

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----



ISO 9001:2015

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

**XÂY DỰNG ỨNG DỤNG CHAT TRONG ANDROID
VỚI FIREBASE**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ Thông tin

HẢI PHÒNG 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

XÂY DỰNG ỨNG DỤNG CHAT TRONG ANDROID VỚI FIREBASE

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Bùi Văn Vũ
Giáo viên hướng dẫn: Th.S Phùng Anh Tuấn
Mã sinh viên: 1412101076

HẢI PHÒNG 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Bùi Văn Vũ

Mã số: 1412101076

Lớp: CT1801

Ngành: Công nghệ Thông tin

Tên đề tài: Xây dựng ứng dụng chat trong Android với Firebase

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

Tìm hiểu về hệ điều hành android.

Tìm hiểu về môi trường lập trình Android Studio.

Tìm hiểu về nền tảng hỗ trợ cho lập trình di động firebase.

Tìm hiểu các kỹ thuật lập trình ứng dụng chat trong android với firebase.

Xây dựng chương trình chat trên điện thoại

b. Các yêu cầu cần giải quyết

Sử dụng công cụ Android Studio để viết chương trình chạy trên thiết bị android.

Tìm hiểu kiến thức cơ bản về nền tảng firebase.

Xây dựng ứng dụng chat android với firebase có các chức năng:

- + Đăng ký, đăng nhập.
- + Hiện thị danh sách tài khoản.
- + Chỉnh sửa profile.
- + Chat trên kênh riêng tư.
- + Xóa tin nhắn chat.
- + Thoat khỏi hệ thống.

Đóng gói chương trình thành file *.apk cho phép cài đặt và chạy trên thiết bị android thật.

2. Các số liệu cần thiết để thiết kế, tính toán

.....

3. Địa điểm thực tập

Trường Đại học Dân lập Hải Phòng

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Phùng Anh Tuấn

Học hàm, học vị: Thạc sỹ

Cơ quan công tác: Trường Đại học Dân lập Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu về môi trường lập trình Android Studio.
- Tìm hiểu về nền tảng hỗ trợ cho lập trình di động firebase.
- Tìm hiểu các kỹ thuật lập trình ứng dụng chat trong android với firebase.
- Xây dựng được ứng dụng chat chạy trên thiết bị ảo và điện thoại thật.

Người hướng dẫn thứ hai:

Họ và tên:.....

Học hàm, học vị:.....

Cơ quan công tác:.....

Nội dung hướng dẫn:.....

Đề tài tốt nghiệp được giao ngày 14 tháng 10 năm 2019

Yêu cầu phải hoàn thành trước ngày 10 tháng 01 năm 2020

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngàytháng.....năm 2020

Hiệu trưởng

GS.TS.NGƯT Trần Hữu Nghị

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: Ngành:.....

Nội dung hướng dẫn:

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

.....
.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T. T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...)

.....
.....
.....
.....
.....
.....
.....
.....
.....

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Đạt Không đạt Điểm:.....

Hải Phòng, ngày tháng ... năm 2020

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIÁNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên:.....

Đơn vị công tác:

Họ và tên sinh viên: Ngành:

Đề tài tốt nghiệp:

1. Phần nhận xét của giảng viên chấm phản biện

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Những mặt còn hạn chế

.....

.....

.....

.....

.....

.....

.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm:.....

Hải Phòng, ngày tháng ... năm 2020

Giảng viên chấm phản biện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Để đồ án này đạt được kết quả như ngày hôm nay lời đầu tiên em xin chân thành cảm ơn đến tất cả mọi người đã hỗ trợ giúp đỡ em về mặt kiến thức cũng như tinh thần để hoàn thành đồ án này.

Em xin chân thành cảm ơn thầy cô giáo khoa Công nghệ thông tin và toàn thể thầy cô giáo trường Đại học Dân Lập Hải Phòng đã giúp đỡ truyền đạt cho em những kiến thức quý báu những năm học tại trường. Đặc biệt em xin cảm ơn thầy Ths. Phùng Anh Tuấn người đã hướng dẫn chỉ bảo trực tiếp em để hoàn thành được đồ án này trong thời gian qua.

Với điều kiện thời gian có hạn cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này của em không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, hoàn thiện hơn chương trình của mình để áp dụng vào thực tế sau này.

Em xin chân thành cảm ơn!

Sinh viên

Bùi Văn Vũ

MỤC LỤC

| | |
|-------------------------------------------------------------------|-----------|
| CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDROID..... | 5 |
| 1.1 Giới thiệu hệ điều hành android..... | 5 |
| 1.1.1 Hệ điều hành | 5 |
| 1.1.2 Hệ điều hành Android | 5 |
| 1.2 Sự ra đời và lịch sử phát triển | 6 |
| 1.2.1 Sự ra đời của Android | 6 |
| 1.2.2 Lịch sử phát triển của Android | 7 |
| 1.3 Các phiên bản của Android..... | 8 |
| 1.3.1 Phiên bản Android 1.0 | 8 |
| 1.3.2 Phiên bản Android 1.5: CupCake | 9 |
| 1.3.3 Phiên bản Android 1.6: Donut | 9 |
| 1.3.4 Phiên bản Android 2.0 và phiên bản Android 2.1: Eclair..... | 9 |
| 1.3.5 Phiên bản Android 2.2: Froyo..... | 9 |
| 1.3.6 Phiên bản Android 2.3: Gingerbread | 10 |
| 1.3.7 Phiên bản Android 3.0: Honeycomb..... | 10 |
| 1.3.8 Phiên bản Android 4.0: Ice Cream Sandwich..... | 10 |
| 1.3.9 Phiên bản Android 4.1: Jelly Bean | 10 |
| 1.3.10 Hệ điều hành Android 4.4: KitKat..... | 11 |
| 1.3.11 Hệ điều hành Android 5.0: Lollipop..... | 11 |
| 1.3.12 Phiên bản Android 6.0: Marshmallow | 11 |
| 1.3.13 Phiên bản Android 7.0:nougat | 11 |
| 1.3.14 Phiên bản 8.0:Oreo..... | 12 |
| 1.3.15 Phiên bản Android P | 12 |
| 1.4 Kiến trúc của hệ điều hành Android | 12 |
| 1.4.1 Linux kernel | 13 |
| 1.4.2 Tầng Library và Android Runtime | 14 |
| 1.4.3 Tầng Application Framework | 15 |
| 1.4.4 Tầng ứng dụng | 16 |
| CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO | 18 |
| 2.1 Sơ lược về Android Studio | 18 |

| | | |
|---------------------------------------------------|----------------------------------------------------|-----------|
| 2.2 | Cài đặt và sử dụng Android Studio..... | 18 |
| 2.2.1 | Cấu hình yêu cầu..... | 18 |
| 2.2.2 | Phần mềm Android Studio..... | 19 |
| 2.2.3 | Dự án trong Android studio..... | 25 |
| 2.3 | Thiết bị ảo trong Android Studio..... | 31 |
| CHƯƠNG 3: Lập trình Chat với Firebase..... | | 41 |
| 3.1 | Tổng quan về Firebase..... | 41 |
| 3.1.1 | Khái niệm..... | 41 |
| 3.1.2 | Lịch sử phát triển của Firebase..... | 41 |
| 3.1.3 | Ưu nhược điểm của Firebase..... | 42 |
| 3.2 | Tổng quan dịch vụ Firebase..... | 42 |
| 3.2.1 | Dịch vụ Firebase Analytics..... | 43 |
| 3.2.2 | Các dịch vụ phát triển và kiểm thử ứng dụng..... | 43 |
| 3.2.3 | Các dịch vụ tăng trưởng và thu hút người dùng..... | 45 |
| 3.3 | Một số kỹ thuật cơ bản..... | 47 |
| 3.3.1 | Tạo dự án trên Firebase..... | 47 |
| 3.3.2 | Nhúng dự án Firebase vào ứng dụng..... | 49 |
| 3.3.3 | Quản lý người dùng..... | 51 |
| 3.3.4 | Chat trên kênh riêng tư..... | 60 |
| 3.3.5 | Xóa tin nhắn chat..... | 62 |
| 3.3.6 | Đăng xuất..... | 64 |
| 3.4 | Bài toán..... | 64 |
| 3.5 | Sơ đồ chức năng..... | 65 |
| 3.6 | Thiết kế dữ liệu..... | 65 |
| 3.6.1 | Bảng dữ liệu người dùng..... | 65 |
| 3.6.2 | Bảng thông tin nội dung tin nhắn..... | 66 |
| 3.7 | Giao diện chương trình..... | 66 |
| 3.7.1 | Giao diện chính..... | 66 |
| 3.7.2 | Giao diện đăng ký..... | 67 |
| 3.7.3 | Giao diện đăng nhập..... | 68 |
| 3.7.4 | Giao diện thông tin người dùng..... | 68 |

| | | |
|---------------------------|-------------------------------------|-----------|
| 3.7.5 | Giao diện danh sách người dùng..... | 70 |
| 3.7.6 | Giao diện chat | 71 |
| KẾT LUẬN | | 72 |
| TÀI LIỆU THAM KHẢO | | 73 |

CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDROID

1.1 Giới thiệu hệ điều hành android

1.1.1 Hệ điều hành

Hệ điều hành là một phần mềm dùng để điều hành, quản lý các thiết bị phần cứng và các tài nguyên phần mềm. Hệ điều hành đóng vai trò trung gian trong việc giao tiếp giữa người sử dụng và phần cứng của máy, cung cấp một môi trường cho phép người sử dụng phát triển và thực hiện các ứng dụng của họ một cách dễ dàng

1.1.2 Hệ điều hành Android

Android là hệ điều hành trên điện thoại di động có màn hình cảm ứng như điện thoại thông minh, máy tính bảng và trên một số đầu phát HD, HD Player, TV. Được phát triển bởi Google và dựa trên nền tảng Linux[1]. Trước đây Android được phát triển bởi công ty liên hợp Android, sau đó được Google mua lại vào năm 2005. Các nhà phát triển viết ứng dụng bản cho Android bằng ngôn ngữ Java. Android ra mắt lần đầu tiên vào ngày 5 tháng 11 năm 2007 và tuyên bố thành lập Liên minh thiết bị cầm tay mở. Một hiệp hội được thành lập với các công ty phần cứng, phần mềm và viễn thông với mục đích tạo nên một chuẩn mở cho điện thoại thông minh trong tương lai.

Android là hệ điều hành mã nguồn mở cùng với việc phát hành mã nguồn mở theo giấy phép Apache của Google. Apache là một giấy phép không có nhiều ràng buộc cùng với mã nguồn mở đã cho phép các nhà phát triển thiết bị di, các nhà mạng và các lập trình viên nhanh chóng tiếp cận điều chỉnh và phân phối Android một cách tự do. Một số lượng lớn lập trình viên chuyên viết các ứng dụng mở rộng chức năng của các thiết bị, bằng ngôn ngữ lập trình Java có sửa đổi.

Nhờ yếu tố mở và một giấy phép không có nhiều ràng buộc, Android ngày nay đã trở thành một trong những nền tảng di động nổi tiếng khắp thế

giới. Bằng chứng là với bản chất “mở” của mình thì Android đã thu hút được một cộng đồng lập trình viên, chuyên viên phát triển ứng dụng, theo thống kê tháng 10 năm 2012 đã xuất hiện khoảng 700.000 ứng dụng Android và số lượng tải từ Google Play, cửa hàng ứng dụng chính của Android được ước tính vào khoảng 25 tỷ lượt. Về mặt thiết bị thì nền tảng này liên tục được các nhà sản xuất thiết bị lựa chọn khi mà Android hội tụ quá nhiều ưu điểm như: vận hành nhẹ nhàng, có khả năng tinh chỉnh tốt và tiết kiệm chi phí để chạy trên các thiết bị công nghệ cao của mình.



Hình 1.1.1: Kho ứng dụng Android

Vào thời điểm quý II năm 2017 Android đã chiếm 87,7% thị phần smartphone trên toàn thế giới với tổng cộng 2 tỷ thiết bị được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Đây có thể được xem là một dấu ấn mạnh mẽ nhất trong công cuộc khẳng định vị thế của Android trên thị trường quốc tế.

1.2 Sự ra đời và lịch sử phát triển

1.2.1 Sự ra đời của Android

Cha đẻ của Android: hệ điều hành Android được lấy tên từ Android, Inc công ty khai sinh ra nó[1]. Công ty này được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin, Rich Miner, và Chris White. Mặc dù những người sáng lập và nhân viên của công ty đều là những người có tiếng tăm, Android Inc vẫn hoạt động một cách âm thầm, chỉ tiết lộ

rằng họ đang làm phần mềm cho điện thoại di động. Cuối năm 2003, Rubin cạn vốn. Bạn thân của ông Steve Perlman, đã hùn vốn 10 ngàn USD nhưng từ chối tham gia vào công ty.

1.2.2 Lịch sử phát triển của Android

Trở thành một phần của Google: ngày 17/08/2005 Google mua lại Android Inc, biến nó thành một phần trực thuộc của Google. Những nhân viên chủ chốt của tổng công ty Android vẫn là Rubin, Miner và While vẫn tiếp tục ở lại phát triển hệ điều hành của họ. Quyết định này đưa ra để sử dụng Linux làm nền tảng cho hệ điều hành Android và điều đó cũng có nghĩa là Android sẽ được cung cấp miễn phí cho các nhà sản xuất điện thoại di động của bên thứ ba. Google và nhóm Android cảm thấy công ty có thể kiếm tiền với các dịch vụ khác sử dụng hệ điều hành, bao gồm cả ứng dụng.

Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động dựa trên nhân Linux, được họ tiếp thị đến các nhà sản xuất thiết bị cầm và các nhà mạng trên những tiền đề về việc cung cấp một hệ thống mềm dẻo, có khả năng nâng cấp mở rộng cao. Một số nguồn tin cho biết trước đó Google đã lên danh sách các thành phần phần cứng và các đối tác phần mềm, đồng thời ra hiệu với các nhà mạng rằng họ sẵn sàng hợp tác ở nhiều cấp độ khác nhau. Ngày càng nhiều suy đoán rằng Google sẽ tham gia thị trường điện thoại di động xuất hiện trong tháng 12 năm 2006. Tin tức của BBC và Nhật báo phố Wall chú thích rằng Google muốn đưa công nghệ tìm kiếm và các ứng dụng của họ vào điện thoại di động và họ đang nỗ lực làm việc để thực hiện điều này. Các phương tiện truyền thông in và online cũng sớm có bài viết về những tin đồn cho rằng Google đang phát triển một thiết bị cầm tay mang thương hiệu của Google. Càng có nhiều suy đoán sau bài viết về việc Google đang định nghĩa các đặc tả công nghệ và trình diễn các mẫu thử với các nhà sản xuất điện thoại di động và nhà mạng.

Android liên tục được phát triển, mỗi bản cập nhật từ Google là mỗi lần Android được tối ưu hóa hoạt động tốt hơn, nhanh và ổn định hơn, hỗ trợ thêm công nghệ mới. Chẳng hạn theo đánh giá thì Android phiên bản 2.2 hoạt động nhanh hơn bản 2.1 tới 450%. Hiện nay, phiên bản mới nhất của Android là phiên bản Android Q.

1.3 Các phiên bản của Android

Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên[2]. Android 1.0 được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, và phiên bản hiện tại Lollipop.

Kỷ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán tại Mỹ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng như giao diện đặc sản của hệ điều hành này đã khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này.

1.3.1 Phiên bản Android 1.0

Lần đầu tiên phiên bản Android được phát hành vào ngày 23 tháng 9 năm 2008. So với những hệ điều hành di động phổ biến thời điểm đó, Android đem đến nhiều trải nghiệm mới mẻ độc đáo như màn hình Home Screen, thành thông báo Notifications được kích hoạt theo phương thức vuốt dọc màn hình theo chiều từ trên xuống. Ngoài ra, giới công nghệ còn hết sức ấn tượng với khả năng liên kết với Google Mail của Android 1.0. Sản phẩm làm lên tên tuổi cho Android sau đó HTC Cream, ra mắt 1 năm sau đó.

1.3.2 Phiên bản Android 1.5: CupCake

Tên mã công khai chính thức đầu tiên của Android đã không xuất hiện cho đến phiên bản 1.5 Cupcake vào ngày 27 tháng 4 năm 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 chiều 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.

1.3.3 Phiên bản Android 1.6: Donut

Google đã nhanh chóng tung ra Android 1.6 Donut vào tháng 9 năm 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,...

1.3.4 Phiên bản Android 2.0 và phiên bản Android 2.1: Eclair

Vào tháng 10 năm 2009, sau khoảng một năm tung ra bản Android 1.0, Google đã tung ra bản 2.0 với tên là Eclair. Đây 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 Google Maps trong số nhiều tính năng và cải tiến mới khác.

1.3.5 Phiên bản Android 2.2: Froyo

Ra mắt vào tháng 5 năm 2010, Android 2.2 Froyo 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 di động Wifi, đẩy thông báo qua dịch vụ C2DM của Android Cloud, hỗ trợ flash và nhiều hơn thế nữa.

1.3.6 Phiên bản Android 2.3: Gingerbread

Được ra mắt vào tháng 9 năm 2010, hiện là phiên bản cũ nhất của hệ điều hành mà Google vẫn đang liệt kê trong trang cập nhật phiên bản hàng tháng. Tính đến ngày 13/09/2017 Google chỉ ra rằng chỉ 0,6% của tất cả các thiết bị Android hiện đang chạy phiên bản 2.3. Hệ điều hành này đã thêm hỗ trợ chức năng giao tiếp NFC.

1.3.7 Phiên bản Android 3.0: Honeycomb

Honeycomb được Google phát hành chỉ để cài đặt trên máy tính bảng và các thiết bị di động có màn hình lớn hơn so với điện thoại thông minh hiện tại. Honeycomb lần đầu tiên được giới thiệu vào tháng 2 năm 2011, cùng với những thiết kế phù hợp với máy tính bảng.

1.3.8 Phiên bản Android 4.0: Ice Cream Sandwich

Được phát hành vào tháng 10 năm 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. 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 sinh trắc học đã phát triển và cải thiện đáng kể từ đó.

1.3.9 Phiên bản Android 4.1: Jelly Bean

Kỷ nguyên Jelly Bean của Android bắt đầu vào tháng 6 năm 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. Hiện tại có khoảng 6.9% của tất cả các sản phẩm Android sử dụng Jelly Bean.

1.3.10 Hệ điều hành Android 4.4: KitKat

KitKat không có nhiều 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.

1.3.11 Hệ điều hành Android 5.0: Lollipop

Được ra mắt lần đầu vào mùa thu năm 2014, Android 5.0 Lollipop là một thay đổi lớn trong giao diện tổng thể của hệ điều hành. Đây là phiên bản đầu tiên sử dụng ngôn ngữ Material Design của Google, sử dụng các hiệu ứng ánh sáng và bóng tối tự do. UI cũng có thay đổi khác cho hình khóa và nhiều hơn nữa. Bản cập nhật Android 5.1 đã có một số thay đổi. tính năng này bao gồm hỗ trợ chính thức cho các cuộc gọi đôi, cuộc gọi thoại HD và bảo vệ thiết bị ngay sau khi khôi phục cài đặt gốc

1.3.12 Phiên bản Android 6.0: Marshmallow

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.

1.3.13 Phiên bản 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.

1.3.14 Phiên bản 8.0:Oreo

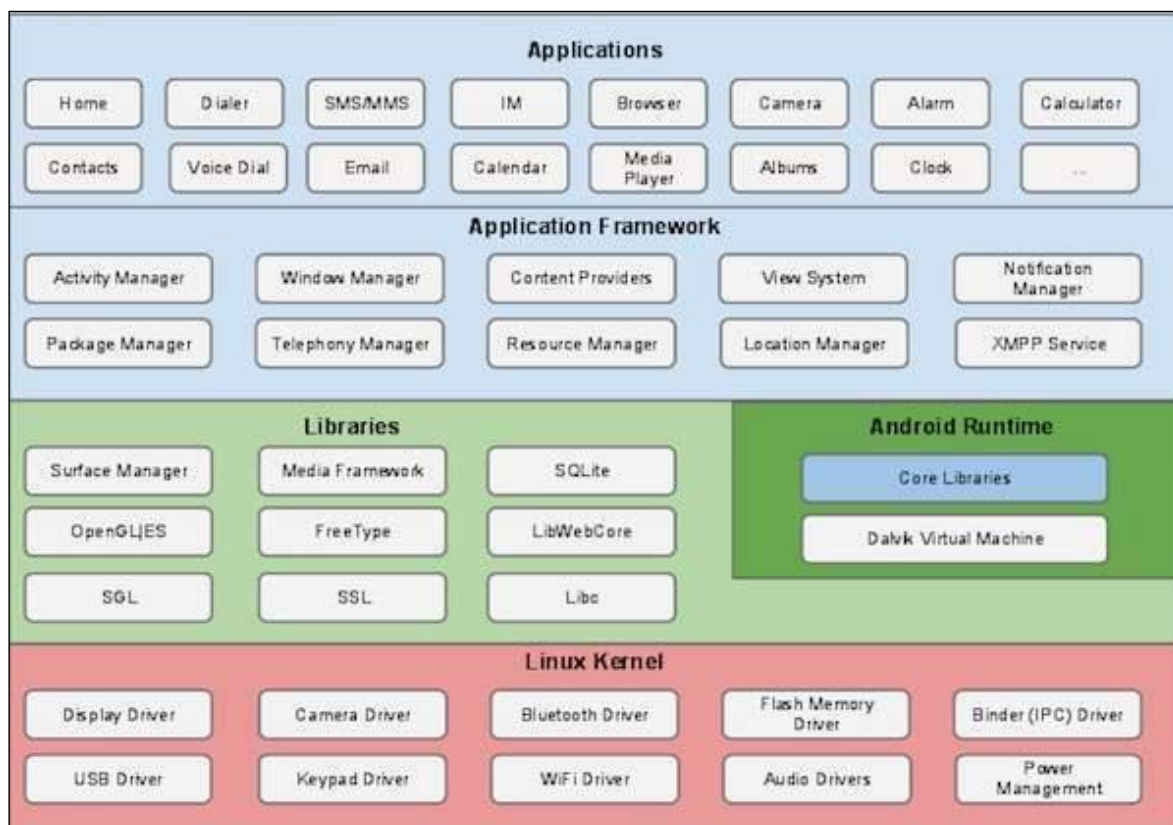
Vào tháng 3 năm 2017, Google chính thức công bố và phát hành bản xem trước dành cho nhà phát triển đầu tiên dành cho Android O, còn được gọi là Android 8.0. Đây là lần thứ hai Google chọn một tên thương hiệu cho Android (Oreo thuộc sở hữu của Nabisco). Android Oreo bao gồm nhiều thay đổi trực quan trong 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.

1.3.15 Phiên bản 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".

1.4 Kiến trúc của hệ điều hành Android

Hệ điều hành Android là một chồng của các thành phần phần mềm được tạm chia thành năm phần và bốn lớp chính như hình dưới đây trong sơ đồ kiến trúc[1].



Hình 1.4.1: Kiến trúc hệ điều hành android

1.4.1 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 đều 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 tiến 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, dung lượng bộ nhớ, kích thước màn hình, nhu cầu kết nối mạng không dây...

Ở tầng này có các thành phần chủ yếu:

- 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.

1.4.2 Tầng Library và Android Runtime

1.4.2.1 Phần Library

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 nhóm như:

- Thư viện hệ thống (System C Library): thư viện dựa trên chuẩn C, được sử dụng chỉ 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 phần 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.

1.4.2.2 Phần 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, Collection, File Access. Thứ hai là 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.

1.4.3 Tầng Application Framework

Tầng này xây dựng bộ công cụ- các phần ở mức cao để các lập trình viên có thể nhanh chóng xây dựng ứng dụng. Nó được viết bằng Java, có khả năng sử dụng chung để tiết kiệm tài nguyên.

Đây là một nền mở, có ưu điểm:

Với hãng sản xuất điện thoại: có thể tùy biến để phù hợp với cấu hình điện thoại mà họ sản xuất như để có nhiều mã, kiểu dáng hợp thị hiếu người dùng. Vì thế nên tuy cùng chung nền tảng Android mà điện thoại Google có thể khác hẳn với HTC, Samsung...

Với lập trình viên: cho phép lập trình viên có thể sử dụng các API ở tầng trên mà không cần phải hiểu rõ cấu trúc bên dưới, tạo điều kiện cho lập trình viên tự do sáng tạo bởi vì chỉ cần quan tâm đến nội dung mà ứng dụng làm việc. Một tập hợp API rất hữu ích được xây dựng sẵn như hệ thống định vị, các dịch vụ chạy nền, liên lạc giữa các ứng dụng, các thành phần cấp cao...

Một số thành phần quan trọng của tầng này:

- 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, layout, string. (Những thành phần không được viết bởi ngôn ngữ lập trình).
- Notification Manager: quản lý việc hiển thị các thông báo như báo có tin nhắn, có E-mail mới...)

1.4.4 Tầng ứng dụng

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, quản lý danh bạ, duyệt web, nhắn tin, lịch làm việc, đọc e-mail, bản đồ, quay phim chụp ảnh...

- Các ứng dụng được cài thêm như các phần mềm trò chơi, từ điển...
- Các chương trình có các đặc điểm:
- 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 dùng hoặc là một background, 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 chạy mà thôi. Điều đó có tác dụng hạn chế sự làm dụng tài nguyên, giúp hệ thống hoạt động tốt hơn.

Các ứng dụng được gán 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 trong việc sử dụng CPU.

Ứng dụng không có điểm vào cố định, không có phương thức main để bắt đầu.

CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO

2.1 Sơ lược về Android Studio

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio[3]. Đây là Môi trường phát triển tích hợp (IDE) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng có thể sử dụng hàng ngày.

Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác.

Android Studio là một hàm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

2.2 Cài đặt và sử dụng Android Studio

2.2.1 Cấu hình yêu cầu

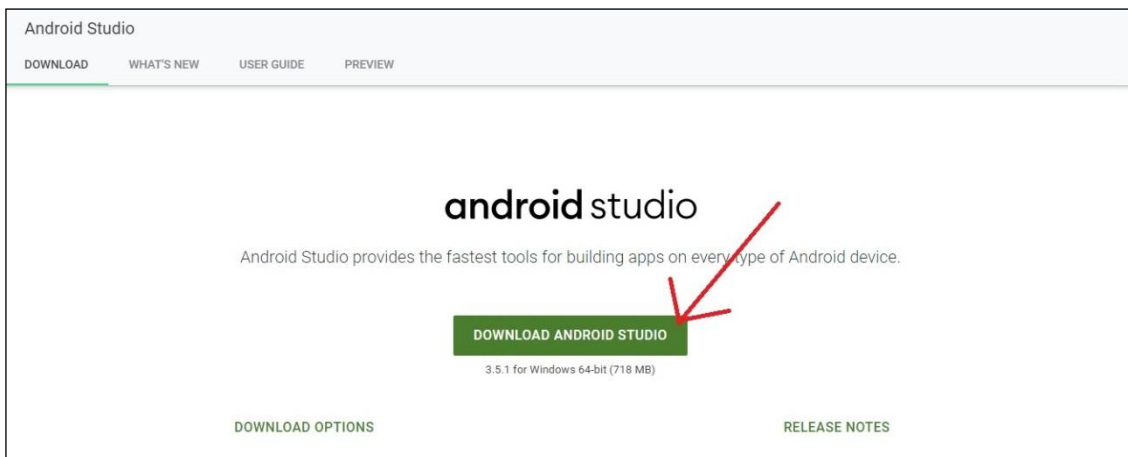
- Microsoft® Windows® 10/ 8.1/7/Vista (32 or 64-bit)
- Tối thiểu 4 GB RAM, cấu hình khuyến cáo: 8 GB RAM
- Ổ cứng trống ít nhất : 1 GB
- Độ phân giải màn hình tối thiểu HD: 1366 x 768
- Java Development Kit (JDK) 7 trở lên

- Lựa chọn thêm cho accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

Một điều cần chú ý là nên sử dụng CPU của Intel để lập trình với Android Studio vì khi giả lập điện thoại, CPU Intel ít bị giật hơn so với CPU Ryzen của AMD. Nhưng nếu ưa thích sử dụng chính máy android của mình để chạy thử nghiệm thì không thành vấn đề, hoàn toàn có thể sử dụng Ryzen.

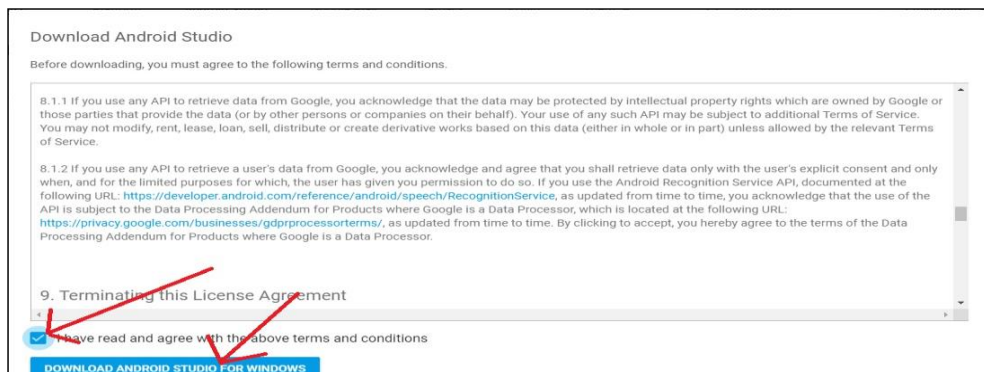
2.2.2 Phần mềm Android Studio

Vào trang <https://developer.android.com/studio/index.html> để tải phiên bản mới nhất:



Hình 2.2.1 Trang download Android Studio

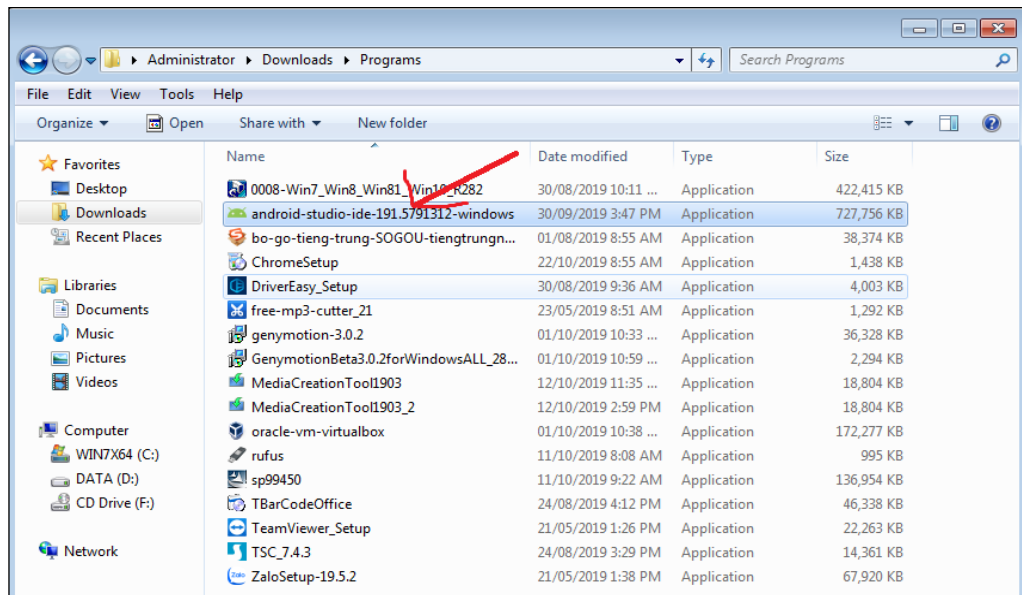
Khi nhấp vào nút Download Android Studio, cần đồng ý với các điều khoản và điều kiện sử dụng.



Hình 2.2.2: Điều khoản và điều kiện sử dụng

Sau khi đọc kỹ thì tích vào **Chấp nhận** và chọn **Download Android Studio For Window**.

Sau khi tải về máy tính xong, chạy file Setup



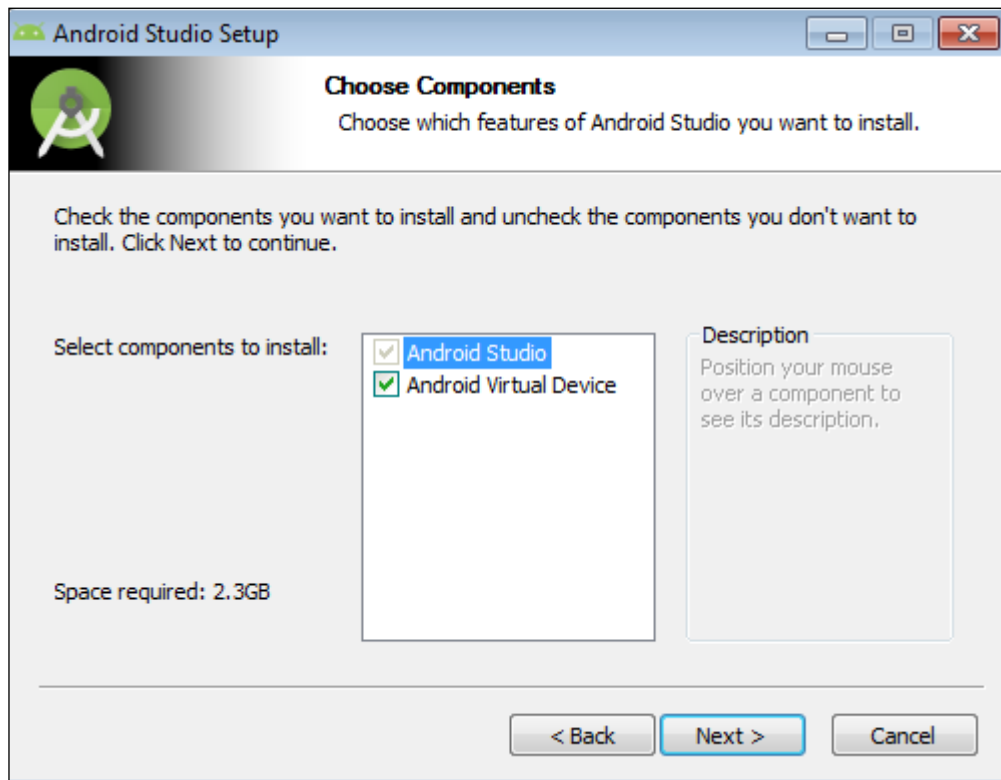
Hình 2.2.3: File setup Android

Bấm next để tiếp tục cài đặt



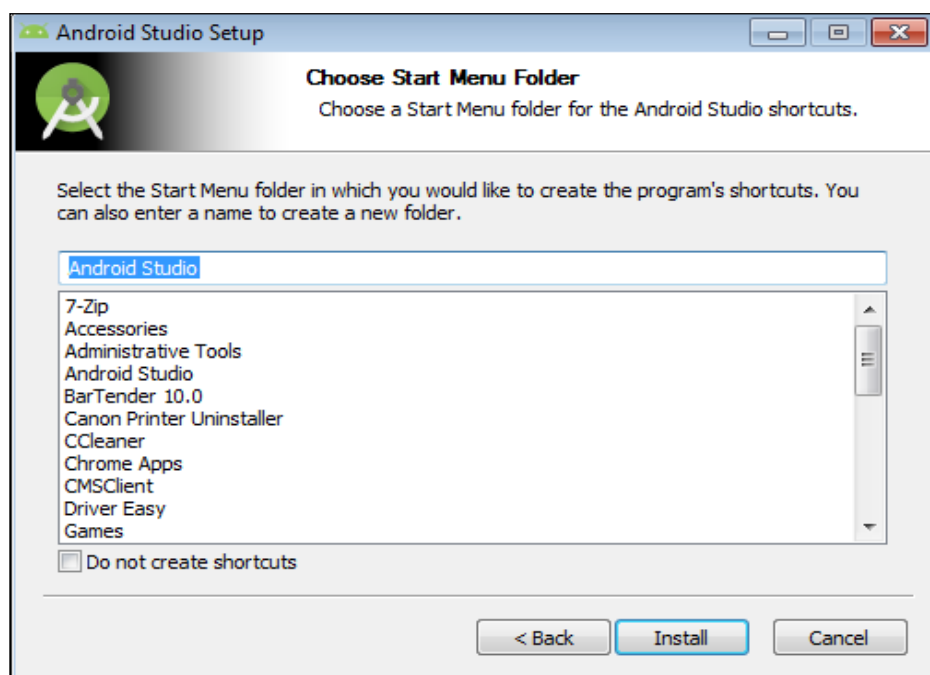
Hình 2.2.4: Giao diện cài đặt android studio

Bấm chọn cài đặt android virtual device để cài máy ảo android. Bấm next để tiếp tục cài đặt như hình bên dưới



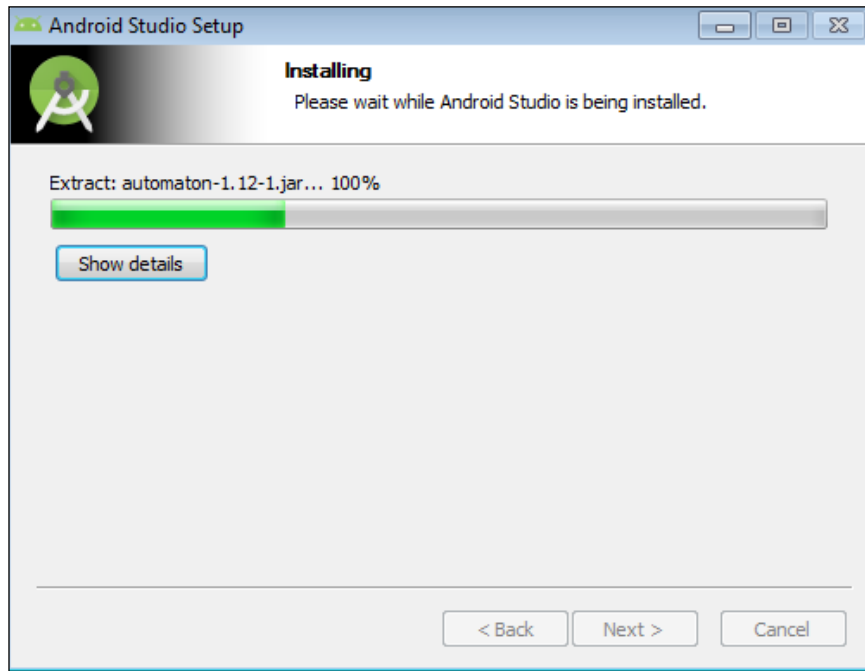
Hình 2.2.5: Giao diện cài đặt AVD

Chọn **Install** để cài đặt android studio.



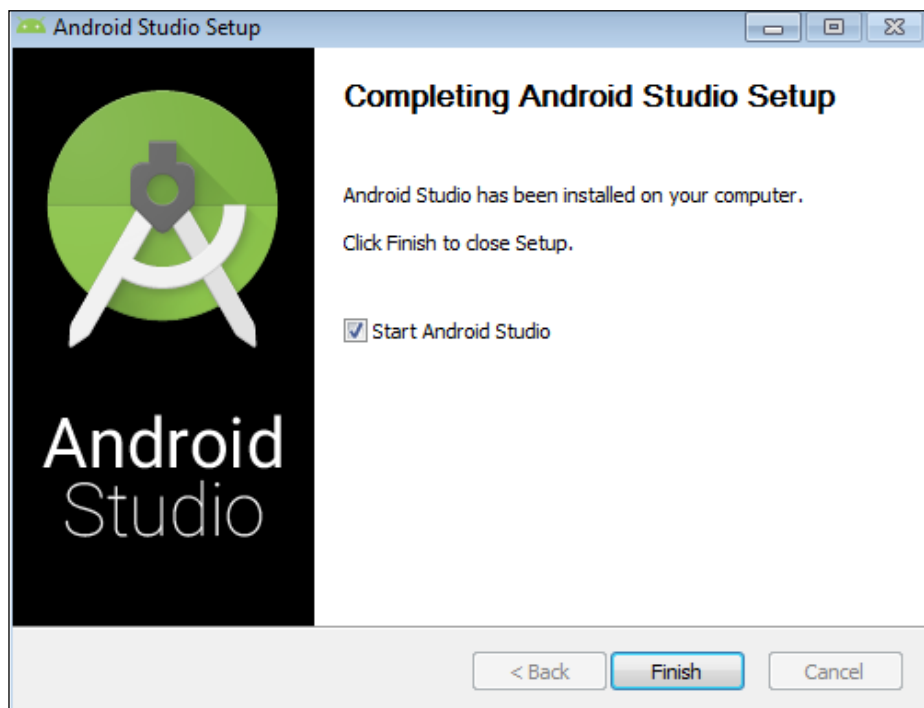
Hình 2.2.7: Bắt đầu cài đặt android studio

Chờ đợi quá trình cài đặt.



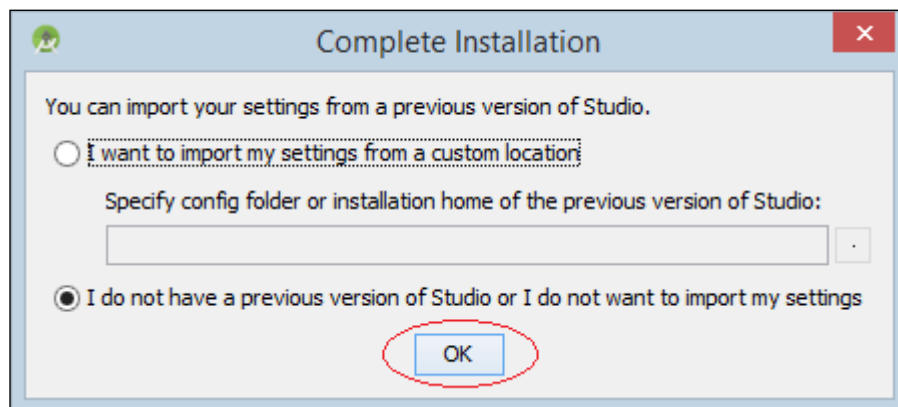
Hình 2.2.8: Quá trình cài đặt android studio

Sau khi quá trình cài đặt android studio hoàn tất, ấn Finish để chạy android studio.



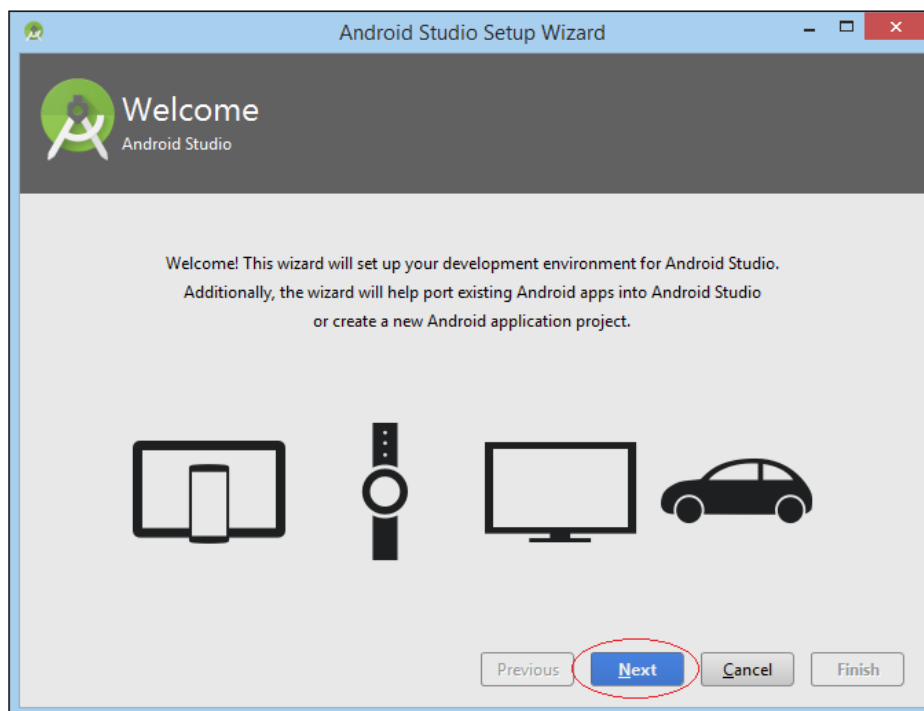
Hình 2.2.8: Cài đặt android studio hoàn thành

Lần đầu chạy Android studio, phần mềm sẽ yêu cầu tải thêm các gói hỗ trợ. Chọn "I Do not import ... settings".



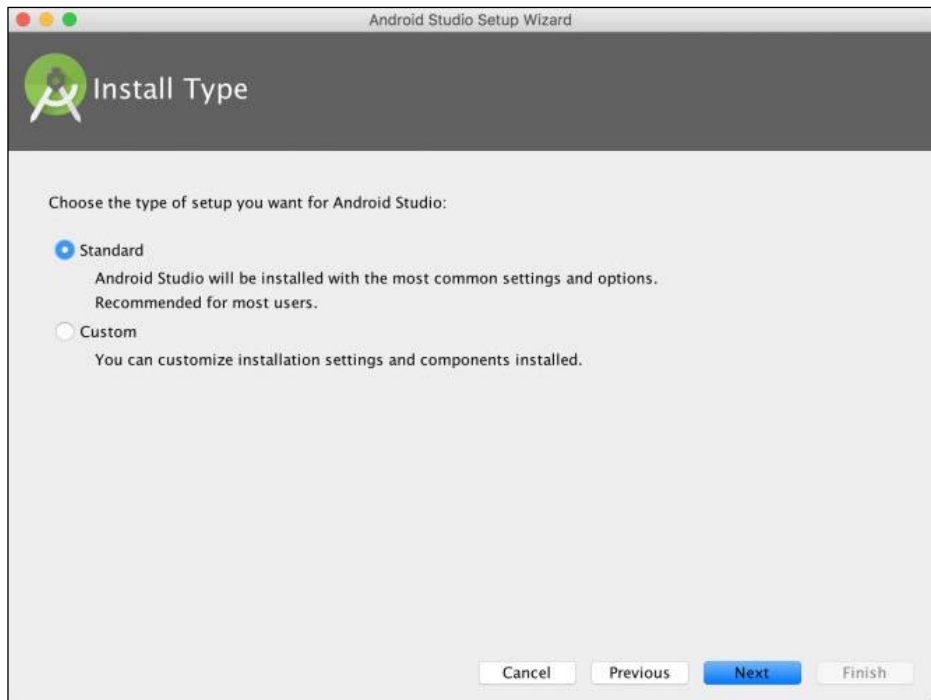
Hình 2.2.9: Phần thiết lập cài đặt cũ

Tại giao diện chào mừng của android studio để bắt đầu thiết lập cơ bản chọn Next.

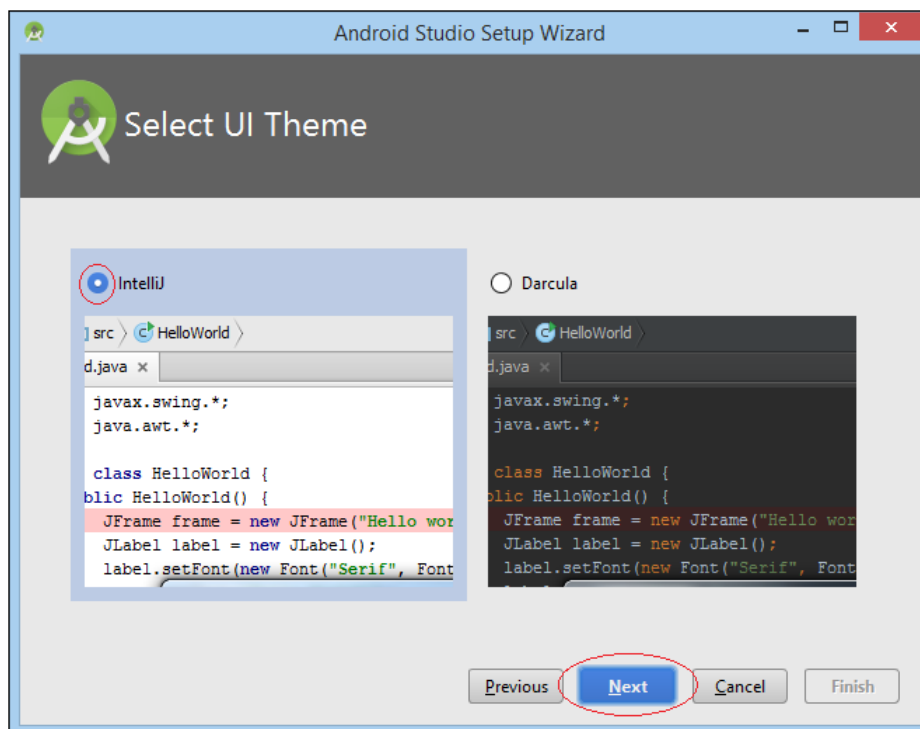


Hình 2.2.9: Giao diện chào mừng Android studio

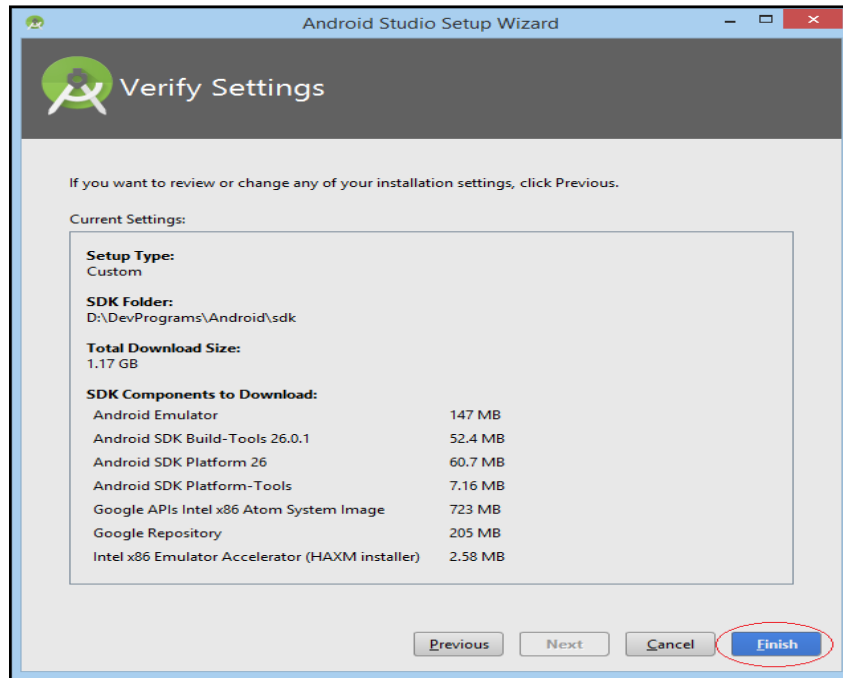
Cài đặt các thư viện hỗ trợ android studio. Chọn kiểu Standard sau đó bấm Next.



Hình 2.2.10: Màn hình chọn kiểu cài đặt cho android studio
 Chọn chủ đề màu sáng hoặc tối tùy thích sau đó bấm Next.



Hình 2.2.11: Màn hình chọn bộ màu chữ/nền cho giao diện người dùng
 Xác nhận lại các thông tin cài đặt như hình bên dưới, chọn Finish để bắt đầu tải và cài đặt hoàn tất các gói hỗ trợ cho android studio.

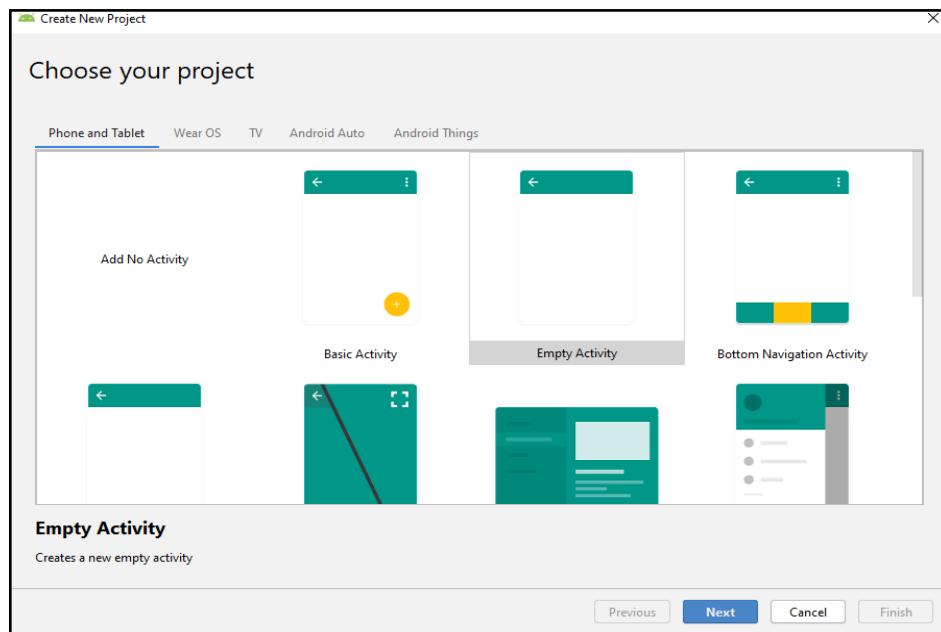


Hình 2.2.12: Màn hình cài đặt xác định các gói hỗ trợ

2.2.3 Dự án trong Android studio

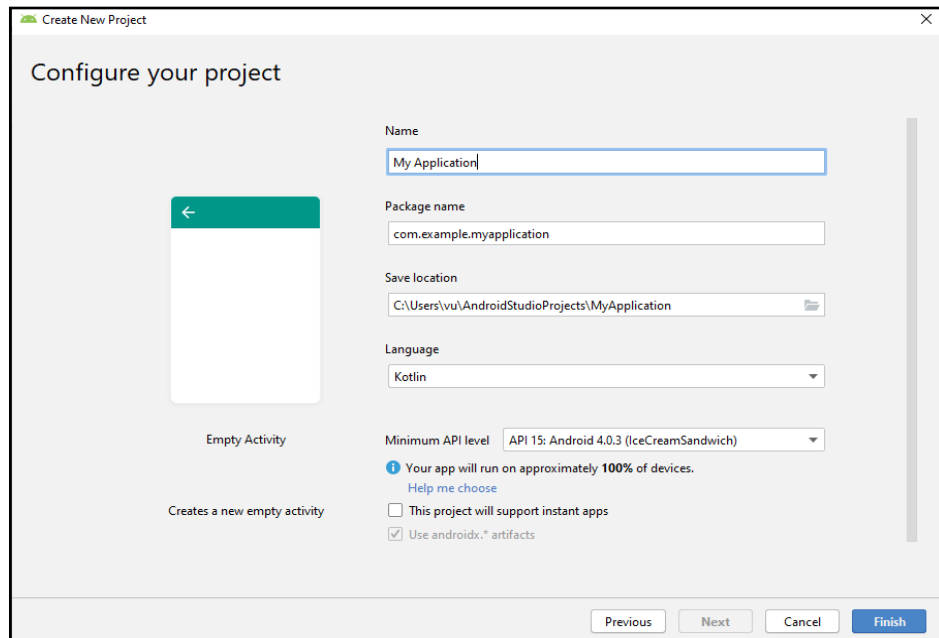
2.2.3.1 Tạo một dự án trên Android Studio

Khởi động Android Studio, chọn File → New → New Project . Chọn Activity, trong hình mình chọn Empty Activity (Việc lựa chọn Activity nào còn tùy thuộc vào mục đích phát triển ứng dụng) → sau đó chọn Next



Hình 2.2.14: Màn hình lựa chọn dự án

Bước tiếp theo có tên là Configure your project. Mục đích muốn người dùng khai báo một số thông tin về project. Người dùng cần điền thông tin như hình bên dưới.



Hình 2.2.15 Màn hình cấu hình dự án

Name: là tên của ứng dụng, tên này xuất hiện ở màn hình của thiết bị android khi người dùng cài đặt ứng dụng lên đó. Người dùng có thể nhìn vào hình nhỏ bên cạnh sẽ thấy tên các ứng dụng xuất hiện phía dưới icon của ứng dụng đó. Người dùng có thể viết hoa tên ứng dụng, hay để khoảng trắng tùy thích, nhưng nhớ là đừng quá dài hay quá ngắn, làm sao cho xúc tích và dễ nhớ.

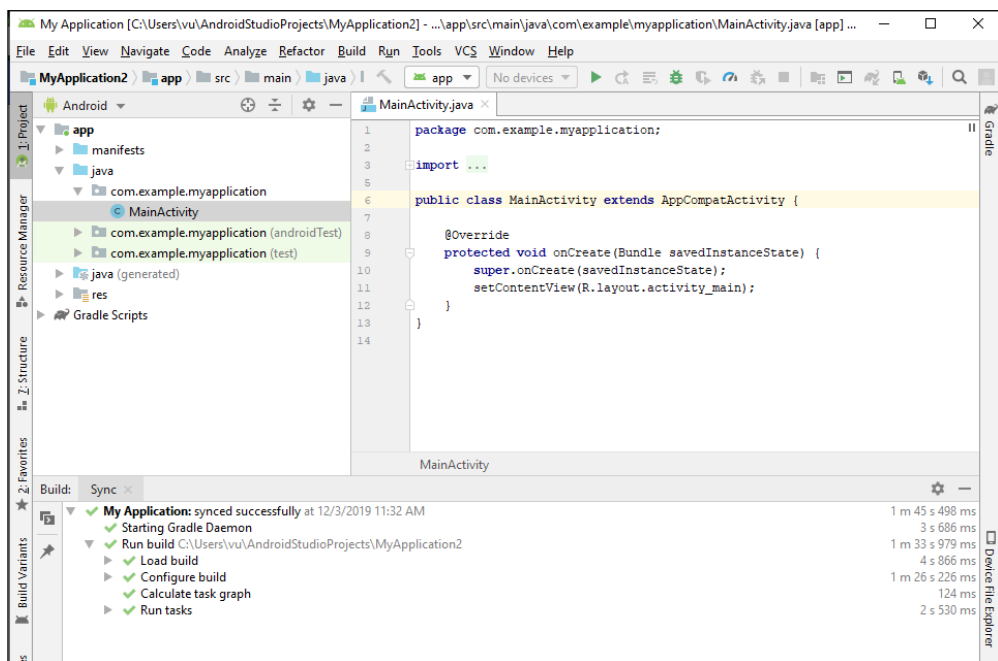
Package name: là tên package của ứng dụng . Ngoài ra thì với Android, package còn là định danh cho từng ứng dụng nữa. Package nên duy nhất và đặc thù nhất của một ứng dụng, sẽ không thể có hai ứng dụng với cùng một tên package được cài lên cùng một thiết bị. Thường thì người ta sẽ đảo ngược tên miền của công ty lại và thêm vào tên của project để tạo thành một package.

Save location: Là đường dẫn đến thư mục chứa project, người dùng có thể để mặc định hoặc tạo đường dẫn đến nơi tùy thích trong ổ cứng.

Language: ngôn ngữ dùng để viết ứng dụng. Có hai ngôn ngữ đó là java và Kotlin.

Minimum API Level: mục này báo cho hệ thống biết ứng dụng được tạo ra sẽ hỗ trợ ngược lại tối đa đến hệ điều hành cũ nhất nào. Nên nhớ là việc ứng dụng càng hỗ trợ hệ điều hành cũ hơn thì người dùng càng phải giải quyết các bài toán tương thích ngược hơn và do đó người sử dụng sẽ càng mất thời gian đầu đầu hơn trong việc phát triển các ứng dụng.

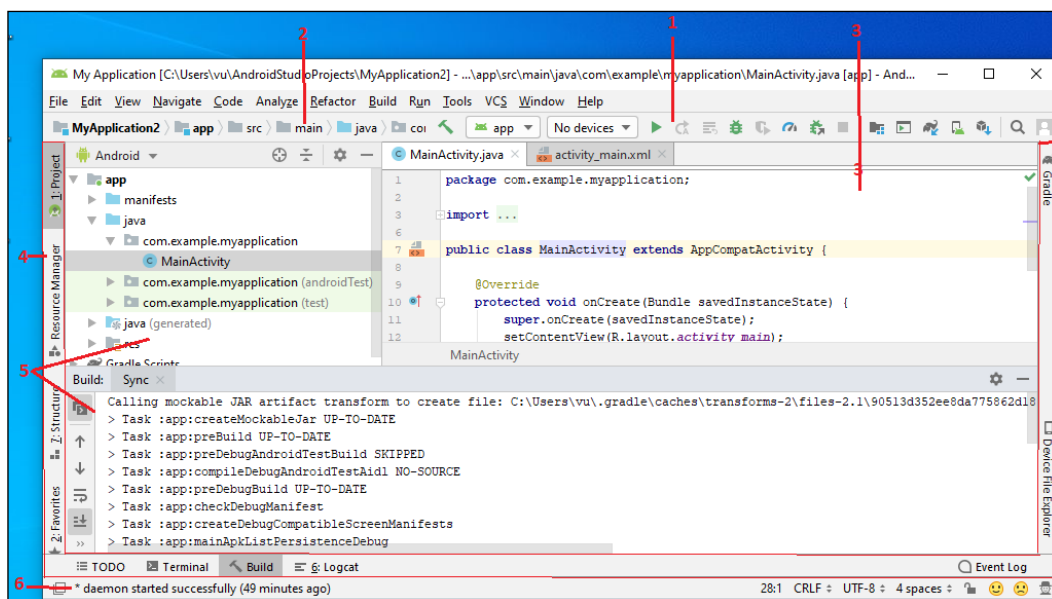
Sau khi tạo thành công project, sẽ được kết quả như hình bên dưới



Hình 2.2.16: Tạo dự án android được hoàn tất

2.2.3.2 Các thành phần của dự án trong Adroid Studio

Sau khi tạo xong một project mới, người dùng sẽ thấy giao diện chính của Android Studio, nhìn một cách tổng quan giao diện này được chia làm các phần chính sau.



Hình 2.2.17: Giao diện làm việc chung của Android Studio

Vị trí số **1 Toolbar** : thanh công cụ, nơi đây có được các nút điều khiển chính, chẳng hạn như các nút mở project, lưu project, cắt, dán dữ liệu,... Hoặc đặc thù hơn với lập trình có các nút khởi chạy ứng dụng, Debug ứng dụng,... Hoặc các quản lý cấp cap như các nút chạy chương trình quản lý Android SDK, chạy chương trình quản lý máy ảo...

Vị trí số **2 Navigation bar**: thanh điều hướng, giúp theo dõi file nào đang được mở, đường dẫn file đó trong project như thế nào.

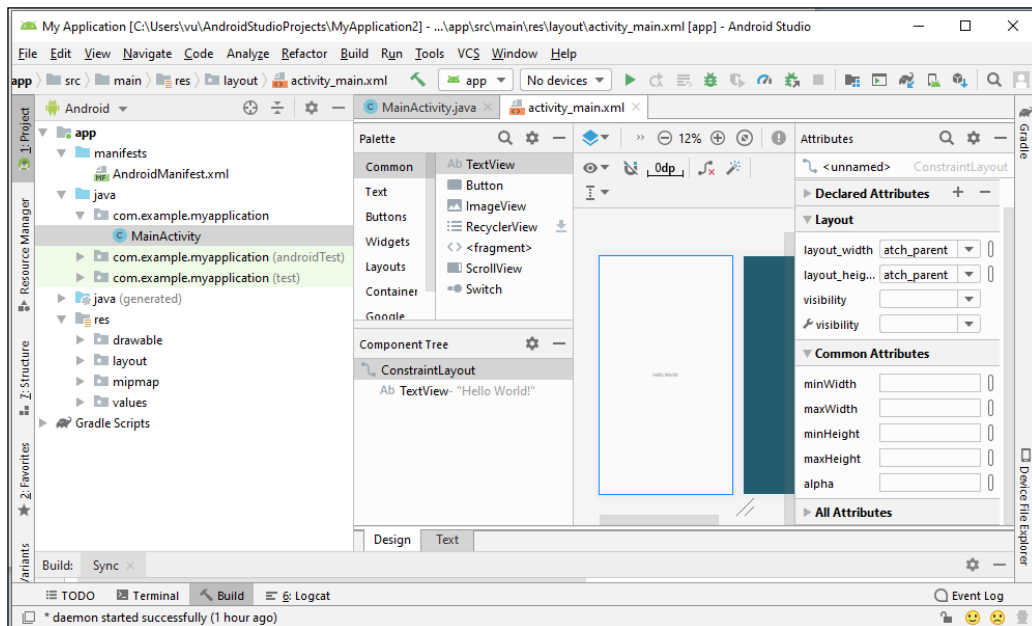
Vị trí số **3 Editor window**: cửa sổ soạn thảo, là nơi chỉnh sửa các dòng code. Đặc biệt hơn ở cửa sổ này đó là tùy vào loại source code, cửa sổ này sẽ xuất hiện khác nhau với từng loại để xem và chỉnh sửa source code dễ dàng. Chẳng hạn như khi người dùng mở một file java code, sẽ khác với mở một file xml, và khác với mở một file ảnh,...

Vị trí số **4 Tool window bar**: các điều khiển cho các công cụ khác. Các công cụ khác chính là công cụ can thiệp vào các công cụ quản lý của hệ thống. Chẳng hạn như quản lý log, quản lý quá trình debug, quản lý kết quả tìm kiếm, xem cây thư mục của project,... Tuy nhiên dàn nút trên đây chỉ là cho phép tắt mở các công cụ tương ứng mà thôi. Mỗi công cụ sẽ được mở ra ở dạng cửa sổ như mục 5.

Vị trí số 5 **Tool windows**: chính là các cửa sổ được điều khiển tắt mở từ thanh số 4 mà mình có nói đến ở đây.

Vị trí số 6 **Status bar**: thanh trạng thái, hiển thị trạng thái của project và của chính trình biên dịch Android Studio này. Người dùng sẽ thấy thông báo ứng dụng đang được thực thi, có thành công không, có lỗi gì không,...

Đây là hình ảnh project trên Android:



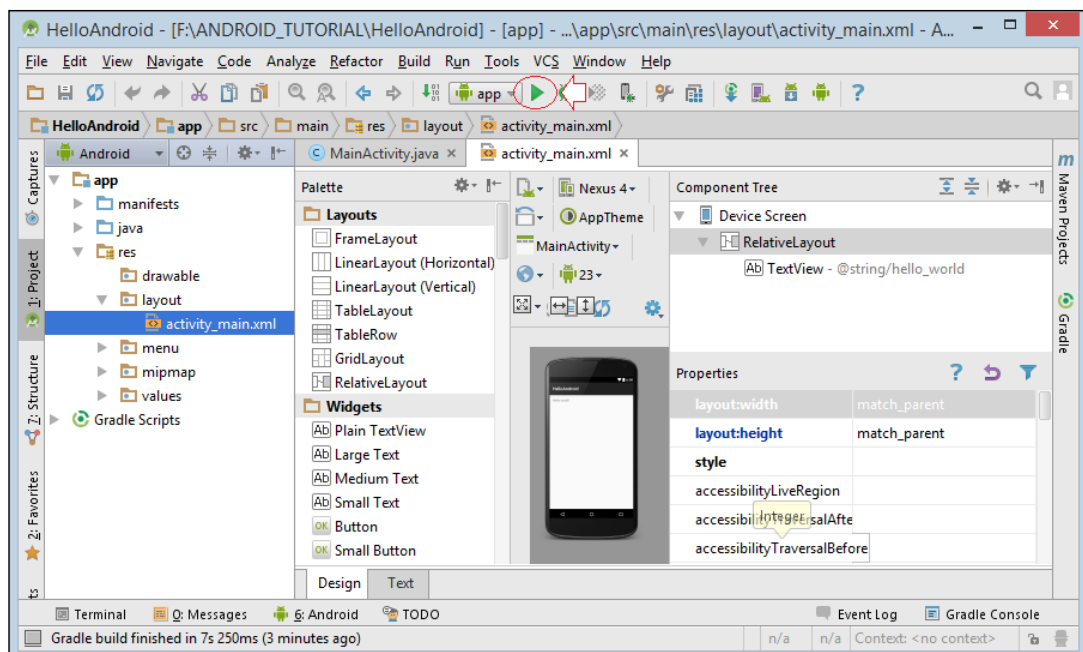
Hình 2.2.18: Project trong android

| File hoặc thư mục | Mô tả |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| manifest | Bên trong chứa AndroidManifest.xml đây là file mô tả các đặc điểm cơ bản của ứng dụng và xác định từng thành phần của nó. |
| java | Thư mục này chứa các file nguồn java cho dự án. Theo mặc định, nó bao gồm một tập tin nguồn MainActivity.java một lớp hoạt động (activity) chạy khi ứng dụng được khởi động. |
| Java(generated) | Bên trong chứa các tập BuildConfig |
| res/drawable | Các phiên bản Android trước đây sử dụng thư |

| | |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| | mục này chứa ảnh, các phiên bản hiện tại sử dụng thư mục mipmap thay thế làm nơi chứa ảnh. Thư mục này gần như không còn dùng. |
| res/layout | Thư mục này chứa các file định nghĩa giao diện người dùng |
| res/mipmap | Chứa các ảnh |
| res/values | Đây là một thư mục cho các tập tin XML khác nhau có chứa một tập hợp các nguồn, chẳng hạn các chuỗi (String) và các định nghĩa màu sắc. |

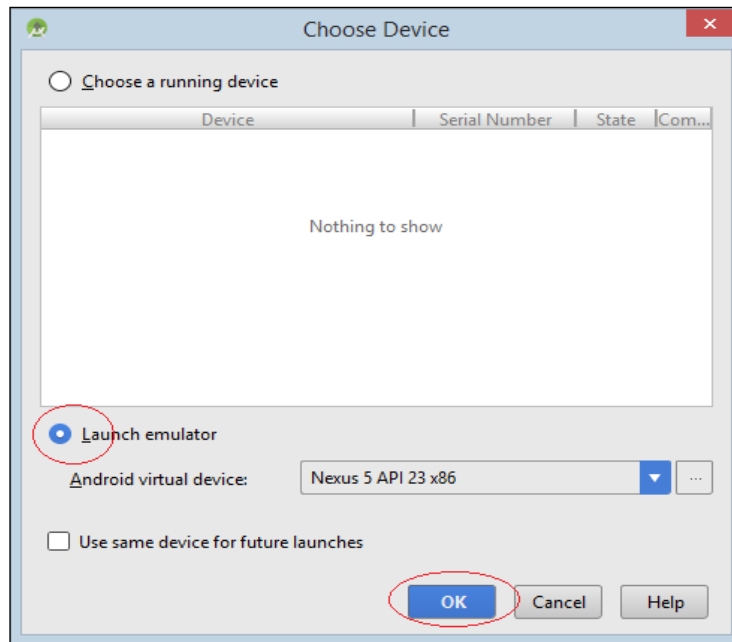
2.2.3.3 Biên dịch và chạy chương trình

Sau khi chương trình được viết hoàn thiện hoặc lập trình viên muốn xem chương trình hoạt động như thế nào thì tiến hành chạy chương trình.



Hình 2.2.19: Bắt đầu chạy chương trình.

Tiếp theo sẽ xuất hiện một màn hình chọn máy ảo cho để chạy chương trình.



Hình 2.2.20: Màn hình chọn máy ảo

Màn hình mô phỏng đã hiển thị chiếc điện thoại và dòng chữ “Hello World”.



Hình 2.2.21: Màn hình chạy mô phỏng chương trình

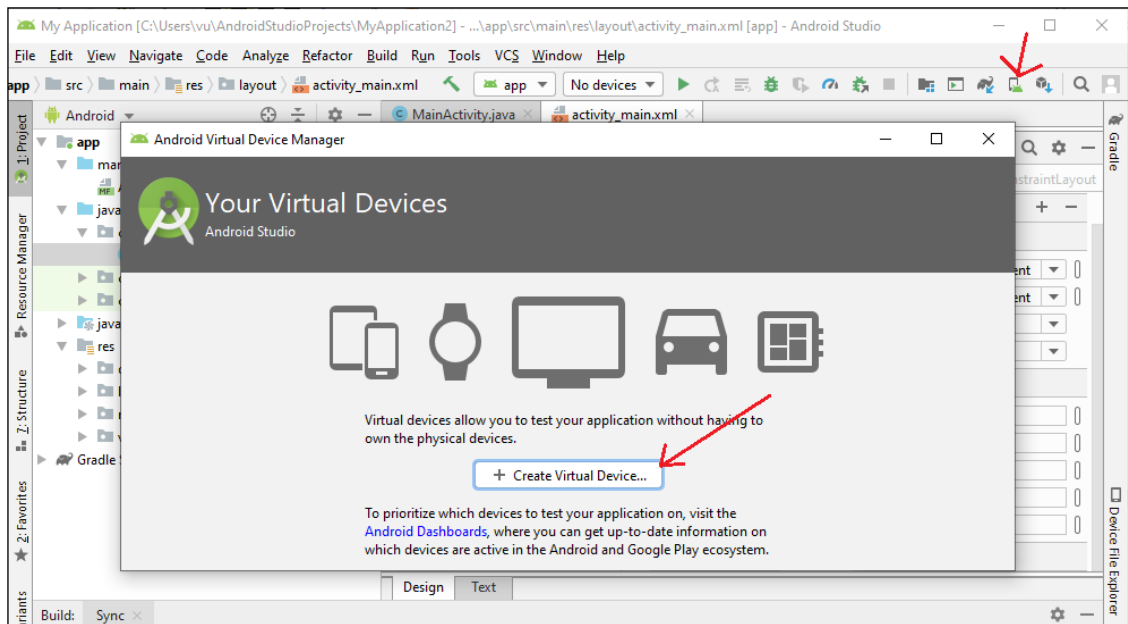
2.3 Thiết bị ảo trong Android Studio

Kiểm thử chương trình là một khâu không thể thiếu trong quá trình phát triển ứng dụng. Sản phẩm của phần mềm cho TBDD chạy trên điện thoại hoặc máy tính bảng, do đó cần có các thiết bị tương ứng để chạy thử chương

trình. Việc sử dụng các thiết bị vật lý có nhược điểm là khó thay đổi phiên bản của HĐH android. Ngoài giải pháp sử dụng thiết bị vật lý để chạy thử CT, lập trình viên còn có thể sử dụng các thiết bị ảo để chạy thử ứng dụng

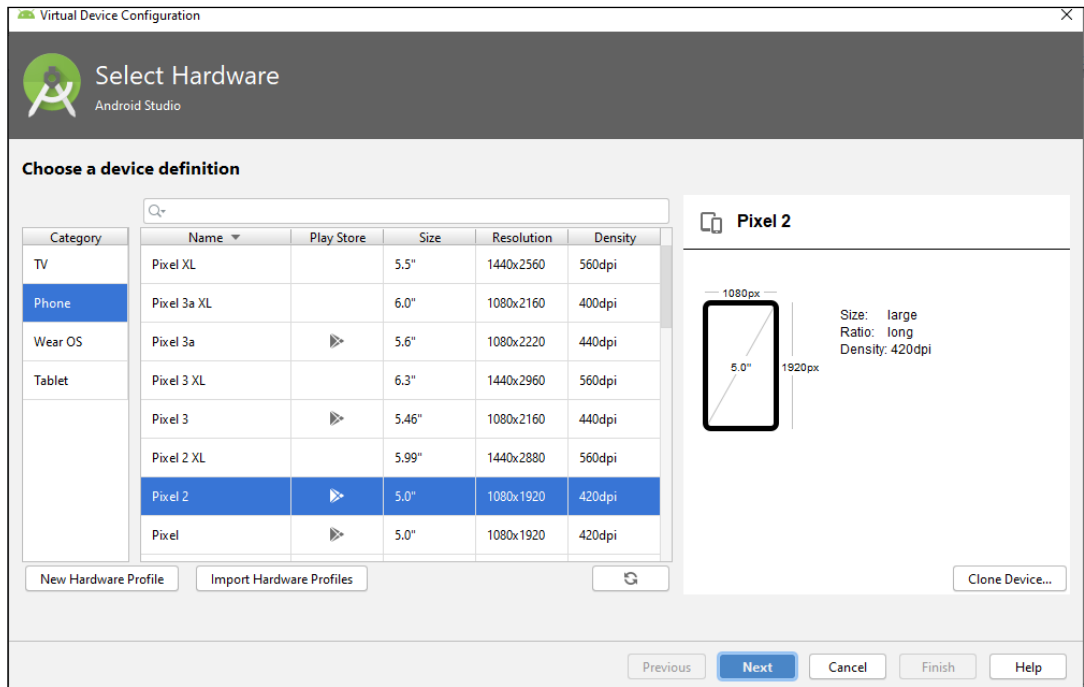
Cách 1 cài đặt thiết bị ảo trong android studio

Để tạo một máy ảo trong giao diện android studio, bấm AVD Manager sau đó chọn Create Virtual Device.



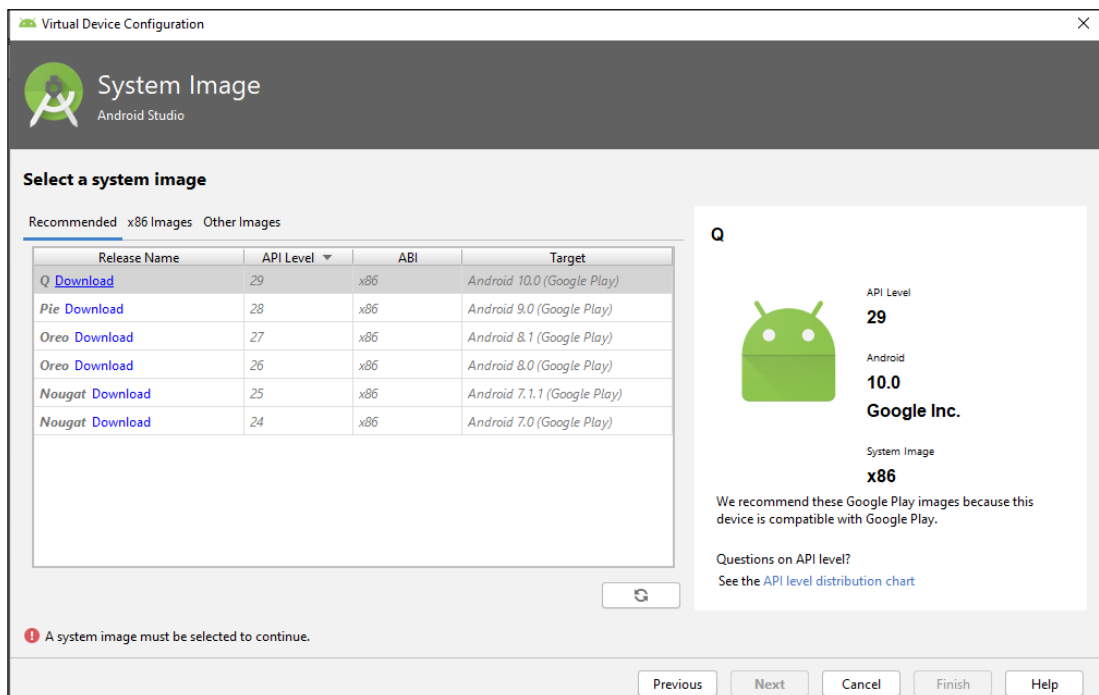
Hình 2.3.1: Tạo một máy ảo android

Chọn Phone và chọn thiết bị có màn hình phù hợp sau đó chọn next.



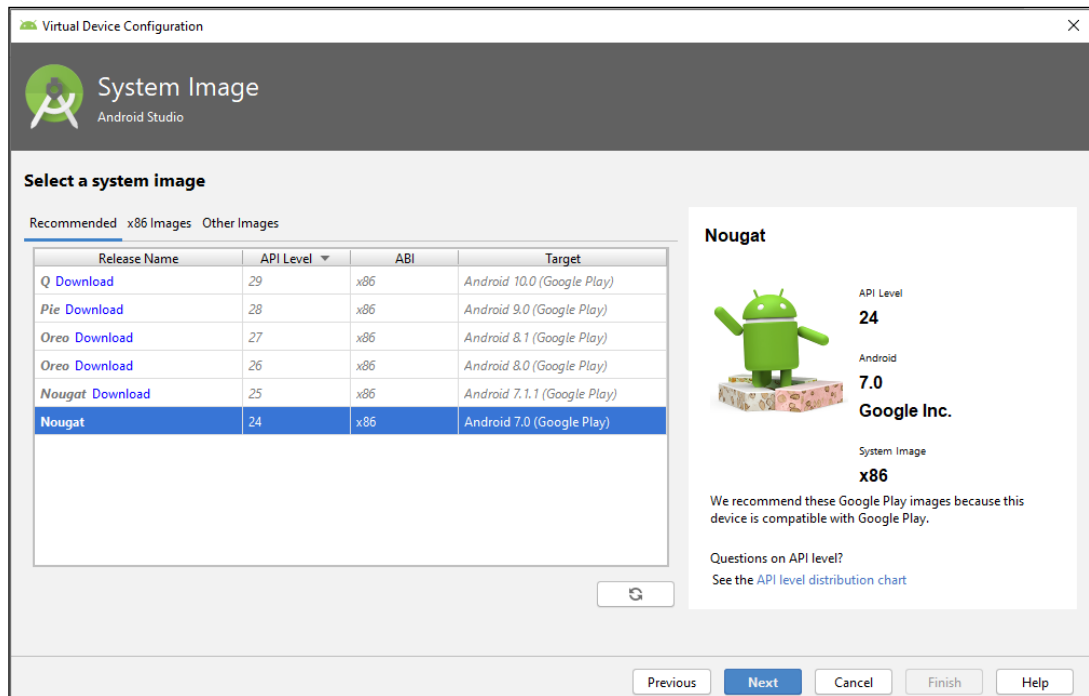
Hình 2.3.2: Các máy ảo android

Chọn và tải xuống phiên bản android thích hợp cho máy ảo android. Chọn download bên cạnh bên cạnh phiên bản android phù hợp và chờ quá trình tải xuống hoàn tất.

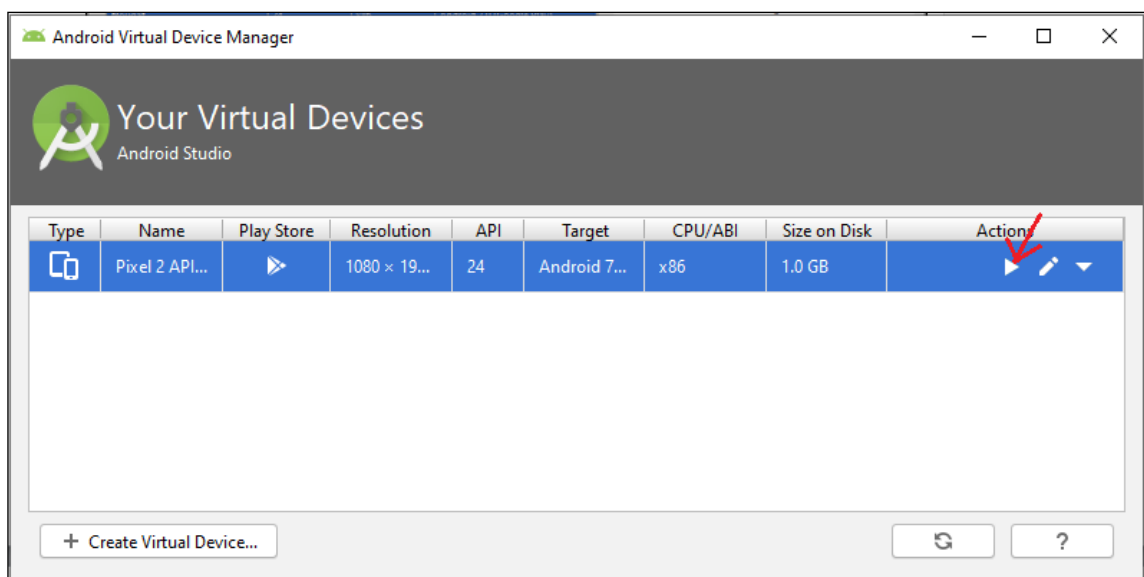


Hình 2.3.3: Các phiên bản android

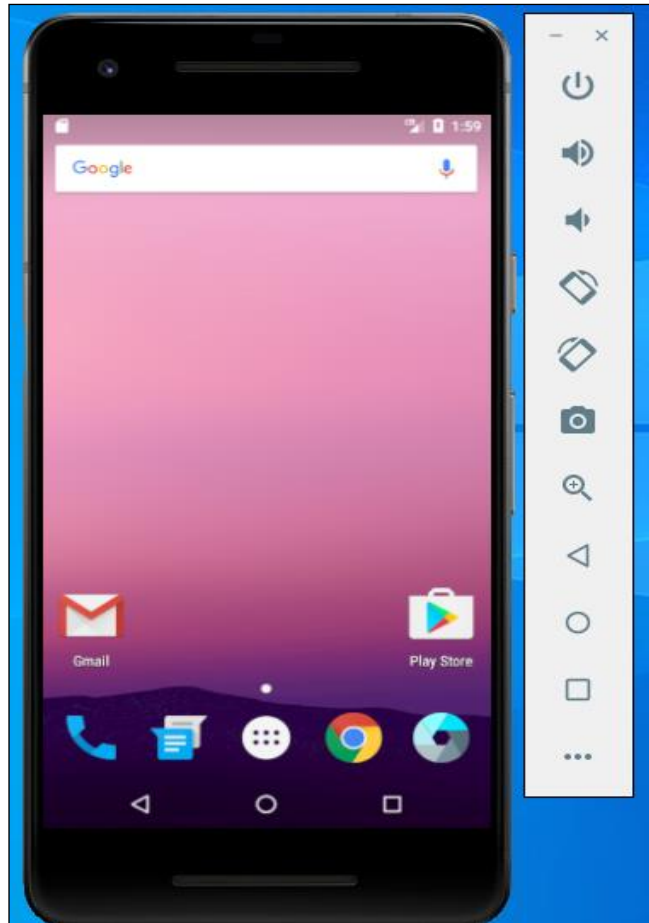
Chọn phiên bản android mình đã tải xuống. Bấm Next để tiếp tục.



Hình 2.3.4: Phiên bản android sau khi tải xuống
Sau khi máy ảo được tạo ra như hình 2.3.5 bấm vào tên máy ảo sau đó bấm vào hình tam giác bên cạnh để khởi động máy ảo.



Hình 2.3.5: Chạy máy ảo
Giao diện thiết bị ảo đã được khởi tạo như hình bên dưới.



Hình 2.3.6: Giao diện máy ảo

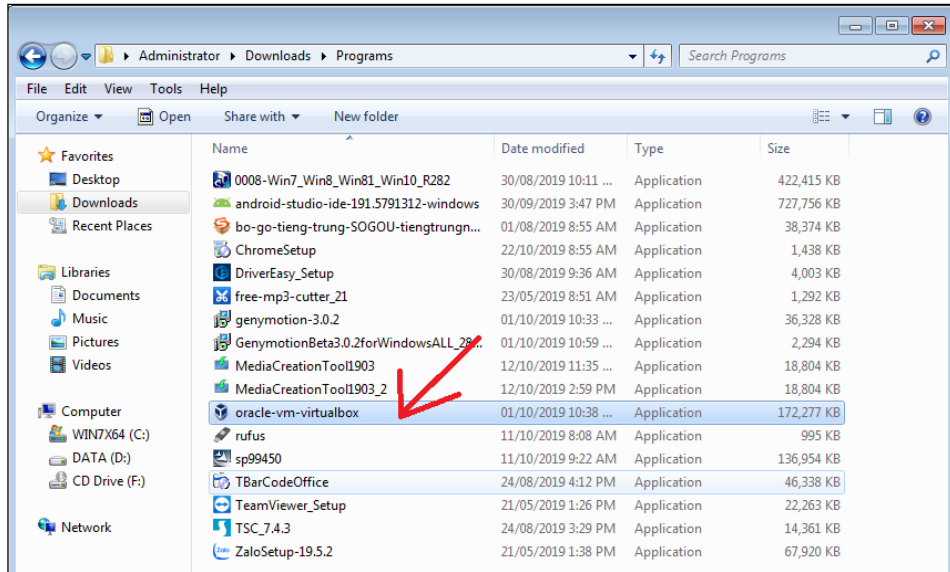
Cách 2 cài đặt máy ảo Genymotion

Trong một số trường hợp có một số máy không hỗ trợ chạy AVD Manager thì có thể dùng một máy ảo khác để chạy thay cho AVD Manager.

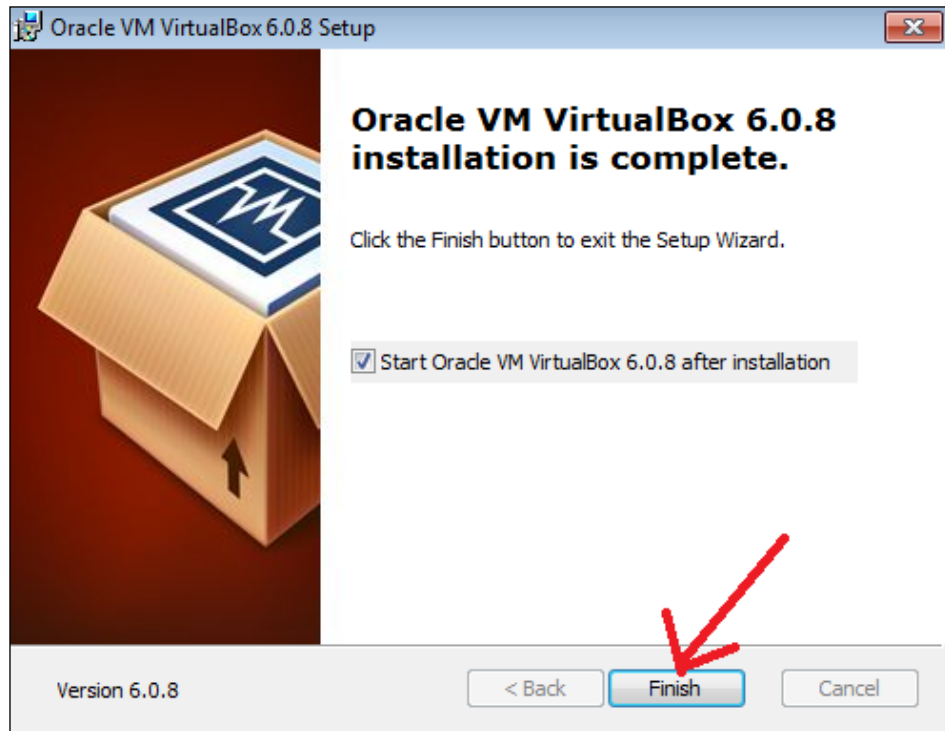
Genymotion là phần mềm giả lập Android, giúp người dùng có thể chạy ứng dụng Android, Game Android trên máy tính của mình[4]. Genymotion hiện nay sở hữu rất nhiều tính năng thông minh hỗ trợ 40 thiết bị ảo tới từ nhiều hãng điện thoại khác nhau như điện thoại Sony, Samsung... tích hợp đầy đủ phiên bản của hệ điều hành từ 4.1 đến 9.0, tốc độ xử lý nhanh nên rất được nhiều người lựa chọn sử dụng.

Khi cài đặt Genymotion thường thì nó đi kèm theo là Virtualbox ở đây mình tải hai file riêng biệt nên cần cài virtualbox

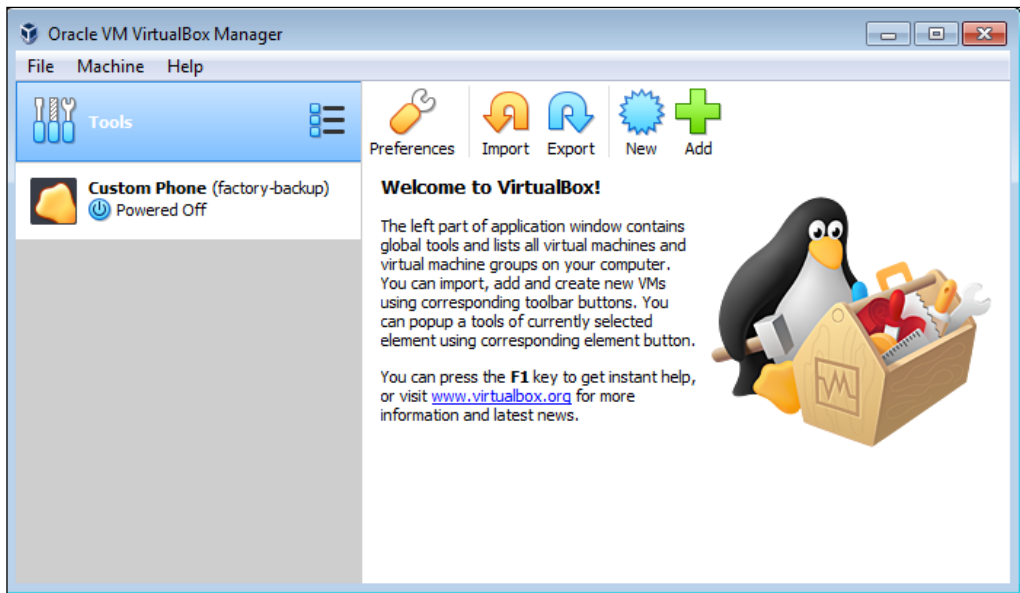
Bước 1: chạy file cài đặt.



Bước 2: nhấn Next với Install tới khi Finish cài đặt kết thúc.



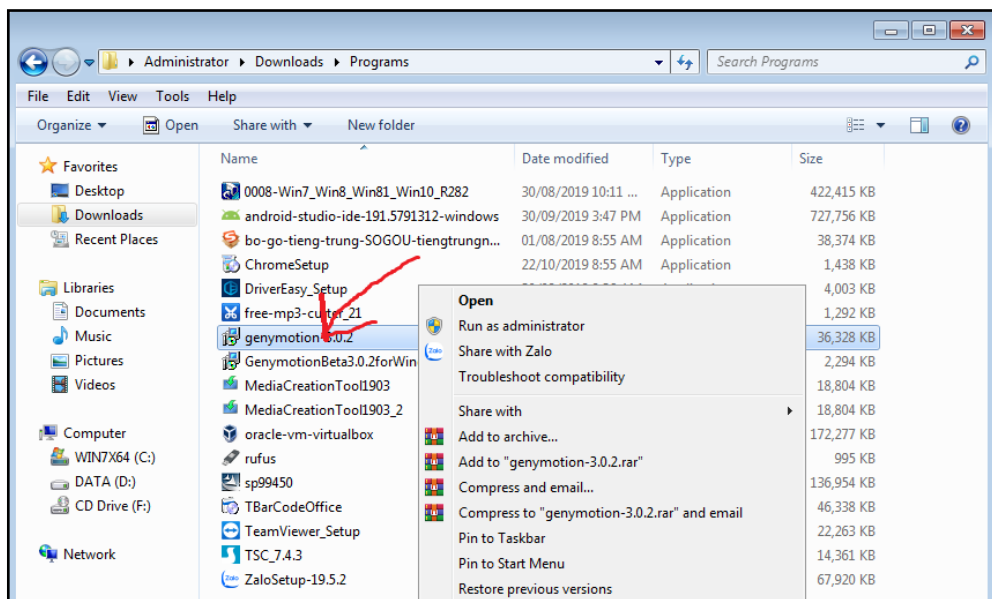
Sau khi cài đặt hoàn tất chúng ta được giao diện như hình bên dưới.



Để cài đặt máy ảo truy cập vào đường link sau để tải về:

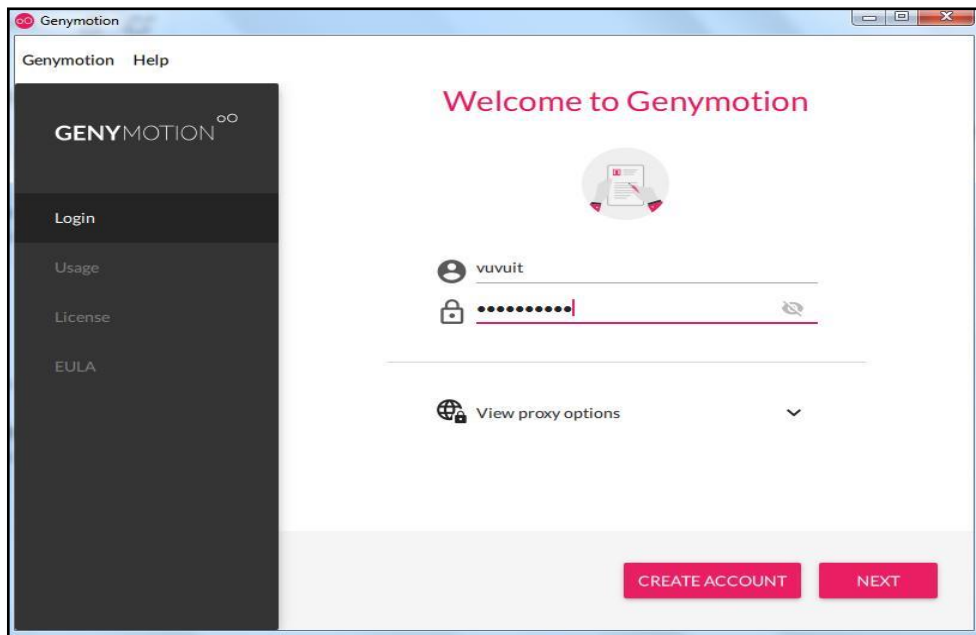
<https://www.genymotion.com/download/>

Sau khi tải về bắt đầu cài đặt quá trình cài đặt chỉ cần ấn Next, Install và finish để kết thúc quá trình cài đặt không có gì đặc biệt.



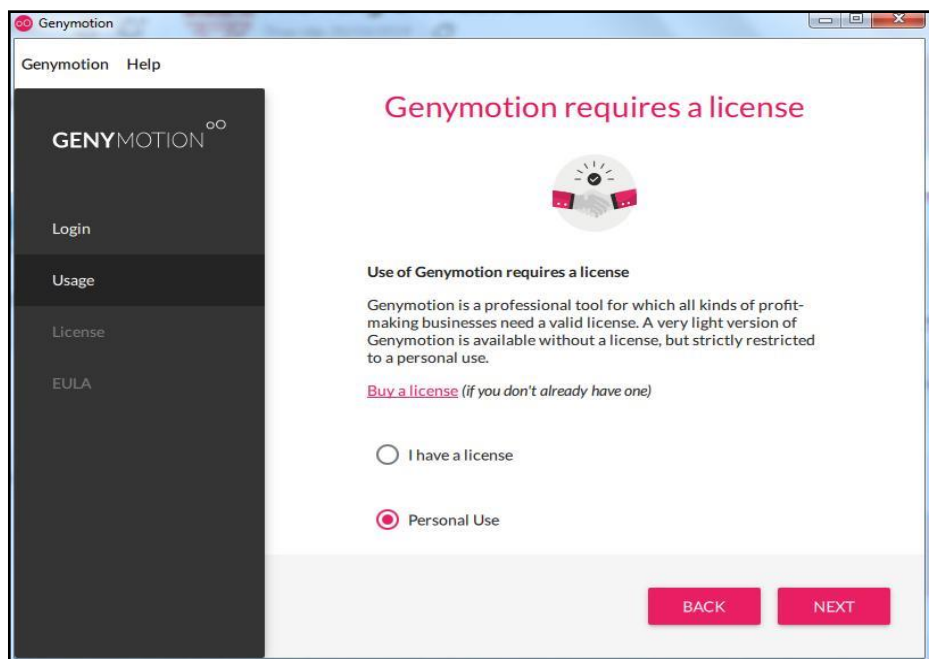
Hình 2.3.7: File cài đặt Genymotion

Sau khi cài đặt hoàn tất màn hình sẽ hiển thị trang đăng nhập để đăng nhập vào genymotion. Sau khi điền đầy đủ thông tin đăng nhập chọn NEXT.



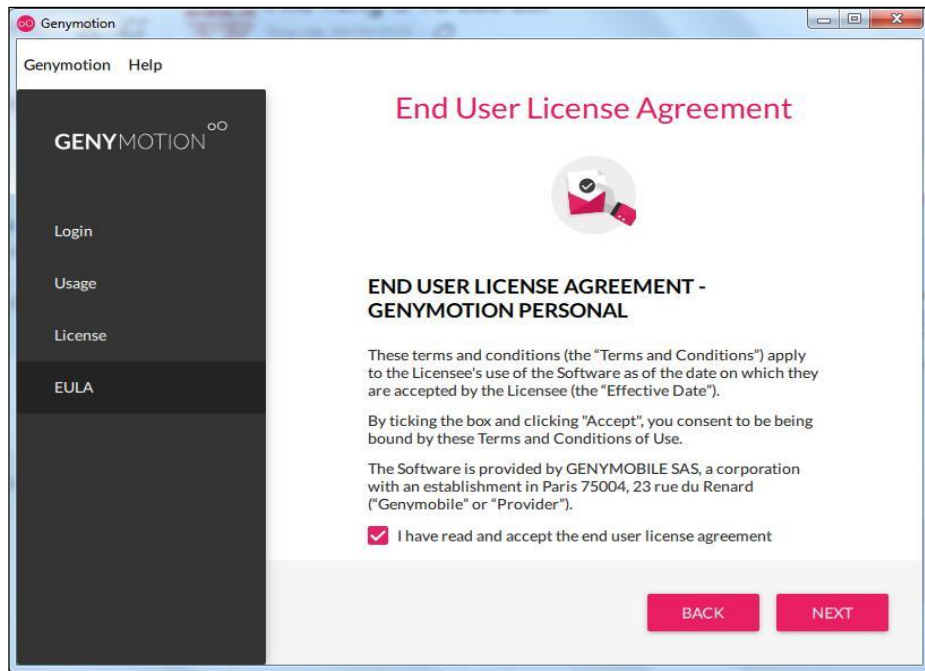
Hình 2.3.8 Màn hình đăng nhập của Genymotion

Tiếp theo Genymotion cần một giấy phép chúng ta sử dụng cá nhân nên chọn Personal User sau đó chọn Next.



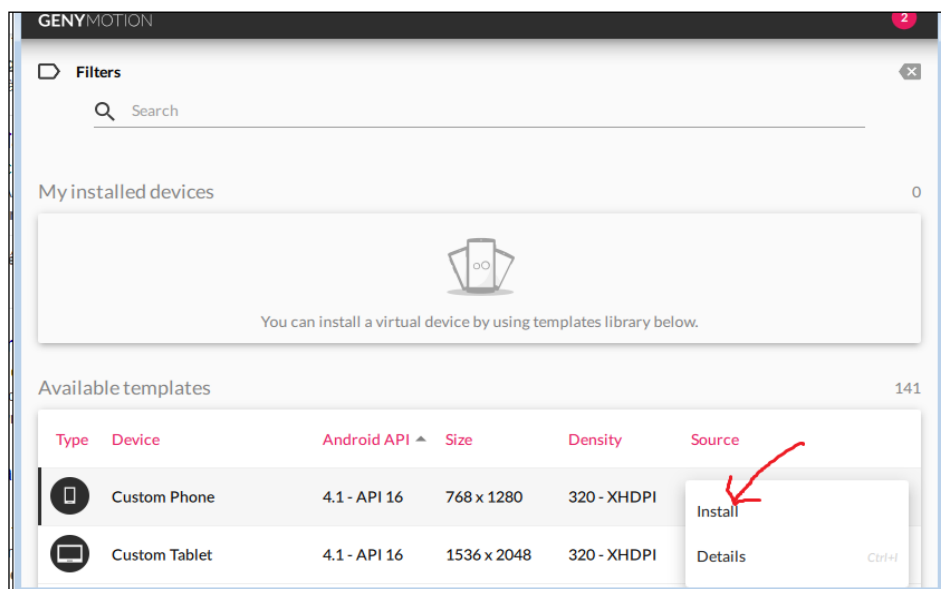
Hình 2.3.9: Genymotion cần có giấy phép

Tiếp theo Genymotion có một số điều khoản tích vào đồng ý các điều khoản sau đó chọn Next.



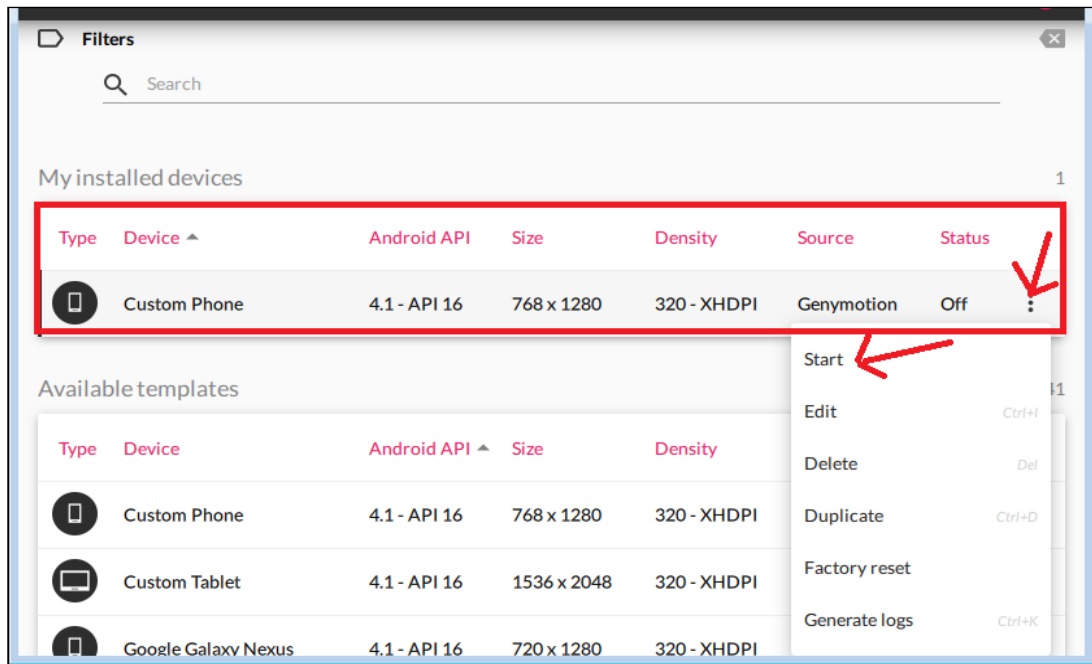
Hình 2.3.10: Thỏa thuận cấp phép người dùng

Sau khi hoàn tất các bước bên trên chúng ta sẽ vào giao diện chính của genymotion, trong giao diện chính sẽ có rất nhiều máy ảo để lựa chọn. Chỉ cần chuột phải vào máy ảo muốn cài và ấn install, sau đó hệ thống sẽ tải máy ảo xuống.

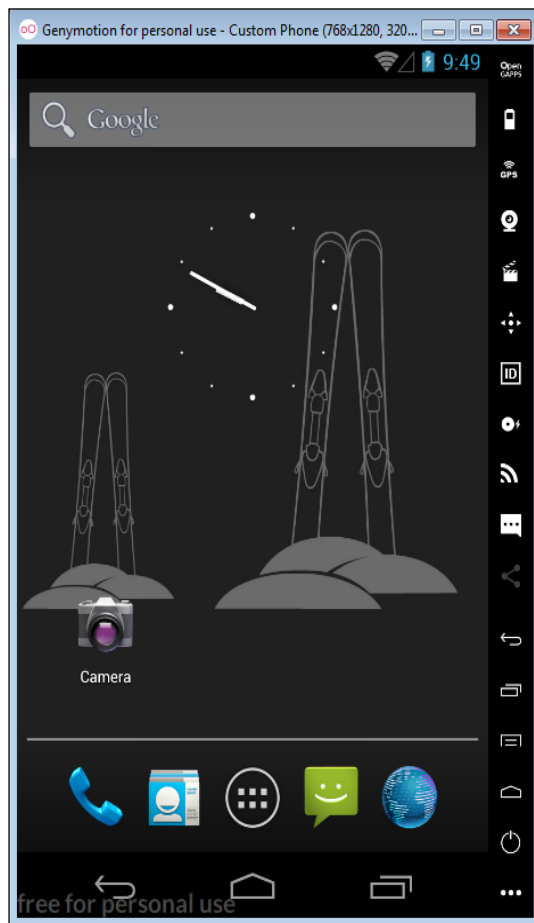


Hình 2.3.11: Giao diện chính của genymotion

Sau khi tải được máy ảo mình muốn ấn chuột trái vào phần ba chấm có mũi tên đỏ, sau đó chọn Start để bắt đầu khởi động thiết bị.



Hình 2.3.12: Màn hình khởi động thiết bị ảo
Sau khi khởi động thành công sẽ được máy ảo như hình bên dưới.



Hình 2.3.13: Máy ảo đã khởi động thành công

CHƯƠNG 3: Lập trình Chat với Firebase

3.1 Tổng quan về Firebase

3.1.1 Khái niệm

Có nhiều cách hiểu về Firebase khi đứng trên các quan điểm khác nhau, sau đây là 2 khái niệm cơ bản[5].

Firebase là một nền tảng di động giúp người dùng nhanh chóng phát triển các ứng dụng chất lượng cao, phát triển ứng dụng cho người dùng quy mô lớn và kiếm được nhiều tiền hơn.

Firebase là một dịch vụ hệ thống 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 mình đang phát triển. Hỗ trợ cả 2 nền tảng Android và IOS, Firebase mạnh mẽ, đa năng, bảo mật và là dịch vụ cần thiết đầu tiên để xây dựng ứng dụng với hàng triệu người sử dụng.

3.1.2 Lịch sử phát triển của Firebase

Firebase có tiền thân là Envolv. Đây là dịch vụ cung cấp những API để người dùng dễ dàng tích hợp tính năng chat vào trang web. Điều thú vị là người dùng Envolv sử dụng dịch vụ để truyền dữ liệu chứ không đơn thuần cho ứng dụng chat. Họ sử dụng Envolv để đồng bộ dữ liệu của những ứng dụng như game online, danh bạ, lịch... Nhận biết được điều này, CEO của Envolv đã tách biệt hệ thống chat và đồng bộ dữ liệu thời gian thực thành 2 mảng riêng biệt. Đến năm 2012, Firebase được thành lập như một công ty cung cấp dịch vụ Backend-as-a-Service theo thời gian thực. Người thấy mùi tiềm năng, năm 2014, Google lập tức mua lại Firebase với giá không được tiết lộ. Và giờ Google phát triển Firebase thành một dịch vụ đồ sộ.

3.1.3 Ưu nhược điểm của Firebase

3.1.3.1 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 cố 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.

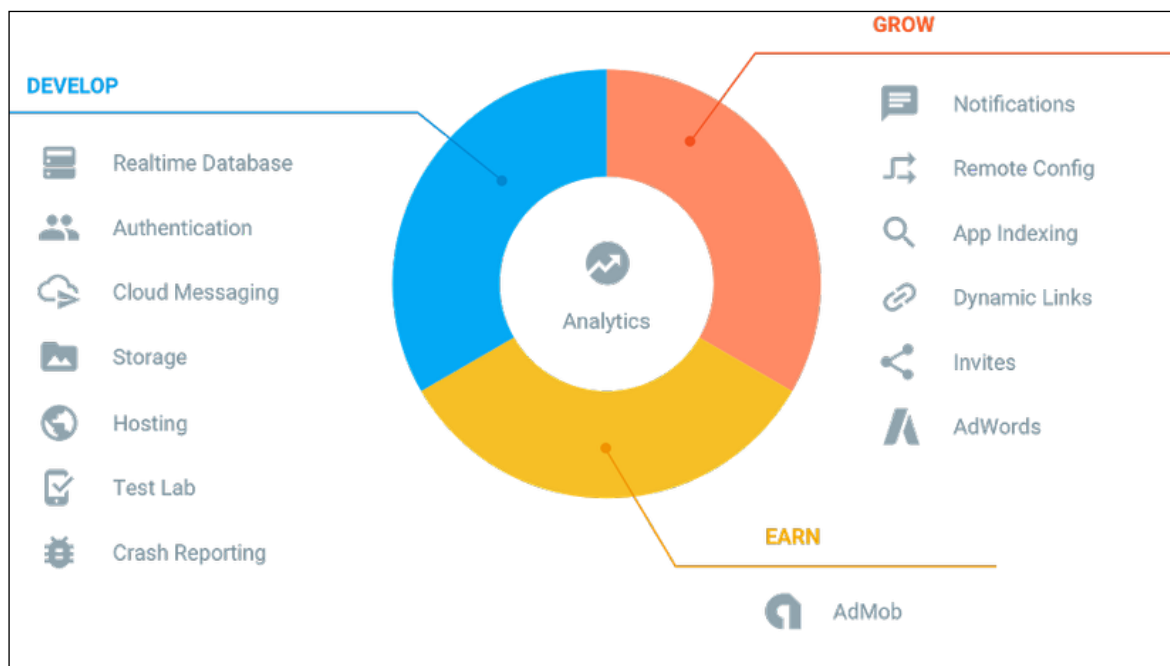
3.1.3.2 Nhược điểm

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 .

3.2 Dịch vụ Firebase

Firestore cung cấp cho người dùng công cụ Firebase Analytics và 2 nhóm sản phẩm chính tập trung vào 2 đối tượng là:

- Develop & test your app: phát triển và kiểm thử các ứng dụng được thiết kế.
- Grow & engage your audience: Phân tích dữ liệu và tối ưu hóa trải nghiệm đối với người dùng.



Hình 3.1.1: Các dịch vụ của Firebase

3.2.1 Dịch vụ Firebase Analytics

Là một giải pháp miễn phí và phân tích không giới hạn. Quản lý hành vi người dùng và cá biệt pháp từ một bảng điều khiển duy nhất. Phân tích 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.

3.2.2 Các dịch vụ phát triển và kiểm thử ứng dụng

Realtime Database: Lưu trữ và đồng bộ dữ liệu người dùng thời gian thực, các ứng dụng hỗ trợ tính năng này có thể lưu trữ dữ liệu máy chủ trong tích tắc. 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, dữ liệu được ghi và đọc với thời gian thấp nhất tính bằng mili giây. Nền tảng này hỗ trợ đồng bộ hóa dữ liệu của người dùng kể cả khi không có kết nối mạng, tạo lên trải nghiệm xuyên suốt bất chấp tình trạng kết nối internet của người sử dụng. Realtime 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. Cách trình bày hợp lý với từng chu trình hoạt động đến xảy ra lỗi, Các báo cáo trực quan giúp người phát triển có thể nắm bắt và xử lý kịp thời các lỗi chính của ứng dụng.

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. Xây dựng giao diện riêng hoặc tận dụng lợi thế của mã nguồn mở, giao diện người dùng tùy biến hoàn toàn.

Cloud Functions: Mở rộng ứng dụng bằng mx phụ trợ tùy chỉnh mà không cần quản lý và quy mô các máy chủ của riêng bạn. 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 webhooks.

Cloud Storage: Lưu trữ và chia sẻ nội dung do người dùng tạo ra như hình ảnh, âm thanh và video với bộ 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 năng bảo mật của Google để tải lên và tải tệp cho các ứng dụng Firebase của người dùng, bất kể chất lượng mạng.

Hosting: Đơn giản hóa lưu trữ web của người dùng với các công cụ được thực hiện cụ thể cho các ứng dụng web hiện đại. Khi tải lên nội dung web, hệ

thông sẽ tự động đẩy chúng đến CDN toàn cầu của hệ thống và cung cấp cho họ chứng chỉ SSL miễn phí để người dùng có được trải nghiệm an toàn, đáng tin cậy, độ trễ thấp dù họ ở đâu.

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. Sử dụng Firebase Test Lab trong suốt vòng đời phát triển để khám phá lỗi và sử dụng không nhất quán để có thể cung cấp một trải nghiệm tuyệt vời trên nhiều thiết bị.

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 phân cụ thể trong ứng dụng và xem chế độ xem tổng hợp trong bảng điều khiển Firebase. Luôn cập nhật thời gian khởi động của ứng dụng và theo dõi các yêu cầu HTTP mà không cần viết bất kỳ mã nào.

3.2.3 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 và 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 đề hoặc phân đoạn người dùng 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 nhắm 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 hội 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 thực 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 đã 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.

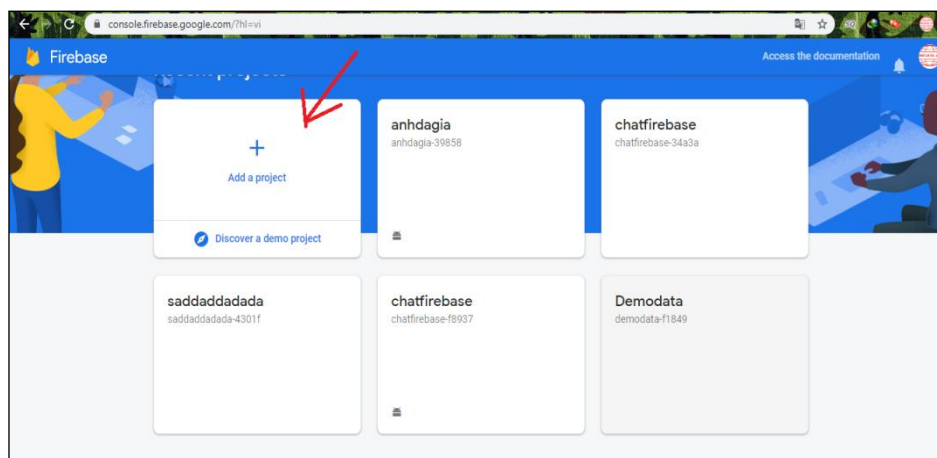
AdMob: Kiếm tiền bằng cách hiển thị quảng cáo hấp dẫn cho khán giả toàn cầu. AdMob có tất cả những ứng dụng người dùng cần để thực hiện chiến lược kiếm tiền trên lớp bậc nhất và để tối đa hóa doanh thu nhỏ mỗi người dùng tạo ra. Nó có thể được điều chỉnh cho ứng dụng của chúng ta, và API của nó được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

AdWords: Có được thông tin và nút 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. Tóm lại: hiện project của mình chỉ tích hợp tính năng Cloud Messaging gửi tin nhắn và thông báo cho người dùng qua các nền tảng Android, iOS. Nên chúng ta sẽ tìm hiểu cách tích hợp Firebase vào project của mình.

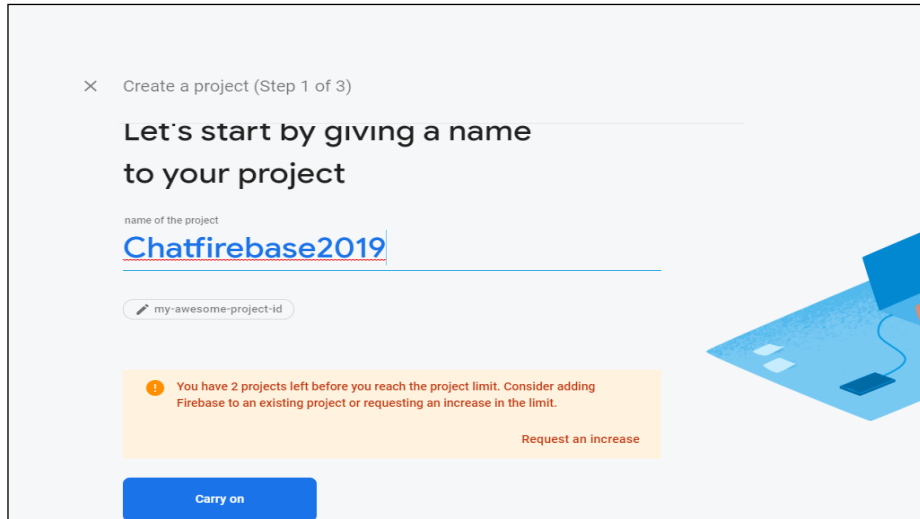
3.3 Một số kỹ thuật cơ bản

3.3.1 Tạo dự án trên Firebase

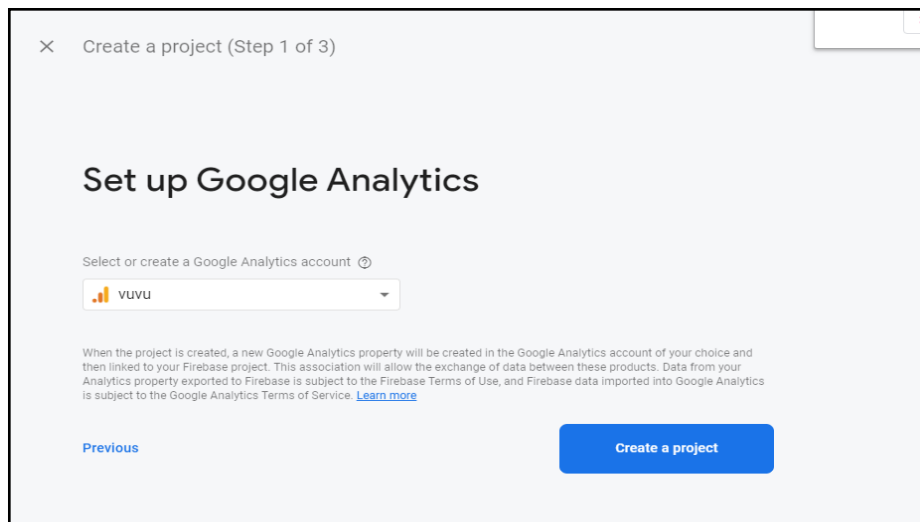
Điều đầu tiên cần làm là đi đến <https://firebase.google.com/> và tạo một tài khoản hoặc dùng tài khoản Gmail của google để truy cập vào giao diện điều khiển. Sau khi truy cập vào giao diện điều khiển, người dùng có thể bắt đầu bằng cách tạo ra các dự án đầu tiên.



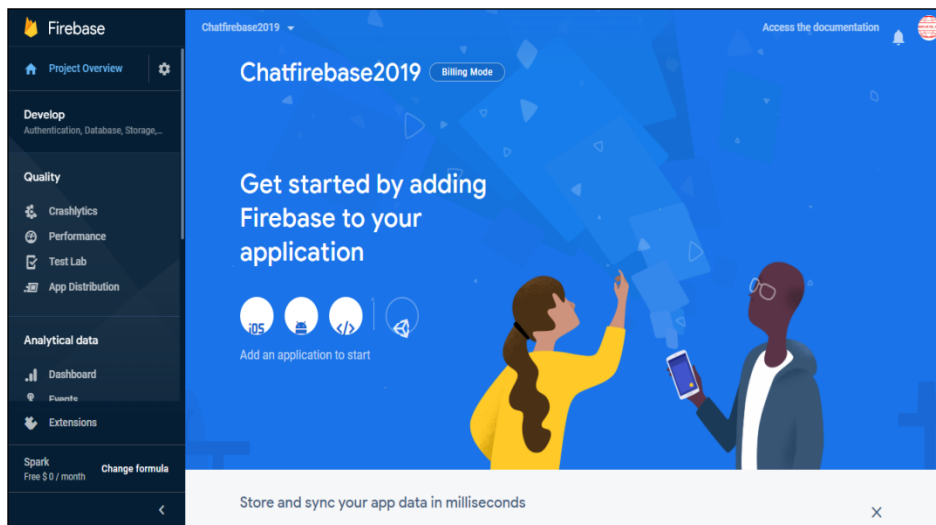
Đặt tên cho dự án của cùng một tên với tên dự án trong android studio



Sau đó nhấn vào Next để tiếp tục tạo dự án trên firebase.



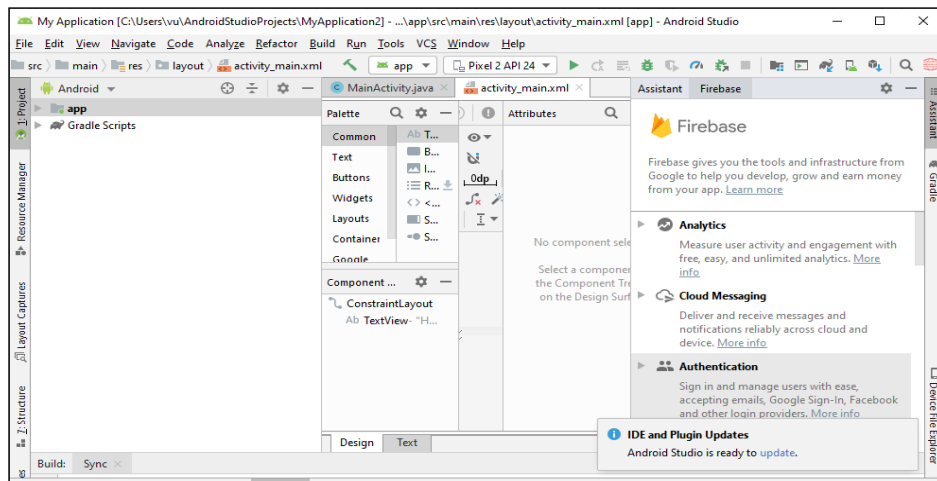
Chọn hoặc tạo tài khoản Google Analytics, sau đó tiếp tục ấn tạo dự án, đợi dự án được tạo ra như hình bên dưới.



3.3.2 Nhúng dự án Firebase vào ứng dụng

Để kết nối dự án vào trong android studio với firebase ta cần làm như sau:

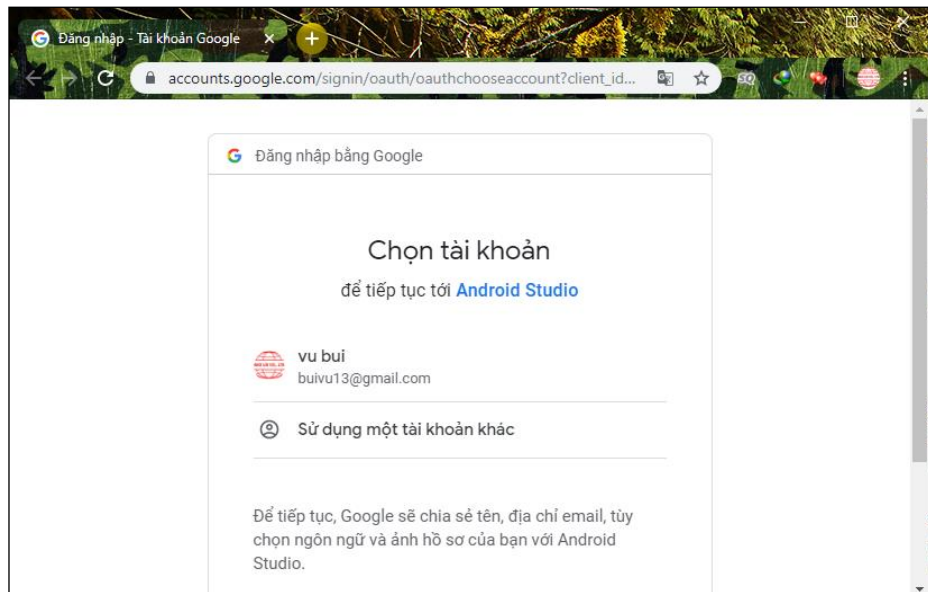
Vào menu Tools/ chọn Firbase, lúc này cửa sổ hướng dẫn cách sử dụng Firebase sẽ được hiển thị ra ở góc phải của cửa sổ Android Studio.



Trong màn hình trên, Firebase cung cấp rất nhiều công cụ ở đây ta làm việc với Realtime Database nên chúng ta sẽ chọn Realtime Database. Lúc này chúng ta thấy nút “Save and retrieve data” hiển thị ra như hình trên. Bấm vào nó nha. Trình trợ giúp cung cấp cho ta 8 bước thao tác (bao gồm kết nối và hướng dẫn lập trình, chúng ta dùng 2 -3 bước là đủ rồi):

- bước 1: Connect your app to firebase
- bước 2: Add the Realtime Database to our app
- bước 3: Configure Firebase Database Rules
- bước 4: Write to your database
- bước 5: Read from your database
- bước 6: Optional: Configure ProGuard
- bước 7: Prepare for Launch
- Bước 8: Next steps

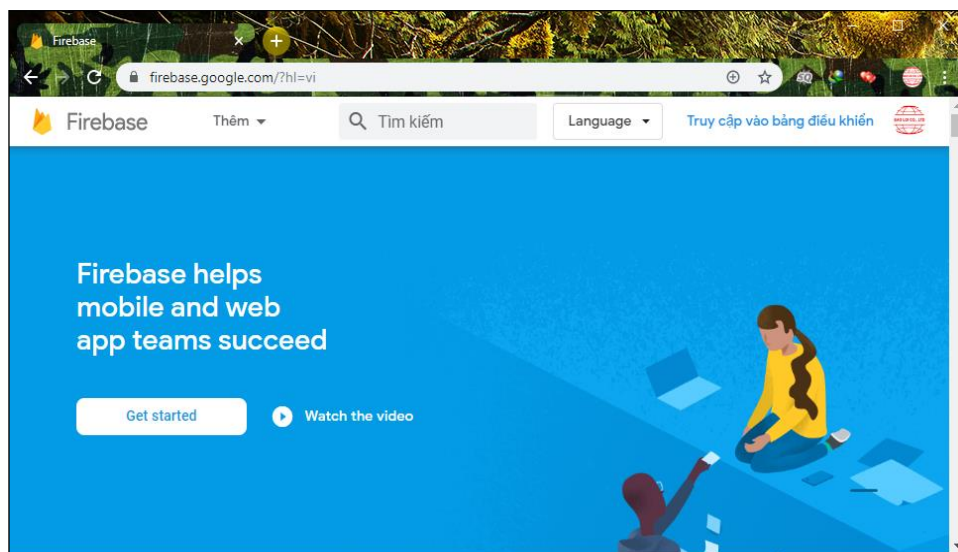
Ở bước 1, nhấn vào nút “Connect to Firebase”, lúc này hệ thống sẽ yêu cầu đăng nhập bằng tài khoản gmail:



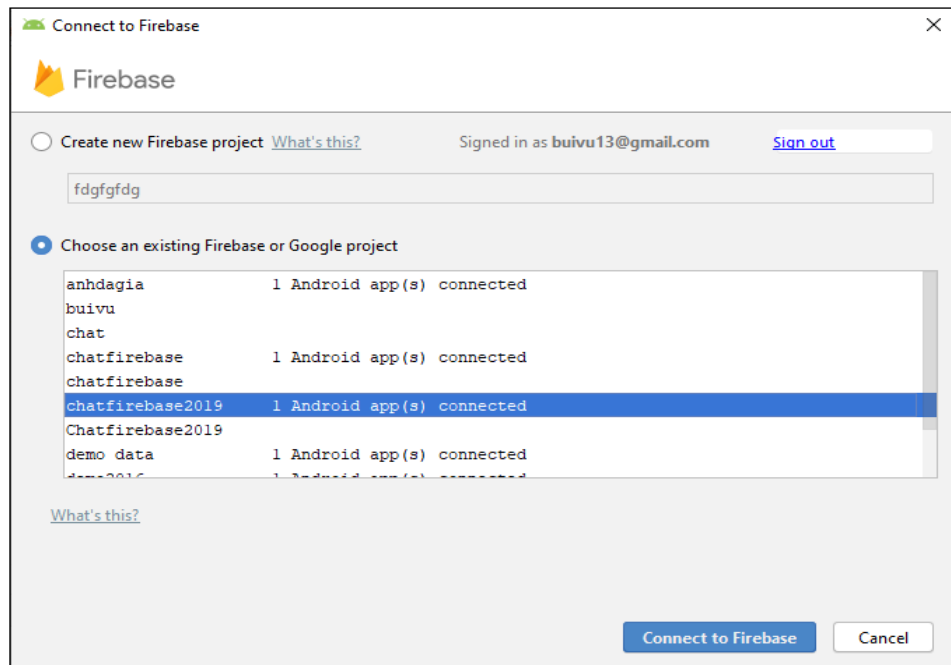
Lựa chọn tài khoản email, ở trên có sẵn email của mình: buivu13@gmail.com, nếu mình muốn làm các email khác thì nhấn vào Sử dụng một tài khoản khác.

Sau khi chọn Email để đăng nhập, Google yêu cầu người dùng cung cấp một số quyền hạn như sau. Nhấn Cho phép để tiếp tục, lúc này màn hình Developer của Android sẽ hiển thị ra.

Ở đây có 2 server: Firebase và Google Cloud, chọn Firebase chúng ta có màn hình kết quả sau:



Chú ý góc phải trên cùng có nút “Truy cập vào bảng điều khiển”, nhấn vào đây để vào trang quản trị của Firebase. Lưu ý, Sau khi chúng ta có màn hình trên, thì lúc này ở Project Android Studio sẽ tự động xuất hiện màn hình xác nhận kết nối Firebase (nhớ để ý để quay lại Android Studio xác nhận):



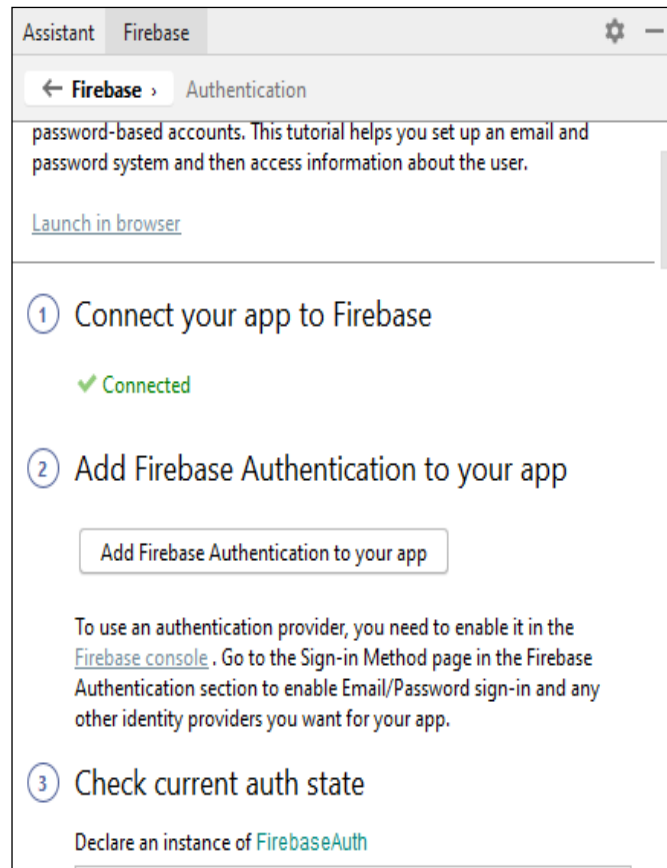
Ở đây có hai lựa chọn nếu chưa có dự án trên Firebase thì chọn Create new Firebase project, sau đó nhập tên dự án vào ô trống. Ở đây mình đã tạo dự án sẵn rồi thì mình sẽ chọn dự án của mình “chatfirebase2019” sau đó ấn Connect to Firebase để kết nối. Hệ thống xử lý một lúc sau khi kết nối xong hệ thống sẽ thông báo Connected.

3.3.3 Quản lý người dùng

Sẽ rất nguy hiểm nếu để cho một người nặc danh tham gia vào trong hệ thống chat. Để ngăn chặn điều này hệ thống yêu cầu người muốn tham gia hệ thống chat phải đăng ký tài khoản và dùng tài khoản này để đăng nhập vào hệ thống chat.

3.3.3.1 Đăng ký

Muốn đăng ký người dùng trước tiên cần phải kết nối và thêm thư viện Firebase vào Android Studio. Ở đây mình đã kết nối rồi chỉ cần thêm thư viện firebase Authentication.



Chúng ta vào Authentication sau đó chọn Add Firebase Authentication to your app để thêm các phụ thuộc cho thư viện Android Firebase Authentication vào tệp Gradle trong thư mục app/build.gradle. Ở đây mình sử dụng cả Realtime Database và Storage nên mình sẽ thêm các phụ thuộc của hai dịch vụ này vào trong Android Firebase làm tương tự như làm với Authentication.

```
implementation 'com.google.firebase:firebase-auth:16.0.5'  
implementation 'com.google.firebase:firebase-database:16.0.4'  
implementation 'com.google.firebase:firebase-storage:16.0.4'
```

Để sử dụng Authentication, cần kích hoạt nó trong bảng điều khiển Firebase. Chuyển đến trang phương thức đăng nhập trong phần Firebase Authentication để bật đăng nhập Email/ Mật khẩu và bất kỳ nhà cung cấp xác thực nào khác muốn cho ứng dụng của mình.

Khai báo biến kiểu FirebaseAuth.

```
private FirebaseAuth mAuth;
```

Trong phương thức onCreate(), khởi tạo FirebaseAuth.

```
mAuth = FirebaseAuth.getInstance();
```

Để đăng ký người dùng mới trước tiên phải kiểm tra xem địa chỉ email và password của người dùng được nhập từ ô EditText có hợp lệ hay không. Email thì cần phải đúng định dạng của email, password thì phải từ 6 ký tự trở lên. Sau khi email và password đã hợp lệ chúng ta sẽ gọi phương thức createUserWithEmailAndPassword để tạo người dùng mới. Nếu đăng ký thành công thì hệ thống sẽ hiển thị thông báo là “Đăng ký thành công” ngược lại sẽ báo “Lỗi!!!” không đăng ký được.

```
private void dangky(){
    String email=mailedt.getText().toString();
    String password=passwordedt.getText().toString();
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches())
    {
        mailedt.setError("Email không hợp lệ");
        mailedt.setFocusable(true);
    }
    else if (password.length()<6)
    {
        passwordedt.setError("Độ dài mật khẩu ít nhất 6 ký tự");
        passwordedt.setFocusable(true);
    }
    else {
        mAuth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        Putindatabase();

                        Toast.makeText(RegisterActivity.this, "Đăng ký thành công ",
                                Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(RegisterActivity.this,
                                DashboardActivity.class));
                    } else {
                        Toast.makeText(RegisterActivity.this, "Lỗi!!!",
                                Toast.LENGTH_SHORT).show();
                    }
                }
            })
    }
    private void Putindatabase() {
        FirebaseUser user=mAuth.getCurrentUser();
        //
    }
}
```

```

String email = user.getEmail();
String uid=user.getUid();
//
HashMap<Object,String> hashMap=new HashMap<>();
//
hashMap.put("email",email);
hashMap.put("uid",uid);
hashMap.put("name", "");
hashMap.put("onlineStatus","online");
hashMap.put("typingTo", "noOne");
hashMap.put("phone", "");
hashMap.put("image", "");
hashMap.put("cover", "");
//
FirebaseDatabase database=FirebaseDatabase.getInstance();
//
DatabaseReference reference=database.getReference("Users");
//
reference.child(uid).setValue(hashMap);
    }
});
}
}

```

Sau khi đăng ký thành công thì lấy thông tin email và password và tạo một số trường khác đẩy vào trong Realtime Database trong hàm Putindatabase.

3.3.3.2 Đăng nhập

Để thực hiện chat trên hệ thống thì người dùng cần phải đăng nhập. Nếu đã có tài khoản của ứng dụng chat thì chỉ cần vào phần đăng nhập thôi. Ở phần đăng nhập cũng kiểm tra xem người dùng nhập email và password của họ được nhập từ ô EditText có đúng đủ theo các điều kiện ở phần đăng ký hay không. Nếu đúng thì chương trình sẽ gọi phương thức signInWithEmailAndPassWord.

```

btndangnhap.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String email=mailedt.getText().toString();
        String password=passwordedt.getText().toString();
        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches())
        {
            mailedt.setError("Email không hợp lệ");
            mailedt.setFocusable(true);
        }
    }
}

```

```

}
else if (password.length()<6)
{
    passwordedt.setError("Độ dài mật khẩu ít nhất 6 ký tự");
    passwordedt.setFocusable(true);

}
else {
    mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new
    OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful())
            {
                if (task.getResult().getAdditionalUserInfo().isNewUser())
                {
                    FirebaseUser user=mAuth.getCurrentUser();
                    //
                    String email = user.getEmail();
                    String uid=user.getUid();
                    //
                    HashMap<Object,String> hashMap=new HashMap<>();
                    //
                    hashMap.put("email",email);
                    hashMap.put("uid",uid);
                    hashMap.put("name","");
                    hashMap.put("onlineStatus","online");
                    hashMap.put("typingTo","noOne");
                    hashMap.put("phone","");
                    hashMap.put("image","");
                    hashMap.put("cover","");
                    //
                    FirebaseDatabase database=FirebaseDatabase.getInstance();
                    //
                    DatabaseReference reference=database.getReference("Users");
                    //
                    reference.child(uid).setValue(hashMap);
                }
                // Putindatabase();
                //
                Toast.makeText(LoginActivity.this, "Đăng nhập thành công ",
                Toast.LENGTH_SHORT).show();
                startActivity(new Intent(LoginActivity.this, DashboardActivity.class));
            }
            else {
                Toast.makeText(LoginActivity.this, "Lỗi!!!",
                Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

    }
    });
}
}
});

```

Nếu đăng nhập thành công hệ thống sẽ thông báo "Đăng nhập thành công" và chuyển tới giao diện trang chủ, ngược lại sẽ báo "Lỗi".

3.3.3.3 Chỉnh sửa thông tin cá nhân

Khi người dùng đăng ký để vào hệ thống chat mới chỉ có một thông tin là địa chỉ email như vuvu@gmail.com. Chức năng chỉnh sửa profile của hệ thống cho phép người dùng hoàn thiện các thông tin mà hệ thống yêu cầu như: họ tên, số điện thoại, ảnh đại diện, ảnh nền sau khi họ đăng nhập thành công. Để có thể chỉnh sửa ảnh cần có một tấm ảnh được chụp bằng thiết bị hay lấy ảnh có sẵn trong thư viện ảnh. Trước tiên cần khai báo và khởi tạo database bằng một biến có kiểu StorageReference.

```
StorageReference storagereference = storage.getReference();
```

Khai báo một biến chứa đường dẫn và tên của hình ảnh, sau đó đẩy nó vào trong. Tạo một tham chiếu tới đường dẫn chính tới biến vừa khai báo. Tiếp theo dùng phương thức putFile để tải ảnh lên Storage.

```
String filepathAndName = storagePath+ ""+ profileOrCoverPhoto +"_"+
user.getId();
StorageReference storageReference2nd=storageReference.child(filepathAndName);
storageReference2nd.putFile(uri).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>()
```

Sau khi tải tệp lên, chúng ta có thể nhận URL để tải xuống tệp bằng cách gọi phương thức getDownloadUrl() trên StorageReference.

```
Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
Uri downloadUri=uriTask.getResult();
```

Nếu tải thành công URL đó thì sẽ đẩy đường dẫn URL sang bên Database để update hình ảnh bằng phương thức:

```
databaseReference.child(user.getId()).updateChildren(results).addOnSuccessListener(new
OnSuccessListener<Void>() {
@Override
```



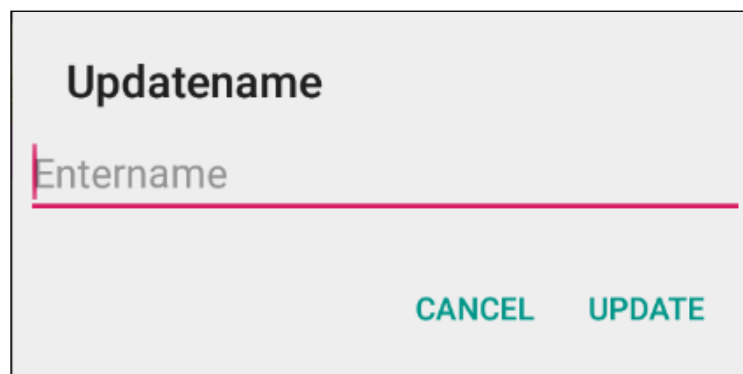
```

public void onSuccess(Void aVoid) {
    pd.dismiss();
    Toast.makeText(getActivity(), "Ảnh đã được cập nhật",
    Toast.LENGTH_SHORT).show();
}
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        pd.dismiss();
        Toast.makeText(getActivity(), "Lỗi cập nhật hình ảnh",
        Toast.LENGTH_SHORT).show();
    }
});
}

```

Cập nhập thành công hệ thống sẽ thông báo " Image Updated ", cập nhập không thành công hệ thống sẽ báo "Error Updated Image". Để cập nhập thêm tên và số điện thoại thì làm như sau:

Trước tiên cần tạo ra một LinearLayout chứa ô EditText để nhập thông tin cần chỉnh sửa.



Sau khi nhập thông tin vào EditText chúng ta sẽ bắt sự kiện nút UPDATE. Để thêm dữ liệu, ở đây mình sử dụng phương thức HashMap.Put() trên đường dẫn đến database.

```

if (!TextUtils.isEmpty(value)){
    pd.show();
    HashMap<String, Object>result=new HashMap<>();
    result.put(key, value);
    databaseReference.child(user.getUid()).updateChildren(result).addOnSuccessListener(new
    OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            //

```

```

        pd.dismiss();
        Toast.makeText(getActivity(), "Updated...", Toast.LENGTH_SHORT).show();
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        pd.dismiss();
        Toast.makeText(getActivity(), "Lỗi"+e.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}
else{
    Toast.makeText(getActivity(), "Please enter"+key, Toast.LENGTH_SHORT).show();
}
}

```

Nếu cập nhập thành công thì bộ nghe sự kiện thành công sẽ được gọi và hiển thị thông báo "Updated", nếu không thành công thì bộ lắng không thành công được gọi và thông báo "Lỗi".

3.3.3.4 Tải danh sách thành viên về ứng dụng

Để hỗ trợ người dùng dễ dàng tìm được bạn chat, ứng dụng sẽ tải tất cả các thành viên của hệ thống về thiết bị. Để có thể hiển thị được danh sách thành viên trước tiên cần tạo ra một class ModelUser để lưu trữ các dữ liệu gồm có các thuộc tính như sau:

```
String name,email,search,phone,image,cover,uid;
```

Các thuộc tính trong class ModelUser gồm có:

- name: Tên
- email: Địa chỉ email
- search: tìm kiếm
- phone: Số điện thoại
- image: ảnh đại diện
- cover: ảnh nền
- uid: Id tài khoản

Ở đây mình tạo ra một khởi tạo Constructor để đẩy dữ liệu lên và tạo một Constructor rỗng để nhận dữ liệu về.

Bình thường nếu sử dụng ArrayAdapter có sẵn, thì listview chỉ hiện thị dữ liệu ở một đoạn text ở trên 1 dòng, còn bây giờ trên một dòng có cả text và image view thì không dùng nó được mà phải custom lại dữ liệu có thể hiển thị được, cho nên chúng ta sẽ tạo class AdapterUser.

Tiếp theo sẽ hiển thị tất cả các tài khoản trong UserFragmentt.java. Tại đây khai báo các biến như sau:

```
RecyclerView recyclerView;  
AdapterUser adapterUser;  
List<ModeIUser> userList;  
FirebaseAuth mAuth;
```

Ở đây mình xây dựng hàm getAllUser để hiển thị danh sách người dùng lên RecyclerView.

```
private void getAllUsers() {  
    //get current user  
    final FirebaseUser fUser= FirebaseAuth.getInstance().getCurrentUser();  
    //Get path of database named "Users" containing users info  
    DatabaseReference ref= FirebaseDatabase.getInstance().getReference("Users");  
    ref.addValueEventListener(new ValueEventListener() {  
        @Override  
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
            userList.clear();  
            for (DataSnapshot ds: dataSnapshot.getChildren()){  
                ModeIUser modeIUser=ds.getValue(ModeIUser.class);  
                //get all users except currently signed in user  
                if (!modeIUser.getUid().equals(fUser.getUid())){  
                    userList.add(modeIUser);  
                    //adapter  
                    adapterUser=new AdapterUser(getActivity(),userList);  
                    //set adapter to recycler view  
                    recyclerView.setAdapter(adapterUser);  
                }  
            }  
        }  
        @Override  
        public void onCancelled(@NonNull DatabaseError databaseError) {  
        }  
    });  
}
```

3.3.4 Chat trên kênh riêng tư

Để có thể chat giữa hai tài khoản với nhau, hiển thị được nội dung tin nhắn, hiển thị được thời gian thực và ảnh đại diện của gửi đầu tiên cần tạo ra một class ModelChat để lưu trữ các dữ liệu gồm có các thuộc tính như sau:

```
String message,receiver,sender,timestamp;  
boolean isSeen;  
boolean redelete;
```

Các thuộc tính trong class ModelChat gồm có:

- message: nội dung tin nhắn
- receiver: Id người nhận
- sender: Id người gửi
- timestamp: thời gian gửi/nhận tin nhắn
- isSeen: đã xem
- redelete: xóa tin nhắn người gửi

Ở đây mình tạo ra một khởi tạo Constructor để đẩy dữ liệu lên và tạo một Constructor rỗng để nhận dữ liệu về.

Bình thường nếu sử dụng ArrayAdapter có sẵn, thì listview chỉ hiển thị dữ liệu ở một đoạn text ở trên 1 dòng, còn bây giờ trên một dòng có cả text và image view thì không dùng nó được mà phải custom lại dữ liệu có thể hiển thị được, cho nên chúng ta sẽ tạo class AdapterChat.

Khi chạm vào tài khoản mà chúng ta muốn chat thì hệ thống sẽ chuyển sang giao diện chat.

```
myHolder.itemView.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent intent = new Intent(context,ChatActivity.class);  
        intent.putExtra("hisUid",hisUID);  
        context.startActivity(intent);  
    }  
}
```

Trong quá trình chuyển đổi giữa hai giao diện đồng thời cũng đây hisUid của người mình muốn chat sang bên ChatActivity. Ở bên ChatActivity muốn lấy hiUid sử dụng câu lệnh như sau:

```
Intent intent = getIntent();
hisUid = intent.getStringExtra("hisUid");
```

Ở bên giao diện chat sẽ bắt sự kiện cho nút gửi tin nhắn. Nếu như dữ liệu ở trong ô EditText mà rỗng thì sẽ hiển thị thông báo "cannot send the empty message". Ngược lại sẽ gọi tới hàm sendMessage(message):

```
private void sendMessage(String message) {
    DatabaseReference databaseReference=FirebaseDatabase.getInstance().getReference();
    String timestamp=String.valueOf(System.currentTimeMillis());
    HashMap<String,Object>hashMap=new HashMap<>();
    hashMap.put("sender",myUid);
    hashMap.put("receiver",hisUid);
    hashMap.put("message",message);
    hashMap.put("timestamp",timestamp);
    hashMap.put("isSeen",false);
    hashMap.put("redelete",true);
    databaseReference.child("Chats").push().setValue(hashMap);
    //reset edittext
    messageEt.setText("");
}
```

Sau khi dữ liệu đã được đẩy lên Database ta sẽ hiển thị nội dung tin nhắn, ảnh người gửi và thời gian thực nhận gửi tin nhắn bằng hàm dưới đây:

```
private void readMessages() {
    chatList = new ArrayList<>();
    DatabaseReference dbRef= FirebaseDatabase.getInstance().getReference("Chats");
    dbRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            chatList.clear();
            for (DataSnapshot ds: dataSnapshot.getChildren()){
                ModelChat chat = ds.getValue(ModelChat.class);
                if ((chat.getReceiver().equals(myUid)&&
chat.getSender().equals(hisUid)&&ds.child("redelete").getValue().equals(true))||
                    chat.getReceiver().equals(hisUid)&&chat.getSender().equals(myUid))
                {
                    chatList.add(chat);
                }
            }
            adapterChat=new AdapterChat(ChatActivity.this,chatList,hisImage);
            adapterChat.notifyDataSetChanged();
            //set adpter to recyclerview
            recyclerView.setAdapter(adapterChat);
        }
    });
}
```

```

    }
}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

}
});
}

```

Ở đây mình khai báo và khởi tạo DatabaseReference để truy xuất vào root Chats. Sau đó mình gọi biến vừa khai báo thêm phương thức addValueEventListener(new ValueEventListener()) và khởi tạo cho nó, nó sẽ tự phát sinh phương thức cho mình gồm có onDataChange và onCancelled. Trong hàm onDataChange thì cứ khi nào có sự thay đổi giá trị trong root Chats thì lập tức trả về trong hàm onDataChange và có một tham số dataSnapshot. Dùng vòng lặp for để lấy tất cả các giá trị trong root Chats. Ở đây kiểm tra điều kiện nếu id người nhận mà bằng myUid và id người gửi bằng hisUid và trường "redelete" có giá trị bằng true, hoặc là id người nhận bằng hisUid và id người gửi bằng myUid thì thêm giá trị nội dung chat vào chatList. Sau đó đẩy vào adapterChat sau rồi đẩy ra hiển thị trong recyclerView.

3.3.5 Xóa tin nhắn chat

Muốn xóa tin nhắn chúng ta bắt sự kiện click vào messageLayout:

```

myHoder.messageLayout.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle("Xóa");
        builder.setMessage("Bạn có chắc chắn muốn xóa tin nhắn này?");
        // delete button
        builder.setPositiveButton("Xóa", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int which) {
                deleteMessage(i);
            }
        });
        // cancel delete button
        builder.setNegativeButton("Hủy", new DialogInterface.OnClickListener() {
            @Override

```

```

public void onClick(DialogInterface dialogInterface, int which) {
    // dismiss dialog
    dialogInterface.dismiss();
}
});
// create and show dialog
builder.create().show();
}
});

```

Tạo ra 2 button "Delete" và "No" ở đây bắt sự kiện của button "Delete" gọi hàm deleteMessage(i).

```

private void deleteMessage(int position) {
    final String myUID = FirebaseAuth.getInstance().getCurrentUser().getUid();
    String timestamp=chatList.get(position).getTimestamp();
    DatabaseReference dbRef = FirebaseDatabase.getInstance().getReference("Chats");
    Query query = dbRef.orderByChild("timestamp").equalTo(timestamp);
    query.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for (DataSnapshot ds: dataSnapshot.getChildren()){
                if (ds.child("sender").getValue().equals(myUID)){
                    DatabaseReference databaseReference =
                    FirebaseDatabase.getInstance().getReference("Chats");
                    HashMap<String,Object> hashMap = new HashMap<>();
                    hashMap.put("senddelete",false);
                    ds.getRef().updateChildren(hashMap);
                    Toast.makeText(context, "Đã xóa...", Toast.LENGTH_SHORT).show();
                }
                else if (ds.child("receiver").getValue().equals(myUID)){
                    DatabaseReference databaseReference =
                    FirebaseDatabase.getInstance().getReference("Chats");
                    HashMap<String,Object> hashMap = new HashMap<>();
                    hashMap.put("redelete",false);
                    ds.getRef().updateChildren(hashMap);
                    Toast.makeText(context, "Đã xóa...", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
}
});

```

Xóa tin nhắn của mình gửi đi:

Lấy id người dùng hiện thời, lấy thời gian chat trên của tin nhắn mình muốn xóa trên ứng dụng so sánh với thời gian của các tin nhắn trong database

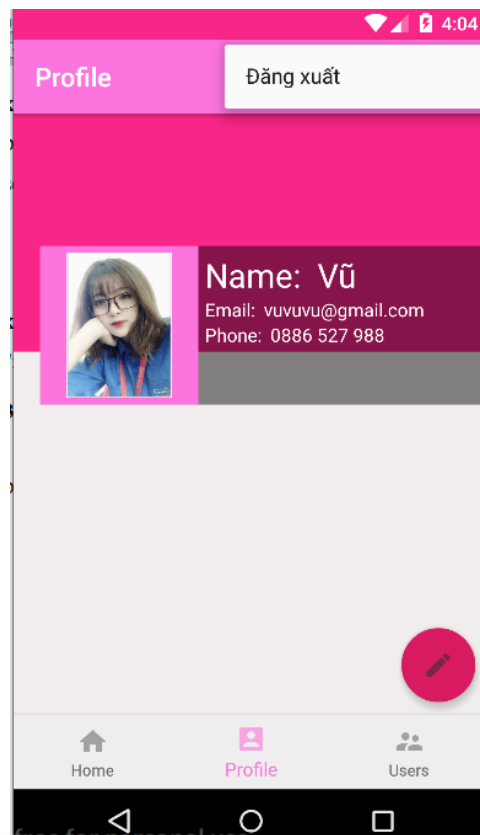
xem có bằng nhau hay không và id của người gửi bằng với myUid thì cho giá trị senddelete = false sẽ xóa được tin nhắn đó.

Xóa tin nhắn của người khác gửi tới cho mình:

Nếu Id người nhận có bằng với myUid, thời gian của tin nhắn mình muốn xóa trên chatList bằng với thời gian tin nhắn trong database thì chuyển giá trị "redelete" bằng true sau đó update lại vào database thì sẽ xóa được tin nhắn đó.

3.3.6 Đăng xuất

Muốn đăng xuất khỏi hệ thống chỉ cần chạm vào phần đăng xuất ở góc trên bên trái ứng dụng để đăng xuất.



Hệ thống sẽ tự động gọi tới phương thức mAuth.signOut() để đăng xuất.

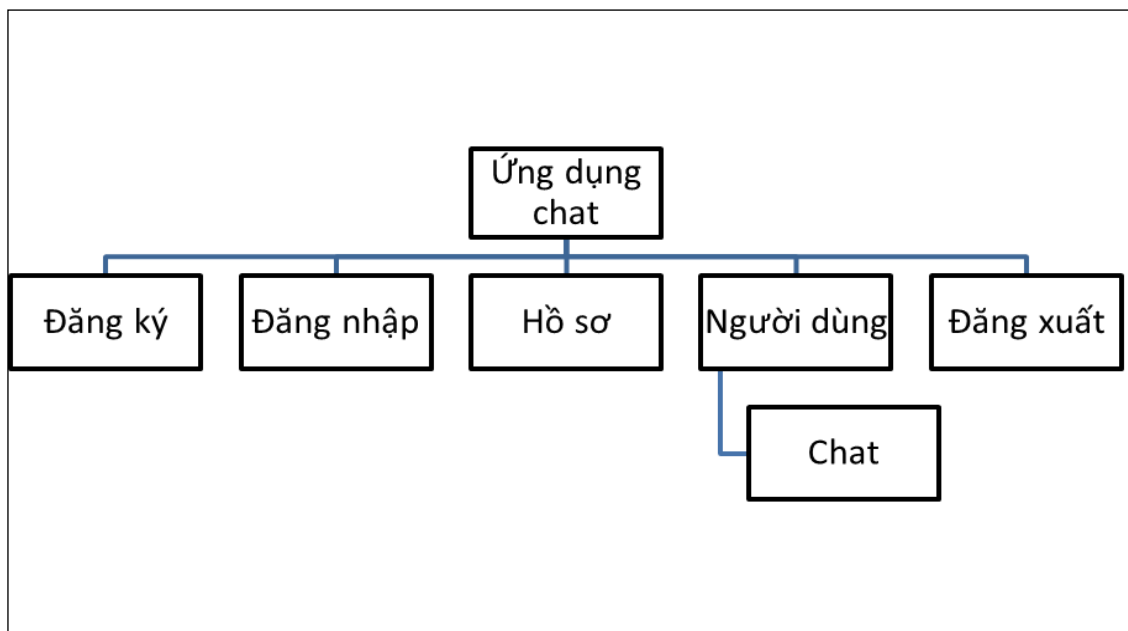
3.4 Bài toán

Trao đổi thông tin với nhau hàng ngày là một nhu cầu tất yếu của mọi người. Cuộc cách mạng công nghiệp 4.0 ngày càng thu hẹp khoảng cách giữa mọi người với nhau bằng các ứng dụng mạng xã hội, chat trực tuyến. Tuy

nhiệm vụ sử dụng các dịch vụ này để chat với nhau trên thiết bị di động vẫn còn có nhiều điểm bất tiện như dung lượng cài ứng dụng đặt lớn, tiêu tốn nhiều bộ nhớ ram và nhất là xử lý gửi và nhận tin nhắn chưa thực thi theo thời gian thực. RealTime Firebase Database của Google Firebase là cơ sở dữ liệu thời gian thực có khả năng xử lý dữ liệu đủ nhanh và cho kết quả phản hồi gần như tức thời. Ứng dụng chat thời gian thực dựa trên RealTime Firebase Database giúp mọi người nhận được các thông tin quan trọng và ngay lập tức từ phía đối tác.

3.5 Sơ đồ chức năng

Sơ đồ phân rã chức năng chương trình chat



Hình 3.5.1: Sơ đồ chức năng chương trình chat

3.6 Thiết kế dữ liệu

3.6.1 Bảng dữ liệu người dùng

| STT | Tên trường | Kiểu dữ liệu | Độ rộng | Mô tả |
|-----|------------|--------------|---------|----------------|
| 1 | name | String | | Tên người dùng |

| | | | | |
|---|--------|--------|--|--------------------|
| 2 | email | String | | Địa chỉ email |
| 3 | search | String | | Tìm kiếm |
| 4 | phone | String | | Số điện thoại |
| 5 | image | String | | đường dẫn hình ảnh |
| 6 | cover | String | | đường dẫn ảnh nền |
| 7 | uid | String | | Id người dùng |

Hình 3.6.1: Dữ liệu người dùng

3.6.2 Bảng thông tin nội dung tin nhắn

| STT | Tên trường | Kiểu dữ liệu | Độ rộng | Mô tả |
|-----|------------|--------------|---------|------------------------|
| 1 | message | String | | nội dung tin nhắn |
| 2 | receiver | String | | Id người nhận |
| 3 | sender | String | | id người gửi |
| 4 | timestamp | String | | thời gian tin nhắn |
| 5 | isSeen | boolean | | đã xem/chưa xem |
| 6 | redelete | boolean | | xóa tin nhắn người gửi |

Hình 3.6.2: Thông tin nội dung tin nhắn

3.7 Giao diện chương trình

3.7.1 Giao diện chính

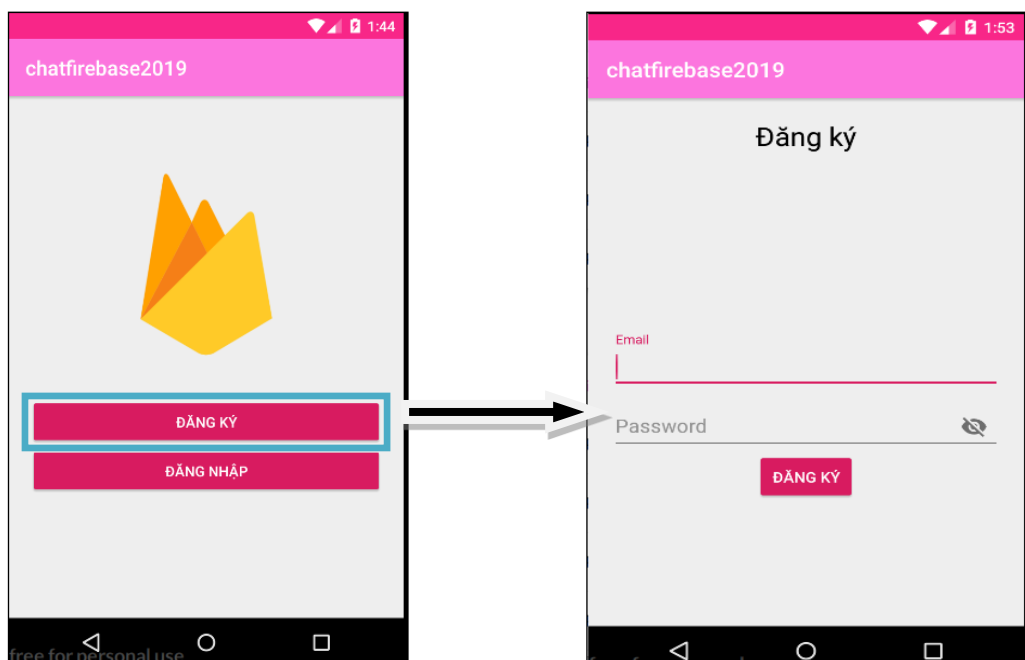
Khi ứng dụng được khởi động. Giao diện sẽ hiển thị như hình bên dưới.



Hình 3.7.1: Giao diện chính của chương trình

3.7.2 Giao diện đăng ký

Sau khi chạm vào nút "ĐĂNG KÝ" ở giao diện chính của chương trình thì chương trình sẽ chuyển qua giao diện đăng ký như hình bên dưới.

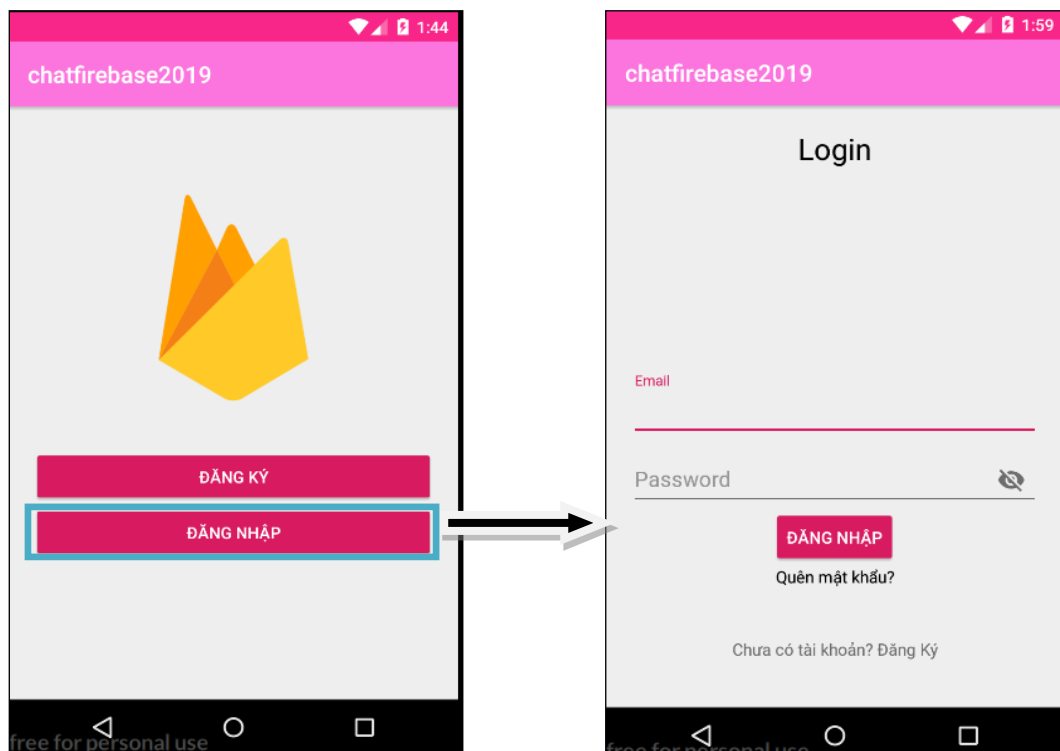


Hình 3.7.2: Giao diện đăng ký tài khoản

Sau khi người dùng nhập thông tin email password người dùng ấn nút "Đăng ký". Đăng ký thành công giao diện sẽ chuyển qua màn hình thông tin người dùng.

3.7.3 Giao diện đăng nhập

Sau khi có tài khoản chạm vào nút "ĐĂNG NHẬP" ở giao diện chính của chương trình thì chương trình sẽ chuyển qua giao diện đăng nhập như hình bên dưới.

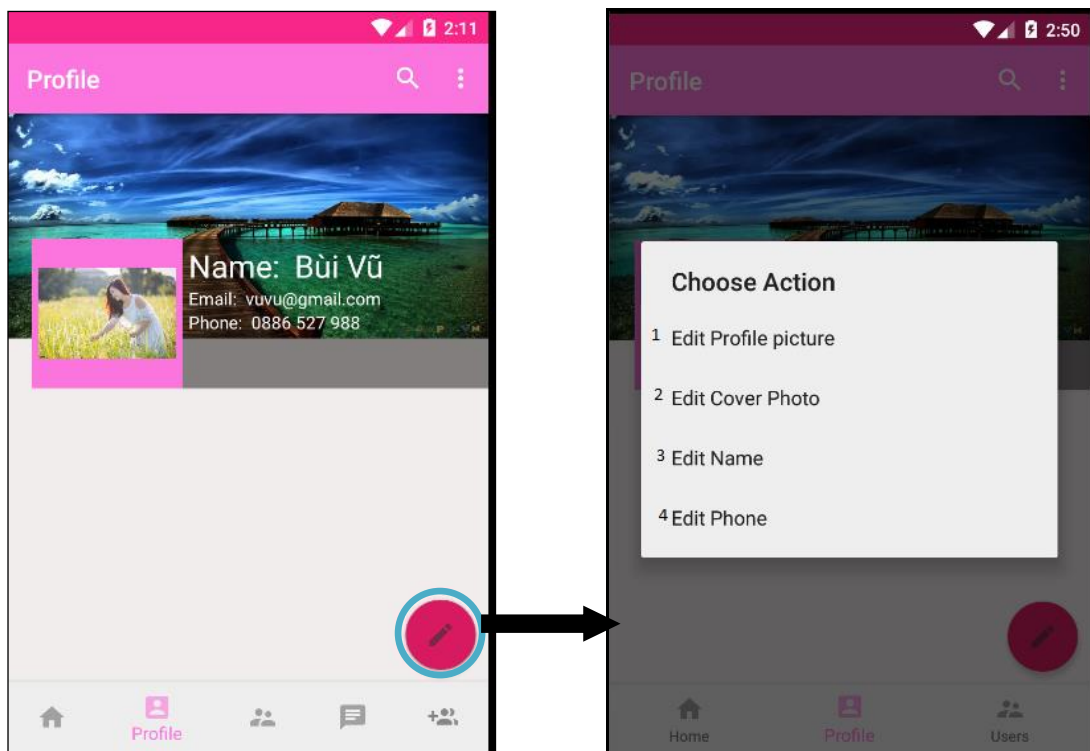


Hình 3.7.3: Giao diện đăng nhập tài khoản

Người dùng nhập email và password sau đó ấn "ĐĂNG NHẬP" nếu thành công hệ thống sẽ chuyển qua giao diện thông tin người dùng.

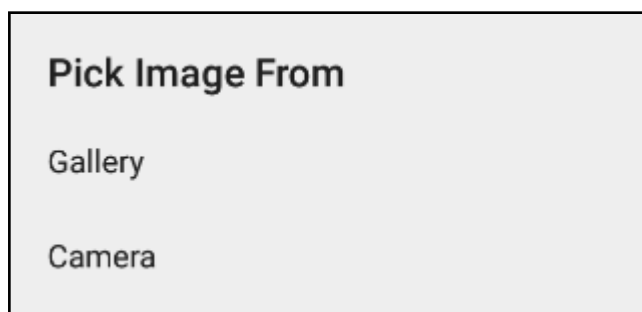
3.7.4 Giao diện thông tin người dùng

Giao diện thông tin người dùng sau khi đăng nhập hoặc đăng ký thành công. Sau khi chạm vào nút chỉnh sửa thông tin cá nhân, một màn hình sẽ hiển thị ra, lựa chọn chỉnh sửa ảnh đại diện, ảnh nền, tên, số điện thoại.



Hình 3.7.4: Giao diện thông tin người dùng

Để chỉnh sửa được ảnh đại diện chạm vào " Edit Profile picture" (1) sau đó sẽ ra được màn hình bên dưới.



Chọn ảnh trong thư viện hoặc ảnh ở camera chụp ảnh. Việc chỉnh sửa ảnh ở "Edit cover Photo" (2) cũng tương tự.

Để chỉnh sửa tên người dùng chạm vào "Edit Name" (3) sau đó một giao diện sẽ hiện ra để nhập thông tin.

Updatename

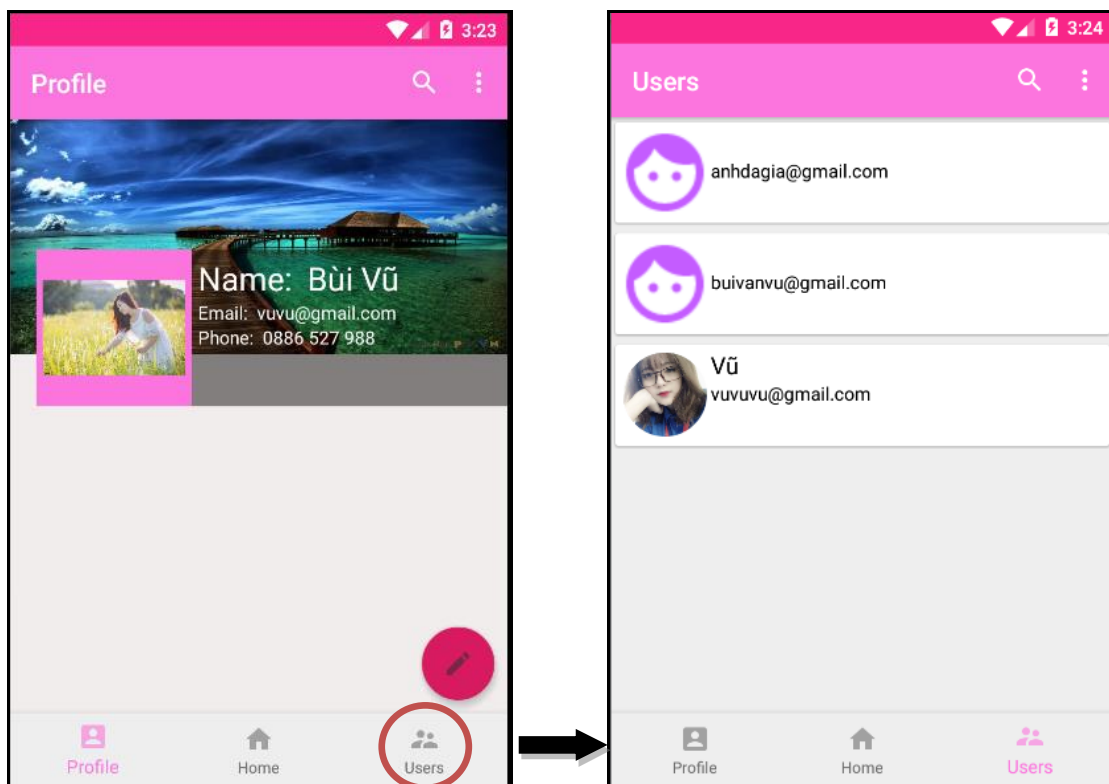
Entername

CANCEL UPDATE

Người dùng chỉ cần nhập tên cần chỉnh sửa và ô nhập dữ liệu và chạm vào "UPDATE" tên của người dùng sẽ được thêm mới hoặc chỉnh sửa. Còn nếu muốn hủy thao tác chạm vào "CANCEL". Với việc chỉnh sửa số điện thoại người dùng cũng tương tự như chỉnh sửa tên.

3.7.5 Giao diện danh sách người dùng

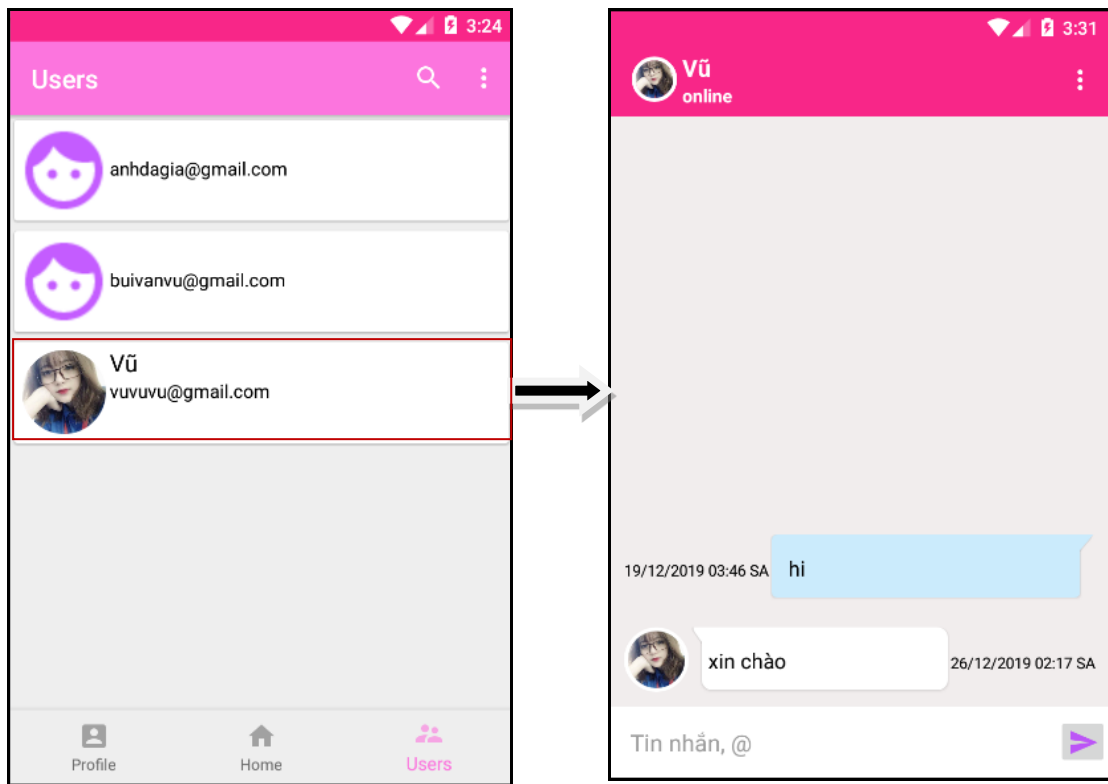
Muốn hiển thị danh sách người dùng chạm vào biểu tượng bên dưới để hiển thị ra được tất cả danh sách người dùng, giúp hiển thị ra được tất cả các tài khoản để tìm kiếm và lựa chọn người mà mình muốn chat.



Hình.3.7.5: Giao diện danh sách người dùng

3.7.6 Giao diện chat

Khi chạm vào một người dùng nào đó ở trên danh sách người dùng mà muốn gửi tin nhắn chương trình sẽ chuyển đến giao diện chat.



Hình.3.7.6: Giao diện Chat

Ở giao diện chat người dùng nhập nội dung vào ô tin nhắn sau đó chạm vào biểu tượng gửi tin nhắn thì lập tức nội dung tin nhắn sẽ được gửi thành công .

KẾT LUẬN

Sau một thời gian tìm hiểu đề tài: "Xây dựng ứng dụng chat trong android với firebase" em đã thực hiện được nội dung đề tài theo yêu cầu đặt ra. Với mục tiêu là tìm hiểu về hệ điều hành Android, tìm hiểu về môi trường lập trình Android Studio và tìm hiểu các kỹ thuật để xây dựng ứng dụng hoàn chỉnh trên thiết bị thật. Thông qua việc tìm hiểu nắm bắt lý thuyết về hệ điều hành Android, bước đầu đã xây dựng thành công chương trình thực nghiệm với các chức năng cơ bản.

- Trong quá trình tìm hiểu lý thuyết về hệ điều hành Android và xây dựng ứng dụng em đã thu được kết quả sau:
- Hiểu được các lớp đối tượng, phương thức hỗ trợ lập trình trên nền hệ điều hành Android.
- Hiểu được các dịch vụ trong Firebase
- Xây dựng ứng dụng với các chức năng: Đăng ký, đăng nhập, hiển thị danh sách người dùng, hiển thị và chỉnh sửa profile, chat trên kênh riêng tư.
- Đóng gói ứng dụng thành file.apk cho phép cài đặt trên điện thoại Android.
- Đưa thành công ứng dụng lên Google Play cho phép người dùng tìm hiểu và cài đặt trên thiết bị thật.

Do thời gian có hạn 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 của em còn nhiều hạn chế, em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và có nhiều chức năng hơn nữa để có thể áp dụng được vào trong đời sống.

TÀI LIỆU THAM KHẢO

- [1]. [https://vi.wikipedia.org/wiki/Android_\(hệ_điều_hành\)](https://vi.wikipedia.org/wiki/Android_(hệ_điều_hành))
- [2]. <https://fptshop.com.vn/tin-tuc/danh-gia/lich-su-phat-trien-cua-android-he-dieu-hanh-di-dong-pho-bien-nhat-the-gioi-55019>
- [3]. <http://androidtmc.blogspot.com/2015/08/android-tutorial-for-beginners-part-1.html>
- [4]. <https://xaiandroid.com/huong-dan-tao-may-ao-android-bang-genymotion-tren-may-tinh.html>
- [5]. <https://androidcoban.com/tong-quan-ve-firebase-google-phien-ban-moi.html>