

Technical Documentation

Table of Contents

1. INTRODUCTION	1
2. Database Schema (Lược đồ cơ sở dữ liệu)	3
3. SYSTEM ARCHITECTURE & DESIGN	7
4. SYSTEM LOGIC & DATA FLOWS.....	9
5. THIRD-PARTY LIBRARIES & IMPLEMENTATION	13
6. CONCLUSION & FUTURE WORK.....	14

1. INTRODUCTION

1.1. Purpose of the Document

Tài liệu này đóng vai trò là **Tài liệu Thiết kế Kỹ thuật (Technical Design Document - TDD)** cho dự án **CampusExpense Manager**. Mục đích chính của tài liệu là cung cấp cái nhìn sâu sắc và toàn diện về kiến trúc hệ thống, cấu trúc dữ liệu, và luồng xử lý logic bên trong ứng dụng.

Tài liệu này được biên soạn để phục vụ cho:

- **Developers:** Hiểu rõ cấu trúc mã nguồn (Source code), sơ đồ lớp (Class Diagram) và cách thức hoạt động của các mô-đun để bảo trì hoặc mở rộng tính năng trong tương lai.
- **Assessors/Stakeholders:** Minh chứng cho tính phức tạp kỹ thuật và khối lượng công việc lập trình (coding effort) đã thực hiện, đáp ứng các yêu cầu khắt khe về Offline-first, bảo mật và hiệu năng¹¹¹.

1.2. Project Overview

CampusExpense Manager là ứng dụng di động Native Android được phát triển bởi đội ngũ BudgetWise Solutions, nhằm giải quyết bài toán quản lý tài chính cá nhân cho sinh viên đại học tại Việt Nam².

Điểm khác biệt cốt lõi của hệ thống là kiến trúc **Offline-first** (Ưu tiên ngoại tuyến). Ứng dụng hoạt động hoàn toàn độc lập trên thiết bị người dùng, không phụ thuộc vào kết nối Internet hay máy

chủ đám mây (Cloud Server), đảm bảo quyền riêng tư dữ liệu tuyệt đối và khả năng truy cập tức thì trong mọi điều kiện mạng³.

1.3. Technology Stack

Dự án được xây dựng dựa trên nền tảng công nghệ ổn định và phổ biến nhất cho phát triển Android hiện nay, đảm bảo khả năng tương thích ngược (Backward Compatibility) tốt.

Component	Specification	Description
Language	Java 11	Ngôn ngữ lập trình chính, đảm bảo tính chặt chẽ và hướng đối tượng (OOP) ⁴ .
IDE	Android Studio Iguana	Phiên bản 2024.2.1, hỗ trợ các công cụ debug và layout inspector mới nhất ⁵ .
Min SDK	API 29 (Android 10)	Đảm bảo hỗ trợ >98% thiết bị sinh viên, tương thích tốt với Scoped Storage ⁶ .
Target SDK	API 36	Tuân thủ các tiêu chuẩn bảo mật và hiệu năng mới nhất của Google Play ⁷ .
Database	SQLite	Cơ sở dữ liệu quan hệ nhúng, truy xuất trực tiếp qua SQLiteOpenHelper (không dùng Room để tối ưu hóa kiểm soát query) ⁸ .
Architecture	MVC / Layered	Mô hình phân lớp rõ ràng: Activities (View/Controller) tách biệt với Models và DatabaseHelper (Model) ⁹ .

1.4. Third-Party Libraries & Dependencies

Để tối ưu hóa thời gian phát triển và nâng cao trải nghiệm người dùng, hệ thống tích hợp các thư viện mã nguồn mở đã được kiểm chứng¹⁰:

1. **MPAndroidChart:** Dùng để vẽ biểu đồ Tròn (Pie Chart) cho phân bố chi tiêu và biểu đồ Đường (Line Chart) cho xu hướng 6 tháng.
2. **iText7:** Thư viện xử lý PDF mạnh mẽ, dùng để tạo báo cáo tài chính chuyên nghiệp với bảng biểu (Table) và định dạng trang in.
3. **AndroidX WorkManager:** Quản lý các tác vụ nền (Background Tasks) tin cậy để tự động tạo các giao dịch định kỳ (Recurring Expenses) ngay cả khi tắt ứng dụng.

4. **AndroidX Biometric**: Cung cấp lớp bảo mật sinh trắc học (Vân tay/Khuôn mặt) chuẩn hóa trên nhiều thiết bị.

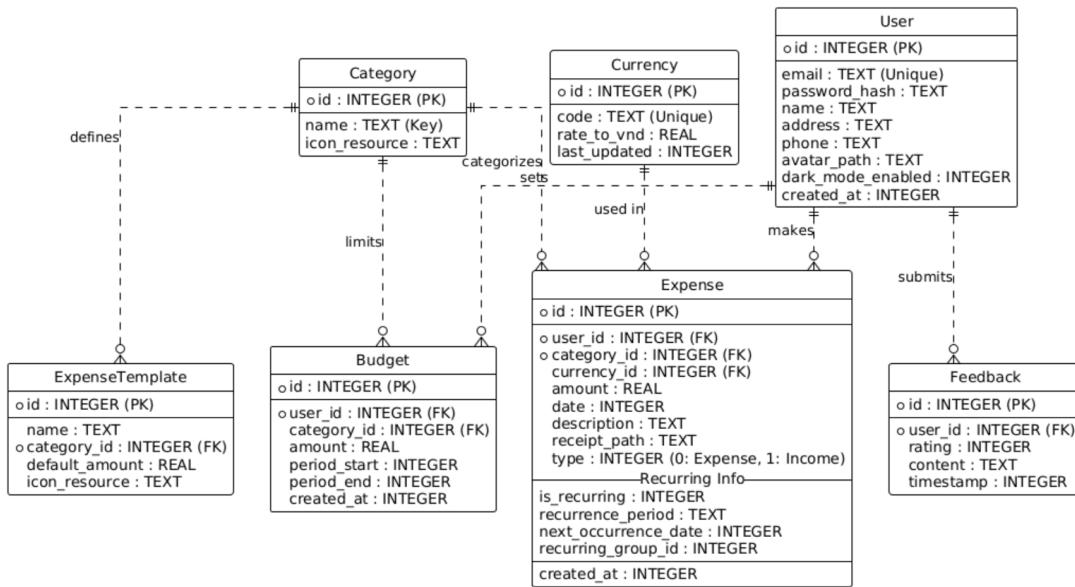
1.5. System Architecture Overview

Hệ thống tuân theo kiến trúc **Monolithic Mobile Application** với thiết kế **Layered Architecture** (Kiến trúc phân lớp) để đảm bảo tính Modularity (Mô-đun hóa) và Maintainability (Khả năng bảo trì)¹¹.

- **Presentation Layer (UI)**: Bao gồm các Activity và XML Layouts, chịu trách nhiệm hiển thị dữ liệu và nhận tương tác từ người dùng.
- **Business Logic Layer**: Các Adapter, CurrencyConverter, RecurringExpenseWorker xử lý các nghiệp vụ tính toán, định dạng tiền tệ và logic lặp lại.
- **Data Access Layer (DAL)**: Class DatabaseHelper đóng vai trò là cổng giao tiếp duy nhất với SQLite, thực hiện các câu lệnh SQL phức tạp (CRUD).
- **Data Layer**: Các Model (POJO) như User, Expense, Budget đại diện cho cấu trúc dữ liệu thực thể.

2. Database Schema (Lược đồ cơ sở dữ liệu)

2.1. Entity Relationship Diagram (ERD) (update)



2.2. Data Dictionary (Từ điển dữ liệu)

1. Table: users

Column Name	Data Type	Constraints	Description
id	INTEGER	PK, Auto Increment	Định danh duy nhất (ID) của người dùng.
email	TEXT	UNIQUE, NOT NULL	Email đăng nhập (không được trùng).
password_hash	TEXT	NOT NULL	Mật khẩu đã mã hóa (SHA-256).
name	TEXT	NOT NULL	Tên hiển thị người dùng.
address	TEXT	Nullable	Địa chỉ liên hệ.
phone	TEXT	Nullable	Số điện thoại.
avatar_path	TEXT	Nullable	Đường dẫn file ảnh đại diện trên máy.
dark_mode_enabled	INTEGER	Default 0	1: Bật Dark Mode, 0: Tắt.
created_at	INTEGER	NOT NULL	Thời gian tạo tài khoản (Unix Timestamp).

2. Table: categories

Column Name	Data Type	Constraints	Description
id	INTEGER	PK, Auto Increment	Định danh danh mục.
name	TEXT	NOT NULL, Unique	Mã danh mục (vd: 'cat_food') dùng để đa ngôn ngữ.

icon_resource	TEXT	NOT NULL	Tên file icon trong res/drawable (vd: 'ic_food').
----------------------	------	----------	---

3. Table: expenses

Column Name	Data Type	Constraints	Description
id	INTEGER	PK, Auto Increment	Định danh giao dịch.
user_id	INTEGER	FK (users.id)	ID người tạo giao dịch.
category_id	INTEGER	FK (categories.id)	ID danh mục chi tiêu.
currency_id	INTEGER	FK (currencies.id)	ID loại tiền tệ.
amount	REAL	NOT NULL	Số tiền giao dịch.
date	INTEGER	NOT NULL	Ngày thực hiện (Unix Timestamp).
description	TEXT	Nullable	Ghi chú/Mô tả thêm.
receipt_path	TEXT	Nullable	Đường dẫn ảnh hóa đơn.
type	INTEGER	Default 0	0: Chi tiêu (Expense), 1: Thu nhập (Income).
is_recurring	INTEGER	Default 0	1: Là giao dịch lặp lại, 0: Giao dịch thường.
recurrence_period	TEXT	Nullable	Chu kỳ lặp ('daily', 'weekly', 'monthly').
next_occurrence	INTEGER	Nullable	Ngày dự kiến tạo giao dịch tiếp theo.
recurring_group_id	INTEGER	Default 0	ID nhóm để liên kết các giao dịch lặp lại với nhau.

4. Table: budgets

Column Name	Data Type	Constraints	Description
-------------	-----------	-------------	-------------

id	INTEGER	PK, Auto Increment	Định danh ngân sách.
user_id	INTEGER	FK (users.id)	Người tạo ngân sách.
category_id	INTEGER	FK (categories.id)	Danh mục áp dụng (0 = Tổng ngân sách).
amount	REAL	NOT NULL	Hạn mức tiền tối đa.
period_start	INTEGER	NOT NULL	Ngày bắt đầu tính.
period_end	INTEGER	NOT NULL	Ngày kết thúc.
created_at	INTEGER	NOT NULL	Ngày tạo bản ghi.

5. Table: expense_templates

Column Name	Data Type	Constraints	Description
id	INTEGER	PK, Auto Increment	Định danh mẫu.
name	TEXT	NOT NULL	Tên hiển thị mẫu (vd: 'Tiền trợ').
category_id	INTEGER	FK (categories.id)	Danh mục mặc định của mẫu.
default_amount	REAL	Default 0	Số tiền gợi ý sẵn.
icon_emoji	TEXT	Nullable	Emoji biểu tượng (vd: 🏠).

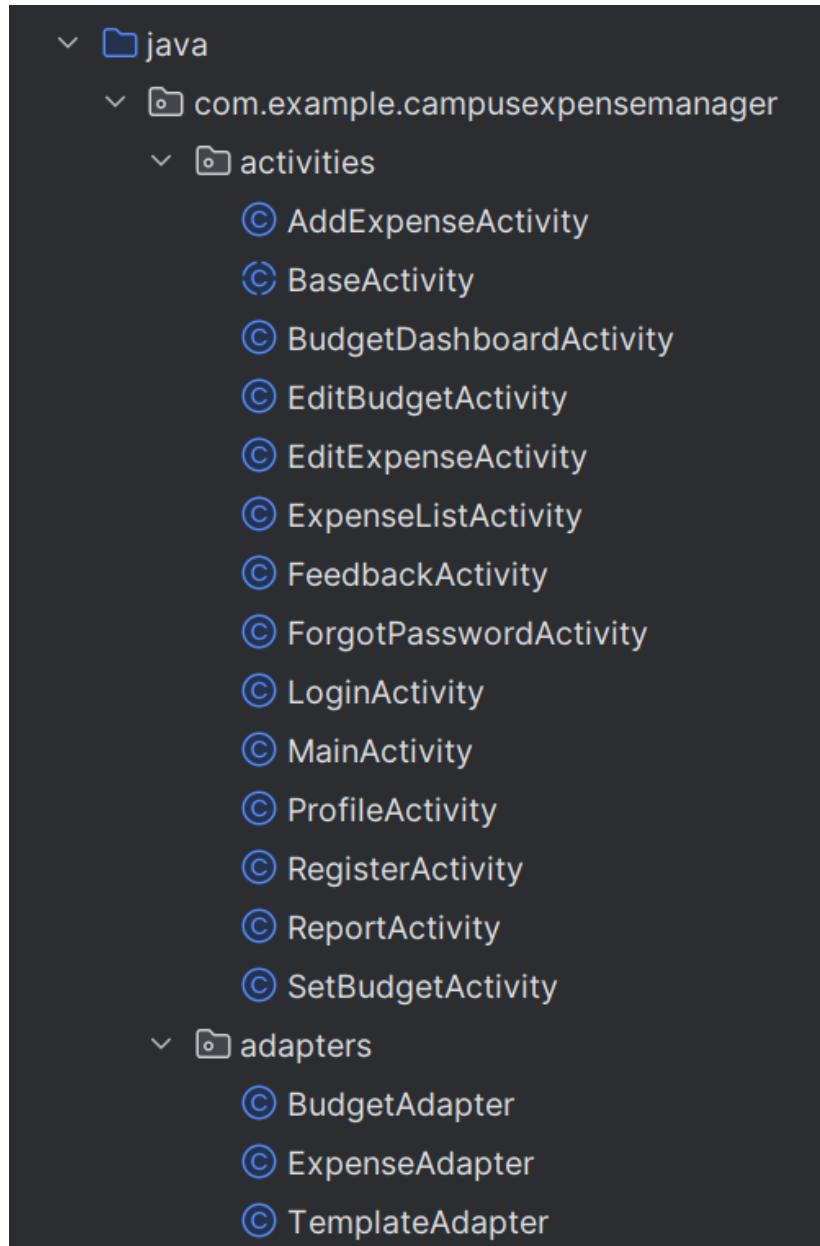
6. Table: feedback

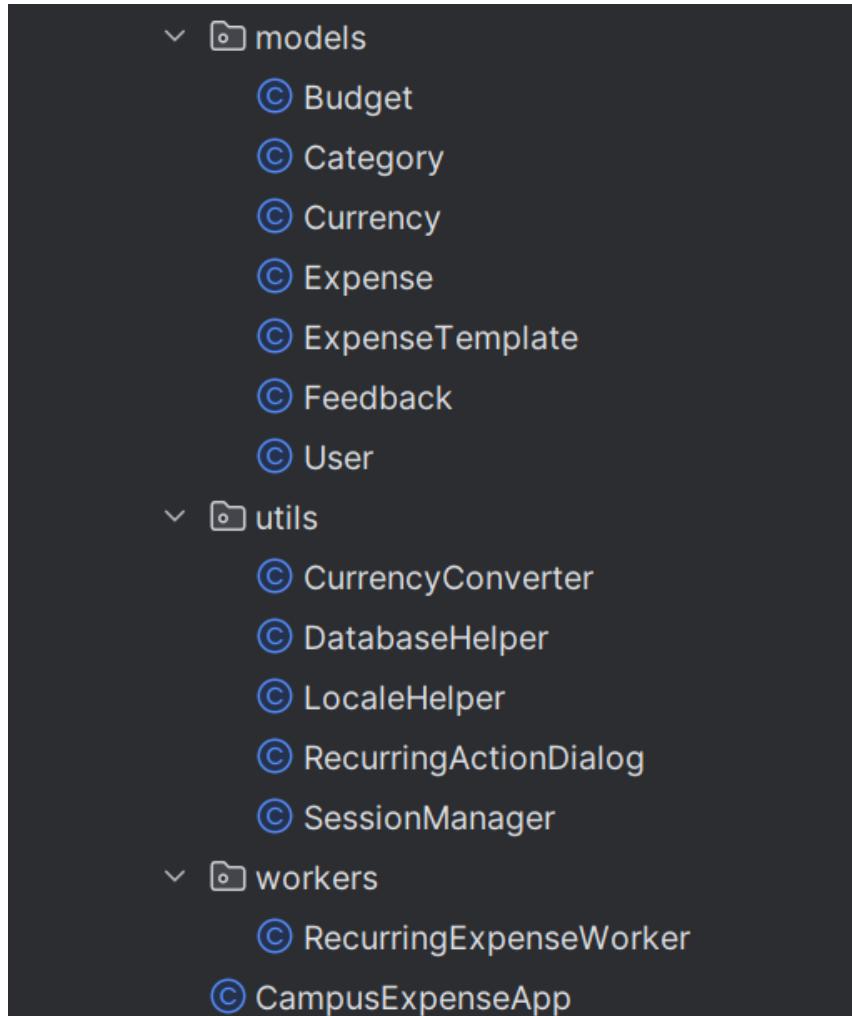
Column Name	Data Type	Constraints	Description
id	INTEGER	PK, Auto Increment	Định danh phản hồi.
user_id	INTEGER	FK (users.id)	Người gửi.
rating	INTEGER	NOT NULL	Số sao đánh giá (1-5).
content	TEXT	Nullable	Nội dung chi tiết.
timestamp	INTEGER	NOT NULL	Thời gian gửi.

3. SYSTEM ARCHITECTURE & DESIGN

3.1. Project Package Structure

Cấu trúc dự án được tổ chức theo mô hình **Package-by-Layer**, giúp phân tách rõ ràng giữa Giao diện (UI), Dữ liệu (Data) và Logic xử lý. Dưới đây là cây thư mục mã nguồn chính của ứng dụng:





3.2. Component Responsibilities

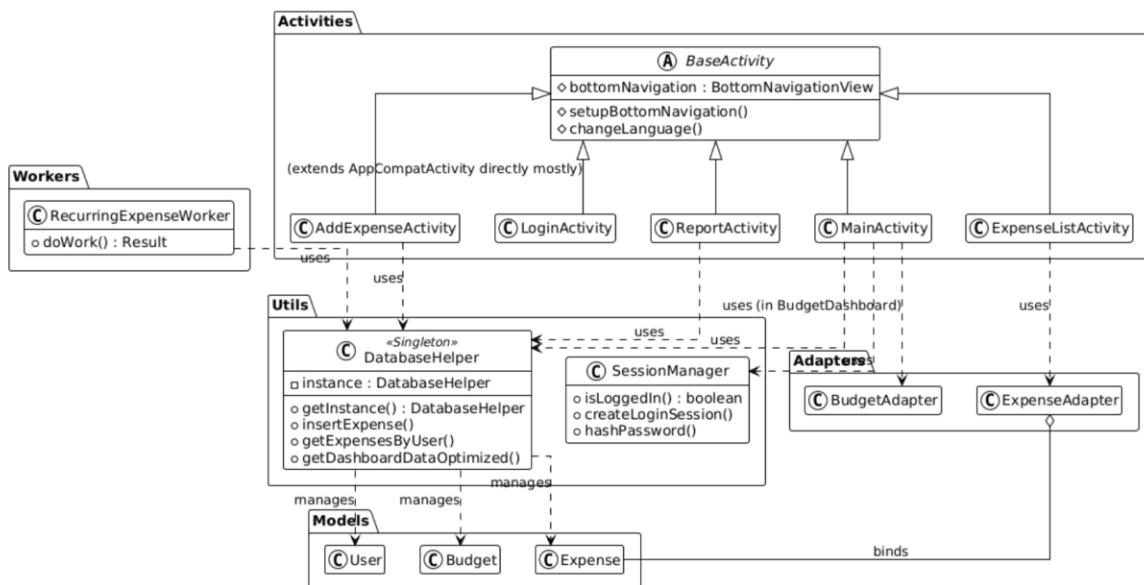
Mô tả chi tiết chức năng của từng gói (Package):

- **activities:** Chứa các lớp kế thừa từ AppCompatActivity. Đặc biệt, BaseActivity đóng vai trò lớp cha (Parent Class) xử lý các logic chung như **Navigation Bottom Bar**, **Đa ngôn ngữ (Locale)** và **Dark Mode** cho tất cả màn hình con.
- **models:** Định nghĩa các đối tượng dữ liệu (Entities). Các lớp này chỉ chứa thuộc tính (attributes), constructor và getter/setter, phản ánh trực tiếp cấu trúc bảng trong SQLite.
- **utils:** Chứa các logic nghiệp vụ cốt lõi:
 - DatabaseHelper: Quản lý vòng đời CSDL và thực thi mọi câu lệnh SQL.

- SessionManager: Quản lý phiên đăng nhập, mã hóa mật khẩu (SHA-256) và lưu cài đặt người dùng.
- workers:** Sử dụng Android WorkManager để thực hiện các tác vụ chạy ngầm (Background Tasks) ngay cả khi ứng dụng đã tắt, đảm bảo các giao dịch định kỳ (Recurring Expenses) luôn được tạo đúng hạn.

3.3. Class Diagram (Sơ đồ Lớp)

Sơ đồ dưới đây minh họa mối quan hệ giữa các thành phần chính trong hệ thống, bao gồm quan hệ kế thừa (Inheritance) và phụ thuộc (Dependency).



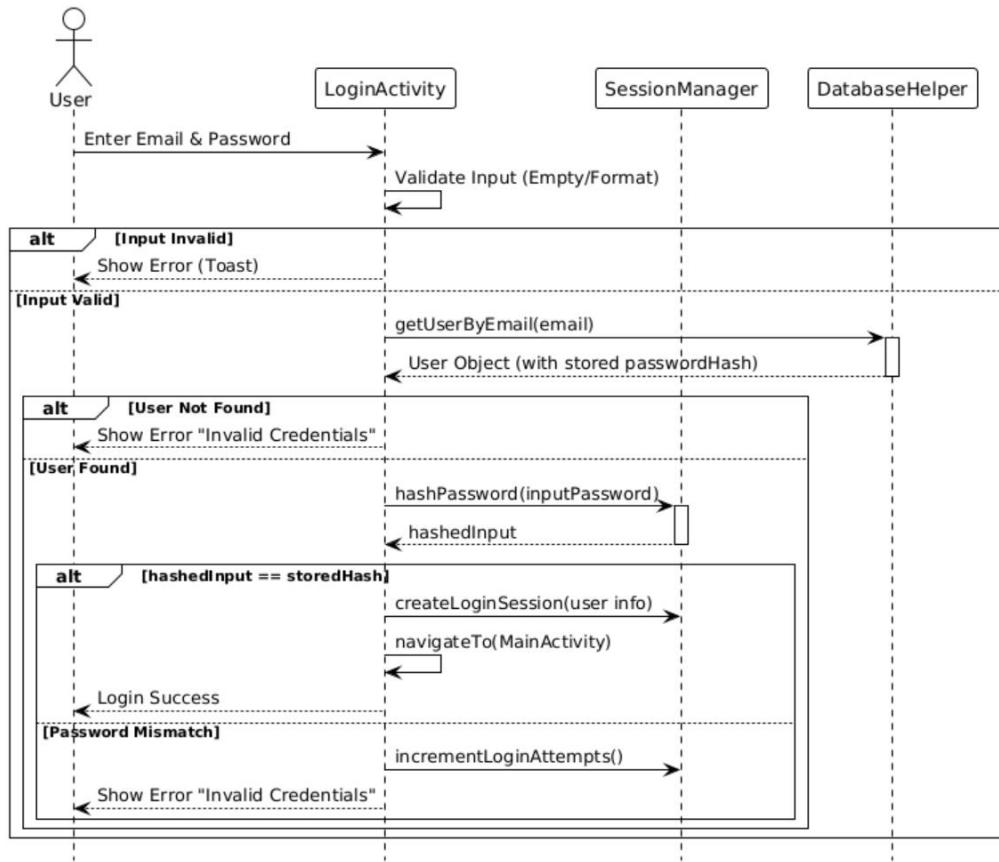
4. SYSTEM LOGIC & DATA FLOWS

Chương này mô tả chi tiết luồng xử lý nội bộ (Internal Data Flow) của các chức năng nghiệp vụ quan trọng nhất trong ứng dụng. Các biểu đồ dưới đây minh họa cách Activities, DatabaseHelper và Models tương tác với nhau.

4.1. User Authentication Flow (Secure Login)

Quy trình đăng nhập đảm bảo bảo mật bằng cách không bao giờ so sánh mật khẩu thô (plain text). Hệ thống băm (hash) mật khẩu người dùng nhập vào và so sánh với chuỗi hash đã lưu trong SQLite.

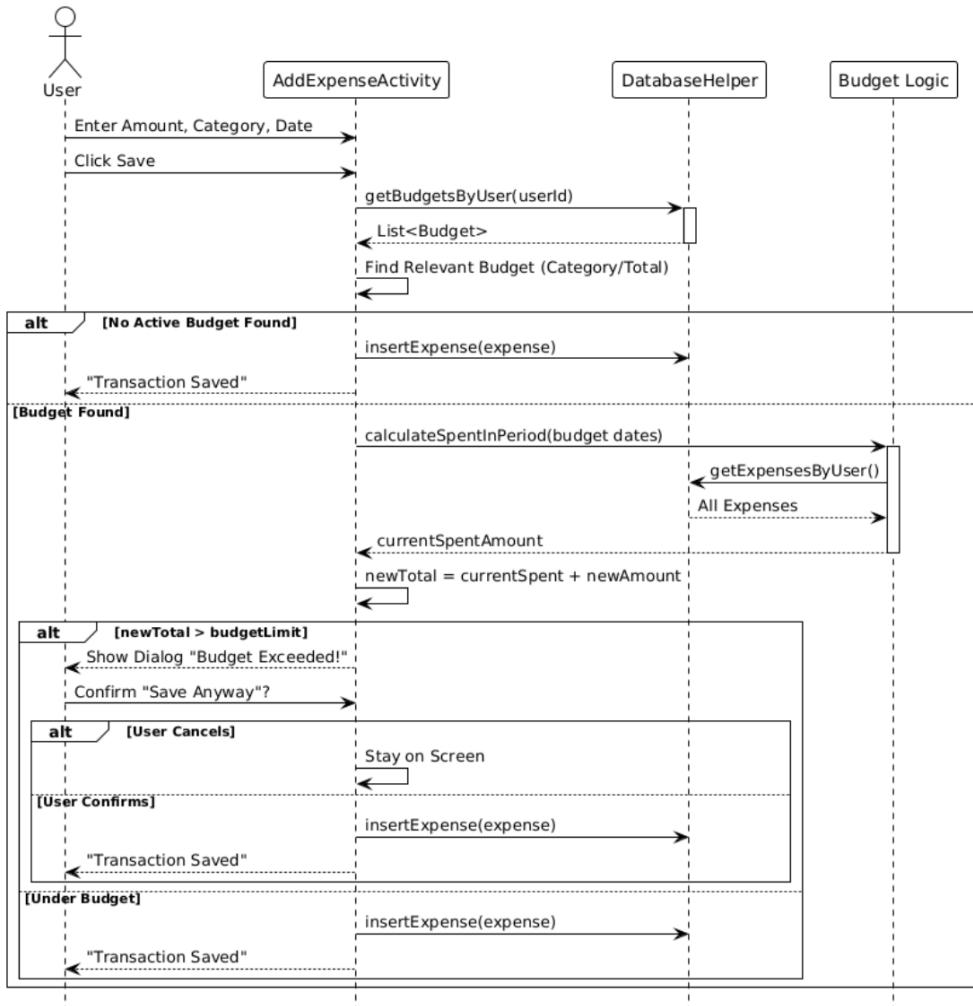
Sequence Diagram: User Login Process



4.2. Expense Creation Flow (with Budget Check)

Đây là luồng nghiệp vụ phức tạp nhất. Trước khi lưu một giao dịch chi tiêu (Expense), hệ thống phải tính toán xem giao dịch này có làm vượt hạn mức ngân sách (Budget) hay không để đưa ra cảnh báo kịp thời.

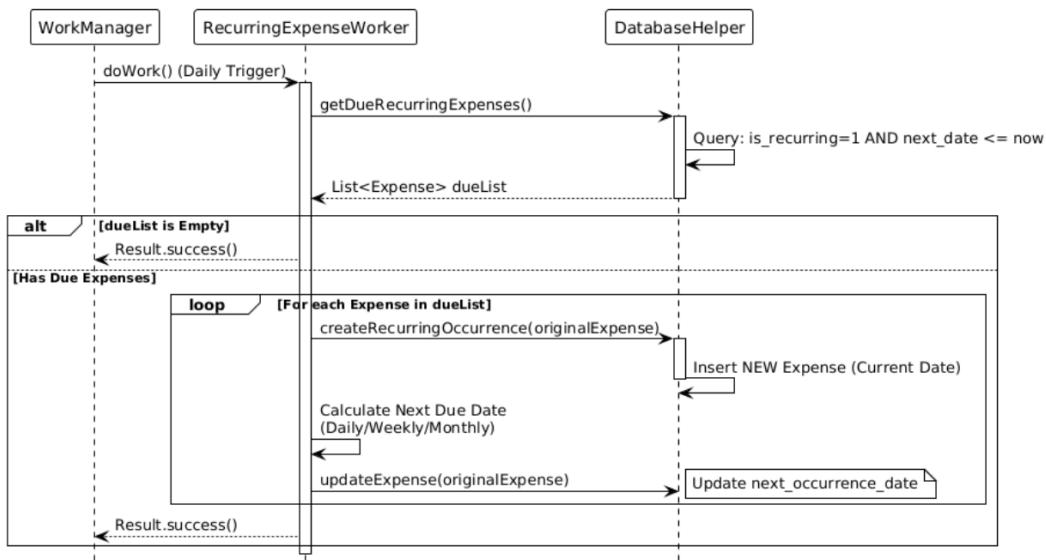
Sequence Diagram: Add Expense with Budget Validation



4.3. Recurring Expense Automation Flow

Tính năng tự động hóa này sử dụng Android WorkManager. Worker chạy ngầm mỗi ngày, quét cơ sở dữ liệu để tìm các giao dịch định kỳ đã đến hạn và tự động tạo bản ghi mới mà không cần người dùng mở ứng dụng.

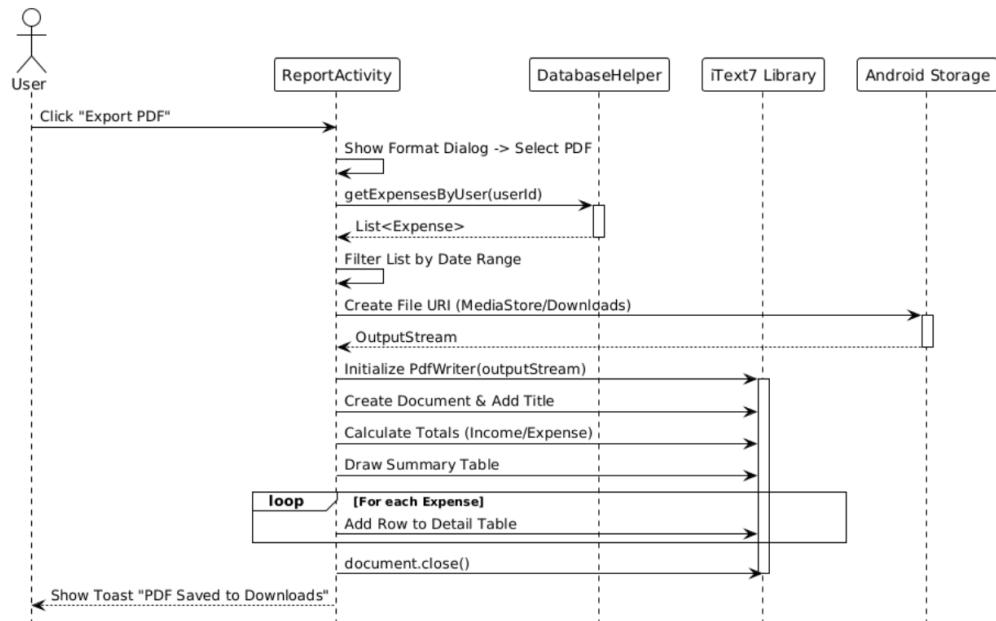
Sequence Diagram: Recurring Expense Worker



4.4. PDF Report Generation Flow

Tính năng xuất báo cáo sử dụng thư viện iText7. Luồng xử lý bao gồm việc truy vấn dữ liệu, định dạng bảng biểu và ghi file vào bộ nhớ thiết bị theo chuẩn Scoped Storage của Android 10+.

Sequence Diagram: PDF Report Export



5. THIRD-PARTY LIBRARIES & IMPLEMENTATION

Để đảm bảo hiệu năng và tính ổn định, dự án sử dụng các thư viện mã nguồn mở tiêu chuẩn công nghiệp. Dưới đây là chi tiết kỹ thuật về cách thức tích hợp và sử dụng chúng.

5.1. MPAndroidChart (Visual Analytics)

- **Mục đích:** Trực quan hóa dữ liệu tài chính giúp người dùng dễ dàng nắm bắt xu hướng chi tiêu.
- **Implementation:**
 - **PieChart:** Được sử dụng trong ReportActivity để hiển thị tỷ lệ phần trăm chi tiêu theo danh mục. Dữ liệu được tổng hợp từ SQLite (getExpensesByUser), gom nhóm theo categoryId, sau đó đưa vào PieDataSet. Các lát cắt (slices) được tô màu động dựa trên bảng màu Sera UI .
 - **LineChart:** Hiển thị xu hướng chi tiêu trong 6 tháng gần nhất. Trục X sử dụng IndexAxisValueFormatter để hiển thị tên tháng (Jan, Feb...), trục Y hiển thị số tiền.

5.2. iText 7 (Professional Reporting)

- **Mục đích:** Tạo báo cáo PDF chất lượng cao, có thể in ấn hoặc gửi qua email.
- **Implementation:**
 - Sử dụng class PdfWriter và PdfDocument để khởi tạo file PDF.
 - **Table Layout:** Dữ liệu được trình bày dưới dạng bảng (com.itextpdf.layout.element.Table) với độ rộng cột tùy chỉnh (UnitValue.createPercentArray).
 - **Scoped Storage:** Logic lưu file tuân thủ nghiêm ngặt bảo mật Android 10+. Trên Android Q+, sử dụng MediaStore và ContentValues để ghi luồng dữ liệu (OutputStream) vào thư mục Downloads mà không cần quyền WRITE_EXTERNAL_STORAGE .

5.3. AndroidX WorkManager (Background Automation)

- **Mục đích:** Xử lý tính năng "Giao dịch định kỳ" (Recurring Expenses) một cách tin cậy ngay cả khi ứng dụng bị tắt hoặc thiết bị khởi động lại.
- **Implementation:**

- **Worker Class:** RecurringExpenseWorker kế thừa từ Worker. Hàm doWork() chứa logic quét CSDL tìm các giao dịch đến hạn (next_occurrence_date <= current_time)
- **Scheduling:** Trong MainActivity, một PeriodicWorkRequest được cấu hình để chạy mỗi 24 giờ (1, TimeUnit.DAYS) với chính sách ExistingPeriodicWorkPolicy.KEEP để tránh trùng lặp tác vụ.

5.4. AndroidX Biometric (Security)

- **Mục đích:** Cung cấp phương thức đăng nhập không mật khẩu (Passwordless) an toàn.
- **Implementation:**
 - Sử dụng BiometricPrompt để hiển thị dialog xác thực chuẩn của hệ thống.
 - Logic kiểm tra phần cứng (BiometricManager.canAuthenticate) được thực hiện trước khi kích hoạt tính năng.
 - Key bảo mật: Email của người dùng được lưu trong SharedPreferences (KEY_BIOMETRIC_EMAIL) chỉ khi xác thực vân tay thành công lần đầu.

6. CONCLUSION & FUTURE WORK

6.1. Technical Achievements

Dự án **CampusExpense Manager** đã hoàn thành mục tiêu xây dựng một ứng dụng quản lý tài chính cá nhân mạnh mẽ, tuân thủ các tiêu chuẩn kỹ thuật hiện đại:

1. **Architecture:** Kiến trúc phân lớp (Layered) rõ ràng, dễ bảo trì.
2. **Offline-first:** Hoạt động hoàn hảo không cần Internet nhờ tối ưu hóa SQLite.
3. **Security:** Áp dụng mã hóa mật khẩu SHA-256 và bảo mật sinh trắc học.
4. **Performance:** Sử dụng Raw SQL Queries cho các tác vụ tổng hợp dữ liệu lớn, đảm bảo phản hồi UI < 50ms.

6.2. Known Limitations

- **Sync:** Chưa hỗ trợ đồng bộ đám mây (Cloud Sync) giữa nhiều thiết bị.
- **Automation:** Chưa có tính năng quét hóa đơn tự động (OCR) bằng AI.

6.3. Future Roadmap (v2.0)

Dựa trên nền tảng kỹ thuật vững chắc hiện tại, các tính năng sau được đề xuất cho phiên bản tiếp theo:

- Tích hợp **Firebase Realtime Database** để sao lưu dữ liệu.
- Sử dụng **Google ML Kit** để trích xuất thông tin từ ảnh chụp hóa đơn.
- Nâng cấp lên kiến trúc **MVVM** với **Jetpack Compose** để hiện đại hóa UI Codebase.