

**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

a. Công nghệ: Flutter.

b. Công cụ: Visual Studio Code, Android Studio, Firebase, Draw.io, Figma.

c. Ngôn ngữ lập trình: Dart, Python.

d. 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	xi
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.	6
1.4 Phương pháp nghiên cứu	6
1.5 Cấu trúc báo cáo đồ án.....	9
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT.....	11
2.1 Tổng quan về quản lý tài chính cá nhân	11
2.1.1 Khái niệm.....	11
2.1.2 Lợi ích của việc quản lý tài chính cá nhân	11
2.1.3 Mục đích của quản lý tài chính cá nhân	12
2.2 Nền tảng công nghệ phát triển ứng dụng.....	12
2.2.1 Tổng quan về ngôn ngữ lập trình Dart	12
2.2.2 Tổng quan về Framework Flutter	13
2.2.3 Tổng quan về nền tảng Firebase	14
2.3 Trí tuệ nhân tạo và học máy trong phân tích dữ liệu	15
2.3.1 Tổng quan về ngôn ngữ lập trình Python	15
2.3.2 Tổng quan về AI	16
2.3.3 Tổng quan về ML	17

2.3.4 Thuật toán K-Means	17
2.3.5 Thuật toán Isolation Forest	20
2.3.6 Mạng nơ-ron hồi tiếp LSTM	21
CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG	23
3.1 Lựa chọn mô hình phát triển.....	23
3.2 Thiết kế chức năng của hệ thống	24
3.2.1 Sơ đồ phân rã chức năng.....	24
3.2.2 Sơ đồ Use case	25
3.2.3 Đặc tả Use case	26
3.3 Thiết kế cơ sở dữ liệu	42
3.3.1 spending Collection (Chi tiêu).....	42
3.3.2 info Collection (Người dùng)	43
3.3.3 budget Collection (Ngân sách)	44
CHƯƠNG 4 : THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH	45
4.1 Kiến trúc hệ thống	45
4.1.1 Cấu trúc phần cứng.....	45
4.1.2 Kiến trúc phần mềm.....	45
4.1.3 Sơ đồ kiến trúc phần mềm hệ thống	46
4.1.4 Biểu đồ phân cấp chức năng.....	48
4.2 Thiết kế giao diện	49
4.2.1 Nguyên tắc thiết kế giao diện	49
4.2.2 Thiết kế các màn hình chức năng chính	50
4.3 Một số chức năng chính của hệ thống	51
4.3.1 Quản lý thu nhập và chi tiêu	52
4.3.2 Thiết lập và quản lý ngân sách	54
4.3.3 Thống kê và báo cáo tài chính	55
4.3.4 Phân cụm hành vi chi tiêu.....	56
4.3.5 Dự báo thu chi trong tương lai.....	59
4.3.6 Phát hiện chi tiêu bất thường	61

KẾT LUẬN VÀ KIẾN NGHỊ.....	64
5.1 Kết quả đạt được.....	64
5.2 Hạn chế.....	65
5.3 Hướng phát triển.....	66
PHỤ LỤC	67
TÀI LIỆU THAM KHẢO	81

DANH MỤC CHỮ VIẾT TẮT

STT	Từ viết tắt	Thuật ngữ	Ghi chú
1	K-Means	K-Means Clustering	
2	LSTM	Long Short-Term Memory	
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	18
Hình 2.2 Nguyên lý hoạt động Isolation Forest	20
Hình 2.3 Cấu trúc LSTM.....	21
Hình 3.1 Sơ đồ phân rã chức năng	24
Hình 3.2 Sơ đồ Use case.....	25
Hình 3.3 Sơ đồ hoạt động đăng nhập	27
Hình 3.4 Sơ đồ hoạt động đăng ký	29
Hình 3.5 Sơ đồ hoạt động quên mật khẩu	31
Hình 3.7 Sơ đồ hoạt động thêm chỉ tiêu	33
Hình 3.8 Sơ đồ hoạt động sửa chỉ tiêu	35
Hình 3.9 Sơ đồ hoạt động xóa chỉ tiêu	37
Hình 4.1 Chức năng thêm, sửa, xóa giao dịch.....	53
Hình 4.2 Chức năng phân loại và theo dõi lịch sử giao dịch.....	53
Hình 4.3 Chức năng thiết lập và điều chỉnh ngân sách.	54
Hình 4.4 Chức năng theo dõi và cảnh báo chỉ tiêu.....	55
Hình 4.5 Chức năng thống kê và báo cáo tài chính.....	56
Hình 4.6 Chức năng Phân cụm hành vi chỉ tiêu	58
Hình 4.7 Chức năng dự báo thu chi trong tương lai.....	60
Hình 4.8 Chức năng phát hiện chỉ tiêu bất thường.....	63

DANH MỤC BẢNG BIỂU

Bảng 3.1 Đặc tả use case đăng nhập.....	26
Bảng 3.2 Đặc tả use case đăng ký	28
Bảng 3.3 Đặc tả use case quên mật khẩu.....	30
Bảng 3.4 Đặc tả use case thêm chi tiêu	32
Bảng 3.5 Đặc tả use case sửa chi tiêu.....	34
Bảng 3.6 Đặc tả use case xóa chi tiêu	36
Bảng 3.7 Use case phân tích và báo cáo tài chính.....	38
Bảng 3.8 Use case phân cụm hành vi chi tiêu	39
Bảng 3.9 Use case phát hiện chi tiêu bất thường.....	40
Bảng 3.10 Use case dự đoán chi tiêu trong tương lai.....	41
Bảng 3.11 Bảng mô tả spending Collection	42
Bảng 3.12 Bảng mô tả info Collection	43
Bảng 3.13 Bảng mô tả budget Collection.....	44

DANH MỤC PHỤ LỤC

Phụ lục 1 Use Case chi tiết của hệ thống.....	70
Phụ lục 2 Use case thay đổi thông tin cá nhân	71
Phụ lục 3 Use case thay đổi ngôn ngữ.....	72
Phụ lục 4 Use case xem thông tin cá nhân	72
Phụ lục 5 Use case tùy chỉnh sáng tối	73
Phụ lục 6 Use case xem lịch sử chi tiêu	74
Phụ lục 7 Đặc tả use case xuất CSV	75
Phụ lục 8 Use case tìm kiếm chi tiêu.....	76
Phụ lục 9 Use case xem chi tiết chi tiêu	77
Phụ lục 10 Use case thêm ngân sách	78
Phụ lục 11 Use case sửa ngân sách	79
Phụ lục 12 Use case xóa ngân sách	80

CHƯƠNG 1 : MỞ ĐẦU

1.1 Tổng quan về đề tài.

Trong những năm gần đây, dưới tác động phát triển mạnh mẽ của ngành công nghệ thông tin và các nền tảng số, hoạt động quản lý và sử dụng tài chính cá nhân của con người đã có nhiều chuyển biến rõ rệt. Sự phổ biến ngày càng rộng rãi của các thiết bị điện tử như điện thoại thông minh, máy tính bảng và máy tính cá nhân ngày càng trở nên phổ biến đã thúc đẩy sự phát triển lan rộng của các hình thức thanh toán không dùng tiền mặt, tiêu biểu như ngân hàng số, ví điện tử và các nền tảng thanh toán trực tuyến đã góp phần mang lại sự thuận tiện và nhanh chóng trong các giao dịch hằng ngày, đặc biệt đối với giới trẻ và người dân đang sinh sống tại khu vực đô thị. Mặc dù vậy, trên thực tế, tiền mặt vẫn giữ vai trò nhất định trong nhiều hoạt động chi tiêu thường ngày, nhất là đối với các giao dịch nhỏ lẻ hoặc tại những khu vực chưa phổ biến phương thức thanh toán điện tử, dẫn đến tình trạng nhiều khoản thu chi không được ghi chép và theo dõi đầy đủ. Điều này khiến người dùng gặp khó khăn trong việc nắm bắt chính xác dòng tiền thực tế của bản thân, từ đó làm giảm hiệu quả quản lý tài chính cá nhân. Bên cạnh đó, đời sống hiện đại đặt ra ngày càng nhiều nhu cầu chi tiêu đa dạng, từ các khoản sinh hoạt cơ bản như ăn uống, đi lại, học tập và giải trí cho đến những mục tiêu dài hạn như tiết kiệm, đầu tư hay xây dựng quỹ dự phòng. Khi số lượng và tần suất các giao dịch ngày càng gia tăng, việc kiểm soát chi tiêu dần trở nên phức tạp hơn, đặc biệt đối với sinh viên và những người mới đi làm nằm trong nhóm đối tượng có thu nhập còn hạn chế trong khi nhu cầu chi tiêu không ngừng mở rộng, đồng thời kinh nghiệm và kỹ năng quản lý tài chính cá nhân vẫn chưa thật sự vững vàng. Thực tế này khiến họ dễ rơi vào tình trạng chi tiêu vượt quá khả năng tài chính của bản thân mà không nhận ra nguyên nhân, nhất là khi chưa phân biệt rõ giữa các khoản chi cần thiết và những khoản chi có thể điều chỉnh. Trong bối cảnh đó, sự phát triển của AI và các phương pháp phân tích dữ liệu hiện đại đã mở ra những hướng tiếp cận mới trong việc hỗ trợ quản lý và ra quyết định tài chính cá nhân. Thông qua việc khai thác và phân tích dữ liệu thu chi trong quá khứ, các hệ thống có thể nhận diện những quy luật, xu hướng cũng như hành vi chi tiêu đặc trưng của từng cá nhân, từ đó không chỉ dừng lại ở việc thống kê số liệu mà còn có khả

năng phân tích, dự báo và đưa ra các gợi ý phù hợp với thói quen, hoàn cảnh và mục tiêu tài chính cụ thể của người dùng, góp phần nâng cao hiệu quả quản lý tài chính cá nhân theo hướng chủ động, khoa học và bền vững.

Mặc dù hiện nay trên thị trường đã xuất hiện nhiều ứng dụng và công cụ hỗ trợ quản lý tài chính cá nhân như Mint, YNAB, Money Lover hay PocketGuard, song 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 thời gian ngắn rồi bỏ dở do cảm thấy việc nhập liệu thủ công tốn nhiều thời gian hoặc chưa mang lại giá trị sử dụng rõ ràng. Bên cạnh đó, phần lớn các ứng dụng hiện nay chủ yếu dừng lại ở chức năng ghi chép và thống kê các khoản thu chi theo ngày, tháng hoặc theo từng danh mục, trong khi chưa thực sự hỗ trợ người dùng hiểu sâu hơn về hành vi chi tiêu của bản thân. Các báo cáo được 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; dù cần thiết, nhưng những thông tin này 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 và cách phân bổ ngân sách sao cho phù hợp với mức thu nhập hiện tại, từ đó gặp không ít khó khăn, đặc biệt đối với những người chưa có nhiều kinh nghiệm trong việc quản lý tài chính cá nhân.

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. Trọng tâm của đề tài là việc thu

thập, lưu trữ và xử lý dữ liệu thu/chi 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. Trong đó, mô hình LSTM được sử dụng để dự báo 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 nhằm phân cụm hành vi chi tiêu, qua đó nhận diện các nhóm thói quen chi tiêu đặc trưng của từng người dùng; đồng thời thuật toán Isolation Forest hỗ trợ phát hiện các giao dịch chi tiêu bất thường hoặc vượt ngưỡng so với hành vi thông thường để hệ thống có thể 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 giúp ứng dụng có thể hoạt động ổn định trên cả hai hệ điều hành Android và iOS, đồng thời giảm thiểu chi phí và thời gian phát triển. Dữ liệu tài chính cá nhân của người dùng được lưu trữ và đồng bộ thông qua nền tảng Firebase, qua đó đảm bảo các yêu cầu về tính bảo mật, tính toàn vẹn của dữ liệu cũng như khả năng truy cập và cập nhật theo thời gian thực. Bên cạnh đó, các chức năng phân tích và hỗ trợ ra quyết định được xây dựng dựa trên việc xử lý và phân tích dữ liệu thu chi của người dùng, tập trung vào việc nhận diện thói quen và xu hướng chi tiêu, từ đó đưa ra các gợi ý phù hợp nhằm hỗ trợ người dùng quản lý tài chính cá nhân một cách hiệu quả và chủ động hơn.

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.

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, Mô hình 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: Nghiên cứu tập trung vào việc xây dựng và đánh giá một ứng dụng hỗ trợ quản lý tài chính cá nhân trên nền tảng số. Phạm vi nghiên cứu giới hạn ở các chức năng thu thập, lưu trữ và phân tích dữ liệu thu chi cá nhân của người dùng, đồng thời đề xuất các gợi ý chi tiêu nhằm nâng cao hiệu quả quản lý tài chính. Đối tượng nghiên cứu là người dùng cá nhân, không mở rộng đến các nghiệp vụ tài chính doanh nghiệp hay đầu tư chuyên sâu.

1.4 Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

- Tìm hiểu công nghệ Flutter và ngôn ngữ lập trình Dart: Nghiên cứu các khái niệm cơ bản về Flutter và ngôn ngữ Dart, phục vụ cho việc xây dựng ứng dụng quản lý tài chính cá nhân đa nền tảng với giao diện thân thiện, trực quan và khả năng vận hành ổn định.

- 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 dữ liệu thu nhập, chi tiêu, tiết kiệm và đầu tư của người dùng, đảm bảo tính an toàn, bảo mật và hiệu quả trong quá trình truy xuất dữ liệu.

- Nghiên cứu các công cụ và thư viện hỗ trợ phát triển ứng dụng Flutter: Tìm hiểu các công cụ, thư viện và framework cần thiết trong Flutter nhằm xây dựng các chức năng quản lý tài chính cá nhân, bao gồm quản lý trạng thái, thiết kế UI và các thư viện hỗ trợ kết nối cơ sở dữ liệu.

- Tìm hiểu các khái niệm cơ bản về AI/ML: Nghiên cứu các khái niệm và nguyên lý cơ bản của trí tuệ nhân tạo và học máy, tập trung vào những mô hình phù hợp với bài toán phân tích dữ liệu tài chính cá nhân. Nội dung nghiên cứu bao gồm các mô hình chuỗi thời gian như LSTM để dự báo xu hướng thu nhập và chi tiêu, thuật toán K-Means để phân tích và phân nhóm hành vi chi tiêu của người dùng, cùng với thuật toán Isolation Forest nhằm phát hiện các giao dịch bất thường. Qua đó, đề tài hướng đến việc làm rõ khả năng ứng dụng AI/ML trong phân tích hành vi chi tiêu và hỗ trợ đưa ra các gợi ý tài chính phù hợp cho người dùng.

- Nghiên cứu các thư viện và công cụ hỗ trợ xây dựng mô hình học máy trong môi trường Python, tiêu biểu như scikit-learn, nhằm phục vụ cho việc phân tích hành vi chi tiêu và phát hiện các giao dịch bất thường. Các mô hình được triển khai trên phía máy chủ và tích hợp với ứng dụng Flutter thông qua API, giúp hệ thống xử lý dữ liệu hiệu quả và đảm bảo tính ổn định trong quá trình sử dụng.

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.

- Phát triển hệ thống xử lý dữ liệu và backend: Xây dựng các chức năng xử lý dữ liệu phía máy chủ nhằm quản lý thông tin người dùng và các giao dịch tài chính cá nhân. Backend có nhiệm vụ tiếp nhận, lưu trữ và truy xuất dữ liệu thu chi từ cơ sở dữ liệu, đồng thời thực hiện các tác vụ phân tích, tổng hợp và cung cấp dữ liệu cho ứng dụng phía người dùng. Các mô hình học máy được triển khai tại backend để phân tích hành vi chi tiêu, phát hiện các giao dịch bất thường và hỗ trợ đưa ra các gợi ý chi tiêu thông minh, đảm bảo hệ thống hoạt động ổn định, an toàn và đáp ứng yêu cầu xử lý dữ liệu trong thực tế.

- 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 đồ án

Chương 1: Mở đầu

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

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

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

Chương 4: Thiết kế và cài đặt chương trình

- Kiến trúc hệ thống
- Thiết kế giao diện

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.
- Hỗ trợ ra quyết định tài chính: Quản lý tài chính cá nhân giúp người dùng theo dõi và đánh giá chính xác tình hình thu chi, từ đó đưa ra các quyết định hợp lý trong chi tiêu, tiết kiệm và đầu tư, góp phần nâng cao hiệu quả sử dụng nguồn tài chính cá nhân.

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. Ngoài ra, Flutter sử dụng ngôn ngữ lập trình Dart, giúp tối ưu hiệu năng và hỗ trợ mạnh mẽ cho việc xây dựng giao diện theo hướng component. Hệ thống widget phong phú và linh hoạt của Flutter cho phép lập trình viên dễ dàng thiết kế giao diện đồng nhất, có khả năng tùy biến cao trên nhiều nền tảng khác nhau.

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. Các hệ thống AI có khả năng học hỏi từ dữ liệu thông qua các thuật toán học máy, từ đó cải thiện độ chính xác và hiệu quả theo thời gian. AI ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, giáo dục, tài chính và thương mại điện tử nhằm hỗ trợ tự động hóa và nâng cao chất lượng 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 Euclide, giữa mỗi điểm dữ liệu và tâm của các cụm.

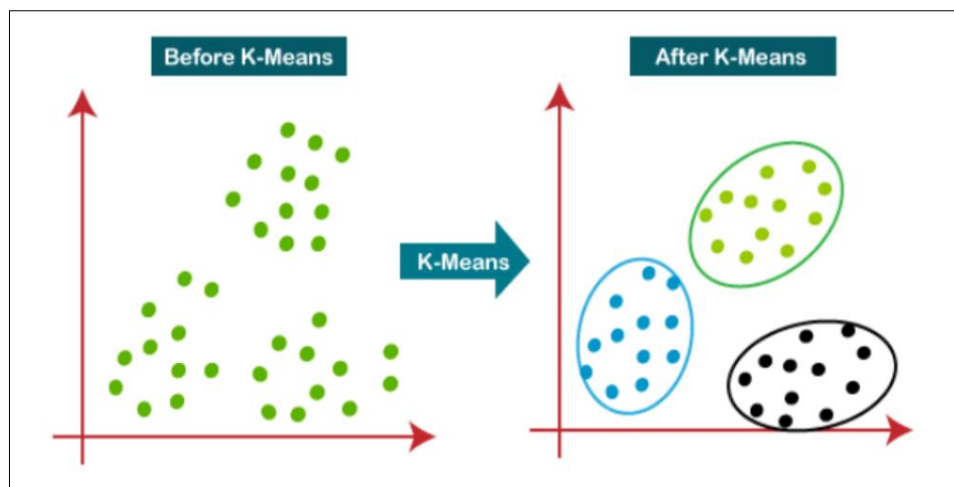
- Mục tiêu chính của thuật toán là tối thiểu hóa tổng bình phương khoảng cách giữa các điểm dữ liệu và tâm cụm mà chúng được gán vào, từ đó đảm bảo tính chặt chẽ và hợp lý của các cụm hình thành.

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.

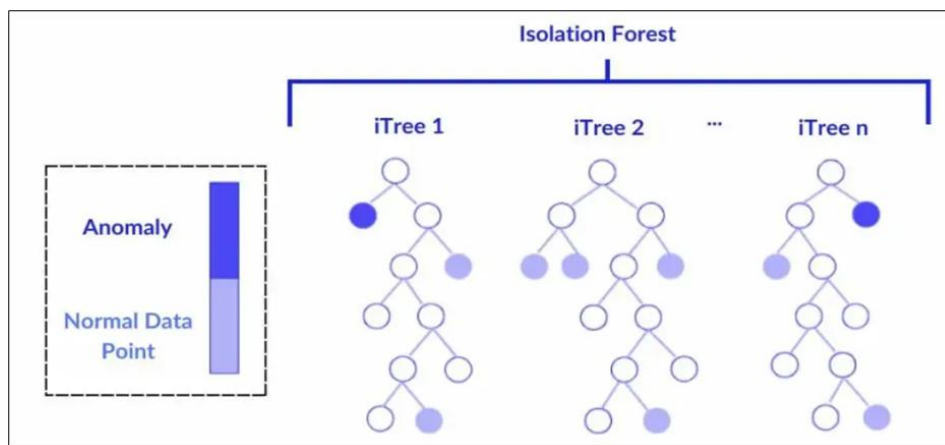
2.3.5 Thuật toán Isolation Forest

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.

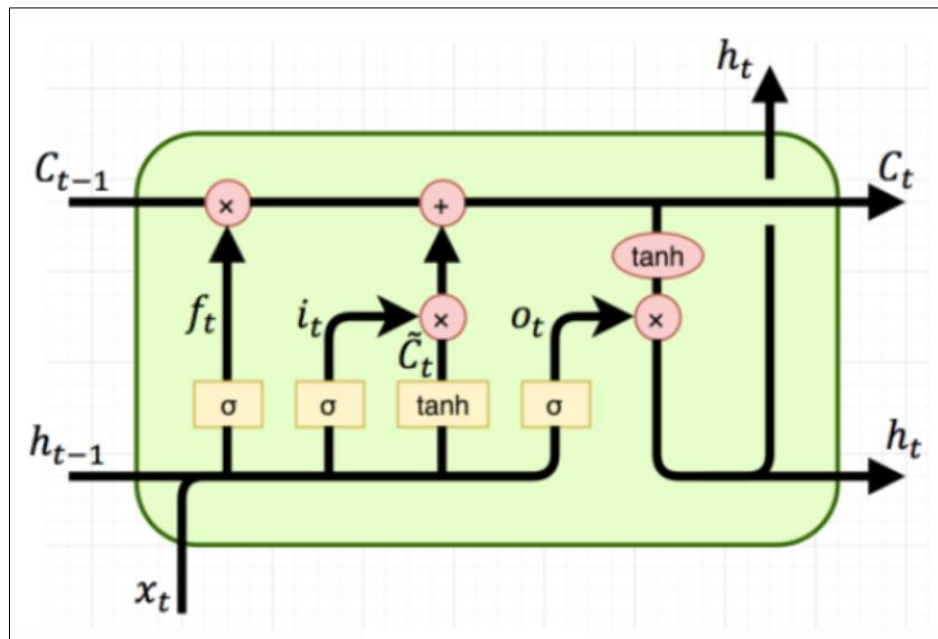
- Phân vùng đệ quy: Quá trình chọn ngẫu nhiên các đặc trưng và phân chia tiếp tục theo cách đệ quy. Cây phát triển cho đến khi mỗi điểm dữ liệu được tách biệt trong nút lá của chính nó hoặc đạt đến độ sâu cây tối đa. Tính ngẫu nhiên trong việc lựa chọn và phân chia đặc trưng đảm bảo rằng các cây không 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]:

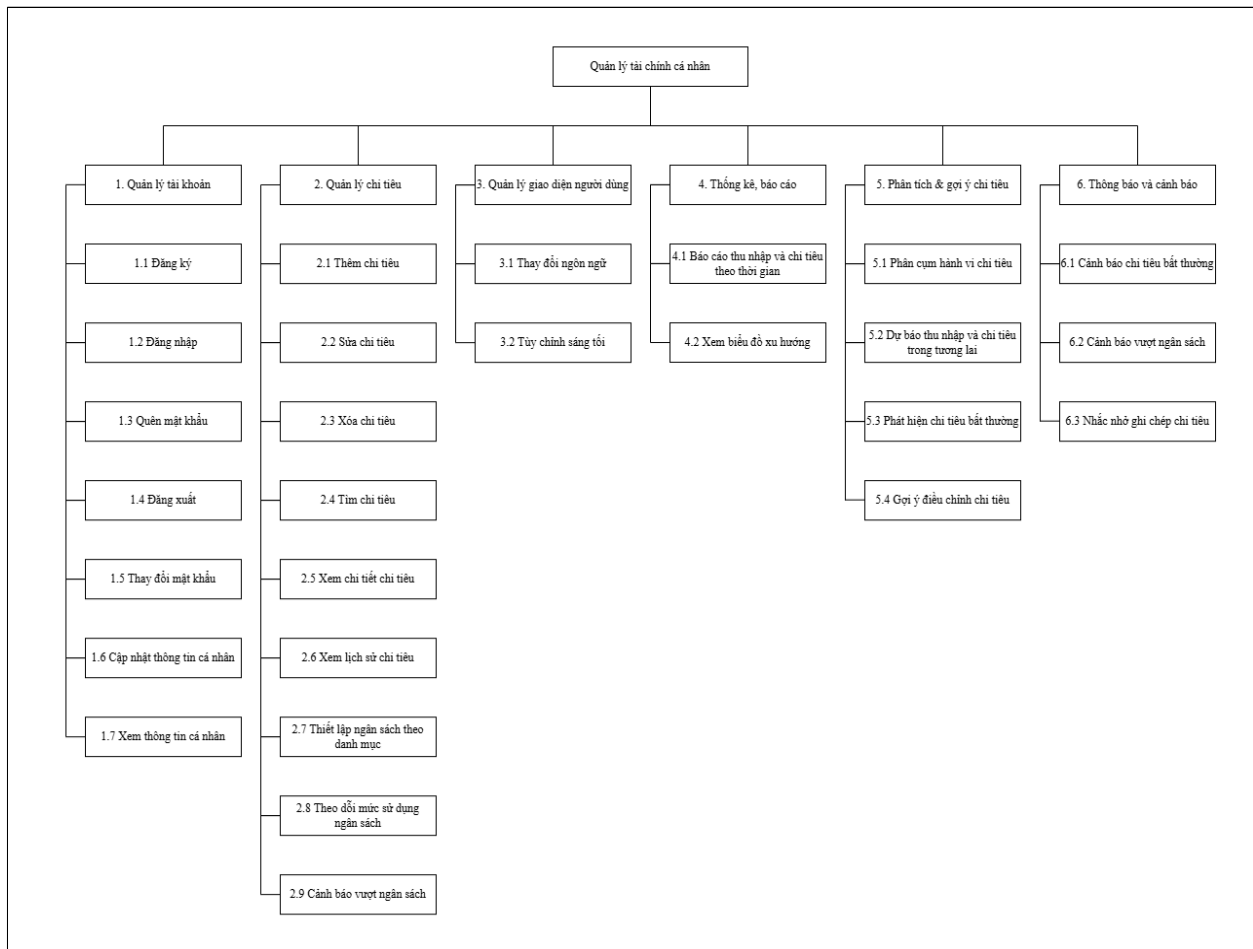
- Nắm vững được công nghệ phát triển của công nghệ.
- Loại bỏ những yêu cầu mập mờ, không rõ ràng.
- Có lượng tài nguyên phát triển phong phú và trình độ chuyên môn, kỹ thuật cao.
- Có thể phù hợp cho dự án nhỏ, ngắn hạn.

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.

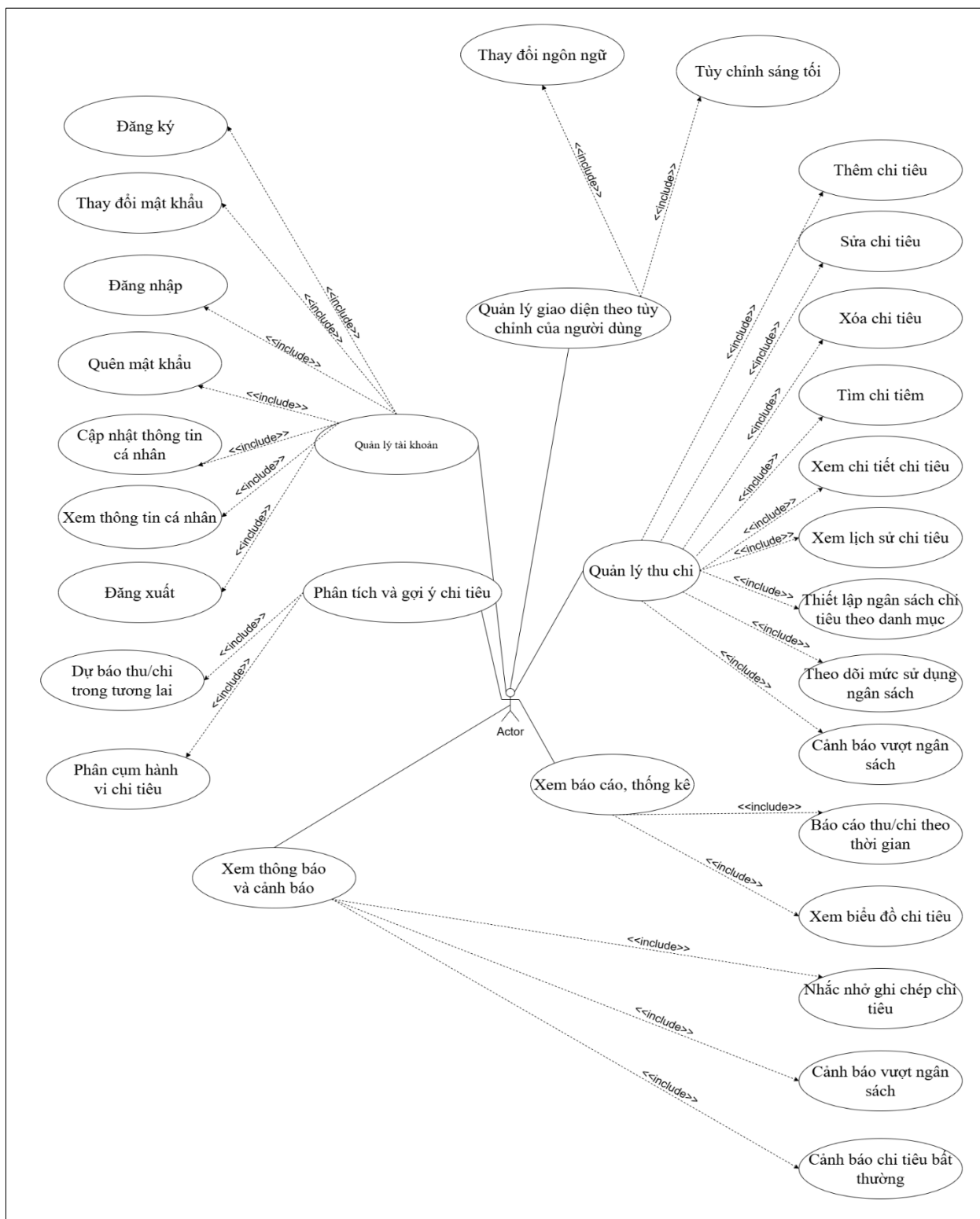
3.2 Thiết kế chức năng của hệ thống

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



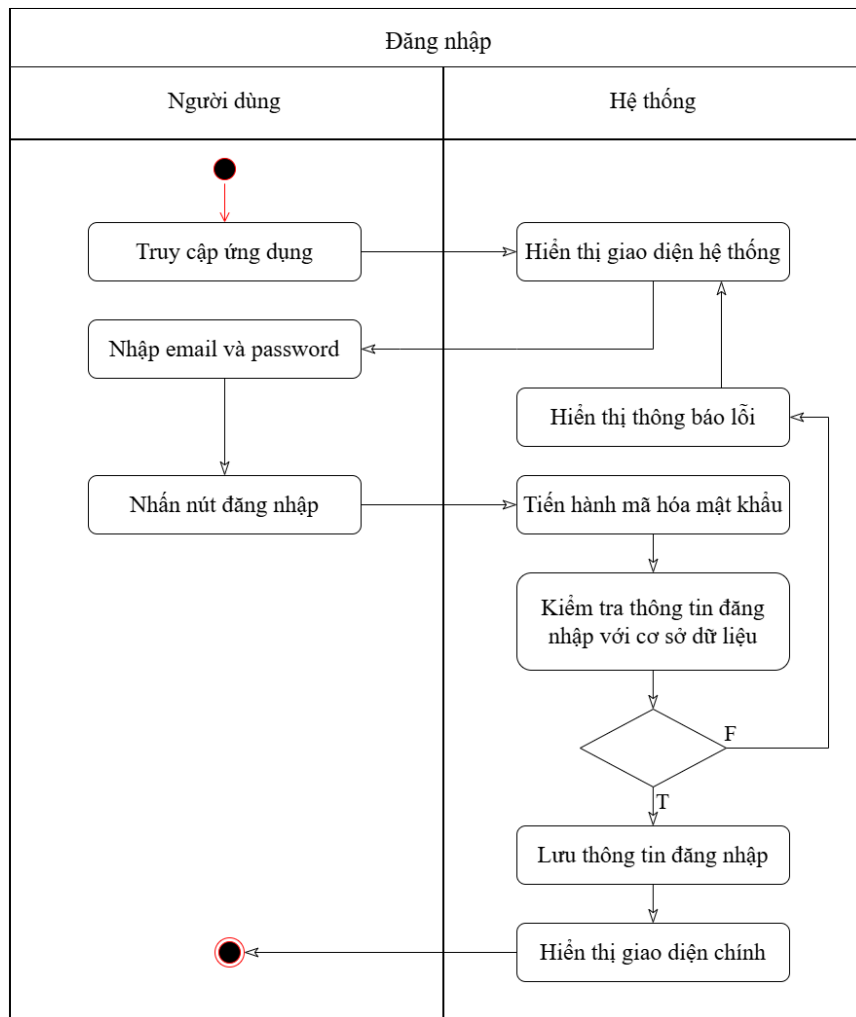
Hình 3.2 Sơ đồ Use case

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

Bảng 3.1 Đặc tả use case đăng nhập



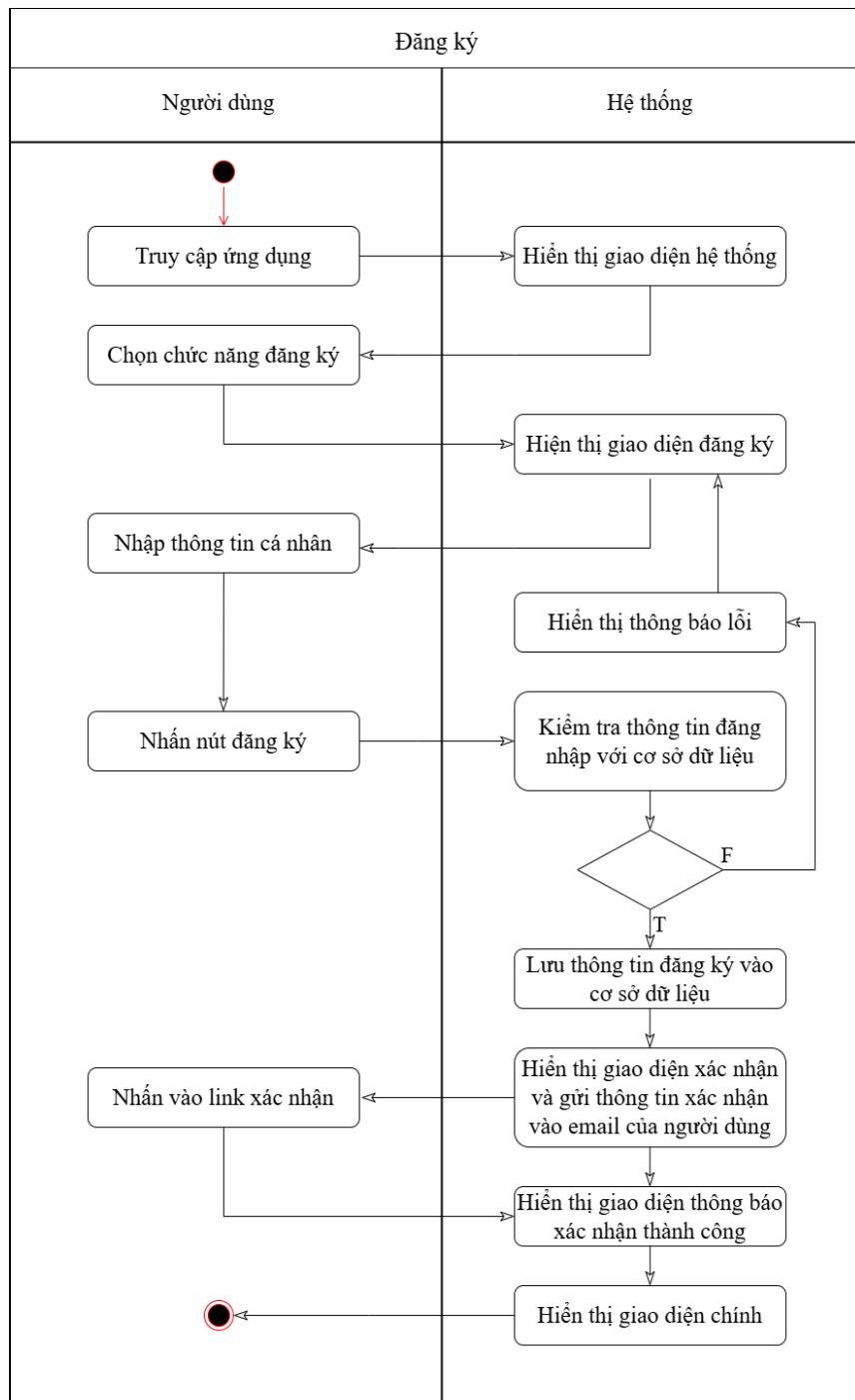
Hình 3.3 Sơ đồ hoạt động đăng nhập

3.2.3.2 Use case đăng ký

Tên use case	Đăng ký 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	Biểu mẫu đăng ký được hiển thị đầy đủ. Thông tin người dùng đã nhập không được lưu trữ khi đăng ký thất bại. Hệ thống không tạo tài khoản khi dữ liệu không hợp lệ.
Đảm bảo thành công	Tài khoản mới được tạo thành công và lưu trữ trong hệ thống. Email xác nhận được gửi đến người dùng.

	Tài khoản được kích hoạt sau khi xác minh email thành công.
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 ở 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 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



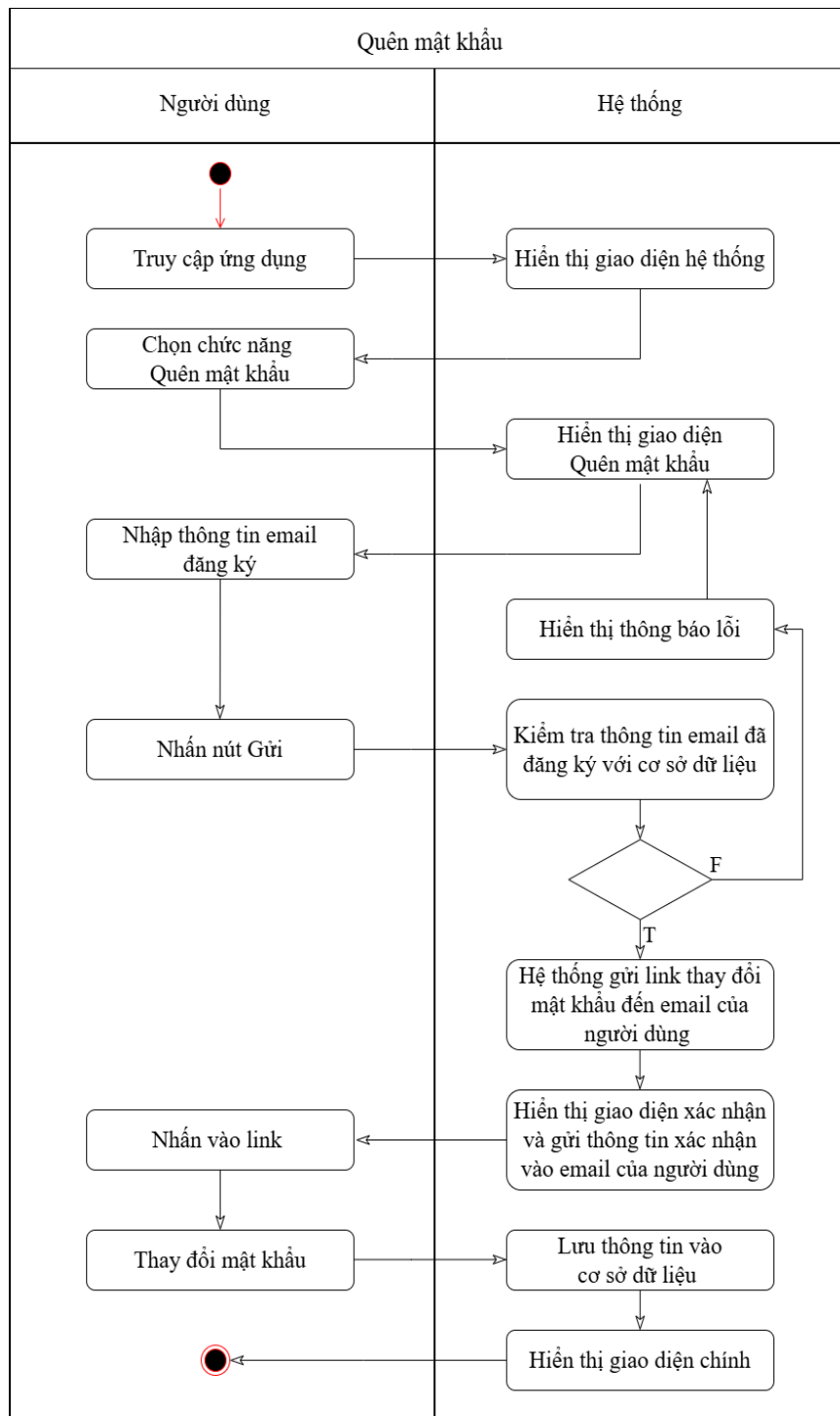
Hình 3.4 Sơ đồ hoạt động đăng ký

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. - 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	Mật khẩu mới được mã hóa trước khi lưu trữ. Liên kết có hiệu lực trong một khoảng thời gian nhất định nhằm đảm bảo an toàn thông tin.

Bảng 3.3 Đặc tả use case quên mật khẩu



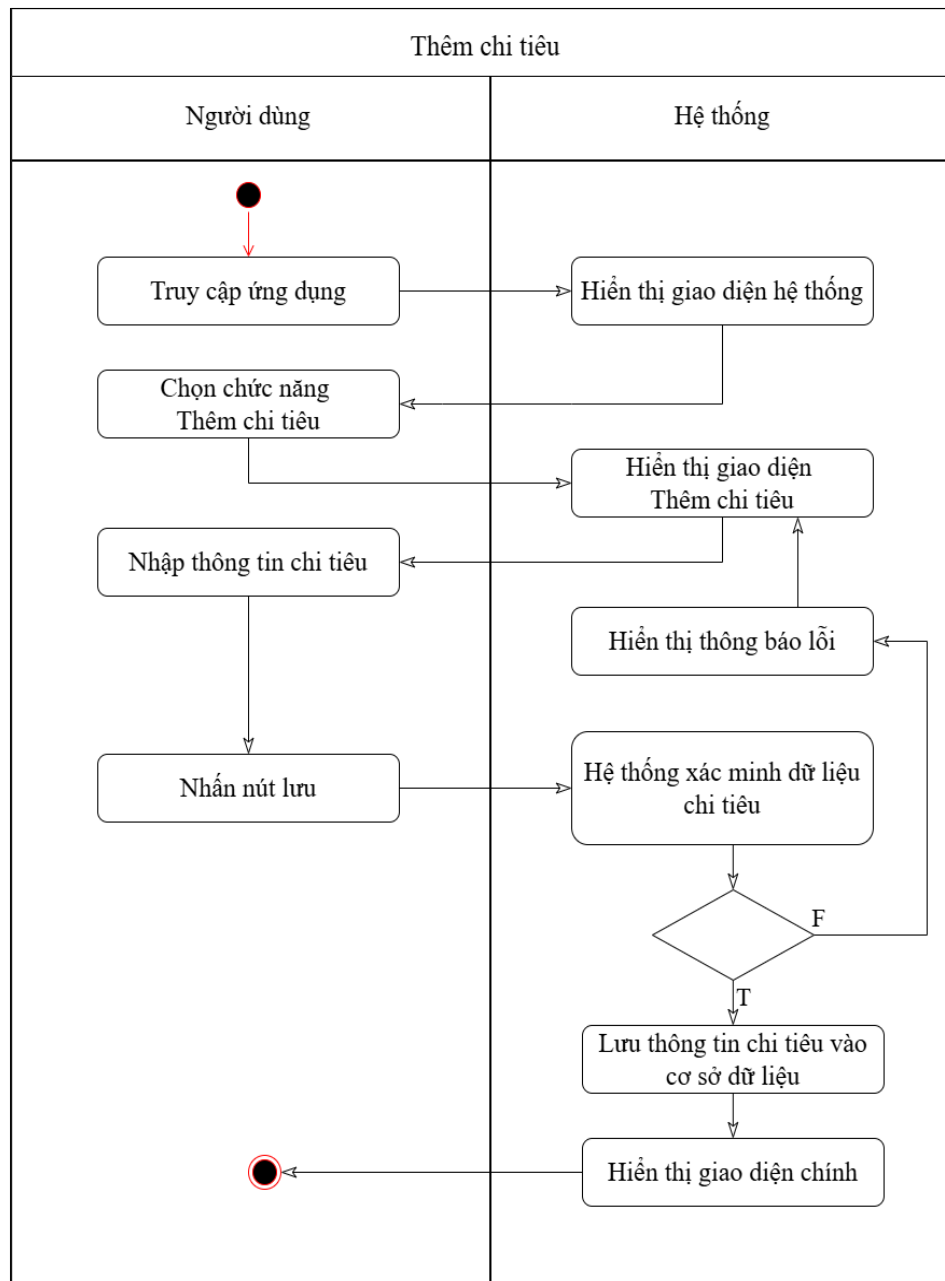
Hình 3.5 Sơ đồ hoạt động quên mật khẩu

3.2.3.4 Use case 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	Các khoản chi tiêu được gắn với tài khoản người dùng đang đăng nhập. Hệ thống có thể sử dụng dữ liệu chi tiêu để phục vụ thống kê và phân tích tài chính.

Bảng 3.4 Đặc tả use case thêm chi tiêu



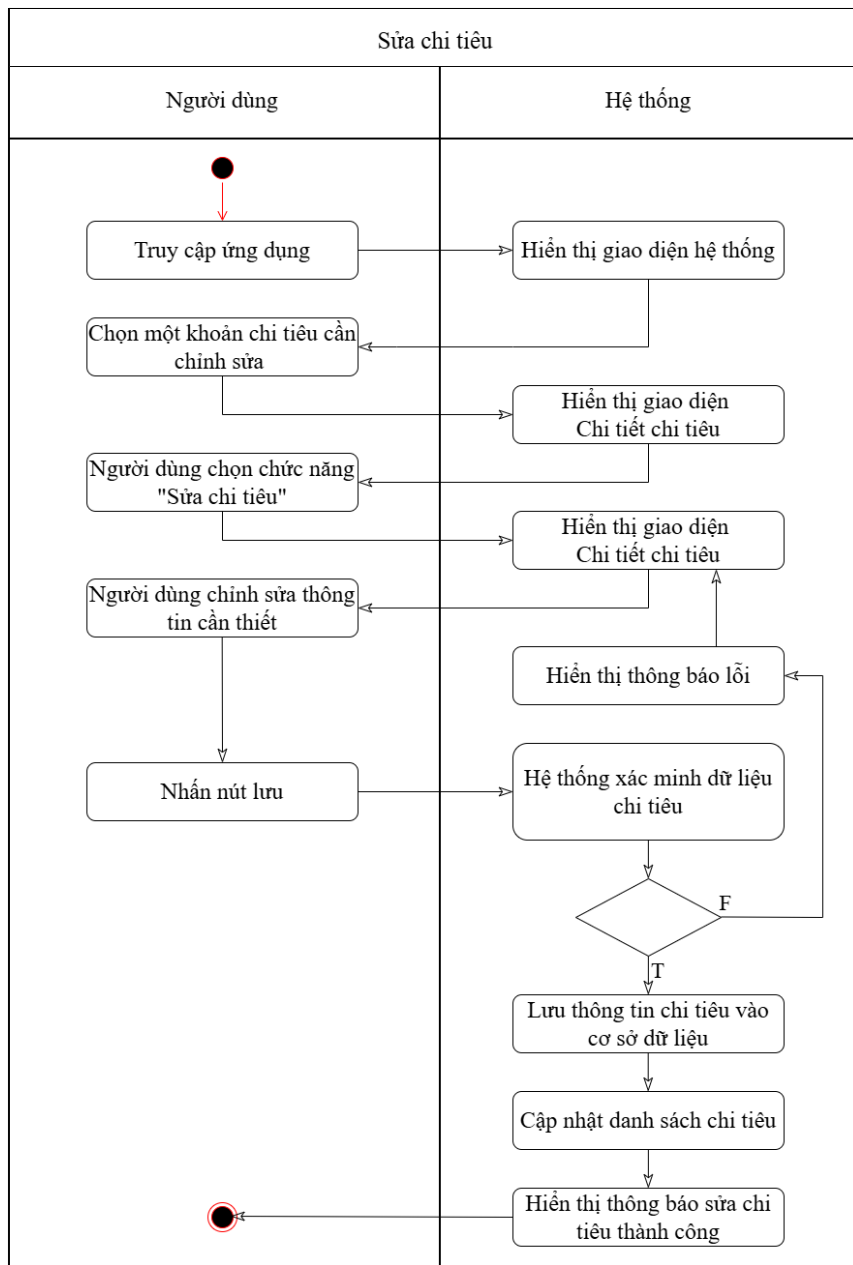
Hình 3.6 Sơ đồ hoạt động thêm chi tiêu

3.2.3.5 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 tài chính chính xác hơn.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có ít nhất một khoản chi tiêu được lưu trong hệ thống.

	Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Danh sách các khoản chi tiêu được hiển thị đầy đủ. Không có dữ liệu chi tiêu nào bị thay đổi khi xảy ra lỗi hoặc thông tin không hợp lệ.
Đảm bảo thành công	Thông tin chi tiêu được cập nhật thành công trong hệ thống. Danh sách chi tiêu được hiển thị đúng với dữ liệu mới.
Kích hoạt	Người dùng chọn chức năng "Sửa" cho một khoản chi tiêu cụ thể trong danh sách.
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	Chỉ cho phép chỉnh sửa các khoản chi tiêu thuộc về tài khoản người dùng đang đăng nhập. Dữ liệu sau khi chỉnh sửa được sử dụng cho các chức năng thống kê và phân tích tài chính.

Bảng 3.5 Đặc tả use case sửa chi tiêu



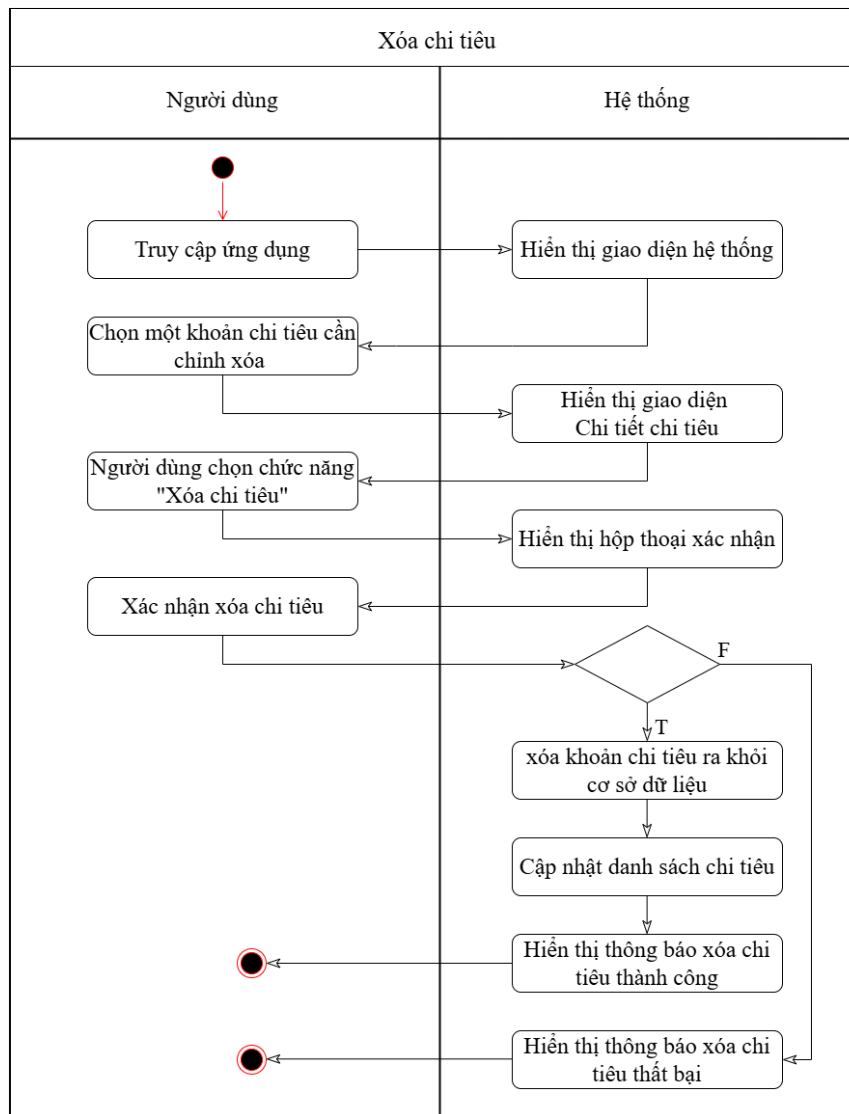
Hình 3.7 Sơ đồ hoạt động sửa chi tiêu

3.2.3.6 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	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có ít nhất một khoản chi tiêu được lưu trong hệ thống.

	Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Danh sách các khoản chi tiêu được hiển thị đầy đủ. Không có dữ liệu nào bị xóa khi xảy ra lỗi hệ thống.
Đảm bảo thành công	Khoản chi tiêu được xóa hoàn toàn khỏi hệ thống. Danh sách chi tiêu của người dùng được cập nhật chính xác.
Kích hoạt	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.
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> <p>8b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	Hệ thống yêu cầu người dùng xác nhận trước khi xóa để tránh thao tác nhầm. Chỉ cho phép xóa các khoản chi tiêu thuộc tài khoản người dùng đang đăng nhập.

Bảng 3.6 Đặc tả use case xóa chi tiêu



Hình 3.8 Sơ đồ hoạt động xóa chi tiêu

3.2.3.7 Use case phân tích và báo cáo tài chính

Tên use case	Phân tích và báo cáo tài chính.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép hệ thống tổng hợp và phân tích dữ liệu thu nhập và chi tiêu của người dùng theo nhiều mốc thời gian khác nhau, từ đó trình bày kết quả dưới dạng báo cáo thống kê và biểu đồ trực quan. Chức năng này giúp người dùng đánh giá tình hình tài chính cá nhân, theo dõi xu hướng thu chi và hỗ trợ ra quyết định quản lý tài chính hiệu quả.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng đã có dữ liệu thu nhập và chi tiêu được lưu trong hệ thống. Hệ thống phân tích dữ liệu hoạt động bình thường. Thiết bị có kết nối Internet.

Đảm bảo tối thiểu	Hệ thống hiển thị báo cáo và thống kê dựa trên dữ liệu hiện có. Không có dữ liệu tài chính nào của người dùng bị thay đổi.
Đảm bảo thành công	Người dùng xem được các báo cáo tổng hợp thu chi theo ngày, tháng hoặc năm. Hệ thống hiển thị biểu đồ trực quan giúp người dùng dễ dàng so sánh và đánh giá tình hình tài chính.
Kích hoạt	Người dùng truy cập chức năng “Phân tích và báo cáo tài chính” 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 “Phân tích và báo cáo tài chính”. - 3. Người dùng lựa chọn khoảng thời gian phân tích (ngày, tháng, năm). - 4. Hệ thống truy xuất dữ liệu thu nhập và chi tiêu tương ứng của người dùng. - 5. Hệ thống tổng hợp dữ liệu và thực hiện phân tích thống kê. - 6. Hệ thống tạo các báo cáo dưới dạng bảng số liệu và biểu đồ (cột, tròn, đường). - 7. Hệ thống hiển thị kết quả phân tích và báo cáo cho người dùng. - 8. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không có dữ liệu trong khoảng thời gian được chọn</p> <p>4a. Không tồn tại dữ liệu thu chi trong khoảng thời gian người dùng yêu cầu.</p> <p>4b. Hệ thống hiển thị thông báo “Không có dữ liệu để phân tích”.</p> <p>E2 – Lỗi xử lý dữ liệu</p> <p>5a. Xảy ra lỗi trong quá trình tổng hợp hoặc phân tích dữ liệu.</p> <p>5b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thử lại sau.</p>
Ghi chú	<p>Kết quả phân cụm mang tính hỗ trợ phân tích và tham khảo.</p> <p>Độ chính xác của việc phân cụm phụ thuộc vào số lượng và chất lượng dữ liệu chi tiêu lịch sử của người dùng.</p> <p>Kết quả phân cụm được sử dụng làm cơ sở cho các chức năng AI nâng cao khác như gợi ý chi tiêu và dự đoán tài chính.</p>

Bảng 3.7 Use case phân tích và báo cáo tài chính

3.2.3.8 Use case phân cụm hành vi chi tiêu

Tên use case	Phân cụm hành vi chi tiêu.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép hệ thống phân tích dữ liệu chi tiêu của người dùng và áp dụng các thuật toán học máy không giám sát (phân cụm) nhằm nhóm các giao dịch và hành vi chi tiêu có đặc điểm tương đồng. Qua đó, hệ thống giúp người dùng hiểu rõ hơn thói quen chi tiêu của bản thân, hỗ trợ cho việc quản lý và điều chỉnh tài chính cá nhân hiệu quả hơn.
Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đã có dữ liệu chi tiêu được lưu trong hệ thống.</p> <p>Hệ thống AI đã được huấn luyện và sẵn sàng hoạt động.</p>

	Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Hệ thống thực hiện phân tích và phân nhóm dữ liệu chỉ tiêu hiện có của người dùng. Không có dữ liệu chỉ tiêu nào của người dùng bị thay đổi hoặc mất mát.
Đảm bảo thành công	Người dùng nhận được kết quả phân cụm hành vi chỉ tiêu một cách rõ ràng. Các nhóm chỉ tiêu phản ánh đúng đặc điểm và thói quen tài chính của người dùng. Kết quả phân cụm được trình bày dễ hiểu và trực quan.
Kích hoạt	Người dùng truy cập chức năng “Phân cụm hành vi chỉ 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 “Phân cụm hành vi chỉ tiêu”. - 3. Hệ thống truy xuất dữ liệu chỉ tiêu lịch sử của người dùng. - 4. Hệ thống trích xuất các đặc trưng liên quan đến hành vi chỉ tiêu. - 5. Hệ thống áp dụng thuật toán phân cụm để nhóm các hành vi chỉ tiêu tương đồng. - 6. Hệ thống tổng hợp và diễn giải kết quả phân cụm. - 7. Hệ thống hiển thị kết quả phân cụm hành vi chỉ tiêu cho người dùng. - 8. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không đủ dữ liệu để phân tích</p> <p>3a. Dữ liệu chỉ 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>Kết quả phân cụm mang tính hỗ trợ phân tích và tham khảo.</p> <p>Độ chính xác của việc phân cụm phụ thuộc vào số lượng và chất lượng dữ liệu chỉ tiêu lịch sử của người dùng.</p> <p>Kết quả phân cụm được sử dụng làm cơ sở cho các chức năng AI nâng cao khác như gợi ý chỉ tiêu và dự đoán tài chính.</p>

Bảng 3.8 Use case phân cụm hành vi chỉ tiêu

3.2.3.9 Use case phát hiện chỉ tiêu bất thường

Tên use case	Phát hiện chỉ tiêu bất thường.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép hệ thống phân tích dữ liệu giao dịch thu chi của người dùng bằng các thuật toán học máy không giám sát nhằm phát hiện các khoản chi tiêu có dấu hiệu bất thường (chỉ tiêu đột biến, sai lệch so với thói quen, trùng lặp hoặc phát sinh không hợp lý theo thời gian), từ đó hỗ trợ người dùng kịp thời kiểm soát và điều chỉnh hành vi chi tiêu.
Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đã có lịch sử chi tiêu đủ lớn để phân tích.</p>

	Hệ thống AI phát hiện bất thường đã được kích hoạt. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Hệ thống hiển thị danh sách các giao dịch có dấu hiệu bất thường dựa trên dữ liệu hiện có. Không có dữ liệu giao dịch nào của người dùng bị chỉnh sửa hoặc xóa.
Đảm bảo thành công	Người dùng nhận được danh sách các giao dịch bất thường kèm theo lý do và mức độ rủi ro (thấp, trung bình, cao). Các cảnh báo được hiển thị rõ ràng, dễ hiểu và có giá trị hỗ trợ ra quyết định tài chính.
Kích hoạt	Người dùng truy cập chức năng “Phát hiện chỉ tiêu bất thường” 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 “Phát hiện chỉ tiêu bất thường”. - 3. Hệ thống truy xuất dữ liệu chỉ tiêu lịch sử của người dùng. - 4. Hệ thống sử dụng mô hình AI để phân tích hành vi chi tiêu Hệ thống tiền xử lý dữ liệu và trích xuất các đặc trưng (số tiền, danh mục, thời gian, tần suất giao dịch...). - 5. Hệ thống sử dụng mô hình học máy không giám sát (Isolation Forest) kết hợp với các luật logic để phát hiện giao dịch bất thường. - 6. Hệ thống xác định nguyên nhân, mức độ nghiêm trọng của từng giao dịch bất thường. - 7. Hệ thống hiển thị danh sách giao dịch bất thường và các cảnh báo liên quan. - 8. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Không phát hiện chỉ tiêu bất thường</p> <p>5a. Không có khoản chi tiêu nào được xác định là bất thường.</p> <p>5b. Hệ thống hiển thị thông báo “Không phát hiện chỉ tiêu bất thường”.</p> <p>E2 – Lỗi phân tích dữ liệu</p> <p>3a. Dữ liệu chỉ tiêu bị thiếu hoặc không đồng bộ.</p> <p>3b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	<p>Kết quả phát hiện bất thường chỉ mang tính hỗ trợ và tham khảo, không thay thế cho quyết định tài chính của người dùng.</p> <p>Độ chính xác của kết quả phụ thuộc vào số lượng và chất lượng dữ liệu giao dịch lịch sử.</p> <p>Hệ thống kết hợp cả mô hình học máy và luật nghiệp vụ để nhằm mục đích tăng độ tin cậy của kết quả phát hiện.</p>

Bảng 3.9 Use case phát hiện chỉ tiêu bất thường

3.2.3.10 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 phân tích dữ liệu thu chi lịch sử của người dùng theo chuỗi thời gian và áp dụng các mô hình trí tuệ nhân tạo để dự đoán xu hướng thu nhập và chi tiêu

	trong các giai đoạn tiếp theo. Chức năng này giúp người dùng chủ động lập kế hoạch tài chính và đưa ra quyết định chi tiêu hợp lý.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống. Người dùng có dữ liệu chi tiêu lịch sử đầy đủ. Mô hình AI dự đoán đã được huấn luyện. Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Hệ thống hiển thị kết quả dự đoán dựa trên dữ liệu hiện có của người dùng. Không có dữ liệu giao dịch nào của người dùng bị thay đổi.
Đảm bảo thành công	Người dùng nhận được kết quả dự đoán thu nhập và chi tiêu cho các ngày tiếp theo kèm theo mức độ tin cậy. Hệ thống cung cấp tổng hợp xu hướng và khuyến nghị tài chính dựa trên kết quả dự đoán.
Kích hoạt	Người dùng truy cập 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 trong tương lai”. - 3. Hệ thống truy xuất dữ liệu thu chi lịch sử của người dùng. - 4. Hệ thống tiền xử lý dữ liệu theo chuỗi thời gian. - 5. Hệ thống lựa chọn chiến lược dự đoán phù hợp dựa trên đặc điểm dữ liệu. - 6. Hệ thống thực hiện dự đoán thu nhập và chi tiêu cho các ngày tiếp theo. - 7. Hệ thống phân tích xu hướng thu chi và sinh ra nhận xét, khuyến nghị tài chính. - 8. Hệ thống hiển thị kết quả dự đoán, biểu đồ xu hướng và thông tin cho người dùng. - 9. 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 phụ thuộc vào hành vi chi tiêu trong quá khứ của người dùng.</p> <p>Độ chính xác của dự đoán sẽ được cải thiện khi dữ liệu lịch sử đầy đủ và ổn định hơn.</p> <p>Hệ thống có thể tự động lựa chọn mô hình phù hợp nhằm đảm bảo kết quả dự đoán liên tục và ổn định.</p>

Bảng 3.10 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 3.11 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 3.12 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 3.13 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 điện toán đám mây, không yêu cầu triển khai máy chủ vật lý riêng tại chỗ. Toàn bộ quá trình lưu trữ dữ liệu, xử lý nghiệp vụ và phân tích thông minh được thực hiện thông qua các dịch vụ đám mây và máy chủ ảo. Cấu trúc phần cứng của hệ thống bao gồm các thành phần chính sau:

- Thiết bị người dùng: Thiết bị người dùng bao gồm các điện thoại thông minh sử dụng hệ điều hành Android và iOS. Người dùng tương tác trực tiếp với hệ thống thông qua ứng dụng di động được phát triển bằng Flutter. Thiết bị di động đảm nhiệm vai trò hiển thị giao diện, thu thập dữ liệu chi tiêu, tiếp nhận thao tác người dùng và gửi yêu cầu đến hệ thống backend để xử lý.

- Máy chủ Firebase: Firebase đóng vai trò là nền tảng máy chủ trung tâm cho việc xác thực người dùng và lưu trữ dữ liệu. Hệ thống sử dụng Firebase Authentication để quản lý đăng nhập (Email/Password, Google) và Cloud Firestore để lưu trữ dữ liệu giao dịch tài chính, thông tin người dùng cũng như các cấu hình của ứng dụng. Việc sử dụng Firebase giúp hệ thống hoạt động ổn định, có khả năng mở rộng linh hoạt và hỗ trợ đồng bộ dữ liệu theo thời gian thực giữa các thiết bị.

- Máy chủ xử lý AI (Backend Python): Bên cạnh Firebase, hệ thống còn sử dụng một máy chủ backend độc lập được triển khai bằng Python, chạy trên môi trường máy chủ ảo hoặc dịch vụ đám mây. Máy chủ này chịu trách nhiệm xử lý các tác vụ phân tích dữ liệu và AI, bao gồm dự báo chi tiêu, phân cụm hành vi người dùng và phát hiện giao dịch bất thường. Máy chủ này giao tiếp với ứng dụng Flutter thông qua các REST API, đóng vai trò là trung tâm xử lý logic AI của toàn hệ thống.

4.1.2 Kiến trúc phần mềm

Hệ thống được thiết kế theo mô hình client – server mở rộng, trong đó Flutter đảm nhiệm vai trò phía client, Firebase đóng vai trò backend dịch vụ và backend AI bằng Python đảm nhiệm xử lý phân tích dữ liệu thông minh. Kiến trúc này giúp giảm thiểu công việc

triển khai và quản lý máy chủ truyền thống, đồng thời tối ưu hóa khả năng đồng bộ dữ liệu và mở rộng các chức năng trí tuệ nhân tạo.

- 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.
- + Ứng dụng Flutter đồng thời đóng vai trò gửi dữ liệu giao dịch đến backend để phục vụ cho quá trình phân tích và dự báo bằng AI.

- 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.

- Backend AI (Python):

- + Hệ thống sử dụng một backend độc lập được xây dựng bằng Python, triển khai thông qua FastAPI và chạy trên Uvicorn.
- + Backend AI chịu trách nhiệm xử lý các nghiệp vụ phân tích dữ liệu tài chính, bao gồm dự báo chi tiêu và thu nhập, phân cụm hành vi người dùng và phát hiện các giao dịch bất thường.
- + Backend AI giao tiếp với ứng dụng Flutter thông qua các REST API.

- API tích hợp:

- + Ứng dụng Flutter sử dụng lớp API Service để gọi các dịch vụ từ backend AI Python và các dịch vụ bên ngoài.
- + Thông qua API, dữ liệu từ Firebase được gửi đến backend AI để phân tích, sau đó kết quả được trả về cho ứng dụng nhằm hiển thị các báo cáo, cảnh báo và gợi ý chi tiêu thông minh cho người dùng.

4.1.3 Sơ đồ kiến trúc phần mềm hệ thống

Sơ đồ kiến trúc phần mềm của hệ thống được xây dựng theo mô hình Client – Backend dịch vụ – Backend AI, trong đó các thành phần giao tiếp với nhau thông qua các giao thức

mạng và REST API. Kiến trúc này giúp hệ thống hoạt động linh hoạt, dễ mở rộng và phù hợp với các ứng dụng có tích hợp AI.

- Client (Ứng dụng Flutter): Ứng dụng di động được phát triển bằng Flutter đóng vai trò là phía client của hệ thống. Người dùng thực hiện các thao tác như đăng nhập, nhập dữ liệu thu chi, xem báo cáo và nhận gợi ý chi tiêu trực tiếp trên ứng dụng. Dữ liệu do người dùng nhập được gửi lên Firebase để lưu trữ, đồng thời được trích xuất và gửi đến backend AI thông qua các API để phục vụ cho việc phân tích và dự báo.

- Backend dịch vụ (Firebase): Firebase đóng vai trò là trung tâm lưu trữ và quản lý dữ liệu của hệ thống. Firebase Authentication chịu trách nhiệm xác thực và quản lý tài khoản người dùng. Cloud Firestore đảm nhiệm việc lưu trữ dữ liệu giao dịch tài chính, thông tin người dùng và các cấu hình liên quan. Firebase hỗ trợ đồng bộ dữ liệu theo thời gian thực, giúp dữ liệu giữa các thiết bị luôn được cập nhật kịp thời.

- Backend AI (Python – FastAPI): Backend AI được xây dựng bằng Python, sử dụng FastAPI và chạy trên Uvicorn, hoạt động như một máy chủ xử lý thông minh độc lập với phía client. Backend này nhận dữ liệu tài chính từ ứng dụng Flutter (hoặc trích xuất gián tiếp từ Firebase), sau đó áp dụng các mô hình học máy như LSTM, K-Means và Isolation Forest để phân tích hành vi chi tiêu, dự báo xu hướng tài chính và phát hiện các giao dịch bất thường. Kết quả phân tích được trả về cho ứng dụng Flutter thông qua các REST API để hiển thị cho người dùng.

- Luồng xử lý dữ liệu:

- + 1. Người dùng thao tác trên ứng dụng Flutter và nhập dữ liệu tài chính.
- + 2. Dữ liệu được lưu trữ và đồng bộ lên Firebase.
- + 3. Ứng dụng Flutter gửi yêu cầu phân tích dữ liệu đến backend AI thông qua API.
- + 4. Backend AI xử lý dữ liệu bằng mô hình học máy và trả về kết quả phân tích.
- + 5. Ứng dụng Flutter nhận kết quả và hiển thị báo cáo, cảnh báo hoặc gợi ý chi tiêu thông minh cho người dùng.

Kiến trúc này giúp tách biệt rõ ràng giữa phần giao diện, phần lưu trữ dữ liệu và phần xử lý trí tuệ nhân tạo, từ đó nâng cao khả năng bảo trì, mở rộng hệ thống cũng như đảm bảo hiệu năng và tính ổn định trong quá trình vận hành.

4.1.4 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 dựa trên tần suất, giá trị và loại hình giao dịch, 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 chi tiêu 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 thu nhập và chi tiêu trong các khoảng thời gian tiếp theo, giúp người dùng chủ động hơn trong việc lập kế hoạch tài chính và kiểm soát ngân sách cá nhân.

- Phát hiện chi tiêu bất thường: Hệ thống áp dụng thuật toán Isolation Forest để phát hiện các giao dịch chi tiêu bất thường so với hành vi tài chính thông thường của người dùng. Các giao dịch có giá trị đột biến, tần suất bất thường hoặc không phù hợp với thói quen chi tiêu trước đó sẽ được hệ thống đánh dấu và cảnh báo. Chức năng này giúp người

dùng kịp thời nhận biết rủi ro tài chính, hạn chế các khoản chi không hợp lý hoặc các giao dịch có dấu hiệu bất thường.

- Gợi ý và hỗ trợ ra quyết định: Dựa trên kết quả phân tích hành vi chi tiêu, dự báo thu chi và phát hiện chi tiêu bất thường, hệ thống đưa ra các gợi ý chi tiêu phù hợp với từng người dùng. Các gợi ý này bao gồm đề xuất điều chỉnh ngân sách, kiểm soát các khoản chi vượt mức và định hướng thói quen tài chính hợp lý, giúp người dùng từng bước hình thành phương pháp quản lý tài chính cá nhân khoa học và bền vững.

4.2 Thiết kế giao diện

Thiết kế giao diện người dùng là một trong những yếu tố quan trọng, có ảnh hưởng trực tiếp đến trải nghiệm sử dụng và mức độ gắn bó lâu dài của người dùng đối với ứng dụng quản lý tài chính cá nhân. Một giao diện được thiết kế hợp lý, trực quan và dễ sử dụng không chỉ giúp người dùng thao tác nhanh chóng, hạn chế sai sót trong quá trình nhập liệu, mà còn hỗ trợ người dùng dễ dàng tiếp cận và theo dõi các thông tin tài chính quan trọng. Thông qua cách trình bày rõ ràng, khoa học và thân thiện, người dùng có thể hiểu rõ hơn về tình hình thu nhập, chi tiêu cũng như thói quen tài chính của bản thân. Từ đó, ứng dụng không chỉ đóng vai trò là công cụ ghi chép thu chi, mà còn trở thành trợ lý hỗ trợ người dùng đưa ra các quyết định chi tiêu hợp lý, góp phần nâng cao hiệu quả quản lý tài chính cá nhân trong cuộc sống hằng ngày.

Trong đề tài này, giao diện ứng dụng được thiết kế với định hướng đơn giản, rõ ràng, thân thiện và phù hợp với nhiều nhóm người dùng khác nhau như sinh viên, người đi làm hoặc người chưa có nhiều kiến thức về tài chính. Bên cạnh đó, giao diện còn được xây dựng sao cho có thể mở rộng và tích hợp các chức năng phân tích thông minh dựa trên trí tuệ nhân tạo mà không gây rối mắt hoặc phức tạp trong quá trình sử dụng.

4.2.1 Nguyên tắc thiết kế giao diện

Giao diện của ứng dụng được thiết kế dựa trên một số nguyên tắc cơ bản nhằm đảm bảo tính hiệu quả và tính thân thiện đối với người dùng:

- Đơn giản và dễ hiểu: Các thành phần giao diện như nút bấm, biểu tượng, biểu đồ và văn bản được bố trí rõ ràng, hạn chế tối đa các chi tiết không cần thiết. Người dùng có thể dễ dàng nhận biết chức năng của từng màn hình mà không cần nhiều hướng dẫn.

- Tính nhất quán: Màu sắc, kiểu chữ, kích thước biểu tượng và cách bố trí các thành phần được sử dụng đồng nhất xuyên suốt toàn bộ ứng dụng. Điều này giúp người dùng nhanh chóng làm quen và không bị bối rối khi chuyển đổi giữa các chức năng khác nhau.

- Hỗ trợ trực quan bằng biểu đồ: Các dữ liệu tài chính được thể hiện thông qua biểu đồ tròn, biểu đồ cột và biểu đồ đường, giúp người dùng dễ dàng quan sát sự thay đổi của thu nhập và chi tiêu theo thời gian. Việc trực quan hóa dữ liệu giúp giảm tải thông tin và tăng khả năng tiếp cận đối với người dùng không chuyên.

- Tối ưu cho thiết bị di động: Giao diện được thiết kế phù hợp với màn hình điện thoại thông minh, đảm bảo thao tác thuận tiện bằng một tay và hiển thị tốt trên nhiều kích thước màn hình khác nhau.

- Hỗ trợ đa ngôn ngữ và chế độ giao diện: Ứng dụng cho phép người dùng thay đổi ngôn ngữ và chế độ hiển thị (sáng/tối), góp phần nâng cao trải nghiệm sử dụng và đáp ứng nhu cầu cá nhân hóa.

4.2.2 Thiết kế các màn hình chức năng chính

Dựa trên các nguyên tắc thiết kế đã đề ra, giao diện ứng dụng được chia thành các màn hình chức năng chính như sau:

- Màn hình đăng nhập và đăng ký: Đây là màn hình đầu tiên người dùng tiếp cận khi sử dụng ứng dụng. Giao diện được thiết kế đơn giản, tập trung vào các trường thông tin cần thiết như email, mật khẩu và các phương thức đăng nhập khác (Google, Apple). Các thông báo lỗi và hướng dẫn được hiển thị rõ ràng nhằm hỗ trợ người dùng trong quá trình xác thực tài khoản.

- Màn hình trang chủ: Màn hình trang chủ cung cấp cái nhìn tổng quan về tình hình tài chính của người dùng trong khoảng thời gian hiện tại. Tại đây, người dùng có thể theo dõi nhanh tổng thu nhập, tổng chi tiêu, số dư còn lại cũng như các khoản chi nổi bật. Các thông tin quan trọng được ưu tiên hiển thị ở vị trí dễ quan sát.

- Màn hình quản lý thu chi: Màn hình này cho phép người dùng thêm, chỉnh sửa và xóa các khoản thu nhập hoặc chi tiêu. Các giao dịch được phân loại theo danh mục như ăn uống, giải trí, học tập, nợ vay, ... giúp việc theo dõi và quản lý tài chính trở nên rõ ràng hơn.

- Màn hình phân tích và báo cáo: Tại đây, hệ thống hiển thị các báo cáo tài chính dưới dạng biểu đồ theo ngày, tháng và năm. Người dùng có thể dễ dàng nhận biết xu hướng chi tiêu, so sánh các khoản thu chi và đánh giá hiệu quả quản lý tài chính của bản thân.

- Màn hình gợi ý và cảnh báo thông minh: Dựa trên kết quả phân tích hành vi chi tiêu, dự báo và phát hiện bất thường từ hệ thống AI, ứng dụng đưa ra các gợi ý phù hợp cũng như cảnh báo khi người dùng có dấu hiệu chi tiêu vượt mức hoặc bất thường. Các thông báo này được trình bày ngắn gọn, dễ hiểu và không gây phiền nhiễu.

- Màn hình cài đặt và quản lý cá nhân: Màn hình này cho phép người dùng cập nhật thông tin cá nhân, thay đổi mật khẩu, thiết lập ngân sách, ngôn ngữ, giao diện và các tùy chọn thông báo khác. Việc tập trung các thiết lập vào một khu vực giúp người dùng dễ dàng quản lý và cá nhân hóa ứng dụng theo nhu cầu.

4.3 Một số chức năng chính của hệ thống

Hệ thống quản lý và phân tích tài chính cá nhân được xây dựng với nhiều chức năng nhằm hỗ trợ người dùng theo dõi, kiểm soát và tối ưu hóa việc sử dụng tài chính của mình. Các chức năng chính của hệ thống bao gồm:

- Quản lý thu nhập và chi tiêu: Cho phép người dùng thêm, chỉnh sửa và xóa các khoản thu nhập và chi tiêu. Dữ liệu được phân loại theo danh mục và thời gian, giúp người dùng dễ dàng theo dõi lịch sử giao dịch.

- Thiết lập và quản lý ngân sách: Người dùng có thể thiết lập ngân sách chi tiêu theo từng danh mục. Hệ thống theo dõi mức chi tiêu thực tế so với ngân sách đã đặt ra, đồng thời đưa ra cảnh báo khi người dùng có nguy cơ chi tiêu vượt ngân sách, qua đó hỗ trợ kiểm soát tài chính hiệu quả hơn.

- Thống kê và báo cáo tài chính: Hệ thống cung cấp các báo cáo tổng hợp dưới dạng biểu đồ và thống kê theo ngày, tháng và năm, giúp người dùng đánh giá tình hình tài chính và nhận biết xu hướng thu chi.

- Phân cụm hành vi chi tiêu: Dựa trên dữ liệu thu chi đã được lưu trữ, hệ thống áp dụng các thuật toán học máy như K-Means để phân tích và phân nhóm hành vi chi tiêu của người dùng, từ đó nhận diện các thói quen tài chính đặc trưng.

- Dự báo thu chi trong tương lai: Chức năng dự báo sử dụng mô hình LSTM để phân tích chuỗi dữ liệu tài chính theo thời gian, giúp người dùng dự đoán xu hướng thu nhập và chi tiêu trong các giai đoạn tiếp theo.

- Phát hiện chi tiêu bất thường: Hệ thống sử dụng các phương pháp học máy không giám sát nhằm phát hiện các giao dịch có dấu hiệu bất thường, hỗ trợ người dùng kịp thời điều chỉnh hành vi chi tiêu.

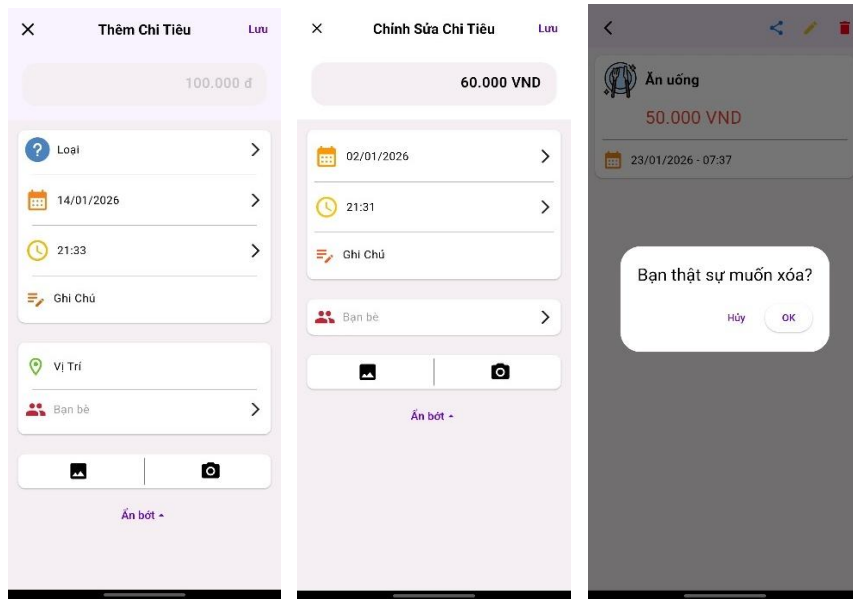
4.3.1 Quản lý thu nhập và chi tiêu

Chức năng Quản lý thu nhập và chi tiêu là chức năng cốt lõi của hệ thống, đóng vai trò nền tảng cho các chức năng phân tích và dự báo tài chính sau này. Chức năng này cho phép người dùng ghi nhận, cập nhật và theo dõi toàn bộ các giao dịch tài chính cá nhân một cách chi tiết và có hệ thống. Cụ thể:

- Thêm khoản thu nhập và chi tiêu: Người dùng có thể dễ dàng nhập các khoản thu hoặc chi mới với đầy đủ thông tin như: số tiền, danh mục (ăn uống, học tập, giải trí, lương, thưởng,...), thời gian giao dịch và ghi chú kèm theo. Việc này giúp dữ liệu tài chính được lưu trữ đầy đủ và chính xác.

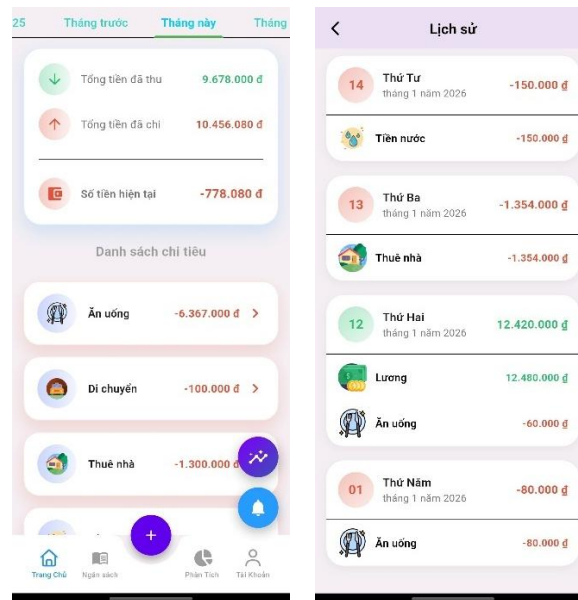
- Chỉnh sửa khoản thu nhập và chi tiêu: Hệ thống cho phép người dùng chỉnh sửa thông tin của các giao dịch đã nhập trong trường hợp nhập sai hoặc cần cập nhật lại nội dung, đảm bảo dữ liệu tài chính luôn phản ánh đúng thực tế.

- Xóa khoản thu nhập và chi tiêu: Người dùng có thể xóa các giao dịch không còn cần thiết hoặc bị nhập nhầm, giúp dữ liệu được quản lý gọn gàng và tránh gây nhiễu cho quá trình phân tích.



Hình 4.1 Chức năng thêm, sửa, xóa giao dịch

- Phân loại theo danh mục và thời gian: Các khoản thu chi được hệ thống tự động phân loại theo danh mục và mốc thời gian (ngày, tháng, năm). Điều này giúp người dùng dễ dàng theo dõi lịch sử giao dịch cũng như so sánh mức thu chi giữa các giai đoạn khác nhau.
- Theo dõi lịch sử giao dịch: Hệ thống cung cấp giao diện hiển thị danh sách các khoản thu chi đã phát sinh, cho phép người dùng xem lại chi tiết từng giao dịch, từ đó nâng cao khả năng kiểm soát và quản lý tài chính cá nhân.



Hình 4.2 Chức năng phân loại và theo dõi lịch sử giao dịch

4.3.2 Thiết lập và quản lý ngân sách

- Thiết lập ngân sách theo danh mục chi tiêu: Hệ thống cho phép người dùng thiết lập hạn mức chi tiêu cho từng danh mục như ăn uống, đi lại, giải trí, mua sắm,... trong một khoảng thời gian xác định (tháng hoặc năm). Việc phân chia ngân sách theo danh mục giúp người dùng kiểm soát chi tiêu chi tiết và khoa học hơn.

- Điều chỉnh ngân sách linh hoạt: Người dùng có thể chỉnh sửa ngân sách đã thiết lập khi có thay đổi về thu nhập hoặc nhu cầu chi tiêu, đảm bảo tính linh hoạt và phù hợp với tình hình tài chính thực tế.

The image displays two side-by-side mobile application screens for budget management. Both screens have a light gray background and a white card-like area for form input.

Left Screen: Thêm ngân sách (Add Budget)

- Header: A back arrow icon and the title "Thêm ngân sách".
- Section: "Thông tin ngân sách" (Budget Information).
- Field 1: "Loại chi tiêu" (Expense Type) with a dropdown menu showing "Chọn loại chi tiêu".
- Field 2: "Giới hạn ngân sách" (Budget Limit) with a dropdown menu showing "\$".
- Button: A blue button labeled "Lưu ngân sách" (Save Budget).

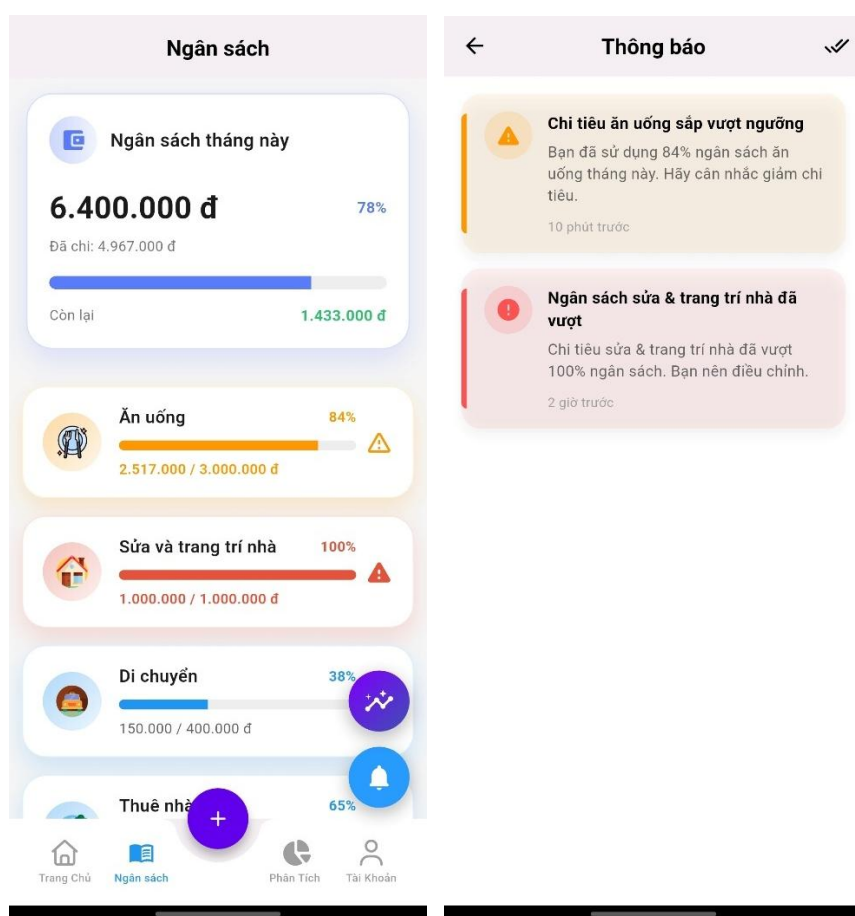
Right Screen: Chỉnh sửa ngân sách (Edit Budget)

- Header: A back arrow icon, the title "Chỉnh sửa ngân sách", and a red delete icon.
- Section: "Thông tin ngân sách" (Budget Information).
- Field 1: "Loại chi tiêu" (Expense Type) with a dropdown menu showing "Ăn uống".
- Field 2: "Giới hạn ngân sách" (Budget Limit) with a dropdown menu showing "\$ 900000".
- Button: A blue button labeled "Cập nhật ngân sách" (Update Budget).

Hình 4.3 Chức năng thiết lập và điều chỉnh ngân sách.

- Theo dõi chi tiêu thực tế so với ngân sách đã đặt: Hệ thống tự động tổng hợp các khoản chi tiêu phát sinh và so sánh với ngân sách tương ứng của từng danh mục, từ đó hiển thị mức độ sử dụng ngân sách theo thời gian thực.

- Cảnh báo khi chi tiêu vượt hoặc sắp vượt ngân sách: Khi mức chi tiêu đạt đến một ngưỡng nhất định (ví dụ 80% hoặc 100% ngân sách), hệ thống sẽ gửi thông báo nhắc nhở, giúp người dùng kịp thời điều chỉnh hành vi chi tiêu.



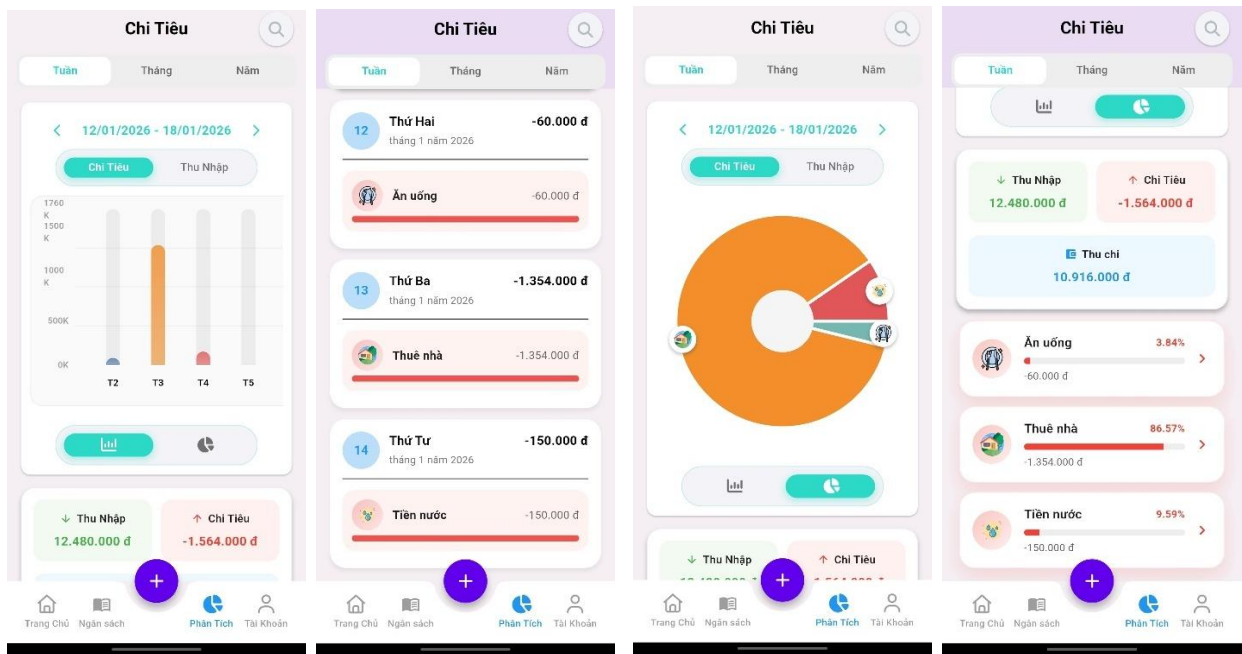
Hình 4.4 Chức năng theo dõi và cảnh báo chi tiêu

4.3.3 Thống kê và báo cáo tài chính

Chức năng Thống kê và báo cáo tài chính giúp người dùng tổng hợp, phân tích và đánh giá toàn bộ tình hình tài chính cá nhân dựa trên các dữ liệu thu nhập và chi tiêu đã được ghi nhận trong hệ thống. Đây là chức năng quan trọng hỗ trợ người dùng nhìn nhận rõ hơn về thói quen chi tiêu cũng như hiệu quả quản lý tài chính theo từng giai đoạn. Cụ thể:

- **Thống kê thu nhập và chi tiêu theo thời gian:** Hệ thống tự động tổng hợp dữ liệu thu nhập và chi tiêu theo các mốc thời gian như ngày, tháng và năm. Điều này giúp người dùng dễ dàng theo dõi sự biến động tài chính và so sánh mức thu chi giữa các giai đoạn chi tiêu khác nhau.

- **Báo cáo tài chính dưới dạng biểu đồ trực quan:** Các số liệu thống kê được trình bày thông qua biểu đồ cột, biểu đồ tròn, giúp người dùng dễ dàng quan sát tỷ lệ thu/chi, cơ cấu chi tiêu theo danh mục và xu hướng tài chính tổng thể.



Hình 4.5 Chức năng thống kê và báo cáo tài chính

4.3.4 Phân cụm hành vi chi tiêu

Dựa trên những dữ liệu thu/chi của người dùng đã được lưu trữ trong hệ thống, chức năng phân tích hành vi chi tiêu được xây dựng nhằm mục đích hỗ trợ người dùng hiểu rõ hơn về thói quen tài chính cá nhân. Hệ thống áp dụng thuật toán học máy K-Means để thực hiện việc phân nhóm các giao dịch chi tiêu dựa trên những đặc trưng quan trọng được trích xuất từ dữ liệu thực tế.

Thu thập và tiền xử lý dữ liệu

- Hệ thống sử dụng các giao dịch chi tiêu đã được người dùng ghi nhận, trong đó chỉ phân tích các khoản chi. Mỗi giao dịch được trích xuất các đặc trưng quan trọng như:

- + Số tiền chi tiêu (được chuẩn hóa bằng log nhằm giảm ảnh hưởng của các khoản chi quá lớn).
- + Danh mục chi tiêu.
- + Thời gian giao dịch (giờ trong ngày, ngày trong tháng, ngày trong tuần).
- + Thông tin ngữ cảnh như: chi tiêu cuối tuần, đầu tháng hoặc cuối tháng.

+ Nhóm tính chất chi tiêu: chi tiêu thiết yếu, chi tiêu sinh hoạt linh hoạt, chi tiêu giải trí và hưởng thụ, đầu tư và tích lũy, nghĩa vụ tài chính, chi tiêu đột xuất và bất thường, chi tiêu phát triển bản thân, chi tiêu xã hội và gia đình.

- Các đặc trưng này được chuẩn hóa bằng StandardScaler nhằm đảm bảo thuật toán K-Means hoạt động ổn định và chính xác.

Phân cụm hành vi chi tiêu bằng K-Means

- Dựa trên tập đặc trưng đã xử lý, hệ thống áp dụng thuật toán K-Means để phân cụm các giao dịch chi tiêu. Số cụm được xác định linh hoạt dựa trên số lượng giao dịch của người dùng nhằm đảm bảo kết quả phân tích có ý nghĩa thực tế.

- Sau khi phân cụm ban đầu, các cụm được gộp và gán nhãn hành vi dựa trên đặc điểm nổi bật, bao gồm:

- + Khoản chi giá trị lớn và đột biến.
- + Chi tiêu sinh hoạt thiết yếu hằng ngày.
- + Chi tiêu giải trí và phong cách sống.
- + Chi tiêu nhỏ lẻ nhưng tần suất cao.
- + Chi tiêu cho đầu tư, tiết kiệm và tương lai.
- + Các khoản chi phát sinh không thường xuyên.

- Mỗi cụm hành vi được mô tả chi tiết bằng tên gọi, đặc điểm, tỷ trọng chi tiêu và các danh mục nổi bật.

Xây dựng hồ sơ hành vi tài chính người dùng

- Từ kết quả phân cụm, hệ thống tổng hợp và xây dựng hồ sơ hành vi tài chính cá nhân, bao gồm:

- + Tổng số tiền đã chi tiêu
- + Giá trị chi tiêu trung bình mỗi giao dịch
- + Số lượng giao dịch
- + Nhóm hành vi chi tiêu chiếm tỷ trọng lớn nhất
- + Phong cách chi tiêu tổng quát của người dùng (ví dụ: thận trọng, cân bằng, thiên về hưởng thụ, đầu tư cho tương lai)

- + Điểm đánh giá sức khỏe tài chính (Financial Health Score)
- Các chỉ số này giúp người dùng có cái nhìn tổng quan và trực quan về tình trạng tài chính cá nhân.

Đề xuất và khuyến nghị thông minh

- Dựa trên kết quả phân tích, hệ thống tự động sinh ra các khuyến nghị chi tiêu thông minh, ví dụ:
 - + Cảnh báo khi chi tiêu thiết yếu hoặc giải trí vượt ngưỡng an toàn.
 - + Nhắc nhở tăng cường tiết kiệm và đầu tư khi tỷ lệ tích lũy còn thấp.
 - + Phát hiện thói quen chi tiêu cảm xúc (mua sắm ban đêm, chi tiêu cuối tuần).
 - + Đề xuất điều chỉnh ngân sách cho các danh mục chi tiêu mất cân đối.
- Các khuyến nghị được cá nhân hóa theo dữ liệu thực tế của từng người dùng, giúp nâng cao hiệu quả quản lý tài chính cá nhân.



Hình 4.6 Chức năng Phân cụm hành vi chi tiêu

4.3.5 Dự báo thu chi trong tương lai

Chức năng Dự báo thu chi trong tương lai được xây dựng nhằm hỗ trợ người dùng chủ động trong việc lập kế hoạch tài chính ngắn hạn. Chức năng này sử dụng mô hình LSTM để phân tích chuỗi dữ liệu tài chính theo thời gian, từ đó dự đoán xu hướng thu nhập và chi tiêu trong các giai đoạn tiếp theo dựa trên dữ liệu lịch sử.

Chuẩn bị và xử lý dữ liệu chuỗi thời gian

- Dữ liệu đầu vào của mô hình dự báo được lấy từ các giao dịch tài chính đã phát sinh của người dùng. Hệ thống tiến hành:

- + Phân tách giao dịch thành thu nhập (giá trị dương) và chi tiêu (giá trị âm).
- + Tổng hợp dữ liệu theo ngày, tạo thành chuỗi thời gian gồm hai thành phần: tổng thu nhập và tổng chi tiêu mỗi ngày.
- + Bổ sung các ngày không phát sinh giao dịch với giá trị bằng 0 nhằm đảm bảo chuỗi dữ liệu liên tục.

- Dữ liệu sau khi tổng hợp được chuẩn hóa bằng MinMaxScaler để đưa về cùng thang đo, giúp mô hình học hiệu quả hơn.

Mô hình dự báo và chiến lược lựa chọn

- Hệ thống áp dụng chiến lược dự báo linh hoạt, kết hợp giữa học sâu và phương pháp thống kê:

- + Khi số lượng dữ liệu đủ lớn và có độ biến động rõ ràng, hệ thống sử dụng mô hình LSTM để học các quy luật theo chuỗi thời gian.

- + Trong trường hợp dữ liệu chưa đủ hoặc không phù hợp, hệ thống tự động chuyển sang phương pháp dự báo thống kê nhằm đảm bảo vẫn có kết quả dự báo ổn định.

- Mô hình LSTM được xây dựng với một lớp LSTM chính và các lớp Dense để dự đoán giá trị thu nhập hoặc chi tiêu trong các ngày tiếp theo, với độ dài chuỗi đầu vào được cấu hình linh hoạt.

Phân tích xu hướng thu chi

- Bên cạnh việc dự báo giá trị cụ thể, hệ thống còn phân tích xu hướng tài chính dựa trên dữ liệu lịch sử gần nhất:

- + So sánh mức thu chi trung bình của hai giai đoạn liên tiếp (14 ngày trước và 14 ngày gần nhất).

- + Xác định xu hướng tăng, giảm hoặc ổn định.

- + Sử dụng kết quả dự báo để xác nhận xu hướng nhằm tăng độ tin cậy.

- Kết quả phân tích được diễn giải dưới dạng mô tả ngôn ngữ tự nhiên, giúp người dùng dễ dàng hiểu được tình trạng tài chính của mình.

Tổng hợp kết quả và khuyến nghị

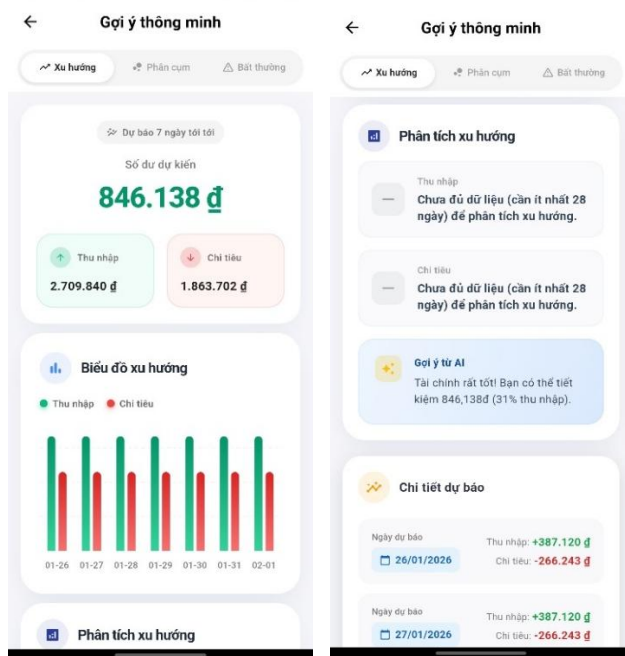
- Dựa trên kết quả dự báo thu nhập và chi tiêu trong tương lai, hệ thống:

- + Tính toán tổng thu, tổng chi và số dư dự kiến trong khoảng thời gian dự báo.

- + Đưa ra đánh giá tình hình tài chính (thặng dư, cân bằng hoặc thâm hụt).

- + Sinh ra khuyến nghị thông minh nhằm hỗ trợ người dùng điều chỉnh kế hoạch chi tiêu và tiết kiệm phù hợp.

- Các khuyến nghị này giúp người dùng nhận biết sớm các rủi ro tài chính và chủ động đưa ra quyết định kịp thời.



Hình 4.7 Chức năng dự báo thu chi trong tương lai

4.3.6 Phát hiện chi tiêu bất thường

Chức năng phát hiện chi tiêu bất thường được xây dựng nhằm hỗ trợ người dùng sớm nhận biết các giao dịch có dấu hiệu bất thường so với thói quen chi tiêu hằng ngày. Hệ thống sử dụng phương pháp học máy không giám sát Isolation Forest, kết hợp với các luật logic nghiệp vụ, để phát hiện và đánh giá mức độ rủi ro của từng giao dịch.

Trích xuất và xây dựng đặc trưng giao dịch

- Mỗi giao dịch tài chính được chuyển đổi thành một tập đặc trưng phục vụ cho quá trình phân tích, bao gồm:

- + Giá trị giao dịch tuyệt đối và logarit hóa số tiền.
- + Thời gian phát sinh giao dịch (giờ trong ngày, thứ trong tuần, ngày trong tháng).
- + Phân loại giao dịch theo nhóm chi tiêu (ăn uống, di chuyển, thuê nhà, giải trí, ...).
- + Thống kê ngắn hạn trung bình và độ lệch chuẩn chi tiêu trong 7 ngày gần nhất.
- + Tần suất giao dịch trong ngày và tổng chi tiêu theo ngày Mức độ sai lệch của giao dịch so với lịch sử chi tiêu cùng loại và toàn bộ dữ liệu (z-score).

- Các đặc trưng này giúp mô hình nhận diện được các giao dịch có giá trị bất thường, thời điểm bất thường hoặc tần suất phát sinh không bình thường.

Phát hiện bất thường bằng Isolation Forest

- Sau khi chuẩn hóa dữ liệu bằng StandardScaler, hệ thống áp dụng mô hình Isolation Forest để phân loại giao dịch:

- + Các giao dịch bình thường được gán nhãn 1
 - + Các giao dịch bất thường được gán nhãn -1
- Mô hình không yêu cầu dữ liệu huấn luyện có nhãn trước, phù hợp với bài toán phát hiện hành vi bất thường trong chi tiêu cá nhân. Mức độ nhạy của mô hình được điều chỉnh thông qua tham số contamination, cho phép tùy biến theo nhu cầu người dùng.

Kết hợp luật nghiệp vụ (Rule-based)

- Bên cạnh mô hình học máy, hệ thống còn sử dụng các luật logic để phát hiện những trường hợp đặc biệt, bao gồm:

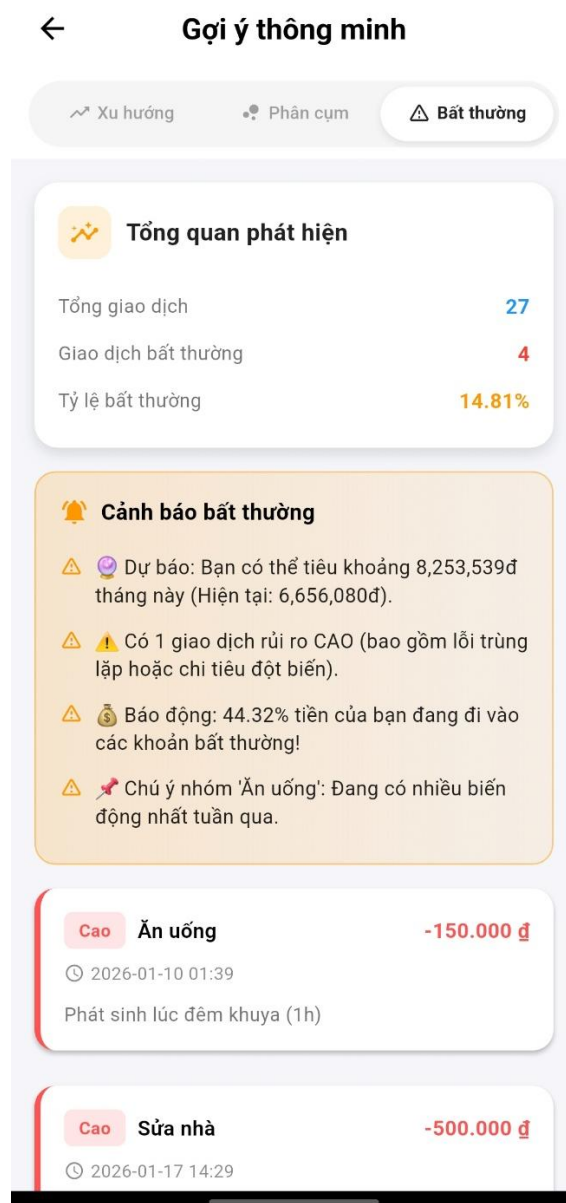
- + Các giao dịch trùng lặp về số tiền, loại chi tiêu và thời điểm phát sinh trong khoảng thời gian ngắn.
- + Các khoản chi phát sinh vào khung giờ bất thường (đêm khuya).
- + Giao dịch có tần suất quá cao trong cùng một ngày.
- Các giao dịch này được đánh dấu bất thường ngay cả khi mô hình học máy không phát hiện rõ ràng, giúp tăng độ tin cậy của hệ thống.

Xác định nguyên nhân và mức độ nghiêm trọng

- Mỗi giao dịch bất thường được hệ thống phân tích để đưa ra nguyên nhân cụ thể:
 - + Số tiền cao gấp nhiều lần mức trung bình gần đây.
 - + Khác biệt lớn so với lịch sử chi tiêu cùng loại.
 - + Tổng chi tiêu trong ngày vượt ngưỡng bình thường.
 - + Phát sinh tại thời điểm không thường xuyên.
- Dựa trên điểm bất thường và độ lệch so với lịch sử, hệ thống phân loại mức độ rủi ro thành ba mức:
 - + Cao: có khả năng gây ảnh hưởng lớn đến tài chính.
 - + Trung bình: cần theo dõi và kiểm soát.
 - + Thấp: khác biệt nhẹ so với thói quen.

Tổng hợp kết quả và cảnh báo người dùng

- Sau khi phân tích, hệ thống:
 - + Tổng hợp số lượng và tỷ lệ giao dịch bất thường.
 - + Thống kê nhóm chi tiêu có nhiều bất thường nhất.
 - + Ước lượng mức độ ảnh hưởng của các giao dịch bất thường đến tổng chi tiêu.
- Dựa trên kết quả này, hệ thống tự động sinh ra các cảnh báo thông minh nhằm mục đích giúp người dùng:
 - + Nhận biết sớm các rủi ro tài chính
 - + Phát hiện lỗi trùng lặp hoặc chi tiêu đột biến
 - + Điều chỉnh hành vi chi tiêu kịp thời



Hình 4.8 Chức năng phát hiện chi tiêu bất thường

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

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 chi 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

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 chi 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 1: 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.

Mã nguồn của hệ thống được quản lý và lưu trữ trên nền tảng GitHub tại địa chỉ:
https://github.com/khanhrukachi/DATN_LuxFinance

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.

Phụ lục 2: hướng dẫn sử dụng

Hướng dẫn sử dụng các chức năng chính của hệ thống quản lý và phân tích tài chính cá nhân. Nội dung được mô tả theo trình tự thao tác của người dùng nhằm giúp người sử dụng dễ dàng tiếp cận và khai thác hiệu quả các chức năng của ứng dụng.

1. Đăng ký và đăng nhập tài khoản

Người dùng mở ứng dụng và lựa chọn chức năng đăng ký tài khoản mới bằng cách nhập các thông tin cá nhân cần thiết. Sau khi đăng ký thành công, người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã tạo để bắt đầu sử dụng các chức năng của ứng dụng.

2. Thêm khoản thu và chi

Sau khi đăng nhập, người dùng có thể thực hiện việc thêm các khoản thu nhập hoặc chi tiêu. Người dùng nhập các thông tin như số tiền, loại giao dịch, danh mục và thời gian phát sinh. Dữ liệu sau khi được lưu sẽ được hệ thống sử dụng để quản lý và phân tích tài chính.

3. Xem lịch sử giao dịch

Hệ thống cho phép người dùng xem danh sách các giao dịch đã phát sinh theo ngày, tháng hoặc năm. Người dùng có thể theo dõi chi tiết từng khoản thu, chi nhằm kiểm soát tình hình tài chính cá nhân.

4. Xem thông kê và báo cáo

Người dùng có thể truy cập chức năng thống kê để xem các báo cáo tổng hợp dưới dạng biểu đồ và số liệu. Các báo cáo được hiển thị theo từng khoảng thời gian như ngày, tháng và năm, giúp người dùng đánh giá tổng quan tình hình thu chi và xu hướng tài chính.

5. Sử dụng chức năng phân tích tài chính bằng AI

Hệ thống hỗ trợ các chức năng phân tích tài chính thông minh dựa trên dữ liệu chi tiêu của người dùng. Thông qua đó, người dùng có thể nhận được các thông tin phân tích, cảnh báo chi tiêu bất thường và dự đoán xu hướng chi tiêu trong tương lai nhằm hỗ trợ việc quản lý tài chính hiệu quả hơn.

6. Chỉnh sửa thông tin cá nhân và đăng xuất

Người dùng có thể cập nhật thông tin cá nhân trong mục hồ sơ để đảm bảo dữ liệu luôn chính xác. Khi kết thúc quá trình sử dụng, người dùng có thể đăng xuất khỏi hệ thống để đảm bảo an toàn cho tài khoản.

Phụ lục 3: Các Use Case chi tiết của hệ thống

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 bảo mật và mật khẩu xác nhận có trùng khớp hay không. - 11. Nếu thông tin hợp lệ, hệ thống mã hóa mật khẩu mới, 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.</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 yêu cầu người dùng đăng nhập lại sau khi thay đổi mật khẩu để đảm bảo an toàn.</p>

Phụ lục 1 Use Case chi tiết của hệ thống

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	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng truy cập vào giao diện Tài khoản.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Thông tin cá nhân hiện tại của người dùng được hiển thị một cách đầy đủ.</p> <p>Không có dữ liệu cá nhân nào bị thay đổi nếu xảy ra lỗi.</p>
Đảm bảo thành công	<p>Thông tin cá nhân của người dùng được cập nhật thành công.</p> <p>Dữ liệu mới được lưu trữ an toàn trong hệ thống.</p>
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.

	<ul style="list-style-type: none"> - 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> <p>12b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
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>

Phụ lục 2 Use case thay đổi thông tin cá nhân

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	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng truy cập vào giao diện Tài khoản.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Giao diện lựa chọn ngôn ngữ được hiển thị đầy đủ.</p> <p>Ngôn ngữ hiện tại của ứng dụng không bị thay đổi nếu bị xảy ra lỗi.</p>
Đảm bảo thành công	<p>Ngôn ngữ ứng dụng được thay đổi thành công theo lựa chọn của người dùng.</p> <p>Thiết lập ngôn ngữ mới được lưu trữ an toàn trong hệ thống.</p>
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>

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

Phụ lục 3 Use case thay đổi ngôn ngữ

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 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).

Phụ lục 4 Use case xem thông tin cá nhân

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	<p>E1 – Lỗi hiển thị giao diện</p> <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	Chế độ sáng/tối được áp dụng cho toàn bộ giao diện ứng dụng. Thiết lập giao diện sẽ được tự động áp dụng cho những lần đăng nhập tiếp theo.

Phụ lục 5 Use case tùy chỉnh sáng tối

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. - 8. Hệ thống hiển thị chi tiết chi tiêu của giao dịch. - 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>

Phụ lục 6 Use case xem lịch sử chi tiêu

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	Tập CSV được tạo thành công. Người dùng tải xuống tập 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 tập dữ liệu định dạng CSV. - 7. Hệ thống cung cấp liên kết tải xuống hoặc tự động tải tập CSV về thiết bị. - 8. Hệ thống hiển thị thông báo “Xuất dữ liệu thành công”. - 9. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Lỗi kết nối cơ sở dữ liệu</p> <p>5a. Hệ thống không thể truy xuất dữ liệu từ cơ sở dữ liệu.</p> <p>5b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thử lại sau.</p> <p>E2 – Lỗi tạo tập CSV</p> <p>6a. Xảy ra lỗi trong quá trình xử lý và tạo tập 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 tập CSV.</p> <p>7b. Hệ thống hiển thị thông báo yêu cầu người dùng giải phóng bộ nhớ.</p>
Ghi chú	Dữ liệu xuất ra CSV có thể bao gồm các thông tin như thời gian, danh mục, số tiền, ... Tập CSV có thể được sử dụng cho các công cụ phân tích như Excel hoặc Google Sheets.

Phụ lục 7 Đặc tả use case xuất CSV

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	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đã có dữ liệu chi tiêu được lưu trong hệ thống.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Giao diện tìm kiếm chi tiêu được hiển thị đầy đủ.</p> <p>Không có dữ liệu chi tiêu nào bị thay đổi nếu xảy ra lỗi.</p>

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 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	Hệ thống có thể hỗ trợ tìm kiếm kết hợp nhiều tiêu chí. Kết quả tìm kiếm được cập nhật theo thời gian thực dựa trên dữ liệu hiện có.

Phụ lục 8 Use case tìm kiếm chi tiêu

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).

Phụ lục 9 Use case xem chi tiết chi tiêu

Use case thêm ngân sách

Tên use case	Thêm ngân sách.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập thiết lập hoặc thêm một ngân sách mới cho từng danh mục chi tiêu trong một khoảng thời gian nhất định (tháng), nhằm hỗ trợ việc quản lý và kiểm soát chi tiêu cá nhân hiệu quả.
Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đang ở giao diện ứng dụng.</p> <p>Thiết bị có kết nối Internet.</p>

Đảm bảo tối thiểu	Giao diện thêm ngân sách được hiển thị đầy đủ. Không có dữ liệu ngân sách nào được lưu khi thông tin không hợp lệ hoặc xảy ra lỗi.
Đảm bảo thành công	Thông tin ngân sách được lưu thành công trong hệ thống. Ngân sách được liên kết với tài khoản người dùng và danh mục chi tiêu tương ứng. Hệ thống sẵn sàng sử dụng dữ liệu ngân sách để theo dõi, cảnh báo và phân tích chi tiêu.
Kích hoạt	Người dùng chọn chức năng “Thêm ngân sách” 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 ngân sách” trên giao diện ứng dụng. - 3. Hệ thống hiển thị giao diện biểu mẫu thêm ngân sách mới. - 4. Người dùng nhập các thông tin ngân sách cần thiết. - 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 ngân sách vào cơ sở dữ liệu. - 8. Hệ thống cập nhật danh sách ngân sách của người dùng. - 9. Hệ thống hiển thị thông báo “Thêm ngân sách 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 (danh mục, số tiền, thời gian).</p> <p>6b. Hệ thống hiển thị thông báo yêu cầu người dùng nhập đầy đủ thông tin.</p> <p>E2 – Số tiền ngân sách không hợp lệ</p> <p>6a. Hệ thống phát hiện số tiền ngân sách không hợp lệ.</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 ngân sách.</p> <p>7b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	Mỗi ngân sách được gán với tài khoản người dùng đang đăng nhập. Ngân sách được sử dụng để so sánh với dữ liệu chi tiêu thực tế. Hệ thống có thể phát sinh cảnh báo khi chi tiêu vượt quá ngân sách đã thiết lập.

Phụ lục 10 Use case thêm ngân sách

Use case sửa ngân sách

Tên use case	Sửa ngân sách.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập chỉnh sửa thông tin của một ngân sách đã được thiết lập trước đó (như số tiền, danh mục), nhằm phù hợp với nhu cầu quản lý tài chính hiện tại.
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống.

	<p>Người dùng đang ở giao diện quản lý ngân sách.</p> <p>Ngân sách cần sửa đã tồn tại trong hệ thống.</p> <p>Thiết bị có kết nối Internet.</p>
Đảm bảo tối thiểu	<p>Giao diện thêm ngân sách được hiển thị đầy đủ.</p> <p>Không có dữ liệu ngân sách nào được lưu khi thông tin không hợp lệ hoặc xảy ra lỗi.</p>
Đảm bảo thành công	<p>Thông tin ngân sách được cập nhật thành công trong hệ thống.</p> <p>Dữ liệu ngân sách được cập nhật chính xác và liên kết đúng với tài khoản người dùng.</p> <p>Các chức năng theo dõi và cảnh báo ngân sách sử dụng dữ liệu mới nhất.</p>
Kích hoạt	Người dùng chọn chức năng “Sửa ngân sách” 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 truy cập chức năng quản lý ngân sách. - 3. Người dùng chọn một ngân sách cần chỉnh sửa. - 4. Hệ thống hiển thị giao diện biểu mẫu chỉnh sửa ngân sách với thông tin hiện tại. - 5. Người dùng chỉnh sửa các thông tin cần thiết. - 6. Người dùng nhấn nút “Lưu”. - 7. Hệ thống kiểm tra tính hợp lệ của các thông tin đã nhập. - 8. Hệ thống cập nhật thông tin ngân sách trong cơ sở dữ liệu. - 9. Hệ thống hiển thị thông báo “Cập nhật ngân sách thành công”. - 10. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Thông tin không hợp lệ</p> <p>8a. Hệ thống phát hiện thông tin chỉnh sửa không hợp lệ hoặc thiếu dữ liệu bắt buộc.</p> <p>8b. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.</p> <p>E2 – 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>Ngân sách sau khi chỉnh sửa sẽ được áp dụng ngay cho việc theo dõi chi tiêu.</p> <p>Các cảnh báo vượt ngân sách được tính toán lại dựa trên dữ liệu mới.</p>

Phụ lục 11 Use case sửa ngân sách

Use case xóa ngân sách

Tên use case	Xóa ngân sách.
Tên tác nhân	Người dùng.
Mô tả	Ca sử dụng cho phép người dùng đã đăng nhập xóa một ngân sách đã được tạo trước đó khỏi hệ thống khi không còn nhu cầu sử dụng.
Điều kiện tiên quyết	<p>Người dùng đã đăng nhập vào hệ thống.</p> <p>Người dùng đang ở giao diện quản lý ngân sách.</p> <p>Ngân sách cần sửa đã tồn tại trong hệ thống.</p>

	Thiết bị có kết nối Internet.
Đảm bảo tối thiểu	Không có dữ liệu nào bị xóa khi người dùng hủy thao tác hoặc xảy ra lỗi hệ thống. Dữ liệu chi tiêu của người dùng không bị ảnh hưởng.
Đảm bảo thành công	Ngân sách được xóa thành công khỏi hệ thống. Danh sách ngân sách của người dùng được cập nhật chính xác.
Kích hoạt	Người dùng chọn chức năng “Xóa ngân sách” 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 truy cập chức năng quản lý ngân sách. - 3. Người dùng chọn một ngân sách cần xóa. - 4. Người dùng chọn chức năng “Xóa ngân sách”. - 5. Hệ thống hiển thị hộp thoại xác nhận xóa ngân sách. - 6. Người dùng xác nhận thao tác xóa. - 7. Hệ thống xóa ngân sách khỏi cơ sở dữ liệu. - 8. Hệ thống cập nhật danh sách ngân sách của người dùng. - 9. Hệ thống hiển thị thông báo “Xóa ngân sách thành công”. - 10. Ca sử dụng kết thúc.
Ngoại lệ	<p>E1 – Thông tin không hợp lệ</p> <p>6a. Người dùng chọn “Hủy” tại hộp thoại xác nhận.</p> <p>6b. Hệ thống đóng hộp thoại và không xóa dữ liệu.</p> <p>E2 – Lỗi hệ thống</p> <p>7a. Xảy ra lỗi trong quá trình xóa dữ liệu.</p> <p>7b. Hệ thống hiển thị thông báo lỗi hệ thống.</p>
Ghi chú	Việc xóa ngân sách không làm thay đổi các khoản chi tiêu đã được lưu trước đó. Người dùng có thể tạo lại ngân sách mới cho cùng danh mục và thời gian nếu cần.

Phụ lục 12 Use case xóa ngân sách

TÀI LIỆU THAM KHẢO

- [1] Kevin Wang-Nava, *Personal Finance – Your Money, Your Life*, Open Educational Resource, 2024.
- [2] C. Garman and R. Forgue, *Personal Finance*, 14th ed., Cengage Learning, Boston, MA, USA, 2022.
- [3] The Dart Team, “Dart Programming Language Specification,” Version 3.0, Google LLC, Mountain View, CA, USA, 2023.
- [4] M. J. Heidari, *Flutter Complete Reference*, 2nd ed., Berkeley, CA, USA: Apress Media LLC, 2023.
- [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*, 3rd ed., O’Reilly Media, Sebastopol, CA, USA, 2022.
- [8] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, Upper Saddle River, NJ, USA, 2020.
- [9] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 2022.
- [10] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed., Pearson, Boston, MA, USA, 2019.
- [11] T. Hastie, R. Tibshirani, and R. J. Tibshirani, *An Introduction to Statistical Learning*, 2nd ed., Springer, 2021.
- [12] H. Xiang et al., “OptIForest: Optimal Isolation Forest for Anomaly Detection,” in Proc. Int. Joint Conf. Artificial Intelligence, 2023.
- [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.