ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN

**KHOA KHOA HỌC MÁY TÍNH**



**LẬP TRÌNH ĐA NỀN TẢNG**

ĐỀ TÀI:

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH KẾT HỢP VỚI BLOCKCHAIN**

**Sinh viên thực hiện** : Nguyễn Xuân Trường  – 21IT247

Nguyễn Quốc An – 21IT325

**Giảng viên hướng dẫn** : ThS.Ngô Lê Quân

**Đà Nẵng, tháng 2 năm 2025**

ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN

**KHOA KHOA HỌC MÁY TÍNH**



**LẬP TRÌNH ĐA NỀN TẢNG**

ĐỀ TÀI:

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH KẾT HỢP BLOCKCHAIN**

**Đà Nẵng, tháng 2 năm 2025**

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |
| --- | --- |
| Nhiệm vụ | Người phụ trách |
| Tạo đặc tả | Nguyễn Quốc An  Nguyễn Xuân Trường |
| Tạo đề cương | Nguyễn Quốc An  Nguyễn Xuân Trường |
| Tạo phân tích thiết kế hệ thống | Nguyễn Quốc An  Nguyễn Xuân Trường |
| Xây dựng giao diện | Nguyễn Quốc An  Nguyễn Xuân Trường |
| Xây dựng database | Nguyễn Quốc An  Nguyễn Xuân Trường |
| Làm báo cáo | Nguyễn Quốc An  Nguyễn Xuân Trường |

**NHẬN XÉT**

........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ Giảng viên hướng dẫn ThS.Ngô Lê Quân

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin cảm ơn đến toàn thể các quý thầy cô giáo khoa Khoa học máy tính đã luôn đồng hành cùng chúng em trong những năm bước vào ngưỡng Đại học, giúp chúng em học cách tự lập và tự mình tìm tòi, khám phá những điều mới mẻ. Thầy cô đã giúp chúng em rất nhiều trong việc định hướng học tập.

Chúng em cũng xin gửi lời cảm ơn đến thầy Ngô Lê Quân, thầy đã định hướng giúp chúng em có thể chọn được một đề tài phù hợp với khả năng của mình, đồng thời giúp đỡ trong việc giải đáp và tìm hiểu các thông tin mà chúng em gặp vướng mắc không thể giải đáp được.

Xuất phát từ những lợi ích trên cùng với sự cho phép và tận tình giúp đỡ và chỉ dạy của ThS.Ngô Lê Quân nên nhóm chúng em xin chọn đề tài “XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH KẾT HỢP BLOCKCHAIN”.

Với đề tài này, chúng em mong muốn áp dụng các kiến thức đã được học trong trường cùng với việc tìm hiểu nghiên cứu ngôn ngữ và môi trường lập trình để tìm hiểu và xây dựng hệ thống quản lý tài chính.

Do thời gian hạn chế và chưa có kinh nghiệm trong nghiên cứu và thực hành nên báo cáo còn nhiều thiếu sót. Chúng em mong nhận được đóng góp ý kiến của thầy cô để đề tài được hoàn thiện hơn

Cuối cùng chúng em xin gửi lời cảm ơn đến tất cả các bạn đã giúp đỡ chúng em có thêm nhiều thông tin bổ ích cho đề tài để hoàn thiện dự án của nhóm mình.

Xin chân thành cảm ơn!

**MỤC LỤC**

Trang

[**MỞ ĐẦU 7**](#_heading=h.1d96cc0)

[**CHƯƠNG I. TỔNG QUAN ĐỀ TÀI 9**](#_heading=h.3x8tuzt)

[1.1. Tên đề tài 10](#_heading=h.2ce457m)

[1.2. Giới thiệu chung 10](#_heading=h.rjefff)

[1.3. Tính cấp thiết của đề tài 10](#_heading=h.3bj1y38)

[1.4. Mục tiêu của đề tài 10](#_heading=h.1qoc8b1)

[1.5. Ý nghĩa khoa học và thực tiễn 10](#_heading=h.4anzqyu)

[1.6. Phạm vi đề tài 11](#_heading=h.2pta16n)

[**CHƯƠNG II. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT 12**](#_heading=)

[2.1 Ngôn ngữ Dart 12](#_heading=h.3oy7u29)

[2.2 Blockchain 13](#_heading=h.1idq7dh)

[2.2.1 Định nghĩa blockchain 13](#_heading=h.2hio093)

[2.2.2 Công dụng của blockchain 13](#_heading=h.wnyagw)

[2.3 PostgreSQL 14](#_heading=h.3gnlt4p)

[2.4 Flutter 15](#_heading=h.4fsjm0b)

[2.4.1 Giới thiệu về Flutter 15](#_heading=h.1a346fx)

[2.4.2 Cài đặt và sử dụng Flutter 16](#_heading=h.3u2rp3q)

[**CHƯƠNG III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 17**](#_heading=h.2981zbj)

[3.1 Phân tích yêu cầu 17](#_heading=h.odc9jc)

[3.1.1 Yêu cầu người dùng 17](#_heading=h.38czs75)

[3.1.2. Yêu cầu chức năng 17](#_heading=h.1nia2ey)

[3.1.3. Yêu cầu phi chức năng 17](#_heading=h.47hxl2r)

[3.1.4. Yêu cầu hệ thống 17](#_heading=h.2mn7vak)

[3.2. Thiết kế 17](#_heading=h.11si5id)

[3.2.1 Tác nhân 17](#_heading=h.3ls5o66)

[3.2.2 Danh sách ca sử dụng 18](#_heading=h.20xfydz)

[3.2.3 Biểu đồ ca sử dụng 18](#_heading=h.4kx3h1s)

[3.2.4 Đặc tả ca sử dụng 20](#_heading=h.thw4kt)

[3.2.6 Biểu đồ hoạt động 21](#_heading=h.4cmhg48)

[3.2.7 Biểu đồ trạng thái 22](#_heading=h.16x20ju)

[**CHƯƠNG IV. TRIỂN KHAI VÀ KẾT QUẢ 23**](#_heading=h.261ztfg)

[4.1 Thêm thư viện 23](#_heading=h.l7a3n9)

[4.2 Xây dựng cơ sở dữ liệu 24](#_heading=h.1kc7wiv)

[4.3 Xây dựng các hàm trong flutter 27](#_heading=h.2jh5peh)

[4.3.1 Các hàm tại chức năng giao dịch 27](#_heading=h.ymfzma)

[4.3.2 Các hàm tại smartcontract 30](#_heading=h.3w19e94)

[4.4 Xây dựng giao diện 31](#_heading=h.qbtyoq)

[4.4.1 Trang đăng nhập 31](#_heading=h.3abhhcj)

[4.4.2 Trang đăng ký 32](#_heading=h.49gfa85)

[4.4.3 Trang chính 32](#_heading=h.13qzunr)

[4.4.4 Trang card 33](#_heading=h.22vxnjd)

[4.4.5 Trang thống kê 33](#_heading=h.320vgez)

[4.4.6 Trang cài đặt 34](#_heading=h.415t9al)

[4.4.7 Trang chuyển tiền 34](#_heading=h.vgdtq7)

[**KẾT LUẬN VÀ KIẾN NGHỊ 35**](#_heading=h.1ulbmlt)

[**TÀI LIỆU THAM KHẢO 37**](#_heading=h.4ekz59m)

**DANH MỤC HÌNH ẢNH**

[Hình 2. 1 Dart…………………………………………………………………... 1](#_heading=h.243i4a2)2

[Hình 2. 2 biến và kiểu dữ liệu…………………………………………………... 1](#_heading=h.j8sehv)2

[Hình 2. 3 câu lệnh điều kiện……………………………………………………. 1](#_heading=h.338fx5o)2

[Hình 2. 4 câu lệnh điều kiện……………………………………………………. 13](#_heading=h.338fx5o)

[Hình 2. 5 Cách viết hàm cơ bản………………………………………………. 13](#_heading=h.338fx5o)

[Hình 2. 6 blockchain……………………………………………………………. 1](#_heading=h.42ddq1a)3

[Hình 2. 7 postgreSQL…………………………………………………………... 1](#_heading=h.1vsw3ci)5

[Hình 2. 8 Flutter…………………………………………………………………](#_heading=h.2uxtw84) 16

[Hình 3. 1 Ca sử dụng chính](#_heading=h.302dr9l) ……………………………………………………..18

[Hình 3. 2 Phân rã ca sử dụng quản lý tài khoản](#_heading=h.1f7o1he) ………………………………...19

[Hình 3. 3 Phân rã ca sử dụng quản lý thẻ](#_heading=h.3z7bk57) ……………………………………….19

[Hình 3. 4 Phân rã ca sử dụng thống kê](#_heading=h.2eclud0) ………………………………………….20

[Hình 3. 5 Biểu đồ hoạt động giao dịch](#_heading=h.2rrrqc1) ………………………………………….21

[Hình 3. 6 Biểu đồ tuần tự giao dịch](#_heading=h.3qwpj7n) ……………………………………………..22

[Hình 4. 1 Các thư viện trong dự án](#_heading=h.356xmb2) ……………………………………………..23

[Hình 4. 2 Cơ sở dữ liệu](#_heading=h.44bvf6o) …………………………………………………………24

[Hình 4. 3 Hàm lấy danh mục](#_heading=h.3im3ia3) ……………………………………………………27

[Hình 4. 4 Hàm lấy thông tin thẻ](#_heading=h.1xrdshw) ………………………………………………...28

[Hình 4. 5 Hàm lấy tên người nhận](#_heading=h.4hr1b5p) ………………………………………………28

[Hình 4. 6 Hàm kiểm tra số dư](#_heading=h.2wwbldi) …………………………………………………..29

[Hình 4. 7 Hàm chuyển tiền](#_heading=h.1c1lvlb) ……………………………………………………..29

[Hình 4. 8 Các hàm tại smartcontract](#_heading=h.2b6jogx) ……………………………………………30

[Hình 4. 9 Trang đăng nhập](#_heading=h.1pgrrkc) ……………………………………………………...31

[Hình 4. 10 Trang đăng ký](#_heading=h.2olpkfy) ……………………………………………………[3](#_heading=h.2olpkfy)2

[Hình 4. 11 Trang chính](#_heading=h.3nqndbk)[……………………………………………………](#_heading=h.1pgrrkc)…….32

[Hình 4. 12 Trang card](#_heading=h.i17xr6)[…………………………………………………](#_heading=h.1pgrrkc)………...33

[Hình 4. 13 Trang thống kê](#_heading=h.1h65qms)[……………………………………………………](#_heading=h.1pgrrkc)…33

[Hình 4. 14 Trang cài đặt](#_heading=h.2gb3jie)[……………………………………………………](#_heading=h.1pgrrkc)…...34

[Hình 4. 15 Trang chuyển tiền](#_heading=h.3fg1ce0)[…………………………………………………](#_heading=h.1pgrrkc)... [3](#_heading=h.3fg1ce0)4

**DANH MỤC BẢNG**

[Bảng 3. 1 Bảng tác nhân………………………………………………………... 18](#_heading=h.1hmsyys)

[Bảng 3. 2 Bảng danh sách ca sử dụng………………………………………….. 18](#_heading=h.2grqrue)

[Bảng 3. 3 Đặc tả ca sử dụng giao dịch…………………………………………. 2](#_heading=h.3tbugp1)0

[Bảng 3. 4 Biểu đồ lớp - 1](#_heading=h.3dhjn8m) ………………………………………………………..21

[Bảng 3. 5 Biểu đồ lớp - 2](#_heading=h.1smtxgf) ………………………………………………………..21

# MỞ ĐẦU

**1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài**

Blockchain là một công nghệ mang tính cách mạng trong lĩnh vực lưu trữ và quản lý dữ liệu, đặc biệt là trong các hệ thống tài chính. Hiện nay, nhiều nghiên cứu đã chỉ ra rằng blockchain có tiềm năng lớn trong việc nâng cao tính minh bạch, bảo mật và khả năng truy xuất dữ liệu trong các giao dịch tài chính. Các hệ thống truyền thống thường phụ thuộc vào bên thứ ba để xác thực và quản lý giao dịch, điều này không chỉ làm tăng chi phí mà còn gây ra các rủi ro về bảo mật.

Trong những năm gần đây, nhiều ngân hàng và tổ chức tài chính lớn trên thế giới đã bắt đầu ứng dụng blockchain để quản lý các giao dịch tài chính, hợp đồng thông minh và hệ thống thanh toán phi tập trung. Ở Việt Nam, việc nghiên cứu và ứng dụng blockchain vào quản lý tài chính vẫn đang trong giai đoạn đầu, nhưng có nhiều tiềm năng để phát triển, đặc biệt là trong bối cảnh nền kinh tế số đang bùng nổ.

**2. Tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài**

* Tính cấp thiết: Với sự phát triển của công nghệ số, đặc biệt là các ứng dụng tài chính phi tập trung (DeFi), nhu cầu về một hệ thống quản lý tài chính an toàn, minh bạch và hiệu quả ngày càng tăng. Việc ứng dụng blockchain giúp loại bỏ các bên trung gian, giảm thiểu rủi ro gian lận và tối ưu hóa chi phí giao dịch.
* Ý nghĩa khoa học: Đề tài nghiên cứu các nguyên tắc và công nghệ nền tảng của blockchain, từ đó áp dụng vào hệ thống quản lý tài chính. Ngoài ra, đề tài còn tìm hiểu cách tích hợp smart contract để tự động hóa giao dịch và tăng cường bảo mật dữ liệu.
* Ý nghĩa thực tiễn: Hệ thống quản lý tài chính dựa trên blockchain giúp người dùng theo dõi, quản lý chi tiêu và thực hiện giao dịch một cách minh bạch và an toàn. Đồng thời, doanh nghiệp có thể sử dụng hệ thống này để tối ưu hóa quy trình kế toán, thanh toán và kiểm soát dòng tiền.

**3. Mục tiêu**

Mục tiêu của đề tài " ỨNG DỤNG BLOCKCHAIN VÀO HỆ THỐNG QUẢN LÝ TÀI CHÍNH" là là xây dựng một hệ thống quản lý tài chính cá nhân dựa trên công nghệ blockchain. Hệ thống này sẽ giúp người dùng:

* Ghi chép, theo dõi và phân tích chi tiêu một cách bảo mật và minh bạch.
* Sử dụng smart contract để tự động hóa các giao dịch tài chính.
* Tăng cường bảo mật thông tin tài chính bằng công nghệ blockchain.
* Giảm thiểu sự phụ thuộc vào bên thứ ba trong quá trình quản lý và giao dịch tài chính.

**4. Công cụ xây dựng ứng dụng**

* Flutter
* Ngôn ngữ Dart
* PostSQL
* Hardhat
* Các thư viện hỗ trợ
* Visual code studio
* Github

**5. Phương pháp thực hiện**

* Nghiên cứu lý thuyết:
* Tìm hiểu công nghệ blockchain và các ứng dụng tài chính hiện có.
* Phân tích các nền tảng blockchain phù hợp với đề tài.
* Xây dựng mô hình dữ liệu và thiết kế hệ thống.
* Phát triển hệ thống:
* Xây dựng giao diện người dùng bằng Flutter.
* Phát triển backend sử dụng PostgreSQL để lưu trữ dữ liệu.
* Viết và triển khai smart contract bằng Hardhat.
* Kết nối ứng dụng với blockchain thông qua web3dart.
* Kiểm thử và đánh giá:
* Kiểm thử chức năng của hệ thống.
* Đánh giá hiệu suất và tính bảo mật của hệ thống.
* Thu thập phản hồi từ người dùng thử nghiệm để cải thiện sản phẩm.
* Triển khai và tổng kết:
* Triển khai hệ thống trên môi trường thực tế.
* Đánh giá kết quả đạt được và đề xuất hướng phát triển trong tương lai.

**6. Bố cục của Đồ án**

Ngoài phần mở đầu và kết luận,bố cục của đề tài còn có thêm 4 chương.

Chương 1: Giới thiệu tổng quan về đề tài.

Chương 2: Cơ sở lý thuyết tập trung tìm hiểu các cơ sở lý thuyết liên quan.

Chương 3: Phân tích thiết kế hệ thống

Chương 4: Triển khai và kết quả

Cuối cùng là phần kết luận và hướng phát triển của đề tài



# CHƯƠNG I. TỔNG QUAN ĐỀ TÀI

## Tên đề tài

Tên đề tài: “ỨNG DỤNG BLOCKCHAIN VÀO HỆ THỐNG QUẢN LÝ TÀI CHÍNH”.

## Giới thiệu chung

Đề tài " ỨNG DỤNG BLOCKCHAIN VÀO HỆ THỐNG QUẢN LÝ TÀI CHÍNH " .

## Tính cấp thiết của đề tài

Trong bối cảnh công nghệ tài chính ngày càng phát triển, nhu cầu về một hệ thống quản lý tài chính an toàn, minh bạch và hiệu quả ngày càng trở nên quan trọng. Tuy nhiên, các hệ thống truyền thống vẫn tồn tại nhiều hạn chế như:

* Rủi ro bảo mật: Dữ liệu tài chính dễ bị tấn công, giả mạo hoặc mất mát.
* Thiếu minh bạch: Nhiều hệ thống tài chính hiện tại hoạt động theo mô hình tập trung, khiến người dùng khó kiểm chứng tính chính xác của dữ liệu.
* Chi phí vận hành cao: Việc duy trì hệ thống tập trung yêu cầu một bộ máy quản lý phức tạp và tốn kém.

Việc ứng dụng công nghệ Blockchain vào hệ thống quản lý tài chính sẽ giúp giải quyết các vấn đề trên bằng cách cung cấp một hệ thống phi tập trung, bảo mật cao và có khả năng kiểm tra dữ liệu công khai. Điều này không chỉ giúp tăng cường bảo mật mà còn cải thiện tính minh bạch và hiệu quả trong quản lý tài chính.

## Mục tiêu của đề tài

Đề tài "Ứng dụng Blockchain vào hệ thống quản lý tài chính" hướng đến các mục tiêu sau:

* Phát triển một hệ thống quản lý tài chính cá nhân có tích hợp công nghệ Blockchain để đảm bảo tính minh bạch và bảo mật dữ liệu.
* Cung cấp giải pháp lưu trữ giao dịch phi tập trung, giúp người dùng kiểm soát dữ liệu tài chính cá nhân mà không cần bên trung gian.
* Xây dựng các tính năng phân tích tài chính dựa trên dữ liệu thu chi, giúp người dùng đưa ra quyết định tài chính hợp lý.
* Hỗ trợ giao dịch tài chính an toàn và nhanh chóng, ứng dụng Smart Contract để tự động hóa các giao dịch mà không cần sự can thiệp của bên thứ ba.

## Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học:

* Đề tài nghiên cứu và ứng dụng công nghệ Blockchain, một trong những xu hướng công nghệ quan trọng trong lĩnh vực tài chính và công nghệ thông tin.
* Góp phần mở rộng phạm vi ứng dụng của Blockchain vào đời sống thực tiễn, đặc biệt trong lĩnh vực quản lý tài chính cá nhân.
* Đánh giá hiệu suất và tính ứng dụng của các công nghệ như Smart Contract, Hardhat, và PostgreSQL trong một hệ thống quản lý tài chính.

Ý nghĩa thực tiễn:

* Cung cấp một ứng dụng quản lý tài chính cá nhân an toàn, minh bạch và hiệu quả cho người dùng.
* Giảm thiểu rủi ro gian lận trong giao dịch tài chính nhờ vào tính bất biến của Blockchain.
* Giúp người dùng quản lý chi tiêu thông minh, từ đó có kế hoạch tài chính hợp lý hơn.
* Ứng dụng có thể được mở rộng để tích hợp với các hệ thống tài chính lớn hơn, như ngân hàng số hoặc ví điện tử.

## Phạm vi đề tài

* Đối tượng nghiên cứu:

Ứng dụng Blockchain vào hệ thống quản lý tài chính cá nhân.

Các công nghệ Smart Contract, Hardhat, PostgreSQL để xử lý dữ liệu giao dịch tài chính.

* Phạm vi thực hiện:

Xây dựng ứng dụng di động quản lý tài chính trên nền tảng Flutter.

Hệ thống sẽ tập trung vào các tính năng chính như: ghi chép thu chi, phân tích tài chính, bảo mật giao dịch bằng Blockchain.

Sử dụng PostgreSQL để lưu trữ dữ liệu ngoài chuỗi (off-chain) và Blockchain Ethereum để lưu trữ các giao dịch quan trọng.

* Giới hạn nghiên cứu:

Ứng dụng tập trung vào quản lý tài chính cá nhân, chưa mở rộng đến quản lý tài chính doanh nghiệp.

Chỉ hỗ trợ giao dịch tài chính cơ bản, chưa tích hợp các phương thức thanh toán phức tạp như ngân hàng hay ví điện tử.



# CHƯƠNG II. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

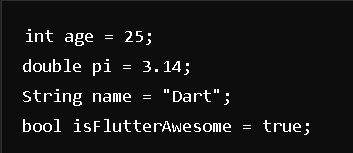
## Ngôn ngữ Dart

Dart là một ngôn ngữ lập trình được phát triển bởi Google, chủ yếu được sử dụng để xây dựng giao diện người dùng trong các ứng dụng di động, web và máy tính để bàn. Dart được thiết kế để tối ưu hóa hiệu suất, có cú pháp dễ học và hỗ trợ lập trình hướng đối tượng.



Hình 2. 1 Dart

* + 1. *Cú pháp cơ bản của Dart*
* Biến và kiểu dữ liệu:



Hình 2. 2 biến và kiểu dữ liệu

* Câu lệnh điều kiện:

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 2. 3 câu lệnh điều kiện

* Vòng lặp:

A black screen with white text

AI-generated content may be incorrect.

Hình 2. 4 câu lệnh điều kiện

* Hàm:

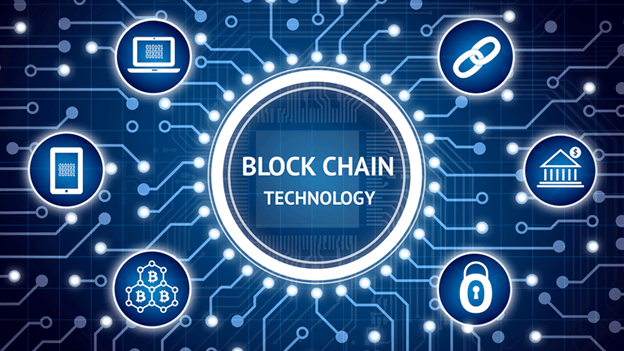
A black screen with white text

AI-generated content may be incorrect.

Hình 2. 5 Cách viết hàm cơ bản

## Blockchain

Blockchain là một công nghệ lưu trữ dữ liệu phân tán, đảm bảo tính minh bạch, bảo mật và không thể thay đổi của dữ liệu.



Hình 2. 6 blockchain

### *2.2.1 Định nghĩa blockchain*

Blockchain là một chuỗi các khối (block) được liên kết với nhau bằng cơ chế mã hóa, mỗi khối chứa một tập hợp dữ liệu giao dịch. Công nghệ này hoạt động theo mô hình phi tập trung, giúp loại bỏ sự phụ thuộc vào bên thứ ba.

### *2.2.2 Công dụng của blockchain*

Blockchain có nhiều ứng dụng trong thực tế, bao gồm:

* Tài chính: Sử dụng trong giao dịch tiền điện tử như Bitcoin, Ethereum.
* Quản lý tài sản: Lưu trữ hợp đồng thông minh (smart contracts).
* Chuỗi cung ứng: Đảm bảo tính minh bạch của quy trình sản xuất và phân phối.
* Y tế: Lưu trữ hồ sơ bệnh án an toàn và bảo mật.
  + 1. *Harthat*

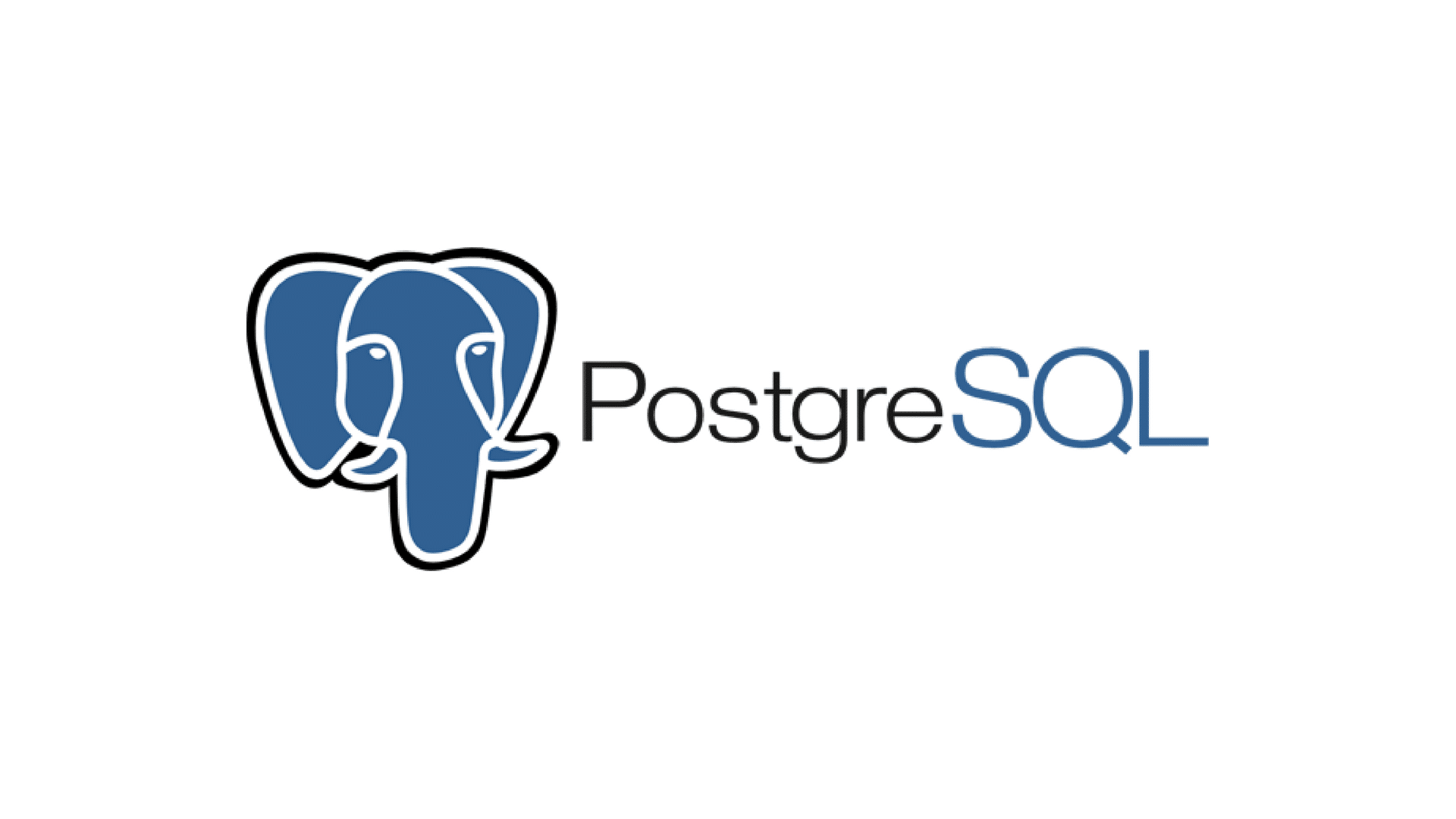
Hardhat là một môi trường phát triển Ethereum (Ethereum development environment) giúp các lập trình viên xây dựng, kiểm thử, triển khai và gỡ lỗi các hợp đồng thông minh (smart contracts). Đây là một công cụ phổ biến trong hệ sinh thái Ethereum, được sử dụng rộng rãi bởi các nhà phát triển blockchain.

*2.2.3.1**Tính năng chính của Hardhat*

1. Chạy mạng blockchain cục bộ (Hardhat Network)
   * Hardhat cung cấp một blockchain cục bộ nhanh chóng và có thể tùy chỉnh, giúp dễ dàng triển khai và kiểm thử hợp đồng thông minh mà không cần kết nối với mạng Ethereum thật.
2. Tích hợp công cụ biên dịch (Compiler Integration)
   * Hardhat hỗ trợ SolidityCompiler(solc) giúp biên dịch các hợp đồng thông minh dễ dàng.
3. Debugging mạnh mẽ
   * Hardhat hỗ trợ console**.**log() trong Solidity để ghi log, giúp dễ dàng debug hợp đồng thông minh.
   * Cung cấp stack trace khi có lỗi, giúp phát hiện vấn đề nhanh chóng.
4. Tự động kiểm thử (Automated Testing)
   * Hardhat hỗ trợ Mocha và Chai để viết và chạy các test cases cho hợp đồng thông minh.
5. Quản lý script và plugin
   * Có thể viết script để tự động hóa các tác vụ như triển khai hợp đồng (deploy), tương tác với hợp đồng, chạy kiểm thử.
   * Hỗ trợ nhiều plugin mạnh mẽ như hardhat-ethers, hardhat-waffle, giúp tích hợp với Ethers.js hoặc Waffle.
6. Tích hợp với MetaMask và các ví khác
   * Hardhat có thể kết nối với MetaMask và các công cụ khác để kiểm tra và tương tác với hợp đồng thông minh.

## PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở mạnh mẽ, hỗ trợ ACID, JSON, và nhiều tính năng nâng cao khác. PostgreSQL phù hợp với các hệ thống yêu cầu bảo mật cao và khả năng mở rộng tốt.



Hình 2. 7 postgreSQL

## Flutter

Flutter là một bộ công cụ phát triển giao diện người dùng mã nguồn mở do Google phát triển, cho phép xây dựng ứng dụng đa nền tảng (Android, iOS, web, desktop) từ cùng một codebase.



Hình 2. 8 Flutter

### *2.4.1 Giới thiệu về Flutter*

Flutter sử dụng ngôn ngữ lập trình Dart và có kiến trúc dựa trên widget. Điểm mạnh của Flutter bao gồm:

* Giao diện đẹp, nhất quán trên nhiều nền tảng.
* Hiệu suất cao nhờ cơ chế rendering Skia.
* Hot reload giúp lập trình viên phát triển nhanh hơn.

### *2.4.2 Cài đặt và sử dụng Flutter*

Cài đặt Flutter

1. Tải về Flutter SDK từ flutter.dev.
2. Cấu hình biến môi trường (PATH).
3. Kiểm tra cài đặt bằng lệnh:

*flutter doctor*

Sử dụng Flutter để tạo ứng dụng

1. Tạo dự án mới:

*flutter create my\_app*

*cd my\_app*

1. Chạy ứng dụng:

*flutter run.*

# 

# CHƯƠNG III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1 Phân tích yêu cầu

### *3.1.1 Yêu cầu người dùng*

Với vai trò là 1 người dùng tôi muốn: ứng dụng giao dịch có ứng dụng công nghệ blockchain, giao dịch đơn giản, thể thể xem các giao dịch, xem thống kê các giao dịch theo hằng tháng

### *3.1.2. Yêu cầu chức năng*

- Xem tài khoản

- Thay đổi tên tài khoản

- Thay đổi số điện thoại

- Xem thông tin thẻ

- Thay đổi mã cvv

- Xem giao dịch

- Tạo giao dịch

- Xem thống kê

### *3.1.3. Yêu cầu phi chức năng*

- Thời gian phản hồi nhanh ( dưới 1 giây)

- Màu sắc ứng dụng đơn giản (xanh, trắng, đen)

- Ít chức năng phức tạp

- Thông tin trực quan, dễ nhìn

### *3.1.4. Yêu cầu hệ thống*

- Chạy đa nền tảng (android, ios)

## 3.2. Thiết kế

### *3.2.1 Tác nhân*

|  |  |
| --- | --- |
| Tác nhân | Chức năng |
| Người dùng | - Xem tài khoản  - Thay đổi tên tài khoản  - Thay đổi số điện thoại  - Xem thông tin thẻ  - Thay đổi mã cvv  - Xem giao dịch  - Tạo giao dịch  - Xem thống kê |

Bảng 3. 1 Bảng tác nhân

### *3.2.2 Danh sách ca sử dụng*

|  |  |  |
| --- | --- | --- |
| ID | Tên ca sử dụng | Tác nhân |
| 1 | Giao dịch | Người dùng |
| 2 | Thống kê | Người dùng |
| 3 | Quản lý tài khoản | Người dùng |
| 4 | Quản lý thẻ | Người dùng |

Bảng 3. 2 Bảng danh sách ca sử dụng

Có 4 ca sử dụng giao dịch, thống kê, quản lý tài khoản, quản lý thẻ đều do người dùng thực hiện

### *3.2.3 Biểu đồ ca sử dụng*

*3.2.3.1 Biểu đồ ca sử dụng chính*

*A diagram of a diagram

AI-generated content may be incorrect.*

Hình 3. 1 Ca sử dụng chính

Tại ca sử dụng chính, người dùng thực hiện các tương tác chính như giao dịch, thống kê, quản ký tài khoản, quản lý thẻ. Tất cả đều phải đăng nhập mới thực hiện được.

*3.2.3.2 Biểu đồ phân rã ca sử dụng*

*3.2.3.2.1 phân rã ca sử dụng quản lý tài khoản*

*A diagram of a diagram

AI-generated content may be incorrect.*

Hình 3. 2 Phân rã ca sử dụng quản lý tài khoản

Tại ca sử dụng quản lý tài khoản, người dùng có thể xem tài khoản, sửa địa chỉ, mật khẩu, số điện thoại, tên người dùng. Đầu tiên phải đăng nhập để có thể sử dụng các chức năng.

*3.2.3.2.2 phân rã ca sử dụng quản lý thẻ*

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3. 3 Phân rã ca sử dụng quản lý thẻ

Tại ca sử dụng quản lý thẻ, người dùng có thể xem thông thẻ, sửa mã cvv.

*3.2.3.2.2 phân rã ca sử dụng thống kê*

*A diagram with text and words

AI-generated content may be incorrect.*

Hình 3. 4 Phân rã ca sử dụng thống kê

Tại ca sử dụng thống kê, người dùng có thể xem danh sách lịch sử giao dịch theo danh mục, chi tiết lịch sử của một giao dịch bất kỳ, xem thống kê giao dịch trong tháng.

### *3.2.4 Đặc tả ca sử dụng*

*3.2.4.4 Giao dịch*

|  |  |
| --- | --- |
| Hoạt động của tác nhân | Hoạt động của hệ thống |
| Nhập địa chỉ người nhận |  |
|  | Trả về tên người dùng ứng với địa chỉ nhận |
| Chọn danh mục |  |
| Ghi nội dung giao dịch |  |
| Nhập số tiền cần gửi |  |
|  | Kiểm tra số dư (nếu số dư đủ), thực hiện giao dịch |

Bảng 3. 3 Đặc tả ca sử dụng giao dịch

*3.2.5 Biểu đồ lớp*

*A screenshot of a computer

AI-generated content may be incorrect.*

Bảng 3. 4 Biểu đồ lớp - 1

*A screenshot of a computer code

AI-generated content may be incorrect.*

Bảng 3. 5 Biểu đồ lớp - 2

### *3.2.6 Biểu đồ hoạt động*

3.2.6.1 Biểu đồ hoạt động giao dịch

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3. 5 Biểu đồ hoạt động giao dịch

Người dùng mở phiên làm việc, người dùng nhập id người nhận, hệ thống kiểm tra id có tồn tại trong hệ thống hay không, nếu không tồn tại thì hiện thông báo không tồn tại người nhận nếu tồn tại thì lấy ra tên của người nhận và điền vào trường tên người nhận, tiếp theo chọn danh mục có sẵn cho giao dịch và điền nội dung giao dịch, sau đó nhập số tiền cần giao dịch, khi ấn gửi hệ thống sẽ lấy số tiền người dùng đã nhập để kiểm tra, nếu số dư không đủ hiện cảnh báo số dư không để thực hiện giao dịch, nếu đủ thì thực hiện giao dịch và chuyển người dùng về màn hình chính.

### *3.2.7 Biểu đồ trạng thái*

3.2.7.1 Biều đồ trạng thái giao dịch

A diagram with text and circles

AI-generated content may be incorrect.

Hình 3. 6 Biểu đồ tuần tự giao dịch

Khi người dùng mở phiên giao dịch, tiến hành đợi trước mỗi hành động từ người dùng và hệ thống giống như hình.

# CHƯƠNG IV. TRIỂN KHAI VÀ KẾT QUẢ

## 4.1 Thêm thư viện

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 4. 1 Các thư viện trong dự án

dependencies (Thư viện chính)

* flutter: SDK Flutter, cung cấp các công cụ để xây dựng ứng dụng trên Android, iOS, web, và desktop.
* postgres: Thư viện kết nối với cơ sở dữ liệu PostgreSQL, giúp truy vấn và thao tác với dữ liệu.
* logger: Cung cấp công cụ để ghi log, giúp dễ dàng debug và theo dõi hoạt động của ứng dụng.
* flutter\_svg: Hỗ trợ hiển thị hình ảnh định dạng SVG trong ứng dụng Flutter.
* google\_fonts: Cung cấp quyền truy cập vào Google Fonts để sử dụng các phông chữ tùy chỉnh.
* fl\_chart: Thư viện giúp vẽ biểu đồ (line chart, bar chart, pie chart, v.v.) trong Flutter.
* web3dart: Thư viện để tương tác với blockchain Ethereum bằng cách sử dụng Web3 (gửi giao dịch, truy vấn hợp đồng thông minh, v.v.).
* http: Thư viện giúp gửi các yêu cầu HTTP (GET, POST, PUT, DELETE) để giao tiếp với API hoặc backend.
* flutter\_dotenv: Hỗ trợ đọc các biến môi trường từ file .env, giúp quản lý thông tin nhạy cảm như API key một cách an toàn.

dev\_dependencies (Thư viện phục vụ phát triển và kiểm thử)

* flutter\_test: Cung cấp các công cụ để viết test (unit test, widget test) cho ứng dụng Flutter.
* flutter\_lints: Bộ quy tắc linting giúp kiểm tra chất lượng code, đảm bảo tuân theo các best practices của Flutter/Dart.
* mockito: Thư viện hỗ trợ tạo mock objects để kiểm thử unit test mà không cần kết nối với các thành phần thực tế.
* build\_runner: Công cụ hỗ trợ tạo code tự động (ví dụ: sinh mã nguồn từ các annotation).
* test: Thư viện hỗ trợ viết và chạy các bài kiểm thử đơn vị cho ứng dụng Dart.
* hive: Cơ sở dữ liệu NoSQL nhẹ, nhanh, lưu trữ dữ liệu cục bộ mà không cần SQLite.
* hive\_flutter: Tiện ích mở rộng giúp Hive tích hợp tốt hơn với Flutter.

## 4.2 Xây dựng cơ sở dữ liệuA computer screen shot of a computer code AI-generated content may be incorrect.

Hình 4. 2 Cơ sở dữ liệu

1. Bảng accounts (Tài khoản người dùng)

Chứa thông tin của người dùng trong hệ thống.

Cấu trúc cột:

* id serial PRIMARY KEY: ID tự động tăng, dùng làm khóa chính.
* username TEXT NOT NULL UNIQUE: Tên đăng nhập, phải là duy nhất.
* passwd TEXT NOT NULL: Mật khẩu của người dùng (nên được mã hóa).
* phone TEXT: Số điện thoại của người dùng.
* address TEXT: Địa chỉ của người dùng.
* time\_created TIMESTAMP DEFAULT NOW(): Thời điểm tài khoản được tạo.
* time\_updated TIMESTAMP DEFAULT NOW(): Thời điểm tài khoản được cập nhật lần cuối.
* creator INT DEFAULT 0: ID của người tạo tài khoản.
* updater INT: ID của người cập nhật tài khoản.

2. Bảng cards (Thông tin thẻ)

Lưu trữ thông tin về thẻ thanh toán liên kết với tài khoản người dùng.

Cấu trúc cột:

* id serial PRIMARY KEY: ID duy nhất của thẻ.
* id\_account INT NOT NULL: ID của tài khoản liên kết với thẻ (khóa ngoại từ accounts).
* card\_number TEXT NOT NULL UNIQUE: Số thẻ duy nhất.
* private\_key TEXT NOT NULL: Khóa riêng tư của thẻ (nên được mã hóa).
* pin TEXT: Mã PIN của thẻ.
* cvv TEXT NOT NULL: Mã CVV (bảo mật) của thẻ.
* total\_amount NUMERIC(15, 2) DEFAULT 0.00: Tổng số tiền hiện có trong thẻ.
* time\_created TIMESTAMP DEFAULT NOW(): Thời điểm thẻ được tạo.
* time\_updated TIMESTAMP DEFAULT NOW(): Thời điểm thẻ được cập nhật lần cuối.
* expiration\_date TIMESTAMP DEFAULT (CURRENT\_TIMESTAMP + INTERVAL '5 years'): Ngày hết hạn của thẻ (5 năm kể từ thời điểm tạo).
* Ràng buộc:
  + CONSTRAINT fk\_account FOREIGN KEY (id\_account) REFERENCES accounts (id) ON DELETE CASCADE: Khi tài khoản bị xóa, tất cả các thẻ liên kết cũng bị xóa.

3. Bảng savings (Tiết kiệm)

Lưu trữ thông tin về các khoản tiết kiệm của người dùng.

Cấu trúc cột:

* id serial PRIMARY KEY: ID duy nhất cho mỗi khoản tiết kiệm.
* id\_account INT NOT NULL: ID của tài khoản sở hữu khoản tiết kiệm.
* goal\_name TEXT NOT NULL: Tên mục tiêu tiết kiệm.
* goal\_amount NUMERIC(15, 2) NOT NULL: Số tiền mục tiêu cần tiết kiệm.
* current\_amount NUMERIC(15, 2) DEFAULT 0.00: Số tiền hiện có trong khoản tiết kiệm.
* deadline TIMESTAMP: Thời hạn cần đạt mục tiêu tiết kiệm.
* Ràng buộc:
  + CONSTRAINT fk\_account FOREIGN KEY (id\_account) REFERENCES accounts (id) ON DELETE CASCADE: Khi tài khoản bị xóa, các khoản tiết kiệm cũng bị xóa.

4. Bảng notifications (Thông báo)

Chứa các thông báo gửi đến người dùng.

Cấu trúc cột:

* id serial PRIMARY KEY: ID duy nhất của thông báo.
* id\_account INT NOT NULL: ID của tài khoản nhận thông báo.
* messages TEXT NOT NULL: Nội dung thông báo.
* sent\_at TIMESTAMP DEFAULT NOW(): Thời điểm gửi thông báo.
* Ràng buộc:
  + CONSTRAINT fk\_account FOREIGN KEY (id\_account) REFERENCES accounts (id) ON DELETE CASCADE: Khi tài khoản bị xóa, thông báo liên quan cũng bị xóa.

5. Bảng categories (Danh mục giao dịch)

Lưu trữ danh mục giao dịch (ví dụ: ăn uống, mua sắm, tiết kiệm, v.v.).

Cấu trúc cột:

* id serial PRIMARY KEY: ID duy nhất của danh mục.
* type\_category INT NOT NULL CHECK (type\_category IN (0, 1)): Loại danh mục (0: thu nhập, 1: chi tiêu).
* name\_category TEXT NOT NULL: Tên danh mục.
* icon TEXT: Biểu tượng của danh mục.
* note TEXT: Ghi chú cho danh mục.
* time\_created TIMESTAMP DEFAULT NOW(): Thời điểm danh mục được tạo.
* time\_updated TIMESTAMP DEFAULT NOW(): Thời điểm danh mục được cập nhật.

6. Bảng transactions (Giao dịch)

Lưu trữ các giao dịch tài chính của người dùng.

Cấu trúc cột:

* id serial PRIMARY KEY: ID duy nhất của giao dịch.
* type\_transaction INT NOT NULL CHECK (type\_transaction IN (0,1)): Loại giao dịch (0: chi tiêu, 1: thu nhập).
* transaction\_hash VARCHAR(256) NOT NULL UNIQUE: Mã giao dịch duy nhất (có thể dùng blockchain hash).
* account\_receiver INT NOT NULL: ID tài khoản nhận tiền.
* name\_receiver VARCHAR(255) NOT NULL: Tên người nhận.
* sender\_id INT NOT NULL: ID tài khoản gửi tiền.
* sender\_name VARCHAR(255) NOT NULL: Tên người gửi.
* amount NUMERIC(10, 2) NOT NULL CHECK (amount >= 0): Số tiền giao dịch (phải >= 0).
* messages TEXT: Ghi chú hoặc mô tả giao dịch.
* timestamps TIMESTAMPTZ DEFAULT NOW(): Thời điểm thực hiện giao dịch.
* category\_id INT: ID danh mục giao dịch (liên kết với bảng categories).
* card\_id INT NOT NULL: ID của thẻ thực hiện giao dịch.
* Ràng buộc:
  + CONSTRAINT fk\_category FOREIGN KEY (category\_id) REFERENCES categories (id): Liên kết với danh mục giao dịch.
  + CONSTRAINT fk\_card FOREIGN KEY (card\_id) REFERENCES cards (id): Liên kết với thẻ thực hiện giao dịch.

## 4.3 Xây dựng các hàm trong flutter

### 4.3.1 Các hàm tại chức năng giao dịch

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 4. 3 Hàm lấy danh mục

getCategory()

* Gửi truy vấn SQL để lấy danh mục giao dịch có type\_category = 0 (loại chi tiêu).
* Dữ liệu trả về được chuyển đổi thành danh sách Map<String, dynamic>.
* Gọi setState() để cập nhật \_category với dữ liệu mới.

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 4. 4 Hàm lấy thông tin thẻ

getCardInfo()

* Lấy thông tin thẻ của người dùng từ database bằng hàm SQL get\_cards\_by\_account(@id).
* Kết quả trả về là danh sách các thẻ có liên kết với tài khoản của widget.userId.
* Nếu có dữ liệu, cập nhật vào dataUser bằng setState().

A computer screen shot of code

AI-generated content may be incorrect.

Hình 4. 5 Hàm lấy tên người nhận

getReceiverAddress(int receiverAddress)

* Truy vấn database để lấy tên người nhận dựa vào địa chỉ (có thể là ID tài khoản hoặc số thẻ).
* Nếu có kết quả, cập nhật ReceiverName.

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 4. 6 Hàm kiểm tra số dư

check\_balance(int id, double mount)

* Kiểm tra số dư của tài khoản bằng cách gọi hàm check\_balance(@id, @mount) trong database.
* Nếu kết quả là false, hiển thị AlertDialog cảnh báo số dư không đủ.
* Nếu số dư đủ, gọi sendMoney(mount) để thực hiện giao dịch.

A screen shot of a computer code

AI-generated content may be incorrect.

Hình 4. 7 Hàm chuyển tiền

sendMoney(double \_mount2)

* Gửi tiền bằng cách gọi hàm insert\_transaction() trong database.
* Dữ liệu được truyền vào bao gồm:
  + 0 (có thể là kiểu giao dịch).
  + generateRandomhash().toString() để tạo hash giao dịch ngẫu nhiên.
  + Địa chỉ người nhận (\_receiverAddressController.text).
  + Tên người nhận (ReceiverName).
  + ID và tên người gửi (widget.userId, widget.username).
  + Số tiền (\_mount2).
  + Nội dung tin nhắn (\_mess.text).
  + ID danh mục (\_selectedCategoryId).
  + ID thẻ người gửi (dataUser[0]['id']).
* Nếu giao dịch thành công, chuyển hướng đến màn hình chính /main với widget.userId.

### 4.3.2 Các hàm tại smartcontract

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 4. 8 Các hàm tại smartcontract

* createAccount(address user, string calldata name) → returns (bool)

Đăng ký tài khoản cho một địa chỉ ví, lưu tên và trạng thái đăng ký của tài khoản.

* getAccount(address user) → returns (string memory, bool)

Trả về tên và trạng thái đăng ký của tài khoản theo địa chỉ ví.

* receive() external payable

Cho phép hợp đồng nhận ETH từ người dùng.

* transfer(address payable recipient, uint256 amount)

Chuyển ETH từ hợp đồng đến một địa chỉ ví cụ thể nếu số dư hợp đồng đủ.

* getBalance() → returns (uint256)

Trả về số dư ETH hiện có trong hợp đồng.

## 4.4 Xây dựng giao diện

### 4.4.1 Trang đăng nhập

A screenshot of a login screen

AI-generated content may be incorrect.

Hình 4. 9 Trang đăng nhập

Người dùng nhập số điện thoại và mật khẩu để đăng nhập, nếu chưa có tài khoản thì ấn vào phần “Sign Up” để tạo tài khoản mới.

### 4.4.2 Trang đăng ký

A screenshot of a login screen

AI-generated content may be incorrect.

Hình 4. 10 Trang đăng ký

Người dùng nhập các thông tin như tên đầy đủ, số điện thoại, địa chỉ và mật khẩu và ấn “Sign Up” để tạo tài khoản, sau khi đăng ký tài khoản thành công, người dùng được chuyển về trang “Sign In”.

### 4.4.3 Trang chính

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4. 11 Trang chính

Sau khi đăng nhập thành công, người dùng được đưa đến trang chính, nơi hiển thị các thông tin cơ bản như: số dư, số tài khoản, mã cvv, thời gian hết hạn của thẻ và các giao dịch đã thực hiện.

### 4.4.4 Trang card

A screenshot of a menu

AI-generated content may be incorrect.

Hình 4. 12 Trang card

Tại trang card, người dùng xem các thông tin của thẻ và có thêm tính năng giới hạn số tiền chi tiêu.

### 4.4.5 Trang thống kê

A screen shot of a graph

AI-generated content may be incorrect.

Hình 4. 13 Trang thống kê

Người dùng sẽ xem được thống kê giao dịch theo danh mục có sẵn được đồ thị hóa, giúp kiểm soát dễ hơn chi tiêu.

### 4.4.6 Trang cài đặt

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4. 14 Trang cài đặt

Người dùng có thể xem thông tin cá nhân, liên hệ với doanh nghiệp, thay đổi mật khẩu tài khoản, xem điều khoản riêng tư.

### 4.4.7 Trang chuyển tiền

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4. 15 Trang chuyển tiền

Người dùng nhập các thông liên quan như: địa chỉ thẻ người nhận, hệ thống tự động lấy tên của người nhận, nội dung chuyển khoản, danh mục, số tiền cần chuyện, sau đó hệ thống kiểm tra số dư và thực hiện giao dịch.

# KẾT LUẬN VÀ KIẾN NGHỊ

1. **Những mặt chưa làm được**

Mặc dù dự án đã đạt được nhiều kết quả quan trọng, vẫn còn một số hạn chế cần được cải thiện:

Hiệu suất và tối ưu hóa

Ứng dụng chưa được tối ưu hóa hoàn toàn cho các thiết bị có cấu hình thấp.

Thời gian phản hồi của một số truy vấn blockchain vẫn còn chậm.

Tích hợp Blockchain

Chưa triển khai đầy đủ các chức năng nâng cao như Smart Contract phức tạp.

Chưa thử nghiệm hệ thống với một lượng lớn giao dịch thực tế.

Bảo mật dữ liệu

Chưa có cơ chế mã hóa mạnh mẽ cho dữ liệu lưu trữ trong PostgreSQL.

Cần cải thiện khả năng chống tấn công DDoS và các lỗ hổng bảo mật khác.

Trải nghiệm người dùng (UI/UX)

Giao diện người dùng chưa được tối ưu hoàn toàn, cần cải thiện về mặt thiết kế và tính trực quan.

Một số chức năng chưa thân thiện với người dùng và cần điều chỉnh.

1. **Những mặt đã làm được**

Dự án đã đạt được nhiều kết quả quan trọng, bao gồm:

Xây dựng ứng dụng tài chính cá nhân với Flutter

Hoàn thành giao diện cơ bản với Flutter.

Đã triển khai một số chức năng như ghi chép thu/chi, phân tích tài chính.

Tích hợp Blockchain vào hệ thống quản lý tài chính

Hoàn thành Smart Contract cơ bản trên nền tảng Ethereum sử dụng Hardhat.

Lưu trữ các giao dịch tài chính trên Blockchain để đảm bảo tính minh bạch.

Phát triển cơ sở dữ liệu với PostgreSQL

Xây dựng hệ thống lưu trữ dữ liệu tài chính.

Kết nối cơ sở dữ liệu với ứng dụng để lưu và truy xuất thông tin hiệu quả.

Xây dựng nền tảng bảo mật cơ bản

Triển khai xác thực người dùng.

Đảm bảo dữ liệu không thể bị thay đổi một cách trái phép nhờ cơ chế Blockchain.

1. **Hướng nghiên cứu**

Trong tương lai, có thể mở rộng và nâng cấp dự án theo các hướng sau:

Tối ưu hiệu suất và mở rộng quy mô

Cải thiện tốc độ xử lý giao dịch bằng cách tối ưu hóa thuật toán và sử dụng các giải pháp Blockchain Layer 2.

Hỗ trợ nhiều loại tiền tệ và tích hợp API ngân hàng để tự động cập nhật giao dịch.

Nâng cao bảo mật

Ứng dụng mã hóa dữ liệu cấp độ trường (Field-level Encryption) trong PostgreSQL.

Nâng cấp cơ chế xác thực bằng sinh trắc học hoặc xác thực hai yếu tố (2FA).

Phát triển tính năng AI hỗ trợ tài chính

Tích hợp Machine Learning để phân tích mô hình chi tiêu và đề xuất tiết kiệm hiệu quả.

Phát triển chatbot hỗ trợ tư vấn tài chính tự động.

Hỗ trợ đa nền tảng

Mở rộng ứng dụng lên nền tảng web và desktop để tiếp cận nhiều người dùng hơn.

Hỗ trợ ứng dụng trên iOS với hiệu suất tối ưu.

Để hoàn thành đề tài này, một lần nữa chúng em xin chân thành cảm ơn thầy giáo TS. Hồ Văn Phi đã quan tâm, giúp đỡ chúng em trong suốt thời gian qua để chúng em hoàn thành được đồ án môn học. Chúng em xin chân thành cảm ơn!



# TÀI LIỆU THAM KHẢO

[1] *Flutter documentation*. (n.d.). <https://docs.flutter.dev/>

[2] *Getting started with Hardhat | Ethereum development environment for professionals by Nomic Foundation*. (n.d.). https://hardhat.org/hardhat-runner/docs/getting-started#overview

[3] *PostgreSQL: documentation*. (n.d.). The PostgreSQL Global Development Group. https://www.postgresql.org/docs/

[4] *Solidity — Solidity 0.8.29 documentation*. (n.d.). https://docs.soliditylang.org/en/latest/