BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN**

**XÂY DỰNG ỨNG DỤNG TÌM ĐỒ THẤT LẠC CHO SINH VIÊN TRƯỜNG ĐẠI HỌC NHA TRANG**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Nguyễn Thái Tuấn**

**Mã số sinh viên: 59132942**

KHÁNH HÒA - 2021

TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KĨ THUẬT PHẦM MỀM**



**ĐỒ ÁN**

**XÂY DỰNG ỨNG DỤNG TÌM ĐỒ THẤT LẠC CHO SINH VIÊN TRƯỜNG ĐẠI HỌC NHA TRANG**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Nguyễn Thái Tuấn

Mã số sinh viên: 59132942

Khánh Hòa, tháng 06/2021

# LỜI CAM ĐOAN

Tôi xin cam đoan mọi kết quả của đề tài ***“Ứng dụng tìm đồ thất lạc cho sinh viên Trường Đại học Nha Trang”***là công trình nghiên cứu của cá nhân tôi và chưa từng được công bố trong bất cứ công trình khoa học nào khác cho tới thời điểm này.

Khánh Hòa, ngày 22 tháng 06 năm 2021

Tác giả đồ án

*(kí và ghi rõ họ tên)*

# LỜI CẢM ƠN

Trong suốt thời gian thực hiện đề tài, tôi đã nhận được sự giúp đỡ của quý phòng ban trường Đại học Nha Trang, Khoa Công nghệ Thông tin nói chung và Bộ môn Kỹ thuật phần mềm nói riêng đã tạo điều kiện tốt nhất cho tôi được hoàn thành đề tài. Đặc biệt là sự hướng dẫn tận tình của ThS Đoàn Vũ Thịnh đã giúp tôi hoàn thành tốt đề tài. Qua đây, tôi xin gửi lời cảm ơn sâu sắc đến sự giúp đỡ này.

Xin chân thành cảm ơn các quý thầy cô trong bộ môn Kỹ thuật phần mềm trong thời gian qua đã truyền đạt và trang bị cho em kiến thức giúp em hoàn thành tốt đề tài tốt nghiệp.

Cuối cùng tôi xin gửi lời cảm ơn chân thành đến gia đình và tất cả bạn bè đã giúp đỡ, động viên tôi trong suốt quá trình học tập và thực hiện đề tài.

Tôi xin chân thành cảm ơn!

Khánh Hòa, ngày 22 tháng 06 năm 2021

Tác giả đồ án

*(kí và ghi rõ họ tên)*

# MỤC LỤC

[Lời cam đoan i](#_Toc76923434)

[Lời cảm ơn ii](#_Toc76923435)

[Mục lục iii](#_Toc76923436)

[Danh mục hình vi](#_Toc76923437)

[Danh mục bảng viii](#_Toc76923438)

[Danh mục các ký hiệu, từ viết tắt ix](#_Toc76923439)

[Lời mở đầu 1](#_Toc76923440)

[Chương I: TỔNG QUAN VẤN ĐỀ 2](#_Toc76923441)

[1.1. Tính cấp thiết của đề tài 2](#_Toc76923442)

[1.2. Mục tiêu đề tài 3](#_Toc76923443)

[1.3. Giải pháp thực hiện 3](#_Toc76923444)

[1.4. Đối tượng và phạm vi nghiên cứu 3](#_Toc76923445)

[1.5. Ý nghĩa lí luận và thực tiễn 3](#_Toc76923446)

[Chương II: CƠ SỞ LÝ THUYẾT 5](#_Toc76923447)

[2.1. Hệ điều hành android 5](#_Toc76923448)

[2.1.1. Giới thiệu 5](#_Toc76923449)

[2.1.2. Giao diện 5](#_Toc76923450)

[2.1.3. Kiến trúc hệ thống 6](#_Toc76923451)

[2.1.4. Ưu điểm và nhược điểm 9](#_Toc76923452)

[2.1.5. Quản lý bộ nhớ 9](#_Toc76923453)

[2.1.6. Bảo mật và riêng tư 10](#_Toc76923454)

[2.1.7. Các phiên bản của hệ điều hành Android 10](#_Toc76923455)

[2.2. các thành phần của ứng dụng Android Studio 11](#_Toc76923456)

[2.2.1. Tập tin cấu hình dự án AndroidManifest.xml 12](#_Toc76923457)

[2.2.2.Tập tin Java 13](#_Toc76923458)

[2.2.3. Tập tin res 14](#_Toc76923459)

[2.2.4. Tâp tin Grandle 14](#_Toc76923460)

[2.2.5. View 15](#_Toc76923461)

[2.2.6. Activity 16](#_Toc76923462)

[2.2.7. Intent 18](#_Toc76923463)

[2.2.8. Service 18](#_Toc76923464)

[2.2.9. Broadcast Receivers 20](#_Toc76923465)

[2.2.10. Content Provider 20](#_Toc76923466)

[2.2.11. Notification 20](#_Toc76923467)

[2.2.12. Tập tin Strings.xml 21](#_Toc76923468)

[2.3. Máy ảo Android 21](#_Toc76923469)

[2.4. Giới thiệu Firebase 22](#_Toc76923470)

[Chương III. PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 27](#_Toc76923471)

[3.1 Phân tích yêu cầu hệ thống 27](#_Toc76923472)

[3.1.1. Mô tả biểu đồ hoạt động chính của ứng dụng 27](#_Toc76923473)

[3.1.2. Chi tiết các quy trình 28](#_Toc76923474)

[3.2. Thiết kế giao diện và chức năng 30](#_Toc76923475)

[3.2.1. Thiết kế giao diện màn hình khởi động 30](#_Toc76923476)

[3.2.2. Thiết kế giao diện đăng nhập 30](#_Toc76923477)

[3.2.3. Thiết kế giao diện đăng ký 31](#_Toc76923478)

[3.2.4. Thiết kế giao diện trang chủ 32](#_Toc76923479)

[3.2.5. Thiết kế giao diện bottom navigation 32](#_Toc76923480)

[3.2.6. Thiết kế giao diện bài đăng 33](#_Toc76923481)

[3.2.7. Thiết kế giao diện thêm bài đăng 33](#_Toc76923482)

[3.2.8. Thiết kế giao diện chi tiết bài đăng 34](#_Toc76923483)

[3.2.9. Thiết kế giao diện danh sách người dùng 35](#_Toc76923484)

[3.2.10. Thiết kế giao diện trang cá nhân 35](#_Toc76923485)

[3.2.11. Thiết kế giao diện nhắn tin 36](#_Toc76923486)

[3.2.12. Thiết kế giao diện danh sách tin nhắn 37](#_Toc76923487)

[3.2.13. Thiết kế giao diện danh sách thông báo 37](#_Toc76923488)

[3.3. Kết quả thực nghiệm 38](#_Toc76923489)

[3.3.1. Giao diện màn hình khởi động 38](#_Toc76923490)

[3.3.2. Giao diện đăng nhập 38](#_Toc76923491)

[3.2.3. Giao diện khôi phục tài khoản 39](#_Toc76923492)

[3.2.4. Giao diện đăng ký 39](#_Toc76923493)

[3.2.5. Giao diện trang chủ 40](#_Toc76923494)

[3.2.6. Giao diện chi tiết bài đăng 40](#_Toc76923495)

[3.2.7. Giao diện danh sách người dùng 41](#_Toc76923496)

[3.2.8. Giao diện trang cá nhân 42](#_Toc76923497)

[3.2.9. Giao diện nhắn tin 42](#_Toc76923498)

[3.2.10. Giao diện danh sách tin nhắn 43](#_Toc76923499)

[3.2.11. Giao diện danh sách thông báo 43](#_Toc76923500)

[3.2.12. Giao diện quản lý tài khoản người dùng 44](#_Toc76923501)

[3.2.13. Giao diện quản lý danh sách bài đăng 44](#_Toc76923502)

[3.2.14. Giao diện quản lý bình luận người dùng 45](#_Toc76923503)

[3.2.15. Giao diện quản lý danh sách tin nhắn 45](#_Toc76923504)

[3.2.16. Giao diện quản lý tương tác 46](#_Toc76923505)

[3.2.17. Giao diện quản lý lưu trữ hình ảnh 46](#_Toc76923506)

[Chương IV: KẾT LUẬT VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI 47](#_Toc76923507)

[4.1. Kết luận 47](#_Toc76923508)

[4.2. Hướng phát triển đề tài 47](#_Toc76923509)

[Chương V: TÀI LIỆU THAM KHẢO 48](#_Toc76923510)

# DANH MỤC HÌNH

[Hình 1.1. Bài đăng tìm đồ trên Facebook 2](#_Toc76885348)

[Hình 1.2. Bài đăng tìm đồ trên diễn đàn 2](#_Toc76885349)

[Hình 2.1. Biểu tượng Android 5](file:///C:\Users\Admin\Desktop\DATN.docx#_Toc76885350)

[Hình 2.2. Giao diện người dùng tùy biến của Android 5](file:///C:\Users\Admin\Desktop\DATN.docx#_Toc76885351)

[Hình 2.3. Kiến trúc hệ điều hành Android 6](#_Toc76885352)

[Hình 2.4.1. Giao diện cửa sổ tệp MainActivity.java 11](#_Toc76885353)

[Hình 2.4.2. Giao diện cửa sổ tệp activity\_main.xml 11](#_Toc76885354)

[Hình 2.5. Tập tin có định dạng Java 13](#_Toc76885355)

[Hình 2.6. Tập tin res 14](#_Toc76885356)

[Hình 2.7. Tập tin có định dạng Grandle 14](#_Toc76885357)

[Hình 2.8. Sơ đồ View 15](#_Toc76885358)

[Hình 2.9. Vòng đời của Activity 17](#_Toc76885359)

[Hình 2.10. Sơ đồ hoạt động của Intent giữa 2 Activity 18](#_Toc76885360)

[Hình 2.11. Sơ đồ chuyển đổi trạng thái của Service 19](#_Toc76885361)

[Hình 2.12. Thông báo trong ứng dụng Android 21](#_Toc76885362)

[Hình 2.13. Giao diện máy ảo Android 22](#_Toc76885363)

[Hình 2.14. Firebase kết nối Android Studio 22](#_Toc76885364)

[Hình 2.15. Dịch vụ Firebase 23](#_Toc76885365)

[Hình 3.1. Biểu đồ chức năng người dùng 27](#_Toc76885366)

[Hình 3.2. Thiết kế giao diện màn hình khởi động 30](#_Toc76885367)

[Hình 3.3. Thiết kế giao diện đăng nhập 31](#_Toc76885368)

[Hình 3.4. Thiết kế giao diện đăng ký 31](#_Toc76885369)

[Hình 3.5. Thiết kế giao diện trang chủ 32](#_Toc76885370)

[Hình 3.6. Thiết kế giao diện bottom navigation 32](#_Toc76885371)

[Hình 3.7. Thiết kế giao diện bài đăng 33](#_Toc76885372)

[Hình 3.8. Thiết kế giao diện thêm bài đăng 34](#_Toc76885373)

[Hình 3.9. Thiết kế giao diện chi tiết bài đăng 35](#_Toc76885374)

[Hình 3.10. Thiết kế giao diện danh sách người dùng 35](#_Toc76885375)

[Hình 3.11. Thiết kế giao diện trang cá nhân 36](#_Toc76885376)

[Hình 3.12. Thiết kế giao diện nhắn tin 36](#_Toc76885377)

[Hình 3.13. Thiết kế giao diện danh sách tin nhắn 37](#_Toc76885378)

[Hình 3.14. Thiết kế giao diện danh sách thông báo. 37](#_Toc76885379)

[Hình 3.15. Giao diện màn hình chính khởi động 38](#_Toc76885380)

[Hình 3.16. Giao diện đăng nhập 38](#_Toc76885381)

[Hình 3.17.Giao diện khôi phục tài khoản 39](#_Toc76885382)

[Hình 3.18. Giao diện đăng ký 39](#_Toc76885383)

[Hình 3.19. Giao diện trang chủ 40](#_Toc76885384)

[Hình 3.20. Giao diện chi tiết bài đăng 41](#_Toc76885385)

[Hình 3.21. Giao diện danh sách người dùng 41](#_Toc76885386)

[Hình 3.22. Giao diện trang cá nhân 42](#_Toc76885387)

[Hình 3.23. Giao diện nhắn tin 42](#_Toc76885388)

[Hình 3.23. Giao diện danh sách tin nhắn 43](#_Toc76885389)

[Hình 3.24. Giao diện thông báo 43](#_Toc76885390)

[Hình 3.25. Giao diện quản lý tài khoản người dùng 44](#_Toc76885391)

[Hình 3.26. Giao diện quản lý danh sách bài đăng 44](#_Toc76885392)

[Hình 3.27. Giao diện quản lý bình luận người dùng 45](#_Toc76885393)

[Hình 3.28. Giao diện quản lý danh sách tin nhắn 45](#_Toc76885394)

[Hình 3.29. Giao diện quản lý tương tác 46](#_Toc76885395)

[Hình 3.30. Giao diện quản lý lưu trữ hình ảnh 46](#_Toc76885396)

# DANH MỤC BẢNG

[Bảng 2.1. Đánh giá ưu và nhược điểm của hệ điều hành Android 9](#_Toc76925033)

[Bảng 2.2. Các phiên bản của hệ điều hành Android 10](#_Toc76925034)

# DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT

|  |  |
| --- | --- |
| SMS  XML  SDK  API  JVM  CDN  IDE  HTTP  RAM | Short Message Service  Extensible Markup Language  Software Development Kit  Application Programming Interface  Java Virtual Machine  Content Delivery Network  Integrated Development Environment  HyperText Transfer Protocol  Random Access Memory |

# 

# LỜI MỞ ĐẦU

Ngày nay Công nghệ thông tin ngày càng phát triển mạnh mẽ và có những bước tiến đáng kể, đặc biệt là sự xuất hiện và thay đổi của điện thoại thông minh. Hầu như mỗi người đều cần một cái điện thoại để giúp cuộc sống tiện lợi hơn. Đối với sinh viên trường Đại học Nha Trang, trong quá trình học không tránh khỏi việc làm mất đồ dùng và vấn đề là làm sao để tìm lại chúng một cách nhanh chóng. Chính vì thế, “***Xây dựng ứng dụng di động tìm đồ thất lạc cho sinh viên Đại học Nha Trang***” được tạo ra nhằm giúp sinh viên thuận tiện hơn trong việc tìm lại đồ dùng mà không cần mất nhiều thời gian. Ngoài ra, hiện nay hệ điều hành Android với các thiết bị di động chạy android ngày càng phát triển và phổ biết. Nên việc xây dụng các ứng dụng chạy trên hệ điều hành android càng cần thiết.

Nội dung của toàn văn báo cáo bao gồm:

**Chương 1:** Tổng quan vấn đề

**Chương 2:** Cơ sở lý thuyết

**Chương 3:** Phân tích, thiết kế hệ thống

**Chương 4:** Kết luận và hướng phát triển đề tài

**Chương 5:** Tài liệu tham khảo

# CHƯƠNG I: TỔNG QUAN VẤN ĐỀ

## 1.1. Tính cấp thiết của đề tài

Điện thoại thông minh là một thiết bị không thể thiếu trong mọi lĩnh vực cuộc sống của con người. Với điện thoại thông minh, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này đã thúc đầy sự khai sinh và phát triển của thương mại điện tử và chính phủ điện từ trên khắp thể giới, làm biến đối đáng kể bộ mặt văn hóa, nâng cao chẩt lượng cuộc sống con người.

|  |  |
| --- | --- |
| Hình 1.1. Bài đăng tìm đồ trên Facebook  ([www.facebook.com/groups/271001000263174 /posts/634010140628923](http://www.facebook.com/groups/271001000263174%20/posts/634010140628923)) | Hình 1.2. Bài đăng tìm đồ trên diễn đàn  ([nhatrangclub.vn/threads/tim-giay-to-danh-roi.2190969/](https://nhatrangclub.vn/threads/tim-giay-to-danh-roi.2190969/) ) |

Trong thời gian học tại Trường Đại học Nha Trang, hầu như mỗi sinh viên đều trải qua việc làm mất hoặc nhặt được đồ dùng. Cách giải quyết truyền thống là sinh viên tự đi tìm đồ, dán các bài đăng tìm đồ tại bảng tin trường vì thế mất nhiều thời gian. Bên cạnh đó sinh viên có thể sử dụng điện thoại thông minh để đăng bài tại các nhóm sinh viên trên Facebook (Hình1.1.1), các diễn đàn tại Nha Trang (Hình 1.1.2). Cách giải quyết này tiện lợi hơn cách truyền thống, nhưng song song với đó có những bất cập như: các nhóm trên Facebook bao gồm rất nhiều loại bài đăng khác nhau rất khó để tìm đồ nhanh chóng.

Với những lí do trên, việc cài đặt “ứng dụng tìm đồ thất lạc” với những chức năng đơn giản, dễ sử dụng đặc biệt chỉ hoạt động trong trường Đại học Nha trang là nhu cầu cần thiết với mỗi sinh viên. Ứng dụng chỉ tập trung vào chức năng tìm và trả đồ dùng vì thế sẽ tiết kiệm rất nhiều thời gian cho sinh viên. Vì vậy, sẽ rất thiếu sót nếu sinh viên chưa cài đặt ứng dụng tìm đồ thất lạc cho điện thoại của mình.

## 1.2. Mục tiêu đề tài

Để tìm đồ thất lạc hoặc nhặt được đồ dùng, sinh viên thường đăng bài trên các diễn đàn, hội nhóm trên Facebook. Tuy nhiên các diễn đàn và hội nhóm này bao gồm nhiều loại bài đăng khác nhau. Vì thế sẽ không tối ưu cho mục đích chính là tìm và trả đồ. “Ứng dụng tìm đồ thất lạc cho sinh viên Trường Đại học Nha Trang” được tạo ra nhằm giải quyết 2 vấn đề chính trên và giúp việc tìm kiếm đồ dùng của sinh viên trở nên nhanh chóng và dễ dàng hơn.

## 1.3. Giải pháp thực hiện

Để xây dựng ứng dụng hoàn chỉnh cần:

Thu thập thông tin các địa điểm trong Trường Đại học Nha Trang và các đồ dùng thường xuyên của sinh viên.

Tìm hiểu môi trường lập trình Android Studio và liên kết với Firebase.

Tìm hiểu xây dựng các chức năng chính của ứng dụng như tìm kiếm, đăng bài, liên hệ người dùng, nhận thông báo từ ứng dụng.

Tìm hiểu môi trường chạy ứng dụng trên thiết bị thật.

## 1.4. Đối tượng và phạm vi nghiên cứu

**Đối tượng nghiên cứu**

Đối tượng nghiên cứu: Sinh viên Trường Đại học Nha Trang

**Phạm vi nghiên cứu**

Phạm vi không gian: Nghiên cứu và tìm hiểu tại Trường Đại học Nha Trang

Phạm vi thời gian: Nghiên cứu được thực hiện trong khoảng thời gian từ 15/03/2021 - 26/06/2021.

Phạm vi nội dung: Thu thập thông tin về những đồ dùng chủ yếu của sinh viên sau đó cập nhật thông tin đưa vào cơ sở dữ liệu để thiết kế ứng dụng.

## 1.5. Ý nghĩa lí luận và thực tiễn

Thông qua việc thực hiện đồ án này, giúp em hệ thống và củng cố được các vấn đề lí luận cơ bản về ứng dụng của công nghệ thông tin trong các hoạt động đăng bài chia sẻ thông tin trên ứng dụng di động, góp phần cho việc phát triển ứng dụng ngày càng logic, hiện đại và chuyên nghiệp hơn.

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## 2.1. Hệ điều hành android

### 2.1.1. Giới thiệu

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Hình 2.1.1 là biểu tượng của hệ điều hành Android (Elgin-Ben, August 2005). Ban đầu, Android được phát triển bởi Android, Inc. với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005 (Elgin-Ben, August 2005). Android ra mắt vào năm 2007 cùng với tuyên bố thành lập liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động (Alliance, November 2012). Chiếc điện thoại đầu tiên chạy Android được bán vào năm 2008. Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache (GSM Arena, 2018). Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi (Shankland-Stephen, November 2007). Tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

( [https://developer.android.com/](https://developer.android.com/guide/platform) )

Hình 2.1. Biểu tượng Android

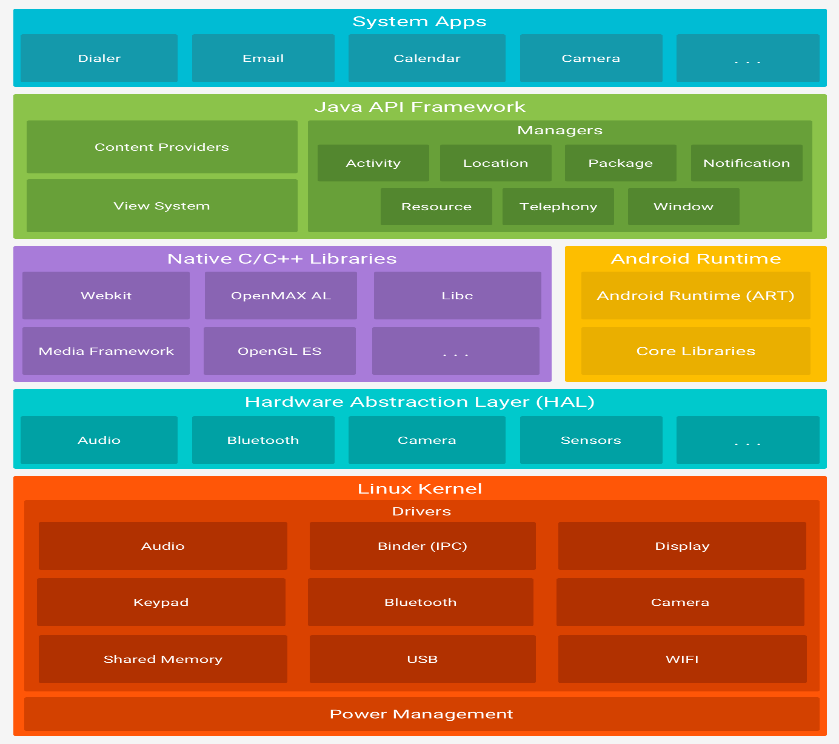
### 2.1.2. Giao diện

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop trên máy tính để bàn. Hình 2.2. hiển thị giao diện người dùng tùy biến của Anroid (Touch Devices, September 2012). Màn hính chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget). Giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone hay iOS. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh. Google đưa ra các bản cập nhật lớn cho Android theo chu kỳ từ 6 đến 9 tháng, mà phần lớn thiết bị đều có thể nhận được qua sóng không dây.

(<https://www.androidcentral.com/>)

Hình 2.2. Giao diện người dùng tùy biến của Android

### 2.1.3. Kiến trúc hệ thống



Hình 2.3. Kiến trúc hệ điều hành Android

( <https://developer.android.com/guide/platform> )

**Linux Kernel**

Hệ điều hành Android được phát triển dựa trên nhân linux, cụ thể là nhân linux phiên bản 2.6, điều đó được thể hiện ở mức dưới cùng. Tất cả mọi hoạt động của điện thoại muốn thi hành được thì đểu được thực hiện ở mức cấp thấp ở lớp này bao gồm quản lý bộ nhớ, giao tiếp với phần cứng, thực hiện bảo mật, quản lý tiền trình. Tuy được phát triển dựa vào nhân linux đã được nâng cấp và sửa đổi rất nhiều để cho phù hợp với tính chất của những thiết bị cầm tay như hạn chế về bộ vi xử lý, dụng lượng bộ nhớ, kích thước màn hình, nhu cầu kết nối mạng không dây (Raja-Haroon, May 2011). Các thành phần bao gồm:

Display Driver: Điều khiển việc hiển thị lên màn hình cũng như thu nhận những điều kiện của người dùng lên màn hình (di chuyển cảm ứng…)

Camera Driver: Điều khiển hoạt động của camera, nhận luồng dữ liệu từ camera trả về.

Bluetooth Driver: Điều khiển thiết bị phát và thu sóng Bluetooth.

USB Driver: Quản lý hoạt động của các cổng giao tiếp USB.

Keypad Driver: Điều khiển bàn phím.

Wifi Driver: Chịu trách nhiệm về việc thu phát sóng wifi.

Audio Driver: Điều khiển các bộ thu phát âm thanh, giải mã các tín hiệu dạng audio thành tín hiệu và ngược lại.

Binder IPC Driver: Chịu trách nhiệm về việc kết nối và liên lạc với mạng vô tuyến như CDMA,GSM, 3G, 4G, E để đảm bảo những chức năng truyền thông được thực hiện.

M-System Driver: Quản lý việc đọc ghi… lên các thiết bị nhớ như thẻ SD, flash.

Power Madagement: Giám sát việc tiêu thụ điện năng.

**Hardware Abstraction Layer (HAL)**

HAL là một lớp trừu tượng phần cứng cung cấp những lời gọi hàm tới phần cứng để thực hiện những công việc nhất định. HAL bao gồm nhiều module khác nhau, mỗi module thực hiện một giao diện cho một loại thành phần phần cứng cụ thể, chẳng hạn như module camera hoặc bluetooth (Chris von Eitzen, December 2011).

**Android Runtime (ART)**

ART chứa các thư viện chương trình viết bằng ngôn ngữ Java có thể hoạt động. Phần này có 2 bộ phận tương tự như mô hình chạy Java trên máy tính thường bao gồm các thư viện lõi (Core Library), chứa các lớp như JAVA IO, Collection, File Access và máy ảo Java (Dalvik Virtual Machine). Mặc dù cũng được viết từ ngôn ngữ Java nhưng một ứng dụng Java của hệ điều hành Android không được chạy bằng JRE của Sun( nay là Oracle) (JVM) mà là chạy bằng máy ảo Dalvik do Google phát hành (Whittaker-Zack, October 2012).

**Native C/C++ Libraries**

Bao gồm các thư viện được viết bằng C/C++ để các phần mềm có thể sử dụng (Yun Qing-Liau, October 2012) các thư viện đó được tập hợp thành một nhóm như:

Thư viện hệ thống (System C Library): thư viện dựa trên chuẩn C, được sử dụng bởi hệ điều hành.

Thư viện Media (Media Library): có nhiều code để hỗ trợ việc phát triển và ghi các loại định dạng âm thanh, hình ảnh, video thông dụng.

Thư viện Web (LibWebCode): đây là thành phần xem nội dung trên web, được sử dụng để xây dựng phầm mềm duyệt web (Android Browser) cũng như để các ứng dụng khác có thể nhúng vào. Nó cực kỳ mạnh, hỗ trợ được nhiều công nghệ mạnh mẽ như HTML5, JavaScript, CSS, DOM, AJAX…

Thư viện SQLite: Hệ cơ sở dữ liệu để các ứng dụng có thể sử dụng.

**Java API Framework**

Tổng hợp các tính năng có sẵn của hệ điều hành Android thông qua các API được viết bằng ngôn ngữ Java.

Activity Manager: Quản lý các chu kỳ sống của một ứng dụng cũng như cung cấp công cụ điều khiển các Activity.

Telephony Manager: Cung cấp công cụ để thực hiện liên lạc như gọi điện thoại.

XMPP Service: Cung cấp công cụ để liên lạc trong thời gian thực.

Location Manager: Cho phép xác định vị trí điện thoại dựa vào hệ thống định vị toàn cầu GPS và Google Maps.

Window Manager: Quản lý việc xây dựng và hiển thị các giao diện người dùng cũng như tổ chức quản lý các giao diện giữa các ứng dụng.

Resource Manager: Quản lý tài nguyên tĩnh của các ứng dụng bao gồm các file hình ảnh, âm thanh, giao diện, chuỗi xử lý. (Những thành phần không được viết bời ngôn ngữ lập trình).

Notication Manager: Quản lý việc hiển thị các thông báo như báo như: tin nhắn, E-mail mới.

**System Apps**

Đây là lớp ứng dụng giao tiếp với người dùng, bao gồm các ứng dụng như: gọi điện, quản lý danh bạ, duyệt web, nhắn tin, lịch làm việc, E-mail, bản đồ, quay phim, chụp ảnh, từ điển… Các chương trình có đặc điểm: Viết bằng Java, phần mở rộng là apk. Khi mỗi ứng dụng được chạy, nó có một phiên bản Vitual Machine được dựng lên để phục vụ cho nó. Nó có thể là một Active Program, chương trình có giao diện với người dùng hoặc là chương trình chạy nền. Android là hệ điều hành đa nhiệm, điều đó có nghĩa là trong cùng một thời điểm, có thể có nhiều chương trình cùng chạy một lúc. Tuy nhiên, với một ứng dụng thì có duy nhất một thực thể (instance) được phép chạ. Điều đó có tác dụng hạn chế sự làm dụng tài nguyên, giúp hệ thống hoạt động tốt hơn. Các ứng dụng được gắn số ID của người sử dụng nhằm phân định quyền hạn khi sử dụng tài nguyên, cấu hình phần cứng và hệ thống. Android là một hệ điều hành có tính mở, khác với nhiều hệ điều hành di động khác. Android cho phép một ứng dụng của bên thứ ba được phép chạy nền. Các ứng dụng đó chỉ có một hạn chế nhỏ đó là nó không được phép sử dụng quá 5~10% công suất CPU, điều đó nhằm để tránh độc quyền trọng việc sử dụng CPU.

### 2.1.4. Ưu điểm và nhược điểm

Bảng 2.1. Đánh giá ưu và nhược điểm của hệ điều hành Android

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Có khả năng tuỳ biến cao, tùy ý chỉnh sửa. | Dễ nhiễm phần mềm độc hại, virus |
| đa dạng sản phẩm, giá cả hợp lý. | Khó kiểm soát chất lượng |
| Kho ứng dụng Google Play Store đồ sộ. | Sự phân mảnh lớn |
| Thân thiện và dễ sử dụng. | Cập nhật không tự động với tất cả thiết bị. |
| Khả năng đa nhiệm | Lỗi bảo mật hệ thống |

### 2.1.5. Quản lý bộ nhớ

Để giảm tối đa điện năng tiêu thụ hệ điều hành Android đã thiết kế quản lý bộ nhớ (RAM) ngược lại với bộ nhớ máy tính hoặc laptop hoạt động mãi mãi khi có nguồn điện năng cung cấp. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động đóng băng trong bộ nhớ trong khi ứng dụng về mặt kỹ thuật vẫn hoạt động, những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào như năng lượng pin hay năng lượng xử lý. Vì thế làm tăng khả năng phản hồi của thiết bị Android vì ứng dụng không nhất thiết phải đóng rồi mở lại từ đầu. Đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

Android quản lý các ứng dụng trong bộ nhớ một cách tự động, khi bộ nhớ không đủ, hệ thống sẽ đóng các ứng dụng đang đóng băng. Tiến trình này được thiết kế ẩn đi với người dùng, để người dùng không cần phải quản lý bộ nhớ hoặc tự tay đóng các ứng dụng. Tuy nhiên, sự che giấu này của hệ thống quản lý bộ nhớ Android đã dẫn đến sự thịnh hành của các ứng dụng quản lý bộ nhớ của bên thứ ba trên của hàng Google Play, những ứng dụng này được cho là có hại nhiều hơn có lợi.

### 2.1.6. Bảo mật và riêng tư

Các ứng dụng Android chạy trong một “sandbox” là một vực riêng rẽ với hệ thống và không được tiếp cận đến phần còn lại của tài nguyên hệ thống, trừ khi người dùng trao quyền truy cập một cách công khai khi cài đặt. Trước khi cài đặt ứng dụng, cửa hàng Chplay sẽ hiển thị tất cả các quyền mà ứng dụng đòi hỏi: ví dụ như một trò chơi cần phải kích hoạt bộ rung hoặc sử dụng bộ nhớ lưu trữ (thẻ nhớ), quyền truy cập dịch vụ tin nhắn ngắn (Short Message Service - SMS) hoặc tiếp cận danh bạ điện thoại.

Điện thoại thông minh sử dụng hệ điều hành Android có khả năng báo cáo vị trí điểm truy cập Wi-Fi, phát hiện di chuyển của người dùng, để xây dựng những cơ sở dữ liệu có chứa vị trí của hàng trăm triệu điểm truy cập. Những cơ sở dữ liệu này tạo nên một bản đồ điện tử để tìm vị trí điện thoại thông minh, cho phép chúng chạy các ứng dụng như Google Latitude, Facebook Places gửi những đoạn quảng cáo dựa trên vị trí.

### 2.1.7. Các phiên bản của hệ điều hành Android

Bảng 2.2. Các phiên bản của hệ điều hành Android

(<https://www.javatpoint.com/android-versions>)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên phiên bản** | **Số hiệu** | **Năm phát hành** | **Mức giao thức** |
| Cupcake | 1.5 | 27/04/2009 | 3 |
| Donut | 1.6 | 15/09/2009 | 4 |
| Eclair | 2.0 – 2.1 | 26/10/2009 | 5 - 7 |
| Froyo | 2.2 – 2.2.3 | 20/5/2010 | 8 |
| Gingerbread | 2.3 – 2.3.7 | 06/12//2010 | 9 - 10 |
| Honeycomb | 3.0 – 3.2.6 | 22/2/2011 | 11 - 13 |
| IceCreamSandwich | 4.0 - 4.0.4 | 18/10/2011 | 14 - 15 |
| Jelly Bean | 4.1 – 4.3.1 | 09/07/2012 | 16 - 18 |
| KitKat | 4.4 – 4.4.4 | 31/10/2013 | 19 - 20 |
| Lollipop | 5.0 – 5.1.1 | 12/11/2014 | 21 - 22 |
| Marshmallow | 6.0 – 6.0.1 | 05/10/2015 | 23 |
| Nougat | 7.0 – 7.1.2 | 22/08/2016 | 24 - 25 |
| Oreo | 8.0 – 8.1 | 21/08/2017 | 27 - 27 |
| Pie | 9.0 | 06/08/2018 | 28 |
| Android 10 | 10.0 | 03/09/2019 | 29 |
| Android 11 | 11 | 30/09/2020 | 30 |

## 2.2. các thành phần của ứng dụng Android Studio

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android Studio được công bố vào ngày 16 tháng 5 năm 2013 tại hội nghị Google I/O. Phiên bản đầu tiên được phát hành vào tháng 6 năm 2014 (Android Developer, June 2015). Android Studio là môi trường phát triển ứng dụng tốt nhất cho ứng dụng của Android.

Cấu trúc một dự án Android Stuido bao gồm 2 cửa sổ chính: (1) là cửa sổ tệp “\*.java” để xử lý code của chương trình, (2) là cửa sổ tệp “\*.xml” để xử lý giao diện của chương trình.

Hình 2.2.1 và hình 2.2.2 là giao diện của Android Studio với ví dụ in ra chữ “Hello word”. Trong đó cửa sổ activity\_main.xml dùng để thiết kế giao diện hiển thị và cửa sổ MainActivity.java để xử lý chức năng hiển thị của cửa sổ trên.

|  |  |
| --- | --- |
|  |  |
| Hình 2.4.1. Giao diện cửa sổ tệp MainActivity.java | Hình 2.4.2. Giao diện cửa sổ tệp activity\_main.xml |

Hình 2.2.1. Cửa sổ xử lý code trong tệp MainActivity.java bao gồm các thành phần: (1) lớp MainActivity được kế thừa từ lớp AppCompatActivity bao gồm các phương thức khởi tạo ứng dụng, (2) phương thức onCreate là phương thức khởi chạy của Activity chứa các hành động khi khởi động ứng dụng, (3) hàm setContentView dùng để thiết lập cho giao diện hiển thị R.layout.activity\_main nằm trong thư mục res/layout.

Hình 2.2.2. Cửa sổ xử lý giao diện tệp activity\_main.xml được viết theo chuẩn ngôn ngữ đánh dấu mở rộng(Extensible Markup Language-XML). Giao diện được thiết kế với ContraintLayout chứa thẻ TextView hiển thị dòng chữ “Hello Word”

### 2.2.1. Tập tin cấu hình dự án AndroidManifest.xml

Trong bất kì một dự án Android nào khi tạo ra đều có một tệp định dạng manifest, tệp này được dùng để đăng ký các màn hình sử dụng trong ứng dụng Android, các permission cũng như các giao diện, chủ để (theme) cho ứng dụng. Đồng thời tập tin manifest.xml cũng chứa thông tin về phiên bản của bộ công cụ phát triển phần mềm (Software Development Kit–SDK) cũng như main activity sẽ chạy đầu tiên. Tập tin này được tự động sinh ra khi tạo một Android project.

Cấu trúc của một manifest.xml hiển thị như sau:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.doantotnghiep\_firebase">

<application  
 android:label="Tìm đồ thất lạc NTU"  
 android:hardwareAccelerated="true"  
 android:allowBackup="true"  
 android:icon="@drawable/lgo"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.DoAnTotNghiep\_Firebase">

<activity  
 android:name=".BottomNavigation"  
 android:theme="@style/AppTheme111">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
</activity>

<service  
 android:name=".ShowNotifications.FirebaseService"  
 android:enabled="true"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="com.google.firebase.INSTANCE\_ID\_SERVICE" />  
 </intent-filter>  
</service>

</application>  
</manifest

**<application></application>:** Thẻ bao quanh các thành phần có liên quan trong ứng dụng. Thuộc tính Android:icon sẽ trỏ đến các Icon có sẵn trong thư mục res/drawable-hdpi. Ứng dụng này sẽ sử dụng hình ic\_launcher làm Icon.

**<activity></activity>:** Thẻ sử dụng để nêu chi tiết về một Activity. Thuộc tính android:name được sử dụng để mô tả chi tiết về tên class của một Activity. Thuộc tính android:label xác định một chuỗi ký tự được sử dụng là nhãn cho Activity đó

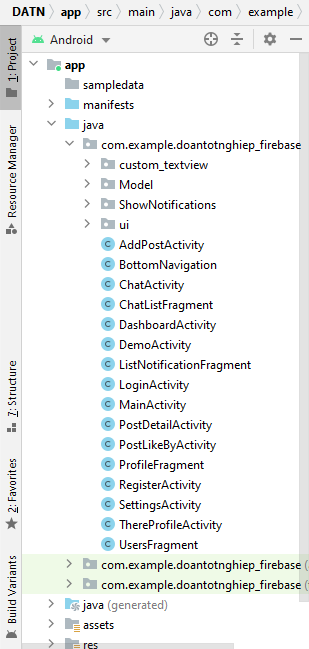
**<action />:** Thuộc tính Android.intent.action.main để cho Android biết rằng Activity này sẽ được khởi động lên đầu tiên khi chạy ứng dụng.

**<category />:** Thuộc tính Android.intent.category.launcher để cho Android biết ứng dụng có thể được chạy từ một Icon trên thiết bị.

**@string:** Sẽ được chỉ định đến file strings.xml, @string/app\_name sẽ được dẫn đến chuỗi app\_name nó là HelloWord.

### 2.2.2.Tập tin Java

Tập tin với định dạng java là một tập tin tự động sinh ra ngay khi tạo ứng dụng, tập tin này được sử dụng để quản lý thuộc tính được khai báo trong tập tin ngôn ngữ đánh dấu mở rộng (Extensible Markup Language – XML) của ứng dụng và các tài nguyên hình ảnh.



Hình 2.5. Tập tin có định dạng Java

### 2.2.3. Tập tin res

Tập tin có định dạng res chứa các tài nguyên mà ứng dụng sẽ sử dụng đến được tổ chức thành các thư mục con như:

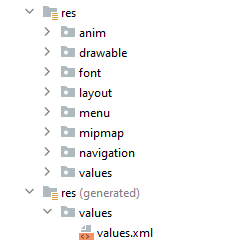
drawable/ : Lưu các đối tượng đồ họa ( ảnh dạng png hoặc xml).

layout/: lưu trữ các tập tin có định dạng xml hiển thị thành phần, bố cục của các thành phần hiển thị được trên màn hình.

mipmap/: Lưu các đối tượng hình ảnh như: biểu tượng ứng dụng ic\_launcher.

values/: chứa các tập tin như color.xml, dimens.xml, string.xml, style.xml, đây là

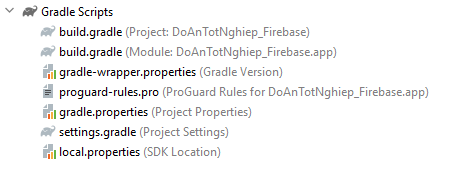
các tập tin xml định nghĩa các giá trị có thể sử dụng trong ứng dụng như màu sắc, kích thước, chuỗi, các giao diện màn hình.



Hình 2.6. Tập tin res

### 2.2.4. Tâp tin Grandle

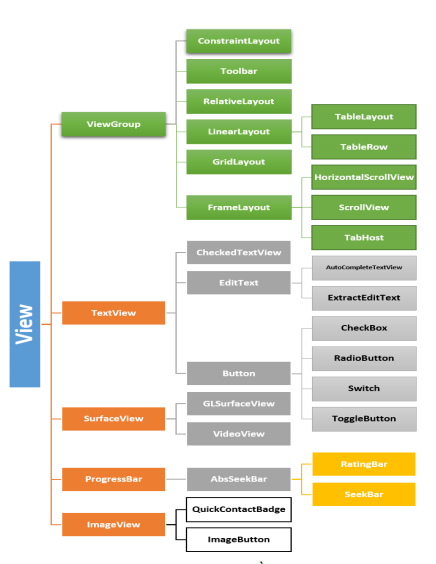
Tập tin Gradle Scrips chứa nhiều nhánh con như build.gradle, local.properties,... là nơi thiết lập các thông số để Gradle xây dựng ứng dụng từ mã nguồn, kết hợp tài nguyên, phân tích xml .. rồi kết hợp tất cả lại với nhau để tạo ra ứng dụng chạy trên máy ảo Java (Java Virtual Machine – JVM) .



Hình 2.7. Tập tin có định dạng Grandle

### 2.2.5. View

Trong một ứng dụng Android, giao diện người dùng được xây dựng từ các đối tượng khung hiển thị (View) và tập hợp các khung hiển thị (ViewGroup). Có nhiều kiểu View và ViewGroup. Mỗi kiểu lại được kế thừa từ lớp View và tất cả các kiểu đó được gọi là các tiện ích mở rộng(Widget). Widget đều có chung các thuộc tính cơ bản như: cách trình bày vị trí, nền, kích thước, lề… Tất cả những thuộc tính chung này được thể hiện trong đối tượng View. Trong Android Platform, các màn hình (screen) luôn được bố trí theo một kiểu cấu trúc phân cấp. Một màn hình là một tập hợp các giao diện (Layout) và các widget được bố trí có thứ tự. Để thể hiện một màn hình, trong hàm onCreate của mỗi màn hình giao diện (Activity) cần phải được gọi hàm setContentView(R.layout.main); hàm này sẽ truyền tải (load) giao diện từ file XML lên để phân tích thành mã nguồn (bytecode).



Hình 2.8. Sơ đồ View

(<https://developer.android.com/topic/libraries/view-binding>)

### 2.2.6. Activity

Activity là một giao diện đồ họa mà người dùng có thể thực hiện mỗi khi kích hoạt ứng dụng. Ứng dụng có thể có nhiều giao diện đồ họa khác nhau (Activity) và có thể gọi lẫn nhau. Activity được dẫn xuất từ lớp Android.app.activity.

Ứng dụng có thể có một hoặc nhiều giao diện đồ họa tùy theo yêu cầu của ứng dụng. Một giao diện đồ họa được đánh dấu là giao diện đồ họa đầu tiên khi khởi động ứng dụng.

Mỗi giao diện đồ họa được đưa ra một cửa sổ mặc định để vẽ vào. Thông thường, cửa sổ này sẽ lấp đầy màn hình, nhưng ta có thể tùy chọn để nó nhỏ hơn màn hình và nằm trên các cửa sổ khác. Ngoài ra nó cũng có thể có thêm các cửa sổ con khác như là hộp thoại,… Nội dung của cửa sổ của hoạt động được cung cấp bởi một hệ thống cấp bậc các View (là đối tượng của lớp Views). Android có một số View đã xây dựng sẵn mà người dùng có thể sử dụng. Gồm có các: Buttons, text fields, scroll bars, menu items, check box, image view, image view button, video view,... Một cây phân cấp View được đặt trong một cửa sổ của Activity bằng phương thức Activity.setContentView(). ContentView là đối tượng View ở gốc của cây phân cấp.

Các giao diện đồ họa trong hệ thống được quản lý bởi một cấu trúc dữ liệu ngăn xếp. Khi có một hoạt động được khởi tạo nó được đẩy vào trong ngăn xếp, chuyển sang trạng thái thực thi và hoạt trộng trước đó sẽ chuyển sang trạng thái chờ. Hoạt động này chỉ trở lại trạng thái kích hoạt khi mà hoạt động vừa khởi tạo kết thúc việc thực thi.

**Một giao diện đồ họa bao gồm 3 trạng thái chính:**

Active: Khi giao diện đồ họa nhận tương tác với người dùng.

Paused: Khi giao diện đồ họa không hoạt động nhưng vẫn hiển thị với người dùng.

Stopped: Khi một giao diện đồ họa hoàn toàn bị che khuất, nó sẽ rơi vào trạng thái Stopped. Tuy nhiên, nó vẫn còn lưu trữ toàn bộ thông tin trạng thái và nó thường bị hệ thống đóng lại khi có tình trạng thiếu bộ nhớ.

Khi chuyển giữa các trạng thái, ứng dụng sẽ gọi các hàm callback ứng với các bước chuyển:

- OnCreate(): Được gọi khi hoạt động được tạo ra lần đầu tiên.

- OnStart(): Được gọi khi hoạt động trở nên hữu hình so với người dùng.

- OnResume(): Được gọi khi hoạt động bắt đầu tương tác với người sử dụng.

- OnPause(): Được gọi để dừng các hoạt động hiện tại và nối lại các hoạt động trước đó.

- OnStop(): Được gọi khi hoạt động không còn hiển thị với người dùng.

- OnDestroy(): Được gọi trước khi hoạt động bị phá hủy bởi hệ thống (bằng tay hoặc bằng hệ thống để bảo tồn bộ nhớ).

- OnRestart(): Được gọi khi hệ thống đã được dừng lại và khởi động lại một lần nữa



Hình 2.9. Vòng đời của Activity

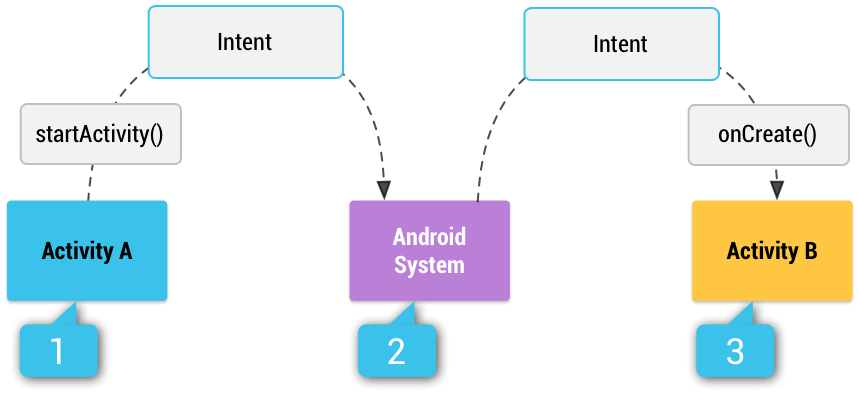
(<https://developer.android.com/activity-lifecycle> )

Hình 2.2.6 mô tả vòng đời của Activity bắt đầu từ khi Activity được kích hoạt sau đó được hệ thống đẩy vào ngăn xếp (Back Stack - BS). Hệ thống gọi các hàm onCreate(), onStart() và onResume(). Nếu một Activity khác chiếm quyền hiển thị, trạng thái của Activity hiện tại sẽ vào trạng thái onPause(). Nếu Activity khác làm cho Activity đang hoạt động không hiển thị thì onStop() được gọi. Nếu Activity đang hoạt động bị Activity khác chiếm quyền sau đó được quay lại thì onResume() được gọi. Nếu Activity đang ở trạng thái onStop() sau đó quay về Activity trước đó thì onRestart() được gọi. Nếu một Activity bị hủy bởi người dùng hoặc hàm finish() được gọi thì onDestroy() được kích hoạt và vòng đời của Activiy kết thúc.

### 2.2.7. Intent

Intent là nền tảng để truyền tải các thông báo. Intent được sử dụng để gửi các thông báo đi nhằm khởi tạo một giao diện đồ họa(Activity) hay dịch vụ(Service) để thực hiện công việc mong muốn.

Intent là một cấu trúc dữ liệu mô tả cách thức, đối tượng thực hiện của một giao diện đồ họa, là cầu nối giữa các giao diện đồ họa với nhau. Ứng dụng Android thường bao gồm nhiều giao diện đồ họa, mỗi giao diện đồ họa hoạt động độc lập với nhau và thực hiện những công việc khác nhau. Intent chính là người đưa thư, giúp các giao diện đồ họa có thể triệu gọi cũng như truyền các dữ liệu cần thiết tới một giao diện đồ họa khác.



Hình 2.10. Sơ đồ hoạt động của Intent giữa 2 Activity

(<https://developer.android.com/intents-filters> )

Hình 2.2.7. mô tả việc gửi dữ liệu từ Activity A đến Activity B. Activity A tạo ra một ý định (Intent) sau đó gọi hàm startActivity(). Hệ thống Android sẽ tìm tất cả ứng dụng (application) mà intent của ứng dụng đó phụ hợp với intent được gửi đi từ Activity A. Khi kết quả trùng khớp, hệ thống sẽ khởi động Activity B bằng cách gọi hàm onCreate() và gửi thông tin từ Activity A.

### 2.2.8. Service

Một Service (dịch vụ) là các đoạn mã được thực thi ngầm bởi hệ thống mà người sử dụng không thấy được. Service sử dụng để cập nhật dữ liệu, đưa ra các cảnh báo. Mỗi Service đều được mở rộng từ lớp cơ sở là Service trong gói Android.app.service. Có thể kết nối tới hoặc kích hoạt một Service thông qua Interface mà Service đưa ra.

Một ứng dụng có thể dễ dàng thực hiện liên kết tới một Service đang chạy (thậm chí khởi động nếu nó chưa thực thi) bằng phương thức Context.bindservice(). Khi đó Service này sẽ cung cấp cho ứng dụng cơ chế để giao tiếp với chúng thông qua giao diện gọi là IBinder (đối với dịch vụ phát nhạc có thể cho phép dừng hoặc chuyển qua bài nhạc kế tiếp).

**Vòng đời của một Service:**

Khi hệ thống có lời gọi tới phương thức Context.startService(). Trong trường hợp này, dịch vụ sẽ được thực hiện liên tục cho tới khi hệ thống gọi phương thức Context.stopService().

Khi các ứng dụng gọi phương thức Context.bindService() để tạo kết nối với dịch vụ (dịch vụ sẽ được khởi tạo nếu tại thời điểm đó nó đang không hoạt động). Ứng dụng sẽ nhận được một đối tượng IBinder do dịch vụ trả lại để có thể gọi các phương thức Callback phù hợp để truy cập tới các trạng thái của dịch vụ. Nếu do lời gọi Context.bindService() mà dịch vụ được khởi tạo thì nó sẽ được thực thi cho tới khi nào kết nối trên (tức là đối tượng IBinder) vẫn còn tồn tại.

Hình 2.2.8 Cho thấy hai phương thức để một dịch vụ có thể được khởi chạy trong hệ thống.



Hình 2.11. Sơ đồ chuyển đổi trạng thái của Service

Hình 2.2.8 giới thiệu dịch vụ xử lý thông báo, thông báo của ứng dụng này sẽ hiển thị trên điện thoại của người dùng.

### 2.2.9. Broadcast Receivers

Chương trình phát sóng (Broadcast Receiver) là một thành phần không làm gì ngoài việc nhận và đáp lại các thông báo Broadcast. Nhiều Broadcast khởi đầu trong mã hệ thống. Một ứng dụng có thể có một số Broadcast Receiver để đáp lại bất cứ thông báo nào mà nó cho là quan trọng. Tất cả các Receiver mở rộng từ lớp cơ sở BroadcastReceiver.

Broadcast Receiver không hiển thị một giao diện người dùng. Tuy nhiên chúng có thể bắt đầu một Activity để đáp lại thông tin mà chúng nhận hay chúng có thể sử dụng NotificationManager để cảnh báo người dùng. Notifications có thể lấy sự chú ý của người dùng bằng nhiều cách khác nhau như là: Sáng màn hình, hiển thị thông báo, rung thiết bị, bật âm thanh nào đấy,… Thông thường, chúng thường lấy một biểu tượng bền vững trong thanh trạng thái và đặt thông báo trên thanh trạng thái, nơi người dùng có thể nhận được thông báo.

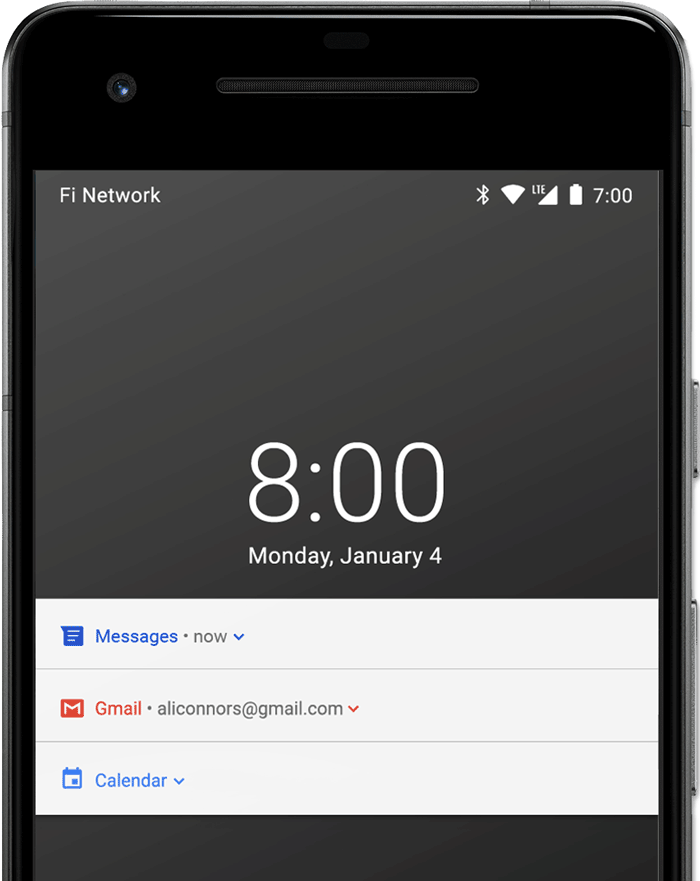
### 2.2.10. Content Provider

Content provider là thành phần cung cấp dữ liệu từ một ứng dụng đến một ứng dụng khác dựa trên các Request. Mỗi Request được xử lý bằng các phương thức của class ContentResolver.

Một Content Provider có thể sử dụng các cách lưu trữ dữ liệu khác nhau, dữ liệu có thể được lưu trữ trong cơ sở dữ liệu (databases), tệp (file), thậm chí thông qua mạng (Network). Mỗi ứng dụng Android chạy trong các tiến trình riêng của chính mình và nó có các điều khoản riêng của nó, điều mà giữ dữ liệu của ứng dụng ẩn với các ứng dụng khác. Tuy nhiên, thỉnh thoảng nó được yêu cầu chia sẻ dữ liệu đến các ứng dụng khác sử dụng Content Provider trong việc này rất hữu ích.

### 2.2.11. Notification

Thông báo (Notification) đưa ra các cảnh báo mà không làm cho các Activity phải ngừng hoạt động. Notification là một cơ chế cho phép tại một thời điểm nhất định trình bày một thông tin trên thanh trạng thái (Status bar) ở phía trên màn hình. Mỗi khi thanh trạng thái được mở ra thì một danh sách những Notification được hiển thị, khi người dùng chọn vào một Notification thì có thể một Activity được kích hoạt.



Hình 2.12. Thông báo trong ứng dụng Android

(<https://developer.android.com/notifications> )

Notification bao gồm các thành phần: (1) Biểu tượng , (2) Ticker text, (3) Notification title text, (4) Notification body text, (5) Intent.

### 2.2.12. Tập tin Strings.xml

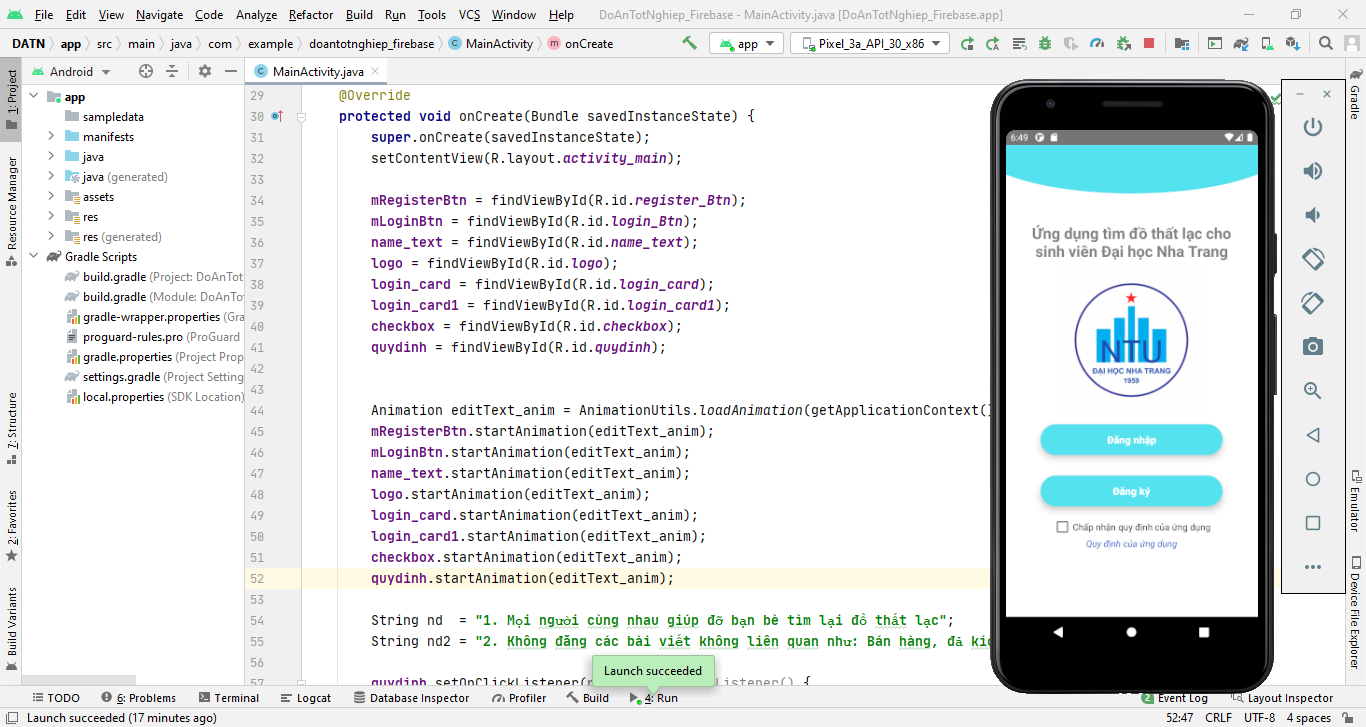
Strings.xml đây là tập tin được đặt trong thư mục res/values chứa tất cả các chuỗi kí tự mà ứng dụng sử dụng như tên của các buttons, các labels, các text.

<**resources**>  
 <**string name="app\_name"**>Tìm đồ thất lạc</**string**>  
 <**string-array name="sort\_types"**>  
 <**item**>Hội trường 1</**item**>  
 <**item**>Hội trường 2</**item**>  
 </**string-array**>  
</**resources**>

Thẻ <resource></resource> chứa các thẻ ký tự ứng dụng sử dụng. Thẻ <string name=”app\_name”></string> chứa tên của ứng dụng. Thẻ <string-array name = “sort\_types”></string-array> chứa các danh sách mục (item) hiển thị.

## 2.3. Máy ảo Android

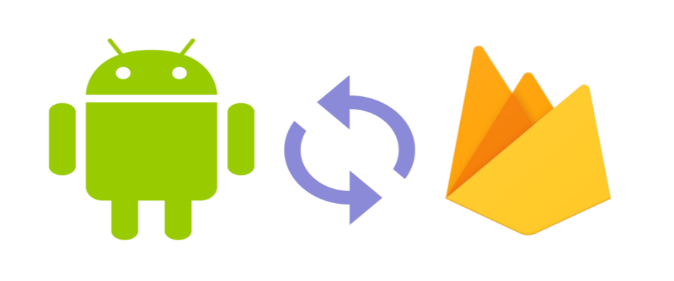
Máy ảo Android (Android Virtual Device - AVD) được hỗ trợ chính thức từ Google là một phần không thể thiếu khi xây dựng ứng dụng cho hệ điều hành Android, giúp thử nghiệm ứng dụng ngay trên máy tính. Máy ảo Android có tính ổn định cao, tiêu tốn bộ nhớ của máy tính ít hơn các máy ảo khác, hỗ trợ giả lập các loại thiết bị từ điện thoại, máy tính bảng, đồng hồ thông minh hay Android TV.



Hình 2.13. Giao diện máy ảo Android

## 2.4. Giới thiệu Firebase

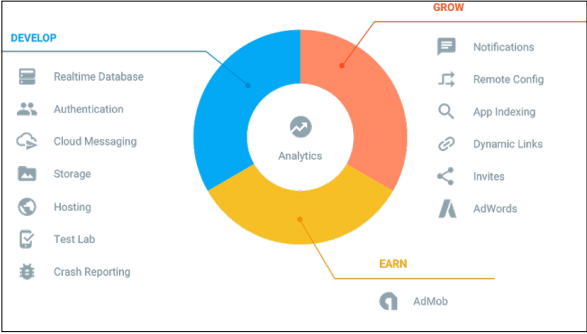
Firebase là một dịch vụ hế thống truy cập dữ liệu (backend) được Google cung cấp sẵn cho ứng dụng Mobile, với Firebase người dùng có thể rút ngắn thời gian phát triển, triển khai và thời gian mở rộng quy mô của ứng dụng mobile. Hỗ trợ cả 2 nền tảng Android và IOS, với các tính năng mạnh mẽ, đa năng, bảo mật và là dịch vụ cần thiết đâu tiên để xây dựng ứng dụng với hàng triệu người sử dụng (Culubic, October 2014).

~~~~

Hình 2.14. Firebase kết nối Android Studio

( <https://developer.android.com/studio/write/firebase> )

Firebase cung cấp cho người dùng công cụ Firebase Analytics và 2 nhóm sản phẩm chính tập trung vào 2 đối tượng: phát triển và kiểm thử các ứng dụng được thiết kế (Develop & test your app) và phân tích dữ liệu và tối ưu hóa trải nghiệm đối với người dùng (Grow & engage your audience).



Hình 2.15. Dịch vụ Firebase

(<https://firebase.google.com/support/guides/service-accounts> )

**Dịch vụ firebase Analytics:** Là một giải pháp miễn phí và phân tích không giới hạn. Quản lý hành vi người dùng và cá biện pháp từ một bảng điều khiển duy nhất. Phân thích thuộc tính và hành vi của người dùng trong bảng điều khiển đơn để đưa ra các quyết định sáng suốt về lộ trình sản phẩm. Nhận thông tin chi tiết về thời gian thực từ báo cáo hoặc xuất dữ liệu thô sự kiện của người dùng tới Google BigQuery để phân tích tùy chỉnh.

**Các dịch vụ phát triển và kiểm thử ứng dựng**

**Realtime Database:** Lưu trữ và đồng bộ dữ liệu người dùng thời gian thực. Các dữ liệu được lưu trữ trong hệ thống cơ sở dữ liệu hỗ trợ NoSQL và được đặt trên nền tảng máy chủ Cloud. Nền tảng này hỗ trợ đồng bộ hóa dữ liệu của người dùng kể cả khi không có kết nối mạng. Reatime Database của Firebase hỗ trợ: Android, ios, web, c++, unity, và cả xamarin.

**Crashlytics:** Hệ thống theo dõi và lưu trữ thông tin lỗi của ứng dụng đang chạy trên máy người dùng. Các thông tin lỗi này được thu thập toàn diện và ngay tức thời.

**Cloud Firestore:** Lưu trữ và đồng bộ dữ liệu giữa người dùng và thiết bị ở quy mô toàn cầu sử dụng cơ sở dữ liệu NoSQL được lưu trữ trên hạ tầng cloud. Cloud Firestore cung cấp tính năng đồng bộ hóa trực tuyến và ngoại tuyến cùng với các truy vấn dữ liệu hiệu quả. Tích hợp với các sản phẩm Firebase khác cho phép xây dựng các ứng dụng thực sự ngay cả khi kết nối Internet bị gián đoạn.

**Authentication:** Quản lý người dùng một cách đơn giản và an toàn. Firebase Auth cung cấp nhiều phương pháp xác thực, bao gồm email và mật khẩu, các nhà cung cấp bên thứ ba như Google hay Facebook, và sử dụng trực tiếp hệ thống tài khoản hiện tại của người dùng.

**Cloud Functions:** Mở rộng ứng dụng phụ trợ tùy chỉnh mà không cần quản lý và quy mô các máy chủ. Các chức năng có thể được kích hoạt bởi các sự kiện, được phát sinh ra bởi các sản phẩm Firebase, dịch vụ Google Cloud hoặc các bên thứ ba có sử dụng tính năng sử dụng dữ liệu thời gian thực (webhooks).

**Cloud Storage:** Lưu trữ và chia sẻ nội dung do người dùng tạo ra như hình ảnh, âm thanh và video với bố nhớ đối tượng mạnh mẽ, đơn giản và tiết kiệm chi phí được xây dựng cho quy mô của Google. Các Firebase SDK cho Cloud Storage thêm tính nắng bảo mật của Google để tải lên tệp cho các ứng dụng Firebase của người dùng.

**Hosting:** Đơn giản hóa lưu trữ web của người dùng với các công cụ được thực hiện cụ thể cho các ứng dụng web. Khi tải lên nội dung web, hệ thống sẽ tự động đẩy chúng đến mạng lưới cung cấp nội dung (Content Delivery Network - CDN).

**Test Lab:** Chạy thử nghiệm tự động và tùy chỉnh cho ứng dụng trên các thiết bị ảo và vật lý do Google cung cấp.

**Performance Monitoring:** Chẩn đoán các vấn đề về hiệu suất ứng dụng xảy ra trên các thiết bị của người dùng. Sử dụng dấu vết để theo dõi hiệu suất của các phần cụ thể trong ứng dụng Luôn cập nhật thời gian khởi động của ứng dụng và theo dõi các yêu cầu giao thức truyền tải (HyperText Transfer Protocol-HTTP) mà không cần viết bất kỳ mã nào.

**Các dịch vụ tăng trưởng và thu hút người dùng**

**Google Analytics**: Phân tích thuộc tính và hành vi của người sử dụng trong một bảng điều khiển đơn để đưa ra các quyết định sáng suốt về lộ trình sản phẩm. Nhận thông tin chi tiết về thời gian thực từ báo cáo hoặc xuất dữ liệu sự kiện thô tới Google BigQuery để phân tích tùy chỉnh.

**Cloud Messaging:** Gửi tin nhắn vả thông báo cho người dùng qua các nền tảng Android, IOS và web một cách miễn phí. Người dùng có thể gửi tin nhắn đến các thiết bị, nhóm thiết bị hoặc các chủ để hoặc phân đoạn người dụng cụ thể. Nhắn tin đám mây Firebase (FCM) thậm chí là các ứng dụng lớn nhất, cung cấp hàng trăm tỷ thư mỗi ngày.

**Predictions:** Firebase Predictions áp dụng deep learning máy học với dữ liệu phân tích của người dùng để tạo các nhóm người dùng năng động dựa trên hành vi dự đoán. Các nhóm người dùng này có thể được sử dụng để nhằm mục tiêu trong các sản phẩm như thông báo, cấu hình từ xa và nhiều hơn nữa.

**Dynamic Links:** Sử dụng liên kết động để cung cấp trải nghiệm người dùng tùy chỉnh cho iOS, Android và web. Người dùng có thể sử dụng chúng để hỗ trợ web di động kể thúc đẩy chuyển đổi ứng dụng gốc, người dùng chia sẻ người dùng, các chiến dịch xã hộ và tiếp thị và hơn thế nữa. Dynamic Links cung cấp cho người dùng các thuộc tính người dùng cần để hiểu rõ hơn về tăng trưởng di động của người dùng. Đây là dịch vụ thay thế của Google.

**Remote Config:** Tùy chỉnh cách ứng dụng của người dùng hiển thị cho mỗi người dùng. Thay đổi giao diện, triển khai các tính năng dần dần, chạy thử nghiệm A/B, cung cấp nội dung tùy chỉnh cho người dùng nhất định hoặc thức hiện các cập nhật khác giám sát tác động của những thay đổi của người dùng và thực hiện điều chỉnh chỉ trong vài phút.

**Invites:** Cho phép người dùng chia sẻ tất cả các khía cạnh của ứng dụng của nhà sản xuất ứng dụng, từ mã giới thiệu đến nội dung yêu thích, qua email hoặc SMS. Giải pháp out-of-the-box này hoạt động của Google Analytics for Firebase, để nhà sản xuất ứng dụng biết khi người dùng mở hoặc cài đặt một ứng dụng qua lời mời.

**App Indexing:** Thu hút lại người dùng bằng các ứng dụng đẵ cài đặt của họ với tích hợp Google tìm kiếm này. Nếu người dùng có ứng dụng của nhà sản xuất ứng dụng và họ tìm kiếm nội dung có liên quan, họ có thể khởi chạy nó trực tiếp từ kết quả. Nếu người dùng chưa có ứng dụng của nhà sản xuất, một thẻ cài đặt sẽ xuất hiện khi họ tìm kiếm các ứng dụng tương tự.

**AdMob:** Kiếm tiền bằng cách hiển thị quảng cáo hấp dẫn cho khán giả toàn cầu. AdMob có tất cả những ứng gi người dùng cần để thực hiện chiến lược kiếm tiền trên lớp bậc nhất và để tối đa hóa doanh thu nhỏ mỗi người dùng tạo ra. Nó có thể được điều chỉnh cho ứng dụng của chúng ta, và API của nó được xây dựng để tích hợp các định dạng quảng cáo phong phú một cách dễ dàng.

**AdWords:** Có được thông tin và níu kéo lại người dùng với khả năng của Google. Chúng ta có thể chạy quảng cáo trên Tìm Kiếm, hiển thị và video cũng như nhắm mục tiêu phân khúc người dùng cụ thể mà chúng ta xác định trong Google Analytics for Firebase. Cải thiện nhằm mục tiêu quảng cáo và tối ưu hóa hiệu suất chiến dịch.

**Ưu điểm và nhược điểm**

***Ưu điểm:*** Xây dựng nhanh chóng mà không tốn thời gian, nhân lực để quản lý hệ thống và cơ sở hạ tầng phía sau: Firebase cung cấp cho người dùng chức năng như phân tích, cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cỗ lỗi để dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm đem lại các trải nghiệm tốt nhất cho họ. Uy tín chất lượng đảm bảo từ Google: Firebase được google hỗ trợ và cung cấp trên nền tảng phần cứng với quy mô rộng khắp thế giới, được các tập đoàn lớn và các ứng dụng với triệu lượt sử dụng từ người dùng. Quản lý cấu hình và trải nghiệm các ứng dụng của

Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.

***Nhược điểm:*** Điểm duy nhất của Firebase chính là phần Realtime Database, mà đúng hơn chỉ là phần Database. Cơ sở dữ liệu của Realtime Database được tổ chức theo kiểu cây không phải là kiểu bảng nên những ai đang quen với SQL có thể gặp khó khăn đôi chút bước đầu.

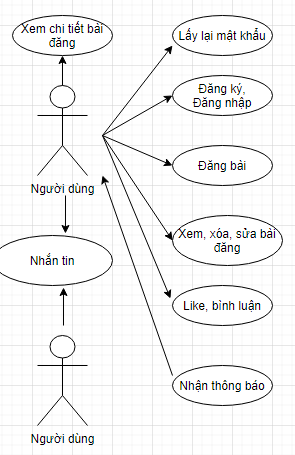
# CHƯƠNG III. PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

## 3.1 Phân tích yêu cầu hệ thống

**Yêu cầu chức năng hệ thống:** Xây dựng ứng dụng tìm đồ thất lạc cho sinh viên Trường Đại học Nha Trang là một ứng dụng có các chức năng chính: (1) Đăng ký, đăng nhập tài khoản. (2) Đăng bài viết tìm đồ hoặc trả đồ. (3) Liên hệ giữa các người dùng. (4) Tìm kiếm tổng quát hoặc nâng cao thông tin bài đăng. (5) Quản lý bài đăng cá nhân của người dùng.

**Yêu cầu phi chức năng:** (1) Ứng dụng phù hợp với các loại điện thoại, hệ điều hành Android.(2) Những chức năng thao tác dễ dàng, mô tả dễ hiểu.

### 3.1.1. Mô tả biểu đồ hoạt động chính của ứng dụng



Hình 3.1. Biểu đồ chức năng người dùng

Ứng dụng cung cấp thông tin tên các địa điểm cụ thể trong trường Đại học Nha Trang, các loại đồ dùng được cập nhật chủ yếu của sinh viên.

Khi cài đặt ứng dụng hoàn tất, người dùng đăng ký tài khoản bao gồm các thông tin: email và mật khẩu để có thể truy cập ứng dụng. Trong quá trình đăng nhập, nếu người dùng nhập sai định dạng tài khoản, ứng dụng sẽ thông báo lỗi và yêu cầu người dùng nhập lại.

Sau khi đăng nhập thành công, tại “*giao diện trang cá nhân*”, người dùng phải cập nhật các thông tin sau: tên người dùng, số điện thoại, ảnh đại diện, ảnh nền người dùng. Tại “*giao diện màn hình chính*”, ứng dụng sẽ hiển thị các bài đăng tìm hoặc trả đồ. Mỗi bài đăng bao gồm các thông tin: Tên người đăng, ngày giờ đăng chi tiết, loại đồ dùng, địa điểm, tình trạng đăng (tìm hoặc trả), hình ảnh đồ dùng (nếu có), chi tiết bài đăng ngắn ngọn, số lượng người thích bài đăng, và số lượng người bình luận bài đăng. Người dùng có thể chọn biểu tượng mở rộng của mỗi bài đăng để xem chi tiết, bài đăng chi tiết sẽ có thêm các thông tin liên hệ bao gồm: họ tên, số điện thoại, địa chỉ đồng thời hiển thị các bình luận theo danh sách, người dùng có thể thêm hoặc xoá bình luận. Người dùng có thể tìm kiếm thông tin về các bài viết trên thanh tìm kiếm hoặc tìm kiếm theo: tình trạng, loại đồ hoặc địa điểm.

Tại giao diện “*danh sách người dùng*”, màn hình giao diện sẽ hiển thị tất cả tài khoản đang sử dụng ứng dụng theo danh sách bao gồm các thông tin: họ tên, email và số điện thoại (được mã hoá 3 ký tự cuối). Người dùng có thể chọn bất kỳ tài khoản nào để: xem trang cá nhân, gọi qua số điện thoại, nhắn tin hoặc tìm kiếm tài khoản bằng cách nhập thông tin trên thanh tìm kiếm.

Tại “*giao diện danh sách tin nhắn*”, màn hình sẽ hiển thị tất cả tin nhắn của người dùng. Mỗi tin nhắn chỉ hiển thị: tin nhắn gần đây nhất, tình trạng người dùng đang online hay offline. Người dùng có thể chọn bất kỳ tin nhắn nào để xem chi tiết, mỗi khi có tin nhắn mới ứng dụng sẽ thông báo trên điện thoại và hiển thị lên vị trí đầu tiên.

Tại “*giao diện thông báo*” màn hình sẽ hiển thị thông báo về lượt thích hoặc bình luận của các người dùng khác về bài đăng của người dùng, người dùng có thể xoá các thông báo này.

Người dùng có thể thêm bài viết tìm hoặc trả đồ, tại màn hình “*giao diện thêm bài viết*”, người dùng nhập thông tin bao gồm: tiêu đề, tìm/trả đồ, loại đồ dùng, địa điểm, hình ảnh (nếu có), mô tả, họ tên, số điện thoại, địa chỉ.

Khi có bài đăng được đăng tải lên, ứng dụng sẽ thông báo cho các người dùng biết, người dùng có thể tắt hoặc bật thông báo này trong mục cài đặt ứng dụng.

### 3.1.2. Chi tiết các quy trình

#### 3.1.3.1. Quy trình đăng ký, đăng nhập tài khoản

Khi sử dụng ứng dụng, người dùng bắt buộc đăng ký tài khoản bao gồm các thông tin: tên tài khoản và mật khẩu. Tài khoản thuộc định dạng theo Email, mật khẩu phải lớn hơn sáu ký tự. Nếu tài khoản đã tồn tại trong ứng dụng, ứng dụng sẽ thông báo cho người dùng. Sau khi đăng ký thành công, người dùng phải nhập đúng tài khoản đã đăng ký để đăng nhập.

#### 3.1.3.2. Quy trình lấy lại mật khẩu

Khi người dùng quên mật khẩu của tài khoản đã đăng ký, tại giao diện đăng ký hoặc đăng nhập, người dùng chọn “*quên mật khẩu*”, sau đó nhập tên tài khoản vào mẫu hiển thị. Nếu tên tài khoản không đúng, hệ thống sẽ báo lỗi và yêu cầu nhập lại. Ngược lại, một đường dẫn sẽ được gửi tới email người dùng đã đăng ký, người dùng truy cập vào đường dẫn đó để nhập mật khẩu mới.

#### 3.1.3.3. Quy trình đăng bài viết

Tại *“giao diện màn hình chính”* người dùng chọn biểu tượng thêm bài đăng hoặc tại “*giao diện trang cá nhân*” người chọn nút thêm bài đăng. Sau đó, màn hình “*giao diện đăng bài*” sẽ xuất hiện bao gồm các thông tin: tiêu đề, tìm/trả đồ, loại đồ dùng, địa điểm, hình ảnh (nếu có), mô tả, họ tên, số điện thoại, địa chỉ. Tiêu đề và phần mô tả bắt buộc không được để trống. Sau khi người dùng nhập xong thông tin và nhấn nút đăng bài, “*giao diện màn hình chính*” sẽ cập nhật bài đăng gần đây nhất lên vị trí đầu tiên so với bài đăng của các người dung khác, đồng thời màn hình “*giao diện trang cá nhân*” của người đăng bài cũng hiển thị bài đăng. Cá nhân người dùng đăng bài có thể sửa, xoá và xem chi tiết bài đăng của mình, các người dùng khác chỉ có thể xem chi tiết bài đăng và xuất hiện nút “*chat*” để nhắn tin với người đăng. Các bài đăng sẽ hiển thị số lượng người thích và bình luận. Người dùng có thể chọn nút “*bình luận*” để bình luận hoặc xem các bình luận của người dùng khác.

#### 3.1.3.4. Quy trình cài đặt nhận thông báo

Người dùng có thể chọn biểu tượng “*cài đặt*” tại “*giao diện màn hình chính*” hoặc tại “*giao diện trang cá nhân*” để cài đặt. Sau đó, màn hình “*giao diện cài đặt thông báo*” sẽ xuất hiện. Người dùng có thể chọn bật và tắt nút “*thông báo*”. Nếu người dùng chọn bật thông báo, mỗi bài đăng được cập nhật sẽ được thông báo tới điện thoại người dùng khác.

#### 3.1.3.5. Quy trình cập nhật thông tin

Tại “*giao diện trang cá nhân*”,sau khi chọn biểu tượng “*chỉnh sửa*”, người dùng có thể chọn thay đổi các thông tin cá nhân bao gồm: họ tên, số điện thoại, email, ảnh nền, ảnh đại diện, mật khẩu.

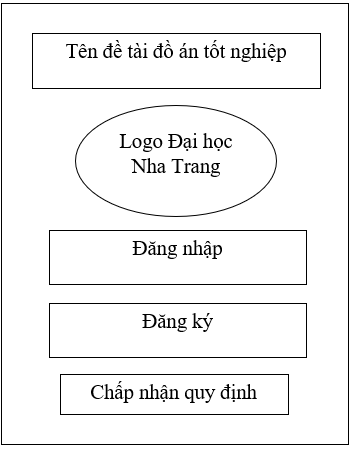
#### 3.1.3.6. Quy trình liên hệ người dùng

Quy trình liên hệ người dùng có thể thực hiện theo hai cách sau: Các bài đăng tại “*giao diện màn hình chính*” sẽ có thêm nút “*chat*” để người dùng khác liên hệ với người đăng bài. Tại “*giao diện danh sách người dùng*”, người dùng chọn tài khoản cần liên hệ, sau đó chọn hình thức liên hệ “*nhắn tin*”. Màn hình “*giao diện nhắn tin sẽ xuất hiện*”, người dùng có thể gửi tin nhắn dạng văn bản và hình ảnh. Màn hình hiển thị thông tin người dùng bao gồm: họ tên, tình trạng online hoặc thời gian hoạt động gần nhất và tình trạng người dùng đang soạn tin. Người dùng chỉ có thể xoá tinh nhắn của mình đã gửi.

## 3.2. Thiết kế giao diện và chức năng

### 3.2.1. Thiết kế giao diện màn hình khởi động

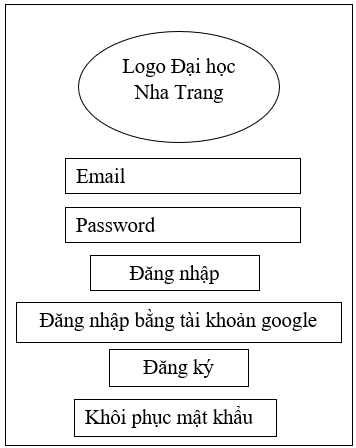
Giao diện của khởi động của ứng dụng bao gồm các thông tin: Tên đề tài đồ án tốt nghiệp, logo Trường Đại học Nha Trang, nút đăng nhập, nút đăng ký và checkbox chấp nhận quy định. Với mỗi thông tin được chọn sẽ trỏ tới giao diện tương ứng. Giao diện của ứng dụng được thiết kế như sau:



Hình 3.2. Thiết kế giao diện màn hình khởi động

### 3.2.2. Thiết kế giao diện đăng nhập

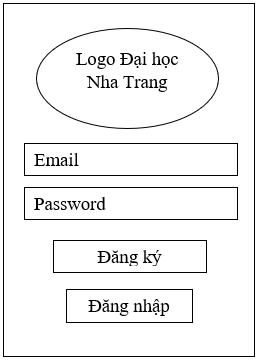
Từ giao diện màn hình khởi động, khi người dùng chọn chức năng đăng nhập, màn hình sẽ chuyển sang *giao diện đăng nhập* bao gồm các thông tin: Logo, tài khoản, mật khẩu, chức năng đăng nhập, chức năng đăng nhập bằng tài khoản Google, chức năng đăng ký và chức năng khôi phục mật khẩu. Nếu người dùng nhập sai email hoặc mật khẩu, ứng dụng sẽ báo lỗi tới người dùng và yêu cầu nhập lại, nếu người dùng nhập đúng thông tin, màn hình ứng dụng sẽ chuyển sang giao diện trang chủ. Người dùng có thể dùng tài khoản Google đang có trong điện thoại để đăng nhập ứng dụng, chọn chức năng quên mật khẩu để khôi phục mật khẩu và chọn chức năng đăng ký để đăng ký tài khoản.



Hình 3.3. Thiết kế giao diện đăng nhập

### 3.2.3. Thiết kế giao diện đăng ký

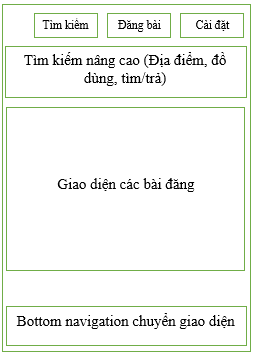
Thông tin giao diện đăng ký bao gồm: Logo, email, mật khẩu, nút đăng ký và nút đăng nhập. Nếu tài khoản đã có trong hệ thống, ứng dụng sẽ thông báo tới người dùng và yêu cầu nhập lại. Sau khi đăng ký thành công màn hình sẽ chuyển sang giao diện đăng nhập.



Hình 3.4. Thiết kế giao diện đăng ký

### 3.2.4. Thiết kế giao diện trang chủ

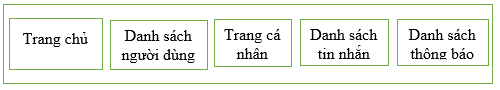
Màn hình gồm các thông tin: Tìm kiếm tổng quát, đăng bài, cài đặt, tìm kiếm nâng cao, giao diện các bài đăng của người dùng và bottom navigation chuyển màn hình. Khi người dùng cần tìm kiếm thông tin, người dùng chọn tìm kiếm tổng quát sau đó nhập thông tin, giao diện bài đăng sẽ hiển thị các bài đăng chứa từ khóa người dùng nhập. Người dùng có thể chọn nút đăng bài để tìm đồ hoặc trả đồ. Bên cạnh đó, người dùng có thể tìm kiếm theo từ khóa tại form tìm kiếm nâng cao bao gồm: địa điểm, đồ dùng, tình trạng tìm hoặc trả đồ. Tại giao diện các bài đăng của người dùng sẽ hiển thị tất cả bài đăng của các người dùng. Màn hình hiển thị bottom navigation để chuyển giao diện màn hình tương ứng.



Hình 3.5. Thiết kế giao diện trang chủ

### 3.2.5. Thiết kế giao diện bottom navigation

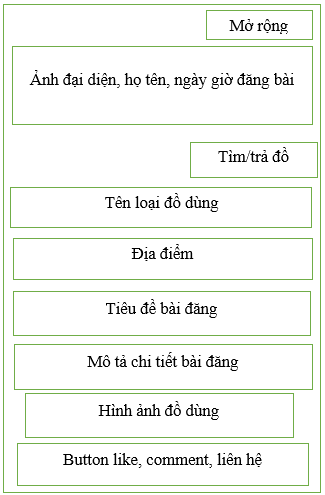
Giao diện bao gồm các chức năng khi người dùng chọn sẽ trỏ tới màn hình giao diện tương ứng bao gồm: trang chủ, danh sách người dùng, trang cá nhân, danh sách tin nhắn, danh sách thông báo. Giao diện này sẽ hiển thị bên dưới tất cả màn hình tương ứng.



Hình 3.6. Thiết kế giao diện bottom navigation

### 3.2.6. Thiết kế giao diện bài đăng

Giao diện bao gồm các thông tin: Mở rộng, ảnh đại diện, họ tên người dùng, ngày giờ người dùng đăng bài, loại tình trạng bài đăng, tên loại đồ dùng, địa điểm nhặt hoặc tìm thấy đồ dùng, tiêu đề bài đăng, mô tả chi tiết bài đăng, hình ảnh đồ dùng nếu có, button like, button comment, button liên hệ. Nếu tài khoản đang hoạt động đăng bài, phần mở rộng sẽ hiển thị các chức năng: xem chi tiết bài đăng, xóa bài đăng, sửa bài đăng. Ngược lại các tài khoản khác chỉ hiển thị chức năng xem chi tiết bài đăng. Người dùng có thể like và bình luận các bài đăng. Khi người dùng chọn chức năng bình luận, màn hình ứng dụng sẽ chuyển sang *giao diện chi tiết bài đăng* đồng thời hiển thị tất cả bình luận bên dưới bài đăng đó. Các tài khoản khác sẽ hiển thị thêm chức năng liên hệ để mở giao diện nhắn tin với người đăng.

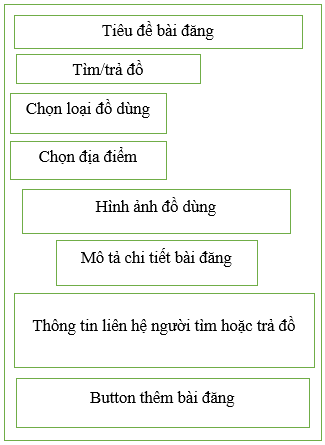


Hình 3.7. Thiết kế giao diện bài đăng

### 3.2.7. Thiết kế giao diện thêm bài đăng

Giao diện thêm bài đăng bao gồm các thông tin: Tiêu đề, chức năng chọn tìm đồ hoặc trả đồ, chọn loại đồ dùng, chọn địa điểm, hình ảnh đồ dùng, mô tả chi tiết, thông tin người cần liên hệ để tìm hoặc trả đồ, nút xác nhận đăng bài.

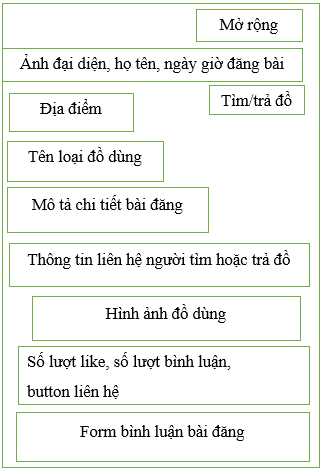
Thông tin loại đồ dùng được tổng hợp các đồ dùng sinh viên hay dùng như: thẻ sinh viên, thẻ xe, chứng minh nhân dân,… Thông tin địa điểm bao gồm tất cả địa điểm trong trường Đại học Nha Trang như: Hội trường 1, đường Thanh Niên,… Các thông tin này được hiển thị theo danh sách. Người dùng có thể tìm kiếm đồ dùng hoặc địa điểm trong danh sách đó để đăng bài.



Hình 3.8. Thiết kế giao diện thêm bài đăng

### 3.2.8. Thiết kế giao diện chi tiết bài đăng

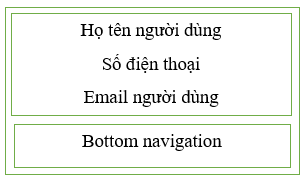
Giao diện màn hình chi tiết bài đăng bao gồm các thông tin: Mở rộng, ảnh đại diện, họ tên người dùng, ngày giờ người dùng đăng bài, loại tình trạng bài đăng, tên loại đồ dùng, địa điểm nhặt hoặc tìm thấy đồ dùng, tiêu đề bài đăng, mô tả chi tiết bài đăng, hình ảnh đồ dùng nếu có, số lượt like, số lượt bình luận, button liên hệ, form thông tin liên hệ bao gồm: họ tên, số điện thoại, địa chỉ và giao diện hiển thị tất cả lượt bình luận. Nếu tài khoản đang hoạt động đăng bài, phần mở rộng sẽ hiển thị các chức năng: xem chi tiết bài đăng, xóa bài đăng, sửa bài đăng. Ngược lại các tài khoản khác chỉ hiển thị chức năng xem chi tiết bài đăng. Người dùng có thể phóng to hình ảnh bằng cách nhấp chọn hình ảnh đó.



Hình 3.9. Thiết kế giao diện chi tiết bài đăng

### 3.2.9. Thiết kế giao diện danh sách người dùng

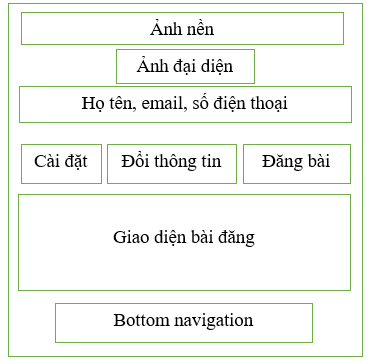
Giao diện hiển thị danh sách tất cả người dùng đã đăng ký tài khoản, mỗi tài khoản bao gồm các thông tin: Họ tên, số điện thoại và email tài khoản đó. Số điện thoại sẽ mã hóa ẩn ba ký tự cuối. Các thông tin này được lấy từ bảng Users. Người dùng có thể nhấn giữ mỗi tài khoản để hiển thị form liên hệ bao gồm: gọi qua số điện thoại hoặc nhắn tin. Màn hình hiển thị bottom navigation để chuyển giao diện màn hình tương ứng.



Hình 3.10. Thiết kế giao diện danh sách người dùng

### 3.2.10. Thiết kế giao diện trang cá nhân

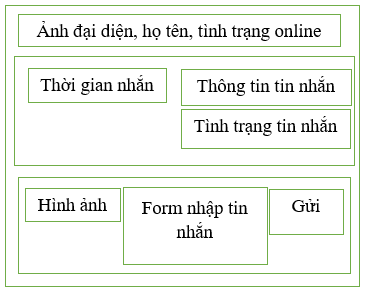
Giao diện và chức năng trang cá nhân bao gồm: Ảnh nền, ảnh đại diện, họ tên, email, số điện thoại của người dùng, button cài đặt nhận thông báo, button đổi thông tin, button đăng bài và giao diện danh sách các bài đăng của cá nhân người dùng. Khi người dụng chọn button đổi thông tin, form tùy chọn thay đổi bao gồm các chức năng đổi thông tin bao gồm: Ảnh nền, ảnh đại diện, họ tên, số điện thoại, email. Khi người dùng chọn button đăng bài, giao diện đăng bài sẽ hiển thị. Giao diện bài đăng sẽ hiển thị tất cả bài đăng của người dùng đang đăng nhập tài khoản hiện tại. Màn hình hiển thị bottom navigation để chuyển giao diện màn hình tương ứng.



Hình 3.11. Thiết kế giao diện trang cá nhân

### 3.2.11. Thiết kế giao diện nhắn tin

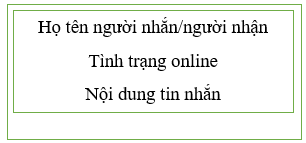
Giao diện và nhắn tin bao gồm: Ảnh đại diện, họ tên, tình trạng online của người dùng, thời gian gửi hoặc nhận tin nhắn, thông tin tin nhắn, tình trạng đã xem hoặc đã nhận của tin nhắn. Nếu tài khoản người dùng đang hoạt động thì tình trạng online sẽ hiển thị, ngược lại sẽ hiển thị thời gian hoạt động cuối cùng của người dùng đó. Nếu người dùng xem tin nhắn thì tình trạng tin nhắn sẽ hiển thị *đã xem*, ngược lại sẽ hiển thị *đã nhận*. Khi người dùng nhắn tin, người dùng có thể chọn form nhập tin nhắn để nhập thông tin, bên cạnh đó người dùng có thể gửi hình ảnh bằng cách chọn biểu tượng hình ảnh bên cạnh form nhập tin nhắn và nhấn gửi. Người dùng có thể chọn xóa tin nhắn của bản thân. Thông tin tin nhắn giữa hai người dùng được bảo mật và được lưu tại table Chats.



Hình 3.12. Thiết kế giao diện nhắn tin

### 3.2.12. Thiết kế giao diện danh sách tin nhắn

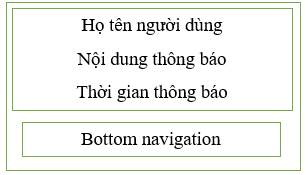
Màn hình sẽ hiển thị tất cả tin nhắn của người dùng với người dùng theo danh sách. Mỗi tin nhắn trong danh sách chỉ hiển thị các thông tin: họ tên người nhắn hoặc người nhận, tình trạng online và nội dung tin nhắn. Nếu người dùng đang hoạt động, tình trạng online sẽ hiển thị màu xanh ngược lại sẽ hiển thị màu xám. Nội dung tin nhắn sẽ hiển thị tối đa 20 kí tự.



Hình 3.13. Thiết kế giao diện danh sách tin nhắn

### 3.2.13. Thiết kế giao diện danh sách thông báo

Khi người dùng nhận thông báo về các lượt thích, bình luận, nhắn tin, ứng dụng sẽ thông báo tới điện thoại người dùng và đồng thời hiển thị tại giao diện danh sách thông báo. Các thông tin này bao gồm: họ tên người dùng, nội dung thông báo và thời gian thông báo. Các thông báo sẽ hiển thị thành danh sách, người dùng có thể xóa thông báo bằng cách nhấn giữ, màn hình sẽ hiển thị form yêu cầu người dùng xác nhận xóa. Các thông tin người được lưu tại table Notifications. Màn hình hiển thị bottom navigation để chuyển giao diện màn hình tương ứng.



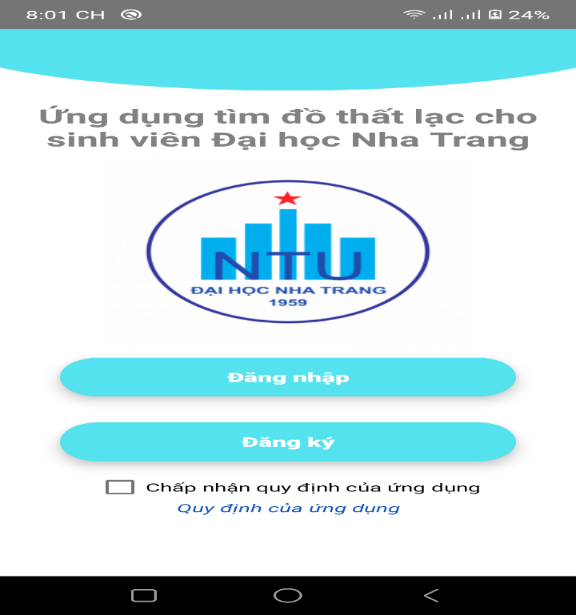
Hình 3.14. Thiết kế giao diện danh sách thông báo.

## 3.3. Kết quả thực nghiệm

Từ các bước thiết kế giao diện được trình bày ở chương 3, sau quá trình cài đặt và thiết kế. Các giao diện được hoàn thành và trình bày như sau:

### 3.3.1. Giao diện màn hình khởi động

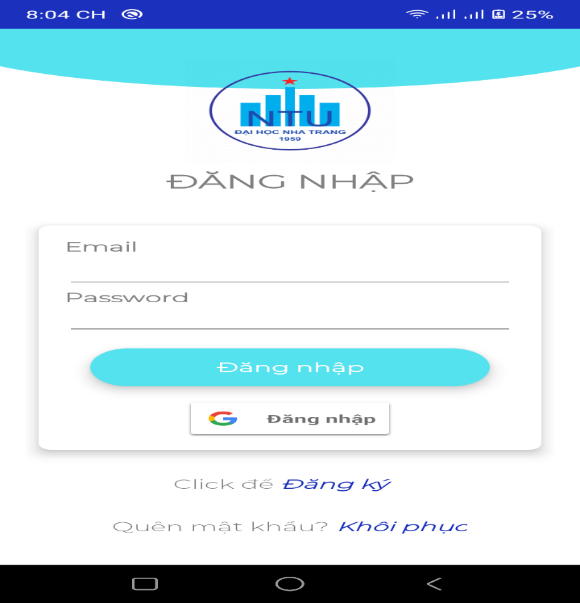
Khi người dùng khởi động ứng dụng, màn hình khởi động sẽ hiển thị. Người dùng có thể đăng nhập, đăng ký tài khoản, xem quy định của ứng dụng.



Hình 3.15. Giao diện màn hình chính khởi động

### 3.3.2. Giao diện đăng nhập

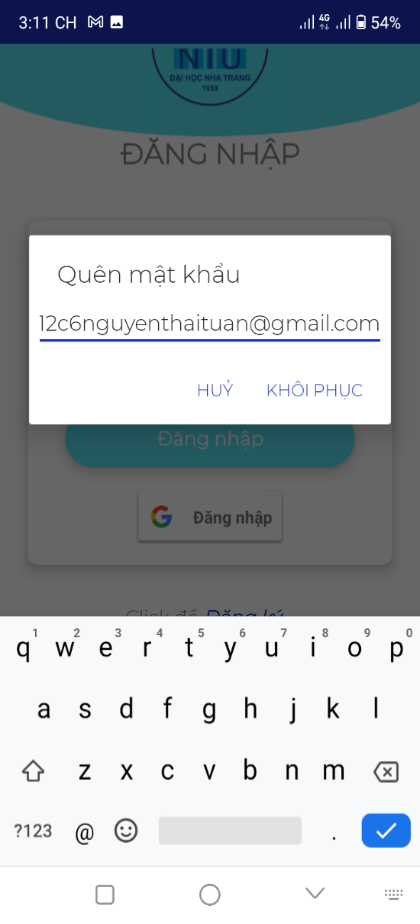
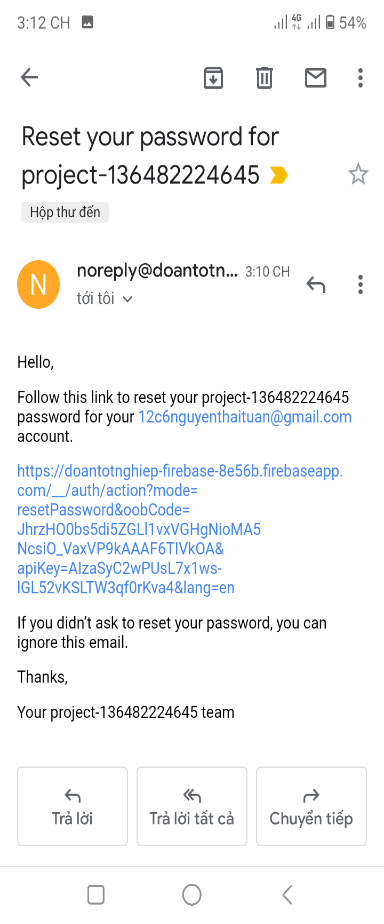
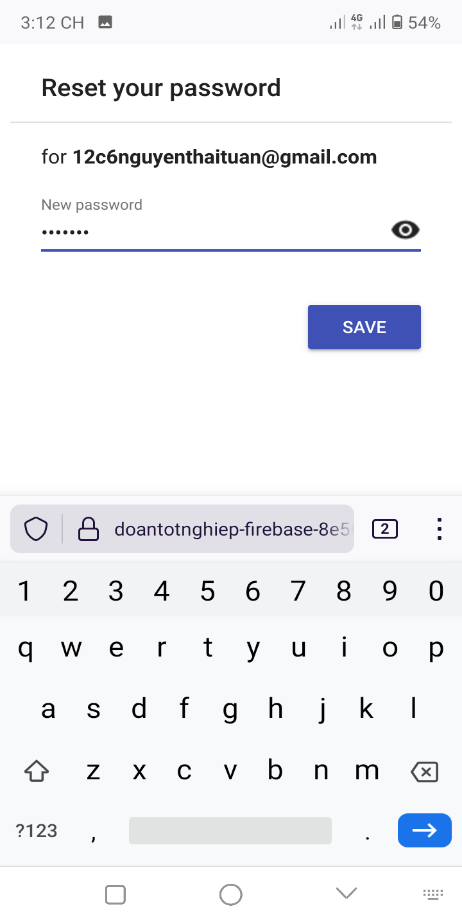
Khi người dùng chọn chức năng đăng nhập tại giao diện màn hình khởi động, màn hình sẽ chuyển sang giao diện đăng nhập. Người dùng nhập thông tin tài khoản và mật khẩu để đăng nhập hoặc đăng nhập bằng tài khoản Google trên điện thoại. Người dùng có thể đăng ký tài khoản mới hoặc khôi phục tài khoản nếu người dùng quên mật khẩu.



Hình 3.16. Giao diện đăng nhập

### 3.2.3. Giao diện khôi phục tài khoản

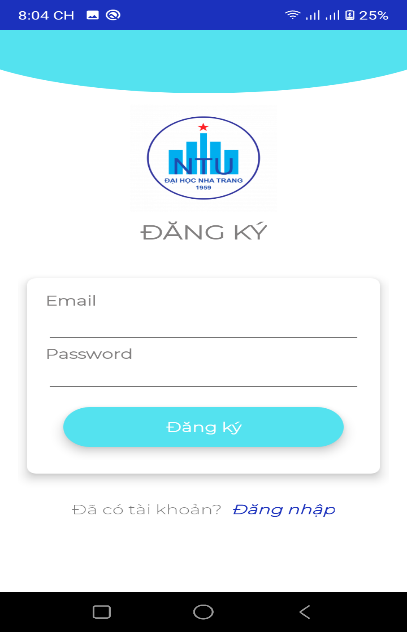
Khi người dùng chọn chức năng *khôi phục* mật khẩu từ giao diện màn hình đăng nhập, ứng dụng sẽ hiển thị form để người dùng nhập tài khoản. Nếu người dùng nhập sai thông tin hoặc định dạng hệ thống sẽ báo lỗi và yêu cầu người dùng nhập lại. Ngược lại, ứng dụng sẽ thông báo tới người dùng kiểm tra email để nhận đường link. Sau khi người dùng nhấn chọn đường link, màn hình sẽ chuyển tới giao diện nhập mật khẩu mới.

Hình 3.17.Giao diện khôi phục tài khoản

### 3.2.4. Giao diện đăng ký

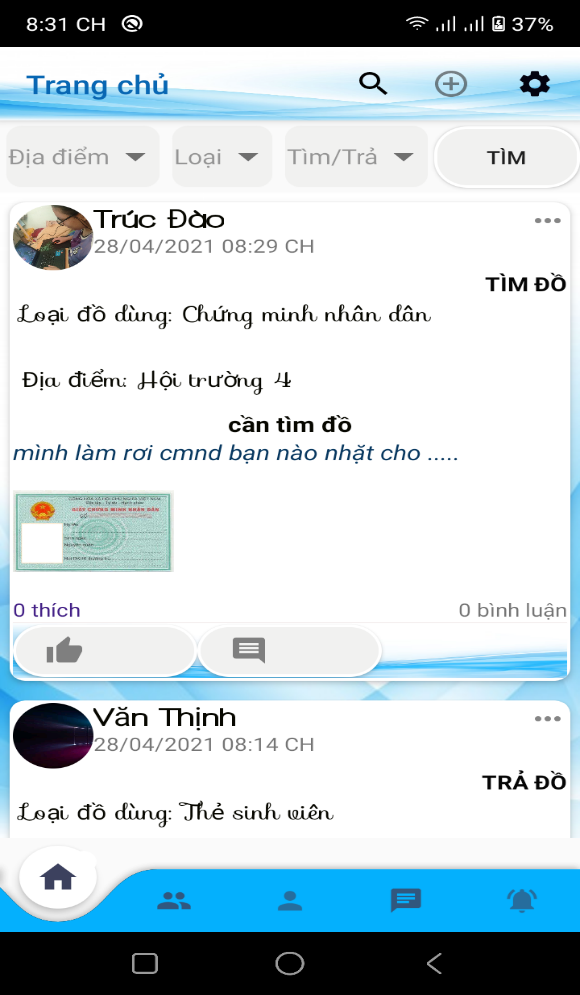
Khi người dùng chọn chức năng đăng ký từ màn hình khởi động hoặc màn hình giao diện đăng nhập, ứng dụng sẽ chuyển tới màn hình đăng ký tài khoản. Tài khoản phải thuộc định dạng Email và mật khẩu phải lớn hơn sáu kí tự. Nếu nhập sai định dạng ứng dụng sẽ báo lỗi, ngược lại sẽ thông báo thành công và chuyển sang giao diện màn hình đăng nhập.



Hình 3.18. Giao diện đăng ký

### 3.2.5. Giao diện trang chủ

Trên thanh tiêu đề của màn hình trang chủ người dùng có thể tìm kiếm tổng quát thông tin về bài đăng sau đó danh sách bài đăng sẽ hiển thị tương ứng. Người dùng có thể đăng bài bằng cách chọn biểu tượng dấu cộng, màn hình giao diện thêm bài đăng sẽ hiển thị. Người dùng có thể thực hiện tìm kiếm thông qua các từ khóa về địa điểm, loại đồ dùng, tình trạng tìm đồ hoặc trả đồ. Màn hình sẽ hiển thị tất cả bài đăng của người dùng được sắp xếp theo thời gian đăng bài gần nhất. Mỗi bài đăng sẽ thể hiện các thông tin khái quát. Để xem chi tiết bài đăng, người dùng chọn biểu tượng mở rộng để xem chi tiết bài đăng.



Hình 3.19. Giao diện trang chủ

### 3.2.6. Giao diện chi tiết bài đăng

Sau khi ứng dụng chuyển sang màn hình giao diện chi tiết bài đăng, tất cả thông tin chi tiết về bài đăng sẽ hiển thị. Người dùng có thể bình luận bên dưới bài đăng. Nếu bài đăng thuộc về tài khoản đang đăng nhập khi người dùng chọn biểu tượng mở rộng sẽ hiển thị các chức năng: sửa bài đăng và xóa bài đăng. Ngược lại sẽ không hiển thị các chức năng đó.



Hình 3.20. Giao diện chi tiết bài đăng

### 3.2.7. Giao diện danh sách người dùng

Khi người dùng chọn biểu tượng *danh sách* tại bottom navigation, ứng dụng sẽ chuyển sang màn hình giao diện danh sách người dùng. Màn hình sẽ hiển thị danh sách các tài khoản người dùng đã đăng ký. Khi người dùng chọn bất kì tài khoản nào, ứng dụng sẽ hiển thị form chức năng chọn thông tin liên hệ bao gồm: nhắn tin tại ứng dụng, gọi qua số điện thoại.



Hình 3.21. Giao diện danh sách người dùng

### 3.2.8. Giao diện trang cá nhân

Khi người dùng chọn biểu tượng *cá nhân* tại bottom navigation, ứng dụng sẽ chuyển sang màn hình giao diện trang cá nhân. Người dùng có thể xem thông tin các bài đăng của tài khoản hiện tại, thông tin cá nhân và có thể xóa hoặc sửa các thông tin đó.



Hình 3.22. Giao diện trang cá nhân

### 3.2.9. Giao diện nhắn tin

Giữa hai người dùng có thể nhắn tin với nhau, gửi thông tin hình ảnh. Đồng thời hiển thị tình trạng hoạt động của người dùng.



Hình 3.23. Giao diện nhắn tin

### 3.2.10. Giao diện danh sách tin nhắn

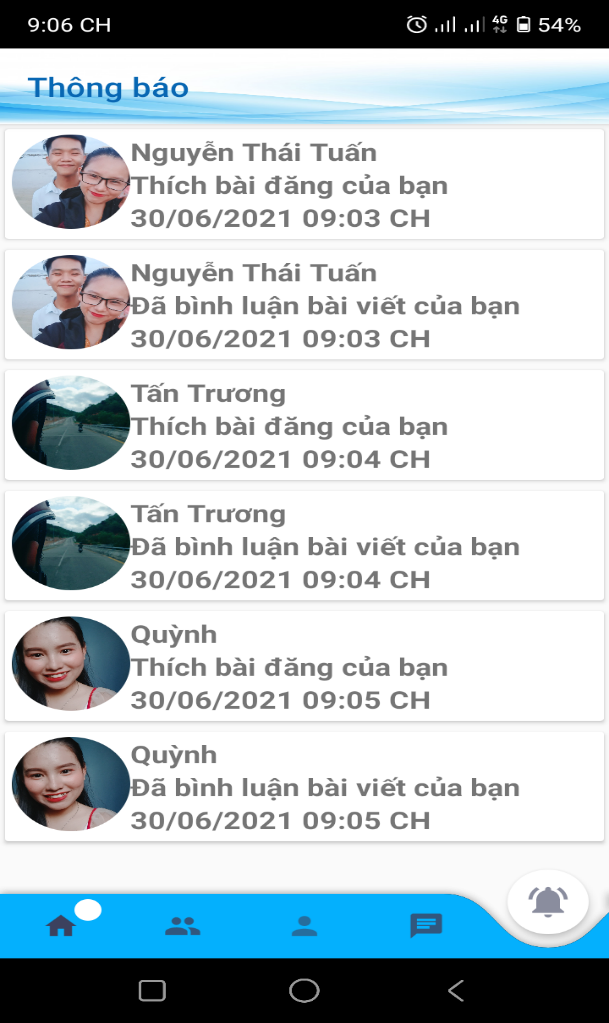
Người dùng có thể xem các tin nhắn đã nhắn với các người dùng khác. Đồng thời hiển thị tình trạng online của tài khoản đó. Nội dung tin nhắn trong giao diện này chỉ hiển thị tin nhắn gần đây nhất.



Hình 3.23. Giao diện danh sách tin nhắn

### 3.2.11. Giao diện danh sách thông báo

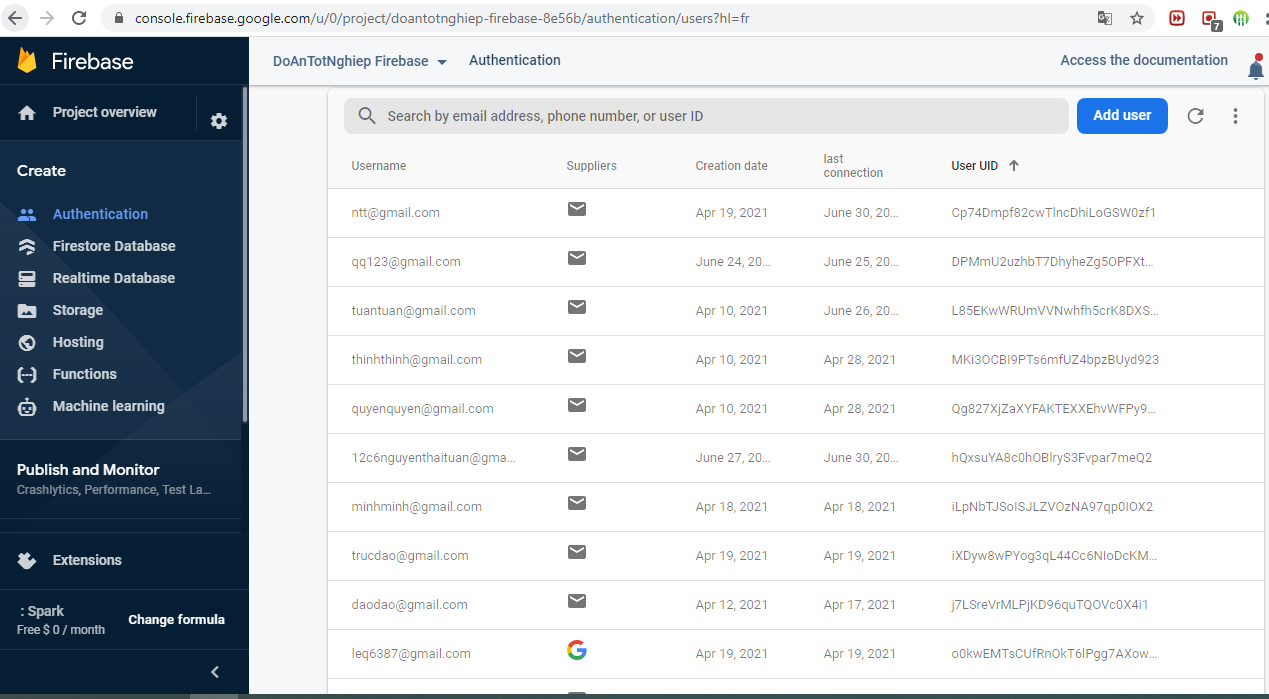
Giao diện hiển thị các thông báo về tình trạng bài đăng như: lượt thích, bình luận và hiển thị thời gian thông báo về tình trạng đó.



Hình 3.24. Giao diện thông báo

### 3.2.12. Giao diện quản lý tài khoản người dùng

Khi người quản trị truy cập trang quản lý người dùng trên Firebase, giao diện sẽ hiển thị tất cả người dùng đã đăng ký tài khoản ứng dụng. Các thông tin quản lý bao gồm: tên tài khoản người dùng, ngày tạo tài khoản, thời gian đăng nhập gần nhất và mã tài khoản.



Hình 3.25. Giao diện quản lý tài khoản người dùng

### 3.2.13. Giao diện quản lý danh sách bài đăng

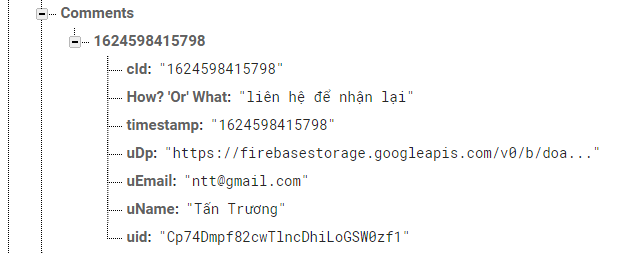
Giao diện sẽ hiển thị tất cả bài đăng có trong ứng dụng, mỗi bài đăng có một mã bài đăng được lưu trong thư mục Posts. Khi chọn vào mã bài đăng nào thì bài đăng đó sẽ hiển thị chi tiết bao gồm các thông tin: thư mục quản lý bình luận của bài đăng, số lượt bình luận, mô tả chi tiết, địa chỉ liên hệ người dùng, họ tên liên hệ người dùng, mã người dùng, địa chỉ hình ảnh lưu đồ dùng, số lượt thích, tên món đồ, địa điểm làm rơi/nhặt được, số địa thoại liên hệ, thời gian đăng bài, tình trạng bài đăng, tiêu đề tài băng, mã người đăng, họ tên người đăng, số điện thoại người đăng.



Hình 3.26. Giao diện quản lý danh sách bài đăng

### 3.2.14. Giao diện quản lý bình luận người dùng

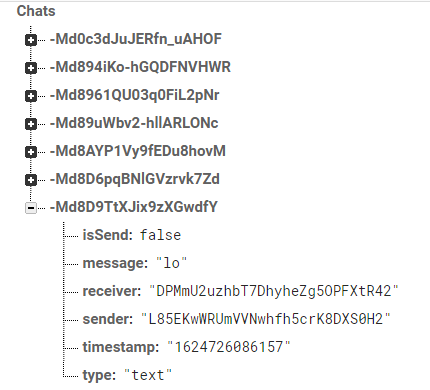
Giao diện quản lý bình luận người dùng(Comments) nằm trong thư mục quản lý bài đăng(Posts). Mỗi bài đăng sẽ bao gồm các bình luận khác nhau. Mỗi bình luận bao gồm các thông tin: Mã người dùng bình luận, thông tin bình luận, thời gian bình luận, họ tên người bình luận, tài khoản người bình luận.



Hình 3.27. Giao diện quản lý bình luận người dùng

### 3.2.15. Giao diện quản lý danh sách tin nhắn

Giao diện quản lý danh sách tin nhắn(Chats) bao gồm quản lý tất cả tin nhắn người dùng trong ứng dụng. Mỗi tin nhắn quản lý có mã tin nhắn khác nhau và bao gồm các thông tin: tình trạng xem, nội dung, mã người gửi, mã người nhận, thời gian gửi và loại tin nhắn.



Hình 3.28. Giao diện quản lý danh sách tin nhắn

### 3.2.16. Giao diện quản lý tương tác

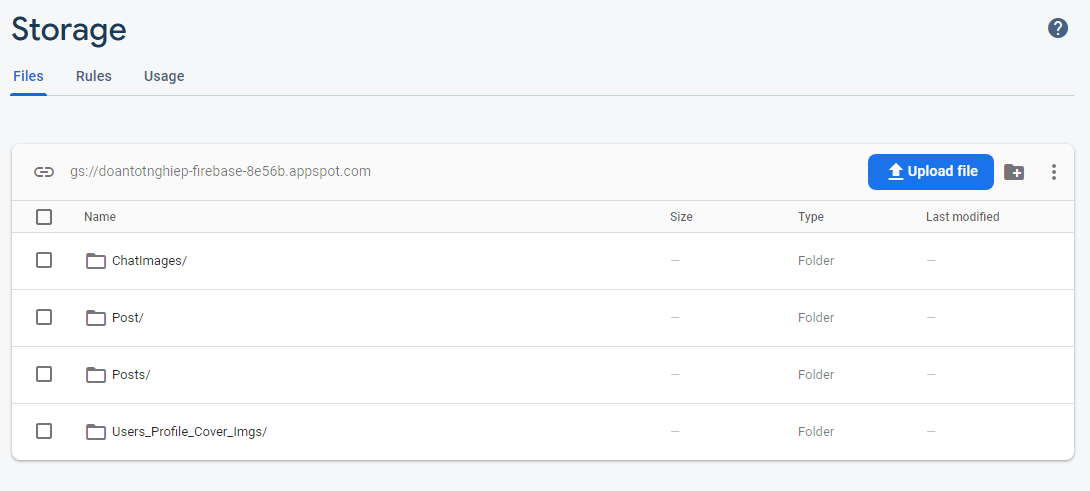
Mỗi bài đăng có các lượt thích nhằm thể hiện tương tác của người dùng với bài đăng. Mỗi quản lý tương tác hiển thị mã người dùng tương tác với bài đăng đó.



Hình 3.29. Giao diện quản lý tương tác

### 3.2.17. Giao diện quản lý lưu trữ hình ảnh

Giao diện sẽ quản lý tất cả hình ảnh người dùng đã đăng trong ứng dụng bao gồm: hình ảnh nhắn tin, hình ảnh bài đăng, hình ảnh người dùng.



Hình 3.30. Giao diện quản lý lưu trữ hình ảnh

# CHƯƠNG IV: KẾT LUẬT VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

## 4.1. Kết luận

Về phần ứng dụng đã triển khai hoàn tất về giao diện và chức năng hệ thống, xử lý dữ liệu thời gian thực cập nhật nhanh chóng và được lưu trữ trên Firebase.

**Ưu điểm:**

* Giao diện dễ nhìn, chức năng dễ sử dụng.
* Dễ dành nâng cấp ứng dụng,
* Kích thước file cài đặt nhẹ (2.88MB) có thể cài đặt trên tất cả điện thoại sử dụng hệ điều hành Android.

**Nhược điểm:**

* Không thể cài đặt trên điện thoại sử dụng hệ điều hành IOS.
* Sử dụng Firebase để lưu trữ cập nhật dữ liệu nên tốc độ đôi lúc bị chậm
* Phụ thuộc vào tốc độ mạng sử dụng.

Đồ án đã là hoàn thành tất cả những yêu cầu về đề tài và sau nhiều lần trải nghiệm cho thấy sản phẩm hoạt động ổn định, ít lỗi, khi bị lỗi, dễ dàng sửa chửa và nâng cấp. Từ đó cho thấy việc phát triển đồ án này đã hoạt động đúng với mong muốn và có thể tiến hành nâng cấp sau này.

## 4.2. Hướng phát triển đề tài

Hướng phát triển của đồ án sau này là phát triển thêm nhiều tính năng nhằm mục đích tạo ra một mạng xã hội dành riêng cho Trường Đại học Nha Trang. Nâng cấp ứng dụng để có thể cài đặt trên hệ điều hành IOS.

# CHƯƠNG V: TÀI LIỆU THAM KHẢO

[5]. https://androidcoban.com/tong-quan-ve-firebase-google-phien-banmoi.html

[6]. Elgin, Ben (August 2005). “Google Buys Android for Its Mobile Arsenal”. Bloomberg Businessweek. Bloomberg. In what could be a key move in its nascent wireless strategy, Google (GOOG) has quietly acquired startup Android, Inc.,...

[7]. Industry Leaders Announce Open Platform for Mobile Devices" Open Handset Alliance . November 5, 2012 .

[8]. “Android Overview” . Open Handset Alliance . Accessed February 15, 2012 .

[9].

[10]. Shankland, Stephen (November 12, 2007). “Google's Android parts ways with Java industry group” . CNET News .

[11]. Google Play Matches Apple's iOS With 700,000 Apps .

[12]. Google Play hits 25 billion downloads | Official Android Blog .

[13]. Raja, Haroon Q. (May 19, 2011). “Android Partitions Explained: boot, system, recovery, data, cache & misc” . Addictivetips.com . Accessed September 15, 2012 .

[14]. Chris von Eitzen (December 23, 2011). “Android drivers to be included in Linux 3.3 kernel” .

[15]. Whittaker, Zack (October 15, 2012). “Google building malware scanner for Google Play: report”

[16]. Yun Qing, Liau. "Phonemakers make Android China-friendly." ZD Net , October 15, 2012.

[17]. "Download Android Studio". Android Developers. Retrieved June 13, 2015.

[18]. “T-Mobile G1 Spec” . Infosites and comparisons . GSM Arena . Accessed September 12, 2018 .

[19]. Google buys Firebase for real-time synchronization of web and mobile apps ", Clubic.com ,October 22, 2014