|  |  |
| --- | --- |
| logomobile.png | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN LẬP TRÌNH DI ĐỘNG ANDROID**

**ĐỀTÀI:**

**APP FastFood**

Ngành : **CÔNG NGHỆ THÔNG TIN**

Giảng viên hướng dẫn : Võ Tấn Dũng

Sinh viên thực hiện: MSSV: Lớp:

Nguyễn Ngọc Nam 1711060833 17DTHA1

Nguyễn Nhật Kha 1711060452 17DTHA1

Lâm Quốc Diệm 1711060566 17DTHA1

Tp.HCM 2021

**MỤC LỤC**

[LỜI CAM ĐOAN 3](#_Toc61361798)

[Danh mục các hình ảnh 4](#_Toc61361799)

[LỜI MỞ ĐẦU 8](#_Toc61361800)

[THÔNG TIN VỀ ĐỀ TÀI 9](#_Toc61361801)

[1. Mô tả đề tài: 9](#_Toc61361802)

[2. Lý do thực hiện đề tài: 9](#_Toc61361803)

[3. Mục tiêu đề tài: 9](#_Toc61361804)

[4. Nội dung thực hiện: 9](#_Toc61361805)

[5. Dự kiến kết quả đạt được: 9](#_Toc61361806)

**[CHƯƠNG I: TỔNG QUAN](#_Toc61361807)** [10](#_Toc61361807)

[1.1 GIỚI THIỆU 10](#_Toc61361808)

[1.1.1. Tên Đề Tài: 10](#_Toc61361809)

[1.1.2. Giới Thiệu Về Ứng Dụng Web: 10](#_Toc61361810)

[1.2 NHIỆM VỤ CỦA ĐỒ ÁN: 10](#_Toc61361811)

[1.2.1 Mục Đích Của Việc Xây Dựng APP MOBILE 10](#_Toc61361812)

[1.2.2 Mục Tiêu Của Việc Xây Dựng APP MOBILE 10](#_Toc61361813)

[1.2.3 Đối Tượng Và Chức Năng Hướng Tới 10](#_Toc61361814)

[1.2.4 Phạm Vi Nghiên Cứu 10](#_Toc61361815)

**[CHƯƠNG II: CƠ SỞ LÝ THUYẾT](#_Toc61361816)** [12](#_Toc61361816)

[2.1: Gioi thiệu hệ điều hành Android Studio 12](#_Toc61361817)

[2.2: Lịch sử phát triển 12](#_Toc61361818)

[2.3:Giao diện Android 14](#_Toc61361819)

[2.4 Ứng dụng Android 15](#_Toc61361820)

[2.5 Quản lý bộ nhớ Android 15](#_Toc61361821)

[2.6: Nhân Linux 16](#_Toc61361822)

[2.7:Lịch nâng cấp 17](#_Toc61361823)

[2.8:Cộng đồng mã nguồn mở: 18](#_Toc61361824)

[2.9 Bảo mật và tính riêng tư của Android 19](#_Toc61361825)

**[CHƯƠNG 3: CÀI ĐẶT VÀ CẤU TRÚC ANDROID](#_Toc61361826)** [21](#_Toc61361826)

**[3.1: Cài đặt lập trình Android Studio 21](#_Toc61361827)**

[3.1.1. Cài đặt JAVA JDK 21](#_Toc61361828)

[3.1.2 Cài đặt Android Studio 22](#_Toc61361829)

**[3.2 Cấu trúc chuẩn trên Android Studio](#_Toc61361830)** [26](#_Toc61361830)

[3.2.1 Thư mục java 27](#_Toc61361831)

[3.2.2 Thư mục res (Resources) 28](#_Toc61361832)

[3.2.3 Thư mục drawable (res/drawable) 28](#_Toc61361833)

[3.2.4 Thư mục layout (res/layout) 28](#_Toc61361834)

[3.2.5 Thư mục Mipmap (res / mipmap) 29](#_Toc61361835)

[3.2.6 Thư mục values (res/values) 29](#_Toc61361836)

[3.2.7 Thư mục Manifests 30](#_Toc61361837)

[3.2.8 Gradle Scripts 31](#_Toc61361838)

[3.2.9 File Android Layout (Activity\_main.xml) 31](#_Toc61361839)

[3.2.10 File Android Main Activity (MainActivity.java) 32](#_Toc61361840)

[3.2.11 File Android Manifest (AndroidManifest.xml) 32](#_Toc61361841)

**[CHƯƠNG 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 34](#_Toc61361842)**

[4.1. Kết quả triển khai một số chức năng chính 34](#_Toc61361843)

[4.2. Cơ sở dữ liệu. 38](#_Toc61361844)

[4.3 Bảng Mô Tả Chức Năng 38](#_Toc61361845)

**[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN](#_Toc61361846)** [40](#_Toc61361846)

**[1:Ưu điểm đề tài](#_Toc61361847)** [40](#_Toc61361847)

**[2:Hạn chế còn tồn tại](#_Toc61361848)** [40](#_Toc61361848)

**[3:Kết Luận](#_Toc61361849)** [40](#_Toc61361849)

**[4:Hướng phát triển](#_Toc61361850)** [40](#_Toc61361850)

[TÀI LIỆU THAM KHẢO 41](#_Toc61361851)

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

LỜI MỞ ĐẦU

Với thời đại 4.0 hiện nay thời buổi của công nghệ phát triển. Chúng ta đã có rất nhiều app booking phòng khách sạn,app mua sắm...nhưng không thể thiếu app booking thức ăn FastFood. Vì nó giúp ta có thể đặt những món ăn nhanh tại những địa điểm gần nhất khi có việc bận không thể ra ngoài. Cùng với sự phát triển của công nghệ hiện nay thì bạn chỉ việc sở hữu 1 chiếc điện thoại là có thể tải app và đặt ngay những món ăn yêu thích một cách dễ dàng.

THÔNG TIN VỀ ĐỀ TÀI

1. Mô tả đề tài:
   1. Ý tưởng:

* Tạo một app FastFood
  1. Chức năng:
* Khách hàng có thể đăng kí đăng nhập
* Khách hàng có thể tìm kiếm những món ăn gần nhất.

1. Lý do thực hiện đề tài:

Với cuộc sống phát triển như ngày nay, việc đặt thức ăn nhanh của giới trẻ hiện nay ngày càng trở nên phổ biến hơn,việc có một app booking cho riêng mình sẽ cho việc học việc làm của mình thuận tiện . Do đó tạo ra một app FastFood sẽ giúp cho việc tìm kiếm thức ăn dễ dàng hơn.

1. Mục tiêu đề tài:

* Tạo ra một app mobile là cách tối ưu nhất để người quản trị của cửa hàng có thể quản lý được những sản phẩm, những đơn hàng,… Và người dùng có thể tìm được thông tin một cách dễ dàng hiệu quả hơn.
* Thiết kế một giao diện đơn giản, dễ sử dụng cho bất kỳ người nào.
* Lưu trữ thông tin nhanh chóng, độ chính xác cao.

1. Nội dung thực hiện:

Gồm 5 phần: Phân tích yêu cầu, thiết kế Cơ sở dữ liệu, thiết kế giao diện, kiểm thử, vận hành và sử dụng.

1. Dự kiến kết quả đạt được:

Hoàn thành app mobile đáp ứng được những mục tiêu đã đặt ra, có thể phát triển và cải tiến sau này.

**CHƯƠNG I: TỔNG QUAN**

1.1 GIỚI THIỆU

1.1.1. Tên Đề Tài:

Xây dựng app FastFood

1.1.2. Giới Thiệu Về Ứng Dụng Web:

Mobile App theo định nghĩa là những ứng dụng di động mà người dùng có thể truy cập vào các nội dung mà nhà phát triển lập trình trên các thiết bị di động. Có hai loại ứng dụng phổ biến mà bạn thường sử dụng đó là Native mobile app và Hybrid mobile app

1.2 NHIỆM VỤ CỦA ĐỒ ÁN:

1.2.1 Mục Đích Của Việc Xây Dựng APP MOBILE

Xây dựng một app FastFood với giao diện dễ nhìn, dễ sử dụng. cho người dùng có thể truy cập vào để tìm kiếm những món ăn cần tìm.

1.2.2 Mục Tiêu Của Việc Xây Dựng APP MOBILE

* Cung cấp cho người dùng những món ăn
* Nắm vững kiến thức về lập trình app mobile
* Phát triển app mobile với quy mô lớn để đáp ứng nhu cầu của người đùng

1.2.3 Đối Tượng Và Chức Năng Hướng Tới

App được xây dựng để phục vụ đối tượng chính là người dùng ( user). Đối với người dùng giúp cho dễ dàng sử dụng website và tìm kiếm được những loại xe mới và tốt nhất chính vì vậy app sẽ có những chức năng sau:

+ Tìm kiếm những món ăn

+ Địa điểm gần nhất

1.2.4 Phạm Vi Nghiên Cứu

* Mục tiêu:
  + Tạo dựng được một app mobile dễ dàng sử dụng.
* Mô tả:
  + App cung cấp đầy đủ cũng như những bản dịch một cách chính xác nhất
* Lợi ích mang lại:
  + Tạo sự tiện lợi, nhanh chóng cho người dùng.
  + Tạo thương hiệu, tiếng vang và lan rộng cho mọi người trong xã hội.
* Các bước thực hiện để hoàn thành dự án:
  + Lập kế hoạch phát triển.
  + Phân tích hệ thống.
  + Thiết kế.
  + Kiểm tra.
  + Biên soạn tài liệu và hướng dẫn sử dụng.

**CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

**2.1: 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 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 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 động và các lập trình viên nhiệt huyết đượcđiều 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 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 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 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.

**2.2: 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. Bên trong của hệ điều hành Android mới là một con robot 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.

**2.3: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 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, tưởng tượng như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hính chính Android thường gồm nhiều biểu tựợng (icon) và tiện ích (widget); biểu tươgg 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 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 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 theoe 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. **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. **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 ngừng 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.

**2.6: 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 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.

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 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 đưaa 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 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 đựợc 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.

**2.7: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 6.0 Marshmallow.

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 đựợc 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 phong 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. Vào năm 2011, Google đã hợp tác cùng một số hãng công nghiệp và ra mắt "Liên minh nâng cấp Android", với lời hứa sẽ nâng cấp 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.

**2.8: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 tốn 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 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 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 hoặc ngừng hỗ trợ cho chính các thiết bị của họ.

* 1. **Bảo mật và tính riêng tư của Android**

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, những 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 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 thâu tóm Open Kernel Labs để xây dựng lại Jellybean trên nền bộ vi kiểm soát dành cho dự án "Knox".

**CHƯƠNG 3: CÀI ĐẶT VÀ CẤU TRÚC ANDROID**

## 3.1: Cài đặt lập trình Android Studio

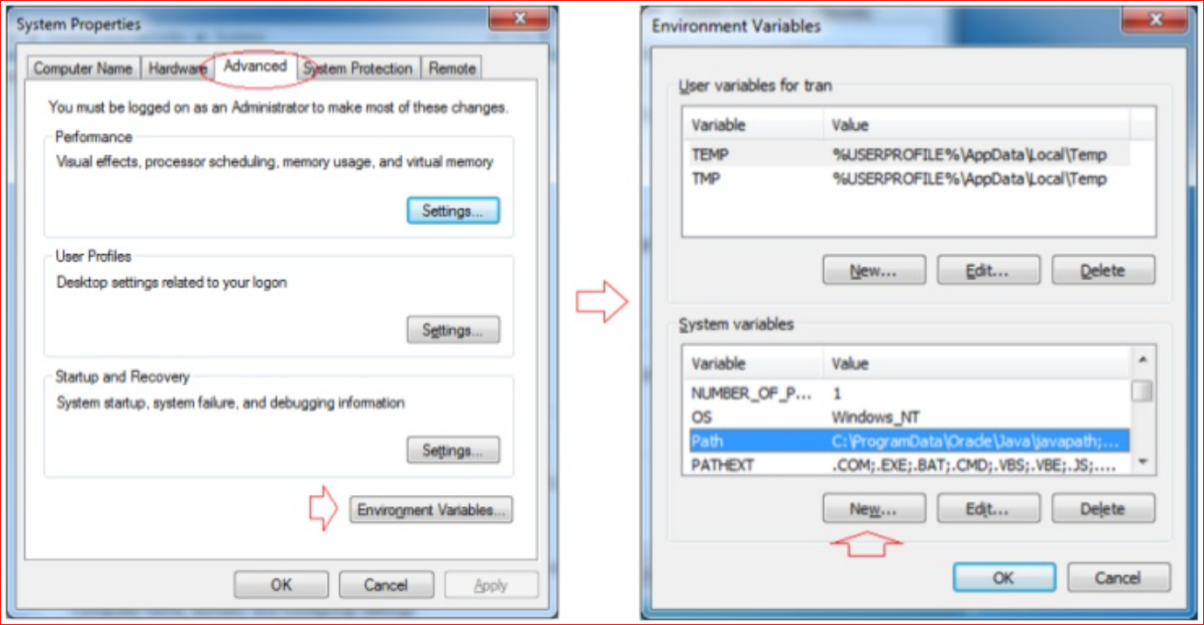
### 3.1.1. Cài đặt JAVA JDK

* **Bước1:** Tải file cài đặt tại đường link :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Lưu ý: Chọn phiên bản tương ứng với hệ diều hành đúng với máy đang sử dụng.

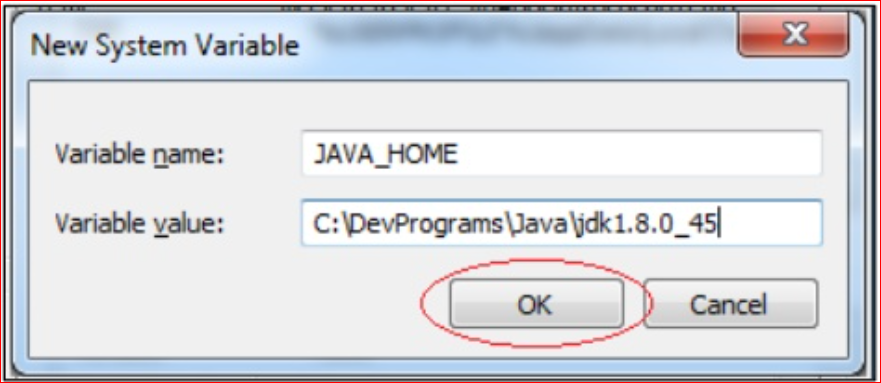
* **Bước 2:** Mở file cài đặt “jdk-\*.exe” để tiến hành cài đặt
* **Bước 3:** Cấu hình biến môi trường cho Java Việc này không bắt buộc, nhưng nếu trên máy tính của cài đặt nhiều phiên bản Java, thì việc cấu hình là cần thiết để xác định phiên bản java nào mặc định được sử dụng.
* Trên Desktop, nhấn phải chuột vào Computer, chọn Properties > Advanced system settings.



Hình 1: Thêm đường dẫn tới thư mục JDK 1

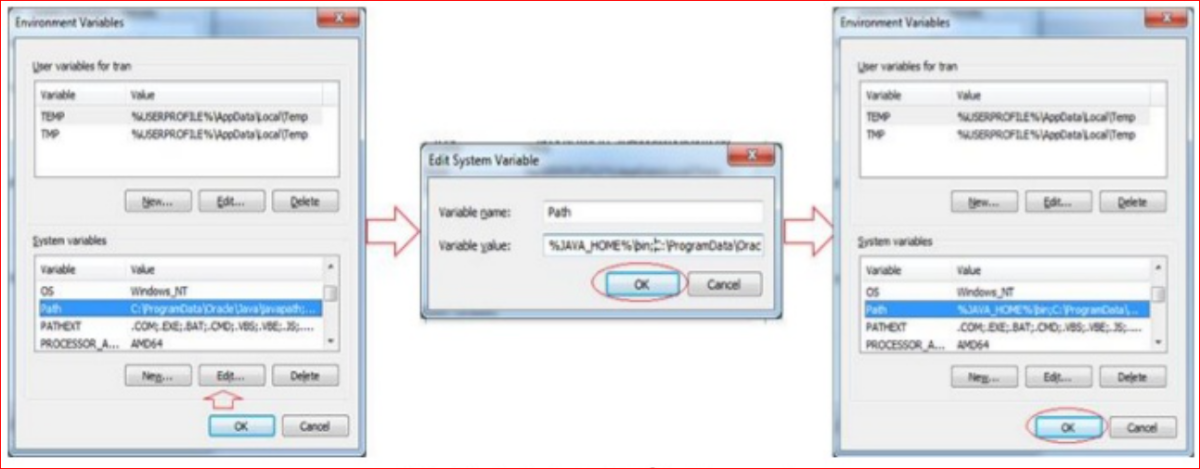
*Môi tường lập trình phát triển tích hợp:*

* Nhập vào đường dẫn tới thư mục JDK.



Hình 2: Thư mục JDK 1

* Tiếp theo đổi tên path là đã cài đặt xong Java JDK



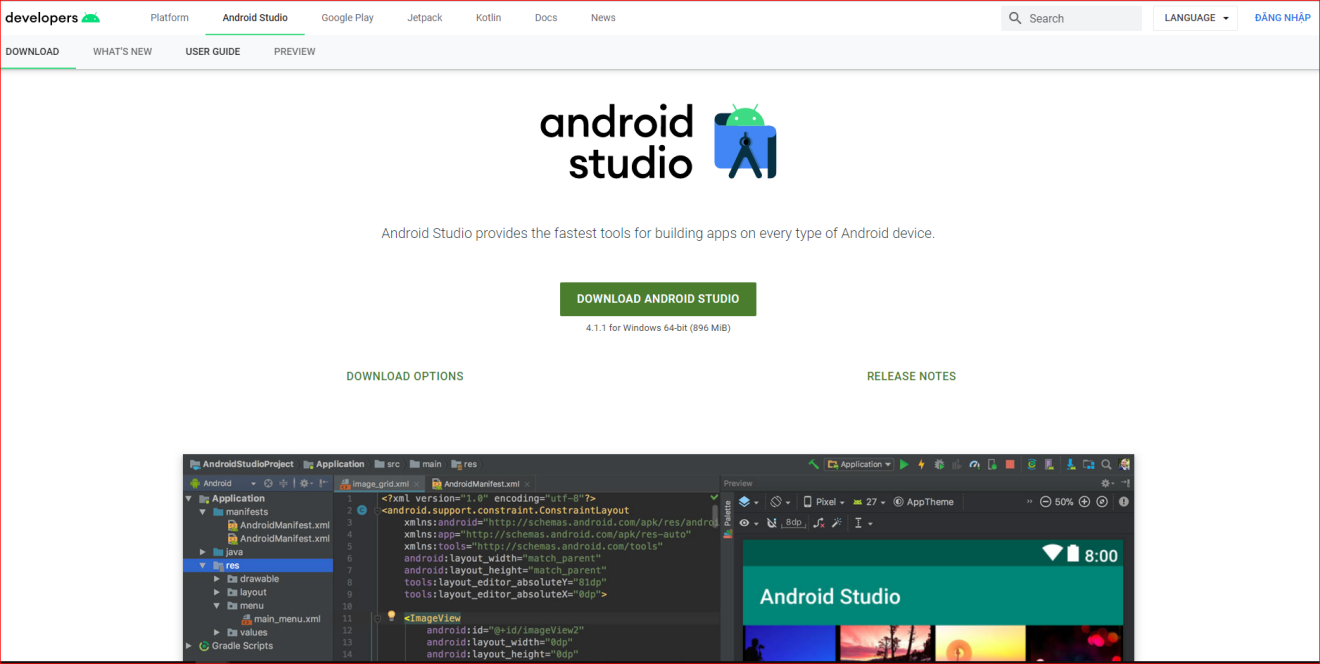
Hình 3: Đổi tên path 1

### 3.1.2 Cài đặt Android Studio

#### a. Cấu hình yêu cầu:

* Microsoft® Windows® 10/8/7/Vista (32 or 64-bit)
* Tối thiểu 4 GB RAM, cấu hình đề nghị: 8 GB RAM
* Ổ cứng trống ít nhất : 20Gb
* Độ phân giải tối thiếu 1280 x 800
* Java Development Kit (JDK) 7 trở lên
* Lựa chọn thêm cho accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

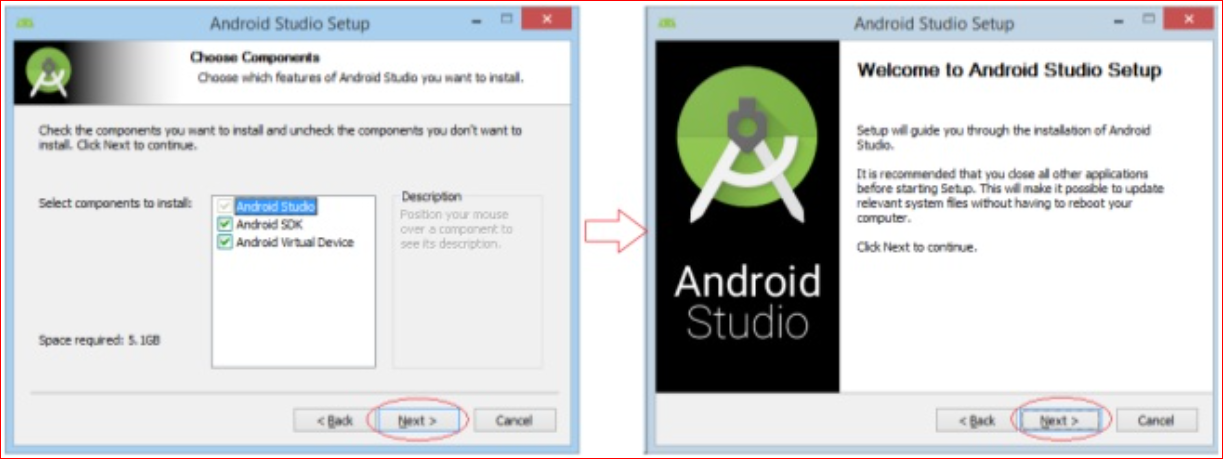
b. Tải file cài đặt Dowload file cài đặt: <https://developer.android.com/studio>



Hình 4: Giao diện trang web 1

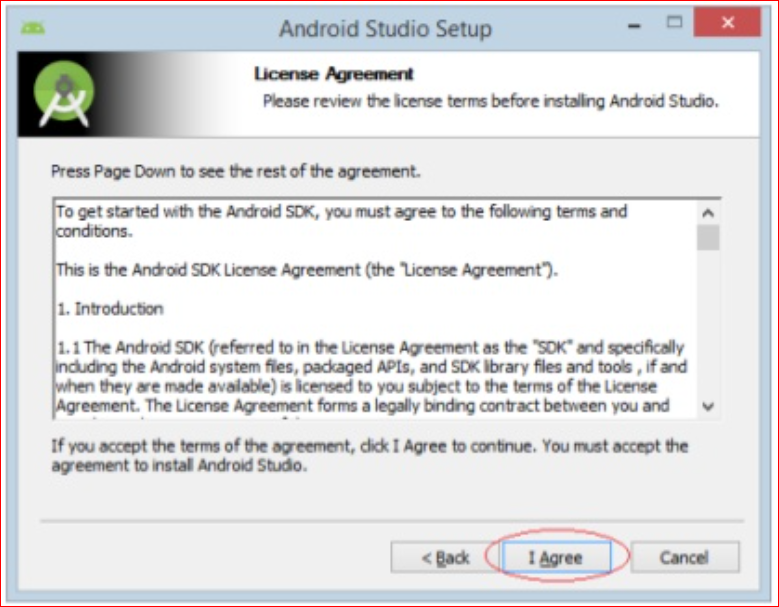
#### c. Cách cài đặt

* **Bước 1:** Mở file cài đặt và ấn Next



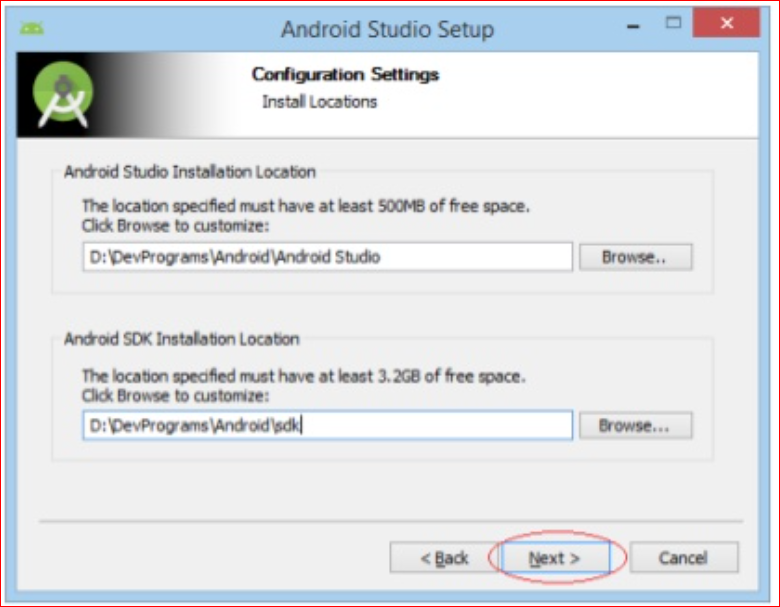
Hình 5: Giao diện cài đặt file 1

* **Bước 2:** Chọn “I agree” để xác nhận, đồng ý vợi các điều khoản và tiếp tục.



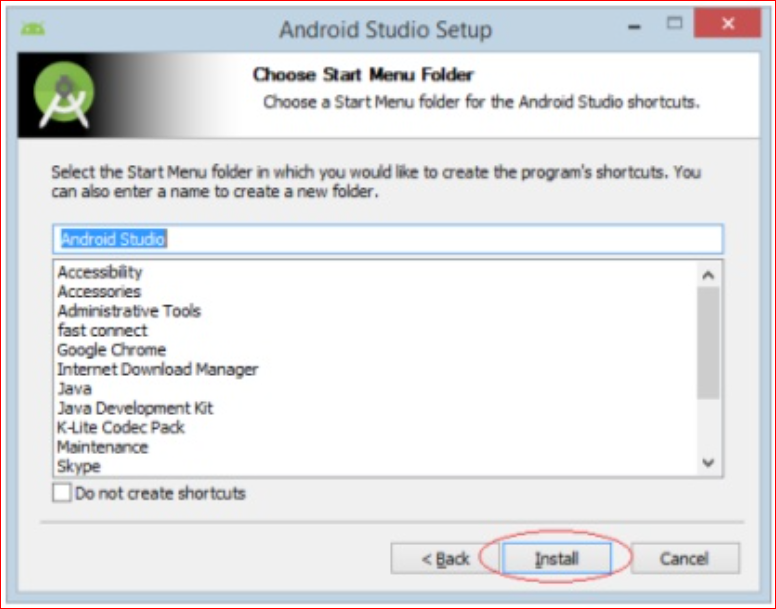
Hình 6: Giao diện điều khoản cài đặt 1

* **Bước 3:** Chọn nơi cài đặt Android Studio và Android SDK



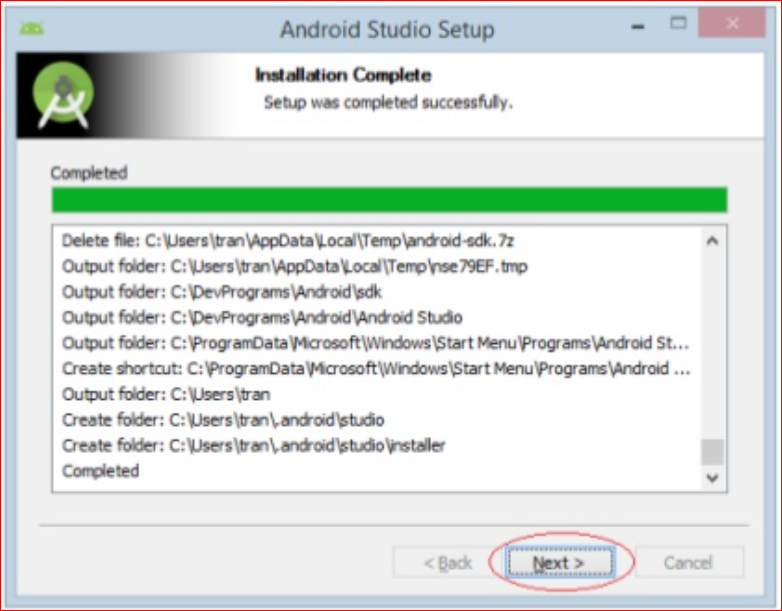
Hình 7:Giao diện cài đặt đường dẫn 1

* **Bước 4:** Chọn Start Menu Folder. Ở bước này, có thể tích tạo short cut cho ứng dụng hoặc không Bấm Install để bắt đầu cài đặt.



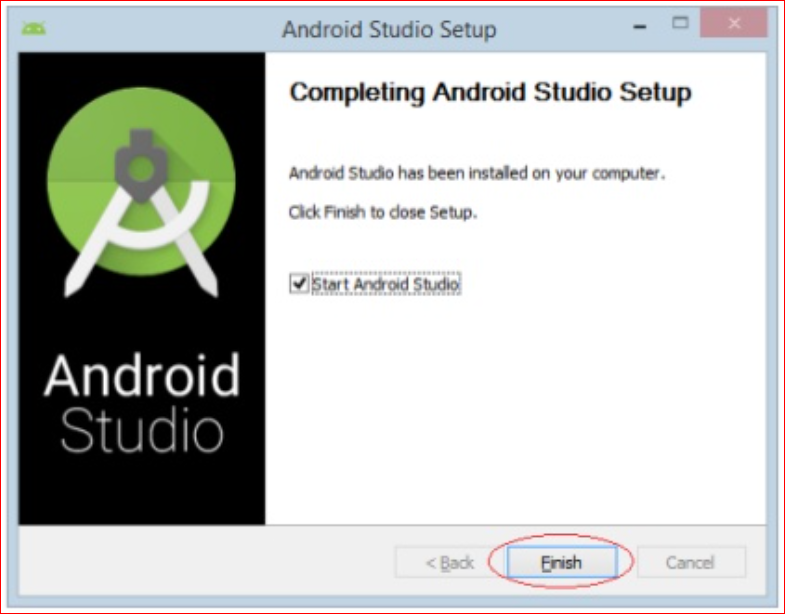
Hình 8:Giao diện đặt tên thư mục cài đặt 1

* **Bước 5:** Chờ hệ thống cài đặt các package cần thiết cho tới khi hoàn thành và bấm Next để tiếp tục.



Hình 9:Giao diện loading package 1

* **Bước 6:** Bấm Finish và click Start Android Studio để khởi động phần mềm.



Hình 10:Giao diện cài đặt thành công 1

* **BroadcastReceiver**

Broadcast Receiver: thành phần thu nhận các Intent bên ngoài gửi tới. VD: bạn

viết 1 chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn

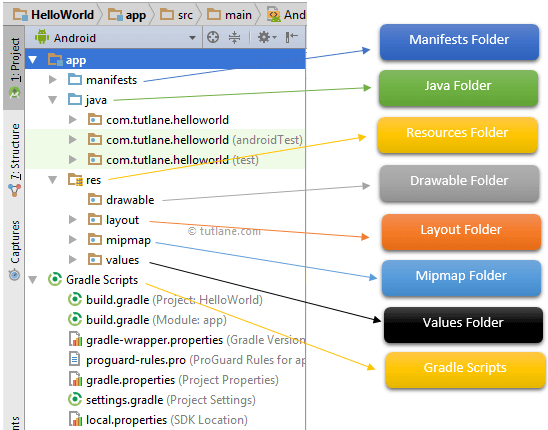
cần 1 BR để nhận biết các Intent là các cuộc gọi tới.

* **Live Wallpapers**

Live Wallpapers cho phép bạn tạo các chuyển động nền cho màn hình home. ( Có thê nói nó giông như là ảnh nền nhưng mà nó sống động hơn).

**3.2 Cấu trúc chuẩn trên Android Studio**

Để phát triển các ứng dụng Android thì bạn cần một IDE, và Android Studio là IDE chính thức được Google cung cấp miễn phí. Nó đã được tích hợp môi trường phát triển nên bạn chỉ việc sử dụng mà không cần quan tâm gì thêm.Khi thiết lập môi trường phát triển Android và tạo một ứng dụng bằng Android Studio thì bạn sẽ nhận được cấu trúc thư mục project sẽ giống như hình dưới đây.

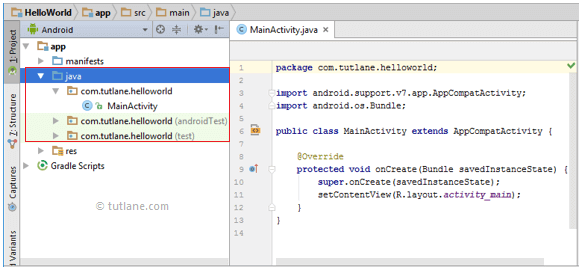


Hình 11:Mô tả 1

Cấu trúc project của bạn có thể khác với minh họa trên. Để xem cấu trúc file thực tế của project, chọn Project từ menu dropdown nằm ở góc trên bên trái (mặc định chọn Android).

**3.2.1 Thư mục java**

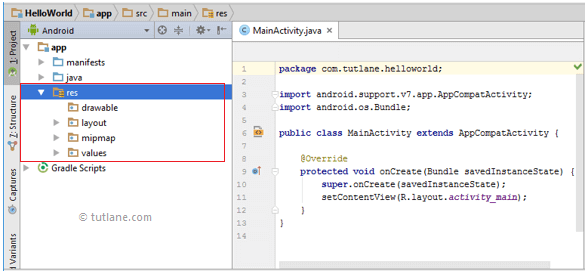
Thư mục này sẽ chứa tất cả các file mã nguồn java ( .java ) mà chúng ta sẽ tạo trong quá trình phát triển ứng dụng, bao gồm cả mã JUnit test code. Bất cứ khi nào tạo bất kỳ project / ứng dụng mới nào, file lớp MainActivity.java sẽ tự động tạo trong package là com.tutlane.helloworld, giống như dưới đây.



Hình 12: Mô tả 1

### 3.2.2 Thư mục res (Resources)

Đây là một thư mục quan trọng sẽ chứa tất cả các resource không phải code, chẳng hạn như ảnh bitmap, UI strings, XML layouts như hiển thị bên dưới.



Hình 13: Mô tả 1

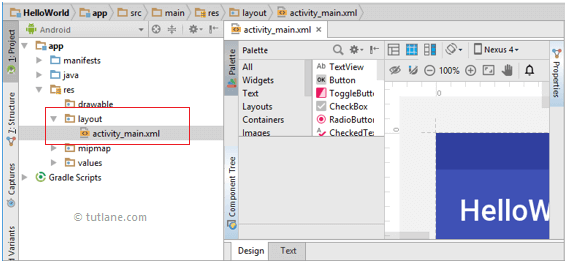
Thư mục res (Resources) sẽ chứa một số loại thư mục khác :

### 3.2.3 Thư mục drawable (res/drawable)

Nó sẽ chứa các dạng ảnh khác nhau. Tốt nhất là nên thêm tất cả các hình ảnh vào thư mục drawable ngoại trừ các biểu tượng ứng dụng / laucher.

### 3.2.4 Thư mục layout (res/layout)

Thư mục này sẽ chứa tất cả các file XML layout đã sử dụng để xác định Giao diện người dùng của ứng dụng. Sau đây là cấu trúc của thư mục layout.

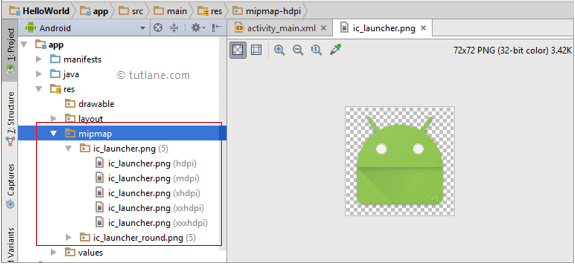


Hình 14: Mô tả 1

### 3.2.5 Thư mục Mipmap (res / mipmap)

Thư mục này sẽ chứa các biểu tượng ứng dụng / laucher được sử dụng để hiển thị trên màn hình chính. Các loại biểu tượng sẽ có tỷ trọng khác nhau như hdpi, mdpi, xhdpi, xxhdpi, xxxhdpi, để sử dụng dựa trên kích thước của thiết bị.

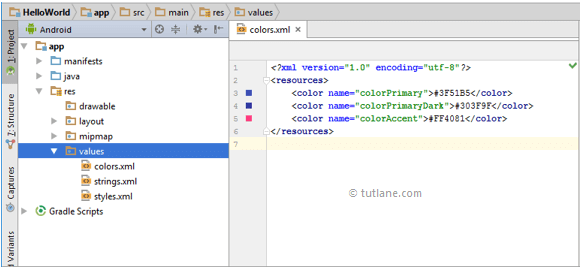
Sau đây là cấu trúc của thư mục mipmap.



Hình 16: Mô tả 1

### 3.2.6 Thư mục values (res/values)

Thư mục này sẽ chứa một số file XML khác nhau, chẳng hạn như strings, colors, styles . Sau đây là cấu trúc của thư mục values.

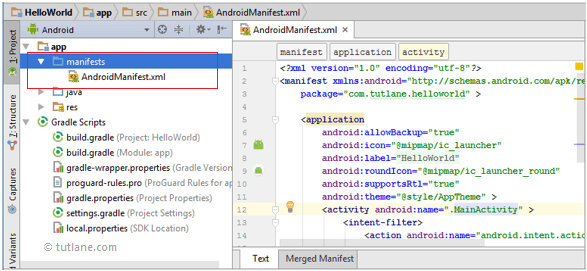


Hình 15: Mô tả 1

### 3.2.7 Thư mục Manifests

Thư mục này sẽ chứa một file Manifest ( AndroidManifest.xml ) cho ứng dụng Android . File manifest này sẽ chứa thông tin về ứng dụng , chẳng hạn như android version, access permissions, metadata, v.v. và các component của ứng dụng. File manifest sẽ hoạt động như một trung gian giữa hệ điều hành Android và ứng dụng .

Sau đây là cấu trúc của thư mục mainfests.

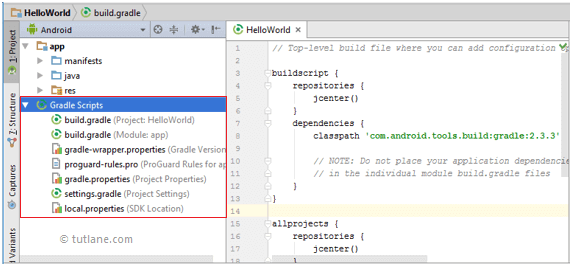


Hình 17: Mô tả 1

### 3.2.8 Gradle Scripts

Trong Android, Gradle là công cụ build hệ thống và Gradle được tích hợp sẵn vào Android Studio, và được điều khiển một cách tự động thông qua Android Studio. Trong gradle có **build.gradle (Project)**và **build.gradle (Module)** được sử dụng để build các cấu hình áp dụng cho tất cả các module ứng dụng hoặc dành riêng cho một mô-đun ứng dụng.

Sau đây là cấu trúc của Gradle Script .

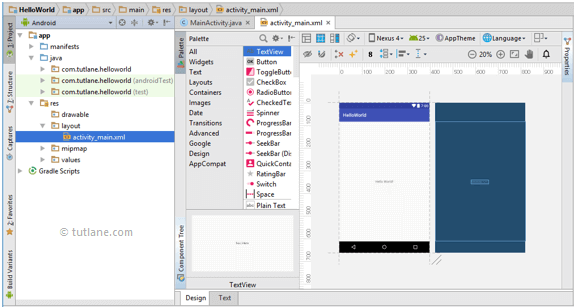


Hình 18: Mô tả 1

Sau đây là các file quan trọng cần để triển khai một ứng dụng trong android studio.

### 3.2.9 File Android Layout (Activity\_main.xml)

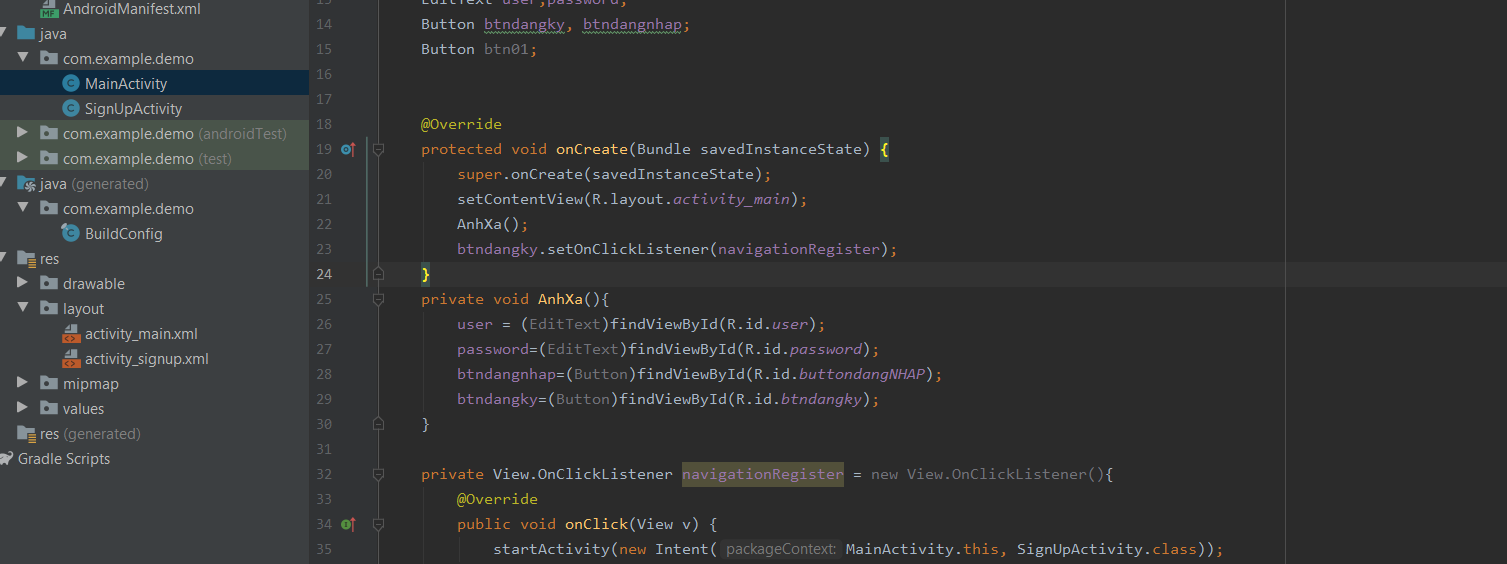
Giao diện người dùng của ứng dụng sẽ được thiết kế trong file này và nó sẽ có hai chế độ **Design**và **Text**. Nó sẽ tồn tại trong thư mục layout. Cấu trúc của file **activity\_main.xml** trong chế độ **Design**như dưới đây.



Hình 19: Mô tả 1

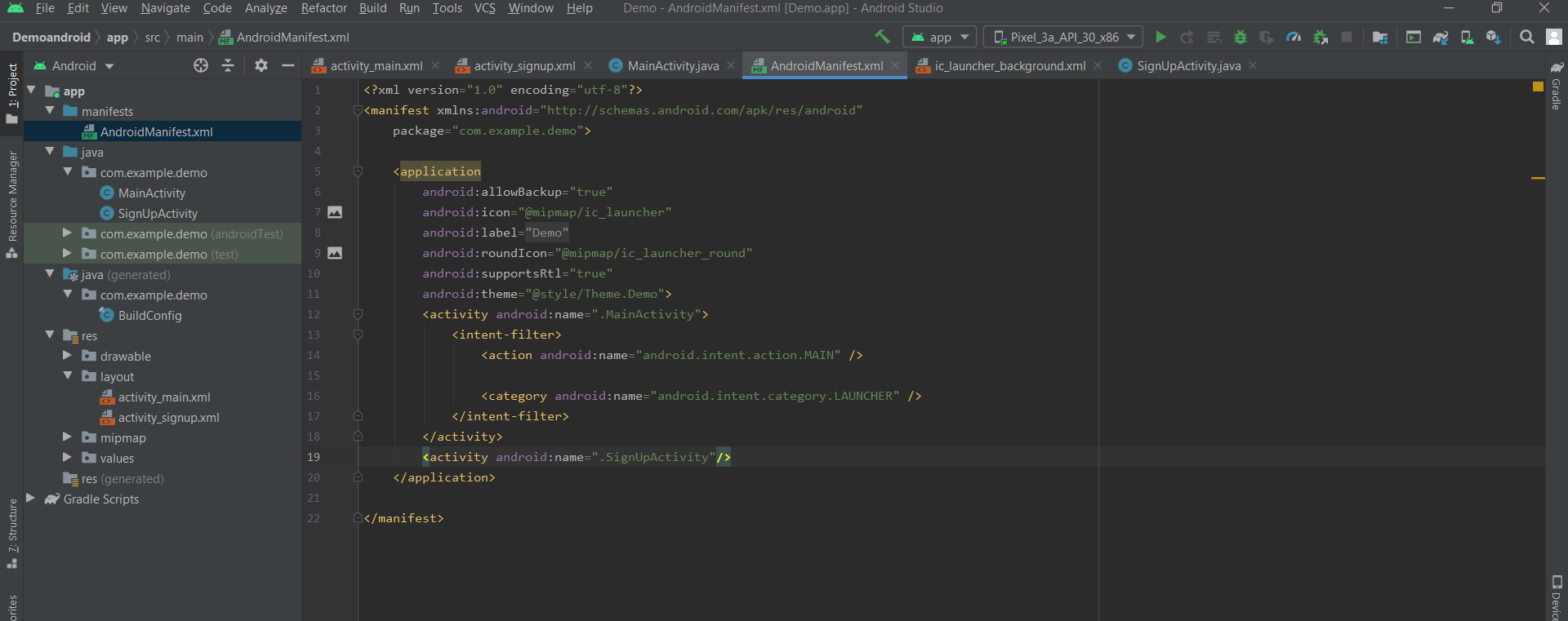
### 3.2.10 File Android Main Activity (MainActivity.java)

File main activity trong ứng dụng Android là **MainActivity.java** và nó nằm trong thư mục **java**. File **MainActivity.java** sẽ chứa mã java để xử lý tất cả các activity liên quan đến ứng dụng .



### 3.2.11 File Android Manifest (AndroidManifest.xml)

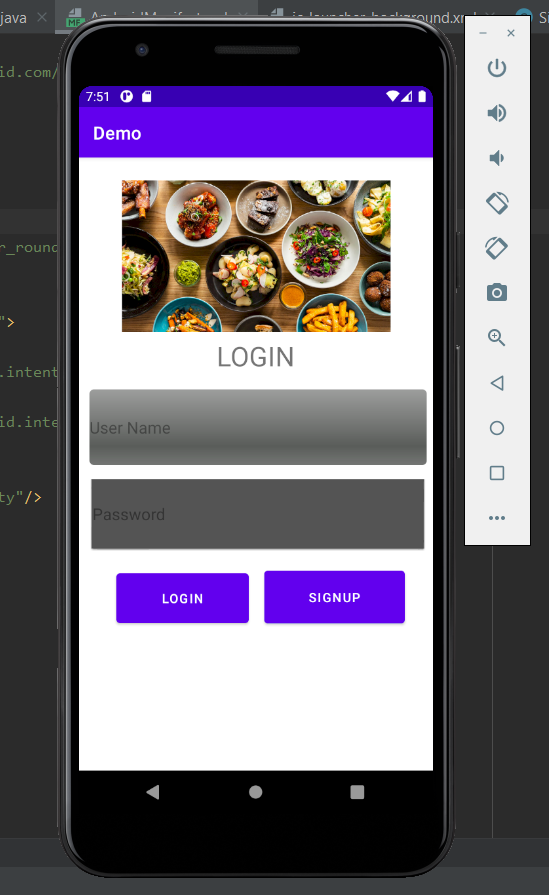
Nói chung, ứng dụng sẽ chứa nhiều activity và cần xác định tất cả các activity trong file AndroidManifest.xml . Trong file manifest, ta cần đề cập đến main activity cho ứng dụng bằng cách sử dụng các thuộc tính MAIN action và LAUNCHER category trong các **intent filters (<intent-filter>)**. Trong trường hợp nếu ta không đề cập đến MAIN action và LAUNCHER category cho main activity, biểu tượng ứng dụng sẽ không xuất hiện trong danh sách ứng dụng của màn hình chính.



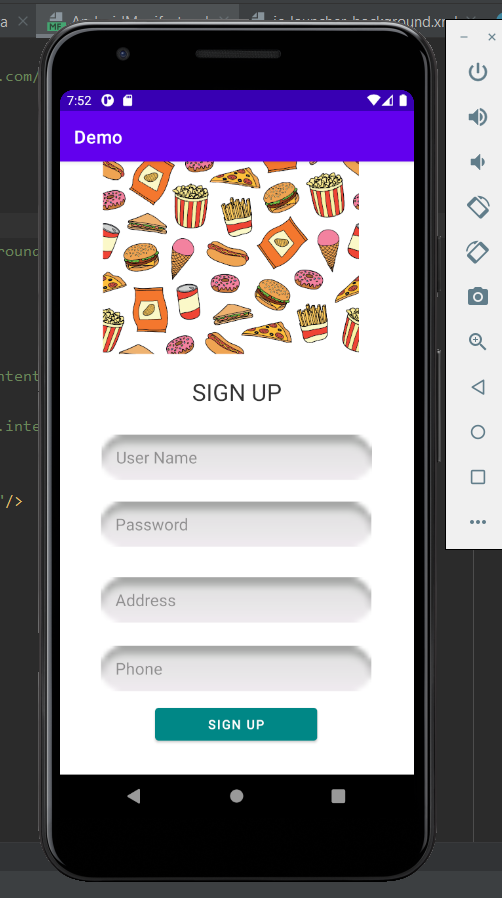
# CHƯƠNG 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

**4.1. Kết quả triển khai một số chức năng chính**

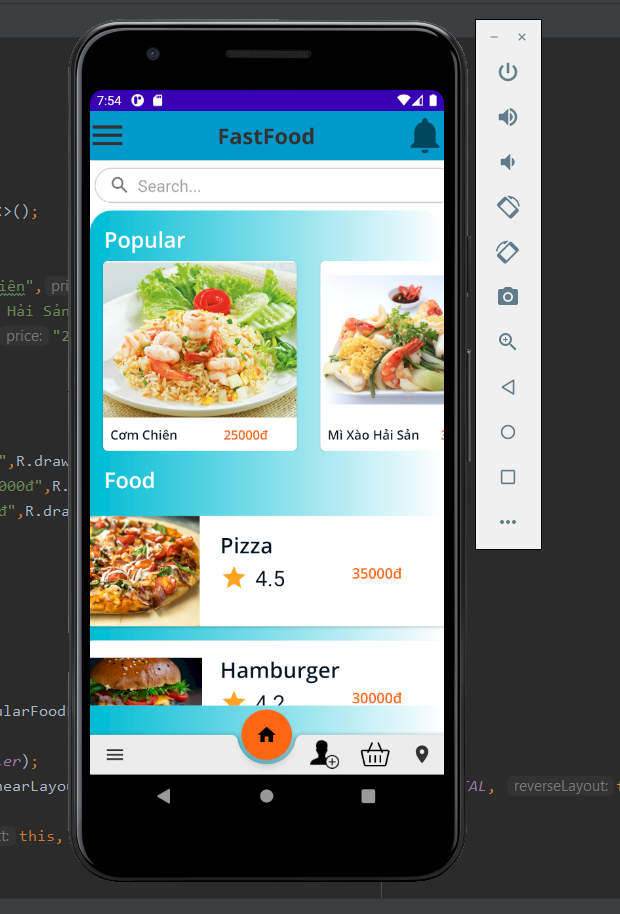
**- Giao diện chính**



**Giao diện đăng kí:**



**Giao diện đặt đồ ăn của người dùng**



CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1:Ưu điểm đề tài**

* Cho phép khách hàng có thể thao tác dễ dàng với các chức năng.
* Đối với chức năng tra cứu, có thể dễ dàng tìm ra những món ăn mà người dùng cần tìm
* Đảm bảo sự tương tác giữa người sử dụng và ứng dụng.
* Giao diện đơn giản, tone màu tươi sáng giúp làm người dùng cảm thấy thoải mái.

**2:Hạn chế còn tồn tại**

* Một số chức năng chưa thật tốt.

**3:Kết Luận**

Sau khi thực hiện đồ án với đề tài App FastFood đã cho em được nhiều kiến thức sâu về lập trình android, đặc biệt là ngôn ngữ dùng để phát triển ứng dụng java. Bên cạnh đó, sau khi thực hiện đồ án đã cung cấp cho em những kinh nghiệm về việc phân tích, thiết kế, triển khai phát triển một ứng dụng.

Tuy em đã cố gắng nhưng còn một số những vấn đề chưa thể làm được.

**4:Hướng phát triển**

* Hoàn thiện các chức năng còn thiếu sót.
* Hoàn thiện tốt nhất, tối ưu các chức năng đã thực hiện.
* Phát triển thêm nhiều chức năng mới nữa.

TÀI LIỆU THAM KHẢO

[1] <https://developer.android.com/studio>

[2] https://freetuts.net/cau-truc-thu-muc-project-android-2158.html