|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **TRẦN QUỐC DOANH** | | **CÔNG NGHỆ THÔNG TIN** | | |  | | --- | | **BỘ CÔNG THƯƠNG** | | **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI** |   **------------------------------------------**       |  | | --- | | ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC | | NGÀNH CÔNG NGHỆ THÔNG TIN | | **XÂY DỰNG ỨNG DỤNG DI ĐỘNG QUẢN LÝ CHẤT LƯỢNG ẢNH VÀ DUNG LƯỢNG LƯU TRỮ THÔNG MINH** |  |  |  |  | | --- | --- | --- | | **CBHD** | **:** | **ThS. Nguyễn Xuân Hoàng** | | **Sinh viên** | **:** | **Trần Quốc Doanh** | | **Mã sinh viên** | **:** | **2021600988** |       *Hà Nội – Năm 2025* |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **BỘ CÔNG THƯƠNG** | | **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI** |   **------------------------------------------**       |  | | --- | | ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC | | NGÀNH CÔNG NGHỆ THÔNG TIN | | **XÂY DỰNG ỨNG DỤNG DI ĐỘNG QUẢN LÝ CHẤT LƯỢNG ẢNH VÀ DUNG LƯỢNG LƯU TRỮ THÔNG MINH** |  |  |  |  | | --- | --- | --- | | **CBHD** | **:** | **ThS. Nguyễn Xuân Hoàng** | | **Sinh viên** | **:** | **Trần Quốc Doanh** | | **Mã sinh viên** | **:** | **2021600988** |       *Hà Nội – Năm 2025* |

# LỜI MỞ ĐẦU

Em xin được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và tổ chức đã hỗ trợ, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài tốt nghiệp: “Xây dựng ứng dụng di động quản lý chất lượng ảnh và dung lượng lưu trữ thông minh”. Em xin gửi lời cảm ơn chân thành tới các thầy cô trong khoa Công nghệ Thông tin đã tận tình giảng dạy và truyền đạt cho em những kiến thức quý báu trong suốt quá trình học tập tại trường. Chính nhờ sự chỉ bảo tận tình ấy, em mới có nền tảng để thực hiện và hoàn thiện đề tài này.

Em xin đặc biệt cảm ơn ThS. Nguyễn Xuân Hoàng – người đã trực tiếp hướng dẫn, động viên và đóng góp nhiều ý kiến chuyên môn quý báu giúp em hoàn thành tốt báo cáo đồ án này trong suốt thời gian qua.

Báo cáo đồ án của em được thực hiện trong khoảng thời gian 9 tuần và được chia thành 5 chương như sau:

* Chương 1: Tổng quan đề tài
* Chương 2: Cơ sở lý thuyết
* Chương 3: Phân tích yêu cầu phần mềm
* Chương 4: Kết quả và đánh giá
* Chương 5: Kết luận và hướng phát triển

Do còn hạn chế về kinh nghiệm thực tiễn cũng như thời gian, chắc chắn báo cáo của em vẫn còn nhiều thiếu sót. Em rất mong nhận được sự đóng góp, nhận xét từ quý thầy cô để hoàn thiện hơn về kiến thức và kỹ năng trong lĩnh vực này.

Em xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 2025

Sinh viên   
Trần Quốc Doanh

# MỤC LỤC

[LỜI MỞ ĐẦU i](#_Toc209609567)

[MỤC LỤC ii](#_Toc209609568)

[DANH MỤC CÁC CHỮ VIẾT TẮT v](#_Toc209609569)

[DANH MỤC HÌNH ẢNH vii](#_Toc209609570)

[DANH MỤC BẢNG BIỂU viii](#_Toc209609571)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc209609572)

[1.1. Giới thiệu đề tài 1](#_Toc209609573)

[1.2. Lý do chọn đề tài 1](#_Toc209609574)

[1.3. Mục tiêu và phạm vi nghiên cứu 2](#_Toc209609575)

[1.4. Ý nghĩa khoa học và thực tiễn của đề tài 3](#_Toc209609576)

[1.5. Phương pháp nghiên cứu 4](#_Toc209609577)

[1.6. Đối tượng nghiên cứu 4](#_Toc209609578)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc209609579)

[2.1. Tổng quan về lập trình di động 5](#_Toc209609580)

[2.1.1. Hybrid App 5](#_Toc209609581)

[2.1.2. Native App 5](#_Toc209609582)

[2.1.3. Web App 6](#_Toc209609583)

[2.2. Công cụ lập trình Android Studio 7](#_Toc209609584)

[2.2.1. Giới thiệu Android Studio 7](#_Toc209609585)

[2.2.2. Sơ lược lịch sử Android Studio 8](#_Toc209609586)

[2.2.3. Ứng dụng của Android Studio 8](#_Toc209609587)

[2.3. Ngôn ngữ lập trình Dart 8](#_Toc209609588)

[2.4. Một số framework trong lập trình Flutter 9](#_Toc209609589)

[2.4.1. Flutter SDK 9](#_Toc209609590)

[2.4.2. AngularDart 9](#_Toc209609591)

[2.4.3. StageXL 9](#_Toc209609592)

[2.5. Hive Database 9](#_Toc209609593)

[2.6. Tổng quan về mô hình Clean Architecture 10](#_Toc209609594)

[2.7. Nguyên tắc SOLID trong phát triển phần mềm 12](#_Toc209609595)

[2.8. Thuật toán xử lý ảnh 14](#_Toc209609596)

[2.8.1. Giới thiệu các thuật toán sử dụng 14](#_Toc209609597)

[2.8.2. Các bước trong quy trình xử lý ảnh 16](#_Toc209609598)

[2.9. Phân tích thư viện xóa nền ảnh 17](#_Toc209609599)

[2.9.1. Giới thiệu thư viện 17](#_Toc209609600)

[2.9.2. Kiến trúc và cơ chế hoạt động 18](#_Toc209609601)

[2.9.3. Ưu điểm 18](#_Toc209609602)

[2.9.4. Nhược điểm 19](#_Toc209609603)

[2.9.5. So sánh với các mô hình đã có sẵn 19](#_Toc209609604)

[2.9.6. Đánh giá thực nghiệm 21](#_Toc209609605)

[2.9.7. Đánh giá tổng quan 21](#_Toc209609606)

[2.9.7. Kết luận 22](#_Toc209609607)

[CHƯƠNG 3: PHÂN TÍCH YÊU CẦU PHẦN MỀM 24](#_Toc209609608)

[3.1. Mô hình hóa chức năng 24](#_Toc209609609)

[3.1.1. Các tác nhân hệ thống 24](#_Toc209609610)

[3.1.2. Các use case hệ thống 24](#_Toc209609611)

[3.1.3. Biểu đồ use case 24](#_Toc209609612)

[3.1.4. Đặc tả use case 24](#_Toc209609613)

[3.1.5. Biểu đồ chi tiết Use case 26](#_Toc209609614)

[3.2. Thiết kế giao diện 32](#_Toc209609615)

[3.2.1. Giao diện Home 32](#_Toc209609616)

[3.2.2. Giao diện Xóa ảnh/video 33](#_Toc209609617)

[3.2.3. Giao diện tách nền ảnh 35](#_Toc209609618)

[3.2.4. Giao diện kiểm tra ảnh gần giống/ trùng khớp 36](#_Toc209609619)

[3.2.5. Giao diện thống kê 37](#_Toc209609620)

[CHƯƠNG 4: KẾT QUẢ VÀ ĐÁNH GIÁ 38](#_Toc209609621)

[4.1. Cài đặt môi trường 38](#_Toc209609622)

[4.2. Cài đặt chương trình 39](#_Toc209609623)

[4.3. Kết quả thu được 41](#_Toc209609624)

[4.3.1.  Dọn dẹp ảnh thông minh (Clean) 42](#_Toc209609625)

[4.3.2.  Xóa nền ảnh tự động (Background Removal) 42](#_Toc209609626)

[4.3.3. Ký ức ảnh (Memory) 42](#_Toc209609627)

[4.3.4. Thống kê chi tiết (Statistics) 42](#_Toc209609628)

[4.3.5. Cài đặt và tùy chọn (Settings) 42](#_Toc209609629)

[4.4. Kết quả kiểm thử 43](#_Toc209609630)

[4.4.1. Chiến lược kiểm thử 43](#_Toc209609631)

[4.4.2. Kết quả kiểm thử 43](#_Toc209609632)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 50](#_Toc209609633)

[TÀI LIỆU THAM KHẢO 51](#_Toc209609634)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Viết tắt** | **Dịch nghĩa** |
| 1 | IDE | Integrated Development Environment – Môi trường phát triển tích hợp |
| 2 | SDK | Software Development Kit – Bộ công cụ phát triển phần mềm |
| 3 | UI | User Interface – Giao diện người dùng |
| 4 | UX | User Experience – Trải nghiệm người dùng |
| 5 | API | Application Programming Interface – Giao diện lập trình ứng dụng |
| 6 | AI | Artificial Intelligence – Trí tuệ nhân tạo |
| 7 | BLoC | Business Logic Component – Thành phần xử lý logic nghiệp vụ |
| 8 | pHash | Perceptual Hash – Hàm băm cảm nhận ảnh |
| 9 | DCT | Discrete Cosine Transform – Biến đổi cosin rời rạc |
| 10 | ONNX | Open Neural Network Exchange – Định dạng mô hình học sâu mở |
| 11 | CNN | Convolutional Neural Network – Mạng nơ-ron tích chập |
| 12 | ML Kit | Machine Learning Kit – Bộ công cụ học máy của Google |
| 13 | SQL | Structured Query Language – Ngôn ngữ truy vấn có cấu trúc |
| 14 | SRP | Single Responsibility Principle – Nguyên tắc trách nhiệm đơn |
| 15 | OCP | Open/Closed Principle – Nguyên tắc đóng/mở |
| 16 | LSP | Liskov Substitution Principle – Nguyên tắc thay thế Liskov |
| 17 | ISP | Interface Segregation Principle – Nguyên tắc phân tách giao diện |
| 18 | DIP | Dependency Inversion Principle – Nguyên tắc đảo ngược phụ thuộc |

# DANH MỤC HÌNH ẢNH

[Hình 2.1 Cấu trúc Clean architecture 11](#_Toc209609635)

[Hình 2.2 Các bước trong quy trình xử lý ảnh 16](#_Toc209609636)

[Hình 3.1 Giao diện màn Home 32](#_Toc209609637)

[Hình 3.2 Giao diện màn Memory 32](#_Toc209609638)

[Hình 3.3 Giao diện màn xóa ảnh 33](#_Toc209609639)

[Hình 3.4 Giao diện màn xóa video 33](#_Toc209609640)

[Hình 3.5 Giao diện màn xem lại ảnh 34](#_Toc209609641)

[Hình 3.6 Giao diện màn xác nhận xóa 34](#_Toc209609642)

[Hình 3.7 Giao diện màn xóa thành công 35](#_Toc209609643)

[Hình 3.8 Giao diện màn trước khi tách nền ảnh 35](#_Toc209609644)

[Hình 3.9 Giao diện màn sau khi tách nền ảnh 36](#_Toc209609645)

[Hình 3.10 Giao diện màn ảnh gần giống 36](#_Toc209609646)

[Hình 3.11 Giao diện màn ảnh trùng khớp 37](#_Toc209609647)

[Hình 3.12 Giao diện màn thống kê 37](#_Toc209609648)

[Hình 4.1 Trạng thái cài đặt môi trường 38](#_Toc209609649)

[Hình 4.2 Giao diện bắt đầu Android Studio 39](#_Toc209609650)

[Hình 4.3 Cửa sổ chọn đường dẫn SDK Flutter 40](#_Toc209609651)

[Hình 4.5 Cửa sổ tạo project 40](#_Toc209609652)

[Hình 4.6 Giao diện project 41](#_Toc209609653)

[Hình 4.7 Giao diện generate file 41](#_Toc209609654)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1 Bảng các use case của hệ thống 24](#_Toc209609655)

[Bảng 3.2 Mô tả use case “Xóa ảnh/video” 24](#_Toc209609656)

[Bảng 3.3 Mô tả use case “Tách nền ảnh” 25](#_Toc209609657)

[Bảng 3.4 Mô tả use case “Kiểm tra ảnh gần giống/ trùng khớp” 25](#_Toc209609658)

[Bảng 4.1 Bảng kiểm thử chức năng Clean (dọn dẹp ảnh) 43](#_Toc209609659)

[Bảng 4.2 Bảng kiểm thử chức năng Memory (ký ức ảnh) 44](#_Toc209609660)

[Bảng 4.3 Bảng kiểm thử chức năng xóa nền ảnh 45](#_Toc209609661)

[Bảng 4.4 Bảng kiểm thử chức năng tìm ảnh tương tự 46](#_Toc209609662)

[Bảng 4.5 Bảng kiểm thử chức năng xóa ảnh 47](#_Toc209609663)

[Bảng 4.6 Bảng kiểm thử chức năng thống kê 47](#_Toc209609664)

[Bảng 4.7 Bảng kiểm thử chức năng cài đặt 48](#_Toc209609665)

[Bảng 4.8 Bảng kiểm thử chức năng quyền 49](#_Toc209609666)

# 

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu đề tài

Trong bối cảnh bùng nổ công nghệ số, điện thoại thông minh ngày càng trở thành công cụ lưu trữ và quản lý hình ảnh quan trọng của người dùng. Tuy nhiên, với thói quen chụp ảnh thường xuyên cùng sự phát triển của công nghệ camera độ phân giải cao, dung lượng bộ nhớ thiết bị nhanh chóng bị chiếm dụng, dẫn đến tình trạng quá tải lưu trữ. Điều này gây ra nhiều bất tiện như: khó quản lý album ảnh, giảm hiệu suất thiết bị, hoặc buộc người dùng phải xóa bớt ảnh một cách thủ công.

Để giải quyết vấn đề trên, đề tài “Xây dựng ứng dụng di động quản lý chất lượng ảnh và dung lượng lưu trữ thông minh” được đề xuất. Ứng dụng hướng đến việc quản lý ảnh thông minh như phân loại, phát hiện ảnh trùng lặp, gợi ý ảnh kém chất lượng để xóa.

Đề tài không chỉ mang tính thực tiễn cao mà còn góp phần hỗ trợ người dùng tiết kiệm bộ nhớ, nâng cao trải nghiệm khi sử dụng thiết bị di động. Đồng thời, việc ứng dụng các công nghệ hiện đại như Flutter để xây dựng ứng dụng cũng tạo điều kiện cho sản phẩm có khả năng phát triển đa nền tảng, mở rộng phạm vi sử dụng trên nhiều hệ điều hành di động phổ biến.

## Lý do chọn đề tài

Việc lưu trữ một số lượng lớn ảnh chất lượng cao thường dẫn đến tình trạng nhanh chóng đầy bộ nhớ, gây khó khăn cho quá trình quản lý dữ liệu, ảnh hưởng đến trải nghiệm sử dụng thiết bị. Bên cạnh đó, nhiều người dùng chưa có thói quen tối ưu hóa ảnh, dẫn đến việc tồn tại nhiều ảnh trùng lặp, ảnh kém chất lượng hoặc dung lượng quá lớn so với nhu cầu thực tế. Điều này không chỉ làm tốn dung lượng bộ nhớ mà còn ảnh hưởng đến hiệu suất hoạt động của thiết bị di động.

Xuất phát từ thực tiễn trên, em lựa chọn đề tài “Xây dựng ứng dụng di động quản lý chất lượng ảnh và dung lượng lưu trữ thông minh” với mong muốn phát triển một công cụ hỗ trợ người dùng:

#### Tìm kiếm và gợi ý ảnh trùng lặp hoặc gần giống để tối ưu dung lượng lưu trữ.

#### Cung cấp giao diện trực quan, dễ sử dụng, giúp người dùng quản lý bộ sưu tập ảnh một cách khoa học và hiệu quả.

#### Hỗ trợ tính năng xóa nền (background) giúp ảnh trở nên nổi bật, tiết kiệm dung lượng và phục vụ cho nhiều mục đích chỉnh sửa, thiết kế.

#### Cho phép người dùng vuốt để lựa chọn xóa ảnh, mang lại trải nghiệm thao tác nhanh chóng, thuận tiện và hiện đại.

#### Đề tài không chỉ mang tính ứng dụng thực tiễn cao, giải quyết trực tiếp nhu cầu của người dùng di động hiện nay, mà còn là cơ hội để em nghiên cứu và áp dụng các kỹ thuật hiện đại trong xử lý ảnh, tối ưu hóa dữ liệu và phát triển ứng dụng di động.

## Mục tiêu và phạm vi nghiên cứu

**Mục tiêu nghiên cứu:**

* Phân tích và hiểu rõ nhu cầu thực tế trong việc quản lý ảnh và tối ưu dung lượng lưu trữ trên thiết bị di động.
* Tìm hiểu và ứng dụng các kỹ thuật xử lý ảnh (nhận diện ảnh trùng lặp, xóa nền) kết hợp với Framework Flutter để xây dựng ứng dụng di động.
* Xây dựng một ứng dụng quản lý ảnh thông minh với các tính năng: phát hiện ảnh trùng lặp/gần giống, xóa nền ảnh, thao tác vuốt để xóa ảnh và tối ưu dung lượng lưu trữ.
* Đảm bảo giao diện người dùng trực quan, dễ sử dụng, phù hợp cho đa dạng đối tượng người dùng di động.

**Phạm vi nghiên cứu:**

Ứng dụng mới xây dựng này sẽ đáp ứng những chức năng chính sau:

* **Phát hiện ảnh trùng lặp và gần giống:** Sử dụng thuật toán xử lý ảnh để tìm kiếm, so sánh và gợi ý xóa ảnh trùng lặp hoặc gần giống, giúp người dùng tối ưu dung lượng lưu trữ.
* **Xóa nền ảnh (Background Removal):** Hỗ trợ người dùng loại bỏ nền ảnh tự động, phục vụ cho việc chỉnh sửa, thiết kế và tiết kiệm dung lượng.
* **Vuốt để lựa chọn xóa ảnh:** Cung cấp thao tác vuốt trực quan trên màn hình để chọn ảnh cần xóa, giúp quá trình quản lý ảnh nhanh chóng và tiện lợi.
* **Quản lý bộ sưu tập ảnh khoa học:** Cho phép sắp xếp, phân loại ảnh theo thời gian, Album giúp người dùng dễ dàng kiểm soát và tìm kiếm.
* **Giao diện thân thiện, dễ sử dụng:** Thiết kế giao diện trực quan, hiện đại, phù hợp với nhiều đối tượng người dùng di động.
* **Tích hợp Flutter và công nghệ xử lý ảnh:** Ứng dụng được phát triển bằng Flutter kết hợp với các thư viện xử lý ảnh hiện đại, đảm bảo khả năng chạy đa nền tảng và hiệu năng tốt.

## Ý nghĩa khoa học và thực tiễn của đề tài

**Ý nghĩa khoa học:**

* Ứng dụng khai thác kết hợp giữa khoa học dữ liệu đa phương tiện và công nghệ phát triển ứng dụng di động, tạo nên giải pháp có tính liên ngành, vừa mang tính nghiên cứu, vừa có khả năng triển khai thực tế.
* Góp phần bổ sung cơ sở khoa học cho các nghiên cứu liên quan đến quản lý dữ liệu lớn dạng ảnh/video trên thiết bị di động, trong bối cảnh dung lượng lưu trữ ngày càng tăng và nhu cầu tối ưu hóa ngày càng cao.

**Ý nghĩa thực tiễn:**

* Ứng dụng giúp người dùng quản lý thư viện ảnh và video một cách thông minh, tiết kiệm thời gian thay vì phải rà soát thủ công.
* Góp phần tối ưu hóa dung lượng lưu trữ trên thiết bị di động, đặc biệt hữu ích với người dùng có bộ nhớ giới hạn hoặc lưu trữ nhiều dữ liệu đa phương tiện.
* Cung cấp tính năng phát hiện ảnh trùng lặp, gần giống để người dùng dễ dàng quyết định giữ hay xóa, từ đó nâng cao hiệu quả sử dụng bộ nhớ.
* Tăng trải nghiệm người dùng nhờ giao diện trực quan (vuốt để xóa/giữ, gợi ý dọn dẹp, thống kê dung lượng tiết kiệm được).
* Có khả năng mở rộng, tích hợp vào các hệ thống quản lý đa phương tiện hoặc dịch vụ lưu trữ đám mây trong tương lai.

## 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.

## Đối tượng nghiên cứu

* Công cụ lập trình Android Studio.
* Framework Flutter.
* Các thuật toán và phương pháp xử lý ảnh phục vụ cho việc:
* Nhận diện ảnh trùng lặp, ảnh gần giống.
* Phát hiện và loại bỏ background ảnh.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. 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.

### 2.1.1. Hybrid App

Hybrid App [4] 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.

### 2.1.2. Native App

Native App [4] 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 thế 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).

### 2.1.3. Web App

Wep App [4] 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 tự bất kỳ đâu hoặc tải về thiết bị cá nhân.

- **Tiếc 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.

## 2.2. Công cụ lập trình Android Studio

### 2.2.1. Giới thiệu Android Studio

Android Studio [5] 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++, …

### 2.2.2. 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..

### 2.2.3. 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.

## 2.3. Ngôn ngữ lập trình Dart

Dart [8][9] 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.

## 2.4. Một số framework trong lập trình Flutter

### 2.4.1. Flutter SDK

Flutter SDK [8][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.

### 2.4.2. AngularDart

AngularDart [8][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.

### 2.4.3. StageXL

StageXL [8][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.

## 2.5. Hive Database

Hive là một hệ cơ sở dữ liệu NoSQL dạng **key–value**, được phát triển bằng ngôn ngữ Dart và tối ưu hóa cho ứng dụng Flutter. Hive không phải là cơ sở dữ liệu quan hệ, mà được thiết kế theo hướng lưu trữ dữ liệu dưới dạng các **Box** (tương tự như bảng), bên trong chứa các cặp **key–value**.

Hive phù hợp với các ứng dụng di động cần tốc độ truy xuất cao, thao tác dữ liệu nhanh chóng và khả năng hoạt động offline. Với việc sử dụng Hive, dữ liệu có thể được lưu trữ trực tiếp trên thiết bị của người dùng, không cần kết nối internet hay một server cơ sở dữ liệu riêng biệt.

**Cấu trúc của Hive:**

* **Box:** Tương tự như một bảng trong cơ sở dữ liệu quan hệ, mỗi Box là nơi chứa dữ liệu.
* **Key:** Được sử dụng để định danh duy nhất cho mỗi bản ghi trong Box. Key thường có kiểu số nguyên hoặc chuỗi.
* **Value:** Là dữ liệu được lưu trữ, có thể là kiểu dữ liệu cơ bản (int, String, bool, double…) hoặc các đối tượng phức tạp (object) khi được ánh xạ thông qua TypeAdapter.
* **TypeAdapter:** Là lớp trung gian cho phép Hive lưu trữ và truy xuất các đối tượng Dart tùy chỉnh.

**Cách thức hoạt động của Hive:**

Hive hoạt động bằng cách tạo ra các tệp dữ liệu nhị phân trên thiết bị di động. Khi ứng dụng cần truy cập dữ liệu, Hive sẽ mở các Box tương ứng và thực hiện các thao tác đọc/ghi thông qua key. Các thao tác này diễn ra hoàn toàn offline và có tốc độ rất nhanh do Hive được viết thuần bằng Dart, không cần cầu nối sang ngôn ngữ khác.

Hive hỗ trợ các thao tác cơ bản như:

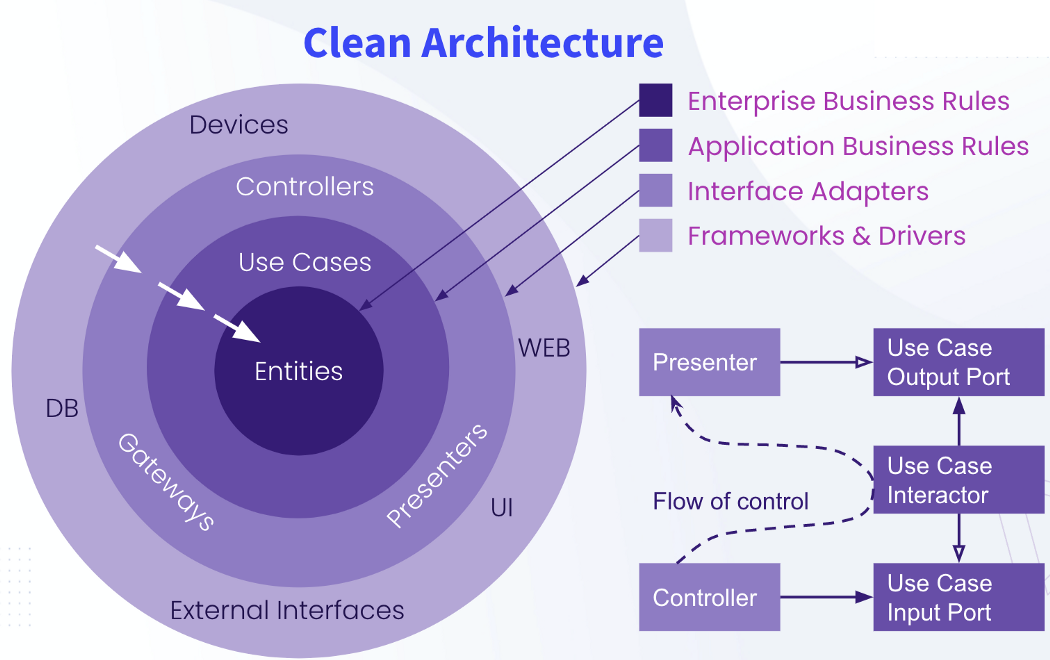
* put(key, value): thêm hoặc cập nhật dữ liệu.
* get(key): lấy dữ liệu theo key.
* delete(key): xóa dữ liệu theo key.
* clear(): xóa toàn bộ dữ liệu trong Box.

**Ưu điểm của Hive Database:**

* + **Hiệu năng cao:** Được viết bằng Dart thuần nên tốc độ truy xuất dữ liệu nhanh, đặc biệt phù hợp với Flutter.
  + **Nhẹ và dễ tích hợp:** Không yêu cầu cài đặt server hay cấu hình phức tạp.
  + **Hỗ trợ lưu trữ object:** Thông qua TypeAdapter, Hive có thể lưu trữ trực tiếp các đối tượng Dart.
  + **Hoạt động offline:** Dữ liệu lưu cục bộ, không cần internet.
  + **Đa nền tảng:** Hỗ trợ Android, iOS, macOS, Windows và Linux.

## 2.6. Tổng quan về mô hình Clean Architecture

Clean Architecture [6][7] là một mô hình kiến trúc phần mềm được Robert C. Martin (Uncle Bob) đề xuất, với mục tiêu xây dựng ứng dụng có cấu trúc rõ ràng, dễ mở rộng, dễ bảo trì, và độc lập với framework. Ý tưởng chính của Clean Architecture là tách biệt các tầng (layer) trong ứng dụng và thiết lập nguyên tắc phụ thuộc ngược (Dependency Rule): *“Mã nguồn trong tầng ngoài có thể phụ thuộc vào tầng trong, nhưng tầng trong không bao giờ được phụ thuộc vào tầng ngoài”*.



Hình 2.1 Cấu trúc Clean architecture

**Các tầng trong Clean Architecture**

Clean Architecture thường được mô tả dưới dạng các vòng tròn đồng tâm, từ trong ra ngoài:

* Entities (Domain / Business Rules):
* Chứa các đối tượng nghiệp vụ cốt lõi, các quy tắc, logic độc lập với framework hay cơ sở dữ liệu.
* Đây là phần “trái tim” của ứng dụng, có thể tái sử dụng cho nhiều hệ thống khác nhau.
* Use Cases (Application Rules):
  + Xác định cách các Entities được sử dụng để thực hiện các chức năng cụ thể của ứng dụng.
  + Chứa logic nghiệp vụ ứng dụng, điều phối luồng dữ liệu giữa các tầng.
* Interface Adapters (Presentation / Data Mapper):
* Chuyển đổi dữ liệu từ định dạng tiện cho tầng Use Cases/Entities sang định dạng tiện cho tầng ngoài (UI, Database, API).
* Bao gồm các Repository, Controller, Presenter, ViewModel.
* Frameworks & Drivers (Outer Layer):
* Là tầng ngoài cùng, chứa các công nghệ cụ thể như UI (Flutter widget), cơ sở dữ liệu (SQLite, Hive), web service (REST API, Firebase).
* Tầng này có thể thay thế mà không ảnh hưởng đến logic nghiệp vụ bên trong.

**Nguyên tắc Dependency Rule**

* + Tầng trong không phụ thuộc vào tầng ngoài.
  + Các tầng giao tiếp với nhau thông qua interface/abstraction.
  + Ví dụ: Use Case không gọi trực tiếp SQLite hay Hive, mà thông qua Repository interface; chi tiết triển khai (implementation) được định nghĩa ở tầng ngoài.

**Lợi ích của Clean Architecture**

* **Dễ bảo trì, mở rộng:** có thể thay đổi UI, database, hay framework mà không ảnh hưởng tới logic nghiệp vụ cốt lõi.
* **Tăng tính tái sử dụng:** Entities và Use Cases có thể dùng lại trong nhiều dự án.
* **Kiểm thử dễ dàng:** có thể viết unit test cho Use Cases và Entities mà không cần phụ thuộc vào UI hay database.
* **Tách biệt mối quan tâm (Separation of Concerns):** mỗi tầng chỉ tập trung vào nhiệm vụ riêng, giúp code gọn gàng và dễ quản lý.

## 2.7. Nguyên tắc SOLID trong phát triển phần mềm

**S**OLID là tập hợp 5 nguyên tắc thiết kế hướng đối tượng do Robert C. Martin (Uncle Bob) đề xuất. Đây là nền tảng quan trọng giúp việc xây dựng hệ thống phần mềm trở nên dễ mở rộng, dễ bảo trì và hạn chế tối đa lỗi phát sinh khi thay đổi yêu cầu. Các nguyên tắc bao gồm:

* **Single Responsibility Principle (SRP)** – Nguyên tắc trách nhiệm đơn
* Mỗi lớp (class) chỉ nên có một trách nhiệm duy nhất, và toàn bộ chức năng của lớp phải phục vụ cho trách nhiệm đó.
* Việc tuân thủ SRP giúp mã nguồn trở nên rõ ràng, dễ đọc và giảm rủi ro khi thay đổi, bởi thay đổi trong một trách nhiệm sẽ không ảnh hưởng đến các trách nhiệm khác.
* **Open/Closed Principle (OCP)** – Nguyên tắc đóng/mở
* Các module, lớp, hay hàm nên được mở rộng nhưng đóng với việc chỉnh sửa.
* Điều này có nghĩa là khi cần thêm chức năng mới, ta nên mở rộng hệ thống bằng cách kế thừa hoặc bổ sung thay vì thay đổi trực tiếp mã nguồn cũ.
* OCP giúp hệ thống dễ mở rộng mà không ảnh hưởng đến tính ổn định.
* **Liskov Substitution Principle (LSP)** – Nguyên tắc thay thế Liskov
* Các lớp con phải có thể thay thế lớp cha mà không làm thay đổi tính đúng đắn của chương trình.
* Nói cách khác, nếu một hàm chấp nhận đối tượng kiểu lớp cha, thì có thể truyền vào đối tượng lớp con mà chương trình vẫn hoạt động bình thường.
* Nguyên tắc này giúp đảm bảo tính kế thừa đúng nghĩa và tránh phá vỡ cấu trúc hướng đối tượng.
* **Interface Segregation Principle (ISP)** – Nguyên tắc phân tách giao diện
* Không nên ép một lớp phải triển khai những phương thức mà nó không cần.
* Thay vì thiết kế một interface lớn, ta nên tách thành nhiều interface nhỏ, phù hợp với nhu cầu sử dụng của từng lớp cụ thể.
* ISP giúp mã nguồn linh hoạt hơn, hạn chế sự phụ thuộc không cần thiết.
* **Dependency Inversion Principle (DIP)** – Nguyên tắc đảo ngược phụ thuộc
* Các module cấp cao không nên phụ thuộc trực tiếp vào các module cấp thấp, mà cả hai nên phụ thuộc vào các abstraction (interface hoặc lớp trừu tượng).
* DIP giúp hệ thống dễ dàng thay thế và mở rộng thành phần, đồng thời hỗ trợ việc viết unit test hiệu quả hơn.

**Ý nghĩa của SOLID**

* Giúp giảm sự phụ thuộc chặt chẽ giữa các thành phần trong hệ thống.
* Tăng khả năng mở rộng, tái sử dụng và bảo trì phần mềm.
* Hạn chế tình trạng code smell (mã nguồn khó hiểu, dễ lỗi) và giúp thiết kế sạch, bền vững hơn.

## 2.8. Thuật toán xử lý ảnh

### 2.8.1. Giới thiệu các thuật toán sử dụng

**Perceptual Hashing**

* **Mục đích:** Perceptual Hashing được dùng để tạo ra một “dấu vân tay” (fingerprint) cho ảnh. Nhờ fingerprint này, hệ thống có thể phát hiện và so sánh những ảnh giống hoặc gần giống nhau, ngay cả khi chúng đã được chỉnh sửa nhẹ như thay đổi kích thước, nén, xoay nhỏ hoặc điều chỉnh độ sáng tối.
* **Nguyên lý hoạt động:** Quá trình hoạt động bắt đầu từ việc chuẩn hóa ảnh về kích thước nhỏ, thường là 8x8 hoặc 16x16 pixel. Ảnh sau đó được chuyển sang thang xám nhằm loại bỏ nhiễu màu và giữ lại thông tin sáng tối. Từ ảnh chuẩn hóa, ta tính toán đặc trưng bằng các phương pháp như biến đổi cosin rời rạc (DCT) hoặc đơn giản là so sánh giá trị trung bình pixel. Cuối cùng, một chuỗi nhị phân được sinh ra để biểu diễn đặc trưng này. Những ảnh tương tự nhau sẽ có chuỗi hash gần giống nhau, trở thành cơ sở cho bước so sánh tiếp theo.

**Hamming Distance**

* **Mục đích:** Sau khi có fingerprint, hệ thống cần một thước đo để đánh giá mức độ giống nhau giữa các chuỗi hash. Hamming Distance được dùng để tính số bit khác nhau giữa hai chuỗi nhị phân có cùng độ dài.
* **Nguyên lý hoạt động:** Khoảng cách Hamming được tính bằng cách so sánh từng bit tại cùng vị trí trong hai chuỗi nhị phân có độ dài bằng nhau. Số vị trí khác nhau chính là giá trị khoảng cách. Khi giá trị này nhỏ, hai ảnh được coi là rất giống; ngược lại, khi khoảng cách lớn, chúng khác biệt rõ rệt. Nhờ đặc tính đơn giản và hiệu quả, Hamming Distance trở thành công cụ phổ biến trong so sánh ảnh dựa trên fingerprint.

**Graph Clustering**

* **Mục đích:** Graph Clustering nhằm gom nhóm các ảnh tương tự thành từng cụm, giúp việc quản lý ảnh dễ dàng hơn, đồng thời hỗ trợ người dùng phát hiện các tập ảnh gần giống hoặc ảnh chụp liên tiếp để loại bỏ bản thừa.
* **Nguyên lý hoạt động:** Quy trình được thực hiện bằng cách gán cho mỗi ảnh một chuỗi hash (fingerprint) sinh ra từ thuật toán Perceptual Hashing. Tất cả các ảnh trong tập dữ liệu được duyệt lần lượt. Với mỗi ảnh chưa được xét, hệ thống khởi tạo một cụm mới và coi ảnh đó là “hạt giống”. Các ảnh còn lại được so sánh với hạt giống dựa trên khoảng cách Hamming giữa các hash. Nếu khoảng cách nhỏ hơn hoặc bằng ngưỡng định trước (maxDistance), ảnh được xem là tương tự và được thêm vào cùng cụm. Kết quả cuối cùng là một tập các cụm, trong đó mỗi cụm chứa nhóm ảnh có mức độ tương đồng cao.

**Byte Comparison**

* **Mục đích:** Trong khi Graph Clustering chỉ phát hiện ảnh gần giống, hệ thống cũng cần một bước chuyên biệt để xác định các ảnh trùng khớp tuyệt đối. Đây là những ảnh giống hệt nhau đến từng byte dữ liệu, thường là do người dùng lưu nhiều bản sao giống nhau.
* **Nguyên lý hoạt động:** Quy trình hoạt động bằng cách duyệt qua toàn bộ tập ảnh. Với mỗi ảnh làm gốc, hệ thống so sánh dữ liệu nhị phân với các ảnh khác chưa xét. Nếu toàn bộ dãy byte của hai ảnh hoàn toàn giống nhau, chúng được coi là bản sao trùng lặp tuyệt đối và được gom vào cùng một cụm. Chỉ những cụm có từ hai ảnh trở lên mới được ghi nhận là kết quả. Phương pháp này đảm bảo độ chính xác tuyệt đối trong việc phát hiện duplicate, nhưng chỉ áp dụng cho trường hợp ảnh giống hệt, không nhận diện được các chỉnh sửa dù rất nhỏ.

### 2.8.2. Các bước trong quy trình xử lý ảnh

**A grid with black text

AI-generated content may be incorrect.**

Hình 2.2 Các bước trong quy trình xử lý ảnh

**Tiền xử lý ảnh**

Ảnh đầu vào sẽ được tiền xử lý để đưa chúng về cùng một dạng chuẩn. Thông thường, ảnh được thay đổi kích thước về một độ phân giải nhỏ cố định (ví dụ 8x8 hoặc 16x16) và chuyển sang thang xám nhằm loại bỏ ảnh hưởng của màu sắc, giữ lại những thông tin cốt lõi về hình dạng và độ sáng. Việc chuẩn hóa này giúp quá trình so sánh trở nên chính xác và ít bị nhiễu hơn.

**Sinh đặc trưng ảnh**

Toàn bộ ma trận điểm ảnh sau khi thu nhỏ sẽ được tính trung bình cường độ sáng. Sau đó, mỗi pixel trong ảnh được so sánh với giá trị trung bình này: nếu pixel sáng hơn hoặc bằng trung bình thì gán thành 1, ngược lại nếu tối hơn thì gán thành 0. Kết quả thu được là một chuỗi nhị phân (hash) đại diện cho đặc trưng ảnh.

**So sánh mức độ tương đồng**

Sử dụng Hamming Distance được sử dụng để đo số bit khác nhau giữa hai chuỗi hash. Khoảng cách Hamming càng nhỏ thì mức độ tương đồng giữa hai ảnh càng cao, ngược lại, khoảng cách lớn cho thấy hai ảnh khác biệt rõ rệt. Đây là bước nền tảng để phân biệt nhóm ảnh gần giống với ảnh hoàn toàn khác nhau.

**Gom nhóm ảnh**

Dựa trên thông tin tương đồng, các ảnh sẽ được gom thành cụm dựa trên độ giống nhau của **chuỗi hash**. Cách làm khá đơn giản: chọn một ảnh làm hạt giống, sau đó so sánh với các ảnh chưa được xử lý. Những ảnh nào có **khoảng cách Hamming** nhỏ hơn hoặc bằng ngưỡng cho phép sẽ được đưa vào cùng cụm và đánh dấu là đã dùng, tránh lặp lại ở các cụm khác. Khi không còn ảnh phù hợp, cụm được chốt và thuật toán tiếp tục với ảnh kế tiếp.

Cách gom này nhanh và dễ triển khai, tuy nhiên chỉ đảm bảo rằng mọi ảnh trong cụm đều gần với hạt giống, chứ không bắt buộc gần lẫn nhau theo bắc cầu. Điều đó có thể khiến một số ảnh đáng ra thuộc cùng cụm lại bị tách ra. Dù vậy, với dữ liệu không quá lớn, phương pháp này vẫn hiệu quả để nhận diện nhóm ảnh tương tự và giúp người dùng dễ quản lý, chọn giữ ảnh tiêu biểu hoặc loại bỏ các bản gần giống.

**Phát hiện ảnh trùng lặp tuyệt đối**

Để đảm bảo việc phát hiện **ảnh trùng lặp tuyệt đối**, em so sánh trực tiếp dữ liệu nhị phân của các tệp ảnh. Nếu toàn bộ byte dữ liệu hoặc chuỗi hash trùng khớp hoàn toàn, ta có thể khẳng định hai ảnh này giống nhau 100% và coi đó là duplicate. Đây là bước quan trọng nhằm loại bỏ các bản sao y hệt, vốn không thể phát hiện chỉ dựa trên Perceptual Hashing.

## 2.9. Phân tích thư viện xóa nền ảnh

### 2.9.1. Giới thiệu thư viện

Trong bối cảnh các ứng dụng di động hiện nay ngày càng cần đến khả năng xử lý ảnh thời gian thực, đặc biệt là chức năng tách nền (background removal) phục vụ cho thương mại điện tử, chỉnh sửa ảnh, hay mạng xã hội, việc sử dụng các dịch vụ xử lý ảnh trực tuyến đặt ra nhiều thách thức về quyền riêng tư và chi phí.

Thư viện image\_background\_remover [13] được phát hành trên nền tảng pub.dev, cung cấp khả năng xóa nền ảnh hoàn toàn ngoại tuyến trong Flutter bằng cách tích hợp mô hình học sâu định dạng ONNX và thực thi qua ONNX Runtime. Điểm nhấn của thư viện là không cần kết nối Internet, không phụ thuộc dịch vụ đám mây, đồng thời hỗ trợ đa nền tảng (Android, iOS, web, desktop) trong hệ sinh thái Flutter.

### 2.9.2. Kiến trúc và cơ chế hoạt động

Kiến trúc của image\_background\_remover được xây dựng theo mô hình pipeline gồm bốn bước:

* Nhập ảnh đầu vào: Ứng dụng Flutter cung cấp ảnh cần xử lý.
* Tiền xử lý: Ảnh được chuẩn hóa về kích thước, màu sắc và định dạng phù hợp với mô hình.
* Suy luận mô hình: Thư viện sử dụng ONNX Runtime để chạy mô hình phân đoạn ảnh (segmentation), cho ra mặt nạ foreground/background.
* Hậu xử lý: Mặt nạ được áp dụng lên ảnh gốc để loại bỏ nền và trả về ảnh kết quả.

Nhờ cơ chế chạy trực tiếp trên thiết bị, toàn bộ quy trình diễn ra ngoại tuyến, đảm bảo tính riêng tư và phản hồi nhanh.

### 2.9.3. Ưu điểm

* Hoạt động ngoại tuyến 100%: dữ liệu ảnh không rời khỏi thiết bị, bảo đảm an toàn và riêng tư.
* Tích hợp thuận tiện: dễ dàng cài đặt qua pub.dev, hỗ trợ sẵn ví dụ và tài liệu đi kèm.
* Không phụ thuộc dịch vụ ngoài: khác với các API thương mại (remove.bg, Slazzer…), không phát sinh chi phí theo lượt sử dụng.
* Đa nền tảng: phù hợp cho các ứng dụng Flutter trên nhiều hệ điều hành.

### 2.9.4. Nhược điểm

* Publisher chưa xác minh trên pub.dev, mức độ tin cậy cộng đồng còn hạn chế.
* Hiệu năng và bộ nhớ: việc chạy mô hình ONNX trực tiếp có thể chậm trên thiết bị cấu hình thấp, đặc biệt với ảnh độ phân giải cao.
* Thiếu tài liệu chi tiết: không công bố kiến trúc mô hình nền tảng (ví dụ backbone CNN hay transformer), khiến khó so sánh về mặt kỹ thuật với các mô hình khác.
* Chưa có benchmark chính thức: chưa có số liệu định lượng (tốc độ, độ chính xác) trên nhiều loại thiết bị.

### 2.9.5. So sánh với các mô hình đã có sẵn

#### 2.9.5.1. So sánh với rembg (U²-Net)

Thư viện rembg là một giải pháp Python phổ biến, hỗ trợ nhiều biến thể mô hình như u2net, u2netp, u2net\_human\_seg và silueta. Rembg cho thấy độ chính xác cao với nhiều loại ảnh khác nhau và có cộng đồng phát triển mạnh mẽ với nhiều mô hình được huấn luyện trước. Tuy nhiên, rembg tồn tại những hạn chế đáng kể khi áp dụng cho ứng dụng di động.

Thứ nhất, rembg chỉ hoạt động trên môi trường Python, đòi hỏi việc thiết lập một server backend riêng biệt cho ứng dụng di động. Điều này không chỉ làm tăng độ phức tạp của kiến trúc hệ thống mà còn gây ra độ trễ cao do các cuộc gọi mạng. Thứ hai, việc phụ thuộc vào kết nối internet làm giảm trải nghiệm người dùng, đặc biệt trong các tình huống mạng không ổn định.

Ngược lại, image\_background\_remover mang lại những lợi thế vượt trội khi hoạt động native trong môi trường Flutter. Thư viện này không yêu cầu thiết lập backend Python, cho phép xử lý thời gian thực trực tiếp trên thiết bị và tích hợp liền mạch với giao diện người dùng Flutter. Điều này đặc biệt quan trọng đối với ứng dụng quản lý ảnh cá nhân như PicTidy, nơi mà tính riêng tư và khả năng hoạt động offline là những yếu tố then chốt.

#### 2.9.5.2. So sánh với ML Kit Selfie Segmentation và Apple Vision

Google ML Kit Selfie Segmentation và Apple Vision Framework đại diện cho các giải pháp được tối ưu hóa bởi các tập đoàn công nghệ lớn. Cả hai framework này đều cho thấy hiệu suất xuất sắc với ảnh chân dung người, được tối ưu hóa cao bởi Google và Apple, có kích thước nhỏ gọn nhờ tận dụng các framework hệ thống có sẵn, và hỗ trợ xử lý thời gian thực cho camera.

Tuy nhiên, những giải pháp này tồn tại những hạn chế nghiêm trọng trong bối cảnh ứng dụng quản lý ảnh đa dạng. Đầu tiên, chúng chỉ giới hạn trong việc xử lý ảnh người, không thể xử lý hiệu quả các đối tượng khác như động vật, đồ vật hay phong cảnh. Thứ hai, sự phụ thuộc vào nền tảng cụ thể tạo ra vấn đề vendor lock-in, với ML Kit yêu cầu Google Play Services và Vision Framework chỉ hoạt động trên iOS.

Image\_background\_remover khắc phục những hạn chế này bằng cách cung cấp khả năng xử lý đa dạng các loại ảnh, từ chân dung người đến động vật, đồ vật và phong cảnh. Thư viện hoạt động độc lập trên cả Android và iOS với cùng một bộ code, đảm bảo chất lượng nhất quán trên mọi nền tảng mà không phụ thuộc vào các dịch vụ bên thứ ba.

#### 2.9.5.3. So sánh với các plugin Flutter khác

Trong hệ sinh thái Flutter, các plugin như background\_remover và flutter\_background\_remover chủ yếu hoạt động như các wrapper cho ML Kit và Apple Vision. Thư viện background\_remover package giới hạn chỉ trên Android và chất lượng xử lý phụ thuộc hoàn toàn vào ML Kit. Còn thư viện flutter\_background\_remover sử dụng Apple Vision trên iOS nhưng không đảm bảo tính nhất quán cross-platform và chỉ hiệu quả với ảnh người.

Image\_background\_remover nổi bật với API nhất quán trên các nền tảng và cung cấp ba tham số tùy chỉnh quan trọng: enhanceEdges để tăng cường phát hiện cạnh, smoothMask để làm mịn mask và threshold để điều chỉnh ngưỡng phân đoạn. Những đặc điểm nổi bật của thư viện bao gồm việc sử dụng ONNX Runtime để chạy mô hình độc lập với system APIs, đảm bảo tính nhất quán cross-platform, khả năng tùy chỉnh thông qua ba tham số chất lượng, tối ưu hóa bộ nhớ cho thiết bị di động và không phụ thuộc vào Google Play Services hay Apple frameworks.

### 2.9.6. Đánh giá thực nghiệm

#### 2.9.6.1. Hiệu suất thực tế

Do thư viện không công bố kết quả benchmark chính thức, nhóm tiến hành đánh giá thử nghiệm với bộ ảnh gồm nhiều loại đối tượng (chân dung, sản phẩm, cảnh vật). Kết quả sơ bộ cho thấy:

* Thư viện có khả năng tách nền tương đối ổn định với ảnh sản phẩm rõ nét.
* Với ảnh chân dung, độ chính xác ở vùng biên tóc chưa bằng ML Kit hoặc rembg U²-Net.
* Thời gian xử lý trung bình trên thiết bị tầm trung (Android 8GB RAM) vào khoảng 2–5 giây cho ảnh có độ phân giải 1080p. So sánh với các giải pháp khác, Remove.bg API đạt thời gian 1-3 giây nhưng yêu cầu kết nối internet và có chất lượng tốt nhất. ML Kit chỉ mất 0.5-1 giây nhưng giới hạn ở ảnh selfie và phụ thuộc nền tảng. Rembg backend có thời gian xử lý 3-8 giây do phải tính thêm độ trễ mạng và yêu cầu server.

#### 2.9.6.2. Tác động bộ nhớ

Về mặt kích thước, mô hình U²-Net ONNX chiếm khoảng 5MB, ONNX Runtime thêm 20MB, tổng cộng tác động khoảng 25MB đến kích thước ứng dụng. Trong quá trình chạy, bộ nhớ đỉnh có thể lên tới 500MB khi xử lý ảnh 1080p, trong khi bộ nhớ cơ bản khi mô hình được tải là khoảng 50MB.

### 2.9.7. Đánh giá tổng quan

#### 2.9.7.1. Lợi thế của image\_background\_remover

* Image\_background\_remover phù hợp với ứng dụng PicTidy vì năm lý do chính. Khả năng xử lý đa dạng các trường hợp sử dụng, đáp ứng nhu cầu của một ứng dụng quản lý ảnh cần xử lý mọi loại ảnh, không chỉ giới hạn ở ảnh selfie.
* Triết lý offline-first quan trọng đối với quyền riêng tư và trải nghiệm người dùng.
* Tính nhất quán nền tảng đảm bảo cùng chất lượng trên Android và iOS.
* Khả năng tùy chỉnh thông qua ba tham số cho phép điều chỉnh kết quả theo nhu cầu cụ thể.
* Độc lập với các hệ sinh thái Google/Apple, tránh vendor lock-in.

#### 2.9.7.2. Những đánh đổi được chấp nhận

Việc lựa chọn image\_background\_remover đi kèm với những đánh đổi được cân nhắc kỹ lưỡng. Kích thước ứng dụng tăng khoảng 25MB là một chi phí chấp nhận được trong bối cảnh thiết bị di động hiện đại có dung lượng lưu trữ lớn. Thời gian xử lý 2-5 giây được coi là hợp lý cho xử lý offline, đặc biệt khi so sánh với độ trễ và rủi ro của các giải pháp trực tuyến. Việc sử dụng bộ nhớ cao trong quá trình xử lý có thể được tối ưu hóa thông qua việc giải phóng mô hình sau khi sử dụng.

### 2.9.7. Kết luận

Nghiên cứu so sánh cho thấy rằng lựa chọn image\_background\_remover cho ứng dụng PicTidy là một quyết định hợp lý và được tính toán kỹ lưỡng. Thư viện này cung cấp sự cân bằng tối ưu giữa chất lượng xử lý và hiệu suất trên thiết bị di động, đồng thời đảm bảo tính độc lập và riêng tư quan trọng đối với một ứng dụng quản lý ảnh cá nhân.

Mặc dù tồn tại những đánh đổi về kích thước ứng dụng và thời gian xử lý, những lợi ích về tính đa dạng, độc lập nền tảng và khả năng hoạt động offline vượt trội hơn hẳn so với các giải pháp thay thế. Việc tích hợp sẵn với Flutter và khả năng tùy chỉnh thông qua các tham số điều chỉnh cũng góp phần làm cho image\_background\_remover trở thành lựa chọn phù hợp nhất cho kiến trúc và mục tiêu của dự án PicTidy.

Nghiên cứu này cũng chỉ ra rằng trong bối cảnh phát triển ứng dụng di động hiện đại, việc lựa chọn công nghệ không chỉ dựa trên hiệu suất thuần túy mà còn cần cân nhắc các yếu tố như tính độc lập, quyền riêng tư, trải nghiệm người dùng và khả năng bảo trì dài hạn.

# CHƯƠNG 3: PHÂN TÍCH YÊU CẦU PHẦN MỀM

## 3.1. Mô hình hóa chức năng

### 3.1.1. Các tác nhân hệ thống

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

### 3.1.2. Các use case hệ thống

Bảng 3.1 Bảng các use case của hệ thống

|  |  |
| --- | --- |
| **Tác nhân** | **Các ca sử dụng** |
| Người dùng | Xóa ảnh/video |
| Tách nền ảnh |
| Kiểm tra ảnh gần giống/ trùng khớp |

### 3.1.3. Biểu đồ use case

### 3.1.4. Đặc tả use case

#### 3.1.4.1. Use case xóa ảnh

Bảng 3.2 Mô tả use case “Xóa ảnh/video”

|  |  |  |
| --- | --- | --- |
| **Mô tả** | | Use case này cho phép người dùng thực hiện thao tác vuốt trái để đánh dấu ảnh sẽ xóa, sau đó xác nhận xóa khỏi thư viện của thiết bị |
| **Tác nhân** | | Người dùng |
| **Tiền điều kiện** | | Người dùng đã mở ứng dụng và danh sách ảnh/video được tải lên thành công |
| **Luồng sự kiện** | **Luồng sự kiện chính** | * + 1. Use case bắt đầu khi người dùng mở ứng dụng     2. Hệ thống hiển thị các album ảnh và các lựa chọn “Camera”, “Screenshot”, “Video”     3. Người dùng thao tác vuốt trái trên ảnh muốn xóa     4. Hệ thống đánh dấu ảnh sẽ xóa     5. Người dùng tiếp tục chọn các ảnh khác hoặc nhấn nút “Delete”     6. Hệ thống hiển thị lại danh sách ảnh đã chọn để người dùng xem lại và xác nhận     7. Người dùng nhấn nút “Delete”, hệ thống hiển thị popup xác nhận     8. Người dùng chọn “OK”, hệ thống xóa các ảnh đã chọn khỏi thư viện |
| **Luồng rẽ nhánh** | * Nếu người dùng chọn “Cancel” ở popup xác nhận, hệ thống hủy thao tác và giữ nguyên ảnh trong thư viện * Tại bất kỳ bước nào trong luồng cơ bản, nếu người dùng ấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn trước đó |
| **Hậu điều kiện** | | Không có |

#### 3.1.4.2. Use case tách nền ảnh

Bảng 3.3 Mô tả use case “Tách nền ảnh”

|  |  |  |
| --- | --- | --- |
| **Mô tả** | | Use case này cho phép người dùng xóa nền ảnh, tạo phiên bản PNG trong suốt để sử dụng riêng |
| **Tác nhân** | | Người dùng |
| **Tiền điều kiện** | | Người dùng đã mở ứng dụng và danh sách ảnh/video được tải lên thành công |
| **Luồng sự kiện** | **Luồng sự kiện chính** | * + 1. Use case bắt đầu khi người dùng mở ứng dụng     2. Hệ thống hiển thị các album ảnh và các lựa chọn “Camera”, “Screenshot”     3. Người dùng chọn chức năng “Remove BG”     4. Hệ thống gửi ảnh đến mô hình xử lý     5. Hệ thống trả về ảnh đã được tách nền     6. Người dùng bấm “Lưu”     7. Ảnh nền trong suốt được lưu vào thư viện thiết bị |
| **Luồng rẽ nhánh** | * Tại bất kỳ bước nào trong luồng cơ bản, nếu người dùng ấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn trước đó |
| **Hậu điều kiện** | | Không có |

#### 3.1.4.3. Use case kiểm tra ảnh gần giống/ trùng khớp

Bảng 3.4 Mô tả use case “Kiểm tra ảnh gần giống/ trùng khớp”

|  |  |  |
| --- | --- | --- |
| **Mô tả** | | Use case này cho phép người dùng phát hiện và quản lý các ảnh trùng hoặc gần giống nhau trong thư viện. |
| **Tác nhân** | | Người dùng |
| **Tiền điều kiện** | | Người dùng đã mở ứng dụng và danh sách ảnh/video được tải lên thành công |
| **Luồng sự kiện** | **Luồng sự kiện chính** | * + 1. Use case bắt đầu khi người dùng mở ứng dụng     2. Người dùng chọn chức năng “Kiểm tra ảnh trùng/gần giống”     3. Hệ thống quét toàn bộ ảnh, tính toán pHash và so sánh độ tương tự     4. Hệ thống nhóm các ảnh giống/trùng vào từng cụm     5. Danh sách cụm được hiển thị để người dùng chọn giữ hoặc xóa     6. Người dùng xác nhận, hệ thống hiển thị pop-up xác nhận     7. Người dùng chọn “OK”, hệ thống xóa các ảnh đã chọn khỏi thư viện |
| **Luồng rẽ nhánh** | * Tại bất kỳ bước nào trong luồng cơ bản, nếu người dùng ấn nút “Back” thì hệ thống sẽ hủy bỏ thao tác vừa chọn và quay trở về màn trước đó |
| **Hậu điều kiện** | | Không có |

### 3.1.5. Biểu đồ chi tiết Use case

#### 3.1.5.1. Use case xóa ảnh/video

**Biểu đồ lớp**



**Biểu đồ trình tự**



#### 3.1.5.2. Use case tách nền ảnh

**Biểu đồ lớp**



**Biểu đồ trình tự**



#### 3.1.5.3. Use case kiểm tra ảnh gần giống/ trùng khớp

**Biểu đồ lớp**



**Biểu đồ trình tự**



## 3.2. Thiết kế giao diện

### 3.2.1. Giao diện Home



Hình 3.1 Giao diện màn Home

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3.2 Giao diện màn Memory

### 3.2.2. Giao diện Xóa ảnh/video

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3.3 Giao diện màn xóa ảnh

A hand reaching out to a dog

AI-generated content may be incorrect.

Hình 3.4 Giao diện màn xóa video

A screenshot of a video game

AI-generated content may be incorrect.

Hình 3.5 Giao diện màn xem lại ảnh

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3.6 Giao diện màn xác nhận xóa

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3.7 Giao diện màn xóa thành công

### 3.2.3. Giao diện tách nền ảnh

A screenshot of a cellphone

AI-generated content may be incorrect.

Hình 3.8 Giao diện màn trước khi tách nền ảnh

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 3.9 Giao diện màn sau khi tách nền ảnh

### 3.2.4. Giao diện kiểm tra ảnh gần giống/ trùng khớp

A screenshot of a cellphone

AI-generated content may be incorrect.

Hình 3.10 Giao diện màn ảnh gần giống

A collage of a person

AI-generated content may be incorrect.

Hình 3.11 Giao diện màn ảnh trùng khớp

### 3.2.5. Giao diện thống kê

A screenshot of a phone

AI-generated content may be incorrect.

Hình 3.12 Giao diện màn thống kê

# CHƯƠNG 4: KẾT QUẢ VÀ ĐÁNH GIÁ

## 4.1. 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

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 4.1 Trạng thái cài đặt môi trường

* Cài đặt plugin Flutter và Dart trong Android Studio

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2 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”.

**Cài đặt Flutter Intl plugin**

* Mở Android Studio.
* Chọn Plugin
* Tìm kiếm “Flutter Intl” trong khung tìm kiếm extensions, chọn Flutter Intl trong danh sách, sau đó nhấn “Install”.

## 4.2. 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.

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 4.3 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:

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 4.5 Cửa sổ tạo project

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 4.6 Giao diện project

* Bước 3: Mở terminal và chạy câu lệnh “dart run build\_runner build -d” để generate các file cần thiết của project

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 4.7 Giao diện generate file

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

## 4.3. Kết quả thu được

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

### 4.3.1.  Dọn dẹp ảnh thông minh (Clean)

* **Phân loại ảnh tự động**: Ứng dụng tự động phân loại ảnh theo các danh mục khác nhau
* **Tìm ảnh tương tự**: Phát hiện và nhóm các ảnh giống nhau hoặc tương tự để người dùng dễ dàng xóa bớt
* **Tìm ảnh trùng lặp**: Xác định các ảnh trùng lặp chính xác để tiết kiệm dung lượng
* **Quản lý album**: Hiển thị tất cả album ảnh với số lượng ảnh trong mỗi album
* **Xóa ảnh hàng loạt**: Cho phép chọn và xóa nhiều ảnh cùng lúc

### 4.3.2.  Xóa nền ảnh tự động (Background Removal)

* **Xóa nền AI**: Sử dụng công nghệ AI để tự động xóa nền ảnh chỉ trong vài giây
* **Giao diện trượt**: Giao diện trượt ngang để xem và xử lý ảnh một cách trực quan
* **Tải xuống ảnh đã xử lý**: Cho phép tải xuống ảnh đã xóa nền
* **Hoàn tác**: Có thể hoàn tác việc xóa nền nếu không hài lòng

### 4.3.3. Ký ức ảnh (Memory)

* **Xem ảnh theo thời gian**: Hiển thị ảnh được sắp xếp theo năm, tháng
* **Ảnh ngẫu nhiên**: Tính năng xem ảnh ngẫu nhiên để khám phá ký ức
* **Ảnh gần đây**: Hiển thị các ảnh được chụp gần đây
* **Giao diện lịch**: Sử dụng lịch để điều hướng theo thời gian

### 4.3.4. Thống kê chi tiết (Statistics)

* **Thống kê dung lượng**: Hiển thị dung lượng ảnh/video đã xóa
* **Lịch thống kê**: Giao diện lịch để xem thống kê theo ngày
* **Báo cáo chi tiết**: Thống kê chi tiết về việc dọn dẹp

### 4.3.5. Cài đặt và tùy chọn (Settings)

* **Đa ngôn ngữ**: Hỗ trợ 15+ ngôn ngữ (Tiếng Việt, English, 中文, 日本語, 한국어, Español, Français, Deutsch, Italiano, Português, Русский, Türkçe, हिन्दी, Bahasa Indonesia)
* **Yêu thích**: Quản lý ảnh yêu thích
* **Thống kê cá nhân**: Xem thống kê cá nhân về việc sử dụng ứng dụng

## 4.4. Kết quả kiểm thử

### 4.4.1. 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.

### 4.4.2. Kết quả kiểm thử

* Tỉ lệ test case đạt (passed): 100%.
* Tỉ 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, Android 13, Android 14, Android 15, Android 16).

**Kết quả chi tiết:**

Bảng 4.1 Bảng kiểm thử chức năng Clean (dọn dẹp ảnh)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC001 | Hiển thị danh sách album ảnh | Kiểm tra việc load và hiển thị các album ảnh từ thiết bị | Thiết bị có 5 album ảnh (Camera, Screenshots, Downloads, DCIM, WhatsApp) | Hiển thị đầy đủ 5 album với tên và số lượng ảnh chính xác | **Đạt** |
| TC002 | Hiển thị số lượng ảnh trong album | Kiểm tra tính chính xác của số lượng ảnh | Album Camera có 150 ảnh, Screenshots có 25 ảnh | Hiển thị "150" cho Camera, "25" cho Screenshots | **Đạt** |
| TC003 | Phân loại ảnh theo loại | Kiểm tra việc phân loại ảnh thành Photo, Video, Screenshot | 100 ảnh thường, 20 video, 30 screenshot | Hiển thị 3 component riêng biệt với số lượng đúng | **Đạt** |
| TC004 | Hiển thị ảnh tương tự | Kiểm tra chức năng tìm ảnh tương tự | Album có 10 ảnh, trong đó 3 ảnh giống nhau | Hiển thị 1 nhóm ảnh tương tự với 3 ảnh | **Đạt** |
| TC005 | Hiển thị ảnh trùng lặp | Kiểm tra chức năng tìm ảnh trùng lặp | Album có 5 ảnh trùng lặp hoàn toàn | Hiển thị 1 nhóm ảnh trùng lặp với 5 ảnh | **Đạt** |
| TC006 | Navigate đến trang Similar | Kiểm tra điều hướng đến trang ảnh tương tự | Click vào component Similar | Chuyển đến trang SimilarPage | **Đạt** |
| TC007 | Navigate đến trang Video | Kiểm tra điều hướng đến trang video | Click vào component Video | Chuyển đến trang SlideWipePage với danh sách video | **Đạt** |
| TC008 | Navigate đến trang Screenshot | Kiểm tra điều hướng đến trang Screenshot | Click vào component Screenshot | Chuyển đến trang SlideWipePage với danh sách screenshot | **Đạt** |

Bảng 4.2 Bảng kiểm thử chức năng Memory (ký ức ảnh)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC009 | Hiển thị ảnh theo năm | Kiểm tra việc nhóm ảnh theo năm | Ảnh từ 2020, 2021, 2022, 2023, 2024 | Hiển thị 5 nhóm năm riêng biệt | **Đạt** |
| TC010 | Hiển thị ảnh theo tháng trong năm | Kiểm tra việc nhóm ảnh theo tháng | Ảnh từ tháng 1-12/2024 | Hiển thị 12 tháng trong năm 2024 | **Đạt** |
| TC011 | Tính năng Recent | Kiểm tra hiển thị ảnh gần đây | Ảnh chụp trong 7 ngày qua | Hiển thị ảnh được sắp xếp theo thời gian gần nhất | **Đạt** |
| TC012 | Tính năng Random | Kiểm tra hiển thị ảnh ngẫu nhiên | Click nút Random | Hiển thị ảnh ngẫu nhiên từ toàn bộ bộ sưu tập | **Đạt** |
| TC013 | Navigate đến ảnh theo năm | Kiểm tra điều hướng đến ảnh của năm cụ thể | Click vào năm 2023 | Chuyển đến trang SlideWipePage với ảnh năm 2023 | **Đạt** |
| TC014 | Navigate đến ảnh theo tháng | Kiểm tra điều hướng đến ảnh của tháng cụ thể | Click vào tháng 6/2024 | Chuyển đến trang SlideWipePage với ảnh tháng 6/2024 | **Đạt** |
| TC015 | Hiển thị số lượng ảnh theo thời gian | Kiểm tra số lượng ảnh hiển thị chính xác | Tháng 6/2024 có 45 ảnh | Hiển thị "45" ảnh cho tháng 6/2024 | **Đạt** |

Bảng 4.3 Bảng kiểm thử chức năng xóa nền ảnh

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC016 | Khởi tạo Background Remover | Kiểm tra việc khởi tạo công cụ xóa nền | Mở trang SlideWipePage | BackgroundRemover được khởi tạo thành công | **Đạt** |
| TC017 | Xóa nền ảnh thành công | Kiểm tra chức năng xóa nền ảnh | Ảnh có nền phức tạp | Hiển thị loading, sau đó hiển thị ảnh đã xóa nền | **Đạt** |
| TC018 | Hiển thị loading khi xóa nền | Kiểm tra UI loading | Click nút xóa nền | Hiển thị animation skeleton và text "Removing background..." | **Đạt** |
| TC019 | Tải xuống ảnh đã xóa nền | Kiểm tra chức năng tải xuống | Ảnh đã được xóa nền | Cho phép tải xuống ảnh đã xử lý | **Đạt** |
| TC020 | Hoàn tác việc xóa nền | Kiểm tra chức năng hoàn tác | Ảnh đã xóa nền, click Undo | Quay lại ảnh gốc chưa xóa nền | **Đạt** |
| TC021 | Xóa nền ảnh không thành công | Kiểm tra xử lý lỗi | Ảnh bị lỗi hoặc không hỗ trợ | Hiển thị thông báo lỗi phù hợp | **Đạt** |
| TC022 | Xóa nền nhiều ảnh liên tiếp | Kiểm tra hiệu suất | Swipe qua nhiều ảnh và xóa nền | Xử lý được nhiều ảnh mà không bị crash | **Đạt** |

Bảng 4.4 Bảng kiểm thử chức năng tìm ảnh tương tự

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC023 | Hiển thị danh sách ảnh tương tự | Kiểm tra việc load ảnh tương tự | Album có 50 ảnh, 10 ảnh tương tự | Hiển thị 1 nhóm ảnh tương tự với 10 ảnh | **Đạt** |
| TC024 | Hiển thị danh sách ảnh trùng lặp | Kiểm tra việc load ảnh trùng lặp | Album có 5 ảnh trùng lặp | Hiển thị 1 nhóm ảnh trùng lặp với 5 ảnh | **Đạt** |
| TC025 | Chuyển đổi tab Similar/Exact | Kiểm tra chức năng toggle | Click toggle giữa Similar và Exact | Chuyển đổi danh sách hiển thị tương ứng | **Đạt** |
| TC026 | Chọn ảnh để xóa | Kiểm tra chức năng chọn ảnh | Click vào ảnh trong danh sách | Ảnh được đánh dấu để xóa với border đỏ | **Đạt** |
| TC027 | Bỏ chọn ảnh | Kiểm tra chức năng bỏ chọn | Click lại vào ảnh đã chọn | Ảnh bỏ đánh dấu xóa | **Đạt** |
| TC028 | Hiển thị số lượng ảnh đã chọn | Kiểm tra counter | Chọn 3 ảnh để xóa | Hiển thị "Delete 3 photo" | **Đạt** |
| TC029 | Hiển thị dung lượng ảnh đã chọn | Kiểm tra tính toán dung lượng | Chọn 5 ảnh tổng 25MB | Hiển thị "25.0 MB" | **Đạt** |
| TC030 | Xác nhận xóa ảnh | Kiểm tra dialog xác nhận | Click nút Delete với 3 ảnh đã chọn | Hiển thị dialog xác nhận với thông tin chính xác | **Đạt** |

Bảng 4.5 Bảng kiểm thử chức năng xóa ảnh

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC031 | Xóa ảnh thành công | Kiểm tra xóa ảnh cơ bản | Chọn 1 ảnh và xác nhận xóa | Ảnh bị xóa khỏi thiết bị | **Đạt** |
| TC032 | Xóa nhiều ảnh cùng lúc | Kiểm tra xóa hàng loạt | Chọn 5 ảnh và xác nhận xóa | Tất cả 5 ảnh bị xóa | **Đạt** |
| TC033 | Xóa ảnh trong SlideWipe | Kiểm tra xóa ảnh khi swipe | Swipe left để xóa ảnh | Ảnh được đánh dấu để xóa | **Đạt** |
| TC034 | Hoàn tác việc xóa | Kiểm tra chức năng Undo | Xóa ảnh, sau đó click Undo | Ảnh được khôi phục | **Đạt** |
| TC035 | Xóa ảnh với API level < 29 | Kiểm tra xử lý cho Android cũ | Thiết bị Android API < 29 | Hiển thị dialog yêu cầu quyền xóa | **Đạt** |
| TC036 | Xóa ảnh với API level >= 29 | Kiểm tra xử lý cho Android mới | Thiết bị Android API >= 29 | Xóa trực tiếp không cần dialog | **Đạt** |
| TC037 | Xóa video | Kiểm tra xóa video | Chọn 1 video và xác nhận xóa | Video bị xóa khỏi thiết bị | **Đạt** |
| TC038 | Hiển thị thông báo xóa thành công | Kiểm tra feedback | Sau khi xóa ảnh | Hiển thị thông báo ảnh đã xóa | **Đạt** |

Bảng 4.6 Bảng kiểm thử chức năng thống kê

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC039 | Hiển thị lịch thống kê | Kiểm tra giao diện lịch | Mở trang My Statistics | Hiển thị lịch với các ngày có hoạt động | **Đạt** |
| TC040 | Chọn khoảng thời gian | Kiểm tra chọn range date | Chọn từ 1/1/2024 đến 31/1/2024 | Range được highlight trên lịch | **Đạt** |
| TC041 | Hiển thị thống kê theo ngày | Kiểm tra thống kê chi tiết | Click vào ngày 15/1/2024 | Hiển thị số ảnh đã xóa trong ngày | **Đạt** |
| TC042 | Navigate đến thống kê chi tiết | Kiểm tra điều hướng | Click nút Next sau khi chọn range | Chuyển đến trang My Statistics Detail | **Đạt** |
| TC043 | Hiển thị tổng dung lượng đã giải phóng | Kiểm tra tính toán tổng | Đã xóa 100 ảnh tổng 500MB | Hiển thị "500.0 MB freed" | **Đạt** |
| TC044 | Hiển thị số ảnh đã xóa | Kiểm tra counter | Đã xóa 50 ảnh | Hiển thị "50 photos deleted" | **Đạt** |

Bảng 4.7 Bảng kiểm thử chức năng cài đặt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC045 | Hiển thị danh sách cài đặt | Kiểm tra menu cài đặt | Mở trang More | Hiển thị Favorites, My Statistics, Languages | **Đạt** |
| TC046 | Navigate đến Favorites | Kiểm tra điều hướng | Click Favorites | Chuyển đến trang quản lý ảnh yêu thích | **Đạt** |
| TC047 | Navigate đến My Statistics | Kiểm tra điều hướng | Click My Statistics | Chuyển đến trang My Statistics | **Đạt** |
| TC048 | Navigate đến Languages | Kiểm tra điều hướng | Click Languages | Chuyển đến trang chọn ngôn ngữ | **Đạt** |
| TC049 | Chọn ngôn ngữ | Kiểm tra đổi ngôn ngữ | Chọn "Tiếng Việt" | Giao diện chuyển sang tiếng Việt | **Đạt** |
| TC050 | Chọn ngôn ngữ English | Kiểm tra đổi ngôn ngữ | Chọn "English" | Giao diện chuyển sang tiếng Anh | **Đạt** |

Bảng 4.8 Bảng kiểm thử chức năng quyền

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| TC056 | Chuyển đổi tab Clean | Kiểm tra bottom navigation | Click tab Clean | Hiển thị trang Clean | **Đạt** |
| TC057 | Chuyển đổi tab Memory | Kiểm tra bottom navigation | Click tab Memory | Hiển thị trang Memory | **Đạt** |
| TC058 | Chuyển đổi tab Settings | Kiểm tra bottom navigation | Click tab Settings | Hiển thị trang More | **Đạt** |
| TC059 | Back navigation | Kiểm tra nút back | Click nút back | Quay lại trang trước | **Đạt** |

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong quá trình thực hiện đề tài “Xây dựng ứng dụng di động quản lý chất lượng ảnh và dung lượng lưu trữ thông minh”, em đã áp dụng Clean Architecture kết hợp BLoC cho tầng trình bày. Cấu trúc hệ thống gồm Domain (entities, use cases, interfaces repository), Data (triển khai repository, truy cập dữ liệu cục bộ/thiết bị), và Presentation (UI, state management bằng BLoC, điều hướng). Em đã hoàn thành khảo sát và phân tích yêu cầu, đặc tả hệ thống, thiết kế luồng dữ liệu và lưu trữ cục bộ (SharedPreferences, Hive), hiểu và sử dụng ngôn ngữ lập trình Dart, xây dựng các chức năng chính gồm phân loại ảnh/video, tìm ảnh tương tự/trùng lặp, xóa nền ảnh bằng AI, trình chiếu và dọn dẹp theo cơ chế vuốt, và thống kê. Đồng thời, em đã lập kế hoạch và thực hiện kiểm thử hệ thống (chức năng, UI/UX, hiệu năng, quyền truy cập), cùng quản lý vòng đời ứng dụng theo chuẩn Clean Architecture.

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 tuỳ chỉnh giao diện, cho phép chọn màu sắc và chế độ sáng/tối theo sở thích cá nhân, tạo ra trải nghiệm cá nhân hóa và thích nghi với điều kiện ánh sáng.
* Nhận phản hồi từ người dùng: Tích hợp hệ thống góp ý và đánh giá trong ứng dụng để thu thập ý kiến, từ đó cải thiện trải nghiệm và tăng tính tương tác.
* Phát triển thành một nền tảng quản lý ảnh: Cho phép người dùng chia sẻ ảnh đã xử lý (xóa nền, dọn dẹp) để mọi người cùng chiêm ngưỡng và thảo luận, hướng đến mô hình cộng đồng riêng.

# 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]. Amela.vn (2023), Phân biệt Native App, Web App và Hybrid App. Nên dùng loại app nào?, Lấy từ: <https://amela.vn>

[5]. OpenPlanning, 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>

[6]. Robert C. Martin (2017), Clean Architecture: A Craftsman’s Guide to Software Structure and Design, Pearson Education, Boston.

[7]. Robert C. Martin (2012), The Clean Architecture, The Clean Coder Blog, Lấy từ: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>

[8]. Google (2025), Flutter documentation, Lấy từ: <https://docs.flutter.dev>

[9]. Dart Team (2025), Effective Dart: Style, Documentation, Usage, Lấy từ: <https://dart.dev/guides/language/effective-dart>

[10]. Felix Angelov et al. (2025), flutter\_bloc – State Management for Flutter, Lấy từ: <https://pub.dev/packages/flutter_bloc>

[11]. BLoC Library (2025), bloclibrary.dev – Documentation, Lấy từ: <https://bloclibrary.dev/#/>

[12]. Caijinglong (2025), photo\_manager – Manage photos and videos on iOS/Android, Lấy từ: <https://pub.dev/packages/photo_manager>

[13]. Image Background Remover (2025), image\_background\_remover – Remove image background on-device, Lấy từ: <https://pub.dev/packages/image_background_remover>

[14]. Google (2025), Scoped Storage and Media Deletion (Android 10+), Android Developers, Lấy từ: <https://developer.android.com/training/data-storage/shared/media>

[15]. W3schools, SQL Introduction (w3schools.com)