

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO CUỐI KỲ MÔN PHÁT TRIỂN ỨNG DỤNG DỊ ĐỘNG

ĐỀ TÀI : ỨNG DỤNG QUẢN LÝ CHI TIÊU

Người hướng dẫn: **GV LÊ VĂN VANG**

Người thực hiện: **LÝ TUẤN AN-52000620**

PREANN MESA– 52000909

Lớp : 20050201

Khoá : 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO CUỐI KỲ MÔN PHÁT TRIỂN ỨNG DỤNG DỊ ĐỘNG

ĐỀ TÀI : ỨNG DỤNG QUẢN LÝ CHI TIÊU

Người hướng dẫn: **GV LÊ VĂN VANG**

Người thực hiện: **LÝ TUẤN AN-52000620**

PREANN MESA– 52000909

Lớp : 20050201

Khoá : 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi lời cảm ơn chân thành nhất đến toàn bộ quý thầy cô Trường Đại học Tôn Đức Thắng, quý thầy cô Khoa Công nghệ thông tin đã dạy dỗ, và truyền đạt những kiến thức quý báu cũng như đã tạo mọi điều kiện thuận lợi nhất để em được tiếp cận và hoàn thành môn phát triển ứng dụng dị động.

Với thời gian và trình độ còn hạn chế, bài báo cáo không thể tránh khỏi những thiếu sót. Kính mong quý thầy chỉ bảo và đóng góp ý kiến để bài báo cáo của em được hoàn thiện hơn. Đó sẽ là hành trang quý giá để em có thể hoàn thiện về kỹ năng và kiến thức liên quan đến những vấn đề sau này.

Lời cuối cùng, em xin kính chúc quý thầy thật nhiều sức khỏe, thành công, hạnh phúc, và luôn giữ mãi sự nhiệt huyết để có thể giúp thêm thật nhiều thế hệ sinh viên Trường Đại Học Tôn Đức Thắng có nhiều sự tự tin và vững kiến thức với môn phát triển ứng dụng dị động. Ngọn lửa của môn học sẽ ngày càng được lan tỏa rộng rãi nhiều hơn!

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Em xin cam đoan đây là sản phẩm báo cáo của riêng em và được sự hướng dẫn của GV LÊ VĂN VANG ;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Preann Mesa

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được, những phát hiện cơ bản trong vòng 1 -2 trang.

MỤC LỤC

CHƯƠNG 1 – MỞ ĐẦU	5
1.1 Giới thiệu đề tài.....	5
1.Mục tiêu nghiên cứu	5
2.Mức độ ý nghĩa chọn đề tài.....	7
3.Đối tượng và khách thể nghiên cứu	7
4. Phương pháp và phạm vi nghiên cứu đề tài.....	8
5. Đóng góp của đề tài.....	8
CHƯƠNG 2 – PHÂN TÍCH THIẾT KẾ HỆ THỐNG	9
2.1 Usecase tổng quát.....	9
2.2 Đặc tả Usecase	11
2.2.1 Đặc tả usecase đăng ký	11
2.2.2 Đặc tả Usecase đăng nhập.....	12
2.2.3 Đặc tả Usecase đăng xuất	14
2.2.4 Đặc tả usecase về thiết lập ví cá nhân.....	16
2.2.5 Usecase thêm các khoản chi tiêu	17
2.2.6 Usecase về thống kê.....	18
2.2.7 Usecase chia sẻ thông tin chi tiêu	18
2.2.8 Usecase thiết lập giới hạn chi tiêu hàng tháng.....	19
2.2.9 Usecase tạo kế hoạch chi tiêu cho tương lai và theo dõi tiến độ	19
2.2.10 Usecase tạo các nhóm chi tiêu	20
2.2.11 Usecase tạo liên kết ngân hàng trực tuyến hoặc e-wallet	20
2.3 Sơ đồ hoạt động	21
2.3.1 Sơ đồ hoạt động đăng nhập.....	21
2.3.2 Sơ đồ hoạt động đăng ký.....	22
2.4 Sơ đồ tuần tự	23

a. Đăng nhập	23
b. Sơ đồ tuần tự xem thống kê	24
2.5 Kiến trúc Andriod	25
CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT	26
3.1 Sử dụng SQLite database trong ứng dụng Andriod.....	26
3.2 Sử dụng Recyclerview trong Andriod	28
3.3 Intent là gì ?.....	29
CHƯƠNG 4 – THIẾT KẾ ỨNG DỤNG	30
TÀI LIỆU THAM KHẢO.....	37

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

f Tần số của dòng điện và điện áp (Hz)

p Mật độ điện tích khối (C/m³)

CÁC CHỮ VIẾT TẮT

CSTD Công suất tác dụng

MF Máy phát điện

BER Tỷ lệ bit lỗi

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1:Usecase tổng quan.....	9
Hình 2:Use đăng nhập	14
Hình 3:Use case đăng xuất.....	16
Hình 4:UC thiết lập ví cá nhân.....	17
Hình 5:UC Thêm các khoản chi.....	18
Hình 6: UC thông kê	18
Hình 7:UC chia sẻ thông tin chi tiêu	19
Hình 8:UC thiết lập giới hạn chi tiêu hàng tháng	19
Hình 9:UC Tạo kế hoạch chi tiêu cho tương lai	20
Hình 10:UC tạo các nhóm chi tiêu theo các mục.....	20
Hình 11: UC liên kết ngân hàng trực tuyến.....	21
Hình 12: Sơ đồ hoạt động đăng nhập.....	21
Hình 13: Sơ đồ hoạt động đăng ký.....	22
Hình 14: Sơ đồ tuần tự đăng nhập.....	23
Hình 15:Sơ đồ tuần tự xem thống kê	24
Hình 16:Kiến trúc hệ điều hành Andriod.....	25

CHƯƠNG 1 – MỞ ĐẦU

1.1 Giới thiệu đề tài

1. Mục tiêu nghiên cứu

Sau đây nhóm chúng em sẽ giới thiệu chung về đề tài ứng dụng quản lý chi tiêu cho sinh viên gồm các chức năng chính như sau:

Ứng dụng gồm có 5 chức năng chính là đăng ký, đăng nhập, thiết lập ví cá nhân, thêm vào các khoản chi tiêu hàng ngày, hàng tuần, hàng tháng hoặc năm, chức năng xem thống kê chi tiêu theo thời gian và loại chi tiêu (ăn uống, giải trí, di chuyển, v.v.).

Chức năng đăng ký, đăng nhập và thiết lập ví cá nhân cho phép người dùng đăng ký tài khoản bằng email hoặc số điện thoại. Sau khi đăng ký tài khoản, người dùng bắt đầu thiết lập ví tiền với số tiền hiện có của mình thông qua nhiều lựa chọn như tiền mặt, ví điện tử và ví ngân hàng.

Chức năng thêm vào các khoản chi tiêu hàng ngày, tuần, tháng hoặc năm cho phép người dùng thêm và quản lý các khoản chi tiêu mỗi ngày, mỗi tuần, mỗi tháng hoặc mỗi năm. Để thêm một khoản chi tiêu, người dùng cần nhập vào thông tin về số tiền, loại chi tiêu (ví dụ như ăn uống, giải trí, đi lại, v.v.), ngày chi tiêu và mục tiêu chi tiêu (tùy chọn). Sau khi thêm chi tiêu, người dùng có thể xem chi tiêu theo ngày, tuần, tháng hoặc năm và so sánh với các mục tiêu đã đặt. Ngoài ra, người dùng có thể sửa hoặc xóa bất kỳ chi tiêu nào mà họ muốn.

Chức năng xem thống kê chi tiêu theo thời gian và loại chi tiêu (ăn uống, giải trí, di chuyển, v.v.) cho phép người dùng xem thống kê chi tiêu của họ theo thời gian và theo các loại chi tiêu khác nhau. Ví dụ, người dùng có thể xem thống kê chi tiêu của họ trong một tuần hoặc một tháng, hoặc xem tổng chi tiêu trong một năm. Điều này giúp họ có một cái nhìn tổng quan về những chi tiêu của họ trong một khoảng thời gian cụ thể. Ngoài ra người dùng cũng có thể xem thống kê chi tiêu theo các loại chi tiêu khác nhau, ví dụ như ăn uống, giải trí, di chuyển, v.v. Điều này giúp họ có một cái nhìn tổng quan về

những chi tiêu của họ trong mỗi loại và có thể quản lý chi tiêu theo loại đó một cách hiệu quả. Hơn nữa chức năng này cung cấp cho người dùng một cách dễ dàng để xem và theo dõi thống kê chi tiêu, giúp họ quản lý chi tiêu một cách hiệu quả và đảm bảo rằng họ không chi tiêu quá nhiều tiền vào các mục không cần thiết.

Chia sẻ thông tin chi tiêu.

- Cho phép người dùng chia sẻ thông tin về chi tiêu của họ với người thân hoặc bạn bè của họ.
- Có thể chia sẻ tổng quan chi tiêu hàng tháng hoặc chi tiêu chi tiết của mỗi khoản chi tiêu. Chức năng này giúp người dùng dễ dàng chia sẻ và hợp tác với người thân hoặc bạn bè trong việc quản lý chi tiêu của mình.

Thiết lập giới hạn chi tiêu hàng tháng và nhận thông báo khi đạt giới hạn.

- Người dùng có thể thiết lập giới hạn cho từng loại chi tiêu hoặc tổng chi tiêu hàng tháng. Thông báo sẽ giúp người dùng dễ dàng theo dõi và giữ kiểm soát chi tiêu của mình, tránh tình trạng chi tiêu quá mức.

Tạo một kế hoạch chi tiêu cho tương lai và theo dõi tiến độ.

- Người dùng có thể tạo kế hoạch cho từng loại chi tiêu hoặc tổng chi tiêu, và xem xét tiến độ của mình so với kế hoạch. Điều này giúp người dùng dễ dàng theo dõi và kiểm soát chi tiêu trong tương lai, giúp tiết kiệm và quản lý tài chính một cách hiệu quả hơn.

Thêm các ghi chú về chi tiêu.

- Cho phép người dùng ghi lại những ghi chú về mỗi khoản chi tiêu của họ. Ghi chú này có thể bao gồm mục đích, địa điểm, hoặc bất kỳ thông tin nào liên quan đến chi tiêu đó. Điều này giúp người dùng ghi nhớ những chi tiêu của mình và có thể dễ dàng truy vấn nếu cần thiết trong tương lai.

Tạo các nhóm chi tiêu.

- Cho phép người dùng chia nhỏ các mục chi tiêu của mình ra thành nhiều nhóm, ví dụ như chi tiêu cho gia đình, chi tiêu cho sức khỏe, chi tiêu cho giải trí v.v. Sau đó, họ có thể theo dõi chi tiêu theo từng nhóm để có một cái nhìn tổng quan về nhu cầu và số tiền đã chi tiêu cho mỗi nhóm.

Tích hợp với các ứng dụng ngân hàng trực tuyến hoặc e-wallet để tự động cập nhật chi tiêu.

- Có thể được hiện thực bằng cách sử dụng API của các ứng dụng ngân hàng trực tuyến hoặc e-wallet để tự động nhận dữ liệu về giao dịch của người dùng và cập nhật vào chi tiêu. Nó giúp cho việc quản lý chi tiêu trở nên dễ dàng và tiết kiệm thời gian hơn.

Tạo các bảng biểu để visualized chi tiêu theo thời gian.

- Có thể giúp người dùng cảm nhận rõ hơn về tình hình chi tiêu của họ theo thời gian. Bằng cách tạo các biểu đồ và biểu đồ cột, người dùng có thể nhanh chóng nhìn thấy những xu hướng và sự thay đổi trong chi tiêu của họ. Việc này cũng giúp họ tìm ra những vấn đề về chi tiêu và cải thiện cách quản lý chi tiêu của họ trong tương lai.

2.Mức dịch ý nghĩa chọn đề tài

Trong thời đại công nghệ 4.0, các ứng dụng quản lý chi tiêu trở thành một công cụ hữu ích để giúp người dùng quản lý tiền bạc của mình một cách hiệu quả hơn. Các ứng dụng này cung cấp các tính năng như theo dõi chi tiêu, lập kế hoạch tài chính, phân loại chi tiêu, đưa ra lời khuyên tiết kiệm, tạo ngân sách và nhiều tính năng khác.

Một ứng dụng quản lý chi tiêu tốt sẽ giúp người dùng tiết kiệm được nhiều thời gian và tiền bạc, đồng thời giúp họ biết được mình đang chi tiêu vào những gì và làm thế nào để quản lý tài chính một cách tốt nhất. Trong đề tài này, chúng ta có thể tìm hiểu về các tính năng cơ bản và phức tạp của các ứng dụng quản lý chi tiêu, cách thiết kế và phát triển các ứng dụng này, và tầm quan trọng của chúng đối với cuộc sống hàng ngày của chúng ta.

3.Đối tượng và khách thể nghiên cứu

- Đối tượng nghiên cứu:
 - + Ngon ngữ lập trình Java.
 - + Công cụ lập trình Andriod studio.
 - + Quy trình nghiệp vụ quản lý chi tiêu theo ngày, tháng, năm.
- Khách thể nghiên cứu:
 - + Người dùng có nhu cầu quản lý chi tiêu của mình

+ Có thể liên kết với tài khoản ngân hàng, ví điện tử

4. Phương pháp và phạm vi nghiên cứu đề tài

a. Phương pháp nghiên cứu tài liệu

- Sử dụng tài liệu hướng dẫn lập trình Andriod từ các diễn đàn công nghệ thông tin(www.topcode.vn ...)
- Sử dụng các vedio hướng dẫn có liên quan từ Youtube.
- Sử dụng thông tin từ website Stackoverflow, chatGPT và một số website khác để sửa lỗi phát sinh trong quá trình làm dự án này.

b. Phương pháp nghiên cứu thực tiễn

- Phương pháp quan sát: Quan sát việc quản lý của một app Money Lover.

c. Phương pháp tổng hợp , khảo sát và phân tích số liệu

- Thu thập thông tin từ việc làm khảo sát bằng form
- Thông kê, tổng kết số liệu.
- Phân tích, đưa ra kết quả và thực hiện.

Phạm vi quay quanh các nghiệp vụ quản lý đảm bảo đầy đủ các chức năng chính của ứng dụng và khai thác các chức năng mới có liên quan đến quản lý chi tiêu.

5. Đóng góp của đề tài

- Ý nghĩa khoa học:

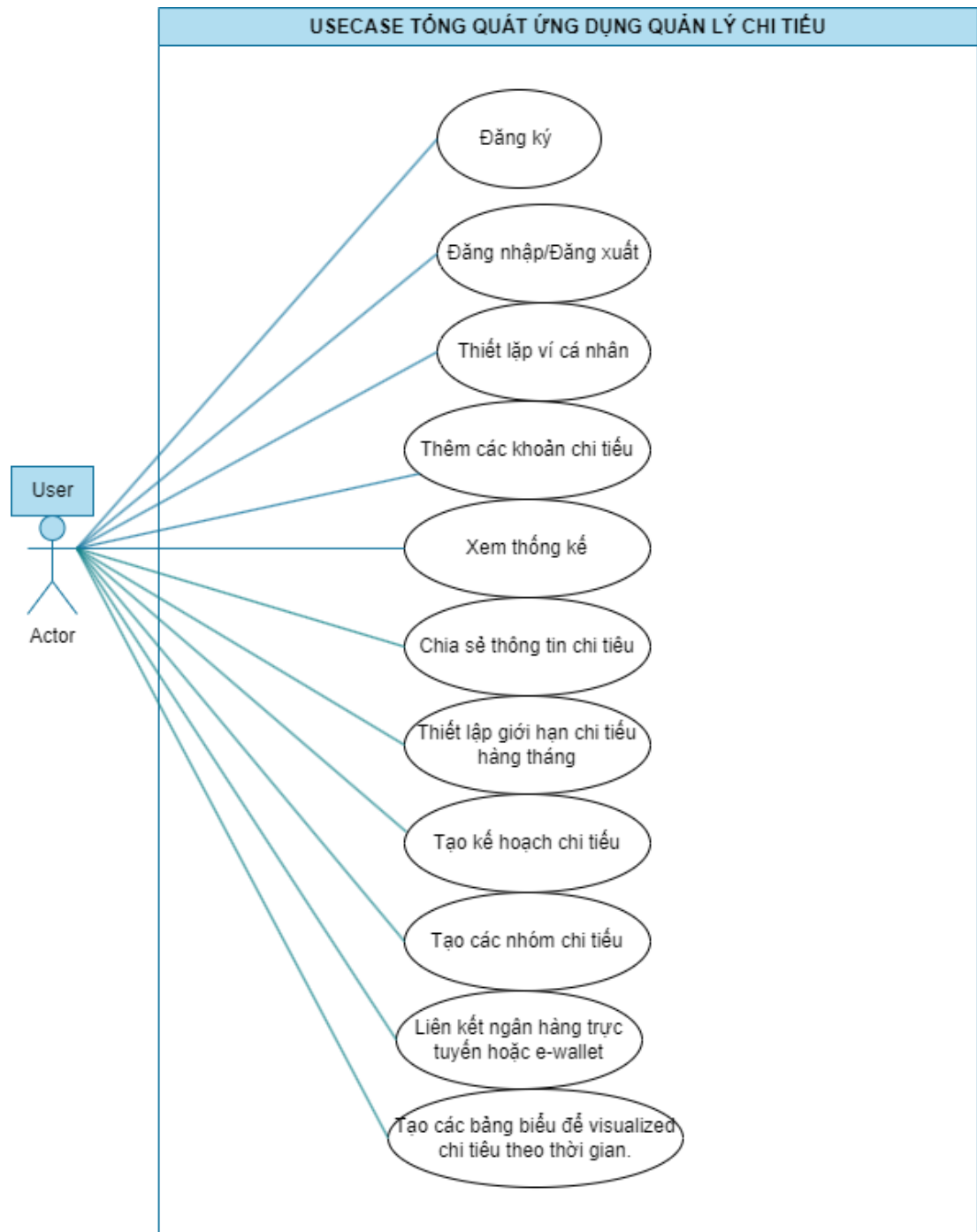
Kết quả nghiên cứu của đề tài góp phần nâng cao nhận thức, khả năng thu thập thông tin, phân tích yêu cầu của sinh viên để xây dựng được ứng dụng dành cho sinh viên và ứng dụng dành cho người quản lý. Bước đầu tiếp cận ngôn ngữ lập trình để xây dựng chương trình.

- Ý nghĩa thực tiễn:

Ứng dụng giúp cho việc quản lý chi tiêu của sinh viên dễ dàng, rõ ràng hơn, bên cạnh đó nó còn giúp cho người tiêu dùng cảm thấy tiện lợi và hài lòng khi sử dụng ứng dụng sản phẩm.

CHƯƠNG 2 – PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1 Usecase tổng quát



Hình 1:Usecase tổng quan

Biểu đồ Usecase tổng quan:

- Tác nhân: User.
- Mô tả Usecase:
 - Đăng kí: đăng kí người dùng mới.
 - Đăng nhập/ đăng xuất: đăng nhập hệ thống bằng tài khoản cá nhân của người dùng và đăng xuất đối với tài khoản đó.
 - Thiết lập ví cá nhân: Sau khi đăng ký tài khoản, người dùng bắt đầu thiết lập ví tiền với số tiền hiện có của mình thông qua nhiều lựa chọn như tiền mặt, ví điện tử, ví ngân hàng.
 - Thêm vào các khoản chi tiêu hàng ngày, tuần, tháng hoặc năm: Cho phép người dùng thêm và quản lý các khoản chi tiêu mỗi ngày, mỗi tuần, mỗi tháng hoặc mỗi năm. Để thêm một khoản chi tiêu, người dùng cần nhập vào thông tin về số tiền, loại chi tiêu (ví dụ như ăn uống, giải trí, đi lại, v.v.), ngày chi tiêu và ghi chú
 - Xem thống kê chi tiêu theo thời gian và loại chi tiêu: cho phép người dùng xem thống kê chi tiêu của họ theo thời gian và theo các loại chi tiêu khác nhau.
 - Chia sẻ thông tin chi tiêu: có thể chia sẻ tổng quan chi tiêu hàng tháng hoặc chi tiêu chi tiết của mỗi khoản chi tiêu. Chức năng này giúp người dùng dễ dàng chia sẻ và hợp tác với người thân hoặc bạn bè trong việc quản lý chi tiêu của mình.
 - Thiết lập giới hạn chi tiêu hàng tháng và nhận thông báo khi đạt giới hạn: người dùng có thể thiết lập giới hạn cho từng loại chi tiêu hoặc tổng chi tiêu hàng tháng. Thông báo sẽ giúp người dùng dễ dàng theo dõi và giữ kiểm soát chi tiêu của mình, tránh tình trạng chi tiêu quá mức.

- Tạo một kế hoạch chi tiêu : người dùng có thể tạo kế hoạch cho từng loại chi tiêu hoặc tổng chi tiêu, và xem xét tiến độ của mình so với kế hoạch. Điều này giúp người dùng dễ dàng theo dõi và kiểm soát chi tiêu trong tương lai, giúp tiết kiệm và quản lý tài chính một cách hiệu quả hơn
 - Tạo các nhóm chi tiêu: người dùng chia nhỏ các mục chi tiêu của mình ra thành nhiều nhóm, ví dụ như chi tiêu cho gia đình, chi tiêu cho sức khỏe, chi tiêu cho giải trí v.v. Sau đó, họ có thể theo dõi chi tiêu theo từng nhóm để có một cái nhìn tổng quan về nhu cầu và số tiền đã chi tiêu cho mỗi nhóm,
 - Liên kết ngân hàng trực tuyến hoặc e-wallet: có thể được hiện thực bằng cách sử dụng API của các ứng dụng ngân hàng trực tuyến hoặc e-wallet để tự động nhận dữ liệu về giao dịch của người dùng và cập nhật vào chi tiêu. Nó giúp cho việc quản lý chi tiêu trở nên dễ dàng và tiết kiệm thời gian hơn.
 - Tạo các bảng biểu để visualized chi tiêu theo thời gian: có thể giúp người dùng cảm nhận rõ hơn về tình hình chi tiêu của họ theo thời gian. Bằng cách tạo các biểu đồ và biểu đồ cột, người dùng có thể nhanh chóng nhìn thấy những xu hướng và sự thay đổi trong chi tiêu của họ.
- Người dùng
 - Người dùng có thể thực hiện tất cả các chức năng trên.

2.2 Đặc tả Usecase

2.2.1 Đặc tả usecase đăng ký

Biểu đồ usecase đăng ký:



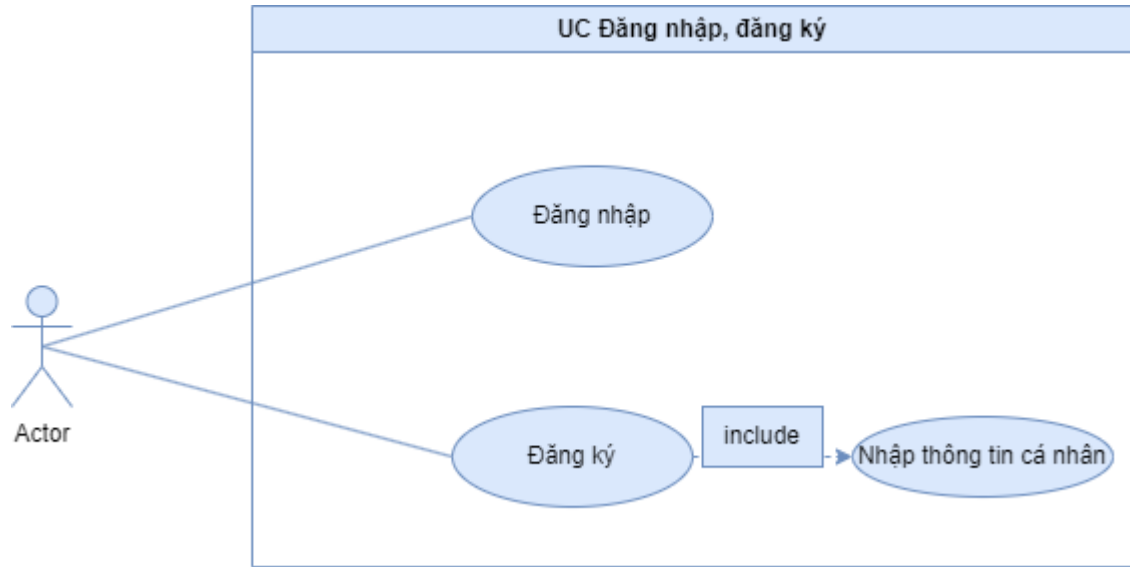
Biểu đồ usecase đăng kí:

- Tác nhân: Người dùng.
- Hệ thống: Đăng kí.
- Mô tả:
 - Người dùng đăng kí tài khoản với hệ thống mới có thể sử dụng vào hệ thống.
 - Hệ thống phân quyền mặc định là user cho tài khoản mới đăng kí.

2.2.2 Đặc tả Usecase đăng nhập

Use case	Đăng nhập
Ngữ cảnh	Tác nhân muốn đăng nhập vào hệ thống.
Sự kiện kích hoạt	Tác nhân nhấn nút Đăng nhập.
Mô tả	Tác nhân đăng nhập vào hệ thống để thực hiện các thao tác
Actors	Người dùng
Điều kiện tiên quyết	Tác nhân đã có tài khoản, mật khẩu để truy cập vào hệ thống

Kết quả	Nếu use case này được thực hiện thành công, tác nhân sẽ đăng nhập vào hệ thống đã được cấp quyền. Ngược lại thì tác nhân không thể truy cập vào hệ thống phân quyền mà chỉ ở trạng thái đăng nhập	
Luồng sự kiện	Tác nhân	Hệ thống
	<p>1. Tác nhân đã truy cập vào trang chủ để đăng nhập vào hệ thống.</p> <p>2. Tác nhân nhập Tên đăng nhập và Mật khẩu. Sau đó, nhấn nút Đăng nhập.</p>	<p>1.1 Hệ thống yêu cầu tác nhân nhập Tên đăng nhập và Mật khẩu đã cung cấp.</p> <p>2.1 Hệ thống kiểm tra thông tin điền vào có trùng khớp với tài khoản đã cung cấp hay không.</p> <p>2.2 Hệ thống hiển thị trạng thái Đăng nhập.</p> <p>2.2.1 Nếu thông tin trùng khớp thì tác nhân được phép truy cập và trang đã được phân quyền trước đó.</p> <p>2.2.2 Ngược lại thì tác nhân không thể truy cập vào hệ thống và vẫn ở trang đăng nhập</p> <p>2.3 Use case kết thúc.</p>
Ngoại lệ	Nếu không điền đủ tên đăng nhập/mật khẩu thì hệ thống sẽ thông báo Đăng nhập thất bại và quay lại cho Tác nhân nhập lại thông tin tài khoản.	

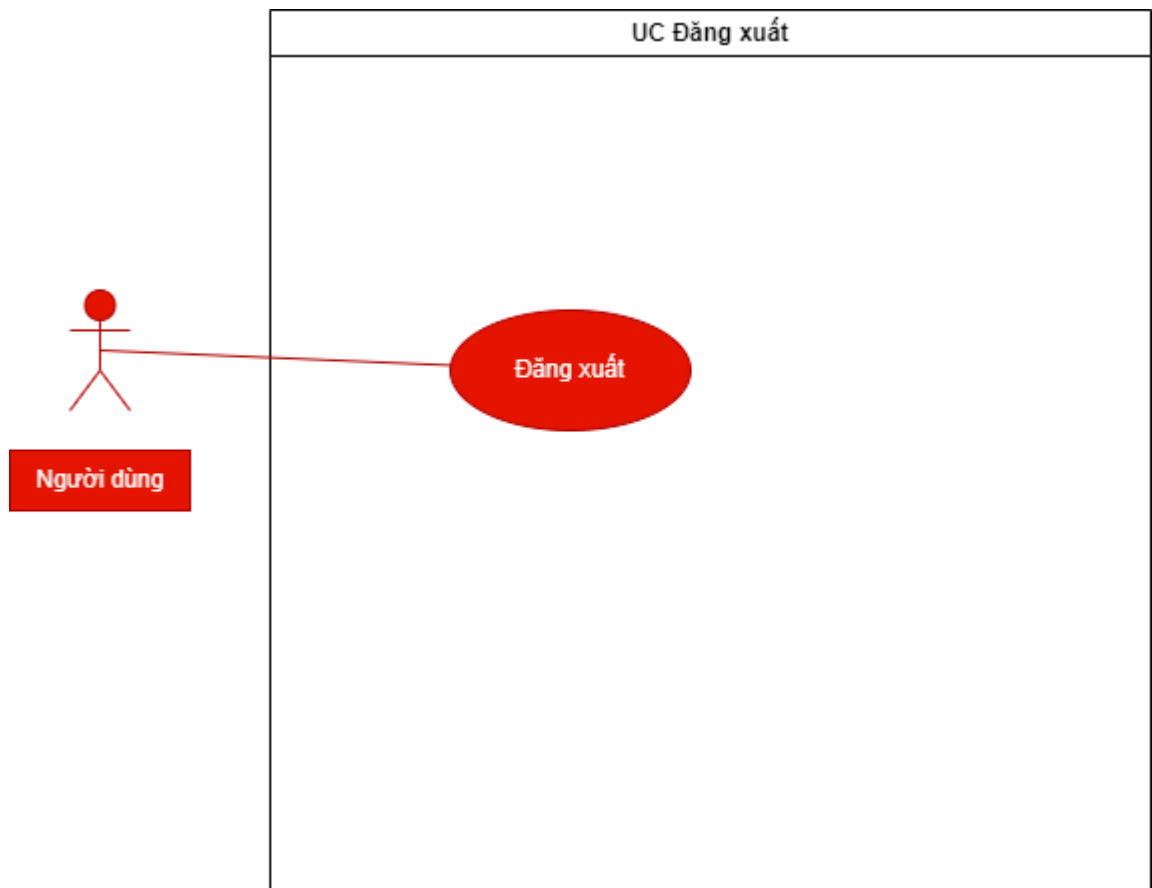


Hình 2: Use đăng nhập

2.2.3 Đặc tả Usecase đăng xuất

Use case	Đăng xuất
Ngữ cảnh	Use case đăng xuất khỏi hệ thống
Sự kiện kích hoạt	Tác nhân nhấn nút Đăng xuất.
Mô tả	Cho phép người dùng thoát khỏi hệ thống
Actors	Client
Điều kiện tiên quyết	Người dùng đã có tài khoản và đăng nhập vào hệ thống

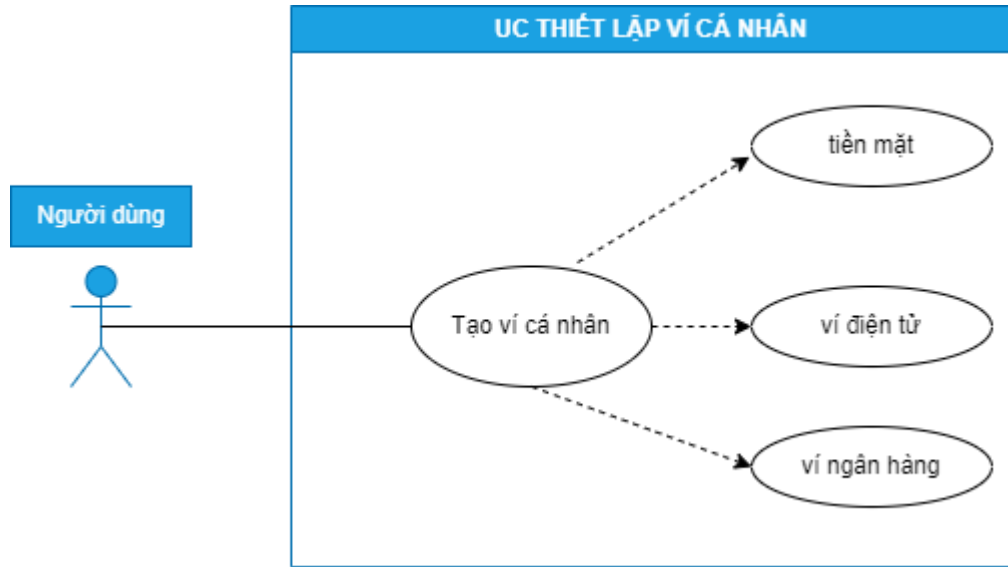
Kết quả	Nếu use case này được thực hiện thành công, thì cho phép người dùng thoát khỏi hệ thống. Ngược lại thì tác nhân không thể xuất ra khỏi hệ thống được.	
Luồng sự kiện	Tác nhân	Hệ thống
	1. Tác nhân nhấn nút đăng xuất	1.1 Hệ thống thoát ra giao diện hiện tại. 1.2 Hệ thống đưa ra giao diện đăng nhập 2. Use case kết thúc.
Ngoại lệ	Không có	



Hình 3: Use case đăng xuất

2.2.4 Đặc tả usecase về thiết lập ví cá nhân

Biểu đồ Usecase về thiết lập ví cá nhân:

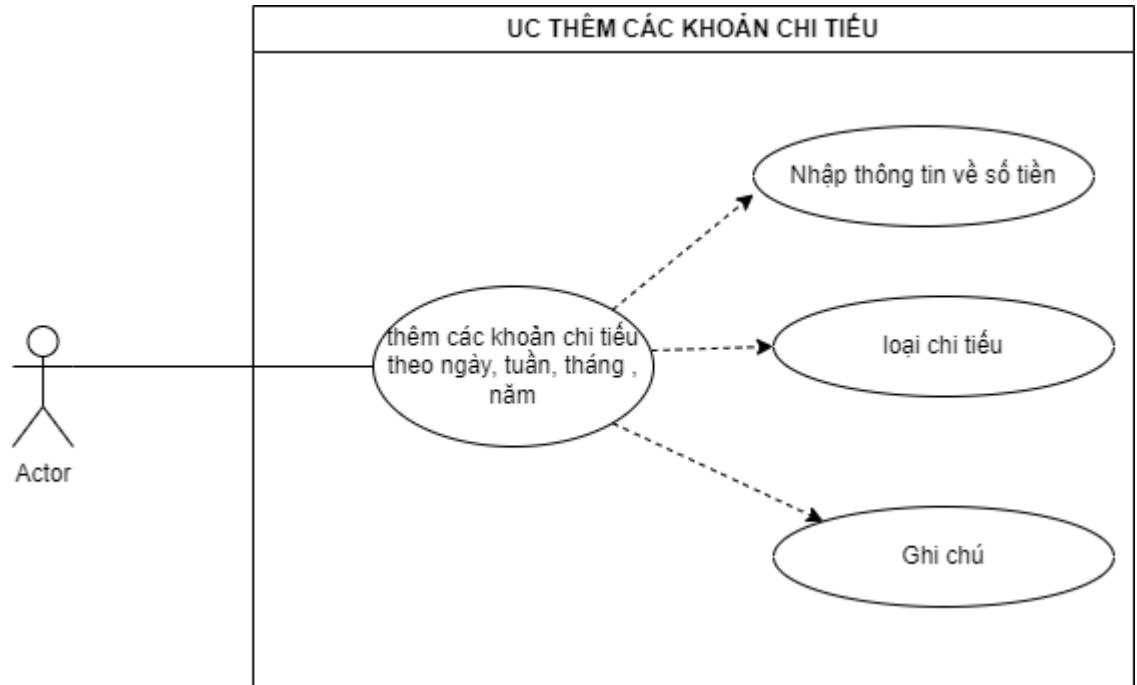


Hình 4:UC thiết lập ví cá nhân

Usecase giao dịch:

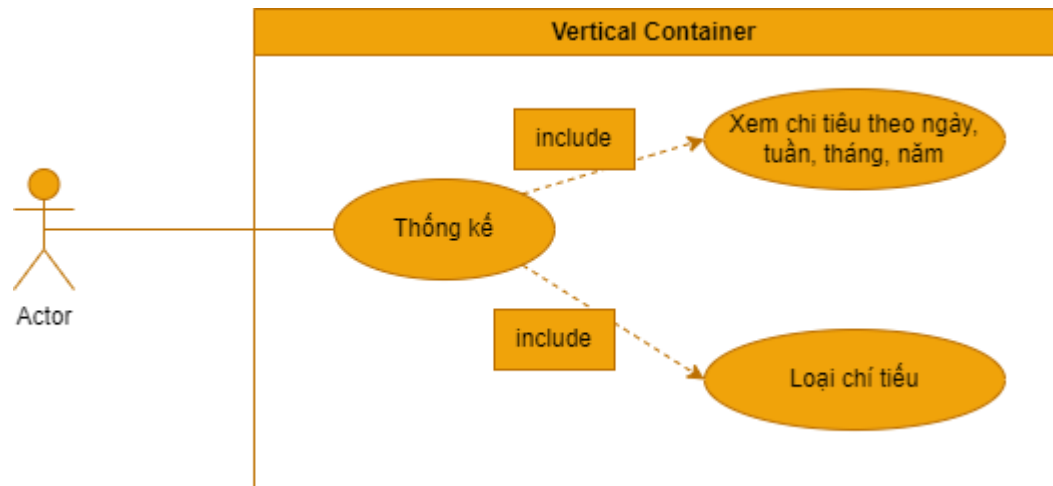
- Tác nhân: Người dùng.
- Hệ thống: thiết lập ví cá nhân
- Mô tả:
 - Người dùng bắt đầu thiết lập ví tiền với số tiền hiện có của mình thông qua nhiều lựa chọn như tiền mặt, ví điện tử, ví ngân hàng.

2.2.5 Usecase thêm các khoản chi tiêu



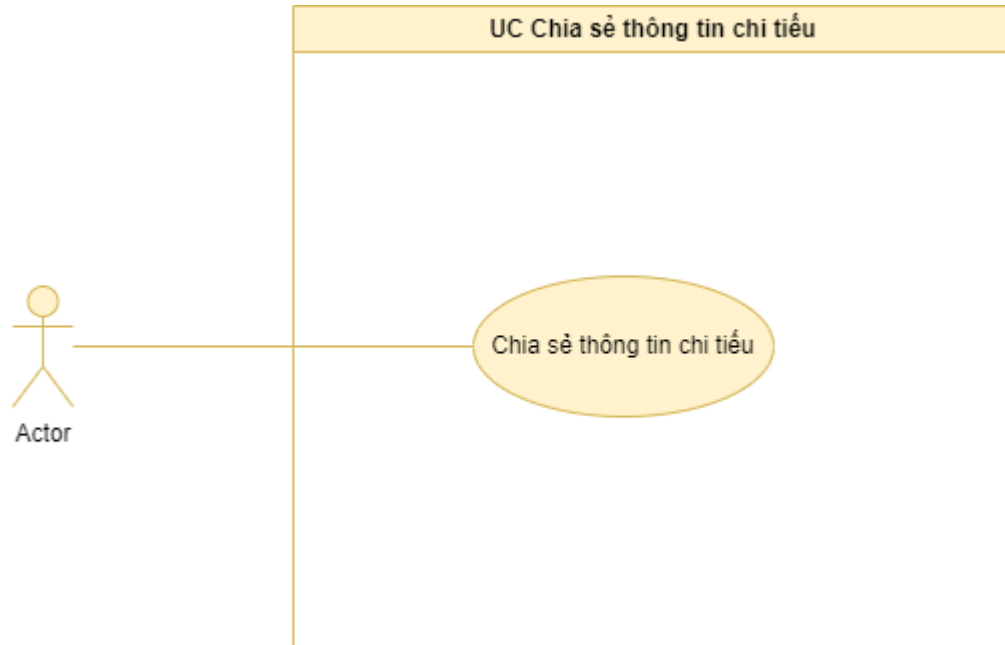
Hình 5:UC Thêm các khoản chi

2.2.6 Usecase về thống kê



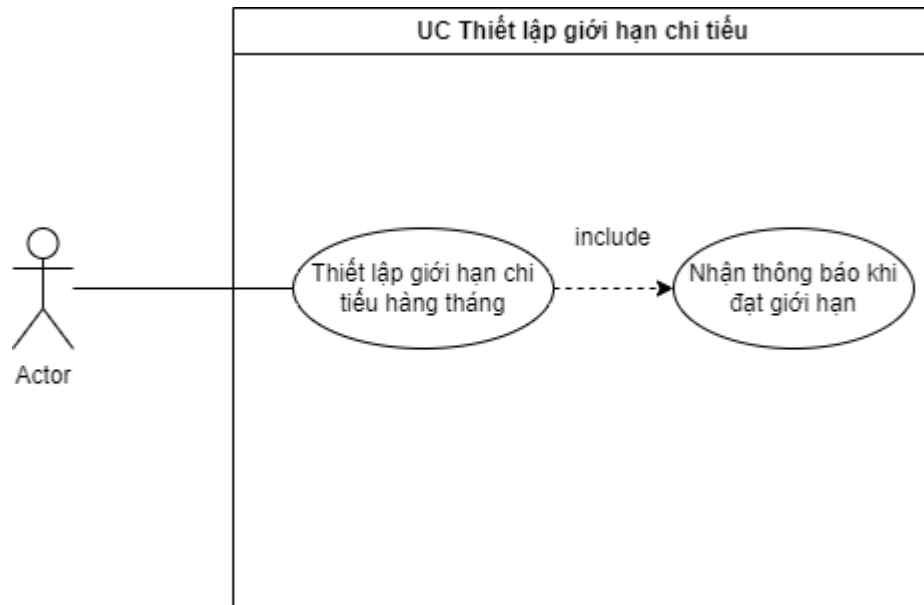
Hình 6: UC thông kê

2.2.7 Usecase chia sẻ thông tin chi tiêu



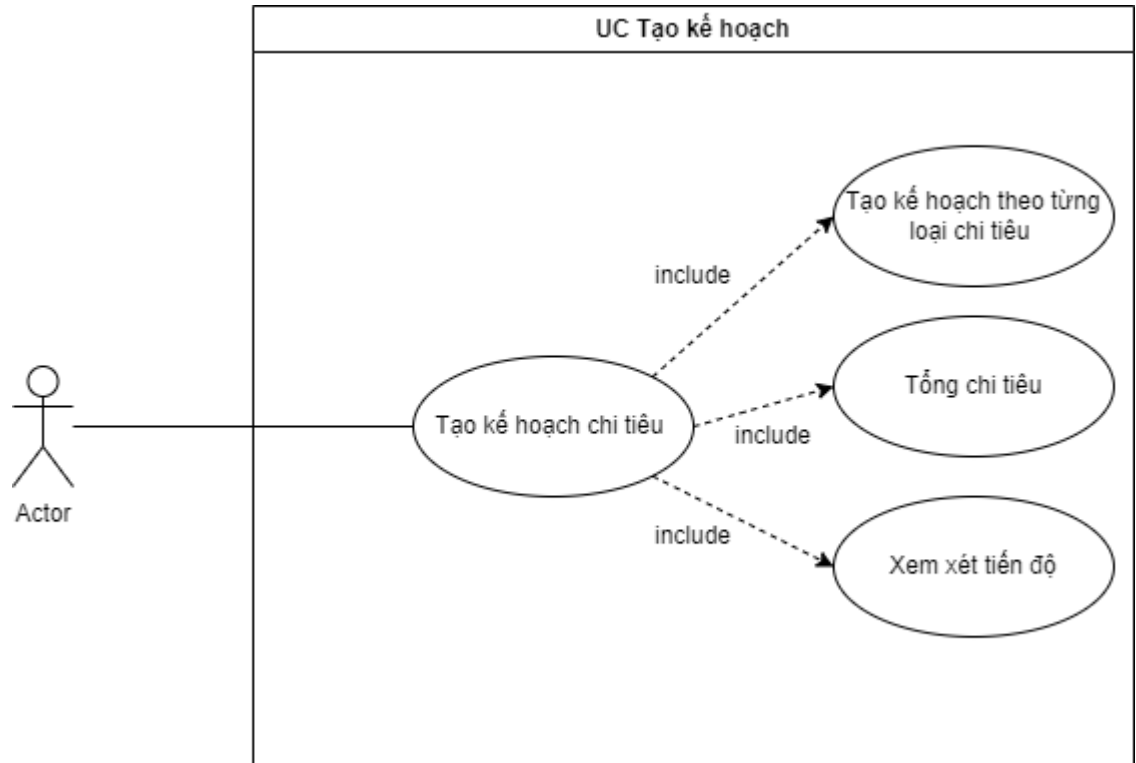
Hình 7:UC chia sẻ thông tin chi tiêu

2.2.8 Usecase thiết lập giới hạn chi tiêu hàng tháng



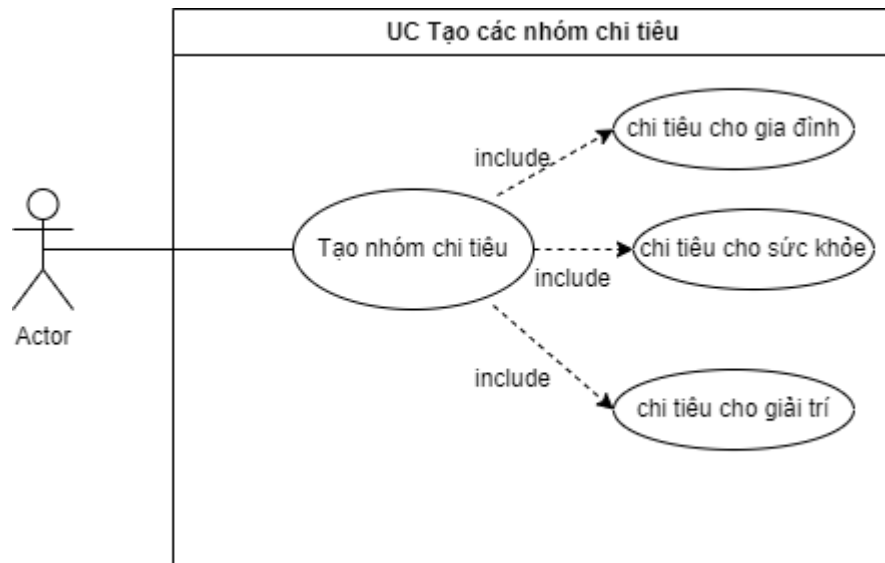
Hình 8:UC thiết lập giới hạn chi tiêu hàng tháng

2.2.9 Usecase tạo kế hoạch chi tiêu cho tương lai và theo dõi tiến độ



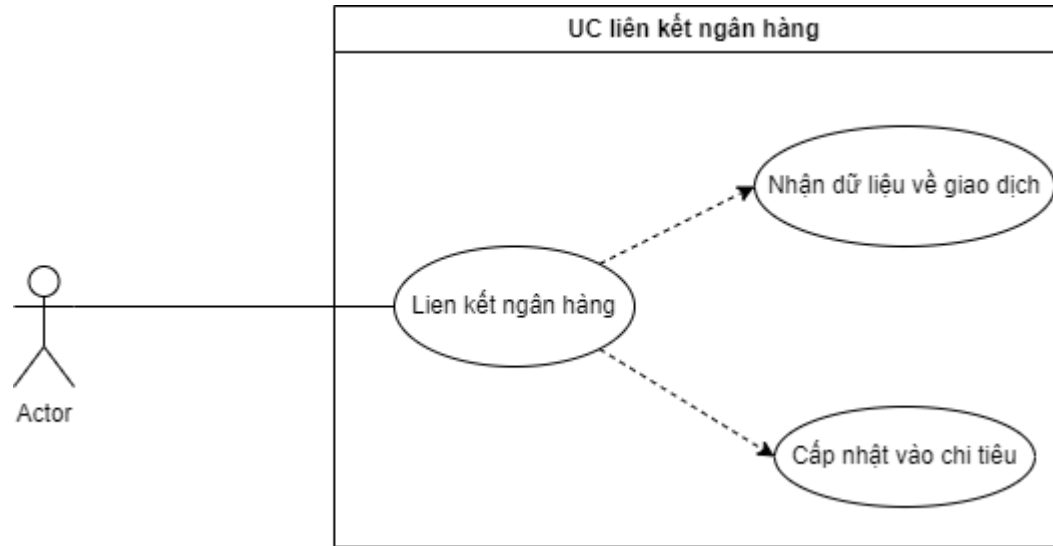
Hình 9:UC Tạo kế hoạch chi tiêu cho tương lai

2.2.10 Usecase tạo các nhóm chi tiêu



Hình 10:UC tạo các nhóm chi tiêu theo các mục

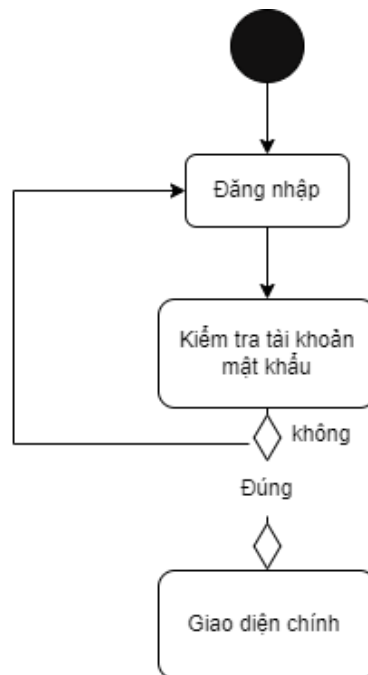
2.2.11 Usecase tạo liên kết ngân hàng trực tuyến hoặc e-wallet



Hình 11: UC liên kết ngân hàng trực tuyến

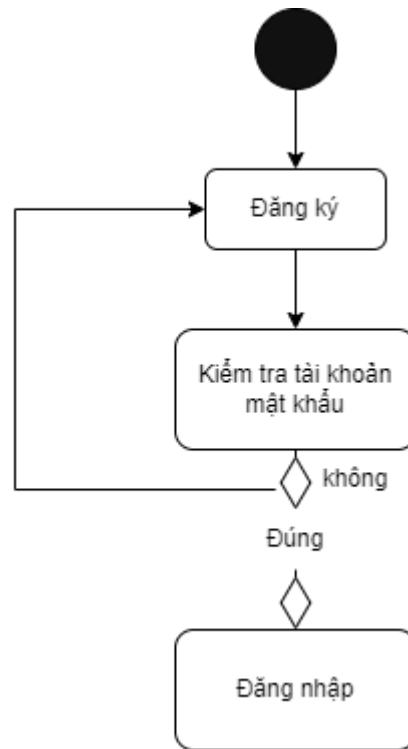
2.3 Sơ đồ hoạt động

2.3.1 Sơ đồ hoạt động đăng nhập



Hình 12: Sơ đồ hoạt động đăng nhập

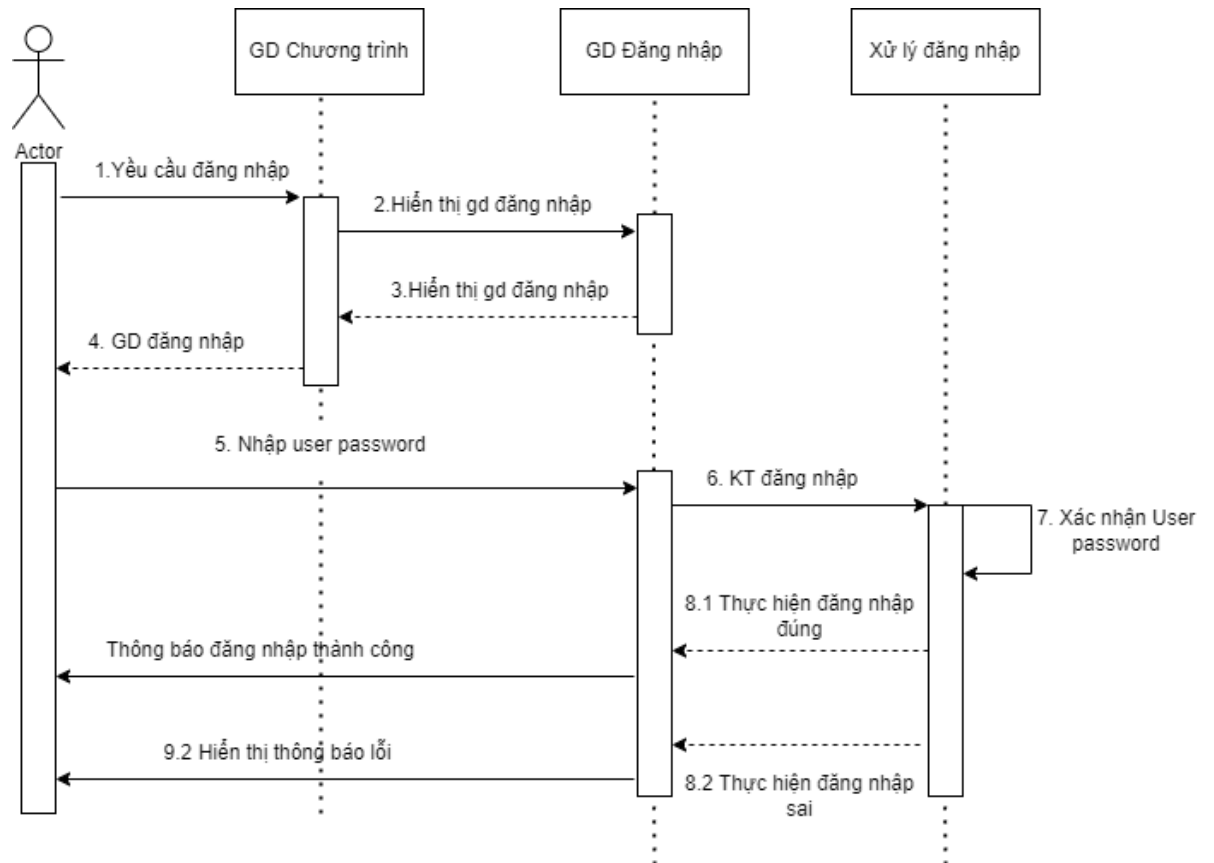
2.3.2 Sơ đồ hoạt động đăng ký



Hình 13: Sơ đồ hoạt động đăng ký

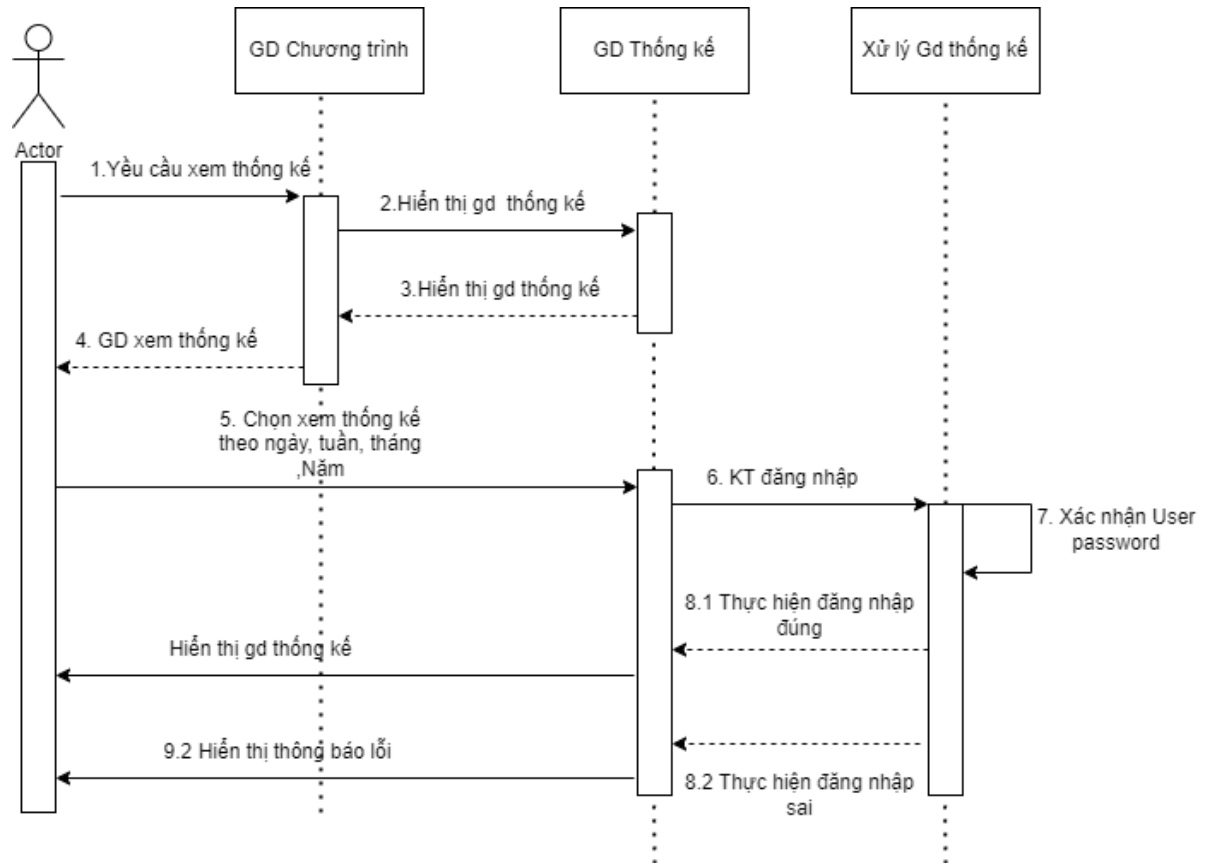
2.4 Sơ đồ tuần tự

a. Đăng nhập



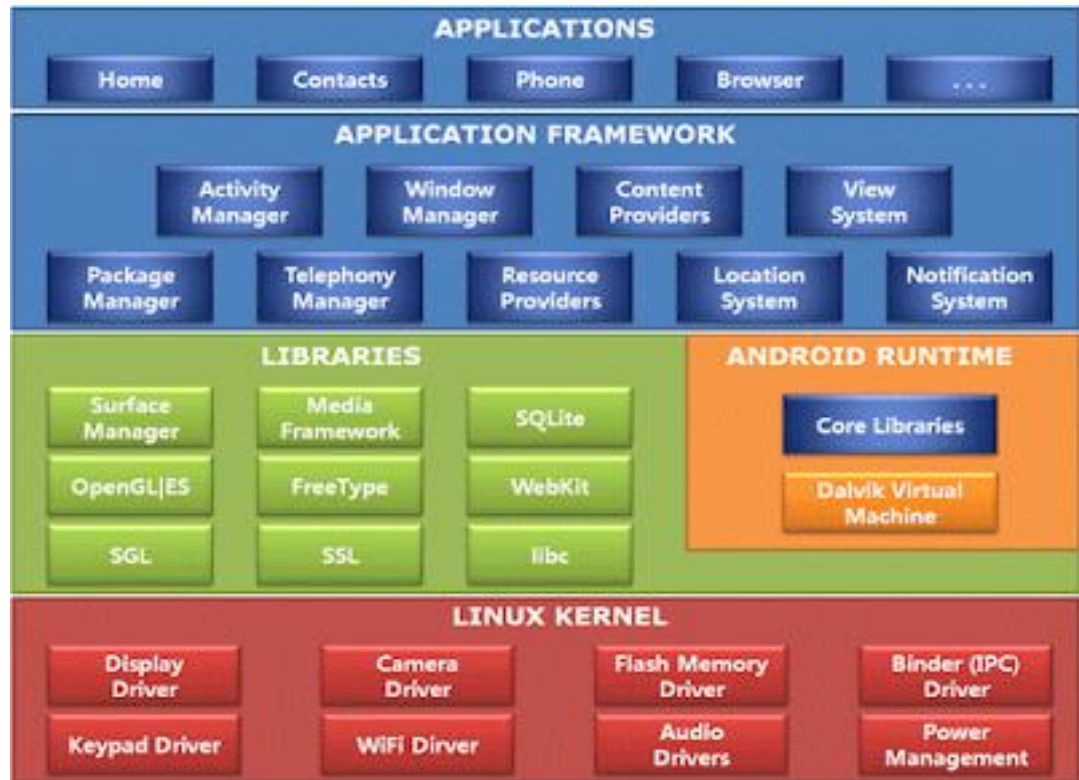
Hình 14: Sơ đồ tuần tự đăng nhập

b. Sơ đồ tuần tự xem thống kê



Hình 15:Sơ đồ tuần tự xem thống kê

2.5 Kiến trúc Andriod

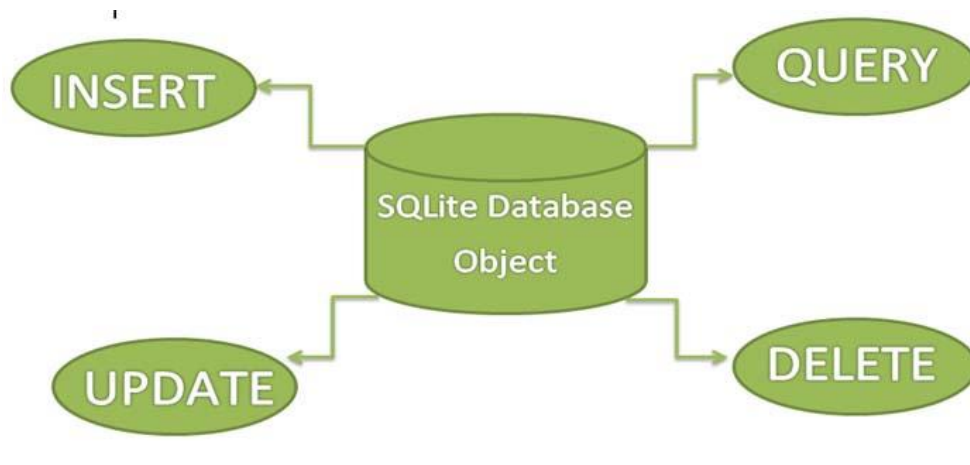


Hình 16:Kiến trúc hệ điều hành Andriod

CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT

3.1 Sử dụng SQLite database trong ứng dụng Android

- SQLite là một cơ sở dữ liệu SQL mã nguồn mở, nó lưu trữ dữ liệu vào một tập tin văn bản trên một thiết bị. Nó mặc định đã được tích hợp trên thiết bị Android. Để truy cập dữ liệu này, bạn không cần phải thiết lập bất kỳ loại kết nối nào cho nó như JDBC, ODBC, ... SQLite được Richard Hipp viết dưới dạng thư viện bằng ngôn ngữ lập trình C.
- SQLite có các ưu điểm sau:
 - Tin cậy: các hoạt động transaction (chuyển giao) nội trong cơ sở dữ liệu được thực hiện trọn vẹn, không gây lỗi khi xảy ra sự cố phần cứng
 - Tuân theo chuẩn SQL92 (chỉ có một vài đặc điểm không hỗ trợ)
 - Không cần cài đặt cấu hình
 - Kích thước chương trình gọn nhẹ, với cấu hình đầy đủ chỉ không đầy 300 kB
 - Thực hiện các thao tác đơn giản nhanh hơn các hệ thống cơ sở dữ liệu khách/chủ khác
 - Không cần phần mềm phụ trợ
 - Phần mềm tự do với mã nguồn mở, được chú thích rõ ràng



Hình 17:SQLite Database

Cách sử dụng SQLite trong lập trình Android

Tương tự như các hệ quản trị cơ sở dữ liệu khác thì khi thao tác với Database thì bạn sẽ có những hành động cổ điển đó là **CRUD: Create, Read, Update, Delete**

Đầu tiên, để thao tác với SQLite, ta phải dùng 2 đối tượng

- **SQLiteOpenHelper**: đối tượng dùng để tạo, nâng cấp, đóng mở kết nối CSDL
- **SQLiteDatabase**: đối tượng dùng để thực thi các câu lệnh SQL trên một CSDL

1. SQLiteOpenHelper

Lớp này có 2 hàm khởi tạo, một hàm 4 tham

số: **SQLiteOpenHelper**(Context context, String name, SQLiteDatabase.CursorFactory factory, int version), một hàm 5 tham

số: **SQLiteOpenHelper**(Context context, String name, SQLiteDatabase.CursorFactory factory, int version, DatabaseErrorHandler errorHandler) , tuy nhiên, ta chủ yếu làm việc với hàm 4 tham số

- Tham số 1: **Context context**: Context là một lớp trừu tượng của hệ thống, chứa thông tin môi trường ứng dụng, cung cấp các phương thức để có thể tương tác với hệ điều hành, giúp chúng ta dễ dàng truy cập và tương tác tới các tài nguyên của hệ thống...
- Tham số 2: **String name**: Tên database
- Tham số 3: **CursorFactory factory**: thường để null
- Tham số 4: **Int version**: version của database

Khi khởi tạo một đối tượng của lớp này, ta phải ghi đè 2 phương thức

1. onCreate(): phương thức này được gọi bởi framework, nếu có yêu cầu truy cập database mà lại chưa khởi tạo database, ở đây ta phải viết code khởi tạo database, cụ thể là khởi tạo bảng (chú ý: khi khởi tạo bảng, ta phải đặt tên khóa chính là **_id**)

2. onUpgrade(): phương thức này được dùng khi ứng dụng của bạn có nhiều phiên bản database đc thêm vào. Nó sẽ cập nhật database hiện có hoặc khởi tạo lại thông qua onCreate().

Lớp này có 2 phương thức **getReadableDatabase()** (chỉ đọc) và **getWritableDatabase()** (cho phép ghi đọc). Thông qua 2 phương thức này, ta có thể tạo ra một đối tượng **SQLiteDatabase**

2. SQLiteDatabase

Để tạo ra một cơ sở dữ liệu, bạn chỉ cần gọi phương thức *openOrCreateDatabase* này với tên cơ sở dữ liệu của bạn và chế độ như một tham số. Nó trả về một thể hiện của cơ sở dữ liệu SQLite mà bạn phải nhận được trong cú pháp object.

Một số hàm có sẵn trong gói cơ sở dữ liệu, Dưới danh sách sau:

- **openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags, DatabaseErrorHandler errorHandler):** Phương thức này chỉ mở thêm cơ sở dữ liệu hiện có với các chế độ cờ thích hợp. Các chế độ cờ phổ biến có thể là OPEN_READWRITE OPEN_READONLY.
- **openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags):** Phương thức này tương tự như phương thức trên vì nó cũng mở ra các cơ sở dữ liệu hiện có, nhưng nó không định nghĩa bất kỳ xử lý để xử lý các lỗi cơ sở dữ liệu.
- **openOrCreateDatabase(String path, SQLiteDatabase.CursorFactory factory):** Phương thức này không chỉ mở ra nhưng lại tạo ra cơ sở dữ liệu nếu nó không tồn tại. Phương thức này là tương đương với phương thức OpenDatabase
- **openOrCreateDatabase(File file, SQLiteDatabase.CursorFactory factory):** Phương thức này cũng tương tự như phương thức trên nhưng phải mất File đối tượng như là một con đường mà không phải là một chuỗi. Nó tương đương với file.getPath()

Lớp này có các phương thức sau để làm việc với SQLite:

- insert()
- update()
- delete()
- execSQL(): thực thi một câu lệnh SQL trực tiếp
- query(): dùng để truy vấn cơ sở dữ liệu

3.2 Sử dụng Recyclerview trong Andriod

RecyclerView là một ViewGroup nó được dùng để chuẩn bị và hiển thị các View tương tự nhau. RecyclerView được cho là sự kế thừa của ListView và GridView , và nó được giới thiệu trong phiên bản [support-v7](#) mới nhất. Một trong những lý do khiến RecyclerView được ưa chuộng là nó là một framework có thể mở rộng, và đặc biệt nó cung cấp khả năng triển khai cả bố cục Hozizontal và Vertical. Sử dụng RecyclerView khi mà data có các thành phần thay đổi trong quá trình chạy dựa trên hành động của người dùng hoặc các dự kiện mạng.

Nếu muốn sử dụng RecyclerView, bạn sẽ cần làm việc với các thành phần sau:

- RecyclerView.Adapter : đây là nơi xử lý dữ liệu và gán cho View.
- Layoutmanager: xác định ra vị trí của các item trong RecyclerView.
- ItemAnimator: Tạo hiệu ứng cho các hành động thêm , sửa, xóa các item điều này rất khó thực hiện với phiên bản tiền nhiệm (ListView)

Để sử dụng RecyclerView, ta cần phải thêm nó vào layout XML của mình và cấu hình một LayoutManager và Adapter cho nó. Sau đó, bạn có thể tùy chỉnh RecyclerView bằng cách thêm ItemDecoration và ItemAnimator nếu cần thiết.

3.3 Intent là gì ?

Intents là một thành phần quan trọng trong android. Nó cho phép các thành phần ứng dụng có thể yêu cầu các hàm từ các thành phần ứng dụng android khác. Ví dụ một activity có thể chạy một activity khác ở bên ngoài để chụp ảnh.

Intents là một objects của **android.content.Intent**. Intents sẽ được gửi đến hệ thống android để xác định hành động bạn muốn thực hiện, đối tượng bạn muốn xử lý.

Intents có thể bao gồm dữ liệu thông qua Bundle. Bundle giống như một cái hộp. Bên nhận sẽ mở bundle ra nhờ key và lấy ra dữ liệu.

Để chạy một activity, broadcast receivers , sử dụng phương thức **startActivity(intent)**. Phương thức này được định nghĩa trong đối tượng **context**. Xem ví dụ dưới đây:

Android intents thường được sử dụng chính:

- Start dịch vụ
- Gọi một activity
- Hiện thị một trang web
- Hiện thị danh sách liên hệ
- Gửi một tin nhắn
- Gọi điện thoại.

Các loại intent

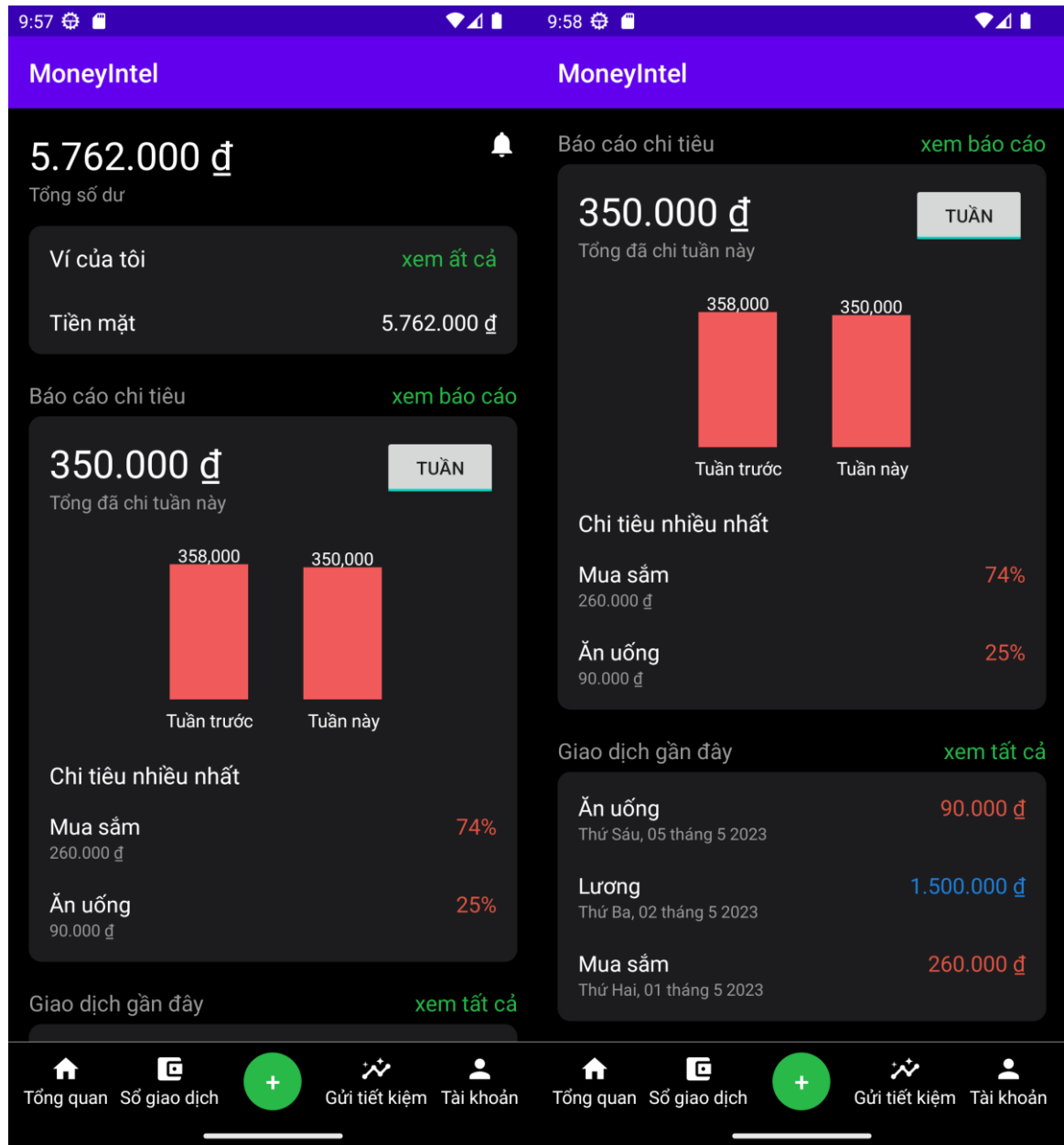
Có 2 loại intent trong Android: implicit và explicit.

CHƯƠNG 4 – THIẾT KẾ ỨNG DỤNG

4.1 Triển khai

a. Giao diện

- Tổng quan



- Sổ giao dịch

10:04

MoneyIntel

số dư

5.762.000 đ

17/04/2023 - 23/04/2023

Tuần trước

Tuần này

Số dư đầu

4.880.000 đ

Số dư cuối

4.522.000 đ

-358.000 đ

xem báo cáo cho giai đoạn này

Thứ Năm, 27 tháng 4 2023

-120.000 đ

Mua sắm

120.000 đ

Chủ Nhật, 30 tháng 4 2023

-238.000 đ

Đi lại

150.000 đ

Taxi đi đầm sen

Ăn uống

88.000 đ

Tổng quan

Sổ giao dịch

+

Gửi tiết kiệm

Tài khoản

10:07

MoneyIntel

số dư

5.762.000 đ

17/04/2023 - 23/04/2023

Tuần trước

Tuần này

Số dư đầu

4.880.000 đ

Số dư cuối

4.522.000 đ

-358.000 đ

xem báo cáo cho giai đoạn này

Thứ Hai, 01 tháng 5 2023

-260.000 đ

Mua sắm

260.000 đ

Thứ Ba, 02 tháng 5 2023

1.500.000 đ

Lương

1.500.000 đ

Nhận lương sớm

Thứ Sáu, 05 tháng 5 2023

-90.000 đ

Ăn uống

90.000 đ

an toi

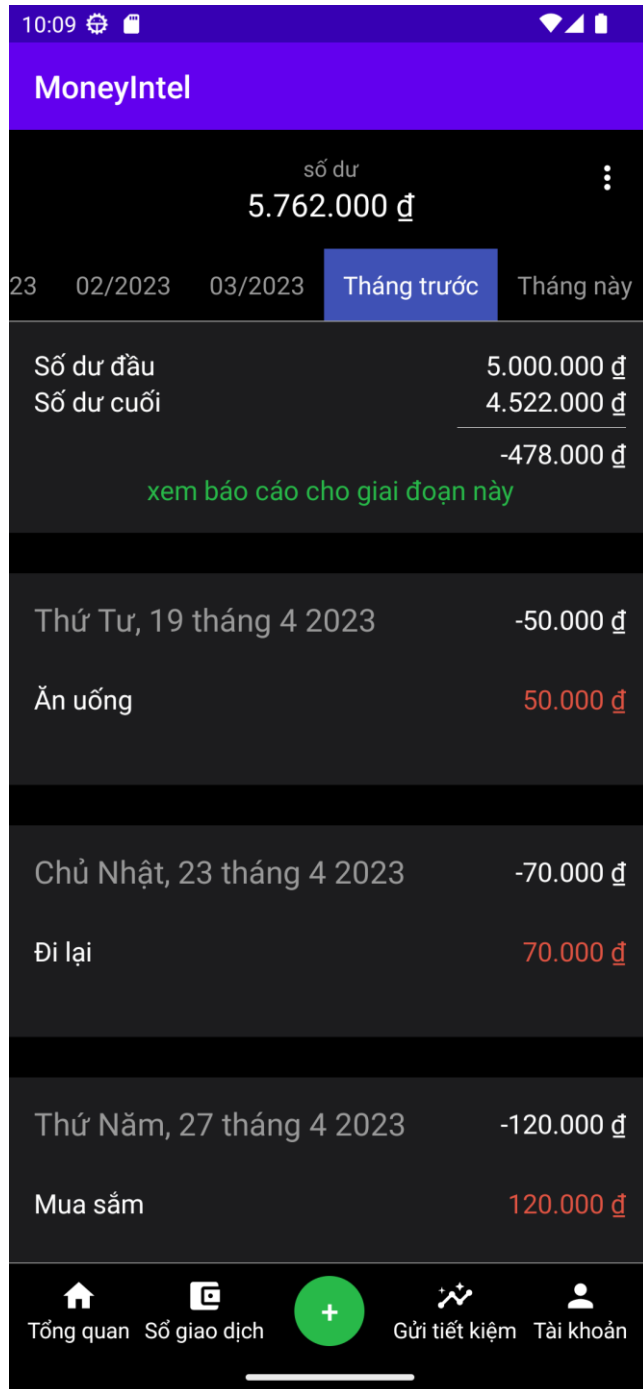
Tổng quan

Sổ giao dịch

+

Gửi tiết kiệm

Tài khoản



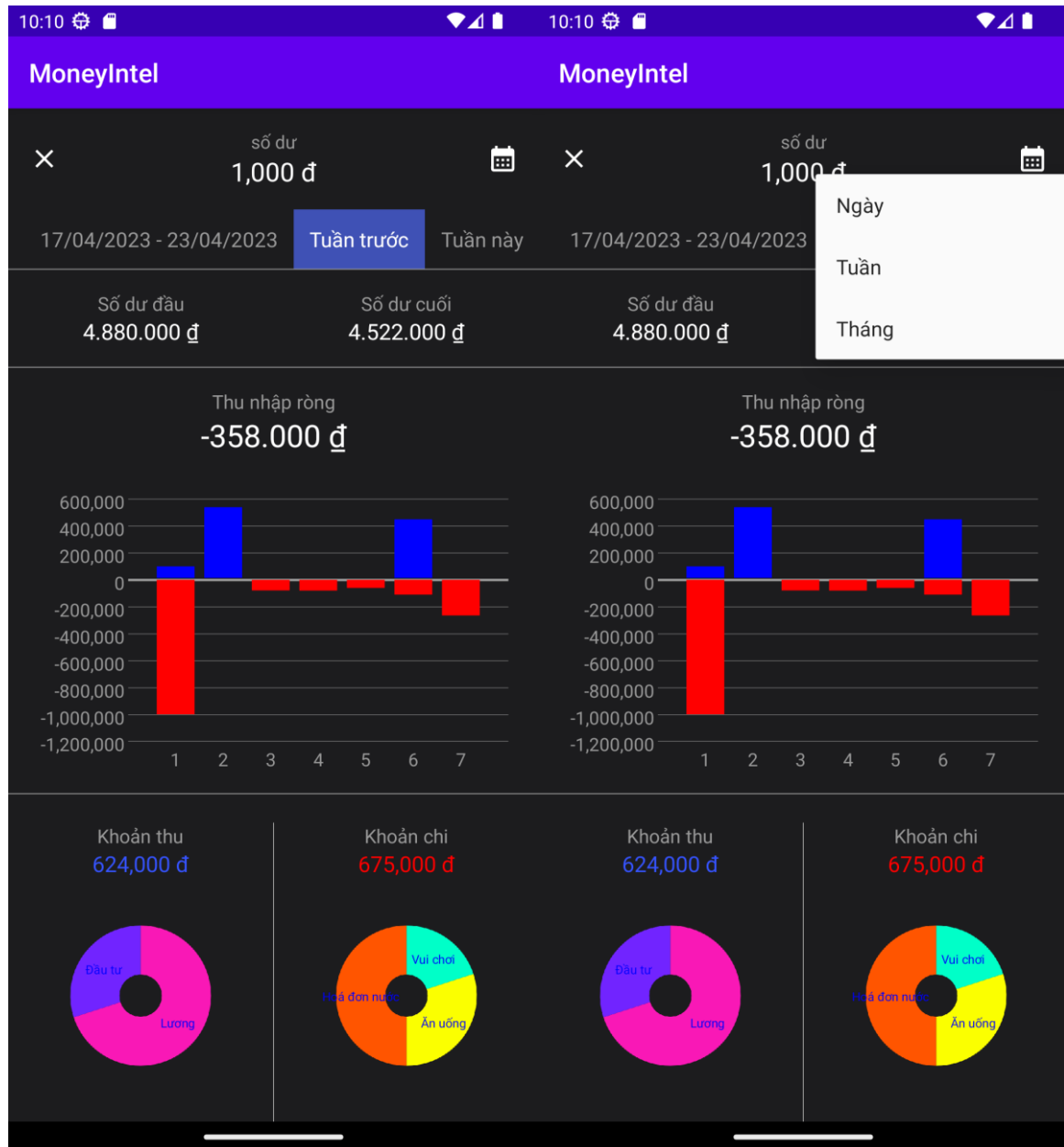
- Thêm giao dịch

The image displays two screenshots of the MoneyIntel mobile application interface for adding a transaction.

Left Screenshot (10:05): The screen is titled "Thêm Giao Dịch" (Add Transaction). It features a purple header with the "MoneyIntel" logo. Below the header, there are three buttons: "Huỷ" (Cancel), "Thêm Giao Dịch", and "Lưu" (Save). The main content area includes four input fields: "Số tiền" (Amount) with a "VND" currency selector and a value of "0"; "Chọn nhóm" (Select category) with a right arrow; "Ghi chú" (Note) with a right arrow; and "Thứ Sáu, 05 tháng 5 2023" (Date) with a right arrow.

Right Screenshot (10:06): This screen shows the "Chọn nhóm" (Select category) modal. It has a purple header with the "MoneyIntel" logo and a back arrow. The main content area lists four categories: "Ăn uống" (Food & Drink), "Đi lại" (Travel), "Mua sắm" (Shopping), and "Lương" (Salary).

- Xem báo cáo



4.2 Cách giải quyết vấn đề

Tham khảo tài liệu từ nhiều nguồn: youtube, chatgpt, android

CHƯƠNG 5 – KẾT LUẬN

5.1 Đạt được

- Thêm giao dịch
- Liệt kê giao dịch gần đây
- Xem thống kê chi tiêu theo tuần, thán
- Xem lịch sử giao dịch theo tuần, tháng
- Xem báo cáo cho các giai đoạn tuần, tháng

5.2 Chưa làm được

- Thêm nhiều loại ví
- Xoá, sửa giao dịch
- Điều chỉnh thông tin tài khoản
- Xem thống kê theo loại chi tiêu

5.3 Hướng phát triển

- Mở rộng chức năng gửi tiết kiệm cho phép khách hàng gửi tiền vào app
- Mở rộng khoảng thời gian thống kê nếu khách hàng đăng ký VIP

CHƯƠNG 6 – ĐÓNG GÓP CỦA THÀNH VIÊN

Lý Tuấn An	75%	Code giao diện và xử lí + báo cáo chương 4, 5
Prean Mesa	25%	Làm báo cáo chương 1,2,3 Làm slide power point

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. <https://viblo.asia/p/su-dung-recyclerview-trong-android-RQqKLNe6l7z>
2. <https://developer.android.com/topic/architecture?hl=vi>
3. <https://hiepsiit.com/detail/android/laptrinhandroid/intent>