

**ĐẠI** **HỌC** **PHENIKAA**

**TRƯỜNG CÔNG NGỆ THÔNG TIN**

**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG**

**(N05)**

**“Thiết kế ứng dụng quản lý chi tiêu”**

**Nhóm 13**

|  |  |
| --- | --- |
| Đỗ Hoài Nam | [22014002@st.phenikaa-uni.edu.vn](mailto:22010392@st.phenikaa-uni.edu.vn) |
| Nguyễn Văn Minh | [22010487@st.phenikaa-uni.edu.vn](mailto:22010508@st.phenikaa-uni.edu.vn) |
| Bùi Minh Đức | 22222222@st.phenikaa-uni.edu.vn |

**Giảng viên hướng dẫn:** ThS. Nguyễn Văn Cường

**LỜI CẢM ƠN**

Chúng tôi xin cảm ơn ThS Nguyễn Văn Cương đã tận tình giảng dạy và hướng dẫn, giúp đỡ chúng em hoàn thành đồ án môn học

**LỜI CAM ĐOAN**

Chúng tôi cam đoan Đồ án môn học là sản phẩm trí tuệ của tập thể chúng tôi. Mọi thông tin, dữ liệu, hình ảnh, etc. được sử dụng từ các nguồn khác đều được trích dẫn đầy đủ và có thể tìm thấy các tài liệu liên quan thông qua mục tài liệu tham khảo.

Chúng tôi xin chịu trách nhiệm hoàn toàn về nội dung của Đồ án môn học mà chúng tôi đã nộp.

Hà nội, ngày 03 tháng 11 năm 2025

Ký tên

PHÂN CÔNG NHIỆM VỤ ĐỒ ÁN

|  |  |
| --- | --- |
| **Danh sách các công việc/nhiệm vụ** | **Mô tả tóm tắt công việc** |
| **1. Thiết kế Kiến trúc & Nền tảng** | - Lựa chọn Flutter và mô hình quản lý trạng thái Provider.- Thiết lập cấu trúc thư mục (core, data, features) và định nghĩa các hằng số chung. |
| **2. Xây dựng Lớp Dữ liệu** | - Tạo các lớp Model (Transaction, Budget, AppNotification) với phương thức toJson/fromJson.- Xây dựng StorageService để lưu và đọc dữ liệu dưới dạng JSON vào SharedPreferences. |
| **3. Xây dựng Logic Nghiệp vụ (Providers)** | - Phát triển các Provider (TransactionProvider, BudgetProvider, etc.) để quản lý trạng thái và xử lý logic nghiệp vụ như tính toán số dư, thêm/sửa/xóa dữ liệu. |
| **4. Xây dựng Giao diện Quản lý Giao dịch** | - Tạo các màn hình để hiển thị danh sách, thêm mới và chỉnh sửa giao dịch (AddTransactionScreen).- Kết nối giao diện với TransactionProvider để cập nhật dữ liệu. |
| **5. Triển khai Module Báo cáo & Ngân sách** | - Tích hợp thư viện biểu đồ (fl\_chart) để trực quan hóa dữ liệu thu/chi.- Xây dựng logic kiểm tra chi tiêu so với ngân sách và hiển thị tiến độ. |
| **6. Triển khai Module Tự động hóa** | - Xây dựng chức năng "Giao dịch định kỳ" và "Mẫu giao dịch" với các Provider và màn hình quản lý tương ứng (ManageRecurringScreen, ManageTemplatesScreen). |
| **7. Tích hợp Hệ thống Thông báo** | - Cấu hình NotificationService để gửi thông báo cục bộ.- Kích hoạt thông báo từ các Provider khi có cảnh báo (số dư thấp, vượt ngân sách, giao dịch định kỳ). |
| **8. Xây dựng Module Cài đặt & Hỗ trợ** | - Tạo màn hình SettingsScreen với các chức năng như "Xóa dữ liệu", "Về ứng dụng".- Xây dựng màn hình UserGuideScreen để hướng dẫn người dùng. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Giai đoạn** | **Mục tiêu chính** | **Nguyễn Văn Minh** | **Bùi Minh Đức** | **Đỗ Hoài Nam** |
| **1. Nền tảng** | Xây dựng bộ khung vững chắc cho dự án. | - Thiết lập dự án, cấu trúc thư mục.- Thiết kế Data Models (Transaction, Budget). | - Thiết kế wireframe/mockup.- Xây dựng theme và layout chính (MainScreen). | - Nghiên cứu thư viện (fl\_chart, notifications).- Viết tài liệu luồng dữ liệu. |
| **2. Giao dịch** | Hoàn thiện chức năng quản lý giao dịch cốt lõi. | - Xây dựng StorageService & TransactionProvider. | - Xây dựng màn hình TransactionScreen & AddTransactionScreen.- Tạo widget TransactionItem. | - Kết nối UI với TransactionProvider.- Lập trình chức năng thêm/sửa/xóa giao dịch. |
| **3. Báo cáo & Ngân sách** | Cung cấp công cụ phân tích và kiểm soát tài chính. | - Xây dựng BudgetProvider.- Viết logic kiểm tra ngân sách. | - Xây dựng màn hình ReportScreen & BudgetScreen.- Thiết kế form thêm ngân sách. | - Tích hợp biểu đồ fl\_chart.- Lập trình chức năng quản lý ngân sách. |
| **4. Tự động hóa & Cài đặt** | Hoàn thiện các tính năng nâng cao và màn hình phụ. | - Xây dựng RecurringTransactionProvider & TemplateProvider.- Viết logic tạo giao dịch tự động. | - Xây dựng màn hình SettingsScreen, quản lý mẫu, giao dịch định kỳ.- Xây dựng màn hình hướng dẫn. | - Tích hợp NotificationService.- Lập trình chức năng "Xóa dữ liệu". |
| **5. Hoàn thiện** | Đảm bảo ứng dụng ổn định, sẵn sàng phát hành. | - Tối ưu hóa, refactor code.- Viết unit test. | - Rà soát, hoàn thiện UI/UX.- Thêm animations, loading. | - Kiểm thử toàn diện Sửa lỗi và viết báo cáo. |

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC PHENIKAA  KHOA CÔNG NGHỆ THÔNG TIN | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1 NĂM HỌC 2025 - 2026** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

**Môn học:** Lập trình cho thiết bị di động **Lớp:** NO1

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **Điểm** |
| Đỗ Hoài Nam |  |
| Nguyễn Văn Minh |  |
| Bùi Minh Đức |  |

**Ngày thi:** 02/11/2025 **Phòng thi:**

Giảng viên chấm thi

(Kí và ghi rõ họ tên)

**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VỀ FLUTTER VÀ DART**

**1.1 Giới thiệu về Flutter**

Flutter là một bộ công cụ phát triển giao diện người dùng (UI toolkit) mã nguồn mở do Google phát triển, cho phép lập trình viên xây dựng ứng dụng đa nền tảng (Android, iOS, Web, Desktop) chỉ với một cơ sở mã nguồn duy nhất. Flutter sử dụng cơ chế render riêng (Skia engine) giúp tạo giao diện mượt mà và đồng nhất trên nhiều nền tảng.

Ưu điểm của Flutter:

* Hiệu suất cao: Ứng dụng Flutter được biên dịch thành mã máy gốc (native code).
* Hot Reload: Giúp lập trình viên xem thay đổi ngay lập tức mà không cần khởi động lại ứng dụng.
* Giao diện linh hoạt: Cung cấp nhiều widget có thể tùy chỉnh để xây dựng UI đẹp mắt.
* Hệ sinh thái mạnh mẽ: Cộng đồng lớn, nhiều thư viện hỗ trợ (packages, plugins).

**1.2 So sánh chi tiết giữa Flutter với các nghệ phổ biến**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **Flutter** | **React Native** | **Xamarin** |
| **Ngôn ngữ lập trình** | **Dart** | **Javascript** | **C#** |
| **Hiệu suất** | Biên dịch trực tiếp thành mã máy, hiệu suất cao | Sử dụng các thành phần UI gốc, hiệu suất gần gốc | Truy cập đầy đủ API gốc, hiệu suất cao |
| **Hệ thống Widget** | Hệ thống widget phong phú, tùy biến cao | Component-based, sử dụng các component UI gốc | Xamarin.Forms cho phép chia sẻ mã nguồn UI |
| **Hot Reload** | Có | Có | Có |
| **Ưu điểm** | Tốc độ phát triển nhanh, giao diện đẹp, cộng đồng hỗ trợ mạnh mẽ | Sử dụng JavaScript phổ biến, tích hợp tốt với hệ sinh thái React, hiệu suất cao | Chia sẻ mã nguồn lớn, sử dụng C#, tích hợp tốt với dịch vụ Microsoft |
| **Nhược điểm** | Dart chưa phổ biến rộng rãi, kích thước ứng dụng lớn | Yêu cầu kiến thức lập trình gốc, cộng đồng không đồng đều | Kích thước ứng dụng lớn, cộng đồng nhỏ hơn |

**1.2 Ngôn ngữ lập trình Dart**

Dart là ngôn ngữ lập trình chính mà Flutter sử dụng, cũng được phát triển bởi Google. Dart là ngôn ngữ hướng đối tượng, có cú pháp tương tự như Java, C#, và JavaScript, nên dễ học với các lập trình viên quen thuộc với các ngôn ngữ này.

Đặc điểm nổi bật của Dart:

* Biên dịch linh hoạt: Dart có thể biên dịch Just-in-Time (JIT) giúp hỗ trợ Hot Reload, và Ahead-of-Time (AOT) để tăng tốc độ thực thi.
* Hỗ trợ lập trình bất đồng bộ (async/await) giúp quản lý các tác vụ như truy cập API hoặc cơ sở dữ liệu hiệu quả.
* Quản lý bộ nhớ tự động (Garbage Collection) giúp giảm lỗi tràn bộ nhớ.
* Hướng đối tượng hoàn toàn: Hỗ trợ class, interface, kế thừa, và generic.

**1.3 Mối quan hệ giữa Flutter và Dart**

Flutter sử dụng Dart để viết cả phần giao diện lẫn logic ứng dụng. Dart cho phép biên dịch mã trực tiếp sang mã máy, giúp Flutter đạt hiệu suất cao tương đương với ứng dụng native. Sự kết hợp này giúp các nhà phát triển viết một lần, chạy mọi nơi mà không phải đánh đổi về tốc độ hoặc trải nghiệm người dùng.

**CHƯƠNG 2. THIẾT KẾ ỨNG DỤNG QUẢN LÍ CHI TIÊU**

**CHƯƠNG 2: THIẾT KẾ ỨNG DỤNG QUẢN LÝ CHI TIÊU**

Chương này trình bày chi tiết về quá trình thiết kế và kiến trúc của ứng dụng "Quản lý Chi tiêu". Nội dung bao gồm tổng quan về hệ thống, các chức năng chính, kiến trúc được áp dụng và các công nghệ sử dụng để xây dựng ứng dụng.

**2.1. Tổng quan về hệ thống**

**2.1.1. Giới thiệu**

Ứng dụng "Quản lý Chi tiêu" là một công cụ tài chính cá nhân được phát triển trên nền tảng Flutter, giúp người dùng dễ dàng theo dõi, quản lý các khoản thu nhập và chi tiêu hàng ngày. Với giao diện trực quan và các tính năng thiết thực, ứng dụng hướng tới mục tiêu trở thành một trợ lý tài chính đáng tin cậy, giúp người dùng có cái nhìn rõ ràng về tình hình tài chính, từ đó đưa ra các quyết định chi tiêu thông minh và hiệu quả hơn.

**2.1.2. Mục tiêu của ứng dụng**

* **Ghi chép nhanh chóng:** Cung cấp một giao diện đơn giản để người dùng có thể ghi lại các giao dịch thu/chi một cách nhanh chóng và thuận tiện.
* **Theo dõi tài chính:** Giúp người dùng nắm bắt được tổng thu nhập, tổng chi tiêu và số dư hiện tại.
* **Phân tích trực quan:** Cung cấp các báo cáo, biểu đồ để phân tích thói quen chi tiêu, giúp người dùng hiểu rõ tiền của mình đã đi đâu.
* **Quản lý ngân sách:** Cho phép người dùng thiết lập các hạn mức chi tiêu (ngân sách) cho từng danh mục và cảnh báo khi sắp vượt ngưỡng.
* **Tự động hóa:** Hỗ trợ tạo các giao dịch định kỳ (ví dụ: tiền thuê nhà, lương) để giảm thiểu thao tác thủ công.
* **Tăng cường nhận thức tài chính:** Thông qua các cảnh báo về số dư thấp hoặc chi tiêu vượt ngân sách, ứng dụng giúp nâng cao ý thức và kỷ luật tài chính của người dùng.

**2.1.3. Các chức năng chính**

Dựa trên phân tích mã nguồn, ứng dụng bao gồm các khối chức năng cốt lõi sau:

1. **Quản lý Giao dịch:**
   * Thêm, sửa, xóa các giao dịch thu nhập và chi tiêu.
   * Phân loại giao dịch theo các danh mục (ví dụ: Ăn uống, Di chuyển, Mua sắm).
   * Xem danh sách lịch sử giao dịch, sắp xếp theo ngày tháng.
   * Lọc giao dịch theo thời gian và loại giao dịch.
2. **Báo cáo và Thống kê:**
   * Màn hình tổng quan (Dashboard) hiển thị số dư, tổng thu/chi trong tháng và các giao dịch gần nhất.
   * Xem báo cáo tài chính theo tháng với biểu đồ tròn phân tích tỷ lệ thu/chi.
3. **Quản lý Ngân sách:**
   * Tạo ngân sách chi tiêu cho các danh mục cụ thể theo tháng.
   * Theo dõi và hiển thị tiến độ chi tiêu so với ngân sách đã đặt.
4. **Tự động hóa và Tiện ích:**
   * **Giao dịch định kỳ:** Thiết lập các giao dịch tự động lặp lại hàng tháng.
   * **Mẫu giao dịch:** Tạo các mẫu giao dịch thường dùng để nhập liệu nhanh hơn.
5. **Hệ thống Thông báo:**
   * Gửi cảnh báo khi số dư tài khoản xuống thấp.
   * Gửi cảnh báo khi chi tiêu sắp vượt hoặc đã vượt ngân sách.
   * Thông báo khi một giao dịch định kỳ được tự động tạo.
6. **Cài đặt và Quản lý dữ liệu:**
   * Xem thông tin phiên bản ứng dụng.
   * Cung cấp chức năng xóa toàn bộ dữ liệu để bắt đầu lại.
   * Cung cấp màn hình hướng dẫn sử dụng cho người dùng mới.

**2.1.4. Kiến trúc hệ thống**

Ứng dụng được xây dựng theo kiến trúc phân lớp (Layered Architecture) kết hợp với mô hình quản lý trạng thái Provider, giúp mã nguồn có tổ chức, dễ bảo trì và mở rộng.

* **Nền tảng và Ngôn ngữ:**
  + **Framework:** Flutter - Cho phép xây dựng ứng dụng cho cả Android và iOS từ một cơ sở mã nguồn duy nhất.
  + **Ngôn ngữ:** Dart.
* **Quản lý trạng thái (State Management):**
  + Sử dụng thư viện provider để quản lý trạng thái của ứng dụng. Các Provider (ví dụ: TransactionProvider, BudgetProvider) đóng vai trò là "nguồn chân lý" (source of truth), chứa logic nghiệp vụ và dữ liệu.
  + Giao diện người dùng (UI) sẽ "lắng nghe" các thay đổi từ Provider và tự động cập nhật lại khi có dữ liệu mới, tách biệt hoàn toàn logic khỏi phần hiển thị.
* **Lưu trữ dữ liệu (Data Persistence):**
  + Dữ liệu của ứng dụng (giao dịch, ngân sách, thông báo...) được lưu trữ cục bộ trên thiết bị của người dùng.
  + Sử dụng StorageService, một lớp trừu tượng hóa việc đọc/ghi dữ liệu, có thể được triển khai bằng shared\_preferences hoặc các giải pháp cơ sở dữ liệu cục bộ khác như Hive/sqflite. Việc này giúp dễ dàng thay đổi cơ chế lưu trữ trong tương lai mà không ảnh hưởng nhiều đến logic nghiệp vụ.
* **Cấu trúc thư mục:**
  + **lib/core:** Chứa các thành phần cốt lõi, dùng chung cho toàn bộ ứng dụng như hằng số ([app\_constants.dart](code-assist-path:d:\quanlychitieu\finance_manage-main\lib\core\constants\app_constants.dart)), các dịch vụ (services), và các hàm tiện ích (utils).
  + **lib/data:** Là lớp dữ liệu, bao gồm các models (định nghĩa cấu trúc đối tượng như Transaction, Budget), providers (quản lý trạng thái và logic nghiệp vụ), và services (tương tác với bộ nhớ, API...).
  + **lib/features:** Chứa các màn hình và widget liên quan đến từng tính năng cụ thể (ví dụ: dashboard, settings, reports). Cách tổ chức này giúp dễ dàng tìm kiếm và phát triển các tính năng độc lập.
  + **lib/widgets:** Chứa các widget có thể tái sử dụng trên nhiều màn hình khác nhau (ví dụ: [empty\_state.dart](code-assist-path:d:\quanlychitieu\finance_manage-main\lib\widgets\empty_state.dart)).
  + **lib/localization:** Chứa các file phục vụ cho việc đa ngôn ngữ hóa ứng dụng.

**2.2. Đối tượng của ứng dụng**

Ứng dụng "Quản lý Chi tiêu" được thiết kế để phục vụ một phổ người dùng rộng rãi, từ những người mới bắt đầu tìm hiểu về quản lý tài chính cá nhân đến những người đã có kinh nghiệm và muốn một công cụ đơn giản, hiệu quả để theo dõi dòng tiền của mình. Đối tượng mục tiêu của ứng dụng có thể được phân thành các nhóm chính sau:

**1. Người đi làm và nhân viên văn phòng:**

* **Đặc điểm:** Có nguồn thu nhập ổn định hàng tháng nhưng cũng có nhiều khoản chi tiêu đa dạng (ăn uống, đi lại, giải trí, hóa đơn...). Họ thường bận rộn và cần một công cụ nhanh chóng để ghi chép chi tiêu mà không tốn nhiều thời gian.
* **Nhu cầu:** Theo dõi dòng tiền ra-vào, đảm bảo chi tiêu không vượt quá thu nhập, và có một cái nhìn tổng quan về sức khỏe tài chính cuối mỗi tháng. Tính năng tạo giao dịch định kỳ (lương, tiền thuê nhà) và xem báo cáo tháng là cực kỳ hữu ích cho nhóm này.

**2. Sinh viên:**

* **Đặc điểm:** Thường có ngân sách hạn hẹp, chủ yếu từ trợ cấp gia đình hoặc việc làm thêm. Các khoản chi tiêu thường tập trung vào học tập, ăn uống, và sinh hoạt phí.
* **Nhu cầu:** Quản lý chi tiêu một cách chặt chẽ để không "thâm hụt" ngân sách. Tính năng thiết lập ngân sách cho từng danh mục (ví dụ: đặt ngân sách cho "Ăn uống" là 2.000.000 VNĐ/tháng) và nhận cảnh báo khi sắp vượt mức là công cụ đắc lực giúp sinh viên kiểm soát tài chính hiệu quả.

**3. Các hộ gia đình:**

* **Đặc điểm:** Người quản lý chi tiêu chính trong gia đình (vợ hoặc chồng) cần theo dõi rất nhiều khoản chi chung như tiền điện nước, mua sắm thực phẩm, học phí cho con, v.v.
* **Nhu cầu:** Nắm bắt được tổng chi phí sinh hoạt của gia đình mỗi tháng, xác định những khoản mục nào đang chiếm tỷ trọng lớn để có kế hoạch điều chỉnh. Ứng dụng giúp họ có một bức tranh tài chính rõ ràng của cả gia đình.

**4. Những người có mục tiêu tài chính cụ thể:**

* **Đặc điểm:** Là những người đang tích cực tiết kiệm để đạt được một mục tiêu nào đó trong tương lai (ví dụ: mua xe, đi du lịch, đầu tư).
* **Nhu cầu:** Cần một công cụ để giám sát thói quen chi tiêu, cắt giảm những khoản không cần thiết và tối đa hóa số tiền tiết kiệm được. Việc phân loại chi tiêu và xem báo cáo trực quan giúp họ nhận ra "lỗ hổng" trong chi tiêu và cải thiện nó.

**2.3. Đặc điểm và các chức năng chính**

Ứng dụng "Quản lý Chi tiêu" được xây dựng như một hệ thống tài chính cá nhân đơn giản, thân thiện và dễ sử dụng, cho phép người dùng theo dõi, ghi chép và phân tích các khoản thu chi hàng ngày một cách hiệu quả. Dữ liệu của người dùng được quản lý và lưu trữ an toàn ngay trên thiết bị.

Ứng dụng được thiết kế với các đặc điểm nổi bật:

* **Giao diện hài hòa, thân thiện:** Sử dụng màu sắc rõ ràng (xanh cho thu nhập, đỏ cho chi tiêu), bố cục hợp lý và các biểu tượng trực quan, giúp người dùng dễ dàng thao tác và nắm bắt thông tin.
* **Màn hình tổng quan (Dashboard):** Tương tự trang chủ, màn hình này hiển thị ngay các thông tin quan trọng nhất như số dư hiện tại, tổng thu/chi trong tháng và danh sách các giao dịch gần đây, giúp người dùng có cái nhìn tổng thể nhanh chóng.
* **Ghi chép giao dịch chi tiết:** Người dùng có thể dễ dàng ghi lại thông tin chi tiết về một khoản thu hoặc chi, bao gồm số tiền, danh mục, mô tả và ngày tháng.
* **Quản lý tài chính chủ động:** Người dùng có thể thiết lập các hạn mức chi tiêu (ngân sách) cho từng danh mục và nhận cảnh báo khi sắp chi tiêu vượt mức, giúp kiểm soát tài chính tốt hơn.
* **Không yêu cầu đăng nhập:** Ứng dụng hoạt động ngoại tuyến và lưu trữ dữ liệu trực tiếp trên điện thoại người dùng, đảm bảo tính riêng tư và sự tiện lợi, không cần kết nối mạng hay tạo tài khoản.
* **Hỗ trợ và hướng dẫn:** Cung cấp màn hình hướng dẫn sử dụng chi tiết và các mục hỗ trợ để người dùng mới có thể nhanh chóng làm quen và khai thác tối đa các tính năng.

**Các module chức năng chính của ứng dụng:**

* **Module Quản lý Giao dịch:**
  + Đây là module cốt lõi, cho phép người dùng thực hiện các thao tác: Thêm, sửa, xóa các giao dịch thu nhập và chi tiêu.
  + Thông tin giao dịch được hiển thị đầy đủ và rõ ràng, bao gồm: tên giao dịch, số tiền, danh mục, ngày tháng và ghi chú.
  + Hỗ trợ lọc và xem lại lịch sử giao dịch theo thời gian hoặc theo loại (thu/chi).
* **Module Quản lý Ngân sách:**
  + Người dùng có thể tạo ra các "ngân sách" (hạn mức chi tiêu) cho một danh mục cụ thể trong một tháng (ví dụ: ngân sách cho "Ăn uống" là 2.000.000 VNĐ).
  + Hệ thống sẽ tự động theo dõi các khoản chi tiêu thuộc danh mục đó và hiển thị tiến độ dưới dạng thanh phần trăm, giúp người dùng biết mình đã chi bao nhiêu so với hạn mức.
* **Module Báo cáo và Thống kê:**
  + Cung cấp các báo cáo tài chính theo tháng.
  + Sử dụng biểu đồ tròn trực quan để phân tích tỷ lệ giữa tổng thu nhập và tổng chi tiêu, giúp người dùng hiểu rõ cơ cấu dòng tiền của mình.
  + Thống kê chi tiết tổng thu, tổng chi và số dư cuối kỳ của tháng được chọn.
* **Module Tự động hóa (Giao dịch định kỳ & Mẫu):**
  + **Giao dịch định kỳ:** Cho phép người dùng thiết lập các khoản thu/chi lặp lại hàng tháng (ví dụ: lương, tiền thuê nhà). Đến ngày đã định, ứng dụng sẽ tự động tạo giao dịch tương ứng.
  + **Mẫu giao dịch:** Người dùng có thể tạo các mẫu cho những giao dịch thường xuyên để việc thêm mới được thực hiện nhanh chóng chỉ với vài cú nhấp.
* **Module Cài đặt và Hỗ trợ:**
  + **Quản lý dữ liệu:** Cung cấp các tùy chọn quản lý dữ liệu như "Xóa tất cả dữ liệu" để bắt đầu lại. Các tính năng như sao lưu và khôi phục đang được lên kế hoạch phát triển.
  + **Thông tin và Hỗ trợ:** Bao gồm màn hình "Về ứng dụng", "Hướng dẫn sử dụng" chi tiết, và các mục liên hệ, góp ý để người dùng có thể tương tác với nhà phát triển.
* **Module Thông báo:**
  + Là một module chạy ngầm, có vai trò gửi các cảnh báo quan trọng đến người dùng, chẳng hạn như:
    - Cảnh báo khi số dư tài khoản xuống thấp.
    - Cảnh báo khi chi tiêu sắp vượt hoặc đã vượt ngân sách đã đặt.
    - Thông báo khi một giao dịch định kỳ được tự động tạo.

**2.3 Xác định các chức năng trong hệ thống**

Dựa trên mục tiêu và đối tượng người dùng đã xác định, hệ thống được xây dựng với các nhóm chức năng chính, được phân bổ theo mức độ quan trọng và tần suất sử dụng của người dùng. Các nhóm chức năng này được thể hiện qua biểu đồ tròn (Hình 2.1), giúp trực quan hóa vai trò và "sức nặng" của từng module trong toàn bộ ứng dụng.

Các nhóm chức năng chính bao gồm:

1. **Quản lý Giao dịch (40%):** Đây là chức năng cốt lõi và quan trọng nhất của ứng dụng. Nó bao gồm tất cả các nghiệp vụ liên quan đến việc ghi chép, theo dõi dòng tiền hàng ngày của người dùng như:
   * Thêm, sửa, xóa các giao dịch thu nhập và chi tiêu.
   * Phân loại giao dịch theo danh mục.
   * Xem lịch sử và lọc giao dịch.
2. **Báo cáo & Phân tích (25%):** Nhóm chức năng này giúp người dùng hiểu rõ thói quen chi tiêu và tình hình tài chính của mình thông qua các công cụ trực quan.
   * Hiển thị tổng quan số dư, thu/chi trên màn hình chính (Dashboard).
   * Cung cấp báo cáo theo tháng với biểu đồ phân tích.
3. **Quản lý Ngân sách (15%):** Chức năng giúp người dùng thiết lập kỷ luật tài chính và kiểm soát chi tiêu một cách chủ động.
   * Tạo hạn mức chi tiêu (ngân sách) cho từng danh mục.
   * Theo dõi tiến độ chi tiêu so với ngân sách đã đặt.
4. **Tự động hóa & Tiện ích (10%):** Nhóm chức năng này giúp giảm thiểu thao tác thủ công và tăng tính tiện lợi cho người dùng.
   * Thiết lập các giao dịch định kỳ (lặp lại hàng tháng).
   * Tạo và quản lý các mẫu giao dịch thường dùng.
5. **Cài đặt & Hỗ trợ (10%):** Bao gồm các chức năng quản trị ứng dụng và hỗ trợ người dùng.
   * Quản lý dữ liệu (sao lưu, khôi phục, xóa dữ liệu).
   * Cung cấp hướng dẫn sử dụng, thông tin ứng dụng và các mục hỗ trợ khác.

**Thiết kế biểu đồ tròn biểu thị tỷ lệ các chức năng:**

Biểu đồ tròn (pie chart) để thể hiện tỷ trọng của mỗi nhóm chức năng trong hệ thống.

2.4 Biểu đồ use case các chức năng của hệ thống

Đối với ứng dụng "Quản lý Chi tiêu", chúng ta xác định hai tác nhân chính:

**Người dùng (User):** Là người trực tiếp sử dụng ứng dụng để thực hiện các hoạt động quản lý tài chính cá nhân.

**Hệ thống (System):** Là tác nhân nội tại, tự động thực hiện các tác vụ nền như tạo giao dịch định kỳ, gửi cảnh báo mà không cần sự can thiệp trực tiếp của người dùng tại thời điểm đó.

Dưới đây là các biểu đồ use case chi tiết cho từng nhóm chức năng và biểu đồ tổng quát của toàn bộ hệ thống.

1. Biểu đồ Use Case Tổng quát

ác nhân:

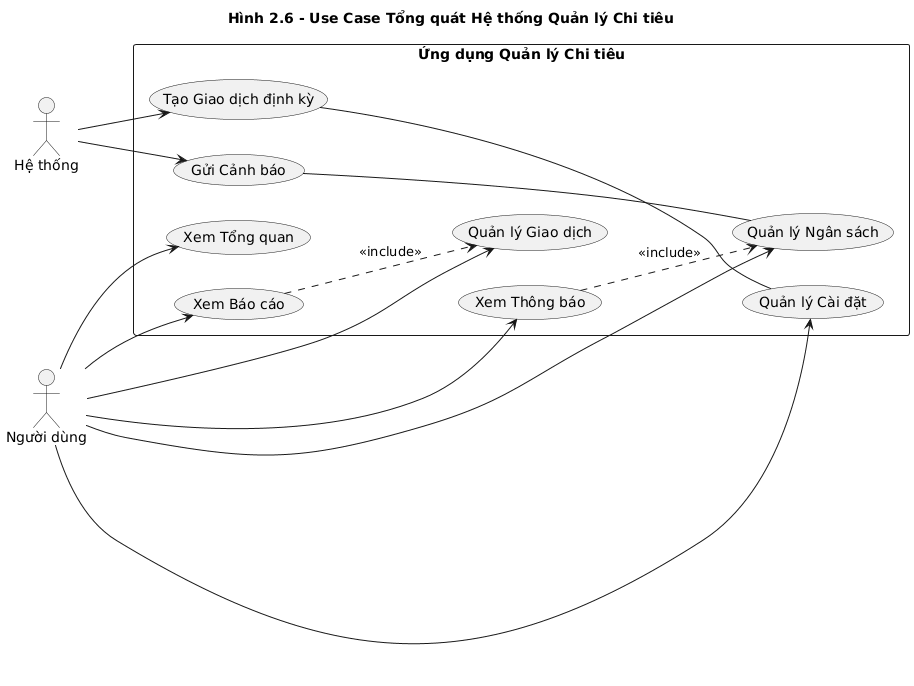
* Người dùng, Hệ thống.

Các Use case chính của Người dùng:

* Xem Tổng quan (dashboard: số dư, chi tiêu gần đây, tiến độ ngân sách).
* Quản lý Giao dịch (thêm/sửa/xóa/xem/ lọc/ thêm từ mẫu).
* Xem Báo cáo (biểu đồ thu–chi, theo danh mục/chu kỳ).
* Quản lý Ngân sách (tạo/xem/chỉnh sửa/xóa).
* Quản lý Cài đặt (mẫu, định kỳ, trợ giúp, xóa dữ liệu).
* Xem Thông báo (lịch sử cảnh báo/hệ thống).

Các Use case chính của Hệ thống:

* Tạo Giao dịch định kỳ, Gửi Cảnh báo.



2. Biểu đồ Use Case: Quản lý Giao dịch (Hình 2.2)

Tác nhân:

* Người dùng – tạo/duyệt/chỉnh sửa/xóa giao dịch, lọc và tạo nhanh từ mẫu.

Phạm vi: Ứng dụng “Quản lý Chi tiêu”.

Tiền điều kiện:

* Người dùng đã mở ứng dụng và (nếu có) đăng nhập.
* Danh mục/đơn vị tiền tệ mặc định tồn tại (hoặc có cấu hình tối thiểu).

Luồng chính (Use case chính: “Quản lý Giao dịch”)

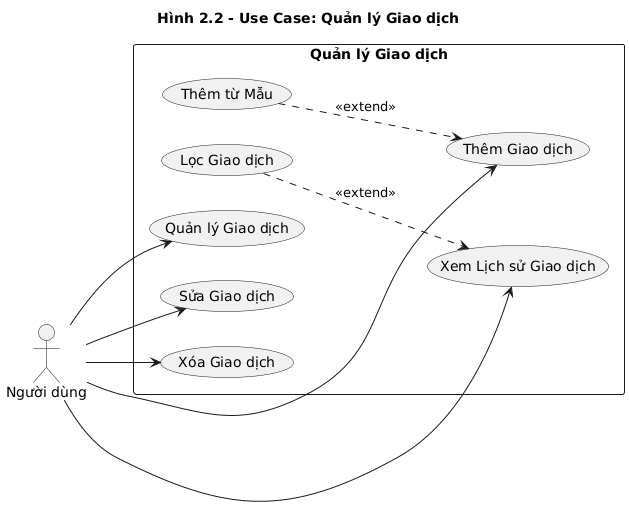
1. Thêm Giao dịch: nhập số tiền, loại (thu/chi), thời điểm, danh mục, ghi chú → lưu.
2. Sửa Giao dịch: chọn giao dịch → cập nhật trường → lưu.
3. Xóa Giao dịch: chọn giao dịch → xác nhận xóa.
4. Xem Lịch sử Giao dịch: duyệt danh sách theo thời gian.
5. Lọc Giao dịch *(extend của Xem Lịch sử)*: lọc theo khoảng thời gian, loại, danh mục, số tiền.
6. Thêm từ Mẫu *(extend của Thêm Giao dịch)*: chọn mẫu có sẵn → tự động điền trường mặc định → chỉnh sửa nhỏ (nếu cần) → lưu.

Ngoại lệ/biến thể:

* Dữ liệu thiếu/không hợp lệ → hiển thị lỗi, yêu cầu nhập lại.
* Hủy thao tác trước khi lưu → không thay đổi dữ liệu.

Hậu điều kiện:

* Danh sách giao dịch phản ánh thay đổi (thêm/sửa/xóa).
* Nhật ký ghi nhận thời điểm thao tác (nếu hệ thống có logging).



3. Biểu đồ Use Case: Quản lý Ngân sách (Hình 2.3)

Tác nhân:

* Người dùng – thiết lập, theo dõi và chỉnh sửa/xóa ngân sách.

Tiền điều kiện:

* Danh mục chi tiêu đã có/được chọn.
* Đơn vị tiền tệ và chu kỳ (tháng) xác định sẵn.

Luồng chính (Use case chính: “Quản lý Ngân sách”)

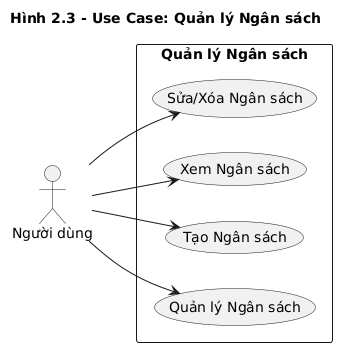
1. Tạo Ngân sách: chọn danh mục → nhập hạn mức/tháng → lưu.
2. Xem Ngân sách: xem % sử dụng, số tiền còn lại, cảnh báo gần chạm ngưỡng.
3. Sửa/Xóa Ngân sách: điều chỉnh hạn mức hoặc xóa cấu hình hiện tại.

Ngoại lệ/biến thể:

* Trùng danh mục trong cùng kỳ → yêu cầu hợp nhất/ghi đè.
* Nhập hạn mức âm/0 → báo lỗi.

Hậu điều kiện:

* Ngân sách ảnh hưởng tới cảnh báo/đồ thị báo cáo.
* Dữ liệu đồng bộ với màn hình tổng quan.



4. Biểu đồ Use Case: Quản lý Cài đặt & Dữ liệu (Hình 2.4)

Tác nhân:

* Người dùng

Luồng chính (Use case chính: “Quản lý Cài đặt”)

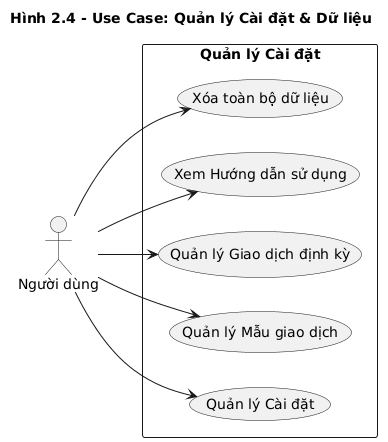
1. Quản lý Mẫu giao dịch: thêm/sửa/xóa các mẫu (template) dùng nhanh.
2. Quản lý Giao dịch định kỳ: thêm/sửa/xóa định kỳ (chu kỳ, ngày chạy, số tiền, danh mục).
3. Xem Hướng dẫn sử dụng: truy cập trang trợ giúp.
4. Xóa toàn bộ dữ liệu: xác nhận 2 bước → sao lưu cục bộ (tùy chọn) → xóa.

Ngoại lệ/biến thể:

* Hủy thao tác xóa dữ liệu → không thay đổi.
* Định kỳ mâu thuẫn (trùng thời điểm/trùng mẫu) → yêu cầu hợp nhất/chỉnh sửa.

Hậu điều kiện:

* Cấu hình mới có hiệu lực ngay cho các màn hình liên quan.
* Định kỳ đồng bộ sang “Tác vụ tự động”.



5. Biểu đồ Use Case: Tác vụ tự động của Hệ thống (Hình 2.5)

Tác nhân:

* Hệ thống – chạy nền theo lịch/điều kiện.

Tiền điều kiện:

* Có cấu hình giao dịch định kỳ hoặc ngưỡng ngân sách/cảnh báo.
* Dịch vụ thông báo (push/local) hoạt động.

Luồng chính (Use case chính)

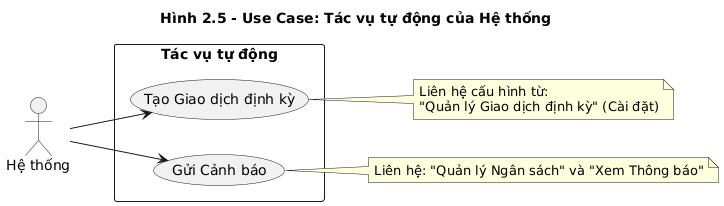
1. Tạo Giao dịch định kỳ: đến ngày/giờ → tạo giao dịch mới dựa trên cấu hình của người dùng → gắn cờ “tự động”.
2. Gửi Cảnh báo: kiểm tra điều kiện (vượt ngân sách, số dư thấp, gần ngày hóa đơn, v.v.) → phát thông báo.

Quan hệ:

* “Tạo Giao dịch định kỳ” liên hệ với “Quản lý Giao dịch định kỳ” ở phần Cài đặt (nguồn cấu hình).
* “Gửi Cảnh báo” liên hệ với “Quản lý Ngân sách” (nguồn ngưỡng so sánh) và “Xem Thông báo”.

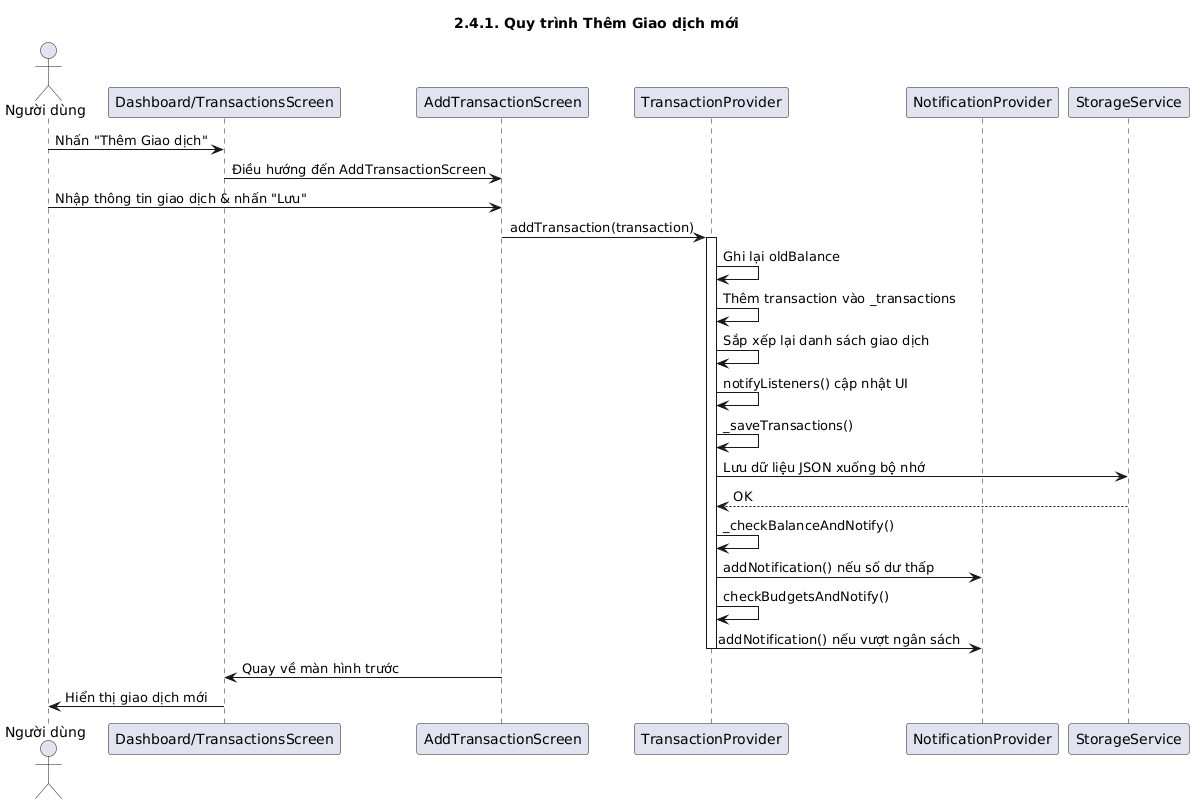
Hậu điều kiện:

* Dòng tiền và thông báo được cập nhật đồng bộ.

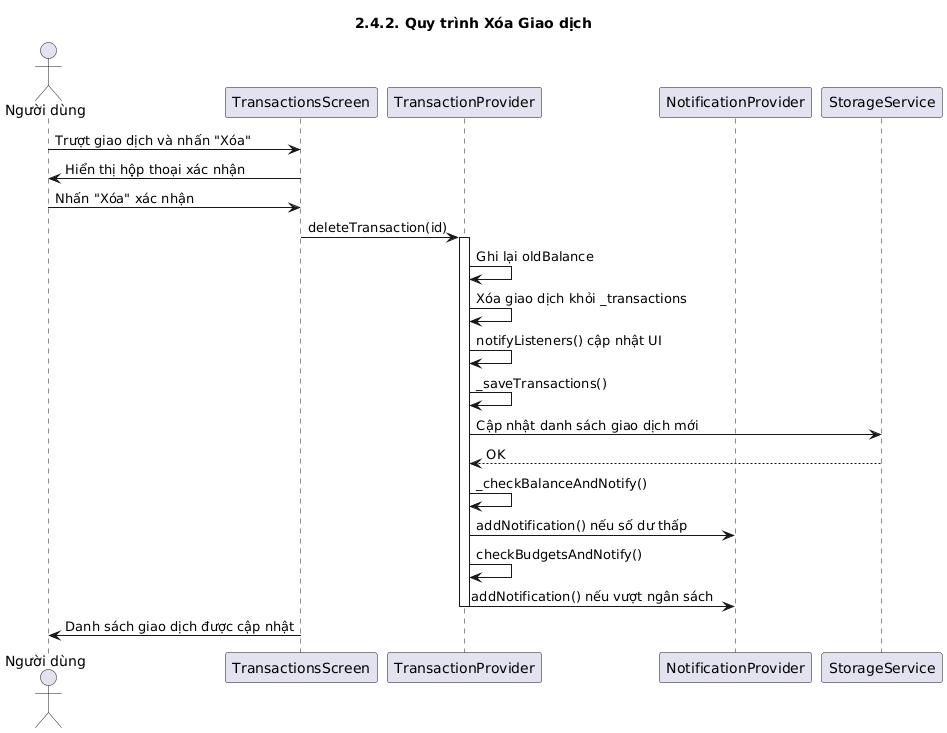


2.5 Biểu đồ tuần tự ( Sequence Diagram)

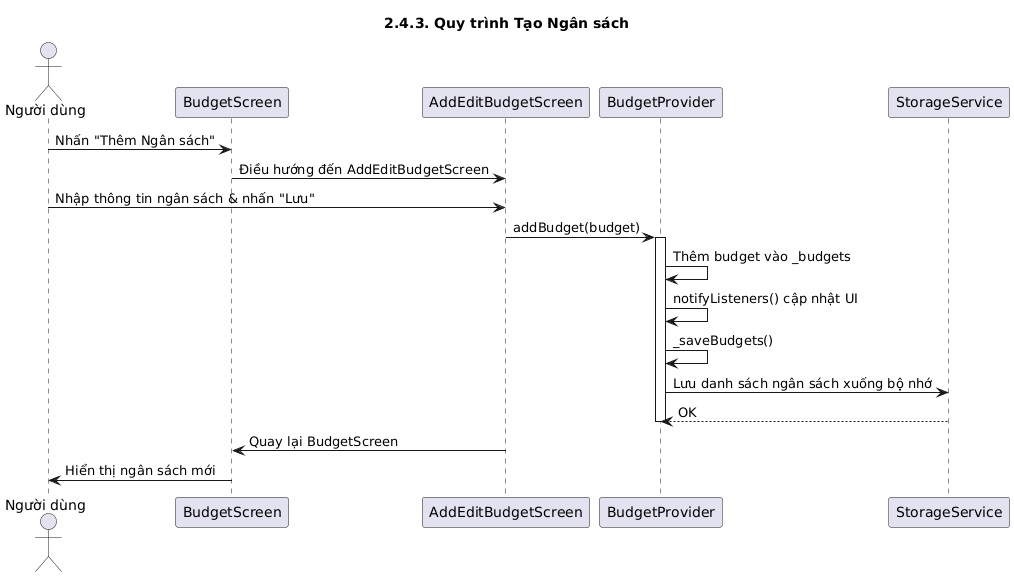
1. Quy trình thêm giao dịch mới



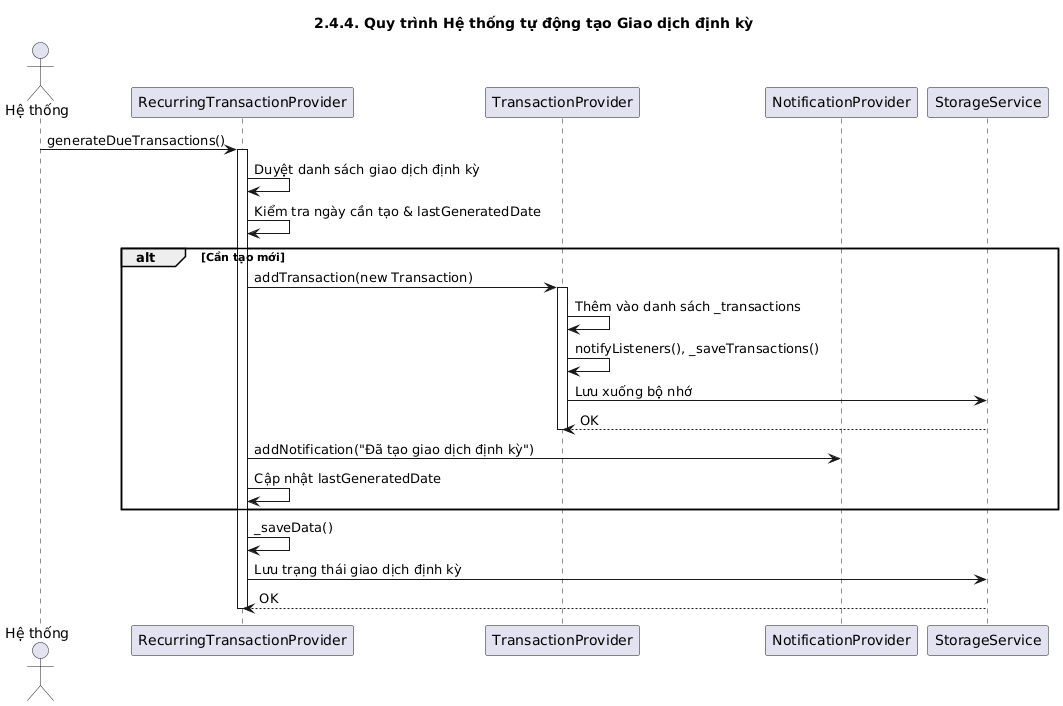
2. Quy trình Xóa giao dịch



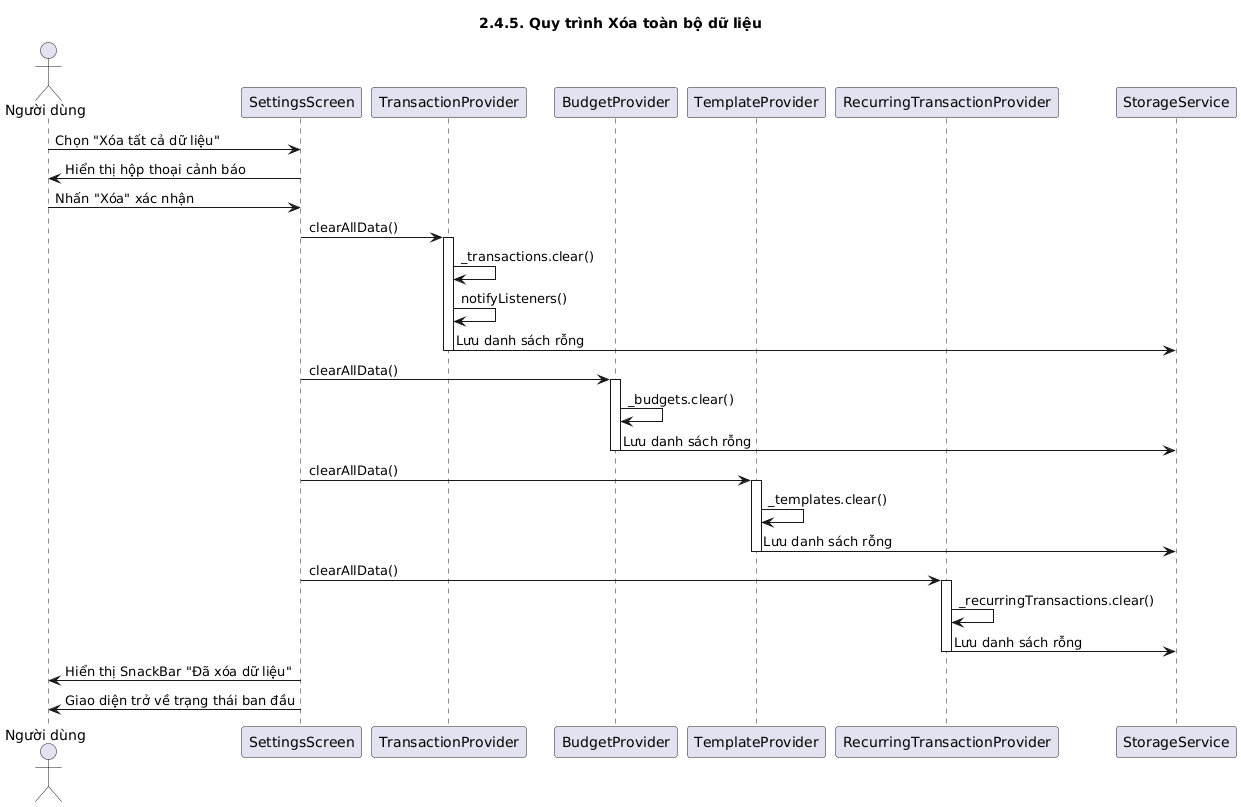
3. Quy trình tạo ngân sách



4. Quy trình tự tạp giao dịch định kì ( Hệ thống)



5. Quy trình xóa toàn bộ dữ liệu

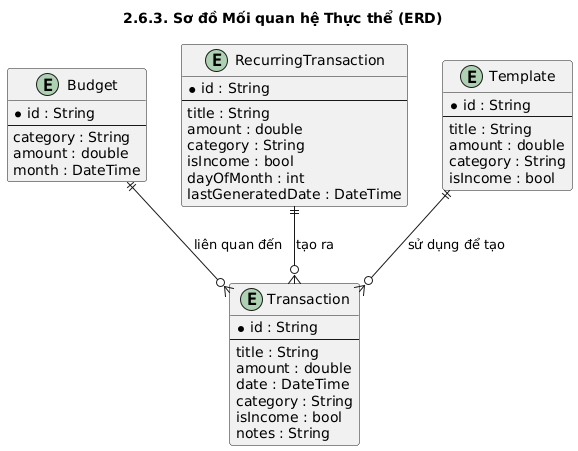


2.6 Thiết kế dữ liệu

1. Phương pháp lưu trữ

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Phương án lựa chọn** | **Giải thích** |
| **Mô hình lưu trữ** | Local Storage (lưu cục bộ trên thiết bị) | Phù hợp ứng dụng offline, không yêu cầu máy chủ. |
| **Thư viện sử dụng** | shared\_preferences (Flutter) | Cho phép lưu dữ liệu key–value đơn giản, bền vững. |
| **Định dạng lưu** | JSON (serialize/deserialize) | Mỗi loại dữ liệu được mã hóa thành chuỗi JSON theo key (transactions, budgets, templates...). |
| **Ưu điểm chính** | - Offline-first - Bảo mật & riêng tư - Hiệu năng cao | Người dùng có thể sử dụng không cần Internet, dữ liệu nằm hoàn toàn trên thiết bị, truy xuất nhanh. |
| **Cách hoạt động** | 1. Khi khởi động: đọc dữ liệu JSON → decode → nạp vào Provider. 2.Khi thay đổi: encode → lưu lại JSON mới. | Đảm bảo đồng bộ giữa bộ nhớ và dữ liệu hiển thị. |

2. Sơ đồ quan hệ thực thể



| **Mối quan hệ** | **Kiểu** | **Mô tả** |
| --- | --- | --- |
| **RecurringTransaction → Transaction** | 1 – N | Một giao dịch định kỳ có thể sinh ra nhiều giao dịch thực tế. |
| **Template → Transaction** | 1 – N | Một mẫu có thể được dùng nhiều lần để tạo giao dịch. |
| **Budget → Transaction** | 1 – N | Một ngân sách liên quan tới nhiều giao dịch trong tháng đó. |

Mối quan hệ logic:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | String | Mã định danh duy nhất | Khóa chính |
| title | String | Tên giao dịch | Bắt buộc |
| amount | double | Số tiền giao dịch | Bắt buộc, > 0 |
| date | DateTime | Ngày thực hiện | Bắt buộc |
| category | String | Danh mục chi tiêu | Bắt buộc |
| isIncome | bool | True nếu thu nhập, False nếu chi tiêu | Bắt buộc |
| notes | String | Ghi chú thêm | Tùy chọn |

Bảng 2.6.1 – Thực thể Transaction (Giao dịch)

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | String | Mã định danh duy nhất | Khóa chính |
| category | String | Danh mục được áp dụng | Bắt buộc |
| amount | double | Hạn mức tối đa chi tiêu | Bắt buộc |
| month | DateTime | Tháng áp dụng ngân sách | Bắt buộc |

Bảng 2.6.2 – Thực thể Budget (Ngân sách)

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | String | Mã định danh duy nhất | Khóa chính |
| title | String | Tên giao dịch định kỳ | Bắt buộc |
| amount | double | Số tiền giao dịch định kỳ | Bắt buộc |
| category | String | Danh mục giao dịch | Bắt buộc |
| isIncome | bool | Thu nhập hay chi tiêu | Bắt buộc |
| dayOfMonth | int | Ngày tạo giao dịch trong tháng | Bắt buộc |
| lastGeneratedDate | DateTime | Ngày cuối cùng đã tạo giao dịch | Tùy chọn |

Bảng 2.6.3 – Thực thể RecurringTransaction (Giao dịch định kỳ)

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | String | Mã định danh duy nhất | Khóa chính |
| title | String | Tên mẫu giao dịch | Bắt buộc |
| amount | double | Số tiền trong mẫu | Bắt buộc |
| category | String | Danh mục của mẫu | Bắt buộc |
| isIncome | bool | Mẫu thu nhập hay chi tiêu | Bắt buộc |

Bảng 2.6.4 – Thực thể Template (Mẫu giao dịch)

**CHƯƠNG 3: TRIỂN KHAI THIẾT KẾ HẾ THỐNG**

.

**3.1. Triển khai**

Ứng dụng "Quản lý Chi tiêu" được triển khai dựa trên các công nghệ và môi trường phát triển hiện đại, đảm bảo hiệu suất và khả năng bảo trì.

* Môi trường phát triển:
  + Hệ điều hành: Windows / macOS.
  + Công cụ lập trình (IDE): Visual Studio Code.
  + Hệ quản trị phiên bản: Git.
* Công nghệ sử dụng:
  + Nền tảng (Framework): Flutter phiên bản 3.x.
  + Ngôn ngữ lập trình: Dart.
  + Kiến trúc: Phân lớp (Layered Architecture) kết hợp với mô hình quản lý trạng thái Provider.
  + Thư viện chính:

provider: Quản lý trạng thái ứng dụng.

shared\_preferences: Lưu trữ dữ liệu cục bộ trên thiết bị.

google\_fonts: Sử dụng các font chữ tùy chỉnh để tăng tính thẩm mỹ.

intl: Định dạng ngày tháng và số liệu.

Quá trình triển khai được thực hiện bằng cách xây dựng các màn hình (UI) theo thiết kế, sau đó kết nối chúng với các lớp xử lý logic (Providers). Dữ liệu được lưu và tải thông qua một lớp dịch vụ (StorageService), giúp tách biệt logic nghiệp vụ khỏi cơ chế lưu trữ cụ thể.

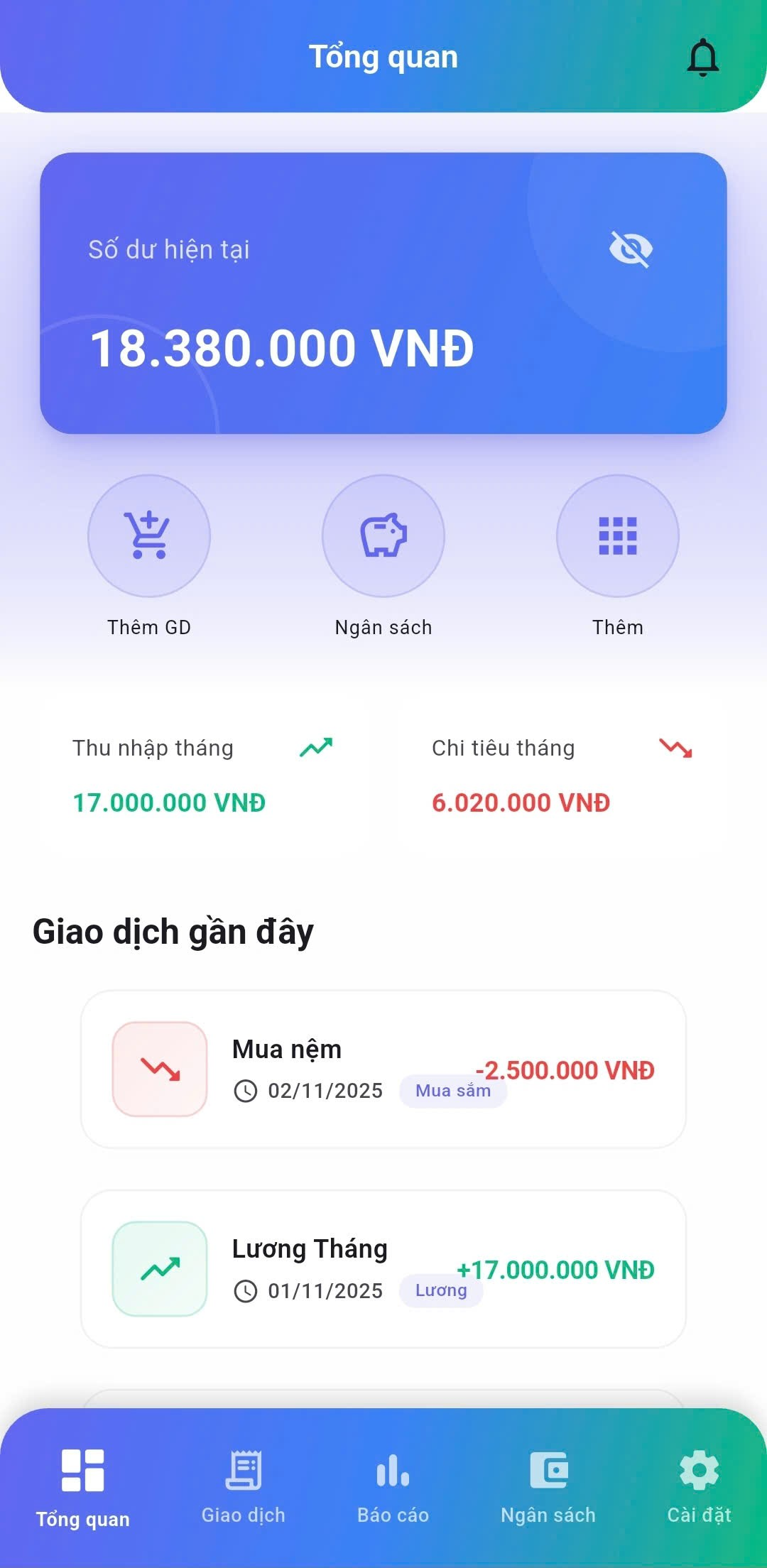
**3.2. Kết quả đạt được**

Sau quá trình triển khai, ứng dụng đã hoàn thành tất cả các chức năng cốt lõi được đề ra trong giai đoạn thiết kế, đáp ứng đầy đủ nhu cầu quản lý tài chính cá nhân của người dùng. Cụ thể:

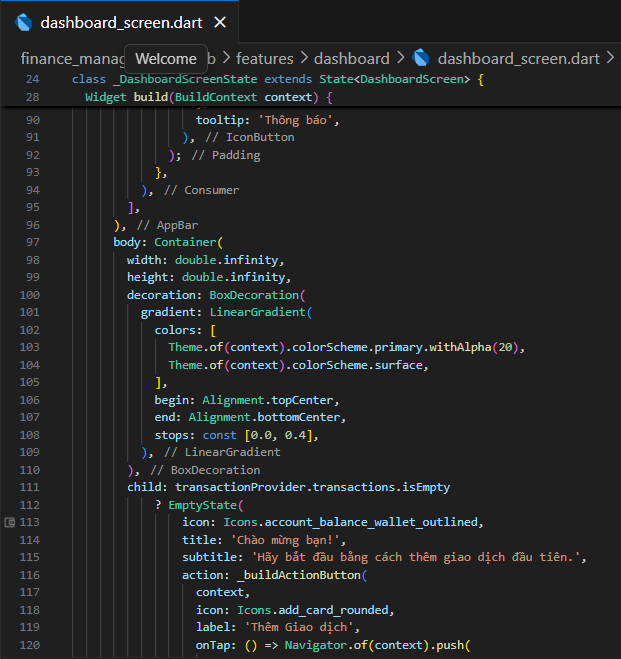
1. Quản lý Giao dịch: Người dùng có thể thêm, sửa, xóa các giao dịch thu/chi một cách dễ dàng. Lịch sử giao dịch được hiển thị rõ ràng và có thể lọc theo thời gian.
2. Bảng điều khiển Tổng quan: Màn hình chính cung cấp cái nhìn nhanh về số dư, tổng thu/chi trong tháng và các giao dịch gần nhất.
3. Báo cáo và Phân tích: Ứng dụng cung cấp báo cáo tháng với biểu đồ tròn trực quan, giúp người dùng phân tích cơ cấu chi tiêu.
4. Quản lý Ngân sách: Chức năng cho phép tạo hạn mức chi tiêu cho từng danh mục và theo dõi tiến độ thực hiện.
5. Tự động hóa:
   * Giao dịch định kỳ: Hệ thống có khả năng tự động tạo các giao dịch lặp lại hàng tháng (lương, thuê nhà...).
   * Mẫu giao dịch: Người dùng có thể tạo sẵn các mẫu để nhập liệu nhanh hơn.
6. Hệ thống Cảnh báo: Ứng dụng tự động gửi thông báo khi số dư xuống thấp hoặc khi chi tiêu vượt ngân sách, giúp người dùng kiểm soát tài chính tốt hơn.
7. Quản lý Dữ liệu và Cài đặt: Cung cấp các tùy chọn quản lý dữ liệu như xóa toàn bộ dữ liệu, quản lý mẫu, quản lý giao dịch định kỳ và truy cập màn hình hướng dẫn sử dụng.

3.3 Giao diện chính thức

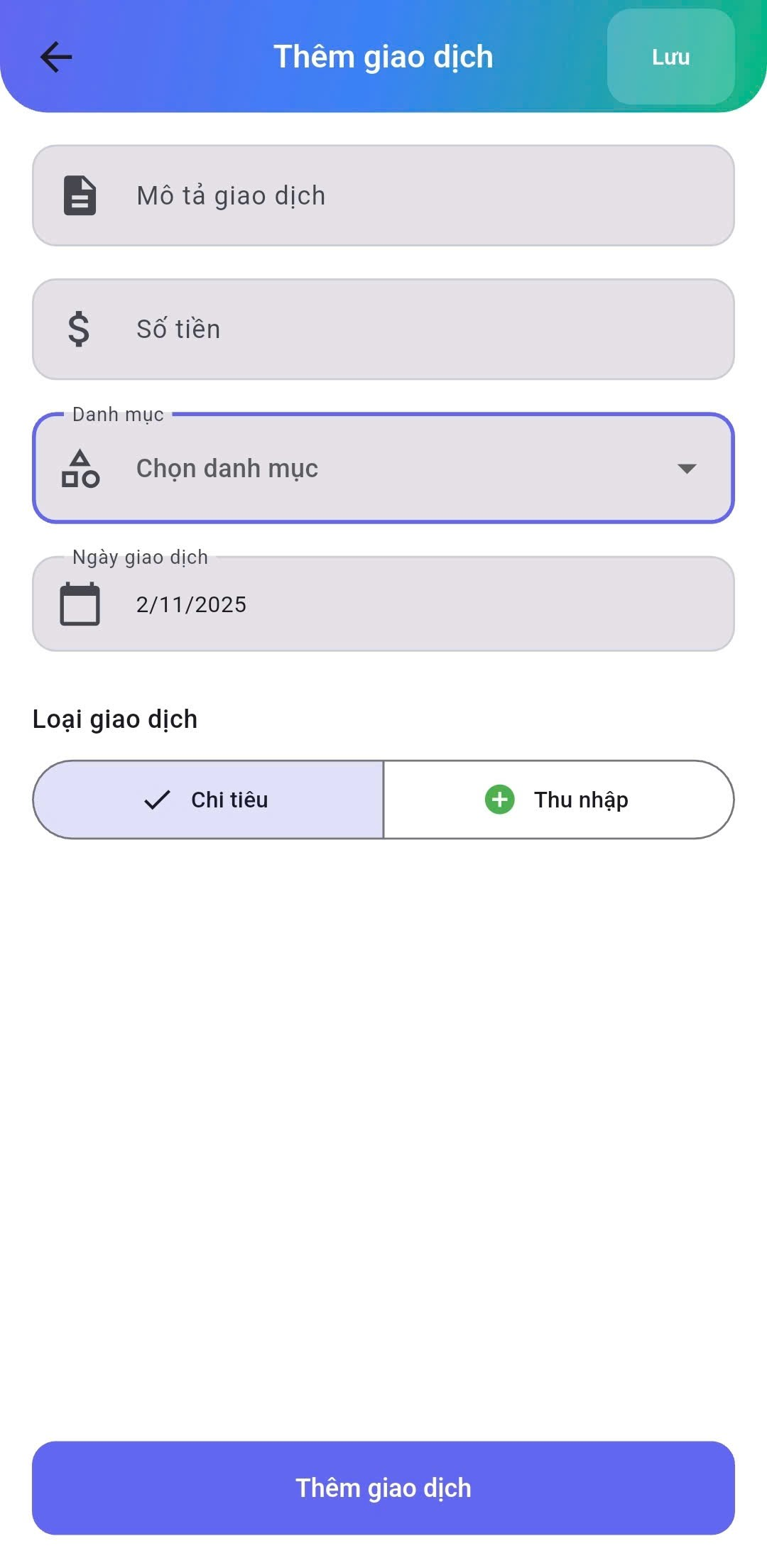
1. Trang tổng quan



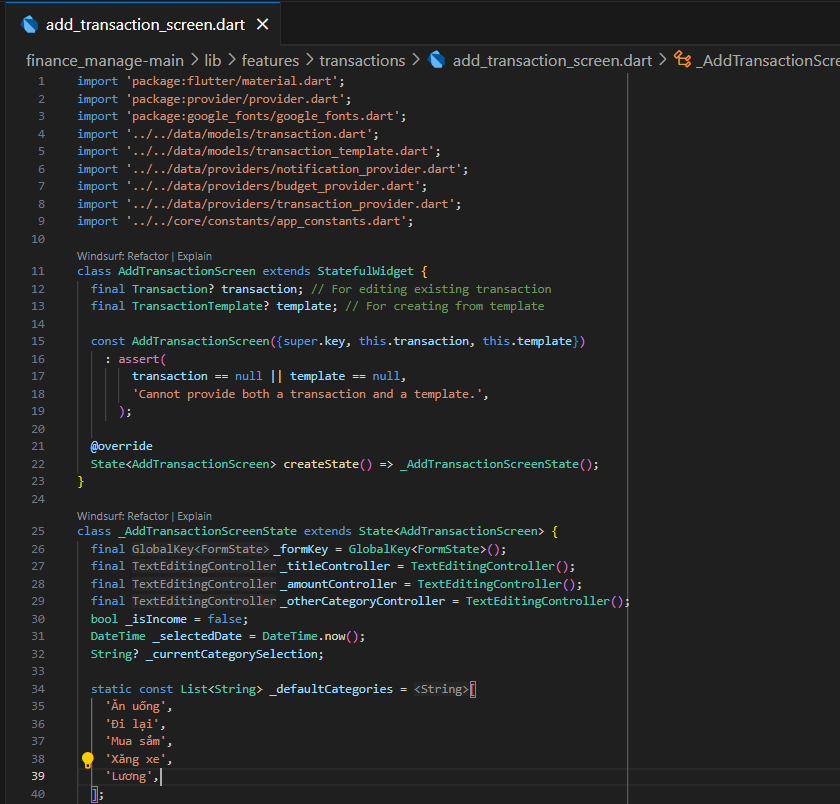
Mã code đại diện



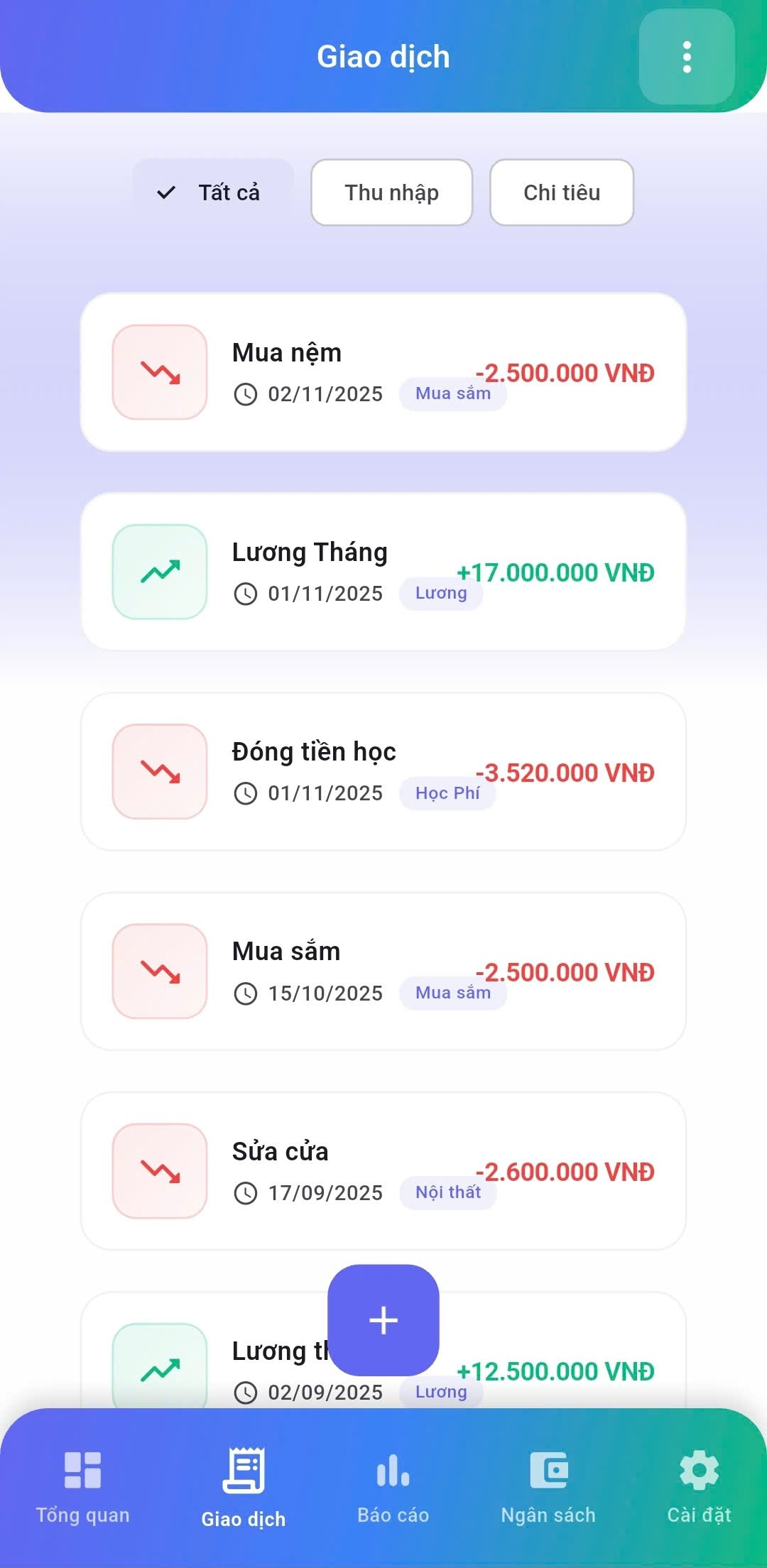
2. Trang thêm mới giao dịch



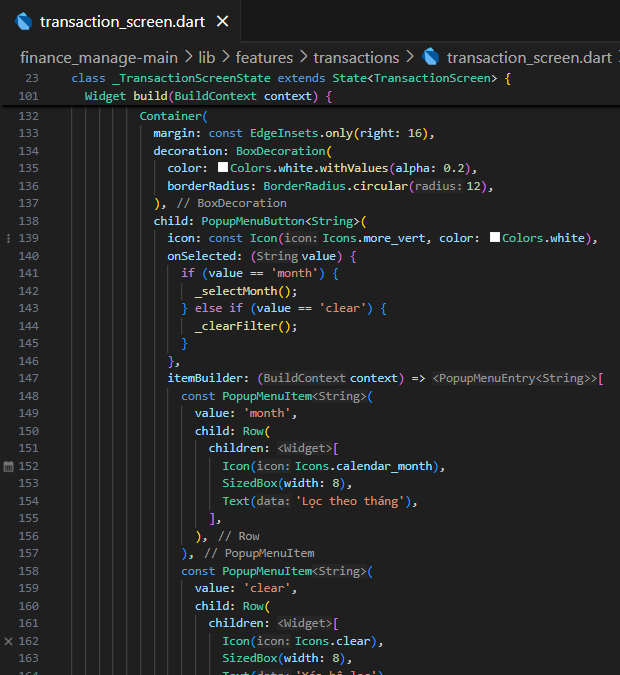
Mã code đại diện



3. Trang hiển thị danh sách giao dịch



Mã code đại diện



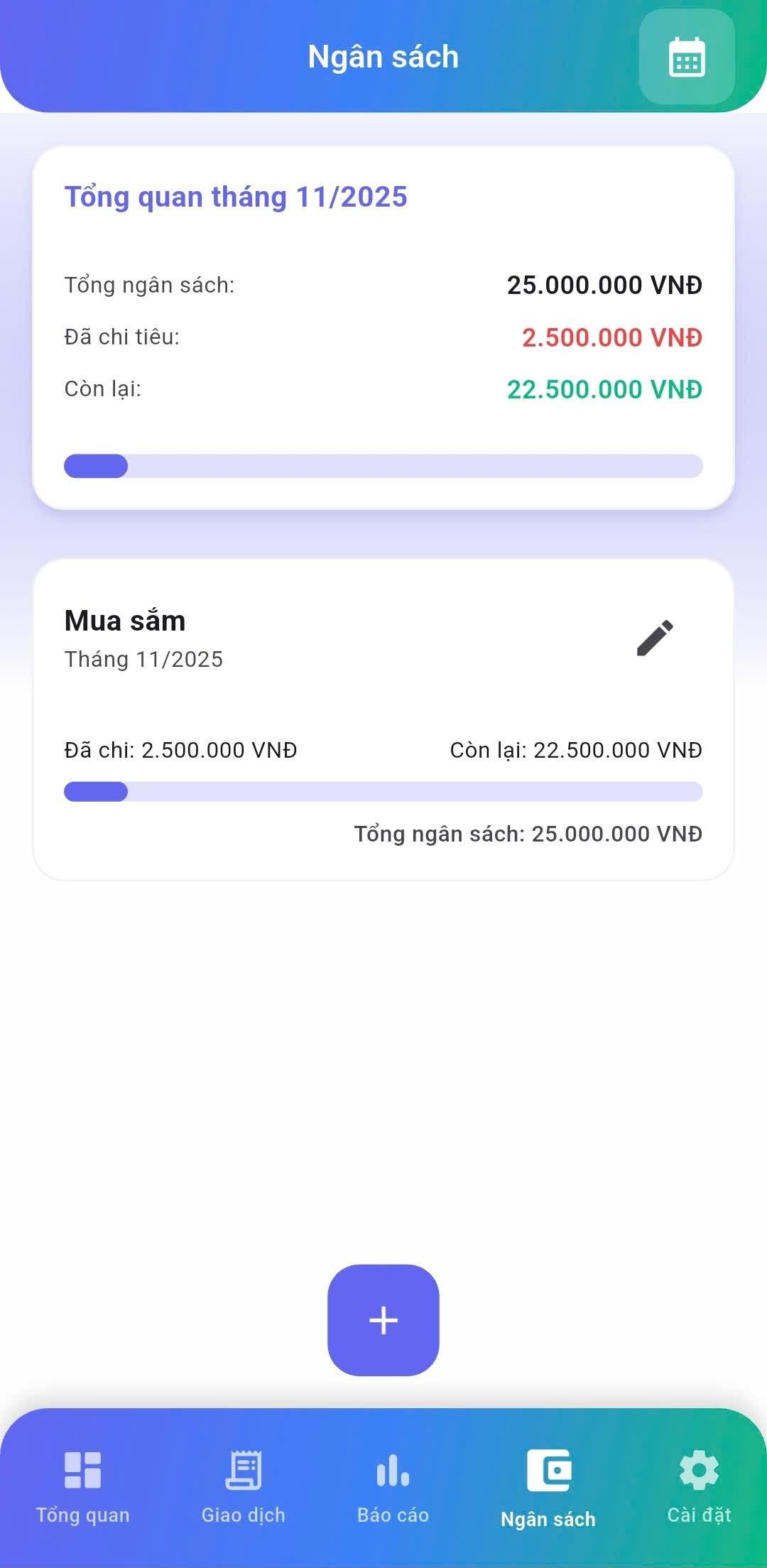
4. Trang hiện thị báo cáo



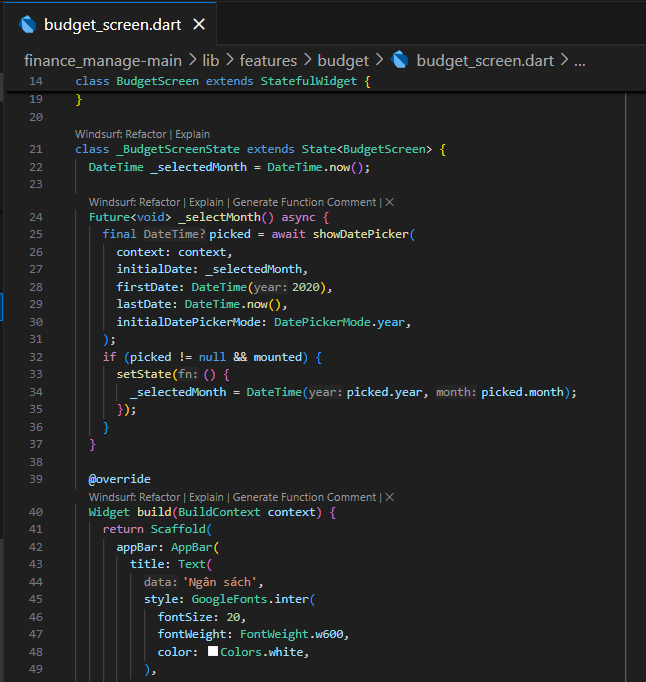
Mã code đại diện



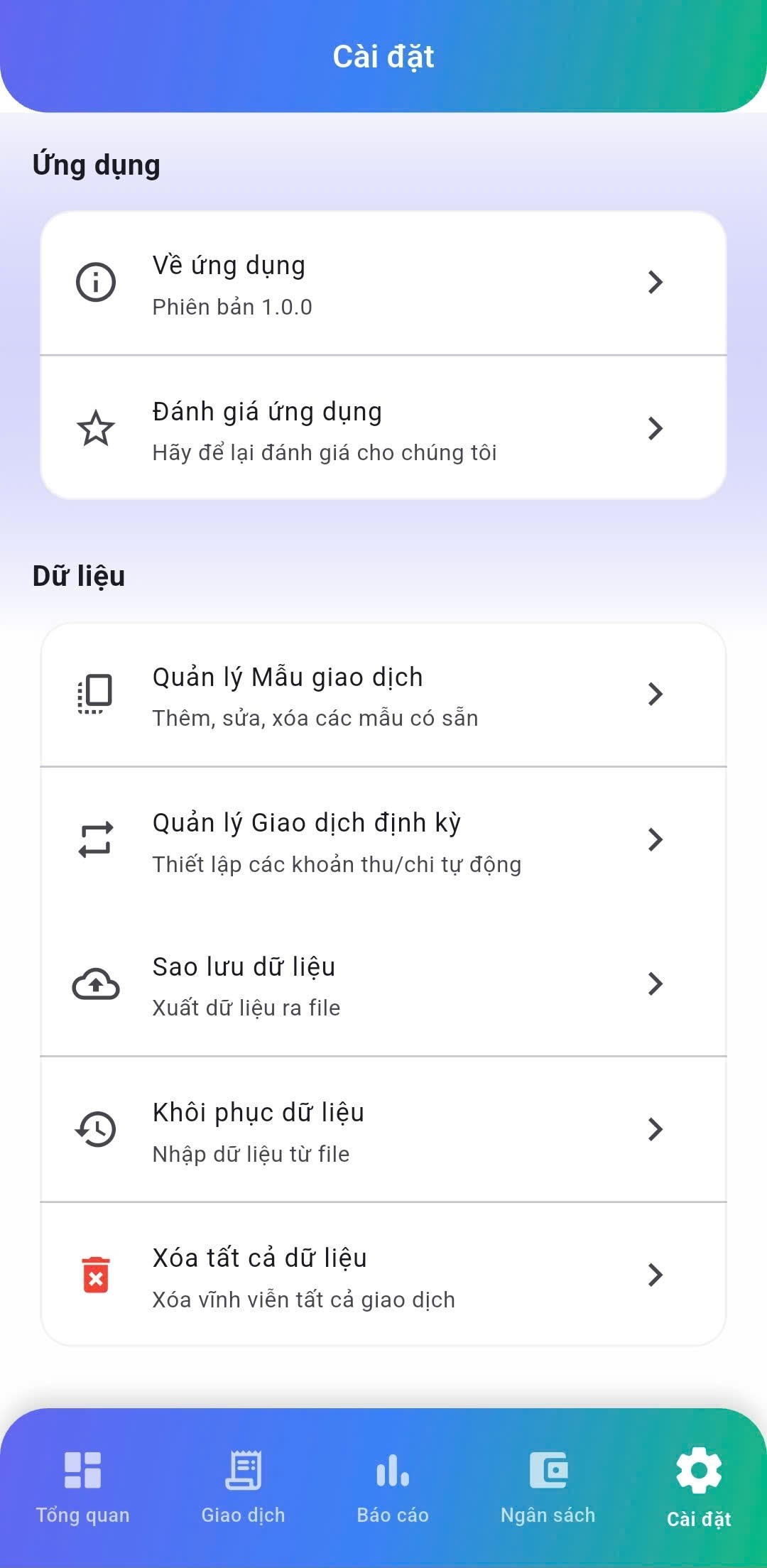
5. Trang hiển thị ngân sách



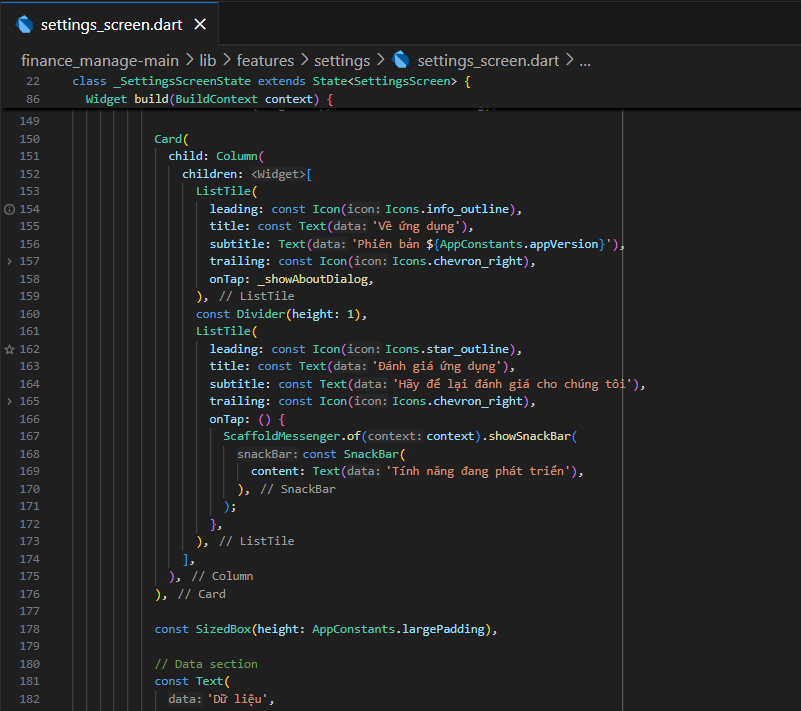
Mã code đại diện



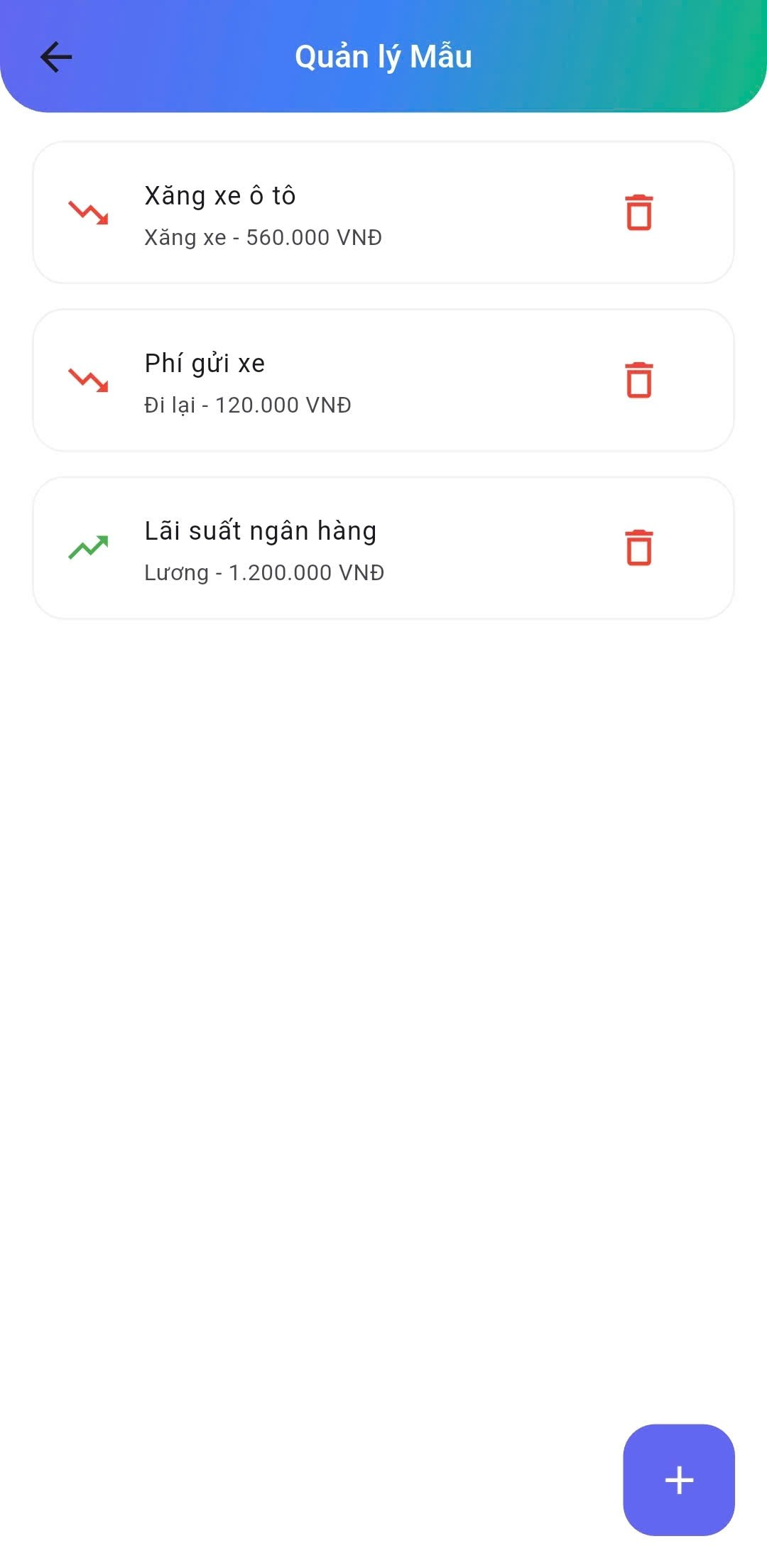
6. Trang hiển thị cài đặt



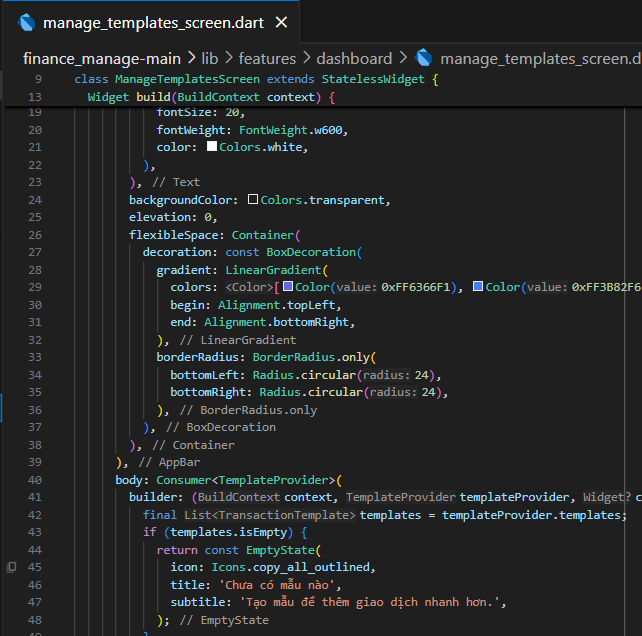
Mã code đại diện



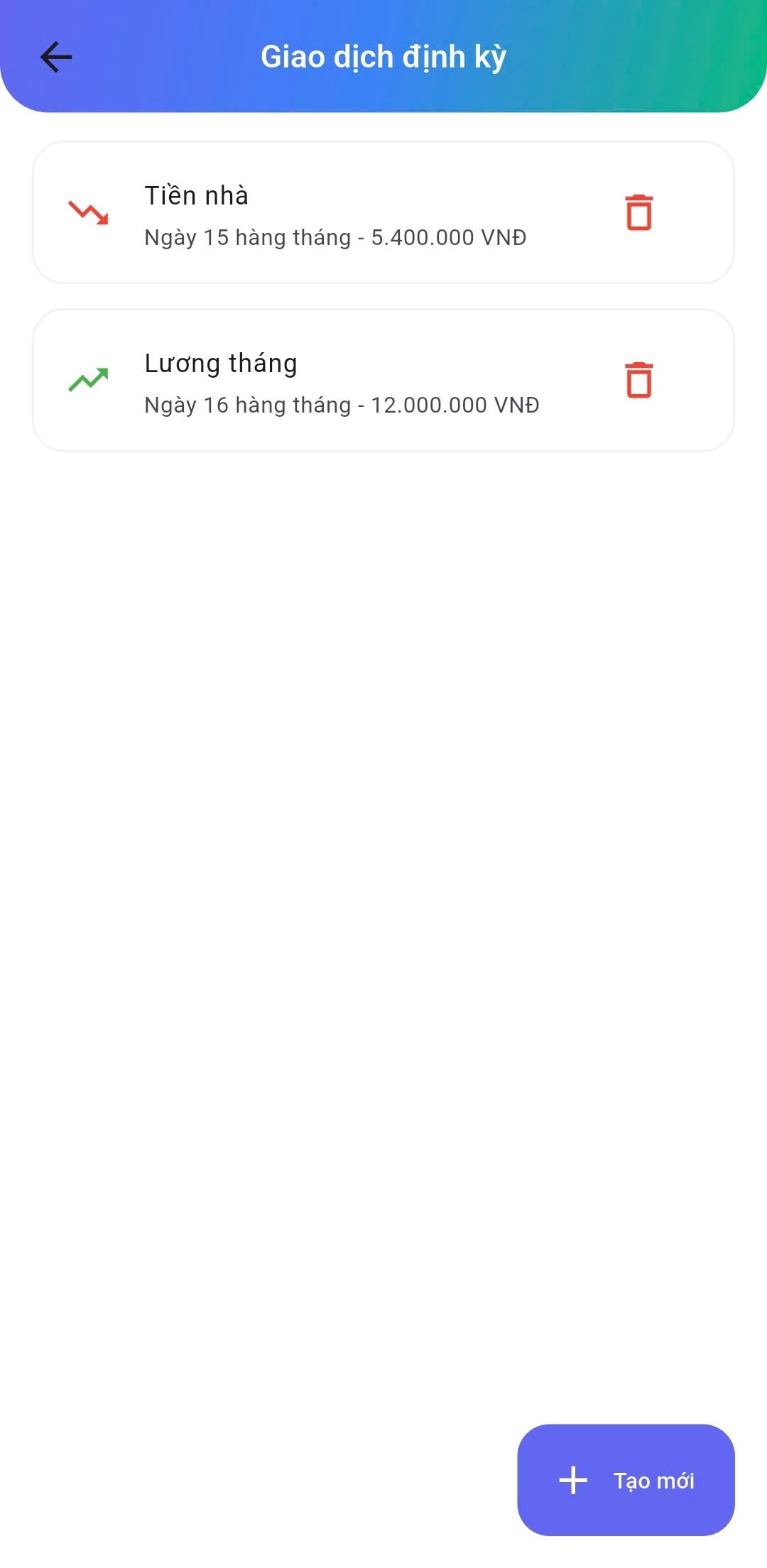
7. Trang quản lí mẫu giao dịch



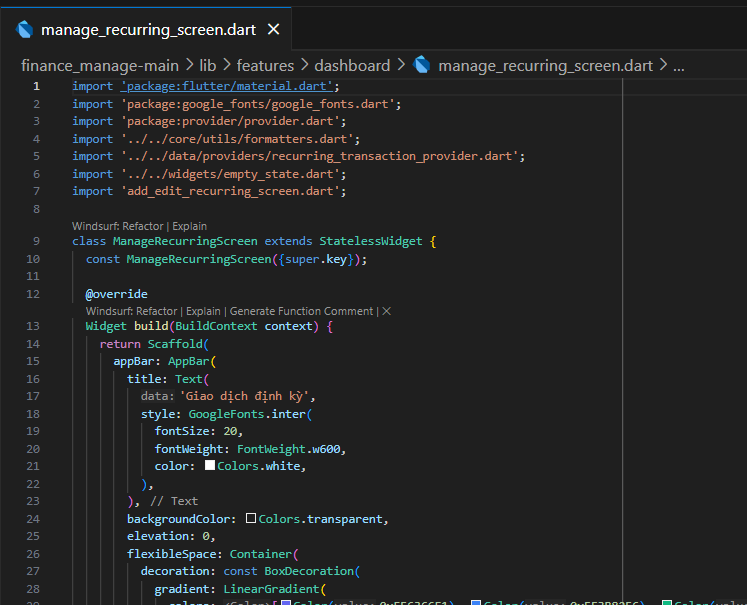
Mã code đại diện



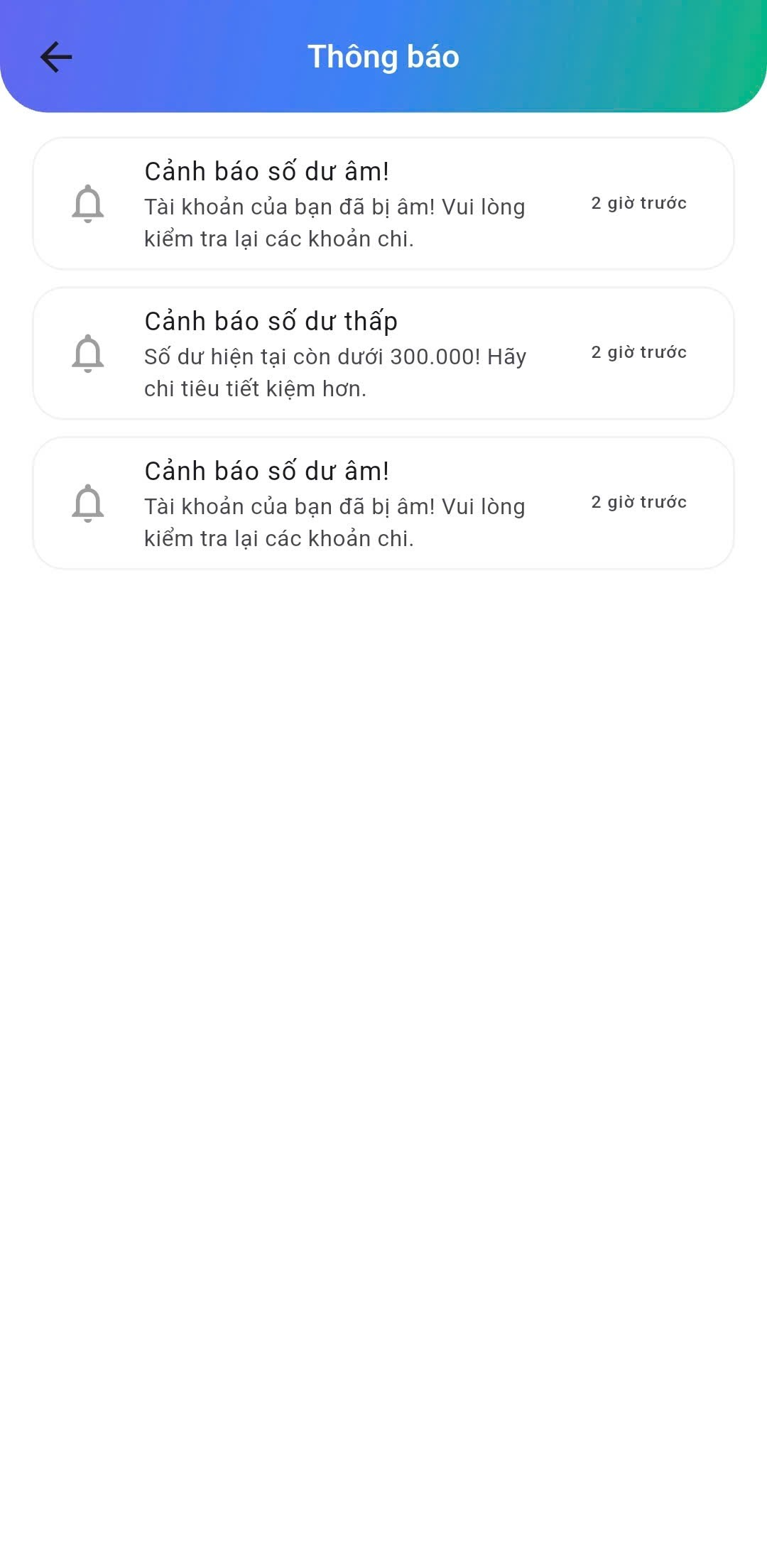
8. Trang quản lí giao dịch định kì



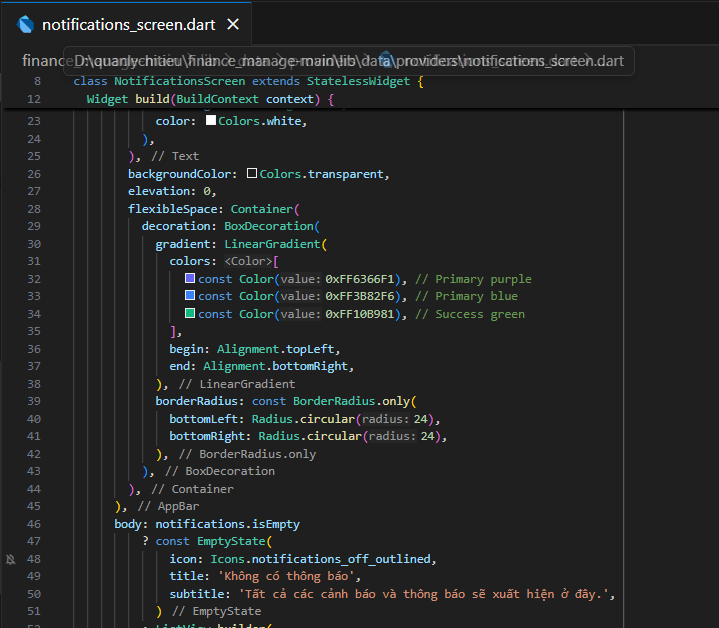
Mã code đại diện



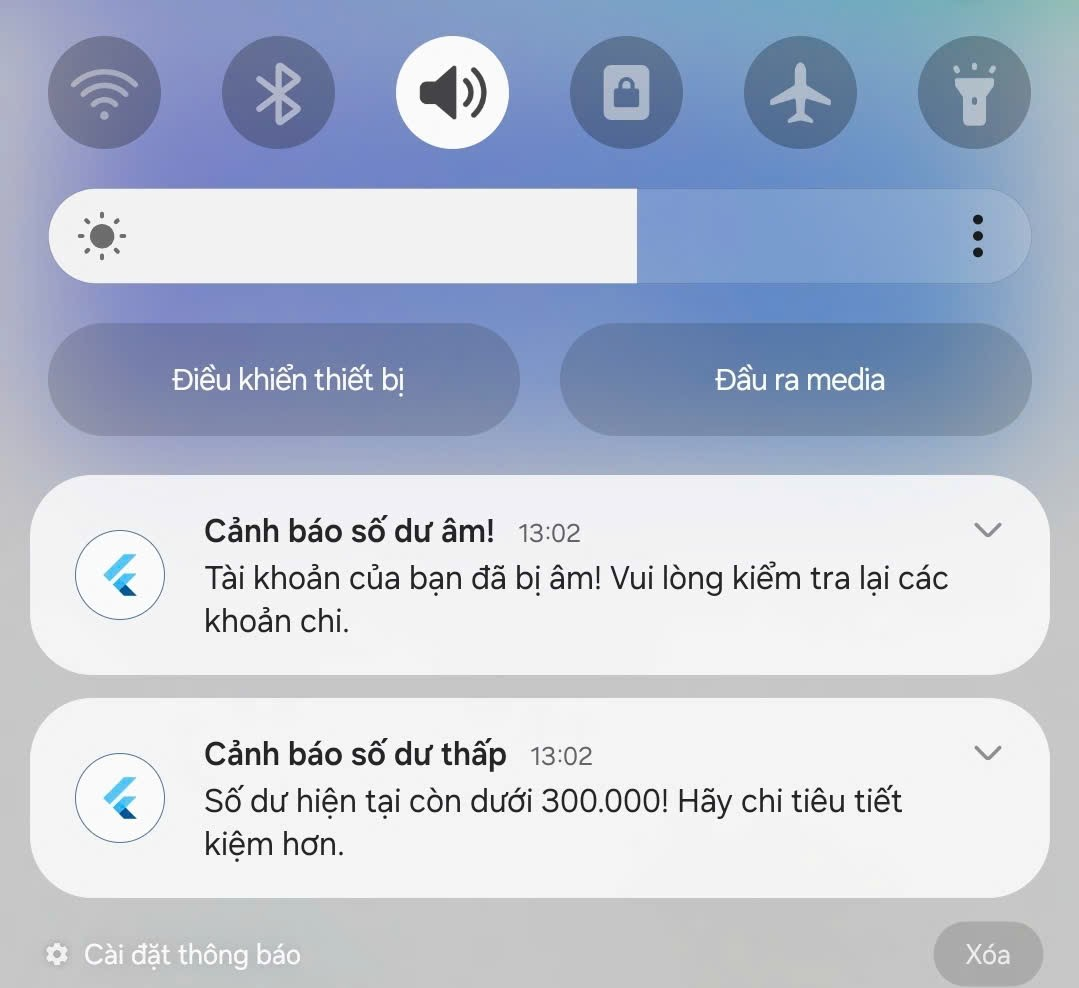
9. Trang quản lí thông báo



Mã code đại diện



10. Phần hiển thị thông báo trên thiết bị



Mã code đại diện



**CHƯƠNG 4: TỔNG KẾT VÀ ĐÁNH GIÁ**

**4.1. Kết quả đạt được**

Ứng dụng đã xây dựng thành công một hệ thống quản lý tài chính cá nhân hoàn chỉnh, hoạt động ổn định trên nền tảng di động với các chức năng chính sau:

1. **Quản lý giao dịch toàn diện:**
   * Người dùng có thể dễ dàng thêm, sửa, xóa các giao dịch thu nhập và chi tiêu.
   * Hệ thống cho phép phân loại giao dịch theo các danh mục có sẵn, giúp việc thống kê trở nên mạch lạc.
   * Lịch sử giao dịch được hiển thị rõ ràng, đi kèm bộ lọc theo tháng và loại giao dịch, giúp người dùng tra cứu thông tin nhanh chóng.
2. **Báo cáo và phân tích trực quan:**
   * Màn hình tổng quan (Dashboard) cung cấp một cái nhìn nhanh về tình hình tài chính: số dư hiện tại, tổng thu/chi trong tháng và các giao dịch gần nhất.
   * Chức năng báo cáo tháng sử dụng biểu đồ tròn để phân tích tỷ lệ thu/chi, giúp người dùng dễ dàng nhận biết cơ cấu dòng tiền của mình.
3. **Hệ thống ngân sách và cảnh báo chủ động:**
   * Người dùng có thể thiết lập hạn mức chi tiêu (ngân sách) cho từng danh mục cụ thể theo tháng.
   * Ứng dụng tự động theo dõi và cảnh báo khi người dùng sắp chi tiêu vượt ngân sách hoặc khi số dư tài khoản xuống thấp, giúp nâng cao kỷ luật tài chính.
4. **Tự động hóa và tiện ích thông minh:**
   * **Giao dịch định kỳ:** Chức năng tự động tạo các giao dịch lặp lại hàng tháng (như lương, tiền thuê nhà) đã được triển khai thành công, giúp tiết kiệm thời gian và giảm thiểu sai sót cho người dùng.
   * **Mẫu giao dịch:** Cho phép người dùng tạo sẵn các mẫu giao dịch thường dùng, giúp việc nhập liệu trở nên nhanh chóng hơn.
5. **Kiến trúc và công nghệ hiện đại:**
   * Ứng dụng được xây dựng trên nền tảng Flutter, cho phép phát triển và triển khai trên cả Android và iOS từ một cơ sở mã nguồn duy nhất.
   * Áp dụng mô hình quản lý trạng thái Provider, giúp tách biệt logic nghiệp vụ và giao diện, làm cho mã nguồn trở nên sạch sẽ, dễ bảo trì và mở rộng.
   * Dữ liệu được lưu trữ cục bộ trên thiết bị người dùng thông qua shared\_preferences, đảm bảo tính riêng tư và cho phép ứng dụng hoạt động ngoại tuyến.

**4.2. Hạn chế của ứng dụng**

Bên cạnh những kết quả đã đạt được, ứng dụng vẫn còn một số hạn chế cần được cải thiện:

1. **Lưu trữ dữ liệu đơn giản:** Việc sử dụng shared\_preferences để lưu trữ toàn bộ dữ liệu dưới dạng chuỗi JSON là phù hợp cho quy mô nhỏ, nhưng sẽ gặp vấn đề về hiệu năng và khả năng truy vấn khi lượng giao dịch của người dùng tăng lên. Giải pháp này cũng không hỗ trợ các truy vấn phức tạp.
2. **Thiếu tính năng đồng bộ hóa và sao lưu:** Dữ liệu hiện chỉ được lưu trên một thiết bị duy nhất. Người dùng sẽ mất toàn bộ dữ liệu nếu đổi hoặc mất điện thoại. Các chức năng "Sao lưu" và "Khôi phục" đã có trên giao diện nhưng chưa được triển khai.
3. **Khả năng tùy biến còn hạn chế:** Người dùng chưa thể tự thêm, sửa, xóa các danh mục thu/chi mà phải sử dụng các danh mục được định sẵn trong mã nguồn.
4. **Chưa hỗ trợ đa ngôn ngữ:** Mặc dù cấu trúc cho việc đa ngôn ngữ (localization) đã được thiết lập (AppLocalizations), nhưng ứng dụng hiện tại vẫn đang hiển thị các chuỗi văn bản bằng tiếng Việt một cách tĩnh, chưa thực sự chuyển đổi ngôn ngữ.
5. **Thiếu các tính năng hỗ trợ nâng cao:** Các mục như "Đánh giá ứng dụng", "FAQs", "Liên hệ" trong phần cài đặt vẫn đang ở trạng thái "tính năng đang phát triển", làm giảm trải nghiệm hỗ trợ người dùng.

**4.3. Hướng phát triển trong tương lai**

Để khắc phục các hạn chế trên và nâng cao giá trị cho người dùng, các hướng phát triển sau đây được đề xuất:

1. **Nâng cấp hệ thống lưu trữ:**
   * Chuyển đổi từ shared\_preferences sang một cơ sở dữ liệu cục bộ mạnh mẽ hơn như **SQLite** (sử dụng gói sqflite) hoặc **Hive**. Điều này sẽ cải thiện đáng kể hiệu năng, khả năng truy vấn và quản lý dữ liệu lớn.
2. **Xây dựng tính năng đồng bộ hóa đám mây:**
   * Tích hợp với các dịch vụ backend (như Firebase Firestore) để cho phép người dùng đăng nhập và đồng bộ hóa dữ liệu trên nhiều thiết bị.
   * Hoàn thiện chức năng **sao lưu và khôi phục dữ liệu** thủ công (ví dụ: xuất ra file .json hoặc .csv) như một giải pháp thay thế trước khi có đồng bộ hóa tự động.
3. **Tăng cường khả năng tùy biến:**
   * Xây dựng màn hình cho phép người dùng **quản lý danh mục** (thêm, sửa, xóa, chọn biểu tượng) theo nhu cầu cá nhân.
   * Cho phép người dùng tùy chỉnh các ngưỡng cảnh báo (số dư thấp, cảnh báo ngân sách).
4. **Hoàn thiện đa ngôn ngữ và các tính năng hỗ trợ:**
   * Hoàn thiện việc triển khai đa ngôn ngữ (ví dụ: tiếng Anh) để tiếp cận tệp người dùng rộng hơn.
   * Xây dựng nội dung cho các mục "FAQs", "Điều khoản dịch vụ", và tích hợp các công cụ để người dùng có thể "Gửi góp ý" hoặc "Liên hệ" trực tiếp từ ứng dụng.
5. **Mở rộng các tính năng phân tích:**
   * Thêm các loại biểu đồ mới (biểu đồ cột, biểu đồ đường) để phân tích chi tiêu theo danh mục hoặc theo dõi xu hướng tài chính qua các tháng.
   * Phát triển tính năng "Ví tài chính", cho phép người dùng quản lý nhiều nguồn tiền khác nhau (ví dụ: tiền mặt, tài khoản ngân hàng, ví điện tử).

**CHƯƠNG 5: TÀI LIỆU THAM KHẢO**

Trong quá trình nghiên cứu, thiết kế và phát triển ứng dụng "Quản lý Chi tiêu", nhóm đã tham khảo và sử dụng các tài liệu, công nghệ và thư viện từ các nguồn uy tín sau đây:

**1. Nền tảng và Ngôn ngữ lập trình:**

* **Flutter:** Trang chủ chính thức của Flutter, cung cấp tài liệu, hướng dẫn và các ví dụ về cách xây dựng ứng dụng đa nền tảng.
  + *URL:* https://flutter.dev/
* **Dart:** Trang chủ chính thức của ngôn ngữ Dart, cung cấp tài liệu chi tiết về cú pháp, thư viện và các khái niệm lập trình.
  + *URL:* https://dart.dev/

**2. Các thư viện (Packages) đã sử dụng:**

* **provider:** Thư viện quản lý trạng thái (state management) được sử dụng để quản lý và chia sẻ dữ liệu ứng dụng (giao dịch, ngân sách) một cách hiệu quả.

*URL:* https://pub.dev/packages/provider

* **shared\_preferences:** Thư viện dùng để lưu trữ dữ liệu đơn giản (key-value) cục bộ trên thiết bị. Trong dự án, nó được dùng để lưu danh sách giao dịch, ngân sách, và các cài đặt của người dùng.

*URL:* https://pub.dev/packages/shared\_preferences

* **fl\_chart:** Một thư viện mạnh mẽ để vẽ các loại biểu đồ. Dự án đã sử dụng thư viện này để tạo biểu đồ tròn và biểu đồ cột trong màn hình "Báo cáo".

*URL:* https://pub.dev/packages/fl\_chart

* **flutter\_local\_notifications:** Thư viện dùng để hiển thị các thông báo cục bộ trên thiết bị, phục vụ cho việc gửi cảnh báo về ngân sách, số dư và thông báo tạo giao dịch định kỳ.

*URL:* https://pub.dev/packages/flutter\_local\_notifications

* **intl:** Cung cấp các công cụ quốc tế hóa và bản địa hóa, được sử dụng trong dự án để định dạng ngày tháng và số tiền theo chuẩn Việt Nam.

*URL:* https://pub.dev/packages/intl

* **google\_fonts:** Cho phép dễ dàng tích hợp và sử dụng các phông chữ từ Google Fonts, giúp giao diện ứng dụng trở nên chuyên nghiệp và nhất quán hơn (ví dụ: phông chữ Inter).

*URL:* https://pub.dev/packages/google\_fonts

**3. Cộng đồng và các nguồn tài liệu khác:**

* **Stack Overflow:** Một diễn đàn hỏi đáp lớn dành cho lập trình viên, nơi cung cấp các giải pháp cho nhiều vấn đề kỹ thuật gặp phải trong quá trình phát triển.

*URL:* https://stackoverflow.com/

* **Medium & các blog công nghệ:** Các bài viết, hướng dẫn chuyên sâu về các kỹ thuật trong Flutter, đặc biệt là về quản lý trạng thái và kiến trúc ứng dụng.