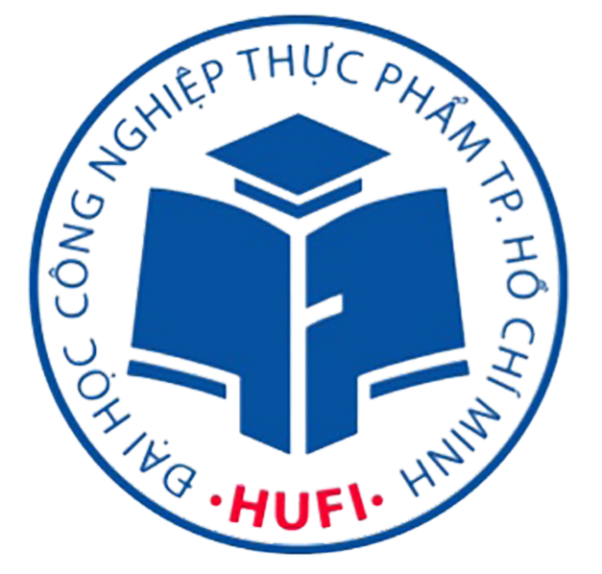
**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠--



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**

**GVHD: Th.S Vũ Đình Ái**

**Nhóm SVTH:**

**Khâu Tấn Đạt 2001160189**

**Lê Vũ Linh 2001160268**

**Đề tài: Xây dựng ứng dụng Chat realtime**

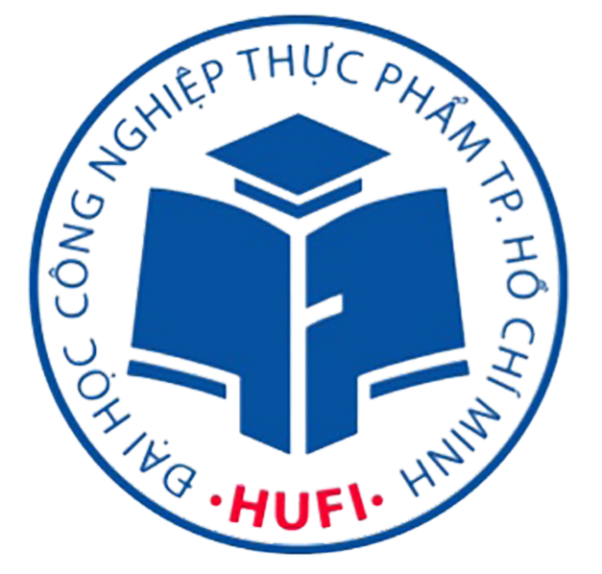
**TP.HCM 12/2019**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠--



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**

**GVHD: Th.S Vũ Đình Ái**

**Nhóm SVTH:**

**Khâu Tấn Đạt 2001160189**

**Lê Vũ Linh 2001160268**

**Đề tài: Xây dựng ứng dụng Chat Realtime**

**TP.HCM 12/2019**

# LỜI MỞ ĐẦU

Mạng điện thoại di động xuất hiện tại Việt Nam từ đầu những năm 1990 và theo thời gian số lượng các thuê bao cũng như các nhà cung cấp dịch vụ đi động tại Việt Nam ngày càng tăng. Do nhu cầu trao đổi thông tin ngày càng tăng và nhu cầu sử dụng sản phẩm công nghệ cao nhiều tính năng, cấu hình cao, chất lượng tốt, kiểu dáng mẫu mà đẹp, phong phú nên nhà cung cấp phải luôn luôn cải thiện, nâng cao những sản phẩm của mình. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kĩ thuật.

Cùng với sự phát triển của thị trường điện thoại di động là sự phát triển mạnh mẽ của xu hướng lập trình phần mềm ứng dụng cho các thiết bị di động. Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động cũng phát triển mạnh mẽ và đang thay đổi từ ngày. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web based Mobile Application đã có rất phát triển trên thị trường truyền thông di động.

Trong 11 năm , hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay, đã được nhà phát triển công nghệ rất nổi tiếng hiện nay là Google. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất.

Ngày nay, với sự phát triển nhanh chóng của xã hội, nhu cầu trao đổi thông tin cũng như trò chuyện trên ứng dụng là rất cần thiết, vì vậy chúng em đã chọn đề tài “Xây dựng ứng dụng Chat” với mục đích nghiên cứu, tìm hiểu về hệ điều hành Android và xây dựng ứng dụng Chat có tên “FreeTalk” để có thể đáp ứng được nhu cầu trò chuyện, tán gẫu, trao đổi thông tin qua lại trong và ngoài nước hiện nay một cách nhanh chóng và tiện lợi.

**MỤC LỤC**

[LỜI MỞ ĐẦU ii](#_Toc26314966)

[DANH MỤC HÌNH ẢNH iv](#_Toc26314967)

[Chương 1. CƠ SỞ LÝ THUYẾT 1](#_Toc26314968)

[1.1. Lịch sử hệ điều hành Android: 1](#_Toc26314969)

[1.2. Kiến trúc Android 2](#_Toc26314970)

[1.2.1. Linux Kernel 2](#_Toc26314971)

[1.2.2. Library và Android Runtime 4](#_Toc26314972)

[1.2.3. Framework 5](#_Toc26314973)

[1.2.4. Application 6](#_Toc26314974)

[1.3. Các thế hệ Android 7](#_Toc26314975)

[1.4. Các thành phần cơ bản của Android 17](#_Toc26314992)

[1.4.1. Các thành phần trong chương trình 17](#_Toc26314993)

[1.5. Các thư viện sử dụng 26](#_Toc26315003)

[Chương 2. TRIỂN KHAI ĐỒ ÁN 27](#_Toc26315004)

[2.1. Mô tả 27](#_Toc26315005)

[2.2. Giao diện và chức năng 27](#_Toc26315006)

[2.2.1. Giao diện chờ và bắt đầu ứng dụng 27](#_Toc26315007)

[2.2.2. Giao diện đăng ký người dùng và đăng nhập vào app 28](#_Toc26315008)

[2.2.3. Giao diện khôi phục mật khẩu 29](#_Toc26315009)

[2.2.4. Giao diện cuộc trò chuyện gần đây, cộng đồng và tìm kiếm người dùng 29](#_Toc26315010)

[2.2.5. Giao diện chat 30](#_Toc26315011)

[2.2.6. Giao diện cá nhân và thay đổi thông tin cá nhân 30](#_Toc26315012)

[2.2.7. Giao diện thông báo tin nhắn mới 31](#_Toc26315014)

[2.3. Cơ sở dữ liệu 31](#_Toc26315015)

[Chương 3. KẾT LUẬN VÀ DEMO 33](#_Toc26315016)

[3.1. Kết luận 33](#_Toc26315017)

[3.2. Hướng phát triển đề tài 33](#_Toc26315018)

[TÀI LIỆU THAM KHẢO 34](#_Toc26315019)

**DANH MỤC HÌNH ẢNH**

[Hình 1.2 Kiến trúc Android 2](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310289)

[Hình 1.2.1 Linux Kernel 3](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310290)

[Hình 1.2.2 Library và Android Runtime 4](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310291)

[Hình 1.2.3 Framework 5](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310292)

[Hình 1.3.1 Android 1.5 Cupcake 7](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310293)

[Hình 1.3.2 Android 1.6 Donut 8](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310294)

[Hình 1.3.3 Android 2.0-2.1 Eclair 9](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310295)

[Hình 1.3.4 Android 2.2 Froyo 10](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310296)

[Hình 1.3.5 Android 2.3 Gingerbread 10](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310297)

[Hình 1.3.6 Android 3.0 Honeycomb 11](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310298)

[Hình 1.3.7 Android 4.0 Ice Cream Sanwich 12](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310299)

[Hình 1.3.8 Android 4.1 Jelly Bean 12](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310300)

[Hình 1.3.9 Android 4.4 KitKat 13](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310301)

[Hình 1.3.10 Android 5.0 Lollipop 14](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310302)

[Hình 1.3.11 Android 6.0 Marshmallow 15](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310303)

[Hình 1.3.12 Android 7.0 Nougat 15](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310304)

[Hình 1.3.13 Android 8.0 Oreo 16](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310305)

[Hình 1.3.14 Android P 17](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310306)

[Hình 1.4.1a Activity 19](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310307)

[Hình 1.4.1b Intent 20](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310308)

[Hình 1.4.1c Service 22](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310309)

[Hình 2.2.1 Giao diện chờ và bắt đầu ứng dụng 27](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310310)

[Hình 2.2.2a Giao diện bắt nhập và đăng ký tài khoản người dùng 28](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310311)

[Hình 2.2.2b Giao diện đăng nhập và xác thực bằng Facebook 28](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310312)

[Hình 2.2.3 Giao diện khôi phục mật khẩu 29](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310313)

[Hình 2.2.4 Giao diện trò chuyện, cộng đồng và tìm kiếm người dùng 29](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310314)

[Hình 2.2.5 Giao diện chat 30](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310315)

[Hình 2.2.6 Giao diện cá nhân và thay đổi thông tin tài khoản 30](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310316)

[Hình 2.2.7 Giao diện thông báo tin nhắn tới 31](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310317)

[Hình 2.3.1 Firebase Auth 31](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310318)

[Hình 2.3.3 Firebase Storage 32](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310319)

[Hình 2.3.2 Firebase Database 32](file:///C:\Users\Admin\Desktop\LTDD_FreeChat.docx#_Toc26310320)

# CƠ SỞ LÝ THUYẾT

## Lịch sử hệ điều hành Android:

Android là 1 nền tảng phần mềm dựa trên mã nguồn mở Linux OS (Kernel 2.6) cho máy di động, máy tính bảng và những phần mềm trung gian (middleware). Nó không đơn thuần là 1 hệ điều hành, 1 công cụ lập trình hay 1 phần mềm trung gian mà nó gồm tất cả.

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên lõi Linux do công ty Android Inc. (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform.

Và sau tiếp, vào cuối năm 2007, thuộc về Liên minh thiết bị cầm tay mã nguồn mở (Open Handset Alliance). Mục tiêu của liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android được thiết kế để phục vụ nhu cầu của các nhà sản xuất thiết, các nhà khai thác và các lập trình viên thiết bị cầm tay.

Phiên bản SDK lần đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK release Candidate 1.0.Trong tháng 10 năm 2008, Google được cấp giấy phép mã nguồn mở cho Android Platform.

Khi Android được phát hành thì một trong số các mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ những ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao diện người dùng.

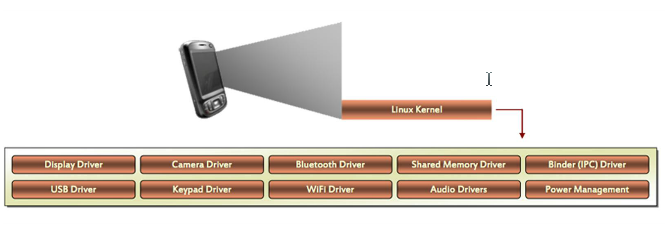
Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên vản vá lỗi 1.1 của hệ điều hành này.Ở cả hai phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý. Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng Tư năm 2009, cùng với một số tính năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng, và các live folder.

## Kiến trúc Android

Mô hình tổng quát các thành phần của hệ điều hành Android:

Hình 1.2 Kiến trúc Android

### Linux Kernel

Hệ điều hành android được phát trển dựa trên hạt nhân linux, cụ thể là hạt nhân linux phiên bản 2.6, điều đó được thể hiện ở lớp dưới cùng này. Tất cả mọi hoạt động của điện thoại muốn thi hành được thì đều được thực hiện ở mức cấp thấp ở lớp này bao gồm quản lý bộ nhớ (memory management), giao tiếp với phần cứng (driver model), thực hiện bảo mật (security), quản lý tiến trình (process). 

Hình 2.2.1 Linux Kernel

Tuy được phát triển dựa vào nhân linux nhưng thực ra nhân linux đã được nâng cấp và sửa đổi rất nhiều để phù hợp với tính chất của những thiết bị cầm tay như hạn chế về bộ vi xử lý, dung lượng bộ nhớ, kích thước màn hình, nhu cần kết nối mạng không dây...

Các thành phần của nhân Linux:

* Display Driver : Điều khiển việc hiển thị lên màn hình cũng như thu nhận những điều khiển của người dùng lên màn hình (di chuyển, cảm ứng...).
* Camera Driver : Điều kiển 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 tính hiệu dạng audio thành tín hiệu số và ngược lại.
* Power Madagement : Giám sát việc tiêu thụ điện năng.
* M-System Driver : Quản lý việc đọc ghi... lên các thiết bị nhớ như thẻ SD, flash.
* 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.

### Library và Android Runtime

Hình 3.2.2 Library và Android Runtime

#### Library

Thành phần này có nhiều thư viện được viết bằng C/C++ để các phần mềm có thể sử dụng, các thư viện đó được tập hợp thành một số nhóm như:

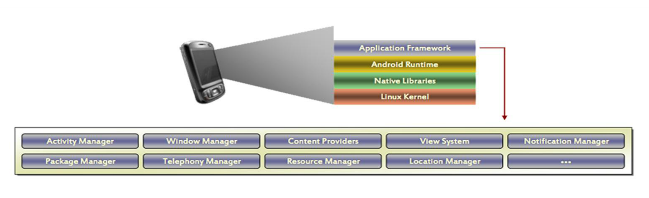
* Thư viện hệ thống (System C library)(Libc) : thư viện dựa trên chuẩn C, được sử dụng chỉ bởi hệ điều hành.
* Thư viện Media (Media Libraries) : Có nhiều codec để hỗ trợ việc phát và ghi các loại định dạng âm thanh, hình ảnh, video thông dụng.
* Thư viện web (LibWebCore) : Đây là thành phần để xem nội dung trên web, được sử dụng để xây dựng phần mềm duyệt web (Android Browse) 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…
* Surface Manage: quản lý hiển thị 2D và 3D.
* SGL công nghệ 2D.
* SQLite : quản lý database của ứng dụng.
* Free Type : dùng cho hiển thị ảnh, font. SSL(Secure Socket Layer): Tầng thiết lập giao dịch an toàn trên 2 ứng dụng trên 1 cổng (socket).
* Open GL/EL: hỗ trợ thư viện tùy chọn 3D tĩnh và động.

#### Android Runtime

Phần này chứa các thư viện mà một 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. Thứ nhất là các thư viện lõi (Core Library) , chứa các lớp như JAVA IO, Collections, File Access. Thứ hai là một 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(Java Runtime Environment) của Sun (nay là Oracle) (JVM) mà là chạy bằng máy ảo Dalvik (máy ảo dalvik đc đặt tên sau khi 1 nhà phát triển phần mềm của google tạo ra nó sau khi đi thăm 1 ngôi làng ở nơi tổ tiên ông sinh sống) do Google phát triển. máy ảo davik thực thi cái tập tin davik(dex), khác với java là chuyển sang dạng bytecode. Định dạng này được tối ưu hóa cho bộ nhớ tối thiểu.

### Framework

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa.

Hình 4.2.3 Framework

Hình 1.2.3 Frameword

Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần. Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

* View Manage:Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như: gridview, tableview, linearlayout…
* 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 việc liên lạc như gọi điện thoại.
* Location Manager : Cho phép xác định vị trí của điện thoại 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.
* Notication Manager : Quản lý việc hiển thị các thông báo (như báo có tin nhắn, có e-mail mới).
* Resource Manager : Quản lý tài nguyên tĩnh của các ứng dụng bao gồm các file hình ảnh, âm thanh, layout, string. (Những thành phần không được viết bởi ngôn ngữ lập trình).
* Content Provider: cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.

### Application

Đây là lớp ứng dụng giao tiếp với người dùng, bao gồm các ứng dụng như: Các ứng dụng cơ bản, được cài đặt đi liền với hệ điều hành là gọi điện(phone), quản lý danh bạ(Contacts), duyệt web (Browser),nhắn tin(SMS),lịch làm việc (Calendar), đọc e-mail (Email-Client), bản đồ (Map), quay phim chụp ảnh (camera)... Các ứng dụng được cài thêm như các phần mềm chứng khoán (Stock), các trò chơi (Game), từ điển, các widget các ứng dụng của bên thứ 3…

Các ứng dụng viết sẽ có những đặc tính sau:

* Viết bằng Java, phần mở rộng là .apk.
* Khi mỗi ứng dụng được chạy, nó có một phiên bản Virtual Machine được dựng lên để phục vụ cho nó. Nó có thể là một Active Program: Chương trình có giao diện với người sử dụng hoặc là một background: chương trình chạy nền hay là dịch vụ.
* 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 mỗi ứng dụng thì có duy nhất một thực thể (instance) được phép chạy mà thôi. Điều đó có tác dụng hạn chế sự lạm dụng tài nguyên, giúp hệ thống hoạt động tốt hơn. Các ứng dụng được phân quyền và tự hệ thống sẽ kill nếu mức ưu tiên thấp.
* Các ứng dụng được gán số ID của người sử dụng nhằn 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 dung quá 5~10% công suất CPU, điều đó nhằn để tránh độc quyền trong việc sử dụng CPU.

## Các thế hệ Android

* Android 1.5 Cupcake

Hình 1..1 Android 1.5 Cupcake

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 2](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/1_ukry.jpg)** |

Tên mã công khai chính thức đầu tiên của Android đã không xuất hiện cho đến phiên bản 1.5 Cupcake vào tháng 4/2009. Nó đã bổ sung khá nhiều tính năng và cải tiến mới so với hai phiên bản công khai đầu tiên, như khả năng tải video lên YouTube, cách hiển thị màn hình của điện thoại tự động xoay sang đúng vị trí và hỗ trợ bàn phím của bên thứ ba.

Một số điện thoại được phát hành với Cupcake bao gồm điện thoại Samsung Galaxy đầu tiên, cùng với HTC Hero.

* Android 1.6 Donut

Hình 1..2 Android 1.6 Donut

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 3](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/2_qkxz.jpg)** |

Google đã nhanh chóng tung ra Android 1.6 Donut trong tháng 9/2009. Một số tính năng mới hỗ trợ cho các nhà mạng sử dụng mạng CDMA. Điều này giúp đưa điện thoại Android được bán trên toàn thế giới.

Các tính năng khác bao gồm việc giới thiệu Quick Search Box và nhanh chóng chuyển đổi giữa Máy ảnh, Máy quay và Thư viện để sắp xếp trải nghiệm chụp media. Donut cũng đã giới thiệu tiện ích Power Control để quản lý Wifi, Bluetooth, GPS,…

Một trong những chiếc điện thoại được bán với Donut được cài đặt là Dell Streak xấu số, màn hình 5 inch rất lớn và đã được mô tả tại thời điểm đó là “điện thoại thông minh/máy tính bảng”. Những ngày này, màn hình 5 inch được xem là có kích thước trung bình cho điện thoại thông minh.

* Android 2.0-2.1 Eclair

Hình 1..3 Android 2.0-2.1 Eclair

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 4](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/3_dvtc.jpg)** |

Vào tháng 10/2009, khoảng một năm sau khi tung ra Android 1.0, Google đã phát hành phiên bản 2.0 của hệ điều hành với tên mã chính thức là Eclair. Phiên bản này là phiên bản đầu tiên bổ sung hỗ trợ Text-to-Speech và cũng giới thiệu các hình nền động, hỗ trợ nhiều tài khoản và điều hướng của Google Maps trong số nhiều tính năng và cải tiến mới khác.

Motorola Droid là điện thoại đầu tiên có Android 2.0 cài đặt sẵn. Đây cũng là là điện thoại Android đầu tiên được Verzion Wireless bán. Mặc dù Google vẫn có thể sử dụng Android làm tên cho hệ điều hành của mình, thuật ngữ “Droid” đã được Lucasfilm đăng ký nhãn hiệu. Motorola đã phải xin phép, và trả một khoản tiền cho Lucasfilm, để sử dụng Droid làm tên cho điện thoại của mình. Motorola tiếp tục sử dụng thương hiệu Droid cho nhiều điện thoại của hãng vào cuối năm 2016.

* Android 2.2 Froyo

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 5](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/4_wyib.jpg)**  Hình 1..4 Android 2.2 Froyo |

Ra mắt vào tháng 5/2010, Android 2.2 Froyo (yogurt đông lạnh) đã chính thức ra mắt. Điện thoại thông minh có cài đặt Froyo có thể tận dụng một số tính năng mới, bao gồm chức năng điểm phát sóng di động Wi-Fi, đẩy thông báo qua dịch vụ C2DM của Android Cloud, hỗ trợ flash và nhiều hơn thế nữa.

Điện thoại thông minh đầu tiên mang thương hiệu Nexus của Google, Nexus One, được tung ra với Android 2.1 cài sẵn vào đầu năm 2010 và nhanh chóng nhận được thông tin cập nhật Froyo vào cuối năm đó. Điều này đánh dấu một cách tiếp cận mới cho Google.

* [](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/5_vkrw.jpg)Android 2.3 Gingerbread

|  |
| --- |
| Hình 1..5 Android 2.3 Gingerbread |

Android 2.3 Gingerbread, được tung ra vào tháng 9/2010, hiện là phiên bản cũ nhất của hệ điều hành mà Google vẫn đang liệt kê trong trang cập nhật phiên bản nền tảng hàng tháng. Tính đến ngày 13/9/2017, Google chỉ ra rằng chỉ 0,6 phần trăm của tất cả các thiết bị Android hiện đang chạy một số phiên bản của Gingerbread.

Hệ điều hành nhận được giao diện người dùng dưới tên Gingerbread và thêm hỗ trợ chức năng giao tiếp NFC. Điện thoại đầu tiên để thêm cả phần cứng Gingerbread và NFC là Nexus S, được đồng phát triển bởi Google và Samsung. Gingerbread cũng đặt nền móng cho selfie, bằng cách hỗ trợ nhiều camera và hỗ trợ trò chuyện video trong Google Talk.

* Android 3.0 Honeycomb

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 7](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/6_sfqz.jpg)**  Hình 1..6 Android 3.0 Honeycomb |

Honeycomb được Google phát hành chỉ để cài đặt trên máy tính bảng và các thiết bị di động khác có màn hình lớn hơn so với điện thoại thông minh hiện tại. Honeycomb lần đầu tiên được giới thiệu vào tháng 2/2011, cùng với máy tính bảng Motolola Xoom đầu tiên và bao gồm các tính năng như giao diện người dùng được thiết kế lại dành riêng cho màn hình lớn cùng với thanh thông báo được đặt ở phía dưới màn hình của máy tính bảng.

* Android 4.0 Ice Cream Sandwich

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 8](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/7_lrog.jpg)**  Hình 1..7 Android 4.0 Ice Cream Sanwich |

Được phát hành vào tháng 10/2011, phiên bản Ice Cream Sandwich của Android đã mang một số tính năng mới cho người dùng. Nó kết hợp nhiều tính năng của phiên bản Honeycomb dành cho máy tính bảng với Gingerbread theo định hướng smartphone. Nó cũng bao gồm một “khay yêu thích” trên màn hình chủ, cùng với sự hỗ trợ đầu tiên để mở khóa điện thoại bằng cách chụp ảnh khuôn mặt của chủ sở hữu. Loại hỗ trợ đăng nhập sinh trắc học đã phát triển và cải thiện đáng kể kể từ đó.

* Android 4.1 Jelly Bean

Hình 1..8 Android 4.1 Jelly Bean

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 9](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/8_zmoj.jpg)** |

Kỷ nguyên Jelly Bean của Android bắt đầu vào tháng 6/2012 với việc phát hành Android 4.1. Một số tính năng mới bao gồm thông báo mới cho thấy nhiều nút nội dung cùng với sự hỗ trợ đầy đủ cho phiên bản trình duyệt web Chrome của Android. Google Now cũng đã xuất hiện trên Search và “Project Butter” được giới thiệu để tăng tốc độ hoạt hình và cải thiện khả năng phản ứng cảm ứng của Android. Màn hình, chụp ảnh HDR và Miracast cũng được hỗ trợ.

Lúc đó, khoảng 6,9 phần trăm của tất cả các sản phẩm Android sử dụng Jelly Bean.

* Android 4.4 KitKat

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 10](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/9_wzms.jpg)**  Hình 1..9 Android 4.4 KitKat |

Tên của Android 4.4 là phiên bản duy nhất của hệ điều hành thực sự sử dụng tên nhãn hiệu cho một miếng kẹo. KitKat không có nhiều các tính năng mới, nhưng nó đã có một điều mà thực sự đã giúp mở rộng thị trường Android tổng thể. Nó đã được tối ưu hóa để chạy trên điện thoại thông minh có ít nhất 512 MB RAM. Điều này cho phép các nhà sản xuất điện thoại để có được phiên bản mới nhất của Android và cài đặt nó trên điện thoại rẻ hơn nhiều.

Điện thoại thông minh Nexus 5 của Google là điện thoại thông minh đầu tiên sử dụng Android 4.4. Mặc dù KitKat đã được phát hành cách đây gần bốn năm, vẫn còn rất nhiều thiết bị vẫn đang sử dụng nó. Các trang cập nhật trang cập nhật nền tảng hiện tại của Google hiển thị rằng 15,1 phần trăm của tất cả các thiết bị Android đang chạy một số phiên bản Android 4.4 KitKat.

* Android 5.0 Lollipop

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 11](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/10_axdh.jpg)**  Hình 1..10 Android 5.0 Lollipop |

Được ra mắt lần đầu vào mùa thu năm 2014, Android 5.0 Lollipop là một bước thay đổi lớn trong giao diện tổng thể của hệ điều hành. Đây là phiên bản đầu tiên của hệ điều hành sử dụng ngôn ngữ Material Design của Google, sử dụng các hiệu ứng ánh sáng và bóng tối tự do. UI cũng có một số thay đổi khác cho Lollipop, bao gồm một thanh điều hướng mới, thông báo phong phú cho màn hình khóa và nhiều hơn nữa.

Bản cập nhật Android 5.1 đã có một số thay đổi. Các tính năng này bao gồm hỗ trợ chính thức cho các cuộc gọi thoại đôi, cuộc gọi thoại HD và bảo vệ thiết bị ngay cả sau khi khôi phục cài đặt gốc.

Điện thoại thông minh Nexus 6 của Google cùng với máy tính bảng Nexus 9 là thiết bị đầu tiên được cài đặt sẵn Lollipop. Android 5.0 Lollipop đang được cài đặt và sử dụng bởi khoảng 29% trong số tất cả các thiết bị Android đang hoạt động.Thực tế thú vị là Google đã sử dụng tên mã là “Lemon Meringue Pie” trong khi phát triển Android 5.0 trước khi quyết định sử dụng Kẹo Lollipop làm tên công khai của hệ điều hành.

* Android 6.0 Marshmallow

Hình 1..11 Android 6.0 Marshmallow

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 12](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/11_czqw.jpg)** |

Android 6.0 Marshmallow được phát hành vào mùa thu năm 2015. Nó bao gồm các tính năng như một ngăn kéo ứng dụng di chuyển theo chiều dọc, Google Now trên Tap, hỗ trợ mở khóa sinh trắc học, hỗ trợ USB Type-C, Android Pay, và nhiều hơn nữa.

* Android 7.0 Nougat[](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/12_hsfv.jpg)

Hình 1..12 Android 7.0 Nougat

Phiên bản 7.0 của hệ điều hành di động của Google được ra mắt vào mùa thu năm 2016. Trước khi Nougat được tiết lộ “Android N” đã được Google gọi nội bộ là “New York Cheesecake”. Một số tính năng mới của Nougat bao gồm các chức năng đa tác vụ tốt hơn cho số lượng ngày càng tăng của điện thoại thông minh có màn hình lớn, chẳng hạn như chế độ chia màn hình cùng với việc chuyển đổi nhanh giữa các ứng dụng.

Google cũng đã sử dụng phiên bản này để đẩy mạnh thị trường smartphone cao cấp. Các điện thoại thông minh có thương hiệu riêng của công ty, Pixel và Pixel XL, cùng với LG V20, là những người đầu tiên được phát hành với Nougat cài sẵn.

* Android 8.0 Oreo

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 14](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/13_lqdt.jpg)**  Hình 1..13 Android 8.0 Oreo |

Vào tháng 3/2017, Google chính thức công bố và phát hành bản xem trước dành cho nhà phát triển đầu tiên dành cho Android O, còn được gọi là Android 8.0.Đây là lần thứ hai Google chọn một tên thương hiệu cho Android (Oreo thuộc sở hữu của Nabisco). Android Oreo bao gồm nhiều thay đổi trực quan trong menu Cài đặt, cùng với hỗ trợ chế độ ảnh-trong-ảnh, các kênh thông báo, API tự động điền mới để quản lý mật khẩu và điền dữ liệu tốt hơn.

* Android P

|  |
| --- |
| **[Khám phá lịch sử của hệ điều hành Android: nguồn gốc, biểu tượng và tên gọi toàn kẹo bánh - ảnh 15](https://image.viettimes.vn/w800/Uploaded/2019/aohuooh/2018_07_06/14_rypo.jpg)**  Hình 1..14 Android P |

Google đã giới thiệu bản xem trước dành cho nhà phát triển đầu tiên của bản cập nhật Android chính tiếp theo vào ngày 7/3. Hiện tại, nó chỉ được gọi là "Android P" và phiên bản xem trước đầu tiên chỉ hoạt động trên điện thoại thông minh Pixel của Google. Một số tính năng trong bản phát hành xem trước đầu tiên bao gồm hỗ trợ định dạng video HDR và HEIF, điều hướng trong Google Maps, cải tiến về bảo mật và hỗ trợ chính thức cho máy ảnh và cảm biến “tai thỏ”.

## Các thành phần cơ bản của Android

1. Các thành phần trong chương trình

* Activity

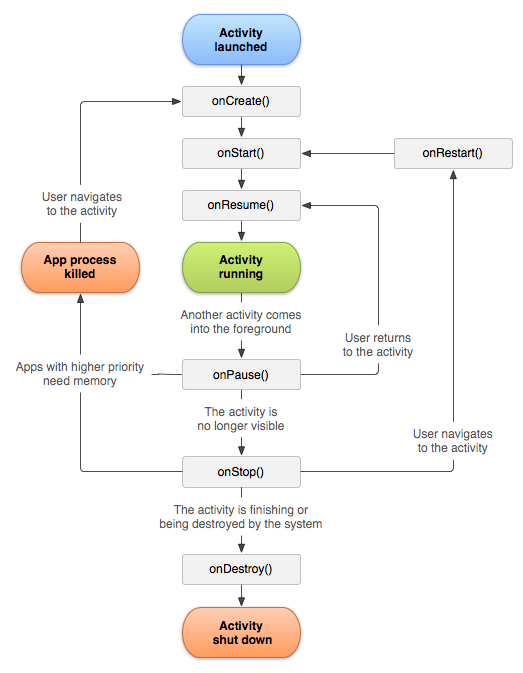
Một activity thể hiện một giao diện đồ họa người dùng. Ví dụ một activity có thể biểu diễn một danh sách các menu item để người dùng có thể chọn và có thể hiển thị ảnh cùng với tiêu đề. Một ứng dụng gửi tin nhắn văn bản có thể có một hoạt động là hiển thị một danh sách các liên hệ để gửi tin nhắn tới, hoạt động thứ hai là viết tin nhắn tới liên hệ được chọn, các hoạt động khác nữa là xem lại tin nhắn cũ hay thay đổi cài đặt. Mặc dù chúng làm việc cùng nhau để tạo thành một giao diện người dùng, mỗi activity độc lập với những cái khác. Mỗi activity là một lớp con của lớp cơ sở Activity.

Một ứng dụng có thể gồm chỉ một activity hay nhiều activity. Thông thường, một trong số các activity được đánh dấu như là activity đầu tiên phải được trình diễn tới người dùng khi ứng dụng được khởi động. Chuyển từ một activity sang activity khác được hoàn thành bằng cách cho activity hiện thời khởi động activity kế tiếp. Mỗi activity được đưa ra một cửa sổ mặc định để vẽ vào. Thông thường, cửa sổ sẽ lấp đầy màn hình, nhưng nó có thể nhỏ hơn màn hình và nằm trên các cửa sổ khác. Một activity có thể sử dụng cửa sổ bổ sung - ví dụ như một pop-up dialog gọi cho một đáp ứng người dùng ở giữa của activity, hay một cửa sổ biểu diễn những cho người dùng với thông tin quan trọng khi họ chọn một item cụ thể trên màn hình.

Nội dung trực quan của cửa sổ được cung cấp bởi một cây phân cấp các đối tượng view dẫn xuất từ lớp View. Mỗi view điều khiển một khoảng hình chữ nhật cụ thể bên trong cửa sổ. View cha chứa và tổ chức bố cục các view con. Các view lá vẽ trong hình chữ nhật mà chúng điều khiển và đáp ứng lại các hành động người dùng trực tiếp ở khoảng trống này. Do đó, các view là nơi mà các tương tác của activity với người dùng diễn ra. Ví dụ một view có thể hiển thị một hình ảnh nhỏ và khởi tạo một hoạt động khi người dùng nhấn vào hình ảnh đó. Android có một số view đã xây dựng sẵn mà bạn có thể sử dụng – gồm có các buttons, text fields, scroll bars, menu items, check boxes…

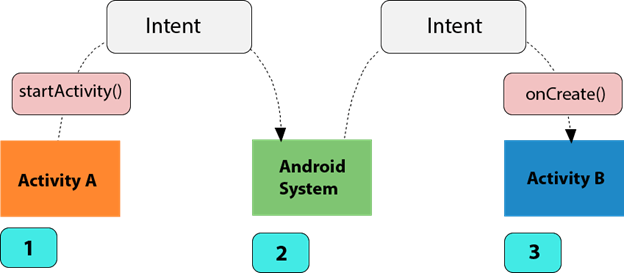
Một cây phân cấp view được đặt trong một cửa sổ của activity bằng phương thức Activity.setContentView(). Content view là đối tượng View ở gốc của cây phân cấp. Class cơ sở Activity định nghĩa một loạt các sự kiện mà điều chỉnh vòng đời của một hoạt động. Class Activity định nghĩa các sự kiện sau đây:

* 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 1..1a Activity

* Intent

Là một cấu trúc dữ liệu mô tả cách thức, đối tượng thực hiện của một Activity. Là cầu nối giữa các Activity: ứng dụng Android thường bao gồm nhiều Activity, mỗi Activity hoạt động độc lập với nhau và thực hiện những công việc khác nhau. Intent chính là người đưa thư, giúp các Activity có thể triệu gọi cũng như truyền các dữ liệu cần thiết tới một Activity khác. Điều này cũng giống như việc di chuyển qua lại giữa các Forms trong lập trình Windows Form.

Hình 1..1b Intent

**Intent có 2 dạng chính sau:**

Explicit Intents: Xác định rỏ một component (Một thể hiện của một class java dùng để thực thi các hành động được đặc tả trong Intent) (thông qua phương thức setComponent(ComponentName) hoặc setClass(Context, Class)) cung cấp lớp sẽ thực thi các hành động được đặc tả trong Intent. Thông thường thì những Intent này không chứa bất kỳ thông tin nào khác (như category, type) mà đơn giản chỉ là cách để ứng dụng mở các Activity khác bên trong một Activity.

Implicit Intents: Không chỉ định một component nào cả, thay vào đó, chúng sẽ chứa đủ thông tin để hệ thống có thể xác định component có sẵn nào là tốt nhất để thực thi hiệu quả cho Intent đó.

Khi sử dụng Implicit intents, do tính chất chuyên quyền của loại Intent này,ta cần phải biết phải làm gì với nó. Công việc này được đảm nhiệm bởi tiến trình của Intent resolution, nó giúp chỉ định Intent đến một Actvity,

BroadcastReceiver, hoặc Service (hoặc thỉnh thoảng có thểlà 2 hay nhiều hơn một activity/receiver) để có thể xử lý các hành động được đặc tả trong Intent.

Bất cứ thành phần nào (Activity, BroadcastReceiver, Service) khi muốn sử dụng trong ứng dụng đều phải được đăng kí trong file AndroidManifest.xml. Trong đó cầnđịnh nghĩa một thẻ <intent-fillter> cung cấp các thông tin để hệ thống có thể xác định được cái mà các component này (Activity, BroadcastReceiver, Service) có thể xử lý được (những action mà component này có thể thực hiện được).

Intent Fillter là bản đặc tả có cấu trúc của các giá trị của Intent dùng để xác định component phù hợp để thực hiệncác hành động được đặc tả trong Intent. Một Intent Fillter nối các hành động, categories và dữ liệu (hoặc thông qua type hoặc sheme) trong Intent. Intent Fillter được khai báo trong AndroidManifest.xml và sử dụng thẻ intent-fillter.

Một Intent Fillter có các thành phần chính sau:

Action: Tên hành động mà component có thể thực thi.

Type:Kiểu hành động mà component có thể thực thi.

Category: Phân nhóm các hành động.

Đối với những dữ liệu không phải là nội dung cụ thể (VD: URI) thì việc xem xét lựa chọn Intent phù hợp sẽ dựa vào lược đồ(Scheme) của dữ liệu được cung cấp (VD: http:// mailto: …)

Luật xác định component phù hợp Intent

Để xác định một thành phần là phù hợp với một Intent hệ thống sẽ tiến hành xem xét từ trên xuống.

Trước tiên khi một Intent được gọi, Android sẽ tìm kiếm những component (Activity, BroadcastReceiver, Service) có action-name phù hợp với Intent.

Nếu có component phù hợp Android sẽmở component đó lên để thực thi các hành động theo yêu cầu.

Nếu có nhiều hơn 1 component có action-name phù hợp thì Android sẽ yêu cầu người dùng chọn component phù hợp.

Ngược lại nếu không có component nào phù hợp Android sẽ tiến hành xem xét kiểu dự liệu của Intent cung cấp xem có component nào có đủ năng lực để sử lý kiểu dữ liệu đó không. Nếu khôngđược Android sẽ tiến hành xem xét scheme của dữ liệu đó để tìm kiếm component phù hợp. Nếu vẫn không tìm được component phù hợp Android sẽ tiến hành xem xét các component có chung Category với Intent để xác định component.

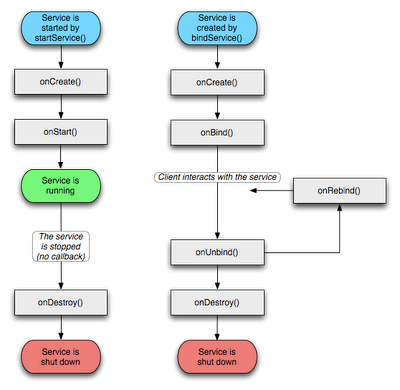
* Service

Một service không có giao diện trực quan, nó chạy trên nền trong một khoảng thời gian không xác định. Ví dụ một service có thể chơi nhạc nền, hay nó nạp dữ liệu trên mạng hay tính toán cái gì đó và cung cấp kết quả cho activity cần đến nó. Mỗi service mở rộng từ lớp cơ sở Service.

Một ví dụ cổ điển là media player chơi các bài hát từ một danh sách. Ứng dụng player có thể có một hay hai activity cho phép người dùng chọn bài hát và bắt đầu chơi. Tuy nhiên, music playback không tự nó được xử lý bởi một activity bởi vì người dùng sẽ mong muốn chơi tiếp nhạc khi họ rời khỏi player và làm cái gì đó khác. Để tiếp tục chơi nhạc, media player activity có thể khởi động một service chạy trên nền. Hệ thống sẽ giữ cho music playback service chạy thậm chí khi activity khởi động nó rời khỏi màn hình.

Trong khi kết nối, bạn có thể giao tiếp với dịch vụ thông qua giao diện mà service đó trưng ra. Ví dụ như trong music service, giao diện này có thể cho phép người dùng pause, rewind, stop và restart lại playback.

Giống như các activity và các thành phần khác khác, service chạy trong thread chính của tiến trình ứng dụng. Vì thế chúng không thể chặn những thành phần khác hay giao diện người dùng, chúng thường tạo ra các thead khác cho các nhiệm vụ hao tốn thời gian.



Hình 1..1c Service

* Broadcast Receiver

Một Broadcast Rreceiver 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 - ví dụ như thông báo múi giờ thay đổi, pin yếu, ảnh đã được chụp, hay người dùng đã thay đổi ngôn ngữ … Các ứng dụng có thể tạo ra các broadcast, chẳng hạn để ứng dụng khác biết được một số dữ liệu đã được tải về thiết bị và sẵn sàng cho việc sử dụng chúng. 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, lóe sáng đèn sau, rung, tạo ra âm thah, vân vân. Chúng thường lấy một biểu tượng bền vững trong thanh trạng thái, cái mà người dùng có thể mở để lấy thông điệp.

* Content Provider

Một content provider tạo ra một tập cụ thể các dữ liệu của ứng dụng khả dụng cho các ứng dụng khác. Dữ liệu có thể được lưu trữ trong hệ thống file, trong một cơ sở dữ liệu SQLite, hay trong một cách khác nào đó. Content provider mở rộng lớp cơ sở ContentProvider để cài đặt một tập các chuẩn các phương thức cho phép các ứng dụng khác đạt được và lưu trữ dữ liệu của kiểu mà nó điều khiển. Tuy nhiên, các ứng dụng không gọi trực tiếp các phương thức này, chúng sử dụng một đối tượng ContentResolver và gọi các phương thức của nó. Một ContentResolver có thể nói chuyện với bất cứ content provider nào, chúng cộng tác với provider để quản lý giao tiếp liên tiến trình.

* Manifest File

Trước khi Android có thể khởi động một thành phần ứng dụng, nó phải biết rằng thành phần đó tồn tại. Vì vậy, ứng dụng khai báo những thành phần của mình trong một manifest file được gắn vào Android package, file .apk này cũng giữ chứa mã của ứng dụng và các tài nguyên. Manifest file là một file XML có cấu trúc và thường được đặt tên là AndroidManifest.xml cho mọi ứng dụng. Nó thực hiện một số bổ sung để khai báo các thành phần của ứng dụng, như là nêu tên các thư viện ứng dụng cần đến, và xác định các quyền hạn của ứng dụng muốn được cấp. Ví dụ như nó khai báo các Activity trong chương trình của chúng ta, khi khởi động chương trình thì Activity nào sẽ được thực hiện đầu tiên, các quyền truy cập Internet, SDCard, hay các vấn đề về gửi nhận tin nhắn, cuộc gọi...

* View

Trong một ứng dụng Android, giao diện người dùng được xây dựng bằng cách sử dụng View và ViewGroup đối tượng. Có nhiều loại quan điểm và các nhóm view, mỗi một trong số đó là hậu duệ của lớp View. View objects là các đơn vị cơ bản của biểu hiện giao diện người dùng trên nền tảng Android. Các class xem như là cơ sở phục vụ cho class con được gọi là "widget", trong đó cung cấp đầy đủ các đối tượng thực hiện giao diện, giống như các lĩnh vực văn bản và nút. Class ViewGroup phục vụ như là cơ sở cho lớp con được gọi là " layouts", cung cấp các loại khác nhau của kiến trúc bố trí, như linear, tabular và relative.

* Các thành phần giao diện Widget

Có rất nhiều cách bố trí giao diện. Sử dụng nhiều hơn và các loại khác nhau của các view group, bạn có thể cấu trúc views con và view groups trong vô số cách. Xác định các nhóm xem được cung cấp bởi Android (gọi là layouts) bao gồm LinearLayout, RelativeLayout, TableLayout, GridLayout và khác. Mỗi cung cấp một bộ duy nhất của các thông số bố trí được sử dụng để xác định vị trí của views con và cơ cấu layout. Để tìm hiểu về một số các loại khác nhau của các view group được sử dụng cho một layout, đọc Giao diện đối tượng thường gặp.

Widget là một object View phục vụ như một giao diện để tương tác với người dùng. Android cung cấp một tập các widgets thực hiện đầy đủ, giống như các button, Checkbox, và text-entry , do đó bạn có thể nhanh chóng xây dựng giao diện người dùng của bạn. Một số widgets được cung cấp bởi Android phức tạp hơn, giống như một date picker, clock, và zoom controls. Nhưng nó không giới hạn trong các loại widgets được cung cấp bởi các nền tảng Android.

* Bắt sự kiện trong Android

Khi đã thêm một số Views/widgets đến giao diện. Để được thông báo về UI events người dùng, cần phải làm một trong hai điều: Xác định một sự kiện nghe và đăng ký nó với các View. Đây là cách lắng nghe sự kiện. Các class View có một phương pháp gọi là On<something>().

Ví dụ: View.OnClickListener (để xử lý "nhấp chuột" trên một View), View.OnTouchListener ( để xử lý các sự kiện màn hình cảm ứng trong một

View), và View.OnKeyListener (để xử lý thiết bị ép quan trọng trong một View). Vì vậy nếu muốn View được thông báo khi nó là " clicked" (chẳng hạn như khi một nút được chọn), thực hiện và xác định OnClickListener của nó gọi method onClick() (nơi thực hiện các hành động sau khi nhấp chuột), và đăng ký nó vào Xem với setOnClickListener().

Ghi đè một callback method hiện cho View. Đây là những gì nên làm gì khi đã thực hiện lớp View của riêng bạn và muốn lắng nghe cho các sự kiện cụ thể xảy ra trong nó. Ví dụ về các sự kiện có thể xử lý bao gồm màn hình là touched onTouchEvent() khi trackball là di chuyển onTrack ballEvent() hoặc khi một phím trên thiết bị được nhấn onKeyDown(). Điều này cho phép xác định các hành vi mặc định cho từng sự kiện bên trong tuỳ chỉnh View và xác định xem sự kiện này cần được thông qua ngày để View con khác. Một lần nữa, đây là những callbacks View class, do đó, cơ hội duy nhất của bạn để xác định đó là khi bạn xây dựng một phần tùy chỉnh.

* Menu

Menu đơn có một phần quan trọng của giao diện người dùng trong một ứng dụng. Menus cung cấp một giao diện đáng tin cậy cho thấy rằng các chức năng ứng dụng và cài đặt. Trong trình đơn ứng dụng phổ biến nhất là tiết lộ bằng cách bấm phím MENU trên thiết bị. Tuy nhiên, cũng có thể thêm Context Menus, có thể hiển thị khi người sử dụng máy nhấn và nắm giữ phím trên một mục. Thực đơn cũng được hệ thống phân cấp cấu trúc bằng cách sử dụng một xem, nhưng bạn không xác định cấu trúc này cho mình.

Thay vào đó, xác định onCreateOptionsMenu() or onCreateContext Menu() gọi method cho hoạt động của mình và tuyên bố các mục mà mình muốn bao gồm trong menu của mình. Trong một thời gian thích hợp, Android sẽ tự động tạo ra hệ thống View phân cấp cần thiết cho menu, và rút ra mỗi trong mỗi menu items đó.

## Các thư viện sử dụng

implementation 'com.android.support:support-v4:28.0.0'  
 implementation 'androidx.appcompat:appcompat:1.0.0-beta01'  
 implementation 'androidx.constraintlayout:constraintlayout:1.1.2'  
 implementation 'androidx.legacy:legacy-support-v4:1.0.0-beta01'  
 testImplementation 'junit:junit:4.12'  
 implementation 'com.google.android.material:material:1.0.0'  
 implementation 'androidx.recyclerview:recyclerview:1.0.0'  
 implementation 'androidx.cardview:cardview: 1.0.0'  
 implementation 'androidx.viewpager:viewpager:1.0.0'  
 implementation 'com.android.volley:volley:1.1.0'  
 implementation 'de.hdodenhof:circleimageview:3.0.1'  
 implementation 'com.squareup.picasso:picasso:2.71828'  
 implementation 'com.squareup.okhttp:okhttp:2.5.0'  
 implementation 'id.zelory:compressor:2.1.0'  
 implementation 'com.mikhaellopez:circularimageview:4.0.2'  
 implementation 'com.facebook.android:facebook-login:[5,6)'  
 implementation 'com.github.bumptech.glide:glide:3.7.0'  
 implementation 'com.android.support:support-v4:28.1.1'  
 implementation 'com.github.joaquimley:faboptions:1.1.2'  
 implementation 'com.android.support:appcompat-v7:28.0.1'  
 implementation 'com.android.support:support-v4:28.0.1'  
 implementation 'com.android.support:design:28.0.1'  
 implementation 'com.google.android.gms:play-services-auth:17.0.0'  
 implementation 'com.google.code.gson:gson:2.8.5'  
 implementation 'com.squareup.retrofit2:retrofit:2.4.0'  
 implementation 'com.squareup.retrofit2:converter-gson:2.4.0'

implementation 'com.google.firebase:firebase-database:19.2.0'  
 implementation 'com.google.firebase:firebase-auth:19.1.0'  
 implementation 'com.google.firebase:firebase-storage:19.1.0'  
 implementation 'com.google.firebase:firebase-messaging:20.0.0'  
 implementation 'com.google.firebase:firebase-core:17.2.1'

Plugin  
apply plugin: 'com.google.gms.google-services'

# TRIỂN KHAI ĐỒ ÁN

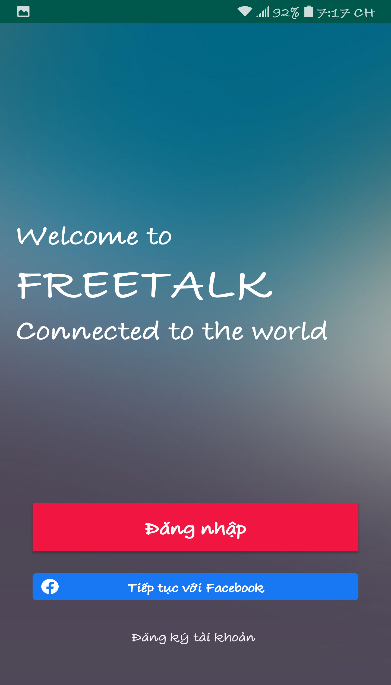
## Mô tả

* Về cơ bản, ứng dụng cần có:
* Giao diện đăng ký tài khoản, đăng nhập bằng Facebook, đăng nhập tài khoản người dùng.
* Giao diện hiển thị danh sách các cuộc trò chuyện với bạn bè, danh sách user ( những người lạ trong cộng đồng Chat ).
* Giao diện cá nhân gồm các thông tin người dùng.
* Giao diện hiển thị các đoạn chat giữa 2 người dùng với nhau.
* Về chức năng chính:
* Gửi tin nhắn giữa 2 người dùng thông qua Firebase gồm có văn bản, gửi hình ảnh và chụp hình.
* Thay đổi thông tin cá nhân gồm hình đại diện, ảnh bìa số điện thoại, tên hiển thị, tìm kiếm người dùng.

## Giao diện và chức năng

### Giao diện chờ và bắt đầu ứng dụng

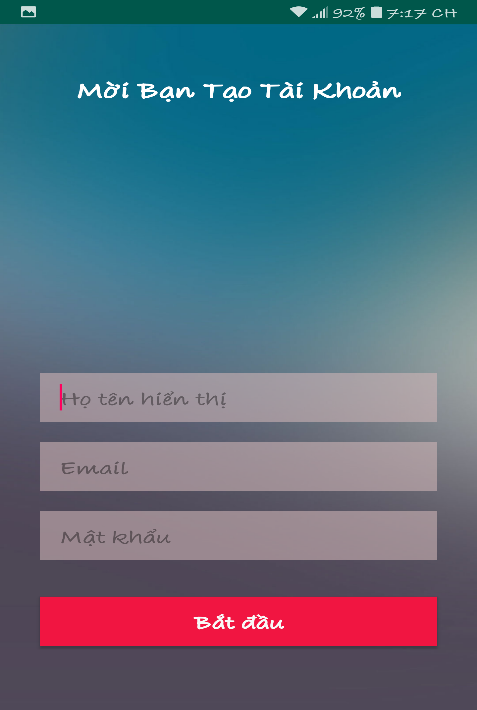
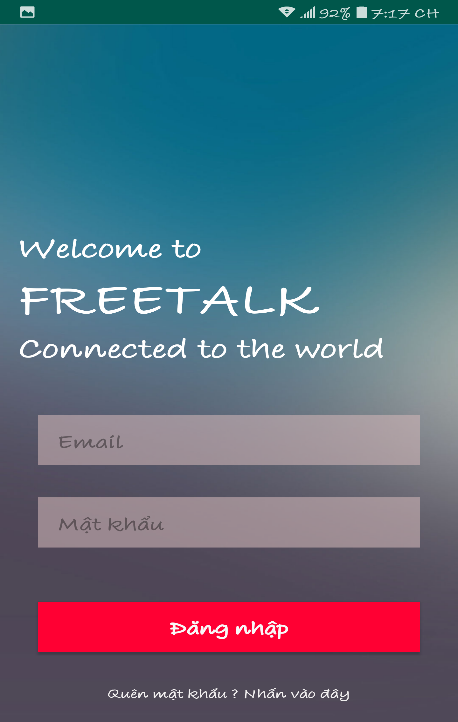
* Chức năng: hiển thị mà hình chờ khi chạy ứng dụng. Sau đó sẽ chuyển đến màn hình đăng nhập bắt đầu sử dụng ứng dụng với chức năng đăng nhập và tạo tài khoản tùy theo người dùng.



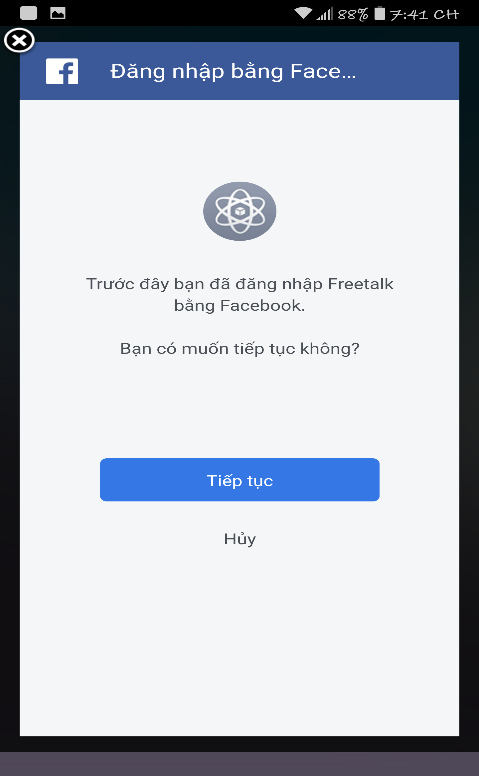
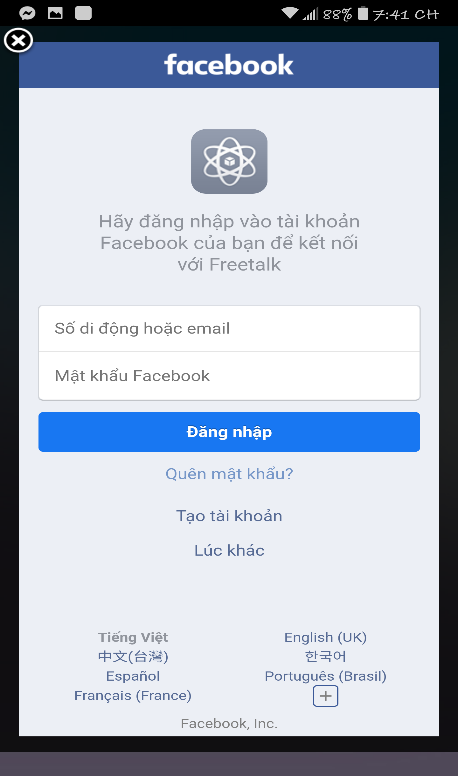
Hình .2.1 Giao diện chờ và bắt đầu ứng dụng

### Giao diện đăng ký người dùng và đăng nhập vào app

* Chức năng: đăng nhập vào ứng dụng, cho người dùng đăng ký khi chưa có tài khoản và đăng nhập bằng Facebook (nếu có).



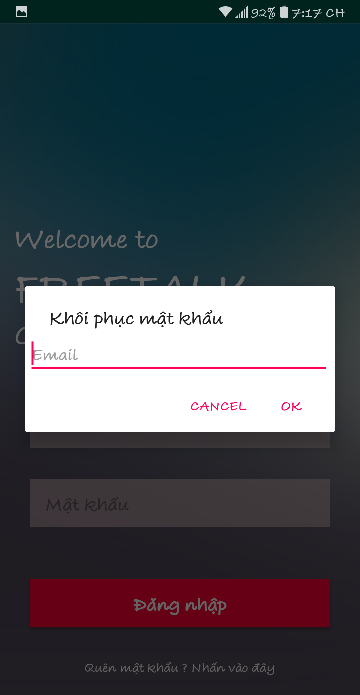
Hình .2.2a Giao diện bắt nhập và đăng ký tài khoản người dùng



Hình .2.2b Giao diện đăng nhập và xác thực bằng Facebook

### Giao diện khôi phục mật khẩu

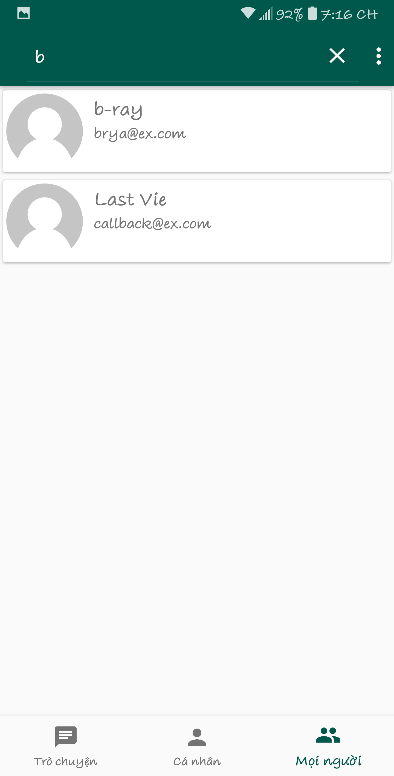
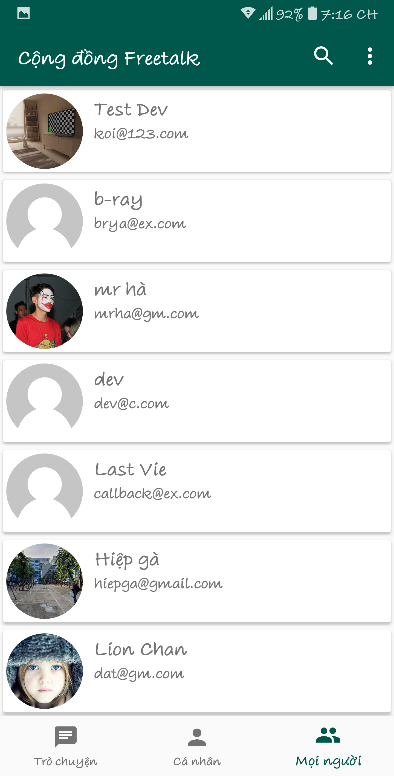
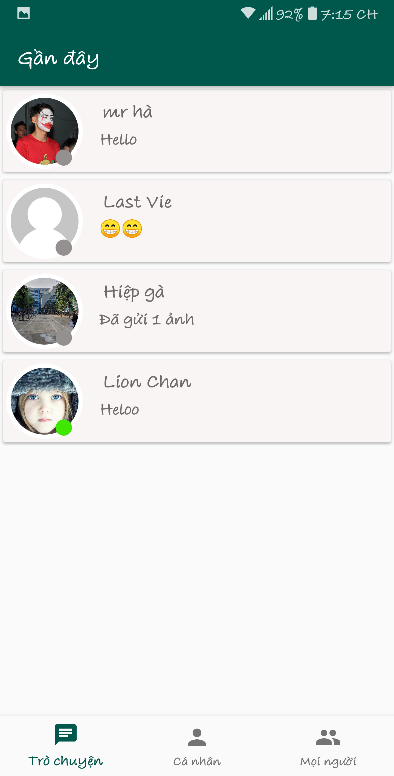
* Chức năng: khi người dùng quên mật khẩu ứng dụng sẽ gửi thông tin thay đổi mật khẩu qua email mà người dùng đã đăng ký.



Hình .2.3 Giao diện khôi phục mật khẩu

### Giao diện cuộc trò chuyện gần đây, cộng đồng và tìm kiếm người dùng

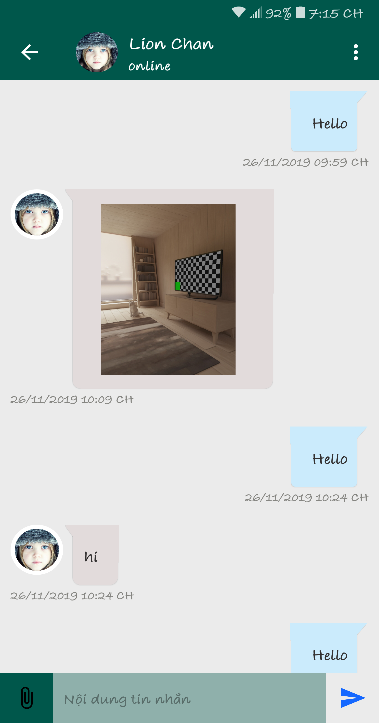
* Chức năng: hiển thị danh sách các cuộc trò chuyện và danh sách cộng đồng cùng sử dụng ứng dụng.



Hình .2.4 Giao diện trò chuyện, cộng đồng và tìm kiếm người dùng

### Giao diện chat

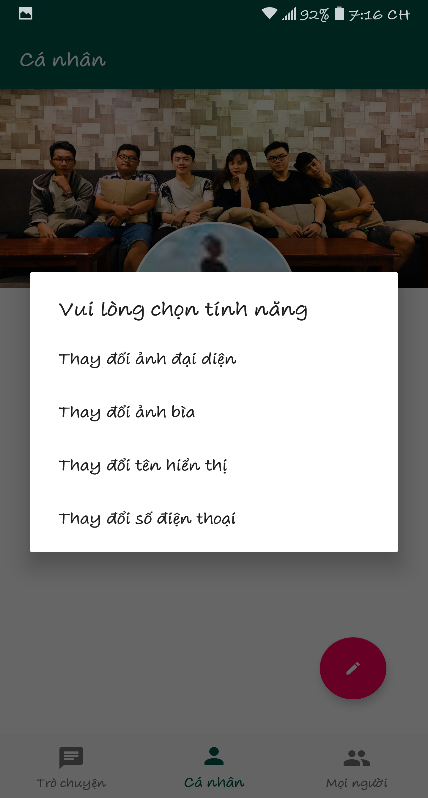
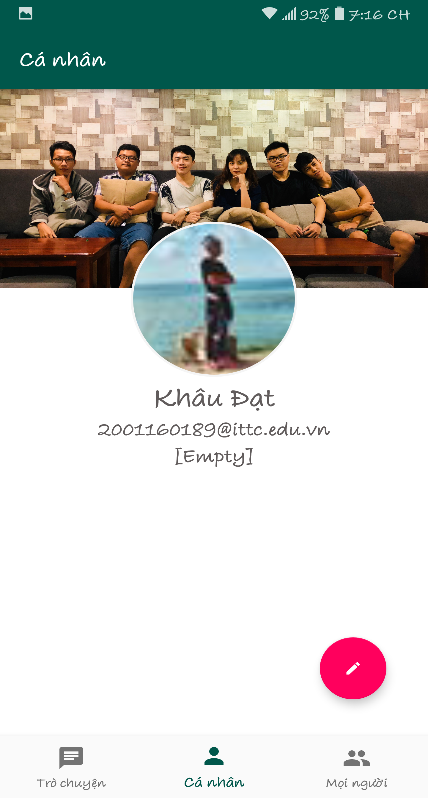
* Chức năng: nhắn tin giữa 2 người dùng, xem tin nhắn, realtime, gửi hình ảnh, chụp ảnh. Bên trái là hiển thị tin nhắn người nhận, bên phải hiển thị tin nhắn người gửi.

****

Hình .2.5 Giao diện chat

### Giao diện cá nhân và thay đổi thông tin cá nhân

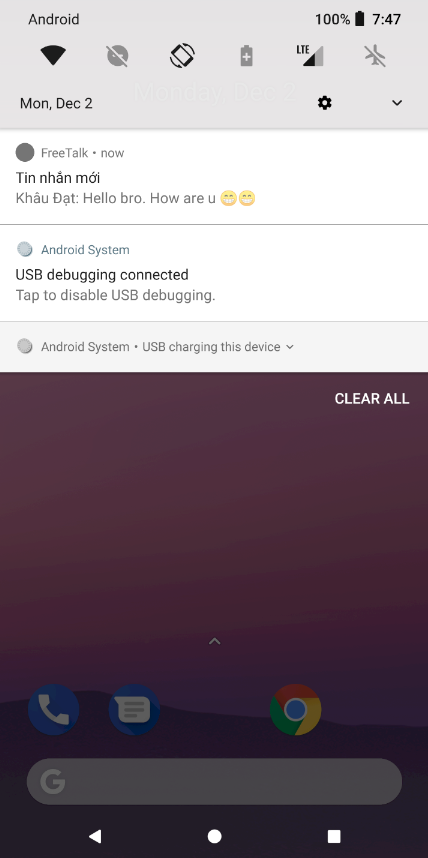
* Chức năng: hiển thị thông tin cá nhân và cho phép chỉnh sửa thông tin như ảnh đại diện, ảnh bìa, tên hiển thị và số điện thoại.



Hình .2.6 Giao diện cá nhân và thay đổi thông tin tài khoản

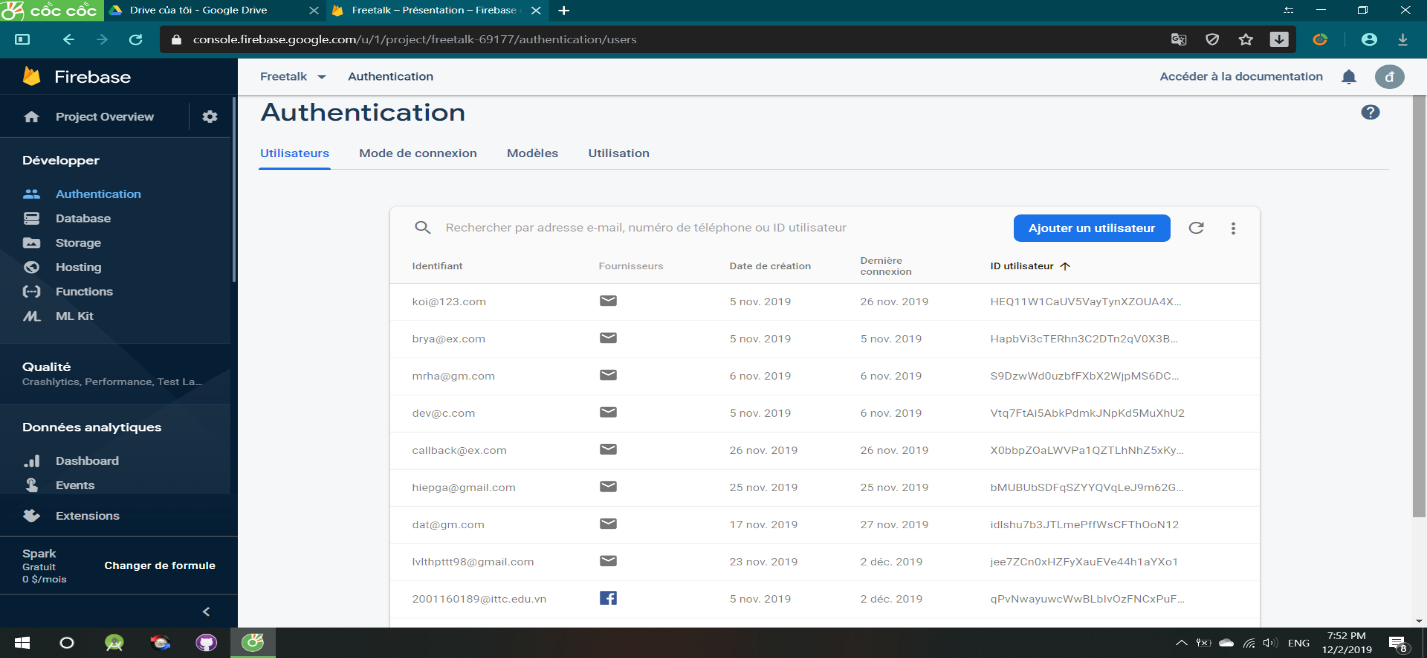
### Giao diện thông báo tin nhắn mới

* Chức năng: thông báo tin nhắn mới tới người dùng.

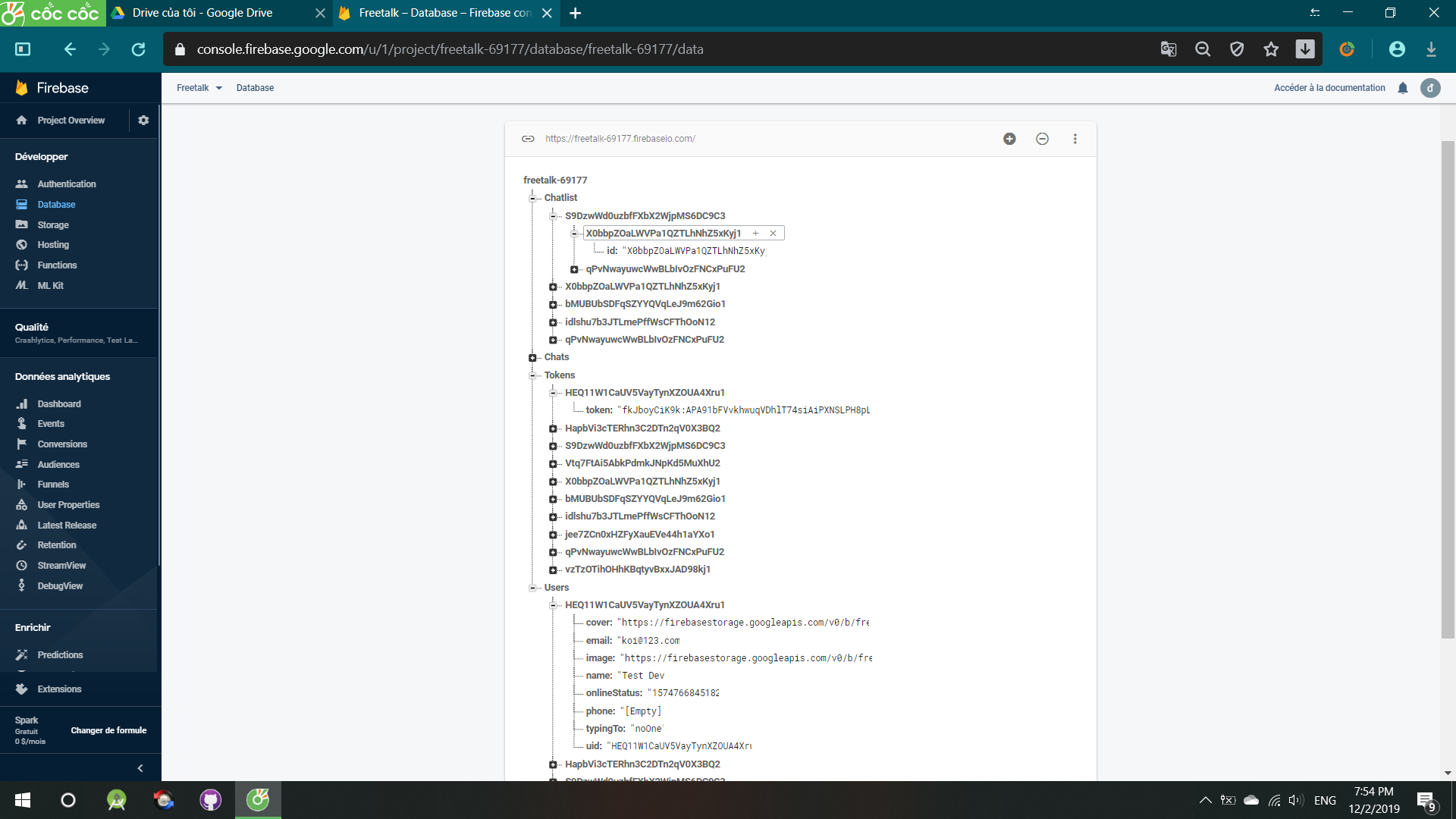
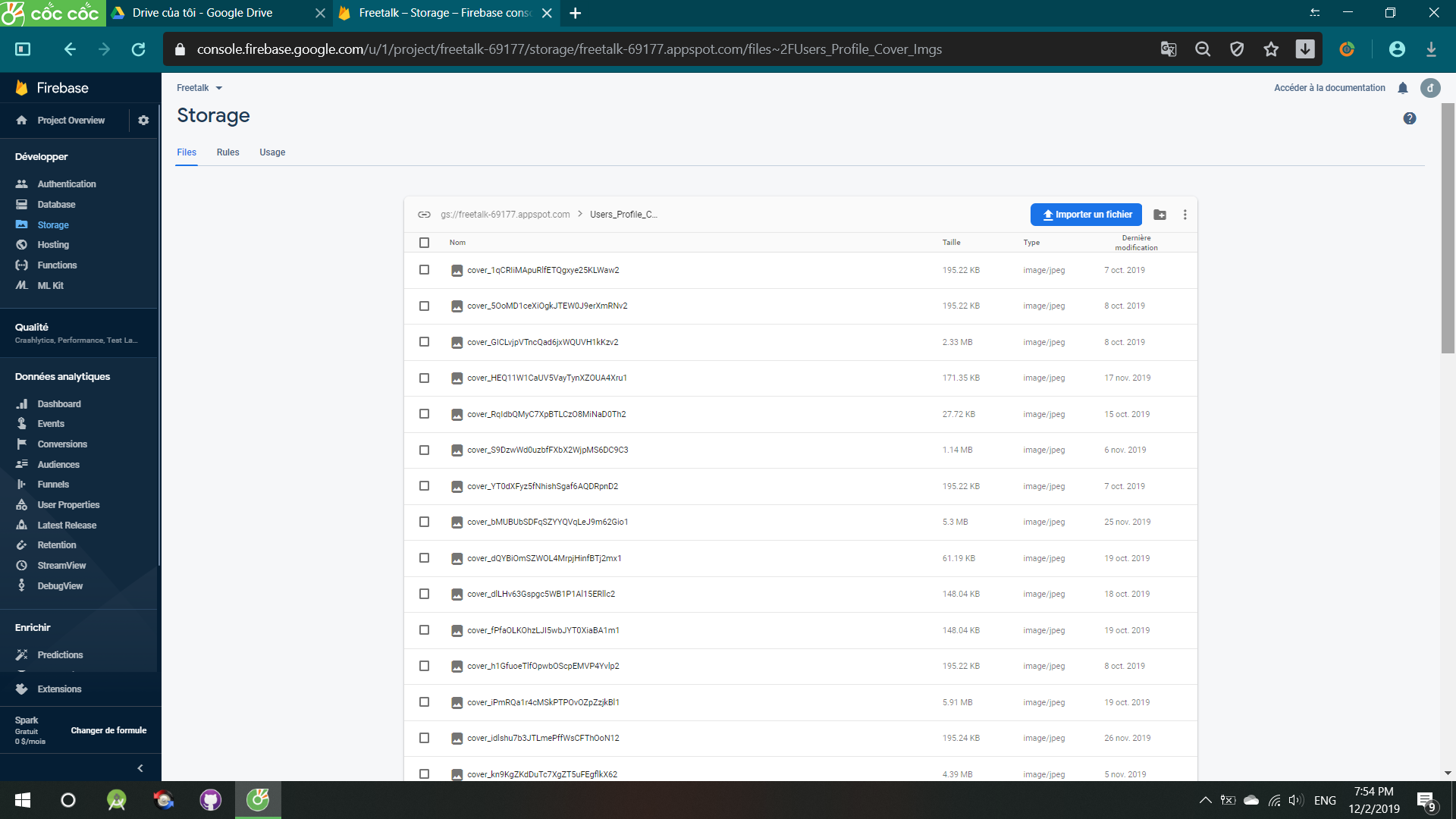


Hình .2.7 Giao diện thông báo tin nhắn tới

## Cơ sở dữ liệu

Ứng dụng lưu trữ dữ liệu, thông tin và người dùng thông qua Google Firebase

Hình .3.1 Firebase Auth



Hình .3.3 Firebase Storage

Hình .3.2 Firebase Database

# KẾT LUẬN VÀ DEMO

## Kết luận

Sau một thời gian tìm hiểu đề tài “Tìm hiểu về Android và xây dựng ứng dụng chat realtime” em đã thực hiện được đề tài của mình theo như dự kiến.Với mục đích là tìm hiểu về hệ điều hành Android, hiểu biết về ngôn ngữ lập trình cho điện thoại di động và tìm hiểu các ứng dụng của hệ điều hành Android trong thế giới di động ngày nay. Thông qua việc tìm hiểu và nắm bắt lý thuyết về hệ điều hành Android để xây dựng ứng dụng chat realtime trên hệ điều hành Android cho điện thoại di động. Sau quá trình tìm hiểu và nghiên cứu đề tài, em đã bước đầu hiểu về hệ điều hành Android chạy trên điện thoại. Biết và nắm vững quá trình tạo một file project trong lập trình trên nền Android. Hiểu và sử dụng được các lớp đối tượng, phương thức hỗ trợ lập trình trên nền hệ điều hành Android. Hiểu được cơ chế làm việc với Firebase để lấy và cập nhật dữ liệu cho những ứng dụng cần thông tin nhanh chóng, thường xuyên và chính xác hơn. Do thời gian ngắn cùng với kiến thức của bản thân còn hạn chế nên em chưa tìm hiểu sâu về các kiến thức liên quan tới hệ điều hành Android nên chương trình vẫn còn rất nhiều hạn chế, em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và nhiều chức năng hơn.

## Hướng phát triển đề tài

Ứng dụng đã phần nào đáp ứng được yêu cầu đặt ra ban đầu. Tuy nhiên, ứng dụng còn nhiều nhược điểm và cần phát triển hơn nữa trong tương lai. So với những phần mềm chat khác, hệ thống còn thiếu một số chức năng khác như:

* Chat nhóm.
* Video chat.
* Voice chat.
* Gửi và nhận file.

Nếu phát triển hệ thống hơn nữa cần hướng tới công nghệ mới hơn, hoàn thiện hơn, cũng như phát triển thêm nhiều tính năng mới không chỉ là các chức năng kể trên. Để đạt được những tính năng mới đòi hỏi quá trình nghiên cứu sâu và kĩ lưỡng không chỉ trên Google Firebase mà còn trên các công nghệ hiện đại khác.

TÀI LIỆU THAM KHẢO

**Tiếng anh**

1. Android Notes for Professionals
2. G. Sparks. (2000) An Introduction to modelling software systems using the Unified [Online]. HYPERLINK <https://sparxsystems.com/>
3. D. Lebrknight R. LeMaster, Object-Oriented Programming &amp; Design”, CSCI 4448.: University of Colorado, 2002.
4. K. Scott M. Fowler, UML Distilled Second Edition – A Brief Guide to the Standard Object Modelling Langguage.: Addison Wesley Book, 1999.

**Tiếng việt**

1. Th.s Vũ Đình Ái, Bài giảng Lập trình di động

**Website**

1. http://vi.wikipedia.org/wiki/Android
2. https://firebase.google.com
3. http://eitguide.net/firebase-bai-mo-dau-gioi-thieu-ve-firebase
4. <https://techtalk.vn/google-firebase-la-gi-tao-ung-dung-chat-firebase.html>
5. <https://hoangquoctuanpro.wordpress.com/gioi-thieu/lap-trinh-android/>
6. https://www.slideshare.net/tunkutunku/slide-bi-ging-lp-trnh-android-dtu-phn-1