**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**----o0o----**



**BÁO CÁO GIỮA KỲ**

**Thực hành lập trình Web**

**Đề tài : Quản lý chi tiêu cá nhân**

**(Smart money)**

*Giáo viên hướng dẫn :* Thầy Nguyễn Mạnh Tuấn

*Nhóm sinh viên thực hiện- Nhóm 5 :*

Bùi Thu Thủy Linh 20184131

Nguyễn Hoàng Sơn 20184187

Nguyễn Hữu Tuấn 20184221

Phạm Thị Mai Tuyết 20184227

Contents

[**Lời nói đầu**](#_heading=h.k0p3vzyyvk10) **4**

[**Phân công thành viên trong nhóm**](#_heading=h.30j0zll) **4**

[**Chương 1. Khảo sát bài toán**](#_heading=h.1fob9te) **5**

[1.1 Mô tả yêu cầu bài toán](#_heading=h.3znysh7) 5

[1.2 Khảo sát bài toán](#_heading=h.2et92p0) 5

[**Chương 2. Đặc tả yêu cầu bài toán**](#_heading=h.tyjcwt) **6**

[2.1 Giới thiệu chung](#_heading=h.3dy6vkm) 6

[2.1.1 Các tác nhân trong hệ thống](#_heading=h.1t3h5sf) 6

[2.1.2 Mối quan hệ giữa các tác nhân](#_heading=h.4d34og8) 6

[2.1.3 Xác định các ca sử dụng](#_heading=h.2s8eyo1) 6

[2.1.4 Xác định quan hệ giữa tác nhân và use case](#_heading=h.17dp8vu) 6

[2.2 Biểu đồ Use case](#_heading=h.3rdcrjn) 7

[2.2.1 Biểu đồ Use case tổng quan](#_heading=h.26in1rg) 7

[2.2.2 Biểu đồ Use case phân rã](#_heading=h.lnxbz9) 8

[Phân rã Use case “CRUD ví”](#_heading=h.zdx6yfwp8bee) 8

[Phân rã Use case “CRUD giao dịch”](#_heading=h.blbfpbqry8i6) 8

[Phân rã Use case “CRUD ngân sách”](#_heading=h.xdystb8xj51e) 9

[Phân rã Use case “Thống kê chi tiêu”](#_heading=h.bjx8j7yiqmmp) 9

[Phân rã Use case “Quản lý người dùng”](#_heading=h.1vcs9yrss9f9) 10

[Phân rã Use case “Quản lý danh mục cá nhân”](#_heading=h.w25i69bi2lp8) 10

[Phân rã Use case “CRUD danh mục ban đầu”](#_heading=h.xqdye7e1yyka) 11

[2.3 Đặc tả các chức năng](#_heading=h.35nkun2) 11

[Usecase “Đăng nhập, đăng ký ”](#_heading=h.bbqnrol482ts) 11

[Usecase “Quản lý ví”](#_heading=h.vbtkqphu04mf) 12

[Usecase “Quản lý giao dịch”](#_heading=h.gbks1ffe2dj0) 12

[Usecase “Quản lý ngân sách”](#_heading=h.rjpqif62ba6o) 13

[Usecase “Xem thống kê chi tiêu”](#_heading=h.ppyydfg5ynxx) 13

[Usecase “Quản lý người dùng”](#_heading=h.c6z2kwjwt4s4) 13

[2.4 Yêu cầu khác](#_heading=h.1ksv4uv) 14

[Chức năng](#_heading=h.gxudun8b04im) 14

[Tính dễ sử dụng](#_heading=h.juj881yn305k) 14

[Các yêu cầu khác](#_heading=h.4e6apyey58i0) 14

[**Chương 3. Phân tích yêu cầu**](#_heading=h.44sinio) **15**

[3.1 Xây dựng biểu đồ Activity Diagram](#_heading=h.2jxsxqh) 15

[Quy trình tạo giao dịch mới](#_heading=h.7i2ja0ipkb9q) 15

[Quy trình chỉnh sửa một giao dịch](#_heading=h.blzo0kpr6bu6) 16

[Quy trình xóa một giao dịch](#_heading=h.28ewbm1jw2mc) 18

[Quy trình tạo một ngân sách](#_heading=h.l5xvaeagni4i) 20

[Quy trình xóa ngân sách](#_heading=h.31m64jkwp9lx) 21

[Quy trình tạo ví](#_heading=h.bq2yyhc9r2u) 23

[Quy trình người dùng thay đổi thông tin](#_heading=h.hc6pmssvwiwe) 25

[Quy trình xem thống kê](#_heading=h.f0kupw447j2g) 27

[Quy trình quản lý người dùng (Admin)](#_heading=h.taqs0wwe7ari) 28

[Authorize user account](#_heading=h.6orxmdj9ifsh) 28

[Browse user account](#_heading=h.hs7nm1ui0gnt) 29

[Search total expense](#_heading=h.dwh03cmj9pba) 30

[3.2 Xây dựng biểu đồ Sequence Diagram](#_heading=h.z337ya) 30

[Sequence diagram User – Create budget](#_heading=h.3q5sasy) 30

[Sequence diagram User – Create transaction](#_heading=h.25b2l0r) 31

[Sequence diagram User – Update transaction](#_heading=h.kgcv8k) 32

[Sequence diagram User – Delete transaction](#_heading=h.34g0dwd) 34

[Sequence diagram User – Create Wallet](#_heading=h.h8qssdjrtzym) 34

[Sequence diagram User – Update Information](#_heading=h.43ky6rz) 36

[Sequence diagram User – Browse expense](#_heading=h.2iq8gzs) 37

[Sequence diagram Admin](#_heading=h.xvir7l) 38

[Sequence diagram Authorize user account](#_heading=h.3hv69ve) 38

[Sequence diagram Browse user account](#_heading=h.1x0gk37) 38

[Sequence diagram Search total expense](#_heading=h.4h042r0) 39

[3.3 Xây dựng biểu đồ lớp phân tích](#_heading=h.3j2qqm3) 40

[3.4 Xây dựng biểu đồ thực thể liên kết (ERD)](#_heading=h.1y810tw) 40

[**Chương 4. Thiết kế chương trình**](#_heading=h.4i7ojhp) **41**

[4.1 Thiết kế kiến trúc](#_heading=h.2xcytpi) 41

[4.2 Thiết kế cơ sở dữ liệu](#_heading=h.1ci93xb) 41

[4.3 Thiết kế chi tiết các gói](#_heading=h.3whwml4) 42

[4.4 Thiết kế chi tiết các lớp](#_heading=h.2bn6wsx) 43

[4.5 Sơ đồ lớp chi tiết](#_heading=h.qsh70q) 43

[4.6 Thiết kế giao diện](#_heading=h.3as4poj) 43

[**Chương 5. Xây dựng chương trình minh họa**](#_heading=h.1pxezwc) **46**

[5.1 Thư viện và công cụ sử dụng](#_heading=h.49x2ik5) 46

[5.2 Kết quả chương trình minh họa](#_heading=h.2p2csry) 46

[5.3 Giao diện minh họa các chức năng của chương trình](#_heading=h.147n2zr) 46

[**Chương 6. Kiểm thử chương trình**](#_heading=h.3o7alnk) **50**

[6.1 Kiểm thử các chức năng User](#_heading=h.23ckvvd) 50

[6.1.1 Kiểm thử cho chức năng đăng ký](#_heading=h.ihv636) 50

[6.1.2 Kiểm thử cho chức năng đăng nhập](#_heading=h.5hzfn0i886m2) 51

[6.1.3 Kiểm thử cho chức năng CRUD ví](#_heading=h.32hioqz) 51

[6.1.3 Kiểm thử cho CRUD ngân sách](#_heading=h.1hmsyys) 53

[6.1.4 Kiểm thử cho chức năng CRUD giao dịch](#_heading=h.41mghml) 54

[6.1.5 Kiểm thử cho chức năng CRUD danh mục](#_heading=h.huajayip2e2u) 56

[6.1.6 Kiểm thử cho chức năng thống kê](#_heading=h.4bdha12ojtlh) 57

[6.1.7 Kiểm thử cho chức năng update thông tin](#_heading=h.fmh0bo9fd2z6) 58

[6.2 Kiểm thử các chức năng Admin](#_heading=h.yezkvziw133x) 59

[6.2.1 Kiểm thử cho chức năng quản lý tài khoản user](#_heading=h.mw1c17nkftqw) 59

[6.2.2 Kiểm thử cho chức năng CRUD danh mục của admin](#_heading=h.cncni6l330pw) 60

[**Chương 7. Hướng dẫn cài đặt và sử dụng**](#_heading=h.hni8tcaydfus) **62**

[7.1 Hướng dẫn cài đặt](#_heading=h.3fwokq0) 62

[7.2 Đối tượng, phạm vi sử dụng](#_heading=h.1v1yuxt) 62

[7.3 Xác định các yêu cầu cài đặt](#_heading=h.4f1mdlm) 62

[7.4 Hướng dẫn chi tiết các bước cài đặt](#_heading=h.2u6wntf) 62

[7.5 Hướng dẫn sử dụng phần mềm](#_heading=h.19c6y18) 63

[**Chương 8 : Nguyên lý thiết kế SOLID**](#_heading=h.3tbugp1) **64**

[**Kết luận và hướng phát triển**](#_heading=h.fh2tp74hvkyb) **66**

[Kết luận](#_heading=h.3fg66p8xkrp5) 66

[Hướng phát triển trong tương lai](#_heading=h.vjmh0mh5hvzx) 66

[**Từ điển thuật ngữ**](#_heading=h.u14xz2x0d9hv) **66**

[**Tài liệu tham khảo**](#_heading=h.28h4qwu) **67**

# 

# 

# 

# 

# 

# 

# 

# 

# Lời nói đầu

Vấn đề tiền bạc hiện nay không chỉ là chuyện về những con số mà đang dần hình thành như một lối sống. Việc xây dựng một website giúp quản lý tài chính cá nhân sẽ có ích cho tất cả những bạn nào đang bắt đầu tập làm người lớn, hoặc cả những người đã lớn rồi mà vẫn hoang mang về chuyện tiền nong. Giúp mọi người dễ dàng quản lý tài chính của mình và đưa ra các quyết định về vấn đề tài chính một cách thông minh.

# Phân công thành viên trong nhóm

| **MSSV** | **Tên thành viên** | **Nhiệm vụ** |
| --- | --- | --- |
| 20184131 | Bùi Thu Thủy Linh | Xác định yêu cầu bài toán.  Xây dựng, vẽ các Sequence Diagram và Activity Diagram liên quan đến Budget và Transaction.  Thiết kế Database, xây dựng code Database .  Code backend+Frontend các chức năng quản lý người dùng, quản lý danh mục của Admin.  Hoàn thành báo cáo. |
| 20184187 | Nguyễn Hoàng Sơn | Xác định yêu cầu bài toán.  Xây dựng, vẽ các Sequence Diagram và Activity Diagram liên quan đến Admin và tổng hợp, vẽ Class Diagram.  Xây dựng sơ đồ E-R cho database.  Code backend cho các chức năng của Category, chỉnh sửa thông tin tài khoản User.  Hoàn thành báo cáo. |
| 20184221 | Nguyễn Hữu Tuấn | Xác định yêu cầu bài toán.  Xây dựng sơ đồ Logic và bảng vật lý cho database  Code backend + frontend cho các chức năng của Transaction, Budget, Wallet.  Quản lý tiến độ công việc chung của nhóm.  Hoàn thành báo cáo. |
| 20184227 | Phạm Thị Mai Tuyết | Xác định yêu cầu bài toán.  Xây dựng, vẽ các Sequence Diagram và Activity Diagram liên quan đến Wallet và chỉnh sửa tài khoản User.  Xây dựng sơ đồ Logic và bảng vật lý cho database.  Code Frontend xây dựng giao diện các trang Transaction, Budget, Wallet, Category, tổng quan.  Hoàn thành báo cáo. |

# Chương 1. Khảo sát bài toán

## 1.1 Mô tả yêu cầu bài toán

Quản lý tài chính cá nhân là một vấn đề khó, mỗi người có một cách tiếp cận và giải quyết riêng. Một số người xây dựng ngân sách, những người khác hạn chế dòng tiền tiêu, những người khác cố gắng tăng dòng tiền thu nhập hiện tại,... Có rất nhiều cách để tiết kiệm và kiếm tiền. Smart Money hoạt động dựa trên nguyên tắc rằng nếu bạn biết tiền của mình đang đi đến đâu, bạn có thể thiết lập một cách chi tiêu hợp lý để có thể tiết kiệm dựa trên số tiền kiếm được bằng cách theo dõi các khoản chi tiêu và thu nhập của mình.

## 1.2 Khảo sát bài toán

Hiện nay, số lượng người đòi hỏi có một hệ thống hỗ trợ quản lý chi tiêu cá nhân đang ngày một tăng từ đó việc phát triển website là một giải pháp hữu ích giúp đáp ứng những yêu cầu trên. Ngoài việc thu thập, thống kê những khoản chi tiêu, người dùng còn mong muốn có thể thiết lập một định mức để xây dựng một kế hoạch chi tiêu hợp lý. Do đó, website xây dựng tính năng ngân sách giúp người dùng đáp ứng yêu cầu trên.

# Chương 2. Đặc tả yêu cầu bài toán

## 2.1 Giới thiệu chung

### 2.1.1 Các tác nhân trong hệ thống

Hệ thống bao gồm 3 tác nhân là khách , người dùng (User) và người quản quản lý (Admin). Khách là người chưa đăng nhập vào hệ thống sẽ có chức năng đăng ký và đăng nhập. Người dùng là khách sau khi đăng ký sẽ đăng nhập vào hệ thống và được thực hiện các chức năng của hệ thống. Admin là người quản trị có chức năng quản lý người dùng, quản lý danh mục.

### 2.1.2 Mối quan hệ giữa các tác nhân

User có quan hệ kế thừa với Khách

### 2.1.3 Xác định các ca sử dụng

Các ca sử dụng của hệ thống : Đăng ký, đăng nhập. CRUD giao dịch, CRUD ví, CRUD ngân sách , thống kê chi tiêu, cập nhật thông tin tài khoản , quản lý người dùng.

### 2.1.4 Xác định quan hệ giữa tác nhân và use case

Khi chưa đăng nhập, khách có thể đăng ký tài khoản mới, đăng nhập, yêu cầu thiết lập lại mật khẩu khi quên mật khẩu, và thiết lập lại mật khẩu khi nhận được chỉ dẫn thiết lập qua email.

Khi khách đăng nhập thành công, hệ thống tạo ra menu chứa các chức năng tương ứng với nhóm người dùng mà người dùng đó thuộc về được mô tả chi tiết ở các phần tiếp theo.

Sau khi đăng nhập, người dùng có thể xem và cập nhật thông tin cá nhân của mình, và có thể thay đổi thay đổi mật khẩu.

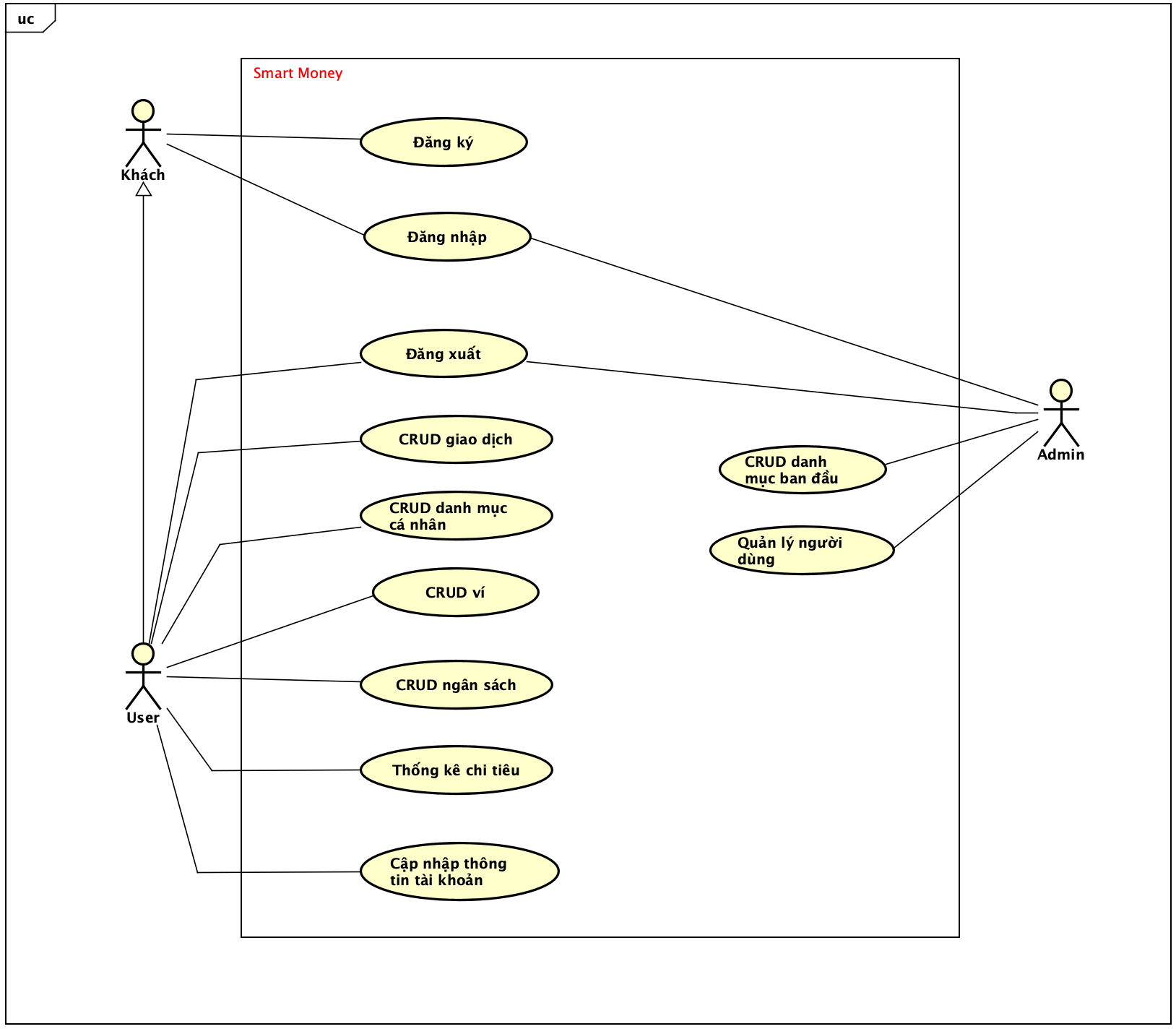
Quản trị viên có thể quản lý người dùng và nhóm người, chức năng và loại chức năng có trong hệ thống.

Đồng thời, quản trị viên có thể phân nhóm cho người dùng, và phân quyền sử dụng chức năng cho nhóm người dùng.

Các use case của người dùng trong biểu đồ use case tổng quan này là use case phức hợp của một nhóm các use case. Chi tiết về các use case phức này được đưa ra trong các biểu đồ phân rã ở phần sau.

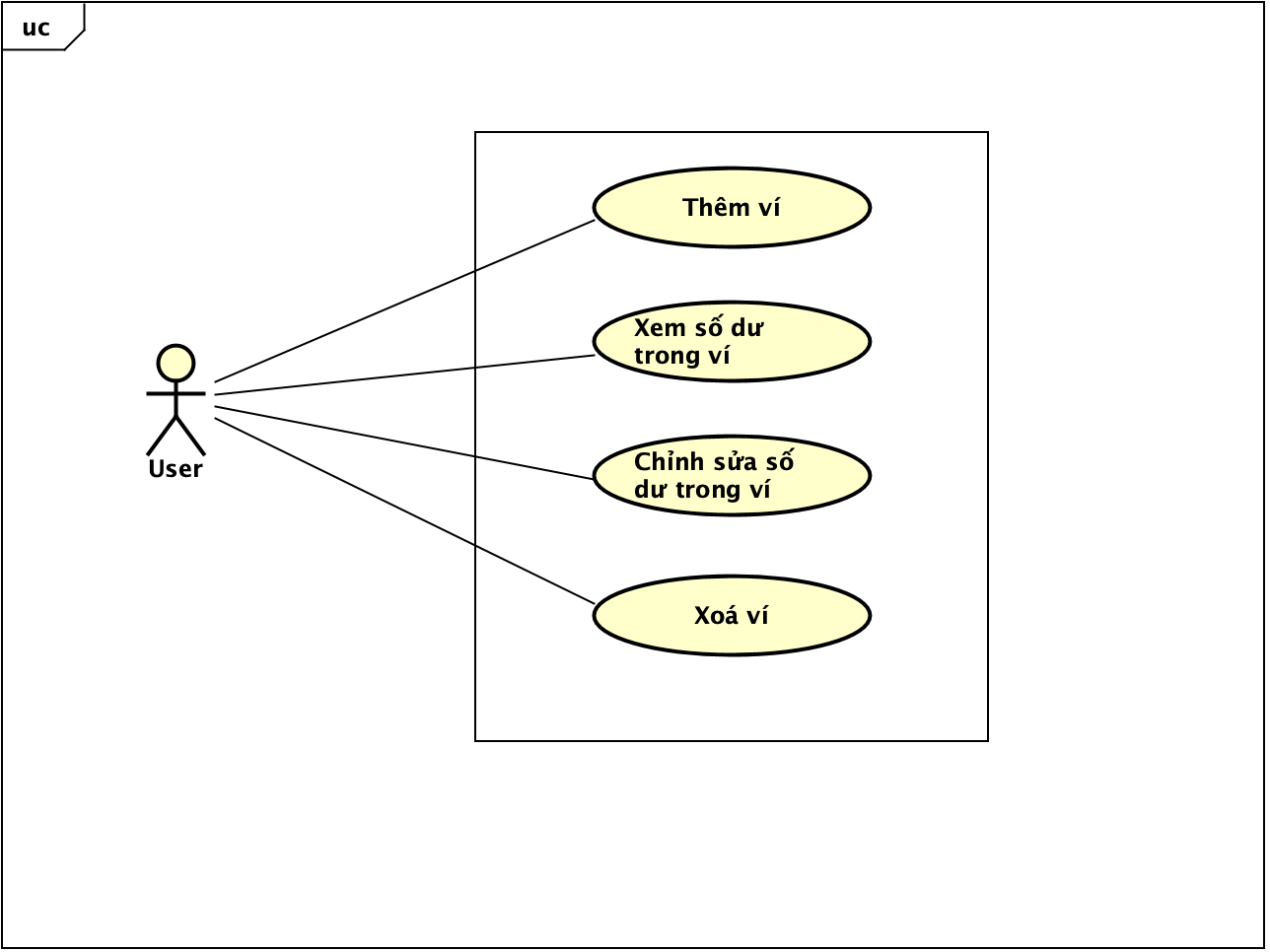
## 2.2 Biểu đồ Use case

### 2.2.1 Biểu đồ Use case tổng quan

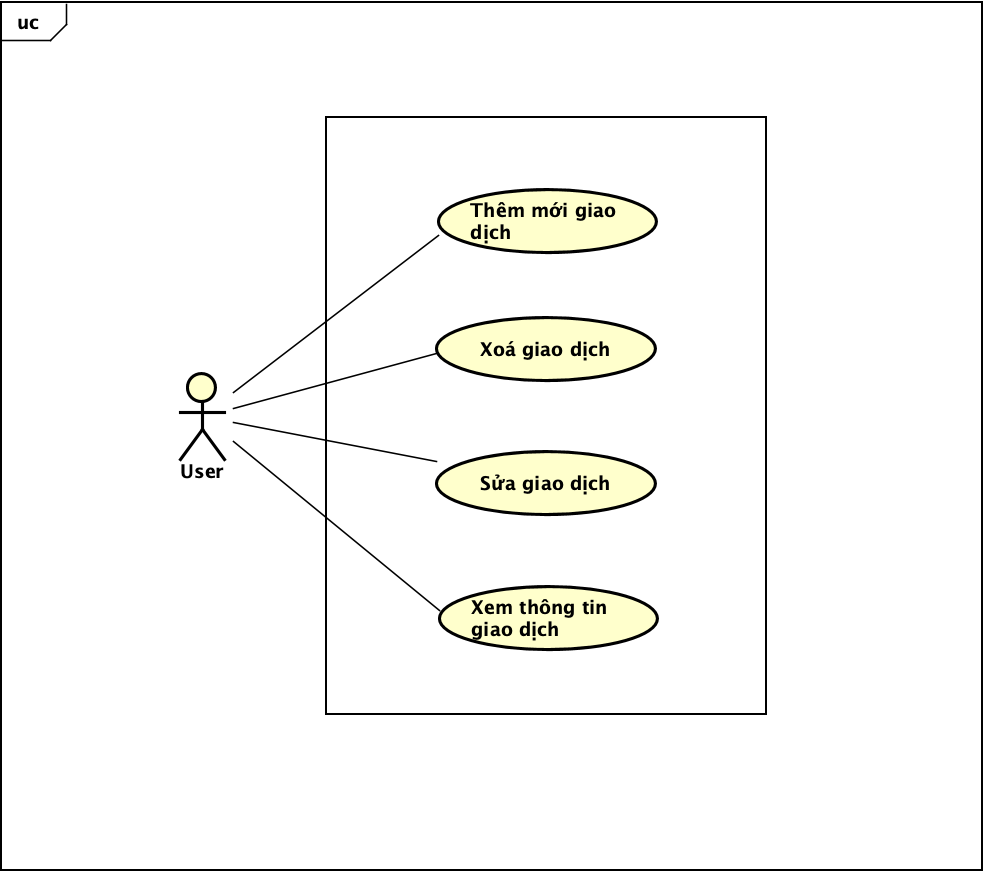


### 2.2.2 Biểu đồ Use case phân rã

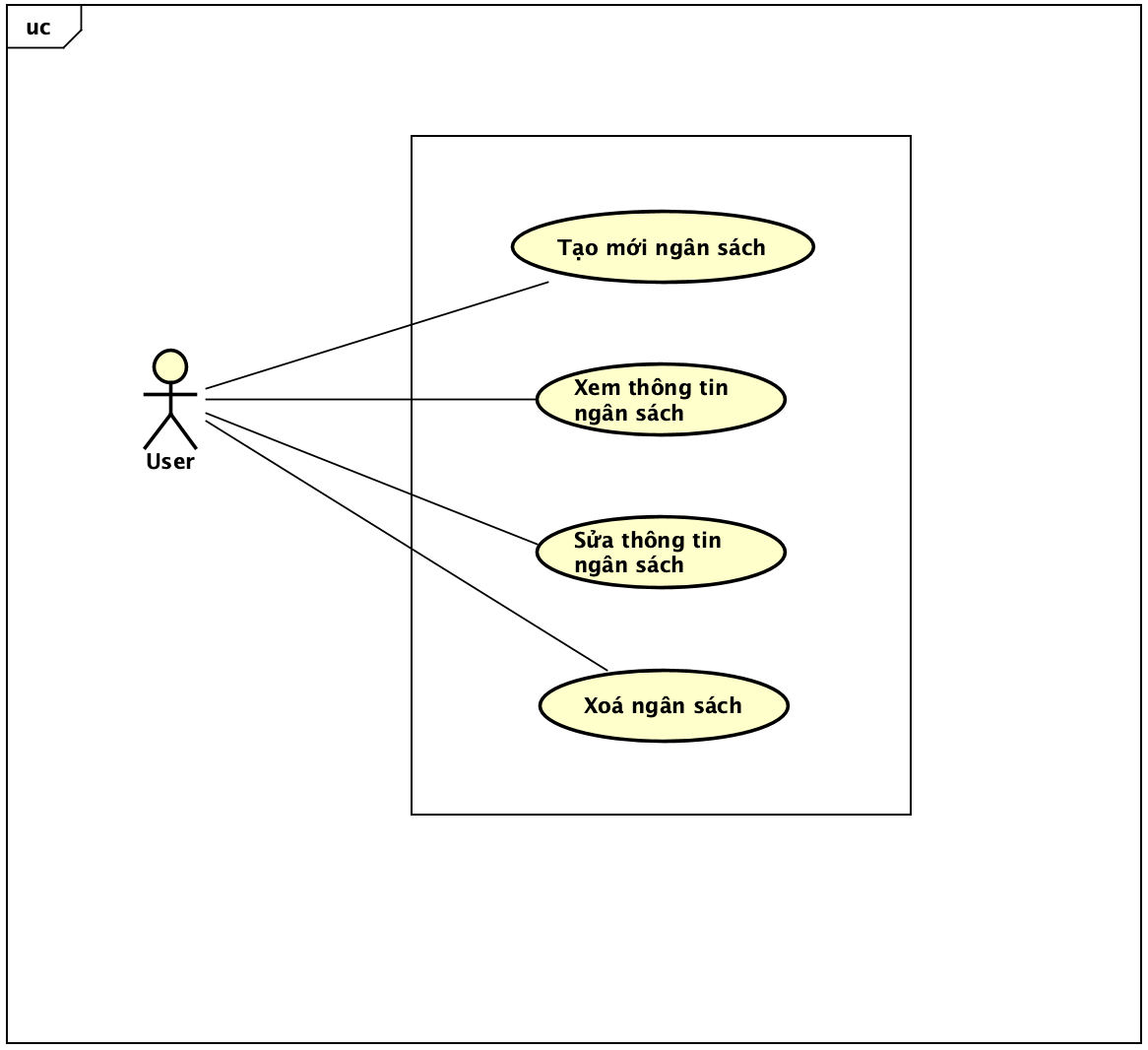
#### Phân rã Use case “CRUD ví”



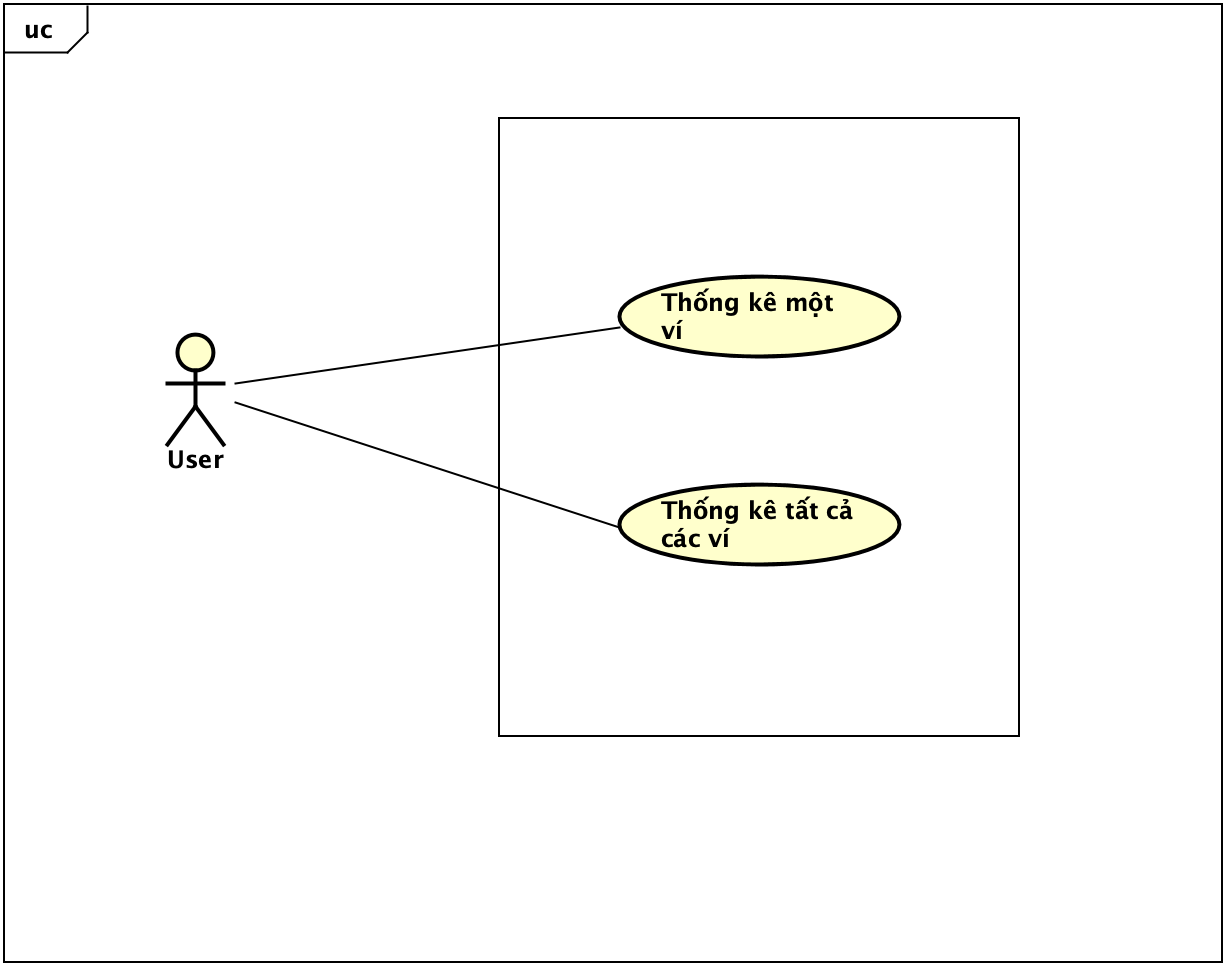
#### Phân rã Use case “CRUD giao dịch”



#### Phân rã Use case “CRUD ngân sách”



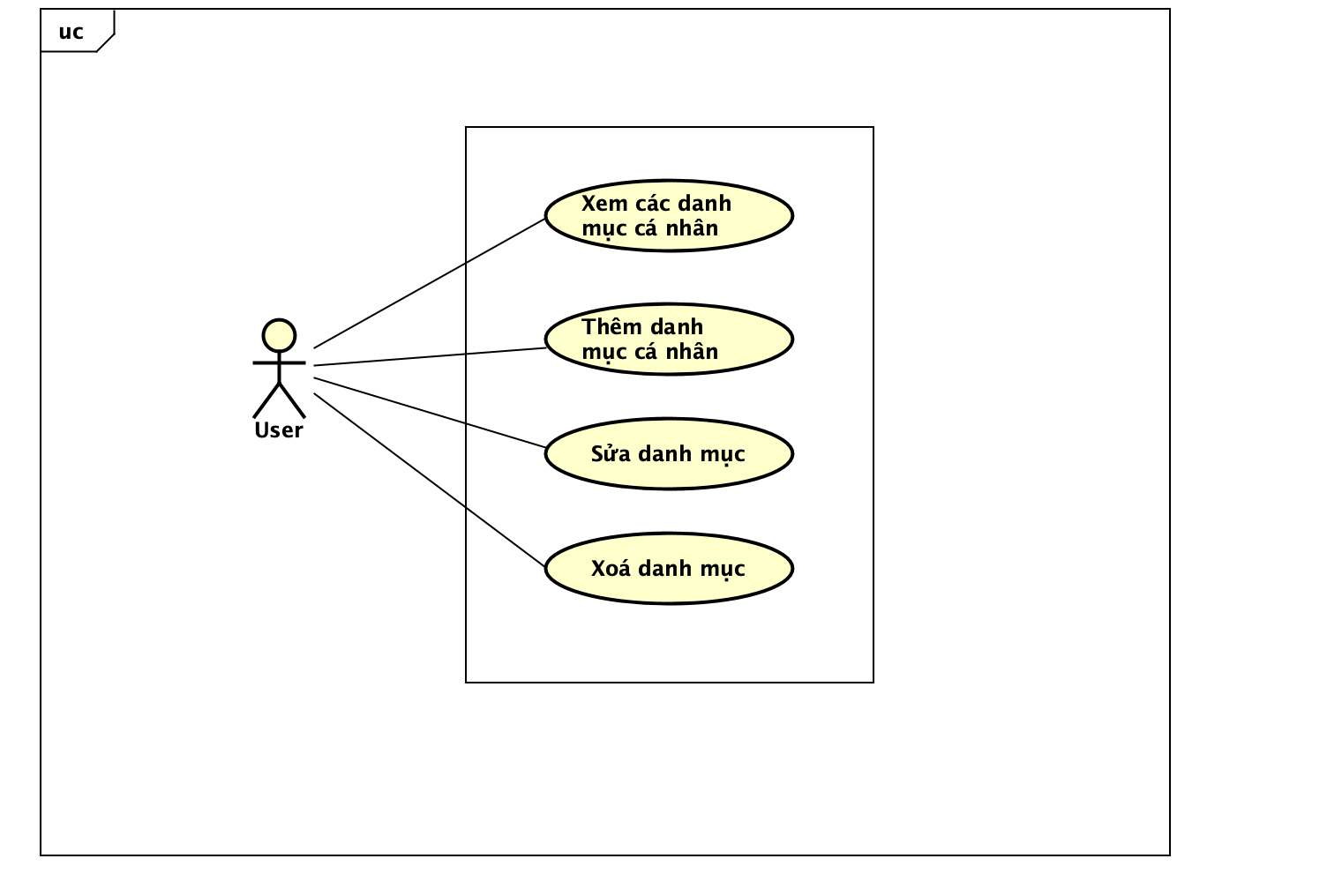
#### Phân rã Use case “Thống kê chi tiêu”



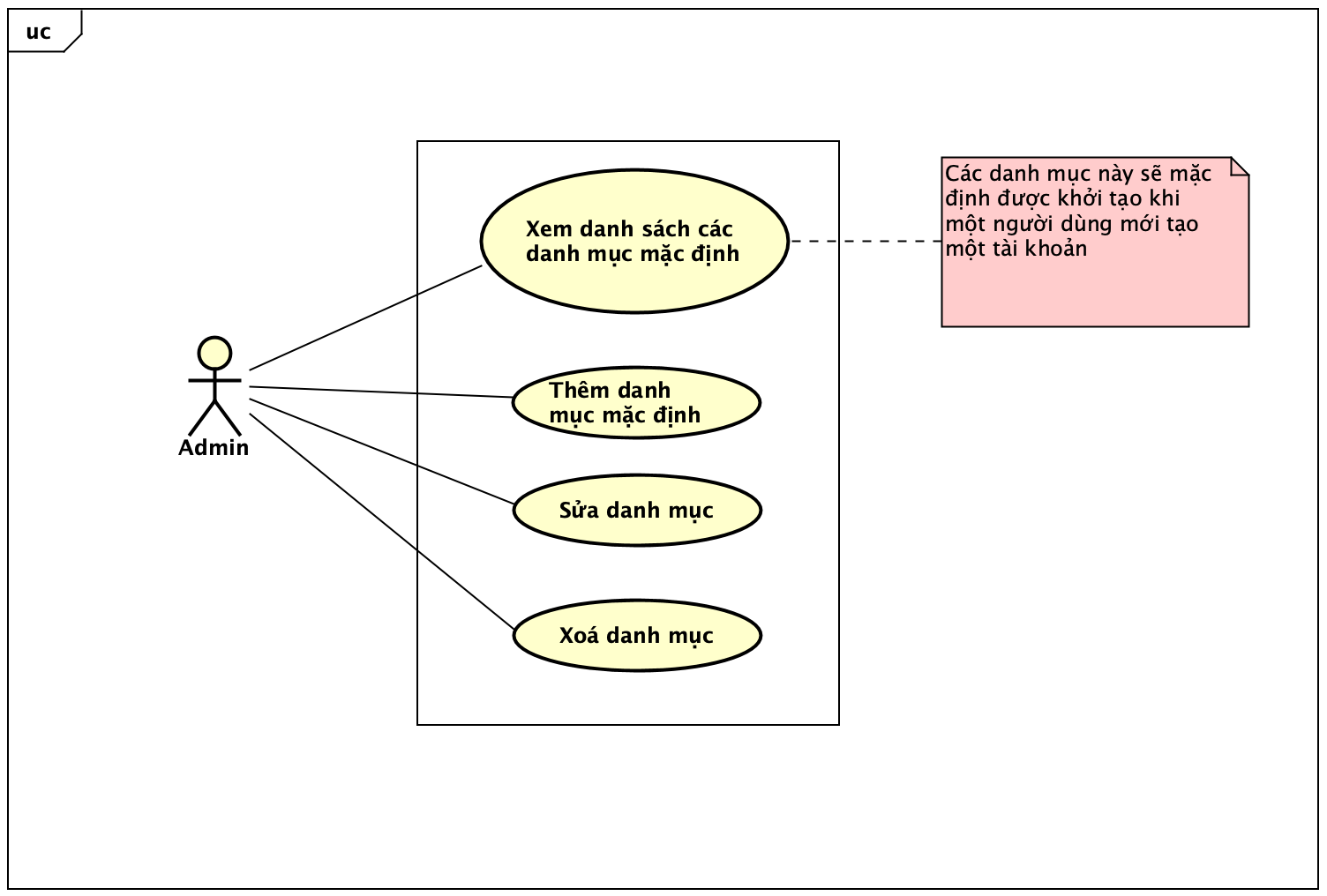
#### Phân rã Use case “Quản lý người dùng”

#### 

#### Phân rã Use case “Quản lý danh mục cá nhân”



#### Phân rã Use case “CRUD danh mục ban đầu”



## 2.3 Đặc tả các chức năng

### Usecase “Đăng nhập, đăng ký ”

Đăng ký : Khi khách hàng chọn đăng ký tài khoản . Hệ thống hiển thị giao diện đăng ký, khách hàng nhập email và mật khẩu .Hệ thống kiểm tra thứ nhất, email người dùng nhập có đúng cú pháp hay không?Nếu email không đúng cú pháp ( ví dụ như thiết @) sẽ yêu cầu người dùng nhập lại đúng email. Sau đó hệ thống kiểm tra email đã tồn tại hay chưa. Nếu đã tồn tại rồi thì báo email đã tồn tại. Sau khi kiểm tra xong email , hệ thống kiểm tra mật khẩu người dùng đặt có đúng yêu cầu hay không? Mật khẩu phải từ 6-12 ký tự. Tiếp theo, mật khẩu nhập lại có đúng mật khẩu ban đầu hay không?Nếu đúng hết các yêu cầu sẽ cho phép tạo tài khoản và chuyển về trang đăng nhập.

Đăng nhập : Khi khách hàng muốn đăng nhập hệ thống , chọn chức năng đăng nhập ->Hệ thống hiện giao diện đăng nhập, khách hàng nhập email và mật khẩu. Hệ thống kiểm tra xem khách hàng đăng nhập đúng hay chưa.Thứ nhất kiểm tra cú pháp email đúng chưa ( có @ hay chưa). Nếu chưa hệ thống gửi thông báo yêu cầu nhập lại email. Sau đó nếu đúng cú pháp, hệ thống kiểm tra xem email đó đăng ký hay chưa. Nếu chưa sẽ yêu cầu người dùng đăng ký. Tiếp theo hệ thống kiểm tra mật khẩu có đúng hay chưa.Nếu sai, hệ thống yêu cầu nhập lại. Nếu đúng, khách đăng nhập vào với tư cách người dùng.

Xóa tài khoản : Nếu đăng nhập là admin, hiển thị giao diện quản lý người dùng dành cho admin. Nếu admin thấy có những tài khoản người dùng lâu không sử dụng, bị bỏ , tài khoản ảo .. -> sẽ xóa tài khoản. Nếu người dùng muốn xóa tài khoản của mình, hệ thống hiển thị những yêu cầu để xóa tài khoản. Người dùng sẽ phải đồng ý, nếu không thì hệ thống không cho phép xóa tài khoản.

### Usecase “Quản lý ví”

Người dùng ấn vào mục ví , hệ thống hiển thị giao diện quản lý ví . Tại giao diện này người dùng có thể tạo ví mới, xóa ví, cập nhật số tiền trong ví.

Nếu người dùng muốn tạo ví, ấn vào mục tạo ví. Hệ thống yêu cầu người dùng nhập vào tổng số tiền thu nhập người đó…..Nếu tạo ví hệ thống kiểm tra tên ví tạo đã tồn tại hay chưa. Nếu đã tồn tại thông báo lỗi. Nếu người dùng muốn cập nhật số dư trong ví của mình ( muốn thêm thu nhập vào ví ) thì ấn mục thêm thu nhập, hệ thống hiện các ví mà người dùng có, sau đó người dùng chọn ví muốn thêm vào. Hệ thống tự cộng vào số dư trong ví đó.

### Usecase “Quản lý giao dịch”

Thêm giao dịch : Khi người dùng ấn vào nút thêm giao dịch, hệ thống hiển thị giao diện:

Thêm khoản chi tiêu : Nếu người dùng chọn chức năng này hiện giao diện bao gồm chọn ví, chọn loại hàng chi tiêu, số tiền chi tiêu bao nhiêu, ngày tháng nào. Khi người dùng điền thông tin đó và chọn lưu, hệ thống lưu và tự động trừ số dư trong ví. Sau đó hệ thống hiển thị giao diện thông tin giao dịch bao gồm số dư trong ví, tiền chi tiêu. Nếu người dùng muốn xem chi tiết thì có thể gọi đến use case tìm kiếm chi tiêu theo thời gian, hoặc use case thống kê để xem chi tiết.

Thêm thu nhập : Hiển thị giao diện yêu cầu người dùng ghi lại thu nhập ( lương). Sau đó hệ thống tự động cộng khoản thu nhập này vào trong ví.

Thêm khoản vay nợ : Người dùng được phép vay từ các khoản khác và hệ thống tự động trừ trong ví.

Tìm kiếm giao dịch : Hệ thống hiển thị giao diện gồm các chuyên mục, ngày, số lượng . Người dùng có thể ấn vào các thời gian mà mình đã chi tiêu, thu nhập , hệ thống hiển thị các thông tin chi tiêu, thu nhập trong thời gian đó.

Xem thông tin giao dịch : Khi người dùng ấn vào mục thống kê ( xem báo cáo) ->Hệ thống hiển thị giao diện đồ thị gồm số tiền đã chi tiêu hoặc thu thập từng ngày trong tháng , và có thể xem chi tiết những ngày đó đã chi tiêu những gì và thu thập bao nhiêu.

Xóa giao dịch : Những giao dịch mà ở thời gian lâu , người dùng muốn xóa có thể ẩn mục xóa giao dịch hiển thị trên giao diện của các giao dịch, hệ thống sẽ xóa các thông tin liên quan đến giao dịch đó .

### Usecase “Quản lý ngân sách”

Người dùng có thể thực hiện các thao tác với budget như là thêm, xóa budget. Mỗi budget sẽ được gán với 1 category xác định. Khi người dùng thực hiện 1 giao dịch ( transaction) vào một danh mục ( category) tương ứng với category của budget, số tiền sẽ được cộng/ trừ vào số dư của budget. Budget sẽ giúp người dùng quản lý chi tiêu chi tiết từng danh mục và các khoản tiền sử dụng cho nó.

### Usecase “Xem thống kê chi tiêu”

Tại giao diện chính của trang web , có hiển thị các thông tin thống kê chi tiêu của người dùng . Người cùng có thể xem số dư hiện có, các khoản chi tiêu thông qua các biểu đồ thống kê. Ngoài ra, người dùng muốn xem chi tiết khoản chi tiêu ấn vào xem chi tiết, hệ thống hiển thị chi tiết khoản chi tiêu đó.

### Usecase “Quản lý người dùng”

Chức năng này dành cho admin. Khi đăng nhập thành công với vai trò admin.Tại giao diện dành cho admin, chọn xem thông tin người dùng. Hệ thống sẽ lấy danh sách các người dùng và hiển thị giao diện có các thông tin người dùng. Trên giao diện này, admin sẽ được phép xóa những tài khoản lâu không sử dụng ( 1-2 năm nếu tài khoản không dùng chưa đến 1 năm không được phép xóa... ) , tuy nhiên admin không có quyền chỉnh sửa thông tin người dùng ( nếu chỉnh sửa hệ thống báo lỗi).

## 2.4 Yêu cầu khác

### Chức năng

* Trong các chuỗi sự kiện của các use case, tất cả các bước có thao tác với CSDL, nếu có lỗi trong quá trình kết nối hoặc thao tác, cần có thông báo lỗi tương ứng để tác nhân biết là lỗi liên quan đến CSDL chứ không liên quan tới lỗi của người dùng
* Các use case do Quản trị viên và Người dùng sử dụng thì Khách cần đăng nhập với vai trò tương ứng
* Định dạng hiển thị chung như sau:
* Số căn phải
* Chữ căn trái
* Font: Arial 14, màu đen
* Nền trắng

### Tính dễ sử dụng

Các chức năng được thiết kế sao cho dễ thao tác. Có hướng dẫn cụ thể lỗi sai của người dùng để người dùng biết định vị lỗi, biết lỗi gì và biết cách sửa lỗi.

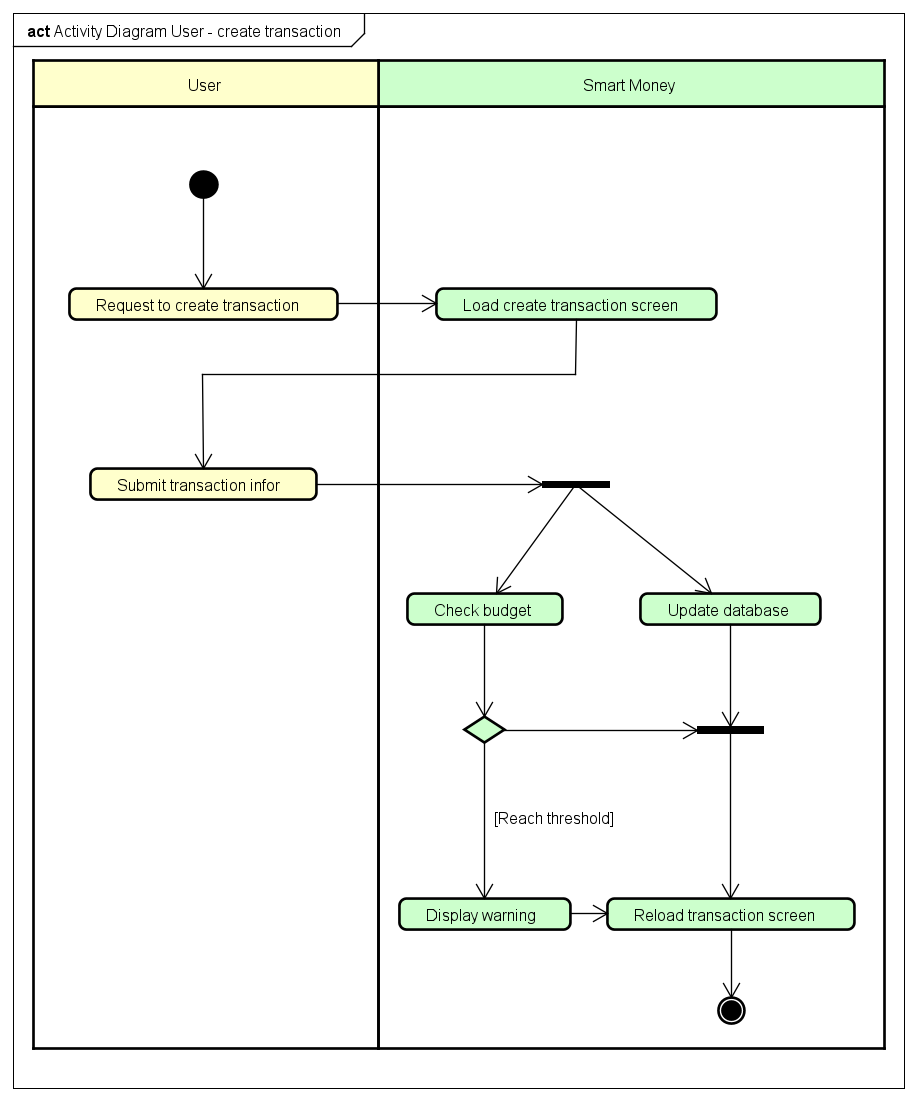
### Các yêu cầu khác

* Thiết kế trang web có tính tin cậy : đảm bảo được những thông tin người dùng không bị lộ.
* Về giao diện thiết kế sao cho người dùng dễ dàng sử dụng, hình ảnh bắt mắt….
* Thiết kế trang web để dễ dàng chỉnh sửa.

# Chương 3. Phân tích yêu cầu

## 3.1 Xây dựng biểu đồ Activity Diagram

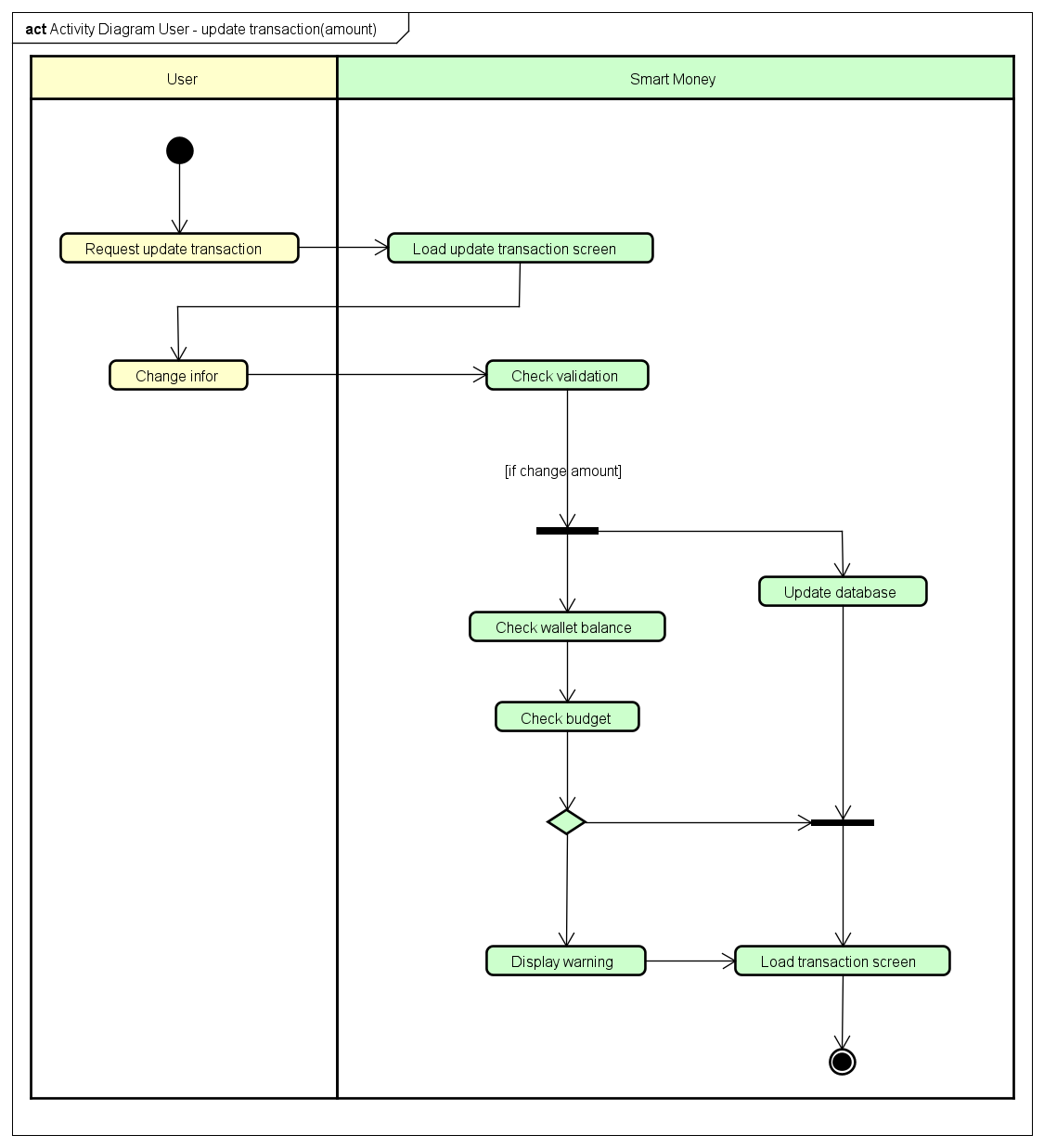
### Quy trình tạo giao dịch mới



Luồng hoạt động:

1. Người dùng nhấn vào button “Create Transaction” gửi yêu cầu tạo một giao dịch mới đến hệ thống.
2. Hệ thống chuyển sang màn hình tạo giao dịch mới gồm các trường thông tin để người dùng nhập.
3. Người dùng thực hiện nhập các thông tin: Tên giao dịch, Danh mục chi tiêu, Ghi chú, Ngày giao dịch, Nhãn, Số tiền, Đơn vị tiền. Trong đó, trường Danh mục chi tiêu là trường select, trường Tên giao dịch, Số tiền, Đơn vị tiền, Ngày giao dịch là các trường bắt buộc điền. Các trường còn lại có thể bỏ trống. Sau đó, nhấn Submit.
4. Với thông tin được nhập từ người dùng, hệ thống sẽ kiểm tra nó với số tiền và tên danh mục của các Ngân sách người dùng đã tạo trước đó. Nếu có tồn tại ngân sách trùng với tên danh mục sẽ tiến hành kiểm tra số dư của ngân sách còn đủ hay không. Nếu đủ, thực hiện lưu vào Database và thông báo tạo giao dịch thành công. Nếu không đủ, hiển thị thông báo với người dùng và trở lại màn hình các giao dịch.

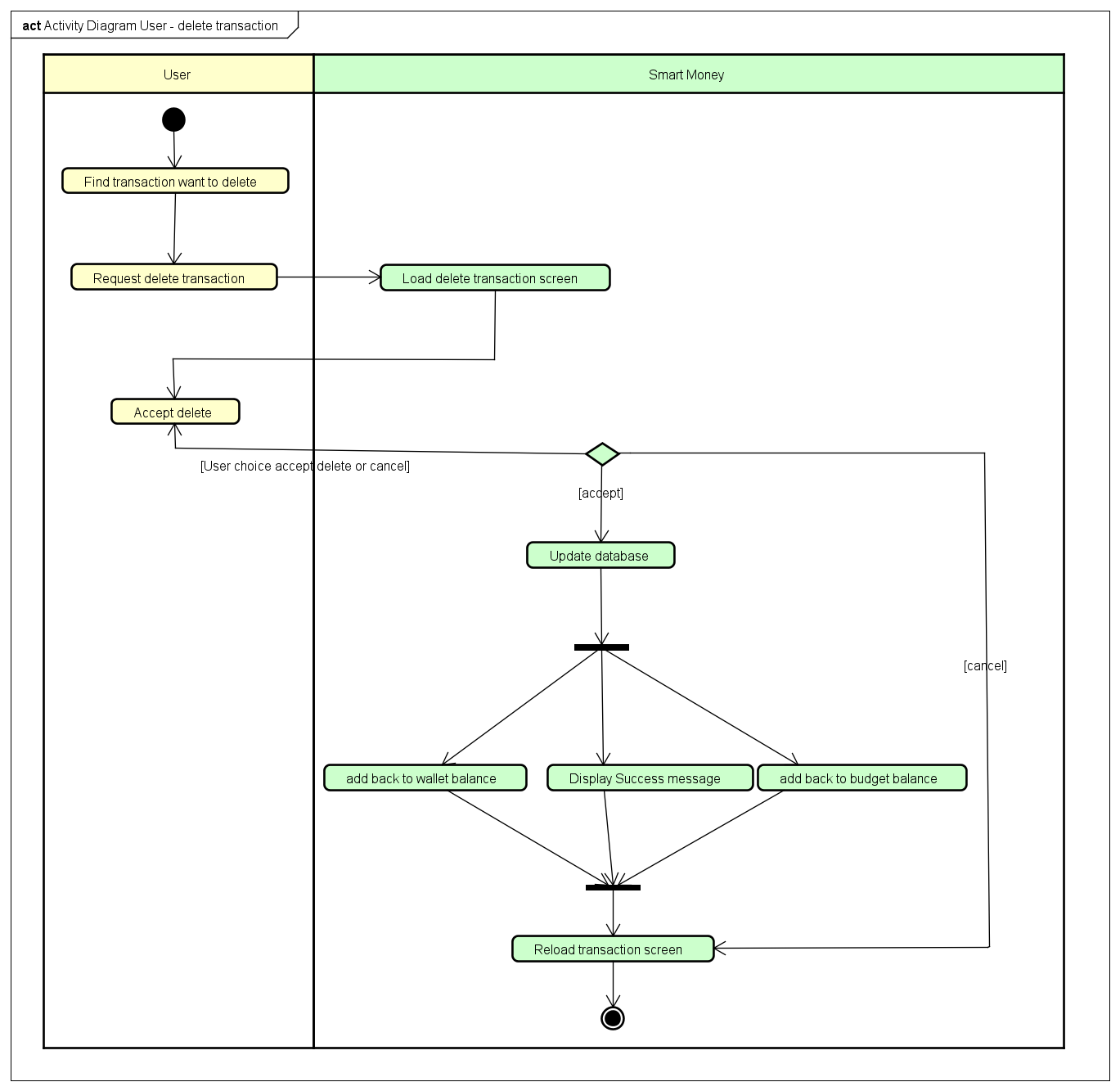
### Quy trình chỉnh sửa một giao dịch



Luồng hoạt động:

1. Người dùng tìm đến giao dịch muốn sửa, nhấn vào button “Update Transaction” gửi yêu cầu chỉnh sửa một giao dịch đến hệ thống.
2. Hệ thống chuyển sang màn hình chỉnh sửa giao dịch gồm các trường thông tin để người dùng nhập.
3. Người dùng thực hiện chỉnh sửa các thông tin: Tên giao dịch, Danh mục chi tiêu, Ghi chú, Ngày giao dịch, Nhãn, Số tiền, Đơn vị tiền. Sau đó, nhấn Submit.
4. Hệ thống kiểm tra thông tin chỉnh sửa của người dùng, nếu người dùng thực hiện chỉnh sửa danh mục có tạo ngân sách, cần thực hiện cập nhật thông tin của ngân sách tương ứng. Kiểm tra số tiền người dùng nhập với số tiền hiện có trong ví tương ứng, nếu phù hợp cập nhật database và thông báo cập nhật thành công. Nếu không thông báo lỗi đến người dùng và quay lại trang các giao dịch.

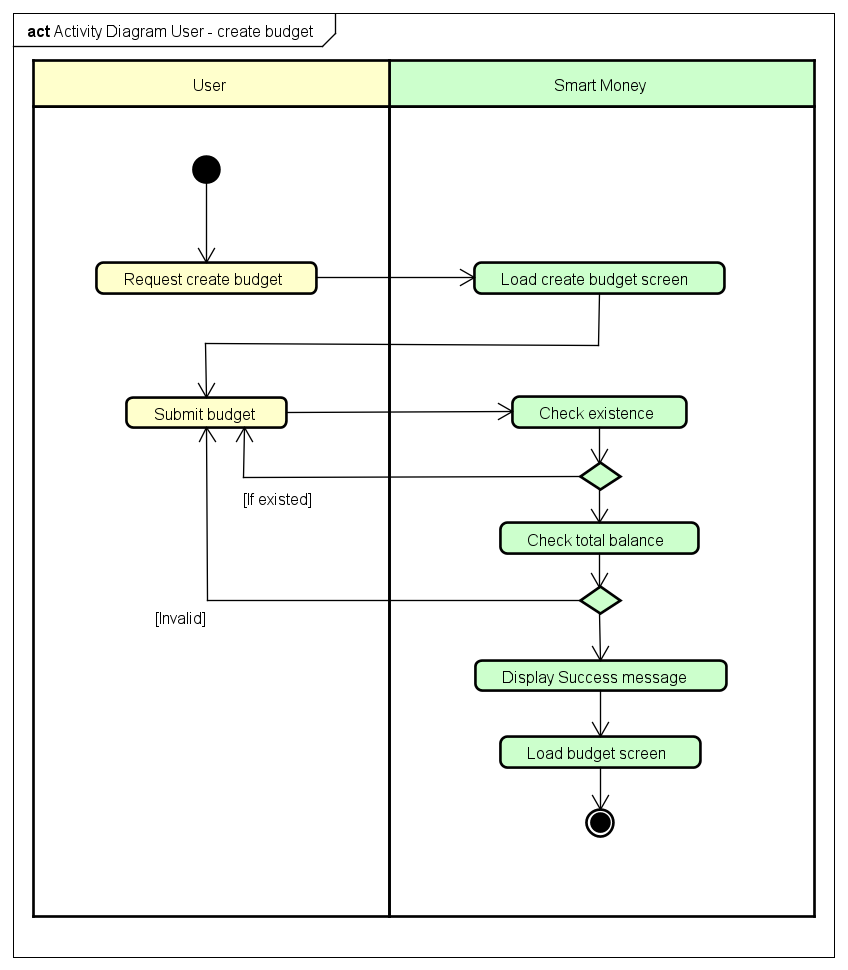
### Quy trình xóa một giao dịch



Luồng hoạt động:

1. Người dùng tìm đến giao dịch muốn xóa, nhấn vào button “Delete Transaction” gửi yêu cầu xóa một giao dịch đến hệ thống.
2. Hệ thống chuyển sang màn hình xác nhận xóa giao dịch gồm các button Accept và Cancel.
3. Nếu Accept, hệ thống xóa database của giao dịch tương ứng, cộng tiền trả lại vào ví và ngân sách tương ứng nếu có. Hiển thị thông báo xóa thành công.
4. Nếu Cancel, hệ thống hiển thị lại trang các giao dịch.

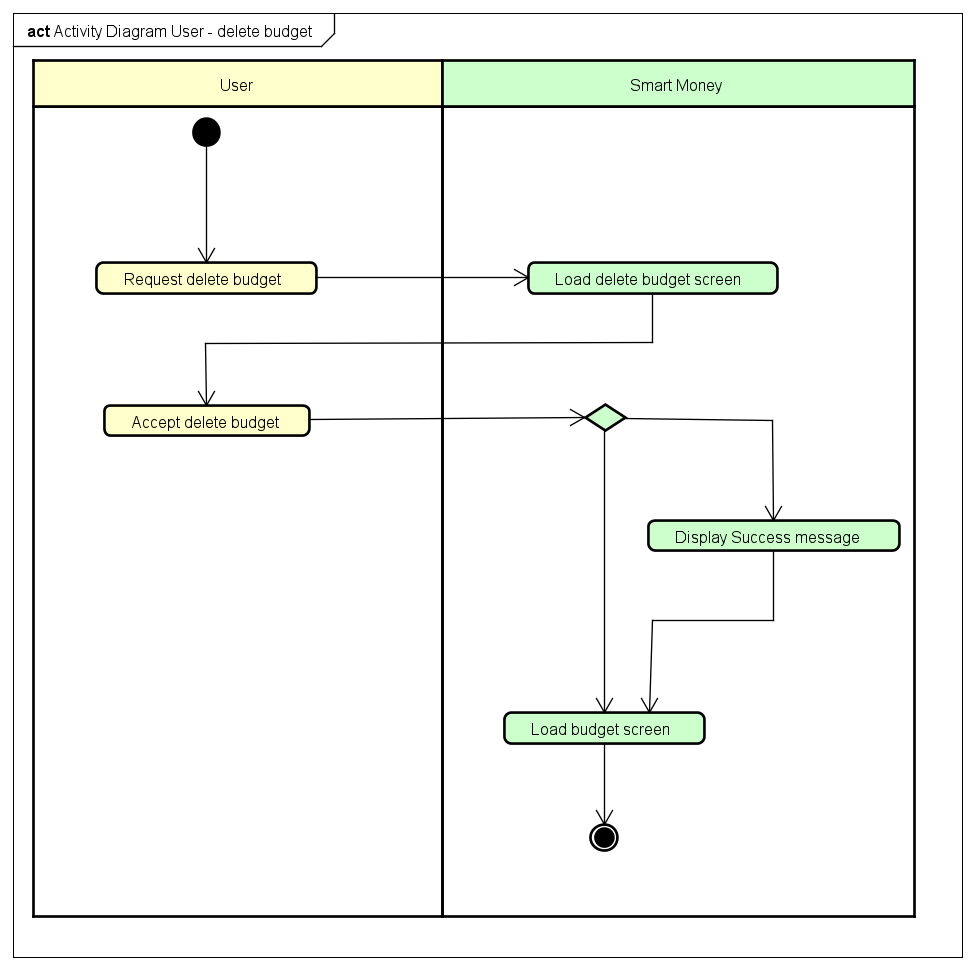
### Quy trình tạo một ngân sách



Luồng hoạt động:

1. Người dùng nhấn vào button “Create Budget” gửi yêu cầu tạo một ngân sách mới đến hệ thống.
2. Hệ thống chuyển sang màn hình tạo ngân sách mới gồm các trường thông tin để người dùng nhập.
3. Người dùng thực hiện nhập các thông tin: Tên ngân sách, Danh mục chi tiêu, Số tiền, Đơn vị tiền. Trong đó, trường Danh mục chi tiêu là trường select, trường Tên giao dịch, Số tiền, Đơn vị tiền,là các trường bắt buộc điền. Sau đó, nhấn Submit.
4. Với thông tin được nhập từ người dùng, hệ thống sẽ kiểm tra nó với tên danh mục của các Ngân sách người dùng đã tạo trước đó. Nếu không tồn tại ngân sách trùng với tên danh mục sẽ tiến hành kiểm tra số dư của ví còn đủ hay không. Nếu đủ, thực hiện lưu vào Database và thông báo tạo ngân sách thành công. Nếu lỗi, hiển thị thông báo với người dùng và trở lại màn hình các giao dịch.

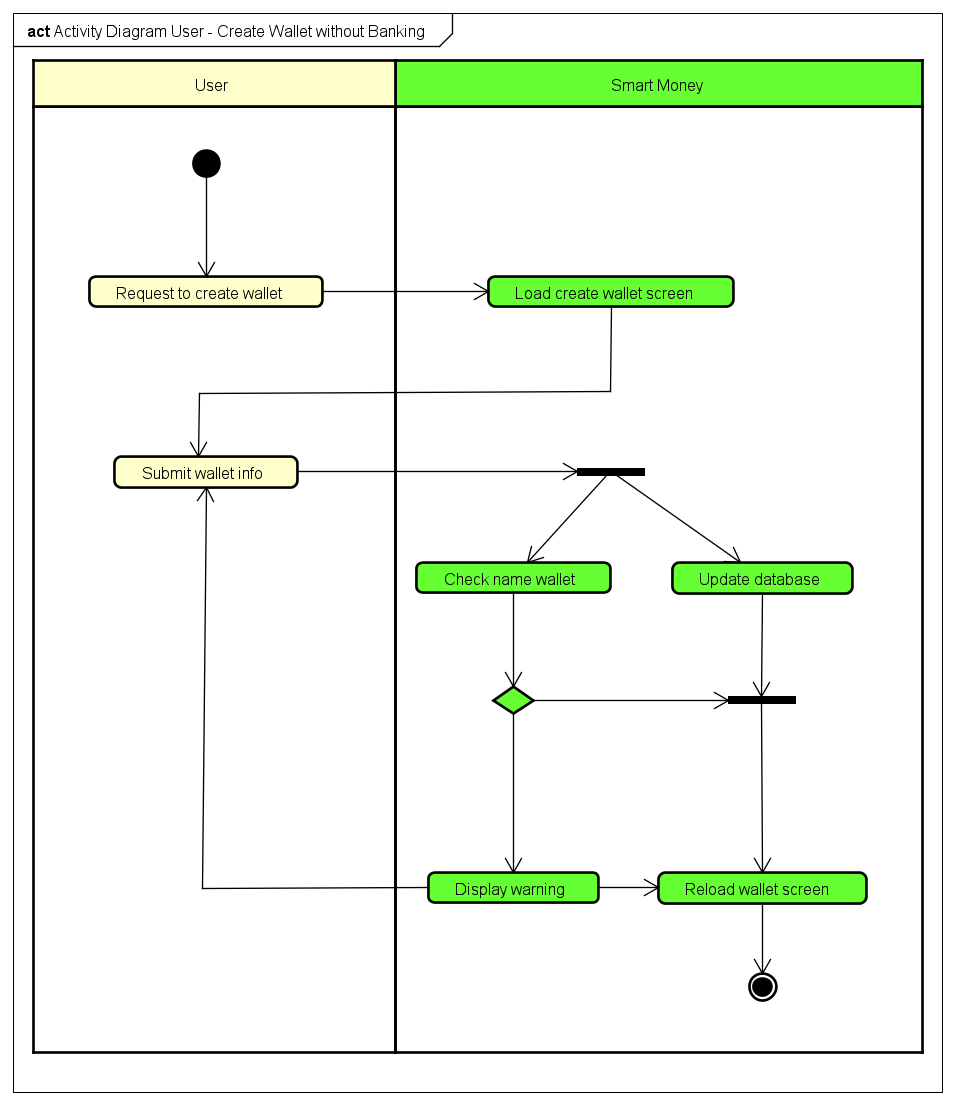
### Quy trình xóa ngân sách



Luồng hoạt động :

1. Người dùng tìm đến ngân sách muốn xóa, nhấn vào button “Delete Budget” gửi yêu cầu xóa một ngân sách đến hệ thống.
2. Hệ thống chuyển sang màn hình xác nhận xóa ngân sách gồm các button Accept và Cancel.
3. Nếu Accept, hệ thống xóa database của ngân sách tương ứng. Hiển thị thông báo xóa thành công.
4. Nếu Cancel, hệ thống hiển thị lại trang ngân sách.

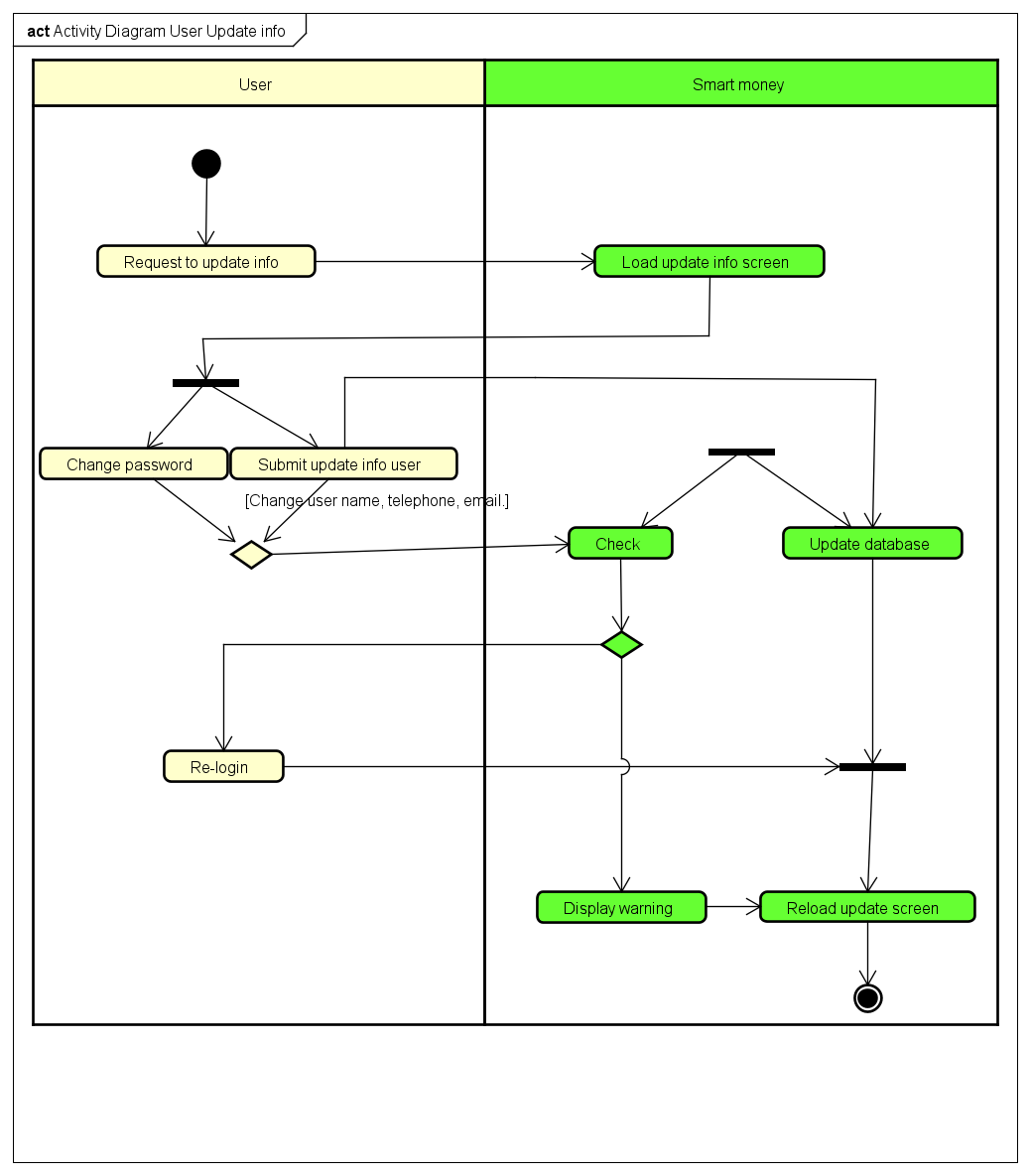
### Quy trình tạo ví



Luồng hoạt động :

1. Người dùng ấn vào nút “Create Wallet”, chọn tạo “Create wallet”
2. Hệ thống sẽ hiển thị giao diện tạo ví mới cho người dùng.
3. Sau đó người dùng nhập các thông tin liên quan đến ví ( tên ví, số dư, đơn vị tiền). Người dùng ấn “submit” , các thông tin được gửi lên hệ thống.
4. Hệ thống sẽ kiểm tra tên ví đã tồn tại hay chưa, nếu tồn tại rồi sẽ báo lỗi cho người dùng, nếu chưa hệ thống lưu dữ liệu. Và gửi về cho người dùng tạo ví thành công.

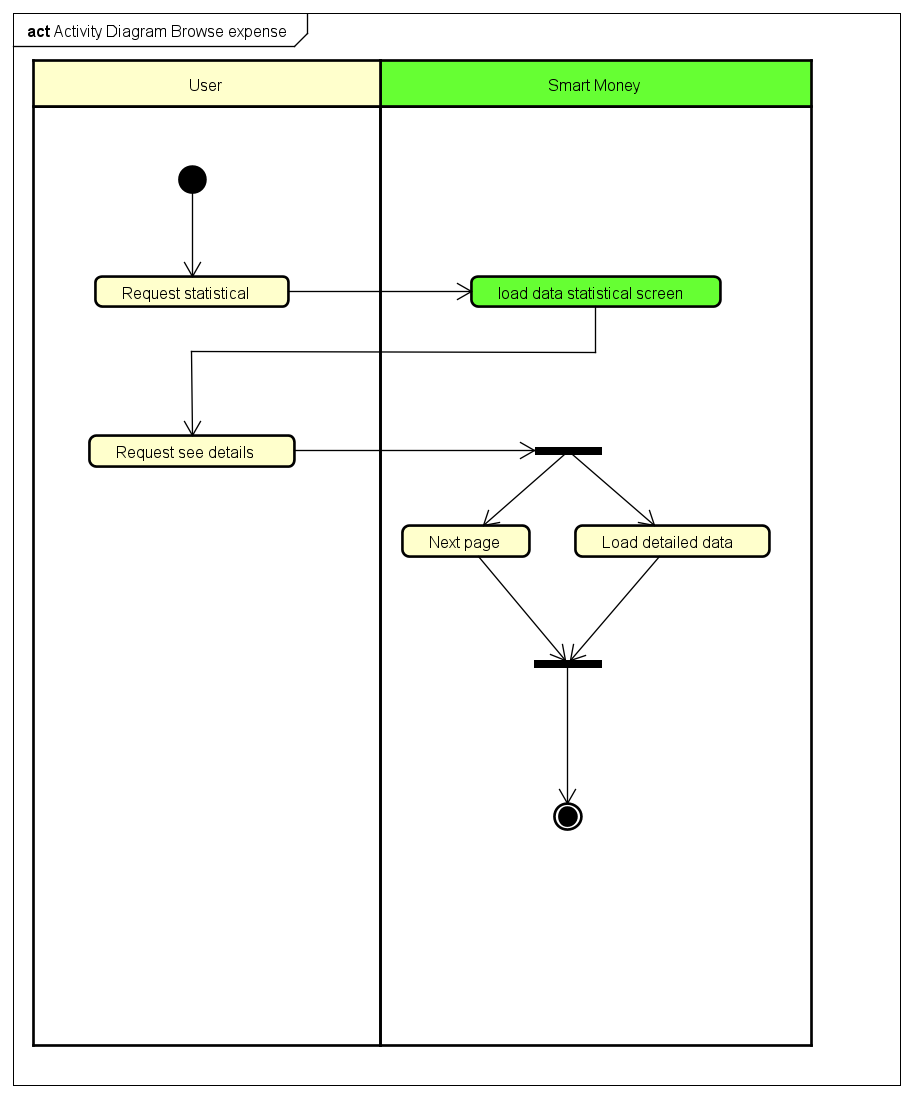
### Quy trình người dùng thay đổi thông tin



Luồng hoạt động :

1. Người dùng ấn vào nút “Setting” chọn “Update information” .
2. Hệ thống hiển thị giao diện thay đổi thông tin.
3. Tại đây người dùng sẽ chọn “Change password” hoặc “Account”. Nếu chọn “Account” , hệ thống hiển thị giao diện các thông tin : Ảnh đại diện, tên, email . Sau khi người dùng thay đổi thông tin ấn “Update”.
4. Hệ thống sẽ lưu các dữ liệu thay đổi.
5. Nếu người dùng chọn “Change password” , hệ thống yêu cầu người dùng nhập mật khẩu cũ , mật khẩu mới.
6. Kiểm tra đúng hệ thống yêu cầu người dùng đăng nhập lại vào hệ thống.

### Quy trình xem thống kê

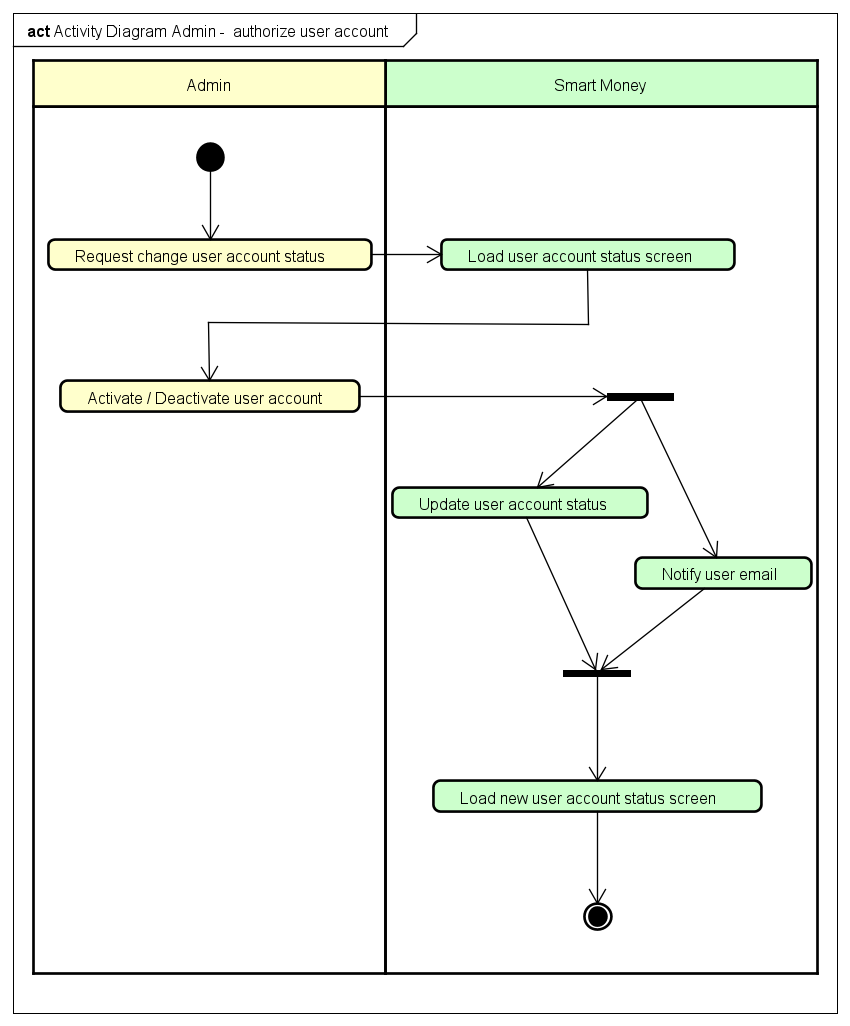


Luồng hoạt động:

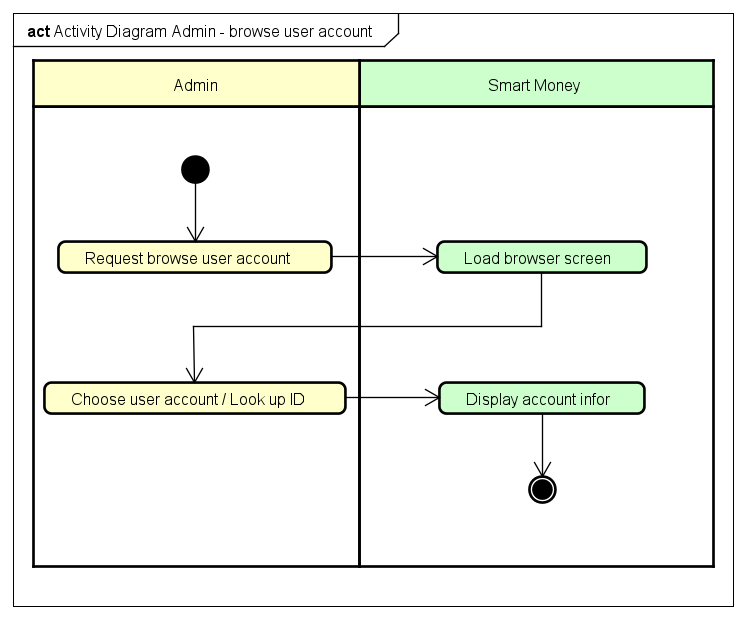
1. Người dùng ấn trang chủ, yêu cầu xem chi tiết thống kê .
2. Hệ thống sẽ load dữ liệu thống kê và chuyển trang mới.
3. Đưa ra màn hình thống kê cho người dùng.

### Quy trình quản lý người dùng (Admin)

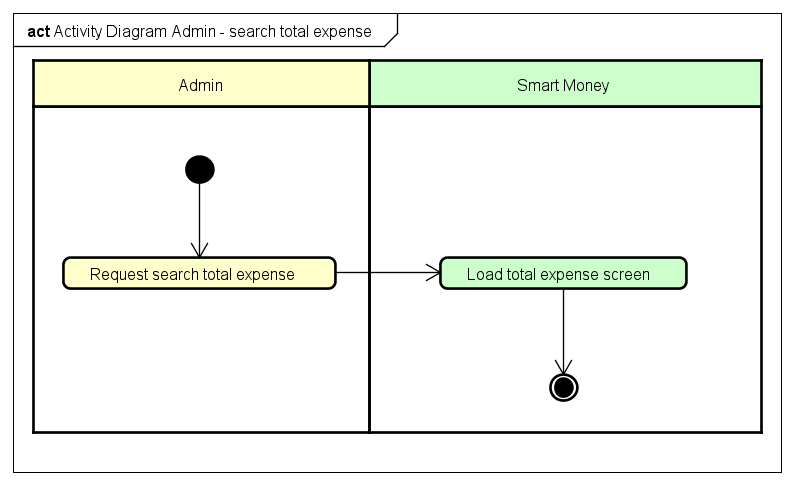
#### Authorize user account



#### Browse user account

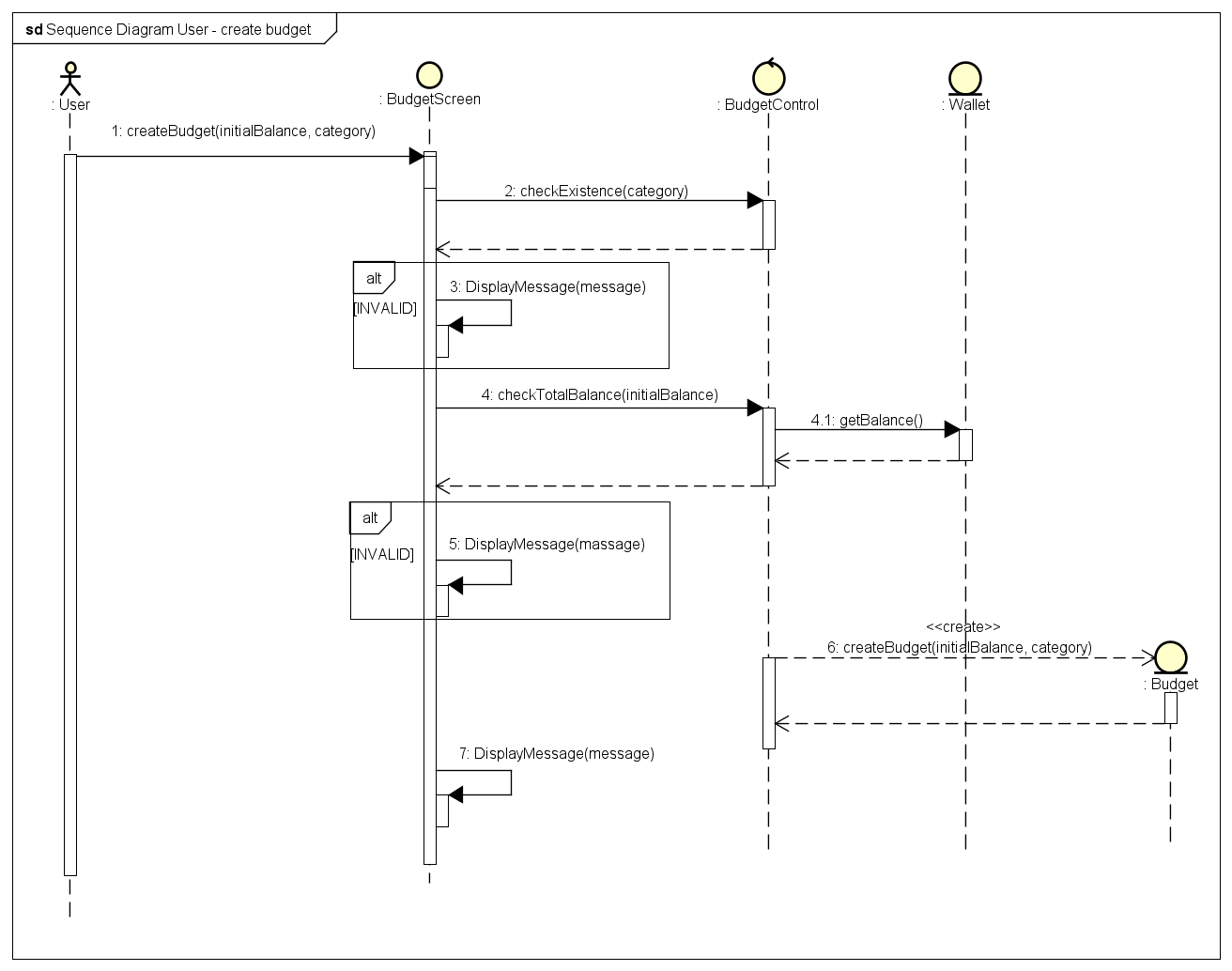


#### Search total expense

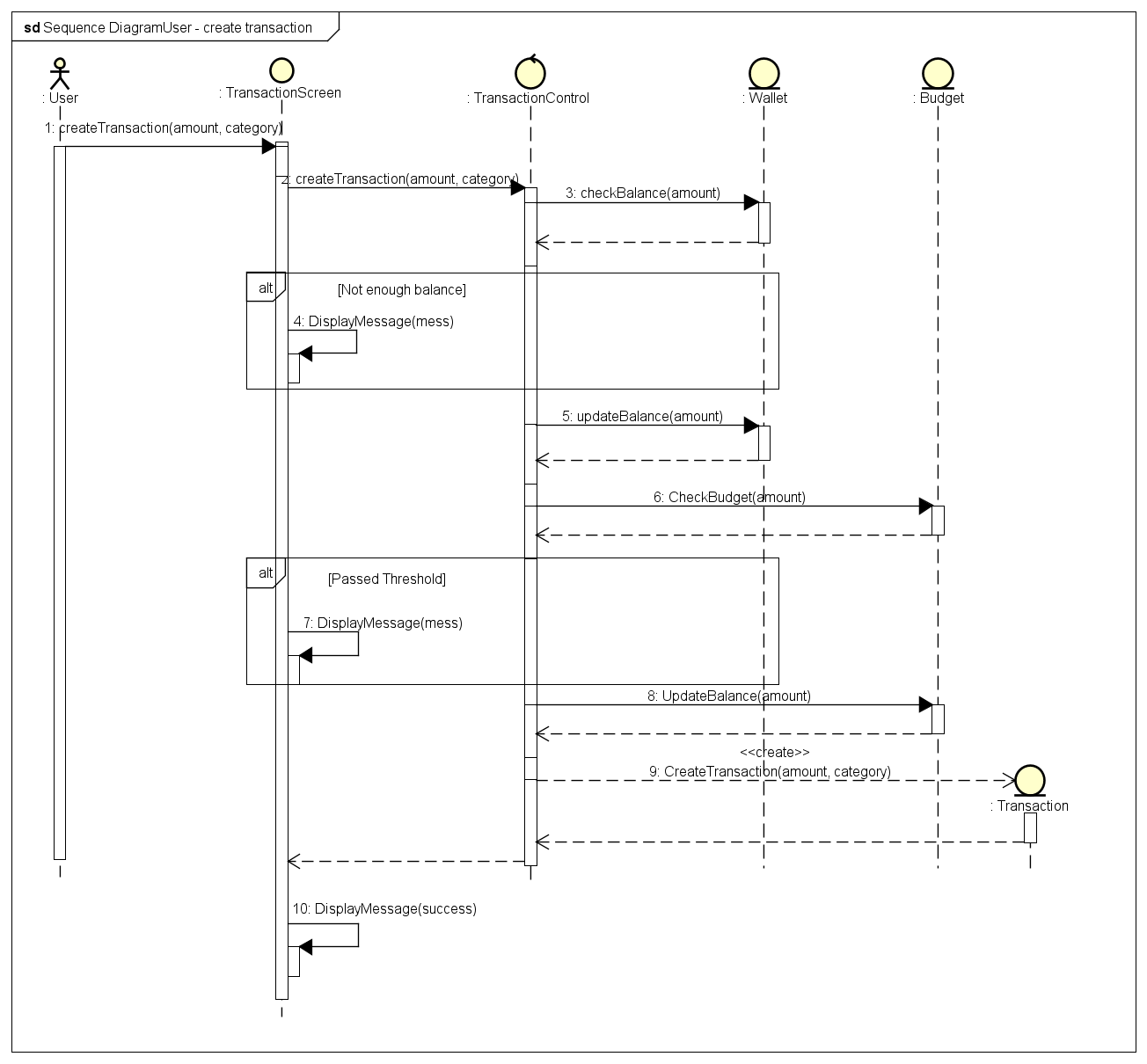


## 3.2 Xây dựng biểu đồ Sequence Diagram

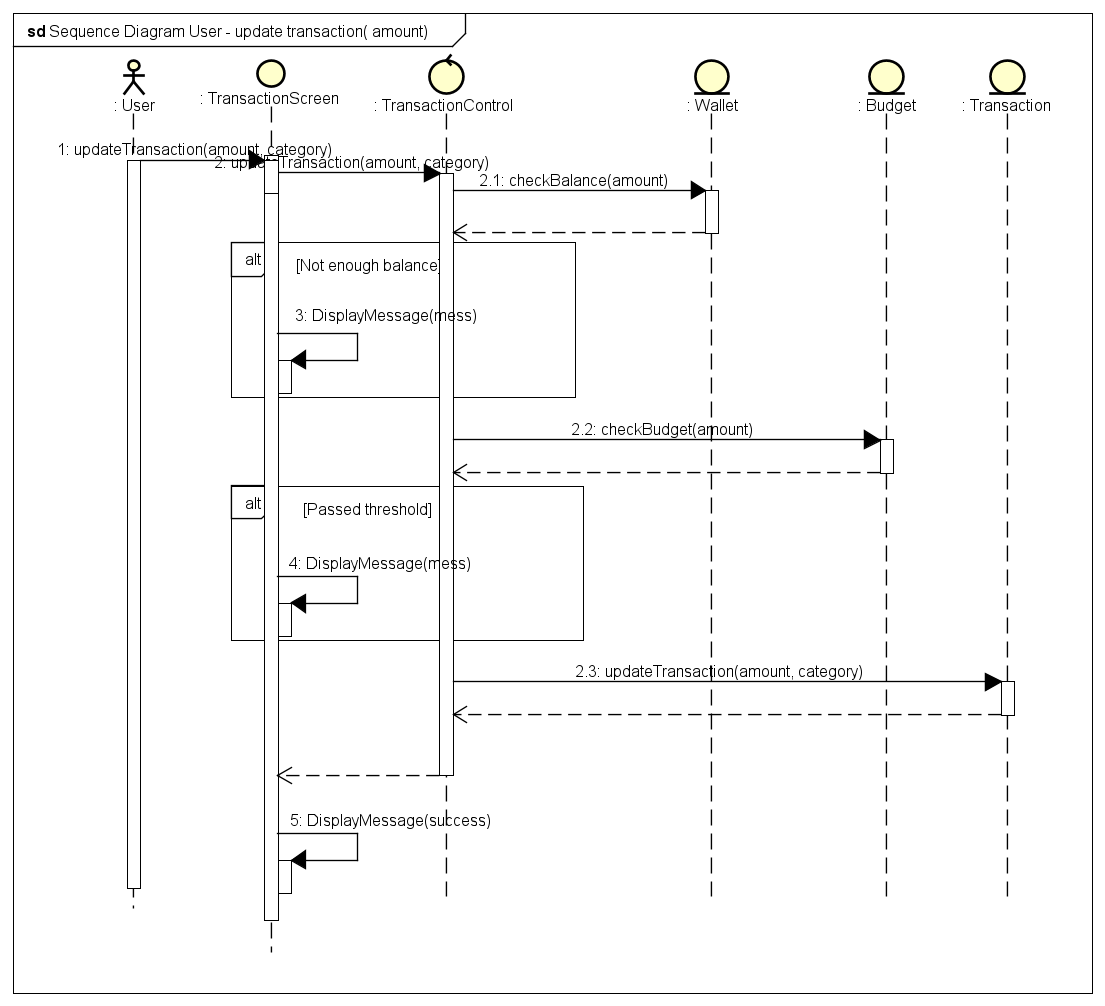
### Sequence diagram User – Create budget



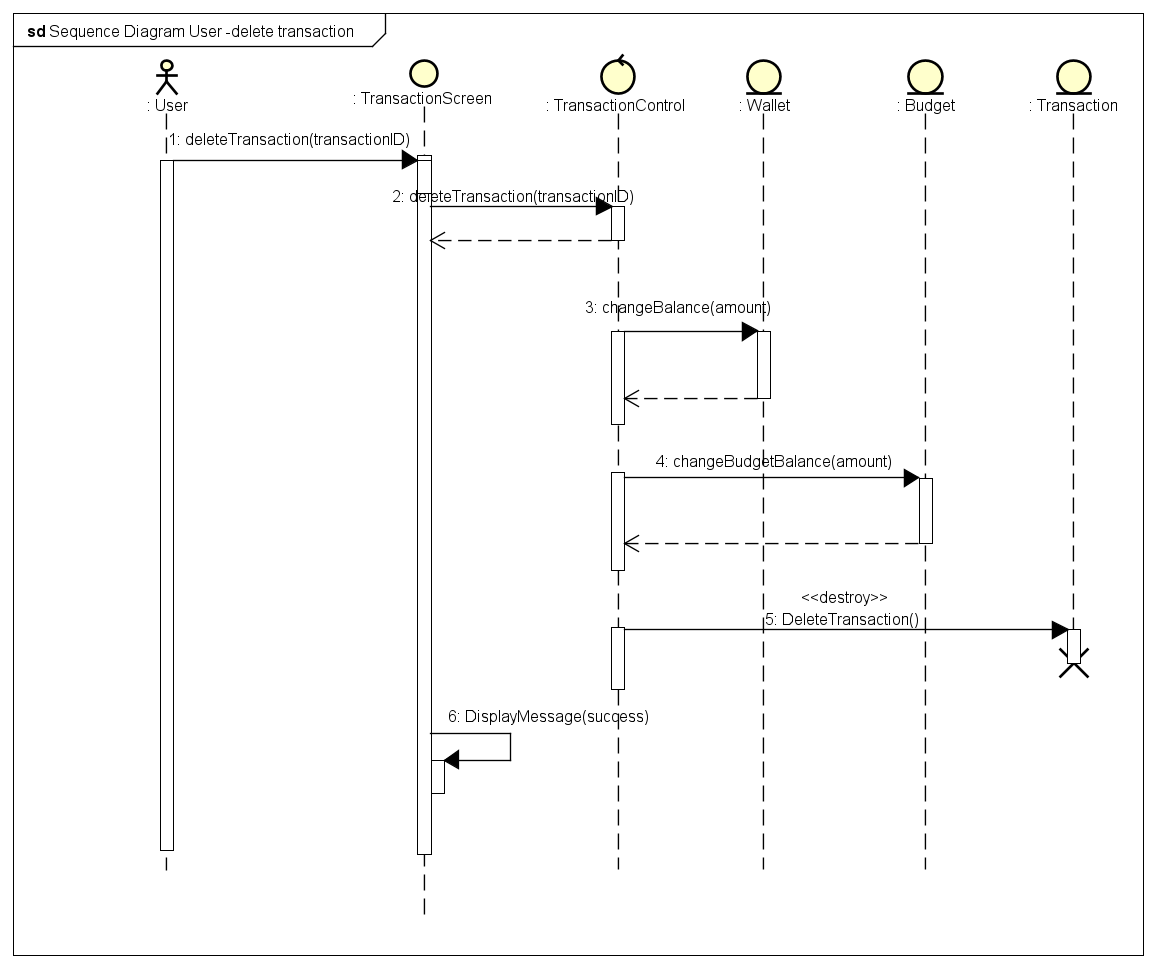
### Sequence diagram User – Create transaction



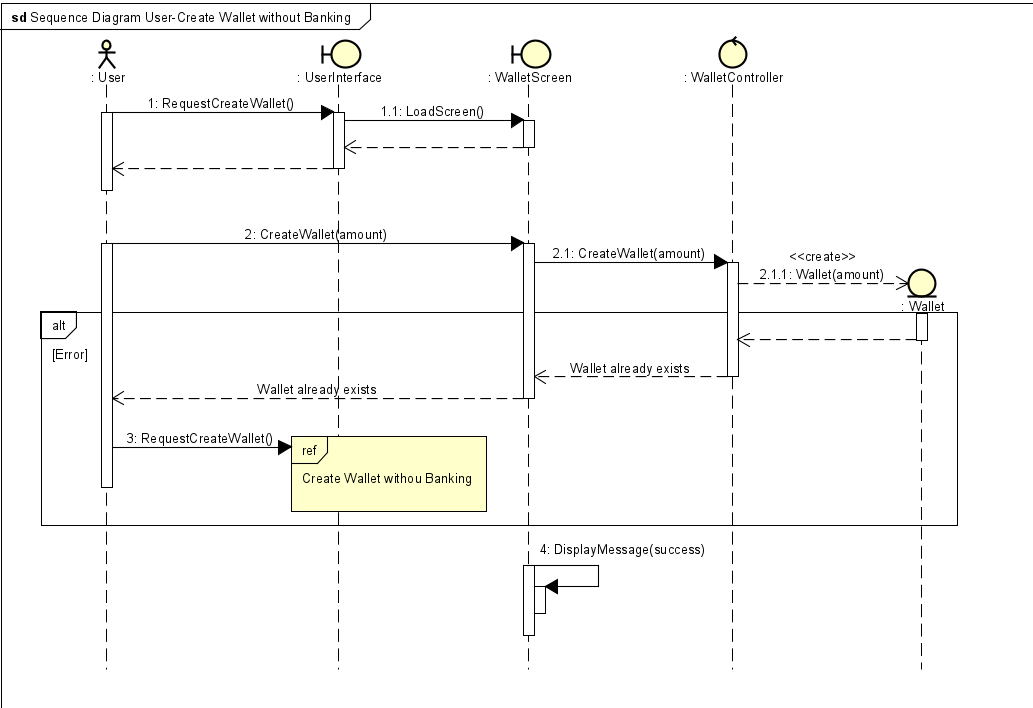
### Sequence diagram User – Update transaction



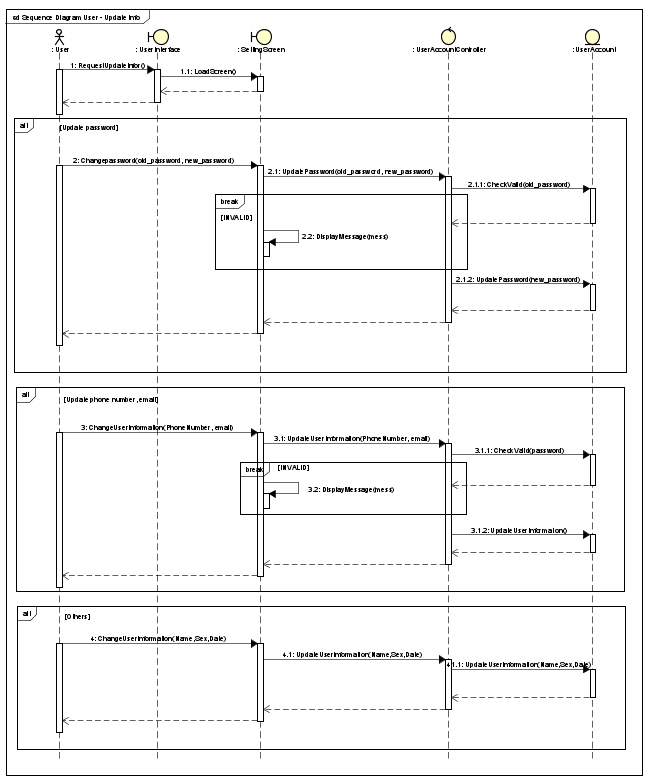
### Sequence diagram User – Delete transaction



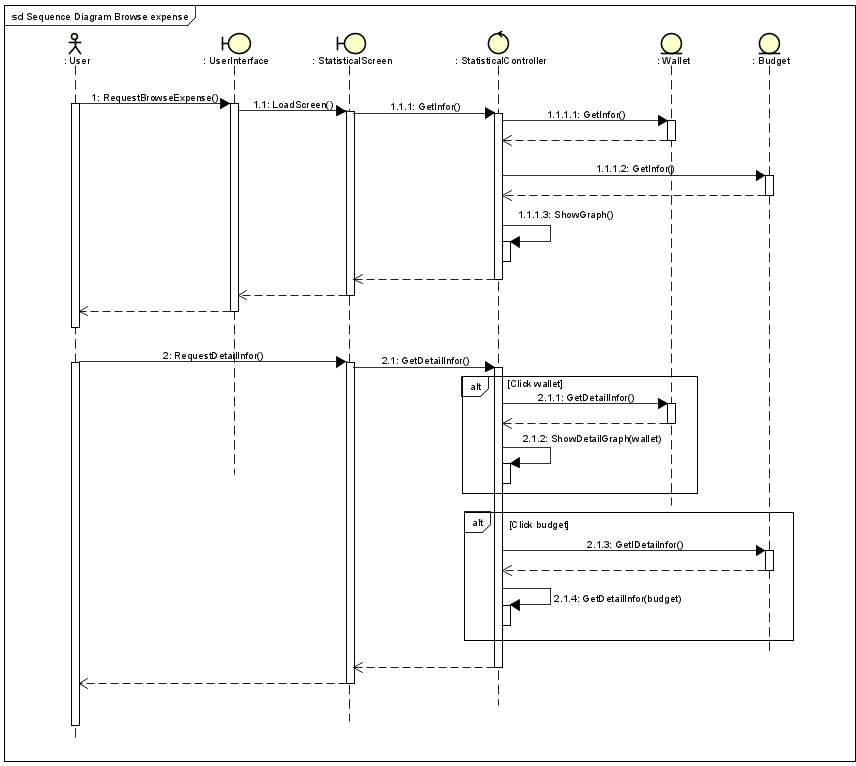
### Sequence diagram User – Create Wallet



### Sequence diagram User – Update Information

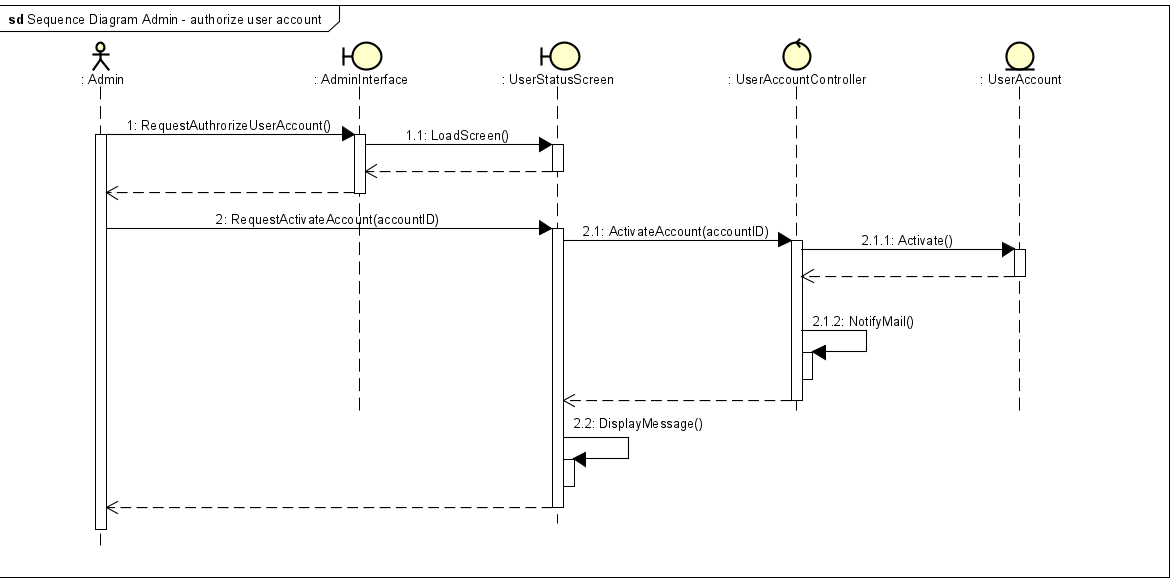


### Sequence diagram User – Browse expense

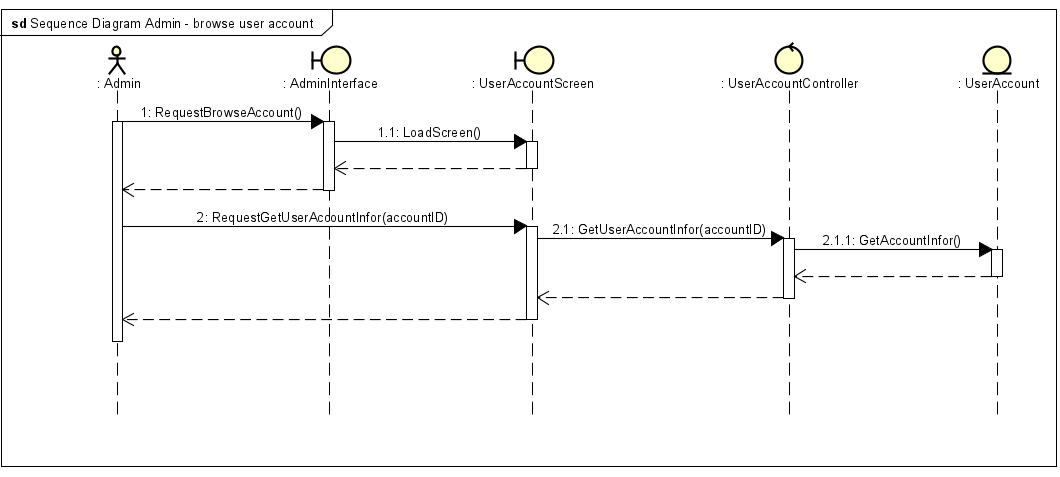


### Sequence diagram Admin

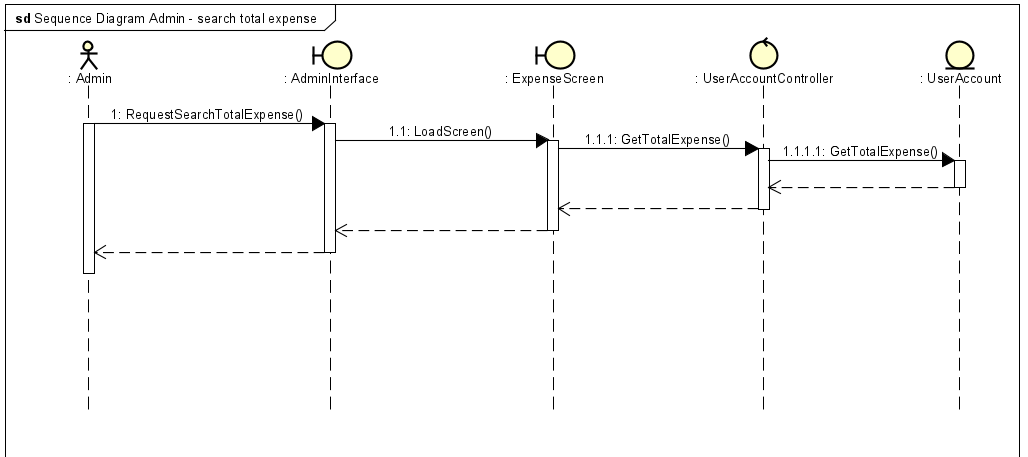
#### Sequence diagram Authorize user account



#### Sequence diagram Browse user account

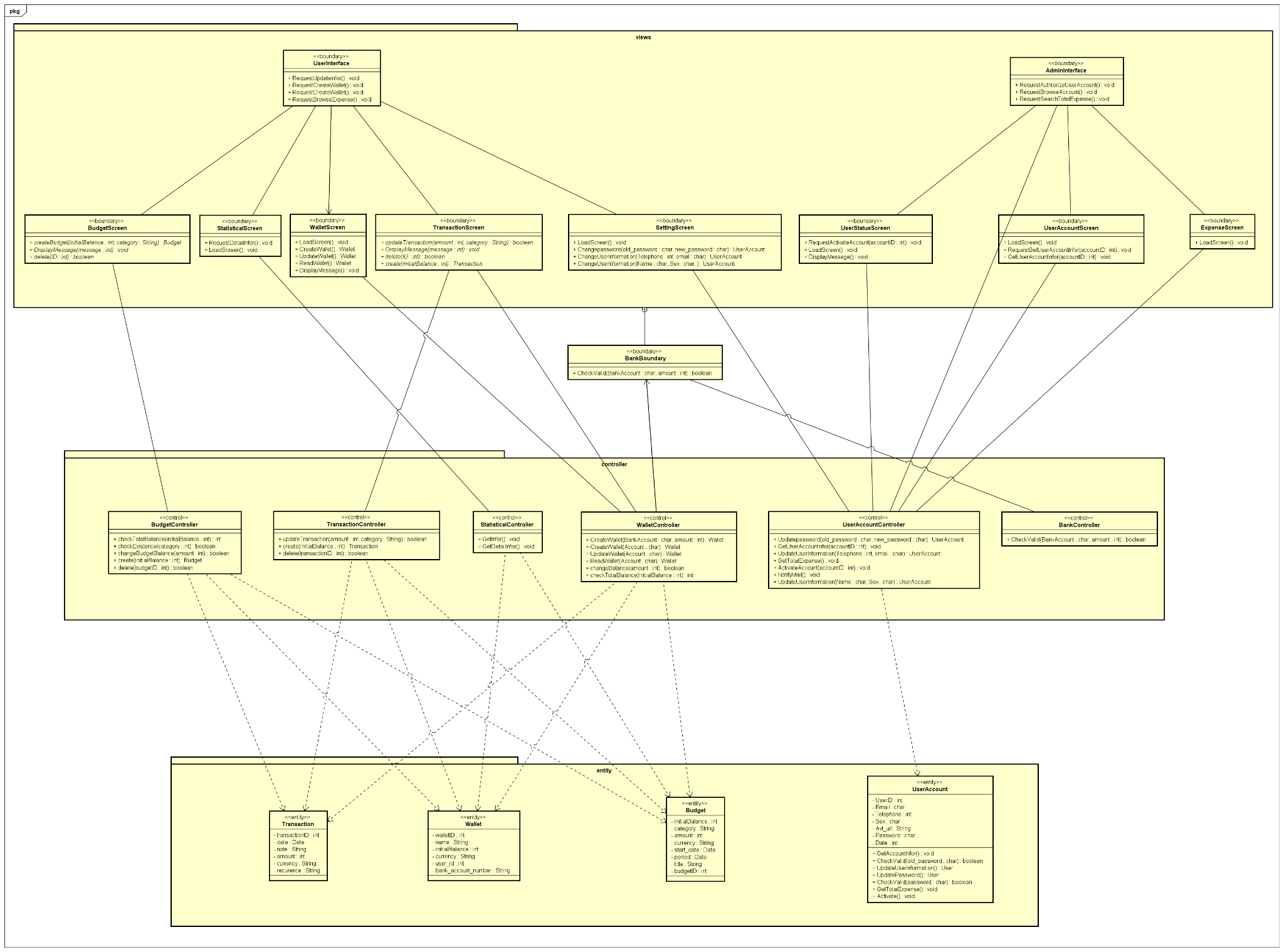


#### Sequence diagram Search total expense



### 

## 3.3 Xây dựng biểu đồ lớp phân tích



## 3.4 Xây dựng biểu đồ thực thể liên kết (ERD)

**Thực thể:**

Transaction

Wallet

Budget

Category

Label

Users

**Quan hệ:**

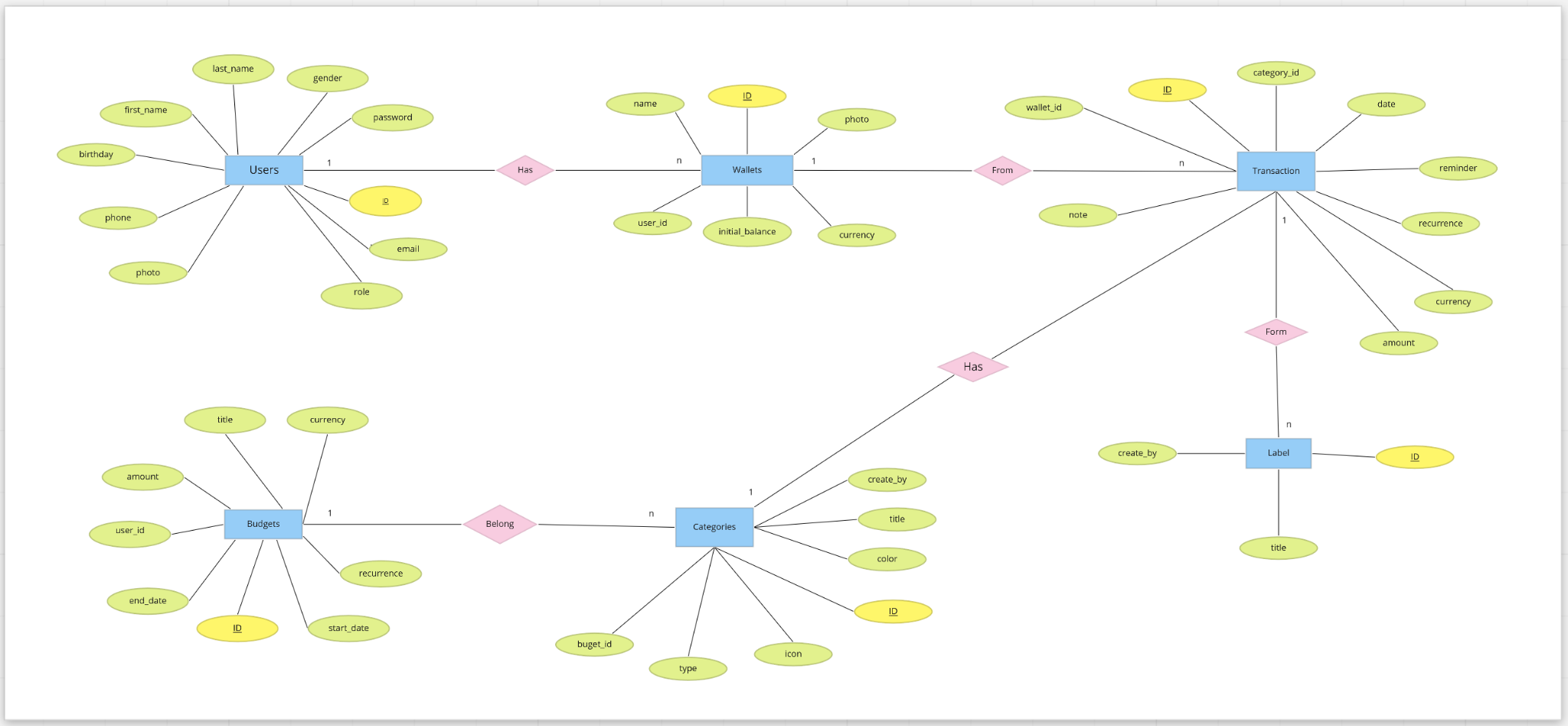
Transaction <from> Wallet

Label <from> Transaction

Budget <belong> to Category

Transaction <has> to Category

Users <has> Wallet

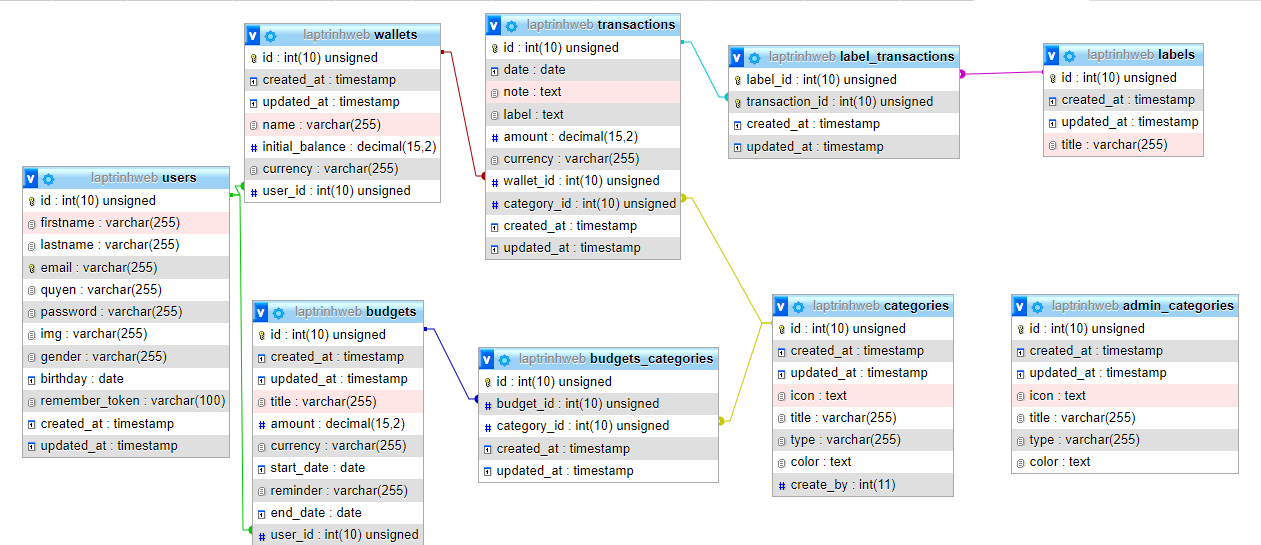


# Chương 4. Thiết kế chương trình

## 4.1 Thiết kế kiến trúc

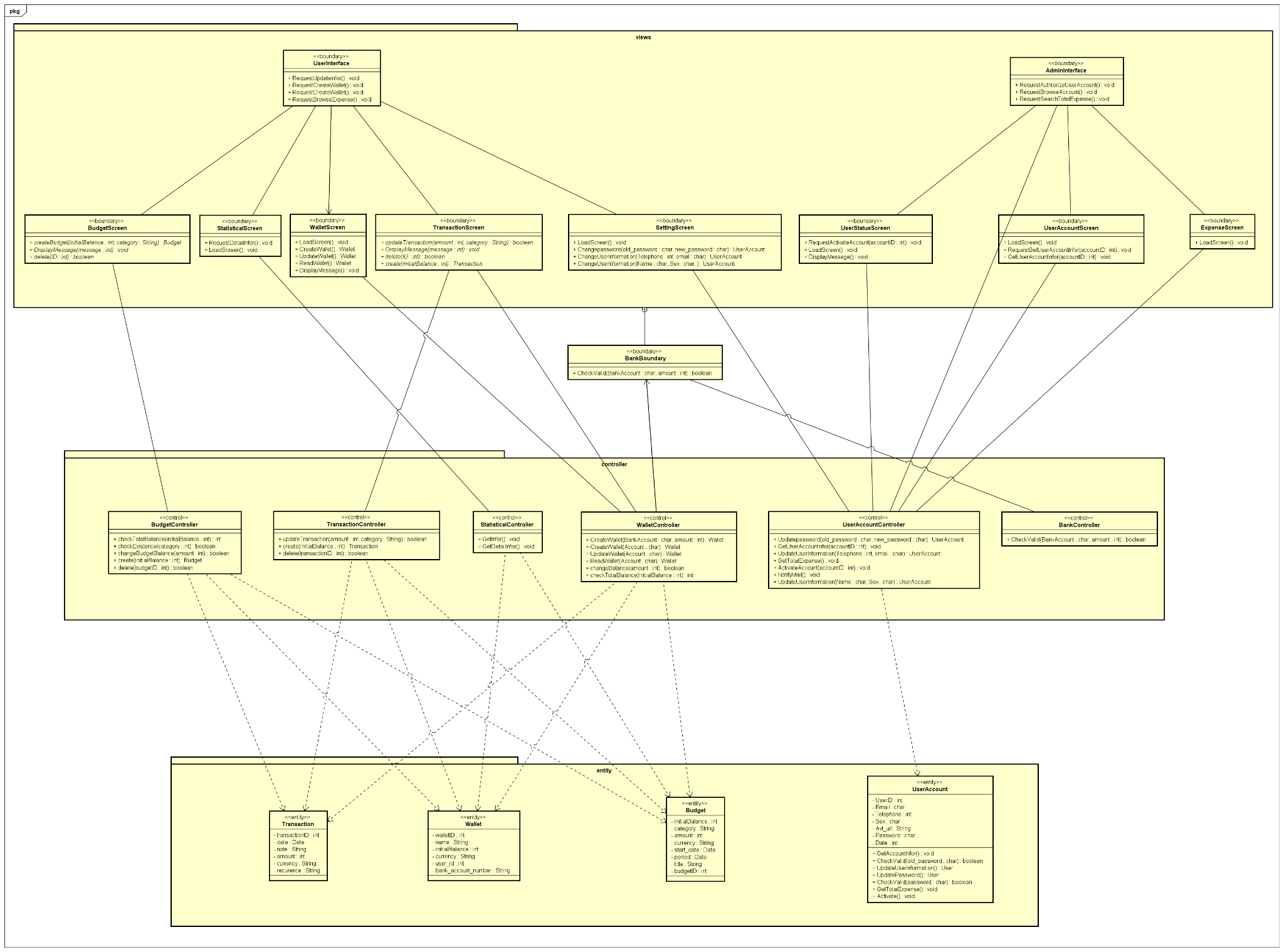
Nhóm tiến hành lựa chọn kiến trúc phần mềm cho ứng dụng Smart Money là kiến trúc 3 lớp MVC. Tiến hành tìm các đối tượng / lớp phân tích cho từng ca sử dụng sau đó thiết kế chuỗi thông điệp giữa chúng để xây dựng các sơ đồ tương tác(Đã được xác định ở trên). Các biểu đồ tuần tự đã được xây dựng để mô hình hóa các hành vi, truyền thông điệp giữa các đối tượng.

## 4.2 Thiết kế cơ sở dữ liệu

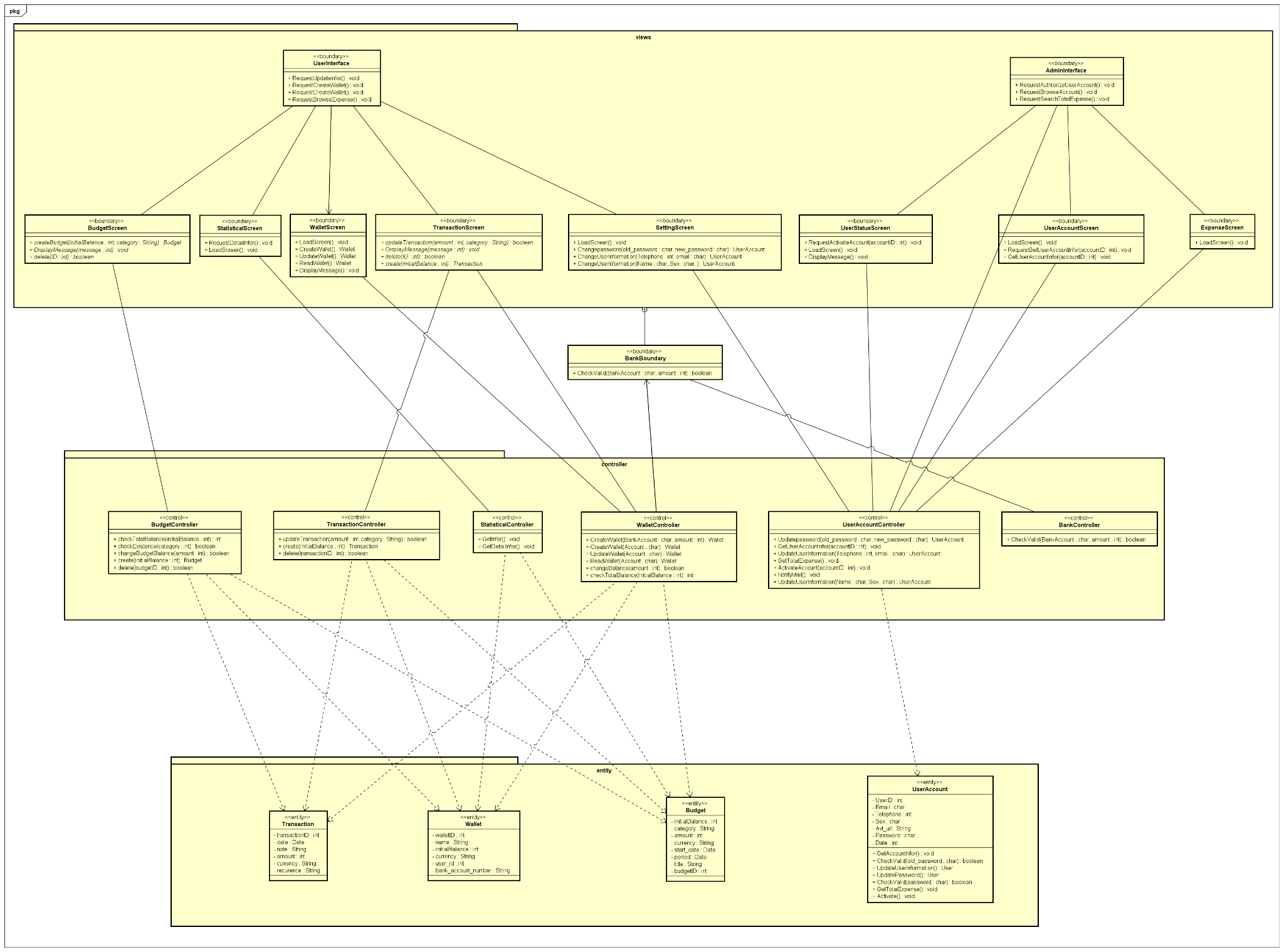


## 4.3 Thiết kế chi tiết các gói

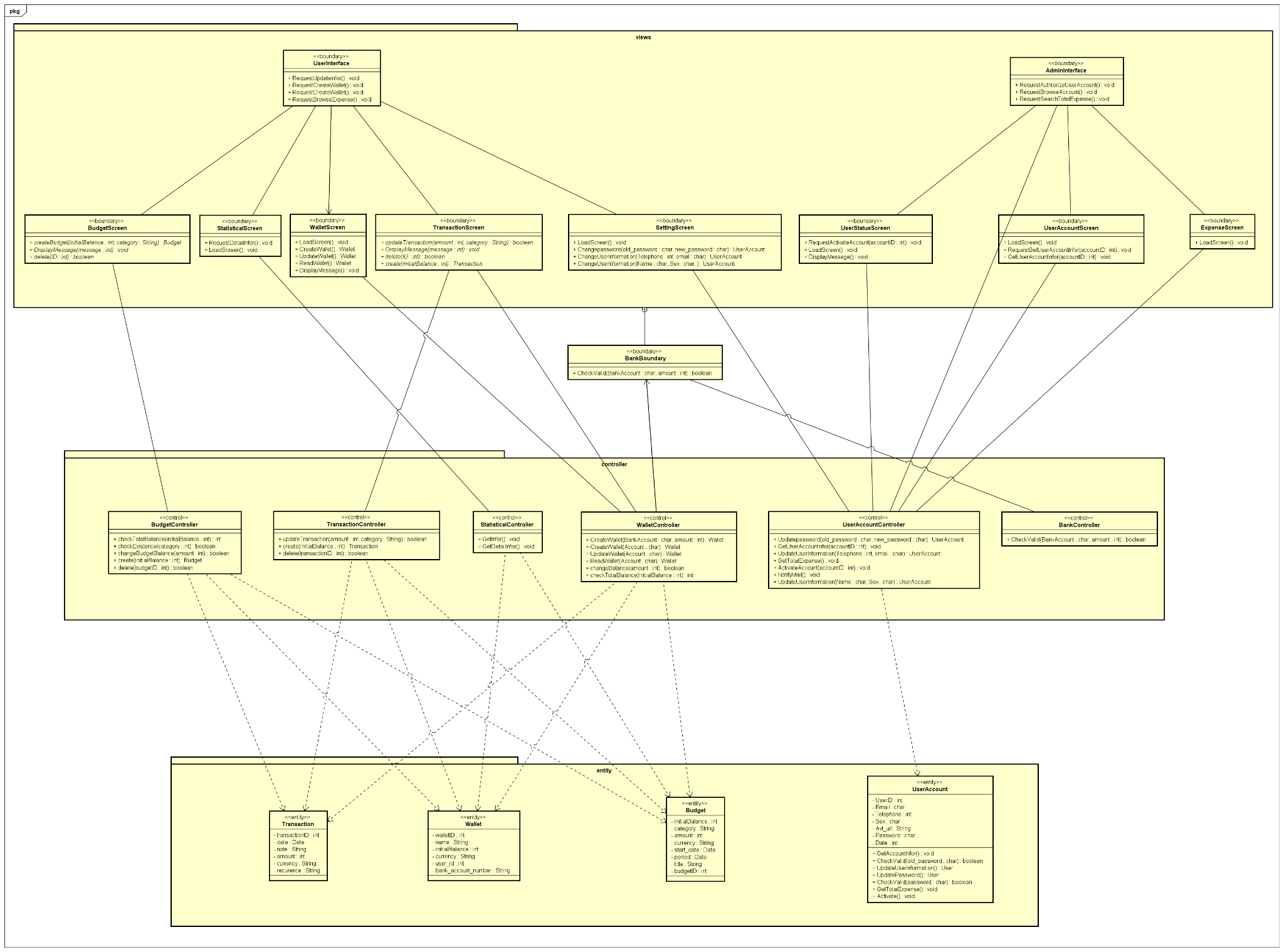
* package liên quan đến hiển thị (các interface)



* package liên quan đến các lớp điều khiển (controller)



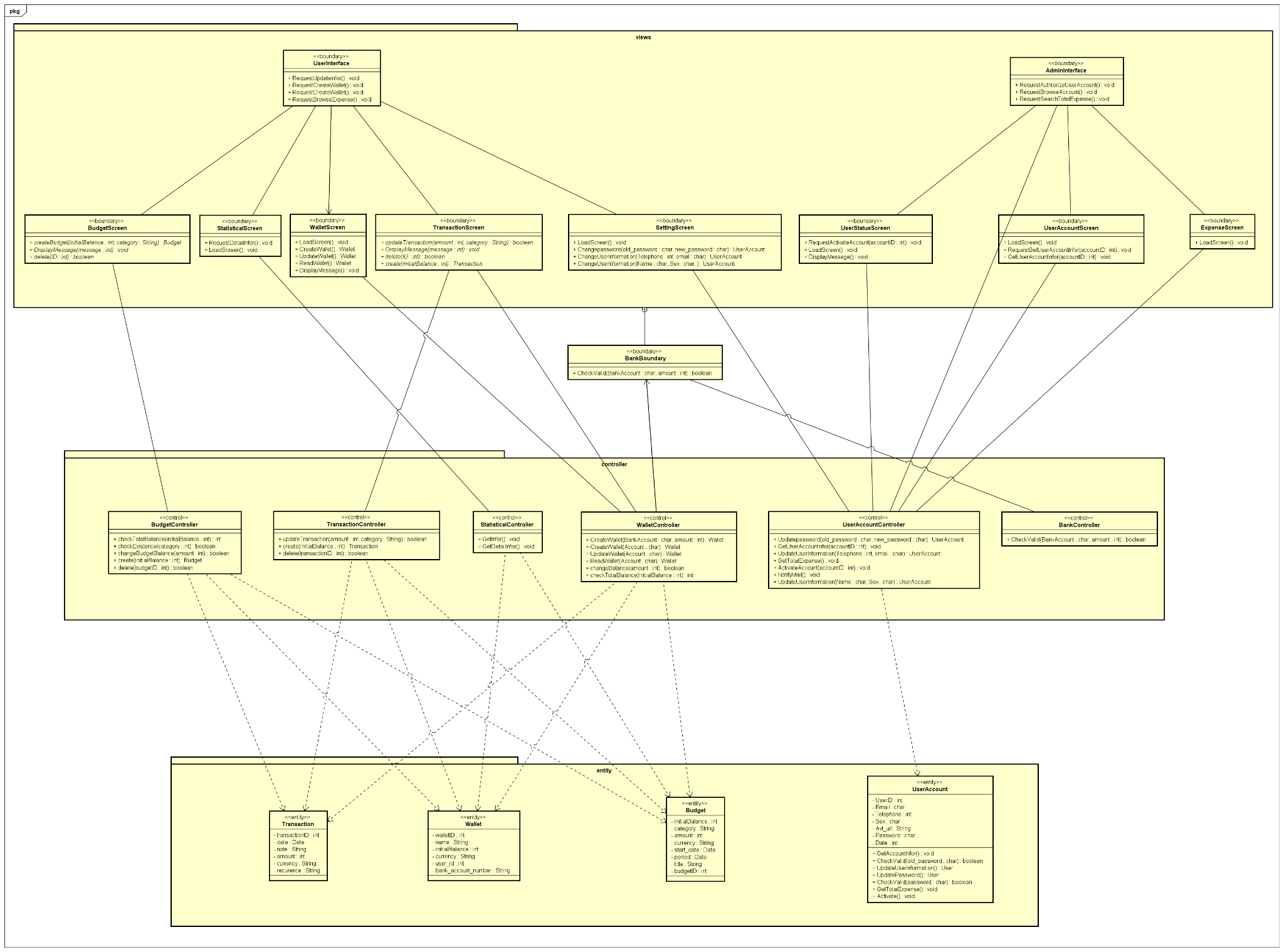
* package liên quan đến các lớp thực thể (entity)



## 4.4 Thiết kế chi tiết các lớp

Thiết kế chi tiết từng lớp được vẽ chi tiết trong sơ đồ lớp dưới đây.

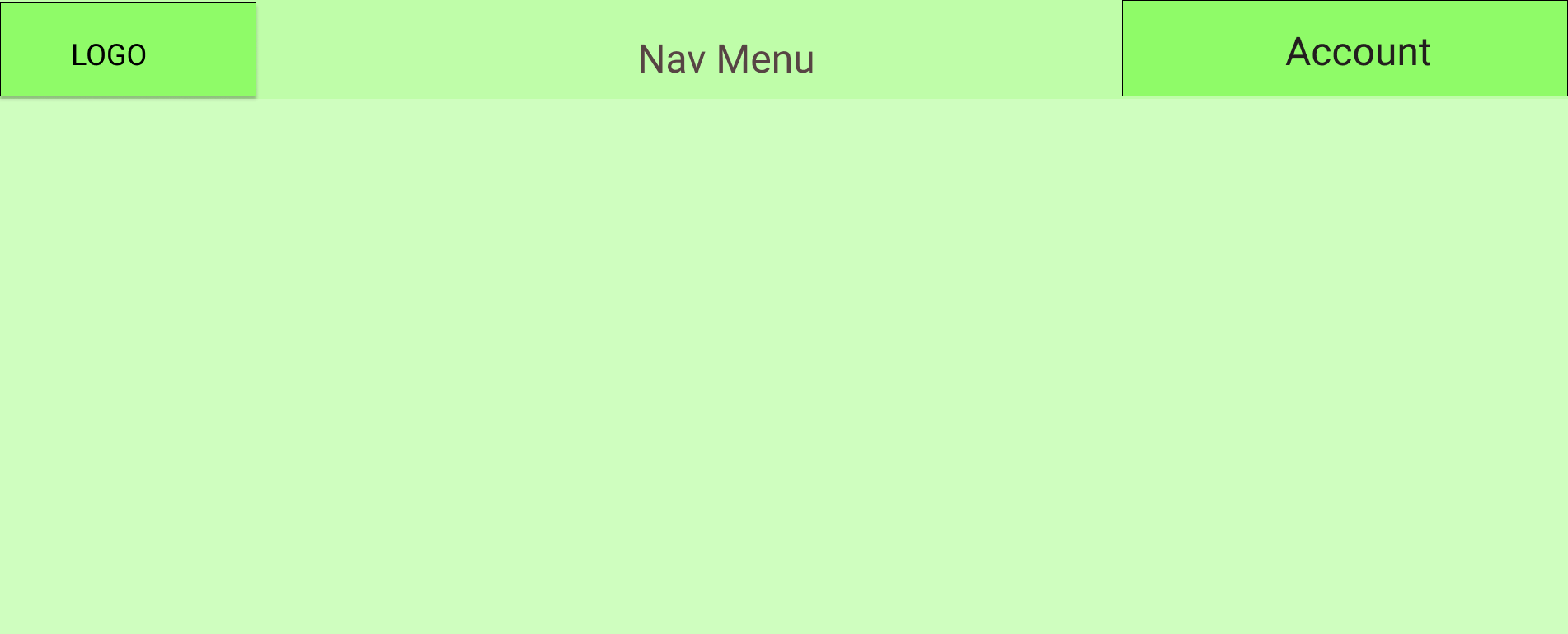
## 4.5 Sơ đồ lớp chi tiết



## 4.6 Thiết kế giao diện

Xây dựng giao diện cơ bản

Chia layout

Header

Sidebar

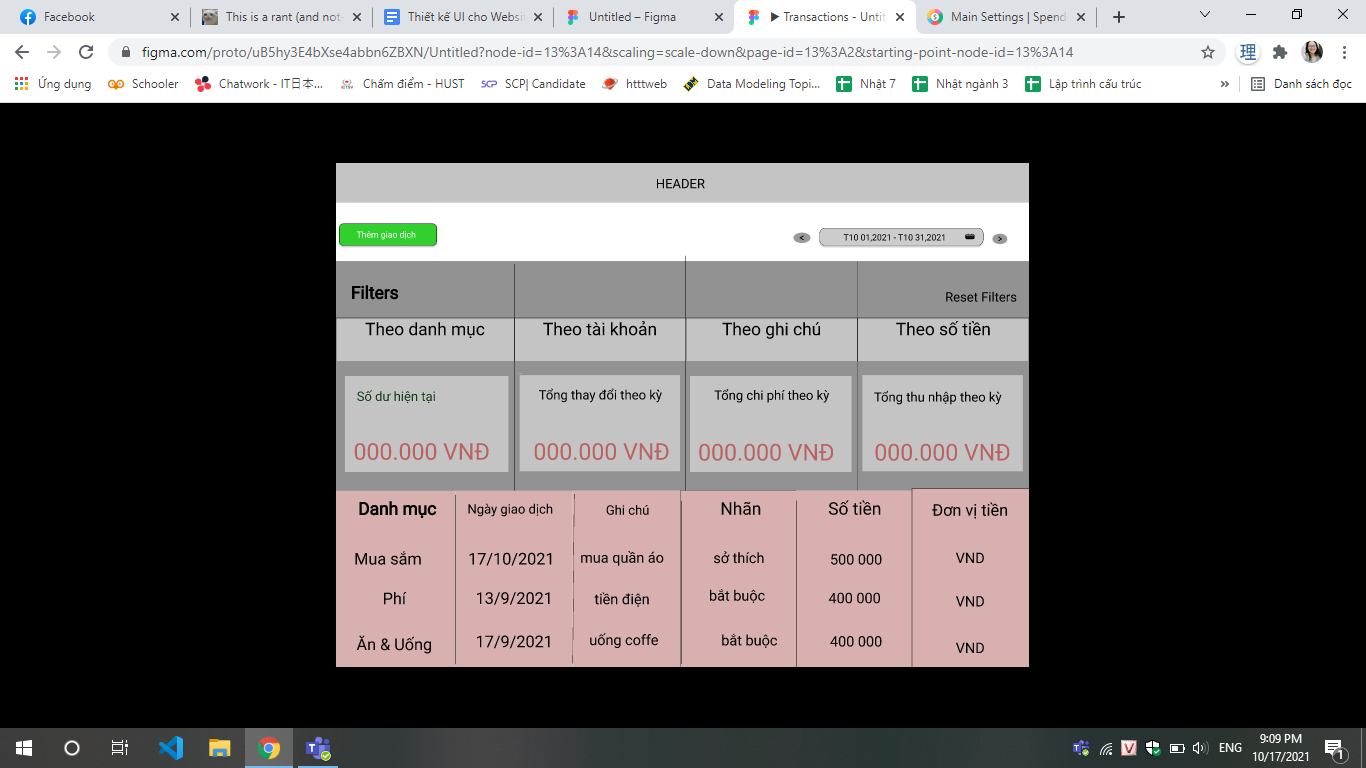


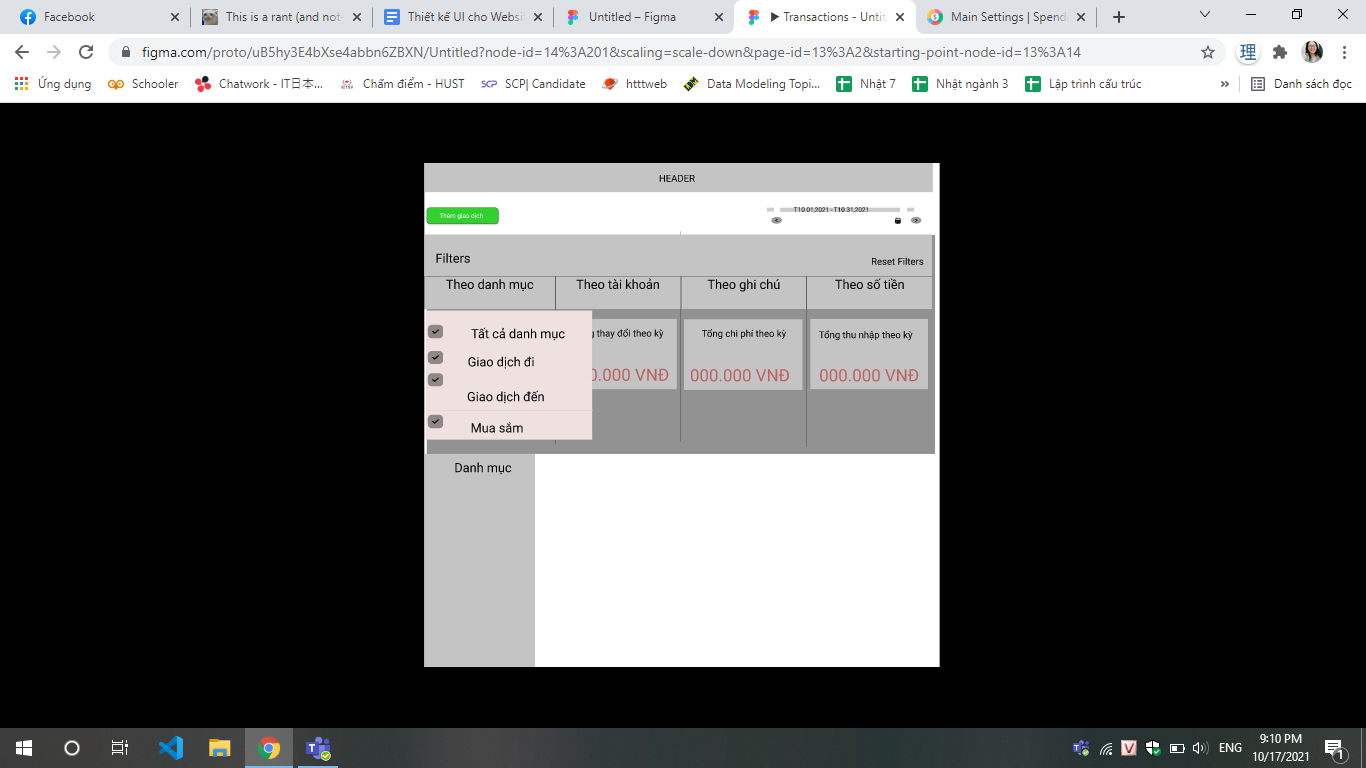
Các Webpage

Dashboard



Transaction





# Chương 5. Xây dựng chương trình minh họa

## 5.1 Thư viện và công cụ sử dụng

Sử dụng Framework Laravel ứng dụng MVC để xây dựng mã nguồn.

Sử dụng Bootstrap 4 để tạo giao diện cho các trang web của Admin.

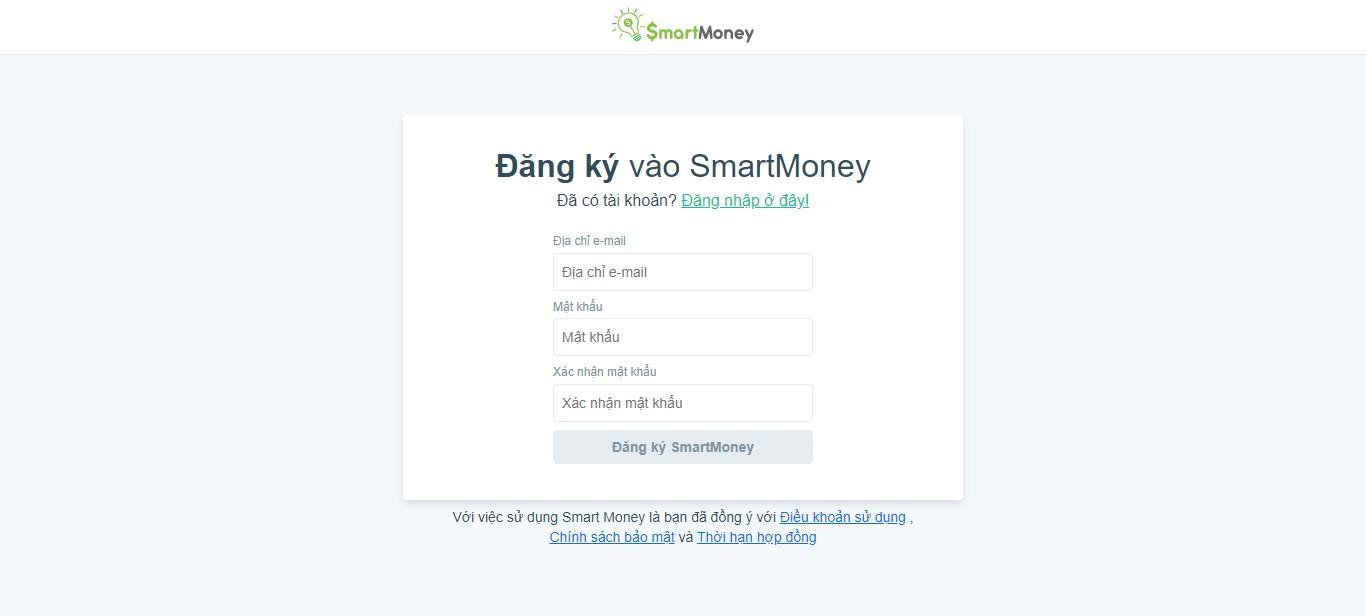
Sử dụng IDE PhpStorm để viết mã nguồn, hệ thống quản trị cơ sở dữ liệu mã nguồn mở:MySQL, Quản lý phiên bản: Git và GitHub

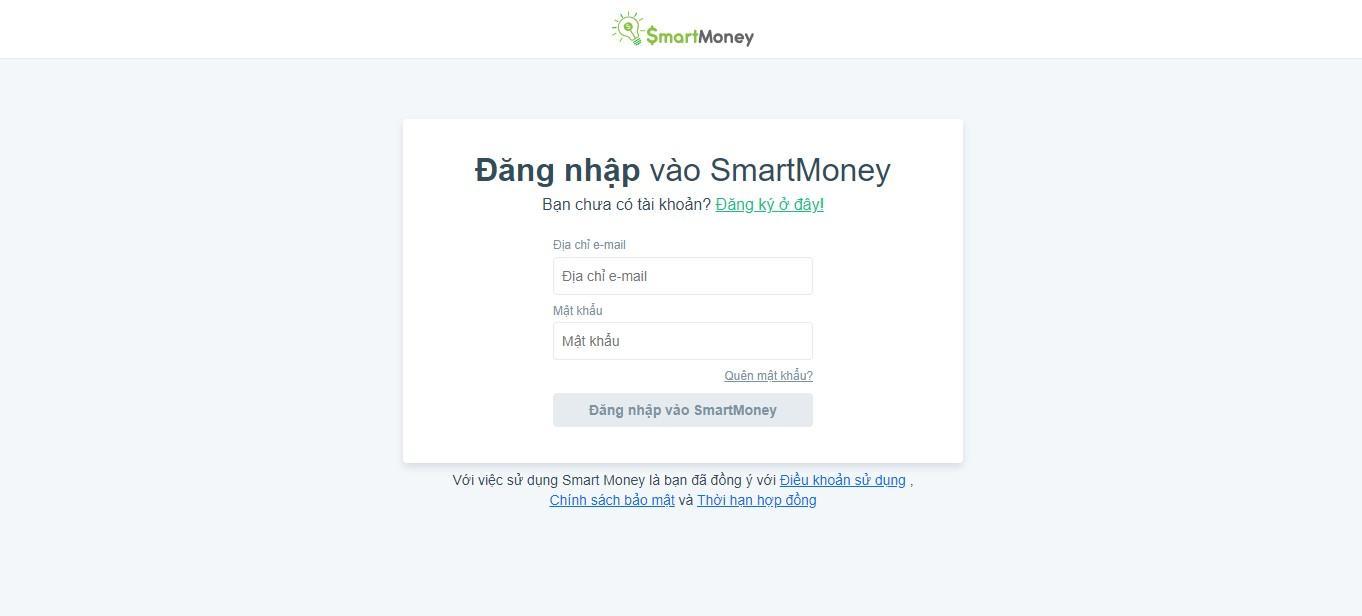
## 5.2 Kết quả chương trình minh họa

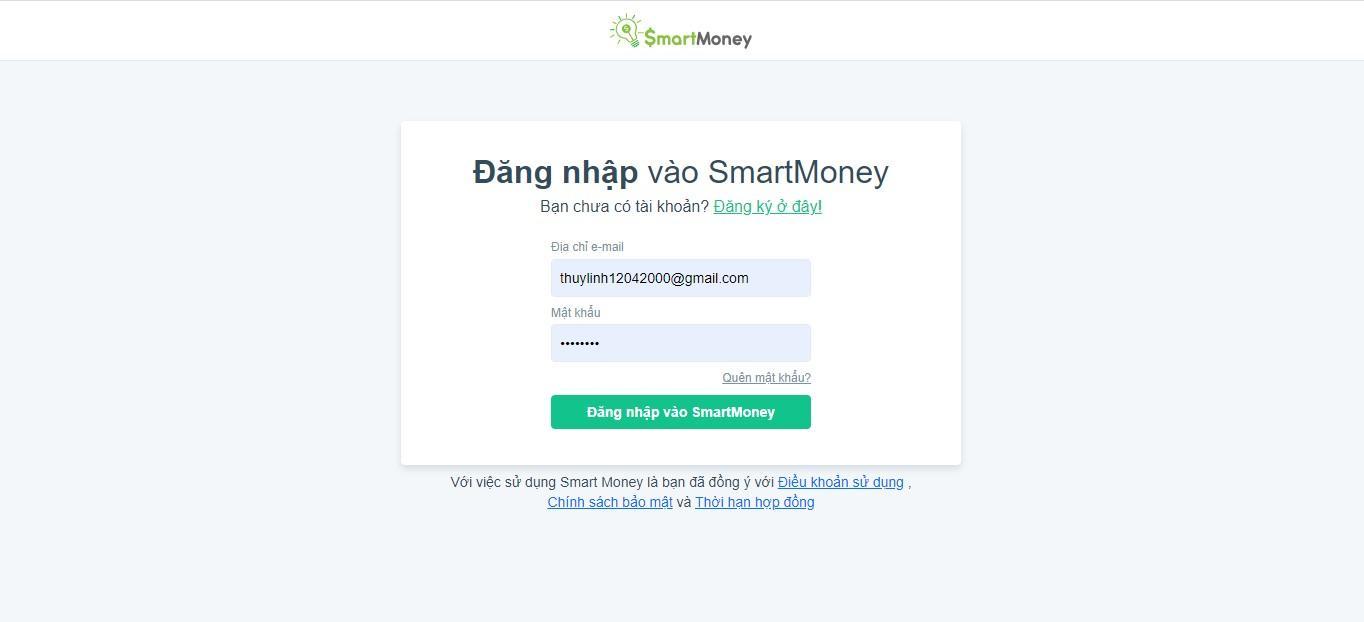
Link sản phẩm: <https://github.com/tuannt2000/SmartMoney>

## 5.3 Giao diện minh họa các chức năng của chương trình

Giao diện chức năng đăng nhập/ đăng ký

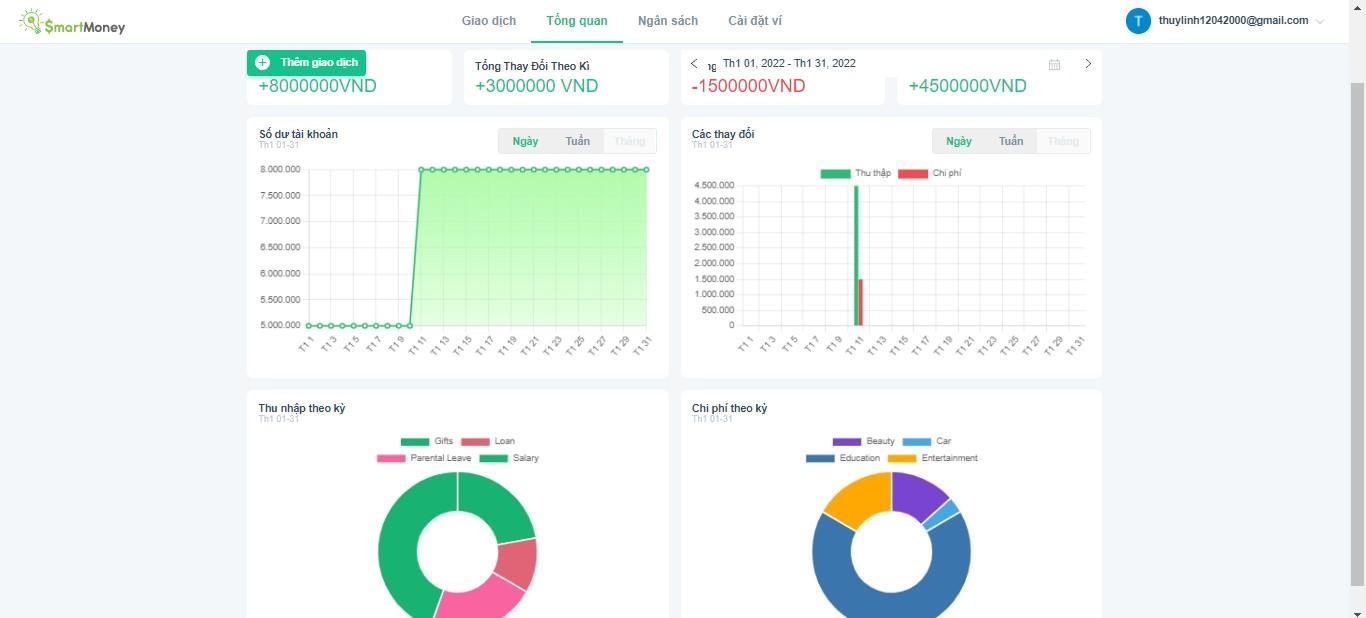




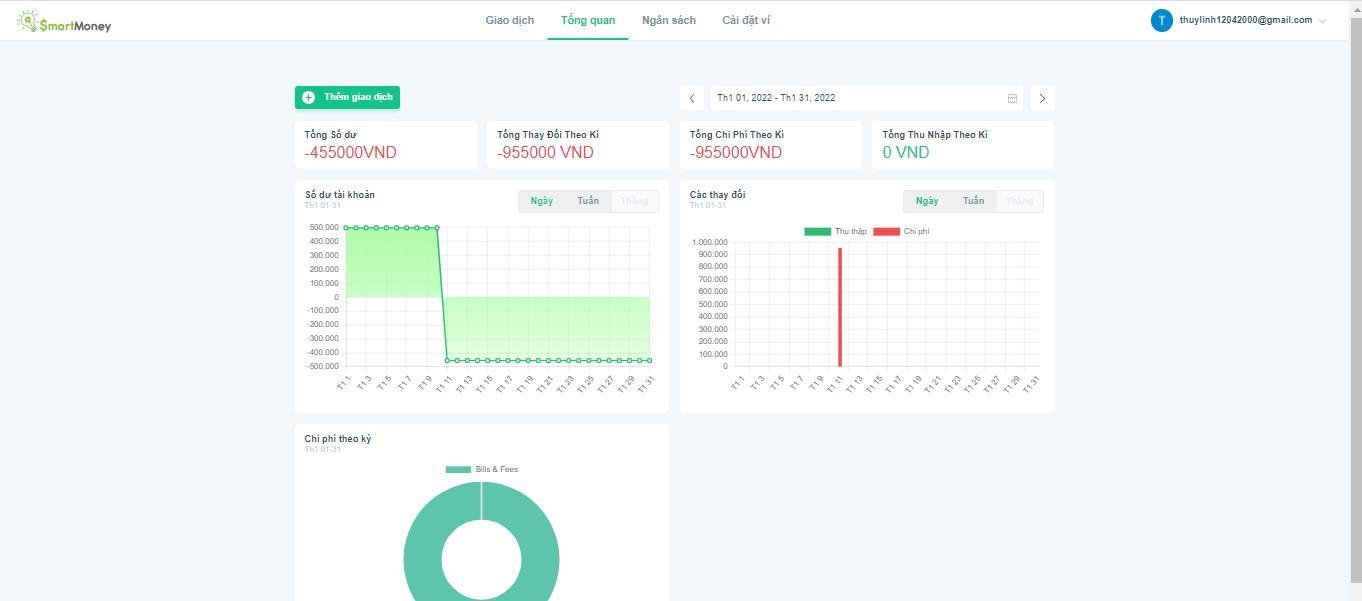


Giao diện chức năng thống kê trong mỗi ví

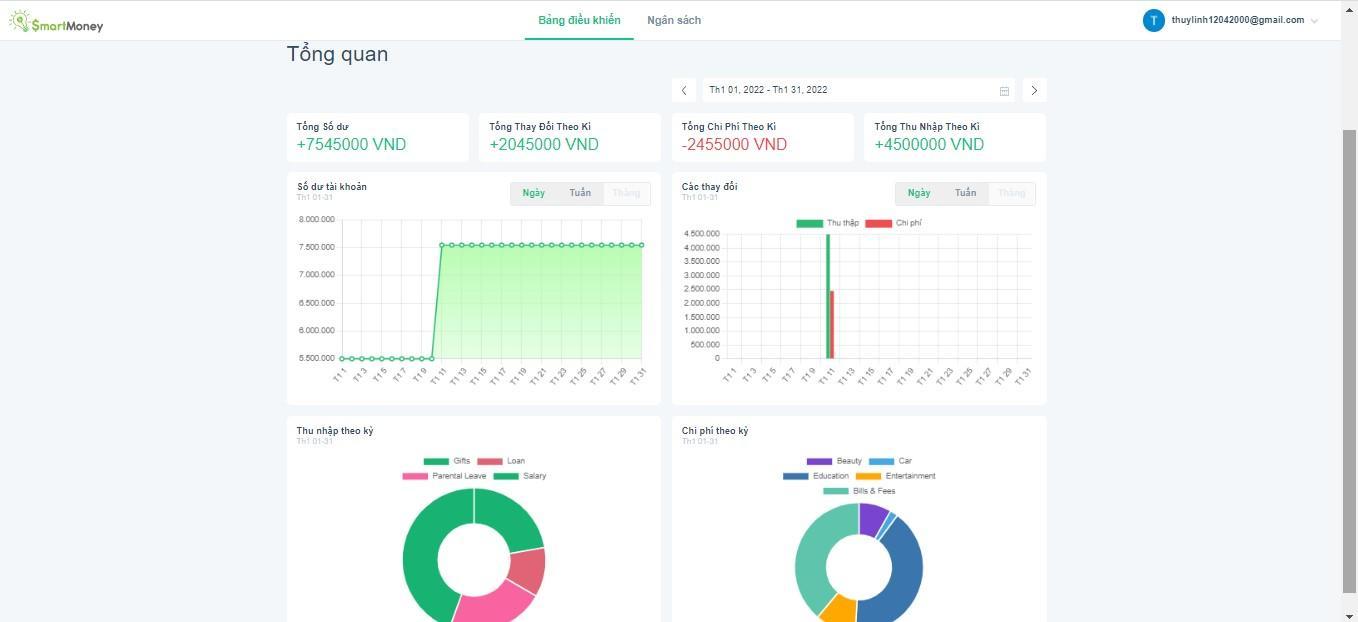
[Thống kê ví 1]



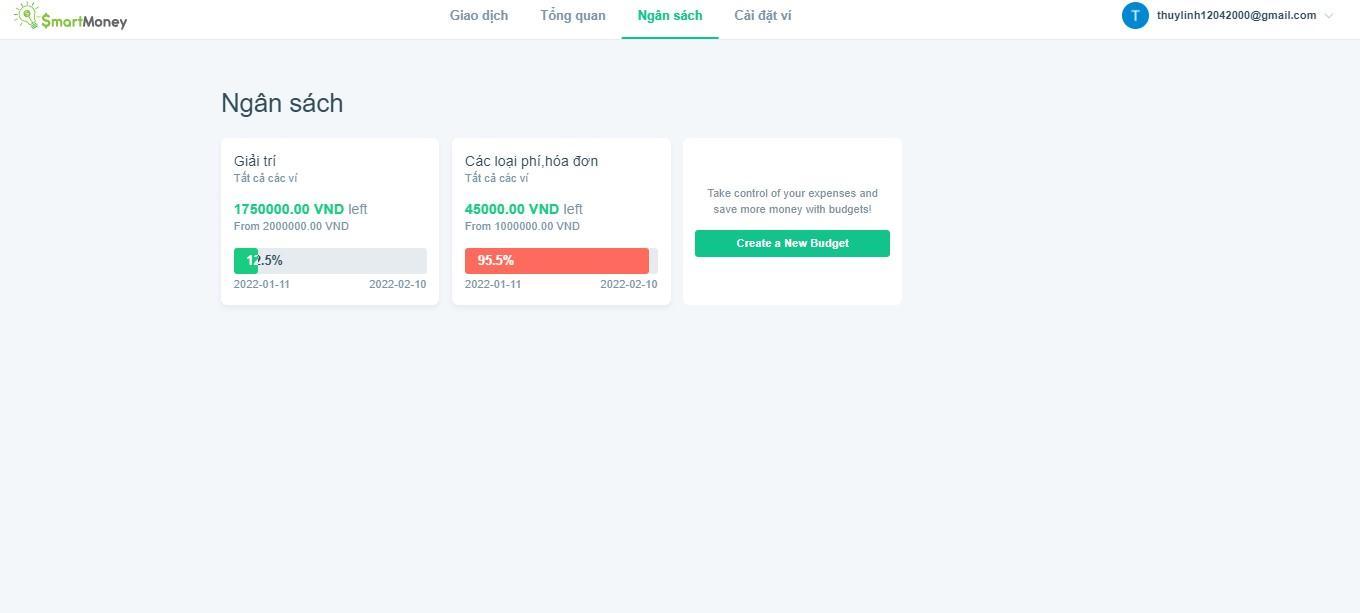
[Thống kê ví 2]



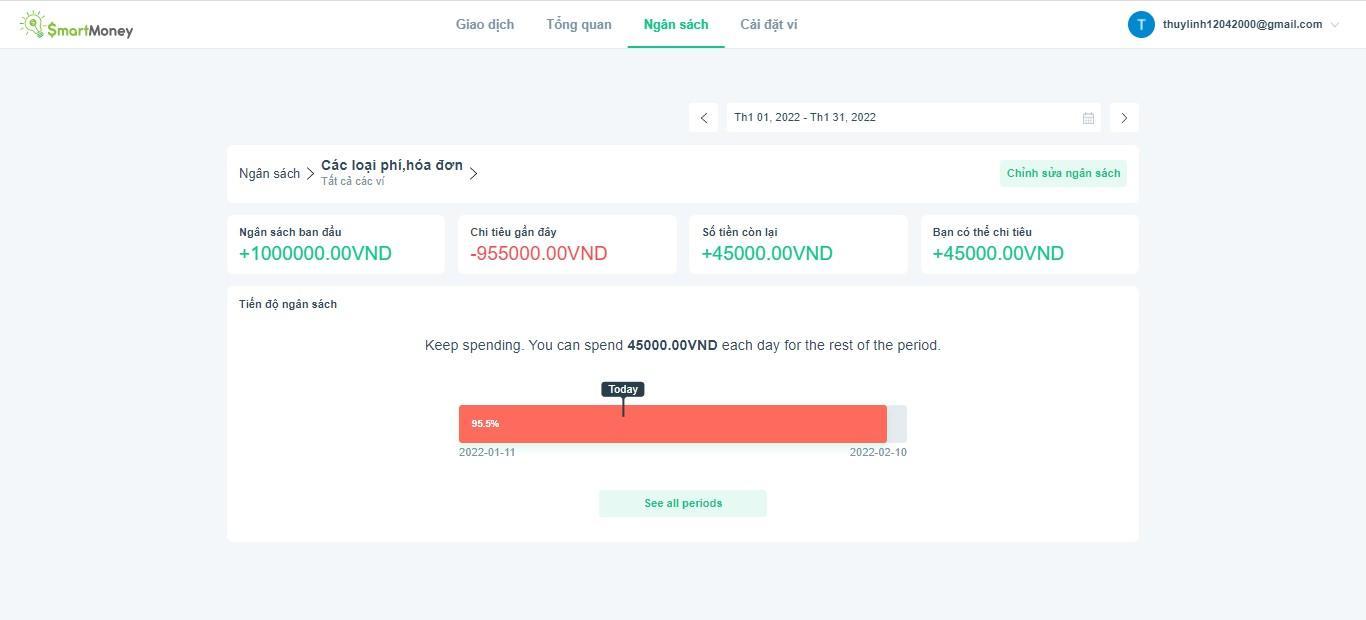
[Thống kê tổng các ví]



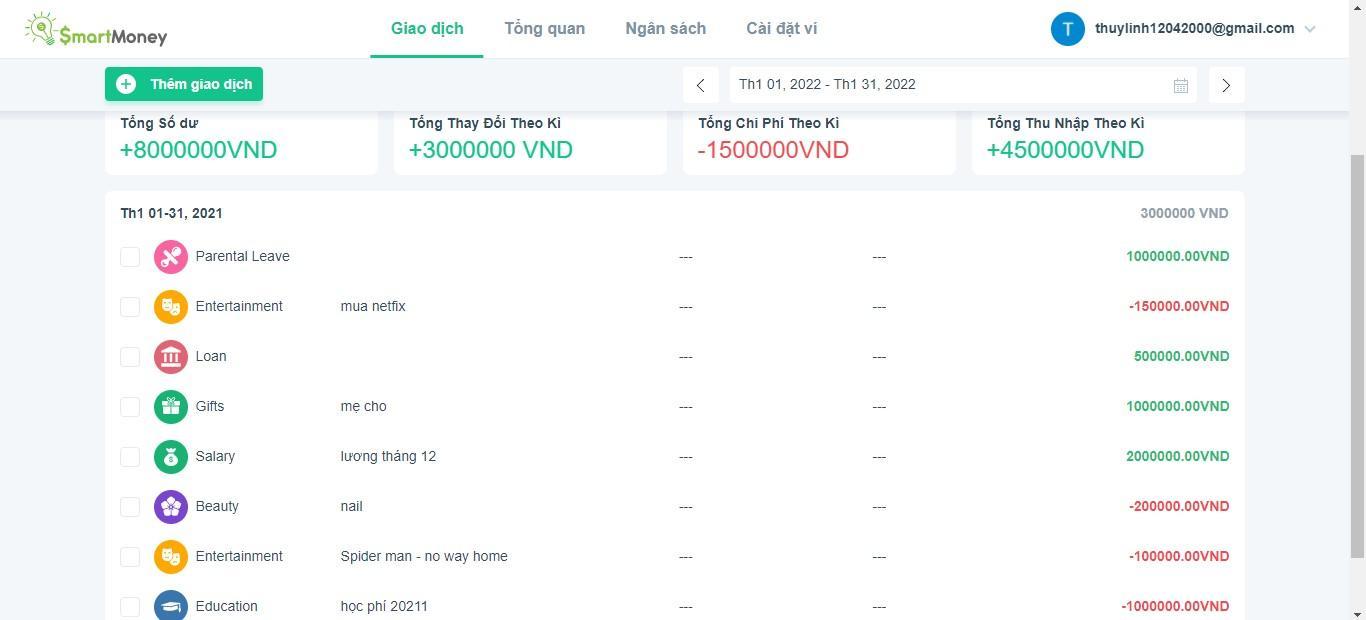
Giao diện chức năng ngân sách



[Chi tiết 1 ngân sách]



Giao diện chức năng giao dịch



# Chương 6. Kiểm thử chương trình

## 6.1 Kiểm thử các chức năng User

### 6.1.1 Kiểm thử cho chức năng đăng ký

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| 1 | Giao diện |  | - Truy cập vào trang đăng ký (Register) của web | - Logo website hiển thị ở trên cùng, chính giữa màn hình  - Title "Đăng ký vào Smart money"  - Trường input địa chỉ email  - Trường input mật khẩu  - Trường input xác nhận mật khẩu  - Button đăng nhập với title "Đăng ký vào Smart money" | Pass |
| 2 | Kiểm tra trường nhập email | Đăng ký email hợp lệ | - Nhập email hợp lệ  (email chưa được đăng ký) | - Thông báo "Đăng ký thành công" | Pass |
| Đăng ký email không hợp lệ (email đã được đăng ký) | - Nhập email | - Hiển thị tin nhắn "Email đã tồn tại" | Pass |
| Đăng ký với trạng thái email để trống | - Đăng ký với trạng thái email để trống | - Button đăng nhập không click được | Pass |
| Đăng ký với email sai định dạng  . | - Nhập vào email sai định dạng  VD: abc@, a@b.c, 12121@gmail.... | - Hiển thị tin nhắn "Email hoặc password không chính xác" | Pass |
| 3 | Kiểm tra trường nhập mật khẩu và xác nhận mật khẩu | Check độ dài password 6-12 | - Check các password có độ dài (5, 6, 8, 12, 13) - Sử dụng phương pháp biên | - Nếu password trong khoảng 6-12: thông báo "Đăng ký thành công"  - Nếu password ngoài khoảng 6-12: thông báo "Password có độ dài 6 đến 12 ký tự" | Pass |
| Check trùng nhau của 2 trường nhập mật khẩu và nhập lại mật khẩu | - Đăng ký thành công với password hợp lệ và trường nhập mật khẩu và nhập lại mật khẩu trùng nhau  - Đăng ký thành công với password hợp lệ và trường nhập mật khẩu và nhập lại mật khẩu không trùng nhau | - Nếu 2 trường trùng nhau: hiển thị thông báo "Đăng ký thành công"  - Nếu 2 trường khác nhau: hiển thị thông báo "Mật khẩu xác nhận không đúng" | Pass |
| Đăng ký với trạng thái password để trống | - Đăng nhập với trạng thái password để trống | - Button đăng nhập không click được | Pass |
| Đăng ký với trạng thái ô xác nhận mật khẩu để trống | - Đăng ký với trạng thái ô xác nhận mật khẩu để trống | - Hiển thị thông báo "Mật khẩu xác nhận không đúng" | Pass |

### 6.1.2 Kiểm thử cho chức năng đăng nhập

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| 1 | Giao diện |  | - Truy cập vào trang login của web | - Logo website hiển thị ở trên cùng, chính giữa màn hình  - Title "Đăng nhập vào Smart money"  - Title "Quên mật khẩu"  - Trường input địa chỉ email  - Trường input password  - Button đăng nhập với title "Đăng nhập vào Smart money" | Pass |
| 2 | Kiểm tra trường input email | Đăng nhập đúng tài khoản | - Đăng nhập thành công với email đã được đăng ký | - Chuyển đến trang home | Pass |
| Đăng nhập sai tài khoản | - Đăng nhập với một email chưa được đăng ký | - Hiển thị tin nhắn "Email hoặc password không chính xác" | Pass |
|  |  | Đăng nhập với trạng thái email để trống | - Đăng nhập với trạng thái email để trống | - Button đăng nhập không click được | Pass |
|  |  | Đăng ký với email sai định dạng  . | - Nhập vào email sai định dạng  VD: abc@, a@b.c, 12121@gmail.... | - Hiển thị tin nhắn "Email hoặc password không chính xác" | Pass |
| 3 | Kiểm tra trường input password | Đăng nhập đúng tài khoản | - Đăng nhập thành công với password đúng | - Chuyển đến trang home | Pass |
| Đăng nhập sai tài khoản | - Đăng nhập thành công với password sai | - Hiển thị tin nhắn "Email hoặc password không chính xác" | Pass |
|  |  | Đăng nhập với trạng thái password để trống | - Đăng nhập với trạng thái password để trống | - Button đăng nhập không click được | Pass |

### 6.1.3 Kiểm thử cho chức năng CRUD ví

* Xem danh sách ví

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| 1 | Giao diện danh sách ví |  | - Truy cập vào trang dashboard | - Trên màn hình trang dashboard sẽ hiển thị danh sách các ví đang có hiện tại và  + Title "Ví"  + Button "Thêm ví mới" | Pass |

* Tạo ví

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| 1 | Giao diện form tạo mới ví |  | - Tại trang dashboard nhấn vào button "Thêm mới ví" | - Form thêm mới ví hiện ra gồm:  + Title "Tạo mới ví"  + Title "Tên ví"  + Trường input tên ví  + Title đơn vị tiền  + Trường input với giá trị mặc định VND  + Title "Đang bắt đầu số dư (tuỳ chọn)"  + Trường input number giá trị mặc định là 0  + Button "Tạo mới" | Pass |
| 2 | Kiểm tra form tạo mới ví | Nhập đủ các trường cần thiết | - Nhập đủ các trường cần thiết  + Tên ví  + Số tiền trong ví | - Trong danh sách ví sẽ xuất hiện thông tin ví mới được tạo gồm:  + Icon  + Tên ví  + Số tiền hiện tại | Pass |
| Bỏ trống thông tin tên ví khi tạo | - Tạo ví với trường tên ví bị bỏ trống | - Button "Tạo mới" không click được | Pass |
| Không nhập trường số dư hiện tại | - Tạo một ví chỉ nhập tên ví không nhập số dư hiện tại | - Trong danh sách ví sẽ xuất hiện thông tin ví mới được tạo gồm:  + Icon  + Tên ví  + Số tiền hiện tại bằng 0 | Pass |

* Sửa ví

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện update |  | - Vào cài đặt của ví | - Trang thông tin của ví gồm  + Tittle "Tên ví"  + Trường input tên ví  + Title đơn vị tiền  + Trường input với giá trị mặc định VND  + Title "Khởi tạo số dư"  + Button "Cập nhập cài đặt" | Pass |
| **2** | Update thông tin ví |  | - Sửa nội dung của một trường bất kỳ của thông tin vi | - Thông báo message với nội dung "Sửa thành công"  - Thông tin ví ở trang dashboard cũng được cập nhật | Pass |

* Xoá ví

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện |  | - Vào cài đặt của ví | - Trang thông tin của ví gồm  + Button "Xóa ví" | Pass |
| **2** | Xoá ví |  | - Nhấn button xoá ví | - Thông báo popup cảnh báo chứa hai button:  + Xoá ví  + Huỷ  - Khi chọn huỷ ví sẽ không được xoá  - Khi chọn xoá ví, ví sẽ được xoá và người dùng sẽ được chuyển hướng đến trang dashboard | Pass |

### 6.1.3 Kiểm thử cho CRUD ngân sách

* Xem ngân sách

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện danh sách ngân sách |  | - Truy cập vào trang dashboard chọn mục ngân sách | - Trên màn hình sẽ hiển thị danh sách các ngân sách đang có hiện tại gồm:  + Title "Ngân sách"  + Button "Create a new budget" | Pass |

* Thêm ngân sách

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện form tạo mới ngân sách |  | - Tại trang ngân sách nhấn vào button "Create a new budget" | - Form thêm mới ví hiện ra gồm:  + Title "Add New Budget"  + Title "Tên ngân sách"  + Trường input tên ngân sách  + Title đơn vị tiền  + Trường input với giá trị mặc định VND  + Title "số lượng"  + Trường input number  + List dropdown các danh mục hiện có  + Button "Tạo mới ngân sách" | Pass |
| **2** | Kiểm tra form tạo mới ngân sách | Nhập đủ các trường cần thiết | - Nhập đủ các trường cần thiết  + Tên ngân sách  + Số lượng  + Chọn các danh mục | - Trong danh sách ngân sách sẽ xuất hiện thông tin danh mục được tạo gồm:  + Tên ngân sách  + Phần trăm đã sử dụng  + Thời gian ngân sách | Pass |
| Bỏ trống trường trong form | - Tạo mới với một trường nào đó bị bỏ trống | - Button "Tạo mới ngân sách" không click được | Pass |

* Sửa ngân sách

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện update |  | - Vào chỉnh sửa ngân sách | - Hiện form thông tin ngân sách gồm  + Title "Sửa ngân sách"  + Trường input Tên Ngân sách  + Title đơn vị tiền  + Trường input với giá trị mặc định VND  + Title "Số lượng"  + List dropdown các danh mục của ngân sách  + Button "Lưu thay đổi"  + Button "Huỷ" | Pass |
| **2** | Update thông tin ngân sách | - Sửa nội dung và lưu thay đổi | - Sửa nội dung của một trường bất kỳ và ấn lưu thay đổi | - Thông báo message với nội dung "Sửa thành công"  - Thông tin ngân sách được cập nhật lại | Pass |
| - Sửa nội dung nhưng không lưu thay đổi | - Sửa nội dung của một trường bất kỳ ấn huỷ | - Thông tin ngân sách được giữ nguyên | Pass |

* Xoá ngân sách

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện |  | - Vào chỉnh sửa ngân sách | - Form thông tin của ngân sách có  + Button "Xoá ngân sách" | Pass |
| **2** | Xoá ngân sách |  | - Nhấn button xoá ngân sách | - Ngân sách sẽ được xoá và người dùng sẽ được chuyển hướng đến trang ngân sách | Pass |

### 6.1.4 Kiểm thử cho chức năng CRUD giao dịch

* Xem giao dịch

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện danh sách giao dịch tại trang dashboard |  | - Truy cập vào một ví tại mục giao dịch | - Trên màn hình sẽ hiển thị danh sách các giao dịch của tất cả các ví hiện tại gồm:  + Thời gian  + Icon danh mục  + Tên danh mục  + Ghi chú  + Số tiền | Pass |

* Thêm giao dịch

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện tạo mới giao dịch |  | - Truy cập vào một ví tại mục giao dịch | - Tại trang giao dịch của ví có:  + Button "Thêm mới giao dịch"  - Khi nhấn vào sẽ có form tạo mới gồm title và trường nhập thông tin tương ứng:  + List dropdown danh mục  + Ngày  + Ghi chú  + Số lượng  + Đơn vị tiền (Mặc định VND)  + Button "Thêm mới giao dịch" | Pass |
| **2** | Kiểm tra form tạo mới giao dịch | Nhập đủ các trường cần thiết | - Nhập đủ các trường cần thiết  + Danh mục  + Ngày  + Ghi chú (tuỳ chọn có thể không nhập)  + Số lượng | - Trong danh sách danh mục sẽ xuất hiện thông tin của giao dịch đó | Pass |
| Bỏ trống trường trong form | - Tạo mới với một trường nào đó bị bỏ trống | - Button "Thêm mới giao dịch" không click được | Pass |

* Sửa giao dịch

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện update |  | - Click vào giao dịch muốn chỉnh sửa trong danh sách các giao dịch | - Hiện form thông tin ngân sách gồm  + List dropdown danh mục  + Ngày  + Ghi chú  + Số lượng  + Đơn vị tiền (Mặc định VND)  + Button "Lưu thay đổi"  + Button "Huỷ"  + Button "Xoá giao dịch" | Pass |
| **2** | Update thông tin giao dịch | - Sửa nội dung và lưu thay đổi | - Sửa nội dung của một trường bất kỳ và ấn lưu thay đổi | - Thông tin giao dịch sẽ được được cập nhật lại | Pass |
| - Sửa nội dung nhưng không lưu thay đổi | - Sửa nội dung của một trường bất kỳ ấn huỷ | - Thông tin ngân sách được giữ nguyên | Pass |

* Xoá giao dịch

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện |  | - Click vào giao dịch muốn chỉnh sửa trong danh sách các giao dịch | - Hiện form thông tin ngân sách gồm  + List dropdown danh mục  + Ngày  + Ghi chú  + Số lượng  + Đơn vị tiền (Mặc định VND)  + Button "Lưu thay đổi"  + Button "Huỷ"  + Button "Xoá giao dịch" | Pass |
| **2** | Xoá giao dịch |  | - Nhấn button xoá giao dịch | - Giao dịch sẽ được xoá khỏi danh sách các giao dịch | Pass |

### 6.1.5 Kiểm thử cho chức năng CRUD danh mục

* Xem danh mục

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện danh sách danh mục |  | - Truy cập vào trang cài đặt tài khoản chọn mục tất cả các danh mục | - Trên màn hình sẽ hiển thị 2 danh sách các danh mục thu và chi đang có hiện tại | Pass |

* Thêm danh mục

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện form tạo mới danh mục |  | - Truy cập vào trang trang tất cả các danh mục | - Title "Thêm một mới danh mục"  - Một form để thêm mới gồm:  + List dropdown chọn icon  + List dropdown chọn màu  + Trường nhập tên  + List dropdown chọn loại danh mục (chi phí/thu nhập)  + Button "tạo mới" | Pass |
| **2** | Kiểm tra form tạo mới danh mục | Nhập đủ các trường cần thiết | - Nhập đủ các trường cần thiết  + Icon  + Màu  + Tên | - Trong danh sách danh mục sẽ xuất hiện thông tin danh mục mới được tạo | Pass |
| Bỏ trống trường trong form | - Tạo mới với một trường nào đó bị bỏ trống | - Button "Tạo mới" không click được | Pass |

* Chỉnh sửa danh mục

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện update |  | - Click vào button chỉnh sửa của danh mục muốn chỉnh sửa | - Hiện form thông tin danh mục gồm  + List dropdown chọn icon  + List dropdown chọn màu  + Trường nhập tên  + Button "Sửa danh mục" | Pass |
| **2** | Update thông tin danh mục | - Sửa nội dung và lưu thay đổi | - Sửa nội dung của một trường bất kỳ và ấn lưu thay đổi | - Thông báo message với nội dung "Sửa thành công"  - Thông tin danh mục được cập nhật lại | Pass |
| - Sửa nội dung nhưng không lưu thay đổi | - Sửa nội dung của một trường bất kỳ ấn huỷ | - Thông tin danh mục được giữ nguyên | Pass |

* Xoá danh mục

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện |  | - Click vào button xoá của danh mục muốn xoá | - Popup cảnh báo gồm message và  + Button "Xoá ngân sách"  + Button "Huỷ" | Pass |
| **2** | Xoá danh mục | - Nhấn button xoá danh mục | - Nhấn button xoá danh mục | - Danh mục sẽ được xoá, danh sách các danh mục sẽ được cập nhật lại | Pass |

### 6.1.6 Kiểm thử cho chức năng thống kê

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện thống kê dạng chữ | - Các mục thống kê các ví | - Truy cập vào trang dashboard | - Trên màn hình sẽ hiển thị 4 mục thống kê :  + Tổng số dư và số tiền  + Tổng số dư thay đổi theo kỳ và số tiền  + Tổng Chi Phí Theo kì và số tiền  + Tổng Thu Nhập Theo kì và số tiền | Pass |
| - Các mục thống kê riêng của một ví | - Chọn ví muốn xem và chọn mục tổng quan | - Trên màn hình sẽ hiển thị 4 mục thống kê :  + Tổng số dư và số tiền  + Tổng số dư thay đổi theo kỳ và số tiền  + Tổng Chi Phí Theo Kì và số tiền  + Tổng Thu Nhập Theo Kì và số tiền | Pass |
| **2** | Giao diện thống kê dạng biểu đồ | - Các mục thống kê các ví | - Truy cập vào trang dashboard | - Trên màn hình hiển thị hiển thị 4 biểu đồ gồm:  + Biểu đồ số dư (biểu đồ đường)  + Biểu đồ các thay đổi của thu và chi (biểu đồ cột)  + Biểu đồ thu nhập (dạng tròn)  + Biểu đồ chi (dạng tròn) | Pass |
| - Các mục thống kê riêng của một ví | - Chọn ví muốn xem và chọn mục tổng quan | - Trên màn hình hiển thị hiển thị 4 biểu đồ gồm:  + Biểu đồ số dư (biểu đồ đường)  + Biểu đồ các thay đổi của thu và chi (biểu đồ cột)  + Biểu đồ thu nhập (dạng tròn)  + Biểu đồ chi (dạng tròn) | Pass |

### 6.1.7 Kiểm thử cho chức năng update thông tin

* Chức năng update thông tin

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện thông tin tài khoản |  | - Truy cập vào cài đặt tài khoản chọn mục tài khoản | - Trên màn hình sẽ hiển thị thông tin tài khoản gồm :  + Avatar  + Button upload avatar  + Trường nhập tên  + Trường nhập họ  + Dropdown chọn giới tính  + Dropdown chọn ngày tháng năm sinh  + Thông tin mail  + Button "Cập nhật cài đặt của tôi" | Pass |
| **2** | Update thông tin tài khoản | - Sửa nội dung và lưu thay đổi | - Sửa nội dung của một trường bất kỳ và ấn lưu thay đổi | - Thông báo message với nội dung "Sửa thành công"  - Thông tin tài khoản được cập nhật lại | Pass |
| - Sửa trường email trùng với một tài khoản khác | - Sửa trường email trùng với một tài khoản khác | - Thông báo message với nội dung "Email đã tồn tại" | Pass |
| - Sửa nội dung nhưng không lưu thay đổi | - Sửa nội dung của một trường bất kỳ ấn huỷ | - Thông tin tài khoản được giữ nguyên | Pass |

* Xoá tài khoản

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện thông tin tài khoản |  | - Truy cập vào cài đặt tài khoản chọn mục tài khoản | - Button "Xoá tài khoản" khi click vào sẽ hiện  pop cảnh báo gồm:  + 5 Checkbox  + Button "Xoá tài khoản"  + Button "Huỷ" | Pass |
| **2** | Thực hiện xoá tài khoản |  | - Click vào xoá tài khoản | - Hiển thị popup cảnh báo xác nhận và 5 checkbox  + Tick vào hết các ô checkbox thì button xoá tài khoản sẽ sáng khi click vào tài khoản sẽ được xoá và người dùng được chuyển đến trang đăng nhập  + Nhấn huỷ thì tài khoản sẽ không được xoá | Pass |

## 6.2 Kiểm thử các chức năng Admin

### 6.2.1 Kiểm thử cho chức năng quản lý tài khoản user

* Xem danh sách tài khoản của user

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện danh sách tài khoản |  | - Truy cập vào trang quản trị của hệ thống và chọn mục user | - Trên màn hình sẽ hiển thị danh sách các tài khoản và thông tin của mỗi tài khoản (không có mật khẩu):  + Name  + Email  + Loại tài khoản  + Ngày tạo  + Button "Xoá" | Pass |

* Xoá tài khoản

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện |  | - Truy cập vào trang quản trị của hệ thống và chọn mục user | - Bên cạnh thông tin tài khoản có button "Xoá" |  |
| **2** | Xoá tài khoản | - Nhấn button xoá ngân sách | - Nhấn button xoá tài khoản | - Tài khoản sẽ được xoá và danh sách các tài khoản sẽ được cập nhật lại |  |

### 6.2.2 Kiểm thử cho chức năng CRUD danh mục của admin

* Xem danh mục của admin

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện danh sách danh mục |  | - Truy cập vào trang quản trị của hệ thống và chọn mục danh mục | - Trên màn hình sẽ hiển thị danh sách các danh mục và thông tin của mỗi danh mục gồm  + STT  + Title  + Type  + Color  + Ngày tạo | Pass |

* Chức năng thêm danh mục của admin

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện form tạo mới danh mục |  | - Click vào button tạo mới | - Title "Thêm một mới danh mục"  - Một form để thêm mới gồm:  + Trường nhập link tới ảnh icon của danh mục  + Trường nhập màu của danh mục  + Trường nhập tên danh mục  + Trường nhập title của danh mục  + Button "Save" | Pass |
| **2** | Kiểm tra form tạo mới danh mục | Nhập đủ các trường cần thiết | - Nhập đủ các trường cần thiết  + Icon  + Màu  + Tên  + Type | - Trong danh sách danh mục sẽ xuất hiện thông tin danh mục mới được tạo  - Khi tạo mới tài khoản những danh mục trong danh sách này sẽ xuất hiện trong tài khoản tạo mới đó | Pass |
| Bỏ trống trường trong form | - Tạo mới với một trường nào đó bị bỏ trống | - Báo lỗi tạo không thành công |  |

* Sửa danh mục

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện update |  | - Click vào button chỉnh sửa của danh mục muốn chỉnh sửa | - Hiện form thông tin danh mục gồm  + Trường nhập link tới ảnh icon của danh mục  + Trường nhập màu của danh mục  + Trường nhập tên danh mục  + Trường nhập title của danh mục  + Button "Save" | Pass |
| **2** | Update thông tin danh mục | - Sửa nội dung và lưu thay đổi | - Sửa nội dung của một trường bất kỳ và ấn lưu thay đổi | - Thông báo message với nội dung "Update thành công"  - Thông tin danh mục được cập nhật lại | Pass |
| Bỏ trống trường trong form | - Tạo mới với một trường nào đó bị bỏ trống | - Báo lỗi sửa không thành công | Pass |
| - Sửa nội dung nhưng không lưu thay đổi | - Sửa nội dung của một trường bất kỳ ấn huỷ | - Thông tin danh mục được giữ nguyên | Pass |

* Xoá danh mục

| **STT** | **Miêu tả test case** | **Sub test case** | **Các bước thực hiện** | **Kết quả mong muốn** | **Trạng thái (Pass/False)** |
| --- | --- | --- | --- | --- | --- |
| **1** | Giao diện |  | - Truy cập vào trang quản trị của hệ thống và chọn mục danh mục | - Button xoá ở cạnh thông tin danh mục | Pass |
| **2** | Xoá danh mục | - Nhấn button xoá | - Nhấn button xoá danh mục | - Danh mục sẽ được xoá, danh sách các danh mục sẽ được cập nhật lại | Pass |

# Chương 7. Hướng dẫn cài đặt và sử dụng

## 7.1 Hướng dẫn cài đặt

Để cài đặt cần thực hiện theo các bước được liệt kê chi tiết ở file readme

Link: <https://github.com/tuannt2000/SmartMoney/blob/main/README.md>

## 7.2 Đối tượng, phạm vi sử dụng

Đối tượng sử dụng:Những người muốn dễ dàng quản lý tài chính của mình và đưa ra các quyết định về vấn đề tài chính một cách thông minh đều có thể sử dụng website mà không giới hạn.

Phạm vi sử dụng: Được sử dụng miễn phí và nhập thông tin các khoản chi tiêu với các giao dịch hợp lý, số tiền giao dịch không vượt quá 15 chữ số.

## 7.3 Xác định các yêu cầu cài đặt

Để có thể build project cần cài Laravel với các yêu cầu bắt buộc như sau:

Composer

cài sẵn apache hoặc nginx, mysql,php , npm(node), git.

## 7.4 Hướng dẫn chi tiết các bước cài đặt

**Bước 1: Mở Terminal và thực hiện clone project và vào thư mục bằng câu lệnh sau:**

git clone https://github.com/tuannt2000/SmartMoney.git

cd SmartMoney

**Bước 2: Chạy composer và npm để cài đặt các gói cần thiết**

composer install

npm install

**Bước 3: Tạo database và config database**

Vào mysql workbech tạo ra database mới

Tạo database có tên là :SmartMoney\_db

Sau đó ta thực hiện lệnh sau để copy ra file env:

cp .env.example .env

Cập nhật file env của bạn như sau:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=SmartMoney\_db

DB\_USERNAME=root

DB\_PASSWORD=

**Bước 4: Tạo ra key cho dự án**

php artisan key:generate

**Bước 5: Tạo ra các bảng và dữ liệu mẫu cho database**

php artisan migrate

php artisan db:seed

## 7.5 Hướng dẫn sử dụng phần mềm

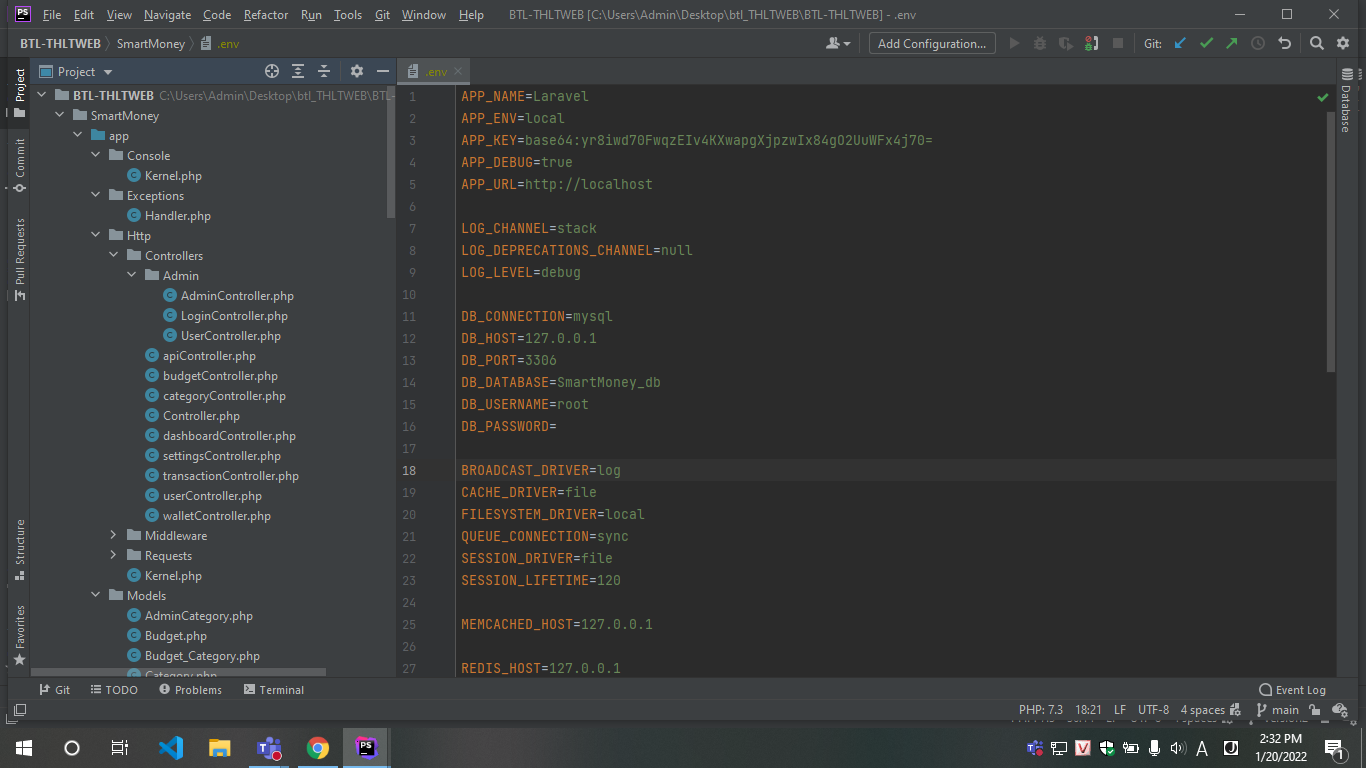
Người dùng truy cập theo đường link: <http://localhost:81/laptrinhweb/public/auth/login> tiến hành đăng ký, đăng nhập và sử dụng các chức năng của website

# Chương 8 : Nguyên lý thiết kế SOLID

**S: Single Responsibility Principle**

Là nguyên lý được sử dụng nhiều nhất vì tính thực tế và hữu ích của nó.Nguyên lý S được áp dụng vào hầu hết các thực thể, các lớp và các phương thức để tương ứng mỗi thành phần trên chỉ đảm nhiệm một nhiệm vụ hay chỉ có một lý do duy nhất để thay đổi.

Hệ thống sử dụng mô hình MVC, mỗi một chức năng có một controller xử lý riêng.

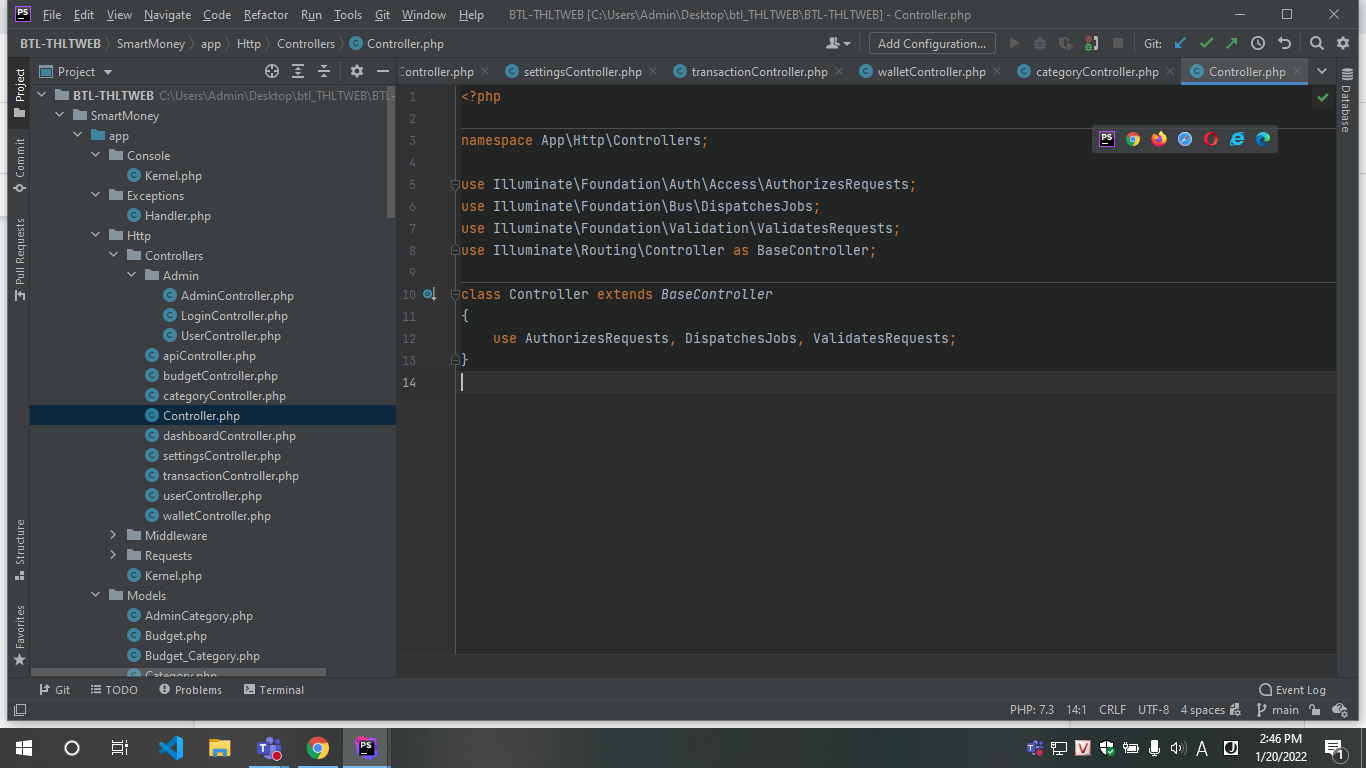


Trong chức năng hiển thị các thống kê, tách riêng thành 2 controller là dashboardController và apiController để đảm nhiệm 2 nhiệm vụ điều khiển liên quan đến hiển thị của các thống kê và cung cấp các dữ liệu vẽ các biểu đồ thống kê.

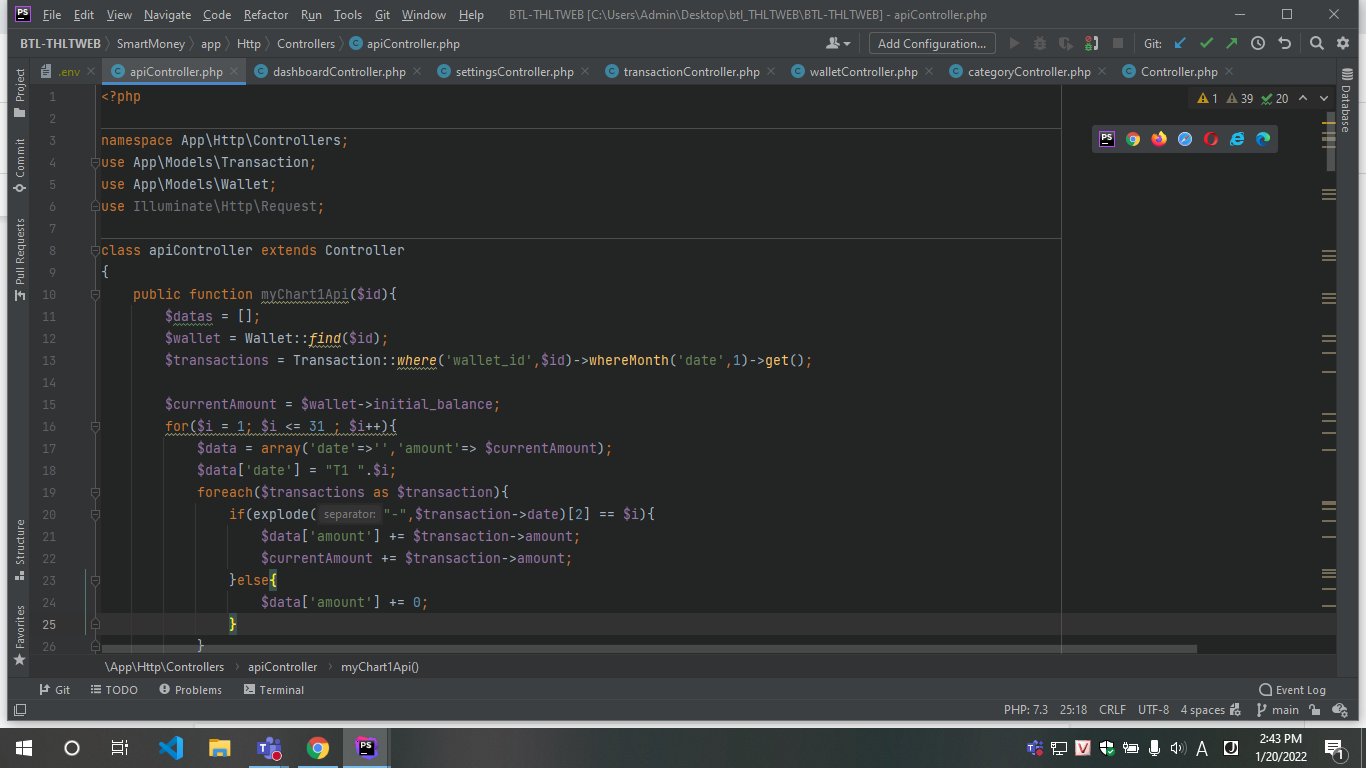
**O: Open Close Principle**

Các classes hay các thực thể nên đóng đối với việc chỉnh sửa và mở cho việc mở rộng.

Ví dụ điển hình trong trang web cho nguyên lý này là ban đầu có một controller chính và sẽ không chỉnh sửa vào controller này



Muốn thực hiện chỉnh sửa cần tạo ra các controller thành phần kế thừa controller này.



**L: Liskov Substitution Principle**

Các class có thể có sự hoán đổi. Nếu có 2 classes làm cùng nhiệm vụ hoặc cùng sử dụng một interface có thể dễ dàng thay đổi một class bằng class còn lại mà không làm ảnh hưởng đến hoạt động của website.

**I: Interface Segregation Principle**

Thay vì một sử dụng một interface quá lớn, quá nhiều phương thức thì chúng ta sẽ tách nhỏ ra thành các interface con với mục đích cụ thể. Vì khi để một interface quá to, các lớp implement sẽ phải implement các phương thức mà bản thân nó không cần dùng đến.

**D: Dependency Inversion principle**

Những thành phần trong một chương trình chỉ nên phụ thuộc vào những cái trừu tượng. Những thành phần trừu tượng không nên phụ thuộc vào một thành phần mang tính cụ thể mà nên ngược lại.

# Kết luận và hướng phát triển

## Kết luận

Trang web đã hoàn thành đầy đủ được các chức năng mong muốn , giống như các kế hoạch đã xây dựng từ trước. Thêm vào đó , bọn em cũng đã giải quyết được những vấn đề phi chuẩn hóa cơ sở dữ liệu.

Tuy nhiên , vẫn còn chức năng liên kết với tài khoản ngân hàng thì nhóm bọn em chưa phát triển. Và dự định sẽ phát triển trong tương lai.

## Hướng phát triển trong tương lai

Trong tương lai website có thể cung cấp thêm chức năng tiết kiệm và gợi ý đầu tư đối với các tài khoản Premium.

Chức năng tiết kiệm xây dựng dựa trên số dư còn lại sau mỗi kỳ của các Budget sẽ được chuyển vào một ví ( một tài khoản ngân hàng) để tiết kiệm và xuất ra thông báo liên quan đến số tiền tiết kiệm sau mỗi kỳ.

Chức năng gợi ý đầu tư xây dựng dựa trên chức năng tiết kiệm, tùy thuộc vào số tiền tiết kiệm mà người dùng hiện có, chức năng sẽ gợi ý cho người dùng đầu tư vào các thị trường tài chính tiềm năng để sinh lợi nhuận cao.

# Từ điển thuật ngữ

CRUD : Tạo, thêm, sửa, xóa

User : Người dùng

Admin : Quản trị viên

SM : Smart Money

Visitor : Khách

# Tài liệu tham khảo

Các slide bài giảng và các Lab thực hành của môn phát triển phần mềm theo chuẩn ITSS và môn thực hành lập trình Web