

# **ĐỒ ÁN CƠ SỞ**

## **Apps Bất Động Sản**

Ngành: **CÔNG NGHỆ THÔNG TIN**  
Chuyên ngành: **Công Nghệ Phần Mềm**

Giảng viên hướng dẫn: **NGUYỄN MẠNH HÙNG**

Sinh viên thực hiện: **Nguyễn Trương Thanh Nhã**

MSSV : 1411060908

Lớp : 15DTH01

TP. Hồ Chí Minh, 2018

# **ĐỒ ÁN CƠ SỞ**

## **Apps Bất Động Sản**

Ngành: **CÔNG NGHỆ THÔNG TIN**  
Chuyên ngành: **Công Nghệ Phần Mềm**

Giảng viên hướng dẫn: **NGUYỄN MẠNH HÙNG**

Sinh viên thực hiện: **Nguyễn Trương Thanh Nhã**

**MSSV : 1411060908**

**Lớp : 15DTH01**

**TP. Hồ Chí Minh, 2018**

<b>LỜI NÓI ĐẦU.....</b>	<b>5</b>
<b>PHẦN MỞ ĐẦU .....</b>	<b>5</b>
<b>Chương 1:Tổng Quan .....</b>	<b>5</b>
1.1 Sơ lược về các hệ điều hành .....	5
1.2 Android .....	6
1.3 Android sdk.....	7
1.6 Đối tượng khách hàng.....	8
<b>CHƯƠNG 2: Cơ sở lý thuyết.....</b>	<b>13</b>
2.1 Delving với máy ảo dalvik.....	13
2.2. Kiến trúc hệ điều hành Android.....	14
2.3. Tầng ứng dụng (applications) .....	14
2.4. Application Framework .....	15
2.5. Library.....	15
2.6. Android runtime.....	16
2.7. Linux kernel.....	16
2.8. Các thành phần trong ứng dụng Android .....	16
<b>Chương 3: Khảo sát và phân tích hệ thống.....</b>	<b>34</b>
3.1 Khảo sát thực tế .....	34
3.3 GIỚI THIỆU CÁC NGÔN NGỮ LẬP TRÌNH ANDROID .....	36
<b>Chương 4:Thiết kế hệ thống .....</b>	<b>38</b>
4.1 Thiết kế giao diện apps .....	38
4.2.Chức năng chính .....	44
4.3 kho dữ liệu .....	44
4.6. Bảng phân tích thiết kế chức năng.....	46
<b>Chương 5:Cài đặt apps .....</b>	<b>47</b>
5.1 Đóng gói ứng dụng .....	47
5.2 Đưa lên ch-play .....	51
<b>Chương 6:Tài liệu tham khảo .....</b>	<b>53</b>

## Mục Lục Hình Ảnh

<b>Chương 1:Tổng Quan</b>	
Hình 1. 1. Hình MBND WORKSHOP.....	9
Hình 1. 2 Có 4 Yếu Tố Cần Phải Nhớ.....	9
Hình 1. 3 Phiếu Bán Hàng.....	10
Hình 1. 4 Cải thiện Phiếu Bán hàng Mở Rộng.....	10
Hình 1. 5 Cải thiện Phiếu Bán hàng Tối Ưu Hoá.....	11
Hình 1. 6 Kết hợp mở rộng và tối ưu hoá.....	11
Hình 1. 7 Bảng thống kê seo của google theo các keyword.....	12

Hình 1. 8 Thời Điểm Các Giai Đoạn Của Khách Hàng .....	12
Hình 1. 9 Xác định nhu cầu của khách hàng .....	12
Hình 1. 10 Tối Ưu Hoá Quản Cáo.....	13
Hình 1. 11 Mô hình quản cáo cũ .....	13
<b>CHƯƠNG 2: Cơ sở lý thuyết</b>	
Hình 2. 1 Cấu trúc Stack hệ thống Android .....	14
Hình 2. 2 Các thành phần trong một Android project.....	16
Hình 2. 3 Khai báo tag <activity> trong file AndroidManifest.xml .....	19
Hình 2. 4 Lược đồ vòng đời của 1 Activity.....	20
Hình 2. 5 Truyền thông tin giữa hai Activity khác nhau.....	22
Hình 2. 6 Lược đồ vòng đời của một service .....	25
Hình 2. 7 Mô hình dữ liệu .....	26
Hình 2. 8 Các thành phần của một url.....	26
Hình 2. 9 Khai báo tag <provider> trong file AndroidManifest.xml.....	28
Hình 2. 10 Ví dụ một cây views .....	29
Hình 2. 11 LinearLayout .....	30
Hình 2. 12 AbsoluteLayout .....	30
Hình 2. 13 TableLayout.....	31
Hình 2. 14 RelativeLayout .....	31
Hình 2. 15 FrameLayout.....	32
Hình 2. 16 ScrollView .....	32
Hình 2. 17 Màn hình làm việc của sqlite manager.....	34
<b>CHƯƠNG 3: Khảo sát và phân tích hệ thống</b>	
<b>CHƯƠNG 4: Thiết kế hệ thống</b>	
Hình 4. 1 Màn Hình Chào .....	38
Hình 4. 2 Màn hình Đăng Nhập .....	39
Hình 4. 3 Màn hình xem tác cả sản phẩm bài viết .....	41
Hình 4. 4 Màn hình menu.....	42
Hình 4. 5 Màn hình tạo bài viết.....	43
Hình 4. 6 Màn hình list tác cả sản phẩm .....	43
Hình 4. 7 Màn hình thông tin ứng dụng.....	44
Hình 4. 8 Màn hình tạo bài viết qua trang tác cả sản phẩm .....	44
Hình 4. 9 Sơ đồ phân cấp chức năng.....	46
<b>CHƯƠNG 5: Cài đặt apps</b>	
Hình 5. 1 Quá trình lên ý tưởng và ra apps .....	48
Hình 5. 2 Hoạt động .....	49
Hình 5. 3 Chọn Generate Signed APK.....	50
Hình 5. 4 Đăng ký chữ ký.....	50
Hình 5. 5 lưu tại nơi chứa thư mục bạn chọn .....	51
Hình 5. 6 Tìm đến thư mục bạn đã build ra apps và xem có file apk không .....	51
Hình 5. 7 Tạo tài khoản google development .....	52

## Mục lục bảng biểu

Chương 1:Tổng quan	
Chương 2:Cơ sở lý thuyết	
Chương 3:Khảo sát và phân tích hệ thống	
Chương 4:Thiết kế hệ thống	
Bảng biểu 4. 1 Đăng Nhập .....	39
Bảng biểu 4. 2:Đăng Ký .....	39

Bảng biểu 4. 3 Reset.....	40
Bảng biểu 4. 4 Bảng phân tích chức năng của khách hàng .....	46

## LỜI NÓI ĐẦU

### PHẦN MỞ ĐẦU

Android là một hệ điều hành mã nguồn mở và dựa trên Linux dành cho các thiết bị di động như điện thoại thông minh và máy tính bảng. Android được phát triển bởi Open Handset Alliance, do Google và các công ty khác đứng đầu. Hướng dẫn này sẽ dạy cho bạn lập trình Android cơ bản và cũng sẽ đưa bạn qua một số khái niệm trước liên quan đến phát triển ứng dụng Android.

Sự hấp dẫn với vẻ đẹp toàn diện các ứng dụng chạy trên nền Android thường “bóng bẩy” và mang tính thẩm mỹ ở nhiều cấp độ. Các hiệu ứng chuyển đổi luôn nhanh và rõ ràng, bố cục layout và typography (nghệ thuật chữ) sinh động và đầy ý nghĩa. Các biểu tượng của ứng dụng đều là những tác phẩm nghệ thuật theo đúng nghĩa của chúng. Cũng giống như việc tạo ra 1 công cụ hữu ích khi thiết kế ứng dụng, bạn hãy cố gắng kết hợp vẻ đẹp sự đơn giản với mục tiêu để tạo ra một trải nghiệm kỳ diệu, vừa dễ dùng lại vừa hữu hiệu. Ứng Dụng Android làm cho cuộc sống trở nên dễ dàng và dễ hiểu hơn. lần đầu tiên người sử dụng ứng dụng, họ cần nắm bắt những tính năng quang trọng nhất một cách trực quan. Tuy nhiên việc thiết kế sẽ không dừng lại ở lần trải nghiệm lần đầu tiên. Các ứng dụng android đã loại bỏ những công việc đang diễn ra như quản lý file và đồng bộ những tác vụ đơn giản không vào giờ đòi hỏi những thủ tục phức tạp và những tác vụ phức tạp đều được thiết kế phù hợp với bàn tay và khối óc con người. Người dùng ở mọi lứa tuổi và nền tảng văn hoá khác nhau đều cảm thấy tự tin trải nghiệm ứng dụng và họ không bao giờ bị choán ngợp bởi quá nhiều lựa chọn hoặc những đồ hoạ động thích hợp trong ứng dụng. Nếu bạn chỉ đơn thuần tạo ra ứng dụng dễ dùng thì vẫn chưa đủ. Các ứng dụng android cho phép người dùng trải nghiệm những điều mới và sử dụng chúng theo những chiều hướng mang tính sáng tạo. Android giúp người dùng kết hợp các ứng dụng thành những quy trình làm việc mới thông qua chế độ đa nhiệm, các thông báo và chia sẻ giữa các ứng dụng. Đồng thời, ứng dụng của bạn cũng cần tạo cảm giác riêng tư, giúp người dùng tiếp cận với công nghệ hiện đại một cách rõ ràng và uyển chuyển

## Chương 1: Tổng Quan

### 1.1 Sơ lược về các hệ điều hành

Hệ điều hành là chương trình chạy trên hệ thống máy tính, quản lý các tài nguyên trên máy tính và là môi trường cho các ứng dụng chạy trên nó. Ngày nay khái niệm hệ điều hành không chỉ là trên máy tính mà còn được mở rộng cho nhiều thiết bị điện tử khác chẳng hạn như điện thoại thông minh (SMART-PHONE), các thiết bị cầm tay máy tính bảng, đồng hồ thông minh...v...v như vậy hệ điều hành chạy trên hệ thống máy tính di động cao, với đặt thù đó, hệ điều hành di động phải có những khả năng đặc biệt mà những hệ điều hành thông thường không có được chẳng hạn như nó phải chạy trên hệ thống máy có cấu hình máy hạn chế về tốc độ vi xử lý, bộ nhớ sử dụng, phải chạy được ổn định liên tục trong suốt thời gian dài mà chỉ sử dụng một lượng điện năng nhỏ, trong suốt thời gian chạy đó có thể duy trì các kết nối mạng không dây để đảm bảo liên lạc ngoài ra còn một số hệ điều hành tiêu biểu:

MS DOS, MS WINDOW , MACOS, LINUX, UNIX .

Trên điện thoại thông minh: Android, Sybian, Window Mobile, iPhone OS, BlackBerry, S60, Bada OS, Palm OS.

Ngoài ra còn có các hệ điều hành chạy trên MainFrame, Server, Thẻ chip...

Trong phạm vi đồ án này chúng ta sẽ nói về hệ điều hành android dành cho điện thoại.

## 1.2 Android

Android –hệ điều hành dành cho điện thoại di động được phát triển bởi google và ngày càng trở nên phổ biến với các hãng liên tục ra mắt các mẫu điện thoại sử dụng android. Android được xây dựng trên nhân linux và được phân phối miễn phí. Không giống như windows mobile và được xây dựng trên nhân linux và được phân phối miễn phí. không giống như windows mobile và apple mobile, tuy cả hai điều cùng cấp môi trường phát triển ứng dụng phong phú và đơn giản dễ tiếp cận nhưng luôn có sự ưu tiên cho các ứng dụng mặc định có sẵn của hệ điều hành android mọi ứng dụng điều viết trên cùng một tập api, thế nên không có sự phân biệt giữa các ứng dụng mặc định và các ứng dụng của bên thứ ba. người dùng hoàn toàn có thể thay thế mọi ứng dụng mặc định bằng các ứng dụng yêu thích của mình thậm chí ngay cả màn hình thực hiện cuộc gọi mà màn hình nhà(Home Screen). Các nhà phát triển ứng dụng cho android dựa trên ngôn ngữ java. Sự ra mắt của android vào ngày 5/11/2007 gắn với sự thành lập liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng và phần mềm viễn thông nhằm mục đích tạo nên chuẩn mở cho điện thoại di động trong tương lai.

Google công bố hầu hết các mã nguồn android theo bản cấp phép Apache.

Các ứng dụng có sẵn trên android

Một điện thoại android thông thường đi kèm với một vài ứng dụng có sẵn, bao gồm:

- Một trình email tương thích với gmail.
- Chương trình quản lý tin nhắn sms.
- Chương trình quản lý thông tin cá nhân, bao gồm cả lịch làm việc, danh bạ và được đồng bộ hoá với google.

Phiên bản thu gọn của google map cho điện thoại bao gồm

- StreetView
- Tìm kiếm địa chỉ
- Chỉ đường
- Tình trạng giao thông...
- Trình duyệt web dựa trên nền webkit.

Chương trình tán gẫu(chat)

Chương trình đa phương tiện(chơi nhạc, xem phim...)

- Android Market Place cho phép người dùng tải về và cài đặt các ứng dụng mới.
- Tất cả các ứng dụng có sẵn đều được viết bằng ngôn ngữ java và sử dụng android sdk.
  - Các dữ liệu về thông tin người dùng được các ứng dụng có sẵn sử dụng như thông tin về danh bạ vẫn hoàn toàn có thể được sử dụng bởi các ứng dụng của bên thứ ba. Tương tự vậy, ứng dụng của bạn hoàn toàn có thể được xử lý các sự kiện như các cuộc gọi đến, nhận một tin nhắn mới.... thay cho các ứng dụng có sẵn.
- Truy cập phần cứng
  - Android bao gồm các thư viện api giúp đơn giản hoá tối đa việc sử dụng phần cứng của thiết bị. Điều đó đảm bảo rằng bạn không cần phải bận tâm nhiều đến việc

ứng dụng của mình có thể chạy như mong đợi trên nhiều thiết bị khác nhau hay không, miễn là thiết bị đó có hỗ trợ android.

➤ Android sdk bao gồm api cho phần cứng: gps, camera, kết nối mạng, wifi, bluetooth, con quay gia tốc, màn hình cảm ứng, quản lý năng lượng....

- Dịch vụ chạy nền

➤ Android hỗ trợ các ứng dụng và dịch vụ được thiết kế chạy ẩn. Do kích thước nhỏ của màn hình điện thoại nên tại một thời điểm chỉ có thể thấy một ứng dụng. Dịch vụ chạy nền giúp tạo ra các thành phần ứng dụng vô hình để thực hiện tự động một tác vụ nào đó mà không cần phải có sự tương tác của người dùng, ví dụ như một dịch vụ chạy nền có chức năng chặn cuộc gọi đến đối với các số điện thoại có trong Black List chẳng hạn.

- Sqlite database

➤ Bởi vì tính chất nhỏ gọn và bị hạn chế về phần cứng của điện thoại di động, cho nên đòi hỏi việc lưu trữ và truy xuất dữ liệu phải nhanh chóng và hiệu quả. Android hỗ trợ hệ quản trị nhỏ gọn sqlite, và cung cấp cho ứng dụng các api để thao tác. Mặc định mỗi ứng dụng đều được chạy trong Sandbox(hộp cát) điều này có nghĩa là nội dung của từng database ứng với từng ứng dụng chỉ có thể truy cập bằng chính ứng dụng đó. Tuy nhiên có các cơ chế các ứng dụng chia sẻ, trao đổi các database với nhau.

- Hệ thống thông báo

➤ Thông báo là cách thức tiêu chuẩn mà ở đó thiết bị báo cho người dùng đã có một sự kiện nào đó đã xảy ra. Chẳng hạn như có cuộc gọi tới, máy sắp hết pin... sử dụng các api bạn có thể cho ứng dụng của mình thông báo tới người dùng bằng âm thanh, rung, hoặc thậm chí cả đèn led của thiết bị.

- Tối ưu hoá bộ nhớ và quản lý tiến trình

➤ Việc quản lý bộ nhớ và tiến trình trong android cũng có một chút khác biệt giống như công nghệ java và .net, android sử dụng một bộ run-time của riêng mình với công nghệ ảo hoá để quản lý bộ nhớ của ứng dụng đang chạy, không giống như những nền tảng khác, android run-time cũng đồng thời quản lý luôn cả thời gian sống của ứng dụng. android đảm bảo các ứng dụng đều được đáp ứng bằng cách dừng và huỷ các tiến trình không cần thiết để giải phóng các tài nguyên cho các tiến trình có độ ưu tiên cao hơn. Trong bối cảnh đó, độ ưu tiên được xác định tùy thuộc vào ứng dụng mà người đang tương tác. Android đảm bảo rằng các ứng dụng được huỷ một cách nhanh chóng, đồng thời chúng là khởi động là nhanh cũng không kém nếu cần. Điều này thật sự quan trọng trong một môi trường mà ở đó bản thân ứng dụng không thể tự kiểm soát được thời gian sống cho mình.

### 1.3 Android sdk

- Android software development kit(sdk) Bộ sdk của android bao gồm mọi thứ cần thiết giúp bạn có thể lập trình, debug, test ứng dụng android.

- Android api: cốt lõi của bộ sdk là thư viện chứa các hàm api và google cũng chỉ sử dụng bộ api này để xây dựng các ứng dụng có sẵn cho android.

Development tool: Sdk bao gồm rất nhiều công cụ để giúp biên dịch, sửa lỗi và hỗ trợ trong việc lập trình ứng dụng.

- Android emulator: trình giả lập thiết bị chạy android thực sự với nhiều skin thay thế cực kỳ tiện lợi cho việc test ứng dụng android ngay trên máy tính mà không cần phải thông qua một thiết bị chạy android thật. Sau đây là kiến trúc của mọi ứng dụng android:

- Activity manager: kiểm soát vòng đời của activity.
- View : xây dựng giao diện người dùng cho activity.
- Notification manager: cung cấp mộ cơ chế thống nhất và an toàn để ứng dụng có thể đưa ra các thông báo cho người dùng.
- Content provider: giúp trao đổi và chia sẻ dữ liệu giữa các ứng dụng với nhau.
- Resource manager: hỗ trợ quản lý các tài nguyên khôn là code như các chuỗi hình ảnh, và âm thanh...
- Các thư viện của android
  - Andoird cung cấp các gói api để phát triển ứng dụng.Sau đây là các api mà tất cả các thiết bị android điều tối thiểu phải hỗ trợ để giúp cho chúng ta một cái nhìn tổng quat về thư viện này.
  - Android.util: gói api lõi, chứa các class cấp thấp như container, string dormatter, xml parsing.
  - Android.os: truy cập tới chức năng của hệ điều hành như gửi nhận tin nhắn, giao tiếp nội bộ giữa các ứng dụng, thời gian...
  - Androd.graphics: cung cấp các lớp liên quan tới xử lý đồ hoạ ở mức thấp.Hỗ trợ các hàm cơ bản như vẽ điểm, vẽ miền, tô màu... trên khung canvas.
  - Android.text: cung cấp các hàm phân tích và xử lý chuỗi.
  - Android.database: cung cấp các lớp cấp thấp cần thiết để làm việc với database.
  - Android.content: dùng để quản lý các tài nguyên, các nội dung và các gói.
  - Android.view: view là lớp cha của mọi lớp giao diện người dùng.
  - Android.widget: được thừa kế từ lớp view, bao gồm các lớp cơ bản để xây dựng giao diện widget như: list, button, layout....
  - Android.map: gói api cấp cao, dùng để truy cập tới các chức năng của google mapp.
  - Andoid.app: gói api cấp cao, bao gồm các activity và service hay lớp cơ sở cho mọi ứng dụng android.
  - Android.telephone: cung cấp cho bạn khả năng tương tác trực tiếp với các chức năng cơ bản của một điện thoại như nghe, gọi, tin nhắn.
  - Andoid.webkit: cung cấp một webview control trên nền webkit để có thể nhúng ứng dụng, cùng với các api điều khiển cơ bản như stop, refresh, cookiemanager...

#### 1.4 Giới thiệu đề tài

Vào ngành công nghiệp 4.0 hiện nay với xu hướng mới trong lĩnh vực phần mềm hiện nay rất nhiều sản nhà đất bất động sản rất muốn giúp bán đi căn hộ nhà ở một cách nhanh nhất và vì vấn đề đó em cùng thảo luận với một vài sản giao dịch bất động sản cùng nhau phát triển và làm ứng dụng này cho nền công nghiệp 4.0 thiện hay bao gồm đối tc chính là sản giao dịch alibaba và sản giao dịch đặt gia và cùng một số sản giao dịch khác hiện nay cùng với các anh chị kinh doanh buôn bán nhà hộ tự bán với tiêu chí bán được căn hộ nhà ở phù hợp với tất cả mọi người với chi phí giá cả thuận tiện cho mọi người, thông thường mọi người hay tìm nhiều căn hộ giá quá cao so với mặt bằng hiện có ứng dụng này có thể giải quyết khắc phục được tình trạng này hiện nay

#### 1.6 Đối tượng khách hàng

Trước tiên, bạn phải xác định được phân khúc dự án bạn chuẩn bị bán để xác định khách hàng mục tiêu của bạn. Đó là dự án thuộc phân khúc nào? Trung cấp, cao cấp hay bình dân. Để xác định được bạn có thể căn cứ vào giá, thiết kế căn hộ, diện tích, giá trị



đầu tư và đặc biệt là thông điệp chủ đầu tư sử dụng. Tiếp đến, hãy nghiên cứu thật chi tiết, đầy đủ về dự án, đối thủ và thị trường. Đối với đối thủ trực tiếp, bạn hãy xem cách họ đã sử dụng thành công trong việc tìm kiếm khách hàng mua dự án. Khâu này giúp bạn biết mình đang ở đâu trong thị trường rộng lớn ngoài kia, hơn nữa bạn còn trở nên hiểu biết về bất động sản. Bạn nên nhớ thông tin ngày càng dễ dàng tiếp cận khách hàng mua bất động sản qua các phương tiện như facebook, google, báo chí, nên khách hàng chắc chắn sẽ tìm hiểu những dự án lân cận, cùng phân khúc thị trường. Nếu khách hàng bảo “chị thấy dự án kia cũng được lắm, căn hộ của em có gì nổi bật không?” mà bạn lúng túng sẽ là một điểm trừ cực kỳ lớn khiến khách hàng mất niềm tin ở bạn. Kinh doanh bất động sản là cuộc đua về niềm tin và trải nghiệm khách hàng.



Nhưng hiện nay lại đang được áp dụng cách làm **ngắn hạn, thời vụ...**

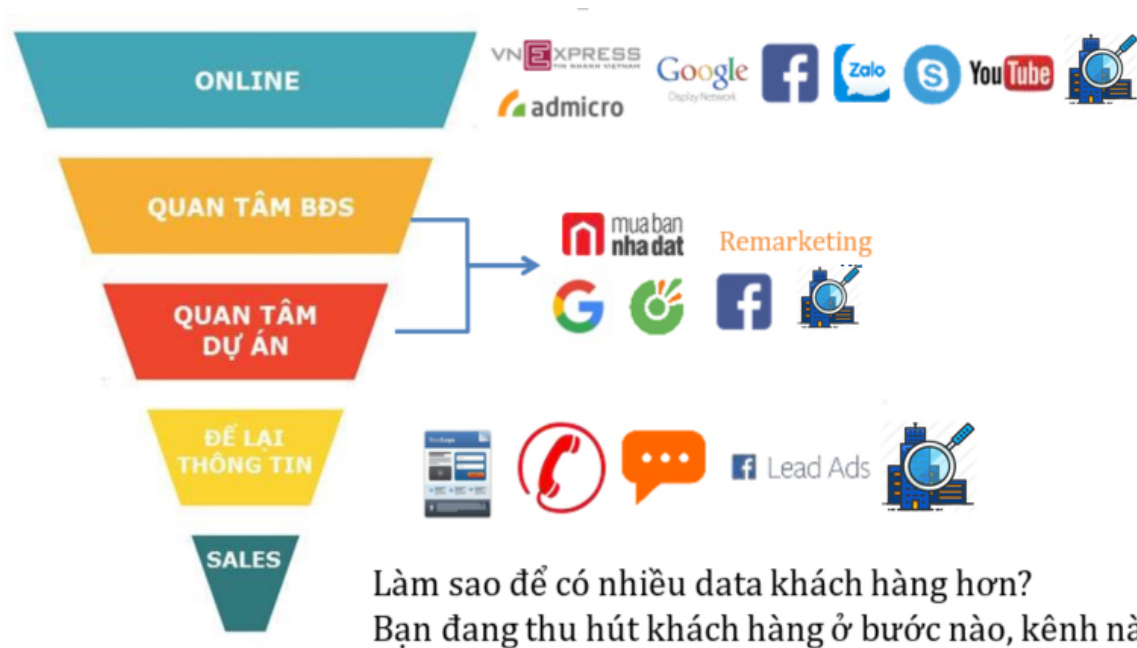
Hiện tại đa phần các công ty chỉ chú trọng vào giai đoạn khi mở bán và tiến hành dự án mà không đầu tư vào giai đoạn trước và sau dự án.

Hình 1. 1. Hình MBND WORKSHOP

**ĐÚNG**

**Khách Hàng**  
**Thông Điệp**  
**Thời Điểm**  
**Kênh**

Hình 1. 2 Có 4 Yếu Tố Cần Phải Nhớ



Hình 1. 3 Phiếu Bán Hàng

## KHẢ NĂNG NHẮM CHỌN CỦA CÁC KÊNH

### Quảng cáo Facebook

- Nhân khẩu học
- Sở thích, hành vi
- Khu vực, địa điểm
- Data sẵn có
- Người vào website
- Người xem videos
- Người tương tác
- Đối tượng tương tự

### Google / Cốc Cốc Search

- Từ khóa chung + địa điểm
- Từ khóa thương hiệu
- Từ khóa đối thủ

### Google Display Network

- Nhân khẩu học
- Từ khóa nội dung
- Chủ đề website
- Khu vực, địa điểm
- Data sẵn có
- Người vào website

### Apps Bất Động Sản

- Quảng Cáo Facebook
- Google/Cốc Cốc Search
- Google Display Network
- Local Ad Network

### Local Ad Network

- Nhân khẩu học
- Từ khóa nội dung
- Chủ đề website
- Khu vực, địa điểm
- Người vào website

Hình 1. 4 Cải thiện Phiếu Bán hàng Mở Rộng

Tăng chi phí quảng cáo và mở rộng đầu vào của phiếu, qua đó tăng số lượng bán hàng cuối phiếu.

Ưu điểm:

Nhanh chóng tăng được số lượng thông tin khách hàng

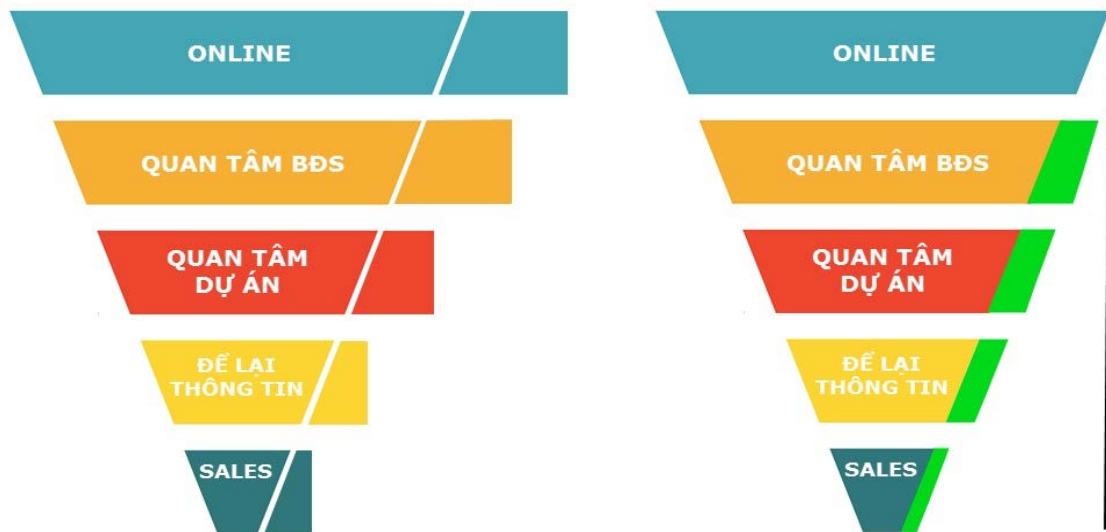
- Tiếp cận được các khách hàng mới

Yếu điểm:

- Chi phí tăng cao

- Tăng số lượng thì giảm chất lượng và hiệu quả

Hình 1. 5 Cải thiện Phiếu Bán hàng Tối Ưu Hoá



Hình 1. 6 Kết hợp mở rộng và tối ưu hoá



1. Google Adwords
2. Google Suggest
3. Keyword Planner
4. Google Analytics
5. Keywordtool.io
6. Ubersuggest.io
7. Similarweb
8. Sociograph

Hình 1. 7 Bảng thống kê seo của google theo các keyword



Hình 1. 8 Thời Điểm Các Giai Đoạn Của Khách Hàng

	HÀNH VI			
NHU CẦU	Rất Quan Tâm	Hơi Quan Tâm	Chưa Rõ	Không Hoạt Động
Đúng Nhu Cầu	Rất Quan Tâm Đúng Nhu Cầu	Hơi Quan Tâm Đúng Nhu Cầu	Chưa Rõ Đúng Nhu Cầu	Không Hoạt Động Đúng Nhu Cầu
Tiềm Năng	Rất Quan Tâm Tiềm Năng	Hơi Quan Tâm Tiềm Năng	Chưa Rõ Tiềm Năng	Không Hoạt Động Tiềm Năng
Chưa Đủ Thông Tin	Rất Quan Tâm Chưa Đủ Thông Tin	Hơi Quan Tâm Chưa Đủ Thông Tin	Chưa Rõ Chưa Đủ Thông Tin	Không Hoạt Động Chưa Đủ Thông Tin
Không Có Thông Tin	Rất Quan Tâm Không Có Thông Tin	Hơi Quan Tâm Không Có Thông Tin	Chưa Rõ Không Có Thông Tin	Không Hoạt Động Không Có Thông Tin
	Sales Team - Bán Hàng			
	Telesales - Tư Vấn Thêm Thông Tin			
	Marketing / Nuôi Dưỡng			

Hình 1. 9 Xác định nhu cầu của khách hàng

●	Ad group	Status ?	Default Max. CPC ?	Clicks ?	Impr. ? ↓	CTR ?	Avg. CPC ?	Cost ?	Avg. Pos. ?
●	Đăng Tin Bán Nhà	Eligible	₫2,300 (enhanced) <input checked="" type="checkbox"/>	894	3,192	28.01%	₫3,381	₫3,023,049	1.4
●	Đăng Tin Bất Động Sản	Eligible	₫2,300 (enhanced) <input checked="" type="checkbox"/>	480	2,400	20.00%	₫2,106	₫1,011,083	1.6
●	Đăng Tin Cho Thuê Nhà	Eligible	₫2,300 (enhanced) <input checked="" type="checkbox"/>	377	1,240	30.40%	₫2,594	₫978,004	1.9
●	Đăng Tin Bán Nhà Miễn Phí	Eligible	₫2,400 (enhanced) <input checked="" type="checkbox"/>	234	755	30.99%	₫2,169	₫507,592	1.7
●	Đăng Tin Nhà Đất	Eligible	₫2,300 (enhanced) <input checked="" type="checkbox"/>	132	570	23.16%	₫1,629	₫215,046	2.3
●	Đăng Tin Thuê Nhà	Eligible	₫3,500 (enhanced) <input checked="" type="checkbox"/>	120	449	26.73%	₫2,215	₫265,786	1.9

Campaign	Delivery	Results	Reach	Cost
<input checked="" type="checkbox"/> mik website visitors ho chi minh	Active	9,643 Link Clicks	51,671	\$0.02 Per Link Click
<input checked="" type="checkbox"/> mik engaged fanpage ho chi minh	Active	5,887 Link Clicks	32,843	\$0.02 Per Link Click
<input checked="" type="checkbox"/> cat mortgage loans ho chi minh	Active	5,282 Link Clicks	30,277	\$0.03 Per Link Click
<input checked="" type="checkbox"/> mik engaged fanpage hanoi	Active	5,168 Link Clicks	18,453	\$0.02 Per Link Click
<input checked="" type="checkbox"/> cat real estate ho chi minh	Active	4,449 Link Clicks	24,060	\$0.02 Per Link Click
<input checked="" type="checkbox"/> tk website visitors ho chi minh	Active	2,279 Link Clicks	25,480	\$0.02 Per Link Click

Hình 1. 10 Tối Ưu Hoá Quản Cáo

**Bất Động Sản Bình Dương ở** **Bất Động Sản Bình Dương.** 5 giờ · Thủ Dầu Một ·

**Cần bán nhà xưởng mặt tiền quốc lộ 13**

- Tổng diện tích 8500m2, có 600m2 thổ cư, nhà xưởng rộng 4000m2.
- Đất hai mặt tiền rất đẹp.
- Kho bãi, tiện nghi đầy đủ, có trạm cân và điện 3 pha.
- Hệ thống camera an ninh.
- Giá bán: 56 tỷ
- ☎☎☎0917.789.114 Trung trong trắng "trình trử".



Hình 1. 11 Mô hình quản cáo cũ

## CHƯƠNG 2: Cơ sở lý thuyết

### 2.1 Delving với máy ảo dalvik

Dalvik là máy ảo giúp các ứng dụng Java chạy được trên các thiết bị di động Android. Nó chạy các ứng dụng đã được chuyển đổi thành một file thực thi Dalvik (dex). Định dạng phù hợp cho các hệ thống mà thường bị hạn chế về bộ



nhớ và tốc độ xử lý. Dalvik đã được thiết kế và viết bởi Dan Bornstein, người đã đặt tên cho nó sau khi đến thăm một ngôi làng đánh cá nhỏ có tên là Dalvík ở đảo Eyjafjörður, nơi mà tổ tiên của ông sinh sống.

Từ góc nhìn của một nhà phát triển thì Dalvik trông giống như máy ảo java (Java Virtual Machine) nhưng thực tế thì hoàn toàn khác. Khi nhà phát triển viết một ứng dụng dành cho Android, anh ta thực hiện các đoạn mã trong môi trường Java. Sau đó, nó sẽ được biên dịch sang các byteCode của Java, tuy nhiên để thực thi được ứng dụng này trên Android thì nhà phát triển phải thực thi một công cụ có tên là dx. Đây là công cụ dùng để chuyển đổi byteCode sang một dạng gọi là dex bytecode. Dex là từ viết tắt của “Dalvik executable” đóng vai trò như cơ chế thực thi các ứng dụng Java.

## 2.2. Kiến trúc hệ điều hành Android

Mô hình sau thể hiện một cách tổng quát các thành phần của hệ điều hành Android. Mỗi phần sẽ được đặc tả một cách chi tiết dưới đây.



Hình 2. 1 Cấu trúc Stack hệ thống Android

## 2.3. Tầng ứng dụng (applications)

Android được tích hợp sẵn một số ứng dụng cần thiết cơ bản như: Contacts, browser, camera, phone... Tất cả ứng dụng chạy trên hệ điều hành Android đều được viết bằng Java.

## 2.4. Application Framework

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa. Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế đơn giản hóa việc sử dụng lại các thành phần, bất kỳ ứng dụng có thể xuất bản khả năng của mình và ứng dụng nào khác sau đó có thể sử dụng những khả năng( có thể hạn chế được bảo mật thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cở bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như : gridView, tableView, linearLayout, ...
- Một “Content Provider” cho phép tất cả ứng dụng truy xuất dữ liệu từ các ứng dụng khác ( chẳng hạn như Contact) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
- Một “Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized string, graphics, and layout file.
- Một “Notifycation Manager” cho tất cả các ứng dụng hiển thị các custom alerts trong status bar.
- Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

## 2.5. Library

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- System C library: a BSD-derived implementation of standard C system library(libc), turned for embedded Linux-based devices.
  - Media Libraries – based on PacketVideo’s OpenCORE, the libraries support playback and recording of many popular audio and video formats, as well as static image file, including: MPEG4, H, 264, MP3, AAC, AMG, JPG, and PNG
  - Surface Manager – Quản lý việc truy xuất vào hệ thống hiển thị.
  - LibWebCore – a modern web browser engine which powers both the Android browsers and an embadable web view.
  - SGL – the underlying 2D graphics engine.
  - 3D Libraries – an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration ( where available) or the included, highly optimized 3D software rasterizer.
  - FreeType – bitmap and vector font rendering.
- SQLite – a powerful and lightweight relational database engine available to all applications.

## 2.6. Android runtime

Android bao gồm tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết để cho một thiết bị có thể chạy nhiều máy ảo hiệu quả. Các VM Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. VM là dựa trên register – base, và chạy các lớp đã được biên dịch bởi một trình biên dịch Java để chuyển đổi thành các định dạng dex. Các VM Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ thấp.

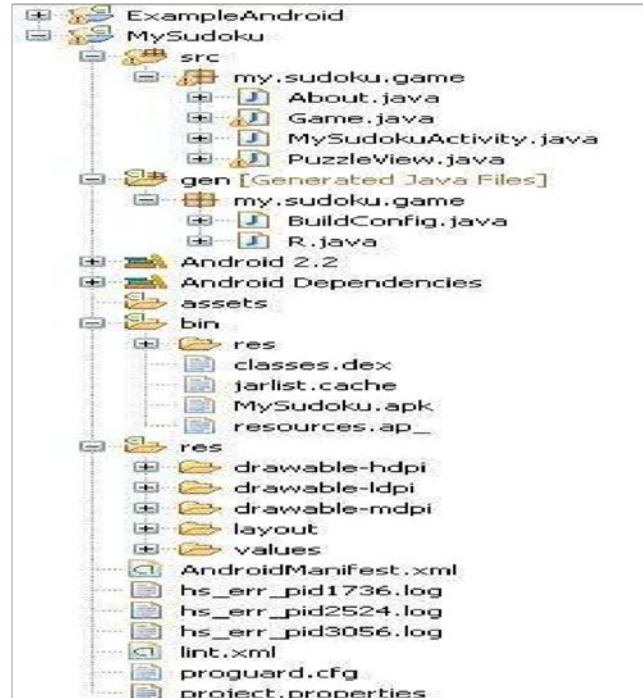
## 2.7. Linux kernel

Android dựa trên Linux phiên bản 2.6 cho hệ thống dịch vụ cốt lõi như security, memory management, process management, network stack, and driver model. Kernel Linux hoạt động như một lớp trừu tượng hóa giữa phần cứng và phần còn lại của phần mềm stack.

## 2.8. Các thành phần trong ứng dụng Android

Android project là một hệ thống thư mục file chứa toàn bộ source code, tài nguyên... mà mục đích cuối cùng là để đóng gói thành một file .apk duy nhất.

Trong một thư mục project, có một số thành phần (file, thư mục con) được tạo ra mặc định, còn lại phần lớn sẽ được tạo ra sau nếu cần trong phát triển ứng dụng.



Hình 2. 2 Các thành phần trong một Android project

- Src: Chứa toàn bộ source code (file .java hoặc .aidl)
- Bin: Thư mục chứa file Output sau khi build. Đây là nơi bạn có thể tìm thấy file .apk



- **Gen:** Chứa file .java tạo ra bởi ADT plug-in, như là file R.java hoặc các giao diện tạo ra từ file AIDL.
- **Res:** Chứa các tài nguyên (resource) cho ứng dụng chẳng hạn như file hình ảnh, file layout, các chuỗi (string)...Dưới đây là các thư mục con của nó.
- **Anim:** Chứa các file .xml dùng cho việc thiết lập các hiệu ứng động(animation).
- **Color:** Chứa các file .xml dùng định nghĩa màu sắc.
- **Drawable:** Chứa hình ảnh (png, jpeg, gif), file .xml định nghĩa cách vẽ các loại hình dạng khác nhau (shape).
- **Layout:** Chứa file .xml dùng để dựng giao diện người dùng.
- **Menu:** Chứa file .xml quy định application menu.
- **Raw:** Chứa các file media, chẳng hạn như .mp3, .ogg
- **Values:** Chứa file .xml định nghĩa các giá trị. Khác với các resource trong thư mục khác, resource ở thư mục này khi định danh trong lớp R thì sẽ không sử dụng file name để định danh mà sẽ được định danh theo quy định bên trong file .xml đó.
- **Xml:** Dùng chứa các file .xml linh tinh khác, chẳng hạn như file .xml quy định app widget, search metadata, ...
- **Libs:** Chứa các thư viện riêng.
- **AndroidManifest.xml:** File kiểm soát các thành phần trong ứng dụng như: activity, service, intent, receiver... tương tác với nhau, cách ứng dụng tương tác với ứng dụng khác, cũng như đăng kí các quyền hạn về sử dụng tài nguyên trong máy.
- **Build.properties:** Tùy chỉnh các thiết lập cho hệ thống build, nếu bạn sử dụng Eclipse thì file này không cần thiết.
- **Build.xml:** Chỉ sử dụng khi dùng dòng lệnh để kiến tạo project.
- **Default.properties:** File này chứa các thiết lập cho project, chẳng hạn như build target, min SDK version...(tốt hơn hết là không nên chỉnh sửa file này bằng tay)
- **File AndroidManifest.xml**

Là nền tảng của mọi ứng dụng Android, file AndroidManifest.xml được đặt trong thư mục root và cho biết những thành phần có trong ứng dụng của: các activities, các services, ...cũng như cách các thành phần ấy gắn bó với nhau.

Mỗi file manifest đều bắt đầu với một thẻ manifest:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.commonware.android.search"
    android:versionCode="1"
    android:versionName="1.0">
    ...
</manifest>
```

Khi tạo file manifest, điều quan trọng nhất là phải cung cấp thuộc tính của package, tức tên của Java package dùng làm cơ sở cho ứng dụng của ta. Sau khi đã đặt tên package, sau này khi cần ghi tên lớp trong file manifest, ta có thể rút ngắn, chẳng hạn với lớp "com.yourapp.android.search.Someclass" ta chỉ cần ghi ".Someclass" là đủ.

Các thành phần manifest khác là :

- **uses-permission:** chỉ định các quyền mà ứng dụng của ta được cấp để hoạt động trôi chảy (như đã nói, các ứng dụng Android nằm dưới nhiều lớp bảo mật khác nhau).
- **permission:** chỉ định các quyền mà các activities hay services yêu cầu các ứng dụng khác phải có mới được truy cập dữ liệu của ứng dụng của ta.

- instrumentation: chỉ định phần code cần được gọi khi xảy ra những sự kiện quan trọng (chẳng hạn khởi động activities) nhằm phục vụ việc ghi chú (logging) và tra soát (monitoring)
  - uses-library: nhằm kết nối với các thành phần có sẵn của Android (như service tra bản đồ, ...)
  - uses-sdk: có thể có hoặc không, chỉ ra phiên bản của Android mà ứng dụng này yêu cầu.
- application: định nghĩa phần trung tâm của ứng dụng của file manifest.

Ví dụ file AndroidManifest:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.commonware.android.search"
    android:versionCode="1"
    android:versionName="1.0">
    <usepermission
        android:name="android.permission.ACCESS_LOCATION"/>
    <usepermission
        android:name="android.permission.ACCESS_GPS"/>
    <usepermission
        android:name="android.permission.ACCESS_ASSISTED_GPS"/>
    <usepermission
        android:name="android.permission.ACCESS_CELL_ID"/>
    <application>
        ...
    </application>
</manifest>
```

Tất nhiên, phần quan trọng của 1 file manifest chính là thành phần application. Mặc định, khi ta tạo 1 project Android mới, ta có sẵn 1 thành phần activity:

```
<?xmlversion="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="vi.du" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8"/>
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name">
        <activity
            android:name=".Android1Activity" android:label="@string/app_name">
            <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

- Thành phần này cung cấp các thông tin sau
  - android: name : tên class hiện thực activity này.
  - android: label : tên activity.

intent-filter : Một thành phần con, chỉ ra dưới điều kiện nào thì activity này được hiển thị.

- Các thành phần cơ bản của ứng dụng Android Activity:

Là thành phần tối quan trọng của bất kỳ một ứng dụng Android nào. Thuật ngữ Activity chỉ một việc mà người dùng có thể thực hiện trong một ứng dụng Android. Do gần như mọi activity đều tương tác với người dùng, lớp Activity đảm nhận việc tạo ra một cửa sổ (window) để người lập trình đặt lên đó một giao diện UI với setContentView(View). Một activity có thể mang nhiều dạng khác nhau: Một cửa sổ toàn màn hình (full screen window), một cửa sổ floating (với windowsIsFloating) hay nằm lồng bên trong 1 activity khác (với ActivityGroup).

- Có 2 phương thức mà gần như mọi lớp con của Activity đều phải hiện thực:
    - onCreate(Bundle) - Nơi khởi tạo activity. Quan trọng hơn, đây chính người lập trình gọi setContentView(int) kèm theo layout để thể hiện UI của riêng mình. Đồng thời còn có findViewById(int) giúp gọi các widget (buttons, text boxes, labels, ..) để dùng trong UI.
    - onPause() - Nơi giải quyết sự kiện người dùng rời khỏi activity. Mọi dữ liệu được người dùng tạo ra tới thời điểm này cần phải được lưu vào ContentProvider.
- Để có thể sử dụng Context.startActivity(), mọi lớp activity đều phải được khai báo với tag <activity> trong file AndroidManifest.xml.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.my_domain.app.helloactivity">
    <application android:label="@string/app_name">
        <activity android:name=".HelloActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Hình 2. 3 Khai báo tag <activity> trong file AndroidManifest.xml

### Vòng đời của Activity

Các activity được quản lý dưới dạng các activity stack - First-In-Last-Out: Khi một activity mới được khởi tạo, nó sẽ được đưa lên trên cùng stack, các activity khác muốn chạy trên nền (foreground) trở lại thì cần phải chờ tới khi Activity mới này kết thúc.

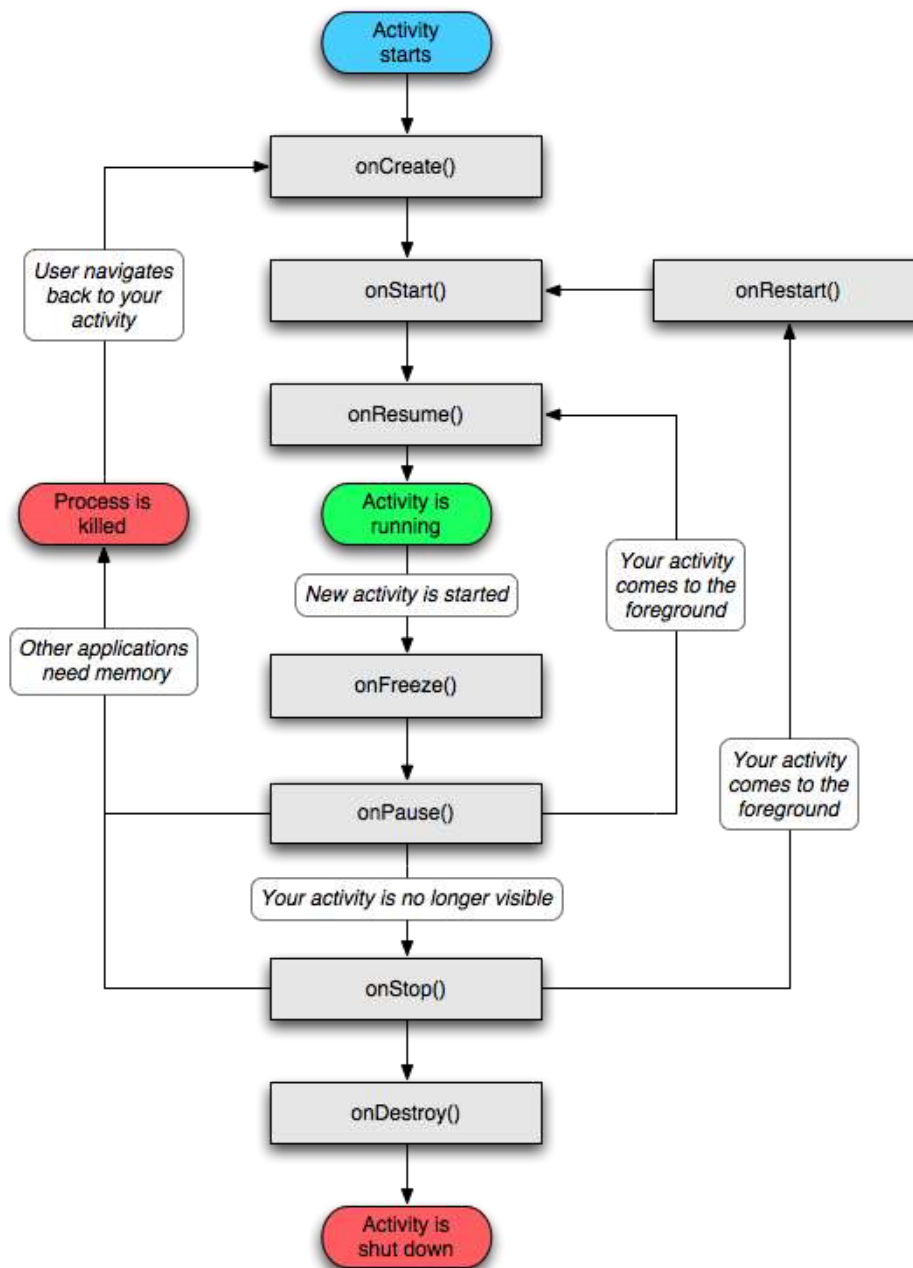
### Một Activity có 4 trạng thái:

Active hay Running: Khi một activity đang chạy trên màn hình.

**Paused:** Khi một activity vẫn đang chạy trên màn hình nhưng đang bị một activity trong suốt (transparent) hay không chiếm toàn màn hình hiển thị phía trên. Tuy vẫn lưu trữ dữ liệu, nhưng các paused activity này sẽ bị hệ thống bắt chấm dứt khi đang thiếu bộ nhớ trầm trọng.

**Stopped:** Khi 1 activity bị che khuất hoàn toàn bởi 1 activity khác. Tuy vẫn lưu trữ dữ liệu, nhưng các stopped activity này sẽ thường xuyên bị hệ thống bắt chấm dứt để dành chỗ cho các tiến trình khác.

**Killed hay Shut down:** Khi 1 activity đang paused hay stopped, hệ thống sẽ xóa activity ấy ra khỏi bộ nhớ.



Hình 2. 4 Lược đồ vòng đời của 1 Activity

Dựa vào lược đồ trên, thấy được có 3 vòng lặp quan trọng sau:

Vòng đời toàn diện (Entire Lifetime): Diễn ra từ lần gọi onCreate(Bundle) đầu tiên và kéo dài tới lần gọi onDestroy() cuối cùng.

Vòng đời thấy được (Visible Lifetime): Diễn ra từ khi gọi onStart() và kéo dài tới khi gọi onStop(). Ở vòng đời này, activity được hiển thị trên màn hình nhưng có thể không tương tác với người dùng ở trên nền. Các phương thức onStart() và onStop() có thể được gọi nhiều lần.

Vòng đời trên nền (Foreground Lifetime): Diễn ra từ khi gọi onResume() và kéo dài tới khi gọi onPause(). Ở vòng đời này, activity nằm trên mọi activity khác và tương tác được với người dùng. 1 activity có thể liên tục thay đổi giữa 2 trạng thái paused và resumed, chẳng hạn khi thiết bị sleep hay 1 intent mới được đưa tới.

Toàn bộ vòng đời của 1 activity được định nghĩa nhờ các phương thức sau:

```
public class Activity extends ApplicationContext
{
    protected void onCreate(Bundle savedInstanceState);
```

Gọi khi mới tạo activity để setup các view, binding dữ liệu, ...  
Kèm theo sau luôn là onStart().

```
    protected void onRestart();
```

Gọi sau khi activity bị stopped và trước khi được khởi động lại.  
Kèm theo sau luôn là onStart().

```
    protected void onStart();
```

Gọi khi activity hiện lên trước mắt người dùng.  
Kèm theo sau là onResume() nếu activity hiện lên nền hay onStop() nếu bị ẩn đi.

```
    protected void onResume();
```

Gọi khi activity bắt đầu tương tác với người dùng và đang trên cùng của activity stack.  
Kèm theo sau luôn là onPause().

```
    protected void onPause(); //Gọi khi hệ thống sắp khởi động lại 1 activity khác trước đó.
```

Kèm theo sau là onResume nếu activity trở lại trên cùng hay onStop() nếu bị ẩn đi.

```
    protected void onStop();
```

Gọi khi activity không còn hiển thị trước người dùng  
Kèm theo sau là onRestart() nếu activity hiện lên trở lại hay onDestroy nếu sắp xóa activity đi.

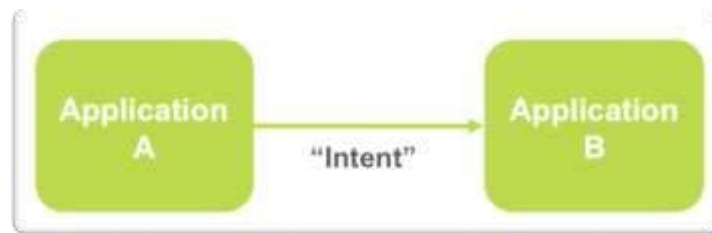
```
    protected void onDestroy();
```

Gọi ngay trước khi kết thúc activity, xảy ra khi hàm finish() được gọi hoặc khi hệ thống yêu cầu buộc phải kết thúc.

}

➤ Intent:

Intent được sử dụng với phương thức `startActivity()` để mở một Activity, và dùng với `broadcastIntent` để gửi nó đến bất kỳ `BroadcastReceiver` liên quan nào, và dùng với `startService(Intent)`, `_bindService(Intent, ServiceConnection, int)` để giao tiếp với các Service chạy dưới nền. Intent cung cấp một chức năng cho phép kết nối hai chương trình khác nhau trong quá trình thực thi (runtime) (Cung cấp khả năng cho phép hai chương trình khác nhau giao tiếp với nhau). Chức năng quan trọng và được sử dụng nhiều nhất của một Intent là mở một Activity, nơi mà nó có thể được dùng như một vật kết nối các Activity lại với nhau



Hình 2. 5 Truyền thông tin giữa hai Activity khác nhau

Sử dụng intent để trao đổi thông tin giữa hai chương trình

Thành phần chính của Intent bao gồm:

Action: Xác định hành động sẽ được thực thi, các hành động này có thể là:

`ACTION_VIEW`, `ACTION_EDIT`, `ACTION_MAIN`...

➤ Data: Các dữ liệu được sử dụng để hành động (Action) thao tác trên nó, như bản ghi về một người nào đó trong cơ sở dữ liệu chứa các contact của thiết bị.

• Ví dụ về thành phần action/data:

`ACTION_VIEW` content: //contacts/people/1 – hiển thị thông tin liên lạc của người có mã là “1”.

`ACTION_DIAL` content: //contacts/people/1 – gọi điện cho người có mã là “1”.

`ACTION_DIAL` tel: 123 – gọi đến số điện thoại “123”

Ngoài ra, Intent còn có các thuộc tính phụ sau:

Category: thông tin chi tiết về hành động được thực thi, ví dụ như

`CATEGORY_LAUNCHER` có nghĩa là nó sẽ xuất hiện trong Launcher như ứng dụng có mức level cao (top-level application), trong khi `CATEGORY_ALTERNATIVE` chứa thông tin danh sách các hành động thay thế mà người dùng có thể thực thi trên phần dữ liệu mà Intent cung cấp.

Type: Chỉ định 1 kiểu dữ liệu chính xác (kiểu MIME) được mang bởi intent. Thường thì type được suy ra từ chính dữ liệu. Bằng cách thiết lập thuộc tính type, bạn có thể vô hiệu hóa sự phỏng đoán kiểu dữ liệu và chỉ định rõ một kiểu chính xác.

Component: Chỉ định rõ tên của lớp thành phần (Một thể hiện của một class java dùng để thực thi các hành động được đặc tả trong Intent) sử dụng cho Intent. Thông thường thì nó được xác định bằng cách tìm kiếm thông tin trong Intent (Các thông tin như Action, data/type, và category) và nối chúng với các component (Một thể hiện của một

Class java dùng để thực thi các hành động được đặc tả trong Intent) có thể đáp ứng được các yêu cầu sử lý của Intent.

Extracts: là một đối tượng Bundle dùng để chứa các thông tin kèm theo được dùng để cung cấp thông tin cần thiết cho component. Ví dụ: Nếu có một Intent đặc tả hành động send email thì những thông tin cần chứa trong đối tượng Bundle là subject, body... Inten Resolution

- Intent có 2 dạng chính sau:
  - Explicit Intents: Xác định rõ một component (Một thể hiện của một class java dùng để thực thi các hành động được đặc tả trong Intent) (thông qua phương thức setComponent(ComponentName) hoặc setClass(Context, Class)) cung cấp lớp sẽ thực thi các hành động được đặc tả trong Intent. Thông thường thì những Intent này không chứa bất kỳ thông tin nào khác (như category, type) mà đơn giản chỉ là cách để ứng dụng mở các Activity khác bên trong một Activity.
  - Implicit Intents: Không chỉ định một component nào cả, thay vào đó, chúng sẽ chứa đủ thông tin để hệ thống có thể xác định component có sẵn nào là tốt nhất để thực thi hiệu quả cho Intent đó. Khi sử dụng Implicit intents, do tính chất chuyên quyền của loại Intent này, ta cần phải biết phải làm gì với nó. Công việc này được đảm nhiệm bởi tiến trình của Intent resolution, nó giúp chỉ định Intent đến một Activity,
  - BroadcastReceiver, hoặc Service (hoặc thỉnh thoảng có thể là 2 hay nhiều hơn một activity/receiver) để có thể xử lý các hành động được đặc tả trong Intent. Bất cứ thành phần nào (Activity, BroadcastReceiver, Service) khi muốn sử dụng trong ứng dụng đều phải được đăng kí trong file AndroidManifest.xml. Trong đó cần định nghĩa một thẻ <intent-filter> cung cấp các thông tin để hệ thống có thể xác định được cái mà các component này (Activity, BroadcastReceiver, Service) có thể xử lý được (những action mà component này có thể thực hiện được).
  - Intent Filter là bản đặc tả có cấu trúc của các giá trị của Intent dùng để xác định component phù hợp để thực hiện các hành động được đặc tả trong Intent. Một Intent Filter nói các hành động, categories và dữ liệu (hoặc thông qua type hoặc scheme) trong Intent. Intent Filter được khai báo trong AndroidManifest.xml và sử dụng thẻ intent-filter.
- Một Intent Filter có các thành phần chính sau:
  - Action: Tên hành động mà component có thể thực thi.
  - Type: Kiểu hành động mà component có thể thực thi.
  - Category: Phân nhóm các hành động.

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

- Luật xác định component phù hợp với intent

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

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

Nếu có component phù hợp Android sẽ mở component đó lên để thực thi các hành động theo yêu cầu. Nếu có nhiều hơn 1 component có action-name phù hợp thì Android sẽ yêu cầu người dùng chọn component phù hợp.

- Ngược lại nếu không có component nào phù hợp Android sẽ tiến hành xem xét kiểu dữ liệu của Intent cung cấp xem có component nào có đủ năng lực để xử lý kiểu dữ liệu đó không. Nếu không được Android sẽ tiến hành xem xét scheme của dữ liệu đó để tìm kiếm component phù hợp. Nếu vẫn không tìm được component phù hợp Android sẽ tiến hành xem xét các component có chung.
  - Category với Intent để xác định component.
    - Service : Một service là một thành phần của ứng dụng, thể hiện mong muốn ứng dụng thực hiện các hành động trong khi không tương tác với người dùng hoặc cung cấp chức năng cho các ứng dụng khác sử dụng. Nói một cách đơn giản, service là các tác vụ (task) chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó. Mỗi class Service phải chứa thẻ service được khai báo trong file AndroidManifest.xml. Services có thể được bắt đầu bởi.
      - Context.startService() và Context.bindService()
- Cũng như các đối tượng của ứng dụng khác, services chạy trên luồng (Thread) của tiến trình chính. Có nghĩa là nếu service của bạn dự định chạy các hành vi có cường độ lớn hoặc các hành vi có thể gây nghẽn mạch, nó sẽ tự sinh ra luồng (Thread) của chính nó để làm thực hiện các tác vụ được đặc tả trong Service.
- Cần lưu ý:

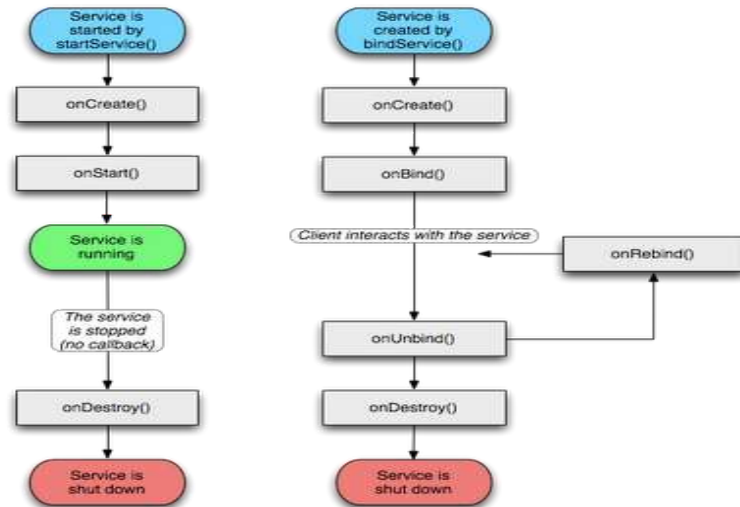
Service không phải là một tiến trình độc lập. Đối tượng Service không chạy trên tiến trình (process) của riêng nó (trừ khi có một yêu cầu đặt biệt), Service chạy trên tiến trình của chương trình.

Service không phải là một luồng (thread).

- Service có chứa 2 đặc trưng cơ bản sau:
  - Là một chức năng để ứng dụng thông báo với hệ thống về việc mà nó muốn thực hiện ở phía dưới nền (ngay cả khi người dùng không tương tác trực tiếp tới ứng dụng). Điều này tương ứng với việc gọi phương thức Context.startService(), nó sẽ yêu cầu hệ thống lên lịch cho service để chạy cho đến khi bị chính service hay người nào đó dừng nó lại. Là một chức năng cho phép ứng dụng để cung cấp các chức năng của nó cho các ứng dụng khác. Điều đó tương ứng với việc gọi Context.bindService(), cho phép một long-standing connection được tạo ra để service có thể tương tác với nó.
  - Khi một Service được tạo ra, việc hệ thống cần làm là tạo ra thành phần và gọi hàm onCreate() và bất kỳ lệnh callback thích hợp nào trên luồng chính. Nó phụ thuộc vào Service thì hành hành vi thích hợp nào, chẳng hạn tạo luồng thứ hai để thực hiện tiến trình. Vòng đời của một service
- Có 2 lý do để một service được chạy bởi hệ thống. Bằng cách gọi phương thức
  - Context.startService() và Context.bindService(). Nếu người dùng gọi phương thức Context.startService() thì hệ thống sẽ nhận về một đối tượng service (tạo mới nó và gọi hàm onCreate() nếu cần thiết) và sau đó lại gọi phương thức onStartCommand(Intent, int, int) với các biến được cung cấp bởi client. Service mới được tạo ra này sẽ chạy cho tới khi có lệnh gọi phương thức



➤ Context.stopService() hoặc stopSelf(). Dù cho một service có được start bao nhiêu lần đi nữa thì nó sẽ bị tắt ngay khi phương thức Context.stopService() hoặc phương thức stopSelf() được gọi. Service có thể dùng phương thức stopSelf(int) để đảm bảo rằng service sẽ không được tắt cho đến khi Intent đã khởi tạo nó được xử lý.



Hình 2. 6 Lược đồ vòng đời của một service

Dựa theo hình trên, 1 service được bắt đầu bởi phương thức startService() là service dùng để thực hiện các tác vụ ngầm dưới nền, service được bắt đầu bởi phương thức bindService() là service được dùng để cung cấp các chức năng cho các chương trình khác.

### ➤ BroadcastReceiver

Là lớp nhận về các intents được gửi bởi sendBroadcast(). Có thể tạo instance cho lớp này bằng 2 cách: hoặc với Context.registerReceiver() hay thông qua tag <receiver> trong file AndroidManifest.xml.

- Có hai lớp broadcast quan trọng cần xét đến:
  - Broadcasts bình thường (Normal broadcasts): Được gửi bởi context.sendBroadcast tới các receivers hoàn toàn không theo 1 trật tự nào, dẫn tới đôi khi tồn tại cùng 1 lúc nhiều receivers khác nhau. Tuy nhanh nhưng không thể trao đổi kết quả nhận về giữa các receivers với nhau.
  - Broadcasts có thứ tự (Ordered broadcasts) - Được gửi bởi Context.sendOrderedBroadcast() tới từng receiver một. Như vậy, kết quả nhận về của 1 receiver có thể trở thành input của 1 receiver khác, hay có thể ngay lập tức dừng việc broadcast tới các receiver khác. Trật tự chạy các receivers được quyết định bởi thuộc tính android: priority trong intent-filter tương ứng.

### ➤ ContentProvider:

Là nơi lưu trữ và cung cấp cách truy cập dữ liệu do các ứng dụng tạo nên. Đây là cách duy nhất mà các ứng dụng có thể chia sẻ dữ liệu của nhau.

Android cung cấp sẵn content providers cho 1 số kiểu dữ liệu thông dụng như âm thanh, video, thông tin số điện thoại, ...(Xem thêm trong gói android.provider). Người lập trình cũng có thể tự tạo ra các class con

(subclass) của Content Provider để lưu trữ kiểu dữ liệu của riêng mình.

- Cách thức hoạt động của Content Provider:

Mọi content provider, tuy có thể khác nhau về cách thức lưu trữ dữ liệu, đều sử dụng chung 1 interface để tìm và trả về dữ liệu, bao gồm việc thêm, bớt và sửa dữ liệu. Việc này được thực hiện thông qua các đối tượng

ContentResolver khi gọi getContentResolver()

ContentResolver cr = getContentResolver();

- Mô hình dữ liệu

Content providers sử dụng dữ liệu dưới dạng 1 table đơn giản, trong đó mỗi hàng là 1 record và mỗi cột là dữ liệu của 1 kiểu nhất định.

_ID	NUMBER	NUMBER_KEY	LABEL	NAME	TYPE
13	(425) 555 6677	425 555 6677	Kirkland office	Bully Pulpit	TYPE_WORK
44	(212) 555-1234	212 555 1234	NY apartment	Alan Vain	TYPE_HOME
45	(212) 555-6657	212 555 6657	Downtown office	Alan Vain	TYPE_MOBILE
53	201.555.4433	201 555 4433	Love Nest	Rex Cars	TYPE_HOME

Hình 2. 7 Mô hình dữ liệu

- Ví dụ mô hình dữ liệu thông tin số điện thoại trong máy

Mỗi record có 1 trường “\_ID” độc nhất để liên kết với các bảng khác.

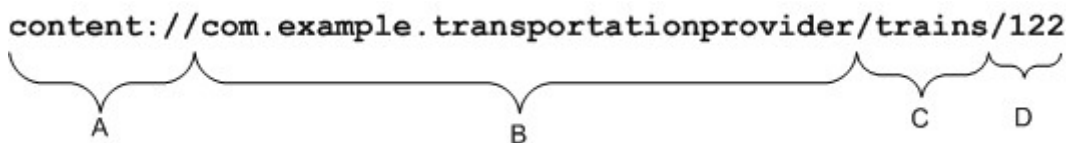
URI (Uniform Resource Identifier)

URI là dãy các ký tự string được dùng để:

Mở 1 trang web trong trình duyệt, bắt đầu với “http: //”

Chỉ tới các file tài nguyên có sẵn trong máy (VD: 1 file video, danh sách contacts, ...), bắt đầu với “content: //”

Dựa vào hình trên



Hình 2. 8 Các thành phần của một url

A Tiền tố mặc định của 1 URI, không thay đổi được.

B Xác định content provider. Đối với các ứng dụng hãng thứ 3, đây chính là tên đầy đủ của 1 lớp.

C Xác định loại dữ liệu nào cần lấy về. Nếu nguồn cung cấp chỉ cho lấy 1 loại duy nhất thì có thể để trống, hoặc nếu có nhiều loại thì có thể phân ra từng phần 1 (land/bus, land/train, sea/ship, ...)

D ID của 1 record, khi cần lấy về chỉ duy nhất 1 record.

Khi tạo 1 content provider, Android tự động tạo 1 constant cho URI của provider đó để đảm bảo code hiệu quả và dễ cập nhật sau này. Constant này - CONTENT\_URI - sẽ xác định cho ContentProvider phải liên kết với bảng nào của nguồn cung cấp.

- Ví dụ:

URI cho bảng liên kết số điện thoại với tên người là:  
android.provider.Contacts.Phones.CONTENT\_URI

URI cho bảng liên kết hình đại diện với tên người là:  
android.provider.Contacts.Photos.CONTENT\_URI

Tìm kiếm 1 Content Provider

Để bắt đầu tìm kiếm (query) 1 content provider, trước nhất người lập trình cần có trong tay các yếu tố sau:

URI xác định provider.

Tên các trường (field) dữ liệu sẽ lấy về.

Các kiểu dữ liệu của những trường trên.

ID, chỉ khi cần tìm 1 record xác định nào đó.

- Để bắt đầu tìm kiếm, người lập trình có thể dùng 1 trong 2 phương thức: ContentResolver.query() hoặc Activity.managedQuery(). Cả 2 phương thức trên đều có chung arguments và đều trả về 1 đối tượng con trỏ (Cursor) public final Cursor query (Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) public final Cursor managedQuery (Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)

- Đầu tiên, dù với query() hay managedQuery() thì ta đều cần cung cấp URI của bên cung cấp, tức CONTENT\_URI. Chú ý nếu muốn giới hạn chỉ tìm kiếm 1 URI duy nhất, ta thêm ID vào cuối URI như sau: "content: //.../23", với 23 là ID ta cần tìm kiếm

Hoặc ta có thể dùng các hàm hỗ trợ như ContentUris.withAppended() và Uri.withAppendedPath().

- Ví dụ:

Uri myPerson = ContentUris.withAppendedId(People.CONTENT\_URI, 23);

Các biến truyền vào khác của query() và managedQuery() là:

projection - Tên của cột dữ liệu cần trả về. Nếu NULL thì trả về tất cả các cột. selection - Tên các hàng dữ liệu sẽ trả về. Sử dụng với WHERE giống như khi truy vấn SQL. Nếu NULL thì trả về tất cả các hàng.

selectionArgs - quy định giá trị trả về,

- VD: selectionArgs = {value\_as\_string}

sẽ ép kiểu dữ liệu trả về dưới dạng string.

sortOrder - Xác định thứ tự sắp xếp các hàng. Sử dụng với ORDER BY giống như khi truy vấn SQL. Nếu NULL thì không theo thứ tự gì cả.

- Tạo mới 1 Content provider

Để tạo mới 1 content provider, cần phải thoả mãn các điều kiện:

- Thiết lập hệ thống lưu trữ dữ liệu như thế nào: bằng các file hay sử dụng cơ sở dữ liệu SQLite Extend lớp ContentProvider.

- Khai báo 1 content provider trong file AndroidManifest.xml
- Để định nghĩa 1 lớp con ContentProvider, ta cần hiện thực 6 phương thức trừu tượng (abstract) được khai báo trong lớp ContentProvider: query() insert() update() delete() getType() onCreate() Phương thức query() phải trả về 1 đối tượng Cursor. Mặc dù Cursor về cơ bản là 1 interface, Android cung cấp sẵn 1 số đối tượng Cursor tạo sẵn. Ví dụ; SQLiteCursor dành cho cơ sở dữ liệu SQLite, MatrixCursor cho dữ liệu không lưu trong SQLite.

- Android cũng khuyên nên gọi ContentResover.notifyChange() để báo cho các listeners biết dữ liệu sẽ được hiệu chỉnh. Ngoài ra người lập trình cũng nên thêm vài thao tác sau để giúp việc truy xuất các lớp hiệu quả hơn:

Định nghĩa một Uri dạng public static final, với tên là CONTENT\_URI (như đã trình bày ở trên).

- Định nghĩa tên các cột mà content provider sẽ trả về. Thông thường thì đây cũng chính là tên các cột trong cơ sở dữ liệu SQLite mà chúng đại diện. Tiếp theo, người lập trình nên định nghĩa các public static String constants để các ứng dụng khác có thể ghi rõ tên các cột và các hướng dẫn khác 1 cột interger tên "\_id" (với constant \_ID) cũng cần được định nghĩa để sử dụng khi cần lấy về các record độc nhất. Trong SQLite, trường \_ID nên có kiểu như sau: INTEGER PRIMARY KEY AUTOINCREMENT, trong đó thành phần AUTOINCREMENT không nhất thiết phải có, nhưng nên được đưa vào để đảm bảo các ID luôn độc nhất, tránh trường hợp khi xóa 1 row thì ID của row tiếp theo sẽ được cấp ID giống hệt của row vừa bị xóa.

- Nếu content provider này xử lý 1 kiểu dữ liệu hoàn toàn mới, người lập trình cần định nghĩa 1 kiểu MIME mới để trả về trong sự thể hiện của ContentProvider.getType().

Để khai báo 1 Content Provider mới, trước nhất cần phải khai báo trong tag <provider> của file AndroidManifest.xml.

```
<provider name="com.example.autos.AutoInfoProvider"
          authorities="com.example.autos.autoinfoprovider"
          . . . />
</provider>
```

Hình 2. 9 Khai báo tag <provider> trong file AndroidManifest.xml

Trong đó, thuộc tính name là tên đầy đủ của lớp con ContentProvider, và authorities là phần xác định provider của URI "content: "

View: View là thành phần cơ bản để xây dựng giao diện người dùng cho 1 ứng dụng Android. View là 1 lớp căn bản của widgets (widgets được dùng để tạo các nút nhấn, text fields, ...).

Lớp con ViewGroup là lớp căn bản của layouts, có thể coi như 1 cái hộp vô hình chứa nhiều Views hay ViewGroups khác và xác định các thuộc tính layout.

Sử dụng Views: Trong 1 cửa sổ màn hình, mọi views đều được sắp xếp theo thứ tự trên 1 cây trong 1 hay nhiều file XML nằm trong res/layout.

Khi tạo 1 cây chứa các views, người lập trình cần xác định những giá trị sau:

**Các thuộc tính** - Chẳng hạn định xem 1 TextView sẽ hiển thị chữ gì. Mỗi view có những thuộc tính và phương thức khác nhau.

**Mức độ tập trung (Focus)** - Thay đổi sự tập trung tùy theo input của người dùng với requestFocus().

**Listeners** - Định các listeners để có hành động khi 1 sự kiện gì đó xảy ra với onFocusChangeListener(View.OnFocusChangeListener)

Nhìn thấy hay không - với setVisibility(int)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
    />
</LinearLayout>
```

Hình 2. 10 Ví dụ một cây views

Một số Views thường dùng:

**TextView:** Hiển thị 1 khung text và cho phép người dùng thay đổi.

**ImageView:** Hiển thị 1 hình ảnh xác định từ file tài nguyên hay qua 1 content provider.

**Button:** Hiển thị 1 nút nhấn.

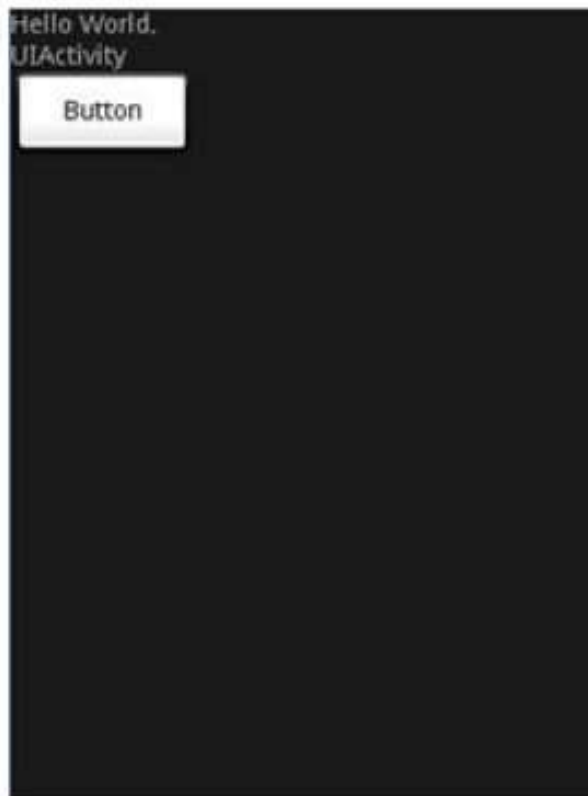
**Checkbox:** Hiển thị 1 nút nhấn với 2 trạng thái khác nhau.

**KeyboardView:** Hiển thị bàn phím ảo để nhập liệu.

**WebView:** Hiển thị các trang web bằng Webkit.

Một số ViewGroups thường dùng:

**LinearLayout:** Các view được xếp theo 1 hàng hay 1 cột duy nhất.



Hình 2. 11 LinearLayout



Hình 2. 12 AbsoluteLayout

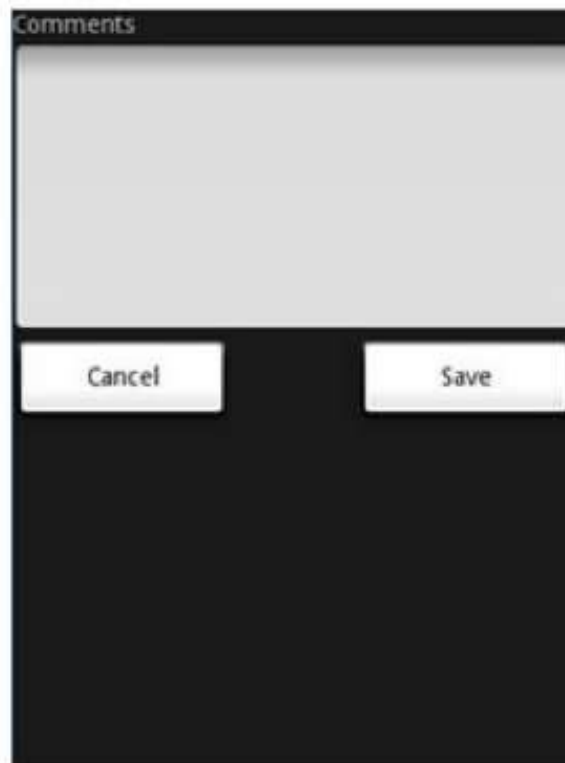
AbsoluteLayout: Cho phép xác định chính xác vị trí của từng view.

TableLayout: Sắp xếp các view theo các cột và hàng. Mỗi hàng có thể có nhiều view, mà mỗi view sẽ tạo nên 1 ô.



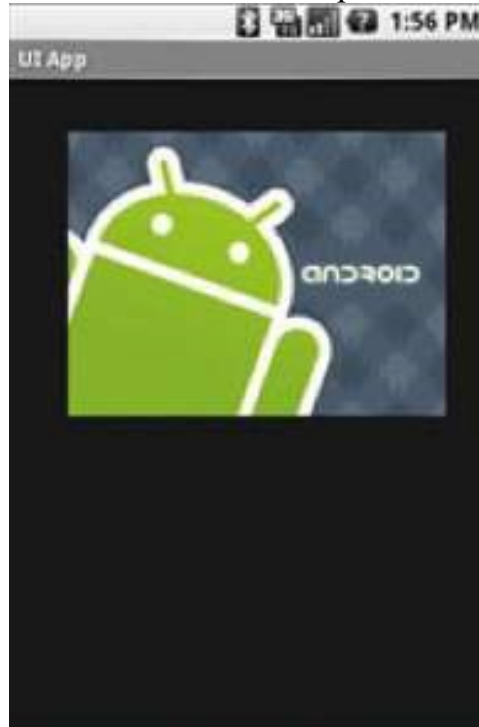
Hình 2. 13 TableLayout

RelativeLayout: Cho phép xác định vị trí các view theo mối quan hệ giữa chúng (VD; canh trái, phải, ...)



Hình 2. 14 RelativeLayout

FrameLayout: Là 1 placeholder cho phép đặt lên đó 1 view duy nhất. View đặt lên FrameLayout luôn được canh lề trái phía trên.



Hình 2. 15 FrameLayout

ScrollView: Là 1 FrameLayout đặc biệt cho phép trượt (scroll) 1 danh sách dài hơn kích thước màn hình.



Hình 2. 16 ScrollView

- Lưu trữ dữ liệu
  - Android có nhiều cách để lưu trữ dữ liệu. Một cách đơn giản nhất là sử dụng hệ thống preferences. Android cho phép các activity và ứng dụng lưu giữ preferences theo dạng cặp từ khóa và giá trị, nằm giữa các lần gọi một activity. Mục đích chính là để lưu trữ thông tin cấu hình chi tiết mà người dùng hiệu chỉnh, chẳng hạn trang web người dùng xem gần đây nhất. Bạn có thể lưu trữ bất kì thông tin nào trong



preferences bằng từ khóa có kiểu String và kiểu dữ liệu nguyên thủy (boolean, String...)

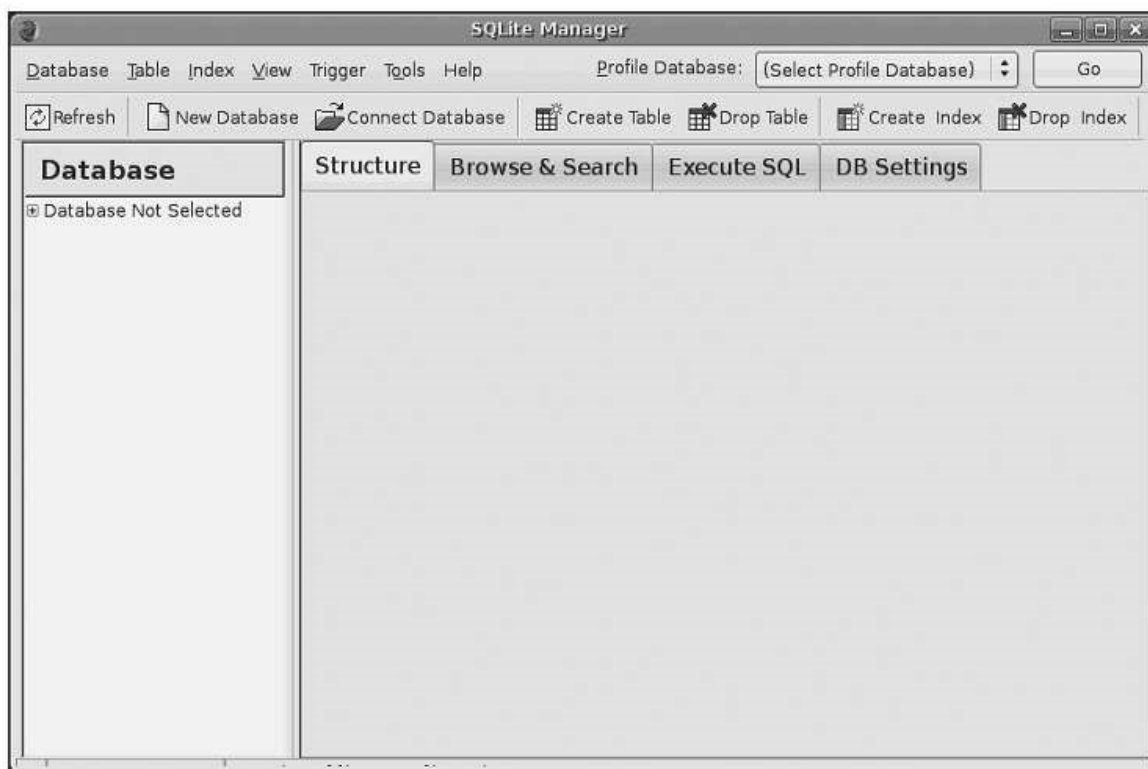
➤ Preferences có thể dùng cho activity độc lập hay được chia sẻ giữa các activity của một ứng dụng. Và tương lai, Preference có thể được chia sẻ giữa các ứng dụng với nhau. Bên cạnh việc sử dụng Preferences để lưu trữ dữ liệu thì cũng còn một cách đơn giản khác để thực hiện công việc lưu trữ dữ liệu là sử dụng tập tin. Android cung cấp 2 mô hình để truy cập tập tin: Một cho các tập tin được đóng gói sẵn cho ứng dụng. Một dành cho tập tin được tạo trên thiết bị thực thi ứng dụng đó. Đối với dữ liệu tĩnh, chẳng hạn danh sách các từ dùng để kiểm tra chính tả. Cách đơn giản nhất để thực hiện điều đó là lưu vào một tập tin, và tập tin này phải được đặt trong thư mục res/raw của project. Nhờ vậy, tập tin sẽ được đặt trong tập tin APK của ứng dụng như một nguồn tài nguyên cơ bản. Ngoài 2 phương pháp trên, Android còn hỗ trợ hệ quản trị cơ sở dữ liệu quan hệ SQLite để quản lý dữ liệu lớn và có quan hệ phức tạp. Nhờ vậy, người lập trình có thể thực hiện quản lý dữ liệu một cách dễ dàng, linh hoạt hơn.

➤ SQLite Khi lập trình trên di động hay các thiết bị có dung lượng bộ nhớ hạn chế, người ta thường dùng SQLite. SQLite là một hệ quản trị cơ sở dữ liệu nhúng được hiện thực từ chuẩn SQL-92. Giống với cái tên của nó, SQLite chiếm dung lượng nhỏ (khoảng 275KB) nên việc truy xuất dữ liệu được nhanh chóng, không chiếm dụng quá nhiều tài nguyên hệ thống. Do SQLite là phần mềm mã nguồn mở nên nó không bị giới hạn tác quyền. Vì lý do đó mà SQLite được nhiều hãng sử dụng (Adobe, Apple, Google, Sun, Symbian) và các dự án mã nguồn mở (Mozilla, PHP, Python). Đặc biệt, đối với Android, SQLite rất thích hợp để tạo cơ sở dữ liệu cho các ứng dụng trên nền Android. Từ khi SQLite có giao diện SQL thì nó càng được sử dụng dễ dàng và thông dụng hơn. SQLite tuy nhẹ hơn so với các hệ cơ sở dữ liệu khác nhưng cũng không khác biệt nhiều. SQLite cũng sử dụng ngôn ngữ truy vấn SQL (SELECT, INSERT, DELETE...), thích hợp cho các ứng dụng trên điện thoại, đặc biệt là ứng dụng trên hệ điều hành Android vì Android Runtime có thể chứa cả SQLite mà không cần phải chia nhỏ ra. SQLite có một hệ thống câu lệnh SQL đầy đủ với các triggers, transactions... Các câu truy vấn cũng như các hệ cơ sở dữ liệu khác. SQLite như bản thu nhỏ của so với các hệ CSDL khác, vì vậy nó không thể có đầy đủ các chức năng trên chiếc điện thoại di động của bạn. SQLite là một lựa chọn thích hợp dành cho ứng dụng trên hệ điều hành Android. Ngoài dung lượng lưu trữ nhỏ gọn, SQLite còn cho phép sử dụng Unicode, kiểu dữ liệu không được cài đặt trong một số phiên bản Android.

➤ Đặc trưng của SQLite được hiện thực từ tiêu chuẩn SQL-92 của một ngôn ngữ SQL nhưng vẫn còn chứa một số khiếm khuyết. Tuy SQLite hỗ trợ trigger nhưng bạn không thể viết trigger cho view. Hoặc SQLite không hỗ trợ lệnh ALTER TABLE, do đó, bạn không thể thực hiện chỉnh sửa hoặc xóa cột trong bảng. SQLite không hỗ trợ ràng buộc khóa ngoại, các transactions lồng nhau, phép kết right outer join, full outer join. SQLite sử dụng kiểu dữ liệu khác biệt so với hệ quản trị cơ sở dữ liệu tương ứng. Bạn có thể insert dữ liệu kiểu string vào cột kiểu integer mà không gặp phải bất kỳ lỗi nào. Vài tiến trình hoặc luồng có thể truy cập tới cùng một cơ sở dữ liệu. Việc đọc dữ liệu có thể chạy song song, còn việc ghi dữ liệu thì không được phép chạy đồng thời. Ngoài các khiếm khuyết trên thì SQLite cung cấp cho người dùng gần như đầy đủ các chức năng mà một hệ cơ sở dữ liệu cần có như tạo database; tạo bảng; thêm, xóa, sửa dữ liệu.

- SQLite Manager

- SQLite Manager là công cụ hiệu quả hỗ trợ cho việc xây dựng cơ sở dữ liệu cho ứng dụng Android đó là SQLite Manager, một add-on miễn phí của Firefox



Hình 2. 17 Màn hình làm việc của sqlite manager

Sử dụng SQLite Manager để quản lý cơ sở dữ liệu Android Tạo database mới : Vào menu Database/ New Database Tạo bảng mới: Tại Database vừa tạo, nhấp chuột phải chọn Create table. Thiết lập các giá trị thích hợp. Xử lý dữ liệu: Chọn bảng cần nhập và chọn tab Browse & Search Nút Add: Thêm dữ liệu Nút Duplicate: Nhân đôi dữ liệu Nút Edit: Chỉnh sửa dữ liệu Nút Delete: xóa dữ liệu Thực thi câu truy vấn: Chọn tab Execute SQL, Nhập câu lệnh SQL vào khung Enter SQL và chọn Run SQL. Kết quả sẽ được trình bày ở khung phía dưới. Nếu câu SQL có lỗi sẽ được hiển thị tại ô Last Error. Tùy chỉnh Database: Chọn tab DB Settings và thiết lập các giá trị như mong muốn Import Database vào ứng dụng Android Sau khi tạo CSDL bằng SQLite Manager, chọn menu File/Export Database. Lưu file .sql vừa tạo vào máy. Để import database vào ứng dụng, người lập trình cần : Build and Run chương trình Android. Khởi động máy ảo xong, chọn nút DDMS ở góc trên bên phải chương trình Eclipse để xem cấu trúc file. Vào thư mục chứa database có đường dẫn như sau : /data/data/your.app.package/databases/your-db-name Sau đó chọn nút import và chọn tập tin database.

### Chương 3: Khảo sát và phân tích hệ thống

#### 3.1 Khảo sát thực tế

Thứ nhất là “thiên thời”, kinh tế vĩ mô của Việt Nam trong 2014 được cải thiện rõ rệt hơn. Nợ xấu của ngân hàng cũng đang được cải thiện. TTCK phát triển ổn định với mức tăng khoảng 15 - 20% so với đầu năm... Thứ hai là “địa lợi”, giá bất động sản tại Hà Nội đã quay lại giá của cách đây 5 - 6 năm. Sự dịch chuyển dân số từ nông thôn ra

thành thị sẽ tiếp tục tăng, khiến nhu cầu về nhà ở tăng cao. Thứ ba là “nhân hòa”, lòng tin của khách hàng đã dần quay trở lại khiến giao dịch tăng dần. Rõ ràng, đến thời điểm này, không còn bàn cãi nhiều khả năng thị trường mạnh nha một chu kỳ. Trong cuộc trò chuyện với người viết, ông Vũ Cương Quyết, Tổng giám đốc Đất Xanh Miền Bắc vừa công bố một cuộc khảo sát ngẫu nhiên từ khách hàng trong 4 năm qua khá thú vị. Cụ thể, nhu cầu đầu cơ trong 2011 chiếm khoảng 8%, tuy nhiên qua 2012 giảm xuống chỉ còn 3%, năm 2013 xuống còn 2% và tháng 10/2014 tăng lên khoảng 4%. Qua khảo sát cũng cho thấy, năm 2011 đầu tư ngắn hạn chiếm 12% và dài hạn chiếm 18%, thì đến năm 2012 lần lượt là 5% và 11%, năm 2013 là 4% và 13% và tháng 10/2014 là 5% và 15%. Qua số liệu, ông Quyết nhận xét, nhu cầu đầu tư dài hạn vẫn còn tồn tại và sẽ tiếp tục tăng. Nhiều nhà đầu tư vẫn đổ tiền vào thị trường bất động sản trong những năm được đánh giá là băng giá của thị trường này. Thời bất động sản “sốt nóng”, hầu hết các chủ đầu tư tập trung vào cao cấp mà quên đi phân khúc trung bình và sơ cấp. Vì vậy, ông Quyết đưa ra lời khuyên, các chủ đầu tư cần định vị phân khúc sản phẩm theo hướng đa dạng, phù hợp cho khách hàng, đồng thời phải phát triển dự án tuân theo nhu cầu thực tế của xã hội. Thực tế, không phải bây giờ, mà nhiều chủ đầu tư từ lâu cũng đã nhận ra được những vấn đề như lời khuyên của ông Quyết đưa ra và họ đã có những thay đổi. Nhiều dự án xin chuyển đổi công năng, xin chia nhỏ căn hộ để phù hợp với nhu cầu của thị trường. Không chỉ giảm giá bán, chủ đầu tư, đơn vị phân phối còn thiết kế nhiều chương trình khuyến mãi hấp dẫn, như mua nhà tặng xe, tặng vàng, tặng tivi, tour du lịch, thậm là mua nhà được tặng nhà... Không chỉ dừng lại ở đó, có chủ đầu tư còn tự nguyện hy sinh lợi nhuận để xây thêm các tiện ích cho khách hàng như bể bơi, công viên, phòng tập gym... Đó chính là lý do giải thích vì sao thanh khoản thị trường trong năm nay tăng đột biến so với các năm trước. Có thể nói, chưa bao giờ vai trò của người mua trên thị trường bất động sản được đề cao như hiện nay. Khách hàng đã được trả lại đúng vị trí “thượng đế” của mình. Tuy nhiên, với một người có kinh nghiệm và đã từng chứng kiến nhiều đợt thăng trầm của thị trường địa ốc như ông Đặng Hùng Võ

### **3.1.1 Tổng Quan Chức năng**

Phân tích chức năng

Chúng ta sẽ sử dụng kiến thức đã tìm hiểu để xây dựng một ứng dụng có tên là bất động sản.

Ban đầu chúng ta sẽ có một giao diện bao gồm

Tạo tài khoản, Đăng nhập tài khoản, mật khẩu và lưu tài khoản, reset mật khẩu về mặc định, Thoát Tài Khoản

Vào Trang chính thông tin tác cả sản phẩm, tạo tài khoản, thông tin apps

Vào Trang Đăng sản phẩm lên trang chính bao gồm

Tên sản phẩm

Thông tin sản phẩm

Giá sản phẩm

Hình sản phẩm

Button đăng sản phẩm

Vào trang chính chọn sản phẩm sẽ ra sản phẩm cần xem

### **3.2 Yêu cầu đặt ra cho hệ thống**

Xây dựng chức năng

Yêu cầu:

Đã cài đặt SDK Android và plugin Android cho Eclipse

Tool sử dụng:

Android Studio/ Eclipse

Android 8.0.1/Genymotion

Trong bất cứ ứng dụng nào thì cũng phải có giao diện để tương tác với người dùng. Vì vậy ta sẽ tìm hiểu và xây dựng một giao diện apps. Trong Android chúng ta có hai cách để làm nên một giao diện. Cách thứ nhất là chúng ta viết code để xây dựng giao diện nhưng nó sẽ có một nhược điểm là làm cho ta rối khi chương trình xảy ra lỗi hoặc nâng cấp, cách thứ hai đó là trong android hỗ trợ chúng ta xây dựng giao diện bằng file XML (Extensible Markup Language) ngôn ngữ đánh dấu mở rộng. Khi sử dụng cách thứ hai cho chương trình của chúng ta sẽ rõ ràng hơn. Vì vậy chúng ta sẽ sử dụng cách thứ hai để xây dựng giao diện cho Bất Động Sản

Đầu tiên chúng ta sẽ tạo giao diện khi chưa vào cho apps

### Màn Hình Chào

Nhằm tạo sự chú ý đến khách hàng màn hình chào còn được gọi là splashscreen rất quan trọng trong doanh nghiệp và ứng dụng apps, màn hình chào splashscreen chúng ta sẽ cho chạy theo thời gian 1 minisecons khi kết thúc 1 milisecons màn hình splashscreen sẽ vào màn hình chính giúp người dùng có thể nhớ tạo sự chú ý đến doanh nghiệp

## **3.3 GIỚI THIỆU CÁC NGÔN NGỮ LẬP TRÌNH ANDROID**

C++ Đây là ngôn ngữ lập trình thích hợp và mạnh mẽ nhất khi xây dựng các ứng dụng di động cho Android và Windows - chủ yếu dành cho lập trình cấp thấp, nó vẫn là ngôn ngữ đi vào nền tảng cho các nhà phát triển ứng dụng trên điện thoại di động. Là một ngôn ngữ lập trình mạnh, thừa hưởng những ưu điểm của ngôn ngữ lập trình C, C++ cho phép các ứng dụng di động được phát triển cho mọi mục đích trên mọi nền tảng tồn tại. Nó có thể không được sang trọng hoặc hợp thời trang, nhưng nó đã thống trị thế giới lập trình ngay cả trước khi cuộc cách mạng điện thoại thông minh.

C # là một ngôn ngữ tuyệt vời. C# là tất cả mọi thứ về Java mà không có bất kỳ phần xấu, được lập trình tốt hơn từ tất cả các cải tiến hàng đầu. Microsoft đã nhìn thấy tiềm năng của Java và quyết định tạo một phiên bản tốt hơn của riêng họ. Trong quá khứ, nhược điểm lớn nhất của C # là nó chỉ có thể chạy trên các hệ thống Windows vì nó dựa vào .Net Framework. Nhưng tất cả điều đó đã thay đổi khi Microsoft mở nguồn .NET Framework vào năm 2014 và mua lại Xamarin vào năm 2016, công ty duy trì Mono (một dự án cho phép các chương trình C # chạy trên nhiều nền tảng). Do đó bạn vẫn có thể dùng được C# để lập trình ứng dụng Android. Ngày nay, bạn có thể sử dụng Xamarin.Android và Xamarin.iOS để tạo các ứng dụng di động bản địa với Visual

Studio hoặc Xamarin Studio. Đây là một điều tuyệt vời bởi vì bạn có thể sử dụng ngôn ngữ trong các ngữ cảnh khác sau này, chẳng hạn như thiết kế các trò chơi phức tạp với Unity và C#. Ví dụ về một ứng dụng được thiết kế với Xamarin? MarketWatch.

Java là một trong những ngôn ngữ ưa thích nhất khi phát triển ứng dụng Android. Một ngôn ngữ lập trình hướng đối tượng được phát triển tại Sun Microsystems (nay thuộc sở hữu của Oracle), Java có thể chạy theo hai cách khác nhau: trong cửa sổ trình duyệt, hoặc trong một máy ảo có thể làm mà không có trình duyệt. Tính linh hoạt này có ý nghĩa rất nhiều khi nói về việc sử dụng lại code và cập nhật phần mềm. Mặc dù Java không có nhiều việc phải làm nếu bạn đang xem xét phát triển iOS, nhưng chắc chắn nó có thể nằm trong danh sách bạn chọn khi nói đến các ứng dụng di động trên các nền tảng, tức là các ứng dụng nền tảng.

Kotlin là một ngôn ngữ phát triển dựa vào Java Virtual Machine được phát triển bởi JetBrains - Công ty phát triển IntelliJ IDE. Các tính năng thú vị của Kotlin đó là trực quan và dễ học, hầu hết các phần của Kotlin rất giống với những gì chúng ta đã biết, IDE Android studio đã được kết hợp Kotlin free.

HTML5, CSS và JavaScript, ban đầu là trifecta cốt lõi cho việc phát triển front-end web, đã phát triển trở nên hữu dụng hơn. Bây giờ bạn có thể thiết kế đa dạng nhiều loại apps, cả điện thoại di động và máy tính để bàn, chỉ cần sử dụng HTML5, CSS và JavaScript. Về bản chất, bạn đang tạo một ứng dụng web (web applications) được hiển thị dưới dạng offline apps thông qua nền tảng ảo thuật. Để thiết kế ứng dụng Android theo cách này, bạn có thể sử dụng Adobe Cordova, một khuôn khổ mã nguồn mở cũng hỗ trợ iOS, Windows 10 Mobile, Blackberry, Firefox và nhiều hơn nữa. Nhưng bên cạnh những tính hữu dụng của nó, Cordova đòi hỏi rất nhiều công sức để tạo ra được một ứng dụng chạy tốt, đó là lý do tại sao nhiều người chọn Ionic Framework để thay thế (vì nó sử dụng Cordova để triển khai cho các nền tảng khác nhau).

Những ứng dụng Android điển hình như Untappd và TripCase. Một sự lựa chọn khác là sử dụng React Native. Thư viện này có thể triển khai trên Android, iOS và nền tảng Windows chung. Nó được duy trì và sử dụng bởi Facebook, Instagram, và các công ty lớn khác, vì vậy bạn biết nó đã được thử nghiệm. Quá trình học tập không phải là dễ dàng, nhưng một khi bạn đã vượt qua được Hurdle ban đầu, bạn sẽ có tất cả sức mạnh và tính linh hoạt mà bạn cần.

Python bản địa nhưng vẫn có những công cụ cho phép bạn tạo apps trên Python và sau đó chuyển đổi chúng thành các APK chạy thành công trên thiết bị Android. Đây là một ví dụ điển hình của Python ngôn ngữ thực tiễn hiệu quả và đó cũng là tin tốt cho những người hâm mộ Python, những người muốn thiết kế các apps nhưng lại không muốn sử dụng các trappings của Java.

Lua là một ngôn ngữ lập trình cũ với mục đích thiết kế ban đầu nhằm để bổ sung cho những chương trình được viết bằng những ngôn ngữ tiên tiến hơn như C, VB.NET, etc. Như vậy, nó có một vài tính năng nổi bật, khác biệt, như là bắt đầu mảng tại vị trí số 1 thay vì 0 và không có các native classes. Ngoài ra Lua còn có thể được sử dụng như một ngôn ngữ lập trình chính trong một vài trường hợp nhất định và Corona SDK là một ví dụ điển hình. Với Corona, bạn có thể tạo ra những ứng dụng đa dạng mà có thể được ứng dụng trên Windows, Mac, Android, iOS và kể cả Apple TV và Android TV. Nó cũng có các tính năng kiểm tiền tích hợp sẵn cộng với một thị trường rộng lớn

nơi bạn có thể mua assets và plugins. Corona thường được sử dụng để thiết kế các trò chơi (như Fun Run 2 và HoPiko) ngoài ra nó cũng được sử dụng để tạo ra các tiện ích chung và các apps kinh doanh (như My Days và Quebec Tourism). Để có thể trở thành lập trình viên Android, tất nhiên bạn không cần phải học hết tất cả các ngôn ngữ lập trình này. Bạn nên chọn ra những mình những ngôn ngữ thể mạnh và nghiên cứu sâu. Chúc các lập trình viên Android sẽ luôn thành công!

## **Chương 4: Thiết kế hệ thống**

### **4.1 Thiết kế giao diện apps**

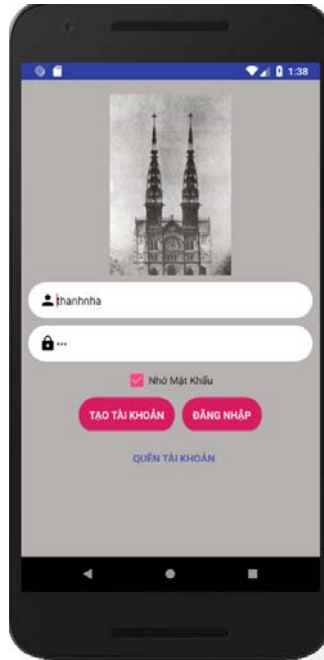
Nhằm tạo sự chú ý đến khách hàng màn hình chào còn được gọi là splashscreen rất quan trọng trong doanh nghiệp và ứng dụng apps, màn hình chào splashscreen chúng ta sẽ cho chạy theo thời gian 1 minisecons khi kết thúc 1 milisecons màn hình splashscreen sẽ vào màn hình chính giúp người dùng có thể nhớ tạo sự chú ý đến doanh nghiệp



Hình 4. 1 Màn Hình Chào

Giúp khách hàng có thể tạo bài viết sản phẩm các căn hộ chúng ta cần cho người dùng đăng nhập vào ứng dụng thông qua các mô hình webservice/webapi/local host/ ở đây tôi dùng local host vì chưa cấu hình được server riêng máy chủ nên chỉ có thể chạy offline và không chạy online được, ở đây chúng tôi cho tam nền màu xám với tam hình đại diện là nhà thờ đức bà năm thập kỷ 90 và cho 2 edittext cho người dùng có thể bấm vào và điền thông tin sau khi đăng ký và có tài khoản thành công người dùng

có thể đăng nhập vào trang quản lý sản phẩm và trang xem tất cả sản phẩm của bạn nếu bạn quên mật khẩu bạn có thể reset mật khẩu thông qua ứng dụng bằng email của bạn đã đăng ký và số điện thoại sẽ xuất thông báo ra cho bạn mật khẩu bạn là mặc định là 12345 để cho khách hàng nhớ mật khẩu, trong màn hình đăng nhập chúng tôi có để 1 checkbox giúp khách hàng sau khi nhập vào tài khoản có thể bấm vào nhớ mật khẩu giúp ứng dụng có thể nhớ được tài khoản và mật khẩu của bạn đã đăng nhập vào lần đầu tiên sau khi đăng nhập lại ứng dụng sẽ tự hiểu đã có tài khoản và đã đăng nhập lại lần thứ 2



Hình 4. 2 Màn hình Đăng Nhập

Bảng biểu 4. 1 Đăng Nhập

Tên đối tượng	Kiểu đối tượng	Dữ liệu	Mục đích
Đăng nhập	TextView	Nhập từ bàn phím	
Tên đăng nhập	EditText	Nhập từ bàn phím	Xác nhận tài khoản
Mật khẩu	EditText	Nhập từ bàn phím	Xác nhận mật khẩu
Lưu đăng nhập	CheckBox		Lưu đăng nhập
Tạo tài khoản	Button		Đăng nhập
Đăng ký	Button		Đăng ký
Quên Tài khoản	Button		Reset

Bảng biểu 4. 2:Đăng Ký

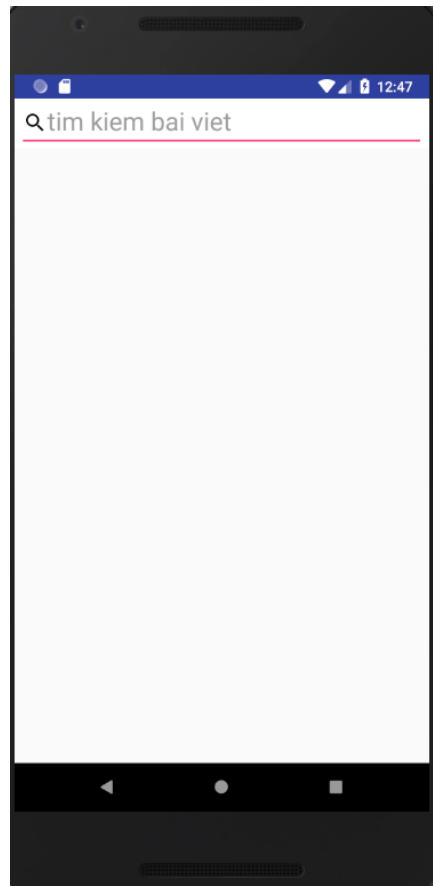
Tên đối tượng	Kiểu đối tượng	Dữ liệu	Mục đích
User-name	EditText	Nhập từ bàn phím	tên đăng nhập
Email	EditText	Nhập từ bàn phím	Tên email
Mật khẩu	EditText	Nhập từ bàn phím	Lưu mật khẩu
Số điện thoại	EditText	Nhập từ bàn phím	Số điện thoại
Đăng ký	Button		Đăng ký
Đăng nhập	Button		Trở về

Bảng biểu 4. 3 Reset

Tên đối tượng	Kiểu đối tượng	Dữ liệu	Mục đích	Mặc định
Email	EditText	Nhập từ bàn phím	Tìm kiếm database	Not null
Reset mật khẩu	Button		Trả về mặc định	12345
Đăng nhập	Button		Đăng nhập	

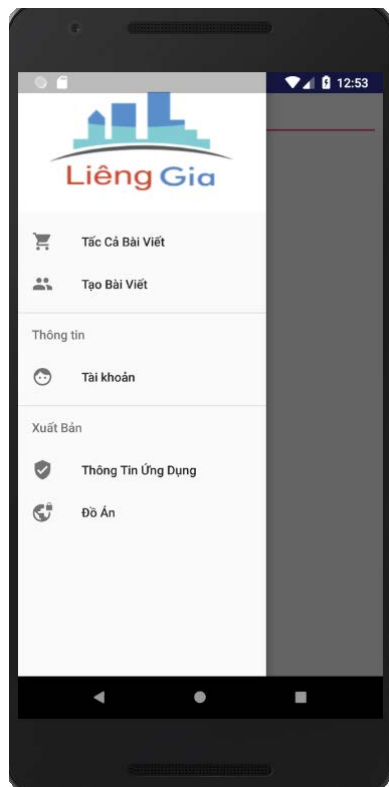
Sau khi đã đăng nhập vào ứng dụng màn hình sẽ xuất ra thanh tìm kiếm bài viết sản phẩm giúp người dùng có thể nhấn vào và tìm bài viết dựa theo các chữ cái đầu hoặc cả tên bài viết giúp người dùng có thể xem được bài viết 1 cách nhanh chóng mà không cần tìm nhiều trên apps, khung trắng phía dưới là màn hình dùng để cho tất cả sản phẩm lên và cho sản phẩm có thể list lên một cách tốt nhất





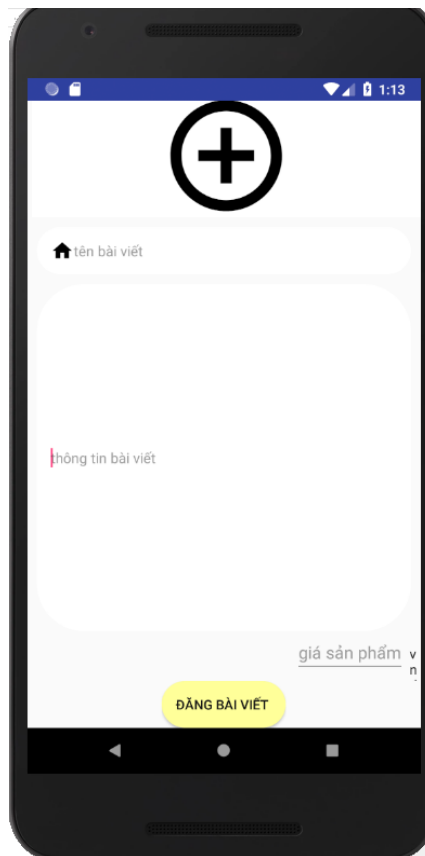
Hình 4. 4 Màn hình xem tác cả sản phẩm bài viết

kéo từ bên trái qua phải chúng ta sẽ thấy các mục đầu tiên sẽ là tấm hình giúp cho khách hàng tạo sự chú ý đến họ, chúng ta sẽ tạo một menu với nhiều mục khác nhau giúp giúp người dùng có thể bấm vào các mục lục trong ứng dụng một cách tốt nhất và cho 1 tấm hình đóng vai trò là 1 header giúp tạo sự chú ý đến khách hàng đang sử dụng ứng dụng từ 1 công ty liêngia, ở mục tác cả bài viết sẽ cho chúng ta xuất ra hiện tác cả sản phẩm của bạn



Hình 4. 3 Màn hình menu

ở mục tạo bài viết khi người dùng bấm vào mục đăng bài viết chúng ta sẽ hiện ra màn hình đăng bài viết giúp cho khách hàng có thể up load hình ảnh lên database hoặc webserver api/web service mục tên bài viết giúp cho khách hàng nhập vào tên bài viết sản phẩm cần đăng bao gồm tên sản phẩm thông tin bài viết giá tiền vnd/\$ và button đăng bài viết



Hình 4. 4 Màn hình tạo bài viết

Sau khi khách hàng đã đăng bài viết xong màn hình tất cả bài viết sẽ hiện ra tất cả sản phẩm bài viết bạn đã đăng

ở mục Tài khoản khi khách hàng bấm vào tài khoản sẽ ra thông tin tài khoản của khách hàng khi đăng ký bao gồm hình liên gia, tài khoản khách hàng, email số điện thoại và button đăng xuất khi khách hàng chọn vào button đăng xuất sẽ đăng xuất tài khoản của khách hàng ra và thoát khỏi ứng dụng

khi khách hàng đăng sản phẩm sẽ list lên tất cả danh sách sản phẩm khách hàng đã đăng lên bao gồm hình, tên bài viết giá tiền đồng VNĐ/đola



Hình 4. 5 Màn hình list tất cả sản phẩm

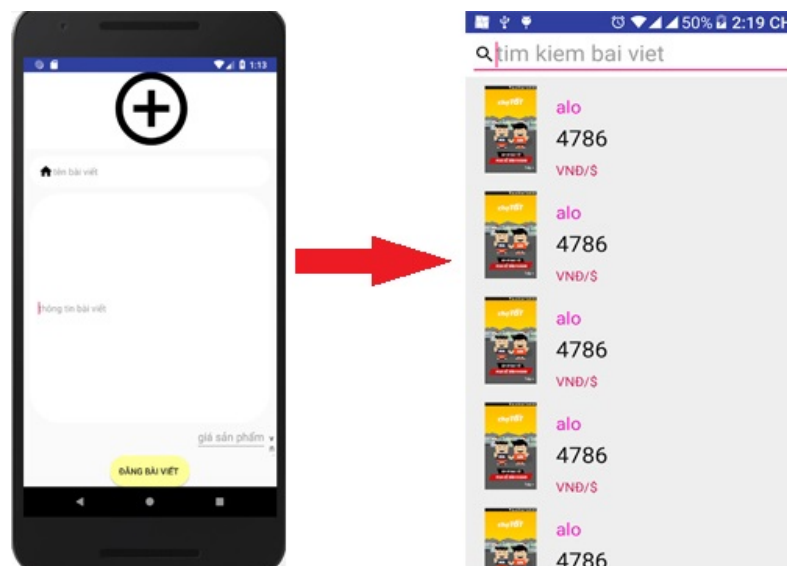
Mục Thông tin ứng dụng giúp người dùng nhớ được ứng dụng do 1 tổ chức/công ty làm ra tạo sự tin tưởng cho người dùng với doanh nghiệp khác



Hình 4. 7 Màn hình thông tin ứng dụng

#### 4.2.Chức năng chính

Hiện thị tác cả sản phẩm lên trang chính Khi khách hàng đã đăng nhập là load hình ảnh lên và giá tiền sản phẩm lên tiến hành bấm vào nút đăng bài viết, bài viết của khách hàng sẽ được lên trang sản phẩm ở trang sản phẩm chúng tôi dùng recyclerview để tối ưu apps



Hình 4. 8 Màn hình tạo bài viết qua trang tác cả sản phẩm

#### 4.3 kho dữ liệu

Dữ liệu từ sản phẩm từ khách hàng được đăng lên sẽ truyền vào server chính sử dụng ngôn ngữ viết PHP-MYSQL để chạy được online 24/24, ở đây chúng tôi dùng local trên máy tính sử dụng ngôn ngữ viết database SQLite chỉ chạy được ở chế độ offline dữ liệu khách hàng sẽ truyền vào database sqlite và truyền ra được sản phẩm của khách hàng đã tạo cho khách hàng lựa chọn các bài sản phẩm bạn vừa đăng, nếu bạn muốn ứng dụng bạn có thể chạy được online cơ bản phải nắm vững các kiến thức về mã nguồn mở nodejs/phpmyadmin-mysql/c#/javascript sau khi đã nắm vững bạn cần phải viết các json cho cơ sở dữ liệu như trên dùng 1 tool apiwebbrow để hỗ trợ dịch ra api/webservice để có thể apps bạn kết nối 1 cách dễ dàng hơn

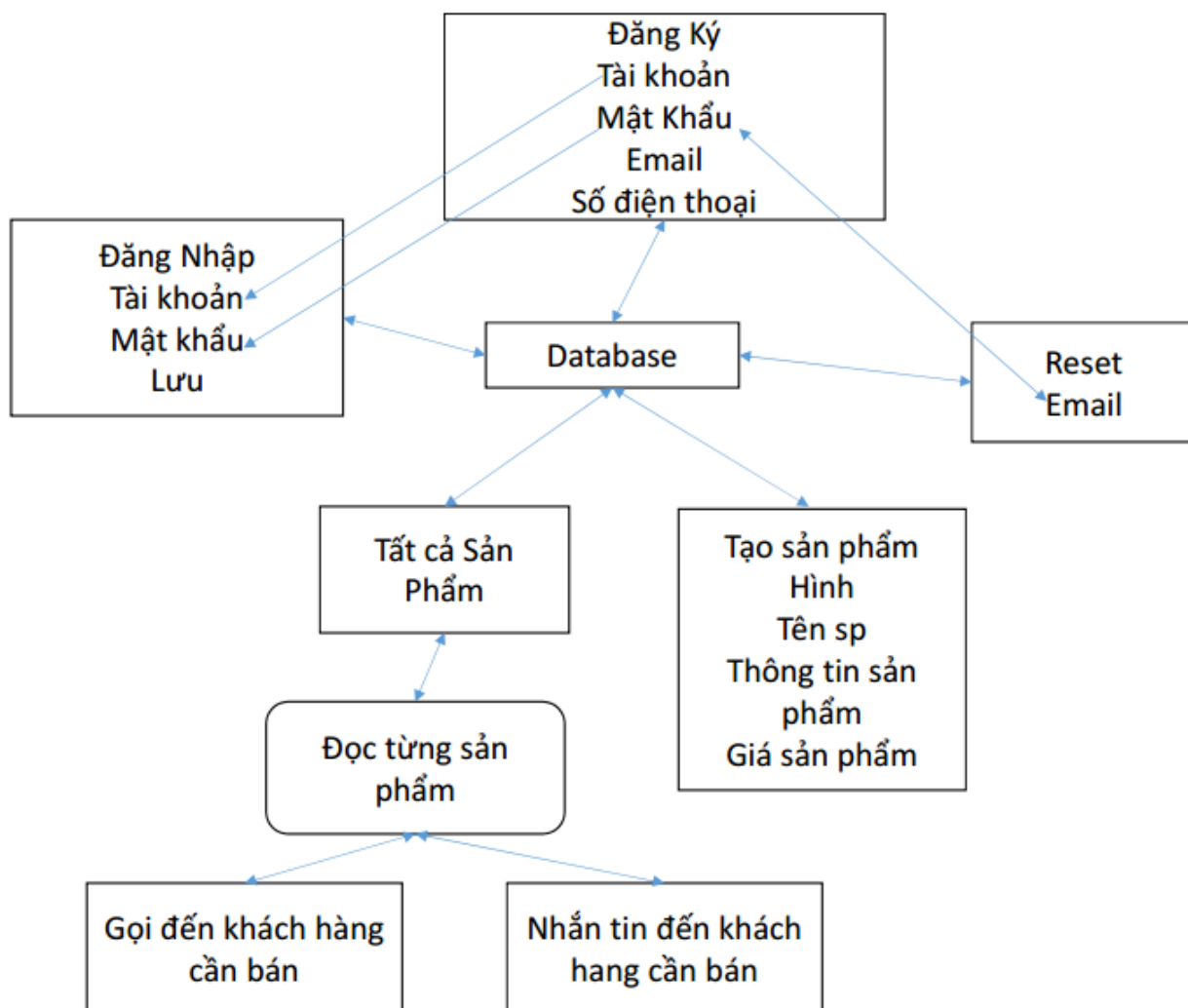
#### **4.4 Các đối tác**

Trong lĩnh vực bất động sản đầy tiềm năng hiện nay ứng dụng bất động sản sẽ đón chào nhiều gương mặt thân quen trong lĩnh vực bất động sản hiện nay ví dụ như

- Sàn giao dịch Alibaba
- Sàn giao dịch Đat Gia
- Sàn giao dịch Unihome
- Sàn giao dịch Vincom
- Sàn giao dịch Vingroup
- Sàn giao dịch Cityland
- Sàn giao dịch Mnland

Và một số sàn giao dịch khác trong lĩnh vực bất động sản và các cò nhà đất hiện nay. Sàn giao dịch Bất Động Sản Đat Gia là nơi có thể đạt niềm tin tưởng nhất để phát triển apps trên nền tảng mobile nhằm kết nối mọi người với nhau cùng nhau mua nhà/căn hộ/đất một cách hợp lý vừa với số tiền mình đang có hiện nay Đat gia Và Liêng gia là hai công ty uy quyền của apps nhằm bảo vệ quyền lợi cho Apps sau này và hiện nay

#### **4.5 Sơ đồ phân cấp chức năng**



Hình 4. 9 Sơ đồ phân cấp chức năng

#### 4.6. Bảng phân tích thiết kế chức năng

Khách hàng là vấn đề cần quan tâm hàng đầu của các ứng dụng, việc đó đòi hỏi người lập phải tạo ra được các chức năng tiện dùng cho khách hàng. Khách hàng đăng ký tài khoản để đăng nhập mua hàng, bán hàng và có thể thực hiện việc liên hệ với quản trị apps khi họ có câu hỏi hay các vấn đề mua hàng cụ thể như:

Bảng biểu 4. 4 Bảng phân tích chức năng của khách hàng

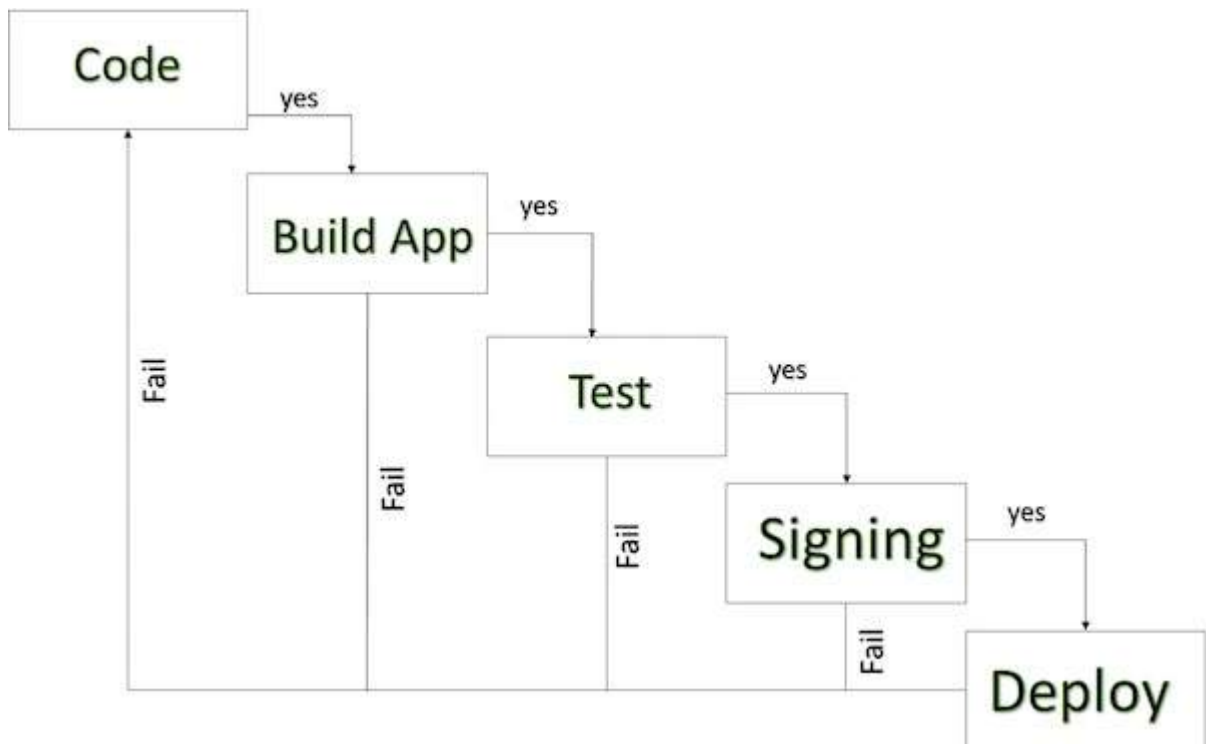
STT	Nội dung	Mô tả chi tiết	Ghi chú
-----	----------	----------------	---------

1	Đăng ký tài khoản	- Khách hàng cần đăng ký tài khoản để đăng nhập vào apps tạo sự bảo mật dành cho khách hàng	
2	Chức năng đăng nhập apps	Khách hàng đăng nhập trở thành thành mua và bán tạo sự thoải mái đến khách hàng	
3	Chức năng quên mật khẩu apps	Khi khách hàng quên mật khẩu sẽ nhập vào email người dùng chọn reset mật khẩu tài khoản sẽ reset về mật định là 12345	
3	Kết nối khách hàng	gọi điện và nhắn tin đến khách hàng a và b	
4	Liên hệ admin	Khách hàng sẽ liên hệ trực tiếp bằng cách gửi email thông qua trang thông tin của ứng dụng	

## **Chương 5:Cài đặt apps**

### **5.1 Đóng gói ứng dụng**

Một khi bạn phát triển và kiểm tra đầy đủ ứng dụng Android, bạn có thể bắt đầu bán hoặc phân phối miễn phí bằng cách sử dụng Google Play (Một thị trường Android nổi tiếng). Bạn cũng có thể phát hành các ứng dụng của bạn bằng cách gửi trực tiếp đến người dùng hoặc bằng cách cho phép người dùng tải về từ trang web của riêng bạn.



Hình 5. 1 Quá trình lên ý tưởng và ra apps

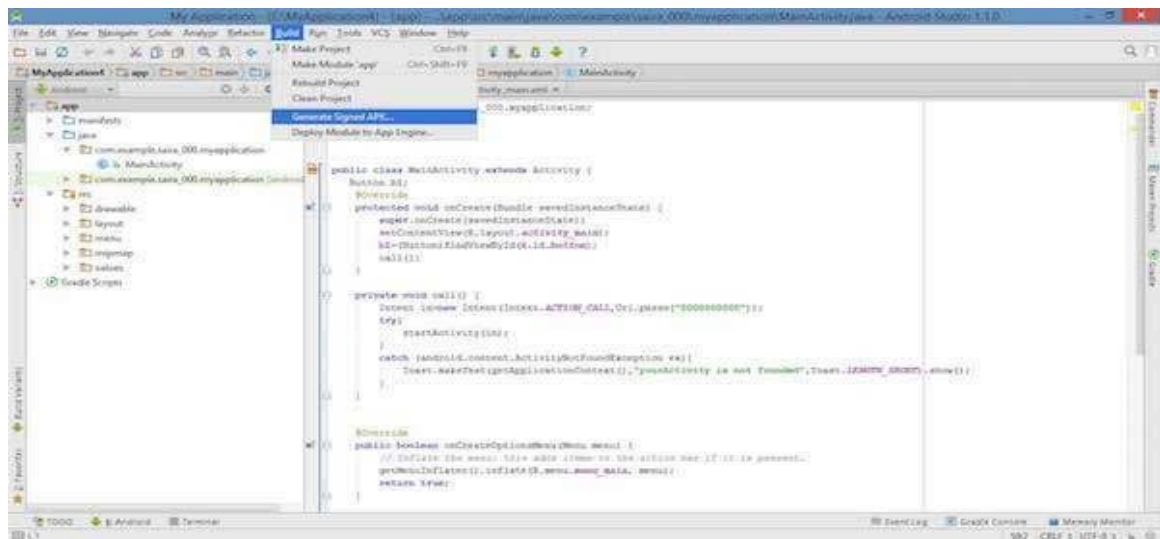
Bạn có thể kiểm tra một quá trình xuất bản chi tiết tại trang web chính thức của Android, nhưng hướng dẫn này sẽ đưa bạn qua các Hình đơn giản để khởi động ứng dụng của bạn trên Google Play. Dưới đây là một danh sách kiểm tra đơn giản này sẽ giúp bạn trong việc khởi chạy ứng dụng Android của bạn.



Bước	Hoạt động
1	<b>Regression Testing:</b> Trước khi bạn xuất bản ứng dụng của bạn, bạn cần phải chắc chắn rằng cuộc họp của những kỳ vọng chất lượng cơ bản cho tất cả các <b>ứng dụng Android</b> , trên tất cả các thiết bị mà bạn đang nhắm mục tiêu. Vì vậy, thực hiện tất cả các thử nghiệm cần thiết trên các thiết bị khác nhau bao gồm cả điện thoại và máy tính bảng.
2	<b>Application Rating :</b> Khi bạn sẽ xuất bản ứng dụng của bạn vào Google Play, bạn sẽ phải chỉ định một đánh giá nội dung cho ứng dụng của bạn, mà thông báo cho người sử dụng Google Play của mức độ trưởng thành của mình. Hiện nay xếp hạng sẵn có (a) Everyone (b) Low maturity (c) Medium maturity (d) High maturity.
3	<b>Targeted Regions:</b> Google Play cho phép bạn kiểm soát những gì các nước và vùng lãnh thổ nơi mà ứng dụng của bạn sẽ được bán. Theo đó bạn phải chăm sóc của việc thiết lập vùng thời gian, nội địa hay bất cứ yêu cầu cụ thể khác theo khu vực mục tiêu.
4	<b>Application Size:</b> Kích Hiện nay, kích thước tối đa cho một apk xuất bản trên Google Play là 50 MB. Nếu ứng dụng của bạn vượt quá kích thước đó, hoặc nếu bạn muốn cung cấp một tải thử cấp, bạn có thể sử dụng APK mở rộng tập tin, mà Google Play sẽ tổ chức miễn phí trên cơ sở hạ tầng máy chủ của mình và xử lý tự động tải về các thiết bị.
5	<b>SDK and Screen Compatibility:</b> Điều quan trọng là để đảm bảo rằng ứng dụng của bạn được thiết kế để chạy đúng trên các phiên bản nền tảng Android và kích cỡ màn hình thiết bị mà bạn muốn nhắm mục tiêu.
6	<b>Application Pricing:</b> Quyết định xem bạn có ứng dụng sẽ được miễn phí hoặc trả tiền là quan trọng bởi vì, trên Google Play, ứng dụng của tự do phải duy trì miễn phí. Nếu bạn muốn bán ứng dụng của bạn sau đó bạn sẽ phải xác định giá trong các loại tiền tệ khác nhau.
7	<b>Promotional Content:</b> Đây là một thực hành tiếp thị tốt để cung cấp một loạt các chất lượng cao tài sản đồ họa để giới thiệu ứng dụng hoặc thương hiệu của bạn. Sau khi xuất bản, các xuất hiện trên trang chi tiết sản phẩm của bạn, trong danh sách cửa hàng và các kết quả tìm kiếm, và các nơi khác.
8	<b>Build and Upload release-ready APK:</b> Việc phát hành sẵn sàng apk là những gì bạn có, bạn sẽ tải lên các Developer Console và phân phối đến người dùng. Bạn có thể kiểm tra đầy đủ chi tiết về làm thế nào để tạo ra một phiên bản phát hành sẵn sàng của ứng dụng của bạn: Chuẩn bị cho phát hành.
9	<b>Finalize Application Detail:</b> Google Play mang đến cho bạn nhiều cách khác nhau để thúc đẩy ứng dụng của bạn và tham gia với người sử dụng trên trang chi tiết sản phẩm của bạn, từ đồ họa đầy màu sắc, chụp màn hình, và video để giới thiệu bản địa hóa, chi tiết phát hành, và các liên kết đến các ứng dụng khác của bạn. Vì vậy, bạn có thể trang trí trang ứng dụng của bạn và cung cấp càng nhiều càng rõ ràng chi tiết sắc nét, bạn có thể cung cấp.

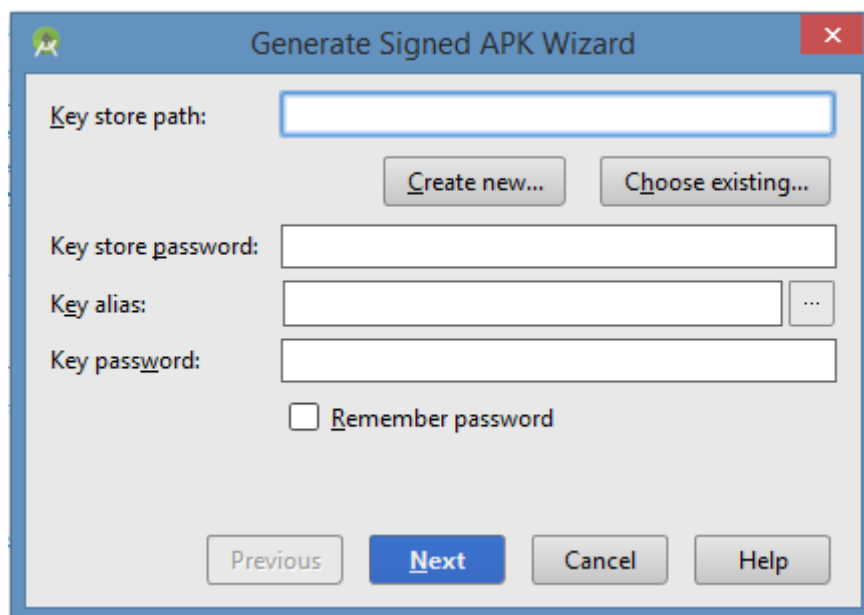
Hình 5. 2 Hoạt động

Để xuất một ứng dụng, chỉ cần mở rằng dự án ứng dụng trong Android Studio và chọn Build → Generate Signed APK từ Android studio của bạn và thực hiện theo các Hình đơn giản để xuất khẩu ứng dụng của bạn:



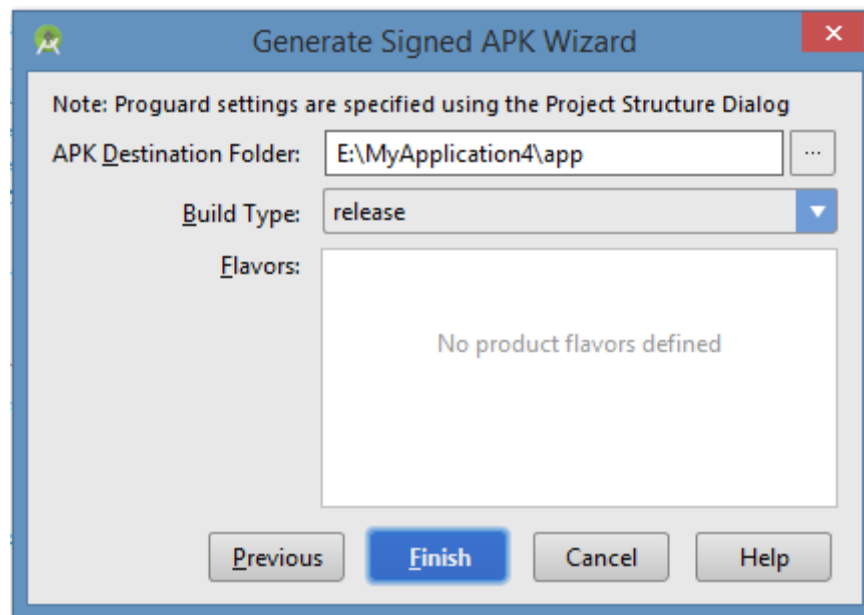
Hình 5. 3 Chọn Generate Signed APK

Tiếp theo chọn, Generate Signed APK như trong ảnh chụp màn hình ở trên và sau đó nhấp vào nó để bạn có được màn hình sau đây bạn sẽ chọn Createnew keystore để lưu trữ ứng dụng của bạn.



Hình 5. 4 Đăng ký chữ ký

Nhập đường dẫn lưu trữ chính của bạn, mật khẩu lưu khóa, chìa khóa bí danh và mật khẩu chính để bảo vệ ứng dụng của bạn và nhấp vào nút Next một lần nữa. Nó sẽ hiển thị sau màn hình cho phép bạn tạo một ứng dụng:



Hình 5. 5 lưu tại nơi chứa thư mục bạn chọn

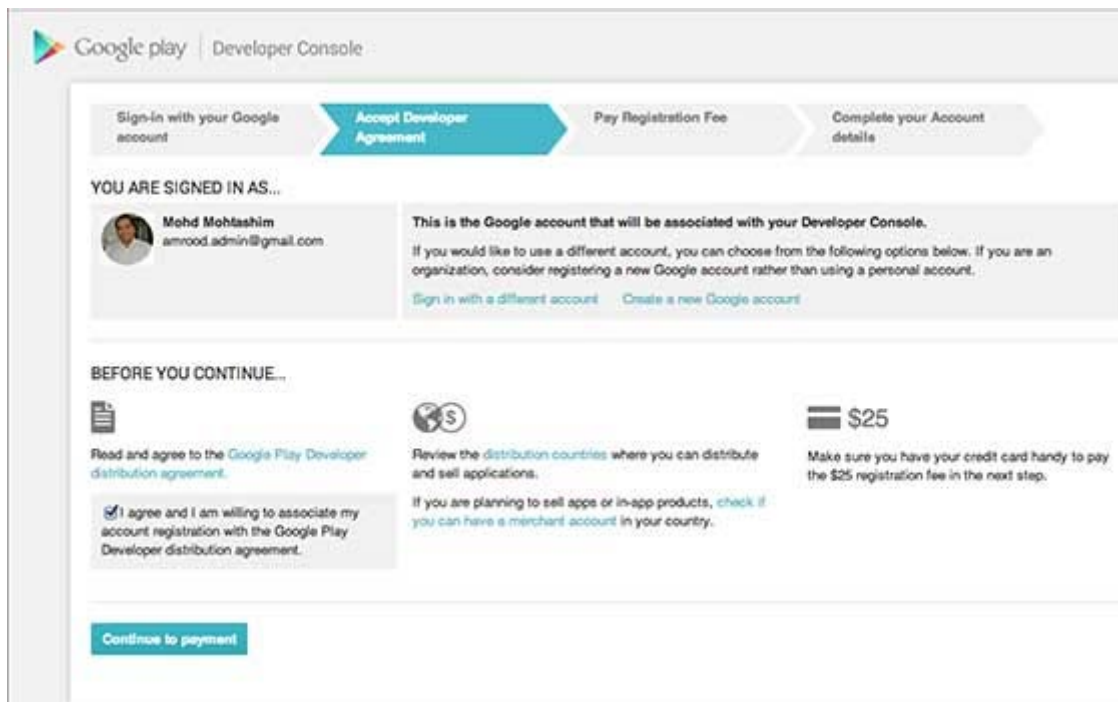
Một khi bạn lấp đầy tất cả các thông tin, như điểm đến ứng dụng, chế tạo loại và hương vị bấm nút hoàn tất khi tạo ra một ứng dụng nó sẽ hiển thị như dưới đây

NAME	DATE MODIFIED	TYPE	SIZE
app-debug.apk	6/1/2018 12:04 PM	APK File	3,697 KB
node.txt	5/29/2018 4:28 PM	Text Document	1 KB
output.json	6/1/2018 12:04 PM	JSON File	1 KB

Hình 5. 6 Tìm đến mục có file apk

## 5.2 Đưa lên ch-play

Hình quan trọng nhất là để đăng ký với Google Play bằng cách sử dụng Google Play Marketplace. Bạn có thể sử dụng ID google hiện tại của bạn nếu bạn có bất kỳ nếu không bạn có thể tạo ra một ID mới của Google và sau đó đăng ký với thị trường. Bạn sẽ có màn hình sau đây để chấp nhận các điều khoản và điều kiện



Hình 5. 7 Tạo tài khoản google development

Bạn có thể sử dụng nút Continue to payment để tiến tới thực hiện thanh toán là \$ 25 như một khoản phí đăng ký và cuối cùng để hoàn thành chi tiết tài khoản của bạn. Khi bạn là một người sử dụng đăng ký tại Google Play, bạn có thể tải lên phát hành sẵn sàng apk cho ứng dụng của bạn và cuối cùng bạn sẽ hoàn thành chi tiết ứng dụng sử dụng trang chi tiết ứng dụng như đã đề cập trong Hình 9 của danh sách kiểm tra đã đề cập ở trên. Bạn không cần Android Studio ký ứng dụng của bạn. Bạn có thể đăng ký ứng dụng của bạn từ dòng lệnh bằng cách sử dụng các công cụ tiêu chuẩn từ Android SDK và JDK. Để đăng ký một ứng dụng trong chế độ phát hành từ dòng lệnh

Tạo một khóa riêng bằng cách sử dụng keytool

```
$ keytool -genkey -v -keystore my-release-key.keystore
alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```

Biên dịch ứng dụng của bạn trong chế độ phát hành để có được một apk unsigned  
Đăng tin ứng dụng của bạn với phím của bạn bằng cách sử dụng jarsigner

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
keystore my-release-key.keystore my_application.apk alias_name
```

Xác minh rằng apk của bạn được ký

```
jarsigner -verify -verbose -certs my_application.apk
```

Gói APK sử dụng zipalign.

```
$ zipalign -v 4 your_project_name-unaligned.apk your_project_name.apk
```

Sau 7 ngày đối với cá nhân google sẽ submit apps của bạn lên google đối với các doanh nghiệp sau 1-2h google sẽ submit apps bạn lên chplay(store)

## **Chương 6:Tài liệu tham khảo**

Lập Trình Java cơ bản tiếng việt udemy online(VietJack)  
Lập Trình Android cơ bản (FPT POLYTECHIC)  
Lập Trình Android nâng cao (FPT POLYTECHIC)  
Kiểm Thử Và Triển Khai Ứng Dụng Android (FPT POLYTECHIC)  
Lập Trình Hướng Đối Tượng (FPT POLYTECHIC)  
Lập Trình Java(FPT POLYTECHIC)  
Lập Trình Java Core(FPT POLYTECHIC)  
Lập Trình Java Swing(FPT POLYTECHIC)  
Lập Trình Java Mạng Máy Tính  
Bảo Mật Lập Trình Mạng Trong Java(Phan Thế Bảo)  
Hướng Dẫn Lập Trình Mạng Bằng Java(Đậu Văn Tuấn/Nguyễn Viết Linh)  
Tự Học JavaServerPages trong 24 giờ(Thuận Thành-Thành Tân)  
Hướng Dẫn Học JavaScript qua ví dụ(FPT POLYTECHIC)  
Kiểm Thử Phần Mềm<Testing>(Trần Tường Thụy-Phạm Quang Hiền)  
Lập Trình Java IOT (ORACLE)  
Lập Trình Java Programing 24-Hour Trainer(YAKOV FAIN)  
Thiết Kế Giao Diện Android Cơ Bản (FPT POLYTECHIC)  
Lập Trình PHP Và MYSQL Cơ Bản (FPT POLYTECHIC)  
Lập Trình PHP Và MYSQL Nâng Cao (FPT POLYTECHIC)  
Video lập trình java apps  
<https://drive.google.com/open?id=0B6UqbP7Q2QG7RVdTRGFKQUhBdTQ>  
[https://drive.google.com/open?id=0B0\\_tJ0GT5ABSTnNUZVluV2NHbnM](https://drive.google.com/open?id=0B0_tJ0GT5ABSTnNUZVluV2NHbnM)  
<https://drive.google.com/open?id=0B8-0VqpTa9ZzYUIFWnp1eER5R28>