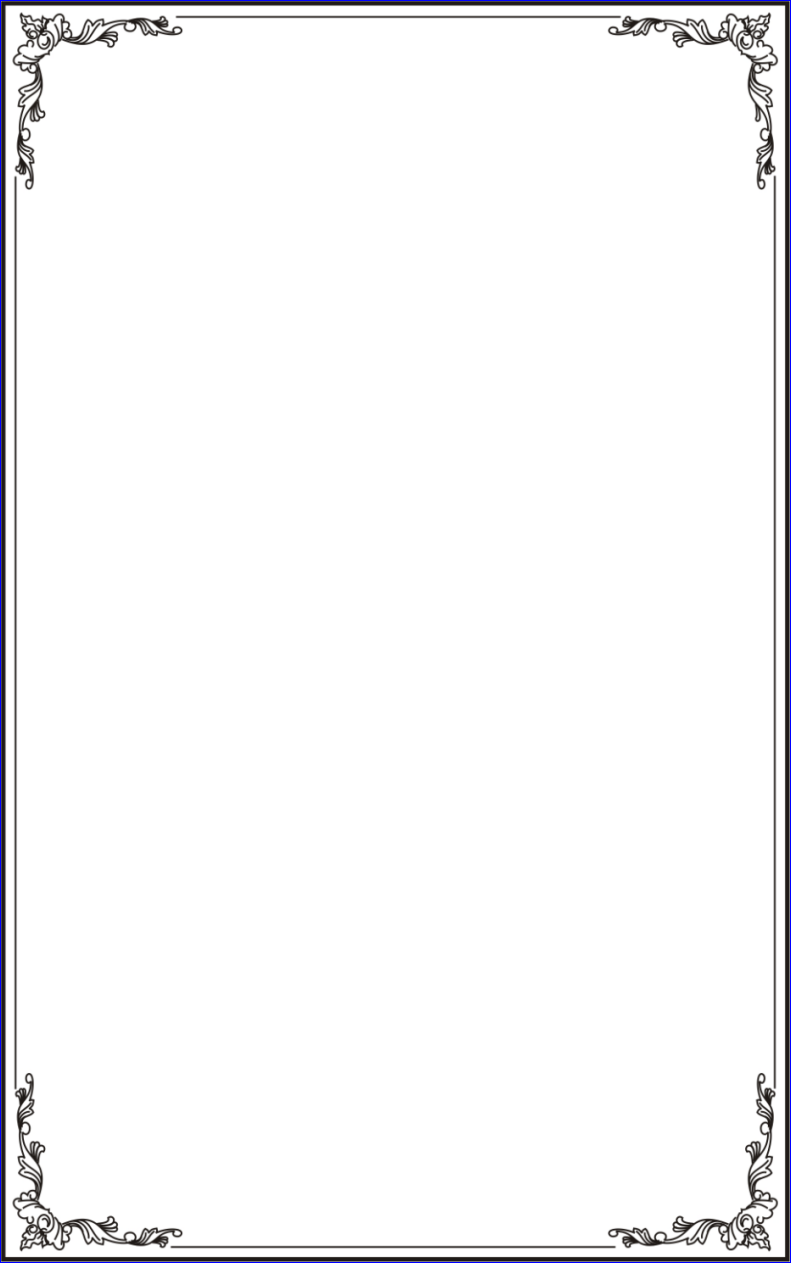
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------------------------**

**BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG CHIA SẺ HÌNH ẢNH TRỰC TUYẾN VỚI ANDROID**

**Giảng viên hướng dẫn:** ThS. Nguyễn Thái Cường

**Nhóm:** 7

**Sinh viên thực hiện:** Nguyễn Minh Quân – 2021605550

Trần Văn Tài – 2021604872

Ninh Thái Lâm – 2021605404

Lục Văn Thế – 2021605606

Cam Tuấn Nghĩa – 2021605105

**Lớp:** 20241IT6029004 **Khóa:** 16

**Hà Nội - Năm 2024**

LỜI MỞ ĐẦU

Để hoàn thành học phần Phát triển ứng dụng trên thiết bị di động, chúng em đã nhận được rất nhiều sự giúp đỡ và hướng dẫn tận tình của thầy Nguyễn Thái Cường. Các thành viên trong nhóm đã nỗ lực hết sức để hoàn thành bài tập lớn một cách tốt nhất, tuy nhiên vì kiến thức và kinh nghiệm còn hạn chế nên khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được những lời đóng góp, nhận xét từ thầy để có thể rút ra được bài học, giúp bài tập lớn được hoàn thiện hơn nữa.

Chúng em xin chân thành cảm ơn!

1. Lý do chọn đề tài

Trong thời đại số hóa ngày nay, việc chia sẻ những câu chuyện, những trải nghiệm của bản thân thông qua hình ảnh ngày càng trở nên phổ biến. Cùng với sự phát triển mạnh mẽ của công nghệ di động, việc chia sẻ hình ảnh trực tuyến đang dần trở thành một phần của cuộc sống hàng ngày. Do đó, việc tạo ra các ứng dụng chia sẻ hình ảnh trên nền tảng di động như Android cũng dần trở nên cần thiết, đáp ứng nhu cầu ngày càng tăng cao.

Ứng dụng chia sẻ hình ảnh không chỉ giúp người dùng lưu giữ và chia sẻ những khoảnh khắc đáng nhớ của mình mà còn tạo ra một cộng đồng nơi mọi người có thể kết nối và tương tác với nhau thông qua hình ảnh. Điều này thúc đẩy chúng ta cần tìm hiểu sâu hơn về cách mà Android Studio hoạt động và tại sao nó trở thành một trong những công cụ phổ biến nhất cho việc phát triển ứng dụng di động.

Nắm vững về Android Studio không chỉ là vấn đề lý thuyết mà còn liên quan đến phát triển kỹ năng nghiên cứu và phân tích, giúp chúng ta hiểu rõ hơn về cơ chế hoạt động của các ứng dụng di động trong thế giới kỹ thuật số ngày nay. Vì vậy, xuất phát từ nhu cầu thực tiễn, chúng em đã chọn đề tài báo cáo bài tập lớn: ***“Xây dựng ứng dụng chia sẻ hình ảnh trực tuyến với Android”***

1. Mục đích

Tập trung tìm hiểu về việc xây dựng ứng dụng chia sẻ hình ảnh trực tuyến với Android Studio, áp dụng thực tiễn những kiến thức đã học để tạo ra một nền tảng cho phép người dùng chia sẻ hình ảnh cá nhân và tương tác trong cộng đồng trực tuyến.

1. Bố cục chính

Ngoài phần mở đầu và kết luận, báo cáo được chia thành 3 chương:

* Chương 1: Tổng quan về đề tài
* Chương 2: Phân tích và thiết kế chương trình
* Chương 3: Thực nghiệm và đánh giá

1. Phương pháp

* Tìm kiếm thông tin, thu thập tài liệu từ các nguồn uy tín như sách về phát triển ứng dụng Android, bài báo khoa học từ các hội thảo về phát triển phần mềm.
* Tìm hiểu, vận dụng và kế thừa một số các thư viện và công cụ đã có trên Internet hoặc tài liệu liên quan để xây dựng ứng dụng chia sẻ hình ảnh trên nền tảng Android.
* Áp dụng các kiến thức đã được học từ học phần Phát triển ứng dụng trên thiết bị di động

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc155312640)

[1. Lý do chọn đề tài 2](#_Toc155312641)

[2. Mục đích 2](#_Toc155312642)

[3. Bố cục chính 2](#_Toc155312643)

[4. Phương pháp 3](#_Toc155312644)

[MỤC LỤC 4](#_Toc155312645)

[Danh mục hình ảnh 5](#_Toc155312646)

[Danh mục bảng 6](#_Toc155312647)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 7](#_Toc155312648)

[1.1. Giới thiệu chung 7](#_Toc155312649)

[1.2. Cơ sở lý thuyết 8](#_Toc155312650)

[1.2.1 Android Studio 8](#_Toc155312651)

[1.2.2 Java Android 14](#_Toc155312652)

[1.2.3 Camera API 16](#_Toc155312653)

[1.2.4 Firebase Authentication 18](#_Toc155312654)

[1.2.5 Firebase Firestore 20](#_Toc155312655)

[1.2.6 Mô hình MVVM 21](#_Toc155312656)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 23](#_Toc155312657)

[2.1 Yêu cầu chức năng 23](#_Toc155312658)

[2.2 Yêu cầu phi chức năng 24](#_Toc155312659)

[2.3 Mô hình hóa 24](#_Toc155312660)

[2.4 Mô tả chi tiết các Use case 25](#_Toc155312661)

[2.5 Thiết kế bảng dữ liệu 31](#_Toc155312662)

[CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ 34](#_Toc155312663)

[3.1 Thực nghiệm chương trình 34](#_Toc155312664)

[3.2 Kiểm thử và đánh giá 39](#_Toc155312665)

[KẾT LUẬN 41](#_Toc155312666)

[TÀI LIỆU THAM KHẢO 42](#_Toc155312667)

Danh mục hình ảnh

Hình 1. Android Studio 9

Hình 2. Cấu trúc dự án Android Studio 10

Hình 3. Tính năng của Android Studio 14

Hình 4. Java Android 15

Hình 5. Camera2 API 16

Hình 6. Firebase Authentication 18

Hình 7. Firebase Firestore 20

Hình 8. Mô hình MVVM 21

Hình 9. Biểu đồ Use case 24

Hình 10. Mô hình cơ sở dữ liệu 25

Hình 11. Màn hình đăng nhập 34

Hình 12. Màn hình đăng ký 35

Hình 13. Màn hình chụp và chia sẻ ảnh 35

Hình 14. Màn hình xem bảng tin và xem ảnh 36

Hình 15. Màn hình quản lý - tìm kiếm bạn bè 37

Hình 16. Màn hình chỉnh sửa tên tài khoản 38

Hình 17. Màn hình thay đổi ảnh đại diện 39

Danh mục bảng

Bảng 1. Mô tả chi tiết use case Đăng ký 25

Bảng 2. Mô tả use case Đăng nhập 26

Bảng 3. Mô tả use case Chia sẻ ảnh 27

Bảng 4. Mô tả use case Xem ảnh 27

Bảng 5. Mô tả use case phản hồi 28

Bảng 6. Mô tả use case Quản lý bạn bè 29

Bảng 7. Mô tả use case quản lý hồ sơ 30

Bảng 8. Bảng dữ liệu người dùng 31

Bảng 9. Bảng dữ liệu bạn bè 32

Bảng 10. Bảng dữ liệu bài đăng 32

Bảng 11. Bảng dữ liệu phản hồi 33

Bảng 12. Bảng dữ liệu bảng tin 33

Bảng 13. Bảng kết quả kiểm thử 39

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

* 1. Giới thiệu chung

Ứng dụng chia sẻ hình ảnh trực tuyến đã trở thành một phương tiện phổ biến để người dùng ghi lại và chia sẻ những khoảnh khắc đáng nhớ của cuộc sống. Khác với các nền tảng truyền thông khác, ứng dụng này mang đến một trải nghiệm chân thật và gần gũi hơn, cho phép người dùng chụp và đăng tải hình ảnh ngay tại thời điểm thực, mà không qua quá trình chỉnh sửa hay can thiệp từ bên ngoài.

Mỗi bức ảnh được chia sẻ trên ứng dụng đều là một khoảnh khắc tức thì, tái hiện chân thật những cảm xúc và sự kiện trong cuộc sống của người dùng. Điều này tạo ra một không gian đáng tin cậy và gần gũi, nơi mà mọi khoảnh khắc đều được giữ nguyên vẹn và chân thật nhất.

Một điểm nổi bật của ứng dụng là việc giới hạn số lượng bạn bè mỗi người dùng có thể kết nối. Thay vì tập trung vào việc thu thập lượt tương tác hay sự chú ý công khai, ứng dụng tập trung vào việc xây dựng một môi trường kết nối chất lượng, nơi mà người dùng có thể tập trung vào những người thực sự quan trọng đối với họ. Điều này không chỉ giúp bảo vệ sự riêng tư mà còn tạo ra một trải nghiệm người dùng trọn vẹn hơn, không bị phân tán bởi các yếu tố ngoài lề.

Ngoài ra, khả năng kết nối người dùng với những người thân thiết cũng là một yếu tố quan trọng. Ứng dụng cho phép người dùng dễ dàng chia sẻ hình ảnh với các bạn bè trong phạm vi được kết nối một cách nhanh chóng và thuận tiện, từ đó tăng cường mối quan hệ và tạo ra những kỷ niệm đáng nhớ.

Với những đặc tính nổi bật này, ứng dụng chia sẻ hình ảnh trực tuyến không chỉ đơn thuần là một công cụ giúp lưu giữ hình ảnh mà còn là một nền tảng tạo ra những trải nghiệm thú vị và ý nghĩa cho người dùng. Không chỉ đáp ứng nhu cầu về sự gần gũi và chân thật, ứng dụng còn đóng vai trò quan trọng trong việc xây dựng và duy trì các mối quan hệ xã hội, mang lại giá trị sống tinh thần sâu sắc cho người dùng trong thời đại số hiện nay.

* 1. Cơ sở lý thuyết

1.2.1. Android Studio

* Khái niệm

Android Studio là một môi trường tích hợp phát triển (Integrated Development Environment – IDE) được phát triển bởi Google, nhằm hỗ trợ các nhà phát triển trong việc xây dựng ứng dụng trên nền tảng Android. Với sự phát triển mạnh mẽ của hệ điều hành Android, Android Studio đã trở thành công cụ chủ đạo, không thể thiếu đối với các nhà phát triển phần mềm di động, nhờ vào sự hỗ trợ toàn diện mà nó mang lại.

Một trong những điểm nổi bật của Android Studio là giao diện người dùng trực quan, giúp các lập trình viên dễ dàng tiếp cận và sử dụng. Giao diện này cung cấp các công cụ mạnh mẽ như trình biên dịch, trình gỡ lỗi, và trình tạo giao diện người dùng (UI), giúp giảm thiểu thời gian phát triển và kiểm thử ứng dụng. Trình biên dịch của Android Studio không chỉ hỗ trợ biên dịch mã nguồn một cách nhanh chóng mà còn cung cấp các cảnh báo và đề xuất nhằm tối ưu hóa mã nguồn, giúp lập trình viên phát hiện và sửa lỗi ngay trong quá trình phát triển.

Android Studio cũng hỗ trợ mạnh mẽ cho việc viết mã nguồn trong các ngôn ngữ lập trình phổ biến như Java và Kotlin. Kotlin, mặc dù mới được giới thiệu trong những năm gần đây, nhưng đã nhanh chóng trở thành lựa chọn ưa thích của nhiều nhà phát triển nhờ vào cú pháp đơn giản, tính an toàn và khả năng tương thích ngược với Java. Ngoài ra, Android Studio còn tích hợp sẵn các thư viện và công cụ phát triển Android, cho phép nhà phát triển dễ dàng tiếp cận các tính năng tiên tiến của hệ điều hành, từ đó tạo ra những ứng dụng sáng tạo và chất lượng.

Không chỉ dừng lại ở việc phát triển ứng dụng, Android Studio còn hỗ trợ quản lý phiên bản thông qua việc tích hợp Git, một hệ thống quản lý mã nguồn phổ biến. Điều này giúp các nhóm phát triển có thể làm việc đồng bộ, theo dõi và quản lý lịch sử thay đổi của mã nguồn một cách hiệu quả.

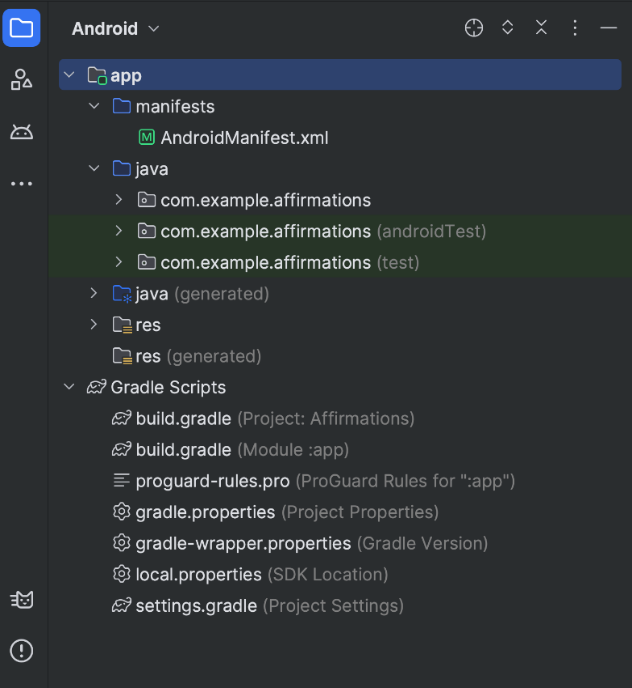
Với những tính năng ưu việt, Android Studio thực sự là một công cụ mạnh mẽ, giúp nhà phát triển dễ dàng xây dựng, kiểm thử và triển khai các ứng dụng Android, từ đó đem lại những trải nghiệm tốt nhất cho người dùng.



Hình 1. Android Studio

Một trong những tính năng quan trọng của công cụ này là khả năng sử dụng Gradle, một hệ thống quản lý dự án mạnh mẽ. Gradle cho phép bạn quản lý phụ thuộc, xây dựng, kiểm thử và đóng gói ứng dụng Android một cách linh hoạt và hiệu quả.

* Cấu trúc dự án

Mỗi dự án trong Android Studio chứa một hoặc nhiều mô-đun có tệp mã nguồn và tệp tài nguyên. Có các loại mô-đun sau:

Hình 2. Cấu trúc dự án Android Studio

* Mô-đun ứng dụng Android
* Mô-đun thư viện
* Mô-đun Google App Engine

Theo mặc định, Android Studio thể hiện các tệp dự án của bạn trong chế độ xem dự án Android, như trong hình 1. Khung hiển thị này được sắp xếp theo mô-đun để bạn có thể truy cập nhanh vào các tệp nguồn chính của dự án. Bạn có thể thấy mọi tệp bản dựng ở cấp cao nhất trong Gradle Scripts (Tập lệnh Gradle).

Mỗi mô-đun ứng dụng có chứa các thư mục sau:

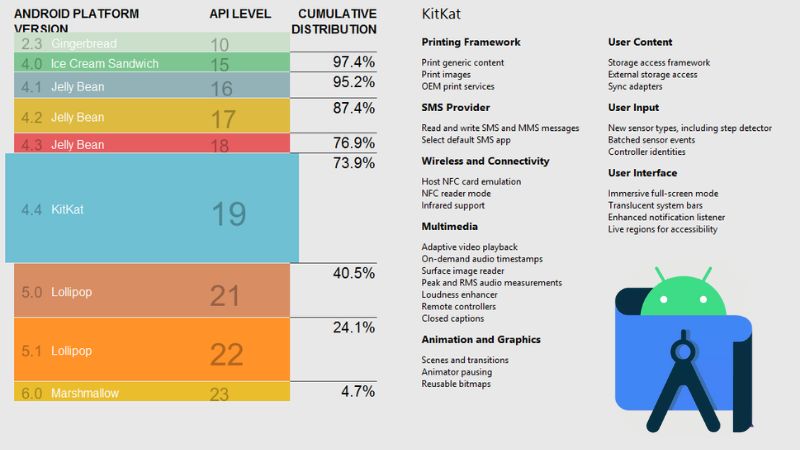
* manifests (tệp kê khai): Chứa tệp AndroidManifest.xml.
* java: Chứa các tệp mã nguồn Java và Kotlin, bao gồm cả mã kiểm thử JUnit.
* res: Chứa mọi tài nguyên không phải đoạn mã, chẳng hạn như chuỗi giao diện người dùng và hình ảnh bitmap.
* Đánh giá

***Một số ưu điểm của Android Studio:***

* Đây là môi trường phát triển phần mềm chính thức của Google, đây cũng chính là chủ sở hữu của hệ điều hành Android.
* Công cụ này cung cấp một loạt các công cụ phát triển và tính năng hỗ trợ cho quá trình phát triển ứng dụng Android. Điều này bao gồm trình biên dịch, trình gỡ lỗi, trình quản lý dự án, thiết kế giao diện và nhiều tính năng khác để tăng năng suất và hiệu quả của nhà phát triển.
* Giao diện người dùng thân thiện và dễ sử dụng, giúp tăng hiệu suất và giảm thời gian cần thiết để phát triển ứng dụng.
* Hỗ trợ bởi một số tài liệu tham khảo và hướng dẫn phong phú. Có sẵn các tài liệu chính thức từ Google, cùng với các diễn đàn lập trình viên Android và các nguồn tài liệu trực tuyến khác để giúp nhà phát triển tìm hiểu và giải quyết các vấn đề phát triển.
* Công cụ này được hỗ trợ thông qua các khóa học đào tạo về lập trình Android cơ bản và nâng cao. Các khóa học này giúp nhà phát triển nắm vững các khái niệm và kỹ năng cần thiết để phát triển ứng dụng Android chất lượng.

***Nhược điểm:***

* Chiếm dụng tài nguyên hệ thống lớn: yêu cầu nhiều RAM, CPU và dung lượng ổ đĩa, ảnh hưởng đến hiệu suất máy tính
* Hiệu suất chậm trên máy cấu hình thấp: chạy chậm, dễ giật lag, đặc biệt với các dự án lớn và trên máy cấu hình yếu
* Hiệu suất chậm trên máy cấu hình thấp: chạy chậm, dễ giật lag, đặc biệt với các dự án lớn và trên máy cấu hình yếu
* Phức tạp và khó học: giao diện và tính năng phức tạp, đòi hỏi thời gian và nỗ lực để làm quen
* Lỗi và sự không ổn định: thỉnh thoảng gặp lỗi, ảnh hưởng đến quá trình phát triển và cập nhật
* Tính tương thích và phụ thuộc hệ thống: phụ thuộc nhiều vào hệ điều hành và phần cứng, cần cập nhật thường xuyên
* Một số tính năng



Hình 3. Tính năng của Android Studio

1.2.2. Java Android

* Khái niệm

Java là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay. Nó được sử dụng nhiều để phát triển phần mềm & web. Hiện nay, Java đã trở thành một ngôn ngữ phổ biến cho các ứng dụng di động. Còn Android được xem là nền tảng dựa trên các thiết bị điện thoại di động do Google phát triển. Trong đó, sự phát triển của Android hầu hết đều dựa trên Java. Một phần lớn các thư viện Java đều có sẵn trong nền tảng Android, chỉ một bộ phận thư viện khác tồn tại trong Android (cho giao diện người dùng,…).

Java là ngôn ngữ lập trình chính thức của hệ điều hành android. Nó cũng là một trong những ngôn ngữ lập trình được sử dụng phổ biến nhất trên thế giới, với ước chừng khoảng 9 triệu nhà phát triển. HĐH Android do Google phát triển sử dụng Java làm cơ sở cho tất cả các ứng dụng android. Trong đó Java Android không hoàn toàn giống như java thông thường. Nhưng nó cũng có nhiều điểm chung.

Java được thiết kế có thể tương thích với nhiều môi trường phát triển, nên nó linh hoạt hơn ngôn ngữ lập trình hướng đến đối tượng C++ và ngôn ngữ lập trình C. Đặc biệt, Java còn có hiệu suất cao hơn với trình thu gom rác, giải phóng bộ nhớ từ các đối tượng không được dùng đến. Bởi vậy Java Android được sử dụng phổ biến trong các thiết bị di động sử dụng hệ điều hành này.

* Các tính năng

Các ứng dụng Android được phát triển bằng ngôn ngữ Java là sự lựa chọn tối ưu cho các ứng dụng Android. Java là một ngôn ngữ lập trình phổ biến được phát triển bởi Sun Microsystems (hiện đang thuộc sở hữu của Oracle). Phát triển sau C và C ++, nên Java kết hợp nhiều tính năng mạnh mẽ của các ngôn ngữ này.



Hình 4. Java Android

Một số tính năng quan trọng của Java là:

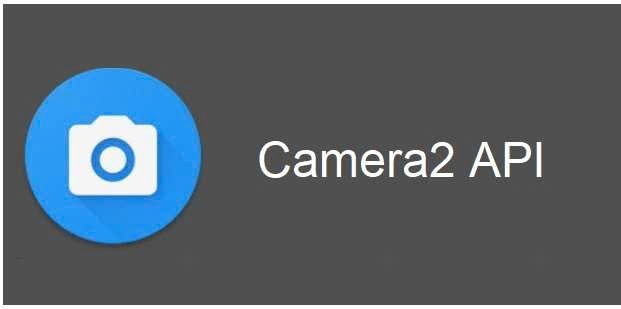
* Ngôn ngữ Java được thiết kế để có nền tảng độc lập & an toàn, được sử dụng cho các máy ảo.
* Ngôn ngữ lập trình Java hướng đối tượng Android chủ yếu dựa vào các nguyên tắc cơ bản Java.
* SDK Android có nhiều thư viện Java chuẩn như thư viện cấu trúc dữ liệu, thư viện đồ họa, thư viện toán học, thư viện mạng… nhiều thứ khác có thể cần đến. Bên cạnh đó, còn có thư viện Android đặc biệt giúp phát triển các ứng dụng Android tốt hơn.

1.2.3. Camera API

* Khái niệm

Camera API, trong đó API là viết tắt của cụm từ tiếng Anh **Application Programming Interface** (nghĩa là giao diện lập trình ứng dụng). Camera API chính là máy ảnh được lập trình một phần mềm giao tiếp liên kết các ứng dụng với nhau.

Camera API là một dịch vụ web RESTful cho phép các nhà phát triển thêm các tính năng quản lý hình ảnh từ camera vào các ứng dụng hoặc trang web của họ. Trên Android, Camera API được hỗ trợ thông qua gói **android.hardware.camera2**. Đây là API chính để điều khiển camera trên các thiết bị Android. Nó có thể được sử dụng để chụp hình hoặc quay video khi bạn đang xây dựng một ứng dụng camera.



Hình 5. Camera2 API

* Một số lớp liên quan trong Camera API
* **android.hardware.camera2**: Gói này là API chính để điều khiển camera trên các thiết bị.
* **Camera**: Lớp này là API đã lỗi thời để điều khiển camera trên các thiết bị.
* **SurfaceView**: Lớp này được sử dụng để hiển thị camera trực tiếp cho người dùng.
* **MediaRecorder**: Lớp này được sử dụng để ghi video từ camera.
* **Intent**: Một loại hành động Intent có kiểu **MediaStore.ACTION\_IMAGE\_CAPTURE** hoặc **MediaStore.ACTION\_VIDEO\_CAPTURE** có thể được sử dụng để chụp hình ảnh hoặc video mà không cần sử dụng trực tiếp đối tượng Camera2.
* Lưu ý
* Việc sử dụng máy ảnh ảnh hưởng đến ứng dụng của khi cài đặt trên Android. Camera API chỉ tương ứng với 1 số Android hữu hiệu hiện nay.
* Trên Android 9 ( Camera API cấp 28) trở lên, các ứng dụng chạy nền không thể truy cập vào máy ảnh API được. Do đó, nếu muốn sử dụng camera API bạn phải thiết lập một phần mềm giả định trước hoặc cài đặt để nâng cấp Android của mình.
* Các hình ảnh hoặc video được chụp từ ứng dụng camera API chỉ được hiển thị trên chính phần mềm ứng dụng, hoặc khi chia sẻ với một ứng dụng camera API khác. Vì vậy, nếu muốn lưu hình ảnh chụp từ Camera API vào thư viện hoặc đăng tải lên các phương tiện truyền thông cần cài đặt Camera API phần Saving Media Files để xem cách triển khai các tùy chọn lưu cho phù hợp.
* Camera API như một chiếc máy ảnh thông minh với sự kết nối giữa phần mềm chụp ảnh API với các nền tảng ứng dụng khác nhau. Camera API chính là sự kết nối để cả hai cỗ máy (Máy ảnh và ứng dụng) có thể hoạt động bình thường. Vì thế khi đóng hoặc ngừng chạy Camera API thì kết nối giữa máy ảnh và ứng dụng sẽ ngừng hoạt động.

1.2.4. Firebase Authentication

* Khái niệm

Firebase Authentication là một hệ thống xác thực dựa trên token cung cấp dịch vụ backend, SDK dễ sử dụng và thư viện giao diện người dùng sẵn có để xác thực người dùng với ứng dụng. Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, các nhà cung cấp danh tính liên kết phổ biến như Google, Facebook và Twitter, và nhiều hơn nữa.



Hình 6. Firebase Authentication

* Đặc điểm
* *Tích hợp chặt chẽ:* Firebase Authentication tích hợp chặt chẽ với các dịch vụ Firebase khác và nó tận dụng các tiêu chuẩn ngành như OAuth 2.0 và OpenID Connect, vì vậy nó có thể dễ dàng tích hợp với backend tùy chỉnh.
* *Hỗ trợ đa dạng*: Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, các nhà cung cấp danh tính liên kết phổ biến như Google, Facebook và Twitter, và nhiều hơn nữa.
* *Nâng cấp*: Khi nâng cấp lên Firebase Authentication với Identity Platform và mở khóa các tính năng bổ sung, chẳng hạn như xác thực đa yếu tố, chức năng chặn, ghi nhật ký hoạt động và kiểm tra người dùng, hỗ trợ SAML và OpenID Connect chung, đa thuê ký và hỗ trợ doanh nghiệp.
* *Dễ sử dụng*: Có thể đăng nhập người dùng vào ứng dụng Firebase của mình bằng cách sử dụng FirebaseUI như một giải pháp xác thực toàn diện hoặc bằng cách sử dụng SDK Firebase Authentication để tích hợp một hoặc nhiều phương thức đăng nhập vào ứng dụng.

1.2.5. Firebase Firestore

* Khái niệm

Firebase Firestore, còn được gọi là Cloud Firestore, là một cơ sở dữ liệu NoSQL dựa trên đám mây, linh hoạt và có thể mở rộng. Dữ liệu được lưu trữ dưới dạng tài liệu, được tổ chức thành các bộ sưu tập, và được đồng bộ hóa theo thời gian thực trên nhiều ứng dụng khách.



Hình 7. Firebase Firestore

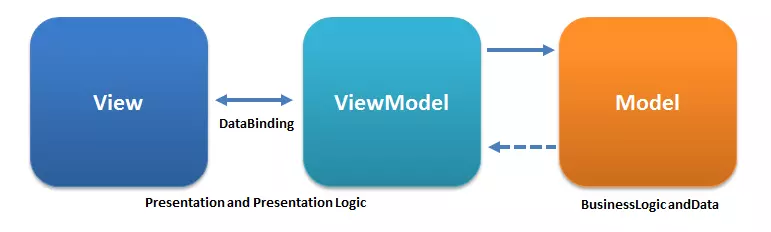
* Đặc điểm
* *Tích hợp chặt chẽ:* Cloud Firestore tích hợp chặt chẽ với các dịch vụ Firebase khác và Google Cloud.
* *Hỗ trợ đa dạng*: Firestore hỗ trợ lưu trữ và đồng bộ hóa dữ liệu cho phát triển ứng dụng di động, web và máy chủ.
* *Hiệu suất:* Hiệu suất truy vấn của Firestore tỷ lệ thuận với kích thước của tập kết quả, không phải kích thước của tập dữ liệu.
* *Đồng bộ hóa dữ liệu:* Firestore sử dụng đồng bộ hóa dữ liệu để cập nhật dữ liệu trên bất kỳ thiết bị nào được kết nối.
* *Hỗ trợ ngoại tuyến:* Firestore lưu trữ dữ liệu mà ứng dụng của bạn đang sử dụng tích cực, vì vậy ứng dụng có thể ghi, đọc, nghe và truy vấn dữ liệu ngay cả khi thiết bị đang ngoại tuyến.
* *Cơ sở hạ tầng mạnh mẽ*: Firestore mang lại lợi ích cơ sở hạ tầng mạnh mẽ của Google Cloud, bao gồm đa vùng dữ liệu tự động, đảm bảo tính nhất quán mạnh, hoạt động hàng loạt nguyên tử và hỗ trợ giao dịch thực.

1.2.6. Mô hình MVVM

* Khái niệm

MVVM là một mô hình kiến trúc cho các dự án phần mềm. Mô hình này được phát triển dựa trên kiến trúc MVP.

MVVM không phải là framework hay thư viện, api… nó chỉ đơn thuần là hướng dẫn định nghĩa cấu trúc ứng dụng.



Hình 8. Mô hình MVVM

* Các thành phần chính
* *View*: Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng.
* *Model*: Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.
* *ViewModel*: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.
* Đặc điểm
* Phân tách rõ ràng: MVVM giúp phân tách giữa logic và view một cách rõ ràng và rạch ròi.
* Hỗ trợ unit test hiệu quả: MVVM hỗ trợ unit test hiệu quả.
* Hỗ trợ tốt cho dự án hỗ trợ nhiều loại thiết bị và kích thước khác nhau.
* ViewModel không hề biết gì về View: Một ViewModel có thể được sử dụng cho nhiều View (one-to-many).
* ViewModel sử dụng Observer design pattern để liên lạc với View (thường được gọi là binding data, có thể là 1 chiều hoặc 2 chiều tùy nhu cầu ứng dụng)

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

* 1. Yêu cầu chức năng

***Đăng ký, đăng nhập***

* Người dùng có thể tạo tài khoản mới bằng cách cung cấp thông tin cá nhân cơ bản hoặc đăng nhập bằng tài khoản đã có.
* Quá trình đăng nhập và đăng ký được bảo mật bằng các phương tiện xác thực an toàn để đảm bảo tính riêng tư như số điện thoại hoặc email.

***Chụp và chia sẻ ảnh***

* Người dùng có thể chụp ảnh trực tiếp từ ứng dụng với giao diện camera tích hợp.
* Ảnh được chụp sẽ được chia sẻ như một khoảnh khắc của bạn với bạn bè mà không thông qua bất kì quá trình chỉnh sửa nào.

***Bảng tin***

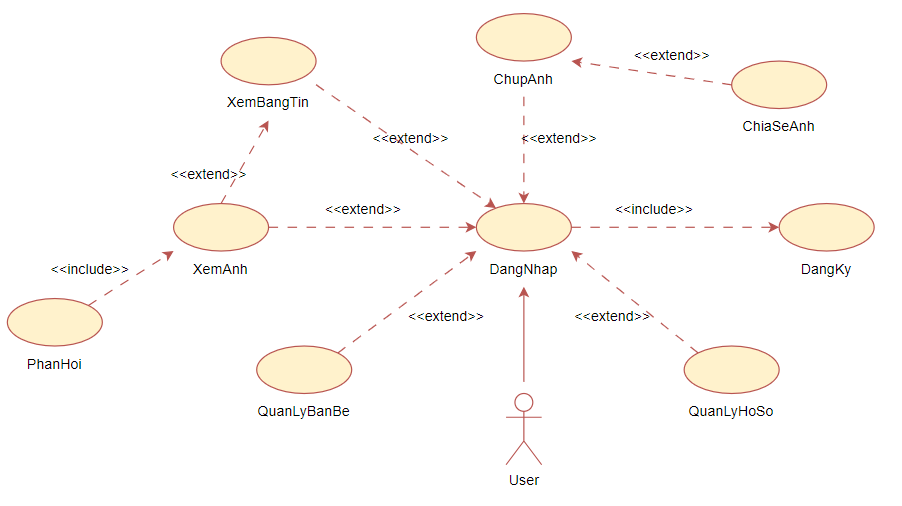
* Bảng tin là nơi tất cả các hình ảnh trong khoảnh khắc mới nhất của bạn bè được hiển thị.
* Cho phép người dùng thích, bình luận và chia sẻ ảnh từ bảng tin. Đồng thời, cho phép người dùng có thể xem lại toàn bộ các khoảnh khắc của bạn bè dưới dạng danh sách lưới
* Người dùng có thể thả icon cảm xúc, bình luận trực tiếp từ bảng tin hoặc chia sẻ bài viết với bạn bè.

***Hồ sơ người dùng***

* Mỗi người dùng có một trang hồ sơ thông tin cá nhân.
* Cho phép thay đổi thông tin cá nhân và các cài đặt liên quan khác.

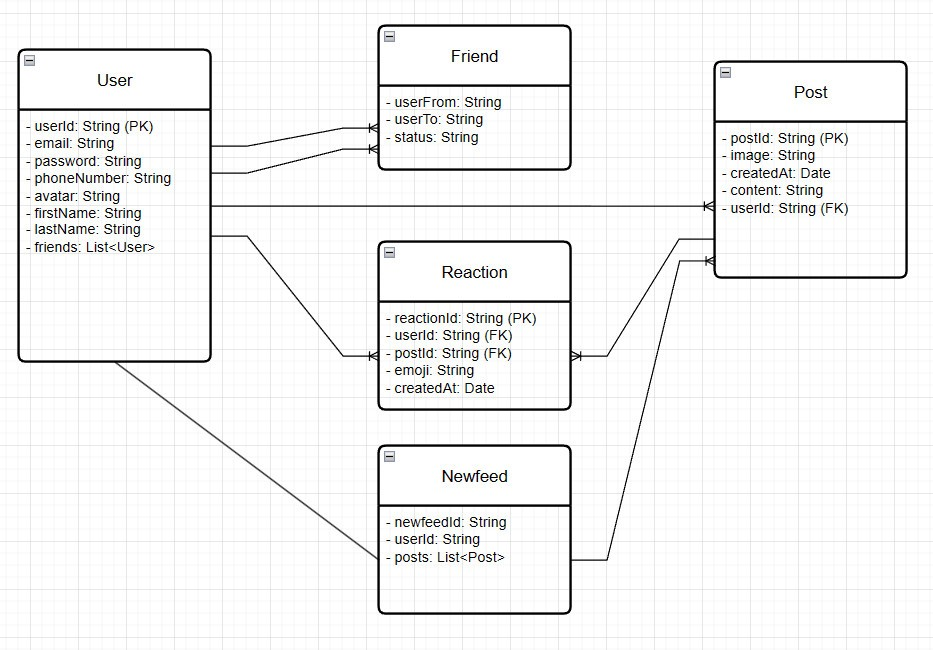
***Bạn bè***

* + Người dùng có thể kết bạn với nhau, gửi lời mời kết bạn và quản lý danh sách bạn bè.
  + Thông báo về hoạt động của bạn bè, như khi họ đăng ảnh mới hoặc thích ảnh của bạn hay bình luận ảnh của bạn.
  1. Yêu cầu phi chức năng
* Hệ thống cần cung cấp cơ chế đăng nhập, đăng ký an toàn, bảo mật thông tin người dùng và phải có xác thực hai yếu tố.
* Ứng dụng phải tích hợp được giao diện camera để người dùng có thể chụp ảnh một cách thuận tiện.
* Bảng tin chung phải đảm bảo được cập nhật theo thời gian thực với các bức ảnh mới từ bạn bè.
* Ứng dụng cần có ngôn ngữ phù hợp, giao diện thân thiện với người dùng, dễ sử dụng, có khả năng tương thích với nhiều thiết bị và hệ điều hành khác nhau, và đảm bảo an toàn thông tin người dùng.
  1. Mô hình hóa
* Mô hình hóa chức năng người dùng



Hình 9. Biểu đồ Use case

* Mô hình cơ sở dữ liệu



Hình 10. Mô hình cơ sở dữ liệu

* 1. Mô tả chi tiết các Use case

Bảng 1. Mô tả chi tiết use case Đăng ký

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng đăng ký. |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Không có |
| **Luồng cơ bản** | 1. Use case này bắt đầu khi người dùng kích vào “Create an accout” trên giao diện hiển thị lần đầu tiên. Hệ thống sẽ hiển thị form yêu cầu nhập email, password, name (Bao gồm First name và Last name). 2. Người dùng kích vào “Continue” sau mỗi bước nhập thông tin. Hệ thống sẽ tạo một tài khoản mới và lưu các thông tin người dùng vừa nhập vào bảng USERS và chuyển về màn hình Đăng nhập. Use case kết thúc. |
| **Luồng rẽ nhánh** | 1. Tại bước 1 của luồng cơ bản, nếu người dùng nhập sai hoặc thiếu thông tin mà form đăng ký yêu cầu thì hệ thống hiển thị lỗi, yêu cầu người dùng nhập lại. 2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Có thể thực hiện Use case đăng nhập |

Bảng 2. Mô tả use case Đăng nhập

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng đăng nhập. |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Không có |
| **Luồng cơ bản** | * 1. Use case này bắt đầu khi người dùng kích vào “Sign in” trên màn hình. Hệ thống sẽ hiển thị giao diện yêu cầu nhập số điện thoại hoặc email   2. Người dùng kích vào “Continue”, chuyển sang màn hình nhập password. Người dùng nhập password sau đó chọn “Continue”, hệ thống truy xuất bảng USERS trong CSDL, xác nhận lại thông tin người dùng nhập và chuyển tới màn hình chụp ảnh. Use case kết thúc. |
| **Luồng rẽ nhánh** | 1. Tại bước 1, 2 của luồng cơ bản, nếu người dùng nhập thông tin tên tài khoản và mật khẩu không khớp với tài khoản nào trong bảng USERS thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc. 2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 3. Mô tả use case Chia sẻ ảnh

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng chia sẻ ảnh. |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Người dùng đã chọn chụp ảnh |
| **Luồng cơ bản** | 1. Use case này bắt đầu khi người dùng kích vào nút chụp ảnh trên màn hình. Hệ thống chuyển sang giao diện chia sẻ ảnh, cho phép nhập nội dung ngắn lên ảnh, mặc định tất cả bạn bè có thể xem ảnh này. 2. Người dùng kích vào biểu tượng chia sẻ. Hệ thống lưu thông tin ảnh vào bảng POST. Use case kết thúc. |
| **Luồng rẽ nhánh** | 1. Tại bước 2 của luồng cơ bản, nếu người dùng chọn biểu tượng “X” thì sẽ trở về màn hình chụp ảnh và use case kết thúc. 2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 4. Mô tả use case Xem ảnh

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng xem ảnh. |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Người dùng đã đăng nhập thành công |
| **Luồng cơ bản** | 1. Use case này bắt đầu khi người dùng chọn biểu tượng danh sách hình ảnh trên màn hình. Hệ thống truy xuất dữ liệu từ bảng NEWFEED và hiển thị bảng tin chứa hình ảnh của người dùng đó cũng như hình ảnh được bạn bè chia sẻ. 2. Người dùng chọn ảnh từ bảng tin. Hệ thống truy xuất dữ liệu thông tin ảnh từ bảng POST và hiện thị lên màn hình giao diện xem hình ảnh bao gồm cả nội dung kèm theo, tên người chia sẻ và thời gian chia sẻ. Use case kết thúc. |
| **Luồng rẽ nhánh** | 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 5. Mô tả use case phản hồi

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng phản hồi. |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Người dùng đã chọn xem ảnh từ bảng tin |
| **Luồng cơ bản** | 1. Use case này bắt đầu khi người dùng chọn xem ảnh bạn bè trong bảng tin của mình. Màn hình hiện thị các icon cảm xúc. 2. Người dùng kích vào icon mình muốn. Hệ thống lưu thông tin icon được chọn vào bảng REACTION và hiển thị nó lên màn hình. Use case kết thúc. |
| **Luồng rẽ nhánh** | 1. Tại bước 1 của luồng cơ bản, nếu người dùng chọn xem ảnh của chính mình thì hiển thị lên những phản hồi nhận được từ bạn bè và thời gian nhận được, use case kết thúc. 2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 6. Mô tả use case Quản lý bạn bè

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng quản lý bạn bè |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Người dùng đã đăng nhập thành công |
| **Luồng cơ bản** | 1. Use case này bắt đầu khi người dùng chọn biểu tượng bạn bè trên màn hình. Hệ thống truy xuất dữ liệu từ bảng FRIEND và hiển thị danh sách bạn bè trên giao diện quản lý.    1. *Tìm kiếm:* Người dùng nhập số điện thoại của người cần tìm vào ô tìm kiếm, sau đó chọn biểu tượng tìm. Hệ thống sẽ so sánh số điện thoại vừa nhập với dữ liệu tương ứng có trong bảng USER và hiển thị kết quả lên màn hình. Use case kết thúc    2. *Xóa:* Người dùng chọn “X” tương ứng với người mình muốn xóa, sau khi xác nhận xóa, hệ thống sẽ xóa thông tin liên quan trong bảng FRIEND và cập nhật lại danh sách bạn bè trên màn hình. Use case kết thúc    3. *Thêm:* Người dùng chọn “Add” tương ứng với người mình muốn thêm bạn bè, hệ thống sẽ gửi thông báo về yêu cầu kết bạn tới người nhận. Use case kết thúc |
| **Luồng rẽ nhánh** | 1. Tại bước 1.1 của luồng cơ bản, nếu người dùng chọn “Cancel” thì số điện thoại đã nhập được xóa đi và use case kết thúc. 2. Tại bước 1.1 của luồng cơ bản, nếu hệ thống không tìm được người dùng có số điện thoại đó thì sẽ hiển thị thông báo và use case kết thúc. 3. Tại bước 1.2 và 1.3 của luồng cơ bản, nếu người dùng chọn “Cancel” thì quay trở về màn hình danh sách bạn bè và use case kết thúc. 4. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 7. Mô tả use case quản lý hồ sơ

|  |  |
| --- | --- |
| **Mô tả vắn tắt** | Use case này cho phép người dùng quản lý hồ sơ |
| **Đối tượng** | Người dùng |
| **Tiền điều kiện** | Người dùng đã đăng nhập thành công |
| **Luồng cơ bản** | 1. Use case này bắt đầu khi người dùng chọn biểu tượng đại diện trên màn hình. Hệ thống chuyển sang giao diện quản lý hồ sơ bao gồm ảnh đại diện, tên người dùng và một số thông tin cài đặt khác.    1. *Thay đổi avatar:* Người dùng kích vào “+” trên avatar, hệ thống cho phép tải ảnh lên từ thư viện cá nhân hoặc chụp ảnh mới.    2. *Thay đổi tên hiển thị:* Người dùng kích vào “Edit info”, hệ thống hiển thị màn hình edit tên gồm First name và Last name. 2. Người dùng chọn “Save” để lưu thay đổi đã thực hiện, Hệ thống sẽ cập nhật lại thông tin về ảnh đại diện (hoặc tên tài khoản) trong bảng USER và hiển thị lên màn hình. Use case kết thúc. |
| **Luồng rẽ nhánh** | 1. Tại bước 2 của luồng cơ bản, nếu người dùng chọn “Cancel” thì lập tức quay trở về màn hình quản lý hồ sơ và use case kết thúc. 2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Các yêu cầu đặc biệt** | Không có |
| **Hậu điều kiện** | Không có |

* 1. Thiết kế bảng dữ liệu

Bảng 8. Bảng dữ liệu người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USER | | | | |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Mô tả |
| UserID | VARCHAR |  | Khóa chính | Mã tài khoản người dùng |
| Email | VARCHAR | 255 | Not null | email người dùng |
| Password | VARCHAR | 255 | Not null | Mật khẩu người dùng |
| PhoneNumber | VARCHAR | 255 | Not null | Số điện thoại người dùng |
| FirstName | VARCHAR | 255 | Not null | Tên của người dùng |
| LastName | VARCHAR | 255 | Not null | Họ của người dùng |

Bảng 9. Bảng dữ liệu bạn bè

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FRIEND | | | | |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Mô tả |
| UserFrom | VARCHAR |  | Not null | Người gửi yêu cầu |
| UserTo | VARCHAR | 255 | Not null | Người nhận |
| Status | VARCHAR | 255 | Not null | Trạng thái |

Bảng 10. Bảng dữ liệu bài đăng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| POST | | | | |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Mô tả |
| PostID | VARCHAR | 255 | Khóa chính | Mã bài đăng |
| Image | IMAGE |  | Not null | Hình ảnh |
| CreateAt | DATETIME |  | Not null | Ngày đăng |
| Content | VARCHAR | 255 |  | Nội dung |
| UserID | VARCHAR | 255 | Khóa ngoại chính | Mã tài khoản người dùng |

Bảng 11. Bảng dữ liệu phản hồi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REACTION | | | | |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Mô tả |
| ReactionID | VARCHAR | 255 | Khóa chính | Mã phản hồi |
| UserID | VARCHAR | 255 | Khóa ngoại chính | Mã tài khoản người dùng |
| PostID | VARCHAR | 255 | Khóa ngoại chính | Mã bài đăng |
| Emoji | IMAGE |  |  |  |
| CreateAt | DATETIME |  | Not null | Ngày phản hồi |

Bảng 12. Bảng dữ liệu bảng tin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NEWFEED | | | | |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Mô tả |
| NewfeedID | VARCHAR | 255 | Khóa chính | Mã phản hồi |
| UserID | VARCHAR | 255 | Khóa ngoại chính | Mã tài khoản người dùng |

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

1. Thực nghiệm chương trình

* Đăng nhập

Chọn **Sign in** => nhập địa chỉ email và mật khẩu để thực hiện đăng nhập

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 11. Màn hình đăng nhập

|  |  |  |
| --- | --- | --- |
| * Đăng ký   Chọn **Create an account**  => Nhập email và mật khẩu (8 ký tự) để thực hiện đăng ký tài khoản |  |  |

Hình 12. Màn hình đăng ký

* Chụp và chia sẻ ảnh

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 13. Màn hình chụp và chia sẻ ảnh

* Xem bảng tin và ảnh cũng như phản hồi ảnh

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 14. Màn hình xem bảng tin và xem ảnh

|  |  |  |
| --- | --- | --- |
| * Tìm kiếm bạn bè   Nhập tên tài khoản hoặc địa chỉ email muốn tìm kiếm.  => Nhấn **Find** |  |  |
|  |  |  |

Hình 15. Màn hình quản lý - tìm kiếm bạn bè

* Chỉnh sửa tên tài khoản

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 16. Màn hình chỉnh sửa tên tài khoản

* Thay đổi ảnh đại diện

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 17. Màn hình thay đổi ảnh đại diện

1. Kiểm thử và đánh giá

Bảng 13. Bảng kết quả kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Case** | **Đầu vào** | **Đầu ra mong muốn** | **Kết quả** |
| **1** | Kiểm tra chức năng đăng nhập | Nhập tài khoản và mật khẩu hợp lệ | Đăng nhập thành công, chuyển trang với chế độ đã đăng nhập | **Đạt** |
| Nhập tài khoản và mật khẩu không hợp lệ | Thông báo đăng nhập không thành công | **Đạt** |
| **2** | Kiểm tra chức năng đăng ký | Nhập thông tin hợp lệ | Thông báo đăng ký thành công | **Đạt** |
| Nhập thông tin không hợp lệ | Nhập lại | **Đạt** |
| **3** | Kiểm tra chức năng chia sẻ ảnh | Chụp ảnh và đăng lên bản tin | Ảnh được hiển thị lên trên bảng tin | **Đạt** |
| **4** | Kiểm tra chức năng phản hồi | Chọn xem ảnh của bạn bè từ bảng tin | Hiển thị màn hình cho phép chọn icon cảm xúc | **Đạt** |
| **5** | Kiểm tra chức năng quản lý bạn bè | Tìm kiếm bạn bè bằng tên hợp lệ | Hiển thị avatar và tên người cần tìm, cho phép gửi yêu cầu kết bạn | **Đạt** |
| Tìm kiếm bạn bè bằng tên không hợp lệ | Hiển thị thông báo không tìm thấy | **Đạt** |
| Chọn xác nhận xóa bạn bè khỏi danh sách | Cập nhật lại danh sách bạn bè | **Đạt** |
| **6** | Kiểm tra chức năng quản lý hồ sơ | Thực hiện thay đổi ảnh đại diện | Cập nhật lại avatar hiển thị | **Đạt** |
| Thực hiện thay đổi tên tài khoản | Cập nhật lại tên tài khoản hiển thị | **Đạt** |

=> **Đánh giá**: Chương trình ứng dụng mềm đã được hoàn thiện và có thể chạy bình thường. Các chức năng hoạt động ổn định và đúng với vai trò của nó.

KẾT LUẬN

Sau thời gian tìm hiểu và làm việc nghiêm túc dưới sự hướng dẫn của thầy Nguyễn Thái Cường, nhóm 7 chúng em đã hoàn thành cuốn báo cáo bài tập lớn với đề tài: “Xây dựng ứng dụng chia sẻ hỉnh ảnh trực tuyến với Android”.

Chúng em đã đúc kết được nhiều kinh nghiệm, học hỏi thêm được nhiều kiến thức trong việc phân tích và thiết kế yêu cầu hệ thống, xây dựng hệ hỗ trợ ra quyết định, góp phần hoàn thiện kỹ năng và tư duy, củng cố các kiến thức đã học, đồng thời tích lũy kinh nghiệm làm bài tập lớn cho các học phần sau này.

1. **Công viện hoàn thành**

* Tìm hiểu và trình bày tổng quan về Android Studio và một số Firebase, mô hình được áp dụng.
* Phân tích và thiết kế, xây dựng được ứng dụng chia sẻ hình ảnh trực tuyến với các chức năng cơ bản cần có.

1. **Hướng phát triển**

* Kiểm tra các trường hợp còn thiếu sót.
* Nâng cao chế độ bảo mật cho ứng dụng.
* Xây dựng thêm nhiều chức năng để hệ thống đầy đủ và toàn vẹn hơn.
* Tìm hiểu xu hướng của người dùng để chỉnh sửa, bổ sung hệ thống phù hợp.
* Củng cố kiến thức, rèn luyện, nâng cao kỹ phân tích và thiết kế chương trình ứng dụng di động trên Android Studio, từ đó tích lũy thêm kinh nghiệm cho bản thân.

LINK GITHUB

https://github.com/CamTuanNghia/Locket\_clon\_adr

**TÀI LIỆU THAM KHẢO**

1. https://viblo.asia/p/tim-hieu-ve-mo-hinh-mvvm-maGK7vW95j2
2. <https://firebase.google.com/docs/auth?hl=vi>
3. <https://firebase.google.com/docs/firestore?hl=vi>
4. <https://developer.android.com/guide/topics/media/camera>