

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



TIÊU LUẬN CHUYÊN NGÀNH

Đề tài:

XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH

GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Hữu Trung

NHÓM SINH VIÊN THỰC HIỆN:

Nguyễn Hồng Thông Điệp

21110166

Đoàn Thái Sơn

21110289

Tp. Hồ Chí Minh, tháng 12 năm 2024

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



TIÊU LUẬN CHUYÊN NGÀNH

Đề tài:

XÂY DỰNG ỨNG DỤNG QUẢN LÝ TÀI CHÍNH

GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Hữu Trung

NHÓM SINH VIÊN THỰC HIỆN:

Nguyễn Hồng Thông Điệp

21110166

Đoàn Thái Sơn

21110289

Tp. Hồ Chí Minh, tháng 12 năm 2024

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 ỨNG 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 ỦNG 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

Trường ĐH Sư Phạm Kỹ Thuật TP.HCM

Khoa: Công nghệ thông tin

ĐỀ CƯƠNG CHI TIẾT TIỂU LUẬN CHUYÊN NGÀNH

Họ và tên sinh viên thực hiện 1: **Nguyễn Hồng Thông Điện** 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

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....	1
PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN.....	2
LỜI CẢM ƠN.....	3
ĐỀ CƯƠNG CHI TIẾT TIỀU LUẬN CHUYÊN NGÀNH	4
MỤC LỤC	6
KẾ HOẠCH THỰC HIỆN	8
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	6
1.3. Firebase SDK	8
1.4. Mailjet	16
1.5. Expo	17
1.6. JWT	18

CHƯƠNG 2 KHẢO SÁT THỰC TRẠNG 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	133
3.3. Thiết kế giao diện	135
CHƯƠNG 4 CÀI ĐẶT VÀ KIỂM THỬ.....	165
4.1. Cài đặt ứng dụng	165
4.2. Kiểm thử chương trình	165
PHẦN KẾT LUẬN	172
1. Kết quả đạt được	172
2. Ưu điểm và nhược điểm.....	173
3. Hướng phát triển	174
TÀI LIỆU THAM KHẢO	177
BẢNG PHÂN CHIA CÔNG VIỆC	179
BÁO CÁO TIÉN ĐỘ	180

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 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 một framework mã nguồn mở được phát triển bởi Facebook vào năm 2015, cho phép các nhà phát triển xây dựng các ứng dụng di động cho cả iOS và Android bằng ngôn ngữ JavaScript. Thay vì viết mã bằng ngôn ngữ gốc của từng nền tảng (Swift cho iOS và Kotlin/Java cho Android), React Native cho phép lập trình viên viết ứng dụng một lần và sử dụng chung mã nguồn cho cả hai hệ điều hành. Nó dựa trên React – một thư viện JavaScript phổ biến cho việc xây dựng giao diện người dùng – và sử dụng các thành phần giao diện tương tự như React, nhưng được ánh xạ thành các thành phần gốc 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 cố gắng mô phỏng những thành phần gốc này như hoạt ảnh, các chi tiết nhỏ sẽ tồn tại rất nhiều công sức và chúng có thể nhanh chóng trở nên lỗi thời.

Chương 1: Cơ sở lý thuyết

- 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:**
 - 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.

Chương 1: Cơ sở lý thuyết

- Gỡ lỗi và quản lý mã nguồn: Khi có sự cố xảy ra, React Native tạo ra một tảng 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]

React Native hoạt động dựa trên 3 threads (chuỗi) sau:

- UI Thread (Main Thread): đây là chuỗi chính chịu trách nhiệm xử lý giao diện và truy cập giao diện người dùng để thực hiện các tác vụ liên quan.
- Shadow Thread: chuỗi này dùng React Native để tính toán bộ cục của ứng dụng và cập nhật giao diện mà không làm ảnh hưởng đến UI Thread.
- Chuỗi JavaScript (JavaScript Thread): Chuỗi này chứa mã JavaScript của ứng dụng và chịu trách nhiệm thực thi (chuỗi thực thi). Mã của chuỗi này thực thi trong một môi trường độc lập và đảm bảo ứng dụng chạy mượt mà.

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]

Chương 1: Cơ sở lý thuyết

Tailwind CSS là một framework CSS utility-first, tức là thay vì sử dụng các lớp CSS truyền thống để định nghĩa giao diện, Tailwind CSS cung cấp các lớp tiện ích sẵn có để lập trình viên có thể trực tiếp áp dụng vào các thành phần HTML. Điều này giúp cho quá trình xây dựng giao diện trở nên 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.
- 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.

Chương 1: Cơ sở lý thuyết

- 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 cung cấp bởi Google, cho phép các nhà phát triển xây dựng ứng dụng một cách dễ dàng với các dịch vụ như cơ sở dữ liệu, lưu trữ tệp, xác thực, và các chức năng backend khác mà không cần viết backend từ đầu. Firebase giúp phát triển các ứng dụng đa nền tảng, đặc biệt là ứng dụng di động, một cách nhanh chóng và hiệu quả.

1.3.2. Ưu điểm và nhược điểm^[7]

- Ưu điểm

- Cơ sở dữ liệu đáng tin cậy và mở rộng: Cơ sở dữ liệu Firebase của Google cung cấp khả năng mạnh mẽ cho việc phát triển ứng dụng web và di động. Một số dịch vụ phổ biến của Firebase bao gồm Realtime Database, Cloud

Chương 1: Cơ sở lý thuyết

Storage, Crashlytics và Analytics, với các tính năng như Đăng nhập một lần (Single Sign-On).

- Lưu trữ nhanh và bảo mật: 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. Hơn nữa, cho dù bạn muốn lưu trữ các dịch vụ vi mô của Express.js, HTML, CSS, hay API, Firebase luôn có dịch vụ lưu trữ sẵn sàng. Điều này cho thấy Firebase hỗ trợ lưu trữ nhiều loại nội dung khác nhau.
- Dịch vụ kiểm thử tốt nhất: Một ưu điểm khác của Firebase là các dịch vụ kiểm thử, 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 kiểm thử khác nhau, không chỉ giới hạn ở một loại duy nhất.
- Xác thực Firebase: Tâm quan trọng của nhận dạng người dùng trong bảo mật ứng dụng là không thẻ phủ nhận. Một trong những tính năng hấp dẫn nhất của nền tảng này trong môi trường bảo mật trực tuyến hiện tại là dịch vụ Firebase Authentication do Google hỗ trợ. Nó cung cấp SDK dễ tích hợp, các khung UI và dịch vụ máy chủ để xác minh người dùng trước khi họ sử dụng một ứng dụng Firebase cụ thể. Firebase Authentication 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 hoạt động với các 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 di động và web của họ. Firebase Analytics cũng có thể giúp cải thiện tỷ lệ giữ chân và tương tác của người dùng với ứng dụng.
 - **Nhược điểm**
- Lưu trữ dữ liệu không tiện lợi: Thông thường, bạn sẽ sử dụng Realtime Database như kho lưu trữ chính, điều này không phải lúc nào cũng là lựa chọn tốt. Vấn đề cơ bản là khả năng truy vấn bị hạn chế. Bạn không thể truy vấn

Chương 1: Cơ sở lý thuyết

nhiều hơn một khóa tại một thời điểm và không có cách nào để lọc dữ liệu.

Toàn bộ cơ sở dữ liệu là một tệp JSON, điều này khác xa so 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.

1.3.3. Firebase Authentication SDK.^[8]

1.3.3.1. Khái niệm

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.

1.3.3.2. 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.
- 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

Chương 1: Cơ sở lý thuyết

- 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.

1.3.4.2. Ưu điểm và nhược điểm

- **Ưu điểm**
- **Dễ dàng tích hợp**: 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.

Chương 1: Cơ sở lý thuyết

- 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

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

Chương 1: Cơ sở lý thuyết

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?

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

Chương 1: Cơ sở lý thuyết

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.
- 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?

Chương 1: Cơ sở lý thuyết

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.
- 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.

Chương 1: Cơ sở lý thuyết

- 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.
- Tối ưu chi phí: Phù hợp cho các doanh nghiệp nhỏ và vừa nhờ vào mức giá hợp lý.

- Nhược điểm:

Chương 1: Cơ sở lý thuyết

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

Chương 1: Cơ sở lý thuyết

- **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:

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 THỰC TRẠNG 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 thực trạng 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

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

- Thông tin về nhà phát triển

Chương 2: Khảo sát thực trạng và xác định yêu cầu

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: Đễ 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.
- 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.

Chương 2: Khảo sát thực trạng và xác định yêu cầu

- 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.

- **Thông tin ứng dụng**

Chương 2: Khảo sát thực trạng và xác định yêu cầu

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 đối 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).
- 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).

Chương 2: Khảo sát thực trạng và xác định yêu cầu

- 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 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,

Chương 2: Khảo sát thực trạng và xác định yêu cầu

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.

- 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:

Chương 2: Khảo sát thực trạng và xác định yêu cầu

- Người dùng có thể nhập 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 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.

Chương 2: Khảo sát thực trạng và xác định yêu cầu

- 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.
- 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.

Chương 2: Khảo sát thực trạng và xác định yêu cầu

- 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 khi 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 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):**

Chương 2: Khảo sát thực trạng và xác định yêu cầu

- 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.
 - 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

Chương 2: Khảo sát thực trạng và xác định yêu cầu

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

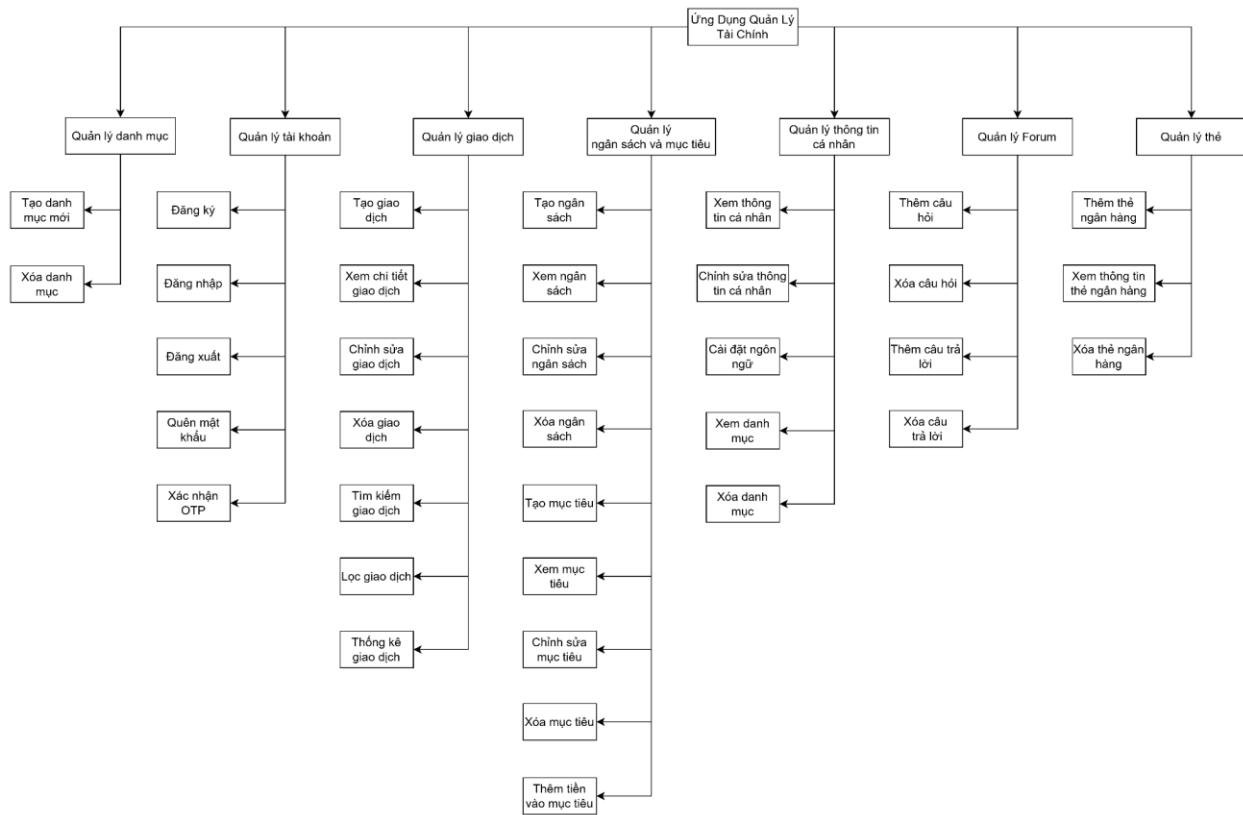
- **Hỗ trợ đa ngôn ngữ và địa phương hóa:**
 - Ứng dụng cần tích hợp khả năng hỗ trợ nhiều ngôn ngữ và tiền tệ khác nhau, giúp người dùng từ nhiều quốc gia có thể sử dụng dễ dàng. Khả năng điều chỉnh theo ngôn ngữ và văn hóa địa phương sẽ giúp ứng dụng phù hợp với đối tượng người dùng toàn cầu.
 - Việc 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

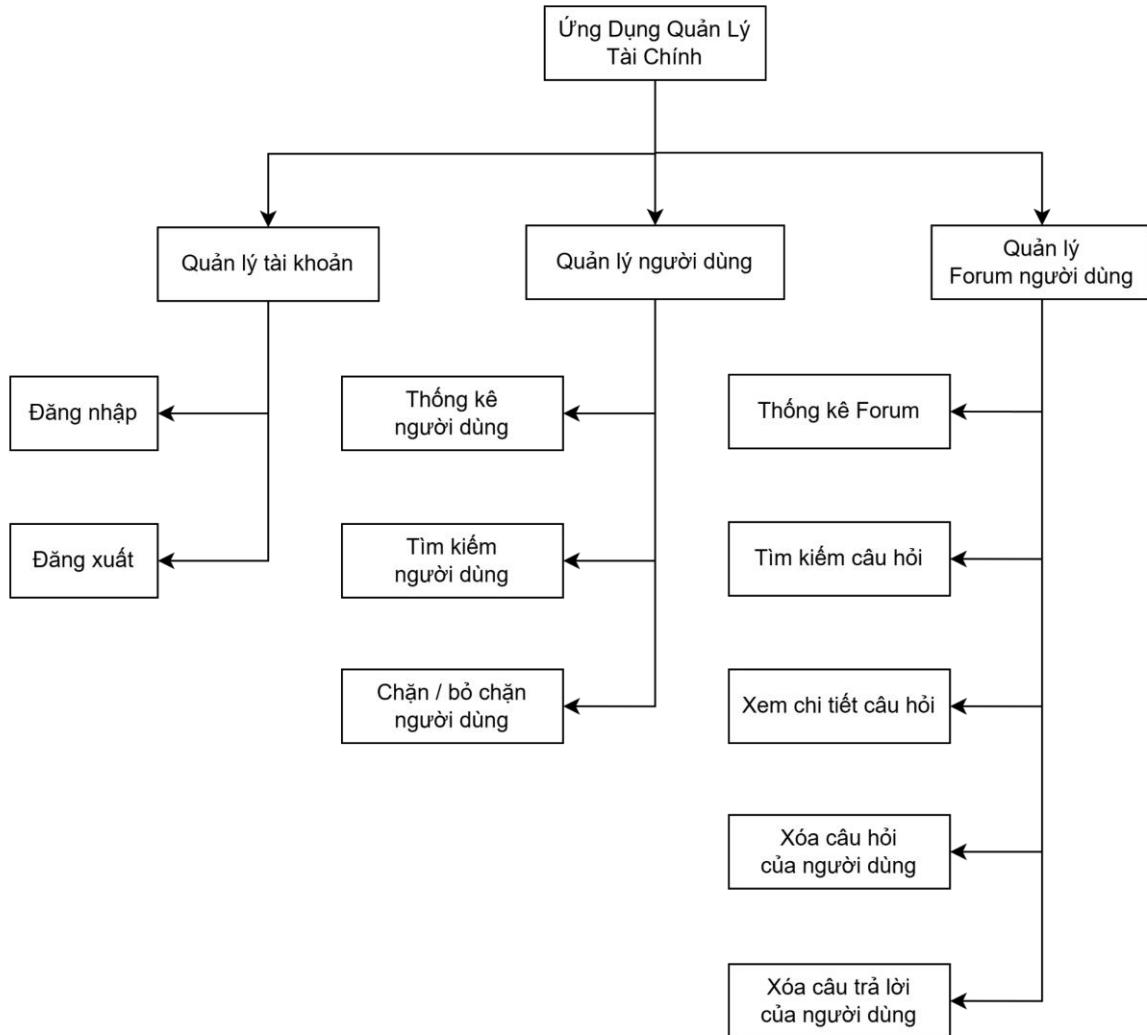
a. Đối với người dùng



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

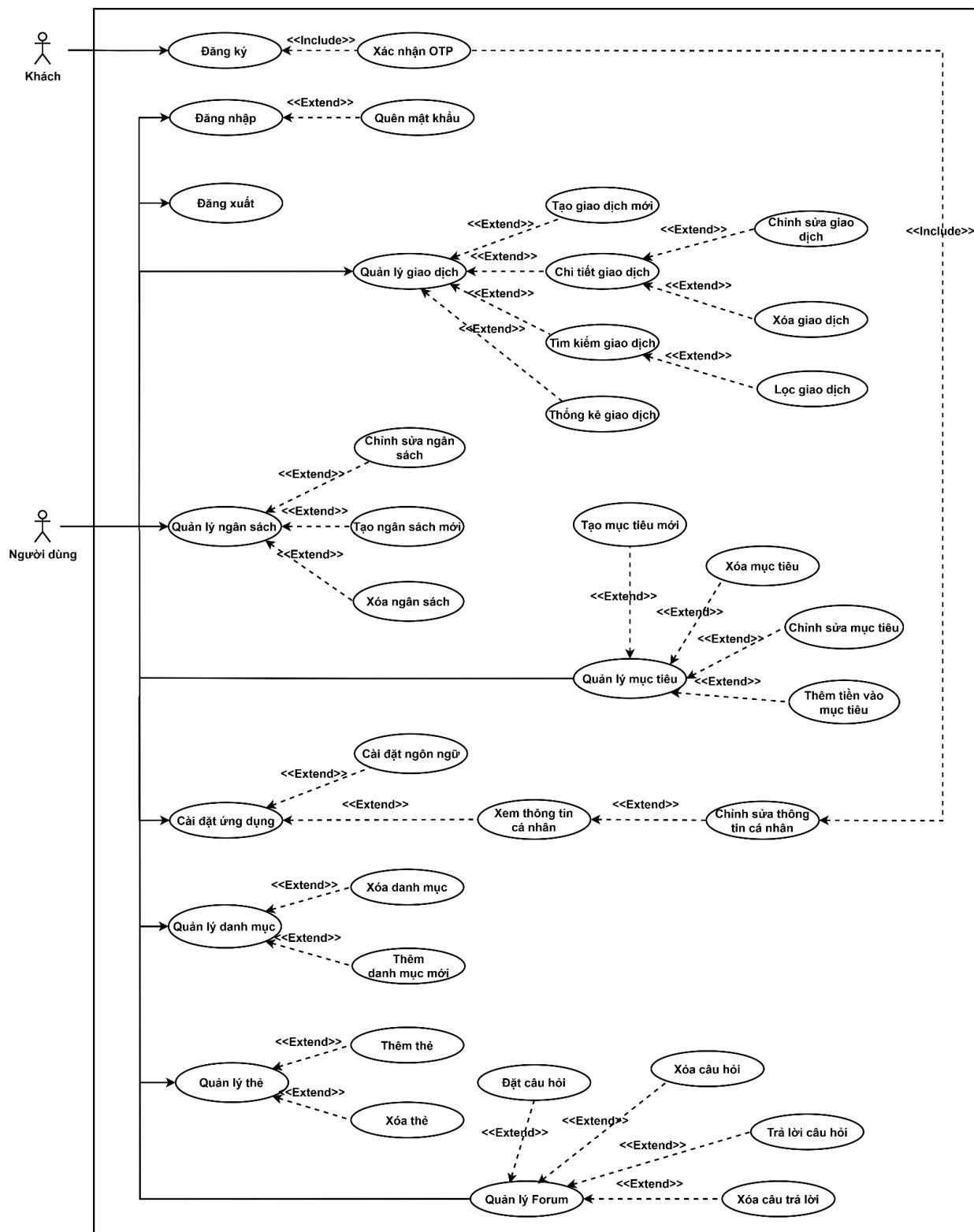


Hình 1b. Sơ đồ phân rã chức năng đối với quản trị viên.

3.1.2. Lược đồ Use case

a. Đối với người dùng

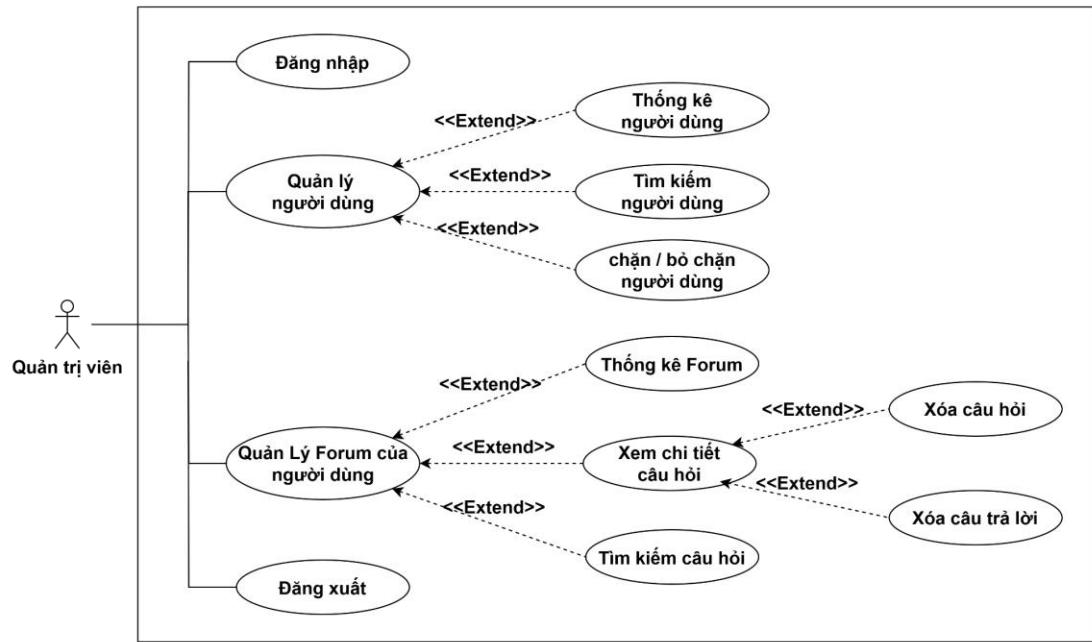
Chương 3: Phân tích thiết kế hệ thống



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

Chương 3: Phân tích thiết kế hệ thống

b. Đối với quản trị viên

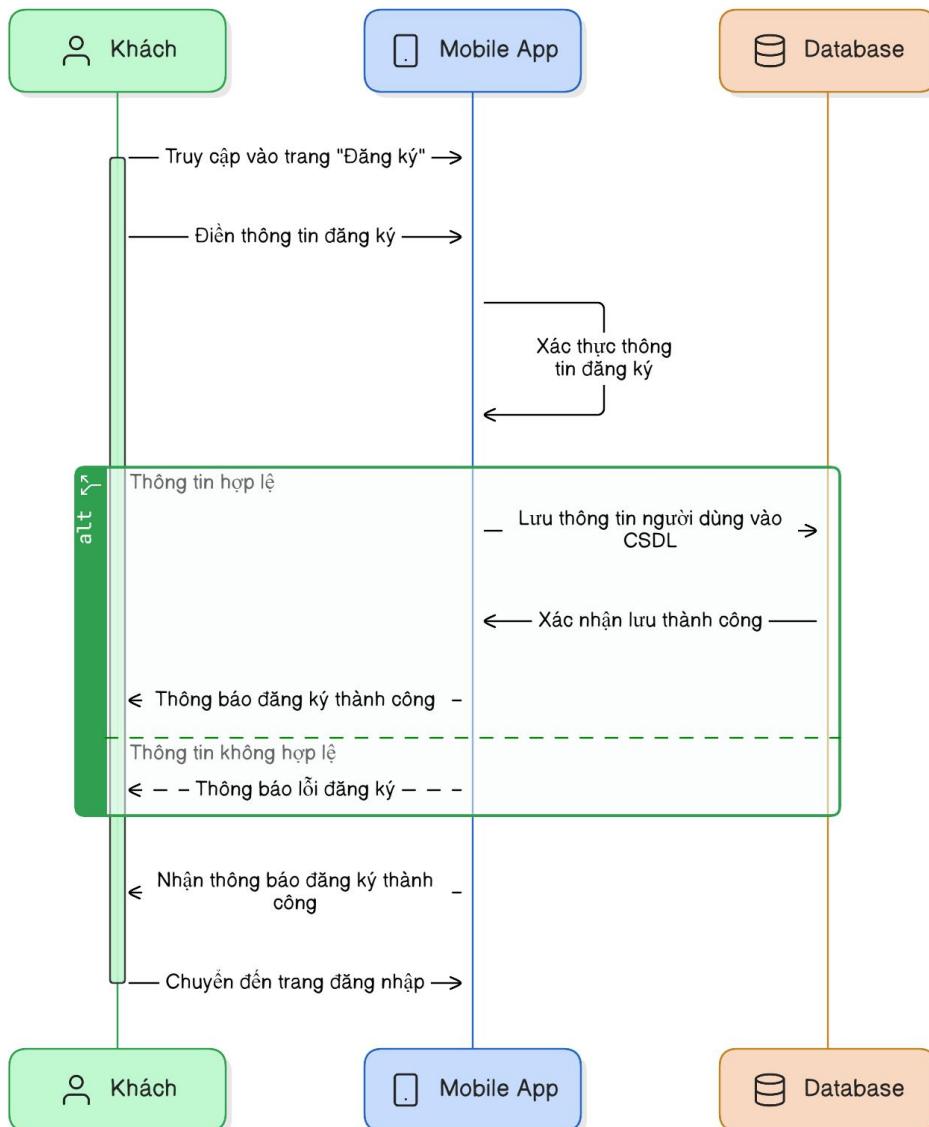


Hình 2b. Lược đồ Use case đối với quản trị viên.

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.

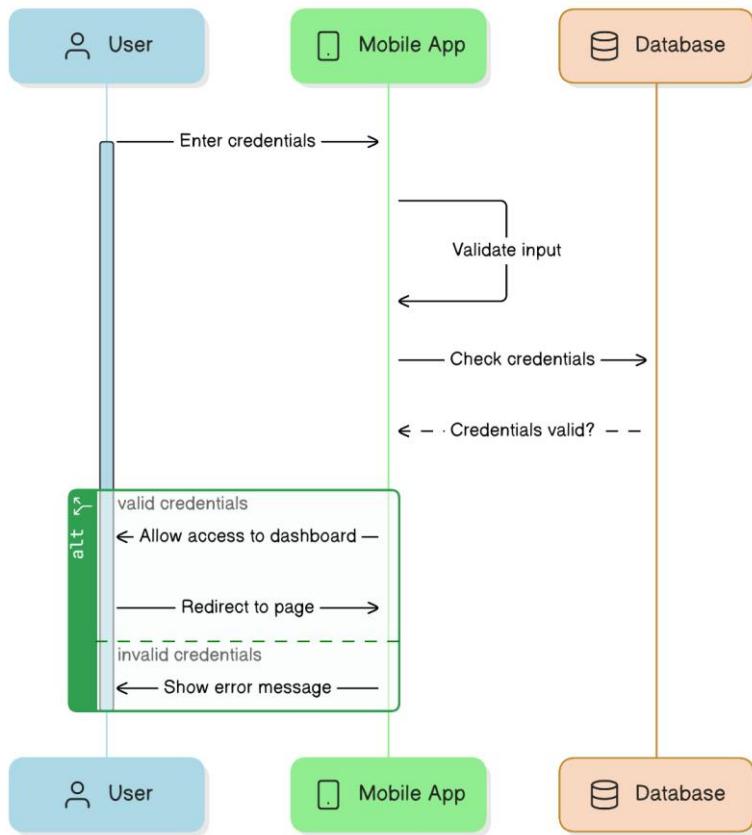
Chương 3: Phân tích thiết kế hệ thống

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.

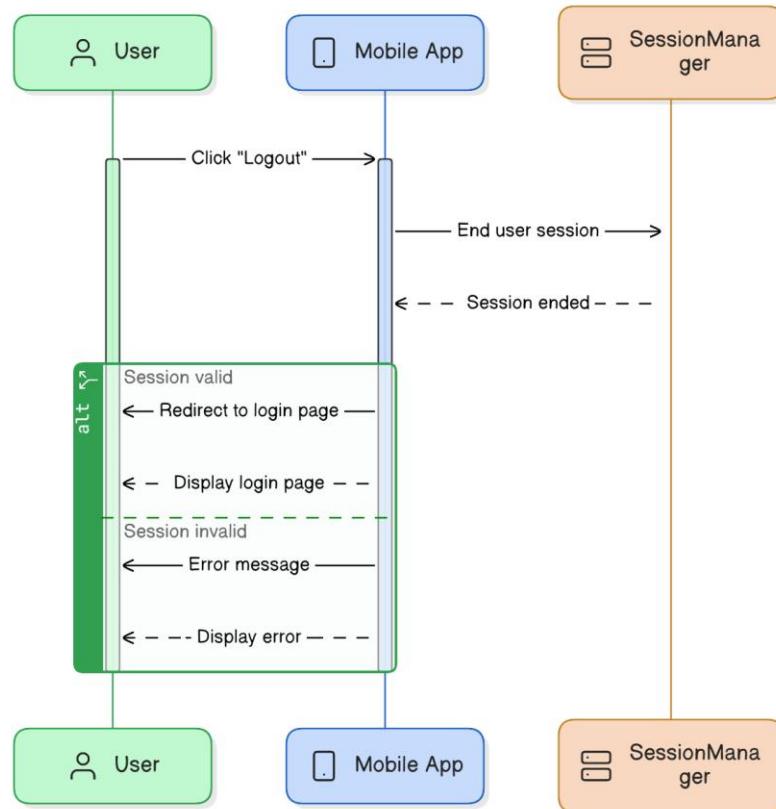
Chương 3: Phân tích thiết kế hệ thống

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"> Người dùng nhập tài khoản và mật khẩu vào các ô input và bấm "Login". Hệ thống kiểm tra input và xác thực. 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

3.1.3.3. Đăng xuất

User Logout Process



Chương 3: Phân tích thiết kế hệ thống

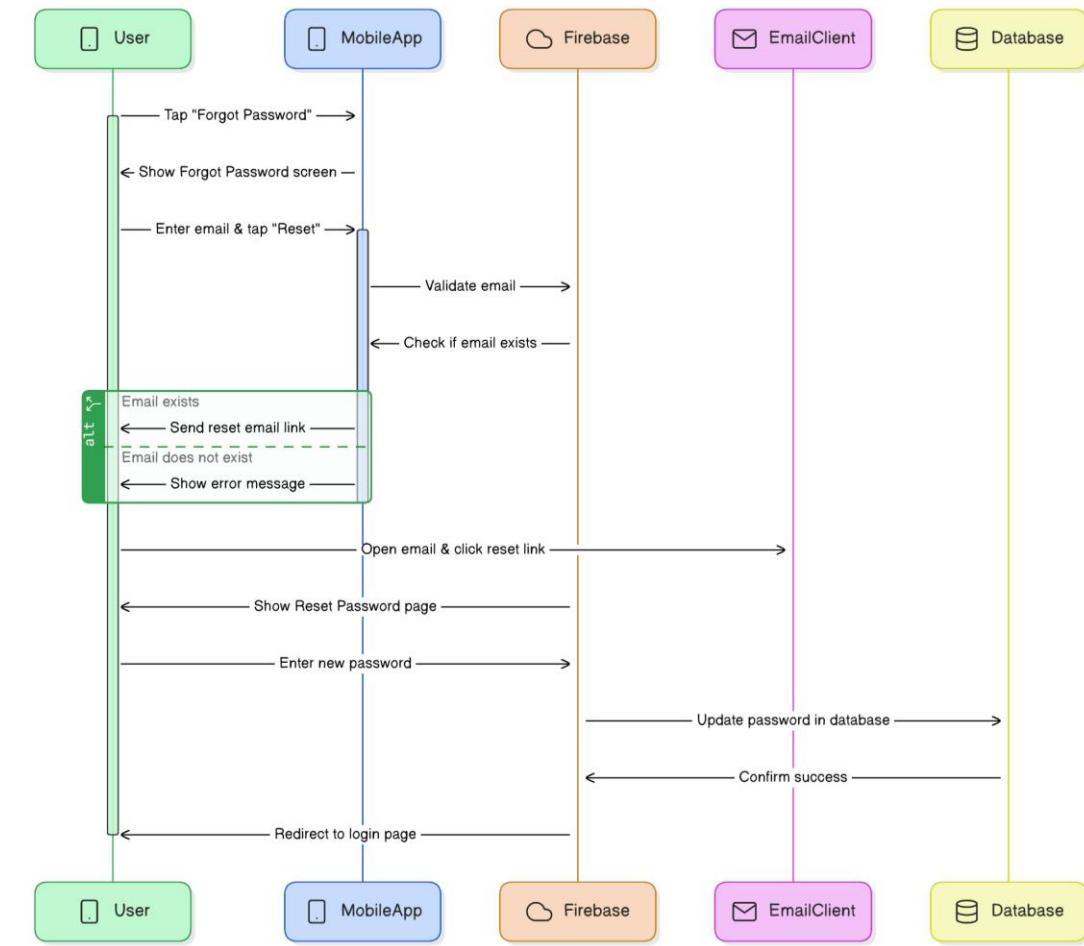
Hình 3.3. Sequence - Đăng xuất

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

Bảng 1.3. Đặc tả Use case - Đăng xuất

Chương 3: Phân tích thiết kế hệ thống

3.1.3.4. Quên mật khẩu



Hình 3.4. Sequence - Quên mật khẩu

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.

Chương 3: Phân tích thiết kế hệ thống

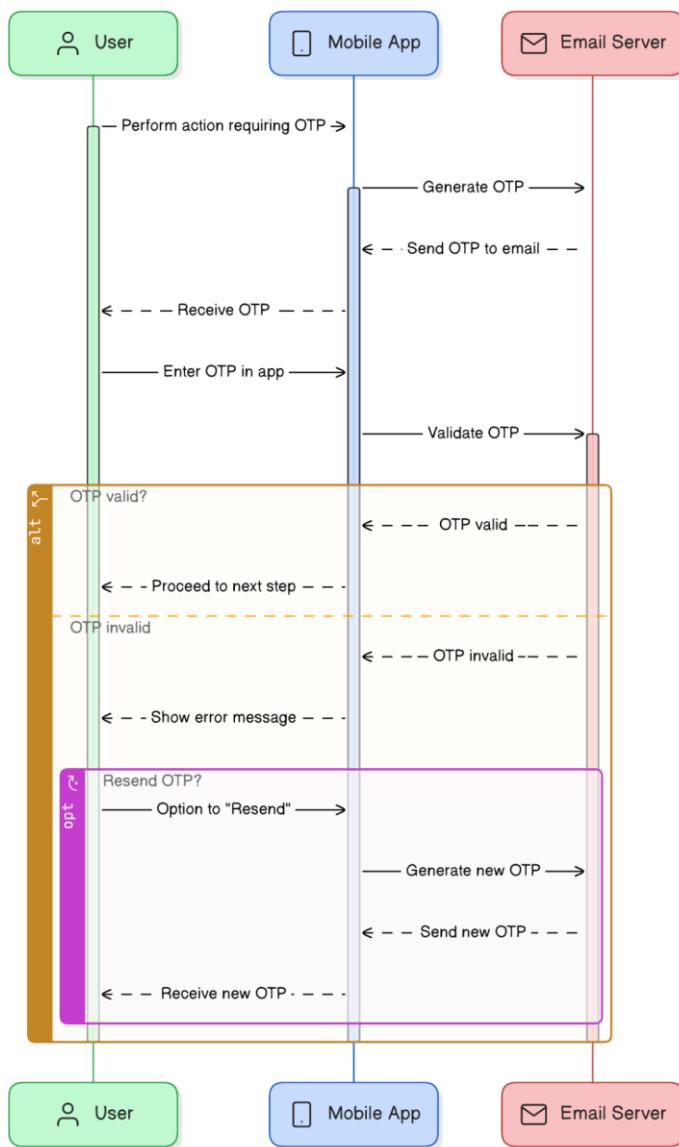
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng chọn "Forgot Password" ở trang Login. 2. Hệ thống hiển thị trang ForgotPassword . 3. Người dùng nhập email của mình và chọn nút “Reset Password”. 4. Hệ thống kiểm tra email và gửi mail đặt lại mật khẩu đến email của người dùng. 5. Người dùng mở email đặt lại mật khẩu và chọn nút đặt lại mật khẩu. 6. Người dùng được chuyển đến trang đặt lại mật khẩu. 7. Người dùng nhập mật khẩu mới và chọn “Reset”. 8. Đặt lại mật khẩu thành công, người dùng được chuyển về trang Login.
Alternative flow	<p>Đặt lại mật khẩu thất bại.</p> <ol style="list-style-type: none"> 1. Hệ thống kiểm tra email thất bại. 2. Hệ thống hiện thông báo lỗi. 3. Người dùng nhập lại email. <p>Use Case quay lại bước 3 của Basic flow.</p>
Extension point	

Bảng 1.4. Đặc tả Use case - Quên mật khẩu

Chương 3: Phân tích thiết kế hệ thống

3.1.3.5. Xác nhận OTP

OTP Verification Process



Hình 3.5. Sequence - Xác nhận OTP

Name	Xác nhận OTP
------	--------------

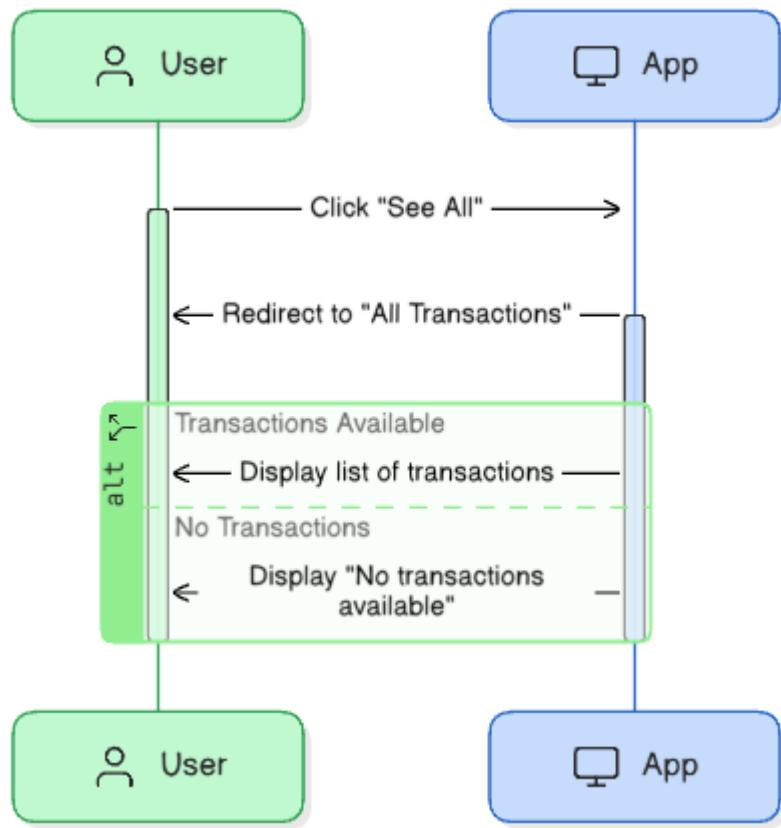
Chương 3: Phân tích thiết kế hệ thống

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.
Extension point	

Bảng 1.5. Đặc tả Use case - Xác nhận OTP

Chương 3: Phân tích thiết kế hệ thống

3.1.3.6. Quản lý giao dịch



Hình 3.6. Sequence - Quản lý giao dịch

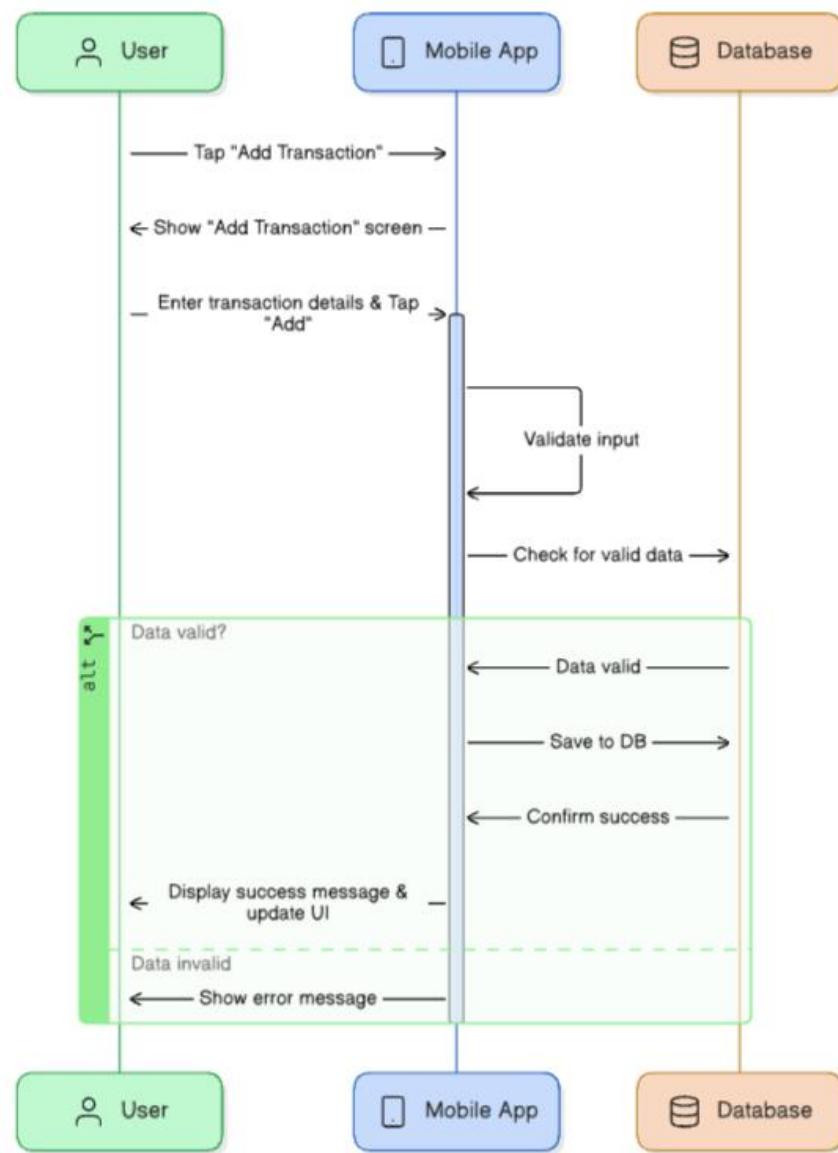
Name	Chi tiết giao dịch.
Brief description	Giúp người dùng quản lý các giao dịch thu chi.
Actor(s)	Người dùng
Pre-conditions	Actors đã có tài khoản trong hệ thống.
Post-conditions	Người dùng mở được trang “All Transactions”
Flow of events	
Basic flow	1. Người dùng nhấp vào “See All” trong trang Home

Chương 3: Phân tích thiết kế hệ thống

	2. Hệ thống chuyển đến trang “All Transactions”
Alternative flow	
Extension point	

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

3.1.3.7. Tạo giao dịch mới



Hình 3.7. Sequence - Tạo giao dịch mới

Chương 3: Phân tích thiết kế hệ thống

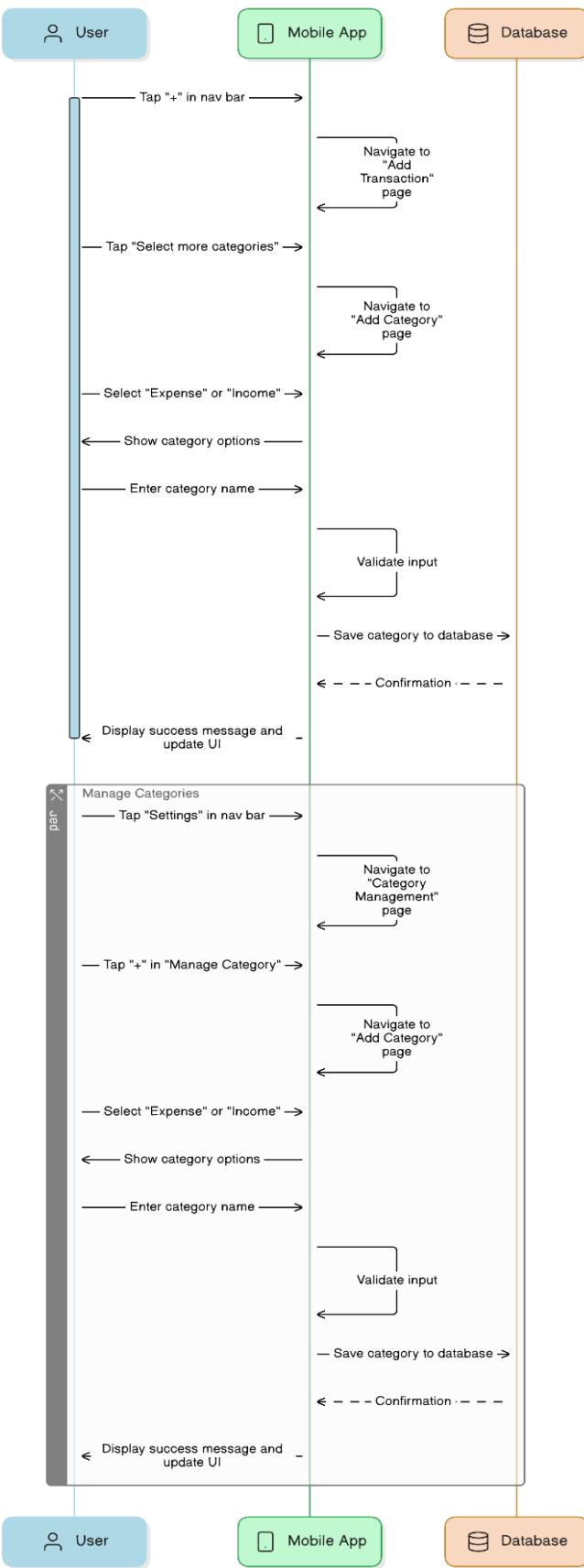
Name	Tạo giao dịch mới.
Brief description	Giúp người dùng thêm thông tin giao dịch mới.
Actor(s)	Người dùng
Pre-conditions	Actors đã có tài khoản trong hệ thống và đã đăng nhập thành công vào hệ thống.
Post-conditions	Hệ thống hiển thị giao dịch vừa thêm của người dùng.
Flow of events	
Basic flow	<ol 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

Chương 3: Phân tích thiết kế hệ thống

3.1.3.8. Thêm danh mục mới

Chương 3: Phân tích thiết kế hệ thống



Chương 3: Phân tích thiết kế hệ thống

Hình 3.8. Sequence - Thêm danh mục mới

Name	Thêm danh mục mới.
Brief description	Người dùng thêm các danh mục chi tiêu và thu nhập.
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 ghi nhận danh mục mới vào cơ sở dữ liệu, hiển thị các danh mục đã thêm trong các trang khác
Flow of events	
Basic flow	<p>Người dùng thêm trong Add Transaction:</p> <ol style="list-style-type: none">1. Người dùng nhấn vào nút “+” ở giữa thanh điều hướng2. Hệ thống điều hướng đến trang “Add Transaction”3. Người dùng nhấn vào “Select more categories”4. Hệ thống chuyển đến trang “Add Category”5. Người dùng chọn “Expense” hoặc “Income” để hệ thống hiển thị loại danh mục tương ứng.6. Người dùng nhập tên cho danh mục7. Người dùng nhấn “Add Category”8. Hệ thống ghi nhận danh mục mới vào cơ sở dữ liệu <p>Người dùng thêm trong “Category Management”:</p> <ol style="list-style-type: none">1. Người dùng nhấn vào “Settings” trên thanh điều hướng2. Người dùng nhấn vào “Manage Category”3. Người dùng nhấn vào icon “+”4. Hệ thống chuyển đến trang “Add Category”

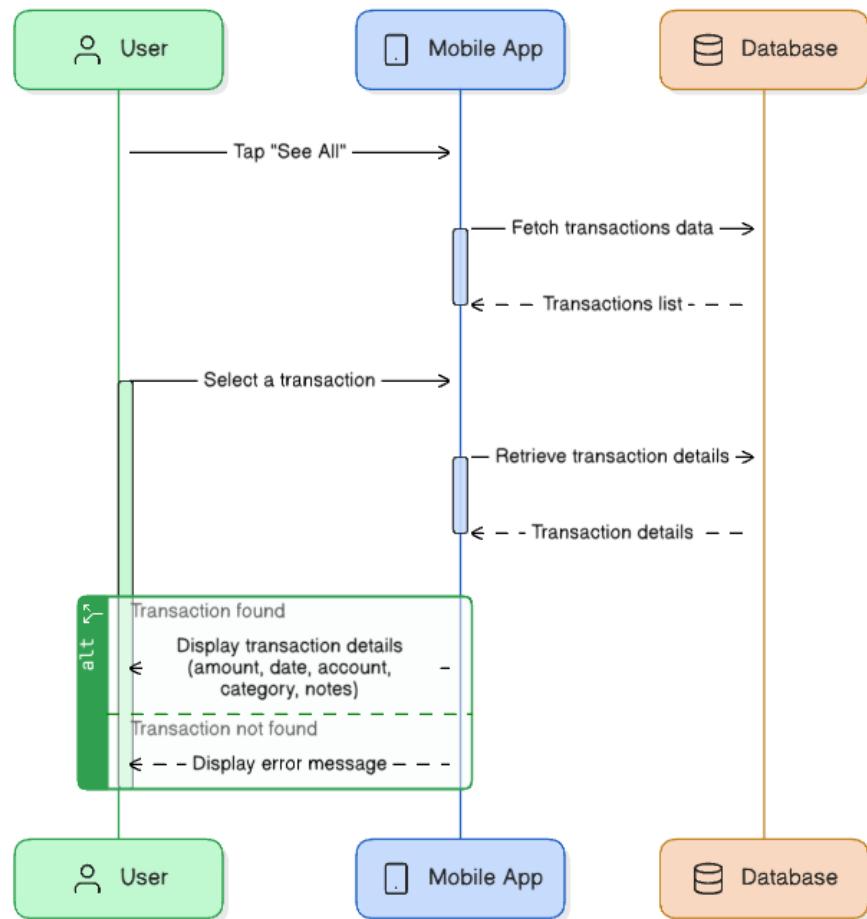
Chương 3: Phân tích thiết kế hệ thống

	<ol style="list-style-type: none">5. Người dùng chọn “Expense” hoặc “Income” để hệ thống hiển thị loại danh mục tương ứng.6. Người dùng nhập tên cho danh mục7. Người dùng nhấn “Add Category”8. Hệ thống ghi nhận danh mục mới vào cơ sở dữ liệu
Alternative flow	
Extension point	

Bảng 1.8. Đặc tả Use case - Thêm danh mục mới

Chương 3: Phân tích thiết kế hệ thống

3.1.3.9. Chi tiết giao dịch



Hình 3.9. Sequence - Chi tiết giao dịch

Name	Chi tiết giao dịch.
Brief description	Người dùng có thể chọn một giao dịch từ danh sách giao dịch và xem chi tiết của giao dịch đó, bao gồm số tiền, tài khoản, danh mục, ngày giờ, và ghi chú lýên quan.
Actor(s)	Người dùng
Pre-conditions	1. Người dùng đã đăng nhập vào hệ thống.

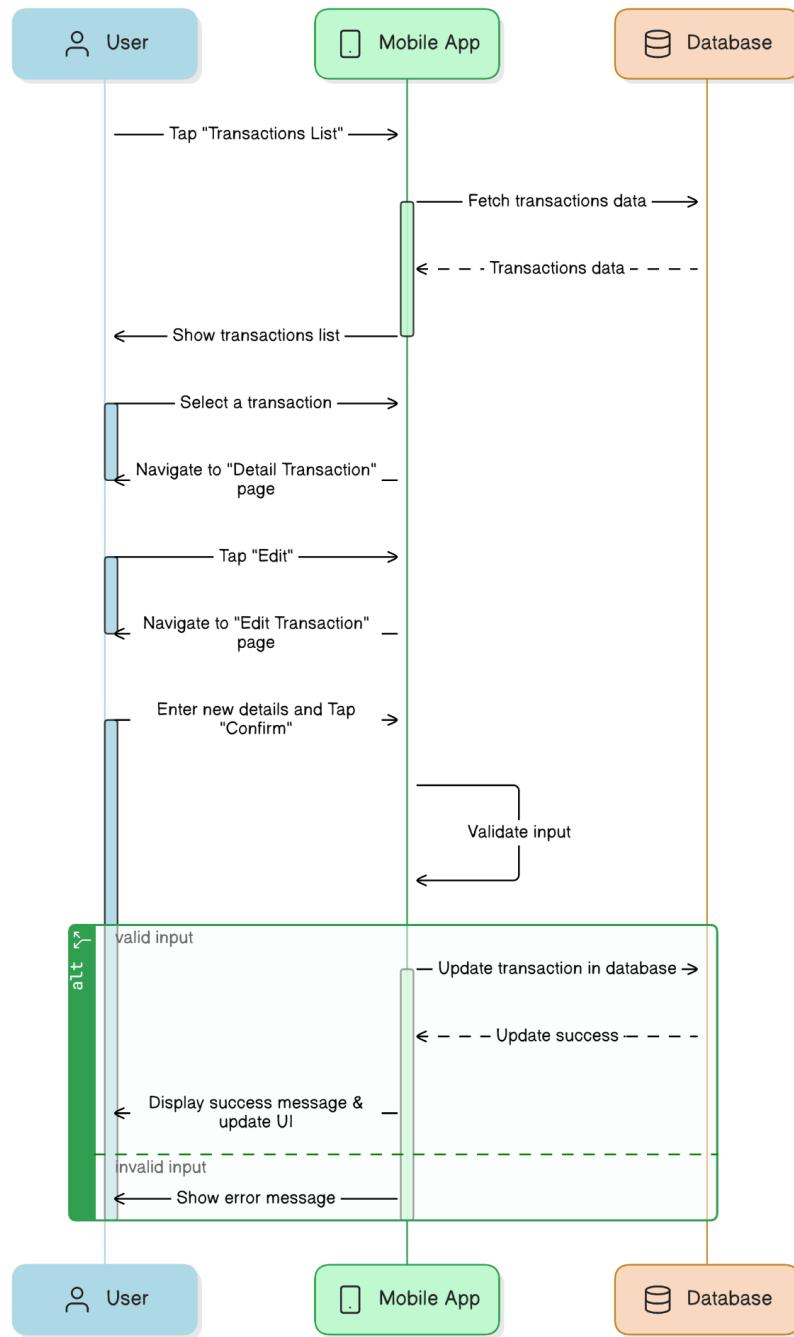
Chương 3: Phân tích thiết kế hệ thống

	2. Giao dịch đã được lưu trữ trước đó trong hệ thống.
Post-conditions	Người dùng có thể xem thông tin chi tiết của giao dịch đã chọn.
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng vào mục "See all" từ giao diện chính.2. Hệ thống hiển thị danh sách các giao dịch.3. Người dùng chọn một giao dịch từ danh sách.4. Hệ thống hiển thị màn hình chi tiết giao dịch, bao gồm số tiền, ngày giờ, tài khoản, danh mục, và các ghi chú liên quan.
Alternative flow	Hệ thống báo không tìm thấy giao dịch.
Extension point	

Bảng 1.9. Đặc tả Use case - Chi tiết giao dịch

Chương 3: Phân tích thiết kế hệ thống

3.1.3.10. Chính sửa giao dịch



Hình 3.10. Sequence - Chính sửa giao dịch

Name	Chỉnh sửa giao dịch.
------	----------------------

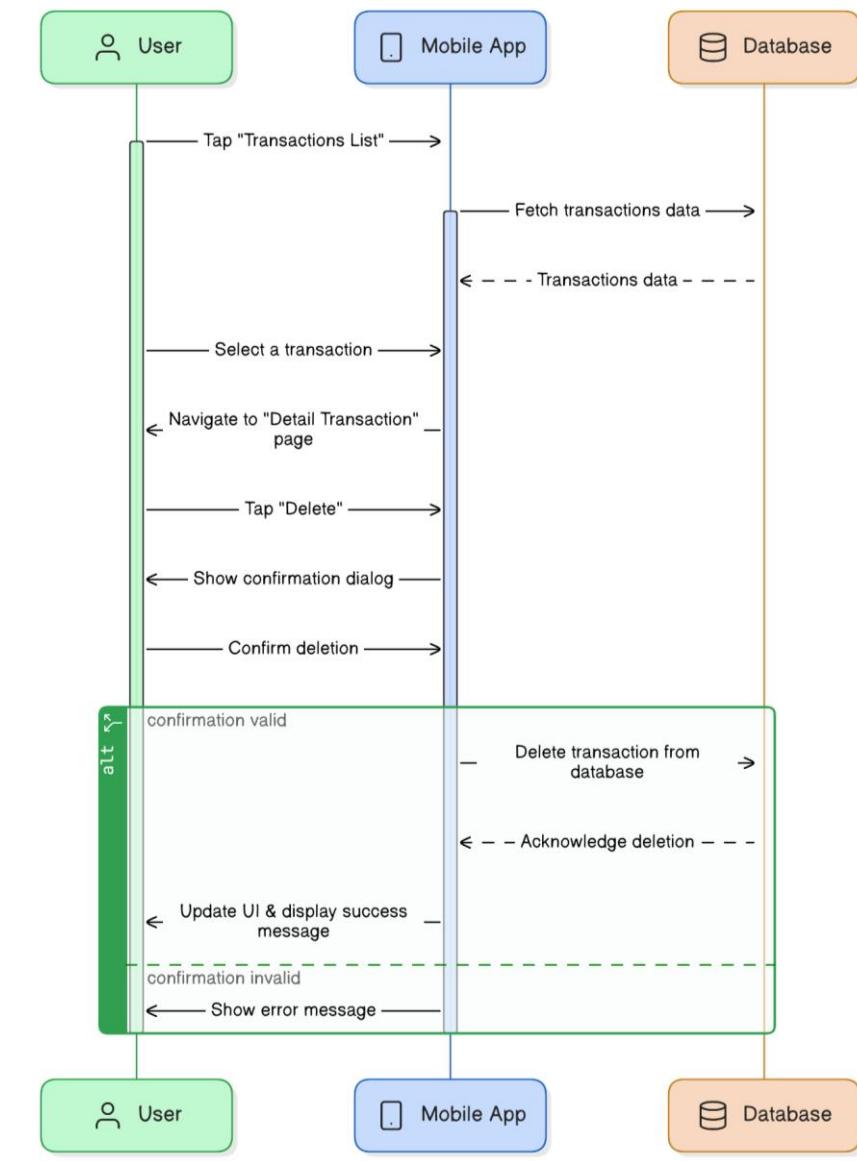
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng chỉnh sửa thông tin của một giao dịch đã tồn tại trong danh sách giao dịch.
Actor(s)	Người dùng
Pre-conditions	Actors đã có các giao dịch được ghi nhận.
Post-conditions	Thông tin giao dịch được cập nhật trong cơ sở dữ liệu.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập phần "Transactions lýst" và chọn một giao dịch cụ thể. 2. Hệ thống sẽ dẫn người dùng vào trang Detail Transaction có chứa nút Edit. 3. Người dùng, nhấn nút Edit để chuyển sang trang thay đổi thông tin giao dịch (ví dụ: số tiền, ghi chú, loại giao dịch) rồi nhấn nút Confirm. 4. Hệ thống xác thực và cập nhật thông tin giao dịch trong cơ sở dữ liệu.
Alternative flow	Nếu thông tin không hợp lệ, hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin chính xác.
Extension point	

Bảng 1.10. Đặc tả Use case - Chính sửa giao dịch

Chương 3: Phân tích thiết kế hệ thống

3.1.3.11. Xóa giao dịch



Hình 3.11. Sequence - Xóa giao dịch

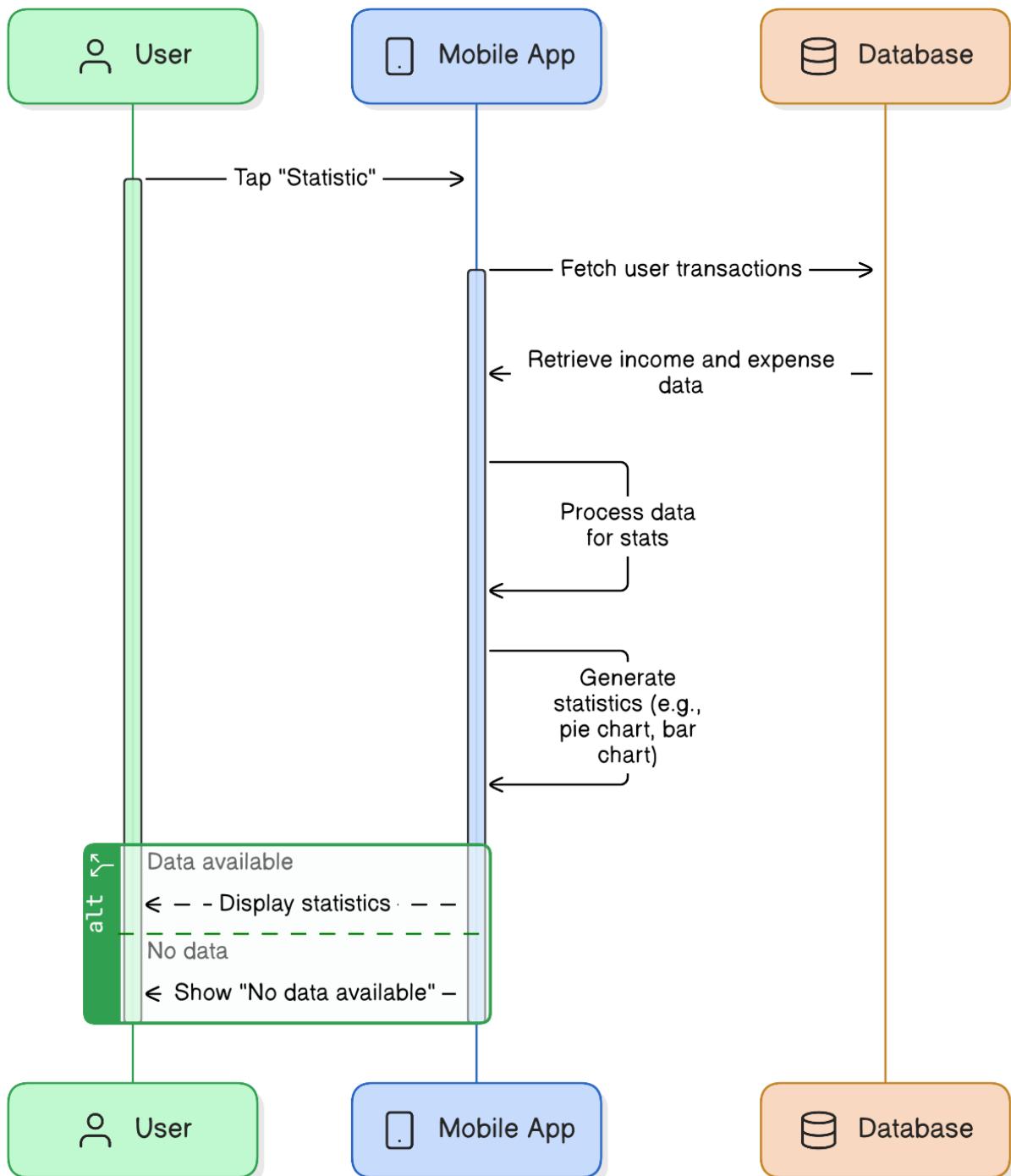
Name	Xóa giao dịch.
Brief description	Người dùng xóa một giao dịch đã tồn tại khỏi danh sách giao dịch của mình.

Chương 3: Phân tích thiết kế hệ thống

Actor(s)	Người dùng
Pre-conditions	Actors đã có các giao dịch trong hệ thống.
Post-conditions	Giao dịch được xóa khỏi cơ sở dữ liệu và không còn xuất hiện trong danh sách giao dịch.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập phần "Transactions list" và chọn một giao dịch. 2. Hệ thống sẽ hiển thị trang Detail Transaction, trong đó có chứa nút Delete Transaction. 3. Người dùng nhấn nút Delete, hệ thống sẽ hiện thông báo để xác thực yêu cầu của người dùng. 4. Người dùng nhấn nút Confirm để xác thực yêu cầu. 5. Hệ thống xóa giao dịch khỏi cơ sở dữ liệu và cập nhật danh sách giao dịch.
Alternative flow	Nếu có lỗi xảy ra khi xóa giao dịch, hệ thống thông báo lỗi và yêu cầu người dùng thử lại.
Extension point	

Bảng 1.11. ĐẶC TẢ USE CASE - XÓA GIAO DỊCH

3.1.3.12. Thống kê giao dịch



Hình 3.12. Sequence - Thống kê giao dịch

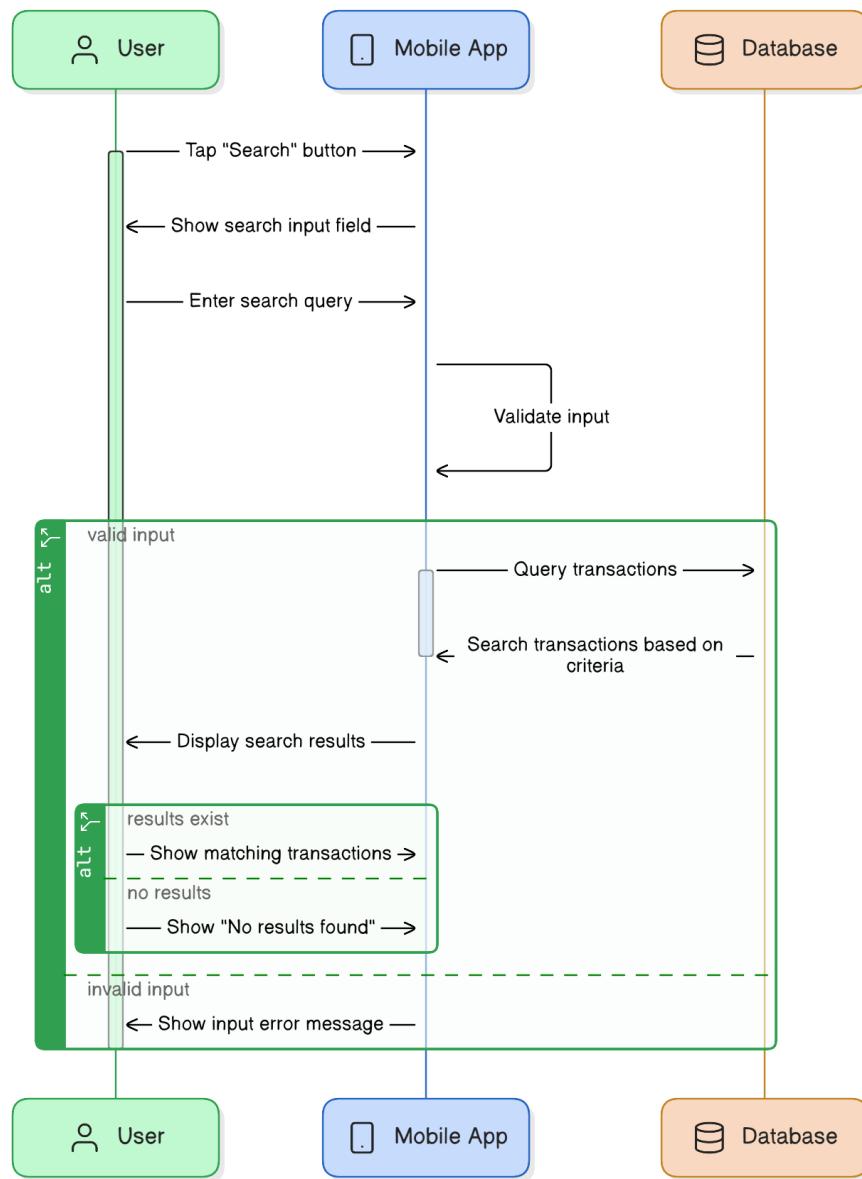
Chương 3: Phân tích thiết kế hệ thống

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	1. Người dùng chọn "Statistic". 2. 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	1. Hệ thống kiểm tra thấy thông tin thu nhập và chi tiêu của người dùng không tồn tại. 2. Hệ thống thông báo người dùng không có thông tin thu nhập và chi tiêu.
Extension point	

Bảng 1.12. Đặc tả Use case - Thống kê giao dịch

Chương 3: Phân tích thiết kế hệ thống

3.1.3.13. Tìm kiếm giao dịch



Hình 3.13. Sequence - Tìm kiếm giao dịch

Name	Tìm kiếm giao dịch.
Brief description	Người dùng có thể thực hiện tìm kiếm Transaction.
Actor(s)	Người dùng

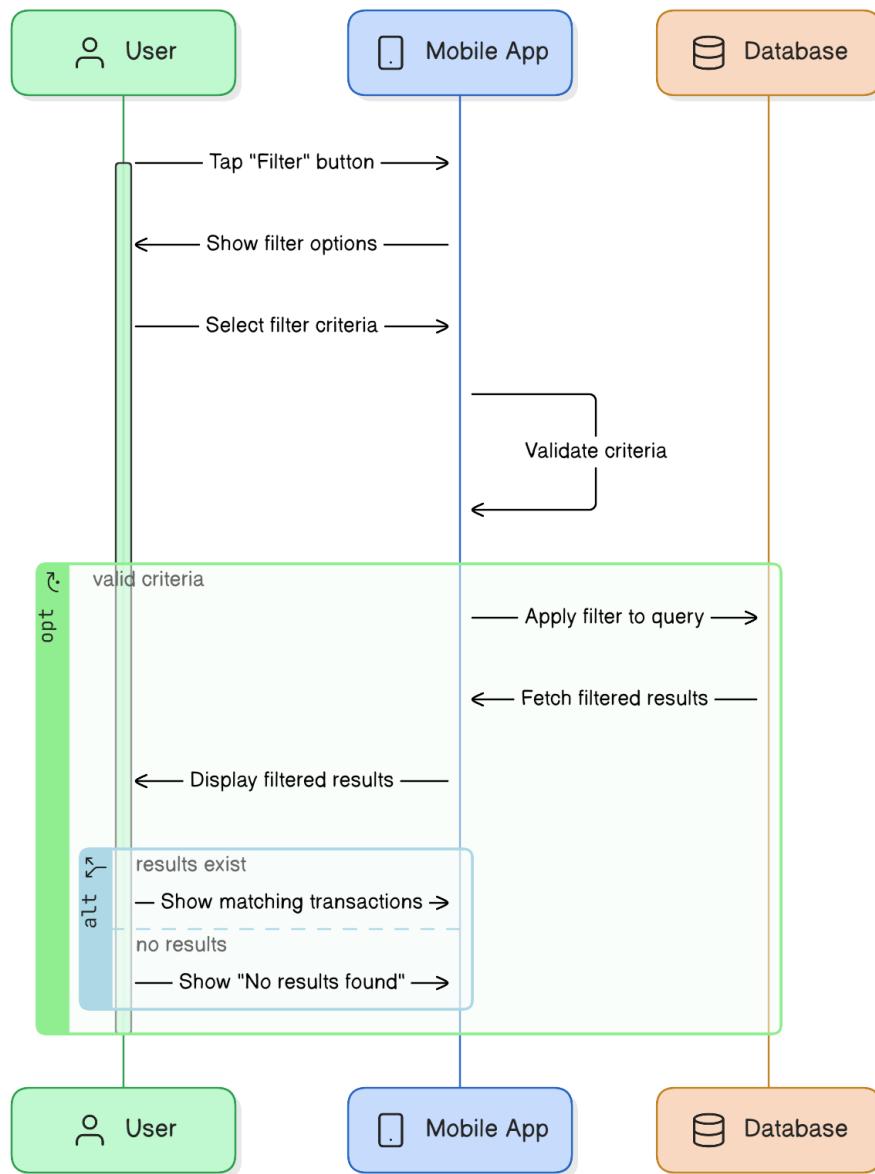
Chương 3: Phân tích thiết kế hệ thống

Pre-conditions	Actors đã có các giao dịch.
Post-conditions	Tìm được các giao dịch.
Flow of events	
Basic flow	<ol style="list-style-type: none">Người dùng bấm vào nút tìm kiếm trong phần Transaction.Người dùng nhập thông tin tìm kiếmHệ thống hiển thị kết quả tìm kiếm.
Alternative flow	Hệ thống không tìm thấy Transaction khi người dùng nhập sai thông tin tìm kiếm hoặc Transaction không tồn tại.
Extension point	

Bảng 1.13. Đặc tả Use case - Tìm kiếm giao dịch

Chương 3: Phân tích thiết kế hệ thống

3.1.3.14. Lọc giao dịch



Hình 3.14. Sequence - Lọc giao dịch

Name	Lọc giao dịch
Brief description	Người dùng có thể thực hiện lọc tìm Transaction.

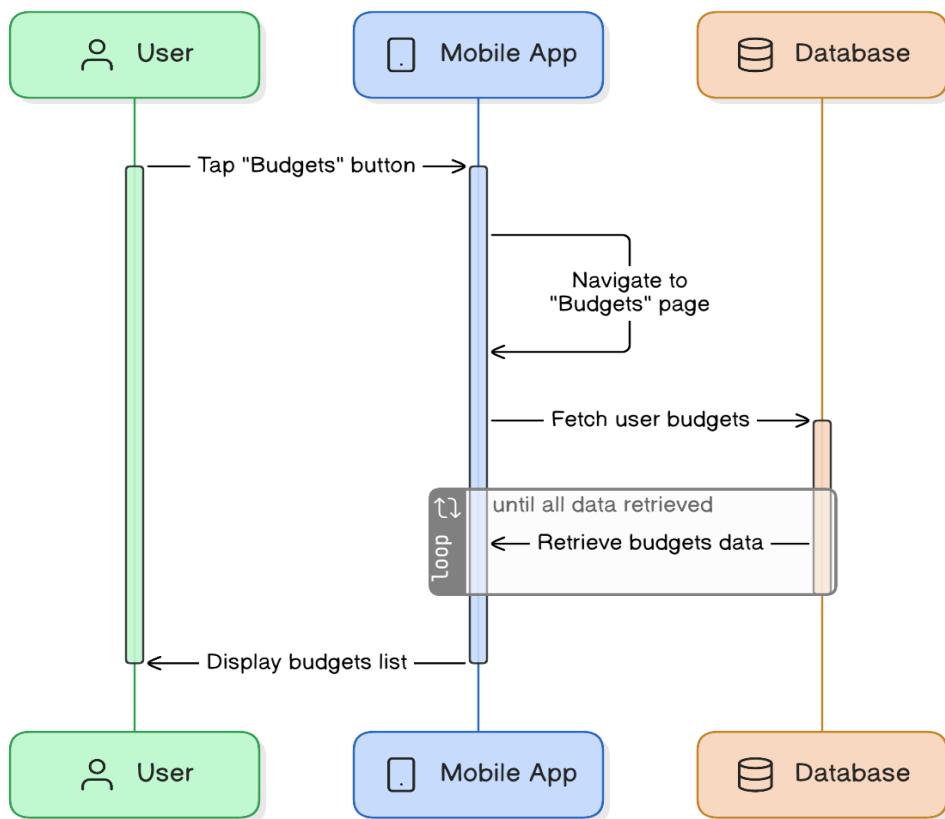
Chương 3: Phân tích thiết kế hệ thống

Actor(s)	Người dùng
Pre-conditions	Actors đã có các giao dịch.
Post-conditions	Tìm được các giao dịch.
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng bấm vào nút lọc trong phần Transaction.2. Người dùng chọn tiêu chí lọc.3. Hệ thống hiển thị kết quả lọc.
Alternative flow	Hệ thống không tìm thấy Transaction khi Transaction không tồn tại.
Extension point	

Bảng 1.14. Đặc tả Use case - Lọc giao dịch

Chương 3: Phân tích thiết kế hệ thống

3.1.3.15. Quản lý ngân sách



Hình 3.15. 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	
Basic flow	1. Người dùng nhấp vào “Budgets” trên thanh điều hướng

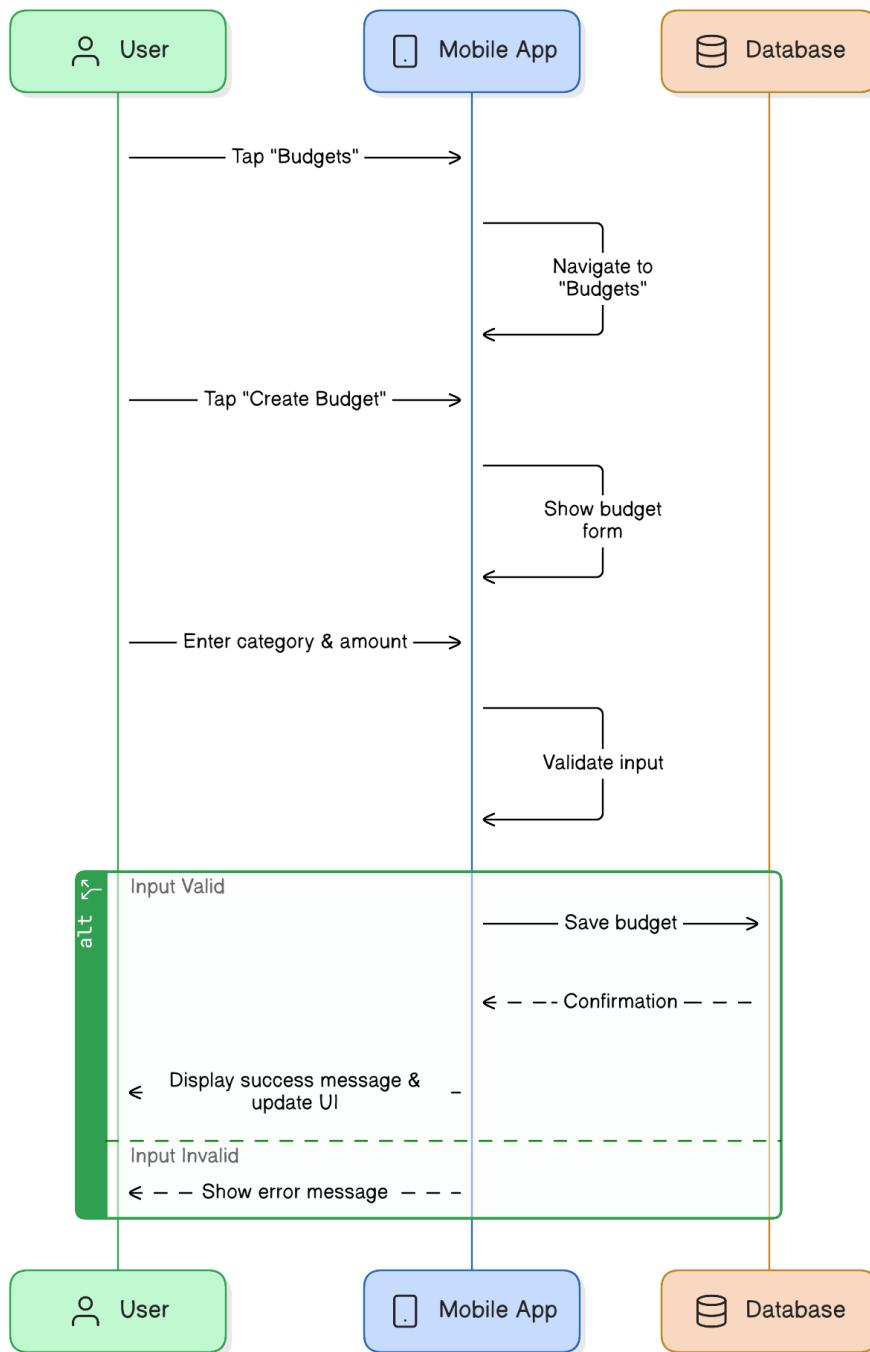
Chương 3: Phân tích thiết kế hệ thống

	2. Hệ thống điều hướng đến trang “Budgets”
Alternative flow	
Extension point	

Bảng 1.15. Đặc tả Use case - Quản lý ngân sách

Chương 3: Phân tích thiết kế hệ thống

3.1.3.16. Tạo ngân sách mới



Hình 3.16. Sequence - Tạo ngân sách mới

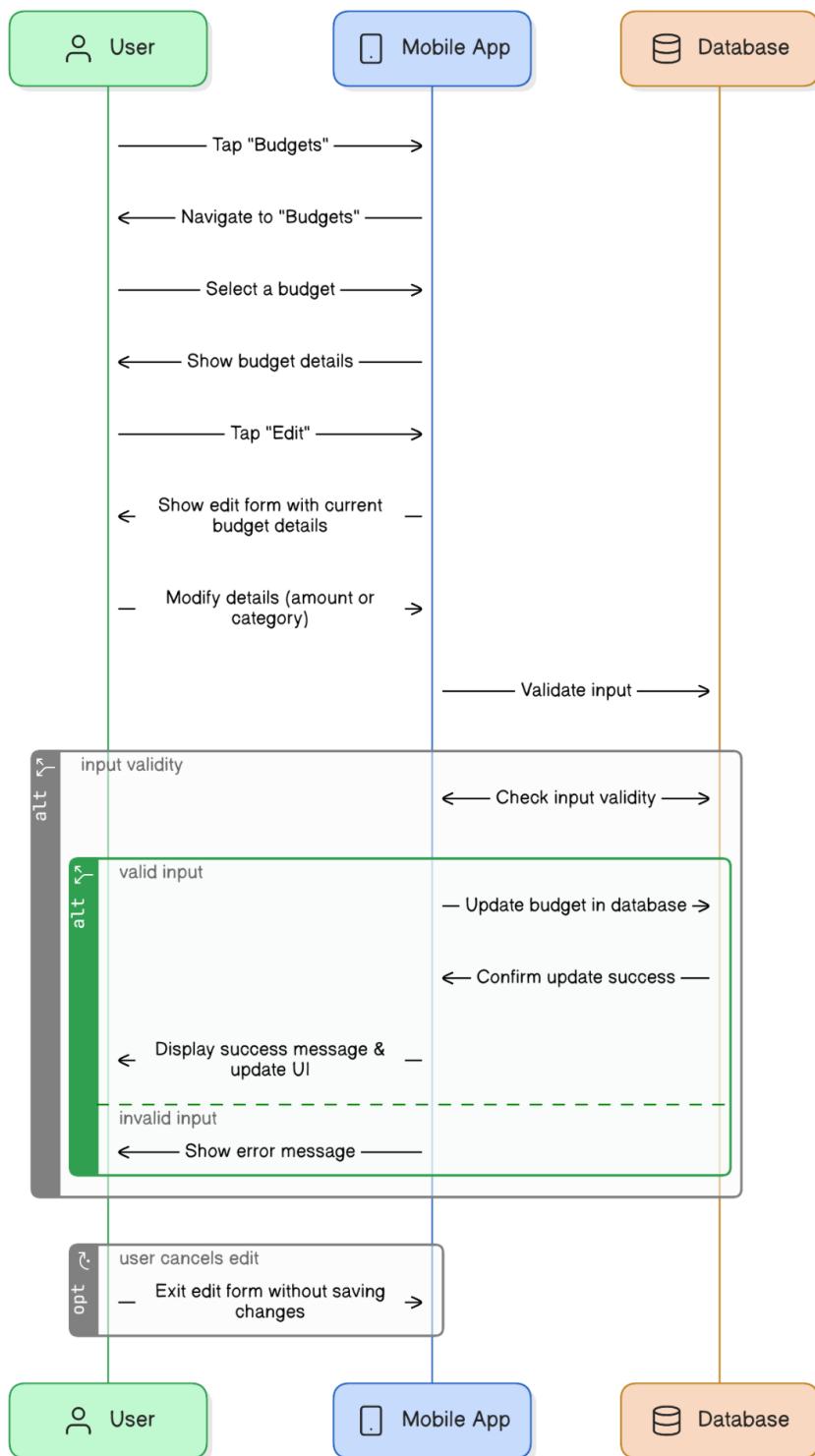
Chương 3: Phân tích thiết kế hệ thống

Name	Tạo ngân sách mới
Brief description	Người dùng tạo ra những hạn mức chi thu (hay cho gọi là ngân sách) cho những danh mục mình mong muốn trong phần Budget.
Actor(s)	Người dùng
Pre-conditions	Người dùng đã có tài khoản trong hệ thống
Post-conditions	Hệ thống sẽ lưu trữ ngân sách trong cơ sở dữ liệu và hiển thị trong danh sách ngân sách của người dùng.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập vào mục "Ngân sách" từ trang chính hoặc mục cài đặt. 2. Người dùng nhấn vào tùy chọn "Tạo ngân sách mới". 3. Hệ thống hiển thị biểu mẫu yêu cầu người dùng nhập thông tin ngân sách, bao gồm: Danh mục và Số tiền 4. Người dùng điền đầy đủ thông tin và nhấn "Xác nhận". 5. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu ngân sách vào cơ sở dữ liệu. 6. Hệ thống thông báo rằng ngân sách đã được tạo thành công và hiển thị ngân sách vừa tạo trong danh sách ngân sách của người dùng.
Alternative flow	<ol style="list-style-type: none"> 1. Nếu người dùng nhập thiếu thông tin hoặc nhập thông tin không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. 2. Người dùng sẽ được yêu cầu sửa đổi thông tin và thực hiện lại các bước từ bước 3.
Extension point	

Bảng 1.16. Đặc tả Use case - Tạo ngân sách mới

Chương 3: Phân tích thiết kế hệ thống

3.1.3.17. Chính sửa ngân sách



Chương 3: Phân tích thiết kế hệ thống

Hình 3.17. Sequence - Chính sửa ngân sách

Name	Chỉnh sửa ngân sách
Brief description	Người dùng có thể chỉnh sửa các thông tin liên quan đến ngân sách mà họ đã tạo trước đó, bao gồm việc điều chỉnh tổng số tiền ngân sách, thay đổi danh mục.
Actor(s)	Người dùng
Pre-conditions	<ol style="list-style-type: none">Người dùng đã đăng nhập vào hệ thống.Người dùng đã có ít nhất một ngân sách được tạo trước đó.
Post-conditions	Hệ thống lưu lại các thay đổi và cập nhật thông tin ngân sách mới nhất.
Flow of events	<ol style="list-style-type: none">Người dùng truy cập vào mục "Ngân sách" và chọn một ngân sách cụ thể từ danh sách.Hệ thống hiển thị chi tiết của ngân sách.Người dùng nhấp vào nút "Chỉnh sửa".Hệ thống hiển thị giao diện chỉnh sửa ngân sách với các trường thông tin có thể thay đổi:Tổng số tiền ngân sách (người dùng có thể tăng/giảm số tiền).Danh mục chi tiêu/thu nhập áp dụng cho ngân sách.Người dùng thực hiện các thay đổi và nhấp "Xác nhận" để lưu lại thông tin.Hệ thống kiểm tra tính hợp lệ của thông tin đầu vào.Nếu mọi thông tin hợp lệ, hệ thống sẽ cập nhật ngân sách với các thay đổi mới.Người dùng được thông báo chỉnh sửa ngân sách thành công và quay lại màn hình chi tiết ngân sách.

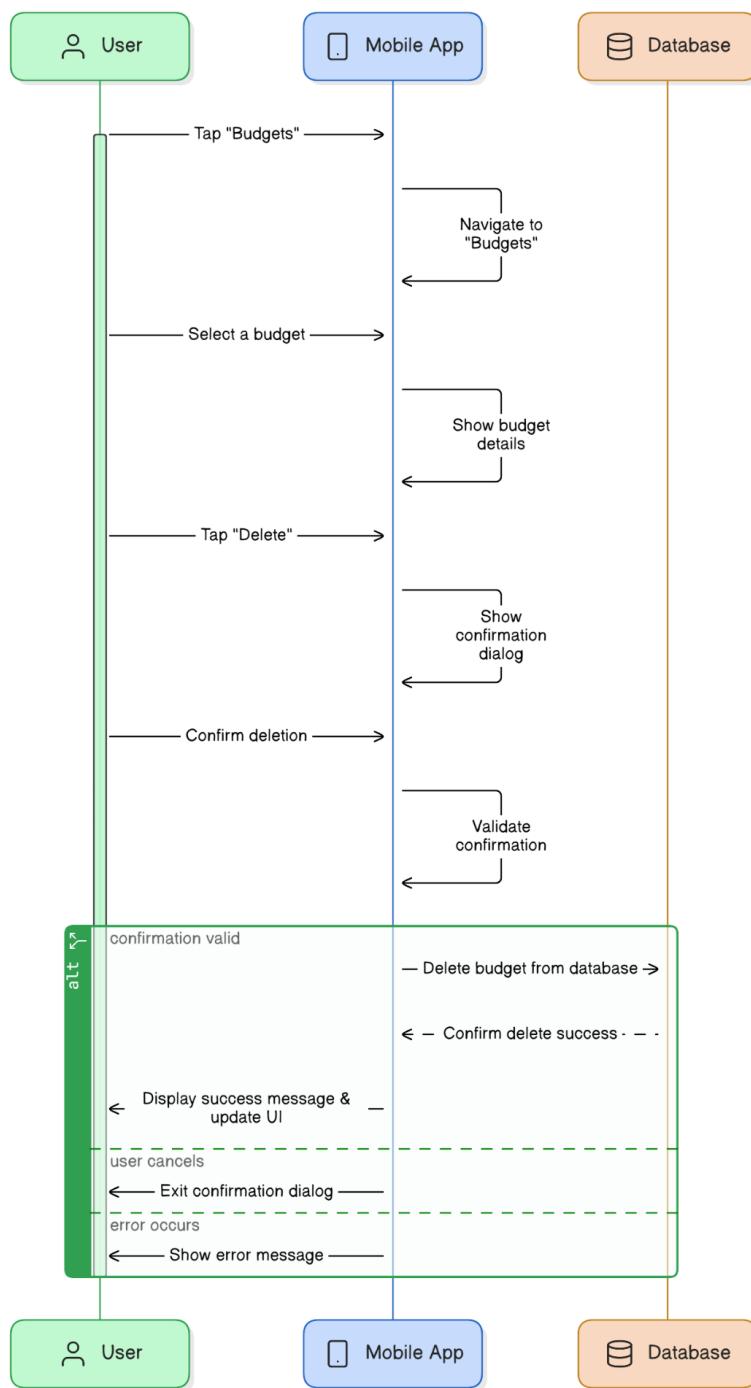
Chương 3: Phân tích thiết kế hệ thống

Basic flow	
Alternative flow	<ol style="list-style-type: none">Nếu người dùng nhập sai dữ liệu (ví dụ số tiền không hợp lệ hoặc thời gian không đúng), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng sửa lại.Người dùng có thể hủy bỏ quá trình chỉnh sửa và quay lại màn hình chi tiết mà không lưu bất kỳ thay đổi nào.
Extension point	

Bảng 1.17. Đặc tả Use case - Chính sửa ngân sách

Chương 3: Phân tích thiết kế hệ thống

3.1.3.18. Xóa ngân sách



Hình 3.18. Sequence - Xóa ngân sách

Chương 3: Phân tích thiết kế hệ thống

Name	Xóa ngân sách
Brief description	Người dùng có thể xóa một ngân sách mà họ đã tạo trước đó. Việc xóa ngân sách sẽ loại bỏ hoàn toàn các thông tin liên quan đến ngân sách đó nhưng không ảnh hưởng đến lịch sử giao dịch đã ghi nhận.
Actor(s)	Người dùng
Pre-conditions	<ol style="list-style-type: none"> Người dùng đã đăng nhập vào hệ thống. Người dùng đã có ít nhất một ngân sách được tạo trước đó.
Post-conditions	Hệ thống xóa toàn bộ thông tin liên quan đến ngân sách, không còn hiển thị trong danh sách ngân sách của người dùng.
Flow of events	
Basic flow	<ol style="list-style-type: none"> Người dùng truy cập vào mục "Ngân sách" và chọn ngân sách mà họ muốn xóa từ danh sách. Hệ thống hiển thị chi tiết của ngân sách. Người dùng nhấp vào nút "Xóa" ngân sách. Hệ thống hiển thị một thông báo xác nhận, yêu cầu người dùng xác nhận họ thực sự muốn xóa ngân sách này. Người dùng xác nhận xóa ngân sách. Hệ thống kiểm tra và xóa ngân sách khỏi cơ sở dữ liệu, cùng với tất cả thông tin liên quan. Người dùng nhận được thông báo ngân sách đã xóa thành công và quay trở lại danh sách ngân sách, nơi ngân sách vừa xóa không còn hiển thị.
Alternative flow	<ol style="list-style-type: none"> Nếu người dùng nhấn "Hủy" tại bước xác nhận, quá trình xóa sẽ bị hủy bỏ và người dùng quay lại màn hình chi tiết ngân sách mà không có bất kỳ thay đổi nào.

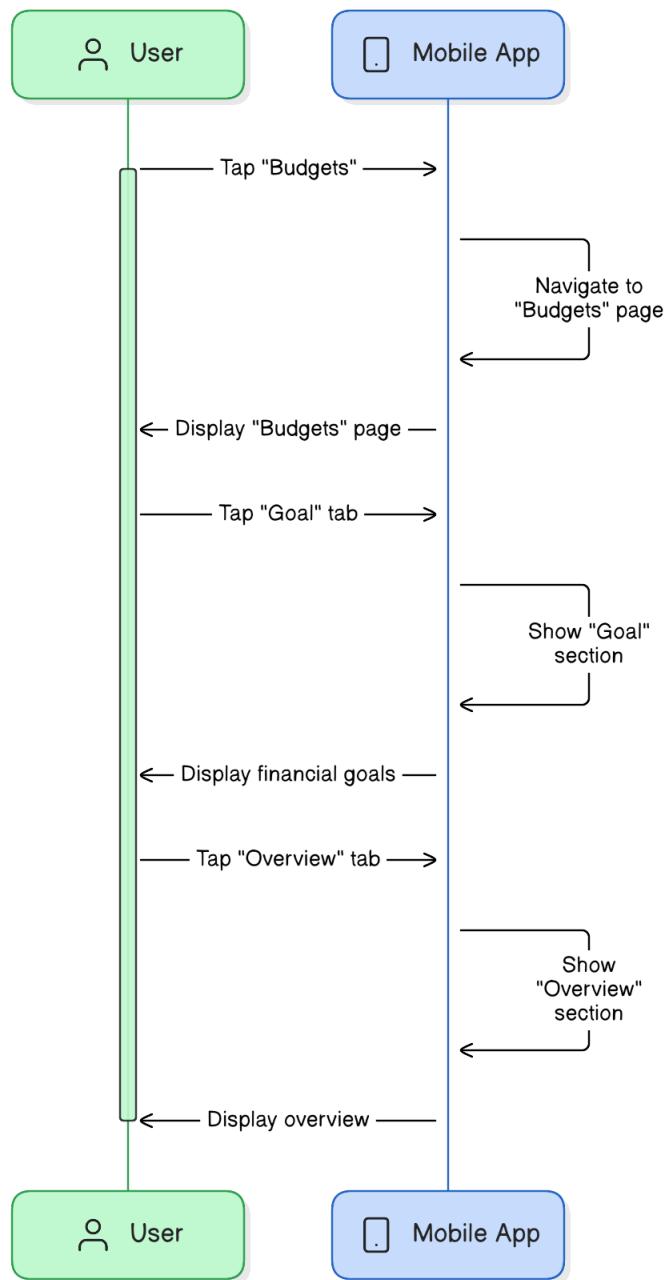
Chương 3: Phân tích thiết kế hệ thống

	2. Nếu hệ thống gặp sự cố trong quá trình xóa (ví dụ lỗi kết nối hoặc cơ sở dữ liệu), người dùng nhận được thông báo lỗi và có thể thử lại sau.
Extension point	

Bảng 1.18. Đặc tả Use case - Xóa ngân sách

Chương 3: Phân tích thiết kế hệ thống

3.1.3.19. Quản lý mục tiêu tài chính



Hình 3.19. Sequence - Quản lý mục tiêu tài chính

Name	Quản lý mục tiêu tài chính
------	----------------------------

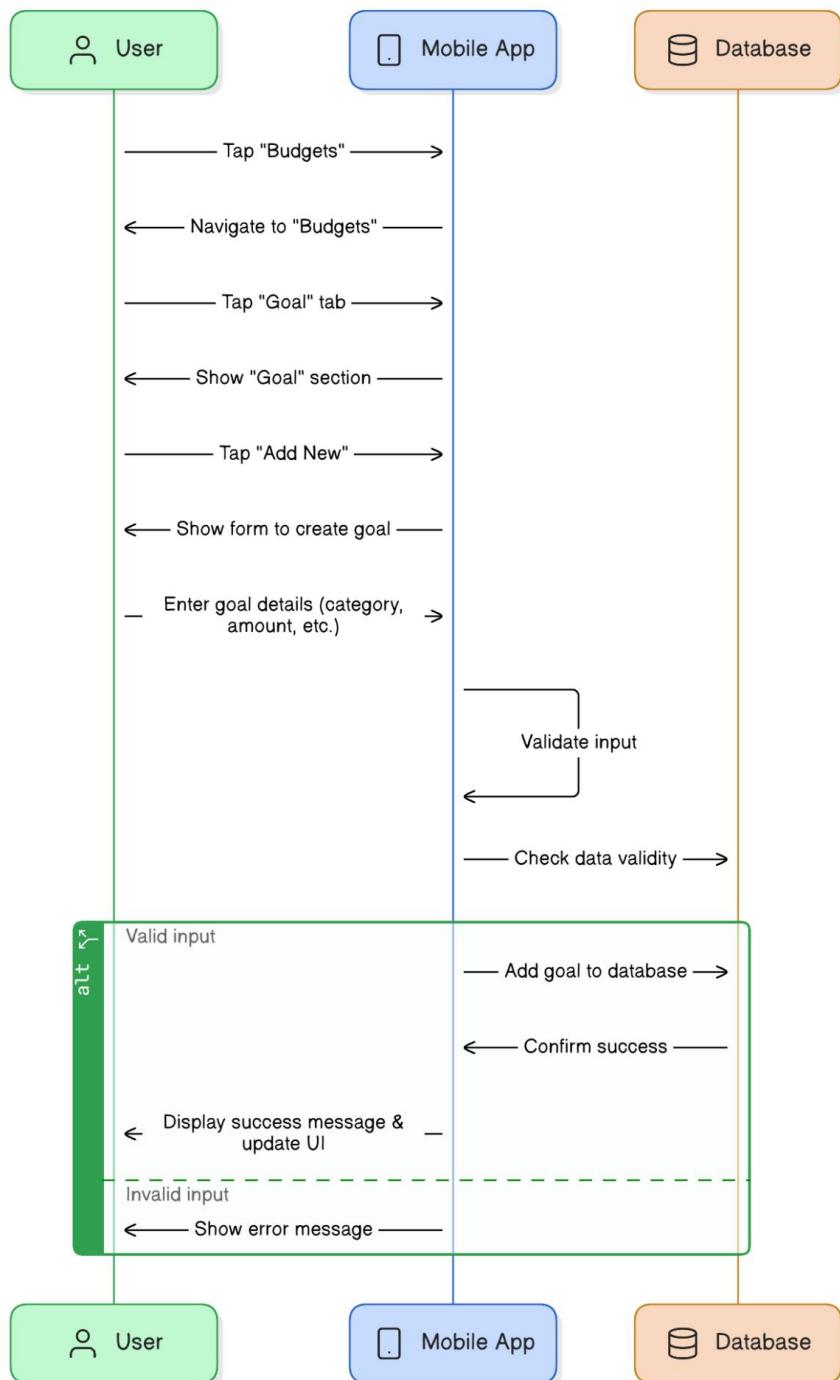
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng quản lý các mục tiêu tiết kiệm trong tương lai
Actor(s)	Người dùng
Pre-conditions	Người dùng đã có tài khoản trên hệ thống
Post-conditions	Người dùng mở được trang tab “Goal” trong trang “Budgets”
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng ở trang “Budgets”2. Người dùng nhấn vào tab “Goal”3. Hệ thống hiển thị “Goal” trên trang “Budget”
Alternative flow	
Extension point	

Bảng 1.19. Đặc tả Use case - Chi tiết mục tiêu

Chương 3: Phân tích thiết kế hệ thống

3.1.3.20. Tạo mục tiêu mới



Hình 3.20. Sequence - Tạo mục tiêu mới

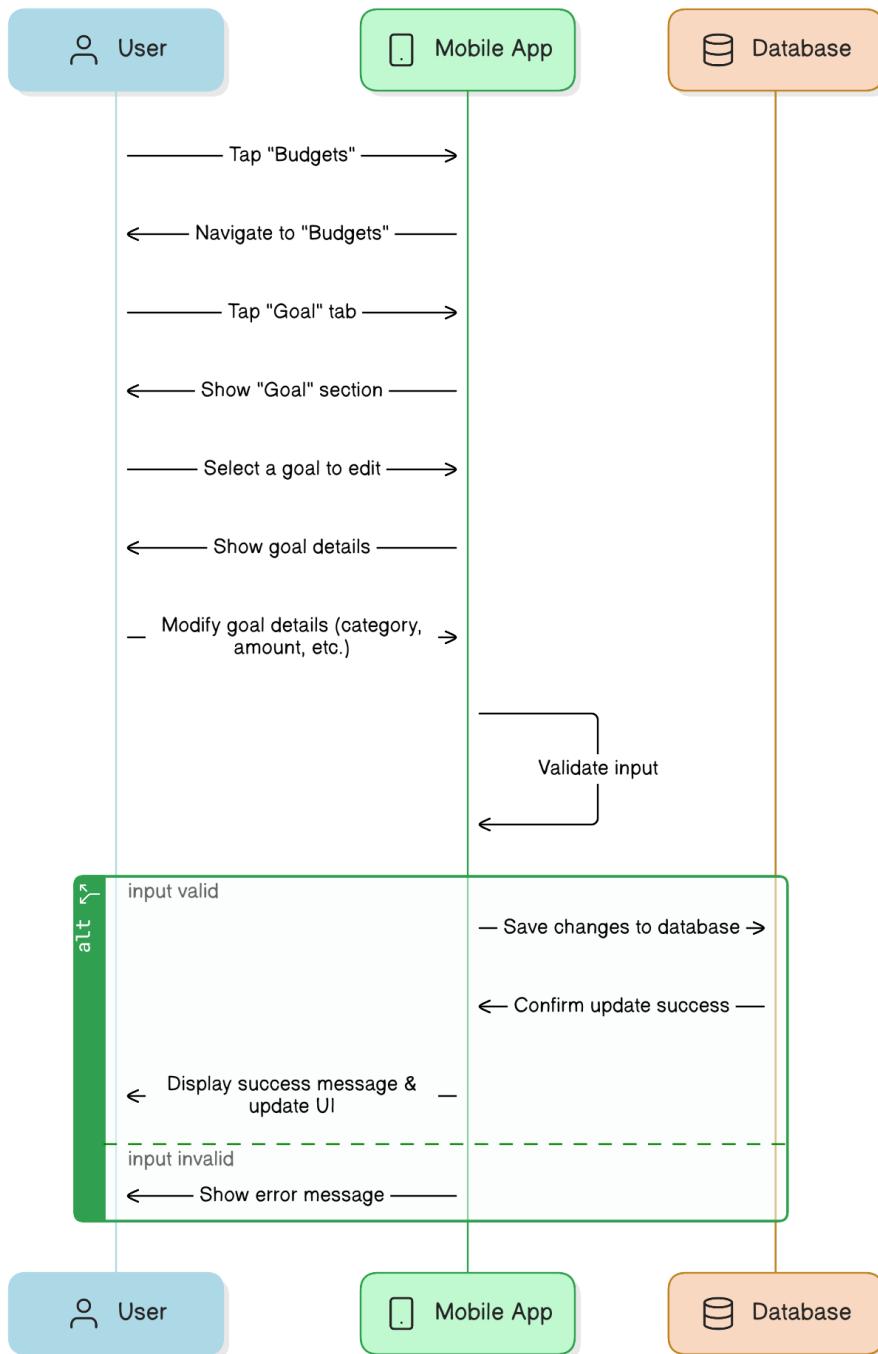
Chương 3: Phân tích thiết kế hệ thống

Name	Tạo mục tiêu mới.
Brief description	Người dùng tạo các mục tiêu tài mới chính trong phần Budget.
Actor(s)	User.
Pre-conditions	Actors đã đăng nhập thành công vào hệ thống.
Post-conditions	Mục tiêu được tạo thành công và hiển thị.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập mục tiêu tài chính trong phần Budget. 2. Hệ thống hiển thị các mục tiêu tài chính theo từng danh mục, như ăn uống, mua sắm, giải trí và các biểu đồ thống kê xu hướng chi tiêu của người dùng. 3. Người dùng Bấm "Add new". 4. Người dùng nhập và chọn thông tin của mục tiêu, sau đó bấm "Add". 5. Hệ thống thông báo thêm thành công và hiển thị mục tiêu tài chính vừa tạo.
Alternative flow	<p>Nếu người dùng nhập các ký tự lạ, định dạng sai thì sẽ không tạo được mục tiêu mới.</p> <ol style="list-style-type: none"> 1. Hệ thống sẽ thông báo thông tin mục tiêu không phù hợp. 2. Người dùng nhập lại thông tin mới. <p>Use case quay trở lại bước 4.</p>
Extension point	

Bảng 1.20. Đặc tả Use case - Tạo mục tiêu mới

Chương 3: Phân tích thiết kế hệ thống

3.1.3.21. Chính sửa mục tiêu



Hình 3.21. Sequence - Chính sửa mục tiêu

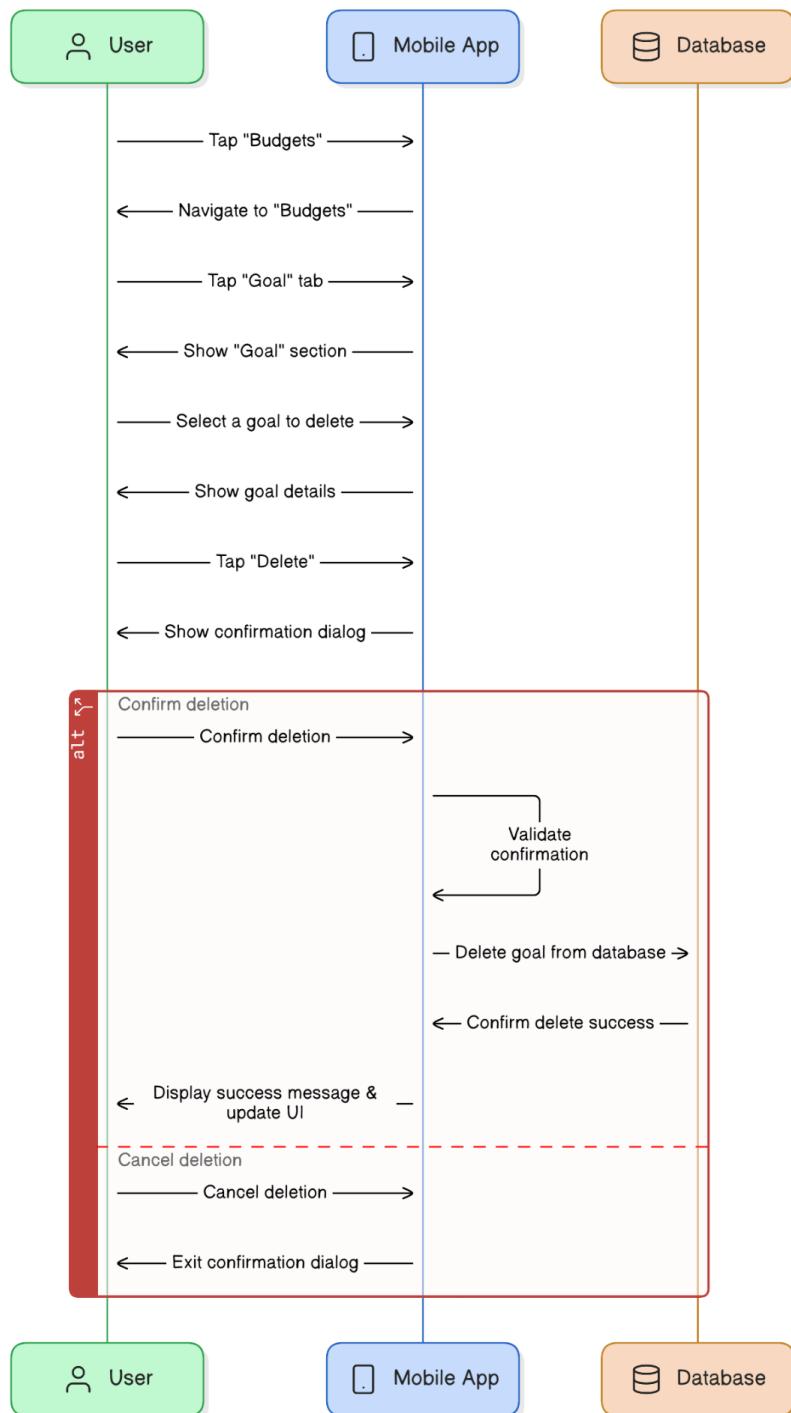
Chương 3: Phân tích thiết kế hệ thống

Name	Sửa mục tiêu tài chính.
Brief description	Người dùng sửa các mục tiêu tài chính trong phần Budget.
Actor(s)	User.
Pre-conditions	Actors đã có các mục tiêu tài chính.
Post-conditions	Cập nhật thành công mục tiêu tài chính.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập mục tiêu tài chính trong phần Budget. 2. Hệ thống hiển thị các mục tiêu tài chính theo từng danh mục, như ăn uống, mua sắm, giải trí và các biểu đồ thống kê xu hướng chi tiêu của người dùng. 3. Người dùng bấm vào mục tiêu cần sửa đổi. 4. Người dùng nhập và chọn thông tin của mục tiêu, sau đó bấm "Change". 5. Hệ thống thông báo sửa thành công, cập nhật và hiển thị mục tiêu tài chính vừa sửa đổi.
Alternative flow	<p>Nếu không có giao mục tiêu tài chính, hệ thống hiển thị thông báo không có mục tiêu nào, nếu người dùng nhập các ký tự lạ, định dạng sai thì sẽ không tạo được mục tiêu mới.</p> <ol style="list-style-type: none"> 1. Hệ thống sẽ thông báo thông tin mục tiêu không phù hợp. 2. Người dùng nhập lại thông tin mới <p>Use case quay trở lại bước 4.</p>
Extension point	

Bảng 1.21. Đặc tả Use case - Chính sửa mục tiêu

Chương 3: Phân tích thiết kế hệ thống

3.1.3.22. Xóa mục tiêu



Hình 3.22. Sequence - Xóa mục tiêu

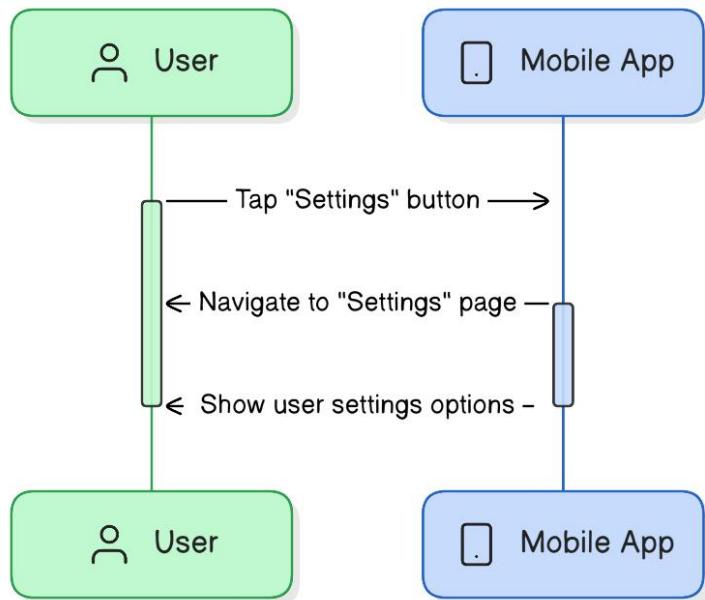
Chương 3: Phân tích thiết kế hệ thống

Name	Xóa mục tiêu tài chính.
Brief description	Người dùng sửa các mục tiêu tài chính trong phần Budget.
Actor(s)	User.
Pre-conditions	Actors đã có các mục tiêu tài chính.
Post-conditions	Cập nhật thành công mục tiêu tài chính.
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng truy cập mục tiêu tài chính trong phần Budget.2. Hệ thống hiển thị các mục tiêu tài chính theo từng danh mục, như ăn uống, mua sắm, giải trí và các biểu đồ thống kê xu hướng chi tiêu của người dùng.3. Người dùng bấm vào mục tiêu cần xóa.4. Người dùng chọn "Delete".5. Hệ thống thông báo xóa thành công và cập nhật.
Alternative flow	Nếu không có giao mục tiêu tài chính, hệ thống hiển thị thông báo không có mục tiêu nào.
Extension point	

Bảng 1.22. Đặc tả Use case - Xóa mục tiêu

Chương 3: Phân tích thiết kế hệ thống

3.1.3.23. Cài đặt ứng dụng



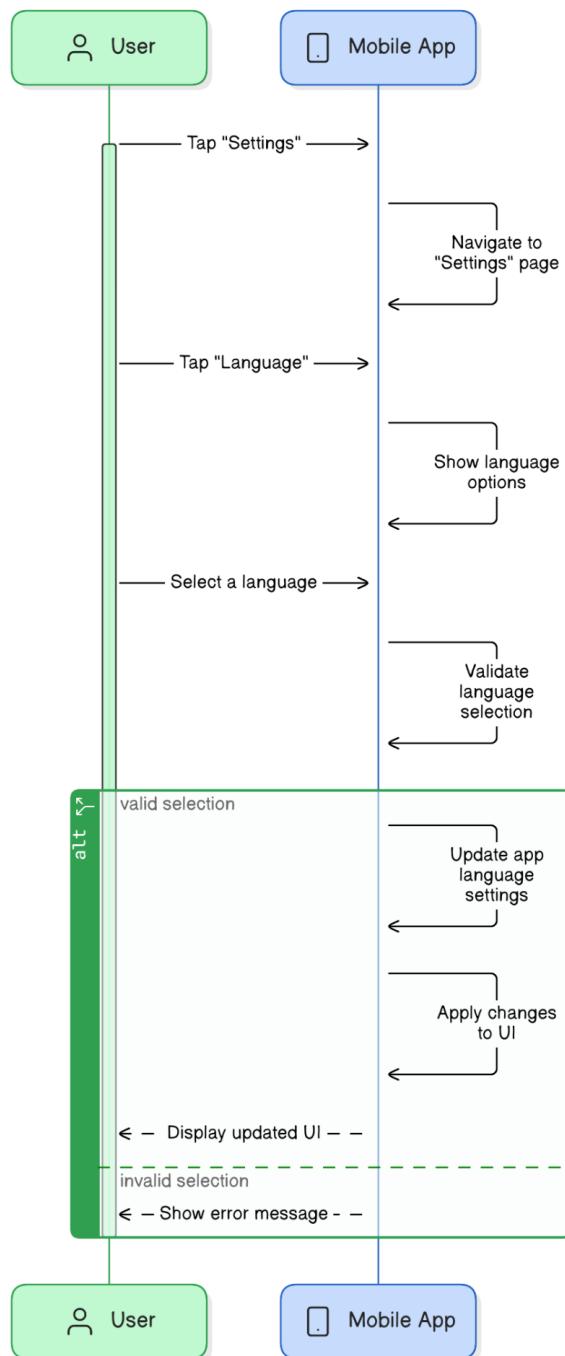
Hình 3.23. Sequence - Cài đặt ứng dụng

Name	Cài đặt ứng dụng
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	<ol style="list-style-type: none">1. Người dùng nhấn vào “Settings” trên thanh điều hướng2. Hệ thống chuyển hướng đến trang ‘Setting’
Alternative flow	
Extension point	

Chương 3: Phân tích thiết kế hệ thống

Bảng 1.23. Đặc tả Use case - Cài đặt ứng dụng

3.1.3.24. Cài đặt ngôn ngữ



Hình 3.24. Sequence - Cài đặt ngôn ngữ

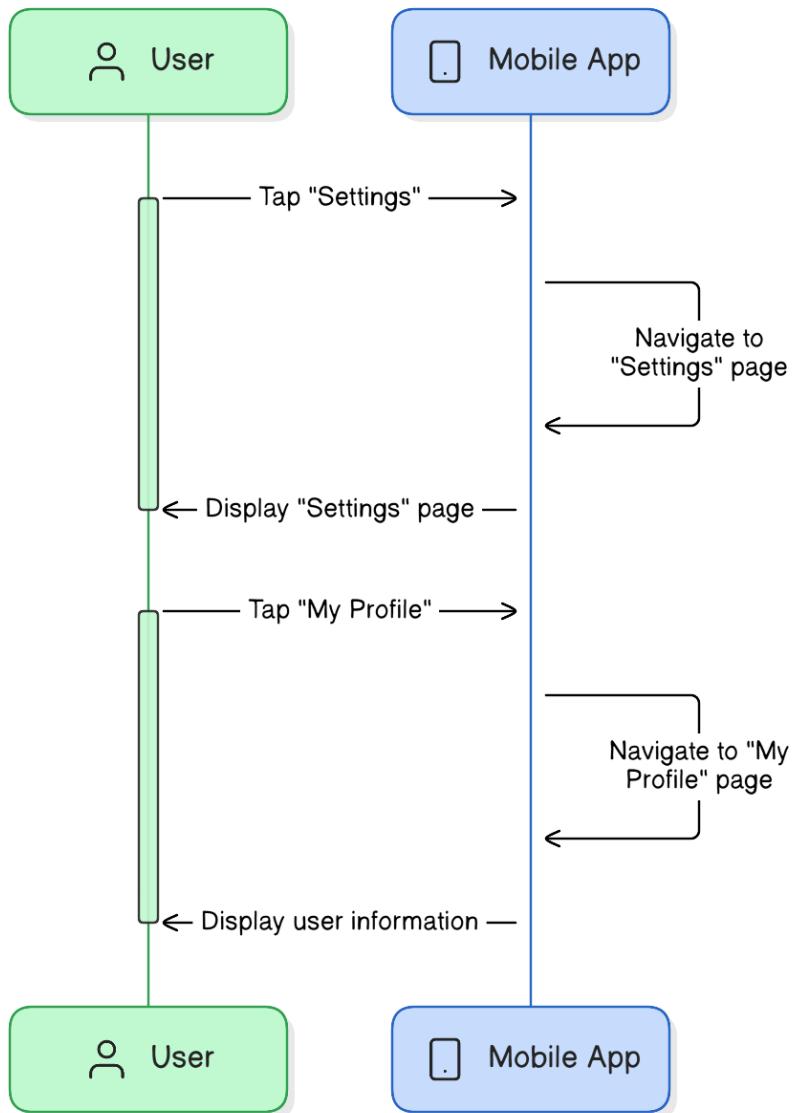
Chương 3: Phân tích thiết kế hệ thống

Name	Cài đặt ngôn ngữ
Brief description	Người dùng thay đổi ngôn ngữ giao diện của ứng dụng.
Actor(s)	Người dùng
Pre-conditions	Actors đã đăng nhập vào hệ thống.
Post-conditions	Ngôn ngữ của giao diện được thay đổi theo lựa chọn của người dùng.
Flow of events	
Basic flow	<ol style="list-style-type: none">Người dùng truy cập phần "Settings" và chọn mục "Language".Người dùng chọn ngôn ngữ mong muốn từ danh sách các ngôn ngữ có sẵn.Hệ thống cập nhật giao diện theo ngôn ngữ được chọn.
Alternative flow	Nếu ngôn ngữ không khả dụng hoặc có lỗi xảy ra khi thay đổi ngôn ngữ, hệ thống thông báo lỗi và yêu cầu người dùng thử lại.
Extension point	

Bảng 1.24. Đặc tả Use case - Cài đặt ngôn ngữ

Chương 3: Phân tích thiết kế hệ thống

3.1.3.25. Quản lý thông tin cá nhân



Hình 3.25. Sequence - Quản lý thông tin cá nhân

Name	Quản lý thông tin cá nhân
Brief description	Người dùng xem được thông tin cá nhân
Actor(s)	Người dùng

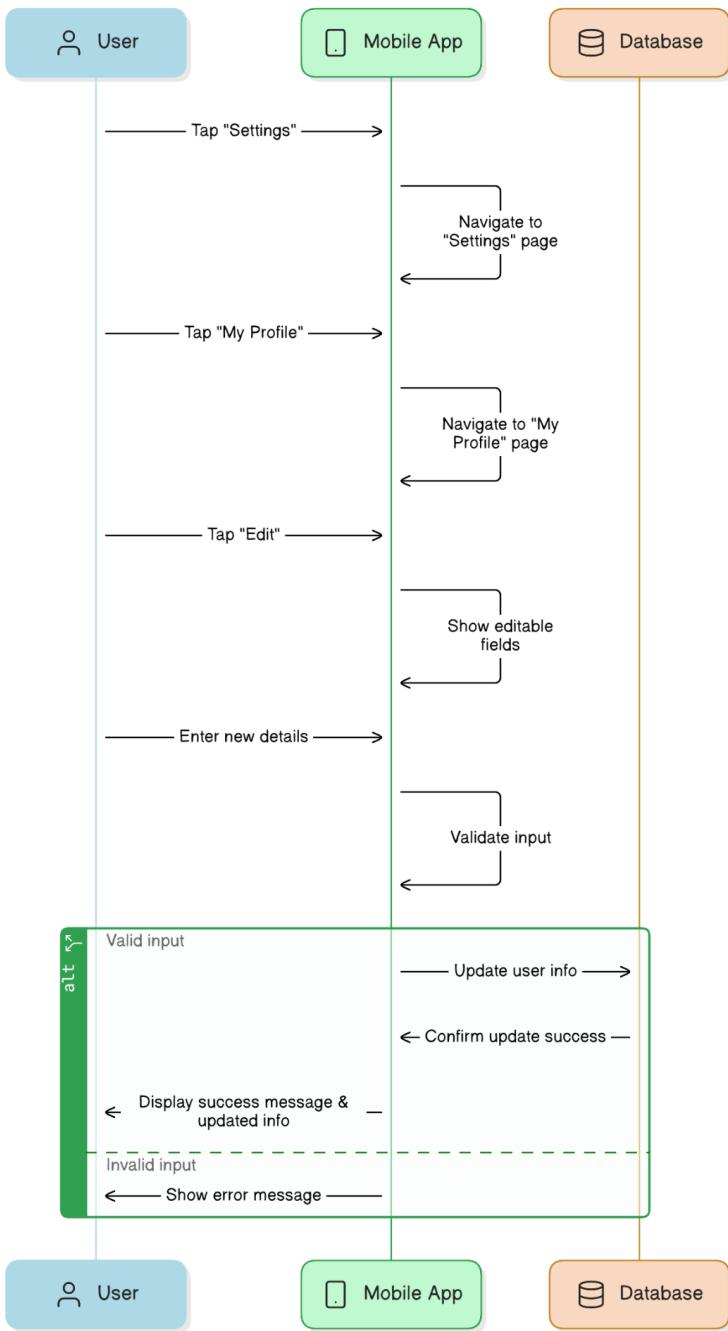
Chương 3: Phân tích thiết kế hệ thống

Pre-conditions	Người dùng đăng nhập vào ứng dụng
Post-conditions	Người dùng chuyển đến trang “My Profile”
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng đang ở trong trang “Settings”2. Người dùng nhấn vào trang “My Profile”3. Hệ thống chuyển hướng người dùng đến trang “My Profile”
Alternative flow	
Extension point	

Bảng 1.25. Đặc tả Use case - Quản lý thông tin cá nhân

Chương 3: Phân tích thiết kế hệ thống

3.1.3.26. Chính sửa thông tin cá nhân



Hình 3.26. Sequence - Chính sửa thông tin cá nhân

Name	Chỉnh sửa thông tin cá nhân.
------	------------------------------

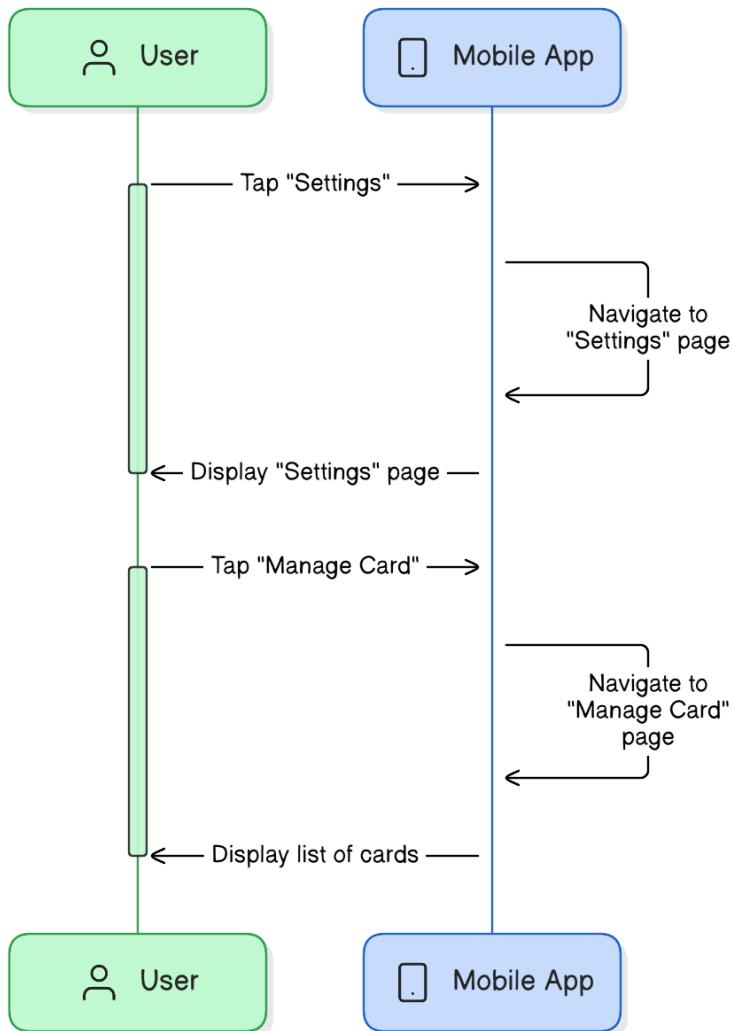
Chương 3: Phân tích thiết kế hệ thống

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 nhấp 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 1.26. Đặc tả Use case - Chính sửa thông tin cá nhân

Chương 3: Phân tích thiết kế hệ thống

3.1.3.27. Quản lý thẻ



Hình 3.27. 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

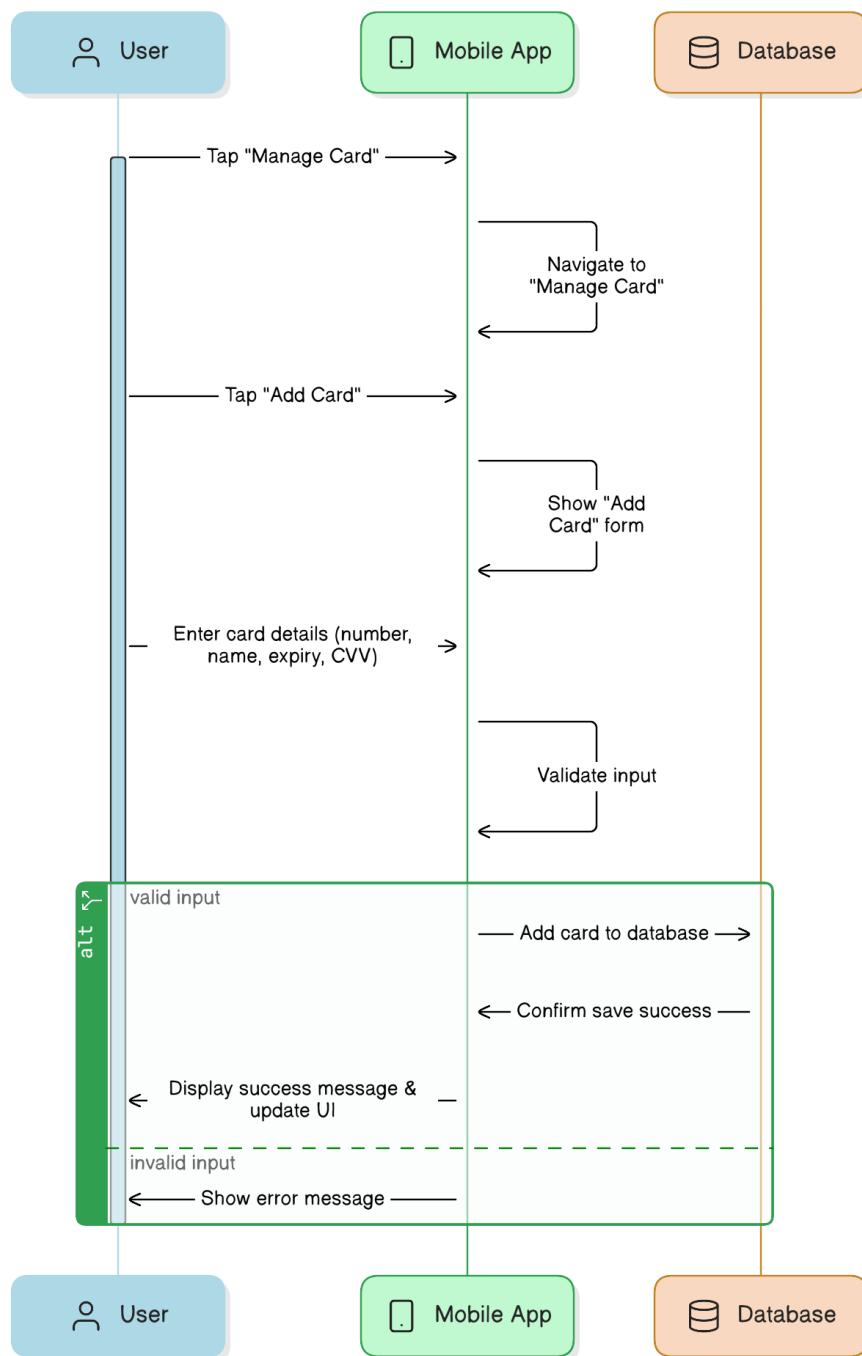
Chương 3: Phân tích thiết kế hệ thống

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 1.27. Đặc tả Use case - Quản lý thẻ

Chương 3: Phân tích thiết kế hệ thống

3.1.3.28. Thêm thẻ



Hình 3.28. Sequence - Thêm thẻ

Name	Thêm thẻ
------	----------

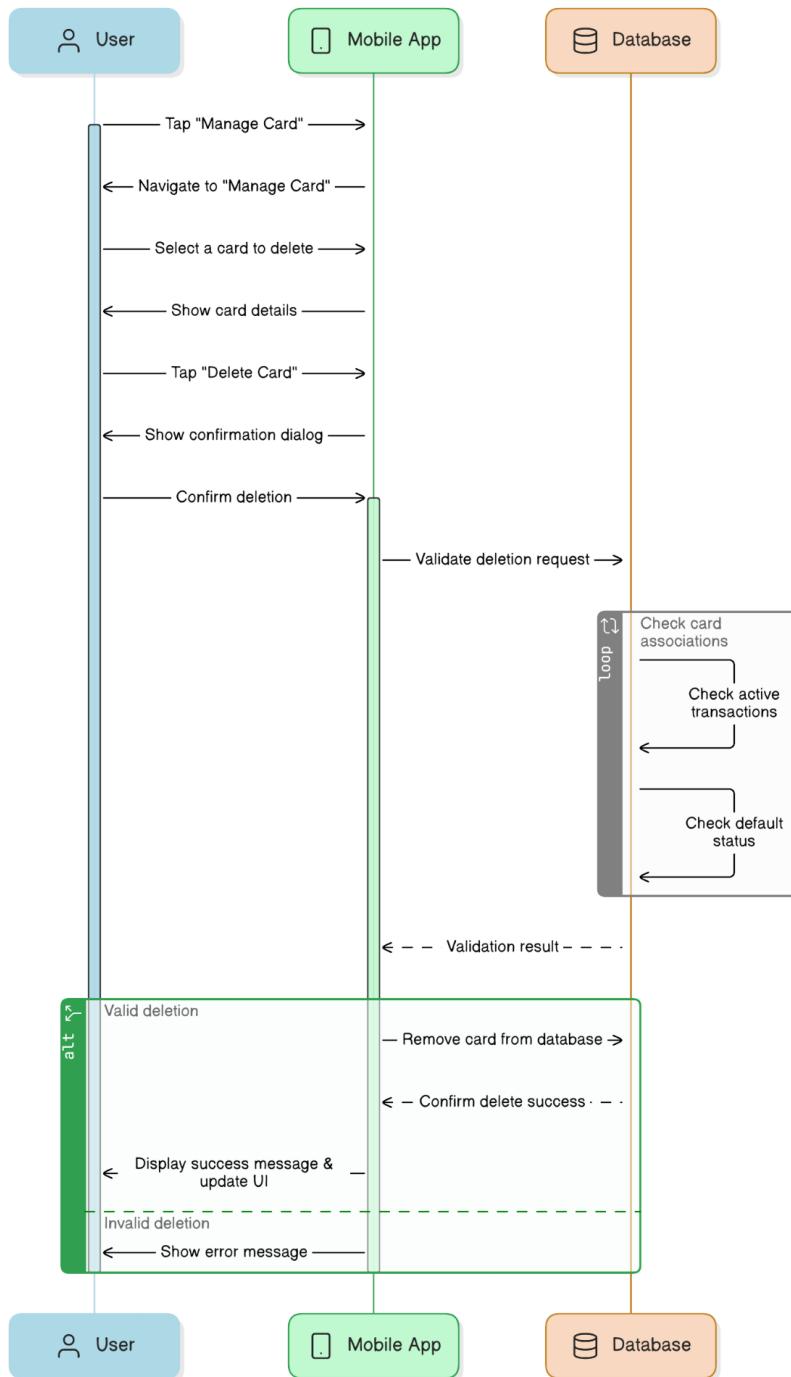
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng có thể thêm một thẻ mới vào hệ thống để sử dụng cho các giao dịch thanh toán và quản lý tài khoản tài chính của họ.
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống.
Post-conditions	Thông tin thẻ mới sẽ được lưu vào hệ thống và người dùng có thể sử dụng thẻ này cho các giao dịch tiếp theo.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang "Thêm thẻ" thông qua mục quản lý thẻ. 2. Người dùng điền các thông tin yêu cầu, bao gồm số thẻ, tên chủ thẻ, ngày hết hạn, CVV. 3. Hệ thống kiểm tra tính hợp lệ của thông tin. 4. Nếu thông tin hợp lệ, hệ thống lưu thẻ mới vào cơ sở dữ liệu. 5. Người dùng nhận thông báo xác nhận rằng thẻ đã được thêm thành công.
Alternative flow	<ol style="list-style-type: none"> 1. Nếu thông tin không hợp lệ (sai số thẻ, CVV không chính xác, ngày hết hạn không đúng,...), hệ thống sẽ hiển thị thông báo lỗi. 2. Người dùng cần sửa lại các thông tin và thực hiện lại bước xác nhận.
Extension point	

Bảng 1.28. Đặc tả Use case - Thêm thẻ

Chương 3: Phân tích thiết kế hệ thống

3.1.3.29. Xóa thẻ



Hình 3.29. Sequence - Xóa thẻ

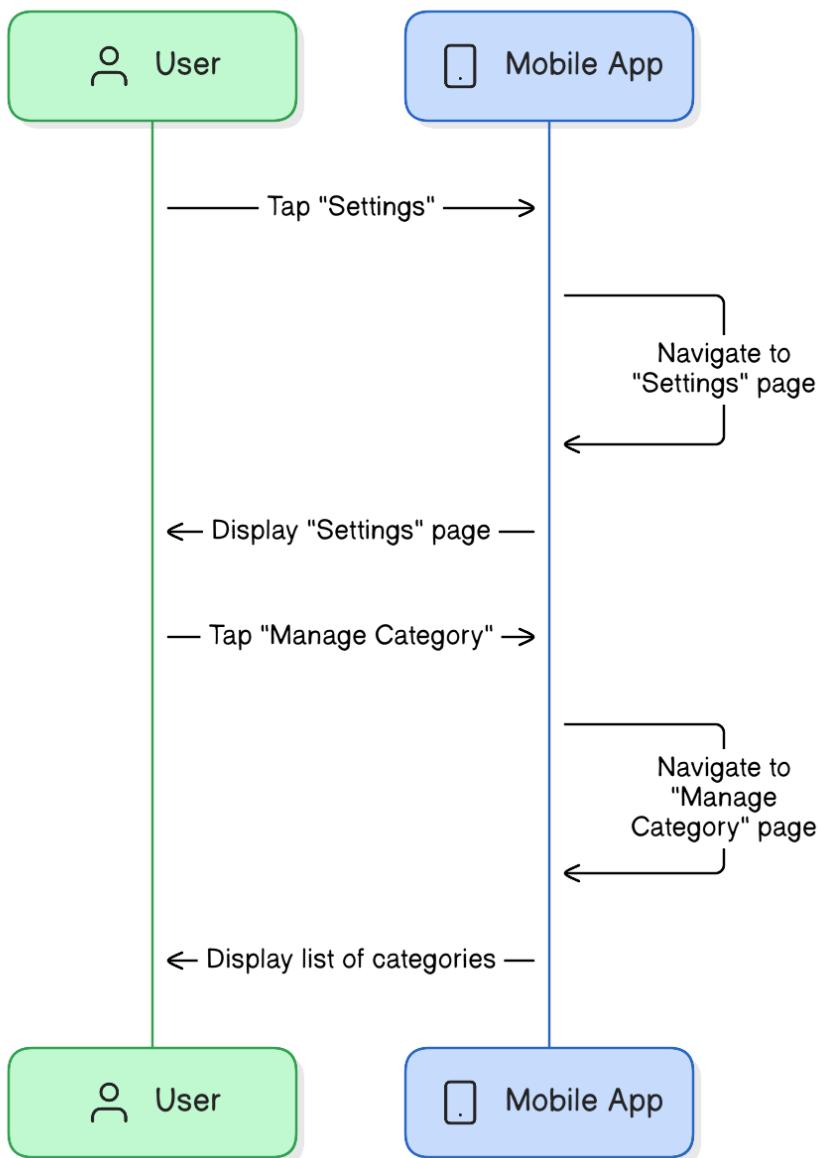
Chương 3: Phân tích thiết kế hệ thống

Name	Xóa thẻ.
Brief description	Người dùng có thể xóa thẻ đã thêm khỏi hệ thống nếu không còn nhu cầu sử dụng hoặc muốn quản lý các thẻ tài chính một cách dễ dàng hơn.
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống.
Post-conditions	Thẻ sẽ được gỡ khỏi tài khoản của người dùng trong hệ thống và không thể sử dụng cho các giao dịch tiếp theo.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang quản lý thẻ từ mục cài đặt hoặc thông qua giao diện quản lý tài khoản. 2. Người dùng chọn thẻ muốn xóa và nhấn nút "Xóa thẻ". 3. Hệ thống yêu cầu người dùng xác nhận hành động xóa thẻ. 4. Người dùng xác nhận xóa. 5. Người dùng nhận được thông báo rằng thẻ đã được xóa thành công.
Alternative flow	<ol style="list-style-type: none"> 1. Nếu thẻ không thể xóa do có các giao dịch đang xử lý hoặc thẻ là thẻ chính cho các giao dịch hiện tại, hệ thống sẽ hiển thị thông báo lỗi. 2. Người dùng có thể chọn một thẻ khác hoặc thực hiện các bước xử lý để đảm bảo thẻ được phép xóa (ví dụ: đổi thẻ mặc định).
Extension point	

Bảng 1.29. Đặc tả Use case - Xóa thẻ

Chương 3: Phân tích thiết kế hệ thống

3.1.3.30. Quản lý danh mục



Hình 3.30. Sequence - Quản lý danh mục

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

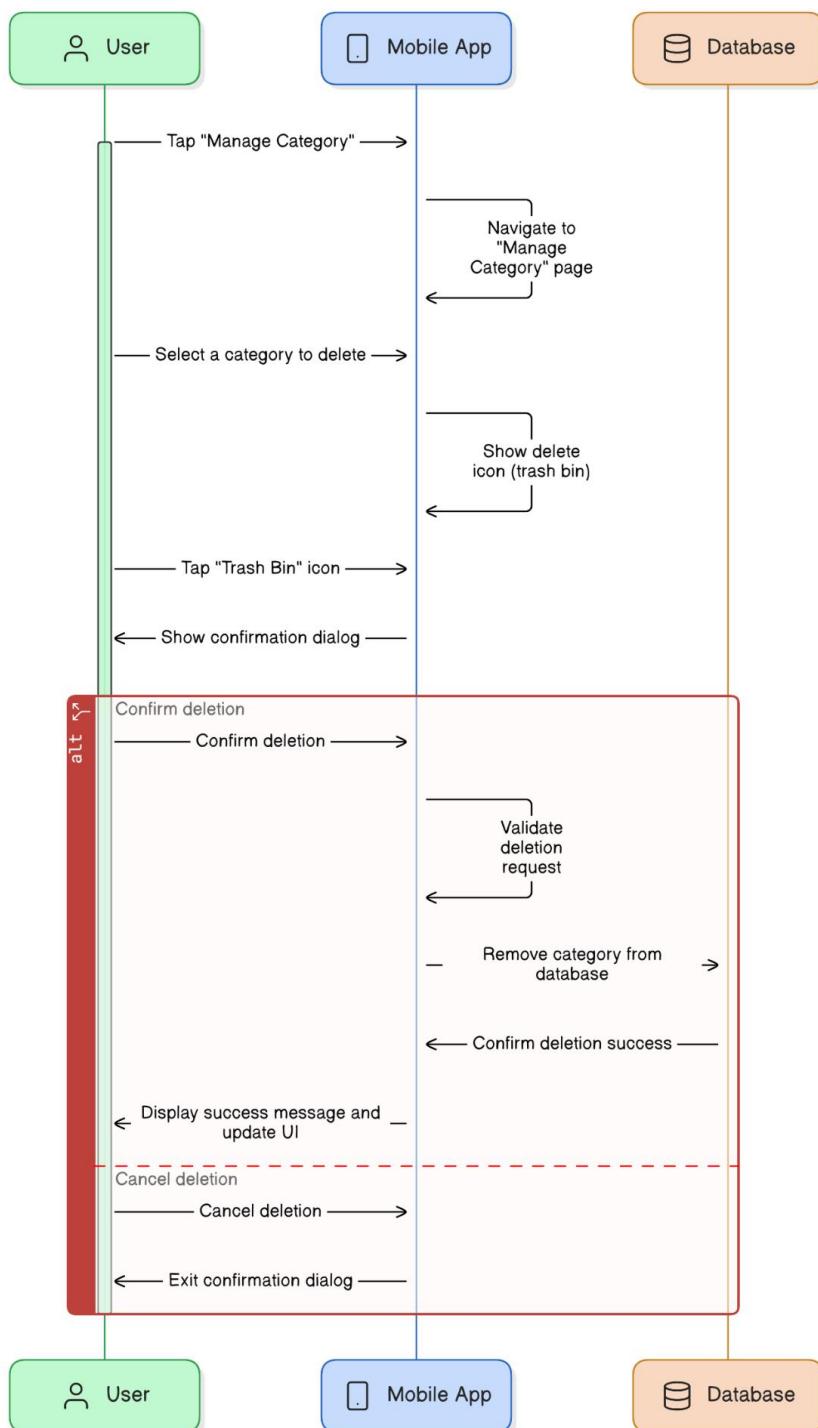
Chương 3: Phân tích thiết kế hệ thống

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

Bảng 1.30. Đặc tả Use case - Quản lý danh mục

Chương 3: Phân tích thiết kế hệ thống

3.1.3.31. Xóa danh mục



Hình 3.31. Sequence - Xóa danh mục

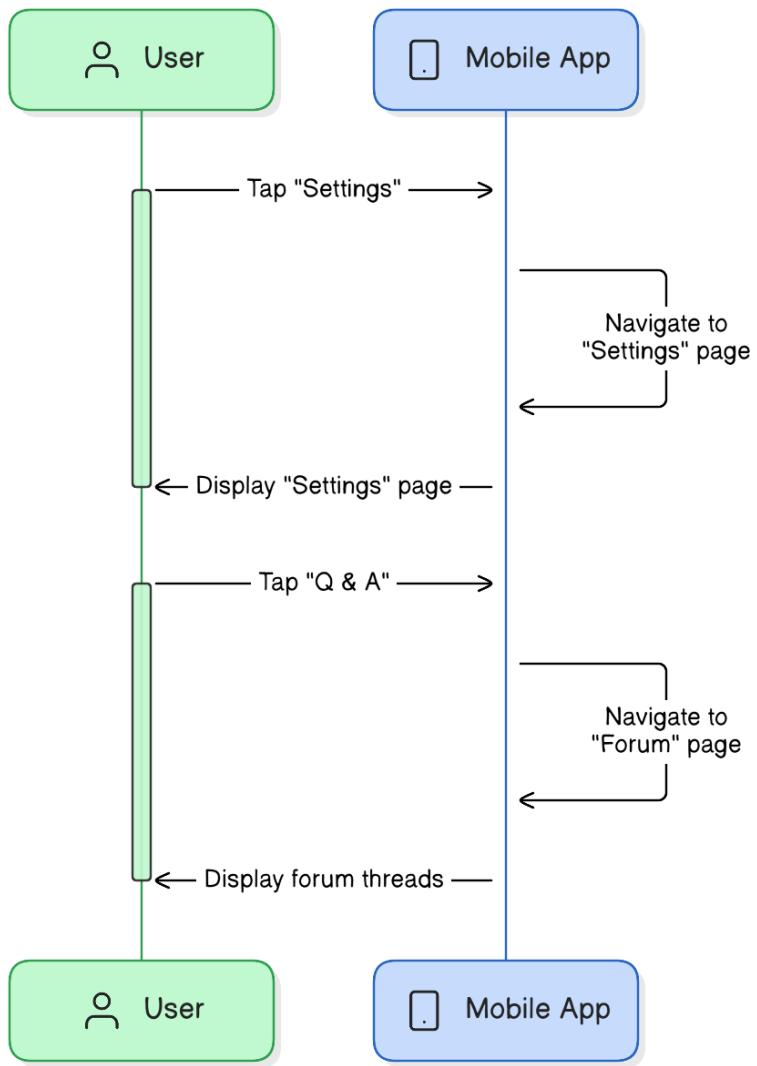
Chương 3: Phân tích thiết kế hệ thống

Name	Xóa danh mục
Brief description	Người dùng xóa đi danh mục mình muốn xóa
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng xóa đi danh mục cần xóa
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Người dùng mở trang “Manage Category” 2. Người dùng nhấn vào biểu tượng thùng rác tương ứng của danh mục 3. Hệ thống hiện pop-up yêu cầu xác nhận xóa 4. Người dùng nhấn “Cancel” để huỷ yêu cầu xóa, hệ thống sẽ quay lại trang “Manage Category” hoặc nhấn “Confirm” để xác nhận xóa 5. Hệ thống xóa danh mục đó khỏi cơ sở dữ liệu
Alternative flow	
Extension point	

Bảng 1.31. ĐẶC TẢ USE CASE - XÓA DANH MỤC

Chương 3: Phân tích thiết kế hệ thống

3.1.3.32. Quản lý Forum



Hình 3.32. Sequence - Quản lý Forum

Name	Quản lý Forum
Brief description	Người dùng có thể trao đổi với các người dùng khác trên diễn đàn
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống

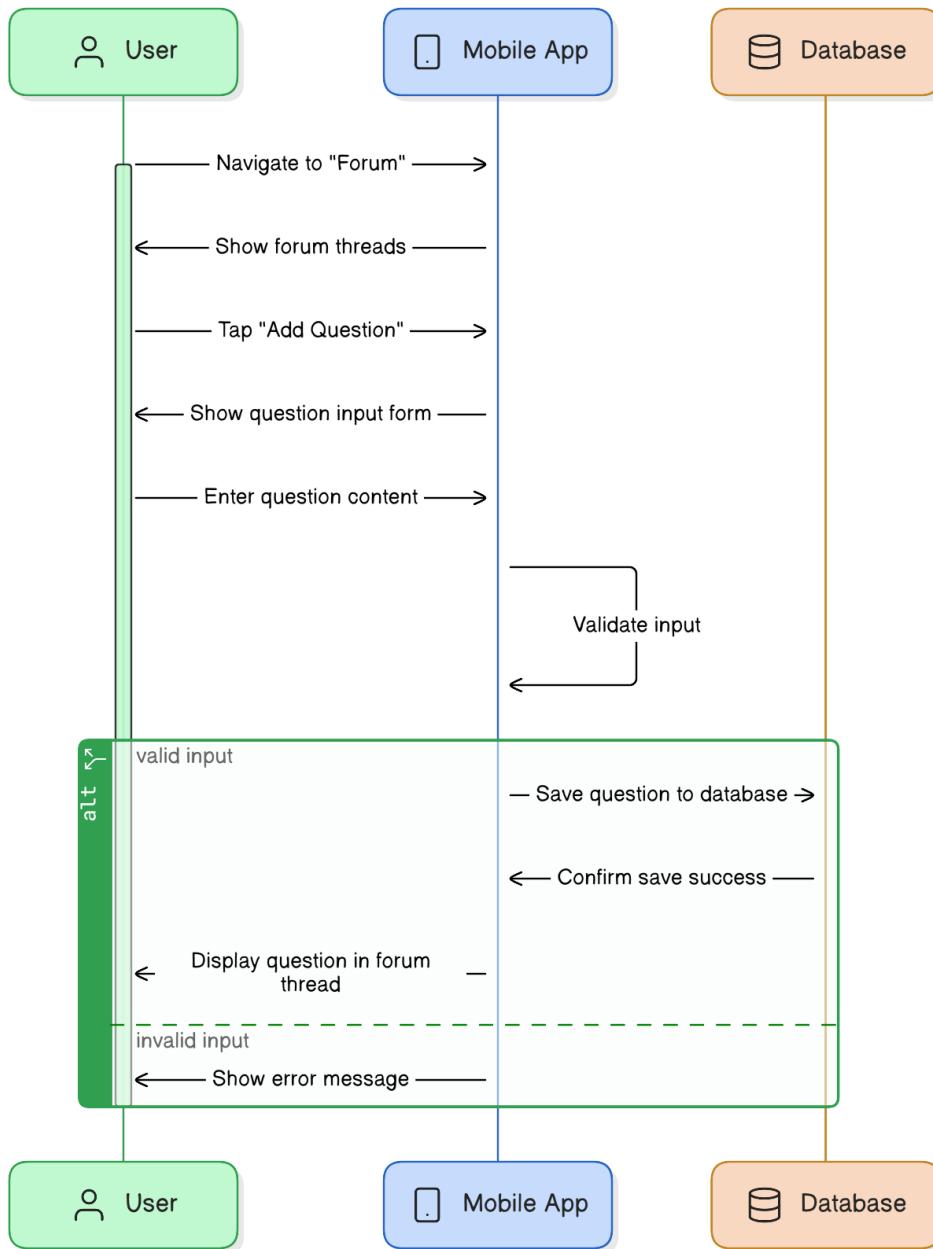
Chương 3: Phân tích thiết kế hệ thống

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 “Settings”2. Người dùng nhấp vào “Q & A”3. Hệ thống chuyển tiếp đến trang “Forum”
Alternative flow	
Extension point	

Bảng 3.32. Đặc tả Use case - Quản lý Forum

Chương 3: Phân tích thiết kế hệ thống

3.1.3.33. Đặt câu hỏi



Hình 3.33. Sequence - Đặt câu hỏi

Name	Đặt câu hỏi.
Brief description	Người dùng đặt câu hỏi của mình lên diễn đàn

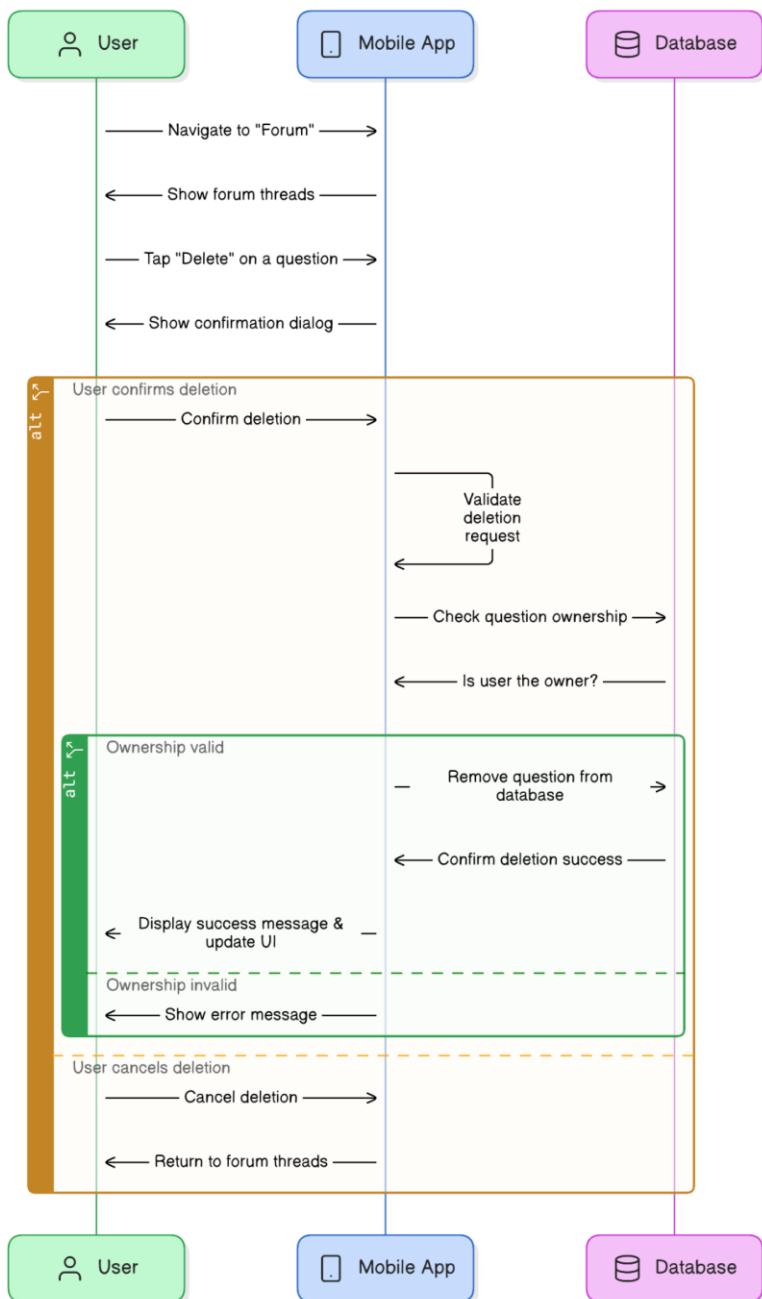
Chương 3: Phân tích thiết kế hệ thống

Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng đăng câu hỏi của mình lên diễn đàn
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng đang ở “Forum”2. Người dùng nhập câu hỏi của mình vào ô nhập3. Người dùng nhấn “Add Question”4. Hệ thống lưu câu hỏi vào cơ sở dữ liệu và hiển thị lên trong trang “Forum”
Alternative flow	
Extension point	

Bảng 1.33. Đặc tả Use case - Đặt câu hỏi

Chương 3: Phân tích thiết kế hệ thống

3.1.3.34. Xóa câu hỏi



Hình 3.34. Sequence - Xóa câu hỏi

Name	Xóa câu hỏi.
------	--------------

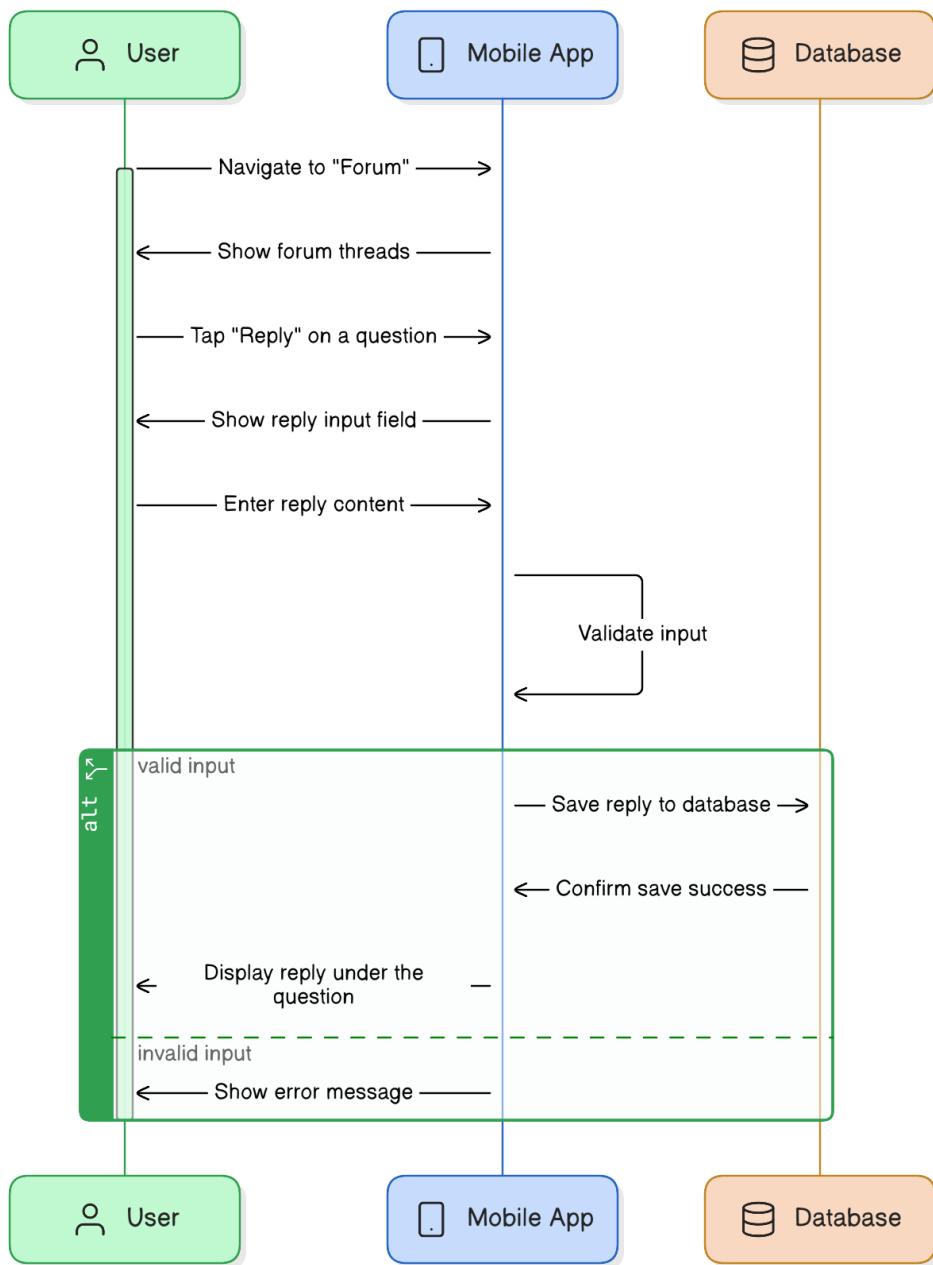
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng xóa đi câu hỏi do mình đăng
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng xóa đi câu hỏi do mình đăng
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng nhấp vào nút “Delete” của câu hỏi2. Hệ thống sẽ hiện thông báo để xác nhận xóa3. Người dùng nhấn “Cancel” để huỷ yêu cầu xóa, hệ thống sẽ quay lại trang “Forum” hoặc nhấn “Confirm” để xác nhận xóa
Alternative flow	Người dùng xóa câu hỏi do người khác đăng, hệ thống sẽ báo lỗi: “You can only delete your own questions”
Extension point	

Bảng 1.34. Đặc tả Use case - Xóa câu hỏi

Chương 3: Phân tích thiết kế hệ thống

3.1.3.35. Trả lời câu hỏi



Hình 3.35. Sequence - Trả lời câu hỏi

Name	Trả lời câu hỏi.
------	------------------

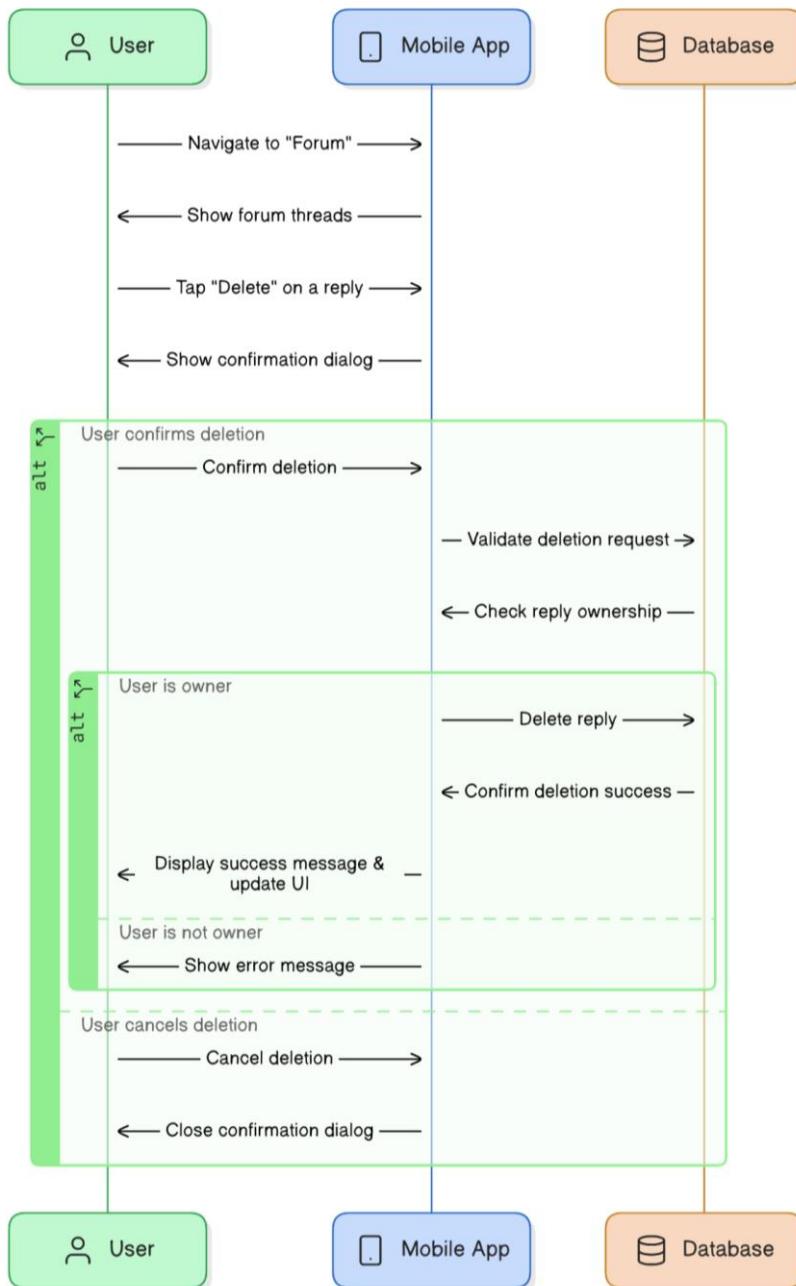
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng trả lời câu hỏi
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng đăng câu trả lời của mình ở dưới câu hỏi
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng nhấp vào nút “Reply”2. Người dùng nhập câu trả lời vào ô nhập3. Người dùng nhấp “Add Reply”4. Hệ thống sẽ lưu câu trả lời vào cơ sở dữ liệu và hiển thị lên màn hình trang “Forum”
Alternative flow	
Extension point	

Bảng 1.35. Đặc tả Use case - Trả lời câu hỏi

Chương 3: Phân tích thiết kế hệ thống

3.1.3.36. Xóa câu trả lời



Hình 3.36. Sequence - Xóa câu trả lời

Name	Xóa câu trả lời.
------	------------------

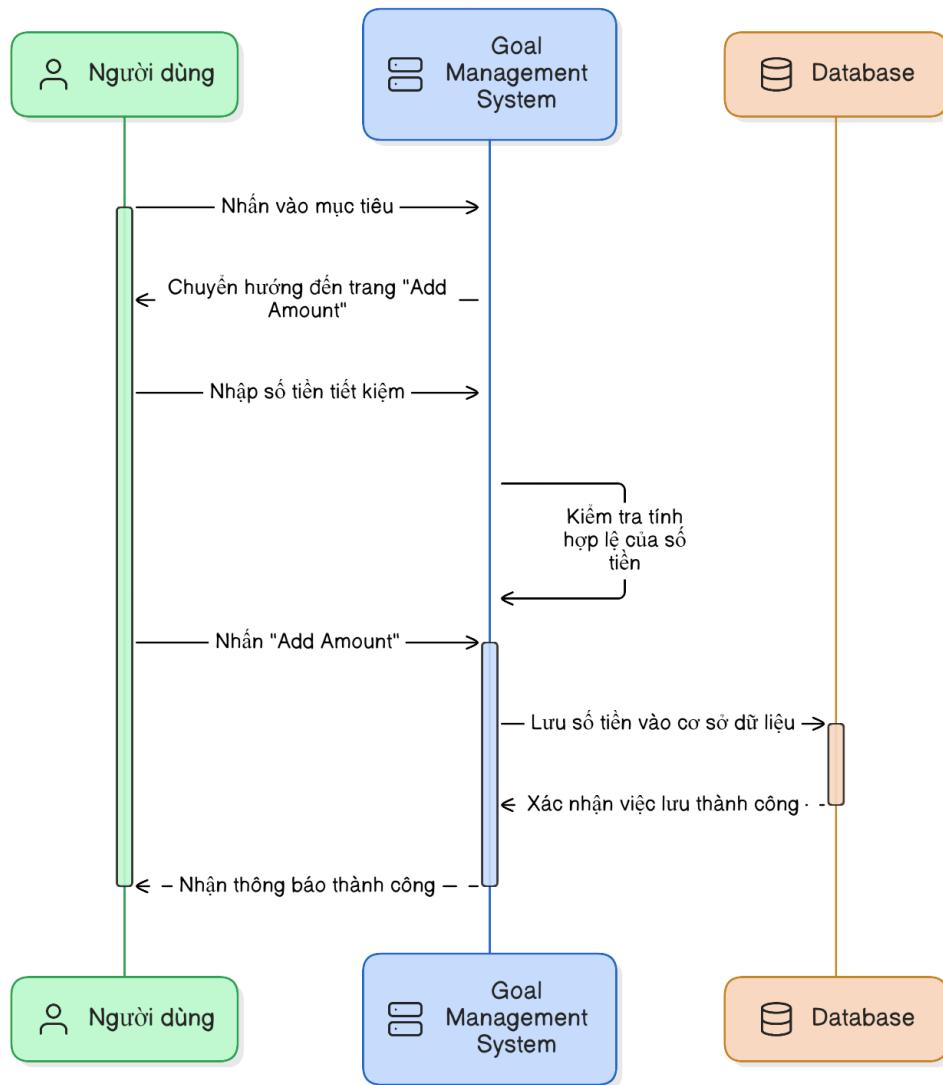
Chương 3: Phân tích thiết kế hệ thống

Brief description	Người dùng xóa đi câu trả lời do mình đăng
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng xóa đi câu trả lời do mình đăng
Flow of events	
Basic flow	<ol style="list-style-type: none">4. Người dùng nhấn vào nút “Delete” của câu hỏi5. Hệ thống sẽ hiện thông báo để xác nhận xóa6. Người dùng nhấn “Cancel” để huỷ yêu cầu xóa, hệ thống sẽ quay lại trang “Forum” hoặc nhấn “Confirm” để xác nhận xóa
Alternative flow	Người dùng xóa câu hỏi do người khác đăng, hệ thống sẽ báo lỗi: “You can only delete your own replies”
Extension point	

Bảng 1.36. Đặc tả Use case - Xóa câu trả lời

Chương 3: Phân tích thiết kế hệ thống

3.1.3.37. Thêm tiền vào mục tiêu



Hình 3.37. Sequence - Thêm tiền vào mục tiêu

Name	Thêm tiền vào mục tiêu
Brief description	Người dùng thêm số tiền tiết kiệm vào mục tiêu
Actor(s)	Người dùng
Pre-conditions	Người dùng đã đăng nhập vào hệ thống

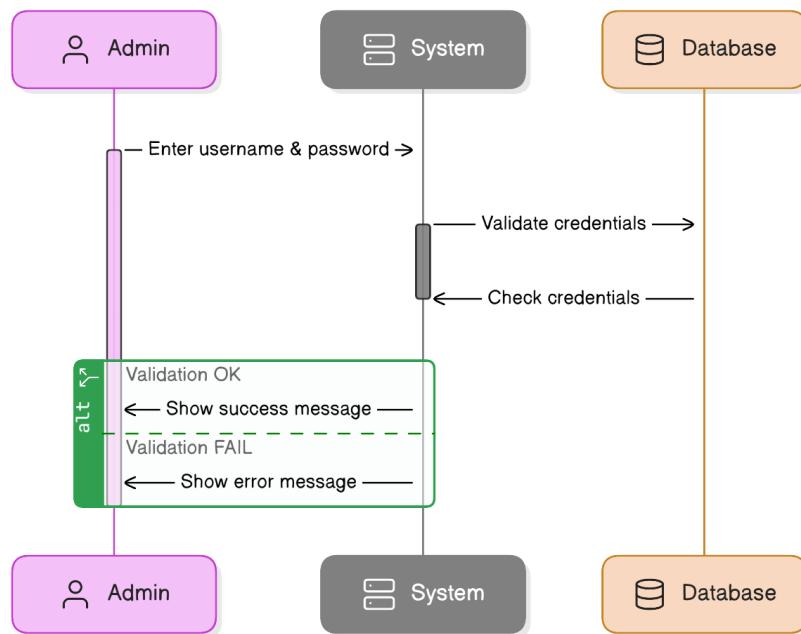
Chương 3: Phân tích thiết kế hệ thống

Post-conditions	Người dùng mở trang quản lý mục tiêu
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Người dùng ở trong trang “Goal”2. Người dùng nhấn vào 1 thẻ mục tiêu3. Hệ thống điều hướng đến trang “Add Amount”4. Người dùng nhập số tiền muốn tiết kiệm vào mục tiêu5. Người dùng nhấn “Add Amount”6. Hệ thống lưu số tiền tiết kiệm được vào cơ sở dữ liệu.
Alternative flow	
Extension point	

Bảng 1.37. Đặc tả Use case - Thêm tiền vào mục tiêu

Chương 3: Phân tích thiết kế hệ thống

3.1.3.38. Đăng nhập Admin



Hình 3.38. Sequence - Đăng nhập Admin

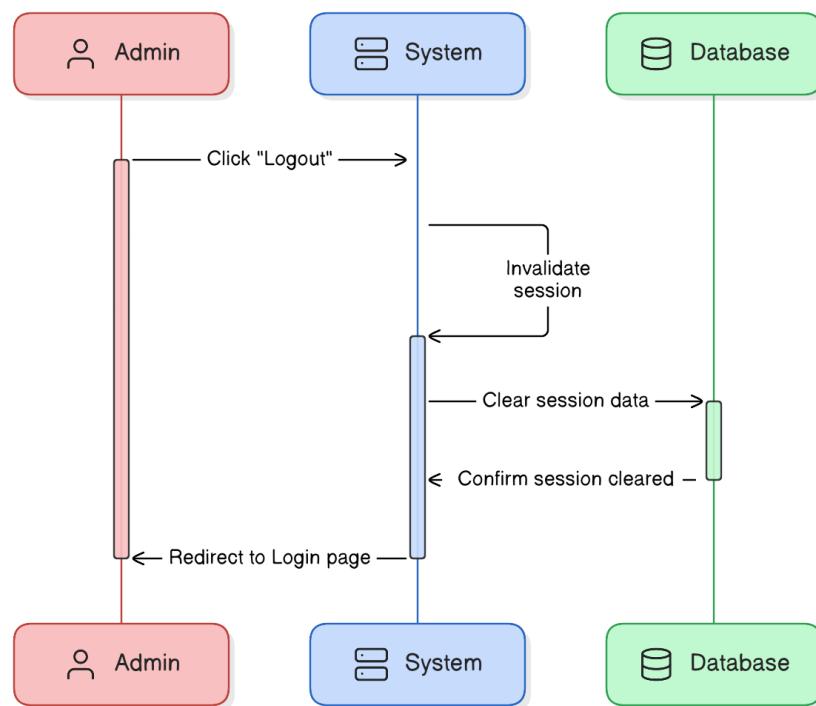
Name	Đăng nhập Admin
Brief description	Quản lý đăng nhập vào hệ thống
Actor(s)	Quản lý
Pre-conditions	Sử dụng tài khoản quản lý
Post-conditions	Quản lý vào trang quản lý
Flow of events	Thông báo lỗi đăng nhập và yêu cầu đăng nhập lại.
Basic flow	Use case được kích hoạt khi người dùng cần đăng nhập vào hệ thống hoặc xác thực danh tính để sử dụng các chức năng của hệ thống: 4. Người dùng nhập tài khoản và mật khẩu vào các ô input và bấm "Login".

Chương 3: Phân tích thiết kế hệ thống

	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 1.38. Đặc tả Use case - Đăng nhập Admin

3.1.3.39. Đăng xuất Admin



Hình 3.39. Sequence - Đăng xuất Admin

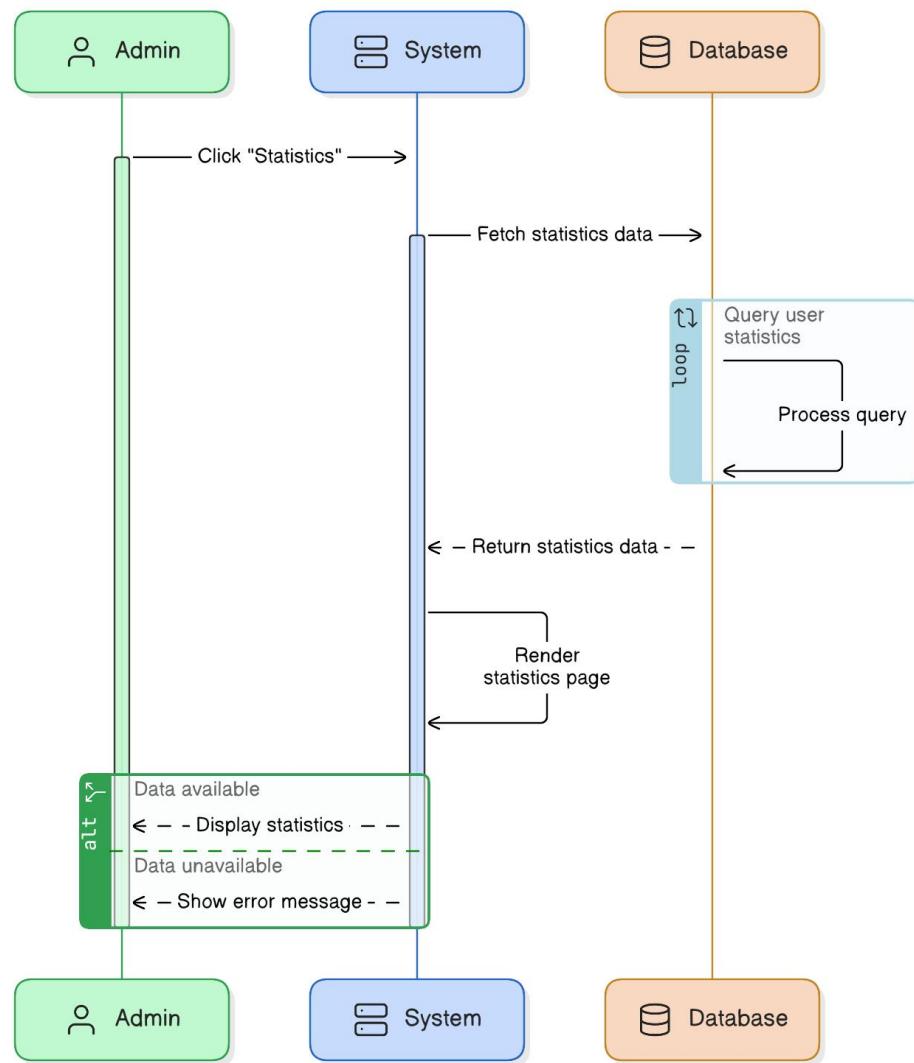
Name	Đăng xuất Admin
Brief description	Đăng xuất tài khoản quản lý khỏi hệ thống
Actor(s)	Quản lý

Chương 3: Phân tích thiết kế hệ thống

Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.
Post-conditions	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 1.39. Đặc tả Use case - Đăng xuất Admin

3.1.3.40. Thống kê người dùng



Hình 3.40. Sequence - Thống kê người dùng

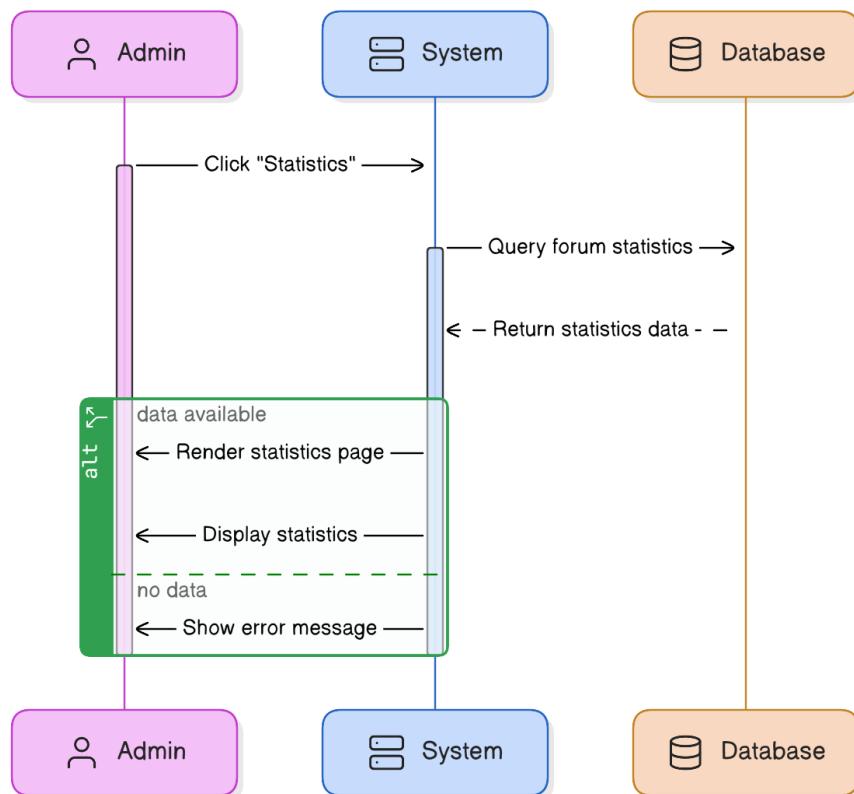
Name	Thống kê người dùng
Brief description	Quản lý xem được thống kê về người dùng hệ thống
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.

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 1.40. Đặc tả Use case - Thống kê người dùng

3.1.3.41. Thống kê Forum



Hình 3.41. Sequence - Thống kê Forum

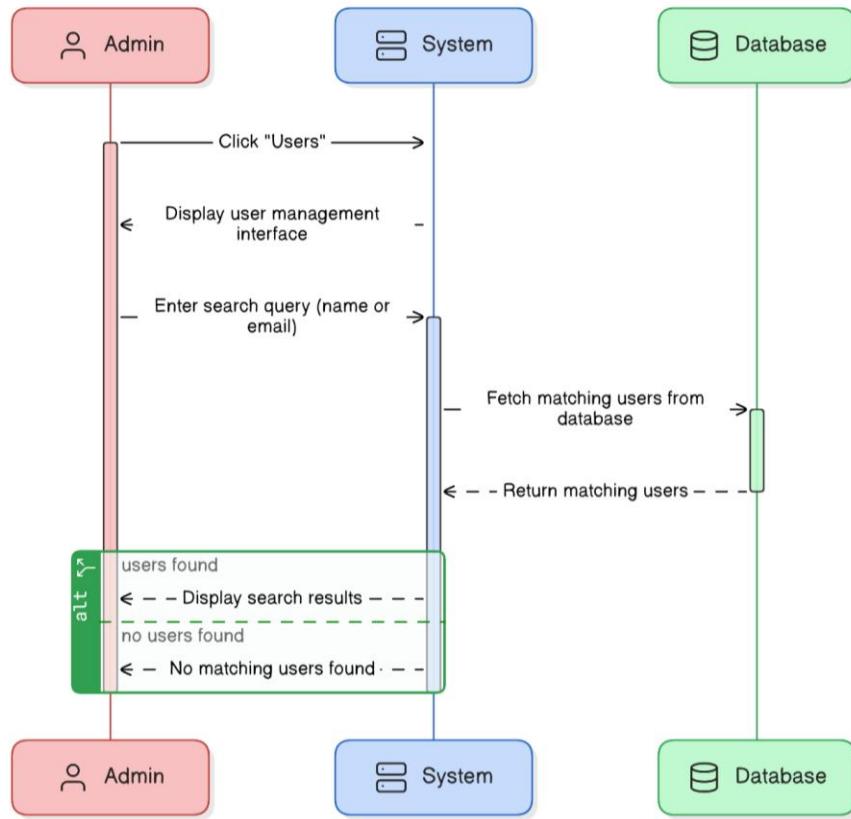
Chương 3: Phân tích thiết kế hệ thống

Name	Thống kê Forum
Brief description	Quản lý xem được thống kê về diễn đàn
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.
Post-conditions	Hiển thị thống kê về diễn đàn
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 diễn đàn.
Alternative flow	
Extension point	

Bảng 1.41. Đặc tả Use case - Thống kê Forum

Chương 3: Phân tích thiết kế hệ thống

3.1.3.42. Tìm kiếm người dùng



Hình 3.42. Sequence - Tìm kiếm người dùng

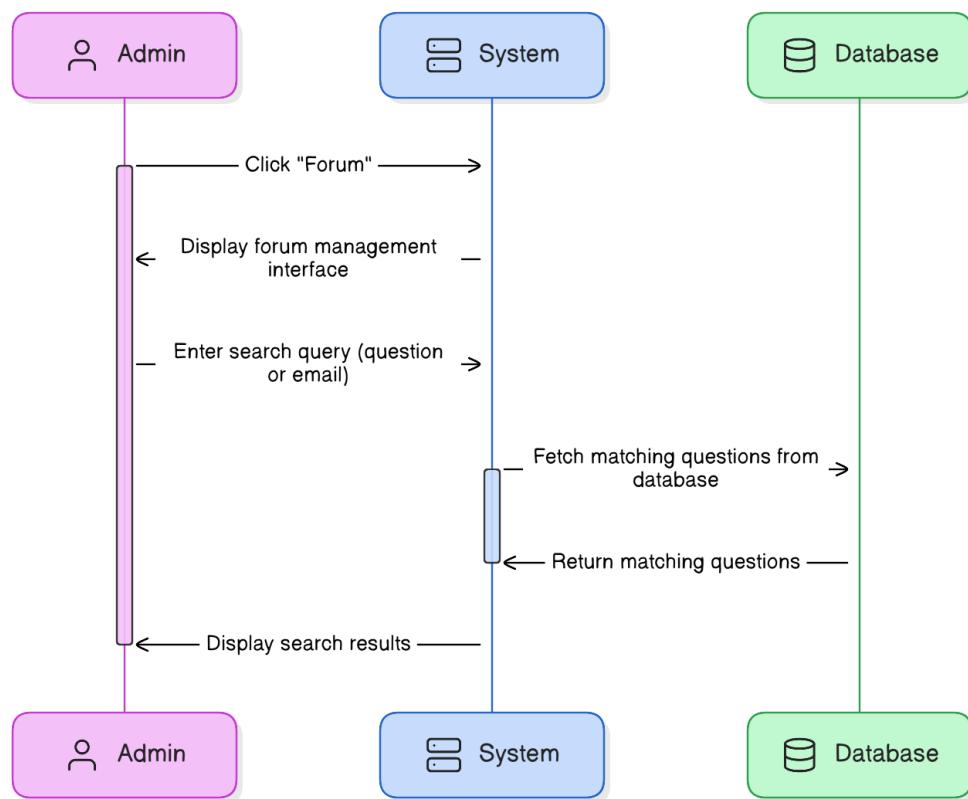
Name	Tìm kiếm người dùng
Brief description	Quản lý tìm kiếm người dùng theo tên hoặc email đăng nhập hệ thống
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.
Post-conditions	Hiển thị thông tin người dùng theo nội dung nhập
Flow of events	
Basic flow	1. Quản lý bấm vào nút "Users" ở Sidebar.

Chương 3: Phân tích thiết kế hệ thống

	2. Hệ thống chuyển đến quản lý người dùng. 3. Hiển thị danh sách người dùng. 4. Quản lý nhập nội dung (tên, email) của người dùng cần tìm 5. Hệ thống hiển thị người dùng cần tìm
Alternative flow	
Extension point	

Bảng 1.42. Đặc tả Use case - Tìm kiếm người dùng

3.1.3.43. Tìm kiếm câu hỏi



Hình 3.43. Sequence - Tìm kiếm câu hỏi

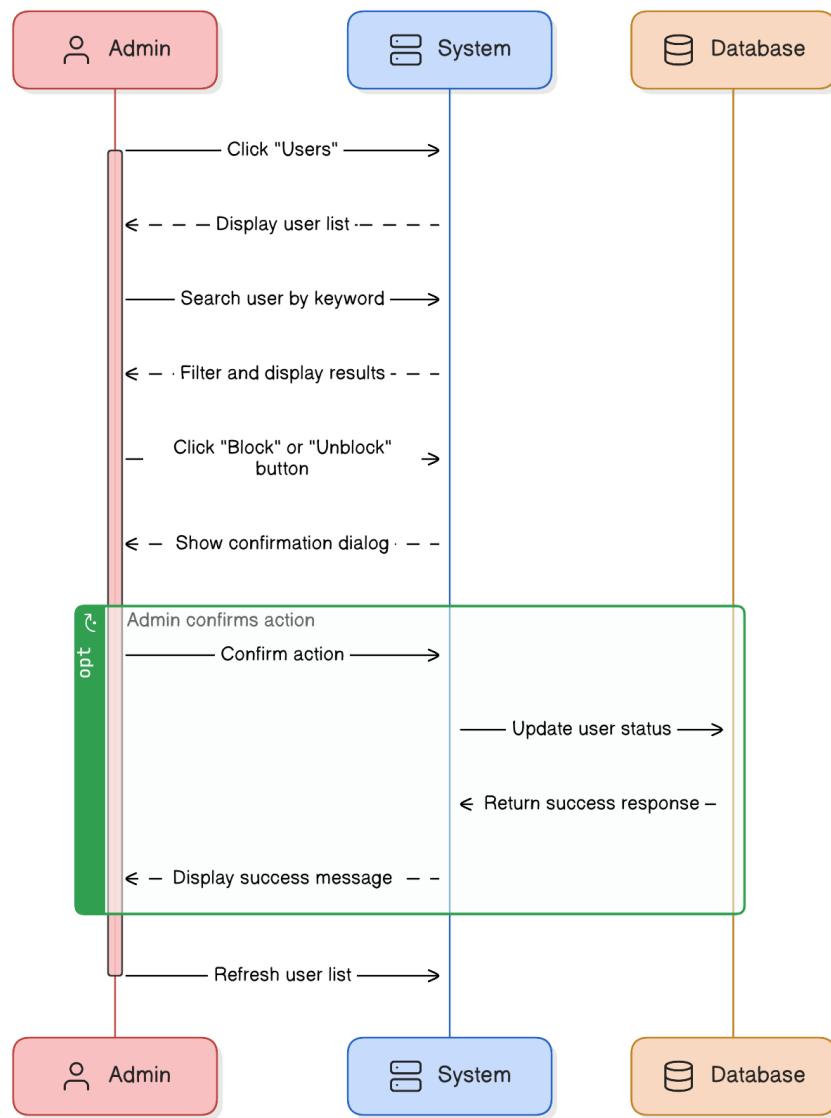
Chương 3: Phân tích thiết kế hệ thống

Name	Tìm kiếm câu hỏi
Brief description	Quản lý tìm kiếm câu hỏi trong diễn đàn theo tên hoặc email đăng nhập hệ thống của người dùng
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.
Post-conditions	Hiển thị câu hỏi của người dùng theo nội dung nhập
Flow of events	
Basic flow	<ol style="list-style-type: none">1. Quản lý bấm vào nút "Forum" ở Sidebar.2. Hệ thống chuyển đến quản lý Forum.3. Hiển thị danh sách câu hỏi.4. Quản lý nhập nội dung câu hỏi cần tìm hoặc email người dùng5. Hệ thống hiển thị câu hỏi cần tìm
Alternative flow	
Extension point	

Bảng 1.43. Đặc tả Use case - Tìm kiếm câu hỏi

Chương 3: Phân tích thiết kế hệ thống

3.1.3.44. Chặn/bỏ chặn người dùng



Hình 3.44. 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"> 1. Quản lý bấm vào menu "Users" trong thanh Sidebar. 2. 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. 3. Quản lý tìm kiếm người dùng bằng cách nhập từ khóa vào ô tìm kiếm. 4. Quản lý bấm vào nút hành động (chặn hoặc bỏ chặn) tương ứng với người dùng. 5. Hệ thống hiển thị hộp thoại xác nhận với thông tin người dùng và các tùy chọn: <ul style="list-style-type: none"> • Nếu chặn: Yêu cầu nhập lý do chặn. • Nếu bỏ chặn: Chỉ cần xác nhận. 6. Quản lý xác nhận hành động: <ul style="list-style-type: none"> • Bấm nút "Block" hoặc "Activate". 7. Hệ thống cập nhật trạng thái người dùng trong cơ sở dữ liệu và hiển thị thông báo hành động thành công. 8. Hộp thoại đóng lại, danh sách người dùng được cập nhật với trạng thái mới.
Alternative flow	<ol style="list-style-type: none"> 1. Hệ thống không thể cập nhật trạng thái người dùng do lỗi kết nối

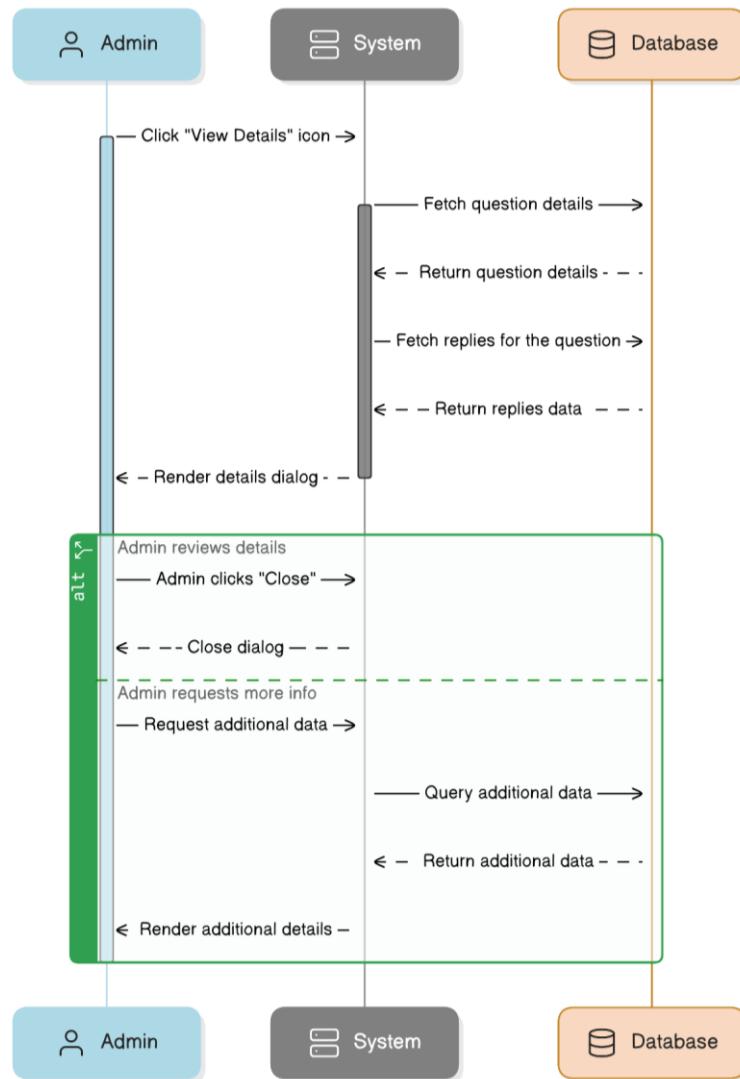
Chương 3: Phân tích thiết kế hệ thống

	<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. <p>2. Quản lý hủy hành động trong hộp thoại</p> <ul style="list-style-type: none">• Quản lý bấm nút "Cancel" trong hộp thoại xác nhận.• Hộp thoại đóng lại và không có thay đổi nào được thực hiện.
Extension point	

Bảng 1.44. Đặc tả Use case - Chặn/bỏ chặn người dùng

Chương 3: Phân tích thiết kế hệ thống

3.1.3.45. Xem chi tiết câu hỏi



Hình 3.44. Sequence - Xem chi tiết câu hỏi

Name	Xem chi tiết câu hỏi
Brief description	Cho phép quản lý xem đầy đủ chi tiết của một câu hỏi trong diễn đàn, bao gồm nội dung câu hỏi, thông tin người đăng và các phản hồi.
Actor(s)	Quản lý

Chương 3: Phân tích thiết kế hệ thống

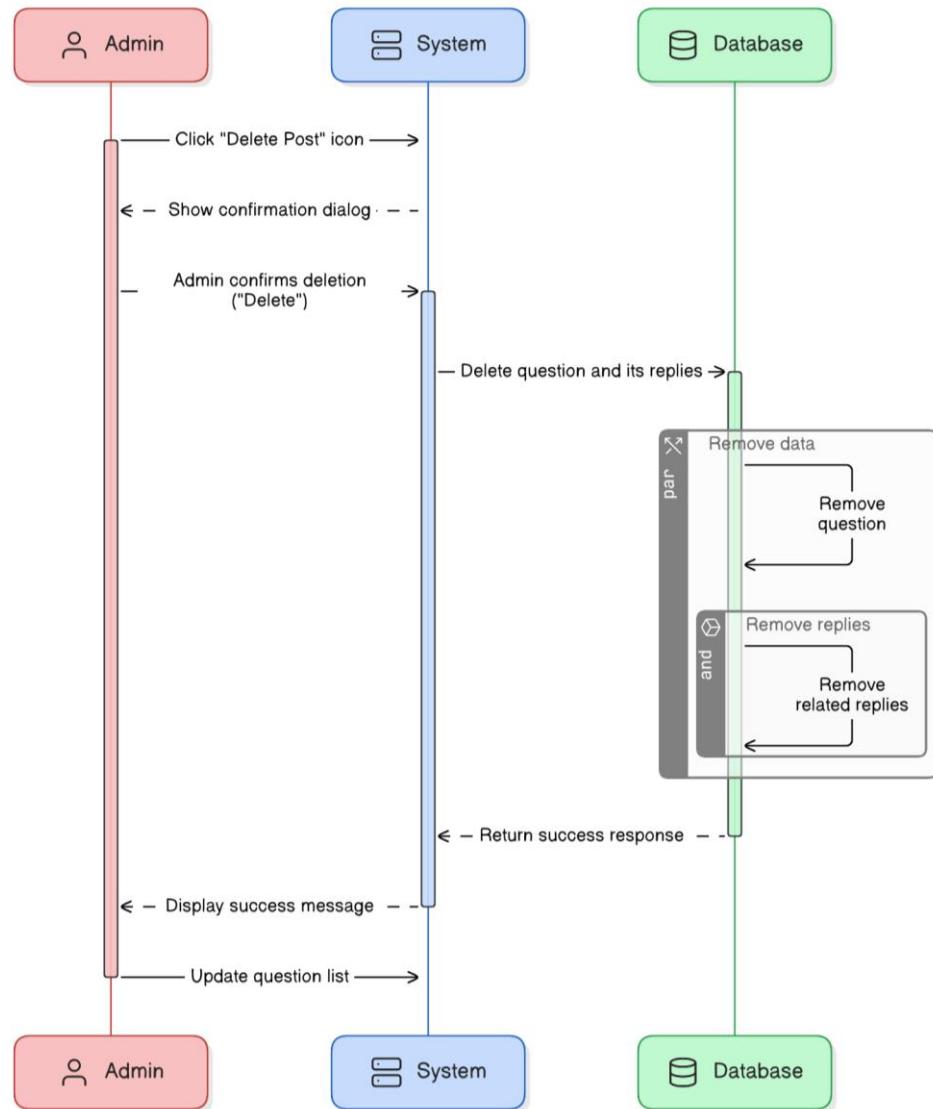
Pre-conditions	<p>Quản lý đã đăng nhập thành công vào hệ thống.</p> <p>Danh sách các câu hỏi đã được tải và hiển thị trên giao diện.</p>
Post-conditions	Chi tiết câu hỏi và danh sách các phản hồi được hiển thị đầy đủ trong hộp thoại.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Quản lý bấm vào biểu tượng "View Details" (icon hình mắt) trong danh sách các câu hỏi. 2. Hệ thống hiển thị hộp thoại chứa thông tin chi tiết về câu hỏi, bao gồm: <ul style="list-style-type: none"> • Nội dung câu hỏi. • Email người đăng. • Thời gian đăng. • Danh sách các phản hồi (nếu có). 3. Quản lý có thể xem chi tiết từng phản hồi, bao gồm: <ul style="list-style-type: none"> • Nội dung phản hồi. • Email người phản hồi. • Thời gian phản hồi. • Trạng thái đã đọc hoặc chưa đọc. 4. Hệ thống cung cấp tùy chọn xóa phản hồi nếu cần thiết. 5. Quản lý bấm nút "Close" để đóng hộp thoại.
Alternative flow	<p>Không có phản hồi cho câu hỏi</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị chi tiết câu hỏi nhưng danh sách phản hồi trống.

Chương 3: Phân tích thiết kế hệ thống

	2. Hệ thống hiển thị thông báo "No replies available for this question."
Extension point	

Bảng 1.44. Đặc tả Use case - Xem chi tiết câu hỏi

3.1.3.46. Xóa câu hỏi của người dùng



Hình 3.44. Sequence - Xóa câu hỏi của người dùng

Chương 3: Phân tích thiết kế hệ thống

Name	Xóa câu hỏi của người dùng
Brief description	Quản lý có thể xóa một câu hỏi trong diễn đàn cùng với toàn bộ phản hồi liên quan.
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống. Danh sách các câu hỏi đã được tải và hiển thị trên giao diện.
Post-conditions	Câu hỏi và toàn bộ phản hồi liên quan được xóa khỏi cơ sở dữ liệu. Giao diện danh sách câu hỏi được cập nhật.
Flow of events	
Basic flow	<ol style="list-style-type: none"> 1. Quản lý bấm vào biểu tượng "Delete Post" (icon thùng rác) bên cạnh câu hỏi muốn xóa trong danh sách câu hỏi. 2. Hệ thống hiển thị hộp thoại xác nhận với nội dung: <ul style="list-style-type: none"> • "Are you sure you want to delete this forum post and all its replies?" • Hai nút lựa chọn: "Cancel" và "Delete". 3. Quản lý bấm nút "Delete" để xác nhận hành động. 4. Hệ thống thực hiện các bước sau: <ul style="list-style-type: none"> • Xóa câu hỏi khỏi cơ sở dữ liệu tại đường dẫn <code>forum/{forumId}</code>. • Xóa toàn bộ phản hồi liên quan (nếu có). • Hiển thị thông báo thành công: "Forum post deleted successfully". 5. Giao diện danh sách câu hỏi được cập nhật để loại bỏ câu hỏi vừa bị xóa.

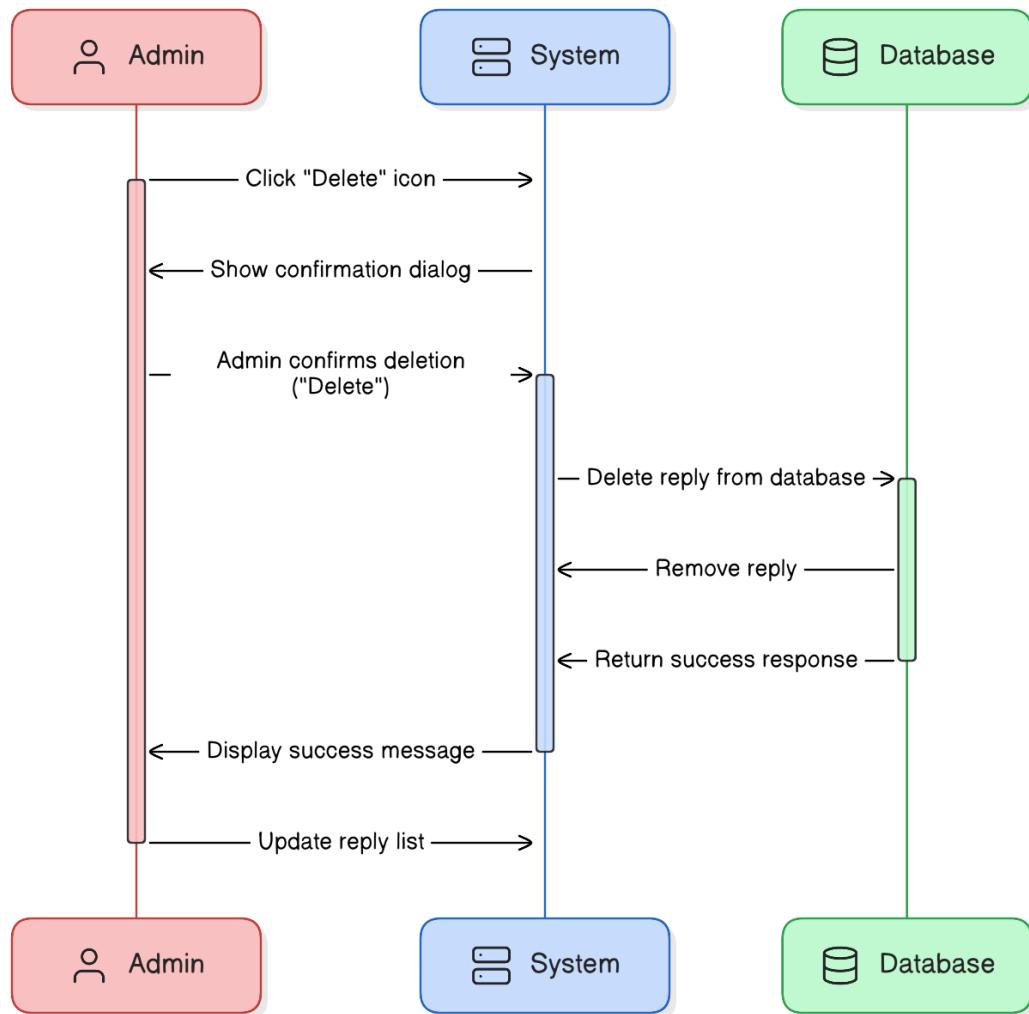
Chương 3: Phân tích thiết kế hệ thống

Alternative flow	<ol style="list-style-type: none">1. Quản lý hủy hành động xóa câu hỏi<ul style="list-style-type: none">• Quản lý bấm nút "Cancel" trong hộp thoại xác nhận.• Hệ thống đóng hộp thoại mà không thực hiện hành động xóa.2. Hệ thống không thể xóa câu hỏi do lỗi kết nối<ul style="list-style-type: none">• Hệ thống hiển thị thông báo lỗi: "Failed to delete forum post. Please try again later."• Câu hỏi không bị xóa và giao diện danh sách câu hỏi giữ nguyên.
Extension point	

Bảng 1.44. ĐẶC TẢ USE CASE - XÓA CÂU HỎI CỦA NGƯỜI DÙNG

Chương 3: Phân tích thiết kế hệ thống

3.1.3.47. Xóa câu trả lời của người dùng



Hình 3.44. Sequence - Xóa câu trả lời của người dùng

Name	Xóa câu trả lời của người dùng
Brief description	Quản lý có thể xóa một trả lời trong diễn đàn cùng với toàn bộ phản hồi liên quan.
Actor(s)	Quản lý
Pre-conditions	Quản lý đã đăng nhập thành công vào hệ thống.

Chương 3: Phân tích thiết kế hệ thống

	Danh sách các câu hỏi đã được tải và hiển thị trên giao diện.
Post-conditions	Câu hỏi trả lời được xóa khỏi cơ sở dữ liệu. Giao diện danh sách trả lời được cập nhật.
Flow of events	
Basic flow	<p>6. Quản lý bấm vào biểu tượng "Delete" (icon thùng rác) bên cạnh trả lời muốn xóa trong danh sách câu trả lời.</p> <p>7. Hệ thống hiển thị hộp thoại xác nhận với nội dung:</p> <ul style="list-style-type: none"> • "Are you sure you want to delete this reply?" • Hai nút lựa chọn: "Cancel" và "Delete". <p>8. Quản lý bấm nút "Delete" để xác nhận hành động.</p> <p>9. Hệ thống thực hiện các bước sau:</p> <ul style="list-style-type: none"> • Xóa trả lời khỏi cơ sở dữ liệu tại đường dẫn forum/{forumId}. • Hiển thị thông báo thành công: "Reply deleted successfully". <p>10. Giao diện danh sách trả lời được cập nhật để loại bỏ trả lời vừa bị xóa.</p>
Alternative flow	<p>3. Quản lý hủy hành động xóa câu hỏi</p> <ul style="list-style-type: none"> • Quản lý bấm nút "Cancel" trong hộp thoại xác nhận. • Hệ thống đóng hộp thoại mà không thực hiện hành động xóa. <p>4. Hệ thống không thể xóa câu hỏi do lỗi kết nối</p> <ul style="list-style-type: none"> • Hệ thống hiển thị thông báo lỗi: "Failed to delete forum post. Please try again later."

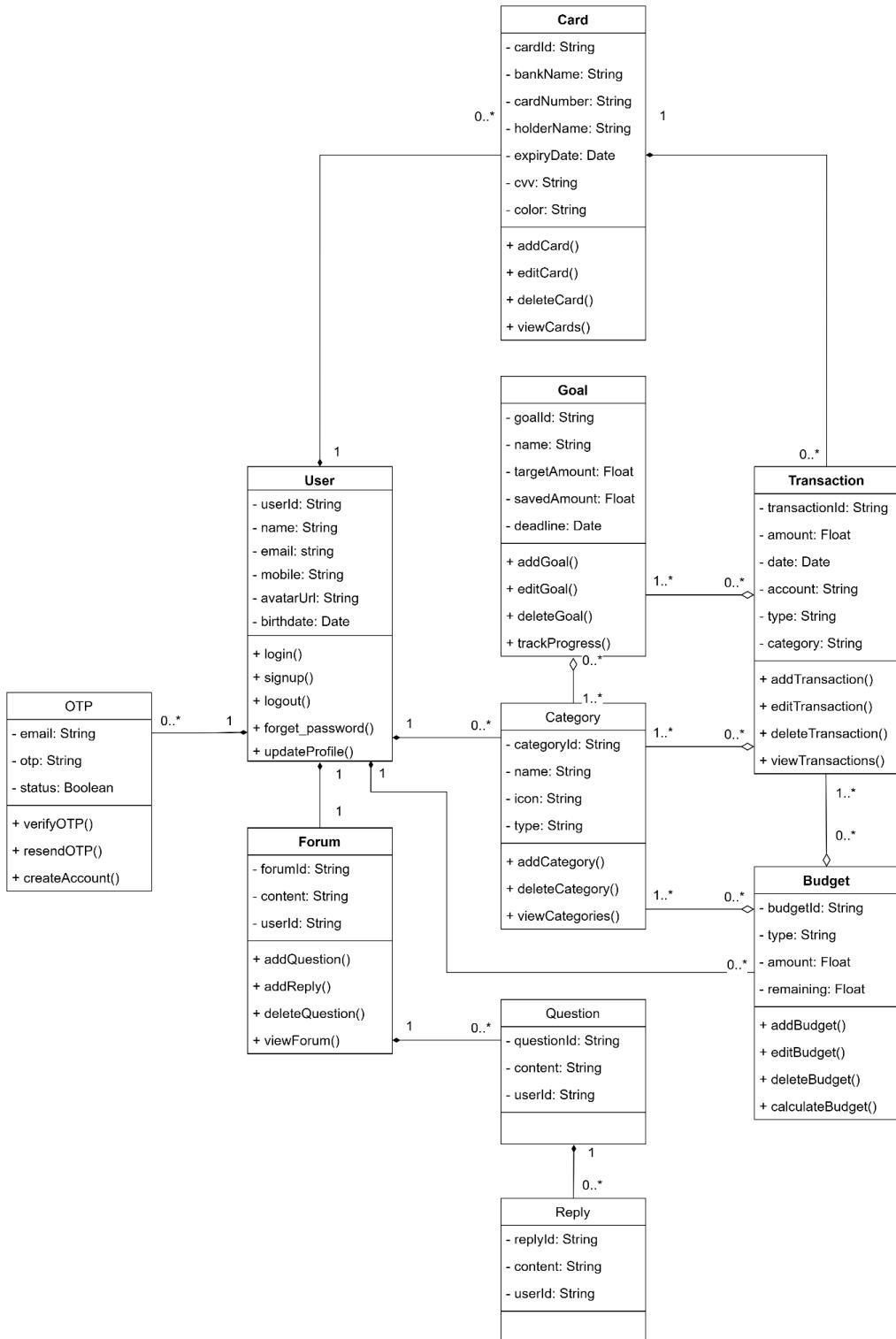
Chương 3: Phân tích thiết kế hệ thống

	<ul style="list-style-type: none">• Câu trả lời không bị xóa và giao diện danh sách câu hỏi giữ nguyên.
Extension point	

Bảng 1.44. Đặc tả Use case - Xóa câu trả lời của người dùng

Chương 3: Phân tích thiết kế hệ thống

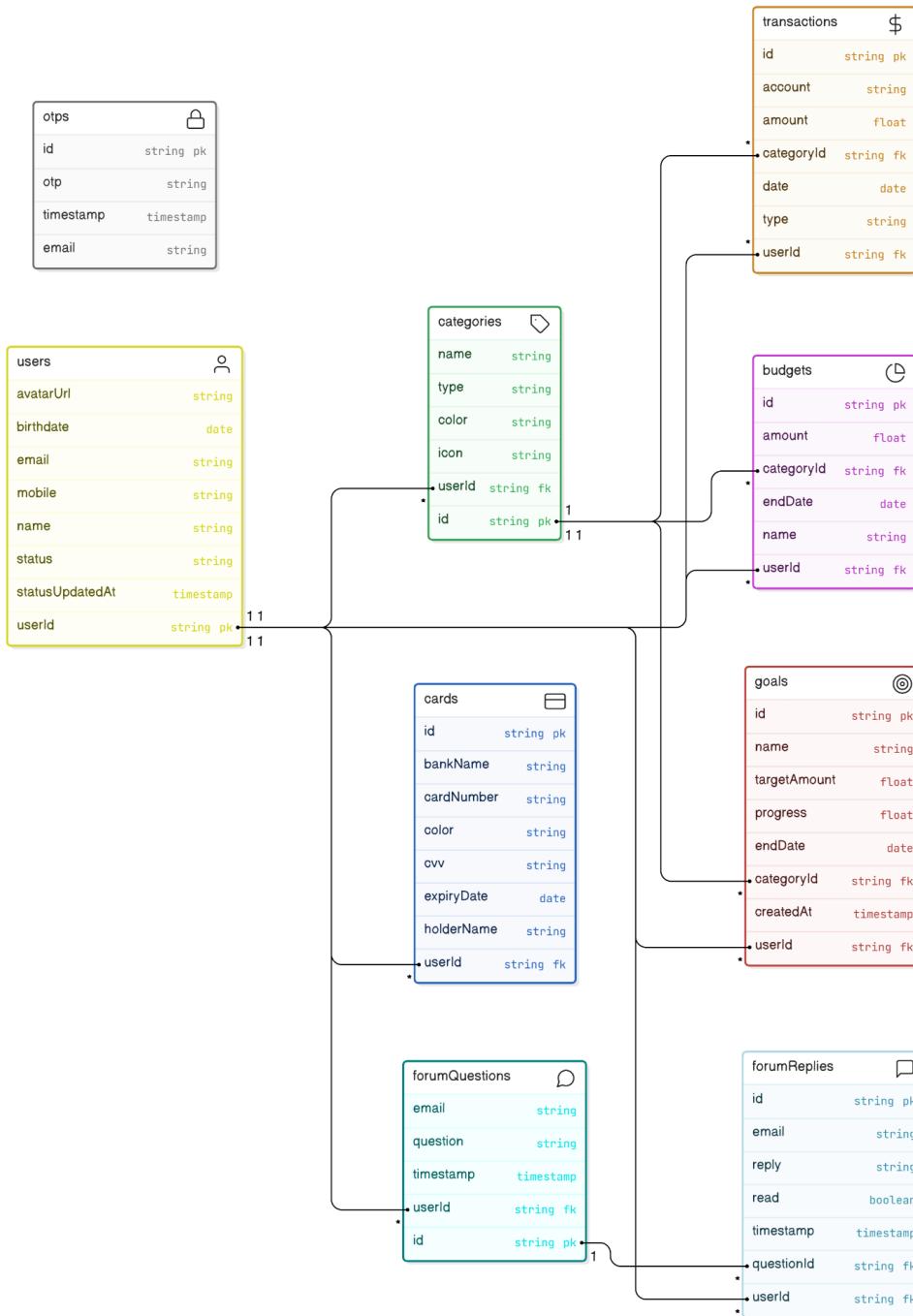
3.1.4. Class Diagram



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

3.2. 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.
- 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.

Chương 3: Phân tích thiết kế hệ thống

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.
 - 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.

Chương 3: Phân tích thiết kế hệ thống

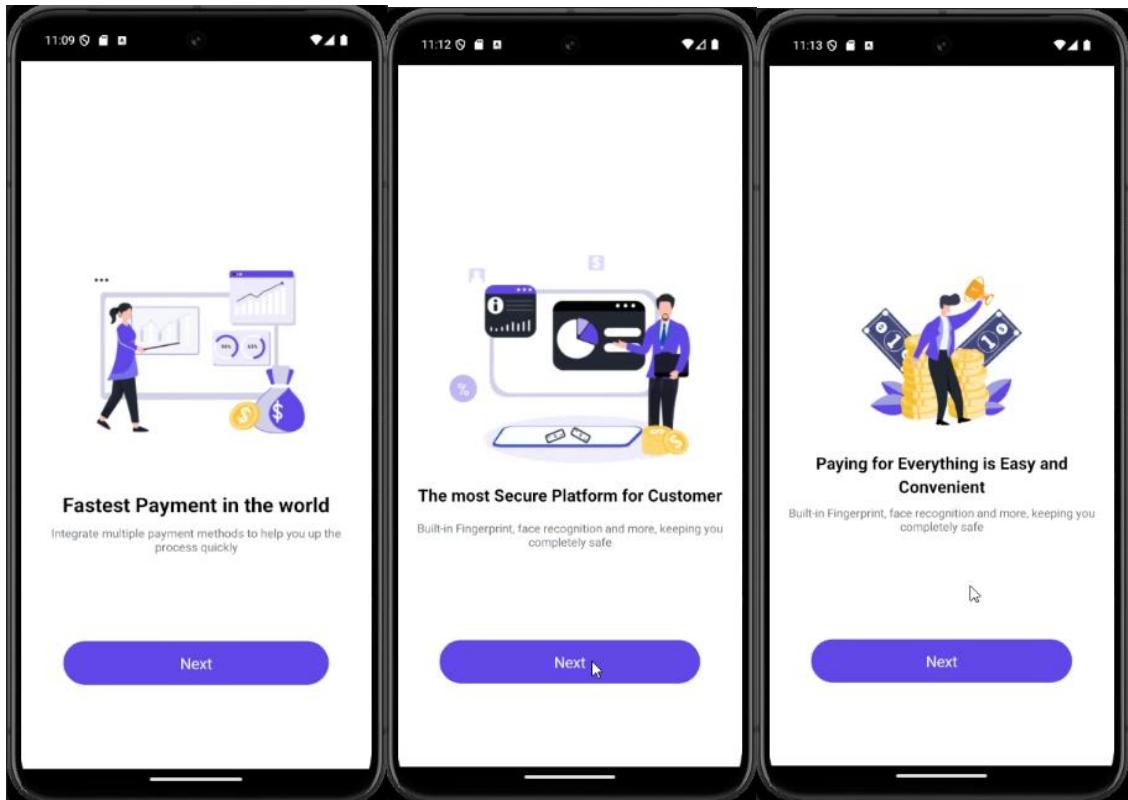
- Backend sẽ được triển khai trên AWS với Firebase làm nền tảng backend để xử lý và lưu trữ dữ liệu.
- Firebase sẽ được sử dụng để triển khai toàn bộ hệ thống backend, từ xác thực người dùng đến lưu trữ dữ liệu và xử lý các chức năng liên quan đến giao dịch tài chính.

3.3. Thiết kế giao diện

3.3.1. Giao diện người dùng:

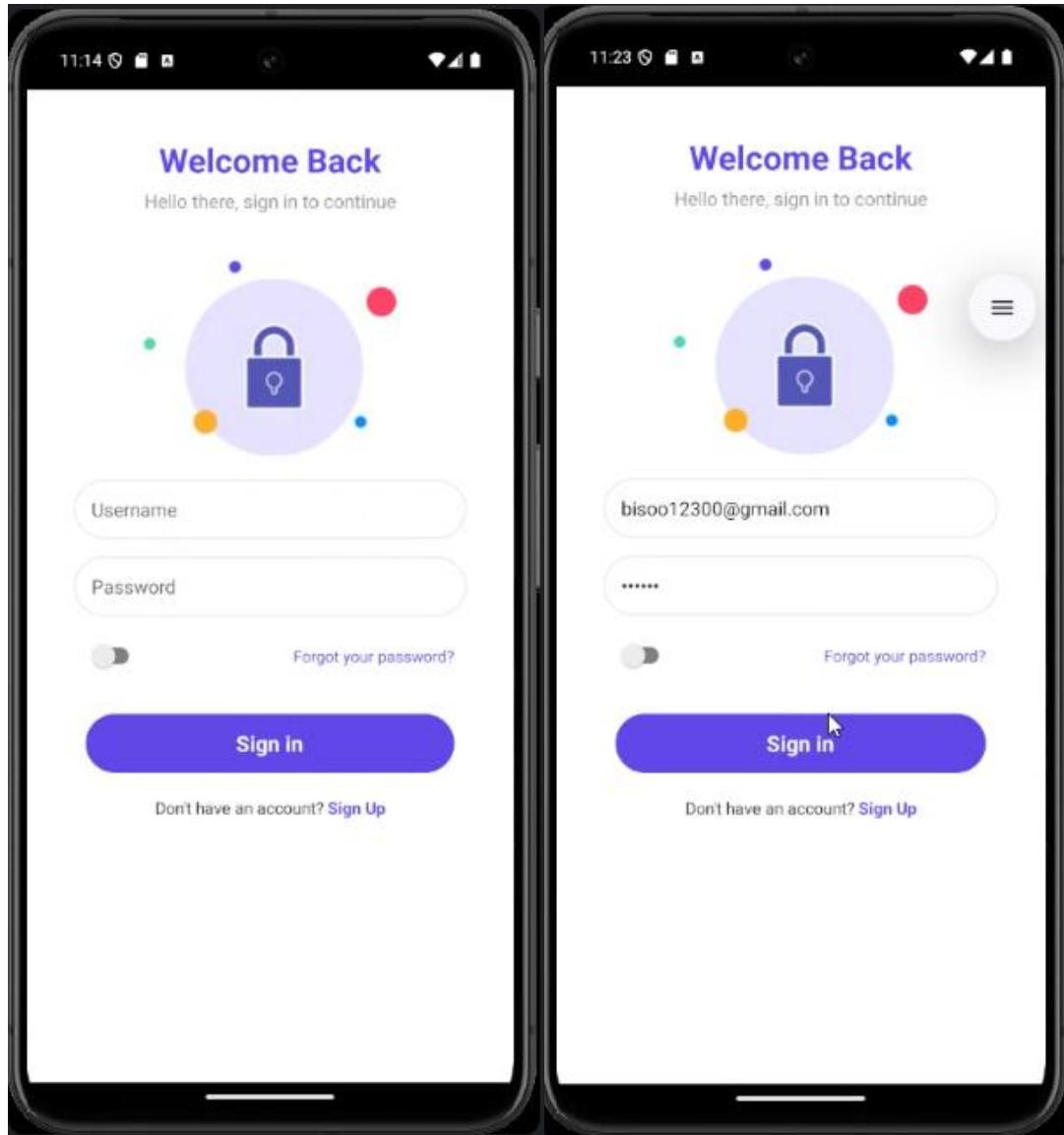
Hướng dẫn sử dụng App Quản Lý Tài Chính.

Người dùng mới truy cập vào App sẽ xuất hiện các Splash:



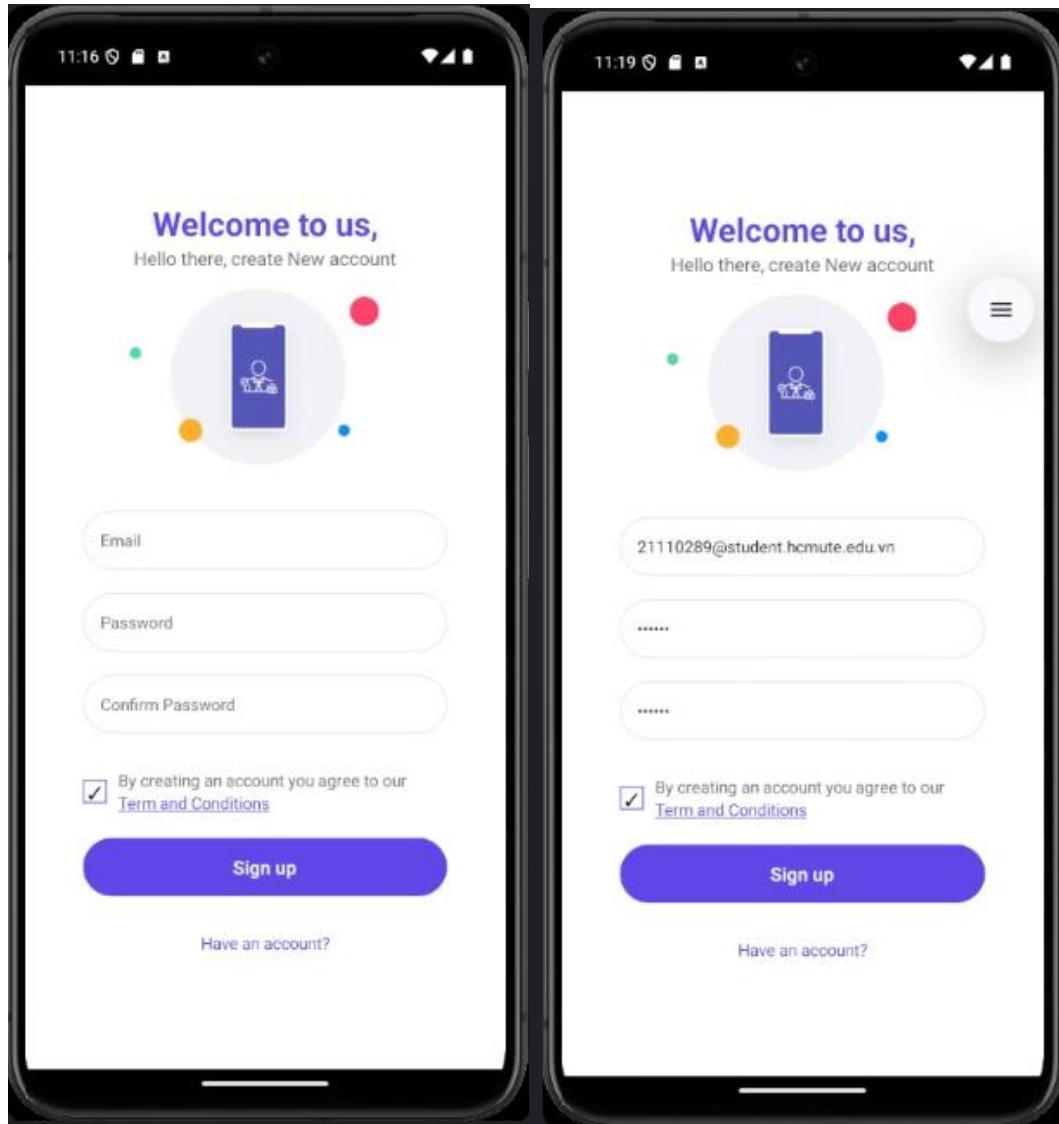
Chương 3: Phân tích thiết kế hệ thống

Sau đó người dùng sẽ được đưa đến trang đăng nhập:



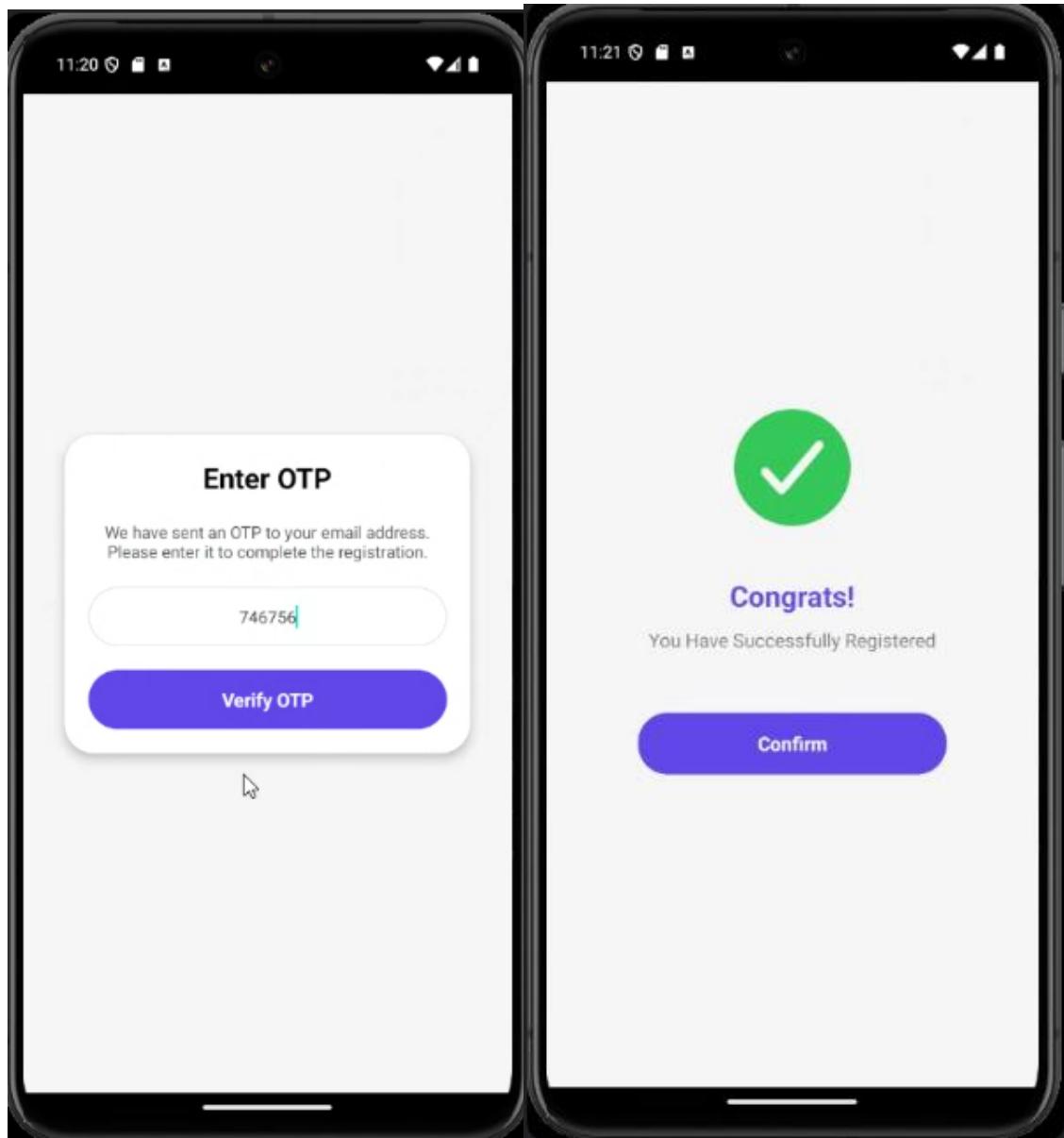
Chương 3: Phân tích thiết kế hệ thống

Phía bên dưới, để tiến đến trang đăng ký người dùng sẽ bấm vào Sign Up:



Chương 3: Phân tích thiết kế hệ thống

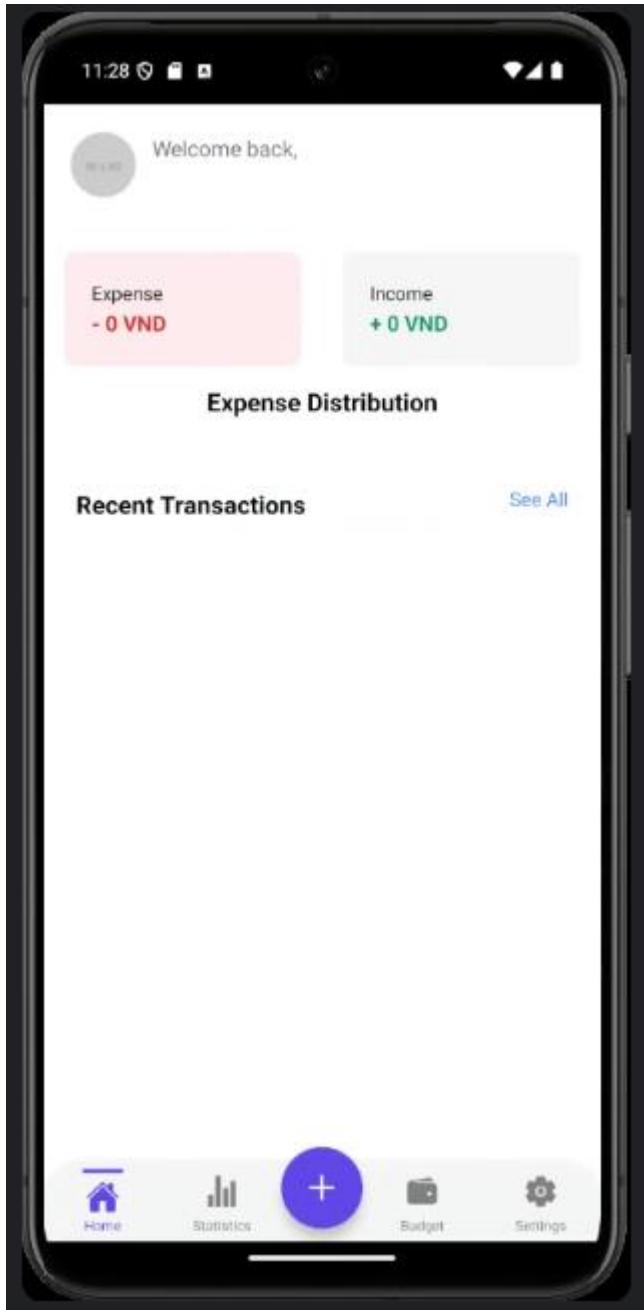
Bấm nút Sign Up sẽ đưa người dùng đến trang Verify OTP:



Sau đó người dùng sẽ được đưa trở về trang đăng nhập.

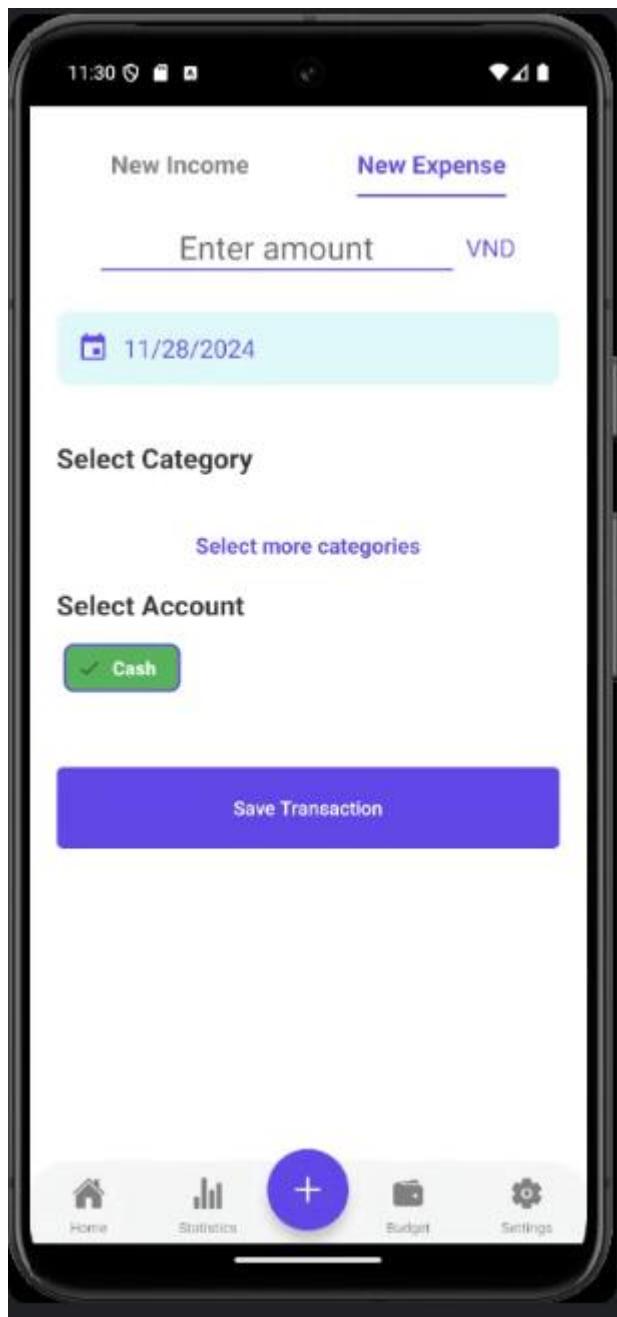
Chương 3: Phân tích thiết kế hệ thống

Sau khi tiến hành đăng nhập, người dùng sẽ được đưa tới trang Home:



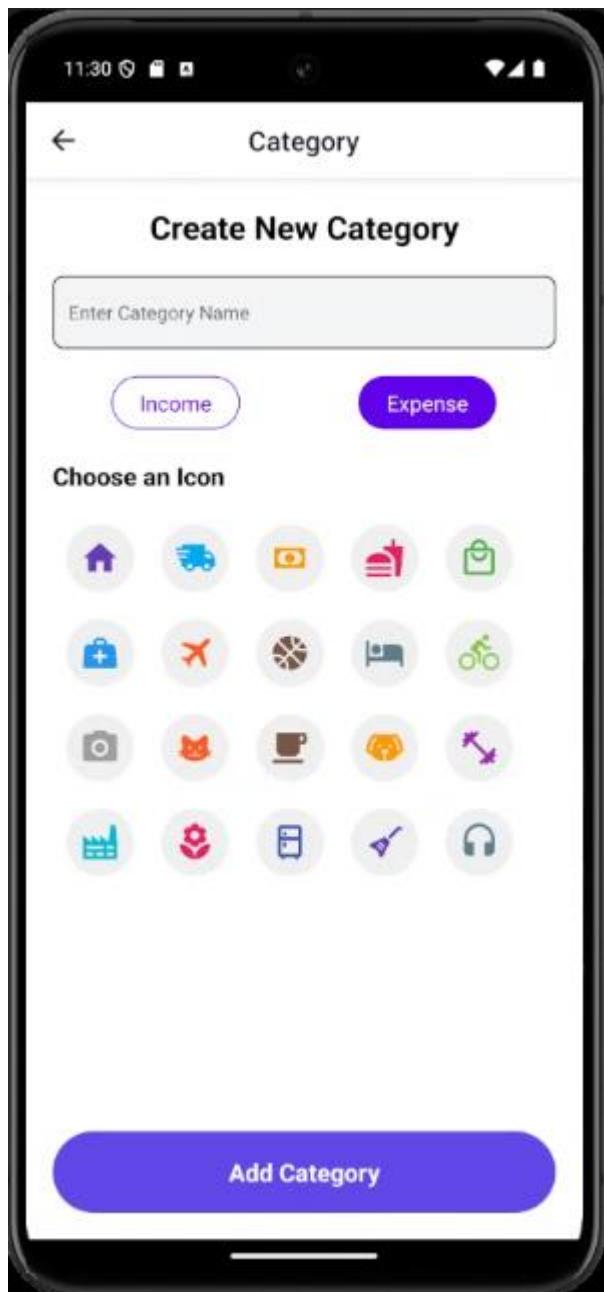
Do vừa mới tạo nên tài khoản người dùng chưa có dữ liệu, tiếp theo người dùng sẽ thực hiện chức năng thêm giao dịch mới bằng cách bấm vào nút "+" bên dưới.

Chương 3: Phân tích thiết kế hệ thống



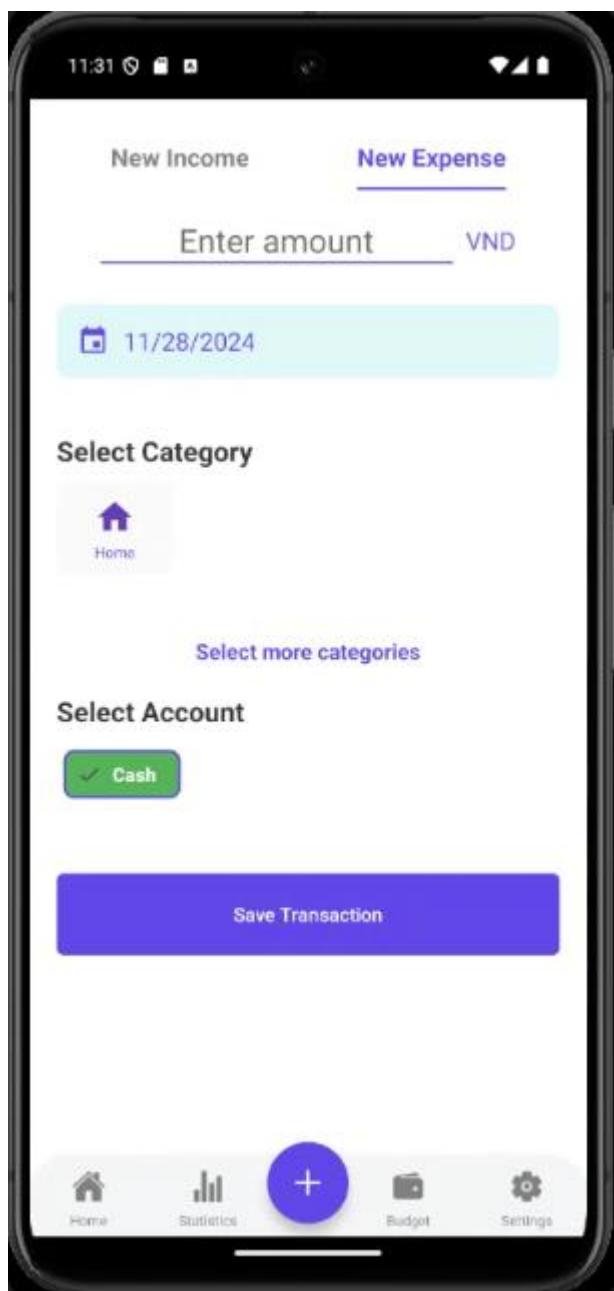
Để thêm được giao dịch người dùng cần phải tạo danh mục.

Chương 3: Phân tích thiết kế hệ thống



Để thêm danh mục, người dùng chọn icon và điền tên danh mục

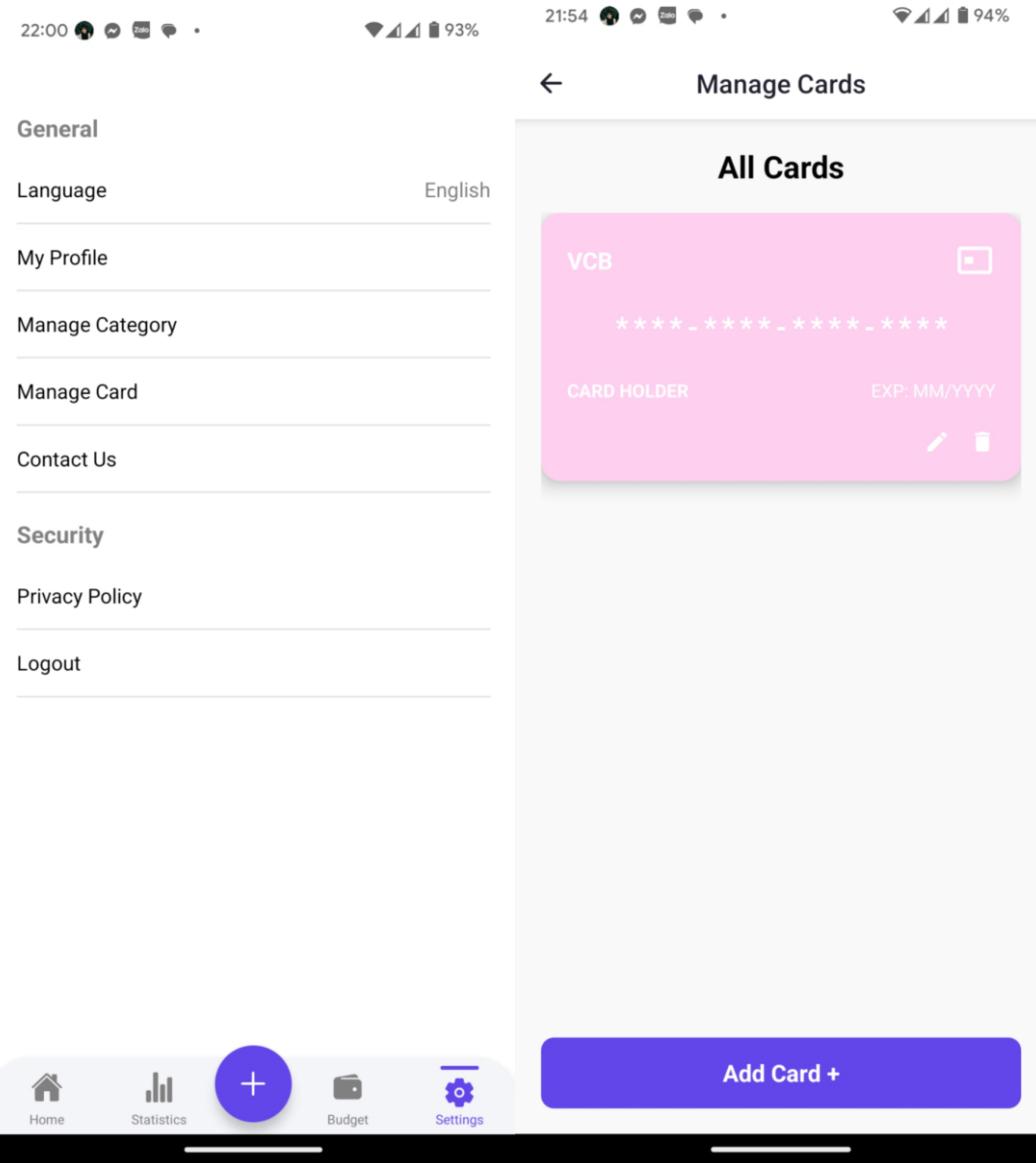
Chương 3: Phân tích thiết kế hệ thống



sau khi bấm thêm, quay lại trang thêm giao dịch đã hiện category,

Chương 3: Phân tích thiết kế hệ thống

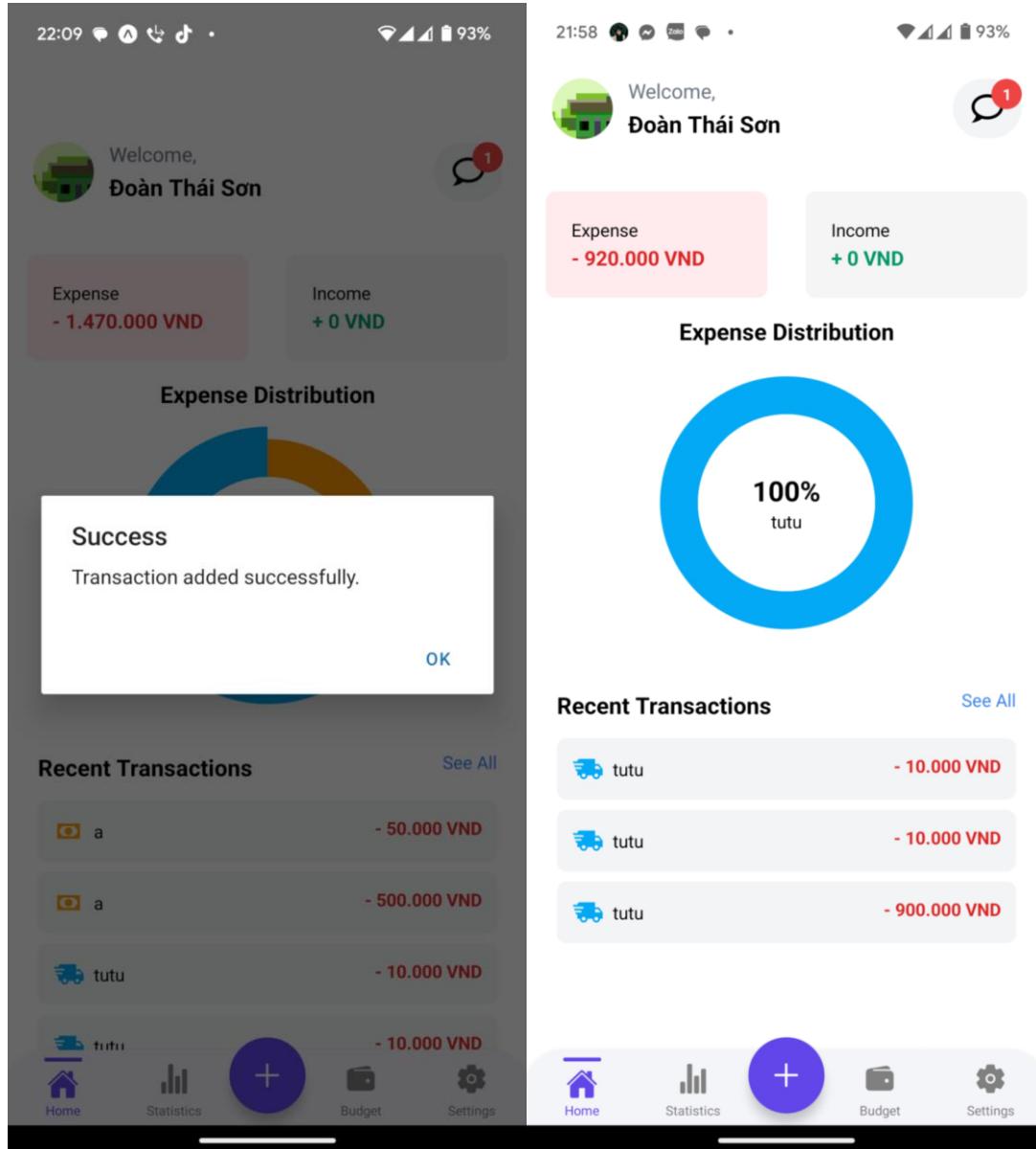
Ở phần account, người dùng sẽ chọn thẻ ngân hàng hoặc tiền mặt, để tiến hành thêm thẻ ta truy cập vào trang cài đặt và bấm vào Manage Card



ta có thể thêm xóa và chỉnh sửa

Chương 3: Phân tích thiết kế hệ thống

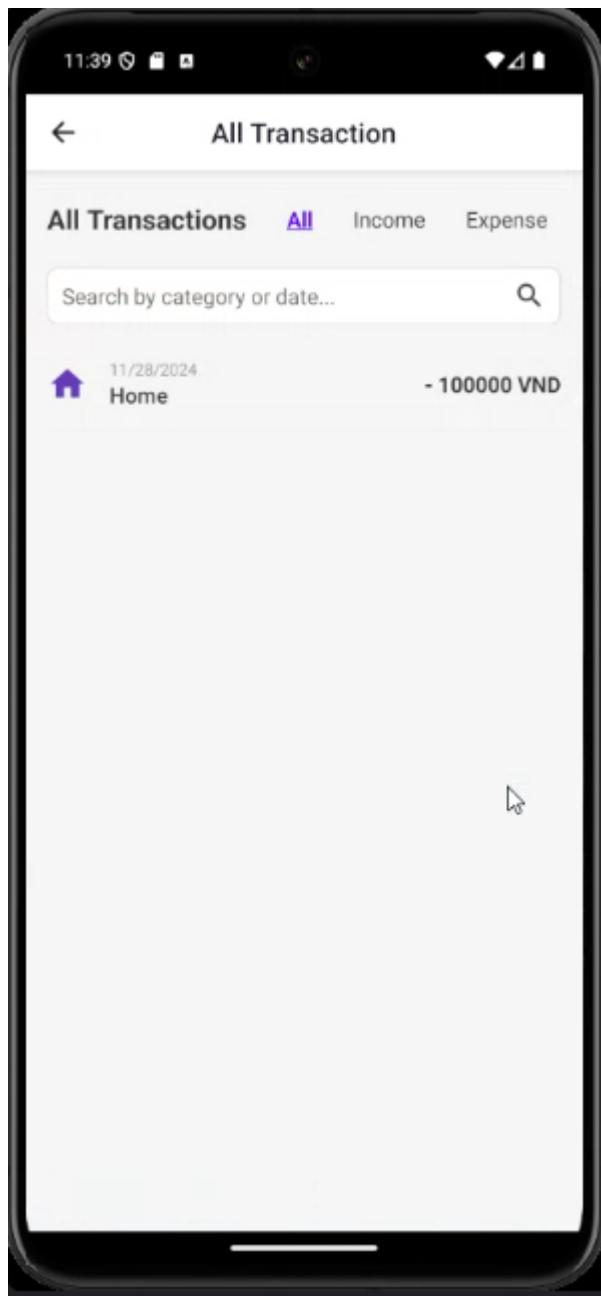
người dùng tiến hành chọn loại giao dịch là income hoặc expense, điền và chọn các thông khác sau đó bấm Save Transaction



người dùng được đưa về trang Home cùng với thông báo và cập nhật dữ liệu đã thêm thành công transaction.

Phía dưới sẽ hiện các transaction vừa thêm bên cạnh là “See All” bấm vào để hiển thị tất cả các transaction.

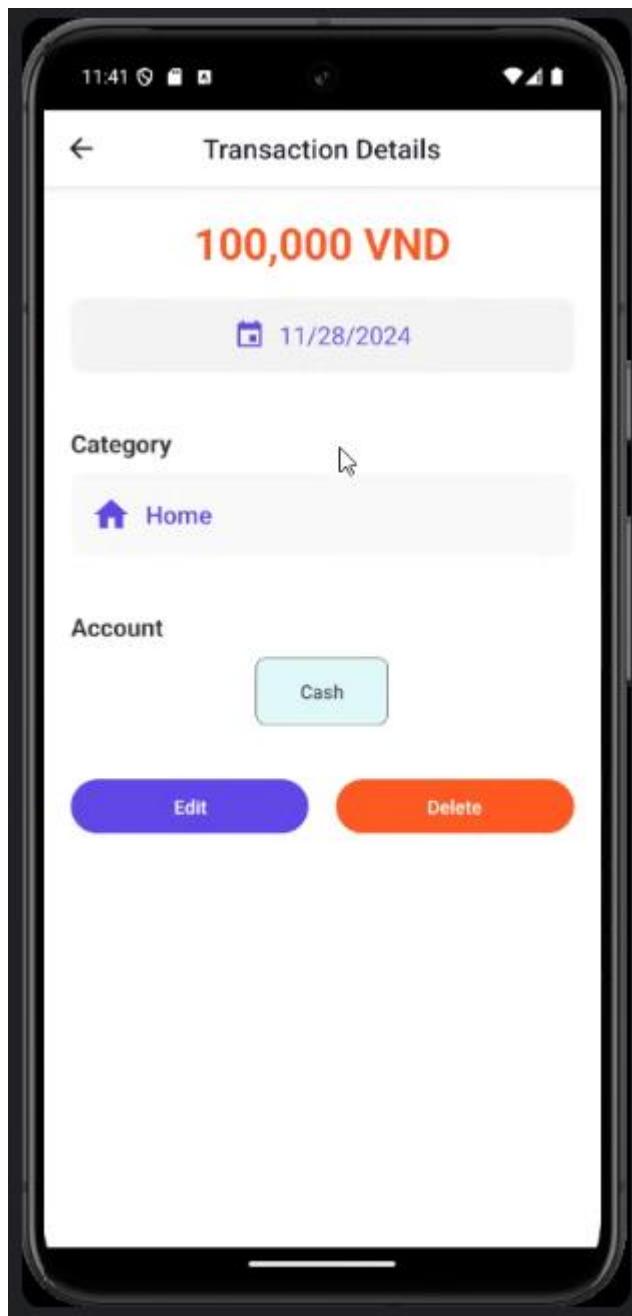
Chương 3: Phân tích thiết kế hệ thống



ở đây có chức năng tìm kiếm, và lọc theo income hoặc expense.

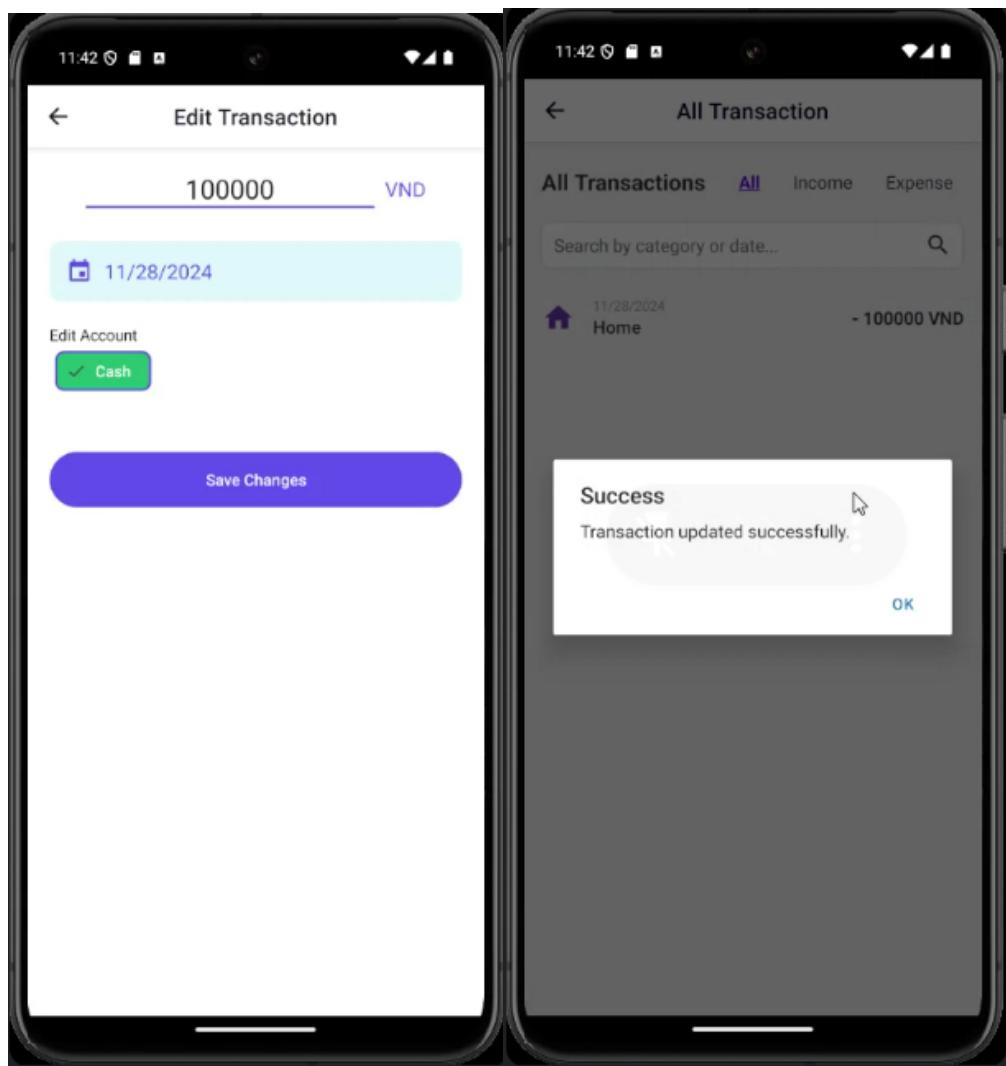
Chương 3: Phân tích thiết kế hệ thống

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



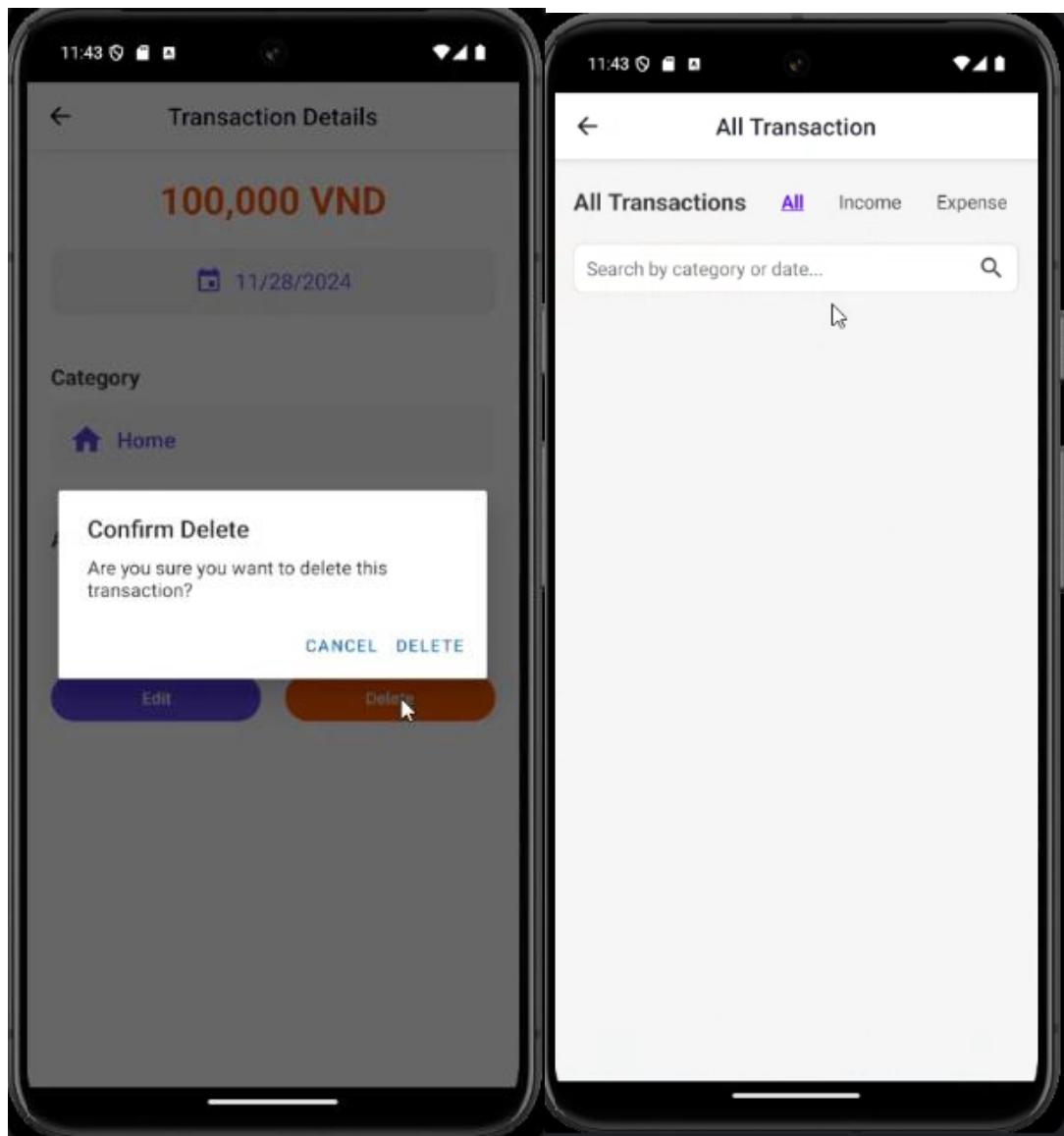
người dùng có thể thực hiện sửa thông tin giao dịch bằng "Edit" hoặc xóa giao dịch bằng "Delete".

Chương 3: Phân tích thiết kế hệ thống



Khi bấm vào Edit, người dùng sửa thông tin cần thiết và bấm Save Changes.

Chương 3: Phân tích thiết kế hệ thống



Khi bấm vào Delete và và confirm delete sẽ xóa giao dịch.

Chương 3: Phân tích thiết kế hệ thống

Để hiệu quả hơn, Tôi xin phép sử dụng 1 tài khoản đã có nhiều thông tin hơn để thực hiện các chức năng .

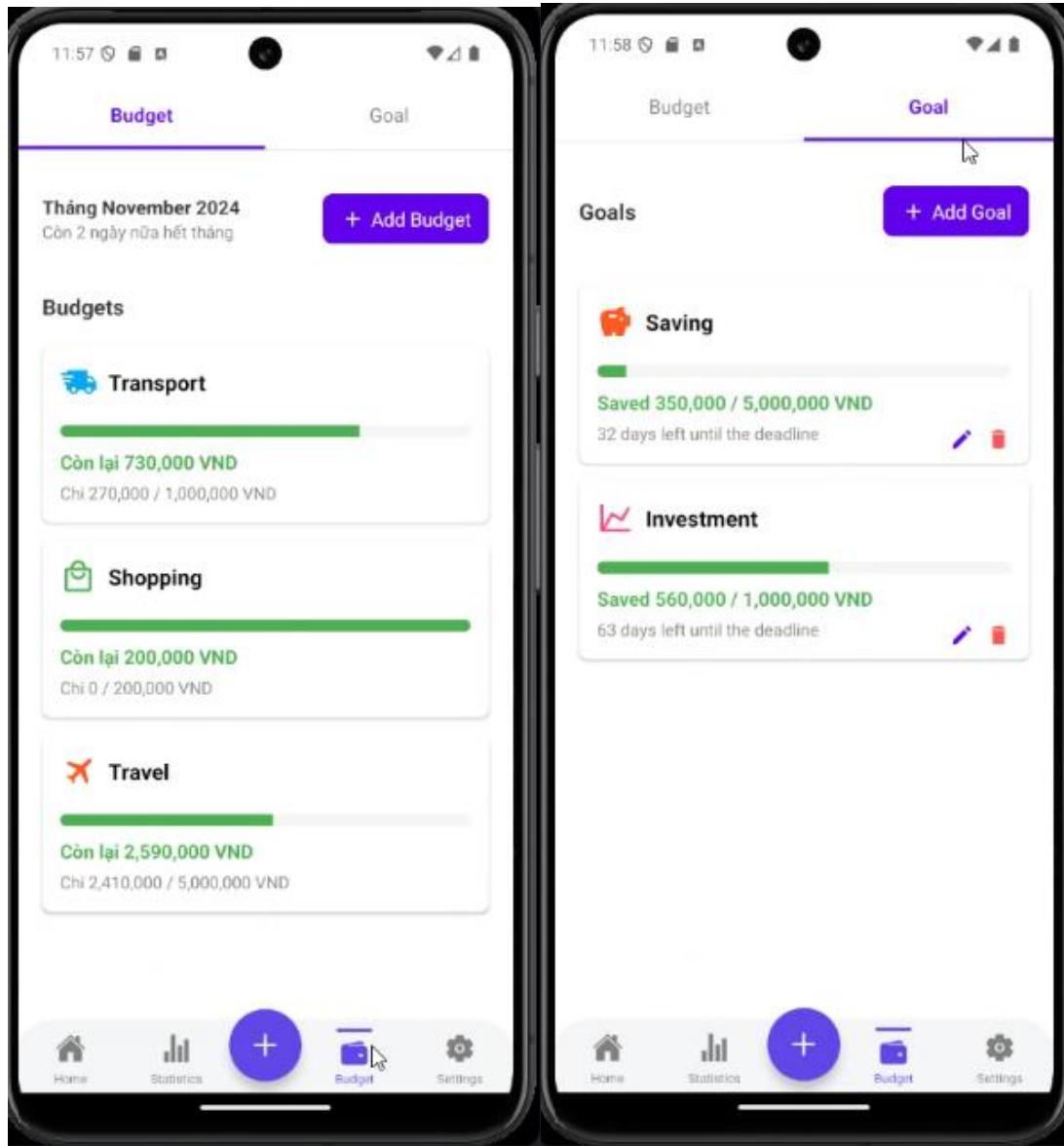
Đến với chức năng thống kê, ta bấm vào nút Statistics bên dưới:



Người dùng được đưa đến trang thống kê, ở đây sẽ thống kê các giao dịch đã tạo, có thể xem theo ngày, tháng.

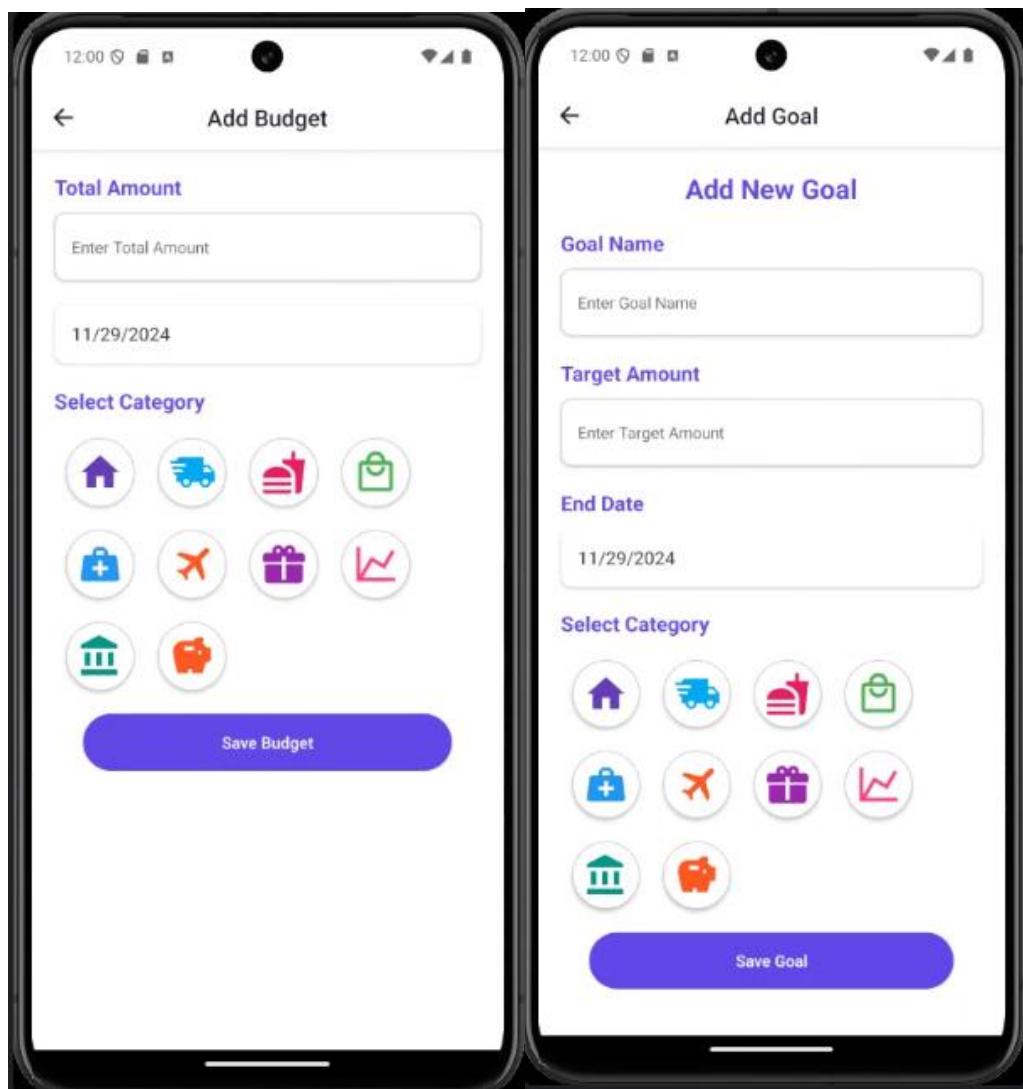
Chương 3: Phân tích thiết kế hệ thống

Để thực hiện chức năng tiếp theo, người dùng bấm vào nút Budget bên dưới để được đưa đến trang Budget:



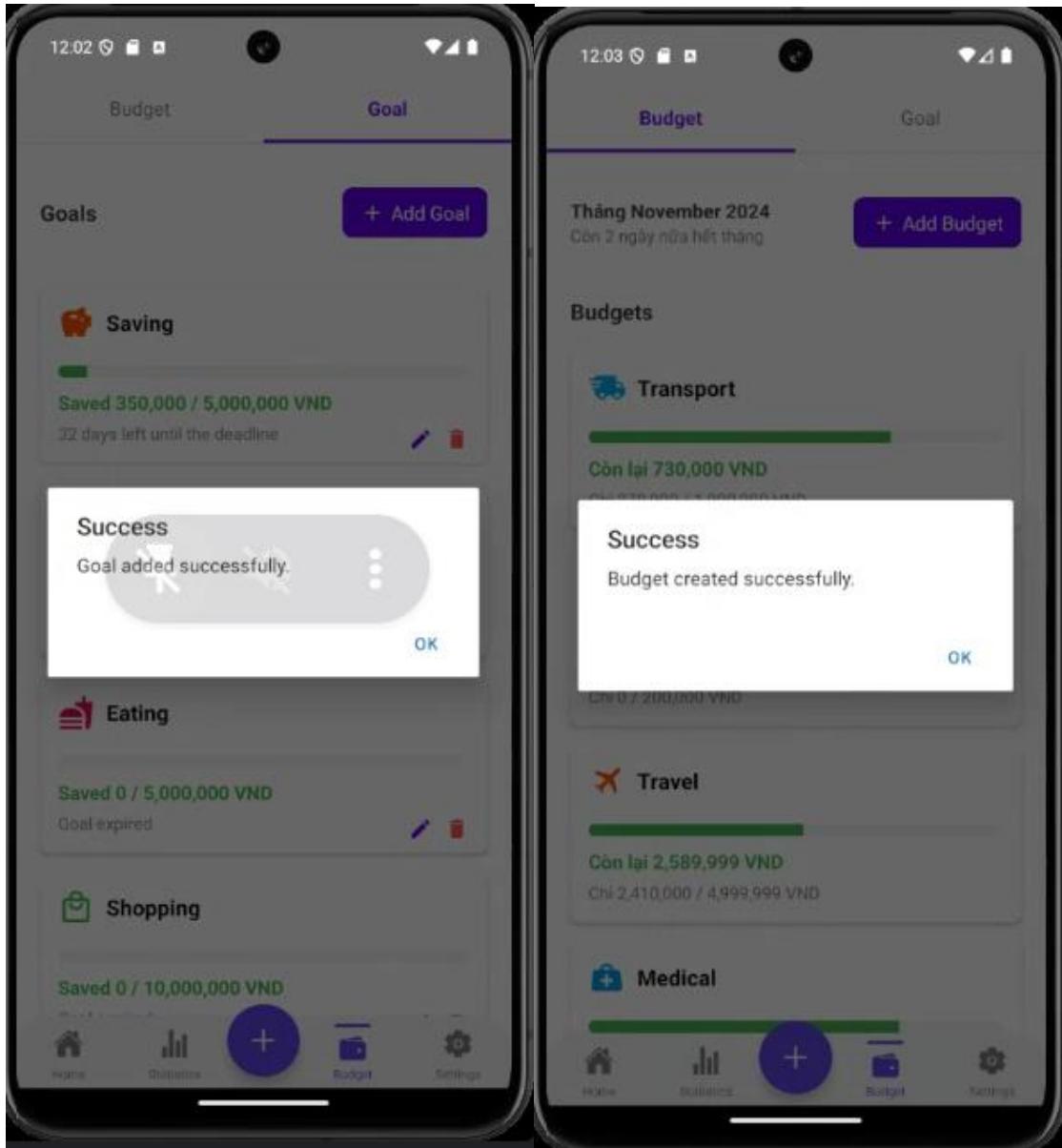
Người dùng có thể thực hiện tạo mới ngân sách và mục tiêu bằng Add budget và Add goal.

Chương 3: Phân tích thiết kế hệ thống



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

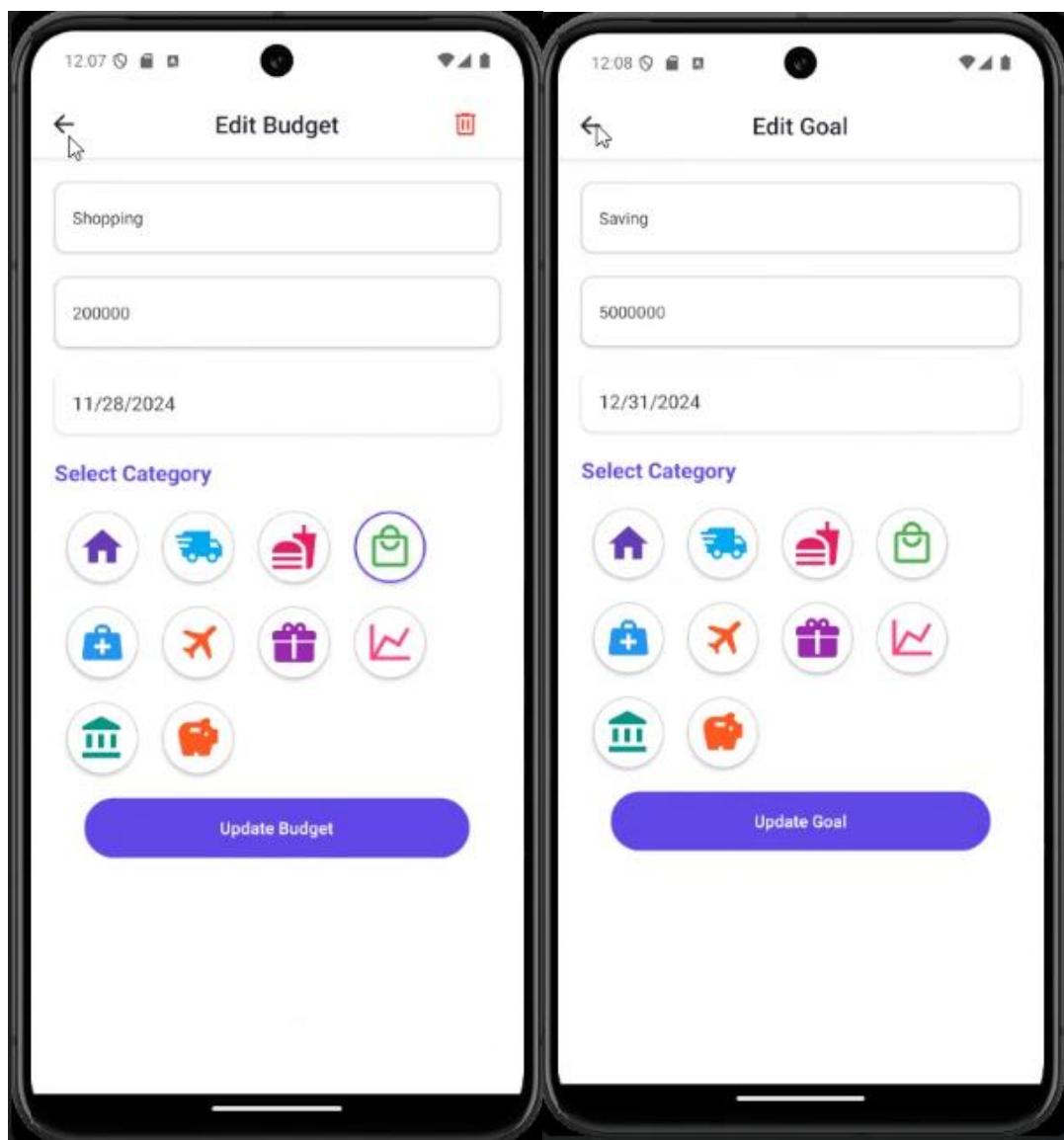
Chương 3: Phân tích thiết kế hệ thống



Thông báo thêm thành công hiển thị lên.

Ngoài ra để quản lý budget và goal người dùng có thể bấm vào nút chỉnh sửa và xóa đối với đối tượng.

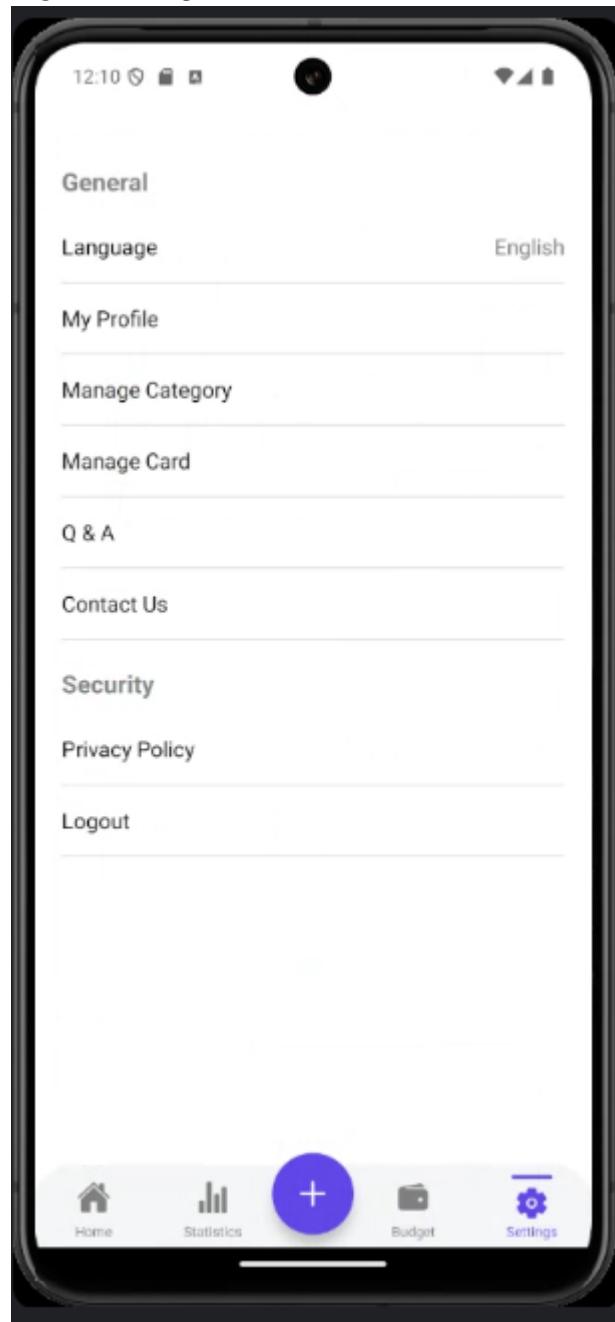
Chương 3: Phân tích thiết kế hệ thống



Sau khi cập nhật thì bấm vào nút Update.

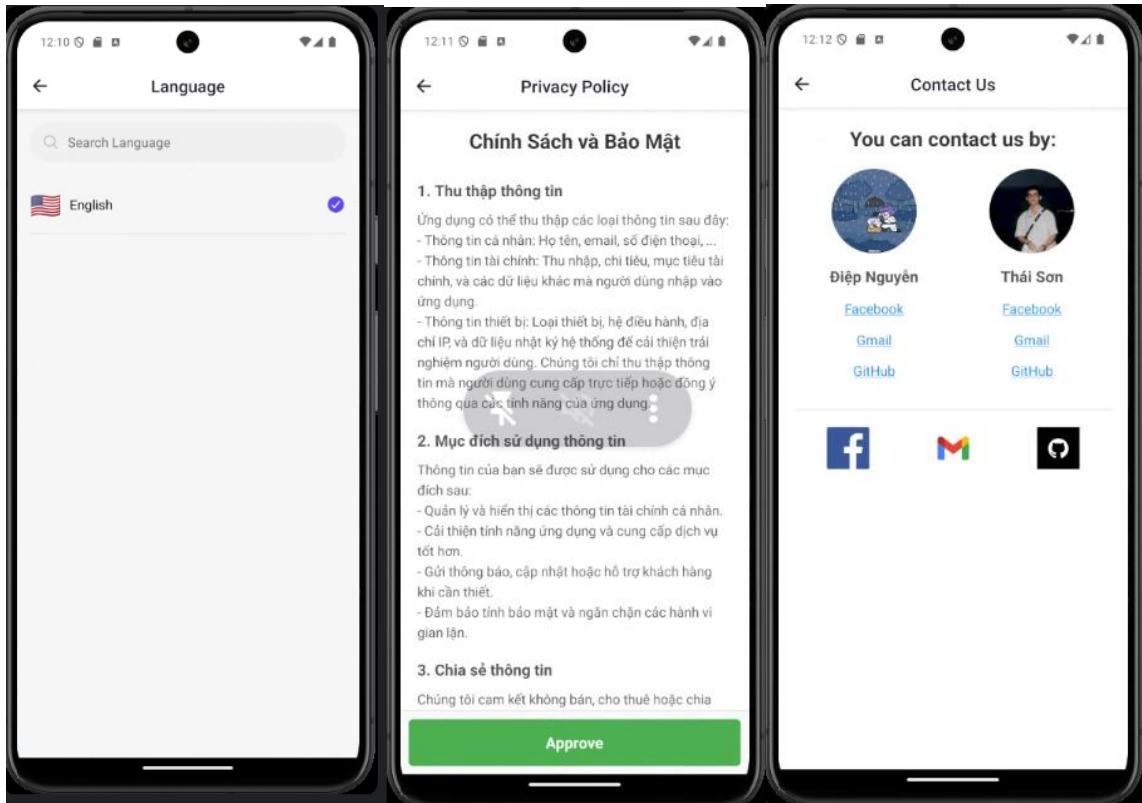
Chương 3: Phân tích thiết kế hệ thống

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



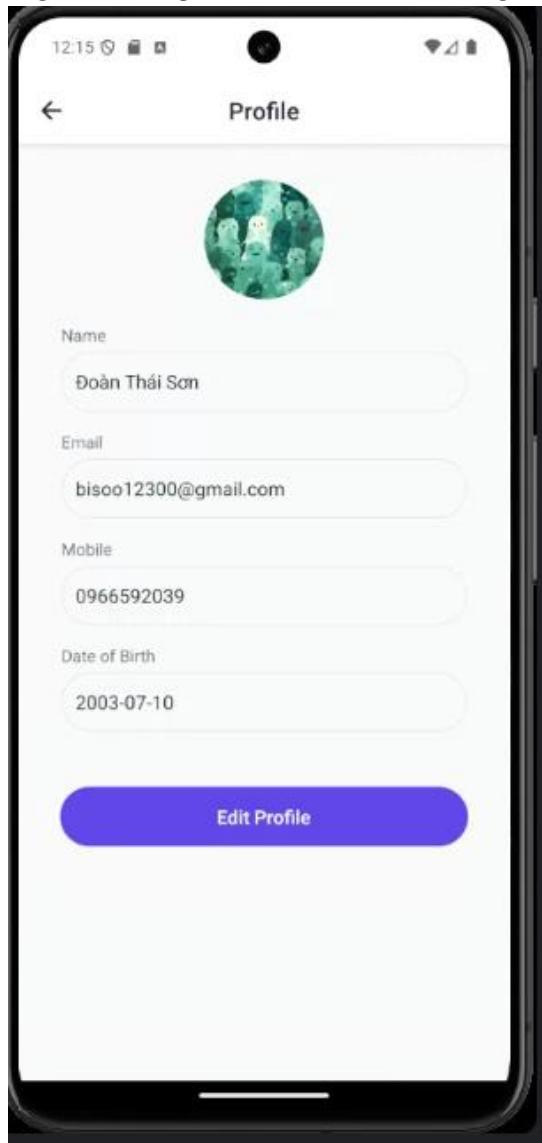
Chương 3: Phân tích thiết kế hệ thống

có thể cài đặt ngôn ngữ, xem chính sách và theo dõi chúng tôi.



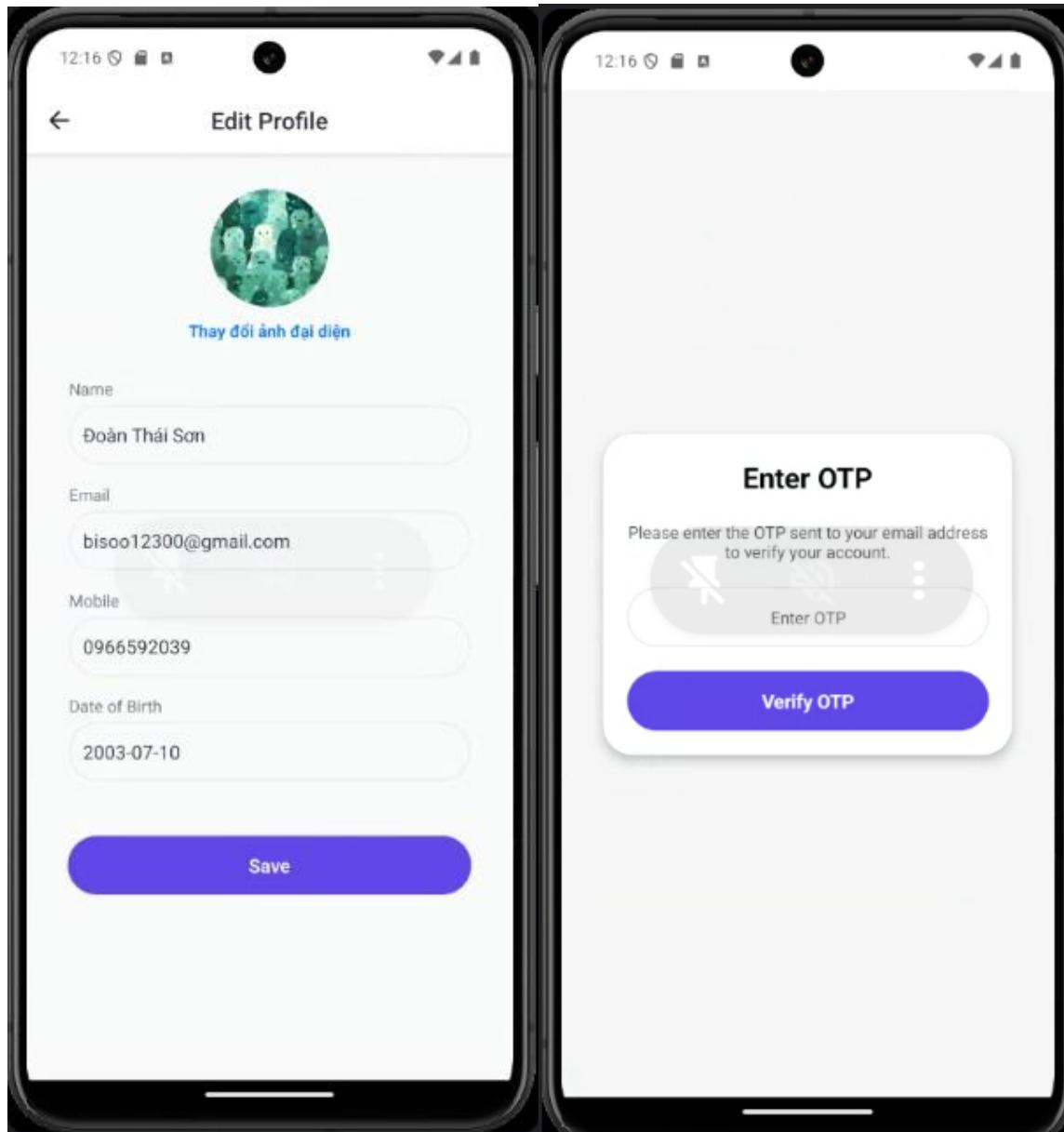
Chương 3: Phân tích thiết kế hệ thống

Người dùng còn có cài đặt thông tin người dùng bằng cách bấm vào My Profile:



Chương 3: Phân tích thiết kế hệ thống

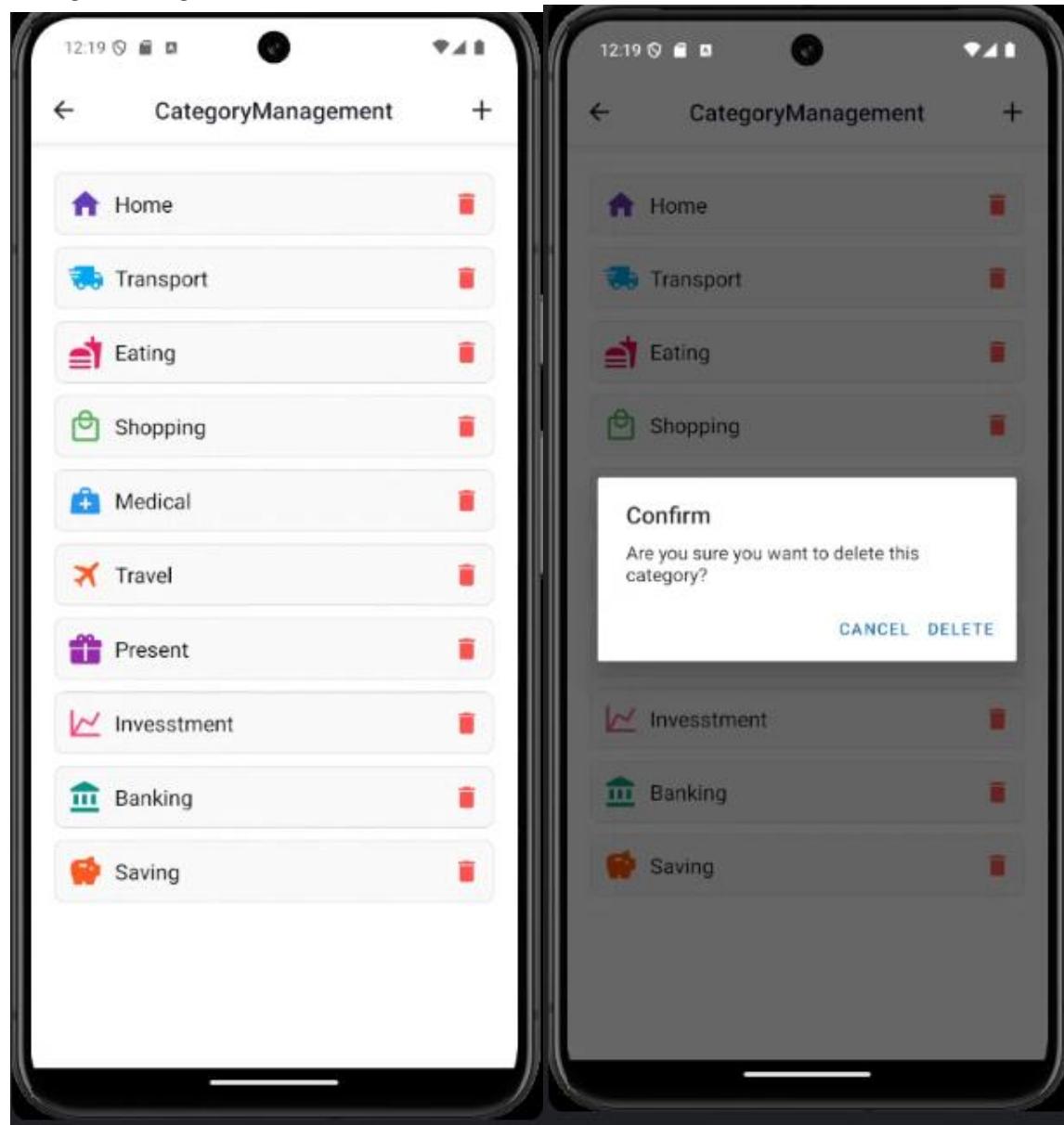
Để cập nhật Profile bấm vào Edit Profile



Sau khi thay đổi thông tin và bấm Save ứng dụng sẽ đưa người dùng đến trang xác thực OTP, nếu OTP được xác định là đúng thì sẽ cập nhật thông tin người dùng.

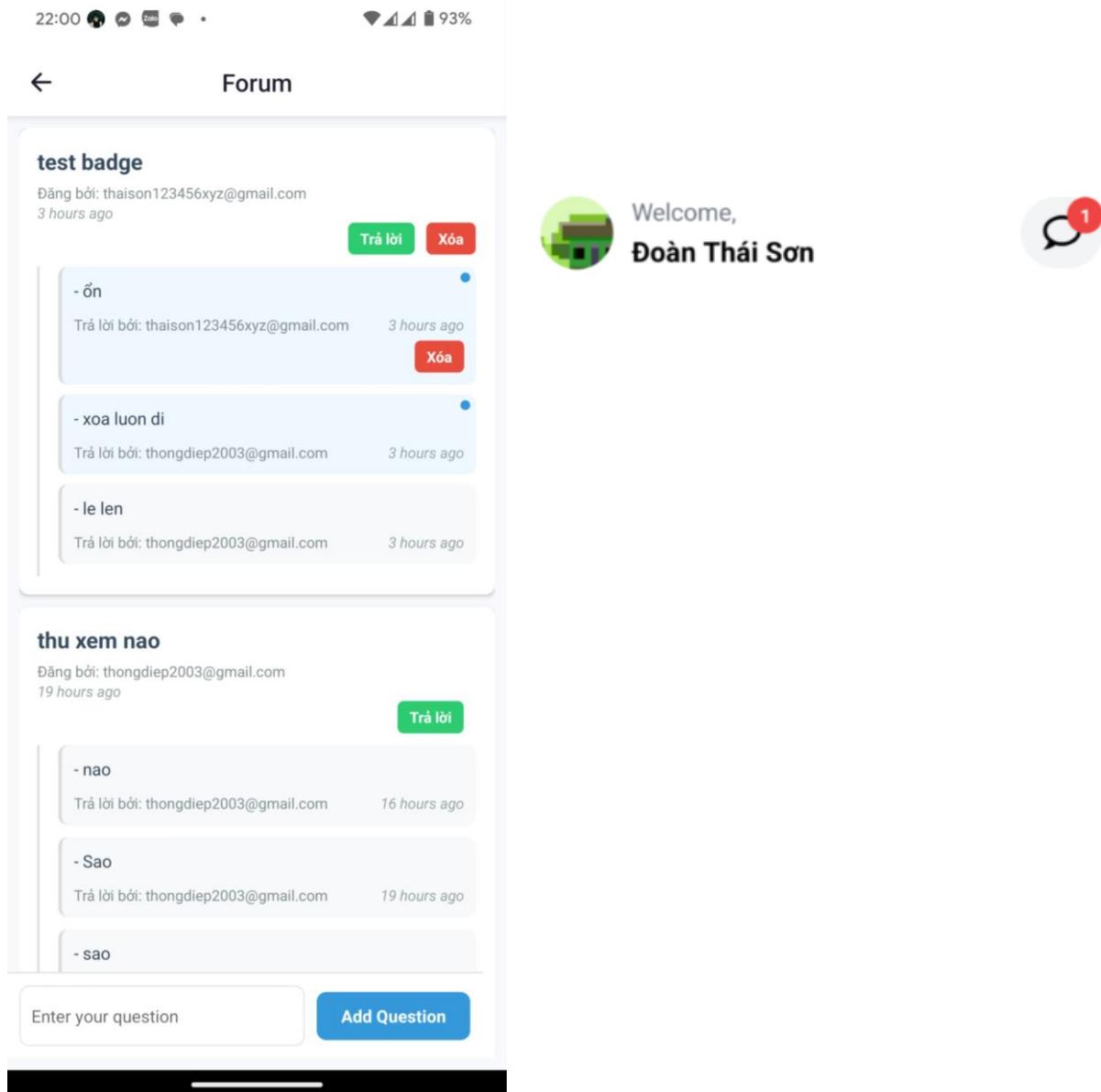
Chương 3: Phân tích thiết kế hệ thống

Người dùng có thể quản lý các Category bằng cách bấm vào Manage Category ở trang Setting.



Chương 3: Phân tích thiết kế hệ thống

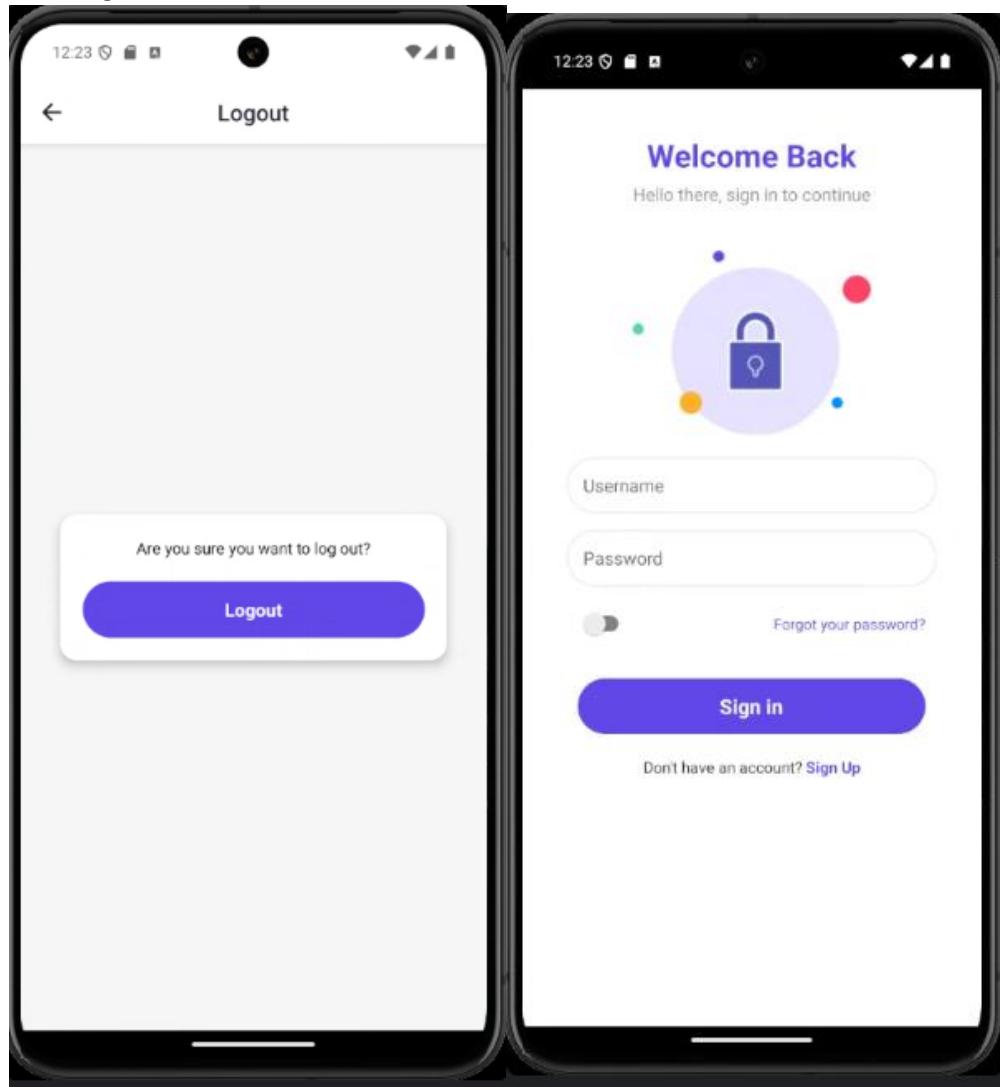
Để thực hiện tính năng hỏi đáp người dùng sẽ bấm vào biểu tượng chat trong trang Home



Người dùng sẽ được đưa đến Forum hỏi đáp, nơi các người dùng sẽ trao đổi bằng việc đặt câu hỏi và trả lời các câu hỏi của người khác, hoặc có thể xóa câu hỏi và xóa câu trả lời.

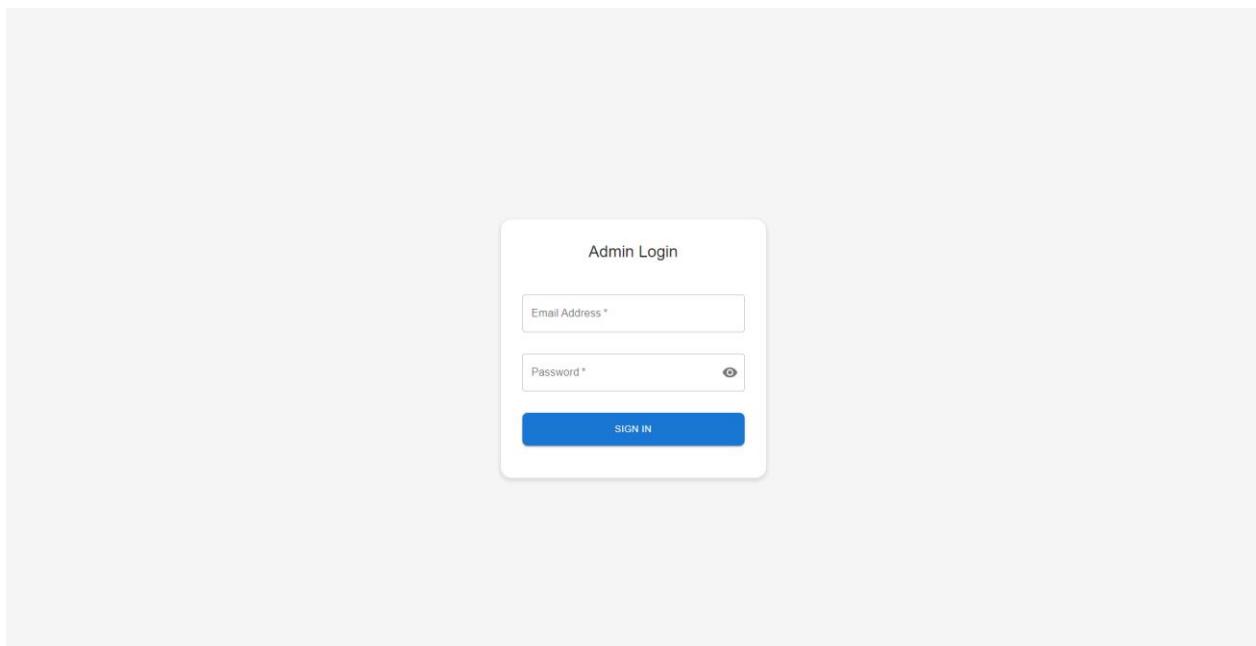
Chương 3: Phân tích thiết kế hệ thống

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



3.3.2. Giao diện quản trị viên:

Chương 3: Phân tích thiết kế hệ thống



Giao diện trang đăng nhập của Web Admin

A screenshot of the Admin Dashboard. The left sidebar has three navigation items: "Users", "Forum", and "Statistics". The main area shows a table of users. The columns are: Email, Name, Mobile, Status, Block Reason, and Actions. The table contains five rows of data. The first four rows have green "active" status buttons, while the fifth row has a red "blocked" button. The "Actions" column for the fifth row contains a checkmark icon. A search bar labeled "Search users" is located above the table. At the bottom left, there is a "Logout" link.

Trang quản lý người dùng

Chương 3: Phân tích thiết kế hệ thống

Admin Dashboard

Logout

Users

Forum

Statistics

Search users

Email	Name	Mobile	Status	Block Reason	Actions
thongdiep2003@gmail.com	Nguyen Hong Thong Diep	0392963132	active	-	
21110289@student.vnu.edu.vn	Thaison	0966592038	active	-	
thaison123456xyz@gmail.com	Thaison	0966592038	active	-	
bisoo12300@gmail.com	Bisoo	0966592038	blocked	-	
21110289@student.vnu.edu.vn	Thaison	0966592038	active	-	

Block User

User: thongdiep2003@gmail.com

Reason for blocking *

CANCEL BLOCK

Hộp thoại ghi nhận lí do ban tài khoản người dùng

Admin Dashboard

Logout

Users

Forum

Statistics

Search forums

Forum Management

Question	Posted By	Posted At	Replies	Actions
test	thaison123456xyz@gmail.com	12/7/2024, 1:37:56 AM	5	
tét2	thaison123456xyz@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	thaison123456xyz@gmail.com	12/7/2024, 6:43:32 PM	3	

Giao diện quản lí Forum

Chương 3: Phân tích thiết kế hệ thống

Admin Dashboard

Forum Management

Forum Post Details

Question: test
Posted by: thaison123456xyz@gmail.com | Date: 12/7/2024, 1:37:56 AM

Replies:

t nghĩ là nén dem trang này ra home page
By: thongdiep2003@gmail.com | Date: N/A | Status: Read

Qua ngon lướn
By: thaison123456xyz@gmail.com | Date: 12/7/2024, 2:21:33 AM | Status: Read

nice
By: thongdiep2003@gmail.com | Date: 12/7/2024, 2:30:22 AM | Status: Read

ma no khong thong bao da them thanh cong, clm :>
By: thongdiep2003@gmail.com | Date: 12/7/2024, 2:30:44 AM | Status: Read

Khong thong bao la tot
By: thaison123456xyz@gmail.com | Date: 12/7/2024, 2:31:12 AM | Status: Read

CLOSE

Logout

Khi Admin nhấp vào biểu tượng con mắt thì sẽ xem được chi tiết thông tin của câu hỏi đó và các câu trả lời của nó

Admin Dashboard

Statistics Dashboard

Total Users: 5

Forum Posts: 4

Total Replies: 15

User Status Distribution

Active Users (50%)
Blocked Users (40%)

Active Users : 3

Most Active Users

User	Posts	Replies	Total Activity
thongdiep2003@gmail.com	1	11	12
thaison123456xyz@gmail.com	3	4	7

Overall Activity Summary

Metric	Value
Average Posts per User	0.80
Average Replies per Post	3.75
User Engagement Rate	60.0%

Logout

Giao diện trang thống kê của Admin bao gồm các thông số:

- **Tổng quan về người dùng**
 - Total Users: Tổng số người dùng trong hệ thống.

Chương 3: Phân tích thiết kế hệ thống

- Active Users: Số người dùng đang hoạt động (không bị khóa).
- Blocked Users: Số người dùng bị khóa.
- Tổng quan về diễn đàn
 - Forum Posts: Tổng số bài viết trong diễn đàn.
 - Total Replies: Tổng số câu trả lời trong diễn đàn.
- Phân phối người dùng
 - User Status Distribution: Biểu đồ phân phối người dùng giữa trạng thái "Active" và "Blocked".
- Người dùng hoạt động nhiều nhất
 - Most Active Users: Bảng liệt kê top 5 người dùng hoạt động tích cực nhất, gồm các cột:
 - User: Email của người dùng.
 - Posts: Số bài viết của người dùng.
 - Replies: Số câu trả lời của người dùng.
 - Total Activity: Tổng số hoạt động (Posts + Replies).
- Tóm tắt hoạt động
 - Average Posts per User: Trung bình số bài viết trên mỗi người dùng.
 - Average Replies per Post: Trung bình số câu trả lời trên mỗi bài viết.
 - User Engagement Rate: Tỷ lệ người dùng đang hoạt động (Active Users / Total Users).

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.).

- 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.

Phần kết luận

- ✓ **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 gia 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

Phần kết luận

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: Cân 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

Thông 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.

Phần kết luận

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

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.

Tài liệu tham khảo

<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. - 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. - Tiến hành kiểm thử.

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			
4			
5			
6			
7			