

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

-----o0o-----



ISO 9001:2015

# ĐỒ ÁN TỐT NGHIỆP

Nghành Công Nghệ Thông Tin

Hải Phòng 2019

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

-----o0o-----

**TÌM HIỂU LẬP TRÌNH CƠ SỞ VỚI ANDROID FIREBASE  
VÀ ỨNG DỤNG**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

Nghành Công Nghệ Thông Tin

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----

**TÌM HIỂU LẬP TRÌNH CƠ SỞ VỚI ANDROID  
FIREBASE VÀ ỨNG DỤNG**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

Ngành: Công nghệ Thông tin

Sinh viên thực hiện : Hoàng Việt Anh

Mã sinh viên : 1512111025

Giáo viên hướng dẫn : Th.S Phùng Anh Tuấn

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

**Độc lập - Tự do - Hạnh phúc**

**-----o0o-----**

## **NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**

**Sinh viên: Hoàng Việt Anh**

**Mã sinh viên :1512111025**

**Lớp : CT1901C**

**Nghành: Công Nghệ Thông Tin**

**Tên đề tài: Tìm hiểu lập trình cơ sở với Android Firebase và ứng dụng**

# 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 hệ điều hành android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tìm hiểu về lập trình android firebase.
- Ứng dụng các kỹ thuật đã tìm hiểu xây dựng chương trình ứng dụng thực nghiệm.

### b. Các yêu cầu cần giải quyết

- Nắm được thao tác cơ bản với cơ sở dữ liệu trong android firebase.
- Sử dụng công cụ Android Studio để viết chương trình chạy trên điện thoại android
- Xây dựng được chương trình thực nghiệm quản lý quản lý danh bạ điện thoại, đóng gói chương trình thành file \*.apk cho phép cài đặt và chạy trên điện thoại android thật.

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

- Sử dụng số liệu thực tế của danh bạ điện thoại HPU

## 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 sĩ

Cơ quan công tác: Khoa Công nghệ Thông tin

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tìm hiểu về lập trình android firebase.
- Ứng dụng các kỹ thuật đã tìm hiểu xây dựng chương trình ứng dụng thực nghiệm

### 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 01 tháng 7 năm 2019

Yêu cầu phải hoàn thành trước ngày 21 tháng 9 năm 2019

**Đã nhận nhiệm vụ: Đ.T.T.N**

**Sinh viên**

Hoàng Việt Anh

**Đã nhận nhiệm vụ: Đ.T.T.N**

**Cán bộ hướng dẫn Đ.T.T.N**

Phùng Anh Tuấn

**Hải Phòng, ngày .....tháng.....năm 2019**

**HIỆU TRƯỞNG**

GS.TS.NGUT Trần Hữu Nghị

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP**

Họ và tên giảng viên: Phùng Anh Tuấn.

Đơn vị công tác: Trường Đại học Dân lập Hải Phòng.

Họ và tên sinh viên: Hoàng Việt Anh - Ngành: Công Nghệ Thông Tin

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tìm hiểu về lập trình android firebase.
- Ứng dụng các kỹ thuật đã tìm hiểu xây dựng chương trình ứng dụng thực nghiệm.

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

- Chưa chủ động trong nghiên cứu tài liệu phục vụ cho nội dung đề án.
- Khả năng làm việc độc lập còn nhiều hạn chế.
- Chưa làm chủ được nội dung đề án.

**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...)**

- Đã thực hiện được các nội dung đề ra trong đề cương.
- Nội dung đề án có tính thực tế.
- Cần trình bày nội dung đề án rõ ràng, logic hơn.

**3. Ý kiến của giảng viên hướng dẫn tốt nghiệp**

Đạt  Không đạt  Điểm:.....

Hải Phòng, ngày ..... tháng 01 năm 2019

**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 CHĂM 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 chăm phản biện**

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

**2. Những mặt còn hạn chế**

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

**3. Ý kiến của giảng viên chăm phản biện**

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

*Hải Phòng, ngày ..... tháng..... năm 2019*

**Giảng viên chăm phản biện**

*(Ký và ghi rõ họ tên)*



## MỤC LỤC

LỜI CẢM ƠN .....	5
Chương 1:.....	6
1.1. Android là gì?.....	6
1.2. Mô tả .....	7
1.2.1. Giao diện.....	7
1.2.2. Ứng dụng.....	8
1.3. Phát triển .....	9
1.3.1. Linux .....	9
1.3.2. Quản lý bộ nhớ.....	10
1.3.3. Lịch sử cập nhật .....	11
1.3.4. Cộng đồng mã nguồn mở.....	12
1.4. Bảo mật và tính riêng tư.....	13
1.5. Giấy phép phát hành .....	14
1.6. Đón nhận .....	15
1.6.1. Máy tính bảng .....	16
1.6.2. Thị phần và tỷ lệ sử dụng.....	17
1.6.3. Tỷ lệ sử dụng các phiên bản Android.....	18
1.6.4. Trình trạng ăn cắp bản quyền ứng dụng .....	18
1.7. Bản quyền và bằng phát minh.....	19
1.8. Các thiết bị khác ngoài điện thoại và máy tính bảng.....	20
1.9. Phần mềm gián điệp.....	20
Chương 2:.....	22
2.1. Sơ lược về Android Studio.....	22
2.2. Các thao tác trong môi trường Android Studio .....	22
2.2.1. Cài đặt Android Studio .....	22
2.2.2. Cấu trúc dự án android studio.....	26
2.2.3. Tạo giao diện chương trình trong android studio .....	32

# Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

---

Chương 3: .....	39
3.1. Firebase là gì .....	39
3.2. Cấu trúc dữ liệu .....	40
3.3. Cơ chế hoạt động .....	41
3.3.1 Thêm mới .....	42
3.3.2. Truy vấn xem dữ liệu .....	43
3.3.3. Cập nhật dữ liệu .....	44
3.3.4. Xóa dữ liệu .....	45
3.4. Tích hợp firebase vào ứng dụng .....	45
3.5. Các thao tác dữ liệu cơ bản trên firebase .....	46
Chương 4: .....	68
4.1. Phát biểu bài toán .....	68
4.2.Sơ đồ chức năng .....	68
4.3. Cơ sở dữ liệu .....	69
4.4.Giao diện chương trình .....	69
4.4.1.Màn hình chính .....	69
4.4.2.Màn hình menu .....	70
4.4.3.Màn hình Thêm contact .....	70
4.4.4.Màn hình Xem Danh sách Contact .....	71
4.4.5.Chức năng tìm kiếm .....	71
4.4.6.Sửa contact .....	72
4.4.7.Xóa Contact .....	73
4.4.8.Màn hình giới thiệu .....	74
4.4.9 Thoát ứng dụng .....	74
KẾT LUẬN .....	75
TÀI LIỆU THAM KHẢO .....	76

## LỜI CẢM ƠN

Để đồ án này đạt kết quả tốt đẹp, em đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, tổ chức, cá nhân. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu làm đồ án. Trước hết em xin gửi lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành đồ án: "Tìm hiểu lập trình cơ sở dữ liệu với Android Firebase và ứng dụng". Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới thầy giáo – Ths. Phùng Anh Tuấn đã quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt đồ án này trong thời gian qua. Em xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học Quản Lý Và Công Nghệ Hải Phòng, Phòng Đào Tạo, các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ em trong suốt quá trình học tập tại trường.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không thể tránh được 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, nâng cao ý thức của mình, phục vụ tốt hơn công việc thực tế sau này.

Xin chân thành cảm ơn!

Hải Phòng ,ngày...tháng...năm 20...

Sinh viên

Hoàng Việt Anh

## Chương 1: Giới thiệu về hệ điều hành Android

### 1.1. Android là gì?

**Android** là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào năm 2008.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian OS vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý. Những dự án này bổ sung các tính năng cao cấp cho những người dùng thích tìm tòi hoặc đưa Android vào các thiết bị ban đầu chạy hệ điều hành khác.

Android chiếm 87,7% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 2 năm 2017, với tổng cộng 2 tỷ thiết bị đã được kích hoạt và 1,3 triệu

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là "cuộc chiến điện thoại thông minh" giữa các công ty công nghệ.

### **1.2. Mô tả**

#### **1.2.1. Giao diện**

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng. Những thiết bị phần cứng bên trong như gia tốc kế, con quay hồi chuyển và cảm biến khoảng cách được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị, giống như đang điều khiển vô-lăng.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (*icon*) và tiện ích (*widget*); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẫu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhớ mà không cần phải mở ứng dụng gọi điện ra. Thông báo sẽ luôn nằm đó cho đến khi người dùng đã đọc hoặc xóa nó đi.

### **1.2.2. Ứng dụng**

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK từ trang web khác. Các ứng dụng trên Cửa hàng Play cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Cửa hàng Play được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dùng. Đến tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Cửa hàng Play ước tính đạt 25 tỷ.

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển, gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên QEMU, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốc dành cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

---

Để vượt qua những hạn chế khi tiếp cận các dịch vụ của Google do sự kiểm duyệt Internet tại Cộng hòa Nhân dân Trung Hoa, các thiết bị Android bán tại Trung Quốc lục địa thường được điều chỉnh chỉ được sử dụng dịch vụ đã được duyệt.

### 1.3. Phát triển

*Android được Google tự phát triển riêng cho đến khi những thay đổi và cập nhật đã hoàn thiện, khi đó mã nguồn mới được công khai. Mã nguồn này, nếu không sửa đổi, chỉ chạy trên một số thiết bị, thường là thiết bị thuộc dòng Nexus. Có nhiều thiết bị có chứa những thành phần được giữ bản quyền do nhà sản xuất đặt vào thiết bị Android của họ.*

#### 1.3.1. Linux

Android có một hạt nhân dựa trên nhân Linux phiên bản 2.6, kể từ Android 4.0 *Ice Cream Sandwich* (bánh ngọt kẹp kem) trở về sau, là phiên bản 3.x, với middleware, thư viện và API viết bằng C, còn phần mềm ứng dụng chạy trên một nền tảng ứng dụng gồm các thư viện tương thích với Java dựa trên Apache Harmony. Android sử dụng máy ảo Dalvik với một trình biên dịch động để chạy 'mã dex' (Dalvik Executable) của Dalvik, thường được biên dịch sang Java bytecode. Nền tảng phần cứng chính của Android là kiến trúc ARM. Người ta cũng hỗ trợ x86 thông qua dự án Android x86, và Google TV cũng sử dụng một phiên bản x86 đặc biệt của Android.

Nhân Linux dùng cho Android đã được Google thực hiện nhiều thay đổi về kiến trúc so với nhân Linux gốc. Android không có sẵn X Window System cũng không hỗ trợ các thư viện GNU chuẩn, nên việc chuyển các ứng dụng hoặc thư viện Linux có sẵn sang Android rất khó khăn. Các ứng dụng C đơn giản và SDL cũng được hỗ trợ bằng cách chèn những đoạn shim Java và sử dụng tương tự JNI, như khi người ta chuyển Jagged Alliance 2 sang Android.

Một số tính năng cũng được Google đóng góp ngược vào nhân Linux, đáng chú ý là tính năng quản lý nguồn điện có tên wakelock, nhưng bị những người lập trình chính cho nhân từ chối vì họ cảm thấy Google không có định sẽ tiếp tục bảo trì đoạn mã do họ viết. Google thông báo vào tháng 4 năm 2010 rằng họ sẽ thuê hai nhận viên để làm việc với cộng đồng nhân Linux, nhưng Greg Kroah-Hartman, người bảo trì nhân Linux hiện tại của nhánh ổn định, đã nói vào tháng 12 năm 2010 rằng ông ta lo ngại rằng Google không còn muốn đưa những thay

đôi của mình vào Linux dòng chính nữa. Một số lập trình viên Android của Google tỏ ý rằng "nhóm Android thấy chán với quy trình đó," vì nhóm họ không có nhiều người và có nhiều việc khẩn cấp cần làm với Android hơn.

Vào tháng 8 năm 2011, Linus Torvalds rằng "rốt cuộc thì Android và Linux cũng sẽ trở lại với một bộ nhân chung, nhưng điều đó có thể sẽ không xảy ra trong 4 hoặc 5 năm nữa". Vào tháng 12 năm 2011, Greg Kroah-Hartman thông báo kích hoạt Dự án Dòng chính Android, nhằm tới việc đưa một số driver, bản vá và tính năng của Android ngược vào nhân Linux, bắt đầu từ Linux 3.3. Linux cũng đưa tính năng autosleep (tự nghỉ hoạt động) và wakelocks vào nhân 3.5, sau nhiều nỗ lực phối trộn trước đó. Tương tác thì vẫn vậy nhưng bản hiện thực trên Linux dòng chính cho phép hai chế độ nghỉ: bộ nhớ (dạng nghỉ truyền thống mà Android sử dụng), và đĩa (là ngủ đông trên máy tính để bàn). Việc trộn sẽ hoàn tất kể từ nhân 3.8, Google đã công khai mã nguồn trong đó có những đoạn thử nghiệm đưa Android về lại nhân 3.8.

Bộ lưu trữ flash trên các thiết bị Android được chia thành nhiều phân vùng, như `"/system"` dành cho hệ điều hành và `"/data"` dành cho dữ liệu người dùng và cài đặt ứng dụng. Khác với các bản phân phối Linux cho máy tính để bàn, người sở hữu thiết bị Android không được trao quyền truy cập root vào hệ điều hành và các phân vùng nhạy cảm như `/system` được thiết lập chỉ đọc. Tuy nhiên, quyền truy cập root có thể chiếm được bằng cách tận dụng những lỗ hổng bảo mật trong Android, điều mà cộng đồng mã nguồn mở thường xuyên sử dụng để nâng cao tính năng thiết bị của họ, kể cả bị những người ác ý sử dụng để cài virus và phần mềm ác ý.

Việc Android có được xem là một bản phân phối Linux hay không vẫn còn là vấn đề gây tranh cãi, tuy được Linux Foundation và Chris DiBona, trưởng nhóm mã nguồn mở Google, ủng hộ. Một số khác, như *linux-magazine.com* thì không đồng ý, do Android không hỗ trợ nhiều công cụ GNU, trong đó có glibc.

### 1.3.2. Quản lý bộ nhớ

Vì các thiết bị Android chủ yếu chạy bằng pin, nên Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa tiêu thụ điện năng, trái với hệ điều hành máy tính để bàn luôn cho rằng máy tính sẽ có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động ngưng nó trong bộ nhớ - trong khi ứng dụng về mặt kỹ thuật vẫn "mở", những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào (như năng lượng pin hay năng lượng



xử lý) và nằm đó cho đến khi nó được cần đến. Cách làm như vậy có lợi kép là vừa làm tăng khả năng phản hồi nói chung của thiết bị Android, vì ứng dụng không nhất phải đóng rồi mở lại từ đầu, vừa đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

Android quản lý các ứng dụng trong bộ nhớ một cách tự động: khi bộ nhớ thấp, hệ thống sẽ bắt đầu diệt ứng dụng và tiến trình không hoạt động được một thời gian, sắp theo thời điểm cuối mà chúng được sử dụng (tức là cũ nhất sẽ bị tắt trước). Tiến trình này được thiết kế ẩn đi với người dùng, để người dùng không cần phải quản lý bộ nhớ hoặc tự tay tắt các ứng dụng. Tuy nhiên, sự che giấu này của hệ thống quản lý bộ nhớ Android đã dẫn đến sự thịnh hành của các ứng dụng tắt chương trình của bên thứ ba trên cửa hàng Google Play; những ứng dụng kiểu như vậy được cho là có hại nhiều hơn có lợi.

### 1.3.3. Lịch sử cập nhật

Google đưa ra các bản cập nhật lớn cho Android theo chu kỳ từ 6 đến 9 tháng, mà phần lớn thiết bị đều có thể nhận được qua sóng không dây. Bản cập nhật lớn mới nhất là Android 9.0 Pie.

So với các hệ điều hành cạnh tranh khác, như iOS, các bản cập nhật Android thường mất thời gian lâu hơn để đến với các thiết bị. Với những thiết bị không thuộc dòng Nexus và Pixel, các bản cập nhật thường đến sau vài tháng kể từ khi phiên bản được chính thức phát hành. Nguyên nhân của việc này một phần là do sự phong phú về phần cứng của các thiết bị Android, nên người ta phải mất thời gian điều chỉnh bản cập nhật cho phù hợp, vì mã nguồn chính thức của Google chỉ chạy được trên những thiết bị Nexus chủ lực của họ. Chuyển Android sang những phần cứng cụ thể là một quy trình tốn thời gian và công sức của các nhà sản xuất thiết bị, những người luôn ưu tiên các thiết bị mới nhất và thường bỏ rơi các thiết bị cũ hơn. Do đó, những chiếc điện thoại thông minh thế hệ cũ thường không được cập nhật nếu nhà sản xuất quyết định rằng nó không đáng để bỏ thời gian, bất kể chiếc điện thoại đó có khả năng chạy bản cập nhật hay không. Vấn đề này còn trầm trọng hơn khi những nhà sản xuất điều chỉnh Android để đưa giao diện và ứng dụng của họ vào, những thứ này cũng sẽ phải làm lại cho mỗi bản cập nhật. Sự chậm trễ còn được đóng góp bởi nhà mạng, sau khi nhận được bản cập nhật từ nhà sản xuất, họ còn điều chỉnh thêm cho phù hợp với nhu cầu rồi thử nghiệm kỹ lưỡng trên hệ thống mạng của họ trước khi chuyển nó đến người dùng.

Việc thiếu các hỗ trợ hậu mãi của nhà sản xuất và nhà mạng đã bị những nhóm người dùng và các trang tin công nghệ chỉ trích rất nhiều. Một số người viết còn nói rằng giới công nghiệp do cái lợi về tài chính đã cố tình *không* cập nhật thiết bị, vì nếu thiết bị hiện tại không cập nhật sẽ thúc đẩy việc mua thiết bị mới, một thái độ được coi là "xúc phạm". *The Guardian* đã than phiền rằng phương cách phân phối bản cập nhật trở nên phức tạp chính vì những nhà sản xuất và nhà mạng đã cố tình làm nó như thế. Vào năm 2011, Google đã hợp tác cùng một số hãng công nghiệp và ra mắt "Liên minh Cập nhật Android", với lời hứa sẽ cập nhật thường xuyên cho các thiết bị trong vòng 18 tháng sau khi ra mắt. Tính đến năm 2012, người ta không còn nghe nhắc đến liên minh này nữa.

### 1.3.4. Cộng đồng mã nguồn mở

Android có một cộng đồng các lập trình viên và những người đam mê rất năng động. Họ sử dụng mã nguồn Android để phát triển và phân phối những phiên bản chỉnh sửa của hệ điều hành. Các bản Android do cộng đồng phát triển thường đem những tính năng và cập nhật mới vào nhanh hơn các kênh chính thức của nhà sản xuất/nhà mạng, tuy không được kiểm thử kỹ lưỡng cũng như không có đảm bảo chất lượng; cung cấp sự hỗ trợ liên tục cho các thiết bị cũ không còn nhận được bản cập nhật chính thức; hoặc mang Android vào những thiết bị ban đầu chạy một hệ điều hành khác, như HP Touchpad. Các bản Android của cộng đồng thường được root sẵn và có những điều chỉnh không phù hợp với những người dùng không rành rẽ, như khả năng ép xung hoặc tăng/giảm áp bộ xử lý của thiết bị. CyanogenMod là firmware của cộng đồng được sử dụng phổ biến nhất, và hoạt động như một tổ chức của số đông khác.

Trước đây, nhà sản xuất thiết bị và nhà mạng tỏ ra thiếu thiện chí với việc phát triển firmware của bên thứ ba. Những nhà sản xuất còn thể hiện lo ngại rằng các thiết bị chạy phần mềm không chính thức sẽ hoạt động không tốt và dẫn đến tổn tiền hỗ trợ. Hơn nữa, các firmware đã thay đổi như CyanogenMod đôi khi còn cung cấp những tính năng, như truyền tải mạng (tethering), mà người dùng bình thường phải trả tiền nhà mạng mới được sử dụng. Kết quả là nhiều thiết bị bắt đầu đặt ra hàng rào kỹ thuật như khóa bootloader hay hạn chế quyền truy cập root. Tuy nhiên, khi phần mềm do cộng đồng phát triển ngày càng trở nên phổ biến, và sau một thông cáo của Thư viện Quốc hội Hoa Kỳ cho phép "jailbreak" (vượt ngục) thiết bị di động, các nhà sản xuất và nhà mạng đã tỏ ra mềm mỏng hơn với các nhà phát triển thứ ba, thậm chí một số hãng như HTC, Motorola, Samsung và Sony, còn hỗ trợ và khuyến khích phát triển. Kết quả của việc này

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

là dần dần nhu cầu tìm ra các hạn chế phần cứng để cài đặt được firmware không chính thức đã bớt đi do ngày càng nhiều thiết bị được phát hành với bootloader đã mở khóa sẵn hoặc *có thể* mở khóa, tương tự như điện thoại dòng Nexus, tuy rằng thông thường họ sẽ yêu cầu người dùng từ bỏ chế độ bảo hành nếu họ làm như vậy. Tuy nhiên, tuy được sự chấp thuận của nhà sản xuất, một số nhà mạng tại Mỹ vẫn bắt buộc điện thoại phải bị khóa.

Việc mở khóa và "hack" điện thoại thông minh và máy tính bảng vẫn còn là tác nhân gây căng thẳng giữa cộng đồng và công nghiệp. Cộng đồng luôn biện hộ rằng sự hỗ trợ không chính thức ngày càng trở nên quan trọng trước việc nền công nghiệp không cung cấp các bản cập nhật thường xuyên và/hoặc ngưng hỗ trợ cho chính các thiết bị của họ.

### **1.4. Bảo mật và tính riêng tư**

Các ứng dụng Android chạy trong một "hộp cát", là một khu vực riêng rẽ với hệ thống và không được tiếp cận đến phần còn lại của tài nguyên hệ thống, trừ khi nó được người dùng trao quyền truy cập một cách công khai khi cài đặt. Trước khi cài đặt ứng dụng, Cửa hàng Play sẽ hiển thị tất cả các quyền mà ứng dụng đòi hỏi: ví dụ như một trò chơi cần phải kích hoạt bộ rung hoặc lưu dữ liệu vào thẻ nhớ SD, nhưng nó không nên cần quyền đọc tin nhắn SMS hoặc tiếp cận danh bạ điện thoại. Sau khi xem xét các quyền này, người dùng có thể chọn đồng ý hoặc từ chối chúng, ứng dụng chỉ được cài đặt khi người dùng đồng ý.

Hệ thống hộp cát và hỏi quyền làm giảm bớt ảnh hưởng của lỗi bảo mật hoặc lỗi chương trình có trong ứng dụng, nhưng sự bối rối của lập trình viên và tài liệu hướng dẫn còn hạn chế đã dẫn tới những ứng dụng hay đòi hỏi những quyền không cần thiết, do đó làm giảm đi hiệu quả của hệ thống này. Một số công ty bảo mật, như Lookout Mobile Security, AVG Technologies, và McAfee, đã phát hành những phần mềm diệt virus cho các thiết bị Android. Phần mềm này không có hiệu quả vì cơ chế hộp cát vẫn áp dụng vào các ứng dụng này, do vậy làm hạn chế khả năng quét sâu vào hệ thống để tìm nguy cơ.

Một nghiên cứu của công ty bảo mật Trend Micro đã liệt kê tình trạng lạm dụng dịch vụ trả tiền là hình thức phần mềm ác ý phổ biến nhất trên Android, trong đó tin nhắn SMS sẽ bị gửi đi từ điện thoại bị nhiễm đến một số điện thoại trả tiền mà người dùng không hề hay biết. Loại phần mềm ác ý khác hiển thị những quảng cáo không mong muốn và gây khó chịu trên thiết bị, hoặc gửi thông tin cá nhân đến bên thứ ba khi chưa được phép. Đe dọa bảo mật trên Android được

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

---

cho là tăng rất nhanh theo cấp số mũ; tuy nhiên, các kỹ sư Google phản bác rằng hiểm họa từ phần mềm ác ý và virus đã bị thổi phồng bởi các công ty bảo mật nhằm mục đích thương mại, và buộc tội ngành công nghiệp bảo mật đang lợi dụng sự sợ hãi để bán phần mềm diệt virus cho người dùng. Google vẫn giữ quan điểm rằng phần mềm ác ý thật sự nguy hiểm là cực kỳ hiếm, và một cuộc điều tra do F-Secure thực hiện cho thấy chỉ có 0,5% số phần mềm ác ý Android là len vào được cửa hàng Google Play.

Google hiện đang sử dụng bộ quét phần mềm ác ý Google Bouncer để theo dõi và quét các ứng dụng trên Cửa hàng Google Play. Nó sẽ đánh dấu các phần mềm bị nghi ngờ và cảnh báo người dùng về những vấn đề có thể xảy ra trước khi họ tải nó về máy. Android phiên bản 4.2 Jelly Bean được phát hành vào năm 2012 cùng với các tính năng bảo mật được cải thiện, bao gồm một bộ quét phần mềm ác ý được cài sẵn trong hệ thống, hoạt động cùng với Google Play nhưng cũng có thể quét các ứng dụng được cài đặt từ nguồn thứ ba, và một hệ thống cảnh báo sẽ thông báo cho người dùng khi một ứng dụng cố gắng gửi một tin nhắn vào số tính tiền, chặn tin nhắn đó lại trừ khi người dùng công khai cho phép nó.

Điện thoại thông minh Android có khả năng báo cáo vị trí của điểm truy cập Wi-Fi, phát hiện ra việc di chuyển của người dùng điện thoại, để xây dựng những cơ sở dữ liệu có chứa vị trí của hàng trăm triệu điểm truy cập. Những cơ sở dữ liệu này tạo nên một bản đồ điện tử để tìm vị trí điện thoại thông minh, cho phép chúng chạy các ứng dụng như Foursquare, Google Latitude, Facebook Places, và gửi những đoạn quảng cáo dựa trên vị trí. Phần mềm theo dõi của bên thứ ba như TaintDroid, một dự án nghiên cứu trong trường đại học, đôi khi có thể biết được khi nào thông tin cá nhân bị gửi đi từ ứng dụng đến các máy chủ đặt ở xa.

Bản chất mã nguồn mở của Android cho phép những nhà thầu bảo mật lấy những thiết bị sẵn có rồi điều chỉnh để sử dụng ở mức độ bảo mật cao hơn. Ví dụ như Samsung đã cộng tác với General Dynamics sau khi họ thu tóm Open Kernel Labs để xây dựng lại Jellybean trên nền bộ vi kiểm soát dành cho dự án "Knox".

### 1.5. Giấy phép phát hành

Mã nguồn của Android được cấp phép theo các giấy phép phần mềm mã nguồn mở tự do. Google đưa phần lớn mã nguồn (bao gồm cả các lớp mạng và điện thoại) theo Giấy phép Apache phiên bản 2.0, và phần còn lại, các thay đổi đối với nhân Linux, theo Giấy phép Công cộng GNU phiên bản 2. Liên minh Thiết

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

bị cầm tay mở đã thực hiện các thay đổi trên nhân Linux, với mã nguồn lúc nào cũng công khai. Phần còn lại của Android được Google phát triển một mình, và mã nguồn chỉ được công bố khi phát hành một phiên bản mới. Thông thường Google cộng tác với một nhà sản xuất phần cứng để cung cấp một thiết bị 'chủ lực' (thuộc dòng Google Nexus) với phiên bản mới nhất của Android, sau đó phát hành mã nguồn sau khi thiết bị này được bán ra.

Vào đầu năm 2011, Google quyết định tạm ngưng phát hành mã nguồn Android phiên bản 3.0 *Honeycomb* dành riêng cho máy tính bảng. Lý do, theo Andy Rubin trong một bài blog Android chính thức, là vì *Honeycomb* đã được làm gấp gáp để phục vụ cho Motorola Xoom, và họ không muốn các bên thứ ba tạo ra một "trải nghiệm người dùng cực kỳ tồi tệ" bằng cách cố gắng đưa vào điện thoại thông minh một phiên bản dành riêng cho máy tính bảng. Mã nguồn một lần nữa được xuất bản công khai vào tháng 11 năm 2011 với sự ra mắt của Android 4.0.

Mặc dù phần mềm là mã nguồn mở, các nhà sản xuất thiết bị không thể sử dụng thương hiệu Android của Google trừ khi Google chứng nhận rằng thiết bị của họ phù hợp với Tài liệu Định nghĩa Tương thích (*Compatibility Definition Document* - CDD). Các thiết bị cũng phải thỏa mãn định nghĩa này thì mới được cấp phép để cài các ứng dụng mã nguồn đóng của Google, gồm cả Google Play. Vì Android không hoàn toàn được phát hành theo giấy phép tương thích GPL, ví dụ như mã nguồn của Google là theo giấy phép Apache license, và cũng vì Google Play cho phép các phần mềm có bản quyền, Richard Stallman và Quỹ phần mềm tự do luôn chỉ trích Android và khuyên người dùng sử dụng hệ điều hành khác như Replicant.

### **1.6. Đón nhận**

Android được đón nhận bằng một thái độ thờ ơ khi nó ra mắt vào năm 2007. Mặc dù những nhà phân tích rất ấn tượng với việc các công ty công nghệ có tiếng tạm hợp tác cùng Google để tạo ra Liên minh thiết bị di động mở, người ta vẫn không rõ liệu các nhà sản xuất có sẵn sàng thay thế hệ điều hành mà họ đang dùng bằng Android hay không. Ý tưởng về một nền tảng phát triển mã nguồn mở dựa trên Linux đã thu hút sự quan tâm, nhưng cũng đẩy lên những lo ngại rằng Android sẽ phải đối mặt với sự cạnh tranh mạnh mẽ từ những tay chơi có hạng trong thị trường điện thoại thông minh, như Nokia và Microsoft, và các hệ điều hành di động đối thủ cũng sử dụng Linux đang trong quá trình phát triển.

Những công ty hàng đầu không giấu sự hoài nghi: Nokia được trích nói rằng "chúng tôi không xem đó là một sự đe dọa," và một thành viên của nhóm Windows Mobile của Microsoft nói rằng "tôi không hiểu rồi họ sẽ có tác động ra sao."

Kể từ đó Android đã phát triển để trở thành hệ điều hành dành cho điện thoại thông minh phổ biến nhất trên thế giới và là "một trong những trải nghiệm di động nhanh nhất hiện nay." Các nhà bình luận thì nhấn mạnh vào bản chất mã nguồn mở của hệ điều hành chính là một trong những yếu tố quyết định sức mạnh, cho phép các công ty như (Kindle Fire), Barnes & Noble (Nook), Ouya, Baidu, và những hãng khác đổi hướng phần mềm và phát hành những phần cứng chạy trên phiên bản Android đã thay đổi của riêng họ. Kết quả, nó được trang web công nghệ Ars Technica mô tả là "đương nhiên là hệ điều hành mặc định khi phát hành phần cứng mới" cho những công ty không có nền tảng di động riêng của họ. Chính sự mở và uyển chuyển này cũng hiện diện ở cấp độ người dùng cuối: Android cho phép người dùng điện thoại điều chỉnh thoải mái thiết bị của họ và ứng dụng thì có sẵn trên các cửa hàng ứng dụng và trang web không phải của Google. Những đặc điểm này được xem là đóng góp vào những thế mạnh chính của điện thoại Android so với các điện thoại khác.

Android cũng bị phê phán vì thiếu sự hỗ trợ hậu mãi từ nhà sản xuất và nhà mạng, nếu so sánh với iOS của Apple. Với những thiết bị không mang nhãn hiệu Nexus, nhà mạng luôn kiểm tra các tiêu chuẩn của họ rồi thực hiện thay đổi cho riêng từng thiết bị (bắt nguồn từ sự điều chỉnh của nhà sản xuất và sự đa dạng của thiết bị Android) được xem là tác nhân chính trì hoãn việc cập nhật. Những nhà bình luận cũng nói rằng ngành công nghiệp thiết bị di động vì lý do lợi nhuận đã cố tình không cập nhật thiết bị của họ, vì thiếu cập nhật trên thiết bị hiện tại sẽ thúc đẩy việc mua thiết bị mới.

### 1.6.1. Máy tính bảng

Mặc dù thành công với điện thoại thông minh, việc sử dụng máy tính bảng Android vẫn còn chậm.

Một trong những nguyên nhân chính là tình huống con gà và quả trứng trong đó người tiêu dùng ngại mua máy tính bảng Android cho thiếu các ứng dụng máy tính bảng chất lượng cao, còn các lập trình viên thì ngại mất thời gian và tiền bạc để phát triển ứng dụng máy tính bảng cho đến khi nào thị phần của chúng đủ

lớn. Nội dung và "hệ sinh thái" ứng dụng đã chứng tỏ rằng nó quan trọng hơn nhiều so với việc "nạp vào chạy" (sức mạnh xử lý phần cứng) khi nói đến máy tính bảng. Do thiếu các ứng dụng dành riêng cho máy tính bảng vào năm 2011, các máy tính bảng Android đời đầu đã phải sử dụng các ứng dụng sẵn có trên điện thoại thông minh dù hiển thị rất kém trên màn hình cỡ lớn, trong khi sự thống trị của Apple iPad được củng cố bởi một số lượng lớn ứng dụng iOS dành riêng cho máy tính bảng.

Mặc dù sự hỗ trợ từ ứng dụng chỉ mới ở mức sơ khai, một lượng đáng kể máy tính bảng Android (cùng với các loại máy tính bảng sử dụng các hệ điều hành khác, như HP TouchPad và BlackBerry Playbook) vẫn được tung ra thị trường trong nỗ lực cạnh tranh với sự thành công của iPad. *InfoWorld* đã nói rằng một số nhà sản xuất Android thoát đầu xem các máy tính bảng của họ như là một "thương vụ Frankenphone", một cơ hội đầu tư thấp ngắn hạn bằng cách đặt một hệ điều hành Android tối ưu cho điện thoại thông minh (trước khi Android 3.0 "Honeycomb" dành cho máy tính bảng ra đời) trên một thiết bị mà không để ý tới giao diện người dùng. Cách làm này, như với Dell Streak, không những thất bại trong việc lôi kéo người dùng mà còn làm ảnh hưởng đến danh tiếng ban đầu của máy tính Android. Hơn nữa, một số máy tính bảng Android như Motorola Xoom được định giá bằng hoặc cao hơn iPad, làm tổn hại sức bán. Một ngoại lệ đó là Amazon Kindle Fire, được phát triển theo cách tiếp cận "chờ mà xem" dựa trên giá rẻ và khả năng truy cập vào hệ sinh thái ứng dụng và nội dung của Amazon.com.

Hiện tượng này bắt đầu thay đổi vào năm 2012 với sự ra mắt của Nexus 7 giá rẻ và một cú hích của Google dành cho các lập trình viên nhằm thúc đẩy họ viết các ứng dụng cho máy tính bảng tốt hơn. Máy tính bảng Android được kỳ vọng sẽ vượt qua iPad trong vòng một vài năm.

### **1.6.2. Thị phần và tỷ lệ sử dụng**

Công ty nghiên cứu thị trường Canalys đã ước tính trong quý 2 năm 2009 rằng Android có 2,8% thị phần điện thoại thông minh được bán ra toàn cầu. Đến quý 4 năm 2010 con số này tăng lên 33% thị phần, trở thành nền tảng điện thoại thông minh bán chạy hàng đầu. Đến quý 3 năm 2011 Gartner ước tính rằng hơn một nửa (52,5%) thị trường điện thoại thông minh thuộc về Android. Đến quý 3 năm 2012 Android đã có 75% thị phần điện thoại thông minh toàn cầu theo nghiên cứu của hãng IDC.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Vào tháng 7 năm 2011, Google nói rằng có 550.000 thiết bị Android mới được kích hoạt mỗi ngày, đỉnh điểm là 400.000 máy một ngày vào tháng 5, và có hơn 100 triệu thiết bị đã được kích hoạt với mức tăng 4,4% mỗi tuần. Vào tháng 9 năm 2012, 500 triệu thiết bị đã được kích hoạt với 1,3 triệu lượt kích hoạt mỗi ngày.

Thị phần của Android có khác nhau theo khu vực. Vào tháng 7 năm 2012, thị phần Android tại Mỹ là 52%, nhưng lên tới 90% tại Trung Quốc.

### **1.6.3. Tỷ lệ sử dụng các phiên bản Android**

Công ty nghiên cứu thị trường Canalys đã ước tính trong quý 2 năm 2009 rằng Android có 2,8% thị phần điện thoại thông minh được bán ra toàn cầu. Đến quý 4 năm 2010 con số này tăng lên 33% thị phần, trở thành nền tảng điện thoại thông minh bán chạy hàng đầu. Đến quý 3 năm 2011 Gartner ước tính rằng hơn một nửa (52,5%) thị trường điện thoại thông minh thuộc về Android. Đến quý 3 năm 2012 Android đã có 75% thị phần điện thoại thông minh toàn cầu theo nghiên cứu của hãng IDC.

Vào tháng 7 năm 2011, Google nói rằng có 550.000 thiết bị Android mới được kích hoạt mỗi ngày, đỉnh điểm là 400.000 máy một ngày vào tháng 5, và có hơn 100 triệu thiết bị đã được kích hoạt với mức tăng 4,4% mỗi tuần. Vào tháng 9 năm 2012, 500 triệu thiết bị đã được kích hoạt với 1,3 triệu lượt kích hoạt mỗi ngày.

Thị phần của Android có khác nhau theo khu vực. Vào tháng 7 năm 2012, thị phần Android tại Mỹ là 52%, nhưng lên tới 90% tại Trung Quốc.

### **1.6.4. Trình trạng ăn cắp bản quyền ứng dụng**

Đã có những lo ngại về việc các ứng dụng trả tiền của Android quá dễ bị ăn cắp. Trong một cuộc phỏng vấn vào tháng 5 năm 2012 với Eurogamer, nhà phát triển Football Manager nói rằng tỷ lệ người chơi ăn cắp so với người chơi trả tiền là 9:1 với trò chơi Football Manager Handheld. Tuy nhiên, không phải tất cả các lập trình viên đều cho rằng tình trạng ăn cắp là một vấn đề; ví dụ như vào tháng 7 năm 2012 các lập trình viên của trò chơi Wind-up Knight nói rằng mức độ ăn cắp trò chơi của họ chỉ khoảng 12%, và phần lớn sản phẩm ăn cắp đến từ Trung Quốc, nơi người ta không thể mua ứng dụng từ Google Play.

Vào năm 2010, Google phát hành một công cụ để xác nhận việc mua bán để sử dụng trong các ứng dụng, nhưng các lập trình viên than phiền rằng như vậy là



chưa đủ và quá dễ để bẻ khóa. Google trả lời rằng công cụ, đặc biệt là bản phát hành đầu tiên, chỉ có ý định làm nền tảng mẫu cho lập trình viên điều chỉnh và xây dựng theo yêu cầu, chứ không phải một giải pháp bảo mật hoàn chỉnh. Vào năm 2012 Google phát hành một tính năng trong Android 4.1 để mã hóa các ứng dụng trả tiền chỉ hoạt động trên thiết bị đã mua ứng dụng đó, nhưng tính năng này đã bị hoãn do vấn đề về kỹ thuật.

### 1.7 .Bản quyền và bằng phát minh

Cả Android và nhà sản xuất điện thoại Android đều bị dính líu đến nhiều vụ kiện tụng về bằng phát minh. Ngày 12 tháng năm 2010, Oracle kiện Google với cáo buộc vi phạm bản quyền và bằng phát minh liên quan đến ngôn ngữ lập trình Java. Oracle ban đầu muốn được đền bù thiệt hại 6,1 tỷ đô la Mỹ, nhưng bị tòa án liên bang Mỹ khước từ mức giá này và yêu cầu Oracle xem xét lại. Để đáp lại, Google đã đưa ra nhiều lý lẽ để bảo vệ, tuyên bố ngược lại là Android không hề vi phạm bằng phát minh hay bản quyền của Oracle, và rằng bằng phát minh của Oracle là vô hiệu, cùng một số lời bào chữa khác. Google nói rằng Android dựa trên Apache Harmony, một hiện thực phòng sạch của thư viện lớp Java (tức là xem hoạt động của thư viện, rồi lập trình lại bắt chước hoạt động đó nhưng không tham khảo hoặc lấy lại mã nguồn của thư viện gốc), rồi sau đó độc lập phát triển ra máy ảo đặt tên là Dalvik. Vào tháng 5 năm 2012 bồi thẩm đoàn của vụ án tuyên rằng Google không vi phạm bằng phát minh của Oracle, và sau đó thẩm phán tuyên rằng cấu trúc của Java API do Google sử dụng không đủ để được giữ bản quyền.

Ngoài vụ kiện trực tiếp chống lại Google, có nhiều cuộc chiến tranh thế mạng khác nhau gián tiếp chống lại Android bằng cách nhắm vào các nhà sản xuất thiết bị Android, nhằm làm nản lòng những nhà sản xuất muốn sử dụng nền tảng này do sự tăng chi phí để đưa thiết bị Android ra thị trường. Cả Apple và Microsoft đều đã kiện một số nhà sản xuất vì vi phạm bằng sáng chế, với cuộc chiến pháp lý chống Samsung dằng dai của Apple là vụ nổi bật nhất. Vào tháng 10 năm 2011 Microsoft nói rằng họ đã ký một số thỏa thuận cấp phép với 10 nhà sản xuất thiết bị Android, những hãng sản xuất 55% lợi nhuận toàn cầu của Android. Những công ty này có cả Samsung lẫn HTC. Vụ dàn xếp bằng phát minh của Samsung với Microsoft có một thỏa thuận rằng Samsung sẽ cung cấp thêm nguồn lực để phát triển và tiếp thị điện thoại chạy hệ điều hành Windows Phone của Microsoft.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Google đã bày tỏ công khai sự bất mãn của họ đối với hệ thống bằng phát minh tại Mỹ, buộc tội Apple, Oracle và Microsoft cố tình đim Android thông qua các vụ kiện, thay vì phải sáng tạo và cạnh tranh bằng các sản phẩm và dịch vụ tốt hơn. Vào năm 2011-12, Google mua lại Motorola Mobility với giá 12,5 tỷ đô la Mỹ, một hành động được xem là phương cách để bảo vệ Android, vì Motorola Mobility nắm giữ hơn 17.000 bằng phát minh. Tháng 12 năm 2011 Google mua lại hơn một nghìn bằng sáng chế từ IBM.

### **1.8. Các thiết bị khác ngoài điện thoại và máy tính bảng**

Bản chất mở và cho phép thay đổi của Android giúp nó xuất hiện trên các thiết bị điện tử khác, như laptop và netbook, smartbook TV thông minh (Google TV) và máy ảnh (Nikon Coolpix S800c và Galaxy Camera). Hơn thế nữa, hệ điều hành Android còn được ứng dụng trong kính mắt thông minh (Project Glass), đồng hồ đeo tay, tai nghe, đầu CD và DVD cho xe hơi, gương soi, máy nghe nhạc bỏ túi và điện thoại để bàn và VoIP. Ouya, một máy trò chơi điện tử chạy Android, đã trở thành một trong những chiến dịch khởi động thành công nhất, gây quỹ được 8,5 triệu đô la Mỹ để phát triển, tiếp sau đó là các máy trò chơi điện tử dựa trên Android như Project Shield của NVIDIA.

Vào năm 2011, Google đã trình diễn "Android@Home", một công nghệ tự động hóa gia đình, sử dụng Android để điều khiển nhiều thiết bị gia dụng như công tắc điện, ổ cắm và thiết bị điều khiển nhiệt độ trong nhà. Chiếc đèn mẫu được quảng cáo là có thể được điều khiển từ điện thoại hoặc máy tính bảng Android, nhưng trưởng nhóm Android Andy Rubin vẫn cảnh trọng cho rằng "tắt mở bóng đèn không phải là việc gì mới," ám chỉ nhiều dịch vụ tự động hóa gia đình đã gặp thất bại trước đây. Ông nói rằng Google có suy nghĩ tham vọng hơn và dự định của công ty là sử dụng vị trí của mình như một nhà cung cấp dịch vụ đám mây để mang sản phẩm Google đến gia đình của khách hàng.

### **1.9. Phần mềm gián điệp**

Theo nhật báo New York Times, một phần mềm gián điệp cài trên một số điện thoại Android "Made in China" không chỉ đánh cắp tin nhắn cá nhân của người dùng, chúng còn theo dõi mọi chuyển động cũng như tắt cả số điện thoại họ liên lạc. Một chiếc điện thoại có phần mềm này gửi tin nhắn về một máy chủ ở thành phố Thượng Hải đăng ký bởi Adups. Một nhà sản xuất điện thoại của Mỹ - BLU Products – cho biết 120.000 điện thoại của họ bị ảnh hưởng và công ty đã phải

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

cập nhật lại firmware (phần mềm hệ thống) để loại bỏ tính năng do thám. Số lượng các thiết bị nhiễm mã độc trên toàn cầu hiện chưa thống kê được nhưng theo số liệu của Adups Technology, trên thế giới có hơn 700 triệu điện thoại, xe hơi và các thiết bị thông minh cài phần mềm của họ. Adups là đối tác cung cấp phần mềm cho hai nhà sản xuất điện thoại di động lớn nhất thế giới là ZTE và Huawei có trụ sở tại Trung Quốc.

## Chương 2: Môi trường lập trình Android Studio

### 2.1.Sơ lược về Android Studio

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android studio dựa vào IntelliJ IDEA, là một IDE tốt cho nhất Java hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho Android.

### 2.2. Các thao tác trong môi trường Android Studio

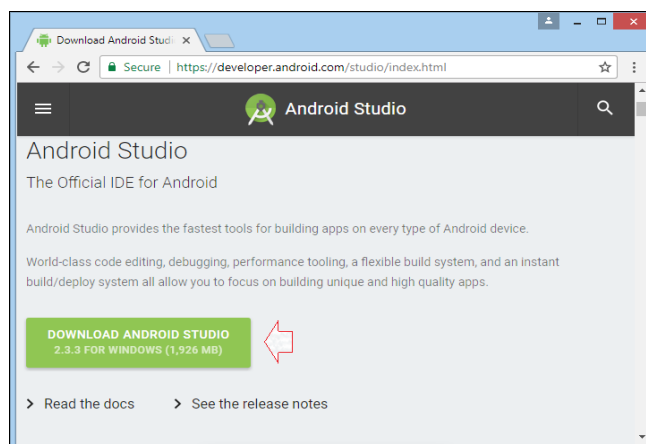
#### 2.2.1.Cài đặt Android Studio

##### a.Yêu cầu phần cứng máy tính

- Microsoft® Windows® 8/7 (32 or 64-bit)
- 4 GB RAM. (Tốt nhất là 8GB)
- 400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator system images và caches
- Độ phân giải tối thiểu 1280 x 800
- Java Development Kit

##### b.Phần mềm Android Studio

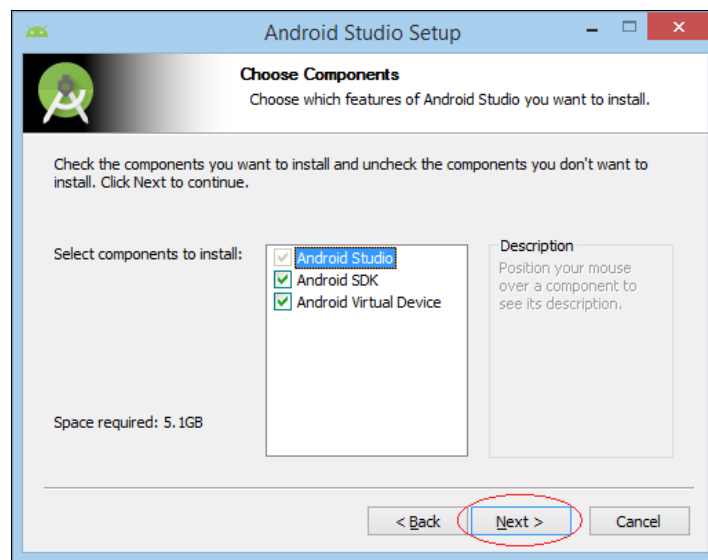
- Vào đường dẫn: “<http://developer.android.com/sdk/index.html>”
- Để download bản mới nhất và tiến hành cài đặt click như hình:



Hình 2.1:Phần tải Android Studio

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

- Khi cài đặt chú ý chọn cả SDK và trình giả lập thiết bị android ảo như hình:



Hình 2.2: Chọn thiết bị giả lập và SDK

- Tiếp tục chọn next và agree cho đến khi hoàn tất
- Đây là màn hình khởi động.



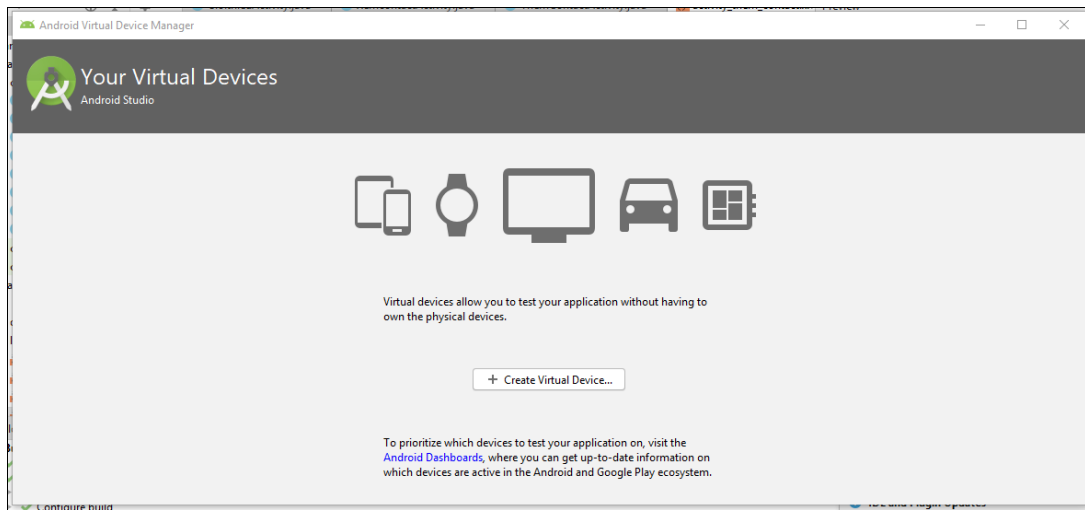
Hình 2.3: Màn hình khởi động

- Các bước cài đặt thiết bị ảo trong android studio

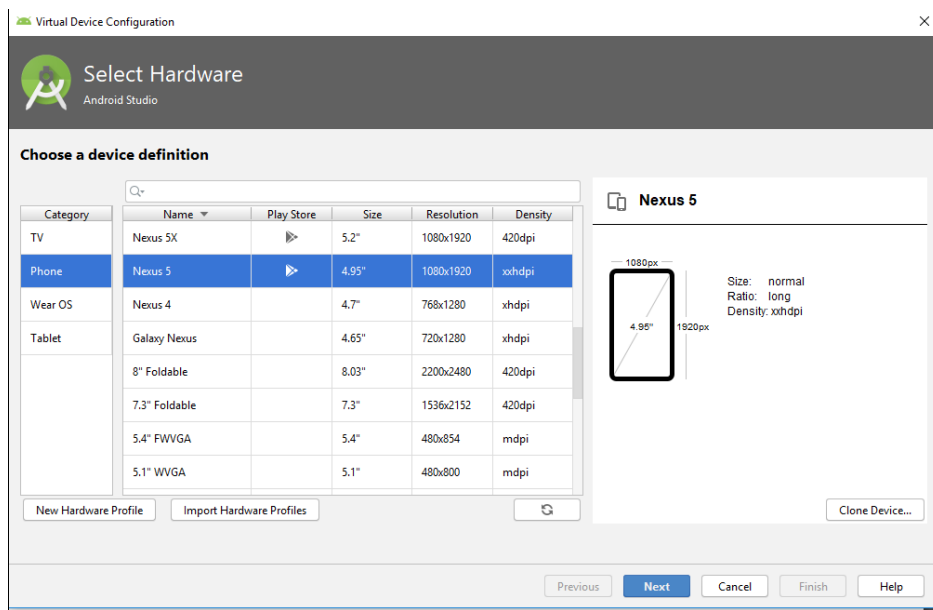
Máy ảo Android là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Trong Android Studio có cung cấp cho chúng ta một máy ảo Android mặc định là Android Virtual Device viết tắt là AVD.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Để cài đặt máy ảo mở Android Studio lên và click vào nút AVD Manager. Cửa sổ AVD Manager sẽ xuất hiện:

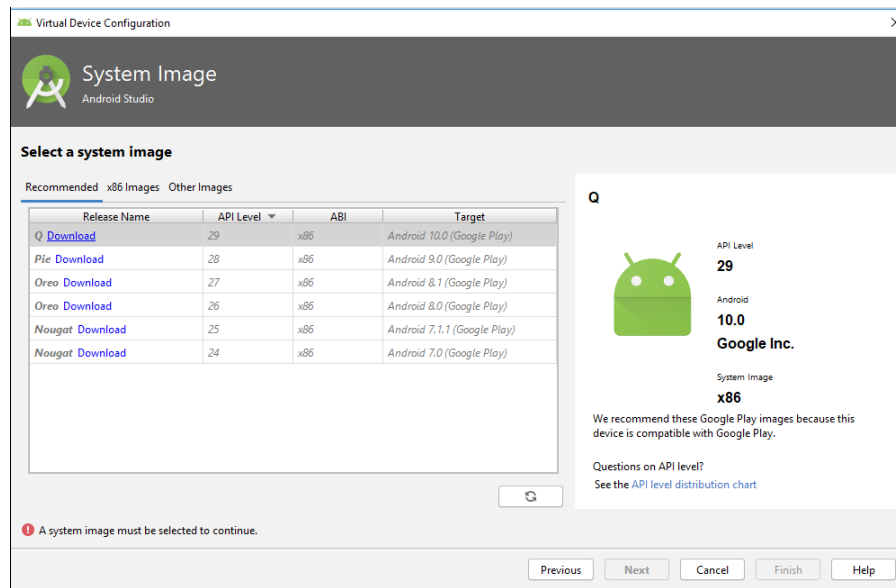


Hình 2.4: Cửa sổ AVD Manager

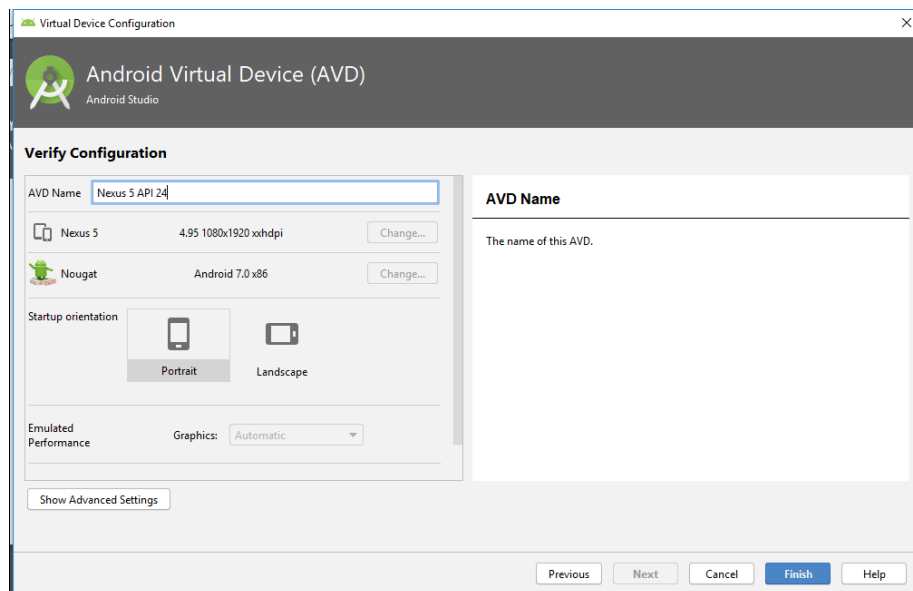


Hình 2.5: Chọn máy ảo phù hợp

Ở đây có sẵn các mẫu điện thoại, ta chọn mẫu mà mình muốn, ví dụ chọn Nexus 5. Chọn xong bấm "Next" để tiếp tục:

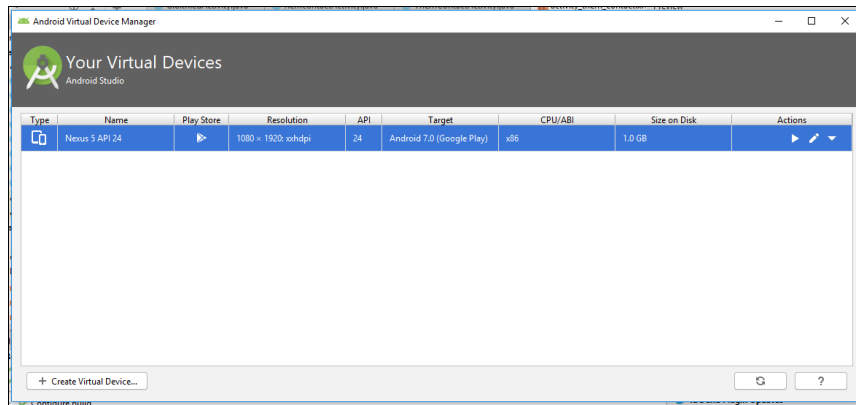


Hình 2.7: Chọn phiên bản cài đặt



Hình 2.8: Thiết lập cấu hình máy ảo

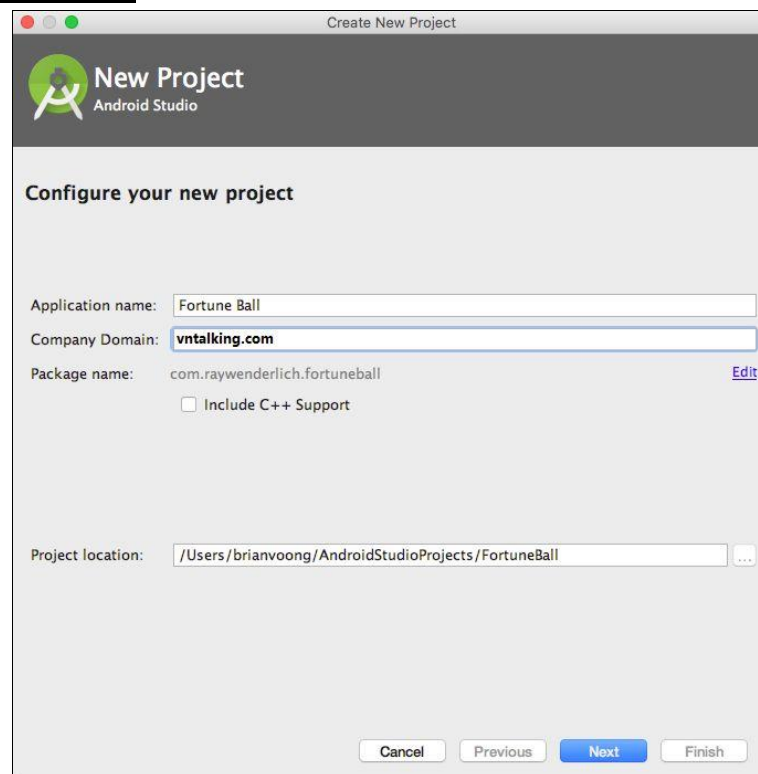
Sau khi tùy chỉnh theo ý muốn ấn “Finish” để tiến hành tạo máy ảo và chờ một lúc để Android studio lưu thông tin máy ảo. Sau khi lưu xong ta sẽ thấy trong cửa sổ AVD Manager có thêm một cái máy ảo nữa:



Hình 2.9: Cửa sổ AVD Manager có thêm một cái máy ảo

## 2.2.2. Cấu trúc dự án android studio

### a. Tạo mới một project.



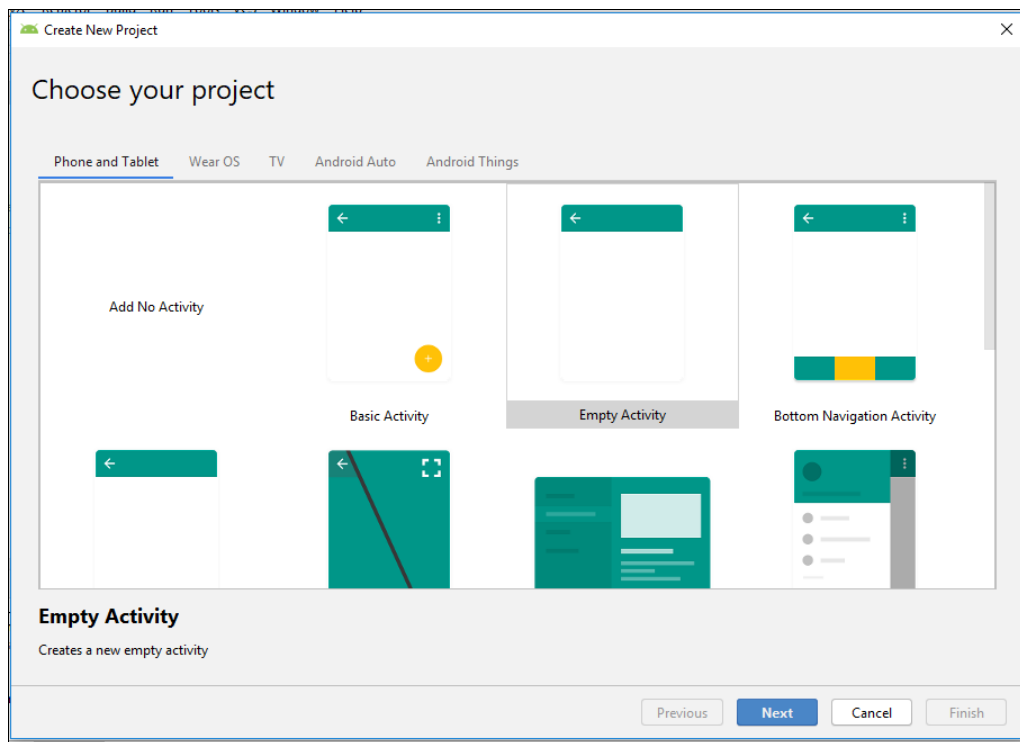
Hình 2.10: Màn hình tạo mới một project

- **Application Name:** Tên ứng dụng muốn đặt
- **Company Domain:** Tên domain công ty, thường được dùng để kết hợp với tên Application để tạo thành Package (chú ý viết thường hết và có ít nhất 1 dấu chấm).
- **Package name:** Nó sẽ tự động nối ngược Company Domain với Application name.



## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

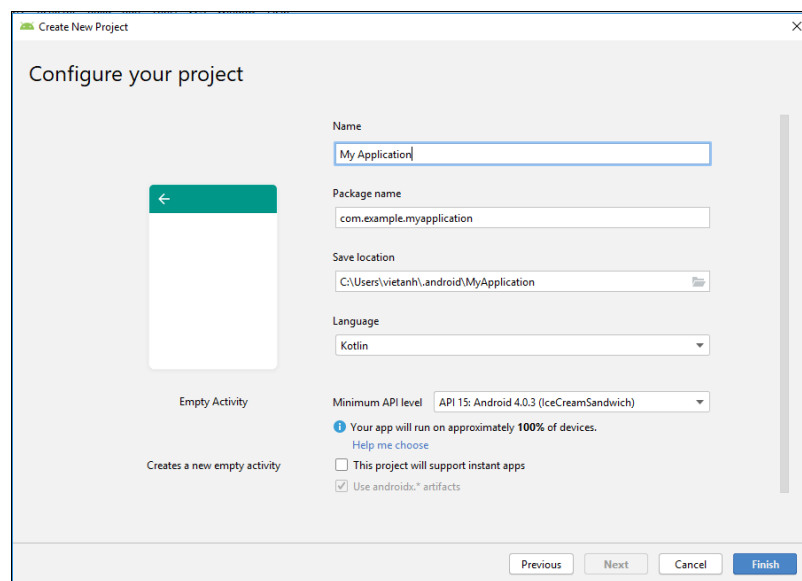
- **Project location:** Là nơi lưu trữ ứng dụng. Sau đó nhấn Next để tiếp tục.



Hình 2.11: Màn hình lựa chọn kiểu Activity

Ở hộp thoại trên cho phép ta lựa chọn là ứng dụng sẽ được viết cho những thiết bị nào (Phone and Tablet, TV, Wear).

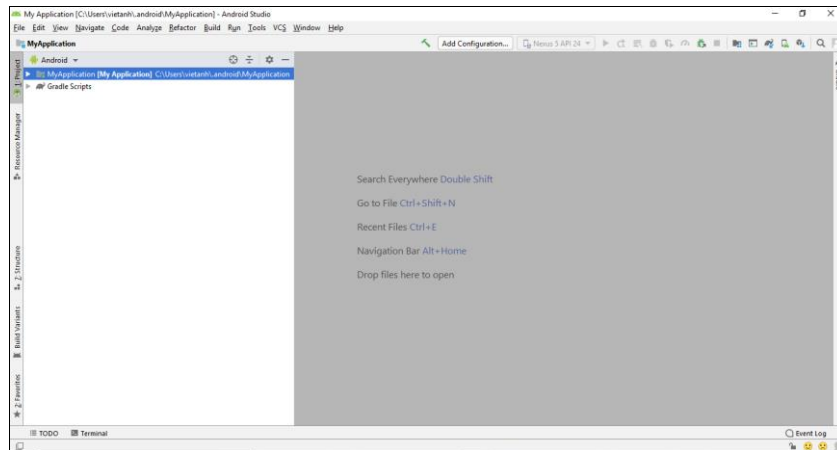
Màn hình này hiển thị cho phép chọn loại Activity mặc định.



Hình 2.12: Cài đặt Project mới

# Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Khi build xong mặc định có màn hình dưới đây:



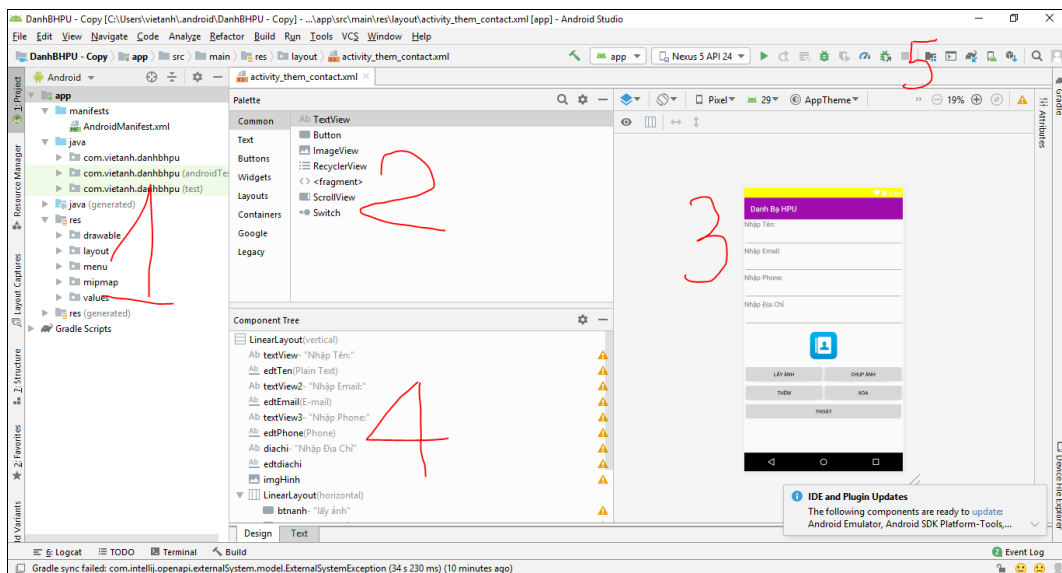
Hình 2.13: Màn hình khi cài đặt xong mặc định

## **b. Màn hình làm việc của dự án android studio.**

Theo mặc định Android Studio hiển thị các files trong project theo góc nhìn Android. Góc nhìn này Android Studio sẽ tổ chức các files theo 3 module

- manifests: chứa file AndroidManifest.xml.
- java: chứa các file mã nguồn Java.
- res: chứa tất cả các file layout, xml, giao diện người dùng(UI), ảnh.

Mở Project mặc định activity\_main.xml sẽ được chọn ta có màn hình như sau:

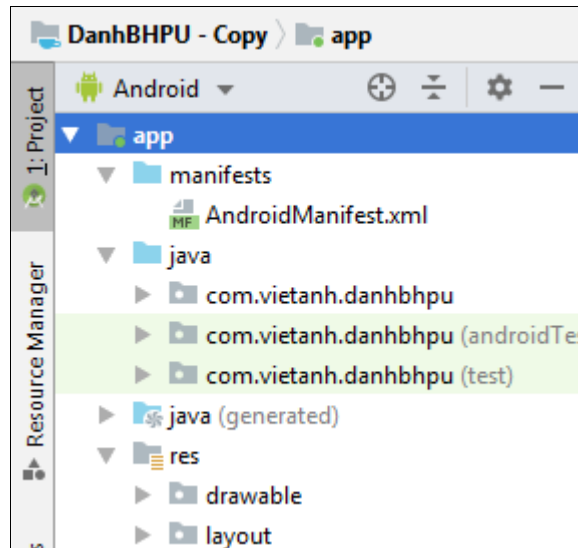


Hình 2.14: 5 vùng lập trình viên thường tương tác

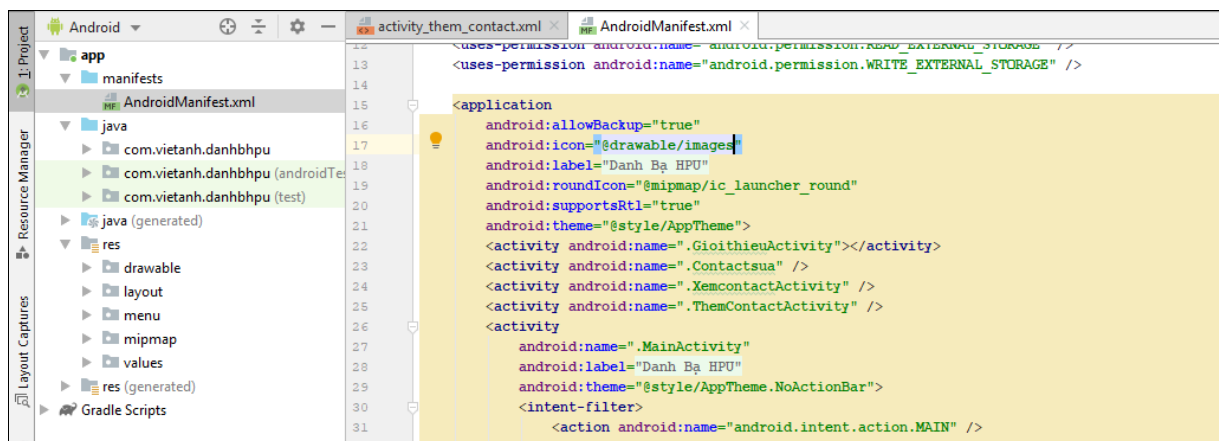
# Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

## - Vùng 1.

Là nơi cấu trúc hệ thống thông tin của Ứng dụng, Ta có thể thay đổi cấu trúc hiển thị (thường để mặc định là Android).



Hình 2.15: màn hình xem ở chế độ Android



Hình 2.16: AndroidManifest.xml

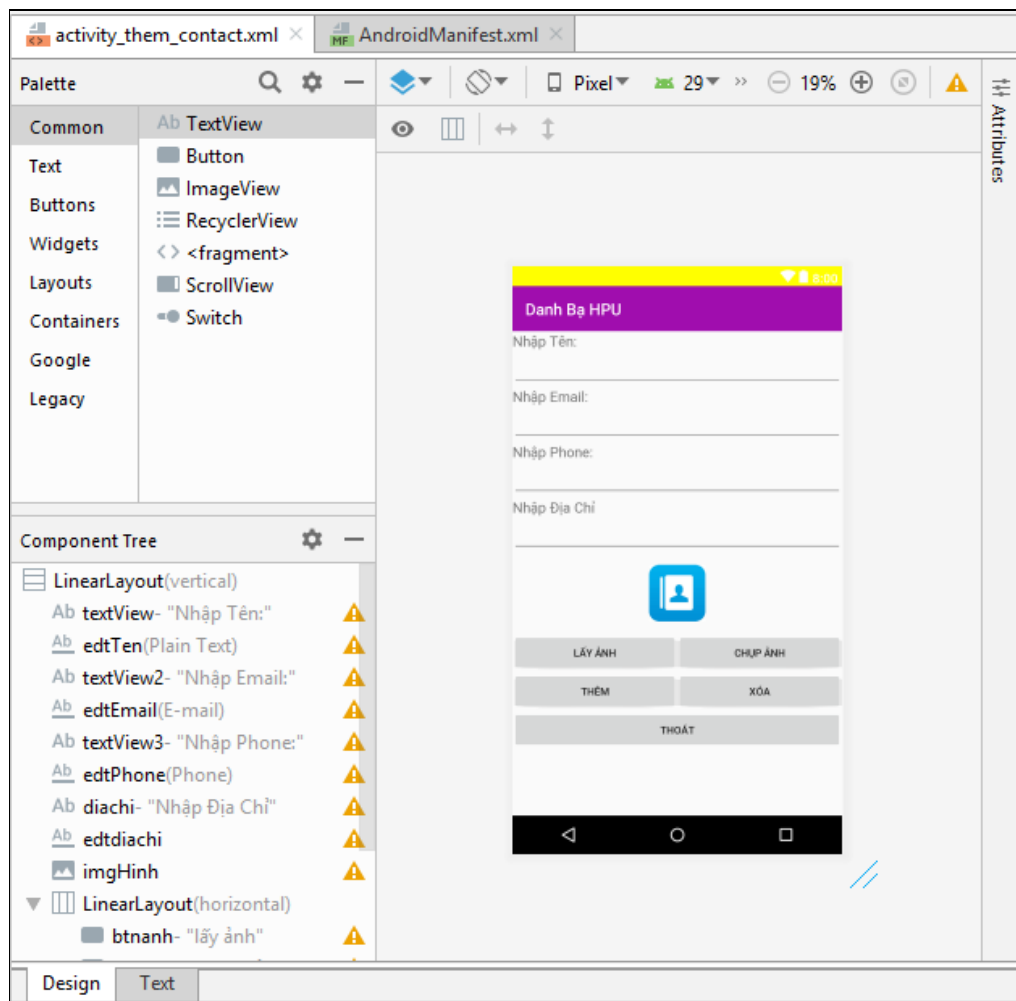
Ta có thể thấy **AndroidManifest.xml** nằm ở đây. File này vô cùng quan trọng trong việc cấu hình ứng dụng.

Các thư mục:

- **drawable**: chứa các file hình ảnh và xml trong ứng dụng.
- **layout**: chứa các giao diện màn hình được thiết kế dưới dạng xml.
- **values**: chứa các file lưu giá trị màu sắc, kích thước, chuỗi,....

### - Vùng 2.

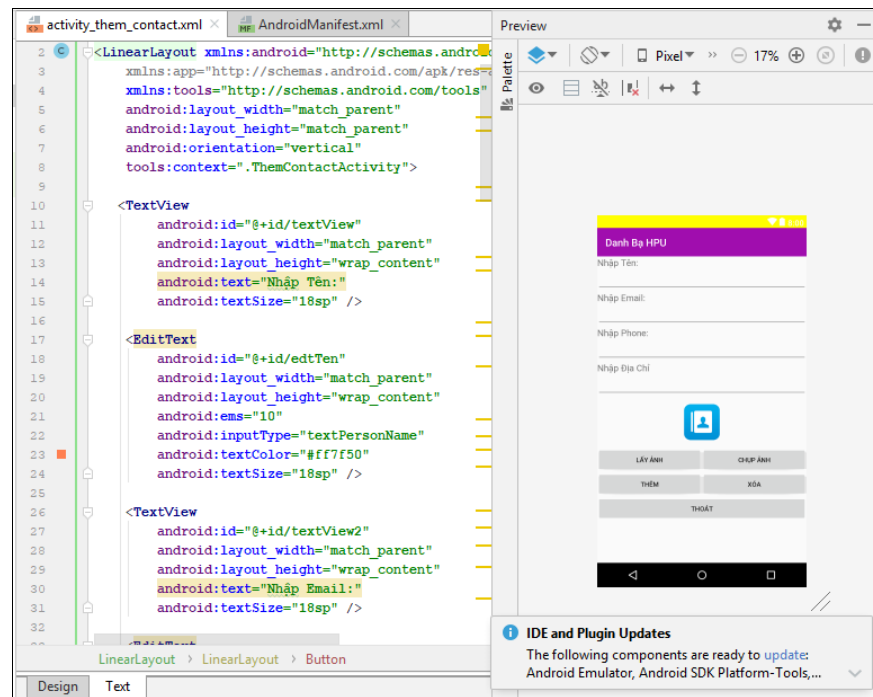
Là vùng khá quan trọng cho những bạn mới bắt đầu lập trình, nó là nơi hiển thị các Control mà Android hỗ trợ, cho phép bạn kéo thả trực tiếp vào vùng 3 (Giao Diện Thiết Bị) để thiết kế.



Hình 2.17: Vùng thiết kế giao diện ứng dụng

Ở vùng số 2 này nó có 2 tab: Design và Text ở góc trái dưới cùng.

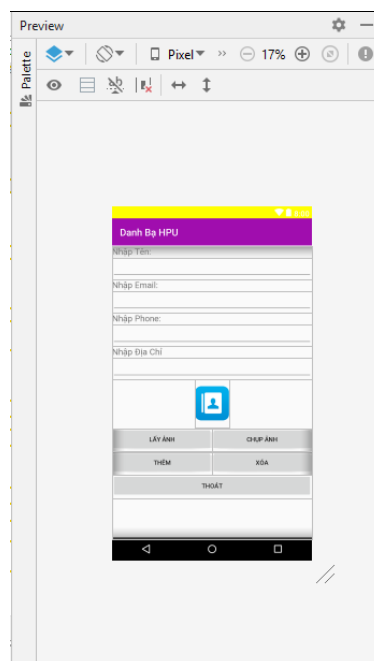
- Tab Design là tab mà ta đang nhìn và thao tác với nó (cho phép thiết kế giao diện bằng cách kéo thả).
- Tab Text là tab cho phép ta thiết kế giao diện bằng viết Tag XML:



Hình 2.18: Tab Text

## -Vùng 3.

Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiểu chính control.



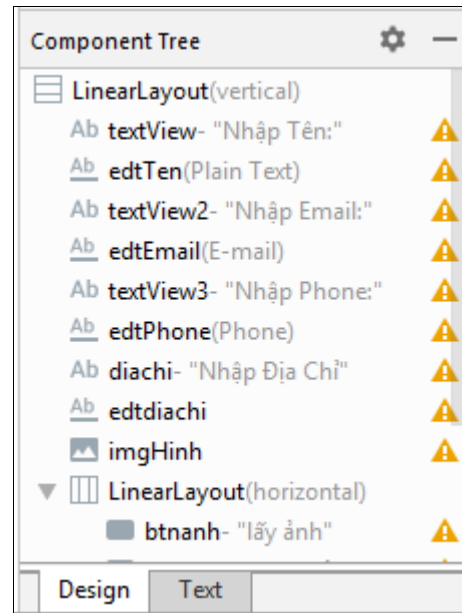
Hình 2.19:Hiển thị kết quả thiết kế giao diện thiết bị

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Vùng 3 ta có thể chọn cách hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉ control, lựa chọn loại thiết bị hiển thị....

### - Vùng 4.

Khi màn hình có nhiều control thì vùng 4 này trở lên hữu ích, nó cho phép hiển thị giao diện theo dạng cấu trúc cây, nên ta dễ dàng quan sát và lựa chọn control khi chúng bị chồng lặp trên giao diện (vùng 3).



Hình 2.20:Hiển thị giao diện cấu trúc cây

### - Vùng 5.

Là vùng các chức năng quan trọng thường dùng trong Android Studio.



Hình 2.21:Thanh chạy và debug ứng dụng

### 2.2.3. Tạo giao diện chương trình trong android studio

#### a. Giới thiệu android Layout

Layout là nơi chứa các control lên giao diện và mỗi layout có một cách sắp xếp các control khác nhau, vì vậy với mỗi cấu trúc giao diện khác nhau ta nên chọn layout cho phù hợp. Sau đây là một số layout cơ bản cho để ta thiết kế giao diện.

## - **FrameLayout.**

Là loại layout cơ bản nhất, nó sẽ được dùng nhiều khi ta sử dụng vẽ giao diện nâng cao sau này. Khi ta kéo các control vào thì mặc định các control sẽ nằm ở vị trí trên cùng bên trái. Các control khi được kéo vào framelayout sẽ bị đè lên nhau, control sau sẽ đè lên control trước. Cách duy nhất để căn các control vào giữa là sử dụng thuộc tính android:layout\_gravity="center.

## - **LinearLayout.**

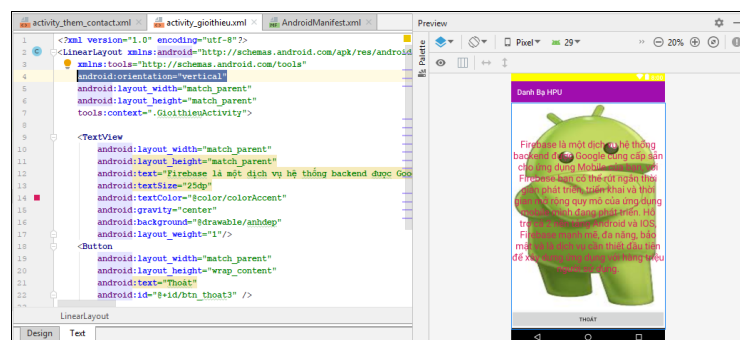
Layout này cho phép ta vẽ giao diện theo 2 hướng, từ trái qua phải hoặc từ trên xuống dưới. Để xét chiều cho các control trong layout ta sử dụng thuộc tính **orientation**.

- Android:orientation="horizontal" : Xếp các control từ trái sang phải (theo cột).
- Android:orientation="vertical" : Xếp các control từ trên xuống dưới (theo hàng).

Với những giao diện có độ phức tạp vừa phải thì dùng LinearLayout là rất hiệu quả, rất thuận tiện trong thiết kế và đi bảo trì ứng dụng sau này.

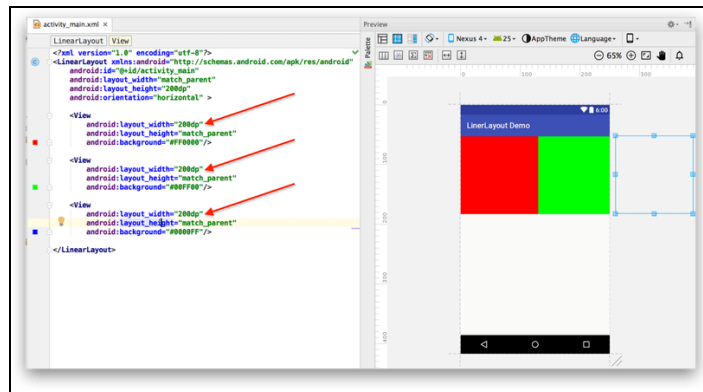
Sau đây là đoạn XML demo cách sử dụng layout này:

Theo hàng



Hình 2.22:Linerlayout theo hàng

Theo cột



Hình 2.23: Linerlayout theo cột

## - RelativeLayout.

Layout này cho phép ta sắp xếp các control theo vị trí tương đối giữa các control khác kể cả control chứa nó. Khi gặp những layout có độ phức tạp cao, có nhiều giao diện nhỏ thì sử dụng RelativeLayout là lựa chọn tốt nhất. Một vài chú ý khi sử dụng layout này:

- Các control đều có id riêng, việc đặt tên id phải rõ ràng dễ hiểu.
- Các control được sắp xếp dựa vào id của các control khác.
- Các control có sự ràng buộc và tương tác với nhau nên khi thay đổi một control sẽ làm thay đổi vị trí của mọi control khác. Vì vậy rất khó trong việc bảo trì nếu giao diện quá phức tạp.

### b. Giới thiệu một số android View cơ bản

**TextView:** là view sử dụng để hiển thị text màn hình. TextView được định nghĩa bởi thẻ trong xml.

**EditText:** là view dùng để lấy giá trị từ người dùng nhập vào. EditText được định nghĩa bởi thẻ trong xml.

**ImageView:** là một view sử dụng rất nhiều trong ứng dụng android, ImageView sử dụng để hiển thị hình ảnh.



## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

**Button:** là view được sử dụng khá nhiều trong android, hầu như sử dụng ở mọi nơi cùng với EditText, TextView. Button có chức năng là làm nhiệm vụ nào đó khi mà người dùng click trong phương thức onClick.

**ListView:** được tạo từ một danh sách các ListItem. ListItem là một dòng (row) riêng lẻ trong listview nơi mà dữ liệu sẽ được hiển thị. Bất kỳ dữ liệu nào trong listview chỉ được hiển thị thông qua listItem. Có thể coi listview như là một nhóm cuộn của các ListItem.

### c. Bắt và xử lý sự kiện trên giao diện.

Sự kiện là một cách hữu ích để thu thập dữ liệu về sự tương tác của người dùng với các thành phần tương tác của ứng dụng. Giống như bấm vào một nút hoặc chạm vào màn hình cảm ứng, vv. Ta có thể nắm bắt những sự kiện trong chương trình và có những xử lý thích hợp theo yêu cầu. Có hai khái niệm liên quan đến quản lý sự kiện Android:

**Event Listeners** là một interface. Event Listeners được sử dụng để đăng ký sự kiện cho các thành phần trong UI. (Đăng ký sự kiện). Trong các giao tiếp event listener có những phương thức sau đây:

- **onClick():** Thuộc View.OnClickListener. Nó được gọi khi người dùng hoặc chạm vào item (khi ở chế độ cảm ứng), hoặc lựa chọn vào item với các phím điều hướng và nhấn nút "enter" phù hợp.
- **onLongClick():** Thuộc View.OnLongClickListener. Nó được gọi khi người dùng chạm và giữ item (khi ở chế độ cảm ứng), hoặc lựa chọn vào item với các phím điều hướng sau đó nhấn và giữ phím "enter".
- **onFocusChange():** Thuộc View.OnFocusChangeListener. Nó được gọi khi người dùng điều hướng ra khỏi item, bằng cách sử dụng phím điều hướng.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

---

- **onKey():** Thuộc View.OnKeyListener. Nó được gọi khi người dùng lựa chọn và nhấn lên item.
- **onTouch():** Thuộc View.OnTouchListener. Nó được gọi khi người dùng thực hiện một hành động xác định đủ điều kiện như là một sự kiện cảm ứng, bao gồm việc nhấn, thoát ra, hoặc bất kỳ cử chỉ chuyển động vẽ trên màn hình (bên trong phạm vi của item).
- **onCreateContextMenu():** Thuộc View.OnCreateContextMenuListener. Nó được gọi khi một menu ngữ cảnh (Context Menu) đang được xây dựng (là kết quả của một "long click"). Xem thêm thông tin về context menus trong hướng dẫn phát triển Menus.

Ví dụ dưới đây cho thấy làm thế nào để đăng ký một bộ bắt sự kiện khi nhấp chuột vào một Button.

```
public class MainActivity extends AppCompatActivity {  
  
    Button button; //Khai báo một button  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        button = findViewById(R.id.bnt_click); //Tìm lại button  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast.makeText(MainActivity.this, "Bạn đã  
click", Toast.LENGTH_SHORT).show();  
            }  
        });  
    }  
}
```

- **onLongClick()** - Trả về một giá trị kiểu boolean để cho biết ta đã dùng sự kiện này và nó không cần thực hiện "long click" thêm nữa. Trả về giá trị *TRUE* để chỉ ra rằng ta đã xử lý sự kiện này và nó nên dừng lại ở đây; trả về *FALSE* nếu ta không xử lý nó và / hoặc sự kiện nên chuyển tới bất kỳ bộ nghe sự kiện **on-click** nào khác.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

- **onKey()** - Trả về một giá trị kiểu boolean để cho biết ta đã dùng sự kiện này và nó không cần được thực hiện thêm. Trả về giá trị *TRUE* để chỉ ra rằng ta đã xử lý sự kiện này và nó nên dừng lại ở đây; trả về *FALSE* nếu ta không xử lý nó và / hoặc sự kiện nên chuyển tới bất kỳ bộ nghe sự kiện **on-key** nào khác.
- **onTouch()** - Trả về một giá trị kiểu boolean để cho biết: liệu bộ nghe của ta đã dùng sự kiện này hay chưa. Điều quan trọng là sự kiện này có thể có nhiều hành động nối tiếp nhau. Vì vậy, nếu trả về *FALSE*, ta biết rằng ta đã không sử dụng và cũng không quan tâm đến hành động tiếp theo từ sự kiện này. Như vậy, ta không được gọi tới bất kỳ thao tác nào khác bên trong sự kiện này.

**Event Handlers** – Là phương thức xử lý khi phát sinh sự kiện. (Xử lý sự kiện)

Nếu ta đang xây dựng một thành phần tùy chỉnh từ View, ta sẽ phải định nghĩa một số phương thức sử dụng như của xử lý sự kiện mặc định. Trong tài liệu về Custom Components, ta sẽ tìm hiểu một số callbacks thường được sử dụng để xử lý sự kiện, bao gồm:

- **onKeyDown(int, KeyEvent)** – Được gọi khi một sự kiện nhấn phím mới xảy ra.
- **onKeyUp(int, KeyEvent)** – Được gọi khi một sự kiện thả phím xảy ra.
- **onTrackballEvent(MotionEvent)** – Được gọi khi một sự kiện chuyển động trackball xảy ra.
- **onTouchEvent(MotionEvent)** – Được gọi khi một sự kiện chuyển động màn hình cảm ứng xảy ra.
- **onFocusChanged(boolean, int, Rect)** – Được gọi khi view được chọn (focus) hoặc bỏ chọn.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Có một số phương thức khác mà ta nên biết, chúng không phải là một phần của lớp View, nhưng có thể trực tiếp tác động đến cách bạn có thể xử lý các sự kiện. Vì vậy, khi quản lý sự kiện phức tạp hơn bên trong một layout, ta nên xem xét các phương pháp sau:

- `Activity.dispatchTouchEvent(MotionEvent)` – Điều này cho phép Activity bắt tất cả các sự kiện chạm màn hình trước khi chúng được gửi đến cửa sổ.
- `ViewGroup.onInterceptTouchEvent(MotionEvent)` – Điều này cho phép ViewGroup xem các sự kiện như chúng được gửi đến các View con.
- `ViewParent.requestDisallowInterceptTouchEvent(boolean)` – Gọi điều này trên View cha để xác định rằng nó không nên bắt các sự kiện chạm màn hình với `onInterceptTouchEvent(MotionEvent)`.

## Chương 3:

### Giới thiệu cơ sở dữ liệu thời gian thực firebase

#### 3.1. Firebase là gì

**Firestore** là một nền tảng ứng dụng di động và web với các công cụ và hạ tầng được thiết kế để giúp các lập trình viên xây dựng các ứng dụng chất lượng cao. Nói ngắn gọn, thay vì trực tiếp cung cấp các ứng dụng, họ cung cấp các dịch vụ nền tảng cho các lập trình viên, chính là các bạn, sử dụng để xây dựng ứng dụng cũng như hỗ trợ các bạn tối ưu hóa, tối đa hóa ứng dụng của mình. Với nhiều dịch vụ chất lượng cao đi kèm mức giá phải chăng, Firestore đã và đang, không chỉ là sự lựa chọn hàng đầu cho các lập trình viên đơn thân (single dev) hay các công ty khởi nghiệp (start ups), mà các công ty, tổ chức lớn có tên tuổi cũng sử dụng “Ngọn lửa” để xây dựng các tính năng, các chương trình mới, cũng như chuyển đổi các dịch vụ trước đây sang hệ thống của Firestore. Chẳng hạn như Shazam, Fabulous và cả chính Google nữa, khi nền tảng nhắn tin Allo được xây dựng trên nền tảng Firestore Realtime Database.

Về mặt lịch sử, Firestore (tiền thân là Evolve) trước đây là một start up được thành lập vào năm 2011 bởi Andrew Lee và James Tamplin. Ban đầu, Evolve chỉ cung cấp cơ sở dữ liệu để các lập trình viên thiết kế các ứng dụng chat (và hiện tại thì để làm quen với realtime db thì bạn cũng làm ứng dụng chat đó thôi). Tuy nhiên, họ nhanh chóng nhận ra tiềm năng sản phẩm của mình khi nhận thấy các khách hàng không sử dụng CSDL để làm ứng dụng chat, mà thay vào đó, để lưu các thông tin như game progress. Bộ đôi Lee và Tamplin quyết định tách mảng realtime ra để thành lập một công ty độc lập – chính là Firestore – vào tháng 4 năm 2012. Sau nhiều lần huy động vốn và gặt hái được những thành công nổi bật, Firestore đã được Google đề ý. Vào tháng 10 năm 2014, Firestore gia nhập gia đình Google.

Cả Google và Firestore đều như hồ mọc thêm cánh. Firestore có điều kiện để phát triển thần tốc, mở rộng số lượng các dịch vụ con, còn Google có được một đội ngũ nhân lực chất lượng cao, năng động, cũng như cơ sở hạ tầng và sự hiệu quả mà các dịch vụ của Firestore mang lại, mà không phải xây dựng lại từ đầu. Hiện tại, Google đã chuyển các dịch vụ nền tảng hỗ trợ các lập trình viên bên ngoài về cho Firestore quản lý, chẳng hạn như Cloud Messaging, AdMob và Analytics.

Firestore, theo hướng đi của Google, chính thức hỗ trợ Android, iOS và Web. Thực tế, macOS cũng được hỗ trợ vì macOS chia sẻ nhiều dòng code với iOS, song vì Google và Firestore muốn sử dụng web cho ứng dụng desktop thay vì native, nên có khá ít tài liệu chính thức nói về Firestore cho macOS, cũng như

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

---

các thư viện cho macOS có thể kém chức năng và không ổn định lắm. Còn về Windows, hiện tại tôi chưa thấy họ lên tiếng nào về việc sẽ chính thức phát hành thư viện cho đứa con của Microsoft, nên nếu các bạn muốn làm ứng dụng cho Windows (UWP) thì chỉ nên (và cũng chỉ có mỗi con đường) làm web-based native apps mà thôi.

### **Ưu điểm:**

Như bạn có thể thấy, Firebase cung cấp rất nhiều dịch vụ hỗ trợ giúp các lập trình viên xây dựng ứng dụng của mình. Việc tích hợp Firebase vào ứng dụng có thể hạ nỗi lo lắng của các dev xuống ở mức coding, đặc biệt là cho các dev viết các ứng dụng có liên quan tới web. Bước đầu tiên, bạn cần có host để chứa cơ sở dữ liệu => đã có Firebase Hosting và Storage. Bạn cần viết code PHP để giao tiếp? Có thể không cần cho lắm, vì Firebase Authentication và Realtime Database đã sẵn sàng cho bạn tích hợp và sử dụng “ngay và luôn”. Bạn muốn gửi thông báo tới các thiết bị? Firebase Cloud Messaging là cái bạn cần. Hơn thế nữa, với Test Lab, Crash Reporting, App Indexing thì Firebase sẽ báo cáo cho bạn các vấn đề có thể xảy ra khi ứng dụng của bạn chạy nhiều trên máy thật. Cũng như Remote Config giúp bạn thay đổi các thiết lập trên ứng dụng của người dùng mà không cần thiết phải publish 1 phiên bản mới...

Cá nhân tôi, sau khi sử dụng Firebase, mặc dù chỉ ở mức độ “cho biết”, nhưng đã có thể kết luận là sẽ sử dụng Firebase cho các ứng dụng có liên quan tới web sau này, bởi tôi chỉ xếp nó sau việc tự thiết kế toàn bộ hạ tầng từ A-Z, còn nếu đem so với các nhà cung cấp khác thì không ai có thể vượt qua được “Ngọn lửa” cả. Các nhà cung cấp khác có thể có một vài dịch vụ chất lượng hơn, nhưng nhìn chung, họ không cung cấp số lượng dịch vụ nhiều như vậy.

### **Nhược điểm:**

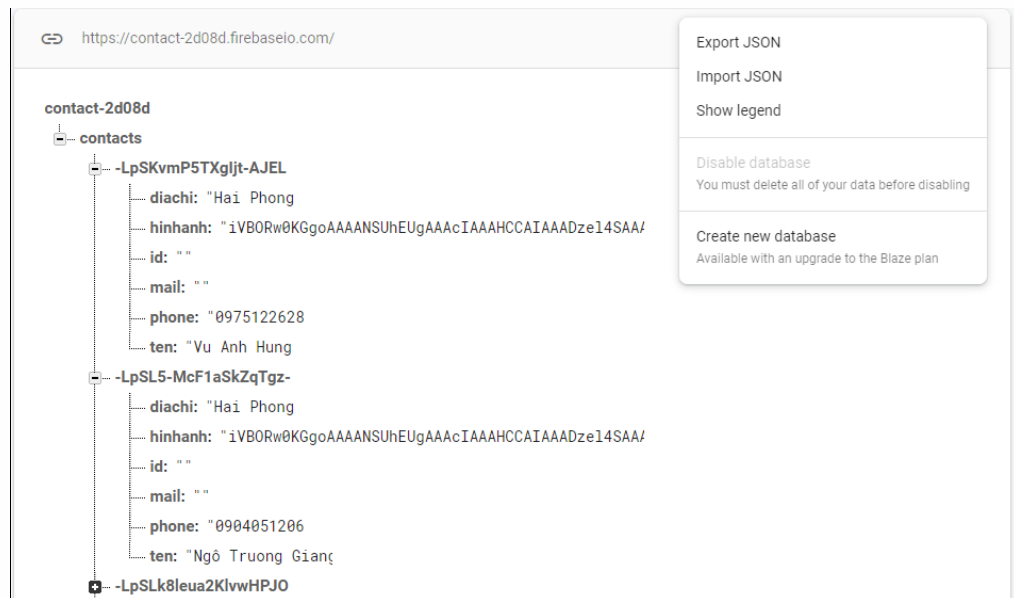
Không có gì là hoàn hảo, và Firebase cũng không là ngoại lệ. Điểm yếu duy nhất của “Ngọn lửa” chính là phần Realtime Database, mà đúng hơn chỉ là phần Database. Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn từ mức đôi chút tới khá nhiều.

### **3.2. Cấu trúc dữ liệu**

Realtime Database Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình viên phát triển nhanh các ứng dụng có tương tác cơ sở dữ liệu (CRUD) một cách nhanh chóng và ngay tức thời (Realtime).

# Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Khi có sự thay đổi dữ liệu trên Database Firebase thì ngay lập tức giao diện của bất kỳ thiết bị nào có sử dụng phần mềm này sẽ tự động cập nhật (gọi là Realtime):



Hình 3.1: Cấu trúc cây Json

Dữ liệu trên Database Firebase có định dạng kiểu cây JSON.

CRUD: Là 4 thao tác không thể thiếu với mọi ứng dụng có tương tác Cơ Sở Dữ liệu.

C (Create- thêm mới).

R (Retrieve – truy vấn xem dữ liệu).

U (Update- cập nhật dữ liệu).

D (Delete- xóa dữ liệu).

### 3.3. Cơ chế hoạt động

Để thực hiện bất kỳ phương thức nào trên cơ sở dữ liệu cho dù đó có thể được đọc hoặc ghi, ta cần phải lấy được các tham chiếu đến cơ sở dữ liệu. Các đoạn

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

chương trình sau đây cho phép bạn tham chiếu đến nút trên cùng của cơ sở dữ liệu JSON. Từ đây ta cần phải sử dụng các tên nút con phải đi qua các nút khác.

```
private DatabaseReference mDatabase;  
  
mDatabase =  
FirebaseDatabase.getInstance().getReference();
```

### 3.3.1 Thêm mới

Để thêm dữ liệu, bạn có thể sử dụng phương thức setValue() trên đường dẫn tham chiếu đến database. Nó sẽ tạo mới và cập nhật giá trị trên đường dẫn được cung cấp. Ví dụ ở dưới đã mã để thêm 1 nút được gọi là “copyright” trên cây json ở mức đỉnh.

```
DatabaseReference mRef =  
mDatabase.getReference("copyright");  
  
mRef.setValue("©2016 androidhive. All rights  
Reserved");
```

Realtime database chấp nhận nhiều loại dữ liệu: String, Long, Double, Boolean, Map<String, Object>, List<Object> để lưu trữ dữ liệu. Ta cũng có thể sử dụng dữ liệu tùy biến của đối tượng để lưu trữ dữ liệu, điều này rất hữu dụng khi lưu trữ đối tượng vào database một cách trực tiếp

Giả sử ta muốn lưu trữ thông tin người dùng vào database. Đầu tiên ta cần tạo User model cùng với constructor rỗng and những thuộc tính khác

```
public class User {  
  
    public String name;  
    public String email;  
  
    // Default constructor required for calls to  
    // dataSnapshot.getValue(User.class)  
    public User() {  
    }  
}
```



```
public User(String name, String email) {
    this.name = name;
    this.email = email;
}
```

Mọi người dùng cần một ID duy nhất, ta có thể tạo ra một bằng cách gọi phương thức `push()` để tạo ra một nút trống rỗng, với khóa duy nhất. Sau đó, được tham chiếu đến nút 'user' bằng phương thức `child()`. Cuối cùng sử dụng phương thức `setValue()` để lưu trữ dữ liệu người dùng.

```
DatabaseReference mDatabase =
FirebaseDatabase.getInstance().getReference("users");

// Creating new user node, which returns the unique
key value
// new user node would be /users/$userid/
String userId = mDatabase.push().getKey();

// creating user object
User user = new User("Ravi Tamada",
"ravi@androidhive.info");

// pushing user to 'users' node using the userId

mDatabase.child(userId).setValue(user);
```

Bằng việc chạy đoạn code ở dưới, 1 nút người dùng mới sẽ được chèn vào database cùng với giá trị khóa duy nhất. Nhìn chung, user id nên được lấy lại bằng cách thực hiện xác thực Firebase trong ứng dụng mà ta cung cấp AUTHID hoạt động như người sử dụng id.

### 3.3.2. Truy vấn xem dữ liệu

Để đọc dữ liệu, bạn cần đính kèm phương thức `ValueEventListener()` để tham chiếu database. Sự kiện này sẽ được kích hoạt bất cứ khi nào có sự thay đổi trong dữ liệu trong thời gian thực. Trong `onDataChange()`, bạn có thể thực hiện các phương thức mong muốn vào dữ liệu mới.

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Dưới đây là sự kiện lắng nghe được kích hoạt bất cứ khi nào có sự thay đổi trong dữ liệu hồ sơ người dùng mà đã tạo ra trước đó.

```
mDatabase.child(userId).addValueEventListener(new
 ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot
 dataSnapshot) {

        User user =
 dataSnapshot.getValue(User.class);

        Log.d(TAG, "User name: " + user.getName() +
 ", email " + user.getEmail());
    }

    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
        Log.w(TAG, "Failed to read value.",
 error.toException());
    }
});
```

### 3.3.3. Cập nhật dữ liệu

Để cập nhật dữ liệu, ta có thể sử dụng cùng phương pháp setValue() để passing giá trị mới. Ta cũng có thể sử dụng phương thức updateChildren() để passing đường dẫn để cập nhật dữ liệu mà không làm ảnh hưởng đến các nút con khác.

Ví dụ, nếu ta muốn cập nhật chỉ email của người dùng, ta có thể sử dụng đoạn code bên

```
String newEmail = 'androidhive@gmail.com';

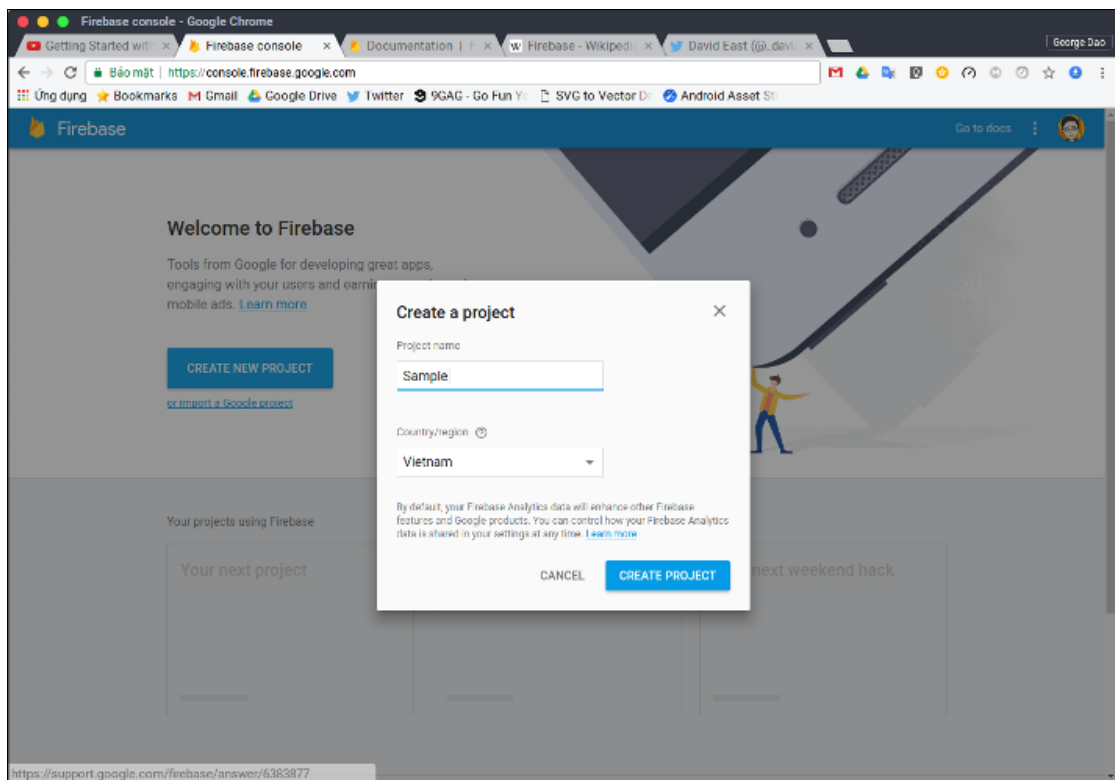
mDatabase.child(userId).child("email").setValue(newEm
ail);
```

## 3.3.4. Xóa dữ liệu

Để xóa dữ liệu, ta có thể gọi phương thức `removeValue()` trong tham chiếu database.

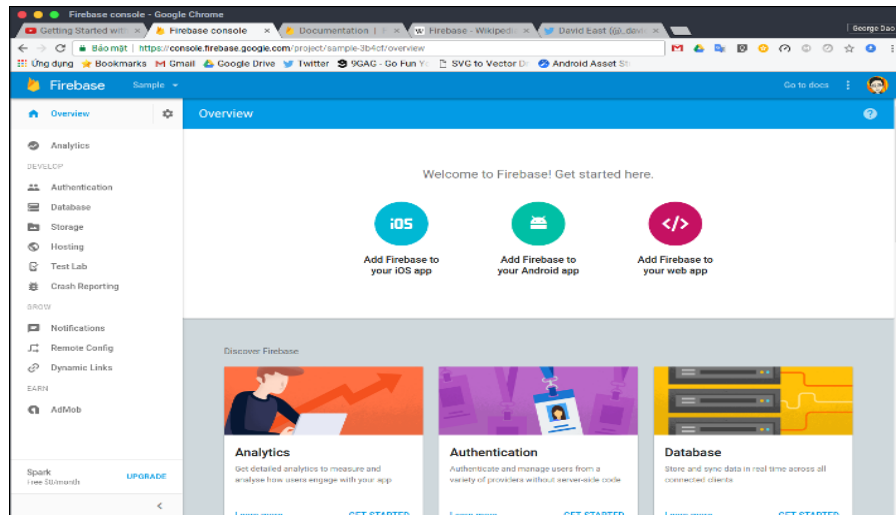
## 3.4. Tích hợp firebase vào ứng dụng

Để tạo một Firebase project mới, các bạn truy cập vào **Firestore Console**, đăng nhập với tài khoản Google của mình, và nhấn **CREATE NEW PROJECT** và tạo project.



Hình 3.2: Tạo tên cho dự án

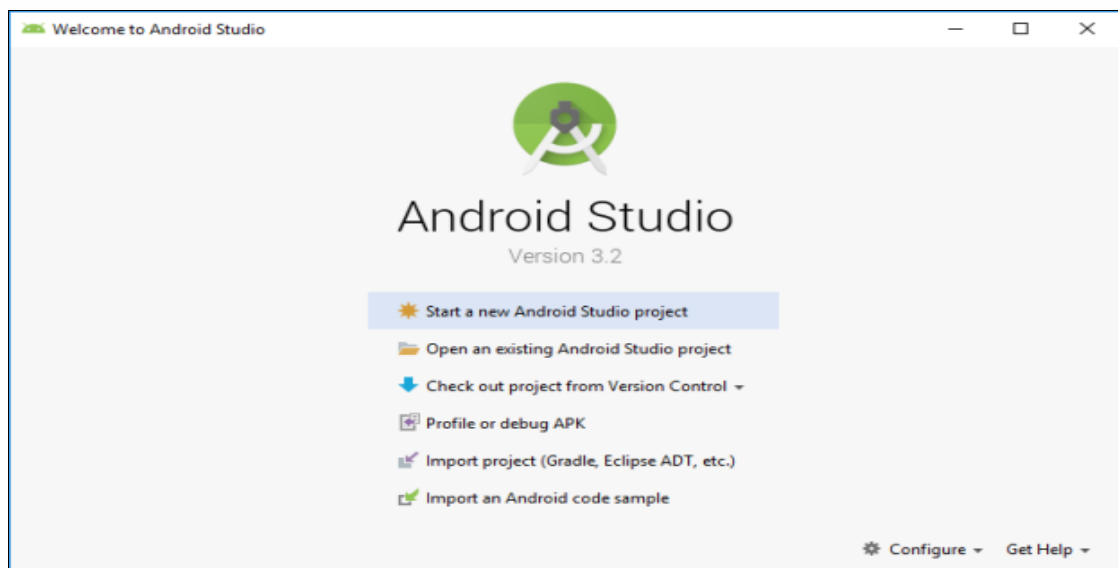
Sau khi tạo xong, bạn sẽ thấy giao diện trang tương tự như sau:



Hình 3.3: Giao diện trang

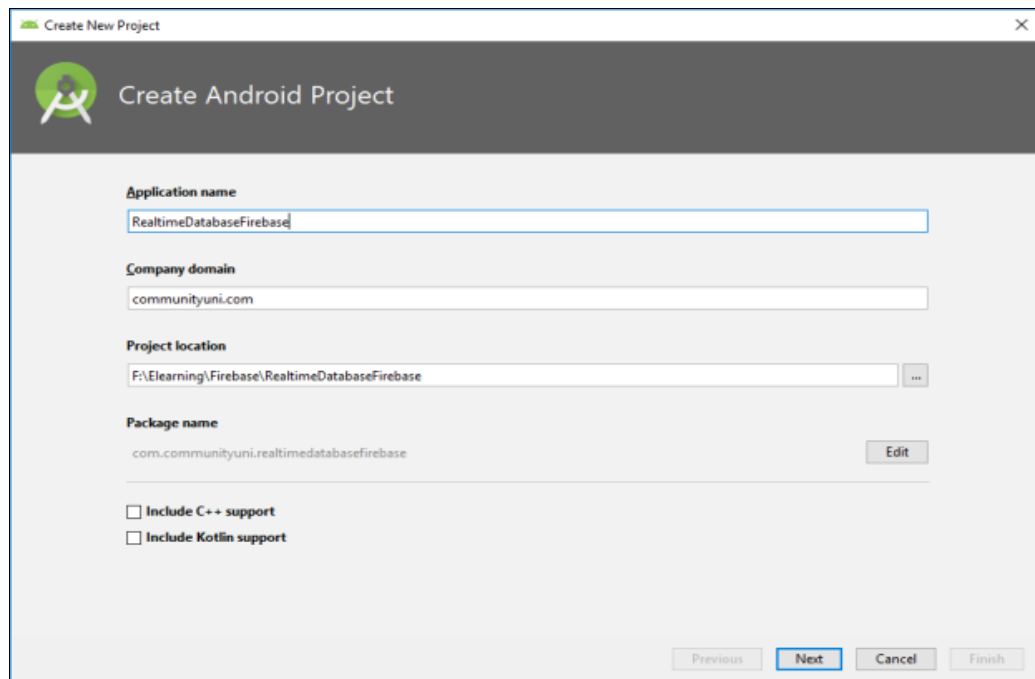
## 3.5. Các thao tác dữ liệu cơ bản trên firebase

Khởi động Android Studio:



Hình 3.4: Màn hình bắt đầu tạo Project

Click Chọn “Start a new Android Studio Project”, lúc này cửa sổ yêu cầu tạo tên Project sẽ hiển thị ra, ta Đặt tên Project là RealtimeDatabaseFirebase:



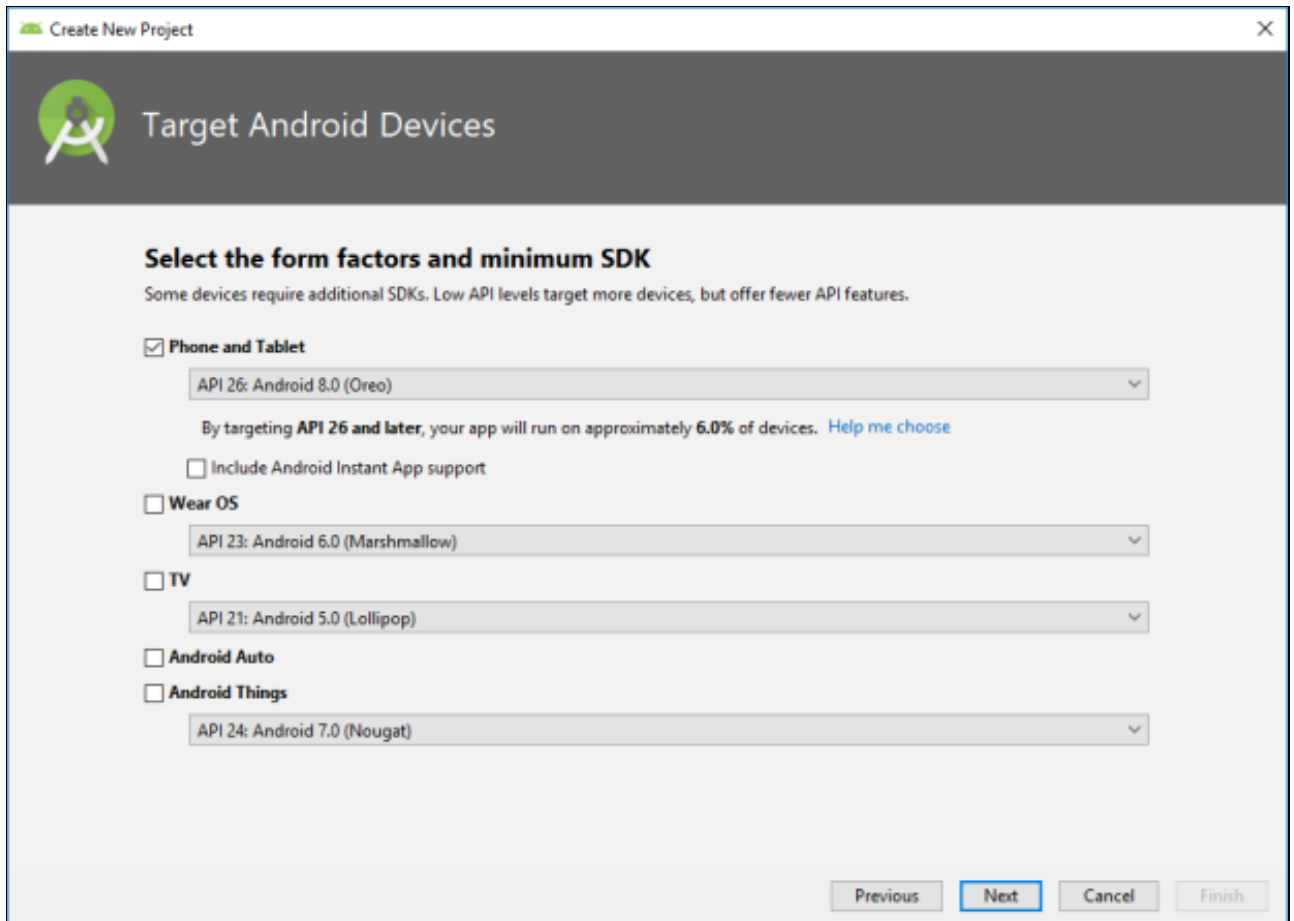
Hình 3.5: Màn hình đặt tên Project

Mục Application Name: Đặt tên cho ứng dụng **RealtimeDatabaseFirebase**

Mục Company Domain: Đặt tên theo domain của bạn, không có thì đặt đại nhưng phải viết thường hết. Bản chất nó là package để tổ chức sắp xếp lại hệ thống các lớp trong Project, cũng là cơ sở để Google quản lý ứng dụng trên Google Play, Developer nữa.

Mục Project Location: Nơi lưu trữ dữ án

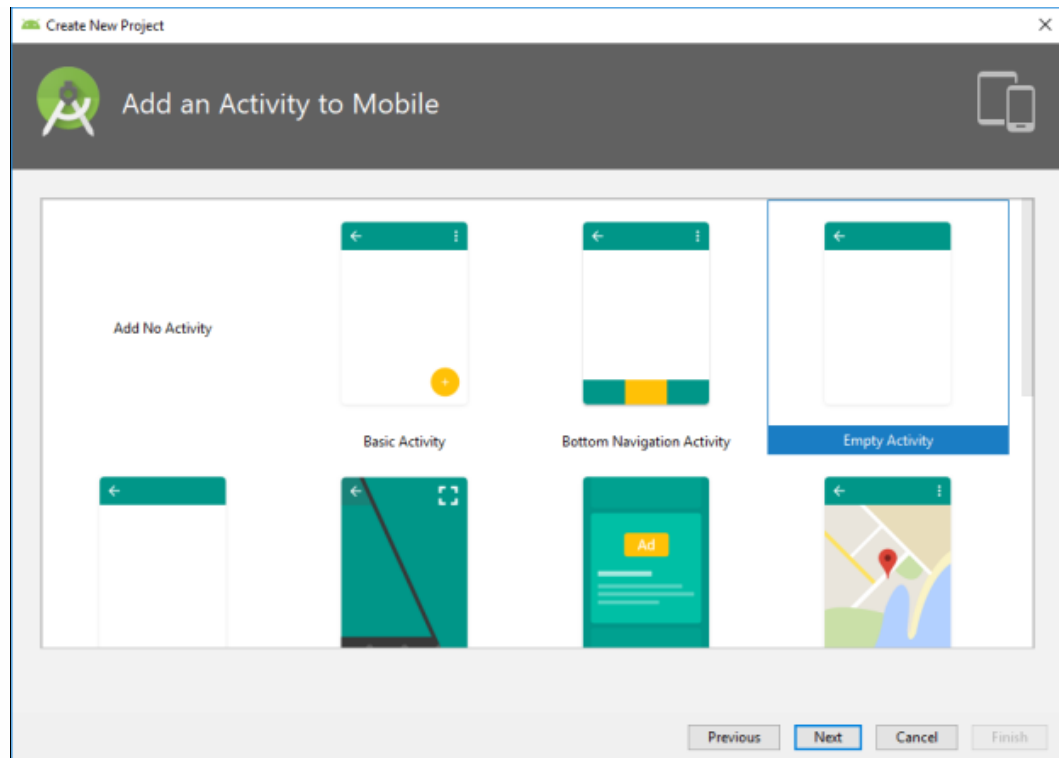
Lưu ý không tick vào C++ hay Kotlin. Sau cung cấp đủ các thông tin rồi thì bấm Next để qua bước tiếp theo:



*Hình 3.6: Cài đặt Project*

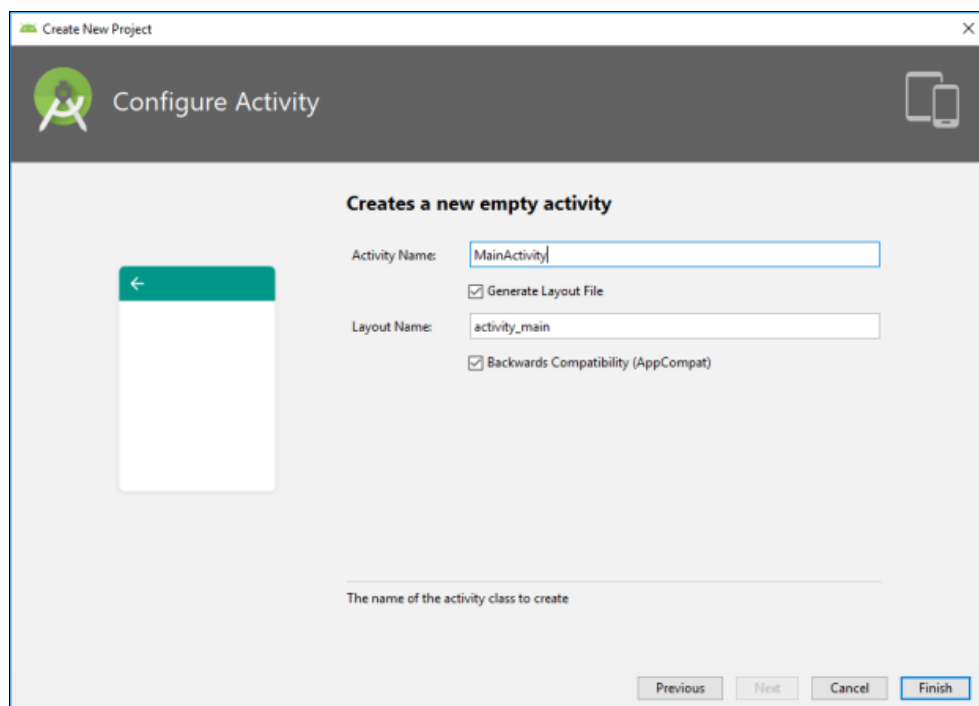
Ở màn hình này ta chọn API là 26 rồi bấm Next:

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng



Hình 3.7: Chọn kiểu dự án

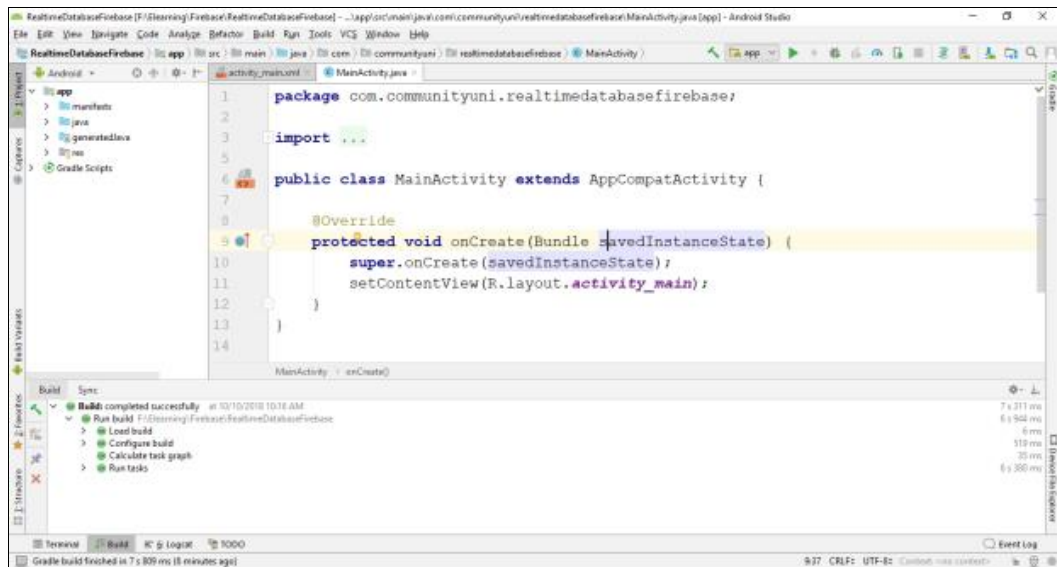
Chọn Empty Activity rồi bấm Next



Hình 3.8: Ấn Finish khi chọn kiểu dự án

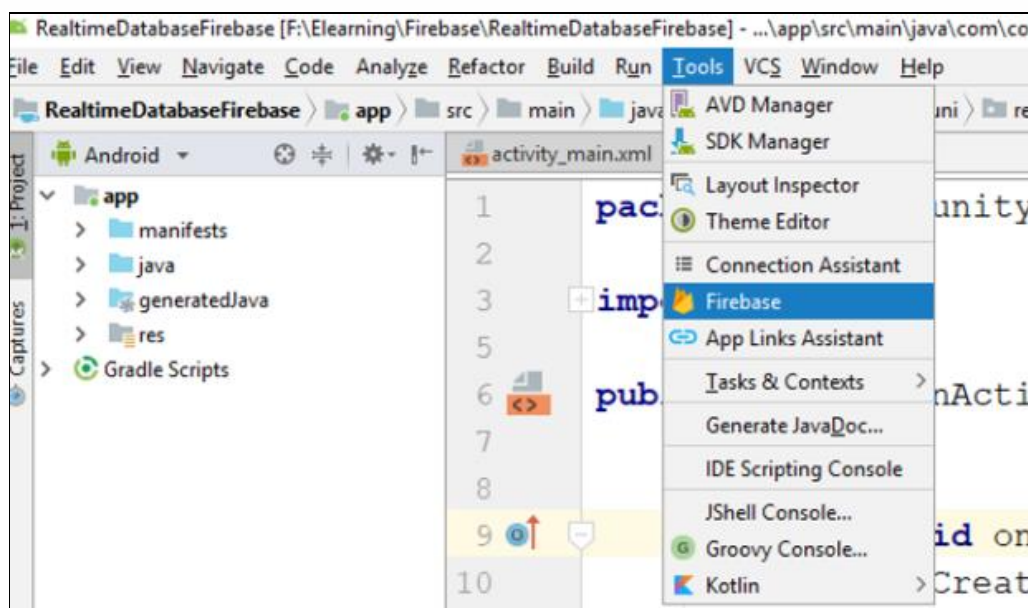
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Bước này là thiết lập màn hình chính chạy: Activity Name là lớp Java để xử lý nghiệp vụ, Layout Name là giao diện cho màn hình đó, để mặc định vậy bấm FINISH



Hình 3.9: Màn hình chính để thiết kế

Như trên là ta đã tạo xong được Project trong Android Studio. Bây giờ bước tiếp theo là kết nối với Firebase để sử dụng Realtime Database Firebase. Ta làm như sau:

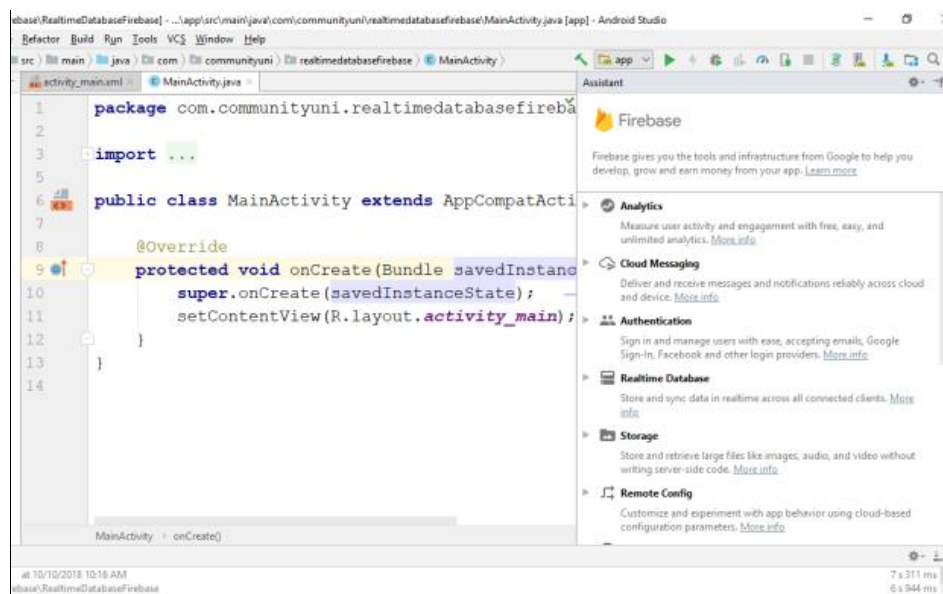


Hình 3.10: Kết nối với Firebase



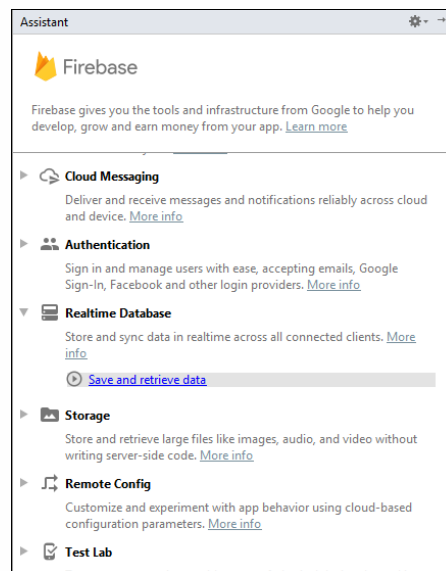
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

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:



Hình 3.11: Kết nối với Firebase

Trong màn hình trên, Firebase cung cấp rất nhiều công cụ, tuy nhiên ta chỉ quan tâm tới Realtime Database. Vì vậy ta nhấn vào Realtime Database:

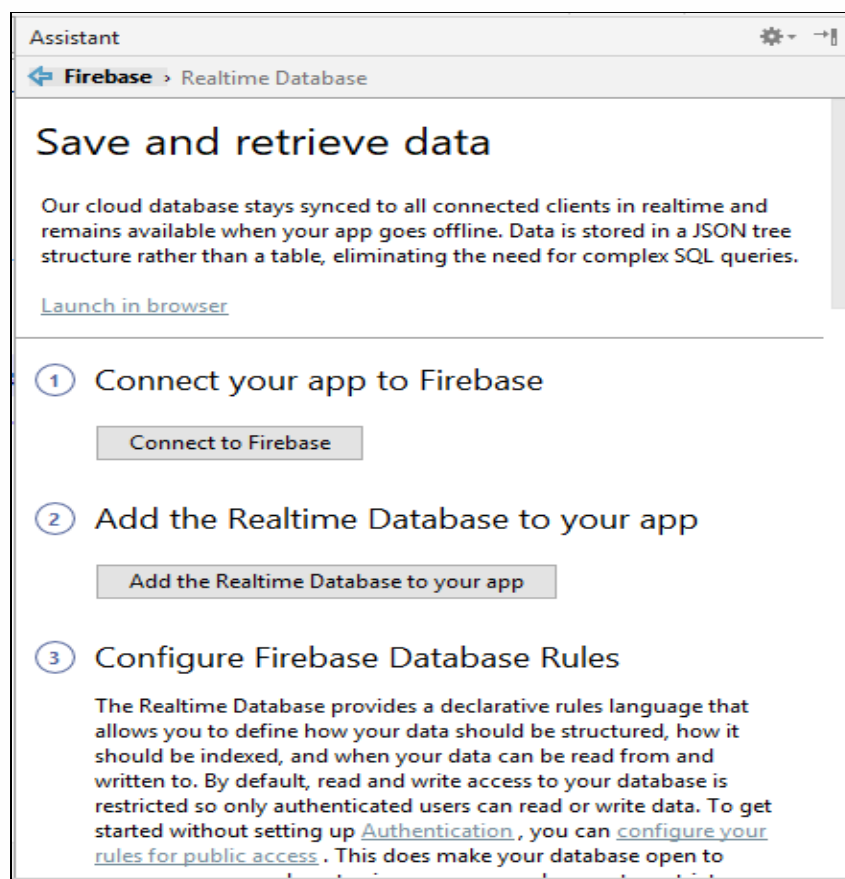


Hình 3.12: Kết nối với Firebase

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Lúc này ta thấy nút “Save and retrieve data” hiển thị ra như hình trên. Trình trợ giúp cung cấp cho ta 8 bước thao tác (bao gồm kết nối và hướng dẫn lập trình, ta dùng 2 -3 bước):

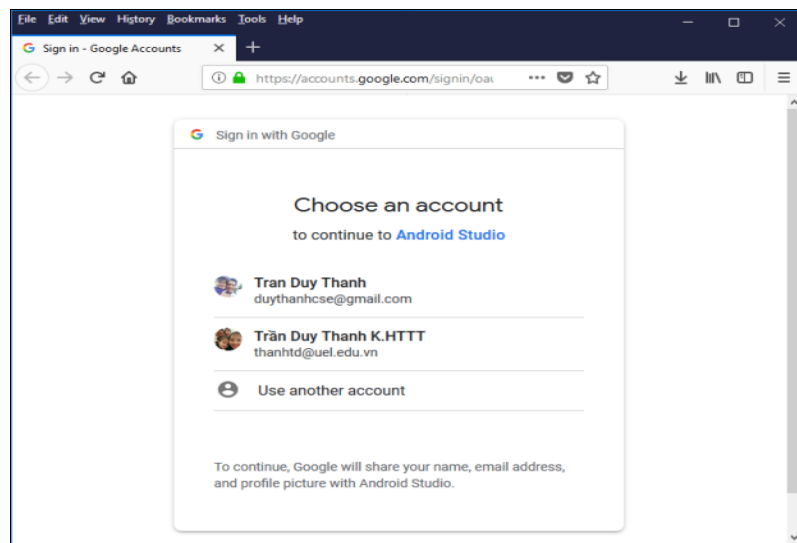
- Bước 1: Connect your app to firebase
- Bước 2: Add the Realtime Database to our app
- Bước 3: Configure Firebase Database Rules
- Bước 4: Write to your database
- Bước 5: Read from your database
- Bước 6: Optional: Configure ProGuard
- Bước 7: Prepare for Launch
- Bước 8: Next steps



Hình 3.13: kết nối vs Firebase

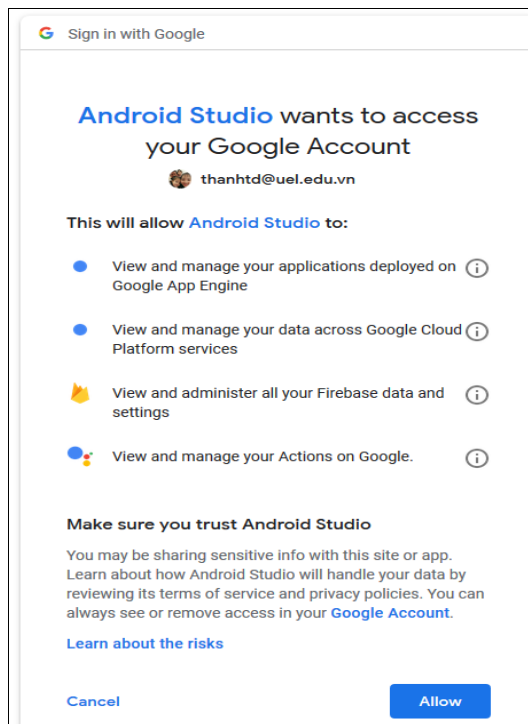
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

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



Hình 3.13: Đăng nhập tài khoản Gmail với Firebase

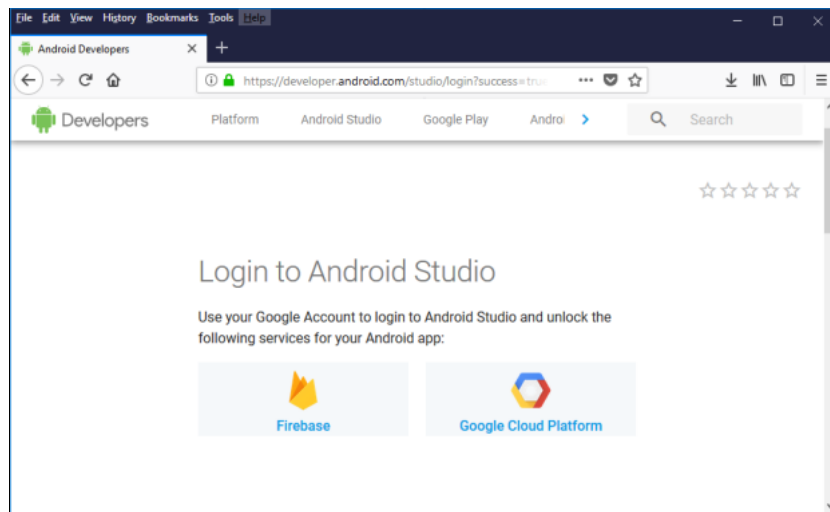
Sau khi chọn Email để đăng nhập, Google yêu cầu ta cung cấp một số quyền hạn như sau:



Hình 3.14: Quyền hạn Google yêu cầu

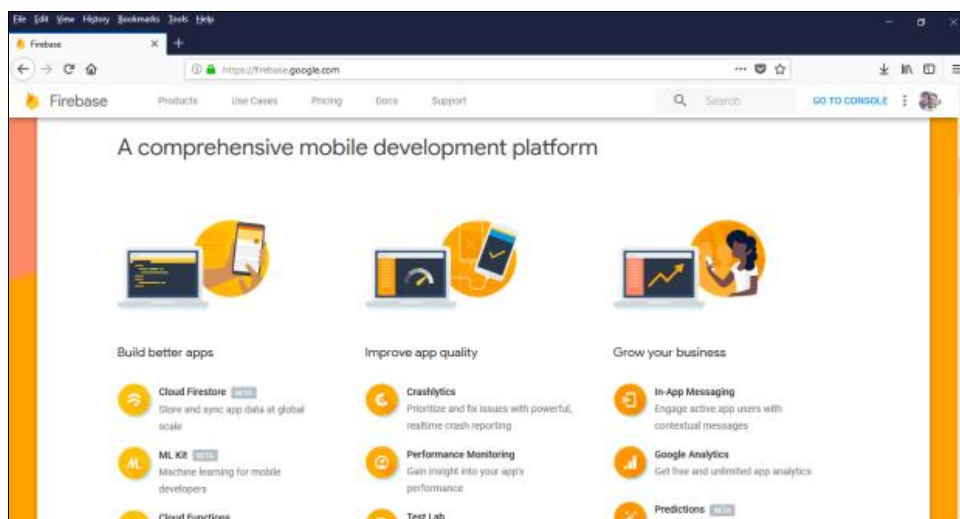
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Ta nhấn Allow để tiếp tục, lúc này màn hình Developer của Android sẽ hiển thị ra:



Hình 3.15: Màn hình chọn Firebase

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

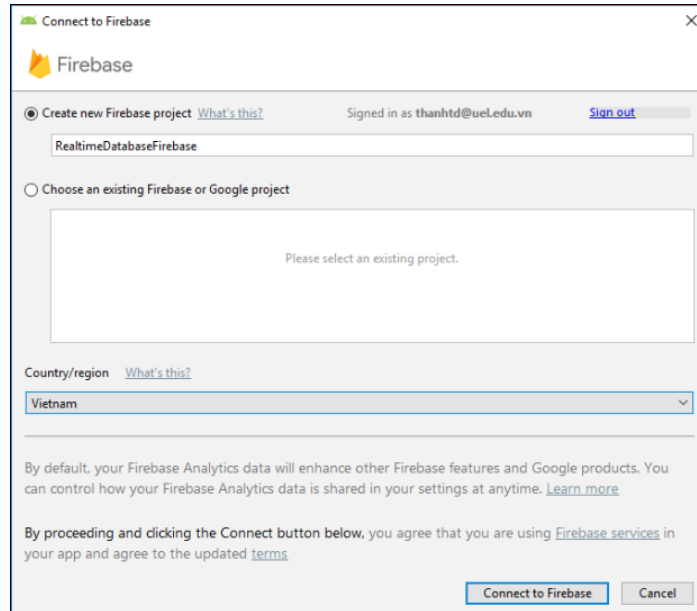


Hình 3.16: Màn hình sau khi chọn Firebase

Chú ý góc phải trên cùng có nút “GO TO CONSOLE”, nhấn vào đây để vào trang quản trị của Firebase.

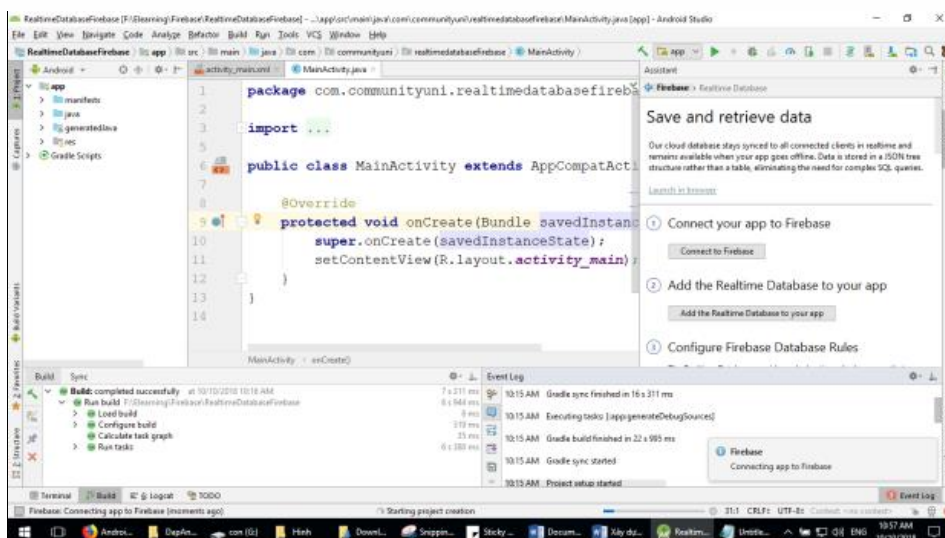
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Lưu ý, Sau khi 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):



Hình 3.17: Màn hình xác nhận kết nối Firebase

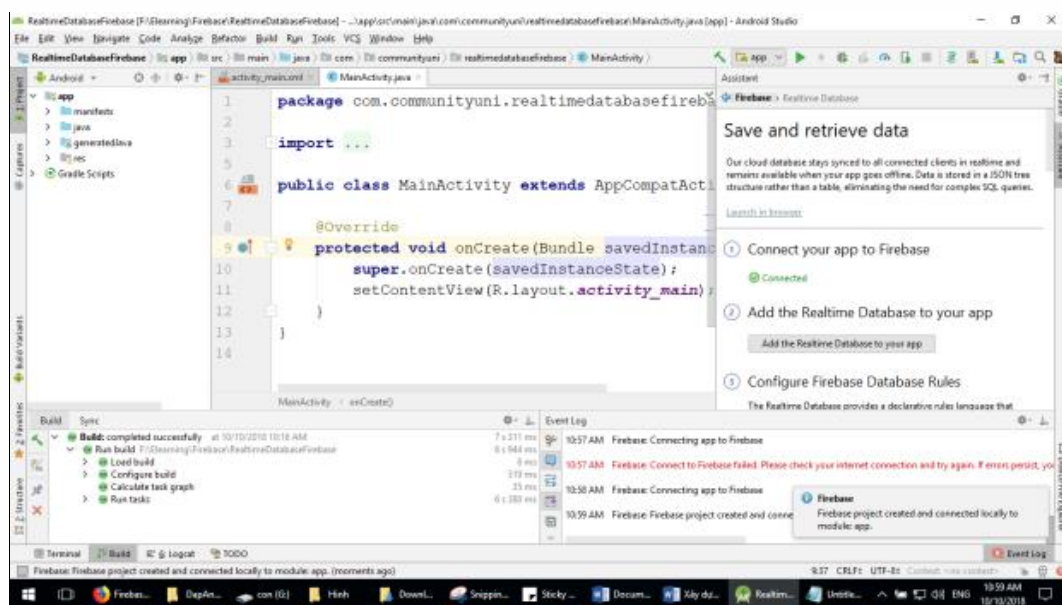
Phía trên là tên ứng dụng “RealtimeDatabaseFirebase”, phần existing chưa có vì ta mới làm. mục Country chọn Vietnam sau đó nhấn nút “Connect to Firebase” để tiến hành kết nối:



Hình 3.18: Màn hình Android Studio kết nối với Firebase

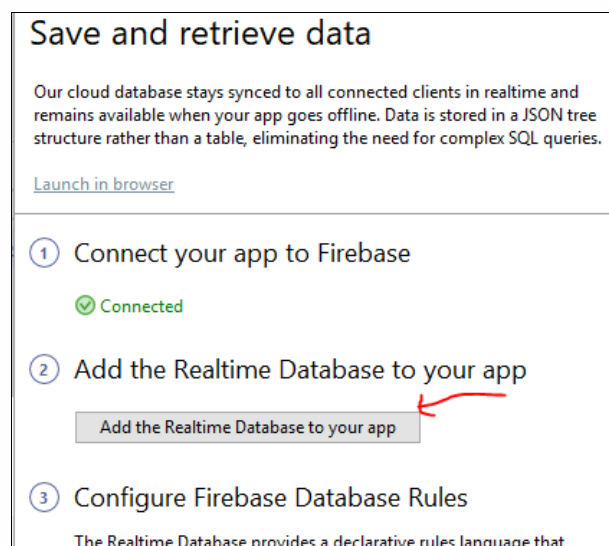
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Ta thấy màn hình nhỏ nhỏ ở góc phải bên dưới không, nó đang báo là đang kết nối ứng dụng tới Firebase (Connecting app to Firebase).



Hình 3.19: Kết quả sau khi kết nối thành công bước 1

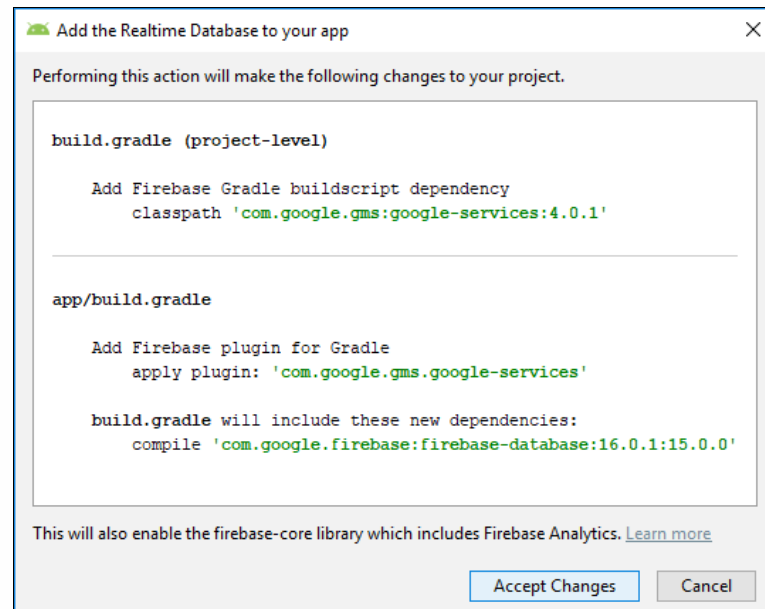
Khi kết nối thành công, bước 1 sẽ được bật đèn xanh như hình trên, và ta nhận được thông báo là thành công ở cửa sổ nhỏ nhỏ ở góc phải bên dưới Android Studio.



Hình 3.20: Màn hình đưa cơ sở dữ liệu thời gian thực vào ứng dụng

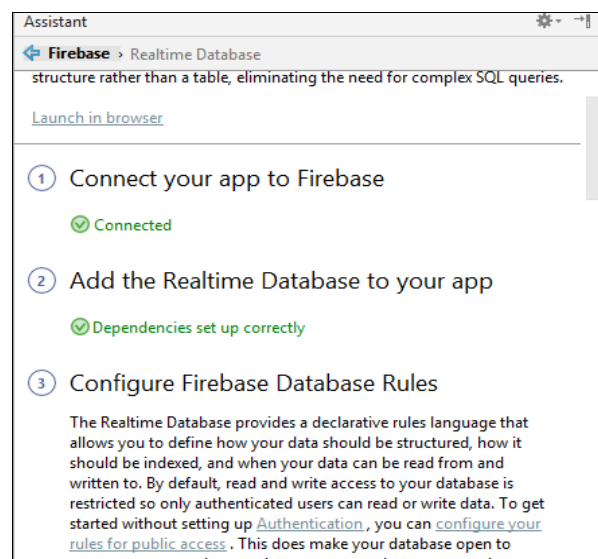
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Khi bước 1 thành công thì ta nhận được thông báo “Connected” màu xanh ở trên tiếp theo ta nhấn vào “Add the Realtime Database to Your app” để đưa cơ sở dữ liệu thời gian thực vào ứng dụng của mình:



Hình 3.21: Màn hình đưa thư viện vào build.gradle

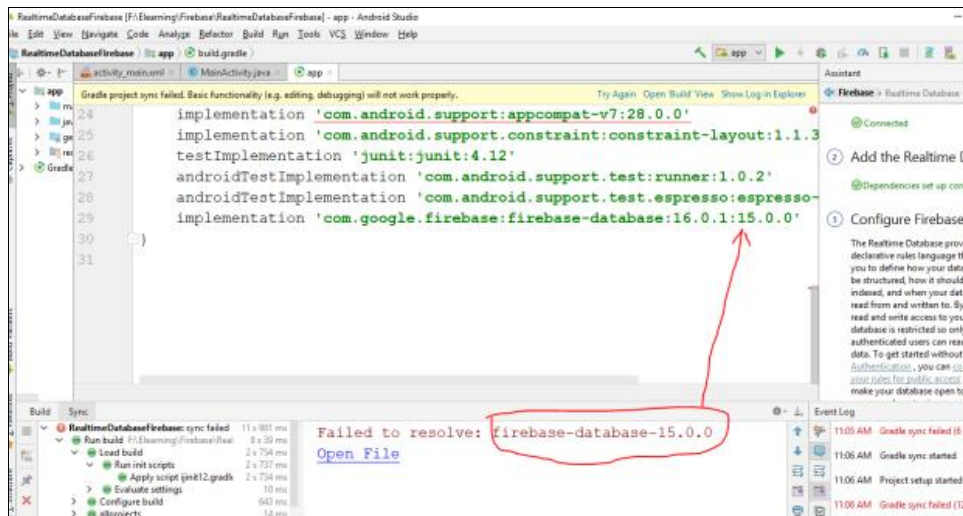
Ta nhấn “Accept Changes” để hệ thống tự động đưa các thư viện vào build.gradle ,nếu thành công ta sẽ thấy màu xanh được bật lên như dưới đây:



Hình 3.22: Màn hình sau khi bước 2 thành công

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Bây giờ ta đợi Android studio Refresh nha, sau đó nếu có thông báo lỗi như dưới này:



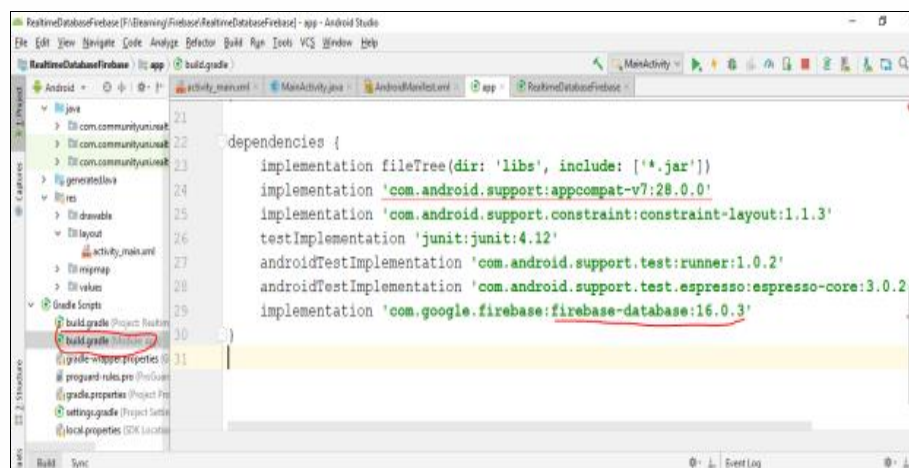
Hình 3.23: Màn hình khi có lỗi xảy ra

Thì ta chỉ việc xóa :15.0.0 đằng sau đi nha (có thử 16.0.1 hay 16.0.3 thấy ổn):

Ta sửa: implementation **'com.google.firebase:firebase-database:16.0.3:15.0.0'**

Thành: implementation **'com.google.firebase:firebase-database:16.0.3'**

Rồi bấm Try Again lại:



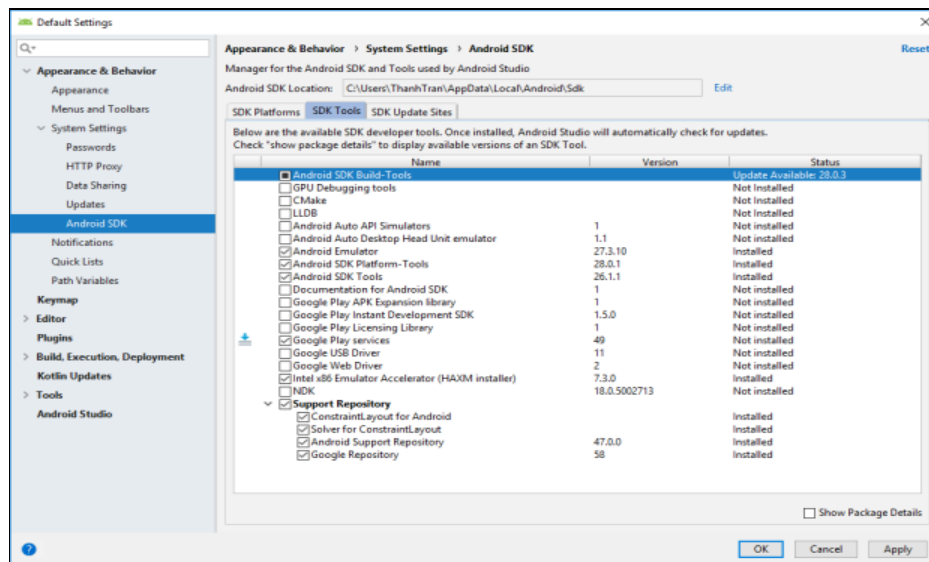
Hình 3.24: Sửa lỗi xảy ra



## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

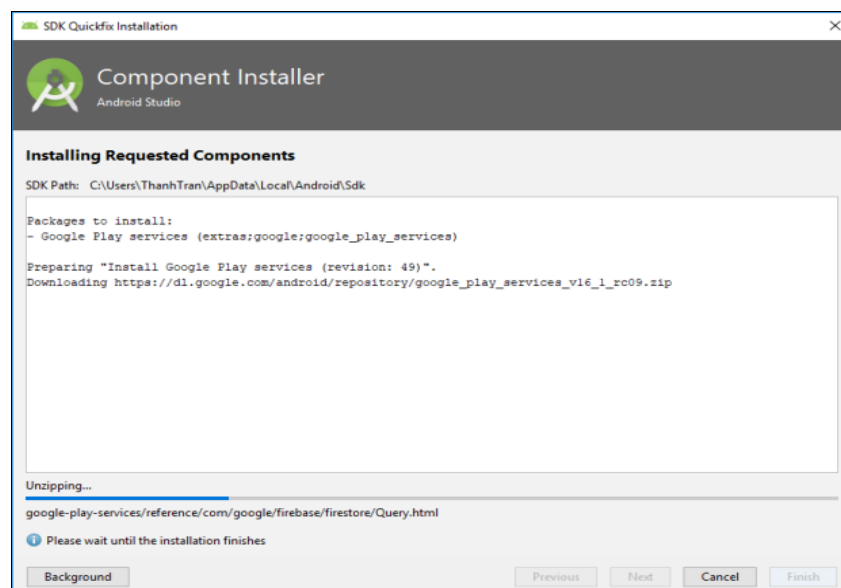
nếu còn lỗi: Thì phải kiểm tra xem, coi chừng Android của bạn chưa cài Google Play Service, các bước kiểm tra và tải như sau:

Vào Menu Tools/ chọn SDK Manager:



Hình 3.25 Màn hình Default Settings

Chọn thẻ SDK Tools/ Tick vào Google Play Services rồi bấm Apply chờ nó tải và cài đặt vào:

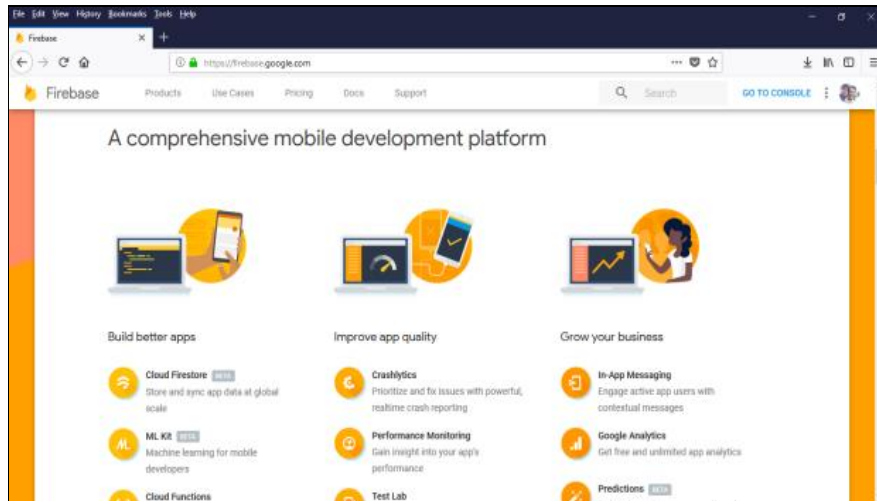


Hình 3.26: Màn hình SDK Quickfix Installation

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

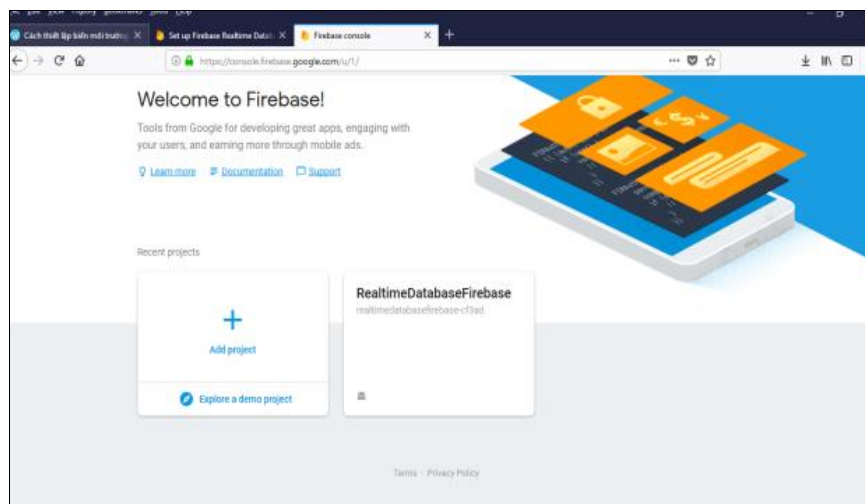
Ta quay lại Website Firebase khi kết nối thành công để kiểm tra cũng như thao tác nha:

Nhìn thấy nút “GO TO CONSOLE” ở góc phải bên trên không? bấm vào nó:



Hình 3.27: Website Firebase

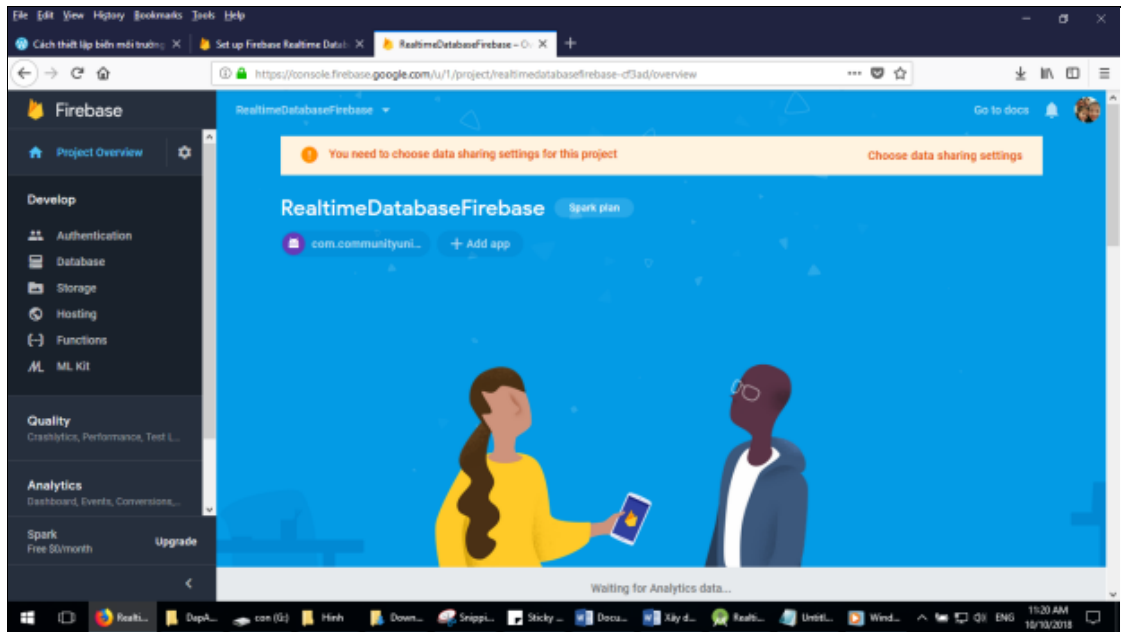
Ta sẽ được dẫn tới màn hình dưới đây:



Hình 3.28: Website Firebase khi kết nối thành công

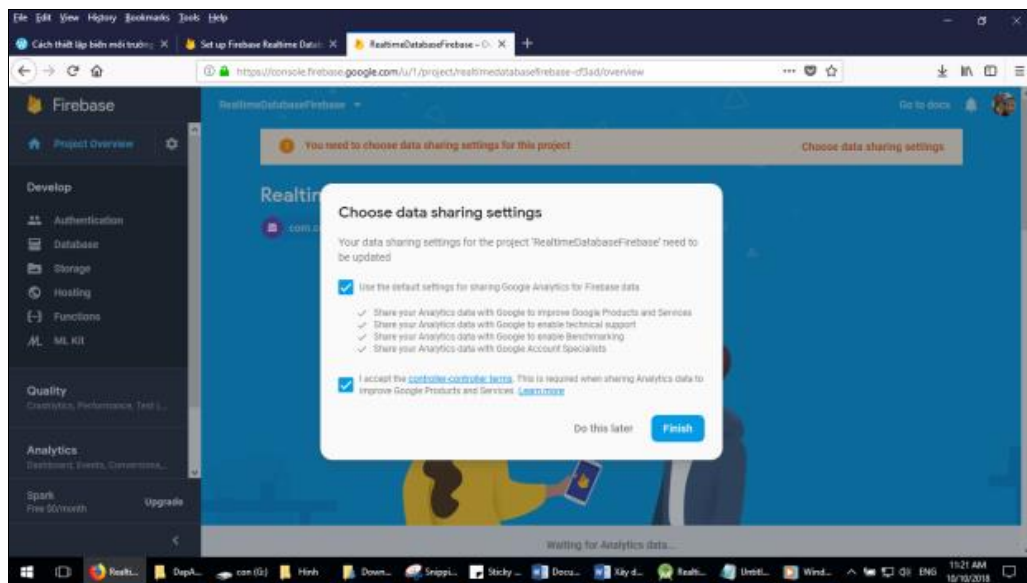
Ta thấy ứng dụng tên là **RealtimeDatabaseFirebase** xuất hiện ở trên không? nhấn vào nó (lưu ý là ở Android Studio kết nối thành công thì ở đây mới tự động thấy nó):

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng



Hình 3.29: Website Firebase

Hệ thống yêu cầu cấu hình chia sẻ dữ liệu, Ta bấm vào “Choose data sharing settings” chỗ dòng cảnh báo bên trên đó:

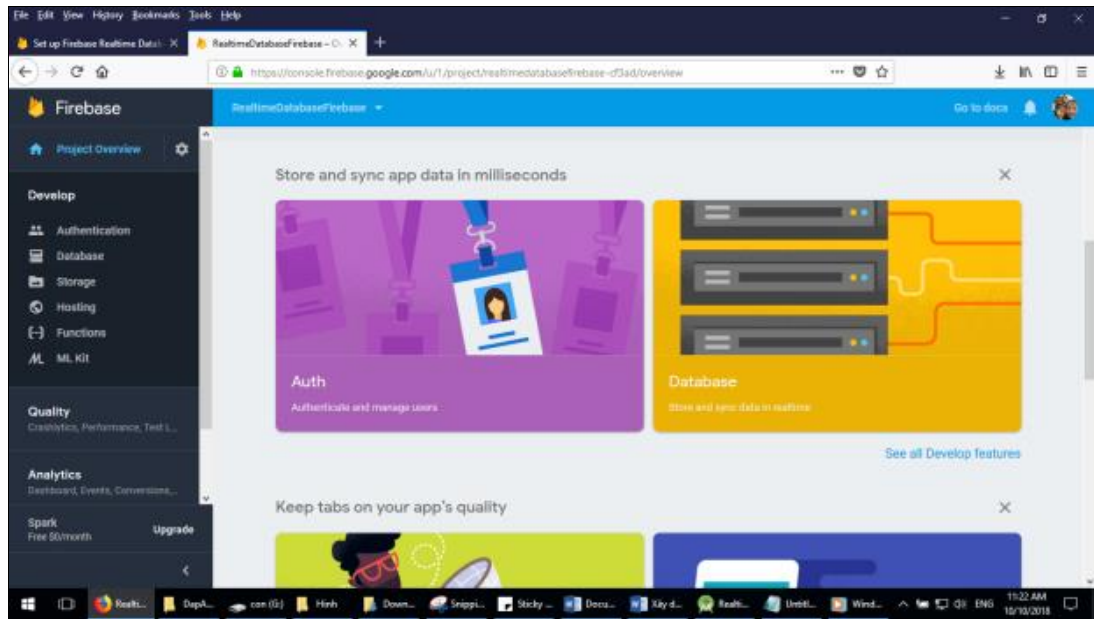


Hình 3.30: Màn hình Choose data sharing settings

Ta tick chọn hết, tick I Accept... rồi nhấn FINISH

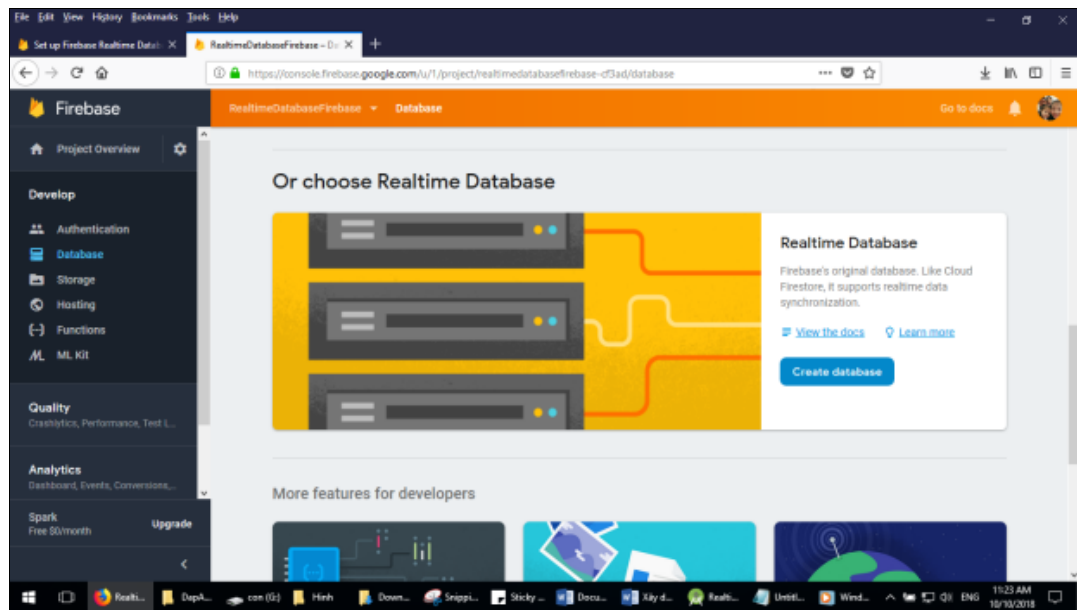
ta chờ hệ thống chạy một chút xíu sẽ xong và ra màn hình dưới đây:

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng



Hình 3.31: Website Firebase

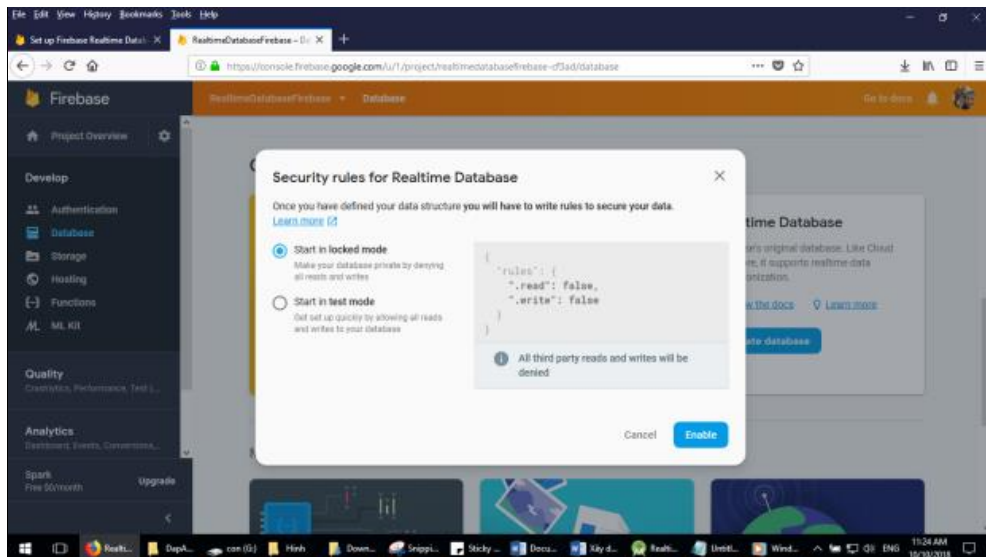
Ta nhìn vào danh sách bên trái màn hình Website — > nhấn vào Database:



Hình 3.32: Website Firebase khi ta bấm vào Database

Sau khi nhấn vào Database, bên nội dung ta kéo xuống nhìn thấy nhóm Realtime Database thì chọn “Create Database”:

## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

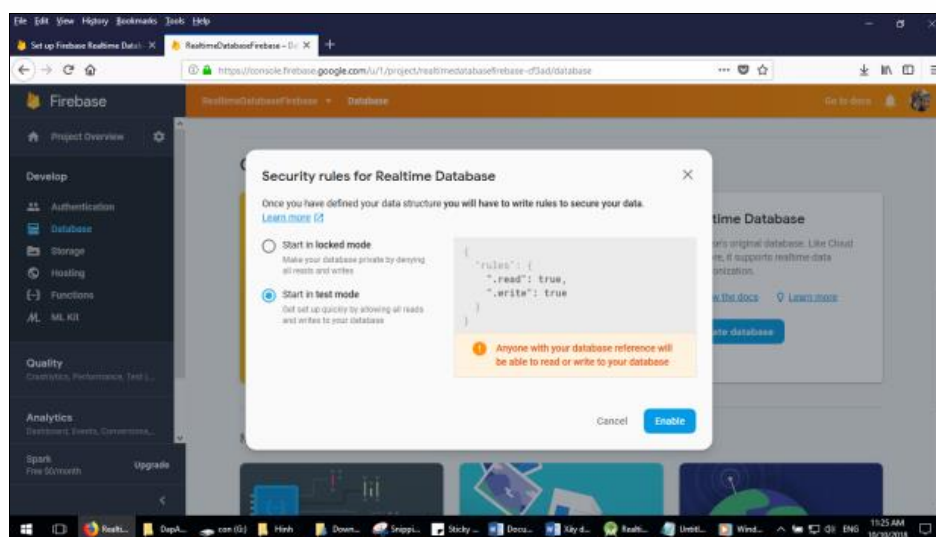


Hình 3.33: Màn hình để chỉnh quyền truy cập

Hệ thống cung cấp 2 mode:

- Start In locked mode, dùng để chạy thực sử (khi đã release sản phẩm)
- Start in Test mode, dùng cho Debug lúc mà ta đang phát triển phần mềm.

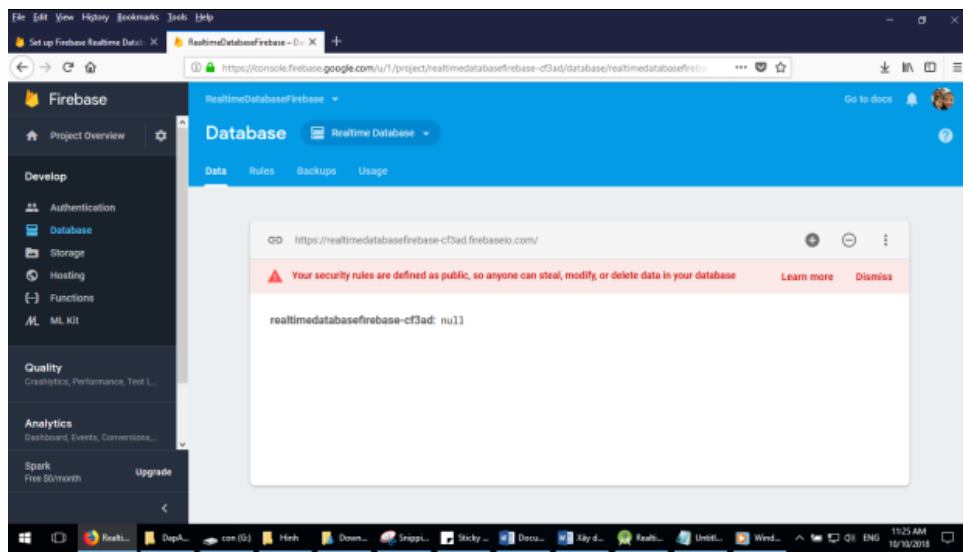
Do ta đang test thì chọn start in test mode, khi nào xong thì chỉnh lại Locked mode sau:



Hình 3.34: Màn hình để chỉnh quyền truy cập

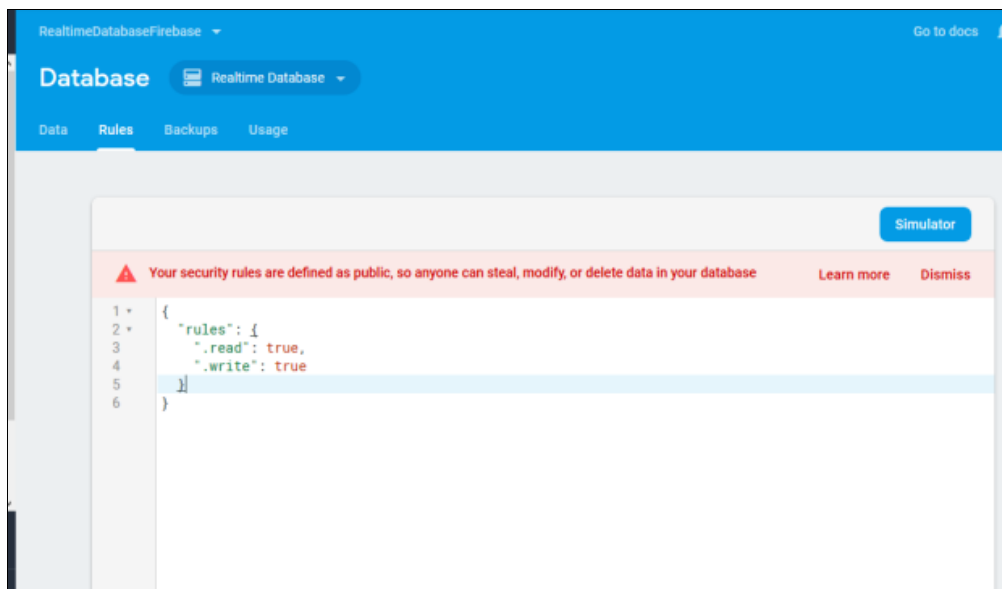
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Chọn Start in test mode xong thì bấm nút Enable nha. Cấu hình thành công ta sẽ có màn hình với thông báo dưới đây:



Hình 3.35: Cấu hình thành công

Nó thông báo về bảo mật, vì ta đang thử nghiệm. Muốn chỉnh lại trong thẻ RULE:

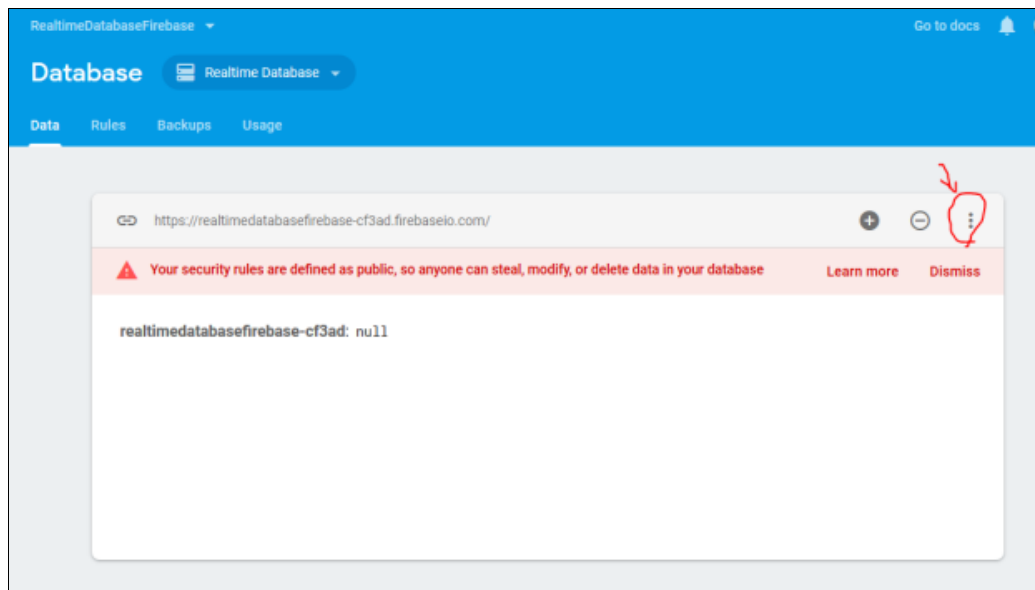


Hình 3.36: Thẻ RULE

Bây giờ ta nhập dữ liệu cho Database Realtime này. Có 2 cách nhập:

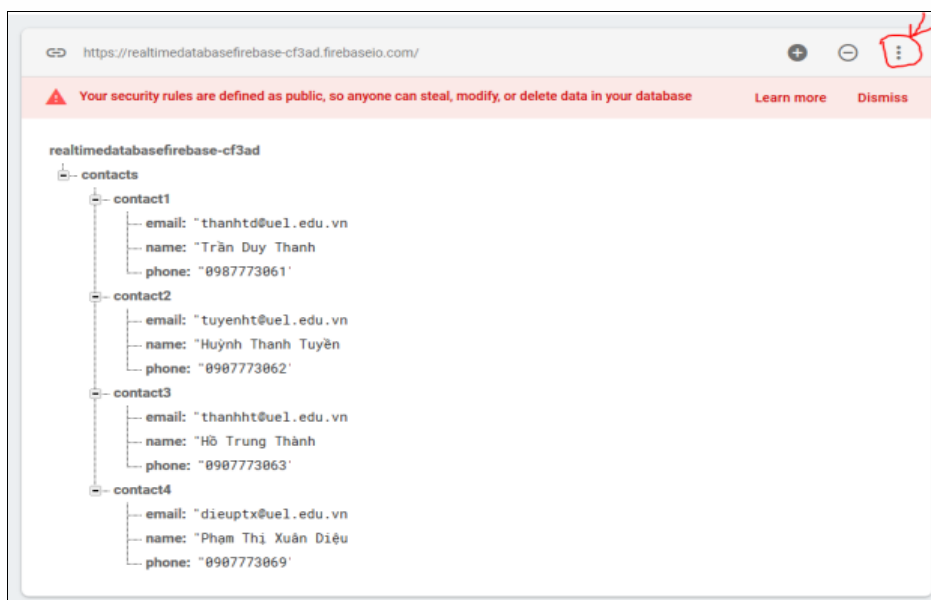
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

- nhập trực tiếp vào màn hình Web
- nhập bằng cách import 1 file JSON theo đúng cấu trúc



Hình 3.37: Thẻ Data

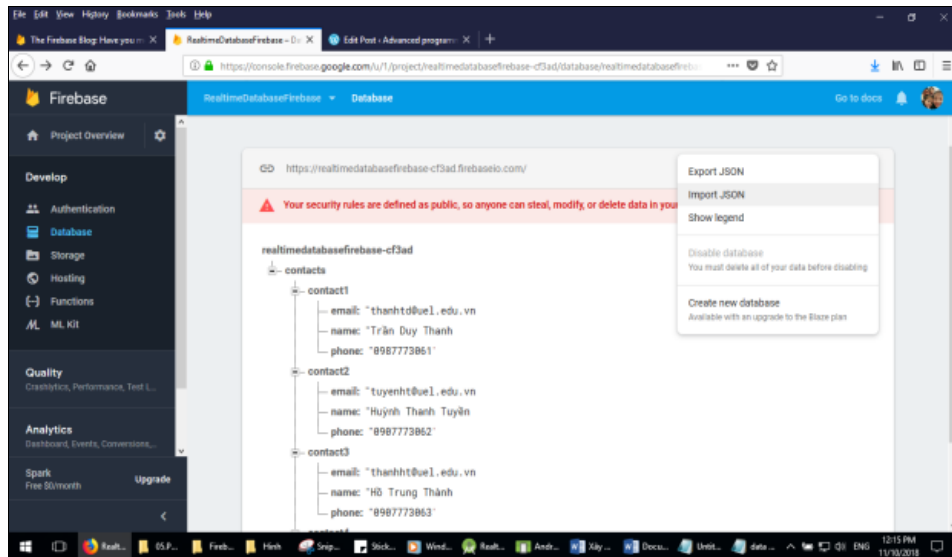
Thấy chữ null ở trên ta click vào chữ null rồi dán dữ liệu JSON theo đúng cấu trúc là OK:



Hình 3.38: Thẻ Data

Hoặc ở góc phải trên cùng nhấn vào ...:

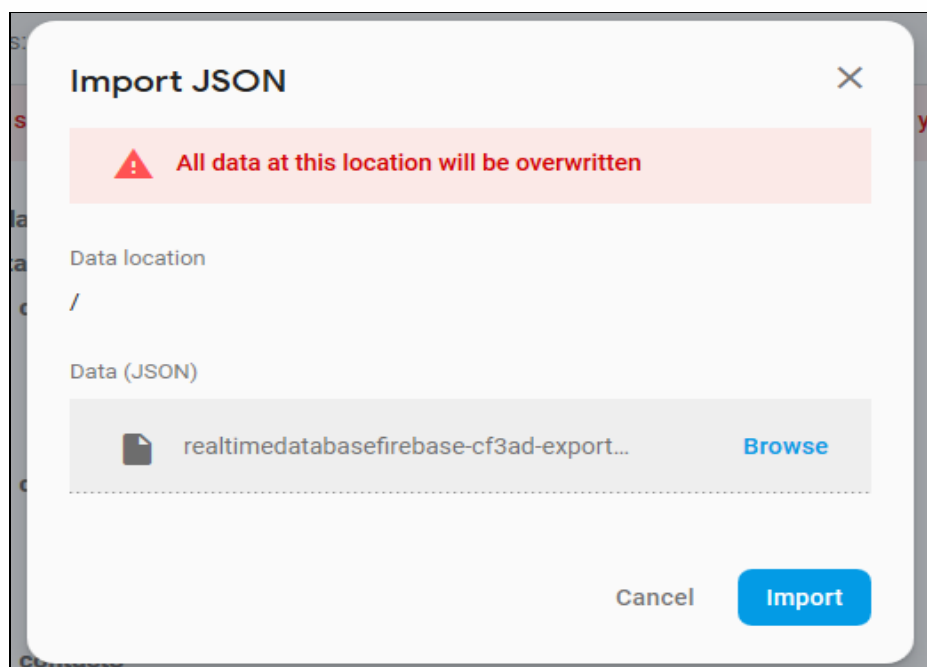
## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng



Hình 3.39: Export JSON và IMPORT JSON

Có chức năng Export JSON là xuất dữ liệu ra file JSON, Import JSON là nhập 1 file dữ liệu JSON ở ngoài và hệ thống Database Realtime của Firebase.

Đây là cấu trúc của file dữ liệu ở trên, bạn lấy mẫu này cho lệ nha, dùng chức năng IMPORT JSON để lưu:



Hình 3.40: Khi ta chọn IMPORT JSON



## Tìm hiểu lập trình cơ sở dữ liệu với android Firebase và ứng dụng

Chọn Browse-> trở tải file tải về được, nhấn IMPORT -> Có kết quả như trên. Như vậy ta đã làm xong bước kết nối Firebase, tạo cơ sở dữ liệu. Bước cuối cùng là ta truy suất Cơ sở dữ liệu Realtime này lên giao diện, thay đổi và quan sát nó cập nhật giao diện tức thời (Realtime).

## Chương 4: Chương trình ứng dụng thử nghiệm

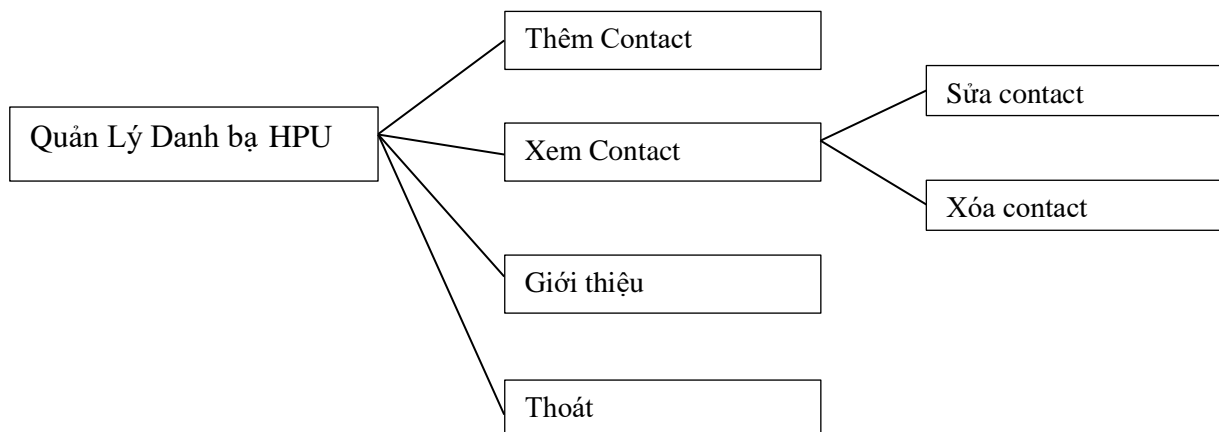
### 4.1. Phát biểu bài toán

Điện thoại Smart Phone chạy hệ điều hành Android ngày càng trở nên không thể thiếu được đối với chúng ta. Chúng vừa là phương tiện giúp chúng ta liên lạc, vừa là phương tiện giúp chúng ta học tập và tra cứu thông tin. Xuất phát từ nhu cầu thực tế, trong thời đại công nghiệp 4.0 hiện nay, thông tin cá nhân: số điện thoại, email, ... là những thông tin quan trọng và cần thiết để các tổ chức và cá nhân liên lạc thông tin với các đối tác. Việc cập nhật các thông tin này cần được thực hiện một cách chính xác và nhanh chóng theo thời gian thực. Người dùng có nhu cầu theo dõi, cập nhật thông tin trực tuyến mọi nơi mọi lúc thông qua smart phone chạy hệ điều hành Android. Các thao tác cần thực hiện:

- Xem danh sách danh bạ điện thoại: Lấy dữ liệu từ cơ sở dữ liệu trên Firebase về hiển thị trên android smart phone.
- Thêm thông tin danh bạ cá nhân : Bổ sung dữ liệu cho vào cơ sở dữ liệu trên Realtime Firebase database từ android smart phone.
- Sửa thông tin danh bạ cá nhân: Sửa dữ liệu đã có trong cơ sở dữ liệu Firebase từ android smart phone.
- Xóa bỏ danh bạ điện thoại cá nhân: Xóa dữ liệu đã có trong cơ sở dữ liệu Firebase từ android smart phone.

Chương trình này sẽ giúp người dùng thực hiện nhu cầu trên một cách dễ dàng.

### 4.2. Sơ đồ chức năng



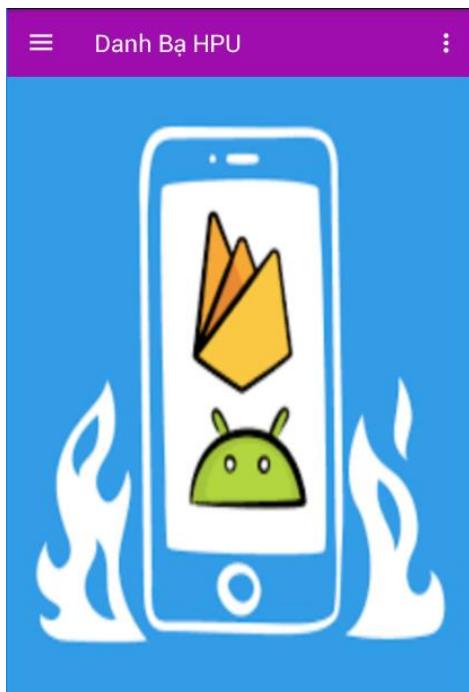
## 4.3. Cơ sở dữ liệu

Thêm Contact

STT	Tên trường	Kiểu trường	Độ rộng	Ghi chú
1	ID	Text		Khóa chính
2	NAME	Text		
3	PHONE	Text		
4	DIACHI	Text		
5	EMAIL	Text		

## 4.4. Giao diện chương trình

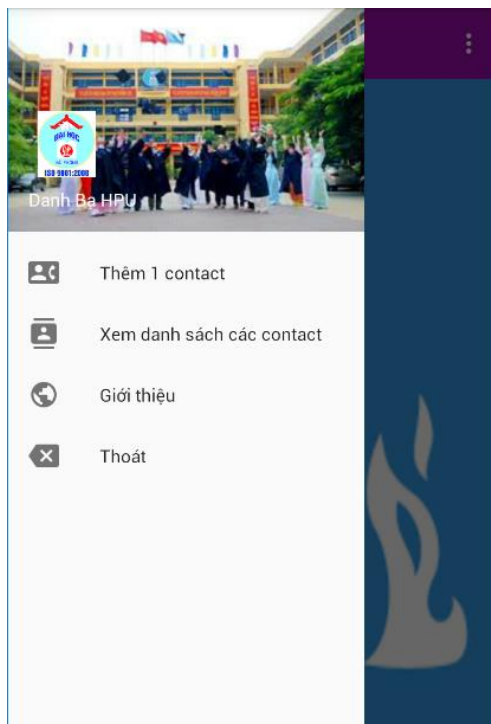
### 4.4.1. Màn hình chính



Khi ta ấn vào ứng dụng Danh Bạ HPU sẽ xuất hiện màn hình chính của chương trình thử nghiệm.

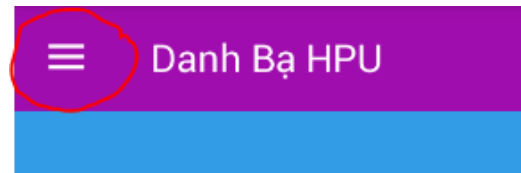
*Hình 4.1: Màn hình chính*

## 4.4.2.Màn hình menu



Hình 4.2:Màn hình menu

Ấn vào kí hiệu ba gạch góc trái trên cùng màn hình chính để ta mở menu cho người dùng có thể lựa chọn.

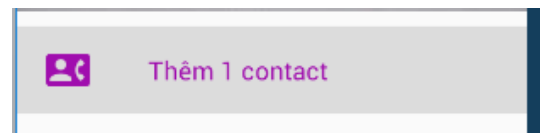


Khi mở menu lựa chọn như bên cạnh ta sẽ thấy có 4 mục Thêm 1 contact, Xem danh sách các contact, Giới thiệu và Thoát.

## 4.4.3.Màn hình Thêm contact

Hình 4.3:Màn hình Thêm Contact

Ta sẽ bắt đầu với mục Thêm 1 contact đầu tiên.



Ở mục Thêm contact ta có thể thấy có các phần nhập thông tin người dùng như Tên, Email, Số điện thoại, Địa chỉ và ảnh.

Để lấy ảnh ta có 3 cách:

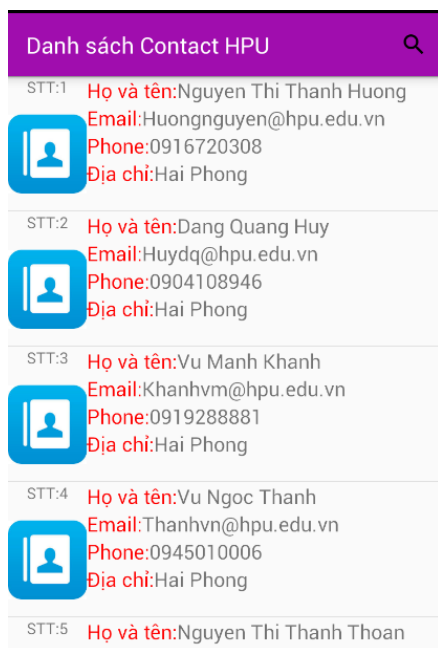
- Lấy ảnh trong anblum máy có sẵn. Ta sẽ dùng nút LẤY ẢNH.
- Chụp ảnh trực tiếp bằng camera thiết bị. Ta sẽ dùng nút CHỤP
- Không cần ảnh, ta sẽ thấy hình mặc định.

Khi ta ấn nút XÓA tất cả dữ liệu ta nhập vào sẽ được xóa hết, tiện cho việc nhập lại thông tin hoặc nhập thêm cho một người dùng khác.

Ta ấn vào nút Thêm thông tin của cá nhân sẽ được đưa trực tiếp lên Firebase và mục Danh sách contact trong ứng dụng.

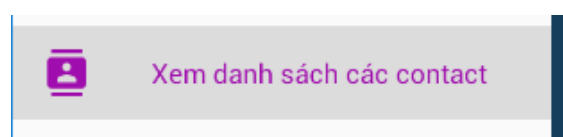
Nút THOÁT để thoát khỏi màn hình.

#### 4.4.4.Màn hình Xem Danh sách Contact



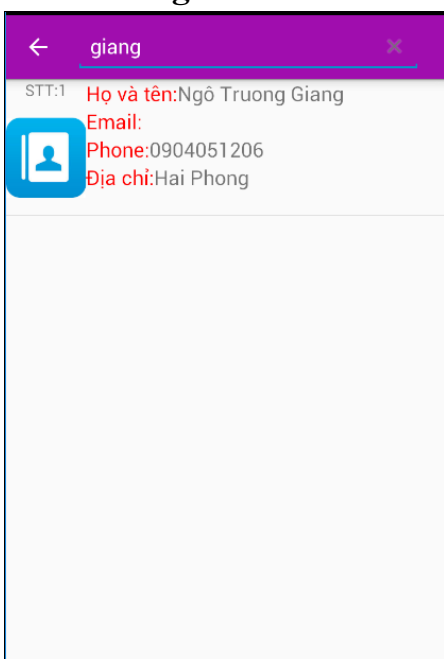
Hình 4.4:Màn hình Xem danh bạ

Chuyển sang màn hình Danh sách Contact HPU ta về màn hình menu chọn.



Khi sang màn hình Xem Danh sách Contact Hpu, ta thấy các danh sách người dùng xếp theo thứ tự thêm lần lượt nhau.

#### 4.4.5.Chức năng tìm kiếm

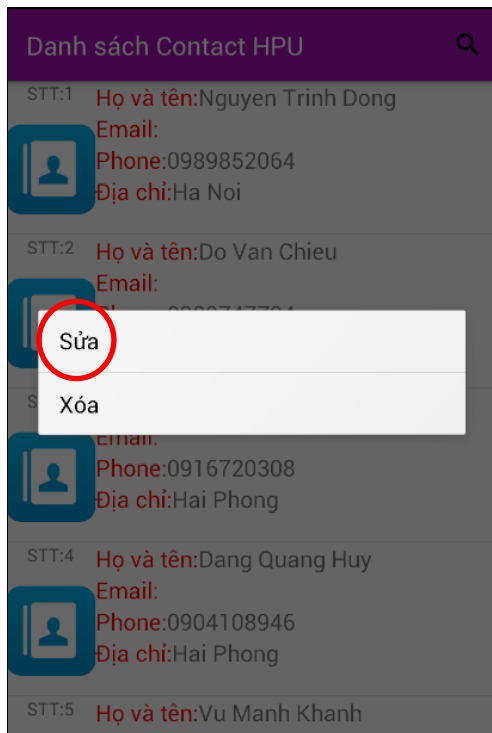


Hình 4.5:Chức năng tìm kiếm theo tên

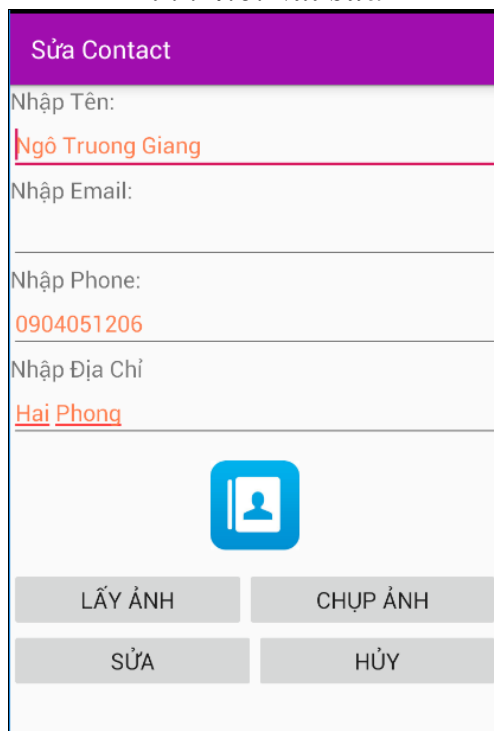
Để tìm kiếm tên mình vừa thêm thành công bên màn hình Thêm Contact sao cho nhanh nhất, ta ấn vào biểu tượng kính lúp phía bên phải góc trên cùng của màn hình Xem Contact để dùng chức năng tìm kiếm.

Chức năng tìm kiếm này được lập trình tìm kiếm theo tên cho dễ nhớ và tiện lợi cho người dùng.VD: ta thường không nhớ số điện thoại.

## 4.4.6.Sửa contact



Hình 4.6:Nút sửa



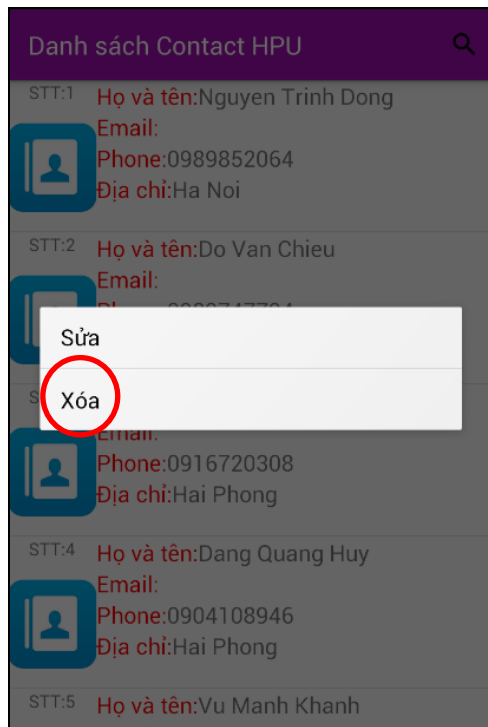
Hình 4.7:Màn hình sửa

Để sửa thông tin cá nhân ta sẽ dùng thao tác ấn giữ vào dòng thông tin cần sửa sẽ hiện hai nút Sửa,Xóa.

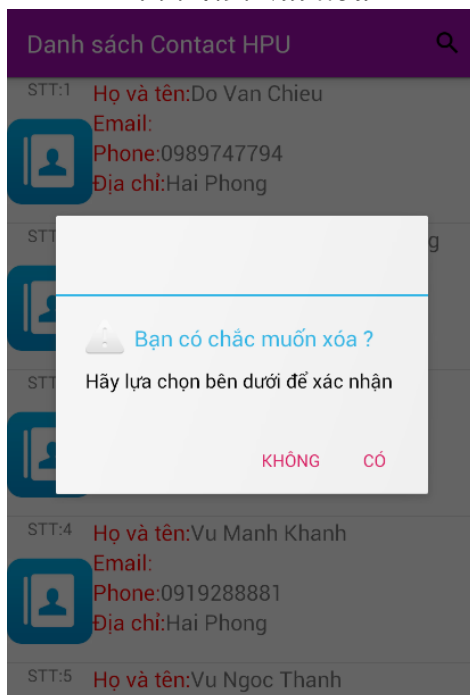
Ta sẽ chọn Sửa lập tức ứng dụng sẽ đưa ta đến màn hình sửa như hình 4.7.

Ở màn hình sửa ta có thể sửa lại thông tin cần sửa,cũng có thể thay ảnh,chụp ảnh như bên màn hình Thêm Contact.

## 4.4.7.Xóa Contact



Hình 4.9:Nút xóa

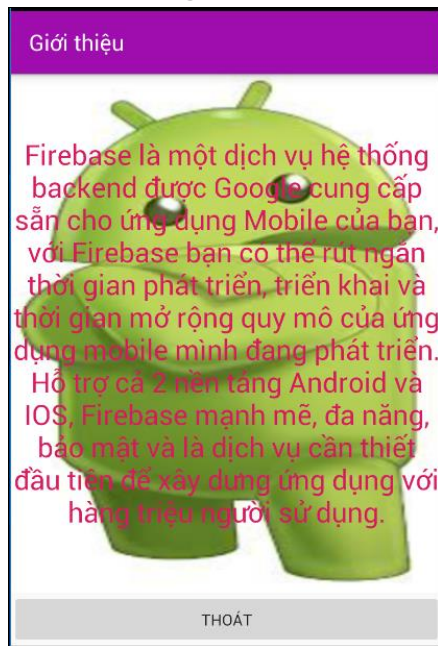


Hình 4.10:Xác nhận Xóa

Nếu ngược lại người dùng muốn xóa thông tin danh bạ của mình trong ứng dụng.Người dùng ấn sử dụng thao tác ấn giữ vào dòng thông tin cần xóa giống như phần Sửa bên trên, ta chọn nút Xóa.

Khi ấn vào nút xóa chương trình sẽ hiển thị phần xác nhận xem người dùng có chắc chắn muốn xóa hay không . Nếu người dùng đồng ý thì ấn CÓ, không đồng ý thì người dùng ấn KHÔNG

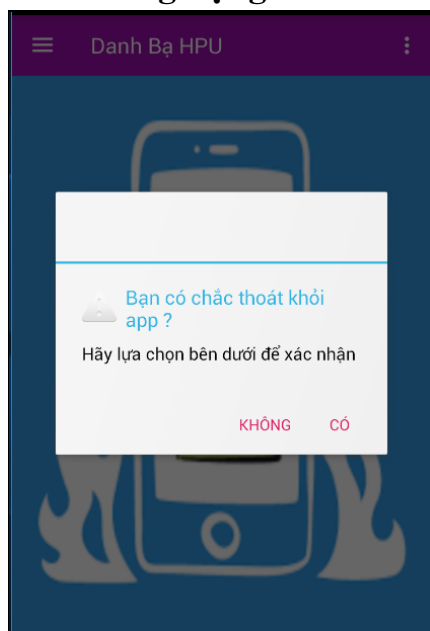
## 4.4.8.Màn hình giới thiệu



Hình 4.11:Màn hình giới thiệu

Ở màn hình này,ta có thể xem thông tin về firebase.Trong tương lai có các phần update ứng dụng sẽ được thông báo thêm ở màn hình giới thiệu.

## 4.4.9 Thoát ứng dụng



Hình 4.12:Xác nhận thoát ứng dụng

Để thoát chương trình ta ấn vào phần màn hình Menu rồi chọn phần **Thoát**:



Ứng dụng sẽ hiển thị phần xác nhận người dùng có muốn thoát khỏi ứng dụng không. Nếu người dùng đồng ý thì ấn **CÓ**. Nếu không muốn thì ấn **KHÔNG**.



## KẾT LUẬN

Trên đây em đã khảo sát trên mặt lý thuyết đối với xây dựng ứng dụng Android. Đồ án hướng tới mục tiêu lập trình cơ sở dữ liệu với Android Firebase và ứng dụng. Trong khoảng thời gian nhất định dành cho việc thực hiện đề tài, nên một số vấn đề vẫn chưa được hoàn chỉnh. Tuy nhiên, đồ án đã đạt được một số kết quả:

- Về lý thuyết: Tìm hiểu, nghiên cứu được cách tạo cơ sở dữ liệu trên Firebase, các kỹ thuật lập trình với cơ sở dữ liệu với Android Firebase và ứng dụng: xây dựng dự án trên Firebase, kiểm tra cấp quyền kết nối Firebase, lấy dữ liệu từ Firebase về Android, tìm kiếm dữ liệu, thêm, sửa, xóa dữ liệu từ Android lên Realtime Firebase database.
- Về thực nghiệm: Sử dụng các kỹ thuật lập trình với cơ sở dữ liệu Firebase để xây dựng được ứng dụng truy xuất cơ sở dữ liệu với các thao tác với dữ liệu như: xem, sửa, thêm, xóa.

Do thời gian hạn chế, nên hiện tại đồ án mới chỉ dừng lại ở thao tác với dữ liệu khi kết nối Firebase. Trong tương lai em sẽ tiếp tục phát triển ứng dụng hoàn chỉnh hơn, thực tế với người dùng hơn.

## TÀI LIỆU THAM KHẢO

### Tài liệu tham khảo trực tuyến

1. <http://khoapham.vn/KhoaPhamTraining/android/snippet/>
2. <https://duythanhcse.wordpress.com/2018/10/11/phan-trien-ung-dung-co-so-du-lieu-thoi-gian-thuc-voi-firebase-phan-1/>