

ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ thông tin và Truyền thông



BÁO CÁO BÀI TẬP LỚN

Hệ thống quản lý trung tâm chăm sóc thú cưng

Học phần: Phát triển phần mềm theo chuẩn ITSS

GVHD: ThS.Nguyễn Mạnh Tuấn

Nhóm 21

Họ và tên	MSSV
Nguyễn Trí Đức	20225812
Đỗ Đắc Duy	20225827
Phan Hoàng Hải	20225715
Nguyễn Minh Khôi	20225642
Nguyễn Bá Hoàng	20225844

Mục Lục

1. Sản phẩm	2
2. Phân công công việc	3
3. Phân tích yêu cầu	3
3.1. Giới thiệu	3
3.1.1. Mục đích	3
3.1.2. Phạm vi	4
3.1.3. Từ điển thuật ngữ	4
3.1.4. Tài liệu tham khảo	4
3.2. Mô tả tổng quan	4
3.2.1. Các tác nhân	4
3.2.2. Biểu đồ use case tổng quan	4
3.2.3. Biểu đồ use case phân rã	6
3.2.4. Quy trình nghiệp vụ	8
3.3. Đặc tả các chức năng	13
3.3.1. Đặc tả UC001 “đăng nhập”	13
3.3.2. Đặc tả UC002 “đăng ký”	15
3.3.3. Đặc tả UC003 “đăng ký khám”	18
3.3.4. Đặc tả UC004 “CRUD thú cưng”	20
3.3.5. Đặc tả UC005 “đăng ký dịch vụ trông giữ thú cưng”	23
3.3.6. Đặc tả UC006 “đăng ký dịch vụ vệ sinh”	24
3.3.7. Đặc tả UC007 “khám cho thú cưng”	24
3.3.8. Đặc tả UC008 “quản lí người dùng”	25
3.3.9. Đặc tả UC009 “quản lí thú cưng”	25
	2

3.3.10. Đặc tả UC010 “quản lí thuốc”	26
3.3.11. Đặc tả UC011 “quản lí dịch vụ”	27
3.3.12. Đặc tả UC012 “quản lí chuồng”	27
3.3.13. Đặc tả UC013 “quản lí lịch khám”	27
3.3.14. Đặc tả UC014 “khám cho thú cưng”	28
3.4. Các yêu cầu khác	28
3.4.1. Chức năng (Functionality)	29
3.4.2. Tính dễ dùng (Usability)	29
3.4.3. Các yêu cầu khác	29
4. Thiết kế kiến trúc	29
4.1. Thiết kế kiến trúc cho UC “đăng nhập”	29
4.2. Thiết kế kiến trúc cho use case “đăng ký”	31
4.3. Thiết kế kiến trúc cho UC “quản lí tài khoản người dùng”	34
4.4. Thiết kế kiến trúc cho UC “đăng ký sử dụng dịch vụ vệ sinh/làm đẹp” (tương tự “đăng ký khám”)	41
4.5. Thiết kế kiến trúc cho UC “Đăng ký khám”	41
4.6. Thiết kế kiến trúc cho UC “khám bệnh”	43
4.7. Thiết kế kiến trúc cho UC “tìm kiếm người dùng”	45
5. Thiết kế chi tiết	47
5.1. GUI Design	47
5.1.1. Thiết kế giao diện người dùng (User Interface Design)	47
5.1.2. Chuẩn hóa cấu hình màn hình	47
5.1.3. Tạo các ảnh màn hình	48
5.2. Class Diagram	59
5.2.1. Class Diagram cho đăng ký, đăng nhập, quản lí tài khoản người dùng	59

5.3. Data Modeling	60
5.3.1. Mô hình khái niệm (Conceptual Data Model)	60
5.3.2. Thiết kế cơ sở dữ liệu(Logical Data Model)	60
6. Test case	61
7. Nguyên tắc thiết kế	61
7.1. Áp dụng Design Concepts	61
7.1.1. Coupling	61
7.1.2. Cohesion	63
7.2. Áp dụng Design Principles SOLID	64
8. Hướng dẫn sử dụng, cài đặt, kiểm thử	66
8.1. Hướng dẫn tải IntelliJ	66
8.2. Tải MySQL	67
8.3. Clone project về local	67
8.4. Set database và môi trường	68
8.5. Chạy dự án	68

1. Sản phẩm

- Link github mã nguồn:

2. Phân công công việc

Thành viên	Nhiệm vụ	Tiến độ
Nguyễn Trí Đức	Code backend	Hoàn thành
Đỗ Đắc Duy	Viết báo cáo Kiểm thử	Hoàn thành
Phan Hoàng Hải	Code Frontend Thiết kế giao diện	Hoàn thành
Nguyễn Minh Khôi	Viết báo cáo Kiểm thử	Hoàn thành
Nguyễn Bá Hoàng	Thiết kế databases Code backend	Hoàn thành

3. Phân tích yêu cầu

3.1. Giới thiệu

3.1.1.Mục đích

Mục tiêu của tài liệu này là để cung cấp mô tả chi tiết, cụ thể về các chức năng của phần mềm hỗ trợ việc quản lý chăm sóc thú cưng, đồng thời mô tả cả thông tin chi tiết phục vụ cho việc phát triển hệ thống trong tương lai. Tài liệu sẽ nói lên đầy đủ về các ràng buộc của hệ thống, giao diện người dùng và các ứng dụng bên ngoài liên kết với hệ thống (có thể là các ví điện tử, tài khoản ngân hàng).

Đây là tài liệu tham khảo đầu vào cho các giai đoạn thiết kế, lập trình, kiểm thử trong quy trình sản xuất phần mềm.

3.1.2. Phạm vi

Phần mềm hệ thống quản lý chăm sóc thú cưng là một phần mềm được thiết kế để hỗ trợ quản lý các hoạt động chăm sóc thú cưng trong các cơ sở thú y, phòng khám thú y hoặc các cửa hàng bán lẻ thú cưng. Người dùng có thể đăng ký để tạo ra tài khoản cho mình, sau đó có thể đăng nhập để sử dụng các chức năng của hệ thống.

3.1.3. Từ điển thuật ngữ

Chưa có

3.1.4. Tài liệu tham khảo

Chưa có

3.2. Mô tả tổng quan

3.2.1. Các tác nhân

Phần mềm có các tác nhân là Khách, Người dùng, Bác sĩ và Admin.

Khách là vai trò của người dùng khi chưa đăng nhập vào hệ thống. Người dùng là vai trò của một người dùng bình thường sau khi đã đăng nhập thành công vào hệ thống.

Bác sĩ có vai trò chức năng khám bệnh cho thú cưng, cũng như kê thuốc.

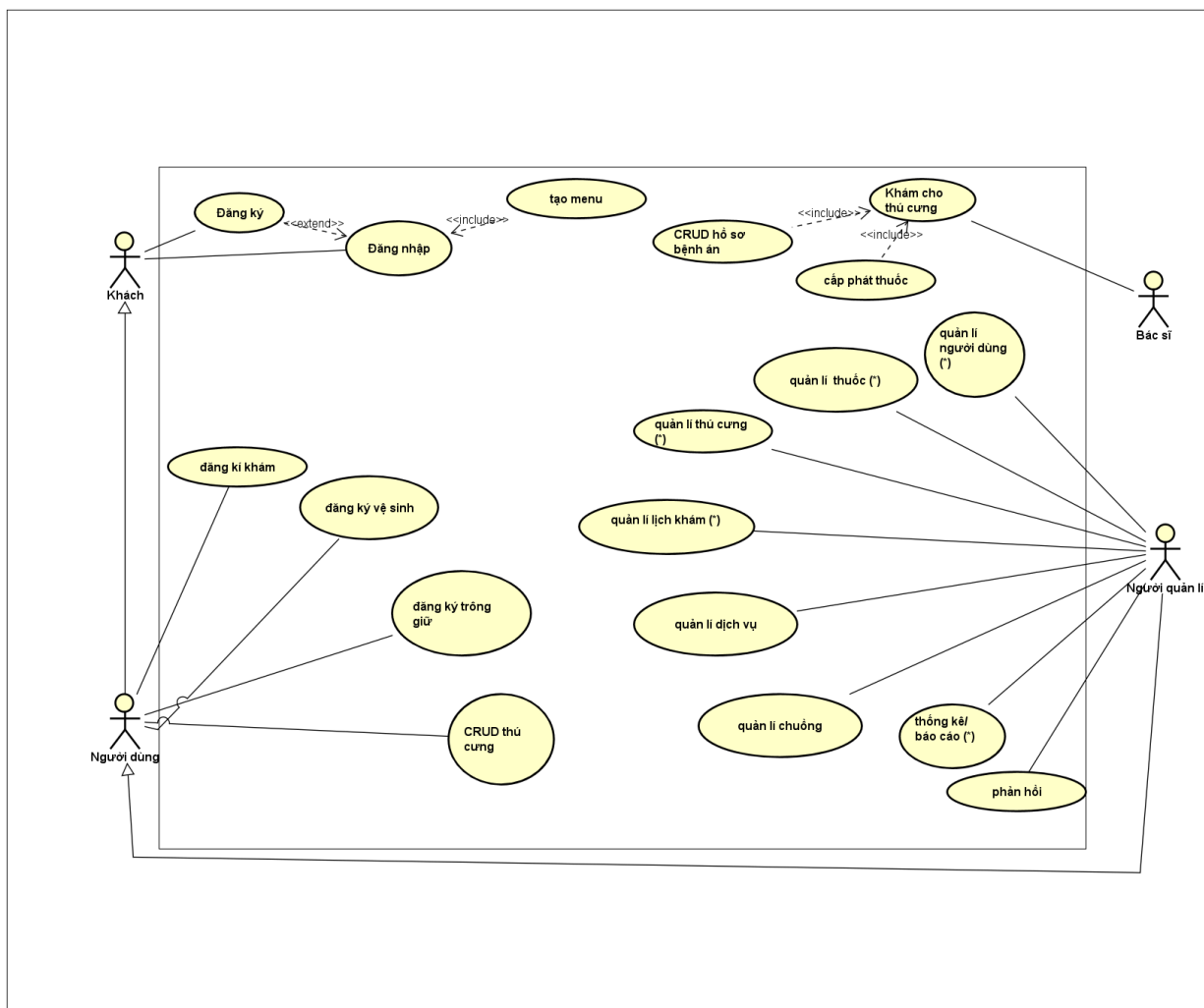
Admin có vai trò quản lý người dùng và các tính năng chung của hệ thống.

3.2.2. Biểu đồ use case tổng quan

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 khách đăng nhập thành công, hệ thống sẽ mở giao diện chứa các chức năng như đăng ký khám, vệ sinh, trông giữ hay là xem thú cưng của bản thân. Khi khách yêu cầu thiết lập lại mật khẩu, hệ thống thực hiện chỉ dẫn thiết lập lại mật khẩu cho người dùng.

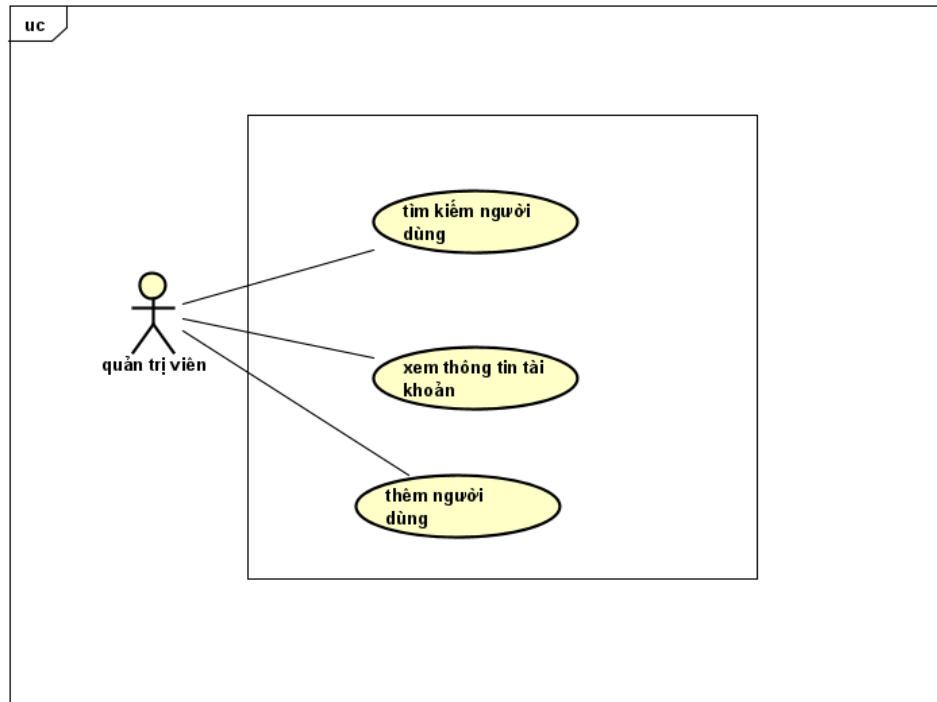
Sau khi đăng nhập, người dùng có thể xem và cập nhật thông tin cá nhân của mình, và có thể thay đổi thay đổi mật khẩu.

Quản trị viên có thể quản lý người dùng, quản lý thú cưng, quản lý thuốc, quản lý dịch vụ, quản lý chuồng, quản lý lịch khám cũng như phản hồi của người dùng.

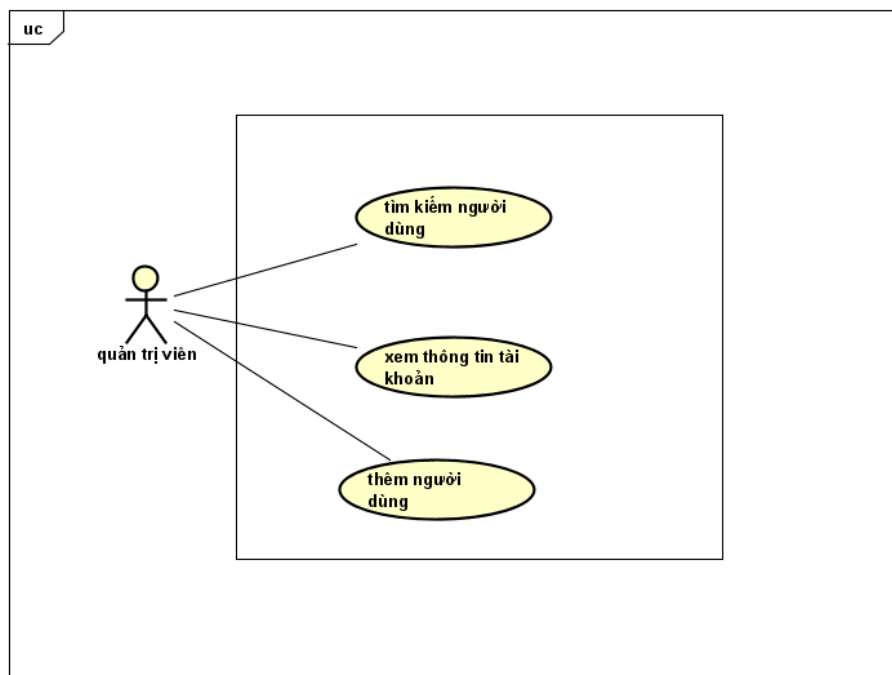


3.2.3. Biểu đồ use case phân rã

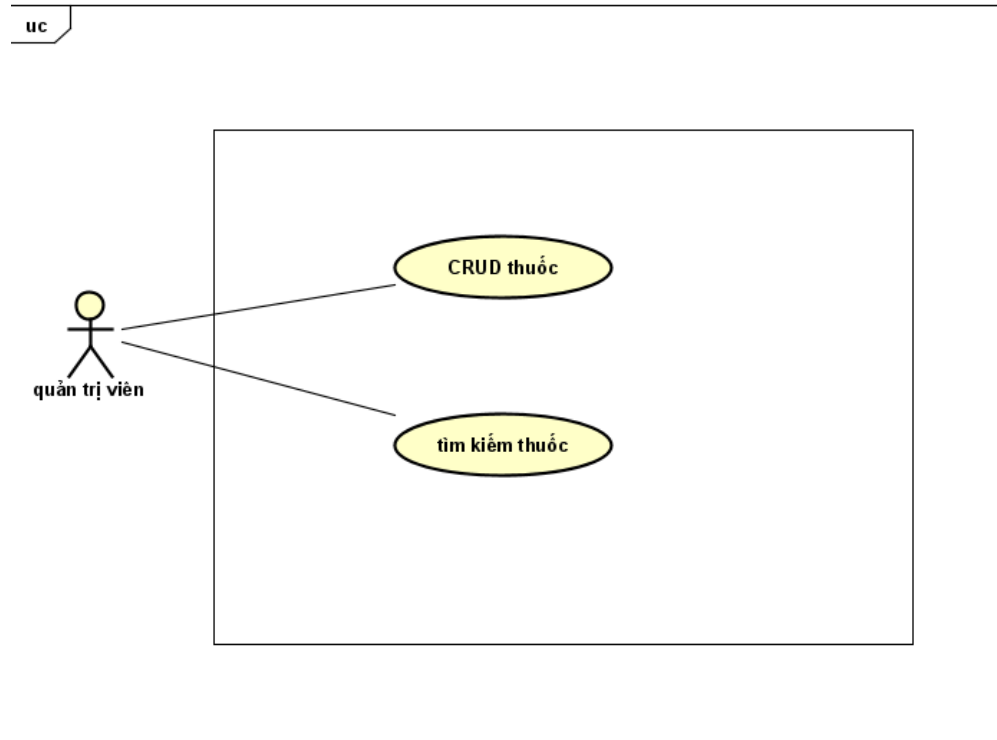
3.2.3.1. Phân rã UC “quản lí người dùng”



3.2.3.2. Phân rã UC “quản lý thú cưng”

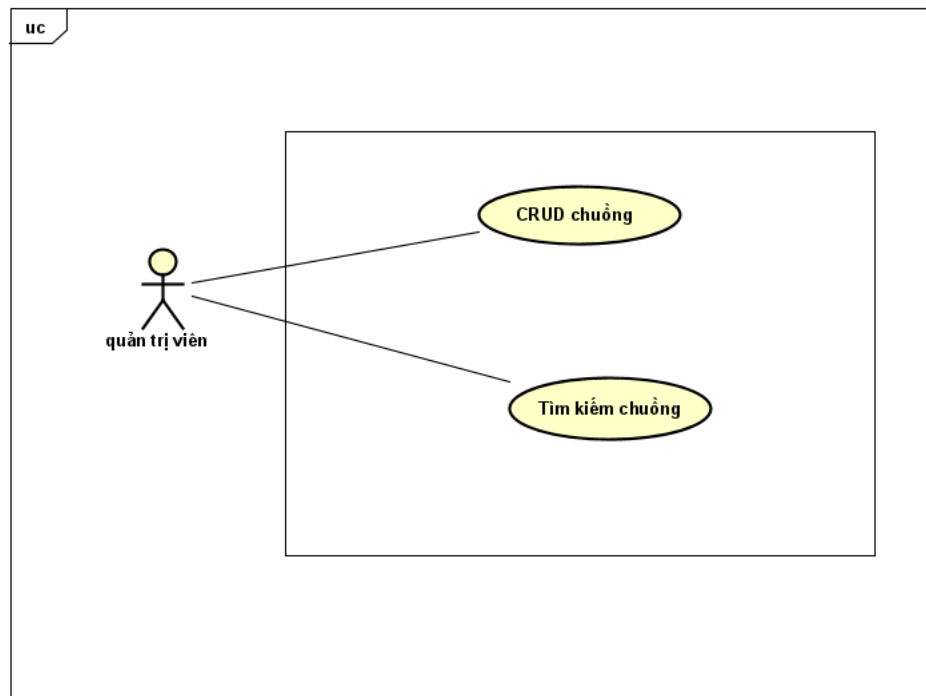


3.2.3.3. Phân rã UC “quản lý thuốc”

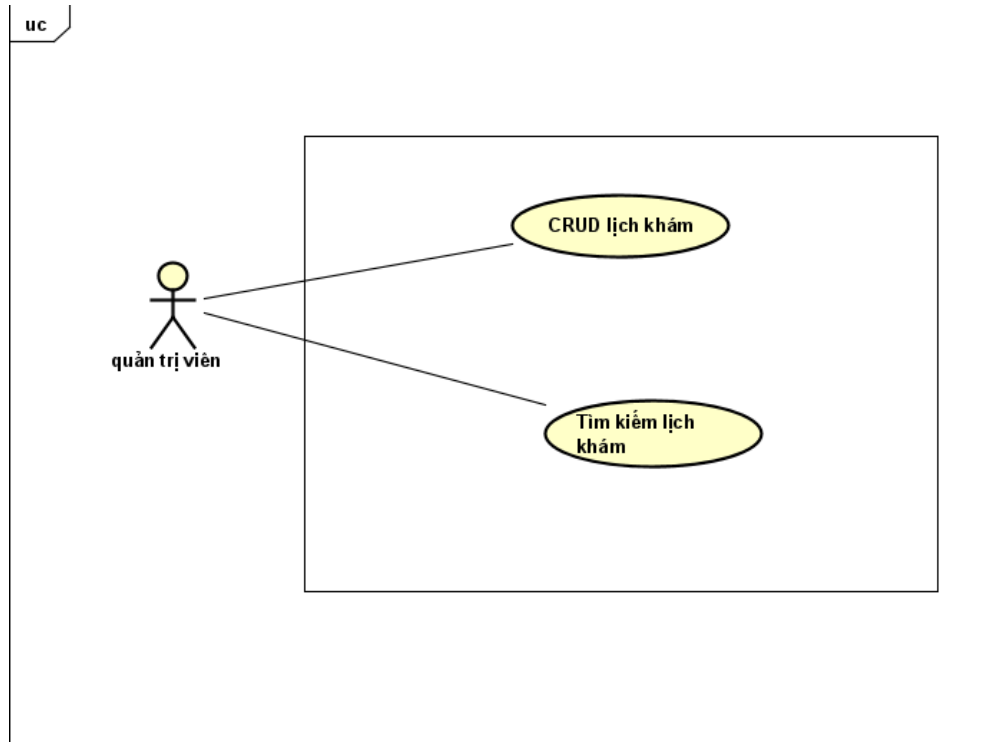


3.2.3.4. Phân rã UC “quản lý dịch vụ”

3.2.3.5. Phân rã UC “quản lý chuồng”



3.2.3.6. Phân rã UC “quản lý lịch khám”



3.2.4. Quy trình nghiệp vụ

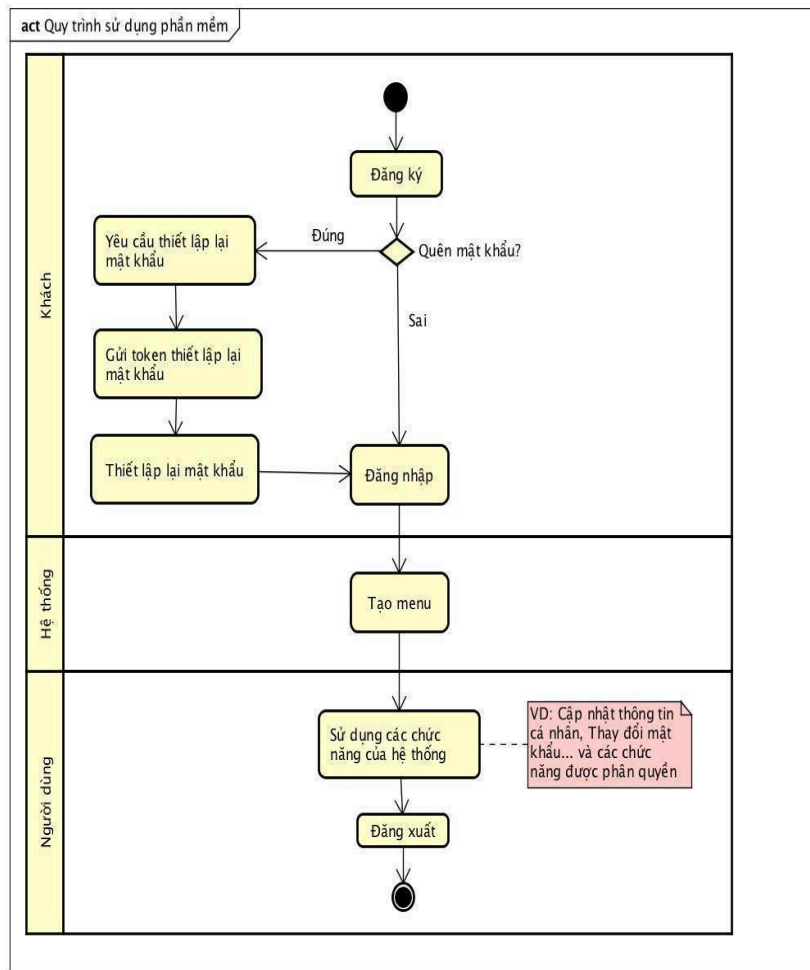
Trong phân hệ này, có những quy trình nghiệp vụ chính: Quy trình sử dụng phần mềm của khách và người dùng, Quy trình quản lý của admin, và quản lý giao dịch thu chi.

Chi tiết về hành động trong các quy trình này được mô hình hoá trong các mục con của từng quy trình.

3.2.4.1. Quy trình sử dụng phần mềm

Khách có thể đăng ký để tạo ra tài khoản cho mình. Sau đó có thể đăng nhập để sử dụng các chức năng của phần mềm. Nếu khách quên mật khẩu, khách có thể yêu cầu hệ thống cho phép thiết lập lại mật khẩu.

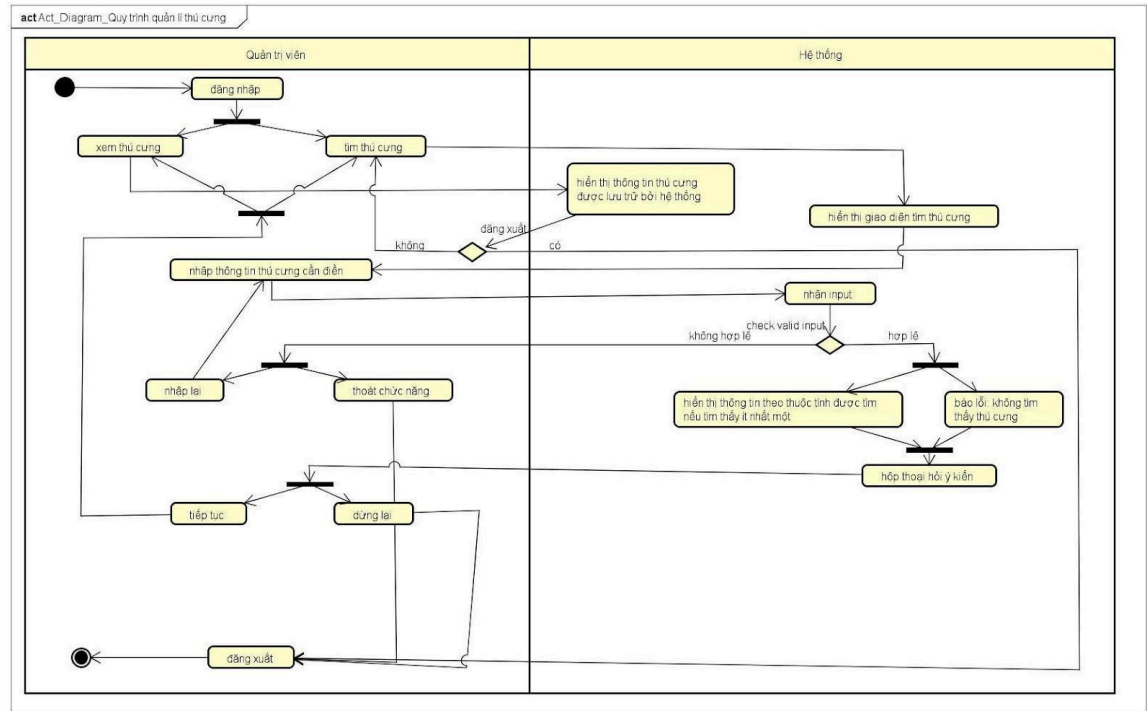
Sau khi đăng nhập thành công, người dùng có thể sử dụng các chức năng để có thể sử dụng các chức năng như xem và cập nhật thông tin cho thú cưng, thay đổi mật khẩu và các chức năng được người quản trị cấp phép



3.2.4.2. Quy trình quản lý thú cưng được chủ nuôi đăng ký

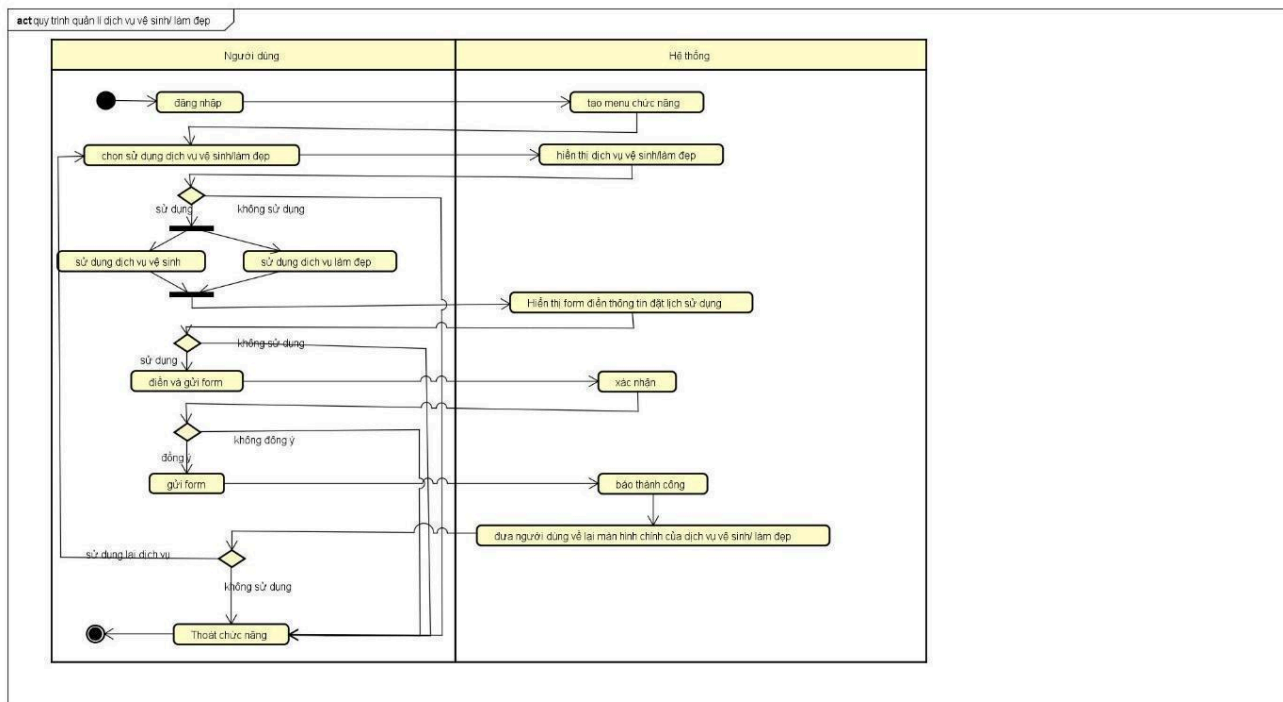
Hệ thống sẽ lưu trữ thông tin về các thú cưng được đăng ký ở trên hệ thống. Sau khi người dùng đăng nhập thành công dưới vai trò quản trị viên họ có thể xem danh sách các thú cưng hiện được lưu trữ cũng như có thể tìm thú cưng để thuận tiện cho việc quản lý và tìm kiếm cũng như tránh được việc người khác có thể tùy ý xem những thông tin riêng tư này khi không được phép.

Tất cả các thông tin trên hệ thống được lưu trữ một cách an toàn và bảo mật, và chỉ có chủ nuôi và nhân viên có quyền truy cập vào hệ thống.



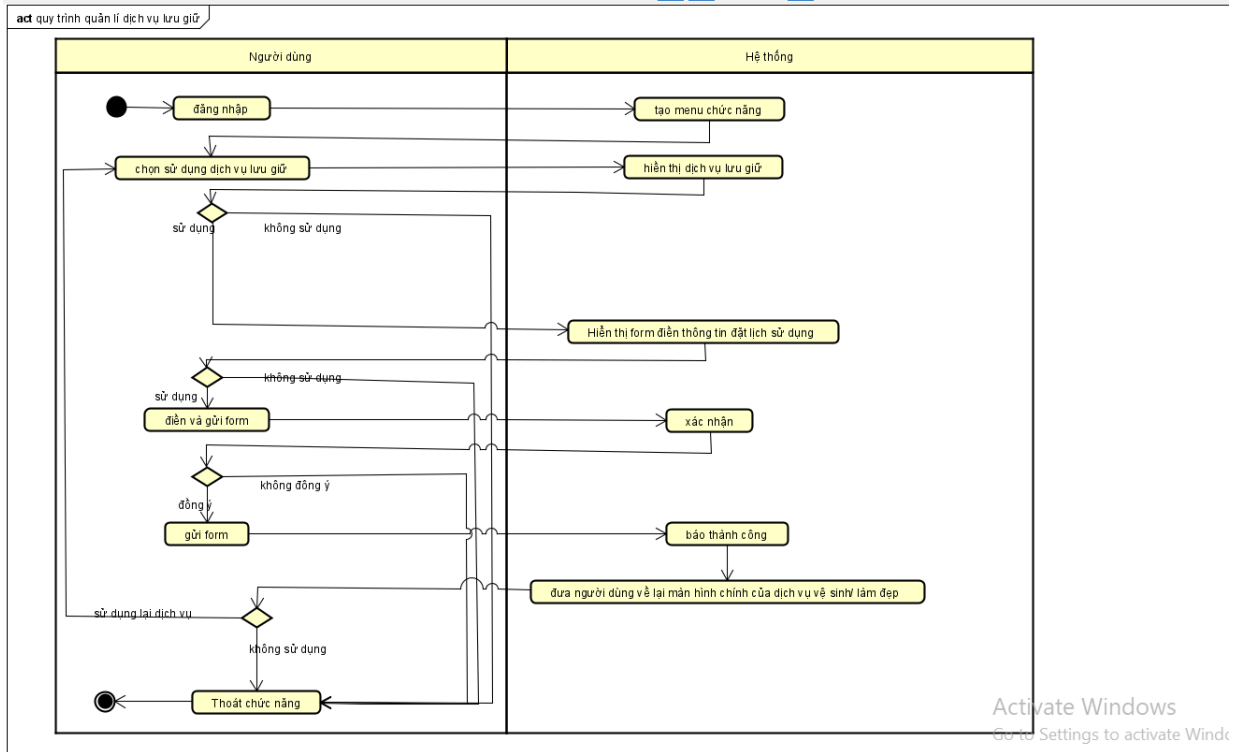
3.2.4.3. Quy trình quản lí dịch vụ

Hệ thống sẽ lưu trữ các dịch vụ vệ sinh/làm đẹp. Ngoài ra hệ thống còn có thể tạo mới dịch vụ mới; cập nhật, đọc và xóa dịch vụ hiện có. Sau khi người dùng đăng nhập thành công chức năng và muốn sử dụng dịch vụ vệ sinh/ làm đẹp thì hệ thống sẽ tạo ra menu gồm thông tin và các dịch vụ mà trung tâm hiện có để người dùng sử dụng. Sau đó hệ thống sẽ cho phép người dùng chọn sử dụng các dịch vụ vệ sinh/làm đẹp hoặc thoát ra để sử dụng chức năng khác của hệ thống

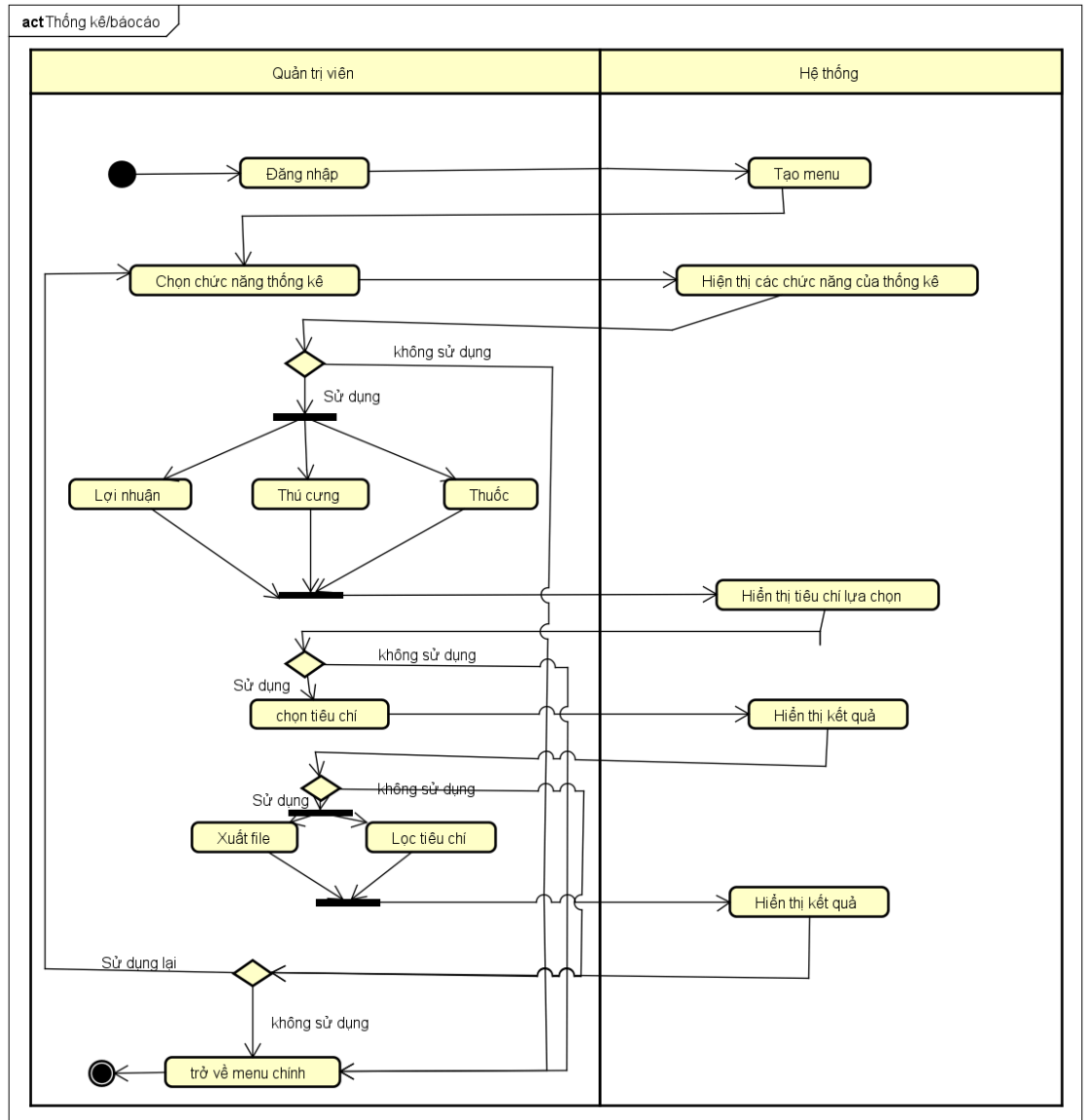


3.2.4.4. Quy trình dịch vụ lưu trữ thú cưng

Sau khi người dùng đăng nhập thành công chức năng và muốn sử dụng dịch vụ lưu giữ thì hệ thống sẽ tạo ra menu gồm thông tin và các dịch vụ mà trung tâm hiện có để người dùng sử dụng. Sau đó hệ thống sẽ cho phép người dùng chọn sử dụng các dịch vụ lưu giữ hoặc thoát ra để sử dụng chức năng khác của hệ thống



3.2.4.5. Quy trình thống kê/báo cáo



3.3. Đặc tả các chức năng

Chi tiết về các use case được đưa ra trong phần 2 được đặc tả trong các phần dưới đây.

3.3.1.Đặc tả UC001 “đăng nhập”

Mã Use case	UC001	Tên Use case	Đăng nhập
Tác nhân	Khách		

Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Khách	chọn chức năng Đăng nhập
	0.	Hệ thống	hiển thị giao diện đăng nhập
	0.	Khách	nhập email và mật khẩu (mô tả phía dưới *)
	0.	Khách	yêu cầu đăng nhập
	0.	Hệ thống	kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa
	0.	Hệ thống	kiểm tra email và mật khẩu có hợp lệ do khách nhập trong hệ thống hay không
	0.	Hệ thống	gọi use case “Tạo menu” kèm email của khách đăng nhập
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	6a.	Hệ thống	thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu
	7a.	Hệ thống	thông báo lỗi: Email và/hoặc mật khẩu chưa đúng nếu không tìm thấy email và mật khẩu trong hệ thống

	7b.	Hệ thống	thông báo lỗi: Tài khoản bị khoá, nếu email/mật khẩu đúng như tài khoản đang bị admin khoá.
	7c1.	Hệ thống	gọi use case “Thay đổi mật khẩu theo yêu cầu” nếu đúng email và mật khẩu; nhưng người dùng được đánh dấu là cần thay đổi mật khẩu
	7c2.	Hệ thống	gọi use case “Tạo menu” kèm email của khách đăng nhập
Hậu điều kiện	Không		

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1.	Email		Có		h.anh@gmail.com
0.	Mật khẩu		Có		ToiLa12#\$

3.3.2.Đặc tả UC002 “đăng ký”

Mã Use case	UC002	Tên Use case	Đăng ký
Tác nhân	Khách		
Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Khách	chọn chức năng Đăng ký
	0.	Hệ thống	hiển thị giao diện đăng ký

	0.	Khách	nhập các thông tin cá nhân (mô tả phía dưới *)
	0.	Khách	yêu cầu đăng ký
	0.	Hệ thống	kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa
	0.	Hệ thống	kiểm tra địa chỉ email của khách có hợp lệ không
	0.	Hệ thống	kiểm tra mật khẩu nhập lại và mật khẩu có trùng nhau hay không
	0.	Hệ thống	kiểm tra mật khẩu có đủ mức độ an toàn hay không
	0.	Hệ thống	lưu thông tin tài khoản và thông báo đăng ký thành công
Luồng sự kiện thay thế			
	STT	Thực hiện bởi	Hành động
	6a.	Hệ thống	thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu
	7a.	Hệ thống	thông báo lỗi: Địa chỉ email không hợp lệ nếu địa chỉ email không hợp lệ
	8a.	Hệ thống	thông báo lỗi: Mật khẩu xác nhận không trùng với Mật khẩu nếu hai mật khẩu không trùng nhau
	9a.	Hệ thống	thông báo lỗi: Mật khẩu cần đảm bảo độ an toàn nếu mật khẩu không đảm bảo độ an toàn được quy định bởi hệ thống (có ít nhất 8 ký tự, trong đó có cả chữ

		cái thường và chữ cái hoa, ít nhất 1 chữ số và 1 ký tự đặc biệt)
Hậu điều kiện	Không	

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1.	Họ		Có		Nguyễn
0.	Tên		Có		Hoàng Anh
0.	Email		Có	Địa chỉ email hợp lệ	h.anh@gmail.com
0.	Mật khẩu		Có	Ít nhất 8 ký tự, gồm cả chữ cái hoa và thường, ít nhất 1 chữ số và ký tự đặc biệt	ToiLa12#\$
0.	Xác nhận mật khẩu		Có	Trùng với Mật khẩu	ToiLa12#\$

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
0.	Ngày sinh	Chỉ chọn trong DatePicker	Có	Ngày hợp lệ	12/12/1991
0.	Giới tính	Lựa chọn Nam hoặc Nữ	Có		Lựa chọn Nữ
0.	Số ĐT		Không	Số ĐT hợp lệ chỉ gồm số, dấu cách, dấu chấm hoặc dấu gạch ngang	0913.123.321

3.3.3.Đặc tả UC003 “đăng ký khám”

Mã Use case	UC003	Tên Use case	Đăng ký khám
Tác nhân	Người dùng		
Tiền điều kiện	Khách đăng nhập thành công với vai trò người dùng		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Chủ nuôi	chọn chức năng Lịch khám
	0.	Hệ thống	hiển thị giao diện Lịch khám

	0.	Chủ nuôi	chọn vào chức năng Đăng ký lịch khám
	0.	Hệ Thống	hiển thị bảng giao diện Đăng ký lịch khám
	0.	Chủ nuôi	nhập các thông tin đăng ký (mô tả phía dưới *)
	0.	Chủ nuôi	Yêu cầu đăng ký
	0.	Hệ thống	kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa
	1.	Hệ thống	lưu thông tin đăng ký và thông báo đăng ký thành công
Luồng sự kiện thay thế			
	STT	Thực hiện bởi	Hành động
	8a.	Hệ thống	Thông báo lỗi nếu chủ nuôi nhập thiếu thông tin
	9a.	Hệ thống	Thông báo lỗi: nếu lịch khám chủ nuôi đăng ký bị trùng lặp với lịch khám đã có trong hệ thống
Hậu điều kiện	Không		

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1.	Tên thú cưng		Có		Dog
2.	Ngày khám		Có		30/7/2023

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
3.	Thời gian		Có		11:00

3.3.4.Đặc tả UC004 “CRUD thú cưng”

Mã Use case	UC004	Tên Use case	CRUD thú cưng
Tác nhân	Người dùng		
Tiền điều kiện	Khách đăng nhập thành công với vai trò người dùng		
Thêm (C):			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Chọn chức năng thêm mới thông tin thú cưng
	0.	Hệ thống	Hiển thị form input bao gồm name, category, age, color, gender
	3.	Người dùng	Điền đầy đủ thông tin và nhấn nút thêm mới
	0.	Hệ thống	Kiểm tra tính hợp lệ
	1.	Hệ thống	Hiển thị thông báo thành công
Luồng sự kiện thay thế	5a1	Hệ thống	Thông báo lỗi nếu người dùng nhập thiếu thông tin
	6a1	Hệ thống	Thông báo lỗi nếu thú cưng đã có
Xem (R):			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động

	1.	Người dùng	Chọn chức năng hiển thị thông tin thú cưng
	0.	Hệ thống	Hiển thị danh sách các thú cưng
	0.	Người dùng	Chọn 1 thú cưng để xem
	0.	Hệ thống	Hiển thị chi tiết thông tin thú cưng
Luồng sự kiện thay thế	3a	Hệ thống	Nếu không có thú cưng nào, hệ thống báo lỗi

Cập nhật (U):

	STT	Thực hiện bởi	Hành động
Luồng sự kiện chính	1.	Người dùng	Chọn chức năng cập nhật thông tin thú cưng
	0.	Hệ thống	hiển thị danh sách thú cưng
	0.	Người dùng	Chọn 1 thú cưng để cập nhật thông tin
	0.	Hệ thống	Hiển thị form để người dùng cập nhật thông tin thú cưng
	1.	Người dùng	Cập nhật thông tin và nhấn nút cập nhật

	2.	Hệ thống	Kiểm tra tính hợp lệ. Cập nhật thông tin
Luồng sự kiện thay thế	3a	Hệ thống	Nếu không có thú cưng, hệ thống hiển thị thông báo
	7a	Hệ thống	Nếu thông tin không hợp lệ, đưa ra thông báo cho người dùng

Xóa (D):

	STT	Thực hiện bởi	Hành động
Luồng sự kiện chính	1.	Người dùng	Chọn chức năng xóa thông tin thú cưng
	0.	Hệ thống	Hiển thị danh sách thú cưng
	0.	Người dùng	Chọn 1 thú cưng để xóa
	0.	Hệ thống	Yêu cầu xác nhận
	0.	Hệ thống	Người dùng xác nhận xóa, hệ thống xóa thông tin thú cưng
Luồng sự kiện thay thế	3a	Hệ thống	Nếu không có thú cưng nào, hệ thống đưa ra thông báo
	6a	Hệ thống	Nếu người dùng không xác nhận, hệ thống không thực hiện xóa

Hậu điều kiện

Không

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1.	Tên thú cưng		Có		Dog
2.	Category		Có		Hoàng Anh
3.	Age		Có		10
4.	Color		Có		Red

3.3.5.Đặc tả UC005 “đăng ký dịch vụ trông giữ thú cưng”

Mã Use case	UC005	Tên Use case	Đăng ký dịch vụ trông giữ thú cưng	
Tác nhân	Người dùng			
Tiền điều kiện	Khách đã đăng nhập thành công với vai trò người dùng			
Luồng sự kiện chính (Thành công)				
	STT	Thực hiện bởi	Hành động	
	1. .	Người dùng	chọn chức năng	Dịch vụ trông giữ
	2.	Hệ thống	hiển thị giao diện	Dịch vụ trông giữ
	3.	Người dùng	chọn vào chức năng	Đăng ký dịch vụ
	4.	Hệ Thống	hiển thị bảng giao diện	Đăng ký dịch vụ
5.	Người dùng	nhập các thông tin đăng ký (mô tả phía dưới *)		

	6.	Người dùng	Yêu cầu đăng ký
	7.	Hệ thống	kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa
	8.	Hệ thống	lưu thông tin đăng ký và thông báo đăng ký thành công
Luồng sự kiện thay thế			
	STT	Thực hiện bởi	Hành động
	7a.	Hệ thống	Thông báo lỗi nếu người dùng nhập thiếu thông tin
Hậu điều kiện	Không		

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Tên thú cưng		Có		dog
2	Ngày		Có		30/7/2023
3	Thời gian		Có		11:00

3.3.6.Đặc tả UC006 “đăng ký dịch vụ vệ sinh”

Giống như UC005

3.3.7.Đặc tả UC007 “khám cho thú cưng”

Mã Use case	UC007	Tên Use case	Khám cho thú cưng
Tác nhân	Bác sĩ		

Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Bác sĩ	Chọn chức năng khám bệnh
	2.	Hệ thống	Hiển thị giao diện khám bệnh
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
Hậu điều kiện	Không		

3.3.8.Đặc tả UC008 “quản lí người dùng”

Giống như UC004 “CRUD thú cưng” , chỉ khác ở đầu vào và đầu ra dữ liệu

* Dữ liệu đầu vào và ra của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Tên đăng nhập				Hoanghai
2	Mật khẩu				123456
3	Họ và tên				Nguyen Hoang Hai
4	Email				Hai@gmail.com
5	Ngày sinh				17/12/2002
6	Giới tính				Male

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
7	Vai trò				Customer

3.3.9.Đặc tả UC009 “quản lí thú cưng”

Giống như UC004 “CRUD thú cưng” , chỉ khác ở đầu vào và đầu ra dữ liệu

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Name				Pig
2	Category				Dog
3	Age				3
4	Color				Red
5	Gender				Male
6	Username				Hai

3.3.10. Đặc tả UC010 “quản lí thuốc”

Giống như UC004 “CRUD thú cưng” , chỉ khác ở đầu vào và đầu ra dữ liệu

* Dữ liệu đầu vào, đầu ra của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Tên thuốc		Có		Phetanol
2	Nơi sản xuất		Có		Việt nam

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
3	Ngày sản xuất		Có		30/7/2023
4	Ngày hết hạn		Có		30/7/2024
5	Công dụng		Có		Chữa đau đầu
6	Hướng dẫn sử dụng		Có		Sau ăn 30 phút
7	Tác dụng phụ				Buồn nôn
8	Giá		Có		50000 vnd

3.3.11. Đặc tả UC011 “quản lí dịch vụ”

Giống như UC004 “CRUD thú cưng” , chỉ khác ở đầu vào và đầu ra dữ liệu

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Tên dịch vụ		Có		Tỉa lông
2	Giá thành		Có		100000 vnd
3	Quy trình		Có		

3.3.12. Đặc tả UC012 “quản lí chuồng”

Giống như UC004 “CRUD thú cưng” , chỉ khác ở đầu vào và đầu ra dữ liệu

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Id chuồng		Có		1
2	Trạng thái chuồng		Có		Có sẵn
3	Thông tin pet				
4	Vị trí				Tầng 2

3.3.13. Đặc tả UC013 “quản lý lịch khám”

Giống như UC004 “CRUD thú cưng”, chỉ khác ở đầu vào và đầu ra dữ liệu

* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1	Thông tin pet		Có		Pig
2	Thời gian		Có		11:00
3	Ngày		Có		30/7/2023

3.3.14. Đặc tả UC014 “khám cho thú cưng”

Mã Use case	UC014	Tên Use case	Khám cho thú cưng
Tác nhân	Bác sĩ		
Tiền điều kiện	Đăng nhập vào hệ thống với vai trò bác sĩ		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Bác sĩ	Chọn thú cưng đã đặt lịch

	2	Hệ thống	hiển thị form điền thông tin về sức khỏe thú cưng
	3	Bác sĩ	Điền form thông tin sức khỏe thú cưng
	4	Hệ thống	Lưu lại thông tin bác sĩ đã điền
Luồng sự kiện thay thế			
	STT	Thực hiện bởi	Hành động
Hậu điều kiện	Không		

*

3.4. Các yêu cầu khác

3.4.1.Chức năng (Functionality)

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

3.4.2.Tính dễ dùng (Usability)

Các chức năng cần được thiết kế sao cho dễ thao tác, đi kèm với hướng dẫn sử dụng. Cần 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.

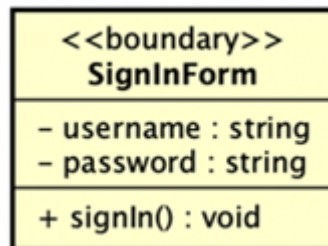
3.4.3. Các yêu cầu khác

Hiệu năng: Với mỗi thao tác, xử lý nhanh gọn không quá 1s

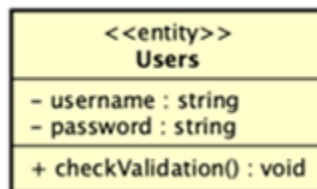
4. Thiết kế kiến trúc

4.1. Thiết kế kiến trúc cho UC “đăng nhập”

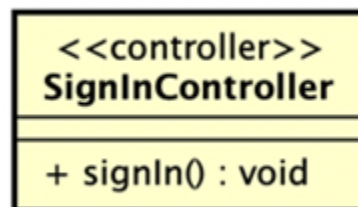
Lớp boundary



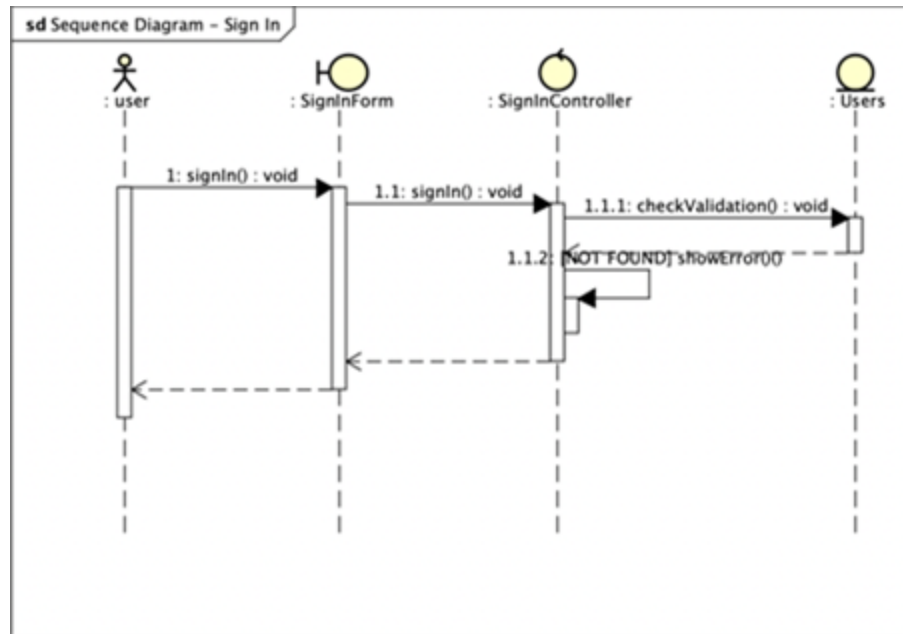
Lớp entity



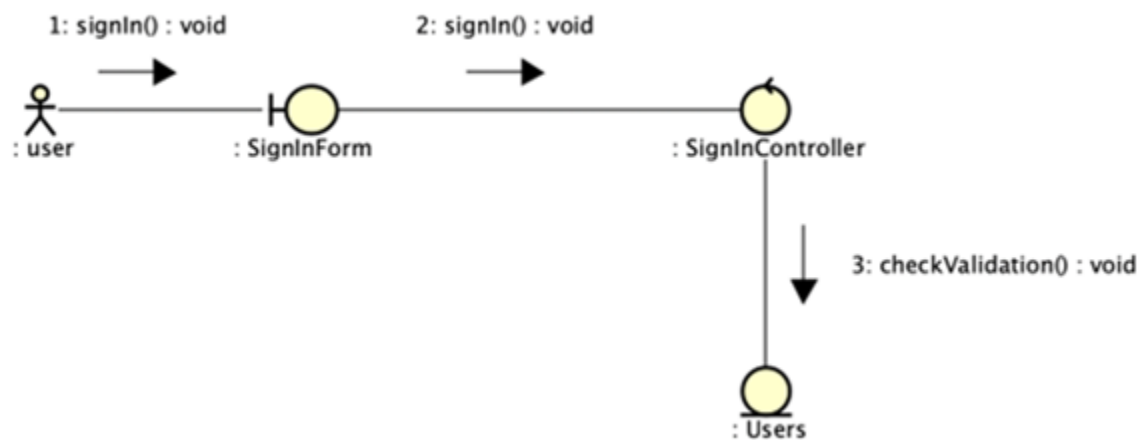
Lớp control



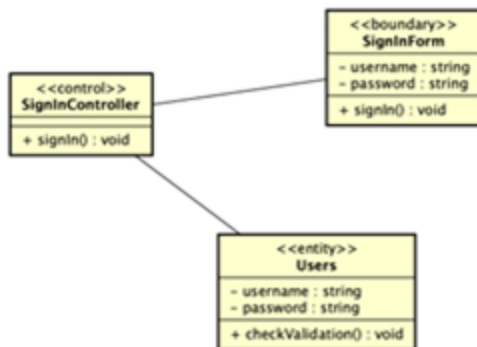
Biểu đồ trình tự



Biểu đồ giao tiếp

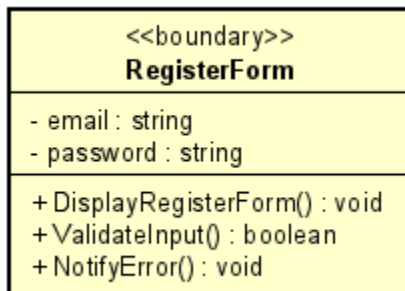


Biểu đồ lớp phân tích

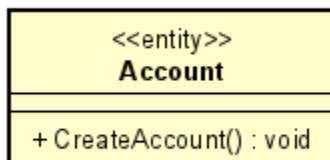


4.2. Thiết kế kiến trúc cho use case “đăng ký”

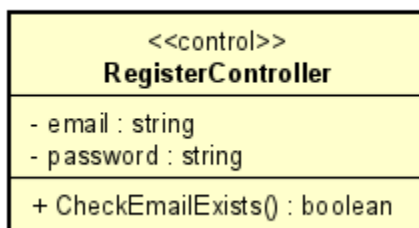
Lớp boundary



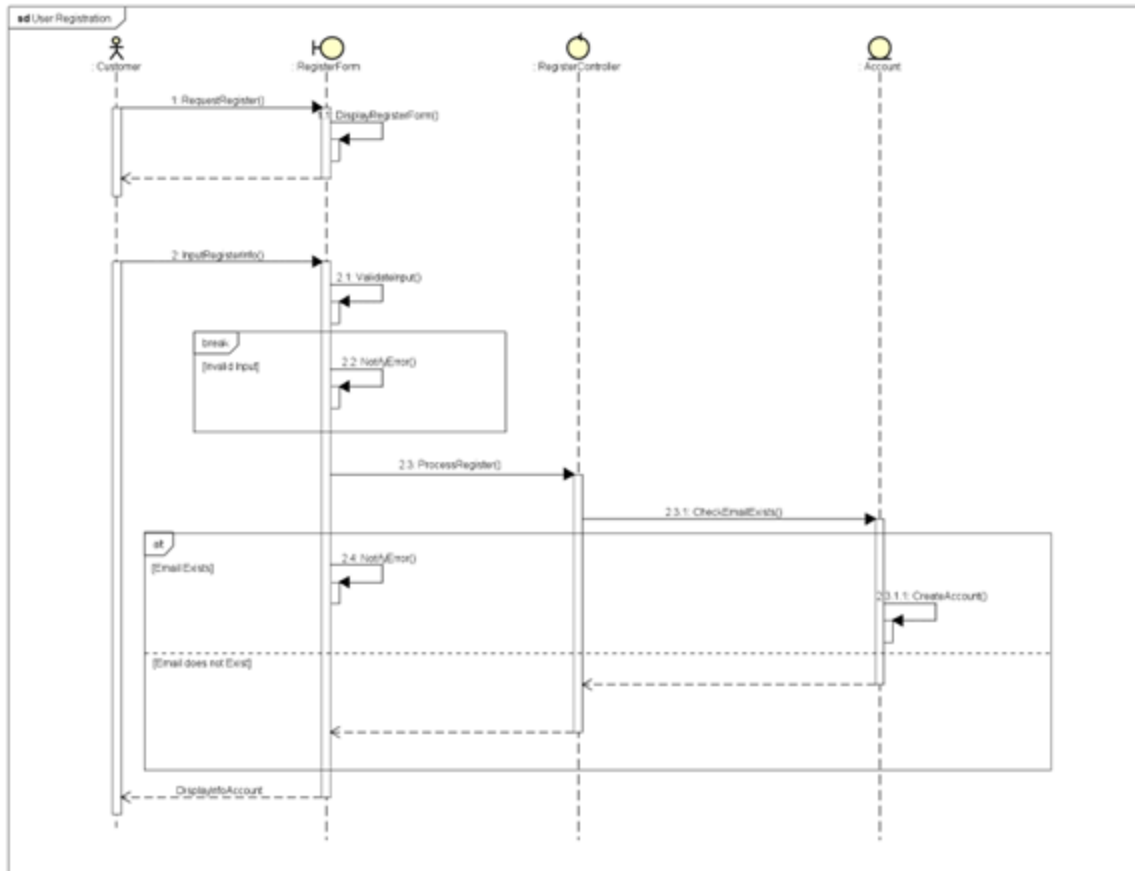
Lớp entity



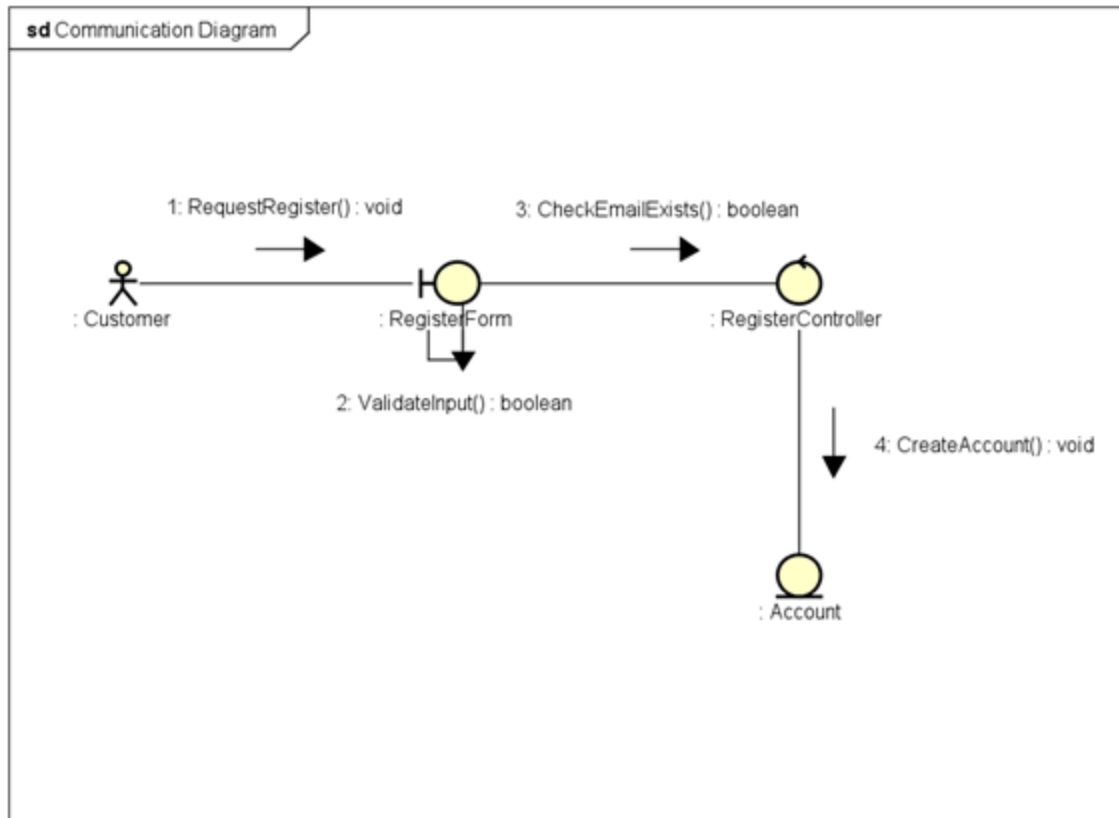
Lớp control



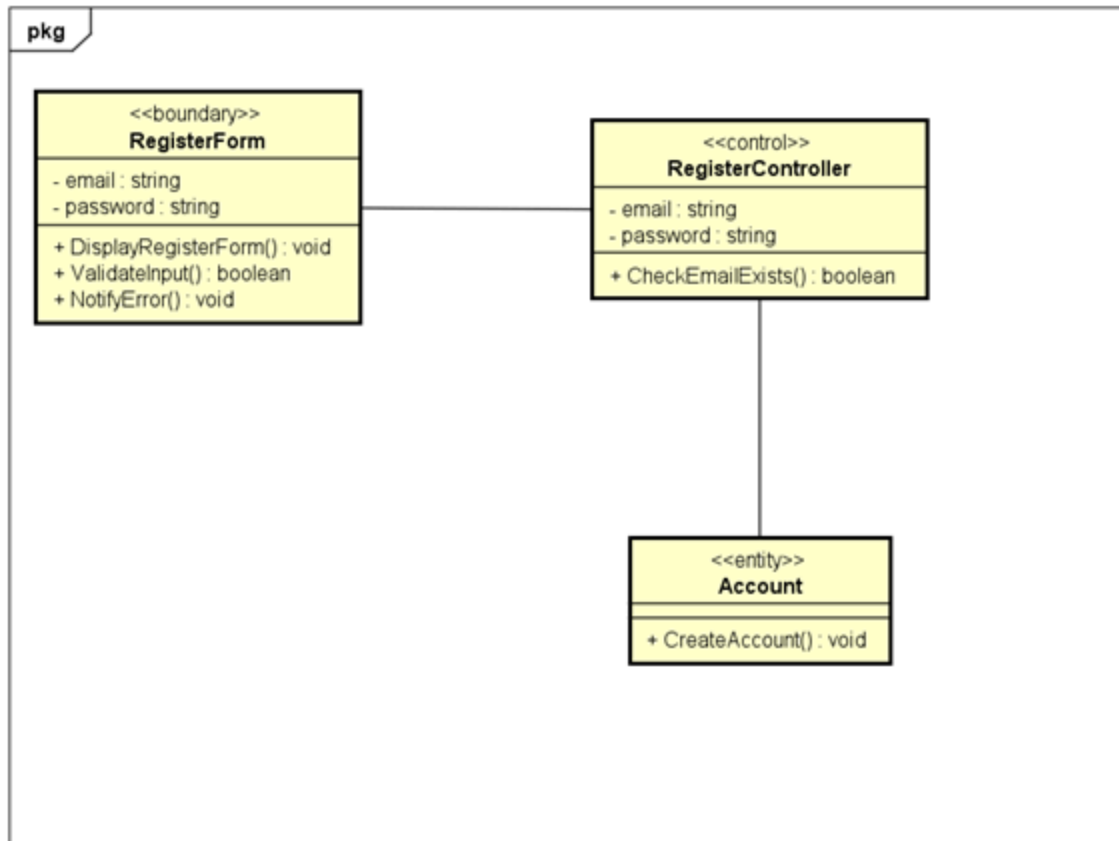
Biểu đồ trình tự



Biểu đồ giao tiếp



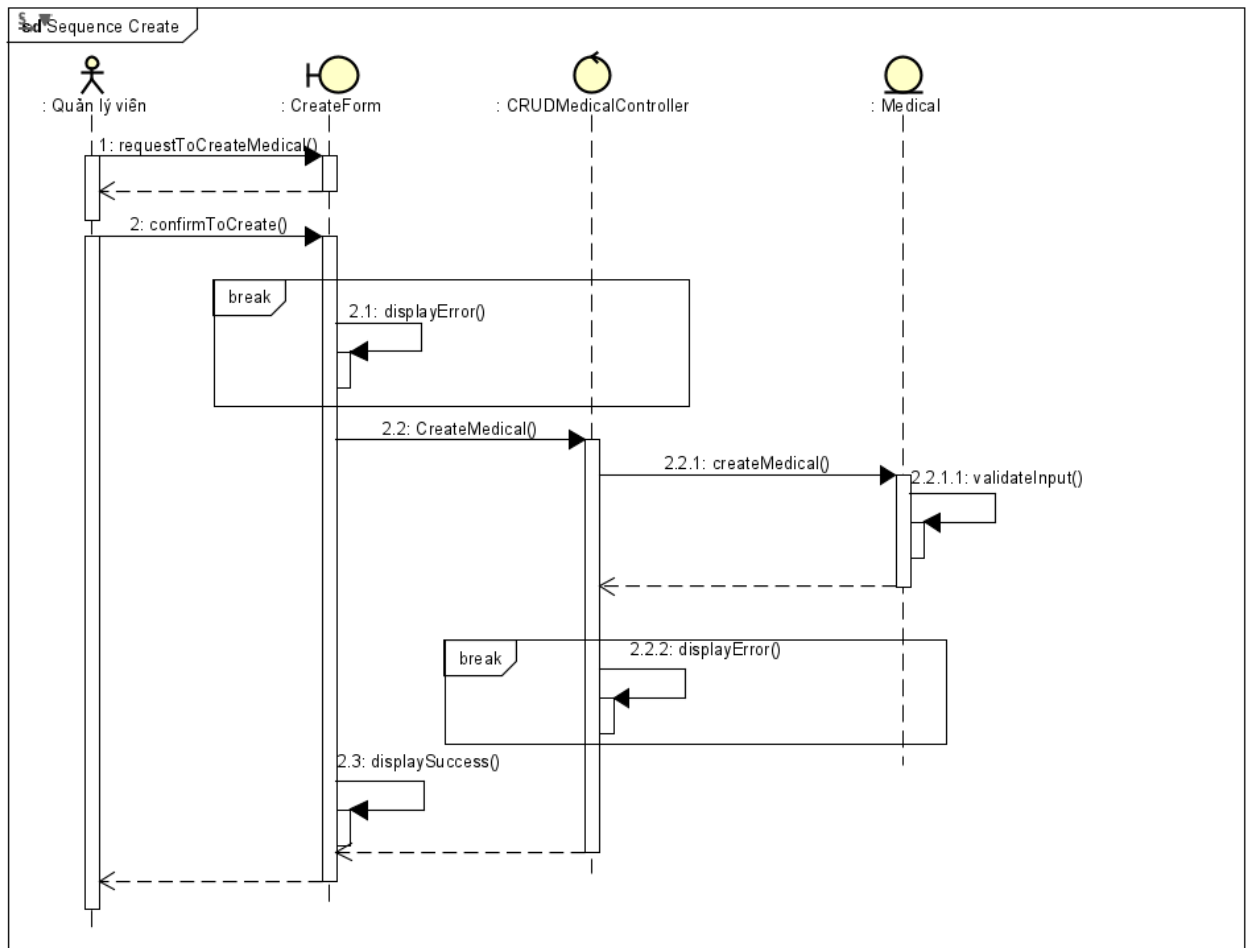
Biểu đồ lớp phân tích

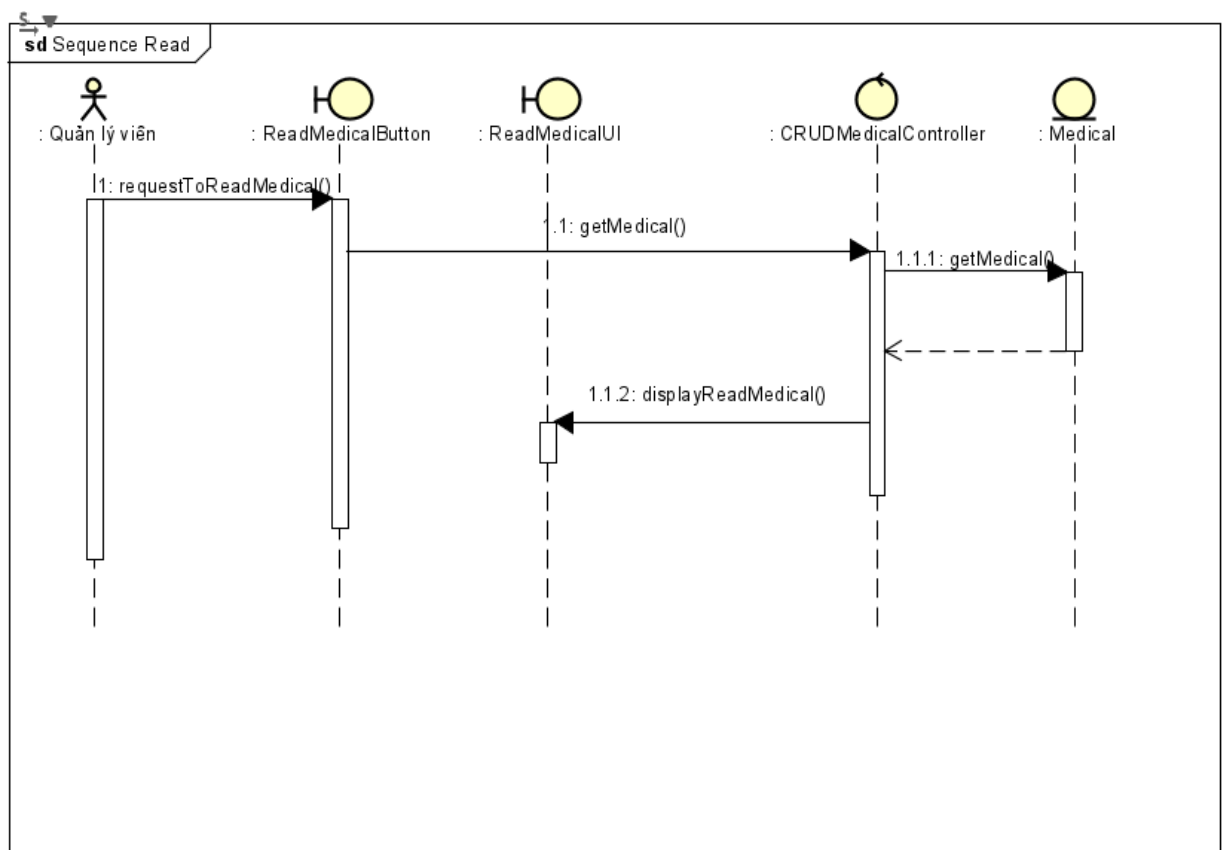
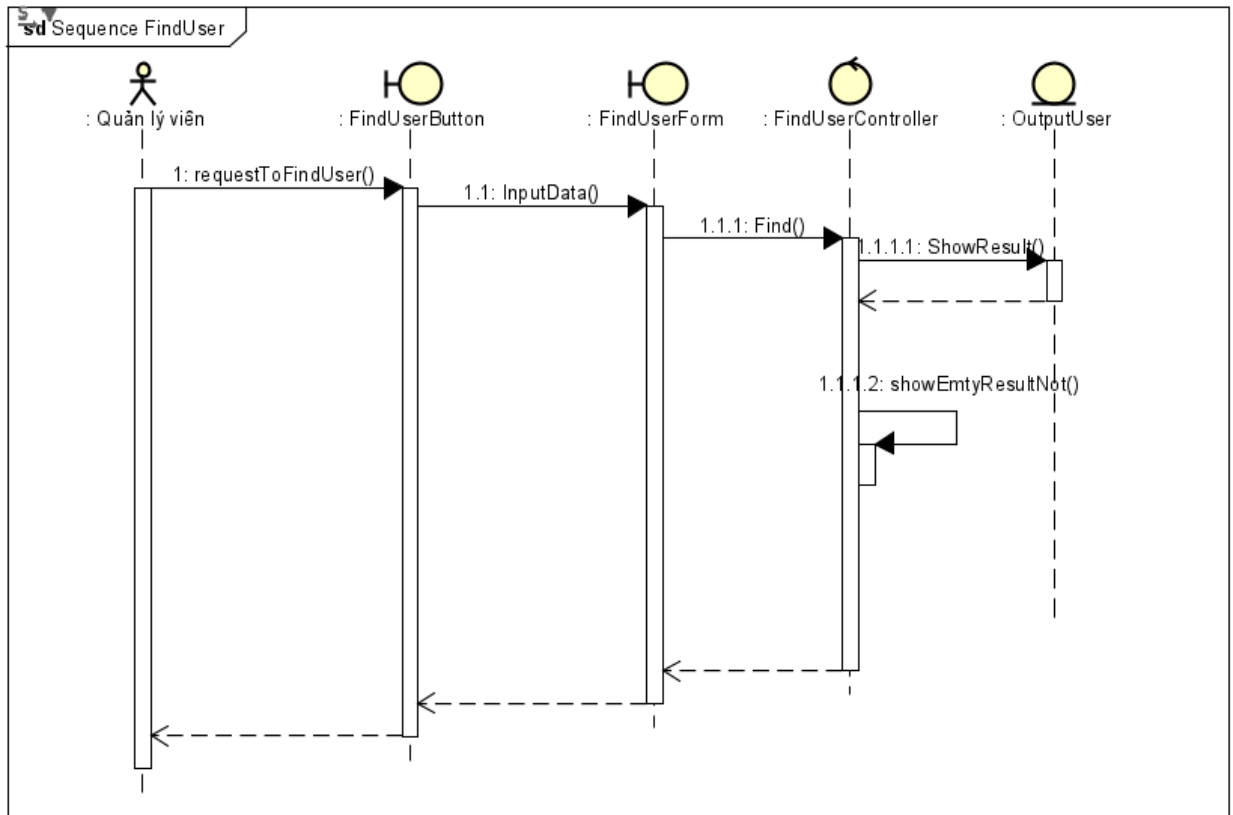


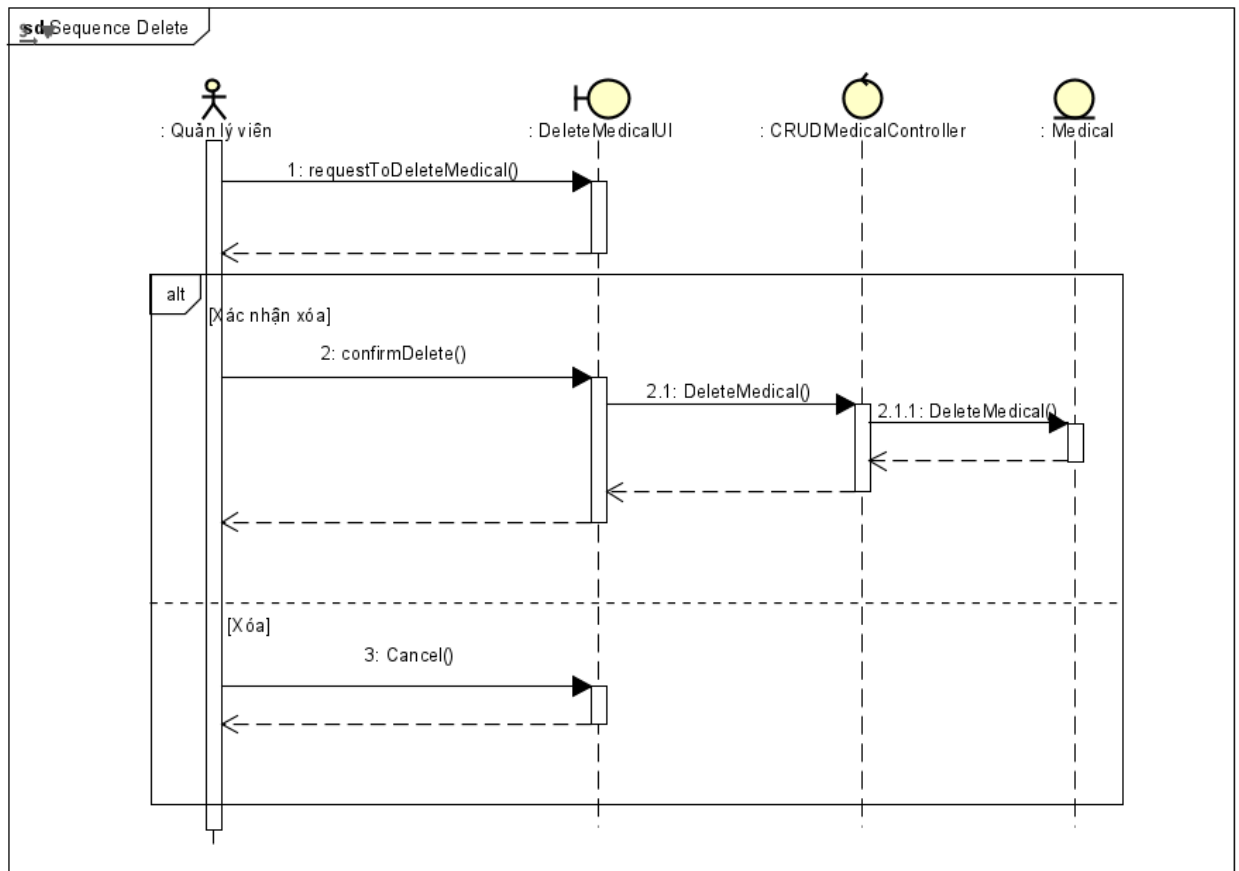
4.3. Thiết kế kiến trúc cho UC “quản lý tài khoản người dùng”

Tương tự cho các UC dạng CRUD (create, read, update, delete khác) như CRUD thú cưng, thuốc, dịch vụ, chuồng, lịch khám

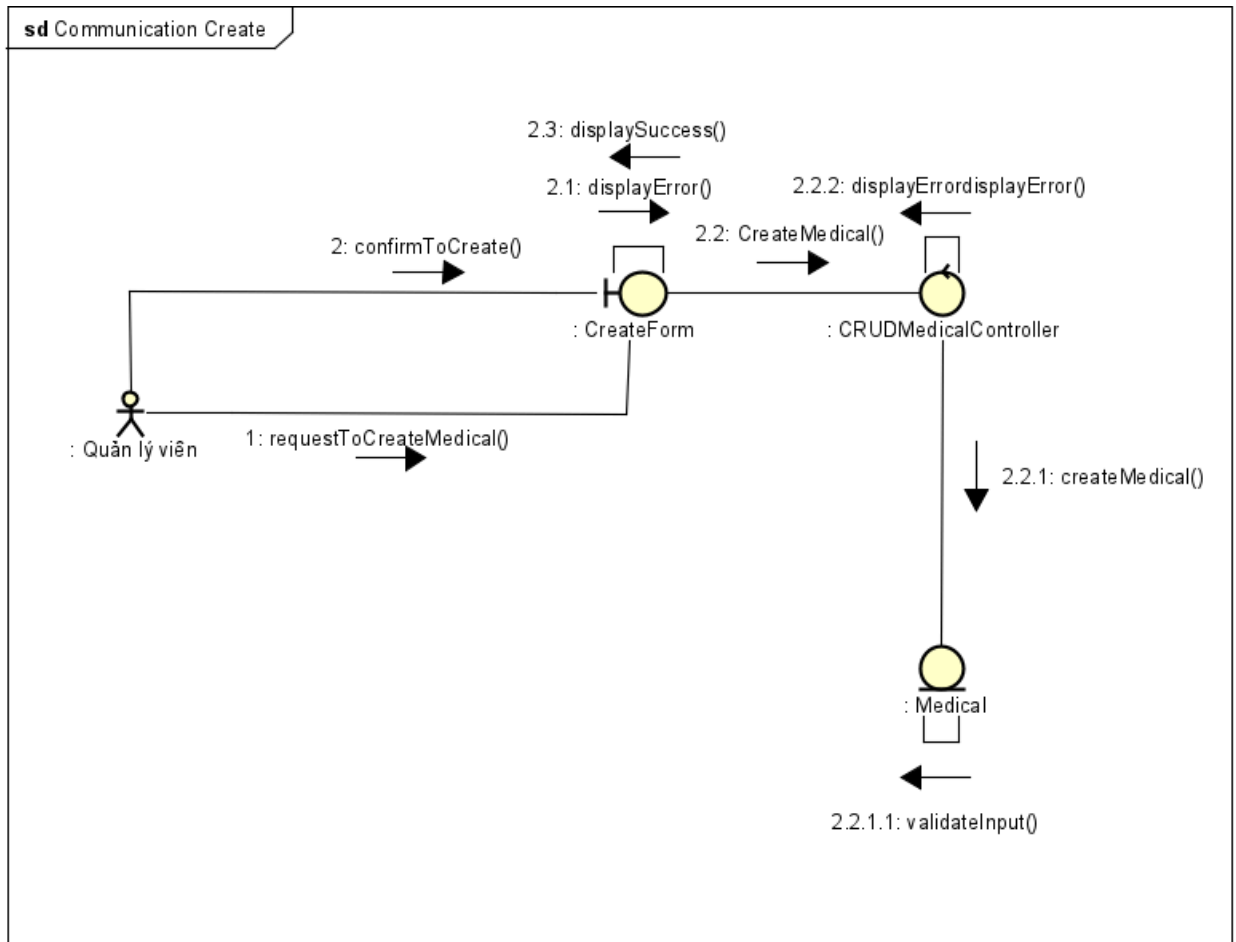
Sequence Diagram



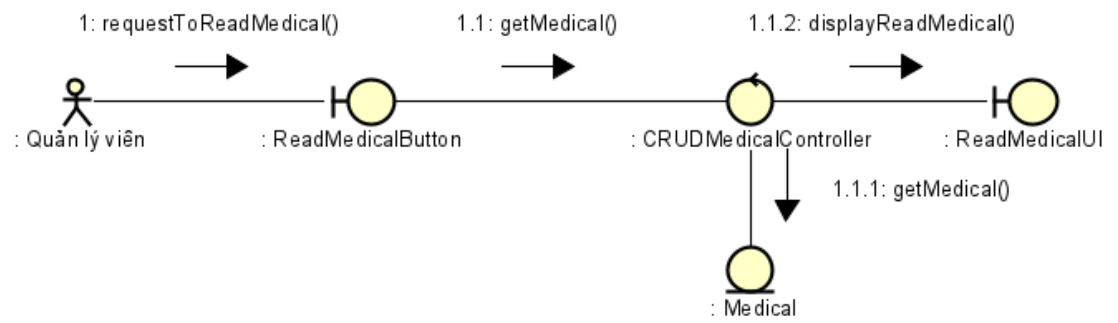


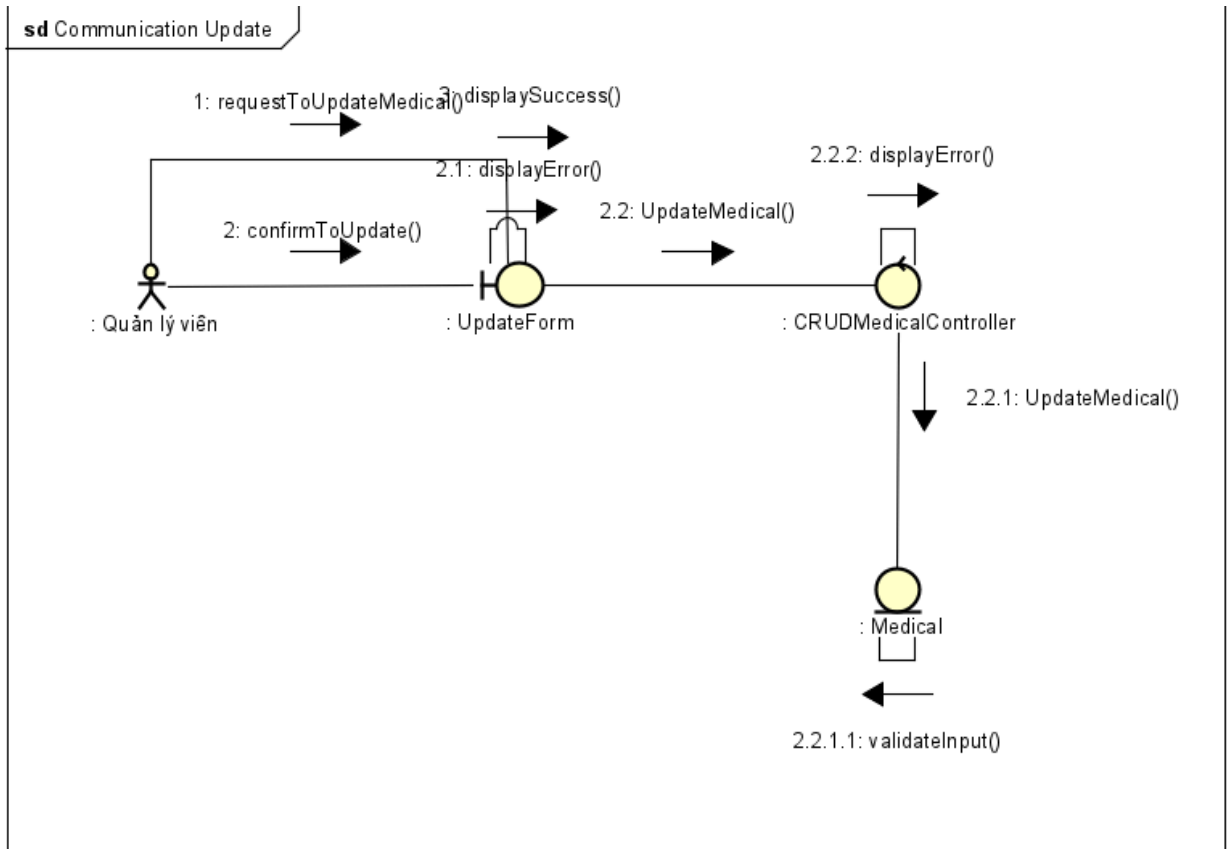


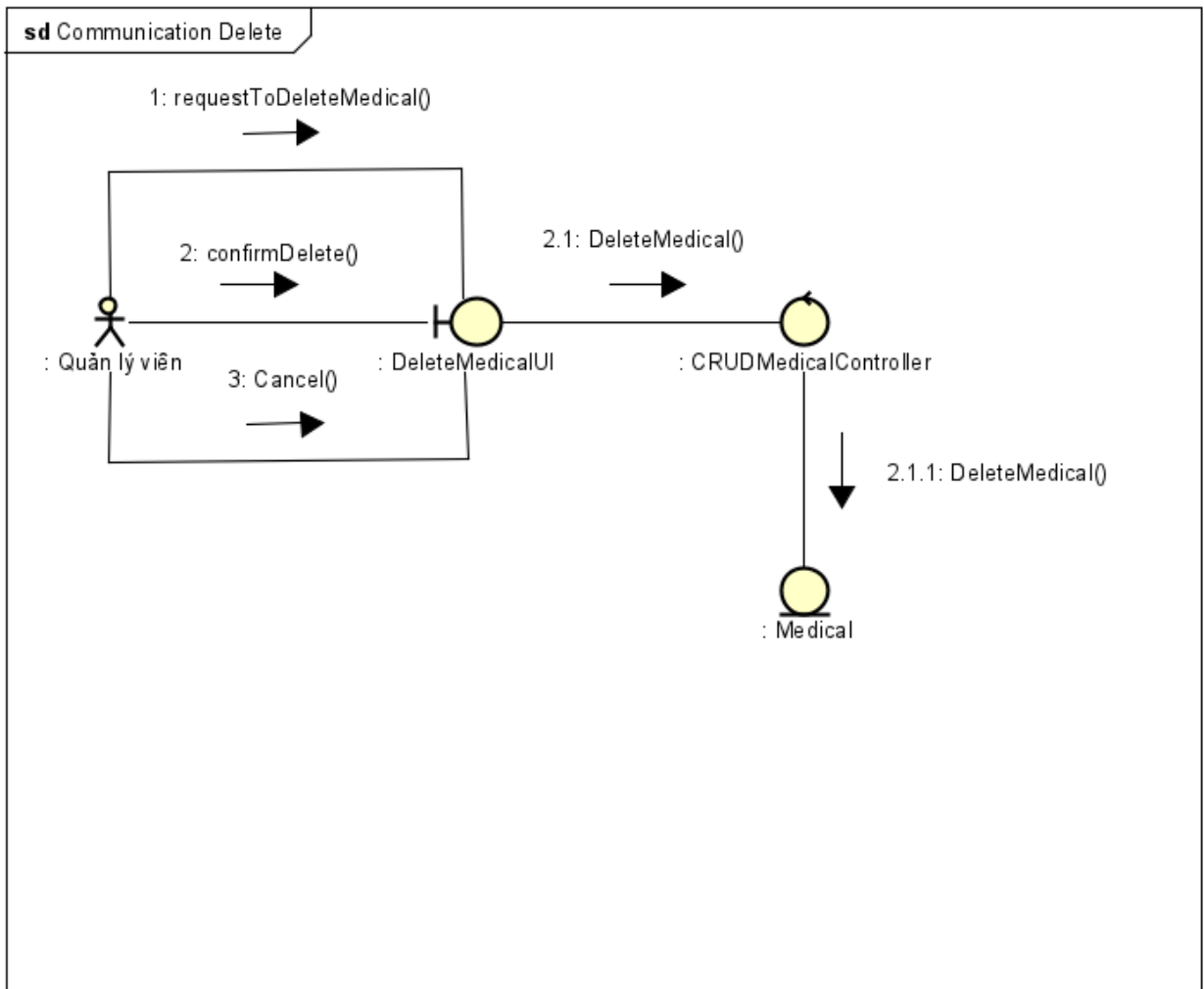
Communication Diagram



sd Communication Read

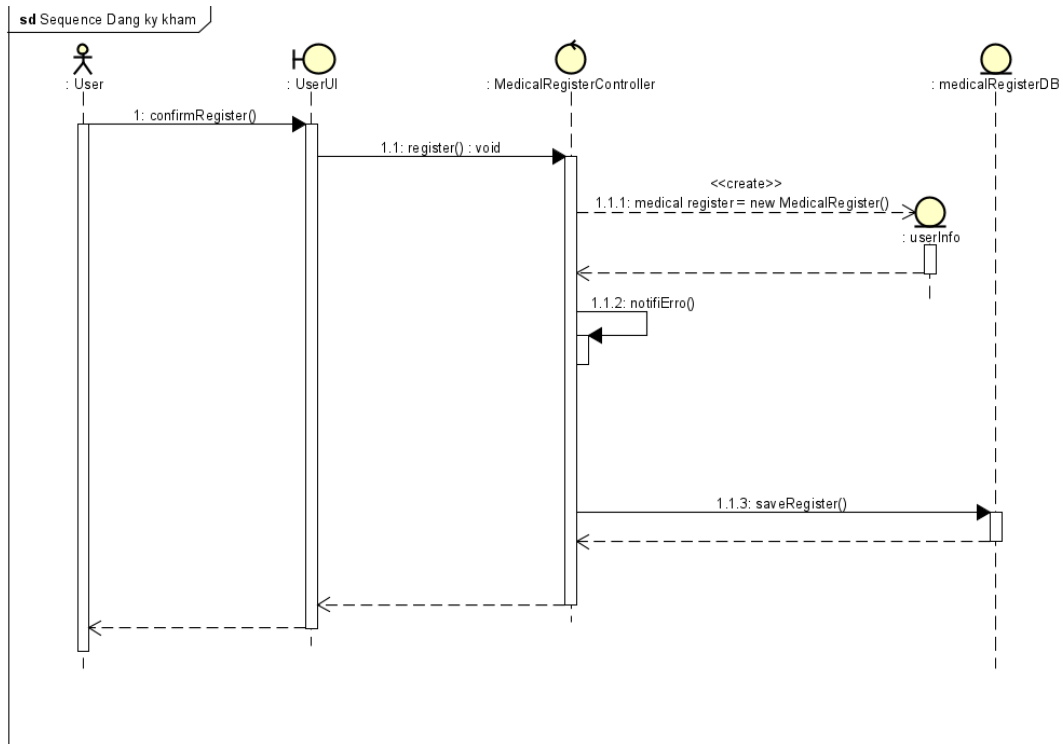




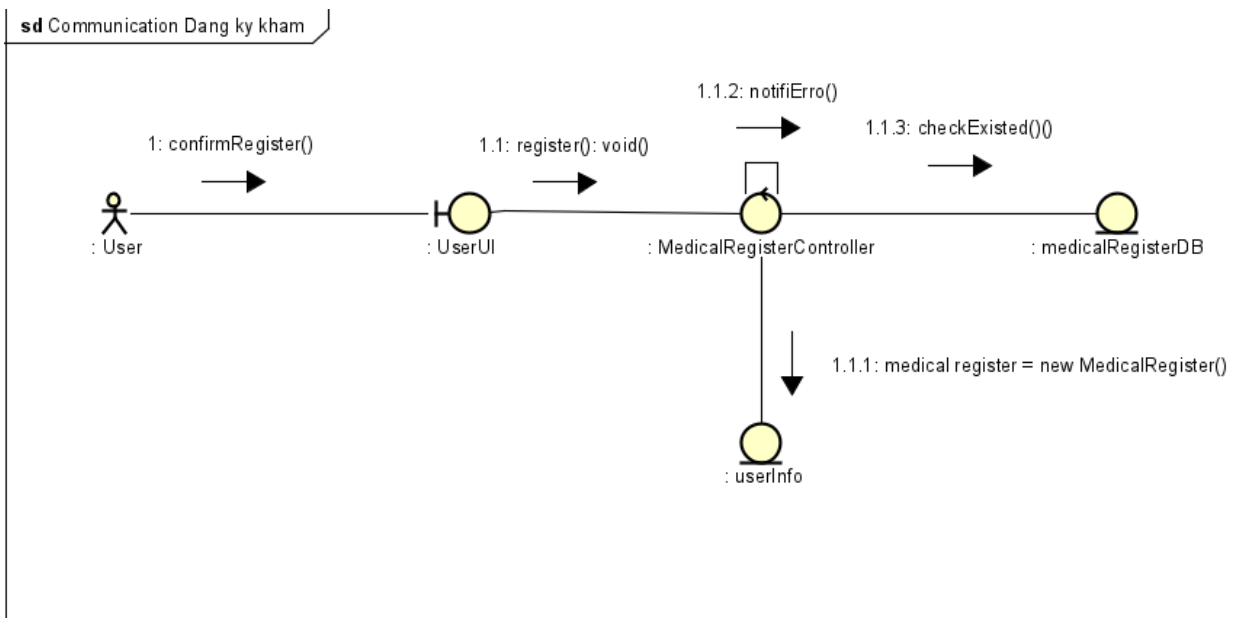


4.4. Thiết kế kiến trúc cho UC “đăng ký sử dụng dịch vụ vệ sinh/làm đẹp” (tương tự “đăng ký khám”)

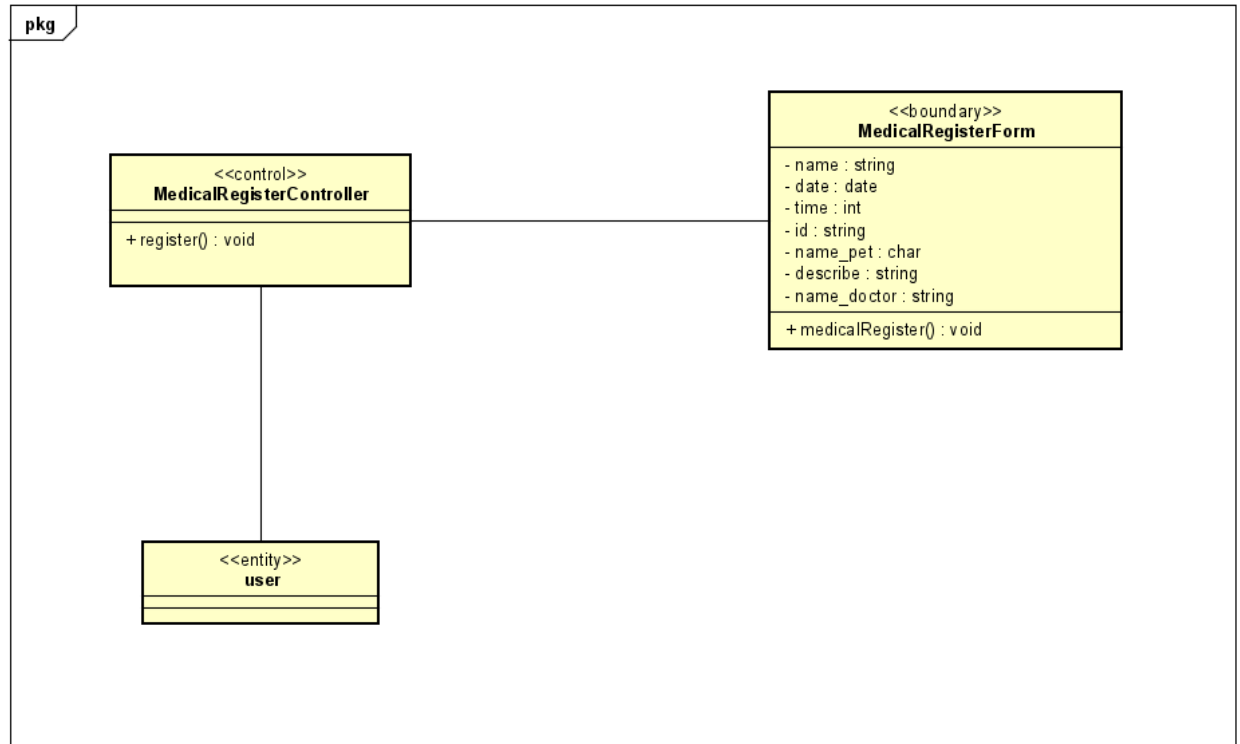
4.5. Thiết kế kiến trúc cho UC “Đăng ký khám”



Communication diagram

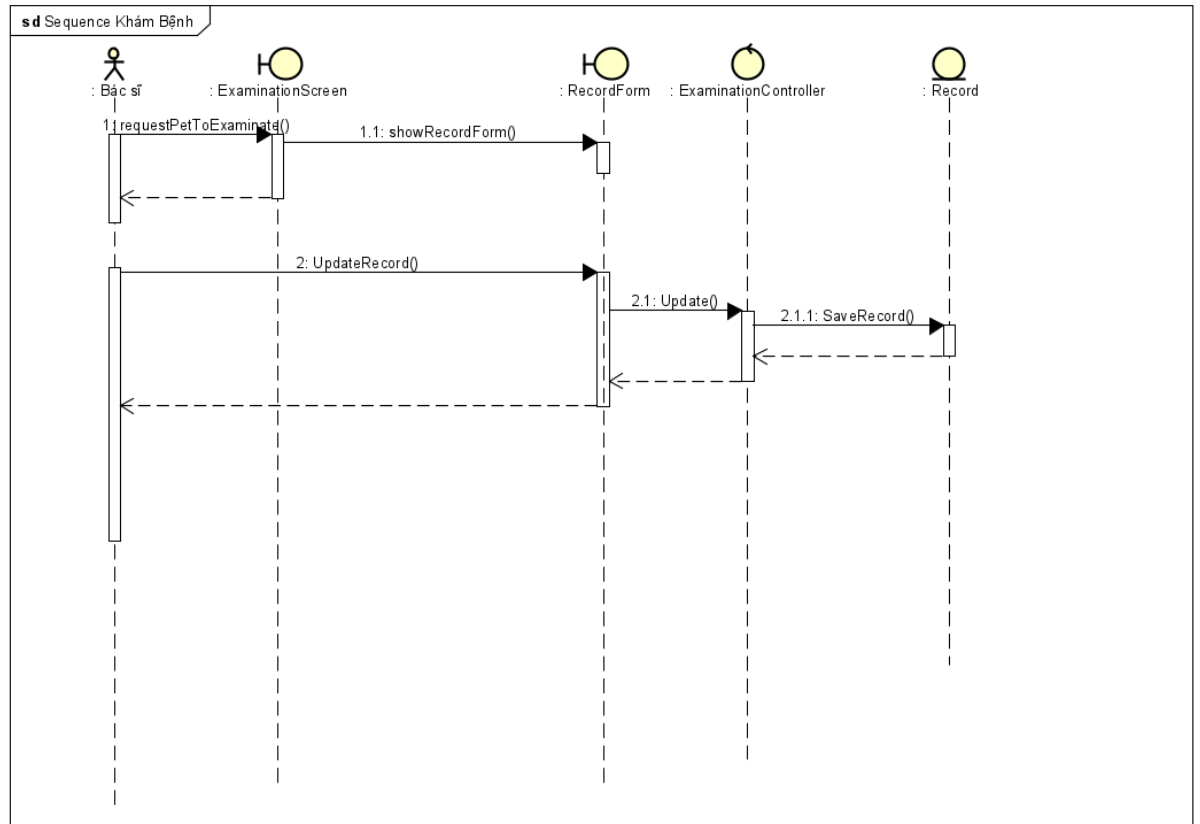


Class Diagram

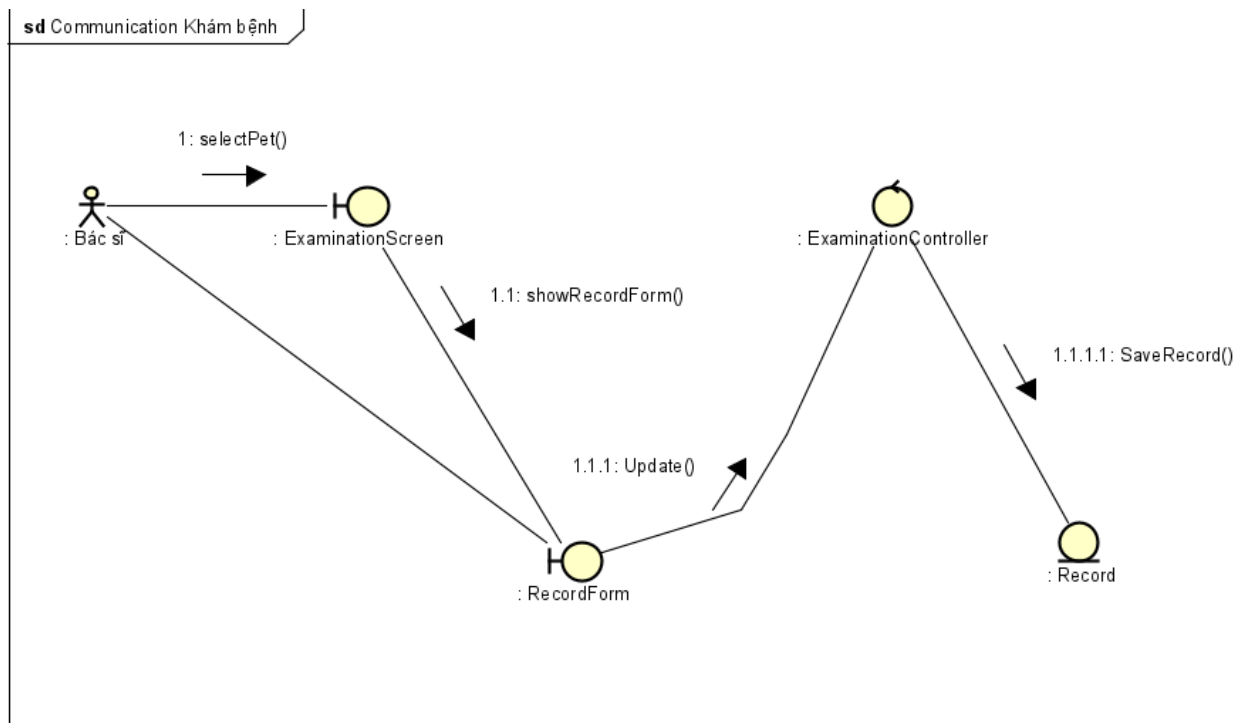


4.6. Thiết kế kiến trúc cho UC “khám bệnh”

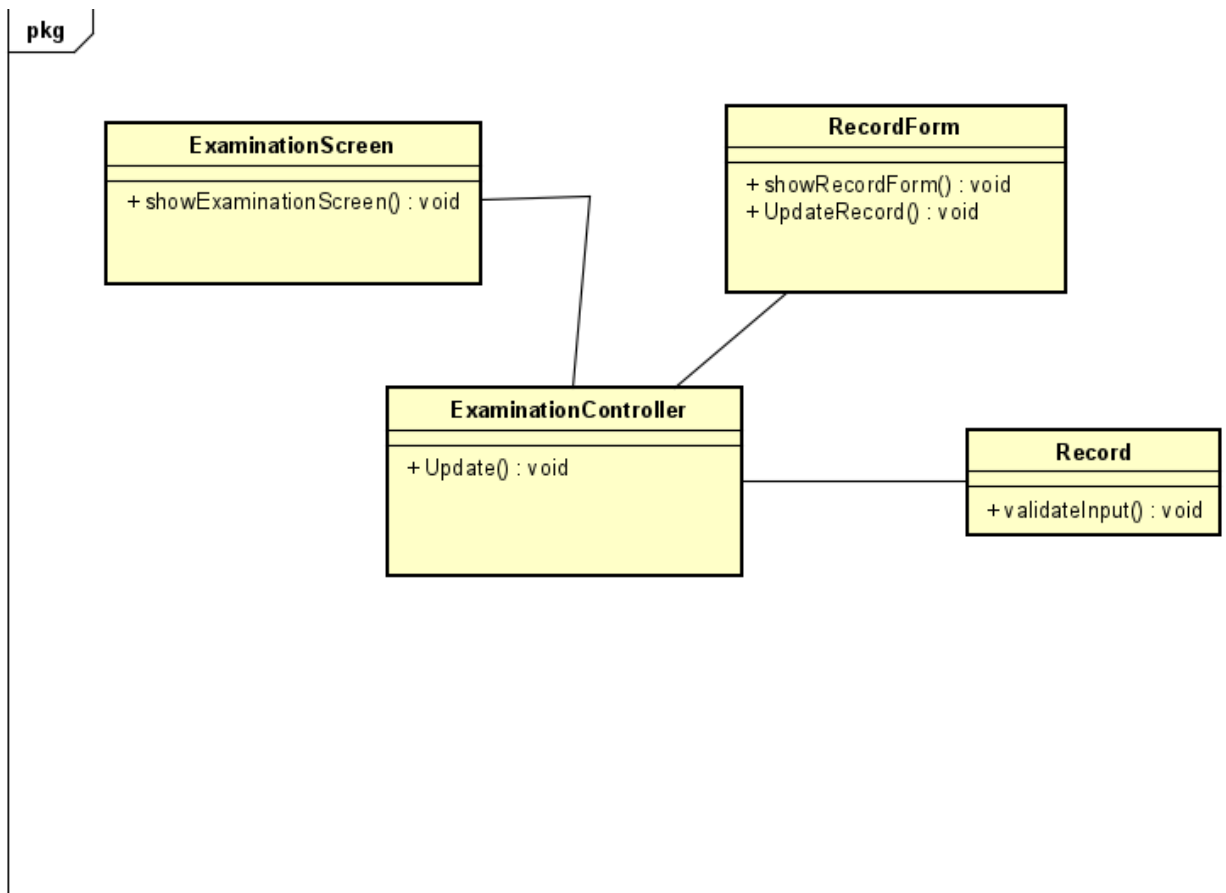
Sequence Diagram



Communication Diagram

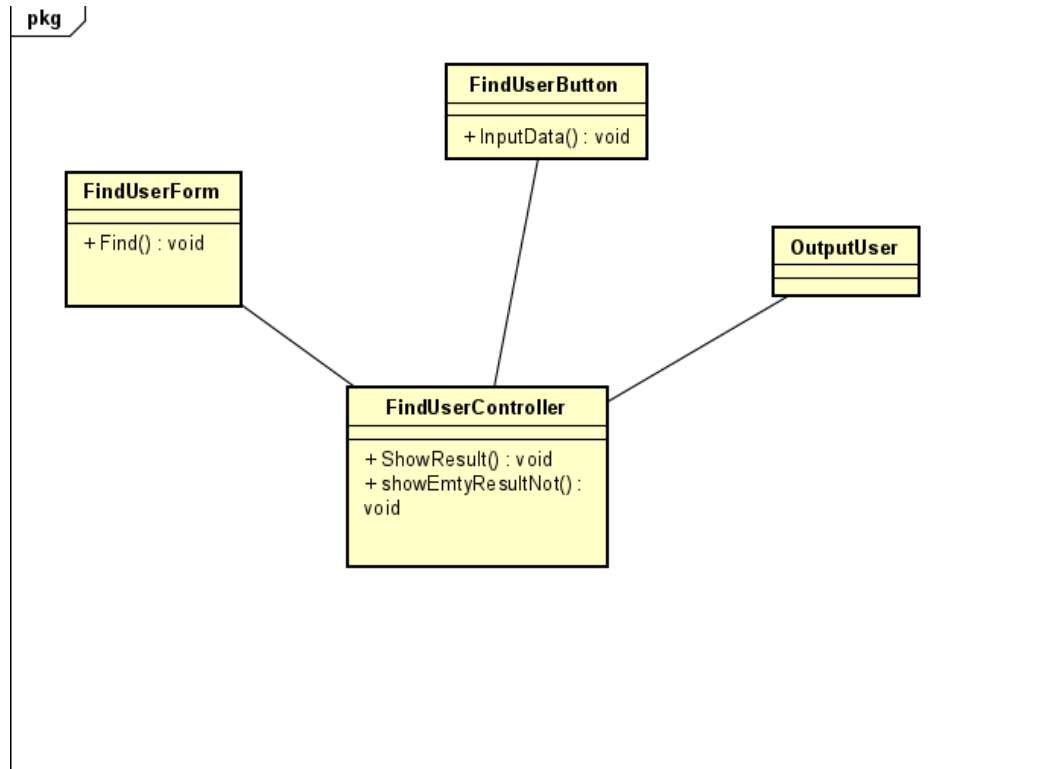


Class Diagram

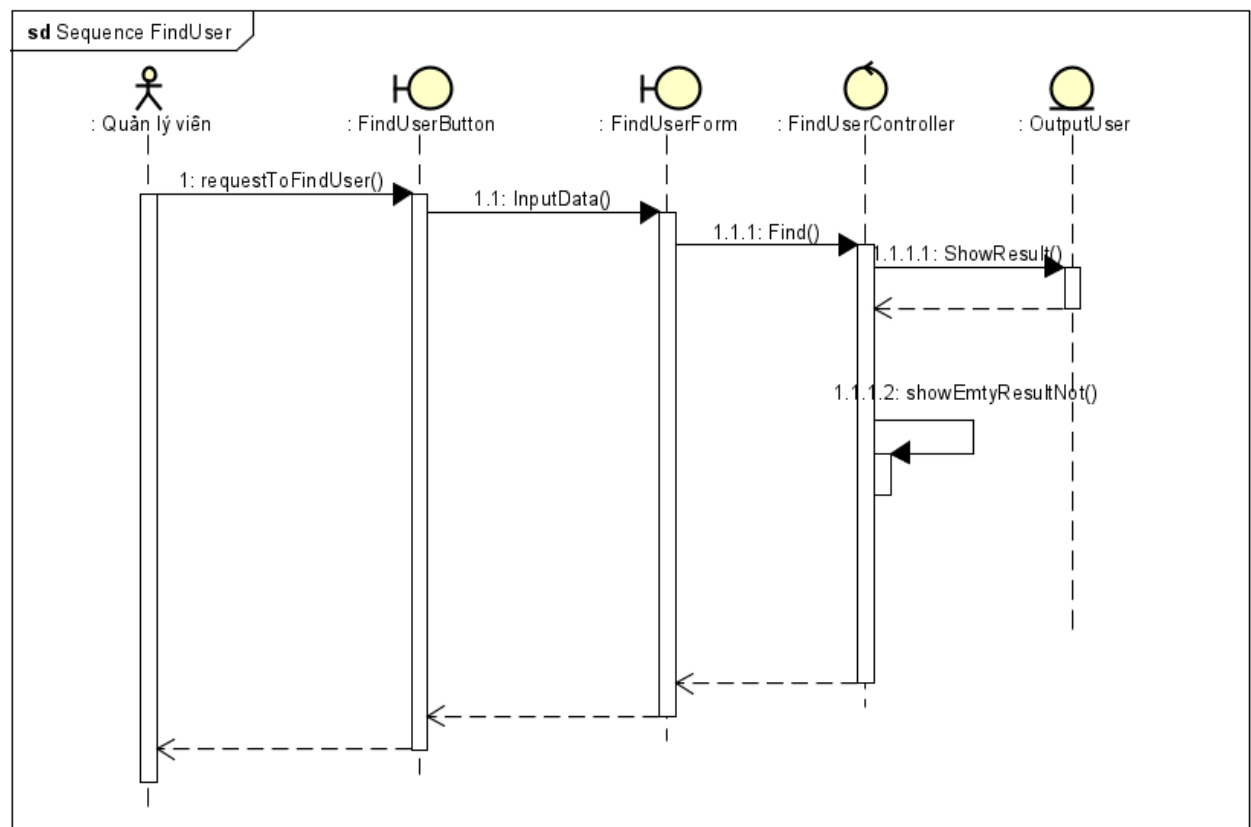


4.7. Thiết kế kiến trúc cho UC “tìm kiếm người dùng”

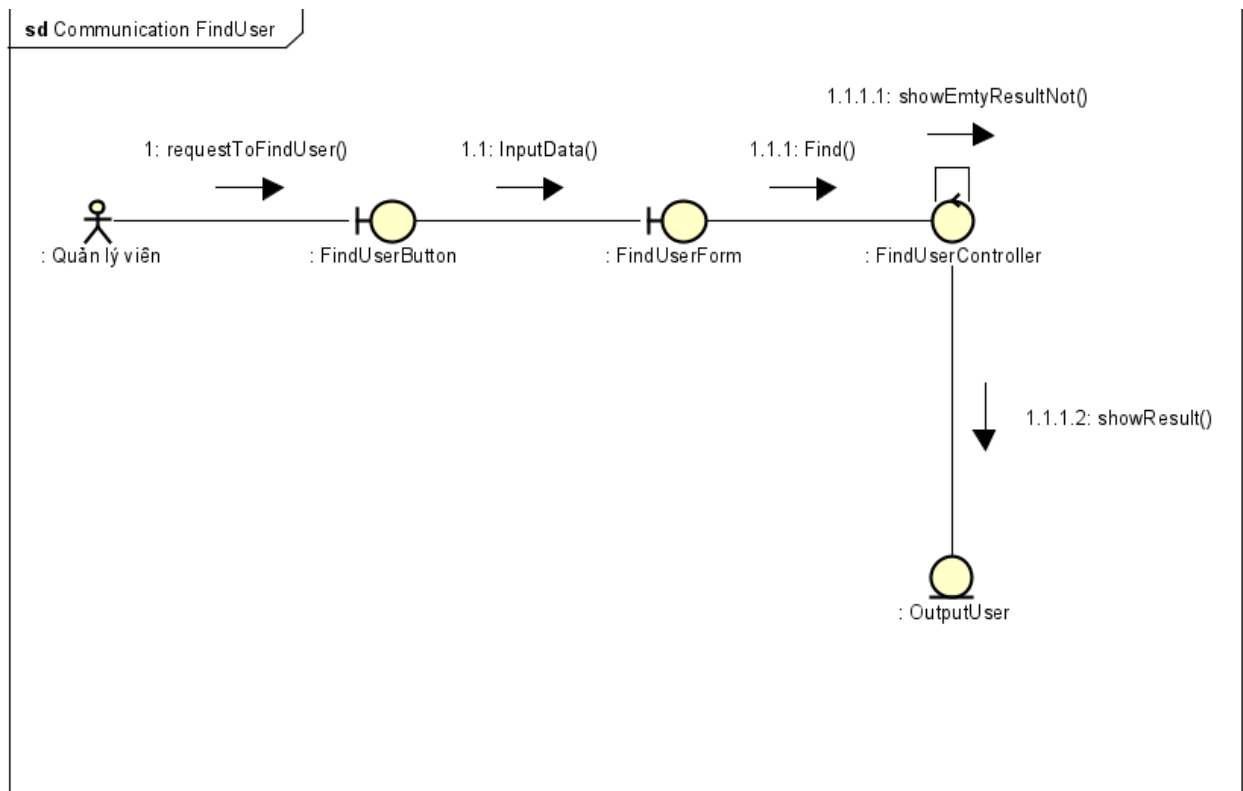
Class Diagram



Sequence Diagram



Communication Diagram



5. Thiết kế chi tiết

5.1. GUI Design

5.1.1. Thiết kế giao diện người dùng (User Interface Design)

Boundary Class được sử dụng để mô hình hình hoá tương tác giữa một hệ thống và môi trường xung quanh. Do đó, chúng có thể được sử dụng để nắm bắt các yêu cầu trên giao diện người dùng. Sự tương tác giữa con người và hệ thống có thể thông qua những loại User Interface (UI) khác nhau như Batch Interface, Command-line Interface (CLI) và Graphical User Interface (GUI).

Trong phần này, chúng ta sẽ sử dụng GUI để minh hoạ thiết kế UI từng bước.

5.1.2. Chuẩn hóa cấu hình màn hình

Display

Số lượng màu được hỗ trợ: 16,777,216 màu

Độ phân giải: 1366 x 768 pixels

Screen

Vị trí của cửa button: Ở dưới cùng (theo chiều dọc) và ở giữa (theo chiều ngang) của khung.

Vị trí của message: Ở giữa trung tâm khung màn hình

Vị trí của screen title: Title đặt ở góc trên bên trái của màn hình.

Sự nhất quán trong hiển thị chữ số: dấu phẩy để phân cách hàng nghìn và chuỗi chỉ bao gồm các ký tự, chữ số, dấu phẩy, dấu chấm, dấu cách, dấu gạch dưới và ký hiệu gạch nối.

Control

Kích thước text: medium size (24px). Font: Segoe UI. Color: #000000

Xử lý check input: Nên kiểm tra xem input có empty hay không.

Tiếp theo, kiểm tra xem input có đúng format hay không.

Dịch chuyển màn hình: Không có các khung chồng lên nhau. Các màn hình được tách biệt. Tuy nhiên, hướng dẫn sử dụng được xem như là 1 popup message vì màn hình chính ở dưới sẽ không thể thao tác trong khi màn hình hướng dẫn sử dụng đang được hiển thị.

Ban đầu khi app khởi chạy thì màn hình splash screen (màn hình chớp) sẽ được hiện lên và sau đó màn hình đầu tiên(Home Screen) sẽ xuất hiện

Nhập input từ bàn phím

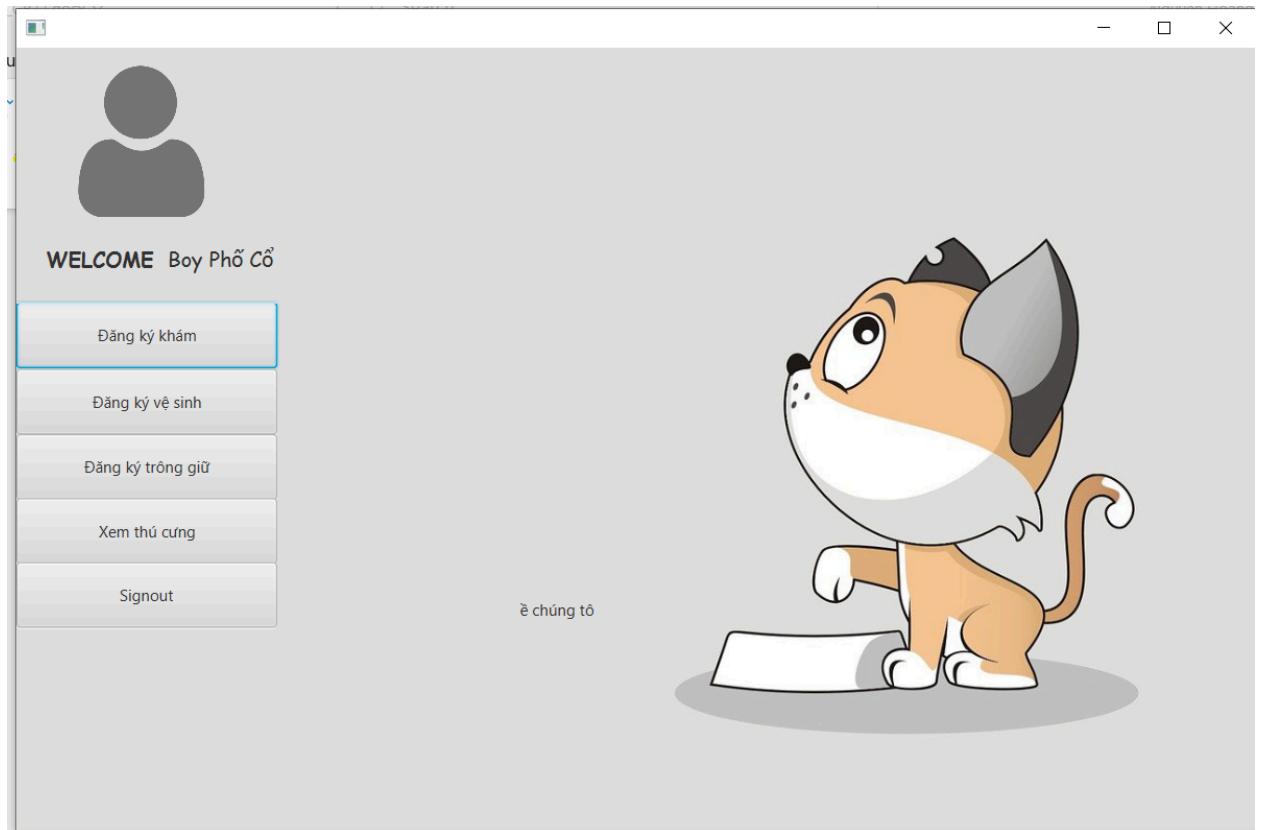
Sẽ không có phím tắt. Có các button quay lại để quay lại các màn hình trước đó. Ngoài ra button “X” nằm ở thanh tiêu đề bên phải để đóng screen

Error

Một thông điệp sẽ được hiện lên để thông báo cho người dùng biết vấn đề đang gặp phải là gì.

5.1.3.Tạo các ảnh màn hình

5.1.3.1. Màn của user



Tên thú cưng


Ngày khám

Thời gian

Lưu

Thoát

Đăng ký khám



Dịch vụ vệ sinh

Tên thú cưng

Dịch vụ


☐ Tia lông ☐ Tắm

☐ Làm móng

Ngày

Thời gian

Lưu Thoát



Dịch vụ Trông Giữ

Tên thú cưng

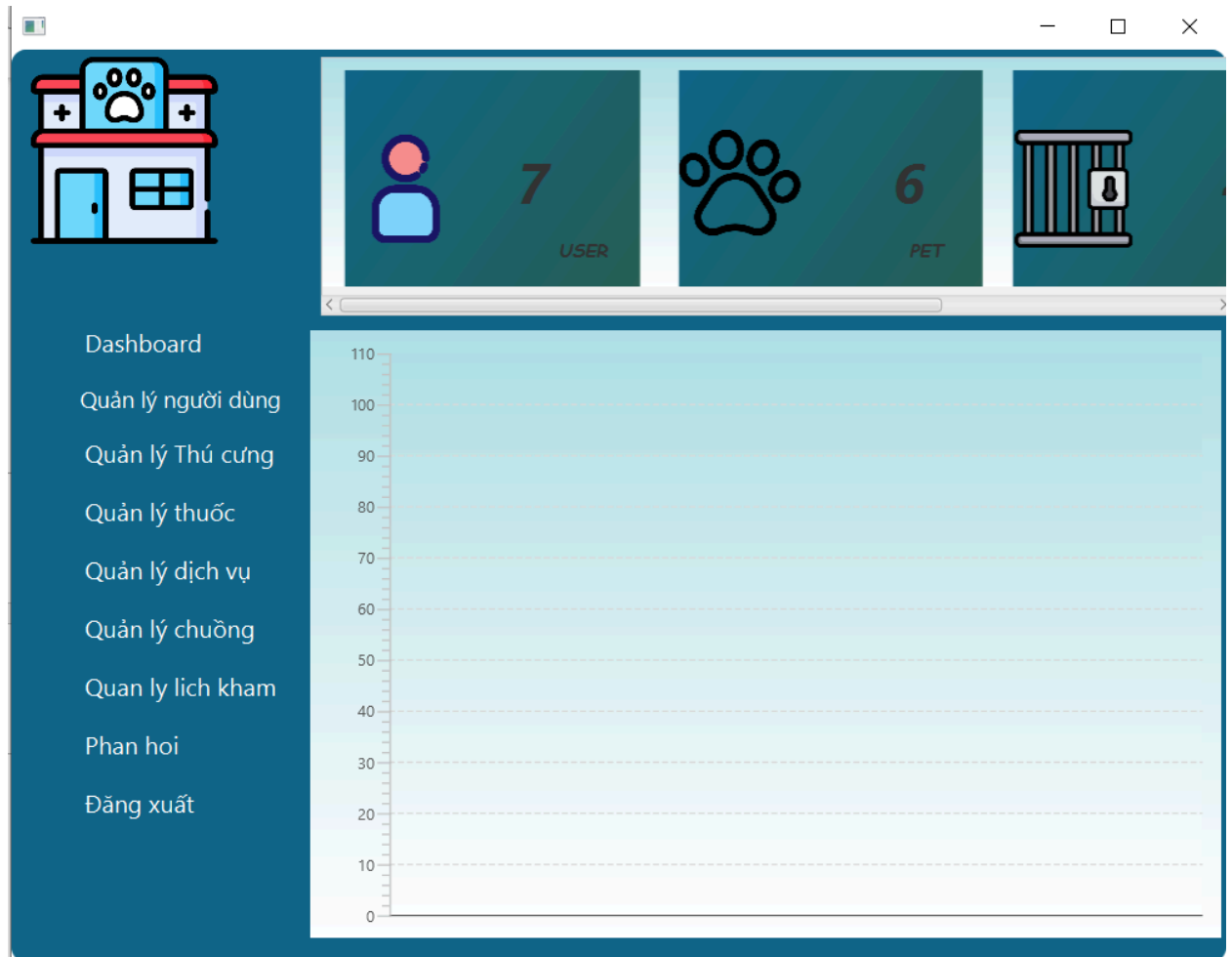
Ngày

Thời gian

Lưu

Thoát





Quản lý người dùng

back

trạng thái : đang sử dụng dịch vụ hay không

add

Xóa

username	tên	email	giới tính	role	birthday
a	Boy P...	a123456@gmail.com	male	customer	2002-01-06
diffusion	Sinh A...	huggingface@sis.edu	female	customer	2022-03-15
tien123	Tien	123@123	male	customer	2020-07-22
q	Dr. Q	q@q	Nữ	doctor	2022-07-11

Dịch vụ chăm sóc				
				back
				Sua Xóa
id pet	dịch vụ	ngày	thời gian	Gia tiền (vnd)
3	Tia lông	2023-07-15	11:00	0
7	Làm móng	2023-07-27	14:00	0

Quản lý chuồng

Back

Chi tiết

Thêm

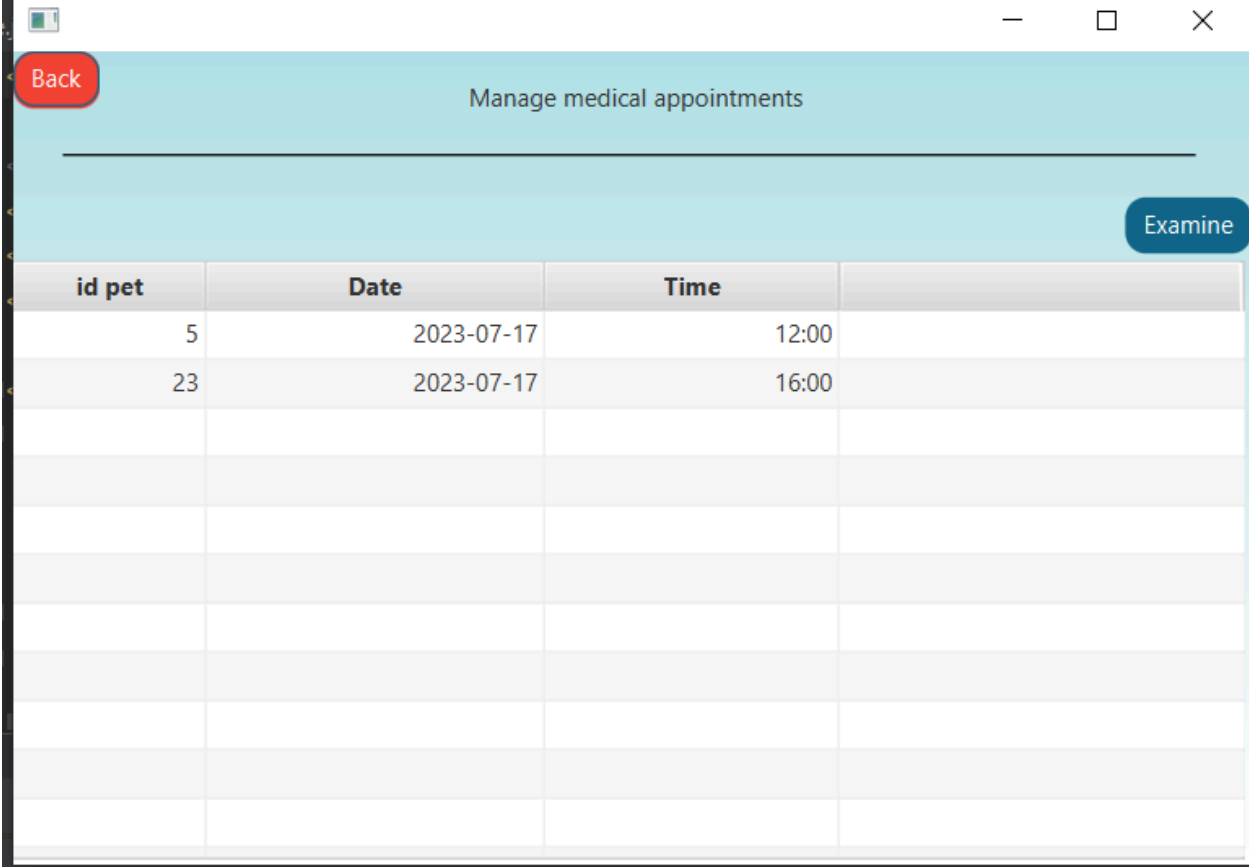
Sửa

Xóa

ID chuồng	Trạng thái	Pet_id
1	đã đặt chỗ	3
2	đã đặt chỗ	1
3	đã đặt chỗ	7
4	trống	null

Quản lý lịch khám			
			back
			Sửa Xóa
id pet	Ngày	Thời gian	
7	2023-07-21	11:00	
1	2023-07-10	11:00	
3	2023-07-20	16:00	
2	2023-07-15	17:00	

5.1.3.3. Màn của bác sĩ



Back

Examine Pet

Pet Name:

Category:

Age:

Symptom:

Diagnostic:

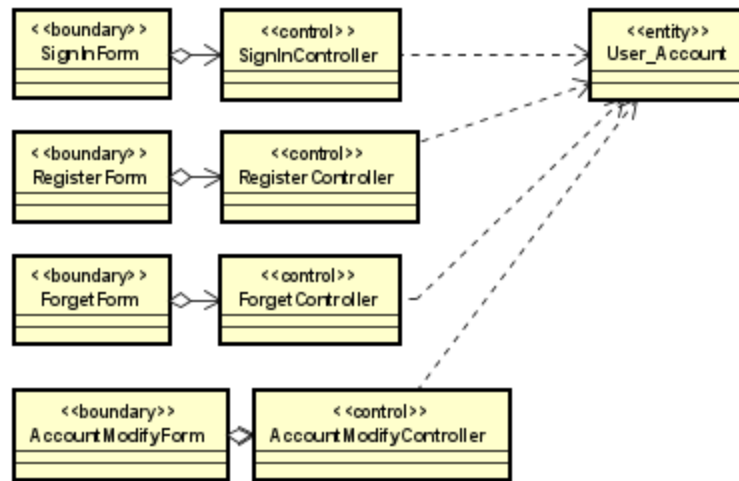
Advice:

Prescribe Medicine:

Prescribed Medicine:

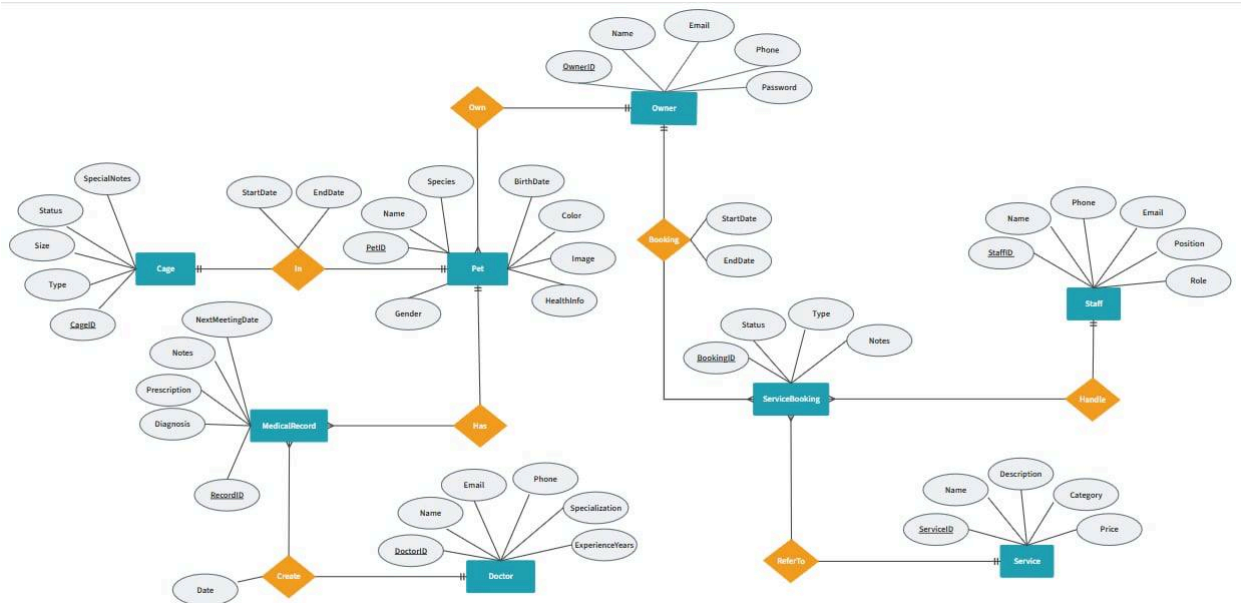
5.2. Class Diagram

5.2.1. Class Diagram cho đăng ký, đăng nhập, quản lý tài khoản người dùng



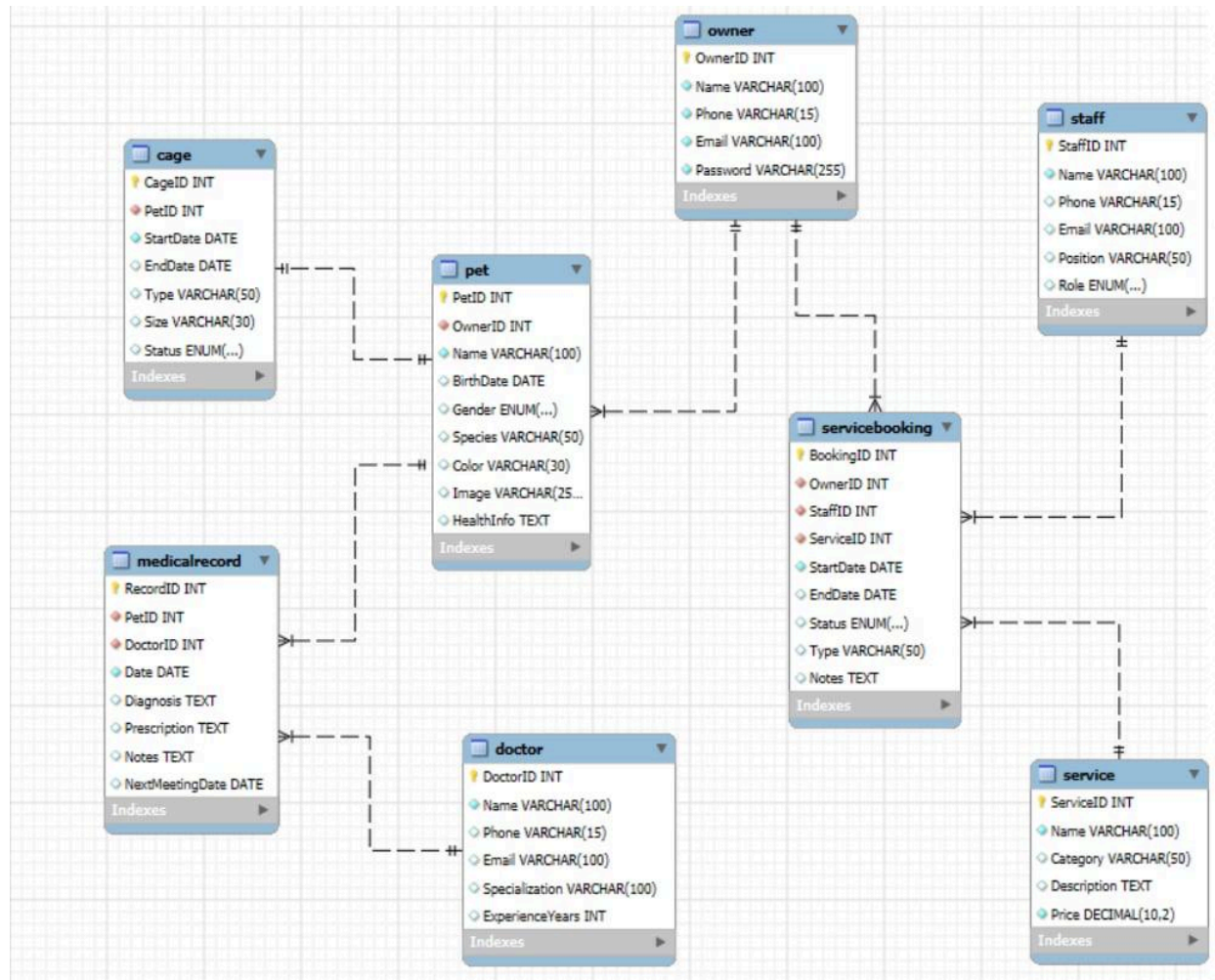
5.3. Data Modeling

5.3.1. Mô hình khái niệm (Conceptual Data Model)



5.3.2. Thiết kế cơ sở dữ liệu (Logical Data Model)

5.3.2.1. Mô hình dữ liệu logic



5.3.2.2. Mô hình dữ liệu vật lý

Trong phần này, chúng ta sẽ thiết kế chi tiết cho từng phần tử trong biểu đồ trên. Ví dụ, trong biểu đồ cơ sở dữ liệu quan hệ (RDBMS), ta thiết kế chi tiết cho từng bảng và ràng buộc.

Chú thích:

PK: Primary Key

FK: Foreign Key

6. Test case

7. Nguyên tắc thiết kế

7.1. Áp dụng Design Concepts

7.1.1.Coupling

Để có một bản thiết kế tốt thì cần phải thiết kế sao cho coupling lỏng để khi có sự thay đổi ở một module thì ảnh hưởng ít nhất có thể đến các module khác. Coupling càng lỏng lẻo, càng tốt. Các code trong dự án đều hướng tới stamp coupling và data coupling.

- **Content coupling**

Đây là vi phạm coupling ở mức cao nhất được định nghĩa là khi mà một component có thể truy cập trực tiếp vào hoạt động bên trong của một component khác ví dụ như trực tiếp truy xuất data, thay đổi data của component

Cách xử lý: tất các trường dữ liệu trong 1 lớp đều được set là private, các đối tượng muốn truy xuất dữ liệu phải thông qua getter và setter method

- **Common Coupling**

Đây là trường hợp 2 module cùng chia sẻ chung dữ liệu, 2 global structure có thể cùng vào chỉnh sửa, truy cập hoặc có chung những khối mã nguồn thì sẽ vi phạm common coupling.

Tuy nhiên, object oriented programming không có common data. Tất cả data đều thuộc về lớp. Ví dụ khi dùng C, bạn có data structure khai báo là global, sau đó sẽ có một số phương thức ghi vào structure, các phương thức khác đọc từ đó, trường hợp này sẽ vi phạm common coupling. Còn trong Object Oriented, không có bất cứ data nào mà không thuộc về lớp. Do đó, project hiện tại (JAVA) không vi phạm common coupling.

- **Control Coupling**

Một module vi phạm control coupling khi nó truyền tham số điều khiển cho các module khác thông qua việc gọi method. Điều này không tốt vì thành phần được gọi sẽ biết được cấu trúc bên trong thành phần gọi và khi cấu trúc này bị thay đổi thì thành phần được gọi sẽ phải thay đổi theo.

- **Stamp Coupling**

Hai lớp được coi là vi phạm stamp coupling nếu một lớp gửi một collection hoặc một object dưới dạng tham số và chỉ một vài phần data

được sử dụng ở lớp thứ hai. Để giảm thiểu level coupling này, chúng ta truyền đủ những tham số cần thiết trong function gọi. Chú ý rằng, stamp coupling là mức có thể chấp nhận được.

Do vấn đề thời gian nên bọn em không thể tạo ra hàng loạt các respond để chia công việc cho các phần này, cho nên hầu hết các modules đều vi phạm Stamp coupling

Giải pháp: tạo ra các respond mà chỉ nhận các thông tin cần thiết cho các modules và trả về các thông tin khác.

- **Data Coupling**

Nếu các module của bạn ở mức data coupling, thiết kế của bạn là thiết kế tốt. Đây là level mà chúng ta hướng đến.

7.1.2.Cohesion

Nhìn chung, để tăng mức độ (level) cohesion, chúng ta có thể thử đặt mỗi phần vào một module khác phù hợp hơn (tạo module mới nếu cần)

- **Coincidental cohesion**
 - o Các subcomponent đặt trong 1 component vì tính ngẫu nhiên
 - o Rõ ràng, chúng ta có thể thấy loại cohesion này trong class Config hoặc class Utils trong package utils
 - o Ngoài những trường hợp trên thì Design không vi phạm cohesion trên.
- **Logical cohesion**
 - o Các thành phần trong 1 module có liên quan đến nhau nhưng về mặt logic chứ không phải chức năng.
 - o Quan sát lại phần Control Coupling bên trên, nếu chúng ta chia đoạn code thành 2 methods và đưa chúng vào trong cùng một lớp, chúng ta có thể đổi mặt với logical cohesion. Do đó, chúng ta cần xem xét đưa chúng vào trong những class hoặc package khác.
 - o Do sử dụng mô hình MVC nên chúng em đã tránh dc cohesion này.
- **Informational cohesion**

- o Các operation có tính độc lập (có input và output riêng nhưng chúng có thể thao tác trên một tập dữ liệu chung là attribute của lớp đó)
- o Nhóm đạt được mức cohesion này.

7.2. Áp dụng Design Principles SOLID

Nguyên tắc SOLID là viết tắt của 5 chữ cái đầu trong 5 nguyên tắc thiết kế hướng đối tượng, giúp cho lập trình viên viết ra những đoạn code dễ đọc, dễ hiểu, dễ maintain. 5 nguyên tắc đó bao gồm: Single responsibility principle, Open-Closed principle, Liskov substitution principle, Interface segregation principle, và Dependency inversion principle.

Áp dụng SOLID vào thiết kế của nhóm :

- Nguyên tắc số 1: Single-responsibility principle (SRP):

Mỗi lớp chỉ nên chịu trách nhiệm về một nhiệm vụ cụ thể.

Theo nguyên lý này, một class chỉ nên giữ một trách nhiệm duy nhất. Một class nếu có quá nhiều chức năng thì sẽ trở nên khó đọc, các chức năng không quá rạch ròi dẫn đến khó khăn trong việc đọc hiểu code.

- Nguyên tắc số 2: Open-closed principle (OCP):

Không được thay đổi một class có sẵn, nhưng có thể mở rộng bằng kế thừa.

Theo nguyên lý này, mỗi khi ta muốn thêm chức năng cho chương trình thì nên viết class mới mở rộng từ class cũ (như bằng cách kế thừa) chứ không nên sửa đổi class cũ. Việc viết class mới mở rộng từ class cũ này dẫn tuy có thể sẽ phát sinh nhiều class, nhưng có điểm lợi là chúng ta sẽ không cần phải test lại các class cũ nữa, mà chỉ tập trung vào test các class mới vừa tạo ra. Thông thường để mở rộng thêm chức năng thì ta cần phải viết thêm code, vậy để thiết kế ra một module có thể dễ dàng mở rộng nhưng lại phải hạn chế sửa đổi code thì ta cần tách những phần

dễ thay đổi ra khỏi phần ít thay đổi sao cho đảm bảo không ảnh hưởng đến phần còn lại.

=> Cải thiện: Giúp cho việc mở rộng code, thêm các entity, các lớp controller, hay service trở lên dễ dàng. Khi có thêm 1 entity mới hay controller, service không phải viết đè lên code cũ, giảm thiểu nguy cơ lỗi và ảnh hưởng tới các code cũ

- Nguyên tắc số 3: Liskov substitution principle (LSP):

Các đối tượng kiểu class con có thể thay thế các đối tượng kiểu class cha mà không gây ra lỗi.

Cần chú ý không nên cho những phương thức không đặc trưng, không mang tính khái quát vào lớp cha để tránh trường hợp lớp con khi kế thừa phải những phương thức đó. Với những phương thức kiểu này thì nên sử dụng interface và cho những class muốn sử dụng phương thức đó implements interface đó.

Trong một chương trình có rất nhiều các class thừa kế từ các class cha là thư viện nhưng không bị biến đổi tính đúng đắn của chương trình.

Nguyên tắc này được nhóm em áp dụng như sau: Tạo ra các abstract class để các lớp con sau này kế thừa. Các lớp con này có thể thay thế lớp cha mà không thay đổi tính đúng đắn của chương trình

=> Cải thiện: Theo LSP, các lớp con có thể thay thế cho lớp cha mà không làm thay đổi tính năng của hệ thống. Giúp giảm sự phụ thuộc và kiểu và tạo ra tính linh hoạt và mở rộng cho hệ thống.

- Nguyên tắc số 4: Interface-segregation principle (ISP):

Thay vì dùng một interface lớn thì nên tách thành nhiều interface nhỏ với từng mục đích cụ thể.

Nếu ta chỉ có duy nhất một interface, ở đó nhét toàn bộ phương thức, thì khi các class muốn implement sẽ phải define lại toàn bộ abstract method trong interface đó. Điều này khiến tốn thời gian và lãng phí tài nguyên không cần thiết. Vậy nên cần chia thành nhiều interface nhỏ hơn, gồm các method liên quan với nhau, thì việc implement sẽ dễ dàng hơn.

- Nguyên tắc số 5: Dependency-inversion principle (DIP):

Các module cấp cao không nên phụ thuộc vào các modules cấp thấp, mà nên phụ thuộc vào abstraction. Chi tiết nên phụ thuộc vào abstraction, abstraction không nên phụ thuộc vào chi tiết.

Những thành phần trong chương trình chỉ nên phụ thuộc vào những cái trừu tượng (abstraction). Những thành phần trừu tượng không nên phụ thuộc vào các thành phần mang tính cụ thể. Những cái trừu tượng (abstraction) là những cái ít thay đổi và biến động, tập hợp những đặc tính chung nhất, bao quát nhất của những cái cụ thể. Những cái cụ thể dù khác nhau thế nào đi nữa đều mang khuôn mẫu mà cái trừu tượng đã định ra. Việc phụ thuộc vào cái trừu tượng sẽ giúp chương trình linh động và thích ứng tốt với các sự thay đổi diễn ra liên tục.

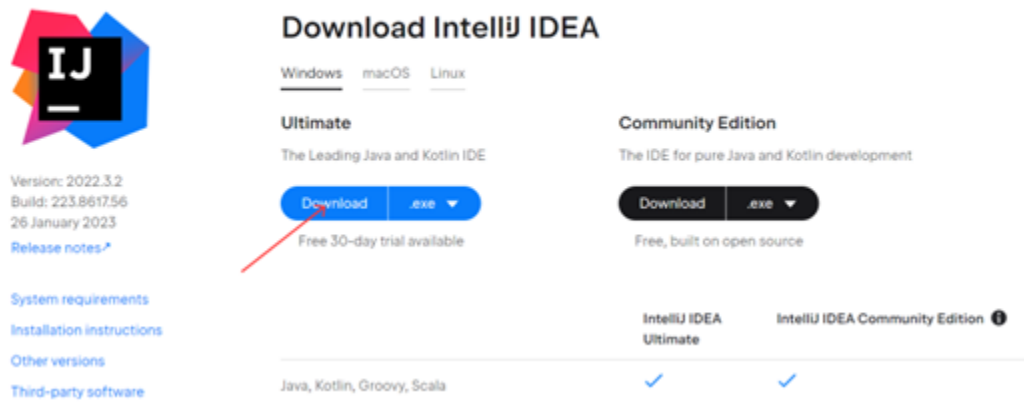
* Nhờ việc áp dụng nguyên lý SOLID, code được tổ chức logic hơn, các method, class cũng được tổ chức rõ ràng, chính xác hơn.

- Tính rõ ràng, dễ hiểu: Khi áp dụng SOLID vào thiết kế, nhất là trong khi làm việc nhóm, ta sẽ tạo ra các hàm dễ hiểu hơn, giúp code clean và dễ đọc hơn, từ đó giúp cho các bạn khác trong nhóm đọc hiểu code của nhau dễ hơn.
- Tính dễ thay đổi: SOLID giúp tạo ra các module, class rõ ràng, mạch lạc, mang tính độc lập cao. Do vậy khi có sự yêu cầu thay đổi, mở rộng, ta cũng dễ dàng thực hiện việc thay đổi hơn.
- Tính tái sử dụng: Nhờ tính rõ ràng, dễ hiểu và dễ thay đổi, mà nếu hệ thống khác cần sử dụng chức năng tương tự, thì nhờ SOLID cũng giúp tái sử dụng module dễ hơn.

8. Hướng dẫn sử dụng, cài đặt, kiểm thử

8.1. Hướng dẫn tải IntelliJ

- Vào trang web
<https://www.jetbrains.com/idea/download/#section=windows>
- Chọn hệ điều hành phù hợp với bạn

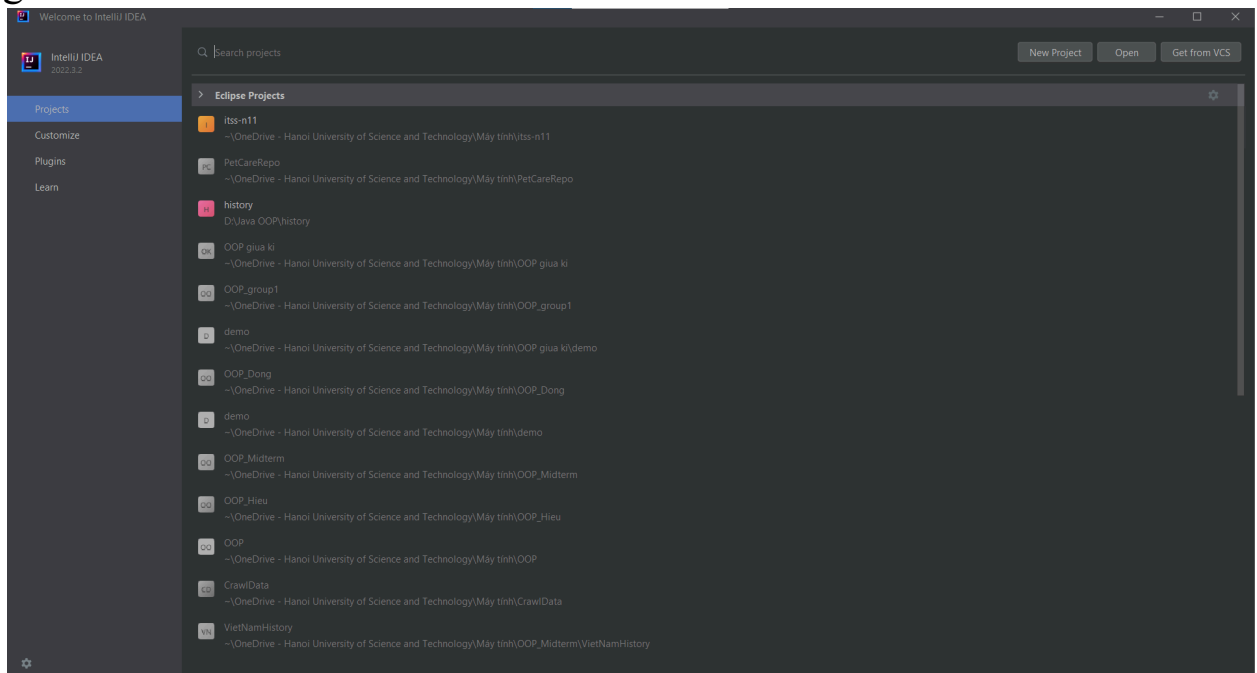


- Chọn Download để tải file setting về và chạy.
 - Sau khi cài đặt xong chạy file IntelliJIdea.exe.
- ### 8.2. Tải MySQL
- Truy cập vào link: <https://dev.mysql.com/downloads/mysql/>
 - Chọn hệ điều hành phù hợp với bạn, sau đó chọn **Download**.
 - Tải file cài đặt về và chạy file để tiến hành cài đặt MySQL.
 - Ở đây nhóm sử dụng database có config như dưới:
 - o Server: localhost

- o Database: mysql
- o Port: 3306
- o Username:
- o Password:
- Hoặc có thể tải về databases sau:
<Link database>

8.3. Clone project về local

- Thực hiện clone dự án về máy
- git clone



8.4. Set database và môi trường

- Sử dụng SDK 20
- Configure những library được đóng góp trong project

8.5. Chạy dự án

Để chạy dự án ta tiến hành vào theo đường dẫn `src/main/java/app/App.java` click chuột phải vào file `App.java` sau đó chọn **Build Project**, rồi chọn **Run Project**. Dự án khởi động và trải nghiệm dự án.