**NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**CỦA GIÁO VIÊN PHẢN BIỆN**

Giáo viên phản biện:.................................................................Bộ môn:

Tên đồ án: **Ứng dụng “Nhận diện ký tự quang học” trong xây dựng quản lý chi tiêu cá nhân trên nền tảng iOS**

Sinh viên thực hiện: **Vương Vũ Bắc Sơn**

Lớp: **E16CQCN01**

Giáo viên hướng dẫn: **Ths. Trịnh Thị Vân Anh**

**NỘI DUNG NHẬN XÉT:**

**I. Nội dung báo cáo**

...................................

......................

**II. Sản phẩm**

**III. Ưu như­ợc điểm**

**IV. Kết luận:**

*Hà Nội, ngày......tháng.......năm 20.....*

GIÁO VIÊN PHẢN BIỆN

**NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**CỦA GIÁO VIÊN PHẢN BIỆN**

Giáo viên phản biện:.................................................................Bộ môn:

Tên đồ án: **Ứng dụng “Nhận diện ký tự quang học” trong xây dựng quản lý chi tiêu cá nhân trên nền tảng iOS**

Sinh viên thực hiện: **Vương Vũ Bắc Sơn**

Lớp: **E16CQCN01**

Giáo viên hướng dẫn: **Ths. Trịnh Thị Vân Anh**

**NỘI DUNG NHẬN XÉT:**

**I. Nội dung báo cáo**

...................................

......................

**II. Sản phẩm**

**III. Ưu như­ợc điểm**

**IV. Kết luận:**

*Hà Nội, ngày......tháng.......năm 20.....*

GIÁO VIÊN PHẢN BIỆN

**Lời cảm ơn**

Đầu tiên, em xin chân thành cảm ơn khoa Công nghệ thông tin 1, Học viện Công nghệ Bưu chính Viễn Thông đã tạo điều kiện thuận lợi cho em học tập và thực hiện đề tài tốt nghiệp này.

Em xin bày tỏ lòng biết ơn sâu sắc tới cô Trịnh Thị Vân Anh đã tận tình hướng dẫn chỉ bảo chúng em trong quá trình thực hiện đề tài. Chúng em xin chân thành cảm ơn thầy cô trong khoa Công nghệ thông tin, khoa Cơ bản đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu trong những năm vừa qua. Tôi cũng xin cảm ơn bạn bè, anh chị em trong và ngoài trường đã quan tâm, giúp đỡ trong quá trình thực hiện đề tài. Và sau cùng, chúng con xin cảm ơn cha mẹ, những người đã sinh thành, dưỡng dục và nuôi dạy chúng con nên người. Suốt đời này chúng con luôn ghi nhớ ơn Người.

Mặc dù đã cố gắng hoàn thành khóa luận trong phạm vi và khả năng cho phép những cũng không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm, góp ý cũng như nhận xét của quý thầy cô và các bạn.

**Hà Nội, tháng 12 năm 2020**

***Sinh viên***

**Vương Vũ Bắc Sơn**

**MỤC LỤC**

[Chương 1: Giới thiệu và tổng quan đề tài 2](#_Toc60954683)

[1.1 Thực trạng vấn đề 2](#_Toc60954684)

[1.2 Xác định yêu cầu bài toán 2](#_Toc60954685)

[1.3 Mục tiêu 5](#_Toc60954686)

[1.4 Đối tượng và phạm vi 5](#_Toc60954687)

[Chương 2: Các công nghệ được sử dụng 6](#_Toc60954688)

[2.1 Giới thiệu chung về hệ điều hành iOS và ngôn ngữ lập trình Swift 6](#_Toc60954689)

[*2.1.1.* *Hệ điều hành iOS* 6](#_Toc60954690)

[*2.1.2.* *Ngôn ngữ lập trình Swift* 7](#_Toc60954691)

[2.2. Giới thiệu về Firebase 8](#_Toc60954692)

[*2.2.1.* *Giới thiệu chung* 8](#_Toc60954693)

[*2.2.2.* *Các dịch vụ của Firebase* 9](#_Toc60954694)

[*2.2.3.* *Hệ quản trị cơ sở dữ liệu Firebase Realtime Database* 9](#_Toc60954695)

[2.3. Triết lý lập trình thế hệ mới – Lập trình phản ứng chức năng (Functional Reactive Programming) 10](#_Toc60954696)

[*2.3.1.* *Giới thiệu về lập trình phản ứng chức năng (Functional Reactive Programming - FRP)* 10](#_Toc60954697)

[*2.3.2.* *Lập trình phản ứng trong Swift với RxSwift* 12](#_Toc60954698)

[2.4. Giới thiệu về “Nhận diện ký tự quang học” (Optical Character Recognition) và sự hỗ trợ của framework Vision – Apple 13](#_Toc60954699)

[*2.4.1.* *Nhận diện ký quang học (Optical Character Recognition)* 13](#_Toc60954700)

[*2.4.2.* *Giới thiệu về Vision Framework của Apple* 15](#_Toc60954703)

[2.5. Mô hình Clean Architecture và các nguyên tắc lập trình S.O.L.I.D 16](#_Toc60954704)

[*2.5.1.* *Mô hình Clean Architecture* 16](#_Toc60954705)

[*2.5.2.* *Nguyên tắc lập trình S.O.L.I.D* 18](#_Toc60954713)

[Chương 3: Phân tích và thiết kế hệ thống 20](#_Toc60954714)

[3.1 Phân tích hệ thống 20](#_Toc60954715)

[*3.1.1.* *Xác định yêu cầu hệ thống* 20](#_Toc60954719)

[*3.1.2.* *Xây dựng bảng thuật ngữ* 21](#_Toc60954724)

[*3.1.3.* *Xác định và mô tả các ca sử dụng của hệ thống* 22](#_Toc60954725)

[*3.1.4.* *Xây dựng kịch bản* 27](#_Toc60954726)

[3.2. Thiết kế hệ thống 46](#_Toc60954727)

[*3.2.1.* *Thiết kế các biểu đồ tuần tự* 46](#_Toc60954728)

[*3.2.2.* *Sơ đồ lớp pha thiết kế* 57](#_Toc60954729)

[*3.2.3.* *Thiết kế cơ sở dữ liệu* 58](#_Toc60954730)

[Chương 4: Xây dựng ứng dụng 59](#_Toc60954731)

[4.1 Yêu cầu hệ thống 59](#_Toc60954732)

[*4.1.1.* *Yêu cầu đối với việc phát triển* 59](#_Toc60954733)

[*4.1.2.* *Yêu cầu đối với thiết bị* 59](#_Toc60954734)

[4.2. Một số hình ảnh của ứng dụng 60](#_Toc60954735)

[*4.2.1.* *Giao diện chức năng Đăng nhập* 60](#_Toc60954741)

[*4.2.2.* *Màn hình chính và thanh chức năng chính* 60](#_Toc60954742)

[*4.2.3.* *Giao diện Quản lý giao dịch* 61](#_Toc60954743)

[*4.2.4.* *Giao diện Quản lý sự kiện* 61](#_Toc60954744)

[*4.2.5.* *Giao diện Quản lý ngân sách* 62](#_Toc60954745)

[*4.2.6.* *Giao diện Chức năng thống kê* 62](#_Toc60954746)

[*4.2.7.* *Giao diện một số chức năng của Tùy chỉnh* 63](#_Toc60954747)

[Kết luận 64](#_Toc60954748)

[1. Đánh giá đề tài 64](#_Toc60954749)

[2. Định hướng phát triển 65](#_Toc60954750)

[Tài liệu tham khảo 66](#_Toc60954751)

**DANH MỤC BẢNG**

[Bảng 1. 1 Mô tả yêu cầu phi chức năng của hệ thống 5](#_Toc61013508)

[Hình 2. 1 Biểu tượng của nhà phát triển Apple 6](#_Toc61013513)

[Hình 2. 2 Sơ đồ mô hình Clean Architecture 16](#_Toc61013514)

[Bảng 3. 1 Bảng thuật ngữ cho hệ thống 21](#_Toc61013515)

[Bảng 3. 2 Kịch bản các ca sử dụng của hệ thống 25](#_Toc61013516)

[Bảng 3. 3 Kịch bản ca sử dụng đăng ký thành viên 27](#_Toc61013517)

[Bảng 3. 4 Kịch bản ca sử dụng đăng nhập 28](#_Toc61013518)

[Bảng 3. 5 Kịch bản ca sử dụng đăng xuất 29](#_Toc61013519)

[Bảng 3. 6 Kịch bản ca sử dụng quên mật khẩu 30](#_Toc61013520)

[Bảng 3. 7 Kịch bản ca sử dụng chỉnh sửa số dư 31](#_Toc61013521)

[Bảng 3. 8 Kịch bản ca sử dụng thêm giao dịch 31](#_Toc61013522)

[Bảng 3. 9 Kịch bản ca sử dụng xem thông tin giao dịch 32](#_Toc61013523)

[Bảng 3. 10 Kịch bản ca sử dụng sửa thông tin giao dịch 32](#_Toc61013524)

[Bảng 3. 11 Kịch bản ca sử dụng xóa giao dịch 33](#_Toc61013525)

[Bảng 3. 12 Kịch bản ca sử dụng thêm sự kiện 34](#_Toc61013526)

[Bảng 3. 13 Kịch bản ca sử dụng xem thông tin sự kiện 34](#_Toc61013527)

[Bảng 3. 14 Kịch bản ca sử dụng sử thông tin sự kiện 35](#_Toc61013528)

[Bảng 3. 15 Kịch bản ca sử dụng xóa sự kiện 36](#_Toc61013529)

[Bảng 3. 16 Kịch bản ca sử dụng thay đổi trạng thái sự kiện 36](#_Toc61013530)

[Bảng 3. 17 Kịch bản ca sử dụng xem danh sách giao dịch của sự kiện 37](#_Toc61013531)

[Bảng 3. 18 Kịch bản ca sử dụng thêm ngân sách 38](#_Toc61013532)

[Bảng 3. 19 Kịch bản ca sử dụng xem thông tin ngân sách 38](#_Toc61013533)

[Bảng 3. 20 Kịch bản ca sử dụng sửa thông tin ngân sách 39](#_Toc61013534)

[Bảng 3. 21 Kịch bản ca sử dụng xóa ngân sách 39](#_Toc61013535)

[Bảng 3. 22 Kịch bản ca sử dụng xem danh sách giao dịch của ngân sách 40](#_Toc61013536)

[Bảng 3. 23 Kịch bản ca sử dụng thống kê theo loại chi tiêu 41](#_Toc61013537)

[Bảng 3. 24 Kịch bản ca sử dụng thống kê theo danh mục chi tiêu 41](#_Toc61013538)

[Bảng 3. 25 Kịch bản ca sử dụng chỉnh sửa thông tin cá nhân 42](#_Toc61013539)

[Bảng 3. 26 Kịch bản ca sử dụng quét hóa đơn 43](#_Toc61013540)

[Bảng 3. 27 Kịch bản ca sử dụng quy đổi ngoại tệ 44](#_Toc61013541)

[Bảng 3. 28 Kịch bản ca sử dụng chế độ du lịch 45](#_Toc61013542)

[Bảng 3. 29 Kịch bản ca sử dụng tìm kiếm ATM 45](#_Toc61013543)

**DANH MỤC HÌNH ẢNH**

[Hình 2. 1 Biểu tượng của nhà phát triển Apple 6](#_Toc60959203)

[Hình 2. 2 Sơ đồ mô hình Clean Architecture 16](#_Toc60959204)

[Hình 3. 1 Biểu đồ ca sử dụng tổng quát 22](#_Toc60959214)

[Hình 3. 2 Biểu đồ phân rã ca sử dụng quản lý giao dịch 23](#_Toc60959215)

[Hình 3. 3 Biểu đồ phân rã ca sử dụng quản lý ngân sách 23](#_Toc60959216)

[Hình 3. 4 Biểu đồ phân rã ca sử dụng quản lý sự kiện 24](#_Toc60959217)

[Hình 3. 5 Biểu đồ phân rã ca sử dụng thống kê 24](#_Toc60959218)

[Hình 3. 6 Biểu đồ phân rã ca sử dụng tùy chỉnh 25](#_Toc60959219)

[Hình 3. 7 Biểu đồ tuần tự chức năng đăng ký 46](#_Toc60959220)

[Hình 3. 8 Biểu đồ tuần tự chức năng đăng nhập 46](#_Toc60959221)

[Hình 3. 9 Biểu đồ tuần tự chức năng tạo giao dịch 47](#_Toc60959222)

[Hình 3. 10 Biểu đồ tuần tự chức năng chức năng sửa thông tin giao dịch 47](#_Toc60959223)

[Hình 3. 11 Biểu đồ tuần tự chức năng chức năng xóa giao dịch 48](#_Toc60959224)

[Hình 3. 12 Biểu đồ tuần tự chức năng tạo sự kiện 48](#_Toc60959225)

[Hình 3. 13 Biểu đồ tuần tự chức năng sửa thông tin sự kiện 49](#_Toc60959226)

[Hình 3. 14 Biểu đồ tuần tự chức năng xóa sự kiện 49](#_Toc60959227)

[Hình 3. 15 Biểu đồ tuần tự chức năng xem danh sách giao dịch trong sự kiện 50](#_Toc60959228)

[Hình 3. 16 Biểu đồ tuần tự chức năng tạo ngân sách 50](#_Toc60959229)

[Hình 3. 17 Biểu đồ tuần tự chức năng sửa thông tin ngân sách 51](#_Toc60959230)

[Hình 3. 18 Biểu đồ tuần tự chức năng xóa ngân sách 51](#_Toc60959231)

[Hình 3. 19 Biểu đồ tuần tự chức năng xem danh sách giao dịch trong ngân sách 52](#_Toc60959232)

[Hình 3. 20 Biểu đồ tuần tự chức năng thông kê theo loại chi tiêu 52](#_Toc60959233)

[Hình 3. 21 Biểu đồ tuần tự chức năng thống kê theo danh mục 53](#_Toc60959234)

[Hình 3. 22 Biểu đồ tuần tự chức năng cập nhật thông tin người dùng 53](#_Toc60959235)

[Hình 3. 23 Biểu đồ tuần tự chức năng thêm danh mục 54](#_Toc60959236)

[Hình 3. 24 Biểu đồ tuần tự chức năng sửa thông tin danh mục 54](#_Toc60959237)

[Hình 3. 25 Biểu đồ tuần tự chức năng xóa danh mục 55](#_Toc60959238)

[Hình 3. 26 Biểu đồ tuần tự chức năng quét hóa đơn 55](#_Toc60959239)

[Hình 3. 27 Biểu đồ tuần tự chức năng quy đổi ngoại tệ 56](#_Toc60959240)

[Hình 3. 28 Biểu đồ tuần tự chức năng chế độ du lịch 56](#_Toc60959241)

[Hình 3. 29 Sơ đồ lớp pha thiết kế 57](#_Toc60959242)

[Hình 3. 30 Sơ đồ thiết kế cơ sở dữ liệu 58](#_Toc60959243)

[Hình 4. 1 Giao diện chức năng đăng nhập 60](#_Toc60959244)

[Hình 4. 2 Màn hình chính và thanh chức năng chính 60](#_Toc60959245)

[Hình 4. 3 Giao diện quản lý giao dịch 61](#_Toc60959246)

[Hình 4. 4 Giao diện quản lý sự kiện 61](#_Toc60959247)

[Hình 4. 5 Giao diện quản lý ngân sách 62](#_Toc60959248)

[Hình 4. 6 Giao diện chức năng thống kê 62](#_Toc60959249)

[Hình 4. 7 Giao diện một số chức năng của Tùy chỉnh 63](#_Toc60959250)

**Phụ lục từ ngữ Tiếng Anh**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ ngữ** | **Dịch nghĩa** |
| 1 | Functional Reactive Programming | Lập trình phản ứng |
| 2 | Optical Character Recognition | Nhận diện ký tự quang học |
| 3 | Integrated development environment | Môi trường phát triển tích hợp |
| 4 | Worldwide Developer Conference | Hội nghị các nhà phát triển toàn cầu |
| 5 | Producer | Nguồn phát |
| 6 | Consumer | Nguồn thu |
| 7 | Single Responsibility | Đơn trách nhiệm |
| 8 | Open/Closed | Đóng/Mở |
| 9 | Liskov Substitution | Thuyết thay thế Liskov |
| 10 | Interface Segregation | Sự tách biệt giao diện |
| 11 | Dependency Inversion | Sự tách biệt phụ thuộc mã nguồn |

**Lời giới thiệu**

Ngày nay, công nghệ thông tin đóng vai trò vô cùng quan trọng trong hầu hết các lĩnh vực. Do đó, việc học hỏi, trau dồi kiến thức là rất cần thiết. Trong những năm gần đây, sự xuất hiện của các thiết bị di động thông minh (smartphone), cùng với sự phát triển chóng mặt của các tập đoàn công nghệ lớn (Apple, Samsung, Google, …) đã và đang thay đổi cuộc sống của chúng ta. Việc ứng dụng công nghệ thông tin cũng như tận dụng tối đa Internet để kết nối mọi người, giúp cho sự trao đổi thông tin một các dễ dàng hơn.

Với việc bùng nổ của thời kỳ công nghệ số cùng cuộc cách mạng công nghệ 4.0 mọi lĩnh vực trong cuộc sống dần được trở nên số hóa, đặc biệt là nhu cầu về mua sắm, thanh toán qua các trang thương mại điện tử (Tiki, Shopee, Lazada, …). Nhờ vào sự hỗ trợ đến từ công nghệ việc mua sắm trở nên dễ dàng hơn bao giờ hết, bên cạnh đó là sự phát triển của các dịch vụ vận chuyển giúp việc mua sắm trở nên vô cùng tiện lợi, chỉ với vài thao tác nhấp trên màn hình điện thoại các mặt hàng đã có thể được giao đến tận nhà của người tiêu dùng. Vì thế việc tiêu xài không kiểm soát cùng sự mất cân bằng trong chi tiêu đã khiến rất nhiều người đau đầu. Từ thực tế đó việc phát triển ứng dụng **“Quản lý chi tiêu cá nhân”** có tên **MyWallet** được ra đời nhằm hỗ trợ người tiêu dùng trong việc cân bằng chi tiêu, cũng như là quản lý chi tiêu của bản thân một cách thông minh và thuận tiện nhất. Nội dung đồ án gồm 4 chương và kết luận:

**Chương 1:** Giới thiệu và tổng quan đề tài

**Chương 2:** Cáccông nghệ được sử dụng

**Chương 3:** Phân tích và thiết kế hệ thống

**Chương 4:** Xây dựng ứng dụng

Kết luận

**Chương 1****: Giới thiệu và tổng quan đề tài**

* 1. **Thực trạng vấn đề**

Khi việc mua sắm đang ngày càng trở nên thuận tiện hơn, cùng sự phát triển của các trang thương mại điện tử ở Việt Nam hiện nay (Tiki, Shopee, Lazada, …), cũng như các dịch vụ vận chuyển hỏa tốc, nhu cầu mua sắm của mọi người dần phát triển và trở thành một xu hướng không thể thiếu trong cuộc sống của mỗi người. Các chương trình khuyến mãi khủng, những ngày hội mua sắm trực tuyến được tổ chức liên tục, càng thúc đẩy việc mua sắm của người tiêu dùng. Từ đó việc chi tiêu bất hợp lý cũng như việc “vung tay quá trán” trong các cuộc mua sắm không kiểm soát khiến nhiều người vô cùng đau đầu và nguồn tiền của họ cũng ra đi một cách nhanh chóng.

Ngày nay những câu hỏi như “Sao vừa nhận lương mà đã hết?” hay “Không biết tháng này chi tiêu những gì mà đã lại hết tiền?” xuất hiện vô cùng nhiều trong cuộc sống, không chỉ đến từ các bạn trẻ, các mẹ các chị mà còn đến từ các đấng mày râu, đối tượng thường được xem là không quan tâm nhiều đến việc mua sắm.

Nhiều người cũng đã xây dựng nên những kế hoạch chi tiêu khoa học, rõ ràng tuy nhiên đa phần vẫn còn sử dụng việc ghi chép hay tổ chức trên giấy ghi chú hoặc sổ tay, khiến việc kiểm soát, kiểm tra vẫn còn gặp rất nhiều khó khăn và bất tiện. Với thực trạng đó để có thể giúp việc quản lý, ghi chép, kiểm soát chi tiêu trở nên thông minh, thuận tiện hơn **“Ứng dụng quản lý chi tiêu – MyWallet”** được ra đời, nhằm hỗ trợ tối đa người dùng trong việc kiểm soát chi tiêu của bản thân, gia đình.

## Xác định yêu cầu bài toán

Xây dựng ứng dụng **MyWallet** hỗ trợ tối đa người dùng trong việc quản lý chi tiêu, ghi chép chi tiêu theo từng ngày, phân loại chi tiêu theo từng danh mục một cách khoa học, bên cạnh đó còn thể hiện quá trình chi tiêu bằng biểu đồ giúp người dùng có một cái nhìn tổng quan hơn. Các ví chi tiêu được xây dựng giúp người dùng kiểm soát được lượng tiền sử dụng cho từng danh mục (ví dụ: Ví mua sắm, Ví sinh hoạt, …), cảnh báo khi người dùng chi tiêu quá hạn mức của ví để giúp người dùng có sự cân bằng cũng như khoa học hơn trong chi tiêu. Chế độ *Sự kiện* giúp người dùng có thể kiểm soát được chi tiêu một cách khoa học hơn thông qua việc quan sát danh sách chi tiêu cho từng sự kiện mà người dùng tham gia. Bên cạnh đó còn là các chức năng mở rộng giúp ứng dụng trở nên thông minh và hiện đại hơn như *chế độ du lịch, quy đổi ngoại tệ, tìm kiếm ngân hàng hoặc cây ATM xung quanh khu vực*. Đặc biệt đó là sự hỗ trợ đến từ trí tuệ nhân tạo với việc áp dụng **“Nhận diện ký tự quang học”** giúp người dùng có thể quét được hóa đơn mua sắm, hỗ trợ tối đa trong việc ghi chép chi tiêu, bớt các thao tác ghi chép thủ công để người dùng tiết kiệm thời gian hơn.

Ứng dụng **MyWallet** sẽ cho phép người dùng thực hiện các chức năng như *Tài khoản,* *Quản lý ví, Quản lý giao dịch, Xem báo cáo, Quản lý ngân sách, Quản lý chi tiêu theo sự kiện, Quét hóa đơn, Quy đổi ngoại tệ, Tìm ngân hàng hoặc ATM gần nhất, …*

* Chức năng tài khoản:
* Hệ thống cho phép người dùng đăng nhập thông qua các tài khoản mạng xã hội như Google hay Facebook.
* Hệ thống cho phép người dùng đăng ký bằng Email và mật khẩu.
* Hệ thống cho phép người dùng lấy lại mật khẩu thông qua tài khoản sử dụng trên ứng dụng.
* Chức năng quản lý ví:
* Mỗi tài khoản cho phép người dùng được tạo nhiều ví để quản lý chi tiêu với các mục đích cụ thể.
* Có nhiều loại ví với các chức năng khác nhau: Ví cơ bản, Ví liên kết, Ví tiết kiệm, …
* Chức năng quản lý giao dịch:
* Người dùng có thể ghi chép các khoản chi tiêu, các khoản thu nhập, vay nợ trong ngày một cách dễ dàng và thuận tiện.
* Giao dịch được tổ chức theo thời gian, danh mục giúp người dùng có thể theo dõi một cách khoa học và dễ dàng.
* Cho phép người dùng sửa đổi, xóa ghi chép, phù hợp với các nhu cầu khác nhau trong quá trình sử dụng
* Chức năng quản lý giao dịch theo ngân sách:
* Cho phép người dùng tạo các ví ngân sách theo từng nhu cầu chi tiêu trong một khoảng thời gian nhất định.
* Hệ thống sẽ đưa ra cảnh báo khi nhận thấy các ngân sách đã vượt quá hạn mức chi tiêu được đặt ra.
* Đưa ra gợi ý chi tiêu phù hợp với người dùng giúp người dùng tổ chức ngân sách một cách thông minh và khoa học nhất.
* Chức năng quản lý giao dịch theo sự kiện:
* Hệ thống cho phép người dùng tạo các sự kiện chi tiêu và theo dõi danh sách ghi chép chi tiêu được sử dụng trong sự kiện đó.
* Cho phép người dùng chỉnh sửa cũng như là xem lịch sử các sự kiện.
* Chức năng xem báo cáo chi tiêu:
* Hệ thống cho phép người dùng theo dõi được báo cáo, thống kê chi tiêu theo tháng, năm.
* Hệ thống cho phép người dùng được xem danh sách chi tiêu dựa theo loại chi tiêu (Chi – Tiêu).
* Hệ thống cho phép người dùng dược xem danh sách chi tiêu dựa theo từng danh mục có sẵn đc cung cấp từ ứng dụng.
* Các loại biểu đồ được sử dụng (biểu đồ cột, biểu đồ tròn) giúp người dùng có một cái nhìn tổng thể cũng khoa học nhất.
* Tổ hợp các chức năng mở rộng của ứng dụng:
* Chức năng quy đổi ngoại tệ: Cho phép người dùng quy đổi ngoại tệ từ VND sang các ngoại tệ phổ biến hiện (USD, Yen, Nhân dân tệ, Won, Bảng Anh, Euro, …).
* Chức năng tìm kiếm ngân hàng và ATM gần nhất: Cho phép người dùng có thể tìm kiếm vị trí của các cây ATM cũng như ngân hàng gần với vị trí của mình nhất.
* Chức năng quét hóa đơn: Sử dụng trí tuệ nhân tạo giúp người dùng trích xuất được thông tin trong các hóa đơn mua sắm, lưu thông tin hóa đơn một cách thuận tiện.
* Yêu cầu phi chức năng:

*Bảng 1. 1 Mô tả yêu cầu phi chức năng của hệ thống*

|  |  |  |
| --- | --- | --- |
| Yêu cầu phi chức năng | Mô tả | Chú thích |
| Thao tác | Môi trường vật lý và kỹ thuật mà ứng dụng hoạt động. | Hệ thống có thể tự động cập nhập dữ liệu.  Hệ thống có thể hoạt động trên hệ điều hành iOS với nhiều version khác nhau. |
| Hiệu năng | Tốc độ, khả năng và độ tin cậy của hệ thống. | Hệ thống sẵn sàng phục vụ người dùng bất cứ lúc nào.  Cơ sở dữ liệu của người dùng có thể được đồng bộ theo thời gian thực trên nhiều máy khác nhau. |
| Bảo mật | Ai có quyền truy cập hệ thống cho một chức năng nào đó. | Người dùng có thể truy cập vào toàn bộ các chức năng của ứng dụng. |

* 1. **Mục tiêu**

Xây dựng được hệ thống quản lý với đầy đủ các chức năng nhằm đáp ứng nhu cầu của người dùng một cách tốt nhất, giúp người dùng thuận tiện hơn trong việc quản lý cũng như khoa học với trong việc ghi chép chi tiêu.

Các công nghệ được sử dụng cần phải phù hợp nhằm hỗ trợ cho việc xây dựng ứng dụng, cũng như các vấn đề, bài toán được đặt ra trong ứng dụng một cách tối ưu và linh hoạt nhất.

Ứng dụng được tạo ra nhằm giúp người dùng tiết kiệm thời gian hơn, áp dụng được các công nghệ hiện có giúp cho ứng dụng trở nên thông minh và làm việc hiệu quả hơn. Không chỉ hoạt động ở độ chính xác cao mà còn kịp thời, đáp ứng được như cầu của người sử dụng.

* 1. **Đối tượng và phạm vi**
* Đối tượng:
* Mọi đối tượng đều có thể sử dụng ứng dụng, không phân biệt tuổi tác hay giới tính.
* Phạm vi:
* Toàn lãnh thổ Việt Nam

**Chương 2: Các công nghệ được sử dụng**

Chương này của đồ án sẽ tập trung giới thiệu về các công nghệ được sử dụng trong việc xây dựng ứng dụng. Tìm hiểu tổng quát về hệ điều hành iOS, ngôn ngữ lập trình Swift, các dịch vụ đến từ Firebase, triết lý lập trình phản ứng chức năng (Functional Reactive Programming) và sự hỗ trợ của trí tuệ nhân tạo trong việc **“Nhận diện ký tự quang học” (Optical Character Recognition)** được hỗ trợ bởi Apple với framework Vision. Ngoài ra là những quy tắc, quy chuẩn trong lập trình để giúp việc tạo ra sản phẩm có một quá trình xây dựng chuyên nghiệp và hiện đại nhất với **“Mô hình Clean Architecture”** và các nguyên tắc quan trọng lập trình **“S.O.L.I.D”**.

1. **Giới thiệu chung về hệ điều hành iOS và ngôn ngữ lập trình Swift**
   * 1. *Hệ điều hành iOS*

Hệ điều hành iOS lúc đầu được gọi là iPhoneOS và đến tháng 6 năm 2010 thì chính thức được đổi tên gọi là “iOS” như hiện nay. iOS là một hệ điều hành điện thoại di động được phát triển bởi Apple và phân phối độc quyền cho phần cứng của Apple. Ban đầu, iOS được công bố năm 2007 cho iPhone, sau đó được mở rộng để hỗ trợ các thiết bị khác của Apple như iPod Touch (tháng 9 năm 2007), iPad (tháng 1 năm 2010), iPad Mini (tháng 11 năm 2012) và thế hệ thứ hai của Apple Tivi trở đi (tháng 9 năm 2010).



Hình 2. 1 Biểu tượng của nhà phát triển Apple

Các ưu, nhược điểm của hệ điều hành iOS

* + *Ưu điểm:*
* Nền tảng ổn định, ứng dụng có khả năng tương thích cao. Ứng dụng trên iOS có thể hoạt động mượt mà vì chỉ được tối ưu hóa cho các thiết bị của Apple.
* Độ tin cậy và bảo mật cao.
* Chu kỳ ra mắt là một năm vì vậy các công nghệ sẽ được liên tục cập nhật.
* Ứng dụng phong phú, chất lượng cao và cập nhật nhanh hơn khi có phiên bản mới, có tính phổ biến cao
  + *Nhược điểm:*
* Không thể chạy nhiều ứng dụng cùng lúc trên màn hình.
* Trao đổi dữ liệu đều phải qua iTunes gây bất tiện và tốn thời gian vì đây là con đường duy nhất.
* Khả năng tùy chỉnh hạn chế.
* Giá thành thiết bị khá cao so với mặt bằng chung.

Kiến trúc của hệ điều hành iOS bao gồm các thành phần cơ bản:

* + **Các tiến trình (processes):** Là những tuyến riêng lẻ kết hợp với dữ liệu để thực hiện những tác vụ, như duy trì hệ thống, chuyển mạch gói dữ liệu, thực hiện giao thức định tuyến.
  + **Nhân (Kernel):** Cung cấp những dịch vụ cơ bản của hệ thống tùy thuộc vào iOS như: quản lý bộ nhớ, lập lịch các tiến trình, … Nó cung cấp quản lý tài nguyên phần cứng (CPU, bộ nhớ) cho các tiến trình.
  + **Bộ đệm gói (Packet buffer):** Cung cấp các bộ đệm toàn cục và kết hợp với chức năng quản lý bộ đệm để lưu trữ gói dữ liệu đang được chuyển mạch.
  + **Trình điều khiển thiết bị (Device drivers):** Làm chức năng điều khiển giao tiếp giữa phần cứng và thiết bị ngoại vi, giao tiếp giữa các tiến trình iOS, Kernel và phần cứng. Chúng cũng giao tiếp với các phần mềm chuyển mạch nhanh (Fast switching software).
  + **Phần mềm chuyển mạch nhanh:** Chức năng chuyển mạch gói dữ liệu cao.
    1. *Ngôn ngữ lập trình Swift*

Swift là một ngôn ngữ lập trình hướng đối tượng của Apple dành cho việc phát triển iOS và OS X, được giới thiệu tại hội nghị WWDC 2014. Swift được mong đợi sẽ tồn tại song song cùng Objective-C, ngôn ngữ lập trình hiện tại dành cho các hệ điều hành của Apple. Swift được thiết kế là một ngôn ngữ có khả năng phòng chống lỗi cao. Nó được biên dịch với trình biên dịch LLVM.

Swift được cho là mang điểm mạnh của cả 2 loại ngôn ngữ script và compile truyền thống. Trong một số thử nghiệm benchmark, Swift cho hiệu năng nhanh hơn Python và thậm chí là nhanh hơn cả Objective-C. Trên bộ IDE (phần mềm lập trình) Xcode của Apple, mã nguồn Swift được đồ thị hóa dựa trên tính năng Playground theo thời gian thực. Điều này có nghĩa rằng lập trình viên có thể chạy và test mã nguồn Swift một cách dễ dàng không kém gì Python.

Swift hiện nay đang mang những lợi ích rất lớn trong việc phát triển ứng dụng di động, khi các lập trình viên có thể kiểm tra công việc theo thời gian thực một cách nhanh chóng và dễ dàng. Ngoài ra việc tích hợp chương trình test cũng giúp lập trình viên tiết kiệm rất nhiều thời gian. Bên cạnh đó Swift với cú pháp rất dễ hiểu và dễ đọc không quá phức tạp và rối. Tuy nhiên Swift lại mang tuổi đời khá trẻ vì vậy khá khăn với những người mới bắt đầu, cần thời gian để làm quen và làm việc một cách nhuần nhuyễn, bên cạnh đó là khả năng quản lý bộ nhớ khá phức tạp.

* 1. **Giới thiệu về Firebase**
     1. *Giới thiệu chung*

Firebase là một nền tảng phát triển ứng dụng di động và web được phát triển bởi Firebase, Inc. vào năm 2011, sau đó được Google mua lại vào năm 2014. Tính đến tháng 10 năm 2018, nền tảng Firebase có 18 sản phẩm, được sử dụng bởi 1,5 triệu ứng dụng.

Firebase là một nền tảng phát triển ứng dụng di động và web. Họ cung cấp rất nhiều công cụ và dịch vụ để phát triển ứng dụng chất lượng, rút ngắn thời gian phát triển và phát triển cơ sở người dùng mà không cần quan tâm đến hạ tầng phần cứng.

Firebase là sự kết hợp giữa nền tảng cloud với hệ thống máy chủ cực kì mạnh mẽ của Google, Firebase cung cấp cho chúng ta những API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng Database.

* + 1. *Các dịch vụ của Firebase*

Firebase cung cấp các dịch vụ nền tảng cho các lập trình viên, sử dụng để xây dựng ứng dụng cũng như hỗ trợ tối ưu hóa, tối đa hóa ứng dụng của mình. Với nhiều dịch vụ chất lượng cao đi kèm với mức giá phải chăng. Firebase đã và đang không chỉ là sự lựa chọn hàng đầu cho các lập trình viên đơn thân (single developer) hay các công ty khởi nghiệp (start-up), mà các công ty, tổ chức lớn có tên tuổi cũng sử dụng Firebase để xây dựng các tính năng, các chương trình mới, cũng như chuyển đổi các dịch vụ trước đây sang hệ thống của Firebase. Chẳng hạn như Shazam, Fabulous và cả chính Google nữa, khi nền tảng nhắn tin Allo được xây dựng trên nền tảng Firebase Realtime Database

1. *Firebase Auth*

Firebase Auth là một dịch vụ có thể xác thực người dùng chỉ sử dụng mã phía máy khách. Nó hỗ trợ các nhà cung cấp cho phép người dùng đăng nhập từ các mạng xã hội lớn như Facebook, GitHub, Twitter và Google. Ngoài ra, nó còn bao gồm một hệ thống quản lý người dùng, theo đó các nhà phát triển cho phép xác thực người dùng bằng Email và mật khẩu đăng nhập được lưu trữ với Firebase.

1. *Firebase Realtime Database*

Một kiến trúc hệ cơ sở dữ liệu cho phép các lập trình viên truy vấn và thay đổi dữ liệu theo thời gian thực. Tốc độ truy vấn nhanh, cài đặt đơn giản với các API được cung cấp đến từ Firebase. Mô hình dữ liệu được lưu trữ theo dạng cây, NoSQL vì thế khá đơn giản trong việc tổ chức dữ liệu, khả năng mở rộng và tính linh hoạt cao.

1. *Firebase Cloud Messaging*

Trước đây được gọi là Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) là một giải pháp đa nền tảng cho các tin nhắn và thông cho Android, iOS và các ứng dụng web, từ năm 2016 FCM được sử dụng miễn phí.

* + 1. *Hệ quản trị cơ sở dữ liệu Firebase Realtime Database*

Cơ sở dữ liệu thời gian thực Firebase là cơ sở dữ liệu NoSQL được lưu trữ trên đám mây cho phép bạn lưu trữ và đồng bộ hóa giữa những người dùng của bạn trong thời gian thực.

Cơ sở dữ liệu thời gian thực thực sự chỉ là một đối tượng JSON lớn mà các lập trình viên có thể quản lý trong thời gian thực.

Chỉ với một API duy nhất, cơ sở dữ liệu Firebase cung cấp cho ứng dụng của bạn cả giá trị hiện tại của dữ liệu và mọi cập nhật cho dữ liệu đó. Đồng bộ hóa thời gian thực giúp người dùng của bạn dễ dàng truy cập dữ liệu của họ từ bất kỳ thiết bị nào, có thể là web hoặc di động. Cơ sở dữ liệu thời gian thực cũng giúp người dùng của bạn hợp tác, trao đổi với nhau. Một lợi ích đáng kinh ngạc khác của cơ sở dữ liệu thời gian thực là nó cung cấp SDK di động và web, cho phép bạn xây dựng ứng dụng của mình mà không cần máy chủ.

Khi người dùng của bạn ngoại tuyến, SDK cơ sở dữ liệu thời gian thực sử dụng bộ đệm cục bộ trên thiết bị để phục vụ và lưu trữ các thay đổi. Khi thiết bị trực tuyến, dữ liệu trên thiết bị sẽ được tự động đồng bộ hóa. Cơ sở dữ liệu thời gian thực cũng có thể tích hợp với Firebase Authentication để cung cấp quy trình xác thực đơn giản và trực quan.

* 1. **Triết lý lập trình thế hệ mới – Lập trình phản ứng chức năng (Functional Reactive Programming)**
     1. *Giới thiệu về lập trình phản ứng chức năng (Functional Reactive Programming - FRP)*

Lập trình phản ứng chức năng là mô hình lập trình hướng tới luồng dữ liệu và sự lan truyền thay đổi. Trong FRP, ta có một loại dữ liệu thể hiện được “giá trị thay đổi theo thời gian”, ta có thể áp dụng các hàm cơ bản đặc trưng của FRP (ví dụ: map, reduce, combine, …). Ví dụ, trong mô hình lập trình thông thường – lập trình mệnh lệnh (Imperative Programming) khi ta có *a = b + c*, giá trịn của *a* thu được chính là tổng giá trị của *b* và *c* tại thời điểm chạy câu lệnh, sau thời điểm đó giá trị *b, c* có thể thay đổi nhưng sẽ không làm thay đổi giá trị của *a*. Đối với FRP thì khác, giá trị của *a* luôn được tự động cập nhật theo giá trị của *b* và *c*. Việc này giống như bảng tính Excel vậy, trong một ô nếu ta ghi công thức *=A1 + B1* thì khi giá trị của ô *A1* hay *B1*  được cập nhật, thì giá trị ô bảng tính vừa rồi cũng được cập nhật theo.

**Observable – Trái tim của lập trình phản ứng**. Về cơ bản, “giá trị thay đổi theo thời gian” trong FRP có hai loại giống như bất cứ loại dữ liệu nào trong tự nhiên: rời rạc và liên tục. Tuy nhiên, do bản chất của máy tính, việc cài đặt hỗ trợ mô hình loại dữ liệu liên tục gặp khó khăn hơn, hầu hết trong các framework hiện nay mới chỉ hỗ trợ cho các cập nhật rời rạc, hướng sự kiện. Trong mô hình này, dữ liệu được mô hình thành các tín hiệu chứa giá trị hiện tại, thay đổi một cách rời rạc gọi là sự kiện. Các sự kiện này thực tế là một dòng sự kiện bất đồng bộ. Như vậy dùng FRP là ta lập trình với dòng dữ liệu bất đồng bộ.

Chúng ta đã quen với việc làm việc với dòng dữ liệu từ lâu. Nguồn xuất dòng dữ liệu được gọi là producer, và consumer sẽ sử dụng dòng dữ liệu đó. Để giải quyết được bài toán producer – consumer này chúng ta đã có hai mẫu thiết kế vô cùng nổi tiếng đó là ***Iterator*** và ***Observable***. Đối với ***Iterator***, *consumer* khi cần dữ liệu thì mới yêu cầu *producer*, giống như khi ta hết thực phẩm mới ra siêu thị mua đồ vậy. Còn đối với ***Observable***, *consumer* đăng ký với *producer*, khi đó *producer* sẽ chuyển trực tiép cho *consumer* ngay khi có sản phẩm. Việc này tương tự như chúng ta đặt báo hằng tháng; sau khi đăng ký, mỗi khi đến kỳ có báo mới là có nhân viên giao báo tới tận nhà chúng ta. Trong FRP, dòng sự kiện được coi là các *Observable*. Các *Observer* đăng ký với *Observable*, và phản ứng với các dữ liệu được *Observable* phát ra. Ví dụ một *Observable* sẽ tạo ra một dòng sự kiện như sau:

Nếu như chi có vậy thì FRP chẳng khác biệt gì so với ta lập trình bình thường và sử dụng các mẫu thiết kế như ***Observable***. Trong FRP, *Observable* được mô hình như một *Collection* bình thường, từ đó ta có thể áp dụng các hàm cho *Collection*, biến đổi từ một *Observable* này sang một *Observable* khác.

So sánh sự khác biệt giữa lập trình mệnh lệnh truyền thống (Imperative Programming) với lập trình phản ứng chức năng (Functional Reactive Programming)

Bảng 2. 1 So sánh lập trình mệnh lệnh và lập trình phản ứng

|  |  |
| --- | --- |
| Lập trình mệnh lệnh | Lập trình phản ứng |
| Mô tả cách chương trình vận hành | Mô tả thứ mà chương trình nên đạt được |
| Sử dụng mệnh lệnh để thay đổi trạng thái chương trình | Dựa trên ý tưởng về phi trạng thái (stateless) |
| Sử dụng các câu lệnh điều kiện, vòng lặp | Đánh giá chức năng và tạo ra các thể hiện thay vì mệnh lệnh |

* + 1. *Lập trình phản ứng trong Swift với RxSwift*

RxSwift là một phiên bản Reactive Extension được viết bằng ngôn ngữ Swift. ReactiveX là sự kết hợp của những ý tưởng hay nhất của các mẫu thiết kế ***Observable*** và ***Iterator*** và ***Lập trình phản ứng chức năng***. RxSwift giúp cho việc cài đặt ý tưởng của ***Lập trình phản ứng chức năng*** trở nên đơn giản hơn rất nhiều, với sự hỗ trợ mạnh mẽ đến từ thư viện RxSwift. Loại bỏ hầu hết các dòng code mệnh lệnh ở trong chương trình mà thay vào đó là cái hàm đặc trưng cơ bản, giúp code ngắn đi và tiết kiệm nhiều thời gian hơn cho lập trình viên.

1. *Observable Sequences*
   * Mọi hoạt động trong **RxSwift** từ việc đăng ký hay xử lý sự kiện đều thông qua một **Observable Sequences** trong **RxSwift**, các kiểu dữ liệu như **Arrays**, **Strings** hoặc **Dictionary** sẽ được chuyển sang **Observable Sequences**. Ta có thể tạo ra **Observable Sequences** của bất kỳ kiểu đối tượng nào tuân theo **Sequence Protocol** của thư viện cơ bản có trong Swift.
   * Observable Sequences sẽ gửi các sự kiện có trong thời gian tồn tại của nó. Trong RxSwift thì một sự kiện sẽ là một dữ liệu dàng Enum với 3 trạng thái:

* **.next(value: T)**: xảy ra khi một hay một tập hợp các giá trị được bổ sung thêm vào **Observable Sequences**. Nó sẽ gửi sự kiện **next** đến cho các đối tượng đã đăng ký.
* **.error(error: Error)**: xảy ra khi có lỗi và sẽ kết thúc luôn **Observable Sequences**.
* **.completed**: nếu **Observable Sequences** hoạt động bình thường và kết thúc mà không có lỗi sự kiện **completed** sẽ được gọi và phát đến cho các đối tượng đăng ký.
  + Để hủy đăng ký nhận sự kiện ta có thể sử dụng hàm **dispose** hoặc thêm đăng ký vào **DisposeBag** để tự động hủy đăng ký, việc này giúp tránh được những lỗi xảy ra liên quan đến bộ nhớ của hệ điều hành.

1. *Subjects trong RxSwift*

**Subject** là một dạng đặc biệt của **Observable Sequence**, ta có thể đăng ký và bổ sung thêm thành phần trong một **Subject**. Trong **RxSwift** có 4 loại **Subject** cơ bản:

* + **PublishSubject**: ta sẽ nhận được tất cả các sự kiện phát sinh sau khi đăng ký.
  + **BehaviourSubject**: sẽ gửi sự kiện phát sinh gần đây nhất và những sự kiện phát sinh sau khi đăng ký.
  + **ReplaySubject**: cho phép ta cấu hình sẽ nhận được bao nhiêu sự kiện phát sinh gần đây nhất khi có một đăng ký mới được thiết lập.
  + **Variable**: là một thể hiện đặc biệt của **BehaviourSubject** giúp việc sử dụng dược dễ dàng, có thể được coi như là một biến bình thường.
  1. **Giới thiệu về “Nhận diện ký tự quang học” (Optical Character Recognition) và sự hỗ trợ của framework Vision – Apple**
     1. *Nhận diện ký quang học (Optical Character Recognition)*

Ứng dụng công nghệ chuyên dùng để có thể trích xuất được chữ có trong các tệp tin ở dạng ảnh. Được biết đến là một công cụ quét kỹ thuật số chuyên dùng để nhận dạng các ký tự, chữ viết tay, hay chữ đánh máy, công nghệ này chuyên dùng để truyền tải, nhập dữ liệu. Đặc biệt, ở OCR có khả năng kỹ thuật số với nhiều dạng tài liệu khác nhau: hóa đơn, hộ chiếu, danh thiếp, tài liệu, văn bản hành chính, … Đến với OCR, những văn bản số hóa, tìm kiếm và chỉnh sửa sẽ được thực hiện điện tử. Đồng thời, chúng giúp tiết kiệm không gian lưu trữ tài liệu bằng việc hiển thị trực tiếp.

Với OCR khi một trang in hay viết tay được quét sẽ được lưu dưới định dạng tệp là TIF. Chúng ta có thể dễ dàng đọc hình ảnh này dưới màn hình hiển thị. Tuy nhiên, tùy thuộc vào máy tính, nó sẽ tồn tại một loạt hình ảnh có chấm trắng hoặc đen. Lúc này, công nghệ nhìn vào từng dòng của hình ảnh để xác định các dấu có khớp nhau hay không.

OCR được ứng dụng trong rất nhiều lĩnh vực của đời sống, tiêu biểu có thể kể đến:

* + **Trợ giúp người mắc bệnh mù và khiếm thị:** Vì sao nói OCR có khả năng hỗ trợ người mắc bệnh mù và khiếm thị, những người có ảnh hưởng về thị giác. Bởi, OCR có khả năng quét và đọc các từ trên màn hình. Từ đây, những người gặp vấn đề thị giác có thể dễ dàng hiểu được chúng.
  + **Tìm kiếm và thực hành dữ liệu:** Thực tế, OCR có khả năng tạo ra những nội dung văn bản riêng của quét tài liệu, giúp chúng ta có thể dễ dàng tìm kiếm và xác định vị trí tài liệu dựa trên từ khóa. Đồng thời, OCR cũng cho phép nhanh nhẹn hơn trong việc chỉnh sửa và xử lý văn bản.
  + **Cập nhật dữ liệu nhanh chóng:** OCR đảm bảo chức năng cải thiện hiệu quả và nhanh chóng cho công việc văn phòng cũng như năng suất cao. Bởi, hầu hết trong quá trình làm việc ở văn phòng như cầu quét tài liệu ngày một lớn. Điều này sẽ giúp tiết kiệm thời gian, đồng thời cập nhật dữ liệu nhanh, chính xác nhất cho người dùng.

Bên cạnh những lợi ích to lớn mà OCR mang lại, cũng không thể tránh khỏi những hạn chế mà công nghệ còn gặp phải:

* + Đa số những phần mềm, ứng dụng chứa OCR chỉ có khả năng nhận dạng chính xác khoảng 80-90% dựa vào hình ảnh rõ nét.
  + Với những hình ảnh truy cập có màu nền và màu chữ khá tương đồng (không có sự chênh lệch lớn) điều này khiến OCR gặp khó khăn trong nhận dạng. Và tất nhiên, kết quả nhận dạng sẽ không được khả thi cho lắm.
  + Ngoài ra, ở thời điểm hiện tại khi các ngôn ngữ ngày càng trở nên phong phú thì công nghệ OCR lại chưa thể đáp ứng đa ngôn ngữ, hiện nay OCR chưa hỗ trợ cho tất cả các ngôn ngữ.
  + Ứng dụng của OCR là rất cao tuy nhiên vẫn cần phải có sự kết hợp thêm các công nghệ hỗ trợ khác như Nhận diện hình ảnh, Phân loại, … Để giúp cho OCR có thể đạt được hiệu quả tốt nhất.
  1. 2. *Giới thiệu về Vision Framework của Apple*

Được ra mắt lần đầu tiên tại Hội nghị của Các nhà phát triển trên toàn cầu của Apple (Worldwide Developer Conference) năm 2017, đây là một trong những thư viện được đánh giá là mạnh mẽ nhất trong các thư viện được giới thiệu vào năm đó. Với Vision framework chúng ta có thể dễ dàng cài đặt được những kỹ thuật của thị giác máy tính vào ứng dụng của bản thân mà không cần có quá nhiều yêu cầu về khả năng kỹ thuật. Với Vision framework chúng ta có thể xây dựng nên một ứng dụng có khả năng thực hiện một số lượng không nhỏ các công việc mạnh mẽ yêu cầu kỹ thuật cao như nhận diện khuôn mặt và các đặc điểm nổi bật (nụ cười, mũi, mắt trái – phải, …), nhận diện mã sản phẩm, phân loại vật thể có trong ảnh, theo dấu của vật thể chuyển động.

Với ai đã từng làm quen với ngôn ngữ lập trình Swift chắc hẳn cũng đã biết qua về một số thư viện liên quan đến trí tuệ nhân tạo trước khi có Vision (ví dụ: Core Image, AV Capture), trong một sự so sánh, các nhà phát triển đã đưa ra được sự đánh giá về Vision so với các thư viện trước nó.

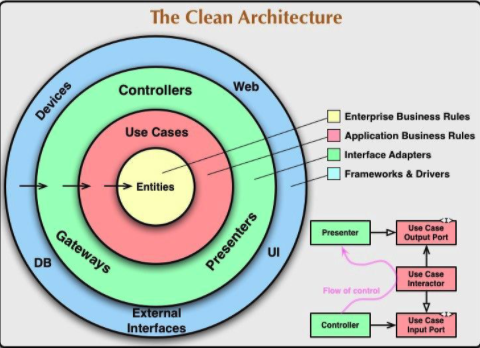
Bảng 2. 2 So sánh các Framework xử lý ảnh Apple

|  |  |  |  |
| --- | --- | --- | --- |
|  | Vision | CoreImage | AV Capture |
| Độ chính xác | Tốt nhất | Tốt hơn | Tốt |
| Thời gian xử lý | Nhanh | Nhanh hơn | Nhanh nhất |
| Năng lượng tiêu hao | Tốt | Tốt hơn | Tốt nhất |
| Khả năng sẵn sàng | iOS, macOS, tvOS | iOS, macOS, tvOS | iOS |

* 1. **Mô hình Clean Architecture và các nguyên tắc lập trình S.O.L.I.D**
     1. *Mô hình Clean Architecture*

Clean Architecture được xây dựng dựa trên tư tưởng “độc lập” kết hợp với các nguyên lý thiết kế hướng đối tượng (đại diện tiêu biểu là Dependency Injection). Độc lập ở đây nghĩa là việc dự án không bị phụ thuộc vào framework và các công cụ sử dụng trong quá trình kiểm thử.

Kiến trúc Clean Architecture chia thành 4 layer với một quy tắc phụ thuộc, các lớp bên trong không nên biết bất kỳ điều gì về các layer bên ngoài. Điều này có nghĩa là nó có quan hệ phụ nên “hướng” vào bên trong, quan hệ theo hướng một chiều, để các lớp không có sự ảnh hưởng đến nhau khi thay đổi tại một lớp.



Hình 2. 2 Sơ đồ mô hình Clean Architecture

***Entities:*** là khái niệm dung để mô tả các logic nghiệp vụ. Đây là lớp quan trọng nhất, là nơi chúng ta giải quyết các vấn đề - mục đích xây dựng app. Layer này không chứa bất kỳ một framework nào, chúng ta có thể chạy nó mà không cần giả lập. Nó giúp chúng ta dễ dàng kiểm thử, bảo trì và phát triển phần logic nghiệp vụ.

***Usecase:*** chứa các nguyên tắc liên quan trực tiếp tới ứng dụng cục bộ, là lớp liên lạc trực tiếp với dữ liệu (cục bộ hoặc ngoại tuyến).

***Interface Adapter:*** tập hợp các adapter phục vụ quá trình tương tác với các công nghệ.

***Framework and Drivers:*** chứa các công cụ về cơ sở dữ liệu và các framework, thông thường bạn sẽ không phải lập trình nhiều ở tầng này. Tuy nhiên cần chắc chắn về mức ưu tiên sử dụng các công cụ này trong dự án.

Thông thường thì một ứng dụng của bạn có thể có tùy ý số lượng các layer. Thường thì một ứng dụng thường sẽ có 3 lớp:

* + **Outer:** Lớp cài đặt: là nơi mà tất cả mọi thứ của framework xảy ra, điều này bao gồm tất cả các công cụ sử dụng trong dự án từ giao diện người dùng đến cơ sở dữ liệu.
  + **Middle:** hoạt động như một lớp để kết nối lớp **Outer** và **Inner**.
  + **Inner:** nơi thao tác trực tiếp với dữ liệu.

Đối với mỗi lớp ở trên các lớp chính đều có trách nhiệm chuyển đổi các model thành các model của lớp thấp hơn trước khi các lớp thấp hơn có thể sử dụng được chúng. Tại sao chúng ta cần phải chuyển đổi model? Ví dụ, các model thuộc về logic nghiệp vụ của dự án có thể không thích hợp cho việc hiển thị lên giao diện cho người dùng cuối, lúc đó chúng ta có thể sẽ phải kết hợp nhiều model logic nghiệp vụ cùng một lúc. Vì vậy, ta cần tạo nên một lớp để có thể dễ dàng hơn cho việc hiển thị lên giao diện người dùng. Việc chuyển đổi model là cần thiết để phù hợp với chức năng của từng lớp.

Lợi ích của Clean Architecture mang lại là rất lớn, và đang dần trở thành xu hướng của việc lập trình ứng dụng di động:

* + Mạch lạc – dễ xem.
  + Linh hoạt – thể hiện ở khả năng độc lập, không phụ thuộc vào framework, cơ sử dữ liệu hay máy chủ của ứng dụng.
  + Dễ dàng kiểm thử.

Tuy nhiên bên cạnh đó, Clean Architecture cũng mang lại một số hạn chế nhất định:

* + Không thể sử dụng framework theo cách mỳ ăn liền – do luật Dependency Inversion.
  + Khó áp dụng với người mới tìm hiểu.
  + Tuy các thành phần đã có sự độc lập nhưng lại phải viết thêm code, cũng như thêm các thành phần dùng để liên kết.
  + Cần xem xét trình độ cũng như khả năng của đội phát triển, vì sẽ cần mất thời gian trong việc nghiên cứu và áp dụng.

1. 5. 2. *Nguyên tắc lập trình S.O.L.I.D*

Một bộ các nguyên tắc tạo thành một tiêu chuẩn trong lập trình giúp cho lập trình viên tạo nên mã nguồn sạch, hạn chế lỗi và có khả năng bảo trì, tránh được những thiết kế xấu. Giúp cho mã nguồn trở nên linh hoạt, logic, dễ quản lý và kiểm soát hơn, việc kiểm tra cũng như đọc mã nguồn cũng trở nên đơn giản hơn.

**S – Single Responsibility -** *Một lớp chỉ nên nắm giữ một trách nhiệm duy nhất*

Nếu một lớp nắm giữ quá nhiều trách nhiệm, nó sẽ tăng khả năng tạo ra lỗi bởi vì khi thực hiện thay đổi tới một trong những thành phần của nó, nó có thể ảnh hưởng tới cả những thành phần khác mà lập trình viên có thể không biết. Mục tiêu ở đây là tách biệt hành vi, để nếu có lỗi xuất hiện trong quá trình thay đổi nó sẽ không ảnh hưởng đến các thành phần không liên quan.

**O – Open/Closed -** *Các lớp nên có khả năng mở rộng, nhưng không nên bị chỉnh sửa*

Thay đổi hành vi hiện  tại của một lớp sẽ gây ảnh hưởng lên toàn bộ hệ thống nơi có sử dụng class đó. Nếu chúng ta muốn một lớp thực hiện nhiều hơn chức năng hơn, ý tưởng tiếp cận ở đây sẽ là thêm vào các chức năng mới chứ không phải thay đổi nó. Mục tiêu ở đây là để có thể mở rộng hành vi của một lớp mà không thay đổi hành vi hiện tại của nó, làm tránh việc tạo ra lỗi khi lớp được sử dụng.

**L – Liskov Substitution -** *Một lớp con được tạo nên từ một lớp cha cần có khả năng hoạt động tương tự lớp cha nếu được thay thế*

Khi một lớp con không thể thực hiện những hành vi giống như lớp cha của nó, việc đó có thể tạo ra lỗi. Nếu có một lớp và tạo thêm một lớp khác từ nó, nó sẽ trở thành mối liên kết cha con, lớp con nên có khả năng thực hiện được tất cả các hành vi mà lớp cha có thể thực hiện (Kế thừa trong lập trình hướng đối tượng). Lớp con nên có khả năng xử lý cùng yêu cầu và đưa ra được kết quả giống hoặc đưa ra kết quả có kiểu dữ liệu cùng loại với kết quả do lớp cha trả ra. Nếu lớp con không thể thực hiện các điều trên, rất có thể lớp con đã bị thay đổi hoàn toàn và điều đó là vi phạm quy tắc. Mục tiêu ở đây là để đảm bảo tính ổn định khi lớp cha và lớp con có thể được dùng một cách giống nhau và không gây ra lỗi.

**I – Interface Segregation** *- Người dùng không nên bị ép buộc phải phụ thuộc vào những chức năng mà họ không sử dụng*

Khi một lớp phải thực hiện các hành vi không hữu dụng, nó sẽ gây ra lãng phí và có thể dẫn đến việc tạo lỗi không mong muốn nếu lớp đó không có khả năng trong việc thực hiện hành vi đó. Một lớp chỉ nên thực hiện các hành vi cần thiết và đảm bảo nhiệm vụ của nó. Bất kỳ hành vi nào khác nên được lược bỏ hoàn toàn hoặc chuyển đến một nơi khác nếu nó sẽ được sử dụng bởi một lớp khác trong tương lai. Mục tiêu ở đây là chia nhỏ tập hợp công việc thành một tập hợp nhỏ hơn từ đó một lớp chỉ thực thi tập hợp công việc được yêu cầu.

**D – Dependency Inversion -** *Các mô-đun cấp cao không nên phụ thuộc vào các mô-đun cấp thấp. Cả hai nên phụ thuộc vào lớp trừu tượng. Lớp trừu tượng không nên phụ thuộc vào chi tiết. Chi tiết nên được phát triển từ lớp trừu tượng*

Chú thích:

* + Mô-đun cấp cao: lớp thực hiện một hành vi với một công cụ.
  + Mô-đun cấp thấp: công cụ cần thiết để có thực hiện hành vi đó.
  + Lớp trừu tượng: đại diện cho một interface (delegate) để kết nối 2 lớp trên.
  + Chi tiết: cách thức mà công cụ hoạt động.

Một lớp không nên hợp nhất với công cụ nó dùng để thực hiện hành vi. Mặt khác, nó nên được hợp nhất với interface (delegate) mà cho phép công cụ kết nối đến lớp đó. Bên cạnh đó, cả lớp và interface (delegate) không nên biết cách thức mà công cụ hoạt động. Tuy nhiên, công cụ cần đảm bảo được những đặc tả đến từ interface (delegate). Mục tiêu ở đây là giảm tải các phụ thuộc của các lớp bậc cao với các lớp bậc thấp bằng cách sử dụng interface (delegate).

# **Chương 3: Phân tích và thiết kế hệ thống**

1. **Phân tích hệ thống**
3. 1. 1. *Xác định yêu cầu hệ thống*
4. *Đối tượng sử dụng*

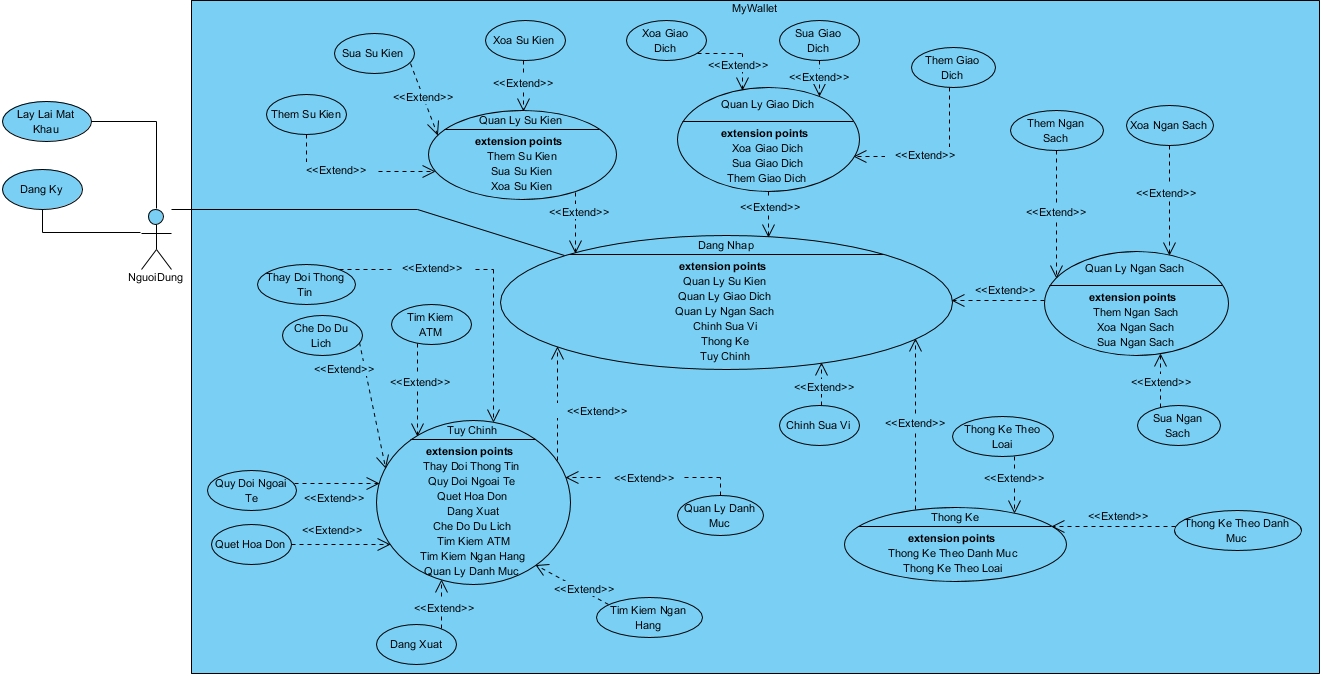
Tất cả người dùng sử dụng hệ điều hành iOS với thiết bị điện thoại iPhone có nhu cầu quản lý chi tiêu cá nhân.

1. *Mô tả chức năng người dùng*
   * + Chức năng đăng nhập, đăng xuất, đăng ký người dùng: Dùng để xác thực đối tượng sử dụng với hệ thống. Người dùng có thể đăng ký bằng email và mật khẩu. Bên cạnh đó người dùng có thể đăng nhập bằng các tài khoản trên các trang mạng xã hội bao gồm Facebook và Google.
     + Chức năng quản lý số dư: Người dùng sau khi đăng nhập có thể điều chỉnh số dư của mình ứng với nguồn tiền của mình đang có để có thể bắt đầu tiến hành ghi chép các giao dịch sau đó.
     + Chức năng quản lý giao dịch: Người dùng sau khi đăng nhập có thể thêm, sửa, xóa các giao dịch, cũng như có thể theo dõi giao dịch đã được sắp xếp sẵn có trong ngày.
     + Chức năng quản lý ngân sách: Người dùng có thêm, sửa, xóa ngân sách cho một nhóm giao dịch để có thể kiểm soát được hạn mức chi tiêu đối với từng danh mục cụ thể. Ứng dụng sẽ đưa ra cảnh báo khi ngân sách bị vượt quá hạn mức, giúp người dùng điều chỉnh việc chi tiêu một cách hợp lý.
     + Chức năng quản lý sự kiện: Người dùng có thể tạo một sự kiện để quản lý được số lượng chi tiêu dành cho sự kiện đó, người dùng có thể thêm, sửa, xóa và điều chỉnh sự kiện.
     + Chức năng thống kê: Người dùng có thể xem thống kê các giao dịch theo ngày hoặc theo tháng, dựa vào các biểu đồ được xây dựng bởi ứng dụng, qua đó có một cái nhìn tổng quát nhất về việc chi tiêu của bản thân trong một khoản thời gian.
     + Chức năng hỗ trợ: Ở đây người dùng có thể sử dụng thêm một số chức năng mở rộng được cung cấp bởi ứng dụng như: *Chế độ du lịch, Quy đổi ngoại tệ, Quét hóa đơn thông minh, Chỉnh sửa thông tin cá nhân.*
3. 1. 2. *Xây dựng bảng thuật ngữ*

Bảng 3. 1 Bảng thuật ngữ cho hệ thống

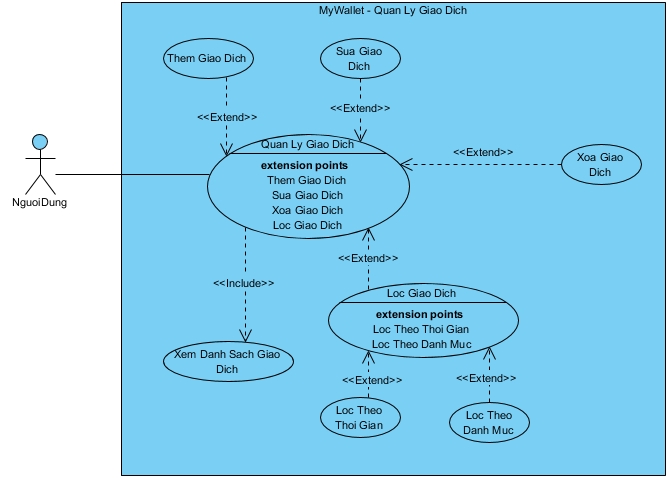
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tiếng Anh** | **Tiếng Việt** | **Mô tả** |
| 1 | Balance | Số dư | Số dư trong tài khoản của người dùng |
| 2 | Transaction | Giao dịch | Các ghi chú được người dùng ghi chép |
| 3 | Budget | Ngân sách | Danh sách các ghi chú chi tiêu được tổ chức theo danh mục |
| 4 | Event | Sự kiện | Danh sách các ghi chú chi tiêu được tổ chức theo sự kiện |
| 5 | Login | Đăng nhập | Là hoạt động người dùng truy cập vào hệ thống |
| 6 | Logout | Đăng xuất | Là hoạt động người dùng thoát khỏi hệ thống |
| 7 | Report | Thống kê | Thống kê dữ liệu chi tiêu của người dùng theo tháng |
| 8 | Settings | Tùy chỉnh | Các chức năng mở rộng của ứng dụng |
| 9 | Bill Scanner | Quét hóa đơn | Tính năng quét hóa đơn thông minh |
| 10 | Categories | Danh mục | Tính năng quản lý danh mục |
| 11 | Travel Mode | Chế độ du lịch | Chế độ du lịch giúp người dùng tổ chức chi tiêu trong sự kiện |
| 12 | Currency | Ngoại tệ | Chức năng hỗ trợ người dùng quy đổi ngoại tệ |

* + 1. *Xác định và mô tả các ca sử dụng của hệ thống*
* Biểu đồ ca sử dụng tổng quát



Hình 3. 1 Biểu đồ ca sử dụng tổng quát

* Biểu đồ phân rã ca sử dụng quản lý giao dịch



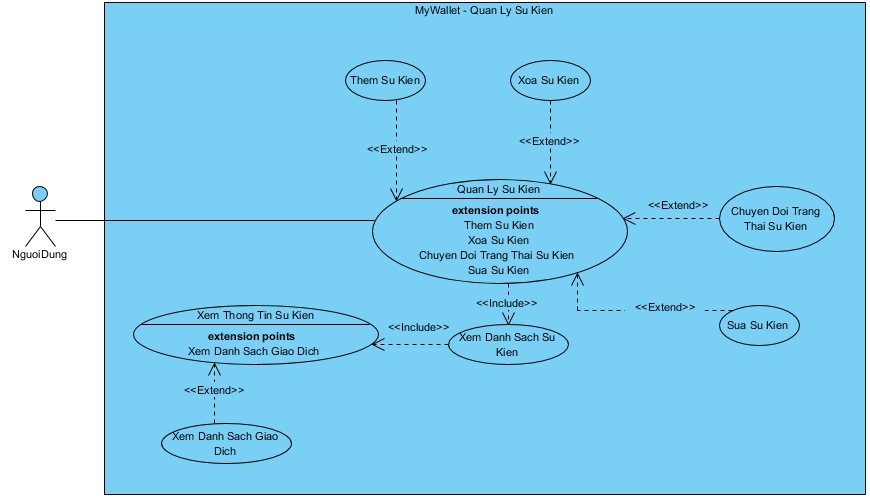
Hình 3. 2 Biểu đồ phân rã ca sử dụng quản lý giao dịch

* Biểu đồ phân rã ca sử dụng quản lý ngân sách



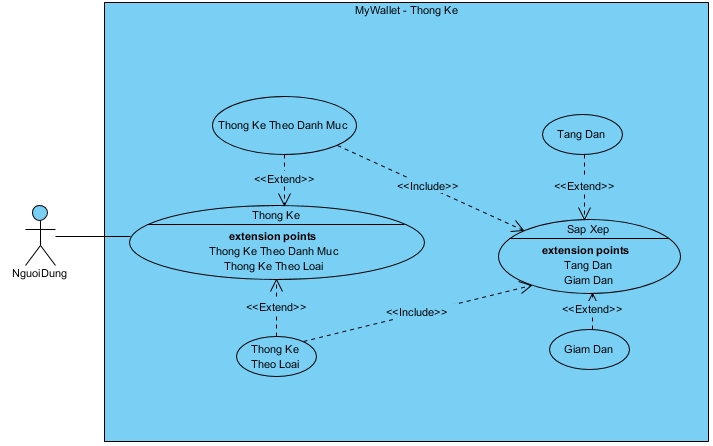
Hình 3. 3 Biểu đồ phân rã ca sử dụng quản lý ngân sách

* Biểu đồ phân rã ca sử dụng quản lý sự kiện



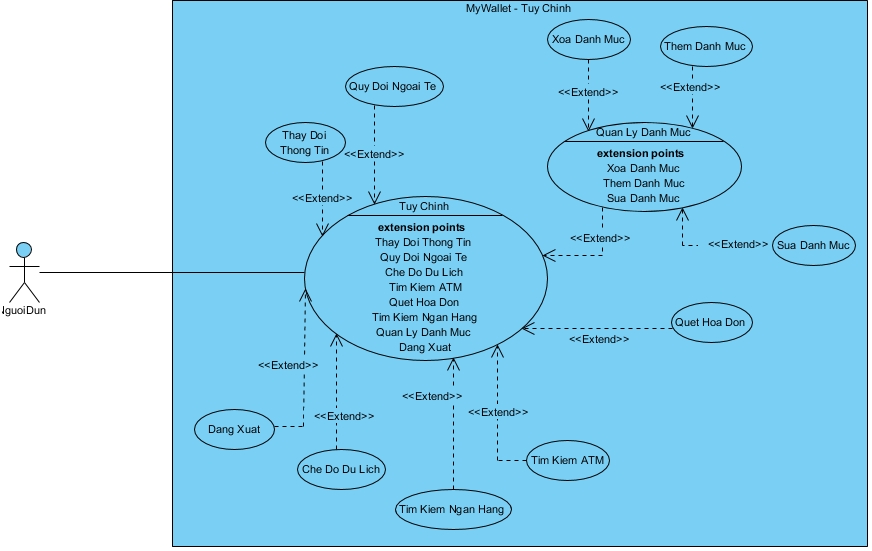
Hình 3. 4 Biểu đồ phân rã ca sử dụng quản lý sự kiện

* Biểu đồ phân rã ca sử dụng thống kê



Hình 3. 5 Biểu đồ phân rã ca sử dụng thống kê

* Biểu đồ phân rã ca sử dụng tùy chỉnh



Hình 3. 6 Biểu đồ phân rã ca sử dụng tùy chỉnh

Dựa vào sơ đồ phân rã các ca sử dụng ta xác định được các ca sử dụng của ứng dụng

Bảng 3. 2 Kịch bản các ca sử dụng của hệ thống

|  |  |  |
| --- | --- | --- |
| STT | Tên ca sử dụng | Mô tả |
| U1 | Đăng ký thành viên | Người dùng đăng ký tài khoản để đăng nhập và sử dụng ứng dụng |
| U2 | Đăng nhập | Người dùng đăng nhập tài khoản để sử dụng ứng dụng |
| U3 | Đăng xuất | Người dùng đăng xuất tài khoản của mình khỏi ứng dụng |
| U4 | Quên mật khẩu | Người dùng lấy lại mật khẩu của tài khoản |
| U5 | Chỉnh sửa số dư | Người dùng cập nhật số dư trong tài khoản sử dụng với ứng dụng |
| U6 | Thêm giao dịch | Người dùng thêm giao dịch vào danh sách, giao dịch được lưu xuống CSDL |
| U7 | Xem thông tin giao dịch | Người dùng xem thông tin chi tiết về một giao dịch đã thêm trong danh sách |
| U8 | Sửa thông tin giao dịch | Người dùng sửa thông tin chi tiết về một giao dịch đã thêm trong danh sách |
| U9 | Xóa giao dịch | Người dùng xóa giao dịch đã thêm trong danh sách |
| U10 | Thêm sự kiện | Người dùng thêm sự kiện vào danh sách sự kiện chi tiêu để quản lý |
| U11 | Xem thông tin sự kiện | Người dùng xem thông tin chi tiết của một sự kiện: đã chi bao nhiêu, thời gian còn lại bao nhiêu ngày... Ngoài ra người dùng có thể sửa, xóa thông tin sự kiện, đánh dấu sự kiện đã kết thúc hoặc chưa kết thúc |
| U12 | Sửa thông tin sự kiện | Người dùng chỉnh sửa thông tin của một sự kiện trong danh sách sự kiện |
| U13 | Xóa sự kiện | Người dùng xóa sự kiện đã thêm khỏi danh sách sự kiện |
| U14 | Thay đổi trạng thái của sự kiện | Người dùng thay đổi trạng thái của một sự kiện là đã kết thúc hay chưa kết thúc |
| U15 | Xem danh sách giao dịch của sự kiện | Người dùng xem danh sách các giao dịch đã ghi chú trong một sự kiện |
| U16 | Thêm ngân sách | Người dùng tạo ngân sách để kiểm soát các khoản chi đối với một nhóm giao dịch nào đó |
| U17 | Xem thông tin ngân sách | Người dùng xem thông tin chi tiết của ngân sách như: đã chi bao nhiêu, ngân sách còn lại bao nhiêu và biểu đồ hiển thị chi tiêu... Ngoài ra người dùng có thể sửa, xóa ngân sách hoặc xem danh sách các giao dịch có trong sự kiện |
| U18 | Sửa thông tin ngân sách | Người dùng có thể sửa thông tin của một ngân sách trong danh sách ngân sách |
| U19 | Xóa ngân sách | Người dùng xóa ngân sách khỏi danh sách ngân sách |
| U20 | Xem danh sách giao dịch của ngân sách | Người dùng xem danh sách các giao dịch đã ghi chú trong một ngân sách |
| U21 | Thống kê theo loại chi tiêu | Người dùng xem biểu đồ chi tiêu được phân theo loại: Chi và Tiêu |
| U22 | Thống kê theo danh mục chi tiêu | Người dùng xem biểu đồ chi tiêu được phân theo từng danh mục chi tiết |
| U23 | Chỉnh sửa thông tin cá nhân | Người dùng cập nhật thông tin cá nhân của mình với ứng dụng |
| U24 | Quét hóa đơn | Người dùng có thể quét hóa đơn thông minh để thêm vào danh sách giao dịch mà không cần ghi chú |
| U25 | Quy đổi ngoại tệ | Người dùng tiến hành quy đổi VND ra các tỉ giá của các đồng ngoại tệ phổ biến |
| U26 | Chế độ du lịch | Người dùng bật chế độ du lịch, các khoản chi tiêu sẽ được tự động cập nhật vào sự kiện có trong chế độ du lịch |
| U27 | Tìm kiếm ATM | Người dùng tìm kiếm các cây ATM trong khu vực gần mình |

* + 1. *Xây dựng kịch bản*
* U1: Đăng ký thành viên

Bảng 3. 3 Kịch bản ca sử dụng đăng ký thành viên

|  |  |
| --- | --- |
| Tên ca sử dụng | Ca sử dụng đăng ký thành viên |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Tài khoản chưa được đăng ký trước đó |
| Đảm bảo tối thiểu | Hệ thống không tạo tài khoản với các thông tin đã nhập và có thể quay lại màn hình trước |
| Đảm bảo thành công | Tài khoản được tạo thành công và có thể đăng nhập vào hệ thống |
| Kích hoạt | Người dùng chọn chức năng đăng ký tài khoản tại màn hình đăng nhập |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện đăng nhập 2. Người dùng chọn chức năng đăng ký tài khoản 3. Hệ thống hiển thị giao diện đăng ký tài khoản 4. Người dùng nhập thông tin tài khoản cần đăng ký và chọn đăng ký 5. Hệ thống tạo tài khoản với thông tin đã cung cấp 6. Hệ thống thông báo đăng ký thành công và trở về giao diện đăng nhập |
| Ngoại lệ | 5.a. Không có kết nối internet 5.a.1 Hệ thống hiển thị thông báo không có internet  5.b Email đã được đăng ký  5.b.1 Hệ thống hiển thị thông báo email đã được đăng ký |

* U2: Đăng nhập

Bảng 3. 4 Kịch bản ca sử dụng đăng nhập

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng đăng nhập |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng ký tài khoản của hệ thống và chưa đăng nhập vào ứng dụng |
| Đảm bảo tối thiểu | Người dùng không đăng nhập được vào hệ thống |
| Đảm bảo thành công | Người dùng đăng nhập thành công vào hệ thống |
| Kích hoạt | Người dùng mở ứng dụng |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện đăng nhập 2. Người dùng đăng nhập    1. Người dùng đăng nhập bằng email và mật khẩu       1. Người dùng nhập email và mật khẩu và ấn nút đăng nhập    2. Người dùng đăng nhập bằng google       1. Người dùng click chọn đăng nhập bằng google       2. Hệ thống hiển thị giao diện đăng nhập google       3. Người dùng đăng nhập vào tài khoản google và đồng ý       4. Hệ thống google xác nhận đăng nhập thành công và trả token về cho ứng dụng 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống thông báo đăng nhập thành công |
| Ngoại lệ | 3.a. Hệ thống thông báo sai thông tin đăng nhập  3.a.1. Hệ thống yêu cầu người dùng nhập lại thông tin  3.a.2. Người dùng nhập lại thông tin và chọn đăng nhập |

* U3: Đăng xuất

Bảng 3. 5 Kịch bản ca sử dụng đăng xuất

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng đăng xuất |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào ứng dụng |
| Đảm bảo tối thiểu | Người dùng không bị đăng xuất khỏi ứng dụng và có thể quay lại màn hình |
| Đảm bảo thành công | Người dùng đăng xuất thành công khỏi hệ thống |
| Kích hoạt | Người dùng chọn chức năng đăng xuất trong phần tùy chỉnh của ứng dụng |
| Chuỗi sự kiện | 1. Người dùng click phần thông tin trong menu 2. Hệ thống hiển thị giao diện Thông tin tài khoản 3. Người dùng chọn “Đăng xuất” 4. Hệ thống đăng xuất người dùng ra khỏi hệ thống 5. Hệ thống thông báo đăng xuất thành công và trở về màn hình đăng nhập |
| Ngoại lệ | 3.a Hệ thống đăng xuất thất bại  3.a.1. Hệ thống hiển thị thông báo đăng xuất thất bại, vui lòng thử lại sau. |

* U4: Quên mật khẩu

Bảng 3. 6 Kịch bản ca sử dụng quên mật khẩu

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng quên mật khẩu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Tài khoản đã được đăng ký trên hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể trở lại màn hình trước |
| Đảm bảo thành công | Người dùng lấy lại được mật khẩu dùng để đăng nhập vào hệ thống |
| Kích hoạt | Người dùng chọn chức năng quên mật khẩu ở màn hình đăng nhập |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện quên mật khẩu 2. Người dùng nhập thông tin của tài khoản(email) sau đó nhấn lấy lại mật khẩu 3. Hệ thống kiểm tra email và gửi yêu cầu lấy lại mật khẩu về email mà người dùng đã cung cấp. 4. Người dùng làm theo hướng dẫn trong email để lấy lại mật khẩu |
| Ngoại lệ | 3.a Email người dùng cung cấp chưa được đăng ký tài khoản  3.a.1. Hệ thống hiển thị thông báo email chưa được đăng ký tài khoản vui lòng kiểm tra lại  3.a.2. Người dùng nhập lại thông tin email |

* U5: Chỉnh sửa số dư

Bảng 3. 7 Kịch bản ca sử dụng chỉnh sửa số dư

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng chỉnh sửa số dư |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Số dư của tài khoản không được chỉnh sửa và người dùng có thể quay về màn hình trước |
| Đảm bảo thành công | Số dư của ví thay đổi thành công và lưu vào CSDL |
| Kích hoạt | Người dùng chọn điều chỉnh số dư tại màn hình chính |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện chỉnh sửa số dư với số dư hiện tại 2. Người dùng chỉnh sửa số dư và nhấn lưu 3. Hệ thống tiến hành kiểm tra thông tin 4. Hệ thống cập nhật số dư vào CSDL 5. Hệ thống thông báo cập nhập số dư thành công và quay về màn hình chính với số dư đã thay đổi. |
| Ngoại lệ | 3.a Hệ thống hiển thị số dư nhập vào không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại số dư  3.a.2 Người dùng nhập lại số dư |

* U6: Thêm giao dịch

Bảng 3. 8 Kịch bản ca sử dụng thêm giao dịch

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng thêm giao dịch |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin về giao dịch không được thêm vào CSDL và người dùng có thể trở lại màn hình trước |
| Đảm bảo thành công | Thông tin về giao dịch mới được thêm vào CSDL |
| Kích hoạt | Người dùng chọn vào nút thêm giao dịch tại màn hình chính |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện thêm giao dịch 2. Người dùng nhập thông tin giao dịch, chọn danh mục hoặc sự kiện mong muốn và chọn “Lưu” 3. Hệ thống kiểm tra thông tin giao dịch và xác nhận thông tin giao dịch hợp lệ 4. Hệ thống thêm thông tin giao dịch vào CSDL 5. Hệ thống thông báo thêm giao dịch thành công và quay lại màn hình trước |
| Ngoại lệ | 3.a Hệ thống hiển thị thông tin giao dịch không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại thông tin giao dịch  3.a.2 Người dùng nhập lại thông tin giao dịch |

* U7: Xem thông tin giao dịch

Bảng 3. 9 Kịch bản ca sử dụng xem thông tin giao dịch

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xem thông tin giao dịch |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Hiển thị thông tin chi tiết của giao dịch |
| Kích hoạt | Người dùng chọn vào giao dịch cần xem thông tin |
| Chuỗi sự kiện | 1. Hệ thống hiển thị các giao dịch 2. Người dùng tìm kiếm các giao dịch cần xem thông tin 3. Người dùng chọn vào giao dịch cần xem thông tin 4. Hệ thống hiển thị thông tin giao dịch cho người dùng |
| Ngoại lệ |  |

* U8: Sửa thông tin giao dịch

Bảng 3. 10 Kịch bản ca sử dụng sửa thông tin giao dịch

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng sửa thông tin giao dịch |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin giao dịch không bị sửa đổi và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về giao dịch được thay đổi trong CSDL |
| Kích hoạt | Chọn chức năng sửa thông tin giao dịch trong giao diện xem thông tin chi tiết của giao dịch |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện sửa thông tin giao dịch 2. Người dùng chỉnh sửa thông tin giao dịch và chọn “Lưu” 3. Hệ thống kiểm tra thông tin giao dịch và xác nhận thông tin giao dịch hợp lệ 4. Hệ thống cập nhập thông tin giao dịch vào cơ sở dữ liệu 5. Hệ thống thông báo chỉnh sửa thông tin giao dịch thành công và trở lại màn hình trước |
| Ngoại lệ | 3.a Hệ thống hiển thị thông tin giao dịch không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại thông tin giao dịch  3.a.2 Người dùng nhập lại thông tin giao dịch |

* U9: Xóa giao dịch

Bảng 3. 11 Kịch bản ca sử dụng xóa giao dịch

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xóa giao dịch |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Thông tin giao dịch không bị xóa khỏi CSDL và có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin giao dịch bị xóa khỏi CSDL |
| Kích hoạt | Người dùng chọn xóa giao dịch trong giao diện hiển thị thông tin chi tiết của giao dịch |
| Chuỗi sự kiện | 1. Hệ thống hiển thị thông báo xác nhận xóa giao dịch 2. Người dùng chọn xác nhận xóa giao dịch 3. Hệ thống xóa giao dịch khỏi CSDL 4. Hệ thống thông báo xóa giao dịch thành công và quay lại màn hình trước |
| Ngoại lệ |  |

* U10: Thêm sự kiện

Bảng 3. 12 Kịch bản ca sử dụng thêm sự kiện

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng thêm sự kiện |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin về sự kiện không được thêm vào CSDL và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về sự kiện được thêm vào CSDL |
| Kích hoạt | Chọn thêm sự kiện trong giao diện quản lý sự kiện |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện thêm sự kiện 2. Người dùng nhập thông tin về sự kiện 3. Hệ thống kiểm tra thông tin sự kiện và xác nhận chính xác 4. Hệ thống thêm sự kiện vào CSDL và thông báo thành công, quay lại màn hình trước |
| Ngoại lệ | 3.a Hệ thống hiển thị thông tin sự kiện không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại thông tin sự kiện  3.a.2 Người dùng nhập lại thông tin sự kiện |

* U11: Xem thông tin sự kiện

Bảng 3. 13 Kịch bản ca sử dụng xem thông tin sự kiện

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xem thông tin sự kiện |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về sự kiện được hiển thị cho người dùng |
| Kích hoạt | Chọn sự kiện cần xem thông tin trong giao diện sự kiện |
| Chuỗi sự kiện | 1. Hệ thống hiển thị danh sách sự kiện trong 2 phần:  * Phần sự kiện đang áp dụng * Phần sự kiện đã kết thúc  1. Người dùng tìm và chọn sự kiện muốn xem 2. Hệ thống hiển thị thông tin chi tiết của sự kiện |
| Ngoại lệ |  |

* U12: Sửa thông tin sự kiện

Bảng 3. 14 Kịch bản ca sử dụng sử thông tin sự kiện

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng sửa thông tin sự kiện |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin sự kiện không bị sửa đổi và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về sự kiện được thay đổi trong CSDL |
| Kích hoạt | Chọn chức năng sửa thông tin sự kiện trong giao diện xem thông tin chi tiết của sự kiện |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện sửa thông tin sự kiện 2. Người dùng chỉnh sửa thông tin sự kiện và chọn “Lưu” 3. Hệ thống kiểm tra thông tin sự kiện và xác nhận thông tin sự kiện hợp lệ 4. Hệ thống cập nhập thông tin sự kiện vào cơ sở dữ liệu 5. Hệ thống thông báo chỉnh sửa thông tin sự kiện thành công và trở lại màn hình trước |
| Ngoại lệ | 3.a Hệ thống hiển thị thông tin sự kiện không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại thông tin sự kiện  3.a.2 Người dùng nhập lại thông tin sự kiện |

* U13: Xóa sự kiện

Bảng 3. 15 Kịch bản ca sử dụng xóa sự kiện

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xóa sự kiện |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin về sự kiện không bị xóa khỏi hệ thống và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về sự kiện được xóa khỏi CSDL |
| Kích hoạt | Chọn xóa sự kiện trong giao diện xem thông tin chi tiết về sự kiện |
| Chuỗi sự kiện | 1. Người dùng chọn sự kiện muốn xóa 2. Hệ thống hiển thị thông tin chi tiết của sự kiện 3. Người dùng chọn “Xóa” 4. Hệ thống hiển thị cảnh báo xóa sự kiện 5. Người dùng xác nhận xóa sự kiện 6. Hệ thống xóa sự kiện khỏi cơ sở dữ liệu và quay lại màn hình trước |
| Ngoại lệ |  |

* U14: Thay đổi trạng thái của sự kiện

Bảng 3. 16 Kịch bản ca sử dụng thay đổi trạng thái sự kiện

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng thay đổi trạng thái của sự kiện |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin sự kiện không bị sửa đổi và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Trạng thái sự kiện được thay đổi và dữ liệu được cập nhập trong CSDL |
| Kích hoạt | Chọn chức năng “Đánh dấu sự kiện đã kết thúc/Đánh dấu sự kiện chưa kết thúc” trong giao diện xem thông tin chi tiết của sự kiện |
| Chuỗi sự kiện | 1. Hệ thống hiển thị thông tin chi tiết của sự kiện 2. Người dùng chọn “Đánh dấu sự kiện đã kết thúc/ Đánh dấu sự kiện chưa kết thúc” 3. Hệ thống thay đổi trạng thái của sự kiện và cập nhập cơ sở dữ liệu |
| Ngoại lệ |  |

* U15: Xem danh sách giao dịch của sự kiện

Bảng 3. 17 Kịch bản ca sử dụng xem danh sách giao dịch của sự kiện

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xem danh sách giao dịch của sự kiện |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Người dùng xem được danh sách giao dịch của sự kiện |
| Kích hoạt | Chọn chức năng “Xem danh sách giao dịch” trong giao diện xem thông tin chi tiết của sự kiện |
| Chuỗi sự kiện | 1. Hệ thống hiển thị thông tin chi tiết của sự kiện 2. Người dùng chọn “Xem danh sách giao dịch” 3. Hệ thống lấy dữ liệu và hiển thị danh sách giao dịch cho người dùng |
| Ngoại lệ | 3.a. Sự kiện chưa có giao dịch nào  3.a.1 Hệ thống hiển thị “Không có giao dịch” |

* U16: Thêm ngân sách

Bảng 3. 18 Kịch bản ca sử dụng thêm ngân sách

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng thêm ngân sách |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin về ngân sách không được thêm vào CSDL và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Các thông tin về ngân sách được thêm vào CSDL |
| Kích hoạt | Chọn thêm ngân sách trong menu của màn hình ngân sách |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện thêm ngân sách 2. Người dùng nhập thông tin về ngân sách và ấn “Lưu” 3. Hệ thống kiểm tra thông tin ngân sách và xác nhận thông tin chính xác 4. Hệ thống thêm ngân sách vào CSDL và thông báo cho người dùng, trở lại màn hình trước đó |
| Ngoại lệ | 3.a Hệ thống hiển thị thông tin ngân sách không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại thông tin ngân sách  3.a.2 Người dùng nhập lại thông tin ngân sách |

* U17: Xem thông tin ngân sách

Bảng 3. 19 Kịch bản ca sử dụng xem thông tin ngân sách

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xem thông tin ngân sách |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về ngân sách được hiển thị cho người dùng |
| Kích hoạt | Chọn ngân sách cần xem thông tin trong giao diện quản lý ngân sách |
| Chuỗi sự kiện | 1. Người dùng tìm và chọn ngân sách muốn xem 2. Hệ thống hiển thị thông tin chi tiết của ngân sách |
| Ngoại lệ |  |

* U18: Sửa thông tin ngân sách

Bảng 3. 20 Kịch bản ca sử dụng sửa thông tin ngân sách

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng sửa thông tin ngân sách |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin của ngân sách không bị thay đổi và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Các thông tin của ngân sách được cập nhập vào CSDL |
| Kích hoạt | Chọn chức năng “Sửa” trong giao diện hiển thị thông tin ngân sách |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện sửa ngân sách 2. Người dùng sửa thông tin ngân sách cần thiết và chọn “Lưu” 3. Hệ thống kiểm tra thông tin ngân sách và xác nhận thông tin hợp lệ 4. Hệ thống cập nhập thông tin ngân sách vào CSDL, trở lại màn hình trước |
| Ngoại lệ | 3.a Hệ thống hiển thị thông tin ngân sách không hợp lệ  3.a.1 Hệ thống yêu cầu người dùng nhập lại thông tin ngân sách  3.a.2 Người dùng nhập lại thông tin ngân sách |

* U19: Xóa ngân sách

Bảng 3. 21 Kịch bản ca sử dụng xóa ngân sách

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xóa ngân sách |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Các thông tin về ngân sách không bị xóa khỏi hệ thống và người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về ngân sách được xóa khỏi CSDL |
| Kích hoạt | Chọn “Xóa” trong giao diện xem thông tin chi tiết ngân sách |
| Chuỗi sự kiện | 1. Người dùng chọn ngân sách muốn xóa 2. Hệ thống hiển thị thông tin chi tiết của ngân sách 3. Người dùng chọn “Xóa” 4. Hệ thống hiển thị cảnh báo xóa ngân sách 5. Người dùng xác nhận xóa ngân sách 6. Hệ thống xóa ngân sách khỏi cơ sở dữ liệu và quay lại màn hình trước |
| Ngoại lệ |  |

* U20: Xem danh sách giao dịch của ngân sách

Bảng 3. 22 Kịch bản ca sử dụng xem danh sách giao dịch của ngân sách

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng xem danh sách giao dịch của ngân sách |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Người dùng xem được danh sách giao dịch của ngân sách |
| Kích hoạt | Chọn chức năng “Xem danh sách giao dịch” trong giao diện xem thông tin chi tiết của ngân sách |
| Chuỗi sự kiện | 1. Hệ thống hiển thị thông tin chi tiết của ngân sách 2. Người dùng chọn “Xem danh sách giao dịch” 3. Hệ thống lấy dữ liệu và hiển thị danh sách giao dịch cho người dùng |
| Ngoại lệ | 3.a. Ngân sách chưa có giao dịch nào  3.a.1 Hệ thống hiển thị giao diện “Không có giao dịch” |

* U21: Thống kê theo loại chi tiêu

Bảng 3. 23 Kịch bản ca sử dụng thống kê theo loại chi tiêu

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng thống kê theo loại chi tiêu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Người dùng xem thống kê theo chi tiêu của các giao dịch trong tháng |
| Kích hoạt | Chọn chức năng “Báo cáo” trong thanh chức năng của ứng dụng, chọn vào biểu đồ chi tiêu theo tháng |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện báo cáo 2. Người dùng chọn xem báo cáo theo chi tiêu với hai thành phần chính: Chi và Tiêu 3. Hệ thống hiển thị báo cáo chi tiêu với danh sách các giao dịch có trong tháng 4. Người dùng tiến hành sắp xếp chi tiêu theo mức độ tăng/giảm mức tiền 5. Hệ thống tiến hành sắp xếp và hiển thị lại danh sách chi tiêu theo mức độ tiêu dùng |
| Ngoại lệ | 3.a. Chưa có giao dịch nào trong tháng  3.a.1. Hệ thống hiển thị danh sách giao dịch rỗng |

* U22: Thống kê theo danh mục chi tiêu

Bảng 3. 24 Kịch bản ca sử dụng thống kê theo danh mục chi tiêu

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng thống kê theo danh mục chi tiêu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Người dùng xem thống kê theo chi tiêu của các giao dịch trong tháng theo từng danh mục |
| Kích hoạt | Chọn chức năng “Báo cáo” trong thanh chức năng của ứng dụng, chọn vào biểu đồ chi tiêu theo danh mục trong tháng |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện báo cáo 2. Người dùng chọn xem báo cáo chi tiêu theo danh mục 3. Hệ thống hiển thị báo cáo chi tiêu với danh sách các giao dịch có trong tháng theo danh mục 4. Người dùng tiến hành sắp xếp chi tiêu theo mức độ tăng/giảm mức tiền 5. Hệ thống tiến hành sắp xếp và hiển thị lại danh sách chi tiêu theo mức độ tiêu dùng của từng danh mục |
| Ngoại lệ | 3.a. Chưa có giao dịch nào trong tháng  3.a.1. Hệ thống hiển thị danh sách giao dịch rỗng |

* U23: Chỉnh sửa thông tin cá nhân

Bảng 3. 25 Kịch bản ca sử dụng chỉnh sửa thông tin cá nhân

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng chỉnh sửa thông tin cá nhân |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Người dùng cập nhật thành công thông tin |
| Kích hoạt | Chọn chức năng Tùy chỉnh trên thanh chức năng của giao diện, chọn thay đổi thay đổi thông tin |
| Chuỗi sự kiện | 1. Hệ thống hiển thị giao diện tùy chỉnh 2. Người dùng chọn thay đổi thông tin cá nhân 3. Hệ thống hiển thị giao diện thay đổi thông tin người dùng 4. Người dùng tiến hành nhập thông tin cá nhân và nhấn “Lưu” 5. Hệ thống thông báo thay đổi thông tin thành công và cập nhật thông tin cho người dùng, lưu thông tin người dùng vào CSDL |
| Ngoại lệ | 4.a. Thông tin nhập vào không hợp lệ hoặc không chuẩn quy tắc  4.a.1. Hệ thống yêu cầu người dùng nhập lại thông tin cá nhân  4.a.2. Người dùng tiến hành nhập lại thông tin cá nhân |

* U24: Quét hóa đơn

Bảng 3. 26 Kịch bản ca sử dụng quét hóa đơn

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng quét hóa đơn |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Thông tin về hóa đơn được trích xuất và lưu vào CSDL như một giao dịch |
| Kích hoạt | Chọn chức năng Tùy chỉnh trên thanh chức năng của giao diện, chọn quét hóa đơn |
| Chuỗi sự kiện | 1. Người dùng chọn quét hóa đơn từ tùy chỉnh 2. Hệ thống hiển thị giao diện quét hóa đơn 3. Người dùng chọn tải ảnh từ thư viện ảnh hoặc chụp ảnh hóa đơn và nhấn “Quét” 4. Hệ thống tiến hành quét và trích xuất thông tin 5. Người dùng nhấn “Lưu giao dịch” 6. Hệ thống thông báo giao dịch được lưu thành công |
| Ngoại lệ | 4.a. Không thể tiến hành quét hóa đơn do không đúng định dạng  4.a.1. Người dùng tiến hành chọn lại ảnh để quét  6.a. Giao dịch không được lưu do các thông tin không chính xác  6.a.1 Người dùng tiến hành chỉnh sửa thông tin và nhấn “Lưu giao dịch”  6.a.2 Hệ thống thông báo giao dịch được lưu thành công |

* U25: Quy đổi ngoại tệ

Bảng 3. 27 Kịch bản ca sử dụng quy đổi ngoại tệ

|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng quy đổi ngoại tệ |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Số tiền người dùng được quy đổi sang các ngoại tệ phổ biến với tỷ giá được cập nhật theo thời gian thực |
| Kích hoạt | Chọn chức năng Tùy chỉnh trên thanh chức năng của giao diện, chọn quy đổi ngoại tệ |
| Chuỗi sự kiện | 1. Người dùng chọn quy đổi ngoại tệ từ giao diện tùy chỉnh 2. Hệ thống hiển thị giao diện quy đổi ngoại tệ 3. Người dùng nhập vào số lượng tiền VND mong muốn quy đổi và nhấn “Quy đổi” 4. Hệ thống tiến hành tính toán và quy đổi 5. Hệ thống hiển thị các tỷ giá ngoại tệ đã được quy đổi |
| Ngoại lệ |  |

* U26: Chế độ du lịch

Bảng 3. 28 Kịch bản ca sử dụng chế độ du lịch

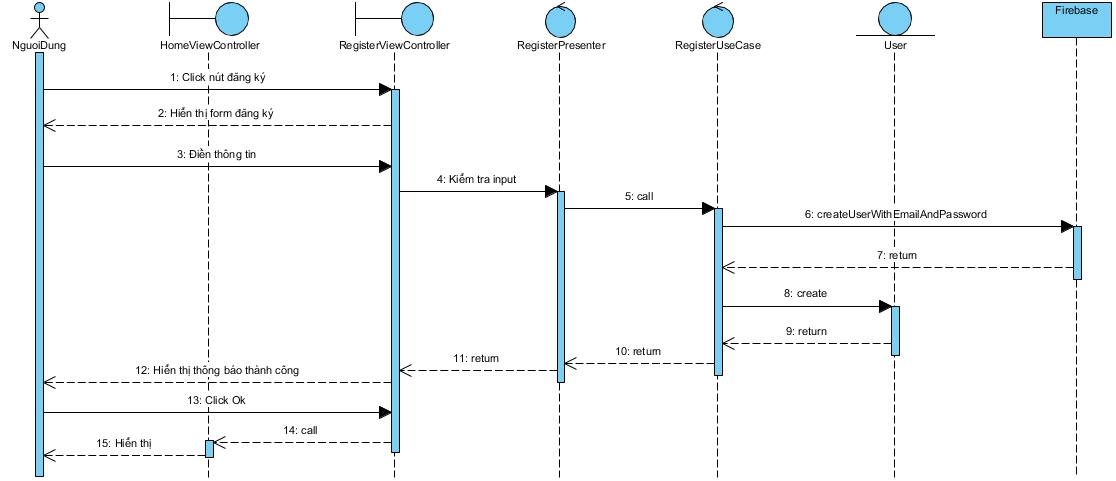
|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng chế độ du lịch |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Chế độ du lịch được bật, tất cả các giao dịch được đưa vào sự kiện cài đặt trong chế độ du lịch |
| Kích hoạt | Chọn chức năng Tùy chỉnh trên thanh chức năng của giao diện, chọn bật chế độ du lịch |
| Chuỗi sự kiện | 1. Người dùng mở chế độ du lịch từ tùy chỉnh 2. Hệ thống hiển thị giao diện chế độ du lịch 3. Người dùng chọn bật chế độ du lịch và chọn sự kiện cài đặt mong muốn (có thể thêm sự kiện từ đây), chọn “Lưu” 4. Hệ thống thông báo chế độ du lịch đã được bật cho người dùng |
| Ngoại lệ |  |

* U27: Tìm kiếm ATM

Bảng 3. 29 Kịch bản ca sử dụng tìm kiếm ATM

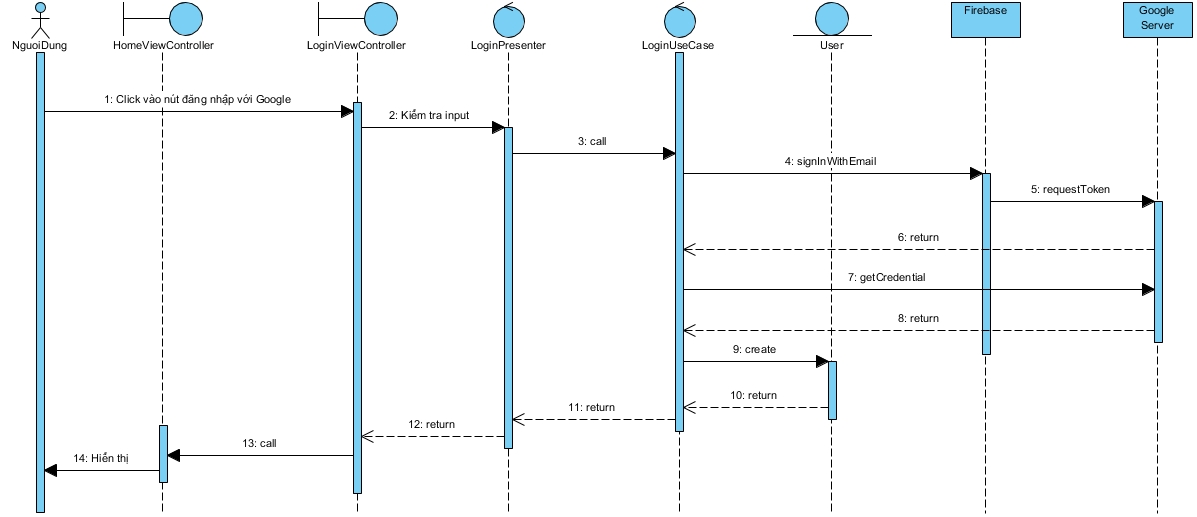
|  |  |
| --- | --- |
| Tên Ca sử dụng | Ca sử dụng Tìm kiếm ATM |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công vào hệ thống |
| Đảm bảo tối thiểu | Người dùng có thể quay lại màn hình trước |
| Đảm bảo thành công | Các ATM có trong khu vực quanh người dùng được hiển thị trên bản đồ |
| Kích hoạt | Chọn chức năng Tùy chỉnh trên thanh chức năng của giao diện, chọn tìm kiếm ATM |
| Chuỗi sự kiện | 1. Người dùng chọn tìm kiếm ATM từ tùy chỉnh 2. Hệ thống hiển thị giao diện với bản đồ và các ATM có trong khu vực quanh người dùng |
| Ngoại lệ |  |

* 1. **Thiết kế hệ thống**
     1. *Thiết kế các biểu đồ tuần tự*
* Chức năng đăng ký



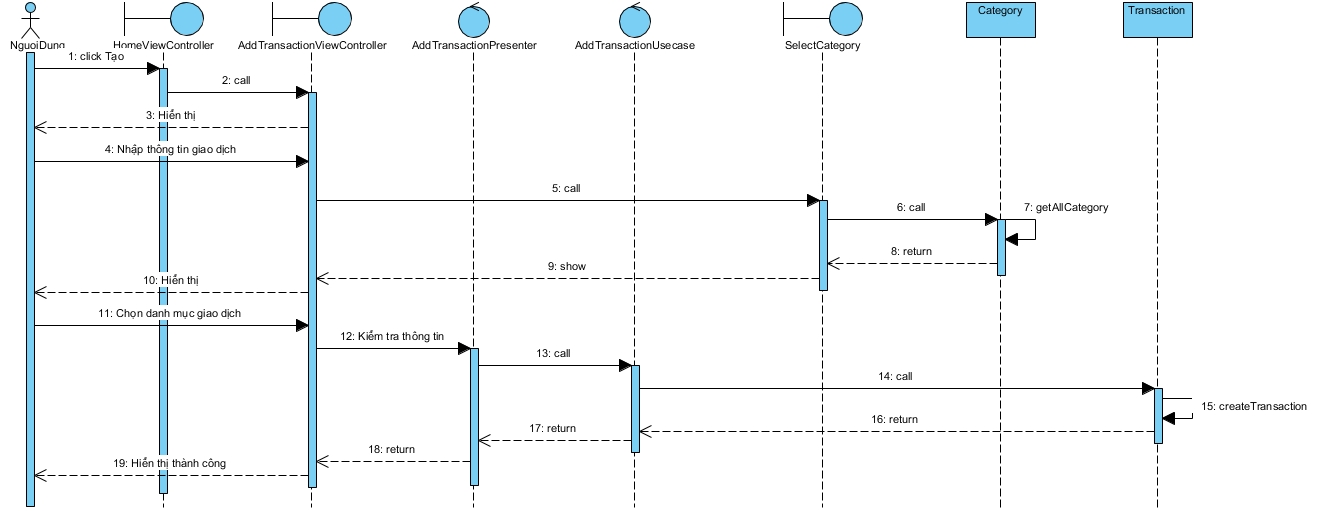
Hình 3. 7 Biểu đồ tuần tự chức năng đăng ký

* Chức năng đăng nhập



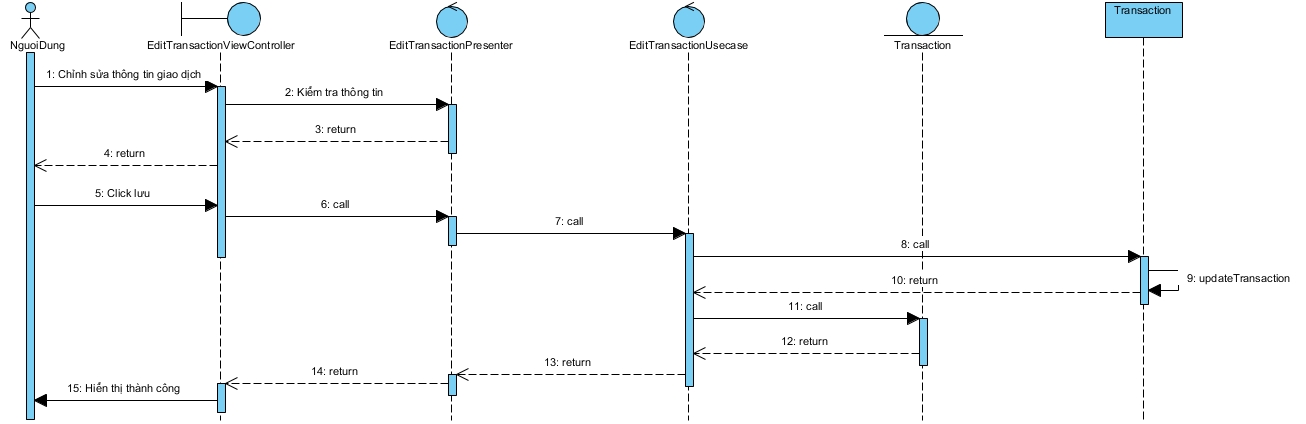
Hình 3. 8 Biểu đồ tuần tự chức năng đăng nhập

* Chức năng tạo giao dịch



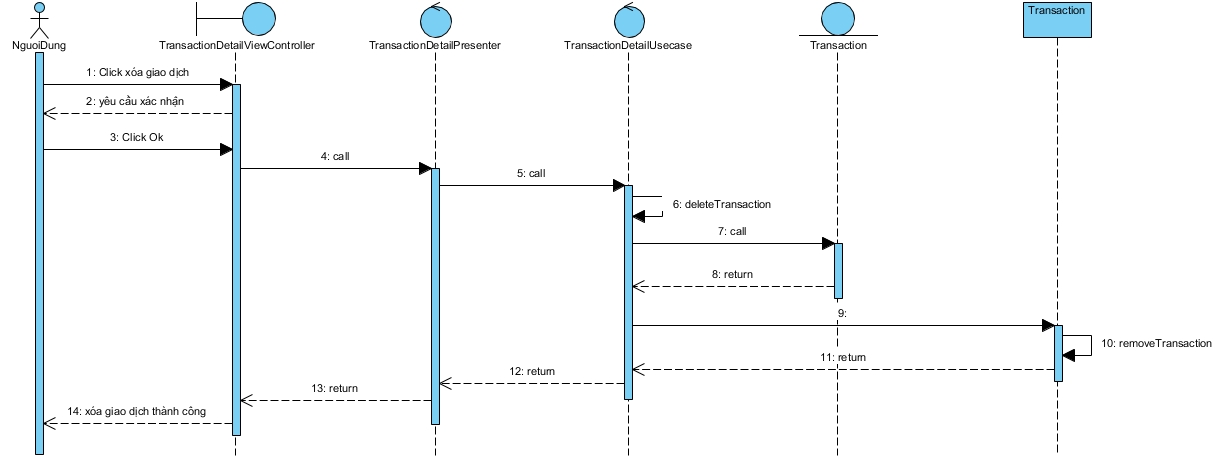
Hình 3. 9 Biểu đồ tuần tự chức năng tạo giao dịch

* Chức năng sửa thông tin giao dịch



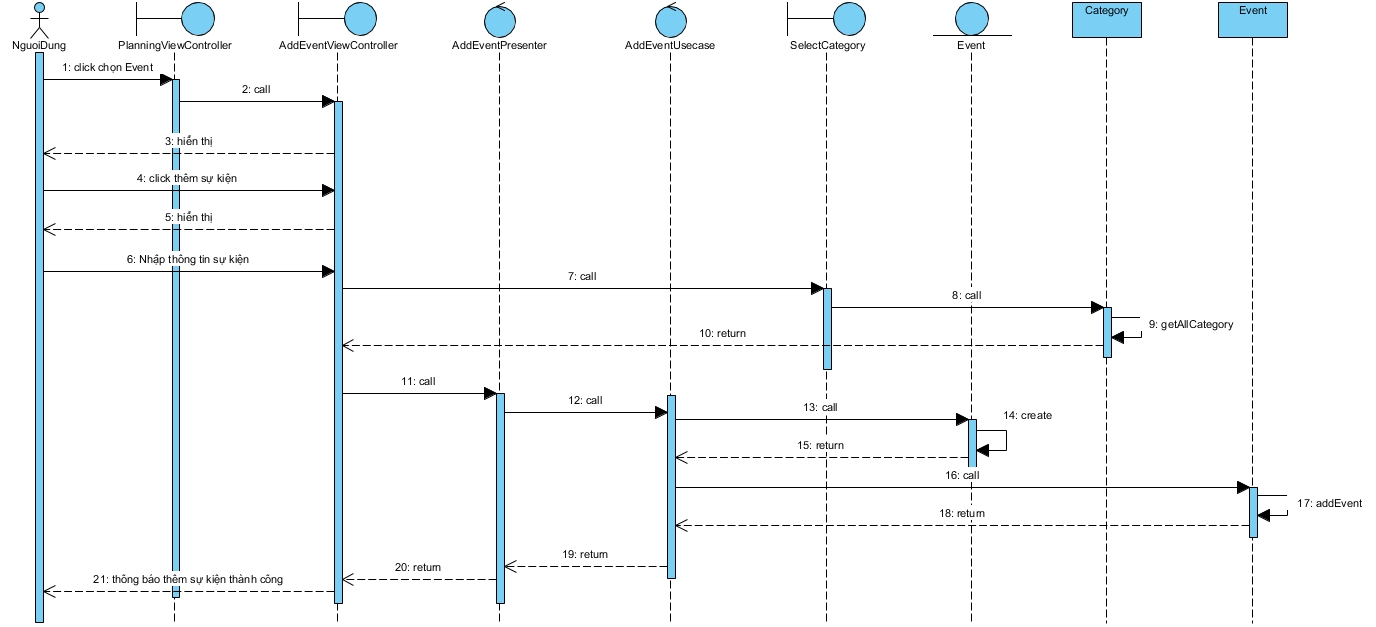
Hình 3. 10 Biểu đồ tuần tự chức năng sửa thông tin giao dịch

* Chức năng xóa giao dịch



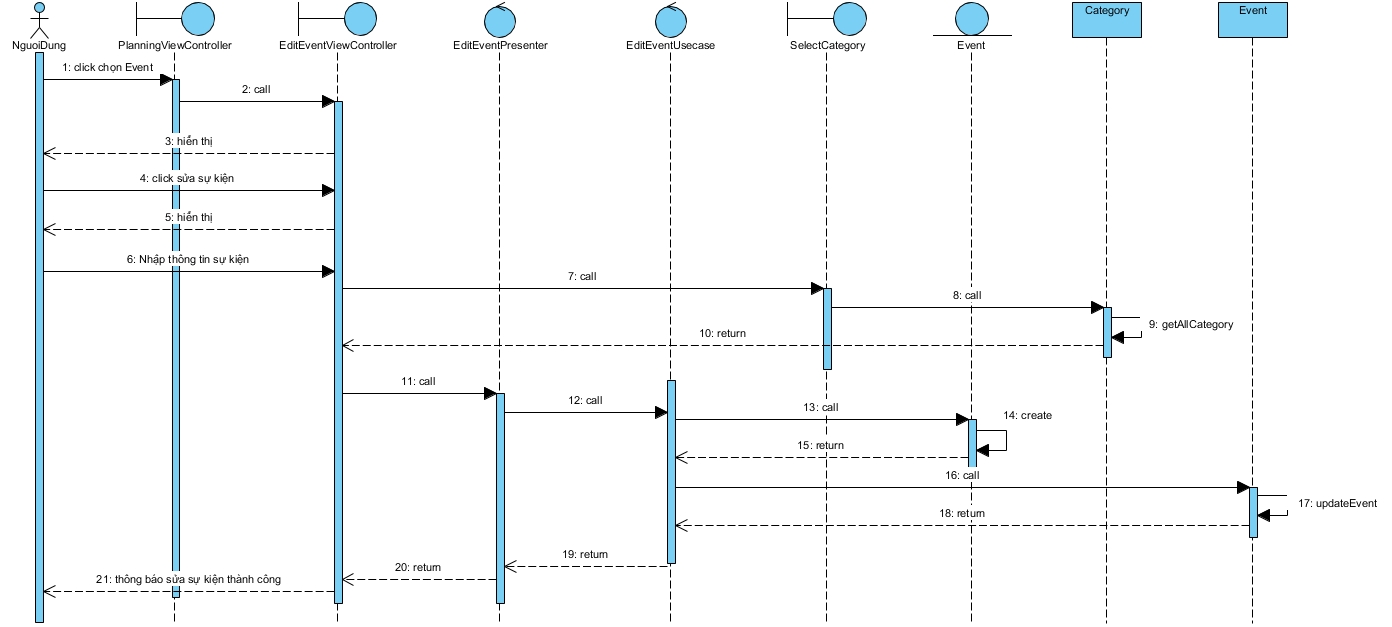
Hình 3. 11 Biểu đồ tuần tự chức năng xóa giao dịch

* Chức năng tạo sự kiện



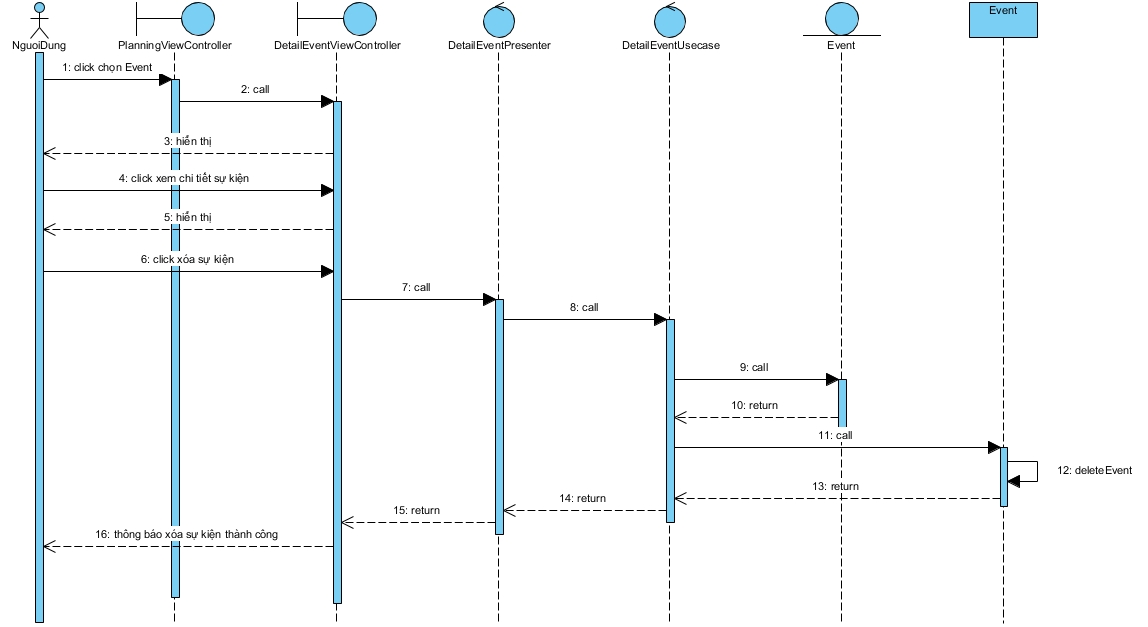
Hình 3. 12 Biểu đồ tuần tự chức năng tạo sự kiện

* Chức năng sửa thông tin sự kiện



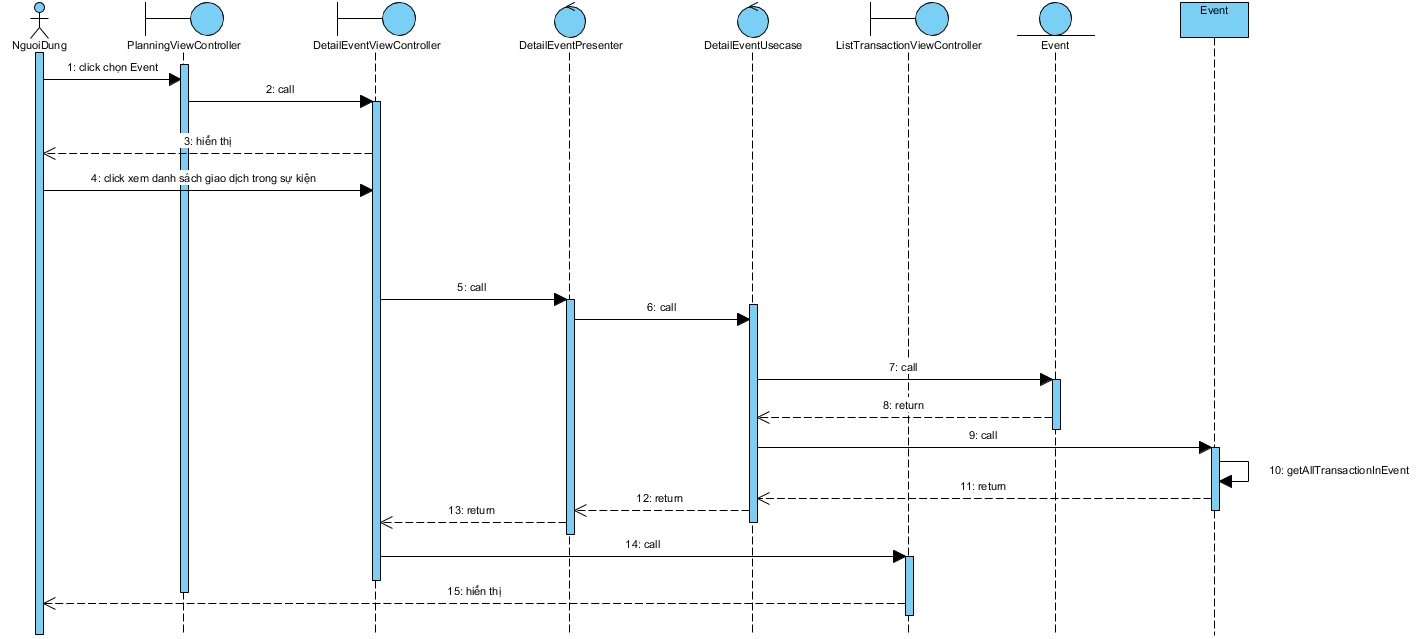
Hình 3. 13 Biểu đồ tuần tự chức năng sửa thông tin sự kiện

* Chức năng xóa sự kiện



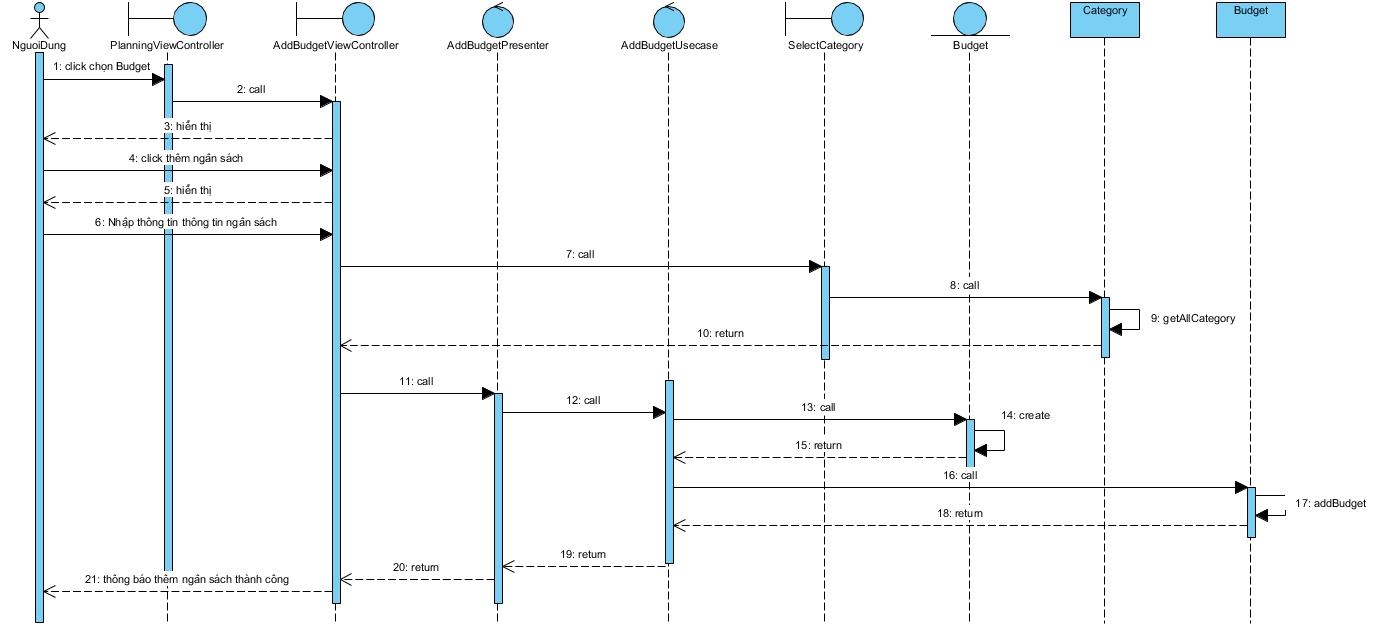
Hình 3. 14 Biểu đồ tuần tự chức năng xóa sự kiện

* Chức năng xem danh sách giao dịch trong sự kiện



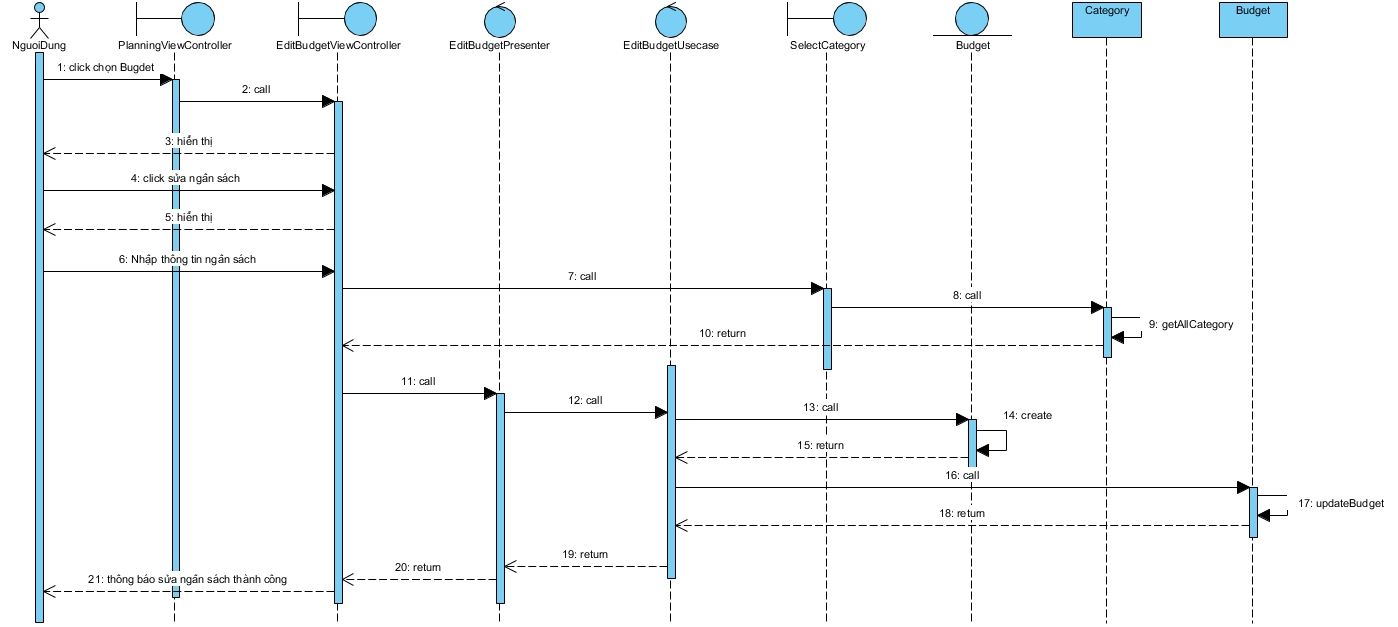
Hình 3. 15 Biểu đồ tuần tự chức năng xem danh sách giao dịch trong sự kiện

* Chức năng tạo ngân sách



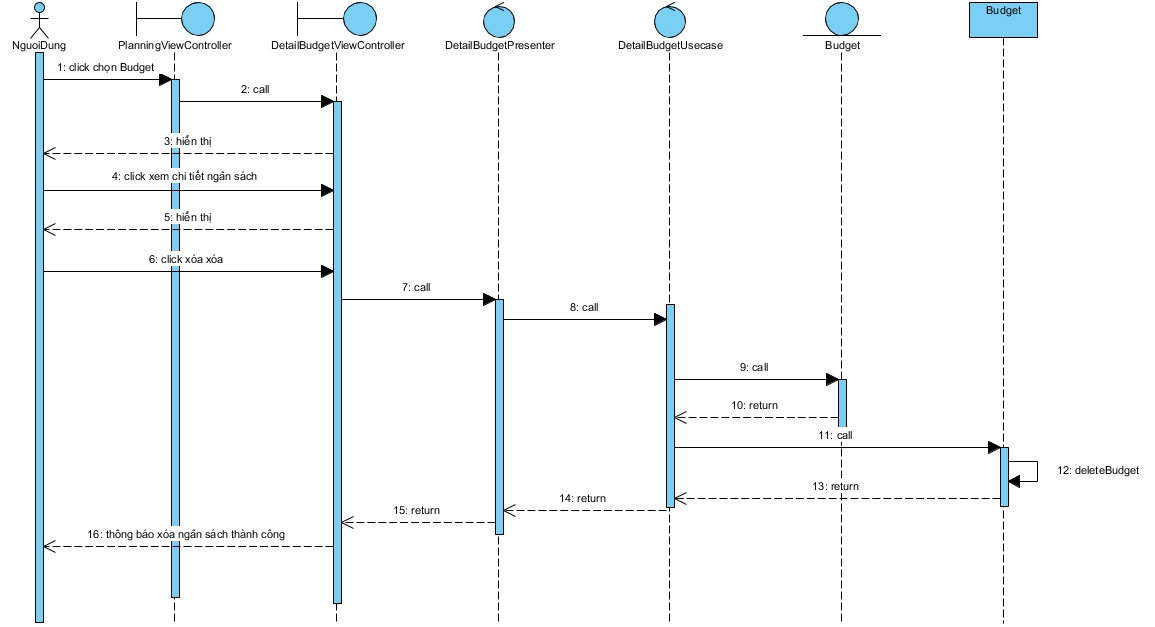
Hình 3. 16 Biểu đồ tuần tự chức năng tạo ngân sách

* Chức năng sửa thông tin ngân sách



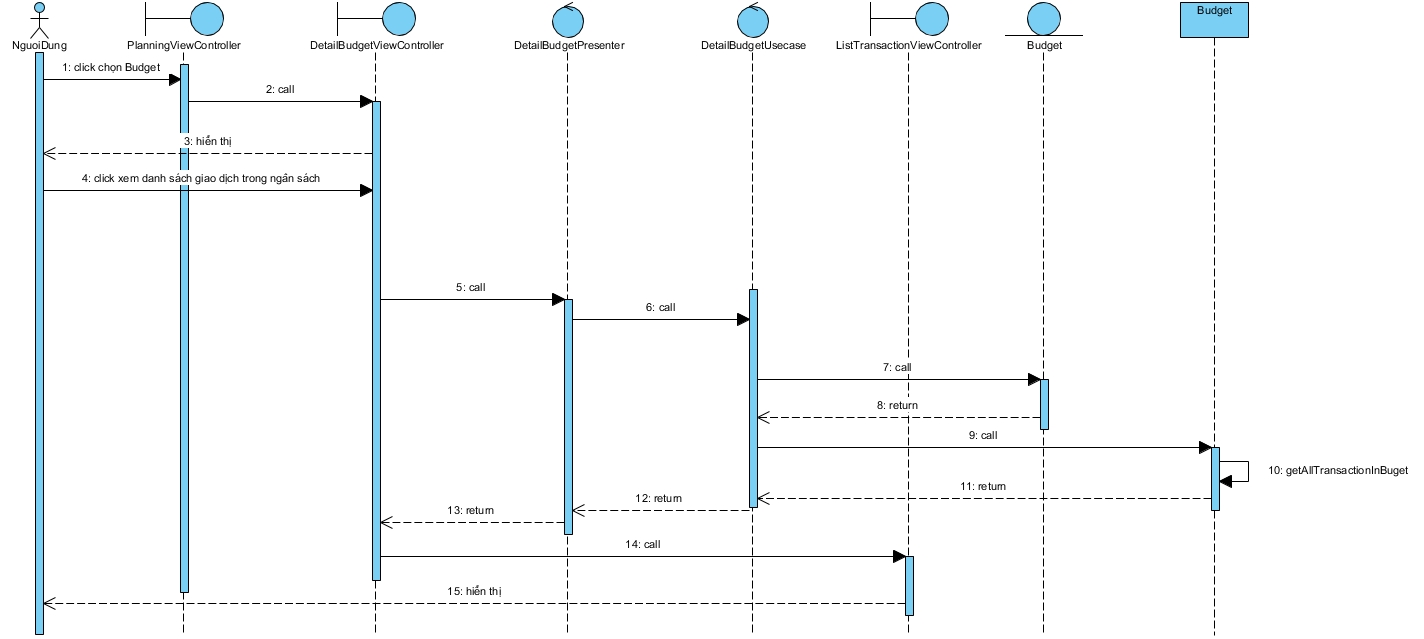
Hình 3. 17 Biểu đồ tuần tự chức năng sửa thông tin ngân sách

* Chức năng xóa ngân sách



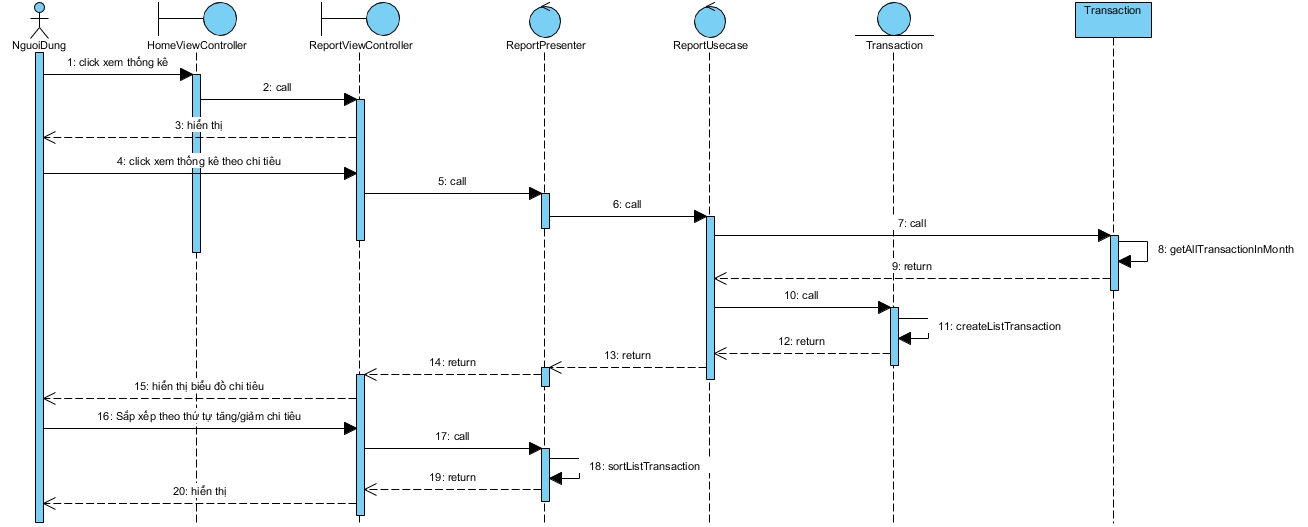
Hình 3. 18 Biểu đồ tuần tự chức năng xóa ngân sách

* Chức năng xem danh sách giao dịch trong ngân sách



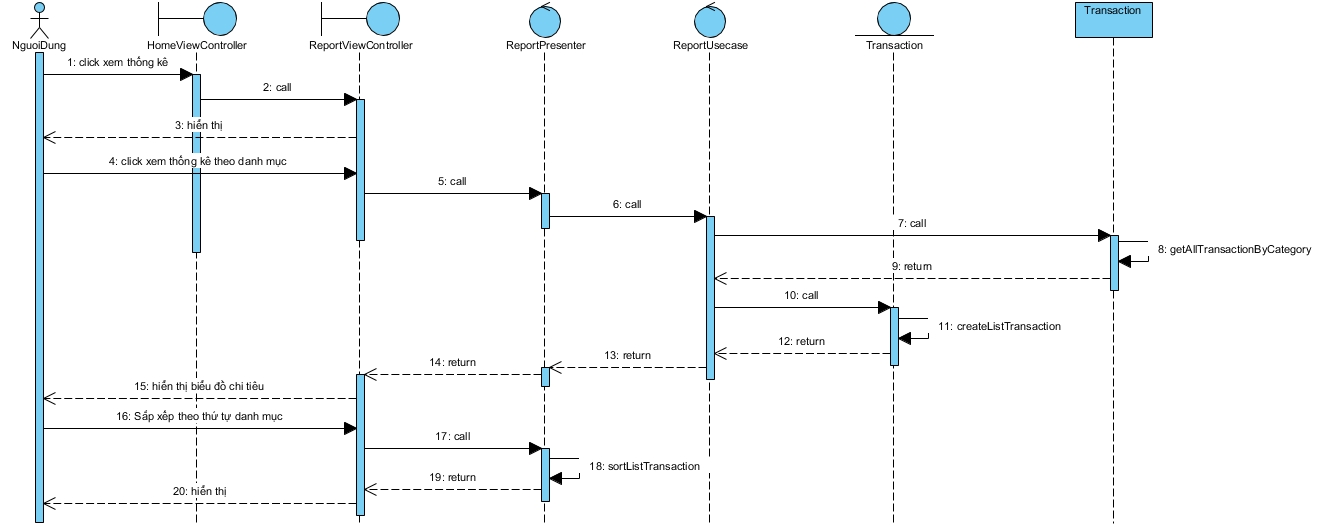
Hình 3. 19 Biểu đồ tuần tự chức năng xem danh sách giao dịch trong ngân sách

* Chức năng thống kê theo loại chi tiêu



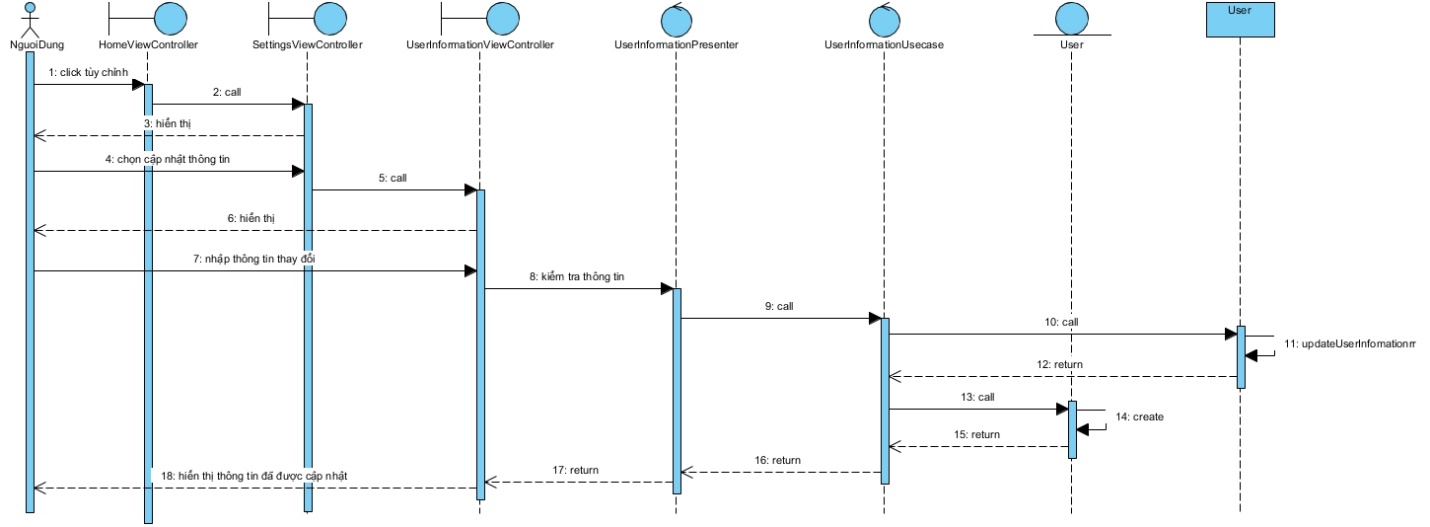
Hình 3. 20 Biểu đồ tuần tự chức năng thống kê theo loại chi tiêu

* Chức năng thống kế theo danh mục



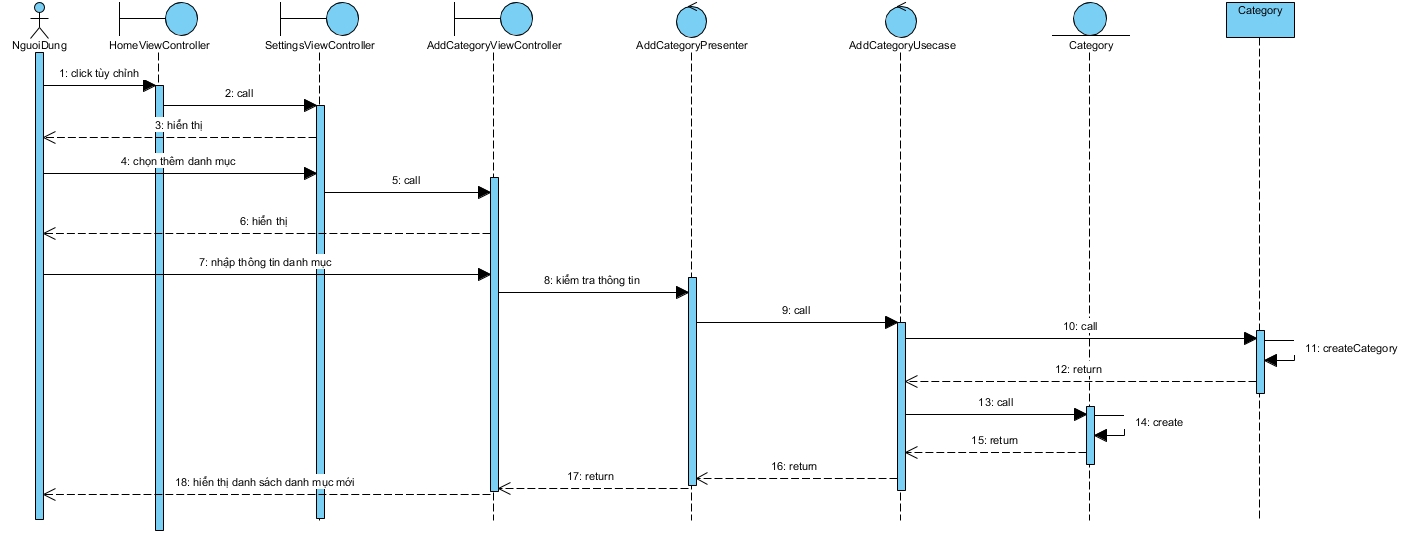
Hình 3. 21 Biểu đồ tuần tự chức năng thống kê theo danh mục

* Chức năng cập nhật thông tin người dùng



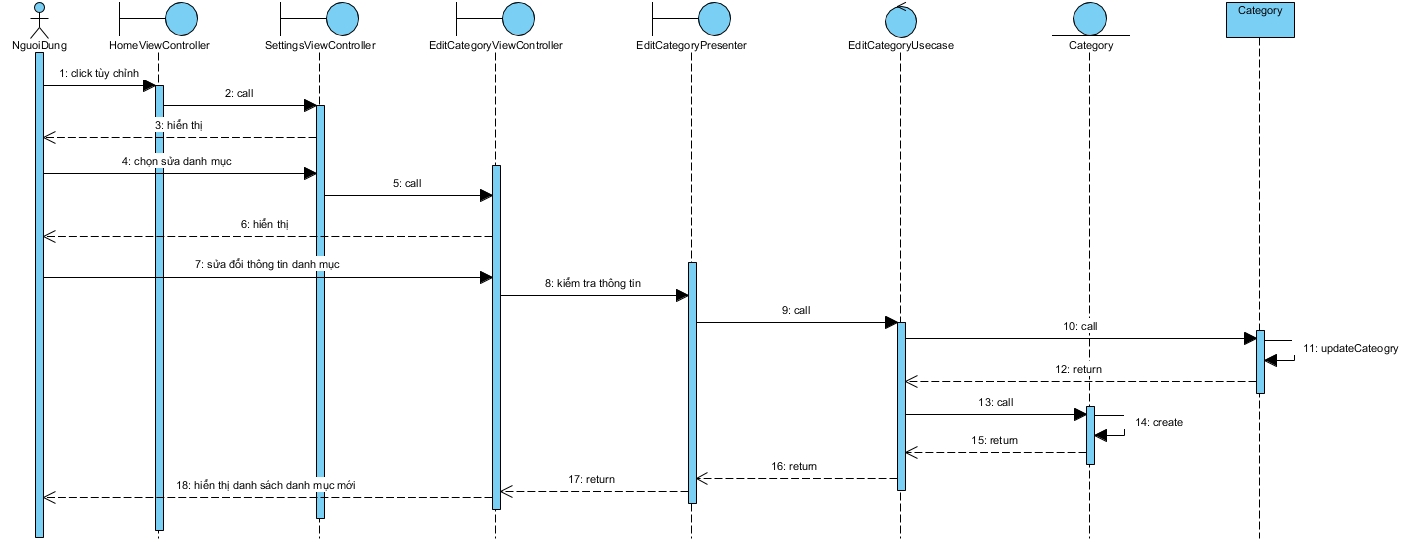
Hình 3. 22 Biểu đồ tuần tự chức năng cập nhật thông tin người dùng

* Chức năng thêm danh mục



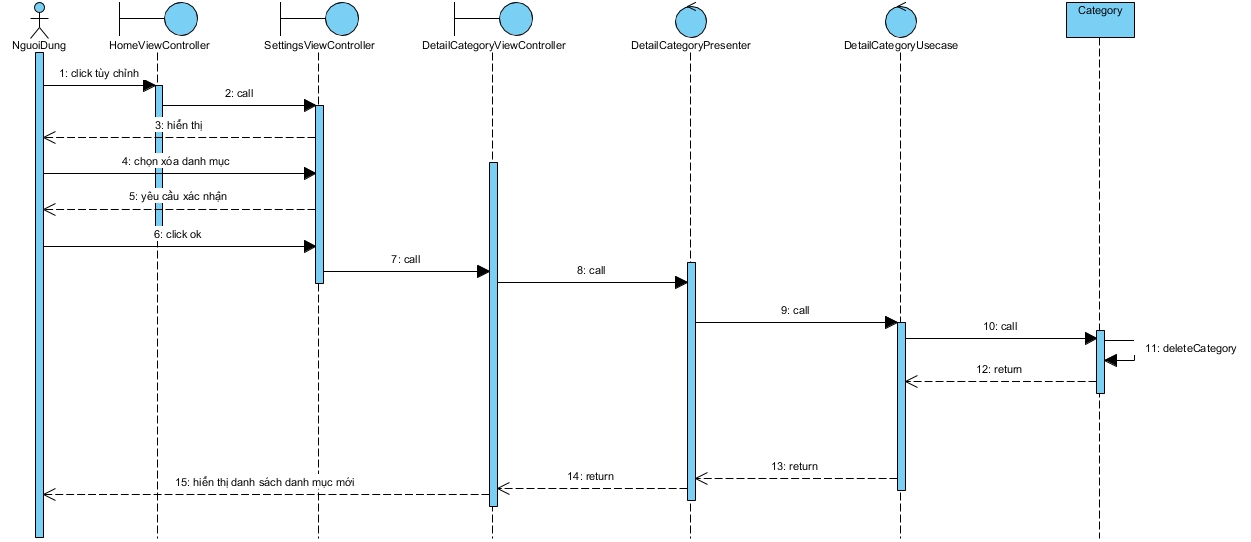
Hình 3. 23 Biểu đồ tuần tự chức năng thêm danh mục

* Chức năng sửa thông tin danh mục



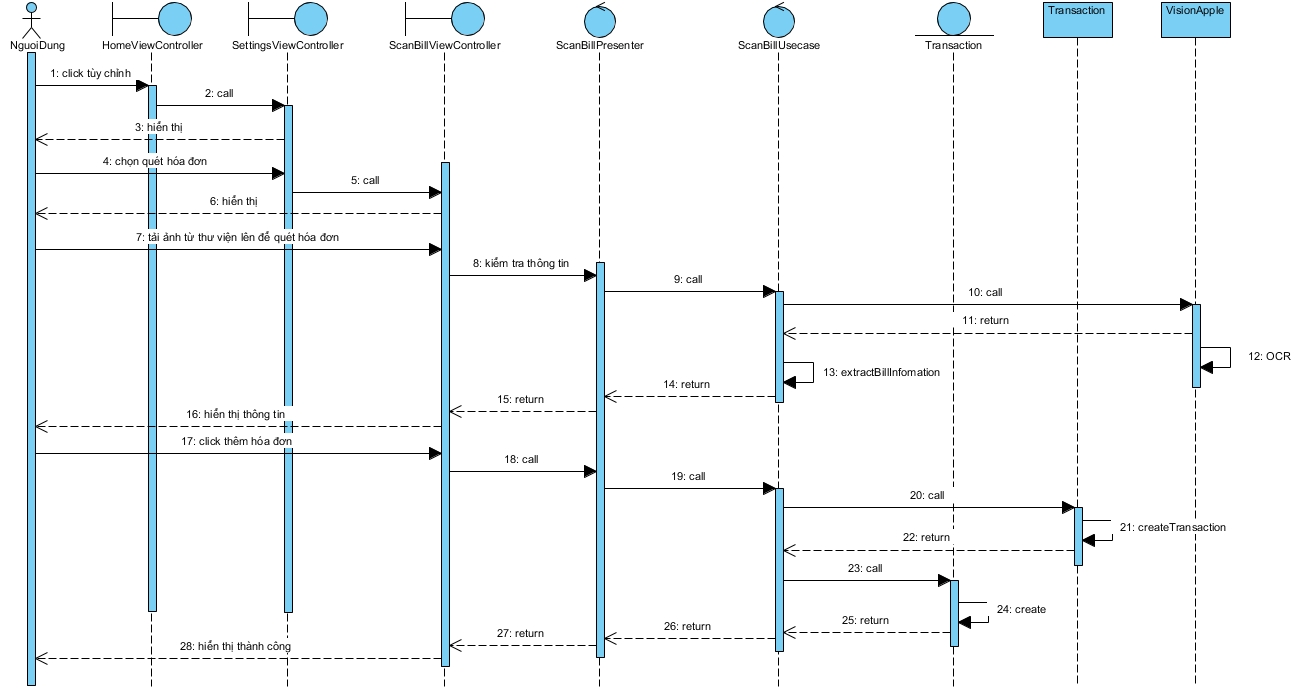
Hình 3. 24 Biểu đồ tuần tự chức năng sửa thông tin danh mục

* Chức năng xóa danh mục



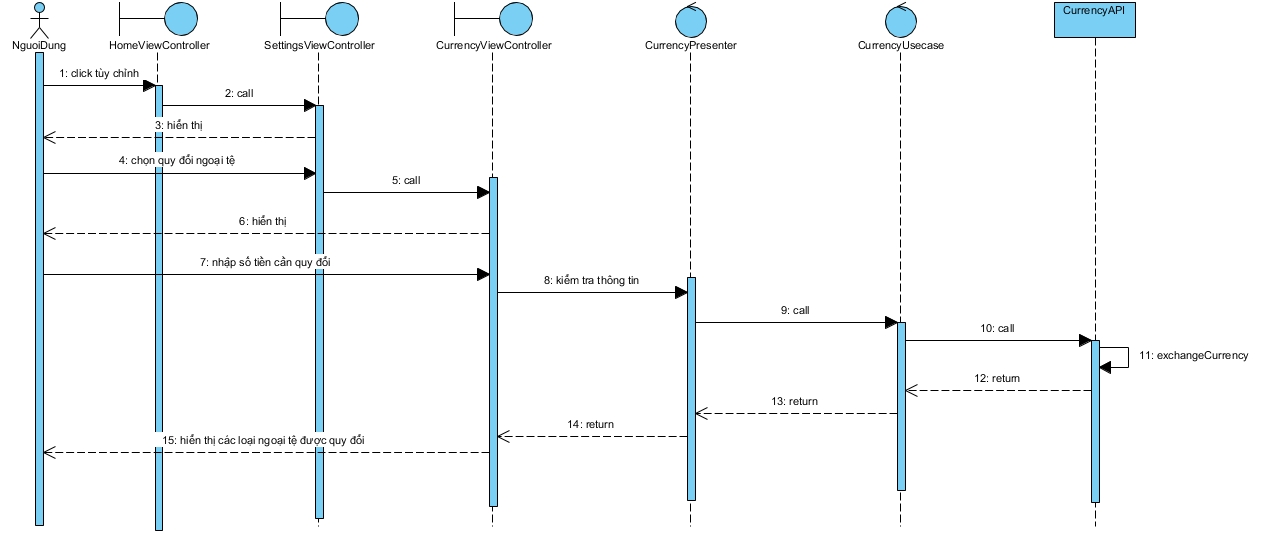
Hình 3. 25 Biểu đồ tuần tự chức năng xóa danh mục

* Chức năng quét hóa đơn



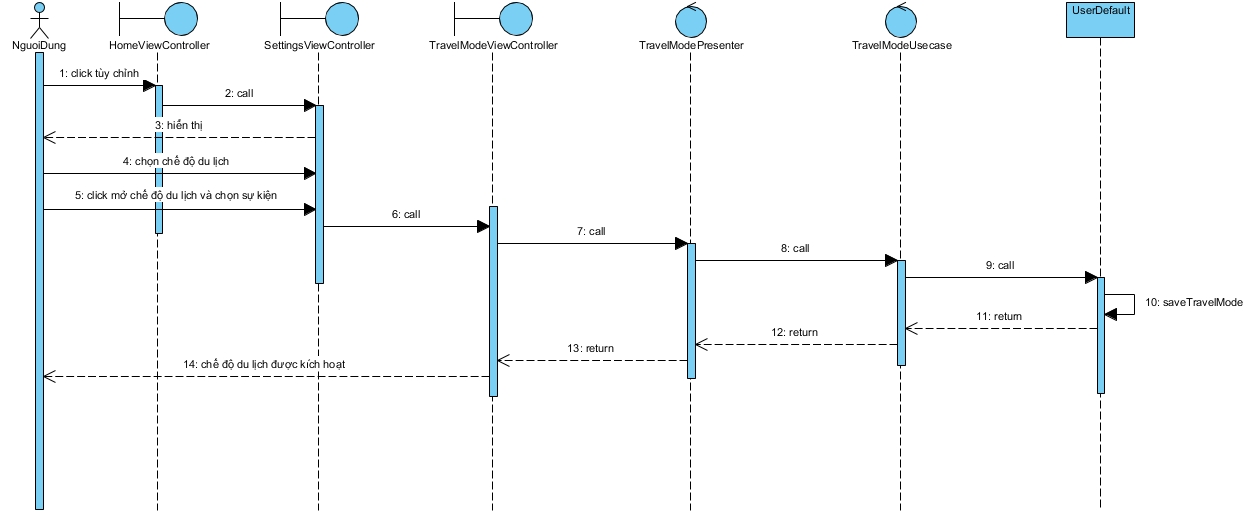
Hình 3. 26 Biểu đồ tuần tự chức năng quét hóa đơn

* Chức năng quy đổi ngoại tệ



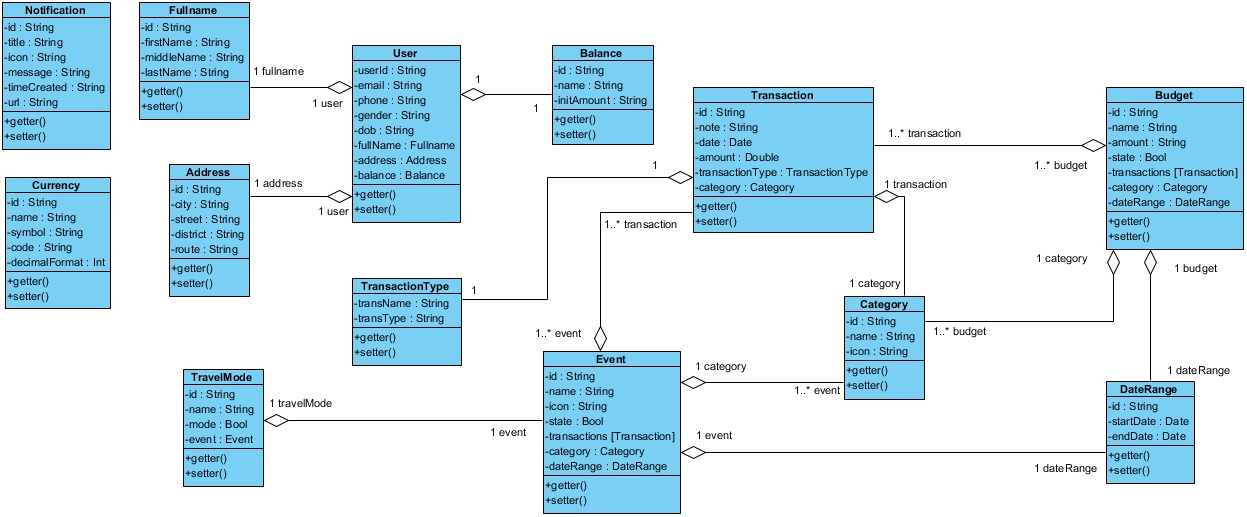
Hình 3. 27 Biểu đồ tuần tự chức năng quy đổi ngoại tệ

* Chức năng Chế độ du lịch



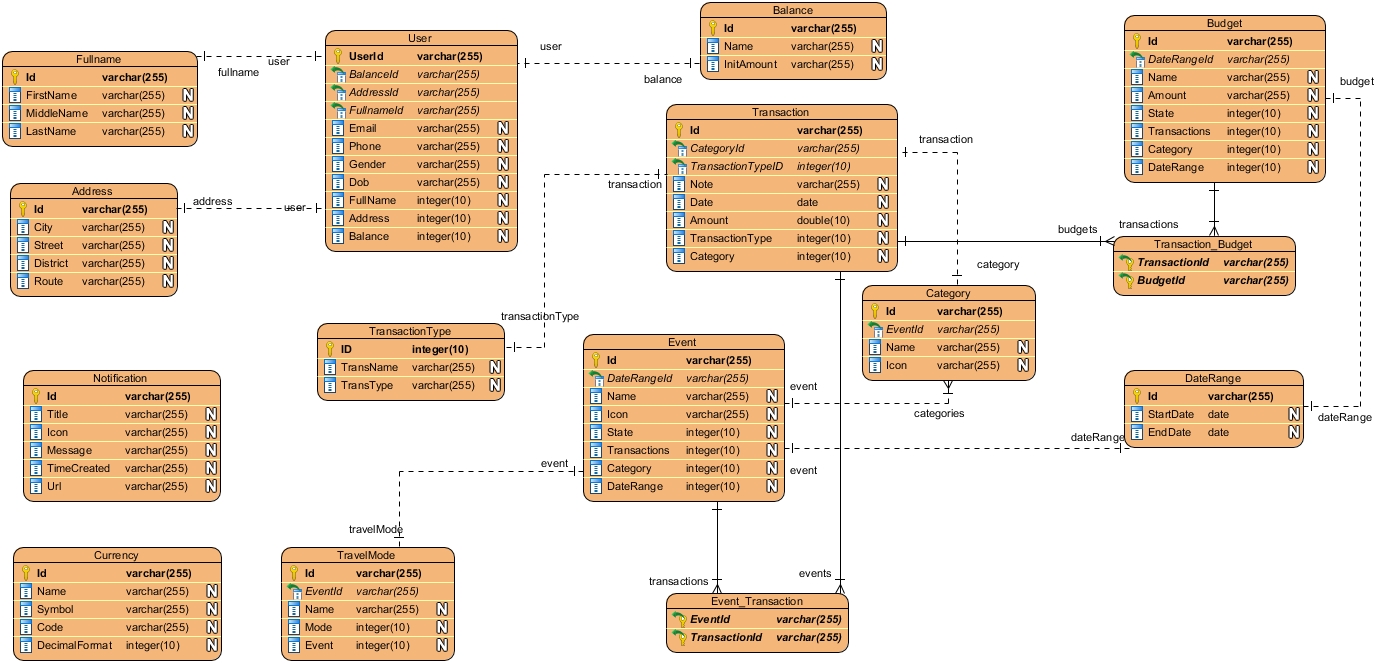
Hình 3. 28 Biểu đồ tuần tự chức năng chế độ du lịch

* + 1. *Sơ đồ lớp pha thiết kế*



Hình 3. 29 Sơ đồ lớp pha thiết kế

* + 1. *Thiết kế cơ sở dữ liệu*

**

Hình 3. 30 Sơ đồ thiết kế cơ sở dữ liệu

# **Chương 4: Xây dựng ứng dụng**

1. **Yêu cầu hệ thống**
   * 1. *Yêu cầu đối với việc phát triển*
        + MacOS Big Sur 11.2
        + Xcode 12: môi trường phát triển tích hợp (IDE) chính thức cho hệ điều hành iOS
        + Firebase
        + Apple Framework và các API của nhà phát triển Apple
     2. *Yêu cầu đối với thiết bị*
        + Máy sử dụng hệ điều hành iOS phiên bản 13 trở lên
        + Máy có cài sẵn App Store
        + Yêu cầu có mạng khi đăng nhập và sử dụng
        + Đáp ứng các yêu cầu của hệ thống:

* Quyền truy cập Internet
* Quyển truy cập thông tin trạng thái mạng
* Quyền truy cập Camera
* Quyền truy cập và đọc ghi bộ nhớ tạm thời

* 1. **Một số hình ảnh của ứng dụng**

3. 2. 1. *Giao diện chức năng Đăng nhập*

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 4. 1 Giao diện chức năng đăng nhập

* + 1. *Màn hình chính và thanh chức năng chính*

|  |  |
| --- | --- |
| C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.42.31.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.45.19.png |

Hình 4. 2 Màn hình chính và thanh chức năng chính

* + 1. *Giao diện Quản lý giao dịch*

|  |  |
| --- | --- |
| C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.42.36.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.45.13.png |

Hình 4. 3 Giao diện quản lý giao dịch

* + 1. *Giao diện Quản lý sự kiện*

|  |  |  |
| --- | --- | --- |
| C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.45.02.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.45.07.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-06 at 22.18.18.png |

Hình 4. 4 Giao diện quản lý sự kiện

* + 1. *Giao diện Quản lý ngân sách*

|  |  |  |
| --- | --- | --- |
| C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-06 at 22.17.03.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-06 at 22.18.48.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-06 at 22.17.06.png |

Hình 4. 5 Giao diện quản lý ngân sách

* + 1. *Giao diện Chức năng thống kê*

|  |  |  |
| --- | --- | --- |
| C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.42.26.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-06 at 22.20.07.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-06 at 22.19.55.png |

Hình 4. 6 Giao diện chức năng thống kê

* + 1. *Giao diện một số chức năng của Tùy chỉnh*

|  |  |  |
| --- | --- | --- |
| C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.44.44.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.44.52.png | C:\Users\vvbsp\OneDrive\Máy tính\College_Stuff\DoAnDaiHoc\giao dien\Simulator Screen Shot - iPhone 8 - 2021-01-04 at 22.44.38.png |

Hình 4. 7 Giao diện một số chức năng của Tùy chỉnh

# **Kết luận**

1. **Đánh giá đề tài**

Sau quá trình nghiên cứu và xây dựng ứng dụng, thực hiện đồ án, đã đạt được một số kết quả như sau:

* Về mặt lý thuyết đạt được:
* Nền tảng iOS
* Cách xây dựng một ứng dụng trên thiết bị di dộng sử dụng nền tảng iOS
* Nắm được nền tảng của ngôn ngữ lập trình Swift trong việc phát triển các ứng dụng trên nền tảng iOS
* Tìm hiểu và tích hợp được các dịch vụ đến từ Firebase, Goolge, Facebook, Apple
* Sử dụng tốt được các nền tảng lưu trữ như Firebase Realtime Database
* Về mặt thực tiễn đạt được:
* Tài khoản: Cho phép người dùng đăng nhập vào hệ thống.
* Quản lý số dư: Cho phép người dùng điều chỉnh số dư với hệ thống.
* Quản lý giao dịch: Cho phép người dùng thực hiện các thao tác như thêm, sửa, xóa giao dịch, quản lý được chi tiêu.
* Quản lý sự kiện: Cho phép người dùng thực hiện các thao tác như thêm, sửa xóa sự kiện, phục vụ việc quản lý chi tiêu thông minh theo từng sự kiện.
* Quản lý ngân sách: Cho phép người dùng thực hiện các thao tác như thêm, sửa xóa ngân sách, phục vụ việc quản lý chi tiêu thông minh với việc quản lý theo các ngân sách, cảnh báo và nhắc nhở người dùng.
* Thống kê: Cho phép người dùng theo dõi được tình hình giao dịch theo tháng, không chỉ theo loại giao dịch mà còn theo các danh mục, giúp người dùng có cái nhìn dễ dàng và khoa học hơn.
* Các chức năng hỗ trợ: Cho phép người dùng sử dụng thêm các chức năng hỗ trợ thông minh khác như “Quy đổi ngoại tệ”, “Quét hóa đơn”, “Chế độ du lịch”, giúp người dùng có những trải nghiệm tốt nhất khi sử dụng ứng dụng

1. **Định hướng phát triển**

* Một số định hướng phát triển trong tương lai mà ứng dụng mong muốn đạt được:
* Đồng bộ được dữ liệu người dùng khi đang ở trạng thái ngoại tuyến.
* Có thể liên kết được với các ví thanh toán điện tử giúp người dùng dễ dàng hơn trong việc chi tiêu.
* Phát triển chức năng quét hóa đơn không chỉ sử dụng ảnh, có thể sử dụng mã QR hoặc có thể định dạng được loại hóa đơn thêm mới từ người dùng.
* Phân tích dữ liệu người dùng và đưa ra các giải pháp, gợi ý hợp lý.

**Tài liệu tham khảo**

**Giáo trình**

[1] Kiến trúc và thiết kế phần mềm – PGS.TS Trần Đình Quế

[2] Phân tích và thiết kế hệ thống thông tin – PGS.TS Trần Đình Quế

**Website**

[1] Trang web dành cho các nhà phát triển của Apple

[2] Trang web chính thức của Firebase

[3] Functional Reactive Programming - Viblo

[4] Vision Framework của Apple được giới thiệu tại WWDC 2017

[5] Các quy tắc lập trình S.O.L.I.D - Viblo