**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**

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

# 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

### a. 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 đồng 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 lí và có thể tiết kiệm từ thu nhập, chúng ta nên biết cách quản lý tài chính các nhân sao cho thật hiệu quả.

### b. 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

### a. Nội dung:

- Tổng quan về bài toán yêu câu.

- Khảo sát bài toán.

- Phân tích và thiết kế hệ thống theo yêu cầu.

- Xây dựng ứng dụng mobile phục vụ và hổ 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.

### b. 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 hổ trợ liên quan.

## Công nghệ, công cụ và ngôn ngữ lập trình

### a. Công nghệ:

- Framework Flutter.

### b.Công cụ:

- Android studio..

### c. Ngôn ngữ lập trình:

- Dart.

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

### a. 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.

### b. Ứng dụng:

- Hổ trợ mọi người kiểm soát , hiểu rỏ 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: Email: [ttdung@utc2.edu.vn](mailto:ttdung@utc2.edu.vn)

|  |  |
| --- | --- |
| **Ngày tháng 07 năm 2021**  **Trưởng BM Công nghệ Thông tin** | **Đã giao nhiệm vụ TKTN**  **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 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 Phạm Thị Miên 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](#_Toc105936875)

[1. Tên đề tài i](#_Toc105936876)

[2. Mục đích, yêu cầu i](#_Toc105936877)

[a. Mục đích : i](#_Toc105936878)

[b. Yêu cầu : i](#_Toc105936879)

[3. Nội dung và phạm vi đề tài ii](#_Toc105936880)

[a. Nội dung: ii](#_Toc105936881)

[b. Phạm vi đề tài: ii](#_Toc105936882)

[4. Công nghệ, công cụ và ngôn ngữ lập trình iii](#_Toc105936883)

[a. Công nghệ: iii](#_Toc105936884)

[b.Công cụ: iii](#_Toc105936885)

[c. Ngôn ngữ lập trình: iii](#_Toc105936886)

[5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng iii](#_Toc105936887)

[a. Kết quả dự kiến đạt được: iii](#_Toc105936888)

[b. Ứng dụng: iii](#_Toc105936889)

[6. Giáo viên và cán bộ hướng dẫn iv](#_Toc105936890)

[LỜI CẢM ƠN v](#_Toc105936891)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN vi](#_Toc105936892)

[MỤC LỤC vii](#_Toc105936893)

[DANH MỤC CÁC CHỮ VIẾT TẮT x](#_Toc105936894)

[DANH MỤC CÁC BẢNG, BIỂU ĐỒ, HÌNH VẼ xi](#_Toc105936895)

[CHƯƠNG 1: MỞ ĐẦU 1](#_Toc105936896)

[1. Tổng quan đề tài : 1](#_Toc105936897)

[2. Mục tiêu nghiên cứu: 1](#_Toc105936898)

[3. Phạm vi: 1](#_Toc105936899)

[4. Cấu trục báo cáo tốt nghiệp: 1](#_Toc105936900)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_Toc105936901)

[1. Giới thiệu về Flutter: 2](#_Toc105936902)

[1.1 Flutter là gì ? 2](#_Toc105936903)

[1.2 Flutter có thể làm được gì? 2](#_Toc105936904)

[2. Ngôn ngữ lập trình Flutter – Dart: 5](#_Toc105936905)

[2.1 Dart là gì? 5](#_Toc105936906)

[2.2 Tại sao lại chọn Dart? 5](#_Toc105936907)

[2.3 Những ưu điểm của Dart? 6](#_Toc105936908)

[3. SharedPreferences 6](#_Toc105936909)

[3.1 SharedPreferences là gì ? 6](#_Toc105936910)

[3.2 Trường hợp nên sử dụng SharedPreferences? 7](#_Toc105936911)

[4. GetX 7](#_Toc105936912)

[4.1 GetX làm được những gì? 7](#_Toc105936913)

[4.2 GetX – State Manager 7](#_Toc105936914)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc105936915)

[1. Sơ lược về hệ thống: 9](#_Toc105936916)

[2. Sơ đồ phân rã chức năng: 9](#_Toc105936917)

[3. Sơ đồ cơ sở dữ liệu: 9](#_Toc105936918)

[4. Sơ đồ UseCase tổng quát: 9](#_Toc105936919)

[CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG 10](#_Toc105936920)

[1. Giao diện bảo mật: 10](#_Toc105936921)

[2. Giao diện trang chủ 10](#_Toc105936922)

[KẾT LUẬN KIẾN NGHỊ 11](#_Toc105936923)

[1. Kết quả 11](#_Toc105936924)

[a. Kết quả đạt được: 11](#_Toc105936925)

[b.Kết quả chưa đạt được: 11](#_Toc105936926)

[2. Hướng phát triển 11](#_Toc105936927)

[PHỤ LỤC 12](#_Toc105936928)

[Hướng dẫn cài đặt 12](#_Toc105936929)

[TÀI LIỆU THAM KHẢO 13](#_Toc105936930)

# 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

# GIAO DIỆN ỨNG DỤNG

[Hình 1.1 Màn hình khóa ứng dụng. 10](#_Toc106536566)

[Hình 1.2 Màn hình nhập pinCode hoặc vân tay. 11](#_Toc106536567)

[Hình 2.1 Màn hình khi chưa có dữ liệu 12](#_Toc106536568)

[Hình 2.2 Màn hình phía trên trang chủ 13](#_Toc106536569)

[Hình 2.3 Màn hình giữa trang chủ khi dữ liệu Chi nhỏ hơn 1. 14](#_Toc106536570)

[Hình 2.4 Màn hình giữa trang chủ khi dữ liệu Chi lớn hơn 1 . 14](#_Toc106536571)

[Hình 2.5 Màn hình phía dưới cùng của trang chủ hiển thị chi tiết dữ liệu Thu Chi đã được nhập. 15](#_Toc106536572)

[Hình 3.1 Màn hình nhập tên người dùng. 16](#_Toc106536573)

[Hình 4.1 Màn hình Thống kê khi chưa có dữ liệu 17](#_Toc106536574)

[Hình 4.2 Màn hình hiển thị thống kê 3 tháng gần nhất. 17](#_Toc106536575)

[Hình 5.1 Màn hình hiển thị thêm dữ liệu thu chi. 18](#_Toc106536576)

[Hình 6.1 Màn hình hiển thị cài đặt 19](#_Toc106536577)

[Hình 6.2 Dialog thông báo xóa tất cả dữ liệu 19](#_Toc106536578)

[Hình 6.3 Dialog nhập tên mới của người dùng 20](#_Toc106536579)

[Hình 6.4 Màn hình hiển thị đơn vị tiền tệ 21](#_Toc106536580)

[Hình 6.5 Dialog thông báo chuyển đơn vị tiền tệ thành công 22](#_Toc106536581)

[Hình 7.1 Snack xóa card đang giữ 23](#_Toc106536582)

[Hình 7.2 Snack thông báo cách xóa card 23](#_Toc106536583)

[Hình 7.3 Snack thông báo người dùng nhập không đúng kiểu số 24](#_Toc106536584)

[Hình 7.4 Snack thông báo số tiền chi hoặc thu đang bị trống và yêu cầu nhập số tiền. 25](#_Toc106536585)

[Hình 8.1 Snack thông báo chưa có dữ liệu thu hoặc chi 26](#_Toc106536586)

[Hình 8.2 Màn hình hiển thị thống kê chi tiết của dữ liệu chi 26](#_Toc106536587)

[Hình 8.3 Màn hình thống kê chi tiết của dữ liệu thu 27](#_Toc106536588)

# MỞ ĐẦU

## Tổng quan đề tài :

## Mục tiêu nghiên cứu:

Trong tương lai 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.

## 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 hổ trợ liên quan.

## Cấu trục báo cáo tốt nghiệp:

**CHƯƠNG 1: Mở đầu.**

**CHƯƠNG 2: Cơ sở lý thuyết.**

**CHƯƠNG 3: Phân tích thiết kế hệ thống.**

**CHƯƠNG 4: Giao Diện ứng dụng.**

**Kết quả và kiến nghị.**

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về Flutter:

### 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.

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.

**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.

### 1.2 Flutter có thể làm được gì?

**HÌNH CẤU TRÚC FLUTTER**

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 hãng 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 hoạ nào cho cả 2 nền tảng iOS và Android. Bạn có thể tưởng tượng Flutter như 1 engine game, có thể điểu 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 :

* 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.

## Ngôn ngữ lập trình Flutter – Dart:

### 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.

**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, nhắm 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 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.

### 2.3 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.

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.

## SharedPreferences

### 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 khá 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.

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.

### 3.2 Trường hợp nên sử dụng SharedPreferences?

SharedPreferences có thể thực hiện lưu trữ và truy xuát 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 hổ 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 dunngj lưu trữ các thông tin dạng cấu trúc hình ứng dụng, cài đặt… thay vì SQLite bởi SharedPreferences đơn giản và nhanh hơn nhiều.

## State Manager

### 4.1 GetX

#### a. GetX làm được những gì?

* State Manager : Quản lý state trong Flutter
* 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

#### b. GetX – State Manager

##### 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 nguyên lý hoạt động của Get. Những bạn nào đã quen làm việc với Rx chắc sẽ hiểu được nguyên lý hoạt động của nó một cách dễ dàng.

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

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Sơ lược về hệ thống:

…

## Sơ đồ phân rã chức năng:

….

## Sơ đồ cơ sở dữ liệu:

…

## Sơ đồ UseCase tổng quát:

…

# GIAO DIỆN ỨNG DỤNG

## 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 1.1 Màn hình khóa ứng dụng.

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



Hình 1.2 Màn hình nhập pinCode hoặc vân tay.

## 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 2.1 Màn hình 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.

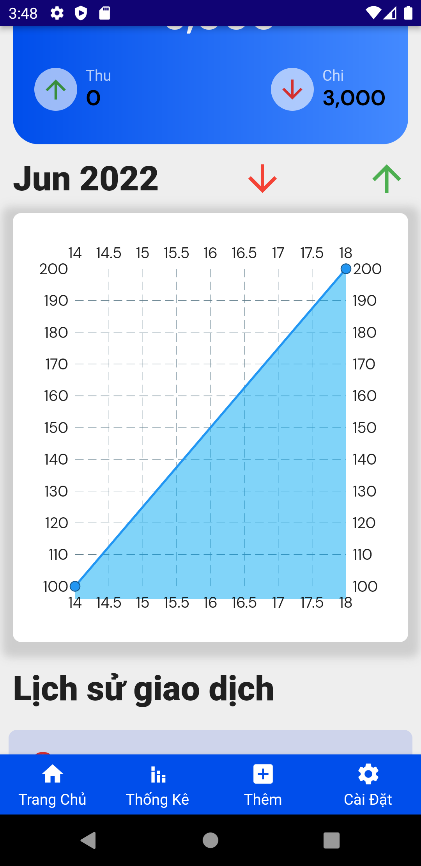


Hình 2.2 Màn hình phía trên trang chủ

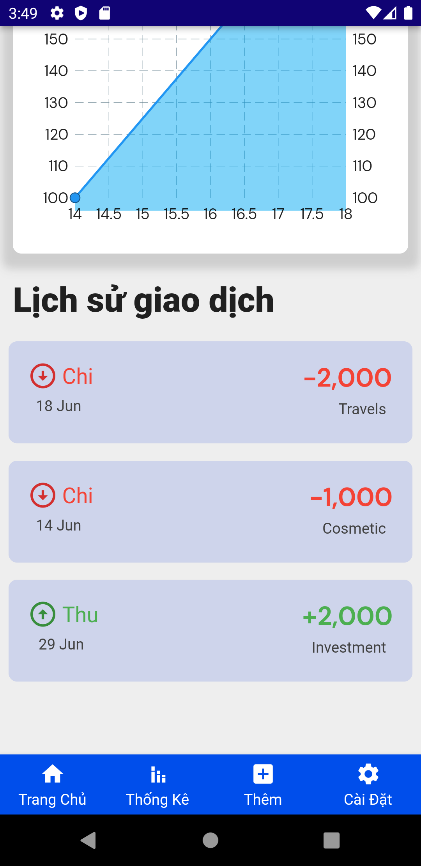
Ở giữa màn hình hiển thị biểu đồ phần dữ liệu đã Chi , với điều kiện dữ liệu Chi lớn hơn 1 mới có thể hiển thị biểu đồ



Hình 2.3 Màn hình giữa trang chủ khi dữ liệu Chi nhỏ hơn 1.



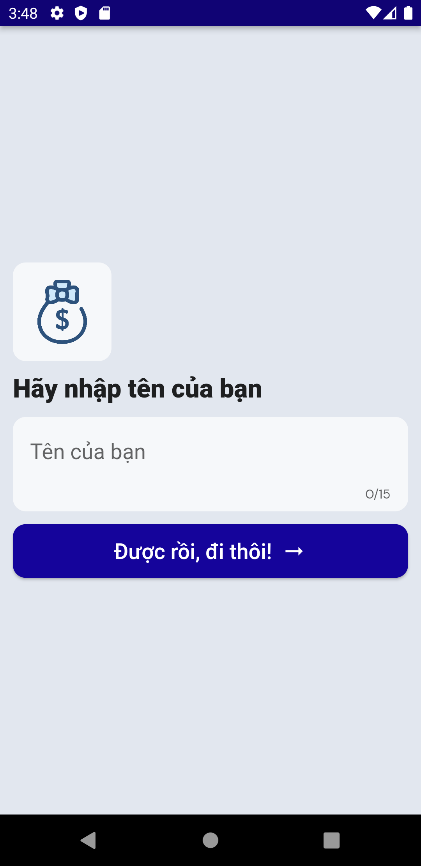
Hình 2.4 Màn hình giữa trang chủ khi dữ liệu Chi lớn hơn 1 .



Hình 2.5 Màn hình phía dưới cùng của trang chủ hiển thị chi tiết dữ liệu Thu Chi đã được nhập.

## Giao diện thêm tên người dùng

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



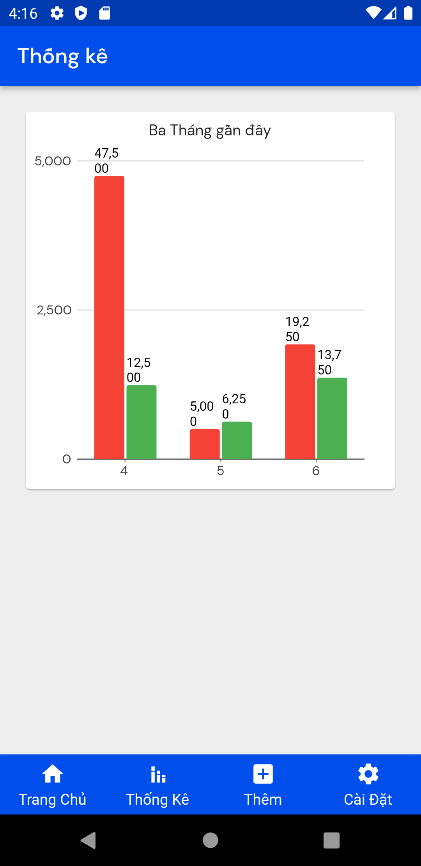
Hình 3.1 Màn hình nhập tên người dùng.

## Giao diện thống kê 3 tháng gần nhất

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



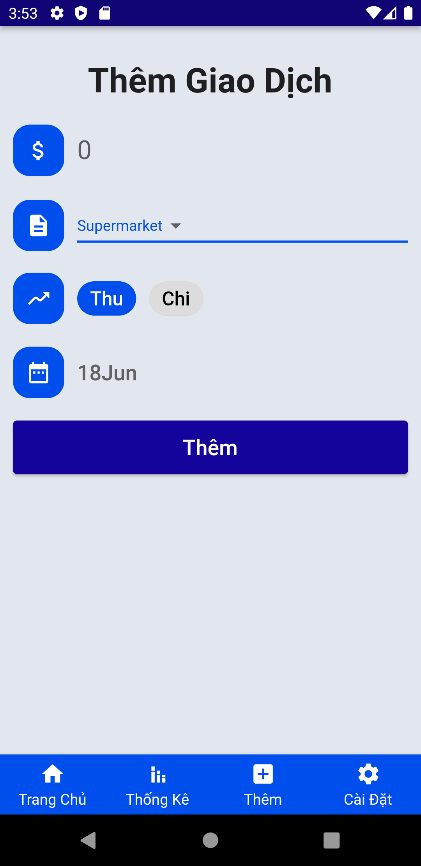
Hình 4.1 Màn hình Thống kê khi chưa có dữ liệu



Hình 4.2 Màn hình hiển thị thống kê 3 tháng gần nhất.

## Giao diện thêm dữ liệu thu chi

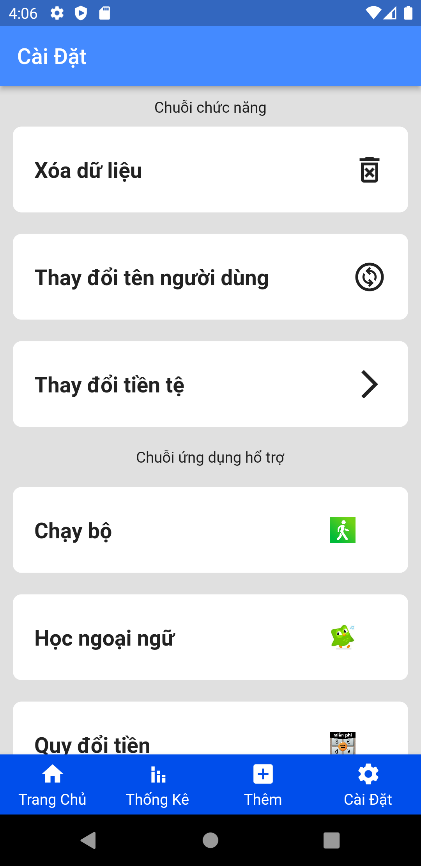
Nhấn vào mục thêm trên thanh NavigatorBottom sẽ được chuyển vào màn hình thêm dữ liệu thu chi.



Hình 5.1 Màn hình hiển thị thêm dữ liệu thu chi.

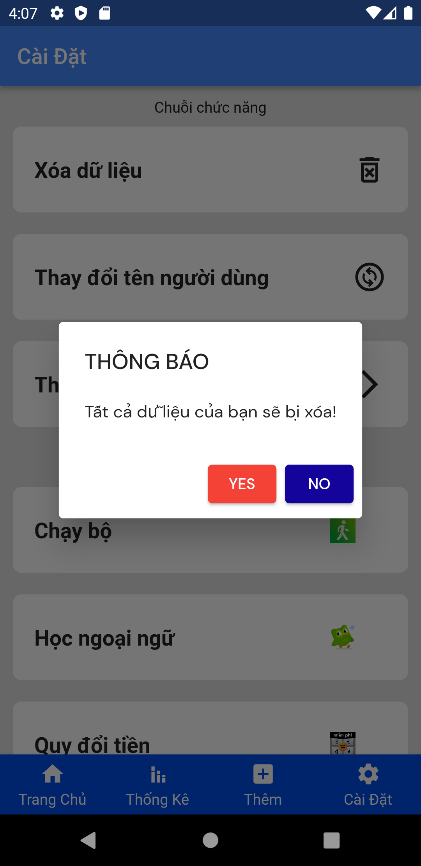
## Giao diện cài đặt

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



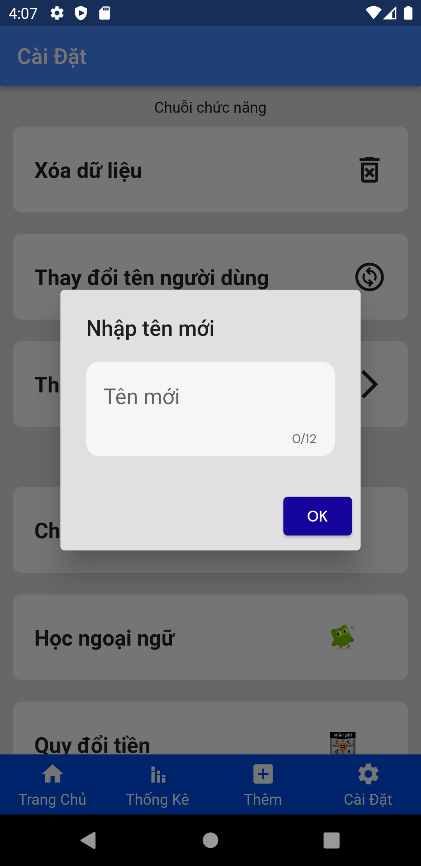
Hình 6.1 Màn hình hiển thị cài đặt

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



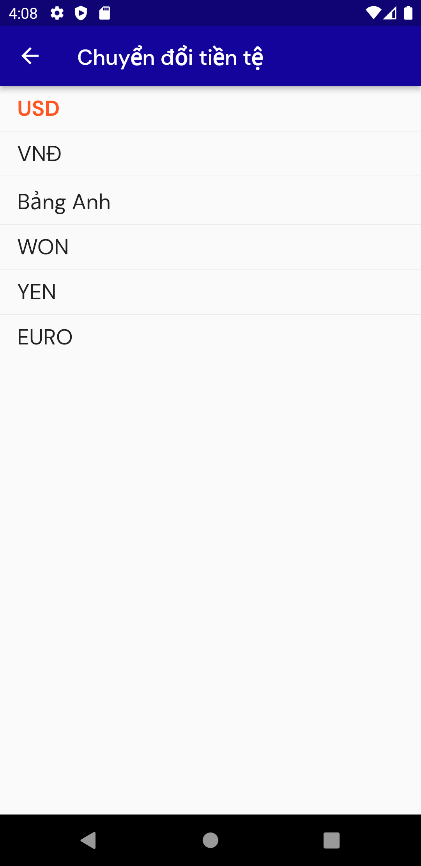
Hình 6.2 Dialog thông báo xóa tất cả dữ liệu

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



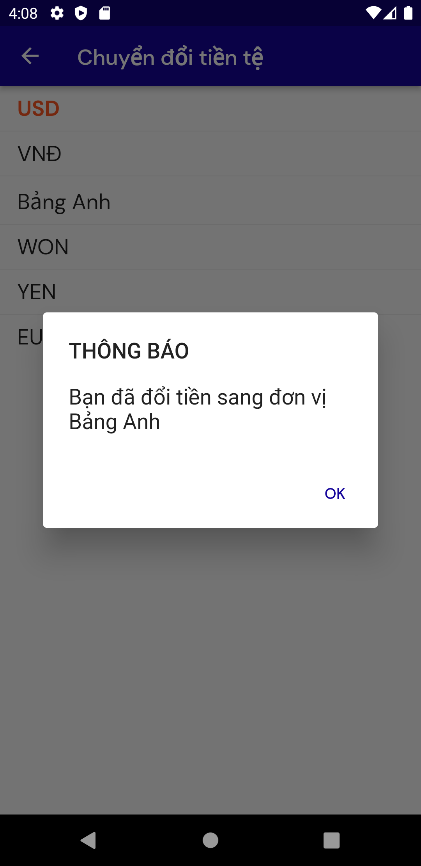
Hình 6.3 Dialog nhập tên mới của người dùng

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



Hình 6.4 Màn hình hiển thị đơn vị tiền tệ

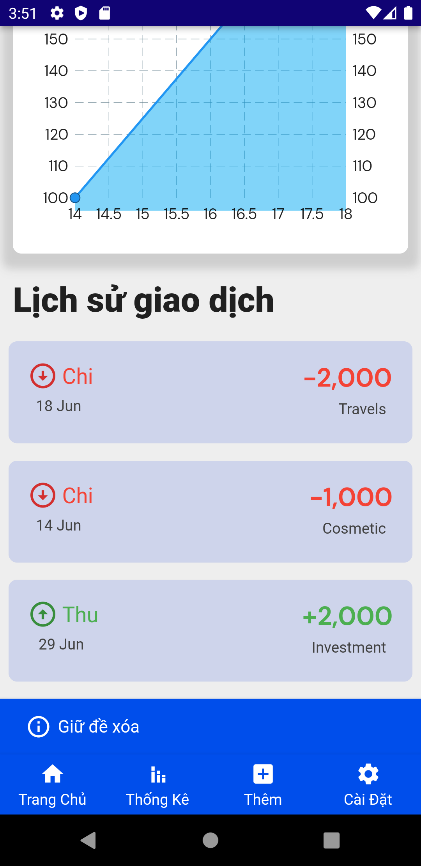
Sau khi nhấn vào 1 trong những đơn vị tiền tệ trong màn hình sẽ hiển thị dialog thông báo chuyển thành công.



Hình 6.5 Dialog thông báo chuyển đơn vị tiền tệ thành công

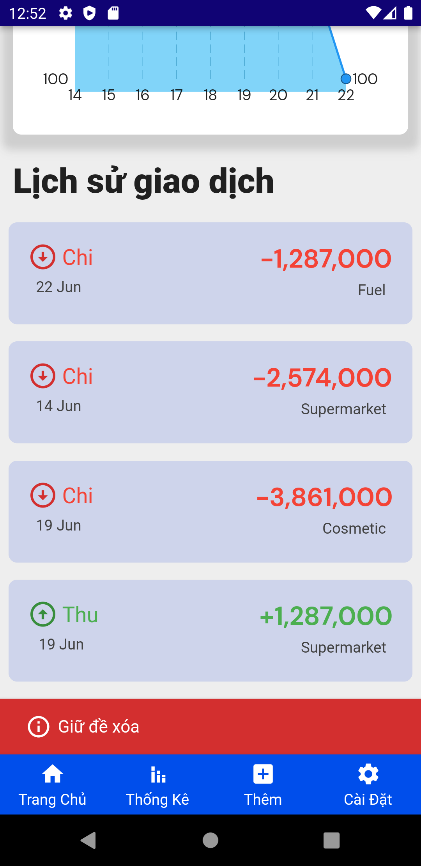
## Giao diện những SnackMessage trong ứng dụng

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



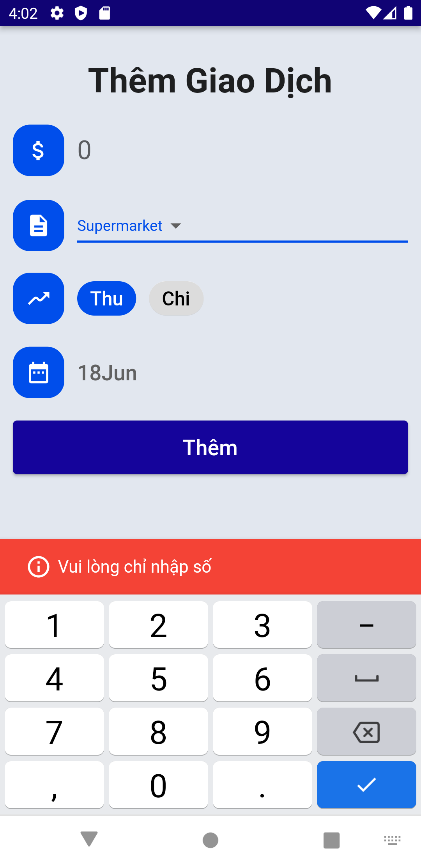
Hình 7.1 Snack xóa card đang giữ

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



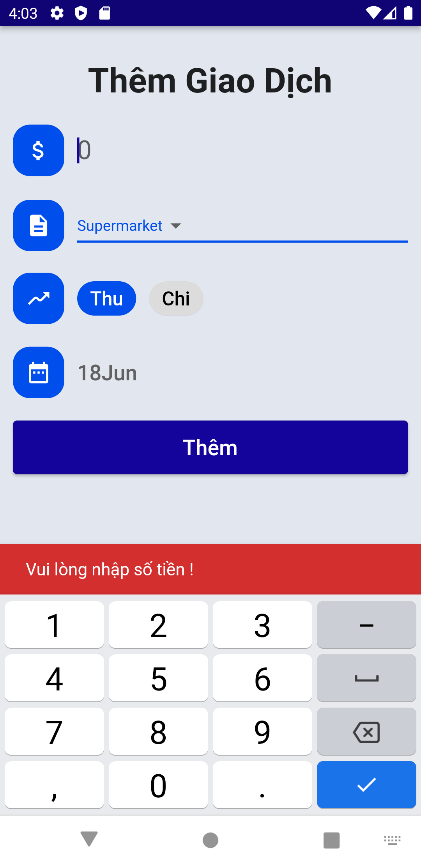
Hình 7.2 Snack thông báo cách xóa card

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



Hình 7.3 Snack thông báo người dùng nhập không đúng kiểu số

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



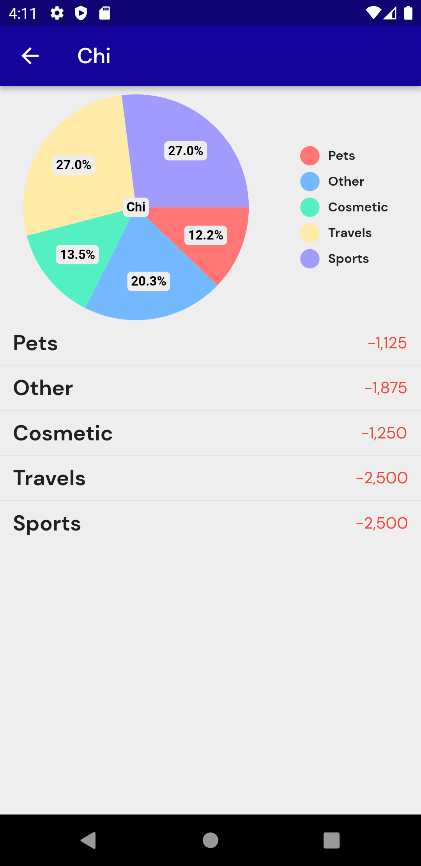
Hình 7.4 Snack thông báo số tiền chi hoặc thu đang bị trống và yêu cầu nhập số tiền.

## Giao diện thống kê chi tiết dữ liệu thu hoặc chi

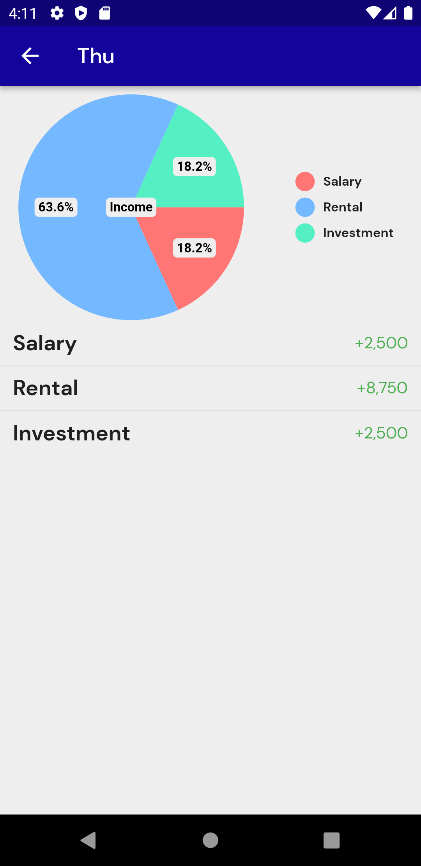
Sau khi nhấn vào biểu tượng chi hoặc thu ở màn hình trang chủ sẽ được chuyển sang màn hình thông kê chi tiết(nếu có dữ liệu) hoặc Snack thông báo chưa có dữ liệu.



Hình 8.1 Snack thông báo chưa có dữ liệu thu hoặc chi



Hình 8.2 Màn hình hiển thị thống kê chi tiết của dữ liệu chi



Hình 8.3 Màn hình thống kê chi tiết của dữ liệu thu

# KẾT LUẬN KIẾN NGHỊ

## Kết quả

### a. Kết quả đạt được:

### b.Kết quả chưa đạt được:

## Hướng phát triển

# PHỤ LỤC

### Hướng dẫn cài đặt

# TÀI LIỆU THAM KHẢO