

BÁO CÁO CUỐI KỲ ĐỒ ÁN MÔN HỌC

Ứng dụng Quản lý chi tiêu

Môn học: Phát triển ứng dụng trên thiết bị di động

GVHD: Th.s Trần Hồng Nghi

Lớp: NT118.011

Danh sách thành viên (Nhóm 3)

STT	Họ tên	MSSV
1	Đinh Văn Trường Giang	21520795
2	Quách Thị Hoài Phương	21520409
3	Huỳnh Phi Linh	21520323
4	Phạm Văn Phú	21522467



LỜI CẢM ƠN

Trước tiên hết, nhóm 3 xin gửi lời cảm ơn đến hội đồng lãnh đạo Trường Đại học Công nghệ Thông tin đã luôn lắng nghe, luôn tạo điều kiện học tập tốt nhất cho nhóm và cũng như toàn thể sinh viên trường. Tiếp đến xin dành lời cảm ơn sâu sắc đến giảng viên bộ môn - cô Trần Hồng Nghi, cô đã truyền đạt những kiến thức thực tế vừa quý báu, vừa vô cùng bổ ích cho nhóm và các bạn cùng lớp trong suốt thời gian học tập vừa qua. Qua môn học và đồ án do cô phụ trách, nhóm đã có thêm cho mình nhiều kiến thức thực tế quan trọng, tinh thần học tập hiệu quả. Khi chưa học môn này cả nhóm còn chưa có những suy nghĩ sâu xa và chưa có định hướng một cách cụ thể, luôn mang nỗi sợ đó là sau khi tốt nghiệp sẽ không thể tìm được việc làm, làm trái ngành hoặc làm đúng ngành nhưng lại không xứng đáng với quá trình, công sức bỏ ra để học đại học với tấm bằng kỹ sư. Sau khi học xong môn này nhóm đã có sự thay đổi trong tư tưởng và nhận thức của mình, tư duy và cách suy nghĩ của bản thân đã tiến triển rõ rệt và ngày càng tăng thêm sự quyết tâm trở thành một kỹ sư tài năng khiến các nhà tuyển dụng phải tranh giành mình chứ mình không phải đi xin việc nữa. Bộ môn “Phát triển ứng dụng di động” là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Mặc dù cả nhóm đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong cô xem xét và góp ý để bài tiểu luận của nhóm được hoàn thiện hơn.



MỤC LỤC

Chương 1. TỔNG QUAN ĐỀ TÀI	8
1.1. Lý do chọn đề tài	8
1.2. Triển khai đề tài	8
1.3. Mục tiêu nghiên cứu.....	9
Chương 2. CƠ SỞ LÝ THUYẾT	10
2.1. Tổng quan về Android Studio.....	10
2.2. Webservice – WAMP	11
Chương 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	13
3.1. Tổng quan về bài toán	13
3.2. Các chức năng chính	13
3.2.1. Quản lý tài khoản và đăng nhập.....	13
3.2.2. Quản lý giao dịch	17
3.2.3. Xem danh sách giao dịch.....	19
3.2.4. Báo cáo và thống kê.....	21
3.2.5. Quản lý danh mục	23
3.2.6. Cài đặt ứng dụng.....	24
3.3. Các chức năng tiện ích.....	24
3.3.1. Thông báo đẩy	24
3.3.2. Giới thiệu ứng dụng.....	25



3.3.3. Quản lý thông tin người dùng	25
3.3.4. Thông tin tác giả, gửi đóng góp ý kiến.....	25
Chương 4. THIẾT KẾ CƠ SỞ DỮ LIỆU.....	26
4.1. Mô tả bảng dữ liệu.....	26
4.1.1. Bảng Thông tin người dùng – ‘user_information’	26
4.1.2. Bảng Thu chi – ‘bill’	26
4.1.3. Bảng Danh mục – ‘category’	27
Chương 5. GIAO DIỆN CHƯƠNG TRÌNH.....	28
5.1. Đăng nhập/đăng ký	28
5.2. Quản lý các chức năng	29
5.3. Quản lý danh mục.....	30
5.4. Quản lý giao dịch	31
5.5. Xem danh sách giao dịch	32
5.6. Thống kê.....	33
5.7. Thông báo đẩy	34
5.8. Cài đặt.....	35
5.9. Hồ sơ thông tin cá nhân	36
5.10. Feedback/Contact.....	37
5.11. Giới thiệu ứng dụng.....	38
Chương 6. Thủ nghiệm hệ thống – Đánh giá hiệu năng.....	39



6.1. Quản lý fragment.....	39
6.2. Hiệu suất AsyncTask thống kê.....	39
6.3. Dung lượng ứng dụng.....	40
Chương 7. TỔNG KẾT	41
7.1. Kết luận.....	41
7.2. Định hướng phát triển	41



DANH MỤC HÌNH ẢNH

Hình 1. Sơ đồ tổng quan	13
Hình 2. Sơ đồ quản lý tài khoản và đăng nhập	14
Hình 3. Sơ đồ quản lý giao dịch.....	18
Hình 4. Sơ đồ xem danh sách giao dịch	20
Hình 5. Sơ đồ báo cáo và thống kê	22
Hình 6. Giao diện đăng nhập/đăng ký.....	28
Hình 7. Giao hiện trang chủ	29
Hình 8. Giao diện thêm mới danh mục/chỉnh sửa danh mục	30
Hình 9. Giao diện quản lý giao dịch	31
Hình 10. Giao diện theo dõi các hóa đơn.....	32
Hình 11. Giao diện biểu đồ thống kê thu/chi	33
Hình 12. Màn hình thông báo đẩy	34
Hình 13. Giao diện cài đặt ngôn ngữ ứng dụng	35
Hình 14. Giao diện hồ sơ thông tin cá nhân.....	36
Hình 15. Giao diện feedback/contact	37
Hình 16. Giao diện giới thiệu ứng dụng.....	38
Hình 17. Giải pháp cho crash chương trình.....	39



DANH MỤC BẢNG BIỂU

Bảng 1. Đăng ký	14
Bảng 2. Đăng nhập	16
Bảng 3. Quản lý giao dịch	18
Bảng 4. Xem danh sách giao dịch.....	20
Bảng 5. Báo cáo và thống kê.....	22
Bảng 6. Quản lý danh mục.....	23
Bảng 7. Bảng user_information	26
Bảng 8. Bảng bill.....	26
Bảng 9. Bảng category	27



Chương 1. TỔNG QUAN ĐỀ TÀI

1.1. Lí do chọn đề tài

- Cần thiết cho cuộc sống hiện đại: Trong thời đại số hóa ngày càng phát triển, việc quản lý tài chính cá nhân và gia đình trở nên quan trọng hơn bao giờ hết. Ứng dụng quản lý thu chi sẽ giúp người dùng nắm rõ tình hình tài chính, giảm thiểu rủi ro và tối ưu hóa quản lý thu chi.
- Cơ hội thúc đẩy kiến thức về lập trình di động: Quá trình phát triển ứng dụng này sẽ cung cấp cho tôi cơ hội để nắm vững các kiến thức về lập trình di động. Việc học cách phát triển ứng dụng trên nền tảng di động có thể cung cấp kỹ năng và kinh nghiệm quý báu, có thể áp dụng trong tương lai cho nhiều dự án khác.
- Đóng góp cho cộng đồng: Ứng dụng quản lý thu chi có tiềm năng hỗ trợ hàng triệu người dùng trong việc quản lý tài chính cá nhân. Điều này sẽ giúp cải thiện cuộc sống của họ, giảm căng thẳng tài chính và tạo ra giá trị cho cộng đồng.
- Phát triển kỹ năng xây dựng sản phẩm: Quá trình phát triển ứng dụng này sẽ đòi hỏi tôi phải nắm vững quy trình phát triển sản phẩm từ khâu thiết kế, phát triển, kiểm thử đến triển khai. Đây là cơ hội để rèn luyện kỹ năng quản lý dự án và tạo ra một sản phẩm chất lượng.
- Thách thức và cơ hội nghiên cứu: Đề tài này đòi hỏi tôi phải tìm hiểu sâu về quản lý thu chi, các ngôn ngữ lập trình di động và các công nghệ liên quan. Việc này sẽ giúp tôi nâng cao kiến thức và kỹ năng nghiên cứu trong lĩnh vực này.

1.2. Triển khai đề tài

- Công việc tìm hiểu và xây dựng ứng dụng quản lý chi tiêu
- Tham khảo, nghiên cứu các ứng dụng quản lý chi tiêu hiện tại:



- Khám phá cách thức hoạt động và giao diện của các ứng dụng quản lý chi tiêu đang có trên thị trường.
- Xem xét các chức năng chính và cách thức sắp xếp giao diện, bố cục, màu sắc, và layout của các ứng dụng.
- Lên kế hoạch và xác định yêu cầu cho ứng dụng:
 - Sử dụng tài nguyên từ Google và các nguồn thông tin khác để thiết lập yêu cầu cụ thể cho ứng dụng quản lý chi tiêu.
 - Khảo sát cơ sở lý thuyết để xác định nhu cầu thực tế của người dùng.
 - Tạo các sơ đồ và mô hình trước khi bắt đầu quá trình lập trình.

1.3. Mục tiêu nghiên cứu

- Hiểu rõ về quy trình phát triển ứng dụng trên Android Studio và các công cụ lập trình liên quan.
- Xác định và phân tích các yêu cầu cụ thể của ứng dụng quản lý chi tiêu.
- Xây dựng một ứng dụng đáp ứng các nhu cầu cơ bản của người dùng, cung cấp tiện ích trực tiếp trên điện thoại di động của họ.
- Ứng dụng giúp người dùng quản lý chi tiêu hàng ngày thông qua việc ghi chép và phân tích các khoản chi tiêu, cung cấp báo cáo và biểu đồ minh họa tình hình tài chính.
- Tạo cơ sở dữ liệu và giao diện để người dùng có thể theo dõi chi tiêu và tạo các danh mục chi tiêu theo nhu cầu cụ thể của họ.



Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về Android Studio

Android là hệ điều hành mã nguồn mở, dựa trên Linux Kernel, dành cho các thiết bị di động nói chung (điện thoại, máy tính bảng, đồng hồ thông minh, máy nghe nhạc,...).

Có nghĩa là Android không chỉ giới hạn trong phạm vi một hệ điều hành cho điện thoại! Nó có thể được nhà sản xuất cài đặt lên đồng hồ, máy nghe nhạc, thiết bị định vị GPS, thậm chí là ô tô (các thiết bị Android Auto).

Android cũng không phải là một thiết bị hay sản phẩm cụ thể, nó là một hệ điều hành dựa trên Linux, nguồn mở, linh hoạt.

Về kiến trúc của hệ điều hành Android:

- **Tầng Applications:** Là tầng chứa các ứng dụng Danh bạ, Gọi điện, Trình duyệt, Nghe nhạc,... các ứng dụng này thường mua máy về đã có sẵn.
- **Tầng Framework:** Là tầng chứa các API để làm việc với hệ điều hành như lấy thông tin danh bạ, quản lý các Activity, quản lý địa điểm, quản lý các View.
- **Tầng Libraries:** Chứa các thư viện, API gần như là cốt lõi của Android, bao gồm bộ quản lý bề mặt cảm ứng (Surface Manager), OpenGL (phục vụ cho việc dựng đồ họa phức tạp),...
- **Tầng Android Runtime:** Chứa các thư viện lõi của Android và máy ảo Dalvik Virtual Machine (từ Android 4 trở lên có thêm máy ảo ART).
- **Tầng Kernel:** Là nhân lõi của hệ điều hành, chứa các tập lệnh, driver giao tiếp giữa phần cứng và phần mềm của Android.

Trong quá trình làm việc, sẽ gần như chỉ làm việc với tầng Applications, Application Framework và Libraries. Chương trình Android được viết bằng ngôn



ngữ Java và được máy ảo DVM / ART trong mỗi thiết bị Android biên dịch ra mã máy.

2.2. Webservice – WAMP

WAMP là một viết tắt của "Windows, Apache, MySQL, PHP," và nó là một môi trường phát triển web phổ biến được sử dụng để phát triển ứng dụng web động dựa trên nền tảng Windows. Dưới đây là một số kiến thức cơ bản về các thành phần chính trong môi trường phát triển WAMP và cách sử dụng chúng trong việc phát triển ứng dụng Android bằng Android Studio:

- Windows: Windows là hệ điều hành sử dụng để chạy máy tính. WAMP thường được phát triển và triển khai trên hệ điều hành Windows, vì vậy cần có máy tính chạy Windows để sử dụng WAMP.
- Apache: Apache là một máy chủ web mã nguồn mở phổ biến được sử dụng để xử lý yêu cầu HTTP. Trong môi trường WAMP, Apache được cấu hình để chạy trên máy tính cá nhân, cho phép phát triển và kiểm tra ứng dụng web trên máy tính cá nhân trước khi triển khai lên máy chủ thực tế.
- MySQL: MySQL là một hệ quản trị cơ sở dữ liệu (DBMS) phổ biến được sử dụng để lưu trữ dữ liệu của ứng dụng web. Có thể sử dụng MySQL để tạo cơ sở dữ liệu và lưu trữ dữ liệu từ ứng dụng Android.
- PHP: PHP là một ngôn ngữ lập trình web phía máy chủ phổ biến. Có thể sử dụng PHP để phát triển các trang web động, xử lý yêu cầu từ ứng dụng Android và tương tác với cơ sở dữ liệu MySQL.

Đối với việc phát triển ứng dụng Android bằng Android Studio và sử dụng môi trường WAMP, có thể thực hiện các công việc sau:

- Phát triển máy chủ API: Sử dụng Apache và PHP trên máy tính chạy Windows để phát triển máy chủ API. API này có thể xử lý yêu cầu từ ứng dụng Android và tương tác với cơ sở dữ liệu MySQL để truy xuất và lưu trữ dữ liệu.



- Kiểm tra ứng dụng Android: Có thể sử dụng máy tính cá nhân làm máy chủ phát triển để kiểm tra ứng dụng Android của mình. Ứng dụng Android có thể gửi yêu cầu đến máy chủ API trên máy tính cá nhân để lấy hoặc cập nhật dữ liệu.
- Triển khai ứng dụng Android: Khi đã hoàn thiện ứng dụng Android, có thể triển khai nó lên một máy chủ thực tế hoặc môi trường sản xuất. Điều này đòi hỏi có một máy chủ web có thể truy cập từ internet và cấu hình lại ứng dụng Android để tương tác với máy chủ đó.

WAMP thường được sử dụng để phát triển và kiểm tra ứng dụng trên máy tính cá nhân và không nên sử dụng nó làm môi trường triển khai cho ứng dụng thực tế trên mạng Internet.



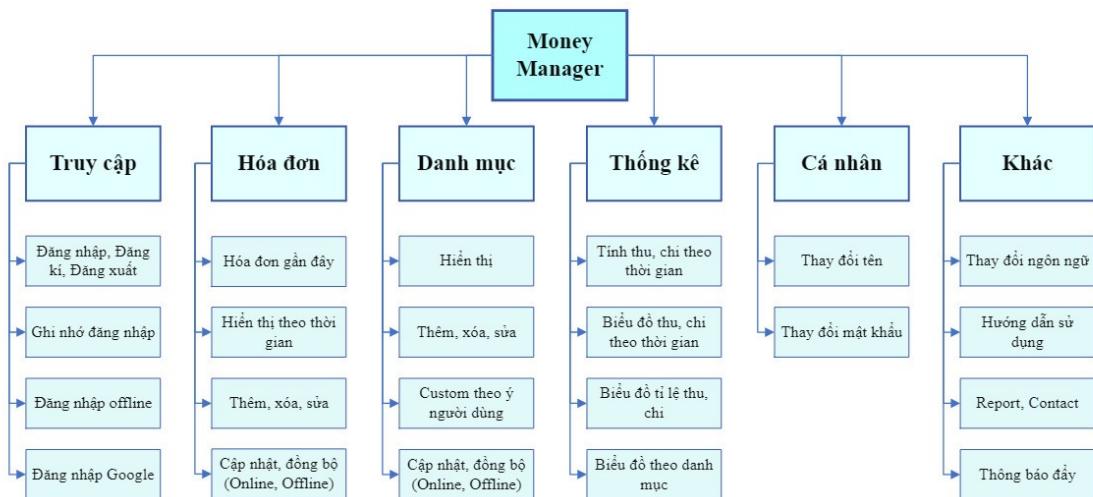
Chương 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Tổng quan về bài toán

Phần mềm xây dựng bằng Android Studio giao diện dễ sử dụng gồm các chức năng sau:

- Tất cả mọi người có nhu cần quản lý chi tiêu.
- Ứng dụng cung cấp cho người dùng nhập liệu, thống kê chi tiêu.

Qua đó, phần mềm sẽ giúp nghiệp vụ quản lý chuyên nghiệp dễ dàng hơn.



Hình 1. Sơ đồ tổng quan

Sơ đồ phân rã chức năng

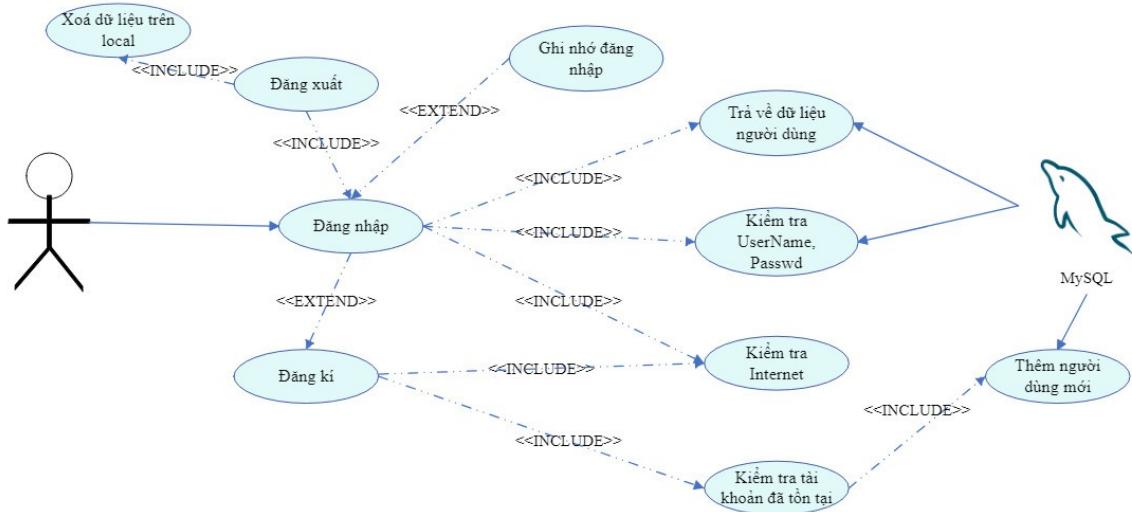
3.2. Các chức năng chính

3.2.1. Quản lý tài khoản và đăng nhập

- Đăng nhập: Người dùng có thể đăng nhập vào tài khoản của họ để truy cập dữ liệu cá nhân.
- Đăng ký tài khoản mới: Người dùng có thể đăng ký tài khoản mới để sử dụng ứng dụng.



- Đăng xuất: Người dùng có thể đăng xuất khỏi tài khoản hiện tại.



Hình 2. Sơ đồ quản lý tài khoản và đăng nhập

Bảng 1. Đăng ký

ĐĂNG KÝ	
Mô tả	Cho phép người dùng tạo tài khoản (họ tên, tên tài khoản, mật khẩu)
Sự kiện kích hoạt	Nhấn vào nút đăng ký
Kiểm tra điều kiện	Kiểm tra sự tồn tại của tài khoản và thông tin đầu vào hợp lệ Kiểm tra kết nối Internet
Xử lý sự kiện	Người dùng thực hiện đăng ký bằng tài khoản Google hoặc thông tin tài khoản, mật khẩu thông thường Nếu đăng ký bằng Google, thực hiện xác minh thành công hay không,



	<p>nếu nó chuyển đến bước kiểm tra sự tồn tại của tài khoản</p> <p>Nếu đăng ký bằng thông tin tài khoản thông thường, người dùng thực hiện nhập đầy đủ thông tin (họ tên, tên tài khoản, mật khẩu)</p> <p>Hệ thống kiểm tra định dạng sau đó kiểm tra kết nối Internet, nếu có thì chuyển đến bước kiểm tra sự tồn tài của tài khoản</p> <p>Ở cả 2 trường hợp, sau khi kiểm tra tài khoản hợp lệ, quy trình đăng ký thành công, chuyển về màn hình đăng nhập</p>
Xử lý ngoại lệ	<p>Tài khoản đã tồn tại: Thông báo tên tài khoản đã tồn tại, trả lại màn hình đăng ký để người dùng thực hiện đăng ký lại</p> <p>Thông tin bị bỏ sót: Thông báo cho người dùng nhập đầy đủ thông tin đầu vào và trả lại màn hình đăng ký để người dùng thực hiện lại</p> <p>Đăng ký khi không có kết nối Internet: Thông báo không có kết nối sẽ được hiển thị để người dùng có thể thực hiện đăng ký lại sau khi có kết nối.</p>



Bảng 2. Đăng nhập

ĐĂNG NHẬP	
Mô tả	Cho phép người dùng đăng nhập vào hệ thống bằng tài khoản đã đăng ký hoặc sử dụng gmail
Sự kiện kích hoạt	Nhấn vào nút đăng nhập (đăng nhập bằng tài khoản đã đăng ký) Nhấn vào biểu tượng Google trên màn hình đăng nhập để thực hiện liên kết với tài khoản gmail để đăng nhập Nhấn vào checkbox ghi nhớ đăng nhập để thực hiện đăng nhập trực tiếp vào tài khoản cho lần sử dụng sau không thông qua quy trình nhập tài khoản và mật khẩu lại từ đầu
Kiểm tra điều kiện	Kiểm tra đầy đủ các trường thông tin đầu vào Kiểm tra sự hợp lệ của tài khoản và mật khẩu
Xử lý sự kiện	Hệ thống kiểm tra checkbox ghi nhớ đăng nhập Nếu có, thực hiện kiểm tra kết nối Internet, ngược lại tiến hành bước tiếp theo



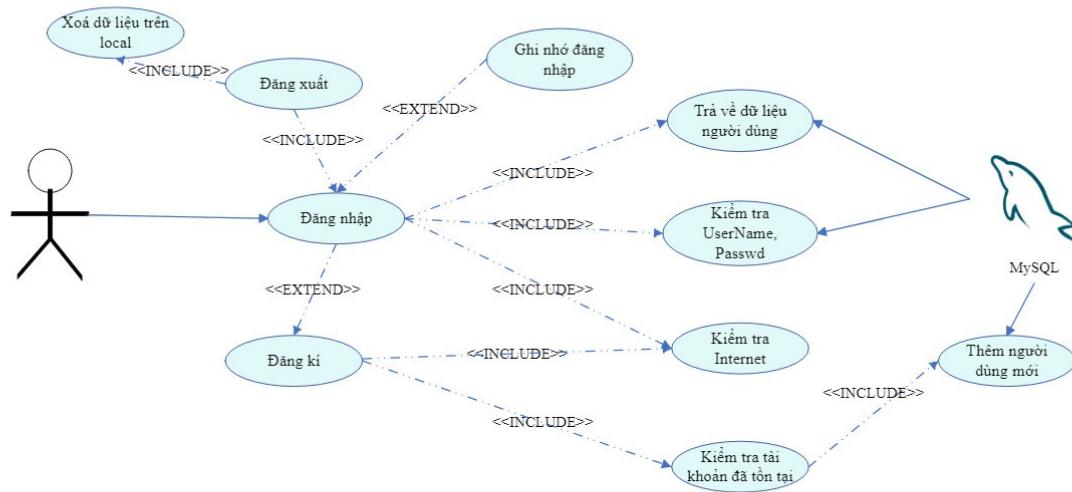
	<p>Kiểm tra Internet: nếu có hệ thống kiểm tra tính hợp lệ của tài khoản, mật khẩu, ngược lại, lấy thông tin từ Shared Preferences và kiểm tra tính hợp lệ</p> <p>Người dùng nhập tài khoản, mật khẩu</p> <p>Hệ thống kiểm tra đầy đủ các trường thông tin, kiểm tra kết nối Internet</p> <p>Kiểm tra kết nối Internet: nếu có, hệ thống thực hiện kiểm tra tính hợp lệ của tài khoản, mật khẩu</p> <p>Nếu đăng nhập thành công, chuyển đến màn hình trang chủ (Home)</p>
Xử lý ngoại lệ	<p>Tài khoản hoặc mật khẩu không hợp lệ: Thông báo sai tài khoản hoặc mật khẩu, quay lại màn hình đăng nhập để người dùng đăng nhập lại</p> <p>Thông tin bị bỏ sót: Thông báo cho người dùng nhập đầy đủ thông tin đầu vào và trở lại màn hình đăng nhập để người dùng thực hiện đăng nhập lại</p>

3.2.2. Quản lý giao dịch

- Thêm giao dịch mặc định: Người dùng có thể thêm các giao dịch mặc định vào hệ thống là các khoản thu chi hàng tháng, hàng quý,... mà không cần phải nhập lại nhiều lần.



- Thêm giao dịch mới: Người dùng có thể thêm một giao dịch mới vào hệ thống bằng cách cung cấp thông tin như số tiền, loại giao dịch (chi tiêu hoặc thu nhập), và mô tả.
- Sửa giao dịch: Người dùng có thể chỉnh sửa thông tin của một giao dịch đã tồn tại.
- Xóa giao dịch: Người dùng có thể xóa một giao dịch khỏi danh sách.



Hình 3. Sơ đồ quản lý giao dịch

Bảng 3. Quản lý giao dịch

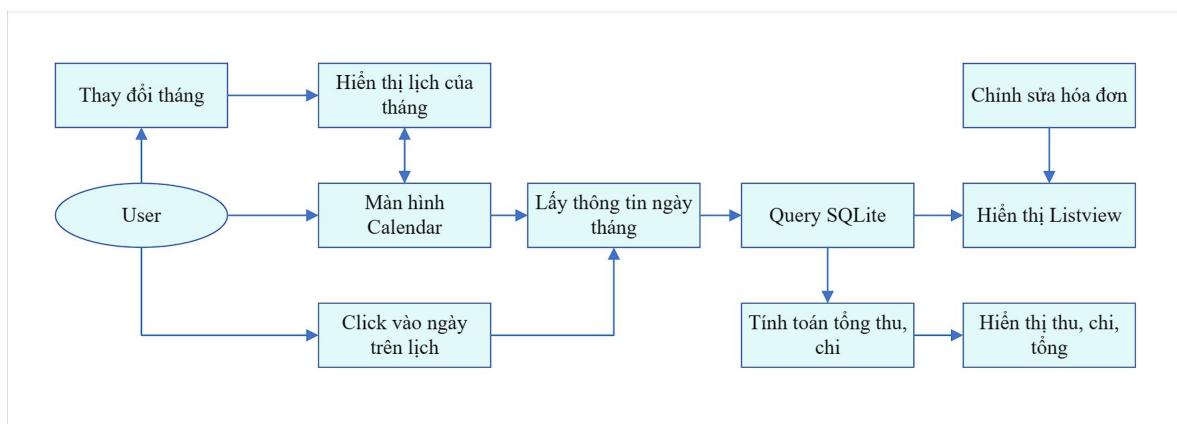
Quản lý giao dịch	
Mô tả	Thêm mới, chỉnh sửa, xoá giao dịch
Sự kiện kích hoạt	Nhấn nút Import tại trang Expense Nhấn vào hóa đơn đã thêm mới được liệt kê tại trang Calendar, chỉnh sửa thông tin tại dialog, nhấn nút chỉnh sửa và xoá
Kiểm tra điều kiện	Kiểm tra kết nối Internet



Xử lý sự kiện	Tại màn hình Expense, thực hiện thêm mới 1 giao dịch, người dùng nhập thông tin hoá đơn Hệ thống thực hiện kiểm tra định dạng Thêm mới hoá đơn, dữ liệu được đưa vào database local Hệ thống kiểm tra kết nối Internet Thêm mới hoá đơn thành công, hoá đơn được hiển thị liệt ra theo ngày, tháng, năm và hiển thị tại màn hình Calendar
Xử lý ngoại lệ	Trường hợp hoá đơn được thêm mới lúc không có Internet, dữ liệu được giữ tại database local sẽ được đồng bộ lên server

3.2.3. Xem danh sách giao dịch

- Xem danh sách tất cả các giao dịch: Người dùng có thể xem danh sách tất cả các giao dịch đã được thêm vào hệ thống.
- Xem chi tiết giao dịch: Người dùng có thể xem chi tiết về một giao dịch cụ thể, bao gồm số tiền, loại giao dịch, mô tả, và ngày thực hiện.



Hình 4. Sơ đồ xem danh sách giao dịch

Bảng 4. Xem danh sách giao dịch

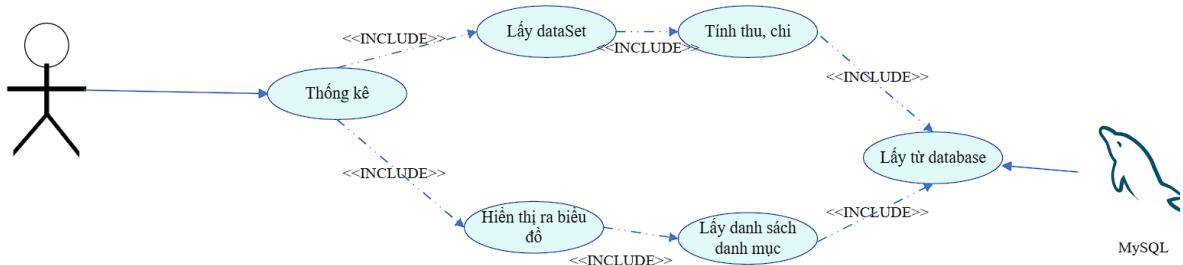
Xem danh sách giao dịch	
Mô tả	Hiển thị các hóa đơn/giao dịch mà người dùng thêm mới, thực hiện tổng thu/chi và chỉnh sửa hóa đơn
Sự kiện kích hoạt	Nhấn vào item(các hóa đơn) của listview được hiển thị ra (danh sách các hóa đơn được chọn theo ngày, tháng, năm) Chỉnh sửa dialog, nhấn vào nút Save để lưu chỉnh sửa và nút xoá để loại bỏ hóa đơn
Kiểm tra điều kiện	Kiểm tra kết nối Internet
Xử lý sự kiện	Người dùng nhấn thay đổi tháng Thực hiện hiển thị lịch tương ứng của tháng



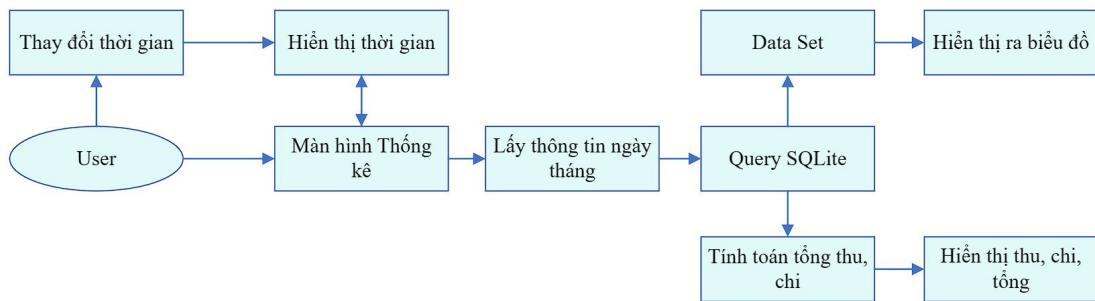
	<p>Người dùng nhấn vào ngày cụ thể trên lịch</p> <p>Thực hiện hiển thị ra danh sách hóa đơn của ngày đó</p> <p>Hệ thống lấy thông tin ngày tháng, query đến database local và thực hiện tính toán tổng thu chi để hiển thị ra màn hình Calendar</p>
Xử lý ngoại lệ	Trường hợp xoá hóa đơn khi không có Internet: Thông báo không có kết nối Internet để người dùng có thể thực hiện xoá sau khi có kết nối lại

3.2.4. Báo cáo và thống kê

- Tạo báo cáo: Người dùng có thể tạo báo cáo về tình hình chi tiêu hoặc thu nhập trong một khoảng thời gian cụ thể.
- Xem biểu đồ thống kê: Người dùng có thể xem biểu đồ hoặc biểu đồ thống kê về các loại chi tiêu hoặc thu nhập.



Hình 5. Sơ đồ chức năng thống kê



Hình 6. Sơ đồ báo cáo và thống kê

Bảng 5. Báo cáo và thống kê

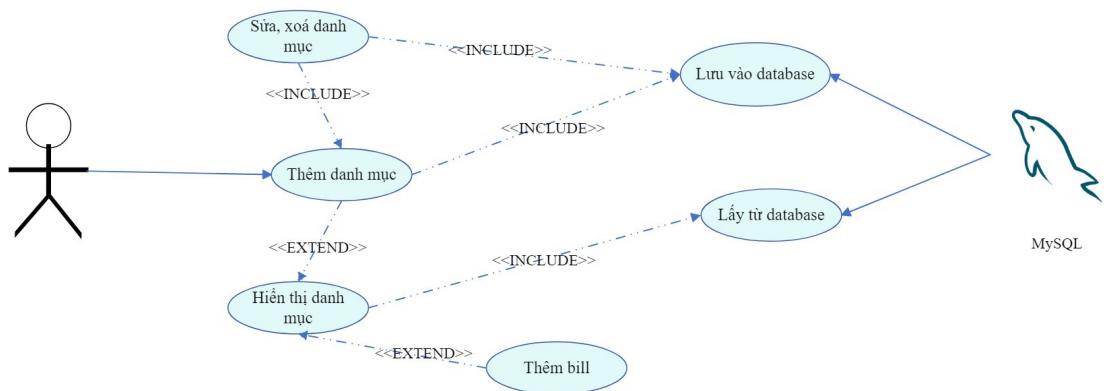
Báo cáo và thống kê	
Mô tả	Biểu đồ thống kê thu/chi
Sự kiện kích hoạt	Nhấn vào biểu đồ để xem chi tiết thu/chi
Kiểm tra điều kiện	
Xử lý sự kiện	Người dùng thực hiện thay đổi thời gian Hệ thống hiển thị thời gian ra màn hình thống kê Hệ thống lấy thông tin ngày tháng Query đến database local Tính toán tổng thu/chi và hiển thị ra màn hình Đồng thời, sử dụng Data Set hiển thị ra biểu đồ



Xử lý ngoại lệ

3.2.5. Quản lý danh mục

- Thêm danh mục mới: Người dùng có thể thêm một danh mục mới cho các loại giao dịch.
- Sửa danh mục: Người dùng có thể chỉnh sửa thông tin về một danh mục đã tồn tại (ví dụ: đổi tên danh mục).
- Xóa danh mục: Người dùng có thể xóa một danh mục đã tồn tại.



Hình 7. Use case quản lý danh mục

Bảng 6. Quản lý danh mục

Quản lý danh mục	
Mô tả	Thêm danh mục mới, sửa và xoá danh mục từ danh sách danh mục
Sự kiện kích hoạt	Nhấn vào những item (danh mục cụ thể) trên gridview để hiển thị dialog Tại dialog, nhấn vào các item cần chỉnh sửa



	Nhấn nút Save để lưu chỉnh sửa và Delete để loại bỏ danh mục ra khỏi danh sách
Kiểm tra điều kiện	Kiểm tra kết nối Internet
Xử lý sự kiện	Người dùng chọn danh mục cụ thể trên gridview Hệ thống lấy danh sách danh mục từ database và hiển thị ra dialog Người dùng thực hiện chỉnh sửa theo ý muốn Hệ thống cập nhật lại danh sách danh mục tại database local, sau đó thực hiện đồng bộ lại với server
Xử lý ngoại lệ	Chỉnh sửa danh mục khi không có Internet: tương tự như hoá đơn, danh mục sẽ được đồng bộ lên server khi có kết nối lại

3.2.6. Cài đặt ứng dụng

- Thay đổi ngôn ngữ

3.3. Các chức năng tiện ích

3.3.1. Thông báo đẩy

- Thông báo khi thời gian người dùng không sử dụng ứng dụng vượt quá 24 giờ.



3.3.2. Giới thiệu ứng dụng

- Các trang thông tin về giới thiệu và hướng dẫn sử dụng ứng dụng sẽ hiện ở lần đầu cài đặt ứng dụng.
- Người dùng có thể xem lại ở giao diện tính năng.

3.3.3. Quản lý thông tin người dùng

- Thay đổi tên.
- Thay đổi mật khẩu

3.3.4. Thông tin tác giả, gửi đóng góp ý kiến

- Xem thông tin liên hệ của tác giả: Người dùng có thể xem, truy cập vào trang thông tin tác giả.
- Gửi đóng góp ý kiến: Người dùng có thể gửi đóng góp ý kiến qua mail.



Chương 4. THIẾT KẾ CƠ SỞ DỮ LIỆU

4.1. Mô tả bảng dữ liệu

4.1.1. Bảng Thông tin người dùng – ‘user_information’

Bảng 7. Bảng user_information

STT	Thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	UserID (PK)	int (11)	Mã người dùng
2	FullName	varchar (50)	Họ và tên
3	UserName	varchar (50)	Tên đăng nhập
4	Password	varchar (50)	Mật khẩu
5	Email	varchar (50)	Email người dùng

4.1.2. Bảng Thu chi – ‘bill’

Bảng 8. Bảng bill

STT	Thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	ID (PK)	varchar(36)	Mã hóa đơn
2	UserID	int (11)	Mã người dùng
3	CategoryID	varchar(36)	Mã phân loại
4	Note	varchar(150)	Tên hóa đơn
5	TimeCreate	datetime	Thời điểm thanh toán
6	Expense	double	Số tiền
7	Sync_status	int(11)	Kiểm tra tình trạng sync của record.



4.1.3. Bảng Danh mục – ‘category’

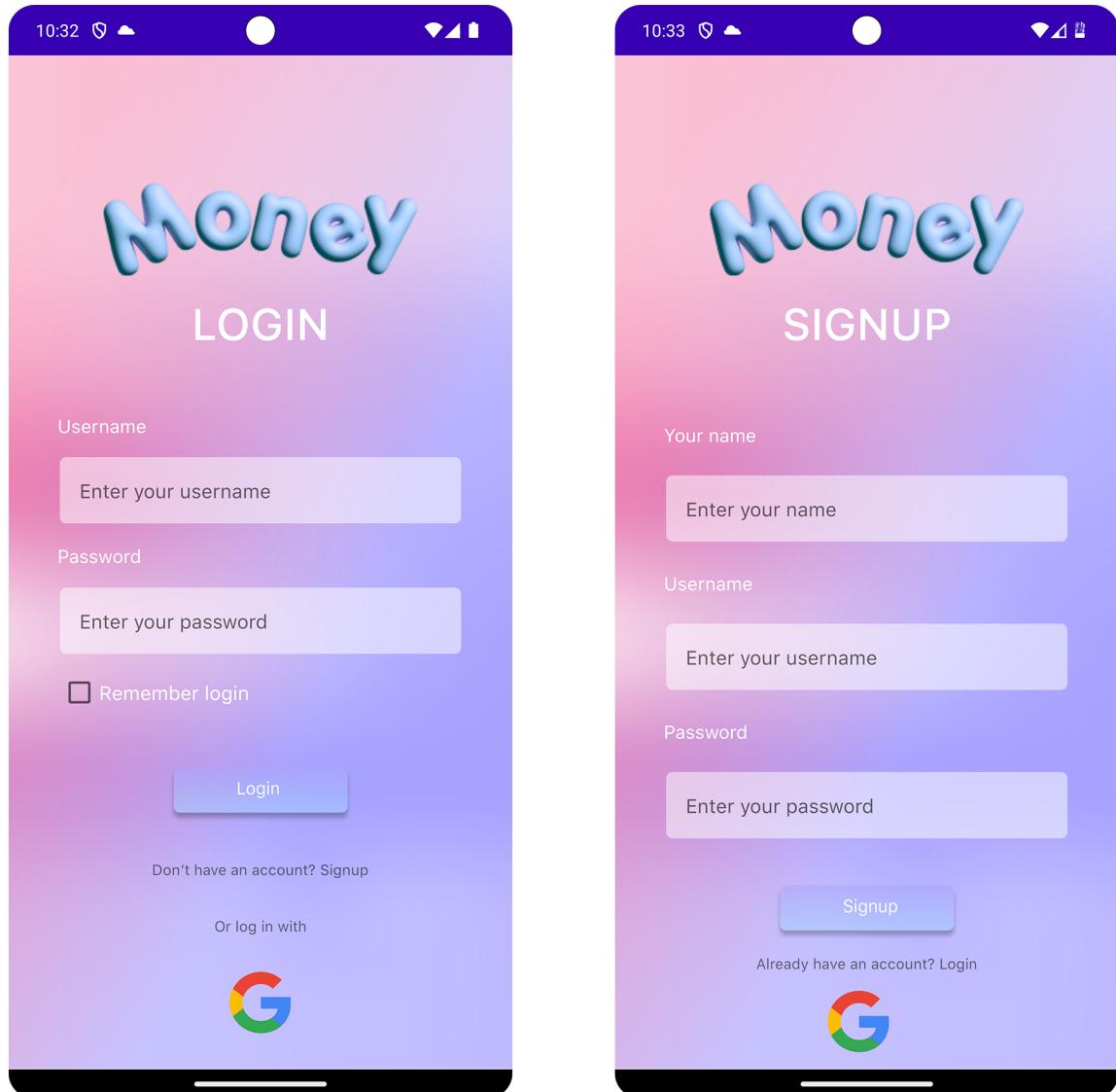
Bảng 9. Bảng category

STT	Thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	ID (PK)	varchar(36)	Mã phân loại
2	UserID	int(11)	Mã người dùng
3	Name	varchar (150)	Tên phân loại
4	Icon	varchar (150)	Icon của phân loại
5	Color	varchar (150)	Color của phân loại
6	Sync_status	int(11)	Kiểm tra tình trạng sync của record.



Chương 5. GIAO DIỆN CHƯƠNG TRÌNH

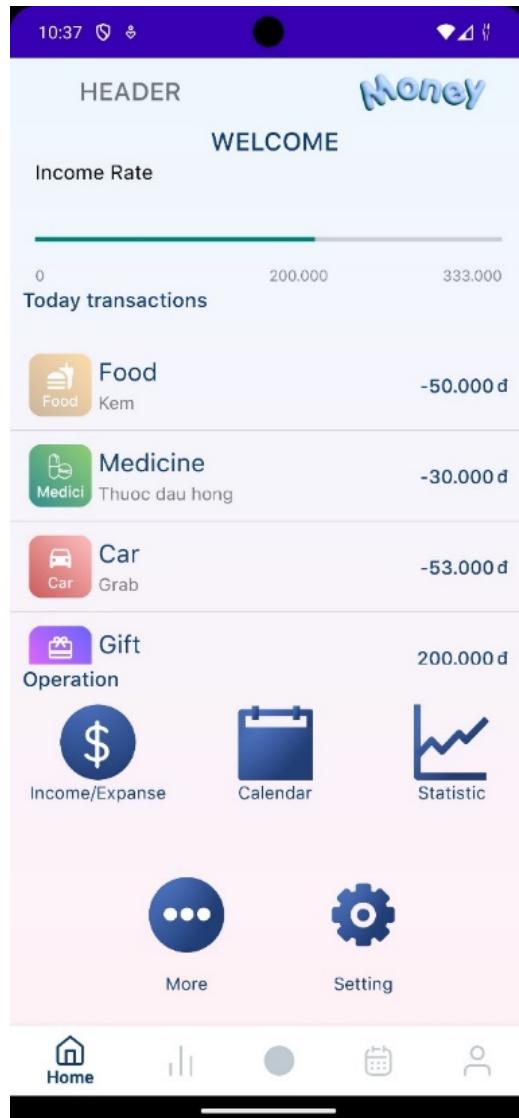
5.1. Đăng nhập/dăng ký



Hình 8. Giao diện đăng nhập/dăng ký



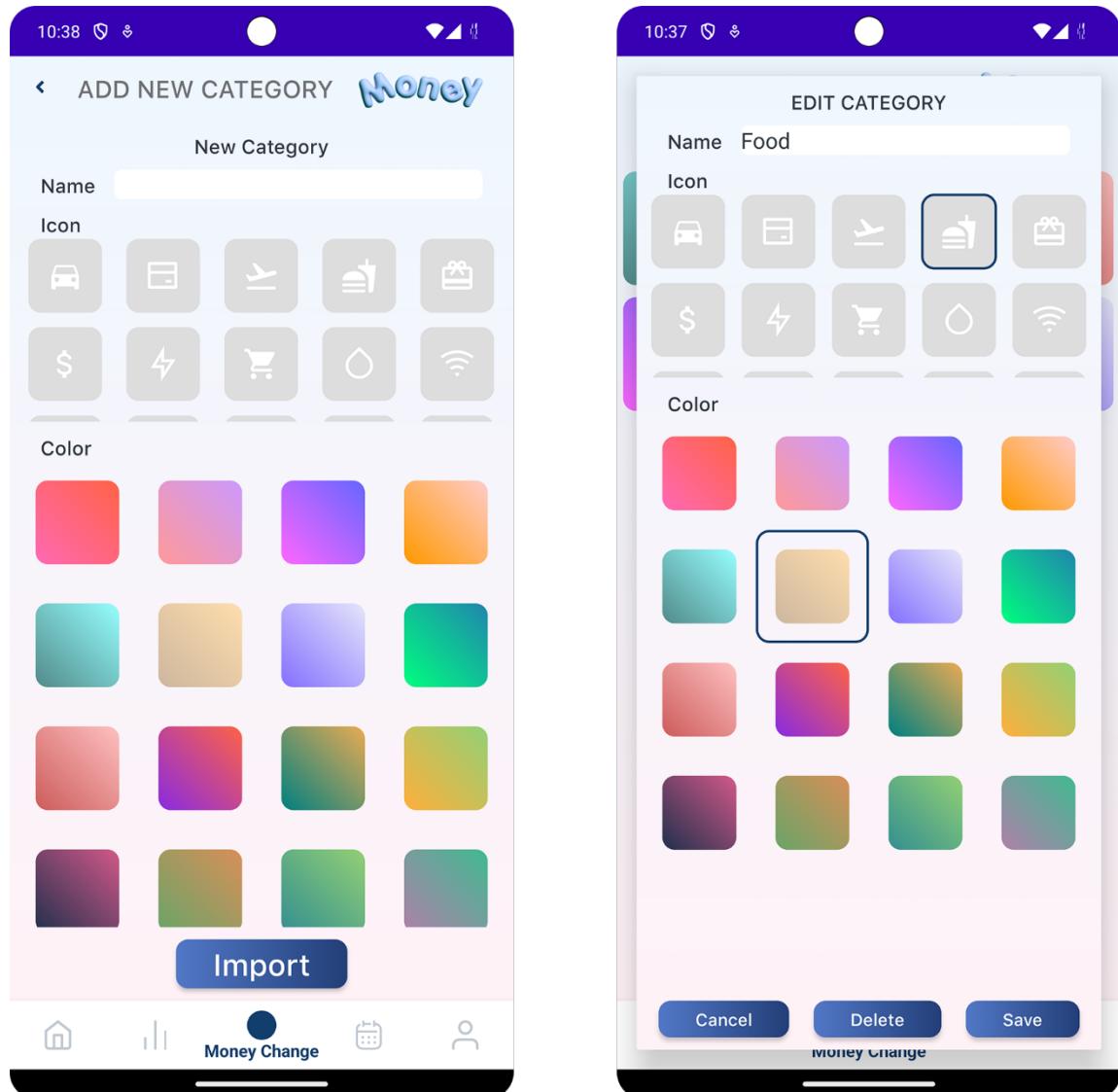
5.2. Quản lý các chức năng



Hình 9. Giao diện trang chủ



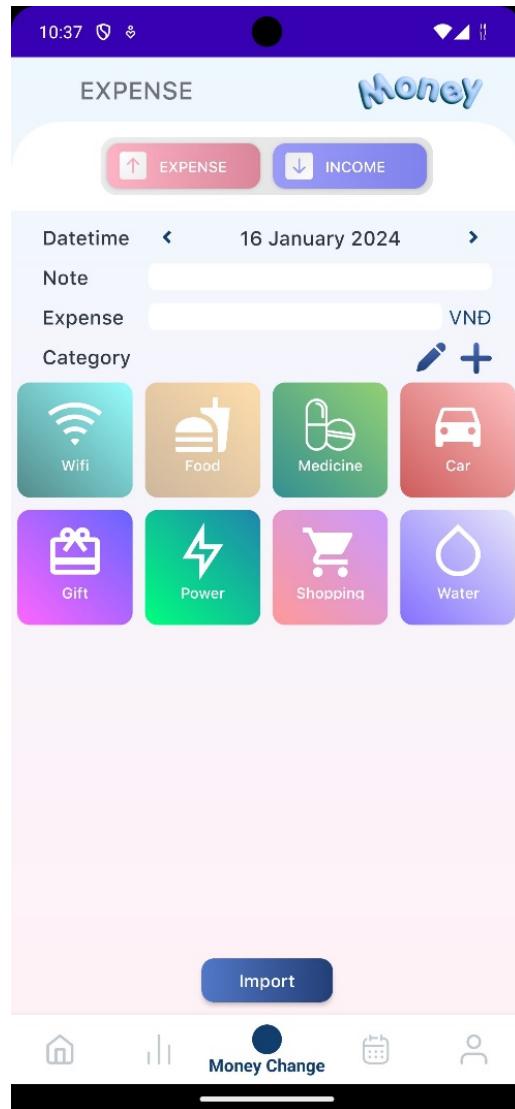
5.3. Quản lý danh mục



Hình 10. Giao diện thêm mới danh mục/chỉnh sửa danh mục



5.4. Quản lý giao dịch



Hình 11. Giao diện quản lý giao dịch



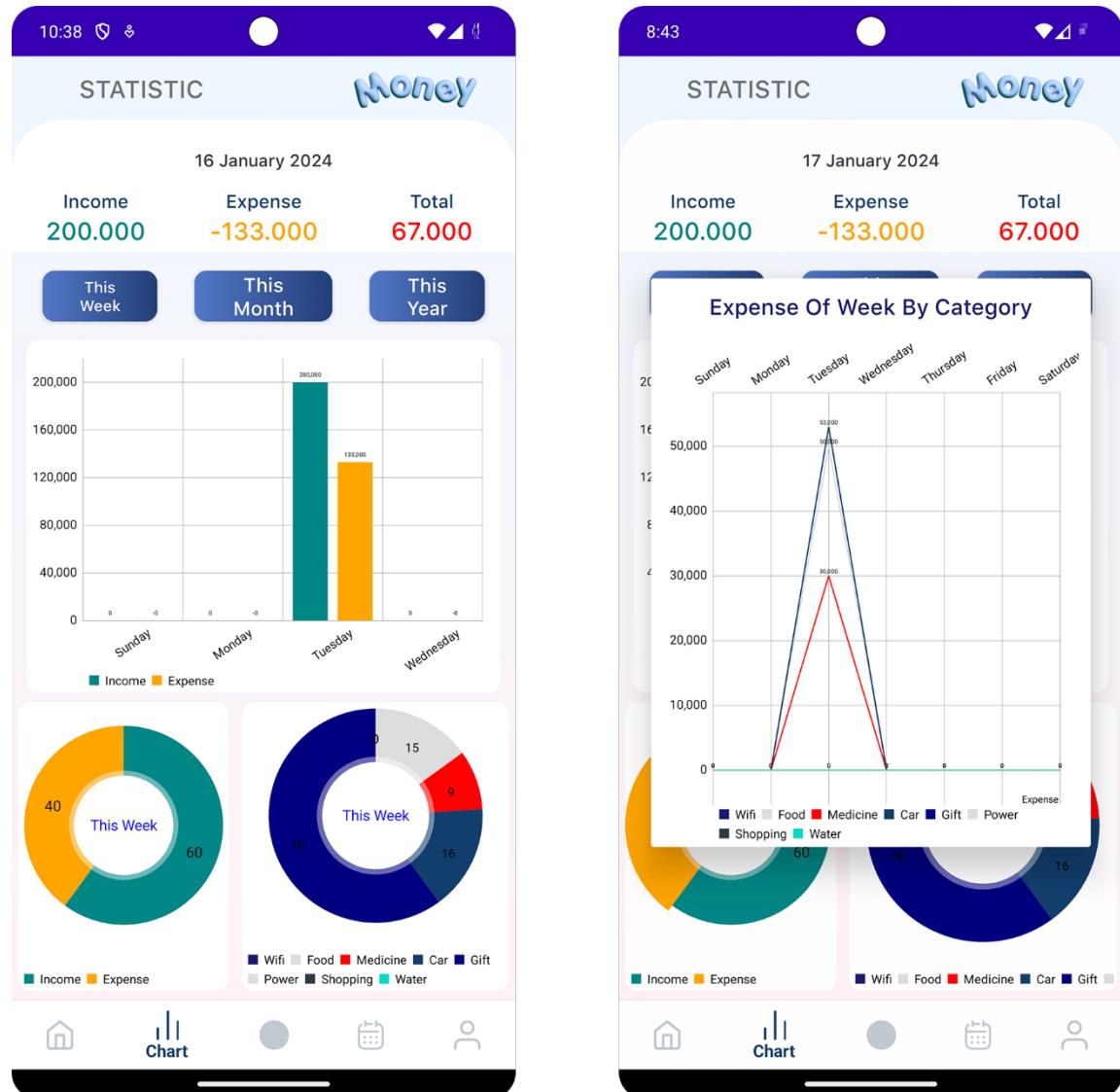
5.5. Xem danh sách giao dịch



Hình 12. Giao diện theo dõi các hóa đơn



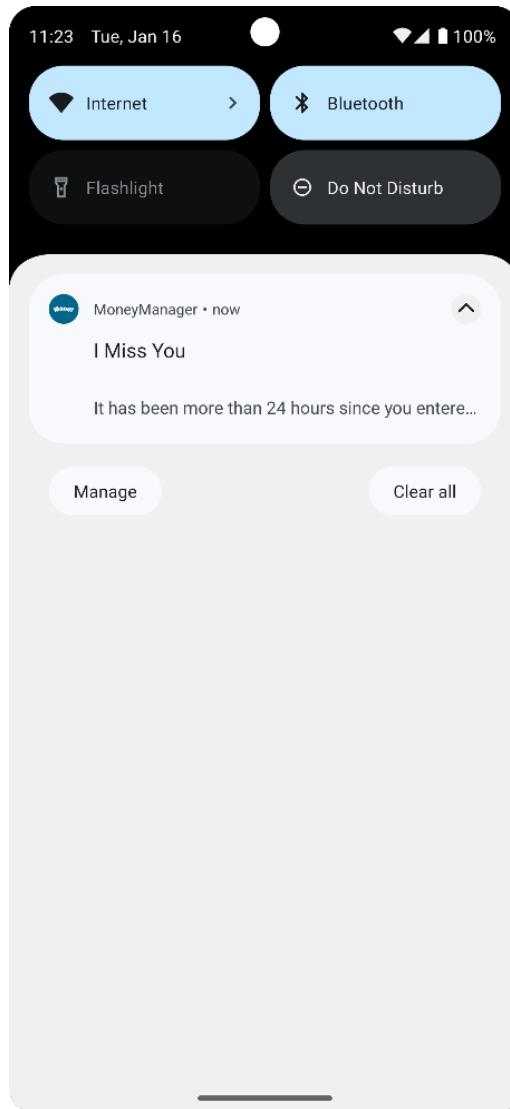
5.6. Thống kê



Hình 13. Giao diện biểu đồ thống kê thu/chi



5.7. Thông báo đẩy



Hình 14. Màn hình thông báo đẩy



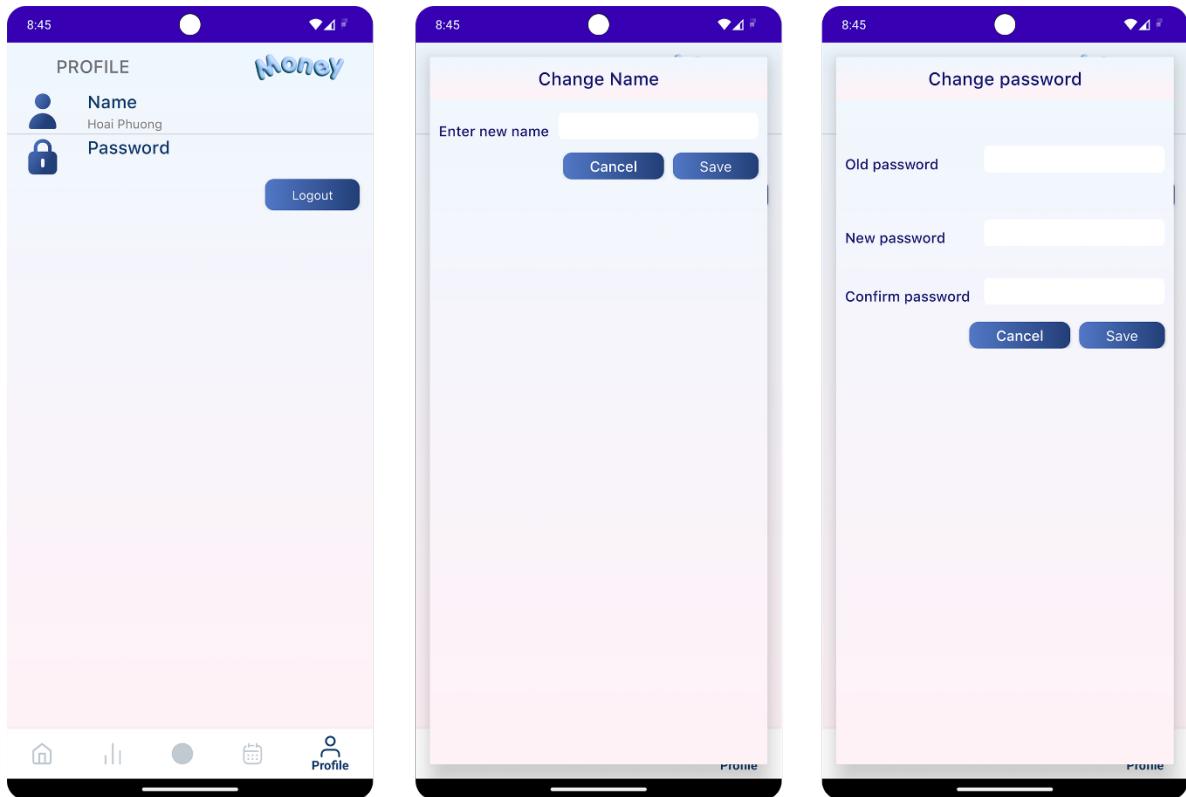
5.8. Cài đặt



Hình 15. Giao diện cài đặt ngôn ngữ ứng dụng



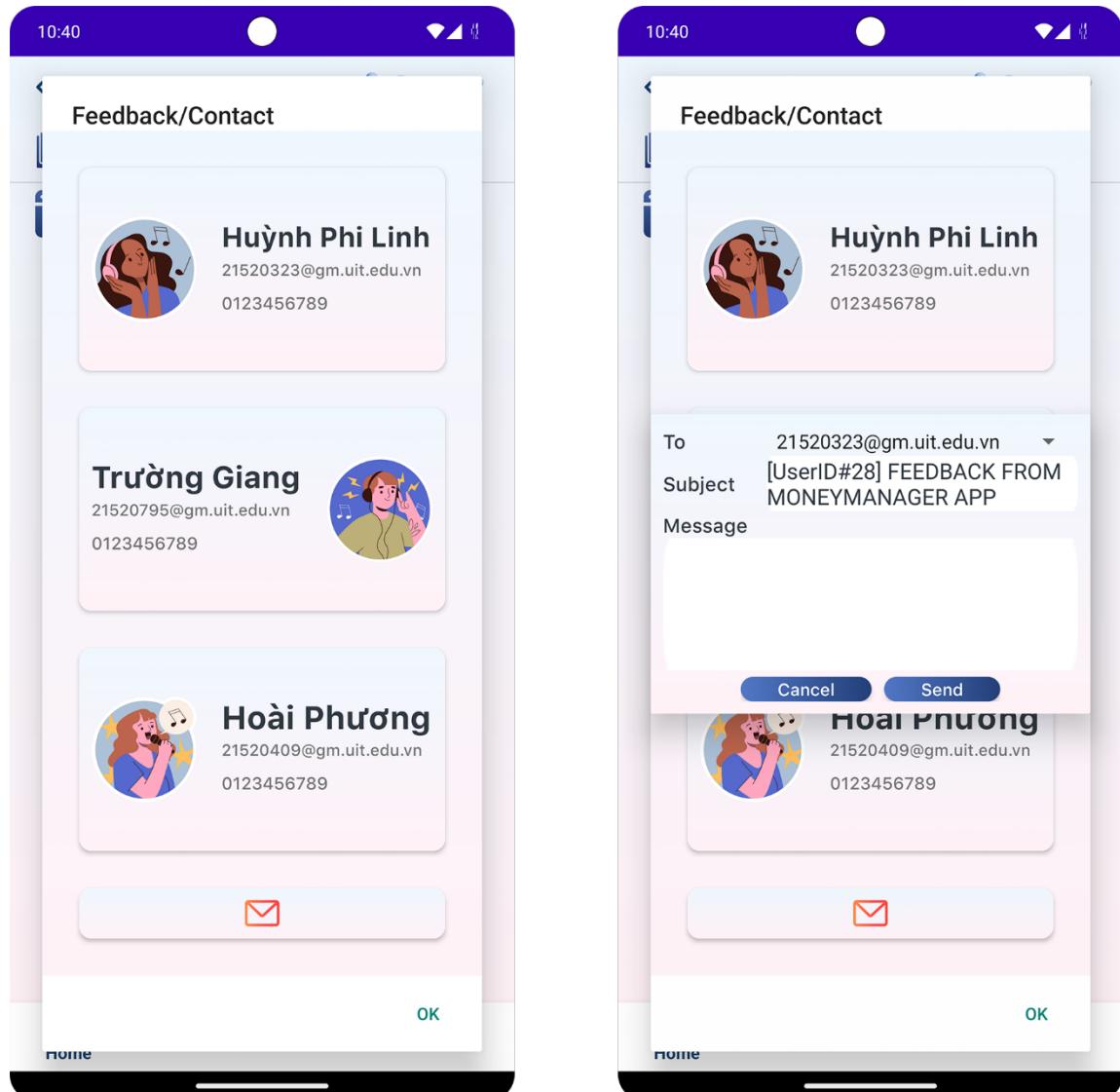
5.9. Hồ sơ thông tin cá nhân



Hình 16. Giao diện hồ sơ thông tin cá nhân



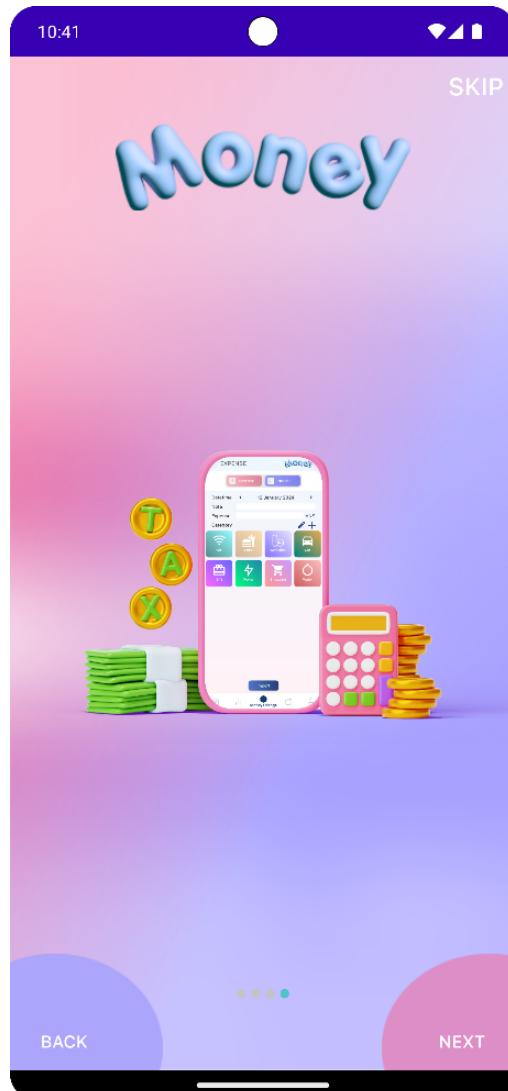
5.10. Feedback/Contact



Hình 17. Giao diện feedback/contact



5.11. Giới thiệu ứng dụng



Hình 18. Giao diện giới thiệu ứng dụng



Chương 6. Thủ nghiệm hệ thống – Đánh giá hiệu năng

6.1. Quản lý fragment

Ứng dụng nhóm thực hiện sử dụng nhiều fragment cho các giao diện chính. Để quản lý chính xác sự thay đổi thêm, xoá fragment khá khó khăn. Đôi khi sẽ gặp tình trạng crash chương trình vì sự cố đóng mở fragment.

Thực tế, nhóm đã gặp nhiều tình trạng bị crash chủ yếu do chưa khai báo fragment mới. Một trường hợp đặc biệt hơn, khi mở fragment Statistic (giao diện Thống kê), ngay khi AsyncTask đang chạy doInBackground để tính toán dữ liệu hiện lên biểu đồ thì chuyển fragment khác. Lúc này chương trình sẽ bị crash. Nguyên nhân là do khi AsyncTask đang chạy thì yêu cầu bắt buộc cần có fragment của nó được gọi. Tuy nhiên người dùng lại chuyển sang fragment khác khiến fragment hiện tại bị ngắt, AsyncTask thì không biết vì sao thiếu fragment nên lập tức chương trình bị crash. Giải pháp đưa ra là trong phương thức onDetach() (phương thức này được gọi khi Fragment không còn được gắn kết với hoạt động (Activity) nữa) của fragment Statistic thì sẽ hủy bỏ AsyncTask.

```
@Override
public void onDetach() {
    super.onDetach();
    if (cal != null && !cal.isCancelled()) {
        cal.cancel(mayInterruptIfRunning: true);
    }
}
```

Hình 19. Giải pháp cho crash chương trình

6.2. Hiệu suất AsyncTask thống kê

Vào giai đoạn đầu khi hiện thực hóa chức năng thống kê với chỉ một biểu đồ đường duy nhất, cũng như khối lượng dữ liệu cần lấy từ cơ sở dữ liệu là chưa nhiều, do đó việc vẽ biểu đồ được thực hiện trơn tru và có độ trễ rất ít.

Tuy nhiên khi thêm 2 biểu đồ tròn và 1 biểu đồ đường vào chức năng thống kê, tệp dữ liệu cần lấy đã tăng lên đáng kể (theo thời gian, theo danh mục, ...) đã ảnh



hưởng đáng kể đến hiệu suất của ứng dụng. Tình trạng load dữ liệu chậm hay thậm chí là crash chương trình xảy ra thường xuyên. Do đó giải pháp được đưa ra áp dụng đó là chuyển toàn bộ tác vụ lấy dữ liệu hóa đơn và tính toán, vẽ biểu đồ vào AsyncTask. Bằng cách này, hiệu suất của chương trình được cải thiện rất rõ rệt, việc chương trình bị thoát giữa chừng rất hiếm khi xảy ra, kể cả khi người dùng thực hiện chuyển giao diện qua lại nhanh.

6.3. Dung lượng ứng dụng

Ứng dụng sử dụng một số tài nguyên trực tiếp là file *.png. Điều này khiến dung lượng ứng dụng tăng lên đáng kể một cách không cần thiết. Giải pháp đưa ra ở đây là chuyển các tài nguyên này thành dạng vector, các file *.xml và xoá các file không cần thiết. Bên cạnh đó có thể sử dụng các tài nguyên drawable chung cho nhiều màn hình. Với các màn hình giao diện vì trước đó kiến thức còn hạn chế nên đã lạm dụng nhiều layout trong một giao diện. Nhóm đã chỉnh sửa, cải tiến và sử dụng ConstraintLayout thay vì RelativeLayout hoặc LinearLayout nếu có thể.



Chương 7. TỔNG KẾT

7.1. Kết luận

Thực hiện đề tài “Ứng dụng quản lý thu chi” giúp nhóm đạt được những kết quả về kiến thức cũng như kinh nghiệm xử lý bài toán. Bên cạnh đó nắm bắt yêu cầu cụ thể của người dùng trong việc quản lý thu chi hàng ngày. Trau dồi kinh nghiệm về xử lý giao diện người dùng với sự chú ý đặc biệt đến trải nghiệm người dùng và thiết kế thẩm mỹ. Điều này giúp người dùng dễ dàng tương tác với ứng dụng, thúc đẩy việc sử dụng lâu dài. Ngoài ra chú trọng đến việc đảm bảo tính an toàn và bảo mật thông tin cá nhân của người dùng. Các thông tin tài khoản và giao dịch được mã hóa và bảo vệ chặt chẽ, giúp người dùng yên tâm khi sử dụng ứng dụng. Quá trình làm đồ án này đã giúp nhóm học hỏi rất nhiều về phát triển ứng dụng di động, quản lý dự án, và tương tác với cộng đồng người dùng. Những kỹ năng này sẽ là nền tảng quan trọng cho sự phát triển cá nhân và sự nghiệp trong tương lai.

7.2. Định hướng phát triển

- Thêm kế hoạch chi tiêu nâng cao:
 - Mở rộng tính năng kế hoạch chi tiêu để cho phép người dùng đặt mục tiêu tài chính cụ thể cho từng danh mục hoặc thời gian cụ thể.
 - Thêm khả năng theo dõi và đánh giá hiệu suất của kế hoạch chi tiêu so với thực tế.
- Tích hợp trí tuệ nhân tạo (AI):
 - Sử dụng công nghệ AI để phân tích thông tin tài chính cá nhân và đề xuất kế hoạch chi tiêu hợp lý dựa trên hành vi và thu chi trước đó.
 - Cung cấp gợi ý về cách tiết kiệm chi phí hoặc tối ưu hóa chi tiêu dựa trên xu hướng người dùng và tình hình tài chính.



- Hệ thống thông báo nâng cao:

- Tích hợp hệ thống thông báo thông minh sử dụng AI để cảnh báo người dùng về những thay đổi lớn trong tình hình tài chính hoặc khi cần điều chỉnh kế hoạch chi tiêu.
- Gửi thông báo nhắc nhở cho người dùng về các kế hoạch chi tiêu hoặc hóa đơn sắp đến.

- Hỗ trợ đa nền tảng:

- Phát triển phiên bản cho các nền tảng khác nhau như iOS để mở rộng cơ hội sử dụng.



TÀI LIỆU THAM KHẢO

- [1] – Android for Developers - <https://developer.android.com/>
- [2] – CalendarTutorialAndroidStudio -
<https://github.com/codeWithCal/CalendarTutorialAndroidStudio>
- [3] – MPAndroidChart - <https://github.com/PhilJay/MPAndroidChart>