**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

BÁO CÁO BÀI TẬP LỚN

MÔN: CÔNG NGHỆ JAVA

**ĐỀ TÀI: QUẢN LÝ CÔNG TÁC TỪ THIỆN**

Giảng viên hướng dẫn: ThS. TRẦN THỊ DUNG

Sinh viên thực hiện: TRẦN THANH NGÂN MSSV:6451071048 PHẠM LÊ ANH KHOA MSSV:6451071035 NGÔ THÀNH KHÔI MSSV:6451071038

Lớp: CQ.64.CNTT

Khoá: K64

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

BÁO CÁO BÀI TẬP LỚN

MÔN: CÔNG NGHỆ JAVA

**ĐỀ TÀI: QUẢN LÝ CÔNG TÁC TỪ THIỆN**

Giảng viên hướng dẫn: ThS. TRẦN THỊ DUNG

Sinh viên thực hiện: TRẦN THANH NGÂN MSSV:6451071048 PHẠM LÊ ANH KHOA MSSV:6451071035 NGÔ THÀNH KHÔI MSSV:6451071038

Lớp: CQ.64.CNTT

Khoá: K64

# LỜI CẢM ƠN

Lời đầu tiên chúng em xin gửi lời cảm ơn chân thành đến Quý Thầy Cô trong Bộ môn Công nghệ Thông tin, Trường Đại học Giao thông Vận tải – Phân hiệu TP. Hồ Chí Minh, cùng lời chúc sức khỏe và thành công. Chúng em vô cùng biết ơn những kiến thức chuyên môn quý báu và sự tận tình hướng dẫn của quý thầy cô trong quá trình học tập vừa qua tại trường.

Kế tiếp, chúng em xin gửi lời cảm ơn sâu sắc đến cô Trần Thị Dung, người đã trực tiếp chỉ bảo, hướng dẫn chúng em. Nhờ sự hướng dẫn tận tình của cô, chúng em đã có thể hoàn thiện bài báo cáo một cách tốt nhất.

Chúng em cũng xin cảm ơn nhà trường đã tạo điều kiện thuận lợi để chúng em được học tập và nghiên cứu. Cảm ơn các bạn trong lớp K64 đã cùng nhau chia sẻ, hỗ trợ trong suốt quá trình làm bài.

Mặc dù nhóm đã nỗ lực hết mình, nhưng do thời gian có hạn và kinh nghiệm còn non trẻ, bài báo cáo không tránh khỏi những thiếu sót. Rất mong quý thầy cô, các bạn bỏ qua những hạn chế của nhóm em.

Lời sau cùng, chúng em xin gửi lời chúc sức khỏe đến toàn thể giảng viên của trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh đặc biệt là các giảng viên thuộc Bộ Môn Công nghệ Thông tin, các bạn thuộc lớp Công nghệ Thông tin K64, các bạn bè khác của bọn em. Chúng em xin chân thành cảm ơn!

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

…...................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***

**Giáo viên hướng dẫn**

**ThS. Trần Thị Dung**

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc4707)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN ii](#_Toc11388)

[MỤC LỤC iii](#_Toc5575)

[DANH MỤC HÌNH ẢNH iv](#_Toc209)

[DANH MỤC CHỮ VIẾT TẮT vi](#_Toc8680)

[ĐÁNH GIÁ VÀ NHẬN XÉT NHÓM vii](#_Toc21300)

[CHƯƠNG 1: MỞ ĐẦU 1](#_Toc4108)

[1.1 Tổng quan về đề tài 1](#_Toc31739)

[1.2 Mục tiêu nghiên cứu 2](#_Toc21176)

[1.3 Phạm vi đề tài 3](#_Toc14614)

[1.4 Cấu trúc báo cáo 4](#_Toc22442)

[CHƯƠNG 2: TÌM HIỂU NGÔN NGỮ LẬP TRÌNH JAVA 5](#_Toc4753)

[2.1 Tổng quan về ngôn ngữ lập trình Java 5](#_Toc15077)

[2.2 Công nghệ java Swing 8](#_Toc8190)

[2.3 Các công nghệ và công cụ sử dụng 9](#_Toc23264)

[CHƯƠNG 3: ỨNG DỤNG VÀO QUẢN LÝ CÔNG TÁC TỪ THIÊN 11](#_Toc7779)

[3.1 Ứng dụng vào bài toán quản lý công tác từ thiện 11](#_Toc8015)

[3.2 Thiết kế cấu trúc hệ thống 19](#_Toc607)

[3.3 Thiết kế giao diện 31](#_Toc3878)

[CHƯƠNG 4: KẾT QUẢ VÀ KIẾN NGHỊ 49](#_Toc20637)

[4.1 Kết quả đạt được 49](#_Toc30661)

[4.2 Kiến nghị 50](#_Toc5847)

[TÀI LIỆU THAM KHẢO 52](#_Toc13933)

# DANH MỤC HÌNH ẢNH

[Hình ảnh 2.1: Hình ảnh về chương trình cơ bản của JAVA 7](#_Toc24843)

[Hình ảnh 3.1 Tổng quát về chức năng chính 13](#_Toc31902)

[Hình ảnh 3.2 Hình ảnh về bảng account 14](#_Toc2095)

[Hình ảnh 3.3 Hình ảnh bảng User 14](#_Toc15308)

[Hình ảnh 3.4 Hình ảnh bảng Organization 15](#_Toc2677)

[Hình ảnh 3.5 Hình ảnh bảng Event 16](#_Toc23725)

[Hình ảnh 3.5 Hình ảnh bảng Donation 17](#_Toc31580)

[Hình ảnh 3.6 Hình ảnh class diagram 18](#_Toc12157)

[Hình ảnh 3.7 Package controller 20](#_Toc12801)

[Hình ảnh 3.8 Hình ảnh package AdminController 21](#_Toc8042)

[Hỉnh ảnh 3.9 Hình ảnh package LoginController 22](#_Toc17214)

[Hình ảnh 3.10 Hình ảnh package icon và image 23](#_Toc29384)

[Hình ảnh 3.11 Hình ảnh package repository 24](#_Toc13704)

[Hình ảnh 3.12 Hình ảnh package IRepository 24](#_Toc20956)

[Hình ảnh 3.13 Hình ảnh package service 25](#_Toc1298)

[Hình ảnh 3.14 Hình ảnh package Iservice 26](#_Toc28939)

[Hình ảnh 3.15 Hình ảnh package utils 27](#_Toc450)

[Hình ảnh 3.16 Hình ảnh package view.Admin 28](#_Toc24941)

[Hình ảnh 3.17 Hình ảnh package view.Login 29](#_Toc7661)

[Hình ảnh 3.18 Hình ảnh package view.User 30](#_Toc25367)

[Hình ảnh 3.19 Hình ảnh package bean 30](#_Toc15368)

[Hình ảnh 3.20 Hình ảnh package component 31](#_Toc11930)

[Hình ảnh 3.21 Hình ảnh giao diện đăng nhập 32](#_Toc15009)

[Hình ảnh 3.22 Hình ảnh giao diện đăng ký 33](#_Toc6022)

[Hình ảnh 3.23 Hình ảnh giao diện người dùng ở Trang chủ 34](#_Toc14686)

[Hình ảnh 3.24 Hình ảnh giao diện người dùng ở trang sự kiện 35](#_Toc14319)

[Hình ảnh 3.25 Giao diện người dùng ở tổ chức sự kiên 36](#_Toc9952)

[Hình ảnh 3.26 Giao diện người dùng ở danh sách quyên góp 37](#_Toc5455)

[Hình ảnh 3.27 Giao diện người dùng ở lịch sử quyên góp 38](#_Toc26288)

[Hình ảnh 3.28 Giao điện người dùng ở Thông tin cá nhân 39](#_Toc28979)

[Hình ảnh 3.29 Giao diện admin ở trang chủ 39](#_Toc6052)

[Hình ảnh 3.30 Giao diện admin ở quản lý chương trình 40](#_Toc19607)

[Hình ảnh 3.31 Giao diện admin ở quản lý chương trình 41](#_Toc1628)

[Hình ảnh 3.32 Giao điện admin ở quản lý quyên góp 42](#_Toc6880)

[Hình ảnh 3.33 Giao điện admin ở quản lý danh mục 43](#_Toc29901)

[Hỉnh ảnh 3.34 Giao điện admin ở quản lý tổ chức 44](#_Toc18890)

[Hỉnh ảnh 3.35 Giao điện admin ở quản lý tổ chức 44](#_Toc26353)

[Hình ảnh 3.36 Giao điện admin ở quản lý tài khoản 45](#_Toc11734)

[Hình ảnh 3.37 Giao điện admin ở thống kê 46](#_Toc26124)

[Hình ảnh 3.38 Giao điện quên mật khẩu 47](#_Toc7915)

[Hình ảnh 3.37 Giao điện thanh toán momo 48](#_Toc30518)

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| ST T | Chữ viết tắt | Ý nghĩa |
| 1 | UI | User Interface – Giao diện người dùng |
| 2 | UX | User Experience – Trải nghiệm người dùng |
| 3 | MVC | Model – View – Controller (Mô hình kiến trúc phần mềm) |
| 4 | DB | Database – Cơ sở dữ liệu |
| 5 | JDBC | Java Database Connectivity – Kết nối cơ sở dữ liệu trong Java |
| 6 | IDE | Integrated Development Environment – Môi trường phát triển tích hợp |
| 7 | SQL | Structured Query Language – Ngôn ngữ truy vấn có cấu trúc |
| 8 | CRUD | Create, Read, Update, Delete – Các thao tác cơ bản với dữ liệu |
| 9 | NTT | Nhà tài trợ |
| 10 | QG | Quyên góp |
| 11 | TTTT | Tổ chức từ thiện |
| 12 | GVHD | Giảng viên hướng dẫn |

# ĐÁNH GIÁ VÀ NHẬN XÉT NHÓM

**Phân công việc trên Jira:**

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Thành viên | Mức độ đóng góp  (5) | Tiến độ hoàn thành  (3) | Thái độ làm việc  (2) | Tổng điểm (10) | Ký tên |
| Phạm Lê Anh Khoa | 5 | 3 | 2 | 10 |  |
| Trần Thanh Ngân | 4.5 | 2.5 | 2 | 9 |  |
| Ngô Thành Khôi | 3.5 | 2 | 2 | 7.5 |  |

# CHƯƠNG 1: MỞ ĐẦU

## Tổng quan về đề tài

Trong những năm gần đây, hoạt động từ thiện đã và đang trở thành một phần quan trọng trong đời sống xã hội, góp phần hỗ trợ những người có hoàn cảnh khó khăn, trẻ em mồ côi, người già neo đơn, người khuyết tật, cũng như các gia đình chịu ảnh hưởng bởi thiên tai, dịch bệnh. Hoạt động từ thiện không chỉ được thực hiện bởi các tổ chức phi chính phủ, mà còn lan rộng đến các doanh nghiệp, trường học và cá nhân có lòng hảo tâm. Tuy nhiên, sự phát triển nhanh chóng về quy mô và phạm vi của các hoạt động từ thiện đặt ra nhiều thách thức trong việc quản lý, theo dõi, thống kê, minh bạch hóa thông tin và báo cáo tài chính.

Hiện tại, ở nhiều tổ chức từ thiện vừa và nhỏ, việc lưu trữ thông tin vẫn chủ yếu thực hiện theo cách thủ công như ghi chép trên giấy, file Excel, hoặc phần mềm đơn giản, dẫn đến khó khăn trong việc:

* Truy xuất dữ liệu nhanh chóng và chính xác;
* Theo dõi lịch sử ủng hộ hoặc cấp phát hỗ trợ;
* Phân loại nhà tài trợ và người nhận theo tiêu chí cụ thể;
* Đảm bảo tính minh bạch và kiểm tra chéo khi cần thiết.

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, nhu cầu xây dựng các hệ thống phần mềm phục vụ công tác từ thiện là rất cấp thiết. Một hệ thống phần mềm tốt sẽ giúp cho tổ chức từ thiện:

* Quản lý tập trung và có tổ chức toàn bộ dữ liệu;
* Ghi nhận đầy đủ lịch sử tài trợ và cấp phát;
* Tự động hóa quá trình thống kê, báo cáo;
* Nâng cao uy tín và lòng tin của cộng đồng.

Đề tài "Quản lý công tác từ thiện" nhằm hiện thực hóa nhu cầu này bằng việc xây dựng một phần mềm ứng dụng được phát triển trên nền tảng Java với giao diện đồ họa sử dụng Java Swing và kết nối cơ sở dữ liệu MySQL. Phần mềm sẽ hỗ trợ quản lý các đối tượng chính như nhà tài trợ, người nhận hỗ trợ, các chương trình từ thiện, đồng thời cung cấp chức năng ghi nhận các khoản

ủng hộ, cấp phát và thống kê số liệu. Qua đó, giúp tổ chức từ thiện quản lý hiệu quả, khoa học, minh bạch và đáng tin cậy hơn.

Ngoài ra, việc xây dựng hệ thống này cũng là cơ hội để sinh viên áp dụng những kiến thức đã học như lập trình hướng đối tượng, quản lý cơ sở dữ liệu, thiết kế giao diện người dùng và phát triển phần mềm thực tế. Đây là bước đệm quan trọng giúp sinh viên nâng cao kỹ năng, sẵn sàng cho các dự án thực tiễn sau khi ra trường.

## Mục tiêu nghiên cứu

Mục tiêu của đề tài “Quản lý công tác từ thiện” là xây dựng một phần mềm ứng dụng giúp tổ chức, doanh nghiệp hoặc nhóm tình nguyện quản lý hiệu quả toàn bộ hoạt động liên quan đến từ thiện. Đề tài được triển khai với các mục tiêu cụ thể như sau:

1. **Xây dựng hệ thống phần mềm ứng dụng viết bằng Java (Java Swing)** có giao diện thân thiện, dễ sử dụng, phù hợp với người dùng không chuyên về CNTT.
2. **Quản lý thông tin nhà tài trợ bao gồm:** họ tên, địa chỉ, số điện thoại, email, số tiền đã ủng hộ,... phục vụ mục đích lưu trữ và thống kê sau này.
3. **Quản lý người nhận hỗ trợ:** lưu trữ đầy đủ thông tin cá nhân, hoàn cảnh khó khăn, địa chỉ cư trú,... phục vụ công tác xét duyệt, cấp phát hỗ trợ minh bạch.
4. **Quản lý các chương trình từ thiện:** tạo lập, chỉnh sửa thông tin về các chương trình như cứu trợ bão lụt, hỗ trợ học sinh nghèo, xây nhà tình thương,... cùng các thông tin liên quan như thời gian, nội dung, ngân sách dự kiến.
5. **Ghi nhận các khoản ủng hộ và cấp phát:** liên kết giữa nhà tài trợ – chương trình từ thiện – người nhận; theo dõi lịch sử các giao dịch (số tiền, thời gian, nội dung).
6. **Thống kê – Báo cáo:** cho phép thống kê số tiền đã ủng hộ, số tiền cấp phát, số lượng người nhận hỗ trợ, số lượng nhà tài trợ,... có thể xuất thành báo cáo dạng bảng hoặc biểu đồ.
7. **Ứng dụng mô hình lập trình hướng đối tượng (OOP)** và mô hình kiến trúc phần mềm MVC (Model – View – Controller) nhằm đảm bảo tính mở rộng, dễ bảo trì và nâng cấp trong tương lai.
8. **Tạo tiền đề để phát triển hệ thống quy mô lớn hơn** (trên nền tảng web hoặc mobile), phục vụ cộng đồng tốt hơn, tăng cường khả năng tương tác và công khai hóa dữ liệu.

Tóm lại, phần mềm nhằm cung cấp một giải pháp quản lý toàn diện, khoa học và hiện đại cho các tổ chức từ thiện, đồng thời là bài thực hành thực tiễn giúp sinh viên vận dụng hiệu quả kiến thức đã học vào một dự án phần mềm cụ thể.

## Phạm vi đề tài

Đề tài “Quản lý công tác từ thiện” được triển khai dưới hình thức xây dựng một phần mềm ứng dụng chạy trên nền tảng desktop, sử dụng ngôn ngữ lập trình Java và cơ sở dữ liệu MySQL. Trong khuôn khổ bài tập lớn, phạm vi nghiên cứu và phát triển của đề tài được xác định cụ thể như sau:

* Về chức năng:
  + Hệ thống cho phép quản lý thông tin của các đối tượng: nhà tài trợ, người nhận hỗ trợ, chương trình từ thiện.
  + Ghi nhận và lưu trữ các khoản ủng hộ từ nhà tài trợ đến các chương trình từ thiện.
  + Ghi nhận thông tin cấp phát hỗ trợ từ chương trình đến người nhận.
  + Hỗ trợ tìm kiếm, tra cứu thông tin theo nhiều tiêu chí.
  + Tính năng thống kê – báo cáo tổng số tiền ủng hộ, tổng cấp phát, số lượng đối tượng liên quan.
  + Xuất báo cáo dạng bảng (JTable) hoặc in (JasperReports).
* Về đối tượng quản lý:
  + Chỉ quản lý các thông tin cơ bản, cần thiết nhất của nhà tài trợ, người nhận, chương trình (không tích hợp ảnh, giấy tờ minh chứng,...).
  + Không bao gồm chức năng quản lý người dùng nhiều cấp độ hoặc phân quyền chi tiết (chỉ 1 tài khoản admin sử dụng).
* Về kỹ thuật:
  + Giao diện được thiết kế bằng Java Swing, ứng dụng dạng desktop.
  + Cơ sở dữ liệu sử dụng MySQL, truy vấn thông qua JDBC.
  + Không triển khai trên nền tảng web hoặc di động.
  + Không sử dụng framework nâng cao (Spring, Hibernate), nhằm phù hợp với kiến thức môn học.
* Về quy mô:
  + Hệ thống hướng đến phục vụ các tổ chức từ thiện vừa và nhỏ, nhóm tình nguyện viên hoặc cá nhân làm thiện nguyện tự phát.
  + Dữ liệu chưa quá lớn, không đặt nặng vấn đề hiệu năng, bảo mật nâng cao hay tối ưu hóa hệ thống.

Như vậy, phạm vi đề tài được thiết kế phù hợp với năng lực sinh viên, đảm bảo có tính thực tiễn nhưng vẫn nằm trong khả năng triển khai trong thời gian học phần. Đề tài có thể được mở rộng trong tương lai nếu tích hợp thêm các công nghệ web hoặc di động.

## Cấu trúc báo cáo

* + 1. Chương 1: Giới thiệu đề tài
    2. Chương 2: Phân tích – Thiết kế hệ thống
    3. Chương 3: Cài đặt phần mềm
    4. Chương 4: Kết quả và kiến nghị

# CHƯƠNG 2: TÌM HIỂU NGÔN NGỮ LẬP TRÌNH JAVA

## Tổng quan về ngôn ngữ lập trình Java

* + 1. Lịch sử ra đời và phát triển

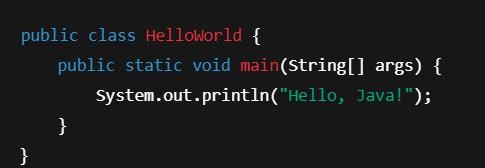
Java được phát triển bởi James Gosling và nhóm kỹ sư tại Sun Microsystems vào năm 1995. Ban đầu, Java được thiết kế cho các thiết bị điện tử gia dụng, nhưng nhanh chóng trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới nhờ triết lý “Write Once, Run Anywhere” – viết một lần, chạy ở mọi nơi. Năm 2010, Oracle Corporation chính thức mua lại Sun Microsystems và trở thành đơn vị phát triển, quản lý Java đến nay.

* + 1. Đặc điểm nổi bật của Java
  + Lập trình Hướng đối tượng (Object-Oriented Programming - OOP):

Java là một ngôn ngữ lập trình hướng đối tượng thuần túy và đầy đủ. Điều này có nghĩa là mọi thứ trong Java đều được tổ chức dưới dạng các đối tượng, tương tác với nhau thông qua các lớp (classes) và phương thức (methods). Các nguyên tắc cơ bản của OOP như:

* Đóng gói (Encapsulation): Giúp che giấu chi tiết triển khai và bảo vệ dữ liệu bên trong một đối tượng, chỉ cho phép truy cập thông qua các giao diện công khai (public interfaces). Điều này nâng cao tính bảo mật và dễ dàng bảo trì.
* Kế thừa (Inheritance): Cho phép các lớp mới (lớp con) kế thừa các thuộc tính và hành vi từ các lớp hiện có (lớp cha), thúc đẩy việc tái sử dụng mã (code reusability) và giảm thiểu sự trùng lặp.
* Đa hình (Polymorphism): Cho phép các đối tượng thuộc các lớp khác nhau được xử lý thông qua một giao diện chung, mang lại sự linh hoạt cao trong thiết kế và triển khai các chức năng.
* Trừu tượng (Abstraction): Tập trung vào việc hiển thị những thông tin cần thiết và ẩn đi các chi tiết phức tạp không liên quan, giúp đơn giản hóa việc quản lý hệ thống lớn. Nhờ các đặc tính này, việc phát triển hệ thống trở nên có cấu trúc, dễ quản lý, dễ mở rộng và bảo trì, đặc biệt phù hợp với một ứng dụng quản lý có nhiều module và nghiệp vụ phức tạp như quản lý từ thiện.
* Độc lập nền tảng (Platform Independence - "Write Once, Run Anywhere"):
* Đây là một trong những ưu điểm nổi bật nhất của Java. Chương trình Java không được biên dịch trực tiếp thành mã máy của một hệ điều hành cụ thể mà được biên dịch thành bytecode. Bytecode này sau đó có thể được thực thi trên bất kỳ nền tảng nào (Windows, macOS, Linux, v.v.) miễn là có cài đặt Môi trường chạy Java (Java Runtime Environment - JRE) hoặc Máy ảo Java (Java Virtual Machine - JVM) tương ứng. Điều này mang lại sự linh hoạt vượt trội, giúp ứng dụng có thể triển khai trên nhiều môi trường khác nhau mà không cần sửa đổi mã nguồn, tiết kiệm thời gian và công sức phát triển.
* Bảo mật cao:
* Java được thiết kế với các cơ chế bảo mật mạnh mẽ ngay từ đầu. JVM thực hiện một loạt các kiểm tra an ninh trong quá trình tải và thực thi bytecode, ngăn chặn mã độc hại truy cập vào các tài nguyên nhạy cảm của hệ thống. Các tính năng bảo mật bao gồm:
* Sandbox Security Model: Hạn chế quyền truy cập của ứng dụng vào hệ thống tập tin và mạng, giảm thiểu rủi ro từ các ứng dụng không đáng tin cậy.
* Garbage Collection: Quản lý bộ nhớ tự động, giúp ngăn chặn các lỗi rò rỉ bộ nhớ hoặc truy cập bộ nhớ trái phép, vốn là nguyên nhân phổ biến của các lỗ hổng bảo mật.
* Exception Handling: Cơ chế xử lý ngoại lệ chặt chẽ giúp ứng dụng phục hồi gracefully sau các lỗi, ngăn chặn các sự cố hệ thống có thể bị lợi dụng. Đối với một hệ thống quản lý từ thiện, nơi thông tin nhạy cảm của người quyên góp và người nhận hỗ trợ được xử lý, tính bảo mật cao của Java là một yếu tố then chốt.
* Thư viện phong phú và cộng đồng lớn:
* Java cung cấp một bộ thư viện chuẩn (Java Standard Library hay Java API) vô cùng mạnh mẽ và toàn diện, bao gồm các gói (packages) cho nhiều mục đích khác nhau như:
* java.lang: Các lớp cơ bản.
* java.io: Xử lý đầu vào/đầu ra.
* java.net: Lập trình mạng.
* java.util: Các cấu trúc dữ liệu và tiện ích.
* javax.swing và java.awt: Phát triển giao diện người dùng đồ họa (GUI) cho ứng dụng desktop. Ngoài ra, có một hệ sinh thái thư viện và framework của bên thứ ba khổng lồ (ví dụ: Spring, Hibernate, Apache Commons) cùng với một cộng đồng lập trình viên Java toàn cầu rất lớn và tích cực. Điều này đồng nghĩa với việc dễ dàng tìm kiếm tài liệu, giải pháp cho các vấn đề phát sinh và hỗ trợ kỹ thuật, giúp quá trình phát triển nhanh chóng và hiệu quả hơn.
* Quản lý bộ nhớ tự động (Automatic Memory Management):
* Java tích hợp một bộ thu gom rác (Garbage Collector - GC) tự động. Nhiệm vụ của GC là tự động giải phóng bộ nhớ đã được cấp phát cho các đối tượng không còn được sử dụng bởi chương trình. Điều này giúp lập trình viên không phải lo lắng về việc quản lý bộ nhớ thủ công (như trong C++), từ đó giảm đáng kể các lỗi liên quan đến rò rỉ bộ nhớ (memory leaks) hoặc truy cập con trỏ không hợp lệ, giúp mã ổn định và đáng tin cậy hơn.Cấu trúc cơ bản của chương trình Java

Một chương trình Java đơn giản luôn bắt đầu với lớp (class) và có phương thức main() như sau:



Hình ảnh 2.1: Hình ảnh về chương trình cơ bản của JAVA

**Giải thích:**

* **public class HelloWorld:** khai báo một lớp có tên HelloWorld.
* **public static void main (String[] args):** phương thức chính, là điểm bắt đầu thực thi chương trình.
* **System.out.println:** in chuỗi ra màn hình.
  + 1. Các thành phần quan trọng trong Java
       - **Biến và kiểu dữ liệu:** gồm các kiểu nguyên thủy (int, float, boolean,...) và kiểu đối tượng.
       - **Câu điều kiện, vòng lặp:** if-else, switch-case, while, for,...
       - **Mảng và chuỗi (Array & String):** xử lý tập hợp dữ liệu.
       - **Lập trình hướng đối tượng:** class, object, kế thừa, đóng gói, đa hình.
       - **Ngoại lệ (Exception):** xử lý lỗi thời gian chạy.
       - **Giao diện người dùng (GUI):** Java Swing, JavaFX.
       - **Truy xuất cơ sở dữ liệu:** sử dụng JDBC để kết nối và thao tác với các hệ quản trị CSDL như MySQL, SQLite,…
    2. Ứng dụng thực tế của Java
       - **Ứng dụng desktop:** sử dụng Java Swing hoặc JavaFX.
       - **Ứng dụng web:** Java EE, Spring Framework.
       - **Ứng dụng di động:** nền tảng Android.
       - **Ứng dụng doanh nghiệp:** hệ thống quản lý ERP, CRM,...
       - **Ứng dụng khoa học, tài chính:** xử lý dữ liệu, mô phỏng,…

## 

## Công nghệ java Swing

* + 1. Khái niệm

Java Swing là một thư viện giao diện người dùng (GUI) thuộc Java Foundation Classes (JFC), được giới thiệu trong Java 1.2. Swing cung cấp các thành phần giao diện (như button, text field, label, table) để xây dựng ứng dụng desktop với giao diện đồ họa. Không giống AWT (Abstract Window Toolkit), Swing được viết hoàn toàn bằng Java, không phụ thuộc vào giao diện gốc của hệ điều hành.

* + 1. Các tính năng nổi bật
       - Đa dạng thành phần: Swing cung cấp các thành phần như JButton, JTextField, JLabel, JTable, hỗ trợ xây dựng giao diện phức tạp.
       - Tùy chỉnh cao: Có thể thay đổi giao diện (Look and Feel) như Windows, Metal, Nimbus.
       - Hỗ trợ sự kiện: Swing tích hợp cơ chế xử lý sự kiện mạnh mẽ (ActionListener, MouseListener).
       - Độc lập nền tảng: Giao diện đồng nhất trên mọi hệ điều hành nhờ được vẽ bằng Java.
    2. Ưu nhược điểm
       - Ưu điểm:
         * Dễ tích hợp với Java, phù hợp cho ứng dụng desktop nhỏ và vừa.
         * Linh hoạt trong thiết kế giao diện.
         * Hỗ trợ tốt cho các ứng dụng đơn giản như quản lý chung cư.
       - Nhược điểm:
         * Yêu cầu kiến thức về lập trình sự kiện để xử lý tương tác người dùng.
         * Hiệu suất thấp với ứng dụng phức tạp hoặc đồ họa nặng.
         * Giao diện không hiện đại, khó cạnh tranh với các công nghệ mới như JavaFX, WPF.

## Các công nghệ và công cụ sử dụng

* NetBeans : Là một IDE (Integrated Development Environment) mã nguồn mở, hỗ trợ phát triển ứng dụng Java, bao gồm cả Swing. NetBeans cung cấp trình soạn thảo mã, giao diện thiết kế GUI kéo-thả, và công cụ debug.
* MySQL : Là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được sử dụng để lưu trữ và quản lý dữ liệu trong các ứng dụng. MySQL hỗ trợ SQL (Structured Query Language) để truy vấn và thao tác dữ liệu.
* GitHub : Là một nền tảng quản lý mã nguồn dựa trên Git, cho phép nhiều lập trình viên làm việc cùng lúc trên một dự án. GitHub hỗ trợ lưu trữ, theo dõi phiên bản, và cộng tác.
* Jira : Là một công cụ quản lý dự án và theo dõi lỗi, được phát triển bởi Atlassian. Jira hỗ trợ lập kế hoạch, phân công nhiệm vụ và theo dõi tiến độ phát triển phần mềm.

# CHƯƠNG 3: ỨNG DỤNG VÀO QUẢN LÝ CÔNG TÁC TỪ THIÊN

## Ứng dụng vào bài toán quản lý công tác từ thiện

* + 1. Mô tả bài toán

Trong xã hội hiện nay, các hoạt động từ thiện đóng vai trò quan trọng trong việc giúp đỡ những người có hoàn cảnh khó khăn, như trẻ em mồ côi, người già neo đơn, hộ nghèo,... Tuy nhiên, quá trình tổ chức và quản lý các hoạt động này thường gặp nhiều khó khăn như:

* + - * Khó kiểm soát được số lượng người đóng góp, người nhận hỗ trợ.
      * Dữ liệu dễ bị thất lạc hoặc nhầm lẫn khi lưu trữ thủ công.
      * Thiếu minh bạch trong quá trình phân phối quà tặng, tiền từ thiện.
      * Khó thống kê, tổng hợp số liệu phục vụ báo cáo và quyết toán.

**Bài toán đặt ra:**

Xây dựng một phần mềm quản lý công tác từ thiện giúp các tổ chức, cá nhân thực hiện hoạt động từ thiện một cách khoa học, chính xác và hiệu quả. Phần mềm sẽ hỗ trợ lưu trữ, tìm kiếm, thống kê, báo cáo thông tin liên quan đến:

* + - * Người đóng góp: họ tên, hình thức đóng góp (tiền, hiện vật), ngày ủng hộ,...
      * Người nhận hỗ trợ: họ tên, địa chỉ, tình trạng khó khăn,...
      * Đợt từ thiện: tên đợt, thời gian tổ chức, danh sách quà tặng,...
      * Hoạt động phát quà: người nhận, phần quà, số lượng,...

**Mục tiêu của phần mềm:**

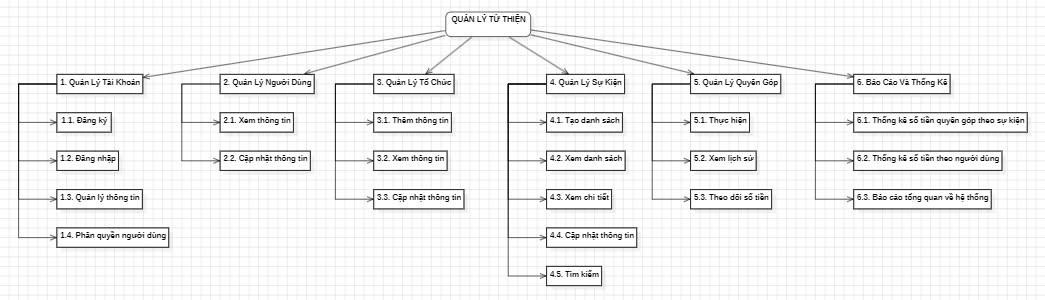
* + - * Hỗ trợ quản lý tập trung, chính xác và minh bạch.
      * Dễ sử dụng, giao diện thân thiện cho người không chuyên về CNTT.
      * Có thể mở rộng, nâng cấp để tích hợp thêm các tính năng khác như phân quyền người dùng, xuất báo cáo PDF, hoặc triển khai trên nền web/mobile.

**Đối tượng sử dụng phần mềm:**

* + - * Tổ chức từ thiện, hội nhóm tình nguyện.
      * Cơ quan đoàn thể, trường học tổ chức chương trình hỗ trợ cộng đồng.
      * Cá nhân có nhu cầu ghi nhận, tổng hợp hoạt động từ thiện.
    1. Thiết kế hệ thống

Thiết kế hệ thống là bước quan trọng nhằm xác định cấu trúc, chức năng và cách vận hành của phần mềm. Phần mềm quản lý công tác từ thiện được thiết kế theo mô hình hướng đối tượng, sử dụng cơ sở dữ liệu quan hệ và áp dụng kiến trúc phần mềm MVC để đảm bảo tính rõ ràng, dễ mở rộng và bảo trì. Phân tích chức năng

|  |  |  |
| --- | --- | --- |
| STT | CHỨC NĂNG  CHÍNH | MÔ TẢ |
| 1 | Quản lý tài  khoản | Thêm, sửa, xoá, tìm kiếm thông tin người ủng hộ (họ  tên, số tiền/quà, hình thức đóng góp, thời gian) |
| 2 | Quản lý người  dùng | Quản lý danh sách người có hoàn cảnh khó khăn cần  được hỗ trợ (họ tên, địa chỉ, tình trạng, liên hệ) |
| 3 | Quản lý tổ  chức | Quản lý thông tin và xác minh các tổ chức tham gia  hoạt động từ thiện. |
| 4 | Quản lý danh mục | Quản lý các dánh mục từ thiện |
| 5 | Quản lý sự  kiện | Tạo, cập nhật và theo dõi tiến độ các sự kiện từ thiện. |
| 6 | Quản lý quyên  Góp | Ghi nhận và kiểm soát nguồn quyên góp  tiền, hiện vật. |
| 7 | Quản lý phân phát | Ghi nhận phân phát và sử dụng nguồn tiền |
| 8 | Báo cáo và  Thống kê | Thống kê, trực quan hóa và xuất báo cáo hoạt động từ  thiện. |



Hình ảnh 3.1 Tổng quát về chức năng chính

* + 1. Thiết kế cơ sở dữ liệu
* **Mục tiêu thiết kế**

Việc thiết kế cơ sở dữ liệu giúp đảm bảo hệ thống lưu trữ, quản lý và truy xuất dữ liệu hiệu quả, nhất quán và linh hoạt. Cơ sở dữ liệu được xây dựng dựa trên các yêu cầu nghiệp vụ thu thập được từ bài toán quản lý hoạt động từ thiện. Hệ thống sử dụng cơ sở dữ liệu quan hệ (MySQL) với các bảng có mối quan hệ rõ ràng và ràng buộc khóa ngoại để đảm bảo toàn vẹn dữ liệu.

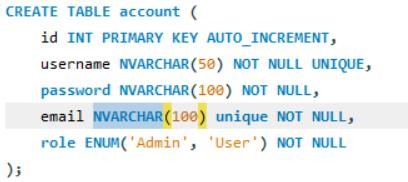
* **Các bảng chính và vai trò Bảng account:**

Tên bảng Kiểu dữ liệu Mô tả

username NVARCHAR(50) Tên đăng nhập của người dùng (khóa chính) password NVARCHAR(100) Mật khẩu người dùng

role NVARCHAR(20) Quyền truy cập (Admin / Người dùng)

email NVARCHAR(100) Email người dùng

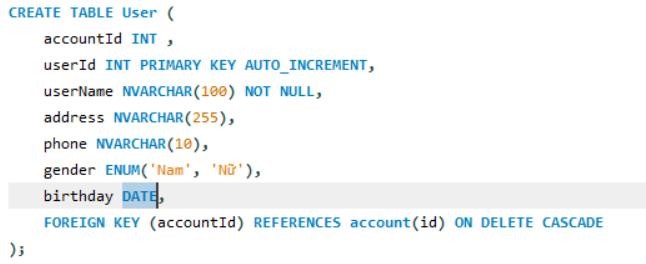


Hình ảnh 3.2 Hình ảnh về bảng account

* + **Chức năng:** Quản lý người dùng trong hệ thống và phân quyền truy cập.

**Bảng User :**

|  |  |  |
| --- | --- | --- |
| Tên bảng  userId | Kiểu dữ liệu  INT | Ý nghĩa  Mã người dùng (tự động tăng) |
| userName | NVARCHAR(100) | Họ và tên người dùng |
| address | NVARCHAR(255) | Địa chỉ người dùng |
| birthday | DATE | Ngày sinh người dùng |
| phone | VARCHAR(10) | Số điện thoại người dùng |
| gender | NVARCHAR(5) | Giới tính “Nam” hoặc “Nữ” |

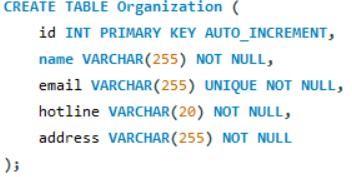
****

Hình ảnh 3.3 Hình ảnh bảng User

* + **Chức năng:** Quản lý thông tin cá nhân người dùng trong hệ thống

**Bảng Organization:**

|  |  |  |
| --- | --- | --- |
| Tên bảng  id | Kiểu dữ liệu  INT | Ý nghĩa  Mã tổ chức (tự động tăng) |
| name | VARCHAR(255) | Tên tổ chức |
| email | VARCHAR(255) | Email tổ chức |
| hotline | VARCHAR(20) | Số điện thoại tổ chức |
| address | VARCHAR(255) | Địa chỉ tổ chức |

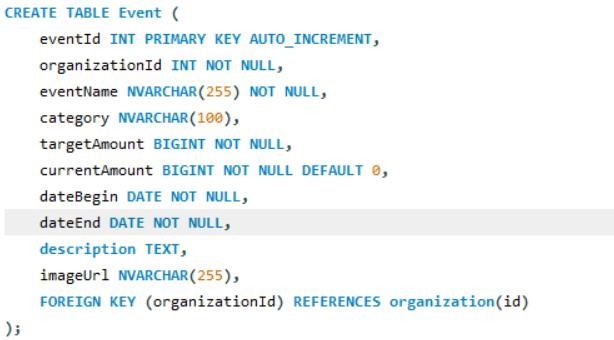
****

Hình ảnh 3.4 Hình ảnh bảng Organization

* + **Chức năng:** Quản lý thông tin của cá tổ chức từ thiện

**BBảng Event:**

|  |  |  |
| --- | --- | --- |
| Tên cột  eventId | Kiểu dữ liệu  INT | Ý nghĩa  Mã sự kiện (tự động tăng) |
| eventName | NVARCHAR(255) | Tên sự kiện |
| targetAmount | BIGINT | Số tiền mục tiêu cần quyên góp |
| currentAmount | BIGINT | Số tiền hiện có |
| dateBegin | DATE | Ngày bắt đầu sự kiện |
| dateEnd | DATE | Ngày kết thúc sự kiện |
| description | TEXT | Mô tả chi tiết về sự kiện |
| imageUrl | NVARCHAR(255) | Đường dẫn hình ảnh minh họa |
| category | NVARCHAR(100) | Chủ đề hoặc loại hình sự kiện |

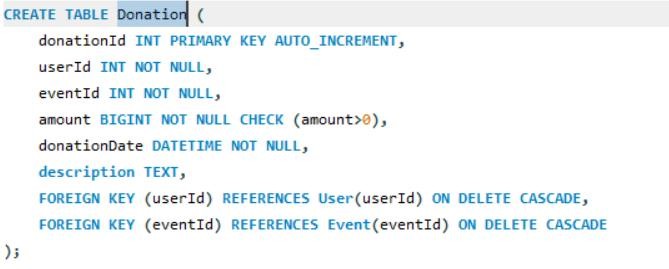


Hình ảnh 3.5 Hình ảnh bảng Event

* + **Chức năng:** Quản lý sự kiên từ thiện

**Bảng Donation:**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ý nghĩa |
| donationId | INT | Mã quyên góp (tự động tăng, khóa chính) |
| userId | INT | Mã người dùng thực hiện quyên góp |
| eventId | INT | Mã sự kiện được quyên góp cho |
| amount | BIGINT | Số tiền quyên góp (lớn hơn 0) |
| donationDate | DATETIME | Ngày giờ quyên góp |
| description | TEXT | Ghi chú, mô tả thêm về lần quyên góp |



Hình ảnh 3.5 Hình ảnh bảng Donation

* + **Chức năng:** dùng để lưu thông tin các lần quyên góp của người dùng cho các sự kiện từ thiện.

**Bảng Category:**

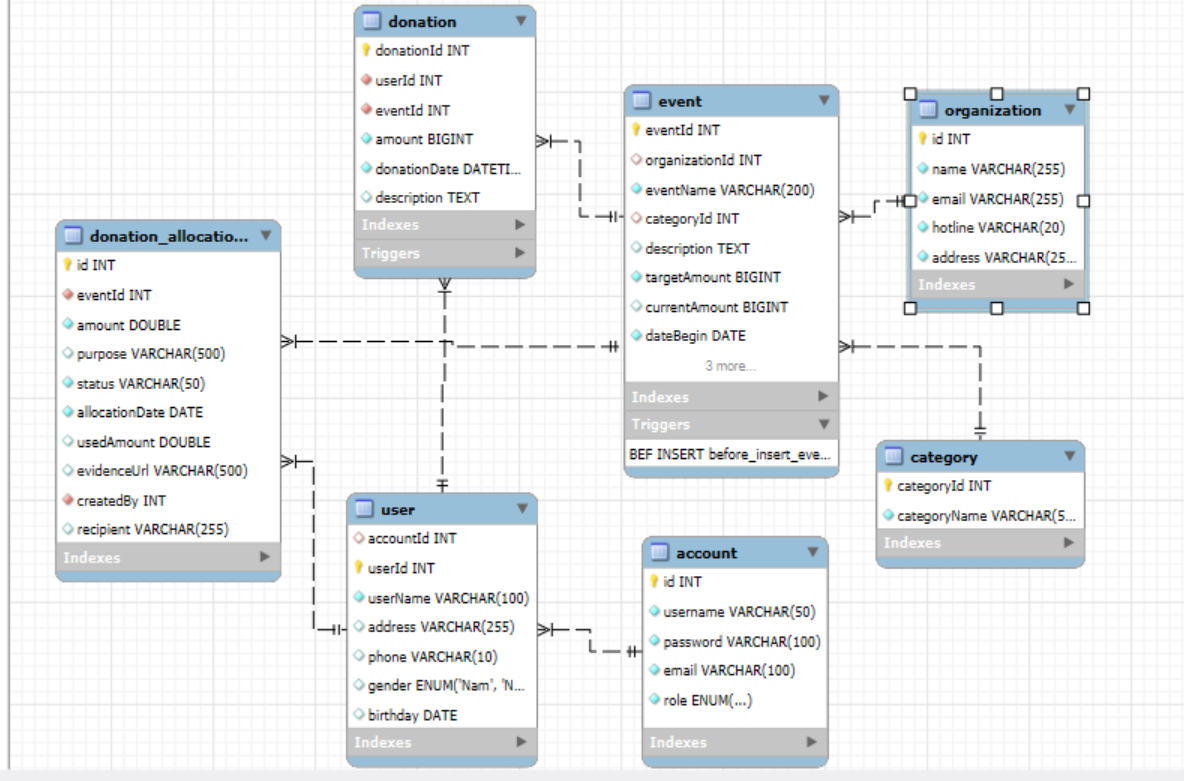
|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ý nghĩa |
| categoryId | INT | Mã danh mục |
| categoryName | VARCHAR(50) | Tên danh mục |

* **Chức năng:** dùng để lưu loại danh mục

**Bảng donation\_allocations:**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ý nghĩa** |
| id | INT | Mã phân phối (khóa chính, tự tăng) |
| eventId | INT | Mã sự kiện được phân phối (liên kết với bảng Event) |
| amount | DOUBLE | Số tiền được phân phối |
| purpose | VARCHAR(500) | Mục đích sử dụng khoản phân phối |
| status | VARCHAR(50) | Trạng thái phân phối (mặc định là 'PENDING') |
| allocationDate | DATE | Ngày thực hiện phân phối |
| usedAmount | DOUBLE | Số tiền đã sử dụng thực tế từ khoản phân phối |
| evidenceUrl | VARCHAR(500) | Đường dẫn minh chứng (hình ảnh, tài liệu…) |
| createdBy | INT | ID người tạo (liên kết với bảng User) |
| recipient | VARCHAR(255) | Người nhận hoặc tổ chức nhận khoản phân phối |

* + 1. Class diagram:



Hình ảnh 3.6 Hình ảnh class diagram

## Thiết kế cấu trúc hệ thống

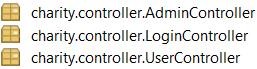
Kiến trúc Model-View-Controller (MVC) là một mẫu thiết kế phần mềm phổ biến, giúp tổ chức ứng dụng thành ba lớp logic riêng biệt, nhằm tách biệt các mối quan tâm và tăng cường khả năng bảo trì, tái sử dụng và mở rộng của mã nguồn. Ba lớp chính trong kiến trúc MVC bao gồm:

* Model: Lớp Model chịu trách nhiệm quản lý dữ liệu của ứng dụng và chứa đựng logic nghiệp vụ liên quan đến việc thao tác với dữ liệu đó. Model không trực tiếp tương tác với người dùng mà cung cấp dữ liệu và các phương thức để truy cập và thay đổi dữ liệu cho các thành phần khác của ứng dụng.
* View: Lớp View đảm nhiệm việc hiển thị thông tin cho người dùng. View nhận dữ liệu từ Model và trình bày nó theo một định dạng cụ thể mà người dùng có thể hiểu được. View cũng có thể gửi các tương tác của người dùng đến Controller để xử lý.
* Controller: Lớp Controller đóng vai trò là trung gian giữa View và Model. Khi người dùng tương tác với View, Controller sẽ nhận yêu cầu, xử lý logic điều khiển cần thiết, và sau đó chọn View phù hợp để hiển thị kết quả cho người dùng. Controller điều phối luồng thông tin giữa Model và View.

Việc áp dụng kiến trúc MVC giúp phân chia rõ ràng trách nhiệm của từng thành phần trong ứng dụng, làm cho mã nguồn dễ hiểu, dễ bảo trì và cho phép nhiều nhà phát triển làm việc độc lập trên các phần khác nhau của ứng dụng.

* + 1. Các Module hoặc Package chính

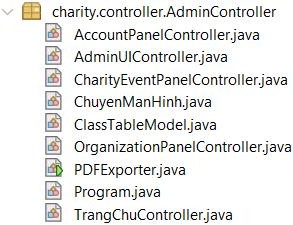
controller : Package controller đóng vai trò là trung tâm điều phối và xử lý các tương tác của người dùng với ứng dụng. Các lớp trong package này, được gọi là Controllers, nhận các yêu cầu từ lớp View (giao diện người dùng), thực hiện logic điều khiển cần thiết, tương tác với lớp Service (thuộc Model) để xử lý dữ liệu, và sau đó quyết định View nào sẽ hiển thị kết quả cho người dùng. Cấu trúc bên trong của package controller được tổ chức thành các thư mục con, cho thấy sự phân chia trách nhiệm và quản lý theo các đối tượng hoặc vai trò khác nhau trong hệ thống:



Hình ảnh 3.7 Package controller

Package charity.controller.AdminController đóng vai trò điều phối các chức năng dành riêng cho quản trị viên hệ thống trong ứng dụng quản lý công tác từ thiện. Gói này bao gồm các lớp xử lý nghiệp vụ liên quan đến quản lý tài khoản, tổ chức, sự kiện từ thiện, xuất báo cáo và điều hướng giao diện người dùng. Cụ thể:

* + - * AccountPanelController.java: Điều khiển chức năng quản lý tài khoản người dùng (xem, sửa, xoá, tìm kiếm).
      * AdminUIController.java: Điều phối toàn bộ giao diện làm việc của quản trị viên, đảm bảo các thành phần giao diện hoạt động đồng bộ.
      * CharityEventPanelController.java: Quản lý sự kiện từ thiện, cho phép thêm mới, cập nhật, xoá hoặc duyệt các sự kiện do người dùng hoặc tổ chức khởi tạo.
      * OrganizationPanelController.java: Quản lý danh sách tổ chức từ thiện, bao gồm chức năng xác minh, cập nhật thông tin tổ chức.
      * ChuyenManHinh.java: Đảm nhận vai trò điều hướng, chuyển đổi giữa các màn hình giao diện (navigation logic).
      * ClassTableModel.java: Cung cấp mô hình dữ liệu tuỳ chỉnh cho các bảng hiển thị trong giao diện (dựa trên JTable).
      * PDFExporter.java: Hỗ trợ xuất dữ liệu từ giao diện quản lý thành file PDF để phục vụ báo cáo, lưu trữ hoặc in ấn.
      * Program.java: Lớp khởi tạo chính của phần giao diện quản trị viên, có thể chứa hàm main() để chạy module độc lập.
      * TrangChuController.java: Xử lý hiển thị và tương tác tại màn hình chính (dashboard) của quản trị viên.
      * LoginController: Xử lý đăng nhập, đăng xuất và phân quyền truy cập vào hệ thống.
      * UserController: Quản lý các thao tác của người dùng thường như xem sự kiện, quyên góp, cập nhật thông tin cá nhân,...



Hình ảnh 3.8 Hình ảnh package AdminController

Package charity.controller.LoginController chịu trách nhiệm xử lý các chức năng liên quan đến xác thực người dùng trong hệ thống. Bao gồm các lớp điều khiển quá trình đăng nhập, đăng ký tài khoản và kiểm tra thông tin người dùng đầu vào. Cụ thể:

* + - * LoginController.java: Xử lý logic đăng nhập cho cả người dùng và quản trị viên, bao gồm kiểm tra thông tin tài khoản, phân quyền và điều hướng tới giao diện tương ứng sau khi đăng nhập thành công.
      * RegisterController.java: Quản lý chức năng đăng ký tài khoản mới, bao gồm kiểm tra dữ liệu đầu vào (họ tên, email, mật khẩu...), lưu dữ liệu vào cơ sở dữ liệu và hiển thị thông báo tương ứng.

Package UserController chịu trách nhiệm xử lý các chức năng liên quan đến người dùng trong hệ thống. Bao gồm các lớp điều khiển giao diện và chức năng quản lý thông tin người dùng, danh sách quyên góp, tổ chức từ thiện, và bảng dữ liệu. Cụ thể:

* + - * ClassTableModel.java:Quản lý mô hình dữ liệu cho bảng hiển thị, định nghĩa cấu trúc và dữ liệu hiển thị trong bảng giao diện người dùng.
      * DonationDialogController.java:Điều khiển hộp thoại nhập liệu quyên góp, xử lý các sự kiện khi người dùng nhập và xác nhận quyên góp.
      * DonationListController.java:Xử lý việc hiển thị danh sách các khoản quyên góp mà người dùng đã thực hiện, bao gồm lọc, tìm kiếm và sắp xếp.
      * InforPanelController.java:Điều khiển giao diện hiển thị thông tin người dùng, cập nhật thông tin cá nhân hoặc xem chi tiết.
      * MainPanelController.java:Quản lý giao diện chính của người dùng sau khi đăng nhập, phối hợp các panel con như danh sách quyên góp, thông tin cá nhân, v.v.
      * MyDonationController.java:Xử lý logic và giao diện liên quan đến những khoản quyên góp do người dùng thực hiện, cho phép chỉnh sửa hoặc hủy quyên góp.
      * OrganizationPanelController.java:Điều khiển phần hiển thị danh sách các tổ chức từ thiện, cho phép người dùng tìm kiếm và chọn nơi quyên góp.
      * UserUIController.java:Điều khiển tổng thể giao diện người dùng cuối (UI), khởi tạo và liên kết các thành phần giao diện liên quan đến người dùng.



Hỉnh ảnh 3.9 Hình ảnh package LoginController

Package icon chịu trách nhiệm quản lý các tài nguyên biểu tượng (icons) dùng trong giao diện của hệ thống. Bao gồm các tệp hình ảnh nhỏ như biểu tượng nút, menu, hoặc trạng thái (ví dụ: biểu tượng thêm, sửa, xóa, đăng xuất...).

* + - * Được sử dụng để làm cho giao diện người dùng trực quan và sinh động hơn.
      * Các icon này có thể được tải từ file .png, .svg, hoặc định dạng khác và gắn vào các thành phần Swing/JavaFX như JButton, JLabel.

Package image chịu trách nhiệm lưu trữ và quản lý các hình ảnh lớn hơn (không phải icon), chẳng hạn như:

* + - * Logo tổ chức từ thiện.
      * Hình nền giao diện.
      * Ảnh đại diện người dùng.
      * Hình ảnh minh họa cho chiến dịch quyên góp.



Hình ảnh 3.10 Hình ảnh package icon và image

Package repository chịu trách nhiệm quản lý việc truy xuất và thao tác dữ liệu từ cơ sở dữ liệu. Đây là tầng kết nối giữa logic nghiệp vụ và dữ liệu, thực hiện các truy vấn, thêm, sửa, xóa dữ liệu từ các bảng liên quan trong hệ thống. Cụ thể:

* + - * AccountRepository.java:Xử lý các thao tác liên quan đến tài khoản người dùng như đăng nhập, kiểm tra tài khoản tồn tại, và cập nhật thông tin tài khoản.
      * AssignedEventRepository.java:Quản lý các sự kiện từ thiện đã được phân công cho người dùng, bao gồm việc gán, hủy và truy xuất thông tin sự kiện đã phân công.
      * CharityEventRepository.java:Thực hiện các thao tác với bảng sự kiện từ thiện: thêm mới, cập nhật, xóa, tìm kiếm sự kiện theo điều kiện cụ thể.
      * ConnectionDB.java:Quản lý kết nối đến cơ sở dữ liệu. Cung cấp phương thức mở và đóng kết nối dùng cho các repository khác.
      * DonationRepository.java:Xử lý các thao tác với dữ liệu quyên góp như ghi nhận, cập nhật, truy xuất thông tin các khoản quyên góp từ người dùng.
      * OrganizationRepository.java:Quản lý dữ liệu liên quan đến tổ chức từ thiện, bao gồm thêm tổ chức mới, cập nhật thông tin và lấy danh sách tổ chức.
      * UserRepository.java:Thao tác với thông tin người dùng như tạo tài khoản, cập nhật hồ sơ cá nhân, truy xuất người dùng theo ID hoặc điều kiện cụ thể.



Hình ảnh 3.11 Hình ảnh package repository



Hình ảnh 3.12 Hình ảnh package IRepository

Package IRepository định nghĩa các giao diện (interface) cho tầng repository. Các giao diện này đóng vai trò quy định các phương thức chuẩn mà các lớp repository cụ thể (trong charity.repository) cần triển khai. Điều này giúp tăng tính mở rộng, linh hoạt và dễ kiểm thử trong thiết kế hệ thống. Cụ thể:

* + - * IAccountRepository.java:Giao diện khai báo các phương thức thao tác với tài khoản người dùng như: đăng nhập, kiểm tra tài khoản, cập nhật thông tin.
      * IAssignedEventRepository.java:Xác định các phương thức thao tác với các sự kiện đã được phân công cho người dùng, chẳng hạn như thêm, xóa, truy xuất.
      * ICharityEventRepository.java:Giao diện định nghĩa các hành động cần thiết đối với dữ liệu sự kiện từ thiện như thêm mới, chỉnh sửa, hoặc lấy danh sách.
      * IDonationRepository.java:Giao diện chuẩn hóa các thao tác như thêm quyên góp, sửa thông tin quyên góp, lấy lịch sử quyên góp của người dùng.
      * IOrganizationRepository.java:Xác định các phương thức thao tác với thông tin tổ chức từ thiện, đảm bảo lớp triển khai tuân thủ đúng các hành vi cần có.
      * IUserRepository.java:Giao diện định nghĩa các phương thức liên quan đến dữ liệu người dùng, như tìm kiếm, thêm mới hoặc cập nhật hồ sơ.

Package service chịu trách nhiệm xử lý logic nghiệp vụ (business logic) của hệ thống. Đây là lớp trung gian giữa tầng controller (giao diện/người dùng) và tầng repository (dữ liệu), giúp tách biệt các quy trình xử lý dữ liệu ra khỏi giao diện và truy xuất dữ liệu. Cụ thể:

* + - * AccountService.java:Xử lý các chức năng nghiệp vụ liên quan đến tài khoản người dùng, như xác thực đăng nhập, thay đổi mật khẩu, hoặc xử lý lỗi đăng nhập sai.
      * CharityEventService.java:Chịu trách nhiệm xử lý logic nghiệp vụ cho các sự kiện từ thiện như đăng ký tham gia, kiểm tra trạng thái sự kiện, hoặc xác minh điều kiện tham gia.
      * DonationService.java:Xử lý các quy trình quyên góp như tạo mới khoản quyên góp, xác thực dữ liệu quyên góp, cập nhật tổng số tiền quyên góp, hoặc thống kê quyên góp theo người dùng.
      * OrganizationService.java:Quản lý logic nghiệp vụ liên quan đến các tổ chức từ thiện, như kiểm tra tính hợp lệ của tổ chức, liên kết tổ chức với sự kiện, hoặc xử lý tìm kiếm nâng cao.
      * UserService.java:Xử lý các nghiệp vụ liên quan đến thông tin người dùng, bao gồm đăng ký, cập nhật hồ sơ, hoặc kiểm tra quyền truy cập cho từng loại người dùng.



Hình ảnh 3.13 Hình ảnh package service

Package IService định nghĩa các giao diện (interface) cho tầng service, nhằm quy định các hành vi và chức năng cần thiết mà các lớp service cụ thể trong charity.service phải triển khai. Mục tiêu là đảm bảo tính đóng gói, dễ bảo trì và hỗ trợ kiểm thử unit test hiệu quả hơn. Cụ thể:

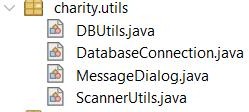
* + - * IAccountService.java:Giao diện khai báo các phương thức xử lý nghiệp vụ liên quan đến tài khoản, ví dụ: đăng nhập, xác thực, đổi mật khẩu.
      * ICharityEventService.java:Quy định các phương thức xử lý liên quan đến sự kiện từ thiện, như tạo sự kiện, đăng ký tham gia, hoặc kiểm tra trạng thái.
      * IDonationService.java:Định nghĩa các hành động xử lý dữ liệu quyên góp như tạo mới, thống kê, kiểm tra số tiền, hoặc lịch sử quyên góp.
      * IOrganizationService.java:Giao diện mô tả các chức năng nghiệp vụ liên quan đến tổ chức từ thiện, như tìm kiếm, xác thực và liên kết sự kiện.
      * IUserService.java:Quy định các hành vi cần có trong xử lý người dùng như đăng ký, cập nhật thông tin cá nhân, phân quyền truy cập...



Hình ảnh 3.14 Hình ảnh package Iservice

Package utils chứa các lớp tiện ích (utility classes) hỗ trợ các thao tác chung, tái sử dụng trong nhiều phần khác nhau của hệ thống. Các lớp này thường không chứa logic nghiệp vụ cụ thể mà cung cấp các phương thức tĩnh dùng để xử lý dữ liệu, hiển thị thông báo, kết nối cơ sở dữ liệu, v.v. Cụ thể:

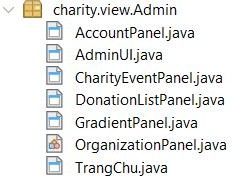
* + - * DBUtils.java:Cung cấp các phương thức tiện ích để thao tác với cơ sở dữ liệu, như đóng kết nối, xử lý truy vấn an toàn, hoặc log lỗi truy vấn.
      * DatabaseConnection.java:Chịu trách nhiệm khởi tạo và quản lý kết nối tới cơ sở dữ liệu, đảm bảo quá trình kết nối được thực hiện hiệu quả và bảo mật.
      * MessageDialog.java:Hỗ trợ hiển thị các hộp thoại thông báo (message boxes) như cảnh báo, lỗi, xác nhận... dùng trong giao diện người dùng (UI), thường sử dụng JOptionPane.
      * ScannerUtils.java:Cung cấp phương thức tiện lợi để nhập dữ liệu từ bàn phím (console) với khả năng kiểm tra định dạng và xử lý lỗi nhập liệu.



Hình ảnh 3.15 Hình ảnh package utils

Package Admin thuộc tầng giao diện (view) dành riêng cho người quản trị hệ thống. Các lớp trong package này chịu trách nhiệm hiển thị và xử lý giao diện đồ họa người dùng (GUI) liên quan đến quản lý tài khoản, sự kiện từ thiện, danh sách quyên góp, tổ chức, v.v. Cụ thể:

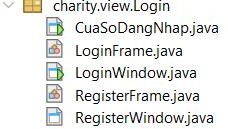
* + - * AccountPanel.java:Hiển thị danh sách tài khoản và hỗ trợ thao tác như tạo, sửa, xoá tài khoản người dùng trong giao diện admin.
      * AdminUI.java:Lớp giao diện chính dành cho quản trị viên, làm trung tâm điều hướng tới các panel quản lý khác như tài khoản, tổ chức, sự kiện...
      * CharityEventPanel.java:Hiển thị và quản lý danh sách các sự kiện từ thiện trong hệ thống. Cho phép thêm mới, cập nhật hoặc xoá sự kiện.
      * DonationListPanel.java:Cung cấp giao diện để xem danh sách các khoản quyên góp, tra cứu theo người dùng, thời gian hoặc sự kiện cụ thể.
      * GradientPanel.java:Lớp hỗ trợ hiệu ứng nền gradient trong giao diện người dùng. Mang tính chất thẩm mỹ và hỗ trợ thiết kế UI.
      * OrganizationPanel.java:Quản lý thông tin các tổ chức từ thiện, bao gồm xem, thêm mới, chỉnh sửa và xoá tổ chức khỏi hệ thống.
      * TrangChu.java:Lớp hiển thị trang chủ giao diện admin với các thông tin tổng quan, thống kê hoặc bảng điều khiển chính.



Hình ảnh 3.16 Hình ảnh package view.Admin

Package Login thuộc tầng giao diện (view) và tập trung vào việc xây dựng các cửa sổ và khung giao diện phục vụ quá trình xác thực người dùng như đăng nhập và đăng ký. Các lớp trong package này hỗ trợ cả người dùng và quản trị viên với các cửa sổ thân thiện và trực quan. Cụ thể:

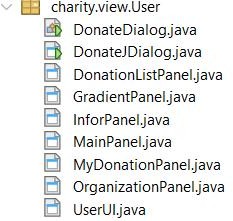
* + - * CuaSoDangNhap.java:Lớp giao diện đăng nhập bằng tiếng Việt, cung cấp các thành phần như ô nhập tên đăng nhập, mật khẩu và nút đăng nhập.
      * LoginFrame.java:Một khung (JFrame) dùng để hiển thị giao diện đăng nhập cho hệ thống, có thể dùng cho cả người dùng và admin.
      * LoginWindow.java:Tương tự như LoginFrame, nhưng có thể được triển khai với kiểu hiển thị khác (ví dụ: JDialog hoặc một giao diện nhỏ hơn), phục vụ mục đích tương tự là đăng nhập.
      * RegisterFrame.java:Khung (JFrame) cho phép người dùng tạo tài khoản mới, bao gồm nhập họ tên, email, mật khẩu và xác nhận thông tin.
      * RegisterWindow.java:Một phiên bản khác hoặc giao diện thay thế cho RegisterFrame, có thể thiết kế lại với bố cục hoặc mục đích sử dụng khác (ví dụ: pop-up nhỏ, dialog đăng ký nhanh).



Hình ảnh 3.17 Hình ảnh package view.Login

Package User thuộc tầng giao diện (view) và phục vụ người dùng cuối khi tham gia các hoạt động như quyên góp, xem thông tin, quản lý tài khoản hoặc sự kiện. Các lớp trong package này cung cấp đầy đủ các panel và dialog để người dùng tương tác thuận tiện với hệ thống. Cụ thể:

* + - * DonateDialog.java:Hộp thoại cho phép người dùng thực hiện quyên góp, nhập thông tin như số tiền, hình thức, và nội dung quyên góp.
      * DonateJDialog.java:Phiên bản khác của hộp thoại quyên góp, có thể dùng JDialog thay vì JFrame để tích hợp mượt mà vào giao diện hiện tại.
      * DonationListPanel.java:Hiển thị danh sách các khoản quyên góp mà người dùng đã thực hiện hoặc đang theo dõi.
      * GradientPanel.java:Lớp hỗ trợ hiệu ứng nền chuyển màu (gradient) trong các giao diện người dùng, nhằm tăng tính thẩm mỹ.
      * InforPanel.java:Panel hiển thị thông tin cá nhân của người dùng như tên, email, vai trò, hoặc trạng thái tài khoản.
      * MainPanel.java:Giao diện chính sau khi người dùng đăng nhập thành công, làm trung tâm điều hướng đến các chức năng khác.
      * MyDonationPanel.java:Cho phép người dùng xem lại các lần quyên góp cá nhân đã thực hiện và tình trạng xử lý của từng lần.
      * OrganizationPanel.java:Hiển thị danh sách các tổ chức từ thiện mà người dùng có thể chọn để quyên góp hoặc xem thêm thông tin.
      * UserUI.java:Giao diện người dùng tổng hợp dành cho người dùng cuối, tích hợp các panel như quyên góp, thông tin cá nhân, tổ chức,...



Hình ảnh 3.18 Hình ảnh package view.User

Bean thường dùng để chứa các lớp Java Bean — đại diện cho các thực thể (entities) đơn giản, giúp truyền dữ liệu giữa các tầng của ứng dụng. Trong hệ thống này, charity.bean chứa lớp:

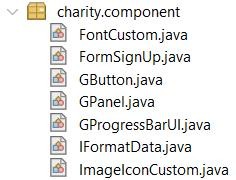
* + - * DanhMuc.java:Lớp đại diện cho một danh mục (category) trong hệ thống. Có thể bao gồm các thuộc tính như mã danh mục, tên, biểu tượng, hoặc liên kết đến chức năng cụ thể (dùng để điều hướng trong giao diện). Thường được sử dụng để cấu hình menu hoặc phân loại nội dung.



Hình ảnh 3.19 Hình ảnh package bean

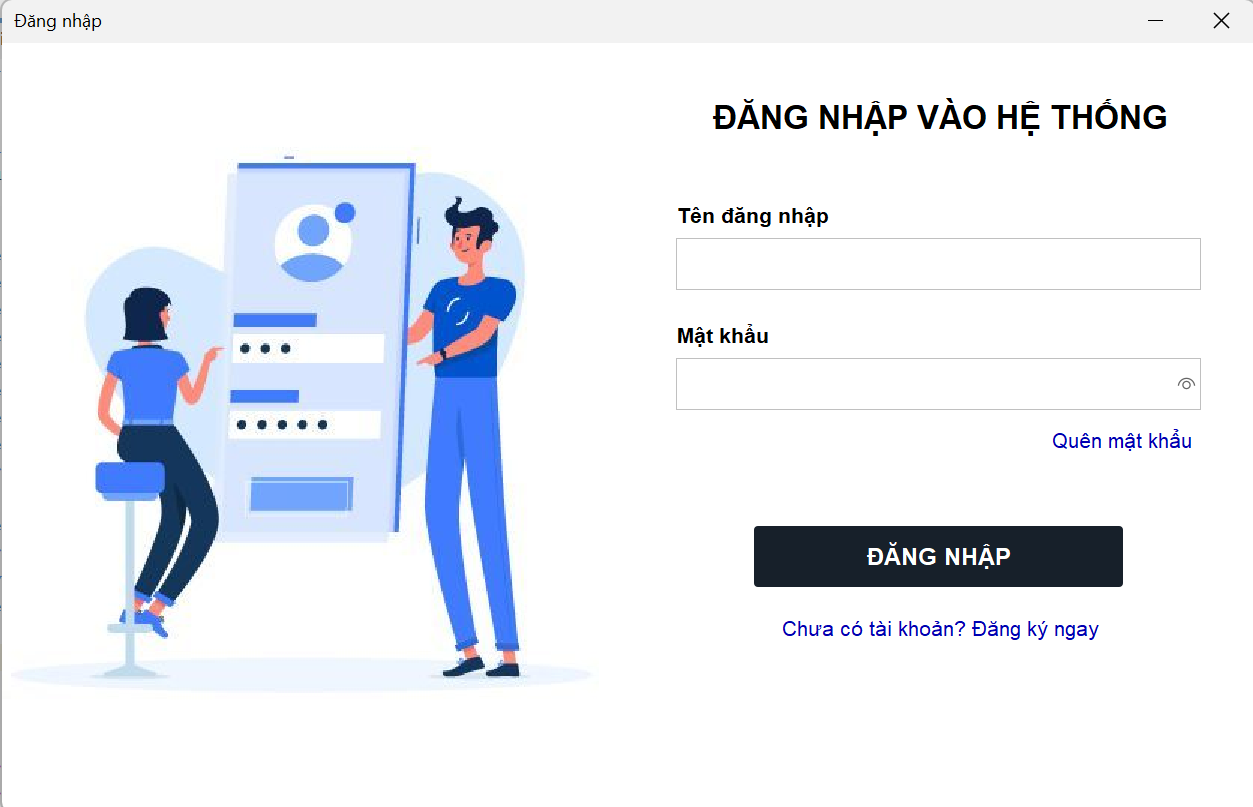
Package component chứa các thành phần giao diện tùy chỉnh được tái sử dụng trong toàn bộ ứng dụng. Những lớp này mở rộng hoặc thay đổi hành vi mặc định của Swing nhằm tạo ra trải nghiệm người dùng nhất quán và thân thiện hơn. Cụ thể gồm:

* + - * FontCustom.java: Quản lý và áp dụng font chữ tùy chỉnh cho các thành phần UI nhằm đảm bảo tính thẩm mỹ và đồng bộ.
      * FormSignUp.java: Biểu mẫu giao diện người dùng để đăng ký tài khoản, có thể bao gồm các trường họ tên, email, mật khẩu,...
      * GButton.java: Lớp nút tùy chỉnh kế thừa JButton, được thiết kế lại giao diện như màu nền, viền, hiệu ứng hover.
      * GPanel.java: Lớp panel tùy chỉnh dùng để thay thế JPanel, có thể tích hợp hiệu ứng đổ bóng hoặc gradient.
      * GProgressBarUI.java: Giao diện tùy chỉnh cho thanh tiến trình (JProgressBar), cải thiện trải nghiệm thị giác.
      * IFormatData.java: Giao diện hoặc lớp hỗ trợ định dạng dữ liệu đầu ra (ví dụ: ngày tháng, số liệu).
      * ImageIconCustom.java: Quản lý biểu tượng hình ảnh được hiển thị trong giao diện, hỗ trợ nạp và xử lý ảnh linh hoạt hơn.



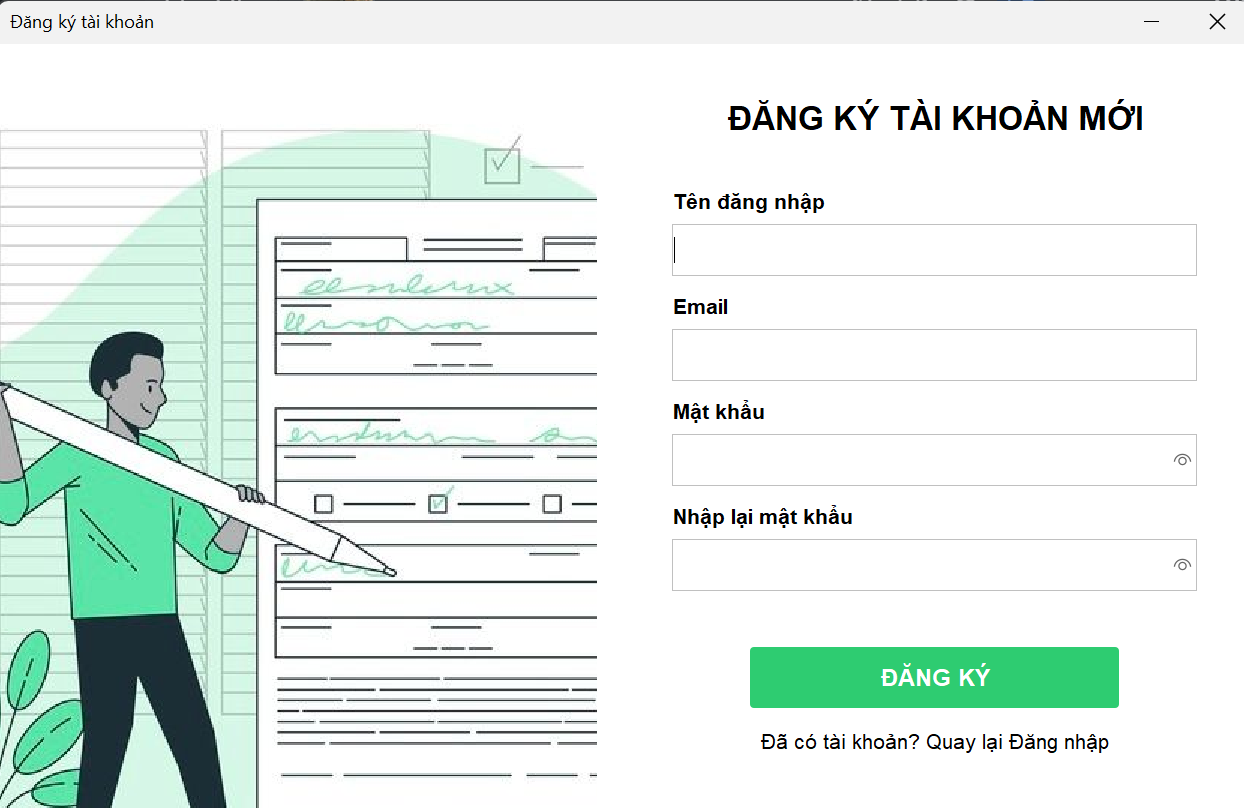
Hình ảnh 3.20 Hình ảnh package component

## Thiết kế giao diện

* + 1. Giao điện đăng nhập người dùng
* Bố cục tổng thể: Giao diện đăng nhập được thiết kế với bố cục gọn gàng, căn giữa, tập trung vào các trường nhập liệu chính. Nền giao diện là một màu xanh lam nhạt, tạo cảm giác chuyên nghiệp và dễ chịu. Cửa sổ ứng dụng hiển thị biểu tượng Java Swing đặc trưng, cho thấy đây là một ứng dụng desktop được xây dựng bằng công nghệ Java.
* Các thành phần trong giao diện: Tiêu đề "Đăng nhập", nhãn và ô nhập Tên đăng nhập, nhãn và ô nhập Mật khẩu, nút "Đăng nhập", nút "Thoát".
* **Liên kết đăng ký:**
  + Văn bản: "Bạn chưa có tài khoản?"
  + Được căn giữa phía dưới cùng của giao diện.
  + Có thể được lập trình để chuyển đến giao diện đăng ký tài khoản mới.
  + Luồng điều hướng (User Flow)
    - Người dùng mở ứng dụng và thấy giao diện đăng nhập.
    - Nhập tên đăng nhập và mật khẩu vào các trường tương ứng.
    - Nhấn nút “Đăng nhập”:
      * Nếu thông tin hợp lệ → chuyển sang giao diện chính (dashboard).
      * Nếu thông tin sai → hiển thị thông báo lỗi (giao diện hiện tại chưa hiển thị lỗi).
    - Nếu chưa có tài khoản, người dùng nhấn vào liên kết "Bạn chưa có tài khoản?" để chuyển sang giao diện đăng ký.
    - Nếu quyên mật khẩu , người dùng nhấn vào “Quên mật khẩu”, một khung quên mật khẩu sẽ xuất hiện. Khi nhập username và email nguời dùng sẽ được cấp một mật khẩu mới .
    - 

Hình ảnh 3.21 Hình ảnh giao diện đăng nhập

* + 1. Giao điện đăng ký
       - Bố cục tổng thể:Giao diện đăng ký được thiết kế với bố cục dọc, căn giữa các thành phần theo chiều ngang. Nền gradient chuyển từ xanh dương nhạt sang xanh ngọc, tương tự giao diện đăng nhập, tạo sự đồng bộ và dễ chịu cho người dùng. Cửa sổ có biểu tượng Java, cho thấy khả năng được xây dựng bằng Java Swing.
       - Các thành phần trong giao diện:Tiêu đề "Đăng ký", nhãn và ô nhập Tên đăng nhập, nhãn và ô nhập Email, nhãn và ô nhập Mật khẩu, nhãn và ô nhập Xác nhận mật khẩu, nút "Đăng ký", liên kết điều hướng "Bạn đã có tài khoản?".
       - Luồng điều hướng (User Flow):Luồng điều hướng (Đăng ký): Người dùng nhập Tên đăng nhập, Email, Mật khẩu, Xác nhận mật khẩu → nhấn nút "Đăng ký" → nếu hợp lệ thì tạo tài khoản và chuyển sang đăng nhập, nếu lỗi thì hiển thị thông báo → người dùng cũng có thể nhấn "Bạn đã có tài khoản?" để quay lại màn hình đăng nhập.



Hình ảnh 3.22 Hình ảnh giao diện đăng ký

* + 1. Giao điện người dùng

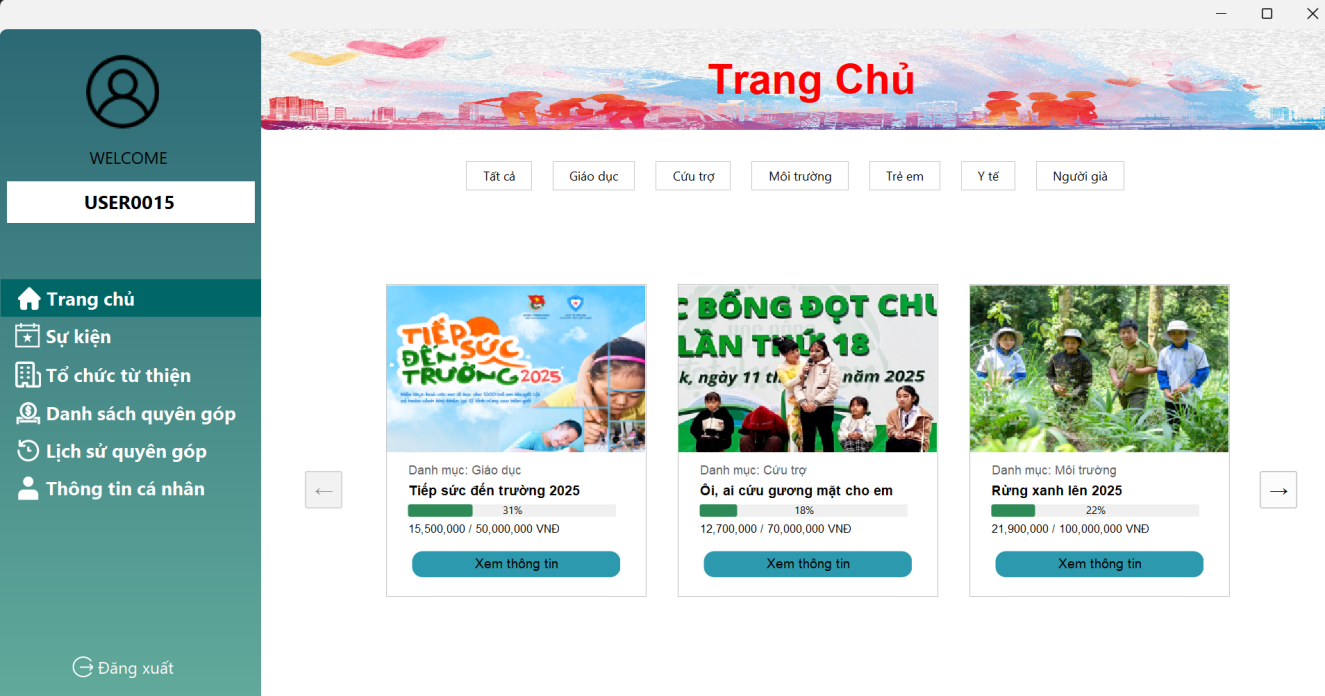
Phần nội dung bên phải của trang chủ hiển thị các sự kiện từ thiện dưới dạng trực quan và dễ theo dõi. Ở phía trên là tiêu đề nổi bật “Trang Chủ” được đặt giữa cùng với hình nền màu sắc sinh động, tạo điểm nhấn cho giao diện.

Ngay bên dưới là **hệ thống bộ lọc theo danh mục** bao gồm các nút: “Tất cả”, “Giáo dục”, “Cứu trợ”, “Môi trường”, “Trẻ em”, “Y tế”, “Người già”. Khi người dùng nhấn vào một nút, danh sách sự kiện bên dưới sẽ tự động thay đổi để chỉ hiển thị các sự kiện thuộc danh mục được chọn.

Phần trung tâm là **danh sách các sự kiện**, được trình bày dạng **thẻ ngang** gồm:

* Ảnh minh họa đại diện cho từng sự kiện.
* Thông tin danh mục.
* Tên sự kiện nổi bật, in đậm.
* Thanh tiến độ hiển thị % số tiền đã quyên góp được so với mục tiêu.
* Dòng hiển thị số tiền hiện tại / số tiền mục tiêu.
* Nút “Xem thông tin” để người dùng truy cập chi tiết sự kiện.

Ngoài ra, hệ thống còn có **nút điều hướng trái/phải** giúp người dùng chuyển qua các sự kiện khác nếu danh sách dài. Tất cả bố cục đều được sắp xếp hợp lý, thân thiện với người dùng và đảm bảo dễ dàng theo dõi tiến độ quyên góp của từng sự kiện.



Hình ảnh 3.23 Hình ảnh giao diện người dùng ở Trang chủ

* + 1. Giao diện người dùng ở trang sự kiện

Khu vực nội dung bên phải hiển thị danh sách các sự kiện từ thiện đang được tổ chức, với bố cục trực quan và khoa học. Ở phía trên là tiêu đề “Sự kiện” nổi bật, giúp người dùng dễ dàng nhận biết chức năng hiện tại.

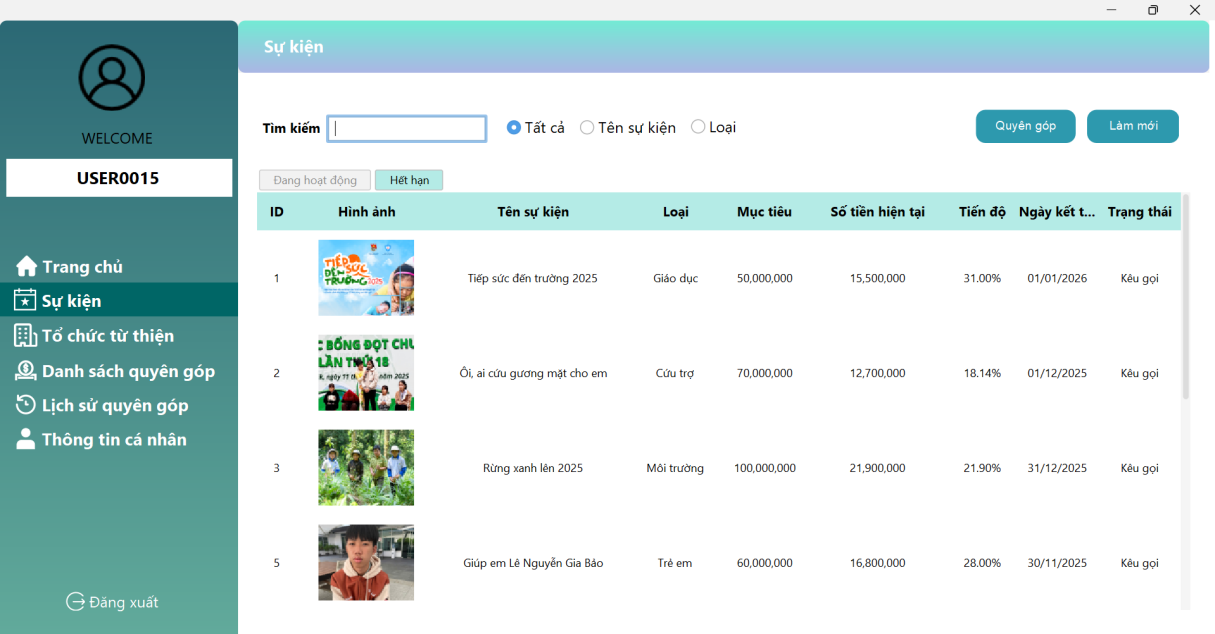
Phía dưới tiêu đề là **thanh tìm kiếm** cùng với bộ lọc tùy chọn theo tiêu chí: **Tất cả**, **Tên sự kiện**, **Loại**. Tùy theo lựa chọn, người dùng có thể nhập từ khóa để tìm kiếm sự kiện theo tên hoặc theo phân loại.

Tiếp theo là hai tab chức năng:

* **Đang hoạt động**: hiển thị các sự kiện còn thời gian quyên góp.
* **Hết hạn**: hiển thị các sự kiện đã kết thúc.

Bên cạnh là hai nút thao tác:

* **Quyên góp**: mở giao diện quyên góp cho sự kiện đang chọn.
* **Làm mới**: tải lại dữ liệu sự kiện mới nhất.

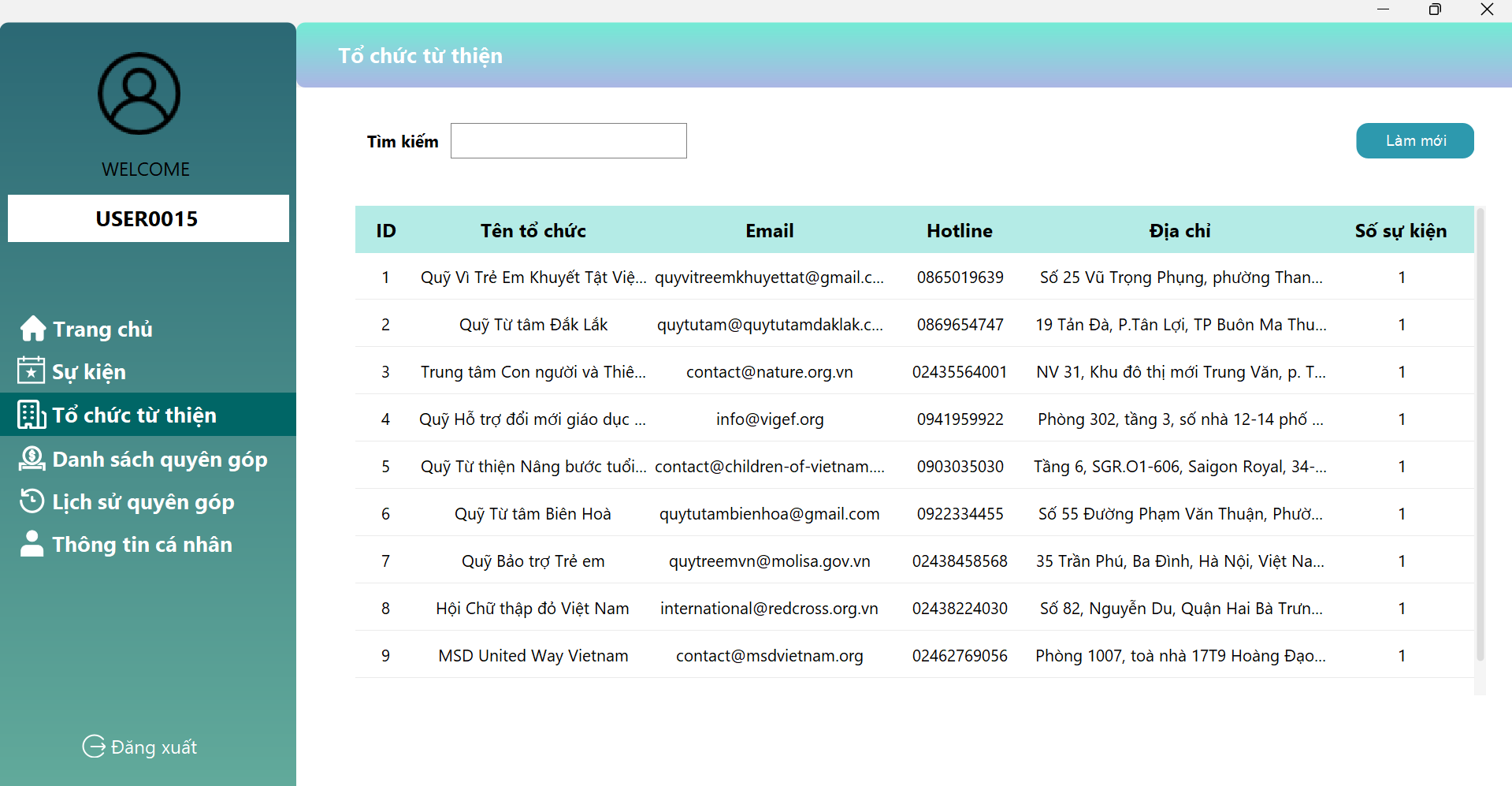


Hình ảnh 3.24 Hình ảnh giao diện người dùng ở trang sự kiện

* + 1. Giao điện người dùng ở trang tổ chức
       - Bố cục tổng thể: Giao diện chia làm hai phần: thanh điều hướng bên trái và phần hiển thị nội dung bên phải với tiêu đề, thanh tìm kiếm, nút làm mới và bảng dữ liệu các tổ chức từ thiện.

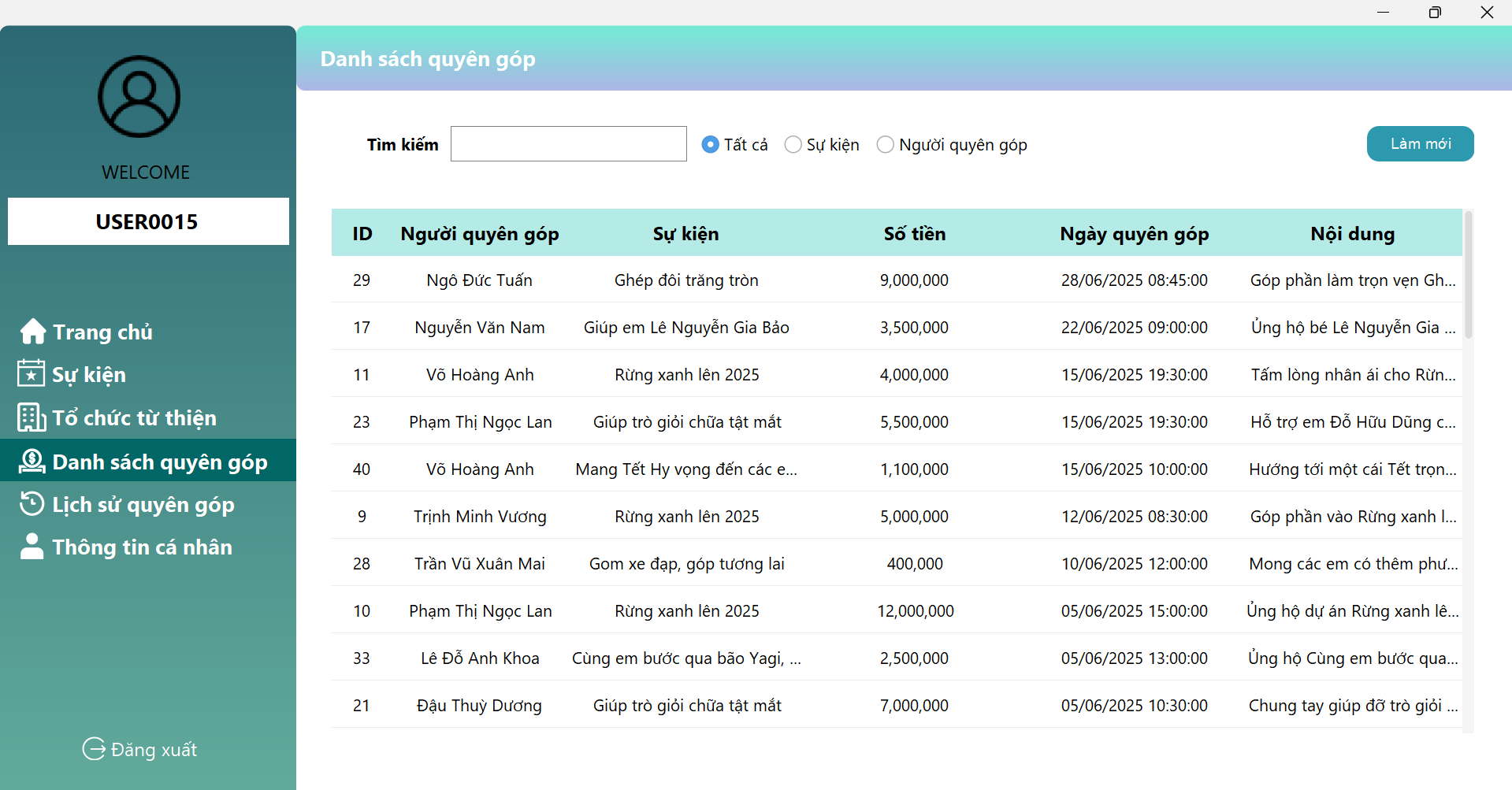
Các thành phần giao diện: Thanh điều hướng (ảnh đại diện, tên người dùng, menu chức năng), tiêu đề “Tổ chức từ thiện”, ô tìm kiếm, nút “Làm mới”, bảng thông tin gồm các cột: ID, Tên tổ chức, Email, Hotline, Địa chỉ, cùng thanh cuộn dọc.

* + - * Luồng điều hướng:Người dùng chọn mục “Tổ chức từ thiện” trong thanh menu → xem danh sách tổ chức → nhập từ khóa vào ô tìm kiếm để lọc → bấm “Làm mới” để tải lại danh sách.



Hình ảnh 3.25 Giao diện người dùng ở tổ chức sự kiên

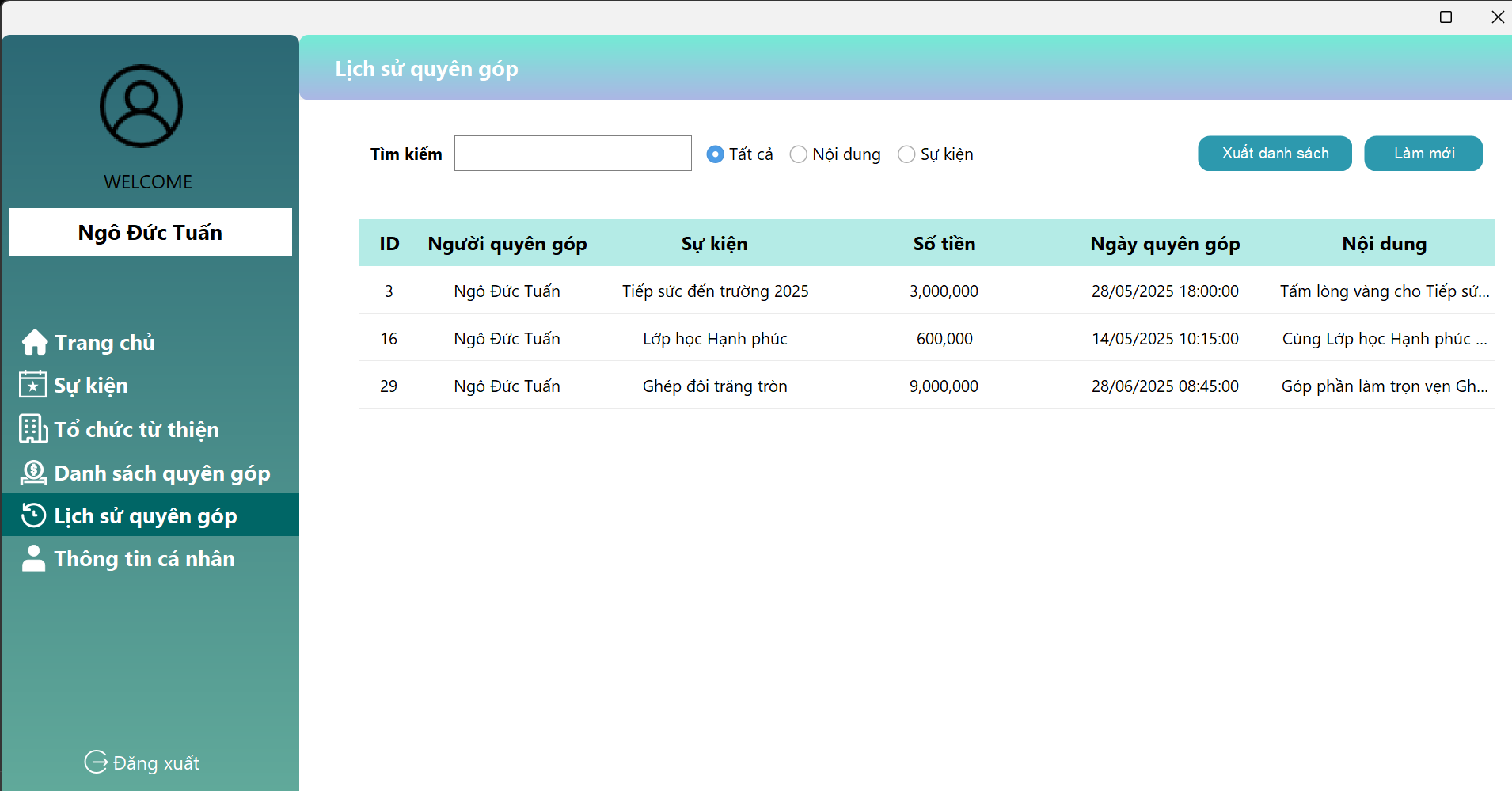
* + 1. Giao điện người dùng ở danh sách quyên góp
       - Bố cục tổng thể: Giao diện gồm thanh điều hướng bên trái và bảng dữ liệu chi tiết ở bên phải, có tiêu đề, công cụ tìm kiếm, nút “Làm mới”, và bảng danh sách quyên góp.
       - Các thành phần giao diện:Thanh menu bên trái, tiêu đề “Danh sách quyên góp”, ô tìm kiếm, nhóm nút radio chọn tiêu chí tìm kiếm (ID, Sự kiện, Người quyên góp), nút “Làm mới”, bảng thông tin quyên góp gồm ID, người quyên góp, sự kiện, số tiền, ngày quyên góp, và nội dung.
       - Luồng điều hướng:Người dùng chọn mục “Danh sách quyên góp” trong thanh menu → nhập từ khóa → chọn tiêu chí tìm kiếm → bấm “Làm mới” để làm sạch dữ liệu → theo dõi danh sách cập nhật theo tiêu chí lọc.



Hình ảnh 3.26 Giao diện người dùng ở danh sách quyên góp

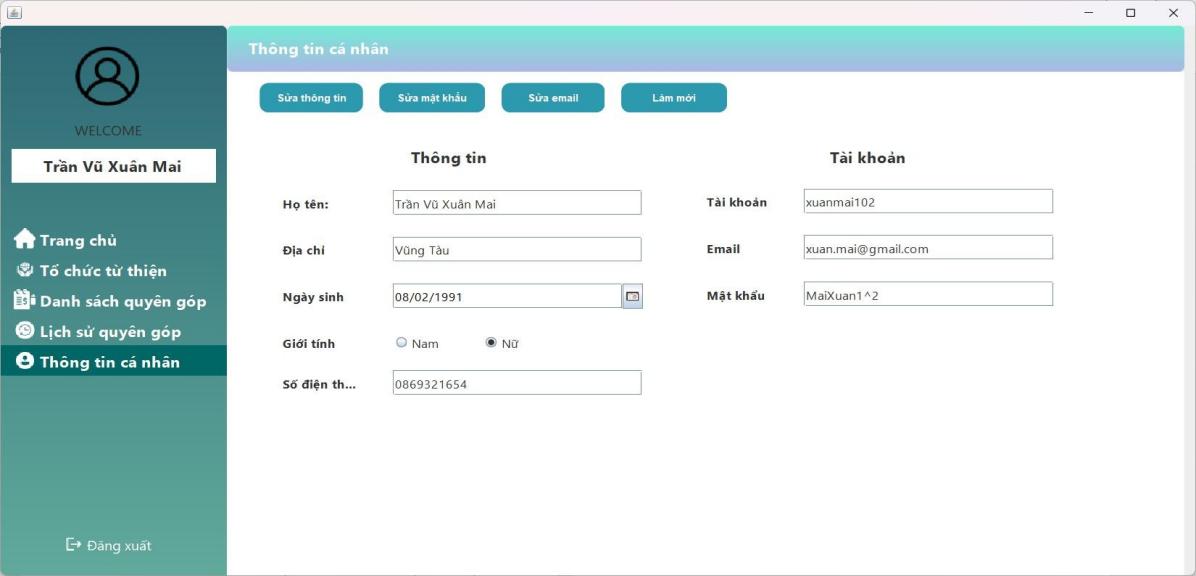
* + 1. Giao điện người dùng ở lịch sử quyên góp
       - Bố cục tổng thể: Giao diện bao gồm thanh điều hướng bên trái và khu vực hiển thị lịch sử quyên góp bên phải với thanh tìm kiếm, bộ lọc tìm kiếm, nút xuất danh sách và bảng liệt kê các khoản quyên góp đã thực hiện.
       - Các thành phần giao diện: Gồm tiêu đề “Lịch sử quyên góp”, ô tìm kiếm, nhóm nút radio để lọc theo ID, sự kiện hoặc nội dung, nút “Làm mới”, nút “Xuất danh sách”, bảng thông tin với các cột: ID, người quyên góp, sự kiện, số tiền, ngày quyên góp, nội dung.
       - Luồng điều hướng: Người dùng nhấn “Lịch sử quyên góp” từ menu trái

→ nhập từ khóa vào ô tìm kiếm → chọn tiêu chí tìm kiếm → có thể bấm “Làm mới” để xóa lọc hoặc “Xuất danh sách” để tải dữ liệu → xem danh sách các lần quyên góp cá nhân.



Hình ảnh 3.27 Giao diện người dùng ở lịch sử quyên góp

* + 1. Giao điện người dùng ở Thông tin cá nhân
       - Bố cục tổng thể:Giao diện chia làm hai phần chính: thông tin cá nhân bên trái và thông tin tài khoản bên phải. Phía trên là các nút chức năng như sửa thông tin, đổi mật khẩu, sửa email và làm mới.
       - Các thành phần giao diện:Người dùng có thể xem và chỉnh sửa: họ tên, địa chỉ, ngày sinh (có chọn lịch), giới tính, số điện thoại, tên tài khoản, email và mật khẩu. Các ô nhập liệu cho phép chỉnh sửa sau khi nhấn các nút tương ứng.
       - Luồng điều hướng:Người dùng nhấn “Thông tin cá nhân” trong menu trái → kiểm tra thông tin hiện tại → nhấn các nút như “Sửa thông tin”, “Sửa mật khẩu”, “Sửa email” để cập nhật → dùng “Làm mới” để tải lại dữ liệu ban đầu nếu cần.



Hình ảnh 3.28 Giao điện người dùng ở Thông tin cá nhân

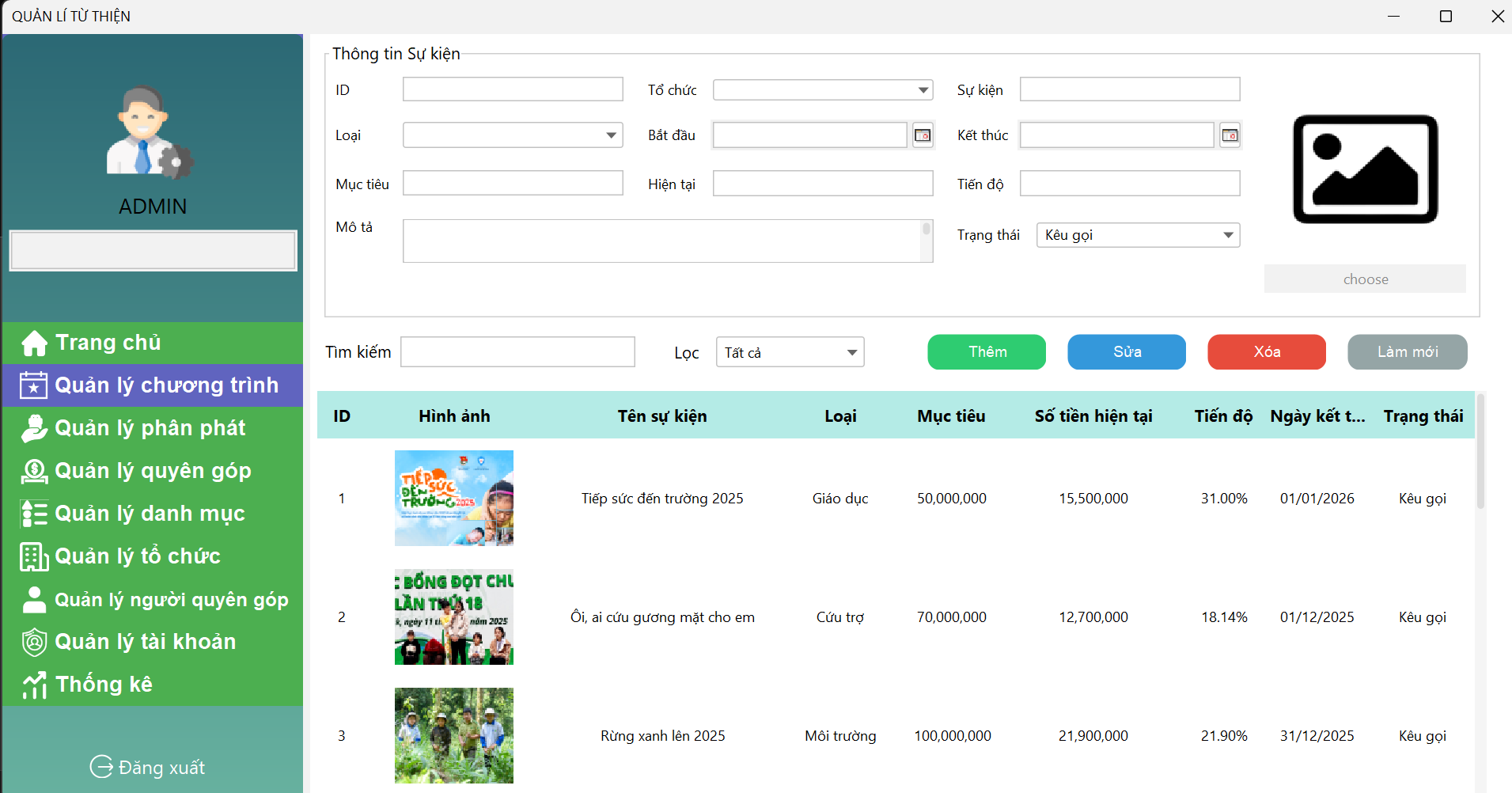
* + 1. Giao diện admin ở trang chủ
       - Bố cục tổng thể: Giao diện chia hai phần: thanh menu bên trái chứa thông tin admin và các chức năng quản lý, phần nội dung bên phải hiển thị tiêu đề và 4 ô thống kê chính giữa màn hình.
       - Các thành phần giao diện:Sidebar gồm ảnh đại diện, vai trò ADMIN, tên “Trần Thanh Ngân”, các mục Trang chủ, Quản lý chương trình, Quản lý quyên góp, Quản lý tổ chức, Quản lý tài khoản và nút Đăng xuất; phần chính có tiêu đề “Trang chủ” và 4 ô thống kê: số lượt quyên góp , sự kiện , tổ chức , tài khoản.



Hình ảnh 3.29 Giao diện admin ở trang chủ

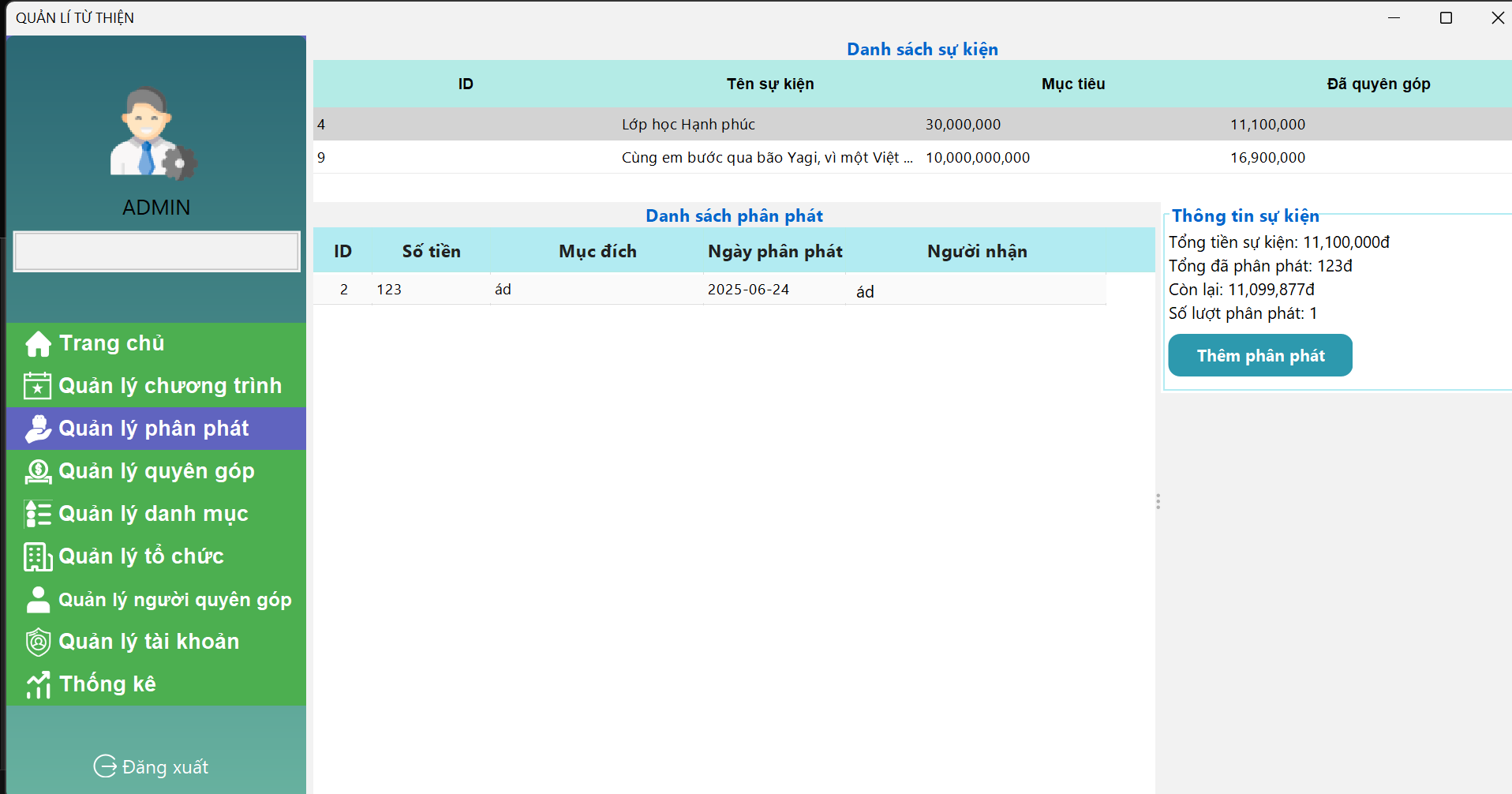
* + 1. Giao điện admin ở quản lý chương trình
       - Bố cục tổng thể: Giao diện chia thành hai vùng: sidebar bên trái chứa menu chức năng và thông tin admin, khung phải hiển thị biểu mẫu nhập liệu ở trên và bảng danh sách sự kiện bên dưới.
       - Các thành phần giao diện: Form trên cùng gồm các trường nhập thông tin sự kiện (ID, tổ chức, loại, mô tả, mục tiêu, ngày bắt đầu, kết thúc, tiến độ...), kèm khung ảnh đại diện sự kiện và các nút Thêm, Sửa, Xóa, Làm mới bên dưới ô tìm kiếm.
       - Luồng điều hướng: Admin vào mục “Quản lý chương trình” từ sidebar

→ xem danh sách các sự kiện đang có → chọn sự kiện để sửa hoặc xóa, hoặc nhập thông tin mới và nhấn Thêm để tạo sự kiện mới.



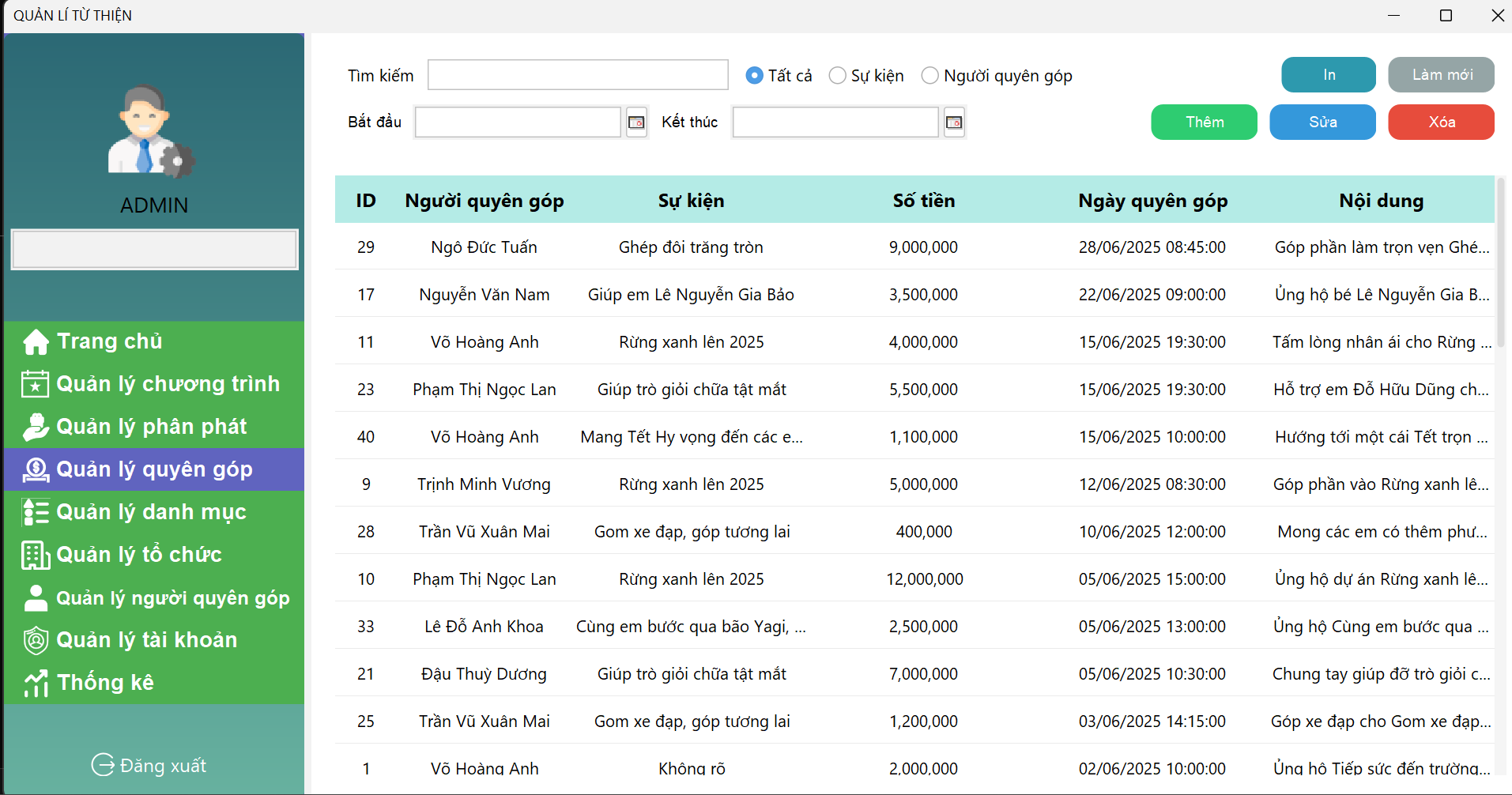
Hình ảnh 3.30 Giao diện admin ở quản lý chương trình

* + 1. Giao điện admin ở quản lý phân phát
* **Bố cục tổng thể:** Giao diện được chia thành ba khu vực chính: phía trên là bảng danh sách các sự kiện từ thiện với thông tin tổng quan như ID, tên sự kiện, mục tiêu và số tiền đã quyên góp; ở giữa là bảng phân phát hiển thị các lượt phân phát tương ứng với sự kiện được chọn; bên phải là khung thông tin tóm tắt về tình hình phân phát của sự kiện, bao gồm tổng tiền đã quyên góp, số tiền đã phân phát, số còn lại và số lượt phân phát.
* **Các thành phần giao diện:** Bảng sự kiện cho phép admin chọn sự kiện để quản lý phân phát. Bảng phân phát bên dưới hiển thị chi tiết các lượt phân phát đã thực hiện, gồm các cột: ID, số tiền, mục đích, ngày phân phát và người nhận. Phần khung thông tin bên phải trình bày số liệu tổng hợp, giúp người dùng dễ dàng theo dõi tình hình phân phối tài chính. Nút “Thêm phân phát” được đặt ngay bên dưới khung này, hỗ trợ thao tác nhanh trong việc cập nhật dữ liệu mới.
* **Luồng điều hướng:** Admin truy cập chức năng “Quản lý phân phát” → chọn một sự kiện từ bảng danh sách phía trên → xem danh sách các lượt phân phát tương ứng ở bảng bên dưới → theo dõi các số liệu tổng quan ở khung bên phải → nhấn nút “Thêm phân phát” để ghi nhận đợt phân phát mới.



Hình ảnh 3.31 Giao diện admin ở quản lý chương trình

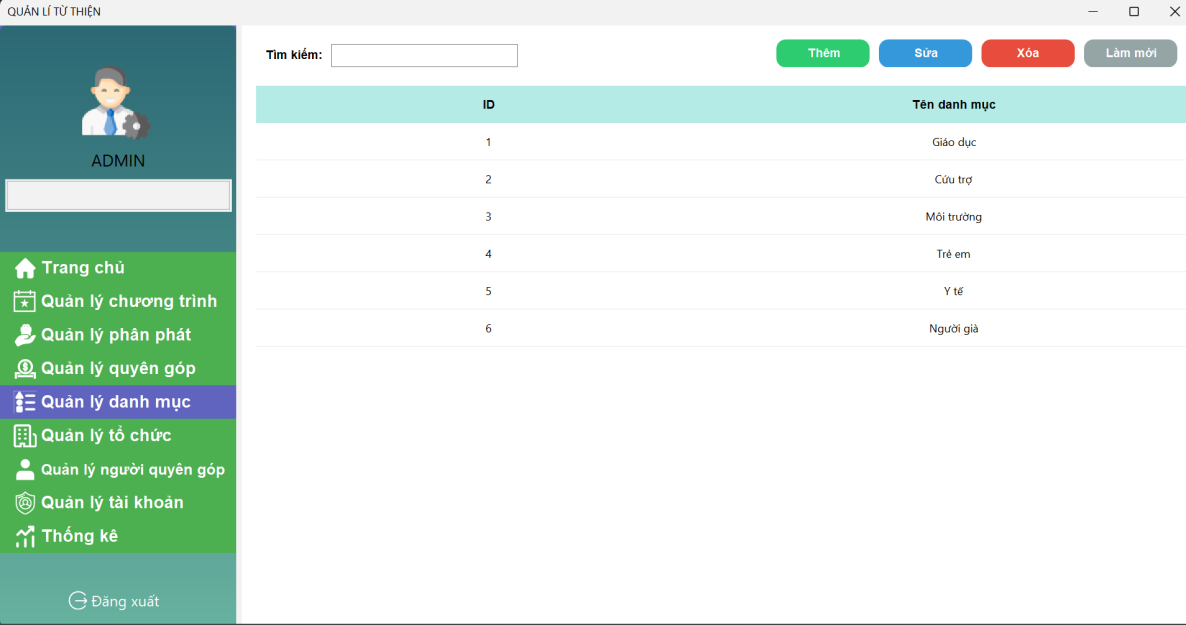
* + 1. Giao điện admin ở quản lý quyên góp
       - Bố cục tổng thể: Giao diện chia hai phần: menu bên trái cố định và bảng quản lý quyên góp bên phải với thanh tìm kiếm ở trên và bảng dữ liệu bên dưới.
       - Các thành phần giao diện: Có ô tìm kiếm kèm lựa chọn tìm theo ID / Sự kiện / Người quyên góp, nút “In” và “Làm mới”, bảng liệt kê các lần quyên góp gồm: ID, người quyên góp, tên sự kiện, số tiền, ngày giờ, nội dung.
       - Luồng điều hướng: Admin chọn “Quản lý quyên góp” từ sidebar → tìm kiếm theo tiêu chí mong muốn → xem danh sách chi tiết các lần quyên góp → nhấn “In” nếu cần xuất dữ liệu, hoặc “Làm mới” để cập nhật lại bảng.



Hình ảnh 3.32 Giao điện admin ở quản lý quyên góp

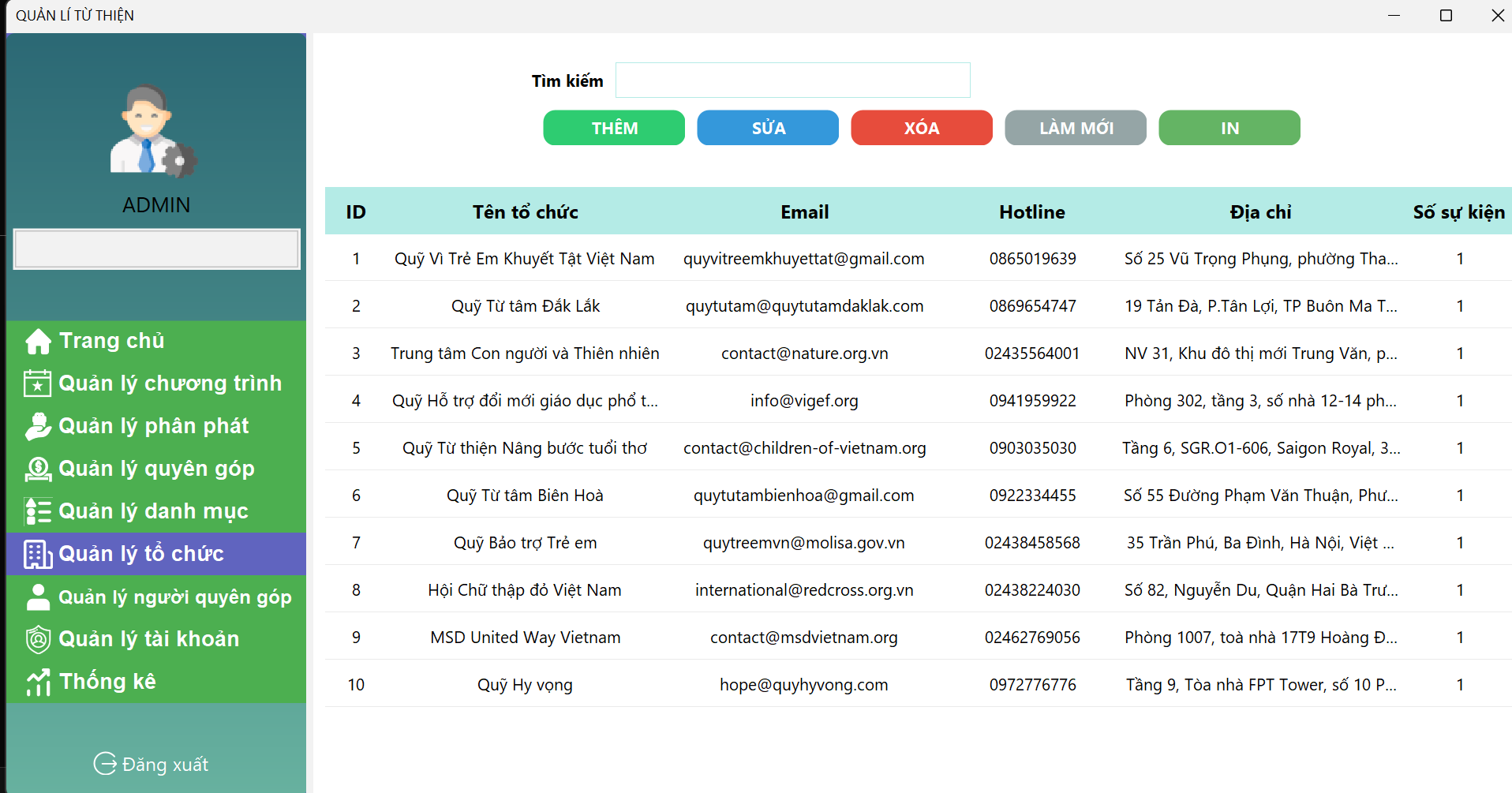
1. * 1. Giao điện admin ở quản lý danh mục

* Bố cục tổng thể: Giao diện được chia làm hai phần chính: phần trên gồm ô tìm kiếm và nhóm các nút chức năng chính, phần dưới là bảng hiển thị danh sách các danh mục với hai cột chính là ID và Tên danh mục.
* Các thành phần giao diện: Ô tìm kiếm cho phép người dùng lọc danh mục theo từ khóa. Phía bên phải là bốn nút chức năng bao gồm: Thêm, Sửa, Xóa, Làm mới, hỗ trợ quản lý danh mục. Dưới các thành phần này là bảng dữ liệu hiển thị danh sách danh mục hiện có trong hệ thống. Mỗi danh mục gồm hai thông tin cơ bản là ID và tên danh mục, giúp việc quản lý trở nên rõ ràng và dễ kiểm soát.
* Luồng điều hướng: Admin chọn mục “Quản lý danh mục” để xem danh sách các danh mục hiện tại → sử dụng ô tìm kiếm để lọc theo từ khóa nếu cần → chọn danh mục để sửa hoặc xóa → hoặc nhấn nút “Thêm” để tạo mới danh mục → nhấn “Làm mới” để tải lại toàn bộ dữ liệu sau khi thực hiện thao tác.



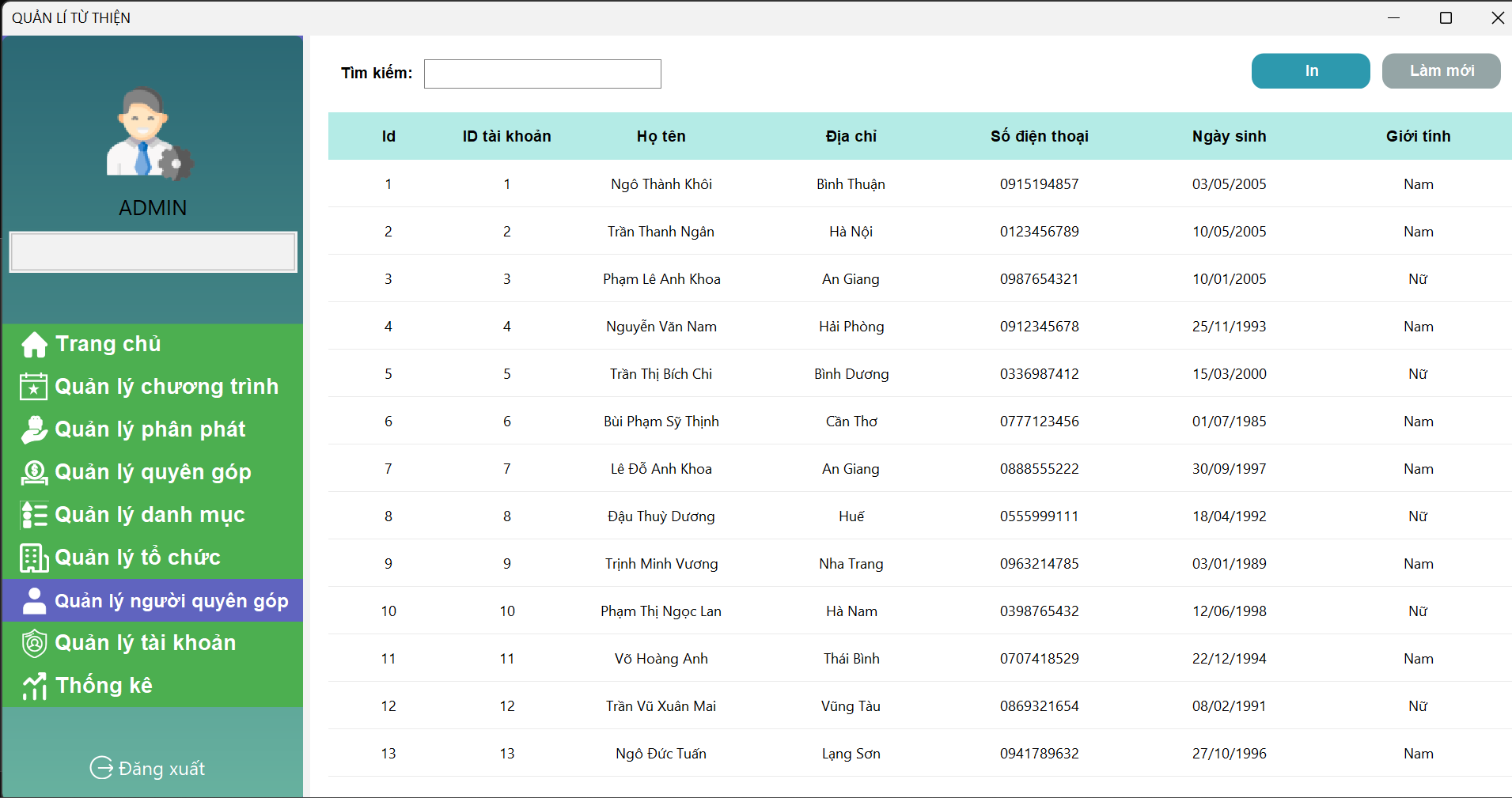
Hình ảnh 3.33 Giao điện admin ở quản lý danh mục

* + 1. Giao điện admin ở quản lý tổ chức
       - Bố cục tổng thể: Màn hình chia làm 2 phần chính: khung tìm kiếm và thao tác phía trên, danh sách tổ chức từ thiện hiển thị dạng bảng phía dưới.
       - Các thành phần giao diện: Gồm các ô nhập tìm kiếm theo mã tổ chức, tên, email, hotline, địa chỉ; các nút thao tác như Thêm, Sửa, Xóa, Clear, Print; bảng danh sách tổ chức với các cột: ID, tên tổ chức, email, hotline, địa chỉ, số sự kiện.
       - Luồng điều hướng: Admin chọn “Quản lý tổ chức” → nhập tiêu chí tìm kiếm nếu cần → thực hiện Thêm/Sửa/Xóa hoặc In dữ liệu tổ chức → danh sách được cập nhật tương ứng bên dưới.



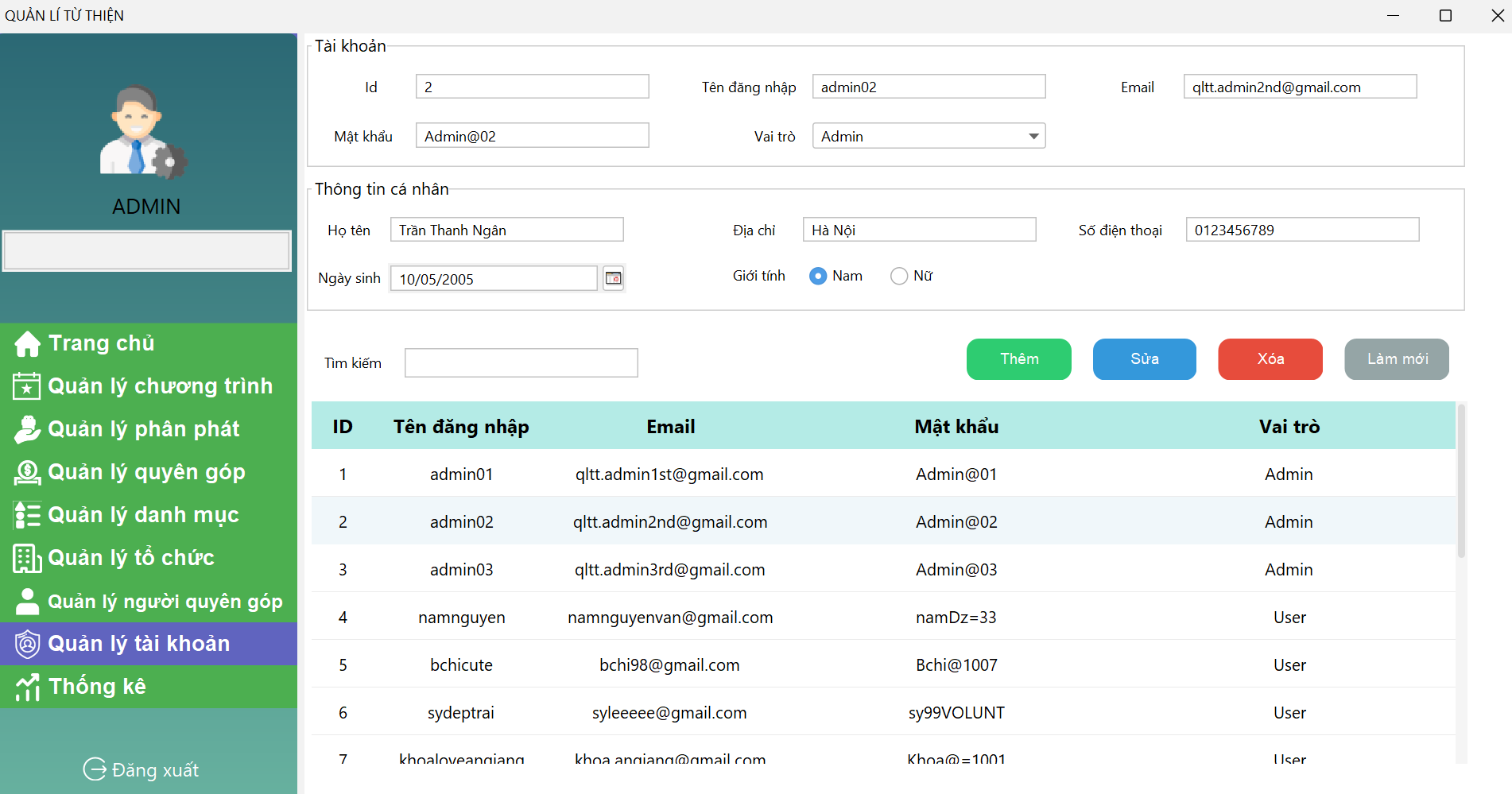
Hỉnh ảnh 3.34 Giao điện admin ở quản lý tổ chức

* + 1. Giao diện admin ở quản lý người quyên góp
* Bố cục tổng thể: Phần bên phải gồm hai khu vực chính: phía trên là ô tìm kiếm và các nút chức năng, phía dưới là bảng hiển thị danh sách người quyên góp. Giao diện được bố trí rõ ràng, thuận tiện cho thao tác quản lý.
* Các thành phần giao diện: Giao diện có ô tìm kiếm dùng để lọc nhanh thông tin. Bên cạnh là hai nút “In” và “Làm mới”. Dưới cùng là bảng danh sách gồm các cột: ID, ID tài khoản, Họ tên, Địa chỉ, Số điện thoại, Ngày sinh và Giới tính.
* Luồng điều hướng: Admin nhập từ khóa vào ô tìm kiếm để lọc dữ liệu. Có thể nhấn “In” để xuất danh sách hoặc “Làm mới” để tải lại dữ liệu. Các thao tác đều tập trung tại khu vực bảng bên dưới.



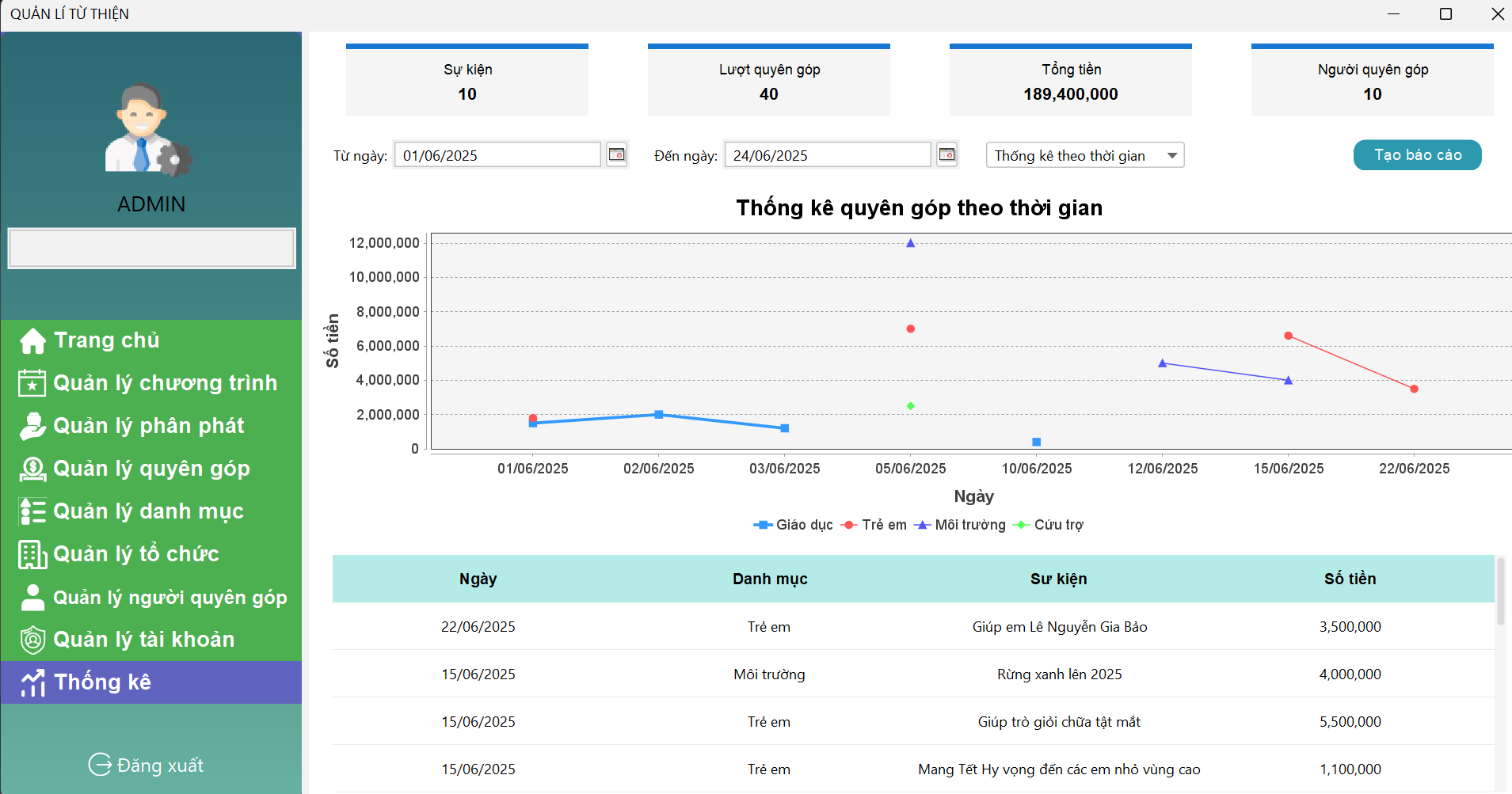
Hỉnh ảnh 3.35 Giao điện admin ở quản lý tổ chức

* + 1. Giao điện admin ở quản lý tài khoản
       - Bố cục tổng thể: Màn hình chia làm 3 phần chính: nhập thông tin tài khoản ở trên, vùng thao tác ở giữa, bảng danh sách tài khoản bên dưới.
       - Các thành phần giao diện: Gồm các ô nhập ID, tên đăng nhập, mật khẩu, email, vai trò; thông tin cá nhân như họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại; kèm các nút chức năng Thêm, Sửa, Xóa, Làm mới; bảng bên dưới liệt kê các tài khoản theo các cột: ID, tên đăng nhập, email, mật khẩu, vai trò.
       - Luồng điều hướng: Admin chọn “Quản lý tài khoản” → nhập hoặc chọn tài khoản cần thao tác → chỉnh sửa/thêm mới thông tin → nhấn nút phù hợp → danh sách cập nhật bên dưới.



Hình ảnh 3.36 Giao điện admin ở quản lý tài khoản

* + 1. Giao diện admin ở trang thống kê
* Bố cục tổng thể: Phần bên phải chia làm ba khu vực: phía trên là các chỉ số tổng hợp; giữa là biểu đồ thống kê trực quan theo thời gian; bên dưới là bảng chi tiết các khoản quyên góp. Giao diện giúp người dùng dễ dàng nắm bắt tổng quan và chi tiết dữ liệu.
* Các thành phần giao diện: Phía trên gồm 4 ô hiển thị số liệu: tổng số sự kiện, lượt quyên góp, tổng số tiền và số người quyên góp. Bên dưới là biểu đồ đường kết hợp điểm, thể hiện số tiền quyên góp theo ngày và danh mục. Dưới cùng là bảng dữ liệu liệt kê theo cột: Ngày, Danh mục, Sự kiện và Số tiền.
* Luồng điều hướng: Admin chọn khoảng thời gian từ ô “Từ ngày” và “Đến ngày”, sau đó chọn kiểu thống kê và nhấn “Tạo báo cáo”. Hệ thống sẽ cập nhật biểu đồ và bảng dữ liệu tương ứng, giúp theo dõi xu hướng và chi tiết các hoạt động quyên góp trong khoảng thời gian đã chọn.



Hình ảnh 3.37 Giao điện admin ở thống kê

* 1. Cách tính năng nổi bật
     1. Tính năng quên mật khẩu
* Mô tả chức năng:

Chức năng "Quên mật khẩu" cho phép người dùng lấy lại mật khẩu trong trường hợp quên hoặc không thể đăng nhập. Người dùng chỉ cần nhập tên đăng nhập và email đã đăng ký. Hệ thống sẽ xác minh thông tin, sau đó tự động gửi một mật khẩu mới về địa chỉ email đã đăng ký.

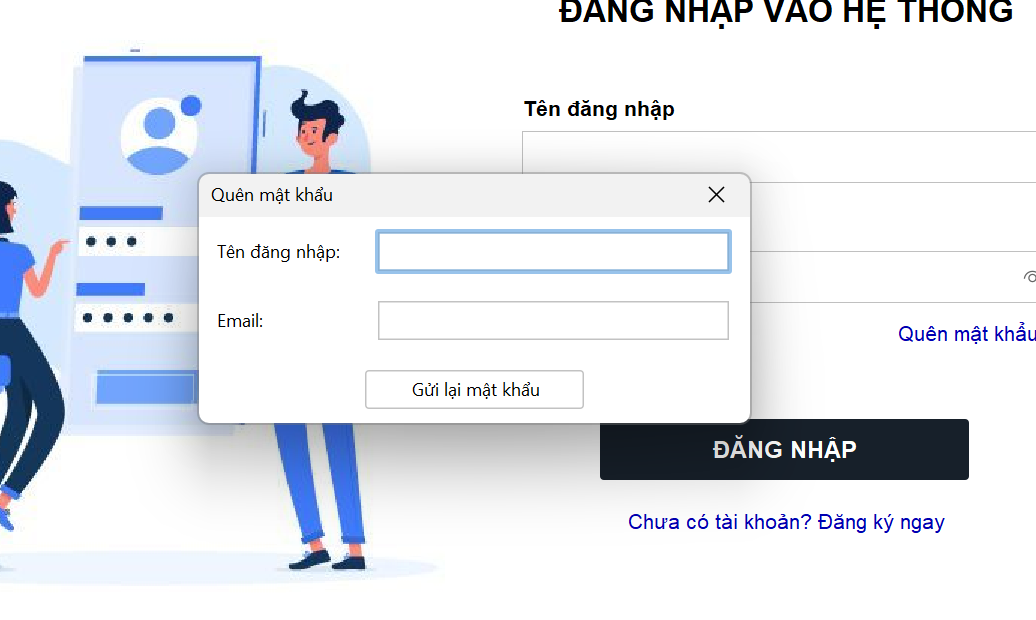
* Quy trình sử dụng:

1. Tại giao diện đăng nhập, người dùng nhấn vào liên kết "Quên mật khẩu?".
2. Một cửa sổ hiển thị, yêu cầu nhập:

* Tên đăng nhập
* Email

1. Sau khi nhấn nút "Gửi lại mật khẩu", hệ thống sẽ:
   * Kiểm tra tính hợp lệ của tên đăng nhập và email.
   * Nếu khớp, tạo một mật khẩu mới ngẫu nhiên.
   * Gửi mật khẩu mới đến email người dùng.
2. Người dùng kiểm tra email và sử dụng mật khẩu mới để đăng nhập lại vào hệ thống.

* Lưu ý bảo mật:
* Mật khẩu mới nên được thay đổi ngay sau khi đăng nhập.
* Email phải chính xác và còn hoạt động để nhận được thông tin khôi phục.



Hình ảnh 3.38 Giao điện quên mật khẩu

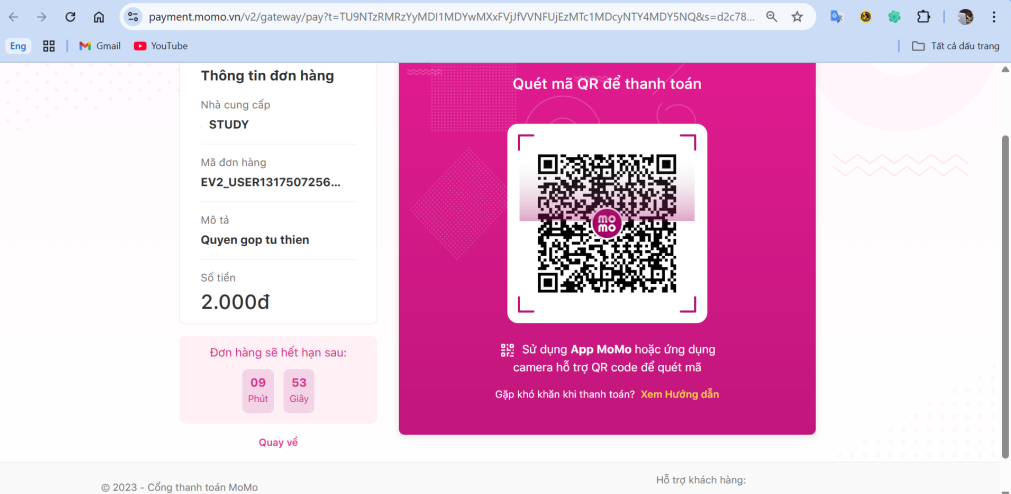
* + 1. Tính năng thanh toán momo
* **Mô tả chức năng:**

Chức năng cho phép người dùng **quyên góp trực tuyến qua ví điện tử MoMo**. Sau khi chọn một sự kiện muốn quyên góp, người dùng nhập số tiền và nhấn nút **"Quyên góp"**. Hệ thống sẽ tự động tạo **mã QR thanh toán MoMo**, hiển thị trên màn hình để người dùng quét và thực hiện thanh toán.

* **Quy trình sử dụng:**

1. Người dùng chọn một sự kiện cần quyên góp.
2. Nhập số tiền và nhấn **"Quyên góp"**.
3. Hệ thống chuyển hướng đến giao diện thanh toán MoMo, hiển thị:
   * Mã đơn hàng
   * Mô tả giao dịch
   * Số tiền
   * **Mã QR MoMo** để quét thanh toán
4. Người dùng mở ứng dụng **MoMo** hoặc các ứng dụng hỗ trợ quét QR để thanh toán.
5. Sau khi thanh toán thành công, hệ thống ghi nhận giao dịch và cập nhật số tiền quyên góp cho sự kiện tương ứng.

* **Lưu ý:**
* Mã QR có **thời hạn sử dụng trong vài phút**.
* Giao dịch cần đúng số tiền và mô tả để hệ thống xác nhận thành công.
* Trong trường hợp lỗi thanh toán, người dùng có thể chọn **“Quay về”** và thử lại.



Hình ảnh 3.37 Giao điện thanh toán momo

# CHƯƠNG 4: KẾT QUẢ VÀ KIẾN NGHỊ

## Kết quả đạt được

Sau quá trình nghiên cứu ngôn ngữ lập trình Java và triển khai ứng dụng quản lý công tác từ thiện, sinh viên đã đạt được những kết quả cụ thể cả về lý thuyết lẫn thực hành như sau:

* + 1. Về kiến thức lý thuyết
       - **Nắm vững kiến thức nền tảng về Java:** Sinh viên đã hiểu rõ cú pháp, kiểu dữ liệu, biến, toán tử, câu lệnh điều kiện và vòng lặp, xử lý mảng, chuỗi, lập trình hướng đối tượng (OOP), kế thừa, đa hình, giao diện, và xử lý ngoại lệ trong Java.
       - **Tiếp cận tư duy kiến trúc phần mềm:** Thông qua mô hình MVC, sinh viên đã hiểu được cách tổ chức mã nguồn hợp lý, tách biệt giữa giao diện, xử lý và dữ liệu. Điều này giúp phần mềm dễ phát triển, bảo trì, và mở rộng.
       - **Tìm hiểu cách làm việc với cơ sở dữ liệu bằng JDBC:** Sinh viên đã học cách thiết lập kết nối giữa Java và cơ sở dữ liệu (MySQL/SQLite), thực hiện truy vấn, cập nhật, xoá dữ liệu thông qua lệnh SQL kết hợp với Java
    2. Về kỹ năng thực hành

Xây dựng được một ứng dụng Java hoàn chỉnh phục vụ quản lý công tác từ thiện, với các chức năng:

* + - * **Quản lý người đóng góp:** Lưu trữ thông tin cá nhân, số tiền/quà đóng góp, thời gian ủng hộ.
      * **Quản lý người nhận hỗ trợ:** Theo dõi danh sách người có hoàn cảnh khó khăn theo từng địa phương.
      * **Quản lý đợt từ thiện:** Tổ chức các đợt phát quà theo tên đợt, thời gian, địa điểm cụ thể.
      * **Ghi nhận phát quà:** Lưu thông tin quà tặng đã phát cho từng người nhận.
      * **Thống kê:** Tổng hợp dữ liệu để phân tích hiệu quả hoạt động từ thiện theo thời gian, đối tượng, khu vực,...

**Thành thạo thao tác với giao diện đồ hoạ Java Swing:** Sử dụng các thành phần như JFrame, JPanel, JTable, JTextField, JButton, JComboBox,... để tạo giao diện thân thiện, dễ sử dụng.

**Ứng dụng kiến thức vào giải quyết bài toán thực tế:** Hiểu được quy trình triển khai phần mềm từ phân tích yêu cầu → thiết kế dữ liệu → xây dựng chức năng → kiểm thử → hoàn thiện.

## Kiến nghị

* + 1. Kiến nghị về mở rộng phần mềm
       - **Tích hợp tính năng phân quyền người dùng:** Cho phép tạo các vai trò như Quản trị viên (Admin), Nhân viên nhập liệu, Người xem,… nhằm tăng tính bảo mật và chuyên nghiệp.
       - **Thêm chức năng xuất báo cáo:** Hỗ trợ in danh sách người nhận, xuất bảng thống kê ra file PDF/Excel để phục vụ kiểm tra, quyết toán.
       - **Tối ưu hoá hệ thống tìm kiếm:** Áp dụng bộ lọc nâng cao để tìm kiếm theo họ tên, địa chỉ, thời gian, loại hỗ trợ,...
       - **Hỗ trợ đa ngôn ngữ (Việt – Anh):** Phù hợp với tổ chức có yếu tố nước ngoài.’
    2. Kiến nghị về nâng cấp nền tảng công nghệ

**Xây dựng phiên bản Web hoặc Mobile:**

* + - * Web app: Triển khai bằng Java Spring Boot hoặc JSP/Servlet để làm việc từ xa, quản lý tập trung.
      * Mobile: Xây dựng app Android bằng Java giúp sử dụng tại nơi tổ chức hoạt động từ thiện (không cần máy tính).

**Sử dụng cơ sở dữ liệu trên nền tảng đám mây (Cloud Database):** Đảm bảo truy cập mọi lúc, mọi nơi, và nâng cao tính liên kết giữa các chi nhánh của tổ chức từ thiện.

* + 1. Kiến nghị với giảng viên và nhà trường

**Tăng cường các đề tài gần với thực tế cuộc sống:** Những đề tài như từ thiện, quản lý hộ nghèo, y tế cộng đồng,... giúp sinh viên vừa rèn kỹ thuật, vừa góp phần hỗ trợ cộng đồng.

**Tổ chức các buổi workshop chuyên sâu:** Hướng dẫn cụ thể về mô hình MVC, thiết kế giao diện Swing nâng cao, hoặc tích hợp Java với API/Web Service.

**Khuyến khích sinh viên triển khai ứng dụng lên nền tảng thực tế:** Thử nghiệm phần mềm với các đơn vị từ thiện địa phương như đoàn trường, tổ chức thiện nguyện,... để kiểm tra khả năng vận hành thực tế.

# TÀI LIỆU THAM KHẢO

1. Lịch sử của java: [https://viblo.asia/p/lich-su-cua-java-BQyJKanMVMe:](https://viblo.asia/p/lich-su-cua-java-BQyJKanMVMe)
2. Tổng quan về java: [https://codelearn.io/learning/java-co-](https://codelearn.io/learning/java-co-ban?tab=introduce)

[ban?tab=introduce](https://codelearn.io/learning/java-co-ban?tab=introduce)

1. Các thành phần cơ bản của Java:[https://kungfutech.edu.vn/bai-](https://kungfutech.edu.vn/bai-viet/java/cau-truc-mot-chuong-trinh-java)

[viet/java/cau-truc-mot-chuong-trinh-java](https://kungfutech.edu.vn/bai-viet/java/cau-truc-mot-chuong-trinh-java)

1. Khái niệm Java Swing: <https://codegym.vn/blog/java-swing-la-gi-2/>