

ĐẶC TẢ YÊU CẦU PHẦN MỀM

ỨNG DỤNG THEO DÕI VÀ QUẢN LÝ CHI TIÊU

CÁ NHÂN

Phiên bản: <x.x>

Mã dự án: <mã dự án>

Mã tài liệu: <Project code>_SRS_<version>.doc

Tác giả :

Tên: Hà Văn Nguyên.
Vai trò: Sinh viên thực hiện.

Tên: Nguyễn Hữu Tâm.
Vai trò: Sinh viên thực hiện.

Xem xét :

Tên: Trịnh Công Duy
Vai trò: Người hướng dẫn.

Phê duyệt :

Tên: Trịnh Công Duy
Vai trò: Người hướng dẫn.

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

MỤC LỤC

1. Giới thiệu	3
1.1. Tổng quan	3
1.2. Mục đích	3
1.3. Phạm vi	3
1.4. Thuật ngữ viết tắt	4
1.5. Tài liệu tham khảo	Lỗi! Thẻ đánh dấu không được xác định.
2. Mô tả tổng quan	4
2.1. Kiến trúc tổng thể	5
2.1.1 Sơ đồ tổng thể phần mềm	5
2.1.2 Phân tích chi tiết	5
2.2. Các chức năng của sản phẩm	6
3. Yêu cầu chức năng	7
3.1. Các tác nhân	7
3.2. Các chức năng của hệ thống	7
3.3. Biểu đồ UseCase tổng quát	7
3.4. Biểu đồ UseCase phân rã	8
3.4.1 Phân rã use case Guess	8
3.4.2 Phân rã use case User	8
3.4.3 Phân rã use case Thêm chi tiêu	9
3.4.4 Phân rã use case Xóa chi tiêu	9
3.5. Quy trình nghiệp vụ	10
3.5.1 Quy trình sử dụng tổng quát phần mềm	10
3.5.2 Quy trình sử dụng chi tiết	11
3.6. Đặc tả use case	12
3.6.1 Đăng nhập	12
3.6.2 Đăng ký	12
3.6.3 Đổi mật khẩu	13
3.6.4 Thêm mục chi tiêu	13
3.6.5 Thêm chi tiêu	13
3.7. Sơ đồ lớp	14
3.8. Biểu đồ tuần tự	15
3.8.1 Chức năng đăng nhập	15
3.8.2 Thêm mục chi tiêu	15
3.8.3 Xóa mục chi tiêu	16
3.8.4 Sửa chi tiêu	16
4. Yêu cầu phi chức năng	16
4.1. Giao diện người dùng	16
4.2. Tính bảo mật	17
4.3. Ràng buộc	17
4.4. Yêu cầu hiệu suất	17
4.5. Yêu cầu khả năng hỗ trợ & bảo trì	17
4.5.1 Quy định về khoản trống	17
4.5.2 Một vài ước hoa thường trong đặt tên	23
4.5.3 Các quy tắc đặt tên Java tiêu chuẩn	23
TÀI LIỆU THAM KHẢO	26

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

1. Giới thiệu

1.1. Tổng quan

Tài liệu này được viết dựa theo chuẩn của Tài liệu đặc tả yêu cầu phần mềm (Software Requirements Specifications - SRS) được giải thích trong "IEEE Recommended Practice for Software Requirements Specifications" và "IEEE Guide for Developing System Requirements Specifications".

Phần I - Giới Thiệu: Mục này bao gồm mục tiêu tổng quát của hệ thống, phạm vi của dự án, và các định nghĩa cơ bản. Nó cung cấp một cái nhìn bao quát và giới thiệu ngữ cảnh mà trong đó hệ thống sẽ được triển khai.

Phần 2: Mô Tả Tổng Quan: Mô tả tổng quan cung cấp cái nhìn rộng lớn về hệ thống, bao gồm: Mô tả môi trường hoạt động của hệ thống. Mô tả về các hành động của người dùng. Mô tả tổng quan về chức năng của hệ thống. Mô tả cách hệ thống phù hợp vào trong môi trường kinh doanh và nhu cầu của người dùng.

Phần 3: Đặc Tả Yêu Cầu Chức Năng: Phần này cung cấp một đặc tả chi tiết về các yêu cầu chức năng của hệ thống, bao gồm: Liệt kê chi tiết các tác vụ mà hệ thống phải thực hiện. Mô tả các trường hợp sử dụng, cả bình thường và ngoại lệ, và mô tả cách hệ thống xử lý. Phác thảo các luồng dữ liệu và thông tin chi tiết về các giao diện người dùng.

Phần 4: Yêu Cầu Phi Chức Năng: Trong phần này, sẽ được nêu rõ các yêu cầu không liên quan trực tiếp đến các chức năng cụ thể của hệ thống như: Hiệu năng và thời gian đáp ứng. Độ tin cậy, khả năng phục hồi sau sự cố và yêu cầu bảo mật. Khả năng mở rộng, bảo trì và hỗ trợ của hệ thống. Các ràng buộc thiết kế và tiêu chuẩn tuân thủ.

Ứng dụng theo dõi và quản lý chi tiêu là một ứng dụng giúp người dùng có thể thống kê và theo dõi việc chi tiêu của mình nhằm đưa ra các quyết định chi tiêu hợp lý

1.2. Mục đích

Mục đích của tài liệu đặc tả yêu cầu phần mềm này là cung cấp một cái nhìn tổng quan, dễ hiểu về các yêu cầu, thành phần của dự án Ứng dụng theo dõi và quản lý chi tiêu cá nhân.

1.3. Phạm vi

Tài liệu đặc tả yêu cầu phần mềm này được xây dựng nhằm phục vụ cho dự án Xây dựng ứng dụng theo dõi và quản lý chi tiêu cá nhân .

Việc phát triển ứng dụng này giúp cho khách hàng , cá nhân sử dụng ứng dụng dễ dàng quản lý và theo dõi chi tiêu của bản thân .Nó bao gồm các chức năng ghi chép chi tiêu hằng ngày , phân loại chi tiêu theo mục đích , báo cáo và thống kê chi tiêu , đặt

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

mục tiêu tiết kiệm , nhắc nhở và cảnh báo các khoản thanh toán tài chính sắp tới từ đó giúp cho người dùng quản lí tài chính một cách tự động.

1.4. Thuật ngữ viết tắt.

Software Requirements Specifications-SRS	Đặc tả yêu cầu phần mềm
Use Case(s)	Biểu đồ mô tả những yêu cầu của hệ thống
GUI	GUI (Graphical User Interface) là giao diện người dùng đồ họa cho phép tương tác với thiết bị thông qua biểu tượng, nút bấm, cửa sổ và menu thay vì lệnh văn bản.
BLL	BLL (Business Logic Layer) là tầng xử lý logic nghiệp vụ trong kiến trúc phần mềm. Nó chịu trách nhiệm thực hiện các quy tắc nghiệp vụ, kiểm tra và xử lý dữ liệu trước khi tương tác với cơ sở dữ liệu. BLL giúp tách biệt logic nghiệp vụ khỏi giao diện người dùng và tầng dữ liệu, làm cho ứng dụng dễ bảo trì và mở rộng.
DAL	DAL (Data Access Layer) là tầng truy cập dữ liệu trong kiến trúc phần mềm. Nó quản lý việc kết nối, truy vấn, và thao tác với cơ sở dữ liệu, tách biệt logic truy cập dữ liệu khỏi các phần khác của ứng dụng, giúp mã nguồn dễ bảo trì và bảo mật hơn.
Hypertext Markup Language (HTML)	Ngôn ngữ đánh dấu siêu văn bản
DTO	DTO (Data Transfer Object) là một đối tượng dùng để truyền dữ liệu giữa các tầng của ứng dụng, đặc biệt giữa tầng giao diện người dùng và tầng logic nghiệp vụ. DTO chỉ chứa dữ liệu, không chứa logic nghiệp vụ, giúp giảm số lượng lời gọi phương thức và tối ưu hóa hiệu suất khi truyền dữ liệu qua mạng hoặc giữa các thành phần của hệ thống.

Bảng 1-1: Từ điển thuật ngữ

2. Mô tả tổng quan

Ứng dụng theo dõi và quản lý chi tiêu cá nhân giúp cho khách hàng , cá nhân sử dụng

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

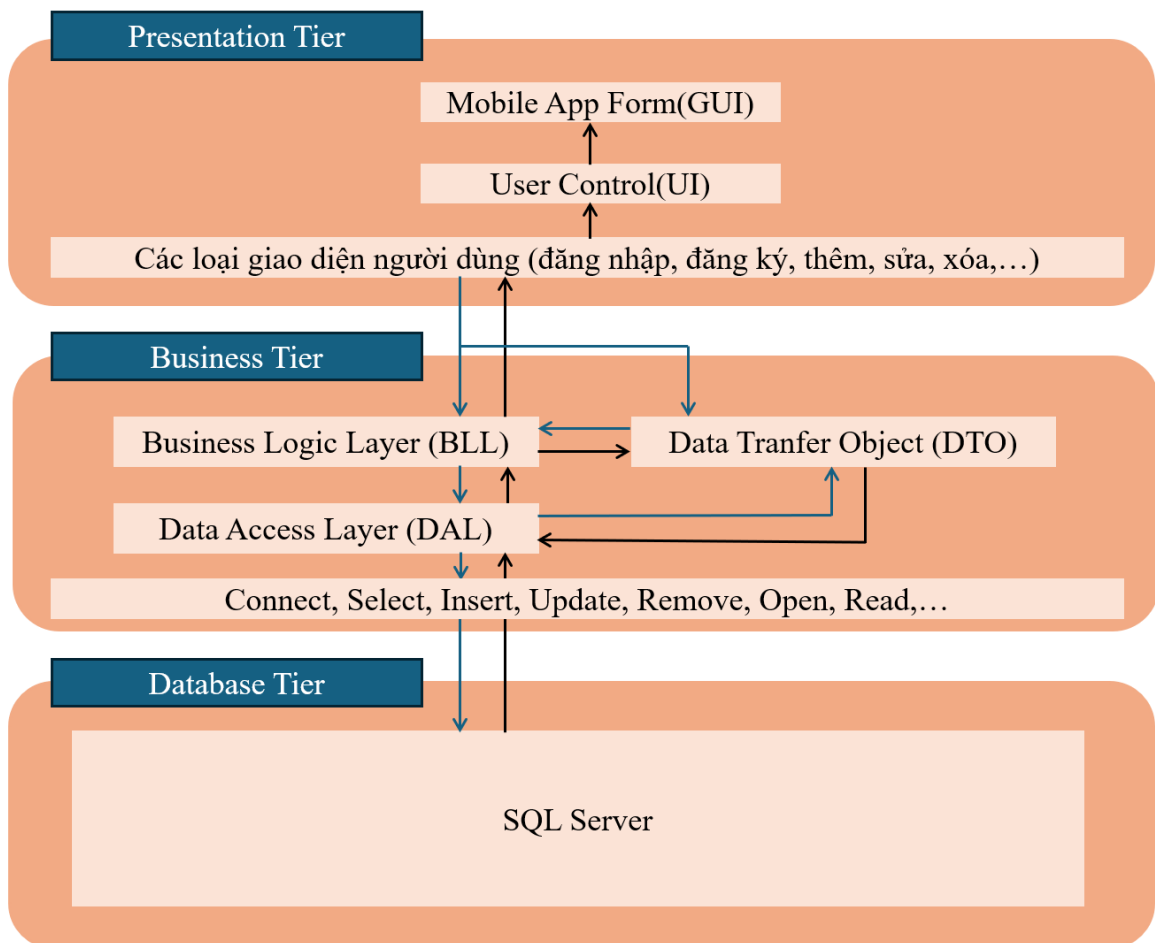
ứng dụng để dàng quản lí và theo dõi chi tiêu của bản thân .

Ứng dụng theo dõi và quản lý chi tiêu cá nhân phù hợp với bất kỳ ai muốn có sự kiểm soát tốt hơn về tài chính cá nhân và xây dựng các thói quen quản lý tài chính tích cực.

Nó bao gồm các chức năng ghi chép chi tiêu hằng ngày , phân loại chi tiêu theo mục đích , báo cáo và thống kê chi tiêu , đặt mục tiêu tiết kiệm , nhắc nhở và cảnh báo các khoản thanh toán tài chính sắp tới từ đó giúp cho người dùng quản lí tài chính một cách tự động.

2.1. Kiến trúc tổng thể

2.1.1 Sơ đồ tổng thể phần mềm



Kiến trúc tổng thể phần mềm là kiến trúc 3 tầng, các tầng có liên quan và gắn kết chặt chẽ với nhau, dữ liệu được truyền từ trên xuống và dưới lên theo sơ đồ trên.

2.1.2 Phân tích chi tiết

Kiến trúc tổng thể của phần mềm là kiến trúc 3 tầng:

- Presentation Tier (Tầng giao diện): là tầng giao tiếp trực tiếp với người dùng, hiển thị các giao diện của chương trình để người dùng thao tác và điều khiển. Ở đây

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

gồm các giao diện như đăng nhập, đăng ký,... Dữ liệu được người dùng nhập vào được gửi xuống các tầng phía dưới để xử lý

- Business Tier (Tầng nghiệp vụ): là tầng xử lý các vấn đề logic, thuật toán từ dữ liệu người dùng đưa vào và gửi kết quả lên tầng phía trên để hiển thị cho người dùng. Thêm vào đó, tầng này giao tiếp trực tiếp với cơ sở dữ liệu nhằm truy suất, cập nhật dữ liệu từ cơ sở dữ liệu kết hợp với dữ liệu từ phía người dùng. Bao gồm thuật toán và các câu lệnh giao tiếp với cơ sở dữ liệu
- Database Tier (Tầng cơ sở dữ liệu): là tầng lưu trữ dữ liệu

2.2. Các chức năng của sản phẩm

1. Đăng nhập và đăng ký: chức năng này nhằm mục đích để người dùng có thể truy cập vào ứng dụng
2. Thêm mục chi tiêu: chức năng này nhằm mục đích để người dùng tạo và quản lý chi tiêu của mình
3. Quản lý chi tiêu: chức năng này bao gồm các chức năng con như thêm, sửa, xóa, xem chi tiêu
4. Thống kê: chức năng giúp người dùng thấy được chi tiêu của bản thân dưới dạng một biểu đồ thống kê

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3. Yêu cầu chức năng

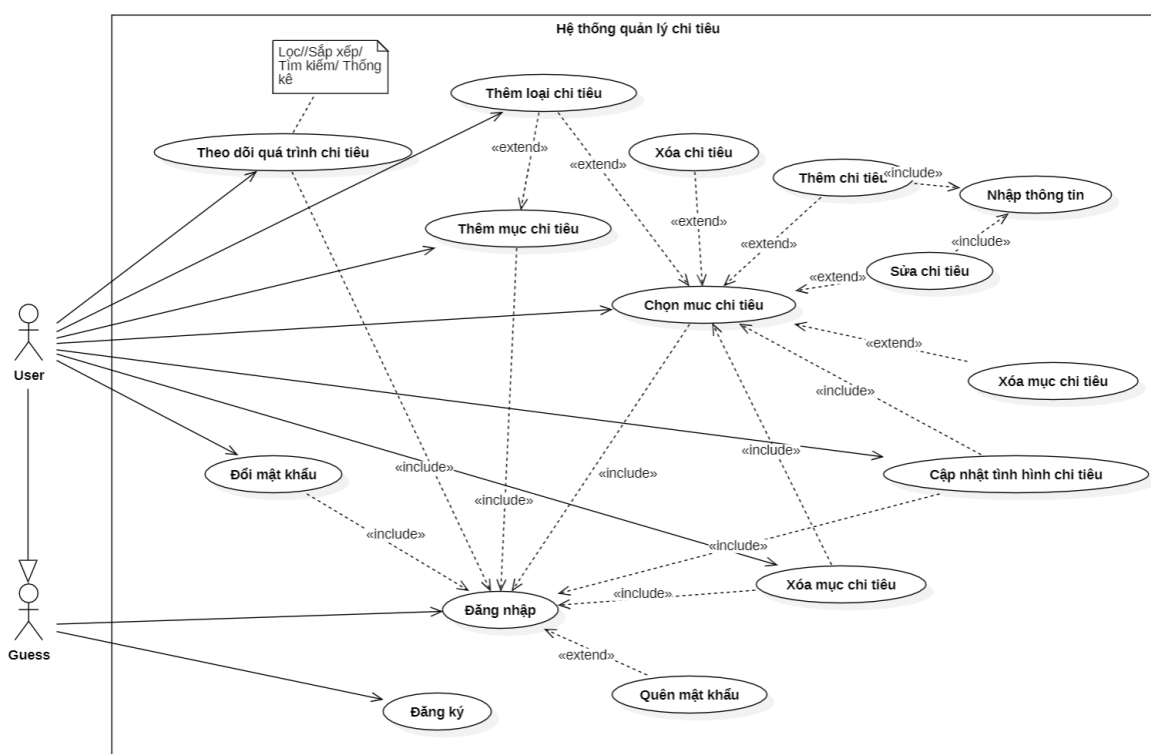
3.1. Các tác nhân

Hệ thống chỉ bao gồm 2 tác nhân là Khách và Người dùng. Khách có vai trò là đăng nhập vào hệ thống khi đã có tài khoản, nếu không thì có thể đăng ký tài khoản mới. Khách đăng nhập vào hệ thống và hệ thống truy suất dữ liệu từ Database để người dùng sử dụng.

3.2. Các chức năng của hệ thống

1. Đăng nhập: chức năng này nhằm mục đích để hệ thống truy xuất đúng dữ liệu của người dùng đó.
2. Đăng ký: để người dùng mới có thể truy cập vào hệ thống khi chưa có tài khoản hoặc cần tạo một tài khoản mới.
3. Thêm mục chi tiêu: chức năng dùng để thêm mục chi tiêu để lưu trữ các chi tiêu, người dùng tạo một hoặc nhiều hơn các mục chi tiêu khác nhau để quản lý các việc chi tiêu có tính chất khác nhau. (Ví dụ, một người có 2 loại chi tiêu là chi tiêu cho công việc và chi tiêu cá nhân)
4. Theo dõi quá trình chi tiêu: gồm các chức năng như lọc, sắp xếp, tìm kiếm, thống kê chi tiêu.
5. Cập nhật tình hình chi tiêu: chức năng thực hiện việc thêm, sửa, xóa chi tiêu trong một mục chi tiêu.
6. Thêm loại chi tiêu: là chức năng giúp người dùng thêm một loại chi tiêu cụ thể. (ví dụ chi tiêu cho việc ăn uống, việc đi lại,...)

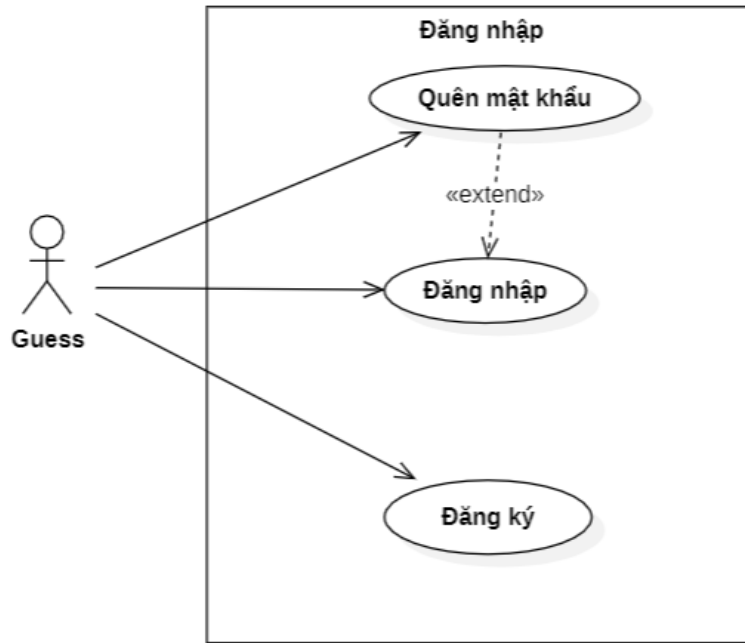
3.3. Biểu đồ UseCase tổng quát



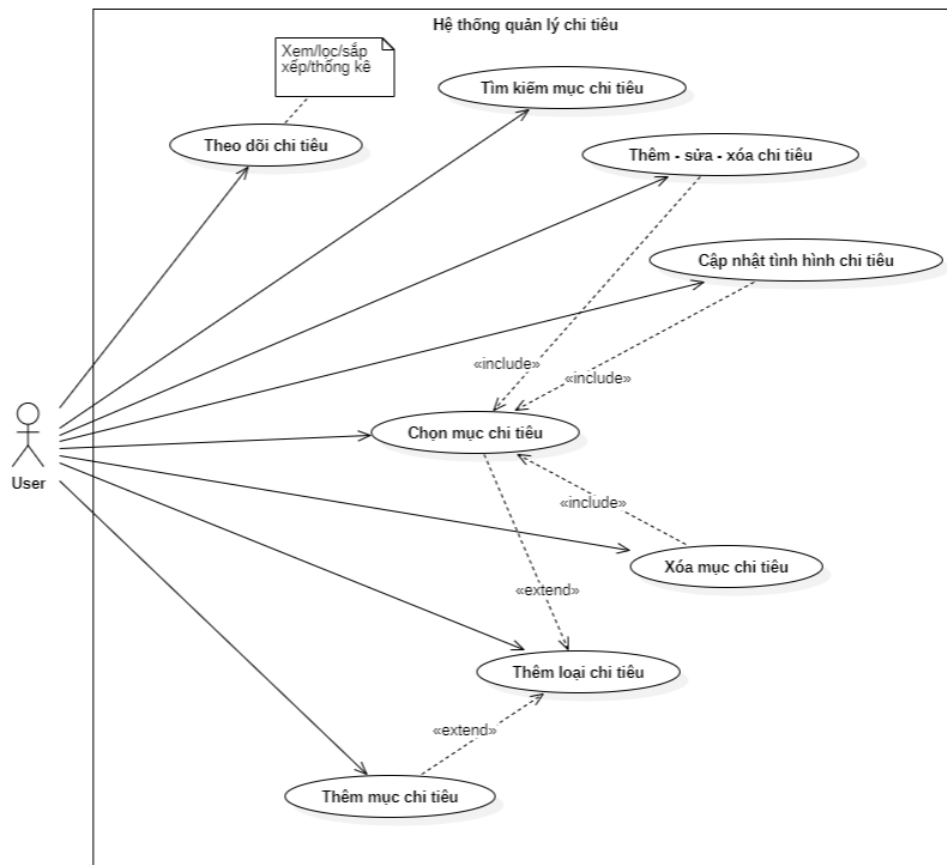
<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.4. Biểu đồ UseCase phân rã

3.4.1 Phân rã use case Guess

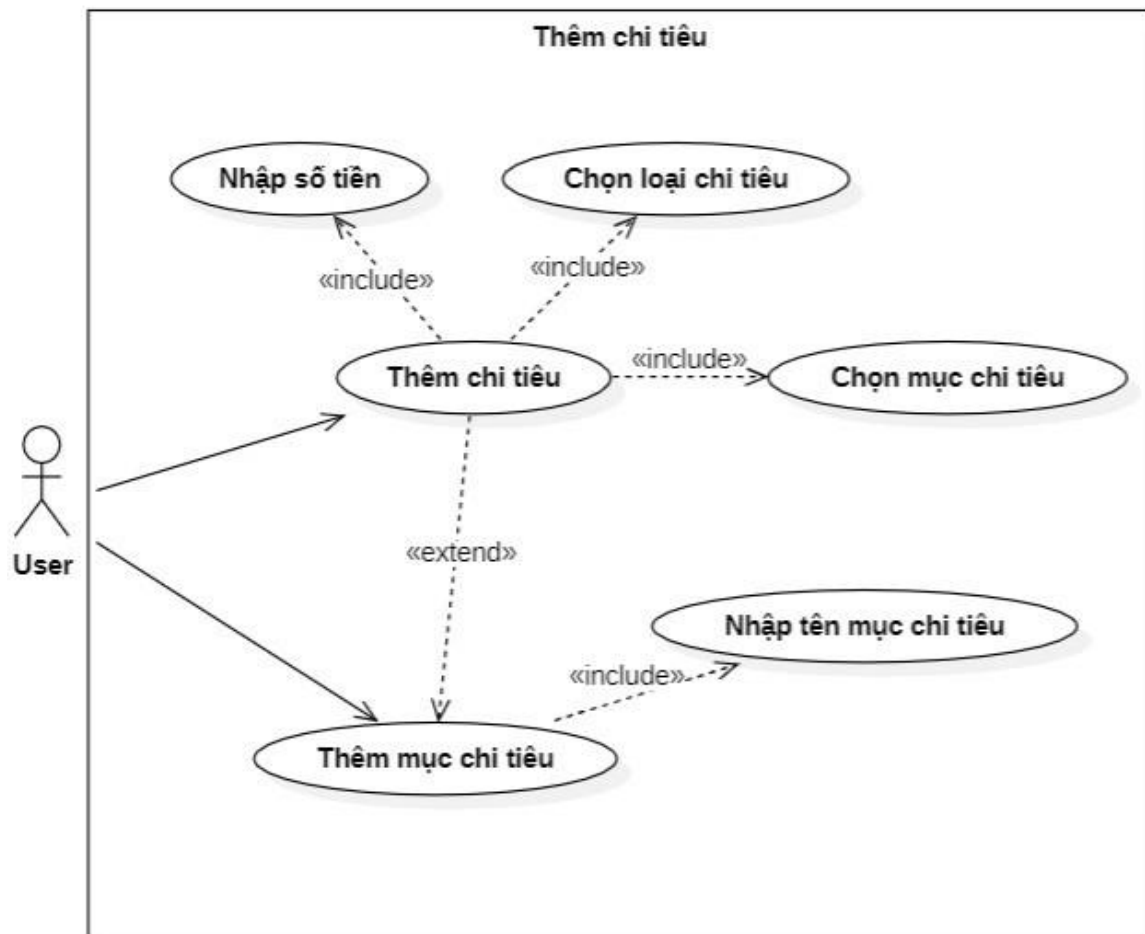


3.4.2 Phân rã use case User

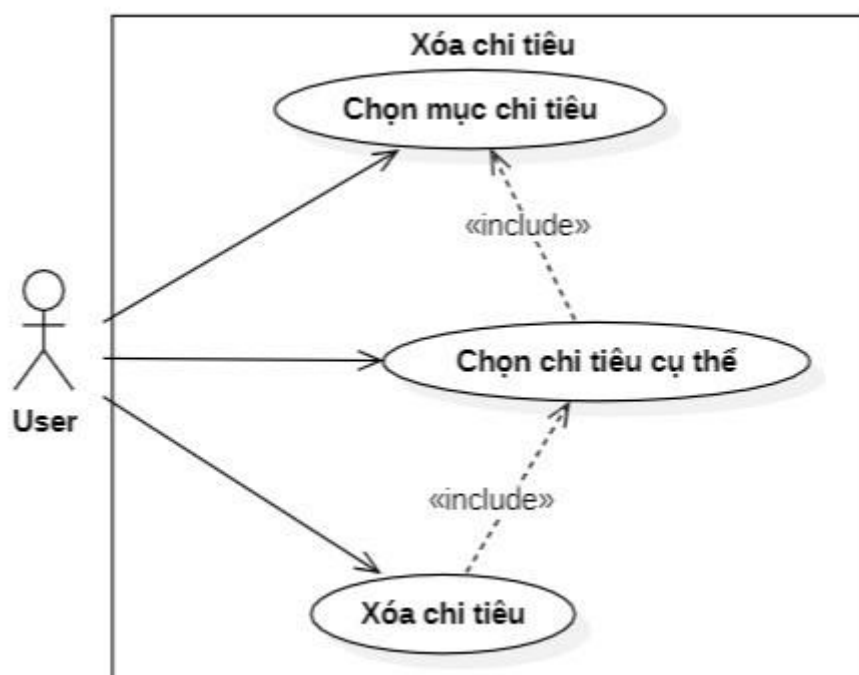


<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.4.3 Phân rã use case Thêm chi tiêu



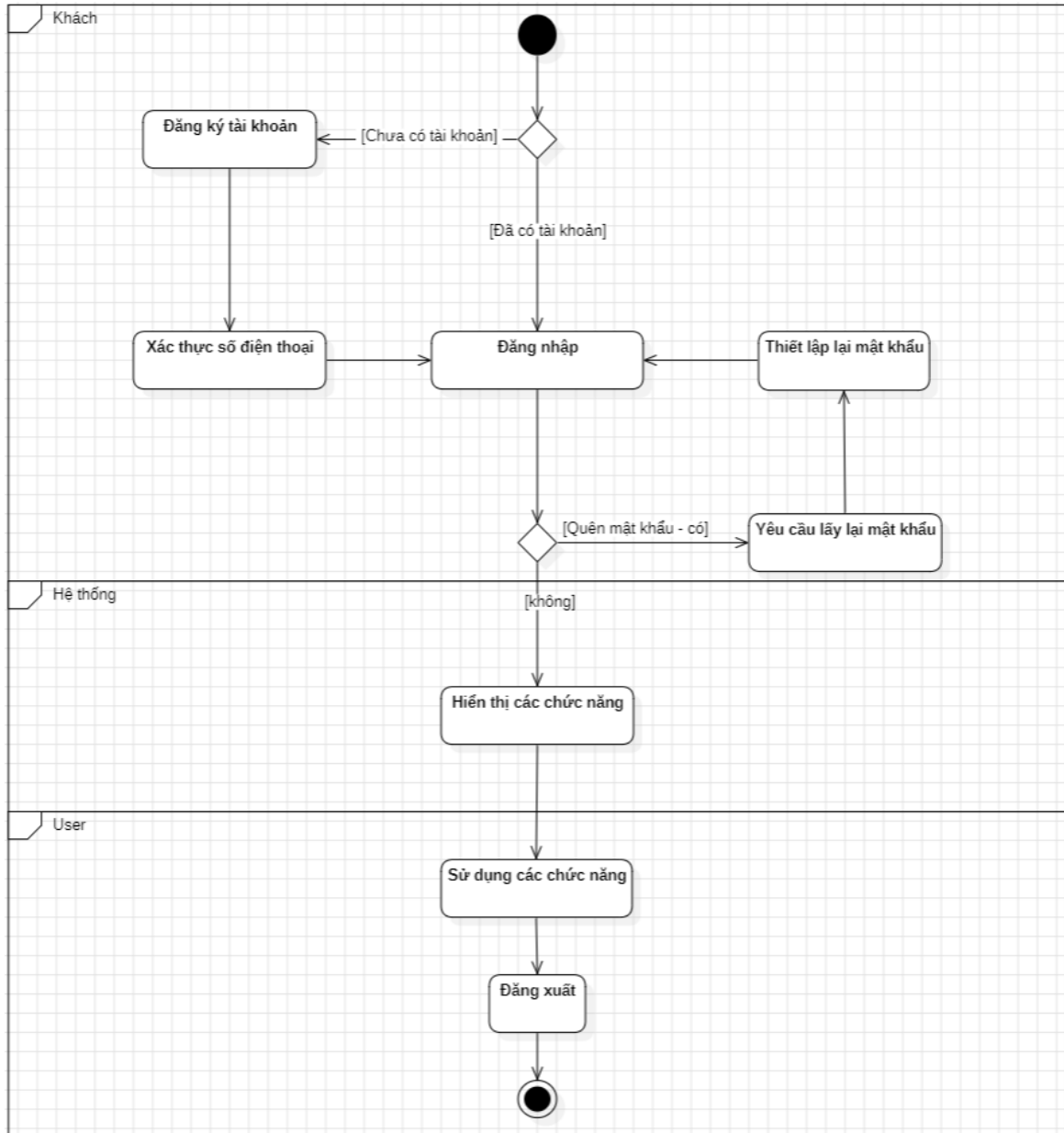
3.4.4 Phân rã use case Xóa chi tiêu



<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

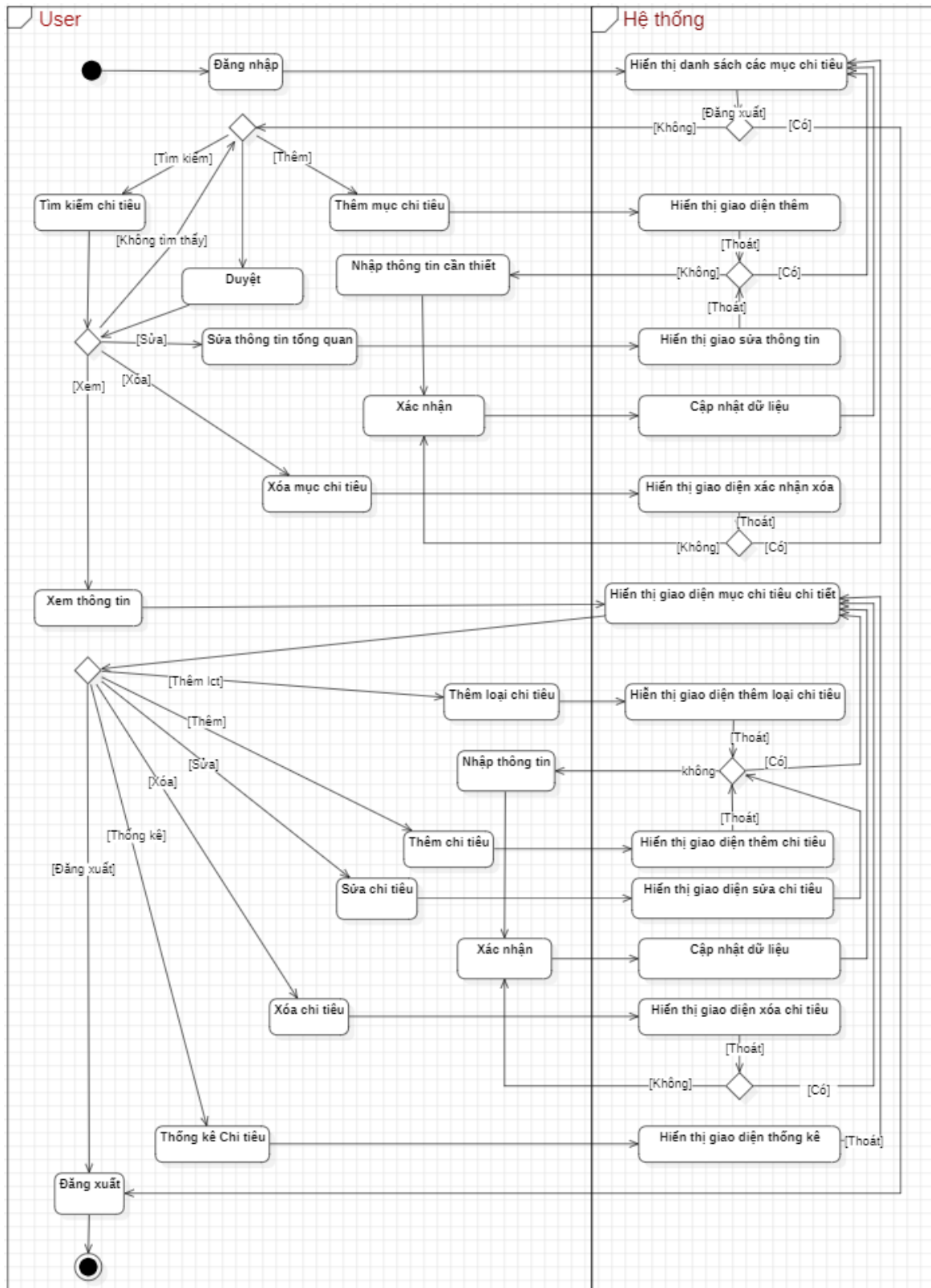
3.5. Quy trình nghiệp vụ

3.5.1 Quy trình sử dụng tổng quát phần mềm



<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.5.2 Quy trình sử dụng chi tiết



<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.6. Đặc tả use case

3.6.1 Đăng nhập

Tên use case	Đăng nhập		
Tác nhân	Khách		
Mô tả	Tác nhân đăng nhập vào hệ thống để sử dụng các chức năng của hệ thống		
Sự kiện kích hoạt	Khi click vào nút đăng nhập trên dao diện		
Tiền điều kiện	Tác nhân đã có tài khoản trên hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Khách	Chọn chức năng đăng nhập
	2	Hệ Thống	Hiển thị giao diện đăng nhập
	3	Khách	Nhập tên đăng nhập và mật khẩu
	4	Khách	Yêu cầu đăng nhập bằng cách click nút đăng nhập trên màn hình
	5	Hệ Thống	Kiểm tra xem khách đã nhập đầy đủ thông tin hay chưa
	6	Hệ Thống	Kiểm tra xem tên đăng nhập và mật khẩu của khách đã nhập có và đúng trong csdl hay không
	7	Hệ Thống	Hiển thị giao diện, chức năng tương ứng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ Thống	Thông báo lỗi: Yêu cầu khách nhập đầy đủ thông tin để có thể tiếp tục nếu khách nhập bị thiếu
	6a	Hệ Thống	Thông báo lỗi: Nếu tên đăng nhập/mật khẩu không chính xác
Hậu điều kiện	Tác nhân đăng nhập được vào hệ thống		

3.6.2 Đăng ký

Tên use case	Đăng ký		
Tác nhân	Khách		
Mô tả	Tác nhân đăng ký tài khoản để sử dụng chức năng của hệ thống		
Sự kiện kích hoạt	Khi click vào nút đăng ký trên dao diện		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Khách	Chọn chức năng đăng ký
	2	Hệ Thống	Hiển thị giao diện đăng ký
	3	Khách	Nhập các thông tin yêu cầu
	4	Khách	Yêu cầu đăng ký bằng cách click nút đăng ký
	5	Hệ Thống	Kiểm tra xem khách đã nhập đầy đủ thông tin hay chưa
	6	Hệ Thống	Kiểm tra tên đăng nhập đã tồn tại hay chưa
	7	Hệ Thống	Kiểm tra mật khẩu nhập lại và mật khẩu có trùng nhau hay không
	8	Hệ Thống	Lưu thông tin tài khoản và thông báo đăng ký thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ Thống	Thông báo lỗi: Yêu cầu nhập đầy đủ thông tin
	6a	Hệ Thống	Thông báo lỗi: Tên tài khoản đã tồn tại
	7a	Hệ Thống	Thông báo lỗi: Mật khẩu không trùng khớp
Hậu điều kiện	Tài khoản được tạo và lưu trữ vào hệ thống		

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.6.3 Đổi mật khẩu

Tên use case	Đăng ký		
Tác nhân	Người dùng		
Mô tả	Tác nhân muốn thiết lập lại mật khẩu		
Sự kiện kích hoạt	Tác nhân click vào nút thay đổi mật khẩu trong phần thông tin cá nhân		
Tiền điều kiện	Tài khoản đã được đăng nhập		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người Dùng	Chọn chức năng thay đổi mật khẩu
	2	Hệ Thống	Hiện thị giao diện thay đổi mật khẩu
	3	Người Dùng	Nhập các thông tin yêu cầu
	4	Người Dùng	Yêu cầu thay đổi mật khẩu bằng cách click nút thay đổi mật khẩu
	5	Hệ Thống	Kiểm tra xem người dùng đã nhập đầy đủ thông tin hay chưa
	6	Hệ Thống	Kiểm tra mật khẩu đã trùng nhau hay không
	7	Hệ Thống	Gửi mã OTP về số điện thoại đã đăng ký
	8	Người Dùng	Nhập mã OTP
	9	Hệ Thống	Kiểm tra xem mã OTP có đúng hay không
	10	Hệ Thống	Lưu mật khẩu đã thay đổi vào csdl
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ Thống	Thông báo lỗi: Yêu cầu nhập đầy đủ thông tin
	6a	Hệ Thống	Thông báo lỗi: Mật khẩu không trùng khớp
	9a	Hệ Thống	Thông báo lỗi: Mã OTP không trùng khớp, yêu cầu nhập lại hoặc gửi lại mã khác
Hậu điều kiện	Mật khẩu của người dùng được thay đổi		

3.6.4 Thêm mục chi tiêu

Tên use case	Đăng ký		
Tác nhân	Người dùng		
Mô tả	Tác nhân thêm mục chi tiêu để quản lý chi ti		
Sự kiện kích hoạt	Tác nhân click vào nút thêm mục chi tiêu ở màn hình chính		
Tiền điều kiện	Tài khoản đã được đăng nhập		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người Dùng	Chọn chức năng thêm mục chi tiêu
	2	Hệ Thống	Hiện thị giao diện thêm mục chi tiêu
	3	Người Dùng	Nhập các thông tin yêu cầu
	4	Hệ Thống	Kiểm tra xem người dùng đã nhập đầy đủ thông tin hay chưa
	5	Hệ Thống	Thêm mục chi tiêu
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ Thống	Thông báo lỗi: Yêu cầu nhập đầy đủ thông tin
Hậu điều kiện	Thêm mục chi tiêu		

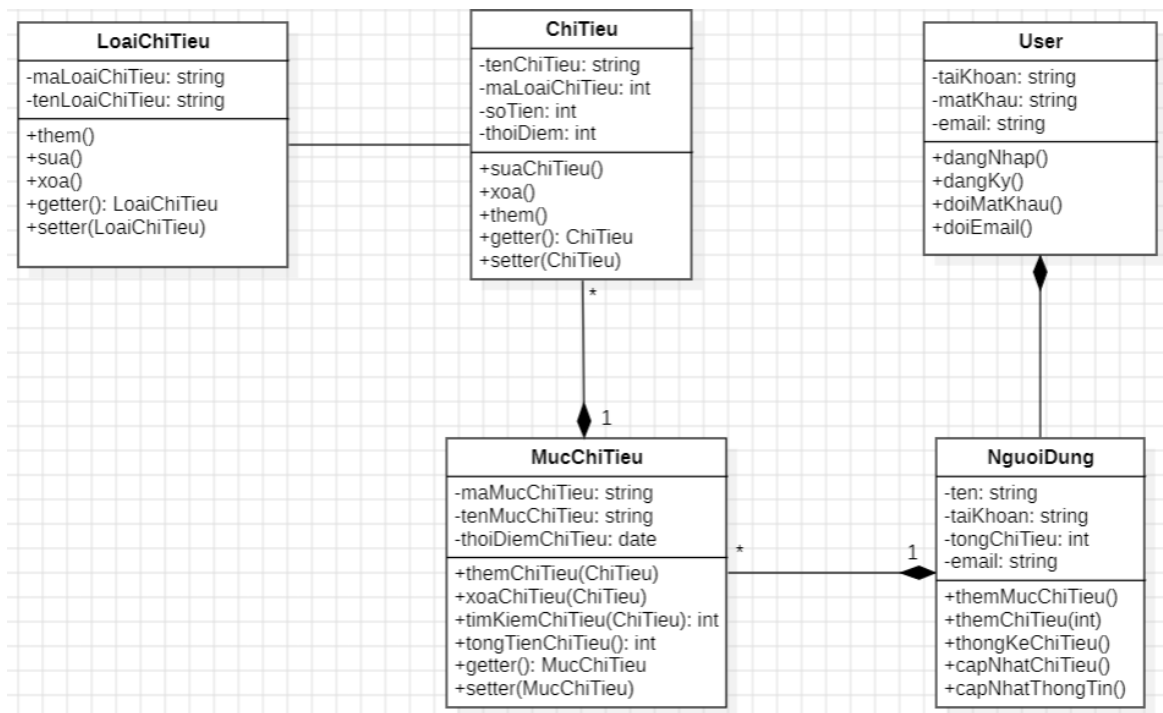
3.6.5 Thêm chi tiêu

Tên use case	Đăng ký		
Tác nhân	Người dùng		
Mô tả	Tác nhân thêm chi tiêu vào mục chi tiêu		
Sự kiện kích hoạt	Tác nhân click vào nút thêm chi tiêu của một mục chi tiêu nào đó trong các mục chi tiêu của mình		
Tiền điều kiện	Mục chi tiêu mà chi tiêu thêm vào đã tồn tại		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

	1	Người Dùng	Chọn chức năng thêm chi tiêu
	2	Hệ Thống	Hiển thị giao diện thêm chi tiêu
	3	Người Dùng	Nhập các thông tin yêu cầu
	4	Người Dùng	Chọn loại chi tiêu cho chi tiêu đó
	5	Hệ Thống	Kiểm tra xem người dùng đã nhập đầy đủ thông tin hay chưa
	6	Hệ Thống	Thêm chi tiêu vào mục chi tiêu và csdl
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ Thống	Thông báo lỗi: Yêu cầu nhập đầy đủ thông tin
Hậu điều kiện	Thêm chi tiêu		

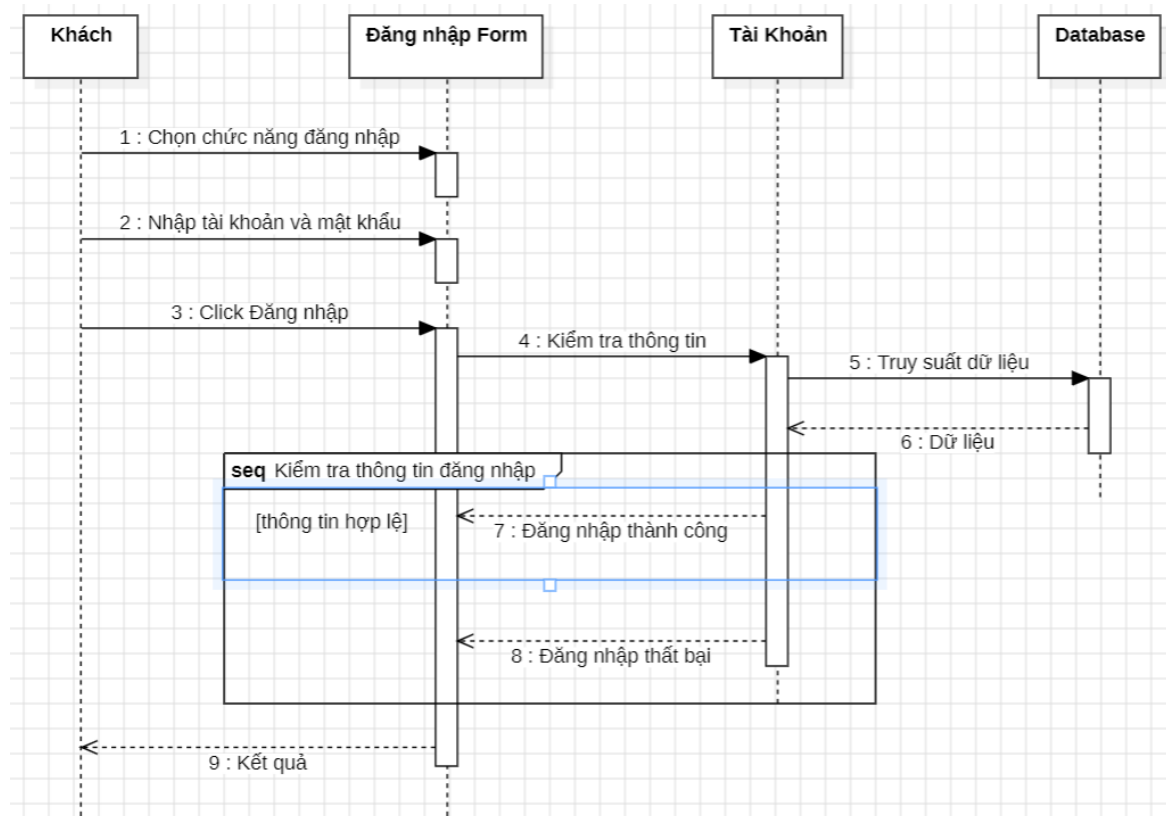
3.7. Sơ đồ lớp



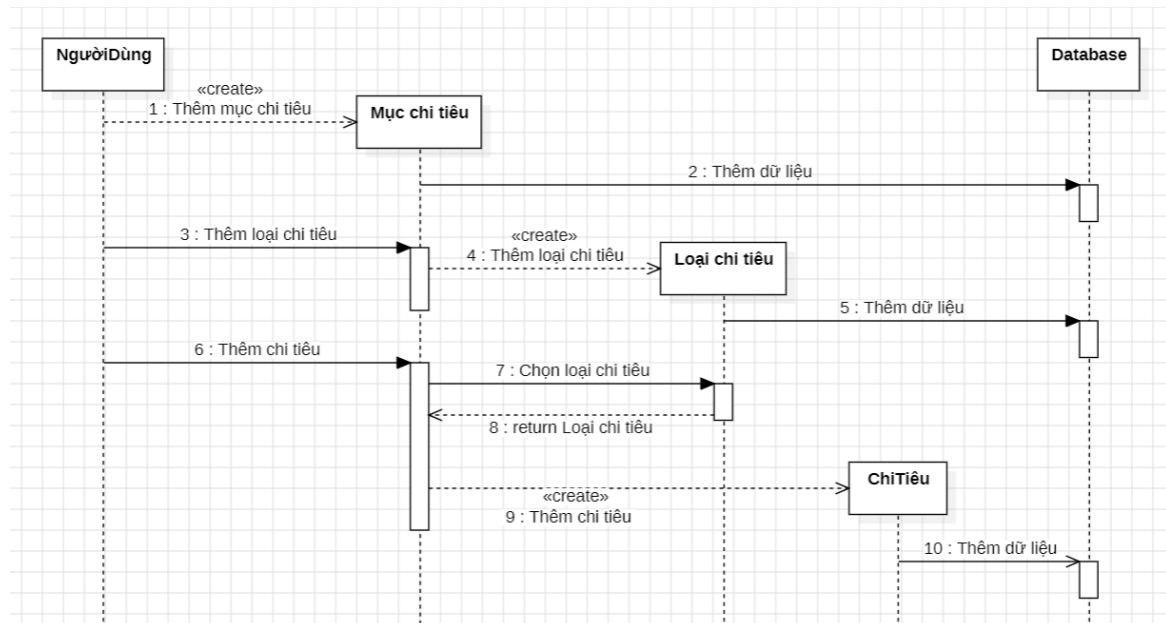
<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.8. Biểu đồ tuần tự

3.8.1 Chức năng đăng nhập

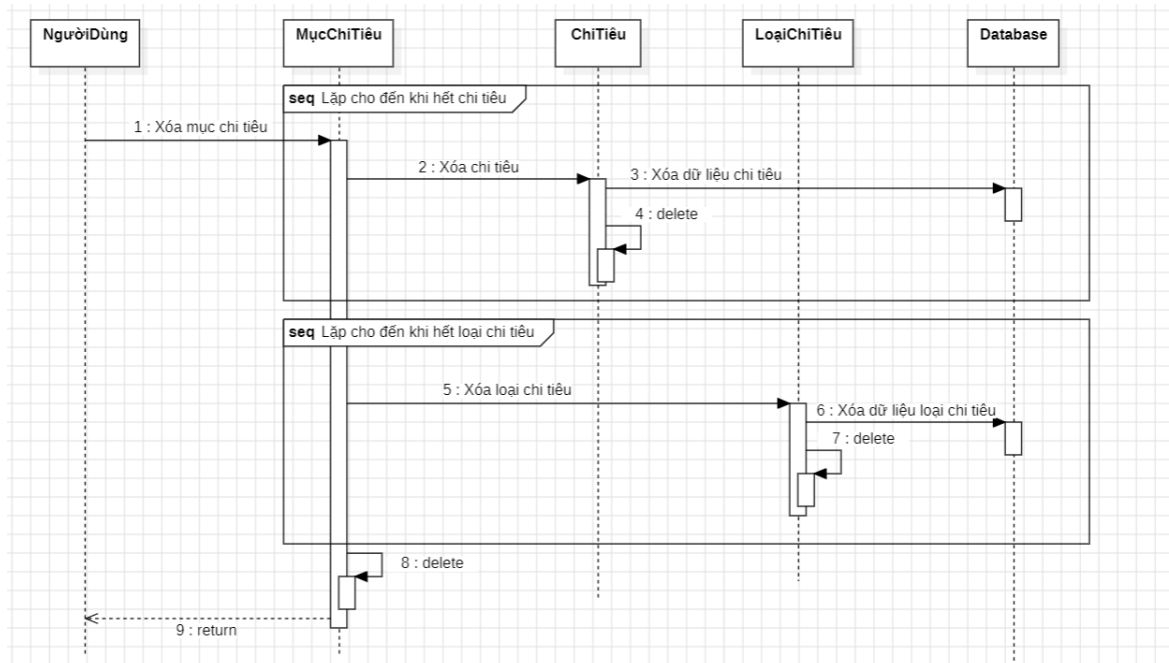


3.8.2 Thêm mục chi tiêu

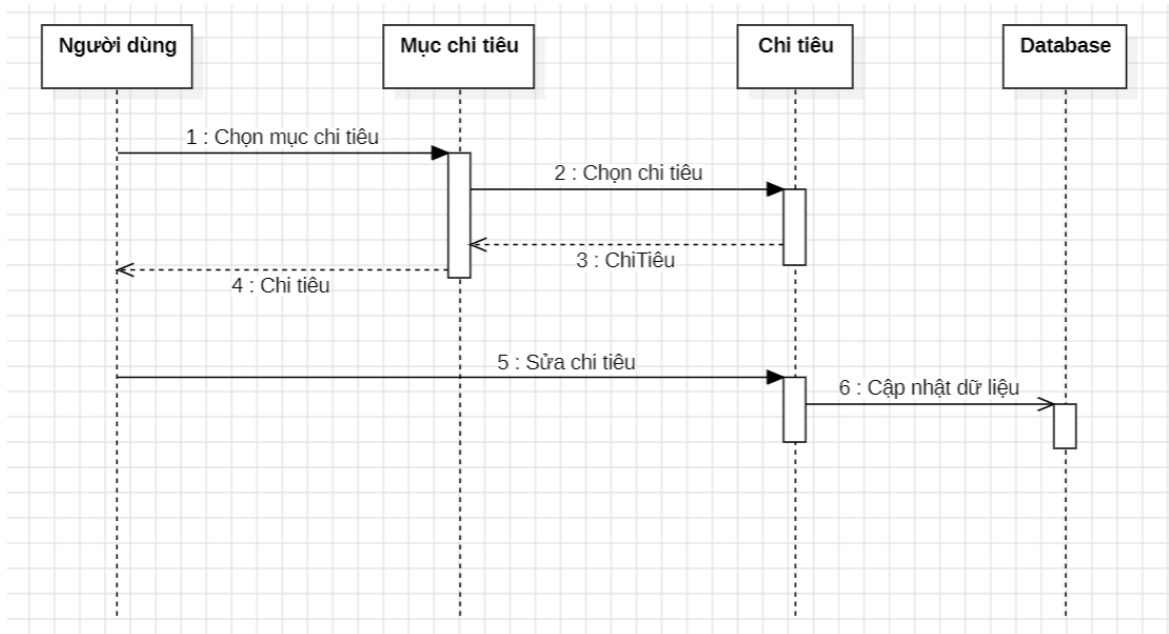


<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

3.8.3 Xóa mục chi tiêu



3.8.4 Sửa chi tiêu



4. Yêu cầu phi chức năng

4.1. Giao diện người dùng

Giao diện hiển thị tốt trên nền tảng di động.

Vì dễ tiện dụng và có thể thao tác với hệ thống mọi lúc thì khuyến khích sử dụng hệ thống trên nền tảng di động.

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

4.2. Tính bảo mật

Người dùng cần có tài khoản để đăng nhập mới có được dữ liệu và sử dụng các chức năng của hệ thống. Người dùng có thể chọn chức năng bảo mật 2 lớp để tăng tính bảo mật của hệ thống.

4.3. Ràng buộc

Hệ thống đơn người dùng và có thể truy cập được khi không có internet và dữ liệu sẽ được lưu trên thiết bị của người dùng nên sẽ không có server. Do đó, hầu như hệ thống không có bất kì ràng buộc nào.

4.4. Yêu cầu hiệu suất

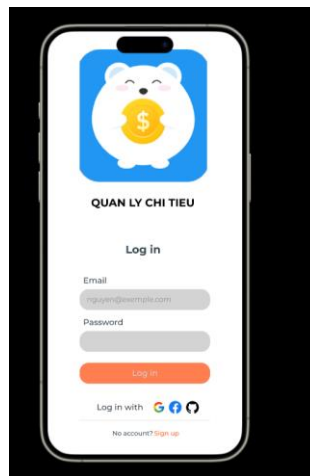
Vì hệ thống đơn người dùng nên độ lớn của dữ liệu chỉ đến từ dữ liệu của người dùng đó, nên sẽ không có quá nhiều dữ liệu được tạo ra. Vì vậy hiệu suất của hệ thống luôn ở mức cao và có thể đáp ứng được nhu cầu thời gian của người dùng.

4.5. Yêu cầu khả năng hỗ trợ & bảo trì

Hệ thống được viết bằng ngôn ngữ lập trình Java.

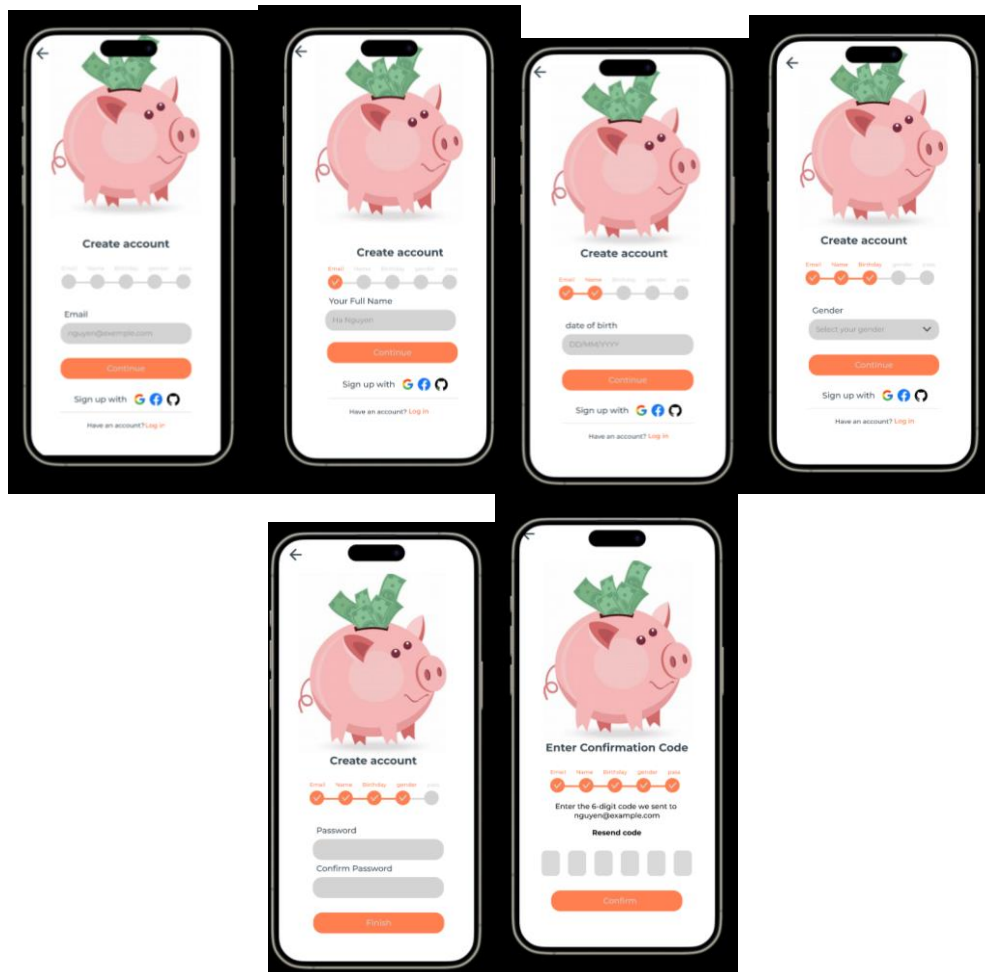
4.5.1 Giao diện .

Xây dựng giao diện demo bằng Figma .



- Màn hình đăng nhập , nhập email , mật khẩu để đăng nhập vào hệ thống . Nếu chưa có tài khoản thì hãy tạo tài khoản.

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

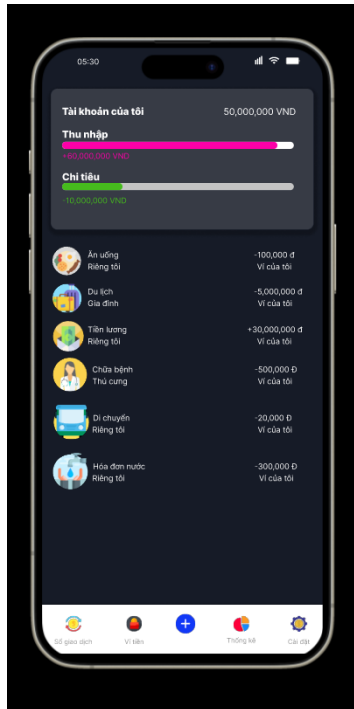


- Nhập các thông tin cá nhân để tạo tài khoản mới .
- Sau đó sẽ chuyển đến màn hình trang chủ .



<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

- Màn hình trang chủ bao gồm các thông tin như các chi tiêu và thu nhập hằng ngày, biểu đồ thống kê chi tiêu, thông tin cơ bản của người dùng.

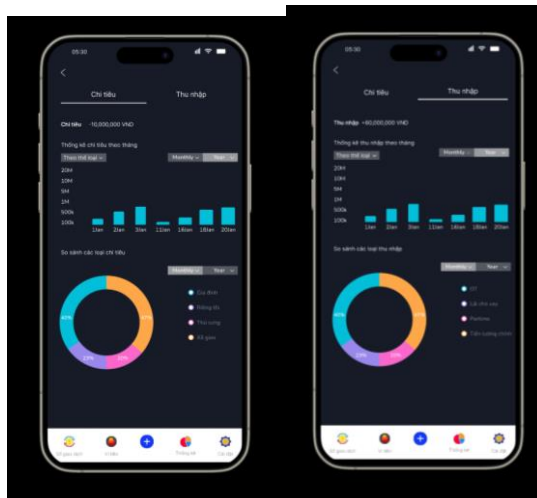


- Màn hình Ví tiền sẽ chứa các thông tin về Tổng thu nhập và Tổng chi tiêu cũng như thông tin về số dư còn lại trong ví.

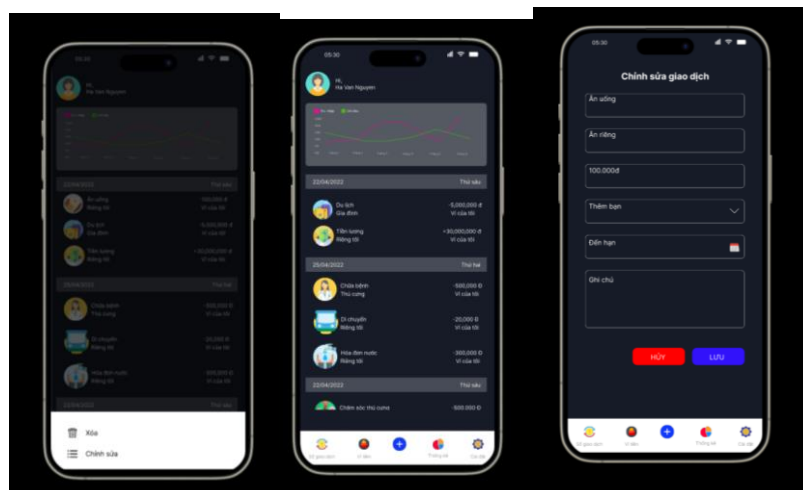
The screenshot shows a 'Giao dịch mới' (New Transaction) form. It includes input fields for 'Chọn nhóm giao dịch', 'Tên giao dịch', 'Số tiền giao dịch', 'Thêm bạn', 'Đến hạn', and a 'Ghi chú' (Note) area. At the bottom, there are 'HỦY' (Cancel) and 'LƯU' (Save) buttons. The bottom navigation bar is identical to the previous screenshot.

- Màn hình thêm chi tiêu : nhập các thông tin cần thiết để thêm một chi tiêu.

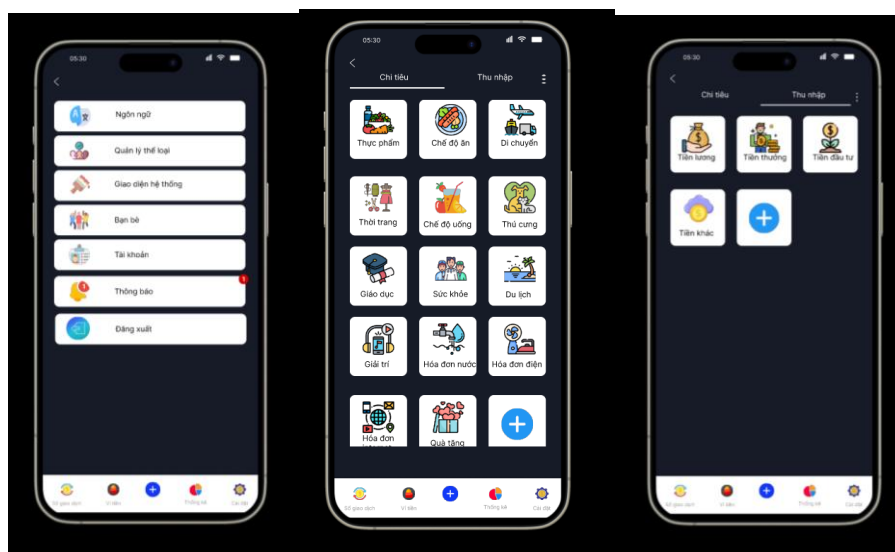
<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM



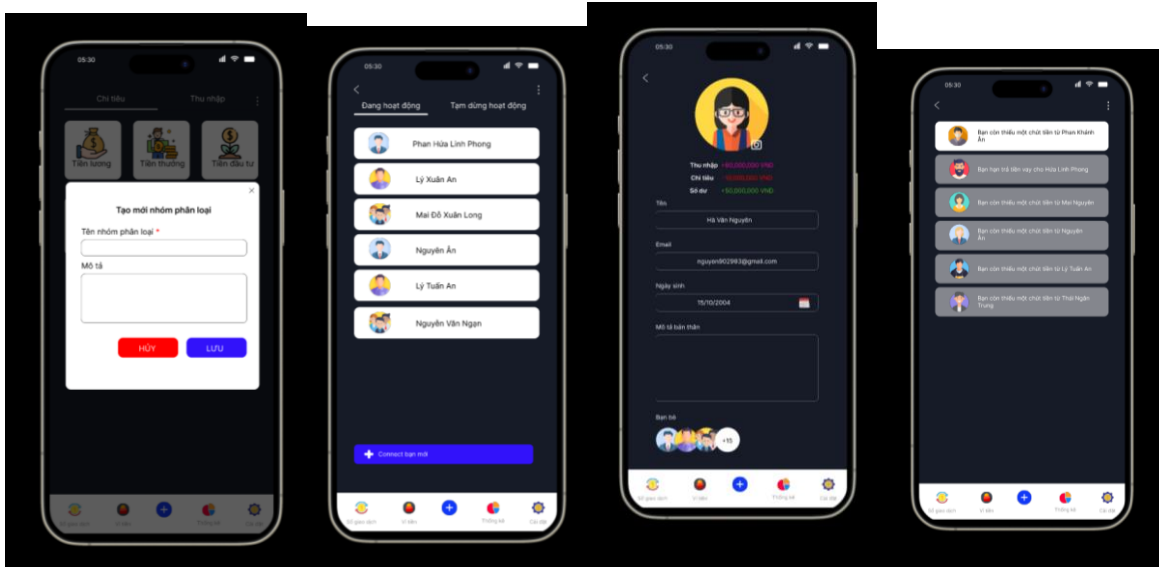
- Màn hình thống kê chi tiêu và thu nhập.



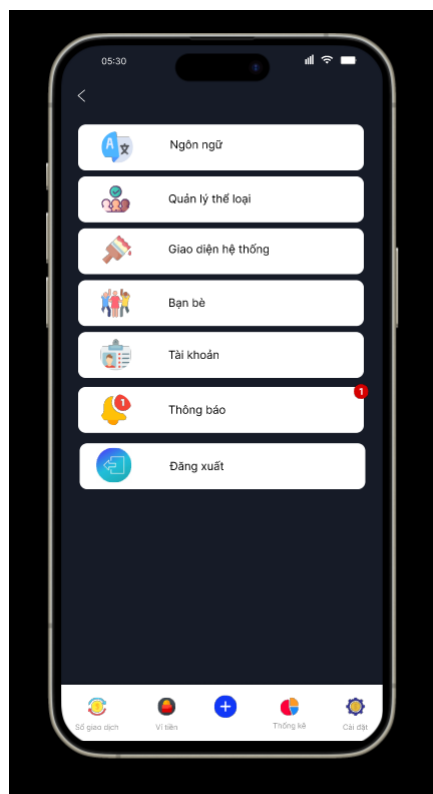
- Xóa và chỉnh sửa chi tiêu.



<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM



- Các cài đặt về Loại chi tiêu và thông tin cá nhân.



- Cuối cùng nhấn nút Đăng xuất để đăng xuất tài khoản .

4.5.2 Quy định về khoản trắng

- Thụt đầu dòng .

Xác định một chuẩn thụt đầu dòng cho toàn bộ mã nguồn của chương trình.

<MÃ DỤ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

- 1 đơn vị thụt đầu dòng = 1 tab(*)
- Hoặc, 1 đơn vị thụt đầu dòng = 5 khoảng trắng

Dòng code thứ 20 dùng 2 đơn vị thụt đầu dòng nghĩa là bấm tab 2 lần(*)

Nên dùng tab thay cho khoảng trắng

- Đỡ tốn công nhập quá nhiều lần khoảng trắng
- Có thể tùy chỉnh một đơn vị tab ứng với bao nhiêu khoảng trắng tùy ý.

Hai dòng code cách nhau một bậc thì sẽ cách nhau một đơn vị thụt đầu dòng.

```
1 package com.gpcoder;
2
3 public class SortAlgorithm {
4
5     private static final int SORT_MIN = 1;
6
7     public void InsertionSort(int[] data, int firstElement, int lastElement) {
8
9         int lowerBoundary = data[firstElement - 1];
10        data[firstElement - 1] = SORT_MIN;
11
12        for (int sortBoundary = firstElement + 1; sortBoundary <= lastElement; sortBoundary++)
13
14            int insertVal = data[sortBoundary];
15            int insertPos = sortBoundary;
16
17            while (insertVal < data[insertPos - 1]) {
18                data[insertPos] = data[insertPos - 1];
19                insertPos = insertPos - 1;
20            }
21
22            data[insertPos] = insertVal;
23
24        }
25
26        data[firstElement - 1] = lowerBoundary;
27
28    }
29
30 }
```

- **Dòng trống .**
 - Những dòng code có quan hệ với nhau(cùng thực hiện chung một công việc) thì gom thành 1 block . Nghĩa là không có dòng trống giữa các đoạn code như trên.
 - Hai block code thì cách nhau ít nhất một dòng trống.
 - Đặt khoảng trắng sau dấu phẩy và dấu chấm phẩy.
 - Đặt khoảng trắng xung quanh các toán tử.
- **Ngoặc tròn.**
 - Dùng dấu ngoặc tròn để:
 - Người đọc hiểu rõ mục đích của bạn.
 - Chắc chắn là trình biên dịch sẽ thực hiện đúng theo ý của bạn.

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

- Hãy quyết định dùng dấu ngoặc tròn khi bạn đang phân vân là có nên dùng dấu ngoặc tròn hay không.
 - **Dấu ngoặc nhọn .**
- Theo tiêu chuẩn Java: dấu “{” phải được đặt cùng dòng với các câu if, for, while,... Nếu bạn nào đã code với C# thì sẽ thấy ngược lại, dấu “{” phải được đặt ở dòng mới.
 - **Comment.**
- Không viết các comment chỉ lặp code, comment thừa. Một số vấn đề gặp phải khi comment không tốt:
 - Các comment chỉ mô tả là lặp code, chứ không cung cấp thêm thông tin gì cho người đọc.
 - Làm code dài hơn.
 - Người đọc tốn thời gian đọc nhiều hơn.
- Viết các comment không cầu kì; càng đơn giản càng tốt.
- Khi dùng nhiều endline comment trên các dòng code liên tiếp nhau thì các comment này phải được canh lề như nhau.
- Nên vừa code vừa viết comment. Tránh trường hợp viết code xong rồi mới viết comment.
- Không nên đung chỗ nào cũng comment, chỉ viết comment khi bạn cảm nhận là đoạn code của mình quá phức tạp.

4.5.3 Một vài ước hoa thường trong đặt tên

- Sử dụng chữ cái Viết Hoa đúng là chìa khóa để tuân theo quy ước đặt tên:
- Lowercase: Là tên mà tất cả các chữ cái trong một từ được viết mà không Viết Hoa (ví dụ: while, if, mypackage).
- Uppercase: Là tên trong đó tất cả các chữ cái trong một từ được viết bằng chữ HOA. Khi có nhiều hơn hai từ trong tên, hãy sử dụng dấu gạch dưới để phân tách chúng (ví dụ: MAX_HOURS, FIRST_DAY_OF_WEEK).
- CamelCase: (còn được gọi là Upper CamelCase) là tên mà mỗi từ mới bắt đầu bằng chữ in hoa (ví dụ: CamelCase, CustomerAccount, PlayingCard).
- Mixed case: (còn được gọi là Lower CamelCase) giống như CamelCase ngoại trừ chữ cái đầu tiên của tên được viết bằng chữ thường (ví dụ: hasChildren, customerFirstName, customerLastName).

4.5.4 Các quy tắc đặt tên Java tiêu chuẩn

a) Quy ước đặt tên Packages tiêu chuẩn .

- Tên **packages** nên được **viết thường**. Với các dự án nhỏ chỉ có một vài package, bạn chỉ cần đặt cho chúng những cái tên đơn giản (nhưng có ý nghĩa!)
- package pokeranalyzer package mycalculator

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

- Nhưng trong các công ty phần mềm và các dự án lớn nơi các package có thể được nhập vào các package khác, các tên thường sẽ được chia nhỏ.
- Thông thường, điều này sẽ bắt đầu với domain công ty trước khi được chia thành các lớp hoặc tính năng .

- `package com.mycompany.utilities`
- `package org.bobscompany.application.userinterface`

b) Quy ước đặt tên Class Java tiêu chuẩn .

- Tên class trong Java phải áp dụng CamelCase. Cố gắng sử dụng danh từ vì một class thường đại diện cho một cái gì đó trong thế giới thực:

- `class Customer`
- `class Account`

c) Quy ước đặt tên Interface tiêu chuẩn.

- Tên Interfaces trong Java phải có áp dụng CamelCase. Chúng có xu hướng có một tên có thể mô tả một hoạt động mà một class có thể làm:

- `interface Comparable`
- `interface Enumerable`

- Lưu ý rằng một số lập trình viên thường muốn phân biệt các Interfaces bằng cách bắt đầu tên bằng 'I':

- `interface IComparable`
- `interface IEnumerable`

d) Quy ước đặt tên phương thức tiêu chuẩn trong Java.

- Tên Method trong Java nên được áp dụng quy tắc Mixed case. Và nên sử dụng các động từ để mô tả những gì phương thức sẽ làm:

- `void calculateTax()`
- `string getSurname()`

e) Quy tắc đặt tên biến tiêu chuẩn trong Java.

- Tên biến trong Java nên tuân theo quy tắc Mixed case. Các tên biến sẽ đại diện cho những gì giá trị của biến đại diện:

- `String firstName`
- `int orderNumber`

- Chỉ sử dụng tên biến ngắn khi các biến **tồn tại ngắn**, chẳng hạn như trong vòng lặp `for` .

- `for (int i=0; i<20;i++) { //i only lives in here }`

f) Quy ước đặt tên biến cho Hằng số (Constant) tiêu chuẩn trong Java .

- Tên hằng số tiêu chuẩn trong Java nên được Viết Hoa (quy tắc uppercase):

- `static final int DEFAULT_WIDTH`
- `static final int MAX_HEIGHT`

g) Biến static, enum .

- Tất cả các từ được viết hoa và phân cách bằng dấu gạch dưới (_).

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

h) Một số quy tắc khác :

- Tránh tình trạng khai báo biến mà không sử dụng: nhiều trình biên dịch warning khi compile (Eclipse IDE).
- Các lệnh if, while, for không nên lồng nhau hơn 3 bậc.
- Chỉ import thư viện sử dụng cần thiết. Không sử dụng import tất cả.
- Ví dụ: sử dụng `import java.util.List`; thay cho `import java.util.*`;

<MÃ DỰ ÁN> - ĐẶC TẢ YÊU CẦU PHẦN MỀM

TÀI LIỆU THAM KHẢO

[1] Tiêu chuẩn coding trong Java (Coding Standards) : <https://gpcoder.com/1775-tieu-chuan-coding-trong-java-coding-standards/>