

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

---



ISO 9001:2015

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Thanh Tùng**  
**Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2018**

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

-----

**XÂY DỰNG ỨNG DỤNG ANDROID LẤY DỮ LIỆU TỰ ĐỘNG  
TỪ TRANG WEB**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Thanh Tùng  
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2018**

Trường ĐHDL Hải Phòng Khóa luận tốt nghiệp

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

-----

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Nguyễn Thanh Tùng

Mã SV: 1412101073

Lớp: CT1802

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng ứng dụng Android lấy dữ liệu tự động từ trang web

## LỜI CẢM ƠN

Lời đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc nhất tới thầy ThS. Phùng Anh Tuấn, thầy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em xin chân thành cảm ơn sự dạy bảo và giúp đỡ của các thầy giáo, cô giáo Khoa Công Nghệ Thông Tin - Trường Đại Học Dân Lập Hải Phòng đã trang bị cho em những kiến thức cơ bản nhất để em có thể hoàn thành tốt báo cáo tốt nghiệp này.

Xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt là các bạn trong lớp CT1802 đã giúp đỡ và đóng góp ý kiến để em hoàn thành chương trình.

Hải Phòng, ngày 10 tháng 08 năm 2018

Sinh viên

Nguyễn Thanh Tùng

|  |    |
|--|----|
| <b>LỜI CẢM ƠN</b> .....  | 1  |
| <b>CHƯƠNG 1 : HỆ ĐIỀU HÀNH ANDROID</b> .....                             | 7  |
| <b>1.1 Giới thiệu về hệ điều hành Android</b> .....                      | 7  |
| <b>1.2. Sự phát triển của hệ điều hành Android</b> .....                 | 8  |
| <b>1.3 Kiến trúc của hệ điều hành Android</b> .....                      | 11 |
| 1.3.1 Thư viện .....   | 11 |
| 1.3.2 Thực thi Android.....  | 12 |
| 1.3.3 Nền tảng Android.....  | 12 |
| 1.3.4 Tầng ứng dụng .....  | 13 |
| <b>1.4 Giao diện hệ điều hành Android</b> .....                          | 13 |
| <b>1.5 Các phiên bản Android</b> .....                                   | 15 |
| <b>CHƯƠNG 2 : MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO</b><br>..... | 24 |
| <b>2.1. Giới thiệu Android Studio</b> .....                              | 24 |
| <b>2.2. Cài đặt môi trường</b> .....                                     | 24 |
| 2.2.1 Cài đặt JAVA JDK .....   | 24 |
| 2.2.2 Cài đặt Android Studio .....                                       | 25 |
| 2.2.3 Phần mềm Genymotion .....  | 30 |
| 2.3.1 Tập cấu hình dự án Android .....                                   | 31 |
| 2.3.2. Thư mục Java.....   | 33 |
| 2.3.3. Thư mục Res .....   | 33 |
| <b>2.4 Thành phần giao diện</b> .....                                    | 34 |
| 2.4.1 Một số lớp nhóm hiển thị cơ bản.....                               | 35 |
| 2.4.2 View .....   | 36 |
| <b>2.5. Vòng đời ứng dụng android</b> .....                              | 39 |
| <b>CHƯƠNG 3 : KỸ THUẬT LẤY DỮ LIỆU TỰ ĐỘNG TỪ TRANG WEB</b> .....        | 43 |
| <b>3.1. Mạng Internet</b> .....  | 43 |
| <b>3.2. Dịch vụ lưu trữ trực tuyến</b> .....                             | 43 |

|  |           |
|--|-----------|
| 3.2.1. Giới thiệu.....   | 43        |
| 3.2.2. Các hoạt Hosting.....   | 44        |
| 3.2.3. Các yêu cầu và tính năng cần thiết của hosting .....            | 44        |
| 3.2.4. Dung lượng hosting .....  | 44        |
| 3.2.5. Tại sao cần phải mua hosting .....                              | 45        |
| <b>3.3. Dịch vụ web .....</b>  | <b>45</b> |
| 3.3.1. Giới thiệu.....   | 45        |
| 3.3.2 Các công nghệ xây dựng dịch vụ Web.....                          | 45        |
| 3.3.3 Đặc điểm của dịch vụ Web.....                                    | 45        |
| <b>3.4. Thư viện Simple HTML Dom.....</b>                              | <b>46</b> |
| 3.4.1. Giới thiệu.....   | 46        |
| <b>3.5. Mô hình lấy dữ liệu từ trang web về thiết bị di động .....</b> | <b>47</b> |
| <b>CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM .....</b>                        | <b>48</b> |
| <b>4.1. Phát biểu bài toán .....</b>                                   | <b>48</b> |
| <b>4.2. Một số giao diện chương trình .....</b>                        | <b>49</b> |
| 4.2.1 Giao diện màn hình chính.....                                    | 49        |
| 4.2.2 Giao diện các mục tin .....                                      | 50        |
| 4.2.3 Giao diện khi nhấn vào mục tin.....                              | 51        |
| <b>TÀI LIỆU THAM KHẢO .....</b>  | <b>53</b> |

## **CHƯƠNG 1 : HỆ ĐIỀU HÀNH ANDROID**

### **1.1 Giới thiệu về hệ điều hành Android**

Android [1] 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.

Chính mã nguồn mở của Android cùng với tính không ràng buộc nhiều đã cho phép các nhà phát triển thiết bị di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị.

Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

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

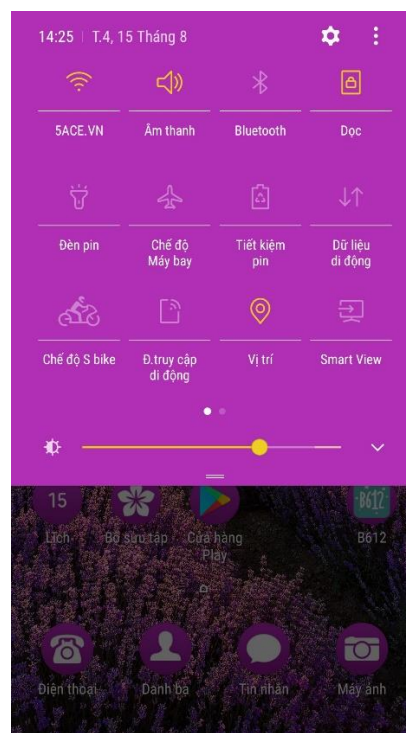
Kỷ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán ra tại Mỹ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng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

## 1.2. Sự phát triển của hệ điều hành Android

**Thanh thông báo vuốt từ trên xuống (Notification bar):** Ngay từ những ngày đầu tiên của Android, thanh thông báo này đã đánh dấu một bước quan trọng mà trước đây chưa hề có hệ điều hành nào làm được - đưa tất cả thông tin tin nhắn, tin thoại hoặc các cuộc gọi nhớ chỉ với thao tác vuốt xuống.



Hình 1.2.1. Phiên bản thời kì đầu



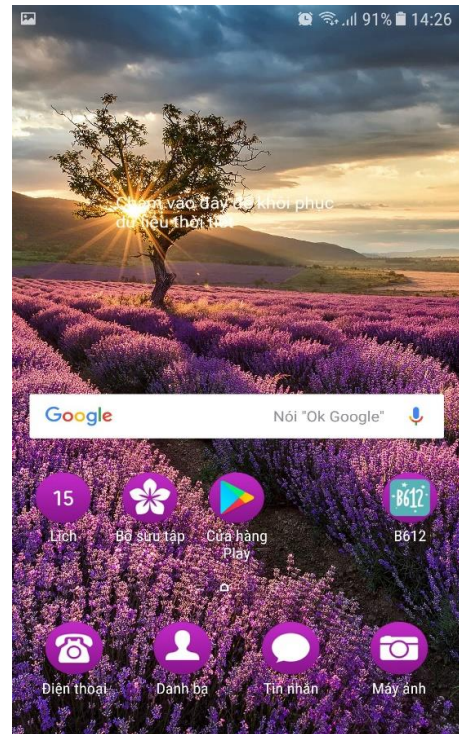
Hình 1.2.2. Android Nougat 7.0

**Màn hình chính (Home Screen) và các widget:** Một điểm khác biệt giữa Android so với các hệ điều hành khác là phần màn hình chính của mình. Bên cạnh việc thay đổi được hình nền, Android còn cho phép người dùng tùy biến màn hình chính của mình với nhiều widgets kèm theo, chẳng hạn như đồng hồ, lịch, trình nghe nhạc, đưa các icon ứng dụng ra ngoài hoặc thậm chí có thể can thiệp sâu hơn để thay đổi toàn bộ giao diện màn hình Home Screen này.



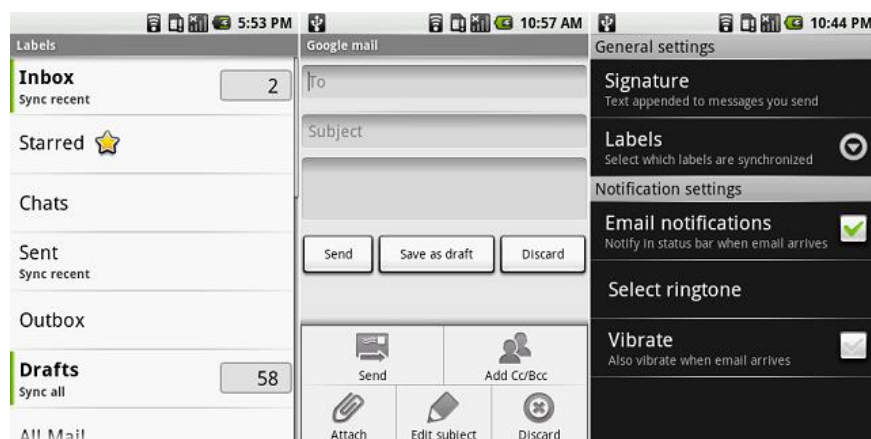


Hình 1.2.3. Màn hình chính Android 1.0



Hình 1.2.4. Màn hình chính Android 7.0

**Đồng bộ và tích hợp chặt chẽ với Gmail:** Vào thời điểm điện thoại G1 được bán ra, Gmail đã hỗ trợ giao thức POP và IMAP để tích hợp với các trình email trên di động. Tuy nhiên, lúc bấy giờ không có bất kì sản phẩm nào có thể hỗ trợ được hoàn toàn những tính năng ưu việt này của Gmail. Mãi cho đến khi Android 1.0 xuất hiện, vấn đề này đã được khắc phục và G1 trở thành chiếc điện thoại mang lại trải nghiệm Gmail tốt nhất trên thị trường lúc bấy giờ.

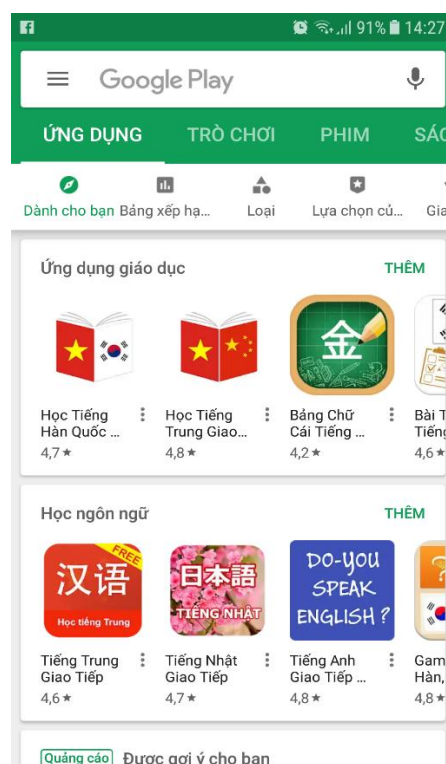


Hình 1.2.5. Gmail trên Android thời kì đầu

**Kho ứng dụng Android:** Thật khó có thể tưởng tượng một chiếc smartphone mà không hề có kho ứng dụng, nhưng vào thời điểm Android mới ra mắt, gần như không có bất kì điện thoại nào có kho ứng dụng nào được tích hợp và chính Android đã mở đầu cuộc cách mạng ứng dụng di động này. Android Market trên G1 thời bấy giờ có rất ít ứng dụng và giao diện cực kỳ đơn giản, hơn nữa tính năng mua ứng dụng trên phiên bản này vẫn chưa được xuất hiện mãi cho đến năm sau - những vấn đề này dễ hiểu vì thời điểm này Android chỉ mới được khai sinh nên mọi thứ còn khá thô sơ.



Hình 1.2.6. Giao diện kho ứng dụng thời kì đầu



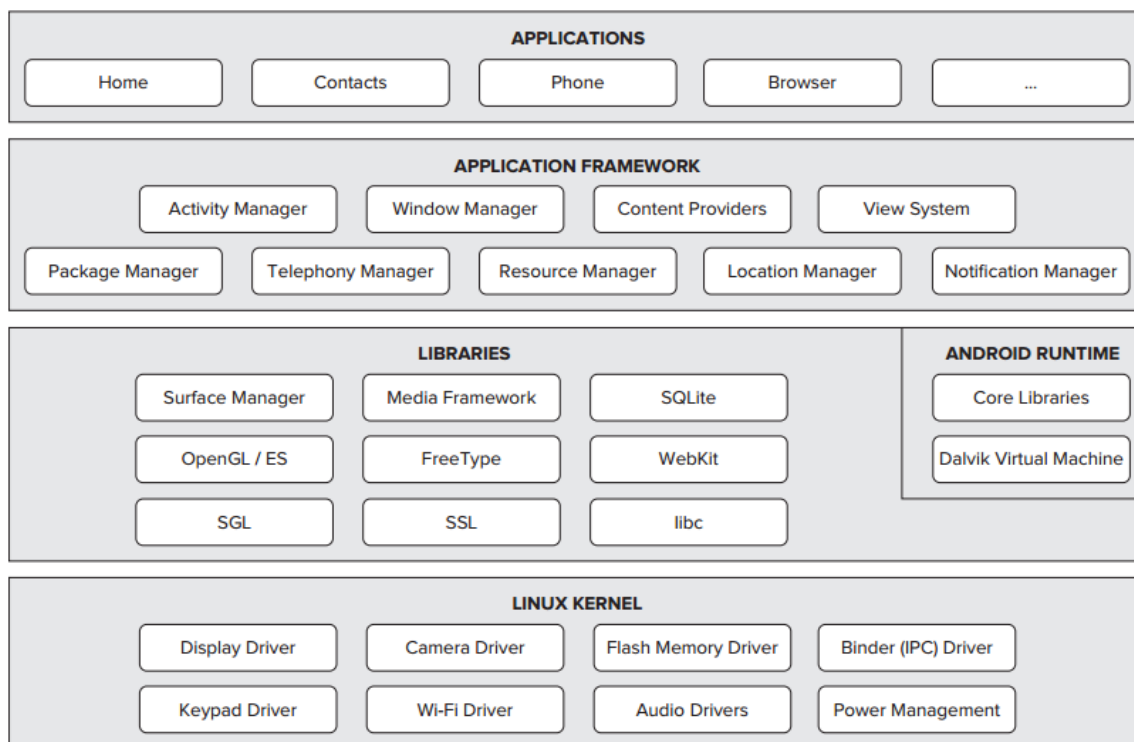
Hình 1.2.7 Giao diện kho ứng dụng Android 7.0

- **Giao diện:** Google đã phát triển giao diện Android phiên bản 1.0 với sự hỗ trợ từ TAT, viết tắt từ The Astonishing Tribe, một công ty thiết kế tương tác của Thụy Điển. Dấu ấn rõ ràng nhất mà TAT để lại trên phiên bản Android từ phiên bản 1.0 cho đến 2.2 chính là widget đồng hồ kim nằm ngoài Home Screen tuy đơn giản

Trường ĐHDL Hải Phòng Khóa luận tốt nghiệp  
 nhưng rất đẹp mắt. Công ty này sau đó ngừng hợp tác với Google và bị RIM mua lại để tập trung phát triển sản phẩm Blackberry cũng như nền tảng BBX sau này.

### 1.3 Kiến trúc của hệ điều hành Android

Android gồm 5 phần chính sau được chứa trong 4 lớp:



Hình 1.3a Cấu trúc hệ thống Android

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.

#### 1.3.1 Thư viện

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:

- Hệ thống thư viện C: một BSD có nguồn gốc từ hệ thống thư viện tiêu chuẩn C (libc), điều chỉnh để nhúng vào các thiết bị dựa trên Linux.

- Thư viện Media - dựa trên PacketVideo's OpenCORE; các thư viện hỗ trợ phát lại và ghi âm của âm thanh phổ biến và các định dạng video, cũng như các tập tin hình ảnh tĩnh, bao gồm cả MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- Bề mặt quản lý - Quản lý việc truy xuất vào hệ thống hiển thị.
- LibWebCore - một công cụ trình duyệt web hiện đại mà quyền hạn cả hai trình duyệt web Android và xem web nhúng.
- SGL - Đồ họa 2D cơ bản của máy.
- Thư viện 3D - một thực hiện dựa vào OpenGL ES 1.0 APIs; các thư viện sử dụng phần cứng tăng tốc 3D (nếu có), tối ưu hóa cao rasterizer phần mềm 3D.
- FreeType- vẽ phông chữ bitmap và vector.

### 1.3.2 Thực thi Android

Android bao gồm một 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 máy ảo Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. Máy ảo là dựa trên register-based, 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.

### 1.3.3 Nền tảng Android

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 để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của và ứng dụng nào khác sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cơ 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 các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) 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 strings, graphics, and layout files.
- Một “Notification Manager” cho phép 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.

#### 1.3.4 Tầng ứng dụng

Là tầng giao tiếp với người dùng, ví dụ: trên thiết bị Android như danh bạ, trình duyệt... mọi ứng dụng viết đều nằm trên tầng này.

### 1.4 Giao diện hệ điều hành 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, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (*icon*) và tiện ích (*widget*); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẫu tin thời sự ngay trên màn hình chính. Màn hình chính có

thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi 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.

## 1.5 Các phiên bản Android

| Phiên bản           | Tên                | Ngày phát hành                 |
|---------------------|--------------------|--------------------------------|
| Android 1.5         | Cupcake            | 27/4/2009                      |
| Android 1.6         | Donut              | 15/9/2009                      |
| Android 2.0 - 2.1   | Eclair             | 26/9/2009 (phát hành lần đầu)  |
| Android 2.2 - 2.2.3 | Froyo              | 20/5/2010 (phát hành lần đầu)  |
| Android 2.3 - 2.3.7 | Gingerbread        | 6/12/2010 (phát hành lần đầu)  |
| Android 3.0 - 3.2.6 | Honeycomb          | 22/2/2011 (phát hành lần đầu)  |
| Android 4.0 - 4.0.4 | Ice Cream Sandwich | 18/10/2011 (phát hành lần đầu) |
| Android 4.1 - 4.3.1 | Jelly Bean         | 9/7/2012 (phát hành lần đầu)   |
| Android 4.4 - 4.4.4 | KitKat             | 31/10/2013 (phát hành lần đầu) |
| Android 5.0 - 5.1.1 | Lollipop           | 12/11/2014 (phát hành lần đầu) |
| Android 6.0 - 6.0.1 | Marshmallow        | 5/10/2015 (phát hành lần đầu)  |
| Android 7.0 - 7.1.2 | Nougat             | 22/8/2016 (phát hành lần đầu)  |
| Android 8.0 - 8.1   | Oreo               | 21/8/2017 (phát hành lần đầu)  |

Hình 1.5a Lịch sử phát triển các phiên bản của android

### Phiên bản Android 1.5

Android 1.5 có lẽ có vai trò cực kì quan trọng trong quá trình trưởng thành của Android khi nó bổ sung cho hệ điều hành này những tính năng nổi bật giúp nó cạnh tranh với các nền tảng đối thủ khác. Đây cũng là bản Android đầu tiên được Google gọi tên theo các món đồ ăn với chữ cái bắt đầu được xếp theo thứ tự alphabet. Về mặt giao diện, Android 1.5 không có nhiều điểm thay đổi so với người tiền nhiệm của mình. Google chỉ điểm thêm vài điểm để làm giao diện trông bóng bẩy, mượt mà hơn một tí, chẳng hạn như widget tìm kiếm có độ trong suốt nhẹ, biểu tượng app drawer có một số hoa văn nhỏ mới, v.v. Nói chung, giao diện không phải là một điểm nhấn của Android 1.5 mà người ta quan tâm nhiều hơn đến các tính năng mới mà nó mang lại.

### Phiên bản Android 1.6

Phiên bản bánh Donut này, mặc dù chỉ thêm có 0.1 vào mã số của Android 1.5 nhưng nó cũng mang lại nhiều cải tiến đáng giá. Một vài điểm trong giao diện được cải thiện, vài tính năng nhỏ được thêm vào, cuối cùng là hỗ trợ cho mạng CDMA. Bổ

sung tính năng có thể chạy trên nhiều độ phân giải và tỉ lệ màn hình khác nhau, cho phép những thiết bị có nhiều độ phân giải hơn là 320 x 480. Hiện nay, chúng ta có những chiếc smartphone Android chạy ở độ phân giải QVGA, HVGA, WVGA, FWVGA, qHD, và 720p. Vài chiếc máy tính bảng còn đạt mức 1920 x 1080 nữa.

### **Phiên bản Android 2.0 - 2.1**

Đầu tháng 9 năm 2009, Android 2.0 Eclair đã được ra mắt trên chiếc Motorola Droid cùng các tính năng mới:

Hỗ trợ nhiều tài khoản người dùng: lần đầu tiên, nhiều tài khoản Google có thể được đăng nhập trên cùng một thiết bị chạy Android. Tài khoản Microsoft Exchange cũng được hỗ trợ trong Eclair. Người dùng có thể duyệt qua danh bạ, email của từng tài khoản. Google giờ đây cho phép những nhà phát triển bên thứ ba tích hợp dịch vụ của họ vào trong mục Account này, đồng thời hỗ trợ tự động đồng bộ hóa.

**Quick Contact:** khi chạm vào một số liên lạc nào đó, có một menu nhỏ sẽ xuất hiện để bạn tương tác theo nhiều cách: gửi email, gọi điện, nhắn tin và hơn thế nữa. Miễn là nơi nào có biểu tượng contact xuất hiện là nơi đó có thể dùng Quick Contact. Sau này có thêm Twitter, Facebook và nhiều dịch vụ đồng bộ khác cũng tích hợp tính năng riêng của mình vào Quick Contact.

**Cải tiến bàn phím ảo:** Giống với chiếc G1, Droid/Milestone có một bàn phím QWERTY đầy đủ dạng trượt ngang nhưng Google cho phép người dùng sử dụng thêm bàn phím ảo mà hãng đã thiết kế lại. Mặc dù tính năng cảm ứng vẫn còn một điều xa vời với Android vào lúc đó (trình duyệt, bản đồ và cả các phần mềm khác không hề dùng được tính năng hai ngón tay để phóng to, thu nhỏ,...) nhưng Eclair lại sử dụng dữ liệu multitouch từ bàn

phím để xác định điểm chạm thứ hai trong lúc người dùng gõ nhanh. Độ chính xác của bàn phím ảo trên Android 2.0 nhờ đó đã được cải tiến đáng kể.

**Trình duyệt mới:** Mặc dù chưa hỗ trợ cảm ứng đa điểm nhưng trình duyệt trên Android 2.0 cũng có nhiều điểm nâng cấp đáng khen. Google đã hỗ trợ HTML5, bao gồm luôn khả năng phát video ở chế độ toàn màn hình. Hộp địa chỉ kết hợp với thanh tìm kiếm (giống với trình duyệt Chrome) cũng lần đầu xuất hiện trên Android. Để bù



lại cho tính năng cảm ứng đa điểm, trình duyệt mới hỗ trợ chạm hai lần để phóng to nội dung trên màn hình, kèm theo đó là hai nút Zoom in, Zoom out.

**Giao diện mới:** các biểu tượng giờ đây đã đẹp hơn, sang trọng hơn, gọn gàng hơn nhiều so với trước. Widget cũng được thiết kế mới để tương thích với độ phân giải cao trên Droid. Giao diện menu cũng đẹp hơn nữa.

Android 2.1 ra mắt chủ yếu để sửa lỗi và thêm hàm API để lập trình viên can thiệp sâu hơn vào hệ thống nhưng nó đã hỗ trợ thêm vài tính năng lý thú như Live Wallpaper, chuyển giọng nói thành văn bản và một màn hình khóa mới. Android 2.1 được đánh dấu bởi sự ra đời của chiếc Nexus One do HTC sản xuất, chiếc điện thoại Nexus đầu tiên chạy Android chuẩn của Google.

### **Phiên bản Android 2.2 - 2.2.3**

Android 2.2 được ra mắt trong năm 2010. Nexus One là chiếc điện thoại đầu tiên được nâng cấp lên Android 2.2, sớm hơn nhiều so với tất cả các hãng khác. Giao diện màn hình chính đã được thay đổi, từ 3 màn hình chính tăng lên thành 5 màn hình. Google đã có nhiều cố gắng để giao diện Android được vui và đẹp hơn, dễ dùng hơn, bắt kịp phân nào với giao diện của bên thứ ba như HTC Sense chẳng hạn.

Froyo còn có một trình xem ảnh mới với khả năng hiển thị hình ảnh 3D: nghiêng máy thì dãy ảnh sẽ nghiêng theo, đồng thời nó mang trong mình nhiều hiệu ứng chuyển động đẹp mắt khác.

Một trong những thay đổi nổi bật ở phiên bản Froyo là tính năng USB tethering và Wi-Fi hotspot. Các thay đổi khác là hỗ trợ dịch vụ Cloud to Device Messaging (C2DM), cho phép đẩy thông báo. Ngoài ra còn cải thiện tốc độ ứng dụng, được hiện thực thông qua biên dịch JIT. Nhiều tính năng bảo mật mới cũng xuất hiện.

### **Phiên bản Android 2.3- 2.3.7**

Khoảng nửa năm sau khi Froyo xuất hiện, Google đã trở lại với bản Android 2.3 Gingerbread. Google giới thiệu nó với nhiều tính năng mới, tập trung vào việc phát triển game, đa phương tiện và phương thức truyền thông mới. Bên cạnh đó, chiếc Nexus S do Samsung sản xuất cũng xuất hiện với vai trò là người kế nhiệm cho Nexus One. Nexus S có vài thay đổi nhỏ so với người anh em Galaxy S.

Giao diện Gingerbread được tinh chỉnh rất nhiều, giúp người dùng làm chủ dễ dàng hơn, dễ sử dụng hơn, và tiết kiệm năng lượng hơn. Một bộ màu đơn giản với màu nền đen khiến cho thanh thông báo, menu và các thành phần giao diện khác nổi bật lên. Những cải thiện trong menu và thiết lập giúp người dùng điều hướng dễ dàng hơn.

Gingerbread sử dụng nhân Linux phiên bản 2.6.35.

#### **Phiên bản Android 3.0 – 3.2.6**

Honeycomb là phiên bản Android dành riêng cho máy tính bảng, và sản phẩm đầu tiên dùng hệ điều hành này Motorola Xoom. Mặc dù Android 3.0 không có nhiều dấu ấn đặc biệt trên thị trường nhưng nó là nền tảng cho Android 4.0 với các tính năng như. Ngoài ra, Honeycomb còn hỗ trợ cho việc bố cục ứng dụng theo nhiều cột để hướng đến việc hỗ trợ máy tính bảng tốt hơn.

#### **Phiên bản Android 4.0 - 4.0.4**

Cuối năm 2011, Google chính thức giới thiệu điện thoại Galaxy Nexus, thiết bị đầu tiên trên thị trường sử dụng Android 4.0 Ice Cream Sandwich. Có thể nói Android 4.0 là sự thay đổi lớn nhất trong lịch sử phát triển của Android tính đến ngày viết bài này. Android 4.0 hỗ trợ một bộ font mới tên là Roboto được cho là tối ưu hóa để dùng trên các màn hình độ phân giải càng ngày càng cao hơn, đồng thời để hiển thị được nhiều thông tin hơn trên màn hình.

Hệ thống thông báo (Notification) già nua của Android đã được làm mới hoàn toàn, đẹp hơn, tiện dụng hơn, nhất là tính năng trượt ngang để xóa từng thông báo riêng lẻ. Tương tự như vậy cho tính năng Recent Apps và cả trình duyệt của máy. Bàn phím cũng được làm mới với khả năng tự động sửa lỗi cao hơn, việc sao chép, cắt dán chữ và nội dung cũng tốt hơn bao giờ hết.

Đây cũng là lần đầu tiên Google hợp nhất hệ điều hành dành cho smartphone và cho máy tính bảng vào làm một.

Android 4.0 nhận được nhiều đánh giá tích cực từ các nhà phê bình, họ khen ngợi giao diện hệ điều hành được tân trang, gọn gàng hơn so với các phiên bản trước, cùng với các cải thiện về hiệu suất và tính năng. Tuy nhiên, các nhà phê bình vẫn cảm thấy các ứng dụng gốc của Android 4.0 vẫn còn thua kém về chất lượng và tính năng so với

các ứng dụng tương đương của bên thứ ba, và một số tính năng mới của hệ điều hành, đặc biệt là "mở khóa bằng khuôn mặt", không cần thiết lắm.

### **Android 4.1 - 4.3.1**

Android Jelly Bean đánh dấu sự ra đời với chiếc máy tính bảng Nexus 7 do Asus sản xuất. Về giao diện, không có nhiều thay đổi so với Android 4.0, vẫn là màn hình chính với thanh dock bên dưới quen thuộc. Sự xuất hiện của Google Now cho thấy rằng Google đã bắt đầu bước chân vào việc cạnh tranh với Siri.

Trên Android 4.1 bạn có thể tìm kiếm bằng giọng nói và các kết quả trả không chỉ đơn giản là những dòng tìm kiếm nữa mà nó được thiết kế theo dạng thẻ đồ họa, thông minh hơn, trực quan hơn. Không chỉ tìm kiếm theo yêu cầu. Nhưng có lẽ quan trọng hơn hết của Jelly Bean không phải là về giao diện hay ứng dụng mới mà về Project Butter giúp mang lại độ mượt chưa từng có cho Android.

Ngày 30/10/2012, Google chính thức tuyên bố cập nhật hệ điều hành Android của hãng lên phiên bản 4.2 và vẫn giữ nguyên tên gọi "Jelly Bean" (có nghĩa là kẹo dẻo đậu). Được hãng gọi là "một hương vị mới của Jelly Bean", Android 4.2 mang trong mình nhiều tính năng mới như hỗ trợ Miracast, bàn phím có thể nhập liệu bằng cách vẽ các đường nét từ kí tự này đến kí tự khác, chế độ chụp ảnh toàn cảnh Photo Sphere, ứng dụng Gmail mới và còn rất nhiều thứ khác nữa.

Một cải tiến lớn và quan trọng của Android 4.2 đó là việc hỗ trợ nhiều tài khoản người dùng trên máy tính bảng để có thể dễ dàng chia sẻ trong gia đình hoặc nơi làm việc. Mỗi tài khoản sẽ có dữ liệu app của riêng họ. Ví dụ, người dùng A cài game Angry Birds, người này chơi và đã ghi được một số điểm cũng như lên được màn cao hơn. Khi người B sử dụng máy bằng tài khoản khác, game này vẫn hiện diện trên máy nhưng dưới dạng một bản cài mới, không có sẵn data của người A.

Lại thêm một thế hệ Jelly Bean nữa và lần này là Android 4.3. Ngày 24/7/2013, Google đã chính thức ra mắt hệ điều hành này song song với chiếc Nexus 7 (2013). Đây là phiên bản Android mới nhất đang có mặt trên thị trường và đi kèm những tính năng mới như hỗ trợ kết nối Bluetooth Smart, bộ API OpenGL ES 3.0, bổ sung tính năng sử dụng Wi-Fi để định vị ngay cả khi người dùng tắt kết nối này đi cùng nhiều thay đổi lớn nhỏ khác. Bên cạnh tính năng Multi User của Android 4.2 như đã nói ở

trên, Google bổ sung thêm một tính năng mới cho Android 4.3 đó là Restricted Profile. Mỗi thành viên sẽ có một "hồ sơ" riêng của mình và chỉ có thể làm được những gì được chỉ định trong "hồ sơ". Ví dụ, cha mẹ có thể khóa tính năng in-app purchase của một ứng dụng nào đó để ngăn con trẻ vô tình mua hàng trăm USD. Ngoài ra, khi áp dụng vào môi trường doanh nghiệp thì nó sẽ cho phép người quản trị phân quyền cho nhân viên dễ dàng hơn.

#### **Phiên bản Android 4.4 – 4.4.4**

Google tiết lộ thêm rằng "mục tiêu của chúng tôi với Android KitKat đó là mang trải nghiệm Android đáng kinh ngạc đến cho mọi người". Android Kitkat xuất hiện đầu tiên trên chiếc smartphone Nexus 5. Những cải tiến trên Android Kitkat bao gồm Chế độ toàn màn hình – Immersive Mode, Hiệu ứng chuyển cảnh màn hình - Transition Manager, Storage Access Framework, Chromium WebView, NFC, Cổng hồng ngoại – Infrared Blasters ...

Giao diện: Tông màu chủ đạo của font chữ là xanh và đen được sử dụng trên Android Jelly Bean được Google thay đổi bằng tông màu đen và ghi trên Android 4.4 Kitkat vừa ra mắt, thể hiện rõ ở các dòng chữ trên thanh thông báo Notification. Bên cạnh đó, Android Kitkat cũng có bộ biểu tượng, folder mới, giao diện nhìn chung được làm phẳng bớt, loại bỏ các chi tiết đồ bóng và thêm hiệu ứng trong suốt vào nhiều phần, như ba phim ảo bên dưới hay thanh thông báo ở phía trên.

Dù là phiên bản mới, Android 4.4 Kitkat lại không đòi hỏi cấu hình phần cứng mạnh hơn, thậm chí hỗ trợ tốt cả những thiết bị cũ với phần cứng không cao, như có RAM chỉ đạt dung lượng 512 MB. Hệ điều hành mới được Google tối ưu khả năng hoạt động, cho hiệu năng cao hơn tới 1,6 lần phiên bản trước.

#### **Phiên bản Android 5.0 – 5.1.1**

Lollipop sau hơn một năm Android 4.0 KitKat ra mắt, Google đã chính thức trình làng phiên bản hậu duệ mới nhất với tên gọi Android Lollipop, được đánh giá là hệ điều hành có những thay đổi rõ rệt và đáng kể nhất trong lịch sử Android từ trước đến nay.

Một trong những thay đổi lớn nhất trong Lollipop là được thiết kế lại giao diện người dùng được xây dựng dựa trên ngôn ngữ thiết kế gọi là Material Design. Những

thay đổi khác bao gồm cải tiến thanh thông báo, có thể truy cập từ màn hình khóa và hiển thị trong ứng dụng như banner trên màn hình. Google còn thay đổi bên trong nền tảng này, với Android Runtime (ART) chính thức thay thế cho Dalvik để cải thiện hiệu năng ứng dụng, và tối ưu hóa việc sử dụng pin, gọi nội bộ là Project Volta.

### **Phiên bản Android 6.0 - 6.0.1**

Vào ngày 5/10/2015 thì Google đã giới thiệu đến người dùng phiên bản Android kế tiếp với mã hiệu Marshmallow (kẹo dẻo) với những thay đổi đột cập nhật này tuy nhỏ nhưng thiên về xu hướng hoàn thiện tương tác người dùng, cho một trải nghiệm tuyệt vời hơn.

Về mặt giao diện người dùng, màn hình chính trên Android M có vẻ có khá ít thay đổi so với phiên bản trước ngoại trừ logo Google, tuy nhiên khi bạn vào trong Menu ứng dụng bạn có thể thấy khá nhiều thay đổi. Các ứng dụng được bố trí và cuộn xuống theo chiều dọc, ở phía trên cùng là thanh tìm kiếm và có một điểm thú vị là máy sẽ tự động gợi ý cho bạn 4 apps để sử dụng tùy theo từng địa điểm mà bạn đang ở.

**Tính năng mới Now on Tap:** khi ở màn hình của bất kỳ ứng dụng (ví dụ trình duyệt web) đang hiển thị thông tin, bạn có thể giữ nút home và máy sẽ bắt đầu "quét" những thông tin đang được hiển thị trên màn hình, gửi về Google và Google sẽ phản hồi lại cho bạn những thông tin, gợi ý có ích liên quan. Việc tích hợp Google theo cách này nâng trải nghiệm người dùng lên một tầm cao mới.

**Tiết kiệm năng lượng hơn:** Khi ở trạng thái không sử dụng, điện thoại Android đốt pin khá nhiều nếu so sánh với các điện thoại hệ điều hành khác. Google đã cho ra đời chế độ Doze để giải quyết vấn đề về pin chờ điện thoại này. Khi máy bạn không được sử dụng, chế độ Doze sẽ được kích hoạt và sẽ hoạt động hơi giống như khi bạn bật chế độ Airplane trên điện thoại vậy: không kết nối, không thông báo, không có bất kỳ việc chạy ứng dụng nào diễn ra trong quá trình này để đảm bảo cho máy ở trạng thái tiết kiệm năng lượng tối đa có thể. Chỉ một số thông báo quan trọng như cuộc gọi, báo thức, tin nhắn là có thể hiển thị trong trạng thái Doze này.

Ngoài ra Android 6. cũng mang lại cho người dùng một số tính năng mới như: sử dụng thẻ SD cắm ngoài như bộ nhớ trong hay chính thức hỗ trợ nhận diện vân tay và Google Pay.

### **Phiên bản Android 7.0 – 7.1.2**

Android 7.0 Nougat được Google tung ra vào ngày 22 tháng 8 năm 2016 và những thiết bị Nexus sẽ là những chiếc smartphone đầu tiên được cập nhật phiên bản mới

này. Tên gọi chính thức của phiên bản Android 7.0 đã được Google xác nhận, đó chính là một loại kẹo truyền thống có tên Nougat.

Android 7.0 hỗ trợ một số tính năng mới đáng kể như:

**Hỗ trợ Menu chuyển đổi nhanh giữa các cài đặt hệ thống:** Một menu hamburger (biểu tượng dấu 3 gạch nằm ngang) đã được thêm vào Cài đặt của hệ thống. Ví dụ như khi bạn đang trong phần cài đặt Bluetooth bạn có thể nhanh chóng chuyển đến các cài đặt khác của hệ thống bằng cách sử dụng Menu bên trái này.

**Trả lời nhanh tin nhắn từ thanh thông báo:** Trong những bổ sung mới, thì đây là tính năng mới mẻ và khá giống với hệ điều hành iOS của Apple. Tin nhắn có thể được trả lời nhanh từ thông báo của chính nó. Với tính năng này người dùng không cần phải rời khỏi ứng dụng hiện tại để trả lời một tin nhắn hoặc thậm chí là mở khóa điện thoại.

**Chế độ chia đôi màn hình:** Cuối cùng thì Google đã thêm tính năng vào Android 7.0, đây là tính năng mà bạn thường thấy trên các máy tính bảng, các dòng điện thoại của Samsung hay LG. Người dùng có thể chia màn hình thành 2 phần, sử dụng 2 ứng dụng song song cùng 1 lúc. Và bạn còn có thể tùy chọn kích thước cửa sổ cho mỗi ứng dụng bằng cách kéo thanh màu đen giữa 2 ứng dụng sang trái hoặc phải, lên hoặc xuống.

**Trung tâm thông báo được làm mới:** Trung tâm thông báo đã được google thiết kế lại. Các biểu tượng, phím tắt trên thanh thông báo được thu gọn và nổi bật hơn, bạn có thể mở rộng các phím tắt bằng phím mũi tên phía bên trái góc trên màn hình. Các thông báo chung từ 1 ứng dụng bây giờ đã được tối ưu gộp lại với nhau thành 1 dòng duy nhất và bạn có thể mở rộng bằng cách sử dụng nút mũi tên hoặc thao tác với 2 ngón tay. Giúp tối ưu hóa không gian trong trung tâm thông báo.

Ngoài ra còn một số tính năng đáng chú ý như: hỗ trợ chế độ thực tế ảo với VR, tích hợp chế độ tiết kiệm dữ liệu, chế độ tiết kiệm pin Doze được cải tiến.

### **Phiên bản Android 8.0 - 8.1**

Oreo" (tên mã phát triển là Android O) là phiên bản lớn thứ tám của hệ điều hành di động Android. Nó được phát hành lần đầu dưới dạng một phiên bản alpha xem trước cho nhà phát triển vào ngày 21 tháng 3 năm 2017. Bản xem trước thứ hai được phát hành ngày 17 tháng 5 năm 2017, được coi là phiên bản beta, và phiên bản xem trước thứ ba được phát hành ngày 8 tháng 6 năm 2017 với phần API được hoàn thiện.

Vào ngày 24 tháng 7 năm 2017, một bản xem trước thứ tư được phát hành bao gồm những tính năng hệ thống cuối cùng cùng với những sửa lỗi và cải tiến mới nhất. Oreo được chính thức phát hành công khai vào ngày 21 tháng 8 năm 2017. Sau đó là Android 8.1 vào 5/12/2017. Oreo có một số cập nhật mới, nhiều điểm cải tiến so với bản tiền nhiệm như sideload (cài ứng dụng không có trên Store), giới hạn dữ liệu của ứng dụng nền, thông báo cho các ứng dụng chạy ở chế độ nền, Picture-in-picture, tự động điền mật khẩu trong ứng dụng, cải thiện thời lượng pin, Project Treble...

## CHƯƠNG 2 : MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO

### 2.1. Giới thiệu Android Studio

Android Studio là một nền tảng IDE (Integrated Development Environment) dùng để phát triển các ứng dụng android, được Google release vào khoảng đầu năm 2015 thay thế cho bản Eclipse cũ. Android Studio được phát triển dựa trên IntelliJ IDEA Community Edition - công cụ lập trình tốt nhất cho java, giúp cho các lập trình viên tạo ứng dụng, thực hiện các thay đổi một cách dễ dàng, bên cạnh đó có thể xem trước trong thời gian thực và thiết kế giao diện đẹp hơn trước. Tiếng Việt cũng đã được tích hợp trong Android Studio. Đặc biệt, Android Studio cho phép người dùng Import Project từ Eclipse sang và logic lập trình cũng tương tự.

### 2.2. Cài đặt môi trường

#### 2.2.1 Cài đặt JAVA JDK

❖ *Bước 1: 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 điều hành đúng với máy đang sử dụng.

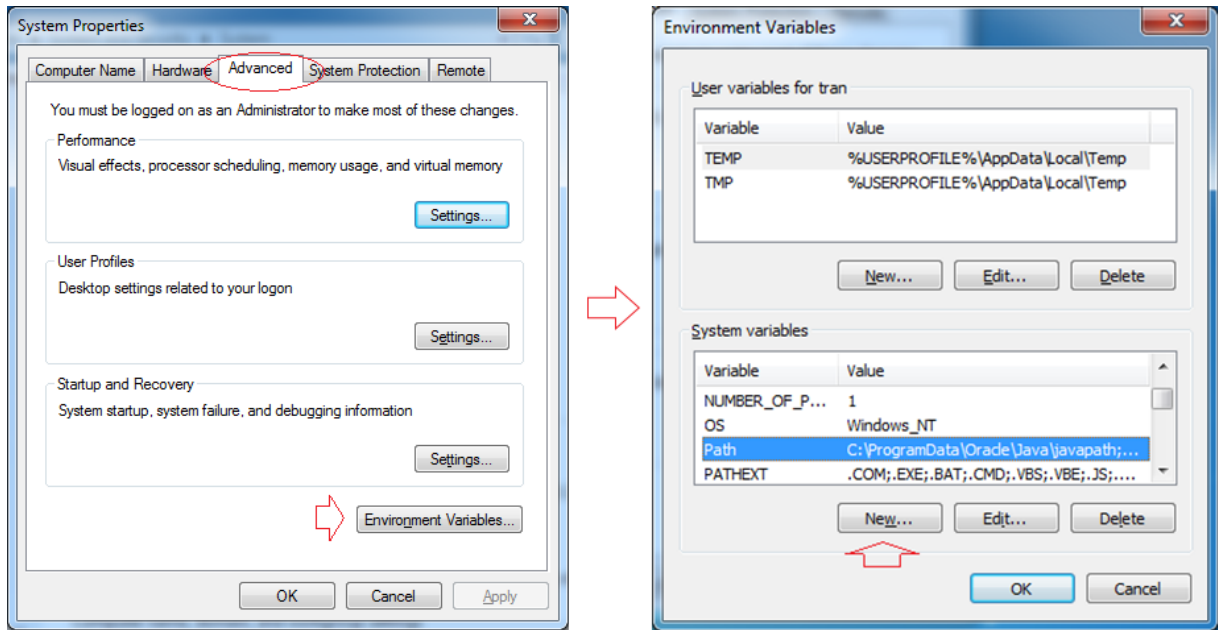
❖ *Bước 2: Mở file cài đặt “jdk-\*.exe” để tiến hành cài đặt*

❖ *Bước 3: Cấu hình biến môi trường cho Java*

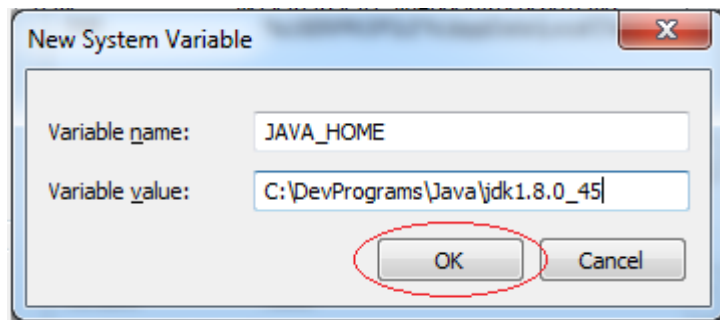
Việc này không bắt buộc, nhưng nếu trên máy tính của cài đặt nhiều phiên bản Java, thì việc cấu hình là cần thiết để xác định phiên bản java nào mặc định được sử dụng.

Trên Desktop, nhấn phải chuột vào Computer, chọn **Properties > Advanced system settings**.

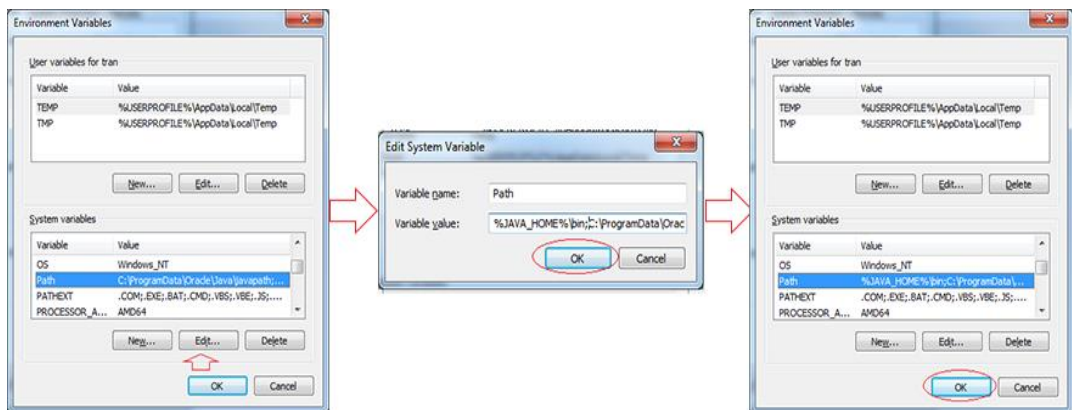




Hình 2.2.1.1 Nhập vào đường dẫn tới thư mục JDK.



Hình 2.2.1.2 Tiếp theo đổi tên path là đã cài đặt xong Java JDK



## 2.2.2 Cài đặt Android Studio

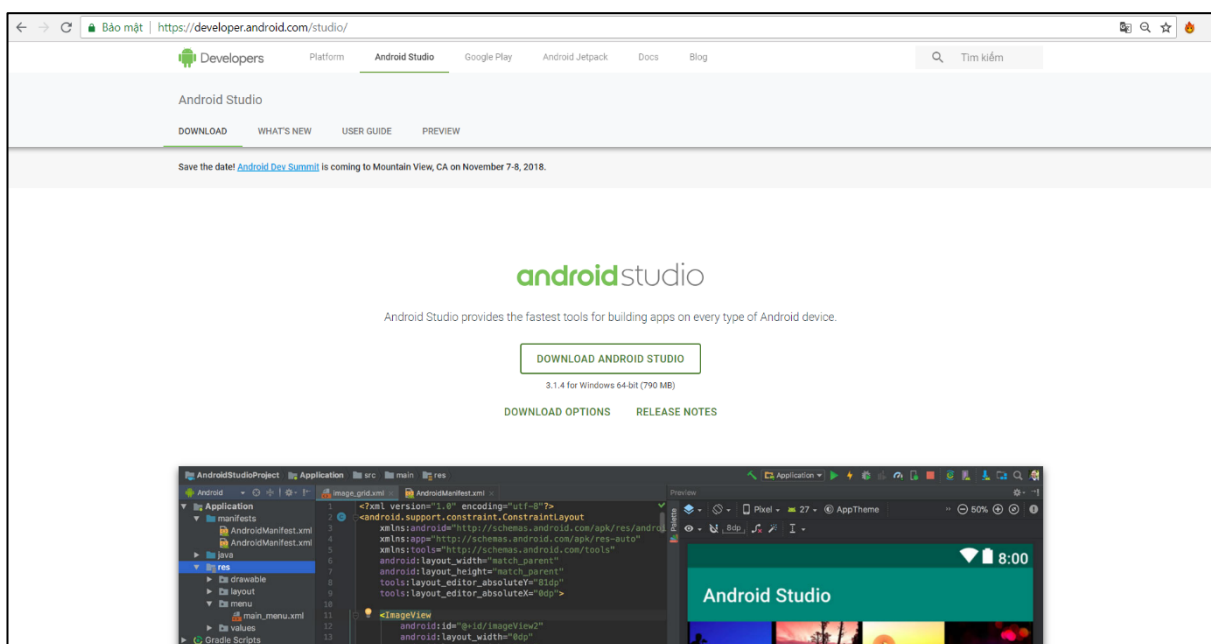
Cấu hình yêu cầu:

- Microsoft® Windows® 7/8/10 ( 32 bit hoặc 64 bit )

- Tối thiểu 4 GB RAM, cấu hình đề nghị: 8 GB RAM
- Tối thiểu 40GB dung lượng đĩa trống, 100GB được đề xuất.
- Độ phân giải tối thiểu 1366 x 768
- Java Development Kit (JDK) 7 trở lên
- Lựa chọn thêm cho accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

Tài file cài đặt

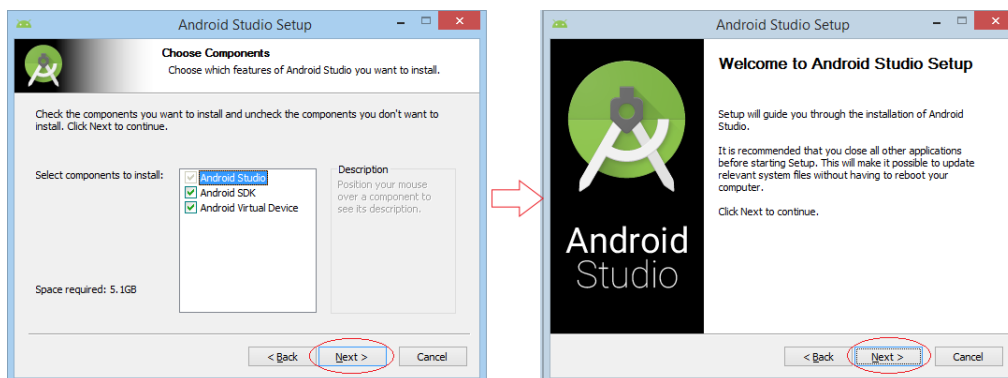
Download file cài đặt: <https://developer.android.com/studio/>



Hình 2.2.2.1 Trang tải Android Studio

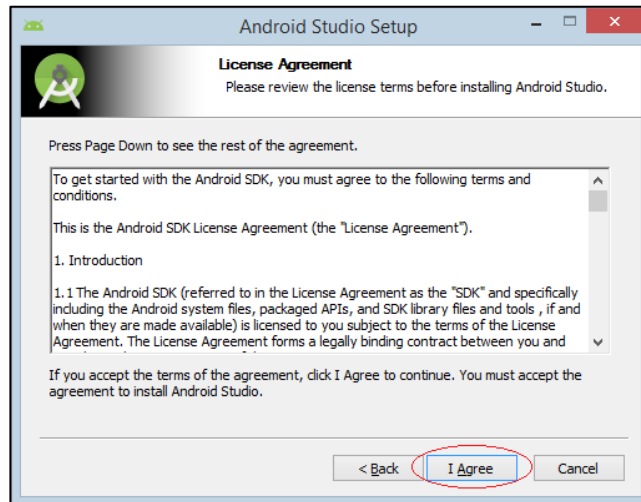
Cách cài đặt :

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

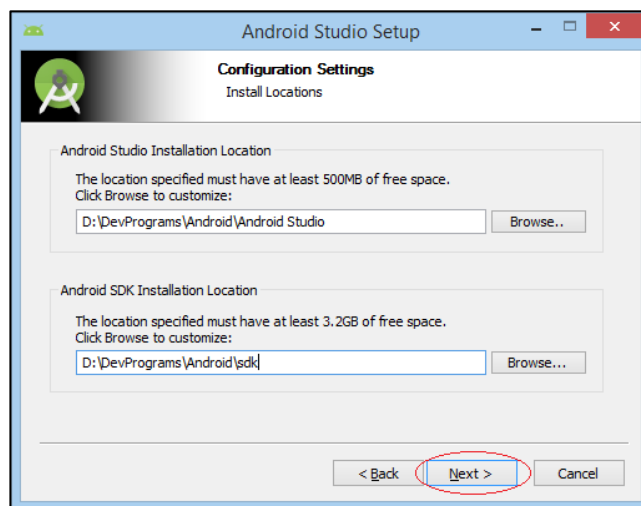




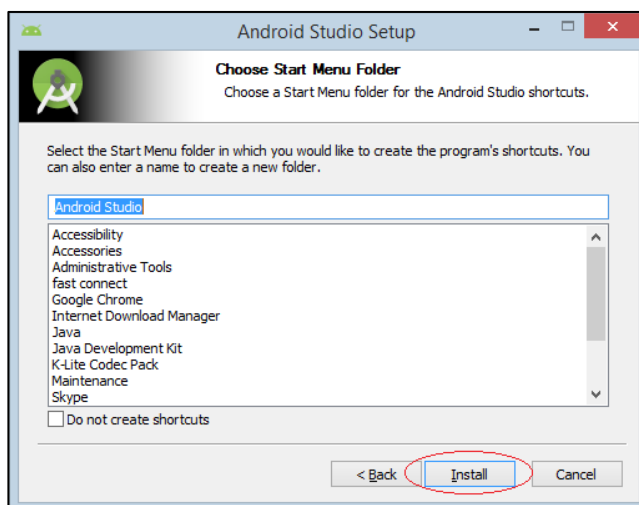
❖ *Bước 2: Chọn “I Agree” để xác nhận, đồng ý với các điều khoản và tiếp tục.*



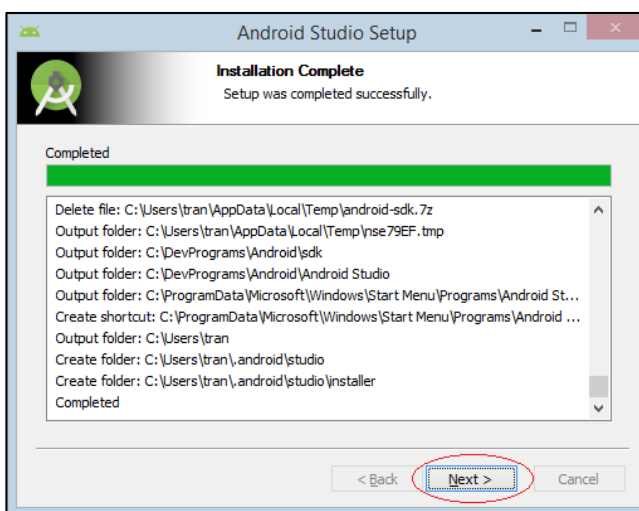
❖ *Bước 3: Chọn nơi cài đặt Android Studio và Android SDK*



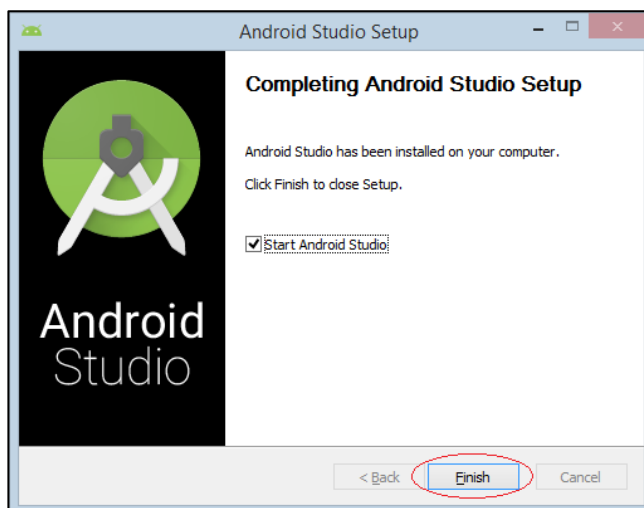
❖ *Bước 4: Chọn Start Menu Folder. Ở bước này, có thể tích tạo shortcuts cho ứng dụng hoặc không Bấm Install để bắt đầu cài đặt.*



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



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



### 2.2.3 Phần mềm Genymotion

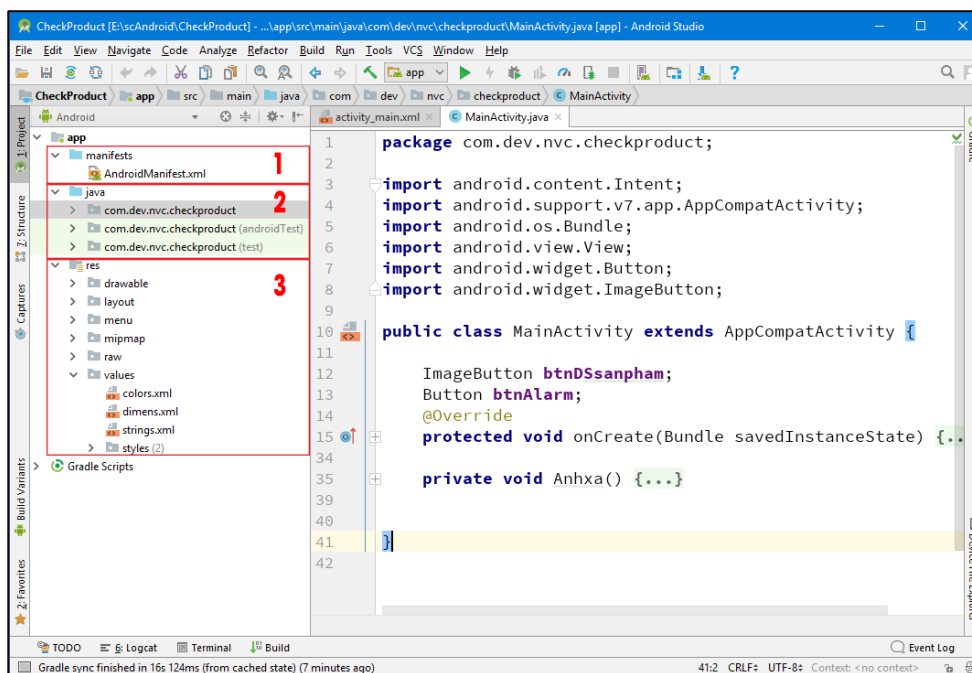
Yêu cầu:

- CPU hỗ trợ VT-x hoặc AMD-V và kích hoạt thiết lập BIOS.
- RAM: Tối thiểu 1 GB.
- Dung lượng trống của ổ cứng: Ít nhất là 4GB để cài đặt Genymotion và các máy ảo chạy Genymotion (đây chỉ là mức yêu cầu tối thiểu, bởi nếu sử dụng nhiều máy ảo cùng lúc và có nhiều ứng dụng, phần mềm cài đặt thì dung lượng trống có thể được yêu cầu nhiều hơn gấp 4 lần).
- Đảm bảo có kết nối Internet
- Độ phân giải màn hình: ít nhất 1024 x 768 pixel
- Oracle **VirtualBox 4.1** trở lên.
- Ngoài ra, người dùng cần có một tài khoản Genymotion để có thể sử dụng.

Tải file cài đặt tại đường dẫn: <https://www.genymotion.com/fun-zone/>

Cách cài đặt:

- Bước 1: Tìm tới vị trí lưu file vừa tải về đầu tiên, sau đó click đúp chuột trái vào file đó để bắt đầu tiến hành. Giao diện đầu tiên sẽ cho lựa chọn ngôn ngữ để sử dụng.
- Bước 2: Tùy theo ý muốn người dùng mà có thể **Next** ngay và cài đặt Genymotion ngay tại vị trí được gợi ý hoặc chọn **Browse** để tìm vị trí khác.



Hình 2.2.3.1. Giao diện của một Project

### 2.3.1 Tập cấu hình dự án Android<sup>1</sup>

#### a. Giới thiệu:

Trong bất kì một Project Android nào khi tạo ra đều có một file AndroidManifest.xml, file này được dùng để định nghĩa các screen sử dụng, các permission cũng như các theme cho ứng dụng. Đồng thời nó cũng chứa thông tin về phiên bản SDK cũng như main activity sẽ chạy đầu tiên. File này được tự động sinh ra khi tạo một Android project. Trong file manifest bao giờ cũng có 3 thành phần chính đó là: application, permission và version. Hay nói cách khác đây là file dùng để config những thuộc tính cho ứng dụng của mà khi ứng dụng khởi chạy hệ điều hành có thể hiểu được và xử lí.

#### b. Chức năng AndroidManifest

- ✓ Nó đặt tên gói Java cho ứng dụng. Tên gói đóng vai trò như một mã nhận diện duy nhất cho ứng dụng.
- ✓ Nó mô tả các thành phần của ứng dụng - hoạt động, dịch vụ, hàm nhận quảng bá, và trình cung cấp nội dung mà ứng dụng được soạn bởi. Nó đặt tên các lớp triển khai từng thành phần và công bố các khả năng của chúng (ví dụ, những tin nhắn

<sup>1</sup> Android Manifest

Intent mà chúng có thể xử lý). Những khai báo này cho phép hệ thống Android biết các thành phần là gì và chúng có thể được khởi chạy trong những điều kiện nào.

- ✓ Nó xác định những tiến trình nào sẽ lưu trữ các thành phần ứng dụng.
- ✓ Nó khai báo các quyền mà ứng dụng phải có để truy cập các phần được bảo vệ của API và tương tác với các ứng dụng khác.
- ✓ Nó cũng khai báo các quyền mà ứng dụng khác phải có để tương tác với các thành phần của ứng dụng.
- ✓ Nó liệt kê các lớp Instrumentation cung cấp tính năng tạo hồ sơ và các thông tin khác khi ứng dụng đang chạy. Những khai báo này chỉ xuất hiện trong bản kê khai khi ứng dụng đang được phát triển và thử nghiệm; chúng bị loại bỏ trước khi ứng dụng được công bố.
- ✓ Nó khai báo mức tối thiểu của API Android mà ứng dụng yêu cầu. Nó liệt kê các thư viện mà ứng dụng phải được liên kết với.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
  <uses-permission /> <permission
/> <permission-tree /> <permission-group />
  <instrumentation />
  <uses-sdk /> <uses-configuration /> <uses-
feature />
  <supports-screens />
  <compatible-screens />
  <supports-gl-texture />
  <application>
    <activity>
      <intent-filter> <action /> <category />
<data /> </intent-filter>
      <meta-data />
    </activity>
    <activity-alias>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </activity-alias>
    <service>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </service>
    <receiver>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </receiver>
    <provider>
      <grant-uri-permission />
      <meta-data />
      <path-permission />
    </provider>
    <uses-library />
  </application>
</manifest>
```



- Cấu trúc file AndroidManifest.xml

**<uses-permission/>** : đây là cú pháp lệnh để xin cấp một quyền gì đó khi ứng dụng tương tác với dữ liệu nào đó mà google không cho phép dùng tùy ý.

**<activity>...</activity>**: đây là thẻ để khai báo các activity trong ứng dụng và các thuộc tính của activity đó, nếu như không khai báo thì khi khởi chạy ứng dụng sẽ lỗi ngay.

**<service>...</service>**: đây là một dịch vụ chạy ngầm trên ứng dụng của kẻ cả khi ứng dụng đã tắt đi và khi sử dụng phải khai báo chúng trong AndroidMainifest.

**<receiver>...</receiver>**: đây là khi báo khi sử dụng Broadcast Reciver, cái này là để lắng nghe các sự kiện thay đổi của hệ thống như khi bắt wifi, khi sạc pin...

**<uses-sdk >..</ uses-sdk >**: Thẻ xác định phiên bản SDK

### 2.3.2. Thư mục Java

Đây chính là nơi chứa các gói (package)của dự án, có thể tạo các gói ở đây và bên trong là các lớp.

### 2.3.3. Thư mục Res

**Drawable**: Đây chính là thư mục chứa các file hình ảnh, config xml... trong dự án android. Ví dụ như muốn ứng dụng sử dụng một hình ảnh nào đó là background thì ảnh đó sẽ bỏ vào thư mục này. Hoặc một muốn điều chỉnh một nút button khi click vào màu xanh còn khi không click vào màu trắng thì sẽ config trong file xml và lưu vào trong này.

**Layout**: Đây chính là thư mục lưu các file xml về giao diện của các màn hình ứng dụng của . Ở trên phần số 1 có các package lưu các class, các class này sẽ kết nối với các file xml trong thư mục layout nào để tạo nên một màn hình có giao diện cho người dùng thao tác.

**Mipmap**: Đây là thư mục mà sẽ chứa ảnh logo ứng dụng chúng ta, lúc này có nói là các file hình ảnh sẽ được chứa trong thư mục drawable nhưng ngoại lệ ảnh logo thì chứa trong thư mục *mipmap* này cho chuẩn.

**Values**: Sẽ có rất nhiều file ở bên trong như sau:

- **color.xml**: đây là file định nghĩa các mã màu trong dự án android của bạn, khi sử dụng màu nào chỉ cần gọi tên mã màu đã định nghĩa trong đây ra là xong.
- **dimens.xml**: đây là file mà sẽ định nghĩa ra các kích thước như cỡ chữ, chiều cao, chiều rộng các view. Tại sao không set cứng mà phải định nghĩa ở đây? Như các biết thì thiết bị Android có cả ngàn chiếc và 1 đồng nhà sản xuất nó khác IOS hoàn toàn thế nên sẽ có rất nhiều kích thước màn hình khác nhau loại 4 inch, 5inch, 5.5 inch... chính vì thế khi set một đoạn văn bản nào đó kích thước là 10sp thì màn hình 4 inch hiển thị rất đẹp chứ qua 5.5 là nó rất nhỏ. Đây là lí do có file dimens.xml nhé.
- **strings.xml**: đây là file định nghĩa các đoạn văn bản trong ứng dụng Android của ví dụ như có 1 đoạn văn bản mà sử dụng đi sử dụng lại trong các màn hình khác nhau, khi set cứng ở nhiều nơi thì khi cần chỉnh sửa thì phải tìm hết tất cả và sửa lại. Bây giờ định nghĩa đoạn văn bản đó trong đây và khi dùng thì gọi ra sử dụng và sau này chỉnh sửa chỉ cần sửa trong đây là xong, nó sẽ apply tất cả mọi nơi.
- **styles.xml**: đây chính là nơi định nghĩa các giao diện của các file layout trong thư mục layout đã nói phía trên. Nếu muốn chỉnh một nút Button chiều cao 10dp, chiều rộng 10dp, màu xanh... và lại sử dụng kiểu thiết kế này ở 5 màn hình khác nhau không thể mỗi màn hình lại định nghĩa lại như thế sẽ làm duplicate code (lặp lại) và sẽ không tối ưu tí nào cả. Thay vào đó chỉ cần định nghĩa một file giao diện như trên và ở mỗi màn hình chỉ cần gọi là xong.

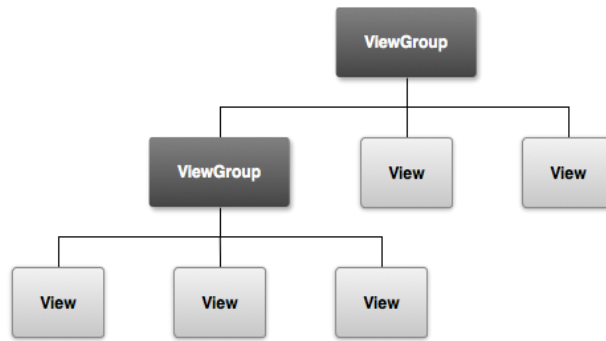
### **Tập Build.grade**

Là file cấu hình cho ứng dụng android thực thi dịch, xây dựng và đóng gói ứng dụng Android . Nó ấn định phiên bản SDK tối thiểu cho ứng dụng.

### **2.4 Thành phần giao diện**

#### *- Nhóm hiển thị*

Một lớp nhóm hiển thị (View Group) là một lớp chứa được lớp hiển thị và nhóm lớp hiển thị khác để tổ chức và sắp xếp các đối tượng View trên một màn hình. Các đối tượng nhóm hiển thị được sử dụng cho việc tạo ra một hệ thống phân cấp của các đối tượng hiển thị (xem bên dưới) do đó có thể tạo các layout phức tạp hơn.



Hình 2.4a. Nhóm hiển thị và các đối tượng

#### 2.4.1 Một số lớp nhóm hiển thị cơ bản

##### ***LinearLayout***

Tồn tại để hiển thị các phần tử theo một thứ tự xếp chồng lên nhau theo chiều ngang hoặc chiều dọc. `LinearLayout` cũng có thể được sử dụng để gán `weight` cho các phần tử `View` con để các phần tử được cách khoảng trên màn hình theo tỉ lệ tương ứng với nhau.

### ***RelativeLayout***

Lớp con này của ViewGroup cho phép hiển thị các phần tử trên màn hình tương đối với nhau, cung cấp nhiều tính linh hoạt hơn và tự do trong cách layout của xuất hiện so với LinearLayout.

### ***FrameLayout***

Được thiết kế để hiển thị một View con tại một thời điểm, FrameLayout vẽ các phần tử trong một ngăn xếp và cung cấp một cách đơn giản để hiển thị một phần tử trên các kích cỡ màn hình khác nhau.

### ***ScrollView***

Một lớp mở rộng của FrameLayout, lớp ScrollView xử lý việc cuộn các đối tượng con của nó trên màn hình.

### ***RecyclerView***

Lớp RecyclerView là một lớp con của ViewGroup, nó liên quan đến các lớp ListView và GridView và nó được cung cấp bởi Google thông qua thư viện hỗ trợ RecyclerView cho các phiên bản Android cũ hơn. Lớp RecyclerView đòi hỏi việc sử dụng các mẫu thiết kế view holder để tái sử dụng phần tử một cách có hiệu quả và nó hỗ trợ việc sử dụng một LayoutManager, một thành phần trang trí, và một phần tử động để làm cho thành phần này vô cùng linh hoạt và đơn giản.

### ***CoordinatorLayout***

Được thêm gần đây vào thư viện hỗ trợ thiết kế, lớp CoordinatorLayout sử dụng một đối tượng Behavior để xác định cách các phần tử View con sẽ được sắp xếp và di chuyển khi người dùng tương tác với ứng dụng của .

#### 2.4.2 View

##### - Khái niệm:

Trong một ứng dụng Android, giao diện người dùng được xây dựng từ các đối tượng View và ViewGroup. Có nhiều kiểu View và ViewGroup. Mỗi một kiểu là một hậu duệ của class View và tất cả các kiểu đó được gọi là các Widget. Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View. Trong Android Platform, các screen luôn được bố trí theo một kiểu cấu

trúc phân. Một screen là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một screen thì trong hàm onCreate của mỗi activity cần phải được gọi một hàm setContentView (R.layout.main); hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode.

- Một số view cơ bản trong android

| <b>Tên View</b>     | <b>Chức năng</b>  |
|---------------------|---|
| <b>TextView</b>     | Cho phép người dùng hiển thị 1 đoạn văn bản lên màn hình mà không cho phép người dùng sửa nó.   |
| <b>EditText</b>     | Cho phép người dùng nhập, xóa, sửa 1 đoạn văn bản vào trong đó. Tất nhiên cũng có thể không cho phép người dùng thao tác với nó.<br>Có nhiều dạng EditText khác nhau như: Plain text, Person Name, Password, Email, Phone, .... |
| <b>Button</b>       | Dùng để bắt các sự kiện khi người dùng thao tác trên giao diện.   |
| <b>ImageButton</b>  | ImageButton: là dạng button nhưng có thể chèn thêm hình ảnh vào để giao diện thêm sinh động, trực quan hơn.   |
| <b>CheckBox</b>     | Một dạng button đặc biệt chỉ có hai trạng thái là check và uncheck.   |
| <b>ToggleButton</b> | Có thể xem nó như một Checkbox và có kèm theo hiệu ứng ánh sáng bật/tắt thể hiện trạng thái Check/Uncheck.  |
| <b>RadioButton</b>  | Chỉ cho phép chọn một mục lựa chọn trong nhóm các lựa chọn.   |
| <b>Spinner</b>      | Là view thể hiện quá trình xử lý diễn ra bên dưới ứng dụng, tạo cảm giác thực cho người dùng. Có 2 dạng progressBar là Large và Horizontal  |
| <b>ListView</b>     | Hiển thị ở dạng danh sách   |
| <b>GridView</b>     | Là view được hiển thị dưới dạng lưới gồm nhiều item con bên trong và ta có thể tùy chỉnh nội dung cũng như các  |

| Tên View                | Chức năng  |
|-------------------------|--|
|                         | đối tượng nằm bên trong item một cách tùy ý.   |
| <b>ViewFlipper</b>      | Cho phép định nghĩa một tập hợp nhiều View nhưng chỉ có 1 View hiển thị tại một thời điểm và hỗ trợ hiệu ứng chuyển đổi giữa các View  |
| <b>QuickContactBage</b> | Hiển thị hình ảnh gắn liền với một đối tượng bao gồm số điện thoại, tên, email đồng thời hỗ trợ việc gọi, nhắn tin sms, email hay tin nhắn tức thời (IM – instant message).  |
| <b>WebView</b>          | Là View được sử dụng hiển thị trang Web, là lớp cơ bản dựa vào nó có thể đưa vào trình duyệt Web, hoặc hiển thị vài nội dung online trong Activity. Với dạng view này ta cần phải cấp quyền sử dụng Internet trong Manifest file |
| <b>ImageView</b>        | Hiển thị hình ảnh  |
| <b>Switch</b>           | Tồn tại hai trạng thái đóng mở   |
| <b>ProgressBar</b>      | Thể hiện tiến trình, mức độ  |
| ...                     | ...  |

- Một số thuộc tính View cơ bản

| Tên thuộc tính       | Công dụng  |
|----------------------|--|
| <b>layout_width</b>  | chiều rộng của view (Tính theo trục Ox từ góc trên bên trái qua phải). Có 2 thuộc tính: <b>wrap_content</b> (Tự dẫn theo nội dung của View), <b>match_parent</b> (Kích thước chiều ngang của View bằng kích thước chiều ngang của đối tượng chứa). |
| <b>layout_heigth</b> | Chiều cao của view (Tính theo trục dp từ góc trên bên trái xuống dưới). Có 3 thuộc tính như <b>layout:width</b>  |
| <b>text</b>          | nội dung của View  |
| <b>texcolor</b>      | Màu chữ  |
| <b>background</b>    | Màu nền  |
| <b>id</b>            | Định danh của View   |

|                   |   |
|-------------------|---|
| <b>gravity</b>    | Căn lề cho nội dung của View  |
| <b>textStyle</b>  | Thiết lập dáng chữ: đậm, nghiêng, gạch chân...                                      |
| <b>textSize</b>   | Thiết lập cỡ chữ.   |
| <b>fontFamily</b> | Thiết lập họ phong chữ  |
| <b>inputType</b>  | Thiết lập kiểu nhập cho dữ liệu đầu vào.  |
| <b>hint</b>       | Đoạn văn bản gợi ý cho người dùng biết về chức năng hay ràng buộc gì đó của View... |
| ...               | ...   |

## 2.5. Vòng đời ứng dụng android

Các Activity trong hệ thống được quản lý như một ngăn xếp activity (activity stack). Khi 1 activity mới bắt đầu nó được đặt lên đầu của ngăn xếp và trở thành Running Activity (activity đang chạy), đồng thời activity trước đó sẽ nằm ngay phía dưới trong ngăn xếp đó, và sẽ không trở nên visible (nhìn thấy) cho đến khi activity ở trên thoát ra khỏi ngăn xếp.

### Một Activity gồm 4 trạng thái chính:

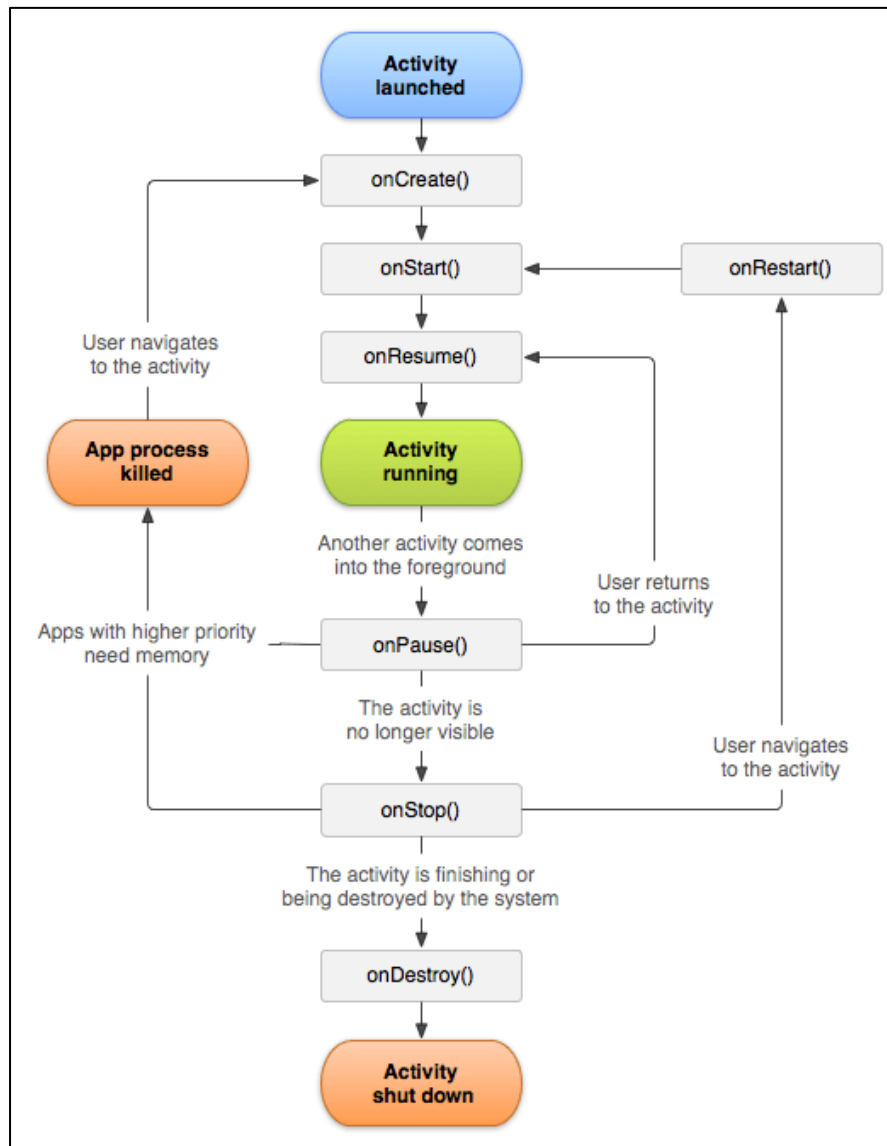
Nếu activity ở phía trên của màn hình (hay ở trên cùng của ngăn xếp), thì nó đang ở trạng thái active (hoạt động) / running (đang chạy). Ví dụ khi ta cần gọi điện thì activity bấm số đó đang ở trạng thái active.

Nếu activity không thể tương tác nhưng vẫn nhìn thấy (khi mà bị che bởi một activity khác nhưng người dùng vẫn có thể nhìn thấy nó ở phía sau) thì activity này đang ở trạng thái paused (tạm dừng). Khi ở trạng thái này activity có thể bị xóa bỏ bởi hệ thống khi thiết bị thiếu bộ nhớ. Ví dụ khi có 1 activity khác dạng dialog hiện lên chỉ che đi 1 phần của activity hiện tại thì activity vào trạng thái paused.

Nếu activity hoàn toàn bị che khuất bởi activity khác thì nó đang ở trạng thái stopped (đã dừng). Activity này vẫn giữ được tất cả trạng thái và thông tin, nhưng không còn hiển thị với người dùng và thường xuyên bị xóa bỏ bởi hệ thống khi thiếu bộ nhớ. Ví dụ khi ta tắt màn hình thì khi đó activity vào trạng thái stopped.

Nếu activity ở trạng thái paused (tạm dừng) hay stopped (đã dừng), hệ thống có thể xóa bỏ activity đó khỏi bộ nhớ bằng cách yêu cầu nó tự kết thúc hoặc xóa bỏ tiến trình

của nó. Khi activity đó hiển thị lại với người dùng thì sẽ được khởi tạo lại và khôi phục lại trạng thái trước đó.



Hình 2.5.1. Vòng đời của một Activity

## 2.6. Thành phần Intents

### a. Khái niệm

Intent là một thành phần quan trọng trong android. Intents cho phép các thành phần ứng dụng có thể yêu cầu các hàm từ các thành phần ứng dụng android khác.

Intent [2] là đối tượng của lớp **android.content.Intent**. Mã của nó có thể gửi Intent vào hệ thống Android với chỉ định thành phần mục tiêu gửi đến.



Một Intent có thể chứa dữ liệu thông qua đối tượng của một lớp Bundle. Dữ liệu này có thể được sử dụng bởi các thành phần tiếp nhận.

### b. Các loại Intent

Android hỗ trợ 2 loại Intent là Intent tường minh (explicit) và Intent không tường minh (implicit).

Một ứng dụng có thể xác định thành phần mục tiêu một cách trực tiếp vào Intent (mục tiêu được yêu cầu là rõ ràng) hoặc yêu cầu hệ thống Android đánh giá các thành phần đã đăng ký trên dữ liệu đích để chọn ra một cái để gửi yêu cầu đến (Intent không tường minh).

**Intent tường minh (Explicit intents):** Là những ý định (intent) chỉ định rõ ràng tên của các thành phần mục tiêu để xử lý; trong đó, trường mục tiêu (tùy chọn) được gán một giá trị cụ thể thông qua các phương thức *setComponent()* hoặc *setClass()*.

```
1 // Sử dụng Bundle cách 1.
2 Intent mIntent = new Intent(this, GreetingActivity.class);
3 Bundle extras = mIntent.getExtras();
4 extras.putString("firstName", "<firstName>");
5 extras.putString("lastName", "<lastName>");
6
7 // Sử dụng Bundle cách 2.
8 Intent mIntent2 = new Intent(this, GreetingActivity.class);
9 Bundle mBundle = new Bundle();
10 mBundle.putString("firstName", "<firstName>");
11 mBundle.putString("lastName", "<lastName>");
12 mIntent2.putExtras(mBundle);
13
14 // Cách 3:
15 // Sử dụng putExtra(), phương thức gọi tắt của Intent.
16 Intent mIntent3 = new Intent(this, GreetingActivity.class);
17 mIntent3.putExtra("firstName", "<firstName>");
18 mIntent3.putExtra("lastName", "<lastName>");
```

Hình 2.6.1. Ví dụ Intents tường minh

Tại **Activity** mục tiêu:

```

1 | Intent intent = this.getIntent();
2 |
3 | String firstName= intent.getStringExtra("firstName");
4 | String lastName = intent.getStringExtra("lastName");
5 |
6 | // Hoặc
7 | Bundle extras = this.getIntent().getExtras();
8 |
9 | String firstName1 = extras.getString("firstName");
10 | String lastName2 = extras.getString("lastName");

```

Hình 2.6.2. Ví dụ Intents tường minh

**Intent không tường minh (Implicit Intents):** Là những ý định (intent) không chỉ định rõ một mục tiêu thành phần, nhưng bao gồm đầy đủ thông tin cho hệ thống để xác định các thành phần có sẵn là tốt nhất để chạy cho mục đích đó. Hãy xem xét một ứng dụng liệt kê các nhà hàng có sẵn ở gần . Khi bấm vào một tùy chọn nhà hàng cụ thể, ứng dụng sẽ hỏi một ứng dụng khác để hiển thị các tuyến đường đến nhà hàng đó. Để đạt được điều này, nó có thể gửi một ý định rõ ràng trực tiếp đến các ứng dụng **Google Maps**, hoặc gửi ý định ngầm, ý định sẽ được chuyển giao cho bất kỳ ứng dụng nào cung cấp các tính năng bản đồ (map) (chẳng hạn, **Yahoo Maps**).

```

1 | // Một Intent không tường minh, yêu cầu xem một URL
2 | // Một đường dẫn URL vốn là thứ thường được xem trên trình duyệt.
3 | // Intent này phù hợp với HelloWorld activity ở trên.
4 |
5 | Intent intent = new Intent (Intent.ACTION_VIEW, Uri.parse("//o7planning.org"));
6 | startActivity(intent);

```

Hình 2.6.3. Ví dụ Intents không tường minh

## **CHƯƠNG 3 : KỸ THUẬT LẤY DỮ LIỆU TỰ ĐỘNG TỪ TRANG WEB**

Như các bạn đã biết, khi bạn xây dựng và phát triển một ứng dụng phân tán với số lượng người dùng lên đến hàng trăm, hàng nghìn người ở nhiều địa điểm khác nhau, khó khăn đầu tiên mà bạn gặp phải là sự giao tiếp giữa Client và Server bị tường lửa (firewalls) và Proxy Server ngăn chặn lại. Hơn thế nữa, hiện nay rất nhiều website không cho phép chúng ta lấy dữ liệu từ trang của họ. Như vậy, chúng ta cần phát triển một thứ có thể giúp ta giải quyết vấn đề này và Web Services là một phương pháp giải quyết vấn đề trên.

### **3.1. Mạng Internet**

Là một mạng máy tính toàn cầu [3] có thể được truy nhập công cộng gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (Packet Switching) dựa trên một giao thức liên mạng được chuẩn hóa (giao thức IP). Hệ thống này bao gồm hàng ngàn máy tính nhỏ hơn các doanh nghiệp, của các viện nghiên cứu và các trường đại học, của người dùng cá nhân và các chính phủ trên toàn cầu.

### **3.2. Dịch vụ lưu trữ trực tuyến**

#### **3.2.1. Giới thiệu**

Hosting là không gian trên máy chủ có cài đặt các dịch vụ Internet như world wide web (www), truyền file (FTP), Mail... bạn có thể chứa nội dung trang web hay dữ liệu trên không gian đó. Hosting đồng thời cũng là nơi diễn ra tất cả các hoạt động giao dịch, trao đổi thông tin giữa website với người sử dụng Internet và hỗ trợ các phần mềm Internet hoạt động. DN có thể chọn thuê web hosting của nhà cung cấp dịch vụ (ISP) có dung lượng phù hợp với dung lượng website. Với bất kỳ hình thức nào (tự trang bị máy chủ hay thuê máy chủ) thì DN cũng nên có các hiểu biết cần thiết về Web Hosting và máy chủ Web.

Lý do bạn phải thuê Hosting để chứa nội dung trang web, dịch vụ mail, ftp, vì những máy tính đó luôn có một địa chỉ cố định khi kết nối vào Internet (đó là địa chỉ IP), còn như nếu bạn truy cập vào internet như thông thường hiện nay thông qua các IPS (Internet Service Provider - Nhà cung cấp dịch vụ Internet) thì địa chỉ IP trên máy

bạn luôn bị thay đổi, do đó dữ liệu trên máy của bạn không thể truy cập được từ những máy khác trên Internet.

### 3.2.2. Các hoạt Hosting

- Shared hosting: Chia sẻ host
- Collocated hosting: Thuê chỗ đặt máy chủ
- Dedicated Server: Máy chủ dùng riêng
- Virtual Private Server: VPS là máy chủ riêng ảo

### 3.2.3. Các yêu cầu và tính năng cần thiết của hosting

- Đầu tiên phải nói đến về vấn đề tốc độ. Máy chủ chạy dịch vụ Web phải có cấu hình đủ lớn để đảm bảo xử lý thông suốt, phục vụ cho số lượng lớn người truy cập. Phải có đường truyền kết nối tốc độ cao để đảm bảo không bị nghẽn mạch dữ liệu.
- Máy chủ phải được người quản trị hệ thống chăm sóc, cập nhật, bảo dưỡng thường xuyên nhằm tránh các rủi ro về mặt kỹ thuật cũng như bảo mật.
- Web Hosting phải có một dung lượng đủ lớn (tính theo MBytes) để lưu giữ được đầy đủ các thông tin, dữ liệu, hình ảnh,... của Website
- Phải có bandwidth (băng thông) đủ lớn để phục vụ các hoạt động giao dịch, trao đổi thông tin của Website
- Phải hỗ trợ truy xuất máy chủ bằng giao thức FTP để cập nhật thông tin.
- Hỗ trợ các ngôn ngữ lập trình cũng như cơ sở dữ liệu để thực thi các phần mềm trên Internet hoặc các công cụ viết sẵn để phục vụ các hoạt động giao dịch trên Website như gửi mail, upload qua trang Web, quản lý sản phẩm, tin tức...
- Hỗ trợ đầy đủ các dịch vụ E-mail như POP3 E-mail, E-mail Forwarding, DNS...
- Có giao diện quản lý Web Hosting để dễ dàng quản lý website, các tài khoản FTP, Email...
- Không bị chèn các banner quảng cáo của nhà cung cấp.

### 3.2.4. Dung lượng hosting

Dung lượng của web hosting là khoảng không gian bạn được phép lưu trữ dữ liệu của mình trên ổ cứng của máy chủ. Như đã nói ở trên, bạn thuê một web hosting cũng giống như bạn thuê văn phòng trong một nhà cao ốc. Vậy ở đây, dung lượng của web hosting cũng giống như diện tích văn phòng của bạn.

### 3.2.5. Tại sao cần phải mua hosting

Nếu không có hosting thì website sẽ chỉ hoạt động được trên máy tính cá nhân và chạy trên Localhost, dữ liệu sẽ không được chia sẻ trên mạng. Vậy nên rất cần thiết để có 1 hosting.

## 3.3. Dịch vụ web

### 3.3.1. Giới thiệu

*Định nghĩa:*

Là sự kết hợp các ứng dụng trên máy tính cá nhân, thiết bị di động với ứng dụng trên các thiết bị khác, các cơ sở dữ liệu và các mạng máy tính để tạo thành một cơ cấu tính toán hoàn hảo mà người sử dụng có thể làm việc, yêu cầu, phân tích, khai thác, cập nhật... thông tin với nó thông qua các mạng Internet hoặc wifi.

### 3.3.2 Các công nghệ xây dựng dịch vụ Web

Phân loại theo công nghệ sử dụng để xây dựng dịch vụ Web cho ứng dụng di động dùng .NET Framework, hiện nay có các công nghệ chính sau:

- Web Service
- WCF Service
- WCF REST Service
- Web API Service

### 3.3.3 Đặc điểm của dịch vụ Web

#### a. Ưu điểm

- Cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.
- Sử dụng các giao thức và chuẩn mở.
- Nâng cao khả năng tái sử dụng.
- Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại.
- Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng trong việc phát triển các ứng dụng phân tán.
- Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống qua các doanh nghiệp khác.

*b. Nhược điểm*

- Vào những khoảng thời gian chết của Web Service sẽ dẫn đến những thiệt hại lớn:
  - + Giao diện không thay đổi
  - + Có thể lỗi nếu một máy khách không được nâng cấp
  - + Thiếu các giao thức cho phân vận hành.
- Có quá nhiều chuẩn cho Web Service khiến người dùng khó nắm bắt
- Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

### 3.4. Thư viện Simple HTML Dom

#### 3.4.1. Giới thiệu

Simple Html Dom [4] là thư viện hỗ trợ lấy tin từ trang web PHP giúp cho việc chúng ta có thể lấy các thuộc tính của các thẻ HTML trong một website. Chúng ta có thể lấy trực tiếp thông tin từ link web hoặc là từ một chuỗi, đồng thời ta có thể lấy về, và sửa đổi nội dung.

a. Cách tải về thư viện

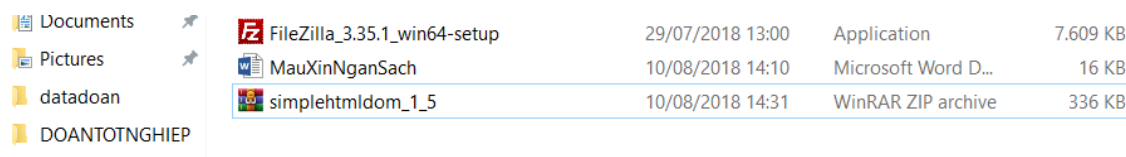
Đầu tiên, các bạn vào đường dẫn này:

<https://sourceforge.net/projects/simplehtmldom/files/latest/download?source=files>

b. Cách sử dụng thư viện

Đường dẫn sẽ tự động tải xuống 1 file xuống với đuôi tệp tin là

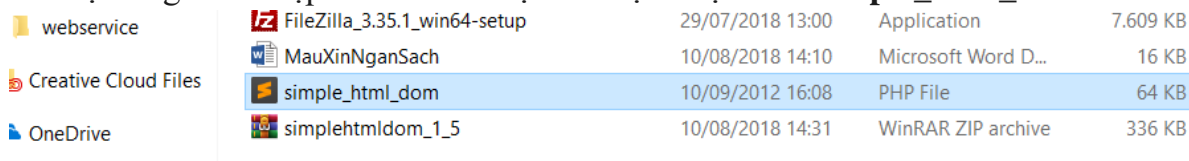
“simplehtmldom\_1\_5.zip” cho bạn



|               |                              |                  |                     |          |
|---------------|------------------------------|------------------|---------------------|----------|
| Documents     | FileZilla_3.35.1_win64-setup | 29/07/2018 13:00 | Application         | 7.609 KB |
| Pictures      | MauXinNganSach               | 10/08/2018 14:10 | Microsoft Word D... | 16 KB    |
| datadoan      | simplehtmldom_1_5            | 10/08/2018 14:31 | WinRAR ZIP archive  | 336 KB   |
| DOANTOTNGHIEP |                              |                  |                     |          |

Hình 3.4.1.1 Tệp tin thư viện Simple HTML Dom

Bạn sẽ giải nén tệp tin ở trên ra bạn sẽ nhận được file “simple\_html\_dom”



|                      |                              |                  |                     |          |
|----------------------|------------------------------|------------------|---------------------|----------|
| webservice           | FileZilla_3.35.1_win64-setup | 29/07/2018 13:00 | Application         | 7.609 KB |
| Creative Cloud Files | MauXinNganSach               | 10/08/2018 14:10 | Microsoft Word D... | 16 KB    |
| OneDrive             | simple_html_dom              | 10/09/2012 16:08 | PHP File            | 64 KB    |
|                      | simplehtmldom_1_5            | 10/08/2018 14:31 | WinRAR ZIP archive  | 336 KB   |

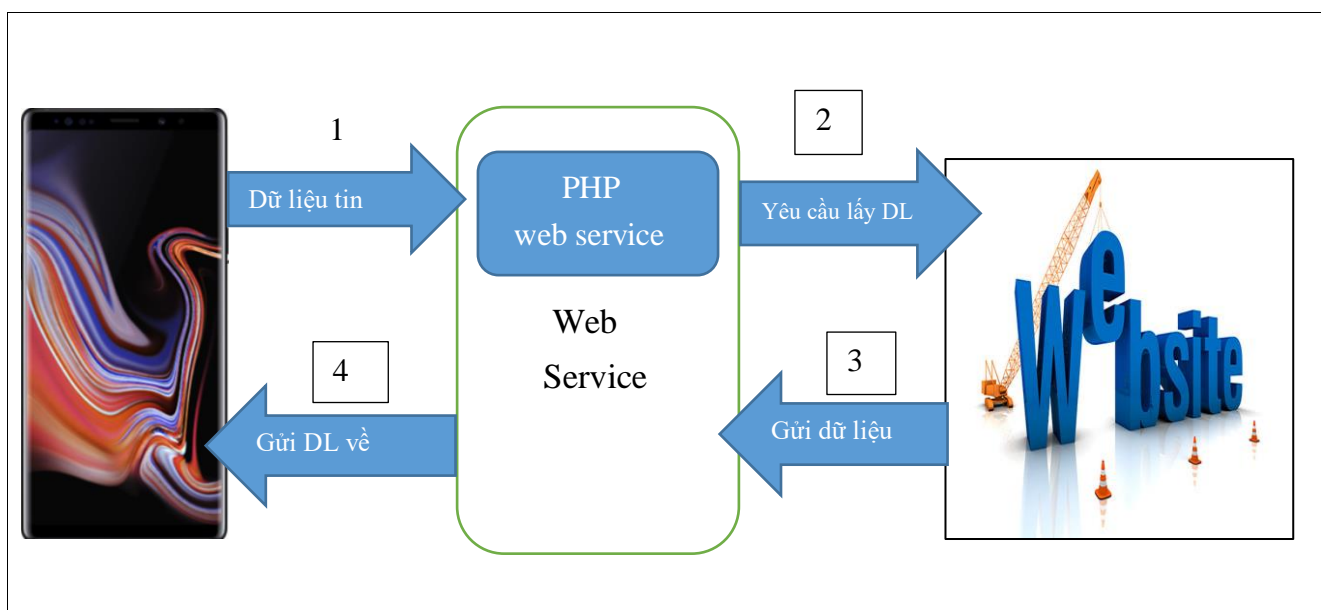
Hình 3.4.1.2 Thư viện khi được dùng trên phần mềm Sublime Text 3

Sau đó bạn kích chuột 2 lần để mở file này lên, bạn sẽ được như hình dưới đây

```
51 define('HDOM_TYPE_ROOT', 5);
52 define('HDOM_TYPE_UNKNOWN', 6);
53 define('HDOM_QUOTE_DOUBLE', 0);
54 define('HDOM_QUOTE_SINGLE', 1);
55 define('HDOM_QUOTE_NO', 3);
56 define('HDOM_INFO_BEGIN', 0);
57 define('HDOM_INFO_END', 1);
58 define('HDOM_INFO_QUOTE', 2);
59 define('HDOM_INFO_SPACE', 3);
60 define('HDOM_INFO_TEXT', 4);
61 define('HDOM_INFO_INNER', 5);
62 define('HDOM_INFO_OUTER', 6);
63 define('HDOM_INFO_ENDSPACE', 7);
64 define('DEFAULT_TARGET_CHARSET', 'UTF-8');
65 define('DEFAULT_BR_TEXT', "\r\n");
66 define('DEFAULT_SPAN_TEXT', " ");
67 if (!defined('MAX_FILE_SIZE'))
68 {
69     define('MAX_FILE_SIZE', 600000);
70 }
71 // helper functions
72 // -----
73 // get html dom from file
74 // $maxLen is defined in the code as PHP_STREAM_COPY_ALL which is defined as -1.
75 function file_get_html($url, $use_include_path = false, $context = null, $offset = 0, $maxLen = -1, $lowercase = true, $forceTagsClosed = true, $target_charset =
    DEFAULT_TARGET_CHARSET, $stripRN = true, $defaultBRText = DEFAULT_BR_TEXT, $defaultSpanText = DEFAULT_SPAN_TEXT)
76 {
77     // We DO force the tags to be terminated.
78     $dom = new simple_html_dom(null, $lowercase, $forceTagsClosed, $target_charset, $stripRN, $defaultBRText, $defaultSpanText);
79     // For sourceforge users: uncomment the next line and comment the retrieve_url_contents line 2 lines down if it is not already done.
80     $contents = file_get_contents($url, $use_include_path, $context, $offset);
81     // Paperg - use our own mechanism for getting the contents as we want to control the timeout.
82     $contents = retrieve_url_contents($url);
83     if (empty($contents) || strlen($contents) > MAX_FILE_SIZE)
84     {
85         return false;
86     }
87 }
```

Hình 3.4.1.3 Một phần code của thư viện SIMPLE HTML DOM

### 3.5. Mô hình lấy dữ liệu từ trang web về thiết bị di động



Hình 3.5.1 Mô hình chuyển hóa dữ liệu từ trang web về thiết bị di động

*Giải pháp:*

- 1) Khi khách hàng thực hiện thao tác chạm vào màn hình mục tin thì ứng dụng sẽ lấy dữ liệu tự động từ web service.
- 2) Web Service được yêu cầu lấy dữ liệu từ website.
- 3) Website sẽ đẩy dữ liệu về Web Service.

4) Web Service sẽ gửi dữ liệu về điện thoại.

## **CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM**

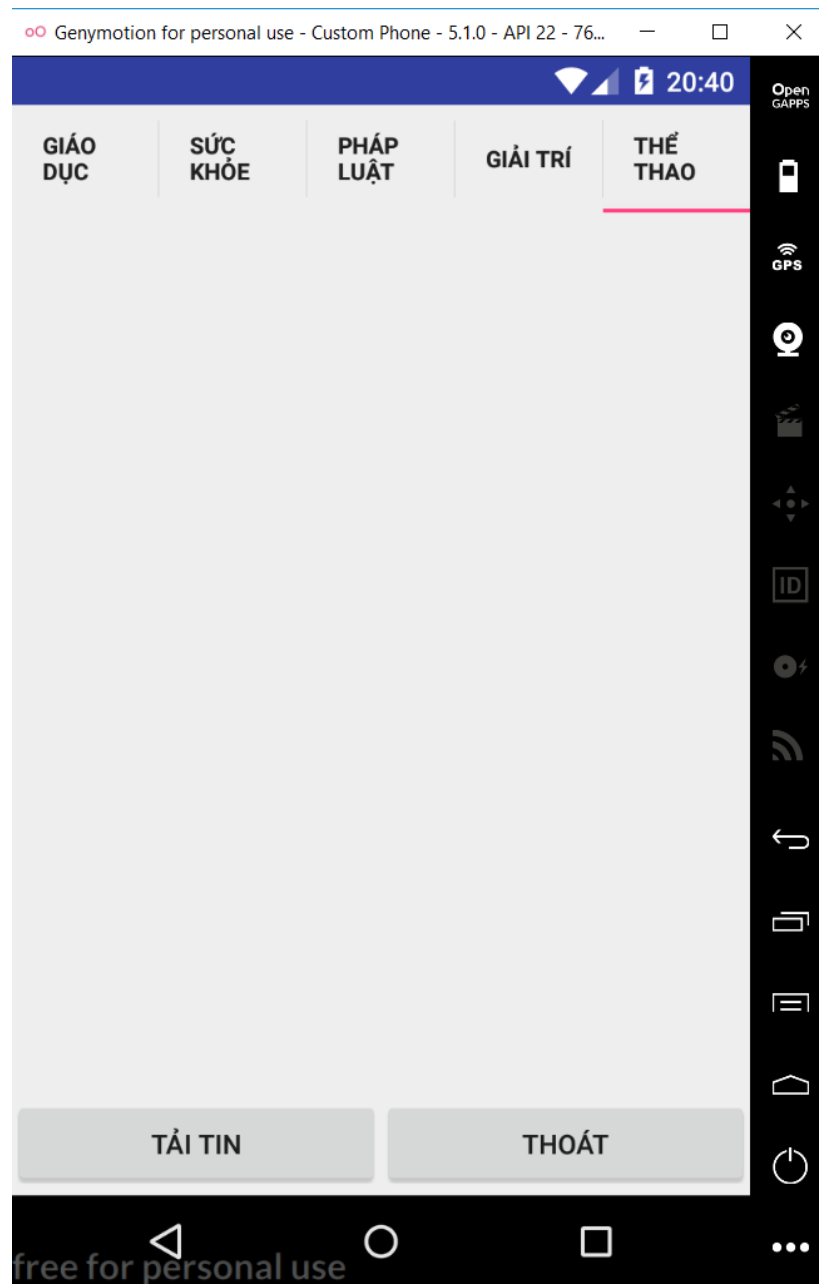
### **4.1. Phát biểu bài toán**

Với nhu cầu đọc tin tức trên các thiết bị thông minh phát triển như hiện nay, việc lấy tin tức cần thiết về các thiết bị di động một cách nhanh nhất và chính xác nhất là một vấn đề được nhiều người quan tâm. Với ứng dụng Android này, người dùng chỉ cần tải ứng dụng về máy sử dụng, việc này sẽ giúp khách hàng được đọc tin một cách dễ dàng.



## 4.2. Một số giao diện chương trình

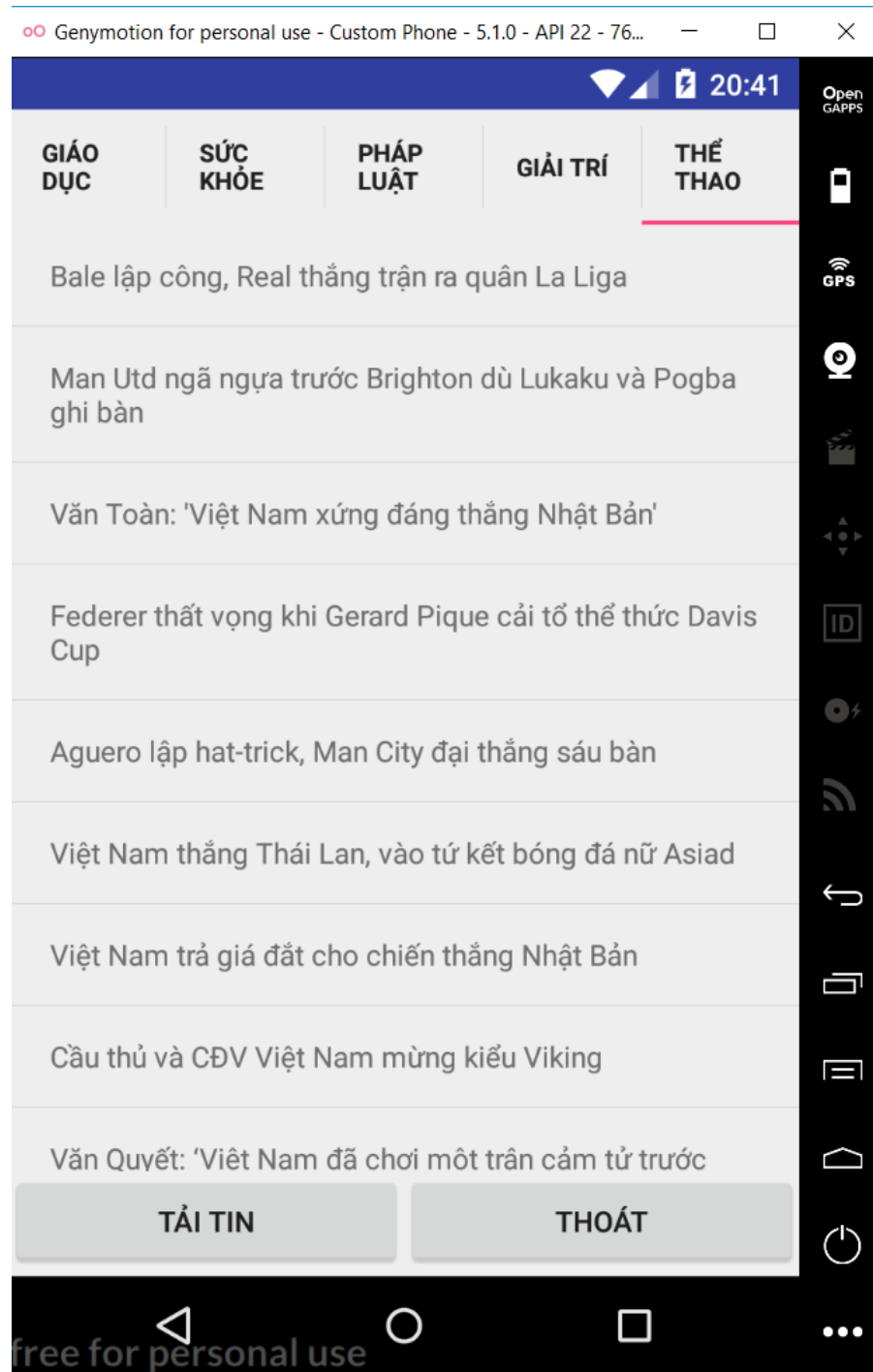
### 4.2.1 Giao diện màn hình chính



Hình 4.2.1.1 Màn hình chính khi mở ứng dụng

- Mô tả : Màn hình này hiển thị giao diện chính khi bắt đầu mở ứng dụng lên gồm có các thông tin sau :
  - + Danh sách các mục cần đọc tin
  - + Nút Tải Tin
  - + Nút Thoát

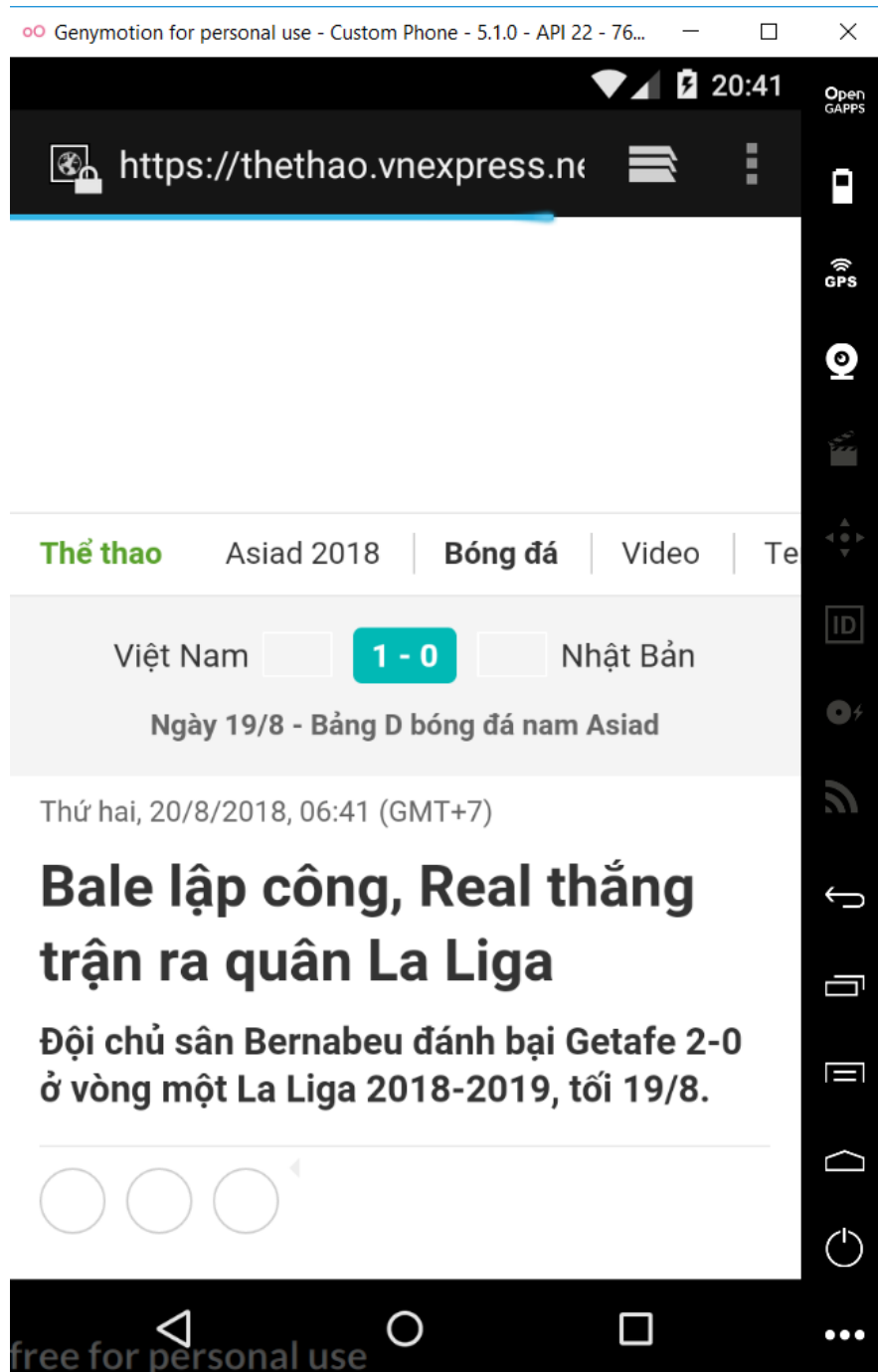
#### 4.2.2 Giao diện các mục tin



Hình 4.2.2.1 Màn hình khi được nhấn vào mục tin thể thao

- Mô tả : Màn hình này hiển thị giao diện sau khi đã tải tin gồm có các thông tin sau :
- + Danh sách các tin cần đọc
- + Nút Tải Tin
- + Nút Thoát

#### 4.2.3 Giao diện khi nhấn vào mục tin



Hình 4.2.3.1 Màn hình khi được nhấn vào mục tin

- Mô tả : Màn hình này hiển thị giao diện sau khi đã nhấn chạm vào các tin cần đọc gồm có các thông tin sau :
  - + Nội dung của toàn tin vừa chạm
  - + Địa chỉ đường dẫn của tin

### **KẾT LUẬN**

Trong đề án này em đã nghiên cứu, tìm hiểu dịch vụ web và cách phát triển ứng dụng trên hệ điều hành Android.

Đề án đã thực hiện được các nhiệm vụ sau:

- Tìm hiểu lịch sử và kiến trúc của hệ điều hành Android.
- Nắm bắt được quy trình làm ra một ứng dụng Android.
- Tìm hiểu về web, web service cho thiết bị di động.
- Tìm hiểu kỹ thuật lấy dữ liệu từ Web Service cho thiết bị di động.
- Các bước phát triển một ứng dụng trên hệ điều hành Android.

Do thời gian còn có hạn, trình độ và kinh nghiệm cũng như kiến thức còn hạn chế nên chương trình vẫn còn nhiều thiếu sót. Em rất mong sẽ nhận được những sự góp ý của thầy cô, bạn bè để chương trình ngày ngày hoàn thiện và có tính ứng dụng nhiều hơn nữa.

Trong tương lai, em sẽ bổ sung và hoàn thiện thêm những tính năng khác để ứng dụng sẽ giúp người dùng được sử dụng một cách dễ dàng hơn nữa.

## TÀI LIỆU THAM KHẢO

- [1] <https://vi.wikipedia.org/wiki>
- [2] <https://o7planning.org/vi/10425/huong-dan-su-dung-intent-trong-android>
- [3] <https://vicogroup.vn/internet-marketing/internet-la-gi-loi-ich-cua-internet-16560.html>
- [4] <https://viblo.asia/p/tim-hieu-ve-simple-html-dom-eXaRlwXdGmx>