

PHẠM THỊ LINH

CÔNG NGHỆ THÔNG TIN

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE QUẢN LÝ PHÒNG GYM SỬ**  
**DỤNG ANGULAR VÀ .NET CORE API**

**GVHD:** ThS. Đặng Quỳnh Nga  
**Sinh viên:** Phạm Thị Linh  
**Mã sinh viên:** 2021608013

Hà Nội – Năm 2025

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  
-----

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE QUẢN LÝ PHÒNG GYM SỬ DỤNG**  
**ANGULAR VÀ .NET CORE API**

<b>GVHD:</b>	<b>ThS. Đặng Quỳnh Nga</b>
<b>Sinh viên:</b>	<b>Phạm Thị Linh</b>
<b>Mã sinh viên:</b>	<b>2021608013</b>

Hà Nội – Năm 2025

## LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc ứng dụng công nghệ vào công tác quản lý là một xu hướng tất yếu nhằm nâng cao hiệu quả hoạt động và tiết kiệm nguồn lực. Với mong muốn vận dụng các kiến thức đã học vào thực tiễn, em đã lựa chọn đề tài “Xây dựng website quản lý phòng gym” làm đề án tốt nghiệp của mình.

Trong quá trình thực hiện đề án, em đã cố gắng tìm hiểu, học hỏi và áp dụng các kiến thức về lập trình web, thiết kế hệ thống và quản lý cơ sở dữ liệu để hoàn thiện sản phẩm. Đề tài không chỉ giúp em rèn luyện kỹ năng lập trình mà còn giúp em hiểu rõ hơn quy trình phát triển một ứng dụng thực tế.

Tuy nhiên, do thời gian thực hiện có hạn và kinh nghiệm thực tế còn chưa nhiều, sản phẩm chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ quý thầy cô để hoàn thiện hơn trong những dự án tiếp theo.

Em xin gửi lời cảm ơn chân thành đến ThS. Đặng Quỳnh Nga – người đã hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề án. Nhờ những góp ý và định hướng từ cô, em đã có thể hoàn thiện đề tài một cách tốt nhất.

Em xin chân thành cảm ơn!

Phạm Thị Linh

**DANH MỤC CÁC TỪ VIẾT TẮT**

SQL	Structured Query Language
SSAS	SQL Server Analysis Services
SSRS	SQL Server Reporting Services
API	Application Programming Interface
RESTful	Representational State Transfer
PT	Huấn luyện viên (Personal Trainer)

## DANH MỤC CÁC HÌNH ẢNH

Hình 1: Các use case chính .....	16
Hình 2: Biểu đồ khái quát cơ sở dữ liệu .....	27
Hình 3: Hình ảnh quan hệ giữa các bảng .....	35
Hình 4: Thiết kế màn hình quản trị .....	57
Hình 5: Thiết kế màn hình Profile .....	57
Hình 6: Thiết kế màn hình Quản lý khách hàng .....	58
Hình 7: Thiết kế màn hình Quản lý nhân viên .....	58
Hình 8: Thiết kế màn hình Quản lý vật chất .....	59
Hình 9: Thiết kế màn hình Quản lý thẻ .....	59
Hình 10: Thiết kế màn hình Quản lý tài khoản .....	60
Hình 11: Sơ đồ điều hướng giữa các màn hình .....	60
Hình 12: Ngôn ngữ Angular .....	61
Hình 13: Microsoft SQL Server là gì ? .....	63
Hình 14: ASP.NET Core Web API là gì? .....	67
Hình 15: Màn hình Trang chủ .....	70
Hình 16: Màn hình Đăng nhập .....	70
Hình 17: Màn hình Quản lý tài khoản .....	71
Hình 18: Màn hình Quản lý tủ đồ .....	71
Hình 19: Màn hình Quản lý khách hàng .....	72
Hình 20: Màn hình Quản lý thẻ .....	72
Hình 21: Màn hình Check in .....	73
Hình 22: Màn hình Quản lý ca tập .....	73

## **DANH MỤC CÁC BẢNG**

Bảng 2-1. CARD_CHECK_IN .....	28
Bảng 2-2. CARD_INFO.....	28
Bảng 2-3. GOODS_EQUIPMENT .....	29
Bảng 2-4. PER_EMPLOYEE .....	29
Bảng 2-5. GOODS_LIST.....	30
Bảng 2-6. SYS_MENU .....	30
Bảng 2-7. GOODS_LOCKER .....	31
Bảng 2-8. GOODS_SHIFT .....	31
Bảng 2-9. PER_CUS_LIST_CARD .....	31
Bảng 2-10. PER_CUS_TRANSACTION.....	32
Bảng 2-11 . SYS_USER .....	32
Bảng 2-12. PER_CUSTOMER.....	33
Bảng 2-13 . SYS_LIST_OTHER.....	34
Bảng 2-14 . SYS_LIST_OTHER_TYPE.....	34
Bảng 4-1. Test case chức năng Đăng nhập .....	75
Bảng 4-2: Test case chức năng Quản lý tài khoản.....	76
Bảng 4-3: Test case chức năng Quản lý tủ đồ.....	77
Bảng 4-4: Test case chức năng Quản lý khách hàng .....	79
Bảng 4-5: Test case chức năng Quản lý thẻ.....	80
Bảng 4-6: Test case chức năng Quản lý ca tập .....	82

## MỤC LỤC

LỜI NÓI ĐẦU .....	1
DANH MỤC CÁC TỪ VIẾT TẮT .....	2
DANH MỤC CÁC HÌNH ẢNH .....	3
DANH MỤC CÁC BẢNG.....	4
MỤC LỤC.....	5
MỞ ĐẦU.....	8
1. Lý do chọn đề tài.....	8
2. Mục tiêu đề tài.....	8
3. Nội dung nghiên cứu .....	9
4. Phạm vi đề tài.....	9
5. Bố cục của đề tài .....	10
CHƯƠNG 1. KHẢO SÁT HỆ THỐNG.....	11
1.1. Mục tiêu .....	11
1.2. Phương pháp.....	11
1.3. Thông tin sơ bộ về hệ thống.....	15
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	16
2.1. Biểu đồ use case .....	16
2.1.1. Các use case chính .....	16
2.1.2. Mô tả chi tiết các use case.....	17
2.2. Thiết kế cơ sở dữ liệu.....	27
2.2.1. Mô hình hóa dữ liệu .....	27

2.2.2. Thiết kế bảng.....	28
2.2.3. Quan hệ các bảng .....	35
2.3. Thiết kế các thành phần của hệ thống .....	36
2.3.1. Use case Đăng nhập .....	36
2.3.2. Use case Quản lý tài khoản .....	39
2.3.3. Use case Quản lý tủ đồ.....	43
2.3.4. Use case Quản lý khách hàng .....	45
2.3.5. Use case Quản lý danh sách ca tập .....	49
2.3.6. Use case Quản lý thẻ .....	53
2.4. Thiết kế giao diện hình dung màn hình.....	57
2.4.1. Màn hình quản trị.....	57
2.4.2. Màn hình Profile .....	57
2.4.3. Màn hình Quản lý khách hàng .....	58
2.4.4. Màn hình Quản lý nhân viên.....	58
2.4.5. Màn hình Quản lý vật chất.....	59
2.4.6. Màn hình Quản lý thẻ.....	59
2.4.7. Màn hình Quản lý tài khoản.....	60
2.4.8. Sơ đồ điều hướng giữa các màn hình.....	60
<b>CHƯƠNG 3. XÂY DỰNG HỆ THỐNG.....</b>	<b>61</b>
3.1. Giới thiệu ngôn ngữ cài đặt.....	61
3.1.1. Giới thiệu về ngôn ngữ Angular .....	61
3.1.2. Giới thiệu về SQL Server.....	63
3.1.3. Giới thiệu về ASP.NET Core Web API.....	66
3.2. Các kết quả đạt được .....	70



3.2.1. Màn hình Trang chủ .....	70
3.2.2. Use case Đăng nhập .....	70
3.2.3. Use case Quản lý tài khoản .....	71
3.2.4. Use case Quản lý tủ đồ.....	71
3.2.5. Use case Quản lý khách hàng .....	72
3.2.6. Use case Quản lý thẻ .....	72
3.2.7. Use case Check in-out.....	73
3.2.8. Use case Quản lý ca tập .....	73
3.3. Kiểm thử hệ thống.....	74
3.3.1. Phạm vi kiểm thử .....	74
3.3.2. Test case .....	74
KẾT LUẬN .....	83
1. Kết quả đạt được .....	83
2. Hạn chế của dự án .....	84
3. Hướng phát triển .....	84
TÀI LIỆU THAM KHẢO.....	86

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Các chuyên gia cho rằng công nghệ sẽ trở thành vũ khí quyết định lợi thế cạnh tranh của một đơn vị. Việc áp dụng công nghệ giúp cơ sở tối ưu quy trình, tiết kiệm thời gian và tăng chất lượng phục vụ khách hàng. Tuy nhiên, việc quản lý thủ công theo mô hình quản lý gặp rất nhiều khó khăn trong các hoạt động như quản lý nhân sự không hiệu quả, dữ liệu khách hàng lưu trữ nhiều nơi, thiếu tính nhất quán, khó khăn khi tra cứu, không sử dụng data để truyền thông hiệu quả. Ngoài ra, việc chăm sóc khách hàng không tốt mang lại trải nghiệm không hài lòng: khách hàng chờ PT, chờ máy tập, không có nhân viên hướng dẫn.

Giải quyết vấn đề này, chúng em đã đưa ra giải pháp website: Quản lý phòng tập gym, yoga & fitness chuyên nghiệp. Giải pháp sẽ hỗ trợ doanh nghiệp của bạn quản lý chặt chẽ quy trình quản lý phòng tập từ khâu nhập thiết bị, quản lý thẻ và các nghiệp vụ thẻ, thanh toán, quản lý thu chi và báo cáo tài chính kinh doanh. Ngoài ra, hệ thống còn cung cấp các chức năng như quản lý thông tin hội viên, cho phép hội viên đặt lịch tập, đăng ký lớp học, theo dõi lịch huấn luyện viên và thanh toán trực tuyến.... Với phương pháp quản lý hữu hiệu, tốn ít chi phí nhất mà hiệu quả nhất sẽ đem lại sự thuận tiện và lợi nhuận tối đa cho khách hàng.

### 2. Mục tiêu đề tài

Website cung cấp các chức năng chính như quản lý thông tin khách hàng, lịch tập, các lớp tập, và các dịch vụ khác của phòng gym. Ngoài ra, website còn hỗ trợ quản lý tài chính, bao gồm việc theo dõi doanh thu, chi phí và lập báo cáo tài chính. Đặc biệt, website còn tích hợp hệ thống đặt lịch tập trực tuyến, giúp khách hàng có thể dễ dàng đăng ký các lớp tập mà không cần phải đến trực tiếp phòng gym. Điều này không chỉ mang lại sự tiện lợi cho khách hàng mà còn giúp quản lý phòng gym trở nên hiệu quả hơn.

### 3. Nội dung nghiên cứu

Nội dung tập trung vào việc phân tích, thiết kế và triển khai hệ thống website quản lý phòng gym từ phía người dùng (frontend) sử dụng Angular, kết hợp với backend được xây dựng bằng .NET Core API và hệ quản trị cơ sở dữ liệu SQL Server. Hệ thống nhằm phục vụ cho việc quản lý khách hàng, huấn luyện viên, gói tập, lịch tập và các hoạt động khác tại phòng gym một cách hiệu quả, tiện lợi và hiện đại. Nội dung cụ thể bao gồm:

- Xây dựng giao diện người dùng (frontend) với Angular: Thiết kế và phát triển các thành phần giao diện như trang đăng nhập, trang quản lý hội viên, quản lý huấn luyện viên, quản lý gói tập, lịch tập và báo cáo thống kê.
- Xây dựng backend với .NET Core API: Phát triển các API RESTful phục vụ cho việc truyền và nhận dữ liệu giữa frontend và backend. Backend xử lý các nghiệp vụ chính như thêm/sửa/xóa thông tin hội viên, lịch tập, gói tập, theo dõi thanh toán, và quản lý người dùng hệ thống. Dữ liệu được lưu trữ và quản lý bằng SQL Server.
- Triển khai và kiểm thử: Tiến hành triển khai hệ thống trên môi trường thực tế, thực hiện kiểm thử chức năng, kiểm thử hiệu suất và đánh giá trải nghiệm người dùng để đảm bảo hệ thống hoạt động ổn định, an toàn và đáp ứng đúng nhu cầu sử dụng.

### 4. Phạm vi đề tài

Phạm vi của đề án tập trung vào việc xây dựng một hệ thống website quản lý phòng gym hoàn chỉnh. Hệ thống sẽ phục vụ cho hai nhóm người dùng chính là quản trị viên và khách hàng, cung cấp đầy đủ các chức năng thiết yếu giúp quản lý và vận hành phòng gym một cách hiệu quả, hiện đại và thuận tiện.

Về phía khách hàng, hệ thống cho phép cập nhật thông tin cá nhân, xem và đăng ký các gói tập, lịch tập, cũng như đăng ký tham gia các lớp huấn luyện. Hội

viên có thể theo dõi thời gian hiệu lực của thẻ tập, lựa chọn huấn luyện viên theo nhu cầu và thực hiện thanh toán trực tuyến đối với các dịch vụ mất phí.

Về phía quản trị viên, hệ thống hỗ trợ quản lý thông tin hội viên, phân loại và theo dõi các gói dịch vụ, gói tập, lịch sử các dịch vụ đã sử dụng, phân công huấn luyện viên, và kiểm soát việc đặt lịch tập luyện. Đồng thời, hệ thống cũng tích hợp chức năng phân quyền người dùng, cho phép giới hạn quyền truy cập theo từng vai trò trong hệ thống.

Ngoài ra, hệ thống còn cung cấp các báo cáo thống kê giúp nhà quản lý có cái nhìn tổng thể và đưa ra các quyết định quản trị phù hợp.

## **5. Bố cục của đề tài**

Ngoài hai phần Mở đầu và Kết luận, Báo cáo đồ án tốt nghiệp có ba chương như sau:

Chương 1: Khảo sát hệ thống dự án xây dựng website quản lý phòng gym.

Chương 2: Trình bày về việc khảo sát và đặc tả hệ thống website quản lý phòng gym, thiết kế chức năng, thiết kế cơ sở dữ liệu và thiết kế giao diện của hệ thống.

Chương 3: Trình bày về việc cài đặt phần mềm và các kết quả đạt được; các phương pháp kiểm thử được sử dụng để kiểm thử website quản lý phòng gym.

## CHƯƠNG 1. KHẢO SÁT HỆ THỐNG

### 1.1. Mục tiêu

- Tìm hiểu và đánh giá nhu cầu thực tế trong việc quản lý phòng tập gym, yoga & fitness hiện nay.
- Đánh giá và cải thiện trải nghiệm người dùng.
- Tìm hiểu và phân tích đối tượng sử dụng hệ thống, bao gồm quản lý và khách hàng.
- Thu thập dữ liệu liên quan đến hành vi người dùng nhằm phục vụ cho việc phân tích và cải thiện chất lượng dịch vụ.

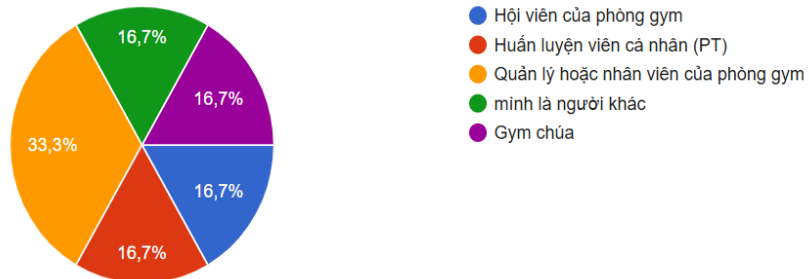
### 1.2. Phương pháp

- Điều tra

1. Bạn là ai trong số những đối tượng sau đây? (Chọn một đáp án)

 Sao chép biểu đồ

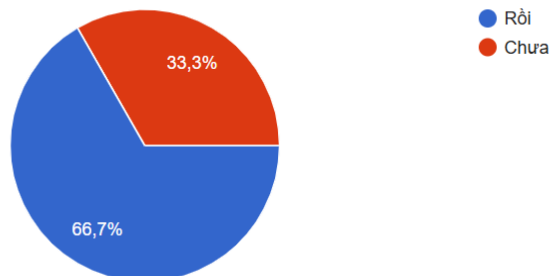
6 câu trả lời



2. Bạn đã từng sử dụng website hoặc ứng dụng quản lý phòng gym chưa?

 Sao chép biểu đồ

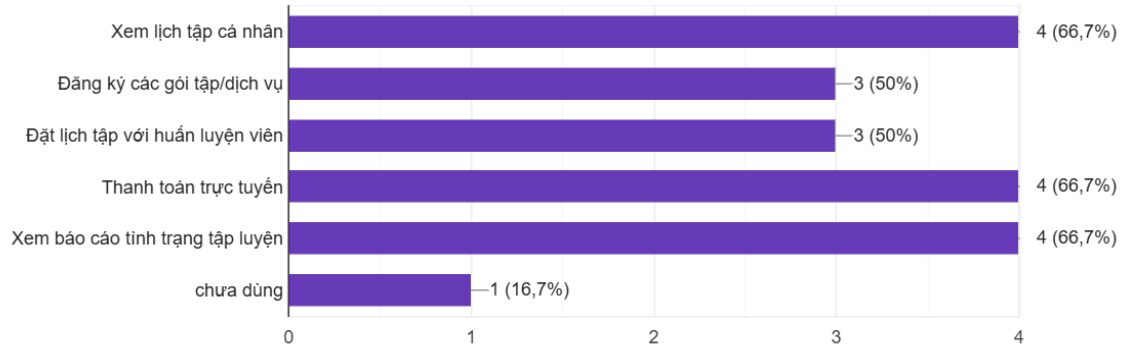
6 câu trả lời



3. Nếu đã sử dụng, bạn thường sử dụng website để làm gì? (Có thể chọn nhiều đáp án)

 Sao chép biểu đồ

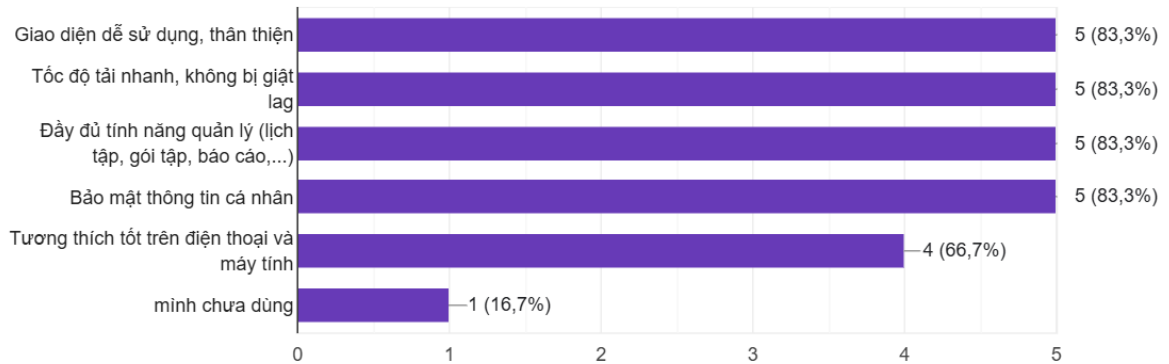
6 câu trả lời



4. Điều gì quan trọng nhất đối với bạn khi sử dụng website quản lý phòng gym? ( Có thể chọn nhiều đáp án )

 Sao chép biểu đồ

6 câu trả lời



5. Bạn gặp khó khăn gì khi sử dụng các website quản lý phòng gym hiện tại (nếu có)?

6 câu trả lời

Ít tiện ích thực tế, tốc độ load chậm

mình chưa dùng

Giao diện khó sử dụng

Không

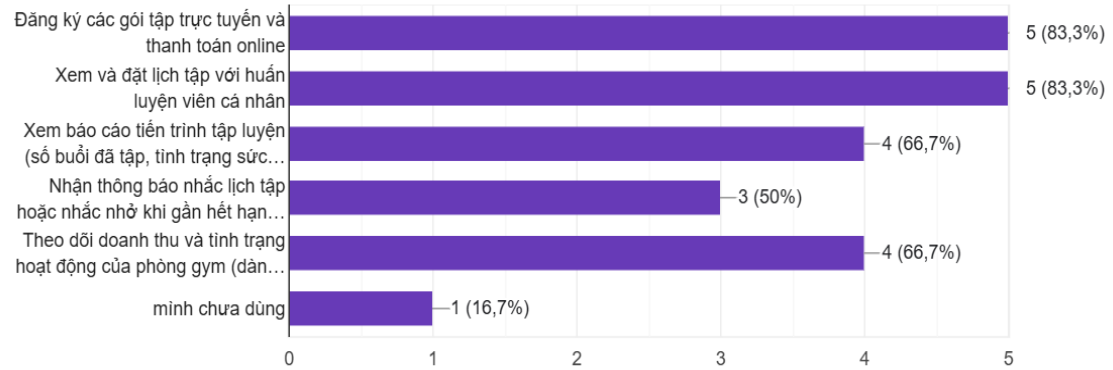
khó sử dụng

tốc độ chậm, nhận diện học viên kém, sắp xếp lịch chưa đc thông minh và chưa tương thích với điện thoại

6. Bạn mong muốn website có thêm những tính năng nào? (Có thể chọn nhiều đáp án)

 Sao chép biểu đồ

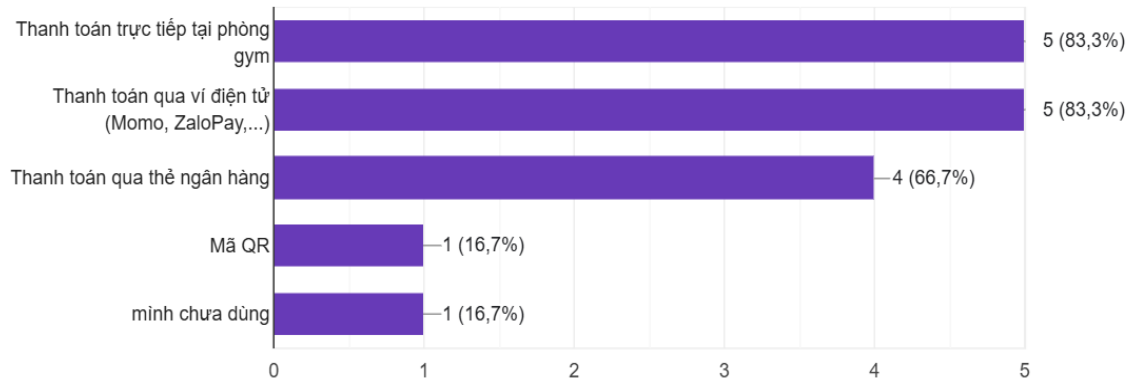
6 câu trả lời



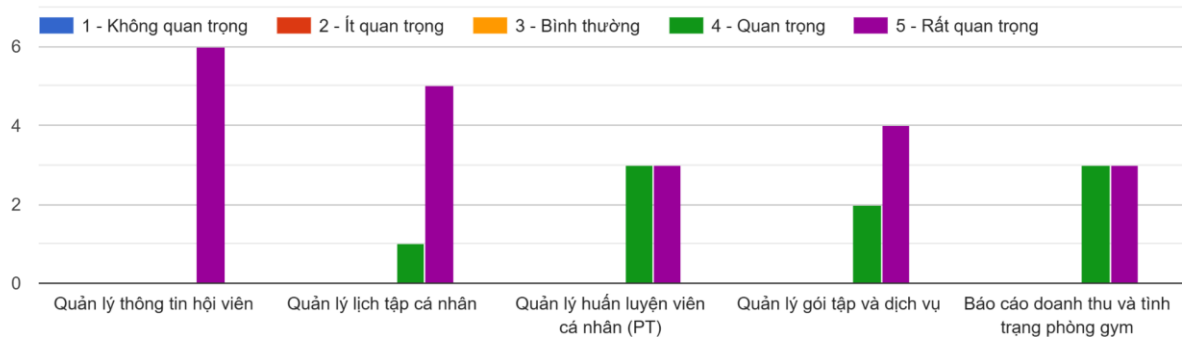
7. Bạn muốn phương thức thanh toán nào được hỗ trợ trên website? (Chọn nhiều đáp án nếu cần)

 Sao chép biểu đồ

6 câu trả lời



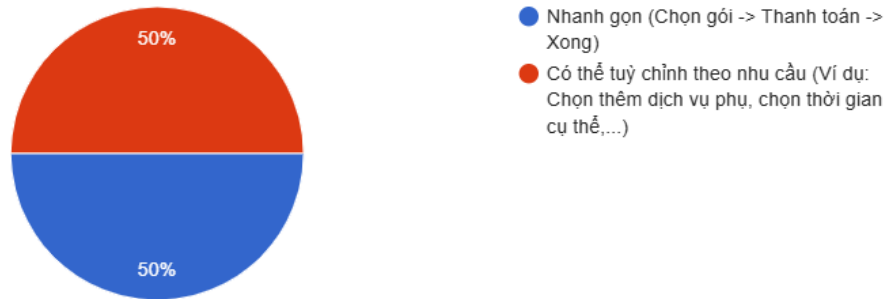
8. Bạn đánh giá mức độ quan trọng của các tính năng sau:



9. Bạn mong muốn quá trình đăng ký gói tập/dịch vụ diễn ra như thế nào?

 Sao chép biểu đồ

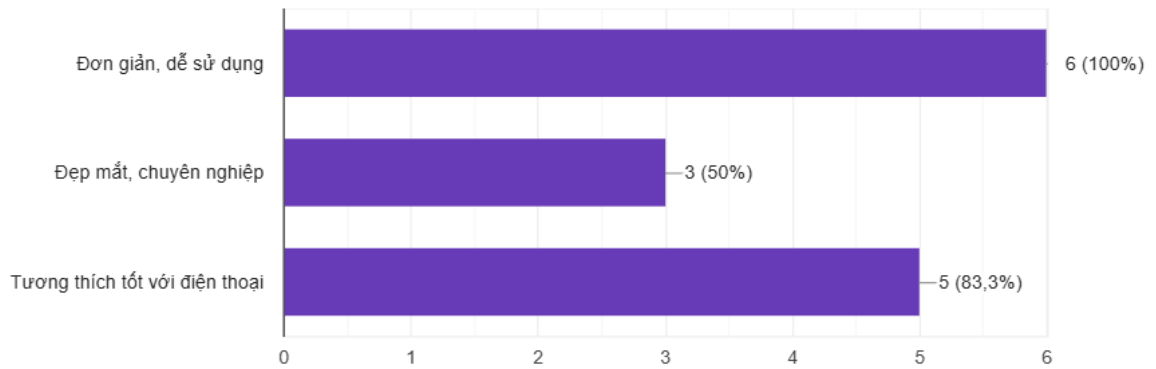
6 câu trả lời



10. Bạn muốn giao diện website như thế nào?

 Sao chép biểu đồ

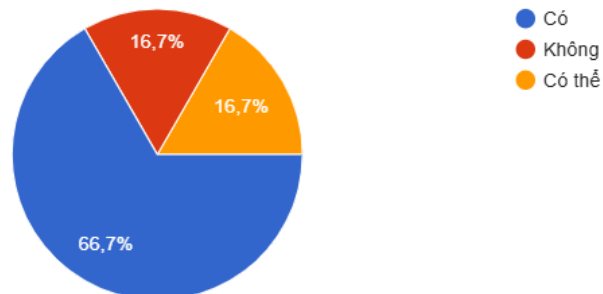
6 câu trả lời



11. Bạn có sẵn sàng sử dụng website này nếu nó được phát triển đầy đủ các tính năng bạn mong muốn không?

 Sao chép biểu đồ

6 câu trả lời





### **1.3. Thông tin sơ bộ về hệ thống**

Website cung cấp giải pháp quản lý phòng tập gym, yoga & fitness giúp người dùng thao tác nhanh chóng và hiệu quả.

Người dùng có thể đăng nhập theo vai trò: quản trị viên hoặc khách hàng.

Quản trị viên có thể quản lý hội viên, huấn luyện viên, thiết bị, gói tập, lịch tập, theo dõi thu chi và báo cáo tài chính.

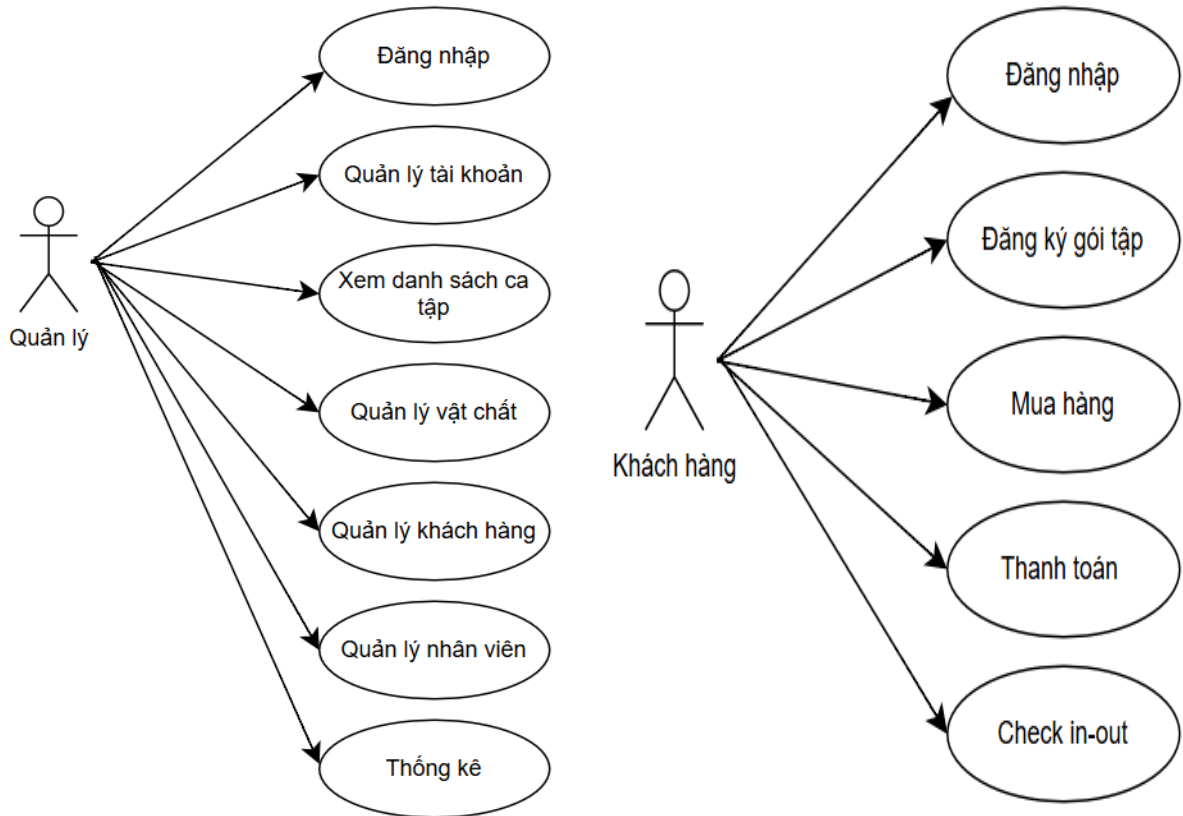
Hội viên có thể xem và đặt lịch tập, đăng ký gói tập, theo dõi quá trình luyện tập, thanh toán và gửi đánh giá.

Hệ thống sẽ cập nhật báo cáo thống kê định kỳ (tuần/tháng) về số lượng hội viên, lượt đăng ký, doanh thu và các lớp học phổ biến.

## CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Biểu đồ use case

#### 2.1.1. Các use case chính



Hình 1: Các use case chính

- 1) **Đăng nhập:** Cho phép người dùng đăng nhập vào tài khoản đã có.
- 2) **Quản lý tài khoản:** Cho phép người quản lý thực hiện các thao tác quản lý tài khoản trên hệ thống.
- 3) **Thống kê:** Giúp người quản lý thống kê xem trong một tháng cửa hàng bán được bao nhiêu sản phẩm, doanh thu bao nhiêu và bán những mặt hàng gì.
- 4) **Quản lý vật chất:** Cho phép người quản lý thực hiện các thao tác khóa/mở tủ đồ cho khách hàng và sửa thông tin trạng thái tủ đồ vào hệ thống.
- 5) **Quản lý khách hàng:** Cho phép người quản lý thực hiện các thao tác quản lý tài khoản khách hàng trên hệ thống.

**6) Quản lý thẻ:** Cho phép người quản lý thực hiện các thao tác quản lý thẻ trên hệ thống.

**7) Quản lý danh sách bài tập:** Cho phép người quản lý có thể quản lý danh sách bài tập vào hệ thống.

**8) Đăng ký gói tập:** Cho phép khách hàng có thể đăng ký các gói tập của phòng tập.

**9) Mua hàng:** Cho phép khách hàng có thể chọn xem thông tin chi tiết sản phẩm và mua sản phẩm mong muốn.

**10) Check in-out:** Cho phép khách hàng có thể check in-out thông qua thẻ.

**11) Quản lý danh sách ca tập:** Cho phép người quản lý xem danh sách ca tập có trong hệ thống.

### **2.1.2. Mô tả chi tiết các use case**

#### **2.1.2.1. Đăng nhập**

**Mô tả:** Use case này cho phép người dùng đăng nhập vào tài khoản đã có.

**Các luồng sự kiện:**

– Luồng cơ bản:

1) Use case này bắt đầu khi người dùng truy cập trang web, hệ thống hiển thị một màn hình điền thông tin đăng nhập vào tài khoản.

2) Người dùng nhập Tên đăng nhập, Mật khẩu và kích chọn “Đăng nhập”, hệ thống đọc bảng SYS\_USER và kiểm tra thông tin tài khoản, nếu thông tin trùng khớp thì chuyển tới trang chủ. Use case kết thúc.

– Luồng rẽ nhánh:

1) Tại bước 2 trong luồng cơ bản, nếu thông tin tài khoản đăng nhập không trùng khớp với cơ sở dữ liệu trong bảng SYS\_USER, hệ thống hiển thị thông báo lỗi.

2) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Không có
- Hậu điều kiện: Không có
- Điểm mở rộng: Không có

#### **2.1.2.2. Quản lý tài khoản**

**Mô tả:** Use case này cho phép người quản lý thực hiện các thao tác quản lý tài khoản trên hệ thống như: thêm mới, sửa, xóa, và phân quyền cho các tài khoản.

##### **Các luồng sự kiện:**

- Luồng cơ bản:

1) Use case bắt đầu khi quản lý đăng nhập vào hệ thống và chọn “Quản lý tài khoản”.

2) Quản lý lựa chọn một trong các thao tác: thêm mới, sửa, xóa, hoặc phân quyền tài khoản.

3) Hệ thống hiển thị form tương ứng với thao tác.

4) Quản lý nhập thông tin cần thiết:

- Thêm: Nhập thông tin tài khoản mới.
- Sửa: Chọn tài khoản và cập nhật các thông tin.
- Xóa: Chọn tài khoản và xác nhận thao tác xóa.
- Phân quyền: Chọn tài khoản và gán quyền phù hợp.

5) Hệ thống kiểm tra và lưu thông tin vào bảng PER\_EMPLOYEE và quyền được cập nhật vào bảng SYS\_USER sau đó hiển thị thông báo thực hiện thành công. Use case kết thúc.

- Luồng rẽ nhánh:

1) Tại bước 4 trong luồng cơ bản, nếu nhập thông tin không hợp lệ, hệ thống hiển thị lỗi tương ứng.

2) Nếu trong quá trình thực hiện thao tác, người dùng nhấn nút “Hủy bỏ”, hệ thống quay lại danh sách tài khoản, use case kết thúc.

3) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Không có
- Hậu điều kiện: Không có

### **2.1.2.3. Thống kê**

**Mô tả:** Use case này sẽ thống kê về số lượng hội viên, doanh thu từ các gói dịch vụ, tình trạng hoạt động của phòng gym.

#### **Các luồng sự kiện:**

- Luồng cơ bản:

1) Use case bắt đầu khi người quản lý đăng nhập vào hệ thống và chọn “Thống kê”.

2) Hệ thống sẽ lấy thông tin từ bảng PER\_CUS\_TRANSACTION và hiển thị số lượng khách hàng, doanh thu trong tháng và danh sách sản phẩm dựa theo yêu cầu của người dùng. Use case kết thúc.

– Luồng rẽ nhánh: Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Người dùng phải là quản lý và đã đăng nhập vào hệ thống
- Hậu điều kiện: Không có
- Điểm mở rộng: Không có

#### 2.1.2.4. Quản lý tủ đồ

**Mô tả:** Use case này cho phép người quản lý thực hiện các thao tác khóa/mở tủ đồ cho khách hàng và sửa thông tin trạng thái tủ đồ vào hệ thống.

##### Các luồng sự kiện:

– Luồng cơ bản:

1) Use case bắt đầu khi quản lý chọn chức năng "Quản lý vật chất" trên hệ thống

2) Hệ thống hiển thị danh sách tủ đồ với các trạng thái hiện tại (khóa/mở)

3) Quản lý thực hiện một trong các thao tác:

- Khóa tủ: Chọn tủ đang mở → nhấn "Khóa"
- Mở tủ: Chọn tủ đã khóa → nhấn "Mở"

4) Hệ thống kiểm tra quyền và cập nhật trạng thái mới vào bảng GOODS\_LOCKER sau đó hiển thị thông báo thành công. Use case kết thúc.

– Luồng rẽ nhánh: Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

– Các yêu cầu đặc biệt: Không có

– Tiên điều kiện: Không có

– Hậu điều kiện: Không có

– Điểm mở rộng: Không có

#### 2.1.2.5. Quản lý khách hàng

**Mô tả:** Use case này cho phép người quản lý thực hiện các thao tác quản lý tài khoản khách hàng trên hệ thống như: thêm mới, sửa, xóa cho các tài khoản.

##### Các luồng sự kiện:

– Luồng cơ bản:

1) Use case bắt đầu khi quản lý đăng nhập vào hệ thống và chọn “Quản lý khách hàng”.

2) Quản lý lựa chọn một trong các thao tác: thêm mới, sửa, xóa tài khoản.

3) Hệ thống hiển thị form tương ứng với thao tác.

4) Quản lý nhập thông tin cần thiết:

- Thêm: Nhập thông tin tài khoản mới.
- Sửa: Chọn tài khoản và cập nhật các thông tin.
- Xóa: Chọn tài khoản và xác nhận thao tác xóa.

5) Hệ thống kiểm tra và lưu thông tin vào bảng PER\_CUSTOMER sau đó hiển thị thông báo thực hiện thành công. Use case kết thúc.

– Luồng rẽ nhánh:

1) Tại bước 4 trong luồng cơ bản, nếu nhập thông tin không hợp lệ, hệ thống hiển thị lỗi tương ứng.

2) Nếu trong quá trình thực hiện thao tác, người dùng nhấn nút “Hủy bỏ”, hệ thống quay lại danh sách tài khoản, use case kết thúc.

3) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Người dùng là quản lý
- Hậu điều kiện: Không có
- Điểm mở rộng: Không có

#### **2.1.2.6. Quản lý thẻ**

**Mô tả:** Use case này cho phép người quản lý thực hiện các thao tác quản lý thẻ trên hệ thống như: thêm mới, sửa, xóa thẻ.

**Các luồng sự kiện:**

- Luồng cơ bản:

1) Use case bắt đầu khi quản lý đăng nhập vào hệ thống và truy cập trang “Quản lý thẻ”.

2) Quản lý lựa chọn một trong các thao tác: thêm mới, sửa, xóa thẻ.

3) Hệ thống hiển thị form tương ứng với thao tác.

4) Quản lý nhập thông tin cần thiết:

- Thêm: Nhập thông tin thẻ mới.
- Sửa: Chọn thẻ và cập nhật các trường thông tin.
- Xóa: Chọn thẻ và xác nhận thao tác xóa.

5) Hệ thống kiểm tra và lưu thông tin vào bảng CARD\_INFO sau đó hiển thị thông báo thực hiện thành công. Use case kết thúc.

– Luồng rẽ nhánh:

1) Tại bước 4 trong luồng cơ bản, nếu nhập thông tin không hợp lệ, hệ thống hiển thị lỗi tương ứng.

2) Nếu trong quá trình thực hiện thao tác, người dùng nhấn nút “Hủy bỏ”, hệ thống quay lại danh sách tài khoản, use case kết thúc.

3) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Người dùng là quản lý
- Hậu điều kiện: Không có
- Điểm mở rộng: Không có

#### **2.1.2.7. Quản lý danh sách bài tập**

**Mô tả:** Use case này sẽ giúp người quản lý có thể quản lý danh sách bài tập trong hệ thống.

**Các luồng sự kiện:**

- Luồng cơ bản:



1) Use case bắt đầu khi người quản lý đăng nhập vào hệ thống và truy cập “Danh sách bài tập”.

2) Người quản lý lựa chọn một trong các thao tác: thêm mới, sửa, xóa bài tập.

3) Hệ thống hiển thị form tương ứng với thao tác.

4) Người quản lý nhập thông tin cần thiết:

- Thêm: Thêm bài tập mới.
- Sửa: Chọn bài tập và cập nhật các thông tin.
- Xóa: Chọn bài tập và xác nhận thao tác xóa.

5) Hệ thống kiểm tra và lưu thông tin vào bảng sau đó hiển thị thông báo thực hiện thành công. Use case kết thúc.

– Luồng rẽ nhánh:

1) Tại bước 4 trong luồng cơ bản, nếu nhập thông tin không hợp lệ, hệ thống hiển thị lỗi tương ứng.

2) Nếu trong quá trình thực hiện thao tác, người dùng nhấn nút “Hủy bỏ”, hệ thống quay lại danh sách tài khoản, use case kết thúc.

3) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Không có
- Hậu điều kiện: Không có
- Các điểm mở rộng: Không có

#### **2.1.2.8. Đăng ký gói tập**

**Mô tả:** Use case này giúp khách hàng có thể đăng ký gói tập của phòng tập

**Các luồng sự kiện:**

– Luồng cơ bản:

1) Use case bắt đầu khi khách hàng đăng nhập vào hệ thống và truy cập “Đăng ký gói tập”.

- 2) Hệ thống hiển thị danh sách các gói tập đang có.
- 3) Khách hàng đăng ký gói tập phù hợp
- 4) Hệ thống cập nhật thông tin gói tập vào bảng PER\_CUSTOMER và hiển thị thông báo đăng ký thành công. Use case kết thúc.

– Luồng rẽ nhánh:

1) Tại bước 4 trong luồng cơ bản, nếu gói tập đã có trong cơ sở dữ liệu thì sẽ thông báo lên giao diện cho khách hàng.

2) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

– Các yêu cầu đặc biệt: Không có

– Tiên điều kiện: Không có

– Hậu điều kiện: Không có

– Điểm mở rộng: Không có

#### **2.1.2.9. Mua hàng**

**Mô tả:** Use case này cho phép khách hàng xem thông tin về sản phẩm như dụng cụ tập luyện, thực phẩm bổ sung và quần áo thể thao được bán tại phòng tập.

##### **Các luồng sự kiện:**

– Luồng cơ bản:

1) Use case bắt đầu khi khách hàng đăng nhập vào hệ thống và truy cập màn hình “Danh sách hàng hóa”.

2) Hệ thống hiển thị danh sách sản phẩm.

3) Khách hàng kích chọn sản phẩm muốn xem chi tiết, hệ thống hiển thị thông tin chi tiết của sản phẩm. Use case kết thúc.

– Luồng rẽ nhánh: Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Không có
- Hậu điều kiện: Không có
- Các điểm mở rộng: Không có

#### **2.1.2.10. Check in-out**

**Mô tả:** Use case này sẽ giúp khách hàng check in-out thông qua mã số thẻ

**Các luồng sự kiện:**

- Luồng cơ bản:
  - 1) Use case bắt đầu khi khách hàng đăng nhập vào hệ thống, chọn “Quản lý thẻ” và chọn “Check in”.
  - 2) Hệ thống sẽ chuyển hướng đến phần check in tương ứng và lưu vào bảng CARD\_CHECK\_IN. Use case kết thúc.
- Luồng rẽ nhánh: Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Không có
- Hậu điều kiện: Không có
- Điểm mở rộng: Không có

#### **2.1.2.11. Quản lý danh sách ca tập**

**Mô tả:** Use case này giúp người quản lý xem danh sách ca tập trong hệ thống.

**Các luồng sự kiện:**

- Luồng cơ bản:
  - 1) Use case bắt đầu khi người quản lý đăng nhập vào hệ thống và truy cập “Danh sách ca tập”.
  - 2) Người quản lý lựa chọn một trong các thao tác: thêm mới, sửa, xóa ca tập.

3) Hệ thống hiển thị form tương ứng với thao tác.

4) Người quản lý nhập thông tin cần thiết:

- Thêm: Thêm ca tập mới.
- Sửa: Chọn ca tập và cập nhật các thông tin.
- Xóa: Chọn ca tập và xác nhận thao tác xóa.

5) Hệ thống kiểm tra và lưu thông tin vào bảng GOODS\_SHIFT sau đó hiển thị thông báo thực hiện thành công. Use case kết thúc.

– Luồng rẽ nhánh:

1) Tại bước 4 trong luồng cơ bản, nếu nhập thông tin không hợp lệ, hệ thống hiển thị lỗi tương ứng.

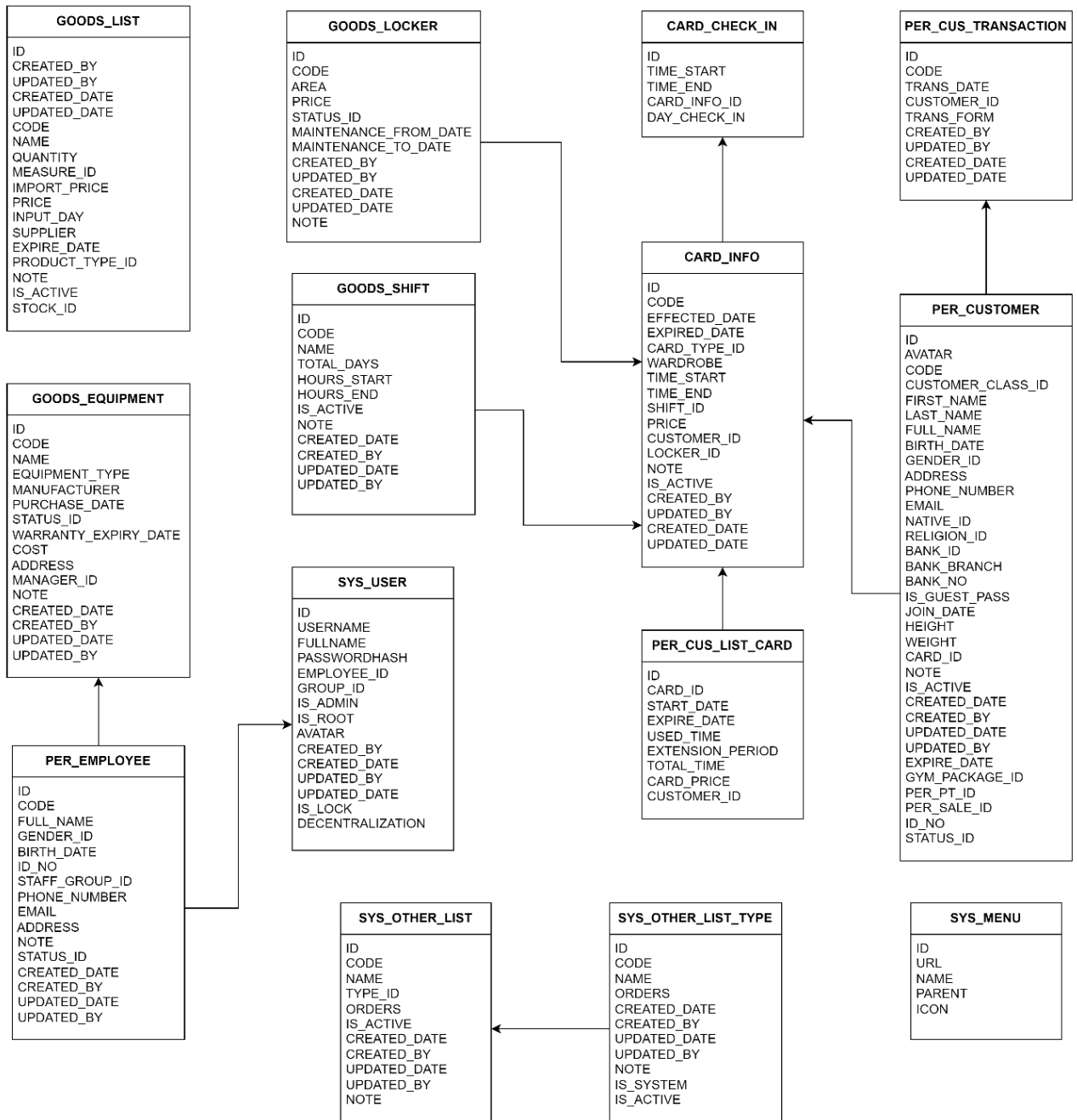
2) Nếu trong quá trình thực hiện thao tác, người dùng nhấn nút “Hủy bỏ”, hệ thống quay lại danh sách tài khoản, use case kết thúc.

3) Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không có
- Tiền điều kiện: Không có
- Hậu điều kiện: Không có
- Các điểm mở rộng: Không có

## 2.2. Thiết kế cơ sở dữ liệu

### 2.2.1. Mô hình hóa dữ liệu



Hình 2: Biểu đồ khái quát cơ sở dữ liệu

### 2.2.2. Thiết kế bảng

	Column Name	Data Type	Allow Nulls
✎	ID	bigint	<input type="checkbox"/>
	TIME_START	datetime2(7)	<input checked="" type="checkbox"/>
	TIME_END	datetime2(7)	<input checked="" type="checkbox"/>
	CARD_INFO_ID	bigint	<input checked="" type="checkbox"/>
	DAY_CHECK_IN	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Bảng 2-1. CARD\_CHECK\_IN*

	Column Name	Data Type	Allow Nulls
✎	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(50)	<input checked="" type="checkbox"/>
	EFFECTED_DATE	nvarchar(500)	<input checked="" type="checkbox"/>
	EXPIRED_DATE	nvarchar(500)	<input checked="" type="checkbox"/>
	EFFECTED_DATE_TIME	datetime2(7)	<input checked="" type="checkbox"/>
	EXPIRED_DATE_TIME	datetime2(7)	<input checked="" type="checkbox"/>
	CARD_TYPE_ID	bigint	<input checked="" type="checkbox"/>
	WARDROBE	bit	<input checked="" type="checkbox"/>
	SHIFT_ID	bigint	<input checked="" type="checkbox"/>
	PRICE	bigint	<input checked="" type="checkbox"/>
	CUSTOMER_ID	bigint	<input checked="" type="checkbox"/>
	LOCKER_ID	bigint	<input checked="" type="checkbox"/>
	NOTE	nvarchar(500)	<input checked="" type="checkbox"/>
	IS_ACTIVE	bit	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	IS_HAVE_PT	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Bảng 2-2. CARD\_INFO*

	Column Name	Data Type	Allow Nulls
PK	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(150)	<input checked="" type="checkbox"/>
	NAME	nvarchar(500)	<input checked="" type="checkbox"/>
	EQUIPMENT_TYPE	bigint	<input checked="" type="checkbox"/>
	MANUFACTURER	nvarchar(MAX)	<input checked="" type="checkbox"/>
	PURCHASE_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	STATUS_ID	bigint	<input checked="" type="checkbox"/>
	WARRANTY_EXPIRY_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	COST	decimal(18, 2)	<input checked="" type="checkbox"/>
	ADDRESS	nvarchar(MAX)	<input checked="" type="checkbox"/>
	MANAGER_ID	bigint	<input checked="" type="checkbox"/>
	NOTE	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(250)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-3. GOODS\_EQUIPMENT

	Column Name	Data Type	Allow Nulls
PK	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(255)	<input type="checkbox"/>
	FULL_NAME	nvarchar(255)	<input type="checkbox"/>
	GENDER_ID	bigint	<input type="checkbox"/>
	BIRTH_DATE	nvarchar(55)	<input type="checkbox"/>
	ID_NO	nvarchar(55)	<input type="checkbox"/>
	STAFF_GROUP_ID	bigint	<input type="checkbox"/>
	PHONE_NUMBER	nvarchar(55)	<input type="checkbox"/>
	EMAIL	nvarchar(55)	<input checked="" type="checkbox"/>
	ADDRESS	nvarchar(255)	<input type="checkbox"/>
	NOTE	nvarchar(255)	<input checked="" type="checkbox"/>
	STATUS_ID	bigint	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-4. PER\_EMPLOYEE

	Column Name	Data Type	Allow Nulls
✎	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(450)	<input checked="" type="checkbox"/>
	NAME	nvarchar(450)	<input checked="" type="checkbox"/>
	PRODUCT_TYPE_ID	bigint	<input checked="" type="checkbox"/>
	SUPPLIER	nvarchar(450)	<input checked="" type="checkbox"/>
	IMPORT_PRICE	decimal(18, 2)	<input checked="" type="checkbox"/>
	PRICE	decimal(18, 2)	<input checked="" type="checkbox"/>
	QUANTITY	bigint	<input checked="" type="checkbox"/>
	MEASURE_ID	bigint	<input checked="" type="checkbox"/>
	RECEIVING_DATE	nvarchar(100)	<input checked="" type="checkbox"/>
	RECEIVING_DATETIME	datetime2(7)	<input checked="" type="checkbox"/>
	EXPIRE_DATE	nvarchar(100)	<input checked="" type="checkbox"/>
	EXPIRE_DATETIME	datetime2(7)	<input checked="" type="checkbox"/>
	LOCATION	nvarchar(450)	<input checked="" type="checkbox"/>
	STATUS	bigint	<input checked="" type="checkbox"/>
	NOTE	nvarchar(450)	<input checked="" type="checkbox"/>
	BATCH_NO	nvarchar(200)	<input checked="" type="checkbox"/>
	WARRANTY_INFOR	nvarchar(450)	<input checked="" type="checkbox"/>
	DESCRIPTION	nvarchar(450)	<input checked="" type="checkbox"/>
	SOURCE	nvarchar(450)	<input checked="" type="checkbox"/>
	MANAGER_ID	bigint	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-5. GOODS\_LIST

	Column Name	Data Type	Allow Nulls
✎	ID	bigint	<input type="checkbox"/>
	URL	nvarchar(500)	<input checked="" type="checkbox"/>
	NAME	nvarchar(500)	<input checked="" type="checkbox"/>
	PARENT	bigint	<input checked="" type="checkbox"/>
	ICON	nvarchar(500)	<input checked="" type="checkbox"/>
	IS_HIDDEN	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-6. SYS\_MENU



	Column Name	Data Type	Allow Nulls
PK	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(450)	<input checked="" type="checkbox"/>
	AREA	bigint	<input checked="" type="checkbox"/>
	PRICE	bigint	<input checked="" type="checkbox"/>
	STATUS_ID	bigint	<input checked="" type="checkbox"/>
	MAINTENANCE_FROM_DATE	nvarchar(450)	<input checked="" type="checkbox"/>
	MAINTENANCE_TO_DATE	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	NOTE	nvarchar(450)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-7. GOODS\_LOCKER

	Column Name	Data Type	Allow Nulls
PK	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(50)	<input checked="" type="checkbox"/>
	NAME	nvarchar(250)	<input checked="" type="checkbox"/>
	TOTAL_DAYS	int	<input checked="" type="checkbox"/>
	HOURS_START	nvarchar(250)	<input checked="" type="checkbox"/>
	HOURS_END	nvarchar(250)	<input checked="" type="checkbox"/>
	IS_ACTIVE	bit	<input checked="" type="checkbox"/>
	NOTE	nvarchar(500)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(250)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-8. GOODS\_SHIFT

	Column Name	Data Type	Allow Nulls
PK	ID	bigint	<input type="checkbox"/>
	CARD_ID	bigint	<input checked="" type="checkbox"/>
	START_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	EXPIRE_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	[USED_TIME]	float	<input checked="" type="checkbox"/>
	EXTENSION_PERIOD	float	<input checked="" type="checkbox"/>
	TOTAL_TIME	float	<input checked="" type="checkbox"/>
	CARD_PRICE	bigint	<input checked="" type="checkbox"/>
	CUSTOMER_ID	bigint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-9. PER\_CUS\_LIST\_CARD

	Column Name	Data Type	Allow Nulls
✎	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(255)	<input checked="" type="checkbox"/>
	TRANS_DATE	nvarchar(255)	<input checked="" type="checkbox"/>
	CUSTOMER_ID	bigint	<input checked="" type="checkbox"/>
	TRANS_FORM	bigint	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


*Bảng 2-10. PER\_CUS\_TRANSACTION*

	Column Name	Data Type	Allow Nulls
✎	ID	nvarchar(36)	<input type="checkbox"/>
	USERNAME	nvarchar(256)	<input checked="" type="checkbox"/>
	FULLNAME	nvarchar(100)	<input checked="" type="checkbox"/>
	PASSWORDHASH	nvarchar(MAX)	<input checked="" type="checkbox"/>
	EMPLOYEE_ID	bigint	<input checked="" type="checkbox"/>
	GROUP_ID	bigint	<input checked="" type="checkbox"/>
	IS_ADMIN	bit	<input checked="" type="checkbox"/>
	IS_ROOT	bit	<input checked="" type="checkbox"/>
	AVATAR	nvarchar(150)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(450)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	IS_LOCK	bit	<input checked="" type="checkbox"/>
	DECENTRALIZATION	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


*Bảng 2-11 . SYS\_USER*

	Column Name	Data Type	Allow Nulls
PK	ID	bigint	<input type="checkbox"/>
	AVATAR	nvarchar(1000)	<input checked="" type="checkbox"/>
	CODE	nvarchar(50)	<input checked="" type="checkbox"/>
	CUSTOMER_CLASS_ID	bigint	<input checked="" type="checkbox"/>
	FIRST_NAME	nvarchar(50)	<input checked="" type="checkbox"/>
	LAST_NAME	nvarchar(50)	<input checked="" type="checkbox"/>
	FULL_NAME	nvarchar(255)	<input checked="" type="checkbox"/>
	BIRTH_DATE	nvarchar(255)	<input checked="" type="checkbox"/>
	GENDER_ID	bigint	<input checked="" type="checkbox"/>
	ADDRESS	nvarchar(500)	<input checked="" type="checkbox"/>
	PHONE_NUMBER	nvarchar(50)	<input checked="" type="checkbox"/>
	EMAIL	nvarchar(255)	<input checked="" type="checkbox"/>
	NATIVE_ID	bigint	<input checked="" type="checkbox"/>
	RELIGION_ID	bigint	<input checked="" type="checkbox"/>
	BANK_ID	bigint	<input checked="" type="checkbox"/>
	BANK_BRANCH	bigint	<input checked="" type="checkbox"/>
	BANK_NO	nvarchar(255)	<input checked="" type="checkbox"/>
	IS_GUEST_PASS	bit	<input checked="" type="checkbox"/>
	JOIN_DATE	nvarchar(255)	<input checked="" type="checkbox"/>
	HEIGHT	nvarchar(10)	<input checked="" type="checkbox"/>
	WEIGHT	nvarchar(10)	<input checked="" type="checkbox"/>
	CARD_ID	bigint	<input checked="" type="checkbox"/>
	NOTE	nvarchar(500)	<input checked="" type="checkbox"/>
	IS_ACTIVE	bit	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	EXPIRE_DATE	nvarchar(255)	<input checked="" type="checkbox"/>
	GYM_PACKAGE_ID	bigint	<input checked="" type="checkbox"/>
	PER_PT_ID	bigint	<input checked="" type="checkbox"/>
	PER_SALE_ID	bigint	<input checked="" type="checkbox"/>
	ID_NO	nvarchar(255)	<input checked="" type="checkbox"/>
	STATUS_ID	bigint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 2-12. PER\_CUSTOMER

	Column Name	Data Type	Allow Nulls
	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(50)	<input checked="" type="checkbox"/>
	NAME	nvarchar(200)	<input checked="" type="checkbox"/>
	TYPE_ID	bigint	<input checked="" type="checkbox"/>
	ORDERS	int	<input checked="" type="checkbox"/>
	IS_ACTIVE	bit	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	NOTE	nvarchar(255)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Bảng 2-13 . SYS\_LIST\_OTHER*

	Column Name	Data Type	Allow Nulls
	ID	bigint	<input type="checkbox"/>
	CODE	nvarchar(50)	<input checked="" type="checkbox"/>
	NAME	nvarchar(200)	<input checked="" type="checkbox"/>
	ORDERS	int	<input checked="" type="checkbox"/>
	CREATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	CREATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	UPDATED_DATE	datetime2(7)	<input checked="" type="checkbox"/>
	UPDATED_BY	nvarchar(255)	<input checked="" type="checkbox"/>
	NOTE	nvarchar(500)	<input checked="" type="checkbox"/>
	IS_SYSTEM	bit	<input checked="" type="checkbox"/>
	IS_ACTIVE	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Bảng 2-14 . SYS\_LIST\_OTHER\_TYPE*

### 2.2.3. Quan hệ các bảng

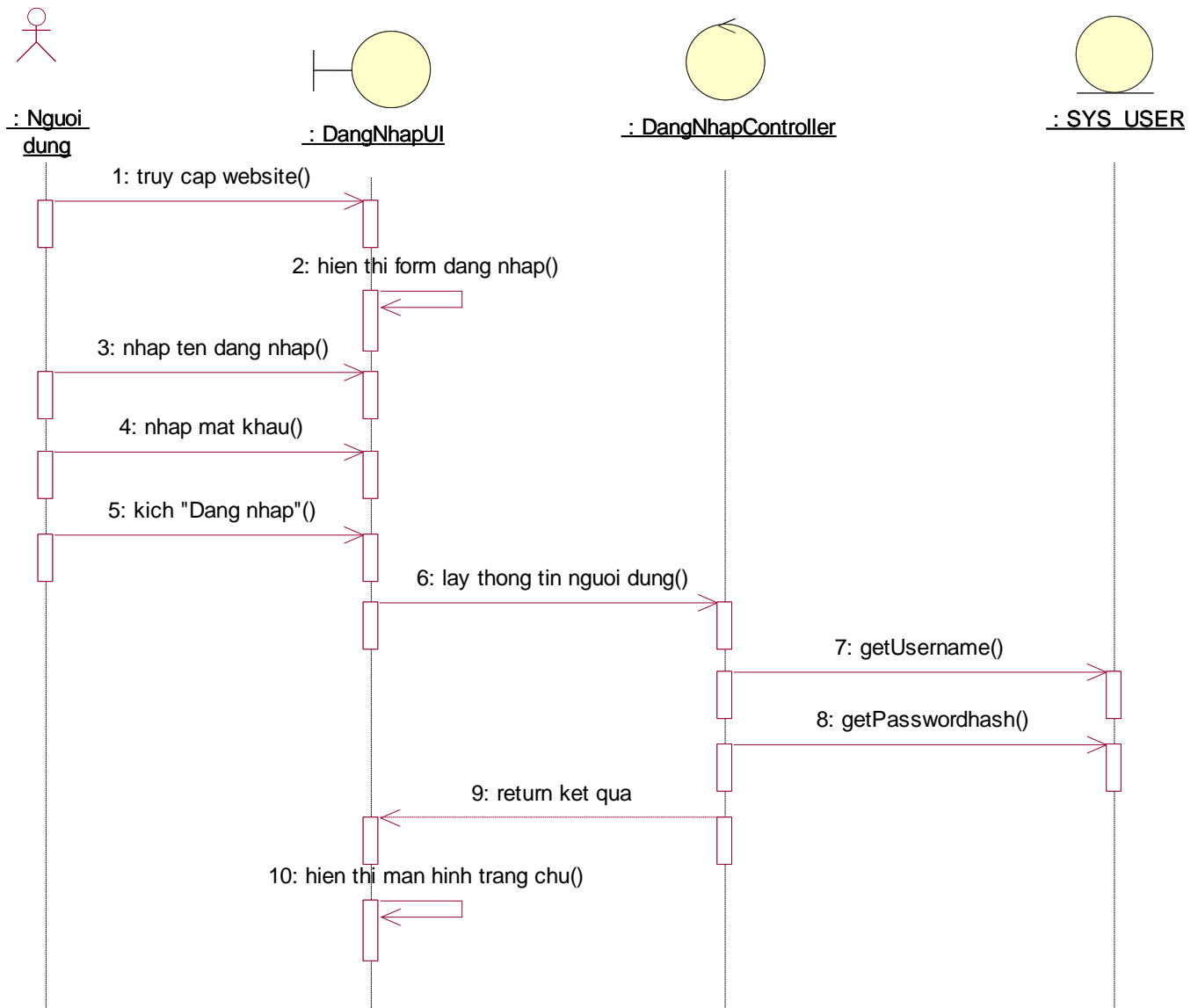


Hình 3: Hình ảnh quan hệ giữa các bảng

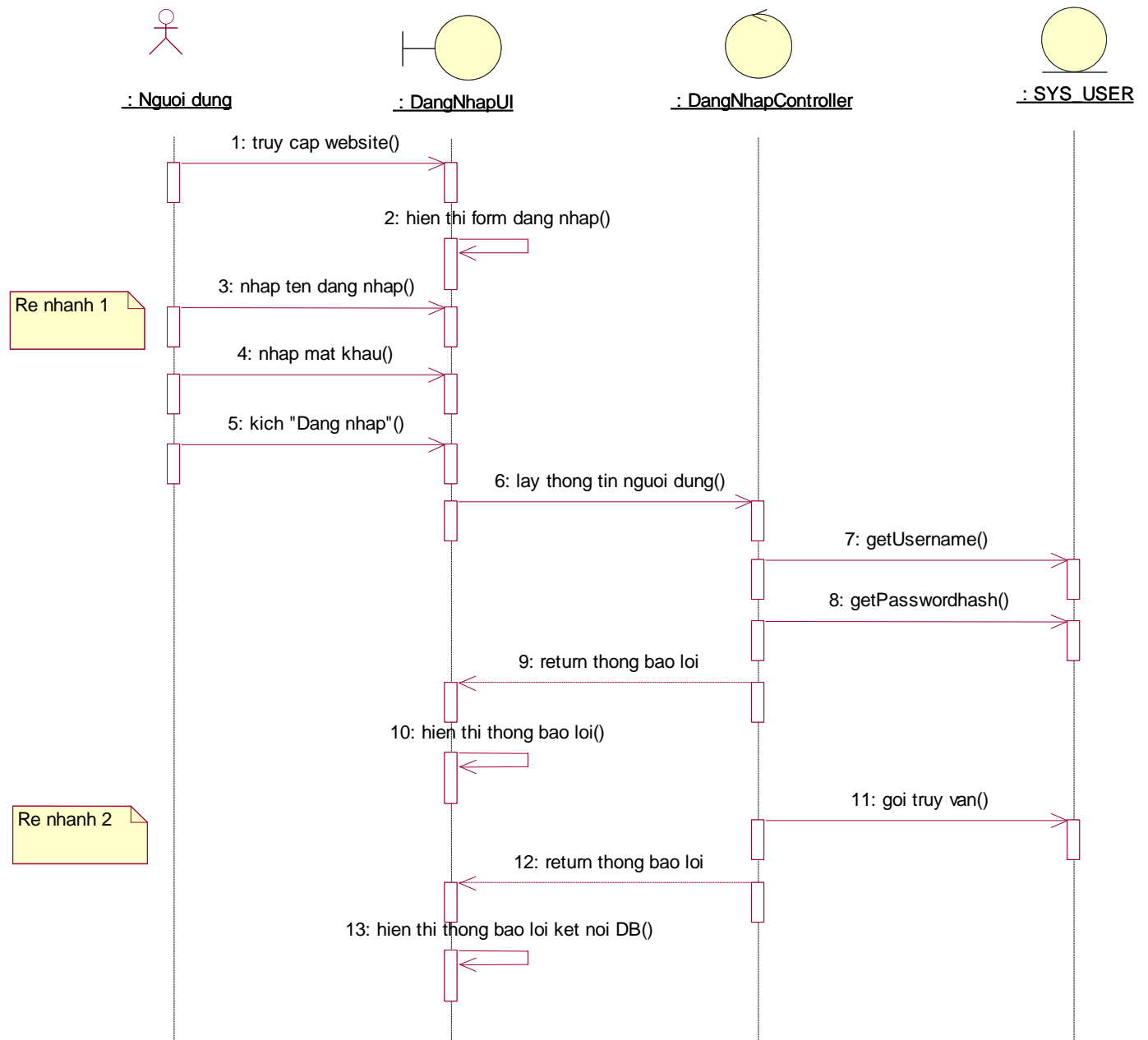
## 2.3. Thiết kế các thành phần của hệ thống

### 2.3.1. Use case Đăng nhập

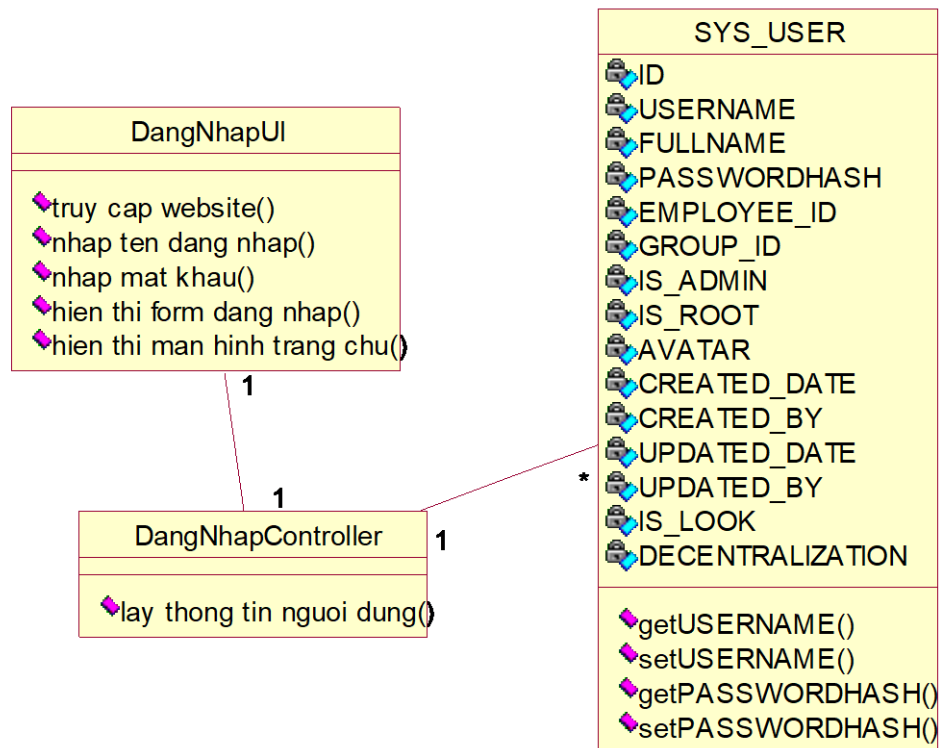
#### 2.3.1.1. Biểu đồ trình tự phân luồng cơ bản



### 2.3.1.2. Biểu đồ trình tự phần luồng rẽ nhánh



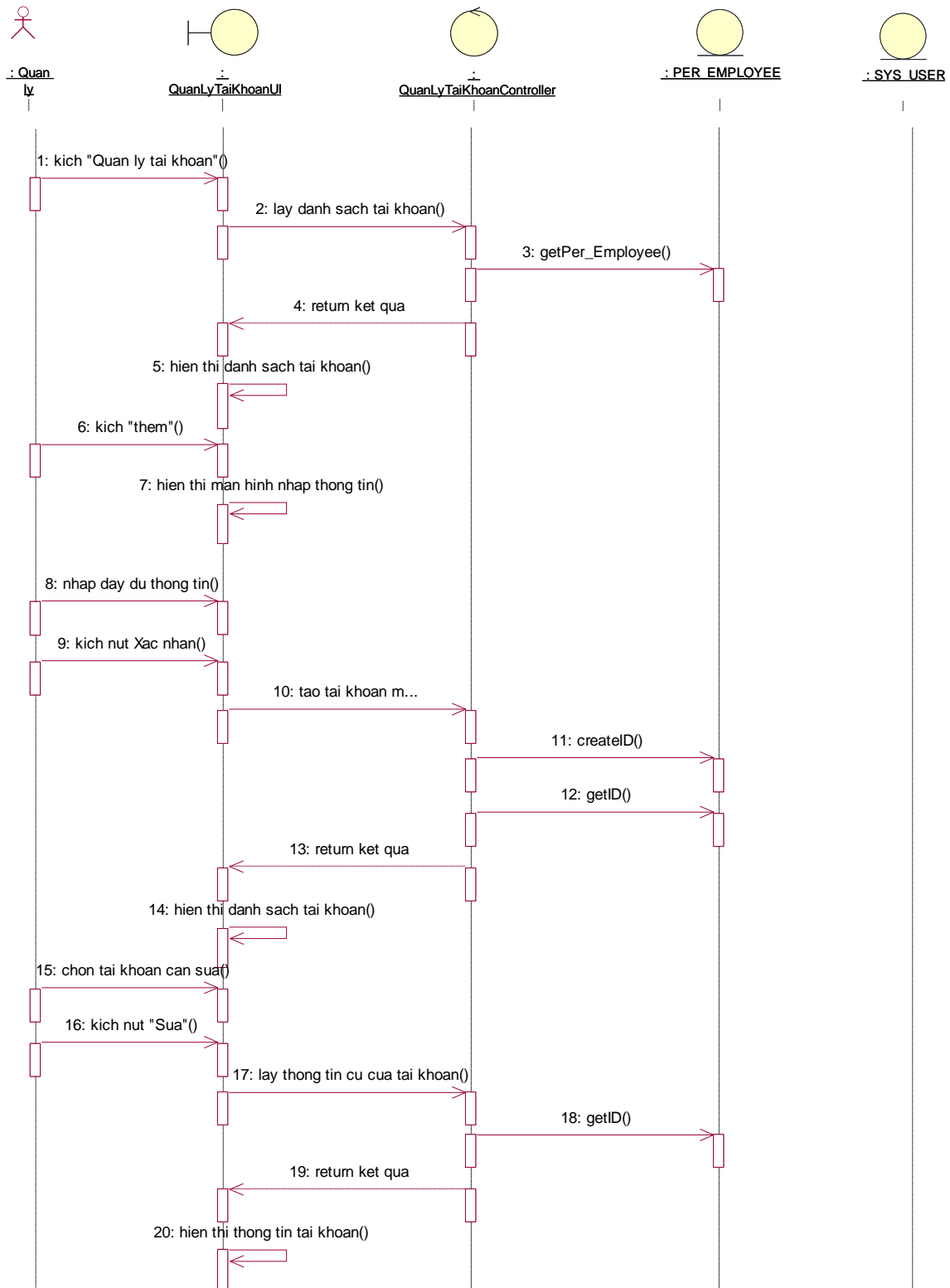
### 2.3.1.3. Biểu đồ lớp phân tích

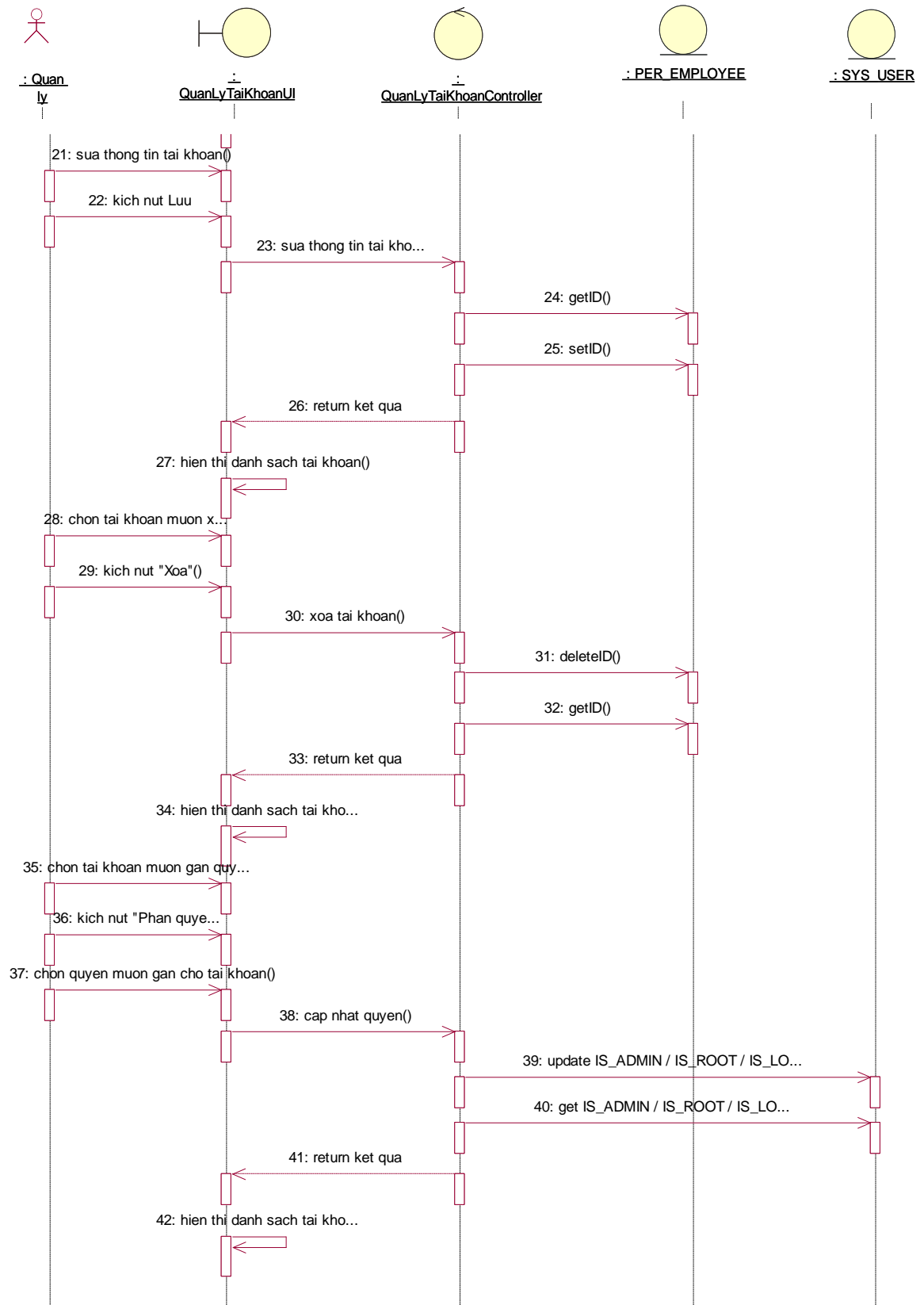




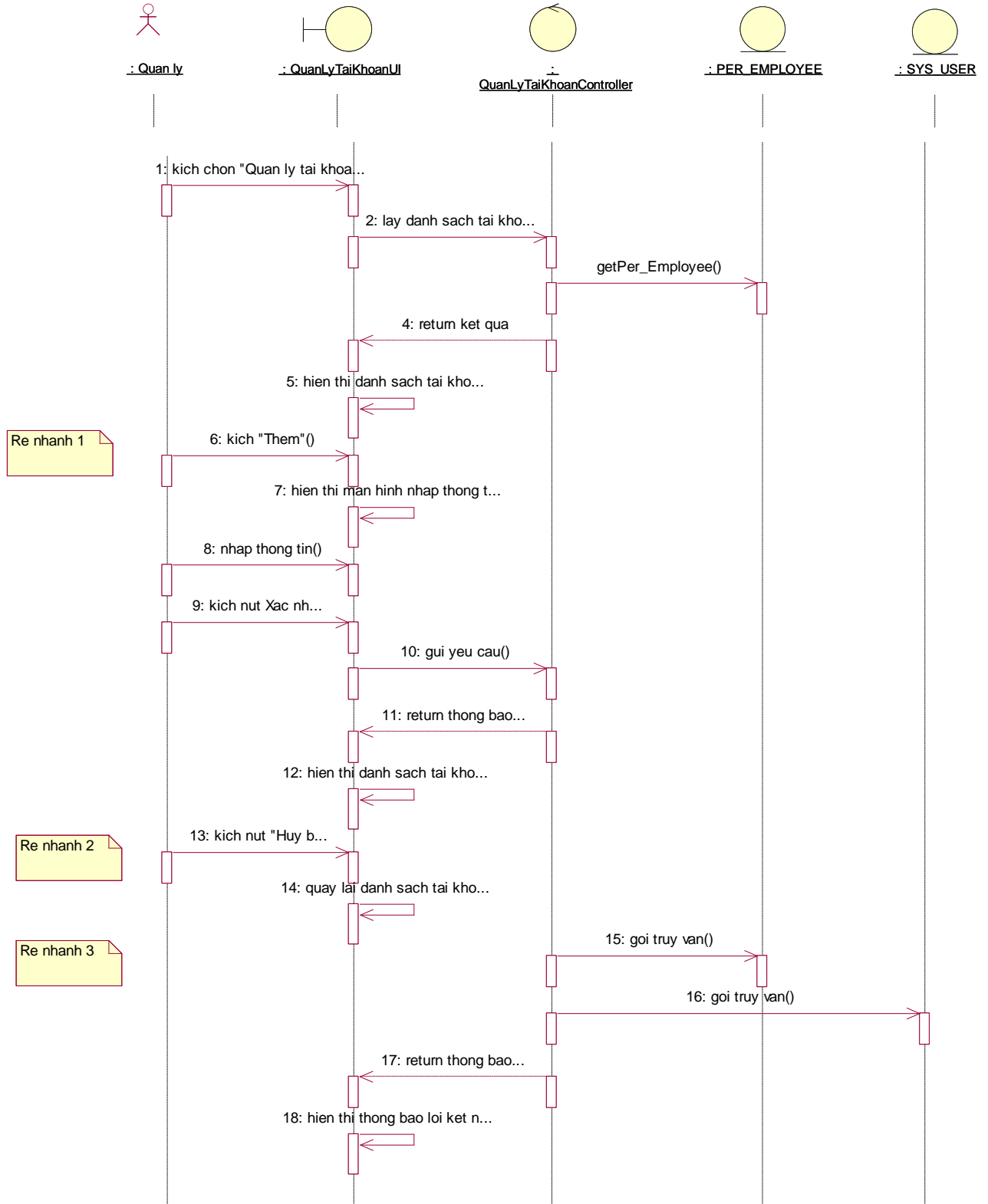
## 2.3.2. Use case Quản lý tài khoản

### 2.3.2.1. Biểu đồ trình tự phân luồng cơ bản

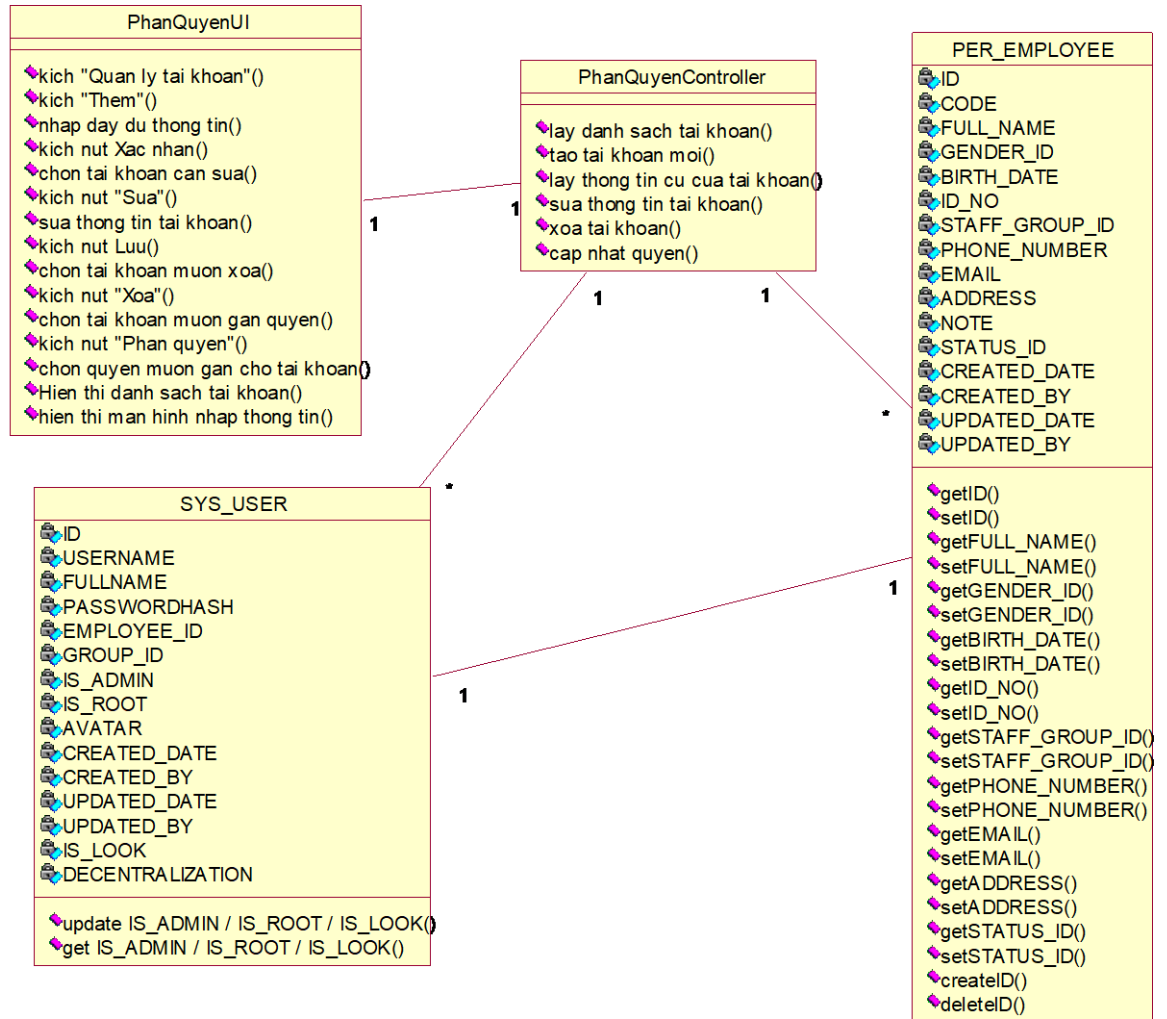




### 2.3.2.2. Biểu đồ trình tự phân luồng rẽ nhánh

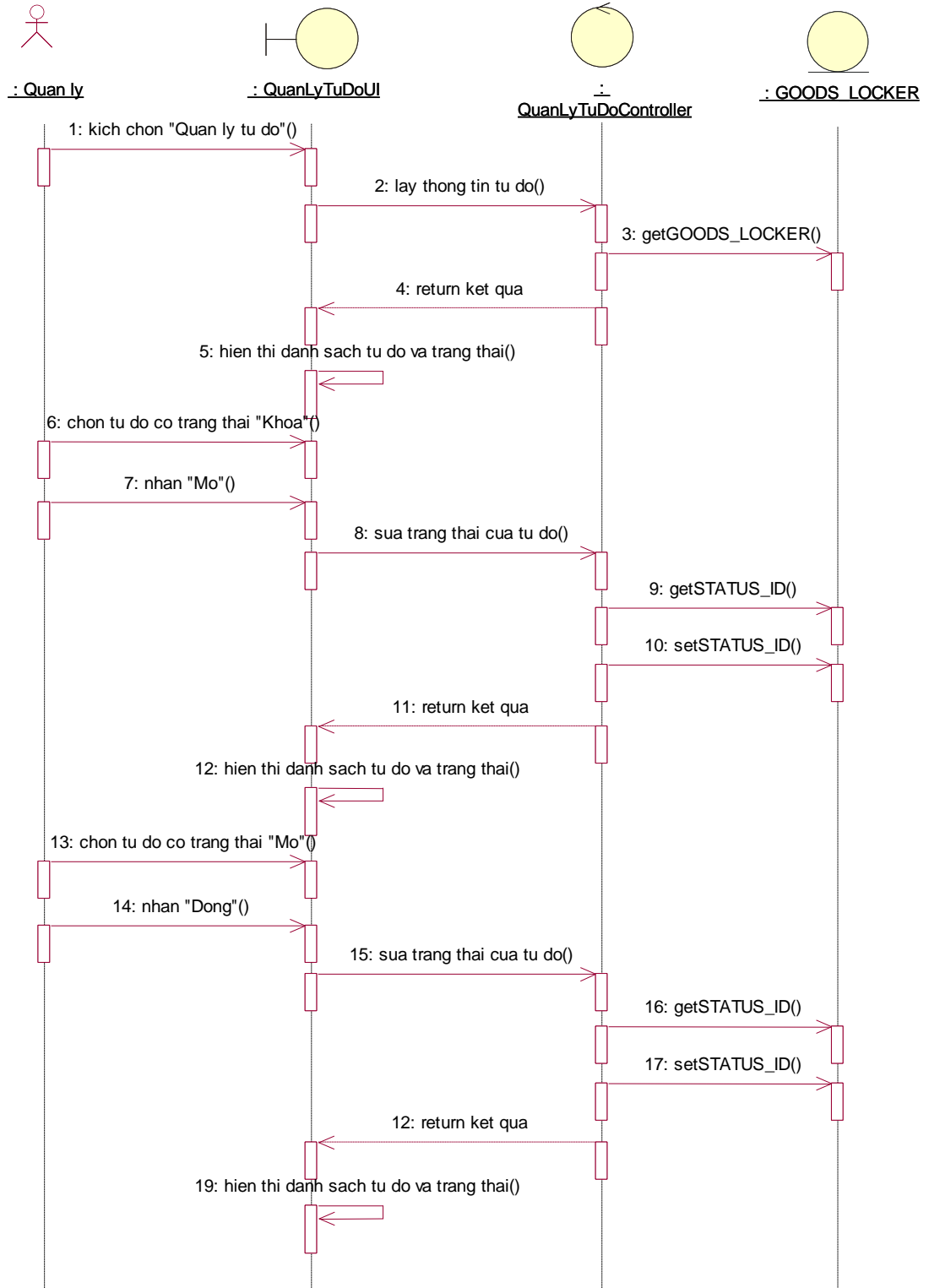


### 2.3.2.3. Biểu đồ lớp phân tích

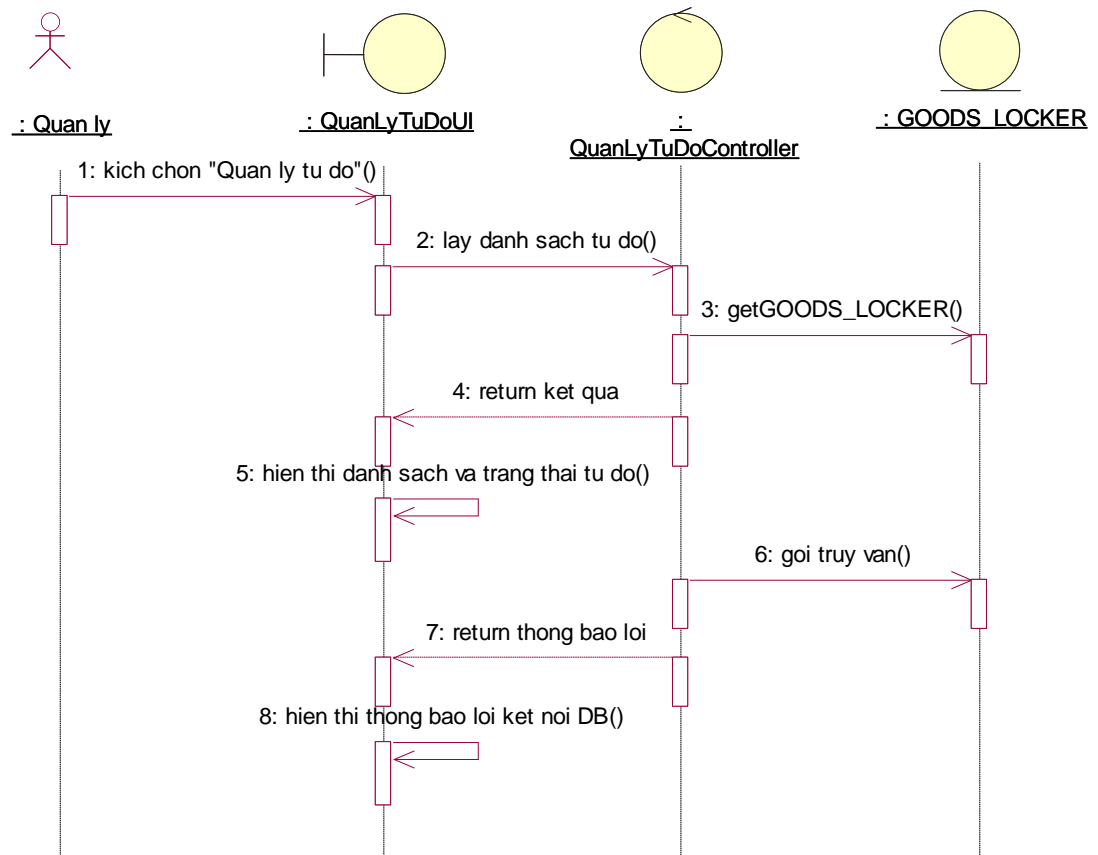


### 2.3.3. Use case Quản lý tủ đồ

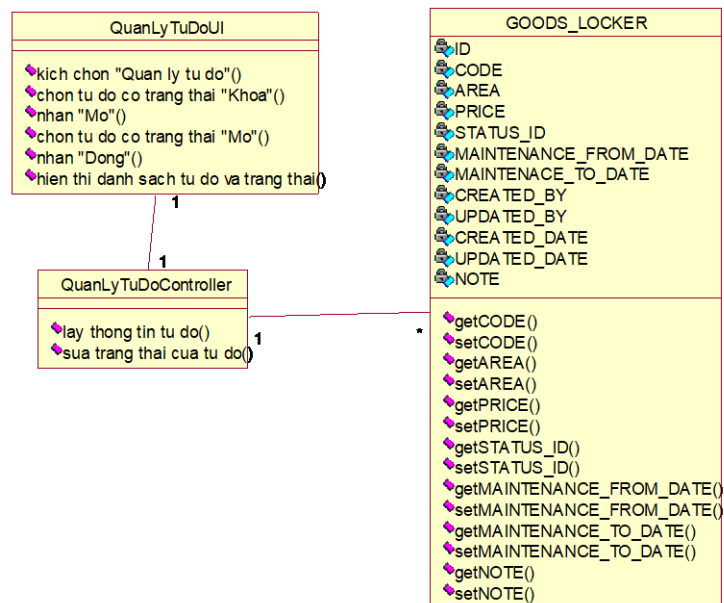
#### 2.3.3.1. Biểu đồ trình tự phân luồng cơ bản



### 2.3.3.2. Biểu đồ trình tự phần luồng rẽ nhánh

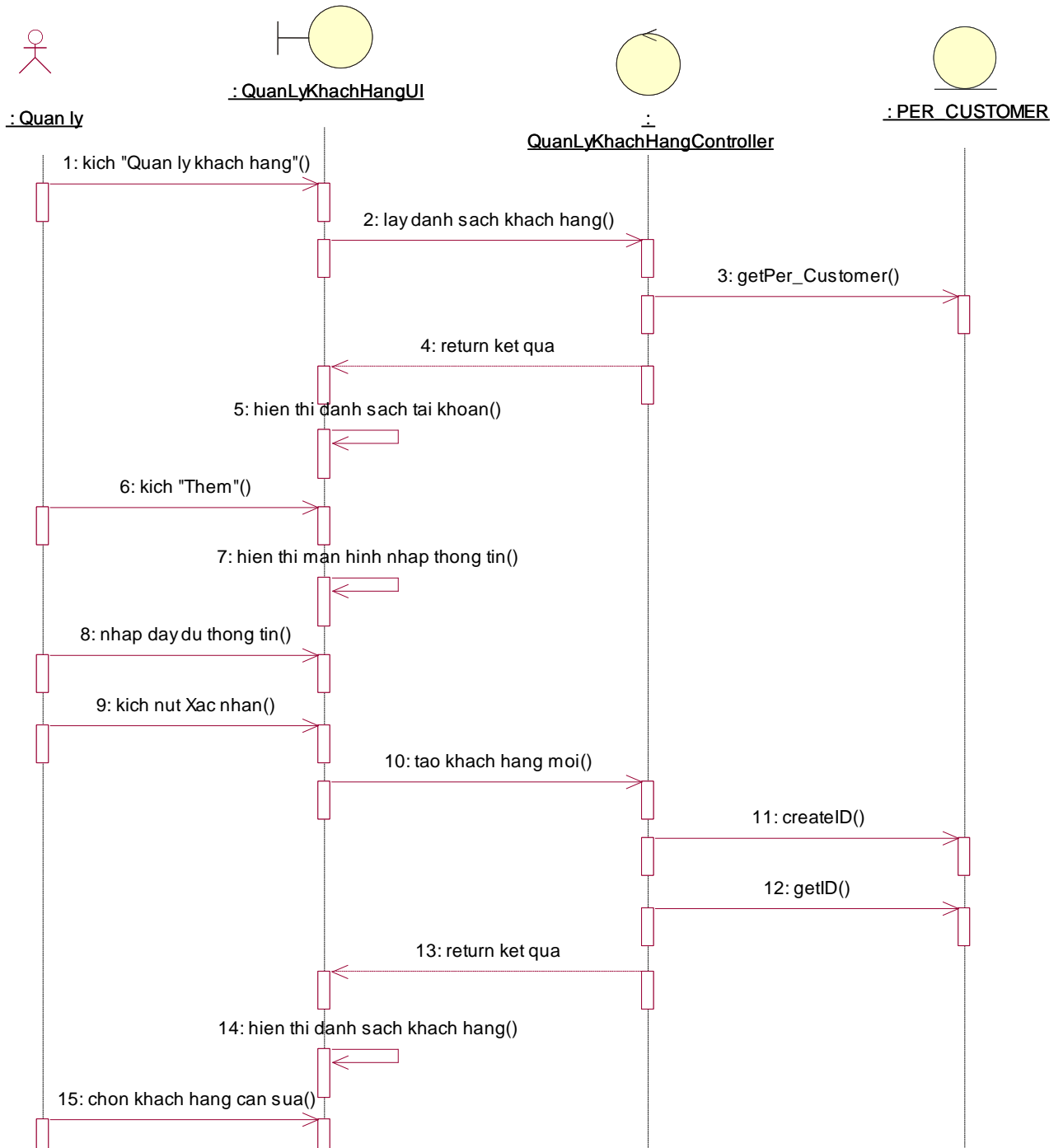


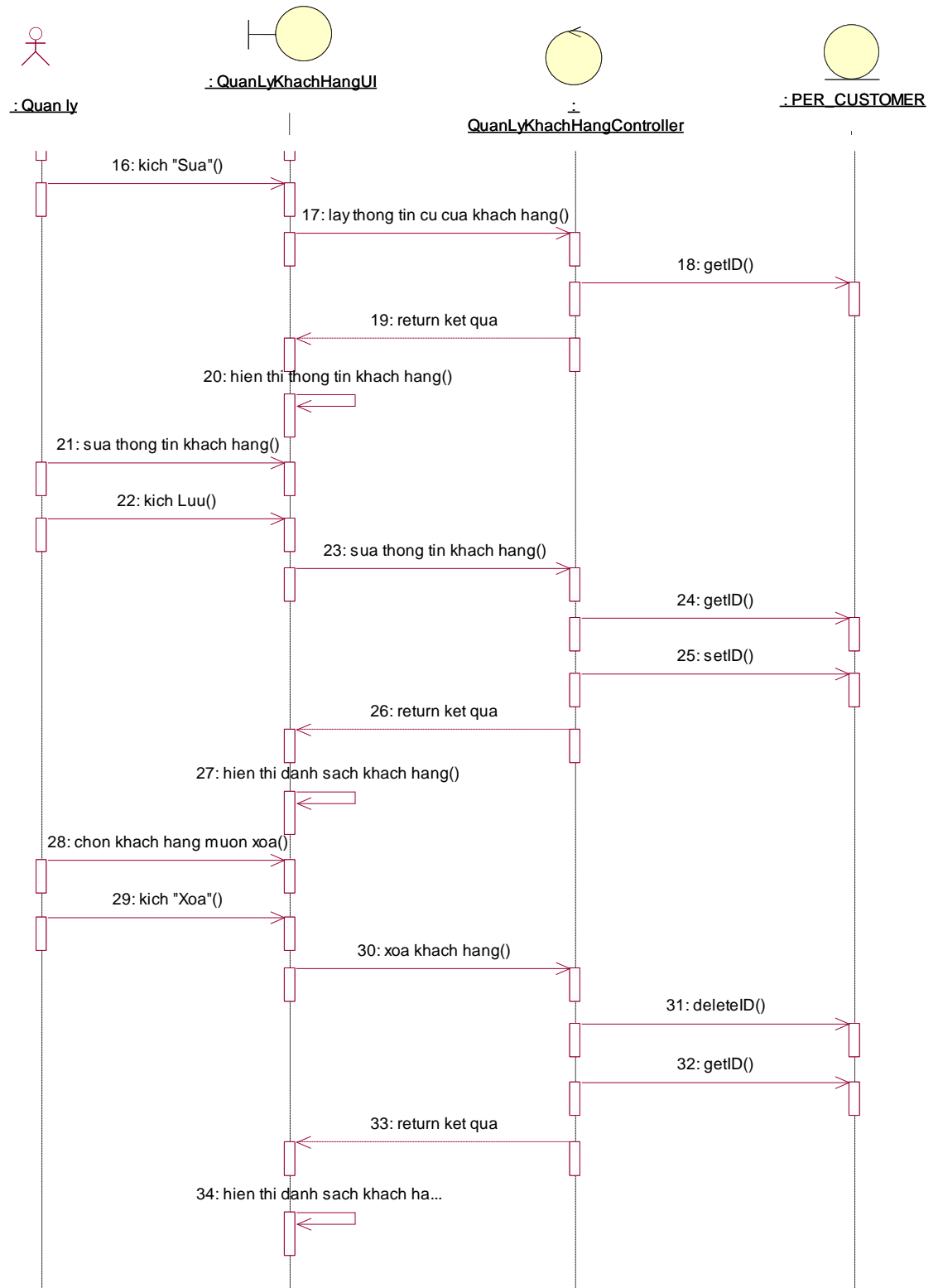
### 2.3.3.3. Biểu đồ lớp phân tích



## 2.3.4. Use case Quản lý khách hàng

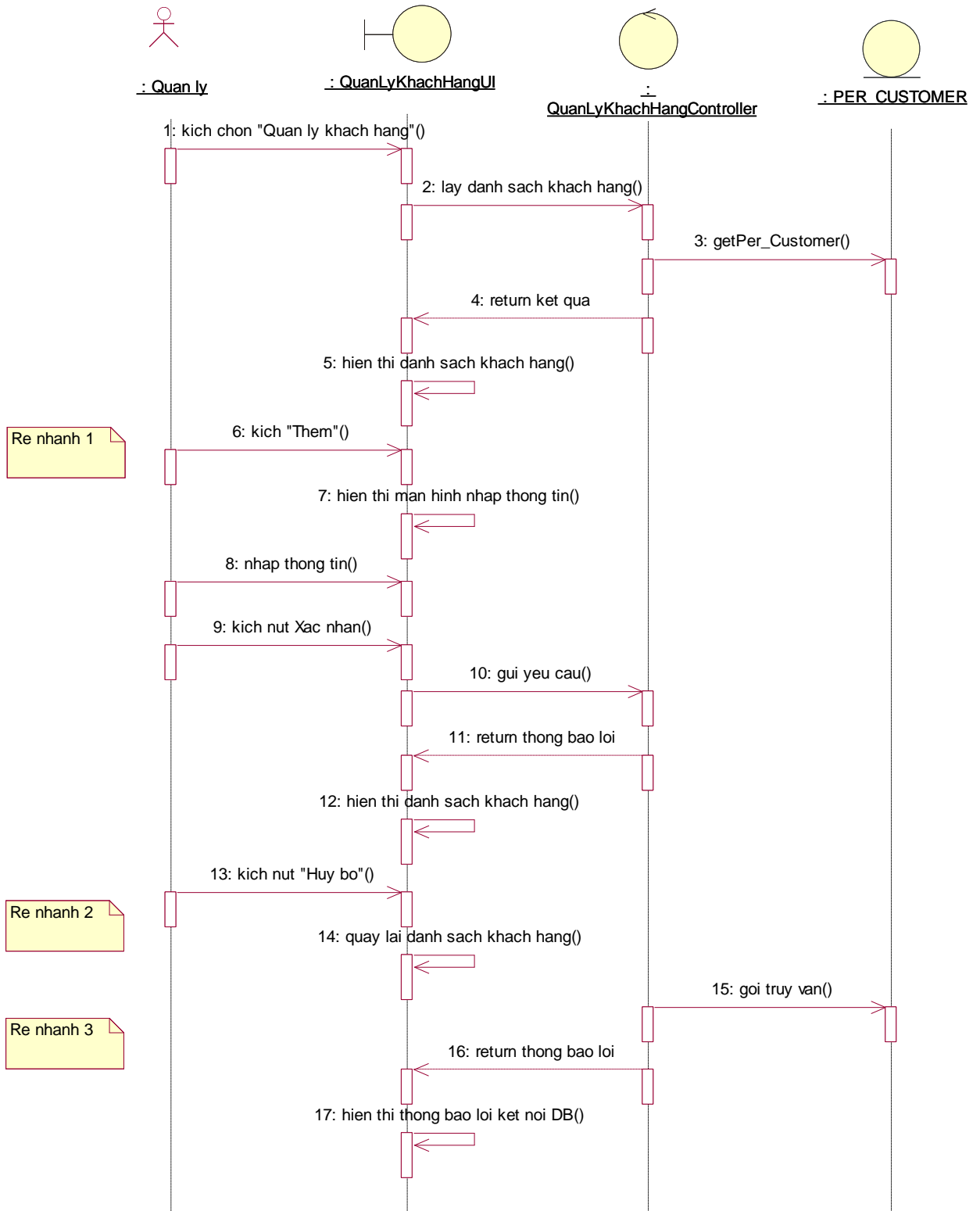
### 2.3.4.1. Biểu đồ trình tự phân luồng cơ bản



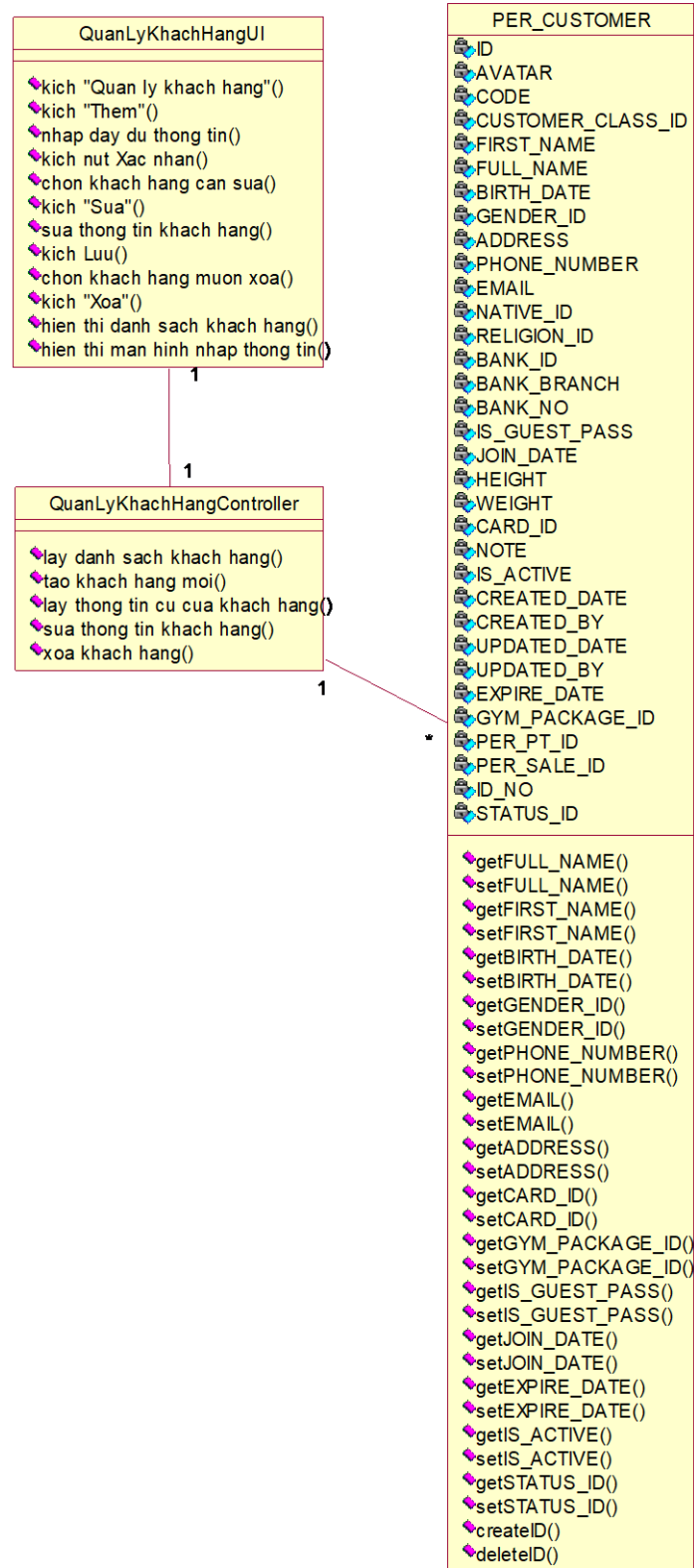




### 2.3.4.2. Biểu đồ trình tự phần luồng rẽ nhánh

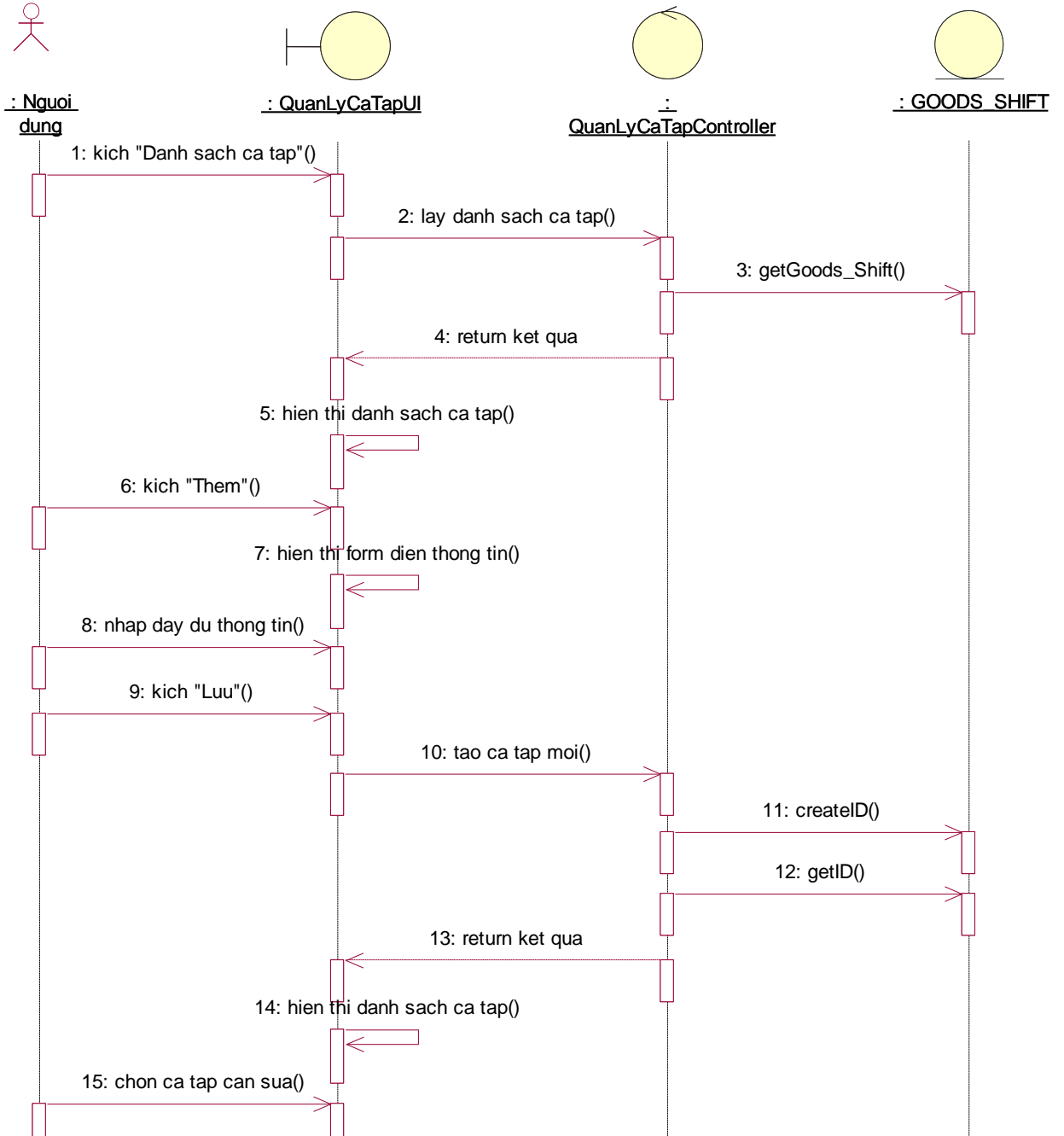


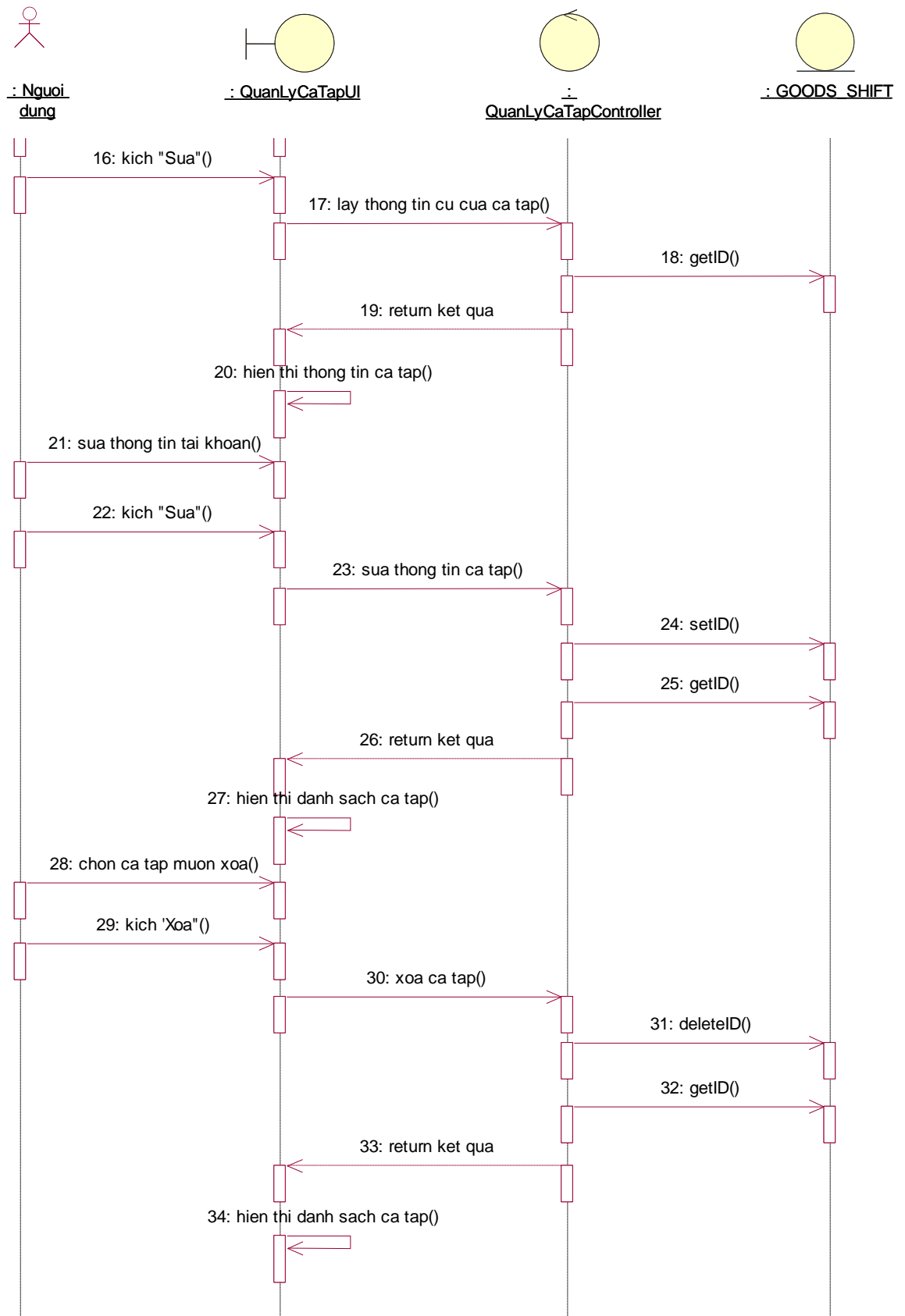
### 2.3.4.3. Biểu đồ lớp phân tích



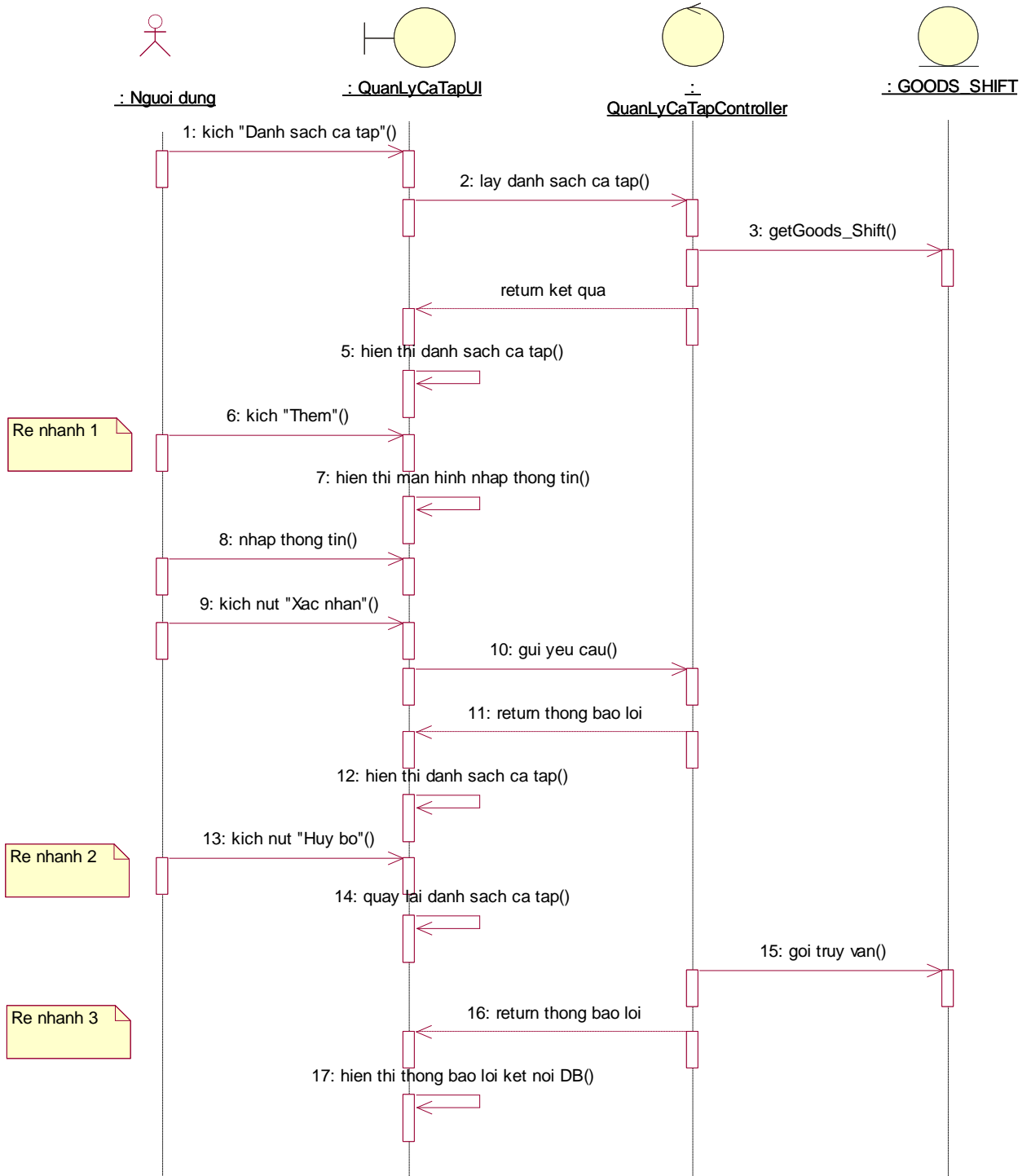
### 2.3.5. Use case Quản lý danh sách ca tập

#### 2.3.5.1. Biểu đồ trình tự phân luồng cơ bản

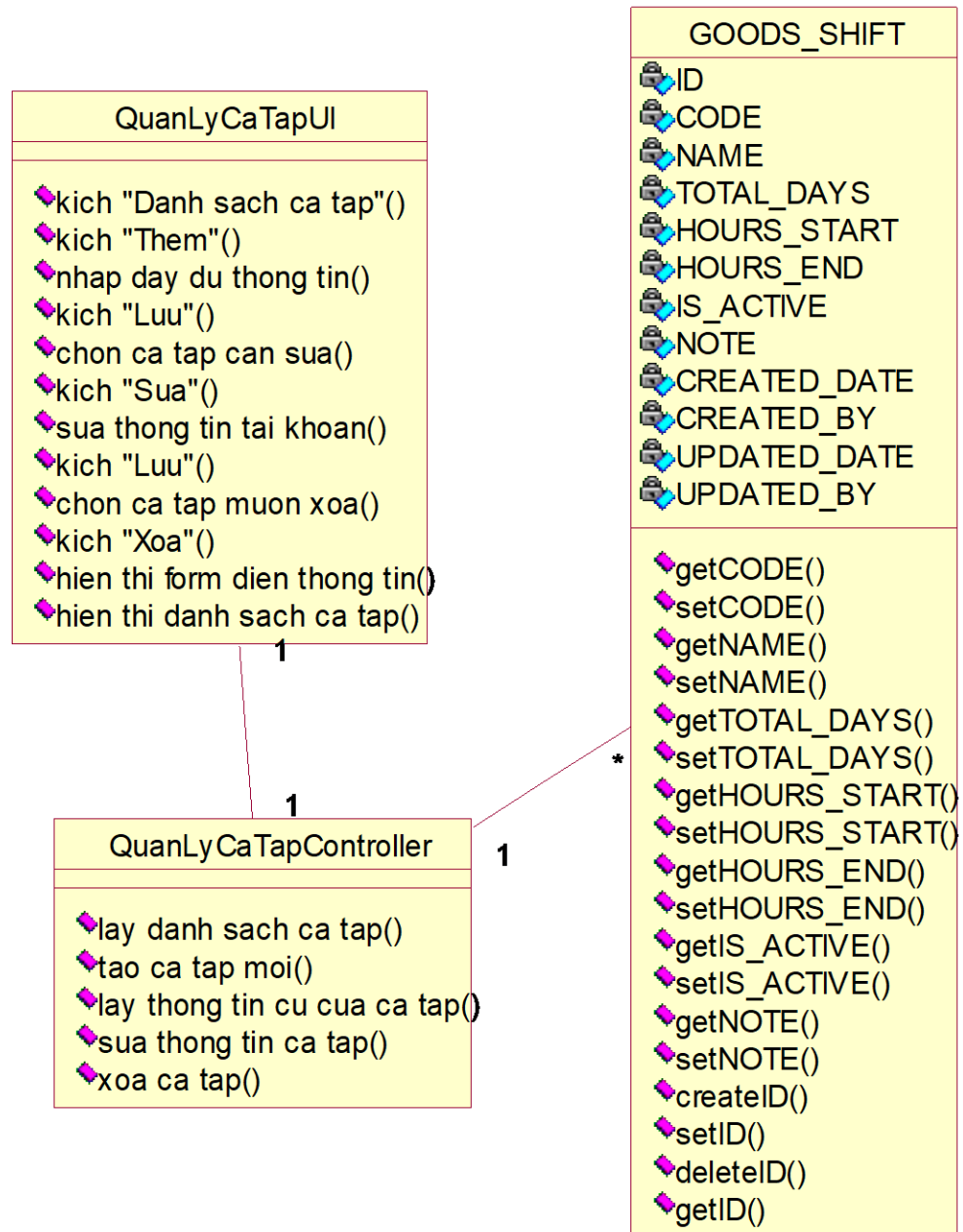




### 2.3.5.2. Biểu đồ trình tự phân luồng rẽ nhánh

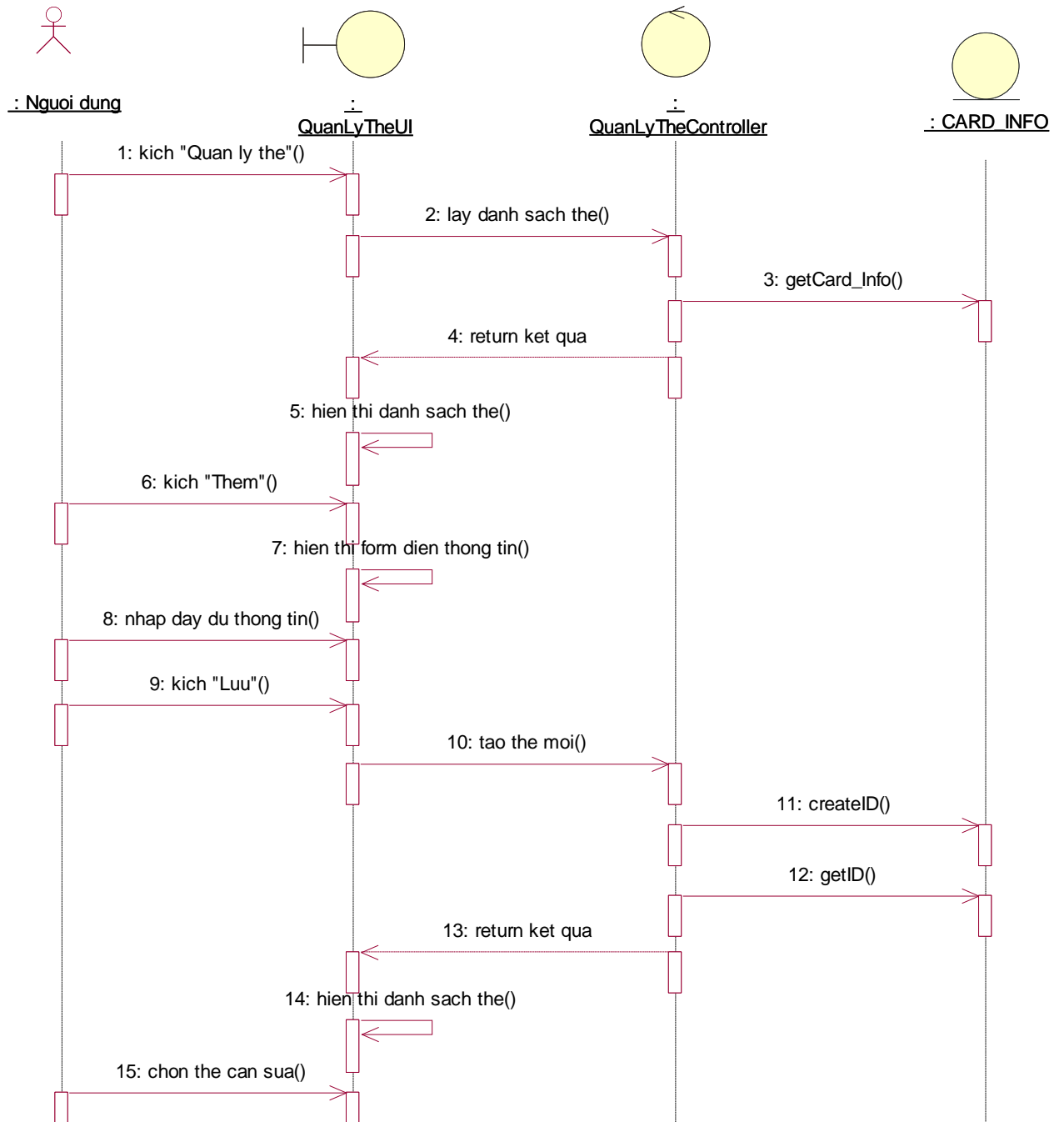


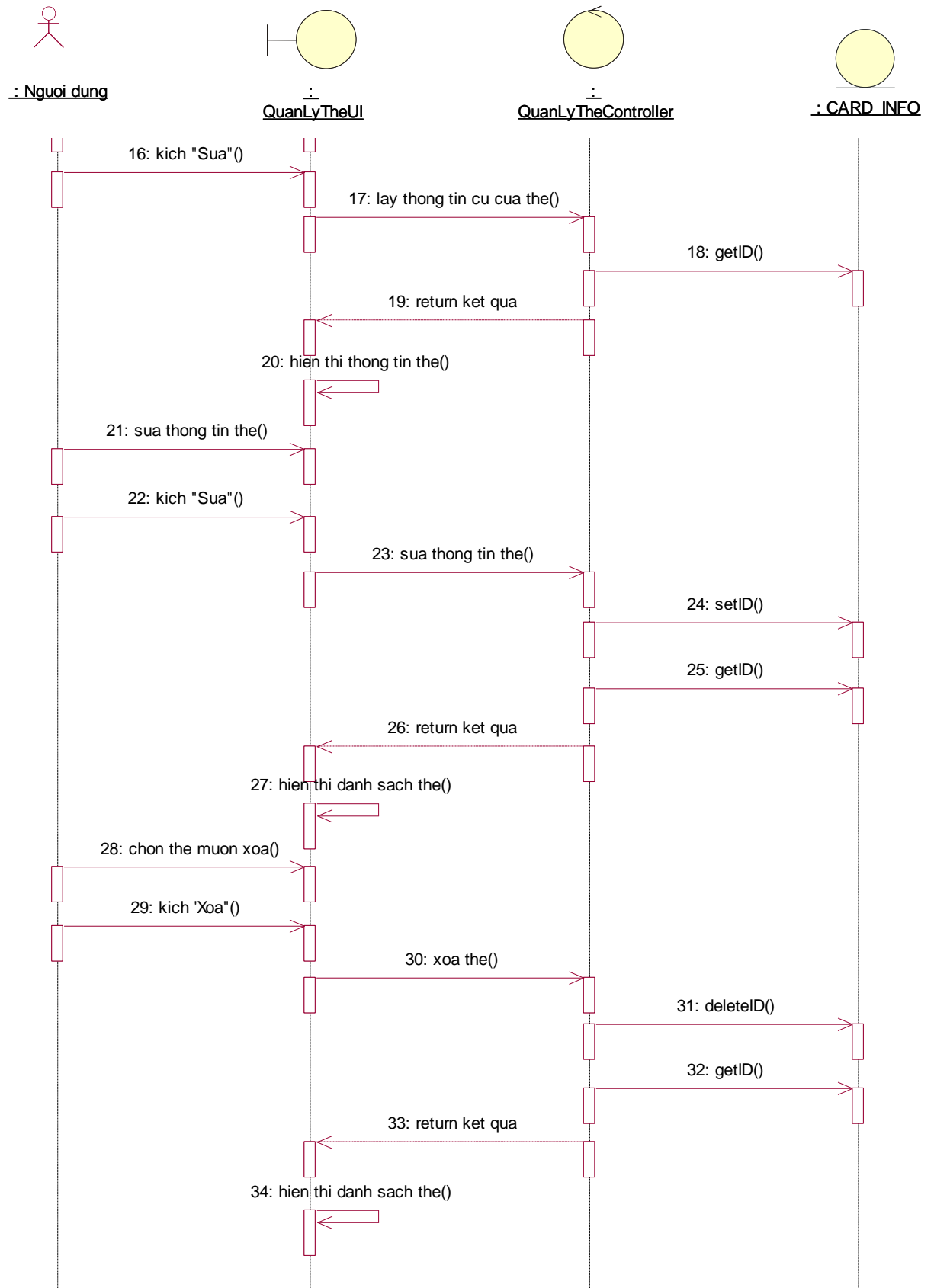
### 2.3.5.3. Biểu đồ lớp phân tích



## 2.3.6. Use case Quản lý thẻ

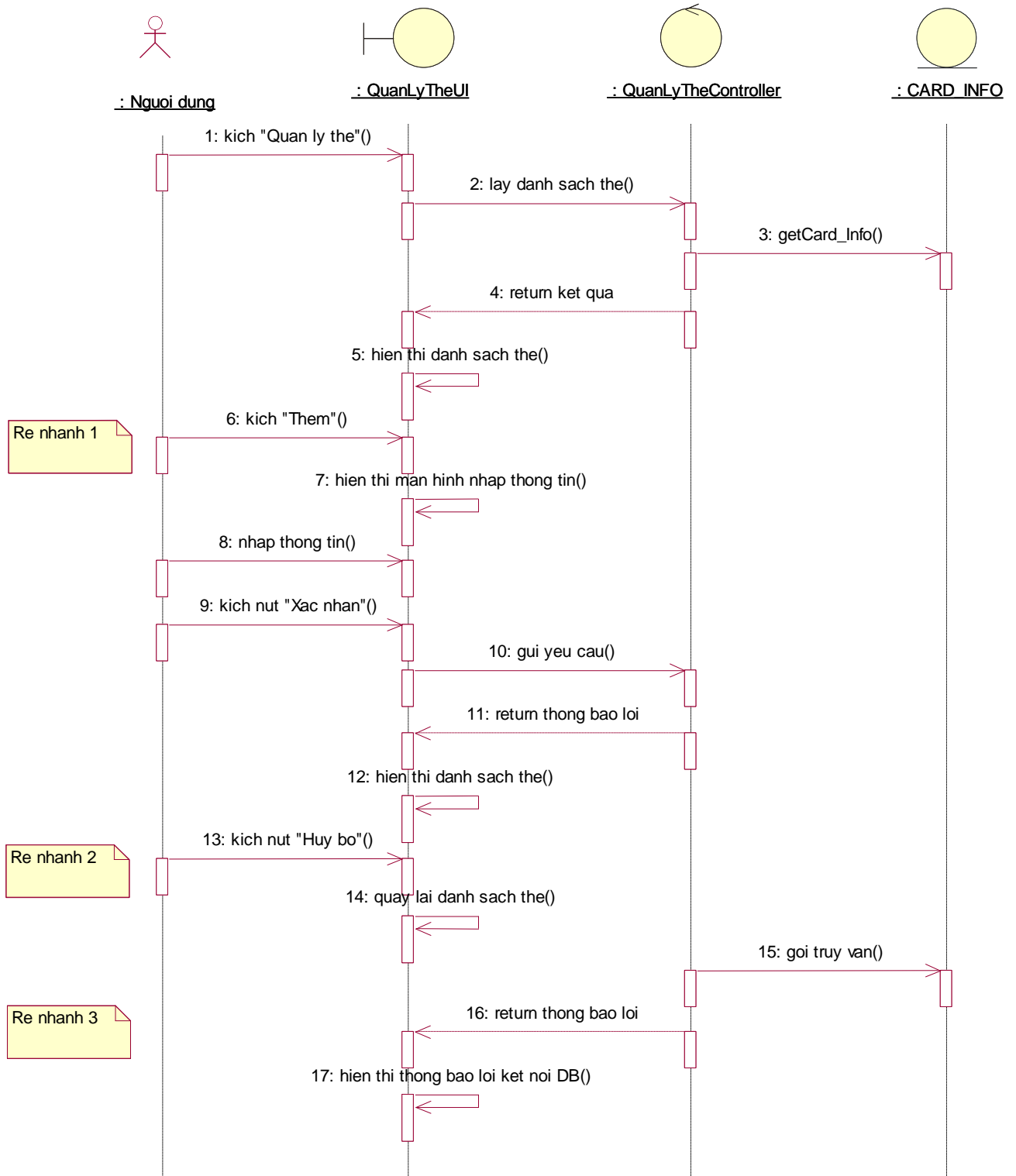
### 2.3.6.1. Biểu đồ trình tự phân luồng cơ bản



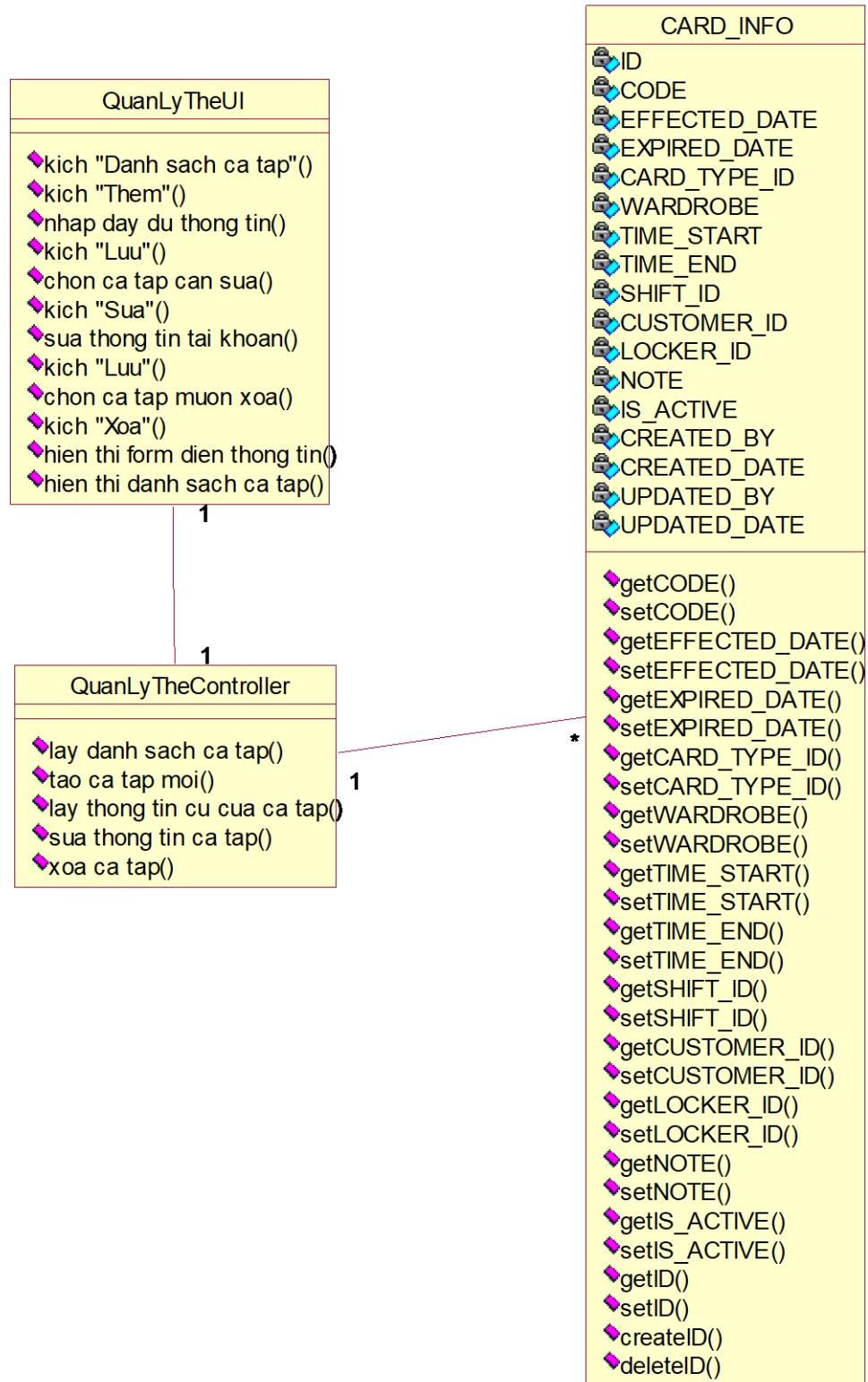




### 2.3.6.2. Biểu đồ trình tự phần luồng rẽ nhánh



### 2.3.6.3. Biểu đồ lớp phân tích





## 2.4. Thiết kế giao diện hình dung màn hình

### 2.4.1. Màn hình quản trị

Gym Help

← → ↻ https://gym-help.web.app/home

**Quản trị**  Tên User


 Xin chào

Menu chính

Dashboard

Thành phần

Quản lý khách hàng  
Quản lý nhân viên  
Quản lý vật chất  
Quản lý thẻ  
Báo cáo  
Hệ thống


**Banner** 

Hình 4: Thiết kế màn hình quản trị

### 2.4.2. Màn hình Profile


Gym Help


← → ↻ https://gym-help.web.app/home

**Thông tin nhân viên** 


Mã nhân viên

Họ và tên

Giới tính  

Ngày sinh  

Số CMND/CCCD

Nhóm nhân viên  

Số điện thoại

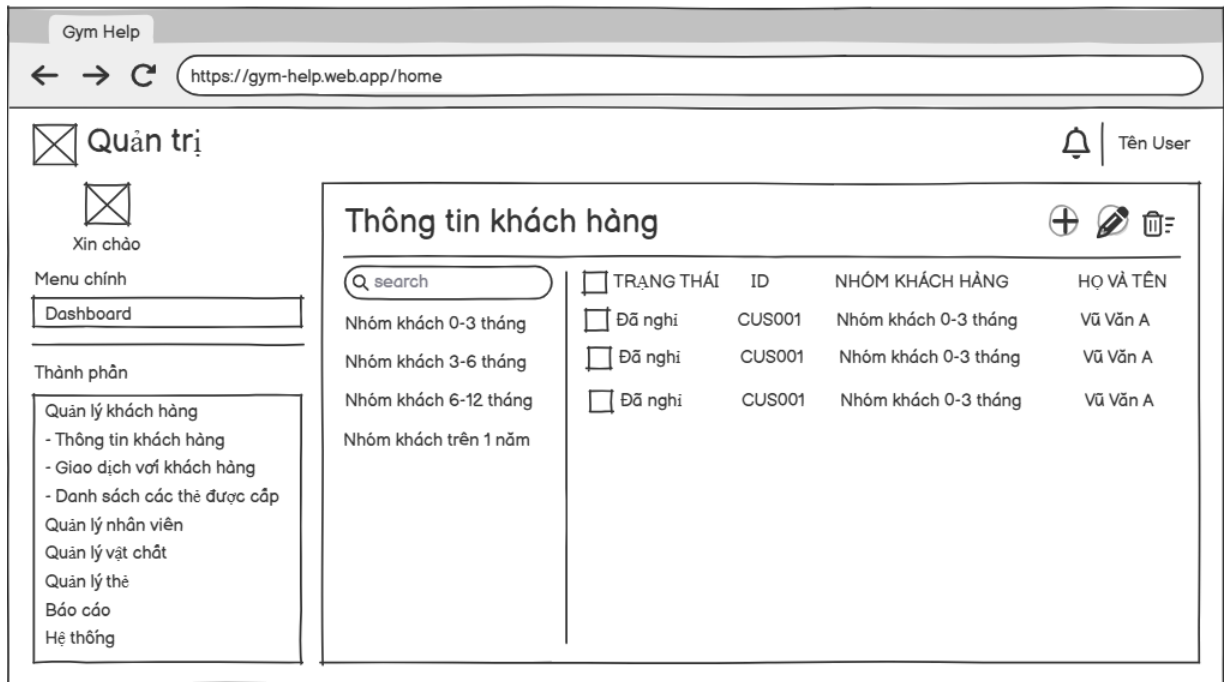
Email cá nhân

Địa chỉ

Ghi chú

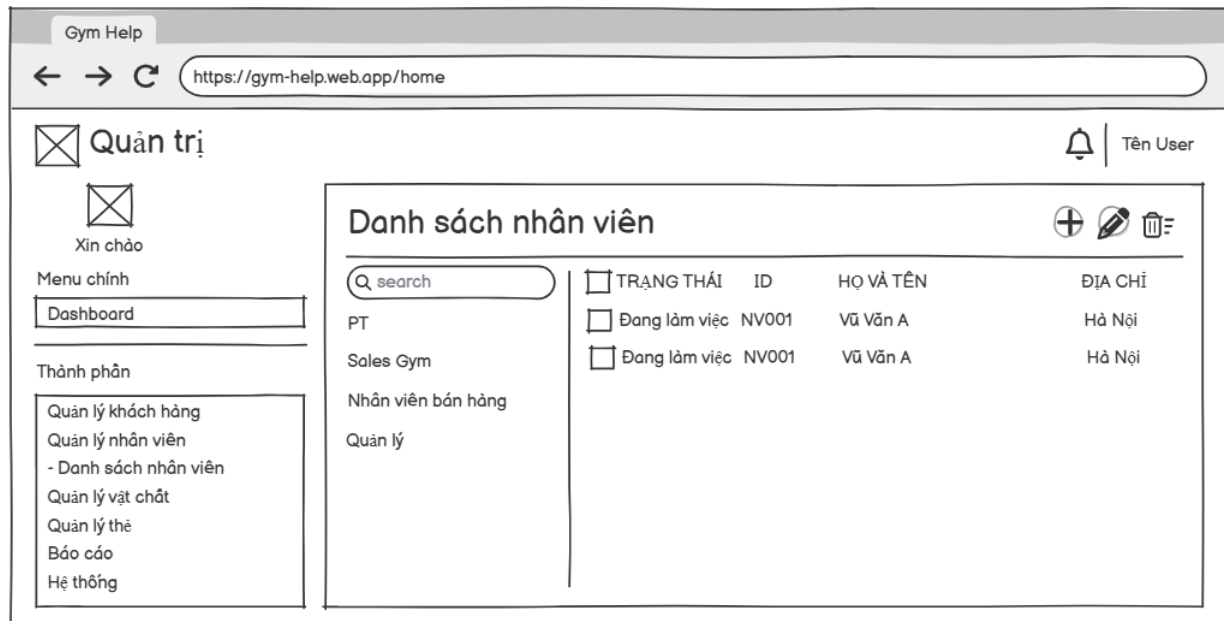
Hình 5: Thiết kế màn hình Profile

### 2.4.3. Màn hình Quản lý khách hàng



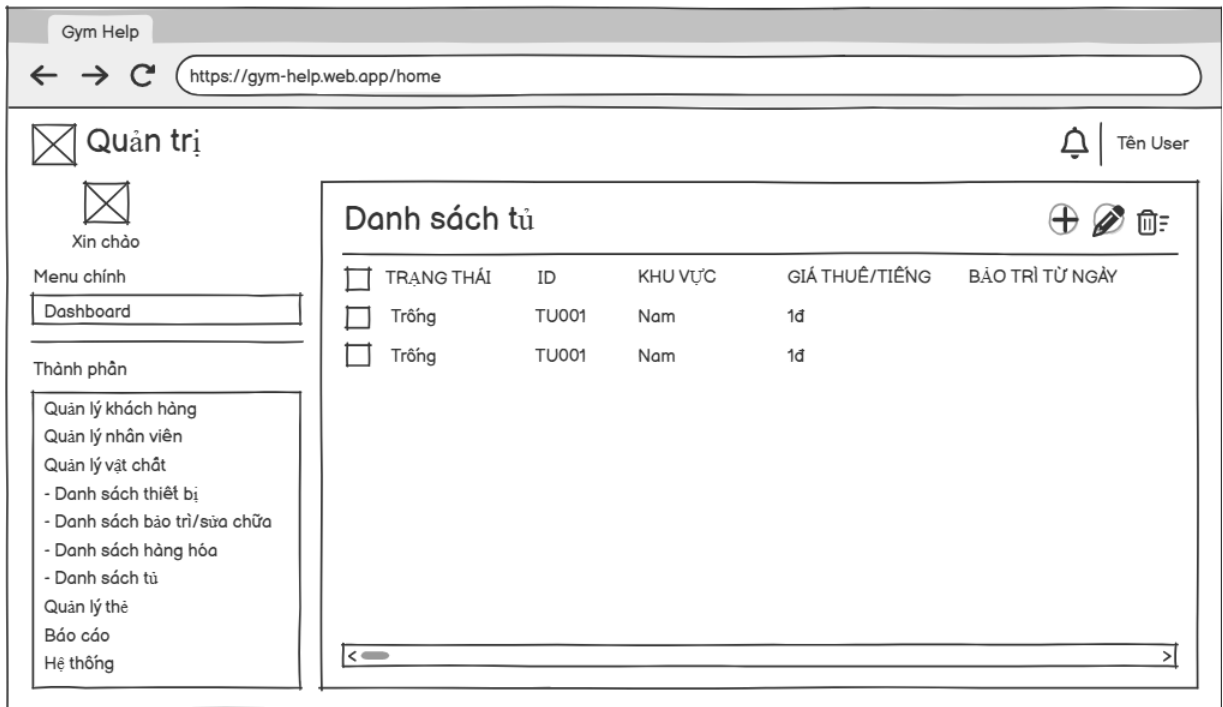
Hình 6: Thiết kế màn hình Quản lý khách hàng

### 2.4.4. Màn hình Quản lý nhân viên



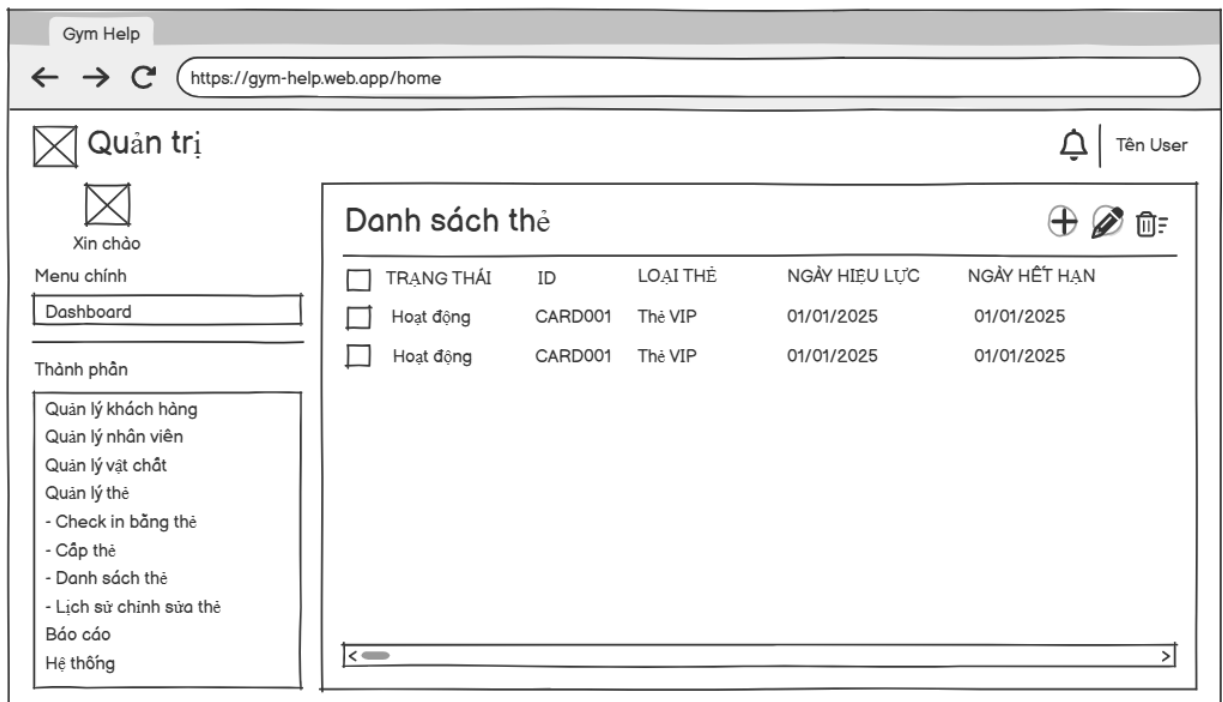
Hình 7: Thiết kế màn hình Quản lý nhân viên

### 2.4.5. Màn hình Quản lý vật chất



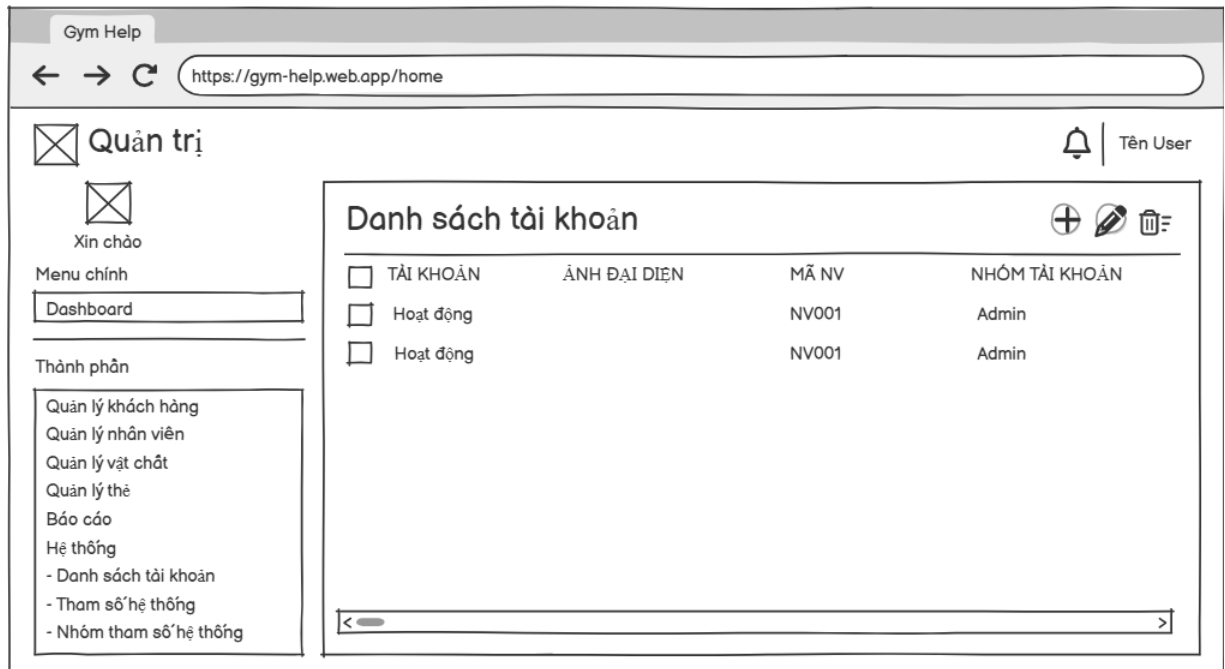
Hình 8: Thiết kế màn hình Quản lý vật chất

### 2.4.6. Màn hình Quản lý thẻ



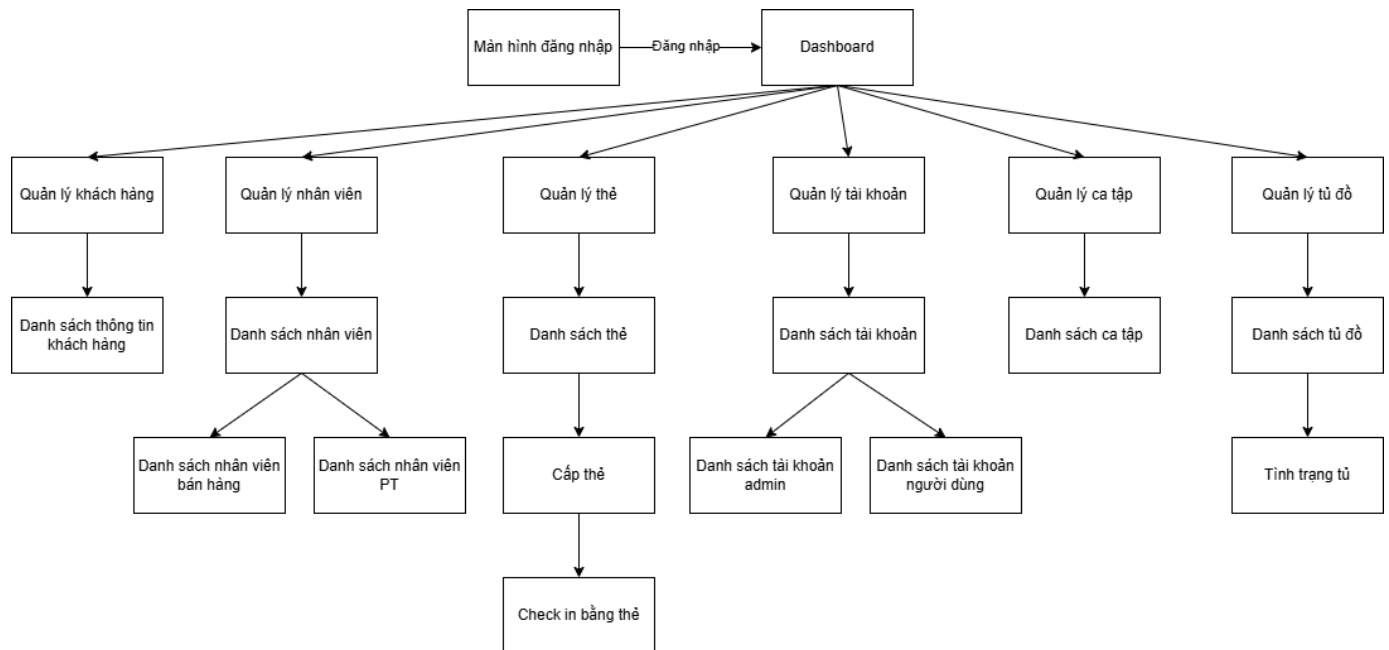
Hình 9: Thiết kế màn hình Quản lý thẻ

### 2.4.7. Màn hình Quản lý tài khoản



Hình 10: Thiết kế màn hình Quản lý tài khoản

### 2.4.8. Sơ đồ điều hướng giữa các màn hình



Hình 11: Sơ đồ điều hướng giữa các màn hình

## CHƯƠNG 3. XÂY DỰNG HỆ THỐNG

### 3.1. Giới thiệu ngôn ngữ cài đặt

#### 3.1.1. Giới thiệu về ngôn ngữ Angular

- **Angular là gì?**

Angular là một framework nổi tiếng, được sử dụng rộng rãi trong việc phát triển ứng dụng web đơn trang (SPAs) và ứng dụng web động. Được Google phát triển, Angular cung cấp một giải pháp mạnh mẽ cho việc tiếp cận cấu trúc, giúp phát triển các ứng dụng web phức tạp một cách hiệu quả. Angular sử dụng TypeScript để xây dựng ứng dụng, và tích hợp nhiều tính năng như two-way data binding, dependency injection, routing, forms. Nền tảng này giúp các nhà phát triển xây dựng và duy trì mã nguồn một cách dễ dàng và hiệu quả.



*Hình 12: Ngôn ngữ Angular*

- **Lịch sử hình thành Angular**

Angular, ban đầu được phát triển bởi Google, ra mắt lần đầu tiên dưới tên gọi AngularJS vào năm 2010. AngularJS nhanh chóng trở thành một trong những framework hàng đầu cho việc phát triển ứng dụng web đơn trang (SPA). Nền tảng này đã cung cấp một giải pháp hiệu quả để xây dựng các ứng dụng web phức tạp.

Vào năm 2016, Angular đã giới thiệu một phiên bản hoàn toàn mới với tên gọi Angular 2 (sau đó đổi thành Angular). Phiên bản này được viết lại từ đầu và sử dụng TypeScript thay vì JavaScript như phiên bản AngularJS. Angular 2, cùng với các phiên bản tiếp theo, đã mang lại nhiều cải tiến và tính năng mới, giúp việc phát triển ứng dụng web trở nên linh hoạt, mạnh mẽ và hiệu quả hơn.

Kể từ đó, Angular đã không ngừng phát triển và được cập nhật với các phiên bản mới định kỳ, mang lại nhiều tính năng và hiệu suất cải thiện đáng kể. Angular đã trở thành một trong những framework phổ biến nhất trong việc phát triển ứng dụng web và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm trên toàn thế giới.

#### • **Ưu điểm và nhược điểm của Angular**

Angular mang đến nhiều ưu điểm đáng chú ý, làm nó trở thành lựa chọn hàng đầu cho việc phát triển ứng dụng web, bao gồm:

- Custom Components: Angular cho phép người dùng xây dựng các component của riêng họ và có thể đóng gói chức năng cùng với logic kết xuất thành các phần có thể tái sử dụng. Nó cũng hoạt động tốt với các component của web.
- Data binding: Angular cho phép người dùng dễ dàng di chuyển dữ liệu từ mã JavaScript sang view và phản hồi với các sự kiện của người dùng mà không cần phải viết bất kỳ mã nào theo cách thủ công.
- Dependency Injection: Angular cho phép người dùng viết các service module và đưa chúng vào bất cứ nơi nào họ cần. Điều này cải thiện khả năng kiểm tra và khả năng tái sử dụng của các service giống nhau.
- Testing: có thể dễ dàng Unit test mọi phần trong ứng dụng của mình.
- Comprehensive: Angular là một full-fledged Framework và có thể cung cấp các giải pháp tiện ích cho giao tiếp máy chủ, định tuyến trong ứng dụng của bạn và hơn thế nữa.



– Browser Compatibility: Angular cung cấp đa nền tảng và tương thích với trình duyệt. Một ứng dụng Angular thường có thể chạy trên tất cả các trình duyệt (Chrome, Firefox) và hệ điều hành, chẳng hạn như Windows, macOS và Linux.

Mặc dù có nhiều ưu điểm, song Angular vẫn tồn tại những hạn chế sau:

– Không an toàn: thông thường, bản chất của Angular là một trong những Framework Front-end, mà Front-end này thường vốn không thể bảo mật bằng Back-end. Chính vì vậy, khi sử dụng API thì bạn cần xây dựng cho một hệ thống kiểm tra dữ liệu sao cho việc trả về được tốt nhất.

– Với một số trình duyệt sở hữu tính năng Disable JavaScript nên có nghĩa là website sẽ không hoàn toàn có thể sử dụng được dựa trên những trình duyệt đó nữa.

– Bạn nên viết mã ứng dụng Angular bằng TypeScript, vì vậy bạn phải học TypeScript.

### 3.1.2. Giới thiệu về SQL Server

- **SQL Server là gì?**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System) do Microsoft phát triển. Nó cung cấp một môi trường cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng và hệ thống thông tin kinh doanh.



*Hình 13: Microsoft SQL Server là gì ?*

- **Lịch sử hình thành SQL Server:**

SQL Server có một lịch sử phát triển dài từ khi nó được Microsoft giới thiệu lần đầu vào những năm 1980. Sau đây là một số tóm tắt về lịch sử ra đời của SQL Server:

- SQL Server 1.0 (1989): SQL Server được Microsoft phát triển lần đầu tiên và được giới thiệu dành riêng cho hệ điều hành OS/2. Phiên bản đầu tiên này chỉ hỗ trợ một phần của ngôn ngữ truy vấn SQL và có các hạn chế về tính năng so với các hệ quản trị cơ sở dữ liệu khác vào thời điểm đó.

- SQL Server 4.2 (1993): SQL Server 4.2 là phiên bản đầu tiên hỗ trợ hệ điều hành Windows NT. Nó đã bắt đầu đưa SQL Server trở thành một giải pháp phổ biến hơn trong doanh nghiệp.

- SQL Server 6.0 (1995): phiên bản này đánh dấu sự xuất hiện của SQL Server Enterprise Edition và Standard Edition, cung cấp các tính năng như khóa cấp độ hàng (row-level locking), lập lịch công việc, và các chức năng mới khác.

- SQL Server 7.0 (1998): SQL Server 7.0 là phiên bản đầu tiên của SQL Server được phát hành sau khi Microsoft mua lại công ty Relational Software Inc., người đã phát triển cơ sở dữ liệu Sybase. Phiên bản này đánh dấu sự chuyển đổi lớn từ cơ sở dữ liệu dựa trên bộ nhớ ra cơ sở dữ liệu dựa trên bộ nhớ và định dạng tệp lưu trữ mới.

- SQL Server 2000 (2000): SQL Server 2000 được phát hành với nhiều cải tiến về hiệu suất, bảo mật và tính năng quản lý. Nó cũng cung cấp tính năng mới như XML, DTS và tích hợp với hệ thống Windows.

- SQL Server 2005 (2005): SQL Server 2005 mang lại nhiều cải tiến về hiệu suất và tính năng so với các phiên bản trước. Nó đưa ra các công nghệ mới như CLR Integration, XML và tích hợp với Visual Studio.

- SQL Server 2008 (2008): SQL Server 2008 tiếp tục cải thiện hiệu suất và bảo mật. Nó cung cấp các tính năng mới như SQL Server Reporting Services, SQL Server Analysis Services và SQL Server Integration Services.

Kể từ đó, Microsoft đã liên tục phát triển và cập nhật SQL Server với các phiên bản mới, đồng thời tích hợp các công nghệ mới như trí tuệ nhân tạo, học máy và tính toán đám mây. SQL Server có một số điểm nổi bật:

- Quản lý dữ liệu: SQL Server cho phép bạn lưu trữ dữ liệu trong các bảng có cấu trúc và tương tác với chúng bằng ngôn ngữ truy vấn SQL.

- Bảo mật dữ liệu: nó cung cấp các cơ chế bảo mật như quản lý quyền truy cập, mã hóa dữ liệu và kiểm soát truy cập vào dữ liệu, giúp bảo vệ thông tin quan trọng.

- Tích hợp với các ứng dụng khác của Microsoft: SQL Server tích hợp chặt chẽ với các ứng dụng khác của Microsoft như .NET Framework, Azure, và các công cụ phát triển như Visual Studio.

- Phân tích và báo cáo: nó cung cấp các công cụ phân tích dữ liệu và báo cáo như SSAS và SSRS để phân tích và hiển thị dữ liệu một cách dễ dàng.

- Tính khả dụng và tính nhất quán: SQL Server cung cấp các tính năng như sao lưu dự phòng, nhân bản dữ liệu và giao thức giao tiếp giữa các máy chủ để đảm bảo tính khả dụng và tính nhất quán của dữ liệu.

- Tích hợp với các công nghệ mới: SQL Server liên tục được cập nhật với các tính năng mới để hỗ trợ các công nghệ mới như trí tuệ nhân tạo, học máy và tính toán điện toán đám mây.

- **Ưu điểm của SQL Server:**

- Tính dễ sử dụng: SQL Server có giao diện người dùng dễ sử dụng và hỗ trợ rất tốt cho người mới bắt đầu.

- Bảo mật mạnh mẽ: cung cấp các tính năng bảo mật mạnh mẽ như mã hóa dữ liệu, quản lý quyền truy cập và kiểm soát truy cập.

- Hiệu năng cao: SQL Server thường có hiệu suất cao, đặc biệt là khi được tối ưu hóa và cấu hình đúng.

- Tích hợp tốt: tích hợp chặt chẽ với các sản phẩm và công nghệ khác của Microsoft như .NET Framework, Azure, Excel và Visual Studio.

- Hỗ trợ tích hợp báo cáo và phân tích: SQL Server đi kèm với các công cụ như SSRS và SSAS để tạo và phân tích báo cáo dữ liệu một cách dễ dàng.

- **Nhược điểm của SQL Server:**

- Chi phí: SQL Server có chi phí đầu tư ban đầu cao, đặc biệt là khi tính tới cả phần cứng và phần mềm.

- Hạn chế trên hệ thống không phải Windows: mặc dù SQL Server có thể chạy trên Linux, nhưng vẫn có những hạn chế so với việc chạy trên hệ điều hành Windows.

- Khả năng mở rộng: trong một số trường hợp, việc mở rộng và quản lý SQL Server có thể trở nên phức tạp, đặc biệt là khi xử lý lượng dữ liệu lớn.

- Cần kiến thức chuyên môn: để tối ưu hóa và quản lý SQL Server hiệu quả, người quản trị cần có kiến thức chuyên môn về hệ thống cơ sở dữ liệu và SQL Server.

### **3.1.3. Giới thiệu về ASP.NET Core Web API**

- **ASP.NET Core Web API là gì?**

ASP.NET Core Web API là một framework của Microsoft cho việc xây dựng các dịch vụ web RESTful trên nền tảng .NET Framework.

- **Lịch sử hình thành ASP.NET Core Web API:**

ASP.NET Core Web API đã trải qua một quá trình phát triển từ khi nó được giới thiệu lần đầu tiên đến hiện tại. Sau đây là một vài tóm tắt về lịch sử hình thành của ASP.NET Core Web API:

- ASP.NET MVC Framework (2007): trước khi ASP.NET Web API ra đời, ASP.NET MVC Framework đã được phát triển và giới thiệu vào năm 2007.

ASP.NET MVC cung cấp một cách tiếp cận mới cho việc xây dựng ứng dụng web dựa trên mô hình kiến trúc MVC.

- ASP.NET Web API 1 (2012): ASP.NET Web API được giới thiệu chính thức vào năm 2012, nhằm mục đích cung cấp một framework cho việc xây dựng các dịch vụ web RESTful trên nền tảng .NET Framework. Phiên bản đầu tiên này đã tạo ra một sự đột phá trong việc phát triển các ứng dụng web dựa trên HTTP.

- ASP.NET Web API 2 (2013): phiên bản ASP.NET Web API 2 được phát hành vào năm 2013, mang lại nhiều cải tiến và tính năng mới như Attribute Routing, Cors, OWIN integration, và IHttpActionResult interface.

- ASP.NET Core (2016): với sự ra đời của .NET Core và ASP.NET Core vào năm 2016, ASP.NET Web API đã trở thành một phần của ASP.NET Core. ASP.NET Core mang lại sự linh hoạt, hiệu suất và tính di động cao hơn so với các phiên bản trước đó.

- ASP.NET Core Web API (Hiện tại): ASP.NET Core Web API tiếp tục được phát triển và cập nhật liên tục theo các phiên bản của ASP.NET Core. Nó là một trong những công nghệ hàng đầu cho việc xây dựng các dịch vụ web RESTful hiện đại trên nền tảng .NET Core.



*Hình 14: ASP.NET Core Web API là gì?*

- **Một số điểm nổi bật của ASP.NET Core Web API:**

- RESTful Service Development: ASP.NET Core Web API cho phép bạn xây dựng các dịch vụ web theo kiến trúc RESTful, giúp tạo ra các API dễ sử dụng, linh hoạt và dễ dàng tích hợp với các ứng dụng khác.

- Hỗ trợ đa dạng định dạng dữ liệu: Web API hỗ trợ các định dạng dữ liệu như JSON, XML, và Atom. Điều này cho phép ứng dụng của bạn tương tác với API bằng nhiều ngôn ngữ lập trình và nền tảng khác nhau.

- Routing và Dependency Injection: ASP.NET Core Web API cung cấp các công cụ mạnh mẽ như routing và dependency injection, giúp bạn quản lý các yêu cầu HTTP một cách dễ dàng và linh hoạt.

- Tích hợp tốt với .NET Framework: Web API tích hợp chặt chẽ với .NET Framework và các công nghệ khác của Microsoft như Entity Framework, ASP.NET MVC và Azure, giúp bạn xây dựng website toàn diện và mạnh mẽ.

- Bảo mật: Web API cung cấp các cơ chế bảo mật như xác thực và ủy quyền, cho phép bạn kiểm soát quyền truy cập vào các dịch vụ của mình.

- Tích hợp tốt với công cụ phát triển: ASP.NET Core Web API được tích hợp sâu vào Visual Studio, cho phép bạn phát triển, thử nghiệm và triển khai các dịch vụ web một cách hiệu quả. Với những điểm mạnh như vậy, ASP.NET Core Web API là một lựa chọn phổ biến cho việc xây dựng các dịch vụ web RESTful trong các ứng dụng .NET.

- **Ưu điểm của ASP.NET Core Web API:**

- Dễ sử dụng và triển khai: ASP.NET Core Web API tích hợp chặt chẽ với .NET Framework hoặc .NET Core, điều này làm cho việc phát triển và triển khai các dịch vụ web RESTful trở nên đơn giản và hiệu quả.

- Tích hợp với các công nghệ .NET khác: Web API có khả năng tích hợp tốt với các công nghệ khác của Microsoft như Entity Framework, ASP.NET MVC, và Azure, giúp tăng tính linh hoạt và mở rộng của ứng dụng.

- Hỗ trợ đa định dạng dữ liệu: ASP.NET Core Web API hỗ trợ nhiều định dạng dữ liệu như JSON, XML, và BSON, giúp tương tác với các ứng dụng khác một cách dễ dàng.

- Hỗ trợ tốt cho RESTful: ASP.NET Core Web API được thiết kế để hỗ trợ việc xây dựng dịch vụ web RESTful, giúp tạo ra các API linh hoạt và dễ bảo trì.

- Bảo mật: Web API cung cấp các cơ chế bảo mật như xác thực và ủy quyền, giúp bảo vệ các dịch vụ và dữ liệu của ứng dụng.

- **Nhược điểm của ASP.NET Core Web API:**

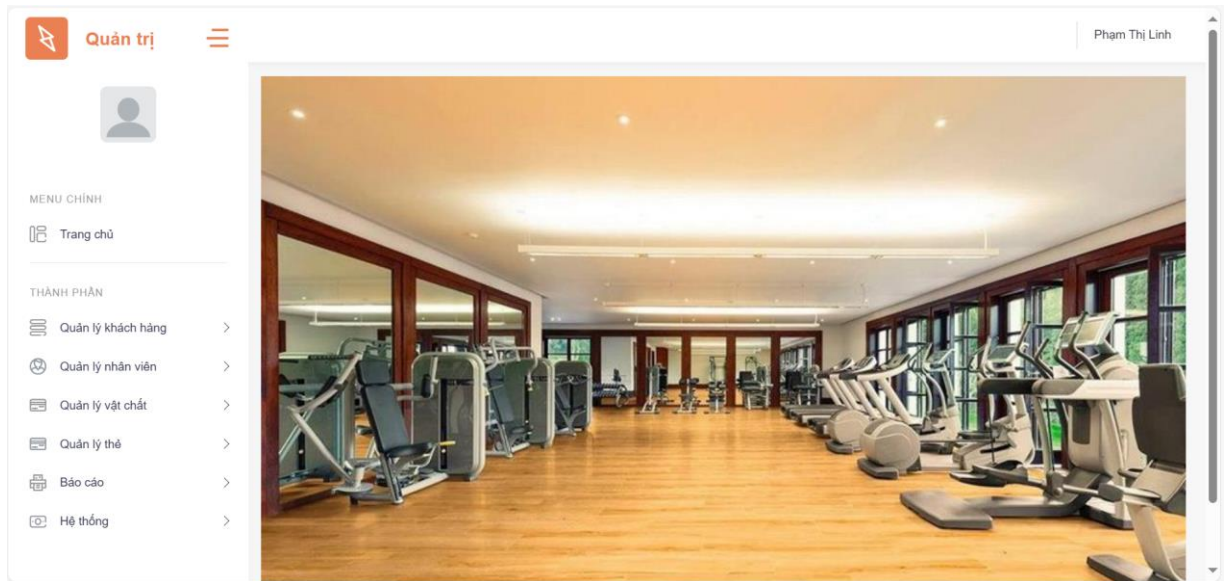
- Khả năng mở rộng: mặc dù Web API cho phép mở rộng ứng dụng, nhưng trong một số trường hợp, việc quản lý và mở rộng các dịch vụ có thể trở nên phức tạp.

- Hiệu suất: trong một số trường hợp, hiệu suất của Web API có thể bị ảnh hưởng do việc xử lý yêu cầu HTTP và dữ liệu lớn.

- Học phức tạp: dù ASP.NET Core Web API dễ sử dụng, nhưng việc hiểu và triển khai các khái niệm RESTful có thể đòi hỏi một khoảng thời gian để học và hiểu rõ.

### 3.2. Các kết quả đạt được

#### 3.2.1. Màn hình Trang chủ



Hình 15: Màn hình Trang chủ

#### 3.2.2. Use case Đăng nhập

### ĐĂNG NHẬP TRANG QUẢN LÝ

Tên đăng nhập

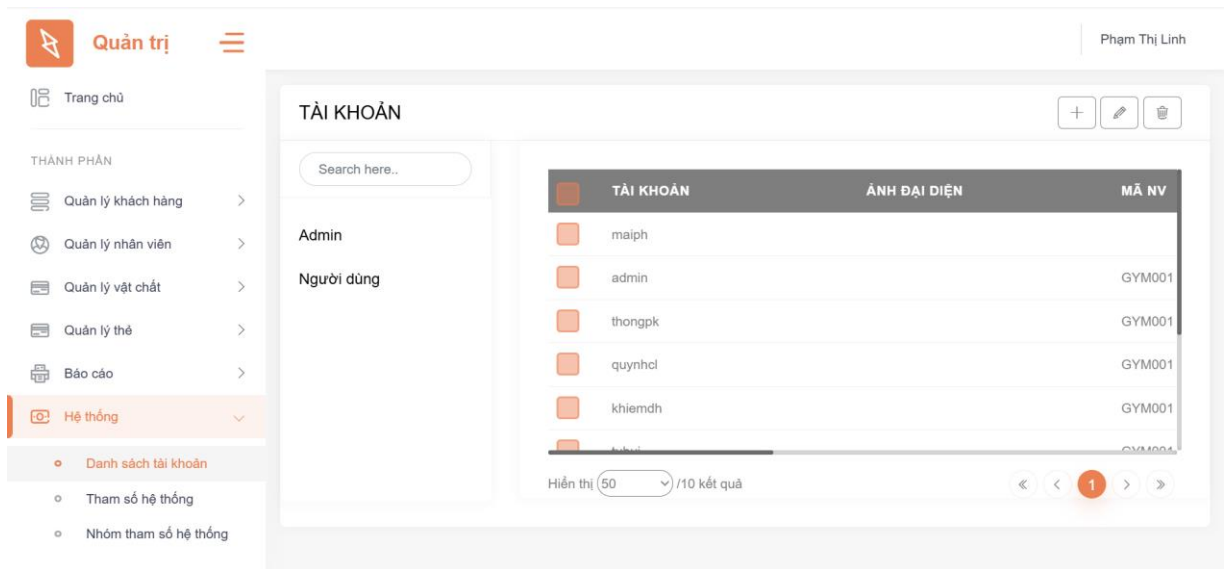
Mật khẩu

Đăng nhập

Hình 16: Màn hình Đăng nhập

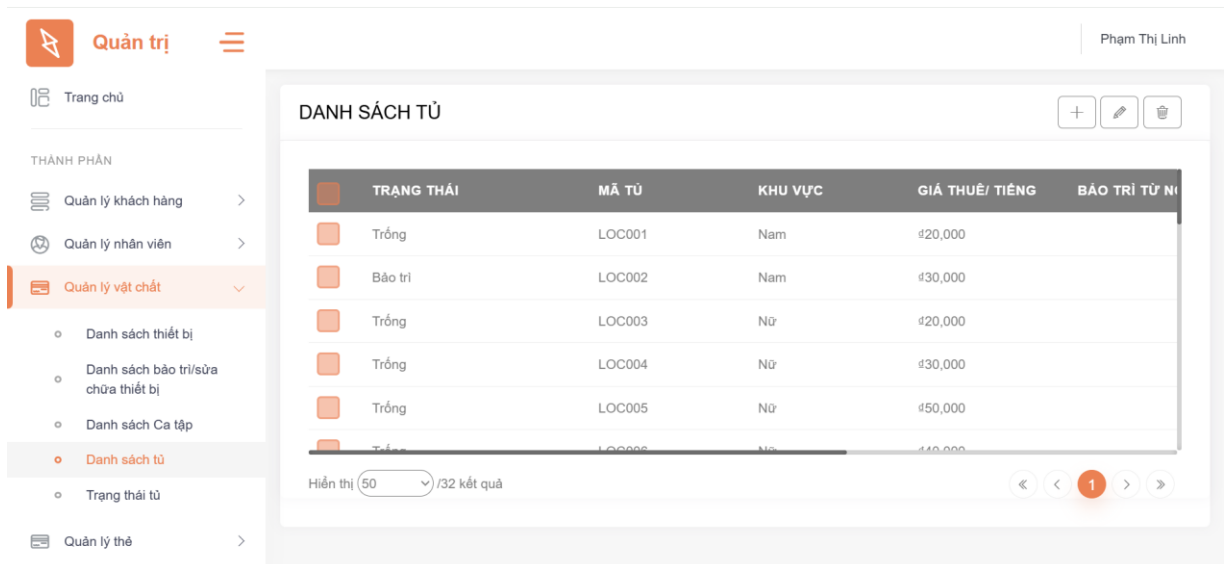


### 3.2.3. Use case Quản lý tài khoản



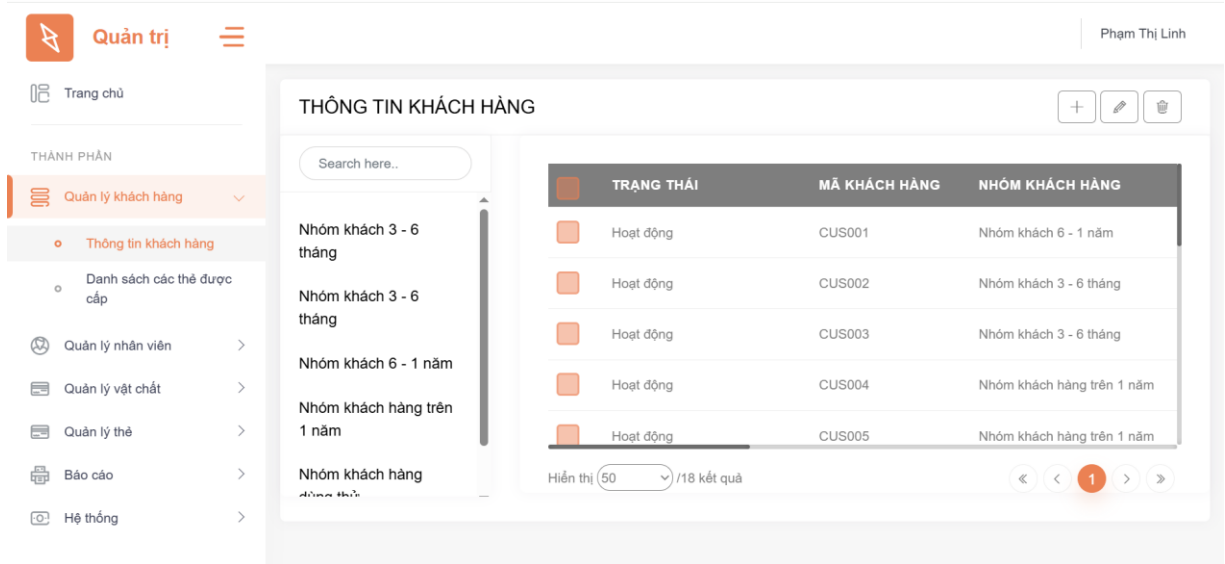
Hình 17: Màn hình Quản lý tài khoản

### 3.2.4. Use case Quản lý tủ đồ



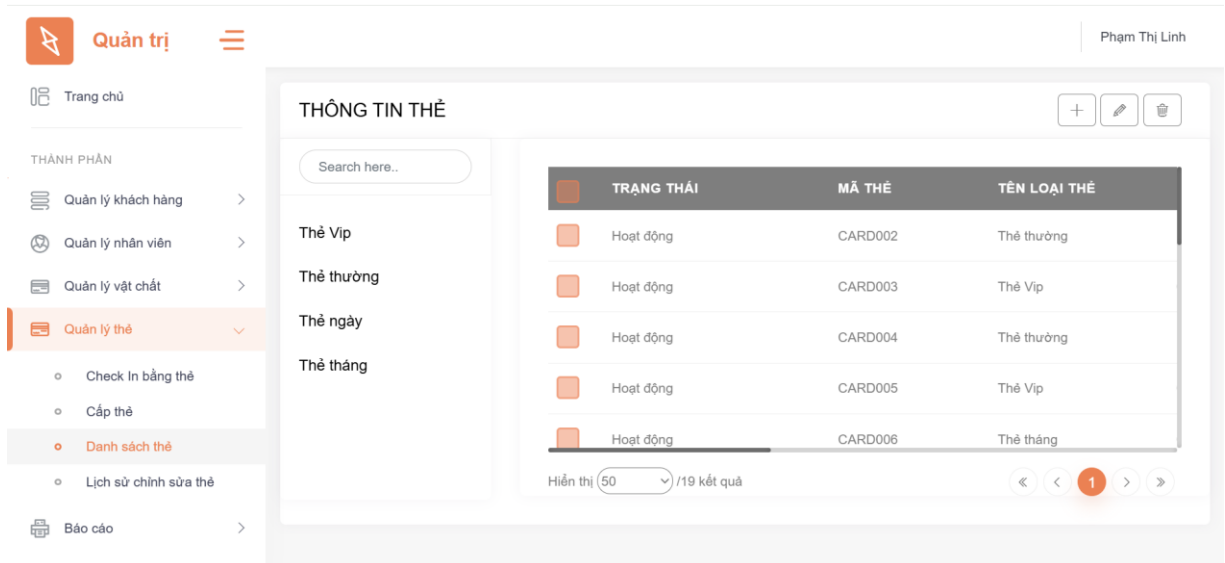
Hình 18: Màn hình Quản lý tủ đồ

### 3.2.5. Use case Quản lý khách hàng



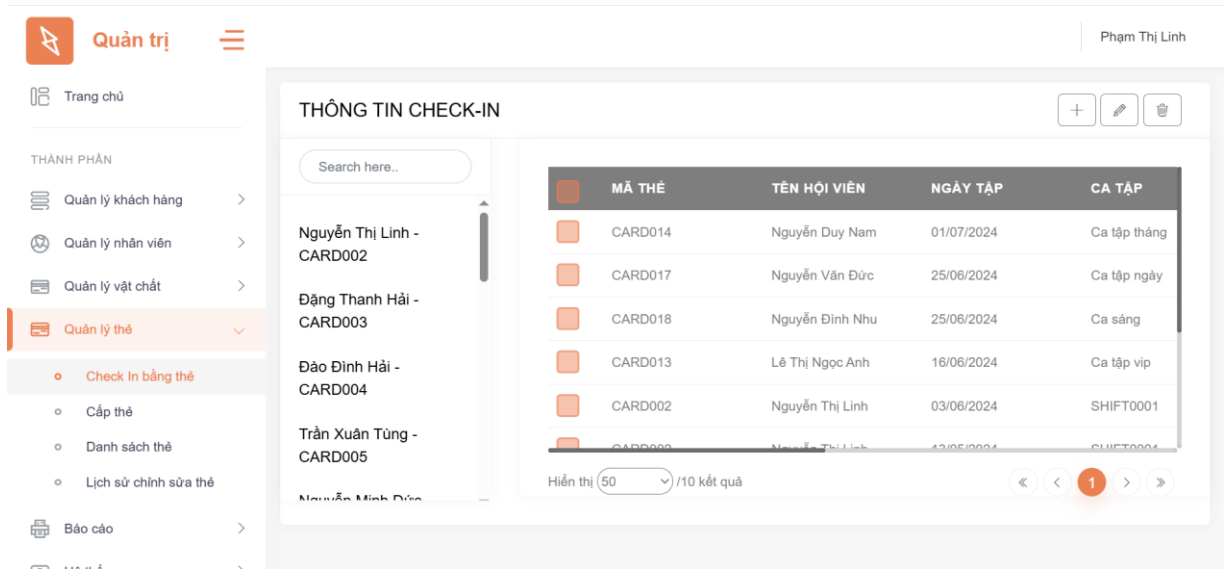
Hình 19: Màn hình Quản lý khách hàng

### 3.2.6. Use case Quản lý thẻ



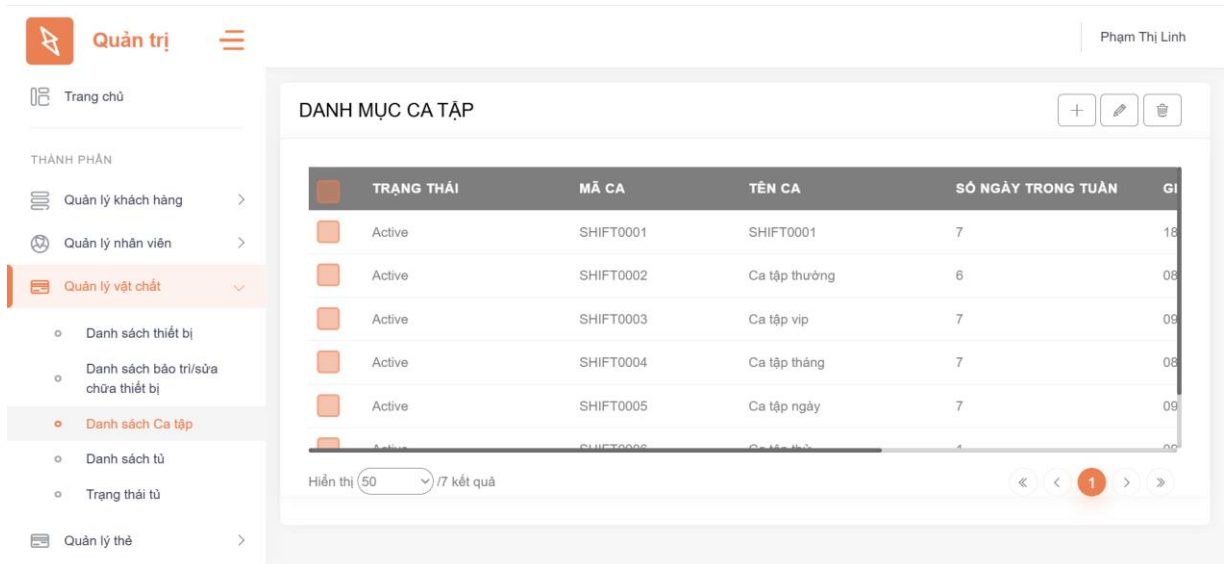
Hình 20: Màn hình Quản lý thẻ

### 3.2.7. Use case Check in-out



Hình 21: Màn hình Check in

### 3.2.8. Use case Quản lý ca tập



Hình 22: Màn hình Quản lý ca tập

### 3.3. Kiểm thử hệ thống

#### 3.3.1. Phạm vi kiểm thử

##### Những chức năng được kiểm thử:

- Đăng nhập: Kiểm tra chức năng đăng nhập vào hệ thống với thông tin tài khoản hợp lệ
- Quản lý tài khoản: Kiểm tra chức năng cho phép người quản lý xem, thêm, sửa, xóa các tài khoản
- Quản lý tủ đồ: Kiểm tra chức năng cho phép người quản lý xem, thêm, sửa xóa thông tin trạng thái tủ đồ
- Quản lý khách hàng: Kiểm tra chức năng cho phép người quản lý xem, thêm, sửa, xóa, lọc thông tin khách hàng
- Quản lý thẻ: Kiểm tra chức năng cho phép người quản lý xem, thêm, sửa, xóa, lọc thông tin thẻ
- Quản lý ca tập: Kiểm tra chức năng cho phép người quản lý xem, thêm, sửa, xóa thông tin ca tập

##### Những chức năng không được kiểm thử:

- Check in-out
- Đăng ký gói tập
- Thống kê/Báo cáo
- Mua hàng

#### 3.3.2. Test case

##### 3.3.2.1. Test case chức năng Đăng nhập

ID	Mô tả	Các bước thực hiện	Kết quả mong muốn
1	Kiểm tra các item của màn hình	Truy cập trang web	Màn hình hiển thị: - Tên đăng nhập textbox - Mật khẩu textbox - Đăng nhập button

2	Kiểm tra giao diện màn hình	Truy cập trang web	<ul style="list-style-type: none"> <li>- Các label, textbox, button có độ dài vừa đủ bằng nhau, dễ nhìn, không xô lệch</li> <li>- Các label cùng kích cỡ, font chữ</li> <li>- Form bố trí hợp lý dễ nhìn, dễ sử dụng</li> </ul>
3	Bỏ trống các trường đăng nhập	1. Bỏ trống textbox 2. Kích “Đăng nhập”	2. Hệ thống hiển thị thông báo lỗi
4	Đăng nhập bằng tài khoản không tồn tại	1. Điền tên đăng nhập, mật khẩu không tồn tại trong cơ sở dữ liệu 2. Kích “Đăng nhập”	2. Hệ thống hiển thị thông báo lỗi
5	Đăng nhập sai tên đăng nhập, mật khẩu	1. Điền sai tên đăng nhập, mật khẩu 2. Kích “Đăng nhập”	2. Hệ thống hiển thị thông báo lỗi
6	Đăng nhập thành công	1. Điền dữ liệu hợp lệ vào textbox 2. Kích “Đăng nhập”	2. Hệ thống hiển thị thông báo đăng nhập thành công và màn hình trang chủ

*Bảng 4-1. Test case chức năng Đăng nhập*

### 3.3.2.2. Test case chức năng Quản lý tài khoản

ID	Mô tả	Các bước thực hiện	Kết quả mong muốn
1	Kiểm tra các item của màn hình	1. Đăng nhập vào trang web 2. Kích “Hệ thống” 3. Kích “Quản lý tài khoản”	Màn hình hiển thị: - Các button thêm, sửa, xóa - Ô tìm kiếm tài khoản - Lưới danh sách các tài khoản
2	Kiểm tra giao diện màn hình	1. Đăng nhập vào trang web 2. Kích “Hệ thống” 3. Kích “Quản lý tài khoản”	<ul style="list-style-type: none"> <li>- Các label, button có độ dài vừa đủ bằng nhau, dễ nhìn, không xô lệch, màu sắc hài hòa</li> <li>- Các label cùng kích cỡ, font chữ</li> </ul>

			- Form bố trí hợp lý dễ nhìn, dễ sử dụng
3	Thêm tài khoản hợp lệ	1. Kích “Thêm” 2. Điền đầy đủ thông tin hợp lệ 3. Kích “Lưu”	1. Hiện thị form điền thông tin 3.1. Hệ thống thêm dữ liệu vào bảng SYS_USER và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công và quay lại màn hình danh sách tài khoản
4	Sửa thông tin tài khoản	1. Chọn 1 tài khoản 2. Kích “Sửa” 3. Điền thông tin mới hợp lệ 4. Kích “Lưu”	2. Hiện thị form với thông tin cũ của tài khoản 4.1. Hệ thống sửa dữ liệu trong bảng SYS_USER 4.2. Hiện thị thông báo thành công và quay lại màn hình danh sách
5	Xóa tài khoản	1. Chọn 1 tài khoản 2. Kích “Xóa” 3. Kích “Đồng ý”	2. Hiện thị pop-up xác nhận xóa 3.1. Hệ thống xóa dữ liệu trong bảng SYS_USER và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công
6	Kiểm tra hoạt động nút “Thoát”	1. Kích “Thêm” 2. Kích “Thoát”	2. Quay trở lại màn hình danh sách tài khoản

Bảng 4-2: Test case chức năng Quản lý tài khoản

### 3.3.2.3. Test case chức năng Quản lý tủ đồ

ID	Mô tả	Các bước thực hiện	Kết quả mong muốn
1	Kiểm tra các item của màn hình	1. Đăng nhập vào trang web 2. Kích “Quản lý vật chất” 3. Kích “Danh sách tủ”	Màn hình hiển thị: - Các button thêm, sửa, xóa - Lưới danh sách tủ đồ

2	Kiểm tra giao diện màn hình	1. Đăng nhập vào trang web 2. Kích “Quản lý vật chất” 3. Kích “Danh sách tủ”	- Các label, button có độ dài vừa đủ bằng nhau, dễ nhìn, không xô lệch, màu sắc hài hòa - Các label cùng kích cỡ, font chữ - Form bố trí hợp lí dễ nhìn, dễ sử dụng
3	Thêm thông tin tủ hợp lệ	1. Kích “Thêm” 2. Điền đầy đủ thông tin hợp lệ 3. Kích “Lưu”	1. Hiện thị form điền thông tin 3.1. Hệ thống thêm dữ liệu vào bảng GOODS_LOCKER và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công và quay lại màn hình danh sách tủ đồ
4	Sửa thông tin tủ đồ	1. Chọn 1 tài khoản 2. Kích “Sửa” 3. Điền thông tin mới hợp lệ 4. Kích “Lưu”	2. Hiện thị form với thông tin cũ của tủ đồ 4.1. Hệ thống sửa dữ liệu trong bảng GOODS_LOCKER 4.2. Hiện thị thông báo thành công và quay lại màn hình danh sách
5	Xóa tủ đồ	1. Chọn 1 tủ đồ 2. Kích “Xóa” 3. Kích “Đồng ý”	2. Hiện thị pop-up xác nhận xóa 3.1. Hệ thống xóa dữ liệu trong bảng GOODS_LOCKER và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công
6	Kiểm tra hoạt động nút “Thoát”	1. Kích “Thêm” 2. Kích “Thoát”	2. Quay trở lại màn hình danh sách tủ đồ

Bảng 4-3: Test case chức năng Quản lý tủ đồ

### 3.3.2.4. Test case chức năng Quản lý khách hàng

ID	Mô tả	Các bước thực hiện	Kết quả mong muốn
1	Kiểm tra các item của màn hình	1. Đăng nhập vào trang web 2. Kích “Quản lý khách hàng” 3. Kích “Thông tin khách hàng”	Màn hình hiển thị: - Các button thêm, sửa, xóa - Ô tìm kiếm khách hàng - Bộ lọc danh sách nhóm khách hàng - Lưới danh sách các khách hàng
2	Kiểm tra giao diện màn hình Quản lý khách hàng	1. Đăng nhập vào trang web 2. Kích “Quản lý khách hàng” 3. Kích “Thông tin khách hàng”	- Các label, button có độ dài vừa đủ bằng nhau, dễ nhìn, không xô lệch, màu sắc hài hòa - Các label cùng kích cỡ, font chữ - Form bố trí hợp lý dễ nhìn, dễ sử dụng
3	Thêm khách hàng hợp lệ	1. Kích “Thêm” 2. Điền đầy đủ thông tin hợp lệ 3. Kích “Lưu”	1. Hiện thị form điền thông tin 3.1. Hệ thống thêm dữ liệu vào bảng PER_CUSTOMER và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công và quay lại màn hình danh sách tài khoản
4	Sửa thông tin khách hàng	1. Chọn 1 khách hàng 2. Kích “Sửa” 3. Điền thông tin mới hợp lệ 4. Kích “Lưu”	2. Hiện thị form với thông tin cũ của khách hàng 4.1. Hệ thống sửa dữ liệu trong bảng PER_CUSTOMER 4.2. Hiện thị thông báo thành công và quay lại màn hình danh sách
5	Xóa khách hàng	1. Chọn 1 khách hàng 2. Kích “Xóa” 3. Kích “Đồng ý”	2. Hiện thị pop-up xác nhận xóa 3.1. Hệ thống xóa dữ liệu trong bảng PER_CUSTOMER và cập nhật lại danh sách



			3.2. Hiện thị thông báo thành công
6	Kiểm tra hoạt động nút “Thoát”	1. Kích “Thêm” 2. Kích “Thoát”	2. Quay trở lại màn hình danh sách khách hàng
7	Kiểm tra chức năng lọc khách hàng theo danh sách	Chọn 1 nhóm khách hàng có trong bộ lọc	Hiện thị chính xác danh sách khách hàng thuộc nhóm khách hàng đó

*Bảng 4-4: Test case chức năng Quản lý khách hàng*

### 3.3.2.5. Test case chức năng Quản lý thẻ

ID	Mô tả	Các bước thực hiện	Kết quả mong muốn
1	Kiểm tra các item của màn hình	1. Đăng nhập vào trang web 2. Kích “Quản lý thẻ” 3. Kích “Danh sách thẻ”	Màn hình hiển thị: - Các button thêm, sửa, xóa - Ô tìm kiếm thẻ - Bộ lọc danh sách loại thẻ - Lưới danh sách thẻ
2	Kiểm tra giao diện màn hình Quản lý thẻ	1. Đăng nhập vào trang web 2. Kích “Quản lý thẻ” 3. Kích “Danh sách thẻ”	- Các label, button có độ dài vừa đủ bằng nhau, dễ nhìn, không xô lệch, màu sắc hài hòa - Các label cùng kích cỡ, font chữ - Form bố trí hợp lý dễ nhìn, dễ sử dụng
3	Thêm thông tin thẻ hợp lệ	1. Kích “Thêm” 2. Điền đầy đủ thông tin hợp lệ 3. Kích “Lưu”	1. Hiện thị form điền thông tin 3.1. Hệ thống thêm dữ liệu vào bảng CARD_INFO và cập nhật lại danh sách

			3.2. Hiện thị thông báo thành công và quay lại màn hình danh sách thẻ
4	Sửa thông tin thẻ	1. Chọn 1 thẻ 2. Kích “Sửa” 3. Điền thông tin mới hợp lệ 4. Kích “Lưu”	2. Hiện thị form với thông tin cũ của thẻ 4.1. Hệ thống sửa dữ liệu trong bảng CARD_INFO 4.2. Hiện thị thông báo thành công và quay lại màn hình danh sách
5	Xóa thẻ	1. Chọn 1 thẻ 2. Kích “Xóa” 3. Kích “Đồng ý”	2. Hiện thị pop-up xác nhận xóa 3.1. Hệ thống xóa dữ liệu trong bảng CARD_INFO và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công
6	Kiểm tra hoạt động nút “Thoát”	1. Kích “Thêm” 2. Kích “Thoát”	2. Quay trở lại màn hình danh sách thẻ
7	Kiểm tra chức năng lọc thẻ theo loại thẻ	Chọn 1 loại thẻ có trong bộ lọc	Hiện thị chính xác danh sách thẻ thuộc loại thẻ đó

Bảng 4-5: Test case chức năng Quản lý thẻ

### 3.3.2.6. Test case chức năng Quản lý ca tập

ID	Mô tả	Các bước thực hiện	Kết quả mong muốn
1	Kiểm tra các item của màn hình	1. Đăng nhập vào trang web 2. Kích “Quản lý vật chất” 3. Kích “Quản lý ca tập”	Màn hình hiển thị: - Các button thêm, sửa, xóa - Lưới danh sách các ca tập
2	Kiểm tra giao diện màn hình Quản lý ca tập	1. Đăng nhập vào trang web 2. Kích “Quản lý vật chất” 3. Kích “Quản lý ca tập”	- Các label, button có độ dài vừa đủ bằng nhau, dễ nhìn, không xô lệch, màu sắc hài hòa - Các label cùng kích cỡ, font chữ - Form bố trí hợp lí dễ nhìn, dễ sử dụng
3	Thêm ca tập hợp lệ	1. Kích “Thêm” 2. Điền đầy đủ thông tin hợp lệ 3. Kích “Lưu”	1. Hiện thị form điền thông tin 3.1. Hệ thống thêm dữ liệu vào bảng GOODS_SHIFT và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công và quay lại màn hình danh sách ca tập
4	Sửa thông tin ca tập	1. Chọn 1 ca tập 2. Kích “Sửa” 3. Điền thông tin mới hợp lệ 4. Kích “Lưu”	2. Hiện thị form với thông tin cũ của ca tập 4.1. Hệ thống sửa dữ liệu trong bảng GOODS_SHIFT

			4.2. Hiện thị thông báo thành công và quay lại màn hình danh sách
5	Xóa ca tập	1. Chọn 1 ca tập 2. Kích “Xóa” 3. Kích “Đồng ý”	2. Hiện thị pop-up xác nhận xóa 3.1. Hệ thống xóa dữ liệu trong bảng GOODS_SHIFT và cập nhật lại danh sách 3.2. Hiện thị thông báo thành công
6	Kiểm tra hoạt động nút “Thoát”	1. Kích “Thêm” 2. Kích “Thoát”	2. Quay trở lại màn hình danh sách ca tập

*Bảng 4-6: Test case chức năng Quản lý ca tập*

## KẾT LUẬN

### 1. Kết quả đạt được

Website quản lý phòng tập gym đã được phát triển với một loạt các chức năng nhằm tối ưu hóa quá trình quản lý các hoạt động trong phòng tập gym. Các chức năng chính bao gồm: Đăng nhập/Đăng xuất, Phân quyền, Xem danh sách ca tập, Quản lý hệ thống, Quản lý vật chất, Quản lý nhân viên, Quản lý khách hàng và Thống kê/Báo cáo.

- Chức năng Đăng nhập/Đăng xuất: chức năng này đã được triển khai thành công, cho phép người dùng đăng nhập vào hệ thống và đăng xuất khi kết thúc phiên làm việc, đảm bảo tính bảo mật thông tin.

- Phân quyền: hệ thống phân quyền đã được xây dựng một cách linh hoạt, cho phép quản trị viên cấp quyền truy cập tới các tính năng cụ thể dựa trên vai trò của từng người dùng.

- Xem danh sách ca tập: chức năng này cho phép nhân viên và quản trị viên xem danh sách các ca tập có sẵn và thực hiện các thao tác liên quan một cách thuận tiện.

- Quản lý hệ thống: hệ thống quản lý được thiết kế để giúp quản trị viên quản lý và điều chỉnh các cài đặt cũng như thông tin cơ bản của phòng tập gym một cách hiệu quả.

- Quản lý vật chất: chức năng này giúp quản trị viên theo dõi và quản lý vật chất của phòng tập gym, bao gồm cả việc thêm mới, cập nhật và xóa bỏ vật chất khi cần thiết.

- Quản lý nhân viên và khách hàng: hệ thống cho phép quản trị viên quản lý thông tin của nhân viên và khách hàng, bao gồm cả việc thêm mới, chỉnh sửa và xóa bỏ thông tin.

- Thống kê/Báo cáo: chức năng thống kê cung cấp các báo cáo tổng quan về hoạt động của phòng tập gym, giúp quản trị viên đánh giá hiệu suất và đưa ra các quyết định chiến lược.

## **2. Hạn chế của dự án**

- Dù mục tiêu đề ra là cho phép hội viên đăng ký và thanh toán các gói tập, dịch vụ bổ sung trực tuyến, nhưng hiện tại hệ thống chưa tích hợp cổng thanh toán trực tuyến, gây hạn chế cho việc đăng ký dịch vụ một cách thuận tiện.

- Chưa hoàn thiện đầy đủ phân thống kê chi tiết.

- Hệ thống đã có quản lý khách hàng và ca tập, nhưng chưa có chức năng cụ thể để phân công huấn luyện viên theo gói tập cá nhân hoặc nhóm cho từng hội viên, dẫn đến việc quản lý còn mang tính thủ công.

- Hiện tại giao diện chính mới chỉ phù hợp trên trình duyệt web máy tính, chưa tối ưu hiển thị trên các thiết bị di động, gây bất tiện cho hội viên khi truy cập hệ thống mọi lúc mọi nơi.

- Chưa có chức năng đặt lịch linh hoạt theo thời gian thực: hệ thống chưa hỗ trợ đặt lịch học theo thời gian thực với cập nhật trạng thái ngay lập tức, nên việc điều phối có thể gặp khó khăn khi triển khai thực tế.

## **3. Hướng phát triển**

- Nâng cao tính năng và trải nghiệm người dùng: tiếp tục phát triển và cải thiện các tính năng của web để mang lại trải nghiệm người dùng tốt nhất có thể.

- Phát triển ứng dụng di động: xây dựng một ứng dụng di động đi kèm với website để cung cấp cho khách hàng một cách tiếp cận thuận lợi hơn và tăng tính di động cho họ.

- Tối ưu hóa quy trình quản lý: tiếp tục tối ưu hóa quy trình quản lý để giảm bớt thời gian và công sức đầu tư vào các hoạt động hàng ngày, từ đó tăng hiệu suất làm việc của nhân viên.

– Tăng cường tính tiện ích: khách hàng có thể dễ dàng đặt lịch tập, theo dõi tiến độ luyện tập, nhận thông tin về chương trình khuyến mãi và tương tác trực tiếp với phòng tập. Nhân viên có thể quản lý thông tin khách hàng, lên lịch hẹn, theo dõi hoạt động và cung cấp hỗ trợ tốt hơn.

## TÀI LIỆU THAM KHẢO

- [1] Nguyễn Thị Hương Lan, *Giáo trình Phân tích thiết kế hướng đối tượng*, NXB Khoa học và Kỹ thuật, 2015.
- [2] “Tổng quan API web công thông tin” [Online]. Available: (Tổng quan về API web công thông tin Power Pages | Microsoft Learn) [Accessed 1 4 2024]
- [3] “Kinh nghiệm quản lý phòng Gym hiệu quả” [Online]. Available: (<https://loffice.vn/kinh-nghiem-quan-ly-phong-gym-hieu-qua>) [Accessed 1 3 2024]
- [4] ".NET documentation," [Online]. Available: (<https://learn.microsoft.com/vivn/dotnet/framework/>) [Accessed 10 4 2024]
- [5] "Angular," [Online]. Available: (<https://angular.io/docs>) [Accessed 1 5 2024].
- [6] "SQL Server migration documentation," [Online]. Available: (<https://learn.microsoft.com/vi-vn/sql/sql-server/migrate/?view=sql-server-ver16>) [Accessed 30 4 2024].