

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP
NGÀNH KHOA HỌC MÁY TÍNH

XÂY DỰNG ỨNG DỤNG DI ĐỘNG DẠY VẼ CHO TRẺ NHỎ
ÁP DỤNG CÔNG NGHỆ XỬ LÝ ẢNH SỬ DỤNG FLUTTER VÀ
KOTLIN TRÊN NỀN TẢNG ANDROID

CBHD : TS. Nguyễn Thị Hoa Huệ

Sinh viên : Nguyễn Văn Đạt

Mã sinh viên: 2020607455

NGUYỄN VĂN ĐẠT

KHOA HỌC MÁY TÍNH

Hà Nội – Năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

NGÀNH: KHOA HỌC MÁY TÍNH

**XÂY DỰNG ỨNG DỤNG DI ĐỘNG DẠY VẼ CHO TRẺ NHỎ ÁP
DỤNG CÔNG NGHỆ XỬ LÝ ẢNH SỬ DỤNG FLUTTER VÀ KOTLIN
TRÊN NỀN TẢNG ANDROID**

Giáo viên hướng dẫn: TS. Nguyễn Thị Hoa Huệ

Sinh viên: Nguyễn Văn Đạt

Mã sinh viên: 2020607455

Lớp: 2020DHKHMT01

Hà Nội - 2024

LỜI CẢM ƠN

Đầu tiên, em xin được gửi lời cảm ơn sâu sắc đến giáo viên hướng dẫn **TS. Nguyễn Thị Hoa Huệ**, người luôn nhiệt tình hướng dẫn, góp ý và chia sẻ kinh nghiệm giúp em vượt qua những khó khăn trong quá trình nghiên cứu và hoàn thiện đề tài của mình.

Tiếp đó, em xin gửi lời cảm ơn chân thành đến toàn thể các quý thầy, cô trong trường Đại học Công nghiệp Hà Nội, đặc biệt là các thầy, cô công tác tại khoa Công nghệ thông tin vì đã tạo điều kiện thuận lợi và giúp đỡ tận tình cho sinh viên chúng em.

Đồng thời, em cũng muốn gửi lời cảm ơn chân thành nhất đến gia đình, bạn bè đã luôn sát cánh, khích lệ và động viên em trong quá trình thực hiện đồ án tốt nghiệp này.

Em xin chân thành cảm ơn quý Thầy Cô và em xin chúc thầy cô thật nhiều sức khỏe để tiếp tục chỉ bảo, dẫn dắt các thế hệ sinh viên bước tiếp trên con đường học tập và rèn luyện bản thân!

Kính thầy, cô!

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT	iv
DANH MỤC BẢNG BIỂU	v
DANH MỤC HÌNH VẼ	vi
MỞ ĐẦU	1
CHƯƠNG 1. TỔNG QUAN VỀ ỨNG DỤNG DI ĐỘNG DẠY VẼ CHO TRẺ NHỎ.....	4
1.1. Giới thiệu chung.....	4
1.2. Mô tả bài toán.....	4
1.3. Khảo sát người dùng	5
1.4. Yêu cầu chức năng	10
1.5. Yêu cầu phi chức năng.....	11
1.6. Yêu cầu thiết bị	11
1.7. Cơ sở lý thuyết	12
1.7.1. Tổng quan về lập trình di động	12
1.7.2. Công cụ lập trình Android Studio	14
1.7.3. Ngôn ngữ lập trình Dart	16
1.7.4. Một số framework trong lập trình Flutter	16
1.7.5. Ngôn ngữ lập trình Kotlin	17
1.7.6. SQLite Database.....	18
1.7.7. Tổng quan về mô hình MVC	20
1.7.8. OpenCV và Thuật toán Xử lý Ảnh	23

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU PHẦN MỀM	29
2.1. Mô hình hóa chức năng.....	29
2.1.1. Các tác nhân của hệ thống.....	29
2.1.2. Các use case của hệ thống.....	29
2.1.3. Biểu đồ use case.	30
2.1.4. Đặc tả Usecase	30
2.1.5. Biểu đồ chi tiết Use Case	39
2.2. Thiết kế giao diện.....	50
2.2.1. Giao diện Home	50
2.2.2. Giao diện vẽ	51
2.2.3. Giao diện danh sách ảnh có sẵn theo chủ đề.....	53
2.2.4. Giao diện quay chụp khi vẽ.....	54
2.2.5. Giao diện xem danh sách ảnh yêu thích.....	56
CHƯƠNG 3. KẾT QUẢ THU ĐƯỢC	57
3.1. Cài đặt môi trường	57
3.2. Cài đặt chương trình.....	58
3.3. Kết quả thu được	61
3.3.1. Chọn ảnh với nhiều nguồn ảnh	61
3.3.2. Phân tích ảnh thành đường nét.....	61
3.3.3. Quay lại quá trình vẽ.....	61
3.3.4. In ảnh.....	61
3.4. Kết quả kiểm thử	61
3.4.1. Chiến lược kiểm thử.....	61

3.4.2. Kết quả kiểm thử	61
KẾT LUẬN	64
TÀI LIỆU THAM KHẢO	65

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Viết tắt	Dịch nghĩa
1	HTML	Hyper Text Markup Language
2	CSS	Cascading Style Sheets
3	GPS	Global Positioning System
4	IDE	Integrated Development Environment
5	URL	Uniform Resource Locator
6	APK	Android Package Kit
7	MVC	Model – View – Controller
8	CSDL	Cơ sở dữ liệu
9	SDK	Software Development Kit
10	SQL	Structured Query Language

DANH MỤC BẢNG BIỂU

Bảng 2.1 Bảng các use case của hệ thống.	29
Bảng 2.2 Mô tả use case “Chọn ảnh”	30
Bảng 2.3 Mô tả use case Vẽ ảnh.....	32
Bảng 2.4 Mô tả usecase "Quay video"	34
Bảng 2.5 Mô tả usecase chụp ảnh.....	35
Bảng 2.6 Mô tả use case “Xuất ảnh”	36
Bảng 2.7 Mô tả usecase “Quản lý ảnh”	37
Bảng 3.1 Bảng kiểm thử chức năng.....	62

DANH MỤC HÌNH VẼ

Hình 1.1 Kết quả khảo sát câu hỏi 1	6
Hình 1.2 Kết quả khảo sát câu hỏi 2	6
Hình 1.3 Kết quả khảo sát câu hỏi 3	7
Hình 1.4 Kết quả khảo sát câu hỏi 4	7
Hình 1.5 Kết quả khảo sát câu hỏi 5	8
Hình 1.6 Kết quả khảo sát câu hỏi 6	8
Hình 1.7 Kết quả khảo sát câu hỏi 7	9
Hình 1.8 Kết quả khảo sát câu hỏi 8	9
Hình 1.9 Kết quả khảo sát câu hỏi 9	10
Hình 1.10 Kết quả khảo sát câu hỏi 10	10
Hình 1.11: Mô hình MVC.....	20
Hình 1.12 Các bước của quy trình xử lý ảnh.....	26
Hình 1.13 Hình ảnh đầu vào	28
Hình 1.14 Hình ảnh kết quả	28
Hình 2.1 Biểu đồ lớp use case "Chọn ảnh.".....	39
Hình 2.2 Biểu đồ trình tự use case chọn ảnh.	40
Hình 2.3 Biểu đồ lớp use case "Vẽ ảnh."	41
Hình 2.4 Biểu đồ trình tự use case "Vẽ ảnh."	42
Hình 2.5 Biểu đồ lớp use case "Quay video"	43
Hình 2.6 Biểu đồ trình tự use case "Quay video."	44
Hình 2.7 Biểu đồ lớp use case "Chụp Ảnh."	45
Hình 2.8 Biểu đồ trình tự use case "Chụp Ảnh"	46

Hình 2.9 Biểu đồ lớp use case "Xuất Ảnh."	46
Hình 2.10 Biểu đồ trình tự use case "Xuất ảnh."	47
Hình 2.11 Biểu đồ lớp use case "Quản lý ảnh."	48
Hình 2.12 Biểu đồ trình tự use case "Quản lý ảnh."	49
Hình 2.13 Giao diện màn Home	50
Hình 2.14 Giao diện xem ảnh sau khi chọn	51
Hình 2.15 Giao diện chức năng vẽ sketch	51
Hình 2.16 Giao diện chức năng vẽ trace	52
Hình 2.17 Giao diện chỉnh sửa ảnh	52
Hình 2.18 Giao diện danh sách ảnh theo chủ đề	53
Hình 2.19 Giao diện chụp ảnh	54
Hình 2.20 Giao diện sau khi chụp	54
Hình 2.21 Giao diện quay video	55
Hình 2.22 Giao diện sau khi quay video	55
Hình 2.23 Giao diện xem ảnh yêu thích	56
Hình 2.24 Giao diện khi xóa hình ảnh	56
Hình 3.1 Trạng thái cài đặt môi trường	57
Hình 3.2 Giao diện bắt đầu Android studio	58
Hình 3.3 Cửa sổ chọn đường dẫn SDK Flutter	59
Hình 3.4 Cửa sổ Tạo project	60
Hình 3.5 Giao diện project	60

MỞ ĐẦU

1. Lý do chọn đề tài.

Đề tài "**Xây dựng ứng dụng di động dạy vẽ cho trẻ nhỏ áp dụng công nghệ xử lý ảnh sử dụng Flutter và Kotlin trên nền tảng Android**" được lựa chọn dựa trên nhu cầu thực tế và xu hướng hiện tại trong việc sử dụng thiết bị di động ở trẻ em. Các yếu tố dưới đây đã đóng góp vào quyết định chọn đề tài này:

- **Trẻ hóa độ tuổi sử dụng thiết bị di động:** Ngày càng có nhiều trẻ nhỏ sử dụng điện thoại thông minh ngay từ khi còn rất nhỏ. Điều này tạo ra một thị trường tiềm năng cho các ứng dụng phục vụ riêng cho trẻ em.
- **Nhu cầu của phụ huynh:** Trong khi bận rộn với công việc, các bậc phụ huynh tìm kiếm những ứng dụng an toàn và bổ ích để trẻ có thể tự giải trí một cách lành mạnh. Một ứng dụng vẽ cho phép trẻ phát triển kỹ năng sáng tạo của mình, đồng thời giúp phụ huynh yên tâm khi biết con mình đang tiếp xúc với nội dung phù hợp và thúc đẩy tư duy nghệ thuật.
- **Xu hướng của các bộ hoạt hình và anime:** Sự phổ biến của hoạt hình và anime đã làm nảy sinh sở thích vẽ các nhân vật yêu thích ở trẻ nhỏ. Tuy nhiên, không phải trẻ nào cũng có khả năng tự vẽ một cách dễ dàng. Ứng dụng với công nghệ xử lý ảnh có thể giúp trẻ dễ dàng hơn trong việc bắt chước và tạo ra các bức vẽ của riêng mình, giảm bớt sự frustation và khuyến khích tiếp tục luyện tập.

Bằng cách phát triển ứng dụng này, em mong muốn cung cấp một công cụ không chỉ giải trí mà còn giáo dục, giúp trẻ em phát triển kỹ năng tư duy sáng tạo, tăng cường khả năng quan sát và mô phỏng, qua đó đóng góp vào quá trình học tập và phát triển toàn diện của trẻ.

2. Mục tiêu, Đối tượng và Phạm vi Nghiên cứu

Mục tiêu nghiên cứu:

- Phân tích và hiểu rõ bài toán nghiệp vụ thực tế liên quan đến việc hỗ trợ học vẽ của trẻ nhỏ.
- Tìm hiểu và ứng dụng Framework Flutter để xây dựng ứng dụng học vẽ.
- Xây dựng một ứng dụng hỗ trợ học vẽ cho trẻ em.
- Đảm bảo giao diện người dùng dễ sử dụng và phù hợp với trẻ em từ 4-12 tuổi và phụ huynh.

Đối tượng nghiên cứu:

- Ngôn ngữ lập trình Kotlin.
- Công cụ lập trình Android Studio.
- Framework Flutter.
- Các thuật toán phân tách ảnh thành các đường nét.
- Cách sử dụng Kotlin trong lập trình Flutter.

Phạm vi nghiên cứu:

- Ứng dụng mới xây dựng này sẽ đáp ứng những chức năng chính sau:
- Phân tách ảnh thành các đường nét: Sử dụng thuật toán để phân tách các hình ảnh thành các đường nét cơ bản, giúp trẻ học vẽ theo các mẫu.
- Hỗ trợ học vẽ cho trẻ nhỏ: Cung cấp các bài học và mẫu vẽ có sẵn để trẻ em có thể thực hành và học vẽ một cách dễ dàng.
- Theo dõi tiến trình học vẽ: Cho phép phụ huynh theo dõi và hỗ trợ quá trình học vẽ của con em mình.
- Giao diện thân thiện: Đảm bảo giao diện trực quan, dễ sử dụng và phù hợp với trẻ nhỏ từ 4-12 tuổi.

- Tích hợp Kotlin và Flutter: Sử dụng Kotlin và Flutter để phát triển ứng dụng đa nền tảng, tận dụng các tính năng của cả hai công nghệ này.

3. Phương pháp nghiên cứu

- Phương pháp nghiên cứu về mặt lý thuyết
 - Nghiên cứu và tiến hành thu thập các tài liệu có liên quan đến đề tài.
 - Tổng hợp các tài liệu đã thu thập và tiến hành phân tích.
 - Chọn lọc các tài liệu nghiên cứu để báo cáo đồ án tốt nghiệp.
- Phương pháp nghiên cứu trong thực nghiệm
 - Tìm hiểu cách thức, tiêu chí kiểm thử chức năng.
 - Phân tích các ứng dụng có chức năng tương tự.
 - Phân tích và tiến hành xây dựng ứng dụng theo phân tích và yêu cầu thực tế.
 - Tìm hiểu ngôn ngữ lập trình Dart, công cụ lập trình Android Studio, framework Flutter
 - Áp dụng công nghệ vào xây dựng hệ thống.
 - Tổng hợp các kiến thức đã học và hoàn thành báo cáo.

4. Bố cục đề tài

Ngoài các phần Mở đầu, Kết luận và Tài liệu tham khảo, báo cáo đồ án được bố cục thành ba chương chính sau.

- Chương 1: Trình bày tổng quan về ứng dụng di động dạy vẽ cho trẻ nhỏ.
- Chương 2: Khảo sát và đặc tả hệ thống, thiết kế hệ thống.
- Chương 3: Cài đặt phần mềm và kết quả đạt được.

CHƯƠNG 1. TỔNG QUAN VỀ ỨNG DỤNG DI ĐỘNG DẠY VẼ CHO TRẺ NHỎ

1.1. Giới thiệu chung

Với sự tiến bộ không ngừng của công nghệ, trẻ em ngày càng bắt đầu tiếp xúc với các thiết bị điện tử từ sớm. Nhiều phụ huynh sử dụng điện thoại di động làm phương tiện giải trí cho trẻ, giúp các em có thể tự chơi trong khi họ bận rộn với công việc. Tuy nhiên, không ít ứng dụng trên thị trường hiện nay có nội dung không lành mạnh, ảnh hưởng đến sự phát triển của trẻ. Bên cạnh đó, sự phổ biến của anime và manga đã trở thành nguồn cảm hứng cho nhiều người muốn vẽ lại các nhân vật yêu thích. Thách thức ở đây là việc tái tạo các nhân vật phức tạp này thường yêu cầu kỹ năng và sự tỉ mỉ cao. Mặc dù có nhiều ứng dụng hỗ trợ vẽ trên điện thoại, nhưng để đạt được độ chính xác và chi tiết cao, người dùng thường cần đến màn hình cảm ứng lớn và các thiết bị phụ trợ như bút cảm ứng, điều này kéo theo chi phí đáng kể cho các thiết bị bổ sung này. Do đó, việc cung cấp các giải pháp giải trí lành mạnh cho trẻ và phát triển công cụ hỗ trợ vẽ hiệu quả, kinh tế hơn là hết sức cần thiết để phù hợp với tài chính của đa số gia đình, đồng thời đáp ứng nhu cầu và sở thích của những người không chuyên về mảng hội họa.

1.2. Mô tả bài toán

Bài toán đặt ra trong đề án tốt nghiệp của em là thiết kế một ứng dụng giúp phân tích hình ảnh thành các đường nét cơ bản và hiển thị chúng trên màn hình điện thoại hoặc máy tính bảng. Ứng dụng này phải đáp ứng các yêu cầu sau để phục vụ hiệu quả cho người dùng, đặc biệt là trẻ em:

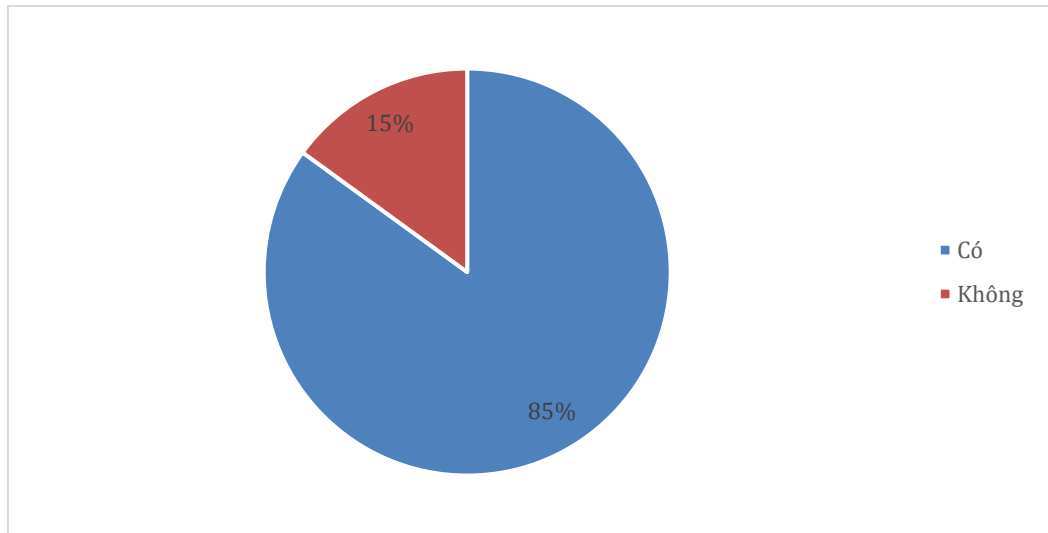
- **Đa dạng nguồn ảnh:** Ứng dụng cần có khả năng lấy ảnh từ nhiều nguồn khác nhau, bao gồm:

- Ảnh có sẵn: Cho phép người dùng chọn ảnh từ bộ sưu tập có sẵn trong thiết bị.
 - Ảnh từ thư viện: Người dùng có thể truy cập và chọn các ảnh từ thư viện ảnh của ứng dụng, nơi chứa các hình ảnh được tối ưu hóa cho việc phân tích và học vẽ.
 - Camera: Cho phép người dùng chụp ảnh mới bằng camera của thiết bị và sử dụng ảnh đó ngay lập tức.
- **Lưu trữ và quản lý hình ảnh:** Lưu giữ hình ảnh trong quá trình sử dụng để dễ dàng truy cập và phân tích lại nếu cần.
 - **In Ảnh và Ghi Lại Quá Trình Vẽ:**
 - In ảnh: Người dùng có thể in các đường nét cơ bản của hình ảnh đã phân tích, hỗ trợ quá trình vẽ trên giấy.
 - Quay video quá trình vẽ: Ứng dụng cũng cung cấp tính năng quay video quá trình vẽ, cho phép người dùng ghi lại từng bước của họ trong việc tái tạo hình ảnh, từ đó có thể xem lại để học hỏi hoặc chia sẻ với người khác.

Những yêu cầu này không chỉ giúp tạo điều kiện thuận lợi cho người dùng trong việc sử dụng ứng dụng để học vẽ mà còn đảm bảo tính linh hoạt và tiện dụng, từ đó thúc đẩy sự sáng tạo và khả năng tự học. Việc lưu trữ các hình ảnh được chọn cùng với khả năng in và ghi lại quá trình vẽ góp phần tạo ra một kho tài nguyên phong phú cho người dùng tham khảo và phát triển kỹ năng vẽ của mình qua thời gian.

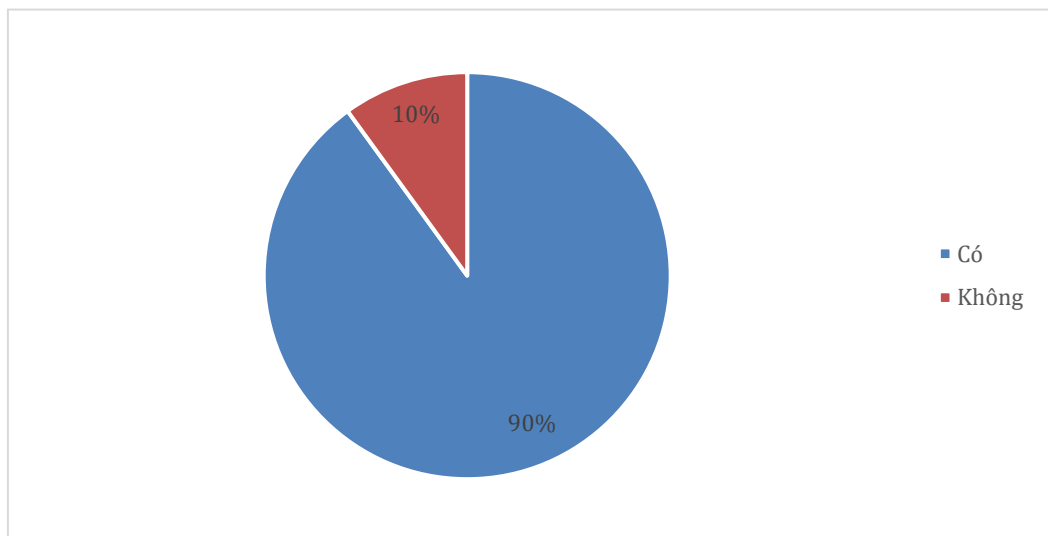
1.3. Khảo sát người dùng

- Câu 1: Bạn có cho phép con sử dụng thiết bị di động không?



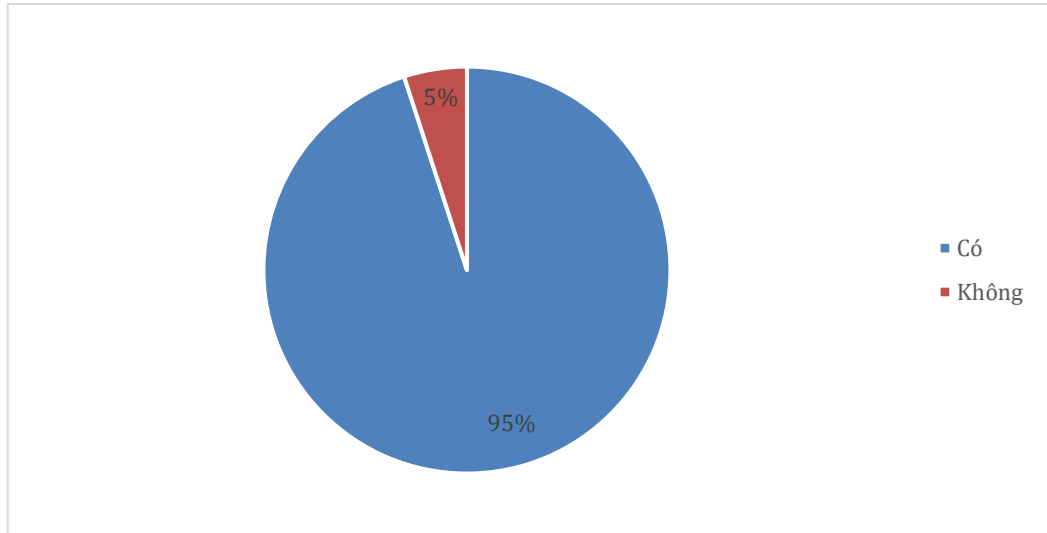
Hình 1.1 Kết quả khảo sát câu hỏi 1

Câu 2: Bạn có muốn tìm kiếm các ứng dụng giúp phát triển kỹ năng sáng tạo cho con không?



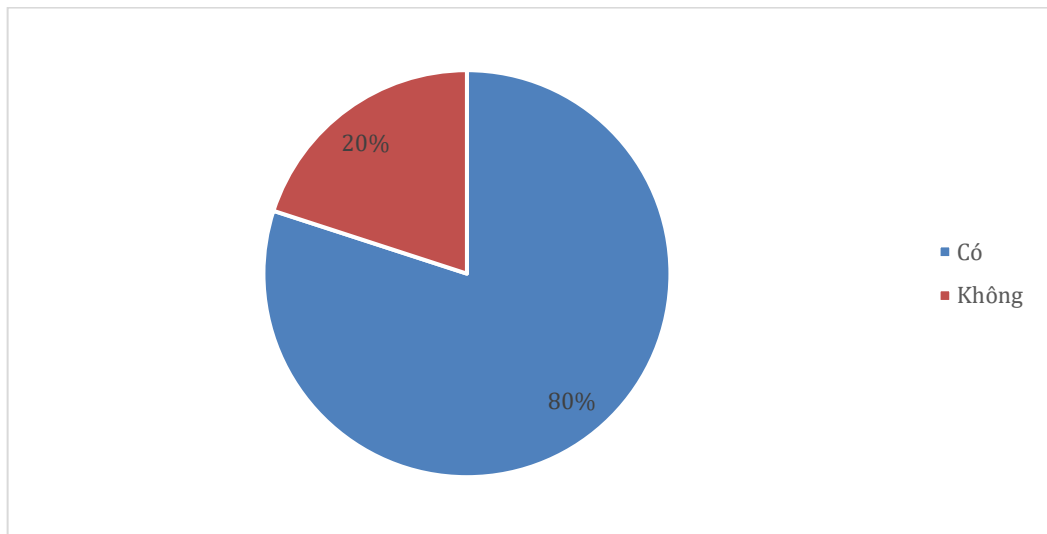
Hình 1.2 Kết quả khảo sát câu hỏi 2

- Câu 3: Bạn có muốn ứng dụng vẽ cho phép con bạn học vẽ các nhân vật hoạt hình hoặc anime yêu thích không?



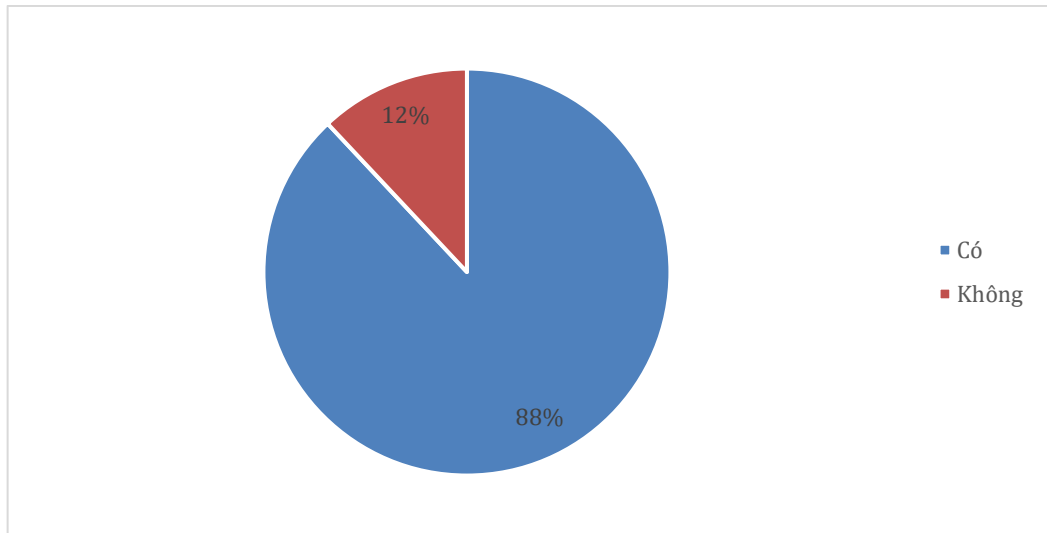
Hình 1.3 Kết quả khảo sát câu hỏi 3

- Câu 4: Bạn có tin tưởng vào ứng dụng sử dụng công nghệ xử lý ảnh để hỗ trợ con bạn học vẽ không?



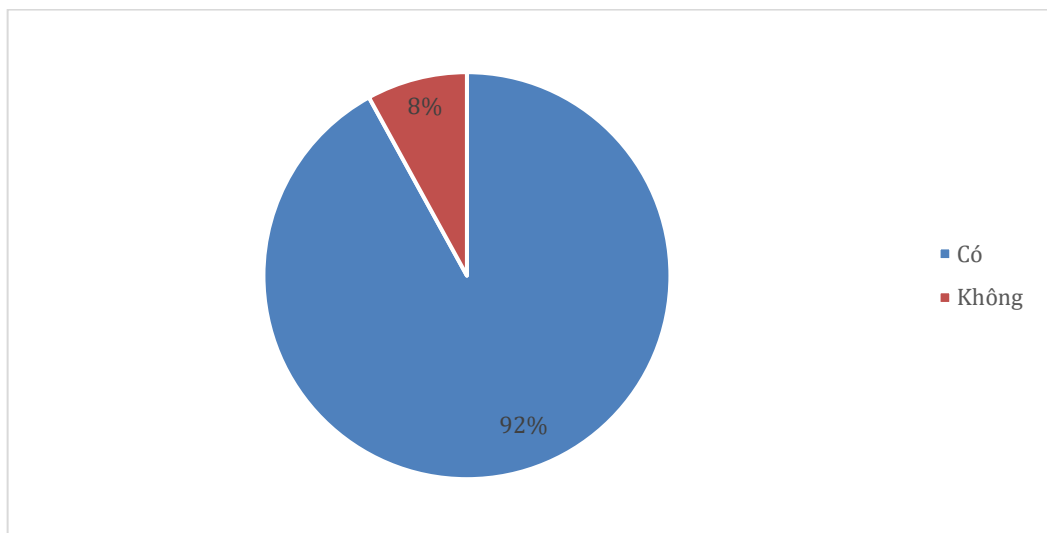
Hình 1.4 Kết quả khảo sát câu hỏi 4

- Câu 5: Bạn có muốn ứng dụng có các tính năng như lưu trữ, chia sẻ, và in ấn các bức vẽ không?



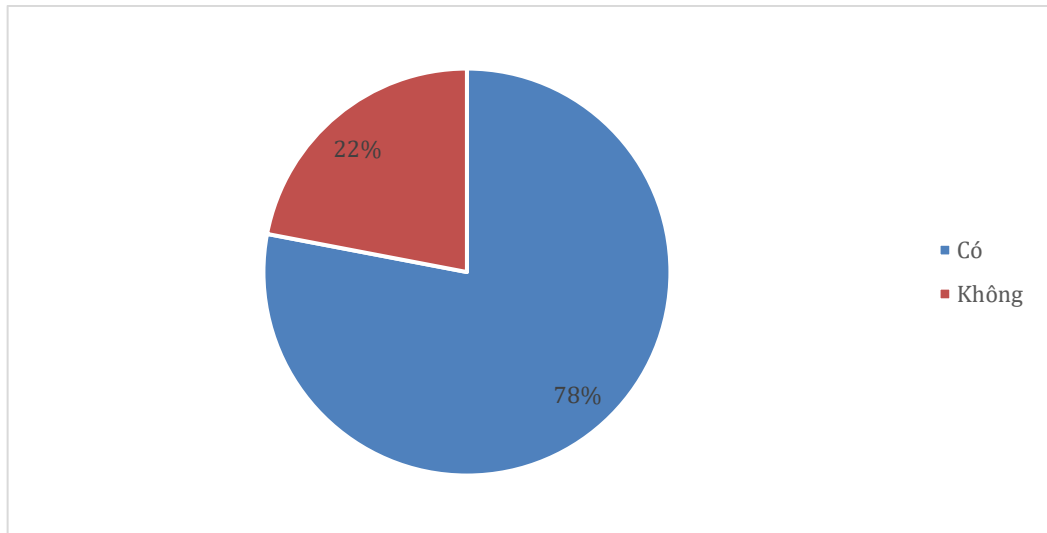
Hình 1.5 Kết quả khảo sát câu hỏi 5

- Câu 6: Bạn có muốn ứng dụng có giao diện dễ sử dụng cho trẻ nhỏ không?



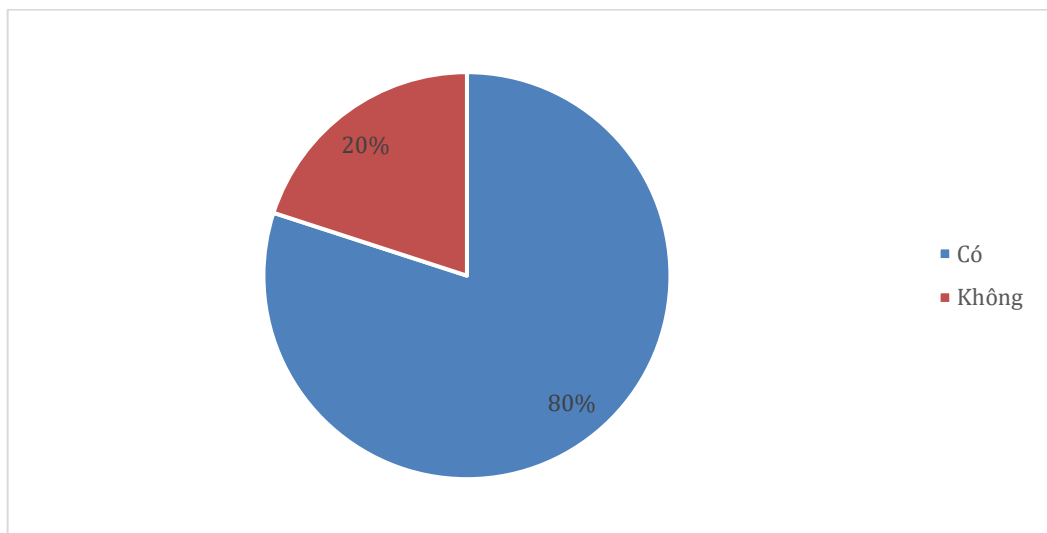
Hình 1.6 Kết quả khảo sát câu hỏi 6

- Câu 7: Bạn có lo ngại về vấn đề an toàn và nội dung phù hợp cho trẻ khi sử dụng ứng dụng di động không?



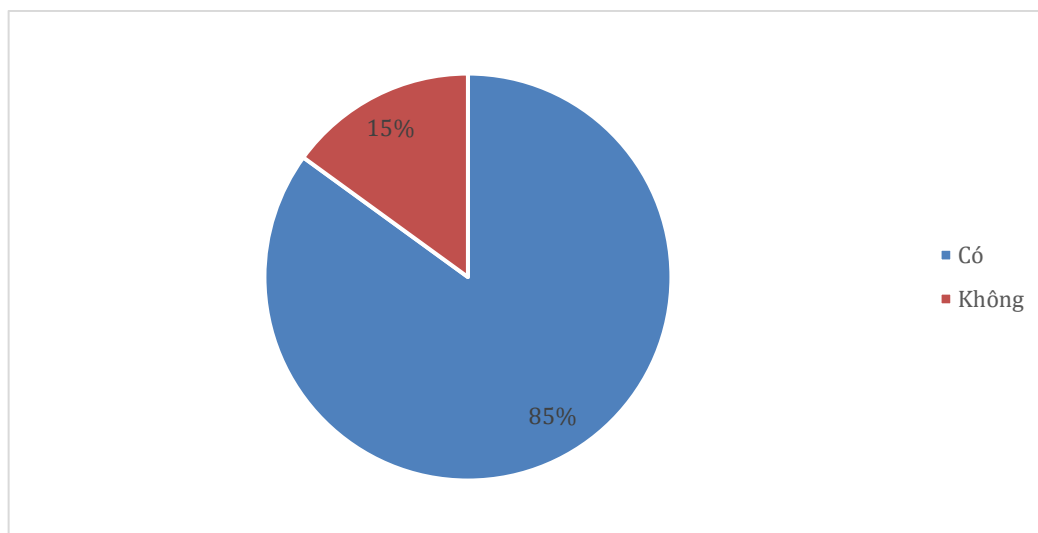
Hình 1.7 Kết quả khảo sát câu hỏi 7

- Câu 8: Con có thích sử dụng điện thoại hoặc máy tính bảng để chơi trò chơi không?



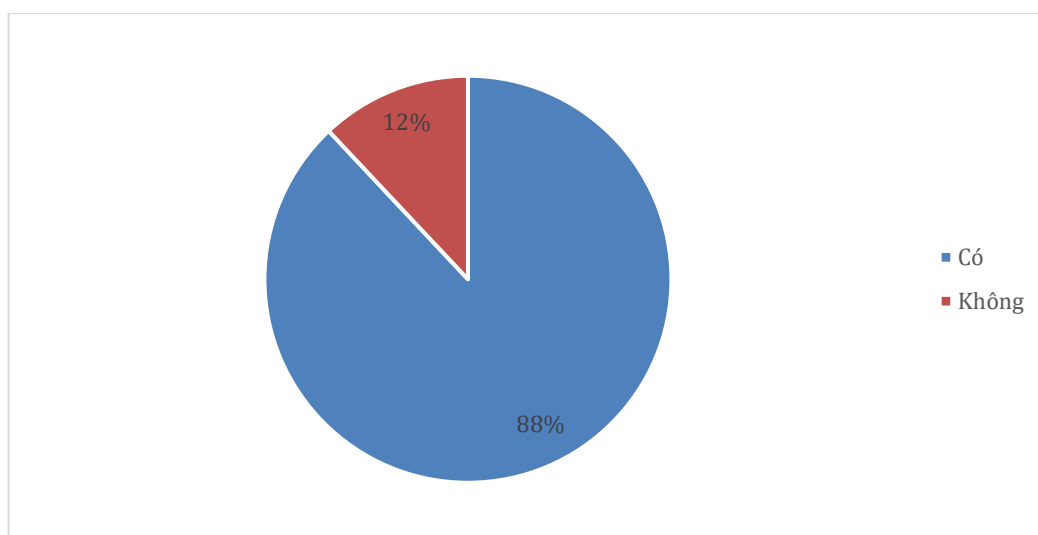
Hình 1.8 Kết quả khảo sát câu hỏi 8

- Câu 9: Con có thích vẽ các nhân vật hoạt hình hoặc anime yêu thích không?



Hình 1.9 Kết quả khảo sát câu hỏi 9

- Câu 10: Con có muốn lưu trữ và xem lại các bức vẽ của mình không?



Hình 1.10 Kết quả khảo sát câu hỏi 10

1.4. Yêu cầu chức năng

- Phân tích ảnh thành đường nét
 - Chuyển đổi ảnh được người dùng chọn thành các đường nét và hiển thị kết quả trên màn hình.

- Sử dụng ảnh từ thiết bị
 - Cho phép người dùng chụp ảnh mới hoặc chọn ảnh từ thư viện thiết bị, sau đó tải lên ứng dụng để xử lý.
- Lưu trữ ảnh yêu thích
 - Cung cấp tính năng lưu trữ ảnh yêu thích, cho phép người dùng xem lại bất cứ lúc nào.
- Quay và chụp ảnh
 - Cho phép người dùng quay video hoặc chụp lại quá trình vẽ, sau đó lưu vào thư viện của thiết bị.

1.5. Yêu cầu phi chức năng

- Giao diện đồ họa dễ hình, dễ thao tác
- Hoạt động của hệ thống ổn định
- Độ tin cậy của ứng dụng
- Hiệu năng của ứng dụng

1.6. Yêu cầu thiết bị

- Hệ điều hành: Android 7.0 (Nougat, API level 24) trở lên
- RAM: 1.5 GB
- Bộ nhớ trong: 32GB
- Màn hình: Độ phân giải tối thiểu 1280x720
- Kết nối Internet ổn định
- Hỗ trợ NDK phiên bản tương ứng với phiên bản Flutter

1.7. Cơ sở lý thuyết

1.7.1. Tổng quan về lập trình di động

Hiện nay có 3 hướng chính xây dựng và phát triển 1 ứng dụng di động là: Native app, Web App và Hybrid app. Mỗi hướng đều có ưu và nhược điểm và kỹ năng khác nhau.

1.7.1.1. Hybrid App

Hybrid App [6] là ứng dụng kết hợp của cả Mobile Web App và Native App. Theo đó, các phần cơ bản của ứng dụng vẫn viết bằng ngôn ngữ web, nhưng được đặt trong native container, nên vẫn có thể đưa lên các App Store.

Ưu điểm:

- Chỉ cần có kiến thức về HTML, CSS, JavaScript ...
- Viết một lần dùng được trên nhiều nền tảng.
- Tận dụng được các chức năng của hệ thống.
- Có thể chạy được ở chế độ offline.

Nhược điểm:

- Hiệu năng chậm hơn so với Native App.
- Không ổn định do khó khăn trong việc debug. Framework sẽ dịch code thành code native và làm khó khăn trong việc sửa lỗi.

1.7.1.2. Native App

Native App [6] là những ứng dụng được viết riêng cho một loại nền tảng như iOS, Android, Windows Phone bằng các ngôn ngữ tương ứng của mọi nền tảng đó ví dụ Java trên Android, Objective – C trên iOS.

Mỗi Native App chỉ chạy được trên một nền tảng và không thể mang sang các nền tảng khác.

Ưu điểm:

- Tận dụng được tính năng có sẵn trên thiết bị như: GPS, Camera, thiết bị thu âm ...
- Hiệu năng cao vì code native chạy trực tiếp trên thiết bị.
- Có thể chạy được ở chế độ offline.

Nhược điểm:

- Không thể kết hợp nhiều nền tảng. Mỗi một ứng dụng chỉ chạy trên một nền tảng nhất định.
- Mỗi hệ điều hành cần phải viết ứng dụng riêng khó đồng bộ giữa các ứng dụng.
- Việc bảo trì hay nâng cấp sẽ làm mất nhiều thời gian. Do phải sửa chữa từng app trên từng hệ điều hành.
- Xây dựng ứng dụng cần dùng các phần mềm riêng biệt theo hệ điều hành (dùng Xcode trên Mac để phát triển ứng dụng IOS, Android Studio để phát triển ứng dụng Android).

1.7.1.3. Web App

Web App [6] được phát triển dựa trên nền tảng của một website đã có sẵn. Một trang web riêng biệt dành cho các thiết bị di động sẽ được tạo ra, sử dụng HTML, CSS và các thư viện hỗ trợ khác.

Web App được thiết kế để hoạt động trên các trình duyệt web hoặc trình duyệt di động, cho phép người dùng thao tác tương tự như khi sử dụng một ứng dụng di động.

Ưu điểm:

- **Trải nghiệm người dùng tốt:** Web app không cần cài đặt hoặc yêu cầu nâng cấp khi sử dụng.
- **Dung lượng lưu trữ không giới hạn:** Không chiếm không gian bộ nhớ

máy tính. Với sự phổ biến của Cloud. Không gian lưu trữ hầu như là vô hạn.

- **Truy cập linh hoạt:** Nhiều người dùng có thể truy cập vào mọi lúc, mọi nơi trên nhiều nền tảng, trình duyệt và thiết bị khác nhau.
- **Bảo mật và an toàn:** Đăng nhập bằng tài khoản được bảo mật. Dữ liệu được lưu trữ trên cloud server, có thể truy cập từ bất kỳ đâu hoặc tải về thiết bị cá nhân.
- **Tiết kiệm chi phí:** Hầu hết các Web app đều miễn phí hoàn toàn hoặc một phần hoặc nếu có phải trả một khoản phí nhỏ. Tiết kiệm chi phí bản quyền không nhỏ cho người sử dụng, đặc biệt là doanh nghiệp.
- **Dễ dàng đăng ký và sử dụng:** Chỉ cần vài phút để đăng ký, thanh toán và sử dụng. Hạn chế tình trạng sao chép và vi phạm bản quyền.
- **Luôn được cập nhật:** Vì mọi người đều truy cập cùng một phiên bản của web app thông qua một URL. Họ sẽ luôn truy cập vào phiên bản cập nhật của phần mềm.

Nhược điểm:

- Hiệu năng không được tốt như Native App và luôn phải chạy online.
- Không thể dùng được các tính năng tích hợp của di động: Notification, chụp hình, nghiêng máy, định vị GPS, các sensor...
- Với một số máy đời cũ, Web app sẽ bị vỡ giao diện, hiển thị sai, hoặc javascript không chạy.

1.7.2. Công cụ lập trình Android Studio

1.7.2.1. Giới thiệu Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức dành cho phát triển nền tảng Android, được phát hành miễn phí theo giấy phép Apache License 2.0.

Android Studio được đóng góp với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

Android Studio cung cấp nhiều tính năng nâng cao hiệu suất khi xây dựng ứng dụng Android như:

- Một hệ thống xây dựng Gradle linh hoạt.
- Trình mô phỏng nhanh và tính năng phong phú.
- Instant Run để đẩy các thay đổi vào ứng dụng đang chạy của bạn mà không cần xây dựng một bản cài APK mới.
- Tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến và nhập mã mẫu.
- Hỗ trợ tích hợp sâu Firebase vào trong các ứng dụng.
- Hỗ trợ nhiều ngôn ngữ như: Java, Kotlin, Dart, C, C++, ...

1.7.2.2. Sơ lược lịch sử Android Studio

Android Studio đã được công bố vào năm 2013 tại hội nghị Google I/O và được phát hành vào năm 2014 sau nhiều phiên bản khác nhau.

1.7.2.3. Ứng dụng của Android Studio

- Debug và cài đặt ứng dụng trên máy có hệ điều hành Android thông qua USB Debug hoặc wifi.
- Generate file APK để cài đặt trên điện thoại hoặc bất kì một trình giả lập Android nào.
- Sử dụng một project Empty để phát triển ứng dụng qua việc xây dựng layout bằng các file .xml và xử lý logic bằng các ngôn ngữ: Java, Kotlin hoặc Dart.

1.7.3. Ngôn ngữ lập trình Dart

Dart là một ngôn ngữ lập trình được phát triển bởi Google vào năm 2011, đặc biệt được sử dụng để phát triển ứng dụng di động và web. Dart được thiết kế để cải thiện các vấn đề của JavaScript và cung cấp nhiều tính năng hiện đại hơn. Dart hỗ trợ kiểu tĩnh và kiểu động, cú pháp đơn giản và dễ đọc, hỗ trợ lập trình hướng đối tượng, hàm bậc cao, và bộ thu gom rác tự động.

Dart cũng hỗ trợ viết code đa nền tảng, cho phép phát triển ứng dụng di động cho cả Android và iOS, cũng như ứng dụng web.

Ưu điểm:

- **Tích hợp tốt với Flutter:** Dart là ngôn ngữ chính của Flutter - một framework phổ biến cho việc phát triển ứng dụng di động đa nền tảng.
- **Hiệu suất cao:** Dart được tối ưu hóa để có hiệu suất tốt. Nhờ có hệ thống quản lý bộ nhớ tự động (garbage collection) và việc biên dịch mã nguồn thành mã máy, các ứng dụng Dart thường có thời gian phản hồi nhanh và tiêu thụ ít tài nguyên.
- **Kiểu dữ liệu tĩnh:** Dart hỗ trợ kiểu dữ liệu tĩnh, giúp phát hiện lỗi trước khi chương trình chạy. Điều này giúp giảm thiểu các lỗi phát sinh trong quá trình thực thi.
- **Cú pháp gần giống Java và JavaScript:** Dart có cú pháp gần giống java và JavaScript, giúp cho việc chuyển đổi giữa chúng trở nên dễ dàng hơn.

1.7.4. Một số framework trong lập trình Flutter

1.7.4.1. Flutter SDK

Flutter SDK [9] là một SDK hỗ trợ phát triển ứng dụng di động đa nền tảng sử dụng Flutter, cung cấp các tính năng như widget, animation, routing, và hỗ trợ viết ứng dụng đa nền tảng cho cả Android và iOS.

1.7.4.2. AngularDart

AngularDart [9] là một framework phát triển ứng dụng web, sử dụng Dart làm ngôn ngữ lập trình chính. AngularDart cung cấp các tính năng như mô hình MVC, dependency injection, routing, và có thể tích hợp với các thư viện khác để phát triển ứng dụng web phong phú.

1.7.4.3. StageXL

StageXL [9] là một framework phát triển trò chơi web, sử dụng Dart làm ngôn ngữ lập trình chính. StageXL cung cấp các tính năng như sprite, animation, sound, và hỗ trợ viết trò chơi web đa nền tảng.

1.7.5. Ngôn ngữ lập trình Kotlin

1.7.5.1. Giới thiệu về Kotlin

Kotlin là một ngôn ngữ lập trình mã nguồn mở được tài trợ và phát triển bởi JetBrains, được thiết kế dựa trên các thư viện Java có sẵn vì thế Kotlin hoàn toàn có thể chạy trên JVM (Java Virtual Machine) và có thể biên dịch sang mã Java cũng như sử dụng tất cả các thư viện mà Java có.

Kể từ phiên bản Android 3.0, Kotlin đã được Google hỗ trợ đầy đủ trong việc phát triển ứng dụng Android và nhúng trực tiếp vào gói cài đặt của IDE để thay thế cho trình biên dịch Java.

1.7.5.2. Sơ lược lịch sử của Kotlin

Kotlin là ngôn ngữ lập trình mã nguồn mở do JetBrains phát triển, chạy trên JVM và có thể biên dịch sang mã Java. Kotlin được Google hỗ trợ cho phát triển ứng dụng Android từ phiên bản Android Studio 3.0.

Ưu điểm của Kotlin

- Cấu trúc code ngắn gọn đơn giản, giảm được khá nhiều base code
- Dễ tiếp cận với người mới bắt đầu.

- Có thể biên dịch thành Java hay JavaScript bytecode.
- Cho phép sử dụng các thư viện của Java thông qua các annotation.
- Là mã nguồn mở nên không tốn kém gì khi sử dụng.
- Null-safety giúp tránh khỏi các đoạn code có thể null gây ra sự cố trong việc thực thi.

1.7.5.3. Ứng dụng của Kotlin

Hiện nay Kotlin được sử dụng rộng rãi trong cộng đồng Android và trở thành ngôn ngữ số 1 được hỗ trợ trên nền tảng này nhờ tính ngắn gọn và rõ ràng hơn Java. Tuy nhiên, nó không thay thế Java trong quá trình xây dựng ứng dụng bởi Kotlin cần sự hỗ trợ của Java để cài đặt ứng dụng trên bất cứ thiết bị nào.

1.7.6. SQLite Database

SQLite là hệ quản trị cơ sở dữ liệu quan hệ nhỏ gọn, mã nguồn mở, viết bằng C, tích hợp vào ứng dụng, hỗ trợ nhiều ngôn ngữ lập trình như C/C++, Python, Java, PHP.

SQLite được tích hợp sẵn trên hầu hết các hệ điều hành phổ biến như Windows, macOS và Linux, và cũng được hỗ trợ trên các nền tảng di động như Android và iOS.

Cấu trúc của SQLite

SQLite là một thư viện mã nguồn mở và được viết bằng ngôn ngữ C. Nó bao gồm một bộ máy truy vấn SQL, một bộ máy quản lý tệp và các thành phần bổ sung khác. Các đối tượng trong SQLite bao gồm bảng, cột, khóa chính và các ràng buộc.

- **Bảng:** Là một đối tượng trong SQLite để lưu trữ dữ liệu. Bảng bao gồm các cột và hàng, trong đó mỗi cột đại diện cho một kiểu dữ liệu và mỗi hàng đại diện cho một bản ghi.

- **Cột:** Là một đối tượng trong SQLite để đại diện cho một kiểu dữ liệu trong bảng. Các kiểu dữ liệu có thể là INTEGER, REAL, TEXT hoặc BLOB.
- **Khóa chính:** Là một cột hoặc tập hợp các cột trong bảng, được sử dụng để định danh duy nhất cho mỗi hàng trong bảng.
- **Ràng buộc:** Là các quy tắc được áp dụng trên bảng để đảm bảo tính toàn vẹn của dữ liệu. Ví dụ, ràng buộc NOT NULL sẽ đảm bảo rằng giá trị của cột không được phép là null.

Cách thức hoạt động của SQLite

SQLite hoạt động như một thư viện được tích hợp vào các ứng dụng. Khi một ứng dụng sử dụng SQLite, nó sẽ tạo ra một tệp tin cơ sở dữ liệu, trong đó sẽ lưu trữ các bảng, cột và các bản ghi. Ứng dụng có thể truy vấn và thay đổi dữ liệu trong cơ sở dữ liệu này bằng cách sử dụng các lệnh SQL.

SQLite hỗ trợ các lệnh SQL như SELECT, INSERT, UPDATE và DELETE để truy vấn và thay đổi dữ liệu trong cơ sở dữ liệu. Nó cũng hỗ trợ các câu lệnh để tạo bảng, chỉnh sửa bảng và xóa bảng. SQLite cũng có thể thực hiện các hoạt động như tạo chỉ mục, ràng buộc và trigger.

Với việc sử dụng cơ sở dữ liệu SQLite, chúng ta có thể lưu trữ thông tin và dữ liệu của người dùng một cách hiệu quả và tiện lợi.

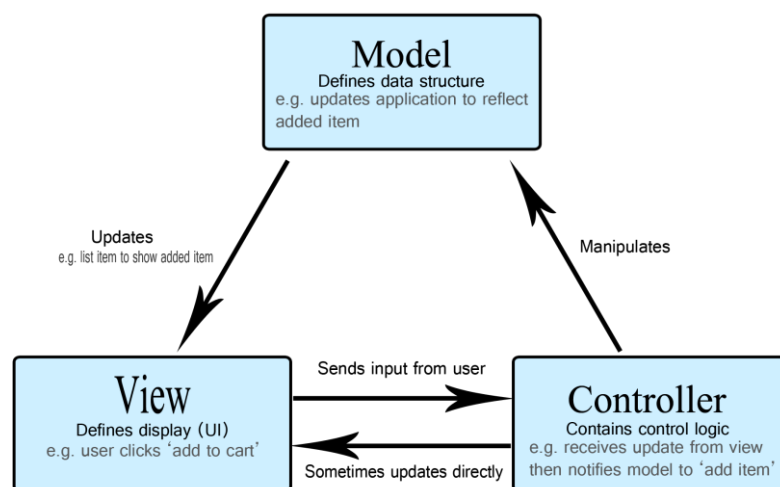
Ưu điểm của SQLite Database:

- **Kích thước nhỏ gọn:** SQLite có dung lượng nhỏ gọn và khả năng tích hợp cao, giúp dễ dàng tích hợp vào các ứng dụng.
- **Không cần server:** SQLite không yêu cầu một server đặc biệt để chạy, vì vậy nó rất tiện lợi cho các ứng dụng đơn giản.
- **Tốc độ truy vấn nhanh:** SQLite có tốc độ truy vấn nhanh và khả năng xử lý dữ liệu tốt.

- Hỗ trợ nhiều ngôn ngữ lập trình: SQLite được hỗ trợ bởi nhiều ngôn ngữ lập trình như C/C++, Python, Java, PHP, v.v.

1.7.7. Tổng quan về mô hình MVC

Mô hình MVC (Model-View-Controller) [8] là một mô hình thiết kế phần mềm được sử dụng để phát triển các ứng dụng có giao diện người dùng. Mô hình này bao gồm ba thành phần chính: Model, View và Controller.



Hình 1.11: Mô hình MVC

Model

Model là thành phần chịu trách nhiệm xử lý dữ liệu. Nó đại diện cho các đối tượng dữ liệu và quy trình xử lý dữ liệu. Model không biết gì về giao diện người dùng, nó chỉ tập trung vào việc xử lý dữ liệu.

View

View là thành phần chịu trách nhiệm hiển thị dữ liệu và tương tác với người dùng. View đại diện cho giao diện người dùng, bao gồm các thành phần như các nút bấm, trường nhập liệu và các thành phần khác. View không biết gì về dữ liệu và quy trình xử lý, nó chỉ tập trung vào việc hiển thị và tương tác với người dùng.

Controller

Controller là thành phần chịu trách nhiệm điều khiển quá trình tương tác giữa Model và View. Controller đóng vai trò là trung gian giữa Model và View, nó nhận lệnh từ View, xử lý dữ liệu từ Model và cập nhật lại View để hiển thị kết quả cho người dùng. Controller là thành phần duy nhất có thể tương tác với cả Model và View.

Mô hình MVC giúp tách biệt logic xử lý dữ liệu và giao diện người dùng, giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn. Nó cũng giúp cho việc thay đổi giao diện người dùng hoặc cơ sở dữ liệu không ảnh hưởng đến logic xử lý dữ liệu và ngược lại.

Một ứng dụng thực tế có thể được thiết kế theo mô hình MVC

- **Model:** Chứa các đối tượng dữ liệu như thông tin người dùng, thông tin sản phẩm, danh sách đơn hàng, v.v. Nó cũng chứa các phương thức để truy xuất và xử lý dữ liệu.
- **View:** Chứa các màn hình giao diện người dùng để hiển thị thông tin và tương tác với người dùng, ví dụ như màn hình đăng nhập, màn hình danh sách sản phẩm, màn hình đặt hàng, v.v.
- **Controller:** Chứa các phương thức để điều khiển quá trình tương tác giữa Model và View, ví dụ như phương thức để xử lý đăng nhập, phương thức để lấy danh sách sản phẩm, phương thức để đặt hàng, v.v.

Với việc sử dụng mô hình MVC, việc phát triển và bảo trì ứng dụng sẽ trở nên dễ dàng hơn và có thể được thực hiện bởi các nhóm phát triển khác nhau độc lập với nhau.

Ưu điểm

- **Tách biệt logic:** MVC giúp tách biệt logic ứng dụng thành các phần riêng biệt (Model, View, và Controller), làm cho mã nguồn dễ hiểu và dễ bảo trì.

- **Tái sử dụng mã nguồn:** Các thành phần trong MVC có thể được tái sử dụng một cách linh hoạt. Model và View có thể được sử dụng lại trong nhiều phần của ứng dụng hoặc trong các ứng dụng khác nhau.
- **Phân chia công việc:** Mô hình này giúp phân chia công việc giữa các thành viên trong nhóm phát triển, với người làm Model tập trung vào logic xử lý dữ liệu, người làm View chịu trách nhiệm về giao diện người dùng, và người làm Controller quản lý sự tương tác.
- **Dễ mở rộng:** Do sự tách biệt giữa các thành phần, việc mở rộng và thay đổi ứng dụng trở nên dễ dàng hơn. Một thành phần có thể được thay thế hoặc mở rộng mà không làm ảnh hưởng đến các thành phần khác.
- **Kiểm thử dễ dàng:** Sự tách biệt giữa Model, View và Controller giúp kiểm thử tốt hơn, vì bạn có thể kiểm thử mỗi thành phần một cách độc lập, giảm rủi ro gây ảnh hưởng lẫn nhau.

Nhược điểm

- **Phức tạp cho ứng dụng nhỏ:** Đối với các ứng dụng nhỏ hoặc đơn giản, việc triển khai mô hình MVC có thể quá phức tạp và làm tăng độ phức tạp của mã nguồn.
- **Tăng độ phức tạp của mã nguồn:** Trong một số trường hợp, việc chia nhỏ ứng dụng thành nhiều thành phần có thể làm tăng độ phức tạp của mã nguồn và làm cho mã trở nên khó hiểu.
- **Khó khăn trong việc đảm bảo sự đồng bộ:** Đôi khi có thể khó khăn để đảm bảo sự đồng bộ giữa Model và View, đặc biệt là trong các ứng dụng lớn và phức tạp.
- **Thừa phần:** Có khả năng xuất hiện thừa phần trong quá trình xử lý, nơi một số logic xử lý có thể được triển khai cả ở Controller và Model, dẫn đến sự phức tạp không cần thiết.

1.7.8. OpenCV và Thuật toán Xử lý Ảnh

OpenCV (Open-Source Computer Vision Library) [4] là một thư viện phần mềm rất phổ biến, được sử dụng rộng rãi trong các ứng dụng thị giác máy tính để xử lý hình ảnh và video. Thư viện này cung cấp các công cụ hữu ích cho việc phát hiện và nhận dạng đối tượng, theo dõi chuyển động, xử lý hình ảnh nâng cao và nhiều chức năng khác.

Các thuật toán xử lý ảnh cơ bản trong OpenCV bao gồm:

- **Phát hiện Cạnh (Edge Detection):** Sử dụng các thuật toán như Sobel, Canny để xác định các cạnh của đối tượng trong hình ảnh.
- **Chuyển đổi Không gian Màu (Color Space Transformation):** Chuyển đổi hình ảnh từ không gian màu RGB sang grayscale hoặc các không gian màu khác để dễ dàng phân tích và xử lý.
- **Làm mịn và Lọc Ảnh (Smoothing and Filtering):** Áp dụng các bộ lọc Gaussian, median để làm mịn hình ảnh, giảm nhiễu và cải thiện chất lượng hình ảnh trước khi phân tích.
- **Phân đoạn Hình Ảnh (Image Segmentation):** Tách biệt các đối tượng trong ảnh, giúp đơn giản hóa quá trình phân tích bằng cách làm nổi bật các đối tượng quan trọng.

Việc sử dụng các thuật toán này trong ứng dụng không chỉ giúp tự động hóa quá trình phân tích hình ảnh mà còn nâng cao độ chính xác và hiệu quả của việc học vẽ, đặc biệt là cho trẻ em và những người mới bắt đầu.

1.7.8.1. Giới thiệu các thuật toán sử dụng

Gaussian Blur

- **Mục đích:** Gaussian Blur được sử dụng để làm mờ ảnh nhằm giảm nhiễu và làm mượt các cạnh trong ảnh, giúp chuẩn bị ảnh cho các bước xử lý tiếp theo.

- **Nguyên lý hoạt động:** Gaussian Blur [5] áp dụng một bộ lọc Gaussian, mà bản chất là một phép tính trung bình trọng số của các điểm ảnh xung quanh mỗi điểm ảnh trong ảnh gốc. Trọng số giảm dần theo khoảng cách từ điểm trung tâm. Điều này có nghĩa là các điểm ảnh càng gần trung tâm của kernel càng có ảnh hưởng lớn đến giá trị trung bình. Kết quả là ảnh trở nên mờ hơn, loại bỏ các chi tiết nhỏ và nhiễu, giúp các bước xử lý sau như phát hiện cạnh trở nên chính xác hơn.

Adaptive Threshold

- **Mục đích:** Adaptive Thresholding được sử dụng để chuyển đổi ảnh từ grayscale (mức xám) sang ảnh nhị phân (đen trắng). Mục tiêu là phân tách rõ ràng các đối tượng trong ảnh bằng cách đặt ngưỡng dựa trên điều kiện ánh sáng cục bộ.
- **Nguyên lý hoạt động:** Adaptive Thresholding [11] chia ảnh thành nhiều vùng nhỏ và áp dụng ngưỡng khác nhau cho từng vùng dựa trên điều kiện ánh sáng cục bộ của vùng đó. Thay vì áp dụng một ngưỡng cố định cho toàn bộ ảnh, phương pháp này điều chỉnh ngưỡng dựa trên giá trị trung bình hoặc giá trị trung bình có trọng số của các điểm ảnh xung quanh. Điều này giúp tạo ra ảnh nhị phân với các đối tượng được phân tách rõ ràng, ngay cả khi ảnh có sự thay đổi ánh sáng lớn.

Canny Edge Detection

- **Mục đích:** Thuật toán Canny Edge Detection được sử dụng để phát hiện các cạnh trong ảnh. Cạnh là những đường viền xác định ranh giới của các đối tượng trong ảnh.
- **Nguyên lý hoạt động:** Canny Edge Detection [5] là một quy trình gồm nhiều bước:
 - **Làm mờ ảnh:** Ảnh được làm mờ bằng Gaussian Blur để giảm nhiễu.

- **Tính gradient:** Gradient của ảnh được tính toán để xác định cường độ và hướng của các cạnh.
- **Non-maximum suppression:** Loại bỏ các điểm không phải là cực đại để làm mỏng các cạnh.
- **Double thresholding:** Áp dụng hai ngưỡng để xác định các điểm mạnh và yếu của cạnh.
- **Edge tracking by hysteresis:** Kết nối các điểm cạnh mạnh và yếu để tạo thành các cạnh liên tục.

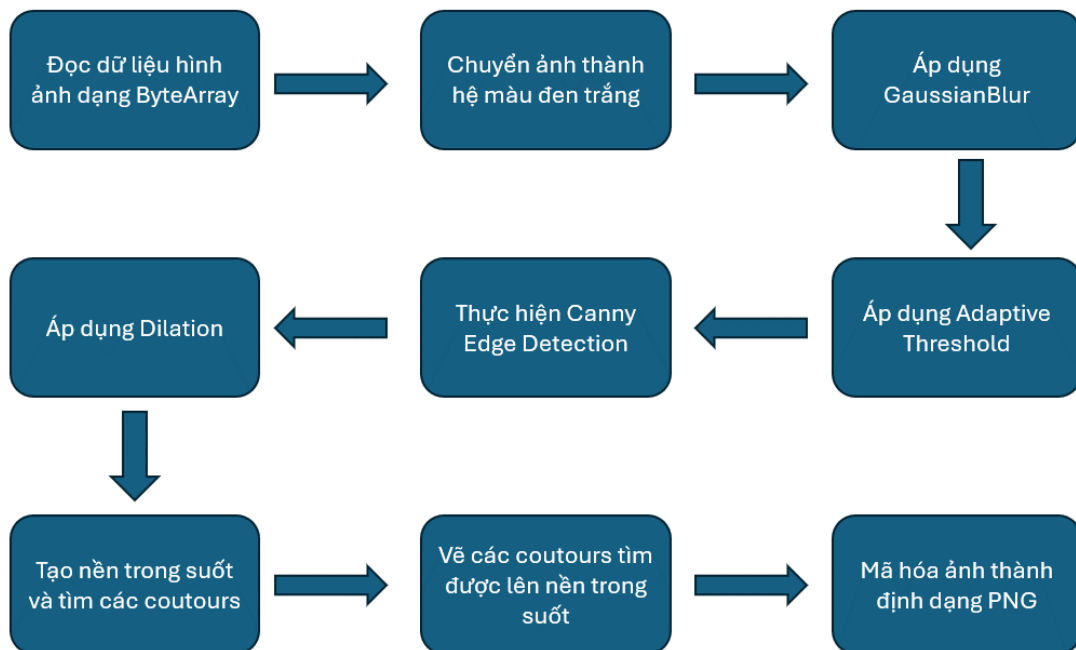
Dilation

- **Mục đích:** Dilation được sử dụng để làm nổi bật và làm dày các cạnh đã phát hiện, giúp làm rõ các đường viền và giảm nhiễu.
- **Nguyên lý hoạt động:** Dilation [5] áp dụng một kernel (cấu trúc phần tử) lên ảnh nhị phân. Kernel này di chuyển qua từng điểm ảnh của ảnh gốc và thay thế giá trị của điểm ảnh hiện tại bằng giá trị cực đại của các điểm ảnh xung quanh trong vùng kernel. Kết quả là các điểm ảnh trắng (foreground) được mở rộng, làm dày các đường viền và giúp các cạnh trở nên rõ ràng hơn.

Contours Detection

- **Mục đích:** Tìm và vẽ các đường viền bao quanh các đối tượng trong ảnh.
- **Nguyên lý hoạt động:** Contours Detection [12] tìm các đường biên của các vùng liên thông trong ảnh nhị phân. Các contours được xác định là các đường bao quanh các vùng có giá trị pixel tương tự và được sử dụng để vẽ các đường viền của các đối tượng. Phương pháp này giúp xác định các đối tượng trong ảnh và làm nổi bật chúng.

1.7.8.2. Các bước trong quy trình xử lý ảnh



Hình 1.12 Các bước của quy trình xử lý ảnh

Bước 1. Đọc ảnh từ ByteArray:

Ảnh được đọc từ dữ liệu đầu vào từ định dạng ByteArray để bắt đầu quy trình chuyển đổi

Bước 2. Chuyển ảnh thành hệ màu đen trắng:

Ảnh sẽ được chuyển đổi sang định dạng grayscale. Việc này giúp giảm bớt sự phức tạp trong xử lý ảnh bằng cách chỉ làm việc với một kênh màu duy nhất.

Bước 3. Áp dụng Gaussian Blur:

Gaussian Blur được áp dụng để làm mờ ảnh, giúp giảm nhiễu và làm mượt các cạnh trong ảnh. Bộ lọc Gaussian tính giá trị trung bình trọng số của các điểm ảnh xung quanh mỗi điểm ảnh, với trọng số giảm dần theo khoảng cách từ điểm trung tâm.

Bước 4. Áp dụng Adaptive Threshold:

Adaptive Thresholding chuyển đổi ảnh grayscale đã làm mờ sang ảnh nhị phân. Phương pháp này tự động điều chỉnh ngưỡng cho từng phần của ảnh dựa trên độ sáng cục bộ, giúp phân tách rõ ràng các đối tượng ngay cả khi có sự thay đổi ánh sáng lớn.

Bước 5. Phát hiện cạnh bằng thuật toán Canny:

Thuật toán Canny Edge Detection được áp dụng để phát hiện các cạnh trong ảnh nhị phân. Quy trình này bao gồm làm mờ ảnh, tính gradient, áp dụng non-maximum suppression, sử dụng double thresholding, và kết nối các điểm cạnh bằng edge tracking by hysteresis.

Bước 6. Dilation (Giãn nở):

Dilation được áp dụng để làm nổi bật và làm dày các cạnh đã phát hiện. Phép toán này mở rộng các điểm ảnh trắng bằng cách thêm các điểm ảnh lân cận vào vùng foreground, làm rõ các cạnh và giảm nhiễu.

Bước 7. Tạo nền trong suốt và tìm contours:

Một hình ảnh với nền trong suốt được khởi tạo để chứa các contours. Các contours của các đối tượng trong ảnh được tìm kiếm và xác định bằng cách tìm các đường biên của các vùng liên thông.

Bước 8. Vẽ contours lên nền trong suốt:

Các contours được vẽ lên hình ảnh với nền trong suốt. Điều này tạo ra một hình ảnh mới chỉ chứa các đường viền của các đối tượng, giúp làm nổi bật các đối tượng trên nền trong suốt.

Bước 9. Mã hóa ảnh thành định dạng PNG:

Hình ảnh đã xử lý được mã hóa thành định dạng PNG để hỗ trợ nền trong suốt. Định dạng PNG được lựa chọn vì nó hỗ trợ độ trong suốt, cho phép lưu trữ các contours trên nền trong suốt một cách hiệu quả.



Hình 1.13 Hình ảnh đầu vào



Hình 1.14 Hình ảnh kết quả

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU PHẦN MỀM

2.1. Mô hình hóa chức năng

2.1.1. Các tác nhân của hệ thống

Người dùng: là những người truy cập vào ứng dụng để thực hiện các chức năng của hệ thống.

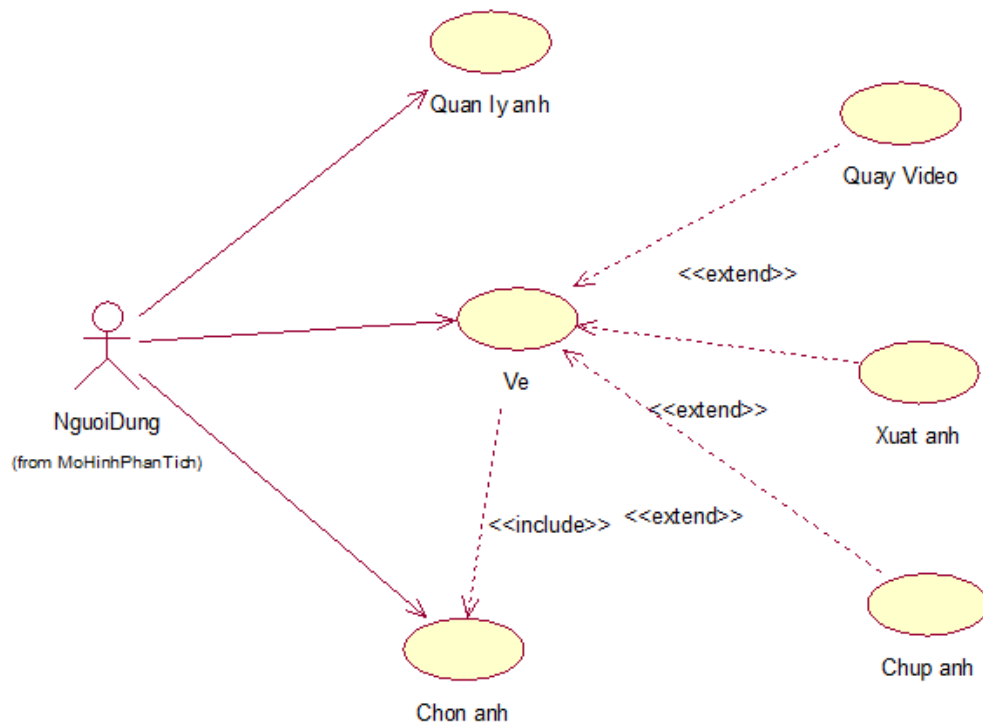
2.1.2. Các use case của hệ thống.

Bảng 2.1 Bảng các use case của hệ thống.

Tác nhân	Các ca sử dụng
Người dùng	Chọn ảnh
	Vẽ Ảnh
	Quay video
	Chụp ảnh
	Xuất ảnh
	Quản lý ảnh

2.1.3. Biểu đồ use case.

2.1.3.1. Biểu đồ use case chính.



Hình 2.1 Biểu đồ use case chính.

2.1.4. Đặc tả Usecase

2.1.4.1. Use case Chọn ảnh.

Bảng 2.2 Mô tả use case “Chọn ảnh”

Mô tả	Use case này cho phép chọn ảnh để vẽ từ ảnh trong ứng dụng, ảnh trong thư viện, ảnh chụp từ camera sau đó hiện thị lên màn hình.
Tác nhân	Người dùng

Tiền điều kiện		Người dùng đã truy cập vào ứng dụng
Luồng sự kiện	Luồng sự kiện chính	<p>Use case bắt đầu khi người dùng mở ứng dụng. Hệ thống hiển thị cảnh có sẵn và các lựa chọn “Camera”, “Photo”, “My Album” lên màn hình.</p> <ol style="list-style-type: none"> 1. Chọn ảnh có sẵn: <ul style="list-style-type: none"> - Người dùng chọn chủ đề mình mong muốn để lựa chọn ảnh. - Người dùng chọn ảnh hiển thị trên màn hình hoặc bấm “See all” để hiển thị thêm ảnh thuộc chủ đề đó sau đó chọn ảnh. Hệ thống nhận ảnh và hiện thị ảnh lên cho người dùng 2. Chọn ảnh từ Camera: <ul style="list-style-type: none"> - Người dùng kích vào nút “Camera” trên màn hình, hệ thống sẽ mở camera của thiết bị. - Người dùng chụp ảnh và nhấn nút lưu. Hệ thống lưu ảnh vào bảng ANHUNGDUNG và hiển thị hình ảnh lên cho người dùng. 3. Chọn ảnh từ thư viện <ul style="list-style-type: none"> - Người dùng kích vào nút “Photo”. Hệ thống mở trình quản lý ảnh của thiết bị.

		<ul style="list-style-type: none"> - Người dùng chọn ảnh từ thư viện ảnh. Hệ thống lấy ảnh trong thiết bị và và hiển thị hình ảnh lên cho người dùng <p>4. Chọn ảnh người dùng đã lưu</p> <ul style="list-style-type: none"> - Người dùng kích vào nút “My album”. Hệ thống lấy danh sách ảnh từ bảng ANHUNGDUNG và hiển thị lên màn hình. - Người dùng kích vào ảnh cần chọn. Hệ thống tải và và hiển thị hình ảnh lên cho người dùng.
	Luồng rẽ nhánh	<ul style="list-style-type: none"> - Tại bất kỳ bước nào trong luồng cơ bản nếu người dùng nhấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn hình trước đó.
Hậu điều kiện		Không có

2.1.4.2. Usecase Vẽ ảnh

Bảng 2.3 Mô tả use case Vẽ ảnh

Mô tả	Use case này cho phép người dùng vẽ dựa trên ảnh đã chọn.
Tác nhân	Người dùng
Tiền điều kiện	Người dùng đã truy cập vào ứng dụng

Luồng sự kiện	Luồng sự kiện chính	<p>Use case bắt đầu khi người dùng kích nút “Continue” ở màn Xem Ảnh. Hệ thống sẽ phân tích ảnh và hiển thị lên màn hình.</p> <ol style="list-style-type: none"> 1. Người dùng chọn các chế độ “Sketch” hoặc “Trace” để vẽ. Hệ thống tải chế độ vẽ và thực hiện vẽ theo yêu cầu. 2. Người dùng điều chỉnh thông số ảnh như “Độ nhiều”, “Cấp độ”, “Độ mờ” để ảnh có kết quả phân tích ưng ý nhất. Hệ thống tải chế độ vẽ và thực hiện theo yêu cầu. 3. Người dùng kích vào các nút “Stroke” hoặc “Original” để thay đổi giữa hình ảnh đã hoặc chưa phân tích. Hệ thống tải chế độ vẽ và thực hiện theo yêu cầu. <p>➔ Use case kết thúc khi người dùng quay lại màn hình chính hoặc thoát khỏi hệ thống.</p>
	Luồng rẽ nhánh	<ol style="list-style-type: none"> 1. Tại bất kỳ bước nào trong luồng cơ bản nếu người dùng nhấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn hình trước đó.
Hậu điều kiện		Không có

2.1.4.3. Usecase Quay video

Bảng 2.4 Mô tả usecase "Quay video"

Mô tả		Use case này cho phép người dùng quay video lại các thao tác vẽ ảnh.
Tác nhân		Người dùng
Tiền điều kiện		Người dùng đã truy cập vào ứng dụng và thực hiện use case “Vẽ Ảnh”
Luồng sự kiện	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Use case bắt đầu khi người dùng kích nút “Record” trên thanh menu hình ảnh. Hệ thống bật tính năng quay video và thực hiện theo yêu cầu. 2. Người dùng kích nút “Record” để quay. Hệ thống quay video. 3. Người dùng kích vào nút “Stop” để dừng quay. Hệ thống dừng quay và phát đoạn video vừa quay lên màn hình. 4. Người dùng kích nút “Save” để lưu video. Hệ thống lưu video vừa quay vào thiết bị của người dùng. 5. Người dùng kích nút “Share” để chia sẻ video. Hệ thống lấy video từ thiết bị và chia sẻ. <p>Use case kết thúc.</p>
	Luồng rẽ nhánh	<ol style="list-style-type: none"> 1. Tại bất kỳ bước nào trong luồng cơ bản nếu người dùng nhấn nút “Back” thì hệ

		thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn hình trước đó.
Hậu điều kiện		Không có

2.1.4.4. Usecase Chụp ảnh

Bảng 2.5 Mô tả usecase chụp ảnh

Mô tả		Use case này cho phép người dùng quay video lại các thao tác vẽ ảnh.
Tác nhân		Người dùng
Tiền điều kiện		Người dùng đã truy cập vào ứng dụng và thực hiện use case “Vẽ Ảnh”
Luồng sự kiện	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Use case bắt đầu khi người dùng kích nút “Photo” trên thanh menu hình ảnh. Hệ thống bật tính năng chụp ảnh và hiển thị ảnh vừa chụp lên màn hình. 2. Người dùng kích nút “Save” để lưu video. Hệ thống lưu ảnh vừa chụp vào thiết bị của người dùng. 3. Người dùng kích nút “Share” để chia sẻ video. Hệ thống lấy video từ thiết bị và chia sẻ. <p>Use case kết thúc.</p>
	Luồng rẽ nhánh	<ol style="list-style-type: none"> 2. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ

		<p>sở dữ liệu thì sẽ hiển thị một thông báo lỗi và use case kết thúc.</p> <p>3. Tại bất kỳ bước nào trong luồng cơ bản nếu người dùng nhấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn hình trước đó.</p>
Hậu điều kiện		Không có

2.1.4.5. Usecase Xuất ảnh

Bảng 2.6 Mô tả use case “Xuất ảnh”

Mô tả		Use case này cho phép người dùng xuất ảnh đã chọn bằng cách in ảnh.
Tác nhân		Người dùng
Tiền điều kiện		Người dùng đã truy cập vào ứng dụng và dùng chức năng vẽ
Luồng sự kiện	Luồng sự kiện chính	<p>1. Use case bắt đầu khi người dùng kích nút “Print” phía dưới thanh menu của màn hình vẽ. Hệ thống hiển thị ảnh lên màn hình.</p> <p>2. Người dùng kích nút “Print”, màn hình hiển thị button “Connect”, người dùng kích vào button để xuất ảnh. Hệ thống xuất sẽ kết nối với máy in và thực hiện in.</p> <p>Use case kết thúc</p>

	Luồng rẽ nhánh	1. Tại bước 2 nêu không có máy in để kết nối, người dùng có thể chọn lưu dưới dạng file .pdf vào thiết bị
Hậu điều kiện		Không có

2.1.4.6. Usecase Quản lý ảnh

Bảng 2.7 Mô tả usecase “Quản lý ảnh”

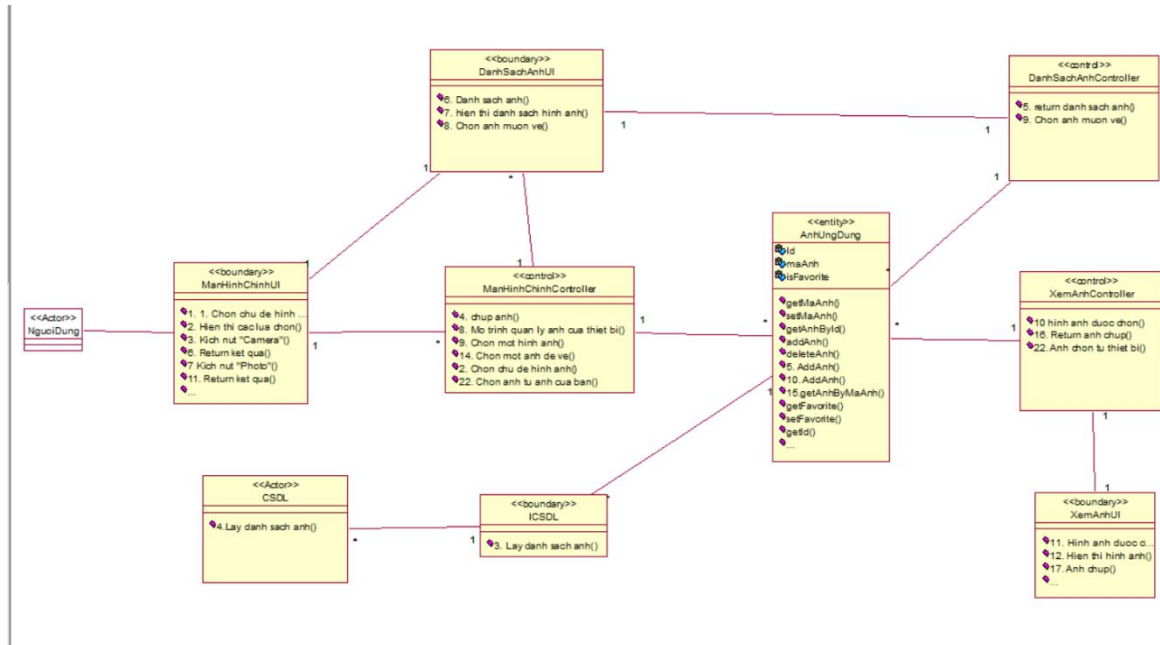
Mô tả		Use case này cho phép người dùng thêm, xóa ảnh, tích ảnh yêu thích.
Tác nhân		Người dùng
Tiền điều kiện		Người dùng đã truy cập vào ứng dụng
Luồng sự kiện	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Use case này bắt đầu khi người dùng kích nút “MyAlbum” trên màn hình ứng dụng. Hệ thống lấy danh sách ảnh từ bảng ANHUNGDUNG và hiển thị lên màn hình. 2. Xóa ảnh: <ul style="list-style-type: none"> - Người dùng bấm giữ hình ảnh. Hệ thống hiển thị màn hình yêu cầu xác nhận xóa. - Người dùng kích nút “DELETE”. Hệ thống sẽ xóa ảnh được chọn ra khỏi bảng ANHUNGDUNG và hiển thị danh sách ảnh đã được cập nhật. 3. Thêm vào ảnh yêu thích

		<ul style="list-style-type: none"> - Người dùng kích hình trái tim trên góc phải của hình ảnh. Hệ thống thay đổi màu trái tim thành hồng và lưu ảnh được chọn vào bảng ANHYEUTHICH. <p>5. Bỏ ảnh yêu thích</p> <ul style="list-style-type: none"> - Người dùng kích hình trái tim màu hồng trên góc phải của hình ảnh. Hệ thống thay đổi màu của trái tim thành không màu và xóa ảnh khỏi bảng ANHYEUTHICH.
	Luồng rẽ nhánh	<p>1. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi và use case kết thúc.</p> <p>2. Tại bất kỳ bước nào trong luồng cơ bản nếu người dùng nhấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn hình trước đó.</p>
	Hậu điều kiện	Không có

2.1.5. Biểu đồ chi tiết Use Case

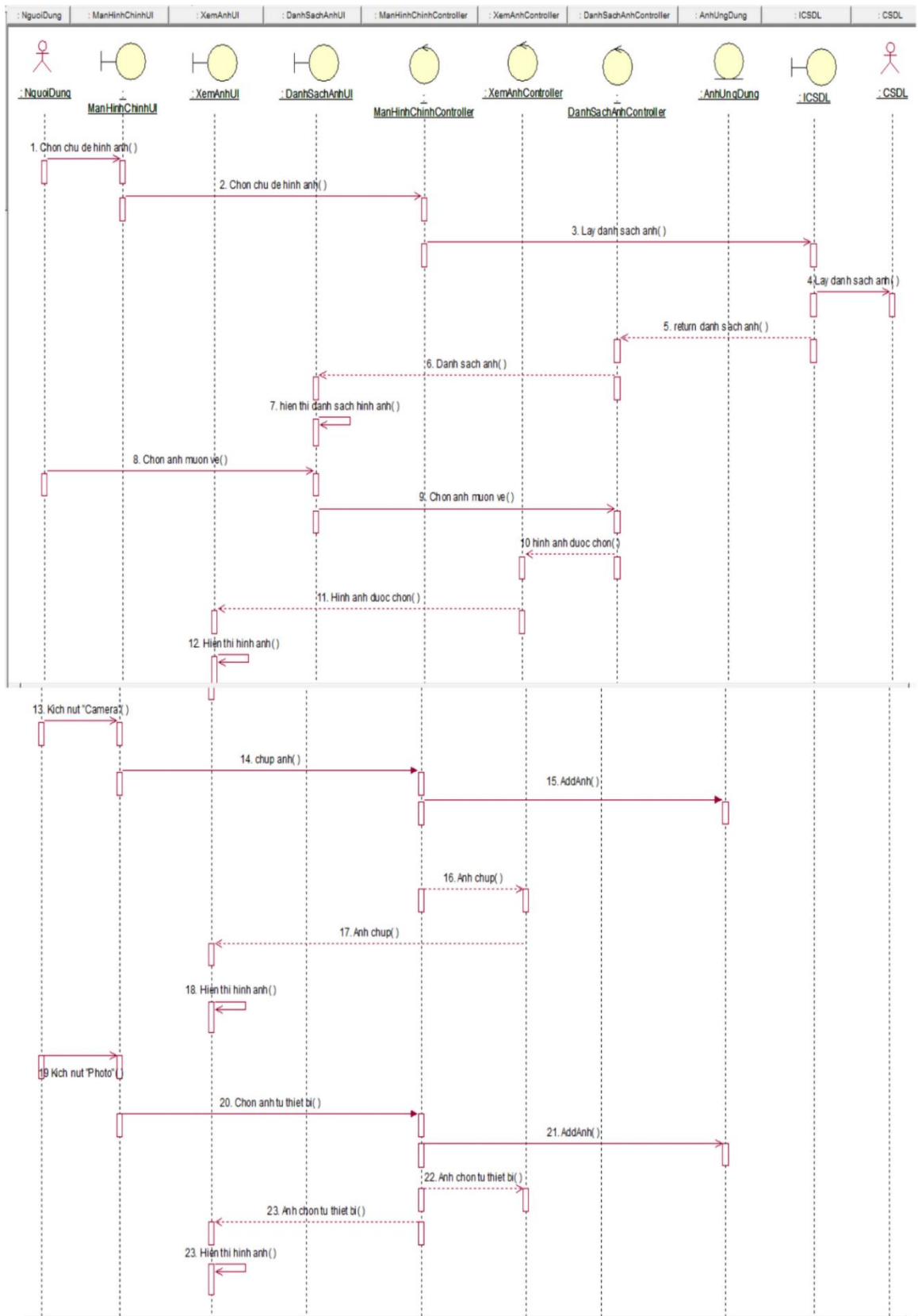
2.1.5.1. Use case Chọn ảnh

Biểu đồ lớp



Hình 2.1 Biểu đồ lớp use case "Chọn ảnh."

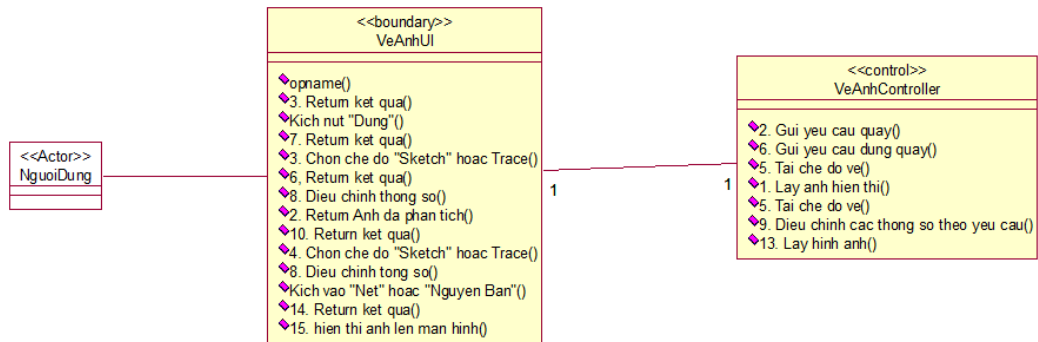
Biểu đồ trình tự



Hình 2.2 Biểu đồ trình tự use case chọn ảnh.

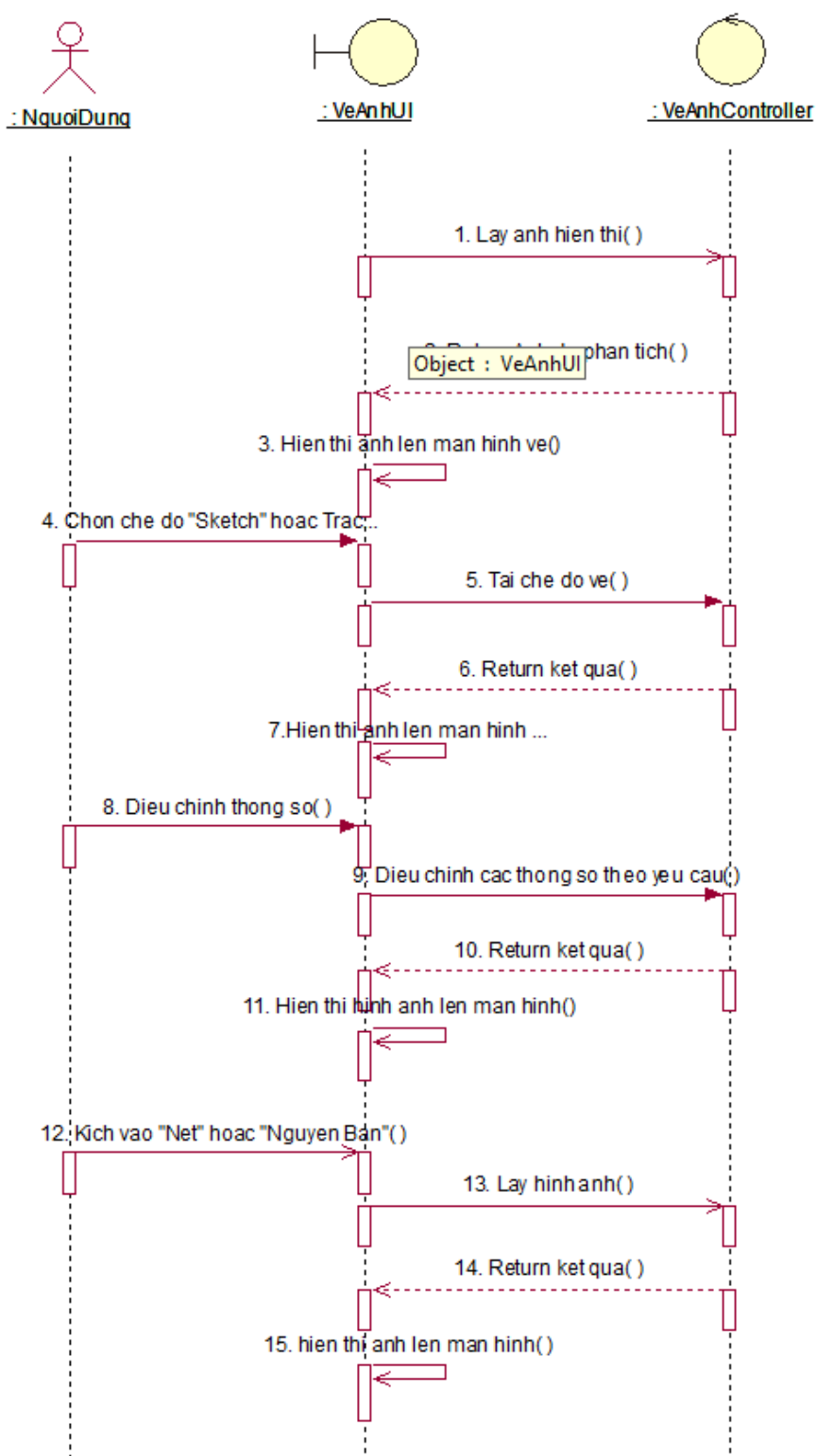
2.1.5.2. Use case Vẽ ảnh

Biểu đồ lớp



Hình 2.3 Biểu đồ lớp use case "Vẽ ảnh."

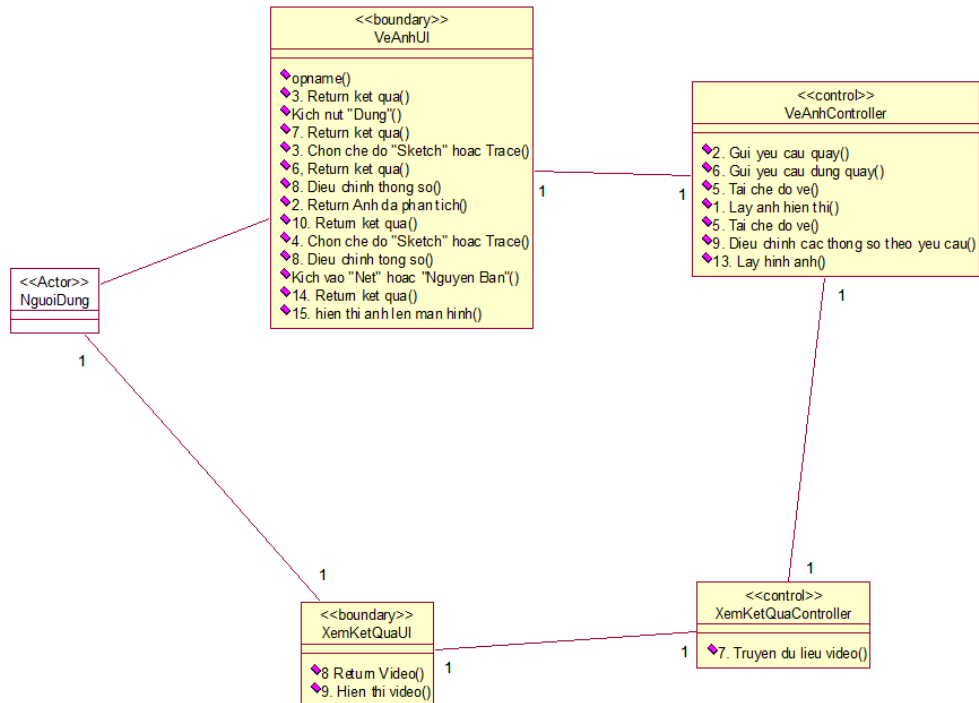
Biểu đồ trình tự



Hình 2.4 Biểu đồ trình tự use case "Vẽ ảnh."

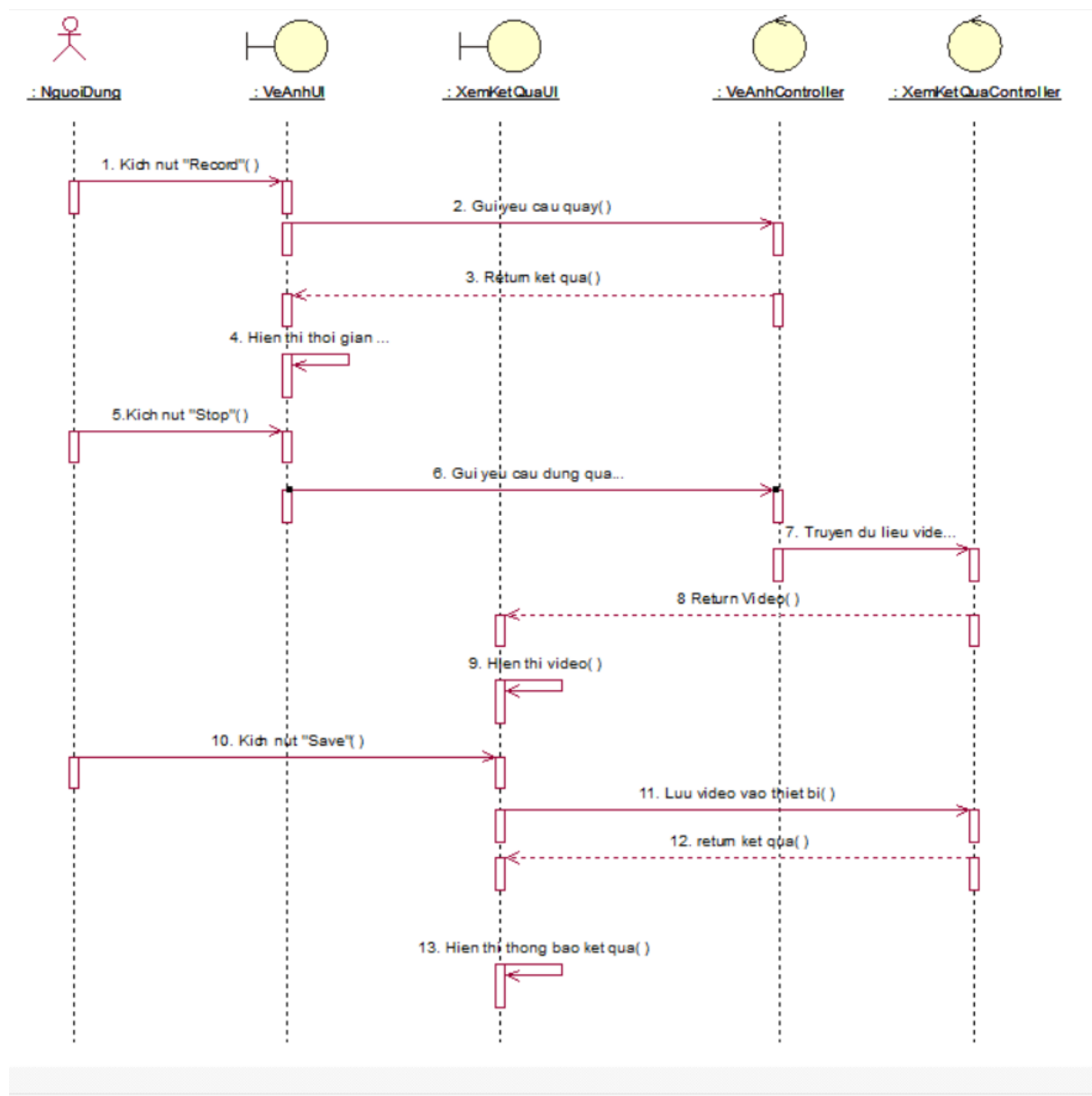
2.1.5.3. Use case Quay Video

Biểu đồ lớp



Hình 2.5 Biểu đồ lớp use case "Quay video"

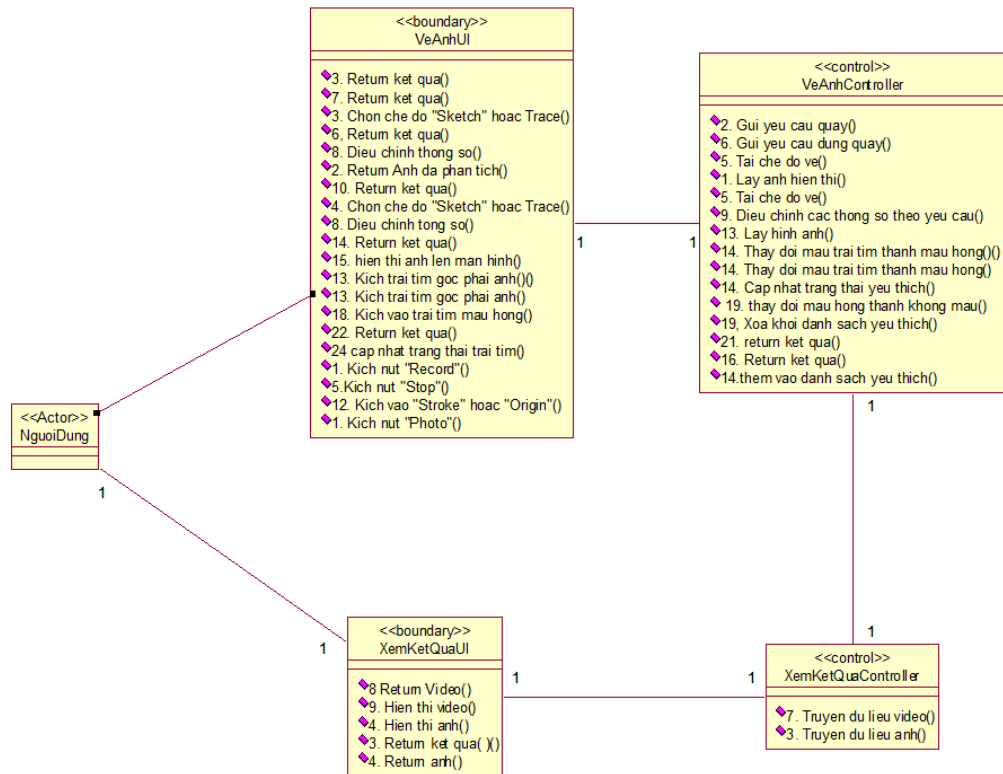
Biểu đồ trình tự



Hình 2.6 Biểu đồ trình tự use case "Quay video."

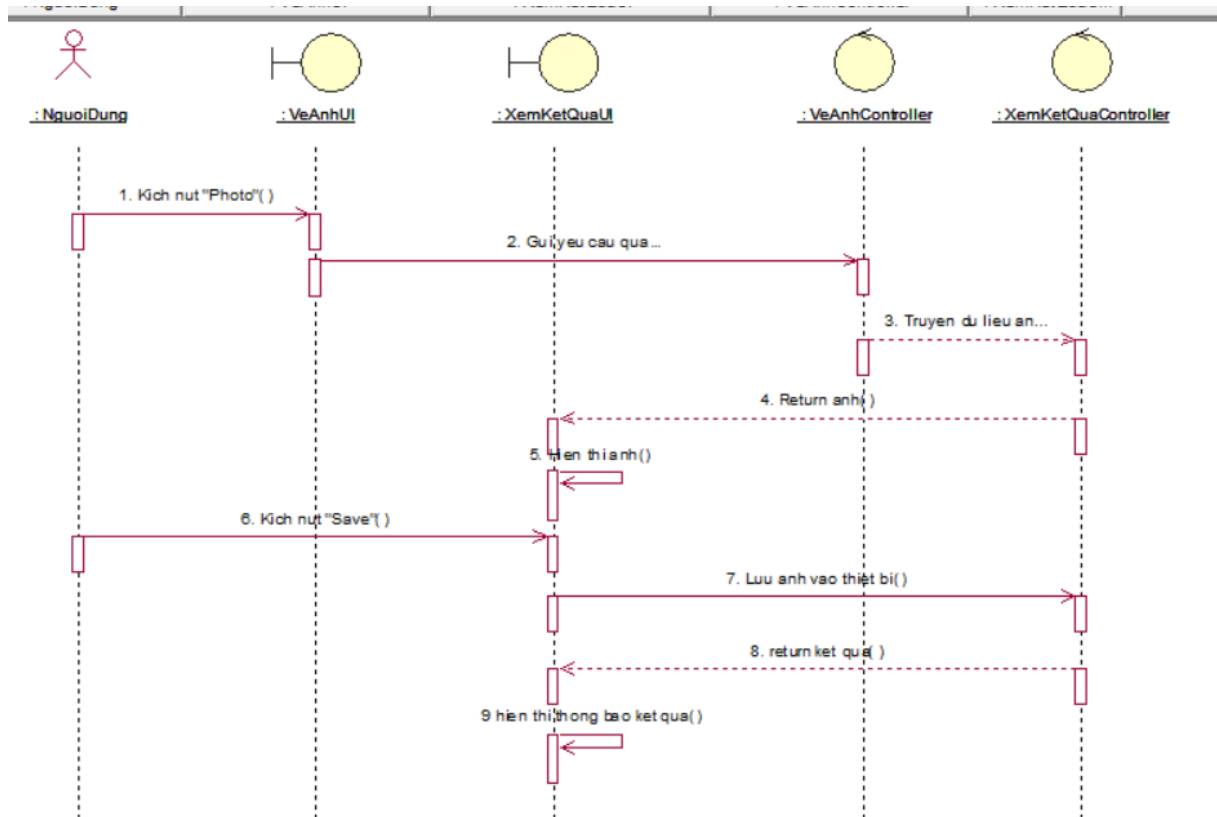
2.1.5.4. Use case Chụp Ảnh

Biểu đồ lớp



Hình 2.7 Biểu đồ lớp use case "Chụp Ảnh."

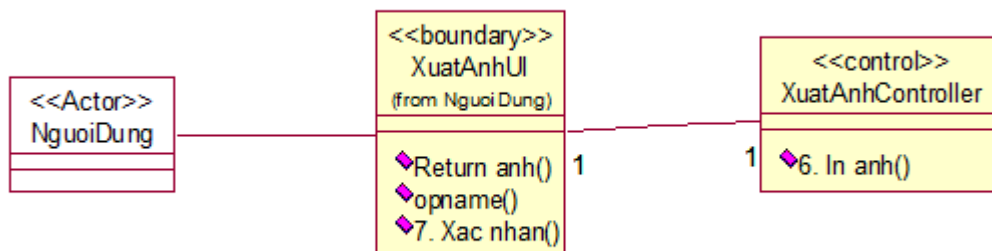
Biểu đồ trình tự



Hình 2.8 Biểu đồ trình tự use case "Chụp Ảnh"

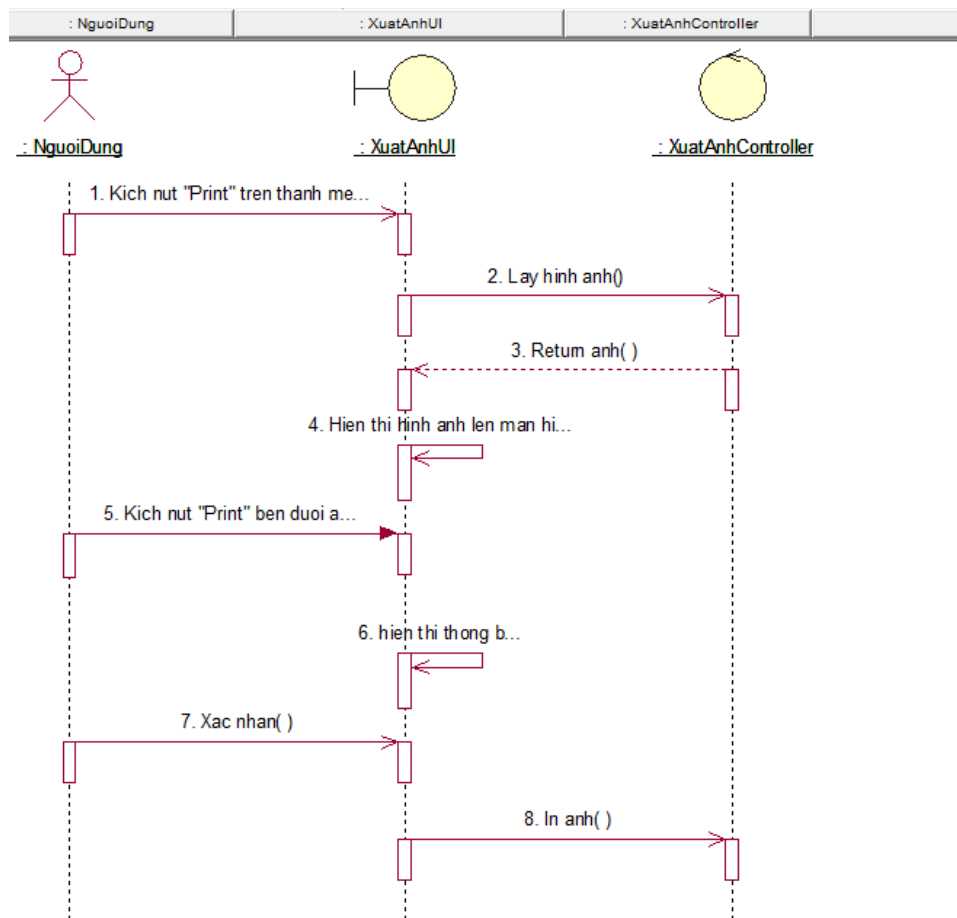
2.1.5.5. Use case Xuất Ảnh

Biểu đồ lớp



Hình 2.9 Biểu đồ lớp use case "Xuất Ảnh."

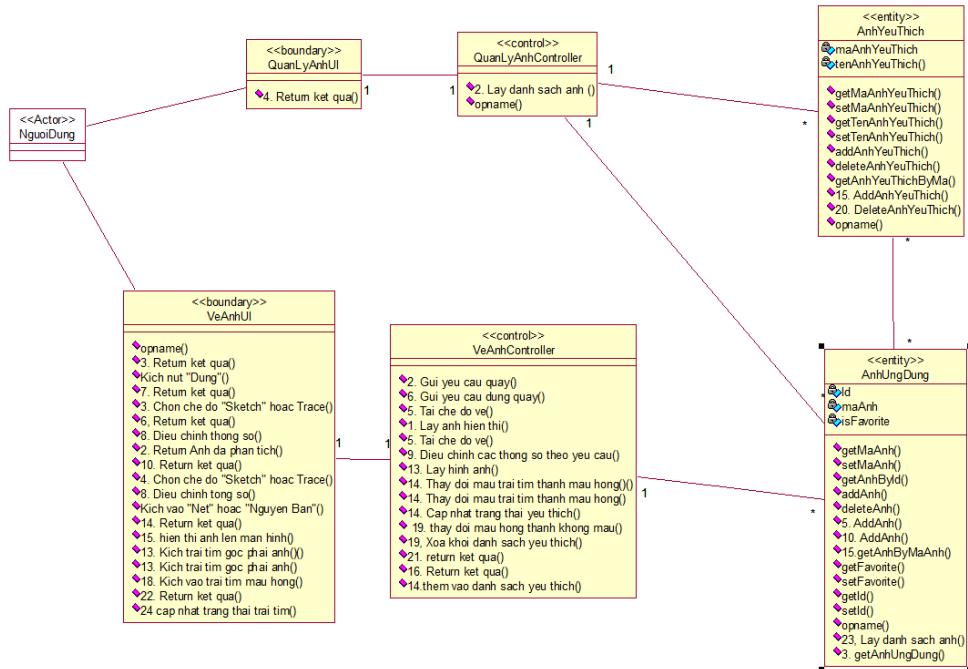
Biểu đồ trình tự



Hình 2.10 Biểu đồ trình tự use case "Xuất ảnh."

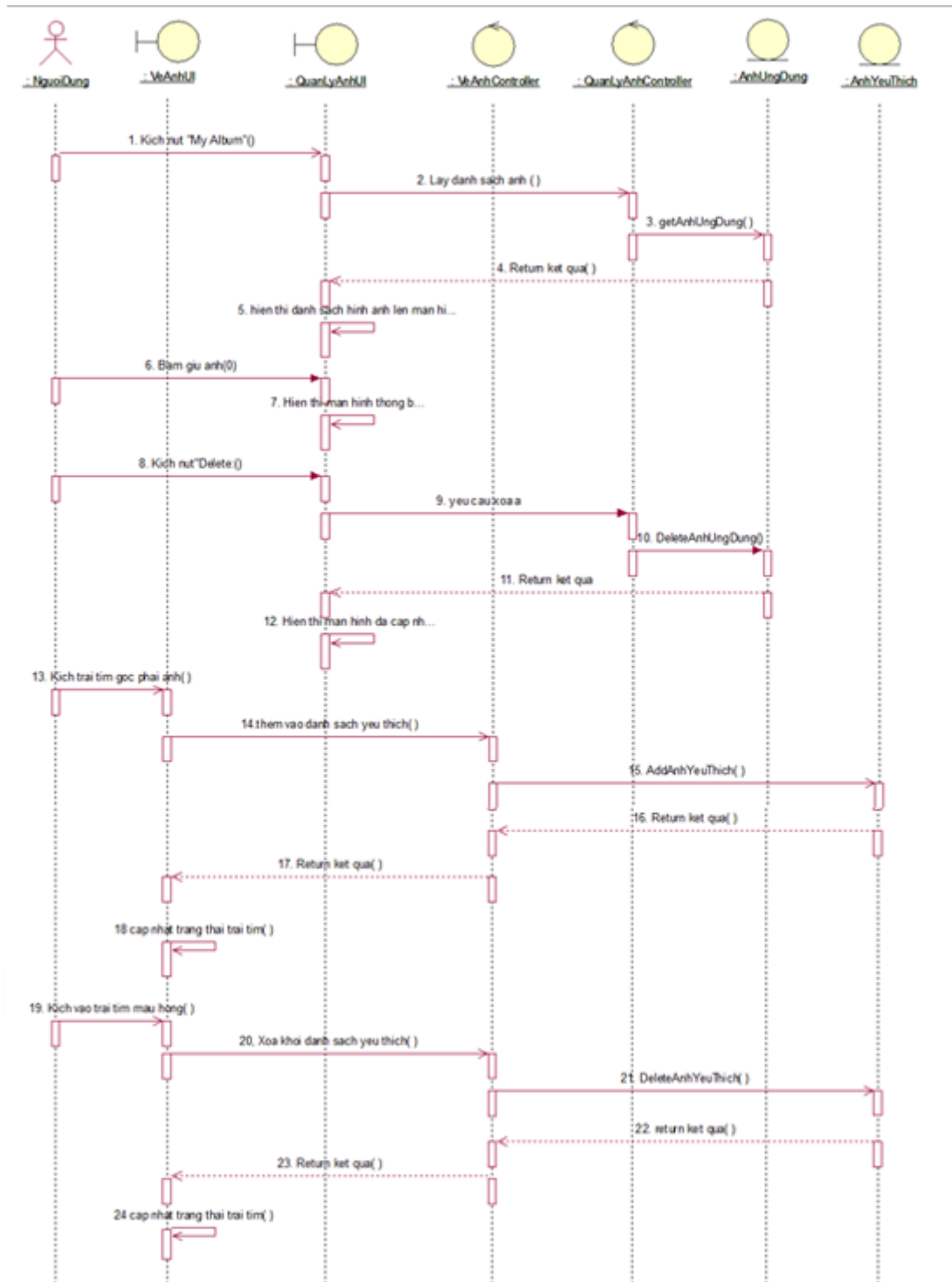
2.1.5.6. Use case Quản lý ảnh

Biểu đồ lớp



Hình 2.11 Biểu đồ lớp use case "Quản lý ảnh."

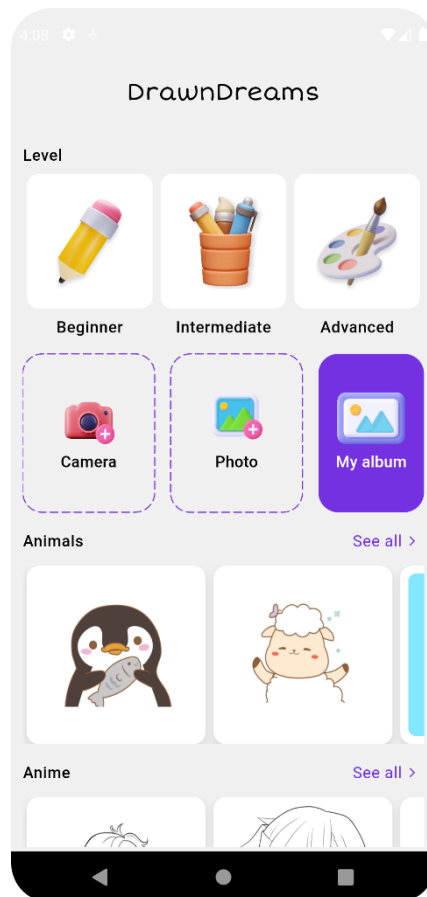
Biểu đồ trình tự



Hình 2.12 Biểu đồ trình tự use case "Quản lý ảnh."

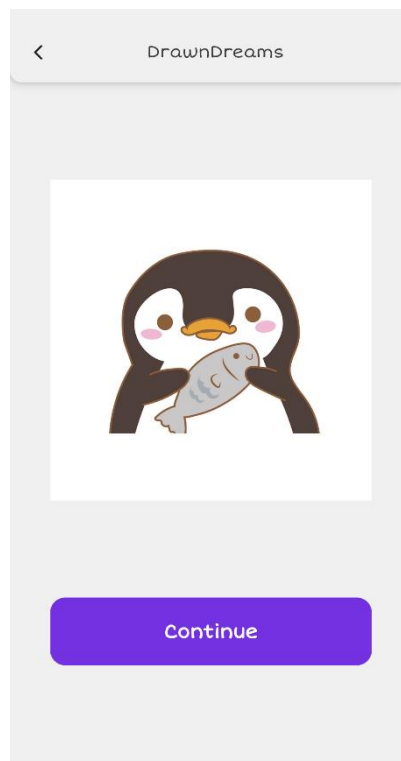
2.2. Thiết kế giao diện

2.2.1. *Giao diện Home*

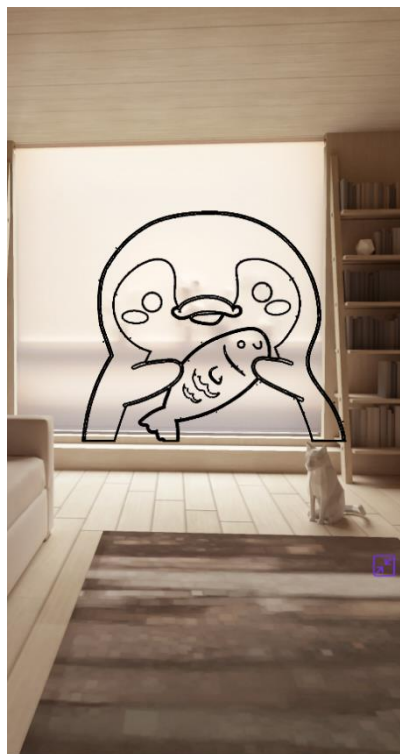


Hình 2.13 *Giao diện màn Home*

2.2.2. Giao diện vẽ



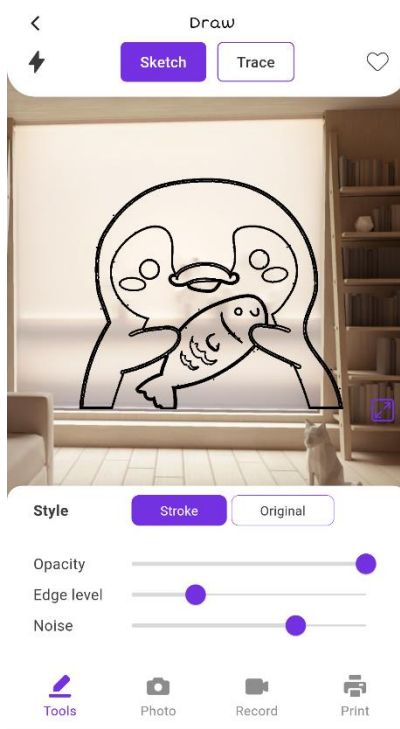
Hình 2.14 Giao diện xem ảnh sau khi chọn



Hình 2.15 Giao diện chức năng vẽ sketch

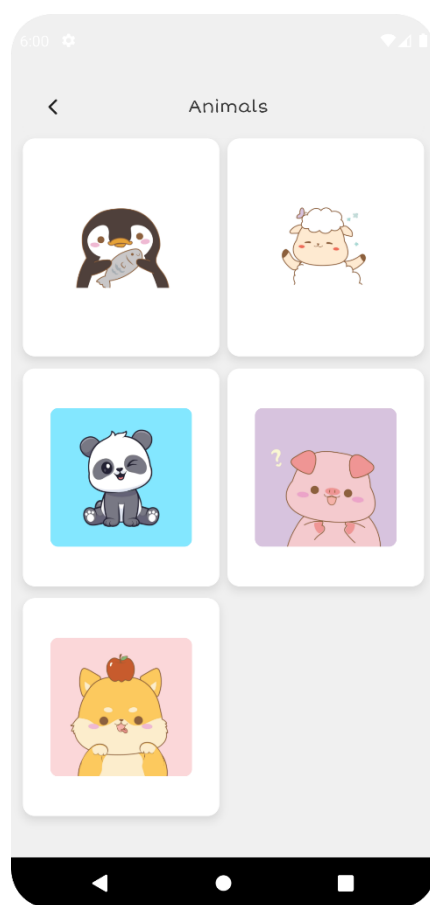


Hình 2.16 Giao diện chức năng vẽ trace



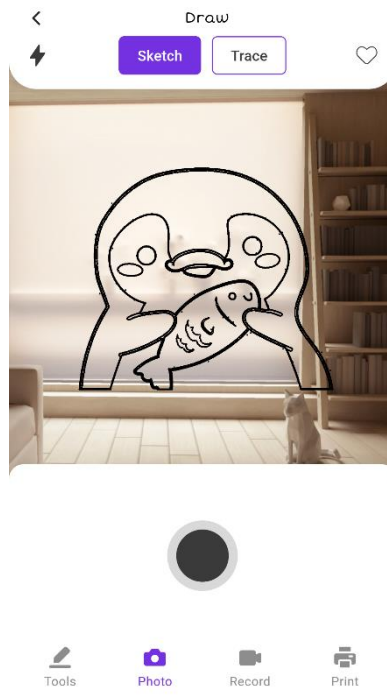
Hình 2.17 Giao diện chỉnh sửa ảnh

2.2.3. Giao diện danh sách ảnh có sẵn theo chủ đề

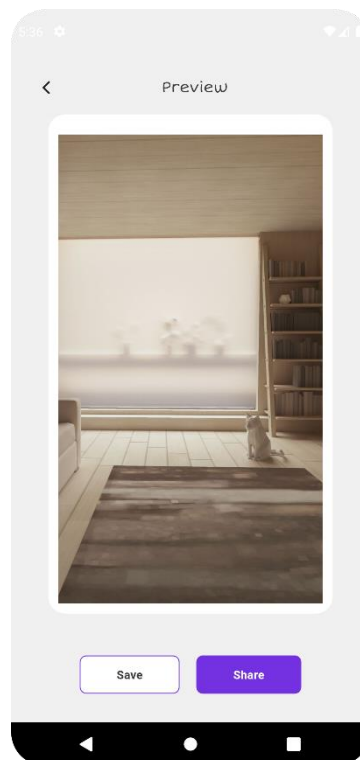


Hình 2.18 Giao diện danh sách ảnh theo chủ đề

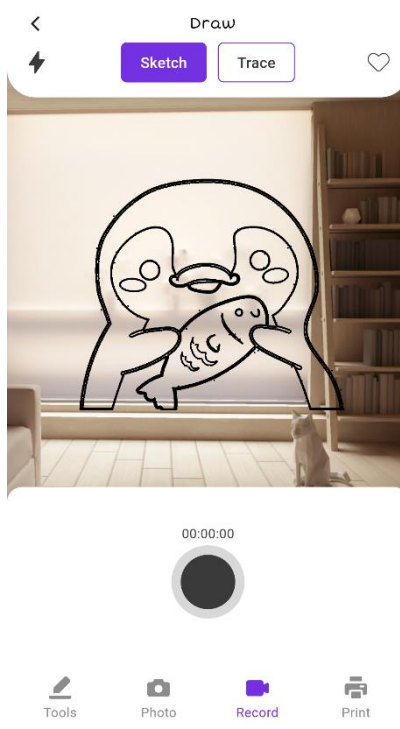
2.2.4. Giao diện quay chụp khi vẽ



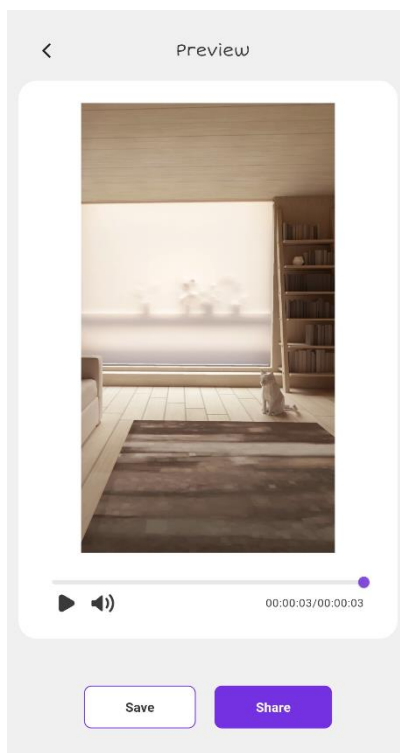
Hình 2.19 Giao diện chụp ảnh



Hình 2.20 Giao diện sau khi chụp

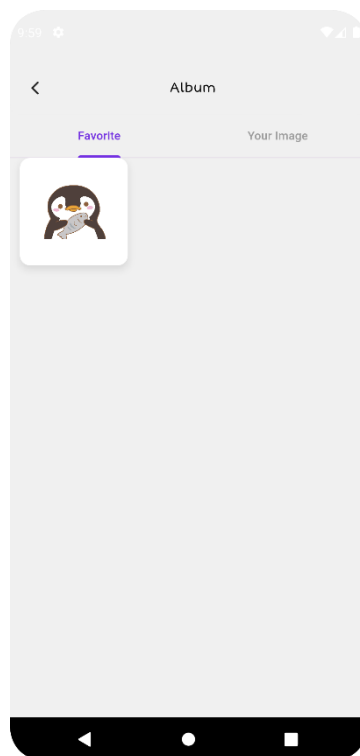


Hình 2.21 Giao diện quay video

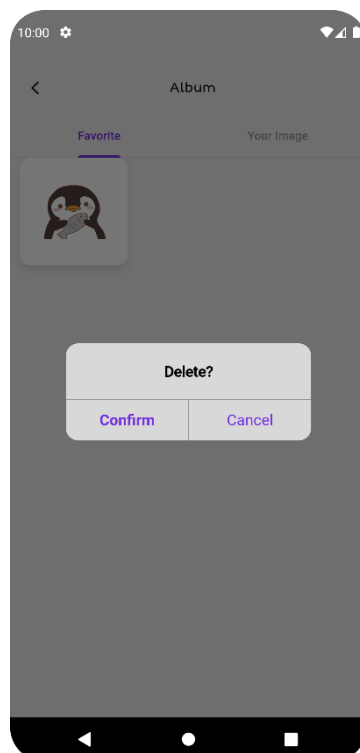


Hình 2.22 Giao diện sau khi quay video

2.2.5. Giao diện xem danh sách ảnh yêu thích



Hình 2.23 Giao diện xem ảnh yêu thích



Hình 2.24 Giao diện khi xóa hình ảnh

CHƯƠNG 3. KẾT QUẢ THU ĐƯỢC

3.1. Cài đặt môi trường

- Cài đặt Android Studio: Truy cập vào link: [Android Studio](#) để tải xuống phiên bản Android Studio mới nhất.
- Cài đặt JDK: Truy cập vào link [Java Downloads | Oracle](#) để tải xuống phiên bản JDK phù hợp
- Cài đặt môi trường Flutter
 - Đối với Window: [Flutter Window](#)
 - Đối với MacOS: [Flutter MacOS](#)
 - Kiểm tra lại với câu lệnh: flutter doctor -v

```
[✓] Flutter (Channel stable, 3.16.5, on Microsoft Windows [Version 10.0.22631.3447], locale en-US)
    • Flutter version 3.16.5 on channel stable at E:\flutter
    • Upstream repository https://github.com/flutter/flutter.git
    • Framework revision 78666c8dc5 (5 months ago), 2023-12-19 16:14:14 -0800
    • Engine revision 3f3e560236
    • Dart version 3.2.3
    • DevTools version 2.28.4

[✓] Windows Version (Installed version of Windows is version 10 or higher)

[!] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
    • Android SDK at C:\Users\PNT\AppData\Local\Android\sdk
    • Platform android-34, build-tools 34.0.0
    • Java binary at: C:\Program Files\Android\Android Studio\jbr\bin\java
    • Java version OpenJDK Runtime Environment (build 17.0.10+0--11572160)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses

[✓] Chrome - develop for the web
    • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

[✗] Visual Studio - develop Windows apps
    ✗ Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components

[✓] Android Studio (version 2023.3)
    • Android Studio at C:\Program Files\Android\Android Studio
    • Flutter plugin can be installed from:
      🔗 https://plugins.jetbrains.com/plugin/9212-flutter
    • Dart plugin can be installed from:
      🔗 https://plugins.jetbrains.com/plugin/6351-dart
    • Java version OpenJDK Runtime Environment (build 17.0.10+0--11572160)

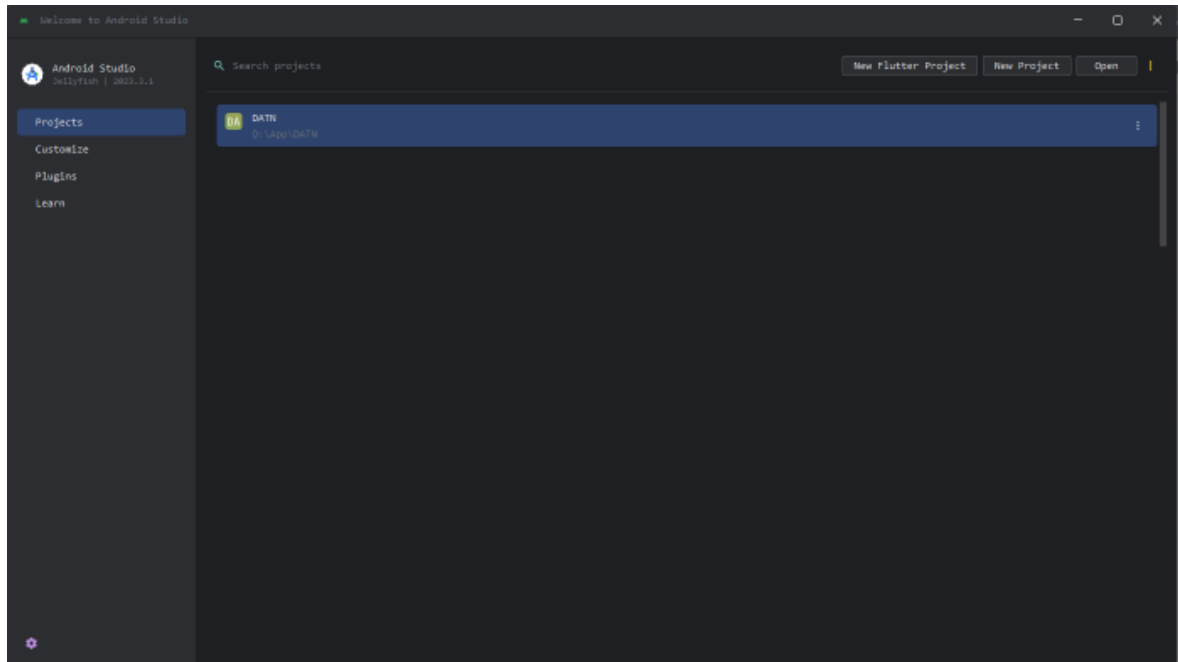
[✓] VS Code (version 1.89.1)
    • VS Code at C:\Users\PNT\AppData\Local\Programs\Microsoft VS Code
    • Flutter extension version 3.89.20240501

[✓] Connected device (3 available)
    • Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.22631.3447]
    • Chrome (web)      • chrome  • web-javascript • Google Chrome 124.0.6367.203
    • Edge (web)        • edge    • web-javascript • Microsoft Edge 124.0.2478.80

[✓] Network resources
    • All expected network resources are available.
```

Hình 3.1 Trạng thái cài đặt môi trường

- Cài đặt plugin Flutter và Dart trong Android Studio



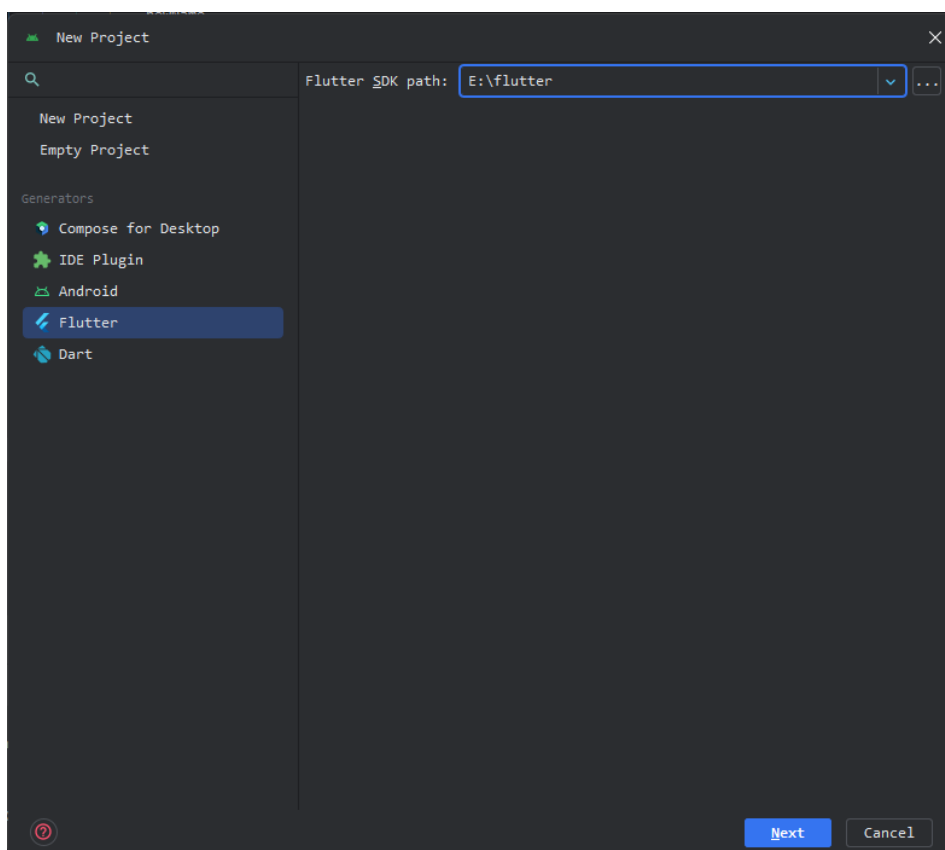
Hình 3.2 Giao diện bắt đầu Android studio

Cài đặt Flutter và Dart plugin

- Mở Android Studio.
- Chọn Plugin
- Tìm kiếm “flutter” trong khung tìm kiếm extensions, chọn Flutter trong danh sách, sau đó nhấn “install”.
- Tìm kiếm “dart” trong khung tìm kiếm extensions, chọn Dart trong danh sách, sau đó nhấn “install”.

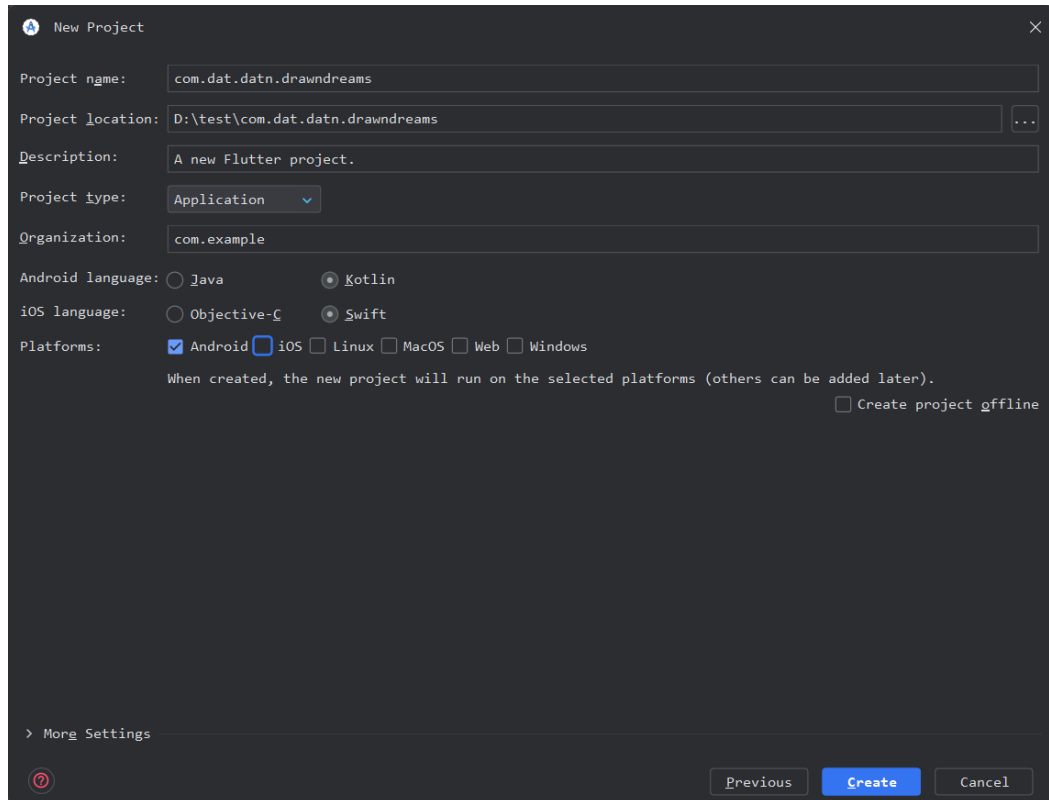
3.2. Cài đặt chương trình

- Bước 1: Chạy Android Studio, sau đó chọn Start a New Flutter Project để tiến hành xây dựng một ứng dụng mới. Chọn đường dẫn SDK Flutter cho dự án, màn hình mới hiển thị ra rồi điền các thông tin ứng dụng như hình.

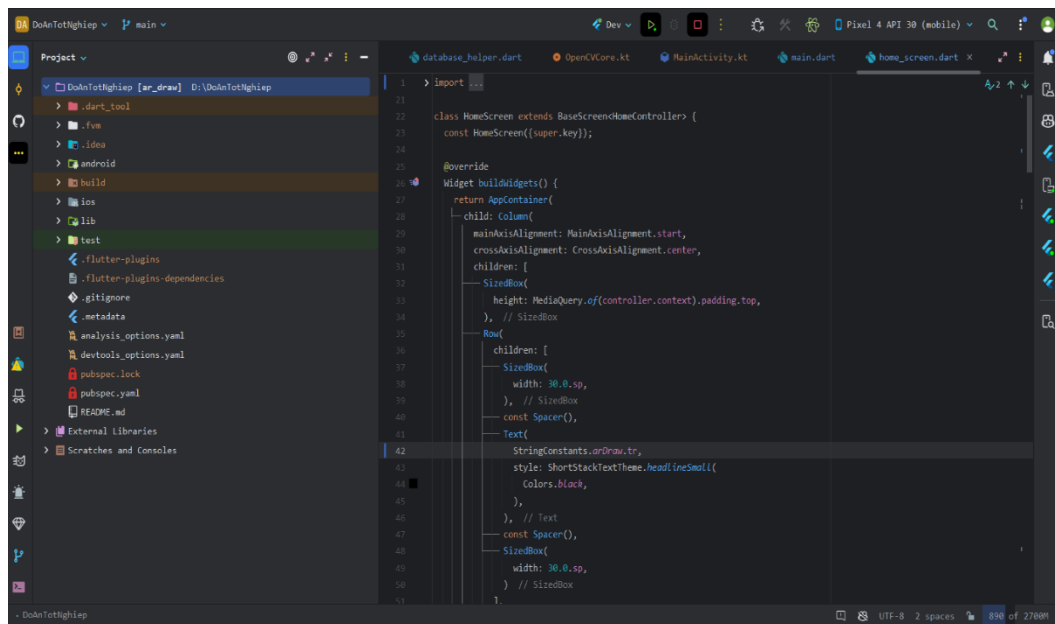


Hình 3.3 Cửa sổ chọn đường dẫn SDK Flutter

- Bước 2: Chọn Next màn hình hiển thị cửa sổ tạo project, sau đó điền thông tin ứng dụng như hình:



Hình 3.4 Cửa sổ Tạo project



Hình 3.5 Giao diện project

Quá trình Generate hoàn thành và tạo được file cài đặt trên hệ điều hành Android.

3.3. Kết quả thu được

Hệ thống được xây dựng với các chức năng, đó là:

3.3.1. Chọn ảnh với nhiều nguồn ảnh

- Chọn ảnh có sẵn trong ứng dụng
- Chọn ảnh từ thư viện của của thiết bị
- Lấy ảnh chụp qua camera của thiết bị

3.3.2. Phân tích ảnh thành đường nét

- Chuyển dữ liệu ảnh đầu vào thành ảnh đường đen trên nền không màu
- Cho phép chỉnh sửa thông số để có kết quả ưng ý nhất

3.3.3. Quay lại quá trình vẽ

- Quay lại quá trình vẽ nếu muốn
- Lưu video đã quay vào bộ nhớ thiết bị

3.3.4. In ảnh

- Truyền hình ảnh đến trình in của thiết bị để thực hiện việc in

3.4. Kết quả kiểm thử

3.4.1. Chiến lược kiểm thử

- Kiểm thử ở mức hệ thống (ST) và kiểm thử chấp nhận (UAT)
- Dùng kiểu kiểm thử thủ công (manual test) bao gồm kiểm thử GUI của từng chức năng.
- Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử GUI và chức năng.

3.4.2. Kết quả kiểm thử

- Tỷ lệ test case đạt (passed): 100%.

- Tỷ lệ test case không đạt (failed): 0%.
- Hệ thống chạy ổn định trên các phiên bản khác nhau (Android 7.0 Nougat, Android 8 Oreo, Android 9 Pie, Android 10, Android 11, Android 12).

Kết quả chi tiết:

Bảng 3.1 Bảng kiểm thử chức năng

STT	Nội dung	Mục đích kiểm thử	Đầu vào	Đầu ra mong muốn	Kết quả
1	Chọn ảnh có sẵn trong ứng dụng	Đảm bảo chức năng chọn ảnh có sẵn hoạt động tốt	Hình ảnh mẫu trong ứng dụng	Hình ảnh được chọn thành công	Đạt
2	Chọn ảnh từ thư viện của thiết bị	Đảm bảo chức năng chọn ảnh từ thư viện hoạt động tốt	Hình ảnh từ thư viện thiết bị	Hình ảnh từ thư viện được chọn thành công	Đạt
3	Lấy ảnh chụp qua camera của thiết bị	Đảm bảo chức năng chụp ảnh từ camera hoạt động tốt	Ảnh chụp từ camera	Ảnh chụp từ camera được lưu và chọn thành công	Đạt

4	Chuyển dữ liệu ảnh thành ảnh đường đen trên nền không màu	Đảm bảo chức năng chuyển đổi ảnh hoạt động tốt	Ảnh đầu vào	Ảnh đường đen trên nền không màu	Đạt
5	Chỉnh sửa thông số để có kết quả ưng ý nhất	Đảm bảo chức năng chỉnh sửa thông số hoạt động tốt	Ảnh đầu vào	Ảnh với thông số chỉnh sửa được người dùng mong muốn	Đạt
6	Quay lại quá trình vẽ	Đảm bảo chức năng quay lại quá trình vẽ hoạt động tốt	Quá trình vẽ	Quá trình vẽ được quay lại và lưu vào bộ nhớ thiết bị	Đạt
7	Lưu video đã quay vào bộ nhớ thiết bị	Đảm bảo chức năng lưu video hoạt động tốt	Video quá trình vẽ	Video được lưu vào bộ nhớ thiết bị thành công	Đạt
8	Truyền hình ảnh đến trình in của thiết bị để thực hiện việc in	Đảm bảo chức năng in ảnh hoạt động tốt	Hình ảnh cần in	Hình ảnh được truyền đến trình in và in thành công	Đạt

KẾT LUẬN

Trong quá trình thực hiện đề tài "xây dựng ứng dụng di động dạy vẽ cho trẻ nhỏ áp dụng công nghệ xử lý ảnh sử dụng flutter và kotlin trên nền tảng android", em đã áp dụng và tổng hợp kiến thức từ trường học và quá trình thực tập. Đã hoàn thành các công việc như khảo sát và phân tích yêu cầu, phân tích thiết kế và đặc tả hệ thống, thiết kế cơ sở dữ liệu, hiểu và sử dụng ngôn ngữ lập trình Dart, xây dựng chức năng đo và theo dõi nhịp tim, theo dõi huyết áp, theo dõi cân nặng. Đồng thời, cũng đã lập kế hoạch và thực hiện kiểm thử hệ thống, áp dụng mô hình MVC và quản lý lifecycle của ứng dụng.

Trong tương lai, em hiện có một số phương hướng để phát triển như sau:

- Phát triển "Style" và chế độ dark mode: Ứng dụng cung cấp sự linh hoạt cho người dùng trong việc tùy chỉnh giao diện, cho phép chọn màu sắc và chế độ sáng/tối theo sở thích cá nhân. Điều này tạo ra trải nghiệm người dùng cá nhân hơn và thích nghi với điều kiện ánh sáng môi trường.
- Nhận phản hồi từ người dùng: Ứng dụng có khả năng nhận phản hồi từ người dùng, thông qua hệ thống góp ý và đánh giá. Điều này giúp cải thiện trải nghiệm người dùng và tăng tính tương tác trong ứng dụng.
- Phát triển thành một mạng xã hội độc lập cho phép người dùng có thể chia sẻ kết quả của quá trình vẽ cho mọi người cùng chiêm ngưỡng và thảo luận.

TÀI LIỆU THAM KHẢO

- [1]. [] Tổ HTTT Đại học Công Nghiệp Hà Nội (2015), **Giáo trình hệ quản trị cơ sở dữ liệu SQL**, Nhà xuất bản Giáo dục, Hà Nội.
- [2]. Tổ HTTT Đại học Công Nghiệp Hà Nội (2015), **Giáo trình phân tích thiết kế hệ thống**, Nhà xuất bản Giáo dục, Hà Nội.
- [3]. Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan (2015), **Giáo trình Phân tích thiết kế hướng đối tượng**, Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội.
- [4]. TopDev, [OpenCV là gì? Học Computer Vision không khó! | TopDev](#)
- [5]. VinBigData (2022), [Xử lý hình ảnh trong Python: từ thuật toán đến công cụ - vinbigdata](#)
- [6]. Amela.vn (2023), [Phân biệt Native App, Web App và Hybrid App. Nên dùng loại app nào? \(amela.vn\)](#)
- [7]. W3schools, [SQL Introduction \(w3schools.com\)](#)
- [8]. Thống PM (2022), [Tất tần tật về mô hình MVC \(viblo.asia\)](#)
- [9]. Flutter, (n.d.), **Flutter**, Lấy từ: <https://docs.flutter.dev/>
- [10]. OpenPlanning, (n.d.), **Hướng dẫn lập trình Android cho người mới bắt đầu**, Lấy từ: <https://openplanning.net/10415/huong-dan-lap-trinh-android-cho-nguoi-moi-bat-dau>
- [11]. Adrian Rosebrock (2022), [Adaptive Thresholding with OpenCV \(cv2.adaptiveThreshold\) - PyImageSearch](#)
- [12]. Google (2024), [OpenCV: Contours : Getting Started](#)