

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

.....



NGUYỄN HỒNG THÔNG ĐIỆP - 21110166

ĐOÀN THÁI SƠN - 21110289

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH

TIÊU LUẬN CHUYÊN NGÀNH CNPM

GIẢNG VIÊN HƯỚNG DẪN:

THS. NGUYỄN HỮU TRUNG

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

.....



NGUYỄN HỒNG THÔNG ĐIỆP - 21110166

ĐOÀN THÁI SƠN - 21110289

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH

TIÊU LUẬN CHUYÊN NGÀNH CNPM

GIẢNG VIÊN HƯỚNG DẪN:

THS. NGUYỄN HỮU TRUNG

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên 1: Nguyễn Hồng Thông Đieber **MSSV:** 21110166

Họ và tên sinh viên 2: Đoàn Thái Sơn **MSSV:** 21110289

Ngành: Công nghệ thông tin

Tên đề tài: XÂY DỰNG ỦNG DỤNG QUẢN LÝ TÀI CHÍNH

Giảng viên hướng dẫn: ThS. Nguyễn Hữu Trung

NHẬN XÉT

1. Về nội dung đề tài và khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại:

6. Điểm:

Tp. Hồ Chí Minh, ngày... tháng 12 năm 2024

Giảng viên hướng dẫn

(Ký & ghi rõ họ tên)

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên sinh viên 1: Nguyễn Hồng Thông Đieber **MSSV:** 21110166

Họ và tên sinh viên 2: Đoàn Thái Sơn **MSSV:** 21110289

Ngành: Công nghệ thông tin

Tên đề tài: XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH

Giảng viên phản biện: ThS. Trương Thị Ngọc Phượng

NHẬN XÉT

1. Về nội dung đề tài và khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại:

6. Điểm:

Tp. Hồ Chí Minh, ngày... tháng 12 năm 2024

Giảng viên phản biện

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN.

Trước khi đi vào nội dung của báo cáo, nhóm chúng em xin chân thành cảm ơn Trường Đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh đã tạo cơ hội cho toàn thể sinh viên khoa Công Nghệ Thông Tin nói riêng và các thành viên nhóm em nói riêng có thể có cơ hội thực tiễn luận chuyên ngành.

Tiếp theo, nhóm chúng em không thể quên gửi lời cảm ơn chân thành đến người đã giúp nhóm chúng em có được sự tự tin về kiến thức để thực hiện được đề tài lần này, thầy Nguyễn Hữu Trung. Trong suốt 15 tuần học, thầy là người trực tiếp chỉ dẫn để nhóm chúng em gặt hái được kiến thức về các vấn đề liên quan việc thực hiện một dự án phần mềm.

Tuy nhóm đã nỗ lực trong suốt quá trình học tập nhưng việc còn tồn tại những thiếu sót về mặt kiến thức là không thể tránh khỏi. Những thiếu sót ấy có thể được hiện hữu trong đồ án của nhóm. Nếu bắt gặp bất cứ sai sót nào, mong quý thầy cô cũng như các bạn có thể phản hồi cho chúng em, nhóm luôn vui lòng ghi nhận lấy những thiếu sót của mình và khắc phục nó nhanh nhất có thể.

Lời cuối cùng, nhóm chúng em kính chúc thầy luôn mạnh khỏe, hạnh phúc, thành công trong cuộc sống.

Nhóm sinh viên thực hiện:

Nguyễn Hồng Thông Đieber

Đoàn Thái Sơn

ĐỀ CƯƠNG CHI TIẾT TIỂU LUẬN CHUYÊN NGÀNH

Họ và tên sinh viên thực hiện 1: **Nguyễn Hồng Thông Đieber** MSSV 2: **21110166**

Họ và tên sinh viên thực hiện 2: **Đoàn Thái Sơn** MSSV 1: **21110289**

Thời gian làm tiểu luận chuyên ngành: Học kỳ 1 năm học 2024-2025

Chuyên ngành: **Công nghệ phần mềm**

Tên đề tài: **XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH**

Giáo viên hướng dẫn: **ThS. Nguyễn Hữu Trung**

Nhiệm vụ của tiểu luận chuyên ngành:

Lý thuyết:

- + React Native, Tailwind CSS.
- + Firebase SDK (JavaScript SDK) bao gồm:
 - o Firebase Authentication SDK.
 - o Firebase Storage SDK.
 - o Firebase Realtime Database SDK.
 - o Firebase Cloud Functions API.
 - o Firebase Token.
- + Mailjet

Thực hành

- + Sử dụng React Native và Tailwind CSS để xây dựng giao diện ứng dụng.
- + Sử dụng Firebase để triển khai backend (quản lý dữ liệu và xác thực):
 - o Firebase Authentication SDK: Dùng để đăng ký, đăng nhập và xác thực người dùng.
 - o Firebase Storage SDK: Để lưu trữ và quản lý avatar người dùng.

- o Firebase Realtime Database SDK: Dùng để lưu trữ thông tin tài chính cá nhân như chi tiêu, thu nhập.
- o Firebase Cloud Functions API: Sử dụng để gửi OTP xác thực qua email bằng Mailjet.
- o Firebase Token: Để xác thực và bảo mật API.
- + Sử dụng Mailjet để gửi OTP qua email.
- + Triển khai hệ thống trên di động.

MỤC LỤC

LỜI CẢM ƠN.....	
ĐỀ CƯƠNG CHI TIẾT TIỂU LUẬN CHUYÊN NGÀNH.....	
MỤC LỤC.....	
KẾ HOẠCH THỰC HIỆN.....	
PHẦN MỞ ĐẦU.....	1
1. Tính cấp thiết của đề tài.....	1
2. Mục đích của đề tài.....	1
3. Cách tiếp cận và nghiên cứu.....	2
3.1. Đối tượng nghiên cứu.....	2
3.2. Phạm vi nghiên cứu.....	2
4. Kết quả đạt được.....	3
5. Ý nghĩa khoa học và thực tiễn.....	3
PHẦN NỘI DUNG.....	4
CHƯƠNG 1 CƠ SỞ LÝ THUYẾT.....	4
1.1. React Native.....	4
1.2. Tailwind CSS.....	7
1.3. Firebase SDK.....	8
1.4. Mailjet.....	16
1.5. Expo.....	17
1.6. JWT.....	18
CHƯƠNG 2 KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU.....	20
2.1. Khảo sát hiện trạng.....	20
2.2. Phân tích yêu cầu.....	26
CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	32
3.1. Mô hình hóa yêu cầu.....	32

3.2. Kiến trúc hệ thống.....	135
3.3. Thiết kế giao diện.....	137
CHƯƠNG 4 CÀI ĐẶT VÀ KIỂM THỬ.....	167
4.1. Cài đặt ứng dụng.....	167
4.2. Kiểm thử chương trình.....	167
PHẦN KẾT LUẬN.....	174
1. Kết quả đạt được.....	174
2. Ưu điểm và nhược điểm.....	175
3. Hướng phát triển.....	176
TÀI LIỆU THAM KHẢO.....	179
BẢNG PHÂN CHIA CÔNG VIỆC.....	181
BÁO CÁO TIẾN ĐỘ.....	182

KẾ HOẠCH THỰC HIỆN

Name	Owner	Start date	Due date
Tìm hiểu thị trường	Nguyen Hong Thong Diep	2024-09-02	2024-09-07
Xác định nhu cầu người tiêu dùng	Đoàn Thái Sơn	2024-09-02	2024-09-07
Lập bảng chi tiết các công việc	Nguyen Hong Thong Diep	2024-09-08	2024-09-09
Tìm hiểu lý thuyết	Nguyen Hong Thong Diep, Đoàn Thái Sơn	2024-09-10	2024-09-16
Mô hình hóa	Nguyen Hong Thong Diep	2024-09-17	2024-09-23
Đặc tả yêu cầu bài toán	Đoàn Thái Sơn	2024-09-17	2024-09-23
Vẽ các lược đồ	Nguyen Hong Thong Diep, Đoàn Thái Sơn	2024-09-24	2024-09-30
Thiết kế Front-end	Nguyen Hong Thong Diep, Đoàn Thái Sơn	2024-10-01	2024-10-19
Xây dựng cơ sở dữ liệu	Đoàn Thái Sơn	2024-10-20	2024-10-31
Thiết kế Back-end	Đoàn Thái Sơn, Nguyen Hong Thong Diep	2024-10-01	2024-11-25
Kiểm thử ứng dụng	Đoàn Thái Sơn	2024-11-26	2024-12-06
Hoàn thiện ứng dụng	Đoàn Thái Sơn	2024-11-29	2024-12-09
Viết hướng dẫn sử dụng	Nguyen Hong Thong Diep	2024-12-10	2024-12-13
Hoàn thiện bài báo cáo	Nguyen Hong Thong Diep	2024-12-14	2024-12-20

PHẦN MỞ ĐẦU

1. Tính cấp thiết của đề tài

Sự phát triển công nghệ giúp cuộc sống ngày tiện lợi hơn, giờ đây việc mua hàng, thanh toán hay chuyển tiền có thể thực hiện chỉ trong một cái chạm. Tuy nhiên, điều vô tình lại khiến nhiều người mua sắm một cách không kiểm soát. Cụ thể hơn, 25% dân số thành thị ở Trung Quốc mắc nợ thẻ tín dụng, do nhiều người chi tiêu vượt quá khả năng tài chính, vậy nên việc quản lý tài chính là vô cùng cần thiết.

Để đáp ứng nhu cầu trên, các ứng dụng quản lý tài chính lần lượt ra mắt. Ứng dụng quản lý tài chính cung cấp các chức năng cơ bản như tạo các giao dịch và lưu lại thông tin, tiến hành thống kê và đưa ra số liệu, lập kế hoạch tiêu dùng, quản lý và kiểm soát nguồn tiền. Từ đó giúp người dùng có thể chi tiêu hợp lý, hiệu quả và có kế hoạch, tránh việc không đủ nguồn lực khi có vấn đề phát sinh, ngoài ra chúng ta còn có thể dễ dàng rà soát những giao dịch đã thực hiện, giúp tiết kiệm thời gian và công sức khi thực hiện thống kê.

Nhận thấy tầm quan trọng của việc quản lý tài chính, chúng em quyết định lựa chọn “Xây dựng ứng dụng quản lý tài chính” làm đề tài môn “Lập Trình Di Động Nâng Cao của mình với hi vọng sẽ xây dựng được một ứng dụng giúp quản lý tài chính cá nhân hiệu quả để bản thân và mọi người cùng dùng.

2. Mục đích của đề tài

Mục tiêu của đề tài là xây dựng ứng dụng có giao diện thân thiện, dễ dàng sử dụng, thao tác nhanh chóng và hiệu quả.

Nhìn chung ứng dụng của chúng em sẽ cung cấp đầy đủ các chức năng cơ bản, có thể ghi lại những chi tiêu và thu nhập, có thể tiến hành thống kê để đưa ra kế hoạch tiêu dùng hợp lý hơn, lập ra mục tiêu trong ngắn hạn và dài hạn để quản lý chặt chẽ bản thân, từ đó hoạch định tương lai cho chính mình.

3. Cách tiếp cận và nghiên cứu

3.1. Đối tượng nghiên cứu

Để nghiên cứu và xây dựng đề tài này, chúng em nghiên cứu nhu cầu sử dụng các ứng dụng quản lý tài chính của sinh viên, công nhân viên, những người đang tích lũy nguồn quỹ cá nhân cho tương lai của mình, nhằm hướng đến sự thuận tiện và hiệu quả cho tất cả người dùng. Đối tượng nghiên cứu của đề tài sẽ bao gồm:

Các chức năng cơ bản của một ứng dụng quản lý tài chính: Đăng ký, đăng nhập, tạo giao dịch, tìm kiếm thống kê các giao dịch trong một khoảng thời gian, lập mục tiêu, giới hạn tiêu dùng, ...

Công nghệ cơ sở dữ liệu thời gian thực trên đám mây của Firebase, giúp lưu trữ và quản lý dữ liệu của hệ thống một cách linh hoạt và hiệu quả.

Firebase Cloud Functions API, một dịch vụ serverless của Firebase, cho phép bạn chạy mã backend tự động để phản hồi các sự kiện được kích hoạt từ Firebase mà không cần quản lý máy chủ vật lý.

Firebase token, một mã thông báo (token) được Firebase sử dụng để xác thực và bảo mật các yêu cầu giữa ứng dụng và các dịch vụ của Firebase. Và cả Mailjet là một dịch vụ email marketing và email API dựa trên nền tảng đám mây.

3.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào các chức năng cơ bản của một ứng dụng quản lý tài chính:

Quản lý tài khoản cá nhân: Nghiên cứu và triển khai các chức năng như đăng ký, đăng nhập, quên mật khẩu, đổi mật khẩu, cài đặt thông tin cá nhân với mục tiêu là một ứng dụng có xác thực bảo mật an toàn và hiệu quả.

Quản lý thu nhập và chi tiêu: Các giao dịch sẽ được ghi lại và thống kê, nhằm giúp người dùng có thể kiểm soát nguồn tiền của mình, cũng như dễ dàng tìm lại thông tin giao dịch trong quá khứ.

Quản lý mục tiêu và hạn mức: Nghiên cứu chức năng đặt ra hạn mức tiêu dùng để kiểm soát bản thân, đặt ra những mục tiêu trong ngắn và dài hạn giúp bản thân tiết kiệm hiệu quả và đúng thời hạn, đạt được những gì đã đề ra trước đó.

4. Kết quả đạt được

Ứng dụng quản lý tài chính có:

- Giao diện
 - Có đầy đủ chức năng cơ bản, rõ ràng dễ sử dụng.
 - Giao diện có tính thẩm mỹ nhất định, phù hợp nhãn quang người dùng.
- Xử lý back-end
 - Hoàn thiện các tính năng của ứng dụng.
 - Phát triển và cải thiện các tính năng khác.
 - Cập nhật và bắt các lỗi thường gặp trong quá trình sử dụng.

5. Ý nghĩa khoa học và thực tiễn

Sau khi hoàn thiện đề tài, chúng em sẽ tiếp tục rèn luyện, tiếp thu những đóng góp của thầy cô hướng dẫn cũng như phản biện để thực hiện, triển khai mở rộng, phát triển các chức năng còn dang dở cũng như lên kế hoạch phát hành ứng dụng thực tế.

Đây vẫn còn là một đề tài mới mẻ với nhiều tiềm năng phát triển, với hi vọng ứng dụng có thể phát hành thực tế, giúp mọi người có thể sử dụng, quản lý và cải thiện bản thân, đặt ra định hướng cụ thể cho tương lai của chính mình.

PHẦN NỘI DUNG

CHƯƠNG 1 CƠ SỞ LÝ THUYẾT

1.1. React Native

1.1.1. Khái niệm^[1]

React Native là sáng kiến mã nguồn mở do Facebook khởi xướng vào năm 2015 nhằm cho phép phát triển các ứng dụng di động cho cả iOS và Android bằng JavaScript. Thay vì viết ứng dụng cho từng nền tảng bằng ngôn ngữ gốc của nền tảng đó, chẳng hạn như Swift cho iOS và Kotlin/Java cho Android, React Native cho phép viết một "ứng dụng" một lần và cùng mã nguồn đó được sử dụng trên cả hai hệ điều hành. Về cơ bản, nó được xây dựng trên React, một thư viện JavaScript phổ biến tập trung vào việc xây dựng giao diện cho người dùng và hoạt động với cùng các khái niệm lập trình như React nhưng được ánh xạ gốc vào các thành phần của hệ điều hành di động.

1.1.2. Ưu điểm và nhược điểm^[2]

- Ưu điểm:

- Hiển thị theo nền tảng cụ thể: Khác biệt với các phương pháp đa nền tảng khác như Cordova hoặc Ionic. Thay vì sử dụng JavaScript, HTML, và CSS trong chế độ xem web, React Native dịch trực tiếp thành các thành phần UI thực sự trên nền tảng, giúp tránh được nhược điểm về hiệu suất và giảm "cảm giác khác biệt" thường gặp ở các ứng dụng đa nền tảng khác.
- Giao diện người dùng gốc: Thông thường các ứng dụng phương pháp phát triển ứng dụng di động khác không có quyền truy cập vào các phần tử giao diện người dùng gốc của nền tảng máy chủ. Nên phải có găng mô phỏng

Chương 1: Cơ sở lý thuyết

những thành phần gốc này như hoạt ảnh, các chi tiết nhỏ sẽ tồn rất nhiều công sức và chúng có thể nhanh chóng trở nên lỗi thời.

- React Native thực sự chuyển mã đánh dấu của bạn thành các thành phần UI gốc: Bằng việc tận dụng các thành phần UI gốc của iOS/Android như Text, Image, ScrollView, giúp ứng dụng có giao diện trực quan và cảm giác như một ứng dụng thực tế trên mọi nền tảng.
 - Hiệu suất cao: React Native sử dụng cơ chế render DOM một lần duy nhất thành mã thực thi native, sau đó cập nhật trực tiếp vào UI thực thông qua các sự kiện. Tránh phải tái render DOM liên tục như trong trường hợp web, giúp tối ưu hiệu suất.
 - Cộng đồng người dùng lớn mạnh và tích cực: Với tính chất mã nguồn mở của nền tảng JavaScript này, mọi người đều có đóng góp và chia sẻ kiến thức của mình vào sự phát triển của framework. Người dùng có thể dễ dàng tìm kiếm sự giúp đỡ và giải quyết vấn đề một cách thuận tiện, và tất cả đều miễn phí.
 - Có thể tái sử dụng code: Phần lớn logic code được viết bằng JavaScript và React có thể chia sẻ giữa web và mobile nếu không có các thành phần đặc thù của từng nền tảng. React Native giúp các lập trình viên hoàn toàn có thể tái sử dụng lại code.
 - Trải nghiệm của nhà phát triển: Nhà phát triển React có thể chuyển kỹ năng từ Web sang di động, viết ứng dụng với hiệu suất và giao diện giống ứng dụng gốc nhanh chóng. React Native cũng cải tiến trải nghiệm phát triển và tiềm năng phát triển đa nền tảng.
 - Hỗ trợ các công nghệ mới: React Native hỗ trợ các tính năng hiện đại như React, Redux, ES6/7/8... giúp phát triển nhanh và hiệu quả.
- **Nhược điểm:**

Chương 1: Cơ sở lý thuyết

- Tính ổn định và phát triển: Đây là rủi ro lớn nhất vì React Native còn khá non trẻ, cùng với hỗ trợ còn thiếu sót trên cả iOS và Android. Tài liệu và một số tính năng vẫn chưa được cập nhật hoàn thiện và đầy đủ.
- Gỡ lỗi và quản lý mã nguồn: Khi có sự cố xảy ra, React Native tạo ra một tầng phức tạp trong quá trình gỡ lỗi và quản lý mã nguồn.. Điều này đặt ra thách thức đối với việc duy trì và mở rộng dự án.
- API và vài tính năng thiếu hỗ trợ: Một số API và tính năng trên cả hai nền tảng vẫn chưa được hỗ trợ đầy đủ. Bạn có thể tự triển khai hỗ trợ nhưng cũng sẽ làm tăng độ phức tạp của mã nguồn.

1.1.3. React Native hoạt động như thế nào? [2]

Ba luồng mà React Native sử dụng như sau: Luồng UI (Luồng chính): Luồng này liên quan đến giao diện và là cách giao diện được giao diện truy cập để hoàn thành các tác vụ liên quan đến giao diện đó. Luồng bóng: Trong luồng này, React Native chỉ tính toán bối cảnh của ứng dụng và cập nhật giao diện bất kỳ lúc nào mà không cần chạm vào Luồng UI. Luồng JavaScript: Đây là luồng nơi mã JavaScript cho ứng dụng được lưu giữ và chịu trách nhiệm thực thi (luồng thực thi). Mã sẽ chạy trong một môi trường bị cô lập - đảm bảo ứng dụng phản hồi đầy đủ.

1.1.4. Những ứng dụng viết bằng React Native

- Facebook
- Skype
- Instagram
- Airbnb
- SoundCloud Pulse
- Walmart
- Yeti Smart Home

1.2. Tailwind CSS

1.2.1. Khái niệm^[3]

Tailwind CSS là khung CSS ưu tiên tiện ích, có nghĩa là Tailwind CSS không sử dụng các lớp CSS truyền thống để xác định giao diện mà cung cấp các lớp tiện ích tích hợp sẵn mà lập trình viên có thể áp dụng trực tiếp vào các phần HTML. Điều này làm cho quá trình xây dựng giao diện linh hoạt và hiệu quả hơn.

1.2.2. Ưu điểm và nhược điểm^[4]

- Ưu điểm

- Thiết kế nhanh chóng: Tailwind CSS cho phép bạn xây dựng giao diện nhanh chóng bằng cách khai báo các lớp CSS trực tiếp trên phần tử HTML, giảm thời gian viết CSS tùy chỉnh.
- Khả năng tùy chỉnh cao: Bạn có thể dễ dàng tùy chỉnh giao diện qua tệp cấu hình và thêm các lớp mới phối hợp với lớp có sẵn.
- Hỗ trợ responsive design: Các lớp CSS cho responsive design giúp dễ dàng điều chỉnh giao diện cho các kích thước màn hình khác nhau.
- Tương thích với nhiều framework: Tailwind CSS có thể tích hợp dễ dàng với các framework front-end như React, Vue.js, Angular...
- Hỗ trợ hiệu ứng tương tác: Các lớp CSS cho hover, focus, active giúp tạo hiệu ứng tương tác dễ dàng.
- Tài liệu chi tiết: Tài liệu của Tailwind CSS rất chi tiết và dễ hiểu, hỗ trợ người dùng mới nhanh chóng làm quen với framework.

- Nhược điểm

- HTML phức tạp: Mã HTML có thể trở nên dài và phức tạp do phải khai báo nhiều lớp CSS trực tiếp trên các phần tử.
- Khó nhớ các lớp: Việc ghi nhớ và sử dụng nhiều lớp CSS có thể đòi hỏi thời gian và công sức, đặc biệt với người mới.

Chương 1: Cơ sở lý thuyết

- Phụ thuộc vào kiến thức CSS: Để sử dụng hiệu quả, bạn cần có kiến thức cơ bản về CSS. Nếu không, việc tùy chỉnh giao diện có thể gặp khó khăn.
- Tốn thời gian làm quen: Cần thời gian để làm quen và nhớ tên các lớp trong Tailwind CSS.

1.2.3. Các tính năng nổi bật^[5]

- Xây dựng giao diện chỉ cần khai báo class trên file html
- Chỉ xuất ra những file css chứa những class đã sử dụng trong giao diện.
- Dễ dàng thiết lập Responsive ngay trên file html
- Hỗ trợ tương tác hover, focus... ngay trên class html
- Tối ưu những cấu trúc html css có tính lặp lại
- Định nghĩa thêm class mới được phối hợp bởi các class có sẵn trong tailwind
- Hỗ trợ chế độ tối Dark Mode
- Dễ dàng mở rộng, chỉnh sửa và biến đổi
- Tăng tốc độ code với Extension thông minh trên VS Code.

1.3. Firebase SDK

1.3.1. Khái niệm^[6]

Firebase SDK là bộ công cụ phát triển phần mềm do Google cung cấp, cho phép các nhà phát triển dễ dàng xây dựng các ứng dụng với cơ sở dữ liệu, lưu trữ tệp, xác thực, v.v. mà không cần phải viết phần phụ trợ từ đầu. Firebase giúp phát triển các ứng dụng đa nền tảng một cách nhanh chóng và hiệu quả, đặc biệt là các ứng dụng di động.

1.3.2. Ưu điểm và nhược điểm^[7]

- **Ưu điểm**

Chương 1: Cơ sở lý thuyết

- Cơ sở dữ liệu đáng tin cậy và có thể mở rộng: Cơ sở dữ liệu Firebase của Google cung cấp các tính năng mạnh mẽ để phát triển ứng dụng web và thiết bị di động. Một số dịch vụ Firebase phổ biến bao gồm Cơ sở dữ liệu thời gian thực, Lưu trữ đám mây, Crashlytics và Analytics cũng như các tính năng như đăng nhập một lần.
- Lưu trữ nhanh chóng và an toàn: Một tính năng thú vị của Firebase là dịch vụ lưu trữ nhanh chóng và an toàn. Firebase hỗ trợ tất cả các loại nội dung, bao gồm ứng dụng web, tài liệu động và tĩnh. Ngoài ra, cho dù bạn muốn lưu trữ các vi dịch vụ Express.js, HTML, CSS hay API, Firebase đều sẵn sàng lưu trữ. Điều này cho thấy Firebase hỗ trợ lưu trữ nhiều loại nội dung khác nhau.
- Dịch vụ thử nghiệm tốt nhất: Một ưu điểm khác của Firebase là dịch vụ thử nghiệm và do đó nhiều nhà phát triển thích sử dụng nền tảng phát triển phần mềm này. Firebase cung cấp cho người dùng nhiều dịch vụ thử nghiệm khác nhau chứ không chỉ một loại.
- Xác thực Firebase: Tâm quan trọng của danh tính người dùng trong bảo mật ứng dụng là không thể phủ nhận. Trong bối cảnh bảo mật trực tuyến hiện tại, một trong những tính năng hấp dẫn nhất của nền tảng là dịch vụ Xác thực Firebase do Google cung cấp. Nó cung cấp SDK, khung giao diện người dùng và dịch vụ máy chủ để tích hợp để xác thực người dùng trước khi sử dụng một ứng dụng Firebase cụ thể. Xác thực Firebase không chỉ chấp nhận email, mật khẩu và số điện thoại mà còn hỗ trợ các nguồn nhận dạng liên kết.
- Google Analytics: Google Analytics có sẵn cho môi trường iOS, Android, Web, C++ và Unity. Công cụ phân tích miễn phí này cung cấp cho nhà phát triển thông tin về cách người dùng tương tác với ứng dụng web và thiết bị di động của họ. Firebase Analytics cũng có thể giúp cải thiện khả năng giữ chân và tương tác của người dùng trong ứng dụng của bạn.

Chương 1: Cơ sở lý thuyết

- Nhược điểm

- Lưu trữ dữ liệu bất tiện: Thông thường, bạn sẽ sử dụng cơ sở dữ liệu trực tiếp làm bộ lưu trữ chính, đây không phải lúc nào cũng là lựa chọn tốt. Vấn đề cơ bản là khả năng truy vấn hạn chế. Bạn không thể truy vấn nhiều khóa cùng một lúc và không có cách nào để lọc dữ liệu. Toàn bộ cơ sở dữ liệu là một tệp JSON, khác xa với định dạng lưu trữ của SQL. Định dạng này cũng loại bỏ khả năng mô hình hóa dữ liệu.
- Giảm khả năng di chuyển dữ liệu: Với sự phổ biến ngày càng tăng của các dịch vụ đám mây như Firebase, các nhà phát triển cần lưu ý về cách dữ liệu của họ sẽ được chuyển giữa các nền tảng. Nếu người dùng không có quyền truy cập hoặc kiểm soát dễ dàng trong việc cập nhật một số phần trong ứng dụng do backend được lưu trữ bên ngoài, thì có thể sẽ xảy ra các vấn đề và không có cách nào giải quyết ngoài việc bắt đầu lại từ đầu, điều mà nhiều người không mong muốn. Firebase Authentication SDK là một dịch vụ của Firebase cung cấp giải pháp dễ dàng và bảo mật cho việc xác thực người dùng trong các ứng dụng di động và web. Nó hỗ trợ nhiều phương pháp đăng nhập như email, số điện thoại, và các phương thức đăng nhập xã hội như Google, Facebook, và Twitter. SDK này giúp nhà phát triển tích hợp các hệ thống xác thực mà không cần phải xây dựng backend phức tạp từ đầu.

1.3.3. Firebase Authentication SDK.^[8]

1.3.3.1. Khái niệm

1.3.3.2. Ưu điểm và nhược điểm

- Ưu điểm

- Dễ triển khai: Firebase Authentication có SDK dễ dàng tích hợp với nhiều nền tảng khác nhau (iOS, Android, Web), giúp tiết kiệm thời gian phát triển.

Chương 1: Cơ sở lý thuyết

- Bảo mật cao: Được hỗ trợ bởi Google, Firebase Authentication cung cấp độ tin cậy cao và bảo mật mạnh mẽ cho việc xác thực người dùng.
 - Hỗ trợ nhiều phương thức xác thực: Firebase cung cấp các tùy chọn đăng nhập linh hoạt bao gồm email, số điện thoại, và các tài khoản mạng xã hội phổ biến như Google, Facebook, và Twitter.
- **Nhược điểm**
- Giới hạn tùy chỉnh: Firebase Authentication không cung cấp nhiều tùy chỉnh sâu cho các yêu cầu xác thực đặc biệt.
 - Phụ thuộc vào hạ tầng Firebase: Khi sử dụng Firebase Authentication, bạn sẽ phải phụ thuộc vào hệ sinh thái Firebase, và việc thay đổi hệ thống sau này có thể khó khăn.

1.3.3.3. Firebase Authentication hoạt động như thế nào?

Firebase Authentication hoạt động dựa trên việc cung cấp SDK cho các nền tảng khác nhau (iOS, Android, Web) để dễ dàng tích hợp vào ứng dụng. Khi người dùng thực hiện đăng nhập, Firebase sẽ xác thực thông tin đăng nhập (email, mật khẩu, số điện thoại, hoặc tài khoản mạng xã hội). Sau khi xác thực thành công, Firebase tạo ra một token JWT (JSON Web Token) để đại diện cho phiên làm việc của người dùng. Token này sẽ được sử dụng trong các yêu cầu tiếp theo đến server nhằm xác thực người dùng và bảo mật ứng dụng.

1.3.4. Firebase Storage SDK.^[8]

1.3.4.1. Khái niệm

Firebase Storage SDK là một dịch vụ của Firebase cung cấp khả năng lưu trữ và quản lý dữ liệu tệp (file) trên đám mây, bao gồm hình ảnh, video, và các loại tệp khác. Nó được tích hợp chặt chẽ với Firebase Authentication để đảm bảo rằng chỉ người dùng được ủy quyền mới có thể truy cập vào dữ liệu. Firebase Storage cũng hỗ trợ chia sẻ dữ liệu giữa các ứng dụng và đồng bộ hóa dữ liệu trên nhiều thiết bị.

Chương 1: Cơ sở lý thuyết

1.3.4.2. Ưu điểm và nhược điểm

- Ưu điểm

- Dễ dàng tích hợp với Firebase Authentication: Firebase Storage SDK tích hợp tốt với Firebase Authentication, đảm bảo rằng dữ liệu chỉ được truy cập bởi người dùng đã được xác thực.
- Hỗ trợ tệp lớn: Firebase Storage được tối ưu hóa để lưu trữ các tệp lớn như hình ảnh và video, đồng thời cho phép truy cập và chia sẻ dễ dàng từ các ứng dụng di động và web.
- Tích hợp đa nền tảng: Firebase Storage hoạt động tốt trên nhiều nền tảng khác nhau như iOS, Android, và Web, giúp việc triển khai và quản lý dữ liệu trở nên đơn giản hơn.

- Nhược điểm

- Chi phí tăng theo quy mô dữ liệu: Khi lượng dữ liệu lưu trữ tăng lên, chi phí dịch vụ Firebase Storage cũng sẽ tăng theo, điều này có thể là một vấn đề đối với các dự án có quy mô lớn.
- Khả năng quản lý tệp và truy vấn dữ liệu phức tạp bị hạn chế: Firebase Storage không cung cấp các tính năng truy vấn dữ liệu phức tạp như trong các hệ thống cơ sở dữ liệu truyền thống, điều này có thể gây khó khăn khi quản lý lượng lớn tệp hoặc dữ liệu có cấu trúc phức tạp.

1.3.4.3. Firebase Storage hoạt động như thế nào?

Firebase Storage SDK hoạt động bằng cách cho phép người dùng tải lên và tải xuống các tệp từ ứng dụng di động hoặc web lên đám mây của Firebase. Dữ liệu tệp được lưu trữ trên Google Cloud Storage, đảm bảo độ bền cao và khả năng truy cập nhanh. Firebase Storage tích hợp với Firebase Authentication để quản lý quyền truy cập, đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập và tương tác với dữ liệu.

1.3.5. Firebase Realtime Database SDK.^[8]

1.3.5.1. Khái niệm

Chương 1: Cơ sở lý thuyết

Firebase Realtime Database là một cơ sở dữ liệu NoSQL hoạt động theo thời gian thực, cho phép lưu trữ và đồng bộ dữ liệu ngay lập tức giữa các thiết bị khách hàng qua kết nối internet. Dữ liệu được lưu trữ dưới dạng JSON và có thể được đồng bộ hóa với tất cả người dùng trong thời gian thực. Nó thường được sử dụng cho các ứng dụng như chat, bảng tin, hoặc những ứng dụng yêu cầu cập nhật dữ liệu ngay lập tức mà không cần làm mới trang.

1.3.5.2. **Ưu điểm và nhược điểm**

- **Ưu điểm**

- **Đồng bộ hóa thời gian thực:** Firebase Realtime Database đồng bộ dữ liệu ngay lập tức với tất cả các thiết bị kết nối, phù hợp cho các ứng dụng yêu cầu thông tin được cập nhật liên tục và nhanh chóng.
- **Hoạt động ngoại tuyến:** Dữ liệu được lưu trữ cục bộ khi không có kết nối internet và sẽ được đồng bộ hóa trở lại khi kết nối mạng được khôi phục. Điều này làm tăng trải nghiệm người dùng trong các ứng dụng di động có thể mất kết nối mạng thường xuyên.
- **Dễ sử dụng:** Firebase Realtime Database dễ tích hợp với các ứng dụng di động và web nhờ SDK đa nền tảng.

- **Nhược điểm**

- **Giới hạn về truy vấn dữ liệu:** Realtime Database không hỗ trợ các truy vấn phức tạp. Bạn chỉ có thể truy vấn dữ liệu dựa trên một khóa duy nhất và thiếu các tính năng nâng cao như trong SQL.
- **Không phù hợp cho các ứng dụng phức tạp:** Do cơ sở dữ liệu chỉ lưu trữ dưới dạng JSON và thiếu khả năng truy vấn nâng cao, Firebase Realtime Database không phù hợp với các ứng dụng yêu cầu quản lý dữ liệu phức tạp hoặc có cấu trúc dữ liệu quan hệ.

1.3.5.3. **Firebase Realtime Database hoạt động như thế nào?**

Chương 1: Cơ sở lý thuyết

Firebase Realtime Database hoạt động bằng cách lưu trữ dữ liệu dưới dạng JSON và đồng bộ hóa ngay lập tức với tất cả người dùng kết nối. Khi có thay đổi ở một phần dữ liệu, mọi thiết bị đang kết nối với cơ sở dữ liệu sẽ nhận được dữ liệu mới mà không cần làm mới. Mọi dữ liệu đều có thể được đồng bộ hóa, kể cả khi ứng dụng đang hoạt động ngoại tuyến, và sẽ được cập nhật lại khi kết nối mạng được khôi phục.

1.3.6. Firebase Cloud Functions API.^[8]

1.3.6.1. Khái niệm

Firebase Cloud Functions là một dịch vụ serverless của Firebase cho phép bạn thực thi mã backend (mã phía máy chủ) dựa trên các sự kiện được kích hoạt từ các dịch vụ của Firebase (như Firebase Authentication, Firestore, hoặc Realtime Database) hoặc các yêu cầu HTTP. Nó loại bỏ nhu cầu quản lý máy chủ riêng và tự động mở rộng dựa trên lưu lượng yêu cầu.

1.3.6.2. Ưu điểm và nhược điểm

- Ưu điểm

- Serverless (không cần quản lý máy chủ): Bạn không cần thiết lập hoặc quản lý máy chủ, Google Cloud tự động đảm nhận việc mở rộng và quản lý tài nguyên.
- Tích hợp sâu với các dịch vụ khác của Firebase: Cloud Functions có thể dễ dàng được kích hoạt từ các sự kiện của Firebase, chẳng hạn như khi có người dùng mới đăng nhập, thay đổi dữ liệu trong cơ sở dữ liệu, hoặc lưu trữ tệp mới trên Firebase Storage.
- Mở rộng tự động: Firebase Cloud Functions tự động mở rộng theo yêu cầu mà không cần bạn phải can thiệp.

- Nhược điểm:

- Độ trễ: Do cần phải khởi động một môi trường mới cho mỗi yêu cầu, nên có thể có độ trễ trong một số trường hợp, đặc biệt khi lần đầu chạy một function mới.

Chương 1: Cơ sở lý thuyết

- Phức tạp hơn với các ứng dụng lớn: Khi số lượng functions tăng lên, việc quản lý và duy trì toàn bộ hệ thống có thể trở nên phức tạp hơn, đặc biệt với các ứng dụng có nhiều yêu cầu backend.

1.3.6.3. Firebase Authentication hoạt động như thế nào?

Firebase Cloud Functions thường được tích hợp với Firebase Authentication để thực thi các tác vụ dựa trên sự kiện đăng nhập của người dùng. Ví dụ, khi người dùng mới đăng ký hoặc đăng nhập, một Cloud Function có thể được kích hoạt để gửi email chào mừng, xác minh tài khoản hoặc thực hiện các hành động backend khác. Firebase Authentication cung cấp mã thông báo JWT (JSON Web Token) đại diện cho người dùng đã xác thực, và Cloud Functions có thể sử dụng mã này để xác thực và xử lý các yêu cầu từ người dùng một cách an toàn.

1.3.7. Firebase Token.^[8]

1.3.7.1. Khái niệm

Firebase Token là một chuỗi ký tự mã hóa được Firebase sử dụng để xác thực và ủy quyền người dùng trong hệ thống. Nó giúp các ứng dụng xác minh danh tính người dùng và đảm bảo quyền truy cập vào các dịch vụ Firebase như Realtime Database, Cloud Firestore, hoặc Cloud Functions.

Firebase Token thường được phát hành sau khi người dùng đăng nhập qua Firebase Authentication, sử dụng các phương thức như Email/Password, Google Sign-In, Facebook, hoặc Anonymous Login. Token có thời gian sống cố định và có thể được làm mới nếu hết hạn.

1.3.7.2. Ưu điểm và nhược điểm

- Ưu điểm

- Bảo mật: Được mã hóa và xác thực để đảm bảo an toàn.
- Tích hợp Firebase: Hỗ trợ các dịch vụ như Realtime Database, Firestore.

Chương 1: Cơ sở lý thuyết

- Đơn giản: Quản lý xác thực dễ dàng thông qua các SDK của Firebase.
- **Nhược điểm:**
 - Thời gian sống cố định: Token phải được làm mới nếu hết hạn.
 - Phụ thuộc Firebase: Khó tích hợp ngoài hệ sinh thái Firebase.
 - Không thu hồi tức thời: Không thể vô hiệu hóa token ngay lập tức nếu bị lộ.

1.3.7.3. Firebase Token hoạt động như thế nào?

Khi người dùng đăng nhập, Firebase tạo một Token (dựa trên JWT) chứa thông tin người dùng và gửi về ứng dụng client. Token này được sử dụng để xác thực trong các yêu cầu tới Firebase, như Firestore hoặc Realtime Database, bằng cách đính kèm trong HTTP Header. Firebase kiểm tra tính hợp lệ của Token để cấp quyền truy cập hoặc từ chối. Token có thời gian sống cố định và sẽ tự động làm mới khi gần hết hạn.

1.4. Mailjet

1.4.1. Khái niệm

Mailjet là một nền tảng dịch vụ email marketing và email giao dịch trên đám mây, cho phép doanh nghiệp gửi và quản lý các chiến dịch email, thông báo và giao dịch như xác nhận đơn hàng hay OTP. Mailjet cung cấp API mạnh mẽ, dễ dàng tích hợp vào các ứng dụng web hoặc di động để gửi email tự động và quản lý quy trình email hiệu quả.

1.4.2. Ưu điểm và nhược điểm

- Ưu điểm

- Dễ sử dụng: Giao diện đơn giản, dễ dàng cho cả những người không có kinh nghiệm kỹ thuật để quản lý các chiến dịch email.
- Hỗ trợ API mạnh mẽ: Tích hợp dễ dàng vào ứng dụng qua API RESTful, phù hợp cho các nhà phát triển sử dụng email giao dịch.
- Báo cáo và phân tích thời gian thực: Mailjet cung cấp công cụ phân tích hiệu quả, giúp theo dõi hiệu suất của các chiến dịch email như tỷ lệ mở, nhấp chuột, và tương tác.

Chương 1: Cơ sở lý thuyết

- Tối ưu chi phí: Phù hợp cho các doanh nghiệp nhỏ và vừa nhờ vào mức giá hợp lý.
- **Nhược điểm:**
 - Giới hạn phân đoạn người dùng: Mailjet có các tùy chọn phân đoạn người dùng ít linh hoạt hơn so với các đối thủ cạnh tranh.
 - Chức năng A/B testing cơ bản: Dù có hỗ trợ A/B testing, nhưng các công cụ này còn hạn chế nếu so sánh với các nền tảng cao cấp khác.

1.4.3. Cách hoạt động

Mailjet hoạt động dựa trên việc tích hợp API vào ứng dụng hoặc sử dụng giao diện trực quan để tạo, quản lý và gửi email. Khi một email được gửi, Mailjet sẽ xử lý và phân phối email tới người nhận, đồng thời theo dõi các chỉ số như tỷ lệ mở email, tỷ lệ nhấp chuột và báo cáo chi tiết. Hệ thống có khả năng tự động gửi email giao dịch như OTP hoặc xác nhận đơn hàng, cũng như quản lý các chiến dịch marketing tự động với các tính năng như A/B testing và phân tích thời gian thực.

1.5. Expo

1.5.1. Khái niệm

Expo là một framework phát triển ứng dụng di động dựa trên React Native, giúp các nhà phát triển dễ dàng xây dựng và triển khai ứng dụng mà không cần cài đặt cấu hình phức tạp. Nó cung cấp các công cụ và thư viện sẵn có, giúp quá trình phát triển trở nên nhanh chóng và đơn giản hơn, đặc biệt là cho các ứng dụng đa nền tảng như iOS và Android.

1.5.2. Ưu điểm và nhược điểm

- Ưu điểm

- Dễ dàng cài đặt và sử dụng: Expo cung cấp môi trường phát triển đơn giản với các công cụ tích hợp sẵn như Expo Go, cho phép nhà phát triển xem và

Chương 1: Cơ sở lý thuyết

thử nghiệm ứng dụng ngay lập tức trên thiết bị di động mà không cần cài hình phức tạp.

- **Hỗ trợ đa nền tảng:** Với Expo, bạn có thể phát triển ứng dụng cho cả iOS và Android mà không cần viết mã riêng cho từng nền tảng, tiết kiệm thời gian và công sức.

- **Cộng đồng và thư viện phong phú:** Expo cung cấp một bộ thư viện phong phú được tích hợp sẵn, giúp nhà phát triển dễ dàng thêm các tính năng mà không cần phải tìm kiếm hoặc cấu hình thêm các thư viện bên ngoài.

- Nhược điểm

- **Hạn chế về quyền kiểm soát:** Expo không cho phép truy cập vào tất cả các API gốc của thiết bị, và đôi khi bạn sẽ gặp khó khăn nếu cần tích hợp các tính năng phức tạp, dẫn đến việc cần tách khỏi Expo để sử dụng React Native CLI.
- **Kích thước ứng dụng lớn hơn:** Do các thư viện và công cụ được tích hợp sẵn, ứng dụng phát triển bằng Expo thường có kích thước lớn hơn so với những ứng dụng sử dụng React Native CLI.
- **Phụ thuộc vào Expo:** Việc cập nhật và thêm các tính năng mới phụ thuộc nhiều vào Expo. Nếu Expo chưa hỗ trợ một tính năng hoặc cập nhật, bạn có thể phải chờ đợi lâu để có thể sử dụng nó.

1.6. JWT

1.6.1. Khái niệm

JWT (JSON Web Token) là một tiêu chuẩn mở (RFC 7519) được sử dụng để chia sẻ thông tin bảo mật giữa các bên dưới dạng JSON. JWT thường được sử dụng trong các hệ thống xác thực và ủy quyền, nơi thông tin người dùng được mã hóa dưới dạng một token để xác định danh tính hoặc cung cấp quyền truy cập vào các tài nguyên.

JWT bao gồm ba phần chính:

Chương 1: Cơ sở lý thuyết

1. Header: Chứa thông tin về thuật toán mã hóa và loại token.
2. Payload: Chứa dữ liệu
3. Signature: Phần mã hóa của token dùng để đảm bảo tính toàn vẹn của dữ liệu.

Token này thường được truyền qua HTTP Header (Authorization: Bearer <token>) hoặc lưu trữ trong cookie, local storage.

1.6.2. Ưu điểm và nhược điểm

- Ưu điểm

- Dễ dàng sử dụng và phổ biến.
- Tự chứa thông tin.
- Độc lập nền tảng.
- An toàn khi sử dụng đúng cách.
- Hiệu suất cao.

- Nhược điểm

- Không thể thu hồi.
- Kích thước lớn.
- Thời gian sống cố định.
- Yêu cầu bảo mật cao.
- Khó quản lý phiên dài hạn.

CHƯƠNG 2 KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU

2.1. Khảo sát hiện trạng

2.1.1. Tìm hiểu các ứng dụng tương tự với đề tài

2.1.1.1. Simplifi (by Quicken - hơn 20 triệu người dùng)

- Nhà phát triển

Ứng dụng Simplifi được phát triển bởi Quicken Inc., một công ty lâu đời trong lĩnh vực phần mềm quản lý tài chính cá nhân. Quicken Inc. nổi tiếng với các sản phẩm như Quicken và Simplifi, mang đến các giải pháp tài chính toàn diện cho cá nhân và doanh nghiệp nhỏ. Quicken có nhiều năm kinh nghiệm trong việc phát triển phần mềm tài chính với mục tiêu giúp người dùng kiểm soát tài chính dễ dàng và hiệu quả.

- Ứng dụng

Simplifi là một ứng dụng quản lý tài chính cá nhân hiện đại, được thiết kế nhằm giúp người dùng theo dõi chi tiêu, lập kế hoạch tài chính, và đạt được các mục tiêu tiết kiệm. Simplifi cung cấp một bảng điều khiển trực quan để dễ dàng theo dõi các giao dịch tài chính, tạo và quản lý ngân sách một cách hiệu quả. Ứng dụng này dành cho những người dùng trẻ tuổi hoặc có nhu cầu theo dõi tài chính đơn giản nhưng đầy đủ tính năng.

- Ưu điểm:

- Trải nghiệm người dùng xuất sắc: Giao diện thân thiện với bảng điều khiển dễ sử dụng.
- Quản lý giao dịch tốt: Simplifi cung cấp công cụ quản lý giao dịch chi tiết và rõ ràng.

Chương 2: Khảo sát và xác định yêu cầu

- Công cụ lập ngân sách hiệu quả: Ứng dụng cung cấp các công cụ lập ngân sách linh hoạt và hiệu quả.
- Báo cáo tài chính rõ ràng: Người dùng có thể dễ dàng truy cập các báo cáo chi tiết về tình hình tài chính của mình.
- Ứng dụng di động tuyệt vời: Có sẵn trên cả iOS và Android với trải nghiệm người dùng tốt.

- **Nhược điểm:**

- Phí đăng ký hàng tháng: Simplifi yêu cầu trả phí hàng tháng hoặc hàng năm để sử dụng.
- Mục tiêu tiết kiệm không liên kết với tài khoản trực tiếp: Mục tiêu tiết kiệm không tự động kết nối với tài khoản ngân hàng trực tiếp.

- **Đối tượng hướng đến:**

Simplifi được thiết kế cho những người trẻ tuổi hoặc những người dùng có nhu cầu đơn giản trong việc theo dõi tài chính cá nhân. Họ muốn quản lý tài khoản tài chính hàng ngày mà không cần phải phân tích quá nhiều biểu đồ hay số liệu phức tạp.

- **Thông số kỹ thuật và cấu hình:**

- Free Version: Không
- Free Credit Report/Score: Có
- Web Interface: Có
- Income/Expense Tracking: Có
- iOS App: Có
- Android App: Có

Chương 2: Khảo sát và xác định yêu cầu

2.1.1.2. YNAB (You Need A Budget) (Hơn 1 triệu lượt tải trên Google Play)

- Thông tin về nhà phát triển

Nhà phát triển của ứng dụng YNAB (You Need A Budget) là YNAB LLC. YNAB LLC tập trung vào việc cung cấp các giải pháp phần mềm quản lý tài chính cá nhân, với triết lý giúp người dùng thay đổi cách họ quản lý tiền bạc. YNAB không chỉ là một công cụ, mà còn là một phương pháp giáo dục người dùng để họ hiểu và cải thiện tài chính của mình theo cách có hệ thống và có chiến lược.

- Thông tin ứng dụng

YNAB (You Need A Budget) là một ứng dụng quản lý tài chính cá nhân được thiết kế để giúp người dùng lập ngân sách hiệu quả. Ứng dụng giúp người dùng theo dõi thu nhập, chi tiêu, và hướng dẫn học cách phân bổ tài chính sao cho tối ưu nhất để đạt được các mục tiêu tài chính dài hạn.

- Ưu điểm:

- Hiệu quả lập ngân sách: Dựa trên các ý tưởng lập ngân sách mạnh mẽ, giúp người dùng kiểm soát chi tiêu một cách hợp lý.
- Linh hoạt: Cho phép điều chỉnh ngân sách để phù hợp với các mục tiêu cá nhân, tăng cơ hội thành công tài chính.
- Tài liệu hỗ trợ phong phú: Cung cấp nhiều hướng dẫn, tài liệu hỗ trợ và các công cụ học tập hữu ích.
- Chia sẻ tài khoản: Hỗ trợ chia sẻ tài khoản với tối đa 6 người, phù hợp cho việc quản lý tài chính gia đình hoặc đối tác.
- Giao diện thân thiện: Dễ sử dụng, trực quan.

- Nhược điểm:

- Yêu cầu thời gian và cam kết: Người dùng cần đầu tư thời gian và cam kết để hiểu và sử dụng ứng dụng một cách hiệu quả.

Chương 2: Khảo sát và xác định yêu cầu

- Không hỗ trợ theo dõi đầu tư và thanh toán hóa đơn: YNAB tập trung vào lập ngân sách, không có chức năng theo dõi đầu tư hoặc thanh toán hóa đơn trực tiếp.
- Đường cong học tập cao: Ứng dụng có một mức độ khó khăn ban đầu, yêu cầu người dùng học cách áp dụng triết lý lập ngân sách của YNAB.

- **Đối tượng hướng đến**

YNAB dành cho những người muốn nhiều hơn từ một ứng dụng tài chính cá nhân, đặc biệt là những người có mục tiêu dài hạn trong việc quản lý chi tiêu. Ứng dụng này phù hợp với những ai sẵn sàng dành thời gian để tìm hiểu và thay đổi cách họ tương tác với tiền bạc, từ nhân viên văn phòng cho đến người tự kinh doanh.

- **Thông số kỹ thuật và cấu hình:**

- Free Version: Không
- Free Credit Report/Score: Không
- Web Interface: Có
- Income/Expense Tracking: Có
- iOS App: Có
- Android App: Có

2.1.1.3. Greenlight (6 triệu người dùng trên App Store)

- **Thông tin về nhà phát triển**

Ứng dụng Greenlight được phát triển bởi Greenlight Financial Technology, Inc., một công ty công nghệ tài chính tập trung vào việc cung cấp các giải pháp tài chính cho gia đình và trẻ em. Công ty cam kết hỗ trợ các gia đình dạy trẻ em cách quản lý tiền bạc một cách có trách nhiệm và an toàn thông qua việc cung cấp các công cụ như ứng dụng quản lý tài chính và thẻ ghi nợ.

Chương 2: Khảo sát và xác định yêu cầu

- Thông tin ứng dụng

Greenlight là một ứng dụng tài chính dành cho trẻ em, được thiết kế để giúp các bậc phụ huynh hướng dẫn con em mình về quản lý tiền bạc. Ứng dụng này cung cấp các tính năng giúp trẻ em học cách tiết kiệm, chi tiêu, đầu tư và xây dựng thói quen tài chính tốt từ khi còn nhỏ. Bằng cách sử dụng ứng dụng và thẻ ghi nợ Greenlight, trẻ em có thể thực hành quản lý tiền dưới sự giám sát của người lớn.

- Ưu điểm:

- Kiểm soát tuyệt vời cho người giám sát: Người lớn có thể theo dõi hoạt động tài chính của con mình và kiểm soát cách sử dụng tiền.
- Hỗ trợ tiết kiệm và đầu tư: Trẻ em có thể học cách tiết kiệm và đầu tư từ sớm thông qua ứng dụng.
- Tính năng an toàn gia đình: Ứng dụng cung cấp các tính năng bảo vệ để đảm bảo an toàn tài chính cho gia đình.
- Hỗ trợ khách hàng 24/7: Có dịch vụ hỗ trợ khách hàng liên tục để giải đáp mọi thắc mắc của người dùng.

- Nhược điểm:

Phí hàng tháng cao cho cấp độ Infinity: Dịch vụ yêu cầu phí đăng ký hàng tháng, bắt đầu từ \$5.99 và tăng cao hơn đôi với cấp độ Infinity. (Greenlight Infinity: Bao gồm tất cả các tính năng của Greenlight Max với lãi suất tiết kiệm 5%², chia sẻ vị trí, phát hiện va chạm, và nhiều tính năng khác (14,98 USD/tháng).)

- Các gói của Greenlight:

- Greenlight Core: Thẻ ghi nợ và ứng dụng giáo dục dành cho trẻ em và thanh thiếu niên để kiếm tiền, tiết kiệm, chi tiêu và cho tặng — cộng với lãi suất 2% trên tiền tiết kiệm² (5,99 USD/tháng).

Chương 2: Khảo sát và xác định yêu cầu

- Greenlight Max: Bao gồm tất cả các tính năng của Greenlight Core với 1% hoàn tiền khi mua sắm, lãi suất 3% trên tiền tiết kiệm², các gói bảo vệ³, và nhiều hơn nữa (9,98 USD/tháng).
- Greenlight Infinity: Bao gồm tất cả các tính năng của Greenlight Max với lãi suất 5% trên tiền tiết kiệm², chia sẻ vị trí, phát hiện va chạm⁴, và nhiều tính năng khác (14,98 USD/tháng).

2.1.2. Yêu cầu đặt ra cho dự án

Dự án xây dựng ứng dụng quản lý tài chính phổ thông được thiết kế với mục tiêu cung cấp một công cụ hữu ích, dễ sử dụng và phù hợp với đa dạng người dùng, từ cá nhân, gia đình đến cả doanh nghiệp nhỏ. Trong bối cảnh tài chính cá nhân ngày càng trở nên phức tạp với nhiều khoản thu nhập, chi tiêu và các quyết định tài chính phức tạp, nhu cầu có một ứng dụng thân thiện, dễ sử dụng nhưng mạnh mẽ về tính năng trở thành yếu tố quan trọng.

Ứng dụng cần phải đáp ứng được các tiêu chuẩn sau:

- Phổ thông và dễ tiếp cận: Ứng dụng cần được thiết kế sao cho có thể sử dụng dễ dàng bởi người dùng thuộc mọi đối tượng, từ những người không am hiểu về công nghệ đến những người dùng thành thạo. Yếu tố dễ sử dụng, không phức tạp là yếu tố hàng đầu, đảm bảo tất cả người dùng có thể nắm bắt nhanh chóng ngay từ lần sử dụng đầu tiên.
- Giao diện thân thiện và trực quan: Một giao diện người dùng (UI) thân thiện, dễ hiểu, nhưng phải đủ tinh tế và đẹp mắt, mang lại trải nghiệm người dùng (UX) tốt. Giao diện phải trực quan để người dùng có thể thực hiện các thao tác cơ bản như ghi nhận thu nhập, chi tiêu, lập kế hoạch tài chính mà không cần phải tra cứu hay tham khảo hướng dẫn quá nhiều.
- Tính năng toàn diện và linh hoạt: Ứng dụng phải có khả năng quản lý tài chính cá nhân toàn diện, từ việc theo dõi thu nhập, chi tiêu đến lập kế hoạch

Chương 2: Khảo sát và xác định yêu cầu

tiết kiệm và đầu tư. Tính năng phải linh hoạt, cho phép người dùng tùy chỉnh theo nhu cầu và mục tiêu tài chính cá nhân của họ.

- An toàn và bảo mật thông tin: Dữ liệu tài chính cá nhân cần được bảo vệ tuyệt đối. Ứng dụng phải có các tính năng bảo mật mạnh mẽ như mã hóa dữ liệu, bảo mật hai lớp (two-factor authentication), đồng thời tuân thủ các quy định bảo mật quốc tế. Điều này đảm bảo thông tin cá nhân và tài chính của người dùng không bị lộ ra ngoài hoặc bị truy cập trái phép.
- Khả năng mở rộng và tích hợp: Ứng dụng cần có khả năng mở rộng trong tương lai, không chỉ trong việc tích hợp thêm các tính năng mới mà còn trong việc đồng bộ hóa với các nền tảng tài chính khác như ngân hàng, ví điện tử, hay các dịch vụ tài chính của bên thứ ba. Việc tích hợp giúp ứng dụng dễ dàng phù hợp với nhiều hệ sinh thái tài chính khác nhau.
- Tương thích đa nền tảng: Để đảm bảo tính tiện lợi, ứng dụng cần hỗ trợ trên nhiều nền tảng bao gồm iOS, Android và phiên bản web. Người dùng có thể quản lý tài chính từ nhiều thiết bị khác nhau mà không gặp rào cản nào về mặt công nghệ.

2.2. Phân tích yêu cầu

2.2.1. Yêu cầu chức năng

2.2.1.1. Đối với người dùng:

- Quản lý tài khoản tài chính cá nhân:

- Người dùng có thể tạo và quản lý nhiều loại tài khoản khác nhau như tài khoản ngân hàng, tài khoản tiết kiệm, ví điện tử, thẻ tín dụng. Tính năng này cho phép đồng bộ hóa giao dịch từ các tài khoản liên kết, từ đó tự động cập nhật dữ liệu chi tiêu và thu nhập.

Chương 2: Khảo sát và xác định yêu cầu

- Cho phép người dùng theo dõi tình hình tài chính theo từng tài khoản, giúp dễ dàng quản lý tổng quan tài chính.
- **Ghi nhận thu nhập và chi tiêu:**
- Người dùng có thể nhận thu nhập và chi tiêu hàng ngày một cách nhanh chóng và dễ dàng. Các khoản chi tiêu sẽ được phân loại theo danh mục như thực phẩm, giải trí, giáo dục, di chuyển, và chi phí cố định (hóa đơn điện, nước...).
 - Hỗ trợ tính năng nhận diện giao dịch tự động qua liên kết ngân hàng hoặc ví điện tử, từ đó tự động ghi lại các giao dịch phát sinh.
- **Lập kế hoạch và quản lý ngân sách:**
- Cung cấp công cụ giúp người dùng lập ngân sách hàng tháng hoặc theo chu kỳ khác, giúp theo dõi và điều chỉnh việc chi tiêu sao cho không vượt quá hạn mức đã đặt ra.
 - Hỗ trợ tính năng thông báo khi sắp hết ngân sách hoặc khi chi tiêu vượt quá hạn mức cho phép.
- **Thiết lập và theo dõi mục tiêu tài chính:**
- Người dùng có thể thiết lập các mục tiêu tài chính cụ thể như tiết kiệm cho một khoản mua sắm lớn, du lịch, hoặc đầu tư dài hạn.
 - Ứng dụng cung cấp biểu đồ trực quan, theo dõi tiến độ mục tiêu theo thời gian thực, giúp người dùng điều chỉnh kế hoạch tiết kiệm và đầu tư nếu cần.
- **Báo cáo và phân tích tài chính:**
- Ứng dụng sẽ cung cấp báo cáo chi tiêu chi tiết theo tuần, tháng, hoặc năm. Báo cáo này sẽ hiển thị tổng quan về thu nhập, chi tiêu, tiết kiệm, và đầu tư

Chương 2: Khảo sát và xác định yêu cầu

của người dùng, từ đó giúp người dùng có cái nhìn rõ ràng hơn về tình hình tài chính của họ.

- Tính năng phân tích chi tiết theo danh mục chi tiêu giúp người dùng biết được khoản mục nào đang chiếm nhiều chi phí nhất, từ đó điều chỉnh thói quen chi tiêu nếu cần.

- **Thông báo và nhắc nhở:**

- Tích hợp tính năng thông báo nhắc nhở người dùng về các khoản chi tiêu.
- Cung cấp tính năng nhắc nhở về mục tiêu tiết kiệm.

- **Tích hợp bảo mật thông tin:**

- Ứng dụng phải tích hợp các biện pháp bảo mật thông tin như mã hóa dữ liệu, xác thực hai yếu tố (two-factor authentication) để đảm bảo an toàn cho tài khoản và dữ liệu tài chính của người dùng.

Đối với quản trị viên của hệ thống

2.2.2. Yêu cầu phi chức năng

- **Tính bảo mật (Security):**

- Dữ liệu tài chính cá nhân của người dùng phải được mã hóa và bảo vệ tuyệt đối. Firebase cung cấp các phương thức mã hóa dữ liệu trong quá trình truyền tải và lưu trữ, đảm bảo rằng không có thông tin tài chính nhạy cảm nào bị lộ ra ngoài.
- Sử dụng xác thực hai yếu tố (2FA) để tăng cường bảo mật cho tài khoản người dùng, đảm bảo rằng chỉ những người được ủy quyền mới có thể truy cập vào tài khoản và thông tin tài chính của họ.

Chương 2: Khảo sát và xác định yêu cầu

- Các quy tắc bảo mật của Firebase (Firebase Security Rules) cần được cấu hình chặt chẽ để giới hạn quyền truy cập vào dữ liệu dựa trên vai trò người dùng.

- **Hiệu suất (Performance):**

- Ứng dụng cần phản hồi nhanh chóng trong vòng dưới 2 giây đối với hầu hết các thao tác thông thường của người dùng như ghi nhận giao dịch, xem báo cáo tài chính, hay cập nhật thông tin cá nhân.
- Ứng dụng phải duy trì hiệu suất tốt ngay cả khi có số lượng lớn người dùng truy cập cùng lúc (thang đo lên đến hàng triệu người dùng). Firebase Firestore và Realtime Database có khả năng mở rộng tự động, giúp ứng dụng duy trì tốc độ truy cập và xử lý dữ liệu mượt mà.
- Giảm thiểu thời gian tải dữ liệu bằng cách sử dụng bộ nhớ đệm và tải dữ liệu một cách không đồng bộ (asynchronous).

- **Khả năng mở rộng (Scalability):**

- Ứng dụng phải có khả năng mở rộng linh hoạt để phục vụ số lượng người dùng lớn mà không ảnh hưởng đến hiệu suất. Firebase Cloud Functions và Firestore hỗ trợ khả năng mở rộng tự động, giúp hệ thống dễ dàng thích ứng khi có thêm nhiều người dùng hoặc nhu cầu sử dụng tăng lên.
- Hệ thống cần được thiết kế sao cho việc thêm tính năng mới hoặc tích hợp với các dịch vụ khác (ví dụ: ngân hàng, ví điện tử) có thể thực hiện một cách dễ dàng mà không làm gián đoạn hoạt động hiện tại.

- **Tính khả dụng (Availability):**

- Ứng dụng phải đảm bảo tính sẵn sàng (uptime) tối thiểu 99.9%, đảm bảo rằng người dùng có thể truy cập và sử dụng dịch vụ liên tục mà không bị gián đoạn. Firebase cung cấp cơ sở hạ tầng có độ tin cậy cao, đảm bảo rằng

Chương 2: Khảo sát và xác định yêu cầu

hệ thống sẽ luôn hoạt động ổn định ngay cả trong các tình huống tải cao hoặc có lỗi phần cứng.

- Ứng dụng cần tích hợp tính năng sao lưu và khôi phục dữ liệu (data backup and recovery) để đảm bảo không mất mát dữ liệu trong trường hợp xảy ra sự cố.
- **Khả năng tương thích đa nền tảng (Cross-Platform Compatibility):**
- Ứng dụng cần hỗ trợ trên nhiều nền tảng khác nhau, bao gồm iOS, Android và phiên bản web. Firebase hỗ trợ dễ dàng việc phát triển đa nền tảng, giúp hệ thống hoạt động mượt mà trên tất cả các thiết bị mà người dùng có thể sử dụng.
- Giao diện người dùng phải tương thích và tối ưu hóa cho cả điện thoại di động và máy tính bảng, đảm bảo trải nghiệm nhất quán trên các thiết bị có kích thước màn hình khác nhau.
- **Khả năng sử dụng (Usability):**
- Giao diện người dùng cần thân thiện và trực quan, giúp người dùng dễ dàng thao tác mà không cần phải có kỹ năng công nghệ cao. Tính năng quản lý tài chính phải được trình bày rõ ràng, dễ hiểu với các hướng dẫn ngắn gọn và gợi ý thao tác.
- Ứng dụng cần cung cấp hướng dẫn sử dụng và tài liệu hỗ trợ ngay trong ứng dụng (in-app tutorials) để giúp người dùng mới làm quen với các tính năng một cách nhanh chóng.
- **Khả năng bảo trì (Maintainability):**
- Hệ thống phải dễ dàng bảo trì và nâng cấp mà không làm gián đoạn quá trình sử dụng của người dùng. Firebase Cloud Functions cho phép quản lý các chức năng backend một cách linh hoạt và dễ bảo trì.

Chương 2: Khảo sát và xác định yêu cầu

- Mọi bản cập nhật hệ thống phải được triển khai mà không ảnh hưởng đến các tính năng hiện tại của người dùng.
- **Tính phản hồi và cập nhật thời gian thực (Real-time Updates):**
- Ứng dụng cần cung cấp tính năng cập nhật dữ liệu và thông tin tài chính thời gian thực. Các giao dịch tài chính của người dùng phải được cập nhật ngay lập tức, cho phép người dùng theo dõi tình hình tài chính một cách chính xác và liên tục.
- Firebase Realtime Database và Firestore cung cấp tính năng cập nhật thời gian thực, giúp đảm bảo người dùng luôn có thông tin mới nhất về tài chính cá nhân của họ.

2.2.3. Đặc điểm đặc biệt

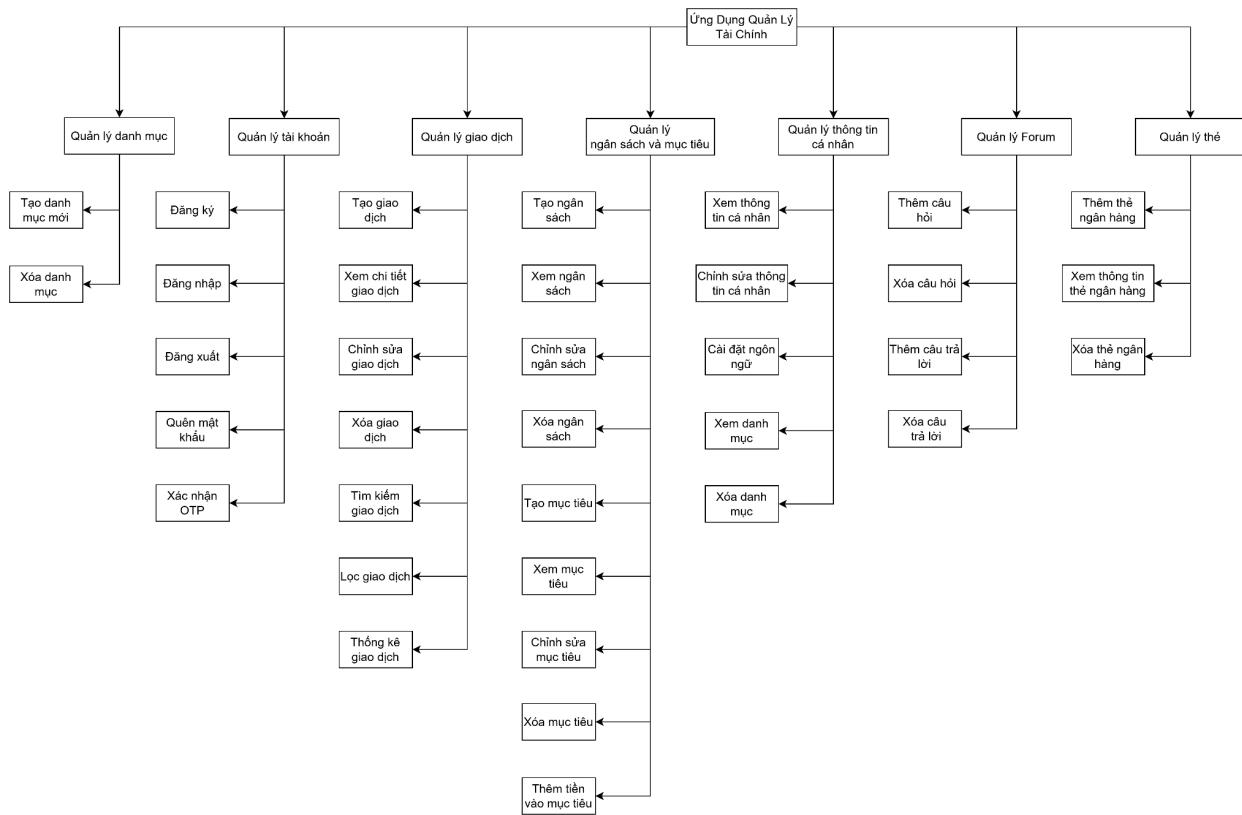
- **Hỗ trợ đa ngôn ngữ và địa phương hóa:**
- Ứng dụng cần tích hợp khả năng hỗ trợ nhiều ngôn ngữ và tiền tệ khác nhau, giúp người dùng từ nhiều quốc gia có thể sử dụng dễ dàng. Khả năng điều chỉnh theo ngôn ngữ và văn hóa địa phương sẽ giúp ứng dụng phù hợp với đối tượng người dùng toàn cầu.
- Việc hỗ trợ tiền tệ địa phương cũng cho phép người dùng dễ dàng theo dõi tài chính cá nhân mà không gặp rào cản về quy đổi tiền tệ.
- **Cảnh báo tài chính thông minh và nhắc nhở cá nhân hóa:**
- Ứng dụng sẽ gửi thông báo và nhắc nhở thông minh khi người dùng sắp đạt đến hạn mức ngân sách. Các thông báo này sẽ được cá nhân hóa dựa trên thói quen chi tiêu và tình hình tài chính của từng người dùng, giúp họ quản lý tài chính một cách chủ động và hiệu quả hơn.

CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Mô hình hóa yêu cầu

3.1.1. Sơ đồ phân rã chức năng

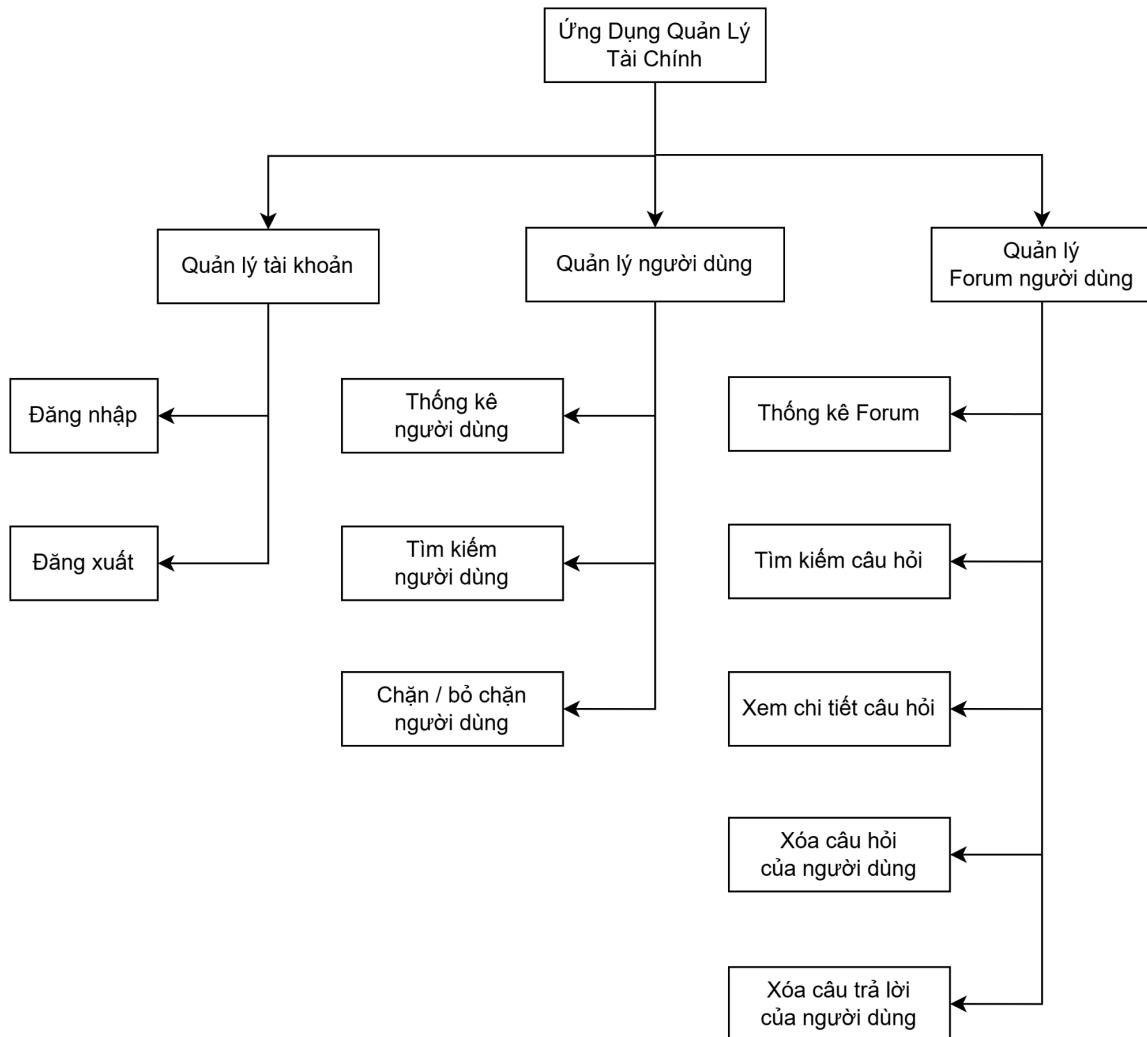
a. Đối với người dùng



Hình 1a. Sơ đồ phân rã chức năng đối với người dùng.

Chương 3: Phân tích thiết kế hệ thống

b. Đối với quản trị viên

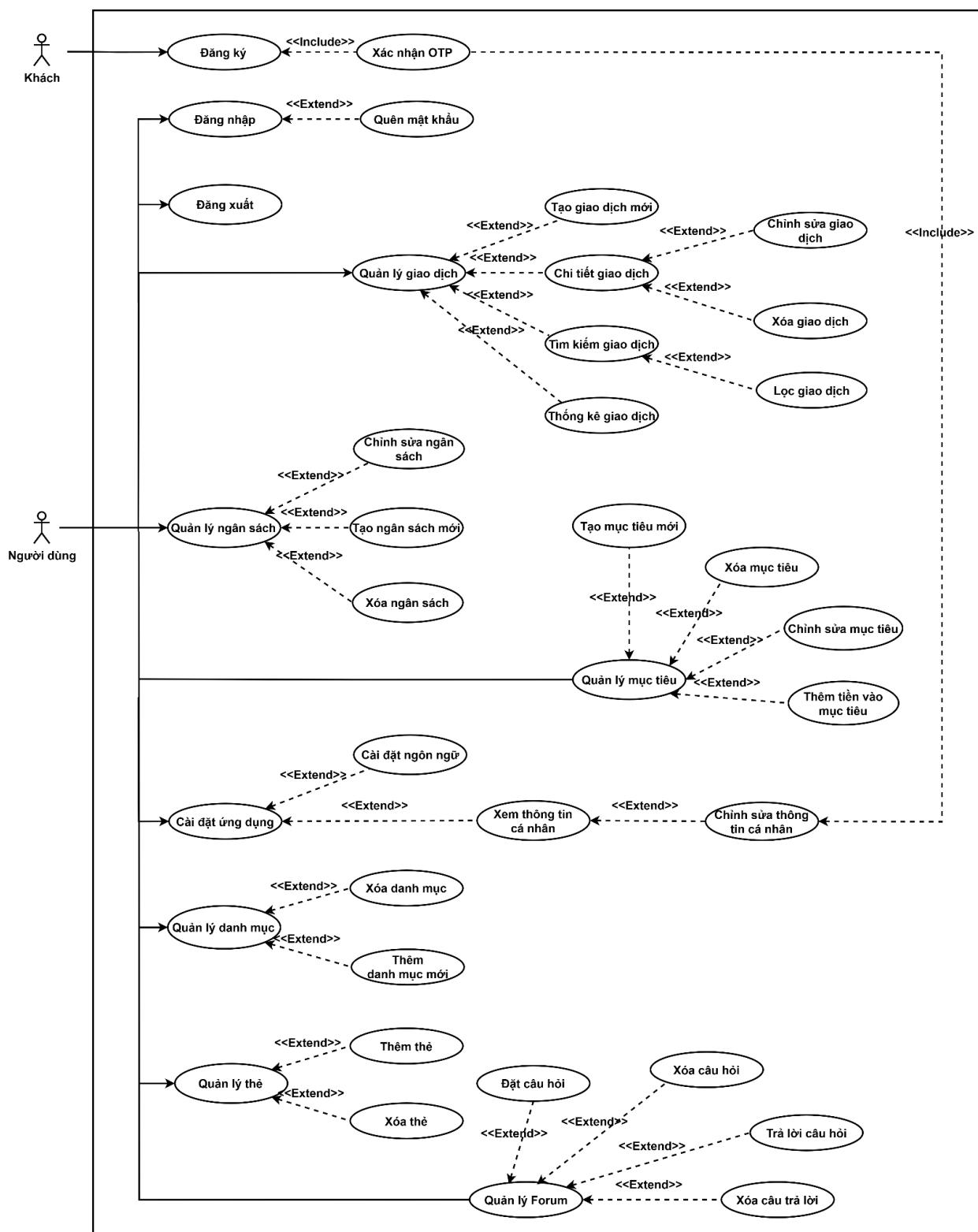


Hình 1b. Sơ đồ phân rã chức năng đối với quản trị viên.

3.1.2. Lược đồ Use case

a. Đối với người dùng

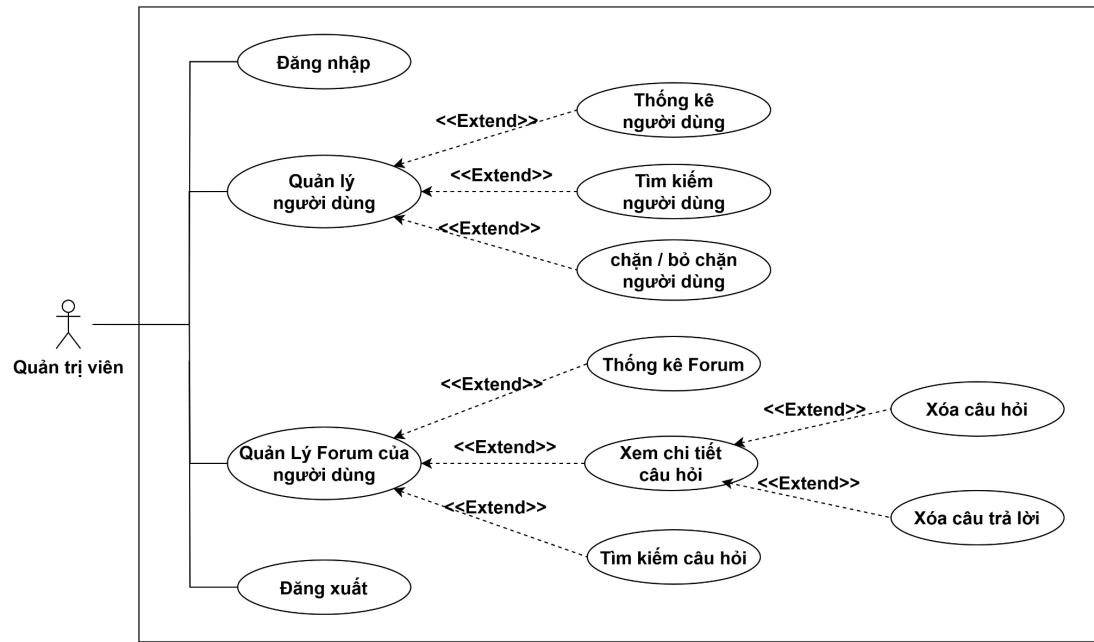
Chương 3: Phân tích thiết kế hệ thống



Hình 2a. Lược đồ Use case đối với người dùng.

Chương 3: Phân tích thiết kế hệ thống

b. Đối với quản trị viên

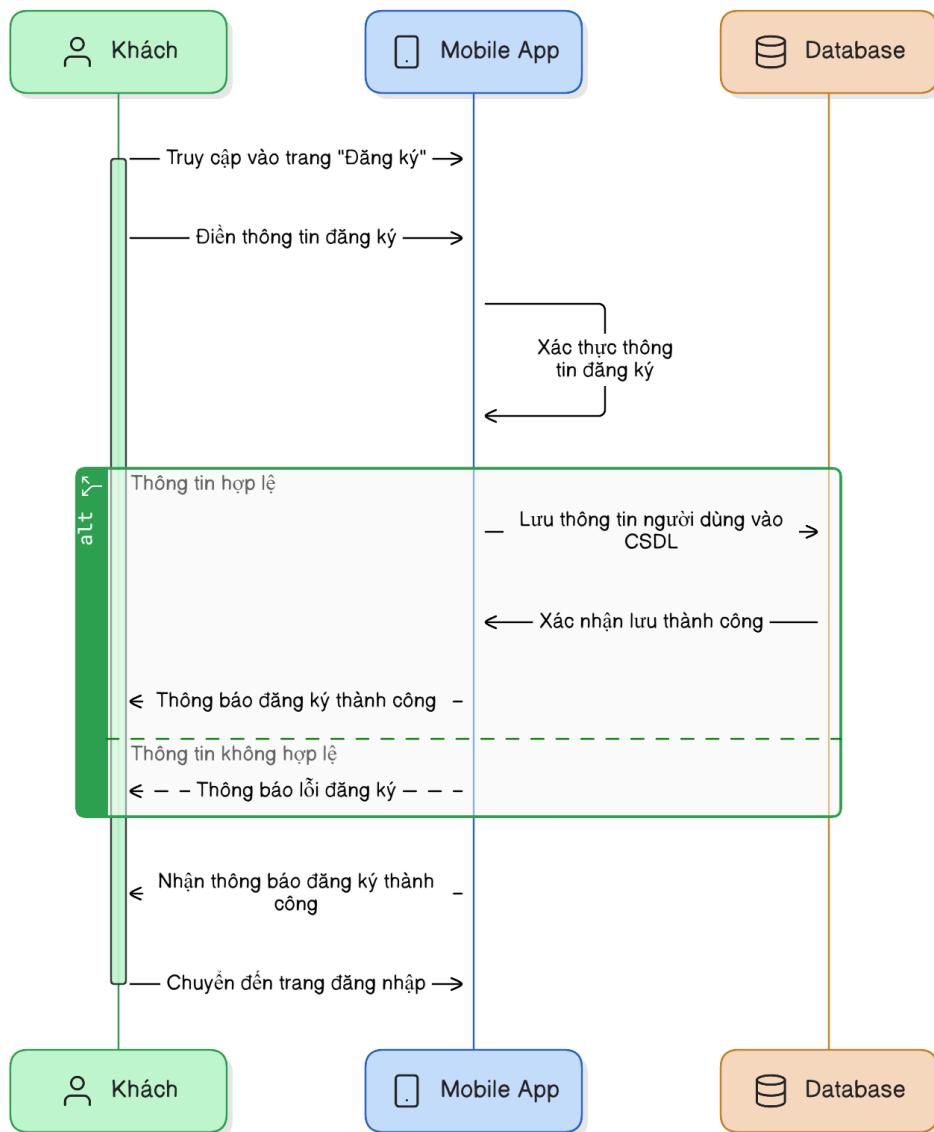


Hình 2b. Lược đồ Use case đối với quản trị viên.

Chương 3: Phân tích thiết kế hệ thống

3.1.3. Phân tích Use case và Sequence

3.1.3.1. Đăng ký tài khoản



Hình 3.1. Sequence - Đăng ký tài khoản

Name	Đăng ký tài khoản
Brief description	Khách (Guest) truy cập tạo mới tài khoản.

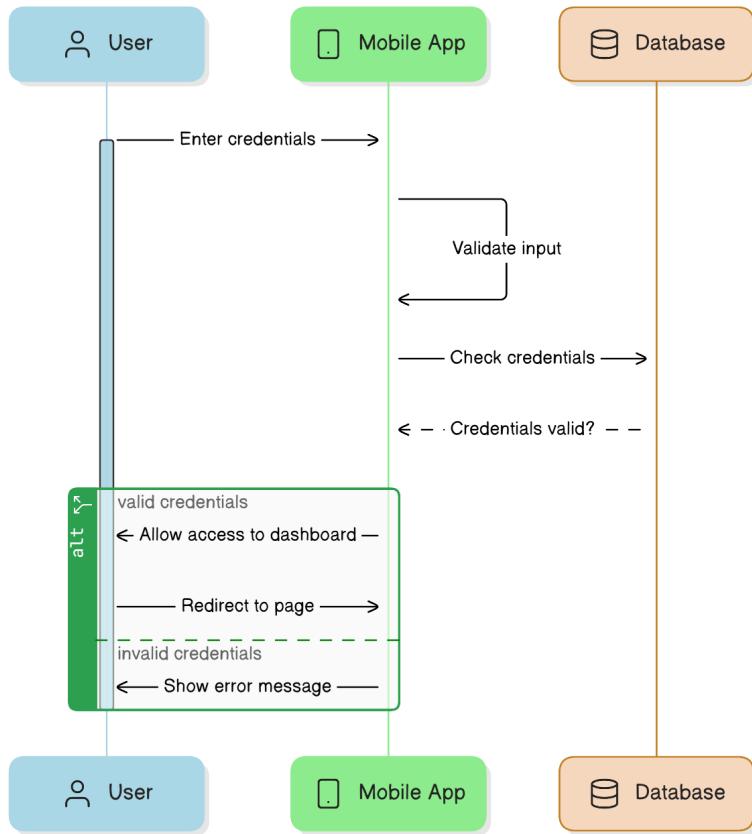
Chương 3: Phân tích thiết kế hệ thống

Actor(s)	Khách
Pre-conditions	
Post-conditions	<ul style="list-style-type: none"> Nếu đăng ký thành công: Người dùng được tạo mới tài khoản, thông tin cá nhân được lưu vào CSDL. Nếu đăng ký không thành công: Thông báo không tạo được tài khoản, buộc người dùng nhập lại thông tin cho chính xác.
Flow of events	
Basic flow	<p>Use case bắt đầu khi khách truy cập vào trang tạo tài khoản.</p> <ol style="list-style-type: none"> Khách điền vào các thông tin mà hệ thống yêu cầu và nhấn ĐĂNG KÝ. Hệ thống xác thực thông tin theo quy định. Hệ thống tạo mới tài khoản và lưu thông tin vào CSDL. Hệ thống thông báo tạo tài khoản thành công và chuyển đến trang đăng nhập.
Alternative flow	<p>Nếu người dùng nhập thiếu thông tin, trùng email hoặc mật khẩu không đúng quy định, khi đó hệ thống sẽ:</p> <ol style="list-style-type: none"> Hệ thống mô tả lý do không thể tạo mới tài khoản. Hệ thống hiển thị lại biểu mẫu cho người dùng chỉnh sửa thông tin đăng ký. Người dùng nhập lại thông tin được yêu cầu, Basic Flow khi đó sẽ tiếp tục tại bước 1.
Extension point	

Bảng 1.1. Đặc tả Use case - Đăng ký tài khoản

Chương 3: Phân tích thiết kế hệ thống

3.1.3.2. Đăng nhập



Hình 3.2. Sequence - Đăng nhập

Name	Đăng nhập
Brief description	Người dùng đăng nhập vào hệ thống.
Actor(s)	Người dùng
Pre-conditions	Không.
Post-conditions	Người dùng được xác thực và vào trang được chỉ định tùy theo từng chức vụ.
Flow of events	Thông báo lỗi đăng nhập và yêu cầu đăng nhập lại.

Chương 3: Phân tích thiết kế hệ thống

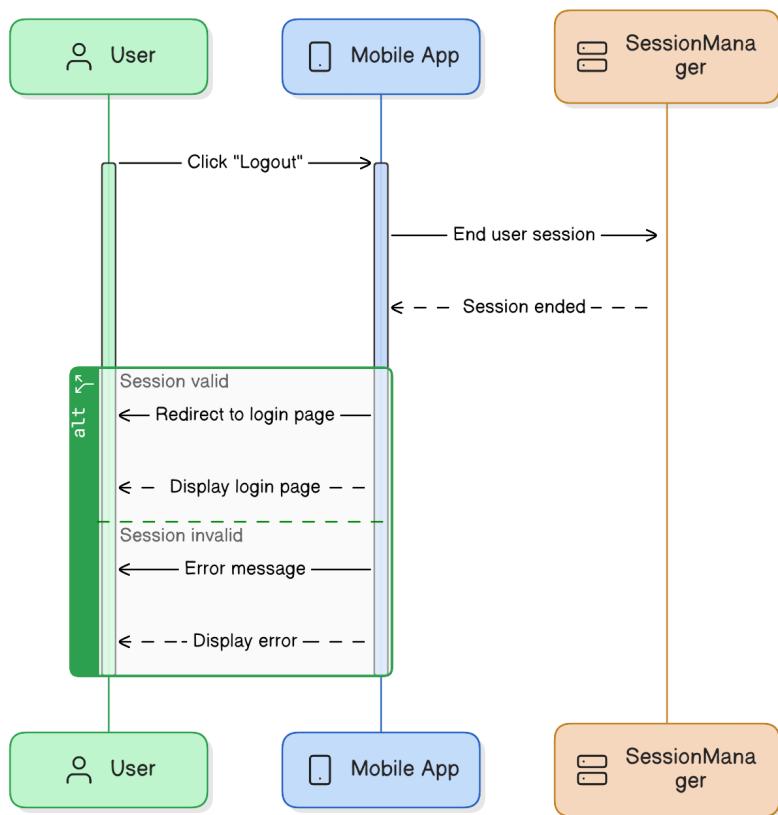
Basic flow	<p>Use case được kích hoạt khi người dùng cần đăng nhập vào hệ thống hoặc xác thực danh tính để sử dụng các chức năng của hệ thống:</p> <ol style="list-style-type: none">1. Người dùng nhập tài khoản và mật khẩu vào các ô input và bấm "Login".2. Hệ thống kiểm tra input và xác thực.3. Hệ thống thông báo xác thực thành công.
Alternative flow	Khi xác thực thất bại hoặc xảy ra lỗi: hệ thống thông báo lỗi sai.
Extension point	

Bảng 1.2. Đặc tả Use case - Đăng nhập

Chương 3: Phân tích thiết kế hệ thống

3.1.3.3. Đăng xuất

User Logout Process



Hình 3.3. Sequence - Đăng xuất

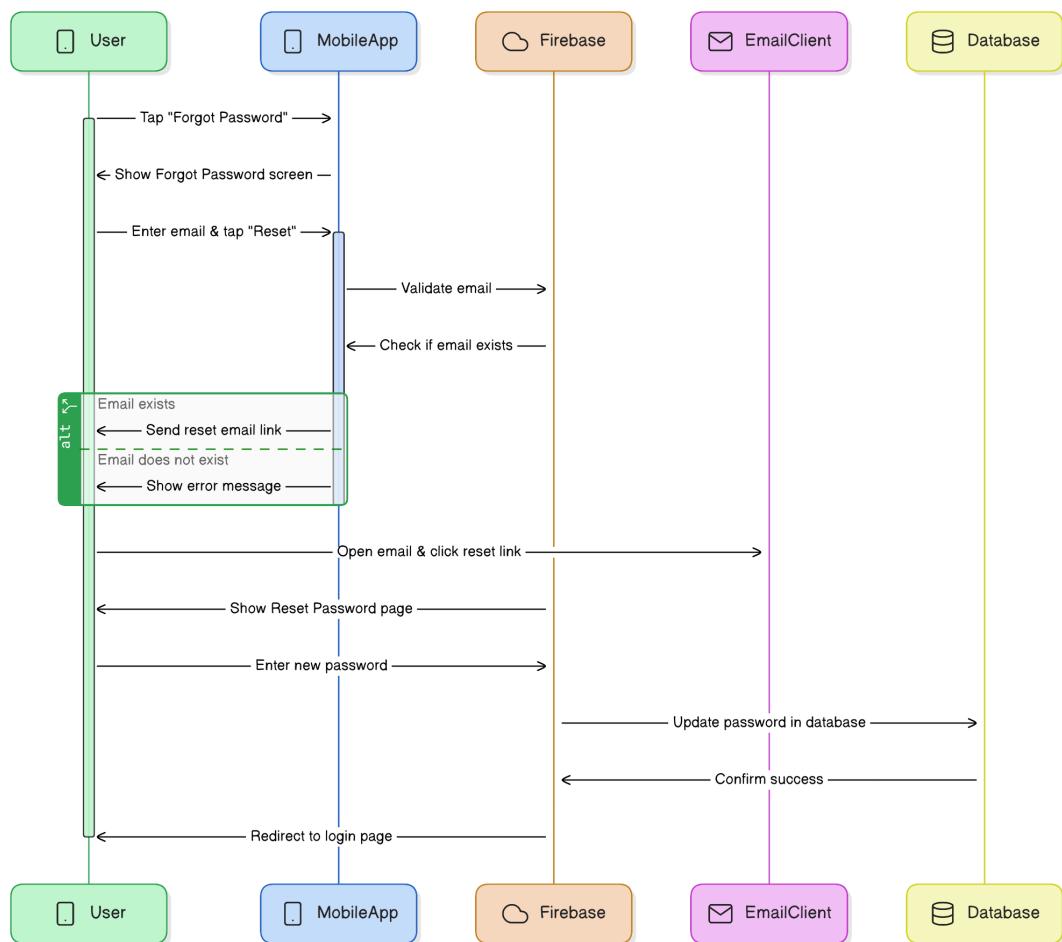
Name	Đăng xuất
Brief description	Đăng xuất tài khoản người dùng khỏi hệ thống.
Actor(s)	Người dùng
Pre-conditions	Actors đã đăng nhập thành công vào hệ thống.
Post-conditions	Tài khoản được đăng xuất thành công ra khỏi hệ thống.
Flow of events	

Chương 3: Phân tích thiết kế hệ thống

Basic flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào nút "Logout" ở trang Setting. 2. Hệ thống thực hiện đăng xuất cho người dùng. 3. Chuyển qua trang đăng nhập.
Alternative flow	
Extension point	

Bảng 1.3. Đặc tả Use case - Đăng xuất

3.1.3.4. Quên mật khẩu



Hình 3.4. Sequence - Quên mật khẩu

Chương 3: Phân tích thiết kế hệ thống

Name	Quên mật khẩu
Brief description	Giúp người dùng đặt lại mật khẩu qua email khi người dùng quên mật khẩu.
Actor(s)	Người dùng
Pre-conditions	Actors đã có tài khoản trong hệ thống.
Post-conditions	Tài khoản của người dùng được cập nhật mật khẩu mới.
Flow of events	
Basic flow	<p>1. Người dùng chọn "Forgot Password" ở trang Login.</p> <p>2. Hệ thống hiển thị trang ForgotPassword .</p> <p>3. Người dùng nhập email của mình và chọn nút “Reset Password”.</p> <p>4. Hệ thống kiểm tra email và gửi mail đặt lại mật khẩu đến email của người dùng.</p> <p>5. Người dùng mở email đặt lại mật khẩu và chọn nút đặt lại mật khẩu.</p> <p>6. Người dùng được chuyển đến trang đặt lại mật khẩu.</p> <p>7. Người dùng nhập mật khẩu mới và chọn “Reset”.</p> <p>8. Đặt lại mật khẩu thành công, người dùng được chuyển về trang Login.</p>
Alternative flow	<p>Đặt lại mật khẩu thất bại.</p> <p>4. Hệ thống kiểm tra email thất bại.</p> <p>5. Hệ thống hiện thông báo lỗi.</p> <p>6. Người dùng nhập lại email.</p> <p>Use Case quay lại bước 3 của Basic flow.</p>

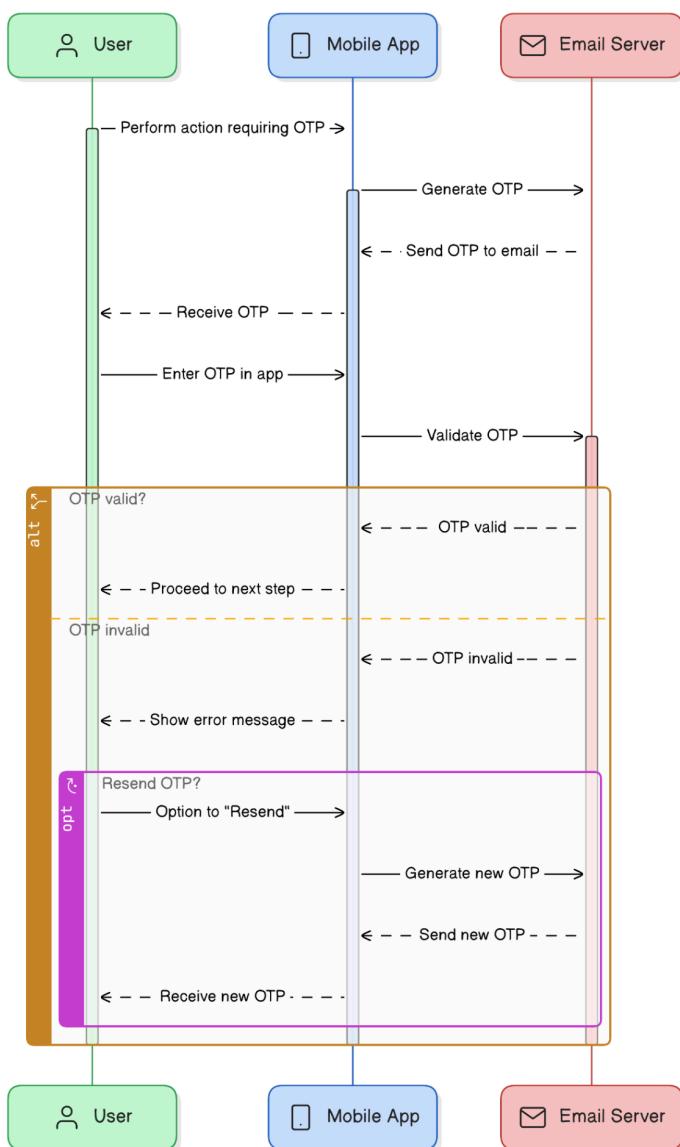
Chương 3: Phân tích thiết kế hệ thống

Extension point	

Bảng 1.4. Đặc tả Use case - Quên mật khẩu

3.1.3.5. Xác nhận OTP

OTP Verification Process



Chương 3: Phân tích thiết kế hệ thống

Hình 3.5. Sequence - Xác nhận OTP

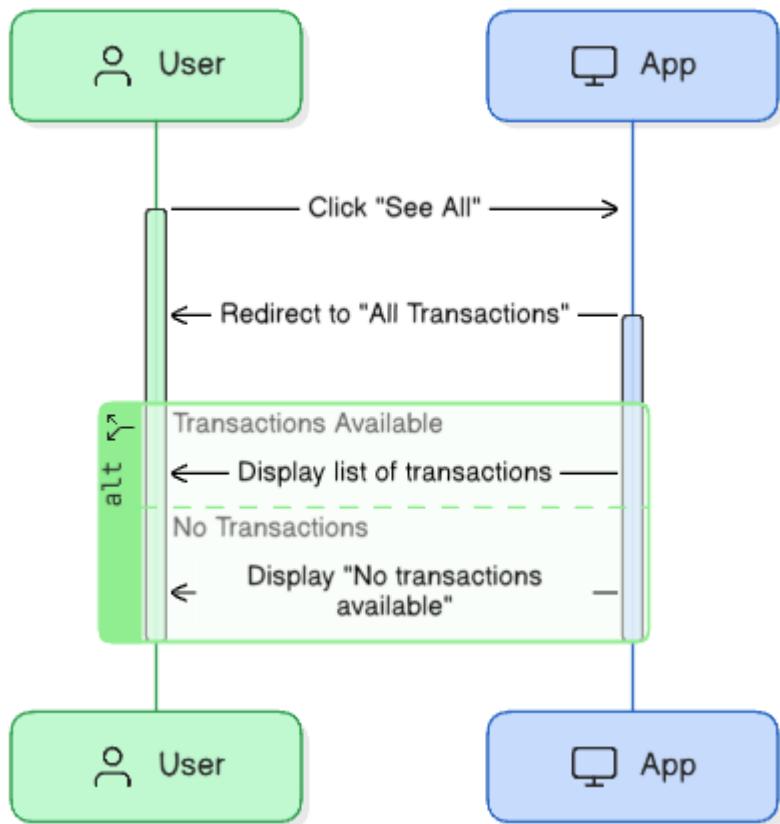
Name	Xác nhận OTP
Brief description	Người dùng xác thực thông tin đăng ký, thay đổi mật khẩu bằng cách nhập mã OTP được gửi tới (qua email).
Actor(s)	Người dùng hoặc Khách
Pre-conditions	
Post-conditions	Người dùng hoàn tất quá trình xác thực và được truy cập hệ thống hoặc được thay đổi mật khẩu.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng thực hiện một hành động yêu cầu xác thực (đăng ký hoặc thay đổi mật khẩu). 2. Hệ thống gửi mã OTP đến người dùng qua phương thức xác thực (email). 3. Người dùng nhập mã OTP nhận được vào ứng dụng/hệ thống. 4. Hệ thống xác nhận mã OTP. 5. Nếu mã OTP đúng, hệ thống cho phép người dùng hoàn tất quy trình xác thực hoặc giao dịch.
Alternative flow	<p>Nếu mã OTP không hợp lệ hoặc hết hạn:</p> <ol style="list-style-type: none"> 1. Hệ thống thông báo mã OTP không hợp lệ hoặc đã hết hạn. 2. Hệ thống cho phép người dùng yêu cầu mã OTP mới hoặc nhập lại mã. <p>Nếu người dùng không nhận được mã OTP:</p> <ol style="list-style-type: none"> 1. Người dùng chọn tùy chọn "Resend". 2. Hệ thống gửi mã OTP mới.

Chương 3: Phân tích thiết kế hệ thống

Extension point	
-----------------	--

Bảng 1.5. Đặc tả Use case - Xác nhận OTP

3.1.3.6. Quản lý giao dịch



Hình 3.6. Sequence - Quản lý giao dịch

Name	Chi tiết giao dịch.
Brief description	Giúp người dùng quản lý các giao dịch thu chi.
Actor(s)	Người dùng

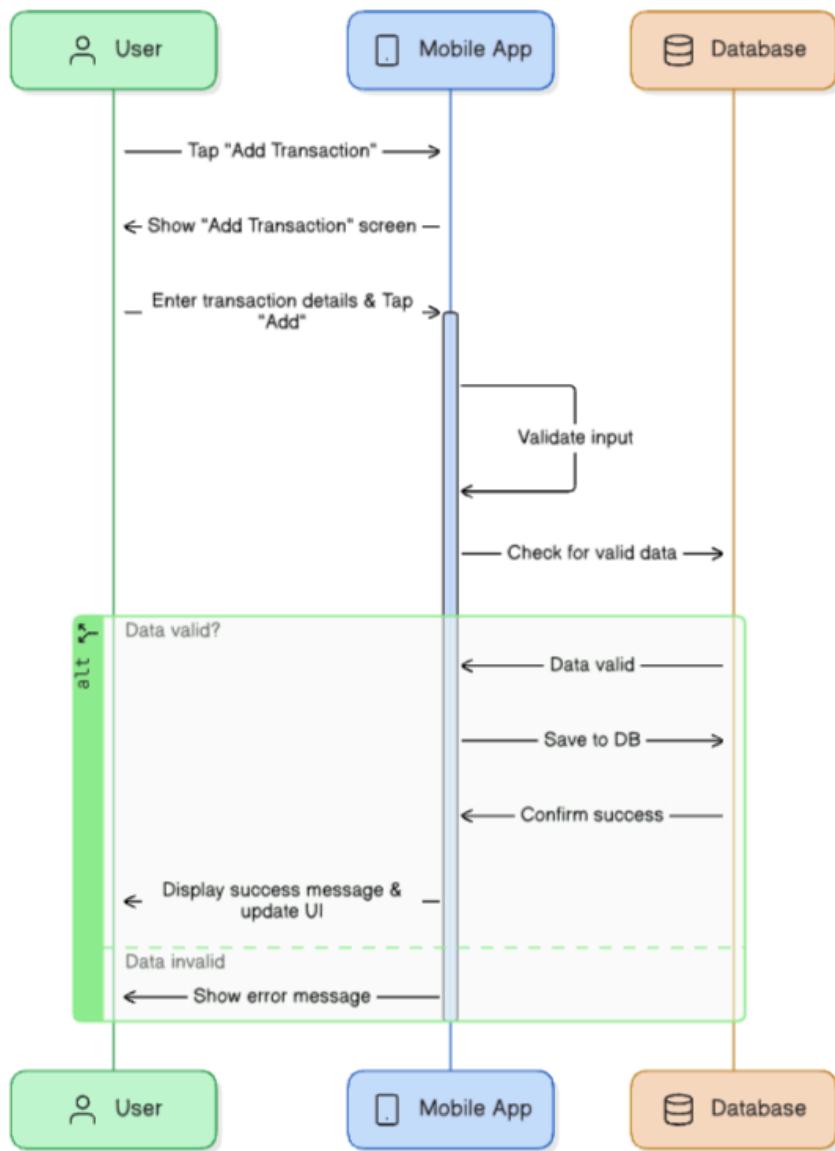
Chương 3: Phân tích thiết kế hệ thống

Pre-conditions	Actors đã có tài khoản trong hệ thống.
Post-conditions	Người dùng mở được trang “All Transactions”
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng nhấn vào “See All” trong trang Home2. Hệ thống chuyển đến trang “All Transactions”
Alternative flow	
Extension point	

Bảng 1.6. Đặc tả Use case - Quản lý giao dịch

Chương 3: Phân tích thiết kế hệ thống

3.1.3.7. Tạo giao dịch mới



Hình 3.7. Sequence - Tạo giao dịch mới

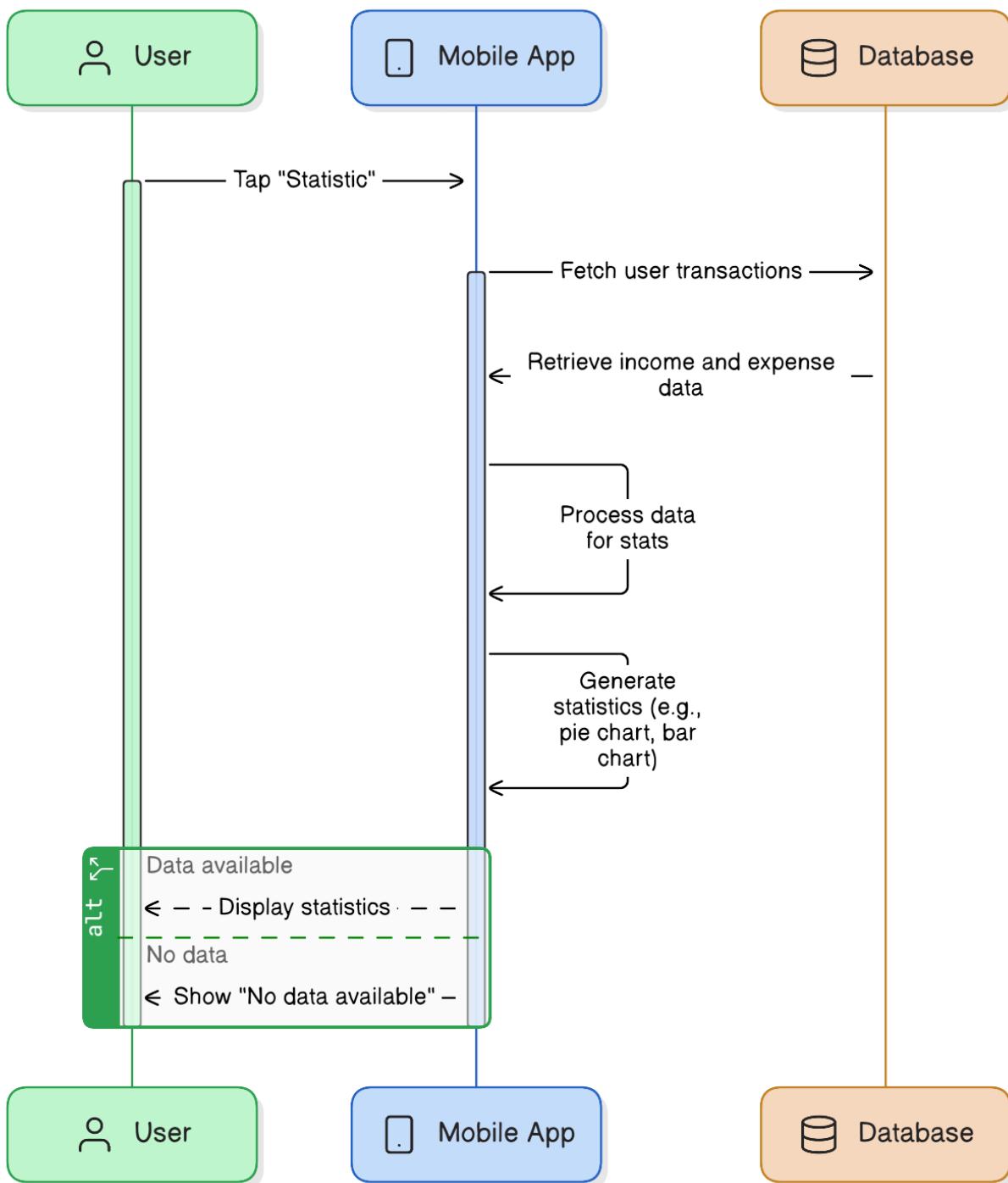
Name	Tạo giao dịch mới.
Brief description	Giúp người dùng thêm thông tin giao dịch mới.
Actor(s)	Người dùng

Chương 3: Phân tích thiết kế hệ thống

Pre-conditions	Actors đã có tài khoản trong hệ thống và đã đăng nhập thành công vào hệ thống.
Post-conditions	Hệ thống hiển thị giao dịch vừa thêm của người dùng.
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng chọn "Add Transaction".2. Người dùng điền và chọn các thông tin giao dịch, sau đó bấm "Add".3. Hệ thống xác nhận và cập nhật vào dữ liệu của người dùng.4. Hệ thống thông báo thêm giao dịch thành công.
Alternative flow	<p>Thêm giao dịch thất bại khi người dùng điền thông tin giao dịch sai định dạng.</p> <ol style="list-style-type: none">1. Hệ thống kiểm tra thông tin giao dịch mới.2. Hệ thống thông báo thông tin giao dịch không hợp lệ và yêu cầu người dùng điền lại. <p>Use case quay lại bước 2 Basic flow.</p>
Extension point	

Bảng 1.7. Đặc tả Use case - Tạo giao dịch mới

3.1.3.8. Thống kê giao dịch



Hình 3.8. Sequence - Thống kê giao dịch

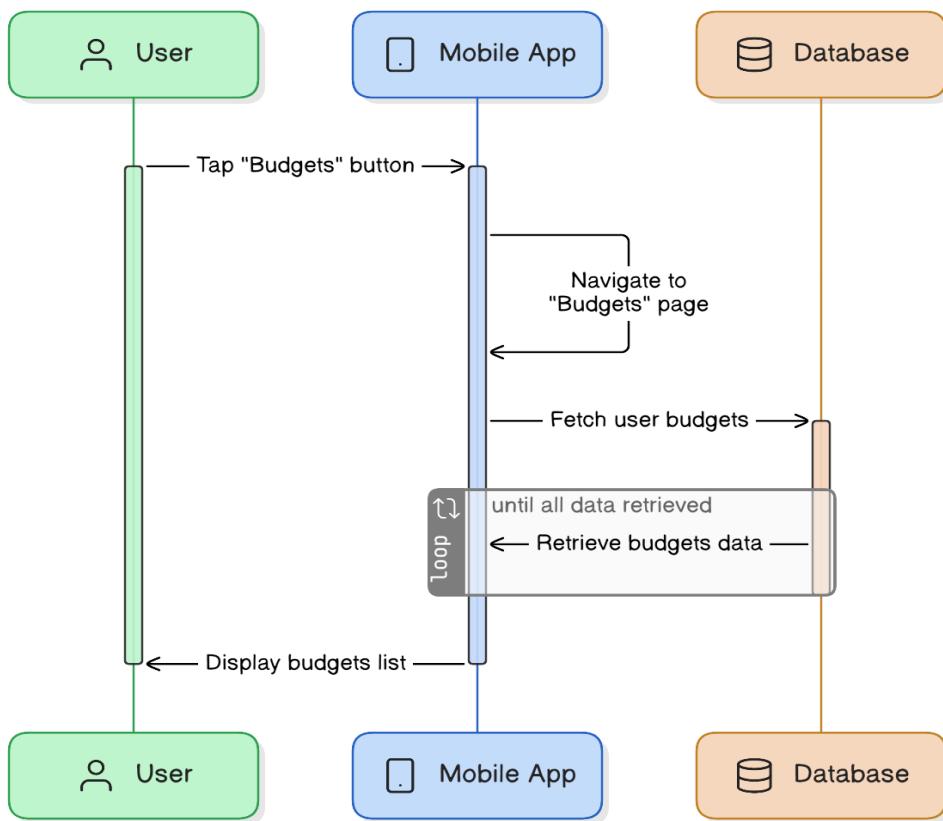
Chương 3: Phân tích thiết kế hệ thống

Name	Thống kê giao dịch.
Brief description	Giúp người dùng thống kê thu nhập và chi tiêu theo từng thời điểm.
Actor(s)	Người dùng
Pre-conditions	Actors đã có tài khoản trong hệ thống và đã đăng nhập thành công vào hệ thống.
Post-conditions	Hệ thống hiển thị thống kê của người dùng.
Flow of events	
Basic flow	<ol style="list-style-type: none">Người dùng chọn "Statistic".Hệ thống truy cập dữ liệu người dùng, thống kê thu nhập và chi tiêu của người dùng và hiển thị ra trang thống kê.
Alternative flow	<ol style="list-style-type: none">Hệ thống kiểm tra thấy thông tin thu nhập và chi tiêu của người dùng không tồn tại.Hệ thống thông báo người dùng không có thông tin thu nhập và chi tiêu.
Extension point	

Bảng 1.8. ĐẶC TẢ USE CASE - THỐNG KÊ GIAO DỊCH

Chương 3: Phân tích thiết kế hệ thống

3.1.3.9. Quản lý ngân sách



Hình 3.9. Sequence - Chi tiết ngân sách

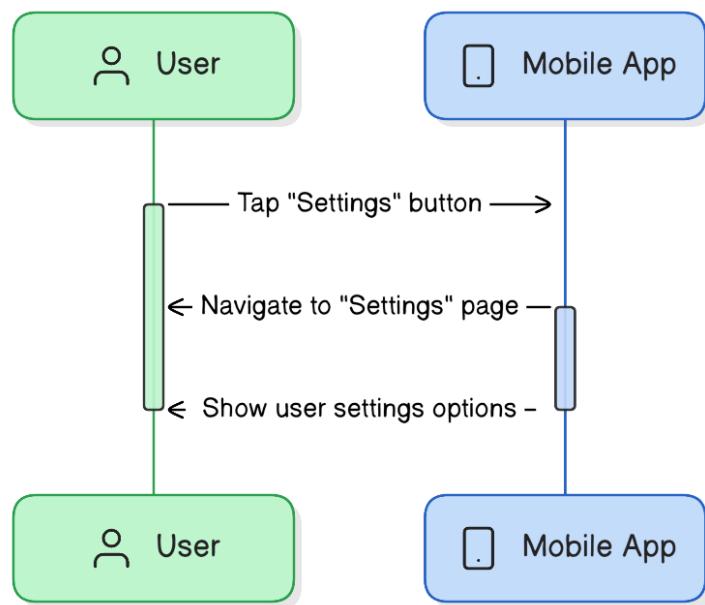
Name	Quản lý ngân sách
Brief description	Người dùng quản lý các ngân sách của mình
Actor(s)	Người dùng
Pre-conditions	Người dùng đã có tài khoản trong hệ thống
Post-conditions	Người dùng mở được trang “Budgets”
Flow of events	

Chương 3: Phân tích thiết kế hệ thống

Basic flow	<ul style="list-style-type: none">- Người dùng nhấn vào “Budgets” trên thanh điều hướng- Hệ thống điều hướng đến trang “Budgets”
Alternative flow	
Extension point	

Bảng 1.9. Đặc tả Use case - Quản lý ngân sách

3.1.3.10. Cài đặt ứng dụng



Hình 3.10. Sequence - Cài đặt ứng dụng

Name	Cài đặt ứng dụng
------	------------------

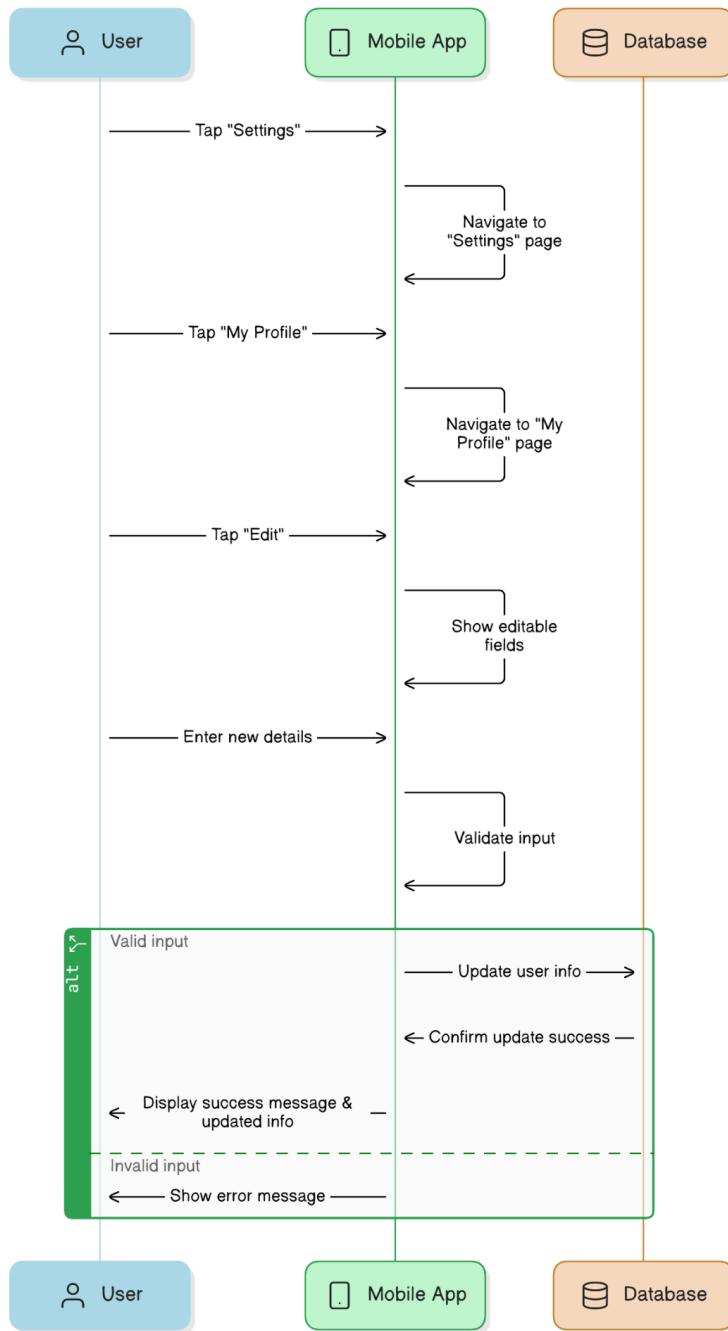
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng vào trang cài đặt của ứng dụng
Actor(s)	Người dùng
Pre-conditions	Người dùng đã có tài khoản trong hệ thống
Post-conditions	Người dùng chuyển hướng đến trang “Settings”
Flow of events	
Basic flow	<ul style="list-style-type: none">• Người dùng nhấn vào “Settings” trên thanh điều hướng• Hệ thống chuyển hướng đến trang ‘Setting’
Alternative flow	
Extension point	

Bảng 1.10. Đặc tả Use case - Cài đặt ứng dụng

Chương 3: Phân tích thiết kế hệ thống

3.1.3.11. Chính sửa thông tin cá nhân



Hình 3.11. Sequence - Chính sửa thông tin cá nhân

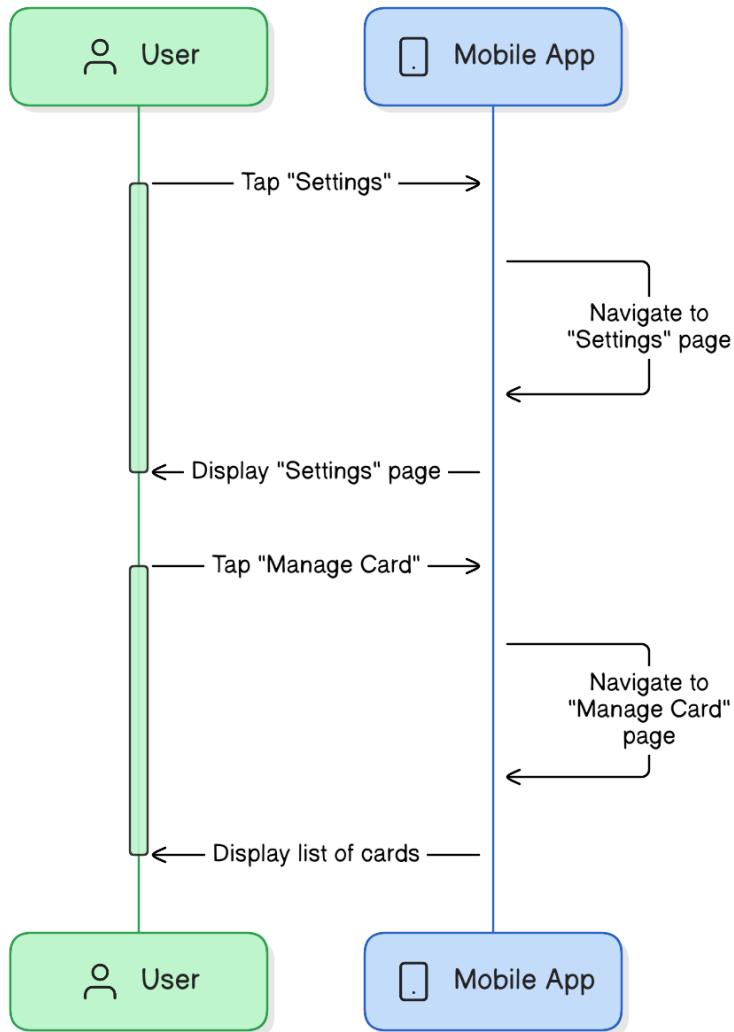
Chương 3: Phân tích thiết kế hệ thống

Name	Chỉnh sửa thông tin cá nhân.
Brief description	Người dùng có chỉnh sửa thông tin cá nhân của mình.
Actor(s)	Người dùng
Pre-conditions	Actors đã đăng nhập thành công.
Post-conditions	Thông tin người dùng đã được chỉnh sửa và cập nhật.
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng bấm vào Setting, chọn Profile và nhấn vào "Edit".2. Người dùng nhập thông tin mới và bấm "Confirm".3. Hệ thống cập nhật và hiển thị thông tin mới.
Alternative flow	Hệ thống không thể lưu thông tin và báo lỗi khi người dùng nhập thông tin sai định dạng.
Extension point	

Bảng 1.11. Đặc tả Use case - Chính sửa thông tin cá nhân

Chương 3: Phân tích thiết kế hệ thống

3.1.3.12. Quản lý thẻ



Hình 3.12. Sequence - Quản lý thẻ

Name	Quản lý thẻ
Brief description	Người dùng quản lý thẻ ngân hàng
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống

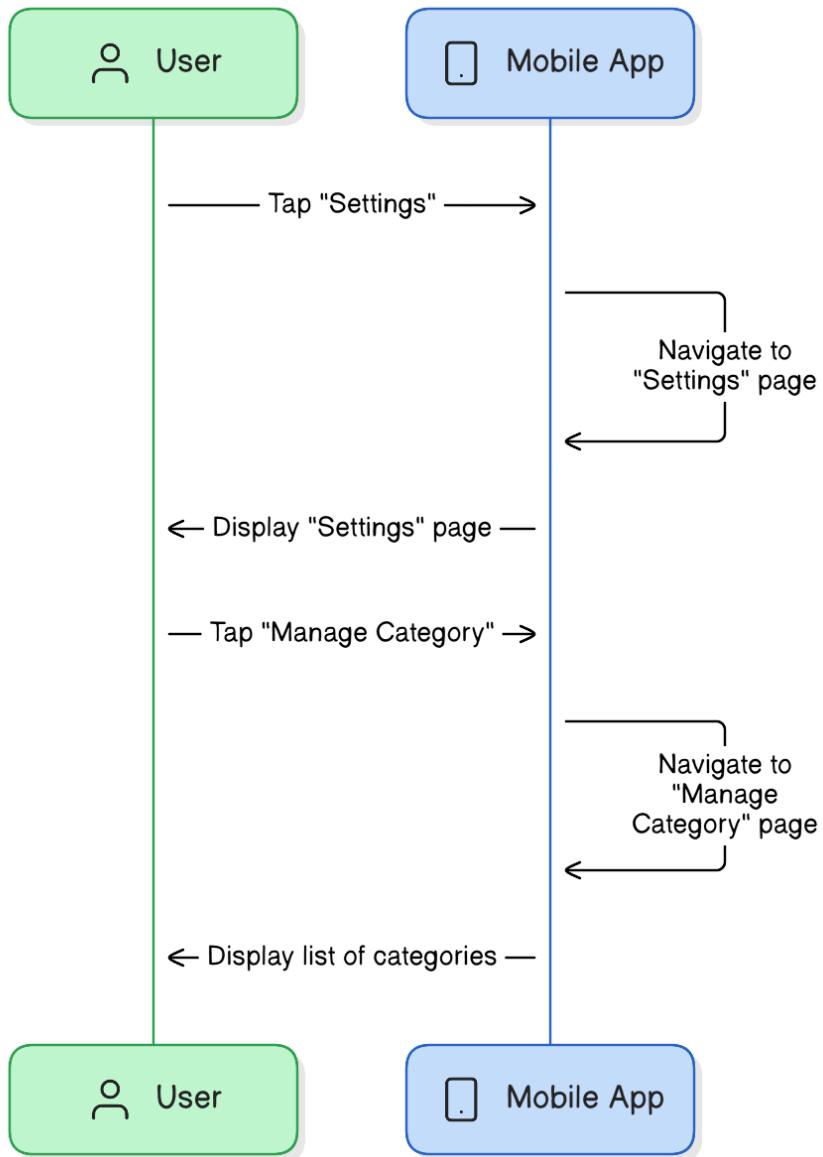
Chương 3: Phân tích thiết kế hệ thống

Post-conditions	Người dùng chuyển hướng đến “Manage Card”
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng đang ở trang “Settings”2. Người dùng nhấn vào “Manage Card”3. Hệ thống chuyển hướng đến “Manage Card”
Alternative flow	
Extension point	

Bảng 1.12. Đặc tả Use case - Quản lý thẻ

Chương 3: Phân tích thiết kế hệ thống

3.1.3.13. Quản danh mục



Hình 3.13. Sequence - Quản lý danh mục

Name	Quản lý danh mục
Brief description	Người dùng quản lý các danh mục

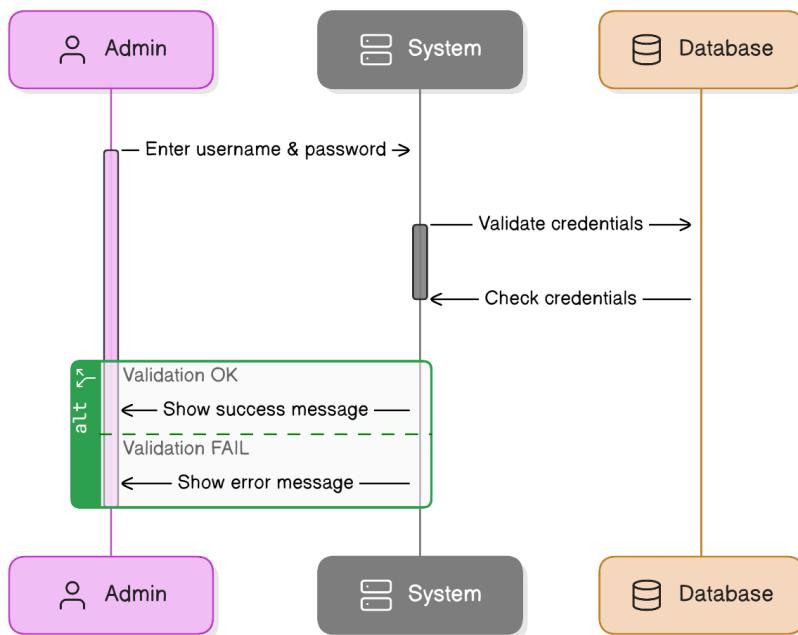
Chương 3: Phân tích thiết kế hệ thống

Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng mở trang “Manage Category”
Flow of events	
Basic flow	<ul style="list-style-type: none">● Người dùng đang ở trang “Settings”● Người dùng nhấn vào “Manage Category”● Hệ thống chuyển hướng đến “Manage Category”
Alternative flow	
Extension point	

Bảng 1.13. ĐẶC TẢ USE CASE - QUẢN LÝ DANH MỤC

Chương 3: Phân tích thiết kế hệ thống

3.1.3.14. Đăng nhập Admin



Hình 3.14. Sequence - Đăng nhập Admin

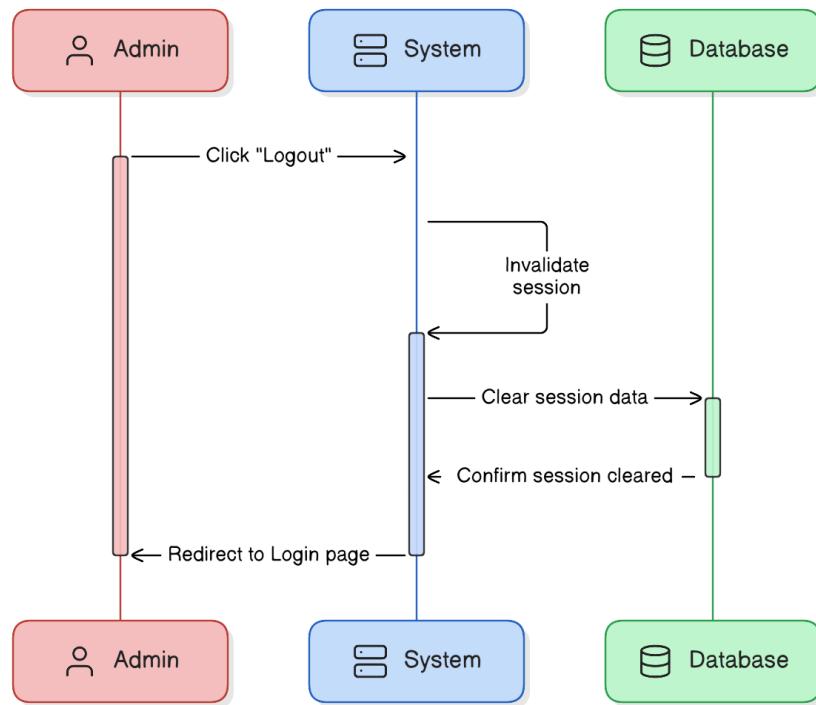
Name	Đăng nhập Admin
Brief description	Quản lý đăng nhập vào hệ thống
Actor(s)	Quản lý
Pre-conditions	Sử dụng tài khoản quản lý
Post-conditions	Quản lý vào trang quản lý
Flow of events	Thông báo lỗi đăng nhập và yêu cầu đăng nhập lại.
Basic flow	Use case được kích hoạt khi người dùng cần đăng nhập vào hệ thống hoặc xác thực danh tính để sử dụng các chức năng của hệ thống:

Chương 3: Phân tích thiết kế hệ thống

	4. Người dùng nhập tài khoản và mật khẩu vào các ô input và bấm "Login". 5. Hệ thống kiểm tra input và xác thực. 6. Hệ thống thông báo xác thực thành công.
Alternative flow	Khi xác thực thất bại hoặc xảy ra lỗi: hệ thống thông báo lỗi sai.
Extension point	

Bảng 1.14. Đặc tả Use case - Đăng nhập Admin

3.1.3.15. Đăng xuất Admin



Hình 3.15. Sequence - Đăng xuất Admin

Name	Đăng xuất Admin
------	------------------------

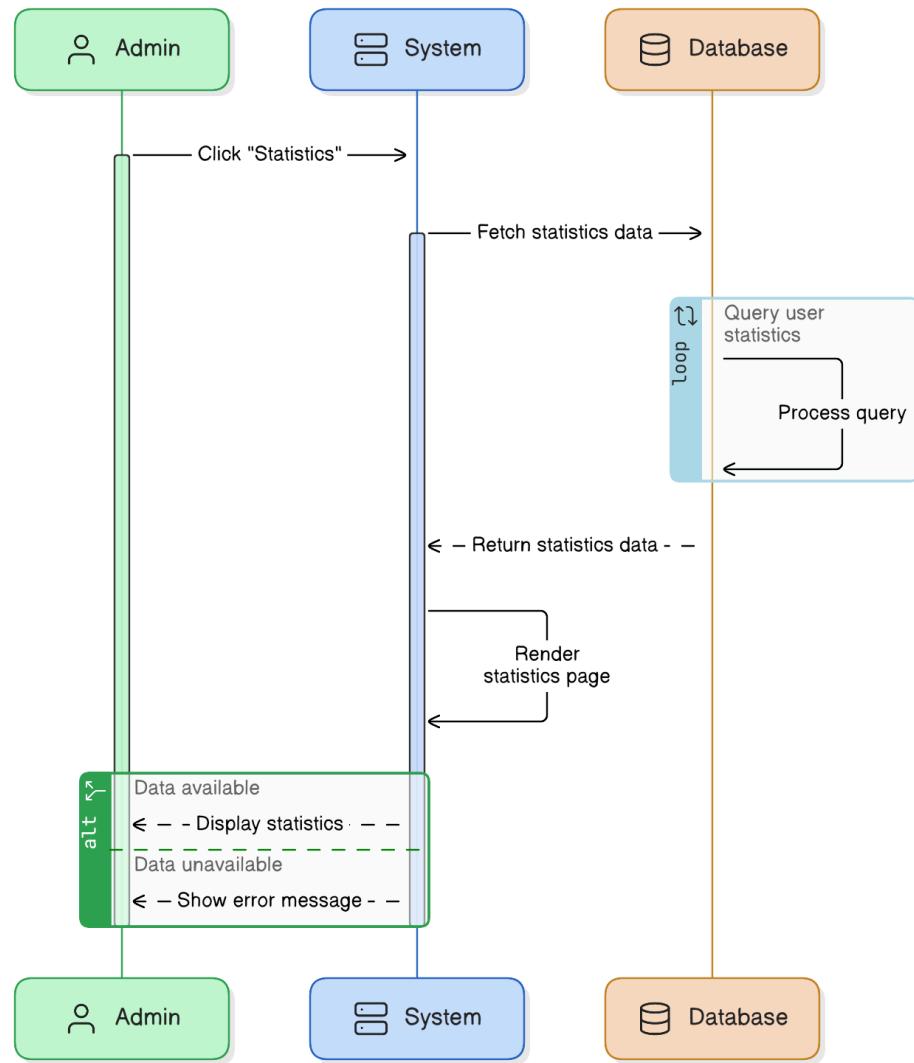
Chương 3: Phân tích thiết kế hệ thống

Brief description	Đăng xuất tài khoản quản lý khỏi hệ thống
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.
Post-conditions	Tài khoản được đăng xuất thành công ra khỏi hệ thống.
Flow of events	
Basic flow	<ol style="list-style-type: none">Người dùng bấm vào nút "Logout" ở Sidebar.Hệ thống thực hiện đăng xuất cho người dùng.Chuyển qua trang đăng nhập.
Alternative flow	
Extension point	

Bảng 1.15. Đặc tả Use case - Đăng xuất Admin

Chương 3: Phân tích thiết kế hệ thống

3.1.3.16. Thống kê người dùng



Hình 3.16. Sequence - Thống kê người dùng

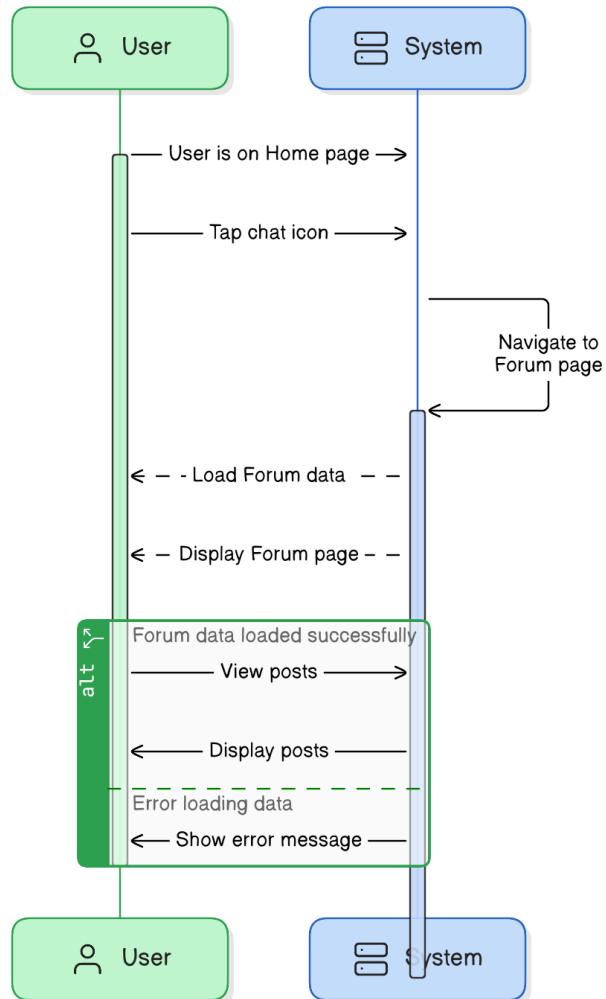
Name	Thống kê người dùng
Brief description	Quản lý xem được thống kê về người dùng hệ thống
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.

Chương 3: Phân tích thiết kế hệ thống

Post-conditions	Hiển thị thống kê về người dùng hệ thống
Flow of events	
Basic flow	<ol style="list-style-type: none">Quản lý bấm vào nút "Statistics" ở Sidebar.Hệ thống chuyển đến hiển thị thống kê.Hiển thị thông tin thống kê người dùng.
Alternative flow	
Extension point	

Bảng 1.16. Đặc tả Use case - Thống kê người dùng

3.1.3.17. Quản lý Forum của người dùng



Hình 3.17. Sequence - Thống kê Forum

Name	Quản lý Forum của người dùng
Brief description	Hệ thống chuyển hướng người dùng đến trang Forum
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống

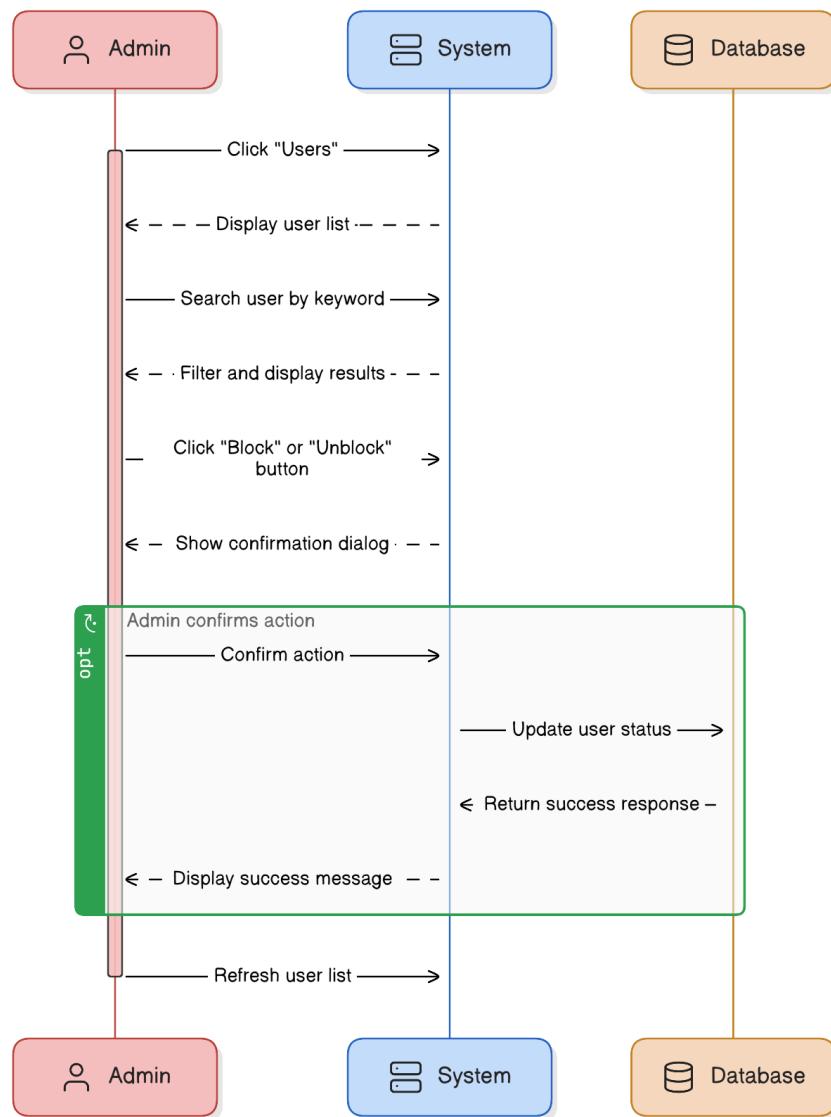
Chương 3: Phân tích thiết kế hệ thống

Post-conditions	Người dùng chuyển hướng đến trang Forum
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng đang ở trang Home của ứng dụng.2. Người dùng nhấp vào biểu tượng bong bóng chat trên header.3. Hệ thống điều hướng người dùng đến trang Forum.4. Hệ thống tải dữ liệu Forum từ cơ sở dữ liệu.5. Hệ thống hiển thị trang Forum với các bài đăng.
Alternative flow	Nếu xảy ra lỗi trong quá trình tải dữ liệu Forum: <ul style="list-style-type: none">• Hệ thống hiển thị thông báo lỗi trên trang Forum.
Extension point	

Bảng 1.17. Đặc tả Use case - Thông kê Forum

Chương 3: Phân tích thiết kế hệ thống

3.1.3.18. Chặn/bỏ chặn người dùng



Hình 3.18. Sequence - Chặn/bỎ chặn người dùng

Name	Chặn/bỎ chặn người dùng
Brief description	Quản lý chặn hoặc bỏ chặn các tài khoản người dùng
Actor(s)	Quản lý

Chương 3: Phân tích thiết kế hệ thống

Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.
Post-conditions	<p>Trạng thái người dùng được cập nhật thành công trong cơ sở dữ liệu (chuyển thành blocked hoặc active).</p> <p>Lý do chặn được lưu lại (nếu có).</p> <p>Giao diện danh sách người dùng cập nhật trạng thái tương ứng.</p>
Flow of events	
Basic flow	<ol style="list-style-type: none"> Quản lý bấm vào menu "Users" trong thanh Sidebar. Hệ thống hiển thị danh sách người dùng với các thông tin: Email, Tên, Số điện thoại, Trạng thái, Lý do chặn và các nút hành động. Quản lý tìm kiếm người dùng bằng cách nhập từ khóa vào ô tìm kiếm. Quản lý bấm vào nút hành động (chặn hoặc bỏ chặn) tương ứng với người dùng. Hệ thống hiển thị hộp thoại xác nhận với thông tin người dùng và các tùy chọn: <ol style="list-style-type: none"> Nếu chặn: Yêu cầu nhập lý do chặn. Nếu bỏ chặn: Chỉ cần xác nhận. Quản lý xác nhận hành động: <ul style="list-style-type: none"> Bấm nút "Block" hoặc "Activate". Hệ thống cập nhật trạng thái người dùng trong cơ sở dữ liệu và hiển thị thông báo hành động thành công. Hộp thoại đóng lại, danh sách người dùng được cập nhật với trạng thái mới.
Alternative flow	<ol style="list-style-type: none"> Hệ thống không thể cập nhật trạng thái người dùng do lỗi kết nối

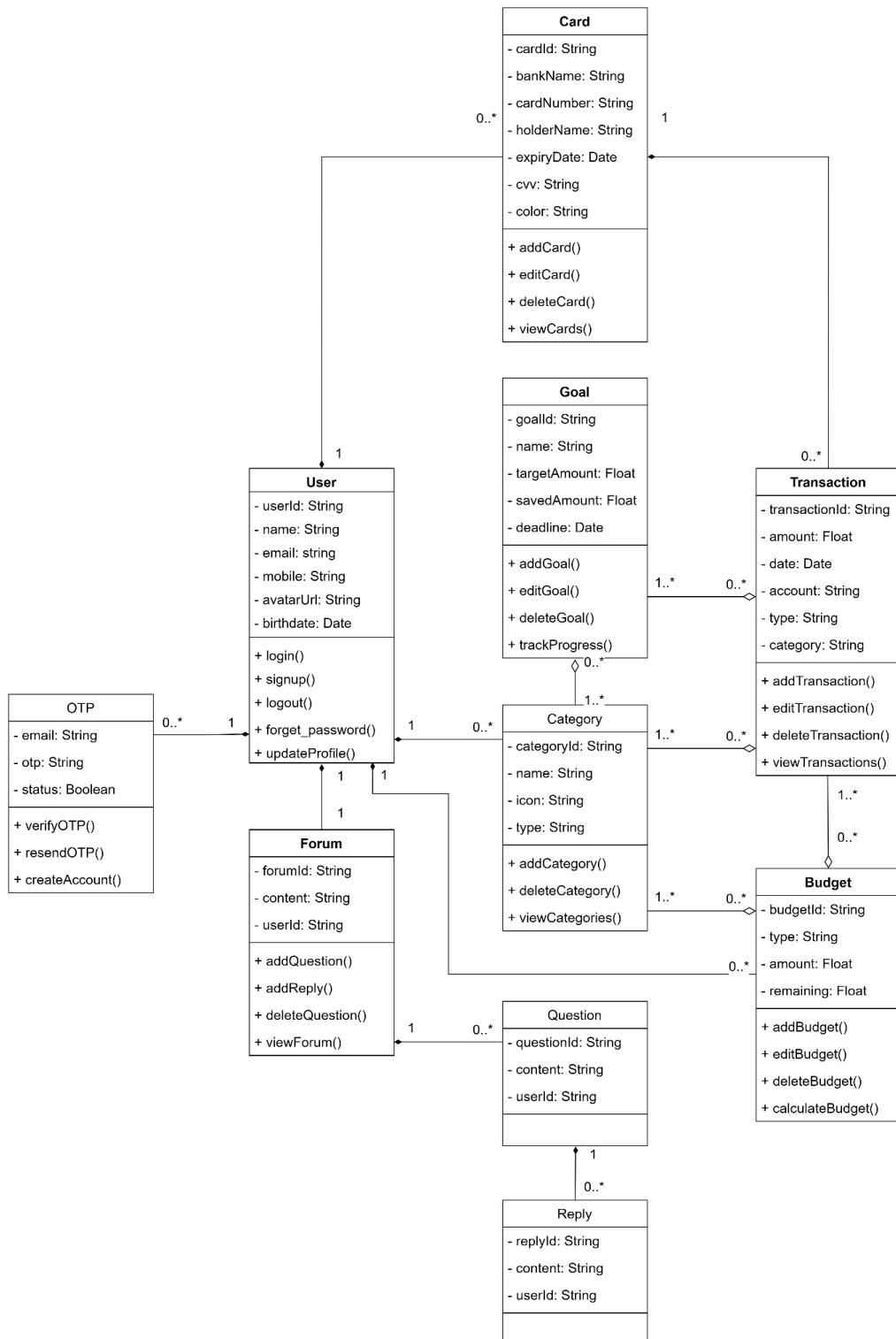
Chương 3: Phân tích thiết kế hệ thống

	<ul style="list-style-type: none">• Hệ thống hiển thị thông báo lỗi: "Không thể cập nhật trạng thái. Vui lòng thử lại."• Hành động không được thực hiện và giao diện giữ nguyên trạng thái cũ. <ol style="list-style-type: none">2. Quản lý hủy hành động trong hộp thoại<ol style="list-style-type: none">1. Quản lý bấm nút "Cancel" trong hộp thoại xác nhận.2. Hộp thoại đóng lại và không có thay đổi nào được thực hiện.
Extension point	

Bảng 1.19. Đặc tả Use case - Chặn/bỏ chặn người dùng

Chương 3: Phân tích thiết kế hệ thống

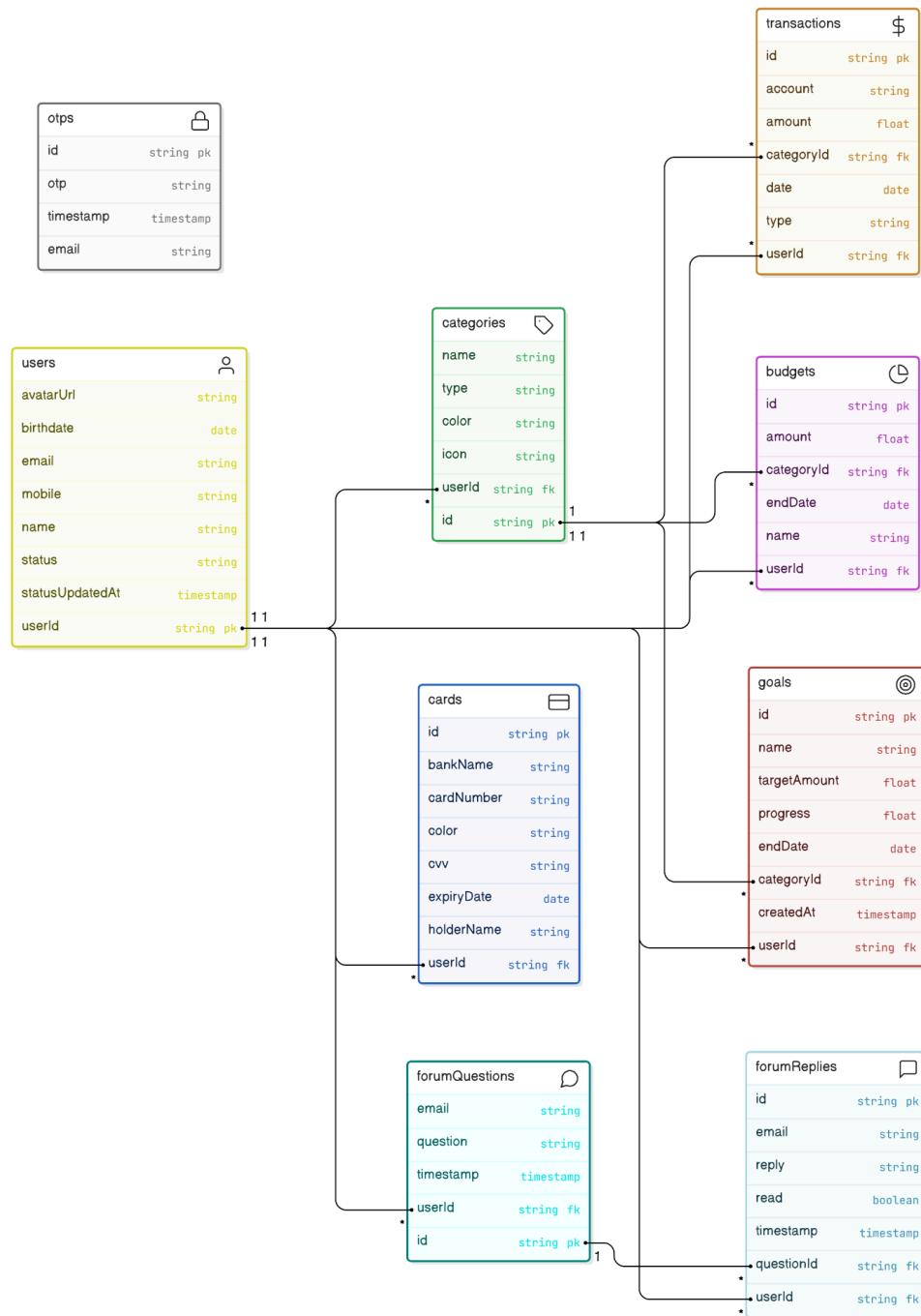
3.1.4. Class Diagram



Hình 4. Lược đồ Class Diagram

Chương 3: Phân tích thiết kế hệ thống

3.1.5. Lược đồ ERD



Hình 5. Lược đồ ERD

Chương 3: Phân tích thiết kế hệ thống

3.1.6. Phân tích lược đồ ERD

Bảng users

- **Mục đích:** Lưu trữ thông tin người dùng trong hệ thống.
- **Các trường:**
 - userid (Primary Key): Định danh duy nhất cho người dùng.
 - name: Tên của người dùng.
 - email: Email của người dùng.
 - avatarUrl: URL ảnh đại diện của người dùng.
 - mobile: Số điện thoại.
 - birthdate: Ngày sinh.
 - status: Trạng thái tài khoản (ví dụ: active, inactive).
 - statusUpdatedAt: Thời gian cập nhật trạng thái.

Bảng categories

- **Mục đích:** Lưu trữ danh mục chi tiêu hoặc thu nhập.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho danh mục.
 - name: Tên danh mục (ví dụ: Food, Transport).
 - type: Loại danh mục (Expense hoặc Income).
 - color: Màu sắc đại diện danh mục.
 - icon: Biểu tượng đại diện danh mục.
 - userid (Foreign Key): Người dùng sở hữu danh mục.

Bảng transactions

- **Mục đích:** Lưu trữ các giao dịch của người dùng.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho giao dịch.

Chương 3: Phân tích thiết kế hệ thống

- account: Tài khoản thực hiện giao dịch.
- amount: Số tiền của giao dịch.
- date: Ngày thực hiện giao dịch.
- type: Loại giao dịch (Expense hoặc Income).
- userid (Foreign Key): Người dùng thực hiện giao dịch.
- categoryId (Foreign Key): Danh mục liên kết với giao dịch.

Bảng budgets

- **Mục đích:** Lưu trữ ngân sách người dùng thiết lập.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho ngân sách.
 - name: Tên ngân sách.
 - amount: Số tiền ngân sách.
 - endDate: Ngày kết thúc ngân sách.
 - categoryId (Foreign Key): Danh mục quan đến ngân sách.
 - userid (Foreign Key): Người dùng thiết lập ngân sách.

Bảng goals

- **Mục đích:** Quản lý các mục tiêu tài chính của người dùng.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho mục tiêu.
 - name: Tên mục tiêu.
 - targetAmount: Số tiền mục tiêu.
 - progress: Tiết độ đạt được.
 - endDate: Ngày kết thúc mục tiêu.
 - createdAt: Thời gian tạo mục tiêu.
 - categoryId (Foreign Key): Danh mục liên kết với mục tiêu.
 - userid (Foreign Key): Người dùng sở hữu mục tiêu.

Chương 3: Phân tích thiết kế hệ thống

Bảng cards

- **Mục đích:** Lưu trữ thông tin thẻ thanh toán của người dùng.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho thẻ.
 - bankName: Tên ngân hàng.
 - cardNumber: Số thẻ.
 - color: Màu đại diện.
 - cvv: Mã bảo mật thẻ.
 - expiryDate: Ngày hết hạn.
 - holderName: Tên chủ thẻ.
 - userid (Foreign Key): Người dùng sở hữu thẻ.

Bảng forumQuestions

- **Mục đích:** Lưu trữ các câu hỏi trong diễn đàn.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho câu hỏi.
 - email: Email người dùng đăng câu hỏi.
 - question: Nội dung câu hỏi.
 - timestamp: Thời gian đăng câu hỏi.
 - userid (Foreign Key): Người dùng liên quan đến câu hỏi.

Bảng forumReplies

- **Mục đích:** Lưu trữ các câu trả lời trong diễn đàn.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho câu trả lời.
 - email: Email người trả lời.
 - reply: Nội dung câu trả lời.

Chương 3: Phân tích thiết kế hệ thống

- read: Trạng thái đọc (true/false).
- timestamp: Thời gian trả lời.
- questionId (Foreign Key): Câu hỏi liên quan.
- userid (Foreign Key): Người dùng liên quan đến câu trả lời.

Bảng otps

- **Mục đích:** Quản lý mã OTP để xác thực.
- **Các trường:**
 - id (Primary Key): Định danh duy nhất cho mã OTP.
 - otp: Mã OTP.
 - timestamp: Thời gian tạo mã OTP.
 - email: Email người dùng liên quan.

3.2. Kiến trúc hệ thống

3.2.1. Tổng quan:

Kiến trúc hệ thống của ứng dụng quản lý tài chính di động bao gồm nhiều thành phần tương tác với nhau để xử lý giao diện người dùng, backend và lưu trữ dữ liệu. Hệ thống này được xây dựng sử dụng React Native, Firebase và các dịch vụ bên thứ ba như Mailjet để gửi OTP và Cloudinary để lưu trữ tệp.

3.2.2. Các thành phần:

- Frontend (React Native):
 - React Native sẽ được sử dụng để xây dựng ứng dụng di động, mang đến giao diện người dùng tương tác và mượt mà.
 - Tailwind CSS sẽ hỗ trợ thiết kế giao diện, giúp tạo ra các thành phần UI linh hoạt và dễ dàng tùy chỉnh.

Chương 3: Phân tích thiết kế hệ thống

- Ứng dụng di động sẽ kết nối và giao tiếp với backend qua các API, được Firebase quản lý.
- Backend (Firebase):
 - Firebase Authentication SDK sẽ quản lý việc đăng ký, đăng nhập và xác thực người dùng.
 - Firebase Storage SDK sẽ lưu trữ và quản lý ảnh đại diện của người dùng.
 - Firebase Realtime Database SDK sẽ lưu trữ và quản lý thông tin tài chính cá nhân, bao gồm chi tiêu, thu nhập, ngân sách, v.v.
 - Firebase Cloud Functions API sẽ xử lý các logic backend như gửi OTP xác thực qua email sử dụng dịch vụ Mailjet.
 - Firebase Token sẽ giúp bảo mật các API và xác thực người dùng.
- Dịch vụ bên thứ ba:
 - Mailjet sẽ được sử dụng để gửi email OTP cho người dùng nhằm xác thực tài khoản.

3.2.3. Chi tiết kiến trúc:

- Frontend (React Native):
 - React Native sẽ được sử dụng để xây dựng giao diện người dùng cho ứng dụng di động. Giao diện sẽ được thiết kế với Tailwind CSS để tạo ra các thành phần UI nhanh chóng và dễ dàng tùy chỉnh.
 - Ứng dụng di động sẽ giao tiếp với Firebase Backend thông qua các API để lấy và gửi dữ liệu, bao gồm thông tin người dùng, dữ liệu tài chính, v.v.
- Backend (Firebase):
 - Firebase Authentication SDK sẽ đảm nhận việc quản lý người dùng, bao gồm đăng ký, đăng nhập và xác thực người dùng qua Firebase Authentication.

Chương 3: Phân tích thiết kế hệ thống

- Firebase Realtime Database SDK sẽ là nơi lưu trữ tất cả thông tin liên quan đến tài chính cá nhân, bao gồm các khoản thu, chi, ngân sách, và lịch sử giao dịch.
 - Firebase Storage SDK sẽ được sử dụng để lưu trữ các tệp media như ảnh đại diện của người dùng hoặc các tệp tài liệu.
 - Firebase Cloud Functions API sẽ xử lý các yêu cầu backend, như gửi OTP qua Mailjet khi người dùng yêu cầu xác thực tài khoản.
 - Firebase Token sẽ giúp bảo mật các giao dịch giữa frontend và backend, đảm bảo rằng chỉ những người dùng đã xác thực mới có quyền truy cập vào dữ liệu.
- Dịch vụ bên thứ ba:
 - Mailjet sẽ được tích hợp để gửi email OTP cho người dùng khi họ đăng ký tài khoản hoặc yêu cầu xác thực thông tin.

3.2.4. Triển khai hệ thống:

- Frontend sẽ được triển khai trên Vercel để phục vụ người dùng di động.
- Backend sẽ được triển khai trên AWS với Firebase làm nền tảng backend để xử lý và lưu trữ dữ liệu.
- Firebase sẽ được sử dụng để triển khai toàn bộ hệ thống backend, từ xác thực người dùng đến lưu trữ dữ liệu và xử lý các chức năng liên quan đến giao dịch tài chính.

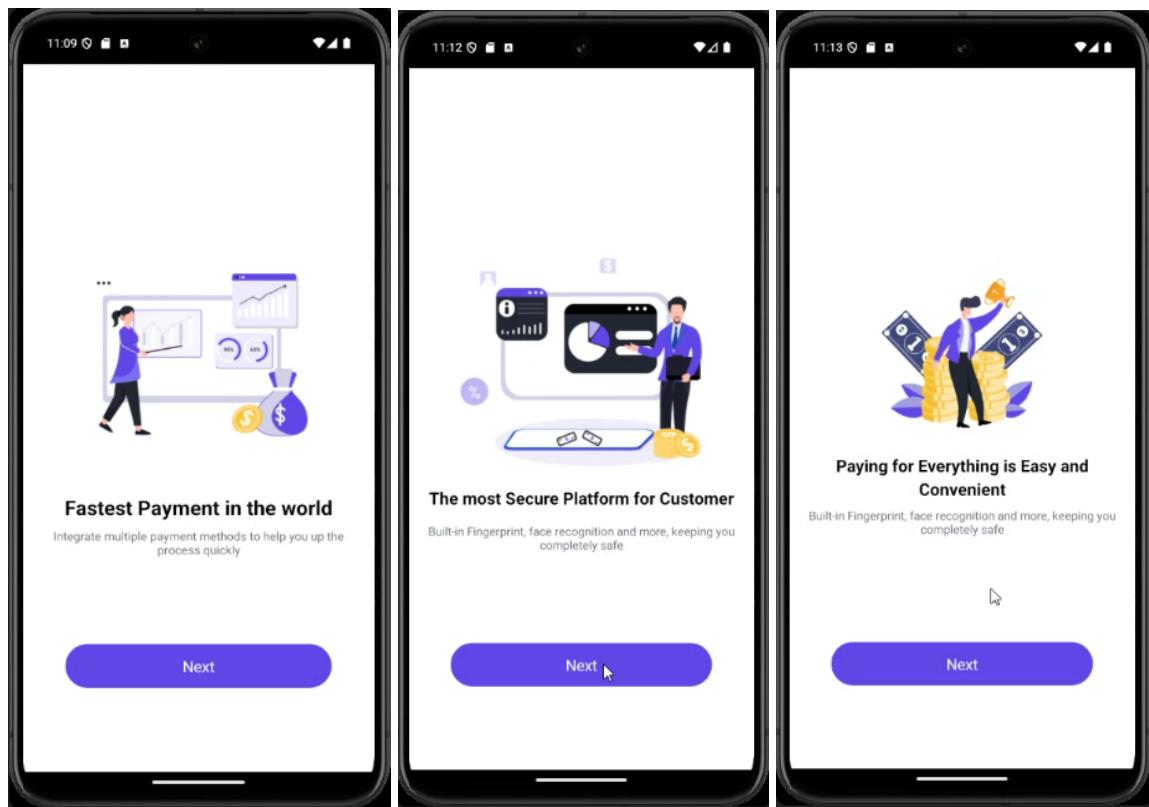
3.3. Thiết kế giao diện

3.3.1. Giao diện người dùng

3.3.1.1. Splash

Người dùng mới truy cập vào App sẽ xuất hiện các Splash để giới thiệu về ứng dụng.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.1 & 6.2 & 6.3. Splash.

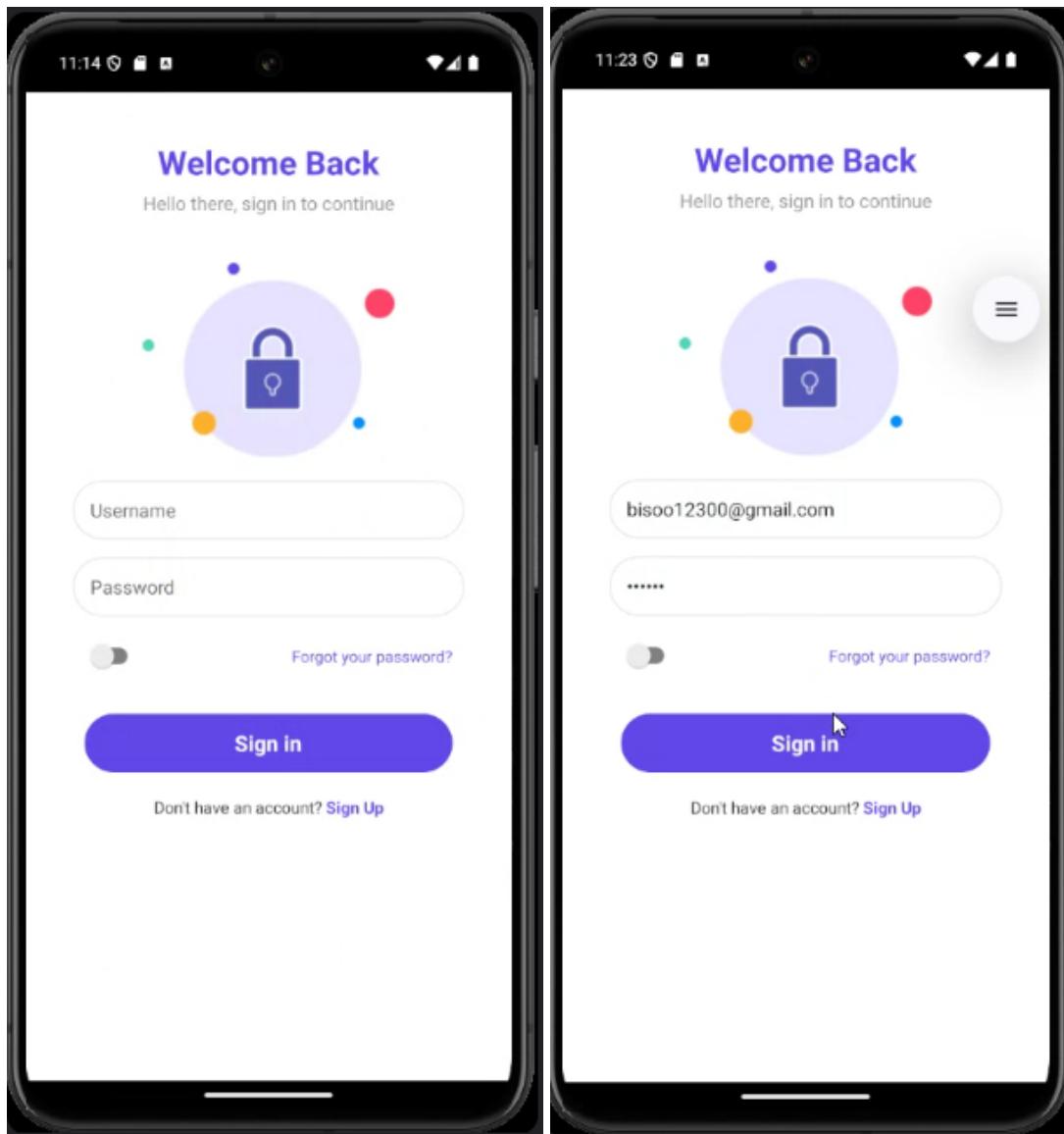
Vị trí	Tên	Chức năng
1	Nút “Next”	Điều hướng đến trang tiếp theo

Bảng 2.1. Mô tả Splash.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.2. Đăng nhập

Đầu tiên người dùng sẽ được đưa đến trang đăng nhập.



Hình 6.4 & 6.5. Đăng nhập.

Vị trí	Tên	Chức năng
1	Khung “Username”	nơi điền email đăng nhập
2	Khung “Password”	nơi điền mật khẩu

Chương 3: Phân tích thiết kế hệ thống

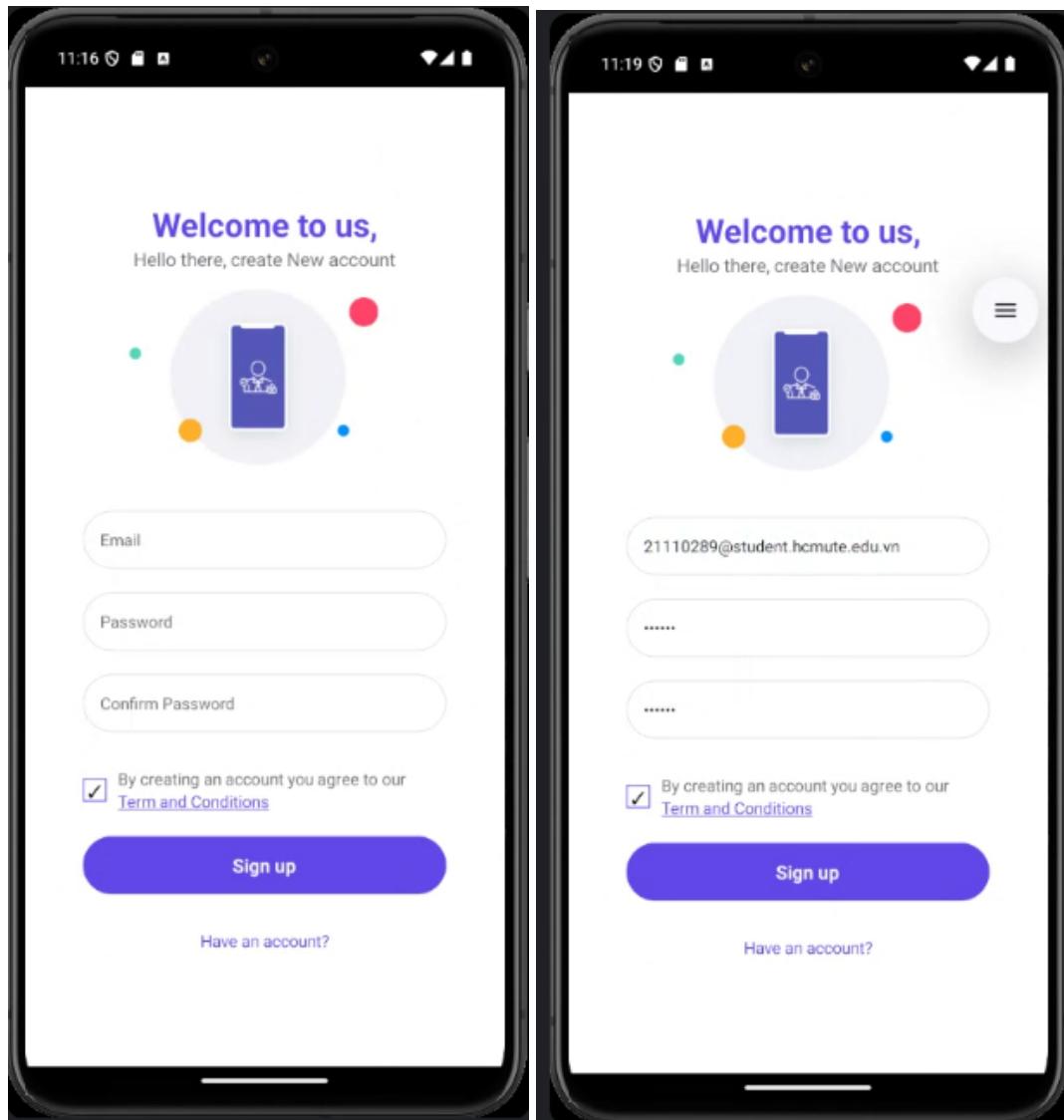
3	“Forgot your password”	Điều hướng đến trang quên mật khẩu
4	Nút “Sign in”	Sau khi điền thông tin đăng nhập, bấm vào sẽ thực hiện đăng nhập và nếu đăng nhập thành công sẽ đưa người dùng đến trang “Home”
5	“Sign Up”	Đưa người dùng đến trang đăng ký

Bảng 2.2. Mô tả trang đăng nhập.

3.3.1.3. Đăng ký

Sau khi bấm vào đăng ký ở trang đăng nhập, người dùng sẽ được đưa đến trang đăng ký.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.6 & 6.7. Đăng ký.

Vị trí	Tên	Chức năng
1	Khung “Email”	Nơi điền email đăng nhập
2	Khung “Password”	Nơi điền mật khẩu
3	Khung “Confirm Password”	Nhập lại mật khẩu
4	Term and conditions	Đưa người dùng đến trang

Chương 3: Phân tích thiết kế hệ thống

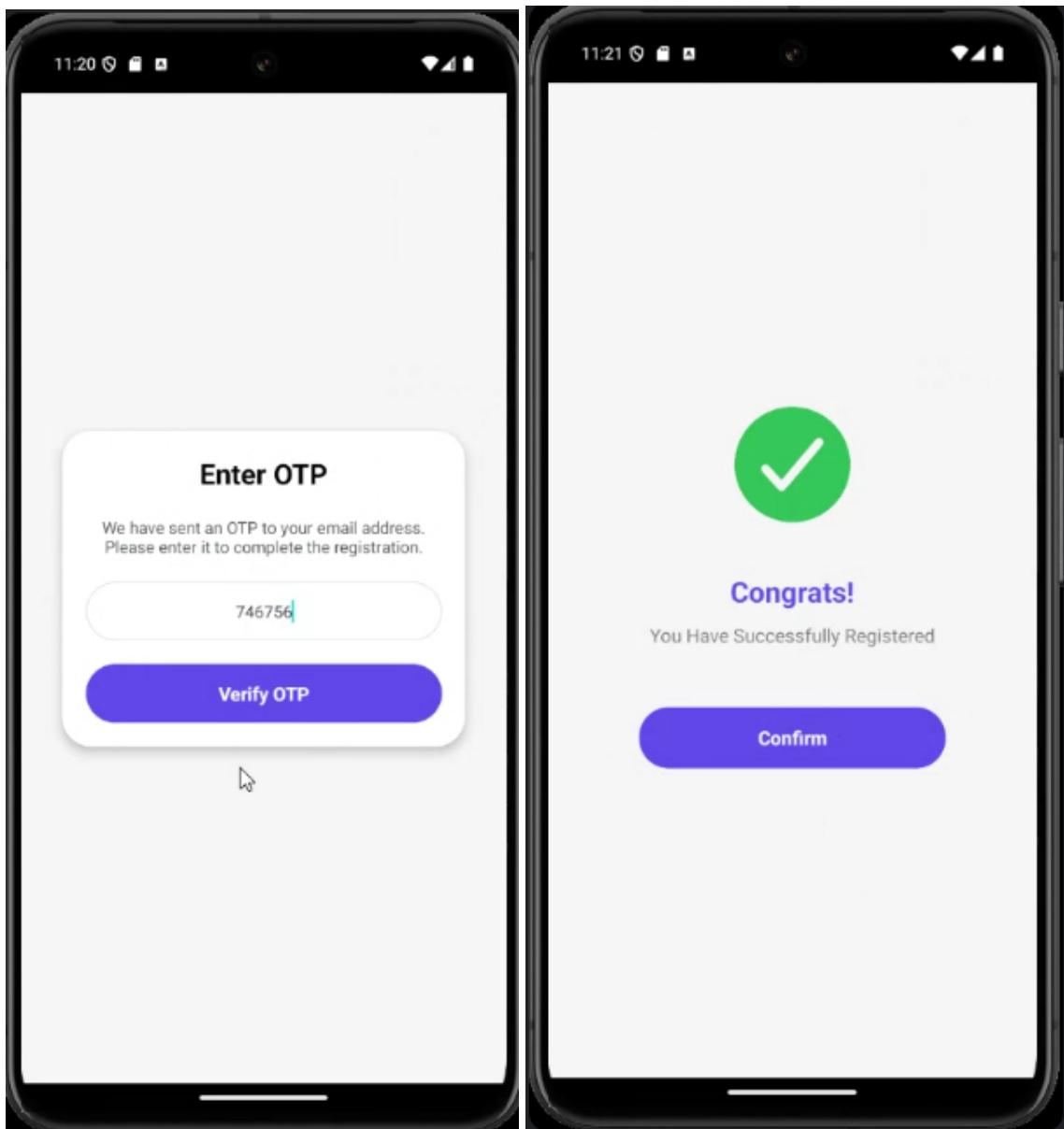
		chính sách
5	Nút “Sign up”	Xác nhận đăng ký sau khi đã điền thông tin, đưa đến trang xác nhận OTP
6	“Have an account”	Đưa người dùng đến trang đăng nhập

Bảng 2.3. Mô tả trang đăng ký.

3.3.1.4. Xác nhận OTP

Đối với các chức năng cần xác nhận OTP để thực hiện, người dùng sẽ được yêu cầu nhập OTP đã được gửi tới email vào để xác nhận.

Chương 3: Phân tích thiết kế hệ thống



Chương 3: Phân tích thiết kế hệ thống

Hình 6.8 & 6.9. Đăng ký.

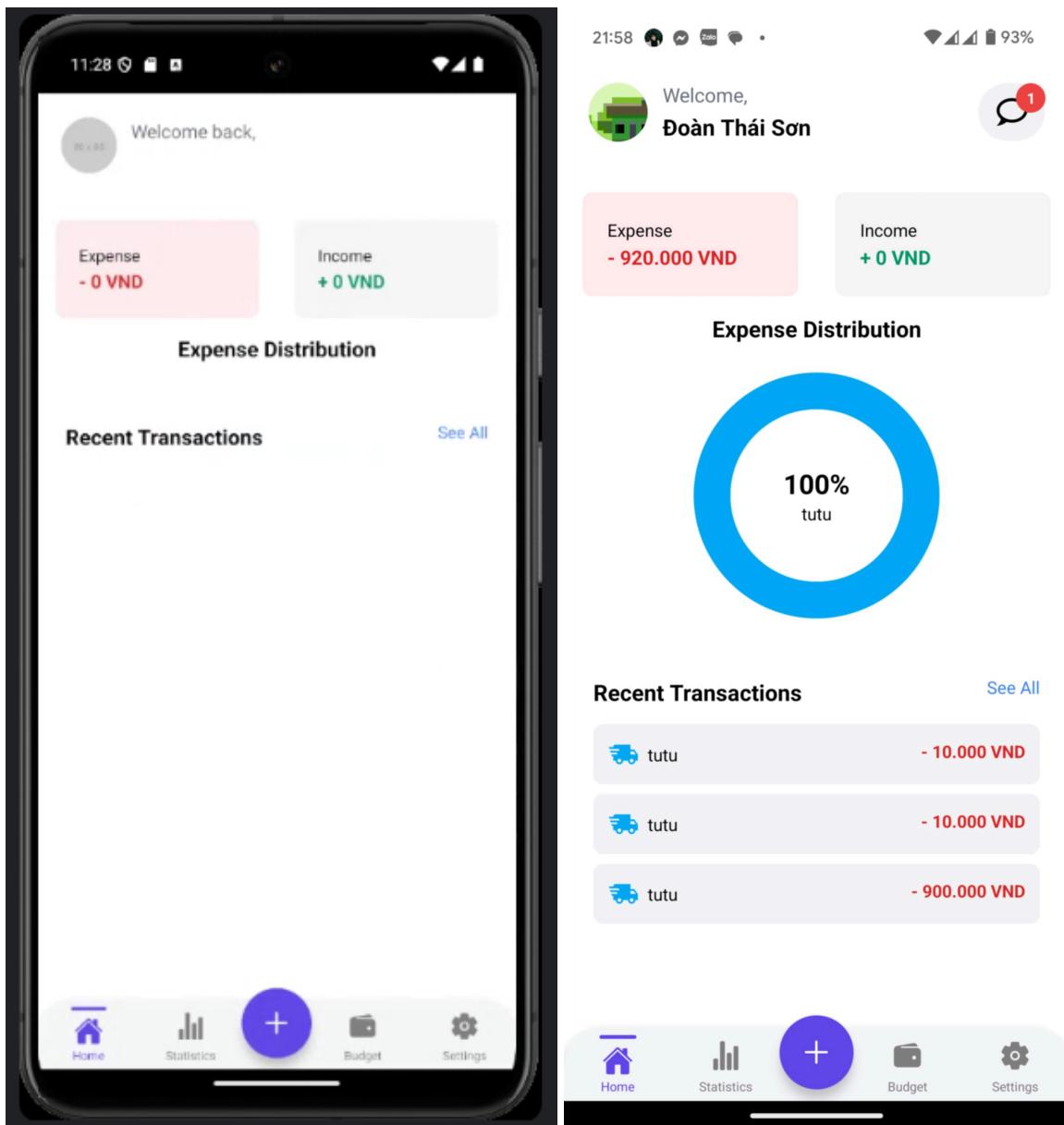
Vị trí	Tên	Chức năng
1	Khung “OTP”	Nơi điền OTP
2	Nút “Verify OTP”	Nhấn để xác nhận OTP, nếu đúng sẽ thực hiện yêu cầu trước đó.
3	Nút “Confirm”	Sau khi xác thực thành công OTP, người dùng được đưa đến trang chúc mừng, nhấn “Confirm” để quay lại

Bảng 2.4. Mô tả xác nhận OTP.

3.3.1.5. Trang chủ

Sau khi đăng nhập người dùng sẽ được đưa đến trang chủ.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.10 & 6.11. Trang chủ.

Vị trí	Tên	Chức năng
1	Thông tin người dùng	Hiển thị thông tin người dùng hiện tại
2	Diễn đàn	Nhấn vào biểu tượng đám

Chương 3: Phân tích thiết kế hệ thống

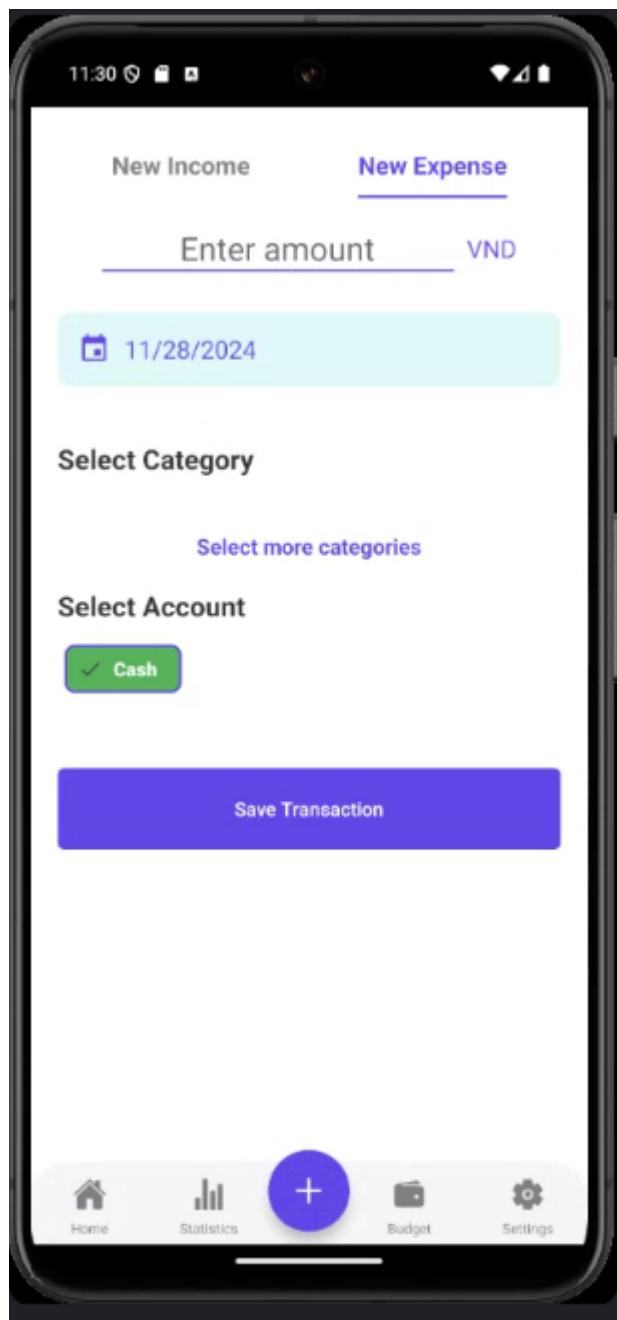
		mây phía góc bên phải để truy cập vào diễn đàn hỏi đáp
3	“Expense”	Nhấn vào để hiển thị biểu đồ vòng thống kê giao dịch chi tiêu của tháng này, cùng với các chi tiêu vừa được tạo phía bên dưới
4	“Income”	Nhấn vào để hiển thị biểu đồ vòng thống kê giao dịch thu nhập của tháng này, cùng với các thu nhập vừa được tạo phía bên dưới
5	“See All”	Đưa người dùng đến trang quản lý giao dịch
6	Thanh điều hướng	Có thể bấm vào để đưa người dùng đến Trang chủ, trang thống kê, tạo giao dịch mới, trang ngân sách, trang cài đặt

Bảng 2.5. Mô tả trang chủ.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.6. Thêm giao dịch

Bằng cách bấm vào nút “+” chính giữa thanh điều hướng, người dùng sẽ được đưa đến trang thêm giao dịch mới.



Hình 6.12. Thêm giao dịch.

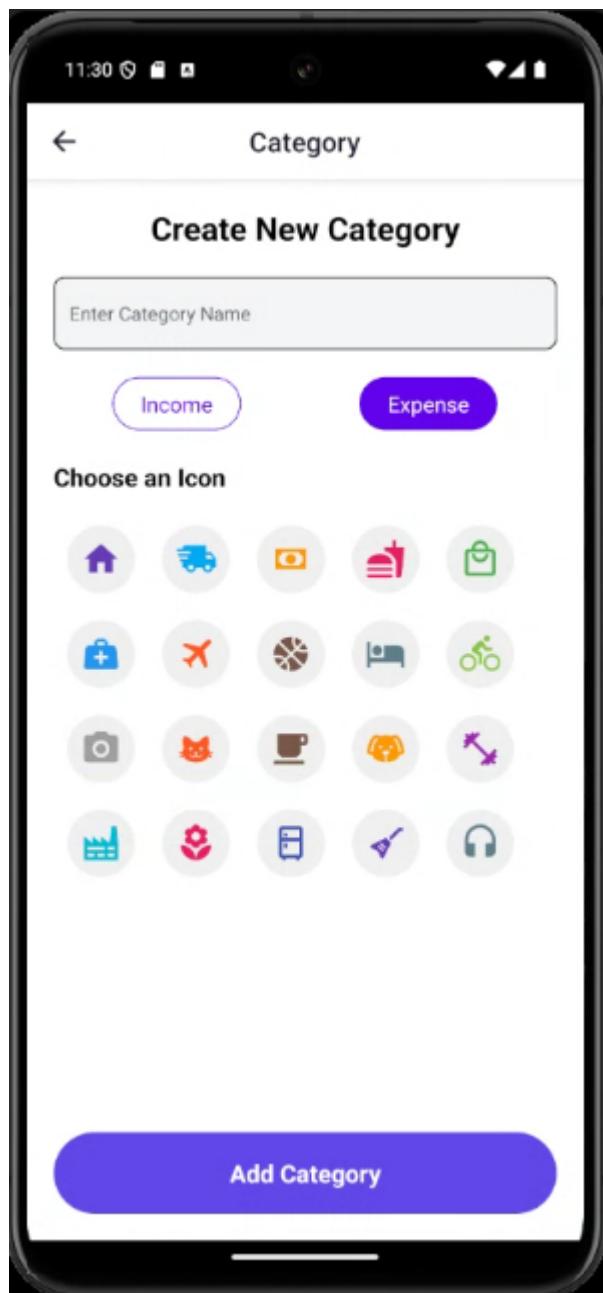
Chương 3: Phân tích thiết kế hệ thống

Vị trí	Tên	Chức năng
1	Số lượng	Điền số tiền của giao dịch
2	Ngày	Ngày đã thực hiện giao dịch
3	Chọn danh mục	Chọn danh mục của giao dịch hoặc tạo mới nếu chưa có bằng “select more categories”
4	Chọn nguồn tiền	Chọn nguồn tiền đã giao dịch.
5	Nút lưu	Tiến hành lưu giao dịch dựa trên thông tin đã cung cấp
6	Thanh điều hướng	Có thể bấm vào để đưa người dùng đến Trang chủ, trang thống kê, tạo giao dịch mới, trang ngân sách, trang cài đặt

Bảng 2.6. Mô tả trang thêm giao dịch mới.

3.3.1.7. Thêm danh mục mới

Bằng cách bấm vào Select more categories hoặc bấm “+” trong trang manage categories, người dùng sẽ được đưa đến trang thêm category mới.



Hình 6.13. Thêm danh mục mới.

Chương 3: Phân tích thiết kế hệ thống

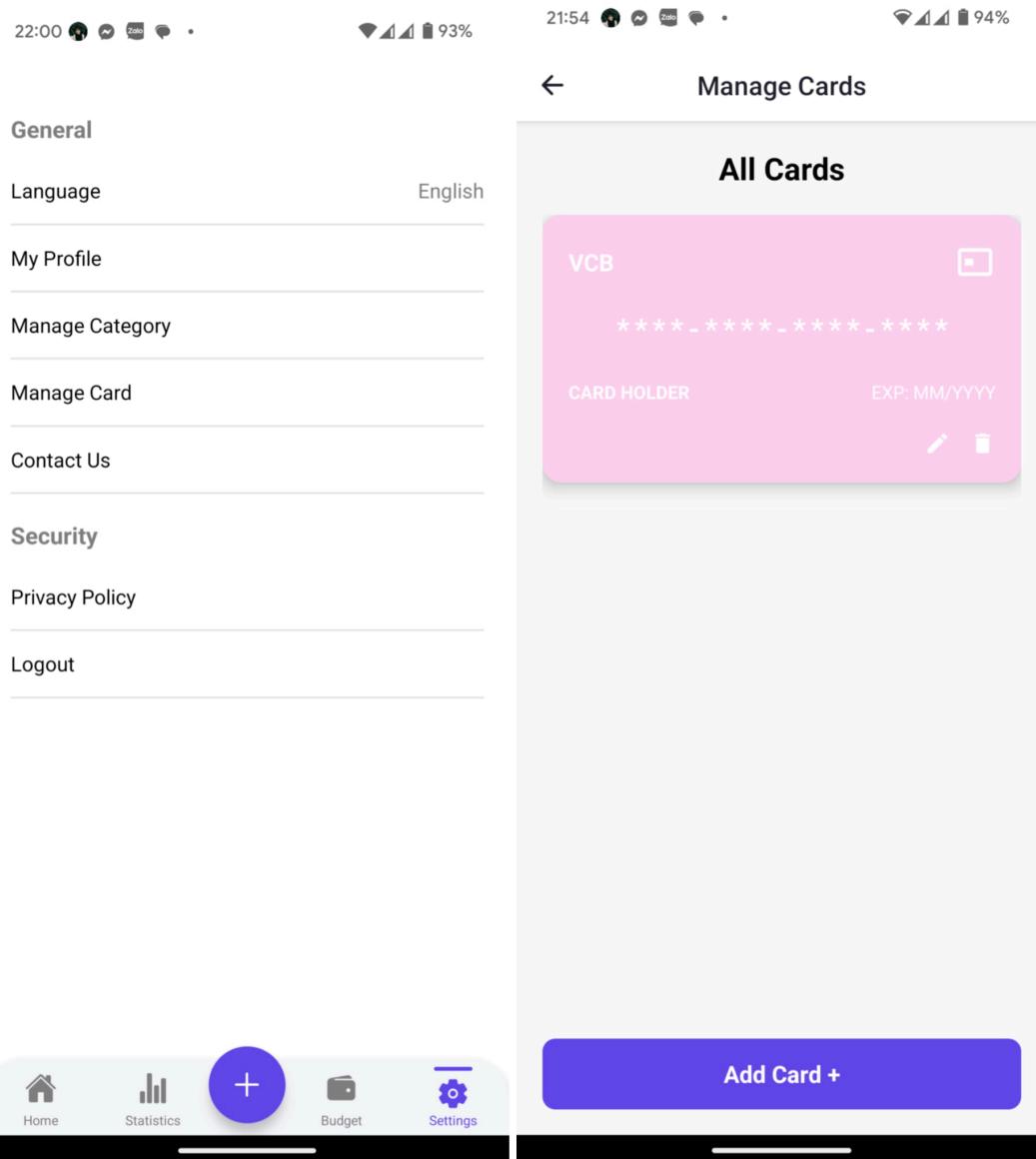
Vị trí	Tên	Chức năng
1	Tên danh mục	Nơi điền tên danh mục mới
2	Loại danh mục	Là chi tiêu hay thu nhập
3	Biểu tượng danh mục	Chọn biểu tượng cho danh mục
4	Nút thêm	Xác nhận và tiến hành thêm danh mục.

Bảng 2.7. Mô tả trang thêm danh mục mới.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.8. Quản lý thẻ

Ở phần account, người dùng sẽ chọn thẻ ngân hàng hoặc tiền mặt, để tiến hành thêm thẻ ta truy cập vào trang cài đặt và bấm vào Manage Card.



Chương 3: Phân tích thiết kế hệ thống

Hình 6.14 & 6.15. Quản lý thẻ.

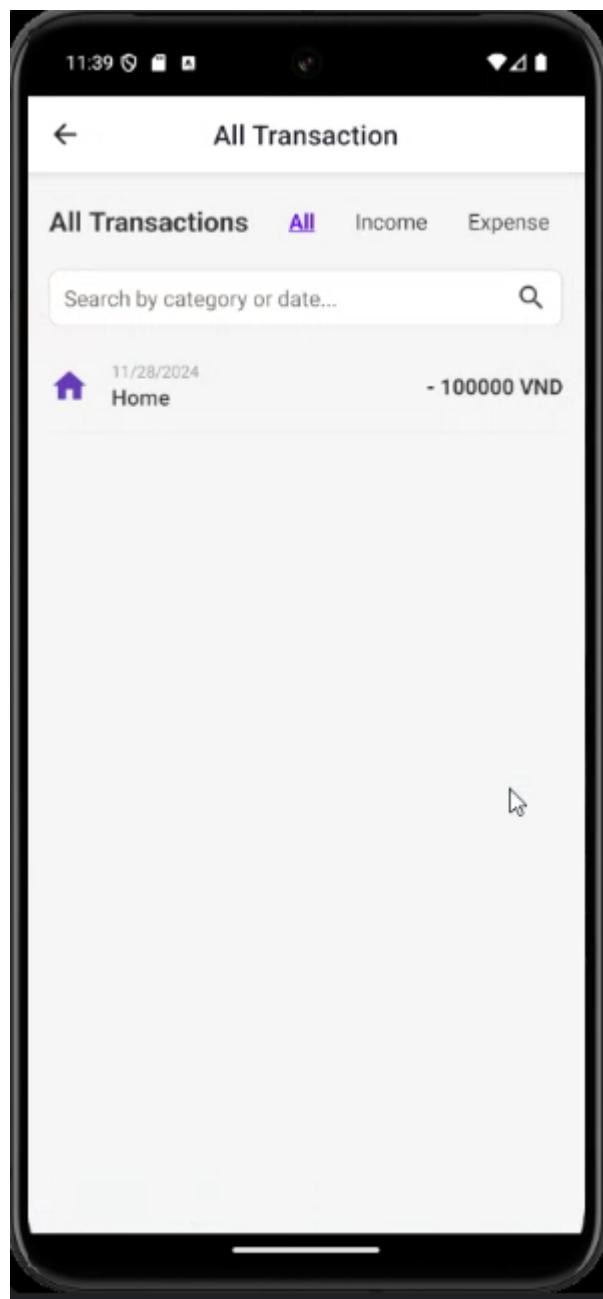
Vị trí	Tên	Chức năng
1	Thông tin thẻ	Hiển thị các thông tin của thẻ
2	Nút sửa	Đưa người dùng đến trang chỉnh sửa thông tin của thẻ đó
3	Nút xóa	Xóa thẻ
4	Nút thêm thẻ	Đưa người dùng đến trang thêm thẻ

Bảng 2.8. Mô tả quản lý thẻ.

3.3.1.9. Quản lý giao dịch

Sau khi bấm vào See All ở trang Home, người dùng được đưa đến trang All Transaction, nơi đây sẽ hiển thị tất cả các giao dịch.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.16. Quản lý giao dịch.

Vị trí	Tên	Chức năng
1	“All”, “Income”,	Các nút thực hiện chức

Chương 3: Phân tích thiết kế hệ thống

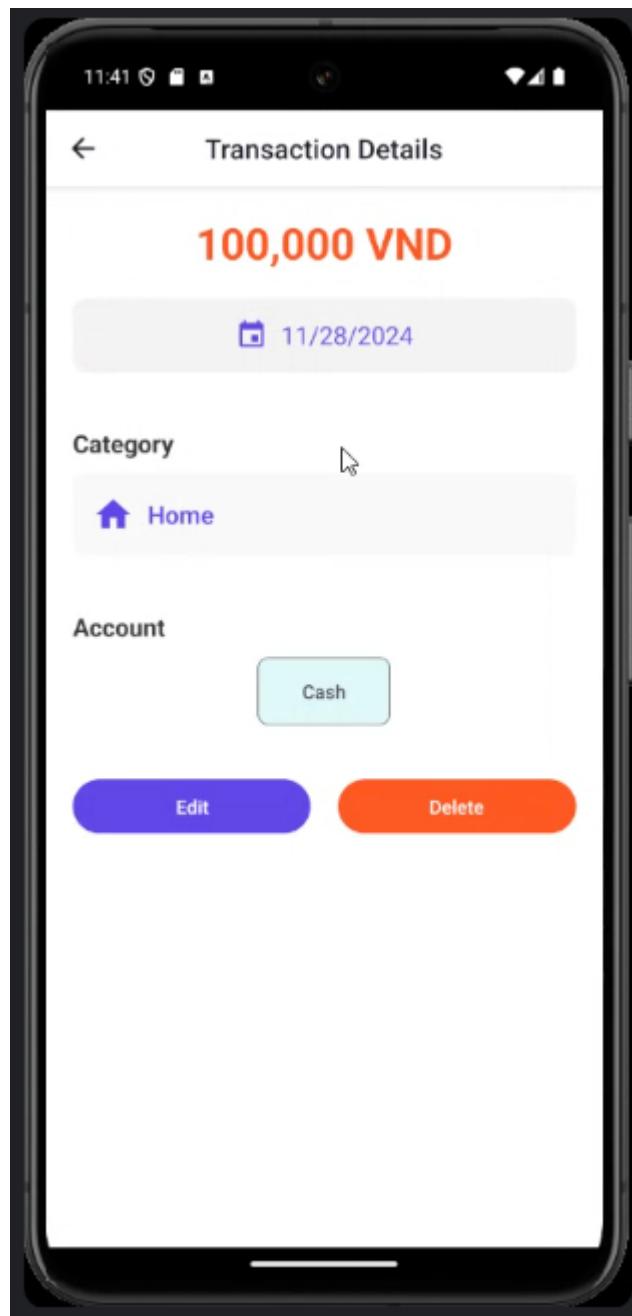
	“Expense”	năng lọc, hiển thị theo tất cả, chi tiêu hay thu nhập.
2	Thanh tìm kiếm	Thực hiện chức năng tìm kiếm giao dịch theo danh mục hoặc ngày.
3	Danh sách các giao dịch	Hiển thị các giao dịch hiện có của người dùng, có thể bấm vào để đưa người dùng đến trang chi tiết giao dịch hoặc giữ để chọn và xóa giao dịch, có thể chọn nhiều và xóa cùng lúc

Bảng 2.9. Mô tả trang quản lý giao dịch.

3.3.1.10. Chi tiết giao dịch

Để xem chi tiết của transaction ta bấm vào transaction đó.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.17. Chi tiết giao dịch.

Vị trí	Tên	Chức năng
1	Số tiền	Hiện số tiền của giao dịch

Chương 3: Phân tích thiết kế hệ thống

2	Ngày	Hiện ngày thực hiện của giao dịch.
3	Danh mục	Hiển thị danh mục của giao dịch.
4	Nguồn	Hiện nguồn của giao dịch đó là ở đâu.
5	Nút “Edit”	Đưa người dùng đến trang chỉnh sửa giao dịch này.
6	Nút “Delete”	Xóa giao dịch này.

Bảng 2.10. Mô tả chi tiết giao dịch.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.11. Thống kê giao dịch

Đến với chức năng thống kê, ta bấm vào nút Statistics bên dưới thanh điều hướng. Người dùng được đưa đến trang thống kê, ở đây sẽ thống kê các giao dịch đã tạo, có thể xem theo ngày, tháng.



Hình 6.18. Thống kê giao dịch.

Chương 3: Phân tích thiết kế hệ thống

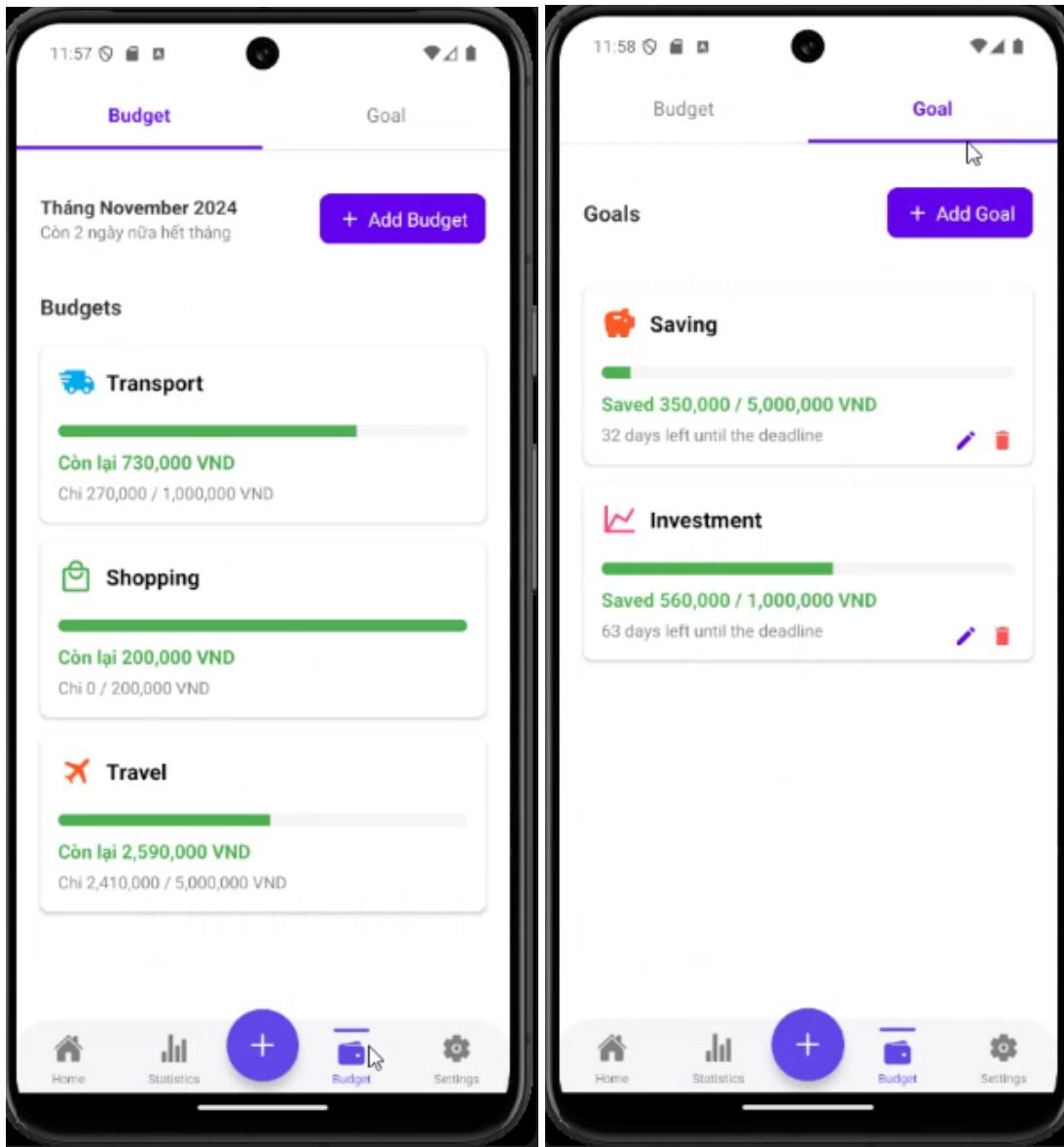
Vị trí	Tên	Chức năng
1	Lịch	Hiển thị các ngày trong tháng, có thể chọn để xem thống kê giao dịch trong ngày đó.
2	“Previous” và “Next”	Lùi về tháng trước đó hoặc tiến tới tháng tiếp theo.
3	Số tiền	Thông kê số tiền chi tiêu và thu nhập, từ đó đưa ra tổng
4	Các biểu đồ	Hiện biểu đồ thống kê thói quen chi tiêu và thu nhập
5	Thanh điều hướng	Có thể bấm vào để đưa người dùng đến Trang chủ, trang thống kê, tạo giao dịch mới, trang ngân sách, trang cài đặt

Bảng 2.11. Thông kê giao dịch.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.12. Ngân sách và mục tiêu tài chính

Để thực hiện chức năng tiếp theo, người dùng bấm vào nút Budget trên thanh điều hướng bên dưới để được đưa đến trang Budget.



Hình 6.19 & 6.20. Ngân sách và mục tiêu.

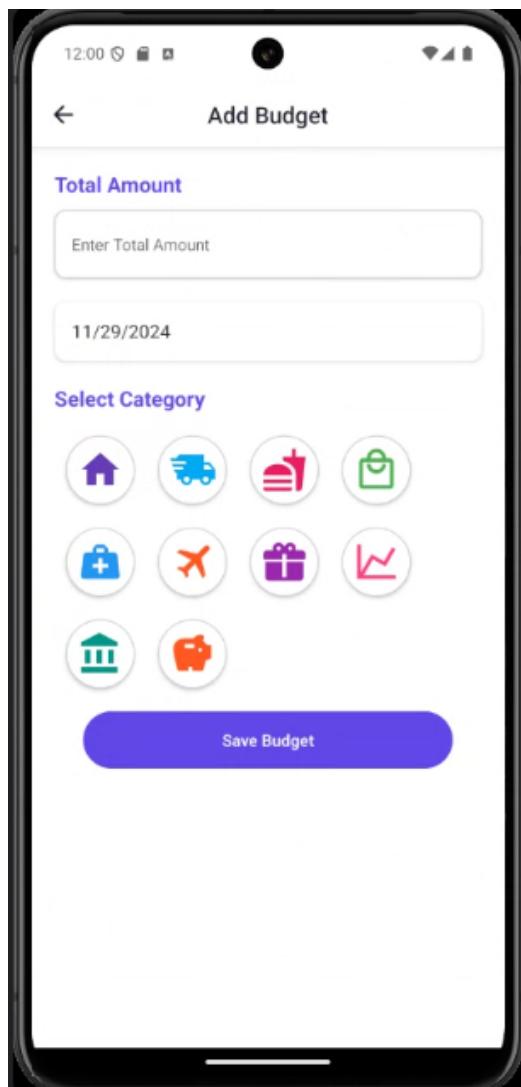
Chương 3: Phân tích thiết kế hệ thống

Vị trí	Tên	Chức năng
1	“Budget”, “Goal”	Chuyển đổi giữa ngân sách và mục tiêu.
2	Nút thêm	đưa người dùng đến trang thêm để thực hiện thêm ngân sách hoặc mục tiêu mới.
3	Danh sách	Hiển thị tất cả các ngân sách hoặc mục tiêu của người dùng.
4	Các Budget trên danh sách	Đối với Budget khi bấm vào sẽ đưa người dùng đến trang chi tiết Budget, nơi ta có thể sửa hoặc xóa Budget.
5	Các Goal trên danh sách	Đối với Goal, có nút sửa và xóa ngay trên Goal, bấm vào Goal để thêm tiền vào mục tiêu đó.

Bảng 2.12. Mô tả ngân sách và mục tiêu.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.13. Thêm ngân sách



Hình 6.21. Thêm ngân sách.

Khi bấm vào Add Budget và Add Goal người dùng thêm thông tin và bấm save để lưu.

Vị trí	Tên	Chức năng
1	Nút "Back"	Quay về trang trước
2	Trường "Total Amount"	Nhập số tiền tổng ngân sách mà người dùng muốn

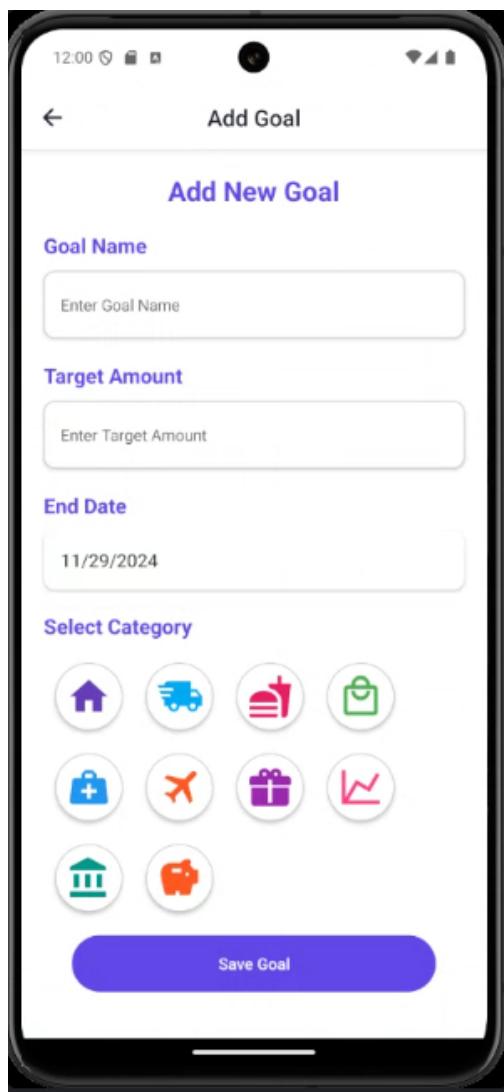
Chương 3: Phân tích thiết kế hệ thống

		đặt.
3	Trường "Ngày bắt đầu"	Chọn ngày bắt đầu cho ngân sách (mặc định là ngày hiện thị sẵn).
4	Mục "Select Category"	Hiển thị danh sách các danh mục, cho phép người dùng chọn một danh mục phù hợp cho ngân sách.
5	Nút "Save Budget"	Nút bấm màu tím ở dưới cùng, cho phép người dùng lưu ngân sách mới.

Bảng 2.13. Mô tả thêm ngân sách.

3.3.1.14. Thêm mục tiêu tài chính

Chương 3: Phân tích thiết kế hệ thống



Hình 6.22. Thêm mục tiêu tài chính.

Vị trí	Tên	Chức năng
1	Nút "Back"	Quay về trang trước
2	Trường "Goal Name"	Đặt tên cho mục tiêu (Nếu không thì lấy theo tên danh mục)
3	Trường "Target Amount"	Nhập số tiền của mục tiêu
4	Trường "Ngày kết thúc"	Chọn ngày kết thúc cho mục tiêu (mặc định là ngày

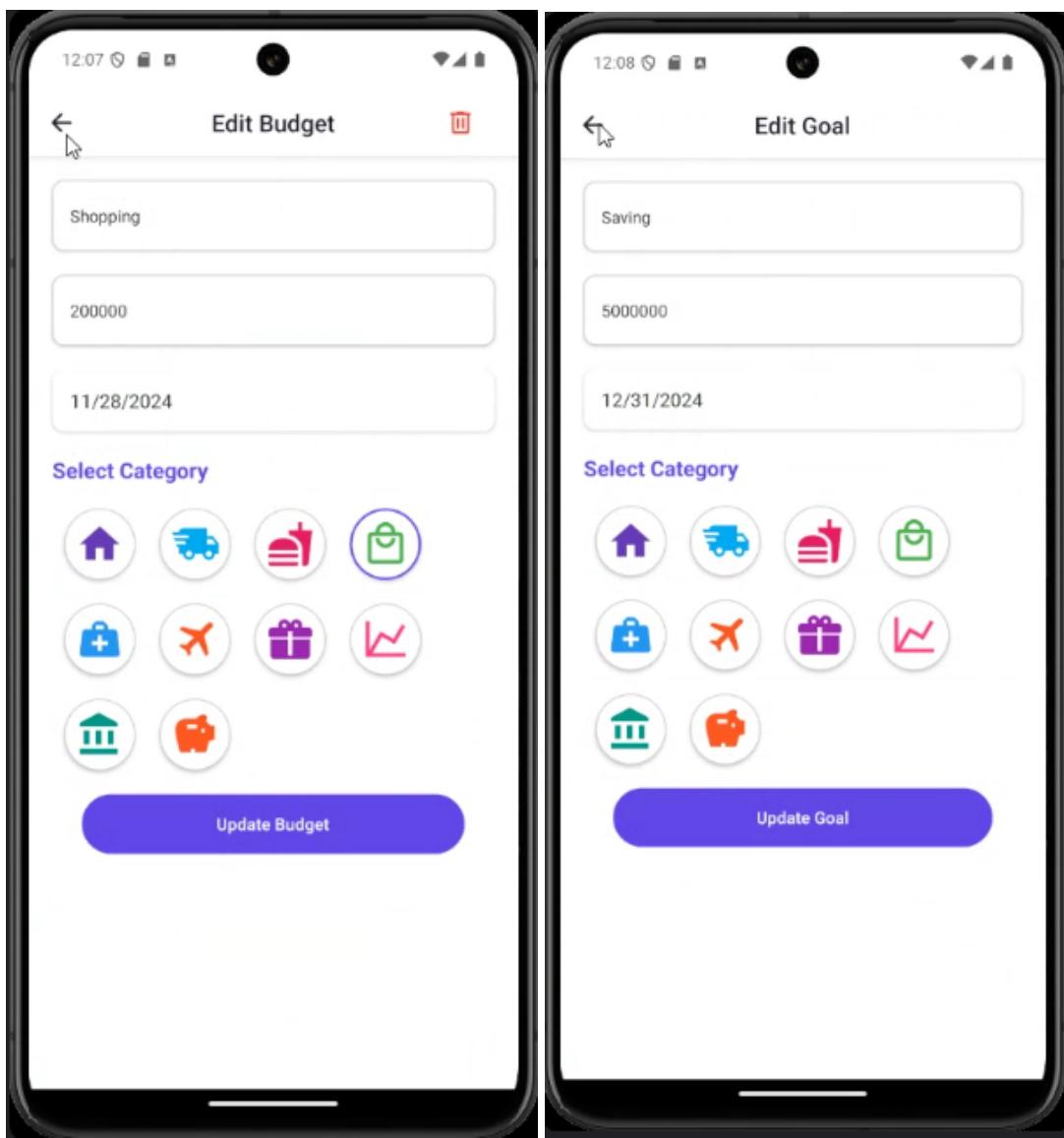
Chương 3: Phân tích thiết kế hệ thống

		hiển thị sẵn).
5	Mục "Select Category"	Hiển thị danh sách các danh mục, cho phép người dùng chọn một danh mục phù hợp cho ngân sách.
6	Nút "Save Goal"	Nút bấm màu tím ở dưới cùng, cho phép người dùng lưu mục tiêu.

Bảng 2.14. Thêm mục tiêu tài chính.

3.3.1.15. Chính sửa ngân sách và mục tiêu tài chính

Chương 3: Phân tích thiết kế hệ thống



Hình 6.23 & 6.24. Thêm danh mục mới.

Vị trí	Tên	Chức năng
1	Nút "Back"	Quay lại trang trước đó
2	Nút "Delete"	Xoá ngân sách / mục tiêu hiện tại
3	Trường "Tên"	Cho phép người dùng chỉnh sửa tên của ngân sách / mục tiêu.

Chương 3: Phân tích thiết kế hệ thống

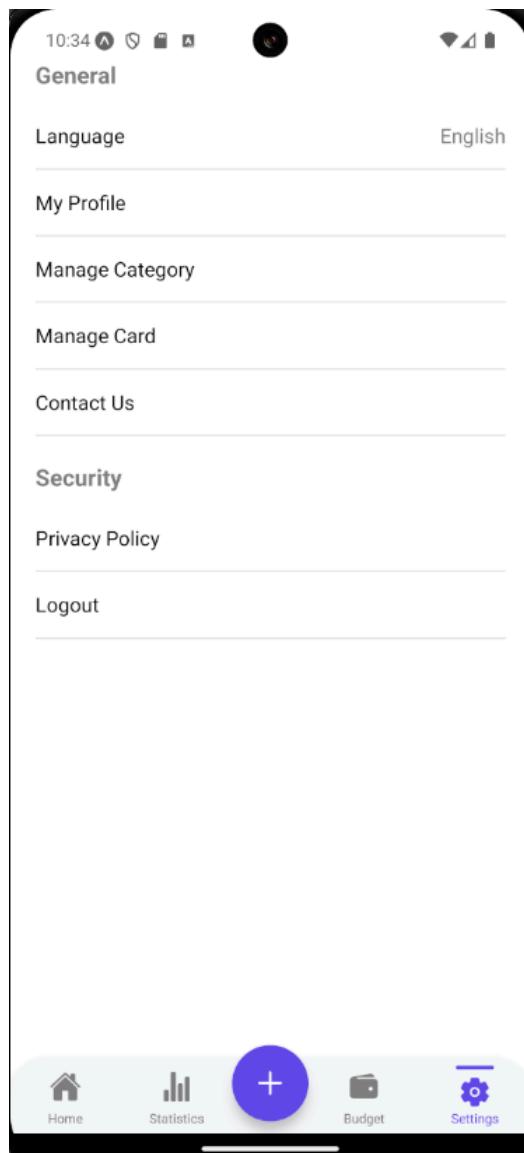
4	Trường "Số tiền"	Cho phép người dùng chỉnh sửa số tiền của ngân sách / mục tiêu.
5	Trường "Ngày"	Hiển thị ngày kết thúc ngân sách / mục tiêu. Người dùng có thể chỉnh sửa ngày này.
6	Mục "Select Category"	Cho phép người dùng chọn danh mục phù hợp cho ngân sách từ danh sách các biểu tượng.
7	Nút "Update"	Cập nhật ngân sách / mục tiêu

Bảng 2.15. Mô tả ngân sách và mục tiêu.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.16. Trang cài đặt

Người dùng có thể tiến hành cài đặt bằng cách bấm vào Settings bên dưới.



Hình 6.25. Trang cài đặt.

Thứ tự	Tên	Chức năng
1	Language	Hiển thị ngôn ngữ hiện tại của ứng dụng và cho phép

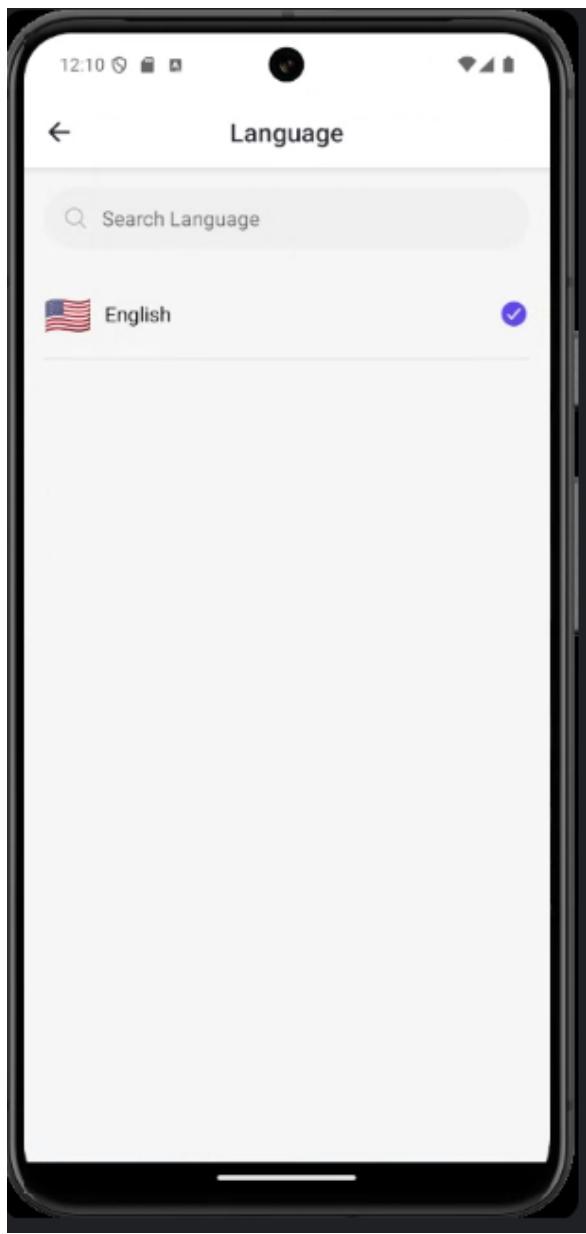
Chương 3: Phân tích thiết kế hệ thống

		người dùng thay đổi ngôn ngữ.
2	My Profile	Điều hướng người dùng đến trang thông tin cá nhân (Profile).
3	Manage Category	Điều hướng người dùng đến trang quản lý danh mục.
4	Manage Card	Điều hướng người dùng đến trang quản lý thẻ.
5	Contact Us	Cung cấp thông tin liên hệ hoặc biểu mẫu hỗ trợ liên lạc với đội ngũ quản trị.
6	Privacy Policy	Điều hướng đến trang chính sách bảo mật của ứng dụng.
7	Logout	Cho phép người dùng đăng xuất khỏi tài khoản.

Bảng 2.16. Mô tả ngân sách và mục tiêu.

3.3.1.17. Cài đặt ngôn ngữ

Chương 3: Phân tích thiết kế hệ thống



Hình 6.26. Cài đặt ngôn ngữ.

Vị trí	Tên	Chức năng
1	Thanh “Search Bar”	Tìm ngôn ngữ .
2	Danh sách ngôn ngữ	Đổi ngôn ngữ ứng dụng.

Bảng 2.17. Mô tả trang cài đặt ngôn ngữ.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.18. Chính sách và bảo mật.



Hình 6.27. Chính sách và bảo mật.

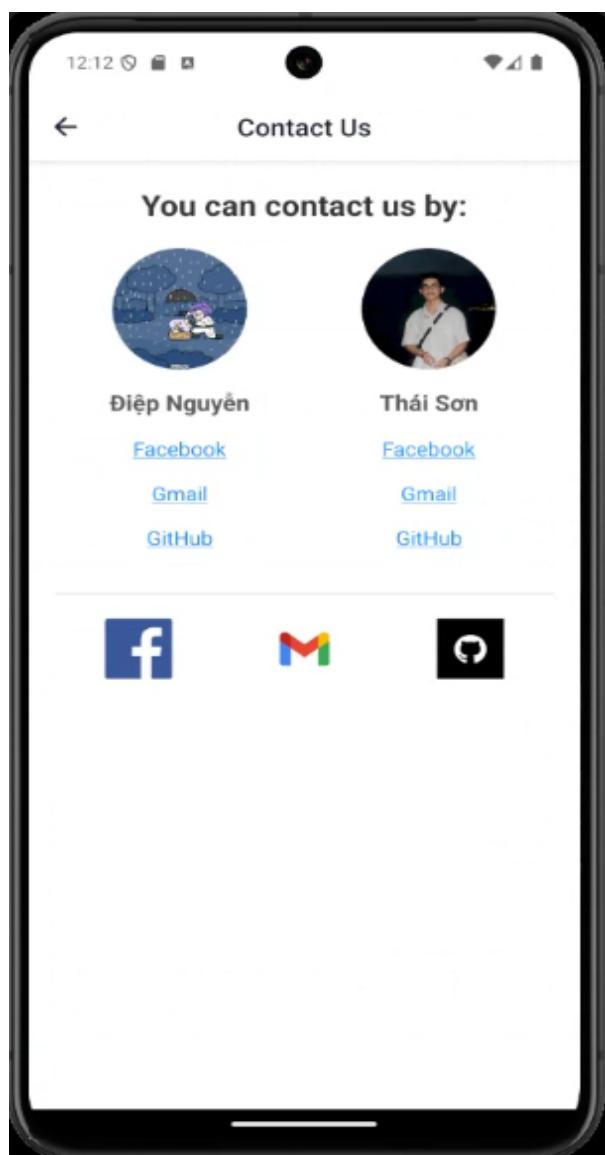
Vị trí	Tên	Chức năng
1	Nội dung chính sách	Hiển thị nội dung chính

Chương 3: Phân tích thiết kế hệ thống

		sách.
2	Nút “Approve”	Người dùng đồng ý với chính sách và thoát về trang trước.

Bảng 2.18. Mô tả trang chính sách và bảo mật.

3.3.1.19. Thông tin liên hệ.



Chương 3: Phân tích thiết kế hệ thống

Hình 6.28. Thông tin liên hệ.

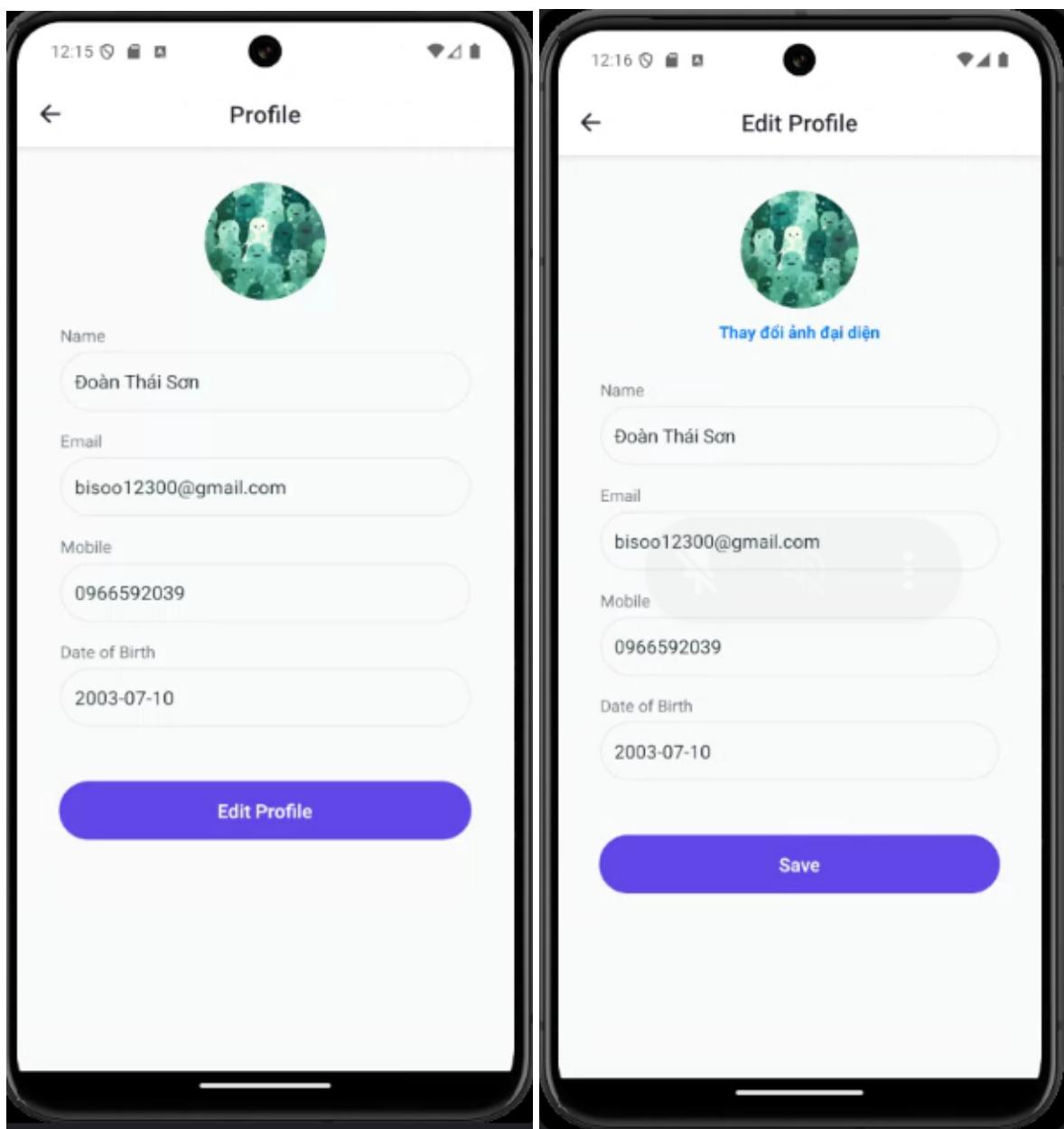
Vị trí	Tên	Chức năng
1	Thông tin nhà phát triển	Hiển thị thông tin nhà phát triển.
2	Nút “Back”	Quay về trang trước.

Bảng 2.19. Mô tả trang thông tin liên hệ.

3.3.1.20. Thông tin cá nhân.

Để cập truy cập thông tin cá nhân, ta bấm vào Setting trên thanh điều hướng, sau đó chọn My Profile , để cập nhật Profile bấm vào Edit Profile.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.29. Thông tin cá nhân & 6.30. Chỉnh sửa thông tin cá nhân.

Vị trí	Tên	Chức năng
1	Ảnh đại diện (Avatar)	Hiển thị ảnh đại diện của người dùng.
2	Các trường thông tin	Hiển thị thông tin người dùng.

Chương 3: Phân tích thiết kế hệ thống

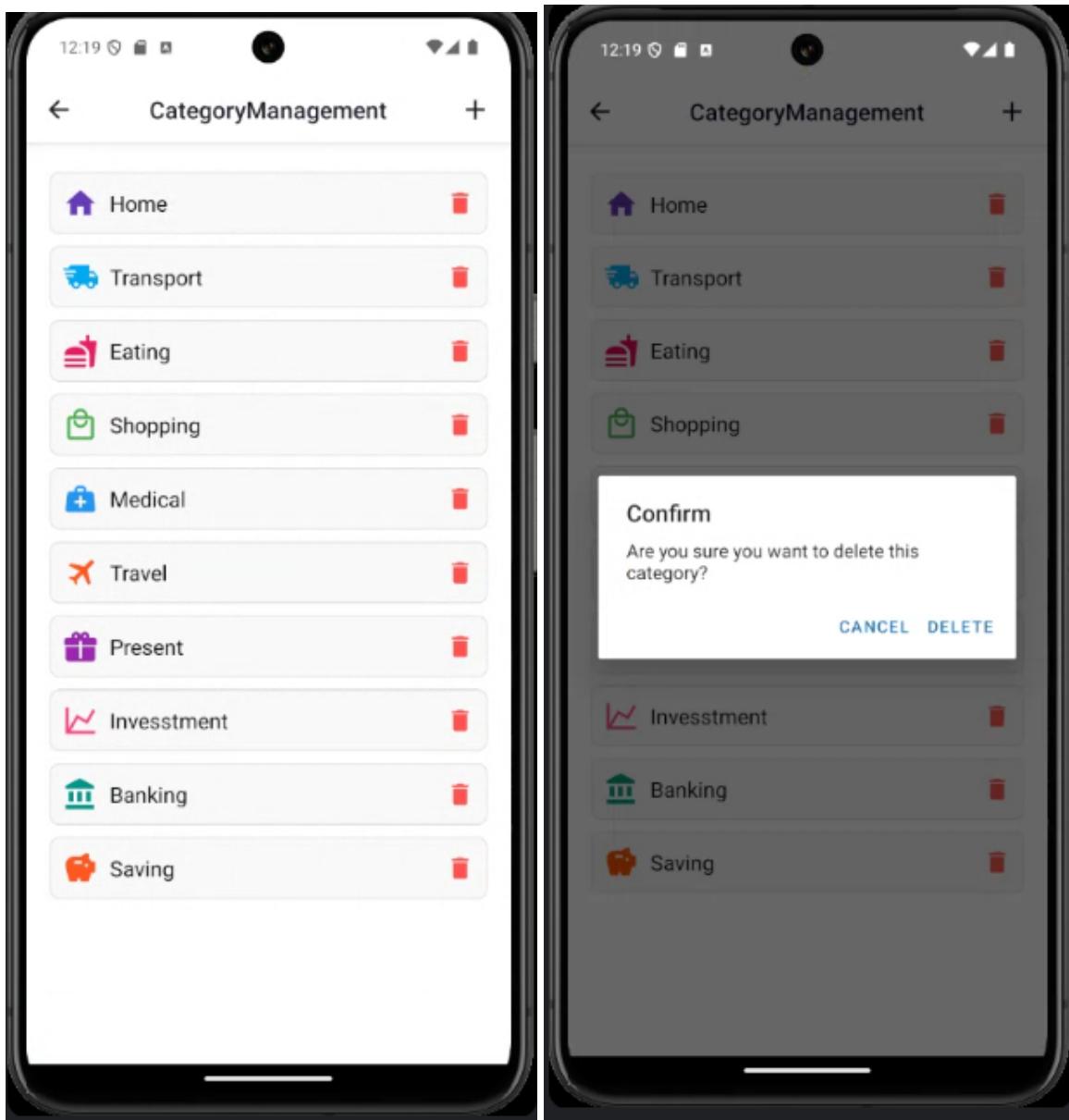
3	Nút "Edit Profile" / hoặc nút "Save"	Chuyển hướng đến trang Edit Profile / hoặc lưu thay đổi.
4	Nút "Back"	Quay về trang trước đó.

Bảng 2.20. Mô tả thông tin cá nhân và chỉnh sửa thông tin cá nhân.

3.3.1.21. Quản lý danh mục.

Để cập truy cập quản lý danh mục, ta bấm vào Setting trên thanh điều hướng, sau đó chọn Manage Categories.

Chương 3: Phân tích thiết kế hệ thống



Hình 6.30 & 6.31. Quản lý danh mục.

Vị trí	Tên	Chức năng
1	Tiêu đề "Category Management"	Hiển thị tiêu đề của trang, xác định đây là chức năng

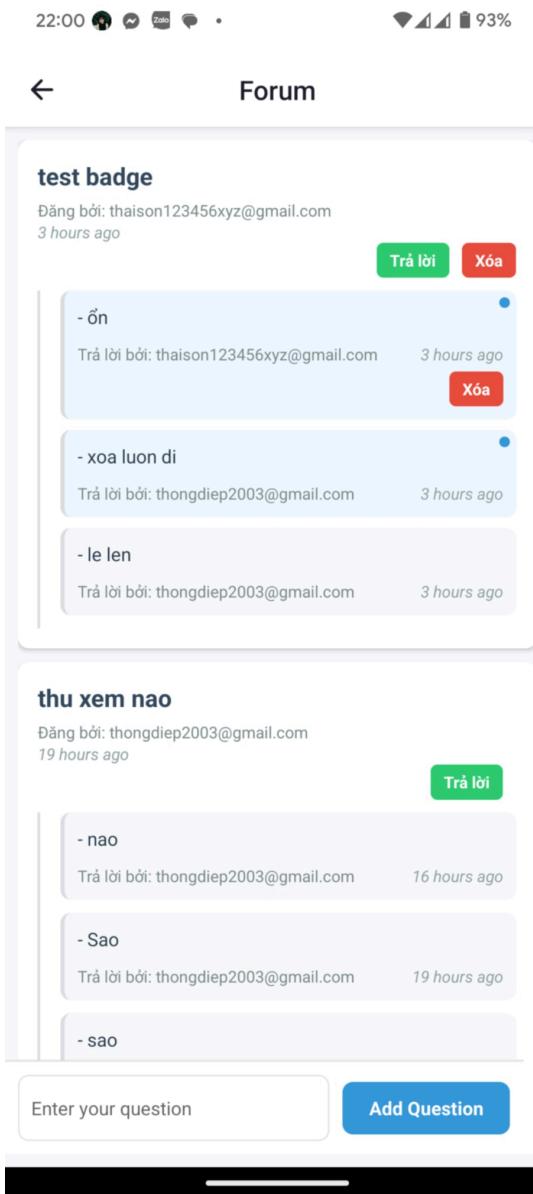
Chương 3: Phân tích thiết kế hệ thống

		quản lý danh mục.
2	Nút "Thêm mới" (+)	Nút bấm ở góc trên cùng bên phải, cho phép người dùng thêm danh mục mới.
3	Danh sách danh mục	Hiển thị danh sách các danh mục hiện có với biểu tượng đại diện và tên.
4	Nút "Xóa" (thùng rác)	Nút nằm bên phải mỗi danh mục, cho phép người dùng xóa danh mục đó.
5	Popup xác nhận	Khi người dùng nhấn nút "Xóa", hiển thị popup xác nhận việc xóa danh mục.
6	Nút "CANCEL"	Nút trong popup, cho phép người dùng hủy hành động xóa danh mục.
7	Nút "DELETE"	Nút trong popup, xác nhận và thực hiện việc xóa danh mục.

Bảng 2.21. Mô tả trang quản lý danh mục.

Chương 3: Phân tích thiết kế hệ thống

3.3.1.22. Diễn đàn.



Hình 6.32. Diễn đàn.

Vị trí	Tên	Chức năng
1	Tiêu đề "Forum"	Hiển thị tiêu đề của trang để xác định đây là chức năng diễn đàn.

Chương 3: Phân tích thiết kế hệ thống

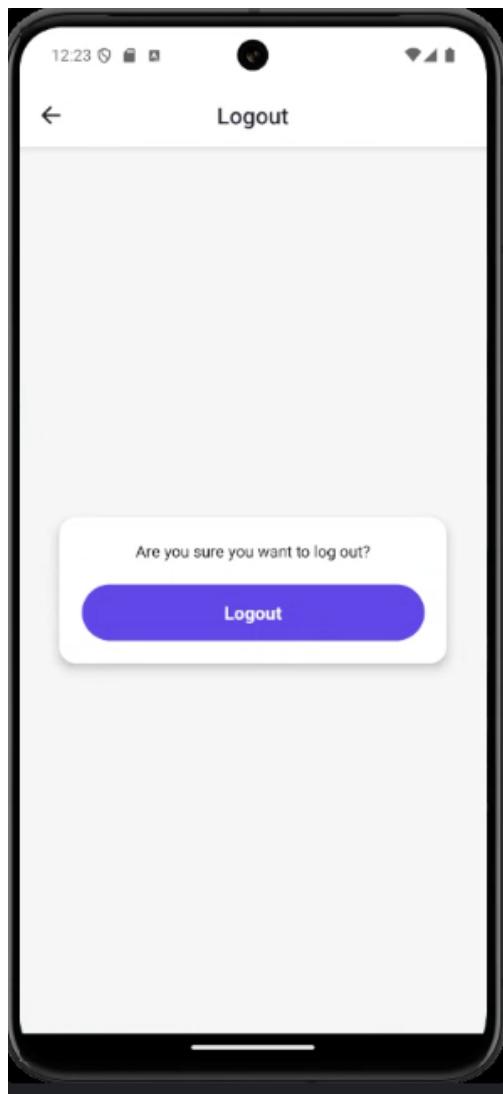
2	Nút quay lại (Back)	Nút nằm góc trái trên cùng, cho phép người dùng quay lại trang trước đó.
3	Câu hỏi (Question)	Hiển thị nội dung câu hỏi của người dùng trong diễn đàn.
4	Tác giả câu hỏi	Hiển thị tên hoặc email của người đăng câu hỏi.
5	Thời gian đăng bài	Hiển thị thời gian bài viết được đăng (ví dụ: "3 hours ago").
6	Nút "Trả lời" (Reply)	Cho phép người dùng thêm phản hồi vào câu hỏi.
7	Nút "Xóa" (Delete)	Cho phép người dùng hoặc admin xóa câu hỏi hoặc câu trả lời.
8	Phản hồi (Reply)	Hiển thị danh sách các phản hồi bên dưới mỗi câu hỏi.
9	Tác giả phản hồi	Hiển thị tên hoặc email của người đã trả lời câu hỏi.
10	Nút nhập câu hỏi	Trường nhập liệu ở dưới cùng, cho phép người dùng đặt câu hỏi mới trong diễn đàn.
11	Nút "Add Question"	Nút bấm để thêm câu hỏi mới vào danh sách câu hỏi trong diễn đàn.

Bảng 2.22. Mô tả trang diễn đàn.

3.3.1.23. Đăng xuất.

Chương 3: Phân tích thiết kế hệ thống

Cuối cùng, người dùng có thể đăng xuất bằng cách bấm vào Log Out ở trang Setting



Hình 6.33. Đăng xuất.

Vị trí	Tên	Chức năng
1	Tiêu đề "Logout"	Hiển thị tiêu đề của trang để xác định đây là chức năng đăng xuất.
2	Câu hỏi xác nhận	Hiển thị thông báo: "Are

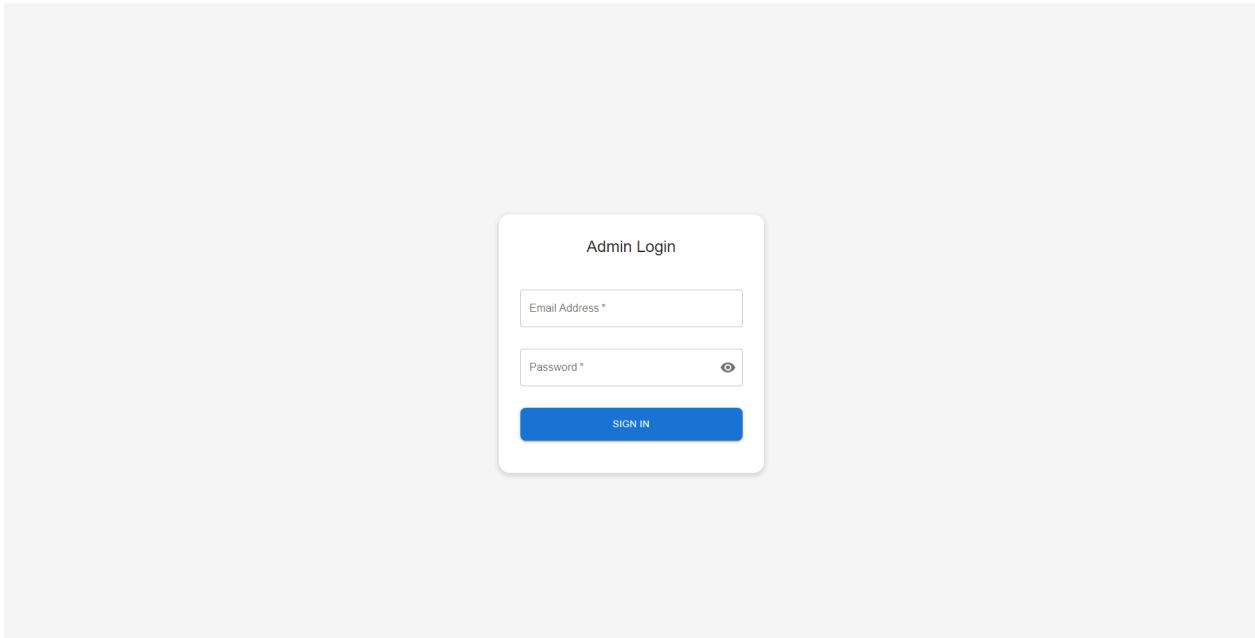
Chương 3: Phân tích thiết kế hệ thống

		you sure you want to log out?" để xác nhận ý định đăng xuất của người dùng.
3	Nút "Logout"	Nút bấm chính màu tím, thực hiện hành động đăng xuất khỏi tài khoản khi người dùng xác nhận.
4	Nút quay lại (Back)	Nút quay lại ở góc trái trên cùng, cho phép người dùng trở về trang trước đó nếu không muốn đăng xuất.

Bảng 2.23. Mô tả ngân sách và mục tiêu.

3.3.2. Giao diện quản trị viên:

3.3.2.1. Đăng nhập tài khoản



Hình 6.34. Đăng nhập.

Vị trí	Tên	Chức năng
--------	-----	-----------

Chương 3: Phân tích thiết kế hệ thống

1	Tiêu đề "Admin Login"	Hiển thị tiêu đề trang để phân biệt trang đăng nhập dành cho quản trị viên.
2	Trường "Email Address"	Nơi người dùng nhập địa chỉ email để đăng nhập. Đây là một trường bắt buộc.
3	Trường "Password"	Nơi người dùng nhập mật khẩu. Trường có biểu tượng "mắt" cho phép bật/tắt hiển thị mật khẩu.
4	Biểu tượng "Eye"	Cho phép người dùng bật/tắt hiển thị mật khẩu trong trường "Password"
5	Nút "SIGN IN"	Nút bấm để thực hiện quá trình xác thực thông tin và đăng nhập vào hệ thống admin.
6	Bố cục hộp đăng nhập	Hộp đăng nhập được căn giữa màn hình với thiết kế đơn giản, nổi bật, giúp tập trung vào nội dung chính.

Bảng 2.24. Mô tả đăng nhập.

3.3.2.2. Quản lý người dùng

Chương 3: Phân tích thiết kế hệ thống

The screenshot shows the Admin Dashboard interface. On the left is a sidebar with 'Users', 'Forum', 'Statistics', and 'Logout' options. The main area has a search bar 'Search users'. Below it is a table with columns: Email, Name, Mobile, Status, Block Reason, and Actions. The table contains the following data:

Email	Name	Mobile	Status	Action
thongdiep2003@gmail.com	Nguyen Hong Thong Diep	0392963132	active	
21110289@student.hcmute.edu.vn	N/A	N/A	active	
thaison123456xyz@gmail.com	Đoàn Thái Sơn	0966592038	active	
bisoo12300@gmail.com	Đoàn Thái Sơn	0966592038	blocked	
21110289@student.hcmute.edu.vn	N/A	N/A	active	

Hình 6.35. Quản lý người dùng.

Vị trí	Tên	Chức năng
1	Sidebar (Thanh bên)	Điều hướng giữa các mục: Users, Forum, Statistics, và Logout.
2	Search Bar	Nơi nhập từ khóa để tìm kiếm người dùng dựa trên email hoặc tên.
3	Bảng danh sách người dùng	Hiển thị thông tin chi tiết về người dùng.
4	Cột Email	Hiển thị địa chỉ email của người dùng.
5	Cột Name	Hiển thị tên người dùng (nếu có). Nếu không có, hiển thị "N/A".
6	Cột Mobile	Hiển thị số điện thoại của người dùng (nếu có). Nếu không có, hiển thị "N/A".

Chương 3: Phân tích thiết kế hệ thống

7	Cột Status	Hiển thị trạng thái của người dùng: active (màu xanh) hoặc blocked (màu đỏ).
8	Cột Block Reason	Hiển thị lý do người dùng bị khóa (nếu có).
9	Cột Actions	Hiển thị các thao tác quản lý: - Biểu tượng khóa (red): Khóa người dùng (chuyển trạng thái sang blocked). - Biểu tượng mở khóa (green): Mở khóa người dùng (chuyển trạng thái sang active).
10	Nút Logout	Đăng xuất khỏi trang quản lý.

Bảng 2.25. Mô tả quản lý người dùng.

3.3.2.3. Quản lý forum

Chương 3: Phân tích thiết kế hệ thống

The screenshot shows the Admin Dashboard with a blue header bar containing the title "Admin Dashboard". Below the header, there is a sidebar with three items: "Users", "Forum", and "Statistics". The main content area is titled "Forum Management" and contains a search bar labeled "Search forums". Below the search bar is a table with four columns: "Question", "Posted By", "Posted At", and "Replies". The table lists four forum posts:

Question	Posted By	Posted At	Replies	Actions
test	thaision123456xyz@gmail.com	12/7/2024, 1:37:56 AM	5	
tết2	thaision123456xyz@gmail.com	12/7/2024, 2:22:08 AM	3	
thu xem nao	thongdiep2003@gmail.com	12/7/2024, 2:31:04 AM	4	
test badge	thaision123456xyz@gmail.com	12/7/2024, 6:43:32 PM	3	

At the bottom left of the dashboard, there is a "Logout" button.

Hình 6.36. Quản lý forum.

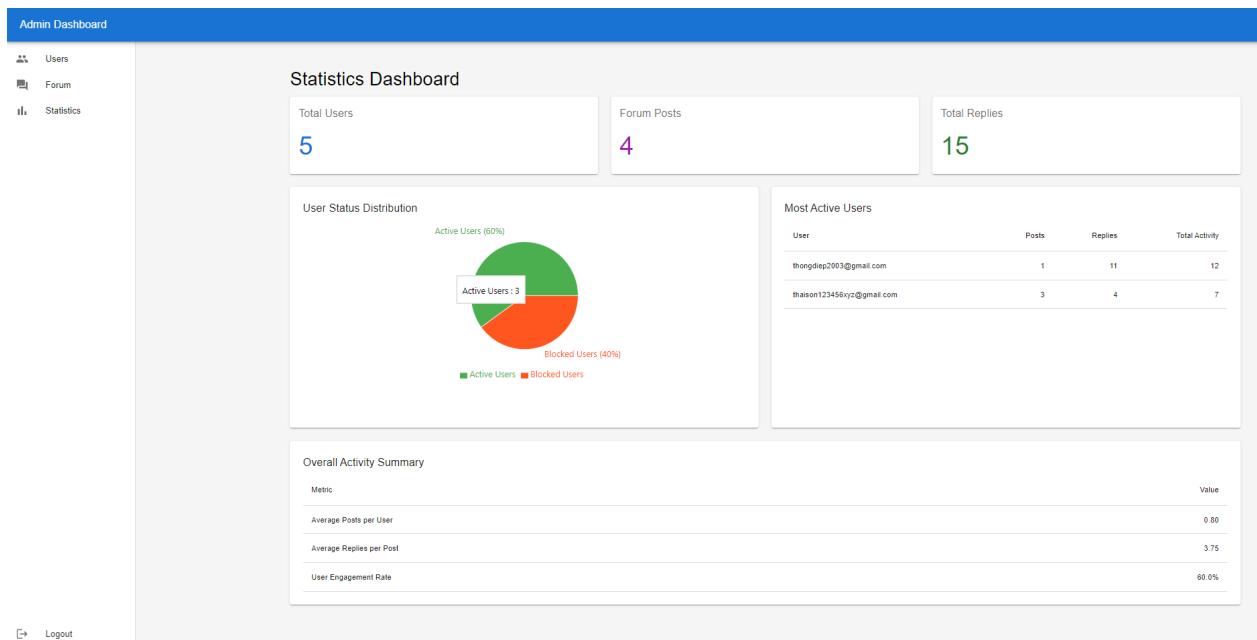
Vị trí	Tên	Chức năng
1	Sidebar (Thanh bên)	Điều hướng giữa các mục: Users, Forum, Statistics, và Logout.
2	Search Bar	Nhập từ khóa để tìm kiếm các bài viết hoặc câu hỏi trong diễn đàn dựa trên nội dung hoặc email.
3	Bảng danh sách bài viết	Hiển thị danh sách các bài viết, câu hỏi trong diễn đàn.
4	Cột "Question"	Hiển thị địa chỉ email của người đăng bài viết.
5	Cột "Posted By"	Hiển thị địa chỉ email của người đăng bài viết.
6	Cột "Posted At"	Hiển thị thời gian bài viết được đăng.
7	Cột "Replies"	Hiển thị số lượng trả lời

Chương 3: Phân tích thiết kế hệ thống

		cho từng bài viết hoặc câu hỏi.
8	Cột "Actions"	Hiển thị các nút chức năng để quản lý bài viết.
9	Nút "View" (mắt)	Xem chi tiết bài viết, bao gồm nội dung và các câu trả lời liên quan.
10	Nút "Delete" (thùng rác)	Xóa bài viết hoặc câu hỏi khỏi diễn đàn.

Bảng 2.26. Mô tả Quản lý forum.

3.3.2.4. Thống kê quản lý



Hình 6.37. Thống kê quản lý.

Vị trí	Tên	Chức năng
1	Tổng số người dùng (Total Users)	Hiển thị tổng số người dùng đã đăng ký trên hệ

Chương 3: Phân tích thiết kế hệ thống

		thống.
2	Tổng số bài viết (Forum Posts)	Hiển thị tổng số bài viết hoặc câu hỏi được đăng trong diễn đàn.
3	Tổng số phản hồi (Total Replies)	Hiển thị tổng số phản hồi cho các bài viết hoặc câu hỏi trong diễn đàn.
4	Biểu đồ phân bố trạng thái người dùng (User Status Distribution)	Biểu đồ tròn hiển thị tỷ lệ người dùng active và blocked.
5	Danh sách người dùng hoạt động nhiều nhất (Most Active Users)	Hiển thị danh sách người dùng với các cột: User (Email), Posts, Replies, và Total Activity.
6	Tóm tắt hoạt động chung (Overall Activity Summary)	Hiển thị các chỉ số hoạt động: Average Posts per User, Average Replies per Post, và User Engagement Rate.

Bảng 2.27. Thống kê quản lý.

CHƯƠNG 4 CÀI ĐẶT VÀ KIỂM THỬ

4.1. Cài đặt ứng dụng

4.1.1. Cài đặt công cụ

- Để có thể chạy được ứng dụng, máy tính phải đảm bảo các yêu cầu sau:
 - Cài đặt Java (phiên bản khuyến nghị là 17.0.5).

4.1.2. Hướng dẫn cài đặt

- Cài Đặt Môi Trường Phát Triển.
- Tạo Dự Án React Native.
- Cài Đặt Các Gói Thư Viện.
- Cấu Hình Firebase.
- Xây Dựng Giao Diện.
- Quản Lý Dữ Liệu với Firebase.
- Gửi OTP qua Mailjet.
- Triển khai Ứng Dụng.

4.2. Kiểm thử chương trình

4.2.1. Kế hoạch kiểm thử

Dự án sử dụng kiểm thử tự động và kiểm thử thủ công để kiểm tra các chức năng của hệ thống, bao gồm:

- Unit test: kiểm thử phần mềm trong đó thực hiện kiểm thử từng đơn vị hoặc thành phần riêng lẻ của phần mềm.
- Integration test: kết hợp các module của ứng dụng và kiểm thử như một ứng dụng hoàn chỉnh.

Chương 4: Cài đặt và kiểm thử

- System test: phương pháp theo dõi và đánh giá hành vi của sản phẩm hoặc hệ thống phần mềm hoàn chỉnh và đã được tích hợp đầy đủ, dựa vào đặc tả và các yêu cầu chức năng đã được xác định trước.
- Acceptance test: kiểm thử nhằm xác định hệ thống phần mềm có đạt yêu cầu kỹ thuật hay không.
- Usability test: xác minh ứng dụng có khả năng ứng dụng cao và dễ sử dụng, giao diện thân thiện với người dùng.

4.2.2. Mục tiêu kiểm thử

Đảm bảo các chức năng của ứng dụng (được xác định trong đặc tả dự án) hoạt động chính xác trong điều kiện thực tế.

4.2.3. Tiêu chí kiểm thử

Các tiêu chí kiểm thử bao gồm:

- Tiêu chí định chỉ: nếu có ít nhất 40% số test case bị lỗi, tạm dừng việc kiểm thử đến khi tất cả các lỗi hiện tại được khắc phục.
- Tiêu chí kết thúc: chỉ định các tiêu chí biểu thị việc hoàn thành thành công giai đoạn kiểm thử.
 - Tỷ lệ chạy test case bắt buộc phải là 100% trừ khi có lý do rõ ràng.
 - Tỷ lệ vượt qua các test case là 80%, việc đạt tỉ lệ này là bắt buộc.

4.2.4. Công cụ kiểm thử

Selenium là một công cụ kiểm thử tự động phổ biến, hỗ trợ kiểm thử các ứng dụng web. Selenium cung cấp các tính năng như:

- WebDriver: Điều khiển trình duyệt từ xa, hỗ trợ nhiều trình duyệt (Chrome, Firefox, Safari, Edge, v.v.).

Chương 4: Cài đặt và kiểm thử

- Selenium Grid: Kiểm thử trên nhiều môi trường cùng lúc để tăng tốc quá trình kiểm thử.
- Selenium IDE: Công cụ ghi và phát lại kiểm thử đơn giản.
- Selenium Remote Control (RC): Cho phép thực hiện kiểm thử tự động trên các ứng dụng web từ xa.

Lý do chọn Selenium:

- Hỗ trợ đa nền tảng (Windows, macOS, Linux).
- Dễ dàng tích hợp với các công cụ khác như TestNG, JUnit, Maven.
- Tính linh hoạt cao trong việc kiểm thử ứng dụng web trên nhiều trình duyệt khác nhau.

4.2.5. Quy trình kiểm thử

Quy trình kiểm thử ứng dụng được thực hiện theo các bước sau:

1. Xác định Mục Tiêu Kiểm Thử:

- Xác định các tính năng và yêu cầu của hệ thống cần kiểm thử, bao gồm các chức năng của ứng dụng tài chính (đăng nhập, giao dịch, quản lý tài khoản, v.v.).

2. Lập Kế Hoạch Kiểm Thử:

- Xây dựng kế hoạch kiểm thử chi tiết, xác định phạm vi kiểm thử, thời gian, và tài nguyên cần thiết.

3. Chuẩn Bị Môi Trường Kiểm Thử:

- Cài đặt các công cụ kiểm thử, cấu hình Selenium WebDriver với trình duyệt tương ứng.

Chương 4: Cài đặt và kiểm thử

- Cài đặt môi trường kiểm thử (ví dụ: hệ điều hành, phần mềm cần thiết, các công cụ kiểm thử).

4. Viết Test Case:

- Soạn các test case cho từng tính năng của ứng dụng, ví dụ:
 - Kiểm tra đăng nhập.
 - Kiểm tra chức năng tạo và xóa giao dịch.
 - Kiểm tra tính toán ngân sách.
- Đảm bảo test case bao gồm các kịch bản hợp lệ và không hợp lệ.

5. Thực Hiện Kiểm Thử:

- Thực hiện các test case đã soạn trên Selenium. WebDriver sẽ tự động mở trình duyệt, thực hiện các hành động và kiểm tra kết quả.
- Ghi lại kết quả kiểm thử và báo cáo lỗi nếu có.

6. Phân Tích Kết Quả Kiểm Thử:

- Xem xét các kết quả kiểm thử, xác định các lỗi hoặc vấn đề phát sinh trong quá trình kiểm thử.
- Đảm bảo rằng tất cả các tính năng hoạt động như mong đợi.

7. Sửa Lỗi và Kiểm Thử Lại:

- Các lỗi phát hiện sẽ được sửa chữa bởi đội ngũ phát triển.
- Kiểm thử lại các tính năng đã sửa và chạy lại các test case để đảm bảo tính ổn định của ứng dụng.

8. Báo Cáo Kiểm Thử:

Chương 4: Cài đặt và kiểm thử

- Tổng hợp và báo cáo kết quả kiểm thử, bao gồm các lỗi, cảnh báo, và đề xuất cải tiến.
- Cung cấp báo cáo kiểm thử cho các bên liên quan.

9. Hoàn Thành Kiểm Thử:

- Sau khi hoàn tất việc kiểm thử và sửa lỗi, kiểm tra lại toàn bộ hệ thống để đảm bảo mọi tính năng đều hoạt động ổn định và chính xác.

4.2.6. Kết quả kiểm thử

- Đối với người dùng.

STT	Test Case	Kết quả mong đợi	Kết quả
1	Thực hiện chức năng đăng ký	Thông báo Đăng ký thành công.	Pass
2	Thực hiện chức năng đăng nhập	Đăng nhập thành công, đưa người dùng đến trang chủ	Pass
3	Thực hiện chức năng quên mật khẩu	Đường dẫn đặt lại mật khẩu được gửi đến email người dùng	Pass
4	Thực hiện chức năng đăng xuất	Đăng xuất thành công, xóa token của người dùng	Pass
5	Thực hiện chức năng thêm giao dịch	Thông báo thành công và hiển thị giao dịch, cập nhật thống kê	Pass
6	Thực hiện chức	Thông báo thay đổi	Pass

Chương 4: Cài đặt và kiểm thử

	năng thay đổi thông tin người dùng	thành công và cập nhật	
7	Thực hiện thêm ngân sách mới	Thông báo thành công và hiển thị ngân sách	Pass
8	Thực hiện chức năng thêm mục tiêu tài chính	Thông báo thành công và hiển thị mục tiêu	Pass
9	Thực hiện thêm danh mục mới	Thông báo thêm danh mục thành công và hiển thị danh mục	Pass
10	Thực hiện xóa giao dịch	Thông báo thành công và cập nhật danh sách giao dịch, ngân sách, thống kê.	Pass
11	Thực hiện đặt câu hỏi trong diễn đàn	Thông báo thêm thành công, Hiển thị câu hỏi	Pass
12	Thực hiện đặt câu trả lời vào câu hỏi của người khác	Thông báo thêm thành công, Hiển thị câu trả lời, ứng dụng gửi thông báo cho người đó	Pass
13	Thực hiện xóa câu hỏi	Thông báo xóa thành công, cập nhật hiển thị	Pass
14	Thực hiện xóa câu trả lời	Thông báo xóa thành công, cập nhật hiển thị	Pass

- Đối với quản trị viên

Chương 4: Cài đặt và kiểm thử

STT	Test Case	Kết quả mong đợi	Kết quả
1	Chức năng đăng nhập	Đăng nhập thành công	Pass
2	Hiển thị dữ liệu	Hiển thị danh sách người dùng, hiển thị danh sách câu hỏi, câu trả lời	Pass
3	Chặn người dùng	Người bị chặn không thể đăng nhập được	Pass
4	Bỏ chặn người dùng	Người dùng bị chặn đăng nhập lại vào app được	Pass
5	Xoá câu hỏi	Câu hỏi và các câu trả lời liên quan bị xoá khỏi cơ sở dữ liệu	Pass
6	Xoá câu trả lời	Câu trả lời bị xoá khỏi cơ sở dữ liệu	Pass
7	Kiểm tra giao diện	Không có lỗi hiển thị	Pass

PHẦN KẾT LUẬN

1. Kết quả đạt được

Sau một thời gian nghiên cứu và hoàn thiện đề tài, nhóm thực hiện đã đạt được những thành quả sau:

- ❖ Về lý thuyết và kỹ năng:
 - ✓ Nhóm đã nắm vững các kiến thức về phân tích và thiết kế hệ thống ứng dụng, mô tả quy trình và dòng sự kiện của các chức năng trong ứng dụng.
 - ✓ Các thành viên trong nhóm đã rèn luyện kỹ năng mô hình hóa hệ thống qua các **Use Case Diagrams, Sequence Diagrams**, giúp hiểu rõ hơn về mối quan hệ giữa các thành phần và luồng xử lý trong hệ thống.
 - ✓ Ngoài ra, nhóm cũng đã học hỏi và áp dụng những nguyên lý cơ bản của thiết kế ứng dụng di động, bao gồm khả năng tương tác với người dùng (UI/UX), bảo mật và xác thực thông tin người dùng.
- ❖ Về ứng dụng: Xây dựng được ứng dụng với các chức năng sau:
 - ✓ **Đăng ký tài khoản:** Cho phép người dùng tạo mới tài khoản và lưu trữ thông tin cá nhân vào cơ sở dữ liệu.
 - ✓ **Đăng nhập và đăng xuất:** Hệ thống cho phép người dùng đăng nhập vào ứng dụng và thực hiện đăng xuất sau khi sử dụng.

- ✓ **Quản lý thông tin người dùng:** Người dùng có thể xem chi tiết thông tin và chỉnh sửa thông tin của mình
- ✓ **Quản lý mục tiêu tiết kiệm:** Người dùng có thể thêm tiền vào các mục tiêu tiết kiệm của mình và theo dõi tiến độ đạt mục tiêu.
- ✓ **Xóa câu hỏi và câu trả lời trên diễn đàn:** Người dùng có thể tạo câu hỏi, trả lời và xóa những câu hỏi hoặc câu trả lời của chính mình trên diễn đàn.
- ✓ **Chức năng Admin quản lý người dùng:** Admin có thể xem thống kê và quản lý tài khoản người dùng.
- ✓ **Chức năng Admin quản lý bài viết trên diễn đàn:** Admin có thể quản lý các bài viết của diễn đàn với tính năng xóa bài viết không phù hợp.

2. Ưu điểm và nhược điểm

Ứng dụng có những ưu điểm như sau:

- ✓ **Tính linh hoạt và dễ sử dụng:** Giao diện của ứng dụng được thiết kế đơn giản, dễ hiểu và dễ sử dụng, giúp người dùng dễ dàng tiếp cận và thực hiện các chức năng như đăng ký, đăng nhập, và quản lý thông tin cá nhân.
- ✓ **Tính bảo mật cao:** Các chức năng như đăng nhập và đăng xuất sử dụng cơ chế xác thực mạnh mẽ, bảo vệ thông tin người dùng khỏi các truy cập trái phép.
- ✓ **Quản lý thông tin hiệu quả:** Các chức năng quản lý thông tin người dùng và mục tiêu tiết kiệm giúp người dùng dễ dàng theo dõi và điều chỉnh thông tin cá nhân cũng như tiến độ thực hiện mục tiêu của mình.

✓ **Tính năng tương tác diễn đàn:** Các chức năng hỏi đáp và trả lời trên diễn đàn thúc đẩy sự tương tác giữa người dùng, tạo không gian trao đổi kiến thức và thông tin.

Tuy nhiên, do giới hạn về thời gian và nhân lực, ứng dụng cũng có một số nhược điểm sau:

✓ **Chưa tối ưu hóa hoàn toàn về hiệu suất:** Mặc dù ứng dụng đã hoạt động ổn định nhưng hiệu suất có thể được cải thiện hơn nữa, đặc biệt là khi ứng dụng có nhiều người dùng cùng lúc.

3. Hướng phát triển

Trong thời gian tới, để nâng cao chất lượng và tính năng của ứng dụng, nhóm thực hiện dự án đề xuất một số hướng phát triển như sau:

❖ Cải thiện hiệu suất và khả năng mở rộng

Tối ưu hóa hiệu suất: Để đảm bảo ứng dụng có thể phục vụ số lượng người dùng lớn mà không gặp phải tình trạng lag hoặc giật, cần tối ưu hóa mã nguồn và các quy trình xử lý trong hệ thống, đặc biệt là đối với các tính năng liên quan đến việc quản lý dữ liệu người dùng và mục tiêu tiết kiệm.

Tăng khả năng mở rộng: Hệ thống có thể sử dụng các công nghệ và kiến trúc hỗ trợ mở rộng như microservices, công nghệ lưu trữ đám mây và công nghệ phân tán để có thể xử lý tốt khi số lượng người dùng tăng. Việc chia tách các chức năng của hệ thống thành các module độc lập sẽ giúp dễ dàng mở rộng và bảo trì trong tương lai.

❖ Tăng cường bảo mật

Xác thực đa yếu tố (MFA): Để bảo vệ tài khoản người dùng khỏi các cuộc tấn công, hệ thống có thể bổ sung tính năng xác thực hai yếu tố (2FA) hoặc xác

thực ba yếu tố (3FA) vào quy trình đăng nhập. Điều này sẽ giúp đảm bảo rằng chỉ có người dùng hợp pháp mới có thể truy cập vào tài khoản của mình.

Mã hóa dữ liệu: Áp dụng các phương pháp mã hóa hiện đại như AES hoặc RSA để bảo vệ dữ liệu nhạy cảm của người dùng, đặc biệt là khi lưu trữ mật khẩu, thông tin cá nhân và các giao dịch tài chính.

❖ Mở rộng tính năng ứng dụng

Hỗ trợ nhiều phương thức thanh toán: Để người dùng có thể dễ dàng thêm tiền vào các mục tiêu tiết kiệm, ứng dụng có thể tích hợp thêm các phương thức thanh toán trực tuyến như thẻ tín dụng, chuyển khoản ngân hàng, hoặc các dịch vụ ví điện tử như Momo, ZaloPay.

❖ Nâng cao giao diện người dùng (UI/UX)

Cải tiến giao diện và trải nghiệm người dùng: Cần cải thiện giao diện người dùng sao cho trực quan hơn, dễ sử dụng và có thể tương thích với nhiều thiết bị di động khác nhau. Các tính năng như màu sắc phù hợp, biểu tượng rõ ràng và thiết kế tối giản sẽ giúp người dùng dễ dàng thao tác và tìm kiếm thông tin.

Tạo trải nghiệm người dùng cá nhân hóa: Dựa trên dữ liệu người dùng, hệ thống có thể cá nhân hóa các mục tiêu tiết kiệm hoặc câu hỏi trên diễn đàn, gợi ý các mục tiêu hoặc chủ đề liên quan dựa trên hành vi và sở thích của người dùng.

❖ Tích hợp các công nghệ mới

AI và Machine Learning: Các mô hình học máy có thể được áp dụng để phân tích hành vi người dùng và cung cấp các đề xuất tài chính thông minh, chẳng hạn như tư vấn mục tiêu tiết kiệm tối ưu dựa trên thu nhập và chi tiêu của người dùng.

Chatbot hỗ trợ người dùng: Một chatbot AI có thể được tích hợp vào ứng dụng để giúp người dùng giải đáp thắc mắc nhanh chóng, từ việc giải thích về các tính năng trong ứng dụng đến việc cung cấp hướng dẫn chi tiết về cách thêm tiền vào mục tiêu tiết kiệm.

❖ **Tăng cường chức năng cho Admin**

Thống kê và báo cáo nâng cao: Để hỗ trợ quản lý người dùng và dữ liệu, Admin có thể có quyền truy cập vào các bảng điều khiển thống kê nâng cao.

Quản lý diễn đàn hiệu quả hơn: Admin có thể có các công cụ để kiểm tra, duyệt và kiểm soát các bài đăng trên diễn đàn, giúp ngăn chặn spam và đảm bảo rằng diễn đàn luôn sạch sẽ và có tính chất xây dựng.

❖ **Hỗ trợ đa ngôn ngữ và quốc tế hóa**

Đa ngôn ngữ: Để mở rộng thị trường và phục vụ người dùng quốc tế, ứng dụng có thể cung cấp nhiều ngôn ngữ khác nhau. Điều này không chỉ giúp người dùng dễ dàng sử dụng mà còn tăng khả năng tiếp cận của ứng dụng ở các thị trường khác nhau.

Tính năng đa tiền tệ: Với việc mở rộng ứng dụng ra các quốc gia khác nhau, có thể tích hợp tính năng đa tiền tệ để người dùng có thể thực hiện giao dịch và theo dõi mục tiêu tiết kiệm bằng các loại tiền tệ khác nhau.

TÀI LIỆU THAM KHẢO

[1] Khái niệm React Native, truy cập 22/9/2024.

<https://careerviet.vn/vi/talentcommunity/wiki-career/react-native-la-gi-tong-quan-ve-lap-trinh-react-native-moi-nhat.35A52331.html>

[2] Ưu điểm và nhược điểm của React Native, React Native hoạt động như thế nào?, Những ứng dụng viết bằng React Native, truy cập 22/9/2024.

<https://fptshop.com.vn/tin-tuc/danh-gia/react-native-la-gi-nhung-uu-diem-va-han-che-cua-react-native-can-nam-truoc-khi-bat-dau-172012>

[3] Khái niệm Tailwind CSS, truy cập 22/9/2024.

<https://viblo.asia/p/tim-hieu-ve-tailwind-css-924lJp6WKPM>

[4] Ưu và nhược điểm của Tailwind CSS, truy cập 22/9/2024.

<https://aptech.fpt.edu.vn/tailwind-la-gi.html>

[5] Các tính năng nổi bật của Tailwind CSS, truy cập 22/9/2024.

<https://unitop.com.vn/tailwind-css.html/>

[6] Khái niệm Firebase SDK, truy cập 22/9/2024.

<https://osdb.io/firebase-pros-and-cons-when-you-should-and-shouldnt-use-firebase-osdb/>

[7] Ưu và nhược điểm của Firebase SDK, truy cập 22/9/2024.

<https://www.linkedin.com/pulse/advantages-disadvantages-firebase-nav-adalyn/>

[8] Khái niệm, ưu và nhược điểm và cách hoạt động của các chức năng trong Firebase, truy cập 22/9/2024.

<https://www.quora.com/What-are-the-pros-and-cons-of-using-Firebase-user-auth-instead-of-building-a-custom-in-regards-to-possible-future-backend-changes>

[9] Khái niệm Mailjet, truy cập 22/9/2024.

<https://gmass.co/blog/mailjet-review/>

[10] Ưu và nhược điểm của Mailjet, truy cập 22/9/2024.

<https://www.emailvendorselection.com/mailjet-review/>

[11] Cách thức hoạt động của Mailjet, truy cập 22/9/2024.

<https://www.trustradius.com/products/mailjet/reviews?qs=pros-and-cons>

[12] Khái niệm, ưu và nhược điểm của Expo, 22/9/2024.

<https://medium.com/@softworthsolutionspvtltd/expo-vs-react-native-cli-7e47c7630039>

[13] Giới thiệu, ưu điểm và nhược điểm của các ứng dụng quản lý tài chính phổ biến hiện nay, truy cập 23/9/2024.

<https://www.pcmag.com/picks/the-best-personal-finance-services>

BẢNG PHÂN CHIA CÔNG VIỆC

Họ và tên	Mã số sinh viên	Công việc
Nguyễn Hồng Thông Điện	21110166	<ul style="list-style-type: none"> - Lên kế hoạch thực hiện cho dự án. - Mô hình hóa yêu cầu bài toán. - Thực hiện các chức năng của người dùng: Login, Register, Forgot Password, Logout, OTP, HomePage , Transaction, Category, Setting, Forum. - Tiến hành kiểm thử. - Hoàn thành báo cáo.
Đoàn Thái Sơn	21110289	<ul style="list-style-type: none"> - Lựa chọn đề tài. - Đặc tả yêu cầu bài toán. - Thực hiện các chức năng người dùng như: Budget, Goal, Card, Statistic, Notify. - Thực hiện các chức năng của admin như: Login, Logout, Forum management, User management. - Hoàn thiện ứng dụng.

BÁO CÁO TIẾN ĐỘ

STT	Thời gian	Nội dung báo cáo	Ghi chú
1	25/09/2024	<ul style="list-style-type: none">- Báo cáo tiến độ thực hiện công việc.- Trao đổi Công nghệ và cơ sở dữ liệu sẽ sử dụng.- Trao đổi về phân quyền hoạt động của ứng dụng.- Mô tả các chức năng.- Trình bày thiết kế file Figma.	
2	28/11/2024	<ul style="list-style-type: none">- Báo cáo tiến độ thực hiện công việc.- Chạy chương trình và thực hiện các chức năng.	
3	12/12/2024	<ul style="list-style-type: none">- Trình bày bài báo cáo.	
4			
5			
6			
7			