

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ THU CHI CÁ
NHÂN

Giảng viên hướng dẫn : Ths. TRẦN THỊ DUNG
Sinh viên thực hiện : NGUYỄN QUANG PHƯỚC
Lớp : Công Nghệ Thông Tin
Khóa : 59

Thành phố Hồ Chí Minh, tháng 7 năm 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ THU CHI CÁ
NHÂN

Giảng viên hướng dẫn : ThS. TRẦN THỊ DUNG
Sinh viên thực hiện : NGUYỄN QUANG PHƯỚC
Lớp : Công Nghệ Thông Tin
Khóa : 59

Thành phố Hồ Chí Minh, tháng 7 năm 2022

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

Mã sinh viên: 5951071080

Họ tên SV: NGUYỄN QUANG PHƯỚC

Khóa: 59

Lớp: Công nghệ thông tin

Tên đề tài

XÂY DỰNG ỨNG DỤNG QUẢN LÝ THU CHI CÁ NHÂN

Mục đích, yêu cầu

2.1. Mục đích :

Quản lý tài chính cá nhân sẽ giúp chúng ta hiểu về dòng tiền và nhận thức rõ về tình hình tài chính của mình hơn. Nhờ đó, chúng ta sẽ biết được mình có cần thêm nguồn thu nhập hoặc phải giảm chi tiêu hay không, hoặc khoản đầu tư nào phù hợp,...Chúng ta sẽ kiểm soát được cách thức hoạt động của dòng tiền của mình.

Bên cạnh thu nhập từ việc đi làm kiếm tiền, hàng tháng chúng ta còn có những khoản chi tiêu. Do đó, để đảm bảo cân bằng về mặt tài chính, chi tiêu hợp lý và có thể tiết kiệm từ thu nhập, chúng ta nên biết cách quản lý tài chính cá nhân sao cho thật hiệu quả.

2.2. Yêu cầu :

- Yêu cầu công nghệ:

- Sử dụng ngôn ngữ lập trình Dart.
- Sử dụng Framework Flutter.
- Sử dụng Android Studio để lập trình.

- Yêu cầu chức năng:

- Ứng dụng bảo mật bằng vân tay, pinCode.
- Thống kê thu chi chi tiết từng tháng.
- Chuyển đổi đơn vị tiền tệ.
- Thống kê thu chi 3 tháng gần nhất.
- Nhập và thay đổi tên người dùng.
- Thêm, Xóa dữ liệu thu chi.
- Đề xuất những ứng dụng tiết kiệm thu chi.

- Yêu cầu phi chức năng:

- Tốc độ truy cập nhanh.
- Giao diện đơn giản phù hợp với người dùng.
- Thao tác đơn giản , dễ dàng sử dụng.

Nội dung và phạm vi đề tài

3.1. Nội dung:

- Tổng quan về bài toán yêu cầu.
- Khảo sát bài toán.
- Phân tích và thiết kế hệ thống theo yêu cầu.
- Xây dựng ứng dụng mobile phục vụ và hỗ trợ cho các cá nhân về việc sử dụng nguồn tiền và làm chủ được dòng tiền của mình đi về đâu.
- Kiểm thử và ứng dụng thực nghiệm.

3.2. Phạm vi đề tài:

- Nghiên cứu vận dụng kiến thức đã được học và kinh nghiệm tham gia phát triển ứng dụng di động.
- Phát triển ứng dụng phù hợp với khả năng bên cạnh sử dụng công nghệ mới bên ngoài trường lớp.

- Nghiên cứu ngôn ngữ Dart, Framework Flutter, SharedPreferences, Hive... và một số thư viện hỗ trợ liên quan.

Công nghệ, công cụ và ngôn ngữ lập trình

4.1. Công nghệ:

- Framework Flutter.

4.2. Công cụ:

- Android studio..

4.3. Ngôn ngữ lập trình:

- Dart.

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

5.1. Kết quả dự kiến đạt được:

- Ứng dụng bảo mật bằng vân tay, pinCode.
- Thống kê thu chi chi tiết từng tháng.
- Chuyển đổi đơn vị tiền tệ.
- Tính phần trăm những mục thu chi.
- Biểu đồ hiển thị các khoản thu chi từng tháng.
- Biểu đồ hiển thị thu chi trong 3 tháng gần nhất.
- Thống kê thu chi 3 tháng gần nhất.
- Nhập và thay đổi tên người dùng.
- Thêm, Xóa dữ liệu thu chi.
- Đề xuất những ứng dụng tiết kiệm thu chi.

5.2. Ứng dụng:

- Hỗ trợ mọi người kiểm soát , hiểu rõ dòng tiền của mỗi cá nhân đã thu chi như thế nào.

Giáo viên và cán bộ hướng dẫn

Họ tên : ThS. Trần Thị Dung

Đơn vị công tác: Bộ Môn CNTT

Điện thoại: 0388389579

Email: ttdung@utc2.edu.vn

Ngày tháng 07 năm 2021

Đã giao nhiệm vụ TKTN

Trưởng BM Công nghệ Thông tin

Giảng viên hướng dẫn

ThS. Trần Phong Nhã

ThS. Trần Thị Dung

Đã nhận nhiệm vụ TKTN

Sinh viên: Nguyễn Quang Phước

Ký tên:

Điện thoại: 0966705629

Email: pnhesi@gmail.com

LỜI CẢM ƠN

Lời đầu tiên em xin cảm ơn Th.s Trần Thị Dung đã giúp đỡ và hướng dẫn nhiệt tình cho em trong suốt thời gian em làm Báo Cáo Tốt Nghiệp này. Tiếp theo em xin cảm ơn tất cả quý Thầy Cô trường Đại Học Giao Thông Vận Tải phân hiệu tại TP.HCM đã giảng dạy và truyền cảm hứng đến tất cả những sinh viên như em, để chúng em có thêm nhiều kiến thức, mở mang thêm cách tư duy, sáng tạo trong cuộc sống.

Em rất mong nhận được lời khuyên, lời góp ý chân thành từ quý Thầy Cô để em tự hoàn thành hơn nữa vốn kiến thức của mình.

Với những kiến thức còn thiếu sót, những kỹ năng còn hạn hẹp, cũng như em đã dành hết thời gian và tâm huyết của mình để hoàn thành Báo Cáo Tốt Nghiệp, em không thể tránh được những sai lầm, em mong Thầy Cô có thể thông cảm và bỏ qua cho em.

Lời cuối cùng em xin chúc quý Thầy Cô phân hiệu cũng như Bộ Môn Công Nghệ Thông Tin và đặc biệt là Cô Th.s Trần Thị Dung có thật nhiều niềm vui, sức khỏe và luôn thành công trong công việc.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp.Hồ Chí Minh , ngày ... tháng 07 năm 2021

Giáo viên hướng dẫn

Trần Thị Dung

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP	i
Tên đề tài.....	i
Mục đích, yêu cầu.....	i
2.1. Mục đích :.....	i
2.2. Yêu cầu :.....	i
Nội dung và phạm vi đề tài	ii
3.1. Nội dung:.....	ii
3.2. Phạm vi đề tài:	ii
Công nghệ, công cụ và ngôn ngữ lập trình	iii
4.1. Công nghệ:	iii
4.2. Công cụ:	iii
4.3. Ngôn ngữ lập trình:	iii
Các kết quả chính dự kiến sẽ đạt được và ứng dụng	iii
5.1. Kết quả dự kiến đạt được:	iii
5.2. Ứng dụng:.....	iii
Giáo viên và cán bộ hướng dẫn	iv
LỜI CẢM ƠN	v
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....	vi
MỤC LỤC	vii
DANH MỤC CÁC CHỮ VIẾT TẮT	x
DANH MỤC CÁC BẢNG, BIỂU ĐỒ, HÌNH VẼ.....	xi
CHƯƠNG 1: MỞ ĐẦU.....	1

1.1 Tổng quan đề tài :	1
1.2 Mục tiêu nghiên cứu:	1
1.3 Phạm vi:	1
1.4 Cấu trúc báo cáo tốt nghiệp:	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	3
2.1 Giới thiệu về Flutter:	3
2.1.1 Flutter là gì ?	3
2.1.2 Cấu trúc của Flutter	4
2.1.3 Ưu điểm và nhược điểm	5
2.2 Ngôn ngữ lập trình Flutter – Dart:	7
2.2.1 Dart là gì?	7
2.2.2 Tại sao lại chọn Dart?	8
2.2.3 Những ưu điểm của Dart?	8
2.3 SharedPreferences	9
2.3.1 SharedPreferences là gì ?	9
2.3.2 Trường hợp nên sử dụng SharedPreferences?	9
2.4 State Manager: Quản lý trạng thái	9
2.4.1 Quản lý trạng thái GetX	10
2.4.2 Quản lý trạng thái Bloc	11
2.5 Local_auth: Bảo mật ứng dụng	13
2.5.1 Mục đích sử dụng:	13
2.5.2 Nguyên lý hoạt động:	13
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	15
3.1 Sơ lược về hệ thống:	15

3.2 Sơ đồ phân rã chức năng:	16
3.3 Sơ đồ UseCase tổng quát:.....	17
3.4 Sơ đồ hoạt động	19
3.4.1 Sơ đồ hoạt động bảo mật.....	19
3.4.2 Sơ đồ hoạt động trang chủ.....	20
3.4.3 Sơ đồ hoạt động cài đặt.....	21
CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG.....	22
4.1 Giao diện bảo mật	22
4.2 Giao diện trang chủ.....	24
4.3 Giao diện thêm tên người dùng	28
4.4 Giao diện thống kê 3 tháng gần nhất	29
4.5 Giao diện thêm dữ liệu thu chi	31
4.6 Giao diện cài đặt	32
4.7 Giao diện những SnackBarMessage trong ứng dụng	37
4.8 Giao diện thống kê chi tiết dữ liệu thu hoặc chi.....	41
KẾT LUẬN VÀ KIẾN NGHỊ.....	43
1. Kết quả	43
1.1 Kết quả đạt được:.....	43
1.2 Kết quả chưa đạt được:	43
2. Hướng phát triển	43
PHỤ LỤC	44
Hướng dẫn cài đặt.....	44
TÀI LIỆU THAM KHẢO	45

DANH MỤC CÁC CHỮ VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	UI	User Interface	
2	AI	Artificial intelligence	
3	OOP	Object-oriented programming	
4	SQLite	SQL database engine	

DANH MỤC CÁC BẢNG, BIỂU ĐỒ, HÌNH VẼ

Hình 2.1 Logo Flutter.....	3
Hình 2.2 Cấu trúc Flutter.....	4
Hình 2.3 Logo Dart	7
Hình 2.4 Logo GetX.....	10
Hình 2.5 Logo Bloc.....	11
Hình 3.1 Sơ đồ phân rã chức năng.....	17
Hình 3.2 Sơ đồ Usecase	18
Hình 3.3 Sơ đồ hoạt động bảo mật	19
Hình 3.4 Sơ đồ hoạt động trang chủ	20
Hình 3.5 Sơ đồ hoạt động cài đặt	21
Hình 4.1 Khóa ứng dụng.	22
Hình 4.2 Đăng nhập pinCode hoặc vân tay.....	23
Hình 4.3 Trang chủ khi chưa có dữ liệu.....	24
Hình 4.4 Trang chủ	25
Hình 4.5 Dữ liệu chi nhỏ hơn 1	26
Hình 4.6 Nhập tên người dùng.	28
Hình 4.7 Thống kê khi chưa có dữ liệu.....	29
Hình 4.8 Thống kê 3 tháng gần nhất.....	30
Hình 4.9 Thêm dữ liệu thu chi.....	31
Hình 4.10 Hiện thị cài đặt	32
Hình 4.11 Dialog thông báo xóa tất cả dữ liệu.....	33
Hình 4.12 Dialog nhập tên mới người dùng.....	34
Hình 4.13 Hiện thị đơn vị tiền tệ	35
Hình 4.14 Dialog thông báo chuyển đơn vị tiền tệ thành công.....	36
Hình 4.15 Thông báo xóa thẻ đang giữ.....	37
Hình 4.16 Thông báo cách xóa thẻ	38
Hình 4.17 Thông báo người dùng nhập không đúng	39
Hình 4.18 Thông báo số tiền bị trống	40

Hình 4.19 Thông báo chưa có dữ liệu thu hoặc chi.....	41
---	----

CHƯƠNG 1: MỞ ĐẦU

1.1 Tổng quan đề tài :

Trong bối cảnh công nghệ thông tin đang chi phối gần như mọi mặt của đời sống, cách quản lý tiền bạc của con người ngày nay, từ kiếm tiền tạo thu nhập đến chi tiêu, cũng đang có nhiều thay đổi so với thế hệ trước.

Với lối suy nghĩ “tuỳ cơ ứng biến” hoặc “đến đâu hay đến đó”, hầu hết chúng ta đều bỏ qua khâu chuẩn bị sẵn các khoản tài chính dự phòng cho những tình huống phát sinh khẩn cấp. Đến khi rủi ro xảy ra, có thể là bệnh tật, thất nghiệp hay tai nạn, bạn không thể trở tay kịp và rơi vào bế tắc, thậm chí là nợ nần. Điều này dẫn đến việc gây áp lực tài chính lên bản thân và cả gia đình.

Trong tương lai Retails Management mong muốn trở thành ứng dụng, công cụ và trở thành người bạn đồng hành với nguồn tiền của mọi người. Cải thiện cách dùng tiền đúng mục đích, tránh đi sự lãng phí không cần thiết.

1.2 Mục tiêu nghiên cứu:

Trong tương lai Retails Managements mong muốn trở thành ứng dụng, công cụ và trở thành người bạn đồng hành với nguồn tiền của mọi người.

Retails Managements sẽ cải thiện cách dùng tiền đúng mục đích, tránh đi sự lãng phí không cần thiết của mọi người.

1.3 Phạm vi:

Nghiên cứu vận dụng kiến thức đã được học và kinh nghiệm tham gia phát triển ứng dụng di động.

Phát triển ứng dụng phù hợp với khả năng bên cạnh sử dụng công nghệ mới bên ngoài trường lớp.

Nghiên cứu ngôn ngữ Dart, Framework Flutter, SharedPreferences, Hive... và một số thư viện hỗ trợ liên quan.

1.4 Cấu trúc báo cáo tốt nghiệp:

Đến với phần mở đầu em sẽ phân tích làm rõ mục đích, nội dung, phạm vi của đề tài.

CHƯƠNG 1: Mở đầu.

Những công nghệ, ngôn ngữ hỗ trợ em trong việc xây dựng ứng dụng Retails Managements.

CHƯƠNG 2: Cơ sở lý thuyết.

Mô tả mọi thứ đã làm được trong ứng dụng thông qua những sơ đồ Erd, sơ đồ hoạt động, sơ đồ phân rã chức năng, sơ đồ Usecase.

CHƯƠNG 3: Phân tích thiết kế hệ thống.

Những hình ảnh thực nghiệm của ứng dụng.

CHƯƠNG 4: Giao diện ứng dụng.

Những kết quả đạt được, những hạn chế cũng như kết quả chưa đạt được trong quá trình xây dựng ứng dụng.

Kết quả và kiến nghị.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Flutter:

2.1.1 Flutter là gì ?

Flutter là bộ công cụ phát triển phần mềm UI framework mã nguồn mở được phát triển bởi Google và phát hành vào tháng 5 năm 2017. Flutter cho phép người dùng tạo ra các giao diện chất lượng cao trên iOS, Android, Linux, Mac, Windows, Web,.. trong khoảng thời gian ngắn. [1]

Flutter sử dụng ngôn ngữ DART cũng do Google phát triển và flutter cũng đã được sử dụng để tạo ra các ứng dụng native cho Google.

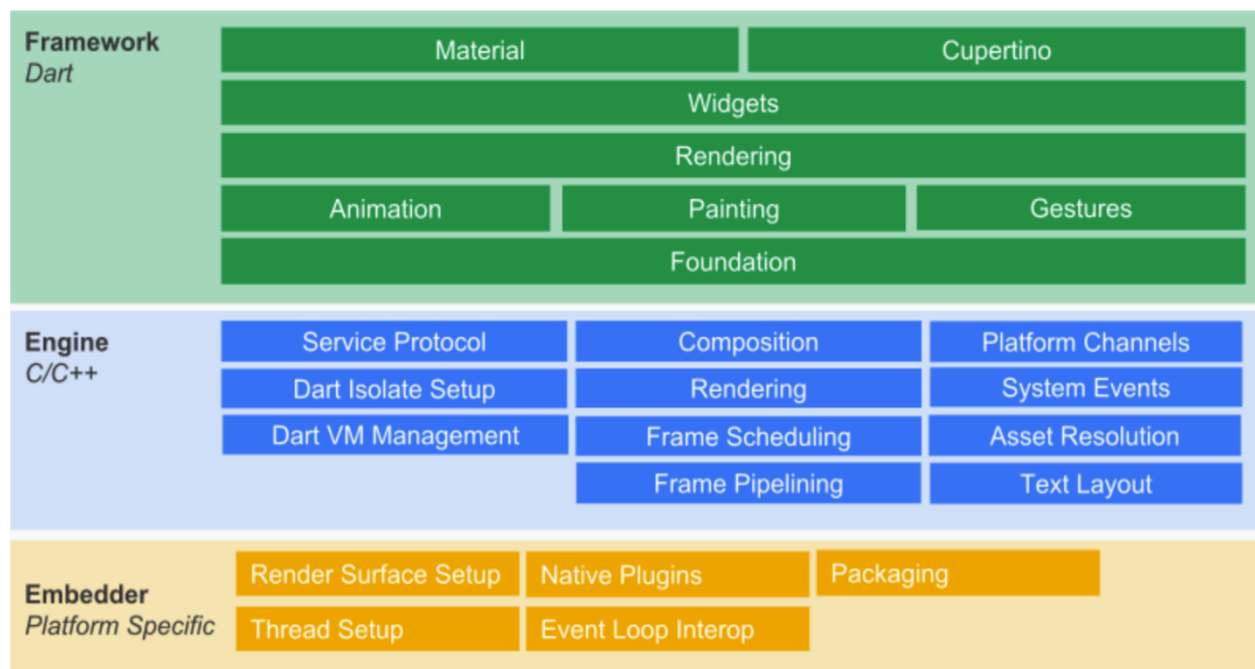


Hình 2.1 Logo Flutter

Flutter hoàn toàn miễn phí và cũng là mã nguồn mở. Flutter có hai thành phần quan trọng :

- Một SDK (Software Development Kit): Một bộ sưu tập các công cụ sẽ giúp chúng ta phát triển các ứng dụng của mình. Điều này bao gồm các công cụ để biên dịch mã của bạn thành mã máy gốc (mã cho iOS và Android).
- Một Framework (UI Library based on widgets): Một tập hợp các thành phần UI có thể tái sử dụng (button, text inputs, slider, v.v.) giúp chúng ta có thể cá nhân hóa tùy theo nhu cầu của riêng mình.

2.1.2 Cấu trúc của Flutter



Hình 2.2 Cấu trúc Flutter

Cấu trúc Flutter chủ yếu bao gồm bốn thành phần:

Flutter Engine: Phương tiện của Flutter.

Foundation Library: Thư viện nền tảng.

Widgets: Các widget.

Design Specific Widgets: Thiết kế những widget riêng.

2.1.2.1. Phương tiện của Flutter

Nó là một thời gian chạy di động cho các ứng dụng di động chất lượng cao và chủ yếu dựa trên ngôn ngữ C++. Nó triển khai các thư viện lõi của Flutter bao gồm hoạt ảnh và đồ họa, I/O tệp và mạng, kiến trúc plugin, hỗ trợ trợ năng và thời gian chạy dart để phát triển, biên dịch và chạy các ứng dụng Flutter. Nó sử dụng thư viện đồ họa mã nguồn mở của Google, Skia, để hiển thị đồ họa cấp thấp.

2.1.2.2. Thư viện nền tảng

Nó chứa tất cả các gói cần thiết cho các khối xây dựng cơ bản để viết một ứng dụng Flutter. Các thư viện này được viết bằng ngôn ngữ Dart.

2.1.2.3. Widgets

Trong Flutter, mọi thứ đều là một widget, đó là khái niệm cốt lõi của framework này. Widget trong Flutter về cơ bản là một thành phần giao diện người dùng ảnh hưởng và kiểm soát chế độ xem và giao diện của ứng dụng. Nó đại diện cho một mô tả bất biến về một phần của giao diện người dùng và bao gồm đồ họa, văn bản, hình dạng và hoạt ảnh được tạo bằng các widget. Các widget tương tự như các thành phần React.

Trong Flutter, ứng dụng tự nó là một widget chứa nhiều widget con. Điều đó có nghĩa là ứng dụng là widget con cấp cao nhất và giao diện người dùng của nó được xây dựng bằng cách sử dụng một hoặc nhiều widget con, bao gồm các widget con phụ. Tính năng này giúp bạn tạo một giao diện người dùng phức tạp rất dễ dàng.

2.1.3 Ưu điểm và nhược điểm

Flutter : Ưu điểm:

- Mạnh về hiệu ứng, hiệu suất ứng dụng rất cao.
- Giao tiếp gần như trực tiếp với hệ thống.
- Ngôn ngữ kiểu tĩnh nhưng với cú pháp hiện đại (tương tự JS, Python, Java), compiler linh động khi dùng AOT (cho sản phẩm cuối) và JIT (cho quá trình phát triển với hot reload).
- Có thể chạy được giả lập mobile ngay trên web, tiện cho việc phát triển. Các bộ đo lường chỉ số hiệu suất được hỗ trợ sẵn giúp lập trình viên kiểm soát tốt hiệu suất của ứng dụng.
- Có thể dùng để xây dựng các nền tảng gắn vào ứng dụng native để tăng hiệu suất.

Flutter : Nhược điểm:

- Bộ render UI được nhóm phát triển gần như viết lại, không liên quan tới UI có sẵn của Framework native, dẫn đến memory sử dụng khá nhiều. Hơn nữa, các UI không đi chung với iOS, mà được phát triển riêng, nghĩa là cùng 1 phiên bản Flutter khi tạo ra ứng dụng cho iOS thì iOS 8.x -> 12.x đều y chang nhau, tương tự như với Android. Nhưng UI của Android thì tất nhiên khác với iOS.
- Phải học thêm ngôn ngữ DART: lập trình viên biết về DART không nhiều, cũng có rủi ro là học xong DART sẽ dính liền luôn với DART ở mảng phát triển ứng dụng mobile. Chứ không uyển chuyển như JS hay Python có thể nhảy qua lại giữa front, back hay AI...
- Là con cưng của Google, tuy nhiên hăng dính nhiều phốt với thói quen “quăng con giữa chợ” nên cũng cần cân nhắc. Tuy nhiên, mình nhận thấy Flutter rất tốt, tốt hơn nhiều so với những cái mà Google từng làm ra như Angular.

Native đúng nghĩa :

- Bộ engine để render UI hoàn toàn thuộc Flutter, vì vậy bạn có thể tạo ra bất kỳ hiệu ứng và đồ họa nào cho cả 2 nền tảng iOS và Android. Bạn có thể tưởng tượng Flutter như 1 engine game, có thể điều chỉnh đến từng pixel và phủ trên đó là hệ thống Widget đa dạng, phong phú.

Hot Reload :

- Khi làm việc với những thứ liên quan đến UI thì Hot Reload thật khiến cho lập trình viên dễ chịu, càng tuyệt hơn khi Flutter cung cấp khả năng ghi nhớ state của ứng dụng. Hot Reload cung cấp cho bạn trải nghiệm tuyệt vời khi lập trình Flutter, so với React Native thì điểm này Flutter hơn.

Dart DevTools :

- DevTools tuy mới ở giai đoạn beta nhưng độ hoàn thiện cũng rất cao, so với DevTools của React/React Native thì DevTools của Flutter tốt hơn.

Nâng cấp dễ dàng :

Nguyễn Quang Phước – K59

- Mặc dù cách cài đặt ban đầu của Flutter là tải về tập tin nén, nhưng khi cập nhật lên phiên bản mới thì chỉ cần dòng lệnh là đủ và khá dễ dàng. Ứng dụng sau khi nâng cấp phiên bản Flutter chạy không gặp trở ngại. Đây là một điểm cộng khá lớn khi xét về độ trưởng thành. Khi sử dụng React Native, các phiên bản từ 0.25, 0.48 có những thay đổi phải sửa bằng tay mới chạy, rất khó chịu.

Cộng đồng học tập phát triển mạnh mẽ:

- Flutter có một cộng đồng mạnh mẽ, nó mới chỉ bắt đầu nhưng hầu hết là cộng đồng cực kỳ chất lượng! Một số website nơi bạn có thể trao đổi thắc mắc của mình: Flutter Awesome, It's all widgets!, Flutter Community.

2.2 Ngôn ngữ lập trình Flutter – Dart:

2.2.1 Dart là gì?

Để lập trình với Flutter, bạn sẽ sử dụng một ngôn ngữ lập trình gọi là Dart. Ngôn ngữ này được tạo bởi Google vào năm 2011, nó đã được cải thiện rất nhiều trong những năm qua.

Dart tập trung vào phát triển front end, và bạn có thể sử dụng nó để tạo ứng dụng di động và cả ứng dụng web.



Hình 2.3 Logo Dart

Một vài thông tin về ngôn ngữ Dart:

- Là ngôn ngữ lập trình đơn giản, dễ hiểu và dễ tiếp cận.
- Là ngôn ngữ lập trình được sử dụng để build lên Flutter Framework.

- Nó ra đời vào năm 2011, nhằm đến việc tạo ra các ứng dụng đa nền tảng – web, mobile, desktop và IoT.
- Là ngôn ngữ tĩnh, theo hướng đối tượng (OOP), functional programming và lexical scoped.
- Dart như 1 sự kết hợp giữa Java và JavaScript nên khi học nó nếu ai đã có nền tảng 1 trong 2 ngôn ngữ kia thì lúc tiếp cận sẽ khá dễ.

2.2.2 Tại sao lại chọn Dart?

Các nhà phát triển tại Google và các nơi khác sử dụng Dart để tạo các ứng dụng chất lượng cao, quan trọng cho iOS, Android và web.

Với các tính năng nhắm đến sự phát triển phía khách hàng, Dart rất phù hợp cho cả ứng dụng di động và web. Dart giúp bạn tạo ra những trải nghiệm đẹp, chất lượng cao trên tất cả các màn hình, với:

- Một ngôn ngữ được tối ưu hóa cho client .
- Framework mạnh mẽ.
- Công cụ linh hoạt.

2.2.3 Những ưu điểm của Dart?

Năng suất Cú pháp Dart rõ ràng và súc tích, công cụ của nó đơn giản nhưng mạnh mẽ. Type-safe giúp bạn xác định sớm các lỗi tinh tế. Dart có các thư viện cốt lõi và một hệ sinh thái gồm hàng ngàn package. [2]

Nhanh Dart cung cấp tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.

Di động Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript.

Dễ gần Dart quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên của nó. Nếu bạn đã biết C ++, C # hoặc Java, bạn có thể làm việc hiệu quả với Dart chỉ sau vài ngày.

Reactive Dart rất phù hợp với lập trình Reactive, với sự hỗ trợ để quản lý các đối tượng tồn tại trong thời gian ngắn, chẳng hạn như các widget UI, thông qua phân bổ đối tượng nhanh. Dart hỗ trợ lập trình không đồng bộ thông qua các tính năng ngôn ngữ và API sử dụng các đối tượng Future và Stream.

2.3 SharedPreferences

2.3.1 SharedPreferences là gì ?

Share Preferences là một cách lưu trữ dữ liệu trong Android. Nó là một interface cho phép lưu trữ dữ liệu bằng cách cặp khóa key và value và nó được lưu dưới dạng file xml trong chính ứng dụng Android luôn nếu bạn xóa app thì nó sẽ mất. [3]

Và các dữ liệu lưu trong SharedPreferences là dạng dữ liệu nguyên thủy như int, float, string, boolean, long.

SharedPreferences chủ yếu sử dụng để lưu các dữ liệu đơn nhỏ lẻ hay các dữ liệu tạm như khi bạn chơi game thì bạn sẽ thấy có điểm số khi chơi và nó sẽ được lưu ở SharedPreferences.

2.3.2 Trường hợp nên sử dụng SharedPreferences?

SharedPreferences có thể thực hiện lưu trữ và truy xuất dữ liệu trong ứng dụng Android tuy nhiên có khá nhiều hạn chế so với SQLite. Vậy SharedPreferences nên được sử dụng trong trường hợp ứng dụng của bạn lưu trữ các dữ liệu nhỏ hoặc dữ liệu không quá phức tạp.

SharedPreferences chỉ hỗ trợ lưu trữ dưới dạng key-value nên nếu áp dụng để truy vấn dữ liệu liên kết giữa các bảng sẽ gây ra khó khăn hơn.

SharedPreferences thường được các lập trình viên sử dụng để lưu trữ các thông tin dạng cấu trúc hình ứng dụng, cài đặt... thay vì SQLite bởi SharedPreferences đơn giản và nhanh hơn nhiều.

2.4 State Manager: Quản lý trạng thái

Quản lý state trong một ứng dụng Flutter là rất quan trọng, việc này ảnh hưởng trực tiếp đến hiệu năng của ứng dụng.

2.4.1 Quản lý trạng thái GetX



Hình 2.4 Logo GetX

2.4.1.1. GetX làm được những gì?

- State Manager : Quản lý state trong Flutter [4]
- Navigation Manager: Quản lý việc điều hướng
- Dependencies Manager: Cung cấp giải pháp dependencies injection tuyệt vời
- Utils function: Các hàm tiện ích cực kỳ hữu ích trong lập trình Flutter

2.4.1.2. GetX – State Management

Ưu điểm:

- BLoC pattern cũng là một cách quản lý state hiệu quả và an toàn. Tuy nhiên code khá rối rắm và tốn nhiều thời gian để implement, trong một số trường hợp thì khó handle. Khó tiếp cận với những người mới.
- MobX cũng là một thư viện tuyệt vời, dễ sử dụng hơn BLoC tuy nhiên lại quá phụ thuộc vào code generation. Có thể gây khó hiểu với người sử dụng và làm phình scope của dự án.

- Provider là một kiểu quản lý state cơ bản và hiệu quả đối với nhiều người. Khi mới bắt đầu mọi người nên sử dụng thuần thục Provider. Tuy nhiên Provider sử dụng InheritedWidget và nó chỉ có thể sử dụng trong widget tree nên nhiều trường hợp sẽ không thể handle được.

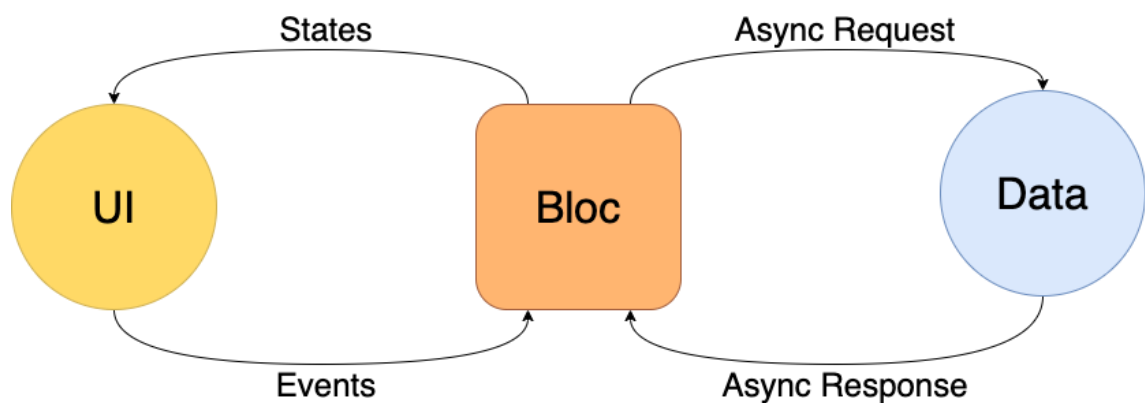
Tiếp theo là những ưu điểm của GetX - State manager:

- Chỉ update những widget cần thiết.
- Sử dụng ít bộ nhớ hơn so với các kiểu quản lý state khác.
- Quên đi StatefulWidget. Với Get các bạn không phải suy nghĩ sử dụng StatefulWidget hay StatelessWidget nữa. Bây giờ bạn chỉ việc một component duy nhất là GetWidget.
- Việc tổ chức cấu trúc project sẽ cực kỳ clear, phần code logic được tách hẳn hoàn toàn so với UI.
- Update widgets without spending ram for that.
- Tối ưu hoá bộ nhớ, bạn sẽ không phải lo lắng việc Out Memory nữa , Get sẽ tự động thu dọn những component không cần thiết.

Nguyên lý hoạt động

Về cơ bản Get hoạt động cũng giống như Rx, cũng có các Observable và các component để lắng nghe thay đổi của Observable.

2.4.2 Quản lý trạng thái Bloc



Hình 2.5 Logo Bloc

2.4.2.1. Bloc làm được gì?

Bloc sẽ nhận các event mà UI gửi qua, sau đó xử lý logic (get data, update data, ...) rồi trả về state cho UI. Dựa vào state nhận được từ Bloc, UI sẽ render lại những view cần

2.4.2.2. Bloc – State Management

Ưu điểm:

Biết ứng dụng của chúng ta đang ở trạng thái nào tại bất kỳ thời điểm nào. [5]

Dễ dàng kiểm tra mọi trường hợp để đảm bảo ứng dụng của chúng ta đang phản hồi thích hợp.

Ghi lại mọi tương tác của người dùng trong ứng dụng của chúng ta để chúng ta có thể đưa ra quyết định dựa trên dữ liệu.

Làm việc hiệu quả nhất có thể và sử dụng lại các thành phần cả trong ứng dụng của chúng ta và trên các ứng dụng khác.

Nhiều người tham gia phát triển cùng một ứng dụng có thể làm việc một cách dễ dàng.

Nguyên lý hoạt động:

Truyền event vào bloc, bloc xử lý và cho ra output là State của UI.

Input sẽ được thêm vào sink của StreamController và phía UI sẽ sử dụng stream để lắng nghe nhận state mỗi khi có event được add vào sink.

StreamController đơn giản chỉ là 1 controller bên trong có 1 stream và controller này giúp ta điều khiển stream đó dễ dàng. Chúng ta có thể thoải mái push các events vào stream, lắng nghe nhận data từ stream đó. Nói cách khác, đây là một con đường khác giúp ta tạo ra 1 Stream và thoải mái emit events và nhận events bất cứ thời điểm nào chúng ta muốn.

Trong mỗi StreamController đều có 2 đối tượng là sink(giúp chúng ta push events đến stream) và stream (giúp chúng ta nhận events từ sink).

2.5 Local_auth: Bảo mật ứng dụng

2.5.1 Mục đích sử dụng:

Khai thác tính độc nhất về cấu tạo hình dạng vân tay của mỗi người, các nhà sinh trắc học sẽ biến nó thành chiếc chìa khoá riêng mà chỉ bạn mới có thể sử dụng, giúp bạn tránh được nhiều phiền toái trong cuộc sống như bị trộm cắp, tránh khỏi sự truy cập trái phép, bảo vệ thông tin cá nhân của mình.

Chức năng vân tay cung cấp các phương tiện để thực hiện xác thực cục bộ, trên thiết bị của người dùng.

2.5.2 Nguyên lý hoạt động:

Nguyên lý hoạt động của Công nghệ nhận dạng vân tay: Khi đặt ngón tay lên trên một thiết bị nhận dạng dấu vân tay, ngay lập tức thiết bị này sẽ quét hình ảnh ngón tay đó và đối chiếu các đặc điểm của ngón tay đó với dữ liệu đã được lưu trữ trong hệ thống. Quá trình xử lý dữ liệu sẽ được thiết bị chuyển sang các dữ liệu số(1010) và ra thông báo rằng dấu vân tay đó là hợp lệ hay không hợp lệ để cho phép hệ thống thực hiện các chức năng tiếp theo. [6]

5.2.1. Ưu điểm:

Có khả năng cải thiện tính bảo mật, kiểm soát truy cập an toàn

Là một giải pháp bảo mật hiện đại và phức tạp nhất, có độ chính xác gần như là tuyệt đối trong quá trình xác thực.

So với phương pháp xác thực truyền thống, bảo mật sinh trắc học có thao tác thực hiện nhanh hơn, người dùng hạn chế tình trạng quên chuỗi mật khẩu dài và phức tạp như trước kia .

Khắc phục hiện tượng quá tải thông tin đăng nhập trên các ứng dụng hoặc thiết bị khác nhau(dùng 1 mật khẩu cho nhiều ứng dụng bảo mật).

Xác thực sinh trắc học có tính linh hoạt, dễ đăng ký và triển khai sử dụng .

5.2.2. Nhược điểm:

Khi tay thấm nước sử dụng vân tay rất khó chính xác.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Sơ lược về hệ thống:

Quản lý tài chính cá nhân là một bài toán khó mà không phải ai cũng được học ở trường lớp, thế nhưng ai cũng phải đương đầu với nó trên đường đời. Vậy làm thế nào để chi tiêu hiệu quả?

Đại đa số người Việt chưa được trang bị kiến thức quản lý tài chính cá nhân bài bản như ở các nước phát triển. Thay vì cần phải quản lý tài chính một cách khoa học bằng các phương pháp chuyên môn, đa số chúng ta thường phân bổ nguồn tiền dựa vào cảm hứng. Chẳng hạn như: hôm nay có nhiều tiền nên bỏ tiết kiệm, hôm sau lại rút ra để sắm cho mình chiếc điện thoại mới...

Nhiều người thường lấy lý do tiền tiêu tháng này còn chưa đủ thì làm sao có tiền dự phòng. Nhưng nhìn vào thực tế, sẽ có những tình huống xảy ra mà chúng ta không thể nào dự đoán trước được. Không ai có thể biết đại dịch Covid xảy ra khiến hàng triệu người thất nghiệp, giảm lương. Không ai có thể đảm bảo một ngày nào đó bạn không thất nghiệp, bạn không bị mất món đồ quan trọng, bệnh tật xảy ra. Nếu lúc đó bạn không có quỹ dự phòng cho mình thì làm sao có thể xoay xở cho tình huống đó?

Kỹ năng quản lý tài chính cá nhân không chỉ là sử dụng hiệu quả số tiền tiêu hàng tháng mà còn là biết dự phòng cho các trường hợp tương lai.

Nhiều người có tiền là tiêu, không biết nên ưu tiên tiêu gì trước sau. Hậu quả là tiền bị sử dụng sai mục đích và chúng ta không tiết kiệm, tích lũy được bao nhiêu.

Nhiều người có thói quen sau khi nhận lương sẽ tiêu sài xả láng, sau đó gần cuối tháng mới cắt xén tiết kiệm.

Trong tương lai Retails Management mong muốn trở thành ứng dụng, công cụ và trở thành người bạn đồng hành với nguồn tiền của mọi người. Cải thiện cách dùng tiền đúng mục đích, tránh đi sự lãng phí không cần thiết.

3.2 Sơ đồ phân rã chức năng:

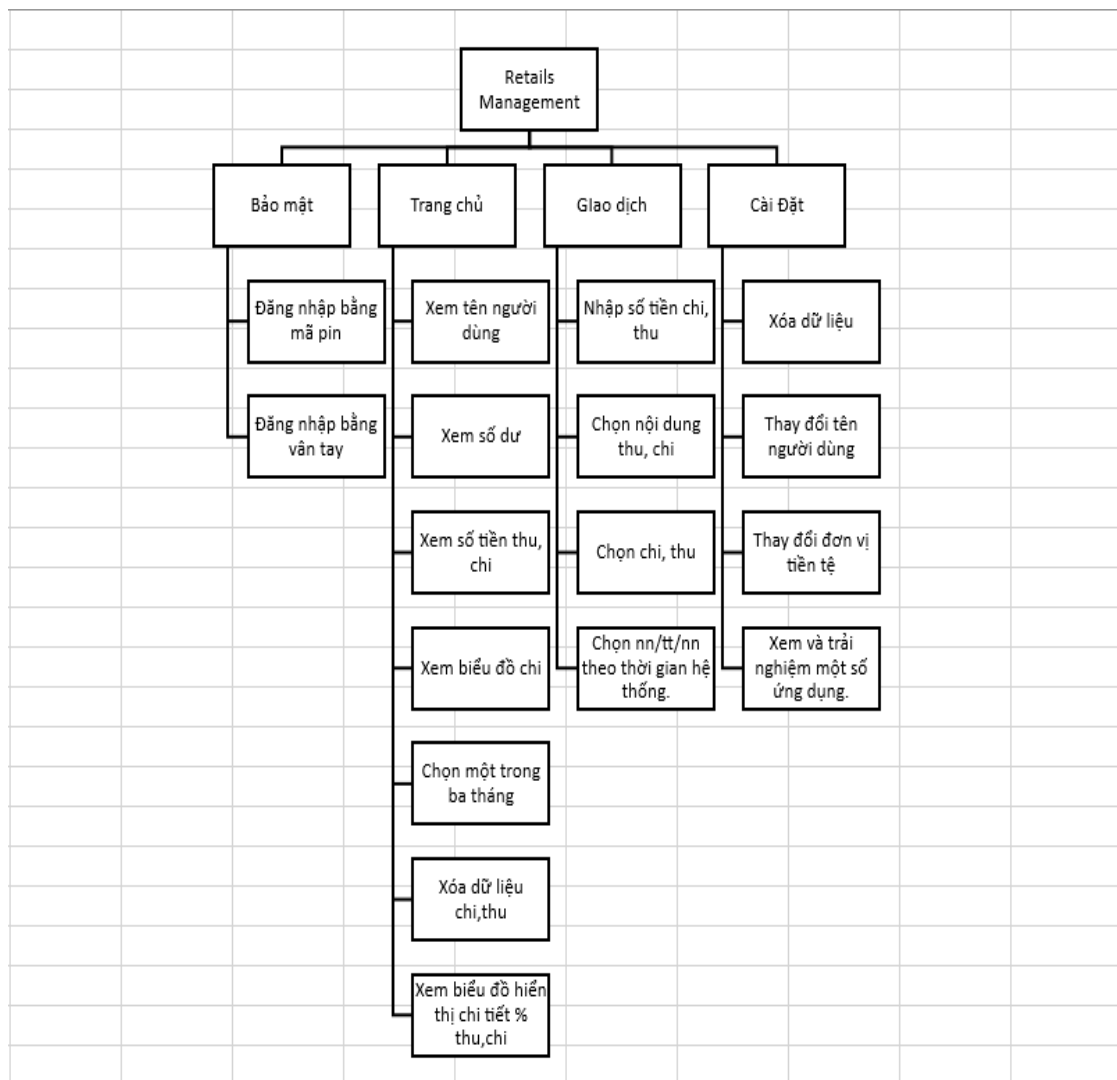
Với những yêu cầu và chức năng được chọn, Retails Management sẽ được xây dựng và ứng dụng vào thực tế để kiểm thử và để hiểu rõ hơn Retails Management làm gì thì chúng ta hãy thông qua vào một số chức năng. Retails Management có 4 mục chức năng chính được thể hiện với sơ đồ phân rã chức năng bên dưới:

Chức năng bảo mật: Nhiều cá nhân muốn có quyền riêng tư của mình về vấn đề tài chính, để đáp ứng như câu trên chúng tôi yêu cầu người dùng khi sử dụng ứng dụng phải cần đăng nhập thông qua 2 cách: đăng nhập bằng mã pin hoặc đăng nhập bằng vân tay có trên thiết bị hệ thống.

Chức năng trang chủ: Trang chủ sẽ hiển thị tên người dùng được nhập vào khi qua bước đăng nhập. Hiển thị biểu đồ số tiền đã chi để cho người dùng thấy rõ những con số mà người dùng đã sử dụng vào những việc gì. Cuối màn hình sẽ hiển thị phần thẻ chứa chi tiết những thông tin số tiền đã chi, thu vào việc gì.

Chức năng giao dịch: Người dùng phải nhập dữ liệu(số tiền,nội dung, ngày sử dụng) đã thu(đã chi) để lấy dữ liệu hiển thị lên trang chủ.Nếu người dùng nhập sai kiểu dữ liệu hoặc không nhập số tiền thì ứng dụng sẽ thông báo yêu cầu với người dùng.

Chức năng cài đặt: Bổ sung những ứng dụng hỗ trợ cho việc quản lý , sử dụng đồng tiền, vì mục đích của mỗi cá nhân muốn làm mới bản thân, để đáp ứng điều đó chúng tôi sẽ cho người dùng thay đổi tên, thay đổi đơn vị tiền tệ, xóa hết dữ liệu đã nhập.



Hình 3.1 Sơ đồ phân rã chức năng

3.3 Sơ đồ UseCase tổng quát:

Để mô tả rõ hơn về các đối tượng và chức năng trong hệ thống, em đã vẽ sơ đồ usecase tổng quát cho các chức năng chính.

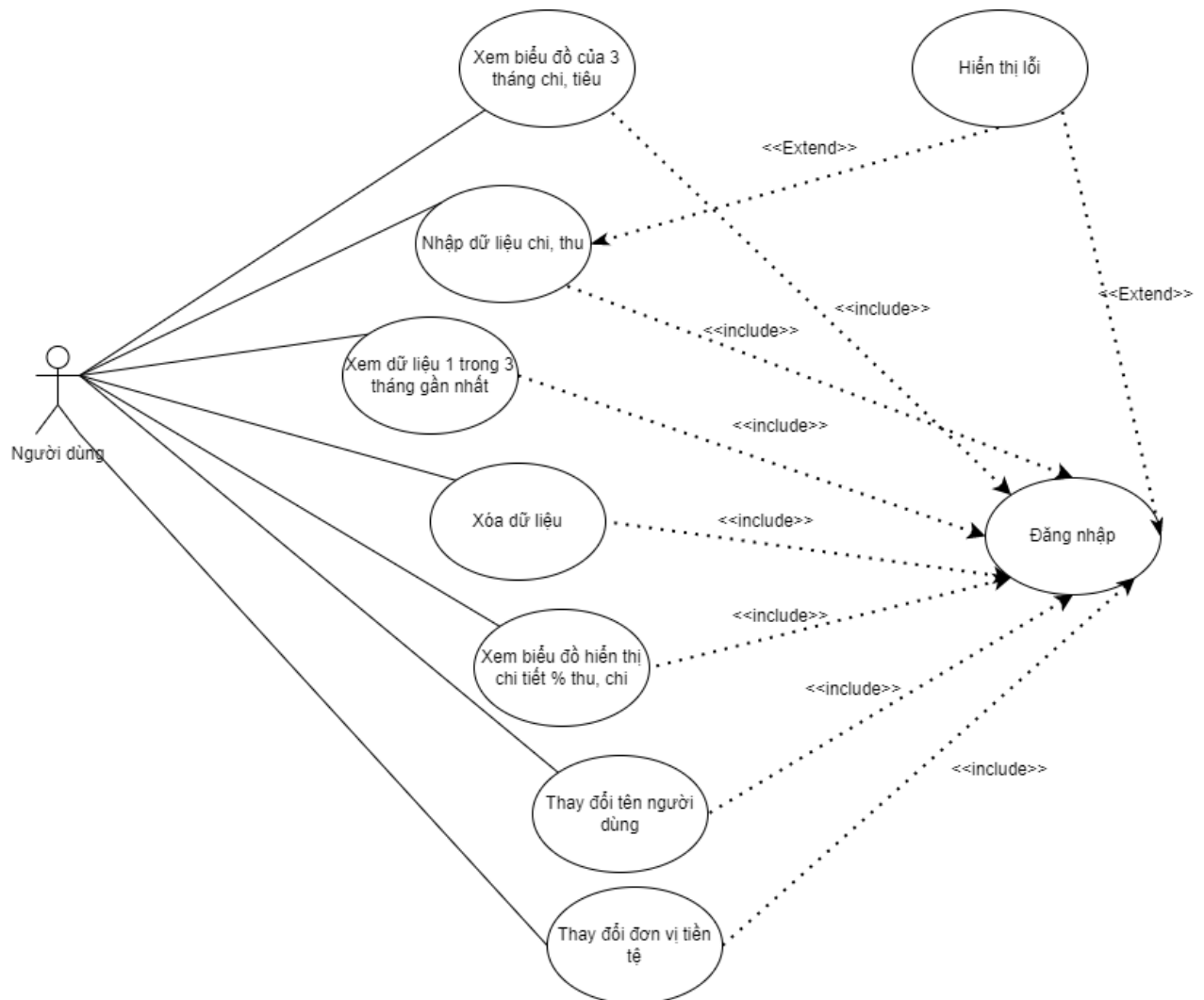
Khi người dùng đăng nhập thành công có thể xem biểu đồ của 3 tháng chi tiêu, nhập dữ liệu chi và thu, xem dữ liệu 1 trong 3 tháng gần nhất, xóa dữ liệu, xem biểu đồ chi tiết % thu chi, thay đổi tên người dùng, thay đổi đơn vị tiền tệ.

Có hai lỗi khi người dùng nhập không đúng:

- Đăng nhập bằng vân tay hoặc mã pin sai thì hệ thống sẽ thông báo lỗi người dùng đã nhập sai.

- Nhập dữ liệu chi thu không phải là kiểu số hoặc để trống phần số tiền thì sẽ được hệ thống thông báo lỗi.

Dưới đây là sơ đồ Usecase mà đã được ứng dụng :

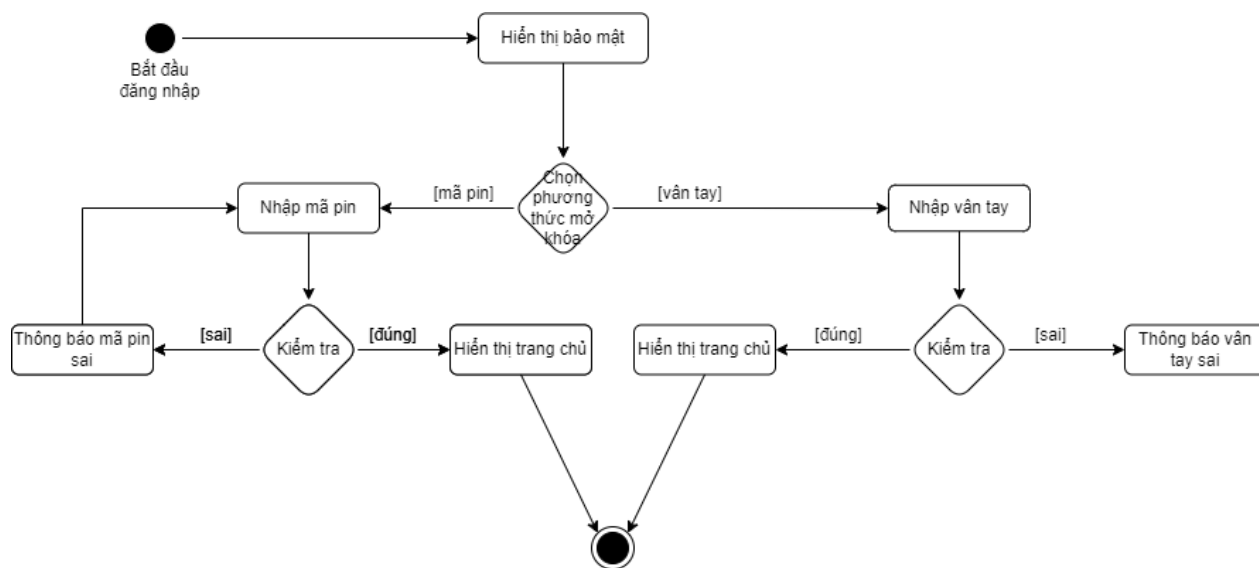


Hình 3.2 Sơ đồ Usecase

3.4 Sơ đồ hoạt động

3.4.1 Sơ đồ hoạt động bảo mật

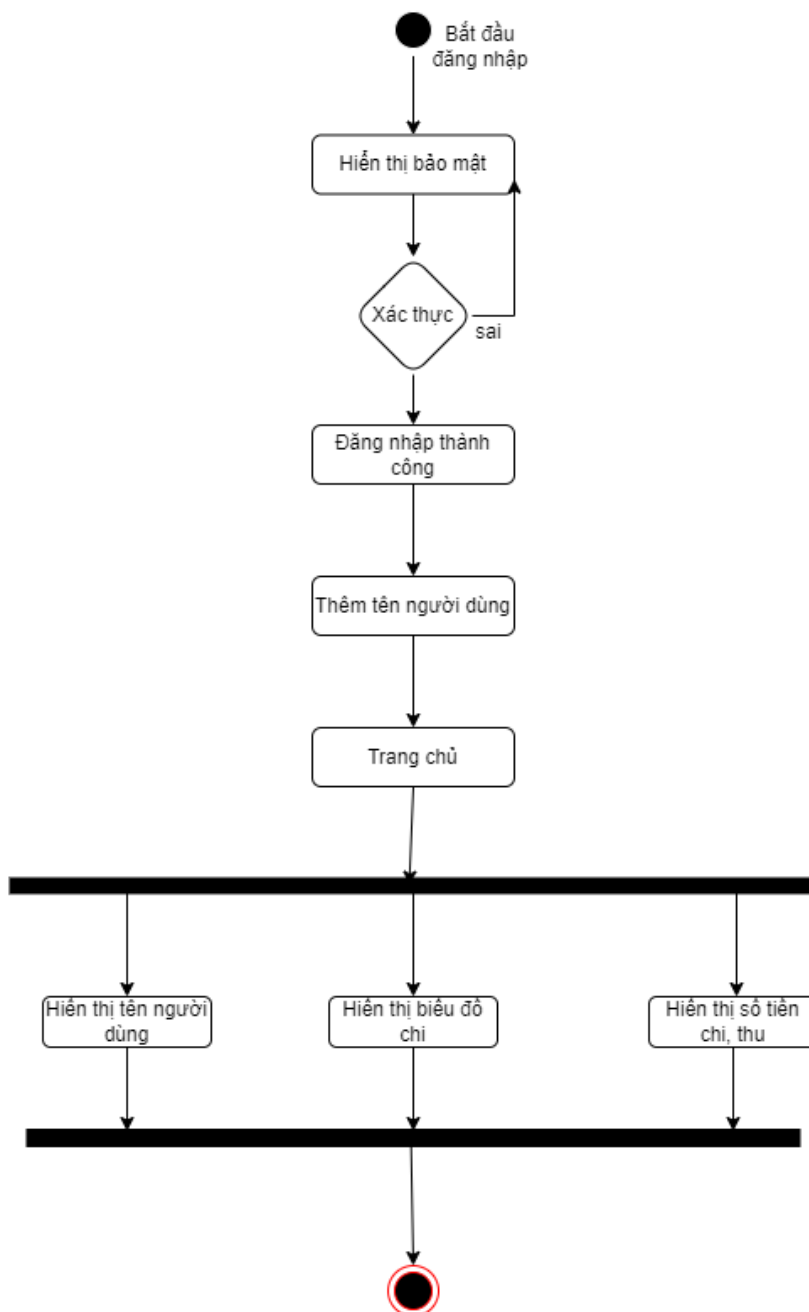
Khi sử dụng ứng dụng người dùng phải nhập mã pin hoặc vân tay để sử dụng ứng dụng. Nếu nhập sai mã pin hoặc vân tay hệ thống sẽ yêu cầu người dùng nhập lại.



Hình 3.3 Sơ đồ hoạt động bảo mật

3.4.2 Sơ đồ hoạt động trang chủ

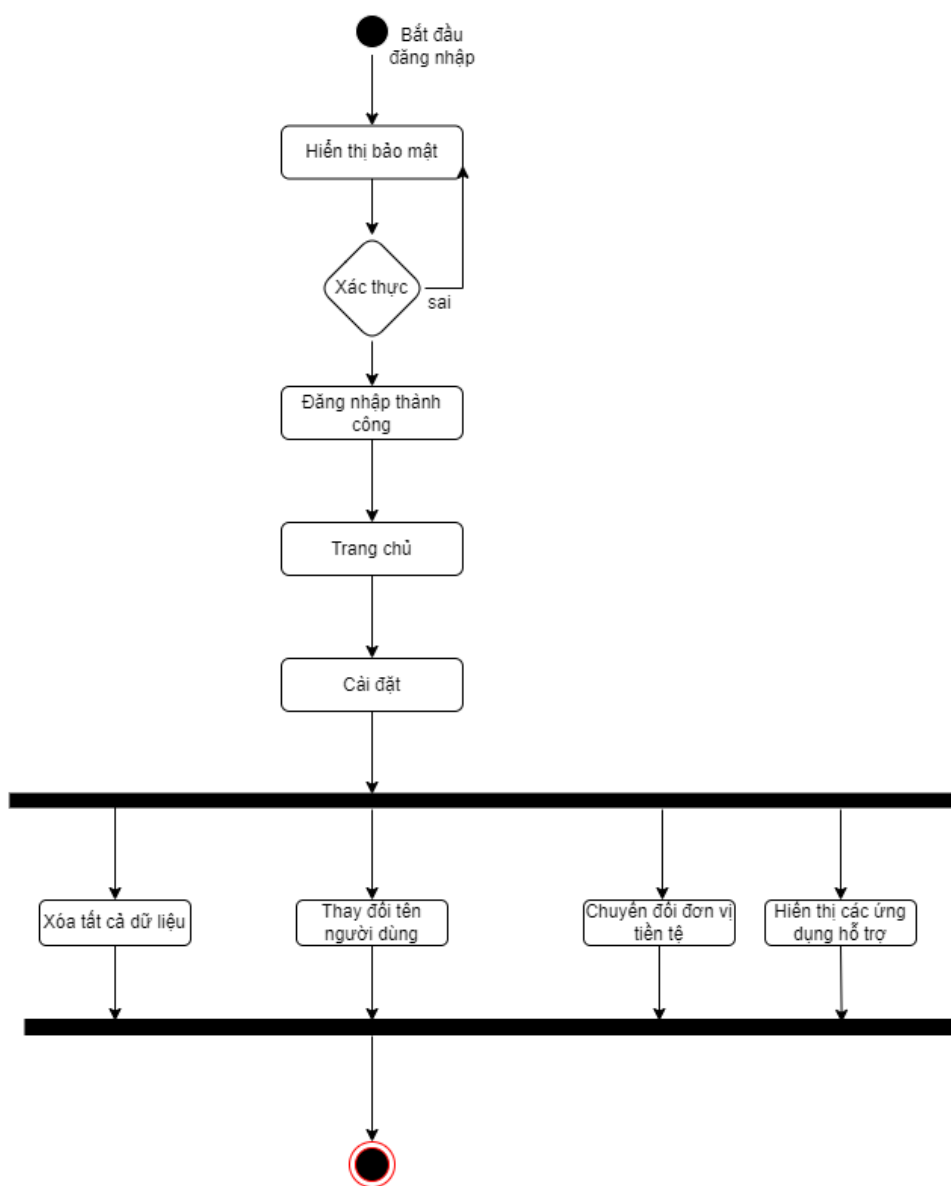
Khi đăng nhập thành công, hệ thống yêu cầu người dùng nhập tên rồi mới được chuyển qua phân trang chủ, tiếp theo phân trang chủ sẽ được hiển thị tên người dùng, hiển thị biểu đồ chi, hiển thị số tiền chi thu .



Hình 3.4 Sơ đồ hoạt động trang chủ

3.4.3 Sơ đồ hoạt động cài đặt

Khi đăng nhập thành công, người dùng có thể sử dụng những chức năng trong phần cài đặt: Xóa tất cả dữ liệu, thay đổi tên người dùng, thay đổi đơn vị tiền tệ, hiển thị một số ứng dụng hỗ trợ.

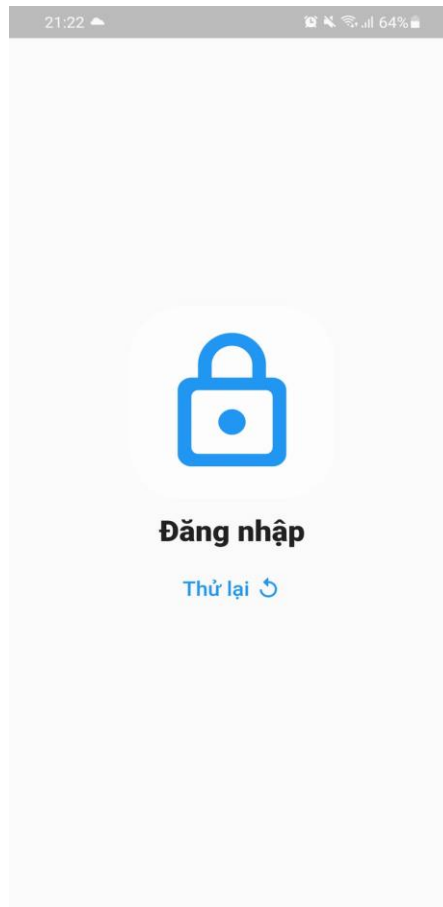


Hình 3.5 Sơ đồ hoạt động cài đặt

CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG

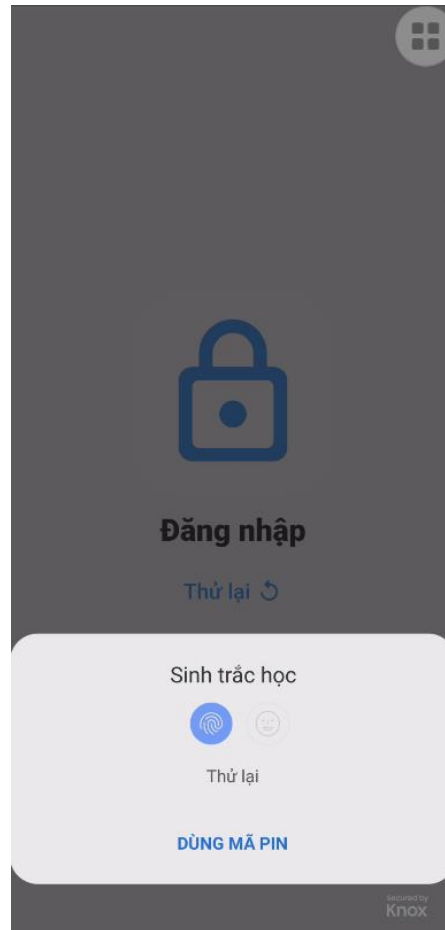
4.1 Giao diện bảo mật

Sau khi click vào ứng dụng sẽ hiển thị và yêu cầu bạn mở khóa bằng cách sử dụng PinCode hoặc vân tay của hệ thống điện thoại.



Hình 4.1 Khóa ứng dụng.

Nhập pinCode hoặc quét vân tay để sử dụng ứng dụng.



Hình 4.2 Đăng nhập pinCode hoặc vân tay.

4.2 Giao diện trang chủ

Sau khi nhập tên màn hình sẽ được chuyển vào màn hình trang chủ , lúc này trang chủ chưa có dữ liệu.



Hình 4.3 Trang chủ khi chưa có dữ liệu

Sau khi nhập dữ liệu màn hình sẽ hiển thị những thông tin đã nhập.

Màn hình phía trên trang chủ hiển thị tên người dùng , 3 tháng gần nhất , thông tin số dư đã được nhập.



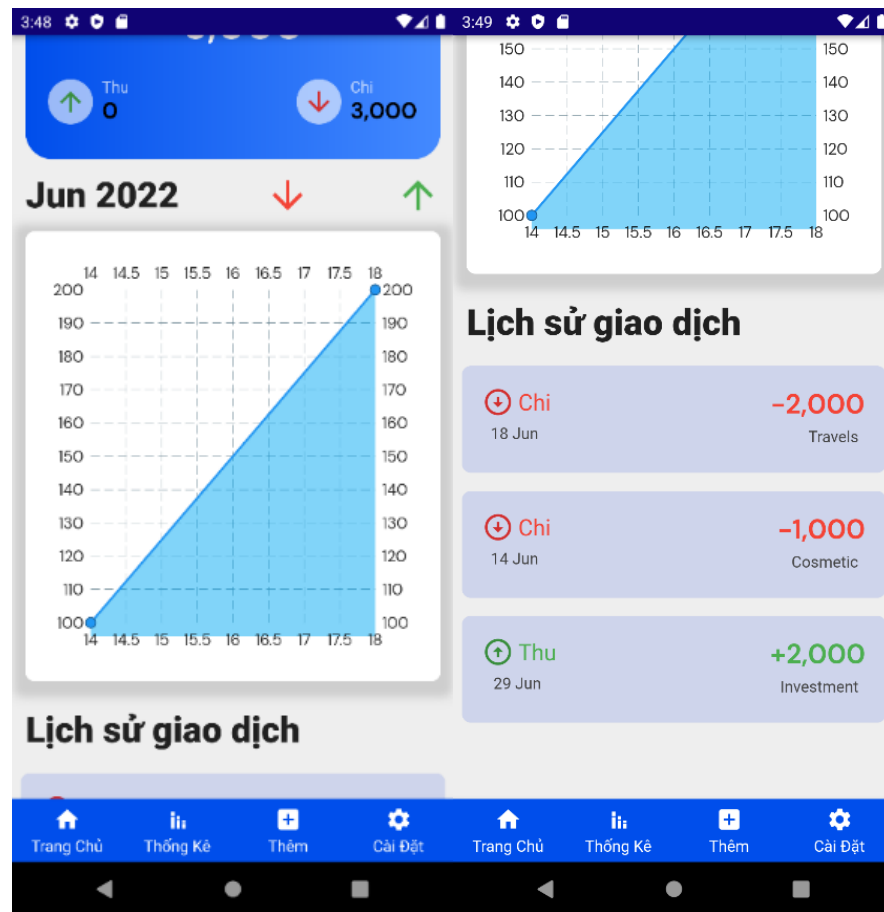
Hình 4.4 Trang chủ

Ở giữa màn hình hiển thị biểu đồ phần dữ liệu đã Chi , với điều kiện dữ liệu Chi lớn hơn 1 mới có thể hiển thị biểu đồ.



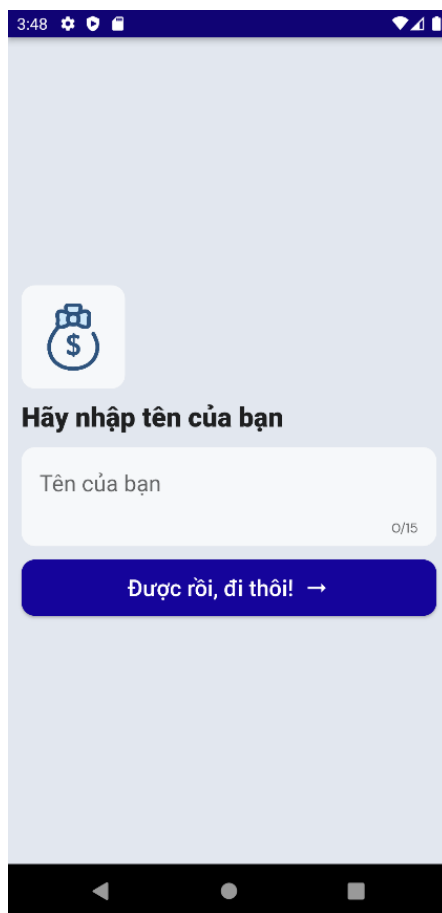
Hình 4.5 Dữ liệu chi nhỏ hơn 1

Ở giữa màn hình hiển thị biểu đồ phần dữ liệu đã Chi , với điều kiện dữ liệu Chi lớn hơn 1 mới có thể hiển thị biểu đồ.



4.3 Giao diện thêm tên người dùng

Sau khi mở khóa thành công ứng dụng sẽ chuyển vào màn hình thêm tên người dùng.

The screenshot shows a mobile application interface for adding a user name. At the top, there is a status bar with the time 3:48 and various icons. Below the status bar, there is a large light blue area. In the center, there is a white rounded rectangle containing a blue icon of a money bag with a dollar sign. Below the icon, the text "Hãy nhập tên của bạn" is displayed in bold. Underneath this text is a white input field with the placeholder text "Tên của bạn" and a character count "0/15" on the right. Below the input field is a blue button with the text "Được rồi, đi thôi! →". At the bottom of the screen, there is a black navigation bar with three icons: a back arrow, a circle, and a square.

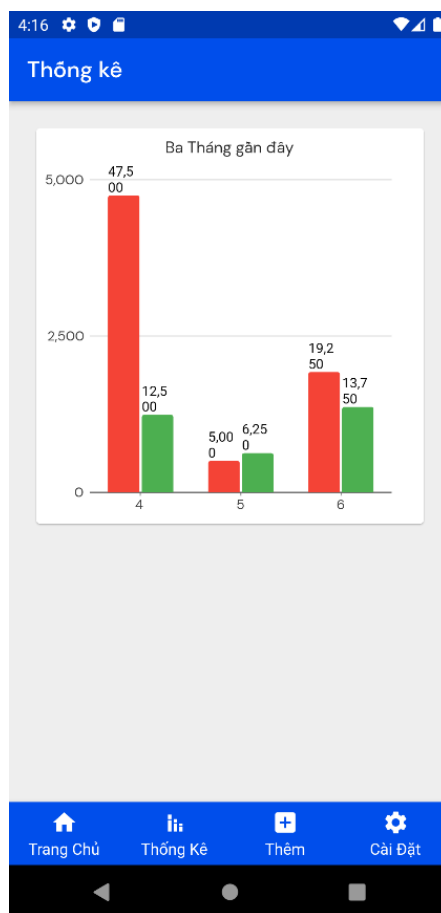
Hình 4.6 Nhập tên người dùng.

4.4 Giao diện thống kê 3 tháng gần nhất

Nhấn vào mục thống kê trên thanh NavigatorBottom sẽ được chuyển vào màn hình thống kê 3 tháng gần nhất.



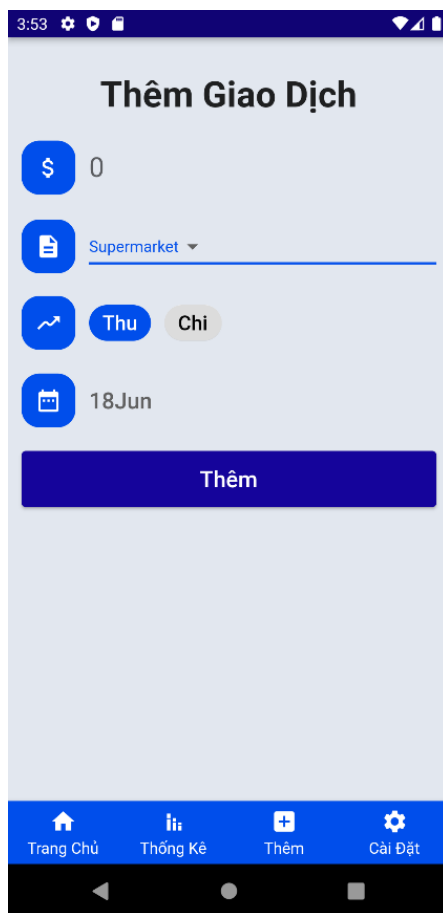
Hình 4.7 Thống kê khi chưa có dữ liệu



Hình 4.8 Thống kê 3 tháng gần nhất.

4.5 Giao diện thêm dữ liệu thu chi

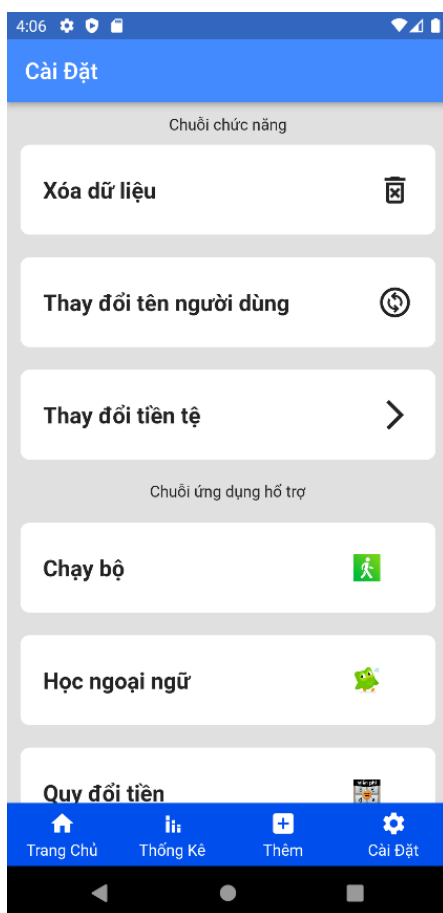
Nhấn vào mục thêm trên thanh NavigatorBottom sẽ được chuyển vào màn hình thêm dữ liệu thu chi.



Hình 4.9 Thêm dữ liệu thu chi.

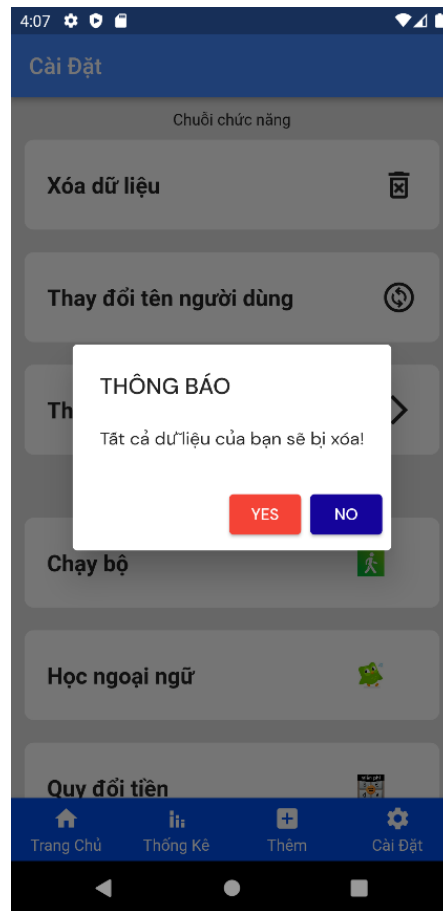
4.6 Giao diện cài đặt

Nhấn vào mục cài đặt trên thanh NavigatorBottom sẽ được chuyển vào màn hình cài đặt



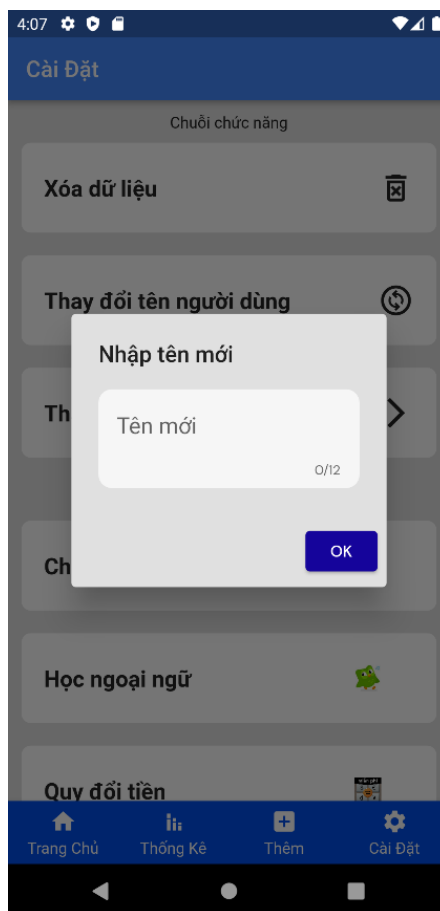
Hình 4.10 Hiển thị cài đặt

Nhấn vào mục Xóa dữ liệu trong màn hình cài đặt sẽ hiển thị dialog thông báo.



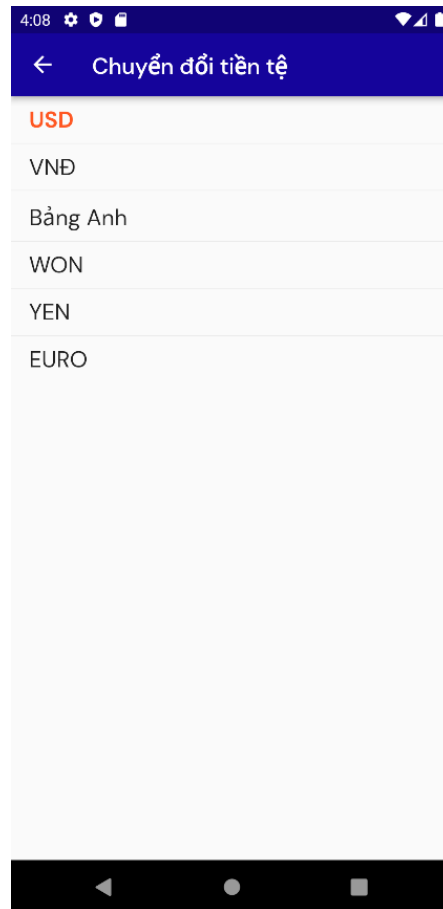
Hình 4.11 Dialog thông báo xóa tất cả dữ liệu

Nhấn vào mục Thay đổi tên người dùng sẽ hiển thị Dialog nhập tên mới của người dùng.



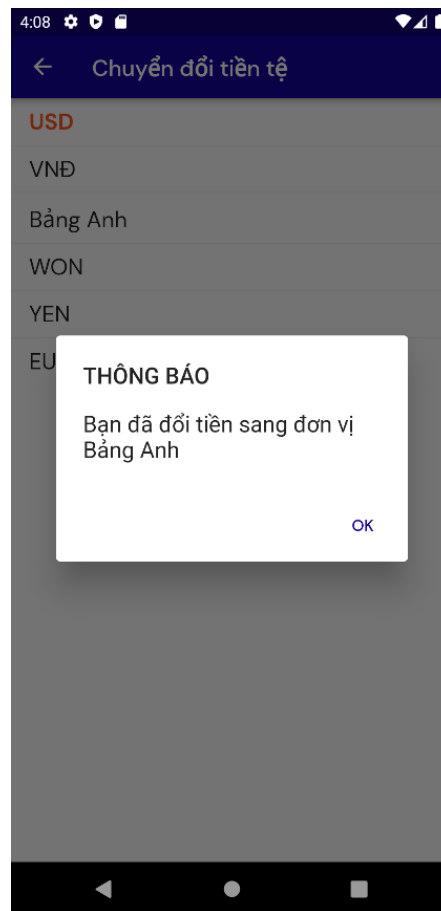
Hình 4.12 Dialog nhập tên mới người dùng

Nhấn vào mục Thay đổi tiền tệ trong màn hình cài đặt sẽ được chuyển qua màn hình Chuyển đổi tiền tệ.



Hình 4.13 Hiển thị đơn vị tiền tệ

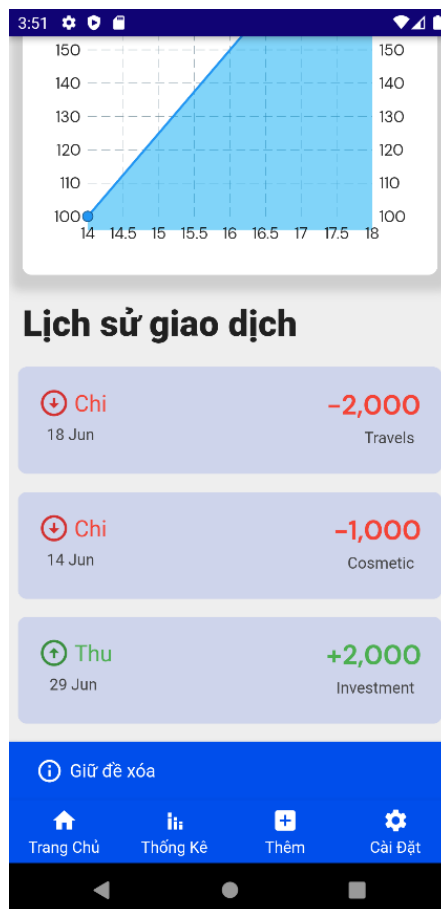
Sau khi nhấn vào 1 trong những đơn vị tiền tệ trong màn hình sẽ hiển thị dialog thông báo chuyển thành công.



Hình 4.14 Dialog thông báo chuyển đơn vị tiền tệ thành công

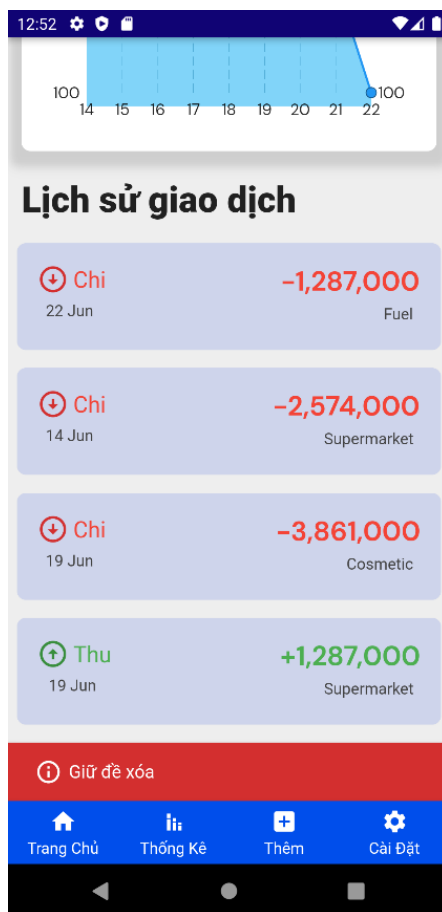
4.7 Giao diện những SnackMessage trong ứng dụng

Snack thông báo giữ 1 card dữ liệu chi hoặc thu sẽ xóa card đó.



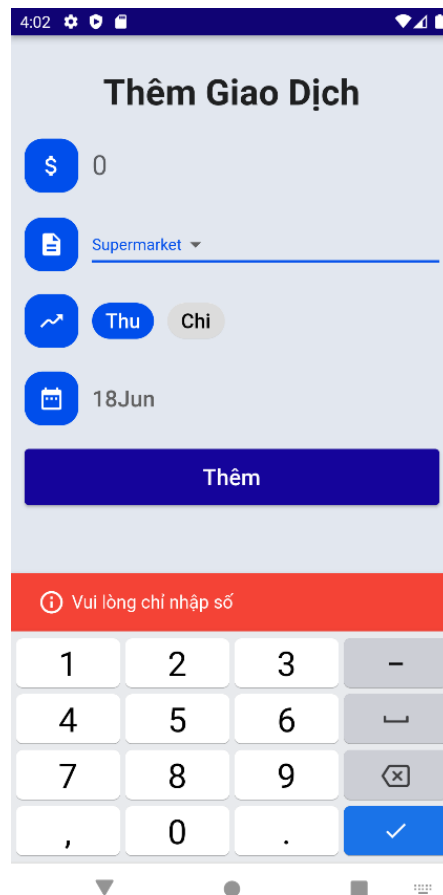
Hình 4.15 Thông báo xóa thẻ đang giữ

Sau khi chạm vào 1 card nào đó sẽ hiển thị Snack thông báo cách xóa card đó.



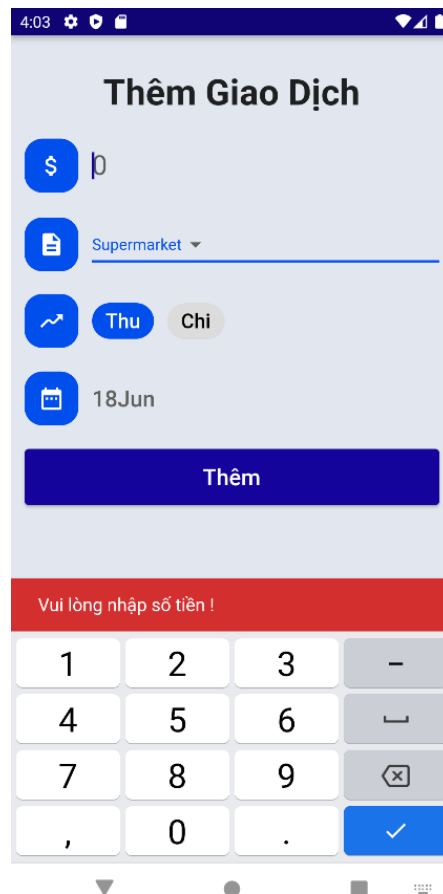
Hình 4.16 Thông báo cách xóa thẻ

Sau khi nhập dữ liệu không phải kiểu số thì sẽ hiển thị Snack thông báo yêu cầu nhập kiểu số.



Hình 4.17 Thông báo người dùng nhập không đúng

Sau khi nhấn nút Thêm mà chưa nhập số tiền chi hoặc thu sẽ hiển thị Snack thông báo yêu cầu nhập số tiền.



Hình 4.18 Thông báo số tiền bị trống

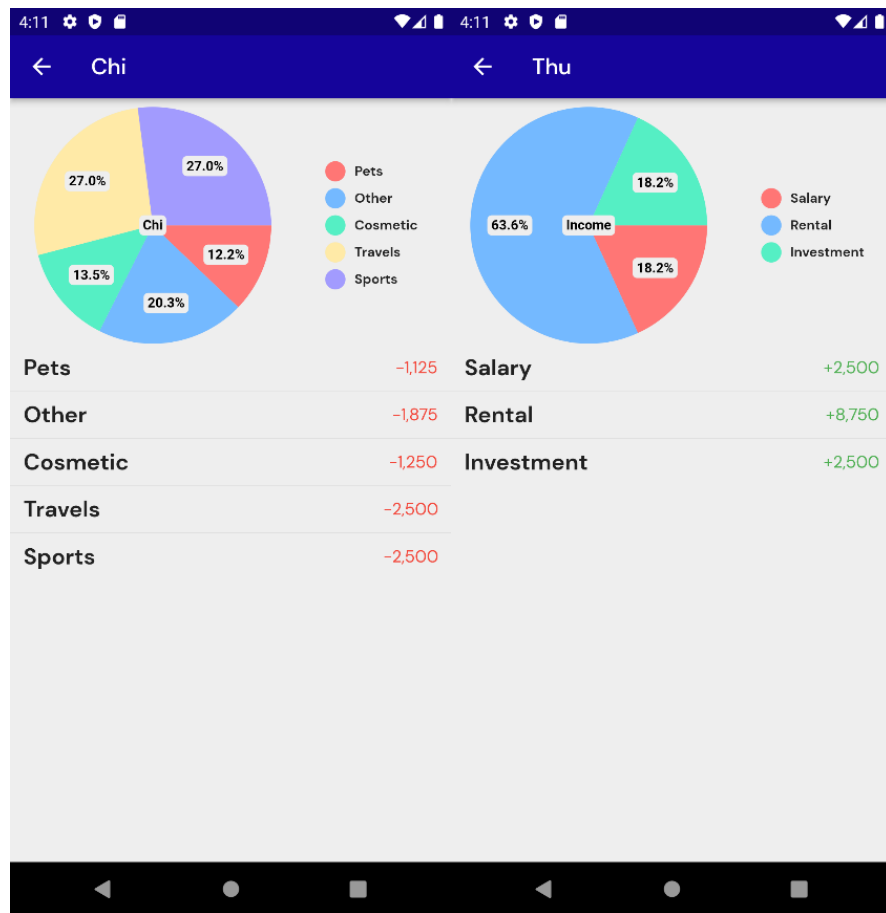
4.8 Giao diện thông kê chi tiết dữ liệu thu hoặc chi

Sau khi nhấn vào biểu tượng chi hoặc thu ở màn hình trang chủ sẽ được chuyển sang màn hình thông kê chi tiết (nếu có dữ liệu) hoặc Snack thông báo chưa có dữ liệu.



Hình 4.19 Thông báo chưa có dữ liệu thu hoặc chi

Màn hình hiển thị chi tiết dữ liệu thu,chi trong một tháng.



KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết quả

1.1 Kết quả đạt được:

Trong quá trình nghiên cứu và hoàn thành đồ án tốt nghiệp với đề tài “Xây dựng ứng dụng đề tài quản lý chi tiêu cá nhân”, em đã đạt được kết quả như sau:

- Ứng dụng bảo mật bằng vân tay, pinCode.
- Thống kê thu chi chi tiết từng tháng.
- Chuyển đổi đơn vị tiền tệ.
- Tính phần trăm những mục thu chi.
- Biểu đồ hiển thị các khoản thu chi từng tháng.
- Biểu đồ hiển thị thu chi trong 3 tháng gần nhất.
- Thống kê thu chi 3 tháng gần nhất.
- Nhập và thay đổi tên người dùng.
- Thêm, Xóa dữ liệu thu chi.
- Đề xuất những ứng dụng tiết kiệm cho việc thu chi.

1.2 Kết quả chưa đạt được:

Bên cạnh những khía cạnh đạt được, do thời gian thực hiện có hạn cùng với trình độ kiến thức còn nhiều hạn chế nên đã còn những thiếu sót một số chức năng như:

- Đặt lịch nhắc nhở.
- Báo cáo thống kê theo năm.
- Hiển thị số chi tiêu trong lịch.

2. Hướng phát triển

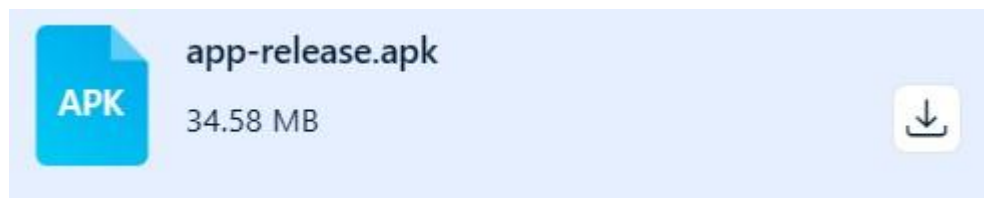
Trong tương lai, nếu có điều kiện đồ án của em sẽ được phát triển theo các hướng sau:

- Liên kết với ngân hàng, ví điện tử để nắm bắt rõ hơn những khoản thanh toán.
- Đặt lịch nhắc nhở.
- Báo cáo thống kê theo năm.
- Hiện thị số chỉ tiêu trong lịch.

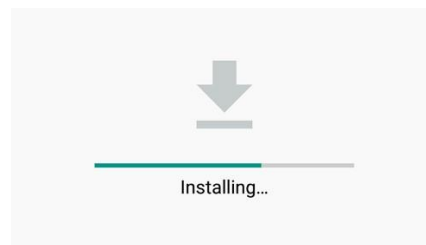
PHỤ LỤC

Hướng dẫn cài đặt

- Download file apk : app-release.apk



- Cài đặt file apk.



TÀI LIỆU THAM KHẢO

- [1] M. Clow, "Tìm hiểu Flutter," in *Learn Google Flutter Fast: 65 Example Apps*, 2019.
- [2] C. Buckett, "Tìm hiểu về ngôn ngữ Dart," in *Dart in action*, 2013.
- [3] T. V. Dev, "Tìm hiểu về Shared-preferences," 2021.
<https://teamvietdev.com/shared-preferences-trong-android/>. [Truy cập 10 6 2022].
- [4] N. T. Minh, "Tìm hiểu về Bloc," 2020. <https://viblo.asia/p/hoc-bloc-pattern-theo-cach-de-hieu-nhat-maGK7JYO5j2>. [Truy cập 15 5 2022].
- [5] S. Biswas, "Tìm hiểu về local_auth", 2020.
https://pub.dev/packages/local_auth. [Truy cập 10 tháng 6 2022].
- [6] N. M. Duong, "Tìm hiểu GetX", 2020. <https://viblo.asia/p/getx-make-flutter-easy-part-1-eW65G1wJZDO>. [Truy cập 12 5 2022].