

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH CÔNG NGHỆ THÔNG TIN

**Xây dựng ứng dụng ví điện tử HoangPay trên
nền tảng Mobile**

CBHD: Ths. Nguyễn Xuân Hoàng
Sinh viên: Phạm Quốc Hoàng
Mã sinh viên: 2020603110

Hà Nội – Năm 2024

Sau bốn năm học tập và rèn luyện tại trường Đại học Công nghiệp Hà Nội, đồ án tốt nghiệp là một dấu ấn quan trọng đánh dấu thời gian một sinh viên đã hoàn thành nhiệm vụ của mình trên ghế giảng đường Đại học. Đây chính là kết quả của sự nỗ lực của bản thân cùng sự giúp đỡ nhiệt tình của thầy cô, bạn bè. Qua đây, em cũng xin gửi lời cảm ơn chân thành tới những người đã giúp đỡ em hoàn thành được đồ án này.

Em xin gửi lời cảm ơn đến các thầy, cô khoa Công Nghệ Thông Tin – Trường Đại học Công nghiệp Hà Nội đã tận tình giảng dạy, truyền đạt những kiến thức quý giá của mình cho em cũng như các bạn sinh viên khác trong suốt những năm học qua. Đặc biệt, em xin gửi lời cảm ơn chân thành đến ***Ths. Nguyễn Xuân Hoàng***. Em cảm ơn thầy trong thời gian vừa qua đã hướng dẫn nhiệt tình, đôn đốc chúng em thực hiện đồ án tốt nghiệp. Nhờ có những lời góp ý, nhận xét của thầy nên đề tài tốt nghiệp: “Xây dựng ứng dụng ví điện tử HoangPay trên nền tảng Mobile” của em mới có thể hoàn thiện. Bước đầu đi vào thực tế còn hạn chế và nhiều bất ngờ nên không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp của thầy cô để em có thể hoàn thiện hơn đề tài của mình.

Em xin chân thành cảm!

Tôi cam đoan các nội dung được trình bày trong báo cáo đồ án này là kết quả làm việc của cá nhân tôi dưới sự hướng dẫn tận tình của Ths.Nguyễn Xuân Hoàng.

Các thông tin, tài liệu trích dẫn trong báo cáo đồ án đều được dẫn nguồn và có độ chính xác cao nhất trong phạm vi hiểu biết của tác giả.

Tôi xin hoàn toàn chịu trách nhiệm về bài báo cáo của mình.

Sinh viên thực hiện

Hoàng

Phạm Quốc Hoàng

Contents

MỞ ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG VÀ MÔI TRƯỜNG LẬP TRÌNH.....	2
1.1 Tổng quan về ngôn ngữ Dart	2
1.2 Framework Flutter	4
1.3 Nền tảng Firebase.....	5
1.4 Visual Studio Code.....	8
1.4.1 Giới thiệu.....	8
1.5 SQL Lite.....	9
1.5.1 Giới thiệu.....	9
1.5.2 Lợi ích khi sử dụng SQL Lite	10
1.5.3 Nhược điểm của SQL Lite	10
1.5.4 Tính năng của SQL lite	10
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	12
2.1 Mô tả bài toán	12
2.2 Tài liệu đặc tả yêu cầu người dùng	13
2.3 Phân tích Use Case.....	16
Use case này cho phép khách hàng đặt lại mật khẩu đã quên.....	18
2.5 Thiết kế cơ sở dữ liệu	35
CHƯƠNG 3: CÀI ĐẶT CHƯƠNG TRÌNH VÀ KẾT QUẢ ĐẠT ĐƯỢC	38
3.1 Công nghệ và môi trường sử dụng.....	38
3.2 Giao diện trang người dùng.....	38
3.3 Kiểm thử.....	52
KẾT LUẬN	53
DANH MỤC TÀI LIỆU THAM KHẢO.....	54

Danh Mục Hình Ảnh

Hình 1.1	Firebase	6
Hình 1.2	Logo phần mềm Visual Studio Code.....	8
Hình 2.1.	Biểu đồ use case của hệ thống	15
Hình 2.2.	Biểu đồ trình tự use case Đăng ký	25
Hình 2.3.	Biểu đồ trình tự use case “Đăng nhập”.....	26
Hình 2.4	Biểu đồ trình tự use case “Xem tin tức”	26
Hình 2.5	Biểu đồ trình tự use case “Quên mật khẩu”	27
Hình 2.6	Biểu đồ trình tự use case “Xem khuyến mại ”	28
Hình 2.7	Biểu đồ trình tự của use case “Xem thông báo”.....	29
Hình 2.8	Biểu đồ trình tự Use case “Tài khoản ví điện tử”.....	29
Hình 2.9	Biểu đồ trình tự use case “Xem chi tiết thẻ”.....	30
Hình 2.10	Biểu đồ trình tự use case “Cài đặt”.....	30
Hình 2.11	Biểu đồ trình tự use case “Nạp tiền”.....	31
Hình 2.12	Biểu đồ trình tự Use case “Quét QR”	32
Hình 2.13	Biểu đồ trình tự use case “Quản lý hóa đơn”	33
Hình 3.1	Giao diện màn hình giới thiệu.....	37
Hình 3.2	Giao diện trang chủ.....	38
Hình 3.3	Giao diện danh khuyến mại	39
Hình 3.4	Giao diện xem danh sách tin tức	40
Hình 3.5	Giao diện đăng nhập	41
Hình 3.6	Giao diện đăng kí tài khoản	43
Hình 3.7	Giao diện quên mật khẩu	45
Hình 3.8	Giao diện màn hình thông báo	46
Hình 3.9	Giao diện màn hình tài khoản ví điện tử	47
Hình 3. 10	Giao diện cài đặt	48
Hình 3. 11	Giao diện Nạp tiền	49
Hình 3. 12	Giao diện quản lý hóa đơn.....	50

Danh Mục Bảng Biểu

Bảng 2.1 Danh sách user case của hệ thống	13
Bảng 2.2 Yêu cầu phi chức năng của hệ thống	14
Bảng 2.3: Bảng User.....	34
Bảng 2.4: Bảng ThôngBao.....	34
Bảng 2.3: Bảng HoaDon	35
Bảng 2.4 Bảng TinTuc.....	36
Bảng 2.5 Bảng KhuyếnMai.....	37
Bảng 3.1. Bảng nội dung kiểm thử	51

1 MỞ ĐẦU

Ví điện tử là một trong những xu hướng tất yếu của nền kinh tế hiện đại. Với sự hỗ trợ của công nghệ, sự bùng nổ của các thiết bị kết nối Internet, doanh nghiệp ngày càng có cơ hội mở rộng kênh bán hàng, tiếp thị sản phẩm đến với người tiêu dùng. Các ứng dụng ví điện tử ra đời góp phần quan trọng và to lớn trong việc thúc đẩy kinh doanh phát triển nhanh chóng, các sản phẩm, dịch vụ đến gần với tay khách hàng hơn giúp việc thanh toán khi mua hàng ngày càng dễ dàng, thuận tiện. Đặc biệt các ứng dụng ví điện tử càng thu hút khách hàng và được xây dựng nhiều hơn.

Nhận thức được tầm quan trọng của ứng dụng ví điện tử, em đã quyết định thực hiện đề tài “**Xây dựng ứng dụng ví điện tử HoangPay trên nền tảng Mobile**” cho đồ án tốt nghiệp lần này. Đề tài hướng đến xây dựng một ứng dụng ví điện tử có đầy đủ chức năng cần thiết mà một ứng dụng ví điện tử thanh toán trực tuyến cần như: đăng nhập, đăng ký, chuyển tiền, liên kết,... Trong quá trình thực hiện đồ án, mặc dù đã cố gắng hết sức nhưng cũng không tránh khỏi những thiếu sót, hạn chế. Em rất mong nhận được lời nhận xét của quý thầy cô và các bạn để đồ án của em được hoàn thiện hơn.

Bài báo cáo bao gồm 3 nội dung chính:

- Chương 1: Tổng quan về công nghệ sử dụng và môi trường lập trình
- Chương 2: Phân tích thiết kế hệ thống
- Chương 3: Cài đặt chương trình và kết quả đạt được

Sinh viên thực hiện

Phạm Quốc Hoàng

CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG VÀ MÔI TRƯỜNG LẬP TRÌNH

1.1 Tổng quan về ngôn ngữ Dart

1.1.1 Giới thiệu

Ngôn ngữ Dart là một ngôn ngữ lập trình hướng đối tượng mã nguồn mở, được Google phát triển vào năm 2011. Đặc trưng của Dart là cú pháp kiểu C, và nó hỗ trợ giao diện, lớp, và các tính năng khác của lập trình hướng đối tượng.

Dart không chỉ dừng lại ở việc phát triển ứng dụng cho một nền tảng, mà còn mở ra cánh cửa cho sự sáng tạo trên cả hai nền tảng quan trọng: Android và iOS.

Dart sử dụng chủ yếu để phát triển ứng dụng web và di động. Dart được thiết kế để hỗ trợ xây dựng ứng dụng có hiệu suất cao và dễ bảo trì. Đặc điểm của Dart bao gồm cú pháp đơn giản, hướng đối tượng mạnh mẽ, khả năng xử lý bất đồng bộ và công cụ phong phú hỗ trợ cho việc lập trình. Dart có thể được sử dụng để lập trình backend cho web, thông qua việc sử dụng framework như Dart's Aqueduct hoặc Angel.

1.1.2 Ưu điểm chính của Dart

Dart hiện đang được nhiều người lựa chọn để thiết kế ứng dụng di động nhờ những ưu điểm sau:

- *Dart có cú pháp dễ đọc và viết*

Cú pháp của Dart tương tự với nhiều ngôn ngữ lập trình phổ biến khác như Java, C#. Viết mã nguồn Dart dễ dàng và ít lỗi hơn so với JavaScript. Nó dễ dàng hơn và nhiều người mới bắt đầu thấy sử dụng ngôn ngữ này quen thuộc không mấy khó khăn khi lập trình.

- *Dart miễn phí*

Dart là một ngôn ngữ mã nguồn mở nên việc cài đặt và tùy biến Dart là miễn phí và tự do. Là một công nghệ mã nguồn mở, Dart có thể được các tổ chức sử dụng mà không phải trả bất kỳ khoản phí nào cho giấy phép hay phí bản quyền nào.

- *Dart hiệu suất cao*

Dart được biên dịch trước khi chạy, giúp tối ưu hóa hiệu suất và tốc độ thực thi. Mã nguồn Dart chạy nhanh hơn mã JavaScript trong nhiều trường hợp. Hỗ trợ cả biên dịch Vừa đúng lúc (JIT) và biên dịch Trước thời hạn (AOT).

- *Cộng đồng Dart mạnh mẽ*

Là một ngôn ngữ mã nguồn mở cùng với sự phổ biến của Dart thì cộng đồng Dart được coi là khá lớn và khá chất lượng.

Với cộng đồng phát triển lớn, việc cập nhật các bản vá lỗi phiên bản hiện tại cũng như thử nghiệm các phiên bản mới khiến Dart rất linh hoạt trong việc hoàn thiện mình.

- *Hỗ trợ đa nền tảng*

Dart không chỉ hỗ trợ phát triển ứng dụng web, mà còn cho phép xây dựng ứng dụng di động (sử dụng Flutter). Flutter cho phép bạn phát triển ứng dụng đa nền tảng với một mã nguồn duy nhất.

Do những ưu điểm trên, em đã quyết định lựa chọn Dart là ngôn ngữ chính để xây dựng ứng dụng của mình

1.2 Framework Flutter

1.2.1 Giới thiệu

Flutter là mobile UI framework của Google để tạo ra các giao diện chất lượng cao trên iOS và Android trong khoảng thời gian ngắn. Flutter hoạt động với những code sẵn có được sử dụng bởi các lập trình viên, các tổ chức. Flutter hoàn toàn miễn phí và cũng là mã nguồn mở.

Flutter sẽ là một phương pháp tốt để thay thế phát triển ứng dụng android. Một framework dựa trên ngôn ngữ lập trình Dart cho phép thiết kế, xây dựng giao diện người dùng đẹp mắt và dễ sử dụng.

1.2.2 Ưu điểm của Framework Flutter

- *Đa nền tảng*

Flutter cho phép bạn phát triển ứng dụng cho nhiều nền tảng như iOS, Android, web và desktop từ một mã nguồn duy nhất. Điều này giúp giảm thời gian và công sức cần thiết cho việc phát triển ứng dụng trên nhiều nền tảng khác nhau. Với một vài điều chỉnh nhỏ một ứng dụng có thể chạy trên cả Android và iOS.

- *Hot Reload*

Hot Reload là tính năng mạnh mẽ của flutter cho phép người dùng thực hiện các thay đổi trong mã nguồn và xem xét kết quả ngay lập tức trên thiết bị mô phỏng hoặc thiết bị thực tế chạy ứng dụng mà không cần phải rebuild hoặc restart ứng dụng.

Ngoài ra Hot Reload còn giúp bạn thêm các tính năng, fix bug tiết kiệm thời gian hơn mà không cần phải thông qua máy ảo, máy android hoặc iOS.

- *Hiệu suất tốt*

Với việc sử dụng Dart và việc biên dịch trực tiếp thành mã máy, Flutter cung cấp hiệu suất cao và đáp ứng tốt, giúp ứng dụng chạy mượt mà và không bị giật lag.

Có rất nhiều các thành phần để xây dựng giao diện của Flutter vô cùng đẹp mắt theo phong cách Material Design và Cupertino, Hỗ trợ nhiều các APIs chuyển động, smooth scrolling, ...

- *Cộng đồng hỗ trợ lớn và mạnh mẽ*

Flutter có một cộng đồng lớn với nhiều tài nguyên, công cụ và thư viện phong phú. Điều này giúp bạn dễ dàng tìm kiếm hỗ trợ và giải pháp cho các vấn đề trong quá trình phát triển.

1.2.3 Nhược điểm của Framework *Flutter*

Ứng dụng Flutter có thể có kích thước lớn hơn so với các ứng dụng được phát triển bằng các công nghệ khác.

Mặc dù Flutter có hiệu suất tốt trên hầu hết các thiết bị di động hiện đại, Nhưng có thể gặp vấn đề về hiệu suất trên các thiết bị cũ hoặc tài nguyên thiết bị có hạn. Và nếu không tối ưu tốt cho ứng dụng thì ứng dụng viết bằng flutter có thể bị chậm và lag khi sử dụng.

Do là một công nghệ mới với tuổi đời trẻ lên nó vẫn có một vài hạn chế với thư viện hỗ trợ từ bên thứ ba của nó vẫn chưa được phát triển đầy đủ đa dạng như các nền tảng khác như Android và IOS.

1.3 Nền tảng Firebase

1.3.1 Giới thiệu

Firebase là nền tảng phát triển ứng dụng di động và web do Google phát triển, cung cấp rất nhiều dịch vụ đám mây cho việc xây dựng, triển khai và quản lý các ứng dụng. Nó được giới thiệu vào năm 2011 và được giới thiệu vào năm 2012 dưới dạng Backend-as-a-Service với chức năng thời gian thực. Sau khi được Google mua lại vào năm 2014 thì Firebase nhanh chóng phát triển thành nền tảng phát triển ứng dụng đa năng của di động và website như ngày nay.

Nền tảng này là sự kết hợp giữa cloud với hệ thống máy chủ của Google để tập trung chính cho 2 đối tượng chính:

- Develop & test your app: Phát triển và thử nghiệm các ứng dụng được thiết kế.
- Grow & engage your audience: Phân tích dữ liệu và tối ưu hóa trải nghiệm với người dùng.

Nó cung cấp cho chúng ta nhiều API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database, bởi vậy giờ đây chúng ta chỉ cần gọi API và phần server của Firebas sẽ lo phần còn lại.



Hình 1.2.1. Firebase

1.3.2 Đặc điểm

- Firebase tích hợp nhiều dịch vụ khác nhau như: Realtime Database, Cloud Firestore, Authentication, Storage, Cloud Functions, Analytics và nhiều dịch vụ khác
- Firebase hỗ trợ thời gian thực cho việc đồng bộ dữ liệu giữa các thiết bị và người dùng Realtime Database, Cloud Firestore cho phép dữ liệu được đồng bộ ngay lập tức khi có sự thay đổi
- Nó được thiết kế dễ dàng sử dụng và tiện lợi cho nhà phát triển. Nó cung cấp các SDK, API cho nhiều ngôn ngữ lập trình, nền tảng. Điều này làm cho việc tích hợp và sử dụng trở nên dễ dàng và thuận tiện. Bạn có thể sử dụng các dịch vụ của nó một cách linh hoạt và mở rộng chúng khi cần thiết phù hợp với nhu cầu của người sử dụng
- Firebase cung cấp các công cụ bảo mật mạnh mẽ như xác thực người dùng, quản lý quyền truy cập và mã xác thực giúp bảo vệ dữ liệu và thông tin cá nhân một cách mạnh mẽ, hiệu quả, linh hoạt.

1.3.3 Những lợi ích khi sử dụng Firebase

- *Dễ dàng bắt đầu và tích hợp*
 - Firebase cung cấp các SDK và công cụ mạnh mẽ cho nhiều nền tảng, giúp cho việc bắt đầu phát triển ứng dụng nhanh chóng và dễ dàng. SDK được cung cấp cho nhiều ngôn ngữ lập trình khác nhau.
 - Nó cung cấp cho người dùng nhiều chức năng như phân tích, cơ sở

dữ liệu, báo cáo hoạt động, và báo cáo sự cố lỗi để bạn nó thể dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm mang lại trải nghiệm tốt nhất

- *Hiệu suất cao và độ tin cậy*
 - Firebase có độ tin cậy cao được đảm bảo từ Google, nó được Google hỗ trợ và cung cấp trên nền tảng phần cứng trên quy mô rộng khắp thế giới, được các tập đoàn lớn và các ứng dụng với hàng triệu lượt sử dụng từ người dùng.
- *Quản lý dữ liệu dễ dàng*
 - Việc quản lý và trải nghiệm các ứng dụng của Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ với nhau.
 - Với Firebase Analytics cung cấp các công cụ phân tích mạnh mẽ hiểu rõ hơn về người dùng của mình và cách họ tương tác với ứng dụng giúp tối ưu hóa trải nghiệm người dùng và tăng hiệu quả của ứng dụng

1.3.4 Nhược điểm khi sử dụng Firebase

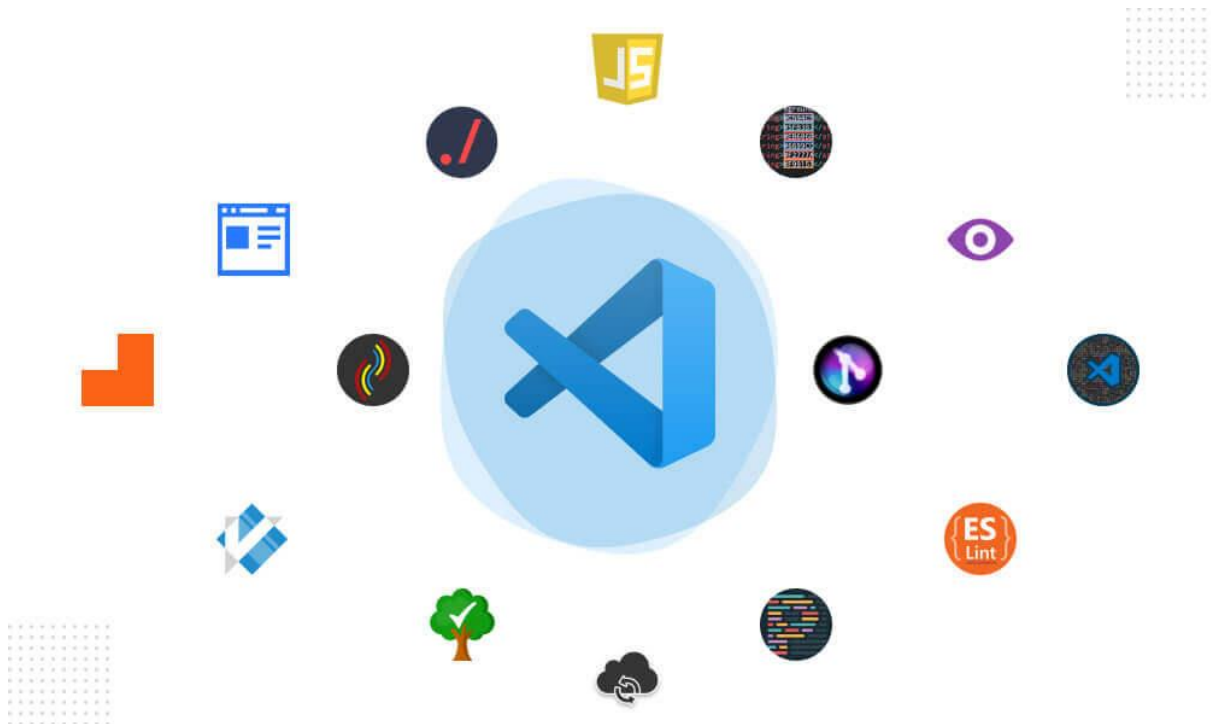
- Firebase không là mã nguồn mở
- Người dùng không truy cập được mã nguồn
- Nền tảng không hoạt động với nhiều quốc gia
- Firebase chỉ sử dụng JSON và hầu như không có SQL
- Firebase chỉ chạy trên Google Cloud
- Truy vấn khá chậm
- Các dịch vụ cung cấp không phải đều miễn phí
- Giá sử dụng dịch vụ Firebase khá cao
- Thiếu hợp đồng doanh nghiệp
- Không cung cấp API GraphQL

1.3.5 Lý do chọn Firebase cho dự án.

- Dễ dàng tích hợp với ứng dụng Flutter sử dụng Dart viết mã sử dụng các tính năng mà FireBase cung cấp chỉ với những bước đơn giản không phức tạp.
- Cung cấp nhiều tính năng mạnh mẽ như: Firebase Authentication, Cloud Firestore, Realtime Database, Cloud Storage, Cloud Functions, Remote

- ## 1.4 Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở và miễn phí được phát triển bởi Microsoft. Nó là một trong những công cụ phổ biến nhất được sử dụng cho việc phát triển phần mềm trên nhiều nền tảng như Windows, macOS và Linux. Không tốn nhiều dung lượng ổ cứng nhưng lại cực kì thông minh, giúp bạn lập trình, vá lỗi, debug, suggest đầy đủ, hiệu quả và nhanh chóng.



Hình 1.4.1. Logo phần mềm Visual Studio Code

- Dễ sử dụng, tùy biến, mở rộng

VS Code có giao diện sử dụng thân thiện và dễ sử dụng. Nó cung cấp

một loạt các tính năng như gợi ý mã, kiểm tra cú pháp, đánh dấu cú pháp và nhiều tính năng khác để người dùng có thể viết mã một cách nhanh chóng, thuận tiện và hiệu quả. Nó có thể tùy chỉnh giao diện và cái đặt tùy ý phù hợp với từng cá nhân sử dụng.

VS Code có một hệ thống mở rộng mạnh mẽ cho phép bạn cài đặt các extension để mở rộng chức năng của nó. Các extension có thể cung cấp tính năng như gợi ý mã, kiểm tra lỗi, quản lý dự án, tích hợp với các công cụ như Docker và Kubernetes, và nhiều tính năng khác.

- Hỗ trợ nhiều ngôn ngữ lập trình

VS Code hỗ trợ nhiều ngôn ngữ lập trình khác nhau như: JavaScript, TypeScript, Python, C#, Java, Điều này rất thuận tiện cho việc phát triển ứng dụng trên nhiều nền tảng và ngôn ngữ một cách dễ dàng

- Tích hợp Git và hỗ trợ Debugging.

VS Code được tích hợp sẵn Git, cho phép người dùng quản lý phiên bản mã nguồn của dự án một cách dễ dàng trực tiếp từ giao diện của nó hoặc trên terminal tích hợp sẵn trong ứng dụng.

VS Code có tính năng debugging tích hợp cho nhiều ngôn ngữ lập trình như JavaScript, TypeScript, Python, C#, và nhiều ngôn ngữ khác. Bạn có thể dễ dàng thực hiện các hoạt động như set breakpoints, xem giá trị biến, và xem stack trace.

1.5 SQL Lite

1.5.1 Giới thiệu

SQLite là hệ quản trị cơ sở dữ liệu (DBMS) quan hệ tương tự như Mysql, ... Đặc điểm nổi bật của SQLite so với các DBMS khác là gọn, nhẹ, đơn giản, đặt biệt không cần mô hình server-client, không cần cài đặt, cấu hình hay khởi động nên không có khái niệm user, password hay quyền hạn trong SQLite Database. Dữ liệu cũng được lưu ở một file duy nhất.

SQLite thường không được sử dụng với các hệ thống lớn nhưng với những hệ thống ở quy mô vừa và nhỏ thì SQLite không thua các DBMS khác về chức năng hay tốc độ. Vì không cần cài đặt hay cấu hình nên SQLite được sử dụng nhiều trong việc phát triển, thử nghiệm ... vì tránh được những rắc rối trong quá trình cài đặt.

1.5.2 Lợi ích khi sử dụng SQL Lite

Việc sử dụng SQLite mang lại nhiều lợi ích đáng kể, việc sử dụng SQLite mang lại nhiều lợi ích cho các ứng dụng nhỏ, di động và nhúng bao gồm tính đơn giản, dung lượng nhỏ, khả năng tương thích rộng rãi và hiệu suất tốt., Những lợi ích đó bao gồm:

- **Dễ Sử Dụng:** SQLite dễ triển khai và sử dụng. Bạn chỉ cần thêm thư viện SQLite vào ứng dụng của mình và bắt đầu làm việc với cơ sở dữ liệu mà không cần cấu hình máy chủ phức tạp.
- **Không Đòi Hỏi Máy Chủ:** Vì không cần một máy chủ riêng biệt, SQLite là một lựa chọn phù hợp cho các ứng dụng nhỏ, di động và nhúng.
- **Dung Lượng Nhỏ:** Thư viện SQLite có dung lượng nhỏ, giúp giảm thiểu tài nguyên hệ thống và chiếm ít không gian lưu trữ.
- **Tương Thích Rộng Rãi:** SQLite có thể làm việc trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux, Android và iOS, làm cho nó trở thành một giải pháp linh hoạt cho các ứng dụng đa nền tảng.
- **Hiệu Suất Tốt:** SQLite cung cấp hiệu suất tốt cho các ứng dụng với tải trọng nhỏ đến trung bình, với việc hỗ trợ một loạt các tùy chọn tối ưu hóa.
- **Hỗ Trợ Đầy Đủ SQL:** SQLite hỗ trợ hầu hết các câu lệnh SQL chuẩn, cho phép bạn thực hiện các thao tác như truy vấn, cập nhật và xóa dữ liệu một cách linh hoạt.
- **Khả Năng Bảo Mật:** SQLite hỗ trợ mã hóa cơ sở dữ liệu, cung cấp tính năng bảo mật cho dữ liệu nhạy cảm.

1.5.3 Nhược điểm của SQL Lite

- Do sử dụng cơ chế coarse-grained locking nên trong cùng một thời điểm SQLite có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có một người có thể ghi dữ liệu.
- SQLite không phải là lựa chọn hoàn hảo để đáp ứng các nhu cầu xử lý trên một khối lượng dữ liệu lớn, phát sinh liên tục.

1.5.4 Tính năng của SQL lite

- Giao dịch trong SQLite tuân thủ theo nguyên tắc (ACID) ngay cả sau hi hệ thống treo và mất điện.
- Không cấu hình:
 - Không cần thiết lập hoặc quản trị
- SQLite hỗ trợ với đầy đủ tính năng với các khả năng nâng cao như các chỉ mục 1 phần, các chỉ mục về các biểu thức, JSON và các biểu thức bảng chung.
- Một sở dữ liệu hoàn chỉnh được lưu trữ trong một tệp đa nền tảng duy nhất. Phù hợp với sử dụng dưới dạng định dạng tệp ứng dụng
- Hỗ trợ các cơ sở dữ liệu có kích thước terabyte và các chuỗi có kích thước gigabyte.
- API
 - Đơn giản để sử dụng
 - Nhanh: Trong một số trường hợp, SQLite nhanh hơn hệ thống tệp tin trực tiếp I/O.
- Bindings cho hàng chục ngôn ngữ khác có sẵn 1 cách riêng biệt.
- Mã nguồn đầy, nguồn mở đủ có thể kiểm tra nhánh 100%.
- Nền tảng đa nền tảng:
 - SQLite là có sẵn trên Android, *BSD, iOS, Linux, Mac, Solaris, Windows,.. Dễ dàng dịch chuyển sang các hệ thống khác.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1 Mô tả bài toán

2.1.1 Giới thiệu đề tài

Đề tài “Xây dựng ứng dụng ví điện tử HoangPay trên nền tảng Mobile” hướng đến xây dựng ứng dụng ví điện tử có các chức năng cần thiết của một ứng dụng ví điện tử thông thường.

Sử dụng ngôn ngữ chính là Dart, Framework Flutter, Firebase và SQL lite để xây dựng ứng dụng.

2.1.2. Khảo sát

Tiến hành khảo sát, tìm hiểu cách hoạt động, các chức năng của các ứng dụng ví điện tử như Mobile Money, Viettelpay, để đưa ra các chức năng phù hợp cho ứng dụng.

Ứng dụng có một số chức năng sau:

- Màn hình giới thiệu trước khi vào ứng dụng.
- Màn hình đăng nhập với tên đăng nhập, mật khẩu, quên mật khẩu, đăng ký.

- Màn hình thông báo chứa các thông báo từ hệ thống: gồm tiêu đề, thời gian, nội dung, ...
- Màn hình trang chủ bao gồm hiển thị thông báo chưa đọc, lịch nhắc hóa đơn, các dịch vụ yêu thích, khuyến mãi, tin tức.
- Màn hình quản lý hóa đơn, hiển thị thông tin về hóa đơn
- Màn hình khuyến mãi hiển thị thông tin khuyến mại.
- Màn hình tài khoản ví điện tử hiển thị thông tin về số dư
- Màn hình quét mã QR quét mã QR và tạo mã QR

Màn hình xem chi tiết thẻ đã liên kết: Tên thẻ, tên chủ thẻ, số thẻ, ngày hiệu lực và nạp tiền vào ví.

2.2 Tài liệu đặc tả yêu cầu người dùng

2.2.1 Yêu cầu chức năng

Bảng 2.1 Danh sách user case của hệ thống

Tác nhân (Actor)	User case	Mô tả
Khách	Đăng ký	Cho phép khách hàng đăng ký tài khoản
	Đăng nhập	Cho phép khách hàng đăng nhập vào hệ thống sử dụng các chức năng của ứng dụng
	Xem khuyến mại	Cho phép khách hàng xem các khuyến mại khi sử dụng ứng dụng.
	Xem tin tức	Cho phép khách hàng xem những tin tức có liên quan.
	Thông báo	Cho phép người dùng xem những thông báo từ hệ thống gửi về.

Hàng	Quản lý hóa đơn	Cho phép người dùng quản lý hóa đơn khi thanh toán khi sử dụng ứng dụng
	Tài khoản ví điện tử	Cho phép người dùng xem thông tin tài khoản, số dư cũng như một vài thông số khác
	Quét mã QR	Cho phép người dùng quét mã QR để thực hiện các giao dịch.
	Quên mật khẩu	Cho phép khách hàng đặt lại mật khẩu khi mà người dùng quên.
	Xem chi tiết thẻ	Cho phép người dùng xem chi tiết thẻ ngân hàng đã liên kết với ví
	Nạp tiền	Cho phép người dùng nạp tiền vào ví.
	Cài đặt	Cho phép người dùng thực hiện chức năng nâng cao và những tính chỉnh khác.

2.2.2 Yêu cầu phi chức năng

Bảng 2.2 Yêu cầu phi chức năng của hệ thống

Mục	Tên yêu cầu	Mô tả yêu cầu
1	Giao diện	Giao diện hệ thống phải dễ sử dụng, thân thiện với người dùng.
2	Tốc độ xử lý	Hệ thống phải xử lý nhanh và chính xác
3	Bảo mật	Tính bảo mật và độ an toàn cao
4	Tính tương thích	Tương thích với đa trình duyệt hiện tại

2.2.3 Biểu đồ các use case của hệ thống



Hình 2.1. Biểu đồ use case của hệ thống

2.3 Phân tích Use Case

2.3.1 Mô tả chi tiết các Use Case

2.3.1.1 Use case “Đăng ký”

Mô tả vắn tắt:

Use case này cho phép khách hàng tạo tài khoản mới.

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case này bắt đầu khi khách hàng nhấn nút “Đăng ký”. Hệ thống sẽ hiển thị form nhập thông tin đăng ký và người dùng nhập số điện thoại và điều khoản sử dụng.
- 2) Khách hàng nhập số điện thoại và mã OTP hệ thống gửi về số điện thoại đó. Rồi người dùng nhập họ và tên của mình rồi nhấn nút tiếp theo.
- 3) Khách hàng nhập mật khẩu và nhập lại mật khẩu theo đúng quy định rồi nhấn nút tiếp theo.
- 4) Khách hàng nếu nhập đúng hệ thống sẽ kích hoạt tài khoản đó và chuyển đến trang đăng ký thành công để bắt đầu đăng nhập. Use case kết thúc.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
- 2) Tại bước 2 trong luồng cơ bản nếu khách hàng nhập thông tin số điện thoại không chính xác thì hệ thống sẽ thông báo. Use case kết thúc.

Tiền điều kiện: Không có.

Hậu điều kiện: Khách hàng đăng kí thành công tài khoản

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.2 Use case “Đăng nhập”

Mô tả vắn tắt:

Use case này cho phép người dùng khách hàng đăng nhập vào hệ thống

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case này bắt đầu khi người dùng đã qua màn hình giới thiệu trước khi sử dụng app. Hệ thống sẽ hiển thị Màn hình đăng nhập nhập tên đăng nhập và mật khẩu.
- 2) Người dùng nhập tên đăng nhập và mật khẩu sau đó ấn nút “Đăng nhập”. Hệ thống kiểm tra tên đăng nhập, mật khẩu và quyền trong cơ sở dữ liệu để hiển thị giao diện tương ứng cho người dùng đó. Use case kết thúc.

Luồng rẽ nhánh:

- 1) Tại bước 2 ở luồng cơ bản nếu đăng nhập sai thông tin tài khoản hệ thống hiển thị thông báo “Tên đăng nhập hoặc mật khẩu không tồn tại”. Use case kết thúc.
- 2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Người dùng chưa đăng nhập vào hệ thống.

Hậu điều kiện: Người dùng đăng nhập thành công và có thể sử dụng chức năng của hệ thống tương ứng với mỗi quyền của tài khoản.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.3 Use case “*Quên mật khẩu*”

Mô tả vắn tắt:

Use case này cho phép khách hàng đặt lại mật khẩu đã quên.

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case này bắt đầu khách kích chọn chức năng “Quên mật khẩu” trên màn hình đăng nhập. Hệ thống sẽ hiển thị màn hình yêu cầu nhập số điện thoại người dùng.
- 2) Người dùng nhập số điện thoại . Hệ thống gửi mã OTP đến số điện thoại đó và người dùng nhập mã OTP.
- 3) Người dùng nhập mã OTP . Hệ thống sẽ hiển thị màn hình khởi tạo mật khẩu mới.
- 4) Người dùng nhập mật khẩu mới, xác nhận lại mật khẩu sau đó nhấn nút “Xác nhận”. Hệ thống sẽ chuyển đến trang đăng nhập để thực hiện đăng nhập. Use case kết thúc.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
- 2) Tại bước 2 ở luồng cơ bản nếu người dùng nhập sai số điện thoại hệ thống yêu cầu người dùng nhập lại số điện thoại. Use case kết thúc.
- 3) Tại bước 4 ở luồng cơ bản, nếu người dùng nhập sai mật khẩu hoặc trường mật khẩu không khớp với trường xác nhận thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc.

Tiền điều kiện: Người dùng quên mật khẩu, không thể đăng nhập vào hệ thống.

Hậu điều kiện: Người dùng đã cập nhật mật khẩu, đăng nhập thành công và có thể sử dụng chức năng của hệ thống

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.4 Use case “Xem khuyến mại”

Mô tả vắn tắt:

Use case này cho phép khách hàng xem các khuyến mại mà người dùng nhận được khi sử dụng ứng dụng.

Luồng sự kiện:

Luồng cơ bản:

Use case này bắt đầu khi khách hàng kích vào “Khuyến mại” menu chính. Hệ thống sẽ hiển thị danh sách khuyến mại lên trên màn hình.

Use case kết thúc.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống

Hậu điều kiện: Cập nhật tài khoản thành công

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.5 Use case “Xem tin tức”

Mô tả vắn tắt:

Use case này cho phép khách hàng xem các tin tức mới nhất khi sử dụng ứng dụng.

Luồng sự kiện:

Luồng cơ bản:

Use case này bắt đầu khi khách hàng kích vào ”Trang chủ” menu chính. Hệ thống sẽ hiển thị danh sách tin tức lên trên màn hình. Use case kết thúc.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Không có.

Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.6 Use case “Xem thông báo”

Mô tả vắn tắt:

Use case này cho phép khách hàng xem thông báo gửi về từ hệ thống.

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case này bắt đầu khi khách hàng kích biểu tượng chiếc “Chuông” trên màn hình trang chủ hoặc màn hình cài đặt. Hệ thống sẽ hiển thị danh sách thông báo lên màn hình.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Người dùng phải đăng nhập vào hệ thống.

Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.7 Use case “Tài khoản ví điện tử”

Mô tả vắn tắt:

Use case này cho phép khách hàng xem thông tin về tài khoản như số dư, số dư khả dụng.

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case bắt đầu khi khách hàng kích vào “avatar tài khoản” trên màn hình trang chủ hoặc màn hình cài đặt. Khi đó hệ thống sẽ lấy thông tin về tài khoản và hiển thị thông tin đó lên màn hình.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu

không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Khách hàng phải đăng nhập vào hệ thống.

Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.8 Use case “Quét mã QR”

Mô tả vắn tắt:

Use case này cho phép khách hàng quét mã QR để thực hiện thanh toán

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case bắt đầu khi khách hàng click vào biểu tượng “Mã QR” trên thanh menu chính. Hệ thống sẽ tạo ra mã giao dịch và hiển thị thông tin mà mã QR quét được ở trang chuyển tiền.
- 2) Khách hàng nhập thông tin cần thiết như số tiền cần chuyển, nội dung chuyển khoản rồi nhấn nút tiếp theo.
- 3) Hệ thống sẽ hiển thị màn hình xác nhận thông tin từ tên người dùng, số tiền chuyển, nội dung chuyển khoản, phí.
- 4) Người dùng nhấn nút xác nhận. Hệ thống gửi mã OTP xác nhận thực hiện giao dịch về máy. Người dùng nhập mã xác nhận và nhấn xác nhận.
- 5) Hệ thống sẽ chuyển đến trang thông tin giao dịch và Usecase kết thúc.

Luồng rẽ nhánh:

- 1) Tại bước 1 trong luồng cơ bản nếu mã QR không chính xác hay ứng dụng không thể đọc được mã QR và use case kết thúc.
- 2) Tại bước 2 trong luồng cơ bản nếu người dùng nhấn mũi tên trở lại thì Hệ thống sẽ điều hướng lại trang quét mã QR và xóa mã giao dịch đã sinh ra.

- 3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối internet thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4)

Tiền điều kiện: Khách hàng phải đăng nhập vào hệ thống.

Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.9 Use case “Xem chi tiết thẻ”

Mô tả vắn tắt:

Use case này cho phép khách hàng xem chi tiết thẻ ngân hàng đã liên kết với tài khoản.

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case bắt đầu khi khách hàng click vào nút “Liên kết ”trên Màn hình trang chủ. Hệ thống sẽ thực hiện chuyển đến trang quản lý thẻ liên kết.
- 2) Khách hàng chọn thẻ ngân hàng muốn xem chi tiết. Hệ thống sẽ chuyển đến trang chi tiết của thẻ ngân hàng đó.

Luồng rẽ nhánh:

- 3) Tại bước 1 trong luồng cơ bản nếu chưa liên kết với thẻ ngân hàng nào thì hệ thống sẽ để trống phần hiển thị danh sách thẻ liên kết và use case kết thúc.
- 4) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Khách hàng phải đăng nhập vào hệ thống.

Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.10 Use case "Cài đặt"

Mô tả vắn tắt:

Use case này cho phép khách hàng xem các cài đặt và các chức năng khác với tài khoản.

Luồng sự kiện:**Luồng cơ bản:**

- 1) Use case bắt đầu khi khách hàng click vào nút "Cài đặt" trên thanh menu chính. Hệ thống sẽ thực hiện chuyển đến trang cài đặt.

Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Khách hàng phải đăng nhập vào hệ thống.

Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

2.3.1.11 Use case "Nạp tiền"

Mô tả vắn tắt:

Use case này cho phép khách hàng nạp tiền vào tài khoản.

Luồng sự kiện:

Luồng cơ bản:

- 1) Use case bắt đầu khi khách hàng click vào nút "Nạp tiền" trên Màn hình trang chủ. Hệ thống sẽ thực hiện chuyển đến trang Nạp tiền.
- 2) Khách hàng nhập hoặc chọn số tiền muốn nạp vào tài khoản và chọn nguồn tiền nạp vào và nhấn tiến theo.
- 3) Hệ thống sẽ hiển thị màn hình xác nhận thông tin trước khi thực hiện nạp tiền vào tài khoản. Khách hàng nhấn nút xác nhận.
- 4) Hệ thống sẽ gửi mã OTP về máy để thực hiện giao dịch. Khách hàng nhập mã OTP và nhấn xác nhận. Hệ thống sẽ chuyển hướng đến trang chi tiết giao dịch. Use case kết thúc.

Luồng rẽ nhánh:

- 1) Tại bước 01 luồng cơ bản nếu chưa liên kết tài khoản nào thì người dùng sẽ phải liên kết với thẻ ngân hàng để thực hiện tiếp chức năng. Use case kết thúc.

Us Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Tiền điều kiện: Khách hàng phải đăng nhập vào hệ thống.

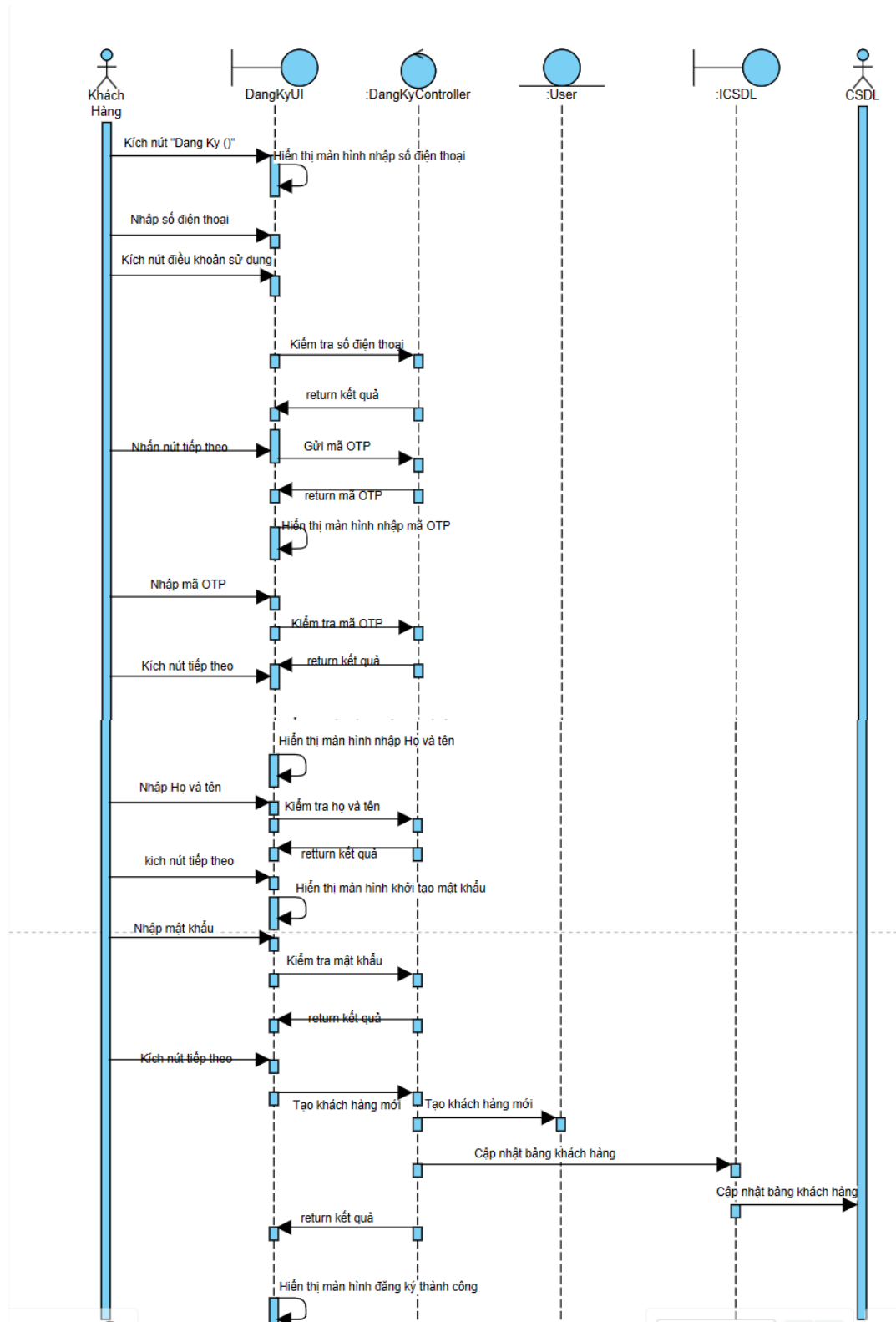
Hậu điều kiện: Không có.

Các yêu cầu đặc biệt: Không có.

Điểm mở rộng: Không có.

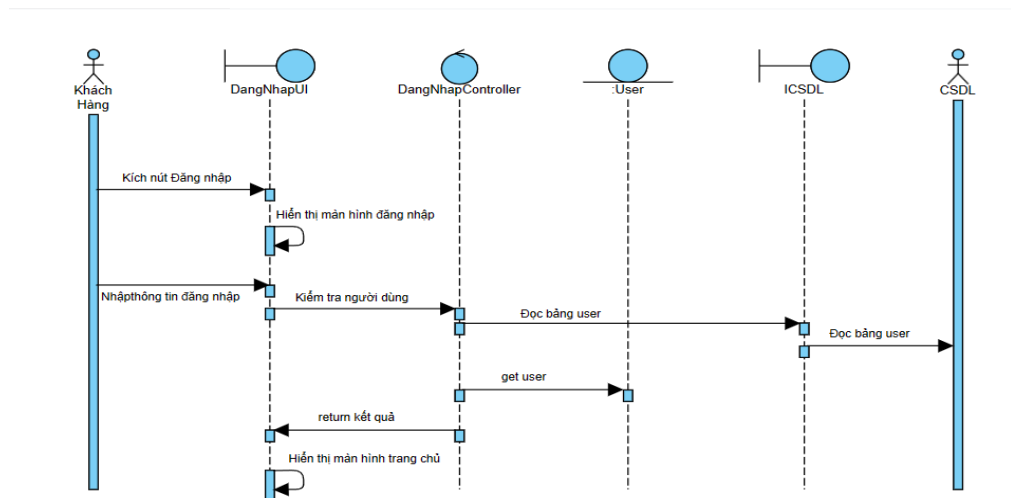
2.4.2 Biểu đồ trình tự

2.4.2.1 Use case “Đăng ký”



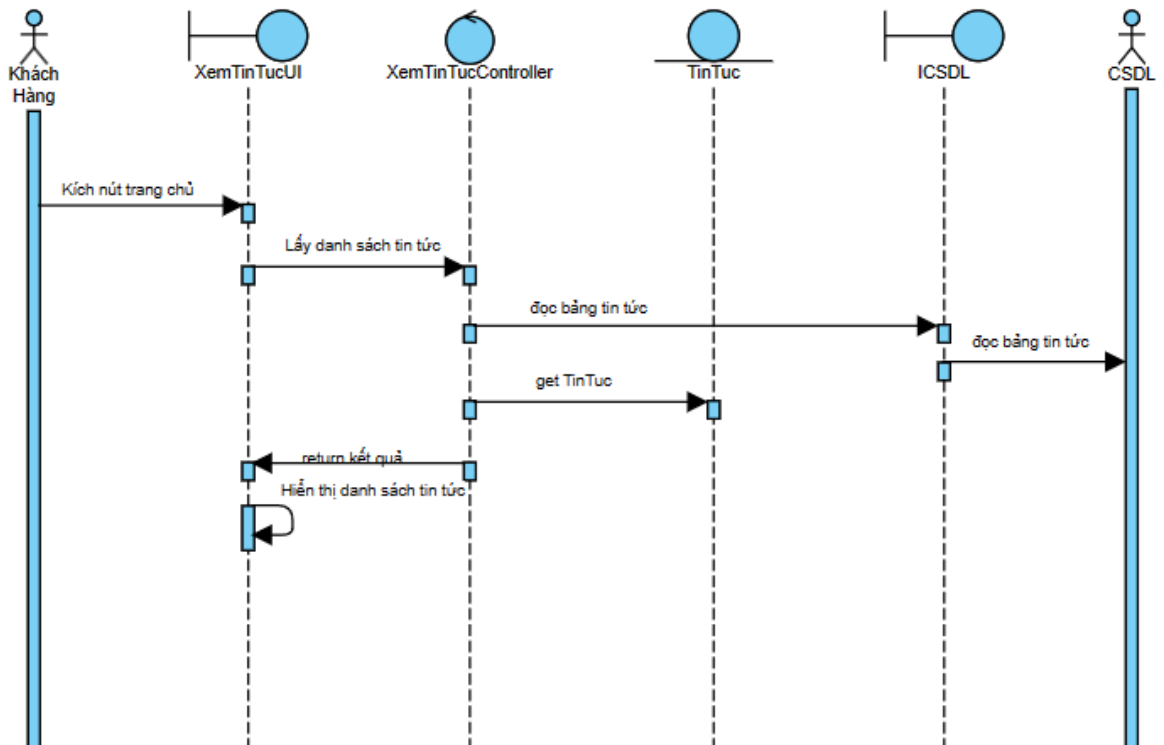
Hình 2.0.2. Biểu đồ trình tự use case Đăng ký

2.4.2.2 Use case “Đăng nhập”



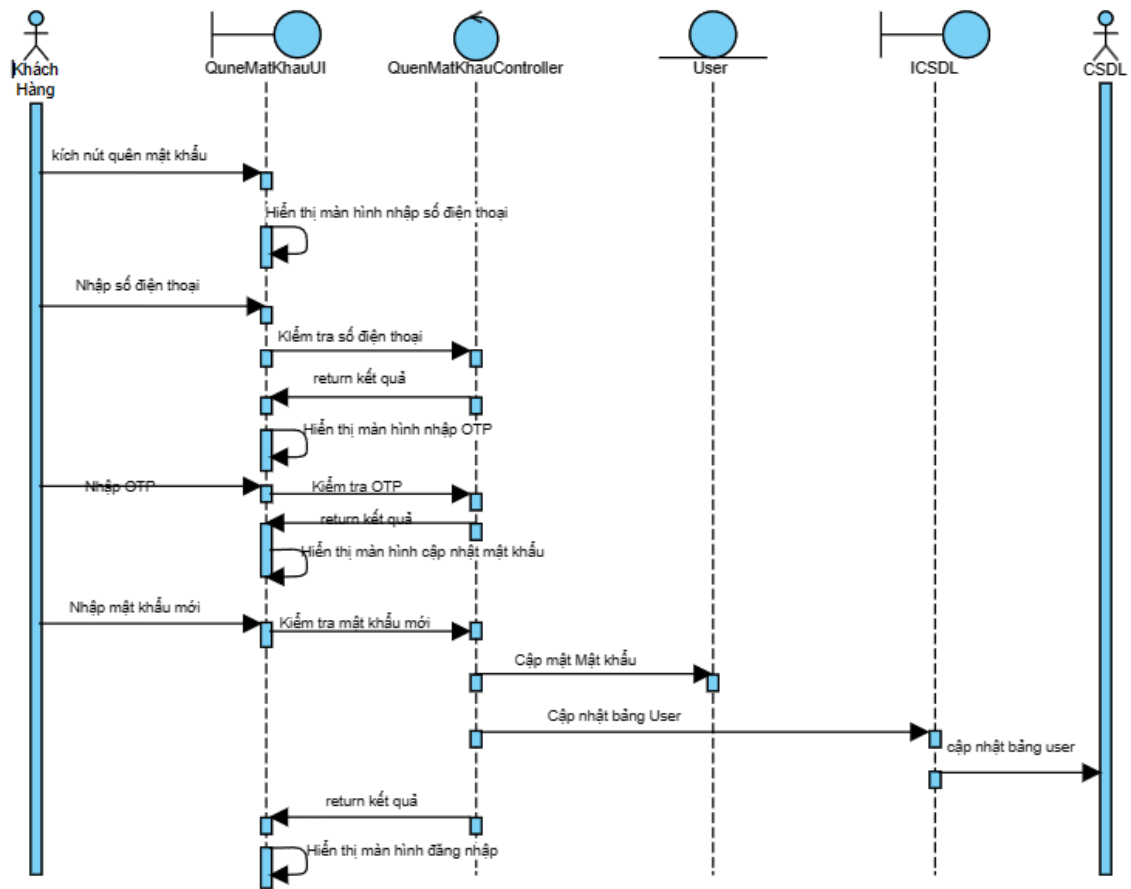
Hình 2.3. Biểu đồ trình tự use case “Đăng nhập”

2.4.2.3 Use case “Xem tin tức”



Hình 2.4 Biểu đồ trình tự use case “Xem tin tức”

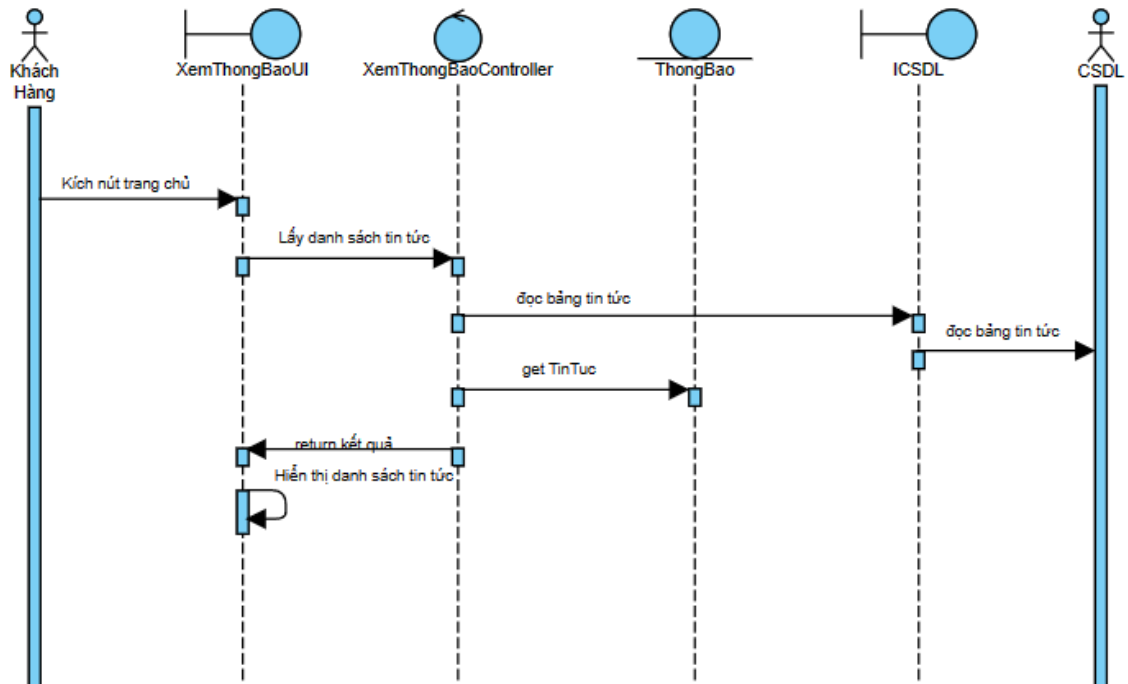
2.4.2.4 Use case “Quên mật khẩu”



Hình 2.5 Biểu đồ trình tự use case “Quên mật khẩu”

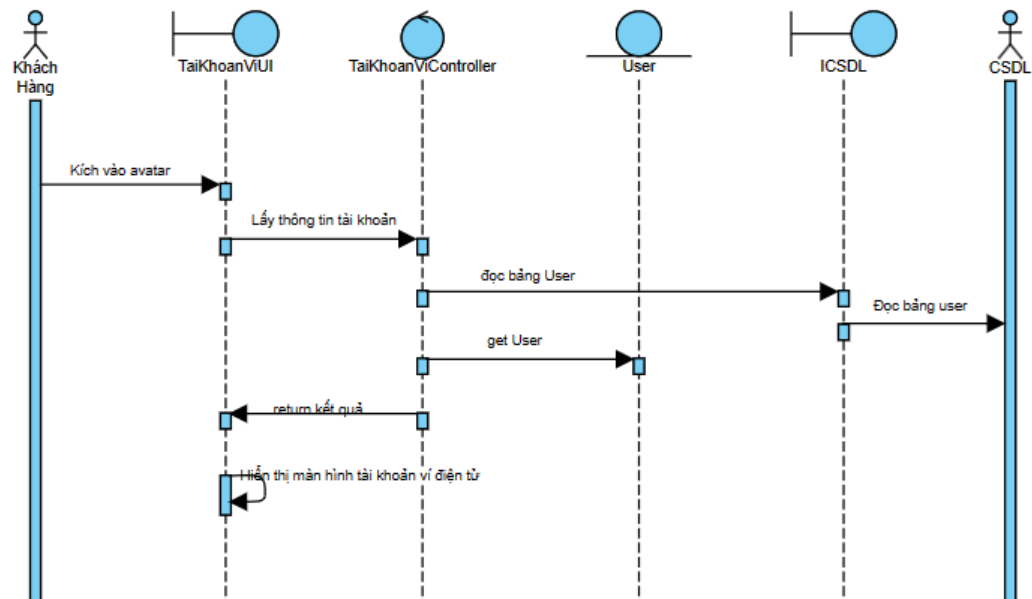
Hình 2.6 Biểu đồ trình tự use case “Xem khuyến mại”

2.4.2.6 Use case “Xem thông báo”



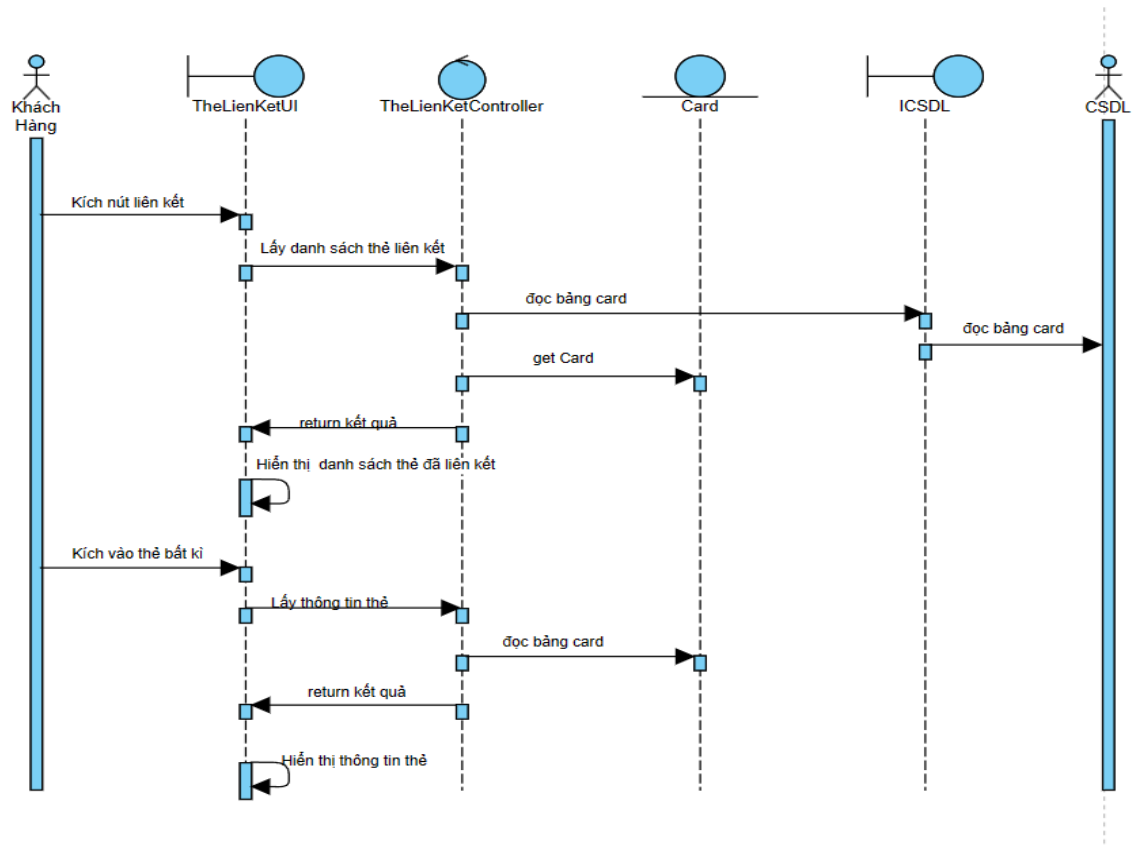
Hình 2.7 Biểu đồ trình tự của use case “Xem thông báo”

2.4.2.1 Use case “Tài khoản ví điện tử”



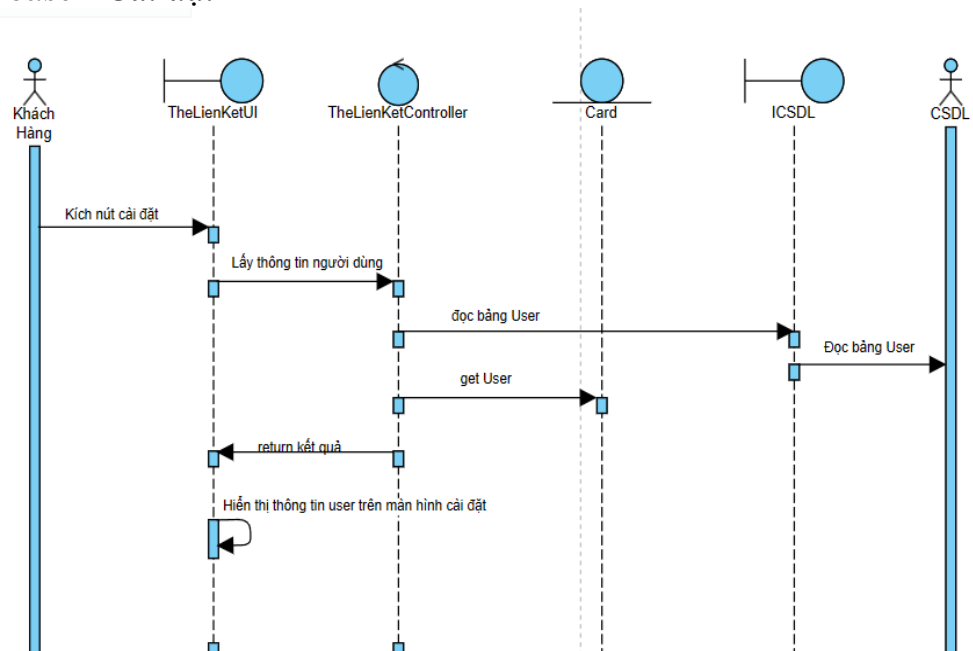
Hình 2.8 Biểu đồ trình tự Use case “Tài khoản ví điện tử”

2.4.2.8. Use case “Xem chi tiết thẻ”



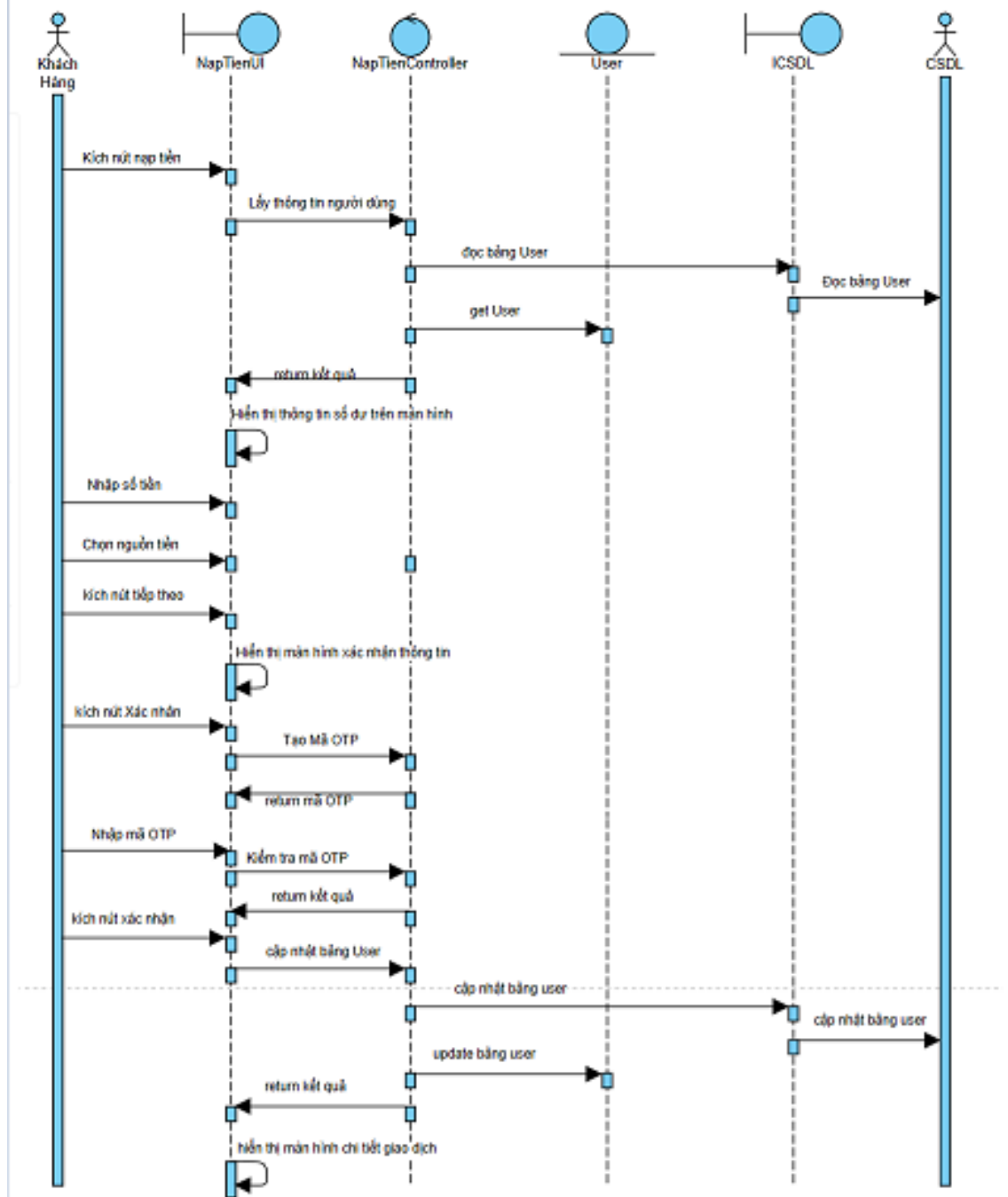
Hình 2.9 Biểu đồ trình tự use case “Xem chi tiết thẻ”

2.4.2.9 Use case “Cài đặt”



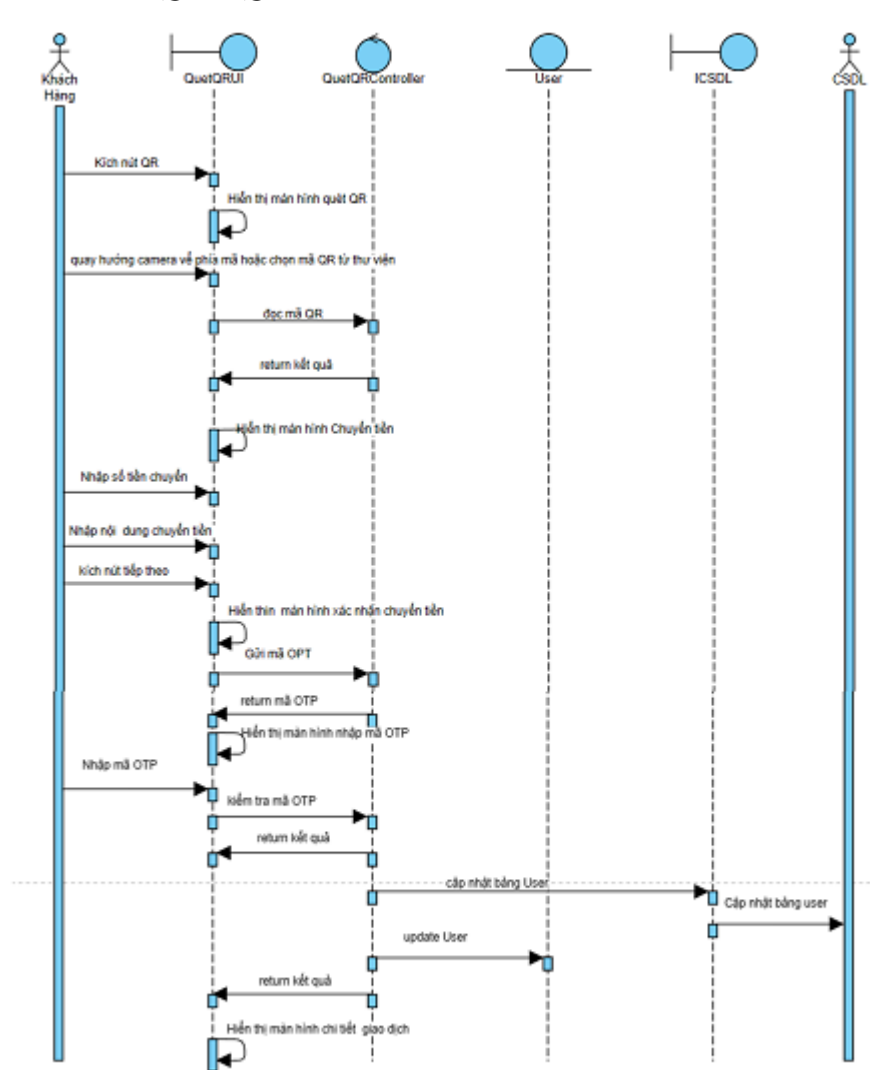
Hình 2.10 Biểu đồ trình tự use case “Cài đặt”

2.4.2.10 Use case “Nạp tiền”



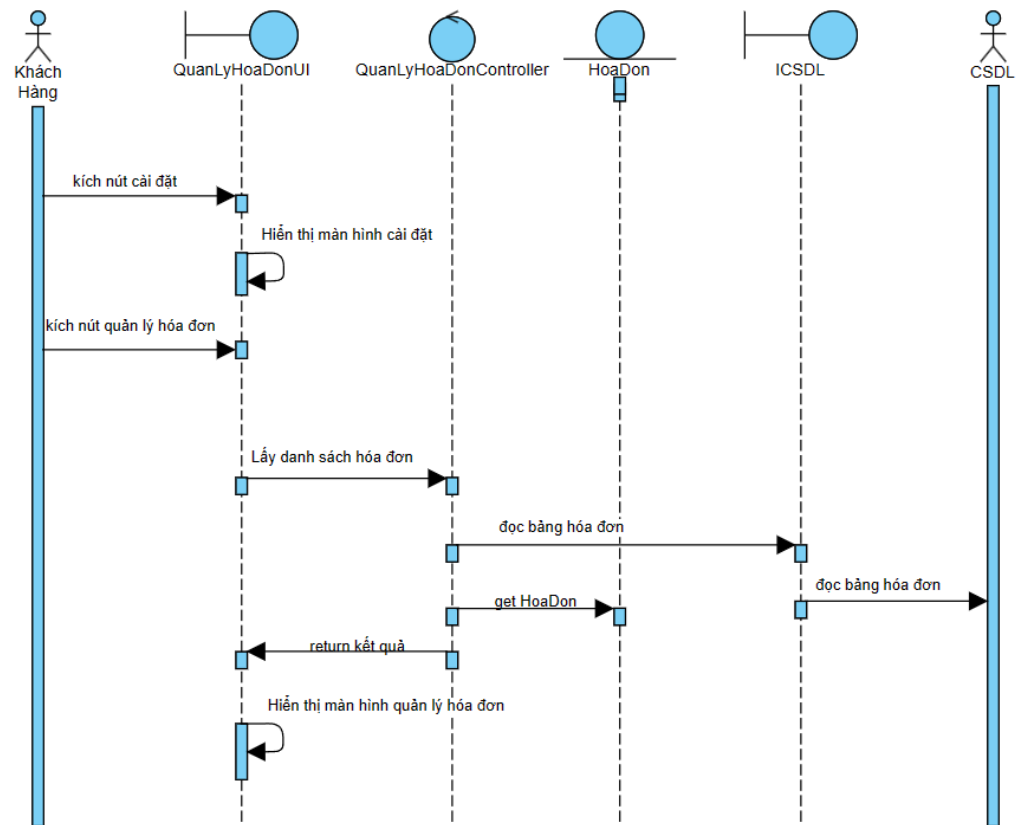
Hình 2.11 Biểu đồ trình tự use case “Nạp tiền”

2.4.2.11. Use case “Quét QR”



Hình 2.12 Biểu đồ trình tự Use case “Quét QR”

2.4.2.12. Use case “Quản lý hóa đơn”



Hình 2.13 Biểu đồ trình tự use case “Quản lý hóa đơn”

2.5 Thiết kế cơ sở dữ liệu

2.5.1 Cấu trúc các bảng

❖ Bảng User

Bảng 2.3: Bảng User

Tên Cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
phoneNumber	String	No	Primary key	Số điện thoại người dùng
userName	String	No	No	Họ và tên người dùng
balance	double	No	No	Số dư

❖ Bảng ThôngBao

Bảng 2.4: Bảng ThôngBao

Tên Cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
notificationID	String	No	Primary key	Id Thông báo
title	String	No	No	Tiêu đề thông báo
type	Int	No	No	Thuộc thông báo số dư hay không
content	String	No	No	Nội dung thông báo
time	String	No	No	Thời gian khởi tạo thông báo

Bảng 2.3: Bảng HoaDon

Tên Cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
hoaDonID	String	No	Primary key	Id hóa đơn
maGiaDich	String	No	No	Mã Giao dịch
NameNhaCungCap	String	No	No	Tên nhà cung cấp dịch vụ
userName	String	No	No	Tên người dùng
status	String	No	No	Trạng thái của giao dịch
type	String	No	No	Loại hóa đơn
time	String	No	No	Thời gian giao dịch
source	String	No	No	Nguồn tiền
sourceNumber	String	No	No	Số thẻ của nguồn tiền
total	double	No	No	Số Tiền

❖ Bảng TinTuc:

Bảng 2.4 Bảng TinTuc

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
newID	String	No	Primary key Auto Increment	Id Tin Tức
title	String	No	No	Tiêu đề khuyến mại
imgPath	String	No	No	Ảnh khuyến mại
content	String	No	No	Nội dung khuyến mại
time	String	No	No	Thời gian khuyến mại

❖ Bảng KhuyenMai:

Bảng 2.5 Bảng KhuyenMai

Tên cột Field	Kiểu dữ liệu Data Type	Null	Ràng buộc Constraint	Mô tả Content
promotionID	String	No	Primary key Auto Increment	Id Khuyến mại
nhaCungCap	String	No	No	Tên nhà cung cấp
title	String	No	No	Tiêu đề khuyến mại
imgPath	String	No	No	Ảnh khuyến mại
content	String	No	No	Nội dung khuyến mại
time	String	No	No	Thời gian khuyến mại

CHƯƠNG 3: CÀI ĐẶT CHƯƠNG TRÌNH VÀ KẾT QUẢ ĐẠT ĐƯỢC

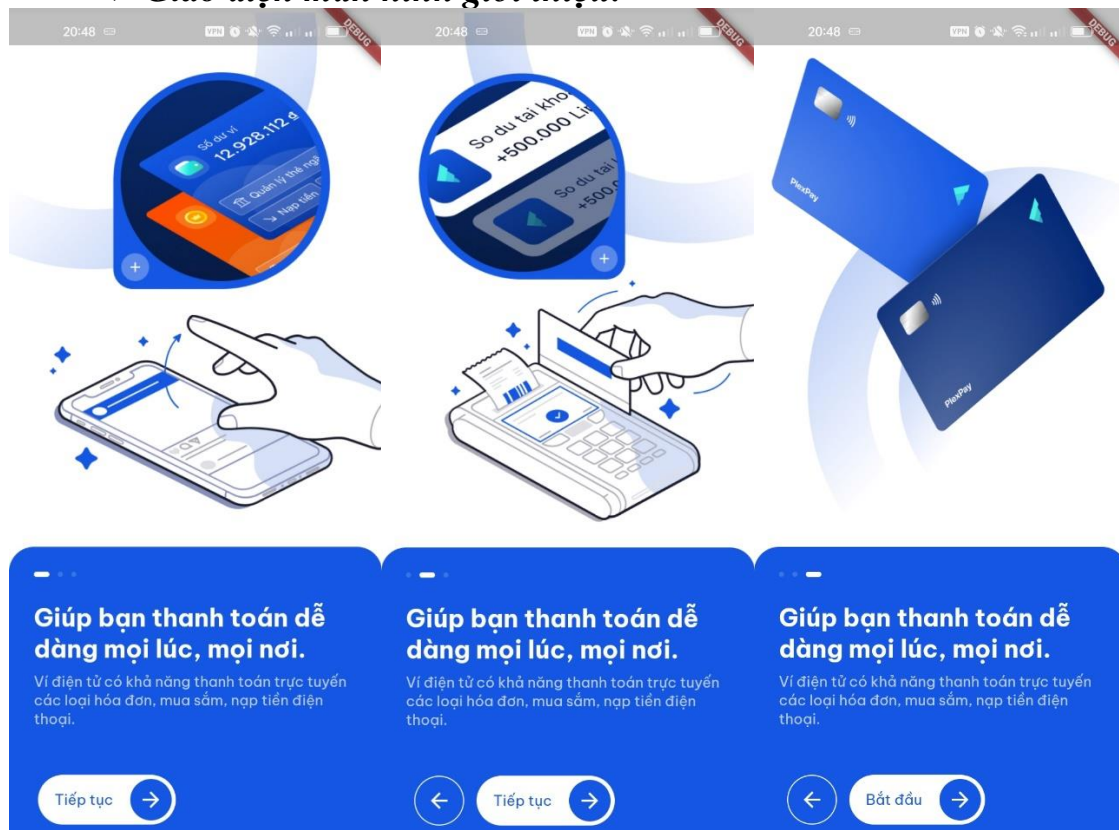
3.1 Công nghệ và môi trường sử dụng

Hiện nay có rất nhiều công nghệ được sử dụng để phát triển và thiết kế ứng dụng di động (Kotlin, Java ...). Do những ưu điểm mà flutter mang lại (đã trình bày ở chương 1) nên em đã quyết định sử dụng ngôn ngữ Dart và Framework Flutter để xây dựng ứng dụng ví điện tử.

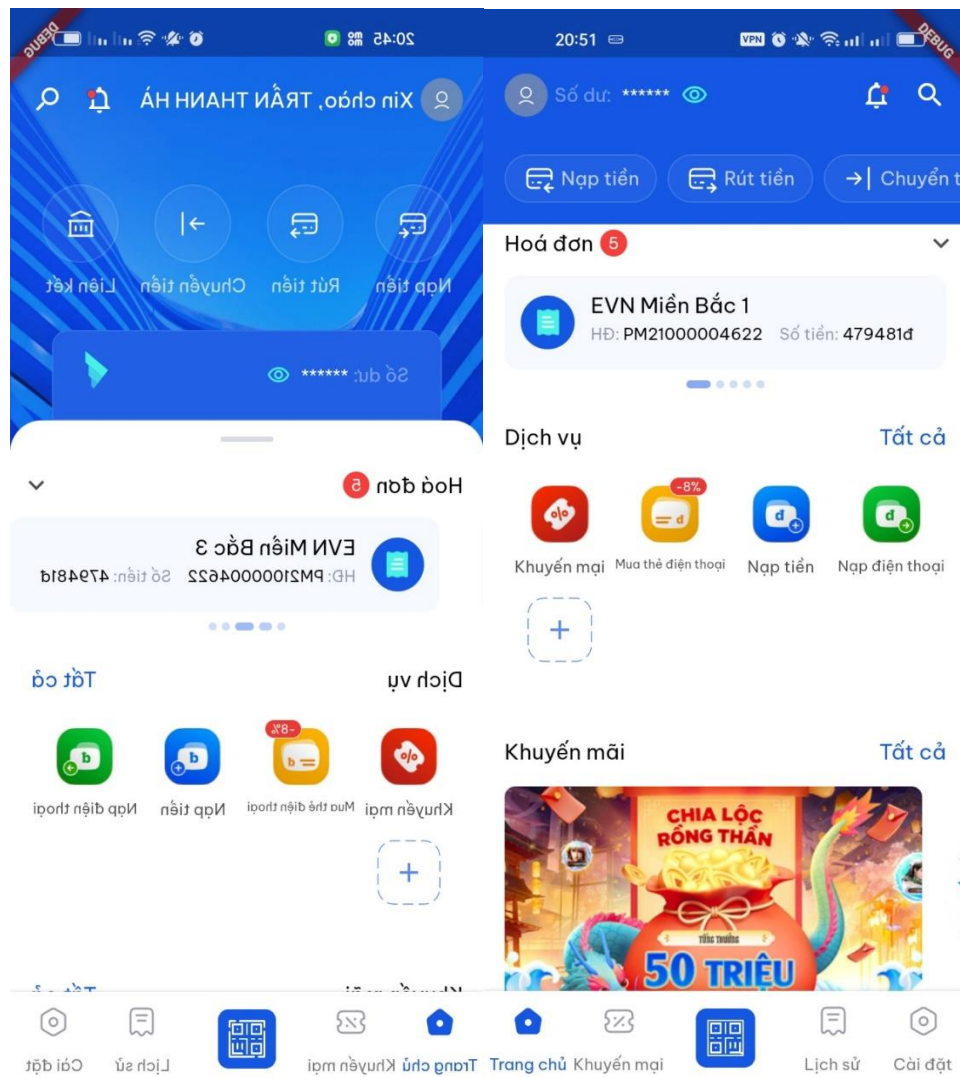
Sử dụng Visual Studio Code hoặc Android Studio làm môi trường để phát triển ứng dụng. Đây là những IDE rất hữu ích cho tối ưu code và debug gỡ lỗi nhanh chóng khi gặp lỗi khi phát triển ứng dụng Android.

3.2 Giao diện trang người dùng

❖ *Giao diện màn hình giới thiệu:*



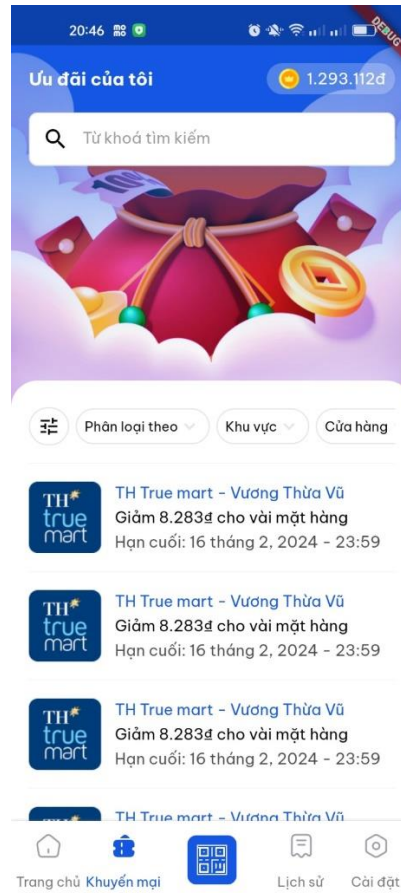
Hình 3.1 Giao diện màn hình giới thiệu

❖ *Giao diện trang chủ*

Hình 3.2 Giao diện trang chủ

❖ ***Giao diện hiển thị danh sách khuyến mại***

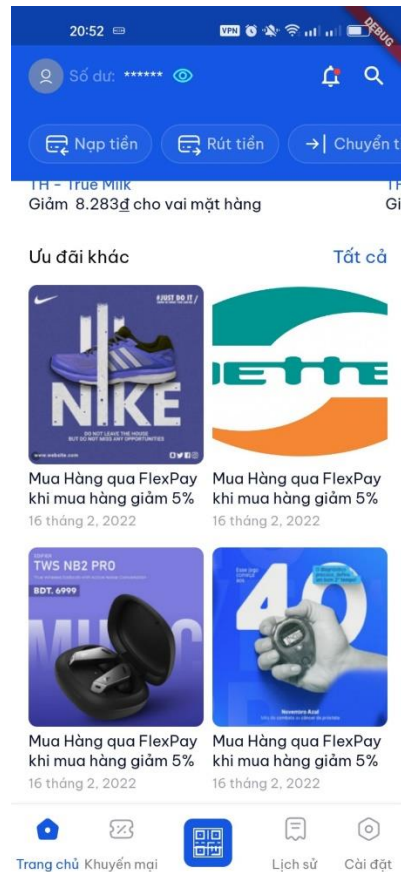
Khi người dùng kích vào nút khuyến mại trên thành menu chính.



Hình 3.3 Giao diện danh khuyến mại

❖ *Giao diện xem danh sách tin tức*

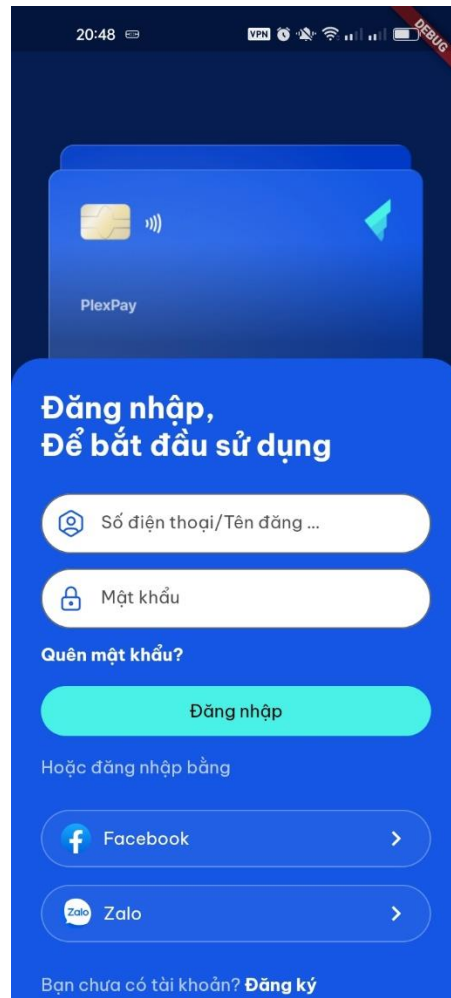
Người dùng kích vào nút trang chủ trên thanh menu. Danh sách tin tức được hiển thị tại cuối trang.



Hình 3.4 Giao diện xem danh sách tin tức

❖ **Giao diện đăng nhập:**

Sau khi qua các bước màn hình giới thiệu thì sẽ hiển thị giao diện màn hình đăng nhập



Hình 3.5 Giao diện đăng nhập

Người dùng nhập phoneNumber và password, hệ thống sẽ kiểm tra số điện thoại, mật khẩu. Nếu đúng số điện thoại và mật khẩu thì đăng nhập thành công. Hệ thống hiển thị màn hình trang chủ người dùng.

Click vào nút “Đăng ký” để tiến hành đăng ký một tài khoản mới nếu chưa có tài khoản.

Click “Quên mật khẩu” trong trường hợp người dùng quên mật khẩu.

❖ Giao diện đăng ký:

The image displays two sequential screens of a mobile application's registration process.

Left Screenshot (20:49): The screen is titled "Đăng ký tài khoản" (Register Account). It contains the text "Số điện thoại đăng ký sẽ là tên đăng nhập tài khoản của bạn." (The registration phone number will be your login name). Below this is a text input field labeled "Số điện thoại" (Phone Number). At the bottom, there is a checkbox with the text "Tôi đồng ý với điều khoản và điều kiện khi sử dụng dịch vụ của FlexPay" (I agree with the terms and conditions when using FlexPay service).

Right Screenshot (20:50): This screen is titled "Xác thực mã OTP" (OTP Verification). It shows the phone number "05555555555" and the instruction "Nhập mã xác thực được gửi vào số điện thoại 05555555555" (Enter the verification code sent to the phone number 05555555555). Below the instruction is a row of six input boxes for the OTP code. A "Gửi lại mã (30s)" (Resend code (30s)) link is present. At the bottom is a numeric keypad with digits 1-9, 0, a decimal point, and a checkmark button.

Both screenshots have a blue "Tiếp Theo" (Next) button at the bottom.

20:50

VPN

DEBUG

← Đăng ký tài khoản

Nhập tên của bạn để tiếp tục tạo tài khoản

Họ và tên

20:51

VPN

DEBUG

← Tạo mật khẩu

Mật khẩu phải chứa ít nhất 10 ký tự.
Chức ít nhất 1 ký tự viết hoa, 1 ký tự đặc biệt.

1111111111A@

.....

Tiếp Theo

()

Tiếp Theo

20:51

VPN

DEBUG

9:41

Xin chào, Linh!

Nạp tiền Rút tiền Chuyển tiền Liên kết

Số dư: ...

Hoà đơn

Điện lực EVN Hà Nội

HD: P00500031662 Số tiền: 394.232đ

Mở tài khoản thành công

Tài khoản đăng nhập: 123456567

Bạn có thể kích hoạt tính năng đăng nhập nhanh bằng sinh trắc học!

Kích hoạt sinh trắc học

Bắt đầu đăng nhập

Hình 3.6 Giao diện đăng kí tài khoản

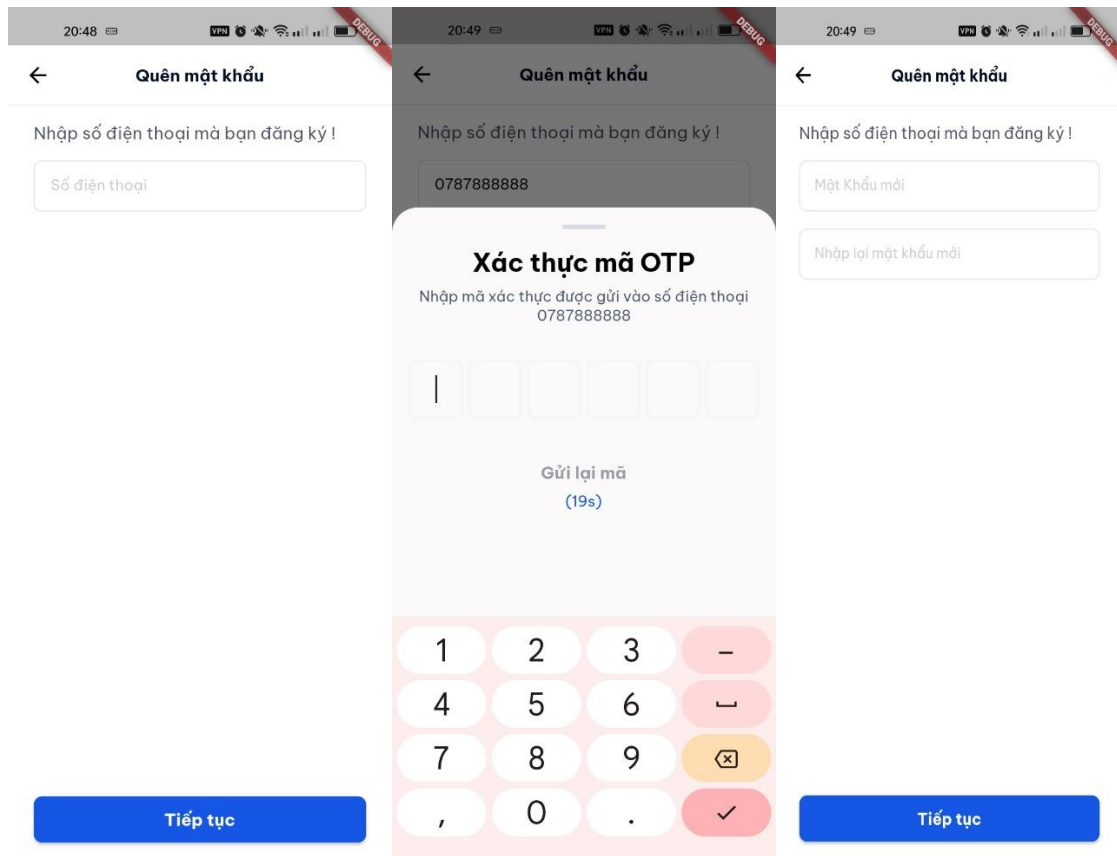
Người dùng nhập thông tin số điện thoại và tích vào ô điều khoản và điều kiện trước khi nhấn tiếp theo.

Hệ thống gửi mã OTP về máy, hiển thị màn hình nhập mã OTP. Người dùng nhập mã OTP để thực hiện bước tiếp theo.

Ở giao diện điền họ và tên người dùng nhập họ và tên của mình rồi nhấn nút tiếp theo.

Nhập mật khẩu theo mẫu quy định được hiển thị trên màn hình rồi nhấn tiếp tục. Nếu thành công thì sẽ chuyển đến trang đăng ký thành công. Nhấn nút bắt đầu đăng nhập để chuyển đến trang đăng nhập để sử dụng ứng dụng.

❖ Giao diện quên mật khẩu:



Hình 3.7 Giao diện quên mật khẩu

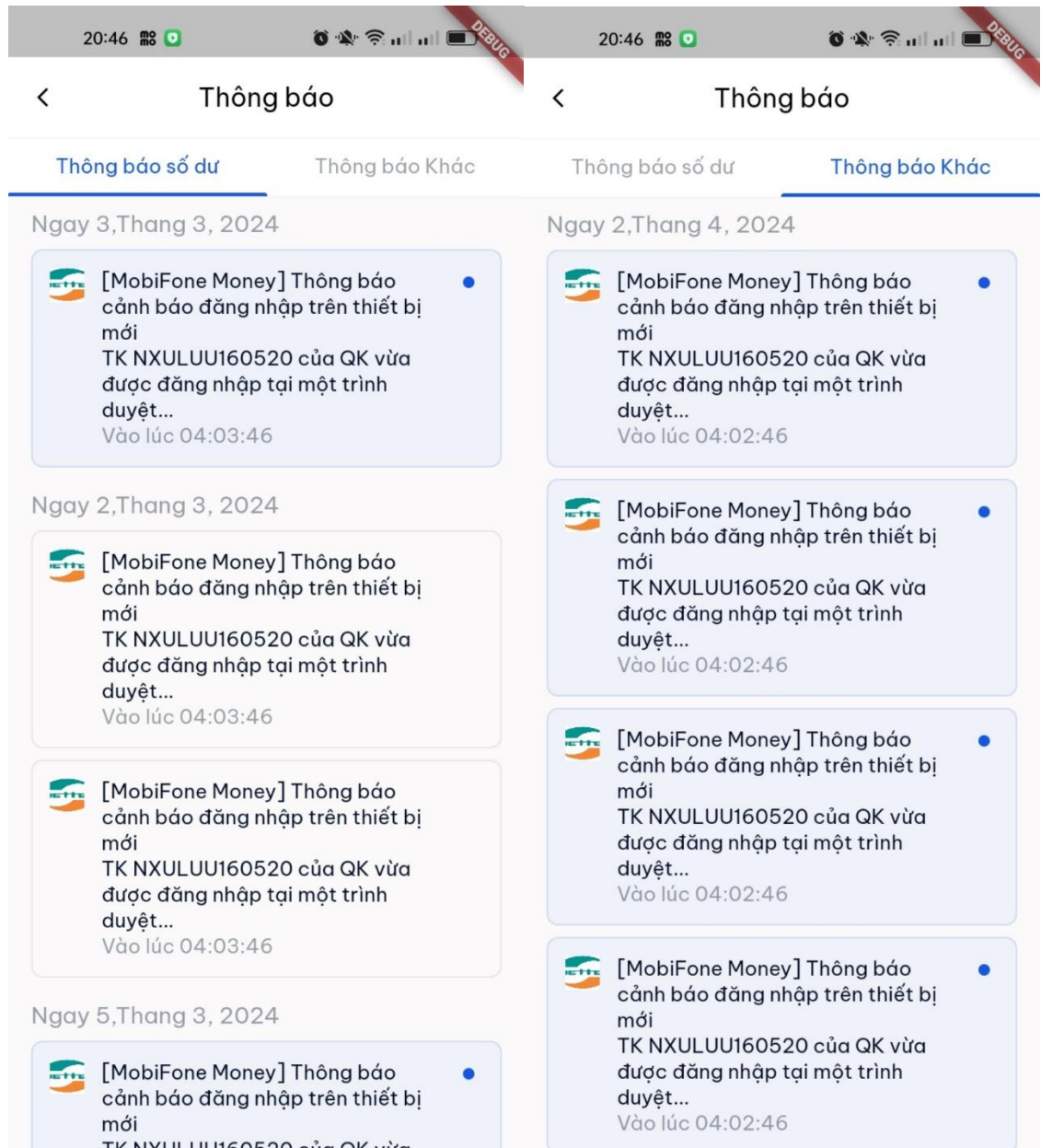
Trên màn hình đăng nhập nếu người dùng quên mật khẩu đăng nhập thì kích vào nút quên mật khẩu. Hệ thống sẽ chuyển trang đến màn hình nhập số điện thoại đã đăng ký tài khoản.

Người dùng nhập Mã OTP mà hệ thống gửi về để có thể chuyển qua bước khởi tạo lại mật khẩu.

Sau khi nhập xong mật khẩu thì sẽ chuyển đến trang đăng nhập để thực hiện đăng nhập vào tài khoản.

❖ *Giao diện màn hình thông báo*

Khi người dùng đã đăng nhập xong. ở màn hình trang chủ hoặc màn hình cài đặt nhấn vào biểu tượng chiếc chuông để hiển thị màn hình thông báo.

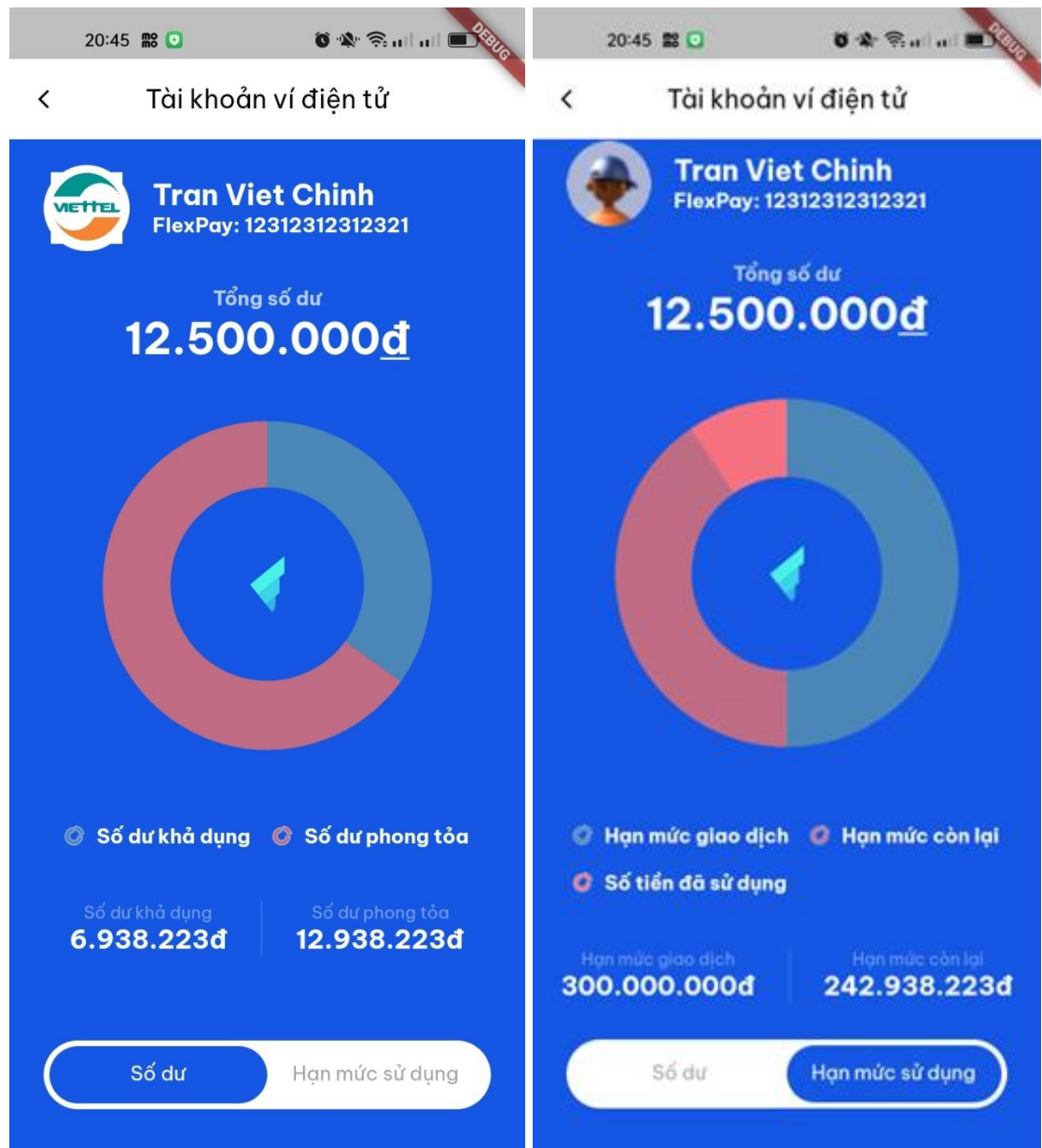


Hình 3.8 Giao diện màn hình thông báo

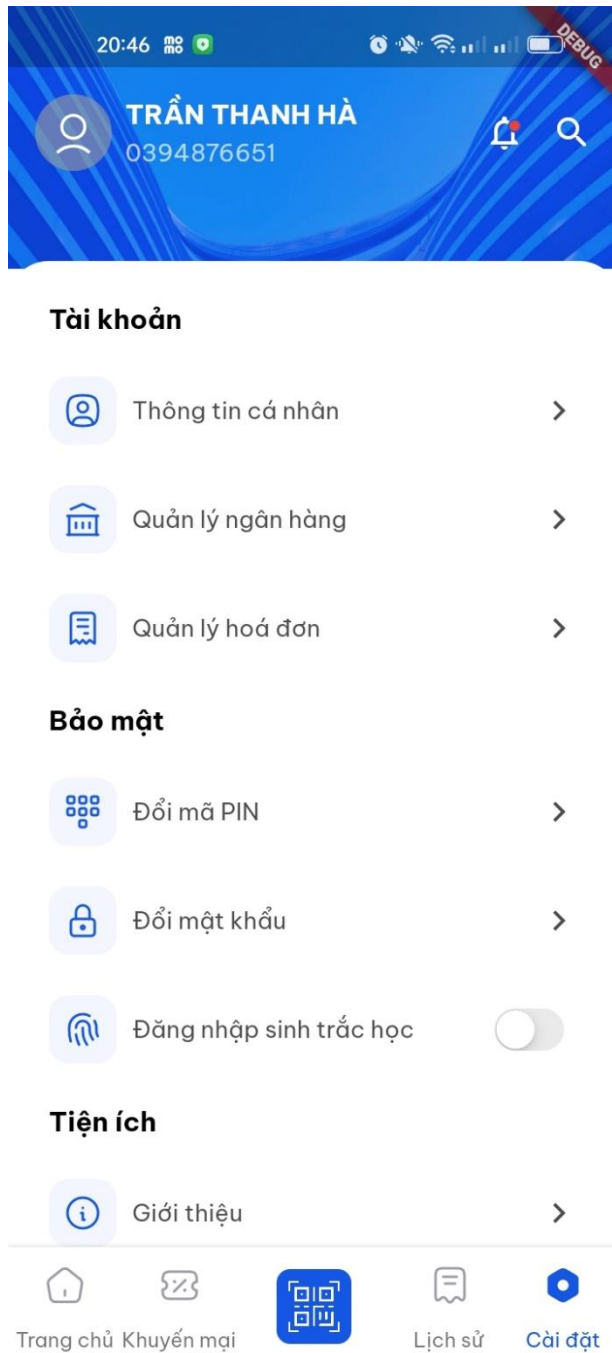
Nếu mà người dùng có thông báo chưa đọc thì sẽ được hiển thị nổi bật hơn những thông báo khác và nhấn vào để đánh dấu đã đọc thông báo đó rồi.

❖ *Giao diện màn hình tài khoản ví điện tử:*

Khi người dùng đã đăng nhập thành công. Kích vào avatar của tài khoản để vào màn hình ví điện tử



Hình 3.9 Giao diện màn hình tài khoản ví điện tử

❖ *Giao diện cài đặt :*

Hình 3.10 Giao diện cài đặt

Người dùng sau khi đăng nhập thành công chọn cài đặt trên thanh menu để vào màn hình cài đặt. Ở đây người dùng có thể truy cập vào màn hình quản lý hóa đơn,..

❖ *Giao diện nạp tiền :*

Nạp tiền

Số dư: 12.938.223 đ

Nhập số tiền cần nạp

0đ

50,000đ 100,000đ 200,000đ 500,000đ

Nạp từ nguồn tiền

MB *5383
Miễn phí giao dịch

Thẻ/TK ngân hàng khác

Tiếp theo

Nạp tiền

Nhập số tiền cần nạp

50,000đ

Ngân hàng MB

Tài khoản gửi MB*5383

Tài khoản nhận Ví điện tử 03333333

Họ tên Nguyễn Viết Tiến

Số tiền 50,000

Phí giao dịch 0

Tổng tiền 50,000đ

Xác nhận

Xác thực OTP

Vui lòng nhập mã OTP được gửi đến số điện thoại ****9526

8 8 8 8 9

Yêu cầu nhập đủ 6 ký tự

Hiệu lực: 01:57

Gửi lại

Xác nhận

Chi tiết giao dịch

Loại giao dịch Nạp điện thoại

Trạng thái Thành công

Mã giao dịch 2343242342

Thời gian 01/01/2024 12:12:12

Tài khoản Ví điện tử xxxx

Phí 2,200

Số tiền 100,000

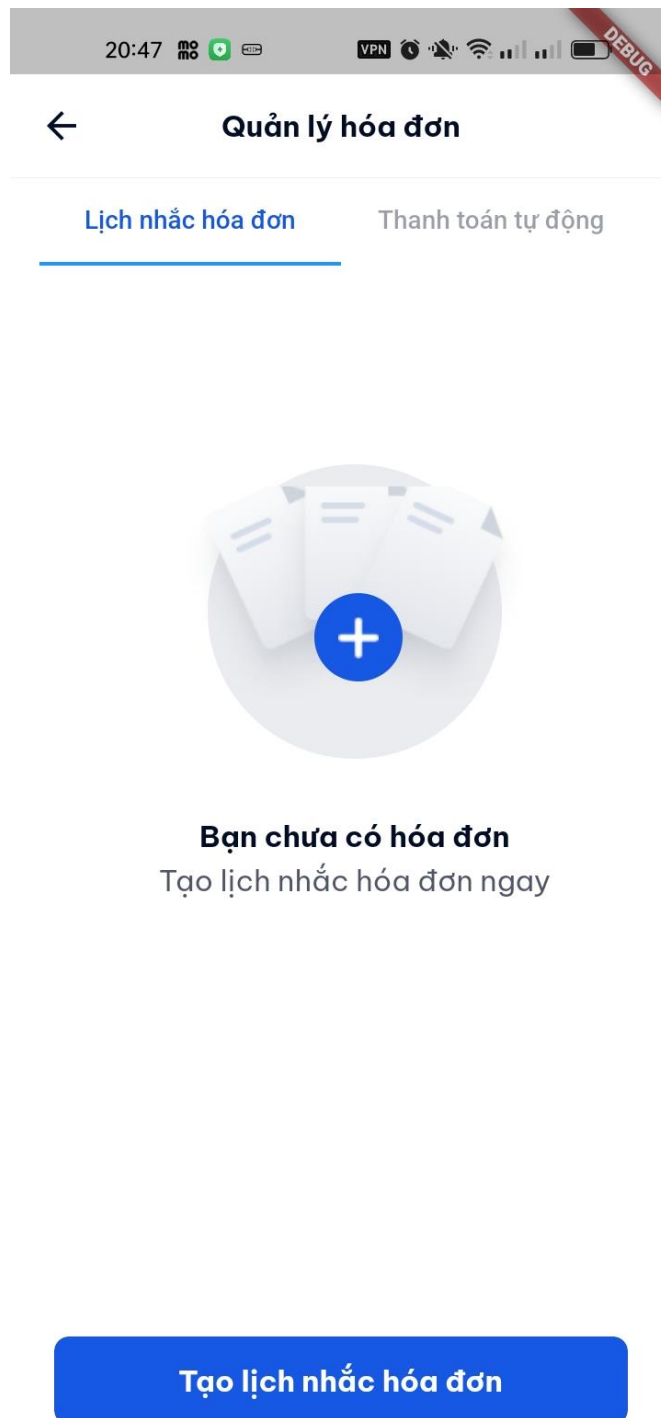
Chia sẻ

Giao dịch lại

Hình 3.11 Giao diện Nạp tiền

Sau khi đăng nhập thành công. Người dùng kích vào nút “Nạp tiền” trên màn hình trang chủ. Lúc này người dùng nhập số tiền và chọn nguồn tiền rồi nhấn xác nhận khi có màn hình xác nhận thông tin giao dịch.

Tiếp theo sẽ là nhập mã OPT hệ thống gửi về lúc này ứng dụng sẽ thực hiện giao dịch trạng thái giao dịch sẽ được hiển thị ở ô trạng thái trên màn hình chi tiết

❖ *Giao diện quản lý hóa đơn :*

Hình 3.12 Giao diện quản lý hóa đơn

Nếu mà người dùng phát sinh giao dịch thì màn hình này sẽ hiển thị danh sách hóa đơn mà người dùng đã thực hiện.

3.3 Kiểm thử

STT	Nội dung kiểm thử	Kết quả	Ghi chú
1	Khởi động ứng dụng	Ứng dụng khởi động khác chậm và các thao tác không được mượt mà.	Cải thiện thêm hiệu năng loại bỏ các phần không liên quan.
2	Điều hướng trang	Các nút và chức năng điều hướng đến đúng trang theo như thiết kế.	
3	Giao diện ứng dụng	Giao diện ứng dụng đã đúng với thiết kế, đúng kích thước, màu sắc	Cần phải kiểm tra trên nhiều điện thoại khác nhau để xem xét giao diện có bị lỗi hay không.
4	Tốc độ ứng dụng	Một số màn hình chạy chậm, có hiện tượng lag, giật.	Tối ưu hiệu năng ứng dụng, loại bỏ các phần không cần thiết, sử dụng thư viện quản lý trạng thái giúp tăng hiệu năng của ứng dụng.
5	Hao pin	ứng dụng chạy không sử dụng quá nhiều pin khi sử dụng	Kiểm tra trên các thiết bị khác

KẾT LUẬN

Qua quá trình làm đồ án tốt nghiệp em đã rút ra cho mình được rất nhiều kinh nghiệm, kiến thức bổ ích phục vụ cho công việc sau khi ra trường.

Với đề tài “Xây dựng ứng dụng ví điện tử HoangPay trên nền tảng Mobile” em đã xây dựng được các chức năng cơ bản của một ứng dụng ví điện tử. Em tập trung vào giao diện người dùng nhiều hơn là các chức năng của ứng dụng.

Ứng dụng ví điện tử của em gồm một số chức năng như sau:

- Phía người dùng: đăng nhập, đăng ký, quên mật khẩu, màn hình giới thiệu, xem danh sách khuyến mại, xem danh sách tin tức, xem danh sách thông báo, xem thông tin về tài khoản, quản lý hóa đơn.

Do thời gian có hạn và kiến thức của em chưa được đủ sâu và rộng nên em chưa thể phát triển được một số tính năng như: chưa thanh toán trực tuyến, xác thực thông tin khách hàng(chứng minh thư, thẻ căn cước), Rút tiền, Chuyển tiền, liên kết thực tế với thẻ ngân hàng, quét mã QR của các ngân hàng.

Hướng phát triển tiếp theo của ứng dụng sẽ là hướng tới một ứng dụng ví điện tử có đầy đủ các tính năng của một ví điện tử đúng nghĩa và em sẽ có thể bổ sung thêm các chức năng của người quản trị như quản lý thông tin khuyến mại, quản lý người dùng, quản lý các dịch vụ,... Nếu trong khả năng cho phép em sẽ phát triển ứng dụng này thành ứng dụng hoàn chỉnh và phát hành nó trên chợ ứng dụng CHPlay, App Store cho cả 2 nền tảng Android và IOS

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Flutter : <https://flutter.dev/>
- [2] Dart: <https://dart.dev/>
- [3] Giáo trình Phân tích thiết kế hệ thống. Tác giả: Nguyễn Thị Thanh Huyền. Nhà xuất bản giáo dục Việt Nam
- [4] Tại sao chọn Flutter: <https://www.geeksforgeeks.org/videos/why-to-choose-flutter-for-android-development-android-tips/>