# LỜI MỞ ĐẦU

Ngày nay, các ứng dụng công nghệ thông tin đã được áp dụng trong mọi lĩnh vực nghiên cứu khoa học kĩ thuật cũng như trong đời sống, máy tính trở thành công cụ trợ giúp đắc lực cho con người trong lưu trữ, phân tích và xử lý thông tin.

Ở nước ta, cùng với sự phát triển chung của nền kinh tế, tin học đang ngày được ứng dụng rộng rãi trong công tác quản lý và từng bước khẳng định sức mạnh cũng như vị trí quan trọng của mình. Hướng đến những công nghệ mới nhất cũng như học hỏi tìm hiểu những thông tin mới nhất của thế giới, góp phần nâng cao sự phát triển của đất nước, hòa nhập với cộng đồng thế giới ,…

Được sự phân công của Khoa Công nghệ thông tin, Trường Cao Đẳng Công Nghệ Thông Tin Đà Nẵng, và sự đồng ý của Giảng viên hướng dẫn Thầy Mai Lam em đã thực hiện đề tài “Lập trình ứng dụng Nhà Trọ”. Đề tài gồm có 4 phần: Tổng quan;Kỹ thuật lập trình; Phân tích, thiết kế hệ thống; DEMO ứng dụng.

Tuy nhiên, do thời gian hạn hẹp, mặc dù đã nỗ lực hết sức mình nhưng chắc rằng đồ án khó tránh khỏi thiếu sót. Em rất mong nhận được sự thông cảm, những lời góp ý và chỉ bảo tận tình của quý Thầy Cô và các bạn.

# MỤC LỤC

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT ANDROID 1](#_Toc516522435)

[1.1 Tổng quan về Android 1](#_Toc516522436)

[1.1.1. Giới thiệu hệ điều hành Android 1](#_Toc516522437)

[1.1.2. Lịch sử phát triển 2](#_Toc516522438)

[1.1.3. Giao diện Android 4](#_Toc516522439)

[1.1.4. Ứng dụng Android 5](#_Toc516522440)

[1.1.5. Quản lý bộ nhớ Android 5](#_Toc516522441)

[1.1.6. Nhân Linux 6](#_Toc516522442)

[1.1.7. Lịch nâng cấp 7](#_Toc516522443)

[1.2. Sơ lược về Android Studio 7](#_Toc516522444)

[1.2.1. Cài đặt android studio 8](#_Toc516522445)

[1.2.2. Cấu trúc dự án android studio 13](#_Toc516522446)

[1.2.3. Tạo giao diện chương trình trong android studio 21](#_Toc516522447)

[1.3. Sơ lược về cơ sở dữ liệu Firebase - Firebase Realtime Database 27](#_Toc516522448)

[1.3.1. Tổng quan về Firebase 27](#_Toc516522449)

[1.3.2. Sử dụng Firebase 28](#_Toc516522450)

[CHƯƠNG 2: KHẢO SÁT THỰC TRẠNG 31](#_Toc516522451)

[2.1. App Nhà trọ tốt 31](#_Toc516522452)

[2.2. App Hello Rent 32](#_Toc516522453)

[CHƯƠNG 3: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 34](#_Toc516522454)

[3.1 Bài toán 34](#_Toc516522455)

[3.2. Người dùng 34](#_Toc516522456)

[3.3 Biểu đồ 34](#_Toc516522457)

[3.3.1 Biểu đồ phân rã chức năng 35](#_Toc516522458)

[3.4 Đặc tả Use-case 38](#_Toc516522459)

[3.5 Bảng users 39](#_Toc516522460)

[CHƯƠNG 4: XÂY DỰNG VÀ PHÁT TRIỂN ỨNG DỤNG DEMO 41](#_Toc516522461)

[4.1. Xây dựng giao diện Google Maps 41](#_Toc516522462)

[4.2. Xây dựng giao diện thêm mới nhà trọ 45](#_Toc516522463)

[4.3. Giao diện danh sách nhà trọ 47](#_Toc516522464)

[4.4. Xây dựng giao diện chi tiết nhà trọ 48](#_Toc516522465)

[KẾT LUẬN 49](#_Toc516522466)

[1. Ưu điểm 49](#_Toc516522467)

[2. Nhược điểm: 49](#_Toc516522468)

[3. Các vấn đề đã thực hiện 49](#_Toc516522469)

[4. Hướng phát triển 49](#_Toc516522470)

[TÀI LIỆU THAM KHẢO 51](#_Toc516522471)

[LỜI CẢM ƠN 52](#_Toc516522472)

# DANH MỤC HÌNH

[Hình 1. Lịch sử hệ điều hành Android 3](#_Toc516471494)

[Hình 2: Website download phần mềm Android Studio 8](#_Toc516471495)

[Hình 3. Giao diện khởi động phần mềm Android Studio 9](#_Toc516471496)

[Hình 4. Giao diện AVD Manager của Android Studio 10](#_Toc516471497)

[Hình 5: Giao diện Create Virtual Device của Android Studio 10](#_Toc516471498)

[Hình 6: Giao diện System Image của Android Studio 11](#_Toc516471499)

[Hình 7: Giao diện Android Virtual Device(AVD) 11](#_Toc516471500)

[Hình 8: Giao diện Your Virtual Devices sau khi thêm máy ảo 12](#_Toc516471501)

[Hình 9: Giao diện tạo mới project Android Studio 13](#_Toc516471502)

[Hình 10: Giao diện cài đặt một project Android Studio 14](#_Toc516471503)

[Hình 11: Giao diện chọn loại Activity của Android Studio 15](#_Toc516471504)

[Hình 12: Giao khởi tạo Activity của Android studio 15](#_Toc516471505)

[Hình 13: Giao diện Build Gradle project của Android Studio 16](#_Toc516471506)

[Hình 14: Giao diện sau khi khởi tạo xong project của Android Studio 17](#_Toc516471507)

[Hình 15: Giao diện màn hình làm việc activity\_main.xml 17](#_Toc516471508)

[Hình 16: Giao diện màn hình ở chế độ Android 18](#_Toc516471509)

[Hình 17: Giao diện AndroidMainfest.xml 18](#_Toc516471510)

[Hình 18: Giao diện activity\_main.xml ở dạng text 19](#_Toc516471511)

[Hình 19: Giao diện control theo cấu trúc cây 20](#_Toc516471512)

[Hình 20: Giao diện vùng thiết lập properties 20](#_Toc516471513)

[Hình 21: Giao diện thanh công cụ hỗ trợ 21](#_Toc516471514)

[Hình 22: Giao diện FrameLayout Android studio 22](#_Toc516471515)

[Hình 23: Giao diện LinearLayout theo hàng 22](#_Toc516471516)

[Hình 24: Giao diện LinearLayout theo cột 23](#_Toc516471517)

[Hình 25: Giao diện demo RelativeLayout 23](#_Toc516471518)

[Hình 26: Ví dụ về sự kiện click trong android 25](#_Toc516471519)

[Hình 27: Ví dụ sự kiện click trong android 25](#_Toc516471520)

[Hình 28: Giao diện trang chủ Firebase 28](#_Toc516471521)

[Hình 29: Giao diện bảng điều khiển Firebase 28](#_Toc516471522)

[Hình 30: Giao diện bảng điều khiển Firebase 29](#_Toc516471523)

[Hình 31: Giao diện tạo mới kết nối project 29](#_Toc516471524)

[Hình 32: Sơ đồ hoạt động 35](#_Toc516471525)

[Hình 33.Biểu đồ hoạt động đăng nhập vào hệ thống 36](#_Toc516471526)

[Hình 34. Biểu đồ hoạt động đăng ký 36](#_Toc516471527)

[Hình 35.Biểu đồ chức năng thêm nhà trọ mới 37](#_Toc516471528)

[Hình 36: Biểu đồ chức năng tìm kiếm 37](#_Toc516471529)

[Hình 37.Biểu đồ tuần tự chức năng thêm nhà trọ mới 38](#_Toc516471530)

[Hình 38.Biểu đồ tuần tự chức năng sửa thông tin nhà trọ 38](#_Toc516471531)

[Hình 39.Biểu đồ tuần tự chức năng xóa nhà trọ 39](#_Toc516471532)

[Hình 41: Giao diện chính của ứng dụng 41](#_Toc516471533)

[Hình 42: Thanh công cụ tìm kiếm 42](#_Toc516471534)

[Hình 43: Hiển thị vị trí của tôi 43](#_Toc516471535)

[Hình 44: Giao diện hiển thị các nút công cụ 44](#_Toc516471536)

[Hình 45: Giao diện thêm mới nhà trọ 45](#_Toc516471537)

[Hình 46: Giao diện danh sách nhà trọ 47](#_Toc516471538)

[Hình 47: Giao diện chi tiết nhà trọ 48](#_Toc516471539)

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT ANDROID

## 1.1 Tổng quan về Android

### 1.1.1. Giới thiệu hệ điều hành Android

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 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 tháng 10 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 đông 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 thiết bị, băng 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 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 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 chạy trên điện thoại và máy tính bảng, Android đã cuất 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 công 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 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ sự kiên liên quan đến bằng phát minh, góp mặt trong cái gọi là “cuộc chiến điện thoại thông minh” giữa các công ty công nghệ.

### 1.1.2. Lịch sử phát triển

Tổng công ty Android (Android, Inc.) được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin (đồng sáng lập công ty Danger),[20] Rich Miner (đồng sáng lập Tổng công ty Viễn thông Wildfire), Nick Sears (từng là Phó giám đốc T-Mobile), và Chris White (trưởng thiết kế và giao diện tại WebTV) để phát triển, theo lời của Rubin, "các thiết bị di động thông minh hơn có thể biết được vị trí và sở thích của người dùng". Dù những người thành lập và nhân viên đều là những người có tiếng tăm, Tổng công ty Android hoạt động một cách âm thầm, chỉ tiết lộ rằng họ đang làm phần mềm dành cho điện thoại di động. Trong năm đó, Rubin hết kinh phí. Steve Perlman, một người bạn thân của Rubin, mang cho ông 10.000 USD tiền mặt nhưng từ chối tham gia vào công ty.

Google mua lại Tổng công ty Android vào ngày 17 tháng 8 năm 2005, biến nó thành một bộ phận trực thuộc Google. Những nhân viên của chủ chốt của Tổng công ty Android, gồm Rubin, Miner và White, vẫn tiếp tục ở lại công ty làm việc sau thương vụ này. Vào thời điểm đó không có nhiều thông tin về Tổng công ty, nhưng nhiều người đồn đoán rằng Google dự tính tham gia thị trường điện thoại di động sau bước đi này. Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động phát triển trên nền nhân Linux. Google quảng bá nền tảng này cho các nhà sản xuất điện thoại và các nhà mạng với lời hứa sẽ cung cấp một hệ thống uyển chuyển và có khả năng nâng cấp. Google đã liên hệ với hàng loạt hãng phần cứng cũng như đối tác phần mềm, bắn tin cho các nhà mạng rằng họ sẵn sàng hợp tác với các cấp độ khác nhau.

Ngày càng nhiều suy đoán rằng Google sẽ tham gia thị trường điện thoại di động xuất hiện trong tháng 12 năm 2006. Tin tức của BBC và Nhật báo phố Wall chú thích rằng Google muốn đưa công nghệ tìm kiếm và các ứng dụng của họ vào điện thoại di động và họ đang nỗ lực làm việc để thực hiện điều này. Các phương tiện truyền thông truyền thống lẫn online cũng viết về tin đồn rằng Google đang phát triển một thiết bị cầm tay mang thương hiệu Google. Một vài tờ báo còn nói rằng trong khi Google vẫn đang thực hiện những bản mô tả kỹ thuật chi tiết, họ đã trình diễn sản phẩm mẫu cho các nhà sản xuất điện thoại di động và nhà mạng. Tháng 9 năm 2007, InformationWeek đăng tải một nghiên cứu của Evalueserve cho biết Google đã nộp một số đơn xin cấp bằng sáng chế trong lĩnh vực điện thoại di động.

Ngày 5 tháng 11 năm 2007, Liên minh thiết bị cầm tay mở (Open Handset Alliance), một hiệp hội bao gồm nhiều công ty trong đó có Texas Instruments, Tập đoàn Broadcom, Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile được thành lập với mục đích phát triển các tiêu chuẩn mở cho thiết bị di động. Cùng ngày, Android cũng được ra mắt với vai trò là sản phẩm đầu tiên của Liên minh, một nền tảng thiết bị di động được xây dựng trên nhân Linux phiên bản 2.6. Chiếc điện thoại chạy Android đầu tiên được bán ra là HTC Dream, phát hành ngày 22 tháng 10 năm 2008. Biểu trưng của hệ điều hành Android mới là một con rôbốt màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.

Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước. Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 Cupcake (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6 Donut (bánh vòng). Phiên bản mới nhất hiện nay là 5.0 Lollipop. Vào năm 2010, Google ra mắt loạt thiết bị Nexus—một dòng sản phẩm bao gồm điện thoại thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên, Nexus One. Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và phần mềm mới nhất của Android.

Năm 2014, Google công báo Android Wear, hệ điều hành dành cho các thiết bị đeo được.



Hình . Lịch sử hệ điều hành Android

### 1.1.3. Giao diện Android

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 hiên thị ngang tuỳ theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng 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 thương 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 mẩu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được băng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính Android có thể tuỳ 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 hính chính, thậm chí bắt chước hình dáng của hệ điều hành khác chẳng hạn như Window Phone. 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 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 theo tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ 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.

### 1.1.4. Ứng dụng Android

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.

### 1.1.5. Quản lý bộ nhớ Android

Vì các thiết bị Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa mức 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.

### 1.1.6. Nhân 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 tƣơng 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.

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 KroahHartman, 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 đổi 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, nhắm 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 kho 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 không hỗ trợ nhiều công cụ GNU, trong đó có glibc.

### 1.1.7. Lịch nâng cấp

Google đưa ra các bản nâng cấp 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 nâng cấp lớn mới nhất là Android 8.0 Oreo

So với các hệ điều hành cạnh tranh khác, như iOS, các bản nâng cấp Android thường mất thời gian lâu hơn để đến với các thiết bị. Với những thiết bị không thuộc dòng Nexus, các bản nâng cấp 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 nâng cấp 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 nâng cấp 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 nâng cấp 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 nâng cấp. Sự chậm trễ còn được đóng góp bởi nhà mạng, sau khi nhận được bản nâng cấp 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 nâng cấp 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 phƣơng cách phân phối bản nâng cấp trở nên phức tạp chính vì những nhà sản xuất và nhà mạng đã cố tình làm nó như thế.

## 1.2. 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.

### 1.2.1. 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

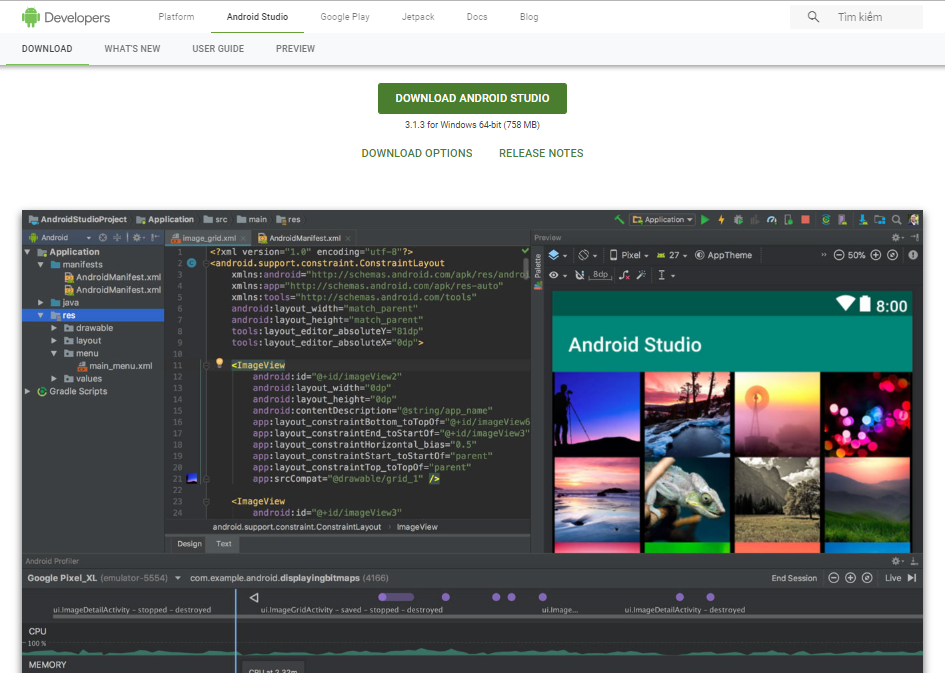
- Độ phân giải tối thiếu 1280 x 800

- Java Development Kit ()

b. Phần mềm android studio:

- Vào đường dẫn: <http://developer.android.com/sdk/index.html>

- Click download để tải bản mới nhất về:



Hình : Website download phần mềm Android Studio

- Sau khi tải về thì mở file lên :



Hình 3. Giao diện cài đặt Android Studio

- Tiếp tục chọn next và agree cho đến khi hoàn tất.

- Đây là màng hình khởi động:

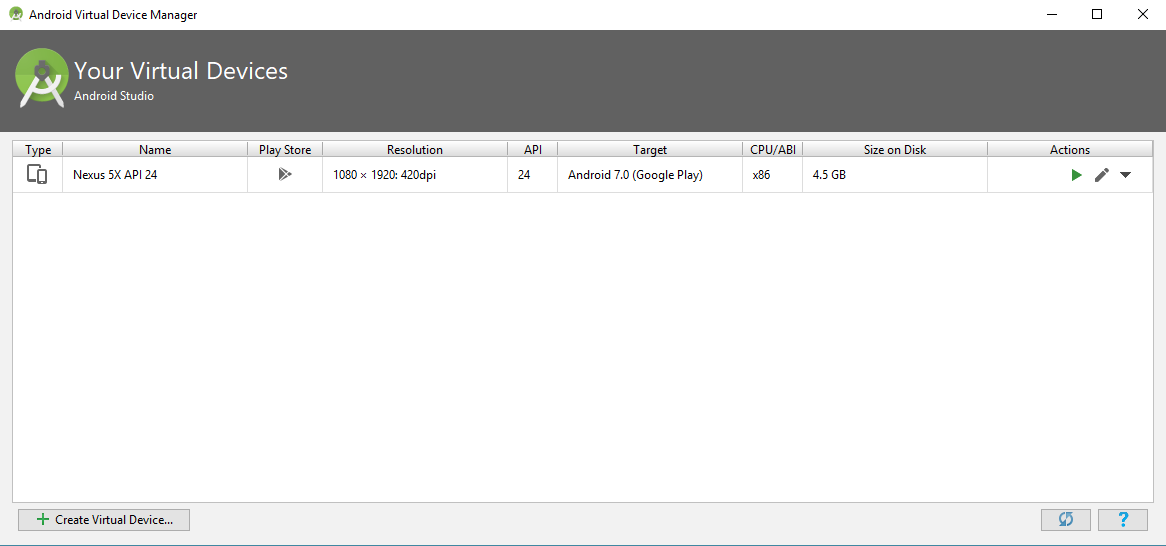


Hình . Giao diện khởi động phần mềm Android Studio

- Các bước cài đặt thiết bị ảo trong android studio

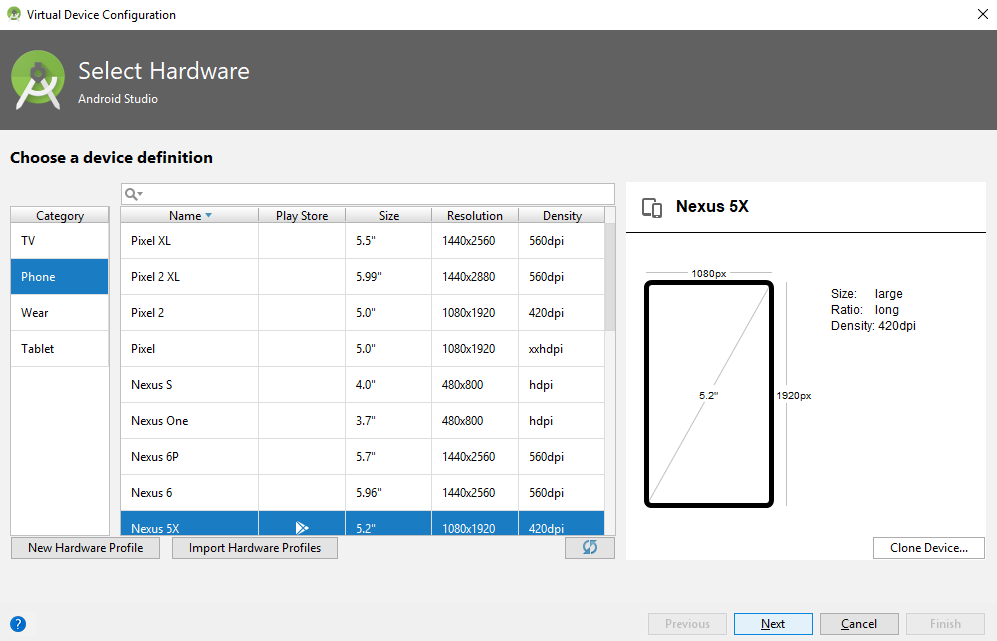
Máy ảo Android là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Trong Android Studio có cung cấp cho chúng ta một máy ảo Android mặc định là Android Virtual Device viết tắt là AVD.

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



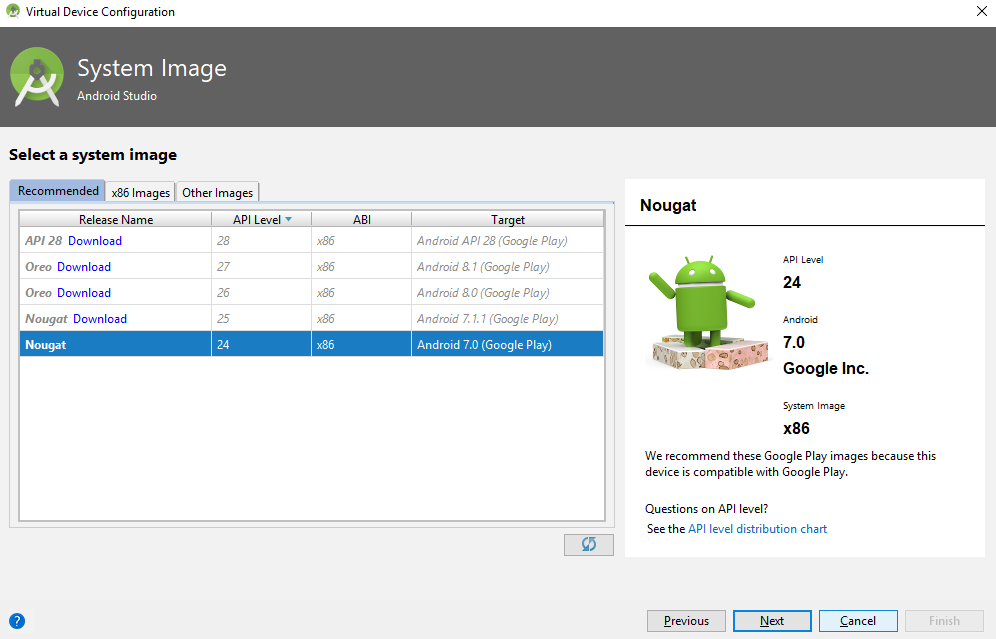
Hình . Giao diện AVD Manager của Android Studio

Ở đây ta đang có sẵn một cái máy ảo, ta sẽ tạo thêm một cái máy ảo nữa bằng cách ấn vào nút Create Virtual Device ở góc dƣới cùng bên trái. Sau đó sẽ hiển thị ra một cửa sổ nữa:



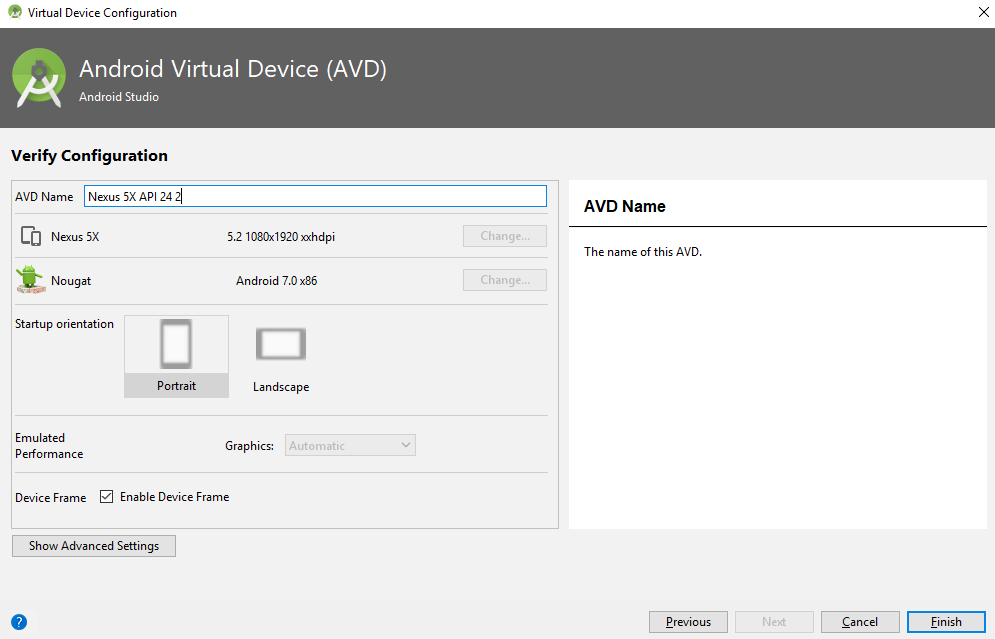
Hình : Giao diện Create Virtual Device của Android Studio

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:



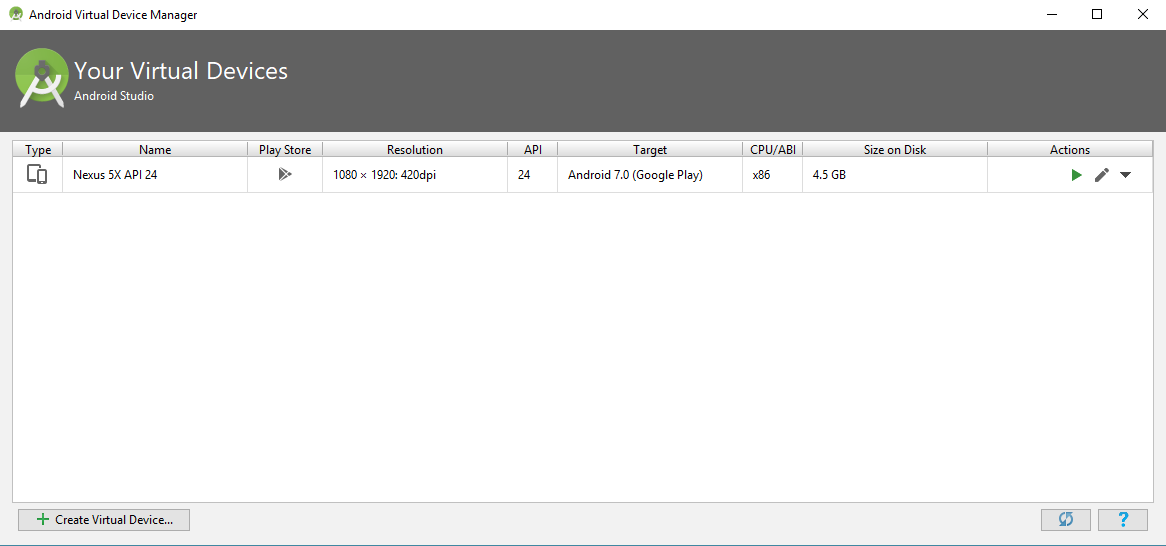
Hình : Giao diện System Image của Android Studio

Ở đây các ta chọn phiên bản hệ điều hành muốn cài sau đó ấn Next để tiếp tục:



Hình : Giao diện Android Virtual Device(AVD)

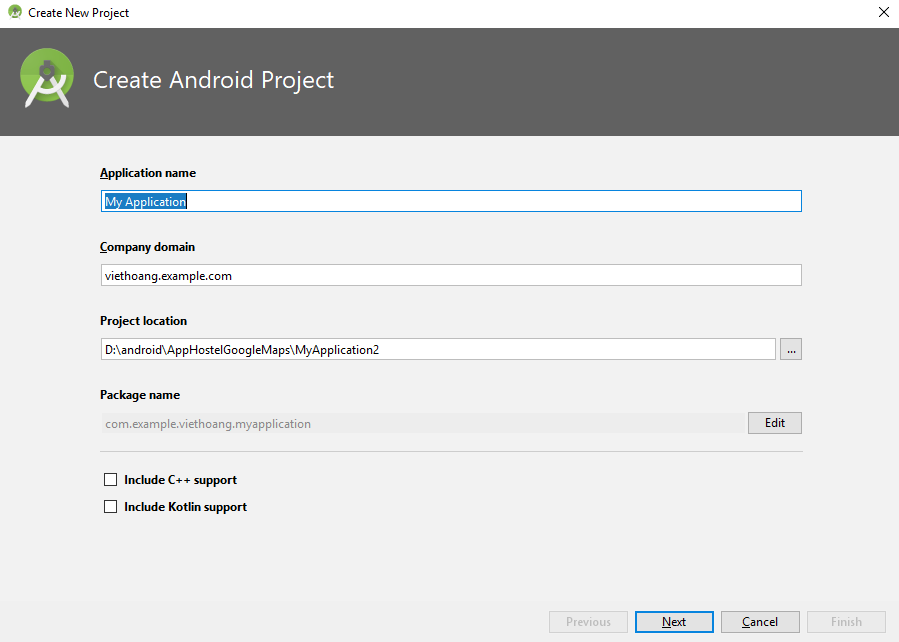
Ta kéo xuống phía dưới sẽ thấy thêm phần tùy chỉnh RAM, bộ nhớ trong, thẻ nhớ,... Sau khi tùy chỉnh theo ý muốn ấn "Finish" để tiến hành tạo máy ảo và chờ một lúc để Android Studio lưu thông tin máy ảo. Sau khi lưu xong ta sẽ thấy trong cửa sổ AVD Manager có thêm một cái máy ảo nữa:



Hình : Giao diện Your Virtual Devices sau khi thêm máy ảo

### 1.2.2. Cấu trúc dự án android studio

a. Tạo mới một project.



Hình : Giao diện tạo mới project Android Studio

- **Application Name**: Tên ứng dụng muốn đặt

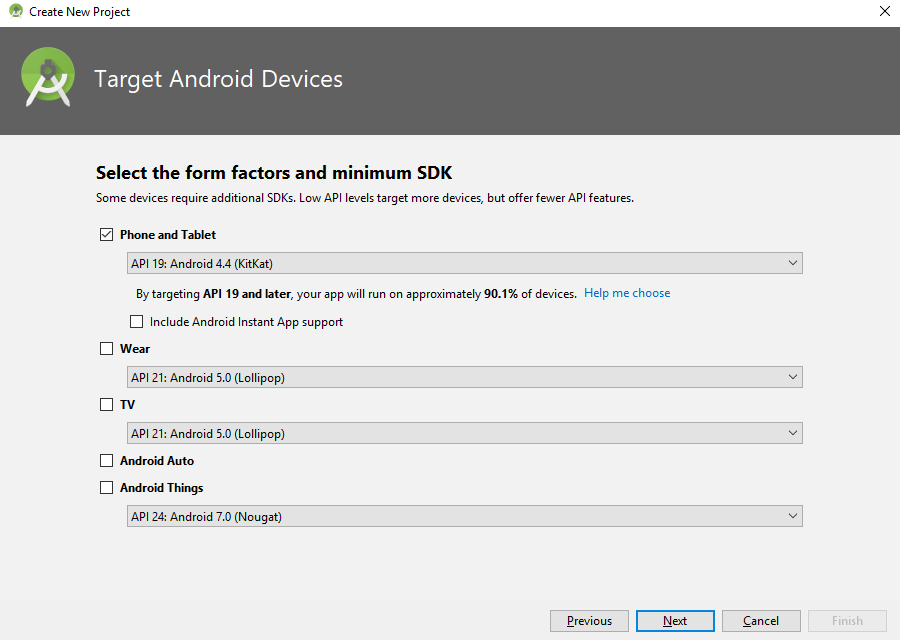
- **Company Domain**: Tên domain công ty, thường được dùng để kết hợp với tên Application để tạo thành Package (chú ý viết thƣờng hết và có ít nhất 1 dấu chấm).

- **Package name**: Nó sẽ tự động nối ngược Company Domain với Application name

- **Project location**: Là nơi lưu trữ ứng dụng.

Sau đó nhấp **Next** để tiếp tục.

b. Cài đặt một project



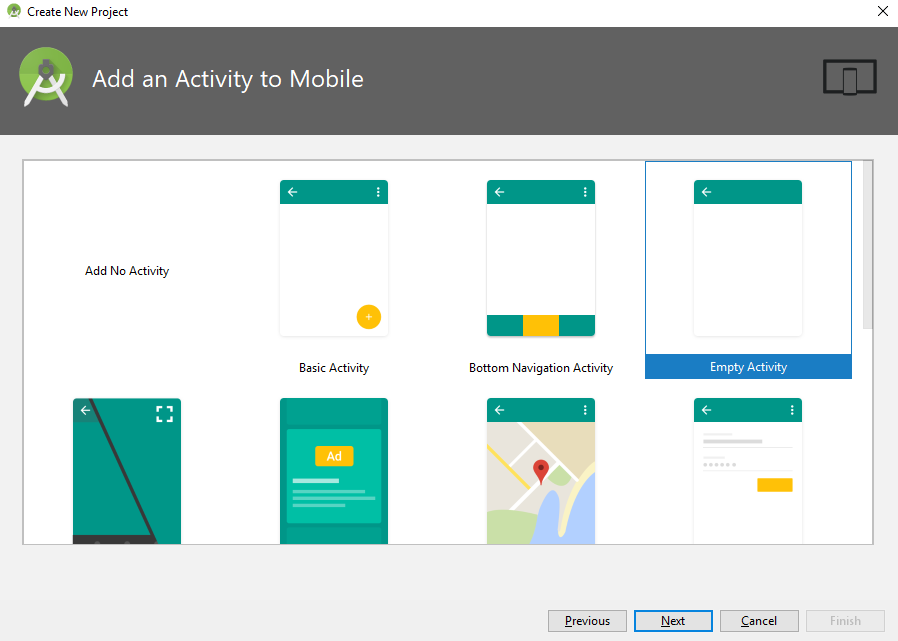
Hình : Giao diện cài đặt một project Android Studio

- 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).

- Ở mục Minium SDK, quy định phiên bản android tối thiếu để chạy ứng dụng.

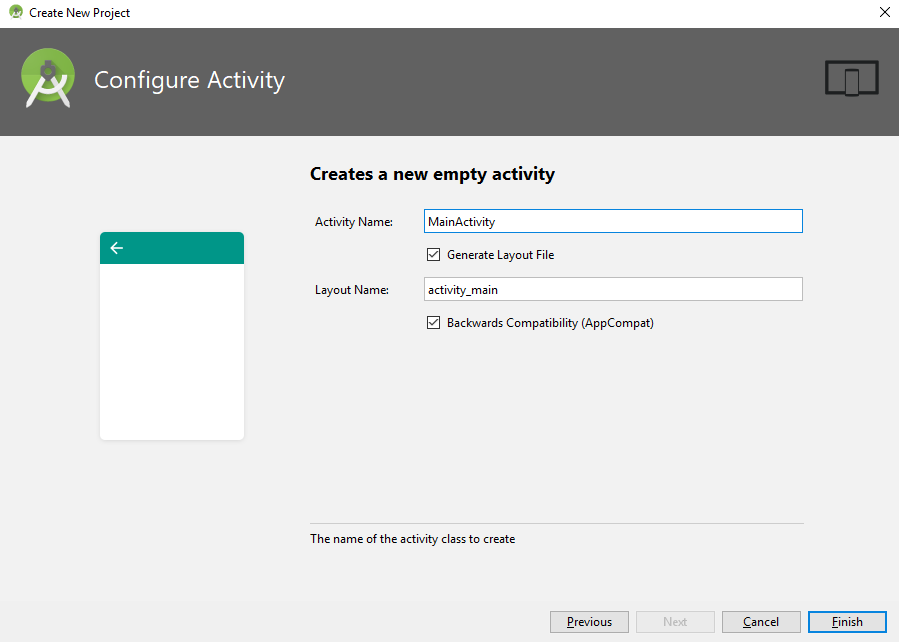
- Hiện nay bản API14 Android 4.0 (IceCreamSandwich) vẫn đứng đầu về số lượng thiết bị sử dụng chiếm tới hơn 90%) nên thường lựa chọn.

Màn hình này hiển thị cho phép chọn Activity:



Hình : Giao diện chọn loại Activity của Android Studio

Chọn EmptyActivity rồi bấm Next:



Hình : Giao khởi tạo Activity của Android studio

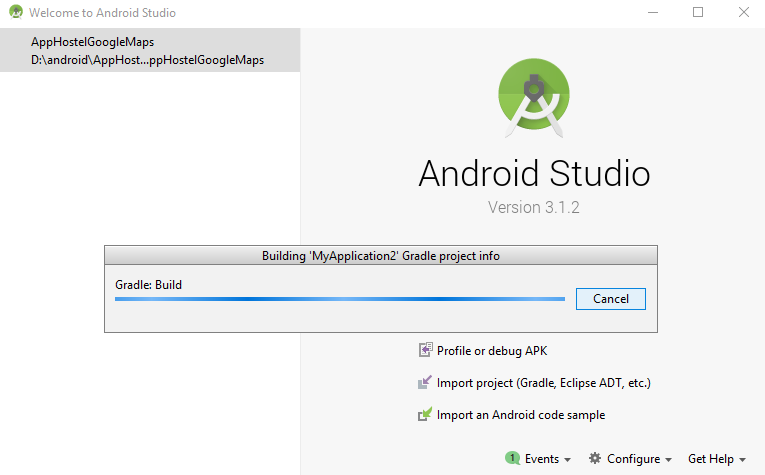
- **Activity Name**: Tên class Activity (java) để ta viết mã lệnh

- **Layout Name**: Tên file XML làm giao diện cho Activity Name.

- **Title**: Tiêu đề hiển thị khi kích hoạt Activity trên thiết bị.

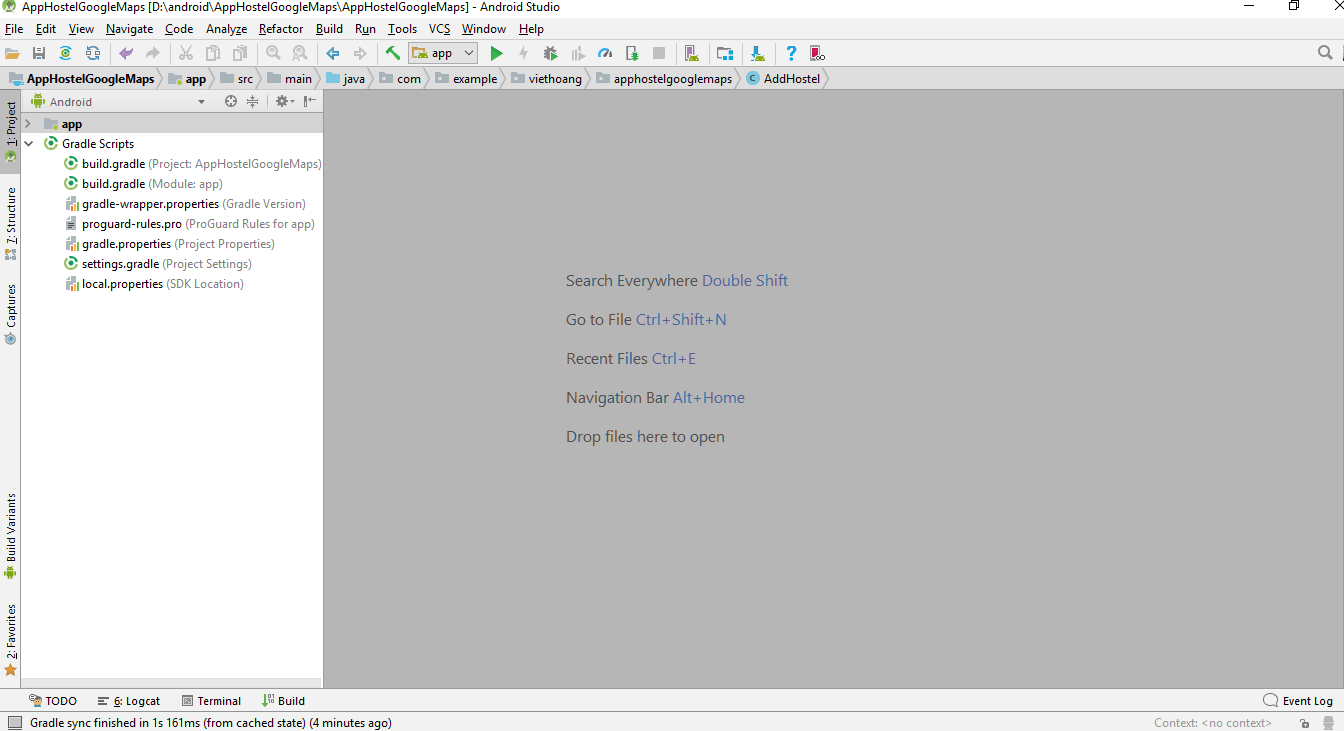
- **Menu Resource Name**: Tên file xml để tạo menu cho phần mềm

Sau khi cấu hình xong, bấm Finish, Màn hình Build Gradle project hiển thị:



Hình : Giao diện Build Gradle project của Android Studio

Khi build xong mặc định có màn hình dưới đây:



Hình : Giao diện sau khi khởi tạo xong project của Android Studio

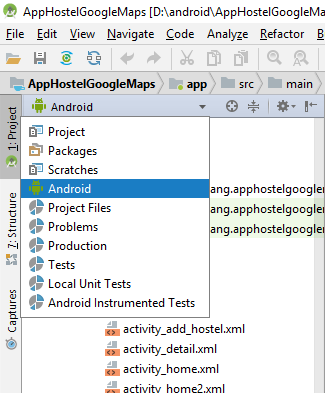
c. Màn hình làm việc của dự án android studio



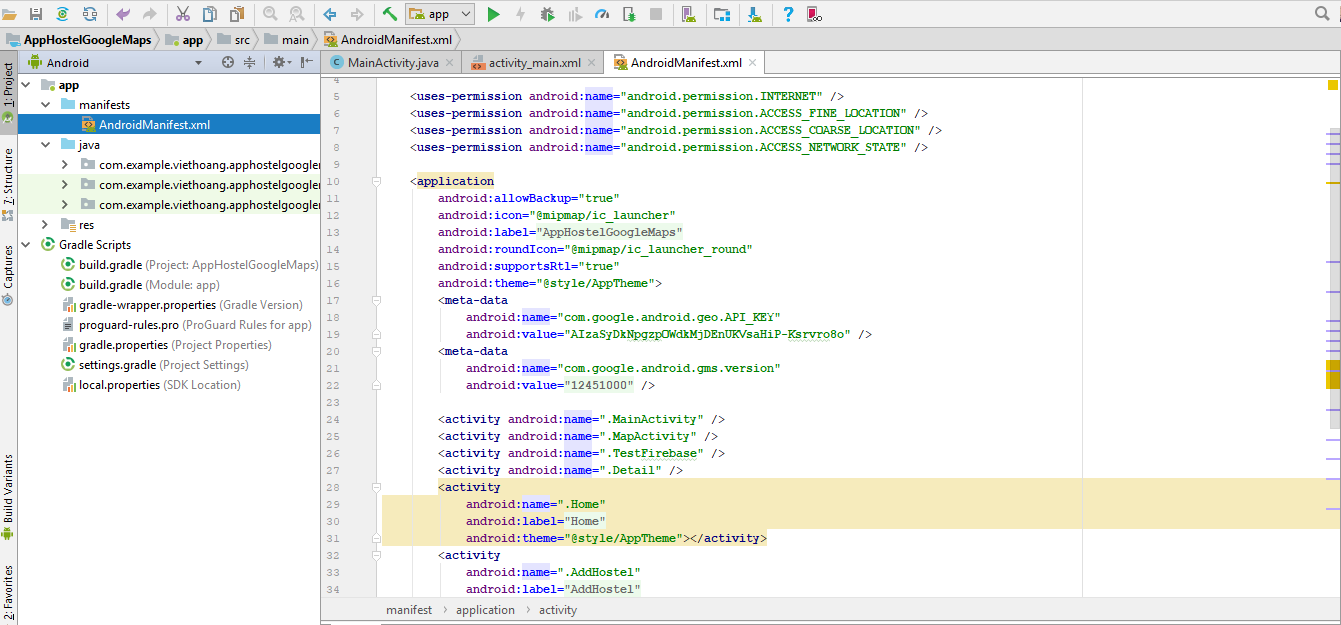
Hình : Giao diện màn hình làm việc activity\_main.xml

Ở trên tạm thời chia làm 6 vùng làm việc mà lập trình viên thƣờng tương tác.

- **Vùng 1**: Là nơi cấu trúc hệ thống thông tin của Ứng dụng, Ta có thể thay đổi cấu trúc hiển thị (thƣờng để mặc định là Android).



Hình : Giao diện màn hình ở chế độ Android



Hình : Giao diện AndroidMainfest.xml

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

Các thư mực:

• drawable: chứa các file hình ảnh và xml trong ứng dụng.

• layout: chứa các giao diện màn hình được thiết kế dưới dạng xml.

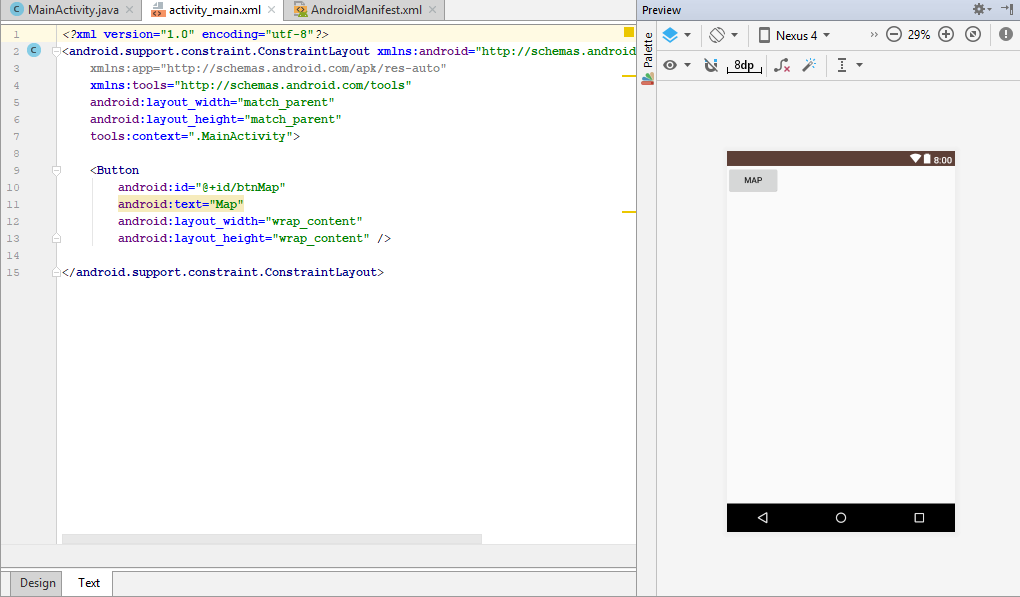
• values: chứa các file lưu giá trị màu sắc, kích thước, chuỗi,....

- **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.

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).

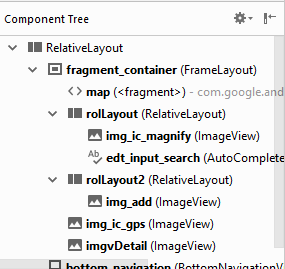
•Tab Text là tab cho phép ta thiết kế giao diện bằng viết Tag XML:



Hình : Giao diện activity\_main.xml ở dạng text

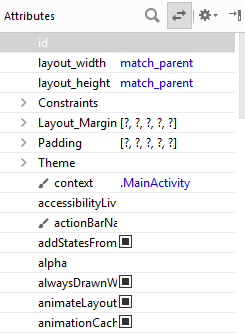
-**Vùng 3**: Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiểu chính control

- **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 lập trên giao diện (vùng 3).



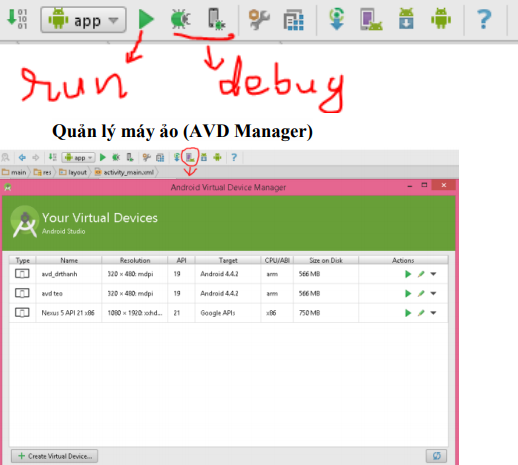
Hình : Giao diện control theo cấu trúc cây

- **Vùng 5**: Vùng này rất quan trọng, đây là vùng cho phép thiết lập trạng thái hay thuộc tính cho các Control trên giao diện.



Hình : Giao diện vùng thiết lập properties

-**Vùng 6**: Là vùng các chức năng quan trọng thường dùng trong Android Studio. Chạy ứng dụng và Debug ứng dụng:



Hình : Giao diện thanh công cụ hỗ trợ

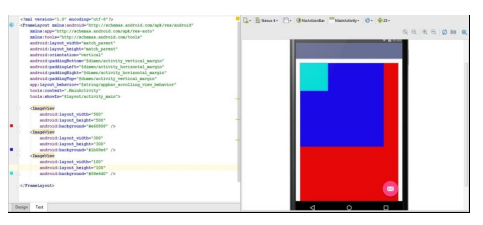
### 1.2.3. Tạo giao diện chương trình trong android studio

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". Ta có thể tham khảo đoạn XML sau để hiểu thêm về framelayout.



Hình : Giao diện FrameLayout Android studio

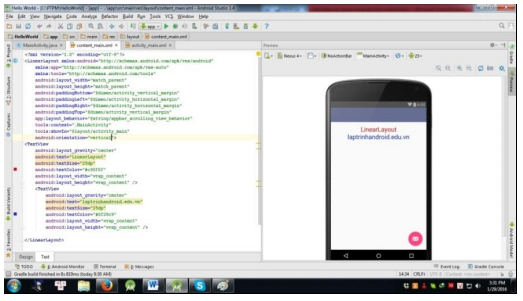
- LinearLayout.

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.

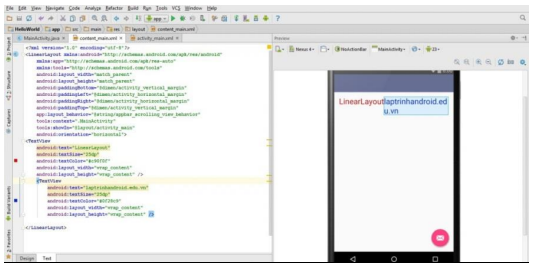
• Android:orientation="horizontal" : Xếp các control từ trái sang phải (theo cột). • Android:orientation="vertical" : Xếp các control từ trên xuống dưới (theo hàng).

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:



Hình : Giao diện LinearLayout theo hàng



Hình : Giao diện LinearLayout theo cột

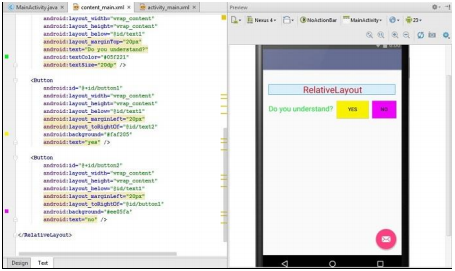
-RelativeLayout.

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.

• Các control đều có id riêng, việc đặt tên id phải rõ rang dễ hiểu.

• Các control được sắp xếp dựa vào id của các control khác.

Ta có thể tham khảo đoạn XML demo sau để hình dung dễ hơn:



Hình : Giao diện demo RelativeLayout

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.

- **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.

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.

**- 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".

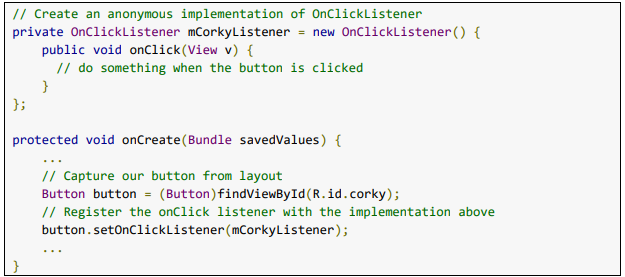
• **onFocusChange():** Thuộc View.OnFocusChangeListener. Nó được gọi khi người dùng điều hướng ra khỏi item, bằng cách sử dụng phím điều hướng.

• **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).

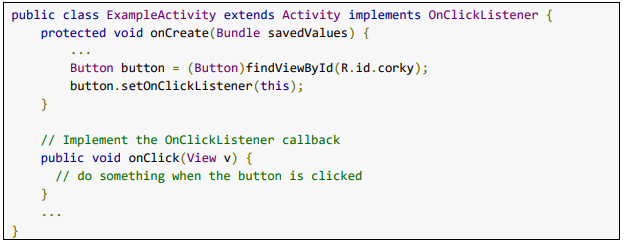
• **onCreateContextMenu():** Thuộc View.OnCreateContextMenuListener. Nó được gọi khi một menu ngữ cảnh (Context Menu) đang được xây dựng (là kết quả của một "long click"). Xem thêm thông tin về context menus trong hướng dẫn phát triển Menus.

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



Hình : Ví dụ về sự kiện click trong android

Ví dụ khác :



Hình : Ví dụ sự kiện click trong android

Chú ý rằng lời gọi onClick() trong ví dụ trên không trả về giá trị, nhưng các phương thức của bộ nghe sự kiện khác phải trả lại một biến kiểu boolean. Lý do phụ thuộc vào sự kiện này. Đây là một vài lý do:

• **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.

• **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 thương được sử dụng để xử lý sự kiện, bao gồm:

• onKeyDown(int, KeyEvent) – Được gọi khi một sự kiện nhấn phìm 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.

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).

## 1.3. Sơ lược về cơ sở dữ liệu Firebase - ****Firebase Realtime Database****

### 1.3.1. Tổng quan về Firebase

- Dữ liệu trong cơ sở dữ liệu Firebase của bạn được lưu trữ dưới dạng JSON và đồng bộ realtime đến mọi kết nối client. Khi bạn xây dựng những ứng dụng đa nền tảng như Android, IOS và JavaScrip SDKs, tất cả các client của bạn sẽ chia sẻ trên một cơ sở dữ liệu Firebase và tự động cập nhật với dữ liệu mới nhất.

- Tự đông tính toán quy mô ứng dụng của bạn

- Khi ứng dụng của bạn muốn phát triển, bạn không cần lo lắng về việc nâng cấp máy chủ, Firebase sẽ xử lý các việc tự động cho bạn. Các máy chủ của Firebase quản lý hàng triệu kết nối đồng thời và hàng tỉ lượt truy vấn mỗi tháng

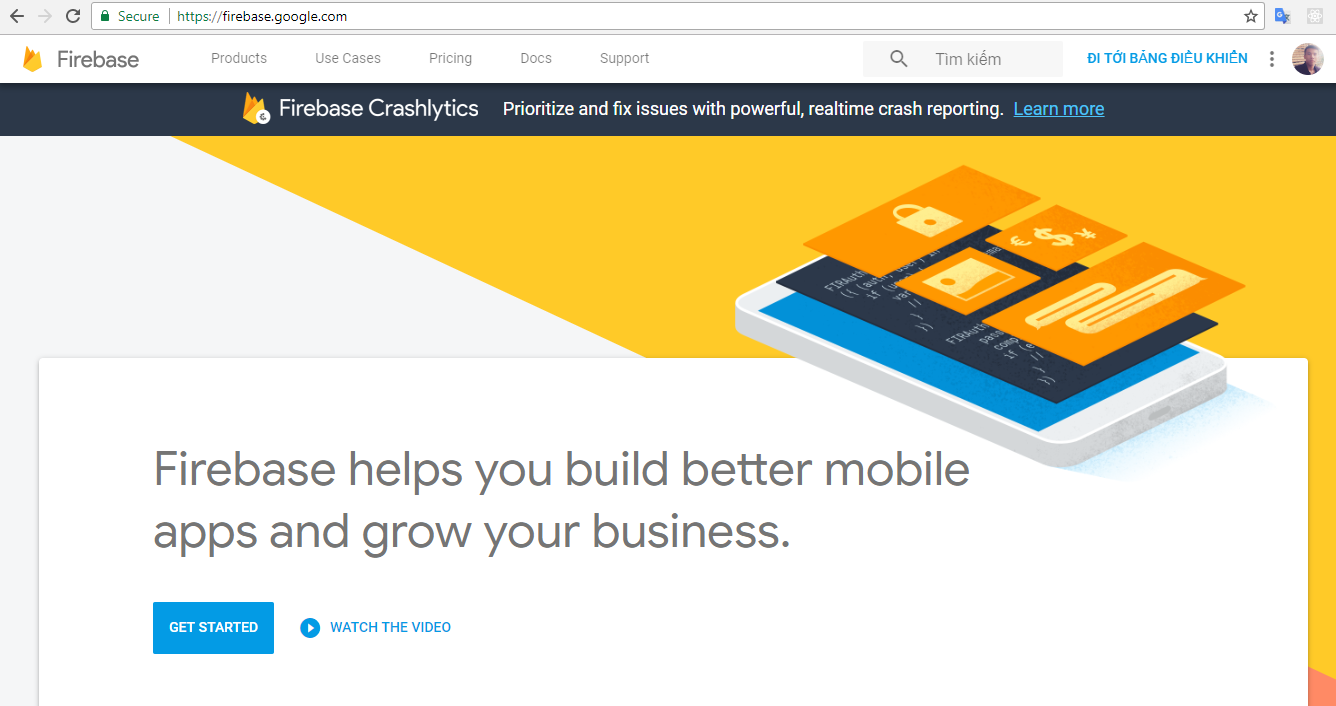
- Tất cả dữ liệu được truyền qua một kết nối ăn toàn SSL với một chứng nhận 2048-bit. Cơ sở dữ liệu truy vấn và xác nhận được điều khiển tại một cấp độ chi tiết sử dụng theo một số các quy tắc mềm dẻo ”security rules language”. Tất cả các logic bảo mật dữ liệu của bạn được tập trung ở một chỗ để dễ dàng cho việc cập nhật và kiểm thử.

- Ứng dụng Firebase của bạn se duy trì tương tác bất chấp một số các vấn đề về inter nét xảy ra. Trước khi bất kỳ dữ liệu được ghi đến server thì tất cả dữ liệu lập tức sẽ được viết vào một cơ sở dữ liệu Firebase ở local. Ngay khi có thể kết nối lại, client đó sẽ nhận bất kỳ thay đồi mà nó thiếu và đồng bộ hoá nó với trạng thái hiện tại server.

- Với Firebase, ta có thể dễ dáng xác thực người dùng từ ứng dụng của bạn trên Android, IOS và JavaScript SKDs chỉ với một đoạn mã. FireBase đã xây dựng chức năng cho việc xác thực người dùng với Email, FaceBook, Twitter, GitHub, Google và xác thực nạc danh. Các ứng dụng sử dụng chức năng xác thực của Firebase có thể giải quyết được vấn đề khi người dùng đăng nhập, nó sẽ tiết kiệm thời gian và rất nhiều các vấn đề về phần back end.

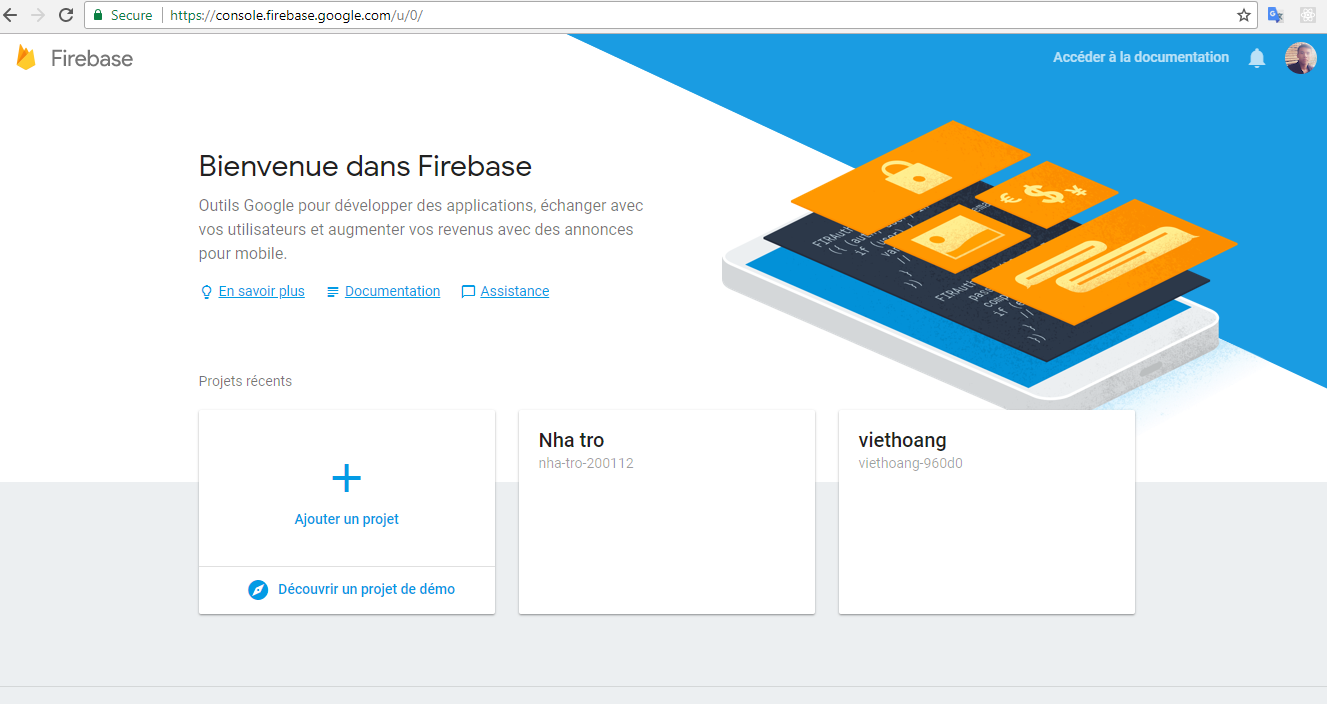
### 1.3.2. Sử dụng Firebase

- Để sử dụng Firebase ta cần truy cập vào trang chủ của Firebase: <https://firebase.google.com>



Hình : Giao diện trang chủ Firebase

- Sau khi truy cập vào trang chủ Firebase, nếu chưa có tài khoản thì ta tiến hành đăng kí và đăng nhập. Sau khi đã đăng nhập tiếp theo chọn mục đi tới bảng điều khiển:

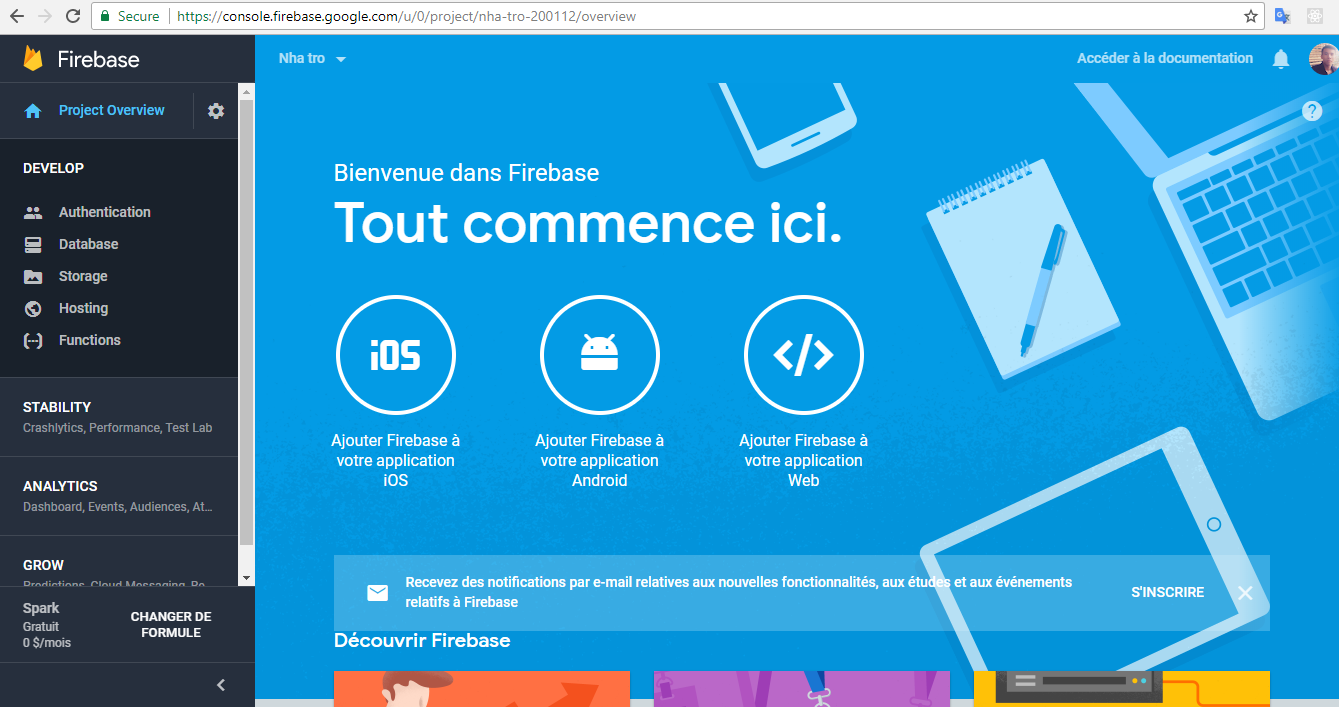


Hình : Giao diện bảng điều khiển Firebase

- Tiếp theo ta tiến hành tạo project mới bằng cách nhấn chọn vào mục “Add a project”.

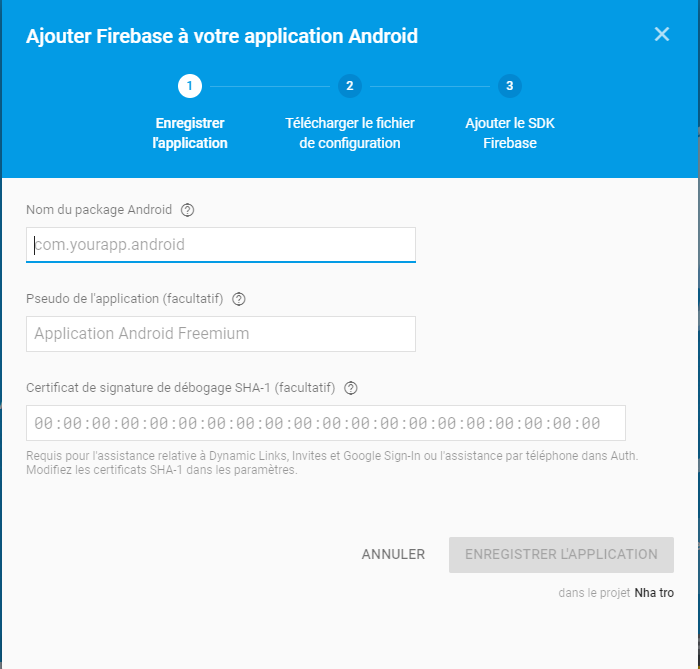
- Ở tab này ta chỉ gõ tên project và địa chỉ hiện tại nước nào là được

- Sau khi đã tạo xong một prject, ta mở project vừa tạo lên:



Hình : Giao diện bảng điều khiển Firebase

- Chọn tiếp biểu tượng Android



Hình : Giao diện tạo mới kết nối project

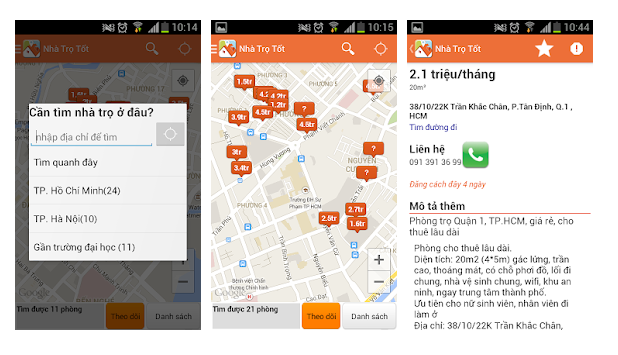
- Ở màn hình này, thì ô text đầu tiên(Package name) Chính là tên package project android

- Chọn Continue để tới bước kế tiếp, sau khi tới bước tiếp thì sẽ tiến hành download file google-service.js và để vào trong thư mục app của project android

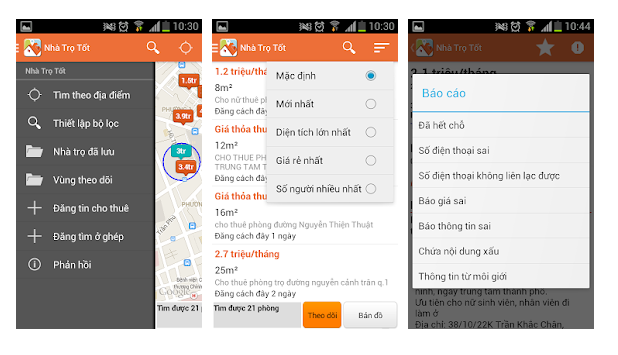
- Tiếp tục làm theo hướng dẫn để hoàn thành kết nối Firebase với project.

# CHƯƠNG 2: KHẢO SÁT THỰC TRẠNG

## 2.1. App Nhà trọ tốt

****

Hình : Giao diện app Nha trọ tốt

****

Hình : Giao diện app Nha trọ tốt

Tínhnăng:

-Xem những phòng trọ mới ngay trên bản đồ Sài Gòn, Hà Nội.

-Bộ lọc giá thấp nhât, giá cao nhất, thời gian đăng rất thuận tiện.

- Tính năng theo dõi nhà trọ mới, ứng dụng sẽ thông báo cho người dùng biết khi có nhà trọ mới trong vùng xuất hiện.

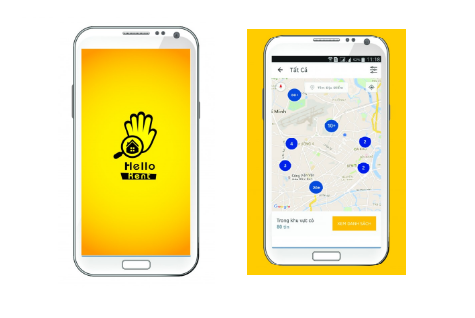
- Cập nhật 500 nhà trọ mới mỗi ngày.

- Chọn lọc và chỉ hiển thị những nhà trọ địa chỉ rõ ràng và mới đăng gần đây.

Điểm khác biệt lớn nhất của NHÀ TRỌ TỐT là tính trực quan, Các nhà trọ đang cho thuê sẽ được hiển thị đánh dấu trên bản đồ Google Maps. Khi muốn tìm nhà trọ ở khu vực nào, người dùng chỉ cần di chuyển bản đồ tới vị trí đó. NHÀ TRỌ TỐT còn cung cấp tính năng tìm kiếm địa điểm, để người dùng có thể tìm nhanh và di chuyển tới vị trí bản đồ nơi mà người dùng có nhu cầu thuê. Một điểm cộng nữa của ứng dụng NHÀ TRỌ TỐT là nguồn dữ liệu rất phong phú. Hiện tại mỗi ngày ứng dụng có khoảng 500 tin đăng mới chủ yếu phân bố ở Sài Gòn và Hà Nội. Các tin này được tổng hợp từ các website tìm nhà trọ trên internet.

## 2.2. App Hello Rent

*Hello Rent* là ứng dụng điện thoại có chức năng tìm kiếm phòng trọ hướng tới các bạn sinh viên, nhân viên văn phòng cần thuê nhà trọ tại các thành phố Hồ Chí Minh và Hà Nội



Hình : Giao diện app Hello Rent

Thông tin phòng trọ sẽ được kiểm duyệt trước cho người dùng. Ngoài ra app này còn hỗ tính năng giải trí *Hello Toon,* người dùng có thể đọc truyện Hàn Quốc đã được dịch sang tiếng việt

Tính năng nổi bật:

- Phân loại bất động sản Căn hộ Chung cư / Văn Phòng, Phòng trọ / Nhà trọ được hiển thị ngay vị trí đầu tiên.

- Tìm kiếm các địa điểm xung quanh bạn hoặc các khu vực trung tâm

- Cung cấp các lựa chọn tìm kiếm theo giá thuê, diện tích…

# CHƯƠNG 3: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

## 3.1 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 về tìm trọ cho sinh viên và công nhân, hỗ trợ cập nhật thông tin điện thoại trực tuyến qua mạng internet mọi nơi mọi lúc thông qua smart phone chạy hệ điều hành Android. Các thao tác cần thực hiện trên cơ sơ dữ liệu internet :

- Hiển thị bản đồ cùng với các địa điểm có nhà trọ trên bản đồ: Lấy dữ liệu từ cơ sở dữ liệu Firebase và hiển trị trên Google Maps

- Xem chi tiết một nhà trọ: truy vấn với từ cơ sở dữ liệu theo điều kiện

- Xem danh sách nhà trọ ở dạng danh sách: Lấy dữ liệu từ cơ sở dữ liệu Firebase và hiển thị ở dạng danh sách

- Thêm mới nhà trọ: Thêm dữ liệu mới vào cơ sở dữ liệu Firebase

## 3.2. Người dùng

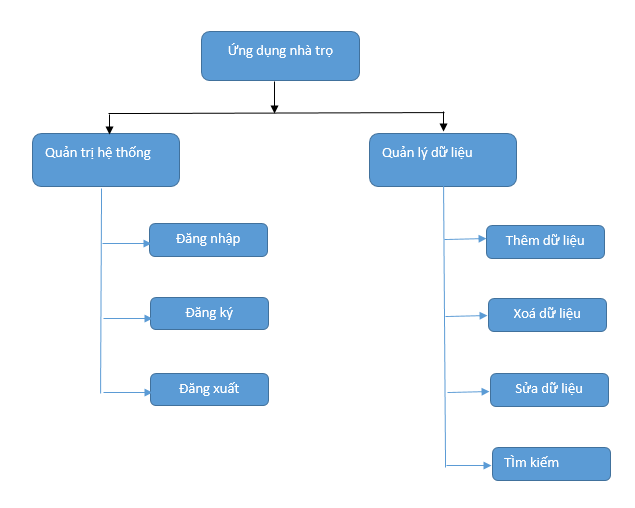
Ứng dụng nhà trọ trước mắt sẽ chỉ hướng tới mọi người dùng, sẽ không có Admin hay User đăng nhập thay vào đó sẽ dùng chung, có nghĩa là mọi người đều là admin

## 3.3 Biểu đồ

Là một sơ đồ hình học dùng để mô tả sự phân rã có thứ bậc các chức năng của hệ thống từ tổng quan đến chi tiết. Mỗi chức năng được ghi trong một khung và nếu cần sẽ được phân thành những chức năng con, số mức phân ra phụ thuộc vào kích cỡ và độ phức tạp của hệ thống

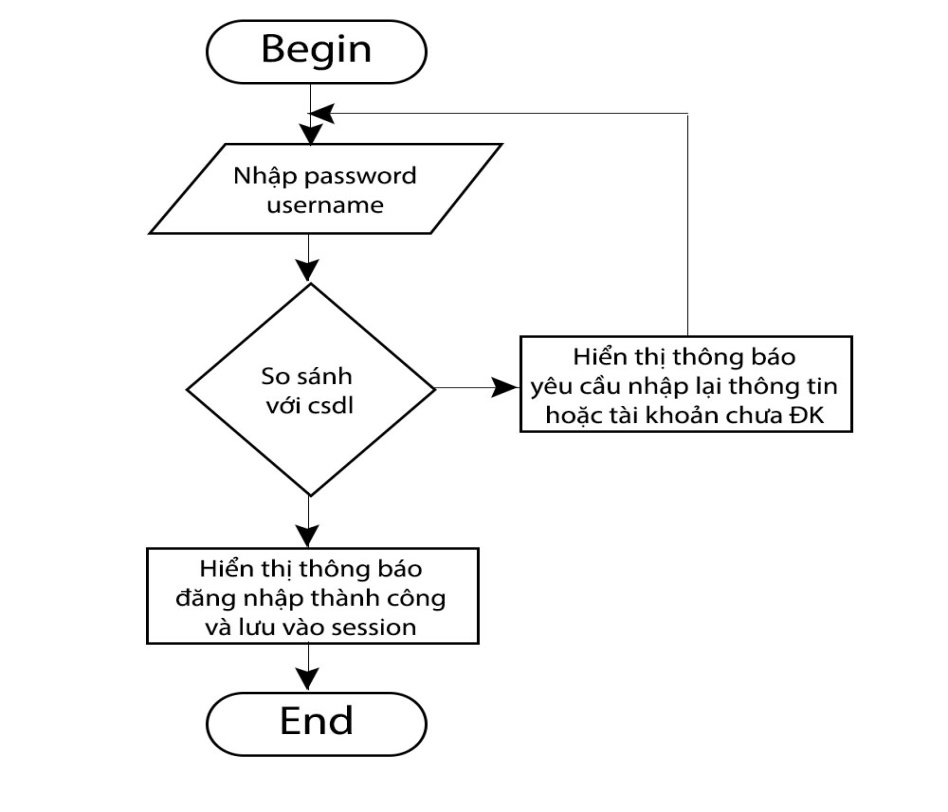
### 3.3.1 Biểu đồ phân rã chức năng

Những chức năng và hoạt động trong hệ thống ứng dụng như: Xem, tìm kiếm, thêm mới…



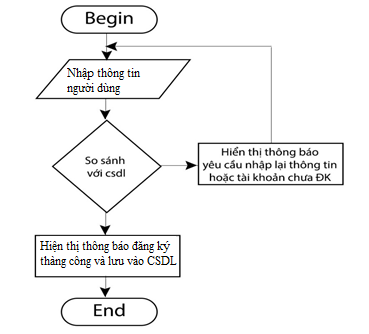
Hình : Sơ đồ hoạt động

**Đăng nhập**



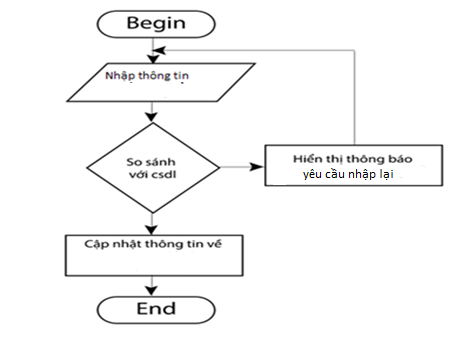
Hình .Biểu đồ hoạt động đăng nhập vào hệ thống

**Đăng ký**



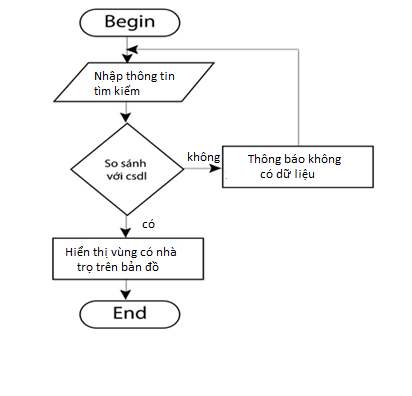
Hình . Biểu đồ hoạt động đăng ký

**Thêm mới nhà trọ**



Hình .Biểu đồ chức năng thêm nhà trọ mới

**Tìm kiếm nhà trọ**

****

Hình : Biểu đồ chức năng tìm kiếm

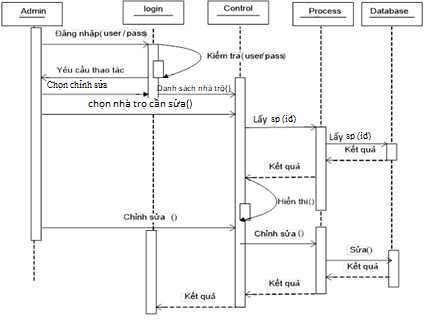
## 3.4 Đặc tả Use-case

- **Chức năng thêm nhà trọ mới**



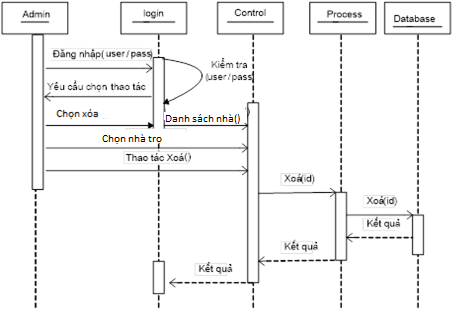
Hình .Biểu đồ tuần tự chức năng thêm nhà trọ mới

**-Chức năng sửa thông tin nhà trọ:**

****

Hình .Biểu đồ tuần tự chức năng sửa thông tin nhà trọ

-**Chức năng xóa sản phẩm cần thiết**



Hình .Biểu đồ tuần tự chức năng xóa nhà trọ

- **Chức năng tìm kiếm : tương tự như các chức năng khác.**

## 3.5 Bảng users

Bảng users sẽ lưu thông tin của user khi đăng ký vào ứng dụng, từ đó ta có thể quản lý user của hệ thống.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Kích thước | Khóa chính | Mô tả |
| 1 | ID | Int |  | ID |  |
| 2 | Username | Varchar | 250 |  | Tên đăng nhập |
|  | fullname | Varchar | 250 |  | Họ tên |
| 3 | Gender | Varchar | 250 |  | Giới tính |
| 4 | Birthday | Varchar | 250 |  | Ngày sinh |
| 5 | City | Varchar | 250 |  | Thành phố |
| 6 | Phone | Varchar | 250 |  | Số điện thoại |
| 7 | Created\_date | Date |  |  | Tự động lấy ngày hiện tại |

## 

3.6 Bảng nhà trọ

Khi thêm mới nhà trọ thì thông tin user cùng thông tin nhà trọ sẽ lưu ở bảng này, bảng này có thêm trường id\_user để biết được user nào đăng tin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Kích thước | Khóa chính | Mô tả |
| 1 | ID | Int |  | ID |  |
| 2 | Id\_user | Int |  |  | Id người dùng |
| 3 | Username | Varchar | 250 |  | Tên đăng nhập |
| 4 | Image | Varchar | 250 |  | Hình ảnh |
| 5 | City | Varchar | 250 |  | Thành phố |
| 6 | Country | Varchar | 250 |  | Quận/Huyện |
| 7 | Ward | Varchar | 250 |  | Phường/Xã |
| 8 | Street | Varchar | 250 |  | Đường |
| 9 | Address | Varchar | 250 |  | Địa chỉ |
| 10 | User\_motel | Varchar | 250 |  | Tên chủ trọ |
| 11 | Phone\_user\_motel | Varchar | 250 |  | Số điện thoại liên hệ |
| 12 | Acreage | Varchar | 250 |  | Diện tích |
| 13 | Price | float |  |  | Giá tiền |
| 14 | Longlat | Varchar | 250 |  | Tọa độ bản đồ |
| 15 | description | Varchar | 250 |  | Mô tả |
| 16 | Created\_date | Date |  |  | Tự động lấy ngày hiện tại |

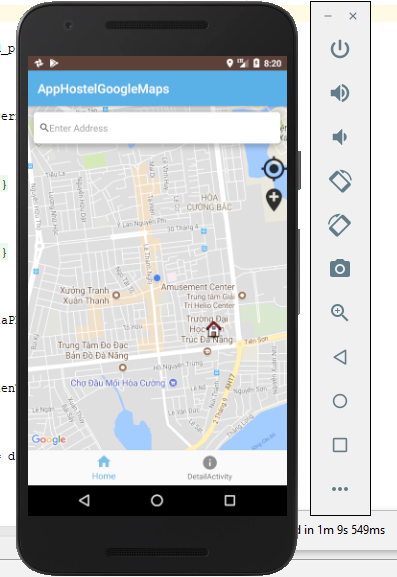
# CHƯƠNG 4: XÂY DỰNG VÀ PHÁT TRIỂN ỨNG DỤNG DEMO

Ở chương 3, đã phân tích và xây dựng cơ sở dữ liệu các chức năng cần có. Nên ở chương 4, tiến hành xây dựng chức năng ứng dụng như sau:

## 4.1. Xây dựng giao diện Google Maps

Giao diện chính của ứng dụng là bản đồ Google Máp, ở đây sẽ hiển thị tất cả các nhà trọ, thanh công cụ tìm kiếm, nút hiển thị vị trí của tôi, nút chi tiết khi click vào nhà trọ trên map.

Giao diện chính của ứng dụng:

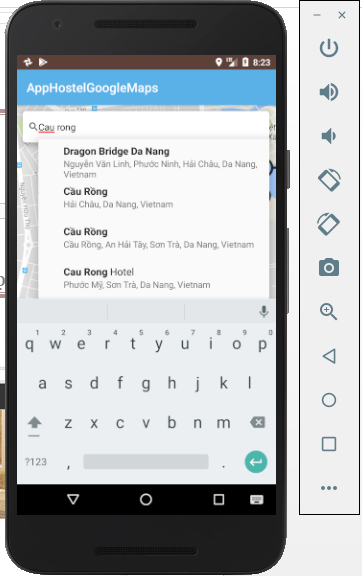


Hình : Giao diện chính của ứng dụng

Mô tả chi tiết:

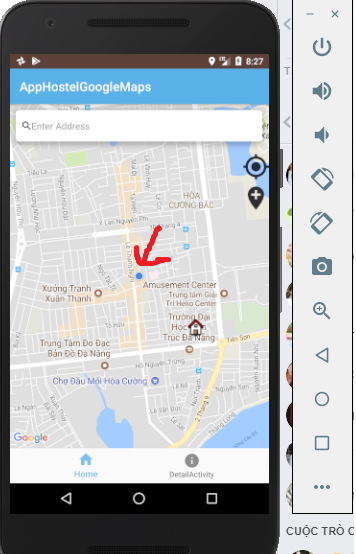
- Google Map hiển thị vị trí các nhà trọ có trong cơ sở dữ liệu, có thể phóng to hoặc thu nhỏ theo ý muốn để xem vị trí nhà trọ.

- Thanh tìm kiếm nhà trọ:



Hình : Thanh công cụ tìm kiếm

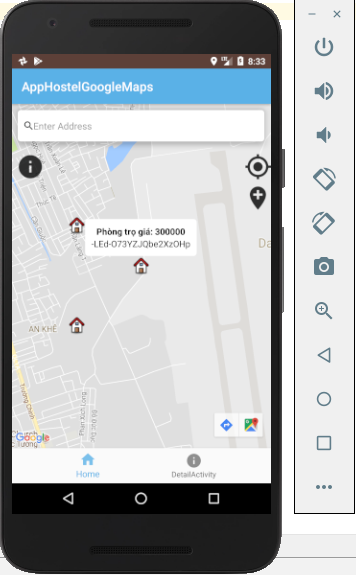
Thanh công cụ tìm kiếm sẽ hỗ trợ tìm kiếm nhà, gợi ý địa chỉ tìm kiếm cho người dùng, giúp người dùng tiếp kiệm thời gian tìm kiếm.



Hình : Hiển thị vị trí của tôi

- Nút Location (vị trí của tôi), khi click thì sẽ hiển thị vị trí của ta ở trên bản đồ.

- Nút thêm mới nhà trọ nằm ở phía dưới nút Location, khi click nút này thì sẽ chuyển qua giao diện thêm nhà trọ.

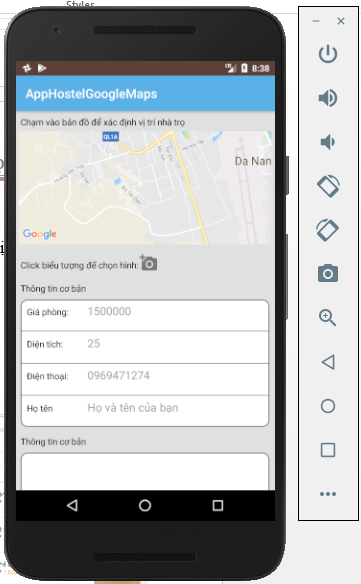


Hình : Giao diện hiển thị các nút công cụ

- Nút chi tiết nhà trọ: khi click vào một nhà trọ trên bản đồ thì nút này sẽ hiện lên, cùng với đó là công cụ chỉ đường của Google Map.

## 4.2. Xây dựng giao diện thêm mới nhà trọ

Khi nhấn vào nút add nhà trọ trên giao diện thì sẽ hiển thị giao diện thêm mới nhà trọ



Hình : Giao diện thêm mới nhà trọ

Ở giao diện này gồm có:

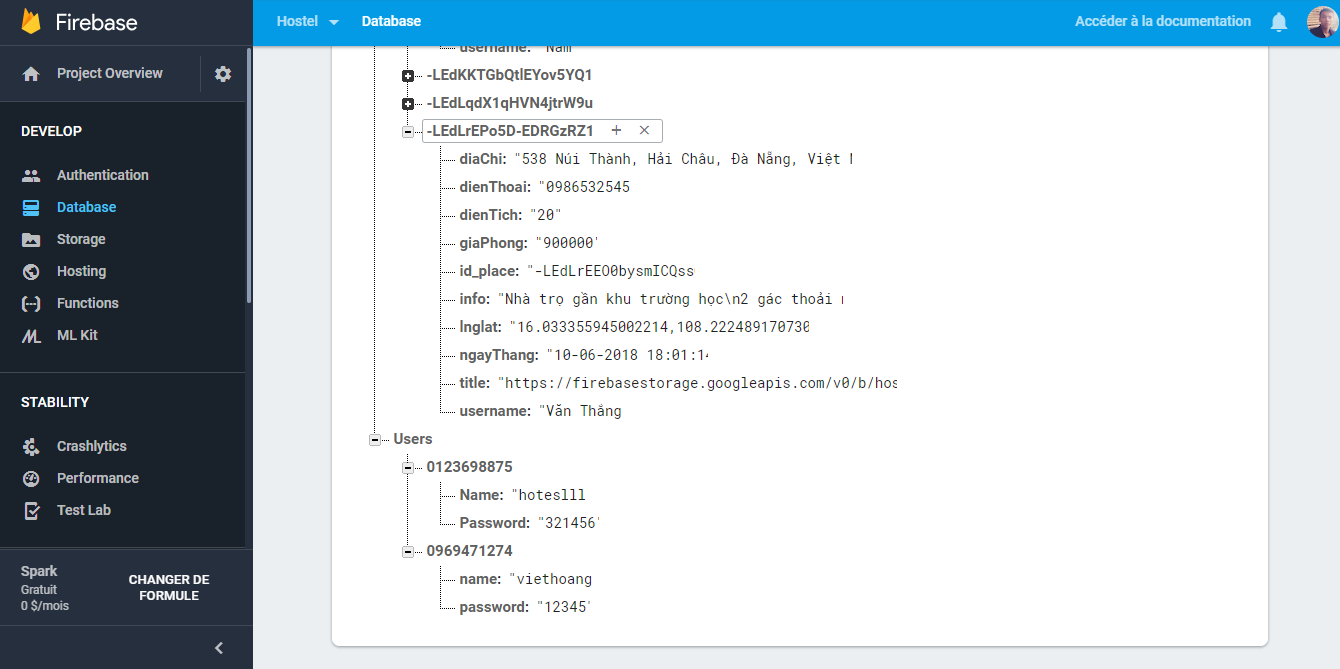
- Bản đồ thu nhỏ giúp người dùng chọn vị trí của nhà trọ muốn đăng mới

- Thêm hình ảnh cho nhà trọ

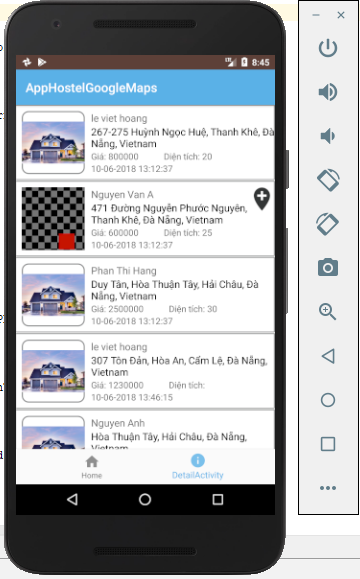
- Các thông tin như giá phòng, diện tích, điện thoại, họ tên

- Thông tin mô tả thêm

Cơ sở dữ liệu sau khi thêm mới nhà trọ:



## 4.3. Giao diện danh sách nhà trọ



Hình : Giao diện danh sách nhà trọ

- Hiển thị nhà trọ theo dạng danh sách, gồm các thông tin cần thiết như: tên người đăng, địa chỉ, giá phòng, diện tích, ngày đăng…

## 4.4. Xây dựng giao diện chi tiết nhà trọ



Hình : Giao diện chi tiết nhà trọ

Giao diện chi tiết nhà trọ sẽ hiển thị thông tin chi tiết của một nhà trọ có trong cơ sở dữ liệu, bao gồm hình ảnh, giá phòng, diện tích, điện thoại, địa chỉ, mô tả thêm nếu có, vị trí trên bản đồ.

# KẾT LUẬN

## 1. Ưu điểm

Đề tài “ **Lập trình ứng dụng Nhà Trọ**” cũng xuất phát từ thực tế ngày nay nhằm tạo một nền tảng cơ sở ban đầu để có thể hỗ trợ thêm cho lập trình viên muốn thiết kế một ứng dụng cho công ty hay cá nhân.

Mặc dù đã có nhiều cố gắng, tìm hiểu các kiến thức đã học, kết hợp tra cứu các tài liệu chuyên nghành nhưng do còn hạn chế về thời gian, khả năng và kinh nghiệm nên không tránh khỏi những thiếu sót nhất định nên đề tài đã hoàn thành ở mức độ sau:

- Tìm hiểu ngôn ngữ lập trình Java(android), cơ sở dữ liệu Firebase, Android API, Google Maps API, Webservice.

- Áp dụng xây dựng ứng dụng thực nghiệm trang **Lập trình ứng dụng nhà trọ** với nhiều chức năng khác nhau

- Áp dụng được Google Maps API của Android

## 2. Nhược điểm:

Hệ thống chưa được tối ưu.

Giao diện đồ họa còn đơn điệu

Ứng dụng quy mô vẫn còn nhỏ, ứng dụng, chức năng chưa đầy đủ

## 3. Các vấn đề đã thực hiện

- Sau thời gian thực hiện đề tài, ứng dụng đã hoàn thành và đạt được một số kết quả sau:

+ Xây dựng thành công trang giao diện chính tích hợp Google Maps API, hiển thị nhà trọ trên bản đồ

+ Chức năng tìm kiếm thông minh của Google Map

+ Tìm hiểu thêm về các công cụ để xây dựng ứng dụng android như: Ngôn ngữ Java, Firebase, Google Map API

+ Xây dựng thành công giao diện thêm mới nhà trọ

+ Phân tích và thiết kế hệ thống để xây dựng được ứng dụng nhà trọ

+ Giao diện của chương trình thân thiện, dễ sử dụng.

## 4. Hướng phát triển

Tìm hiểu sâu hơn về ngôn ngữ Java , Android để có thể đáp ứng nhiều hơn nữa nhu cầu của người sử dụng, phát triển và tối ưu hóa hệ thống.

Kết hợp ngôn ngữ Java với cơ sở dữ liệu Firebase và các cơ sở dữ liệu khác.

Phát triển giao diện đồ họa bắt mắt hơn.

Phát triển quy mô ứng dụng, bổ sung thêm các chức năng khác.

Để hoàn thành đề tài này, một lần nữa em xin chân thành cảm ơn Thầy Mai Lam là người đã hướng dẫn, giúp đỡ em trong suốt thời gian qua. Em xin chân thành cảm ơn !

# TÀI LIỆU THAM KHẢO

[1] <https://firebase.google.com/docs>

[2] <https://developer.android.com>

[3] <https://developers.google.com>

# LỜI CẢM ƠN

Nhiệm vụ của một sinh viên là phải hiểu rõ và vận dụng những kiến thức mình đã học được ở nhà trường vào thực tế. Đây là một bước cần thiết để một người sinh viên có thể rèn luyện những kỹ năng để trở thành một kỹ sư. Với “Đồ án tốt nghiệp” em xin chọn đề tài: “Lập trình ứng dụng Nhà Trọ”. Dưới sự hướng dẫn của giáo viên Mai Lam , em đã hoàn thành ứng dụng. Do thời gian và trình độ còn hạn chế nên đề tài của em vẫn còn nhiều thiếu sót. Rất mong các thầy cô chỉ bảo trong buổi bảo vệ để em rút ra được những kinh nghiệm cho quá trình học tập và làm việc sau này. Qua đây, em xin chân thành cảm ơn các thầy cô giáo, đặc biệt là Thầy Mai Lam đã trực tiếp hướng dẫn và trang bị cho em những kiến thức kỹ năng chuyên môn bổ ích và cũng là người đã trực tiếp và tận tình hướng dẫn em hoàn thành đồ án tốt nghiệp. Em cũng xin gửi lời cảm ơn đến các thầy giáo của bộ môn, các thầy cô đã tận tình giúp đỡ để em có điều kiện hoàn thành được đề tài.

Sau cùng, em xin kính chúc quý Thầy Cô trong Trường Cao Đẳng Công Nghệ Thông Tin thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Trân trọng.

Đà Nẵng, ngày tháng năm 2018

Sinh viên

Lê Việt Hoàng

NHẬN XÉT CỦA GIÁO VIÊN

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Đà Nẵng, ngày tháng năm 2018

Giảng viên hướng dẫn

(ký và ghi rõ)