

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA**



BÀI TẬP LỚN

TÊN HỌC PHẦN : LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG

LỚP : N01, NHÓM : 9

HỌC KỲ II NĂM HỌC 2023 - 2024

ĐỀ TÀI:

ỨNG DỤNG MOBILE CAT

Mã Sinh Viên	Họ và Tên	Điểm
21011589	Trần Trọng Đài	
21011588	Trần Dương	
21010644	Nguyễn Văn Sơn	

Hà Nội, năm 2024

BẢNG PHÂN CÔNG

STT	Mã sinh viên	Họ và tên	Nhiệm vụ	Thời gian hoàn thành	Ghi chú
1	21011589	Trần Trọng Đài (Trưởng nhóm)	Trang chủ , giao diện yêu thích , báo cáo	Đúng hạn	
2	21011588	Trần Dương	Đăng ký , đăng nhập , yêu thích , cài đặt cá nhân , báo cáo .	Đúng hạn	
3	21010644	Nguyễn Văn Sơn	Tìm kiếm , báo cáo .	Đúng hạn	

MỤC LỤC

CHƯƠNG 1 : XÁC ĐỊNH VẤN ĐỀ	4
1.1 Đặt vấn đề	4
1.2 Yêu cầu chung	4
1.3 Giải pháp đề xuất	6
CHƯƠNG 2 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.	7
2.1. Phân tích yêu cầu	7
2.2. Các kịch bản của hệ thống (Use-cases)	7
2.3. Mô hình Use-case	17
2.4. Đặc tả Use-case	18
2.5. Các màn hình giao diện người dùng	26
CHƯƠNG 3 : TRIỂN KHAI	33
3.1 Giải pháp công nghệ	33
3.2 Thử nghiệm	36
KẾT LUẬN	37

CHƯƠNG 1 : XÁC ĐỊNH VẤN ĐỀ

1.1 Đặt vấn đề

Vì khó khăn trong việc tìm kiếm thông tin về Mèo, người dùng thường phải dành nhiều thời gian và nỗ lực. Việc này có thể gây bất tiện và làm giảm trải nghiệm của họ khi muốn hiểu về các loại mèo đặc biệt hoặc cách chăm sóc chúng.

Thiếu nguồn tài liệu đáng tin cậy cũng là một thách thức. Trên internet, thông tin về mèo có thể phong phú nhưng không luôn đảm bảo tính chính xác và đáng tin cậy. Điều này có thể dẫn đến nguy cơ không hiểu rõ về mèo hoặc cung cấp chăm sóc không đúng.

Thông tin về mèo thường phân mảnh trên nhiều nguồn khác nhau, từ sách đến trang web và các ứng dụng di động khác. Điều này gây ra sự phức tạp và khó khăn trong việc tìm kiếm và tiếp cận thông tin một cách thuận tiện.

Người dùng mong muốn có một trải nghiệm đơn giản và thuận tiện khi tìm kiếm thông tin về mèo. Việc thiếu hỗ trợ cho các chức năng cơ bản như tìm kiếm, chi tiết và yêu thích có thể làm giảm sự hài lòng của họ.

Dự án của chúng tôi nhằm giải quyết những vấn đề trên bằng cách cung cấp một ứng dụng di động đầy đủ, tin cậy và dễ sử dụng về mèo. Bằng cách kết hợp công nghệ Flutter và dữ liệu từ API về mèo, chúng tôi hy vọng sẽ mang lại cho người dùng trải nghiệm tốt nhất khi khám phá và hiểu biết về thế giới của mèo.

1.2 Yêu cầu chung

- Giao diện người dùng
 - + Thiết kế giao diện người dùng thân thiện, dễ sử dụng và thẩm mỹ.
 - + Đảm bảo tính thống nhất và mượt mà trên cả các thiết bị di động

Android và iOS.

- Chức năng tìm kiếm:

- + Cung cấp khả năng tìm kiếm mèo theo tên, từ khóa hoặc thành phần.
- + Hiển thị kết quả tìm kiếm một cách nhanh chóng và chính xác.

- Chức năng chi tiết:

- + Hiển thị thông tin chi tiết về mỗi loại Mèo, bao gồm tên , độ thông minh, nguồn gốc , mô tả chi tiết

- Chức năng yêu thích (favorite):

- + Cho phép người dùng lưu lại các loại Mèo mà họ yêu thích.

Cung cấp khả năng quản lý và truy cập nhanh chóng vào danh sách các loại mèo đã lưu.

- Tính năng offline:

- + Hỗ trợ tính năng lưu trữ dữ liệu để sử dụng khi không có kết nối mạng.
- + Đảm bảo tính ổn định và đồng bộ dữ liệu khi có kết nối mạng trở lại.

-Bảo mật và quyền riêng tư:

- + Bảo vệ thông tin cá nhân của người dùng.
- + Đảm bảo tính bảo mật cho dữ liệu được lưu trữ và truyền tải.

- Thử nghiệm và kiểm thử:

- + Tiến hành thử nghiệm chức năng và kiểm tra tính ổn định của ứng dụng trên các thiết bị và môi trường khác nhau.
- + Sửa chữa và cải thiện dự án dựa trên phản hồi từ người dùng và kết quả kiểm thử.

- Dữ liệu và API:

- + Sử dụng API để lấy dữ liệu về các loại mèo.

- + Đảm bảo tính ổn định và đáng tin cậy của kết nối API.

1.3 Giải pháp đề xuất

Trong việc phát triển một ứng dụng đồ uống hiệu quả và hấp dẫn, việc xác định và triển khai các giải pháp phù hợp là quan trọng đối với việc đáp ứng nhu cầu đa dạng của người dùng. Với sự tiến bộ trong công nghệ và dữ liệu, có nhiều cơ hội để cung cấp trải nghiệm đồ uống đáng tin cậy, chi tiết và dễ sử dụng hơn bao giờ hết. Dưới đây là một số giải pháp đề xuất được đưa ra để phát triển một ứng dụng đồ uống đa nền tảng tối ưu, cung cấp thông tin đồ uống đáng tin cậy và linh hoạt cho người dùng:

Tích hợp nhiều nguồn dữ liệu thời tiết: Đề xuất sử dụng nhiều nguồn dữ liệu về các loại mèo khác nhau API, hay dữ liệu từ các trang cập nhập chia sẻ thông tin đa dạng và chính xác nhất.

Giao diện người dùng thân thiện: Xây dựng giao diện người dùng thân thiện, dễ sử dụng và linh hoạt trên cả nền tảng di động và web.

CHƯƠNG 2 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.

2.1. Phân tích yêu cầu

Tìm kiếm (Search):

- Chức năng cho phép người dùng tìm kiếm các loại Mèo theo tên, từ khóa hoặc thành phần.
- Kết quả tìm kiếm được hiển thị một cách nhanh chóng và chính xác, giúp người dùng dễ dàng chọn lựa.

Chi tiết (Detail):

- Cho phép người dùng xem thông tin chi tiết về mỗi loại Mèo, bao gồm tên, hình ảnh, thành phần và lịch sử.
- Cung cấp các thông tin mô tả đầy đủ và chính xác để giúp người dùng hiểu rõ hơn về mỗi loại Mèo.

Yêu thích (Favorite):

- Chức năng cho phép người dùng lưu lại các loại mèo mà họ yêu thích vào danh sách yêu thích của mình.
- Người dùng có thể thêm, xóa hoặc chỉnh sửa các Mèo trong danh sách yêu thích của mình.

Quản lý tài khoản:

- Cho phép người dùng đăng nhập và đăng ký tài khoản để quản lý thông tin cá nhân và cài đặt ứng dụng.
- Đảm bảo tính bảo mật thông tin cá nhân của người dùng và quản lý tài khoản một cách an toàn.
- Cho phép người dùng có thể thay đổi mật khẩu và thông tin cá nhân đã đăng ký trong cài đặt.

2.2. Các kịch bản của hệ thống (Use-cases)

Kịch bản Use case Đăng nhập

Mã Use case	01	Tên Use case	Đăng nhập
Tác nhân	User		
Mô tả	Tác nhân đăng nhập vào hệ thống và sử dụng các chức năng của hệ thống tương ứng		
Sự kiện kích hoạt	Click vào nút đăng nhập có trên giao diện		
Tiền điều kiện	Tác nhân đã có tài khoản trên hệ thống		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	User	Chọn chức năng Đăng nhập
	2	Hệ thống	Hiển thị giao diện đăng nhập
	3	User	Nhập tên email, password (mô tả phía dưới *)
	4	User	Yêu cầu đăng nhập
	5	Hệ thống	Kiểm tra xem User đã nhập các trường bắt buộc hay chưa
	6	Hệ thống	Kiểm tra xem email, password nhập có

			hợp lệ hay không
	7	Hệ thống	Hiển thị chức năng tương ứng đối với User
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	6a	Hệ thống	Thông báo lỗi: cần nhập các trường bắt buộc nếu nhập thiếu
	7a	Hệ thống	Thông báo lỗi email hoặc password chưa đúng nếu không tìm thấy

Bảng 2.1: Kịch bản chức năng “Đăng nhập”

Dữ liệu đầu vào gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Email	Email	Có	Đúng định dạng	dai@gmail.com
2	Mật khẩu	Password	Có	Tối thiểu 6 kí tự	123456

Bảng 2.2: Dữ liệu chức năng “Đăng nhập”

2.2.2. Kịch bản Use case Đăng ký

Mã Use case	02	Tên Use case	Đăng ký
Tác nhân	User		
Mô tả	Tác nhân đăng ký tài khoản để sử dụng các chức năng của hệ thống tương ứng		
Sự kiện kích hoạt	Click vào nút đăng ký ở phần giao diện đăng nhập		
Tiền điều kiện	Không		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	User	Chọn chức năng Đăng ký
	2	Hệ thống	Hiển thị giao diện đăng ký
	3	User	Nhập các thông tin tài khoản (mô tả phía dưới *)
	4	User	Yêu cầu đăng ký
	5	Hệ thống	Kiểm tra xem khách đã nhập các trường bắt buộc hay chưa

	6	Hệ thống	Kiểm tra xem địa chỉ email có hợp lệ hay không
	7	Hệ thống	Kiểm tra xem email có bị trùng lặp hay không
	8	Hệ thống	Lưu thông tin tài khoản và chuyển đến trang đăng nhập
Luồng sự kiện thay thế	ST T	Thực hiện bởi	Hành động
	5a	Hệ thống	Thông báo lỗi: cần nhập các trường bắt buộc nếu nhập thiếu
	6a	Hệ thống	Thông báo lỗi: địa chỉ email không hợp lệ nếu địa chỉ email không hợp lệ
	7a	Hệ thống	thông báo lỗi: email đã tồn tại
	Tài khoản đã được tạo và lưu trữ vào hệ thống		

Bảng 2.3: Kịch bản chức năng “Đăng ký”

Dữ liệu đầu vào gồm các trường dữ liệu sau:

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	First name	FirstName	Có	Không rỗng	Đài
2	Last name	LastName	Có	Không rỗng	Trần
3	Age	age	Có	số dương	21
2	Email	Email	Có	Địa chỉ email hợp lệ, chưa được đăng ký trước đó	Dai@gmail.com
3	Mật khẩu	Password	Có	Tối thiểu 6 kí tự	Dai123456

Bảng 2.4: Kịch bản chức năng “Đăng ký”

2.2.3. Kịch bản Use case Tìm kiếm

Mã Use case	03	Tên Use case	Tìm kiếm
Tác nhân	User		
Mô tả	Tìm kiếm giống Mèo theo mong muốn		

Sự kiện kích hoạt	Click search button , nhập từ khóa và click vào tìm kiếm		
Tiền điều kiện	Sau khi đăng nhập thành công		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	User	Chọn chức năng tìm kiếm
	2	User	Nhập tên Mèo vào phần tìm kiếm
	3	User	Yêu cầu tìm kiếm
	4	Hệ thống	Tìm và lấy thông tin mèo thỏa mãn các tiêu chí tìm kiếm
	5	Hệ thống	Hiển thị thông tin về các loài mèo có tên chứa từ khóa tìm kiếm .
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Thông báo lỗi: Giống mèo không tồn tại.

Bảng 2.5: Kịch bản Tìm kiếm

2.2.4. Kịch bản Use case Xem thông tin Mèo

Mã Use case	04	Tên Use case	Xem thông tin chi tiết về Mèo
Tác nhân	User		
Mô tả	Xem thông tin chi tiết về Mèo (Mô tả, tên, độ thông minh, nguồn gốc ,...)		
Sự kiện kích hoạt	Click vào thẻ thành phần Mèo		
Tiền điều kiện	Sau khi đăng nhập thành công		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	User	Chọn chức năng đăng nhập
	2	Hệ thống	Sau khi đăng nhập thành công chuyển đến trang chủ
	3	Hệ thống	Hiển thị danh sách Mèo
	4	User	Chọn loài mèo bất kỳ
	5	Hệ thống	Hiển thị thông tin về Mèo ứng với yêu cầu của người dùng

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Thông báo lỗi: Không có thông tin

Bảng 2.6: Kịch bản Xem thông tin mèo

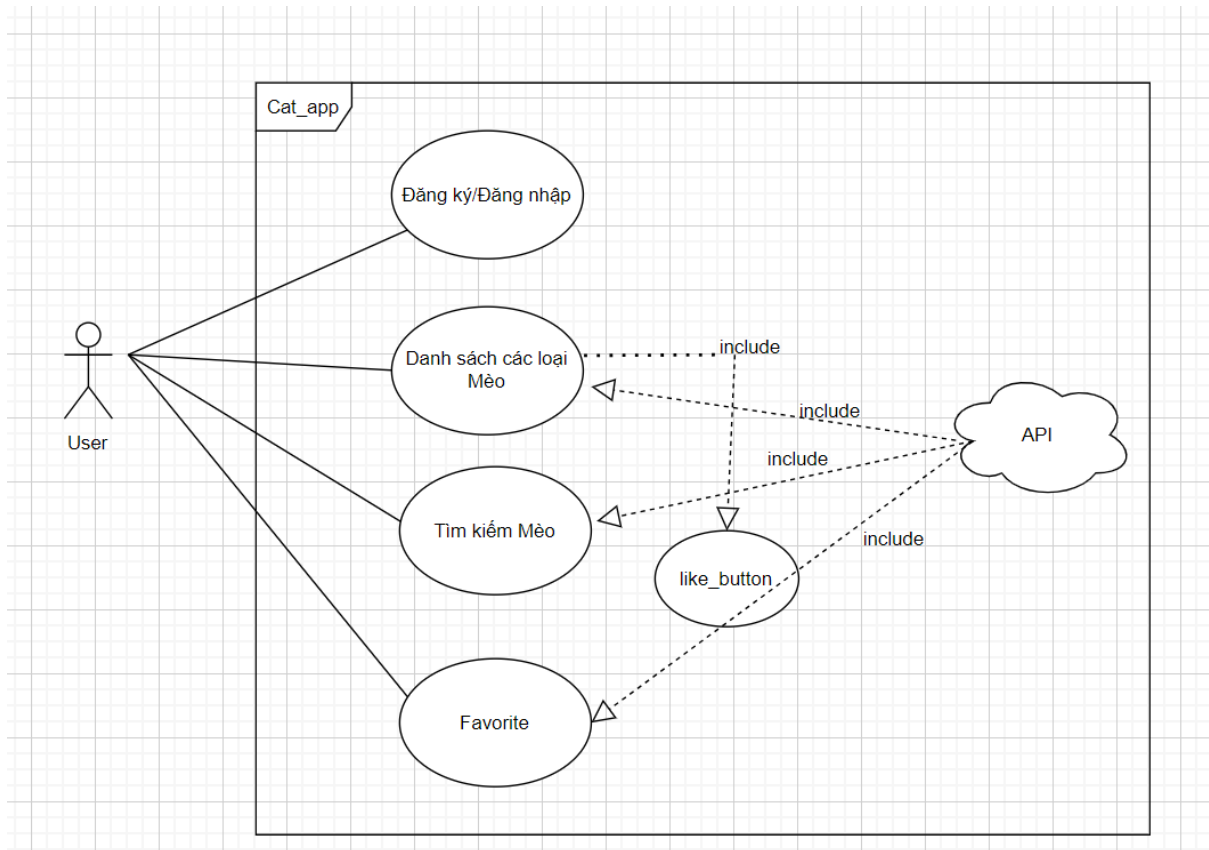
2.2.5. Kịch bản Use case Mèo yêu thích

Mã Use case	05	Tên Use case	Mèo yêu thích
Tác nhân	User		
Mô tả	Hiển thị danh sách những chú mèo yêu thích được người dùng chọn lựa		
Sự kiện kích hoạt	Ấn vào biểu tượng favorite		
Tiền điều kiện	Sau khi đăng nhập thành công		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1	User	Click vào biểu tượng favorite
	2	Hệ thống	Sau khi nhận được kết quả hệ thống lưu lại thông tin id của loài mèo và hiển thị lên danh sách bên page favourite

	3	Hệ thống	Hiển thị danh sách loại mèo được chọn khi người dùng click vào favourite
	4	User	Chọn biểu tượng trái tim để vào page favourite
	5	Hệ thống	Hiển thị danh sách favourite
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Thông báo lỗi: Chưa có chú mèo nào

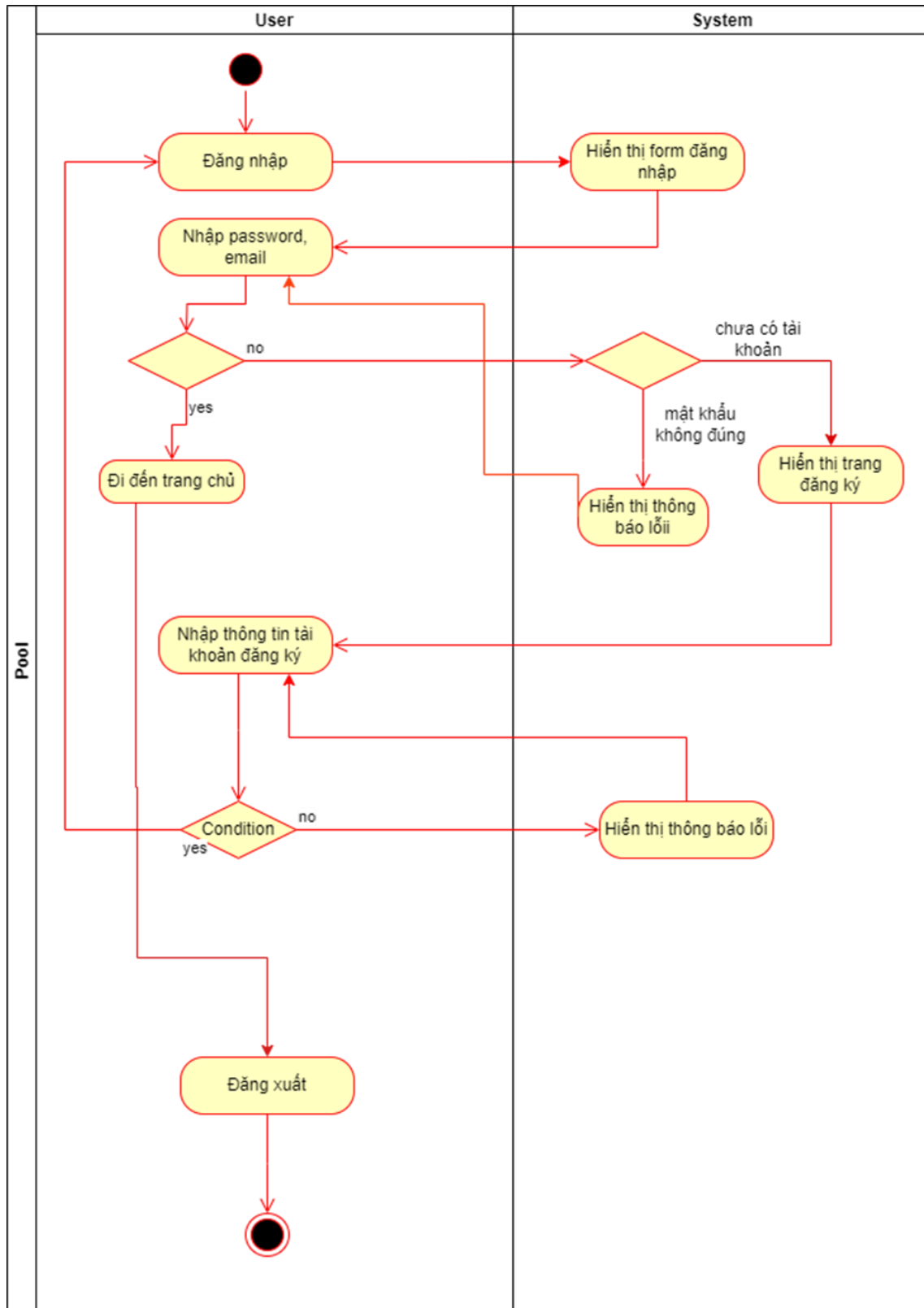
Bảng 2.6: Kịch bản Mèo yêu thích

2.3. Mô hình Use-case



Hình 2.1: Mô hình Use-case

2.4. Đặc tả Use-case



Hình 2.2: Biểu đồ hoạt động Đăng nhập- Đăng ký

***Đăng nhập**

Mô tả ngắn gọn: Cho phép người dùng đăng nhập vào hệ thống và sử dụng các tính năng của hệ thống.

Tác nhân: User.

Luồng sự kiện:

Luồng chính

Use-case sẽ bắt đầu khi tác nhân truy cập vào liên kết trang web của hệ thống.

- Hệ thống hiển thị trang đăng nhập gồm các trường nhập liệu email/số điện thoại và mật khẩu.
- Tác nhân nhập thông tin đăng nhập và xác nhận.
- Hệ thống kiểm tra và đưa tác nhân vào bảng điều khiển (dashboard) chính.

Luồng thay thế

Thông tin đăng nhập không hợp lệ: Tại luồng chính, nếu thông tin đăng nhập tác nhân nhập không hợp lệ, sai hoặc không tồn tại thì hệ thống sẽ hiển thị thông báo lỗi.

Yêu cầu đặc biệt: Không

Điều kiện tiên quyết: Người dùng đã có tài khoản trên hệ thống

Điều kiện cuối: Nếu tác nhân đăng nhập thành công thì trạng thái hệ thống sẽ thay đổi. Ngược lại, trạng thái hệ thống sẽ không đổi.

Điểm mở rộng: Không

*** Đăng ký**

- **Mô tả:**

Người dùng nếu chưa tạo tài khoản trước đó sẽ thực hiện thao tác đăng ký tài khoản mới.

- **Các tác nhân:** Người dùng chưa đăng ký
- **Điều kiện kích hoạt sử dụng:** Người dùng chưa có tài khoản
- **Tiền điều kiện:** Thiết bị được kết nối internet
- **Hậu điều kiện:** Không có

Các luồng sự kiện:

+ Luồng cơ bản:

Người dùng truy cập vào ứng dụng, nếu trước đó chưa tạo tài khoản chọn chức năng đăng ký tài khoản.

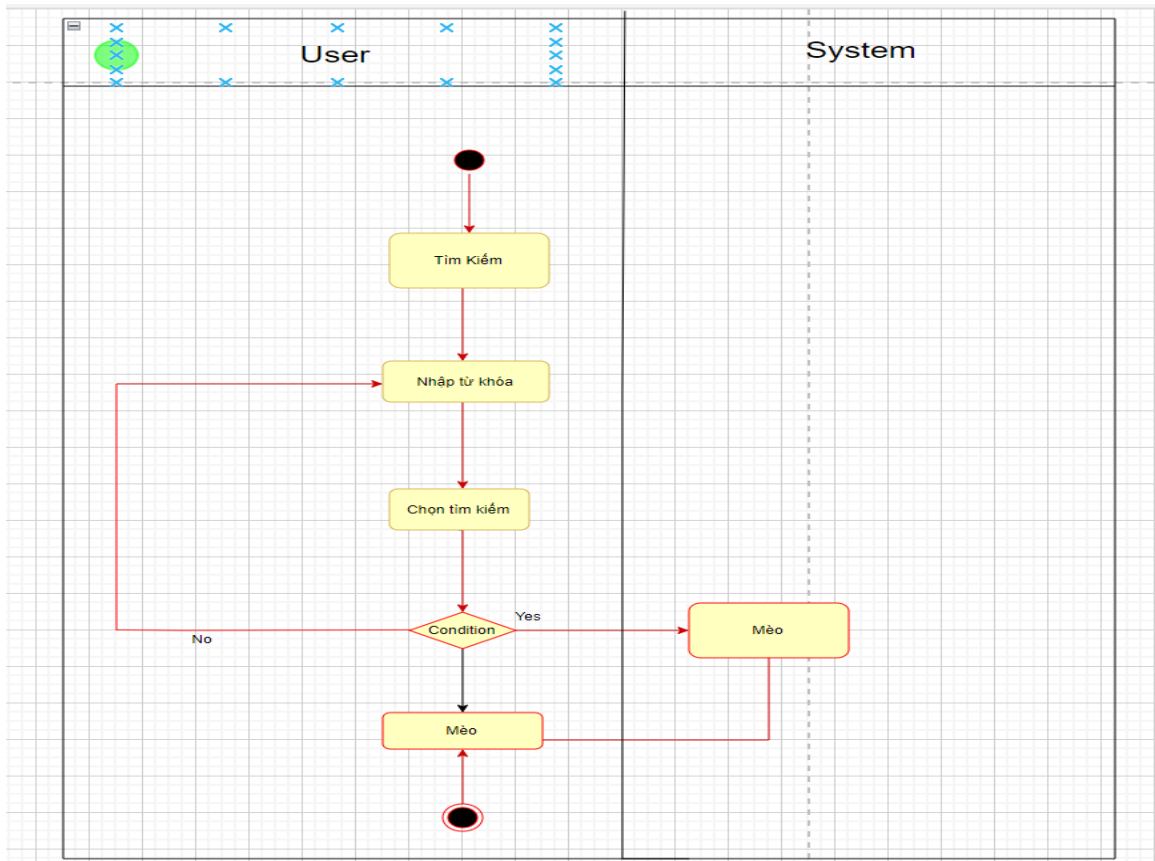
Người dùng thực hiện nhập các thông tin cần có: Họ tên , tuổi, email, password.

+ Luồng thay thế: Người dùng có thể đăng nhập trực tiếp bằng Facebook.

+ Luồng ngoại lệ: Hệ thống thực hiện đối chiếu nếu email đã tồn tại sẽ hiển thị thông báo:”Email đã tồn tại”

- **Yêu cầu phi chức năng:** không có
- **Extension point:** không có

***Tìm kiếm**



Hình 2.3: Biểu đồ hoạt động Tìm kiếm

- **Mô tả:** Người dùng sử dụng chức năng tìm kiếm Mèo theo nhu cầu.
 - **Các tác nhân:** Người dùng, hệ thống
 - **Điều kiện kích hoạt ca sử dụng:** Người dùng có nhu cầu xem các giống mèo, các loại mèo.
 - **Tiền điều kiện:** Thiết bị kết nối Internet
 - **Hậu điều kiện:** Không có
 - **Các luồng sự kiện:**
- + **Luồng sự kiện cơ bản:**
1. Người dùng truy cập vào trang tìm kiếm .
 2. Người dùng nhập từ khóa tìm kiếm vào thanh tìm kiếm và nhấn Enter hoặc nút tìm kiếm.
 3. Hệ thống nhận từ khóa tìm kiếm và truy vấn cơ sở dữ liệu để tìm.

4. Hệ thống hiển thị kết quả tìm kiếm cho người dùng.

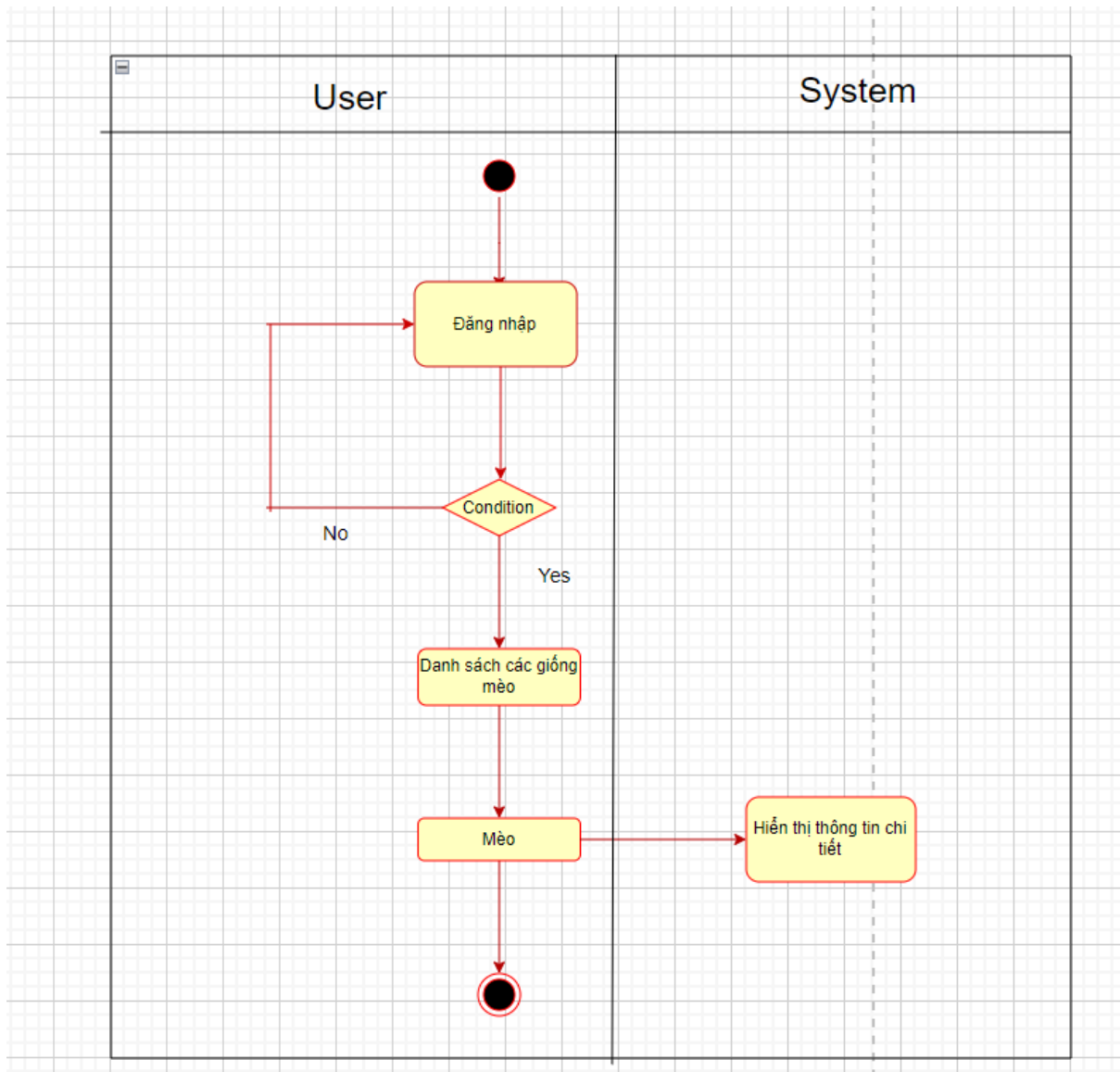
+ Luồng ngoại lệ:

Người dùng không nhập từ khóa tìm kiếm và nhấn Enter hoặc nút tìm kiếm. =>Hệ thống hiển thị một thông báo lỗi yêu cầu người dùng nhập từ khóa.

Cơ sở dữ liệu không trả về kết quả tìm kiếm nào cho từ khóa tìm kiếm.
=>Hệ thống hiển thị một thông báo rỗng cho người dùng.

- **Yêu cầu phi chức năng:** Không có
- **Extension point:** Không có

***Xem thông tin chi tiết**



Hình 2.4: Biểu đồ hoạt động Xem thông tin Mèo

- **Mô tả:** Người dùng sau khi đăng nhập thành công có thể xem được thông tin giống mèo sau khi chọn vào chú mèo đó.
- **Các tác nhân:** Người dùng, hệ thống
- **Điều kiện kích hoạt ca sử dụng:** Người dùng có nhu cầu xem thông tin chi tiết về loại Mèo đó đó
- **Tiền điều kiện:** Thiết bị kết nối Internet
- **Hậu điều kiện:** Không có

- **Các luồng sự kiện:**

+ **Luồng sự kiện cơ bản:**

1. Người dùng truy cập vào trang chủ .
2. Hệ thống hiển thị danh sách Mèo đề xuất.
3. Người dùng chọn Mèo muốn tìm hiểu
4. Hệ thống hiển thị kết quả tương ứng tùy theo lựa chọn của người dùng

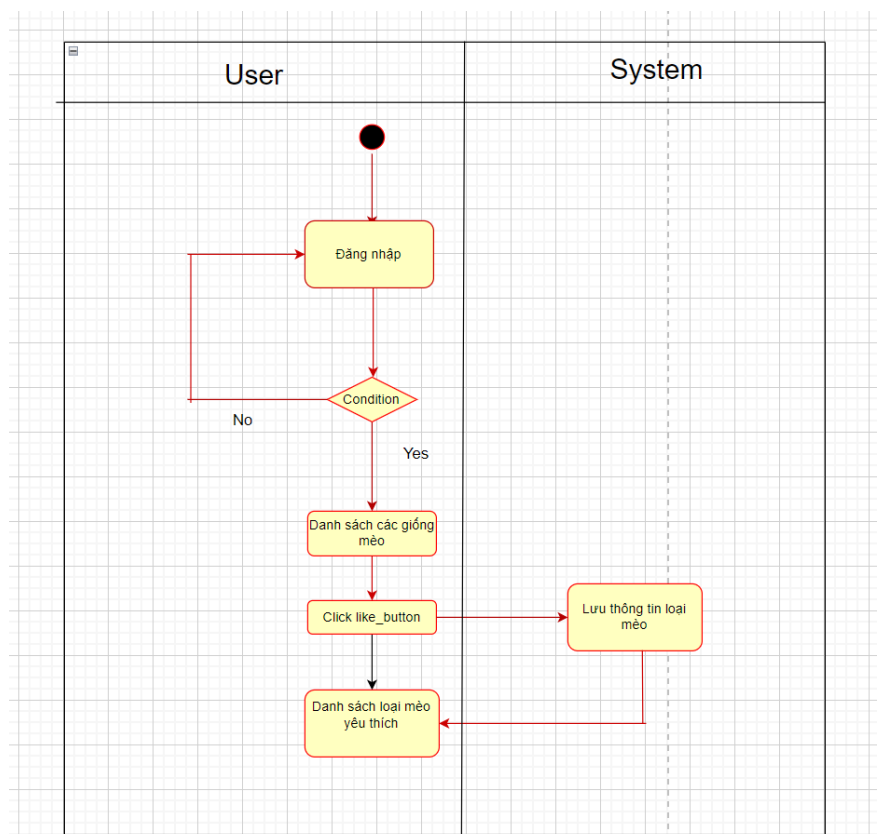
+ **Luồng ngoại lệ:** Không có

+ **Luồng thay thế:** Không có

- **Yêu cầu phi chức năng:** Không có

- **Extension point:** Không có

***Mèo yêu thích**

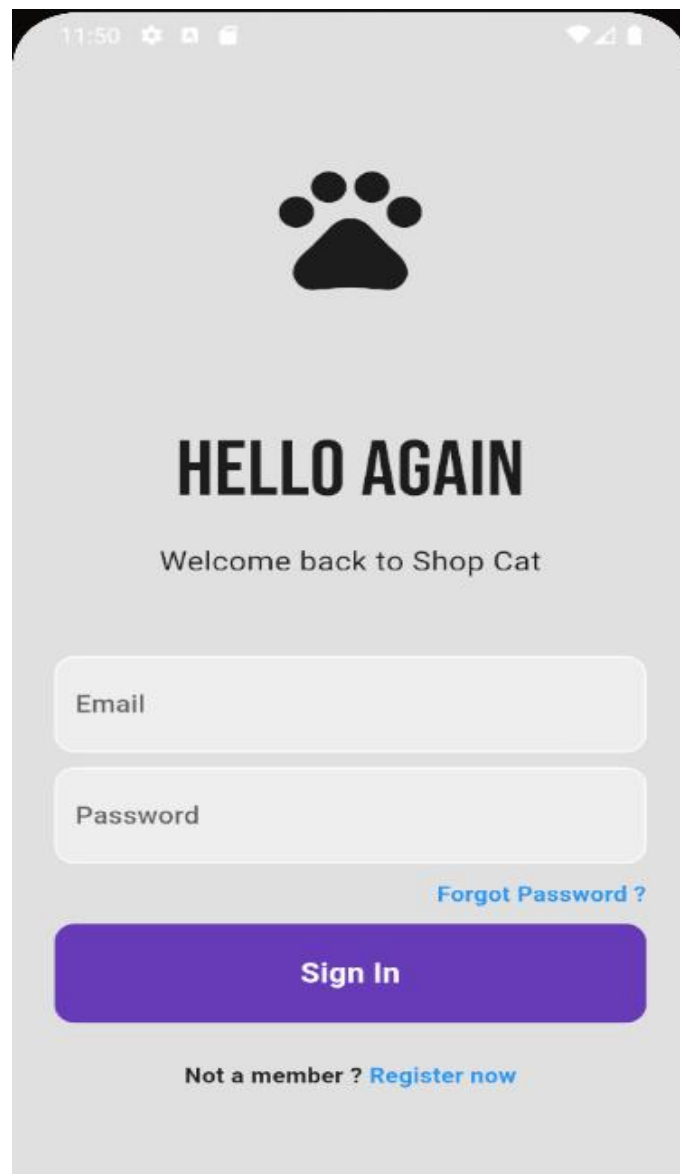


Hình 2.5: Biểu đồ hoạt động Mèo yêu thích

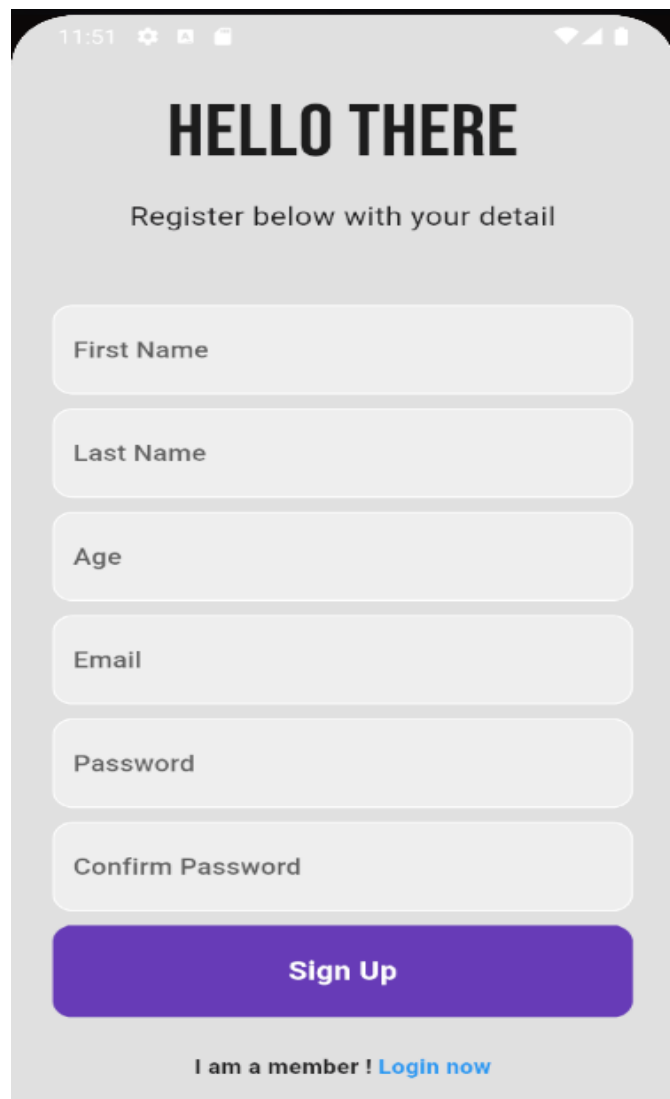
- **Mô tả:** Hệ thống lưu giữ thông tin loài mèo mà người dùng click vào biểu tượng yêu thích.
- **Các tác nhân:** người dùng, hệ thống
- **Điều kiện kích hoạt sử dụng:** User đã đăng nhập vào ứng dụng, chọn loài mèo yêu thích
- **Tiền điều kiện:** Thiết bị kết nối internet.
- **Hậu điều kiện:** Không có
- **Các luồng sự kiện:**
- + **Luồng sự kiện chính:**
 1. Người dùng click vào nút Like_Button
 2. Hệ thống lưu thông tin và trả về danh sách có loài mèo yêu thích thông qua loài mèo được chọn(True or False)
 3. Truy cập page Favourite.
 4. Hệ thống hiển thị danh sách favourite.
- **Yêu cầu phi chức năng:** Không có
- **Extension point:** Không có

2.5. Các màn hình giao diện người dùng

*Giao diện đăng nhập



***Giao diện đăng ký**



A mobile application registration screen with a light gray background. At the top, the status bar shows the time 11:51 and various icons. The main heading is "HELLO THERE" in large, bold, black capital letters. Below it, the instruction "Register below with your detail" is centered. The registration form consists of six rounded rectangular input fields stacked vertically, each with a placeholder label: "First Name", "Last Name", "Age", "Email", "Password", and "Confirm Password". Below these fields is a prominent purple button with the text "Sign Up" in white. At the bottom, there is a line of text: "I am a member ! [Login now](#)".

11:51

HELLO THERE

Register below with your detail

First Name

Last Name

Age

Email

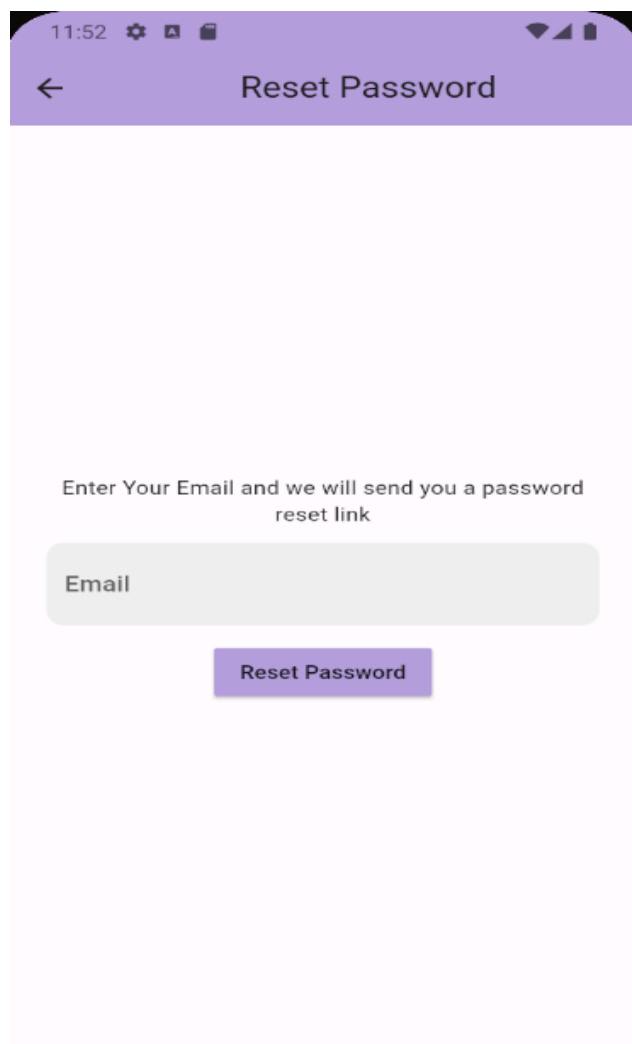
Password

Confirm Password

Sign Up

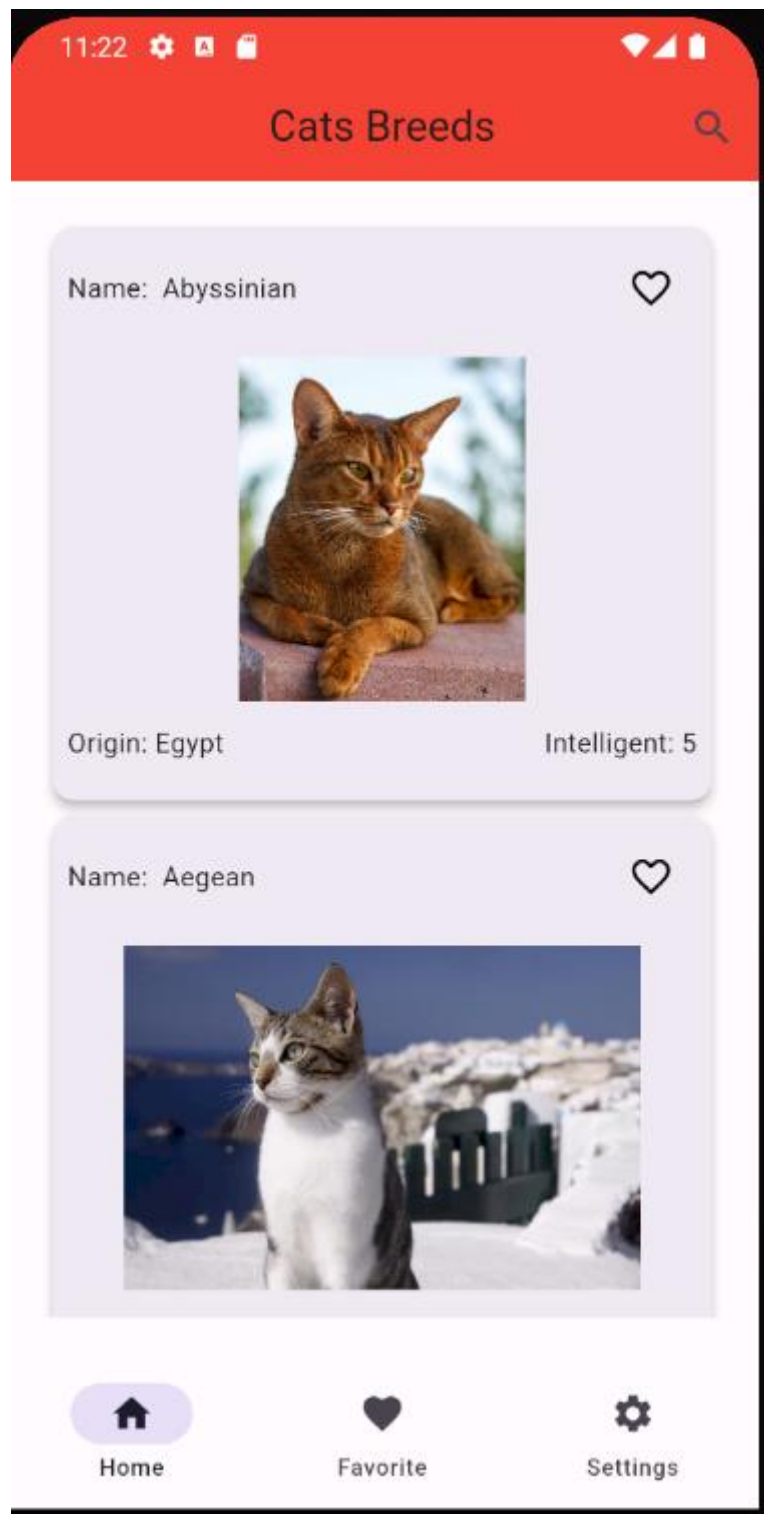
I am a member ! [Login now](#)

***Giao diện quên mật khẩu**



The image shows a mobile application interface for resetting a password. At the top, there is a purple header bar with a back arrow on the left and the text "Reset Password" in the center. Below the header, the background is a light pink color. In the center of the screen, there is a text prompt: "Enter Your Email and we will send you a password reset link". Below this prompt is a light gray rounded rectangular input field with the placeholder text "Email". Underneath the input field is a purple rectangular button with the text "Reset Password" in white.

***Giao diện trang chủ**



Hình 2.6: Giao diện trang chủ

Để thiết kế ra giao diện trang chủ chúng cho một hệ thống app Mèo thì chúng tôi sử dụng một tone màu trầm để làm tăng vẻ đẹp cho chủ thể. Về bố cục trên appBar sẽ là tên app, phần body sẽ hiển thị ảnh của loài mèo, góc trên sẽ là tên và một nút bấm yêu thích, góc dưới sẽ là nguồn gốc và độ thông minh của mèo. Nút bấm like_button cho phép người dùng click vào để lưu về giá trị yêu thích.

***Giao diện giống mèo yêu thích**



Hình 2.7:Giao diện giống mèo yêu thích.

Sau khi người dùng click vào like_button thì hệ thống sẽ lưu giống mèo đó và truy vấn sang trang favourite và hiển thị trả về đồ uống yêu mà người dùng đã ấn like.

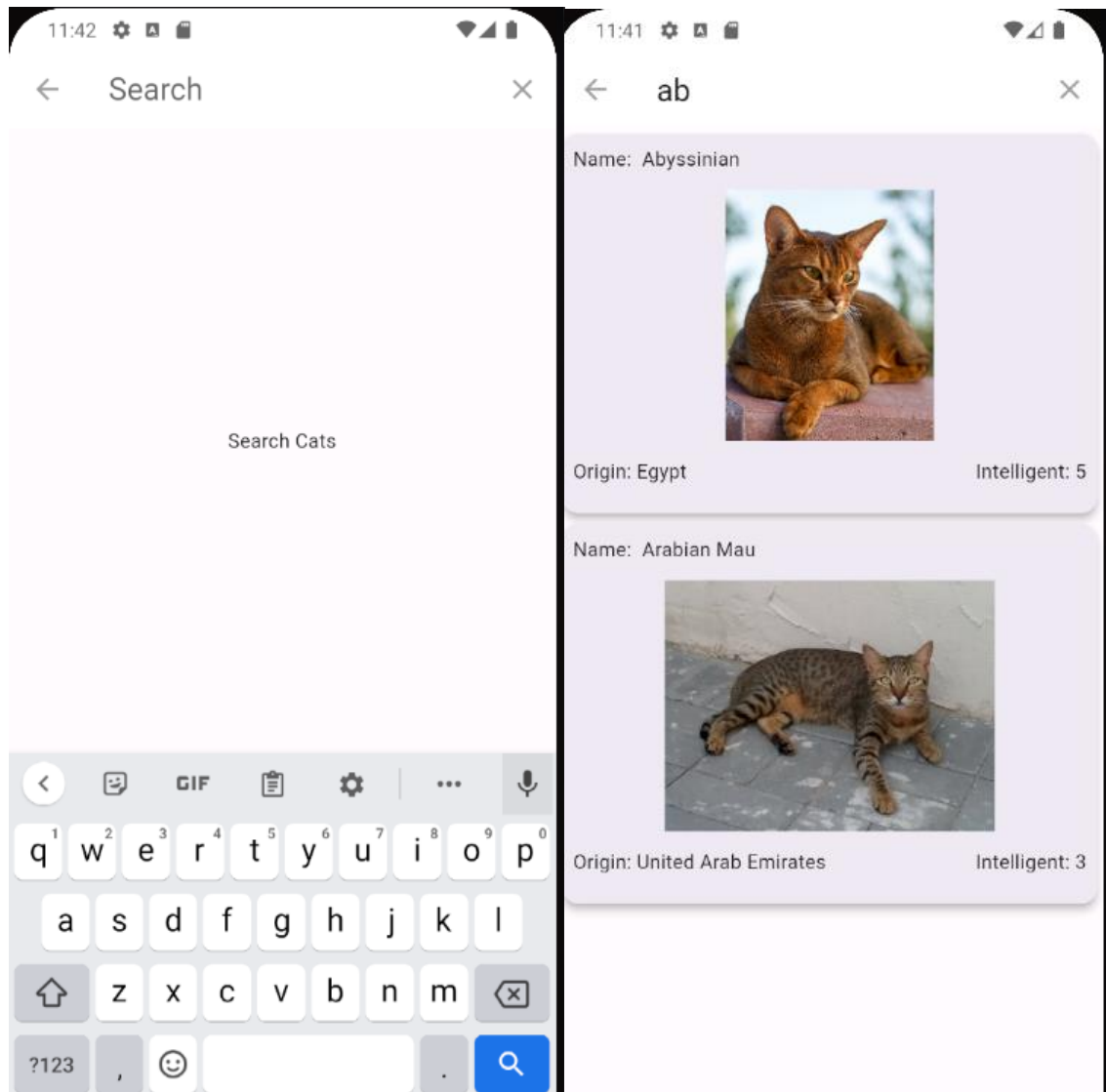
***Giao diện thông tin chi tiết.**



Hình 2.8:Giao diện chi tiết giống mèo

Giao diện thông tin sẽ hiển thị ra hình ảnh của giống mèo đó. Bên dưới hình ảnh sẽ hiển thị ra các thông tin chi tiết như nguồn gốc , độ thông minh , tuổi thọ trung bình , các đặc trưng và mô tả .

***Giao diện tìm kiếm Giống Mèo.**



Hình 2.9: Giao diện tìm kiếm Giống mèo.

Ở đây hệ thống hiển thị ra một giao diện tìm kiếm cơ bản có thanh Search và kết quả trả về bên dưới. Sau khi người dùng điền kết quả vào thanh Search hệ thống trả về kết quả thông qua thông tin được nhập vào. Ở đây người dùng cũng có thể ấn like_button và xem thông tin chi tiết của mèo.

CHƯƠNG 3 : TRIỂN KHAI

3.1 Giải pháp công nghệ

*Khai báo lớp CatBreedsImage

```
class CatBreedsImage {
    String? id;
    String? url;
    List<Breed> breeds;
    int? width;
    int? height;

    CatBreedsImage({
        required this.id,
        required this.url,
        required this.breeds,
        required this.width,
        required this.height,
    });

    factory CatBreedsImage.fromJson(Map<String, dynamic> json) => CatBreedsImage(
        id: json["id"],
        url: json["url"],
        breeds: List<Breed>.from(json["breeds"].map((x) => Breed.fromJson(x))),
        width: json["width"],
        height: json["height"],
    ); // CatBreedsImage

    Map<String, dynamic> toJson() => {
        "id": id,
        "url": url,
        "breeds": List<dynamic>.from(breeds.map((x) => x.toJson())),
        "width": width,
        "height": height,
    };
}
```

Hình 3.2: Code khai báo class CatBreedsImage.

‘id’: Mã của giống mèo, mỗi giống mèo có một mã riêng .

‘url’ : Đường link dẫn đến ảnh của mèo.

‘breed’ : Một mảng chứa các thông tin chi tiết của giống mèo đó

‘width’ : Chiều rộng của ảnh mèo.

‘height’ : Chiều cao của ảnh mèo.

-Phương thức:

‘**fromJson**’: Chuyển đổi dữ liệu JSON thành danh sách các đối tượng CatBreedsImage.

‘**toJson**’: Chuyển đổi một đối tượng của lớp thành một đối tượng JSON

*Khai báo phương thức like_button

```
final user = FirebaseAuth.instance.currentUser!;  
Future addCatFavorite(String id, String email) async {  
    await FirebaseFirestore.instance.collection('cat_favorite').add({  
        'id': id,  
        'email': email,  
    });  
}
```

Hình 3.3: Code khai báo phương thức Like_button

‘**AddCatFavorite**’ được thiết kế để xử lý sự kiện khi người dùng nhấn vào nút yêu thích vào giống mèo nào thì phương thức này xử lý việc thêm một giống mèo đó vào danh sách yêu thích của người dùng.

Khai báo các đường dẫn API

```
class Common {  
    final String baseUrl = "https://api.thecatapi.com/v1";  
    final String baseUrlDetail = "https://api.thecatapi.com/v1/images/";  
    final String apiKey = 'live_w09IiJwLFyZYIIhyoHF16XQ87Y6zMBQ8VOTdTIQbwUhRRb1aaOsIEnxcl2mSylvm3';  
    final String baseUrlImageCats = 'https://cdn2.thecatapi.com/images/';  
    final String baseUrlLoadingCats = 'https://img.icons8.com/cotton/256/null/cat--v2.png';  
    final String baseUrlLogoCat = 'https://img.icons8.com/cotton/256/null/cat--v2.png';  
}
```

baseUrl: Một chuỗi hằng số đại diện cho URL cơ sở của API TheCatAPI. Đây là nơi mà các yêu cầu API sẽ được gửi đến.

baseUrlDetail: Một chuỗi hằng số đại diện cho URL cơ sở của API TheCatAPI khi truy vấn chi tiết hình ảnh.

apiKey: Một chuỗi hằng số đại diện cho khóa API được sử dụng để xác thực các yêu cầu API. Khóa này cần được cung cấp khi gửi yêu cầu đến API để được xác thực.

baseUrlImageCats: Một chuỗi hằng số đại diện cho URL cơ sở của hình ảnh mèo trên TheCatAPI. Đây là nơi các hình ảnh mèo được lưu trữ và có thể truy cập thông qua API.

baseUrlLoadingCats: Một chuỗi hằng số đại diện cho URL của một biểu tượng hoặc hình ảnh được sử dụng để hiển thị khi đang tải dữ liệu của mèo. Đây có thể là một hình ảnh đại diện cho việc tải dữ liệu mèo.

baseUrlLogoCat: Một chuỗi hằng số đại diện cho URL của biểu tượng hoặc hình ảnh đại diện cho logo của TheCatAPI hoặc các sản phẩm liên quan đến mèo.

*Khai báo và lấy thông tin các giống Mèo

```
Future<dynamic> getListBreeds(
    {Map<String, dynamic> params = const {}, String? query}) async {
    String content = '';
    List<CatBreeds> myListBreeds = [];

    content = (params.keys
        .map((key) => '$key=${Uri.encodeQueryComponent(params[key])}')
        .join('&'));
    content = (params.keys
        .map((key) => '$key=${Uri.encodeQueryComponent(params[key])}')
        .join('&'));

    final Map<String, String> headers = {
        'Content-Type': 'application/x-www-form-urlencoded',
        'Content-Length': utf8.encode(content).length.toString(),
        'x-api-key': Common().apiKey,
    };

    try {
        debugPrint('-> ${Common().baseUrl + EndPoints().breeds}$content');
        await http
            .get(
                headers: headers,
                Uri.parse('${Common().baseUrl + EndPoints().breeds}$content'),
            )
            .then((response) {
                if (response.statusCode == 200) {
                    List jsonList =
                        jsonDecode(response.body.replaceAll('\\', '')) as List;
                    myListBreeds = jsonList
                        .map((jsonElement) => CatBreeds.fromJson(jsonElement))
                        .toList();
                } else {
                    myListBreeds = [];
                    debugPrint('Loi ket noi:');
                    debugPrint(response.statusCode.toString());
                }
            });
    } catch (e) {
        debugPrint('Loi ket noi:');
        debugPrint(e.toString());
        return [];
    }
    return myListBreeds;
}
```

Hình 3.4: Code khai báo địa chỉ API

Khai báo phương thức: 'getListBreeds()'

- Xây dựng URL yêu cầu API dựa trên id của giống mèo.
- Gửi yêu cầu HTTP đến API để lấy dữ liệu về các giống mèo.
- Phân tích dữ liệu trả về từ API thành một mảng các đối tượng CatBreeds .



Hình 3.5: Mô hình hoạt động API

Trong ứng dụng Flutter của chúng tôi, chúng tôi sử dụng một API từ trang web "TheCatApi" để lấy thông tin về các loại mèo.

- TheCatApi là một API công cộng cung cấp thông tin về các loại mèo, bao gồm tên, thông tin chi tiết, hình ảnh,...

-Các chức năng chính:

1. Tìm kiếm giống mèo: Cho phép tìm kiếm các giống mèo dựa trên tên.
2. Xem chi tiết giống mèo: Cung cấp thông tin chi tiết về một giống mèo cụ thể, bao gồm tên, thông tin chi tiết và hình ảnh của nó.
3. Cập nhật danh sách yêu thích: Cho phép người dùng thêm hoặc xóa các giống mèo trong danh sách yêu thích của họ.
4. Cài đặt thông tin cá nhân của tài khoản, đổi mật khẩu tài khoản.

3.2 Thử nghiệm

- Ứng dụng được xây dựng dựa trên nhu cầu và được hoàn thành trong thời gian không quá dài. Sau khi test, ứng dụng có hiệu quả vận hành tương đối ổn, các thông tin, tính toán được cập nhật chính xác, giao diện đẹp, dễ sử dụng, phù hợp với yêu cầu của người dùng,

KẾT LUẬN

Ứng dụng của nhóm đã đáp ứng được cơ bản yêu cầu của người dùng. Phần mềm được thiết kế với các chức năng hiển thị các giống mèo và cập nhật vào dữ liệu cùng một số tính năng khác. Vì thời gian xây dựng phần mềm không quá dài nên chưa thể cập nhật được thêm các chức năng vượt trội hơn nhưng phần mềm vẫn đảm bảo các chức năng theo yêu cầu người dùng. Trong tương lai, nhóm sẽ cải thiện thêm về mặt hình ảnh và các tính năng tốt hơn nữa.