

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  
KHOA: CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN  
HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**CHỦ ĐỀ 1 : ỨNG DỤNG QUẢN LÝ**

**Đề tài 20: Xây dựng ứng dụng quản lý tài chính trên nền tảng Android sử dụng  
Android studio**

Sinh viên thực hiện	Lớp	Khóa
Hà Tiến Dũng	DCCNTT12.10.12	K12
Nguyễn Văn Đạt	DCCNTT12.10.12	K12
Vũ Thanh Hải	DCCNTT12.10.12	K12

**Bắc Ninh, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  
KHOA: CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN  
HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**Nhóm: 5**

**Chủ đề 1 : Ứng dụng quản lý**

**Đề tài 20: Xây dựng ứng dụng quản lý tài chính trên nền tảng Android sử dụng  
Android studio**

STT	Sinh viên thực hiện	Mã sinh viên	Điểm bằng số	Điểm bằng chữ
1	Hà Tiến Dũng	20213409		
2	Nguyễn Văn Đạt	20213571		
3	Vũ Thanh Hải	20213345		

**CÁN BỘ CHẤM 1**  
(Ký và ghi rõ họ tên)

**CÁN BỘ CHẤM 2**  
(Ký và ghi rõ họ tên)

## MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT .....	1
DANH MỤC HÌNH ẢNH.....	2
DANH MỤC BẢNG BIỂU .....	4
CHƯƠNG 1: MỞ ĐẦU .....	5
1.1 Giới thiệu về đề tài.....	5
1.2 Giới hạn phạm vi nghiên cứu đề tài .....	5
1.3 Phương pháp nghiên cứu.....	6
1.4 Xác định yêu cầu phải thực hiện .....	6
1.5 Các công nghệ sử dụng .....	6
1.5.1 Ngôn ngữ lập trình Java.....	6
1.5.2 Ứng dụng lập trình Android Studio .....	9
1.5.3 Hệ quản trị cơ sở dữ liệu SQLite .....	11
1.5.4 Ứng dụng đồ họa DB Browser for SQLite .....	12
CHƯƠNG 2: QUY TRÌNH NGHIÊN CỨU THỰC TRẠNG .....	15
2.1 Quá trình khảo sát thực trạng .....	15
2.1.1 Quy trình khảo sát thực trạng .....	15
2.1.2 Kết quả khảo sát thực trạng .....	16
2.2 Đặc tả bài toán.....	17
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	18
3.1 Các chức năng của hệ thống .....	18
3.1.1 Tài khoản .....	18
3.1.2 Quản lý khoản thu.....	18
3.1.3 Quản lý khoản chi.....	18
3.1.4 Thống kê .....	18
3.1.5 Biểu đồ phân rã chức năng .....	19
3.2 Biểu đồ luồng dữ liệu.....	19
3.2.1 Biểu đồ luồng dữ liệu mức ngữ cảnh.....	20
3.2.2 Biểu đồ luồng dữ liệu mức 1 .....	20
3.3 Biểu đồ Usecase .....	22

3.3.1 Danh sách các actor của hệ thống.....	22
3.3.2 Danh sách các actor của hệ thống.....	22
3.3.3 Vẽ biểu đồ Usecase.....	23
3.3.4 Đặc tả Usecase.....	26
3.4 Biểu đồ lớp.....	39
3.4.1 Biểu đồ lớp tổng quát.....	39
3.4.2 Đặc tả biểu đồ lớp.....	40
3.3 Biểu đồ mối quan hệ thực thể .....	42
3.5 Các bảng trong cơ sở dữ liệu .....	42
CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH.....	44
4.1 Cấu trúc mã nguồn chương trình.....	44
4.2 Các thiết kế giao diện chương trình .....	46
4.2.1 Giao diện đăng nhập.....	46
4.2.2 Giao diện đăng ký.....	47
4.2.3 Giao diện quên mật khẩu .....	48
4.2.4 Giao diện đổi mật khẩu.....	49
4.2.5 Giao diện thêm khoản thu, khoản chi .....	50
4.2.6 Giao diện form chọn ngày .....	51
4.2.7 Giao diện trang chính.....	52
4.2.8 Giao diện trang chủ.....	53
4.2.9 Giao diện ngân sách.....	54
4.2.10 Giao diện khoản chi.....	55
4.2.11 Giao diện thống kê.....	56
4.2.12 Giao diện item recycle view khoản chi.....	57
4.2.13 Giao diện item recycle view khoản thu .....	57
4.2.14 Giao diện sửa, xóa của khoản thu, khoản chi .....	58
CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH.....	60
KẾT LUẬN .....	72
TÀI LIỆU THAM KHẢO .....	73

## DANH MỤC CÁC TỪ VIẾT TẮT

STT	Chữ viết tắt	Giải thích
1	SDK	Software Development Kit
2	JVM	Java Virtual Machine
3	OOP	Object-Oriented Program
4	CRM	Customer Relationship Management
5	IDE	Integrated Development Environment
6	CPU	Central Processing Unit
7	API	Application Programming Interface
8	SQL	Structured Query Language

## DANH MỤC HÌNH ẢNH

Hình 3.1: Biểu đồ phân rã chức năng .....	19
Hình 3.2: Biểu đồ luồng dữ liệu mức ngữ cảnh .....	20
Hình 3.3: Biểu đồ luồng dữ liệu mức 1 tài khoản .....	20
Hình 3.4: Biểu đồ luồng dữ liệu mức 1 quản lý khoản thu .....	21
Hình 3.5: Biểu đồ luồng dữ liệu mức 1 quản lý khoản chi.....	21
Hình 3.6: Biểu đồ luồng dữ liệu mức 1 thống kê .....	22
Hình 3.7: Biểu đồ usecase tổng quát .....	23
Hình 3.8: Biểu đồ usecase các chức năng tài khoản.....	24
Hình 3.9: Biểu đồ usecase quản lý khoản thu .....	24
Hình 3.10: Biểu đồ usecase quản lý khoản thu .....	25
Hình 3.11: Biểu đồ usecase thống kê .....	25
Hình 3.12: Biểu đồ lớp .....	39
Hình 3.13: Biểu đồ mối quan hệ thực thể.....	42
Hình 4.1: Cấu trúc mã nguồn chương trình.....	44
Hình 4.2: Giao diện đăng nhập.....	46
Hình 4.3: Giao diện đăng ký.....	47
Hình 4.4: Giao diện quên mật khẩu.....	48
Hình 4.5: Giao diện đổi mật khẩu .....	49
Hình 4.6: Giao diện thêm khoản thu, khoản chi.....	50
Hình 4.7: Giao diện form chọn ngày .....	51
Hình 4.8: Giao diện trang chính .....	52
Hình 4.9: Giao diện trang chủ .....	53
Hình 4.10: Giao diện ngân sách .....	54
Hình 4.11: Giao diện khoản chi.....	55
Hình 4.12: Giao diện thống kê .....	56
Hình 4.13: Giao diện recycle view khoản chi .....	57
Hình 4.14: Giao diện recycle view khoản thu .....	57
Hình 4.15: Giao diện sửa, xóa của khoản thu, khoản chi .....	58
Hình 5.1: Giao diện người dùng đăng nhập ứng dụng .....	60
Hình 5.2: Giao diện người dùng đăng ký tài khoản .....	61
Hình 5.3: Giao diện người dùng quên mật khẩu .....	62
Hình 5.4: Giao diện người dùng đổi mật khẩu .....	63

Hình 5.5: Giao diện chính các chức năng của người dùng.....	64
Hình 5.6: Giao diện khi thêm khoản chi mới. ....	65
Hình 5.7: Giao diện khi thêm khoản thu mới. ....	66
Hình 5.8: Giao diện chức năng ngân sách. ....	67
Hình 5.9: Giao diện chức năng sửa xóa khoản thu.....	68
Hình 5.10: Giao diện chức năng chi tiêu.....	69
Hình 5.11: Giao diện chức năng sửa xóa khoản chi .....	70
Hình 5.12: Giao diện chức năng thống kê. ....	71

## DANH MỤC BẢNG BIỂU

Bảng tbl3.1: bộ ký hiệu sử dụng.....	19
Bảng tbl3.2: danh sách actor hệ thống.....	22
Bảng tbl3.3: danh sách usecase hệ thống .....	22
Bảng tbl3.4 :Mô tả lớp TaiKhoan .....	40
Bảng tbl3.5 :Mô tả lớp.....	40
Bảng tbl3.6 :Mô tả lớp.....	41
Bảng tbl3.7 :Mô tả lớp.....	41
Bảng tbl3.8 :tblTaiKhoan .....	42
Bảng tbl3.9 :tblDanhMucChiTieu .....	43
Bảng tbl3.10 :tblKhoanChiTieu .....	43
Bảng tbl3.11 :tblThuNhap .....	43
Bảng tbl4.1 :Bảng thành phần chính của giao diện đăng nhập .....	46
Bảng tbl4.2 :Bảng thành phần chính của giao diện đăng ký .....	47
Bảng tbl4.3 :Bảng thành phần chính của giao diện quên mật khẩu .....	48
Bảng tbl4.4 :Bảng thành phần chính của giao diện đổi mật khẩu .....	49
Bảng tbl4.5 :Bảng thành phần chính của giao diện thêm khoản thu, khoản chi .....	50
Bảng tbl4.6 :Bảng thành phần chính của giao diện form chọn ngày.....	51
Bảng tbl4.7 :Bảng thành phần chính của giao diện trang chính.....	52
Bảng tbl4.8 :Bảng thành phần chính của giao diện trang chủ .....	53
Bảng tbl4.9 :Bảng thành phần chính của giao diện ngân sách .....	54
Bảng tbl4.10 :Bảng thành phần chính của giao diện khoản chi .....	55
Bảng tbl4.11 :Bảng thành phần chính của giao diện thống kê .....	56
Bảng tbl4.12 :Bảng thành phần chính của giao diện .....	57
Bảng tbl4.13 :Bảng thành phần chính của giao diện .....	57
Bảng tbl4.14 :Bảng thành phần chính của giao diện thêm khoản thu, khoản chi .....	58



# CHƯƠNG 1: MỞ ĐẦU

## 1.1 Giới thiệu về đề tài

Ngày nay, thiết bị di động sử dụng hệ điều hành android đã là một phần quá quen thuộc với tất cả những người sử dụng điện thoại thông minh. Là một hệ điều hành mã nguồn mở được phát triển bởi Google, dành cho các thiết bị di động và máy tính bảng. Được ra mắt vào đầu năm 2008, android đã nhanh chóng trở thành hệ điều hành di động phổ biến nhất trên thế giới nhờ tính linh hoạt, khả năng tùy biến và cộng đồng phát triển ứng dụng rộng lớn. Hệ điều hành này hỗ trợ nhiều ngôn ngữ lập trình, phổ biến nhất là java, kotlin và C++, đồng thời cung cấp bộ công cụ phát triển phần mềm (SDK) mạnh mẽ, giúp các lập trình viên dễ xây dựng và triển khai ứng dụng.

Ngoài ra, Android cũng cung cấp quyền truy cập Google Play Store – một kho ứng dụng khổng lồ, nơi người dùng có thể tải và cài đặt hàng triệu ứng dụng với các chức năng đa dạng, từ giải trí, giáo dục, công việc đến quản lý,...

Với cương vị là một sinh viên đang được học môn phát triển ứng dụng cho thiết bị di động. Nhóm em đã có cơ hội được áp dụng những kiến thức môn học được về lập trình android và các công cụ phát triển cho ứng dụng di động và nhóm chúng em quyết định chọn đề tài ứng dụng quản lý và xây dựng ứng dụng sổ thu chi: quản lý tài chính trên nền tảng Android sử dụng Android studio, ứng dụng giúp người dùng quản lý tình hình thu nhập và chi tiêu của mình trong ngày, tháng, năm. Với giao diện trực quan và tính năng dễ sử dụng phân chia các khoản thu qua các danh mục như ăn uống, đi lại, giải trí, sinh hoạt,... Ngoài ra ứng dụng còn có biểu đồ tròn, biểu đồ cột giúp người dùng hiểu rõ hơn về cách phân bổ tài chính của mình, giúp người dùng có thể điều chỉnh các khoản chi phí một cách hợp lý hơn trong tương lai.

Ứng dụng của nhóm em mang tính học tập và mong muốn hiểu biết hơn về phát triển ứng dụng trên nền tảng Android nên có thể vẫn còn nhiều thiếu sót mong các thầy cô đọc và góp ý.

## 1.2 Giới hạn phạm vi nghiên cứu đề tài

- Đề tài sẽ giới hạn phạm vi nghiên cứu là các hoạt động liên quan đến tác vụ quản lý chi tiêu như:

- + Quản lý các chi tiêu trong ngày, tháng, năm, thêm các chi tiêu
- + Quản lý các nguồn thu trong ngày, tháng, năm, thêm nguồn thu là các khoản thu nhập mà người dùng thu được.

+ Thống kê các khoản thu và khoản chi trong thời gian mong muốn bằng biểu đồ.

### **1.3 Phương pháp nghiên cứu**

- Phương pháp thu thập thông tin: Thu thập thông tin về lý luận và tài thực tiễn quản lý tài chính, thu chi từ các nguồn tài liệu khác nhau, bao gồm sách, báo, website,...

- Phương pháp phân tích, tổng hợp: phân tích, tổng hợp các thông tin thu nhập được để xây dựng ứng dụng quản lý tài chính, quản lý thu chi.

- Phương pháp thử nghiệm: Thử nghiệm hệ thống quản lý tài chính, quản lý thu chi hiện đại trong thực tế để đánh giá hiệu quả của hệ thống.

### **1.4 Xác định yêu cầu phải thực hiện**

\* Mục tiêu đặt ra:

- Xây dựng ứng dụng Sổ thu chi: quản lý tài chính bao gồm các chức năng: Quản lý khoản chi, Quản lý khoản thu, thống kê.

\* Kết quả:

- Tăng cường khả năng quản lý:

+ Quản lý nguồn thu: Giúp người dùng lưu trữ và quản lý các nguồn thu là các thu nhập của người dùng trong hiện tại và tương lai, bao gồm số tiền, nguồn thu, ngày và ghi chú nguồn thu.

+ Quản lý khoản chi: Giúp người dùng lưu trữ và quản lý các khoản chi tiêu của người dùng bao gồm số tiền, danh mục chi tiêu, ngày và ghi chú khoản chi.

- Tăng cường khả năng thống kê:

+ Thống kê khoản thu và khoản chi tiêu sử dụng các biểu đồ giúp dễ dàng quan sát tổng quan.

+ Sử dụng biểu đồ cột so sánh giữa tình hình thu và chi trong từng khoảng thời gian để có sự so sánh giúp người dùng xác định mình chi tiêu đã hợp lý chưa.

+ Sử dụng biểu đồ tròn so sánh số phần trăm chi tiêu vào các danh mục để người dùng nắm bắt mình đã chi tiêu vào mục gì nhiều nhất trong khoảng thời gian nào.

### **1.5 Các công nghệ sử dụng**

#### **1.5.1 Ngôn ngữ lập trình Java**

- Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

- Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

- Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

- Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

- Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch.

+ Thông dịch (Interpreter): Nó dịch từng lệnh rồi chạy từng lệnh, lần sau muốn chạy lại thì phải dịch lại.

+ Biên dịch (Compiler): Code sau khi được biên dịch sẽ tạo ra 1 file thường là .exe, và file .exe này có thể đem sử dụng lại không cần biên dịch nữa.

- Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể như sau:

+ Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã byte code. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành machine code (hay native code) khi nhận được yêu cầu chạy chương trình.

- Ưu điểm: Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.

- Nhược điểm: Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được).

- Đặc điểm:

+ Ngôn ngữ hoàn toàn hướng đối tượng (Pure OOP).

+ Ngôn ngữ đa nền: "Viết một lần, Chạy trên nhiều nền".

+ Ngôn ngữ đa luồng (multi-threading): xử lý và tính toán song song.

+ Ngôn ngữ phân tán (distributed): cho phép các đối tượng của một ứng dụng được phân bố và thực thi trên các máy tính khác nhau.

+ Ngôn ngữ động: cho phép mã lệnh của một chương trình được tải từ một máy tính về máy của người yêu cầu thực thi chương trình.

+ Ngôn ngữ an toàn: hạn chế các thao tác nguy hiểm cho máy tính thật.

+ Ngôn ngữ đơn giản, dễ học, kiến trúc chương trình đơn giản, trong sáng.

- Các phiên bản của Java:

+ Java Standard Edition (Java SE) – Là một nền tảng cơ bản cho phép phát triển giao diện điều khiển, các ứng dụng mạng và các ứng dụng dạng Win Form.

+ Java Enterprise Edition (Java EE) – Được xây dựng trên nền tảng Java SE, giúp phát triển các ứng dụng web, các ứng dụng ở cấp doanh nghiệp, ...

+ Java Mobile Edition (Java ME) – Là một nền tảng cho phép phát triển các ứng dụng nhúng vào các thiết bị điện tử như mobile, ...

\* Các thành phần của Java SE Platform:

- Gồm 2 thành phần:

+ JRE (Java Runtime Environment): cung cấp JVM (Java Virtual Machine) và thư viện được sử dụng để chạy chương trình Java.

+ JDK (Java Development Kit): được biết đến như bộ cung cụ phát triển Java, bao gồm: trình biên dịch và trình gỡ rối được sử dụng để phát triển các ứng dụng Java.

+ Java ngôn ngữ thuần đối tượng (pure object): => Tất cả đều được định nghĩa trong các lớp (class)

+ Trong một ứng dụng có một lớp thực thi được.

- Lớp thực thi được:

+ Có tên lớp trùng với tên tập tin chứa nó.

+ Phải khai báo public.

+ Có chứa phương thức được thực thi đầu tiên:

*public static void main (String argv []) {...}*

+ Nếu nhiều lớp được định nghĩa trong một tập tin: => chỉ có một lớp được khai báo public.

\* Ứng dụng của Java:

- Java được sử dụng cho nhiều loại ứng dụng khác nhau, bao gồm:

+ Ứng dụng web: Java là một ngôn ngữ lập trình phổ biến cho phát triển ứng dụng web. Nó được sử dụng để tạo ra các trang web động, ứng dụng web dựa trên mô hình-xem-điều khiển và các ứng dụng web dựa trên dịch vụ.

+ Ứng dụng di động: Java là một lựa chọn phổ biến cho phát triển ứng dụng di động. Nó được sử dụng để tạo ra các ứng dụng Android và IOS.

+ Ứng dụng doanh nghiệp: Java được sử dụng để tạo ra các ứng dụng doanh nghiệp, chẳng hạn như hệ thống quản lý quan hệ khách hàng (CRM), hệ thống quản lý nguồn lực doanh nghiệp (ERP) và hệ thống phân tích dữ liệu.

+ Ứng dụng hệ thống: Java được sử dụng để tạo ra các ứng dụng hệ thống, chẳng hạn như trình duyệt web, hệ điều hành và máy chủ web.

- Có nhiều môi trường phát triển Java (IDE) khác nhau có sẵn, bao gồm:

+ Eclipse: Eclipse là một IDE phổ biến cho phát triển Java. Nó cung cấp một loạt các tính năng, bao gồm trình biên dịch, trình gỡ lỗi, trình chỉnh sửa mã và công cụ xây dựng.

+ IntelliJ IDEA: IntelliJ IDEA là một IDE khác phổ biến cho phát triển Java. Nó cung cấp một loạt các tính năng nâng cao, bao gồm hỗ trợ cho các công nghệ Java mới nhất và khả năng tự động hoàn thành mã.

+ NetBeans: NetBeans là một IDE nguồn mở cho phát triển Java. Nó cung cấp một loạt các tính năng, bao gồm trình biên dịch, trình gỡ lỗi, trình chỉnh sửa mã và công cụ xây dựng.

### **1.5.2 Ứng dụng lập trình Android Studio**

- Android Studio là môi trường phát triển tích hợp (IDE) chính thức do Google phát triển dành riêng cho việc phát triển ứng dụng trên nền tảng Android. Được xây dựng dựa trên IntelliJ IDEA, một trong những IDE hàng đầu trong lĩnh vực lập trình, Android Studio cung cấp một bộ công cụ mạnh mẽ và toàn diện, hỗ trợ lập trình viên trong việc thiết kế, phát triển, gỡ lỗi và triển khai ứng dụng Android. Từ khi ra mắt lần đầu vào năm 2013, Android Studio đã nhanh chóng trở thành lựa chọn hàng đầu cho các nhà phát triển ứng dụng di động.

- Tính năng nổi bật của Android Studio:

+ Tích hợp Android SDK: Android Studio được tích hợp với Android Software Development Kit (SDK), cho phép lập trình viên dễ dàng truy cập và sử dụng các API và thư viện cần thiết để phát triển ứng dụng Android. Điều này giúp tiết kiệm thời gian và công sức khi phải cài đặt và cấu hình từng thành phần một cách thủ công.

+ Hỗ trợ đa ngôn ngữ lập trình: Android Studio hỗ trợ hai ngôn ngữ lập trình chính là Java và Kotlin. Kotlin, được Google khuyến nghị cho phát triển ứng dụng Android, mang

đến cú pháp rõ ràng và tối ưu, giúp lập trình viên dễ dàng viết mã hơn và giảm thiểu lỗi. Điều này giúp cho quá trình phát triển trở nên nhanh chóng và hiệu quả.

- + Thiết kế giao diện trực quan: Android Studio cung cấp Android Layout Editor, một công cụ thiết kế giao diện người dùng mạnh mẽ cho phép lập trình viên xây dựng và tùy chỉnh giao diện bằng cách kéo thả các thành phần (widgets) vào layout. Tính năng “ConstraintLayout” cho phép thiết kế giao diện tương thích với nhiều kích thước màn hình khác nhau, giúp tối ưu hóa trải nghiệm người dùng.

- + Hệ thống xây dựng Gradle: Gradle là hệ thống quản lý xây dựng được tích hợp trong Android Studio, cho phép lập trình viên dễ dàng quản lý phụ thuộc (dependencies) và tự động hóa quy trình xây dựng ứng dụng. Nhờ đó, việc biên dịch mã nguồn và đóng gói ứng dụng trở nên nhanh chóng và hiệu quả.

- + Trình mô phỏng thiết bị: Android Emulator là một công cụ quan trọng trong Android Studio, cho phép lập trình viên kiểm thử ứng dụng trên nhiều cấu hình thiết bị khác nhau mà không cần phải sử dụng thiết bị thực tế. Trình mô phỏng này hỗ trợ nhiều phiên bản Android và cấu hình phần cứng, giúp đảm bảo ứng dụng hoạt động ổn định trên mọi thiết bị.

- + Gỡ lỗi và kiểm tra hiệu năng: Android Studio cung cấp các công cụ gỡ lỗi mạnh mẽ cho phép lập trình viên theo dõi và sửa lỗi trong mã nguồn của mình. Hệ thống phân tích hiệu năng giúp phát hiện các vấn đề về hiệu suất, như mức sử dụng CPU, bộ nhớ, và thời gian xử lý, từ đó tối ưu hóa ứng dụng cho trải nghiệm người dùng tốt nhất.

- + Tính năng hỗ trợ phát triển: Android Studio có tính năng hỗ trợ phát triển mạnh mẽ, bao gồm tự động hoàn thành mã (code completion), kiểm tra lỗi ngay lập tức (instant error checking), và các gợi ý mã (code suggestions). Những tính năng này giúp lập trình viên tiết kiệm thời gian và tăng hiệu quả công việc.

- Ưu điểm của Android Studio:

- + Cộng đồng và tài liệu hỗ trợ phong phú: Android Studio được phát triển và duy trì bởi Google, nên được hỗ trợ bởi một cộng đồng lập trình viên lớn. Có rất nhiều tài liệu hướng dẫn, diễn đàn và nguồn tài nguyên sẵn có giúp lập trình viên giải quyết vấn đề và nâng cao kỹ năng.

- + Cập nhật thường xuyên: Google liên tục cập nhật và cải tiến Android Studio với các tính năng mới, cải thiện hiệu suất và hỗ trợ các công nghệ mới nhất trong phát triển ứng dụng Android.

+ Khả năng mở rộng và tích hợp: Android Studio cho phép lập trình viên tích hợp nhiều thư viện bên thứ ba và các công cụ phát triển khác, giúp mở rộng chức năng của ứng dụng. Điều này cũng cho phép lập trình viên dễ dàng tích hợp các API của Google như Firebase, Google Maps, và nhiều dịch vụ khác.

- Ứng dụng trong phát triển ứng dụng di động: lập trình viên có thể phát triển nhiều loại ứng dụng khác nhau, từ ứng dụng di động đơn giản cho đến các ứng dụng phức tạp sử dụng máy học (machine learning), thực tế ảo (augmented reality), và nhiều công nghệ tiên tiến khác. Điều này mở ra nhiều cơ hội cho lập trình viên sáng tạo và thực hiện ý tưởng của mình.

### **1.5.3 Hệ quản trị cơ sở dữ liệu SQLite**

- SQLite là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nhẹ, độc lập, và hoàn toàn được nhúng, được thiết kế để sử dụng trong các ứng dụng di động và máy tính để bàn. Kể từ khi ra mắt vào năm 2000, SQLite đã nhanh chóng trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới, nhờ vào tính đơn giản, hiệu suất cao và khả năng dễ dàng tích hợp vào các ứng dụng.

- Đặc điểm nổi bật của SQLite:

+ Nhẹ và độc lập: SQLite là một thư viện C nhỏ gọn, không cần một máy chủ riêng biệt để chạy. Điều này có nghĩa là nó có thể được nhúng trực tiếp vào ứng dụng mà không cần cài đặt thêm, giúp giảm thiểu yêu cầu về tài nguyên hệ thống.

+ Cơ sở dữ liệu quan hệ: SQLite hỗ trợ các tính năng của cơ sở dữ liệu quan hệ, bao gồm bảng, chỉ mục, khóa chính và khóa ngoại. Điều này cho phép người dùng lưu trữ và truy xuất dữ liệu một cách có tổ chức và hiệu quả.

+ Hoàn toàn miễn phí: SQLite được phát hành dưới dạng phần mềm nguồn mở và hoàn toàn miễn phí để sử dụng, giúp giảm chi phí cho việc phát triển ứng dụng.

+ Khả năng xử lý đồng thời: SQLite hỗ trợ nhiều truy vấn đồng thời, cho phép nhiều ứng dụng hoặc người dùng truy cập dữ liệu cùng một lúc mà không gặp phải xung đột. Điều này rất quan trọng trong các ứng dụng di động, nơi nhiều tác vụ có thể được thực hiện song song.

+ Tính đơn giản và dễ sử dụng: SQLite có cú pháp SQL đơn giản và dễ hiểu, giúp lập trình viên dễ dàng học hỏi và sử dụng để thực hiện các truy vấn và thao tác trên cơ sở dữ liệu. Cách thiết lập và sử dụng SQLite cũng rất đơn giản, không yêu cầu cấu hình phức tạp.

- Cấu trúc và hoạt động của SQLite:

- + Cấu trúc dữ liệu: SQLite lưu trữ dữ liệu trong các tệp tin đơn (single file), với mỗi tệp tin chứa toàn bộ cơ sở dữ liệu. Điều này giúp việc sao lưu, di chuyển hoặc chia sẻ cơ sở dữ liệu trở nên dễ dàng.

- + Ngôn ngữ truy vấn: SQLite sử dụng ngôn ngữ SQL (Structured Query Language) để thực hiện các truy vấn. Các lệnh cơ bản như SELECT, INSERT, UPDATE, và DELETE cho phép người dùng thực hiện các thao tác trên dữ liệu một cách hiệu quả.

- + Chế độ lưu trữ: SQLite hỗ trợ nhiều chế độ lưu trữ, bao gồm chế độ lưu trữ trong bộ nhớ (in-memory) và chế độ lưu trữ trong tệp. Điều này cho phép người dùng lựa chọn cách thức lưu trữ phù hợp với yêu cầu của ứng dụng.

- Ưu điểm của SQLite:

- + Hiệu suất cao: SQLite có hiệu suất cao trong việc thực hiện các truy vấn và thao tác trên dữ liệu. Nó được tối ưu hóa cho các ứng dụng yêu cầu tốc độ cao và thời gian phản hồi nhanh.

- + Dễ dàng tích hợp: SQLite dễ dàng tích hợp vào các ứng dụng Android, iOS, và nhiều nền tảng khác, giúp lập trình viên phát triển ứng dụng mà không cần lo lắng về cấu hình phức tạp.

- + Không yêu cầu cài đặt: Vì SQLite là một thư viện nhúng, nó không yêu cầu cài đặt phần mềm bổ sung, giúp giảm thiểu rắc rối cho người dùng cuối.

- Ứng dụng của SQLite:

- + Ứng dụng di động: SQLite thường được sử dụng trong các ứng dụng Android và iOS để lưu trữ dữ liệu người dùng, thiết lập, và trạng thái ứng dụng. Ví dụ, một ứng dụng quản lý chi tiêu có thể sử dụng SQLite để lưu trữ thông tin giao dịch và ngân sách.

- + Ứng dụng desktop: SQLite cũng được sử dụng trong các ứng dụng máy tính để bàn, như trình soạn thảo văn bản, trình quản lý ảnh, và nhiều ứng dụng khác, để lưu trữ dữ liệu cục bộ.

- + Web Application: Một số ứng dụng web sử dụng SQLite như một giải pháp lưu trữ cơ sở dữ liệu cục bộ, đặc biệt trong các ứng dụng chạy trên trình duyệt.

### **1.5.4 Ứng dụng đồ họa DB Browser for SQLite**

- DB Browser for SQLite là một ứng dụng đồ họa miễn phí và mã nguồn mở, được thiết kế để tạo, chỉnh sửa, và quản lý các cơ sở dữ liệu SQLite một cách dễ dàng và trực



quan. Nó cung cấp một giao diện người dùng thân thiện giúp người dùng, đặc biệt là những người không quen thuộc với dòng lệnh, có thể tương tác với cơ sở dữ liệu mà không gặp khó khăn.

- Tính năng nổi bật của DB Browser for SQLite:

+ Giao diện người dùng trực quan: DB Browser for SQLite có giao diện thân thiện với người dùng, cho phép người dùng dễ dàng điều hướng và thực hiện các thao tác mà không cần biết nhiều về SQL. Giao diện này giúp người dùng mới có thể nhanh chóng làm quen và sử dụng ứng dụng.

+ Tạo và chỉnh sửa cơ sở dữ liệu: Người dùng có thể dễ dàng tạo và chỉnh sửa cơ sở dữ liệu SQLite mới hoặc hiện có. DB Browser cho phép tạo bảng, chỉ mục, và thiết lập các khóa chính và khóa ngoại mà không cần viết mã SQL.

+ Chạy truy vấn SQL: DB Browser cho phép người dùng thực hiện các truy vấn SQL trực tiếp thông qua giao diện. Người dùng có thể viết và chạy các câu lệnh SQL, xem kết quả ngay lập tức và xuất dữ liệu sang nhiều định dạng khác nhau như CSV, JSON, hoặc SQL.

+ Xuất và nhập dữ liệu: DB Browser cung cấp các tùy chọn xuất và nhập dữ liệu từ các tệp khác nhau. Điều này cho phép người dùng dễ dàng chuyển đổi dữ liệu giữa SQLite và các định dạng khác, giúp việc chia sẻ và sao lưu dữ liệu trở nên đơn giản hơn.

+ Quản lý dữ liệu và bảng: Người dùng có thể dễ dàng xem, thêm, sửa đổi và xóa dữ liệu trong các bảng. DB Browser cung cấp chế độ xem bảng rõ ràng, giúp người dùng dễ dàng theo dõi và quản lý dữ liệu.

- Ưu điểm của DB Browser for SQLite:

+ Miễn phí và mã nguồn mở: DB Browser for SQLite là phần mềm miễn phí, và mã nguồn mở, cho phép người dùng tải về và sử dụng mà không phải trả phí. Điều này cũng cho phép người dùng tham gia vào quá trình phát triển phần mềm.

+ Hỗ trợ đa nền tảng: DB Browser có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS, và Linux, giúp người dùng dễ dàng làm việc trên nhiều môi trường khác nhau.

+ Cập nhật thường xuyên: DB Browser for SQLite được cập nhật thường xuyên với các tính năng mới và sửa lỗi, giúp đảm bảo hiệu suất ổn định và bảo mật cho người dùng.

- Ứng dụng của DB Browser for SQLite:

+ Phát triển ứng dụng: Lập trình viên sử dụng DB Browser để kiểm tra và quản lý dữ liệu trong các ứng dụng SQLite, từ các ứng dụng di động đến ứng dụng máy tính để bàn.

+ Phân tích dữ liệu: Các nhà phân tích dữ liệu có thể sử dụng DB Browser để truy vấn và xử lý dữ liệu trong cơ sở dữ liệu SQLite, hỗ trợ cho quá trình phân tích và báo cáo.

+ Giao dịch: DB Browser là công cụ tuyệt vời cho sinh viên và người học muốn tìm hiểu về cơ sở dữ liệu và SQL, nhờ vào giao diện trực quan và dễ sử dụng.

## **CHƯƠNG 2: QUY TRÌNH NGHIÊN CỨU THỰC TRẠNG**

### **2.1 Quá trình khảo sát thực trạng**

#### **2.1.1 Quy trình khảo sát thực trạng**

\* Địa điểm: Văn phòng, trường học, hoặc các tổ chức có liên quan đến quản lý tài chính.

\* Hình thức khảo sát: Phỏng vấn trực tiếp và khảo sát trực tuyến (thông qua bảng hỏi).

\* Đối tượng khảo sát người dùng ứng dụng:

- Câu hỏi khảo sát:

+ Bạn thường sử dụng ứng dụng quản lý thu chi để làm gì?

+ Những vấn đề nào bạn gặp phải khi sử dụng ứng dụng hiện tại?

+ Bạn mong muốn ứng dụng quản lý thu chi có những tính năng gì?

+ Bạn có thể mô tả quy trình quản lý thu chi của bạn hiện tại không?

+ Bạn có gặp khó khăn trong việc theo dõi các khoản thu chi không?

- Dữ liệu thu được:

+ Người dùng cho thấy họ sử dụng ứng dụng để ghi chép, theo dõi và phân tích các khoản thu chi.

+ Một số vấn đề họ gặp phải bao gồm giao diện khó sử dụng, thiếu tính năng báo cáo chi tiết, và khó khăn trong việc xuất dữ liệu.

+ Người dùng mong muốn có các tính năng như tự động hóa việc nhập liệu, báo cáo thống kê tự động, và thông báo khi có khoản thu chi đáng chú ý.

\* Đối tượng khảo sát kế toán viên hoặc người quản lý tài chính:

- Câu hỏi khảo sát:

+ Theo bạn, những vấn đề nào tồn tại trong quá trình quản lý thu chi hiện nay?

+ Bạn cho rằng ứng dụng quản lý thu chi cần được cải thiện về những vấn đề gì?

+ Theo bạn, ứng dụng quản lý tài chính cần đáp ứng những yêu cầu gì để hỗ trợ công việc của bạn?

+ Bạn có gặp khó khăn trong việc lập báo cáo tài chính không?

- Dữ liệu thu được:

+ Kế toán viên cho rằng hiện tại quá trình quản lý thu chi còn nhiều bất cập, như việc nhập liệu thủ công tốn thời gian, khó khăn trong việc tổng hợp dữ liệu từ nhiều nguồn khác nhau.

+ Họ mong muốn ứng dụng có thể tự động tổng hợp và tạo báo cáo tài chính theo thời gian thực, đồng thời cung cấp giao diện dễ sử dụng hơn.

### **2.1.2 Kết quả khảo sát thực trạng**

- Thực trạng quản lý thu chi hiện nay: Theo kết quả khảo sát, các công việc quản lý thu chi hiện tại chủ yếu được thực hiện bằng tay hoặc qua các bảng tính, dẫn đến một số vấn đề như:

+ Tốn nhiều thời gian và công sức cho việc nhập liệu và tổng hợp số liệu.

+ Dễ xảy ra sai sót trong quá trình nhập liệu và báo cáo.

+ Khó khăn trong việc phân tích và theo dõi các khoản thu chi.

- Nhu cầu sử dụng ứng dụng quản lý tài chính, thu chi: Đa số người dùng đều bày tỏ nhu cầu sử dụng ứng dụng quản lý thu chi, vì nó mang lại lợi ích như:

+ Tăng cường hiệu quả quản lý tài chính.

+ Tiết kiệm thời gian và công sức trong việc nhập liệu và báo cáo.

+ Nâng cao độ chính xác và đáng tin cậy của các báo cáo tài chính.

- Nhu cầu về tính năng cần có của ứng dụng:

+ Tìm kiếm và phân loại các khoản thu chi theo ngày, loại, và mục đích.

+ Cung cấp các biểu đồ và báo cáo thống kê để người dùng có thể theo dõi tình hình tài chính một cách trực quan.

+ Cập nhật thông tin tài chính thường xuyên và tự động, giảm thiểu việc nhập liệu thủ công.

- Yêu cầu đối với ứng dụng:

+ Chính xác: thông tin về các khoản thu chi phải được lưu trữ và xử lý một cách chính xác.

+ Đầy đủ: ứng dụng cần lưu trữ đủ thông tin về từng khoản thu chi và các hoạt động tài chính.

+ Linh hoạt: ứng dụng phải có khả năng tùy chỉnh để đáp ứng nhu cầu của từng người dùng.

- + An toàn: thông tin tài chính phải được bảo mật và tránh các truy cập trái phép.

## **2.2 Đặc tả bài toán**

- Một số đặc tả cơ bản:

- + Quản lý thông tin về thu chi: lưu trữ thông tin chi tiết về các khoản thu chi, bao gồm ngày, số tiền, mô tả, loại thu chi, và tài khoản thực hiện.

- + Tìm kiếm và tra cứu: Hệ thống cần cung cấp chức năng tìm kiếm và tra cứu thông tin về các khoản thu chi theo ngày, tháng, năm.

- + Thống kê và báo cáo: Cung cấp chức năng thống kê và báo cáo về tình hình tài chính, số lượng khoản thu chi, và các thông tin khác để hỗ trợ quản lý tài chính hiệu quả.

- Các phương pháp giải quyết bài toán:

- + Phương pháp thủ công: sử dụng các phương pháp thủ công như sổ sách hoặc bảng tính để quản lý các khoản thu chi.

- + Phương pháp tự động hóa: sử dụng các phần mềm quản lý thu chi để tự động hóa các hoạt động quản lý, giúp tiết kiệm thời gian và tăng độ chính xác trong công việc.

## **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

### **3.1 Các chức năng của hệ thống**

#### **3.1.1 Tài khoản**

- Đăng nhập
- Đăng ký
- Quên mật khẩu

#### **3.1.2 Quản lý khoản thu**

- Lưu khoản thu
- Sửa khoản thu
- Xóa khoản thu
- Tìm kiếm theo thời gian

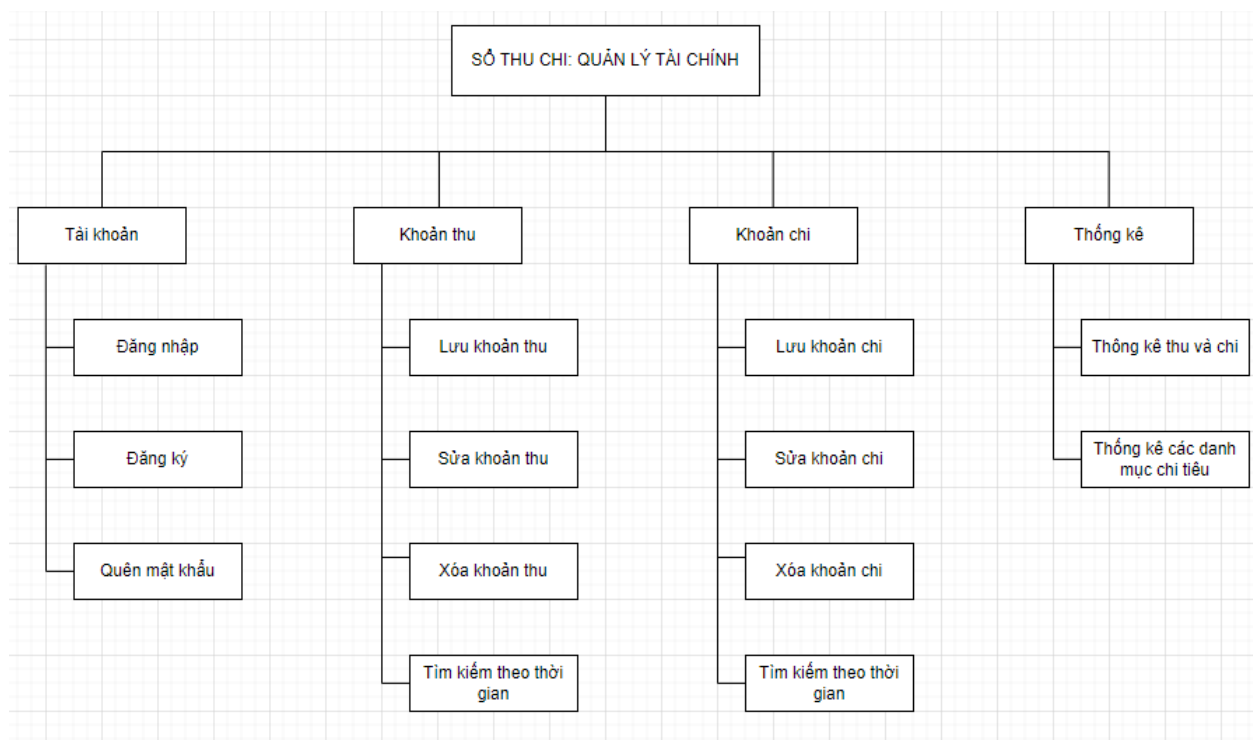
#### **3.1.3 Quản lý khoản chi**

- Lưu khoản chi
- Sửa khoản chi
- Xóa khoản chi
- Tìm kiếm theo thời gian

#### **3.1.4 Thống kê**

- Thống kê khoản thu và chi
- Thống kê danh mục các khoản chi

### 3.1.5 Biểu đồ phân rã chức năng



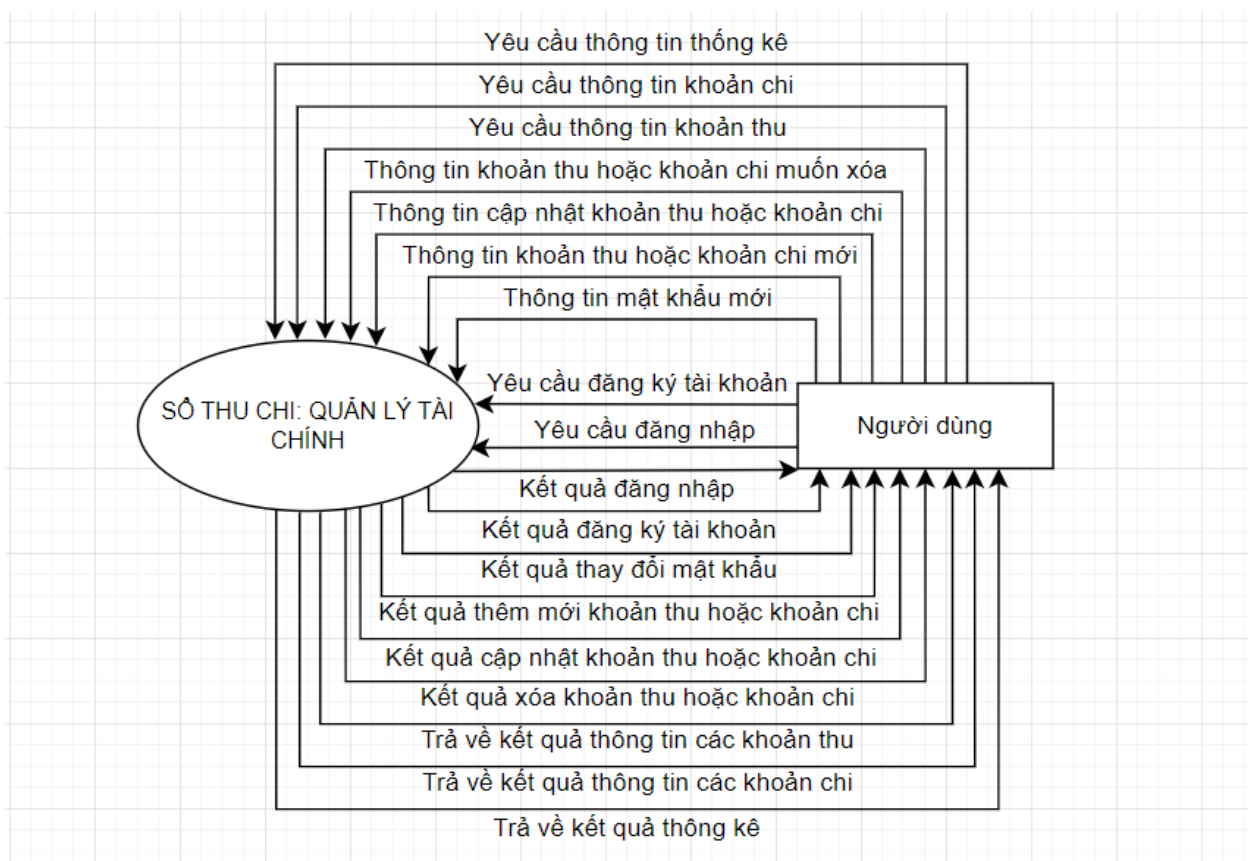
Hình 3.1: Biểu đồ phân rã chức năng

### 3.2 Biểu đồ luồng dữ liệu

Bảng tbl3.1: bộ ký hiệu sử dụng

Khái niệm	Ký pháp của Demarco & Yourdon
Luồng dữ liệu	
Tác nhân	
Tiến trình	
Kho dữ liệu	

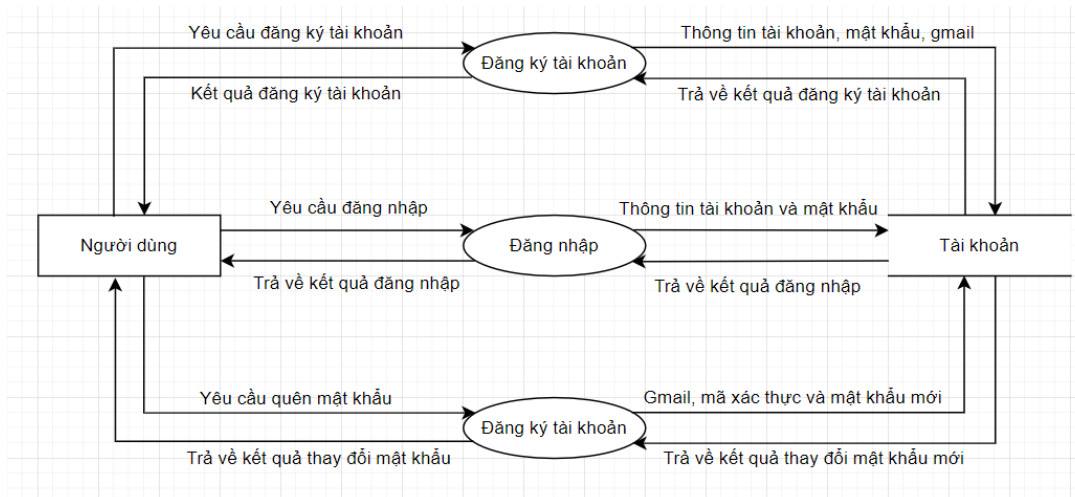
### 3.2.1 Biểu đồ luồng dữ liệu mức ngữ cảnh



Hình 3.2: Biểu đồ luồng dữ liệu mức ngữ cảnh

### 3.2.2 Biểu đồ luồng dữ liệu mức 1

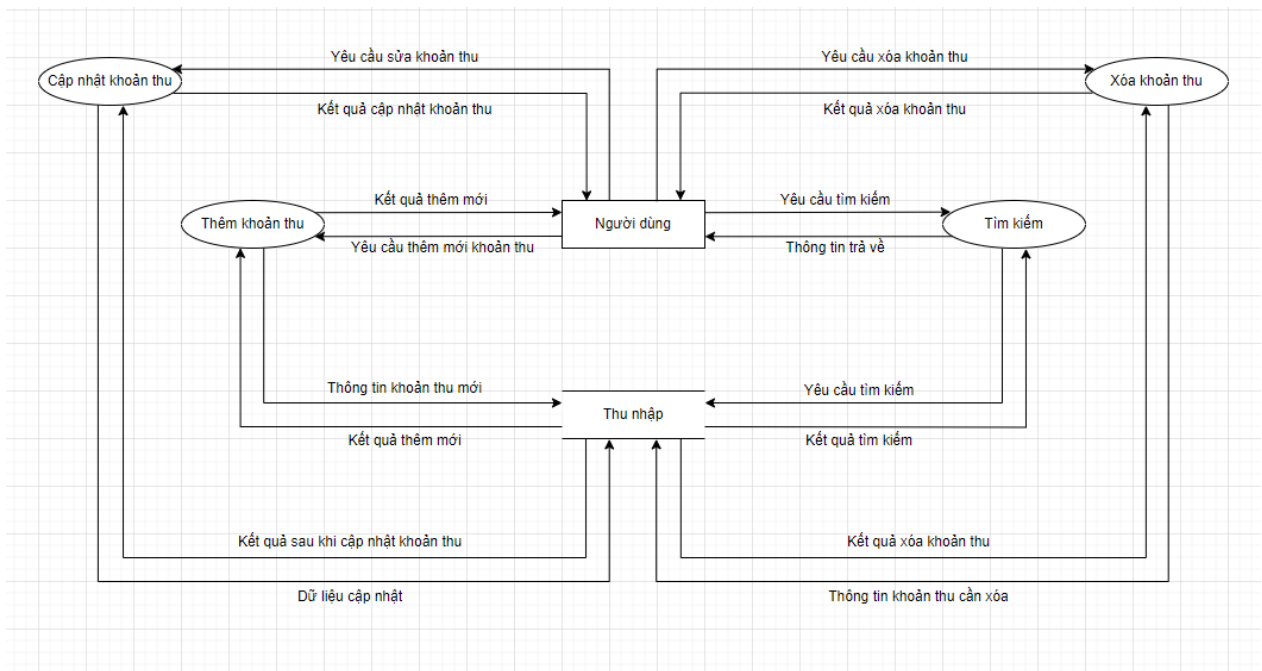
- Mức 1 tiến trình tài khoản



Hình 3.3: Biểu đồ luồng dữ liệu mức 1 tài khoản

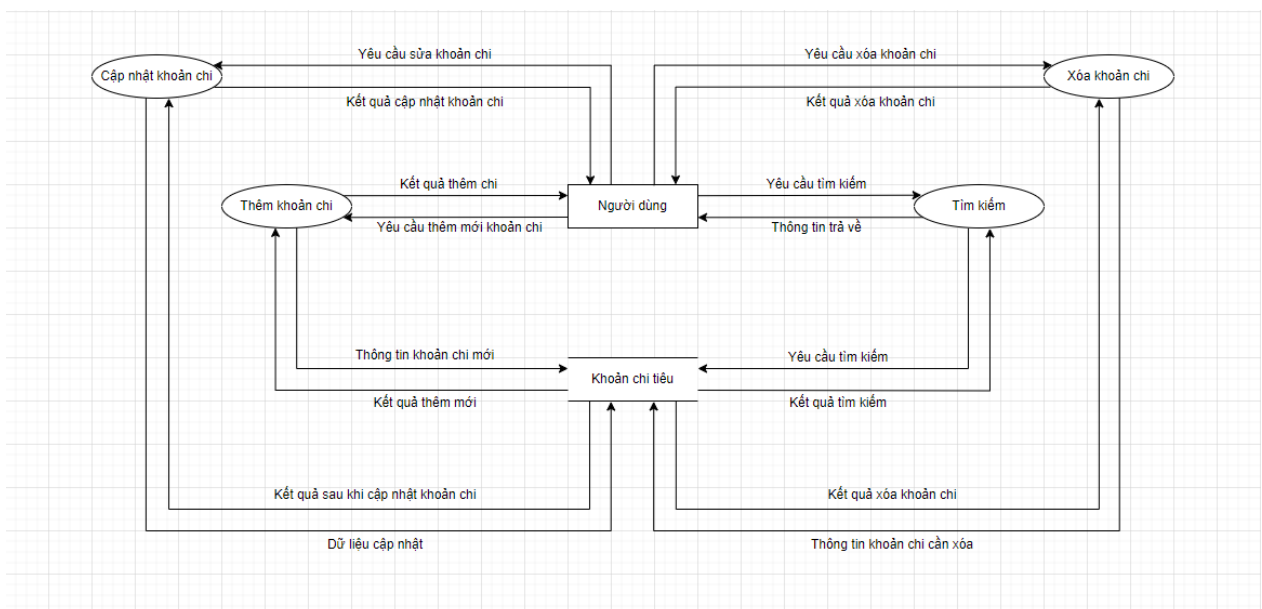


### - Mức 1 tiến trình quản lý khoản thu



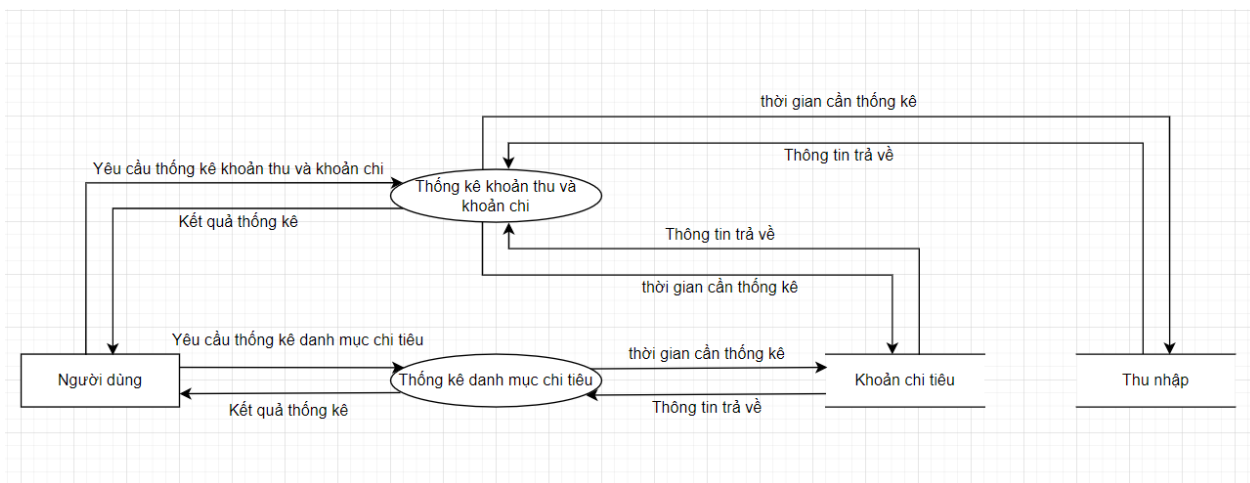
Hình 3.4: Biểu đồ luồng dữ liệu mức 1 quản lý khoản thu

### - Mức 1 tiến trình quản lý khoản chi



Hình 3.5: Biểu đồ luồng dữ liệu mức 1 quản lý khoản chi

- Mức 1 tiến trình thống kê



Hình 3.6: Biểu đồ luồng dữ liệu mức 1 thống kê

### 3.3 Biểu đồ Usecase

### 3.3.1 Danh sách các actor của hệ thống

*Bảng tbl3.2: danh sách actor hệ thống*

STT	Tên actor	Ý nghĩa
1	Người dùng	Có quyền sử dụng các chức năng sẵn có của hệ thống

### 3.3.2 Danh sách các actor của hệ thống

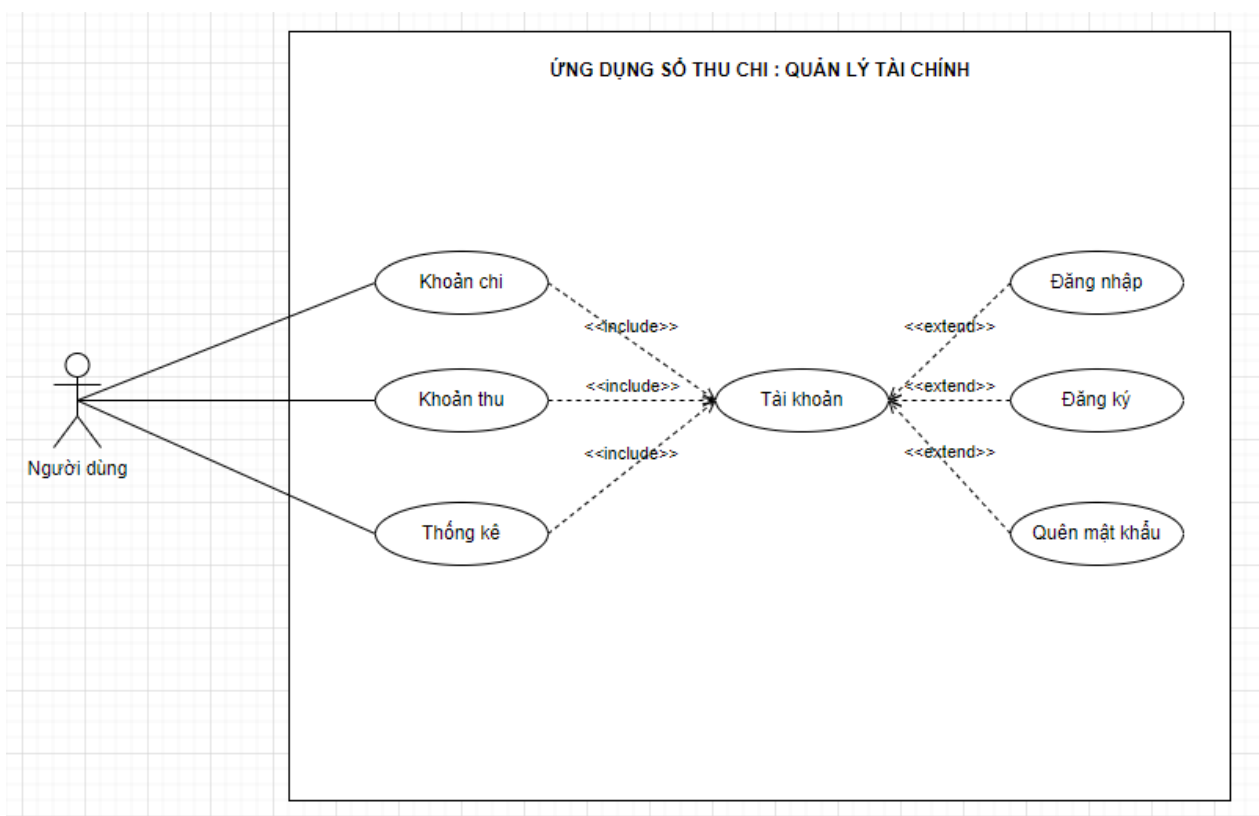
*Bảng tbl3.3: danh sách usecase hệ thống*

STT	Tên usecase	Ý nghĩa
1	Đăng nhập	Cho phép người dùng đăng nhập vào hệ thống
2	Đăng ký	Cho phép người dùng đăng ký tài khoản
3	Quên mật khẩu	Cho phép người dùng đổi mật khẩu mới khi quên mật khẩu
4	Lưu khoản thu	Cho phép người dùng thêm khoản thu mới
5	Sửa khoản thu	Cho phép người dùng sửa khoản thu mong muốn
6	Xóa khoản thu	Cho phép người dùng xóa khoản thu mong muốn
7	Tìm kiếm khoản thu theo thời gian	Cho phép người dùng xem được thông tin các khoản thu theo thời gian mong muốn
8	Lưu khoản chi	Cho phép người dùng thêm khoản chi mới
9	Sửa khoản chi	Cho phép người dùng sửa khoản chi mong muốn

10	Xóa khoản chi	Cho phép người dùng xóa khoản chi mong muốn
11	Tìm kiếm khoản chi theo thời gian	Cho phép người dùng xem được thông tin các khoản chi theo thời gian mong muốn
12	Thống kê thu và chi	Cho phép người dùng xem được tình hình thu nhập và chi tiêu của mình
13	Thống kê các danh mục chi tiêu	Cho phép người dùng xem được thống kê các danh mục đã chi tiêu

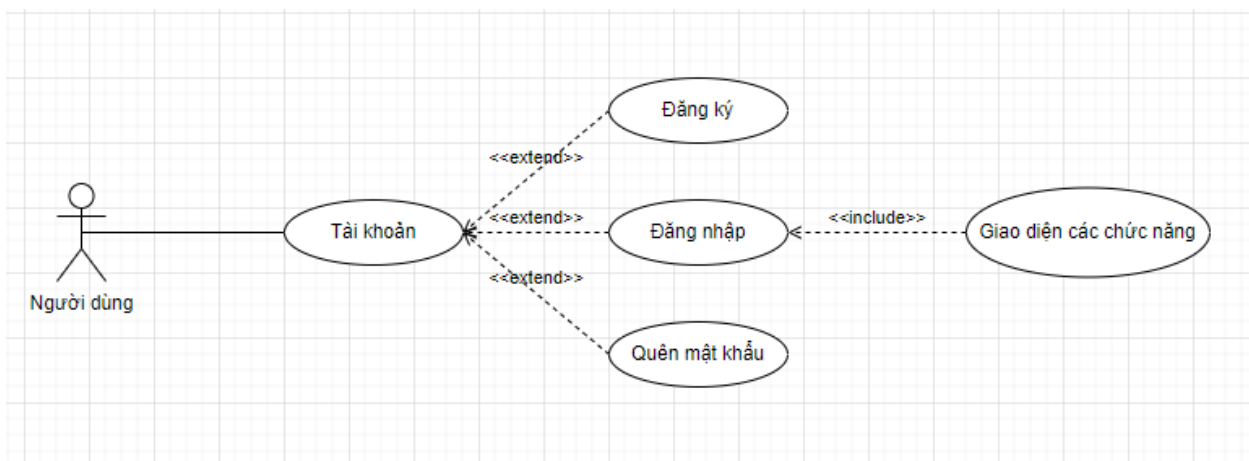
### 3.3.3 Vẽ biểu đồ Usecase

#### 3.3.3.1 Biểu đồ Usecase tổng quát



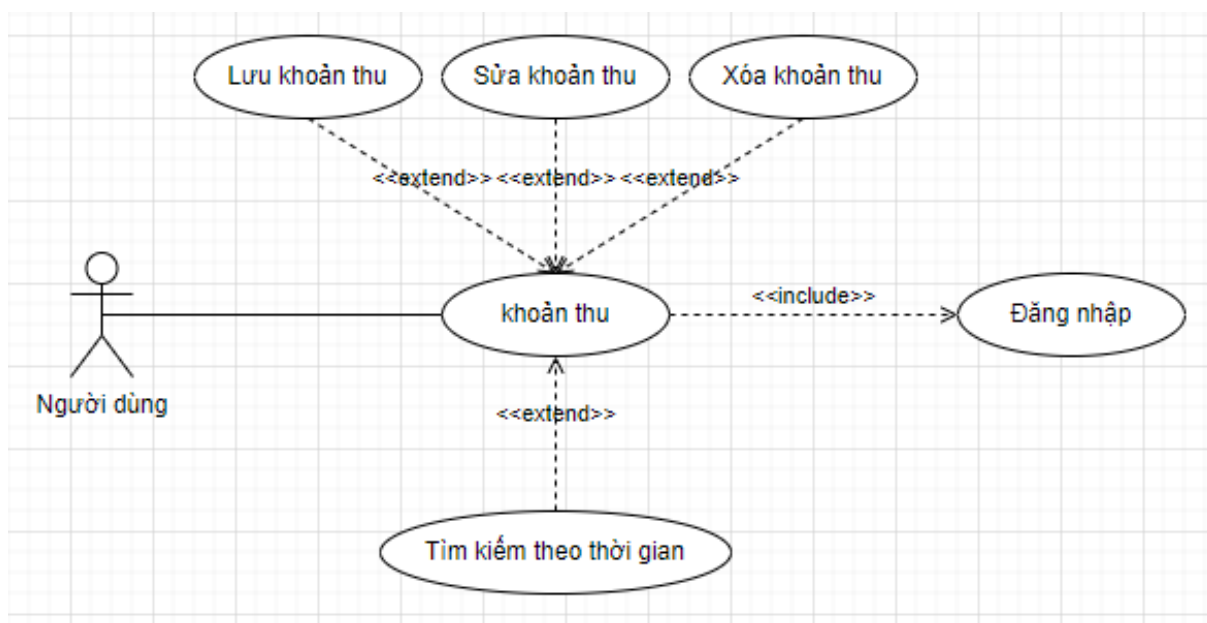
Hình 3.7: Biểu đồ usecase tổng quát

### 3.3.3.2 Biểu đồ Usecase các chức năng tài khoản



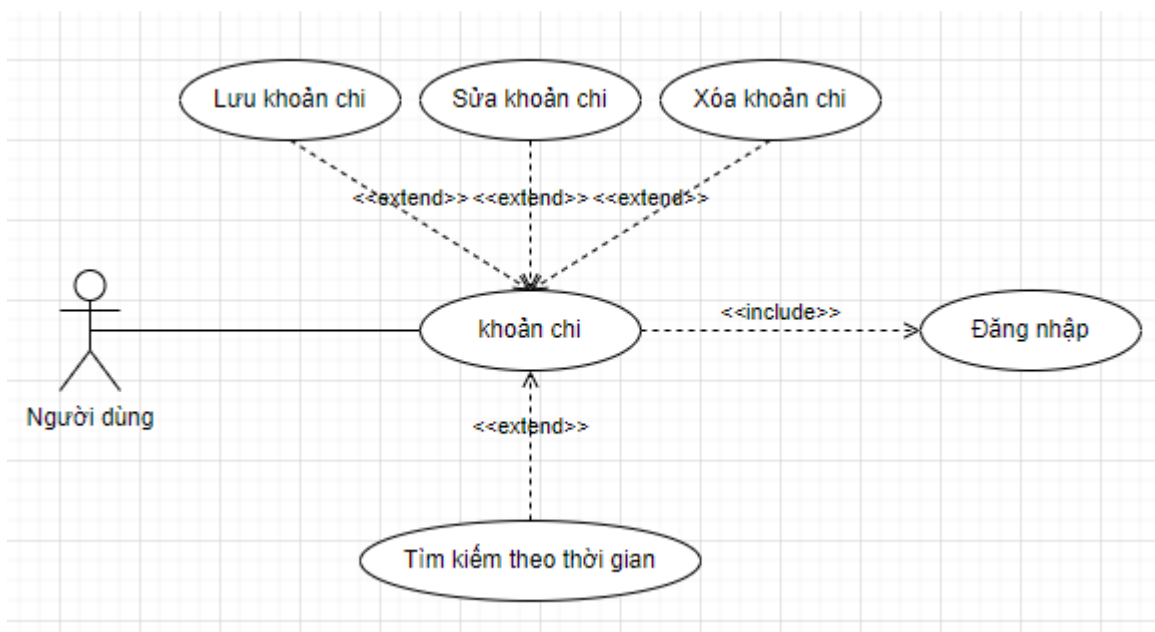
Hình 3.8: Biểu đồ usecase các chức năng tài khoản

### 3.3.3.3 Biểu đồ Usecase quản lý khoản thu



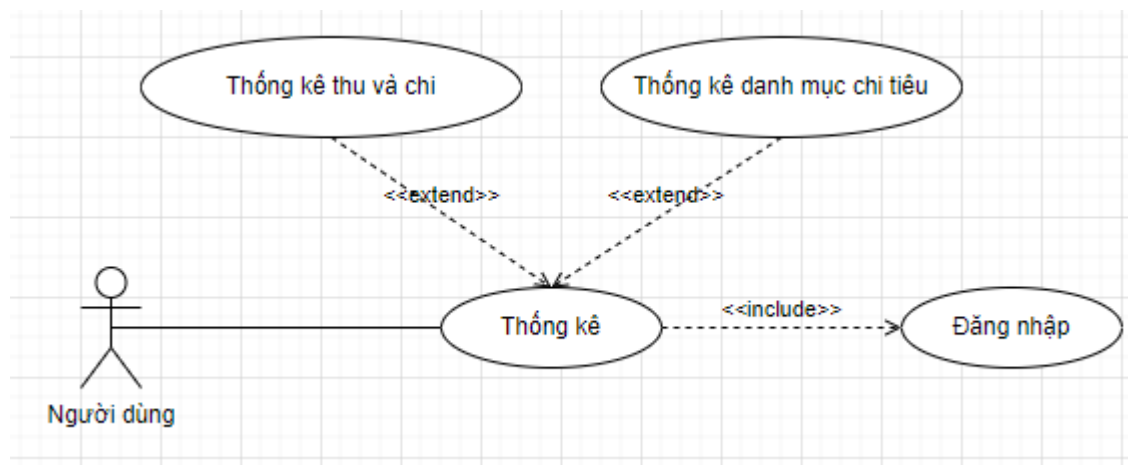
Hình 3.9: Biểu đồ usecase quản lý khoản thu

### 3.3.3.4 Biểu đồ Usecase quản lý khoản chi



Hình 3.10: Biểu đồ usecase quản lý khoản thu

### 3.3.3.5 Biểu đồ Usecase thống kê



Hình 3.11: Biểu đồ usecase thống kê

### 3.3.4 Đặc tả Usecase

#### 3.3.4.1 Đặc tả Usecase tài khoản

##### **\*Đặc tả usecase Đăng nhập tài khoản**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện đăng nhập vào một tài khoản trong hệ thống để sử dụng ứng dụng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng mở ứng dụng.
- Hệ thống hiển thị ra giao diện đăng nhập.
- Người dùng nhập thông tin tài khoản và mật khẩu.
- Hệ thống kiểm tra xem tài khoản và mật khẩu có hợp lệ không, nếu hợp lệ chuyển hướng sang giao diện các chức năng hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:
  - Người dùng hủy đăng nhập tài khoản.
  - Hệ thống thoát ra ngoài.
  - Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin tài khoản hoặc mật khẩu sai.
  - Hệ thống hiển thị thông báo sai tài khoản hoặc mật khẩu.
  - Kết thúc usecase.

- Trạng thái hệ thống trước khi usecase đăng nhập tài khoản được thực hiện: người dùng chỉ cần vào ứng dụng số thu chi: quản lý tài chính trên điện thoại.

- Trạng thái hệ thống sau khi usecase đăng ký tài khoản được thực hiện:

+ Trường hợp thành công: Hệ thống chuyển giao diện đăng nhập sang giao diện các chức năng.

+ Trường hợp không thành công: Hệ thống thông báo sai tài khoản hoặc mật khẩu và không chuyển hướng giao diện.

- Điểm mở rộng: không.

- Giao diện: Giao diện đăng nhập.

**\*Đặc tả usecase Đăng ký tài khoản**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện đăng ký một tài khoản truy cập vào ứng dụng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng chọn chức năng đăng ký tài khoản.
- Hệ thống hiển thị ra giao diện đăng ký tài khoản.
- Người dùng nhập các thông tin cần thiết.
- Người dùng chọn gửi gmail xác nhận.
- Hệ thống gửi mã xác nhận tới gmail người dùng.
- Người dùng nhập mã xác nhận và chọn đăng ký tài khoản.
- Hệ thống kiểm tra mã xác nhận hợp lệ hay không sau nếu hợp lệ thêm thông tin người dùng vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:
  - Người dùng hủy đăng ký tài khoản.
  - Hệ thống trở về giao diện đăng nhập.
  - Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin người dùng đăng ký không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.

- Trạng thái hệ thống trước khi usecase đăng ký tài khoản được thực hiện: người dùng chỉ cần vào ứng dụng số thu chi: quản lý tài chính trên điện thoại.

- Trạng thái hệ thống sau khi usecase đăng ký tài khoản được thực hiện:

+ Trường hợp thành công: Thông tin tài khoản người dùng được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin người dùng vào hệ thống.

- Diêm mở rộng: không.
- Giao diện: Giao diện đăng ký.

### **\*Đặc tả usecase Quên mật khẩu**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc thay đổi lại mật khẩu đăng nhập vào hệ thống khi quên mật khẩu.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng chọn chức năng quên mật khẩu tài khoản.
- Hệ thống hiện thị ra giao diện quên mật khẩu.
- Người dùng nhập địa chỉ gmail đã đăng ký trước đó.
- Hệ thống hiển thị tên tài khoản của gmail đó.
- Người dùng chọn gửi mã xác nhận.
- Hệ thống gửi một mã xác nhận tới gmail của người dùng.
- Người dùng nhập mã xác nhận và chọn xác nhận.
- Hệ thống kiểm tra mã xác nhận hợp lệ hay không sau nếu hợp lệ chuyển hướng sang giao diện đổi mật khẩu.
- Người dùng nhập mật khẩu mới và xác nhận mật khẩu
- Hệ thống lưu mật khẩu đã thay đổi vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:
  - Người dùng hủy quên mật khẩu.
  - Hệ thống trở về giao diện đăng nhập.
  - Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin người dùng nhập gmail không hợp lệ.
  - Hệ thống hiển thị thông báo địa chỉ gmail chưa đăng ký.
  - Kết thúc usecase.
- Dòng sự kiện thứ ba:
  - Người dùng nhập mã xác thực sai.



- Hệ thống hiển thị thông báo sai mã xác thực.
- Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase quên mật khẩu tài khoản được thực hiện: người dùng chỉ cần vào ứng dụng sổ thu chi: quản lý tài chính trên điện thoại.
- Trạng thái hệ thống sau khi usecase quên mật tài khoản được thực hiện:
  - + Trường hợp thành công: Thông tin mật khẩu mới của người dùng được cập nhật trong hệ thống.
  - + Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật mật khẩu mới người dùng vào hệ thống.
- Điểm mở rộng: không.
- Giao diện: Giao diện quên mật khẩu.

#### **3.3.4.2 Đặc tả Usecase Quản lý khoản thu**

##### **\*Đặc tả usecase Lưu khoản thu**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc lưu khoản thu nhập của mình vào hệ thống
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn thêm ở menu giao diện chính
    - Hệ thống hiển thị ra giao diện thêm.
    - Người dùng chọn khoản thu.
    - Người dùng nhập các thông tin cần thiết như số tiền, nguồn thu, ngày, mô tả sau đó ấn lưu.
    - Hệ thống kiểm thông tin có hợp lệ không, nếu có thì thêm thông tin khoản thu vào hệ thống.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:
    - Dòng sự kiện thứ nhất:
      - Người dùng hủy thêm khoản thu.
      - Hệ thống trở về giao diện chính.

- Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin khoản thu không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase lưu khoản thu được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase lưu khoản thu được thực hiện:
  - + Trường hợp thành công: Thông tin khoản thu mới được thêm vào hệ thống.
  - + Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin khoản thu mới vào hệ thống.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng thêm.

#### **\*Đặc tả usecase Sửa khoản thu**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc sửa khoản thu của mình trong hệ thống khi có sai sót.
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn khoản thu ở menu giao diện chính
    - Hệ thống hiển thị ra giao diện danh sách các khoản thu.
    - Người dùng chọn khoản thu muốn sửa.
    - Hệ thống chuyển sang giao diện sửa khoản thu người dùng chọn.
    - Người dùng sửa các thông tin muốn sửa và ấn lưu.
    - Hệ thống kiểm thông tin có hợp lệ không, nếu có thì cập nhật thông tin khoản thu đã sửa vào hệ thống.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:
    - Dòng sự kiện thứ nhất:
      - Người dùng hủy sửa khoản thu.

- Hệ thống trở về giao diện chính.
- Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin sửa khoản chi không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase sửa khoản thu được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase sửa xóa khoản thu được thực hiện:
  - + Trường hợp thành công: Thông tin sửa của khoản thu được cập nhật vào hệ thống.
  - + Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin khoản thu đã sửa vào hệ thống.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng khoản thu.

#### **\*Đặc tả usecase Xóa khoản thu**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc xóa khoản thu của mình trong hệ thống.
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn khoản thu ở menu giao diện chính
    - Hệ thống hiển thị ra giao diện danh sách các khoản thu.
    - Người dùng chọn khoản thu muốn xóa.
    - Hệ thống chuyển sang giao diện sửa xóa khoản thu người dùng chọn.
    - Người dùng ấn xóa.
    - Hệ thống kiểm hợp lệ không, nếu có thì xóa thông tin khoản thu đã chọn khỏi hệ thống.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:
    - Dòng sự kiện thứ nhất:

- Người dùng hủy xóa khoản thu.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Xóa khoản thu không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase xóa khoản thu được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase xóa khoản thu được thực hiện:
  - + Trường hợp thành công: Xóa thông tin khoản thu khỏi hệ thống.
  - + Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin khoản thu khỏi hệ thống.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng khoản thu.

#### **\*Đặc tả usecase Tìm kiếm khoản thu theo thời gian**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc tìm kiếm khoản thu trong hệ thống.
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn khoản thu ở menu giao diện chính
    - Hệ thống hiển thị ra giao diện danh sách các khoản thu.
    - Người dùng chọn các lựa chọn ngày hoặc tháng hoặc năm.
    - Hệ thống chuyển sang danh sách các giao diện tương ứng.
    - Người dùng chọn thời gian cụ thể muốn xem.
    - Hệ thống hiển thị danh sách các khoản thu tương ứng theo thời gian người dùng chọn.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:
  - Người dùng hủy tìm kiếm khoản thu.
  - Hệ thống trở về giao diện khoản thu.
  - Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase tìm kiếm khoản thu theo thời gian được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase tìm kiếm khoản thu theo thời gian được thực hiện:
  - + Trường hợp thành công: Hiện thị danh sách các khoản thu theo thời gian được chọn.
  - + Trường hợp không thành công: Hệ thống thông báo không có dữ liệu.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng khoản thu.

### 3.3.4.3 Đặc tả Usecase Quản lý khoản chi

#### \*Đặc tả usecase Lưu khoản chi

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc lưu khoản chi của mình vào hệ thống
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn thêm ở menu giao diện chính
    - Hệ thống hiện thị ra giao diện thêm.
    - Người dùng chọn khoản chi.
    - Người dùng nhập các thông tin cần thiết như số tiền, danh mục, ngày, mô tả sau đó ấn lưu.
    - Hệ thống kiểm thông tin có hợp lệ không, nếu có thì thêm thông tin khoản chi vào hệ thống.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:
    - Dòng sự kiện thứ nhất:

- Người dùng hủy thêm khoản chi.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin khoản chi không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase lưu khoản chi được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase lưu khoản chi được thực hiện:
  - + Trường hợp thành công: Thông tin khoản chi mới được thêm vào hệ thống.
  - + Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin khoản chi mới vào hệ thống.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng thêm.

#### **\*Đặc tả usecase Sửa khoản chi**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc sửa khoản chi của mình trong hệ thống khi có sai sót.
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn khoản chi ở menu giao diện chính
    - Hệ thống hiển thị ra giao diện danh sách các khoản chi.
    - Người dùng chọn khoản chi muốn sửa.
    - Hệ thống chuyển sang giao diện sửa khoản chi người dùng chọn.
    - Người dùng sửa các thông tin muốn sửa và ấn lưu.
    - Hệ thống kiểm tra thông tin có hợp lệ không, nếu có thì cập nhật thông tin khoản chi đã sửa vào hệ thống.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:
  - Người dùng hủy sửa khoản chi.
  - Hệ thống trở về giao diện chính.
  - Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Thông tin sửa khoản chi không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.

- Trạng thái hệ thống trước khi usecase sửa khoản chi được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi usecase sửa xóa khoản chi được thực hiện:

+ Trường hợp thành công: Thông tin sửa của khoản chi được cập nhật vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin khoản chi đã sửa vào hệ thống.

- Diêm mở rộng: không.

- Giao diện: Giao diện chức năng khoản chi.

#### **\*Đặc tả usecase Xóa khoản chi**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc xóa khoản chi của mình trong hệ thống.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng chọn khoản thu ở menu giao diện chính
- Hệ thống hiện thị ra giao diện danh sách các khoản chi.
- Người dùng chọn khoản chi muốn xóa.
- Hệ thống chuyển sang giao diện sửa xóa khoản chi người dùng chọn.
- Người dùng ấn xóa.
- Hệ thống kiểm hợp lệ không, nếu có thì xóa thông tin khoản chi đã chọn khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:
  - Người dùng hủy xóa khoản chi.
  - Hệ thống trở về giao diện chính.
  - Kết thúc usecase.
- Dòng sự kiện thứ hai:
  - Xóa khoản chi không hợp lệ.
  - Hệ thống hiển thị thông báo lỗi.
  - Kết thúc usecase.

- Trạng thái hệ thống trước khi usecase xóa khoản chi được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi usecase xóa khoản chi được thực hiện:

+ Trường hợp thành công: Xóa thông tin khoản chi khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin khoản chi khỏi hệ thống.

- Điểm mở rộng: không.

- Giao diện: Giao diện chức năng khoản chi.

#### **\*Đặc tả usecase Tìm kiếm khoản chi theo thời gian**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc tìm kiếm khoản chi trong hệ thống.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng chọn khoản chi ở menu giao diện chính
- Hệ thống hiển thị ra giao diện danh sách các khoản chi.
- Người dùng chọn các lựa chọn ngày hoặc tháng hoặc năm.
- Hệ thống chuyển sang danh sách các giao diện tương ứng.
- Người dùng chọn thời gian cụ thể muốn xem.
- Hệ thống hiển thị danh sách các khoản chi tương ứng theo thời gian người dùng chọn.
- Kết thúc usecase.

+ Dòng sự kiện phụ:



- Dòng sự kiện thứ nhất:
  - Người dùng hủy tìm kiếm khoản chi.
  - Hệ thống trở về giao diện khoản chi.
  - Kết thúc usecase.
- Trạng thái hệ thống trước khi usecase tìm kiếm khoản chi theo thời gian được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase tìm kiếm khoản chi theo thời gian được thực hiện:
  - + Trường hợp thành công: Hiện thị danh sách các khoản chi theo thời gian được chọn.
  - + Trường hợp không thành công: Hệ thống thông báo không có dữ liệu.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng khoản chi.

#### **3.3.4.4 Đặc tả Usecase Thống kê**

##### **\*Đặc tả usecase Thống kê khoản thu và chi**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc xem tình hình thu chi của mình.
- Dòng sự kiện:
  - + Dòng sự kiện chính:
    - Người dùng chọn thống kê ở menu giao diện chính
    - Hệ thống hiện thị ra giao diện thống kê.
    - Người dùng chọn các lựa chọn ngày hoặc tháng hoặc năm.
    - Hệ thống chuyển sang danh sách các giao diện tương ứng.
    - Người dùng chọn thời gian cụ thể muốn xem.
    - Hệ thống hiển thị ra biểu đồ cột thể hiện tình hình thu và chi của người dùng trong khoảng thời gian mà người dùng thống kê.
    - Kết thúc usecase.
  - + Dòng sự kiện phụ:
    - Dòng sự kiện thứ nhất:

- Người dùng hủy chọn thời gian.
  - Hệ thống trở về giao diện thống kê.
  - Kết thúc usecase.
- Trang thái hệ thống trước khi usecase thống kê được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trang thái hệ thống sau khi usecase thống kê được thực hiện:
- + Trường hợp thành công: Hiện thị biểu đồ cột thống kê khoản thu và chi theo thời gian được chọn.
- + Trường hợp không thành công: Hệ thống thông báo không có dữ liệu.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng thống kê.

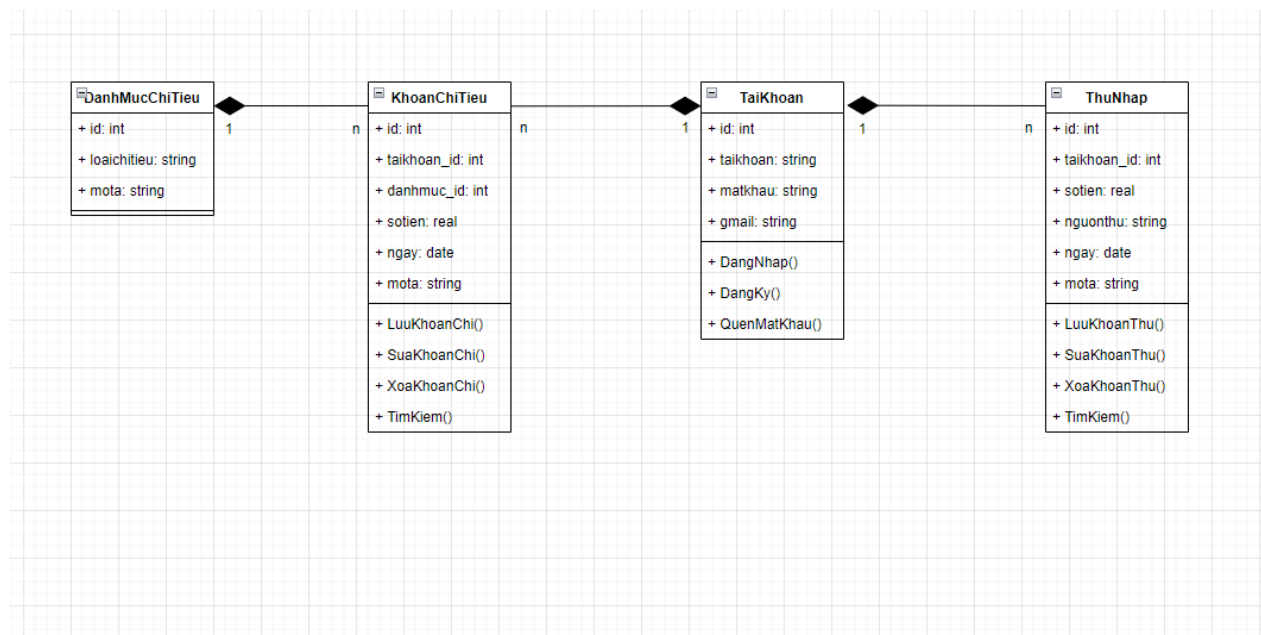
#### **\*Đặc tả usecase Thống kê danh mục chi tiêu**

- Tóm tắt: Người dùng sử dụng usecase này để thực hiện việc xem tình hình chi tiêu theo danh mục của mình.
- Dòng sự kiện:
- + Dòng sự kiện chính:
- Người dùng chọn thống kê ở menu giao diện chính
  - Hệ thống hiện thị ra giao diện thống kê.
  - Người dùng chọn các lựa chọn ngày hoặc tháng hoặc năm.
  - Hệ thống chuyển sang danh sách các giao diện tương ứng.
  - Người dùng chọn thời gian cụ thể muốn xem.
  - Hệ thống hiện thị ra biểu đồ tròn thể hiện tình hình chi tiêu theo các danh mục của người dùng trong khoảng thời gian mà người dùng thống kê.
  - Kết thúc usecase.
- + Dòng sự kiện phụ:
- Dòng sự kiện thứ nhất:
    - Người dùng hủy chọn thời gian.
    - Hệ thống trở về giao diện thống kê.
    - Kết thúc usecase.

- Trạng thái hệ thống trước khi usecase thống kê được thực hiện: người dùng cần đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi usecase thống kê được thực hiện:
  - + Trường hợp thành công: Hiển thị biểu đồ tròn thống kê khoản chi tiêu theo danh mục theo thời gian được chọn.
  - + Trường hợp không thành công: Hệ thống thông báo không có dữ liệu.
- Điểm mở rộng: không.
- Giao diện: Giao diện chức năng thống kê.

### 3.4 Biểu đồ lớp


#### 3.4.1 Biểu đồ lớp tổng quát



Hình 3.12: Biểu đồ lớp

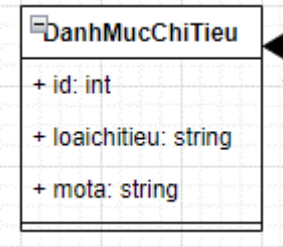
#### 3.4.2 Đặc tả biểu đồ lớp

Bảng tbl3.4 :Mô tả lớp TaiKhoan

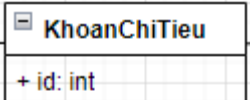
 <b>TaiKhoan</b> + id: int	Định nghĩa	Chứa các thông tin của tài khoản

	Thuộc tính	<p>id : int: chứa id của tài khoản người dùng</p> <p>taikhoan: string: Chứa tên tài khoản của người dùng</p> <p>matkhau: string: Chứa mật khẩu tài khoản của người dùng.</p> <p>gmail: string: Chứa thông tin địa chỉ gmail của người dùng.</p>
	Phương thức	<p>DangNhap(): Thực hiện chức năng đăng nhập vào hệ thống bằng tài khoản và mật khẩu.</p> <p>DangKy(): Thực hiện việc đăng ký tài khoản mới để đăng nhập vào hệ thống.</p> <p>QuenMatKhau(): Thực hiện việc thay đổi mật khẩu mới khi quên mật khẩu hiện tại.</p>

Bảng tbl3.5 :Mô tả lớp

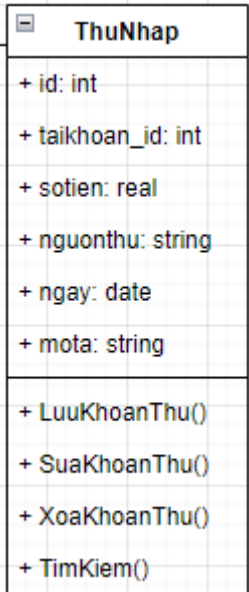
 <pre> classDiagram     class DanhMucChiTieu {         +id: int         +loaichitieu: string         +mota: string     } </pre>	Định nghĩa	Chứa các thông tin của danh mục chi tiêu
	Thuộc tính	<p>id: int: chứa id của danh mục chi tiêu</p> <p>loaichitieu: string: chứa các loại chi tiêu như ăn uống, đi lại, dịch vụ,...</p> <p>mota: string: chứa mô tả của loại chi tiêu.</p>
	Phương thức	

Bảng tbl3.6 :Mô tả lớp

 <pre> classDiagram     class KhoanChiTieu {         +id: int     } </pre>	Định nghĩa	Chứa các thông tin của khoản chi tiêu
	Thuộc tính	id: int: chứa id của khoản chi tiêu

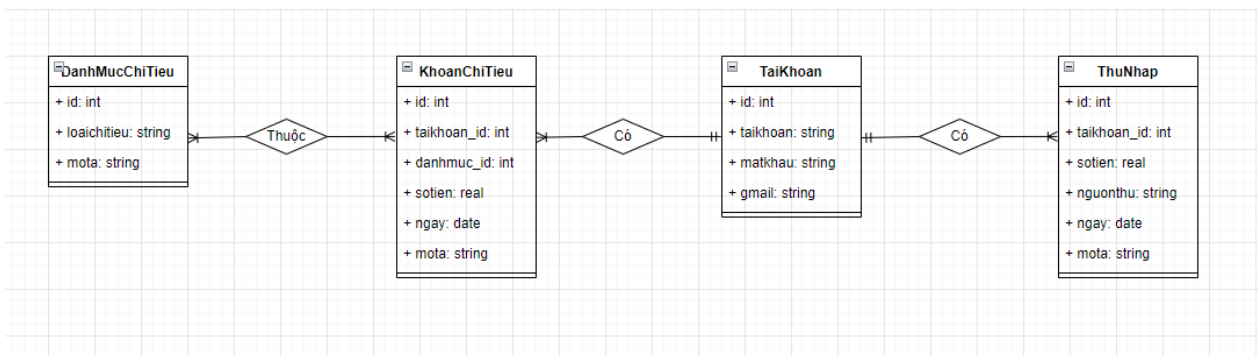
		taikhoan_id: int: chứa id của tài khoản người dùng danhmuc_id: int: chứa id của danh mục chi tiêu sotien: real: chứa số tiền của khoản chi tiêu ngay: string: chứa ngày khoản chi tiêu mota: string: chứa mô tả của khoản chi tiêu
	Phương thức	LuuKhoanChi(): Thực hiện việc lưu một khoản chi tiêu mới vào hệ thống SuaKhoanChi(): Thực hiện việc sửa thông tin của một khoản chi tiêu mong muốn trong hệ thống XoaKhoanChi(): Thực hiện việc xóa một khoản chi tiêu trong hệ thống. TimKiem(): Thực hiện việc tìm kiếm các khoản chi tiêu theo thời gian mong muốn.

Bảng tbl3.7 :Mô tả lớp

 <pre> classDiagram     class ThuNhap {         +id: int         +taikhoan_id: int         +sotien: real         +nguồnthu: string         +ngay: date         +mota: string         +LuuKhoanThu()         +SuaKhoanThu()         +XoaKhoanThu()         +TimKiem()     }           </pre>	Định nghĩa	Chứa các thông tin của khoản thu nhập
	Thuộc tính	id: int: chứa id của khoản thu taikhoan_id: int: chứa id của tài khoản người dùng sotien: real: chứa số tiền của khoản thu nguonthu: string: chứa thông tin nguồn thu của khoản thu ngay: string: chứa ngày khoản thu mota: string: chứa mô tả của khoản thu
	Phương thức	LuuKhoanThu(): Thực hiện việc lưu một khoản thu mới vào hệ thống

		<p>SuaKhoanThu(): Thực hiện việc sửa thông tin của một khoản thu mong muốn trong hệ thống</p> <p>XoaKhoanThu(): Thực hiện việc xóa một khoản thu trong hệ thống.</p> <p>TimKiem(): Thực hiện việc tìm kiếm các khoản thu theo thời gian mong muốn.</p>
--	--	--

### 3.3 Biểu đồ mối quan hệ thực thể



Hình 3.13: Biểu đồ mối quan hệ thực thể

### 3.5 Các bảng trong cơ sở dữ liệu

Bảng tbl3.8 :tblTaiKhoan

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	id	int	not null	PK	id của tài khoản
2	taikhoan	text	not null		tên tài khoản
3	matkhou	text	not null		mật khẩu tài khoản
4	gmail	text	not null		địa chỉ gmail

Bảng tbl3.9 :tblDanhMucChiTieu

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	id	int	not null	PK	id của loại chi tiêu
2	loaichitieu	text	not null		tên loại chi tiêu

3	mota	text	not null		mô tả
---	------	------	----------	--	-------

Bảng tbl3.10 :tblKhoanChiTieu

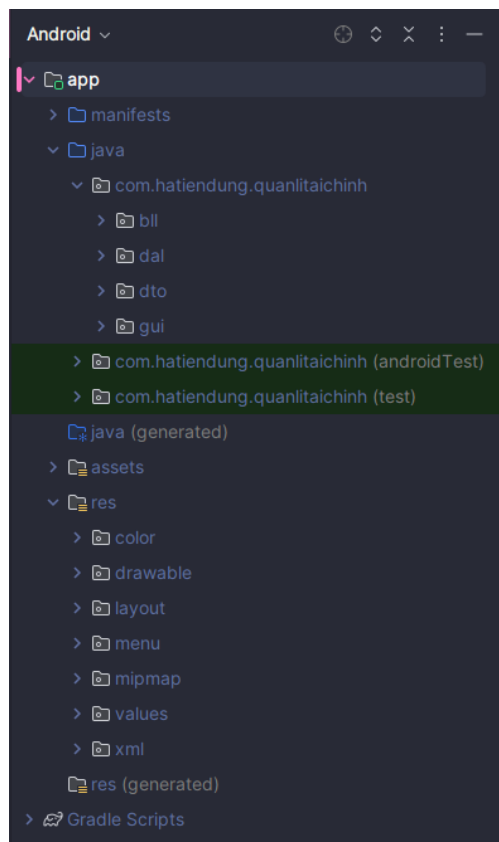
STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	id	int	not null	PK	id của khoản chi
2	taikhoan_id	int	not null	FK	id của tài khoản
3	danhmuc_id	int	not null	FK	id của danh mục
4	sotien	real	not null		số tiền của khoản chi
5	ngay	text	not null		ngày của khoản chi
6	mota	text			mô tả

Bảng tbl3.11 :tblThuNhap

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	id	int	not null	PK	id của khoản thu
2	taikhoan_id	int	not null	FK	id của tài khoản
3	sotien	real	not null		số tiền của khoản chi
4	nguồnthu	text	not null		nguồn thu của khoản thu
5	ngay	text	not null		ngày của khoản thu
6	mota	text			mô tả

## CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH

### 4.1 Cấu trúc mã nguồn chương trình



Hình 4.1: Cấu trúc mã nguồn chương trình

\* Thư mục manifests: Chứa AndroidManifest.xml, file cấu hình chính của ứng dụng Android. File này định nghĩa các thành phần cơ bản như: các activity, services, permissions mà ứng dụng cần và các cấu hình khác của ứng dụng.

\* Thư mục java: Chứa các package được tạo và tự sinh:

- package com.hatiendung.quanlitaichinh: Đây là package chính của dự án bao gồm các lớp và các thành phần phục vụ cho việc phát triển ứng dụng:

+ Package bll: Là tầng Business Logic Layer. Chứa các lớp xử lý logic chính của ứng dụng, chẳng hạn như tính toán, quản lý dữ liệu hoặc điều khiển các thành phần khác.

+ Package dal: Là tầng Data Access Layer: Chứa các lớp quản lý việc lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu.

+ Package dto: Là tầng Data Transfer Object: Chứa các lớp đại diện cho một đối tượng giúp trao đổi dữ liệu với các thành phần khác trong ứng dụng.



+ Package gui: Graphical User Interface: Chứa các lớp đại diện cho giao diện người dùng là các activity và fragment.

- package com.hatiendung.quanlitaichinh (androidTest): Đây là thư mục chứa các lớp và tài nguyên để kiểm thử tự động (UI tests) chạy trên thiết bị Android thực hoặc máy ảo.

- package com.hatiendung.quanlitaichinh (test): chứa các bài kiểm thử đơn vị (Unit tests) cho các lớp trong ứng dụng. Các bài kiểm thử này thường được chạy trên JVM (Java Virtual Machine) thay vì thiết bị Android thực.

\* Thư mục res: Thư mục chứa tài nguyên của ứng dụng:

- color: Chứa các định nghĩa về màu sắc (tệp XML) mà bạn có thể sử dụng trong giao diện người dùng.

- drawable: Chứa các tệp đồ họa (ảnh, hình vẽ vector, v.v.) được sử dụng trong ứng dụng.

- layout: Chứa các tệp XML định nghĩa bố cục (layout) của giao diện người dùng.

- menu: Chứa các tệp XML định nghĩa các menu của ứng dụng.

- mipmap: Thư mục này chứa các tài nguyên hình ảnh như icon của ứng dụng. Đây là nơi chứa các kích thước khác nhau của một icon để tương thích với các độ phân giải màn hình khác nhau.

- values: Chứa các tệp XML định nghĩa các giá trị chung như chuỗi ký tự, kích thước, kiểu dáng,...

- xml: Chứa các tệp cấu hình XML khác.

\* Thư mục Gradle Scripts: Thư mục này chứa các tệp cấu hình Gradle, một công cụ build tự động được Android Studio sử dụng. Nó quản lý các dependency, cấu hình dự án và quá trình build ứng dụng.

## 4.2 Các thiết kế giao diện chương trình

### 4.2.1 Giao diện đăng nhập



Hình 4.2: Giao diện đăng nhập

Bảng tbl4.1 :Bảng thành phần chính của giao diện đăng nhập

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện đăng nhập	
2	txtTaiKhoan	EditText	Nơi điền tên tài khoản đăng nhập	
3	txtMatKhau	EditText	Nơi điền mật khẩu đăng nhập	
4	btnDangNhap	Button	Nút đăng nhập tài khoản	
5	btnOpenQuenMK	TextView	Nút mở giao diện quên mật khẩu	
6	btnOpenDangKy	TextView	Nút mở giao diện đăng ký tài khoản	

### 4.2.2 Giao diện đăng ký

Hình 4.3: Giao diện đăng ký

Bảng tbl4.2 :Bảng thành phần chính của giao diện đăng ký

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện đăng ký	
2	txtTaiKhoanDK	EditText	Nơi nhập tên tài khoản	
3	txtMatKhouDK	EditText	Nơi nhập mật khẩu tài khoản	
4	txtGmailDK	EditText	Nơi nhập gmail đăng ký	
5	txtMaXacThucDK	EditText	Nơi nhập mã xác thực	
6	btnGuiMaXacThucDK	Button	Nút gửi mã xác thực	
7	btnDangKyTK	Button	Nút đăng ký	
8	btnQuayLaiDangNhap	TextView	Nút quay lại giao diện đăng nhập	

### 4.2.3 Giao diện quên mật khẩu

Hình 4.4: Giao diện quên mật khẩu

Bảng tbl4.3 :Bảng thành phần chính của giao diện quên mật khẩu

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện quên mật khẩu	
2	txtGmailQMK	EditText	Nơi nhập gmail	
3	txtTaiKhoanQMK	EditText	Nơi hiển thị tên tài khoản	
4	txtMaXacNhanQMK	EditText	Nơi nhập mã xác thực	
5	btnChonGmail	Button	Nút tìm tên tài khoản bằng gmail	
6	btnGuiMaQMK	Button	Nút gửi mã xác thực	
7	btnXacNhanQMK	Button	Nút xác nhận mã xác thực	

#### 4.2.4 Giao diện đổi mật khẩu

ĐỔI MẬT KHẨU

Nhập mật khẩu mới...

Xác nhận mật khẩu mới...

☐ Hiện thị mật khẩu

Đổi mật khẩu

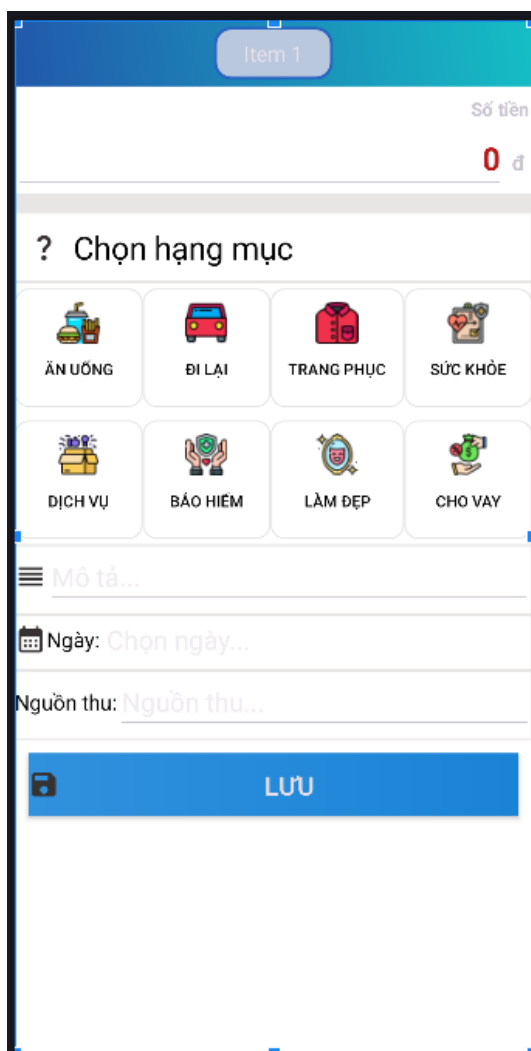
[Quay lại trang đăng nhập?](#)

Hình 4.5: Giao diện đổi mật khẩu

Bảng tbl4.4 :Bảng thành phần chính của giao diện đổi mật khẩu

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện đổi mật khẩu	
2	txtMatKhouMoi	EditText	Nơi nhập mật khẩu mới	
3	txtXNMatKhouMoi	EditText	Nơi xác nhận mật khẩu mới	
4	ckbCheckShowHideMK	CheckBox	Nút hiện thị mật khẩu	
5	btnDoiMatKhou	Button	Nút đổi mật khẩu	
6	btnQuayLaiTrangDangNhap	TextView	Nút quay lại trang đăng nhập	

#### 4.2.5 Giao diện thêm khoản thu, khoản chi



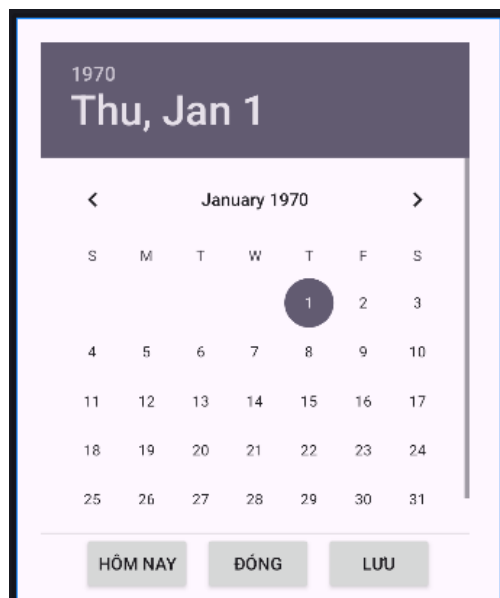
Hình 4.6: Giao diện thêm khoản thu, khoản chi

Bảng tbl4.5 :Bảng thành phần chính của giao diện thêm khoản thu, khoản chi

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	txtHangMuc	TextView	Hiện thị hạng mục được chọn	Khoản chi
2	txtSoTien	EditText	Nơi nhập số tiền	
3	txtMota	EditText	Nơi nhập mô tả	
4	txtChonNgay	EditText	Nơi chọn ngày	
5	txtNgonThu	EditText	Nơi nhập nguồn thu	Khoản thu
6	btnAnUong	Button	Nút chọn hạng mục ăn uống	Khoản chi
7	btnDiLai	Button	Nút chọn hạng mục đi lại	Khoản chi

8	btnTrangPhuc	Button	Nút chọn hạng mục trang phục	Khoản chi
9	btnSucKhoe	Button	Nút chọn hạng mục sức khỏe	Khoản chi
10	btnDichVu	Button	Nút chọn hạng mục dịch vụ	Khoản chi
11	btnBaoHiem	Button	Nút chọn hạng mục bảo hiểm	Khoản chi
12	btnLamDep	Button	Nút chọn hạng mục làm đẹp	Khoản chi
13	btnChoVay	Button	Nút chọn hạng mục cho vay	Khoản chi
14	btnLuu	Button	Nút lưu	
15	spinner_chonmuc	Spinner	Nút chọn khoản thu hoặc khoản chi	

#### 4.2.6 Giao diện form chọn ngày

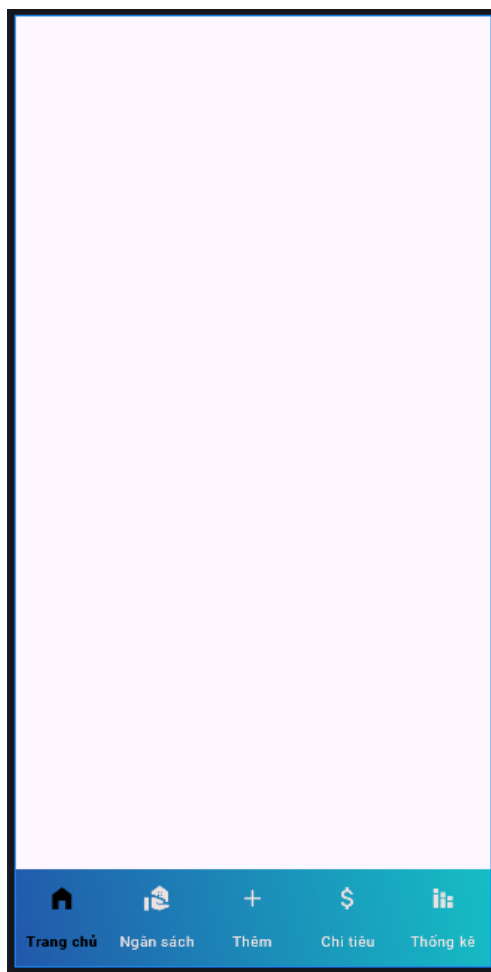


Hình 4.7: Giao diện form chọn ngày

Bảng tbl4.6 :Bảng thành phần chính của giao diện form chọn ngày

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	datepicker	DatePicker	Nơi chọn ngày	
2	btnToday	Button	Nút chọn ngày hôm nay	
3	btnCloseDate	Button	Nút đóng form chọn ngày	
4	btnSaveDate	Button	Nút lưu ngày đã chọn	

#### 4.2.7 Giao diện trang chính



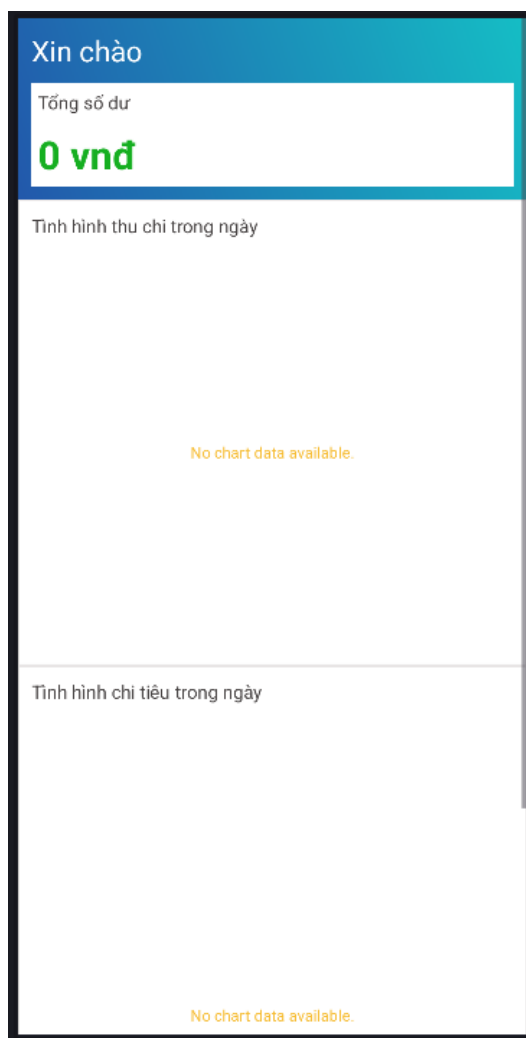
Hình 4.8: Giao diện trang chính

Bảng tbl4.7 :Bảng thành phần chính của giao diện trang chính

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	frame_layout_view	FrameLayout	Hiện các giao diện Fragment	
2	bottomNavigationView	BottomNavigationView	Hiện thị menu các chức năng	



#### 4.2.8 Giao diện trang chủ



Hình 4.9: Giao diện trang chủ

Bảng tbl4.8 :Bảng thành phần chính của giao diện trang chủ

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	txtXinChaoTK	TextView	Tiêu đề xin chào tài khoản	
2	txtTongSoDu	TextView	Hiển thị tổng số dư	
3	barChartThuChi	BarChart	Biểu đồ cột tình hình thu chi	
4	PiaChart	PieChart	Biểu đồ tròn chi tiêu theo danh mục	

#### 4.2.9 Giao diện ngân sách

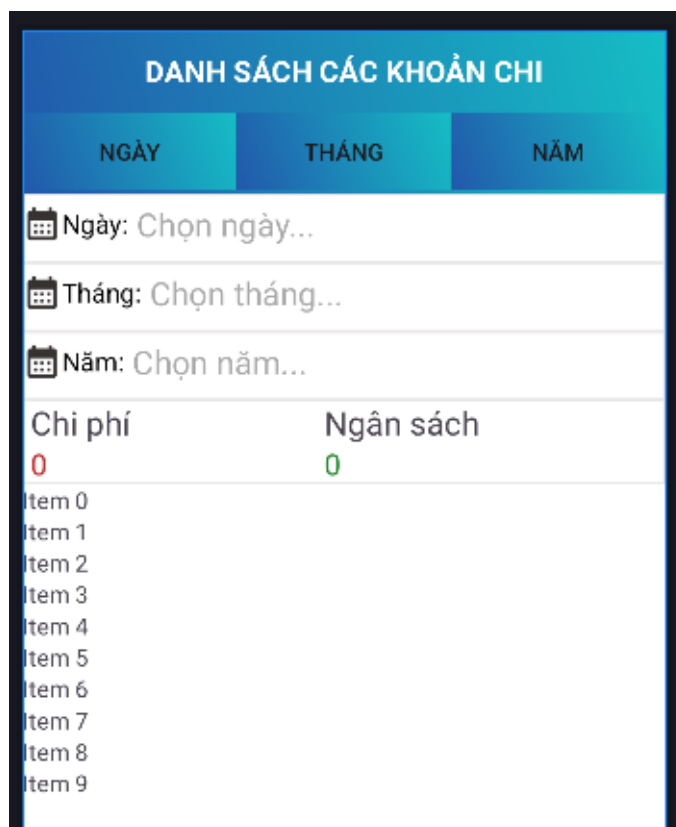


Hình 4.10: Giao diện ngân sách

Bảng tbl4.9 :Bảng thành phần chính của giao diện ngân sách

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện ngân sách	
2	txtChonNgaySKT	TextView	chọn ngày	
3	txtChonThangSKT	TextView	chọn tháng	
4	txtChonNamSKT	TextView	chọn năm	
5	txtNganSachTong	TextView	hiển thị ngân sách tổng	
6	txtNganSachHT	TextView	hiển thị ngân sách hiện tại	
7	chonNgaySKT	Button	chọn hiển thị theo ngày	
8	chonThangSKT	Button	chọn hiển thị theo tháng	
9	chonNamSKT	Button	chọn chọn hiển thị theo năm	
10	list_itemKT	RecyclerView	hiển thị danh sách khoản thu	

#### 4.2.10 Giao diện khoản chi



Hình 4.11: Giao diện khoản chi

Bảng tbl4.10 :Bảng thành phần chính của giao diện khoản chi

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện khoản chi	
2	txtChonNgaySKC	TextView	chọn ngày	
3	txtChonThangSKC	TextView	chọn tháng	
4	txtChonNamSKC	TextView	chọn năm	
5	txtChiPhiKC	TextView	hiển thị chi phí	
6	txtNganSachKC	TextView	hiển thị ngân sách hiện tại	
7	chonNgaySKC	Button	chọn hiển thị theo ngày	
8	chonThangSKC	Button	chọn hiển thị theo tháng	
9	chonNamSKC	Button	chọn chọn hiển thị theo năm	
10	list_itemKC	RecyclerView	hiển thị danh sách khoản chi	

#### 4.2.11 Giao diện thống kê

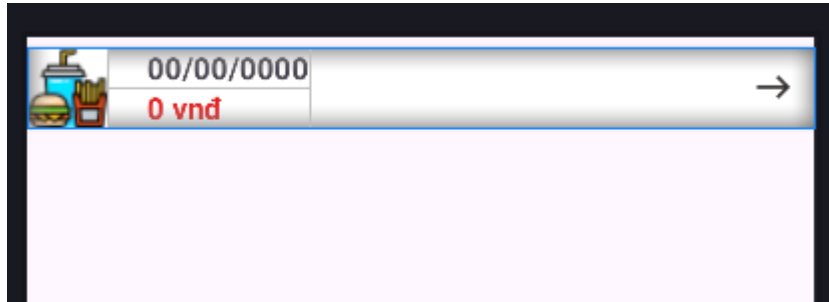
Hình 4.12: Giao diện thống kê

Bảng tbl4.11 :Bảng thành phần chính của giao diện thống kê

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1		TextView	Tiêu đề giao diện thống kê	
2	txtChonNgayTK	TextView	chọn ngày	
3	txtChonThangTK	TextView	chọn tháng	
4	txtChonNamTK	TextView	chọn năm	
5	chonNgayTK	Button	chọn thống kê theo ngày	
6	chonThangTK	Button	chọn thống kê theo tháng	
7	chonNamTK	Button	chọn thống kê theo năm	

8	barChartThuChiTK	BarChart	Biểu đồ cột thu chi	
9	pieChartTK	PieChart	Biểu đồ trong danh mục chi	

#### 4.2.12 Giao diện item recycle view khoản chi

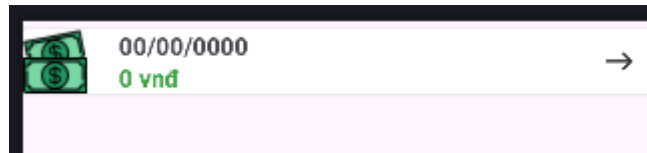


Hình 4.13: Giao diện recycle view khoản chi

Bảng tbl4.12 :Bảng thành phần chính của giao diện

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	itemiconKhoanChi	ImageView	icon của danh mục khoản chi	
2	itemNgayKhoanChi	TextView	ngày của khoản chi	
3	itemSoTienKhoanChi	TextView	số tiền của khoản chi	
4		TextView	Hiển thị icon next	

#### 4.2.13 Giao diện item recycle view khoản thu



Hình 4.14: Giao diện recycle view khoản thu

Bảng tbl4.13 :Bảng thành phần chính của giao diện

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	itemiconKhoanThu	ImageView	icon khoản thu	
2	itemNgayKhoanThu	TextView	ngày của khoản thu	
3	itemSoTienKhoanThu	TextView	số tiền của khoản thu	
4		TextView	Hiển thị icon next	

#### 4.2.14 Giao diện sửa, xóa của khoản thu, khoản chi

THÔNG TIN CHI TIẾT

Số tiền

0 đ

? Chọn hạng mục

ĂN UỐNG

ĐI LẠI

TRANG PHỤC

SỨC KHỎE

DỊCH VỤ

BẢO HIỂM

LÀM ĐẸP

CHO VAY

☰

Mô tả...

📅

Ngày: Chọn ngày...

📄

Nguồn thu: Nguồn thu...

💾

LƯU

🗑️

XÓA

Hình 4.15: Giao diện sửa, xóa của khoản thu, khoản chi

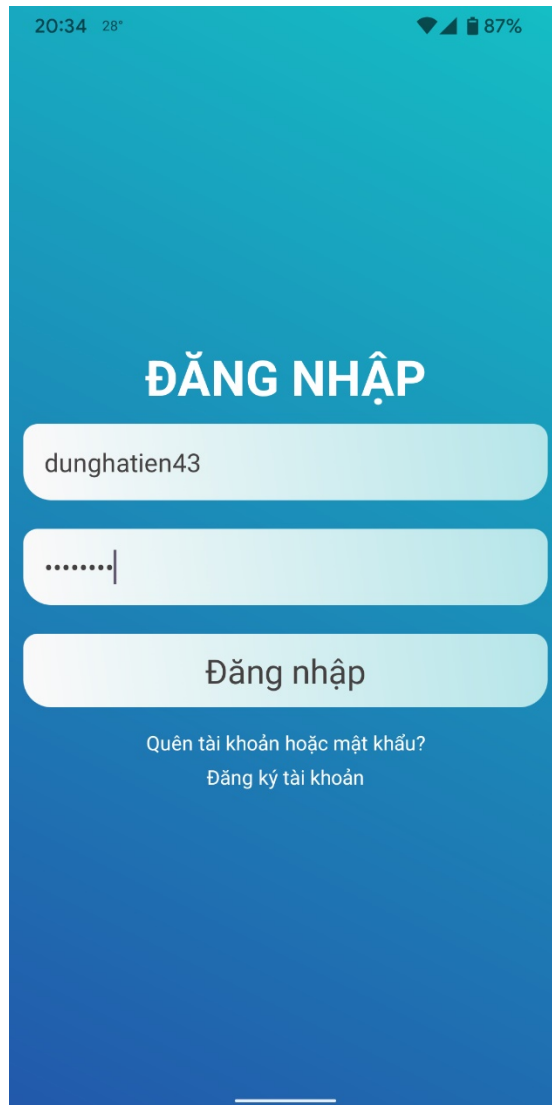
Bảng tbl4.14 :Bảng thành phần chính của giao diện thêm khoản thu, khoản chi

STT	Tên đối tượng	Kiểu	Ý nghĩa	Ghi chú
1	txtHangMucSuaXoa	TextView	Hiển thị hạng mục đã chọn	Khoản chi
2	txtSoTienSuaXoa	EditText	Nơi nhập số tiền	
3	txtMotaSuaXoa	EditText	Nơi nhập mô tả	
4	txtChonNgaySuaXoa	EditText	Nơi chọn ngày	
5	txtNguonThuSuaXoa	EditText	Nơi nhập nguồn thu	Khoản thu
6	btnAnUongSuaXoa	Button	nút chọn hạng mục ăn uống	Khoản chi

7	btnDiLaiSuaXoa	Button	Nút chọn hạng mục đi lại	Khoản chi
8	btnTrangPhucSuaXoa	Button	Nút chọn hạng mục trang phục	Khoản chi
9	btnSucKhoeSuaXoa	Button	Nút chọn hạng mục sức khỏe	Khoản chi
10	btnDichVuSuaXoa	Button	Nút chọn hạng mục dịch vụ	Khoản chi
11	btnBaoHiemSuaXoa	Button	Nút chọn hạng mục bảo hiểm	Khoản chi
12	btnLamDepSuaXoa	Button	Nút chọn hạng mục làm đẹp	Khoản chi
13	btnChoVaySuaXoa	Button	Nút chọn hạng mục cho vay	Khoản chi
14	btnSua	Button	Nút sửa	
15	btnXoa	Button	Nút xóa	
16		TextView	Tiêu đề giao diện sửa, xóa	

## CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH

- Khi mở ứng dụng giao diện đăng nhập sẽ hiển thị người dùng, giao diện đăng nhập sẽ có 3 lựa chọn là đăng nhập, đăng ký và quên mật khẩu. Người dùng nhập tài khoản và mật khẩu sau đó ấn đăng nhập để truy cập vào giao diện chức năng.
- Giao diện khi người dùng đăng nhập:



Hình 5.1: Giao diện người dùng đăng nhập ứng dụng



- Nếu người dùng muốn đăng ký tài khoản, đầu tiên chọn đăng ký tài khoản ở giao diện đăng nhập. Sau đó nhập tên tài khoản, mật khẩu và gmail. Sau đó chọn gửi mã, ứng dụng sẽ gửi một mã xác thực đến gmail mà người dùng đăng ký. Cuối cùng người dùng nhập mã xác thực được gửi đến nếu chính xác tài khoản sẽ được đăng ký thành công.

- Giao diện khi đăng ký tài khoản:

20:34 28° 87%

## ĐĂNG KÝ TÀI KHOẢN

nguyenvandat

.....

aesusu111@gmail.com

27075 Gửi mã

Đăng ký

Đã có tài khoản? Đăng nhập.

Hình 5.2: Giao diện người dùng đăng ký tài khoản

- Nếu người dùng quên mật khẩu tài khoản có thể chọn mục quên tài khoản hoặc mật khẩu. Ứng dụng sẽ chuyển sang giao diện quên mật khẩu. Người dùng nhập gmail đã đăng ký và ấn tìm, tài khoản tương ứng với gmail sẽ được hiển thị. Sau đó chọn gửi mã và nhập mã được gửi đến gmail và ấn xác nhận. Nếu chính xác sẽ chuyển sang giao diện thay mật khẩu mới.

- Giao diện khi người dùng quên mật khẩu:

20:33 28° 87%

## QUÊN MẬT KHẨU

dunghasonlla@gmail.com Tìm

dunghatien43 Gửi mã

78194

Xác nhận

Hình 5.3: Giao diện người dùng quên mật khẩu

- Sau chuyển sang giao diện thay mật khẩu, người dùng nhập mật khẩu mới và xác nhận mật khẩu mới và chọn đổi mật khẩu. Sau đó ứng dụng sẽ quay trở lại giao diện đăng nhập.
- Giao diện khi người dùng thay mật khẩu:

20:33 28° 87%

## ĐỔI MẬT KHẨU

.....

.....

☐ Hiện thị mật khẩu

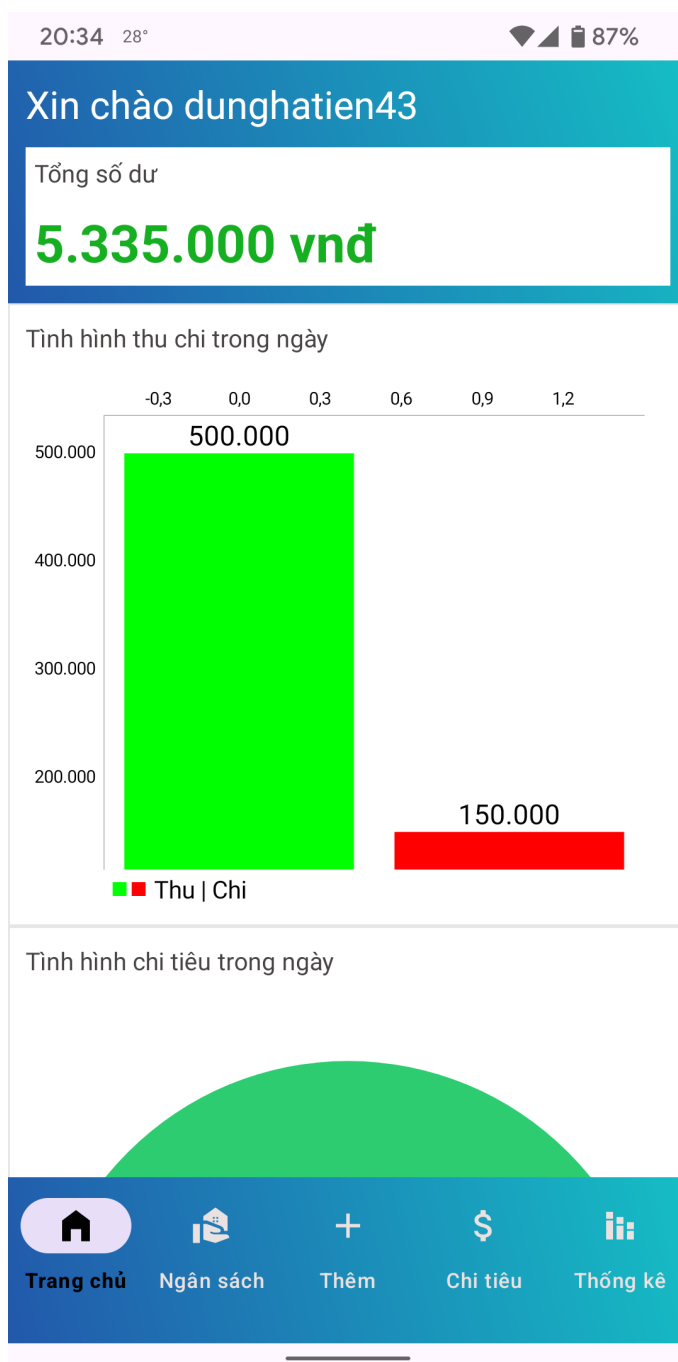
Đổi mật khẩu

[Quay lại trang đăng nhập?](#)

Hình 5.4: Giao diện người dùng đổi mật khẩu

Sau khi đăng nhập thành công, người dùng được chuyển đến giao diện chính các chức năng. Hiển thị mặc định là giao diện trang chủ gồm có: Tiêu đề xin chào tài khoản, số dư hiện tại của người dùng, biểu đồ cột thể hiện tình hình thu và chi trong ngày hiện tại và biểu đồ tròn thể hiện các loại danh mục chi tiêu trong ngày hiện tại.

- Giao diện khi đăng nhập thành công:



Hình 5.5: Giao diện chính các chức năng của người dùng

- Để thêm khoản chi người dùng ấn chức năng thêm ở menu chức năng ở dưới. giao diện thêm khoản chi sẽ xuất hiện người dùng ấn vào spinner ở trên và chọn khoản chi để thêm khoản chi. Giao diện sẽ hiển thị các thành phần để thêm khoản chi. Người dùng nhập số tiền, chọn danh mục chi tiêu, nhập ghi chú, chọn ngày và ấn lưu để thêm khoản chi. Hệ thống sẽ thông báo thêm thành công nếu hợp lệ.

- Giao diện khi thêm khoản chi:

20:35 28° 87%

Khoản chi

Số tiền

**525000 đ**

? Sức khỏe

ĂN UỐNG	ĐI LẠI	TRANG PHỤC	SỨC KHỎE
DỊCH VỤ	BẢO HIỂM	LÀM ĐẸP	CHO VAY

☰ Khám sức khỏe

📅 Ngày: 27-09-2024

📁 LƯU

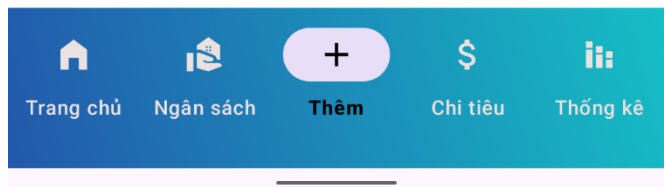
Trang chủ Ngân sách **Thêm** Chi tiêu Thống kê

Hình 5.6: Giao diện khi thêm khoản chi mới.

- Để thêm khoản thu, người dùng ấn chức năng thêm ở thanh menu bên dưới. Giao diện thêm sẽ hiển thị người dùng ấn vào spinner và chọn khoản thu để thêm khoản thu, giao diện sẽ hiển thị các thành phần thích hợp để thêm khoản thu. Tiếp theo người dùng nhập số tiền, nhập mô tả, chọn ngày, nhập nguồn thu và ấn lưu để thêm khoản thu mới. Nếu hợp lệ hệ thống sẽ thông báo thành công.

- Giao diện khi thêm khoản thu mới:

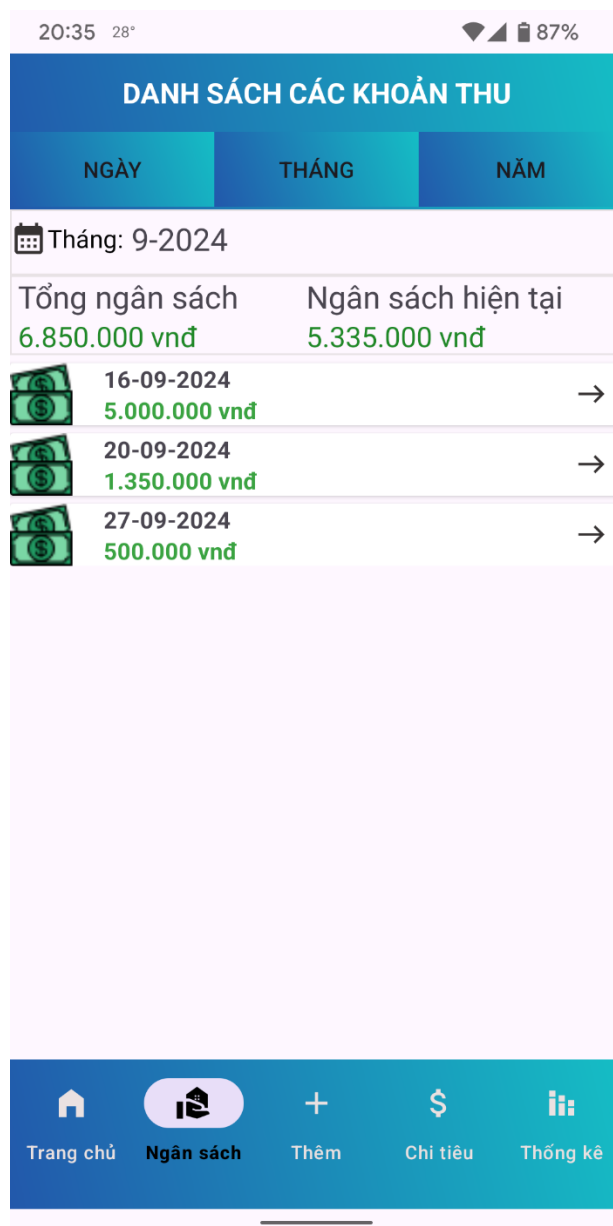
The screenshot shows a mobile application interface for adding a new income entry. At the top, there is a status bar with the time 20:36, temperature 28°, and battery level 86%. Below the status bar is a header with a blue gradient and a white button labeled "Nguồn thu". The main content area has a white background and contains the following elements: a label "Số tiền" (Amount) followed by the value "5000000 đ" in red; a section with a grey header containing a hamburger menu icon and the text "Lương làm thêm tháng 9"; a date field with a calendar icon and the text "Ngày: 27-09-2024"; a source field with the text "Nguồn thu: Lương"; and a blue button with a floppy disk icon and the text "LƯU".



Hình 5.7: Giao diện khi thêm khoản thu mới.

- Để xem danh sách các khoản thu đã thêm người dùng chọn chức năng ngân sách ở menu các chức năng bên dưới, giao diện danh sách các khoản thu sẽ được hiển thị bao gồm tiêu đề chọn hiển thị theo ngày, tháng hoặc năm. Người dùng có thể chọn ngày, tháng, năm để hiển thị các khoản thu cùng thời gian tương ứng. Giao diện còn hiển thị tổng các khoản thu đã thêm và ngân sách hiện tại. Người dùng có thể chọn vào khoản thu ở danh sách nếu muốn chỉnh sửa hoặc xóa.

- Giao diện khi xem danh sách các khoản thu:



Hình 5.8: Giao diện chức năng ngân sách.

- Để sửa hoặc xóa khoản thu người dùng chọn một khoản thu muốn sửa hoặc xóa trong danh sách các khoản thu. Giao diện sửa hoặc xóa của khoản thu đó sẽ được hiển thị, người dùng sửa các thông tin muốn sửa sau đó ấn lưu để cập nhật khoản thu muốn sửa hoặc ấn xóa nếu như muốn xóa khoản thu đó.

- Giao diện khi sửa hoặc xóa khoản thu:

20:37 28°

86%

THÔNG TIN CHI TIẾT

Số tiền

500000 đ

☰

Mẹ cho

📅

Ngày: 27-09-2024

Nguồn thu: Gia đình

💾

LƯU

🗑

XÓA

🏠

Trang chủ

🏠

Ngân sách

+

Thêm

\$

Chi tiêu

📊

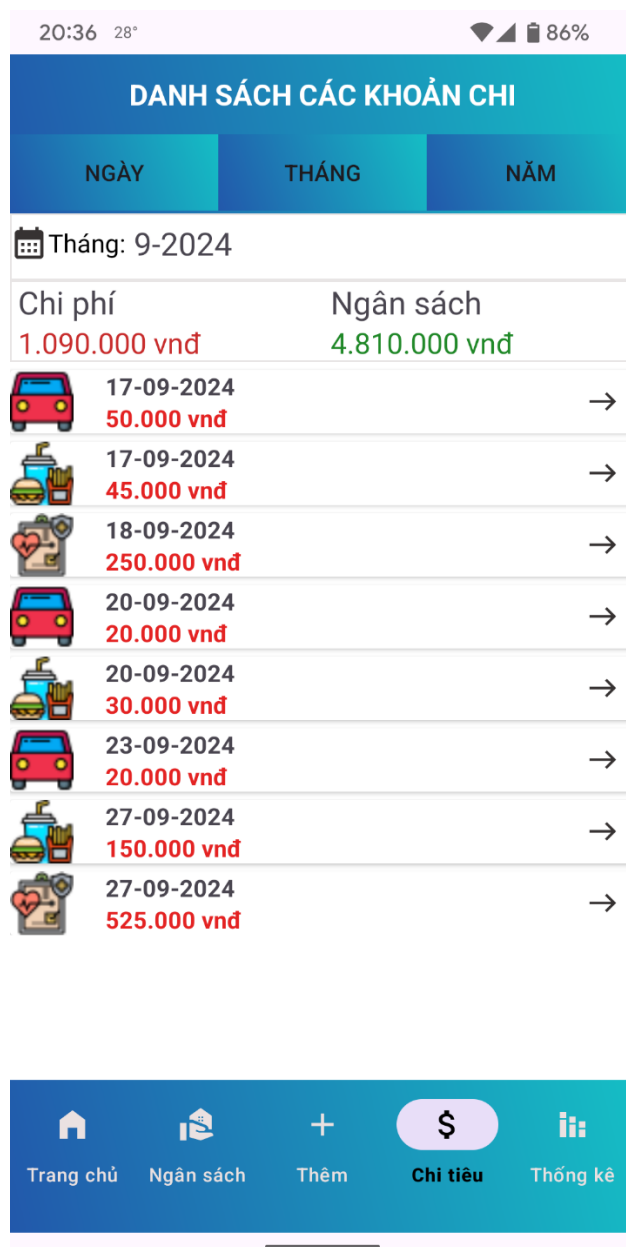
Thống kê

Hình 5.9: Giao diện chức năng sửa xóa khoản thu



- Để xem danh sách các khoản chi đã thêm người dùng chọn chức năng chi tiêu ở menu các chức năng bên dưới, giao diện danh sách các khoản chi sẽ được hiển thị bao gồm tiêu đề chọn hiển thị theo ngày, tháng hoặc năm. Người dùng có thể chọn ngày, tháng, năm để hiển thị các khoản chi cùng thời gian tương ứng. Giao diện còn hiển thị chi phí các khoản thu đang hiển thị và ngân sách còn lại của người dùng. Người dùng có thể chọn vào khoản chi ở danh sách nếu muốn chỉnh sửa hoặc xóa.

- Giao diện khi xem danh sách các khoản chi:



Hình 5.10: Giao diện chức năng chi tiêu

- Để sửa hoặc xóa khoản chi người dùng chọn một khoản chi muốn sửa hoặc xóa trong danh sách các khoản chi. Giao diện sửa hoặc xóa của khoản chi đó sẽ được hiển thị, người dùng sửa các thông tin muốn sửa sau đó ấn lưu để cập nhật khoản chi muốn sửa hoặc ấn xóa nếu như muốn xóa khoản chi đó.

- Giao diện khi sửa hoặc xóa khoản chi:

20:37 28°

86%

THÔNG TIN CHI TIẾT

Số tiền

50000 đ

? Đi lại

ĂN UỐNG

ĐI LẠI

TRANG PHỤC

SỨC KHỎE

DỊCH VỤ

BẢO HIỂM

LÀM ĐẸP

CHO VAY

☰ Đổ xăng xe

📅 Ngày: 17-09-2024

LƯU

XÓA

Trang chủ

Ngân sách

Thêm

Chi tiêu

Thống kê

Hình 5.11: Giao diện chức năng sửa xóa khoản chi

- Để xem tình hình thu và chi tiêu của mình người dùng chọn chức năng thống kê, giao diện thống kê sẽ được hiển thị. Người dùng có thể chọn thống kê theo ngày, tháng hoặc năm và có thể chọn ngày, tháng, năm muốn thống kê. Giao diện sẽ hiển thị các biểu đồ cột cho thống kê khoản thu và khoản chi và biểu đồ tròn cho thống kê danh mục chi tiêu theo thời gian tương ứng mà người dùng chọn.

- Giao diện khi thống kê:



Hình 5.12: Giao diện chức năng thống kê.

## KẾT LUẬN

- Qua quá trình học tập, nghe giảng sau đó bắt tay vào thực hiện xây dựng một sản phẩm ứng dụng số thu chi: quản lý tài chính trên nền tảng Android sử dụng Android Studio thì nhóm chúng em cũng đã hoàn thành một sản phẩm tương đối như mong đợi. Qua quá trình thực hiện các thành viên trong nhóm cũng đã hiểu tương đối thế nào là phát triển ứng dụng trên nền tảng Android. Ứng dụng của nhóm em cũng đã đáp ứng nhu cầu của một ứng dụng quản lý tài chính, quản lý chi tiêu, có thể thêm các khoản chi, các khoản thu. Ngoài ra còn có thống kê giúp người dùng có thể xem tình hình chi tiêu của bản thân một cách tổng quan và dễ dàng. Ứng dụng cũng có thiết kế các giao diện dễ nhìn dễ sử dụng, thân thiện với người dùng. Nhóm chúng em cũng xin cảm ơn thầy Trần Bá Hùng là giảng viên môn phát triển ứng dụng cho thiết bị di động, trong thời gian học tập thầy đã giảng dạy chúng em rất chi tiết và thực hành trực tiếp cho chúng em học hỏi trong các bài học. Nhóm chúng em rất mong có sự góp ý của thầy và các thầy cô đọc tài liệu này để chúng em cải thiện hơn trong tương lai.

- Mong muốn cải thiện: Xây dựng các chức năng phức tạp hơn trong chủ đề này như: có các chức năng phân tích dựa vào thói quen chi tiêu của người dùng, có thể liên kết với các ứng dụng ngân hàng, ứng dụng ví điện tử để người dùng dễ dàng quản lý nguồn thu và chi tiêu của mình hơn, thiết kế giao diện đẹp mắt và nhiều hiệu ứng đẹp nhưng vẫn đạt tiêu chí dễ sử dụng và thân thiện với người dùng hơn.

## **TÀI LIỆU THAM KHẢO**

Các tài liệu và kiến thức do giảng viên cung cấp.
---

Một số kiến thức lý thuyết có tham khảo công cụ Ai: Chat gpt, Bing Ai, Google Bard
--