

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: NGHIÊN CỨU KẾT HỢP THUẬT TOÁN K-MEANS,
ISOLATION FOREST VỚI MÔ HÌNH LSTM ĐỂ XÂY DỰNG
ỨNG DỤNG QUẢN LÝ CHI TIÊU CÁ NHÂN**

Giảng viên hướng dẫn: THS. TRẦN PHONG NHÃ

Sinh viên thực hiện: PHẠM QUỐC KHÁNH

Lớp : CQ.62.CNTT

Khoá : K62

Tp. Hồ Chí Minh, năm 2026

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: NGHIÊN CỨU KẾT HỢP THUẬT TOÁN K-MEANS,
ISOLATION FOREST VỚI MÔ HÌNH LSTM ĐỂ XÂY DỰNG
ỨNG DỤNG QUẢN LÝ CHI TIÊU CÁ NHÂN**

Giảng viên hướng dẫn: THS. TRẦN PHONG NHÃ

Sinh viên thực hiện: PHẠM QUỐC KHÁNH

Lớp : CQ.62.CNTT

Khoá : K62

Tp. Hồ Chí Minh, năm 2026

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 6251071048

Họ tên SV: Phạm Quốc Khánh

Khóa: K62

Lớp: CQ.62.CNTT

1. Tên đề tài

Nghiên cứu kết hợp thuật toán K-MEANS, ISOLATION FOREST với mô hình LSTM để xây dựng ứng dụng quản lý chi tiêu cá nhân.

2. Mục đích, yêu cầu

a, Mục đích

Xây dựng ứng dụng di động hỗ trợ quản lý thu chi cá nhân, đồng thời áp dụng thuật toán K-Means để phân tích và phân nhóm hành vi chi tiêu của người dùng, sử dụng Isolation Forest nhằm phát hiện các khoản chi tiêu bất thường, và ứng dụng mô hình LSTM để dự đoán xu hướng chi tiêu trong tương lai. Từ đó, hệ thống đưa ra các gợi ý phù hợp, giúp người dùng quản lý tài chính cá nhân một cách hiệu quả hơn.

b, Yêu cầu

- Ứng dụng cho phép người dùng ghi nhận, chỉnh sửa và theo dõi các khoản thu nhập, chi tiêu.
- Áp dụng thuật toán K-Means để phân nhóm hành vi chi tiêu, Isolation Forest để phát hiện chi tiêu bất thường và LSTM để dự đoán xu hướng chi tiêu.

- Giao diện thân thiện, dễ sử dụng; dữ liệu được lưu trữ và bảo mật an toàn.
- Hệ thống hoạt động ổn định trên nền tảng di động và có khả năng ứng dụng thực tế.

3. Nội dung và phạm vi đề tài

a, Nội dung nghiên cứu

- Thiết kế cơ sở dữ liệu cho hệ thống quản lý tài chính cá nhân: Cơ sở dữ liệu sẽ lưu trữ các thông tin như người dùng, thu nhập, chi tiêu, ngân sách và các giao dịch tài chính cá nhân.
- Xây dựng giao diện người dùng hấp dẫn, sinh động và thân thiện: Thiết kế giao diện giúp người dùng dễ dàng thao tác với các chức năng quản lý tài chính cá nhân, bao gồm nhập thu nhập, chi tiêu, xem báo cáo tài chính và lập ngân sách kế hoạch chi tiêu theo tháng.
- Xây dựng các chức năng chính: Quản lý thu nhập, chi tiêu, lập kế hoạch tài chính, cung cấp báo cáo tài chính chi tiết cho người dùng.
- Áp dụng thuật toán K-Means để phân nhóm hành vi chi tiêu, Isolation Forest để phát hiện chi tiêu bất thường và LSTM để dự đoán xu hướng chi tiêu.

b) Phạm vi đề tài

- Phạm vi nghiên cứu của đề tài được giới hạn trong lĩnh vực quản lý tài chính cá nhân cho người dùng cá nhân trên nền tảng di động.
- Dữ liệu sử dụng trong đề tài là dữ liệu chi tiêu, thu nhập và thông tin liên quan do người dùng trực tiếp nhập vào trong quá trình sử dụng ứng dụng.
- Đề tài không nghiên cứu các bài toán tài chính ở quy mô doanh nghiệp, tổ chức hay các hệ thống tài chính chuyên sâu như kế toán, đầu tư hoặc ngân hàng, mà tập trung vào việc hỗ trợ người dùng cá nhân quản lý và cải thiện thói quen chi tiêu hằng ngày.

4. Công nghệ, công cụ và ngôn ngữ lập trình

- Công nghệ: Flutter.
- Công cụ: Android Studio, Firebase, Draw.io, Figma.
- Ngôn ngữ lập trình: Dart, Python.
- Cơ sở dữ liệu: Firebase.

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

- Ứng dụng di động quản lý chi tiêu cá nhân hoàn chỉnh, cho phép người dùng ghi nhận, theo dõi và quản lý các khoản thu chi.
- Hệ thống phân tích hành vi chi tiêu của người dùng dựa trên thuật toán K-Means, nhằm phân nhóm các đặc điểm chi tiêu.
- Phát hiện các khoản chi tiêu bất thường sử dụng thuật toán Isolation Forest.
- Dự đoán xu hướng chi tiêu trong tương lai bằng LSTM.
- Ứng dụng có khả năng triển khai và sử dụng thực tế, hỗ trợ người dùng ra quyết định tài chính cá nhân hiệu quả hơn.

6. Giáo viên và cán bộ hướng dẫn

Họ tên: ThS. Trần Phong Nhã

Đơn vị công tác: Bộ môn Công nghệ Thông tin – Trường Đại học Giao thông Vận tải
phân hiệu tại Thành phố Hồ Chí Minh.

Điện thoại:

Email:

Ngày 16 tháng 01 năm 2026

Trưởng BM Công nghệ Thông tin

Đã giao nhiệm vụ TKTN

Giáo viên hướng dẫn

ThS. Trần Phong Nhã

Đã nhận nhiệm vụ TKTN

Sinh viên: Phạm Quốc Khánh

Điện thoại: 0354074991

Ký tên:

Email: 6251071048@st.utc2.edu.vn

LỜI CẢM ƠN

Lời đầu tiên, em xin trân trọng gửi đến Quý Thầy, Cô Bộ môn Công nghệ Thông tin – Trường Đại học Giao thông Vận tải, Phân hiệu tại Thành phố Hồ Chí Minh lời chúc sức khỏe và lời cảm ơn chân thành nhất. Trong suốt quá trình học tập và rèn luyện tại trường, Quý Thầy, Cô đã tận tình giảng dạy, truyền đạt cho em những kiến thức chuyên môn quý báu cùng với nhiều kinh nghiệm thực tiễn, tạo nên nền tảng vững chắc để em hoàn thành đồ án tốt nghiệp.

Em xin chân thành cảm ơn Ban Giám hiệu nhà trường đã tạo điều kiện thuận lợi về cơ sở vật chất và môi trường học tập, giúp sinh viên có điều kiện học tập, nghiên cứu và phát triển bản thân một cách toàn diện. Đồng thời, chúng em xin gửi lời cảm ơn sâu sắc đến tập thể giảng viên Bộ môn Công nghệ Thông tin đã luôn đồng hành, hỗ trợ và tận tâm hướng dẫn chúng em trong suốt quá trình học tập tại trường.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến ThS. Trần Phong Nhã, người đã trực tiếp hướng dẫn và chỉ bảo tận tình cho em trong suốt quá trình thực hiện đồ án tốt nghiệp “Nghiên cứu kết hợp thuật toán K-MEANS, ISOLATION FOREST với mô hình LSTM để xây dựng ứng dụng quản lý chi tiêu cá nhân”. Những góp ý chuyên môn, định hướng nghiên cứu và sự động viên của thầy đã giúp em tiếp cận được các bài toán thực tế, từng bước hoàn thiện nội dung và chất lượng của đồ án.

Em cũng xin gửi lời cảm ơn đến các bạn sinh viên lớp Công nghệ Thông tin K62 đã luôn sẵn sàng chia sẻ, hỗ trợ và động viên nhau trong suốt quá trình học tập và thực hiện đồ án.

Mặc dù em đã rất cố gắng trong quá trình thực hiện đồ án tốt nghiệp, tuy nhiên do thời gian và kinh nghiệm thực tế còn hạn chế nên không tránh khỏi những thiếu sót. Em kính mong nhận được sự cảm thông và những ý kiến đóng góp quý báu từ Quý Thầy, Cô để đồ án được hoàn thiện hơn.

Lời cuối cùng, chúng em xin kính chúc Quý Thầy, Cô cùng gia đình dồi dào sức khỏe và thành công trong công tác giảng dạy. Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm

Giáo viên hướng dẫn

ThS. Trần Phong Nhã

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP	i
LỜI CẢM ƠN.....	iv
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	v
MỤC LỤC	vi
DANH MỤC CHỮ VIẾT TẮT	ix
DANH MỤC HÌNH ẢNH.....	x
DANH MỤC BẢNG BIỂU	xii
CHƯƠNG 1 : MỞ ĐẦU.....	1
1.1 Tổng quan về đề tài.....	1
1.2 Mục tiêu nghiên cứu.	4
1.3 Đối tượng và phạm vi nghiên cứu.	7
1.4 Phương pháp nghiên cứu	8
1.5 Cấu trúc báo cáo bài tập lớn	11
1.5.1 Chương 1: Tổng quan bài toán	11
1.5.2 Chương 2: Cơ sở lý thuyết.....	11
1.5.3 Chương 3: Phân tích thiết kế hệ thống	11
1.5.4 Chương 4: Thiết kế và cài đặt chương trình	11
1.5.5 Chương 5: Kết luận và kiến nghị.....	12
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT.....	13
2.1 Tổng quan về quản lý tài chính cá nhân	13
2.1.1 Khái niệm.....	13
2.1.2 Lợi ích của việc quản lý tài chính cá nhân	13
2.1.3 Mục đích của quản lý tài chính cá nhân	13
2.2 Nền tảng công nghệ phát triển ứng dụng.....	14
2.2.1 Tổng quan về ngôn ngữ lập trình Dart	14
2.2.2 Tổng quan về Framework Flutter	15

2.2.3 Tổng quan về nền tảng Firebase	16
2.3 Trí tuệ nhân tạo và học máy trong phân tích dữ liệu	17
2.3.1 Tổng quan về ngôn ngữ lập trình Python	17
2.3.2 Tổng quan về AI	18
2.3.3 Tổng quan về ML	18
2.3.4 Thuật toán K-Means	19
2.3.5 Thuật toán Isolation Forest	21
2.3.6 Mạng nơ-ron hồi tiếp LSTM	22
CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG	25
3.1 Lựa chọn mô hình phát triển.....	25
3.2 Thiết kế chức năng của hệ thống	26
3.2.1 Sơ đồ phân rã chức năng.....	26
3.2.2 Sơ đồ Use case	27
3.2.3 Đặc tả Use case	28
3.3 Thiết kế cơ sở dữ liệu	68
3.3.1 spending Collection (Chi tiêu).....	68
3.3.2 info Collection (Người dùng)	70
3.3.3 budget Collection (Ngân sách)	71
CHƯƠNG 4 : THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH	73
4.1 Kiến trúc hệ thống	73
4.1.1 Cấu trúc phần cứng.....	73
4.1.2 Kiến trúc phần mềm.....	73
4.1.3 Biểu đồ phân cấp chức năng.....	74
4.2 Thiết kế giao diện	75
4.2.1 Giao diện đăng nhập	75
4.2.2 Giao diện đăng ký	76
4.2.3 Giao diện quên mật khẩu	77
4.2.4 Giao diện đổi mật khẩu.....	78
4.2.5 Giao diện tài khoản.....	79

4.2.6 Giao diện xem thông tin cá nhân	80
4.2.7 Giao diện chỉnh sửa thông tin cá nhân	81
4.2.8 Giao diện thêm chi tiêu.....	82
4.2.9 Giao diện xem chi tiết chi tiêu	83
4.2.10 Giao diện chỉnh sửa chi tiêu	84
4.2.11 Giao diện thêm ngân sách.....	85
4.2.12 Giao diện chỉnh sửa ngân sách	86
4.2.13 Giao diện trang chủ.....	87
4.2.14 Giao diện ngân sách.....	88
4.2.15 Giao diện phân tích.....	89
4.2.16 Giao diện tìm kiếm và bộ lọc tìm kiếm	91
KẾT LUẬN VÀ KIẾN NGHỊ	92
5.1 Kết quả đạt được	92
5.2 Hạn chế của hệ thống.....	93
5.3 Hướng phát triển của hệ thống	94
PHỤ LỤC	95
TÀI LIỆU THAM KHẢO	96

DANH MỤC CHỮ VIẾT TẮT

STT	Từ viết tắt	Thuật ngữ	Ghi chú
1	K-Means	K-Means Clustering	
2	LSTM	<i>Long Short-Term Memory</i>	
3	UML	Unified Modeling Language	
4	ERD	Entity–Relationship Diagram	
5	AI	Artificial Intelligence	
6	ML	Machine Learning	
7	UX	User Experience	
8	UI	User Interface	
9	I/O	Input / Output	
10	SSL	Secure Sockets Layer	
11	API	Application Programming Interface	
12	CNN	Convolutional Neural Network	
13	SDK	Software Development Kit	

DANH MỤC HÌNH ẢNH

Hình 2.1 Nguyên lý hoạt động K-means	20
Hình 2.2 Nguyên lý hoạt động Isolation Forest	22
Hình 2.3 Cấu trúc LSTM.....	23
Hình 3.1 Sơ đồ phân rã chức năng	26
Hình 3.2 Sơ đồ Use case.....	27
Hình 3.3 Sơ đồ hoạt động đăng nhập	30
Hình 3.4 Sơ đồ hoạt động đăng ký	33
Hình 3.5 Sơ đồ hoạt động quên mật khẩu	36
Hình 3.6 Sơ đồ hoạt động thay đổi mật khẩu	39
Hình 3.7 Sơ đồ hoạt động thêm chỉ tiêu	42
Hình 3.8 Sơ đồ hoạt động sửa chỉ tiêu	45
Hình 3.9 Sơ đồ hoạt động xóa chỉ tiêu	47
Hình 3.10 Sơ đồ hoạt động xem thông tin cá nhân	49
Hình 3.11 Sơ đồ hoạt động thay đổi thông tin cá nhân	51
Hình 3.12 Sơ đồ hoạt động thay đổi ngôn ngữ.....	53
Hình 3.13 Sơ đồ hoạt động tùy chỉnh sáng tối	55
Hình 3.14 Sơ đồ hoạt động xem lịch sử chỉ tiêu	57
Hình 3.15 Sơ đồ hoạt động xuất CSV	59
Hình 3.16 Sơ đồ hoạt động tìm kiếm chỉ tiêu.....	62
Hình 3.17 Sơ đồ hoạt động xem chi tiết chỉ tiêu.....	64
Hình 4.1 Giao diện đăng nhập.....	75
Hình 4.2 Giao diện đăng ký tài khoản.....	76
Hình 4.3 Giao diện quên mật khẩu.....	77
Hình 4.4 Giao diện đổi mật khẩu.....	78
Hình 4.5 Giao diện tài khoản.....	79
Hình 4.6 Giao diện xem thông tin cá nhân.....	80

Hình 4.7 Giao diện chỉnh sửa thông tin cá nhân	81
Hình 4.8 Giao diện thêm chi tiêu.....	82
Hình 4.9 Giao diện xem chi tiết chi tiêu.....	83
Hình 4.10 Giao diện chỉnh sửa chi tiêu	84
Hình 4.11 Giao diện thêm ngân sách.....	85
Hình 4.12 Giao diện chỉnh sửa ngân sách	86
Hình 4.13 Giao diện trang chủ	87
Hình 4.14 Giao diện ngân sách.....	88
Hình 4.15 Giao diện phân tích bằng biểu đồ cột.....	89
Hình 4.16 giao diện phân tích bằng biểu đồ tròn	90
Hình 4.17 Giao dịch tìm kiếm và bộ lọc tìm kiếm	91

DANH MỤC BẢNG BIỂU

Bảng 3.1 Đặc tả use case đăng nhập.....	29
Bảng 3.2 Đặc tả use case đăng ký	32
Bảng 3.3 Đặc tả use case quên mật khẩu.....	35
Bảng 3.4 Đặc tả use case thay đổi mật khẩu	38
Bảng 3.5 Đặc tả use case thêm chi tiêu	41
Bảng 3.6 Đặc tả use case sửa chi tiêu.....	44
Bảng 3.7 Đặc tả use case xóa chi tiêu	47
Bảng 3.8 Đặc tả use case xem thông tin cá nhân	49
Bảng 3.9 Đặc tả use case thay đổi thông tin cá nhân	51
Bảng 3.10 Đặc tả use case thay đổi ngôn ngữ.....	53
Bảng 3.11 Đặc tả use case tùy chỉnh sáng tối	55
Bảng 3.12 Đặc tả use case xem lịch sử chi tiêu.....	57
Bảng 3.13 Đặc tả use case xuất CSV	59
Bảng 3.14 Đặc tả use case tìm kiếm chi tiêu.....	61
Bảng 3.15 Đặc tả use case xem chi tiết chi tiêu	64
Bảng 3.16 Use case gợi ý chi tiêu thông minh	65
Bảng 3.17 Use case phát hiện chi tiêu bất thường.....	67
Bảng 3.18 Use case dự đoán chi tiêu trong tương lai.....	68
Bảng 3.19 Bảng mô tả spending Collection	69
Bảng 3.20 Bảng mô tả info Collection	70
Bảng 3.21 Bảng mô tả budget Collection.....	71

CHƯƠNG 1 : MỞ ĐẦU

1.1 Tổng quan về đề tài.

Trong những năm gần đây, sự phát triển mạnh mẽ của công nghệ thông tin và các nền tảng số đã tạo ra nhiều thay đổi sâu sắc trong đời sống xã hội, đặc biệt là trong lĩnh vực tài chính cá nhân. Việc sử dụng các thiết bị điện tử như điện thoại thông minh, máy tính bảng và máy tính cá nhân ngày càng trở nên phổ biến, góp phần thay đổi thói quen thanh toán, mua sắm và quản lý tài chính của người dùng. Các hình thức thanh toán không dùng tiền mặt như ngân hàng số, ví điện tử và các nền tảng thanh toán trực tuyến đang dần trở thành lựa chọn ưu tiên, nhất là đối với giới trẻ và người dân sinh sống tại khu vực đô thị, nhờ tính nhanh chóng và tiện lợi mà chúng mang lại.

Tuy nhiên, trong thực tế, tiền mặt vẫn giữ vai trò quan trọng trong nhiều hoạt động chi tiêu hằng ngày, đặc biệt là đối với các giao dịch nhỏ lẻ hoặc tại các khu vực, môi trường sống chưa phổ biến thanh toán điện tử. Sự tồn tại song song giữa tiền mặt và các phương thức thanh toán số khiến cho việc theo dõi và tổng hợp đầy đủ các khoản thu chi cá nhân trở nên khó khăn hơn. Nhiều khoản chi tiêu bằng tiền mặt thường không được ghi chép lại, dẫn đến tình trạng người dùng không nắm rõ được dòng tiền thực tế của bản thân, từ đó gây ảnh hưởng đến hiệu quả quản lý tài chính.

Trong đời sống thường ngày, mỗi cá nhân phải đối mặt với nhiều khoản chi tiêu khác nhau, bao gồm các chi phí sinh hoạt cơ bản như ăn uống, đi lại, học tập, giải trí, mua sắm, cũng như các khoản chi cho những kế hoạch dài hạn như tiết kiệm, đầu tư hoặc xây dựng quỹ dự phòng. Khi số lượng giao dịch ngày càng tăng và diễn ra với tần suất cao, việc kiểm soát các khoản chi tiêu này trở nên phức tạp hơn. Nếu không có phương pháp quản lý phù hợp, người dùng rất dễ rơi vào tình trạng chi tiêu vượt quá khả năng tài chính mà không nhận ra nguyên nhân cụ thể. Điều này đặc biệt dễ xảy ra khi người dùng không phân biệt rõ đâu là các khoản chi cần thiết và đâu là các khoản chi có thể hạn chế hoặc điều chỉnh.

Đối với sinh viên và những người mới đi làm, việc quản lý tài chính cá nhân lại càng trở nên khó khăn hơn do thu nhập còn hạn chế trong khi nhu cầu chi tiêu ngày càng gia tăng. Việc thiếu kinh nghiệm và kỹ năng quản lý tài chính có thể dẫn đến tình trạng mất

cân đối giữa thu nhập và chi tiêu, gây ảnh hưởng đến khả năng tiết kiệm và lập kế hoạch tài chính lâu dài. Chính vì vậy, nhu cầu sử dụng một công cụ hỗ trợ quản lý tài chính hiệu quả, dễ sử dụng và phù hợp với thói quen của người dùng ngày càng trở nên cần thiết.

Mặc dù hiện nay đã có nhiều ứng dụng và công cụ hỗ trợ quản lý tài chính cá nhân được phát triển và phổ biến trên thị trường như Mint, YNAB, Money Lover hay PocketGuard, tuy nhiên trên thực tế không phải người dùng nào cũng khai thác hiệu quả các giải pháp này. Một bộ phận người dùng chỉ sử dụng ứng dụng trong một thời gian ngắn rồi bỏ dở do cảm thấy việc nhập liệu thủ công tốn nhiều thời gian hoặc không mang lại giá trị sử dụng rõ ràng. Bên cạnh đó, phần lớn các ứng dụng hiện tại chủ yếu dừng lại ở chức năng ghi chép và thống kê số liệu thu chi theo ngày, tháng hoặc theo từng danh mục, mà chưa thực sự hỗ trợ người dùng hiểu sâu hơn về hành vi chi tiêu của bản thân.

Các báo cáo mà những ứng dụng này cung cấp thường mang tính tổng hợp, chẳng hạn như tổng chi tiêu trong một khoảng thời gian nhất định hoặc tỷ lệ chi tiêu theo từng nhóm danh mục. Mặc dù những thông tin này là cần thiết, nhưng vẫn chưa đủ để giúp người dùng đưa ra các quyết định tài chính hợp lý và có cơ sở khoa học. Trong nhiều trường hợp, người dùng vẫn phải tự đánh giá đâu là khoản chi tiêu cần thiết, đâu là khoản chi có thể tối ưu, cũng như cách phân bổ ngân sách sao cho phù hợp với mức thu nhập hiện tại. Điều này gây ra không ít khó khăn, đặc biệt đối với những người chưa có nhiều kinh nghiệm trong việc quản lý tài chính cá nhân.

Cùng với sự phát triển mạnh mẽ của trí tuệ nhân tạo và các phương pháp phân tích dữ liệu, nhiều cách tiếp cận mới trong việc hỗ trợ ra quyết định đã được nghiên cứu và ứng dụng trong nhiều lĩnh vực khác nhau. Việc khai thác và phân tích dữ liệu lịch sử cho phép các hệ thống học hỏi từ những thông tin đã có, qua đó nhận diện các quy luật, xu hướng cũng như những mối quan hệ tiềm ẩn trong dữ liệu mà con người khó có thể nhận ra nếu chỉ dựa vào quan sát hoặc tổng hợp thông tin theo cách truyền thống. Khi được ứng dụng vào lĩnh vực tài chính cá nhân, các phương pháp này cho phép việc phân tích dữ liệu thu chi không chỉ dừng lại ở mức thống kê, mà còn mang tính dự báo, phân tích hành vi và hỗ trợ ra quyết định một cách chủ động.

Thực tế cho thấy mỗi cá nhân có thói quen chi tiêu, nhu cầu và mục tiêu tài chính khác nhau. Có người tập trung vào các khoản chi sinh hoạt hằng ngày, trong khi có người lại dành phần lớn ngân sách cho học tập, giải trí hoặc tích lũy cho những kế hoạch dài hạn trong tương lai. Sự khác biệt này khiến các gợi ý tài chính mang tính chung chung khó mang lại hiệu quả đối với tất cả người dùng. Do đó, một hệ thống có khả năng phân tích hành vi chi tiêu của từng cá nhân và đưa ra các gợi ý phù hợp với hoàn cảnh, thói quen và mục tiêu cụ thể sẽ giúp người dùng dễ dàng điều chỉnh cách sử dụng tiền của mình theo hướng hợp lý hơn.

Trên cơ sở khảo sát nhu cầu thực tế của người dùng và phân tích các ứng dụng quản lý tài chính cá nhân hiện có, có thể nhận thấy rằng người dùng không chỉ cần một công cụ ghi nhận thu chi đơn thuần, mà còn mong muốn hệ thống có khả năng hỗ trợ lập kế hoạch tài chính, đặt mục tiêu tiết kiệm, quản lý ngân sách theo từng danh mục và đưa ra cảnh báo khi chi tiêu vượt ngưỡng hoặc có dấu hiệu mất cân đối tài chính. Đồng thời, người dùng ngày càng quan tâm đến các yếu tố như giao diện thân thiện, tính trực quan, khả năng cá nhân hóa và đặc biệt là tính bảo mật của dữ liệu tài chính cá nhân.

Xuất phát từ những vấn đề và nhu cầu nêu trên, đề tài “Nghiên cứu kết hợp thuật toán K-MEANS, ISOLATION FOREST với mô hình LSTM để xây dựng ứng dụng quản lý chi tiêu cá nhân” được thực hiện nhằm giải quyết bài toán quản lý và tối ưu hóa tài chính cá nhân dựa trên dữ liệu thực tế và hành vi chi tiêu của người dùng. Bài toán trọng tâm của đề tài là thu thập, lưu trữ và xử lý dữ liệu thu nhập và chi tiêu cá nhân một cách có hệ thống, từ đó ứng dụng các kỹ thuật học máy và học sâu để phân tích hành vi tài chính, dự báo xu hướng thu chi trong tương lai và phát hiện các khoản chi tiêu bất thường.

Cụ thể, mô hình LSTM được sử dụng để dự báo xu hướng thu nhập và chi tiêu theo chuỗi thời gian, giúp người dùng có cái nhìn tổng quan và chủ động hơn trong việc lập kế hoạch tài chính. Thuật toán K-Means được áp dụng để phân cụm hành vi chi tiêu, từ đó nhận diện các nhóm thói quen chi tiêu đặc trưng của từng người dùng. Bên cạnh đó, mô hình Isolation Forest được sử dụng nhằm phát hiện các giao dịch chi tiêu bất thường hoặc vượt ngưỡng so với hành vi chi tiêu thông thường, qua đó hỗ trợ hệ thống đưa ra các cảnh báo kịp thời.

Dựa trên kết quả phân tích và dự báo, ứng dụng cung cấp các gợi ý chi tiêu, cảnh báo vượt ngân sách và các báo cáo tài chính trực quan dưới dạng biểu đồ, giúp người dùng dễ dàng theo dõi sự biến động của thu nhập và chi tiêu theo từng giai đoạn. Hệ thống được triển khai dưới dạng ứng dụng di động nhằm đáp ứng nhu cầu sử dụng linh hoạt trong đời sống hằng ngày. Việc lựa chọn nền tảng Flutter cho phép ứng dụng hoạt động ổn định trên cả hai hệ điều hành Android và iOS. Dữ liệu tài chính cá nhân được lưu trữ và đồng bộ thông qua Firebase, đảm bảo tính bảo mật, toàn vẹn và khả năng truy cập theo thời gian thực. Các mô hình học máy được tích hợp trực tiếp trên thiết bị thông qua TensorFlow Lite, góp phần giảm sự phụ thuộc vào máy chủ và nâng cao quyền riêng tư cho người dùng.

Thông qua việc xây dựng và triển khai hệ thống này, đề tài hướng đến mục tiêu hỗ trợ người dùng hình thành thói quen quản lý tài chính cá nhân một cách khoa học, chủ động và bền vững hơn. Ứng dụng không chỉ đóng vai trò là công cụ ghi chép tài chính, mà còn trở thành một công cụ hỗ trợ ra quyết định thông minh, giúp người dùng hiểu rõ hơn về hành vi chi tiêu của bản thân, tối ưu hóa ngân sách và xây dựng kế hoạch tài chính phù hợp với mục tiêu trong từng giai đoạn của cuộc sống.

1.2 Mục tiêu nghiên cứu.

Trong bối cảnh chuyển đổi số diễn ra mạnh mẽ cùng với sự phát triển nhanh chóng của trí tuệ nhân tạo, việc quản lý tài chính cá nhân ngày càng đóng vai trò quan trọng và trở thành một nhu cầu thiết yếu đối với mỗi cá nhân. Việc kiểm soát chi tiêu, theo dõi thu nhập, xây dựng kế hoạch tiết kiệm và đầu tư hợp lý không chỉ giúp người dùng duy trì sự ổn định tài chính trong ngắn hạn mà còn góp phần nâng cao khả năng chủ động trong việc định hướng và thực hiện các mục tiêu tài chính dài hạn. Tuy nhiên, trên thực tế, nhiều người dùng hiện nay vẫn gặp khó khăn trong công tác quản lý tài chính cá nhân do thiếu các công cụ hỗ trợ hiệu quả và phù hợp. Điều này dẫn đến tình trạng chi tiêu thiếu kiểm soát, chưa hình thành được kế hoạch tài chính rõ ràng và khó đạt được các mục tiêu tài chính trong tương lai.

Đề tài “Nghiên cứu kết hợp thuật toán K-MEANS, ISOLATION FOREST với mô hình LSTM để xây dựng ứng dụng quản lý chi tiêu cá nhân.” được thực hiện với mục tiêu nghiên

cứu và xây dựng một hệ thống quản lý tài chính cá nhân mang tính toàn diện. Hệ thống được thiết kế nhằm kết hợp giữa giao diện thân thiện, dễ sử dụng với các chức năng quản lý tài chính cơ bản, đồng thời tích hợp khả năng phân tích và gợi ý chi tiêu dựa trên dữ liệu thực tế của người dùng. Thông qua việc ứng dụng các mô hình học máy, hệ thống hướng tới việc hỗ trợ người dùng đưa ra các quyết định tài chính hợp lý hơn, phù hợp với thói quen và điều kiện cá nhân.

Mục tiêu tổng quát của đề tài là xây dựng một hệ thống quản lý tài chính cá nhân hoạt động ổn định trên nền tảng thiết bị di động (Android và iOS), cho phép người dùng ghi chép, theo dõi và phân tích hiệu quả các khoản thu chi, tiết kiệm, đầu tư và nợ. Trên cơ sở dữ liệu thu thập được, hệ thống cung cấp các gợi ý chi tiêu mang tính cá nhân hóa, góp phần hỗ trợ người dùng nâng cao hiệu quả quản lý tài chính và hướng tới việc sử dụng nguồn tài chính một cách hợp lý và bền vững hơn.

Các mục tiêu nghiên cứu cụ thể của đề tài bao gồm:

- Tổng quan về quản lý tài chính cá nhân: Khảo sát các phương pháp, công cụ và ứng dụng hiện có trong lĩnh vực quản lý tài chính cá nhân; đánh giá ưu nhược điểm, khả năng ứng dụng thực tiễn và nhu cầu người dùng. Từ đó, xác định hướng phát triển phù hợp cho một ứng dụng quản lý tài chính hiện đại, mang tính thực tế cao và dễ sử dụng.

- Khảo sát yêu cầu bài toán: Nghiên cứu, phân tích các yêu cầu của người dùng trong việc quản lý tài chính cá nhân, bao gồm các chức năng cần thiết như ghi nhận thu nhập, chi tiêu, theo dõi tiết kiệm, lập kế hoạch tài chính, tạo báo cáo chi tiết và gợi ý chi tiêu. Việc khảo sát này nhằm đảm bảo ứng dụng đáp ứng đúng nhu cầu thực tế và mang lại giá trị sử dụng cao nhất cho người dùng.

- Xây dựng hệ thống quản lý tài chính cá nhân với Flutter và Firebase: Tận dụng ưu điểm của công nghệ Flutter để xây dựng ứng dụng đa nền tảng với hiệu suất cao, giao diện đẹp, mượt mà. Sử dụng Firebase làm cơ sở dữ liệu chính, giúp lưu trữ, đồng bộ hóa dữ liệu người dùng theo thời gian thực và đảm bảo tính an toàn, bảo mật.

- Nghiên cứu ngôn ngữ lập trình Dart và công nghệ Flutter: Áp dụng ngôn ngữ lập trình Dart để xử lý dữ liệu, xây dựng logic ứng dụng, đồng thời tận dụng Flutter để thiết kế giao

diện người dùng trực quan, dễ sử dụng, đảm bảo trải nghiệm thân thiện và nhất quán trên nhiều thiết bị.

- Phân tích và thiết kế hệ thống: Tiến hành phân tích yêu cầu, xác định mô hình dữ liệu, chức năng, quy trình nghiệp vụ và luồng hoạt động trong hệ thống. Thiết kế hệ thống thông qua các sơ đồ UML (Use Case, Activity Diagram, ERD...) để đảm bảo tính rõ ràng, dễ hiểu, dễ mở rộng và thuận tiện trong quá trình phát triển.

- Xây dựng ứng dụng quản lý tài chính cá nhân với các mô-đun chính sau:

- + Mô-đun quản lý người dùng: Cho phép người dùng đăng ký, đăng nhập và quản lý thông tin cá nhân. Hỗ trợ bảo mật tài khoản thông qua xác thực email và đăng nhập Google. Người dùng có thể cập nhật thông tin cá nhân, thay đổi mật khẩu và tùy chỉnh các cài đặt tài chính riêng.

- + Mô-đun quản lý thu nhập và chi tiêu: Hỗ trợ người dùng ghi nhận các khoản thu nhập và chi tiêu hằng ngày, phân loại theo danh mục cụ thể (ăn uống, mua sắm, giải trí, học tập,...). Ứng dụng tự động thống kê, tổng hợp và phân tích các khoản thu chi, giúp người dùng nắm bắt được tình hình tài chính hiện tại.

- + Mô-đun báo cáo tài chính: Cho phép hiển thị các báo cáo trực quan về thu nhập, chi tiêu, tiết kiệm và đầu tư của người dùng theo từng giai đoạn (ngày, tuần, tháng, năm) dựa trên các biểu đồ thống kê như biểu đồ tròn, biểu đồ cột giúp người dùng dễ dàng phân tích xu hướng tài chính, từ đó đưa ra các quyết định chi tiêu hợp lý hơn.

- + Mô-đun gợi ý chi tiêu thông minh: Thực hiện phân tích dữ liệu thu nhập và chi tiêu của người dùng dựa trên dữ liệu lịch sử. Trong đó, thuật toán LSTM được sử dụng để dự báo xu hướng thu/chi trong các khoảng thời gian tiếp theo, làm cơ sở đánh giá mức độ hợp lý của các khoản chi hiện tại. Bên cạnh đó, thuật toán K-Means được áp dụng nhằm phân cụm hành vi chi tiêu, qua đó xác định các nhóm thói quen chi tiêu đặc trưng của người dùng. Đồng thời, thuật toán Isolation Forest được sử dụng để phát hiện các giao dịch chi tiêu bất thường hoặc vượt ngưỡng so với hành vi chi tiêu thông thường. Trên cơ sở kết quả phân tích dữ liệu hệ thống đưa ra các gợi ý chi tiêu và cảnh báo phù hợp nhằm mục đích hỗ trợ người dùng kiểm soát ngân sách, hạn chế chi tiêu không hợp lý và từng bước hình thành thói quen tài chính bền vững.

+ Kiểm thử và đánh giá hệ thống: Tiến hành kiểm thử toàn bộ ứng dụng, bao gồm kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử giao diện và kiểm thử người dùng thực tế. Đánh giá mức độ chính xác của các tính năng, hiệu suất hoạt động, khả năng đồng bộ hóa dữ liệu và chất lượng gợi ý từ AI. Kết quả kiểm thử giúp hoàn thiện và tối ưu hóa ứng dụng trước khi triển khai thực tế.

1.3 Đối tượng và phạm vi nghiên cứu.

Đối tượng nghiên cứu: Nghiên cứu kết hợp thuật toán K-MEANS, ISOLATION FOREST với mô hình LSTM để xây dựng ứng dụng quản lý chi tiêu cá nhân.

Phạm vi nghiên cứu:

- Xây dựng ứng dụng quản lý tài chính cá nhân: Tập trung phát triển ứng dụng trên nền tảng Flutter, một framework đa nền tảng sử dụng ngôn ngữ lập trình Dart, nhằm cung cấp một hệ thống quản lý tài chính cá nhân dễ sử dụng và có khả năng hoạt động mượt mà trên cả thiết bị di động và web.

- Tìm hiểu công nghệ Flutter và ngôn ngữ lập trình Dart: Nghiên cứu và áp dụng các tính năng của Flutter và Dart để xây dựng một ứng dụng với giao diện thân thiện, hiệu suất cao và đáp ứng tốt trên nhiều nền tảng khác nhau, từ Android, iOS đến web. Flutter giúp tối ưu hóa quá trình phát triển nhờ khả năng tạo ra ứng dụng đa nền tảng từ một codebase duy nhất.

- Quản lý cơ sở dữ liệu: Nghiên cứu và thiết kế cơ sở dữ liệu phù hợp để lưu trữ thông tin người dùng và các giao dịch tài chính cá nhân, đảm bảo an toàn và bảo mật dữ liệu. Việc nghiên cứu bao gồm tích hợp các giải pháp cơ sở dữ liệu như Firebase vào ứng dụng Flutter.

- Công nghệ AI/ML: Công nghệ phân tích và gợi ý chi tiêu thông minh được xây dựng dựa trên việc ứng dụng các thuật toán học máy và học sâu phù hợp với đặc thù dữ liệu tài chính cá nhân. Cụ thể: Thuật toán LSTM được sử dụng để phân tích chuỗi thời gian và dự báo xu hướng thu/chỉ của người dùng. Thuật toán K-Means được áp dụng nhằm phân tích và phân cụm hành vi chi tiêu, qua đó nhận diện các nhóm thói quen chi tiêu đặc trưng. Đồng thời, thuật toán Isolation Forest được sử dụng để phát hiện các giao dịch chi tiêu bất thường so với hành vi chi tiêu thông thường. Các mô hình này được tích hợp thông qua TensorFlow

Lite, cho phép triển khai trực tiếp trên thiết bị di động, đảm bảo hiệu suất xử lý, giảm độ trễ và tối ưu tài nguyên hệ thống.

- Phân tích và tối ưu trải nghiệm người dùng về UX/UI: Nghiên cứu các phương pháp xây dựng giao diện người dùng dễ sử dụng, trực quan và thân thiện với người dùng. Điều này bao gồm việc tối ưu các chức năng như nhập liệu tài chính, theo dõi chi tiêu, và lập báo cáo tài chính một cách tiện lợi và nhanh chóng.

1.4 Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

- Tìm hiểu các khái niệm cơ bản về công nghệ Flutter và hệ quản trị cơ sở dữ liệu Firebase: Nghiên cứu về Flutter, framework được sử dụng để phát triển ứng dụng đa nền tảng, và cách tích hợp cơ sở dữ liệu với Firebase, giúp lưu trữ thông tin tài chính cá nhân của người dùng.

- Tìm hiểu các công nghệ và ngôn ngữ lập trình Dart và các ngôn ngữ liên quan đến quản lý cơ sở dữ liệu: Nghiên cứu cách sử dụng ngôn ngữ Dart (ngôn ngữ chính trong Flutter) cùng với các công nghệ web khác để xây dựng ứng dụng quản lý tài chính cá nhân có giao diện thân thiện, hiệu quả và bảo mật cao.

- Tìm hiểu các công cụ lập trình trong Flutter: Nghiên cứu các công cụ, thư viện và framework hỗ trợ phát triển ứng dụng Flutter nhằm xây dựng các chức năng quản lý tài chính cá nhân. Điều này bao gồm cả việc tích hợp các gói quản lý trạng thái, UI, và các thư viện quản lý cơ sở dữ liệu.

- Tìm hiểu hệ quản trị cơ sở dữ liệu Firebase: Nghiên cứu cách tích hợp Firebase vào ứng dụng Flutter để lưu trữ và quản lý thông tin về thu nhập, chi tiêu, tiết kiệm và đầu tư của người dùng, đảm bảo an toàn và hiệu quả khi truy xuất dữ liệu.

- Tìm hiểu các khái niệm cơ bản về AI/ML: Tập trung nghiên cứu và tìm hiểu các khái niệm, nguyên lý cơ bản liên quan đến AI/ML, đặc biệt là các mô hình có khả năng phân tích và học từ dữ liệu tài chính cá nhân. Nội dung nghiên cứu hướng đến việc nắm vững cách thức hoạt động của các mô hình học máy trong việc xử lý dữ liệu theo thời gian, khai thác đặc trưng hành vi người dùng và phát hiện các mẫu bất thường trong chi tiêu. Cụ thể,

tập trung nghiên cứu các nhóm mô hình phù hợp với bài toán tài chính cá nhân như các mô hình chuỗi thời gian, điển hình là thuật toán LSTM, nhằm phục vụ cho việc dự báo xu hướng thu nhập và chi tiêu trong tương lai; thuật toán phân cụm K-Means để phân tích và nhóm các hành vi chi tiêu của người dùng; cùng với đó là thuật toán không giám sát Isolation Forest nhằm nhận diện các giao dịch bất thường hoặc dấu hiệu chi tiêu không hợp lý. Trên cơ sở nghiên cứu này, đề tài làm rõ khả năng ứng dụng các mô hình AI/ ML vào việc phân tích hành vi chi tiêu và dự đoán xu hướng tài chính cá nhân, từ đó hỗ trợ hệ thống đưa ra các gợi ý chi tiêu phù hợp và hiệu quả hơn cho người dùng.

- Tìm hiểu các công cụ và thư viện AI trên nền tảng di động: Tiến hành nghiên cứu các công cụ hỗ trợ việc triển khai mô hình học máy trên thiết bị di động, trong đó tập trung tìm hiểu TensorFlow Lite. Công cụ này được sử dụng nhằm xây dựng và tích hợp các mô hình phân tích dữ liệu tài chính trực tiếp vào ứng dụng Flutter, giúp quá trình xử lý diễn ra nhanh chóng và đảm bảo tối ưu tài nguyên trên thiết bị di động.

Phương pháp thực nghiệm:

- Khảo sát và phân tích yêu cầu: Tiến hành khảo sát nhu cầu và yêu cầu của người dùng trong việc quản lý và kiểm soát chi tiêu cá nhân. Quá trình khảo sát tập trung tìm hiểu các chức năng cần thiết của ứng dụng như theo dõi thu nhập và chi tiêu, quản lý ngân sách theo từng danh mục, xem báo cáo và thống kê tài chính theo thời gian, cũng như các tính năng gợi ý chi tiêu thông minh dựa trên hành vi sử dụng của người dùng. Bên cạnh đó, việc phân tích yêu cầu còn bao gồm khảo sát và đánh giá các ứng dụng quản lý tài chính hiện có trên thị trường nhằm nắm bắt xu hướng phát triển, giao diện người dùng và các chức năng quan trọng cần được triển khai trong ứng dụng, từ đó làm cơ sở đề xuất giải pháp phù hợp và xây dựng hệ thống đáp ứng tốt hơn về nhu cầu thực tế của người dùng.

- Thiết kế cơ sở dữ liệu: Quá trình thiết kế cơ sở dữ liệu được thực hiện nhằm xây dựng nền tảng lưu trữ thông tin cho hệ thống ứng dụng thông minh hỗ trợ quản lý và gợi ý chi tiêu cá nhân. Cơ sở dữ liệu được thiết kế bao gồm các bảng dữ liệu, mối quan hệ giữa các bảng và các trường thông tin cần thiết để lưu trữ và quản lý dữ liệu liên quan đến thu nhập, chi tiêu, ngân sách cũng như các danh mục đầu tư của người dùng. Trong quá trình thiết kế, cấu trúc dữ liệu được xây dựng theo hướng khoa học và hợp lý, giúp dữ liệu được tổ chức

rõ ràng và thuận tiện cho việc truy xuất, xử lý và phân tích hành vi chi tiêu. Bên cạnh đó, cơ sở dữ liệu cũng được tối ưu hóa nhằm nâng cao hiệu năng hệ thống, đảm bảo tốc độ truy vấn ổn định và tăng cường tính an toàn, bảo mật cho dữ liệu người dùng trong suốt quá trình vận hành ứng dụng.

- Xác định kiến trúc hệ thống: Kiến trúc tổng thể của ứng dụng được xác định và xây dựng ngay từ đầu nhằm làm nền tảng cho toàn bộ quá trình phát triển về sau. Hệ thống được thiết kế theo hướng mô-đun, bao gồm các chức năng chính như quản lý thu nhập, quản lý chi tiêu, quản lý ngân sách, báo cáo tài chính, gợi ý chi tiêu và phân tích hành vi người dùng, qua đó giúp phân tách rõ ràng từng thành phần trong ứng dụng. Cách tiếp cận này giúp quá trình phát triển và triển khai các chức năng diễn ra thuận lợi hơn, đồng thời hỗ trợ hiệu quả cho công tác bảo trì, nâng cấp và mở rộng hệ thống trong tương lai. Nhờ có kiến trúc được tổ chức rõ ràng và hợp lý, ứng dụng có thể dễ dàng bổ sung các tính năng mới mà không gây ảnh hưởng đến các chức năng đã có, qua đó đảm bảo tính ổn định và hiệu quả trong suốt quá trình vận hành.

- Phát triển giao diện người dùng: Xây dựng UI hấp dẫn và thân thiện, giúp người dùng dễ dàng quản lý tài chính cá nhân. Các thành phần giao diện được thiết kế sao cho bố cục rõ ràng, màu sắc hài hòa và phù hợp với thói quen sử dụng của người dùng. Ứng dụng sử dụng Flutter để xây dựng các giao diện có tính tương tác cao, đảm bảo khả năng hiển thị nhất quán và ổn định trên cả hai nền tảng Android và iOS. Thông qua việc tối ưu trải nghiệm người dùng và khả năng tương thích đa nền tảng, giao diện ứng dụng góp phần làm nâng cao hiệu quả sử dụng và mức độ hài lòng của người dùng trong quá trình quản lý tài chính cá nhân.

- Đánh giá và kiểm thử: Tiến hành kiểm thử một cách toàn diện nhằm đánh giá mức độ ổn định và độ chính xác của các chức năng đã xây dựng. Quá trình kiểm thử tập trung vào việc kiểm tra hoạt động của từng chức năng trong các tình huống sử dụng thực tế, đảm bảo ứng dụng vận hành đúng theo yêu cầu đề ra và không phát sinh lỗi ảnh hưởng đến trải nghiệm người dùng. Bên cạnh kiểm thử chức năng, ứng dụng còn được kiểm thử về hiệu năng nhằm đánh giá khả năng xử lý và phản hồi của hệ thống khi làm việc với lượng dữ liệu lớn hoặc khi có nhiều thao tác liên tiếp. Trong suốt quá trình kiểm thử, các ý kiến và

phản hồi từ người dùng thử nghiệm được ghi nhận, phân tích và sử dụng làm cơ sở để điều chỉnh, cải thiện và hoàn thiện ứng dụng, giúp sản phẩm đáp ứng tốt hơn nhu cầu thực tế của người dùng trước khi đưa vào sử dụng chính thức.

- Triển khai: Sau khi hoàn thành việc phát triển và kiểm thử, triển khai ứng dụng trên các nền tảng di động (Android, iOS). Quá trình triển khai bao gồm cài đặt, cấu hình hệ thống, hướng dẫn người dùng cách sử dụng ứng dụng và cung cấp nhu cầu hỗ trợ kỹ thuật nếu cần thiết.

- Đánh giá và cải tiến: Sau khi hệ thống được triển khai, tiến hành thu thập phản hồi từ người dùng, đánh giá hiệu suất ứng dụng và tìm kiếm các cải tiến. Dựa trên phản hồi này, thực hiện các điều chỉnh và cập nhật cần thiết nhằm nâng cao trải nghiệm người dùng và tối ưu hiệu suất hệ thống.

1.5 Cấu trúc báo cáo bài tập lớn

1.5.1 Chương 1: Tổng quan bài toán

- Tổng quan về đề tài
- Mục tiêu nghiên cứu
- Đối tượng và phạm vi nghiên cứu
- Phương pháp nghiên cứu

1.5.2 Chương 2: Cơ sở lý thuyết

- Tổng quan về quản lý tài chính cá nhân
- Nền tảng công nghệ phát triển ứng
- Trí tuệ nhân tạo và học máy trong phân tích dữ liệu

1.5.3 Chương 3: Phân tích thiết kế hệ thống

- Lựa chọn mô hình phát triển
- Thiết kế chức năng của hệ thống
- Thiết kế cơ sở dữ liệu

1.5.4 Chương 4: Thiết kế và cài đặt chương trình

- Kiến trúc hệ thống
- Thiết kế giao diện

1.5.5 Chương 5: Kết luận và kiến nghị

- Kết quả đạt được
- Ưu điểm hệ thống
- Hạn chế của hệ thống

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về quản lý tài chính cá nhân

2.1.1 Khái niệm

Quản lý tài chính cá nhân là quá trình quản lý và kiểm soát các khoản thu nhập, chi tiêu, đầu tư và tiết kiệm của một cá nhân hoặc gia đình. Nó bao gồm việc theo dõi và phân tích các khoản thu nhập hàng tháng, xác định và ưu tiên các mục chi tiêu, tạo ngân sách, quản lý nợ nần, đầu tư thông minh và xây dựng quỹ tiết kiệm [1].

2.1.2 Lợi ích của việc quản lý tài chính cá nhân

Quản lý tài chính cá nhân mang lại nhiều lợi ích quan trọng [1][2]:

- Kiểm soát tài chính: Quản lý tài chính cá nhân giúp bạn có kiểm soát hoàn toàn về thu chi của mình. Bạn biết được tiền vào và tiền ra của mình, từ đó có thể xác định được nguồn thu nhập và các khoản chi tiêu cần thiết.
- Tiết kiệm và đầu tư: Khi bạn quản lý tài chính cá nhân, bạn có thể xác định được các khoản tiền dư để tiết kiệm hoặc đầu tư. Điều này giúp bạn tích lũy dần dần để có khả năng phục vụ cho các mục tiêu ngắn hạn và dài hạn.
- Tránh nợ nần: Quản lý tài chính cá nhân giúp bạn tránh rơi vào cảnh nợ nần không kiểm soát. Bằng cách theo dõi kỹ thuật số của bạn, bạn có thể điều chỉnh chi tiêu sao cho phù hợp với thu nhập hiện có và tránh việc sử dụng quá mức thẻ tín dụng hoặc vay nợ.
- Xây dựng tương lai tài chính: Quản lý tài chính cá nhân giúp bạn xác định và đạt được các mục tiêu tài chính trong cuộc sống. Bạn có thể lập kế hoạch để mua nhà, mua xe, trang trải chi phí học phí cho con cái, hay chuẩn bị cho việc hưu trí.
- Giảm căng thẳng: Khi bạn có kiểm soát về tài chính cá nhân, bạn sẽ cảm thấy tự tin và yên tâm hơn về khả năng quản lý tiền bạc của mình. Điều này giúp giảm căng thẳng và lo âu liên quan đến tiền bạc trong cuộc sống hàng ngày.

2.1.3 Mục đích của quản lý tài chính cá nhân

Mục đích của quản lý tài chính cá nhân là giúp bạn kiểm soát và quản lý tài chính cá nhân một cách hiệu quả. Đây là quá trình theo dõi, phân tích và điều chỉnh thu nhập, chi tiêu, đầu tư và tiết kiệm của bạn để đạt được các mục tiêu tài chính cá nhân [1][2]:

Bằng cách theo dõi thu nhập và chi tiêu hàng ngày, bạn có thể xác định được số tiền có thể tiết kiệm và đầu tư vào các cơ hội sinh lợi cao hơn.

Quản lý tài chính cá nhân giúp bạn xác định rõ ràng các mục tiêu ngắn hạn và dài hạn, ví dụ như: Mua nhà, đi du lịch hay chuẩn bị cho tuổi già.

Khi quản lý hiệu quả thu nhập và chi tiêu, bạn có thể giảm thiểu nợ phải trả hàng tháng hoặc thanh toán nhanh hơn.

Khi bạn có kế hoạch quản lý tài chính cá nhân, bạn sẽ có khả năng ứng phó tốt hơn với những tình huống khó khăn.

Quản lý tài chính cá nhân giúp bạn xây dựng một quỹ tiết kiệm và đảm bảo rằng bạn có đủ tiền để chi trả các khoản chi phí cần thiết trong cuộc sống hàng ngày.

2.2 Nền tảng công nghệ phát triển ứng dụng

2.2.1 Tổng quan về ngôn ngữ lập trình Dart

Khái niệm [3]:

- Dart là một ngôn ngữ lập trình máy tính được sử dụng để tạo các ứng dụng phần mềm. Nó được thiết kế để dễ học và dễ hiểu nhờ cú pháp và cấu trúc đơn giản.

- Dart là ngôn ngữ hướng đối tượng, tập trung vào việc tạo ra các phần mã có thể tái sử dụng được gọi là đối tượng. Nó có thể được sử dụng để xây dựng các chương trình độc lập hoặc để phát triển ứng dụng bằng các framework như Flutter.

Các đặc trưng của Dart [3]:

- Hướng đối tượng (Object-Oriented): Dart là ngôn ngữ lập trình hướng đối tượng, hỗ trợ các khái niệm như lớp, đối tượng, kế thừa và đa hình.

- Bất đồng bộ (Asynchronous): Dart có hỗ trợ tích hợp cho lập trình bất đồng bộ. Nghĩa là thực hiện các tác vụ đồng thời mà không chặn việc thực thi các mã khác, giúp cải thiện khả năng phản hồi và hiệu quả của ứng dụng, đặc biệt đối với các tác vụ như yêu cầu mạng, thao tác tệp và tương tác người dùng.

- Hỗ trợ đa nền tảng (Multi-Platform Support): Dart được thiết kế để hỗ trợ nhiều nền tảng, cho phép xây dựng ứng dụng cho các hệ điều hành khác nhau bao gồm di động (iOS và Android), web và máy tính để bàn.

- Tích hợp Flutter (Flutter Integration): Dart là ngôn ngữ lập trình chính cho Flutter, một framework phổ biến để xây dựng giao diện người dùng. Sự kết hợp giữa Flutter và Dart cho phép các nhà phát triển tạo ra các ứng dụng trực quan hấp dẫn và phản hồi nhanh cho di động, web và máy tính để bàn từ một codebase duy nhất.

2.2.2 Tổng quan về Framework Flutter

Khái niệm [4]:

- Flutter là một bộ công cụ và framework phát triển giao diện người dùng đa nền tảng, cho phép xây dựng các ứng dụng có giao diện đẹp và hiệu suất cao, chạy trên nhiều hệ điều hành khác nhau như iOS, Android, Web và máy tính để bàn. Flutter là một nền tảng miễn phí, mã nguồn mở, được khởi xướng và bảo trợ bởi Google, đồng thời liên tục được phát triển và duy trì bởi đội ngũ kỹ sư của Google cùng với sự đóng góp của cộng đồng lập trình viên trên toàn thế giới.

- Một trong những ưu điểm nổi bật của Flutter là khả năng biên dịch trực tiếp mã nguồn thành ứng dụng gốc (native), không phụ thuộc vào WebView hay các cầu nối JavaScript, từ đó giúp ứng dụng đạt hiệu suất cao và phản hồi nhanh. Flutter cho phép lập trình viên viết mã nguồn một lần và triển khai trên nhiều nền tảng khác nhau, bao gồm ứng dụng web, ứng dụng iOS và ứng dụng Android, góp phần giảm chi phí phát triển và rút ngắn thời gian đưa sản phẩm vào thực tế. Nhờ những đặc điểm này, Flutter hiện đang được hàng nghìn nhà phát triển và tổ chức trên toàn cầu tin dùng để xây dựng các ứng dụng thương mại và sản phẩm thực tế.

Các đặc trưng và cách hoạt động của Flutter [4]:

- Flutter là một framework UI reactive, declarative: cung cấp một ánh xạ từ trạng thái ứng dụng sang trạng thái giao diện và framework sẽ tự động cập nhật giao diện khi trạng thái ứng dụng thay đổi.

- Widget: widget là những khối xây dựng cơ bản của giao diện người dùng ứng dụng, Mỗi widget là một khai báo bất biến của một phần giao diện người dùng. Widget lồng ghép vào widget cha của nó và có thể nhận ngữ cảnh từ đó. Các ứng dụng cập nhật giao diện người dùng bằng cách yêu cầu framework thay thế một widget trong hệ thống phân cấp bằng một widget khác; sau đó, framework sẽ so sánh các widget mới và cũ để cập nhật giao diện người dùng một cách hiệu quả.

- Ngôn ngữ lập trình chính là Dart: Dart là ngôn ngữ lập trình chính cho Flutter. Framework Flutter được viết bằng ngôn ngữ Dart và cung cấp một bộ thư viện phong phú cho nền tảng, bố cục và nền tảng.

- Flutter Engine: Ở cốt lõi của Flutter là Flutter Engine, được viết chủ yếu bằng C++. Engine này chịu trách nhiệm raster hóa các cảnh đã tổng hợp khi một khung hình mới cần được vẽ. Nó cung cấp triển khai cấp thấp của API cốt lõi của Flutter, bao gồm đồ họa (thông qua Impeller hoặc Skia), bố cục văn bản, I/O tệp và mạng, hỗ trợ khả năng truy cập, kiến trúc plugin, và thời gian chạy Dart cùng bộ công cụ biên dịch. Mã Dart của Flutter được biên dịch thành mã gốc, sử dụng Impeller để hiển thị.

2.2.3 Tổng quan về nền tảng Firebase

Khái niệm [5]:

- Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.

- Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.

Một số dịch vụ Firebase [5]:

- Realtime Database: Realtime Database là một cơ sở dữ liệu thời gian thực. Ngay sau khi bạn đăng ký tài khoản trên Firebase, bạn sẽ nhận được Realtime Database được lưu trữ dưới dạng JSON và được đồng bộ hóa theo thời gian thực đối với mọi kết nối.

- Authentication: Authentication là tính năng giúp xác thực danh tính của người dùng ứng dụng. Firebase cung cấp các bước xác thực thông qua Email, Facebook, Twitter, GitHub hay Google. Điều này giúp cho các thông tin cá nhân của khách hàng được bảo vệ một cách tốt nhất, hạn chế được tình trạng bị hacker đánh cắp. Đồng thời việc xác thực danh tính qua Firebase sẽ giúp người dùng tiếp cận sản phẩm nhanh chóng và an toàn hơn.

- Cloud Storage: Cloud Storage là tính năng cho phép lưu trữ và quản lý nội dung đã tạo ra như ảnh, video, nội dung, văn bản, ... Firebase Storage cung cấp các API hỗ trợ bạn upload và download các file từ ứng dụng một cách trơn tru mà không cần quan tâm đến chất lượng đường truyền mạng với độ bảo mật cao.

- Cloud Firestore: Cloud Firestore được phát triển từ tính năng Realtime Database. Trải qua nhiều lần nâng cấp, Cloud Firestore có giao diện trực quan và khả năng mở rộng ưu việt hơn so với Realtime Database. Tính năng này của Firebase giúp đồng bộ mọi dữ liệu trên các ứng dụng thông qua việc đăng ký thời gian thực và cung cấp hỗ trợ ngoại tuyến cho thiết bị di động cũng như website.

- Analytics: Analytics giúp bạn có thể phân tích hành vi của người sử dụng ứng dụng của bạn. Qua đó, bạn sẽ biết được khách hàng thường xuyên truy cập tính năng nào và các thông tin về hiệu quả quảng cáo, tình trạng trả phí, ... để có thể đưa ra được chiến lược phát triển phù hợp. Để thực hiện tính năng Analytics của Firebase, bạn cần cài đặt SDK.

2.3 Trí tuệ nhân tạo và học máy trong phân tích dữ liệu

2.3.1 Tổng quan về ngôn ngữ lập trình Python

Khái niệm [6]:

- Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và ML. Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Python trong AI và ML [7]:

- Python được xem là một trong những ngôn ngữ lập trình phù hợp nhất cho các bài toán AI và ML. Điều này xuất phát từ nhiều lý do: trước hết, cú pháp của Python đơn giản

và dễ đọc, giúp lập trình viên tập trung vào giải quyết vấn đề thay vì phải xử lý cú pháp phức tạp.

- Python sở hữu một hệ sinh thái thư viện đồ sộ bao gồm các thư viện chuyên về AI và học máy như TensorFlow, Keras, Scikit-learn, NumPy và Pandas, hỗ trợ xử lý dữ liệu, huấn luyện mô hình và trực quan hóa kết quả. Ngoài ra, Python có cộng đồng phát triển lớn mạnh, dễ dàng tìm kiếm tài nguyên, tài liệu và sự hỗ trợ, tạo điều kiện thuận lợi cho cả người mới bắt đầu và nhà phát triển chuyên sâu trong lĩnh vực AI. Nhờ những ưu điểm này, Python trở thành công cụ phổ biến trong các nghiên cứu và triển khai học máy, từ việc xử lý dữ liệu đến xây dựng các mô hình dự đoán phức tạp.

2.3.2 Tổng quan về AI

Khái niệm [8]:

AI là lĩnh vực khoa học máy tính tập trung vào việc phát triển các hệ thống có khả năng thực hiện các nhiệm vụ đòi hỏi trí thông minh của con người như nhận diện giọng nói, hiểu ngôn ngữ tự nhiên, và ra quyết định.

Một số ứng dụng AI trong cuộc sống [8]:

- Chẩn đoán bệnh: Các hệ thống AI có thể phân tích hình ảnh y khoa để phát hiện sớm các bệnh lý như ung thư.
- Dự đoán giá bất động sản: AI có thể phân tích dữ liệu thị trường và các yếu tố khác để dự đoán giá trị bất động sản.
- Phát hiện gian lận thẻ tín dụng: Các thuật toán AI giúp phát hiện các giao dịch đáng ngờ và ngăn chặn gian lận thẻ tín dụng.

2.3.3 Tổng quan về ML

Khái niệm [9]

- ML là một nhánh của AI cho phép máy tính học từ dữ liệu và cải thiện hiệu suất của mình theo thời gian mà không cần phải lập trình cụ thể cho từng nhiệm vụ.

Quá trình máy học bao gồm ba giai đoạn chính [9]:

- Đào tạo (Training): Máy học từ một lượng lớn dữ liệu để nhận biết các mẫu và mối quan hệ trong dữ liệu.

- Dự đoán (Prediction): Máy sử dụng những gì đã học để đưa ra dự đoán về dữ liệu mới.
- Ra quyết định (Decision Making): Dựa trên dự đoán, máy ra quyết định hoặc thực hiện hành động mà không cần sự can thiệp của con người.

Yêu cầu để học máy hiệu quả [9]:

- Lượng dữ liệu đào tạo lớn: Máy cần một lượng lớn dữ liệu để học và nhận biết các mẫu một cách chính xác.
- Sức mạnh tính toán đáng kể: Các thuật toán học máy yêu cầu sức mạnh tính toán lớn để xử lý và phân tích dữ liệu.
- Thuật toán mạnh mẽ: Các thuật toán học máy cần phải mạnh mẽ và hiệu quả để có thể học và cải thiện từ dữ liệu.

2.3.4 Thuật toán K-Means

Khái niệm [11]:

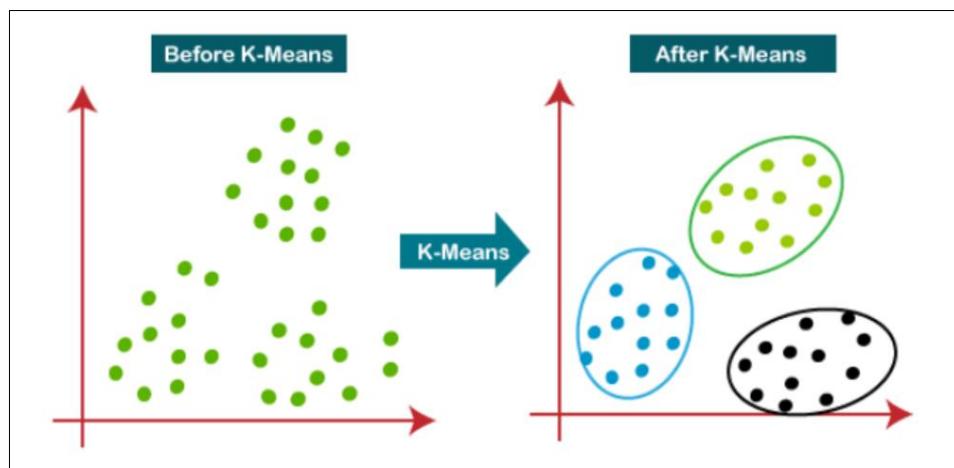
- K-Means là một thuật toán học máy không giám sát (Unsupervised Learning), được sử dụng để phân chia một tập dữ liệu chưa được gán nhãn thành các nhóm (cụm) dựa trên mức độ tương đồng giữa các điểm dữ liệu. Mỗi cụm được biểu diễn bởi một điểm trung tâm, gọi là tâm cụm (centroid).
- Thuật toán K-Means yêu cầu người dùng xác định trước số lượng cụm cần phân chia, ký hiệu là k . Mục tiêu của thuật toán là tối ưu hóa việc phân nhóm sao cho các điểm dữ liệu trong cùng một cụm có mức độ tương đồng cao nhất, trong khi các cụm khác nhau có sự khác biệt rõ rệt.

Nguyên lý hoạt động K-Means [11]:

- Thuật toán K-Means hoạt động dựa trên việc phân loại các điểm dữ liệu vào các cụm thông qua một phép đo khoảng cách toán học, phổ biến nhất là khoảng cách Euclidean, giữa mỗi điểm dữ liệu và tâm của các cụm.
- Mục tiêu chính của thuật toán là tối thiểu hóa tổng bình phương khoảng cách giữa các điểm dữ liệu và tâm cụm mà chúng được gán vào, từ đó đảm bảo tính chặt chẽ và hợp lý của các cụm hình thành.

Quy trình hoạt động của thuật toán K-Means bao gồm các bước chính sau [10]:

- Bước đầu tiên là khởi tạo k tâm cụm, trong đó: k là số lượng cụm được chọn cho một bộ dữ liệu cụ thể. Phương pháp này sử dụng hoặc là chọn ngẫu nhiên, hoặc là phương pháp lấy mẫu tâm cụm ban đầu.
- Bước tiếp theo bao gồm một quá trình lặp lại hai bước dựa trên thuật toán học máy kỳ vọng tối đa (Expectation Maximization). Bước kỳ vọng gán mỗi điểm dữ liệu vào tâm cụm gần nhất dựa trên khoảng cách (thường là khoảng cách Euclide).
- Bước tối đa hóa tính toán trung bình của tất cả các điểm trong mỗi cụm và gán lại tâm của cụm đó. Quá trình này được lặp lại cho đến khi vị trí của các tâm cụm hội tụ hoặc số lần lặp tối đa đã được thực hiện.



Hình 2.1 Nguyên lý hoạt động K-means

Đánh giá ưu điểm K-Means [10]:

- Tính đơn giản và trực quan: K-Means nổi bật là một trong các thuật toán phân cụm dễ nắm bắt nhất. Cơ chế hoạt động của nó rất minh bạch và việc cài đặt tương đối dễ dàng.
- Hiệu suất tính toán tốt: Thuật toán này tỏ ra hiệu quả khi có khả năng xử lý các tập dữ liệu có kích thước đáng kể mà không đòi hỏi chi phí tính toán quá cao, đặc biệt khi số lượng cụm k được chọn là nhỏ.
- Khả năng mở rộng: Nhờ vào đặc tính tuyến tính vốn có, K-Means có khả năng mở rộng rất tốt để làm việc với những bộ dữ liệu khổng lồ, thậm chí có thể áp dụng hiệu quả trên các tập dữ liệu chứa hàng triệu điểm.
- Tính linh hoạt: K-Means có thể được ứng dụng trên nhiều dạng dữ liệu đa dạng, không chỉ giới hạn ở dữ liệu dạng số mà còn bao gồm cả dữ liệu thuộc tính.

Đánh giá nhược điểm K-Means [10]:

- Yêu cầu xác định trước số cụm (k): Một trong những nhược điểm đáng kể nhất của K-Means là việc bắt buộc người dùng phải cung cấp số lượng cụm k mong muốn trước khi thuật toán bắt đầu chạy. Đây có thể là một thách thức nếu không có thông tin tiên nghiệm về số cụm tối ưu cho dữ liệu.

- Độ nhạy với điểm khởi tạo tâm cụm: Kết quả cuối cùng của thuật toán K-Means chịu ảnh hưởng lớn từ vị trí của các tâm cụm được chọn ban đầu. Nếu việc khởi tạo không tốt, thuật toán có nguy cơ bị kẹt lại ở một giải pháp tối ưu cục bộ, thay vì tìm ra giải pháp tối ưu toàn cục.

- Giả định về hình dạng cụm: K-Means hoạt động hiệu quả nhất khi các cụm dữ liệu có dạng hình cầu hoặc gần cầu và kích thước tương đương nhau. Thuật toán gặp khó khăn trong việc nhận diện chính xác các cụm có hình dạng phức tạp, phi tuyến tính hoặc kích thước không đồng đều.

- Nhạy cảm với dữ liệu nhiễu và ngoại lai: Thuật toán K-Means rất dễ bị ảnh hưởng bởi các điểm dữ liệu ngoại lai. Sự hiện diện của các điểm này có thể làm lệch vị trí của tâm cụm, kéo chúng ra xa khỏi trung tâm thực sự của nhóm dữ liệu mà chúng thuộc về.

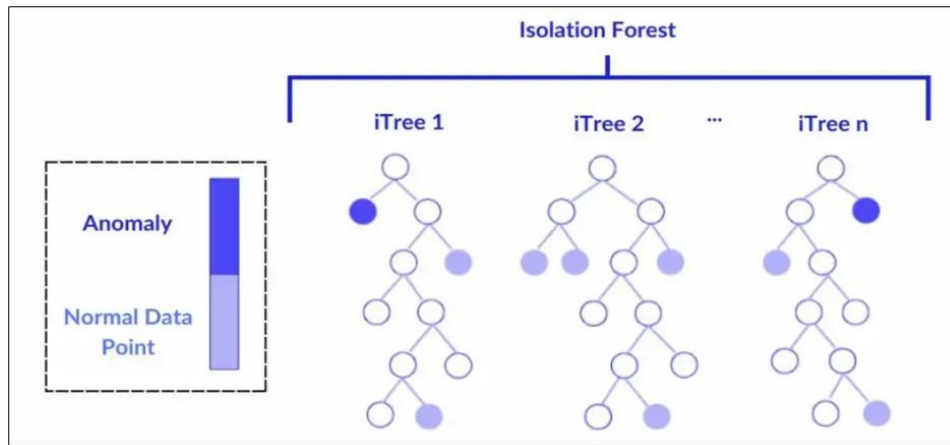
2.3.5 Thuật toán Isolation Forest

Khái niệm [12][13]:

- Isolation Forest là một thuật toán mạnh mẽ và hiệu quả được thiết kế rõ ràng để phát hiện sự bất thường. Isolation Forest hoạt động theo một nguyên tắc độc đáo: nó cô lập các điểm bất thường thay vì lập hồ sơ các điểm dữ liệu trung bình. Điều này làm cho nó đặc biệt hiệu quả trong việc xác định các mẫu hiếm và bất thường trong dữ liệu.

Nguyên lý hoạt động Isolation Forest:

- Isolation Forest hoạt động theo nguyên tắc độc đáo: cô lập các điểm bất thường bằng cách khai thác các đặc điểm riêng biệt của chúng thay vì lập mô hình dữ liệu thông thường. Cách tiếp cận trực quan và hiệu quả này khiến nó trở thành lựa chọn phổ biến cho các nhiệm vụ phát hiện sự bất thường. Dưới đây là một cái nhìn chi tiết về cách nó hoạt động [13].



Hình 2.2 Nguyên lý hoạt động Isolation Forest

Xây dựng Isolation Forest [12]:

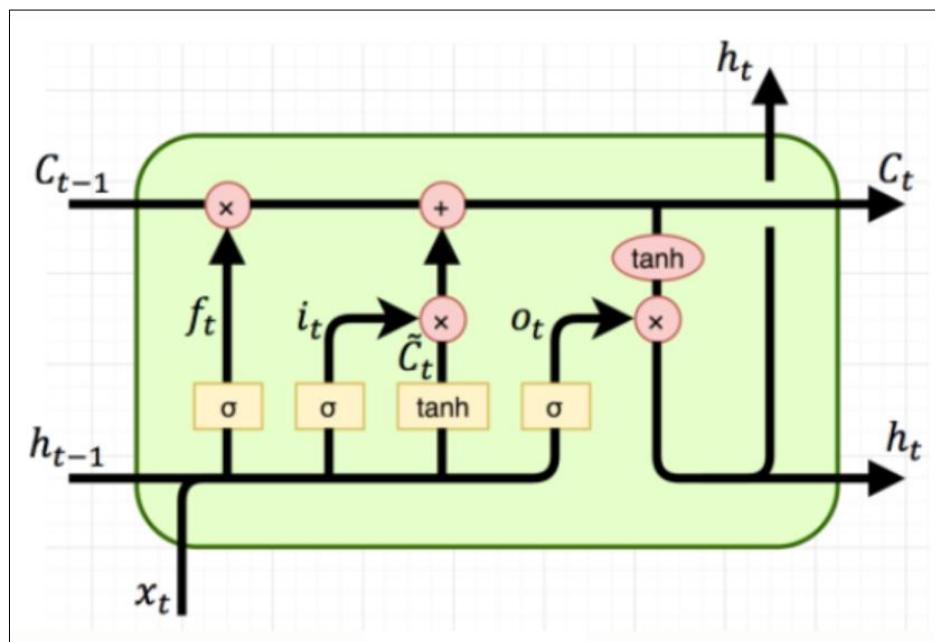
- Cốt lõi của thuật toán Rừng cách ly nằm ở việc sử dụng cây cách ly. Những cây này được xây dựng để cô lập các điểm dữ liệu thông qua phân vùng đệ quy.
- Lấy mẫu con ngẫu nhiên: Thuật toán bắt đầu bằng cách tạo nhiều cây cách ly bằng cách sử dụng các tập hợp con dữ liệu ngẫu nhiên. Mỗi tập hợp con có thể là một phần của tổng tập dữ liệu, điều này đảm bảo thuật toán vẫn hoạt động hiệu quả về mặt tính toán.
- Chia ngẫu nhiên: Đối với mỗi cây, việc xây dựng bắt đầu bằng cách chọn ngẫu nhiên một đối tượng từ dữ liệu và sau đó chọn giá trị phân chia ngẫu nhiên trong phạm vi của đối tượng địa lý đó. Sự phân chia này chia dữ liệu thành hai phần.
- Phân vùng đệ quy: Quá trình chọn ngẫu nhiên các đặc trưng và phân chia tiếp tục theo cách đệ quy. Cây phát triển cho đến khi mỗi điểm dữ liệu được tách biệt trong nút lá của chính nó hoặc đạt đến độ sâu cây tối đa. Tính ngẫu nhiên trong việc lựa chọn và phân chia đặc trưng đảm bảo rằng các cây không thiên về bất kỳ cấu trúc dữ liệu cụ thể nào.

2.3.6 Mạng nơ-ron hồi tiếp LSTM

Khái niệm [14]:

- LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy RNN. Kiến trúc này được tạo ra nhằm mục đích xử lý và ghi nhớ thông tin hiệu quả qua các chuỗi dữ liệu có độ dài lớn [9].
- Trong khi các mô hình RNN truyền thống thường gặp khó khăn khi cần lưu giữ thông tin trong thời gian dài do ảnh hưởng của hiện tượng suy giảm độ dốc (vanishing gradient),

LSTM với cấu trúc độc đáo của mình lại có khả năng duy trì thông tin qua các khoảng thời gian kéo dài, qua đó giải quyết được vấn đề cổ hữu này.



Hình 2.3 Cấu trúc LSTM

Cấu trúc của LSTM [14]:

- Cổng đầu vào (Input Gate): Bộ phận này có nhiệm vụ kiểm soát và quyết định xem những phần thông tin nào từ đầu vào tại bước thời gian hiện tại nên được thêm vào và lưu trữ trong trạng thái của ô nhớ.

- Cổng quên (Forget Gate): Có chức năng xác định và quyết định xem phần thông tin nào trong trạng thái ô nhớ từ bước trước cần được loại bỏ hoặc “quên đi” ở bước hiện tại.

- Cổng đầu ra (Output Gate): Bộ phận này điều tiết việc thông tin nào từ trạng thái ô nhớ (sau khi đã được xử lý) sẽ được phép truyền ra ngoài dưới dạng kết quả đầu ra (hoặc trạng thái ẩn) của đơn vị LSTM tại bước thời gian đó.

Quy trình hoạt động cốt lõi diễn ra như sau [14]:

- Bước 1: Tiếp nhận và xử lý thông tin đầu vào: Tại mỗi bước thời gian, đơn vị LSTM nhận dữ liệu đầu vào và sử dụng các cổng để đánh giá, lọc và xác định xem những thông tin nào là quan trọng và cần được duy trì cho các bước xử lý tiếp theo.

- Bước 2: Cập nhật trạng thái của ô nhớ: Dựa trên tín hiệu điều khiển từ các cổng (cổng quên và cổng đầu vào), trạng thái bên trong của ô nhớ (cell state) sẽ được cập nhật. Cổng

quên loại bỏ những thông tin không còn cần thiết từ trạng thái trước, trong khi cổng đầu vào cho phép bổ sung những thông tin mới, phù hợp vào trạng thái hiện tại.

- Bước 3: Tạo và xuất thông tin đầu ra: Cuối cùng, thông qua cổng đầu ra, đơn vị LSTM sẽ tính toán và tạo ra kết quả đầu ra (hoặc trạng thái ẩn để truyền cho bước tiếp theo) tại bước thời gian đó, dựa trên trạng thái ô nhớ đã được cập nhật và lọc.

CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Lựa chọn mô hình phát triển

Trong quá trình xây dựng ứng dụng quản lý tài chính cá nhân, việc lựa chọn mô hình phát triển phần mềm có ý nghĩa đặc biệt quan trọng, bởi nó ảnh hưởng trực tiếp đến cách thức tổ chức công việc, triển khai các chức năng, quản lý tiến độ thực hiện cũng như đảm bảo chất lượng tổng thể của hệ thống. Một mô hình phát triển phù hợp sẽ giúp quá trình xây dựng phần mềm được triển khai một cách có hệ thống, hạn chế các rủi ro phát sinh và nâng cao hiệu quả trong từng giai đoạn phát triển. Vì vậy, trước khi tiến hành thiết kế và cài đặt hệ thống, nhóm thực hiện đã nghiên cứu, phân tích và so sánh một số mô hình phát triển phần mềm phổ biến hiện nay. Trên cơ sở đặc điểm của đề tài, phạm vi nghiên cứu và yêu cầu của ứng dụng quản lý tài chính cá nhân, đã quyết định lựa chọn mô hình Waterfall để áp dụng trong quá trình thực hiện đề tài.

Mô hình Waterfall hay còn gọi là mô hình thác nước. Được biết tới là một trong những mô hình quản lý dự án dễ hiểu nhất hiện nay, mô hình Waterfall là một phương pháp quản lý dự án dựa trên quy trình thiết kế tuần tự và liên tiếp. Trong mô hình Waterfall, các giai đoạn của dự án được thực hiện lần lượt và nối tiếp nhau. Giai đoạn mới chỉ được bắt đầu khi giai đoạn trước nó đã được hoàn thành [10].

Việc áp dụng mô hình Waterfall được khuyến khích khi người thực hiện nắm rõ yêu cầu của dự án tốt nhất, đòi hỏi về tính rõ ràng và tính ổn định cao như [10]:

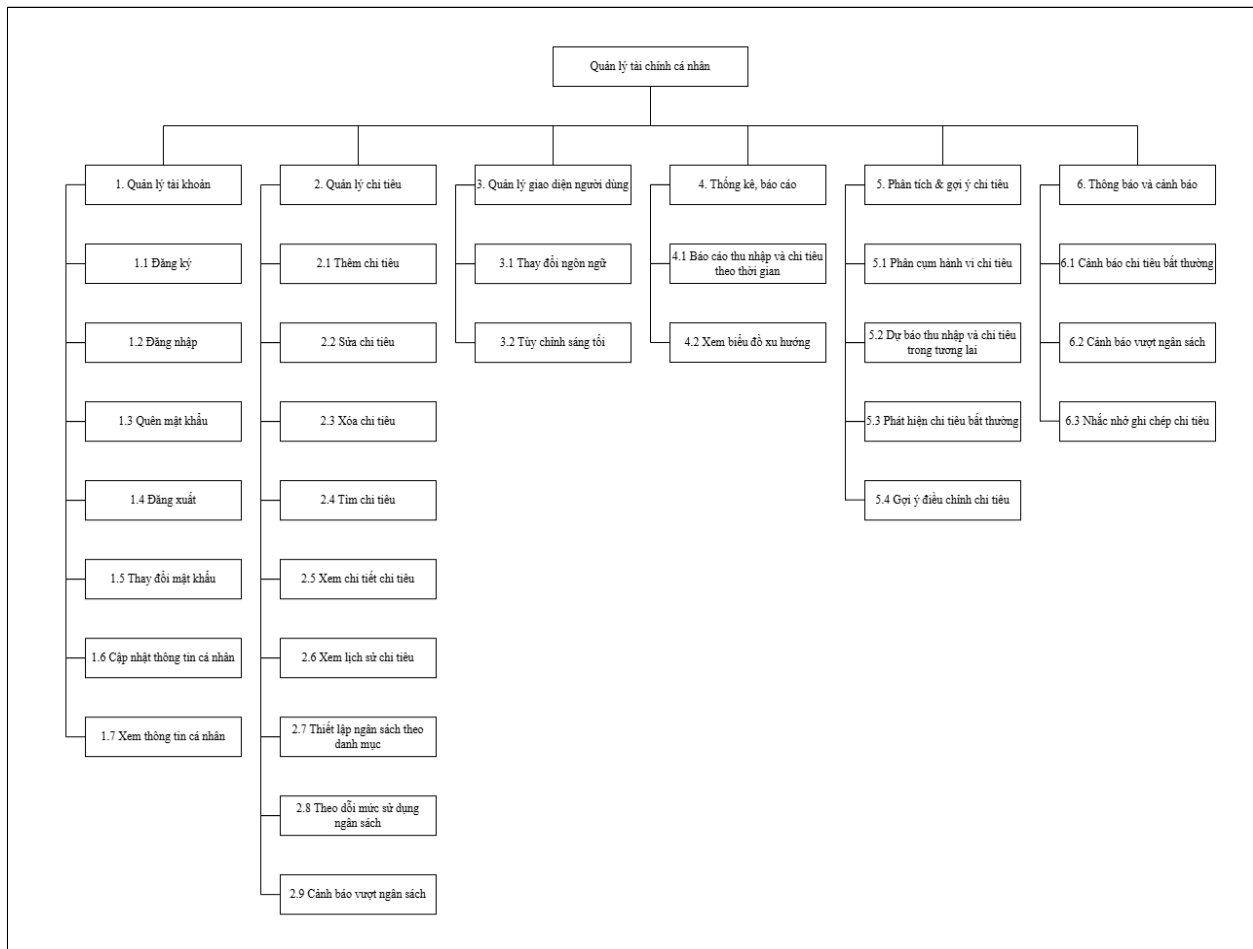
- Nắm vững được công nghệ phát triển của công nghệ.
- Loại bỏ những yêu cầu mập mờ, không rõ ràng.
- Có lượng tài nguyên phát triển phong phú và trình độ chuyên môn, kỹ thuật cao.
- Có thể phù hợp cho dự án nhỏ, ngắn hạn.

Bên cạnh những điều kiện nêu trên, mô hình Waterfall được lựa chọn cho đề tài này vì phạm vi và mục tiêu nghiên cứu đã được xác định rõ ràng ngay từ đầu. Các chức năng cốt lõi của ứng dụng quản lý tài chính cá nhân như quản lý thu nhập và chi tiêu, phân loại giao dịch, thống kê và báo cáo tài chính, cũng như tích hợp các mô hình phân tích dữ liệu thông minh đều đã được định hướng cụ thể trong quá trình khảo sát và phân tích yêu cầu. Trong

suốt quá trình thực hiện đề tài, các yêu cầu này ít có sự thay đổi lớn, do đó mô hình Waterfall với quy trình phát triển tuyến tính giúp thực hiện dễ dàng kiểm soát tiến độ, tổ chức công việc theo từng giai đoạn rõ ràng và đảm bảo mỗi bước đều được hoàn thiện trước khi chuyển sang giai đoạn tiếp theo.

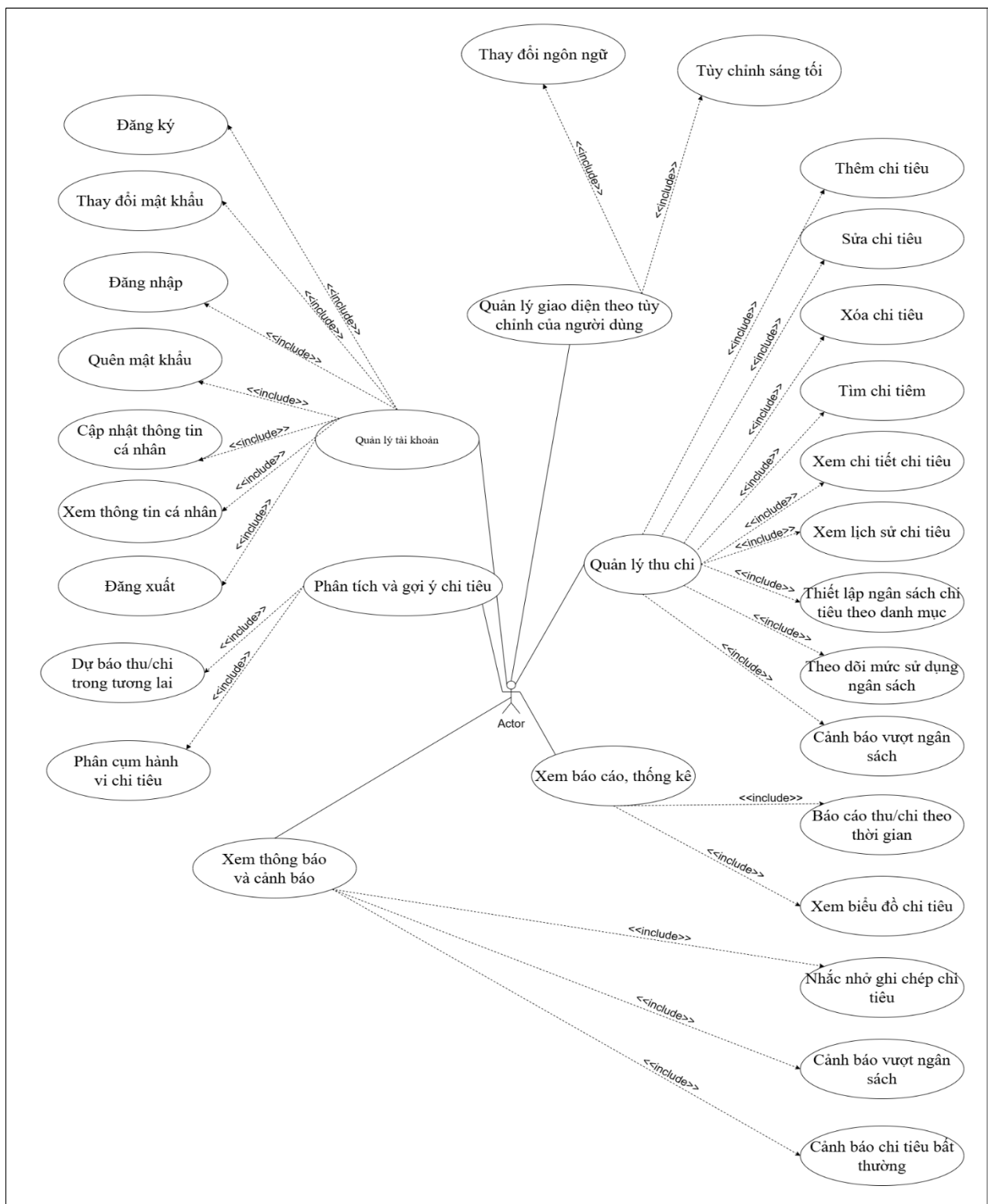
3.2 Thiết kế chức năng của hệ thống

3.2.1 Sơ đồ phân rã chức năng



Hình 3.1 Sơ đồ phân rã chức năng

3.2.2 Sơ đồ Use case



Hình 3.2 Sơ đồ Use case

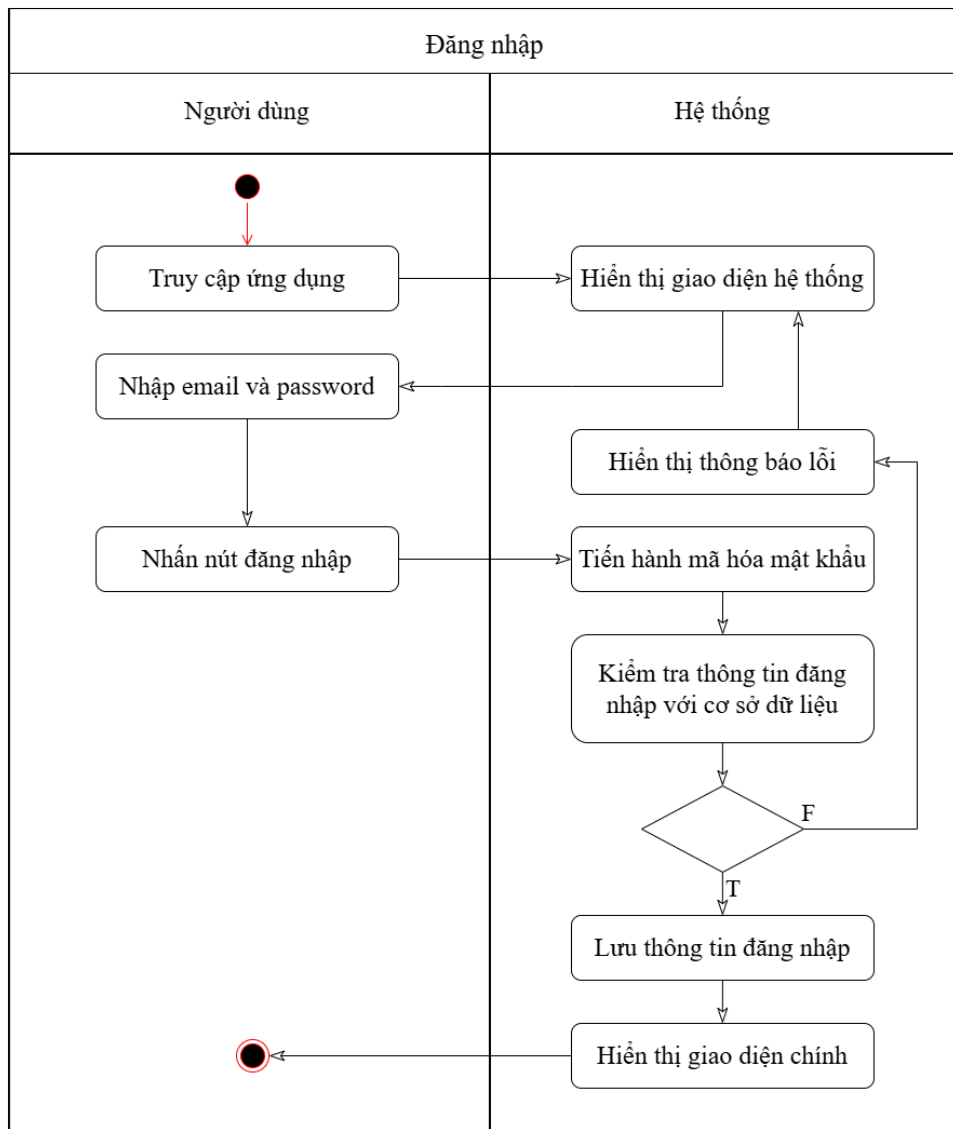
3.2.3 Đặc tả Use case

3.2.3.1 Use case đăng nhập

Tên use case	Đăng nhập
Tên tác nhân	Người dùng
Mô tả	Ca sử dụng cho phép người dùng đã có tài khoản truy cập vào hệ thống thông qua việc nhập thông tin xác thực hợp lệ.
Điều kiện tiên quyết	Người dùng đã có tài khoản trong hệ thống. Người dùng chưa đăng nhập vào hệ thống. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Thông tin đăng nhập đã nhập không được lưu trữ. Hệ thống quay về giao diện đăng nhập ban đầu.
Đảm bảo thành công	Người dùng đăng nhập thành công vào hệ thống. Phiên đăng nhập được khởi tạo. Giao diện chính của ứng dụng được hiển thị.
Kích hoạt	Người dùng truy cập vào hệ thống, chọn chức năng Đăng nhập.
Chuỗi sự kiện chính	<ul style="list-style-type: none">- 1. Người dùng truy cập vào ứng dụng.- 2. Hệ thống hiển thị giao diện đăng nhập.- 3. Người dùng nhập tên đăng nhập (email) và mật khẩu.- 4. Người dùng nhấn nút “Đăng nhập”.- 5. Hệ thống tiến hành mã hóa mật khẩu.- 6. Hệ thống kiểm tra thông tin đăng nhập với cơ sở dữ liệu.- 7. Nếu thông tin hợp lệ, hệ thống tạo phiên đăng nhập cho người dùng.- 8. Hệ thống chuyển người dùng đến giao diện chính.- 9. Ca sử dụng kết thúc.

Ngoại lệ	<p>E1 – Thông tin đăng nhập không hợp lệ</p> <p>5a. Hệ thống phát hiện tên đăng nhập hoặc mật khẩu không chính xác.</p> <p>5b. Hệ thống hiển thị thông báo sai thông tin đăng nhập.</p> <p>5c. Người dùng được yêu cầu nhập lại thông tin.</p> <p>E2 – Lỗi hệ thống</p> <p>6a. Xảy ra lỗi trong quá trình xử lý hoặc kết nối cơ sở dữ liệu.</p> <p>6b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Mật khẩu người dùng được mã hóa trước khi đối chiếu với cơ sở dữ liệu.</p> <p>Hệ thống chỉ cho phép đăng nhập đối với các tài khoản đã được xác minh.</p>

Bảng 3.1 Đặc tả use case đăng nhập



Hình 3.3 Sơ đồ hoạt động đăng nhập

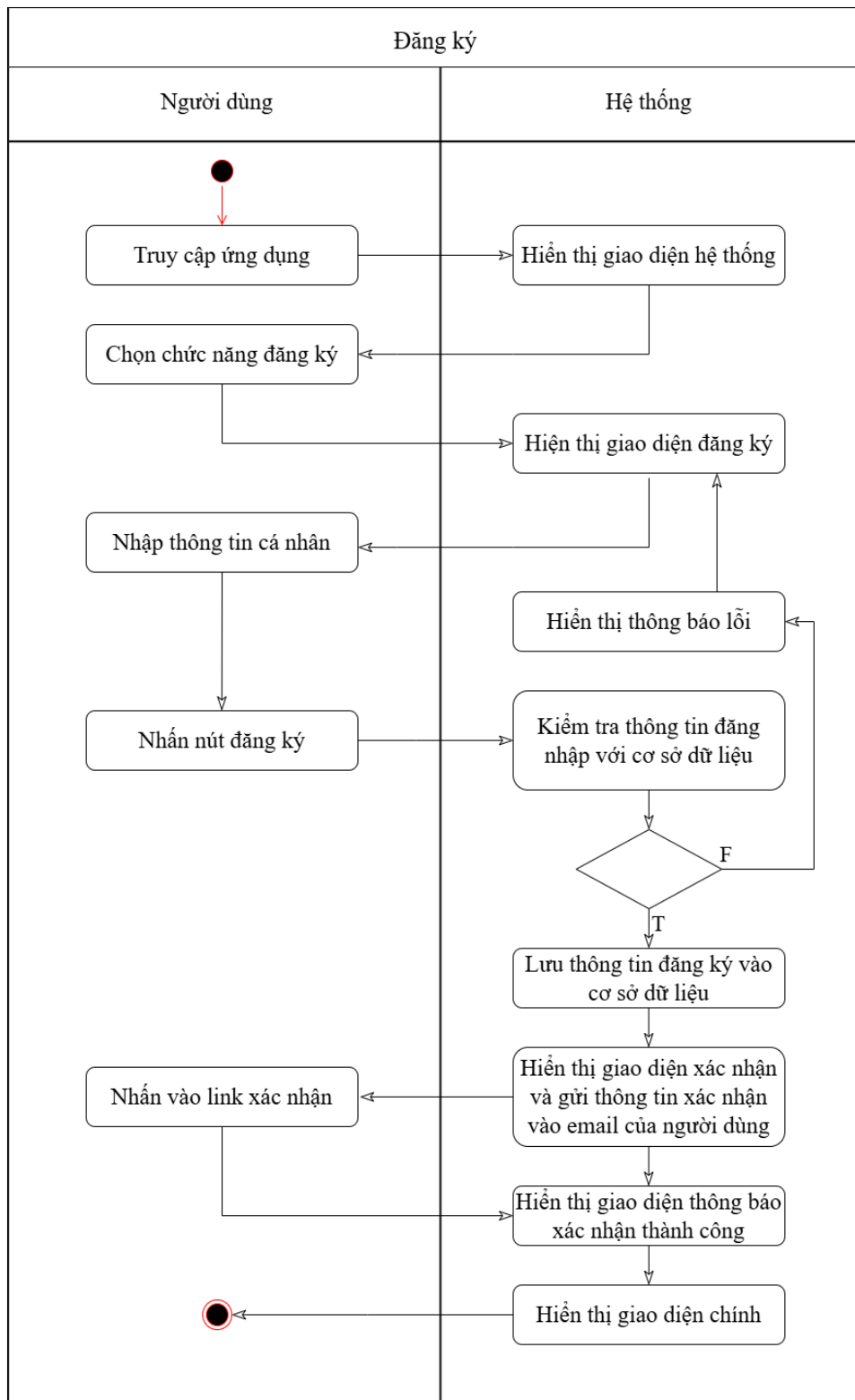
3.2.3.2 Use case đăng kí

Tên use case	Đăng kí tài khoản.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng tạo một tài khoản mới trong hệ thống thông qua việc cung cấp các thông tin cần thiết và thực hiện xác minh email.
Điều kiện tiên quyết	Người dùng chưa có tài khoản trong hệ thống. Người dùng truy cập vào giao diện đăng ký tài khoản.

	Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	<p>Biểu mẫu đăng ký được hiển thị đầy đủ.</p> <p>Thông tin người dùng đã nhập không được lưu trữ khi đăng ký thất bại.</p> <p>Hệ thống không tạo tài khoản khi dữ liệu không hợp lệ.</p>
Đảm bảo thành công	<p>Tài khoản mới được tạo thành công và lưu trữ trong hệ thống.</p> <p>Email xác nhận được gửi đến người dùng.</p> <p>Tài khoản được kích hoạt sau khi xác minh email thành công.</p>
Kích hoạt	Người dùng truy cập vào giao diện hệ thống và chọn chức năng “Đăng ký”.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng truy cập vào ứng dụng. - 2. Người dùng chọn chức năng “Đăng ký” tại giao diện đăng nhập. - 3. Hệ thống hiển thị biểu mẫu đăng ký tài khoản. - 4. Người dùng nhập đầy đủ thông tin đăng ký (tên đăng nhập, email, mật khẩu, ...). - 5. Người dùng nhấn nút “Đăng ký”. - 6. Hệ thống kiểm tra tính hợp lệ của các thông tin đã nhập. - 7. Hệ thống tiến hành mã hóa mật khẩu. - 8. Hệ thống tạo tài khoản mới và lưu trữ vào cơ sở dữ liệu với trạng thái chưa xác minh. - 9. Hệ thống gửi email xác nhận đến email của người dùng. - 10. Người dùng truy cập email, nhấn vào liên kết xác nhận. - 11. Hệ thống xác minh tài khoản thành công. - 12. Hệ thống hiển thị thông báo xác nhận đăng ký thành công trên giao diện ứng dụng. - 13. Ca sử dụng kết thúc.

Ngoại lệ	<p>E1 – Email không hợp lệ</p> <p>6a. Hệ thống phát hiện email không đúng định dạng.</p> <p>6b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E2 – Mật khẩu không đúng định dạng</p> <p>6a. Hệ thống phát hiện mật khẩu không đáp ứng yêu cầu bảo mật (độ dài, ký tự đặc biệt, ...).</p> <p>6b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E3 – Thông tin không đầy đủ</p> <p>6a. Hệ thống phát hiện thiếu thông tin bắt buộc.</p> <p>6b. Hệ thống hiển thị thông báo yêu cầu người dùng điền đầy đủ thông tin.</p> <p>E4 – Email đã tồn tại</p> <p>8a. Hệ thống phát hiện email đã được đăng ký.</p> <p>8b. Hệ thống hiển thị thông báo email đã tồn tại và đề nghị người dùng đăng nhập.</p> <p>E5 – Lỗi hệ thống</p> <p>8a. Xảy ra lỗi trong quá trình tạo tài khoản hoặc gửi email xác nhận.</p> <p>8b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Mật khẩu được mã hóa trước khi lưu trữ vào cơ sở dữ liệu.</p> <p>Người dùng chỉ có thể đăng nhập sau khi hoàn tất xác minh email.</p>

Bảng 3.2 Đặc tả use case đăng ký



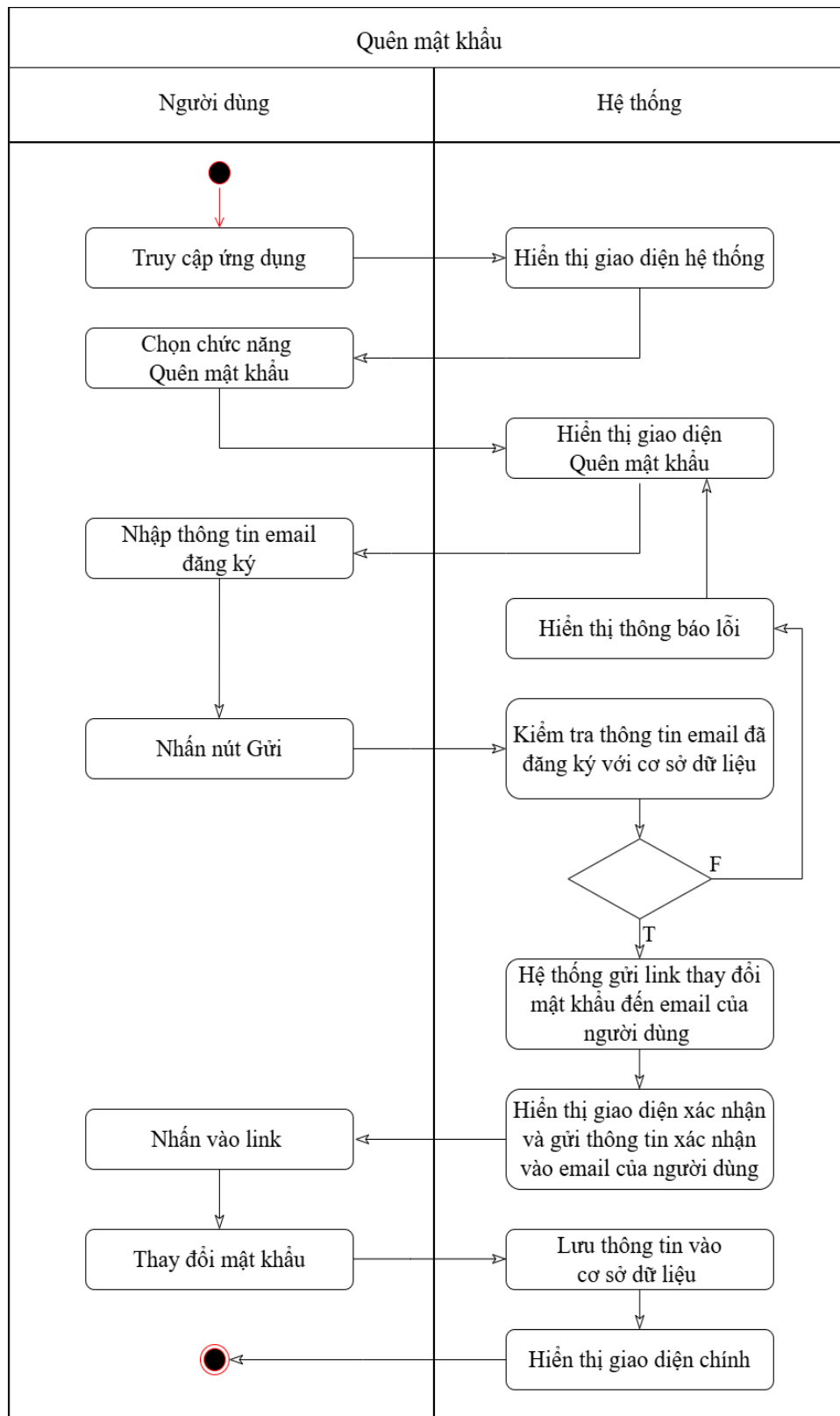
Hình 3.4 Sơ đồ hoạt động đăng ký

3.2.3.3 Use case quên mật khẩu

Tên use case	Quên mật khẩu.
Tên tác nhân	Người dùng.
Người dùng	Ca sử dụng cho phép người dùng đã có tài khoản khôi phục mật khẩu trong trường hợp quên mật khẩu thông qua liên kết được gửi bởi hệ thống.
Điều kiện tiên quyết	Người dùng đã có tài khoản trong hệ thống. Người dùng chưa đăng nhập vào hệ thống. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Giao diện khôi phục mật khẩu được hiển thị đầy đủ. Mật khẩu cũ không bị thay đổi nếu quá trình khôi phục thất bại. Thông tin xác nhận được gửi đúng theo yêu cầu của người dùng.
Đảm bảo thành công	Mật khẩu mới được cập nhật thành công trong hệ thống. Người dùng có thể đăng nhập bằng mật khẩu mới. Phiên khôi phục mật khẩu được kết thúc an toàn.
Kích hoạt	Người dùng chọn chức năng “Quên mật khẩu” tại giao diện đăng nhập.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng truy cập ứng dụng. - 2. Người dùng chọn chức năng “Quên mật khẩu” trên giao diện đăng nhập. - 3. Hệ thống hiển thị biểu mẫu yêu cầu nhập email đăng ký. - 4. Người dùng nhập email và nhấn "Xác nhận". - 5. Hệ thống kiểm tra thông tin người dùng trên cơ sở dữ liệu. - 6. Hệ thống gửi liên kết khôi phục mật khẩu đến người dùng. - 7. Người dùng truy cập liên kết theo hướng dẫn.

	<ul style="list-style-type: none"> - 8. Hệ thống hiển thị giao diện đặt lại mật khẩu mới. - 9. Người dùng nhập mật khẩu mới và xác nhận. - 10. Hệ thống mã hóa mật khẩu mới và cập nhật mật khẩu mới vào cơ sở dữ liệu. - 11. Hệ thống hiển thị thông báo “khôi phục mật khẩu thành công”. - 12. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Email không tồn tại</p> <p>5a. Hệ thống phát hiện thông tin không tồn tại trong hệ thống.</p> <p>5b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng kiểm tra lại.</p> <p>E2 – Liên kết hết hạn</p> <p>7a. Hệ thống phát hiện liên kết không còn hiệu lực.</p> <p>7b. Hệ thống hiển thị thông báo và cho phép người dùng yêu cầu gửi lại.</p> <p>E3 – Lỗi hệ thống</p> <p>6a. Xảy ra lỗi trong quá trình gửi email cập nhật mật khẩu.</p> <p>6b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Mật khẩu mới được mã hóa trước khi lưu trữ.</p> <p>Liên kết chỉ có hiệu lực trong một khoảng thời gian nhất định nhằm đảm bảo an toàn thông tin.</p>

Bảng 3.3 Đặc tả use case quên mật khẩu



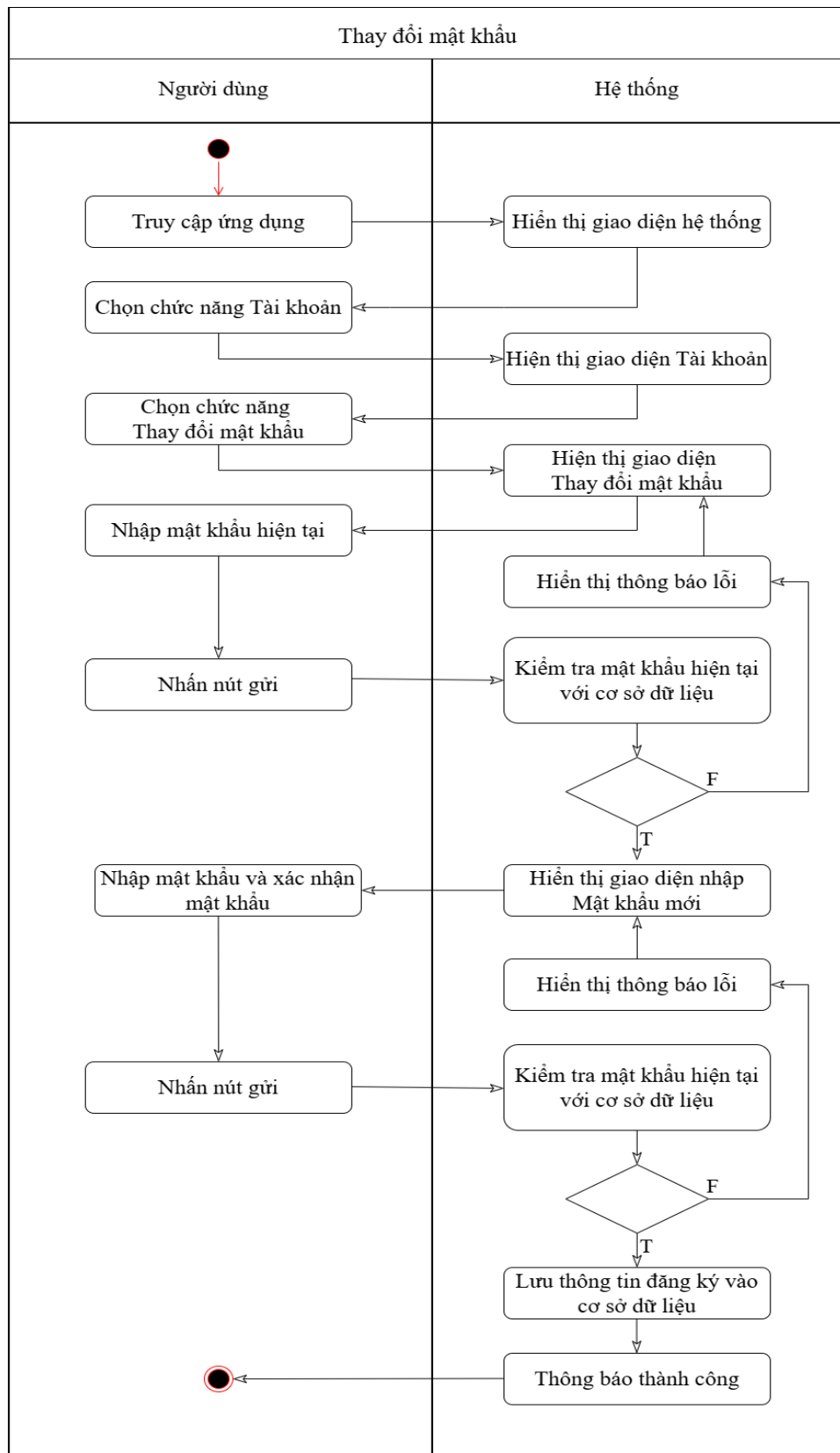
Hình 3.5 Sơ đồ hoạt động quên mật khẩu

3.2.3.4 Use case thay đổi mật khẩu

Tên use case	Thay đổi mật khẩu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập thay đổi mật khẩu hiện tại nhằm tăng cường bảo mật tài khoản.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đang ở giao diện ứng dụng. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Giao diện thay đổi mật khẩu được hiển thị đầy đủ. Mật khẩu cũ không bị thay đổi nếu quá trình cập nhật thất bại. Thông tin mật khẩu mới không được lưu khi xảy ra lỗi.
Đảm bảo thành công	Mật khẩu mới được cập nhật thành công trong hệ thống. Người dùng có thể sử dụng mật khẩu mới để đăng nhập ở những lần tiếp theo. Bảo mật tài khoản được đảm bảo.
Kích hoạt	Người dùng chọn chức năng "Thay đổi mật khẩu" trong giao diện tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Thay đổi mật khẩu”. - 5. Hệ thống hiển thị giao diện yêu cầu nhập mật khẩu hiện tại. - 6. Người dùng nhập mật khẩu hiện tại. - 7. Hệ thống kiểm tra tính chính xác của mật khẩu hiện tại. - 8. Hệ thống hiển thị giao diện nhập mật khẩu mới và xác nhận mật khẩu. - 9. Người dùng nhập mật khẩu mới và xác nhận mật khẩu mới. - 10. Hệ thống kiểm tra mật khẩu mới có đáp ứng các tiêu chí

	<p>bảo mật và mật khẩu xác nhận có trùng khớp hay không.</p> <ul style="list-style-type: none"> - 11. Nếu thông tin hợp lệ, hệ thống mã hóa mật khẩu mới và cập nhật vào cơ sở dữ liệu. - 12. Hệ thống hiển thị "Thay đổi mật khẩu thành công". - 13. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Mật khẩu hiện tại không đúng</p> <p>7a. Hệ thống phát hiện mật khẩu hiện tại không chính xác.</p> <p>7b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại mật khẩu.</p> <p>E2 – Mật khẩu mới và mật khẩu xác nhận không khớp</p> <p>10a. Hệ thống phát hiện mật khẩu mới và mật khẩu xác nhận không trùng khớp.</p> <p>10b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E3 – Mật khẩu mới không đáp ứng tiêu chí bảo mật</p> <p>10a. Hệ thống phát hiện mật khẩu mới không đạt yêu cầu bảo mật (độ dài, ký tự đặc biệt, ...).</p> <p>10b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập mật khẩu khác.</p> <p>E4 – Lỗi hệ thống / Cơ sở dữ liệu</p> <p>11a. Xảy ra lỗi trong quá trình cập nhật mật khẩu.</p> <p>11b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Mật khẩu được mã hóa trước khi lưu trữ trong cơ sở dữ liệu.</p> <p>Hệ thống có thể yêu cầu người dùng đăng nhập lại sau khi thay đổi mật khẩu để đảm bảo an toàn.</p>

Bảng 3.4 Đặc tả use case thay đổi mật khẩu



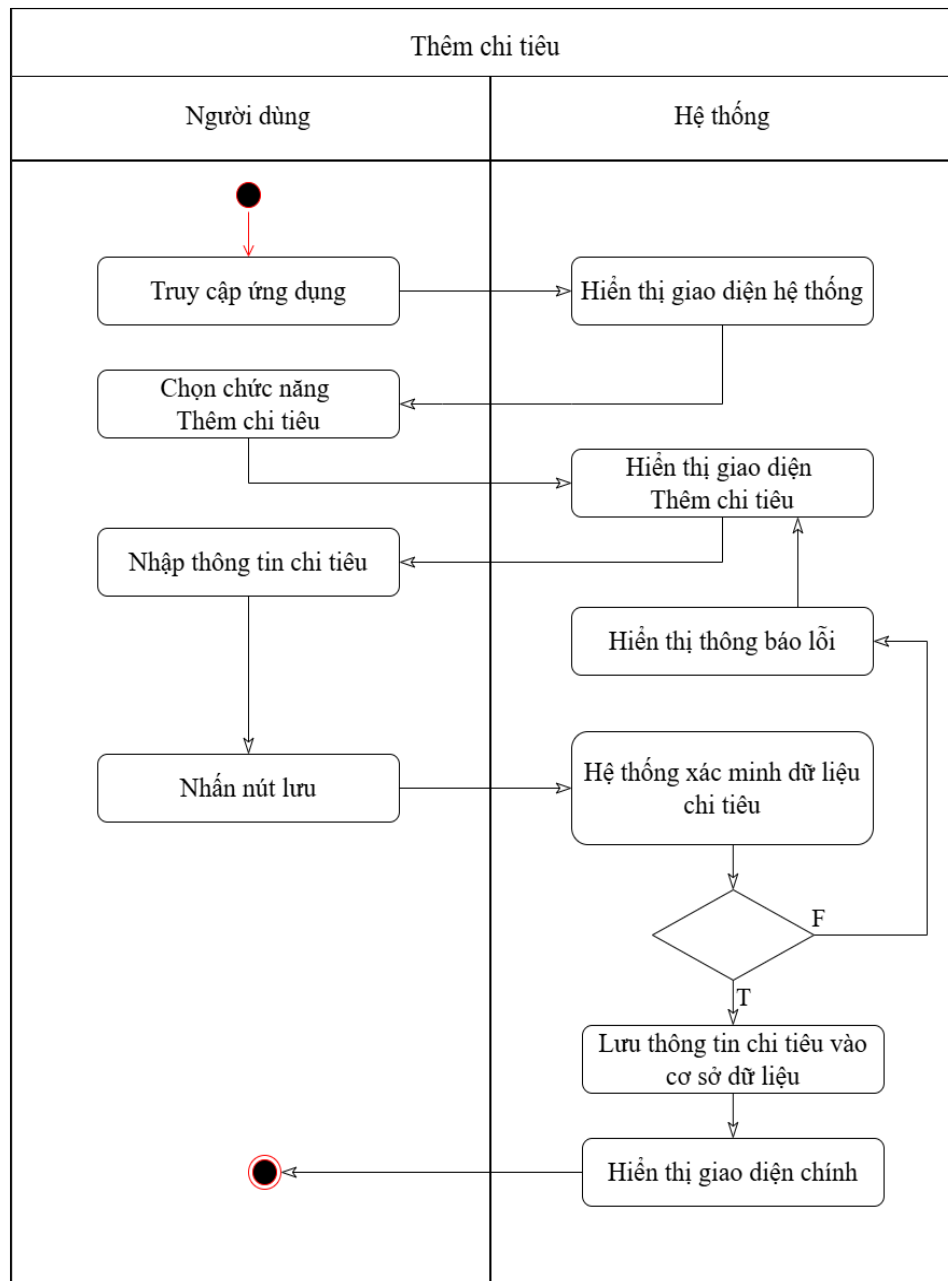
Hình 3.6 Sơ đồ hoạt động thay đổi mật khẩu

3.2.3.5 Use case thêm chi tiêu

Tên use case	Thêm chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập thêm một khoản chi tiêu mới vào hệ thống để phục vụ việc quản lý và theo dõi tài chính cá nhân.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đang ở giao diện ứng dụng. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Giao diện thêm chi tiêu được hiển thị đầy đủ. Không có dữ liệu chi tiêu nào được lưu khi thông tin không hợp lệ hoặc xảy ra lỗi.
Đảm bảo thành công	Thông tin chi tiêu được lưu thành công trong hệ thống. Danh sách chi tiêu của người dùng được cập nhật chính xác.
Kích hoạt	Người dùng chọn chức năng “Thêm chi tiêu” trên giao diện ứng dụng.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn chức năng "Thêm chi tiêu" trên giao diện ứng dụng. - 3. Hệ thống hiển thị giao diện biểu mẫu thêm chi tiêu mới. - 4. Người dùng nhập các thông tin chi tiêu cần thiết (số tiền, loại chi tiêu, ngày, giờ, ghi chú nếu có). - 5. Người dùng nhấn nút “Lưu”. - 6. Hệ thống kiểm tra tính hợp lệ của các thông tin đã nhập. - 7. Hệ thống lưu thông tin chi tiêu vào cơ sở dữ liệu. - 8. Hệ thống cập nhật danh sách chi tiêu của người dùng. - 9. Hệ thống hiển thị thông báo “Thêm chi tiêu thành công”. - 10. Ca sử dụng kết thúc.

Ngoại lệ	<p>E1 – Thông tin không đầy đủ</p> <p>6a. Hệ thống phát hiện thiếu thông tin bắt buộc.</p> <p>6b. Hệ thống hiển thị thông báo yêu cầu người dùng nhập đầy đủ thông tin.</p> <p>E2 – Số tiền không hợp lệ</p> <p>6a. Hệ thống phát hiện số tiền nhập vào không hợp lệ (âm, bằng 0 hoặc không đúng định dạng).</p> <p>6b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E3 – Lỗi hệ thống</p> <p>7a. Xảy ra lỗi trong quá trình lưu dữ liệu.</p> <p>7b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Các khoản chi tiêu được gán với tài khoản người dùng đang đăng nhập.</p> <p>Hệ thống có thể sử dụng dữ liệu chi tiêu để phục vụ thống kê và phân tích tài chính.</p>

Bảng 3.5 Đặc tả use case thêm chi tiêu



Hình 3.7 Sơ đồ hoạt động thêm chi tiêu

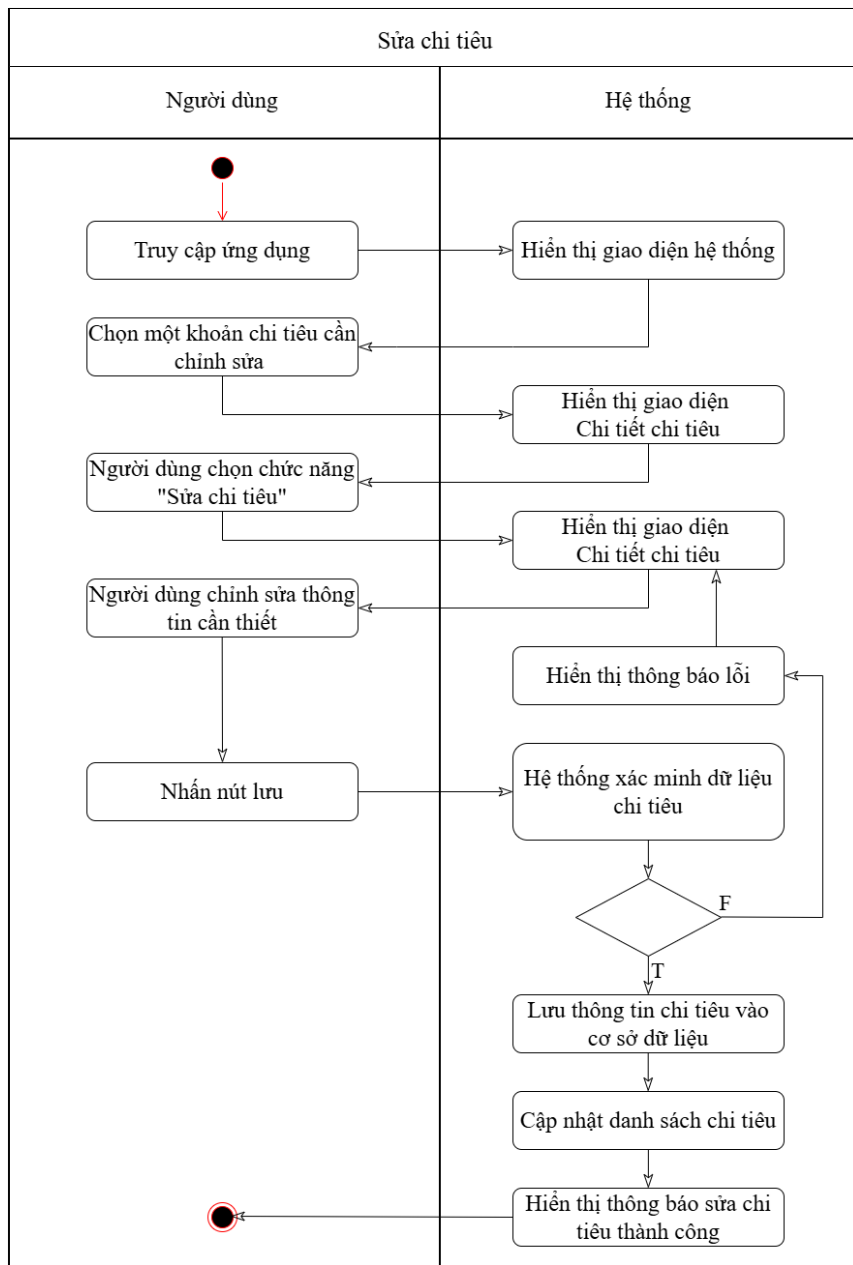
3.2.3.6 Use case sửa chi tiêu

Tên use case	Sửa chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập chỉnh sửa thông tin của một khoản chi tiêu đã được lưu trong hệ thống nhằm cập nhật dữ liệu tài chính chính xác hơn.

Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đã có ít nhất một khoản chi tiêu được lưu trong hệ thống.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Danh sách các khoản chi tiêu được hiển thị đầy đủ.</p> <p>Không có dữ liệu chi tiêu nào bị thay đổi khi xảy ra lỗi hoặc thông tin không hợp lệ.</p>
Đảm bảo thành công	<p>Thông tin chi tiêu được cập nhật thành công trong hệ thống.</p> <p>Danh sách chi tiêu được hiển thị đúng với dữ liệu mới.</p>
Kích hoạt	<p>Người dùng chọn chức năng "Sửa" cho một khoản chi tiêu cụ thể trong danh sách.</p>
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Hệ thống hiển thị danh sách các khoản chi tiêu hiện có. - 3. Người dùng chọn một khoản chi tiêu cần chỉnh sửa. - 4. Hệ thống hiển thị giao diện Chi tiết chi tiêu. - 5. Người dùng chọn chức năng “Sửa chi tiêu”. - 6. Hệ thống hiển thị biểu mẫu sửa chi tiêu với các trường thông tin (số tiền, loại chi tiêu, ngày, giờ, ghi chú nếu có). - 7. Người dùng chỉnh sửa thông tin cần thiết, nhấn nút “Lưu”. - 8. Hệ thống kiểm tra tính hợp lệ của các thông tin đã nhập. - 9. Hệ thống cập nhật thông tin chi tiêu vào cơ sở dữ liệu. - 10. Hệ thống cập nhật danh sách chi tiêu. - 11. Hệ thống hiển thị thông báo “Sửa chi tiêu thành công”. - 12. Ca sử dụng kết thúc.

Ngoại lệ	<p>E1 – Thông tin không đầy đủ</p> <p>8a. Hệ thống phát hiện thiếu thông tin bắt buộc.</p> <p>8b. Hệ thống hiển thị thông báo yêu cầu người dùng nhập đầy đủ thông tin.</p> <p>E2 – Số tiền không hợp lệ</p> <p>8a. Hệ thống phát hiện số tiền không hợp lệ (âm, bằng 0 hoặc sai định dạng).</p> <p>8b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E3 – Lỗi hệ thống</p> <p>9a. Xảy ra lỗi trong quá trình cập nhật dữ liệu.</p> <p>9b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Chỉ cho phép chỉnh sửa các khoản chi tiêu thuộc về tài khoản người dùng đang đăng nhập.</p> <p>Dữ liệu sau khi chỉnh sửa được sử dụng cho các chức năng thống kê và phân tích tài chính.</p>

Bảng 3.6 Đặc tả use case sửa chi tiêu



Hình 3.8 Sơ đồ hoạt động sửa chi tiêu

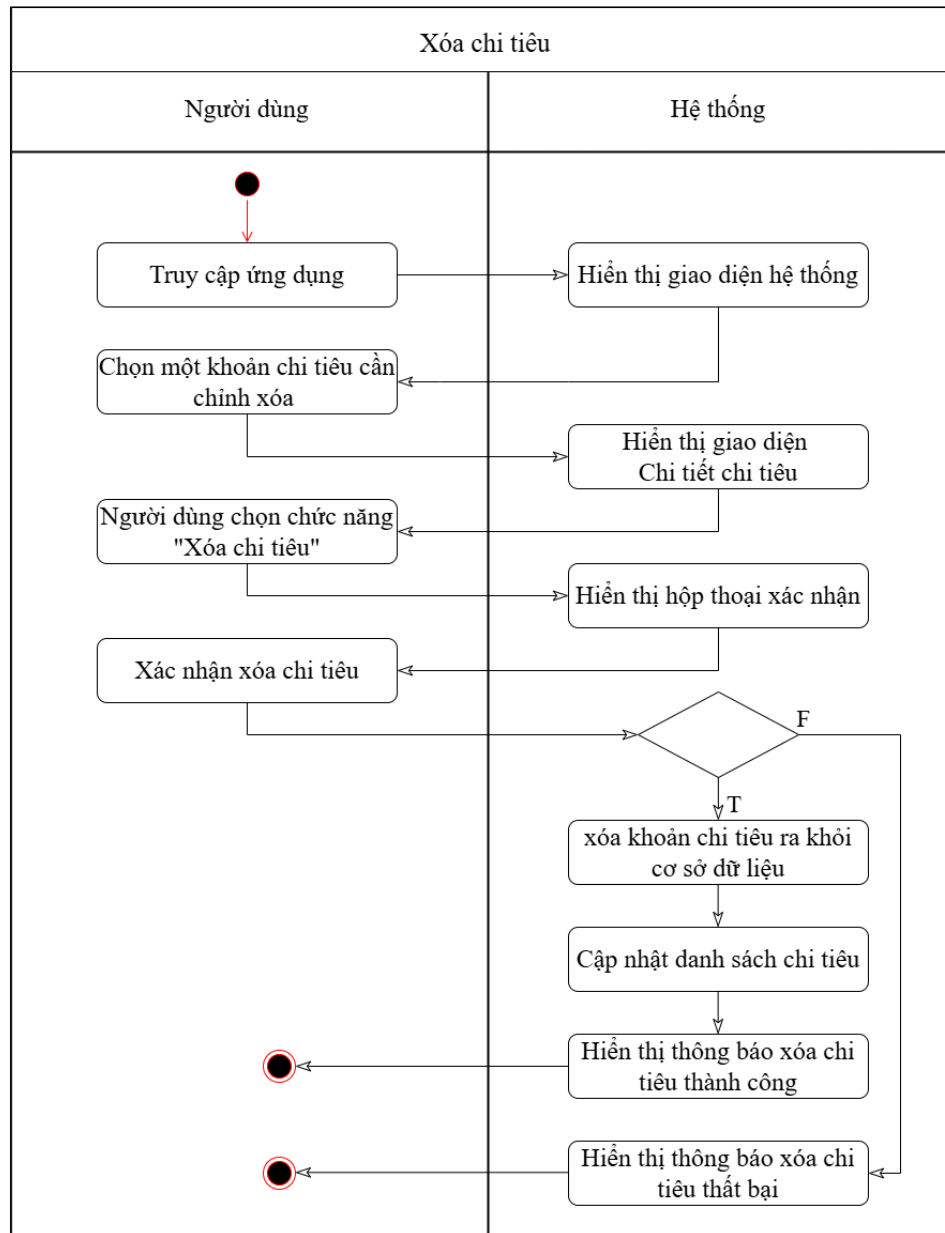
3.2.3.7 Use case xóa chi tiêu

Tên use case	Xóa chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xóa một khoản chi tiêu không còn cần thiết khỏi hệ thống.

Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đã có ít nhất một khoản chi tiêu được lưu trong hệ thống.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Danh sách các khoản chi tiêu được hiển thị đầy đủ.</p> <p>Không có dữ liệu nào bị xóa khi xảy ra lỗi hệ thống.</p>
Đảm bảo thành công	<p>Khoản chi tiêu được xóa hoàn toàn khỏi hệ thống.</p> <p>Danh sách chi tiêu của người dùng được cập nhật chính xác.</p>
Kích hoạt	<p>Người dùng chọn chức năng “Xóa” đối với một khoản chi tiêu cụ thể trong danh sách chi tiêu hoặc tại giao diện chi tiết chi tiêu.</p>
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Hệ thống hiển thị danh sách các khoản chi tiêu hiện có. - 3. Người dùng chọn một khoản chi tiêu cần xóa. - 4. Hệ thống hiển thị giao diện Chi tiết chi tiêu. - 5. Người dùng chọn chức năng “Xóa chi tiêu”. - 6. Hệ thống hiển thị hộp thoại xác nhận xóa chi tiêu. - 7. Người dùng xác nhận thao tác xóa. - 8. Hệ thống xóa khoản chi tiêu khỏi cơ sở dữ liệu. - 9. Hệ thống cập nhật danh sách chi tiêu. - 10. Hệ thống hiển thị thông báo “Xóa chi tiêu thành công”. - 11. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không có chi tiêu</p> <p>2a. Hệ thống không tìm thấy khoản chi tiêu nào trong danh sách.</p> <p>2b. Hệ thống hiển thị thông báo “Không có chi tiêu để xóa”.</p> <p>E2 – Lỗi hệ thống</p> <p>8a. Xảy ra lỗi trong quá trình xóa dữ liệu.</p>

	8b. Hệ thống hiển thị thông báo lỗi hệ thống.
Ghi chú	<p>Hệ thống yêu cầu người dùng xác nhận trước khi xóa để tránh thao tác nhầm.</p> <p>Chỉ cho phép xóa các khoản chi tiêu thuộc tài khoản người dùng đang đăng nhập.</p>

Bảng 3.7 Đặc tả use case xóa chi tiêu

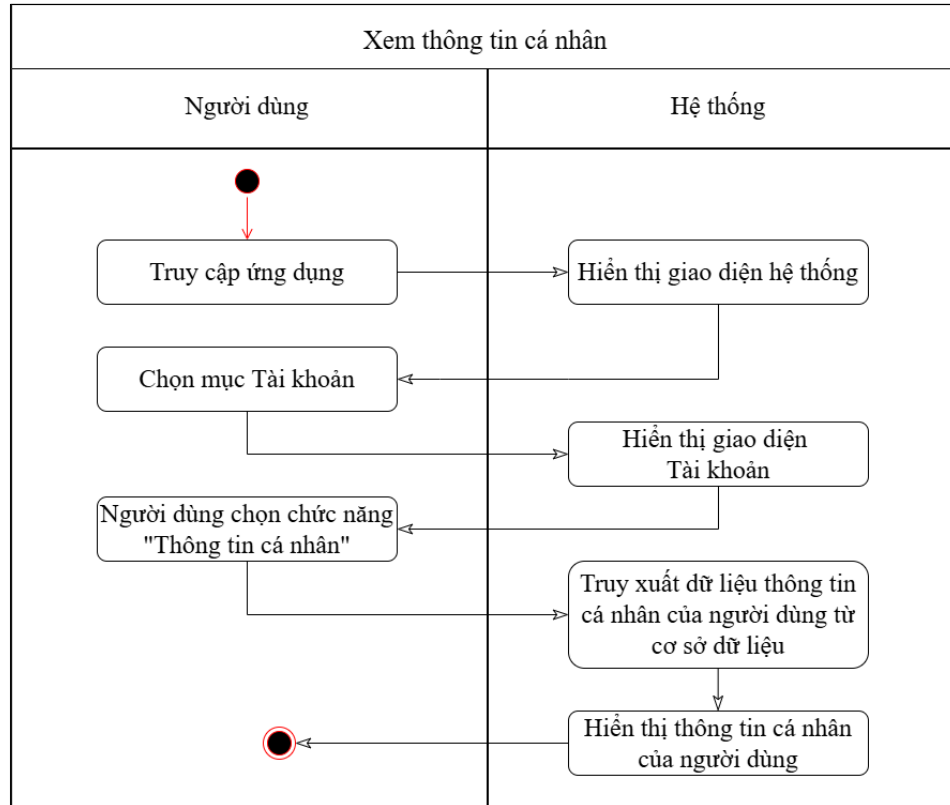


Hình 3.9 Sơ đồ hoạt động xóa chi tiêu

3.2.3.8 Use case xem thông tin cá nhân

Tên use case	Xem thông tin cá nhân
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xem toàn bộ thông tin cá nhân được lưu trữ trong hệ thống.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đang ở giao diện ứng dụng. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Thông tin cá nhân của người dùng được hiển thị đầy đủ. Không có dữ liệu nào bị thay đổi trong quá trình xem thông tin.
Đảm bảo thành công	Người dùng xem được toàn bộ thông tin cá nhân được lưu trữ trong hệ thống. Dữ liệu hiển thị chính xác và nhất quán với cơ sở dữ liệu.
Kích hoạt	Người dùng chọn mục “Thông tin cá nhân” trong giao diện Tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Thông tin cá nhân”. - 5. Hệ thống truy xuất dữ liệu thông tin cá nhân của người dùng từ cơ sở dữ liệu. - 6. Hệ thống hiển thị thông tin cá nhân của người dùng. - 7. Ca sử dụng kết thúc.
Ngoại lệ	Lỗi hệ thống <ul style="list-style-type: none"> 5a. Xảy ra lỗi trong quá trình truy xuất dữ liệu. 5b. Hệ thống hiển thị thông báo lỗi hệ thống.
Ghi chú	Người dùng chỉ có quyền xem thông tin cá nhân của mình. Thông tin hiển thị ở chế độ chỉ đọc (read-only).

Bảng 3.8 Đặc tả use case xem thông tin cá nhân



Hình 3.10 Sơ đồ hoạt động xem thông tin cá nhân

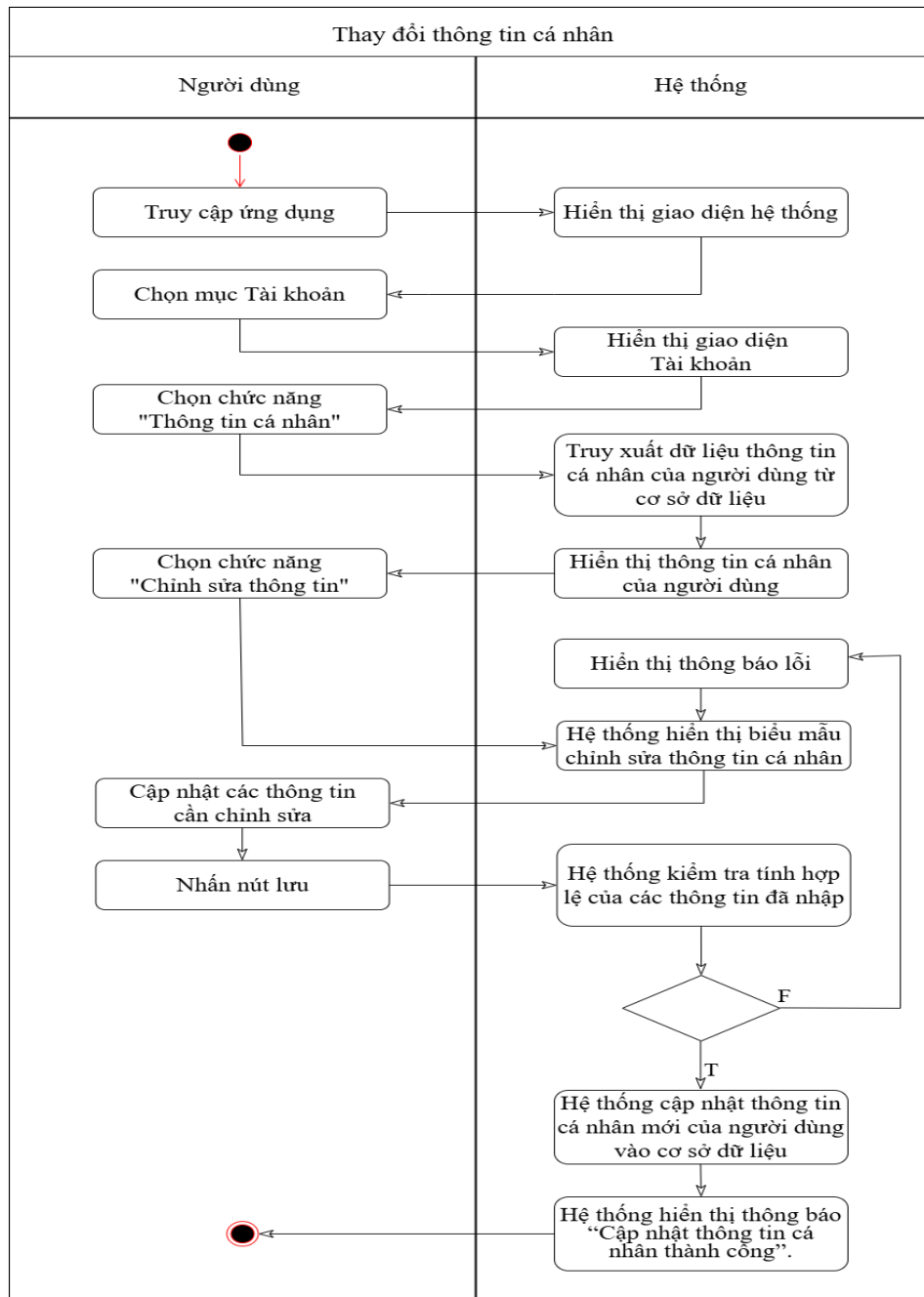
3.2.3.9 Use case thay đổi thông tin cá nhân

Tên use case	Thay đổi thông tin cá nhân
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xem và cập nhật thông tin cá nhân được lưu trữ trong hệ thống.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng truy cập vào giao diện Tài khoản. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Thông tin cá nhân hiện tại của người dùng được hiển thị một cách đầy đủ. Không có dữ liệu cá nhân nào bị thay đổi nếu xảy ra lỗi.
Đảm bảo thành công	Thông tin cá nhân của người dùng được cập nhật thành công.

	Dữ liệu mới được lưu trữ an toàn trong hệ thống.
Kích hoạt	Người dùng chọn mục “Thông tin cá nhân” trong giao diện Tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Thông tin cá nhân”. - 5. Hệ thống truy xuất dữ liệu thông tin cá nhân của người dùng từ cơ sở dữ liệu. - 6. Hệ thống hiển thị đầy đủ thông tin của người dùng. - 7. Người dùng chọn chức năng “Chỉnh sửa thông tin”. - 8. Hệ thống hiển thị biểu mẫu chỉnh sửa thông tin cá nhân. - 9. Người dùng cập nhật các thông tin cần thiết (họ tên, số điện thoại, ngày sinh, ...). - 10. Người dùng nhấn nút “Lưu”. - 11. Hệ thống kiểm tra tính hợp lệ của thông tin đã nhập. - 12. Hệ thống cập nhật thông tin cá nhân mới của người dùng vào cơ sở dữ liệu. - 13. Hệ thống hiển thị thông báo “Cập nhật thông tin cá nhân thành công”. - 14. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Thông tin không hợp lệ</p> <p>11a. Hệ thống phát hiện thông tin không hợp lệ hoặc thiếu dữ liệu bắt buộc.</p> <p>11b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E2 – Lỗi hệ thống</p> <p>12a. Xảy ra lỗi trong quá trình cập nhật dữ liệu.</p>

	12b. Hệ thống hiển thị thông báo lỗi hệ thống.
Ghi chú	<p>Người dùng chỉ có thể chỉnh sửa thông tin cá nhân.</p> <p>Một số thông tin quan trọng (email đăng nhập) có thể không cho phép chỉnh sửa trực tiếp.</p>

Bảng 3.9 Đặc tả use case thay đổi thông tin cá nhân



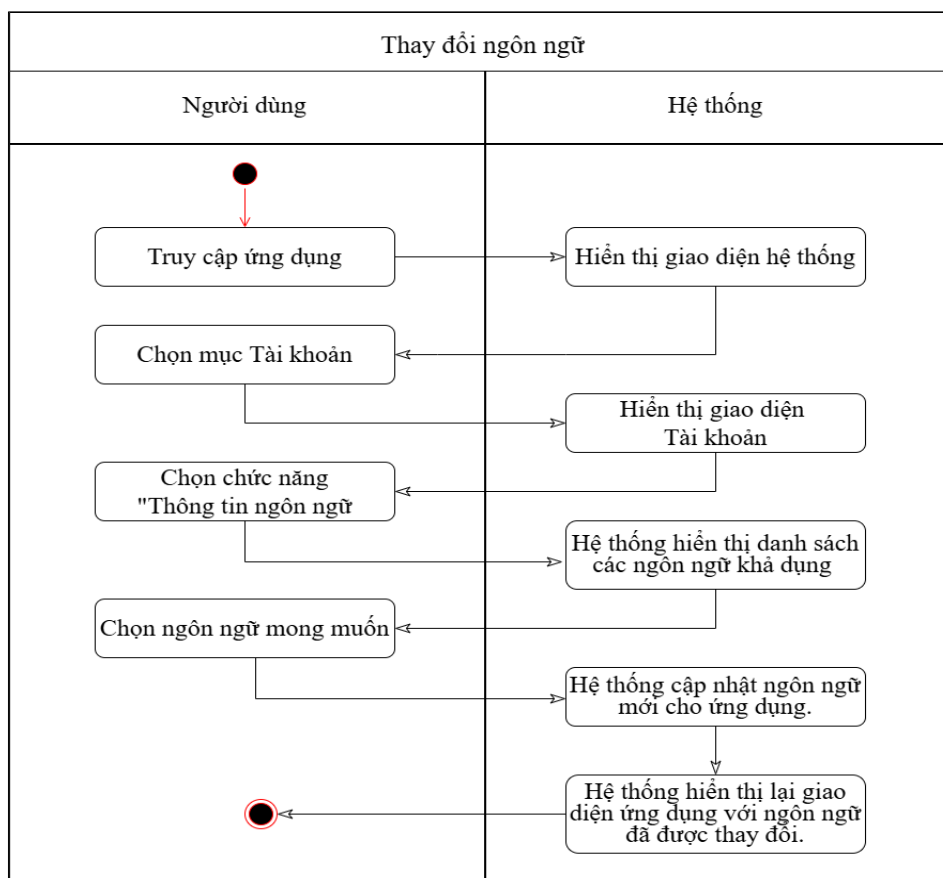
Hình 3.11 Sơ đồ hoạt động thay đổi thông tin cá nhân

3.2.3.10 Use case thay đổi ngôn ngữ

Tên use case	Thay đổi ngôn ngữ
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập thay đổi ngôn ngữ hiển thị của ứng dụng theo nhu cầu cá nhân.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng truy cập vào giao diện Tài khoản. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Giao diện lựa chọn ngôn ngữ được hiển thị đầy đủ. Ngôn ngữ hiện tại của ứng dụng không bị thay đổi nếu bị xảy ra lỗi.
Đảm bảo thành công	Ngôn ngữ ứng dụng được thay đổi thành công theo lựa chọn của người dùng. Thiết lập ngôn ngữ mới được lưu trữ an toàn trong hệ thống.
Kích hoạt	Người dùng chọn mục “Thay đổi ngôn ngữ” trong giao diện Tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Thay đổi ngôn ngữ”. - 5. Hệ thống hiển thị danh sách các ngôn ngữ khả dụng (Tiếng Việt, Tiếng Anh) - 6. Người dùng chọn ngôn ngữ mong muốn. - 7. Người dùng xác nhận lựa chọn. - 8. Hệ thống cập nhật ngôn ngữ mới cho ứng dụng. - 9. Hệ thống hiển thị lại giao diện ứng dụng với ngôn ngữ đã được thay đổi. - 10. Ca sử dụng kết thúc.

Ngoại lệ	<p>E1 – Ngôn ngữ không khả dụng</p> <p>6a. Ngôn ngữ được chọn không tồn tại hoặc không được hệ thống hỗ trợ.</p> <p>6b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng chọn lại ngôn ngữ khác.</p> <p>E2 – Lỗi hệ thống</p> <p>8a. Xảy ra lỗi trong quá trình cập nhật ngôn ngữ.</p> <p>8b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Ngôn ngữ được thay đổi sẽ được áp dụng cho toàn bộ giao diện ứng dụng.</p> <p>Thiết lập ngôn ngữ được lưu lại và tự động áp dụng cho những lần đăng nhập tiếp theo.</p>

Bảng 3.10 Đặc tả use case thay đổi ngôn ngữ



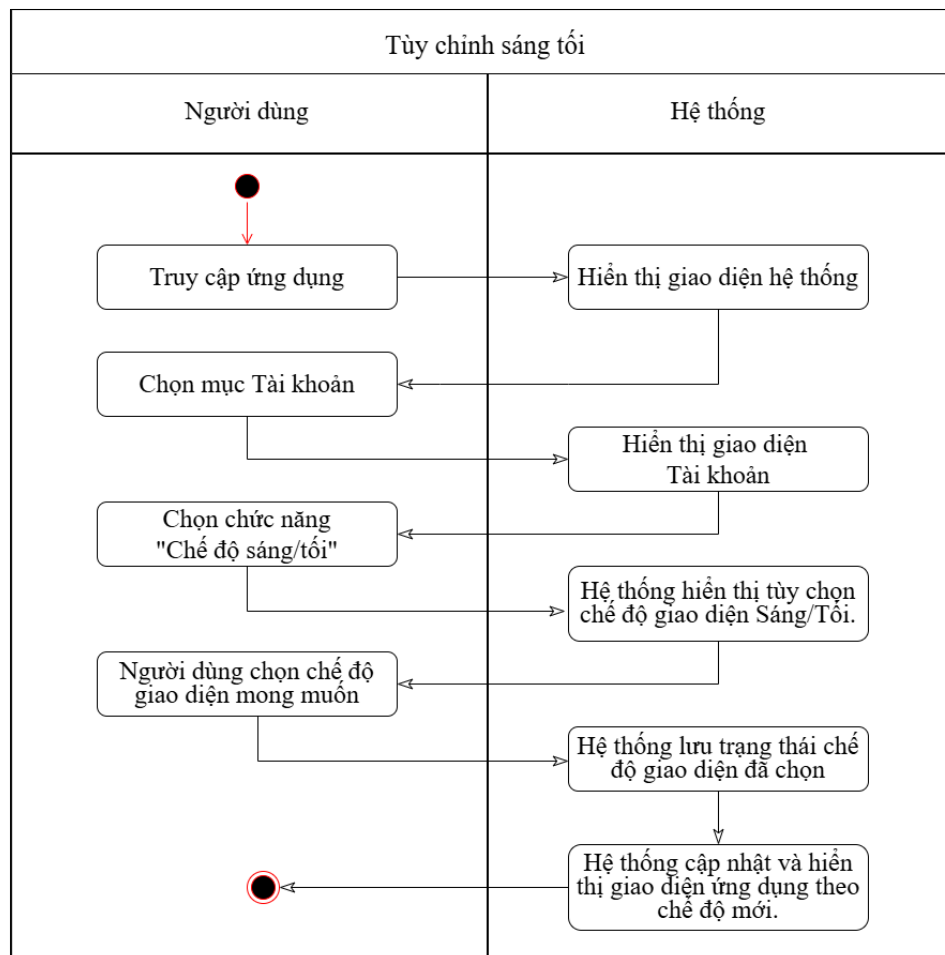
Hình 3.12 Sơ đồ hoạt động thay đổi ngôn ngữ

3.2.3.11 Use case tùy chỉnh sáng tối

Tên use case	Tùy chỉnh sáng tối.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập thay đổi giao diện ứng dụng giữa chế độ sáng và chế độ tối nhằm tăng trải nghiệm sử dụng và phù hợp với nhu cầu cá nhân.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng truy cập vào giao diện Tài khoản. Thiết bị hoạt động bình thường.
Đảm bảo tối thiểu	Giao diện tùy chọn chế độ sáng/tối được hiển thị đầy đủ. Chế độ giao diện hiện tại không bị thay đổi nếu xảy ra lỗi.
Đảm bảo thành công	Giao diện ứng dụng được chuyển đổi thành công sang chế độ sáng hoặc tối theo lựa chọn của người dùng. Thiết lập giao diện được lưu trữ và áp dụng cho các lần sử dụng tiếp theo.
Kích hoạt	Người dùng chọn mục “Chế độ sáng/tối” trong giao diện Tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Chế độ sáng/tối”. - 5. Hệ thống hiển thị tùy chọn chế độ giao diện (Sáng / Tối). - 6. Người dùng chọn chế độ giao diện mong muốn. - 7. Hệ thống lưu trạng thái chế độ giao diện đã chọn. - 8. Hệ thống cập nhật và hiển thị giao diện ứng dụng theo chế độ mới. - 9. Ca sử dụng kết thúc.
Ngoại lệ	E1 – Lỗi hiển thị giao diện

	<p>8a. Giao diện không thể cập nhật theo chế độ đã chọn.</p> <p>8b. Hệ thống hiển thị thông báo lỗi và giữ nguyên chế độ giao diện hiện tại.</p> <p>E2 – Lỗi hệ thống</p> <p>7a. Xảy ra lỗi trong quá trình lưu trạng thái chế độ giao diện.</p> <p>7b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Chế độ sáng/tối được áp dụng cho toàn bộ giao diện ứng dụng.</p> <p>Thiết lập giao diện sẽ được tự động áp dụng cho những lần đăng nhập tiếp theo.</p>

Bảng 3.11 Đặc tả use case tùy chỉnh sáng tối



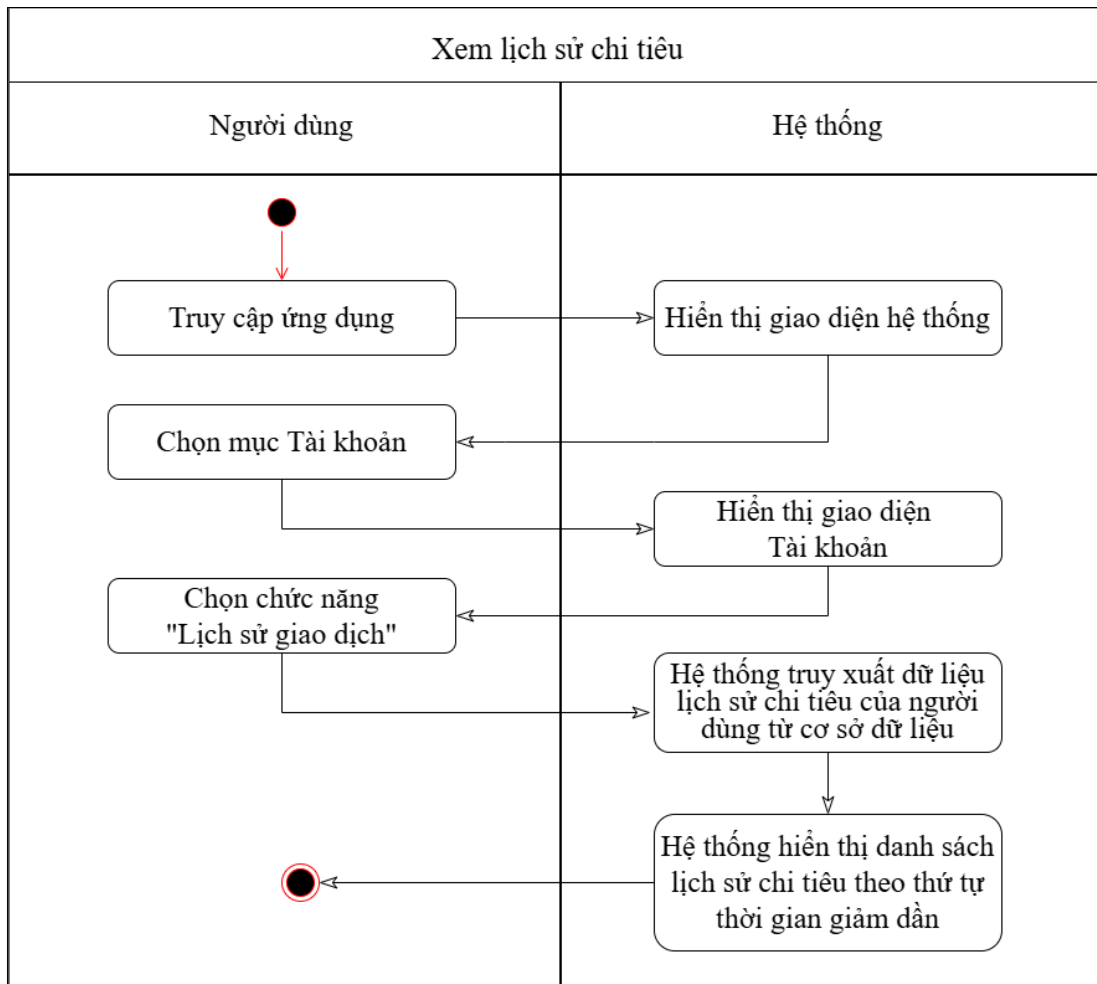
Hình 3.13 Sơ đồ hoạt động tùy chỉnh sáng tối

3.2.3.12 Use case xem lịch sử chi tiêu

Tên use case	Xem lịch sử chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xem danh sách lịch sử chi tiêu được lưu trữ trong hệ thống, giúp theo dõi và quản lý các khoản chi tiêu theo thời gian.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có dữ liệu chi tiêu được lưu trong hệ thống. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Danh sách lịch sử chi tiêu được hiển thị theo thứ tự thời gian. Không có dữ liệu chi tiêu nào bị thay đổi nếu xảy ra lỗi.
Đảm bảo thành công	Lịch sử chi tiêu được hiển thị đầy đủ và chính xác. Người dùng có thể xem chi tiết từng khoản chi tiêu trong khoảng thời gian được chọn.
Kích hoạt	Người dùng chọn chức năng “Lịch sử chi tiêu” trong giao diện Tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Lịch sử chi tiêu”. - 5. Hệ thống truy xuất dữ liệu lịch sử chi tiêu của người dùng từ cơ sở dữ liệu. - 6. Hệ thống hiển thị danh sách lịch sử chi tiêu theo thứ tự thời gian giảm dần (từ gần nhất đến xa nhất), bao gồm thời gian, danh mục, số tiền, ghi chú, ... - 7. Người dùng có thể chọn từng khoản chi tiêu để xem chi tiết hoặc thực hiện chỉnh sửa/xóa. - 8. Ca sử dụng kết thúc.

Ngoại lệ	<p>E1 – Lỗi hiển thị giao diện</p> <p>6a. Danh sách lịch sử chi tiêu không hiển thị đúng định dạng.</p> <p>6b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thử lại.</p> <p>E2 – Lỗi hệ thống</p> <p>5a. Xảy ra lỗi trong quá trình truy xuất dữ liệu.</p> <p>5b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
----------	--

Bảng 3.12 Đặc tả use case xem lịch sử chi tiêu



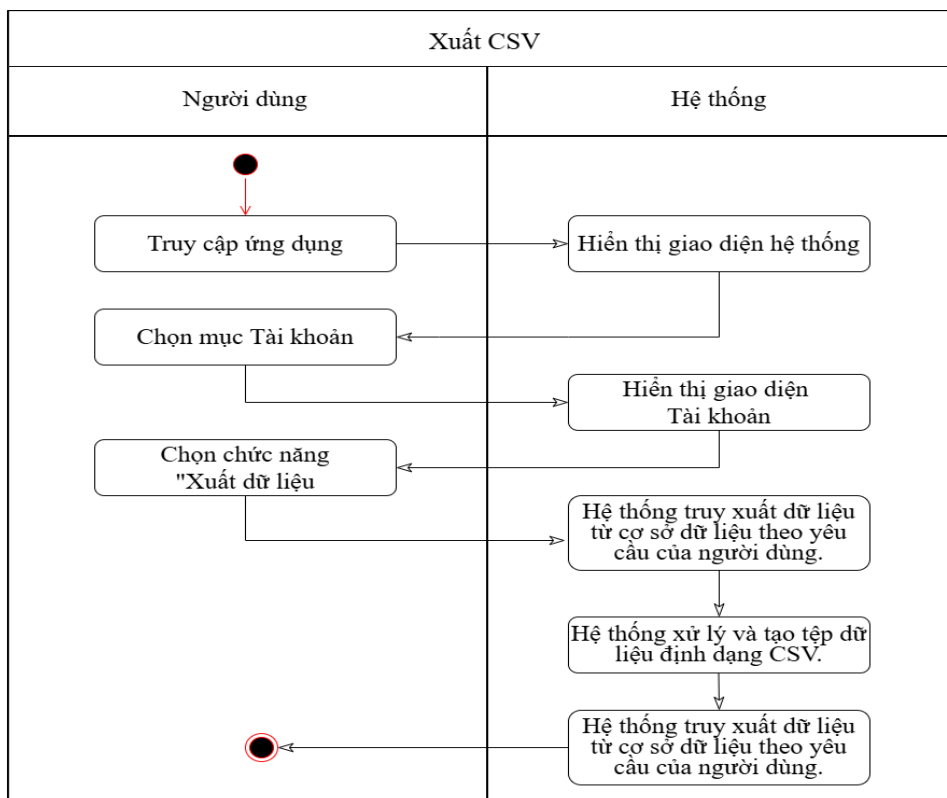
Hình 3.14 Sơ đồ hoạt động xem lịch sử chi tiêu

3.2.3.13 Use case xuất CSV

Tên use case	Xuất CSV.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xuất dữ liệu từ hệ thống ra tệp định dạng CSV nhằm phục vụ mục đích lưu trữ, phân tích hoặc chia sẻ dữ liệu.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng có quyền truy cập dữ liệu cần xuất. Thiết bị có kết nối Internet và đủ bộ nhớ lưu trữ.
Đảm bảo tối thiểu	Hệ thống tạo tệp CSV chứa dữ liệu được yêu cầu. Không có dữ liệu nào bị thay đổi trong hệ thống nếu xảy ra lỗi.
Đảm bảo thành công	Tệp CSV được tạo thành công. Người dùng tải xuống tệp CSV về thiết bị một cách an toàn.
Kích hoạt	Người dùng chọn chức năng “Xuất dữ liệu” trong giao diện Tài khoản.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn mục “Tài khoản”. - 3. Hệ thống hiển thị giao diện Tài khoản. - 4. Người dùng chọn chức năng “Xuất dữ liệu”. - 5. Hệ thống truy xuất dữ liệu từ cơ sở dữ liệu theo yêu cầu của người dùng. - 6. Hệ thống xử lý và tạo tệp dữ liệu định dạng CSV. - 7. Hệ thống cung cấp liên kết tải xuống hoặc tự động tải tệp CSV về thiết bị của người dùng. - 8. Hệ thống hiển thị thông báo “Xuất dữ liệu thành công”. - 9. Ca sử dụng kết thúc.
Ngoại lệ	E1 – Lỗi kết nối cơ sở dữ liệu 5a. Hệ thống không thể truy xuất dữ liệu từ cơ sở dữ liệu.

	<p>5b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thử lại sau.</p> <p>E2 – Lỗi tạo tệp CSV</p> <p>6a. Xảy ra lỗi trong quá trình xử lý và tạo tệp CSV.</p> <p>6b. Hệ thống hiển thị thông báo lỗi xuất dữ liệu.</p> <p>E3 – Bộ nhớ thiết bị không đủ</p> <p>7a. Thiết bị người dùng không đủ bộ nhớ để lưu tệp CSV.</p> <p>7b. Hệ thống hiển thị thông báo yêu cầu người dùng giải phóng bộ nhớ.</p>
Ghi chú	<p>Dữ liệu xuất ra CSV có thể bao gồm các thông tin như thời gian, danh mục, số tiền, ghi chú, ...</p> <p>Tệp CSV có thể được sử dụng cho các công cụ phân tích như Excel hoặc Google Sheets.</p>

Bảng 3.13 Đặc tả use case xuất CSV



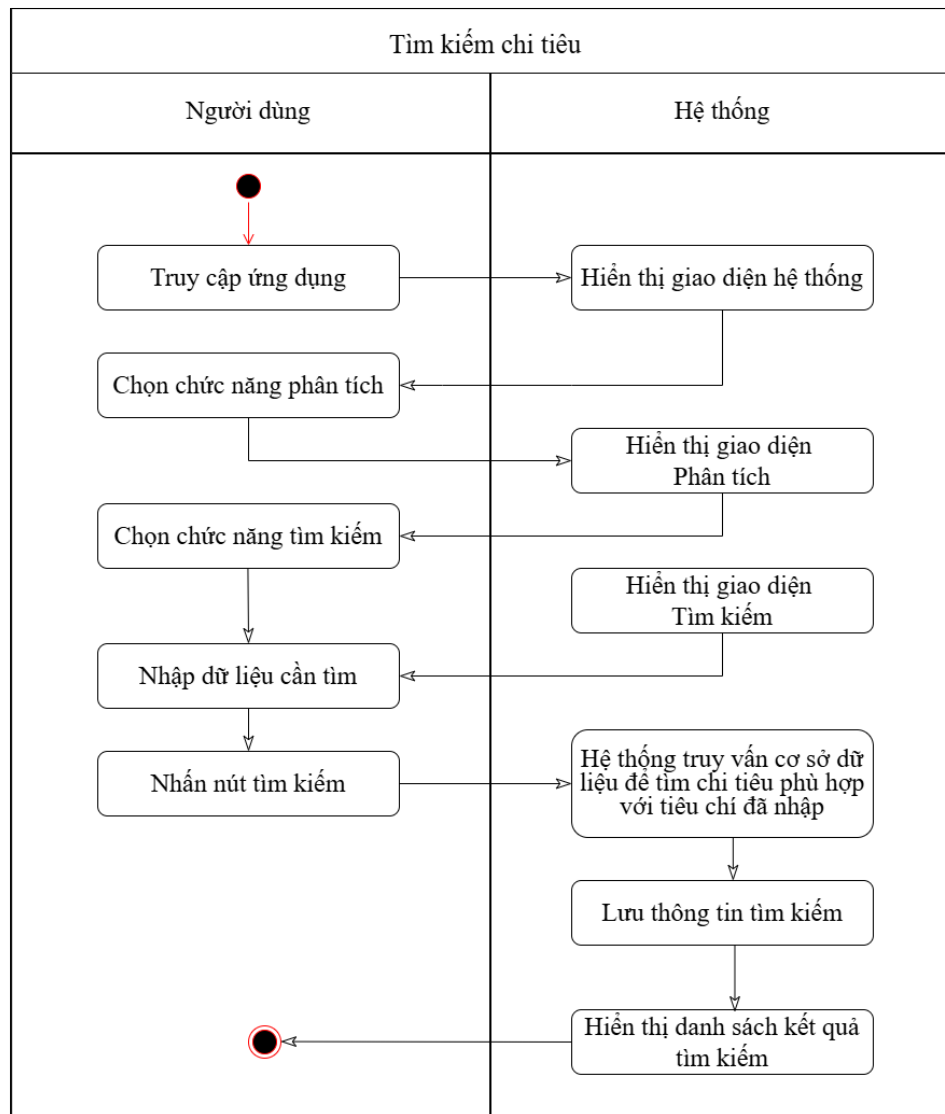
Hình 3.15 Sơ đồ hoạt động xuất CSV

3.2.3.14 Use case tìm kiếm chi tiêu

Tên use case	Tìm kiếm chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập tìm kiếm các khoản chi tiêu được lưu trong hệ thống dựa trên từ khóa hoặc các tiêu chí cụ thể như thời gian, số tiền, danh mục và ghi chú.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có dữ liệu chi tiêu được lưu trong hệ thống. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Giao diện tìm kiếm chi tiêu được hiển thị đầy đủ. Không có dữ liệu chi tiêu nào bị thay đổi nếu xảy ra lỗi.
Đảm bảo thành công	Danh sách các khoản chi tiêu thỏa mãn điều kiện tìm kiếm được hiển thị chính xác. Người dùng có thể xem chi tiết từng khoản chi tiêu trong kết quả tìm kiếm.
Kích hoạt	Người dùng chọn chức năng “Tìm kiếm chi tiêu” trên giao diện “Phân tích”.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn chức năng “Tìm kiếm chi tiêu”. - 3. Hệ thống hiển thị giao diện tìm kiếm với các trường tùy chọn (thời gian, khoảng tiền, danh mục, ghi chú, ...). - 4. Người dùng nhập tiêu chí tìm kiếm theo nhu cầu. - 5. Người dùng nhấn nút “Tìm kiếm”. - 6. Hệ thống truy vấn cơ sở dữ liệu để tìm các khoản chi tiêu phù hợp với tiêu chí đã nhập. - 7. Hệ thống hiển thị danh sách kết quả tìm kiếm, bao gồm thời gian, số tiền, danh mục và ghi chú. - 8. Người dùng chọn khoản chi tiêu để xem chi tiết giao dịch.

	- 9. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không tìm thấy kết quả</p> <p>7a. Hệ thống không tìm thấy khoản chi tiêu nào phù hợp với tiêu chí tìm kiếm.</p> <p>7b. Hệ thống hiển thị thông báo “Không có kết quả phù hợp”.</p> <p>E2 – Chưa nhập tiêu chí tìm kiếm</p> <p>5a. Người dùng nhấn nút tìm kiếm khi chưa nhập tiêu chí.</p> <p>5b. Hệ thống hiển thị thông báo yêu cầu người dùng nhập ít nhất một tiêu chí tìm kiếm.</p> <p>E3 – Lỗi kết nối cơ sở dữ liệu</p> <p>6a. Hệ thống không thể truy vấn dữ liệu từ cơ sở dữ liệu.</p> <p>6b. Hệ thống hiển thị thông báo lỗi hệ thống.</p> <p>E4 – Dữ liệu không đồng bộ hoặc bị lỗi</p> <p>6a. Dữ liệu chi tiêu không đầy đủ hoặc xảy ra lỗi đồng bộ.</p> <p>6b. Hệ thống hiển thị thông báo, yêu cầu người dùng thử lại.</p>
Ghi chú	<p>Hệ thống có thể hỗ trợ tìm kiếm kết hợp nhiều tiêu chí.</p> <p>Kết quả tìm kiếm được cập nhật theo thời gian thực dựa trên dữ liệu hiện có.</p>

Bảng 3.14 Đặc tả use case tìm kiếm chi tiêu



Hình 3.16 Sơ đồ hoạt động tìm kiếm chi tiêu

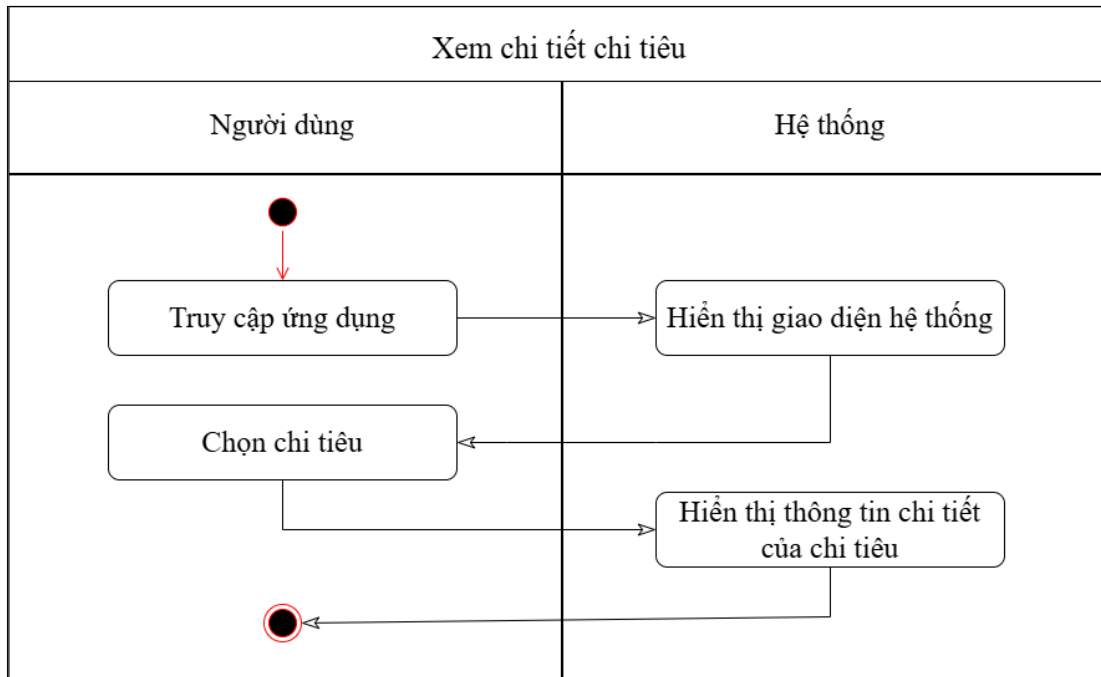
3.2.3.15 Use case xem chi tiết chi tiêu

Tên use case	Xem chi tiết chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xem thông tin chi tiết của một khoản chi tiêu cụ thể được lưu trữ trong hệ thống nhằm phục vụ việc theo dõi và quản lý tài chính cá nhân.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có các khoản chi tiêu được lưu trong hệ thống.

	Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Thông tin chi tiết của khoản chi tiêu được hiển thị đầy đủ. Không có dữ liệu chi tiêu nào bị thay đổi nếu xảy ra lỗi.
Đảm bảo thành công	Người dùng xem được đầy đủ thông tin chi tiết của khoản chi tiêu đã chọn. Dữ liệu hiển thị chính xác với dữ liệu được lưu trong hệ thống.
Kích hoạt	Người dùng chọn một khoản chi tiêu từ danh sách chi tiêu hoặc lịch sử chi tiêu trên giao diện ứng dụng.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn một khoản chi tiêu từ danh sách hoặc giao diện Lịch sử chi tiêu. - 3. Hệ thống truy xuất dữ liệu chi tiết của khoản chi tiêu từ cơ sở dữ liệu. - 4. Hệ thống hiển thị thông tin chi tiết của khoản chi tiêu, bao gồm ngày, giờ, số tiền, danh mục, ghi chú và các thông tin liên quan khác. - 5. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không tìm thấy dữ liệu chi tiêu</p> <p>3a. Hệ thống không tìm thấy dữ liệu của chi tiêu được chọn.</p> <p>3b. Hệ thống hiển thị thông báo lỗi, quay lại danh sách chi tiêu.</p> <p>E2 – Lỗi kết nối cơ sở dữ liệu</p> <p>3a. Xảy ra lỗi trong quá trình truy xuất dữ liệu từ cơ sở dữ liệu.</p> <p>3b. Hệ thống hiển thị thông báo lỗi hệ thống.</p> <p>E3 – Dữ liệu không đồng bộ</p> <p>3a. Dữ liệu chi tiêu hiển thị không khớp với dữ liệu hiện có trong hệ thống.</p> <p>3b. Hệ thống hiển thị thông báo và yêu cầu người dùng thử lại.</p>
Ghi chú	Người dùng chỉ có quyền xem chi tiết các khoản chi tiêu của

	chính mình. Từ giao diện chi tiết, người dùng có thể thực hiện các chức năng chỉnh sửa hoặc xóa chi tiêu (nếu được cho phép).
--	--

Bảng 3.15 Đặc tả use case xem chi tiết chi tiêu



Hình 3.17 Sơ đồ hoạt động xem chi tiết chi tiêu

3.2.3.16 Use case gợi ý chi tiêu thông minh

Tên use case	Gợi ý chi tiêu thông minh.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép hệ thống phân tích dữ liệu chi tiêu của người dùng bằng các thuật toán AI để đưa ra các gợi ý chi tiêu hợp lý, giúp người dùng quản lý tài chính cá nhân hiệu quả hơn.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có dữ liệu chi tiêu được lưu trong hệ thống. Hệ thống AI đã được huấn luyện và sẵn sàng hoạt động. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Hệ thống hiển thị các gợi ý chi tiêu dựa trên dữ liệu hiện có.

	Không có dữ liệu chi tiêu nào của người dùng bị thay đổi.
Đảm bảo thành công	Người dùng nhận được các gợi ý chi tiêu phù hợp với thói quen và tình hình tài chính cá nhân. Gợi ý được hiển thị rõ ràng và dễ hiểu.
Kích hoạt	Người dùng truy cập chức năng “Gợi ý chi tiêu thông minh” trên giao diện ứng dụng.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn chức năng “Gợi ý chi tiêu thông minh”. - 3. Hệ thống truy xuất dữ liệu chi tiêu lịch sử của người dùng. - 4. Hệ thống phân tích dữ liệu bằng mô hình trí tuệ nhân tạo - 5. Hệ thống tạo ra các gợi ý chi tiêu phù hợp (tiết kiệm, điều chỉnh chi tiêu, phân bổ ngân sách). - 6. Hệ thống hiển thị danh sách gợi ý chi tiêu cho người dùng - 7. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không đủ dữ liệu để phân tích</p> <p>3a. Dữ liệu chi tiêu không đủ để hệ thống AI phân tích.</p> <p>3b. Hệ thống hiển thị thông báo yêu cầu người dùng bổ sung dữ liệu.</p> <p>E2 – Lỗi hệ thống AI</p> <p>4a. Xảy ra lỗi trong quá trình phân tích dữ liệu.</p> <p>4b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Gợi ý chỉ mang tính tham khảo, không thay thế quyết định tài chính của người dùng.</p> <p>Độ chính xác của gợi ý phụ thuộc vào dữ liệu chi tiêu lịch sử.</p>

Bảng 3.16 Use case gợi ý chi tiêu thông minh

3.2.3.17 Use case phát hiện chi tiêu bất thường

Tên use case	Phát hiện chi tiêu bất thường.
Tên tác nhân	Người dùng.

Mô tả	Ca sử dụng cho phép hệ thống sử dụng trí tuệ nhân tạo để phát hiện các khoản chi tiêu bất thường so với thói quen chi tiêu trước đó của người dùng, giúp cảnh báo kịp thời các rủi ro tài chính.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có lịch sử chi tiêu đủ lớn để phân tích. Hệ thống AI phát hiện bất thường đã được kích hoạt. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Hệ thống phân tích dữ liệu chi tiêu mà không làm thay đổi dữ liệu gốc. Các khoản chi tiêu được kiểm tra theo mô hình AI.
Đảm bảo thành công	Các khoản chi tiêu bất thường được phát hiện và hiển thị cho người dùng. Người dùng nhận được cảnh báo rõ ràng về chi tiêu bất thường.
Kích hoạt	Hệ thống tự động kích hoạt khi có dữ liệu chi tiêu mới hoặc người dùng chọn chức năng “Phát hiện chi tiêu bất thường”.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Hệ thống truy xuất dữ liệu chi tiêu lịch sử của người dùng. - 3. Hệ thống sử dụng mô hình AI để phân tích hành vi chi tiêu. - 4. Hệ thống so sánh khoản chi tiêu mới với các mẫu chi tiêu trước đó. - 5. Hệ thống phát hiện khoản chi tiêu có dấu hiệu bất thường. - 6. Hệ thống hiển thị cảnh báo các chi tiêu bất thường cho người dùng. - 7. Ca sử dụng kết thúc.
Ngoại lệ	E1 – Không phát hiện chi tiêu bất thường 5a. Không có khoản chi tiêu nào được xác định là bất thường. 5b. Hệ thống hiển thị thông báo “Không phát hiện chi tiêu bất thường”.

	<p>E2 – Lỗi phân tích dữ liệu</p> <p>3a. Dữ liệu chi tiêu bị thiếu hoặc không đồng bộ.</p> <p>3b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Thuật toán phát hiện bất thường sử dụng Isolation Forest.</p> <p>Người dùng có thể xác nhận hoặc bỏ qua cảnh báo.</p>

Bảng 3.17 Use case phát hiện chi tiêu bất thường

3.2.3.18 Use case dự đoán chi tiêu trong tương lai

Tên use case	Dự đoán chi tiêu trong tương lai.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép hệ thống sử dụng trí tuệ nhân tạo để dự đoán mức chi tiêu của người dùng trong tương lai dựa trên dữ liệu chi tiêu lịch sử, hỗ trợ lập kế hoạch tài chính cá nhân.
Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng có dữ liệu chi tiêu lịch sử đầy đủ.</p> <p>Mô hình AI dự đoán đã được huấn luyện.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Hệ thống hiển thị kết quả dự đoán mà không làm thay đổi dữ liệu gốc.</p> <p>Dữ liệu dự đoán được trình bày rõ ràng.</p>
Đảm bảo thành công	<p>Người dùng xem được mức chi tiêu dự đoán theo thời gian (ngày, tháng hoặc năm).</p> <p>Kết quả dự đoán hỗ trợ người dùng lập kế hoạch tài chính.</p>
Kích hoạt	Người dùng chọn chức năng “Dự đoán chi tiêu” trên giao diện ứng dụng.
Chuỗi sự kiện chính	<ul style="list-style-type: none"> - 1. Người dùng đăng nhập vào ứng dụng. - 2. Người dùng chọn chức năng “Dự đoán chi tiêu”. - 3. Hệ thống truy xuất dữ liệu chi tiêu lịch sử của người dùng.

	<ul style="list-style-type: none"> - 4. Hệ thống phân tích dữ liệu bằng mô hình AI dự đoán. - 5. Hệ thống tạo kết quả dự đoán chi tiêu trong tương lai. - 6. Hệ thống hiển thị kết quả dự đoán dưới dạng số liệu hoặc biểu đồ. - 7. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không đủ dữ liệu để dự đoán</p> <p>3a. Dữ liệu lịch sử chi tiêu không đủ để tạo mô hình dự đoán.</p> <p>3b. Hệ thống hiển thị thông báo yêu cầu bổ sung dữ liệu.</p> <p>E2 – Lỗi mô hình dự đoán</p> <p>4a. Mô hình AI không thể tạo kết quả dự đoán.</p> <p>4b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Kết quả dự đoán mang tính tham khảo và có thể sai lệch so với thực tế.</p> <p>Độ chính xác của thuật toán phụ thuộc vào chất lượng và số lượng dữ liệu lịch sử.</p>

Bảng 3.18 Use case dự đoán chi tiêu trong tương lai

3.3 Thiết kế cơ sở dữ liệu

Thiết kế cơ sở dữ liệu (CSDL) cho hệ thống quản lý tài chính cá nhân sử dụng Firebase Firestore.

CSDL được thiết kế để tương thích với cấu trúc NoSQL, nơi dữ liệu được lưu trữ dưới dạng collection và document:

3.3.1 spending Collection (Chi tiêu)

Field	Description
id	Định danh duy nhất cho tài liệu trong collection spending.
money	Số tiền giao dịch ((-) chi phí, (+) thu nhập).
note	Ghi chú về giao dịch.
type	Loại giao dịch, có thể là thu nhập, chi phí, hoặc loại khác.

typeName	Tên loại giao dịch
location	Vị trí của giao dịch
date	Ngày giờ của giao dịch
image	Hình ảnh liên quan đến giao dịch
friends	Danh sách người liên quan đến giao dịch.
isExpense	Cờ xác định giao dịch là chi tiêu.

Bảng 3.19 Bảng mô tả spending Collection

Cấu trúc tài liệu Firestore:

```
{
  "spending": {
    "id": "id",
    "money": "money",
    "note": "note",
    "type": "type",
    "typeName": "typeName",
    "location": "location",
    "date": "date",
    "image": "image",
    "friends": ["friendId1", "friendId2"],
    "day": "day",
    "month": "month",
    "year": "year",
    "weekday": "weekday",
    "hour": "hour",
    "isExpense": "true/false",
    "isIncome": "true/false"
  }
}
```

3.3.2 info Collection (Người dùng)

Field	Description
id	Định danh duy nhất của người dùng trong Firebase Authentication / Firestore
name	Tên đầy đủ của người dùng
birthday	Ngày sinh của người dùng
avatar	Ảnh đại diện người dùng
gender	Giới tính của người dùng.
money	Số tiền hiện có của người dùng
currentAddress	Địa chỉ cư trú hiện tại
maritalStatus	Tình trạng hôn nhân
job	Nghề nghiệp hiện tại
educationLevel	Trình độ học vấn
averageMonthlyIncome	Thu nhập trung bình hàng tháng
lifestyle	Phong cách sống (tiết kiệm, cân bằng, chi tiêu nhiều, ...)
riskTolerance	Mức độ chấp nhận rủi ro tài chính
hobbies	Danh sách sở thích cá nhân
hasCompletedSurvey	Trạng thái hoàn thành khảo sát hành vi tài chính
createdAt	Thời điểm tạo tài khoản

Bảng 3.20 Bảng mô tả info Collection

Cấu trúc tài liệu Firestore:

```
{
  "user": {
    "id": "string",
    "name": "string",
    "birthday": "string",
    "avatar": "string",
    "gender": true,
    "money": 0,
    "currentAddress": "string",
    "maritalStatus": "string",
```

```

"job": "string",
"educationLevel": "string",
"averageMonthlyIncome": 0,
"lifestyle": "string",
"riskTolerance": "string",
"hobbies": [
  "string"
],
"hasCompletedSurvey": true/false,
"createdAt": "timestamp"
}
}

```

3.3.3 budget Collection (Ngân sách)

Field	Description
id	Định danh duy nhất của ngân sách trong Firestore
limitMoney	Giới hạn chi tiêu được thiết lập cho ngân sách
type	Mã loại ngân sách, tương ứng với danh mục chi tiêu
typeName	Tên danh mục ngân sách
month	Tháng áp dụng ngân sách
year	Năm áp dụng ngân sách
isActive	Trạng thái hoạt động của ngân sách
createdAt	Thời điểm tạo ngân sách

Bảng 3.21 Bảng mô tả budget Collection

Cấu trúc tài liệu Firestore:

```

{
  "budget": {
    "id": "string",
    "limitMoney": 0,
    "type": 0,
    "typeName": "string",
    "month": 1,

```

```
"year": 2025,  
  "isActive": true,  
  "createdAt": "timestamp"  
}  
}
```

CHƯƠNG 4 : THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH

4.1 Kiến trúc hệ thống

4.1.1 Cấu trúc phần cứng

Hệ thống được xây dựng theo mô hình ứng dụng di động kết hợp với nền tảng đám mây, không yêu cầu máy chủ vật lý riêng. Thành phần phần cứng của hệ thống bao gồm hai bộ phận chính:

- Máy chủ Firebase: Firebase đóng vai trò là máy chủ trung tâm của hệ thống, chịu trách nhiệm lưu trữ dữ liệu giao dịch, thông tin người dùng và các cấu hình liên quan đến ứng dụng. Dữ liệu được lưu trữ trên nền tảng đám mây thông qua Firestore, cho phép hệ thống hoạt động ổn định và có khả năng mở rộng linh hoạt. Firebase hỗ trợ cơ chế đồng bộ dữ liệu theo thời gian thực, giúp dữ liệu luôn được cập nhật kịp thời giữa các thiết bị.
- Thiết bị người dùng: Thiết bị người dùng bao gồm các điện thoại di động sử dụng hệ điều hành Android và iOS. Người dùng tương tác với hệ thống thông qua ứng dụng được phát triển bằng Flutter. Thiết bị di động đảm nhiệm vai trò hiển thị giao diện, tiếp nhận thao tác của người dùng và gửi dữ liệu đến hệ thống backend để xử lý và lưu trữ.

4.1.2 Kiến trúc phần mềm

Hệ thống được thiết kế theo mô hình client–server, trong đó Flutter đảm nhiệm vai trò phía client và Firebase đóng vai trò backend. Kiến trúc này giúp giảm thiểu công việc triển khai và quản lý máy chủ, đồng thời tối ưu hóa khả năng đồng bộ dữ liệu.

- Frontend (Client):
 - + Sử dụng Flutter, framework phát triển ứng dụng đa nền tảng.
 - + Giao diện được thiết kế trực quan và thân thiện, hỗ trợ các tính năng như quản lý giao dịch, theo dõi mục tiêu và hiển thị báo cáo tài chính.
- Backend (Firebase):
 - + Firebase Authentication: Quản lý đăng nhập, đăng ký và xác thực người dùng.
 - + Firestore Database và Storage: Lưu trữ và truy xuất dữ liệu.
- API tích hợp: Sử dụng lớp API Service trong Flutter để gọi các dịch vụ bên ngoài

4.1.3 Biểu đồ phân cấp chức năng

Hệ thống ứng dụng quản lý chi tiêu cá nhân được thiết kế với các nhóm chức năng chính nhằm hỗ trợ người dùng theo dõi và phân tích tình hình tài chính cá nhân một cách hiệu quả. Các chức năng được phân chia rõ ràng để đảm bảo tính logic, dễ mở rộng và thuận tiện trong quá trình sử dụng:

- Quản lý người dùng: Chức năng quản lý người dùng cho phép người dùng đăng ký tài khoản, đăng nhập và xác thực thông tin thông qua Firebase Authentication. Mỗi tài khoản người dùng được quản lý độc lập, đảm bảo dữ liệu tài chính cá nhân không bị trùng lặp hoặc truy cập trái phép. Ngoài ra, người dùng có thể cập nhật thông tin cá nhân và quản lý trạng thái tài khoản trong quá trình sử dụng ứng dụng.

- Quản lý thu nhập và chi tiêu: Đây là chức năng cốt lõi của ứng dụng, cho phép người dùng ghi nhận các khoản thu nhập và chi tiêu hằng ngày. Người dùng có thể nhập thông tin giao dịch như số tiền, danh mục, thời gian và ghi chú. Hệ thống hỗ trợ phân loại các khoản chi theo từng nhóm như sinh hoạt, học tập, giải trí, mua sắm, ... nhằm giúp việc theo dõi và thống kê tài chính trở nên rõ ràng và dễ dàng hơn.

- Phân tích hành vi chi tiêu: Dựa trên dữ liệu thu chi đã được lưu trữ, hệ thống thực hiện phân tích hành vi chi tiêu của người dùng bằng thuật toán K-Means. Thuật toán này giúp phân nhóm các thói quen chi tiêu, từ đó nhận diện các kiểu hành vi tài chính khác nhau của người dùng. Kết quả phân tích được sử dụng làm cơ sở cho các chức năng gợi ý và cảnh báo trong hệ thống.

- Dự báo thu chi: Chức năng dự báo thu chi sử dụng mô hình LSTM để phân tích chuỗi dữ liệu thu nhập và chi tiêu theo thời gian. Trên cơ sở dữ liệu lịch sử, hệ thống dự đoán xu hướng chi tiêu và thu nhập trong các khoảng thời gian tiếp theo, giúp người dùng chủ động hơn trong việc lập kế hoạch tài chính cá nhân.

- Phát hiện chi tiêu bất thường: Ứng dụng cung cấp các báo cáo tài chính tổng hợp dưới dạng biểu đồ và thống kê theo ngày, tháng và năm. Thông qua các báo cáo này, người dùng có thể theo dõi sự biến động của thu nhập và chi tiêu, đánh giá hiệu quả quản lý tài chính và đưa ra các quyết định phù hợp.

- Gợi ý và hỗ trợ ra quyết định: Dựa trên kết quả phân tích hành vi, dự báo và phát hiện bất thường, hệ thống đưa ra các gợi ý chỉ tiêu phù hợp với từng người dùng. Các gợi ý này giúp người dùng điều chỉnh ngân sách, tối ưu hóa chi tiêu và từng bước hình thành thói quen quản lý tài chính khoa học và bền vững.

4.2 Thiết kế giao diện

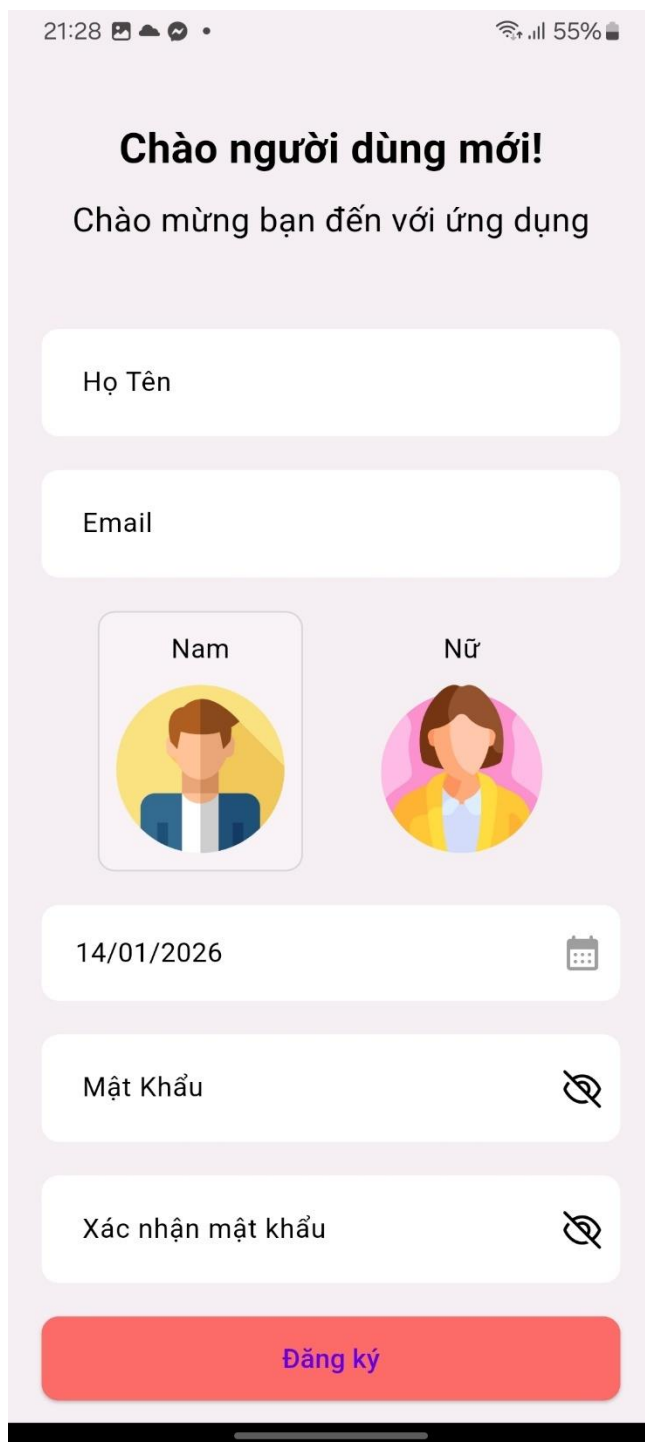
4.2.1 Giao diện đăng nhập

Giao diện đăng nhập: Cho phép người dùng đăng nhập vào hệ thống.

Hình 4.1 Giao diện đăng nhập

4.2.2 Giao diện đăng ký

Giao diện đăng ký: Cho phép người dùng đăng ký tài khoản và tiến hành xác thực tài khoản qua email.

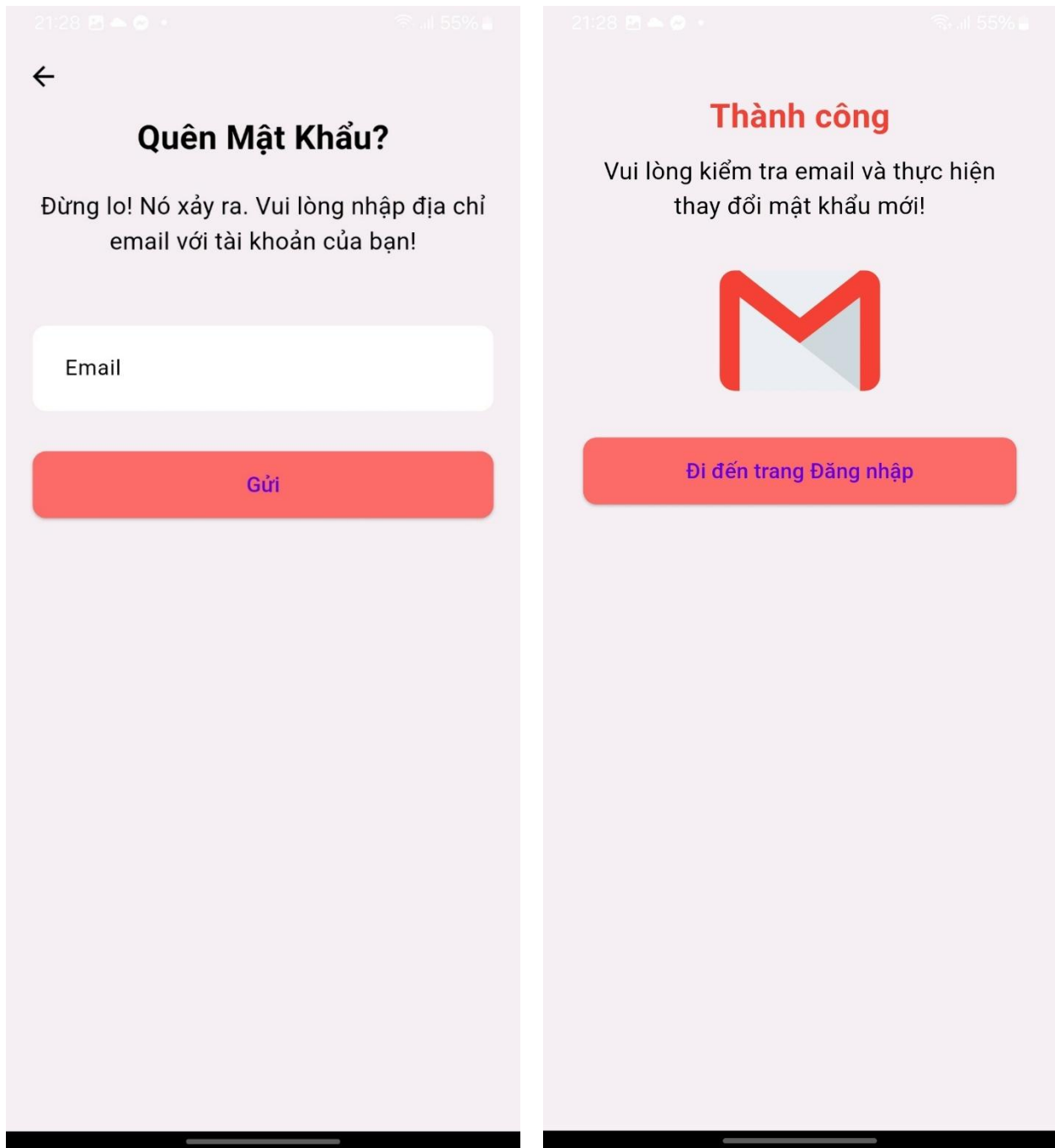


The image shows a mobile application registration screen. At the top, the status bar displays the time 21:28, signal strength, and 55% battery. The main heading is "Chào người dùng mới!" (Hello new user!) followed by the subtitle "Chào mừng bạn đến với ứng dụng" (Welcome you to the application). The form consists of several input fields: "Họ Tên" (Last Name), "Email", a date field showing "14/01/2026" with a calendar icon, "Mật Khẩu" (Password) with a toggle icon, and "Xác nhận mật khẩu" (Confirm password) with a toggle icon. Between the email and date fields are two gender selection options: "Nam" (Male) with a male icon and "Nữ" (Female) with a female icon. At the bottom is a large red button labeled "Đăng ký" (Register).

Hình 4.2 Giao diện đăng ký tài khoản

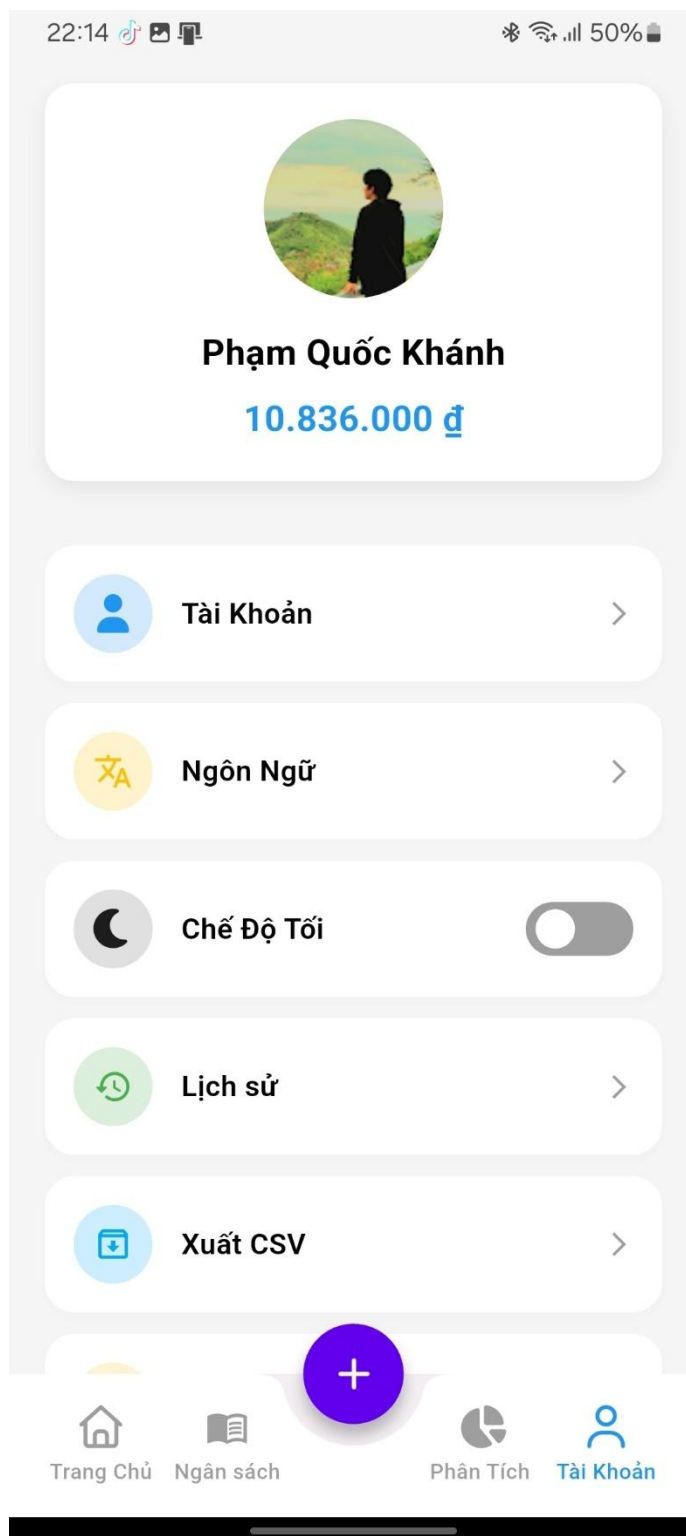
4.2.3 Giao diện quên mật khẩu

Giao diện quên mật khẩu: Cho phép người dùng thay đổi mật khẩu qua email.



Hình 4.3 Giao diện quên mật khẩu

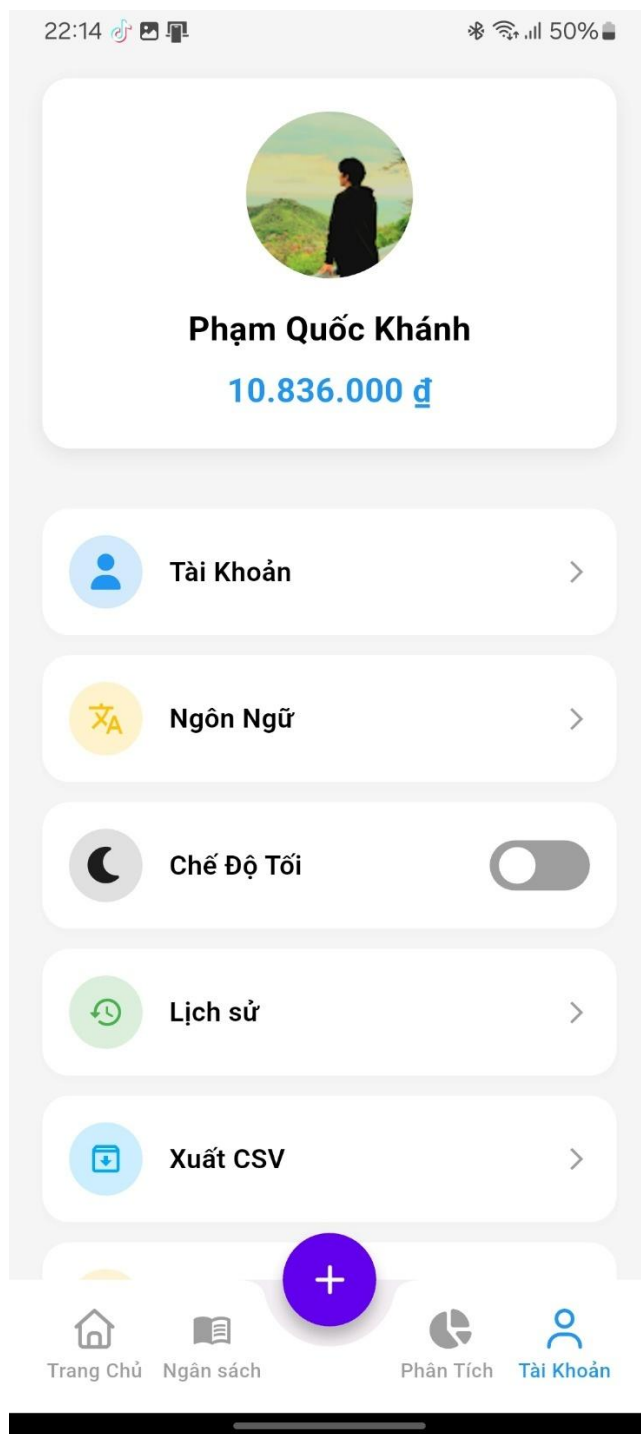
4.2.4 Giao diện đổi mật khẩu



Hình 4.4 Giao diện đổi mật khẩu

4.2.5 Giao diện tài khoản

Giao diện tài khoản: Cho phép người dùng chỉnh sửa giao diện sáng/tối, đổi ngôn ngữ Việt/Anh, xem thông tin tài khoản, xuất CSV và lịch sử giao dịch.



Hình 4.5 Giao diện tài khoản

4.2.6 Giao diện xem thông tin cá nhân

Giao diện xem thông tin cá nhân: Cho phép người dùng xem lại thông tin của mình



Hình 4.6 Giao diện xem thông tin cá nhân

4.2.7 Giao diện chỉnh sửa thông tin cá nhân

Giao diện chỉnh sửa thông tin cá nhân: Cho phép người dùng chỉnh sửa một số thông tin cá nhân và đổi ảnh đại diện.

The image displays two mobile application screens for account management, both titled "Tài Khoản" (Account). The left screen shows the profile view with a circular profile picture, a plus icon for changing it, and input fields for "Họ Tên" (Last Name) with value "Phạm Quốc Khánh", "Ngày Sinh" (Date of Birth) with value "07/03/2003", and "Nơi ở hiện tại" (Current Address) with value "Đắk Lắk". The right screen shows the edit view with dropdown menus for "Nơi ở hiện tại" (Đắk Lắk), "Tình trạng hôn nhân" (Độc thân), "Ngành nghề" (Học sinh), "Trình độ học vấn" (Đại học), "Lối sống" (Tiết kiệm), "Khẩu vị rủi ro" (Thấp), and "Sở thích" (Xem phim, Nghe nhạc, Podcast, Chơi ...). A red "Lưu" (Save) button is at the bottom.

Hình 4.7 Giao diện chỉnh sửa thông tin cá nhân

4.2.8 Giao diện thêm chi tiêu

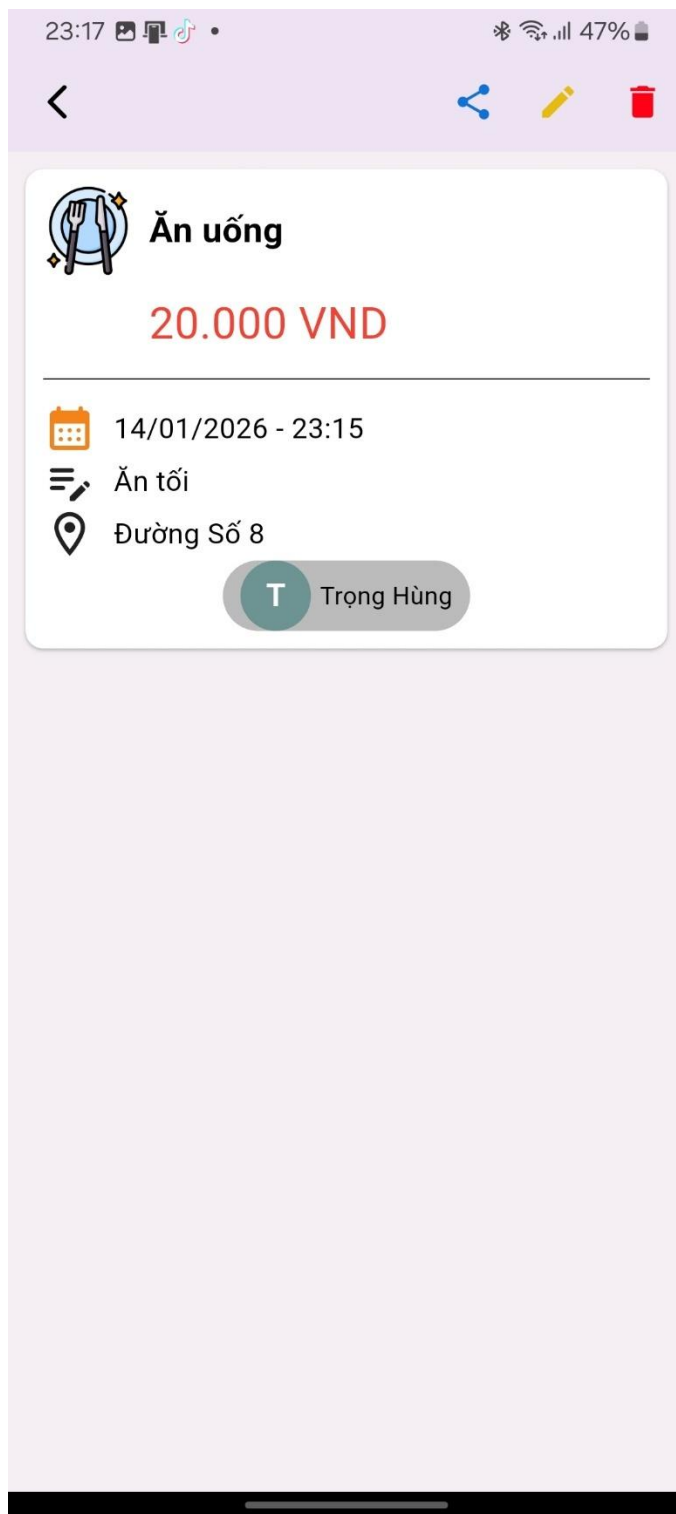
Giao diện thêm chi tiêu: Cho phép người dùng thêm mới các khoản thu/chi.

The screenshot shows a mobile application interface for adding an expense. At the top, the status bar displays the time 21:34, signal strength, and 54% battery. The app's title bar is purple with a close button (X) on the left, the title 'Thêm Chi Tiêu' in the center, and a 'Lưu' (Save) button on the right. Below the title bar is a light purple rounded rectangle containing the amount '100.000 đ'. The main content area is a light pink rounded rectangle containing several input fields: a 'Loại' (Category) field with a question mark icon, a date field showing '14/01/2026' with a calendar icon, a time field showing '21:33' with a clock icon, and a 'Ghi Chú' (Note) field with a list icon. Below these is a 'Vị Trí' (Location) field with a location pin icon, and a 'Bạn bè' (Friends) field with a group of people icon. At the bottom of the form are two buttons: a gallery icon and a camera icon. Below the form is a purple button labeled 'Ẩn bớt' (Hide more) with a downward arrow. The bottom of the screen shows a black home indicator bar.

Hình 4.8 Giao diện thêm chi tiêu

4.2.9 Giao diện xem chi tiết chi tiêu

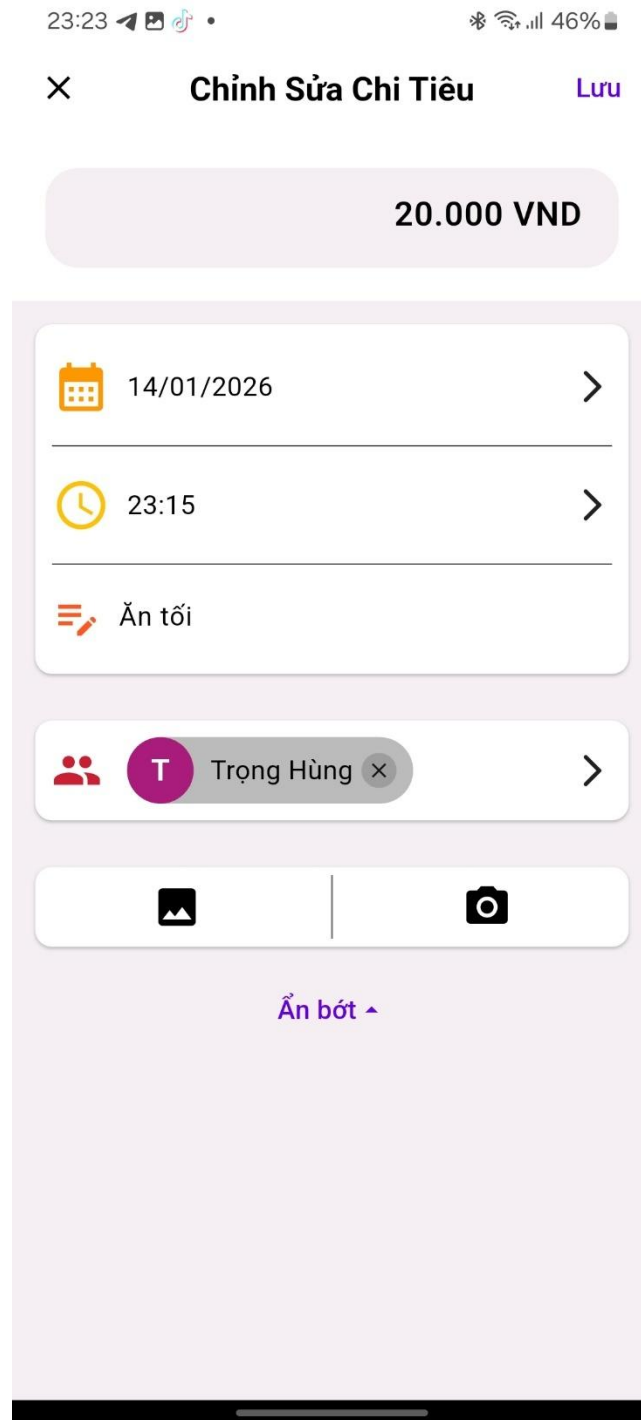
Giao diện xem chi tiết chi tiêu: Cho phép người dùng xem lại chi tiết các chi tiêu.



Hình 4.9 Giao diện xem chi tiết chi tiêu

4.2.10 Giao diện chỉnh sửa chi tiêu

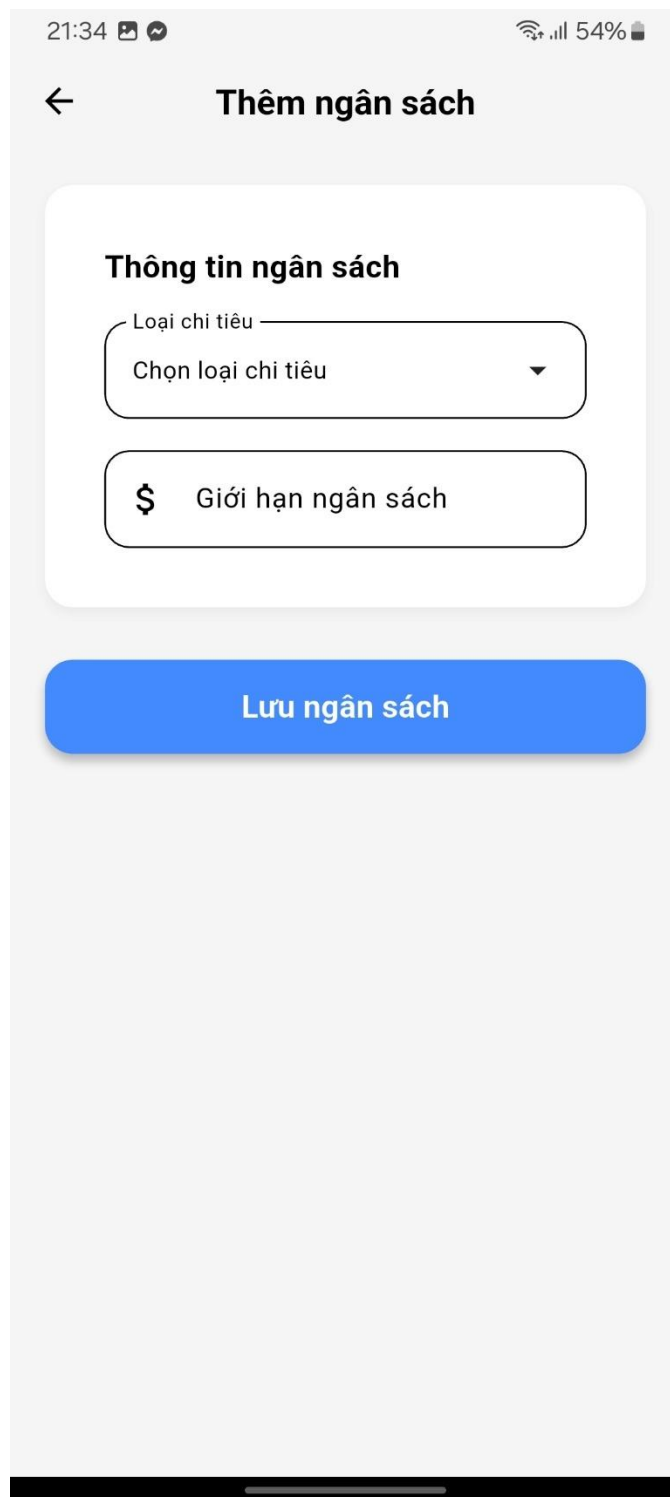
Giao diện chỉnh sửa chi tiêu: Cho phép người dùng thay đổi thông tin chi tiêu.



Hình 4.10 Giao diện chỉnh sửa chi tiêu

4.2.11 Giao diện thêm ngân sách

Giao diện thêm ngân sách: Cho phép người dùng thêm mới ngân sách.



The screenshot shows a mobile application interface for adding a budget. At the top, the status bar displays the time 21:34, signal strength, and 54% battery. The app's header features a back arrow and the title "Thêm ngân sách". Below this, a white card titled "Thông tin ngân sách" contains two input fields: a dropdown menu labeled "Loại chi tiêu" with the placeholder text "Chọn loại chi tiêu", and a text input field labeled "\$ Giới hạn ngân sách". A prominent blue button labeled "Lưu ngân sách" is positioned below the card. The bottom of the screen shows a black home indicator bar.

Hình 4.11 Giao diện thêm ngân sách

4.2.12 Giao diện chỉnh sửa ngân sách

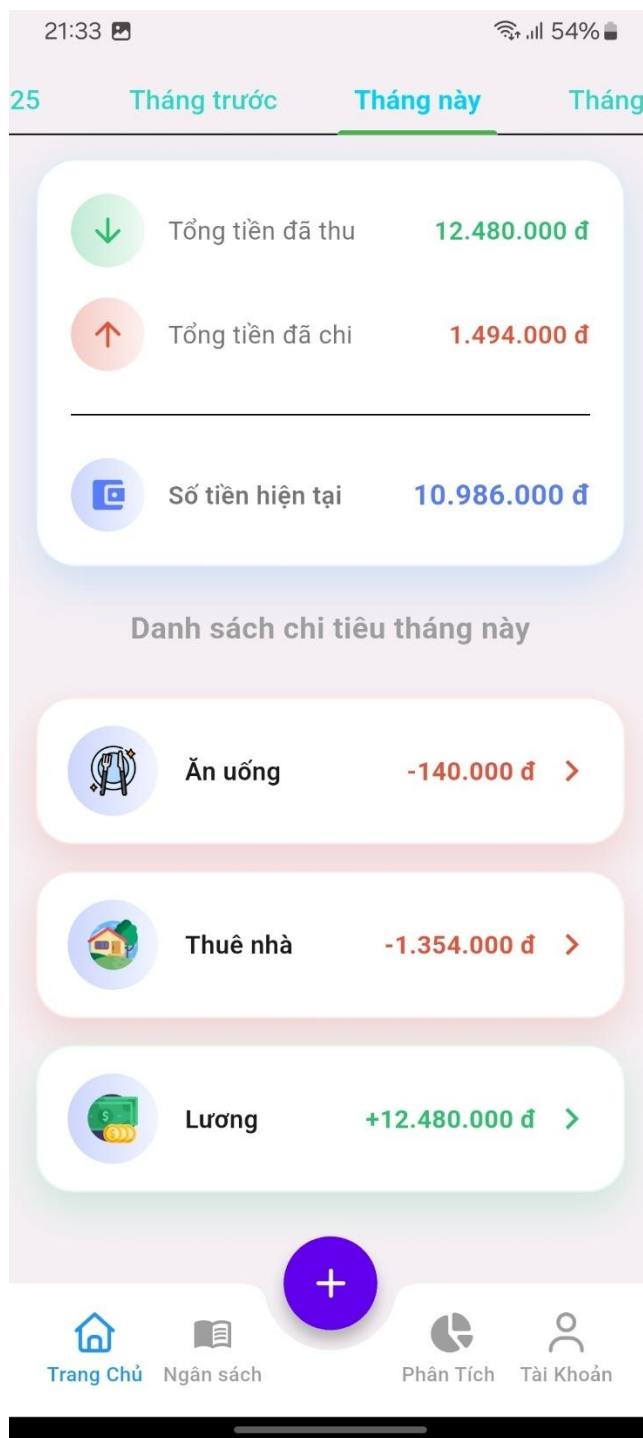
Giao diện chỉnh sửa ngân sách: Cho phép người dùng chỉnh sửa ngân sách.

The screenshot shows a mobile application interface for editing a budget. At the top, the status bar displays the time 21:34, signal strength, and 54% battery. The app's header bar has a back arrow, the title 'Chỉnh sửa ngân sách', and a red trash icon. Below the header is a white rounded rectangle containing the title 'Thông tin ngân sách'. Inside this rectangle are two form fields: a dropdown menu labeled 'Loại chi tiêu' with 'Ăn uống' selected, and a text input labeled 'Giới hạn ngân sách' with '\$ 900000'. Below the white rectangle is a large blue button with the text 'Cập nhật ngân sách'.

Hình 4.12 Giao diện chỉnh sửa ngân sách

4.2.13 Giao diện trang chủ

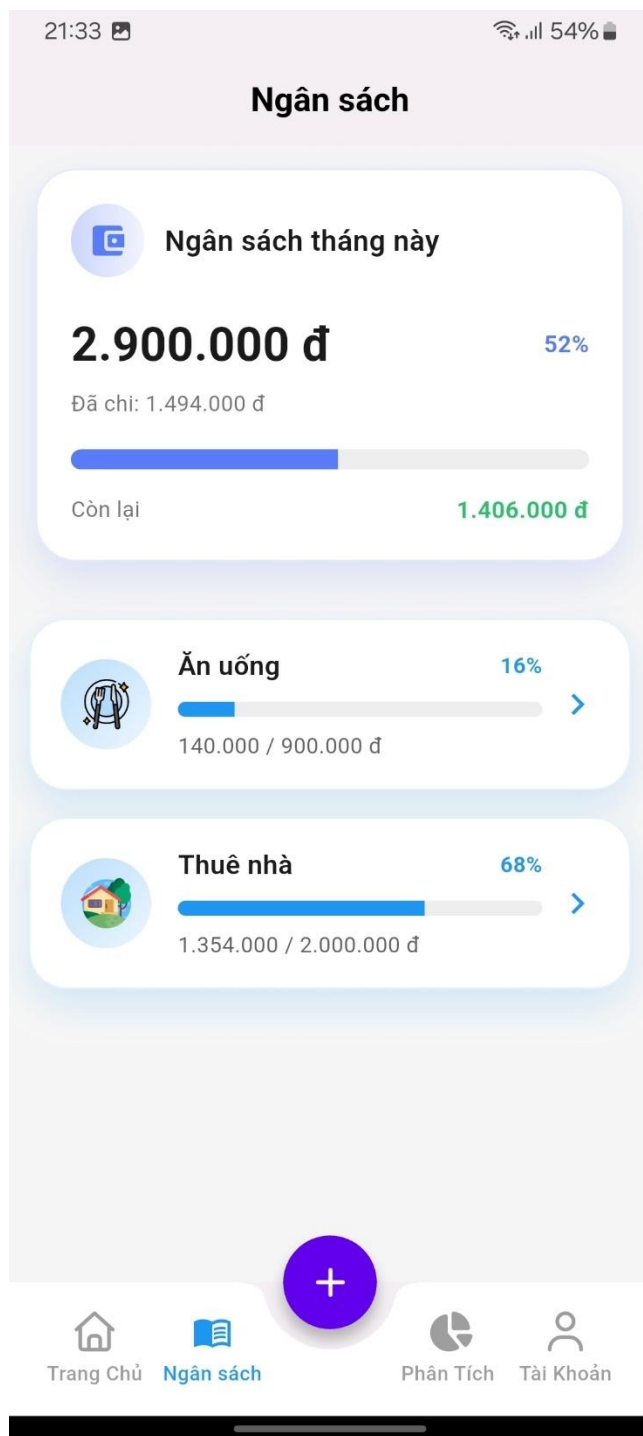
Giao diện trang chủ: Cho phép người dùng xem tổng thu nhập, tổng chi tiêu hàng tháng và xem danh sách chi tiêu của tháng hiện tại.



Hình 4.13 Giao diện trang chủ

4.2.14 Giao diện ngân sách

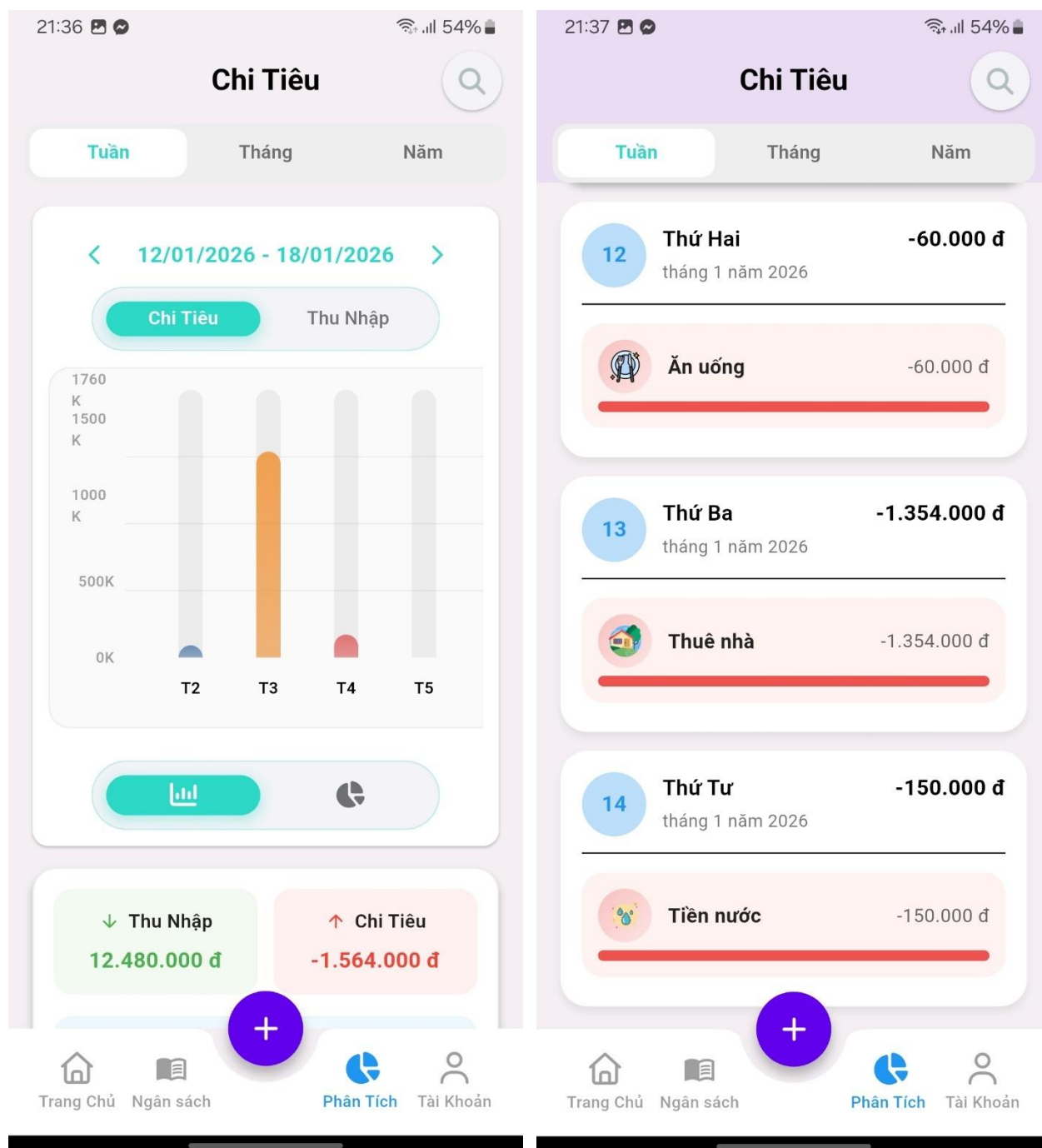
Giao diện ngân sách: Cho phép người dùng xem tổng ngân sách của tháng, tình trạng ngân sách hiện tại và danh sách ngân sách theo từng danh mục.



Hình 4.14 Giao diện ngân sách

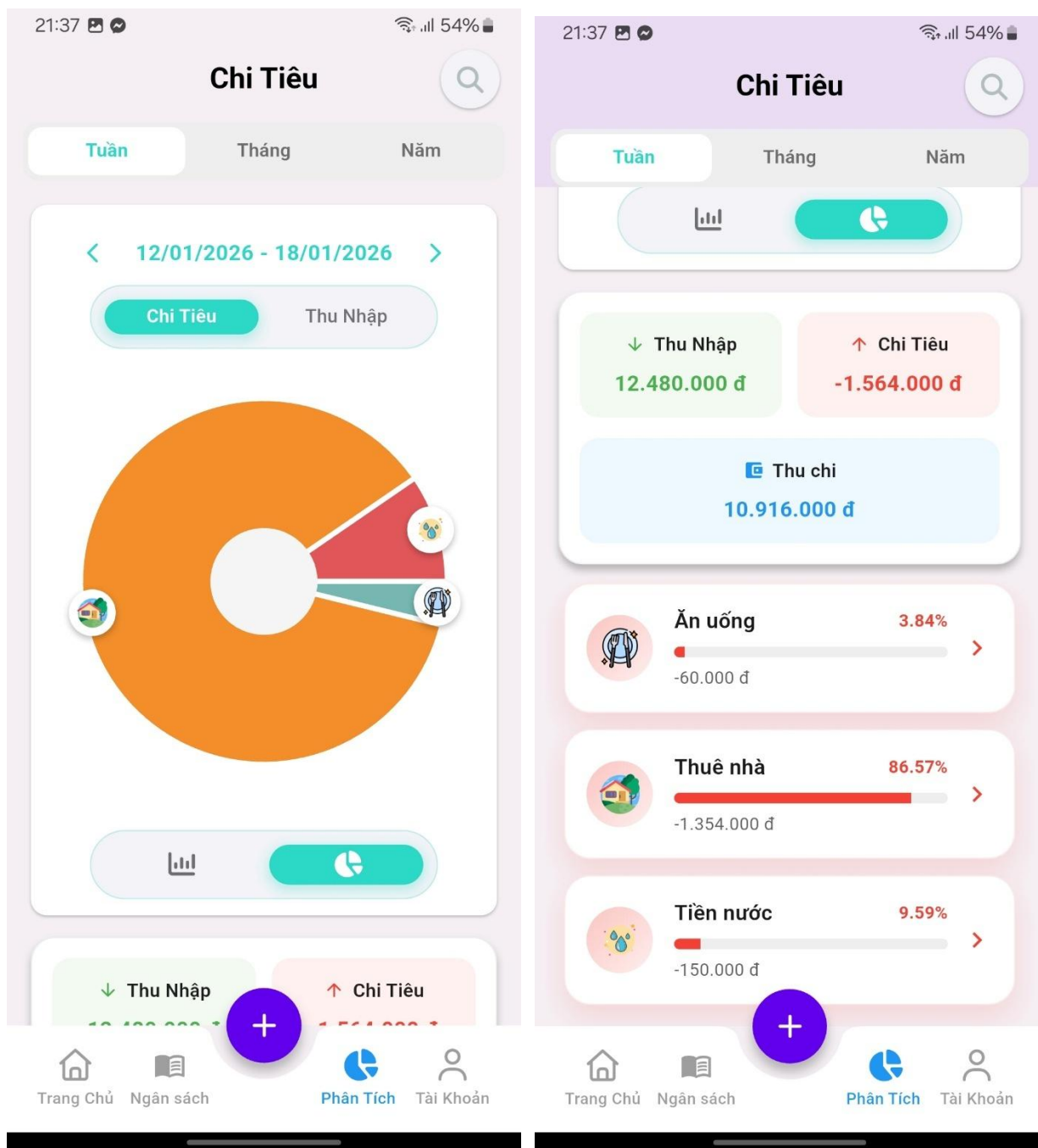
4.2.15 Giao diện phân tích

Giao diện phân tích bằng biểu đồ cột: Cho phép người dùng so sánh giao dịch của tuần, tháng, năm và hiển thị danh sách sử dụng từ đầu tháng đến cuối tháng.



Hình 4.15 Giao diện phân tích bằng biểu đồ cột

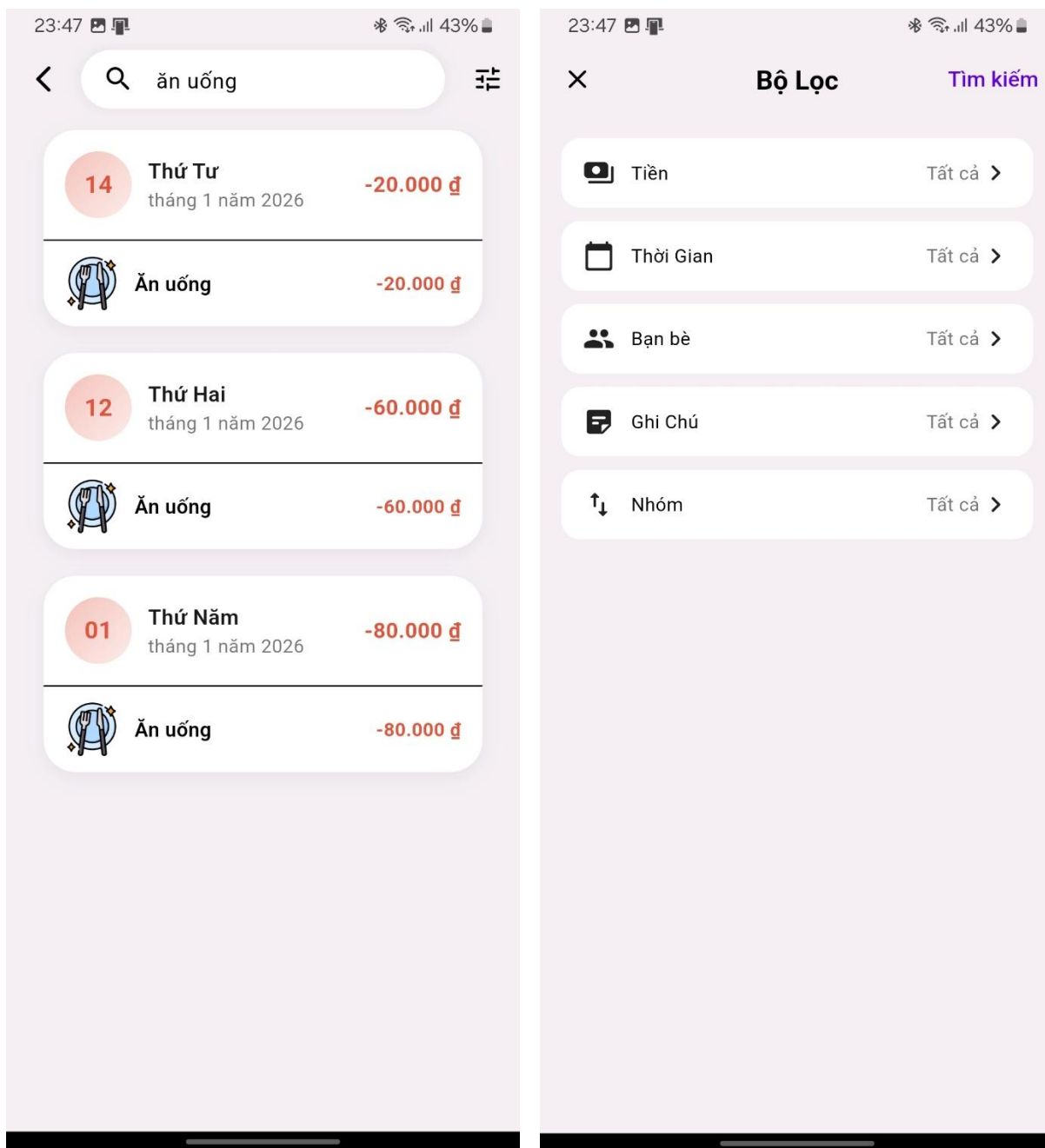
Giao diện phân tích bằng biểu đồ tròn: Chop phéo người dùng so sánh giao dịch của tuần, tháng, năm và hiển thị danh sách sử dụng từ đầu tháng đến cuối tháng.



Hình 4.16 giao diện phân tích bằng biểu đồ tròn

4.2.16 Giao diện tìm kiếm và bộ lọc tìm kiếm

Giao diện tìm kiếm và bộ lọc tìm kiếm: cho phép người dùng lọc và tìm kiếm chi tiêu một cách dễ dàng.



Hình 4.17 Giao diện tìm kiếm và bộ lọc tìm kiếm

KẾT LUẬN VÀ KIẾN NGHỊ

5.1 Kết quả đạt được

Trong quá trình nghiên cứu và triển khai đề tài “Nghiên cứu kết hợp thuật toán K-Means, Isolation Forest với mô hình LSTM để xây dựng ứng dụng quản lý chi tiêu cá nhân”, nhóm đã đạt được một số kết quả quan trọng như sau:

- Xây dựng thành công một ứng dụng quản lý tài chính cá nhân trên nền tảng di động (Android và iOS) sử dụng Flutter, đáp ứng được các nhu cầu cơ bản của người dùng trong việc ghi nhận, theo dõi và thống kê thu/chi hằng ngày.

- Thiết kế và triển khai hệ thống cơ sở dữ liệu trên Firebase, cho phép lưu trữ, đồng bộ và quản lý dữ liệu tài chính cá nhân theo thời gian thực, đảm bảo tính nhất quán và an toàn dữ liệu.

- Hoàn thiện các mô-đun chức năng chính của hệ thống như: quản lý người dùng, quản lý thu/chi, phân loại theo danh mục, báo cáo tài chính trực quan bằng biểu đồ.

- Nghiên cứu và ứng dụng thành công các thuật toán học máy và học sâu vào bài toán quản lý tài chính cá nhân:

- + Mô hình LSTM được sử dụng để dự báo xu hướng thu nhập và chi tiêu theo chuỗi thời gian.

- + Thuật toán K-Means hỗ trợ phân cụm hành vi chi tiêu, giúp nhận diện các nhóm thói quen chi tiêu đặc trưng của người dùng.

- + Thuật toán Isolation Forest được áp dụng để phát hiện các giao dịch chi tiêu bất thường hoặc vượt ngưỡng thông thường.

- Tích hợp các mô hình AI vào ứng dụng thông qua TensorFlow Lite, cho phép xử lý trực tiếp trên thiết bị di động, giảm độ trễ và tăng cường quyền riêng tư cho người dùng.

- Thực hiện kiểm thử và đánh giá hệ thống, cho thấy ứng dụng hoạt động ổn định, giao diện thân thiện, các chức năng cơ bản đáp ứng đúng yêu cầu đặt ra ban đầu.

Nhìn chung, đề tài đã đạt được mục tiêu đề ra là xây dựng một hệ thống quản lý chi tiêu cá nhân có tích hợp trí tuệ nhân tạo, hỗ trợ người dùng theo dõi, phân tích và định hướng hành vi tài chính một cách khoa học hơn.

Ưu điểm của hệ thống:

- Hệ thống ứng dụng quản lý chi tiêu cá nhân được xây dựng trong đề tài có một số ưu điểm nổi bật như sau:

- Tính đa nền tảng cao: Việc sử dụng Flutter giúp ứng dụng có thể hoạt động ổn định trên cả Android và iOS với một mã nguồn duy nhất, tiết kiệm thời gian phát triển và dễ dàng mở rộng trong tương lai.

- Giao diện thân thiện, trực quan: Ứng dụng được thiết kế với bố cục rõ ràng, dễ sử dụng, phù hợp với nhiều đối tượng người dùng, đặc biệt là sinh viên và người mới đi làm.

- Khả năng quản lý tài chính toàn diện: Hệ thống hỗ trợ đầy đủ các chức năng từ ghi nhận thu nhập, chi tiêu, phân loại danh mục đến thống kê, báo cáo và theo dõi biến động tài chính theo thời gian.

- Ứng dụng trí tuệ nhân tạo trong phân tích dữ liệu: Việc kết hợp các thuật toán LSTM, K-Means và Isolation Forest giúp hệ thống không chỉ dừng lại ở mức thống kê, mà còn có khả năng phân tích hành vi, dự báo xu hướng và phát hiện bất thường.

- Tính cá nhân hóa cao: Các gợi ý chi tiêu và cảnh báo được xây dựng dựa trên dữ liệu và hành vi thực tế của từng người dùng, giúp nâng cao hiệu quả quản lý tài chính cá nhân.

- Bảo mật và đồng bộ dữ liệu tốt: Firebase hỗ trợ xác thực người dùng, đồng bộ dữ liệu theo thời gian thực và đảm bảo an toàn thông tin tài chính cá nhân.

- Khả năng mở rộng: Kiến trúc hệ thống được thiết kế theo hướng mô-đun, thuận lợi cho việc bảo trì, nâng cấp và bổ sung các chức năng mới.

5.2 Hạn chế của hệ thống

Bên cạnh những kết quả đạt được, hệ thống vẫn còn tồn tại một số hạn chế nhất định:

- Dữ liệu huấn luyện còn hạn chế: Các mô hình AI chủ yếu được huấn luyện trên tập dữ liệu có quy mô chưa lớn, do đó độ chính xác của kết quả dự báo và phân tích hành vi vẫn còn phụ thuộc nhiều vào chất lượng và số lượng dữ liệu đầu vào.

- Chưa tự động hóa hoàn toàn việc thu thập dữ liệu: Phần lớn dữ liệu thu nhập và chi tiêu vẫn cần người dùng nhập thủ công, điều này có thể gây bất tiện và ảnh hưởng đến trải nghiệm sử dụng lâu dài.

- Khả năng gợi ý còn ở mức cơ bản: Các gợi ý chỉ tiêu và cảnh báo chủ yếu dựa trên các ngưỡng và mô hình thống kê – học máy cơ bản, chưa kết hợp sâu với các yếu tố như mục tiêu tài chính dài hạn, tình hình kinh tế hoặc biến động thu nhập bất thường.

- Chưa hỗ trợ đầy đủ các nghiệp vụ tài chính nâng cao: Ứng dụng chưa đi sâu vào các chức năng như quản lý đầu tư chi tiết, quản lý nợ, lập kế hoạch tài chính dài hạn.

- Hiệu năng AI trên thiết bị còn phụ thuộc phần cứng: Việc triển khai mô hình trực tiếp trên thiết bị di động có thể gặp hạn chế về hiệu năng đối với các thiết bị cấu hình thấp.

5.3 Hướng phát triển của hệ thống

Trong thời gian tới, hệ thống có thể được tiếp tục nghiên cứu và phát triển theo một số hướng sau:

- Mở rộng và cải thiện chất lượng dữ liệu: Thu thập thêm dữ liệu thực tế từ nhiều người dùng hơn nhằm nâng cao độ chính xác của các mô hình học máy và học sâu.

- Tự động hóa việc ghi nhận giao dịch: Tích hợp với ngân hàng số, ví điện tử hoặc SMS banking để tự động ghi nhận thu nhập và chi tiêu, giảm thao tác thủ công cho người dùng.

- Nâng cao khả năng gợi ý thông minh: Kết hợp thêm các mô hình AI nâng cao hoặc các kỹ thuật học tăng cường (Reinforcement Learning) để đưa ra các gợi ý chỉ tiêu và lập kế hoạch tài chính sát với mục tiêu cá nhân hơn.

- Mở rộng chức năng quản lý tài chính: Bổ sung các tính năng như quản lý nợ, quản lý đầu tư, lập kế hoạch tài chính dài hạn.

- Cải thiện trải nghiệm người dùng (UX/UI): Tiếp tục tối ưu giao diện, tăng tính cá nhân hóa và khả năng tùy chỉnh theo thói quen sử dụng của từng người dùng.

- Triển khai trên nền tảng web: Mở rộng hệ thống sang nền tảng web để tăng tính linh hoạt và khả năng tiếp cận của người dùng.

- Tăng cường bảo mật dữ liệu: Áp dụng thêm các cơ chế mã hóa, xác thực nâng cao nhằm đảm bảo an toàn tuyệt đối cho dữ liệu tài chính cá nhân.

PHỤ LỤC

Phụ lục: hướng dẫn cài đặt

Yêu cầu hệ thống:

- Ứng dụng được thiết kế để cài đặt và sử dụng trên thiết bị di động, do đó yêu cầu hệ thống tập trung vào cấu hình tối thiểu của thiết bị người dùng nhằm đảm bảo ứng dụng hoạt động ổn định.

- Thiết bị:

+ Điện thoại thông minh hoặc máy tính bảng.

- Hệ điều hành:

+ Android 8.0 trở lên hoặc

+ iOS 13.0 trở lên.

- Cấu hình phần cứng tối thiểu:

+ RAM từ 3 GB.

+ Dung lượng bộ nhớ trống tối thiểu 100 MB.

- Kết nối mạng:

+ Thiết bị cần có kết nối Internet để phục vụ đăng nhập, xác thực người dùng và đồng bộ dữ liệu.

Cài đặt ứng dụng trên thiết bị di động:

- Bước 1: Tải file cài đặt của ứng dụng từ nguồn được cung cấp.

- Bước 2: Trên thiết bị Android, cho phép cài đặt ứng dụng từ nguồn không xác định (nếu hệ thống yêu cầu).

- Bước 3: Mở file cài đặt và tiến hành cài đặt ứng dụng.

- Bước 4: Sau khi cài đặt hoàn tất, mở ứng dụng từ màn hình chính để bắt đầu sử dụng.

TÀI LIỆU THAM KHẢO

- [1] The Saylor Foundation, *Personal Finance*. Washington, DC, USA, 2012.
- [2] Office of Investor Education and Advocacy, U.S. Securities and Exchange Commission, *Saving and Investing: A Roadmap to Your Financial Security Through Saving and Investing*, Washington, DC, USA, 2011.
- [3] The Dart Team, *Dart Programming Language Specification*, 5th edition, Version 2.10, Google, 2021.
- [4] Rap Payne, *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*, Berkeley, CA, USA: Apress, 2019.
- [5] Google Firebase Team, “Firebase Documentation,” Google. [Online]. Available: <https://firebase.google.com/docs> [Accessed: 18-Jan-2026].
- [6] J. Hunt, *A Beginners Guide to Python 3 Programming*, Cham, Switzerland: Springer Nature, 2023.
- [7] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., Sebastopol, CA, USA: O’Reilly Media, 2019.
- [8] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Upper Saddle River, NJ, USA: Prentice Hall, 2020.
- [9] T. M. Mitchell, *Machine Learning*, New York, NY, USA: McGraw-Hill, 1997.
- [10] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., San Francisco, CA, USA: Morgan Kaufmann, 2012.
- [11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., New York, NY, USA: Springer, 2009.
- [12] F. T. Liu, K. M. Ting, and Z.-H. Zhou, *Isolation Forest*, in Proc. IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, 2008, pp. 413–422.
- [13] C. C. Aggarwal, *Outlier Analysis*, 2nd ed. Cham, Switzerland: Springer, 2017.
- [14] F. M. Salem, *Recurrent Neural Networks: From Simple to Gated Architectures*, Cham, Switzerland: Springer, 2022.