**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

A logo with a person holding a book

AI-generated content may be incorrect.

**ĐỒ ÁN CUỐI KỲ**

**NGÀNH CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG APP QUẢN LÝ CHI TIÊU**

Giảng viên hướng dẫn: **ThS. Nguyễn Thanh Truyền**

Sinh viên thực hiện: **Trần Thanh Phương**

Mã số sinh viên: **1150080165**

Lớp: **11\_ĐH\_CNPM1**

Khoá: **2022-2026**

**TP. Hồ Chí Minh, 25 tháng 8 năm 2025**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

A logo with a person holding a book

AI-generated content may be incorrect.

**ĐỒ ÁN CUỐI KỲ**

**ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG APP QUẢN LÝ CHI TIÊU**

Giảng viên hướng dẫn: **ThS. Nguyễn Thanh Truyền**

Sinh viên thực hiện: **Trần Thanh Phương**

Mã số sinh viên: **1150080165**

Lớp: **11\_ĐH\_CNPM1**

Khoá: **2022-2026**

**TP. Hồ Chí Minh, 25 tháng 08 năm 2025**

**LỜI CẢM ƠN**

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến gia đình – những người đã luôn ở bên động viên, tạo điều kiện và là điểm tựa tinh thần vững chắc để em yên tâm học tập tại Trường Đại học Tài Nguyên và Môi Trường TP. Hồ Chí Minh.

Em cũng xin gửi lời cảm ơn chân thành nhất đến thầy **Nguyễn Thanh Truyền**, người đã trực tiếp hướng dẫn, tận tình chỉ bảo và luôn đưa ra những lời khuyên quý báu trong suốt quá trình em thực hiện **đề tài xây dựng ứng dụng quản lý chi tiêu**. Sự hỗ trợ và định hướng của thầy đã giúp em vượt qua nhiều khó khăn, hoàn thiện đề tài ngày một tốt hơn.

Em xin trân trọng cảm ơn tất cả quý thầy cô Trường Đại học Tài Nguyên và Môi Trường TP.HCM, đặc biệt là quý thầy cô Khoa Công Nghệ Thông Tin. Những kiến thức, kinh nghiệm và sự tận tụy trong giảng dạy của quý thầy cô đã giúp em có nền tảng vững vàng để phát triển và hoàn thành đề tài này.

Mặc dù đã nỗ lực hoàn thành trong phạm vi khả năng cho phép, nhưng chắc chắn đề tài của em vẫn còn những thiếu sót. Em rất mong nhận được những ý kiến đóng góp chân thành từ quý thầy cô để có thể tiếp tục cải thiện, hoàn thiện sản phẩm, đồng thời tích lũy thêm kinh nghiệm cho bản thân để làm hành trang cho công việc và cuộc sống sau này.

**Sinh viên thực hiện**

**Trần Thanh Phương**

**LỜI CAM ĐOAN**

Em xin cam đoan rằng đồ án với đề tài **“Xây dựng ứng dụng quản lý chi tiêu”** là kết quả nghiên cứu và thực hiện độc lập của riêng em, dưới sự hướng dẫn của **ThS. Nguyễn Thanh Truyền**.

Toàn bộ nội dung trong đề tài được thực hiện một cách trung thực, không sao chép từ bất kỳ đồ án hay công trình nghiên cứu nào khác. Những tài liệu tham khảo được sử dụng trong quá trình thực hiện đều đã được trích dẫn và ghi rõ nguồn gốc.

Em xin hoàn toàn chịu trách nhiệm trước nhà trường nếu có bất kỳ hành vi sao chép, gian dối hay không trung thực nào liên quan đến kết quả của đồ án này.

**Sinh viên thực hiện**

**Trần Thanh Phương**

**TÓM TẮT**

Trong bối cảnh nhu cầu quản lý tài chính cá nhân ngày càng trở nên quan trọng, việc ứng dụng công nghệ vào theo dõi thu chi đóng vai trò thiết yếu trong việc giúp người dùng kiểm soát dòng tiền, đánh giá thói quen chi tiêu và xây dựng kế hoạch tài chính hiệu quả. Việc ghi chép thủ công thường gây mất thời gian, dễ thất lạc dữ liệu và thiếu tính tổng hợp, từ đó làm giảm khả năng theo dõi tài chính một cách khoa học. Điều này đã thúc đẩy nhu cầu về một giải pháp số hóa thuận tiện, dễ sử dụng và mang lại hiệu quả cao trong quản lý tài chính cá nhân.

Đề tài “Xây dựng ứng dụng quản lý chi tiêu” được thực hiện với mục tiêu phát triển một công cụ hỗ trợ người dùng ghi nhận thu nhập và chi tiêu hằng ngày, phân loại giao dịch, quản lý ví tiền và theo dõi biến động số dư. Bên cạnh đó, ứng dụng cung cấp hệ thống thống kê bằng biểu đồ và báo cáo theo nhiều khoảng thời gian, giúp người dùng dễ dàng đánh giá tình hình tài chính cũng như đưa ra các quyết định chi tiêu hợp lý hơn.

Về mặt kỹ thuật, ứng dụng được xây dựng theo mô hình Client–Server với Flutter đảm nhiệm phần giao diện người dùng, mang lại trải nghiệm trực quan và linh hoạt. Backend được phát triển bằng Spring Boot nhằm xử lý nghiệp vụ, cung cấp các API REST và quản lý dữ liệu một cách ổn định. Cơ sở dữ liệu MySQL được sử dụng để đảm bảo tính toàn vẹn dữ liệu, trong khi cơ chế xác thực JWT giúp bảo vệ thông tin người dùng. Tất cả các API được kiểm thử bằng Postman nhằm bảo đảm tính chính xác và ổn định trước khi tích hợp vào hệ thống.

   Kết quả của đề tài là một ứng dụng hoàn chỉnh, đáp ứng tốt nhu cầu quản lý tài chính cá nhân thông qua các chức năng như ghi nhận giao dịch, quản lý ví, phân loại thu chi và hiển thị báo cáo trực quan. Đây cũng là nền tảng quan trọng để mở rộng trong tương lai với các tính năng như nhắc nhở thanh toán, đồng bộ đám mây hoặc chia sẻ ngân sách, góp phần nâng cao hiệu quả quản lý tài chính cho người dùng.

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc215557512)

[CHƯƠNG 1. TỔNG QUAN TÀI LIỆU 3](#_Toc215557513)

[1.1. Giới thiệu đề tài 3](#_Toc215557514)

[1.1.1. Nhu cầu thực tế của đề tài 3](#_Toc215557515)

[1.1.2. Lý do chọn đề tài 3](#_Toc215557516)

[1.1.3. Mục tiêu đề tài 3](#_Toc215557517)

[1.1.5. Phương pháp nghiên cứu 4](#_Toc215557518)

[1.1.6. Nội dung nghiên cứu 5](#_Toc215557519)

[1.1.7. Khảo sát đề tài 5](#_Toc215557520)

[1.1.8. Phát biểu bài toán 6](#_Toc215557521)

[1.2. Cơ sở lý thuyết 7](#_Toc215557522)

[1.2.1. VSCode (Visual Studio Code) 7](#_Toc215557523)

[1.2.2. Ngôn ngữ Dart 7](#_Toc215557524)

[1.2.3. Flutter 9](#_Toc215557525)

[1.2.4 Java 10](#_Toc215557526)

[1.2.5. Spring Boot 10](#_Toc215557527)

[1.2.6 MySQL 11](#_Toc215557528)

[1.2.7 Postman 12](#_Toc215557529)

[1.2.8 Cloudinary 13](#_Toc215557530)

[1.2.9 Git / GitHub 14](#_Toc215557531)

[1.2.5. Kiến trúc của app báo cáo chi tiêu 14](#_Toc215557532)

[CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN 17](#_Toc215557533)

[2.1. Đặc tả yêu cầu 17](#_Toc215557534)

[2.1.1. Danh sách các Actor 17](#_Toc215557535)

[2.1.2. Danh sách các Usecase 17](#_Toc215557536)

[2.2. Phân tích và thiết kế hệ thống 18](#_Toc215557537)

[2.2.1. Kiến trúc hệ thống 18](#_Toc215557538)

[2.2.2. Thiết kế hệ thống 21](#_Toc215557539)

[2.2.3. Thiết kế cơ sở dữ liệu 44](#_Toc215557540)

[CHƯƠNG 3. CHƯƠNG 3. CÀI ĐẶT THỬ NGHIỆM 52](#_Toc215557541)

[3.1. Kết quả giao diện đạt được 52](#_Toc215557542)

[3.1.1. Giao diện người dùng 52](#_Toc215557543)

[3.1.2. Giao diện Admin 76](#_Toc215557544)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 77](#_Toc215557545)

[1. Kết luận 77](#_Toc215557546)

[2. Hướng phát triển 77](#_Toc215557547)

[TÀI LIỆU THAM KHẢO. 78](#_Toc215557548)

**DANH MỤC SƠ ĐỒ, HÌNH**

[Hình 1. 1 Logo vscode 7](#_Toc215571752)

[Hình 1. 2 Logo ngôn ngữ Dart 8](#_Toc215571753)

[Hình 1.3 Ứng dụng của ngôn ngữ Dart. 8](#_Toc215571754)

[Hình 1. 4 Logo flutter 9](#_Toc215571755)

[Hình 1. 5 Logo Java 10](#_Toc215571756)

[Hình 1. 6 Logo Spring Boot 11](#_Toc215571757)

[Hình 1. 7 Logo MySQL 11](#_Toc215571758)

[Hình 1. 8 Logo ứng dụng Postman 12](#_Toc215571759)

[Hình 1. 9 Logo ứng dụng Cloudinary 13](#_Toc215571760)

[Hình 1. 10 Logo ứng dụng GitHub 14](#_Toc215571761)

[Hình 2. 1 Sơ đồ tổng quan hệ thống 19](#_Toc215571771)

[Hình 2. 2 Sơ đồ usecase tổng quát 22](#_Toc215571772)

[Hình 2. 3 DFD Level 0 – Context Diagram 24](#_Toc215571773)

[Hình 2. 4 Biểu đồ hoạt động đăng ký 25](#_Toc215571774)

[Hình 2. 5 Biểu đồ hoạt động đăng nhập 26](#_Toc215571775)

[Hình 2. 6 Biểu đồ hoạt động thêm thu nhập 27](#_Toc215571776)

[Hình 2. 7 Biểu đồ hoạt động thêm chi tiêu 28](#_Toc215571777)

[Hình 2. 8 Biểu đồ hoạt động xem báo cáo 29](#_Toc215571778)

[Hình 2. 9 Biểu đồ hoạt động quản lý ví 30](#_Toc215571779)

[Hình 2. 10 Biểu đồ hoạt động quản lý danh mục 31](#_Toc215571780)

[Hình 3. 1 Giao diện danh mục chi tiêu 52](#_Toc215571787)

[Hình 3. 2 Hình ảnh giao diện thu nhập 53](#_Toc215571788)

[Hình 3. 3 Giao diện trang đăng ký 54](#_Toc215571789)

[Hình 3. 4 Giao diện đăng nhập 55](#_Toc215571790)

[Hình 3. 5 Giao diện thêm khoản chi 56](#_Toc215571791)

[Hình 3. 6 Giao diện thêm khoản thu 57](#_Toc215571792)

[Hình 3. 7 Giao diện xem thống kê thu nhập 58](#_Toc215571793)

[Hình 3. 8 Giao diện xem thống kê thu nhập 59](#_Toc215571794)

[Hình 3. 9 Giao diện báo cáo và thống kê 60](#_Toc215571795)

[Hình 3. 10 Giao diện AI dự đoán chi tiêu 61](#_Toc215571796)

[Hình 3. 11 Giao diện giao dịch định kì 62](#_Toc215571797)

[Hình 3. 12 Giao diện tạo giao dịch định kỳ 63](#_Toc215571798)

[Hình 3. 13 Giao diện quản lý ví 64](#_Toc215571799)

[Hình 3. 14 Giao diện tạo ví mới 65](#_Toc215571800)

[Hình 3. 15 Giao diện quản lý mục tiêu tài chính 66](#_Toc215571801)

[Hình 3. 16 Giao diện tạo mục tiêu tài chính 67](#_Toc215571802)

[Hình 3. 17 Giao diện cập nhật tiến độ mục tiêu tài chính 68](#_Toc215571803)

[Hình 3. 18 Giao diện thêm ngân sách 69](#_Toc215571804)

[Hình 3. 19 Giao diện thống kê ngân sách 70](#_Toc215571805)

[Hình 3. 20 Giao diện xem thống kê ngân sách bằng lịch 71](#_Toc215571806)

[Hình 3. 21 Giao diện thông báo 72](#_Toc215571807)

[Hình 3. 22 Giao diện hồ sơ cá nhân 73](#_Toc215571808)

[Hình 3. 23 Giao diện đổi mật khẩu 74](#_Toc215571809)

[Hình 3. 24 Giao diện xuất excel 75](#_Toc215571810)

[Hình 3. 25 Giao diện admin quản lý người dùng 76](#_Toc215571811)

[Hình 3. 26 Giao diện quản lý danh mục 76](#_Toc215571812)

**DANH MỤC BẢNG BIỂU**

[Bảng 1 Danh sách các Actor 17](#_Toc215571818)

[Bảng 2 Danh sách các Usecase 18](#_Toc215571819)

[Bảng 3 Mô tả usecase tổng quát 24](#_Toc215571820)

[Bảng 4 Quy trình đăng ký người dùng 33](#_Toc215571821)

[Bảng 5 Quy trình đăng nhập người dùng 35](#_Toc215571822)

[Bảng 6 Quy trình thêm thu nhập 36](#_Toc215571823)

[Bảng 7 Quy trình thêm khoản chi 38](#_Toc215571824)

[Bảng 8 Quy trình xem báo cáo 39](#_Toc215571825)

[Bảng 9 Quy trình quản lý ví 40](#_Toc215571826)

[Bảng 10 Quản lý danh mục 42](#_Toc215571827)

[Bảng 11 Bảng Users 46](#_Toc215571828)

[Bảng 12 Bảng Categories 46](#_Toc215571829)

[Bảng 13 Bảng Expenses 47](#_Toc215571830)

[Bảng 14 Bảng Incomes 47](#_Toc215571831)

[Bảng 15 Bảng Budgets 48](#_Toc215571832)

[Bảng 16 Bảng Recurring\_transactions 48](#_Toc215571833)

[Bảng 17 Bảng Goals 49](#_Toc215571834)

[Bảng 18 Bảng Reports 49](#_Toc215571835)

[Bảng 19 Bảng Notifications 50](#_Toc215571836)

[Bảng 20 Bảng Settings 51](#_Toc215571837)

**DANH SÁCH CÁC CỤM TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **TỪ VIẾT TẮT/ THUẬT NGỮ** | **Ý NGHĨA** |
| **API** | **Application Programming Interface** |
| **REST** | **Representational State Transfer** |
| **CRUD** | **Create – Read – Update – Delete** |
| **JWT** | **JSON Web Token** |
| **JSON** | **JavaScript Object Notation** |
| **UI** | **User Interface** |
| **UX** | **User Experience** |
| **DB** | **Database** |
| **MySQL** | **Hệ quản trị cơ sở dữ liệu MySQL** |
| **IDE** | **Integrated Development Environment** |
| **OOP** | **Object-Oriented Programming** |
| **MVC/MVVM** | **Các mô hình kiến trúc phần mềm** |
| **API Test (Postman)** | **Các mô hình kiến trúc phần mềm** |
| **Frontend** | **Giao diện người dùng** |
| **Backend** | **Xử lý logic phía server** |
| **Flutter** | **Bộ công cụ phát triển ứng dụng đa nền tảng** |
| **Dart** | **Ngôn ngữ lập trình dùng cho Flutter** |
| **Spring Boot** | **Framework xây dựng backend Java** |
| **CSDL** | **Cơ sở dữ liệu** |

# LỜI MỞ ĐẦU

Đề tài “Xây dựng ứng dụng báo cáo chi tiêu” được thực hiện nhằm nghiên cứu và phát triển một giải pháp hỗ trợ người dùng quản lý tài chính cá nhân một cách trực quan, khoa học và hiệu quả. Cơ sở khoa học của đề tài dựa trên các nguyên lý quản lý dữ liệu, phân tích tổng hợp theo thời gian và ứng dụng công nghệ phần mềm trong việc tự động hóa quá trình ghi nhận thu – chi.

  Về tính thực tiễn, nhu cầu kiểm soát dòng tiền và quản lý chi tiêu cá nhân ngày càng tăng trong bối cảnh cuộc sống hiện đại dẫn đến khối lượng giao dịch phát sinh nhiều và đa dạng. Việc ghi chép thủ công dễ gây thất lạc dữ liệu, khó thống kê và không mang lại cái nhìn tổng thể về tình hình tài chính. Do đó, việc xây dựng một ứng dụng hỗ trợ quản lý thu chi là rất cần thiết, giúp người dùng tiết kiệm thời gian, theo dõi chi tiêu chính xác và đưa ra các quyết định tài chính hợp lý hơn.

  Mục tiêu chính của đề tài là xây dựng một ứng dụng di động cho phép người dùng ghi nhận các khoản thu nhập và chi tiêu hằng ngày, quản lý ví tiền, phân loại giao dịch theo danh mục, theo dõi số dư và xem báo cáo thống kê bằng biểu đồ theo nhiều khoảng thời gian. Bên cạnh đó, đề tài còn hướng đến việc đảm bảo tính an toàn dữ liệu và mang lại giao diện thân thiện, phù hợp với thói quen sử dụng của người dùng.

  Về phương pháp nghiên cứu, đề tài áp dụng mô hình phát triển phần mềm theo hướng phân tích yêu cầu, thiết kế hệ thống, xây dựng và kiểm thử ứng dụng. Các công nghệ sử dụng bao gồm Flutter để phát triển giao diện người dùng, Spring Boot cho backend xử lý nghiệp vụ và MySQL làm hệ quản trị cơ sở dữ liệu. Cơ chế xác thực bằng JWT được áp dụng nhằm tăng cường bảo mật. Toàn bộ các API được kiểm thử bằng Postman trước khi tích hợp nhằm bảo đảm tính chính xác và ổn định.

  Nội dung công việc triển khai theo từng bước nghiên cứu tương ứng, bao gồm khảo sát và phân tích yêu cầu, đề xuất mô hình hệ thống, thiết kế giao diện và cơ sở dữ liệu, xây dựng chức năng frontend – backend, kết nối dữ liệu, kiểm thử và hoàn thiện ứng dụng.

  Đối tượng nghiên cứu là người dùng cá nhân có nhu cầu quản lý thu và chi hằng ngày. Phạm vi nghiên cứu tập trung vào các chức năng cốt lõi như ghi nhận giao dịch, quản lý ví tiền, phân loại danh mục và thống kê báo cáo. Đề tài được xây dựng theo hướng tự nghiên cứu và phát triển, không thuộc phạm vi triển khai cho một doanh nghiệp cụ thể.

  Quyển báo cáo này trình bày đầy đủ các nội dung liên quan đến quá trình xây dựng ứng dụng, bao gồm cơ sở lý thuyết, phân tích và thiết kế hệ thống, thực thi các chức năng, kiểm thử và đánh giá kết quả, đồng thời đề xuất các hướng phát triển nhằm hoàn thiện ứng dụng trong tương lai.

# TỔNG QUAN TÀI LIỆU

## 1.1. Giới thiệu đề tài

### 1.1.1. Nhu cầu thực tế của đề tài

Trong bối cảnh đời sống hiện đại với nhiều khoản chi tiêu phức tạp, việc quản lý tài chính cá nhân trở thành nhu cầu thiết yếu đối với mỗi người. Tuy nhiên, phần lớn người dùng vẫn ghi chép thu – chi theo cách thủ công hoặc không theo dõi thường xuyên, dẫn đến khó khăn trong việc kiểm soát dòng tiền và thiếu cơ sở để xây dựng kế hoạch tài chính dài hạn. Sự phát triển mạnh mẽ của công nghệ di động đã tạo điều kiện thuận lợi để tự động hóa quá trình ghi nhận và phân tích dữ liệu tài chính. Một ứng dụng báo cáo chi tiêu có khả năng lưu trữ, phân loại giao dịch và thống kê trực quan theo các mốc thời gian khác nhau sẽ giúp người dùng hiểu rõ thói quen chi tiêu của mình, từ đó đưa ra các quyết định tài chính hợp lý và khoa học hơn.

Chính nhu cầu thực tế này trở thành động lực quan trọng cho việc nghiên cứu và triển khai đề tài.

### 1.1.2. Lý do chọn đề tài

Việc quản lý tài chính cá nhân từ lâu đã là vấn đề được quan tâm, nhưng trong thực tế, nhiều người vẫn chưa có thói quen theo dõi thu chi một cách khoa học. Việc ghi chép dựa trên cảm tính dẫn đến tình trạng khó kiểm soát chi tiêu, dễ phát sinh các khoản ngoài dự kiến và không có dữ liệu để điều chỉnh thói quen tài chính. Bên cạnh đó, các ứng dụng hiện có đôi khi chứa quá nhiều chức năng phức tạp, gây khó khăn cho người mới bắt đầu.

Chính vì vậy, việc xây dựng một ứng dụng báo cáo chi tiêu đơn giản, trực quan nhưng vẫn đảm bảo đầy đủ chức năng cần thiết là một nhu cầu phù hợp với thực tiễn. Thông qua quá trình thực hiện đề tài, em cũng có cơ hội nâng cao kỹ năng lập trình, tiếp cận mô hình phát triển ứng dụng thực tế và hiểu rõ hơn về quy trình xây dựng một hệ thống phần mềm hoàn chỉnh, từ đó góp phần củng cố kiến thức chuyên môn và mở rộng khả năng ứng dụng trong tương lai.

### 1.1.3. Mục tiêu đề tài

Mục tiêu tổng quát của đề tài là xây dựng một ứng dụng báo cáo chi tiêu hỗ trợ người dùng quản lý thu nhập và chi tiêu cá nhân một cách trực quan, thuận tiện và hiệu quả. Ở mức độ cụ thể, đề tài hướng đến việc cung cấp giao diện thân thiện, dễ sử dụng; hỗ trợ ghi nhận thu – chi nhanh chóng; phân loại giao dịch theo từng danh mục; quản lý nhiều ví tiền; và hiển thị báo cáo thống kê bằng biểu đồ theo nhiều khoảng thời gian. Bên cạnh đó, hệ thống phải đảm bảo tính chính xác trong việc lưu trữ và truy xuất dữ liệu, đồng thời được thiết kế theo hướng dễ mở rộng để có thể phát triển thêm các chức năng trong tương lai nhằm đáp ứng nhu cầu ngày càng cao của người dùng.

**Đối tượng nghiên cứu:**

Đối tượng nghiên cứu của đề tài là người dùng cá nhân có nhu cầu theo dõi và quản lý tài chính hằng ngày, bao gồm các hoạt động như ghi nhận thu nhập, chi tiêu, quản lý ví tiền và xem báo cáo thống kê. Đây là nhóm người dùng mong muốn một công cụ hỗ trợ đơn giản, trực quan nhưng vẫn đáp ứng đầy đủ các chức năng cốt lõi giúp họ kiểm soát chi tiêu và xây dựng thói quen tài chính khoa học hơn.

**Phạm vi nghiên cứu:**

Phạm vi nghiên cứu của đề tài tập trung vào các kiến thức và kỹ thuật liên quan đến việc xây dựng ứng dụng di động theo mô hình client–server. Đề tài bao gồm việc tìm hiểu và áp dụng các công nghệ như Flutter cho giao diện người dùng, Spring Boot cho backend và MySQL cho hệ quản trị cơ sở dữ liệu. Các chức năng được triển khai trong phạm vi này gồm ghi nhận thu – chi, phân loại giao dịch, quản lý nhiều ví tiền và hiển thị báo cáo thống kê theo thời gian. Đề tài chỉ tập trung vào quản lý tài chính cá nhân ở mức cơ bản, không tích hợp với hệ thống ngân hàng hoặc thanh toán trực tuyến và được xây dựng với mục đích nghiên cứu cá nhân, không triển khai cho doanh nghiệp cụ thể.

### 1.1.5. Phương pháp nghiên cứu

Phương pháp nghiên cứu của đề tài được xây dựng dựa trên quy trình phát triển phần mềm, bao gồm các bước khảo sát, phân tích, thiết kế, xây dựng và kiểm thử hệ thống. Trước tiên, đề tài tiến hành thu thập và phân tích nhu cầu thực tế của người dùng thông qua khảo sát các ứng dụng quản lý chi tiêu hiện có và tổng hợp các yêu cầu phổ biến trong quản lý tài chính cá nhân. Trên cơ sở đó, hệ thống được thiết kế theo mô hình client–server, trong đó Flutter được sử dụng để xây dựng giao diện người dùng và Spring Boot đảm nhiệm phần xử lý nghiệp vụ trên backend. MySQL được lựa chọn làm hệ quản trị cơ sở dữ liệu nhằm đảm bảo khả năng lưu trữ và truy xuất dữ liệu ổn định. Trong quá trình xây dựng ứng dụng, các API được kiểm thử bằng Postman để đánh giá độ chính xác và tính ổn định trước khi tích hợp vào giao diện. Cuối cùng, ứng dụng được kiểm thử chức năng nhằm đảm bảo các thao tác ghi nhận thu – chi, quản lý ví và hiển thị báo cáo hoạt động đúng yêu cầu và mang lại trải nghiệm sử dụng tốt cho người dùng.

### 1.1.6. Nội dung nghiên cứu

Nội dung nghiên cứu của đề tài được triển khai dựa trên quá trình khảo sát, phân tích và xây dựng hệ thống theo mô hình phát triển phần mềm. Trước hết, đề tài tập trung tìm hiểu các ứng dụng quản lý chi tiêu hiện có để tổng hợp những ưu điểm, hạn chế và rút ra các yêu cầu phù hợp cho hệ thống cần xây dựng. Đồng thời, đề tài tiến hành nghiên cứu các kiến thức liên quan đến quản lý tài chính cá nhân như phương pháp phân loại thu – chi, mô hình ví tiền và các dạng báo cáo thống kê phổ biến. Bên cạnh đó, các công nghệ sử dụng trong quá trình phát triển cũng được tìm hiểu chi tiết, bao gồm Flutter cho giao diện người dùng, Spring Boot cho backend và MySQL cho hệ quản trị cơ sở dữ liệu.

Trên cơ sở những kiến thức đã thu thập, đề tài tiến hành phân tích hệ thống để xác định luồng xử lý chính, mô hình dữ liệu, giao diện người dùng và các chức năng cốt lõi của ứng dụng. Tiếp theo, quá trình lập trình được thực hiện nhằm xây dựng các chức năng quan trọng như ghi nhận thu – chi, quản lý ví, phân loại giao dịch, xem lịch sử hiển thị biểu đồ thống kê, tạo giao dịch định kỳ…. Cuối cùng, hệ thống được kiểm thử, đánh giá và điều chỉnh nhằm đảm bảo ứng dụng hoạt động ổn định, chính xác và đáp ứng đầy đủ yêu cầu đã đề ra.

### 1.1.7. Khảo sát đề tài

Trong quá trình khảo sát, đề tài đã tiến hành tìm hiểu một số ứng dụng quản lý chi tiêu phổ biến hiện nay như Money Lover, Sổ Thu Chi và Misa Money Keeper… Các ứng dụng này đều cung cấp những chức năng cơ bản như ghi thu – chi, quản lý ví, phân loại danh mục và hiển thị báo cáo thống kê. Tuy nhiên, qua quá trình sử dụng, có thể nhận thấy mỗi ứng dụng đều tồn tại một số hạn chế, chẳng hạn như giao diện phức tạp, nhiều tính năng dư thừa, một số chức năng khó thao tác đối với người dùng mới hoặc chưa đáp ứng tốt nhu cầu cá nhân hóa theo thói quen tài chính của từng người dùng.

Bên cạnh việc khảo sát các sản phẩm hiện có, đề tài cũng thực hiện thu thập nhu cầu thực tế của người dùng thông qua phản hồi và quan sát hành vi. Kết quả khảo sát cho thấy người dùng đặc biệt quan tâm đến khả năng nhập thu – chi nhanh chóng, giao diện đơn giản và dễ hiểu, hệ thống phân loại rõ ràng và báo cáo trực quan theo ngày, tuần và tháng. Ngoài ra, nhu cầu quản lý nhiều ví tiền cùng lúc và yêu cầu về bảo mật dữ liệu cá nhân cũng được đánh giá rất cao. Một số ứng dụng hiện tại được người dùng phản ánh là hoạt động chưa thật sự mượt mà, gây trở ngại trong quá trình ghi nhận giao dịch hằng ngày.

Từ quá trình khảo sát, có thể nhận định rằng một ứng dụng báo cáo chi tiêu cần tập trung vào sự đơn giản, dễ sử dụng, tốc độ xử lý nhanh và đảm bảo tính bảo mật dữ liệu. Đề tài xây dựng ứng dụng theo hướng tối ưu thao tác nhập liệu, tinh gọn các chức năng không cần thiết, đồng thời cung cấp các biểu đồ trực quan giúp người dùng dễ dàng theo dõi thói quen chi tiêu. Những kết quả khảo sát này đóng vai trò quan trọng trong việc xác định hướng phát triển và đề xuất các chức năng phù hợp cho hệ thống, đảm bảo rằng ứng dụng đáp ứng được nhu cầu thực tế của người dùng.

### 1.1.8. Phát biểu bài toán

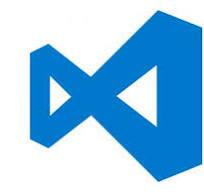
Từ nhu cầu thực tế trong việc quản lý tài chính cá nhân và kết quả khảo sát người dùng, bài toán đặt ra là xây dựng một ứng dụng báo cáo chi tiêu có khả năng hỗ trợ người dùng theo dõi thu nhập và chi tiêu một cách thuận tiện, khoa học và chính xác. Ứng dụng cần cung cấp giao diện thân thiện, dễ thao tác, cho phép người dùng nhanh chóng ghi nhận các khoản thu – chi hằng ngày, đồng thời hỗ trợ phân loại giao dịch theo từng nhóm để giúp việc quản lý trở nên rõ ràng hơn. Bên cạnh đó, hệ thống phải cho phép quản lý nhiều ví tiền khác nhau như ví tiền mặt, ví ngân hàng hoặc ví phụ, và cung cấp lịch sử giao dịch chi tiết theo từng mốc thời gian.

Ngoài việc hỗ trợ ghi nhận dữ liệu, ứng dụng còn phải cung cấp các báo cáo và biểu đồ thống kê trực quan nhằm giúp người dùng đánh giá thói quen chi tiêu và xây dựng kế hoạch tài chính hợp lý. Hệ thống cũng cần đảm bảo khả năng lưu trữ và xử lý dữ liệu ổn định, bảo mật thông tin người dùng và duy trì tính toàn vẹn của dữ liệu. Đồng thời, ứng dụng phải có khả năng mở rộng để phát triển thêm các chức năng nâng cao trong tương lai như đồng bộ dữ liệu đám mây, nhắc nhở thanh toán hoặc gợi ý chi tiêu thông minh. Với các yêu cầu đó, bài toán hướng tới việc xây dựng một ứng dụng tiện lợi, an toàn và hữu ích cho người dùng trong việc quản lý chi tiêu cá nhân.

## 1.2. Cơ sở lý thuyết

### 1.2.1. VSCode (Visual Studio Code)

Visual Studio Code là trình soạn thảo mã nguồn miễn phí do Microsoft phát triển, hỗ trợ đa nền tảng và phù hợp cho nhiều loại dự án khác nhau. Đây là công cụ chính được sử dụng trong quá trình phát triển ứng dụng báo cáo chi tiêu.



Hình 1. 1 Logo vscode

**Công dụng của VSCode:**

* Dùng để viết, chỉnh sửa và quản lý mã nguồn của dự án Flutter và Spring Boot.
* Hỗ trợ chạy lệnh và thao tác trực tiếp thông qua Terminal tích hợp.
* Cho phép quản lý file, thư mục và cấu trúc dự án một cách trực quan.

**Các chức năng chính:**

* Hỗ trợ nhiều ngôn ngữ lập trình: Dart, Java, JavaScript, C/C++,…
* Kho tiện ích mở rộng (Extensions) phong phú như Flutter, Dart, Spring Tools, MySQL,…
* IntelliSense: gợi ý code, nhắc cú pháp và phát hiện lỗi trong quá trình nhập liệu.
* Tích hợp Git/GitHub giúp theo dõi thay đổi, commit và đồng bộ mã nguồn.
* Debug hỗ trợ nhiều ngôn ngữ, giúp sửa lỗi nhanh và trực quan.
* Mở nhiều file cùng lúc (màn hình đa nhiệm).

### 1.2.2. Ngôn ngữ Dart

Dart là ngôn ngữ lập trình do Google phát triển, được thiết kế tối ưu cho việc xây dựng giao diện người dùng trên nhiều nền tảng khác nhau. Đây cũng là ngôn ngữ chính được sử dụng trong Flutter, giúp tạo ra các ứng dụng di động có hiệu năng cao và giao diện mượt mà. Dart hỗ trợ lập trình hướng đối tượng, cú pháp rõ ràng, dễ đọc và dễ tiếp cận đối với người mới. Bên cạnh đó, Dart cung cấp cơ chế biên dịch linh hoạt, vừa hỗ trợ JIT trong quá trình phát triển để tăng tốc độ chạy thử.

A blue diamond shaped object

AI-generated content may be incorrect.

Hình 1. 2 Logo ngôn ngữ Dart

Nhờ sự kết hợp giữa tốc độ, tính linh hoạt và khả năng tích hợp tốt với Flutter, Dart trở thành lựa chọn phù hợp cho việc xây dựng ứng dụng báo cáo chi tiêu trên thiết bị di động, đảm bảo tính ổn định, dễ bảo trì và dễ mở rộng trong tương lai.

A diagram of a diagram

AI-generated content may be incorrect.

Hình 1.3 Ứng dụng của ngôn ngữ Dart.

Dart được ứng dụng trong nhiều lĩnh vực:

* Phát triển ứng dụng di động đa nền tảng thông qua Flutter.
* Xây dựng ứng dụng Web nhờ khả năng biên dịch sang JavaScript.
* Phát triển ứng dụng phía server với thư viện dart:io và các framework hỗ trợ.
* Thiết kế giao diện người dùng với hệ thống widget mạnh mẽ của Flutter.

### 1.2.3. Flutter

Flutter là bộ SDK (Software Development Kit) do Google phát triển, dùng để xây dựng giao diện người dùng cho ứng dụng đa nền tảng. Flutter cho phép lập trình viên tạo ứng dụng chạy trên Android, iOS, Web và Desktop chỉ với một bộ mã nguồn duy nhất. Flutter sử dụng ngôn ngữ Dart và cung cấp hệ thống widget mạnh mẽ giúp tạo ra giao diện đẹp, mượt mà và nhất quán.

A blue and white logo

AI-generated content may be incorrect.

Hình 1. 4 Logo flutter

#### Ứng dụng của Flutter:

* Phát triển ứng dụng di động đa nền tảng với hiệu năng cao, hoạt động mượt mà trên cả Android và iOS.
* Xây dựng giao diện người dùng (UI) tùy chỉnh thông qua hệ thống widget phong phú và khả năng thiết kế linh hoạt.
* Tối ưu hóa thời gian phát triển nhờ tính năng Hot Reload cho phép xem kết quả chỉnh sửa ngay lập tức.
* Hỗ trợ phát triển ứng dụng Web và Desktop, mở rộng phạm vi sử dụng của cùng một mã nguồn.
* Tích hợp dễ dàng với API backend, phù hợp cho các ứng dụng cần giao tiếp dữ liệu thường xuyên như app báo cáo chi tiêu.

Flutter là nền tảng phù hợp để xây dựng ứng dụng báo cáo chi tiêu nhờ khả năng phát triển nhanh, giao diện đẹp và hiệu năng ổn định trên nhiều thiết bị.

### 1.2.4 Java

Java là ngôn ngữ lập trình hướng đối tượng phổ biến, được sử dụng rộng rãi trong phát triển ứng dụng doanh nghiệp, web và hệ thống backend. Java hoạt động độc lập với nền tảng nhờ cơ chế “Write Once, Run Anywhere”, giúp các ứng dụng có tính ổn định cao và dễ triển khai trên nhiều môi trường khác nhau. Trong đề tài này, Java được sử dụng để xây dựng backend với Spring Boot.



Hình 1. 5 Logo Java

**Ứng dụng của Java:**

* Phát triển backend cho các hệ thống web và ứng dụng di động.
* Xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu thông qua API.
* Xây dựng các dịch vụ RESTful để kết nối giữa ứng dụng và server.
* Phát triển các ứng dụng doanh nghiệp yêu cầu độ ổn định và khả năng mở rộng cao.
* Tương tác với MySQL, PostgreSQL,... thông qua JDBC hoặc các thư viện ORM.

### 1.2.5. Spring Boot

Spring Boot là một framework mạnh mẽ của Java, được thiết kế để đơn giản hóa quá trình xây dựng ứng dụng backend. Với khả năng cấu hình tự động, hệ thống thư viện phong phú và cơ chế quản lý phụ thuộc hiệu quả, Spring Boot giúp lập trình viên phát triển các dịch vụ web và API RESTful nhanh chóng, ổn định và dễ bảo trì. Đây là nền tảng backend được sử dụng trong ứng dụng báo cáo chi tiêu để xử lý dữ liệu và giao tiếp với hệ thống.



Hình 1. 6 Logo Spring Boot

**Ứng dụng của Spring Boot :**

* Xây dựng API RESTful phục vụ cho việc giao tiếp giữa ứng dụng Flutter và máy chủ.
* Xử lý logic nghiệp vụ, bao gồm quản lý người dùng, ví tiền, thu – chi và thống kê.
* Kết nối và tương tác với cơ sở dữ liệu MySQL thông qua JPA/Hibernate.
* Quản lý bảo mật với cơ chế xác thực người dùng, đặc biệt là JWT.
* Tạo cấu trúc backend ổn định, dễ mở rộng khi thêm chức năng mới.

Nhờ khả năng tự động cấu hình, bảo mật tốt và hỗ trợ mở rộng linh hoạt, Spring Boot là lựa chọn phù hợp để xây dựng backend cho ứng dụng báo cáo chi tiêu.

### 1.2.6 MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ phổ biến, được sử dụng rộng rãi nhờ tính ổn định, tốc độ truy vấn nhanh và dễ triển khai. MySQL hoạt động dựa trên mô hình dữ liệu có cấu trúc gồm bảng, cột và hàng, giúp việc lưu trữ và xử lý dữ liệu trở nên rõ ràng và khoa học. Trong ứng dụng báo cáo chi tiêu, MySQL được dùng để lưu trữ toàn bộ dữ liệu giao dịch, ví tiền và thông tin người dùng.



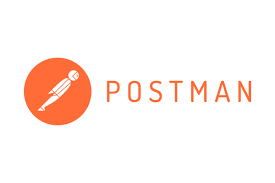
Hình 1. 7 Logo MySQL

**Ứng dụng của MySQL:**

* Lưu trữ dữ liệu thu – chi, thông tin ví tiền, người dùng và danh mục giao dịch.
* Hỗ trợ truy vấn dữ liệu nhanh, phục vụ chức năng thống kê theo ngày, tuần, tháng.
* Đảm bảo tính toàn vẹn dữ liệu, tránh thất thoát hoặc trùng lặp thông tin trong quá trình xử lý.
* Kết nối trực tiếp với Spring Boot, dễ dàng thao tác thông qua JPA/Hibernate.
* Hỗ trợ mở rộng dữ liệu khi số lượng giao dịch tăng theo thời gian.

### 1.2.7 Postman

Postman là công cụ hỗ trợ kiểm thử API phổ biến, được sử dụng rộng rãi trong quá trình phát triển phần mềm. Công cụ này cho phép gửi các yêu cầu HTTP như GET, POST, PUT, DELETE… đến server và quan sát kết quả trả về. Nhờ giao diện dễ sử dụng và khả năng tùy chỉnh linh hoạt, Postman giúp lập trình viên kiểm tra API nhanh chóng, chính xác trước khi tích hợp vào ứng dụng.



Hình 1. 8 Logo ứng dụng Postman

**Ứng dụng của Postman:**

* Kiểm thử API RESTful của Spring Boot, đảm bảo dữ liệu trả về đúng định dạng và hoạt động ổn định.
* Giả lập dữ liệu đầu vào để kiểm tra các chức năng ghi thu – chi, đăng nhập, đăng ký và thống kê.
* Phát hiện lỗi backend nhanh chóng thông qua phản hồi HTTP và thông báo lỗi trả về từ server.
* Tối ưu API trước khi kết nối với Flutter, giúp quá trình phát triển frontend dễ dàng và tránh lỗi phát sinh.
* Lưu trữ bộ sưu tập API để dễ dàng kiểm thử lại khi cần cập nhật hoặc mở rộng chức năng.

Postman giúp đảm bảo backend hoạt động chính xác và ổn định trước khi tích hợp vào ứng dụng báo cáo chi tiêu, rút ngắn thời gian sửa lỗi và nâng cao chất lượng hệ thống.

### 1.2.8 Cloudinary

Cloudinary là nền tảng lưu trữ và quản lý hình ảnh trên đám mây, cung cấp các API giúp tải lên, tối ưu hóa và phân phối hình ảnh nhanh chóng. Đây là giải pháp phổ biến cho các ứng dụng cần xử lý hoặc lưu trữ hình ảnh nhưng không muốn lưu trực tiếp trên server để tránh tăng dung lượng và giảm hiệu năng.



Hình 1. 9 Logo ứng dụng Cloudinary

**Ứng dụng của Cloudinary:**

* Lưu trữ hình ảnh trên đám mây, giảm tải dung lượng cho server backend.
* Tối ưu hóa hình ảnh tự động, giúp giảm kích thước nhưng vẫn đảm bảo chất lượng.
* Tạo URL ảnh trực tiếp để hiển thị trong ứng dụng mà không cần tải xuống máy chủ.
* Hỗ trợ thao tác upload nhanh, thuận tiện khi người dùng cần thêm hình ảnh ví hoặc hình đại diện (nếu có sử dụng trong ứng dụng).
* Dễ dàng tích hợp với Spring Boot thông qua API REST.

Với khả năng tối ưu hóa và quản lý hình ảnh hiệu quả, Cloudinary giúp ứng dụng báo cáo chi tiêu hoạt động ổn định hơn khi cần xử lý nội dung hình ảnh.

### 1.2.9 Git / GitHub

GitHub là nền tảng lưu trữ và quản lý mã nguồn dựa trên hệ thống Git, hỗ trợ theo dõi phiên bản và cộng tác trong quá trình phát triển phần mềm. Đây là công cụ quan trọng giúp lập trình viên quản lý dự án một cách khoa học, đảm bảo an toàn dữ liệu và dễ dàng phối hợp khi làm việc nhóm hoặc phát triển dài hạn.



Hình 1. 10 Logo ứng dụng GitHub

* Lưu trữ toàn bộ mã nguồn của frontend (Flutter) và backend (Spring Boot).
* Quản lý phiên bản mã nguồn, giúp dễ dàng theo dõi thay đổi và phục hồi khi cần thiết.
* Hỗ trợ làm việc trên nhiều thiết bị, đảm bảo không mất dữ liệu trong quá trình phát triển.
* Tạo môi trường khoa học để tổ chức mã nguồn theo từng nhánh và hợp nhất (merge) khi hoàn thành chức năng.
* Hỗ trợ kiểm tra, rà soát mã nguồn thông qua commit history.
* Tích hợp tốt với VSCode và IntelliJ IDEA, giúp đẩy/pull code nhanh chóng.

### 1.2.5. Kiến trúc của app báo cáo chi tiêu

#### 1.2.5.1. Frontend (Giao diện người dùng)

Frontend là phần giao diện người dùng (UI) mà khách hàng tương tác trực tiếp, bao gồm các thành phần sau:

* **Màn hình Danh Mục:** cho phép người dùng lựa chọn các danh mục phục vụ cho việc thêm giao dịch. Danh mục được chia thành hai nhóm riêng biệt gồm **danh mục chi tiêu** và **danh mục thu nhập**. Giao diện được thiết kế dạng lưới (grid) với biểu tượng minh họa rõ ràng, dễ nhìn, kết hợp cùng thanh điều hướng và nút mở menu nhanh giúp người dùng thao tác thuận tiện.
* **Menu Chức Năng:** được hiển thị dưới dạng bảng nổi, cho phép người dùng truy cập nhanh đến các chức năng quan trọng của ứng dụng như: thống kê, báo cáo tổng hợp, AI dự đoán chi tiêu, giao dịch định kỳ, lịch sử giao dịch, ví, mục tiêu, ngân sách. Menu được thiết kế đơn giản, dễ thao tác và tạo trải nghiệm nhất quán xuyên suốt ứng dụng
* **Trang Cài Đặt :** cho phép người dùng tùy chỉnh các thông số cá nhân trong ứng dụng, bao gồm: ngôn ngữ hiển thị, đơn vị tiền tệ, bật/tắt thông báo, hiện số dư ở trang chủ, tự động sao lưu, cài đặt chu kỳ nhắc nhở (hàng tuần, hàng tháng…), chọn giờ nhắc nhở. Thiết kế thân thiện, nhấn mạnh vào khả năng tùy chỉnh linh hoạt.
* **Trang Hồ Sơ Cá Nhân (Profile):** trang này cho phép: xem ảnh đại diện, họ tên, email, chỉnh sửa thông tin cá nhân, cập nhật ảnh đại diện, đổi mật khẩu.Giao diện hiện đại, dễ thao tác với ảnh và thông tin người dùng.
* **Trang Thêm Giao Dịch:** hỗtrợ người dùng thêm: khoản chi, khoản thu. Cho phép chọn: danh mục, số tiền, ngày, ghi chú, ví. Giao diện đơn giản, tối ưu thao tác nhập liệu nhanh.

#### 1.2.5.2. Backend (Phía máy chủ)

Backend là phần xử lý dữ liệu, nghiệp vụ và bảo mật của ứng dụng quản lý chi tiêu. Phía máy chủ được xây dựng bằng **Spring Boot**, sử dụng **JWT Authentication** và cơ sở dữ liệu **MySQL**. Các chức năng chính gồm:

* **Quản Lý Người Dùng**: **Quản Lý Người Dùng:** cung cấp API cho phép đăng ký, đăng nhập, xác thực người dùng bằng JWT, lấy thông tin tài khoản, cập nhật hồ sơ cá nhân, đổi mật khẩu và tải ảnh đại diện. Toàn bộ mật khẩu được mã hóa bằng PasswordEncoder nhằm đảm bảo an toàn thông tin người dùng.

**- Quản Lý Giao Dịch**: cho phép lấy danh sách danh mục thu – chi, phân nhóm theo từng loại, và hỗ trợ thêm, sửa, xóa danh mục Danh mục lưu trữ icon và tên, phục vụ việc thêm giao dịch từ phía Frontend.

* **Quản Lý Danh Mục**: cho phép lấy danh sách danh mục thu – chi, phân nhóm theo từng loại, và hỗ trợ thêm, sửa, xóa danh mục (nếu có phân quyền). Danh mục lưu trữ icon và tên, phục vụ việc thêm giao dịch từ phía Frontend
* **Quản Lý Ví (Wallet)**: xử lý việc tạo ví mới, chỉnh sửa thông tin ví, xóa ví, xem chi tiết số dư, đồng bộ số dư khi phát sinh giao dịch và quản lý nhiều ví của người dùng, giúp theo dõi tài chính chính xác.
* **Quản Lý Ngân Sách (Budget)**: cho phép tạo ngân sách theo từng danh mục, theo dõi tiến độ chi tiêu, hiển thị cảnh báo khi gần vượt ngân sách và hỗ trợ chỉnh sửa hoặc xóa ngân sách theo nhu cầu người dùng.
* **Thống Kê & Báo Cáo**: cung cấp API trả về dữ liệu tổng thu – tổng chi theo tháng, chi tiêu theo từng danh mục, lịch sử chi tiêu theo dòng thời gian, báo cáo định kỳ và so sánh thống kê giữa các tháng.
* **Giao Dịch Định Kỳ (Recurring Transactions)**: xử lý việc tự động sinh giao dịch theo chu kỳ ngày, tuần, tháng hoặc năm; hệ thống tự kiểm tra thời điểm đến hạn và tạo giao dịch tương ứng.

**- Quản Lý Cài Đặt Người Dùng**: lưu trữ và xử lý các thiết lập cá nhân như ngôn ngữ ứng dụng, đơn vị tiền tệ, bật/tắt thông báo, chế độ sao lưu tự động, chu kỳ nhắc nhở và thời gian nhắc nhở, đảm bảo trải nghiệm tùy chỉnh theo nhu cầu mỗi người dùng.

* **AI Dự Đoán Chi Tiêu:** phân tích dữ liệu thu – chi của người dùng và trả về dự đoán chi tiêu trong tháng, xác định danh mục có xu hướng tăng chi tiêu và cung cấp dữ liệu thống kê để Frontend hiển thị biểu đồ dự đoán.
* **AI Gợi Ý Danh Mục:** phân tích nội dung giao dịch dựa trên ghi chú, số tiền và thói quen chi tiêu trước đó của người dùng để gợi ý danh mục phù hợp khi thêm giao dịch mới. Tính năng này giúp giảm thao tác lựa chọn danh mục và tăng độ chính xác trong phân loại giao dịch.
* **Xuất File Excel:** hỗ trợ sinh báo cáo tài chính dưới dạng file Excel (.xlsx) dựa trên dữ liệu giao dịch; backend tạo file thông qua thư viện xử lý Excel và gửi về Frontend để người dùng tải xuống nhanh chóng và thuận tiện.

# PHƯƠNG PHÁP THỰC HIỆN

## 2.1. Đặc tả yêu cầu

### 2.1.1. Danh sách các Actor

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa / Ghi chú** |
| **1** | **Người dùng** | Là người trực tiếp sử dụng ứng dụng để quản lý thu – chi. |
| **2** | **Admin** | Là quản trị viên của hệ thống, có quyền quản lý người dùng và quản lý danh mục thông qua trang web quản trị. |

Bảng 1 Danh sách các Actor

### 2.1.2. Danh sách các Usecase

|  |  |  |  |
| --- | --- | --- | --- |
| **Quyền hệ thống** | **STT** | **Tên Use Case** | **Ý nghĩa / Ghi chú** |
| **Người dùng** | 1 | Đăng ký tài khoản | Tạo tài khoản người dùng mới để bắt đầu sử dụng hệ thống. |
|  | 2 | Đăng nhập | Xác thực người dùng để truy cập ứng dụng. |
|  | 3 | Đăng xuất | Thoát khỏi ứng dụng và kết thúc phiên. |
|  | 4 | Thêm giao dịch | Ghi nhận giao dịch thu hoặc chi mới. |
|  | 5 | Sửa giao dịch | Chỉnh sửa thông tin của giao dịch đã tạo. |
|  | 6 | Xóa giao dịch | Xóa giao dịch không còn sử dụng. |
|  | 7 | Xem danh sách giao dịch | Hiển thị giao dịch theo ngày, tuần hoặc tháng. |
|  | 8 | Lọc giao dịch | Lọc theo danh mục, ví hoặc thời gian. |
|  | 9 | Xem danh mục | Hiển thị danh mục thu – chi có sẵn. |
|  | 10 | Quản lý ví | Thêm, sửa, xóa ví v xem số dư. |
|  | 11 | Quản lý ngân sách | Tạo hoặc chỉnh sửa ngân sách theo danh mục. |
|  | 12 | Quản lý mục tiêu | Tạo, sửa hoặc xóa mục tiêu tài chính. |
|  | 13 | Quản lý giao dịch định kỳ | Tạo hoặc chỉnh sửa giao dịch lặp lại theo chu kỳ. |
|  | 14 | Xem thống kê | Hiển thị biểu đồ thu – chi theo thời gian. |
|  | 15 | Xem báo cáo tổng hợp | Tổng quan tài chính theo tháng hoặc danh mục. |
|  | 16 | Cập nhật thông tin cá nhân | Chỉnh sửa tên, email, ảnh đại diện,… |
|  | 17 | Đổi mật khẩu | Thay đổi mật khẩu để tăng bảo mật. |
|  | 18 | Xuất file Excel | Tải báo cáo giao dịch dưới dạng Excel. |
|  | 19 | AI gợi ý danh mục | Gợi ý danh mục khi nhập giao dịch. |
|  | 20 | AI dự đoán chi tiêu | Dự đoán chi tiêu tháng dựa trên lịch sử giao dịch. |
|  |  |  |  |
|  | 21 | Quản lý cài đặt | Tùy chỉnh ngôn ngữ, đơn vị tiền tệ, thông báo,… |
| **Admin** | 22 | Quản lý người dùng | Xem danh sách, sửa thông tin hoặc xóa người dùng. |
|  | 23 | Quản lý danh mục | Thêm, sửa hoặc xóa danh mục hệ thống. |

Bảng 2 Danh sách các Usecase

## 2.2. Phân tích và thiết kế hệ thống

### 2.2.1. Kiến trúc hệ thống

Hệ thống Ứng dụng báo cáo chi tiêu cá nhân được xây dựng dựa trên kiến trúc đa tầng (Multi–Layer Architecture) theo mô hình Client – Server. Toàn bộ dữ liệu và xử lý nghiệp vụ được tổ chức tách biệt giữa phía máy người dùng và phía máy chủ, đảm bảo khả năng mở rộng, tính ổn định và an toàn dữ liệu.

A diagram of a software process

AI-generated content may be incorrect.

Hình 2. 1 Sơ đồ tổng quan hệ thống

**Frontend (Ứng dụng di động Flutter)**

Phía Frontend là nơi người dùng trực tiếp thao tác thông qua ứng dụng di động được xây dựng bằng Flutter. Frontend gồm hai thành phần chính:

**UI/UX – Mobile App: Đây là giao diện người dùng của ứng dụng.**  
Nhiệm vụ:

- Hiển thị dữ liệu thu – chi, danh mục, ví, thống kê, biểu đồ

- Cho phép người dùng nhập giao dịch mới

- Tương tác với các chức năng như báo cáo, ngân sách, mục tiêu

- Gửi request đến Backend thông qua HTTP/REST

**AI Suggestion – Gợi ý danh mục: Tính năng AI phía client giúp:**

- Gợi ý danh mục phù hợp dựa trên mô tả giao dịch, số tiền, hành vi trước đó

- Giảm thao tác nhập dữ liệu cho người dùng

- Hoạt động kết hợp với Backend để phân tích lịch sử chi tiêu

Luồng hoạt động: Người dùng nhập giao dịch → Frontend gửi request → Backend phân tích → trả về danh mục gợi ý → hiển thị trên ứng dụng.

**Backend (Spring Boot)**

Backend là nơi xử lý logic nghiệp vụ và trao đổi dữ liệu với cơ sở dữ liệu. Các thành phần chính trong Backend:

**API Suggestion**

* Module cung cấp API gợi ý danh mục dựa trên AI.
* Nhận request từ Frontend và trả về danh mục gợi ý.

**AI Prediction – Dự đoán chi tiêu**

* Phân tích lịch sử giao dịch
* Dự đoán tổng chi tiêu tháng tiếp theo
* Phát hiện danh mục tiêu xài bất thường

**JWT Authentication**

* Xác thực người dùng khi đăng nhập
* Mã hóa token JWT
* Kiểm soát truy cập API
* Bảo vệ dữ liệu tài chính cá nhân

**Excel Export Engine**

* Nhận yêu cầu xuất file Excel từ người dùng
* Sinh file .xlsx dựa trên dữ liệu giao dịch
* Trả file về cho Frontend tải xuống

**Databases / Storage**

Hệ thống lưu trữ dữ liệu của ứng dụng.

* Người dùng
* Giao dịch thu – chi
* Danh mục
* Ví
* Ngân sách
* Giao dịch định kỳ
* Thống kê

**Cloudinary**

* Lưu ảnh đại diện người dùng
* Lưu icon hoặc hình ảnh nếu cần
* Lưu trữ trên đám mây để giảm tải cho server
* Backend gửi ảnh lên Cloudinary → nhận URL → lưu vào MySQL.

**Luồng giao tiếp**

* Mobile App gửi request (HTTP/REST)
* Backend xử lý request
* Backend trả JSON về cho app

**Backend → MySQL**

* Lưu thông tin mới (giao dịch / ngân sách / ví)
* Truy vấn dữ liệu để hiển thị lên app
* Cập nhật dữ liệu theo thời gian thực

**Backend → Cloudinary**

* Upload ảnh đại diện
* Nhận URL trả về
* Lưu url vào MySQL

### 2.2.2. Thiết kế hệ thống

#### 2.2.2.1. Sơ đồ UseCase

**1. Sơ đồ use case tổng quát**

A diagram with circles and text

AI-generated content may be incorrect.

Hình 2. 2 Sơ đồ usecase tổng quát

|  |  |
| --- | --- |
| **Tên use case** | **Use case tổng quát – User** |
| **Actor** | Người dùng (User) |
| **Mô tả** | Người dùng sau khi đăng nhập có thể thực hiện các chức năng quản lý tài chính cá nhân bao gồm: xem danh mục, quản lý thu/chi, ví, ngân sách, mục tiêu, giao dịch định kỳ, lịch sử giao dịch, thống kê & báo cáo, chỉnh sửa hồ sơ cá nhân và cài đặt hệ thống. |
| **Điều kiện tiên quyết** | - Người dùng đã đăng ký và đăng nhập thành công vào hệ thống. - Có kết nối Internet. - Có ít nhất một ví hoặc dữ liệu giao dịch (để sử dụng các chức năng quản lý). |
| **Luồng chính (Main Flow)** | **1.** User mở ứng dụng và đăng nhập hệ thống. **2.** Hệ thống xác thực tài khoản và chuyển đến giao diện chính. **3.** Người dùng có thể thực hiện các thao tác: • **Xem danh mục** – hiển thị các loại danh mục thu/chi. • **Quản lý thu/chi** – thêm, sửa, xóa các giao dịch thu chi cá nhân. • **Quản lý ví** – theo dõi các loại ví (tiền mặt, ngân hàng, ví điện tử). • **Quản lý ngân sách** – thiết lập và kiểm soát hạn mức chi tiêu. • **Quản lý mục tiêu** – đặt mục tiêu tài chính và theo dõi tiến độ. • **Giao dịch định kỳ** – tạo và quản lý các khoản thu/chi tự động lặp lại. • **Xem lịch sử giao dịch** – xem lại các giao dịch đã thực hiện. • **Thống kê & Báo cáo** – xem biểu đồ, thống kê chi tiêu. • **Hồ sơ cá nhân** – chỉnh sửa thông tin cá nhân. • **Cài đặt** – thay đổi ngôn ngữ, bảo mật, giao diện,… |
| **Luồng phụ (Alternative Flow)** | - Nếu sai thông tin đăng nhập → hiển thị thông báo lỗi. - Nếu không có dữ liệu → thông báo “Chưa có giao dịch nào”. - Nếu mất kết nối mạng → thông báo lỗi và yêu cầu thử lại. - Nếu nhập sai định dạng dữ liệu (ví dụ số tiền, email) → hiển thị cảnh báo. |

Bảng 3 Mô tả usecase tổng quát

#### 2.2.2.2. DFD (sơ đồ luồng dữ liệu)

**A diagram of a diagram

AI-generated content may be incorrect.**

Hình 2. 3 DFD Level 0 – Context Diagram

Sơ đồ DFD Level 0 hay còn gọi là Sơ đồ bối cảnh (Context Diagram) mô tả tổng quan hệ thống quản lý chi tiêu như một tiến trình duy nhất. Tại mức này, hệ thống chỉ thể hiện mối quan hệ trao đổi dữ liệu giữa tác nhân ngoài là người dùng và hệ thống. Các chức năng bên trong hệ thống cũng như các kho dữ liệu chưa được thể hiện ở mức này. Mục đích của Level 0 là giúp hình dung phạm vi hệ thống và các luồng thông tin chính ra vào hệ thống.

#### 2.2.2.3. Activity Diagram (Biểu đồ hoạt động)

**1. Biểu đồ hoạt động đăng ký**

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2. 4 Biểu đồ hoạt động đăng ký

**Luồng hoạt động**

Người dùng truy cập màn hình đăng ký và nhập email. Hệ thống kiểm tra xem email đã tồn tại hay chưa. Nếu email đã tồn tại, hệ thống thông báo lỗi và yêu cầu nhập lại. Nếu chưa, người dùng tiếp tục nhập mật khẩu, xác nhận mật khẩu và họ tên. Sau khi gửi yêu cầu đăng ký, hệ thống tạo tài khoản mới. Nếu đăng ký thành công, người dùng được chuyển về màn hình đăng nhập, ngược lại hệ thống hiển thị thông báo lỗi và yêu cầu thực hiện lại thao tác đăng ký.

**2. Biểu đồ hoạt động đăng nhập**

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 2. 5 Biểu đồ hoạt động đăng nhập

**Luồng hoạt động**

Người dùng truy cập vào màn hình đăng nhập, nhập email và mật khẩu. Hệ thống kiểm tra tính hợp lệ của thông tin đăng nhập. Nếu sai email hoặc mật khẩu, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại. Nếu thông tin chính xác, hệ thống xác thực đăng nhập thành công và chuyển người dùng đến màn hình chính của ứng dụng.

**3. Biểu đồ hoạt động thêm thu nhập**

A diagram with yellow squares and yellow text

AI-generated content may be incorrect.

Hình 2. 6 Biểu đồ hoạt động thêm thu nhập

**Luồng hoạt động**

Người dùng chọn chức năng “Thêm thu nhập” và nhập các thông tin cần thiết như số tiền, danh mục thu nhập, ngày thực hiện và ví muốn cộng tiền vào. Hệ thống kiểm tra tính hợp lệ của dữ liệu, ví dụ số tiền phải lớn hơn 0 và ví phải hợp lệ. Nếu dữ liệu không hợp lệ, hệ thống báo lỗi và yêu cầu nhập lại. Nếu hợp lệ, dữ liệu thu nhập được lưu vào cơ sở dữ liệu và số dư của ví được cộng thêm tương ứng. Sau đó người dùng được quay về màn hình chính.

**4. Biểu đồ hoạt động thêm chi tiêu**

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2. 7 Biểu đồ hoạt động thêm chi tiêu

**Luồng hoạt động**

Người dùng chọn chức năng “Thêm chi tiêu” và nhập các thông tin như số tiền, danh mục chi tiêu, ngày, ví sử dụng để thanh toán. Hệ thống kiểm tra số dư ví. Nếu số dư không đủ, hệ thống thông báo lỗi “Số dư không đủ” và cho phép người dùng điều chỉnh lại thông tin. Nếu đủ tiền, hệ thống lưu dữ liệu chi tiêu vào cơ sở dữ liệu và trừ số tiền tương ứng khỏi ví. Cuối cùng, người dùng được đưa về màn hình chính.

**5. Biểu đồ hoạt động xem báo cáo**

A diagram with text and arrows

AI-generated content may be incorrect.

Hình 2. 8 Biểu đồ hoạt động xem báo cáo

**Luồng hoạt động**

Người dùng chọn chức năng “Xem báo cáo”. Hệ thống lấy dữ liệu từ Income DB, Expense DB và Wallet DB. Sau đó hệ thống tổng hợp các thông số như tổng thu, tổng chi, số dư hiện tại và tạo các biểu đồ trực quan theo thời gian hoặc theo danh mục chi tiêu. Báo cáo được hiển thị để người dùng xem. Khi xem xong, người dùng có thể quay lại màn hình chính.

**6. Biểu đồ hoạt động quản lý ví**

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 2. 9 Biểu đồ hoạt động quản lý ví

**Luồng hoạt động**

Người dùng chọn “Quản lý ví” và được cung cấp các tùy chọn thêm, sửa hoặc xóa ví. Nếu thêm ví, người dùng nhập tên ví và số dư ban đầu rồi lưu vào Wallet DB. Nếu chỉnh sửa ví, người dùng chọn ví muốn sửa rồi cập nhật thông tin mới. Nếu xóa ví, người dùng chọn ví cần xóa và hệ thống sẽ loại bỏ ví đó khỏi cơ sở dữ liệu. Sau mỗi thao tác, người dùng quay lại màn hình chính.

**7. Biểu đồ hoạt động quản lý danh mục**

A diagram of a company

AI-generated content may be incorrect.

Hình 2. 10 Biểu đồ hoạt động quản lý danh mục

**Luồng hoạt động**

Người dùng chọn “Quản lý danh mục” và có thể thêm, chỉnh sửa hoặc xóa danh mục. Khi thêm, người dùng nhập tên danh mục và lưu lại. Khi chỉnh sửa, người dùng chọn danh mục hiện có và thay đổi tên hoặc biểu tượng. Khi xóa, người dùng chọn một danh mục và hệ thống sẽ xóa nó khỏi cơ sở dữ liệu. Sau khi hoàn thành thao tác, người dùng trở về màn hình chính.

#### 2.2.2.3. Sequence Diagram (Sơ đồ tuần tự)

**1. Biểu đồ tuần tự đăng ký**

A diagram with text and words

AI-generated content may be incorrect.

Sơ đồ 2.1 Biểu đồ tuần tự đăng ký

**Mô tả quy trình đăng ký người dùng**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| **1** | Người dùng → App | Người dùng chọn chức năng “Đăng ký”. |
| **2** | App → Người dùng | Ứng dụng hiển thị form nhập Email, Mật khẩu, Xác nhận mật khẩu và Họ tên. |
| **3** | Người dùng → App | Người dùng nhập thông tin đăng ký và nhấn “Đăng ký”. |
| **4** | App → Backend API | App gửi yêu cầu đăng ký (POST /api/auth/register) đến Backend. |
| **5** | Backend API → Database | Backend kiểm tra tính duy nhất của Email trong Database. |
| **6** | Database → Backend API | Database trả về kết quả email tồn tại hoặc không tồn tại. |
| **7a** | Backend API → App | **(Nhánh lỗi)** Backend trả về lỗi 409 Conflict nếu email đã được sử dụng. |
| **8a** | App → Người dùng | Ứng dụng hiển thị thông báo lỗi “Email đã tồn tại”, yêu cầu người dùng nhập lại. |
| **7b** | Backend API → Database | **(Nhánh hợp lệ)** Backend tạo tài khoản mới trong database. |

Bảng 4 Quy trình đăng ký người dùng

**2. Biểu đồ tuần tự đăng nhập**

A diagram of a computer

AI-generated content may be incorrect.

Sơ đồ 2.2 Biểu đồ tuần tự đăng nhập

**Mô tả quy trình đăng nhập người dùng**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| 1 | Người dùng → App | Chọn chức năng “Đăng nhập”. |
| 2 | App → Người dùng | Hiển thị form nhập Email và Mật khẩu. |
| 3 | Người dùng → App | Nhập thông tin đăng nhập. |
| 4 | App → Backend API | Gửi yêu cầu POST /api/auth/login. |
| 5 | Backend API → Database | Kiểm tra email. |
| 6 | Database → Backend API | Trả kết quả. |
| 7a | Backend API → App | (Email không tồn tại) Trả lỗi 404. |
| 8a | App → Người dùng | Thông báo “Email không tồn tại”. |
| 7b | Backend API → Database | (Email tồn tại) Kiểm tra mật khẩu. |

Bảng 5 Quy trình đăng nhập người dùng

**3. Biểu đồ tuần tự thêm khoản thu**

A diagram of a software project

AI-generated content may be incorrect.

Sơ đồ 2.3 Biểu đồ tuần tự thêm khoản thu

**Mô tả quy trình thêm thu nhập**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| 1 | Người dùng → App | Chọn chức năng “Thêm khoản thu/ thu nhập”. |
| 2 | App → Người dùng | Hiển thị form nhập Thu nhập. |
| 3 | Người dùng → App | Nhập thông tin thu nhập. |
| 4 | App → Backend API | Gửi yêu cầu POST /api/income. |
| 5 | Backend API → Database | Kiểm tra ví hợp lệ hay không. |
| 6 | Database → Backend API | Trả kết quả kiểm tra. |
| 7a | Backend API → App | (Ví không hợp lệ) Trả lỗi 400. |
| 8a | App → Người dùng | Hiển thị lỗi “Ví không tồn tại”. |
| 7b | Backend API → Database | Lưu thu nhập vào cơ sở dữ liệu. |
| 8b | Database → Backend API | Xác nhận lưu thành công. |
| 9b | Backend API → Database | Cập nhật số dư ví. |
| 10b | Database → Backend API | Xác nhận cập nhật thành công. |
| 11b | Backend API → App | Trả về 200 Created. |
| 12b | App → Người dùng | Điều hướng về màn hình chính. |

Bảng 6 Quy trình thêm thu nhập

**4. Biểu đồ tuần tự thêm chi tiêu**

**A diagram of a computer program

AI-generated content may be incorrect.**

Sơ đồ 2.4 Biểu đồ tuần tự thêm chi tiêu

**Mô tả quy trình thêm chi tiêu**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| 1 | Người dùng → App | Chọn “Thêm chi tiêu”. |
| 2 | App → Người dùng | Hiển thị form nhập Chi tiêu. |
| 3 | Người dùng → App | Nhập thông tin. |
| 4 | App → Backend API | Gửi yêu cầu POST /api/expense. |
| 5 | Backend API → Database | Kiểm tra số dư ví. |
| 6 | Database → Backend API | Trả số dư. |
| 7a | Backend API → App | (Không đủ tiền) Trả lỗi 400. |
| 8a | App → Người dùng | Hiển thị “Số dư không đủ”. |
| 7b | Backend API → Database | Lưu chi tiêu vào DB. |
| 8b | Database → Backend API | Lưu thành công. |
| 9b | Backend API → Database | Trừ số dư ví. |
| 10b | Database → Backend API | Cập nhật thành công. |
| 11b | Backend API → App | Trả 200 Created. |
| 12b | App → Người dùng | Điều hướng về màn hình chính. |

Bảng 7 Quy trình thêm khoản chi

**5. Biểu đồ tuần tự xem báo cáo**

A diagram with text and arrows

AI-generated content may be incorrect.

Sơ đồ 2.5 Biểu đồ tuần tự xem báo cáo

**Mô tả quy trình xem báo cáo**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| 1 | Người dùng → App | Chọn chức năng “Xem báo cáo”. |
| 2 | App → Backend API | Gửi yêu cầu GET /api/dashboard. |
| 3 | Backend API → Database | Lấy dữ liệu thu nhập. |
| 4 | Database → Backend API | Trả dữ liệu thu nhập. |
| 5 | Backend API → Database | Lấy dữ liệu chi tiêu. |
| 6 | Database → Backend API | Trả dữ liệu chi tiêu. |
| 7 | Backend API → App | Trả dữ liệu thống kê (Income, Expense, Balance, Charts). |
| 8 | App → Người dùng | Hiển thị báo cáo và biểu đồ. |

Bảng 8 Quy trình xem báo cáo

**6. Biểu đồ tuần tự quản lý ví**

A diagram of a number of objects

AI-generated content may be incorrect.

Sơ đồ 2.6 Biểu đồ tuần tự quản lý ví

**Mô tả quy trình quản lý ví**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| 1 | Người dùng → App | Chọn “Quản lý ví”. |
| 2 | App → Backend API | Gửi yêu cầu GET /api/wallets. |
| 3 | Backend API → Database | Lấy danh sách ví. |
| 4 | Database → Backend API | Trả danh sách ví. |
| 5 | Backend API → App | Hiển thị danh sách ví. |
| 6 | Người dùng → App | Chọn hành động (Thêm / Sửa / Xóa ví). |
| 7 | App → Backend API | Gửi yêu cầu tương ứng (POST/PUT/DELETE). |
| 8 | Backend API → Database | Xử lý lưu/chỉnh sửa/xóa dữ liệu. |
| 9 | Database → Backend API | Trả kết quả thao tác. |
| 10 | Backend API → App | Phản hồi kết quả. |

Bảng 9 Quy trình quản lý ví

**7. Biểu đồ tuần tự quản lý danh mục**

A diagram of a software project

AI-generated content may be incorrect.

Sơ đồ 2.7 Biểu đồ tuần tự quản lý danh mục

**Mô tả quy trình quản lý danh mục**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Thành phần tham gia** | **Mô tả luồng chính** |
| 1 | Người dùng → App | Chọn “Quản lý danh mục”. |
| 2 | App → Backend API | Gửi yêu cầu GET /api/categories. |
| 3 | Backend API → Database | Lấy danh sách danh mục. |
| 4 | Database → Backend API | Trả danh sách danh mục. |
| 5 | Backend API → App | Hiển thị danh mục. |
| 6 | Người dùng → App | Chọn Thêm/Sửa/Xóa danh mục. |
| 7 | App → Backend API | Gửi yêu cầu (POST/PUT/DELETE). |
| 8 | Backend API → Database | Lưu, chỉnh sửa hoặc xóa danh mục. |
| 9 | Database → Backend API | Trả kết quả thao tác. |
| 10 | Backend API → App | Trả phản hồi thành công. |
| 11 | App → Người dùng | Hiển thị và quay lại màn hình quản lý danh mục. |

Bảng 10 Quản lý danh mục

#### 2.2.2.4. Class Diagram (Sơ đồ lớp)

A diagram of a computer

AI-generated content may be incorrect.

Sơ đồ 2.8 Sơ đồ class

#### 2.2.2.5. ERD (Entity Relationship Diagram)

A computer screen shot of a computer

AI-generated content may be incorrect.

Sơ đồ 2.9 Sơ đồ thực thể quản hệ

### 2.2.3. Thiết kế cơ sở dữ liệu

**1. Bảng users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| user\_id | BIGINT | PK, AUTO\_INCREMENT | Khóa chính định danh người dùng |
| user\_name | VARCHAR(255) | NOT NULL | Tên hiển thị của người dùng |
| email | VARCHAR(255) | UNIQUE, NOT NULL | Email dùng để đăng nhập |
| phone | VARCHAR(20) | NULL | Số điện thoại của người dùng |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Thời điểm tạo tài khoản |
| is\_active | TINYINT(1) | DEFAULT 1 | 1 = hoạt động, 0 = khóa |
| is\_deleted | TINYINT(1) | DEFAULT 0 | 1 = đã xóa mềm |
| password | VARCHAR(255) | NULL | Mật khẩu đã mã hóa (BCrypt) |
| verified | TINYINT(1) | DEFAULT 1 | Trạng thái xác minh tài khoản |
| role | ENUM('USER','ADMIN') | DEFAULT 'USER' | Phân quyền người dùng |
| avatar\_url | VARCHAR(500) | NULL | Ảnh đại diện lưu trên Cloudinary |
| full\_name | VARCHAR(255) | NULL | Tên đầy đủ (khác với user\_name) |

Bảng 11 Bảng Users

**2. Bảng categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| user\_id | BIGINT | PK, AUTO\_INCREMENT | Khóa chính định danh người dùng |
| user\_name | VARCHAR(255) | NOT NULL | Tên hiển thị của người dùng |
| email | VARCHAR(255) | UNIQUE, NOT NULL | Email dùng để đăng nhập |
| phone | VARCHAR(20) | NULL | Số điện thoại của người dùng |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Thời điểm tạo tài khoản |

Bảng 12 Bảng Categories

**3. Bảng expenses**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu** | **Ràng buộc** | **Mô tả** |
| expense\_id | BIGINT | PK | ID chi tiêu |
| user\_id | BIGINT | FK | Người thực hiện |
| category\_id | BIGINT | FK | Danh mục chi tiêu |
| amount | DECIMAL | NOT NULL | Số tiền |
| note | VARCHAR(255) | NULL | Ghi chú |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày tạo |
| expense\_date | DATE | NOT NULL | Ngày phát sinh |
| |  | | --- | |  |   Wallet\_id | BIGINT | FK | Ví được trừ tiền |

Bảng 13 Bảng Expenses

**4. Bảng incomes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| income\_id | BIGINT | PK, AUTO\_INCREMENT | Khóa chính thu nhập |
| user\_id | BIGINT | FK → users(user\_id), NOT NULL | Người dùng sở hữu khoản thu |
| category\_id | BIGINT | FK → categories(category\_id), NOT NULL | Danh mục thu nhập |
| amount | DECIMAL(15,2) | NOT NULL | Số tiền thu |
| note | VARCHAR(255) | NULL | Ghi chú |
| income\_date | DATE | NOT NULL | Ngày thu |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Thời gian tạo bản ghi |

Bảng 14 Bảng Incomes

**5. Bảng budgets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| budget\_id | BIGINT | PK, AUTO\_INCREMENT | Khóa chính ngân sách |
| user\_id | BIGINT | FK → users(user\_id), NOT NULL | Người đặt ngân sách |
| category\_id | BIGINT | FK → categories(category\_id), NOT NULL | Danh mục ngân sách |
| amount\_limit | DECIMAL(15,2) | NOT NULL | Giới hạn chi |
| start\_date | DATE | NOT NULL | Ngày bắt đầu |
| end\_date | DATE | NOT NULL | Ngày kết thúc |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày tạo ngân sách |

Bảng 15 Bảng Budgets

**6. Bảng Recurring\_transactions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| budget\_id | BIGINT | PK, AUTO\_INCREMENT | Khóa chính ngân sách |
| user\_id | BIGINT | FK → users(user\_id), NOT NULL | Người đặt ngân sách |
| category\_id | BIGINT | FK → categories(category\_id), NOT NULL | Danh mục ngân sách |
| amount\_limit | DECIMAL(15,2) | NOT NULL | Giới hạn chi |
| start\_date | DATE | NOT NULL | Ngày bắt đầu |

Bảng 16 Bảng Recurring\_transactions

**7. Bảng goals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu** | **Mô tả** | **Trường** |
| recurring\_id | BIGINT | PK | recurring\_id |
| user\_id | BIGINT | FK | user\_id |
| category\_id | BIGINT | FK | category\_id |
| amount | DECIMAL | Số tiền cố định | amount |
| note | VARCHAR(255) | Ghi chú | note |
| frequency | ENUM | daily/weekly/monthly | frequency |
| next\_date | DATE | Lần tiếp theo | next\_date |
| created\_at | TIMESTAMP | Khi tạo | created\_at |
| wallet\_id | BIGINT | Ví áp dụng | wallet\_id |
| category\_name | VARCHAR | Hiển thị danh mục (denormalized) | category\_name |

Bảng 17 Bảng Goals

**8. Bảng reports**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu** | **Mô tả** | **Trường** |
| report\_id | BIGINT | PK | report\_id |
| user\_id | BIGINT | FK | user\_id |
| report\_type | VARCHAR(50) | monthly/yearly/weekly/custom | report\_type |
| start\_date | DATE | Từ ngày | start\_date |
| end\_date | DATE | Đến ngày | end\_date |
| total\_income | DECIMAL | Tổng thu | total\_income |
| total\_expense | DECIMAL | Tổng chi | total\_expense |
| created\_at | TIMESTAMP | Ngày lập báo cáo | created\_at |

Bảng 18 Bảng Reports

**9. Bảng notifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| notification\_id | BIGINT | PK, AUTO\_INCREMENT | Khóa chính thông báo |
| user\_id | BIGINT | FK users(user\_id), NOT NULL | Người nhận |
| notification\_title | VARCHAR(255) | NULL | Tiêu đề |
| notification\_message | TEXT | NULL | Nội dung |
| type | ENUM('budget','goal','transaction','system') | DEFAULT 'system' | Loại thông báo |
| is\_read | BOOLEAN | DEFAULT FALSE | Đã đọc hay chưa |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Ngày tạo |

Bảng 19 Bảng Notifications

**10. Bảng settings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu** | **Mô tả** | **Trường** |
| setting\_id | BIGINT | PK | setting\_id |
| user\_id | BIGINT | FK | user\_id |
| currency | VARCHAR | VND, USD… | currency |
| language | VARCHAR | vi, en… | language |
| theme | VARCHAR | light/dark | theme |
| notification\_enabled | TINYINT | Bật thông báo | notification\_enabled |
| dark\_mode | TINYINT | Chế độ tối | dark\_mode |
| date\_format | VARCHAR | dd/MM/yyyy | date\_format |
| default\_wallet\_id | BIGINT | Ví mặc định | default\_wallet\_id |
| show\_balance\_on\_home | TINYINT | Hiện số dư màn home | show\_balance\_on\_home |
| auto\_backup | TINYINT | Bật sao lưu | auto\_backup |
| backup\_frequency | VARCHAR | weekly, monthly | backup\_frequency |
| reminder\_time | TIME | Giờ nhắc nhở | reminder\_time |
| budget\_alert\_enabled | TINYINT | Báo động ngân sách | budget\_alert\_enabled |
| budget\_threshold | DOUBLE | % vượt mức | budget\_threshold |

Bảng 20 Bảng Settings

# CHƯƠNG 3. CÀI ĐẶT THỬ NGHIỆM

## 3.1. Kết quả giao diện đạt được

### 3.1.1. Giao diện người dùng

**1. Trang danh mục (màn hình nhập thu/ chi)**

**A screenshot of a cell phone

AI-generated content may be incorrect.**

Hình 3. 1 Giao diện danh mục chi tiêu

**A screenshot of a phone

AI-generated content may be incorrect.**

Hình 3. 2 Hình ảnh giao diện thu nhập

**Mô tả giao diện**

Giao diện **Danh mục** được thiết kế trực quan và dễ sử dụng, giúp người dùng quản lý và lựa chọn các danh mục chi tiêu – thu nhập một cách nhanh chóng. Phần đầu màn hình là thanh tiêu đề với màu cam nhạt, hiển thị tên trang "Danh mục" cùng các biểu tượng chức năng như thông báo, cài đặt và menu tùy chọn. Bên dưới là thanh điều hướng gồm hai tab “Chi tiêu” và “Thu nhập”, cho phép người dùng chuyển đổi giữa hai nhóm danh mục khác nhau. Mỗi tab hiển thị danh sách các danh mục dưới dạng các thẻ vuông bo tròn, có màu nền sáng và được minh họa bằng icon sinh động kèm tên danh mục phía dưới. Các thẻ được sắp xếp theo dạng lưới ba cột, tạo cảm giác gọn gàng và dễ quan sát. Danh mục trong tab Chi tiêu gồm các nhóm như Ăn uống, Nhà ở, Giáo dục, Đi lại, Y tế – Sức khỏe, Giải trí, v.v., trong khi tab Thu nhập hiển thị các nguồn thu như Lương, Tiền thưởng, Kinh doanh, Cho thuê nhà, Freelance hay Cashback. Ở cuối màn hình, ứng dụng cung cấp hai nút hành động “Nhập khoản chi” và “Nhập khoản thu” với màu sắc đặc trưng, cho phép người dùng thao tác nhanh chóng sau khi chọn danh mục phù hợp. Tổng thể, giao diện được tối ưu theo hướng trực quan, hiện đại và thân thiện, đảm bảo trải nghiệm mượt mà cho người dùng.

**2. Giao diện đăng ký**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 3 Giao diện trang đăng ký

**Mô tả giao diện đăng ký**

Giao diện **Đăng ký tài khoản** được thiết kế đơn giản, hiện đại với tông màu xanh nhạt tạo cảm giác nhẹ nhàng và thân thiện. Phần đầu màn hình hiển thị hình minh họa chú heo đất tiết kiệm, gợi ý chức năng quản lý chi tiêu của ứng dụng. Bên dưới là tiêu đề “Tạo tài khoản mới” kèm biểu tượng trái tim, cùng dòng mô tả ngắn giúp người dùng hiểu mục đích của bước đăng ký. Khu vực nhập liệu bao gồm các trường: Họ và tên, Số điện thoại, Email, Mật khẩu và Xác nhận mật khẩu. Mỗi trường đều có biểu tượng minh họa giúp tăng tính trực quan; riêng các trường mật khẩu có thêm nút ẩn/hiện để tăng tính bảo mật và tiện dụng. Phần hành động chính là nút “Đăng ký” với màu cam nổi bật, bo góc mềm mại, thu hút sự chú ý và dễ dàng thao tác. Cuối màn hình, ứng dụng cung cấp liên kết “Đăng nhập ngay” dành cho những người đã có tài khoản, giúp điều hướng người dùng hợp lý. Tổng thể, giao diện được trình bày rõ ràng, trực quan và mang lại trải nghiệm đăng ký nhanh gọn, dễ sử dụng.

**3. Giao diện đăng nhập**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 4 Giao diện đăng nhập

**Mô tả giao diện đăng nhập**

Giao diện đăng nhập được thiết kế đơn giản và hiện đại với tông màu xanh nhạt chủ đạo, tạo cảm giác nhẹ nhàng và dễ nhìn. Ở giữa màn hình là hình minh họa chú heo tiết kiệm giúp giao diện sinh động và phù hợp với chủ đề quản lý chi tiêu. Bên dưới là tiêu đề “Chào mừng trở lại” nổi bật và dòng mô tả ngắn gọn. Hai ô nhập Email và Mật khẩu được bo tròn, có icon minh họa và bố cục rõ ràng, kèm tùy chọn ẩn/hiện mật khẩu. Nút “Đăng nhập” màu cam nổi bật giúp thu hút thao tác. Cuối cùng là liên kết “Đăng ký ngay” dành cho người dùng chưa có tài khoản, được đặt gọn gàng phía dưới để dễ quan sát và sử dụng.

**4. Giao diện nhập khoản chi /thu**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 5 Giao diện thêm khoản chi

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 6 Giao diện thêm khoản thu

**Mô tả giao diện nhập khoản thu/ chi**

Giao diện **Thêm giao dịch** được thiết kế tối giản với nền màu be nhẹ, bao gồm các trường nhập **Số tiền**, **Ghi chú**, lựa chọn **Ngày** và **Ví**. Mọi thành phần được căn thẳng hàng, rõ ràng và dễ thao tác. Điểm khác biệt nằm ở nút hành động: khi thêm **khoản chi**, nút sẽ có màu **đỏ** nổi bật; khi thêm **khoản thu**, nút chuyển sang **màu xanh lá**, giúp người dùng dễ phân biệt hai loại giao dịch. Thiết kế trực quan, rõ ràng và tập trung vào trải nghiệm nhập liệu nhanh.

**5. Giao diện xem thống kê thu nhập**

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 3. 7 Giao diện xem thống kê thu nhập

**Mô tả giao diện xem thống kê thu nhập**

Giao diện **Thống kê chi tiêu** sử dụng tông màu sáng và bố cục trực quan, hiển thị tổng số tiền đã chi trong một thẻ nổi bật màu đỏ nhạt. Bên dưới là biểu đồ hình tròn minh họa tỷ lệ chi tiêu theo từng danh mục, giúp người dùng dễ dàng hình dung mức độ phân bổ. Danh sách các danh mục được trình bày rõ ràng với màu sắc tương ứng, kèm theo số tiền chi tiết cho từng mục, tạo nên một giao diện trực quan, dễ đọc và hỗ trợ người dùng theo dõi chi tiêu hiệu quả.

**6. Giao diện xem thống kê thu nhập**

A screenshot of a graph

AI-generated content may be incorrect.

Hình 3. 8 Giao diện xem thống kê thu nhập

**Mô tả giao diện:**

Giao diện **Thống kê thu nhập** có thiết kế sáng và dễ nhìn, với thẻ “Tổng thu” được làm nổi bật bằng màu xanh thể hiện sự tăng trưởng tích cực. Phần biểu đồ tròn bên dưới hiển thị tỷ lệ thu nhập theo từng danh mục, giúp người dùng nắm bắt nhanh nguồn thu chính. Danh sách danh mục được đặt rõ ràng kèm biểu tượng màu tương ứng và số tiền chi tiết, tạo nên một bố cục trực quan, gọn gàng và hỗ trợ người dùng theo dõi thu nhập một cách hiệu quả.

**7. Giao diện báo cáo & thống kê**

A screenshot of a graph

AI-generated content may be incorrect.

Hình 3. 9 Giao diện báo cáo và thống kê

**Mô tả giao diện:**

Giao diện **Báo cáo & Thống kê** hiển thị tổng quan tài chính theo năm với bố cục trực quan và rõ ràng. Phần đầu cho phép người dùng chọn năm, tiếp theo là biểu đồ cột so sánh giữa **thu** và **chi**, giúp nhận biết mức chênh lệch một cách nhanh chóng. Bên dưới là biểu đồ tròn phân bổ thu – chi theo tỷ lệ, kèm theo dòng tổng kết số tiền thu và chi. Cuối màn hình là danh sách các báo cáo theo tháng và tuần, mỗi mục trình bày khoảng thời gian và tổng số thu – chi, được thiết kế dưới dạng thẻ nổi bật giúp người dùng dễ theo dõi và truy cập thông tin chi tiết.

**8. Giao diện AI dự đoán chi tiêu**

**A screenshot of a phone

AI-generated content may be incorrect.**

Hình 3. 10 Giao diện AI dự đoán chi tiêu

**Mô tả giao diện:**

Giao diện **AI dự đoán chi tiêu** trình bày thông tin rõ ràng và dễ theo dõi với các thẻ nổi bật. Phần đầu hiển thị số tiền dự đoán chi tiêu trong tháng và số tiền đã tiêu cho đến hiện tại. Bên dưới là bảng phân tích chi tiêu theo từng tuần do AI gợi ý, gồm số tiền đã dùng và mức chi hợp lý AI đề xuất. Các mục tuần được tách riêng bằng đường kẻ, giúp bố cục gọn gàng và dễ đọc. Cuối màn hình là thẻ thông báo trạng thái chi tiêu màu xanh, cho biết người dùng đang chi tiêu ở mức ổn định và hợp lý.

**9. Giao diện giao dịch định kì**

**A white screen with black border

AI-generated content may be incorrect.**

Hình 3. 11 Giao diện giao dịch định kì

**Mô tả giao diện giao dịch định kỳ**

Giao diện **Giao dịch định kỳ** được thiết kế đơn giản và gọn gàng, hiển thị mỗi giao dịch dưới dạng một thẻ thông tin màu be nhạt. Thẻ bao gồm các dữ liệu như số tiền, danh mục, chu kỳ lặp và ngày thực hiện kế tiếp. Bên cạnh đó có biểu tượng thùng rác màu đỏ để xóa nhanh giao dịch. Góc dưới bên phải là nút **+** nổi bật, cho phép người dùng thêm giao dịch định kỳ mới. Bố cục rõ ràng, dễ nhìn và thuận tiện cho việc quản lý các khoản chi lặp lại.

**10. Giao diện tạo giao dịch định kỳ**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 12 Giao diện tạo giao dịch định kỳ

**Mô tả giao diện giao dịch định kỳ**

Giao diện Tạo giao dịch định kỳ xuất hiện dưới dạng một hộp thoại nổi bật ở giữa màn hình, với nền màu be sáng và góc bo tròn mềm mại. Bên trong gồm các trường chọn danh mục, nhập số tiền và ghi chú, tiếp theo là lựa chọn tần suất lặp như “Hàng tháng”. Phía dưới là nút Tạo được làm nổi nhẹ, giúp người dùng dễ dàng hoàn tất thao tác. Thiết kế tập trung, rõ ràng và tối ưu cho việc thêm nhanh một giao dịch định kỳ mới.

**11. Giao diện quản lý ví**

**A white surface with a black border

AI-generated content may be incorrect.**

Hình 3. 13 Giao diện quản lý ví

**Mô tả giao diện quản lý ví**

Giao diện **Danh sách ví** được thiết kế tối giản với một thẻ ví hiển thị thông tin gồm tên ví và loại ví, kèm biểu tượng minh họa màu tím giúp dễ nhận diện. Thẻ có góc bo tròn nhẹ và đổ bóng nhẹ tạo cảm giác nổi bật. Ở góc dưới bên phải là nút **+** dạng nút nổi màu be, cho phép người dùng thêm ví mới. Bố cục rộng rãi, đơn giản và trực quan, phù hợp cho việc quản lý các loại ví trong ứng dụng.

**12. Giao diện tạo ví mới**

**A screenshot of a phone

AI-generated content may be incorrect.**

Hình 3. 14 Giao diện tạo ví mới

**Mô tả giao diện tạo ví mới**

Giao diện **Thêm ví mới** xuất hiện dưới dạng hộp thoại nổi màu be nhẹ với góc bo tròn mềm mại. Phía trên là tiêu đề kèm biểu tượng chiếc ví, tạo cảm giác trực quan. Người dùng có thể nhập *Tên ví*, *Số dư* và chọn *Loại ví* thông qua danh sách thả xuống. Bên dưới là hai nút **Hủy** và **Thêm**, được bố trí rõ ràng giúp thao tác nhanh. Toàn bộ thiết kế mang tính tập trung, gọn gàng và tiện dụng cho việc tạo ví mới.

**12. Giao diện quản lý mục tiêu tài chính**

**A screenshot of a chat

AI-generated content may be incorrect.**

Hình 3. 15 Giao diện quản lý mục tiêu tài chính

**Mô tả giao diện quản lý mục tiêu tài chính**

Giao diện Mục tiêu tài chính hiển thị mỗi mục tiêu dưới dạng một thẻ thông tin rõ ràng với nền sáng và góc bo tròn. Thẻ bao gồm tên mục tiêu, số tiền cần đạt và số tiền đã tích lũy, kèm theo một thanh tiến độ thể hiện tỷ lệ hoàn thành. Bên phải có hai biểu tượng: dấu + để thêm tiền vào mục tiêu và thùng rác để xóa mục tiêu. Ngoài ra, ngày dự kiến hoàn thành cũng được hiển thị bên dưới. Góc dưới bên phải có nút + nổi bật giúp người dùng tạo mục tiêu mới. Thiết kế gọn gàng, trực quan và dễ theo dõi tiến trình tiết kiệm.

**13. Giao diện tạo mục tiêu tài chính**

**A screenshot of a phone

AI-generated content may be incorrect.**

Hình 3. 16 Giao diện tạo mục tiêu tài chính

**Mô tả giao diện mục tiêu tài chính**

Giao diện **Thêm mục tiêu mới** được hiển thị dưới dạng hộp thoại nổi với nền màu be nhẹ và góc bo tròn mềm mại, tạo cảm giác trực quan và tập trung. Người dùng có thể nhập *Tên mục tiêu*, *Số tiền cần đạt*, chọn *Ví* và *Danh mục* thông qua các danh sách thả xuống. Bên dưới là tùy chọn chọn hạn hoàn thành kèm biểu tượng lịch. Cuối cùng là hai nút **Hủy** và **Thêm**, được bố trí rõ ràng giúp thao tác nhanh. Thiết kế gọn gàng và thân thiện, hỗ trợ người dùng thiết lập mục tiêu tài chính một cách dễ dàng.

**14. Giao diện cập nhật tiến độ trong mục tiêu tài chính**

**A screenshot of a phone

AI-generated content may be incorrect.**

Hình 3. 17 Giao diện cập nhật tiến độ mục tiêu tài chính

**Mô tả giao diện cập nhật tiến độ mục tiêu tài tài chính**

Giao diện **Cập nhật tiến độ** xuất hiện dưới dạng hộp thoại nổi với nền be nhẹ và góc bo mềm, giúp người dùng tập trung vào thao tác. Hộp thoại hiển thị một ô nhập để người dùng điền số tiền muốn thêm vào mục tiêu, kèm theo hai nút **Hủy** và **Lưu** được bố trí rõ ràng bên dưới. Thiết kế đơn giản, trực quan và hỗ trợ nhanh quá trình cập nhật tiến độ tiết kiệm cho mục tiêu tài chính.

**15. Giao diện thêm ngân sách mới**

**A screenshot of a phone

AI-generated content may be incorrect.**

Hình 3. 18 Giao diện thêm ngân sách

**Mô tả giao diện thêm ngân sách**

Giao diện **Thêm ngân sách mới** hiển thị dưới dạng hộp thoại nổi với nền be sáng và góc bo tròn, cho phép người dùng chọn danh mục, chọn ví (tùy chọn) và nhập hạn mức chi tiêu. Bên dưới là hai nút Hủy và Lưu, bố trí rõ ràng giúp thao tác nhanh. Ở phần Ngân sách, người dùng có thể xem dữ liệu qua hai chế độ: **biểu đồ** và **lịch**. Chế độ biểu đồ cung cấp cái nhìn trực quan về mức sử dụng ngân sách theo từng danh mục, hiển thị tỉ lệ và mức độ tiêu vượt hoặc tiết kiệm. Chế độ xem theo **lịch** cho phép quan sát chi tiêu theo từng ngày trong tháng, đánh dấu các ngày có phát sinh và hiển thị chi tiết khi người dùng chọn vào, giúp đánh giá thói quen chi tiêu theo thời gian.

**16. Giao diện ngân sách**

**A screenshot of a graph

AI-generated content may be incorrect.**

Hình 3. 19 Giao diện thống kê ngân sách

**Mô tả giao diện thống kê ngân sách**

Giao diện **Thống kê ngân sách** hiển thị phần phân tích chi tiêu bằng biểu đồ cột, so sánh trực quan giữa **hạn mức**, **đã chi** và **vượt hạn mức** của từng danh mục. Mỗi cột được thể hiện bằng màu sắc khác nhau, giúp người dùng dễ dàng nhận biết mức độ sử dụng ngân sách. Bên dưới biểu đồ là danh sách các danh mục cùng thông tin chi tiết về số tiền đã chi so với hạn mức và phần trăm sử dụng. Thiết kế tổng thể rõ ràng, trực quan và hỗ trợ người dùng theo dõi ngân sách một cách hiệu quả.

**17. Giao diện xem thống kê bằng lịch**

A screenshot of a calendar

AI-generated content may be incorrect.

Hình 3. 20 Giao diện xem thống kê ngân sách bằng lịch

**Mô tả giao diện lịch ngân sách**

Giao diện **Lịch ngân sách** cho phép người dùng theo dõi mức độ sử dụng ngân sách theo từng tháng trong năm thông qua một lịch trực quan. Ở phần trên cùng là lịch tháng, nơi các ngày có phát sinh chi tiêu được đánh dấu bằng ký hiệu màu giúp người dùng dễ dàng nhận biết thời điểm có giao dịch. Người dùng có thể chuyển đổi tháng bằng các nút điều hướng để xem thông tin ngân sách của các tháng trước hoặc sau.

Bên dưới lịch là phần hiển thị **ngân sách chi tiết theo tháng**, bao gồm danh sách các danh mục đã đặt ngân sách cùng thông tin mức chi tiêu thực tế. Mỗi danh mục hiển thị số tiền **đã chi**, **hạn mức ngân sách**, và **tỉ lệ phần trăm sử dụng**, giúp người dùng nhanh chóng đánh giá mức độ tiêu dùng so với kế hoạch đã đặt ra. Thiết kế giao diện đơn giản, dễ quan sát, hỗ trợ người dùng kiểm soát ngân sách theo từng tháng một cách trực quan và hiệu quả.

**18. Giao diện thông báo**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 21 Giao diện thông báo

**Mô tả giao diện thông báo**

Giao diện **Thông báo** liệt kê các hoạt động tài chính gần đây dưới dạng các thẻ thông tin có nền sáng và góc bo tròn. Mỗi thông báo đi kèm biểu tượng minh họa như chuông, dấu X hay biểu tượng tiền, giúp người dùng dễ nhận biết loại sự kiện. Nội dung thông báo trình bày ngắn gọn, bao gồm việc tạo ngân sách mới, xóa chi tiêu, hoặc thêm giao dịch mới, kèm theo chi tiết danh mục và ví liên quan. Các thẻ được sắp xếp dọc, dễ đọc và dễ tương tác, hỗ trợ người dùng theo dõi mọi thay đổi trong tài khoản một cách nhanh chóng và trực quan.

**19. Giao diện hồ sơ cá nhân**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 22 Giao diện hồ sơ cá nhân

**Mô tả giao diện hồ sơ cá nhân**

Giao diện **Hồ sơ cá nhân** được thiết kế rõ ràng và thân thiện, với ảnh đại diện đặt ở vị trí trung tâm phía trên, kèm nút thay đổi ảnh nhỏ ở góc. Bên dưới là tên người dùng và email được hiển thị nổi bật. Các thông tin quan trọng như số điện thoại, ngày tạo tài khoản, số lượng ví và tổng số dư được trình bày bằng các biểu tượng trực quan và bố cục dễ đọc. Phần chỉnh sửa gồm hai ô nhập *Họ tên* và *Số điện thoại*, phía dưới là nút cập nhật thông tin màu vàng nhẹ và nút đổi mật khẩu tách biệt. Tổng thể giao diện gọn gàng, hiện đại và thuận tiện cho người dùng quản lý thông tin cá nhân.

**20. Giao diện đổi mật khẩu**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3. 23 Giao diện đổi mật khẩu

**Mô tả giao diện đổi mật khẩu**

Giao diện **Đổi mật khẩu** được thiết kế đơn giản và tập trung, gồm ba trường nhập: *mật khẩu cũ*, *mật khẩu mới* và *xác nhận mật khẩu*. Các ô được bố trí theo chiều dọc, khoảng cách thoáng giúp dễ thao tác. Bên dưới là nút Đổi mật khẩu màu cam nổi bật, giúp người dùng dễ dàng hoàn tất quá trình thay đổi. Giao diện tối giản, rõ ràng và phù hợp với một thao tác bảo mật quan trọng.

**21. Giao diện xuất file excel**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3. 24 Giao diện xuất excel

**Mô tả giao diện xuất file excel**

Đây là màn hình xem file Excel giao dịch, được hiển thị dưới dạng xem bảng dữ liệu đầy đủ. Bảng bao gồm các cột như ID, Loại giao dịch (Thu/Chi), Danh mục, Ghi chú, Số tiền, Ví sử dụng và Ngày tạo. Các dòng dữ liệu được trình bày rõ ràng theo từng giao dịch, giúp người dùng dễ dàng theo dõi, đối chiếu và kiểm tra lịch sử chi tiêu – thu nhập một cách chi tiết. Giao diện thể hiện giống như mở một file Excel thật, hỗ trợ tốt cho việc kiểm tra và xuất báo cáo tài chính.

### 3.1.2. Giao diện Admin

**1. Quản lý người dùng**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 3. 25 Giao diện admin quản lý người dùng

**Mô tả giao diện admin** **quản lý người dùng**

Giao diện quản lý người dùng được trình bày dưới dạng bảng dữ liệu rõ ràng và chuyên nghiệp. Mỗi cột hiển thị một thông tin quan trọng như Id, Username, Email, Full Name, Phone, Role và Active, giúp admin dễ dàng theo dõi và quản lý tài khoản. Các hàng được sắp xếp gọn gàng, có thể cuộn để xem thêm dữ liệu. Ở cuối mỗi hàng là các nút chức năng như Sửa và Xóa, hỗ trợ thao tác nhanh với từng người dùng. Tổng thể giao diện sáng sủa, hiện đại và phù hợp cho hệ thống quản trị.

**2. Giao diện admin quản lý danh mục**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3. 26 Giao diện quản lý danh mục

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Kết luận

Qua quá trình nghiên cứu, phân tích và triển khai, đề tài “Ứng dụng báo chi tiêu” đã hoàn thành các mục tiêu đặt ra ban đầu, từ việc xây dựng hệ thống quản lý giao dịch đến việc cung cấp các báo cáo trực quan giúp người dùng theo dõi tài chính một cách dễ dàng và hiệu quả. Hệ thống đã đáp ứng đầy đủ các chức năng cốt lõi như ghi nhận thu – chi, thống kê theo danh mục, quản lý ví, đặt mục tiêu tài chính, theo dõi ngân sách và quản lý giao dịch định kỳ. Bên cạnh đó, ứng dụng còn hỗ trợ xuất báo cáo, hiển thị biểu đồ chi tiết và tích hợp AI dự đoán chi tiêu nhằm nâng cao trải nghiệm người dùng.

Về mặt kỹ thuật, đồ án đã áp dụng thành công mô hình Flutter cho giao diện người dùng và Spring Boot cho backend, đảm bảo tính ổn định, khả năng mở rộng và tốc độ phản hồi tốt. Cơ sở dữ liệu MySQL được thiết kế hợp lý, hỗ trợ truy vấn nhanh và quản lý dữ liệu tài chính hiệu quả. Hệ thống quản trị (Admin) cũng được xây dựng để dễ dàng theo dõi người dùng, danh mục và các dữ liệu quan trọng khác.

Kết quả này cho thấy ứng dụng không chỉ đáp ứng được nhu cầu quản lý chi tiêu cá nhân hằng ngày mà còn có thể phát triển thành một giải pháp hỗ trợ tài chính cá nhân toàn diện. Trong tương lai, hệ thống có thể mở rộng thêm các tính năng như đồng bộ đa thiết bị, gợi ý tiết kiệm thông minh, chia sẻ ví với thành viên gia đình hoặc tích hợp thanh toán trực tuyến.

## 2. Hướng phát triển

Trong thời gian tới, hệ thống Ứng dụng báo cáo chi tiêu sẽ được mở rộng với các hướng phát triển chính như sau:  
Ứng dụng sẽ đồng bộ dữ liệu đa thiết bị thông qua lưu trữ đám mây, hỗ trợ người dùng sử dụng trên nhiều nền tảng. AI phân tích chi tiêu được nâng cấp để dự đoán, cảnh báo vượt mức và gợi ý tiết kiệm thông minh. Giao diện sẽ bổ sung biểu đồ nâng cao, quét hóa đơn tự động (OCR) và chia sẻ ví/ngân sách gia đình.

Hệ thống sẽ tích hợp thanh toán và liên kết ngân hàng, tăng cường bảo mật với xác thực 2FA, mã hóa dữ liệu và cảnh báo đăng nhập. Ngoài ra, dự án hướng tới phát triển phiên bản Web và Desktop, tối ưu trải nghiệm người dùng (UX) với hiệu năng cao, hỗ trợ Dark Mode, và thông báo thông minh giúp nhắc nhở, tổng kết, và gợi ý chi tiêu định kỳ.

# TÀI LIỆU THAM KHẢO.

(*7. JavaCore.Pdf*, n.d.; *Expense Manager Flutter Application*, n.d.; *Tài Liệu Javaa - Tìm Trên Google*, n.d.; Rajas et al., n.d.)

(*Phần mềm quản lý chi tiêu - Đúng “bí quyết” mới hiệu quả!*, 2025)

(*Front End Web Developer | Online Course | Udacity*, n.d.)

(*App Quản Lí Chi Tiêu Cá Nhân Theo Tháng | PDF*, n.d.)

(*Lập Trình Ứng Dụng Mobile Với Dart Và Flutter (P1)*, n.d.)

(*Flutter Là Gì? Famework Di Động Được Yêu Thích Nhất Hiện Nay*, n.d.)

(*Dart Overview*, n.d.)

(*Top 10 app quản lý chi tiêu miễn phí, tốt nhất (2025 cập nhật)*, 2024)

(*Top 12 App Quản Lý Chi Tiêu Miễn Phí, Dễ Dùng Nhất 2025*, n.d.)

(*Tìm Hiểu Kiến Thức Cơ Bản về API*, 2021)

(*API Là Gì? - QuanTriMang.Com*, n.d.)

(*Basic Structure | Swagger Docs*, n.d.)

(*What Is API*, 2023)

(*Expenses Manager App*, n.d.)

(*User Registration/Authentication Flow on a REST API - Stack Overflow*, n.d.)

(*Getting Started | Building an Application with Spring Boot*, n.d.)

(*Flutter 01 | PDF*, n.d.)

(*Giáo Trình MYSQL | PDF*, n.d.)