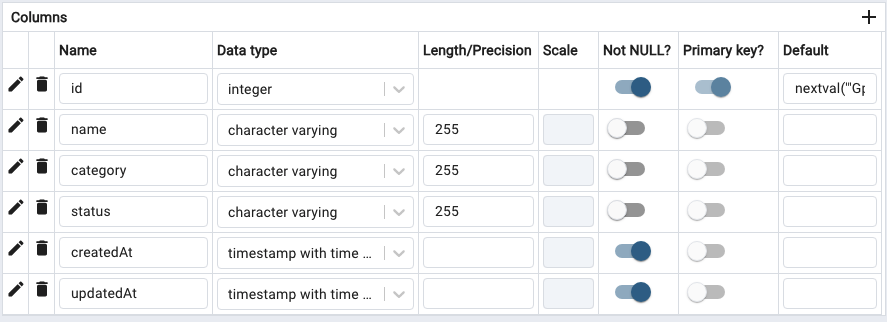
* **Bảng Accounts (Người dùng)**

Bảng 2.2 Bảng Accounts (Người dùng)

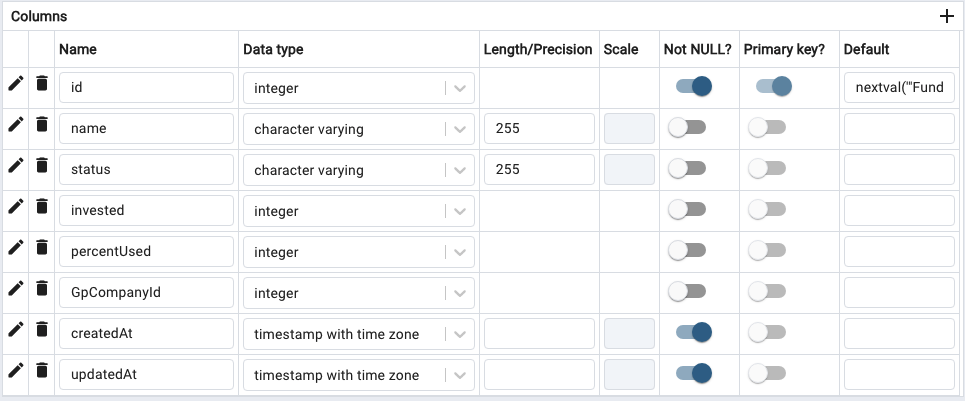
****

* **Bảng GPCompanies (Công ty)**

Bảng 2.3 Bảng GPCompanies (Bài viết)

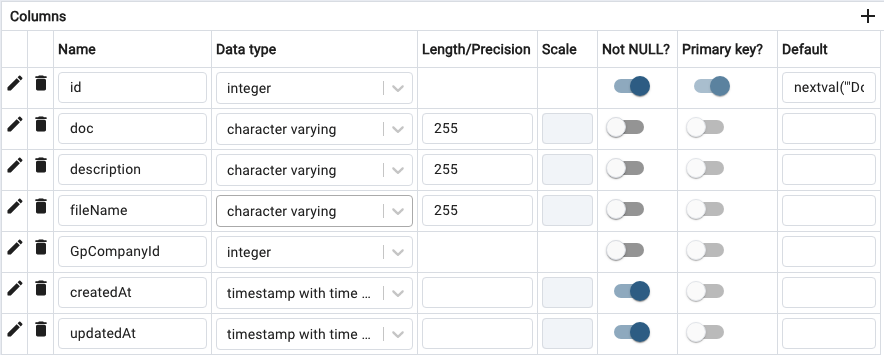
****

* **Bảng Funds (Quỹ)**

Bảng 2.4 Bảng Funds (Qũy)

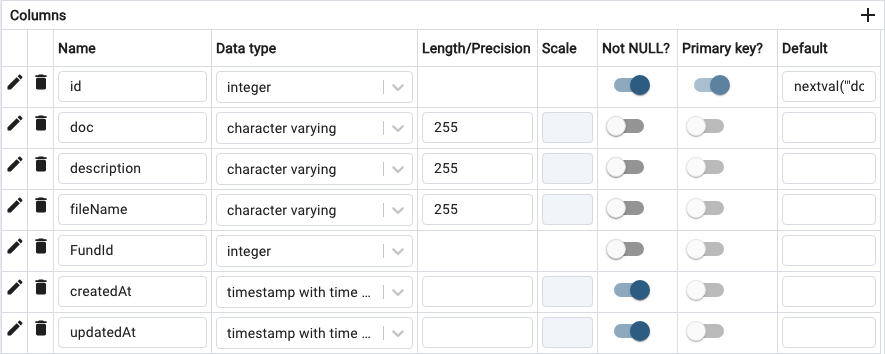
* **Bảng DocGPs (Tài liệu cho công ty)**

Bảng 2.5 Bảng DocGPs (Tài liệu cho công ty)



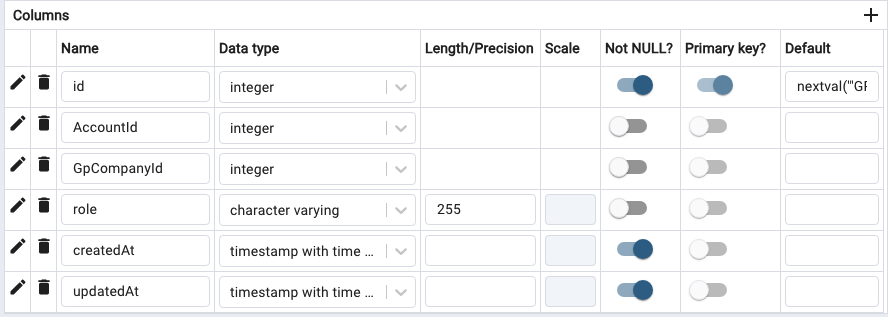
* **Bảng DocFunds (Tài liệu cho quỹ)**

Bảng 2.6 Bảng DocFunds(Tài liệu cho quỹ)



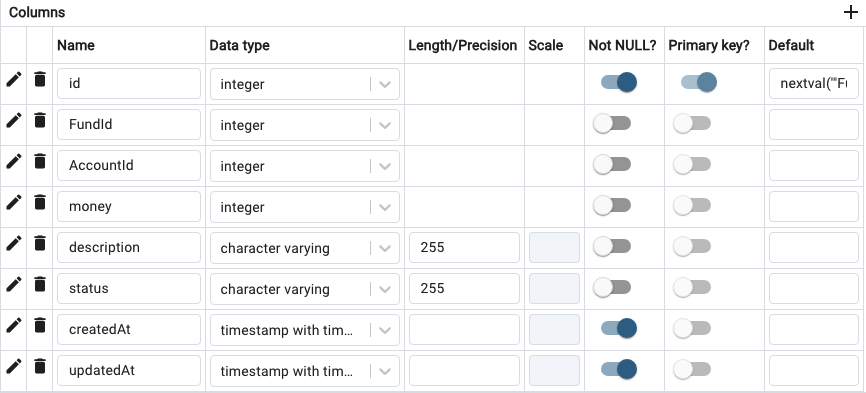
* **Bảng GPCompanyAccounts (Quyền tài khoản với công ty)**

Bảng 2.7 Bảng **GPCompanyAccounts** (quyền tài khoản với công ty)



* **Bảng FundAccounts (Thông tin đầu tư của tài khoản đối với quỹ)**

Bảng 2.8 Bảng FundAccount (Thông tin đầu tư của tài khoản đối với quỹ)

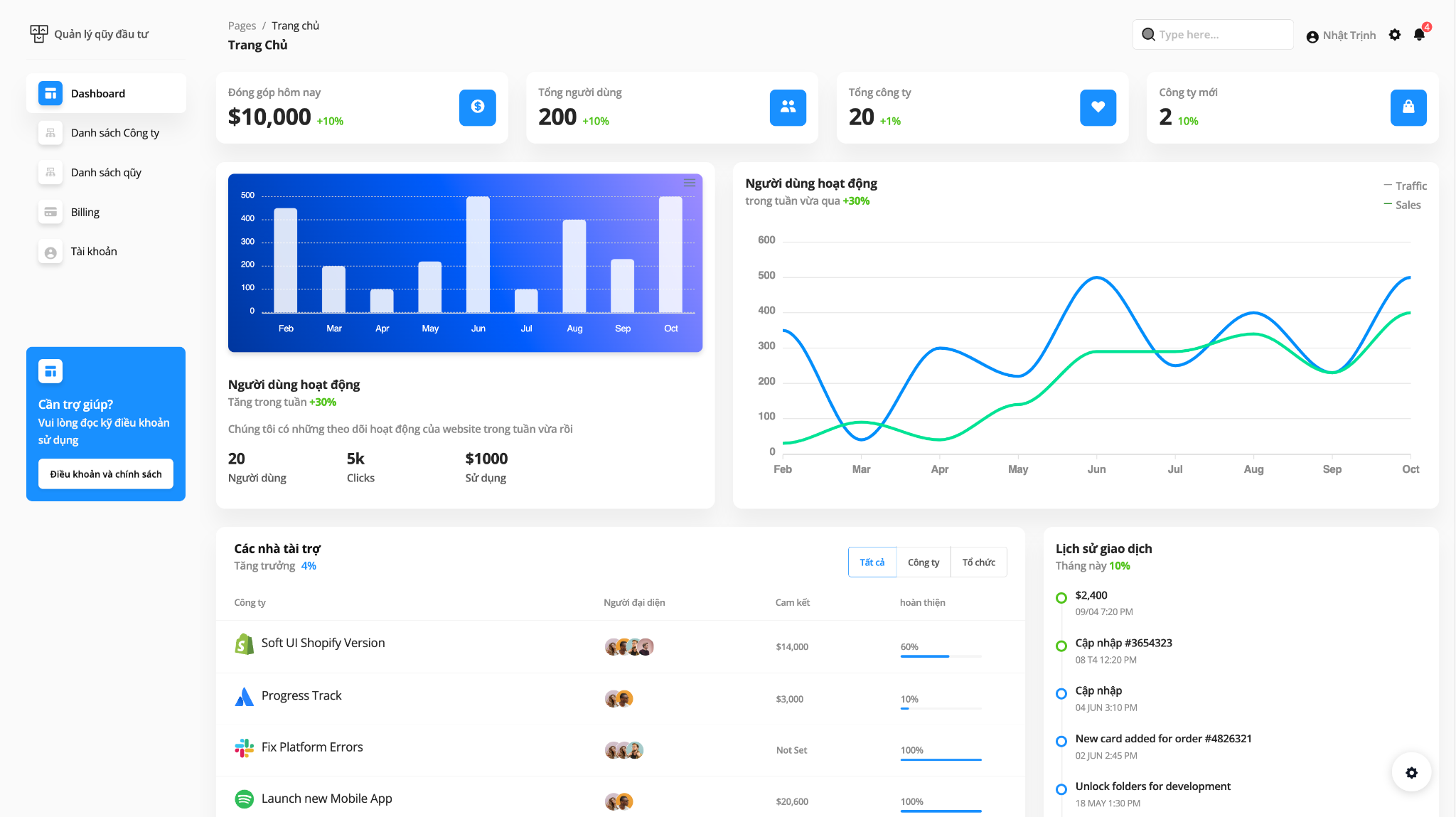


# KẾT QUẢ, KIỂM THỬ VÀ ĐÁNH GIÁ

## Kết quả giao diện màn hình

### Giao diện màn hình trang chủ

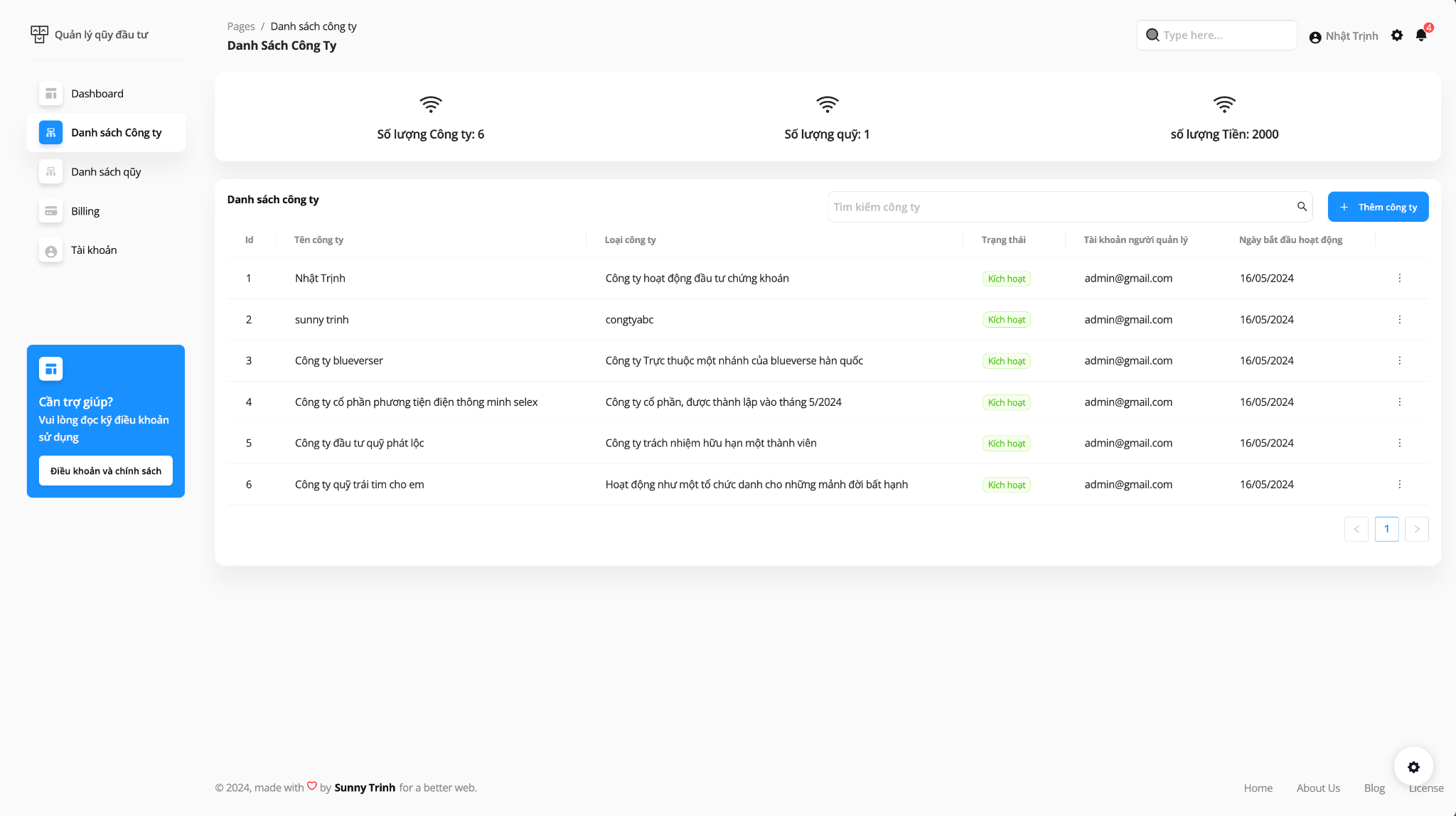
* Mô tả: Người dùng truy cập màn hình trang chủ. Hiện thị trang thái của hệ thống.



Hình 3.1 Giao diện màn hình trang chủ

### Giao diện màn hình tìm kiếm công ty.

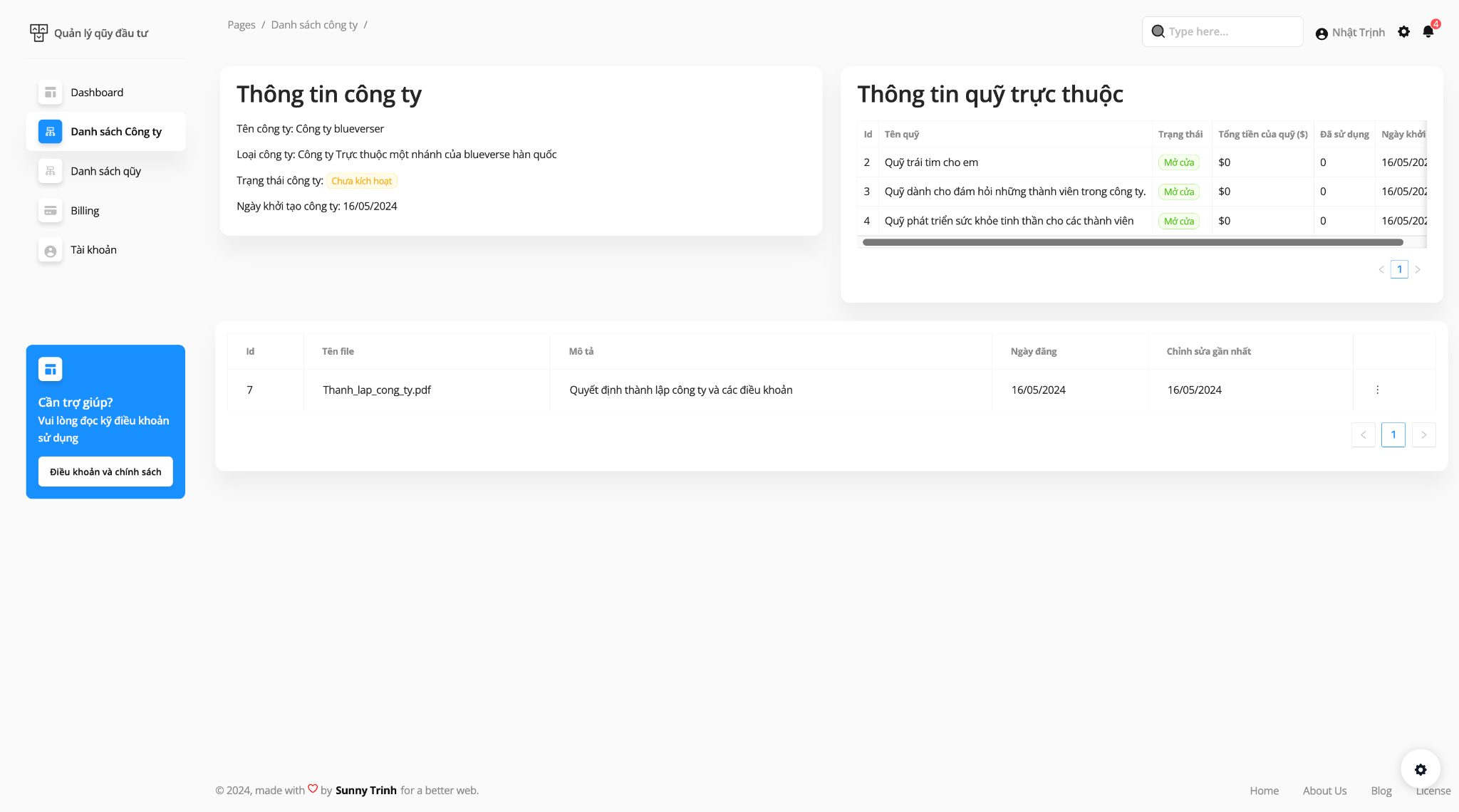
* Mô tả: Người dùng truy cập màn hình tìm kiếm công ty theo tên công ty. mà tài khoản đó đang quản lý.



Hình 3.2 Giao diện màn hình tìm kiếm công ty.

### Giao diện màn hình chi tiết những thông tin liên quan đến công ty.

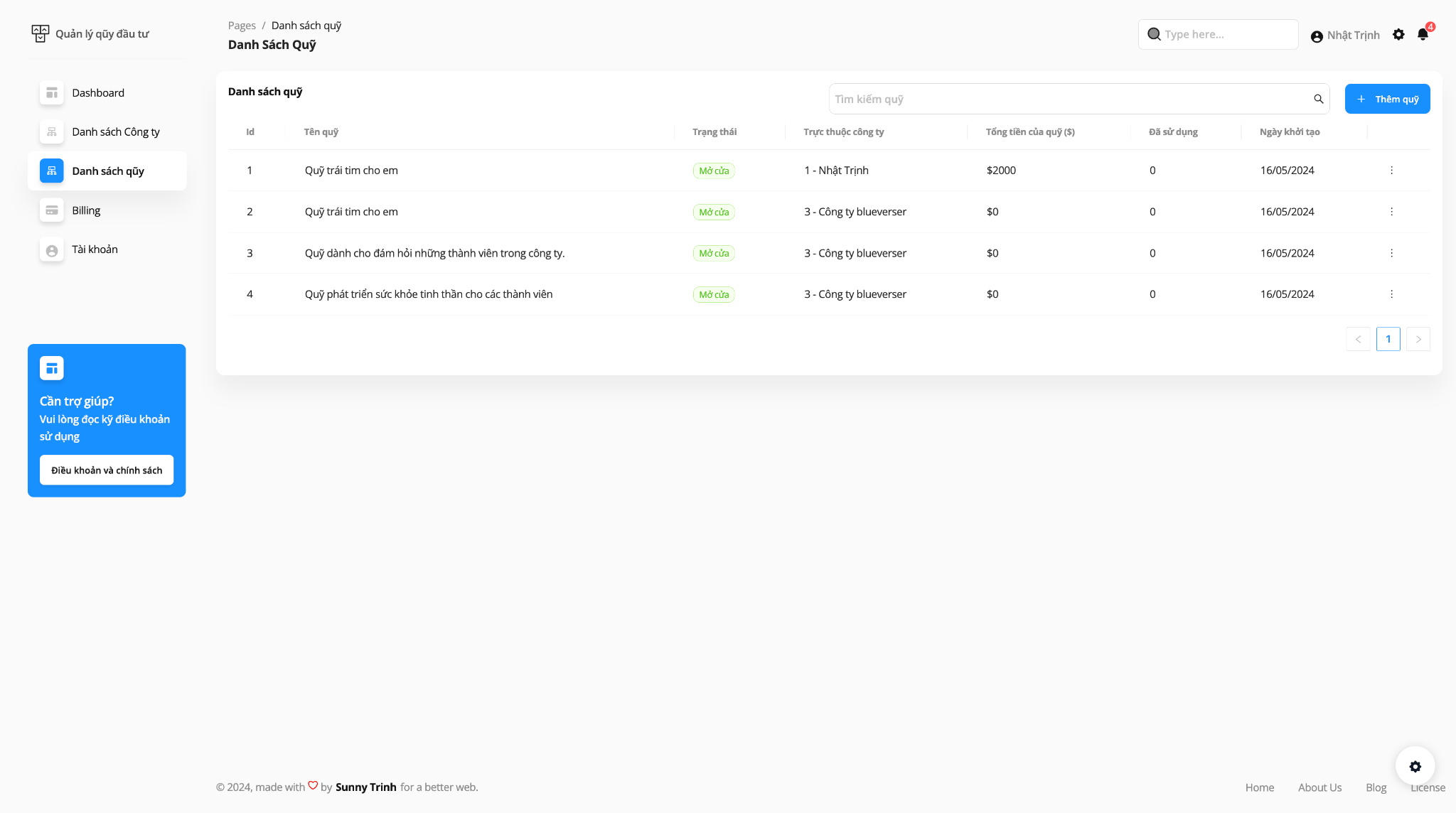
* Mô tả: Người sử dụng có thể thấy được các quỹ trực thuộc công ty, và các giấy tờ thông tin liên quan khác đến công ty.



Hình 3.3 Giao diện thông tin chi tiết về công ty.

### Giao diện danh sách các quỹ

* Mô tả: Sau khi đăng nhập, người dùng truy cập màn hình quản lý quỹ, sẽ thấy được các quỹ mà người dùng quản lý hoặc người dùng đã đóng góp.



Hình 3.4 Giao diện danh sách quỹ trực thuộc

## Kế hoạch kiểm thử

* Những chức năng được kiểm thử:
  + *Đăng nhập:* Người dùng đăng nhập được vào hệ thống.
  + *Đăng ký:* Người dùng đăng ký tài khoản mới
  + *Tạo công ty:* Người dùng tạo công ty.
  + *Tìm kiếm công ty:* Người dùng có thể tìm kiếm công ty.
  + *Chỉnh sửa công ty:* Người dùng chỉnh sửa thông tin công ty.
  + *Tạo quỹ:* Người dùng tạo quỹ.
  + *Đóng góp vào quỹ:* Người dùng nhập số tiền đóng góp vào quỹ.
  + *Cập nhập giấy tờ cho công ty:* Người dùng cập nhập giấy tờ cho công ty.
  + *Cập nhập giấy tờ cho quỹ:* Người dùng cập nhập giấy tờ cho quỹ.

## Chiến lược kiểm thử

* Kiểm thử ở mức hệ thống (ST) và kiểm thử chấp nhận (UAT)
* Dùng kiểu kiểm thử thủ công (manual test) bao gồm kiểm thử GUI và từng chức năng.
* Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử GUI và chức năng.

## Mô hình thử nghiệm

* CPU: AMD Ryzen 5 5600X
* Mainboard: ASUS Prime B550M-A
* RAM: Corsair Vengeance LPX 16GB DDR4 3200MHz
* SSD: Samsung 980 EVO Plus 512GB NVMe M.2
* Card đồ hoạ: NVIDIA GeForce GTX 1660 SUPER 6GB GDDR6
* Hệ điều hành: Windows 10 Pro
* Trình duyệt: Microsoft Edge, Firefox, Chrome

## Kết quả kiểm thử

* Tỉ lệ test case đạt (Passed): 100%
* Tỉ lệ test case không đạt (Failed): 0%
* Hệ thống chạy ổn định với cấu hình (Xem mục 3.3)
* Hệ thống chạy ổn định trên các trình duyệt web khác nhau (Firefox và Google Chrome, Microsoft Edge).
* Kết quả chi tiết (Xem phụ lục).

# KẾT LUẬN

Đề tài ***“Hệ Thống Quản Lý Fund”***, qua quá trình thực hiện đề tài này em đã tổng hợp lại và vận dụng được rất những kiến thức tại trường, tại công ty Blueverse Việt Nam nơi mà em đã thực tập để áp dụng vào thực tiễn như: Mô hình MVC, tối ưu hóa cho SEO, tăng trải nghiệm người dùng, ReactJs, Nodejs và nhiều kiến thức khác.

Nhiều người mới gặp khó khăn khi những thông tin về giấy tờ liên quan quỹ quá rời rạc. Việc xây dựng một hệ thống quản lý quỹ sẽ bắt nhịp xu thế, đáp ứng đầy đủ thông tin về quỹ và minh bạch các thông tin giấy tờ.

Ngoài ra, việc xây dựng một hệ thống quản lý quỹ mang lại nhiều lợi ích kinh tế và xã hội. Nó sẽ tạo ra các công việc mới, cải thiện thu nhập và đóng góp vào sự phát triển kinh tế và xã hội của Việt Nam.

Tuy nhiên trong quá trình thực hiện đề tài, do thời gian có hạn, chưa am hiểu rõ về nghiệp vụ và kiến thức thực tế nên dẫn đến còn nhiều khó khăn trong khi triển khai đề tài làm cho ứng dụng vẫn chưa hoàn thiện, đầy đủ chức năng như mong muốn.

Hướng phát triển của tiếp theo của đề tài sẽ là chỉnh sửa, hoàn thiện các chức năng một cách tối ưu nhất, cập nhật thêm các chức năng khác như: cho phép chỉnh sửa file trực tiếp trên hệ thống, kết hợp với các bên để có thể thanh toán trên web, thông báo thời gian thực.

Hà Nội, ngày 18 tháng 05 năm 2024

Sinh viên thực hiện

***Trịnh Dương Nhật***

# TÀI LIỆU THAM KHẢO

[1] Nguyễn Văn Ba, Phân Tích Thiết Kế Các Hệ Thống Thông Tin Quản Lý, Nhà xuất bản Khoa Học Kỹ Thuật, 2002.

[2] Nguyễn Trường Sinh, Sử dụng PHP & MySQL thiết kế web động, Nhà Xuất Bản Minh Khai.

[3] Thạc Bình Cường, Phân tích thiết kế hệ thống thông tin, Nhà xuất bản Khoa Học Kỹ Thuật, 2002.

**PHỤ LỤC**

*Bảng 1 Bảng kế hoạch phỏng vấn*

|  |  |
| --- | --- |
| **Kế hoạch phỏng vấn** | |
| Người được hỏi:  **Phạm Tuấn Tú** | Người phỏng vấn:  **Trịnh Dương Nhật** |
| Địa chỉ: Lê Văn Lương, Thanh Xuân, Hà Nội | Thời gian hẹn: 60 phút  Thời điểm bắt đầu: 10h30  Thời điểm kết thúc: 11h30 |
| Đối tượng: Người hiện tại đang quản lý 4 quỹ lớn nhỏ. | Các yêu cầu đòi hỏi: Trao đổi về những tồn tại đang gặp phải khi quản lý một lúc nhiều quỹ |
| Chương trình:   * Giới thiệu * Tổng quan về dự án * Chủ đề được đề cập: Xin phép được ghi âm * Câu hỏi 1: Hiện tại một lúc anh phải quản lý nhiều quỹ như thế? Anh có thể chia sẻ cách anh quản lý các quỹ như thế nào? * Câu hỏi 2: Anh thấy có khó khăn gì khi quản lý quỹ như trên không? * Câu hỏi 3: Nếu có một hệ thống có thể xử lý được những bất cập đó, anh có mong muốn gì hơn để tối ưu hóa quá trình quản lý của anh không? * Câu hỏi 4: Anh thường sử dụng những thiết bị nào để xử lý những thao tác quản lý quỹ ? * Câu hỏi 5:Anh có sẵn sàng xuống trả tiền để có thể trải nghiệm tối đa những tính tăng giúp anh tối ưu hóa việc quản lý không? * Câu hỏi 6: Những thống kê nào giúp anh có thể xem được tình trạng của quỹ trong từng giai đoạn? * Tổng hợp những nội dung chính ý kiến của người được hỏi   Kết thúc (thỏa thuận) | Ước lượng thời gian:  5 phút  5 phút    10 phút  5 phút  10 phút  5 phút  10 phút  5 phút  5 phút |
|  | Dự kiến tổng cộng: 60 phút |

*Bảng 2 Bảng kiểm thử chức năng Đăng nhập*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Mục đích kiểm thử | Các bước thực hiện | Kết quả mong muốn | Kết quả đạt được |
|  | Giao diện | | | |
| 1 | Kiểm tra màn hình đăng nhập | 1. Mở màn hình đăng nhập 2. Kiểm tra màn hình đăng nhập | Màn hình hiển thị: - “Tên đăng nhập”, “Mật khẩu” textbox - Text link “Quên mật khẩu” - Text link “Bạn chưa có tài khoản?”  - “Đăng nhập” button | PASS |
| 2 | Kiểm tra tổng thể giao diện màn hình đăng nhập | 1. Mở màn hình đăng nhập 2. Kiểm tra tổng thể giao diện màn hình đăng nhập | - Các label và textbox, rộng và khoảng cách bằng nhau  - Các label sử dụng cùng 1 loại font, cỡ chữ, căn lề trái - Kiểm tra lỗi chính tả - Form đc bố trí hợp lý, dễ sử dụng - Giống với thiết kế đã dựng | PASS |
|  | Chức năng đăng nhập | | | |
| 3 | Kiểm tra đăng nhập tài khoản thành công | 1. Đăng nhập bằng tài khoản đã có trong CSDL 2. Click button "Đăng nhập" | 2. Đăng nhập thành công hiển thị màn hình trang chủ | PASS |
| 4 | Kiểm tra đăng nhập tài khoản không thành công | 1. Đăng nhập bằng tài khoản chưa có trong CSDL 2. Click button "Đăng nhập" | 2. Hiển thị message "Tên user hoặc mật khẩu không chính xác" | PASS |
|  | Textbox tên đăng nhập | | | |
| 5 | Kiểm tra Tên đăng nhập là trường bắt buộc | 1. Để trống trường Tên đăng nhập 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng nhập" | 3. Hiển thị Message "Tên tài khoản không được để trống" | PASS |
| 6 | Kiểm tra nhập khoảng trắng ở Tên đăng nhập (Trimspace) | 1. Nhập khoảng trắng đầu và cuối vào Tên đăng nhập có tồn tại trong DB 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng nhập" | 3. Đăng nhập thành công trim space đầu cuối hiển thị màn hình trang chủ | PASS |
| 7 | Kiểm tra phân biệt chữ hoa, chữ thường | 1. Đăng nhập với Tên dăng nhập là chữ thường có trong DB 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng nhập" | 3. Đăng nhập thành công hiển thị màn hình trang chủ | PASS |
| 8 | 1. Đăng nhập với tên đăng nhập là chữ hoa trùng với chữ thường ở trên có trong DB 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng nhập" | 3. Đăng nhập thành công không phân biệt chữ hoa chữ thường ở tên đăng nhập. Hiển thị Message "Tên user hoặc mật khẩu sai" | PASS |
|  | Textbox mật khẩu | | | |
| 10 | Kiểm tra Mật khẩu là trường bắt buộc | 1. Để trống trường Mật khẩu 2. Các thông tin khác được nhập hợp lệ 3. Click button “Đăng nhập” | 3. Hiển thị Message "Mật khẩu không được để trống" | PASS |
| 11 | Kiểm tra Mật khẩu không chính xác | 1. Nhập sai mật khẩu so với tài khoản có trong CSDL 2. Các trường thông tin khác được nhập hợp lệ 3. Click button "Đăng nhập" | 3. Đăng nhập không thành công hiển thị Message "Tên user hoặc mật khẩu không chính xác" | PASS |
| 12 | Kiểm tra phân biệt chữ hoa, chữ thường | 1. Nhập mật khẩu là chữ thường có trong DB 2. Các thông tin khác được nhập hợp lệ 3. Click button “Đăng nhập” | 3. Hiển thị message đăng nhập thành công hiển thị màn hình trang chủ | PASS |
| 1. Nhập mật khẩu là chữ Hoa trùng với chữ thường ở trên có trong DB 2. Các thông tin khác được nhập hợp lệ 3. Click button “Đăng nhập” | 3. Hiển thị message đăng nhập không thành công | PASS |
| 13 | Kiểm tra mã hóa Mật khẩu | 1. Nhập Mật khẩu | 1. Mật khẩu hiển thị dưới dạng mã hóa \*\*\*\*\*\* | PASS |

*Bảng 3 Bảng kiểm thử chức năng Đăng ký*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Mục đích kiểm thử | Các bước thực hiện | Kết quả mong muốn | Kết quả dạt được |
|  | Giao diện | | | |
| 1 | 1.Kiểm tra giao diện màn hình Đăng ký | 1. Mở màn hình đăng ký 2. Kiểm tra sự đầy đủ của các trường thông tin trên màn hình | Màn hình với đầy đủ thông tin. 2. Giá trị hiển thị của các trường như sau: - Họ và tên textbox\* - Email textbox\* - Tài khoản textbox\* - Mật khẩu textbox\*  - Nhập lại mật khẩu textbox\* - Giới tính radio button\* - Đăng ký button - Button "Đăng nhập" - Text link "Trang chủ" | PASS |
| 2 | Kiểm tra tổng thể giao diện màn hình Đăng ký | 1. Mở màn hình Đăng ký 2. Kiểm tra tổng thể giao diện màn hình Đăng ký | 1. Các label, textbox, combo có độ dài, rộng và khoảng cách bằng nhau, không xô lệch 2. Các label sử dụng cùng 1 loại font, cỡ chữ, căn lề trái 3. Kiểm tra lỗi chính tả 4. Form đc bố trí hợp lý, dễ sử dụng 5. Giống với thiết kế đã dựng. | PASS |
|  | Họ tên textbox | | | |
| 2 | Kiểm tra Họ tên thành công | 1. Nhập họ tên = ký tự 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Đăng ký thành công và chuyển sang màn hình đăng nhập. | PASS |
| 3 | Kiểm tra Họ tên là trường bắt buộc | 1. Để trống trường Họ tên 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị message "Họ tên không được để trống" | PASS |
| 4 | Kiểm tra nhập Họ tên là 30 ký tự nhập vào. | 1. Nhập 30 ký tự vào Họ tên 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Đăng ký thành công và chuyển sang màn hình đăng nhập. | PASS |
| 5 | Kiểm tra nhập Họ tên là ký tự đặc biệt | 1. Nhập Họ tên bao gồm các kí tự đặc biệt (@#$%^\*~/\\,|). 2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1. Hiển thị Message "Bạn cần nhập đúng định dạng. Chỉ được phép nhập các kí tự chữ." | PASS |
| 6 | Kiểm tra Trim Space trong Họ tên textbox | 1. Nhập Họ tên có khoảng trắng ở đầu và cuối. 2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1.1 Đăng ký thành công, đăng nhập và chuyển sang màn hình trang chủ. 1.2 Thực hiện Trim Space ở đầu cuối dữ liệu khi lưu vào CSDL | PASS |
|  | SĐT textbox | | | |
| 7 | Kiểm tra SĐT là trường bắt buộc | 1. Để trống trường số điện thoại 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại không được để trống" | PASS |
| 7 | Kiểm tra SĐT là trường bắt buộc | 1. Để trống trường Tài khoản 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại không được để trống" | PASS |
| 8 | Kiểm tra SĐT đúng định dạng | 1. Nhập ký tự đầu là 0 và 9 kí tự sau là số vào SĐT 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Đăng ký thành công và chuyển sang màn hình đăng nhập. | PASS |
| 9 | Kiểm tra nhập kí tự đầu tiên không là kí tự 0 vào SĐT | 1. Nhập ký tự đầu không là 0 và 9 kí tự sau là số vào SĐT 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại không đúng định dạng" | PASS |
| 10 | Kiểm tra nhập 11 ký tự vào SĐT | 1. Nhập 11 ký tự vào SĐT 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại có 10 chữ số" | PASS |
| 11 | Kiểm tra nhập 9 ký tự vào SĐT | 1. Nhập 9 ký tự vào SĐT 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại có 10 chữ số" | PASS |
| 12 | Kiểm tra nhập SĐT là ký tự đặc biệt | 1. Nhập SĐT bao gồm các kí tự đặc biệt (@#$%^\*~/\\,|). 2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại không đúng định dạng" | PASS |
| 13 | Kiểm tra nhập SĐT có khoảng trắng | 1. Nhập SĐT có khoảng trắng 2. Các trường khác nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Số điện thoại không đúng định dạng" | PASS |
|  | Email textbox | | | |
| 14 | Kiểm tra Email là trường bắt buộc | 1. Để trống trường Email 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message " Email không được để trống." | PASS |
| 15 | Kiểm tra nhập 50 ký tự vào Email | 1. Nhập 50 ký tự vào Email 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị message " Đăng ký thành công" | PASS |
| 16 | Kiểm tra định dạng Email không hợp lệ | 1. Nhập tên email là ký tự đặc biệt (#$%^\*~/\\,|). 2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1. Hiển thị Message "Địa chỉ email không hợp lệ." | PASS |
| 1. Nhập định dạng email ko có @  2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | PASS |
|
| 17 | Kiểm tra Trim Space trong Email textbox | 1. Nhập Email có khoảng trắng đầu, cuối 2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1.1 Đăng ký thành công, đăng nhập và chuyển sang màn hình trang chủ. 1.2 Thực hiện Trim Space ở đầu cuối dữ liệu khi lưu vào CSDL | PASS |
| 18 | Kiểm tra nhập ký tự trắng vào giữa Email | 1. Nhập Email có khoảng trắng ở giữa 2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1. Hiển thị Message "Địa chỉ email không hợp lệ." | PASS |
| 19 | Kiểm tra nhập Email hợp lệ | 1. Nhập địa chỉ email hợp lệ: abc@gmail.com abc.xyz@gmail.com  2. Nhập hợp lệ các trường còn lại 3. Click button "Đăng ký" | 1. Đăng ký thành công, đăng nhập và chuyển sang màn hình trang chủ. | PASS |
|  | Tài khoản textbox | | | |
| 20 | Kiểm tra Tài khoản thành công | 1. Nhập tài khoản = ký tự 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Đăng ký thành công và chuyển sang màn hình đăng nhập. | PASS |
| 21 | Kiểm tra Tài khoản là trường bắt buộc | 1. Để trống trường Tài khoản 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Tài khoản không được để trống" | PASS |
| 22 | Kiểm tra tài khoản đã tồn tại | 1. Nhập Tài khoản đã tồn tại trong Databse 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Username đã được đăng ký trước" | PASS |
| 23 | Kiểm tra nhập ký tự trắng vào giữa Tài khoản | 1. Nhập ký tự trắng vào giữa Tài khoản 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Username viết liền không dấu" | PASS |
| 24 | Kiểm tra nhập ký tự có dấu vào Tài khoản | 1. Nhập ký tự có dấu vào Tài khoản 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Username viết liền không dấu" |  |
| 25 | Kiểm tra nhập 31 ký tự vào Tài khoản | 1. Nhập 31 ký tự vào Tài khoản 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Hiển thị Message "Username tối đa 30 ký tự". | PASS |
|  | Mật khẩu textbox | | | |
| 26 | Kiểm tra Mật khẩu thành công | 1. Nhập mật khẩu = ký tự  2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Đăng ký thành công và chuyển sang đăng nhập. | PASS |
| 27 | Kiểm tra Mật khẩu là trường bắt buộc | 1. Để trống trường Mật khẩu 2. Các thông tin khác được nhập hợp lệ 3 Click button "Đăng ký" | 1. Hiển thị Message "Mật khẩu không được để trống" | PASS |
| 28 | Kiểm tra nhập 10 ký tự vào Mật khẩu | 1. Nhập 10 ký tự vào Mật khẩu 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 1. Đăng ký thành công và chuyển sang màn hình đăng nhập. | PASS |
| 29 | Kiểm tra Trim Space trong Mật khẩu textbox | 1. Nhập Mật khẩu có khoảng trắng ở đầu và cuối. 2. Nhập hợp lệ các trường còn lại 3. Click button "Gửi" | 1.1 Đăng ký thành công, đăng nhập và chuyển sang màn hình trang chủ. 1.2 Thực hiện Trim Space ở đầu cuối dữ liệu khi lưu vào CSDL | PASS |
| 30 | Kiểm tra Mật khẩu thành công | 1. Nhập Mậy khẩu = ký tự 2. Các thông tin khác được nhập hợp lệ 3. Click button "Đăng ký" | 3. Đăng ký thành công về giao diện trang đăng nhập | PASS |