

**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 : Trần Thanh Bình**

**Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2019**

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

-----

**XÂY DỰNG ỨNG DỤNG ANDROID LẤY THÔNG TIN  
DỰ BÁO THỜI TIẾT**

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

**Sinh viên : Trần Thanh Bình  
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2019**

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

-----

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Trần Thanh Bình

Mã SV: 1412101018

Lớp: CT1801

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng ứng dụng Android lấy thông tin dự báo thời tiết

## LỜI CẢM ƠN

Qua thời gian học tập và nghiên cứu tại trường Đại học Dân lập Hải Phòng, đầu tiên em xin chân thành cảm ơn sâu sắc tới thầy giáo GS.TS NSUT Trần Hữu Nghị hiệu trưởng trường Đại học Dân Lập Hải Phòng, Ban giám hiệu nhà trường, Bộ môn tin học, các Phòng ban nhà trường đã tạo điều kiện tốt nhất cho chúng em học tập và nghiên cứu trong thời gian qua.

Em xin chân thành cảm ơn tới tất cả thầy cô giáo trong Khoa Công nghệ thông tin - trường Đại Học Dân Lập Hải Phòng, chân thành cảm ơn các thầy giáo, cô giáo đã giảng dạy và truyền đạt những kiến thức bổ ích cho chúng em suốt thời gian học tập tại trường, giúp em có thêm nhiều kiến thức và hiểu rõ hơn các vấn đề mình nghiên cứu, để em có thể hoàn thành đồ án này.

Đặc biệt em xin chân thành cảm ơn thầy Ths.Phùng Anh Tuấn– Khoa Công nghệ thông tin - Trường Đại Học Dân Lập Hải Phòng. Trong thời gian làm tốt nghiệp vừa qua, thầy đã giành nhiều thời gian quý báu và tâm huyết để hướng dẫn em hoàn thành đề tài này.

Dưới đây là kết quả của quá trình tìm hiểu và nghiên cứu mà em đã đạt được trong thời gian vừa qua. Tuy có nhiều cố gắng học hỏi trau dồi để nâng cao kiến thức nhưng không thể tránh khỏi những sai sót. Em rất mong được những sự góp ý quý báu của các thầy giáo, cô giáo cũng như tất cả các bạn để bài luận văn của em được hoàn thiện hơn. Em xin chân thành cảm ơn!

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

Sinh viên

Trần Thanh Bình

## LỜI MỞ ĐẦU

Hiện nay Công nghệ thông tin vô cùng phát triển thì mọi người đều sử dụng máy vi tính hoặc điện thoại di động để làm việc và việc cập nhật thông tin. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web bases Mobile Application đã rất phát triển trên thị trường truyền thông di động.

Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất.

Ngày nay với sự phát triển nhanh chóng của xã hội, nhu cầu đi du lịch ngày càng nhiều, với mục đích nghiên cứu, tìm hiểu về ứng dụng trên android để cập thời tiết rất dễ sử dụng giúp bạn luôn cập nhật thông tin thời tiết.

## MỤC LỤC

|   |           |
|---|-----------|
| <b>LỜI CẢM ƠN</b> .....   | <b>1</b>  |
| <b>LỜI MỞ ĐẦU</b> .....   | <b>5</b>  |
| <b>MỤC LỤC</b> .....  | <b>6</b>  |
| <b>CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID</b> .....                | <b>7</b>  |
| 1.1. Giới Thiệu Hệ Điều Hành Android .....                            | 7         |
| 1.2. Sự Phát Triển Của Hệ Điều Hành Android .....                     | 8         |
| 1.3. Kiến Trúc Của Hệ Điều Hành Android .....                         | 10        |
| <b>CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO</b> .....            | <b>16</b> |
| 2.2. Cài Đặt Android Studio .....                                     | 16        |
| 2.2.1. Yêu Cầu Phần Cứng Máy Tính .....                               | 16        |
| 2.2.3. Thiết Bị Ảo Trong Android Studio .....                         | 18        |
| 2.3. Cấu Trúc Dự Án Android Studio .....                              | 22        |
| 2.3.1. Tạo Mới Một Project .....                                      | 22        |
| 2.3.2. Cài Đặt Một Project .....                                      | 23        |
| 2.3.3. Màn hình làm việc của dự án Android Studio .....               | 25        |
| 2.4. Các Thành Phần Trong Một Ứng Dụng Android .....                  | 28        |
| 2.5. Tạo Giao Diện Chương Trình Trong Android Studio .....            | 31        |
| 2.5.1. Giới Thiệu Android Layout .....                                | 31        |
| 2.5.2. Giới Thiệu Một Số Android View Cơ Bản .....                    | 34        |
| 2.5.3. Bắt Và Xử Lý Sự Kiện Trên Giao Diện .....                      | 35        |
| <b>CHƯƠNG 3: KỸ THUẬT XÂY DỰNG ỨNG DỤNG THỜI TIẾT</b> .....           | <b>37</b> |
| 3.1. Tìm Hiểu Về Dịch Vụ OpenWeatherMap .....                         | 37        |
| 3.1.1. Khái Niệm .....  | 37        |
| 3.1.2. Các Tính Năng Nổi Bật Của API Thời Tiết Open Weather Map ..... | 37        |
| 3.2. Đăng Ký Và Kích Hoạt Sử Dụng Key API Openweathermap .....        | 38        |
| 3.3. Thiết Kế Giao Diện Ứng Dụng Bằng Ngôn Ngữ XML .....              | 39        |
| 3.4. Kỹ Thuật Lấy Nội Dung Trả Về .....                               | 47        |
| <b>CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM</b> .....                       | <b>58</b> |
| 4.1. Phát Biểu Bài Toán .....   | 58        |
| 4.2. Mô Hình Chức Năng .....  | 58        |
| 4.3. Mô Hình Hoạt Động .....  | 59        |
| 4.4. Thiết Kế Giao Diện Giao Chương Trình .....                       | 59        |
| 4.5. Giao Diện Ứng Dụng .....   | 62        |
| <b>KẾT LUẬN</b> .....   | <b>65</b> |
| <b>TÀI LIỆU THAM KHẢO</b> .....                                       | <b>67</b> |

# CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID

## 1.1. Giới Thiệu Hệ Điều Hành Android

Android là một hệ điều hành dựa trên nền tảng Linux, được thiết kế dành cho các thiết bị di động có màn hình cảm ứng 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[1].

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.

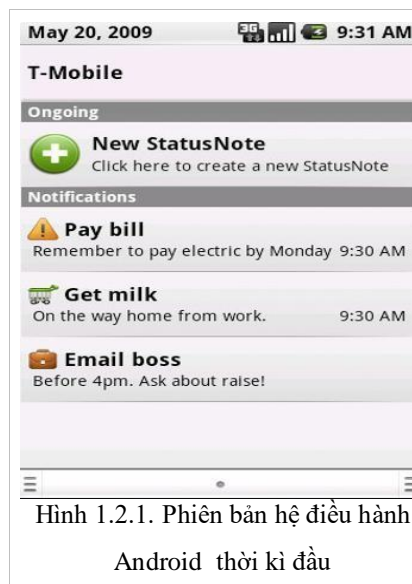
Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên, Android 1.0, được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, Lollipop, Marshmallow, Nougat và bây giờ là Oreo[1].

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

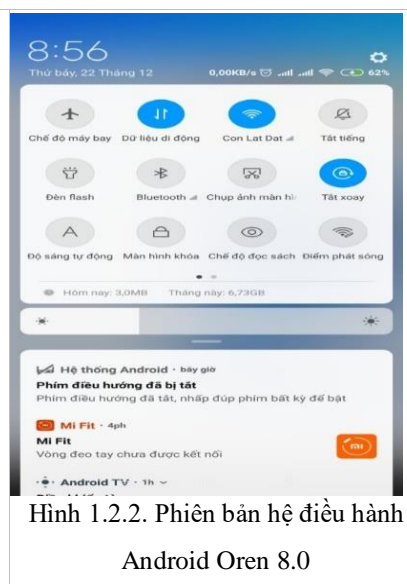
tính năng cũng như giao diện đặc sản của hệ điều hành này đã khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này[1].

## 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 như hình 1.2.1 và hình 1.2.2



Hình 1.2.1. Phiên bản hệ điều hành Android thời kì đầu



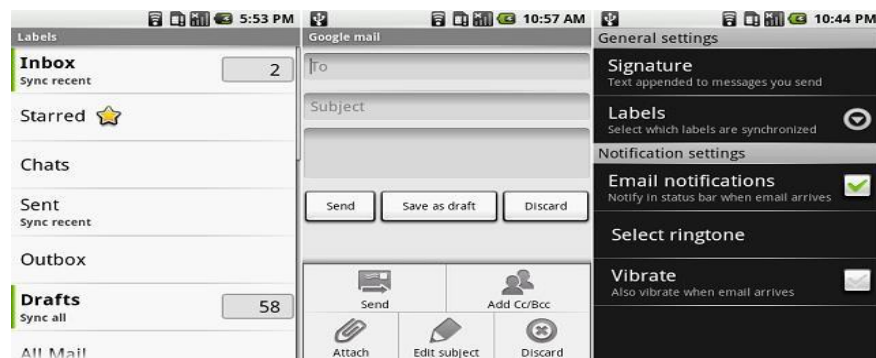
Hình 1.2.2. Phiên bản hệ điều hành Android Oren 8.0

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 như hình 1.2.3 và hình 1.2.4.





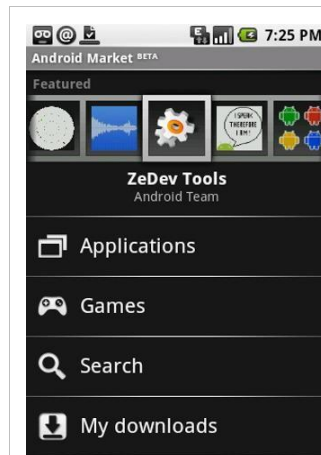
Đồng bộ và tích hợp chặt chẽ với Gmail: Vào thời điểm điện thoại G1 được bán ra, Gmail đã hỗ trợ giao thức POP và IMAP để tích hợp với các trình email trên di động. Tuy nhiên, lúc bấy giờ không có bất kì sản phẩm nào có thể hỗ trợ được hoàn toàn những tính năng ưu việt này của Gmail. Mãi cho đến khi 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 [2]. Giao diện gmail của phiên bản Android thời kỳ đầu như hình 1.2.5.



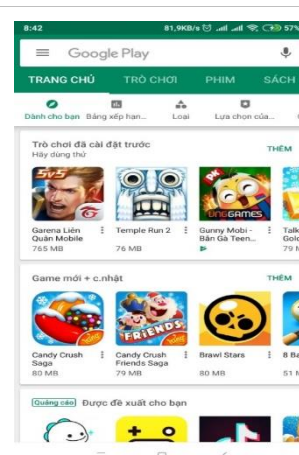
Hình 1.2.5. Gmail trên Android thời kỳ đầu

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 như hình 1.2.6.



Hình 1.2.6. Giao diện kho ứng dụng thời kì đầu

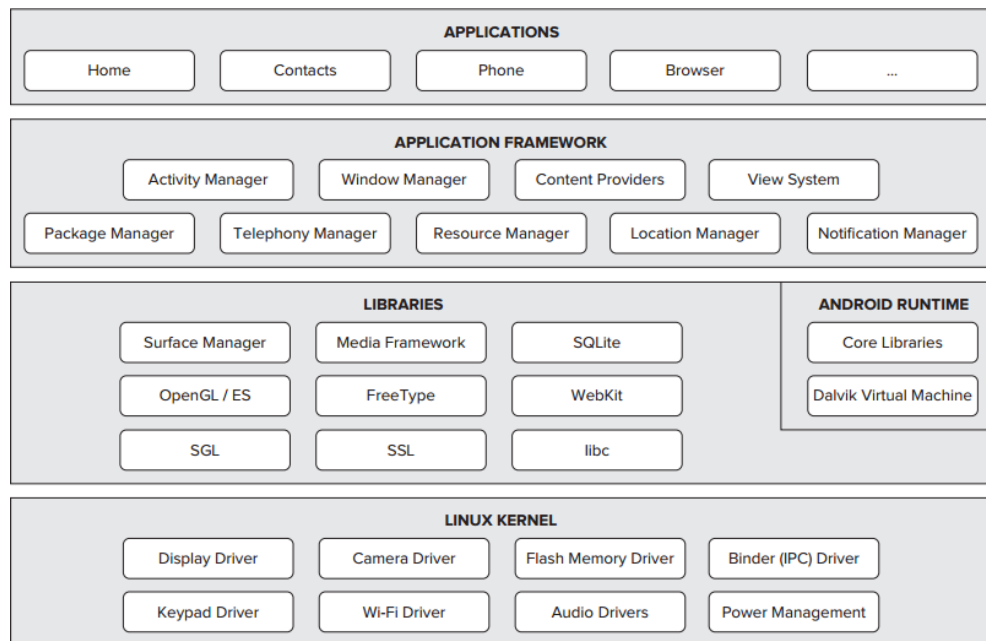


Hình 1.2.7 Giao diện kho ứng dụng Android 8.0

**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 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[2].

### 1.3. Kiến Trúc Của Hệ Điều Hành Android

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



Hình 1.3.1. 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[3].

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, PNG[3].

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.

Thực thi 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.

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[4].

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

Tập hợp các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như: gridview, tableview, linearlayout

“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.

“Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.

“Notifcation Manager” cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar. Activity Maanager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

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.

#### 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 [4].

### 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 1.3.2. Lịch sử phát triển các phiên bản của Android

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 LẬP TRÌNH ANDROID STUDIO

### 2.1. Sơ lược về Android Studio

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android studio dựa vào IntelliJ IDEA, là một IDE tốt cho nhất hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho Android [5].

### 2.2. Cài Đặt Android Studio

#### 2.2.1. Yêu Cầu Phần Cứng Máy Tính

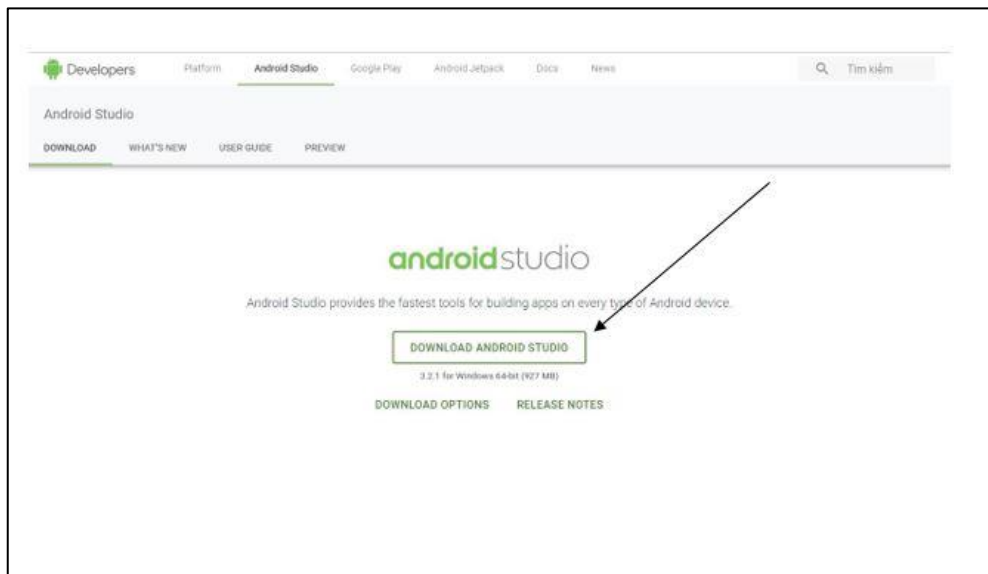
##### Hệ điều hành Windows

- Microsoft® Windows® 10/8/7 (32 or 64-bit)
- GB RAM. (Khuyến cáo là 8GB)
- Chip core I3 trở lên
- 400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator
- Độ phân giải tối thiểu 1366 x 768[6]

#### 2.2.2. Phần mềm Android Studio

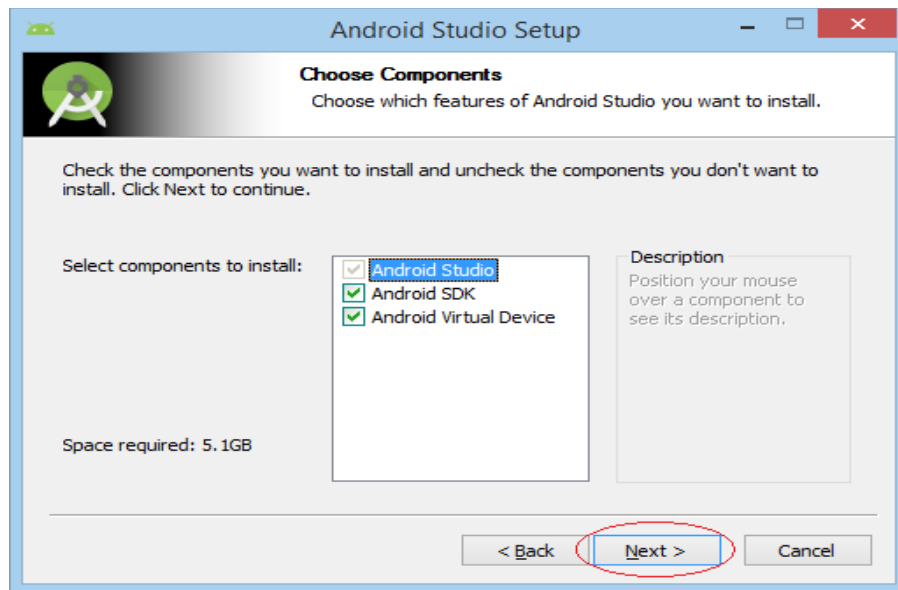
- Vào đường dẫn: “<https://developer.android.com/studio/>” để download bản mới nhất và tiến hành cài đặt như hình 2.2.1.1:





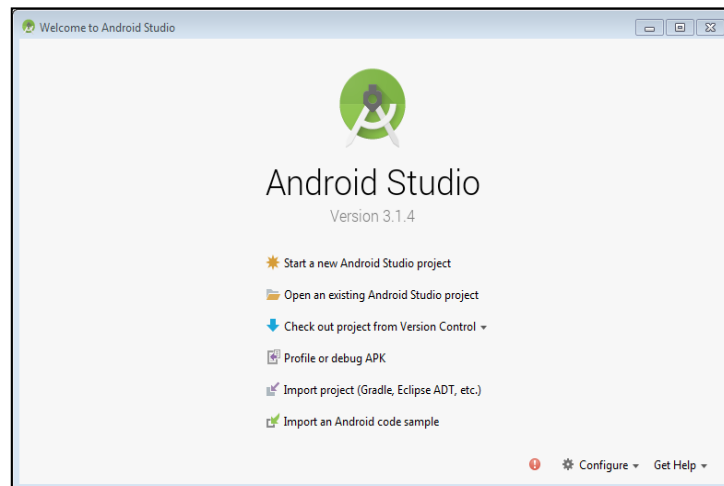
Hình 2.2.1.1. Trang download Android Studio

- Khi cài đặt chú ý chọn cả SDK và trình giả lập thiết bị Android Studio Device như hình 2.2.1.2.



Hình 2.2.1.2. Giao diện cài đặt SDK và AVD

- Tiếp tục chọn next và agree cho đến khi hoàn tất.
- Khi việc cài đặt hoàn tất, chúng ta sẽ được như hình 2.2.1.3:

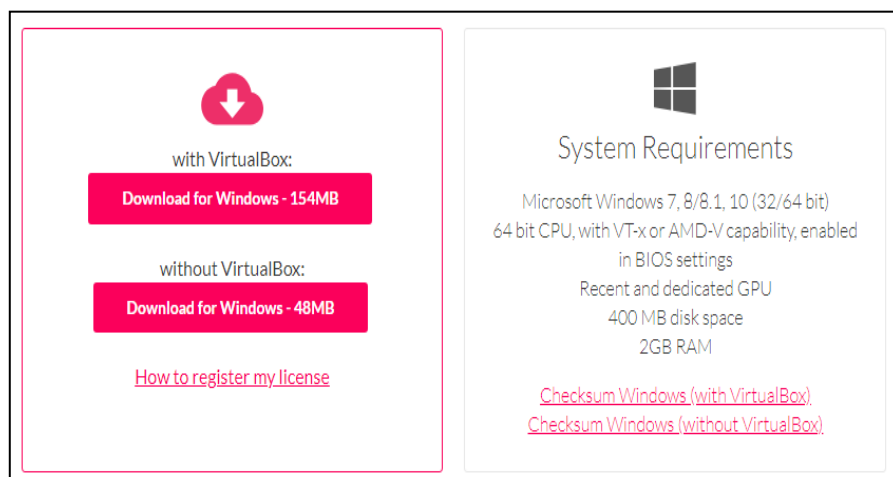


Hình 2.2.1.3. Giao diện cửa sổ Android Studio khi cài đặt hoàn tất

### 2.2.3. Thiết Bị Ảo Trong Android Studio

Máy ảo Android là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Và công cụ máy ảo tiện dụng hiện giờ là Genymotion[6].

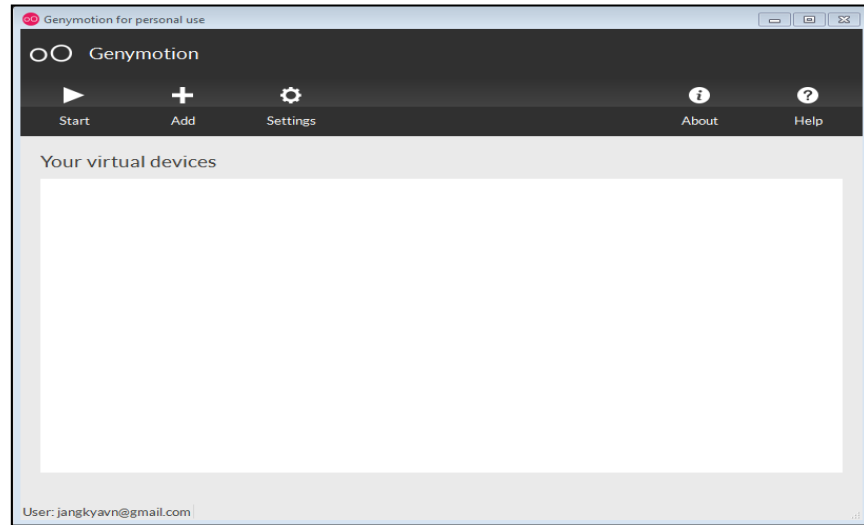
- Để cài đặt máy ảo Genymotion ta truy cập vào đường dẫn:  
<https://www.genymotion.com/download/>
- Ta phải tạo một tài khoản rồi đăng nhập vào mới thấy được mục này:



Hình 2.2.3.1. Giao diện download máy ảo Genymotion

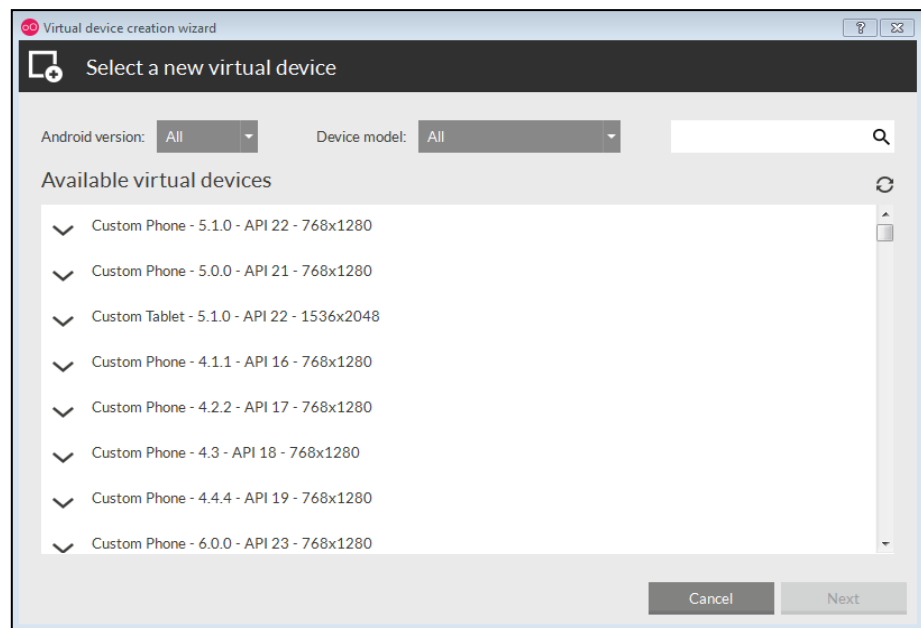
- Ở đây ta nên tải phiên bản “with VirtuaBox” , vì chương trình tích hợp sẵn VirtuaBox cho chúng ta vì máy ảo này phải có VirtuaBox mới chạy được.

- Sau khi cài xong sẽ có giao diện như hình 2.2.3.2:



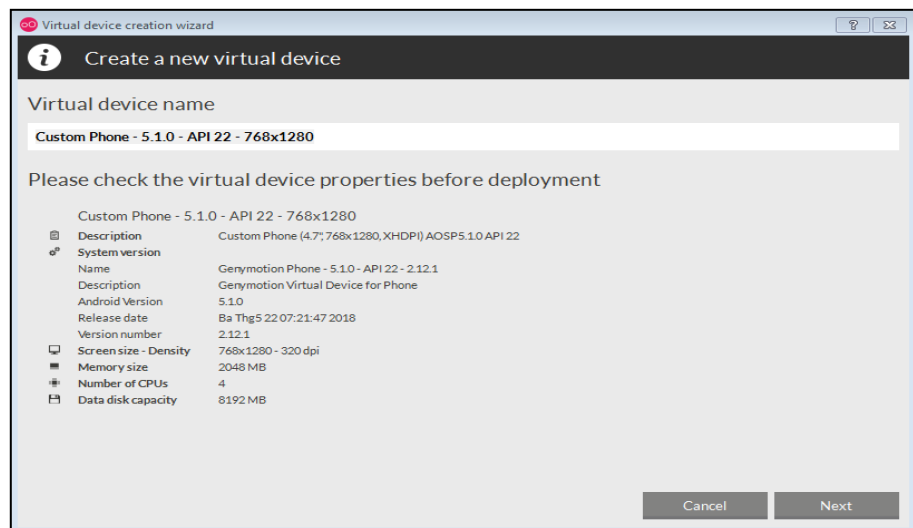
Hình 2.2.3.2. Giao diện chính của Genymotion

- Ta cần đăng nhập vào tạo một cái máy ảo
- Các phím chức năng của máy ảo Genymotion
  - **Start:** Bắt đầu khởi động máy ảo
  - **Add:** Tạo máy ảo
  - **Setting:** Cài đặt
- Nhấn vào Add để tạo máy ảo.



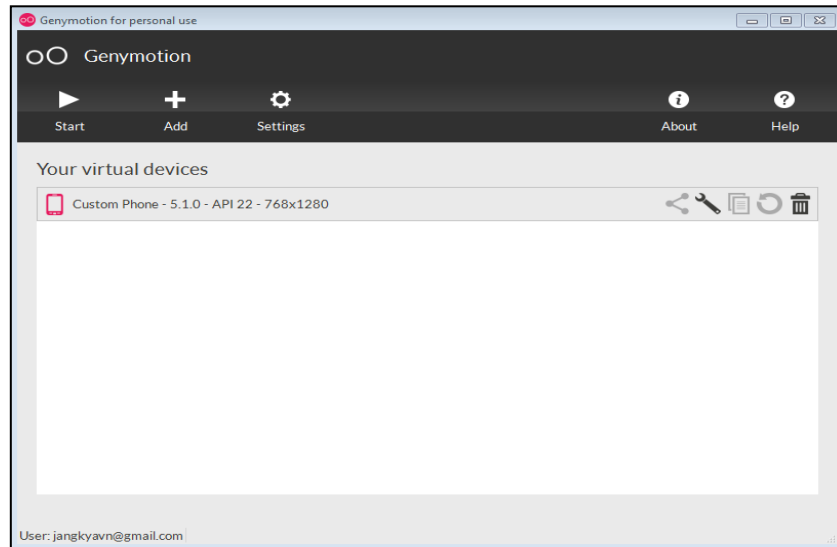
Hình 2.2.3.3. Giao diện chọn máy ảo

- Sau đó chọn một hoặc nhiều thiết bị theo ý muốn (như ở đây chúng ta chọn Custom Phone - 5.1.0 - API 22 - 768x1280 ) :
  - **Custom Phone**: là tên thiết bị máy ảo
  - **5.1.0**: là phiên bản hệ điều hành Android
  - **API**: Application Programming Interface (giao diện lập trình ứng dụng)
  - **768x1280**: Độ phân giải màn hình
- Bạn chọn thiết bị sau đó nhấn Next sẽ có giao diện thông tin thiết



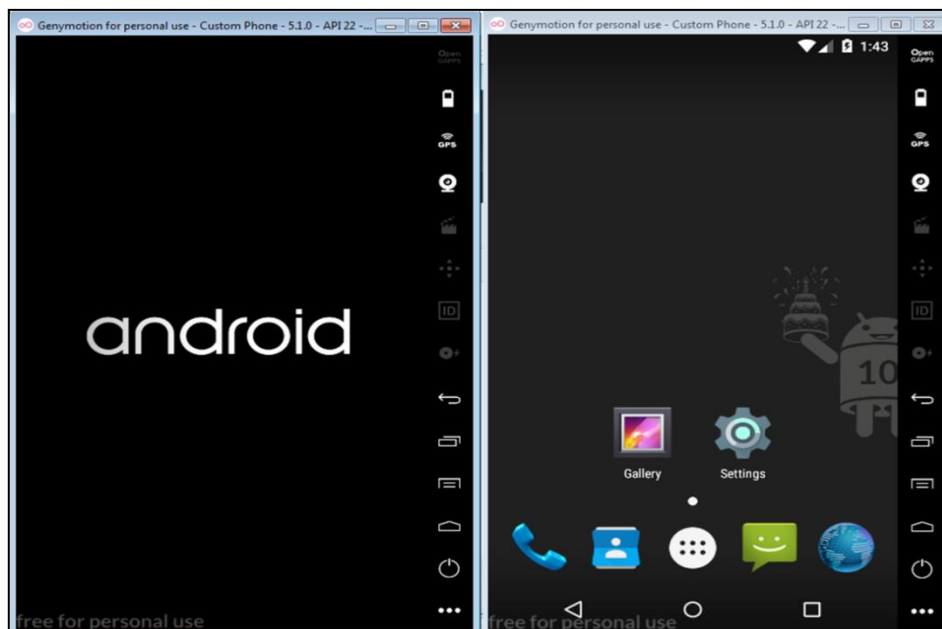
Hình 2.2.3.4. Giao diện xem thông tin máy ảo đang chọn

- Sau đó tiếp tục nhấn Next để hoàn thành cài máy ảo.



Hình 2.2.3.5. Giao diện hiển thị thông tin máy ảo sau khi được tạo

- Ta tiếp tục nhấn vào thiết bị và nhấn Start để khởi động máy ảo.

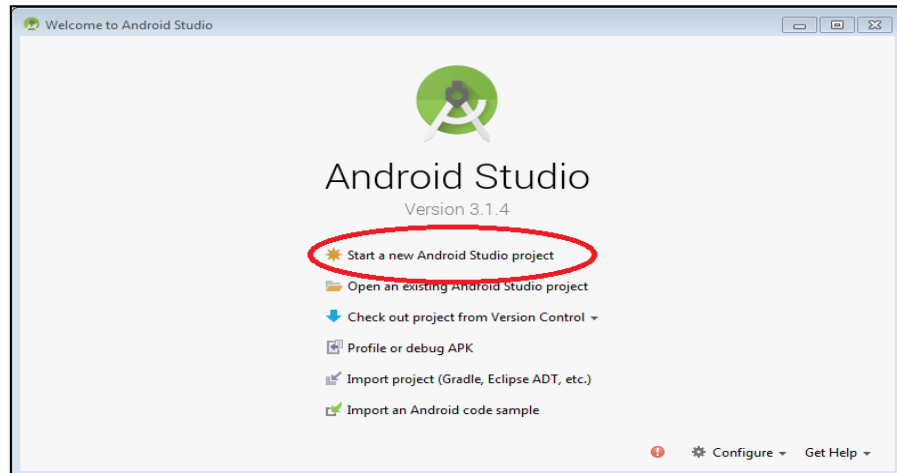


Hình 2.2.3.6. Giao diện màn hình đang khởi động và khi khởi động xong của máy ảo Genymotion

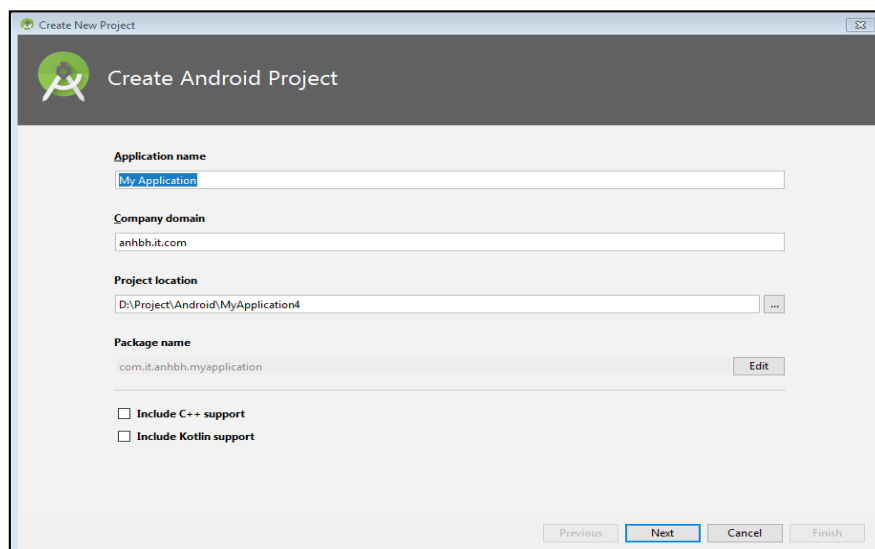
## 2.3. Cấu Trúc Dự Án Android Studio

### 2.3.1. Tạo Mới Một Project

- Click vào dòng “Start a new Android Studio project” như hình 2.3.1.1.



Hình 2.3.1.1. Giao diện cửa sổ đầu tiên khi chọn Android Studio

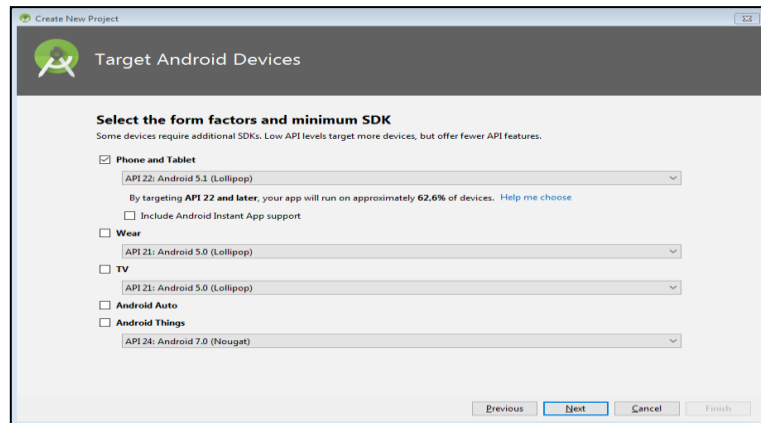


Hình 2.3.1.2. Giao diện tạo project Android

- Khi đó xuất hiện màn hình tạo project Android gồm các thành phần sau:
  - **Application Name:** Tên ứng dụng muốn đặt
  - **Company Domain:** Tên domain công ty, thường được dùng để kết hợp với tên Application để tạo thành Package (chú ý viết thường hết và có ít nhất 1 dấu chấm).

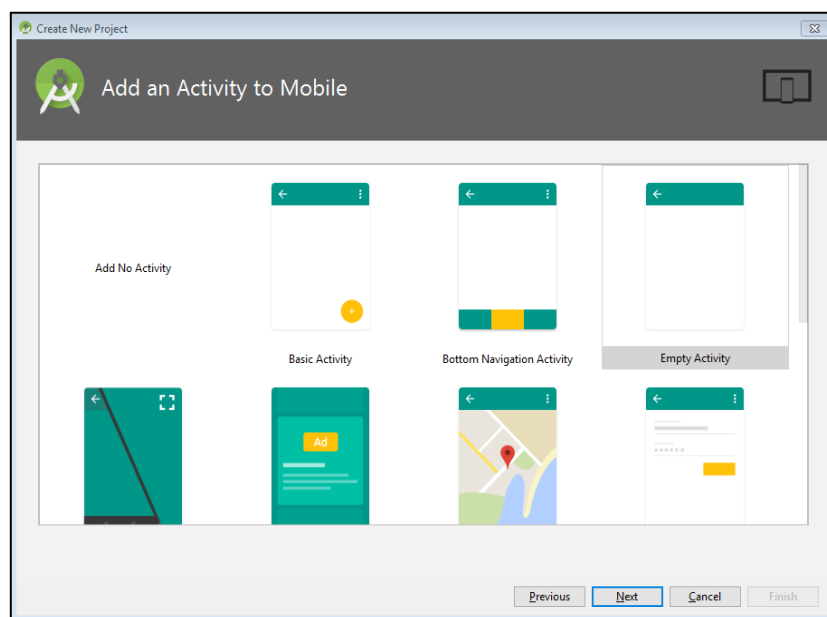
- **Package name:** Nó sẽ tự động nối ngược Company Domain với Application name
- **Project location:** Là nơi lưu trữ ứng dụng
- Sau đó nhấp Next để tiếp tục.

### 2.3.2. Cài Đặt Một Project



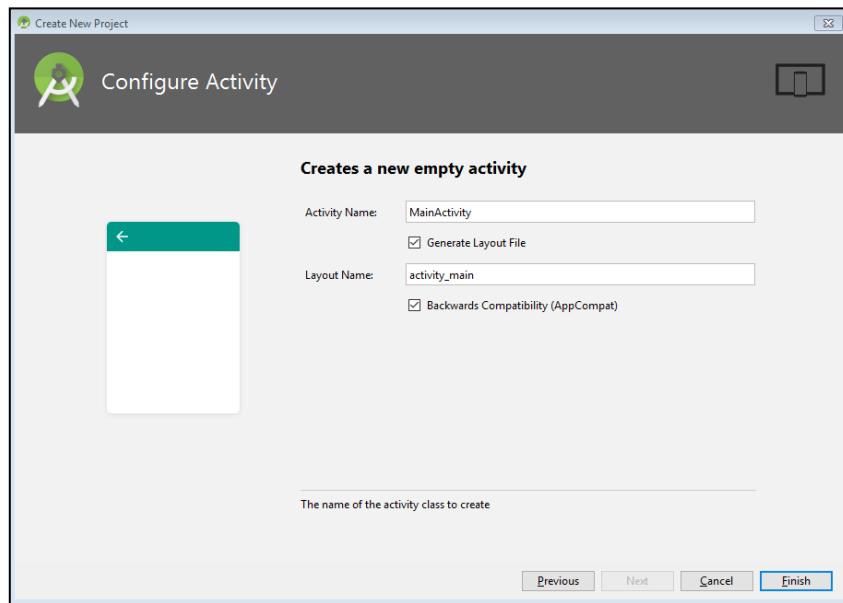
Hình 2.3.2.1. Giao diện chọn thiết bị và phiên bản hệ điều hành

- Ở hộp thoại trên cho phép ta lựa chọn là ứng dụng sẽ được viết cho những thiết bị nào (Phone and Tablet, TV, Wear)
- Chọn phiên bản Android phù hợp rồi nhấn Next.



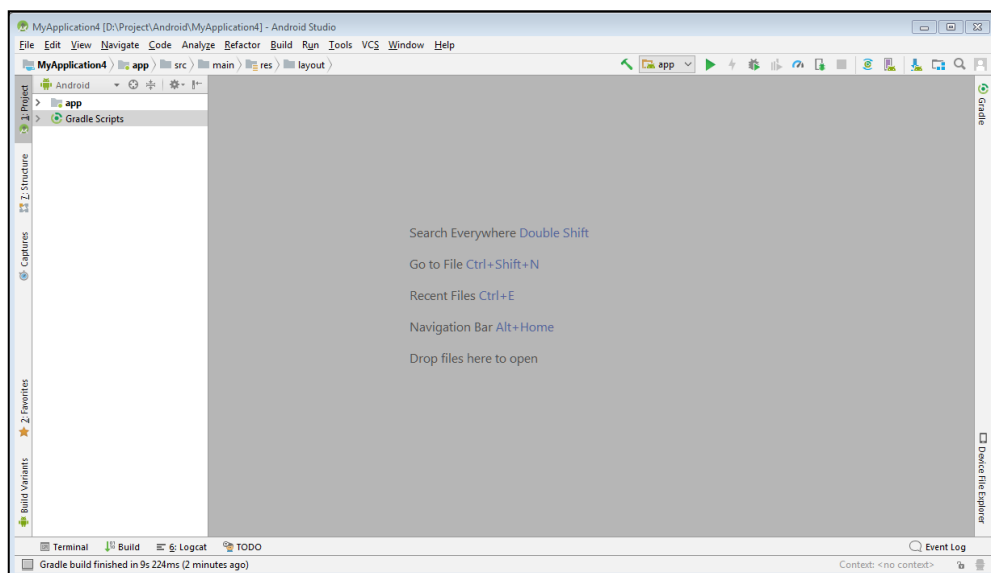
Hình 2.3.2.2. Chọn kiểu màn hình cho ứng dụng

- Chọn **Empty Activity** rồi bấm Next



Hình 2.3.2.3. Đặt tên cho màn hình

- **Activity Name:** Tên class Activity (java) để ta viết mã lệnh
- **Layout Name:** Tên file XML làm giao diện cho Activity Name.
- Sau khi cấu hình xong, bấm Finish
- Khi build xong mặc định có màn hình như hình 2.3.2.4.

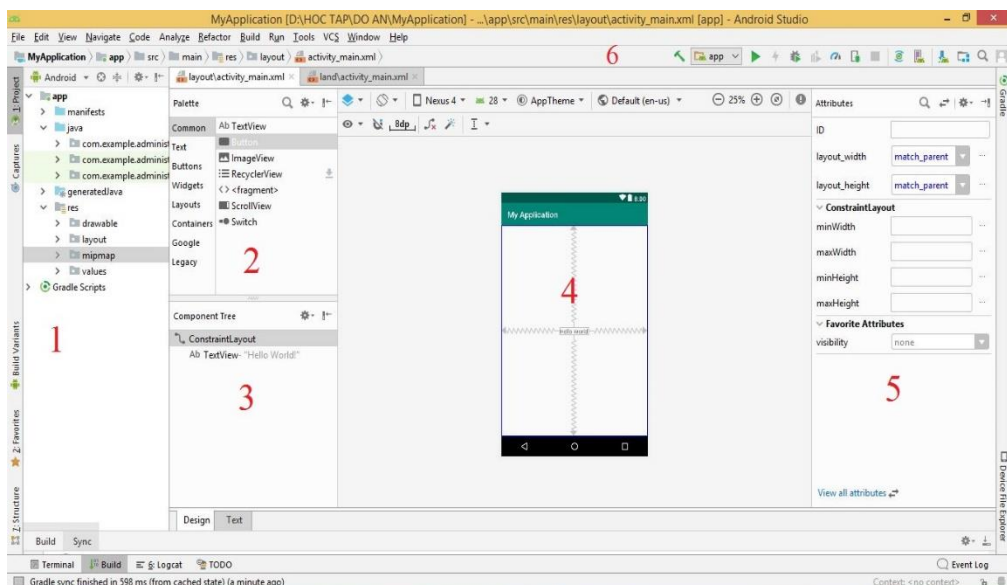


Hình 2.3.2.4. Giao diện sau khi ra tạo thành công project



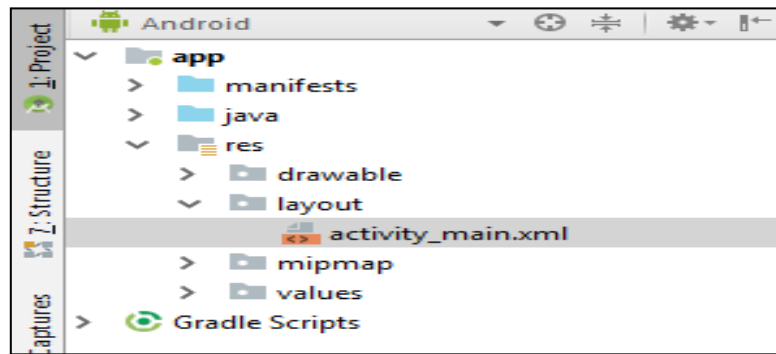
### 2.3.3. Màn hình làm việc của dự án Android Studio

- Theo mặc định Android Studio hiển thị các files trong project theo góc nhìn Android. Góc nhìn này Android Studio sẽ tổ chức các files theo 3 module:
  - **manifests:** chứa file AndroidManifest.xml.
  - **java:** chứa các file mã nguồn Java.
  - **res:** chứa tất cả các file layout, xml, giao diện người dùng (UI), ảnh. Mặc định **activity\_main.xml** sẽ cho ta có màn hình như hình 2.3.3.1.



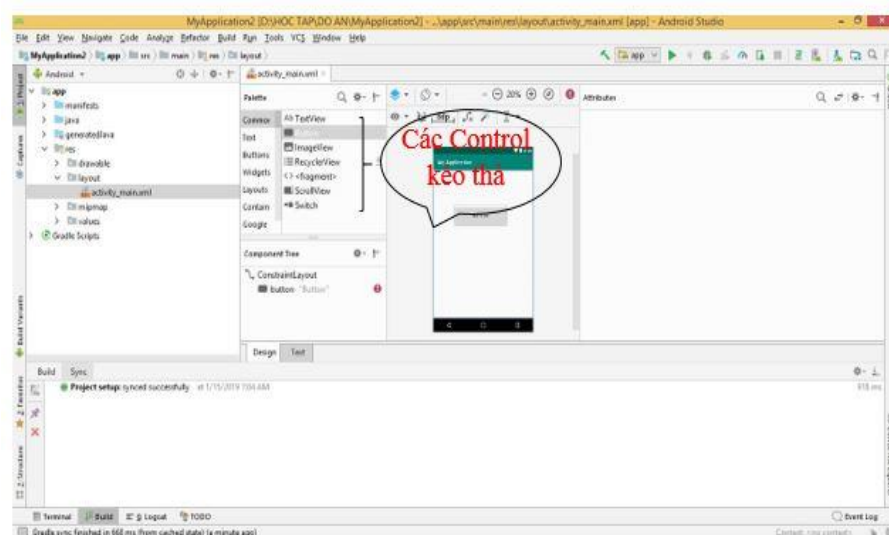
Hình 2.3.3.1. Các vùng làm việc trong Android Studio

- Ở hình 2.3.7 tạm thời chia làm 6 vùng làm việc mà lập trình viên thường tương tác:
  - **Vùng 1:** Là nơi cấu trúc hệ thống thông tin của ứng dụng, Ta có thể thay đổi cấu trúc hiển thị (thường để mặc định là Android).



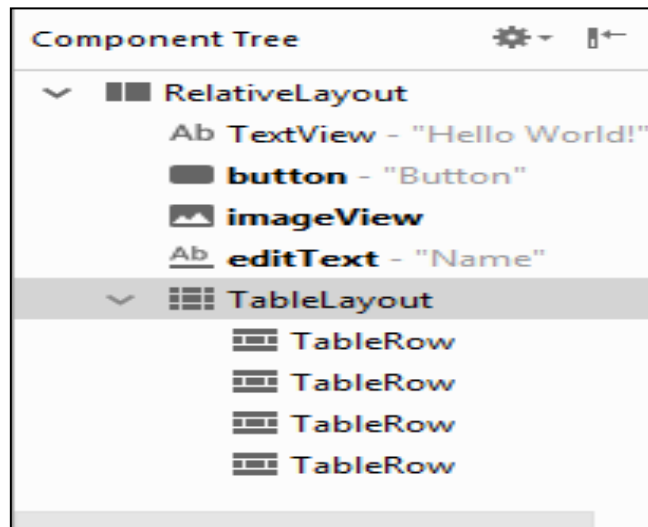
Hình 2.3.3.2. Cấu trúc thư mục dưới dạng Android

- **Vùng 2** : Là vùng khá quan trọng cho những bạn mới bắt đầu lập trình, nó là nơi hiển thị các Control mà Android hỗ trợ, cho phép bạn kéo thả trực tiếp vào vùng 4 để thiết kế.



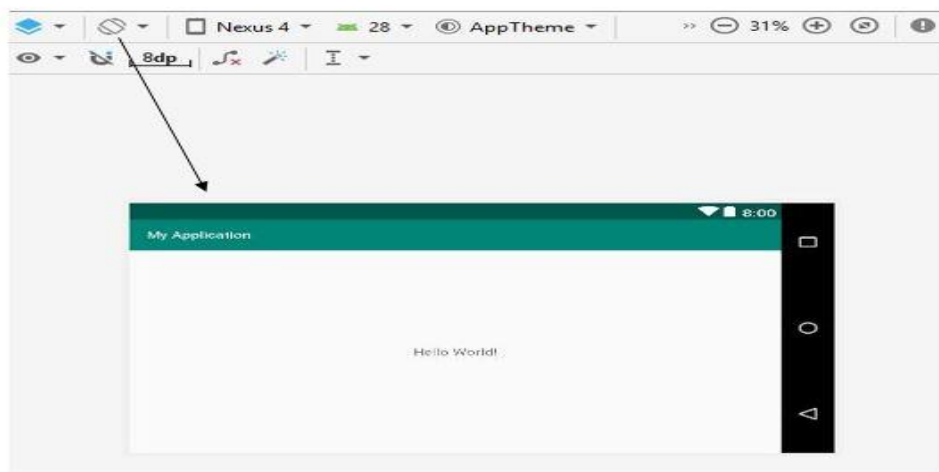
Hình 2.3.3.3. Các control để kéo thả vào màn hình

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



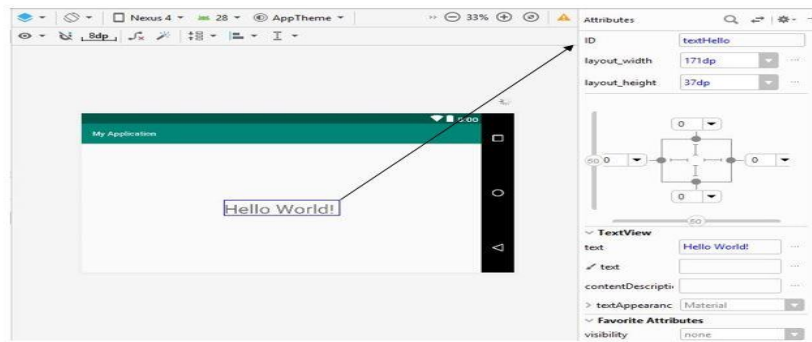
Hình 2.3.3.4. Cấu trúc giao diện theo dạng cây

- **Vùng 4** : Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiểu chính control.



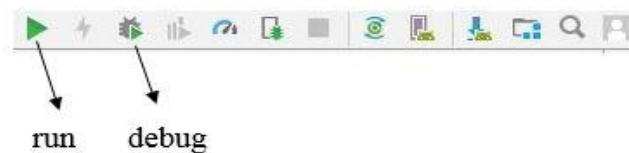
Hình 2.3.3.5. Vùng giao diện để các control kéo thả vào  
Vùng 4 ta có thể chọn cách hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉ control, lựa chọn loại thiết bị hiển thị....

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



Hình 2.3.3.6. Vùng hiện thị thuộc tính của control nằm trên màn hình

- **Vùng 6** : Là vùng các chức năng quan trọng thường dùng trong Android Studio.

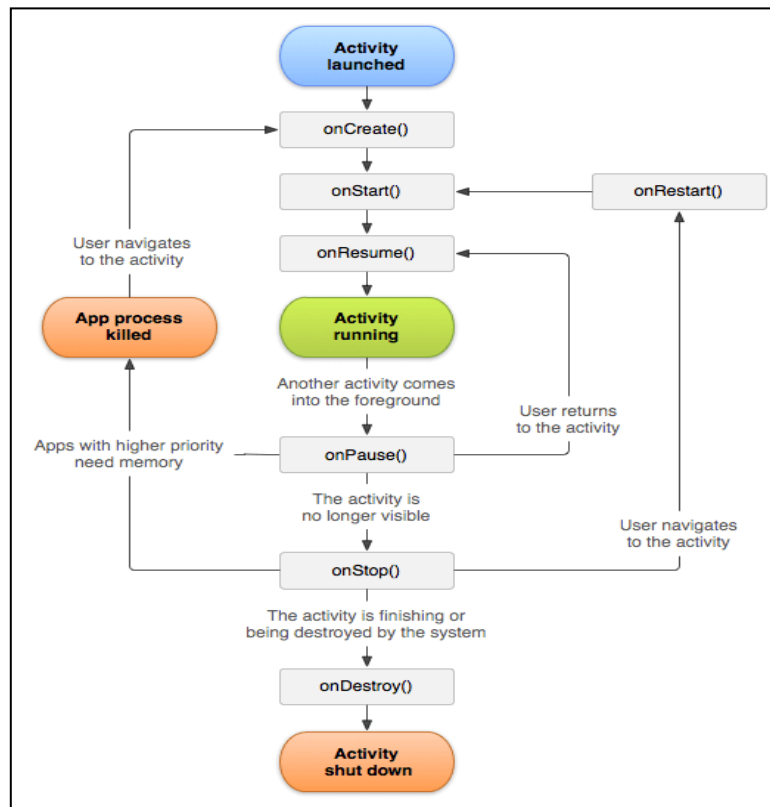


Hình 2.3.3.7. Các nút quan trọng trên Android Studio

## 2.4. Các Thành Phần Trong Một Ứng Dụng Android

Các thành phần cơ bản nhất tạo nên một ứng dụng Android. Các thành phần này giúp cho chúng ta có một cái nhìn tổng quan về cấu trúc ứng dụng.

- **Activity** : là một thành phần của ứng dụng Android nó sẽ biểu diễn giao diện của một màn hình nào đó trong ứng dụng của bạn. Một ứng dụng có thể sẽ có nhiều màn hình và mỗi màn hình có thể là một Activity. **Class Activity** định nghĩa nhiều phương thức cho các sự kiện, bạn không cần phải thực thi tất cả các phương thức này, tuy nhiên việc hiểu rõ và áp dụng chúng vào ứng dụng của bạn khá quan trọng, giúp cho ứng dụng của bạn có các hành vi như mong đợi của người dùng[7].



Hình 2.4.1. Vòng đời của Activity

- **onCreate()**: Nó là phương thức đầu tiên được gọi dùng để tạo một **activity** vào lần đầu tiên **activity** được gọi.
- **onStart()**: Sẽ được gọi khi nó hiện hữu với người dùng.
- **onResume()**: Sẽ được gọi khi người dùng tương tác với các ứng dụng.
- **onPause()**: Tạm dừng một **activity**, không nhận dữ liệu do người dùng nhập vào và không thể thực thi lệnh nào. Phương thức này được gọi khi **activity** hiện tại đang được tạm dừng, và **activity** trước đó đang được tiếp tục
- **onStop()**: Được gọi khi một activity đã không được nhìn thấy trong thời gian dài.
- **onDestroy()**: Được gọi trước khi hệ thống hủy **activity**.
- **onRestart()**: Được gọi khi activity cần được dùng trở lại sau khi bị gọi **onStop()**.

- **Service:** thành phần chạy ẩn trong Android từ lúc chạy cho tới khi thiết bị cầm tay tắt đi. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.  
VD: một chiếc MP3 chơi các bài nhạc được lưu trong hàng đợi, ngay cả khi người sử dụng chạy vào ứng dụng khác thì ứng dụng của bạn vẫn phải cần thực hiện các Services để trình diễn các nhiệm vụ mà không có một giao diện người dùng cụ thể nào cả.
- **Broadcast Receiver:** Thành phần thu nhận các Intent bên ngoài gửi tới.  
Ví dụ: bạn viết 1 chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần 1 BR để nhận biết các Intent là các cuộc gọi tới.
- **Content Provider:** Là một kho dữ liệu chia sẻ, nó cung cấp cách truy cập dữ liệu và giúp chia sẻ dữ liệu giữa các ứng dụng.
- **Intent:** Là thành phần tối quan trọng trong một ứng dụng android, nó là một định dạng chuẩn dùng để đưa dữ liệu lên đó và khởi tạo một Activity hay truyền dữ liệu giữa các Activity, thậm chí là truyền dữ liệu giữa các ứng dụng đang thực thi, hoặc nó có thể được dùng để khởi tạo một Service.  
Ví dụ: Khi bạn mở một trang web, thì phải có một thành phần nào đó gửi dữ liệu đi để tạo một activity hiển thị trang web, đó chính là intent.
- **Notification:** Đưa ra các thông báo khi ứng dụng android đang được thực thi và không làm ảnh hưởng gì đến ứng dụng thực thi. Thông thường các lập trình viên hay dùng dạng thông báo là Toast và AlertDialog.
- **View:** View là thành phần thiết yếu của giao diện người dùng, nó bao gồm các control như button, radio button, textview, checkbox, các layout,.... trong java các thành phần này được gọi là components chứ

không phải view.

Khi một view được tạo các lập trình viên thường quan tâm tới các thông số cơ bản sau:

- Các thuộc tính cần thiết về kích thước, màu sắc,...
- Cho phép view này được nhìn thấy hoặc bị ẩn
- Xử lý sự kiện khi người dùng thao tác với view này (listener)

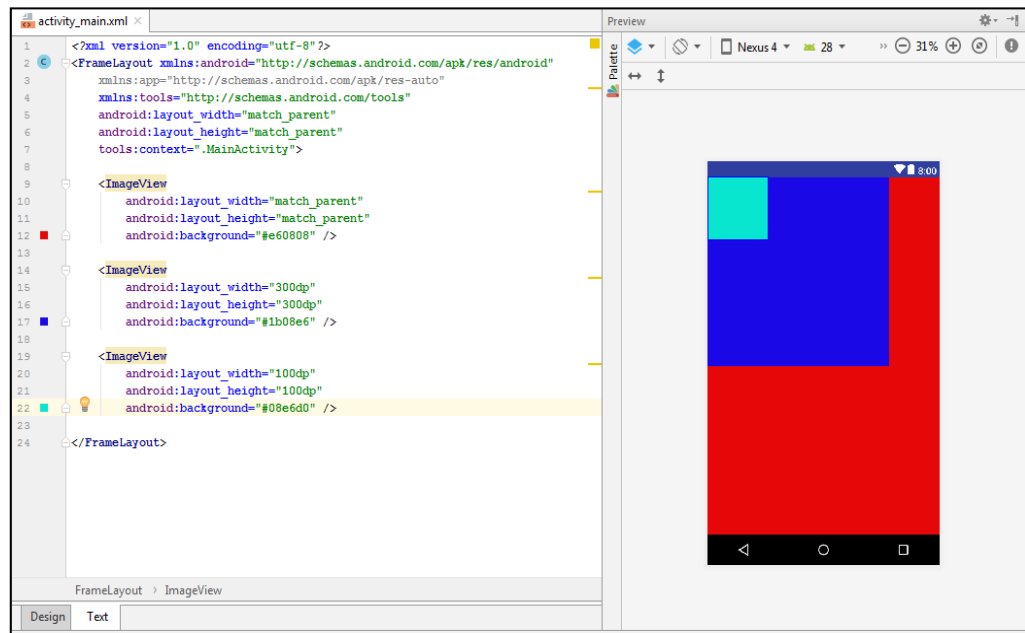
❖ **Lưu ý:** Activity, Service, Broadcast Receiver và Content Provider là những thành phần chính tạo nên ứng dụng Android do đó chúng phải được khai báo trong AndroidManifest.xml trước khi sử dụng

## 2.5. Tạo Giao Diện Chương Trình Trong Android Studio

### 2.5.1. Giới Thiệu Android Layout

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

- **FrameLayout:** Là loại layout cơ bản nhất, nó sẽ được dùng nhiều khi ta sử dụng vẽ giao diện nâng cao sau này. Khi ta kéo các control vào thì mặc định các control sẽ nằm ở vị trí trên cùng bên trái. Các control khi được kéo vào framelayout sẽ bị đè lên nhau, control sau sẽ đè lên control trước. Cách duy nhất để căn các control vào giữa là sử dụng thuộc tính **android:layout\_gravity="center"**. Ta có thể tham khảo đoạn XML sau để hiểu thêm về framelayout như hình 2.5.1.1.



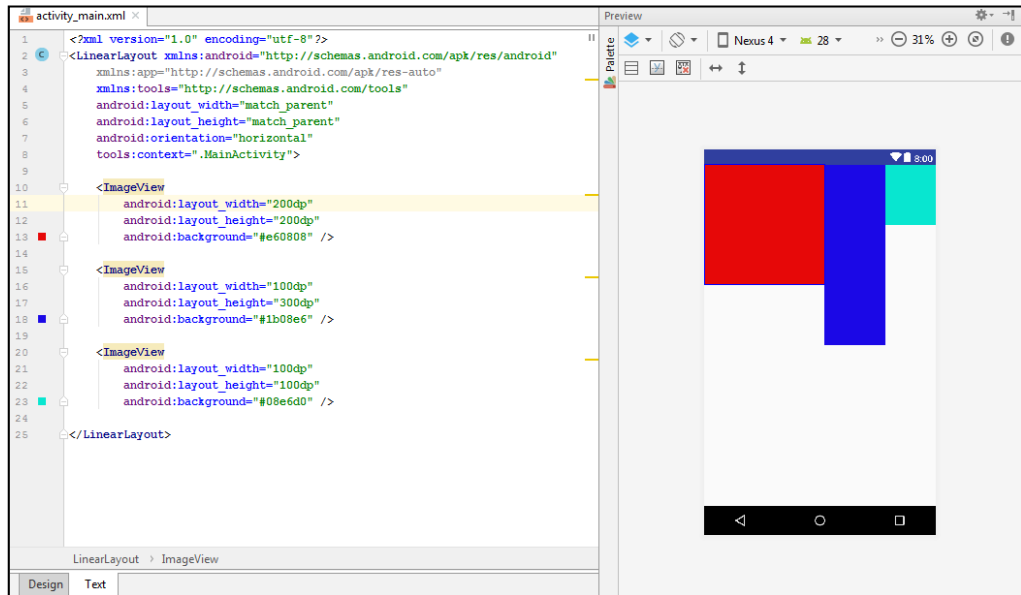
Hình 2.5.1.1. Giao diện sử dụng FrameLayout

- **LinearLayout:** Layout này cho phép ta vẽ giao diện theo 2 hướng, từ trái qua phải hoặc từ trên xuống dưới. Để xét chiều cho các control trong layout ta sử dụng thuộc tính **orientation**.
  - `Android:orientation="horizontal"` : Xếp các control từ trái sang phải (theo hàng).
  - `Android:orientation="vertical"` : Xếp các control từ trên xuống dưới (theo cột)

Với những giao diện có độ phức tạp vừa phải thì dùng LinearLayout là rất hiệu quả, rất thuận tiện trong thiết kế và đi bảo trì ứng dụng sau này. Sau đây là đoạn XML demo cách sử dụng layout:

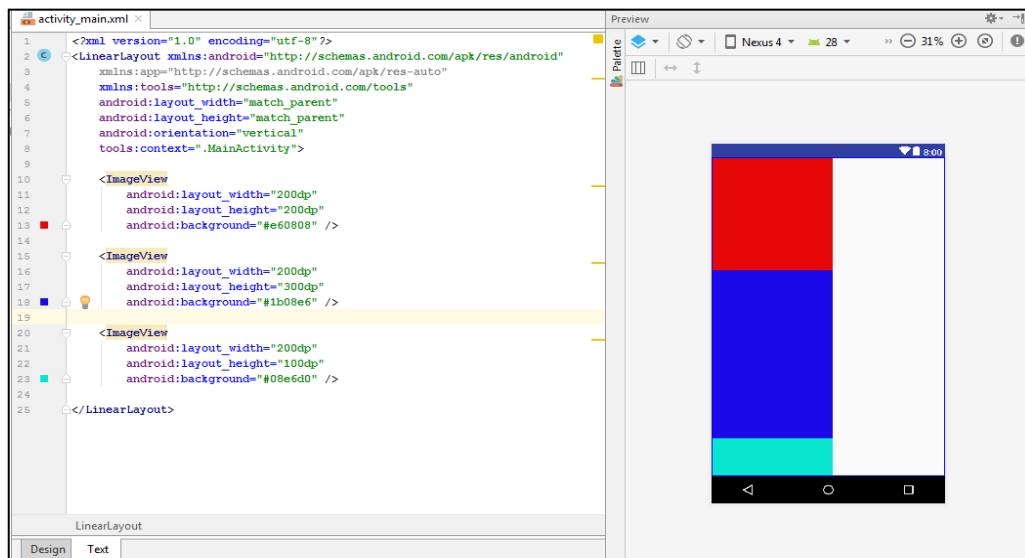


- Theo hàng



Hình 2.5.1.2. Giao diện LinearLayout theo chiều ngang

- Theo cột

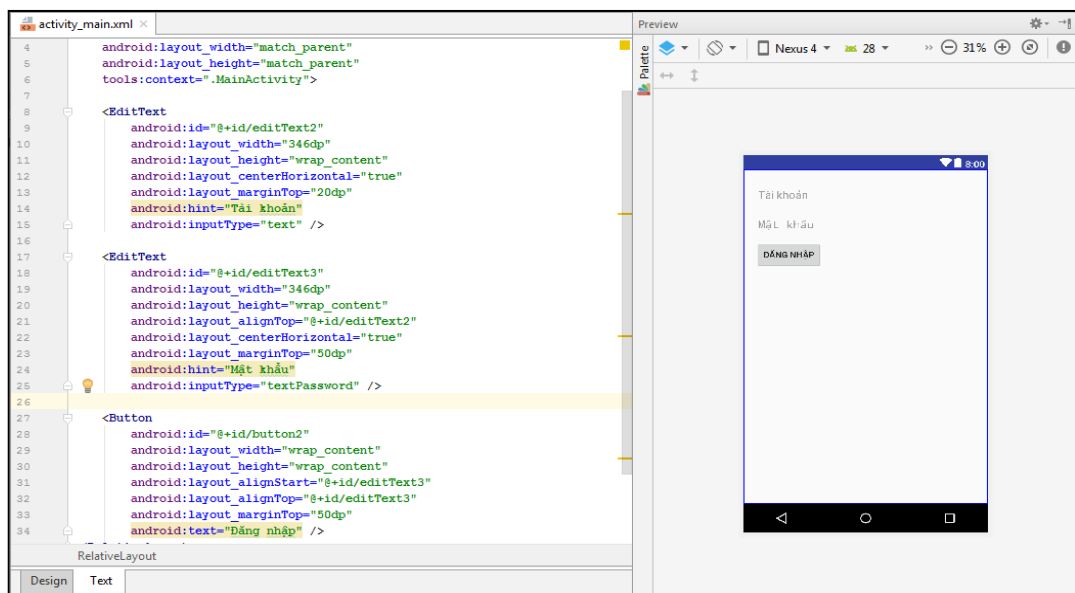


Hình 2.5.1.3. Giao diện LinearLayout theo chiều dọc

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

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

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



Hình 2.5.1.4. Giao diện RelativeLayout

## 2.5.2. Giới Thiệu Một Số Android View Cơ Bản

- **TextView**: là view sử dụng để hiển thị text màn hình. TextView được định nghĩa bởi thẻ trong xml.
- **EditText**: là view dùng để lấy giá trị từ người dùng nhập vào. EditText được định nghĩa bởi thẻ trong xml.
- **ImageView**: là một view sử dụng rất nhiều trong ứng dụng android, ImageView sử dụng để hiển thị hình ảnh.
- **Button**: là view được sử dụng khá nhiều trong android, hầu như sử dụng ở mọi nơi cùng với EditText, TextView. Button có chức năng là làm nhiệm vụ nào đó khi mà người dùng click trong phương thức onClick.

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

### 2.5.3. Bắt Và Xử Lý Sự Kiện Trên Giao Diện

Sự kiện là một cách hữu ích để thu thập dữ liệu về sự tương tác của người dùng với các thành phần tương tác của ứng dụng. Giống như bấm vào một nút hoặc chạm vào màn hình cảm ứng, vv. Ta có thể nắm bắt những sự kiện trong chương trình và có những xử lý thích hợp theo yêu cầu[6].

#### Có hai khái niệm liên quan đến quản lý sự kiện Android

- **Event Listeners** là một interface. Event Listeners được sử dụng để đăng ký sự kiện cho các thành phần trong UI. (Đăng ký sự kiện)
- **Event Handlers** – Là phương thức xử lý khi phát sinh sự kiện. (Xử lý sự kiện)

#### Một số sự kiện thường gặp trong Android

| Event Handler | Event Listener Name   |
|---------------|---|
| onClick()     | <b>OnClickListener()</b><br>Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, image vv. Chúng ta sẽ sử dụng phương onClick() để xử lý sự kiện. |
| onLongClick() | <b>OnLongClickListener()</b><br>Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, image vv. trong một hoặc nhiều giây. Chúng ta sẽ sử dụng     |

|                       |  |
|-----------------------|--|
|                       | phương onLongClick() để xử lý sự kiện.   |
| onFocusChange()       | <b>OnFocusChangeListener()</b><br>Sự kiện phát sinh khi widget mất focus.  |
| onKey()               | <b>OnFocusChangeListener()</b><br>Sự kiện phát sinh khi người dùng focus trên widget và nhấn (presse) hoặc thả (release) một phím trên thiết bị. |
| onTouch()             | <b>OnTouchListener()</b><br>Sự kiện phát sinh khi người dùng nhấn phím, nhả phím, hoặc bất kỳ cử chỉ chuyển động trên màn hình.                  |
| onMenuItemClick()     | <b>OnMenuItemClickListener()</b><br>Sự kiện phát sinh khi người dùng chọn một mục trong menu.  |
| onCreateContextMenu() | <b>OnCreateContextMenuListener()</b><br>Sự kiện phát sinh khi người dùng chọn một mục trong menu ngữ cảnh (Context Menu)                         |

## CHƯƠNG 3: KỸ THUẬT XÂY DỰNG ỨNG DỤNG THỜI TIẾT

### 3.1. Tìm Hiểu Về Dịch Vụ OpenWeatherMap

#### 3.1.1. Khái Niệm

OpenWeatherMap API này cung cấp dịch vụ dữ liệu thời tiết và dự báo miễn phí, thích hợp cho bất kỳ dịch vụ bản đồ như các ứng dụng web và điện thoại thông minh. Ý tưởng được lấy cảm hứng từ OpenStreetMap và Wikipedia nhằm cung cấp thông tin miễn phí và sẵn có cho mọi người.

OpenWeatherMap cung cấp nhiều dữ liệu thời tiết như bản đồ thời tiết hiện tại, dự báo tuần, lượng mưa, gió, mây, dữ liệu từ các trạm thời tiết và nhiều thứ khác. Dữ liệu thời tiết được nhận từ các dịch vụ phát sóng khí tượng toàn cầu và hơn 40.000 trạm khí tượng.

Bạn có thể nhận được bất kỳ dữ liệu thời tiết nào cho ứng dụng của bạn bằng cách sử dụng chuỗi JSON hoặc XML.

#### 3.1.2. Các Tính Năng Nổi Bật Của API Thời Tiết Open Weather Map

- *Dữ liệu thời tiết hiện tại (Current weather data)*

Truy cập dữ liệu thời tiết hiện tại cho bất kỳ vị trí nào bao gồm hơn 200.000 thành phố. Thời tiết hiện tại thường xuyên được cập nhật dựa trên các mô hình và dữ liệu toàn cầu từ hơn 40.000 trạm thời tiết. Dữ liệu có sẵn ở định dạng JSON, XML hoặc HTML

- *Dự báo thời tiết 5 ngày tiếp theo cập nhật liên tục sau 3 giờ.*

Dự báo 5 ngày có sẵn tại bất kỳ địa điểm hoặc thành phố nào. Dự báo 5 ngày bao gồm dữ liệu thời tiết cứ sau 3 giờ. Dự báo có sẵn trong JSON và XML.

- *Dự báo 16 ngày tiếp theo*

Dự báo 16 ngày có sẵn tại bất kỳ địa điểm hoặc thành phố nào. Dự báo 16 ngày bao gồm thời tiết hàng ngày. Dự báo có sẵn trong JSON và XML

- *Dữ liệu lịch sử thời tiết*

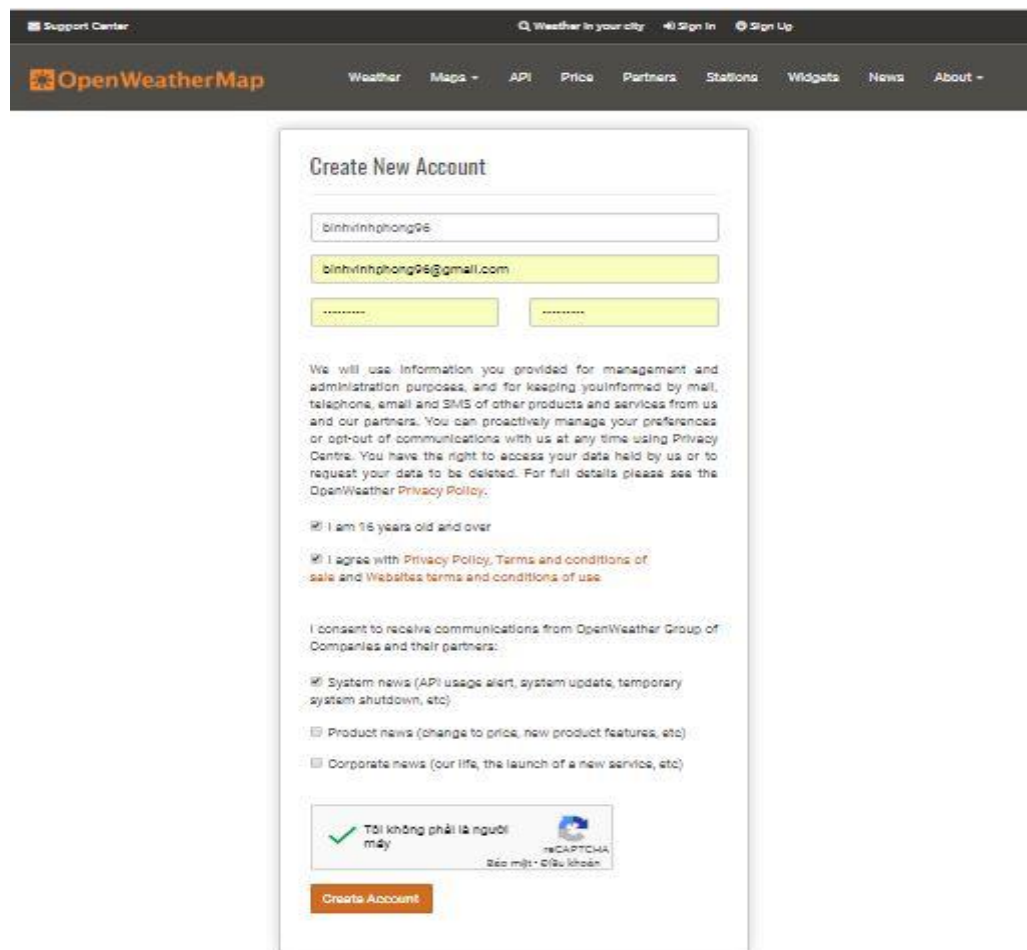
API OpenWeatherMap cung cấp dữ liệu lịch sử thời tiết cho hơn 37.000 thành phố. Dữ liệu lịch sử trong 1 tháng trước đó có sẵn trong tài khoản Starter, cho 1 năm trước trong tài khoản Medium và trong 5 năm trong History Bulk

### 3.2. Đăng Ký Và Kích Hoạt Sử Dụng Key API Openweathermap

- *Đầu tiên ta đăng ký tài khoản Open weather API để lấy API Key*

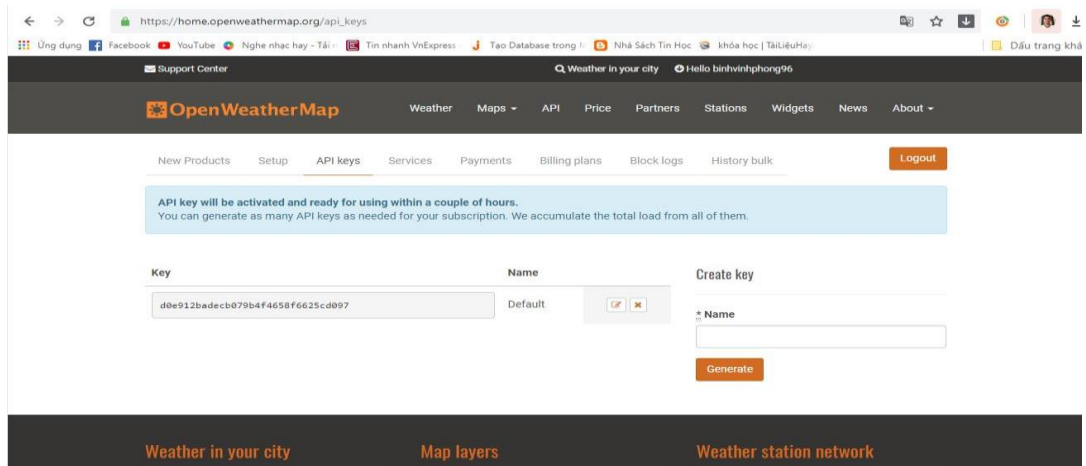
Ta truy cập đường link

[https://home.openweathermap.org/users/sign\\_up](https://home.openweathermap.org/users/sign_up) để đăng ký tài khoản miễn phí của OpenWeatherMap và nhận key miễn phí duy nhất cho tài khoản đó



Hình 3.2.1. Trang chủ đăng ký tài khoản OpenWeatherMap

- Khi đăng ký và đăng nhập thành công, chúng ta sẽ có API key bên dưới:



Hình 3.2.2. Trang chủ OpenWeatherMap

- Kiểm tra chuỗi JSON truy vấn dữ liệu từ API

URL lấy thông tin thời tiết theo kinh độ vĩ độ

<http://api.openweathermap.org/data/2.5/weather?q=HaiPhong&appid=d0e912badecb079b4f4658f6625cd097>

Được kết quả



Hình 3.2.3 : URL thông tin thời tiết

### 3.3. Thiết Kế Giao Diện Ứng Dụng Bằng Ngôn Ngữ XML

- Tại màn hình chính sẽ tiến hành tạo màn hình thời tiết trong ngày giao diện đơn giản để hiển thị các thông tin như tên thành phố, quốc gia, trạng thái .....

Được thiết kế theo TextView và ImageView

### activity\_main.xml

|   |  |
|---|--|
| <pre> &lt;LinearLayout     android:layout_width="match_parent"     android:layout_height="match_parent"     android:background="@drawable/weather"     android:orientation="vertical"&gt;     &lt;TextView         android:layout_marginTop="70dp"         android:id="@+id/txttp"         android:layout_width="match_parent"         android:layout_height="wrap_content"         android:text="Tên thành phố"         android:textSize="15dp" /&gt;     &lt;TextView         android:id="@+id/txtqg"         android:layout_width="match_parent"         android:layout_height="wrap_content"         android:layout_marginTop="2dp"         android:text="Tên quốc gia"         android:textSize="15dp" /&gt;     &lt;ImageView         android:id="@+id/imgmay"         android:layout_width="120dp"         android:layout_height="120dp"         android:layout_gravity="center"         android:layout_marginTop="2dp"         app:srcCompat="@drawable/cloudse" /&gt;     &lt;TextView         android:id="@+id/txtnhietdo"         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:layout_gravity="center"         android:layout_marginTop="10dp"         android:text="Nhiệt độ"         android:textColor="#f20" </pre> | <pre> &lt;LinearLayout     android:layout_width="0dp"     android:layout_height="wrap_content"     android:layout_weight="1"     android:gravity="center"     android:orientation="vertical"&gt;     &lt;ImageView         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:src="@drawable/dropa" /&gt;     &lt;TextView         android:id="@+id/txtdrop"         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:text="Độ ẩm"         android:textSize="20dp" /&gt;     &lt;/LinearLayout&gt;     &lt;LinearLayout         android:layout_width="0dp"         android:layout_height="wrap_content"         android:layout_weight="1"         android:gravity="center"         android:orientation="vertical"&gt;     &lt;ImageView         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:src="@drawable/clouda" /&gt;     &lt;TextView         android:id="@+id/txtcloud"         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:text="Mây"         android:textSize="20dp" /&gt;     &lt;/LinearLayout&gt; </pre> |
|---|--|



|  |   |
|--|---|
| <pre>         android:textSize="25dp" /&gt;     &lt;TextView         android:id="@+id/txttrangthai"         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:layout_gravity="center"         android:layout_marginTop="10dp"         android:text="Trạng thái"         android:textSize="20dp" /&gt;     &lt;LinearLayout         android:layout_width="match_parent"         android:layout_height="wrap_content"         android:layout_marginTop="20dp"         android:orientation="horizontal"         android:weightSum="3"&gt;     </pre> | <pre>     &lt;LinearLayout         android:layout_width="0dp"         android:layout_height="wrap_content"         android:layout_weight="1"         android:gravity="center"         android:orientation="vertical"&gt;         &lt;ImageView             android:layout_width="wrap_content"             android:layout_height="wrap_content"             android:src="@drawable/winda" /&gt;         &lt;TextView             android:id="@+id/txtwind"             android:layout_width="wrap_content"             android:layout_height="wrap_content"             android:text="Gió"             android:textSize="20dp" /&gt;         &lt;/LinearLayout&gt;     &lt;/LinearLayout&gt;     </pre> |
|--|---|

- Tiếp đó tạo hiển thị thông tin thời tiết 4 ngày tiếp theo và thời gian cập nhật thời tiết, được hiển thị theo RecyclerView và TextView.

### activity\_main.xml

|  |   |
|--|---|
| <pre> &lt;android.support.v7.widget.RecyclerView     android:id="@+id/recycler_view"     android:layout_width="400dp"     android:layout_height="120dp"     android:layout_gravity="center"     android:layout_marginTop="20dp"     android:orientation="horizontal"     app:layoutManager="android.support.v7.widget.LinearLayoutManager"     app:reverseLayout="true" /&gt;     </pre> | <pre> &lt;TextView     android:id="@+id/txtday"     android:layout_width="wrap_content"     android:layout_height="wrap_content"     android:layout_gravity="center"     android:layout_marginTop="0dp"     android:text="Ngày cập nhật"     android:textColor="#0a0a0a"     android:textSize="20dp" /&gt;     </pre> |
|--|---|

- Tại giao diện hiển thị thông tin thời tiết 4 ngày tiếp theo

Được hiển thị theo TextView và ImageView

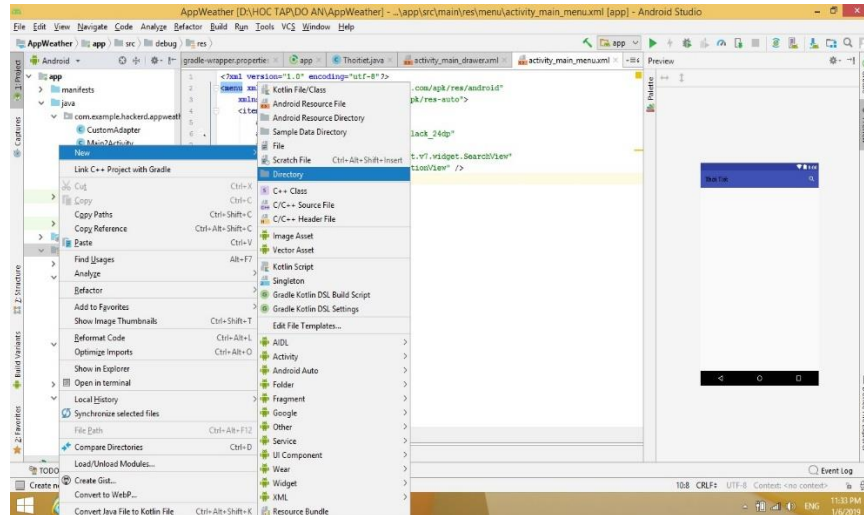
### Recycler\_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical">
    <TextView
        android:id="@+id/tvDay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Chủ nhật" />
    <ImageView
        android:id="@+id/ivImage"
        android:layout_width="70dp"
        android:layout_height="70dp" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/tvMinTemp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="11°C" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text=" - " />
        <TextView
            android:id="@+id/tvMaxTemp"
            android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:text="15°C" />  
</LinearLayout> </LinearLayout>
```

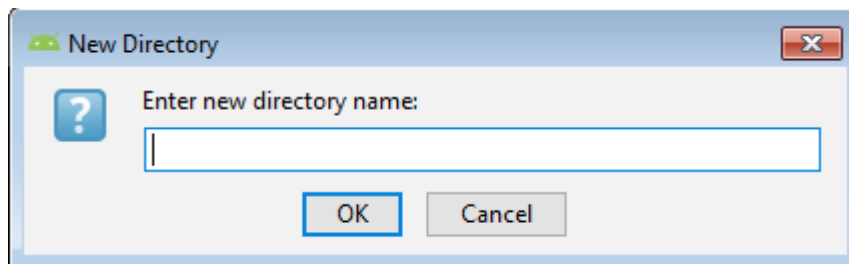
- Tạo menu tìm kiếm theo tên thành phố

Ta vào phần res ở cấu trúc project -> chuột phải -> New -> Directory



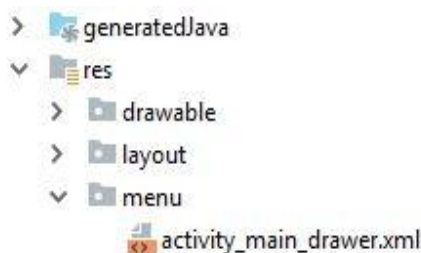
Hình 3.3.1 : Tạo thư mục menu

Đặt tên thư mục cho menu



Hình 3.3.2 : Đặt tên cho thư mục menu

- Click chuột phải vào thư mục menu vừa tạo chọn new và chọn Menu resource file ở đây mình đặt tên cho menu là activity\_main\_menu



Hình 3.3.3 : Menu sau khi được tạo

- Giao diện chính của menu khai báo trong file **activity\_main\_menu.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"

</menu>
```

### Giao diện tìm kiếm theo tên thành phố

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto">
<item
android:id="@+id/menu_search"
android:icon="@drawable/ic_search_black_24dp"
android:title="Tìm kiếm"
app:actionViewClass="android.support.v7.widget.SearchView"
app:showAsAction="ifRoom|collapseActionView" />
</menu>
```

- Tiếp theo sẽ tiến hành tạo màn hình hiển thị thông tin dự báo thời tiết về 7 ngày tiếp theo

Được hiển thị theo ImageView, TextView, ListView

### **activity\_main2.class**

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

<ImageView
android:id="@+id/imgback"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="5dp"
android:layout_marginTop="5dp"
```

```
app:srcCompat="@drawable/backa"  
tools:layout_editor_absoluteX="2dp"  
tools:layout_editor_absoluteY="7dp" />  
<TextView  
    android:layout_marginLeft="5dp"  
    android:layout_marginTop="3dp"  
    android:id="@+id/txtviewtp"  
    android:textColor="#0037ff"  
    android:textStyle="bold"  
    android:text="Tên thành phố"  
    android:textSize="30dp"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />  
    <ListView  
        android:id="@+id/lvview"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
</ListView>  
</LinearLayout>
```

- Tạo giao diện hiển thị thông tin nhiệt độ cao nhất nhiệt độ thấp nhất ngày tháng, trạng thái và icon được hiển thị bằng TextView và ImageView

### **dong\_listview.class**

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal"  
    android:weightSum="2"  
    android:gravity="center">  
    <LinearLayout  
        android:gravity="center"  
        android:orientation="vertical"
```

```
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content">
    <TextView
        android:gravity="center"
        android:textColor="#0d0d0d"
        android:textSize="20dp"
        android:id="@+id/txtviewngay"
        android:text="Ngày tháng"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:textSize="20dp"
        android:text="Trạng thái"
        android:id="@+id/txtviewtrangthai"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
<LinearLayout
    android:layout_weight="1"
    android:layout_width="0dp"
    android:layout_height="wrap_content">
<LinearLayout
    android:gravity="center"
    android:weightSum="3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:id="@+id/imgvtrangthai"
        android:src="@drawable/clouda"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="100dp" />
    <TextView
        android:id="@+id/txtmax"
```

```
        android:text="Max temp"
        android:textSize="18dp"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/txtmin"
        android:text="Min temp"
        android:textSize="18dp"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content" />
</LinearLayout>
</LinearLayout> </LinearLayout>
```

### 3.4. Kỹ Thuật Lấy Nội Dung Trả Về

**Bước 1:** Tiến hành thêm thư viện dependence vào trong file **build.gradle**:

```
implementation 'com.android.volley:volley:1.1.0
```

```
implementation 'com.squareup.picasso:picasso:2.5.2
```

- **Picasso**: là một thư viện Android mã nguồn mở rất phổ biến, dùng để load hoặc hiển thị hình ảnh trong ứng dụng Android.
- **Volley**: là một thư viện HTTP giúp việc kết nối các ứng dụng Android trở nên dễ dàng hơn và quan trọng nhất là nhanh hơn. Volley có sẵn trên GitHub.

**Bước 2 :** Tiếp theo cấp quyền cho phép ứng dụng lấy dữ liệu từ Internet trong **AndroidManifest**.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hackerd.appweather">
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>
```

**Bước 3:** Tiếp theo khai báo các đối tượng thể hiện vị trí cần lấy thông tin thời tiết chứa các thuộc tính như tên thành phố, nhiệt độ, trạng thái .....

```
Public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
    TextView txttp, txtqg, txtnhietdo, txttrangthai, txtdrop, txtcloud,txtwind, txtday;
    ImageView imgmay;
    String City = "";
    String city;
    RecyclerView recyclerView;
    ArrayList<Thoitiet> arrThoiTiet;
    RecyclerViewAdapter recyclerViewAdapter;
    SearchView searchView;
    DrawerLayout drawerLayout;
    Toolbar toolbar;
    NavigationView navigationView;
    private void Anhxa() {
        txttp = findViewById(R.id.txttp);
        txtqg = findViewById(R.id.txtqg);
        txtnhietdo = findViewById(R.id.txtnhietdo);
        txttrangthai = findViewById(R.id.txttrangthai);
        txtdrop = findViewById(R.id.txtdrop);
        txtcloud = findViewById(R.id.txtcloud);
        txtwind = findViewById(R.id.txtwind);
        txtday = findViewById(R.id.txtday);
        imgmay = findViewById(R.id.imgmay);
        recyclerView = findViewById(R.id.recycler_view);
        arrThoiTiet = new ArrayList<>();
        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        drawerLayout = findViewById(R.id.drawer_layout);
        navigationView = findViewById(R.id.nav_view);
```



**Bước 4:** Sau đó tạo 1 hàm `GetCurrentWeatherData(String data)` để gắn giá trị truyền vào lấy chuỗi JSON từ API theo địa điểm và để chuyển đổi dữ liệu từ JSON trả về thông tin.

### **MainActivity.class**

```
public void GetCurrentWeatherData(String data) {
    RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
    String url = "http://api.openweathermap.org/data/2.5/weather?q=" + data +
"&lang=vi&units=metric&appid=d0e912badecb079b4f4658f6625cd097";
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    String day = jsonObject.getString("dt");
                    String name = jsonObject.getString("name");
                    txtt.setText("Tên thành phố : " + name);
                    long l = Long.valueOf(day);
                    Date date = new Date(l * 1000L);
                    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("EEEE
yyyy-MM-dd HH:mm:ss");
                    String Day = simpleDateFormat.format(date);
                    txtday.setText(Day);
                    JSONArray jsonArrayWeather = jsonObject.getJSONArray("weather");
                    JSONObject jsonObjectWeather = jsonArrayWeather.getJSONObject(0);
                    String status = jsonObjectWeather.getString("description");
                    String icon = jsonObjectWeather.getString("icon");
                    Picasso.with(MainActivity.this).load("http://openweathermap.org/img/w/" +
icon + ".png").into(imgmay);
                    txttrangthai.setText(status);
                    JSONObject jsonObjectMain = jsonObject.getJSONObject("main");
                    String nhietdo = jsonObjectMain.getString("temp");
                    String doam = jsonObjectMain.getString("humidity");
                    Double a = Double.valueOf(nhietdo);
```

```

String Nhietdo = String.valueOf(a.intValue());
txtnhietdo.setText(Nhietdo + "°C");
txtdrop.setText(doam + "%");
JSONObject jsonObjectWind = jsonObject.getJSONObject("wind");
String gio = jsonObjectWind.getString("speed");
txtwind.setText(gio + "m/s");
JSONObject jsonObjectClouds = jsonObject.getJSONObject("clouds");
String may = jsonObjectClouds.getString("all");
txtcloud.setText(may + "%");
JSONObject jsonObjectSys = jsonObject.getJSONObject("sys");
String country = jsonObjectSys.getString("country");
txtqg.setText("Tên quốc gia : " + country);
} catch (JSONException e) {
    e.printStackTrace();
}
},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
    }
});
requestQueue.add(stringRequest);
}

```

**Bước 5:** Ở đây dùng truyền biến city để lấy giá trị khi người dùng nhập tên thành phố cần tìm của onQueryTextChanged.

Gắn giá trị GetCurrentWeatherData(city)

```

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.activity_main_menu, menu);
    SearchView searchView =(SearchView)
    menu.findItem(R.id.menu_search).getActionView();
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            return false;
        }
    }
}

```

```

@Override
public boolean onQueryTextChange(String newText) {
    city = newText.toString();
    if (city.equals("")) {
        City ="HaiPhong";
        city=City;
        GetCurrentWeatherData(City);
        Get4DaysData(City);
    } else {
        GetCurrentWeatherData(city);
        Get4DaysData(city);
    }
    return false;
}
});
return super.onCreateOptionsMenu(menu);
}

```

**Bước 6:** Khai báo các thuộc tính có trong màn hình và gán thuộc tính cho các thuộc tính.

**MainActivity2.class**

|   |   |
|---|---|
| <pre> public class Main2Activity extends AppCompatActivity {     String tenthanhpho = "";     ImageView imgback;     TextView txtviewtp; </pre> | <pre> private void Anhxa() {     imgback=findViewById(R.id.imgback);     txtviewtp=findViewById(R.id.txtviewtp);     lvview=findViewById(R.id.lvview); } </pre> |
|---|---|

- Bên MainAcvittiy đã gửi cho MainActivity2 một intent và truyền dữ liệu vào intent bằng cách **putExtra** thì bên MainActivity2 để nhận được dữ liệu thì phải getIntent() và tiến hành **getStringExtra** để lấy dữ liệu.

### MainActivity2.class

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    Anhxa();
    Intent intent = getIntent();
    String city = intent.getStringExtra("name");
    Log.d("ketqua", "Dữ liệu truyền qua : "+city);
    if (city.equals("")){
        tenthanhpho = "HaiPhong";
        Get7DaysData(tenthanhpho);
    }else {
        tenthanhpho = city;
        Get7DaysData(tenthanhpho); }
    imgback.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            onBackPressed(); }
    });}
```

**Bước 7:** Tiếp theo sẽ tạo một class Thoitiet để tạo ra một đối tượng lưu các dữ liệu sau khi bóc tách từng thành phần.

### Thoitiet.class

|  |   |
|--|---|
| <pre>package com.example.hackerd.appweather; public class Thoitiet {     public String Day;     public String Status;     public String Image;     public String MaxTemp;     public String MinTemp;     public Thoitiet(String day, String status, String image, String</pre> | <pre>public String getStatus() {     return Status; } public void setStatus(String status) {     Status = status; } public String getImage() {     return Image; } public void setImage(String image) {     Image = image; } public String getMaxTemp() {     return MaxTemp; }</pre> |
|--|---|

|   |   |
|---|---|
| <pre> maxTemp, String minTemp) {     Day = day;     Status = status;     Image = image;     MaxTemp = maxTemp;     MinTemp = minTemp; } public String getDay() {     return Day; } public void setDay(String day) {     Day = day; } </pre> | <pre> public void setMaxTemp(String maxTemp) {     MaxTemp = maxTemp; } public String getMinTemp() {     return MinTemp; } public void setMinTemp(String minTemp) {     MinTemp = minTemp; } } </pre> |
|---|---|

- Các thuộc tính của class **Thoitiet** gồm:
  - **Day**: Ngày thông báo
  - **Status**: Trạng thái
  - **Image**: Ảnh trạng thái
  - **MaxTemp**: Nhiệt độ cao nhất
  - **MinTemp**: Nhiệt độ thấp nhất
- Bình thường nếu sử dụng luôn **ArrayAdapter** có sẵn, thì listview chỉ hiển thị dữ liệu dạng 1 đoạn text ở trên 1 row, còn bây giờ trên một row ta có cả text và image view thì không dùng nó được mà chúng ta phải custom lại để dữ liệu có thể hiển thị được, cho nên ta sẽ tạo class **CustomAdapter**.

### **CustomAdapter.class**

|   |
|---|
| <pre> public class CustomAdapter extends BaseAdapter{     Context context;     ArrayList&lt;Thoitiet&gt; arrayList;     public CustomAdapter(Context context, ArrayList&lt;Thoitiet&gt; arrayList) {         this.context = context;         this.arrayList = arrayList; } } </pre> |
|---|

```
@Override
public int getCount() {
    return arrayList.size(); }

@Override
public Object getItem(int position) {
    return arrayList.get(position); }

@Override
public long getItemId(int position) {
    return 0; }

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    convertView = inflater.inflate(R.layout.dong_listview,null);
    Thoitiet thoitiet = arrayList.get(position);
    TextView txtviewngay = convertView.findViewById(R.id.txtviewngay);
    TextView txtviewtrangthai = convertView.findViewById(R.id.txtviewtrangthai);
    TextView txtmax = convertView.findViewById(R.id.txtmax);
    TextView txtmin = convertView.findViewById(R.id.txtmin);
    ImageView imgvtrangthai = convertView.findViewById(R.id.imgvtrangthai);
    txtviewngay.setText(thoitiet.Day);
    txtviewtrangthai.setText(thoitiet.Status);
    txtmax.setText(thoitiet.MaxTemp+"°C");
    txtmin.setText(thoitiet.MinTemp+"°C");
    Picasso.with(context).load("http://openweathermap.org/img/w/"+thoitiet.Image+".png")
.into(imgvtrangthai);
    return convertView;
} }
```

**Bước 8:** Tiếp theo bóc tách dữ liệu JSON trả về 7 ngày tiếp theo.

### MainActivity2

```
private void Get7DaysData(String data) {
    String url =
"http://api.openweathermap.org/data/2.5/forecast/daily?q="+data+"&lang=vi&units=metric&cnt=7
&appid=__d0e912badecb079b4f4658f6625cd097";
```

```

RequestQueue requestQueue = Volley.newRequestQueue(Main2Activity.this);
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonObject = new JSONObject(response);
                JSONObject jsonObjectCity = jsonObject.getJSONObject("city");
                String name = jsonObjectCity.getString("name");
                txtviewtp.setText("Vị Trí : "+name);
                JSONArray jsonArrayList = jsonObject.getJSONArray("list");
                for (int i = 0; i < jsonArrayList.length(); i++){
                    JSONObject jsonObjectList = jsonArrayList.getJSONObject(i);
                    String ngay = jsonObjectList.getString("dt");
                    long l = Long.valueOf(ngay);
                    Date date = new Date(l*1000L);
                    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("EEEE ");
                    String Day = simpleDateFormat.format(date);
                    JSONObject jsonObjectTemp = jsonObjectList.getJSONObject("temp");
                    String max = jsonObjectTemp.getString("max");
                    String min = jsonObjectTemp.getString("min");
                    Double a = Double.valueOf(max);
                    Double b = Double.valueOf(min);
                    String Nhiетdomax = String.valueOf(a.intValue());
                    String Nhiетdomin = String.valueOf(b.intValue());
                    JSONArray jsonArrayWeather = jsonObjectList.getJSONArray("weather");
                    JSONObject jsonObjectWeather = jsonArrayWeather.getJSONObject(0);
                    String status = jsonObjectWeather.getString("description");
                    String icon = jsonObjectWeather.getString("icon");
                    mangthoietiet.add(new Thoitiet(Day,status,icon,Nhiетdomax,Nhiетdomin));
                }
                customAdapter.notifyDataSetChanged();
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    },
    new Response.ErrorListener() {
        @Override
    
```

```

        public void onResponse(VolleyError error) {
            }
        });
        requestQueue.add(stringRequest);
    }

```

**Bước 9:** Tiếp theo bóc tách dữ liệu JSON trả về 4 ngày tiếp theo.

### MainActivity.class

```

private void Get4DaysData(String data) {
    String url =
    "http://api.openweathermap.org/data/2.5/forecast/daily?q="+data+"&lang=vi&units=metric&cnt=4&appid=
    d0e912badecb079b4f4658f6625cd097";
    RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    JSONObject jsonObjectCity = jsonObject.getJSONObject("city");
                    JSONArray jsonArrayList = jsonObject.getJSONArray("list");
                    for (int i = 0; i < jsonArrayList.length(); i++){
                        JSONObject jsonObjectList = jsonArrayList.getJSONObject(i);
                        String ngay = jsonObjectList.getString("dt");
                        long l = Long.valueOf(ngay);
                        Date date = new Date(l*1000L);
                        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("EEEE ");
                        String Day = simpleDateFormat.format(date);
                        JSONObject jsonObjectTemp = jsonObjectList.getJSONObject("temp");
                        String max = jsonObjectTemp.getString("max");
                        String min = jsonObjectTemp.getString("min");
                        Double a = Double.valueOf(max);
                        Double b = Double.valueOf(min);
                        String Nhiетdomax = String.valueOf(a.intValue());
                        String Nhiетdomin = String.valueOf(b.intValue());
                        JSONArray jsonArrayWeather = jsonObjectList.getJSONArray("weather");
                        JSONObject jsonObjectWeather = jsonArrayWeather.getJSONObject(0);
                        String status = jsonObjectWeather.getString("description");
                        String icon = jsonObjectWeather.getString("icon");

```



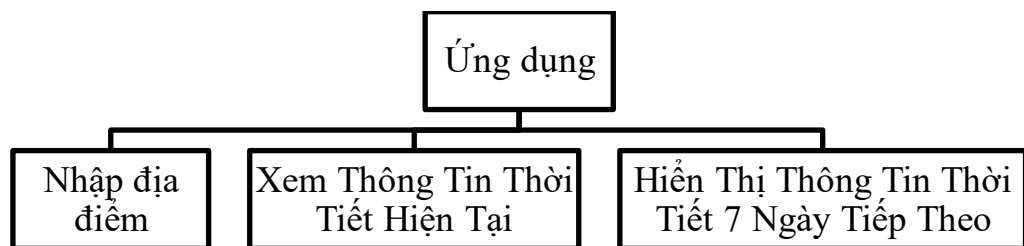
```
        arrThoiTiet.add(new Thoitiet(Day,status,icon,Nhietdomax,Nhietdomin));
    }
    Collections.reverse(arrThoiTiet);
    recyclerViewAdapter = new RecyclerViewAdapter(MainActivity.this, arrThoiTiet);
    recyclerView.setAdapter(recyclerViewAdapter);
} catch (JSONException e) {
    e.printStackTrace();
}    }    },
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
    }    });
requestQueue.add(stringRequest);
}
```

## CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM

### 4.1. Phát Biểu Bài Toán

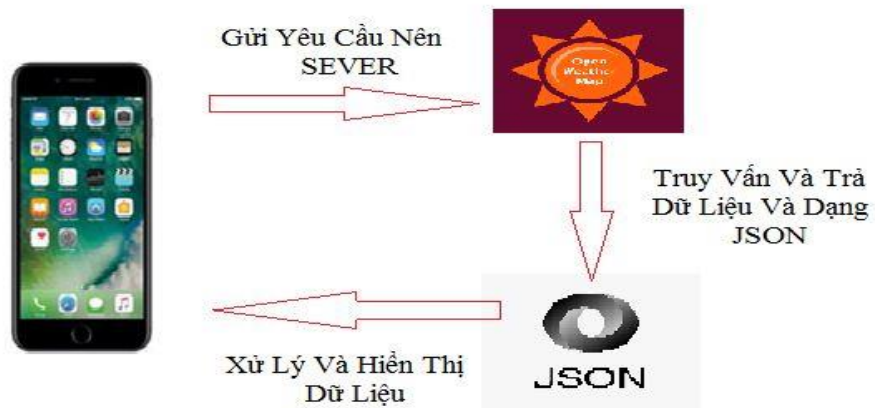
Ngày nay nhu cầu đi lại, tổ chức sự kiện của con người ngày càng tăng cao. Vì vậy, nếu nắm bắt được tình hình thời tiết như lượng mưa, độ ẩm, nhiệt độ thì sẽ dễ dàng sắp xếp cũng như tổ chức sự kiện, cuộc vui chơi một cách tốt nhất. Hiểu được nhu cầu cần thiết đó, em đã quyết định xây dựng ứng dụng “Dự báo thời tiết” để giúp người dùng có thể cập nhật tình hình thời tiết một cách nhanh nhất, mới nhất và chính xác nhất để người dùng có thể dễ dàng có những quyết định tốt nhất cho những dự định sắp đến của mình. Tốc độ phát triển nhanh chóng của các công nghệ di động hiện nay đã tạo ra các cơ hội cho những lập trình viên tự do thể hiện những ý tưởng sáng tạo của mình. Làm cho các ứng dụng ngày càng thân thiện để đáp ứng nhu cầu cuộc sống của mỗi cá nhân.

### 4.2. Mô Hình Chức Năng



- Nhập địa điểm : Ứng dụng cho phép người dùng nhập địa điểm cần tìm
- Xem thông tin thời tiết hiện tại : Ứng dụng hiển thị thông tin tình hình thời tiết như lượng mưa, độ ẩm, nhiệt độ .....
- Hiển thị thông tin thời tiết 7 ngày tiếp theo : Ứng dụng hiển thị thông tin tình hình nhiệt độ cao nhất và thấp nhất .

### 4.3. Mô Hình Hoạt Động



### 4.4. Thiết Kế Giao Diện Giao Chương Trình

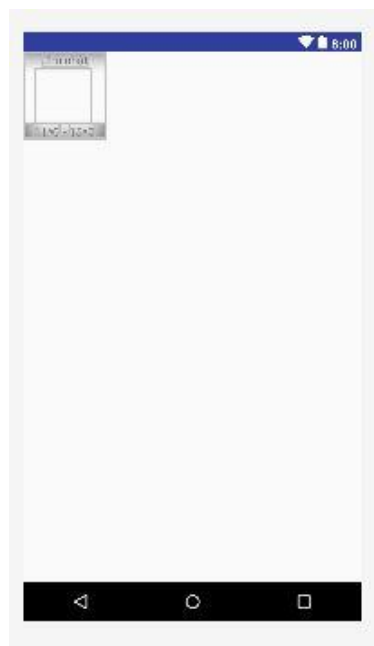
- Thiết kế giao diện chương trình tương tự như mục 3.3 tại chương 3
- Sau thiết kế xong sẽ được màn hình đề mô.



Hình 4.4.1. Thiết kế màn hình chính



Hình 4.4.2 Thiết kế thanh menu



Hình 4.4.3. Thiết kế dòng RecyclerView 4 ngày tiếp theo



Hình 4.4.4. Thiết kế giao diện 7 ngày



Hình 4.4.5. Thiết kế dòng listview 7 ngày tiếp theo

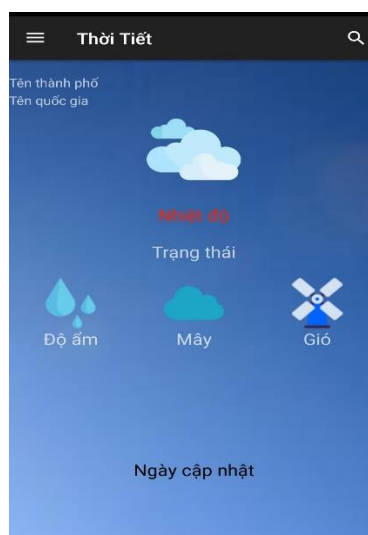
## 4.5. Giao Diện Ứng Dụng

Giao diện màn hình chính hiển thị thông tin thời tiết



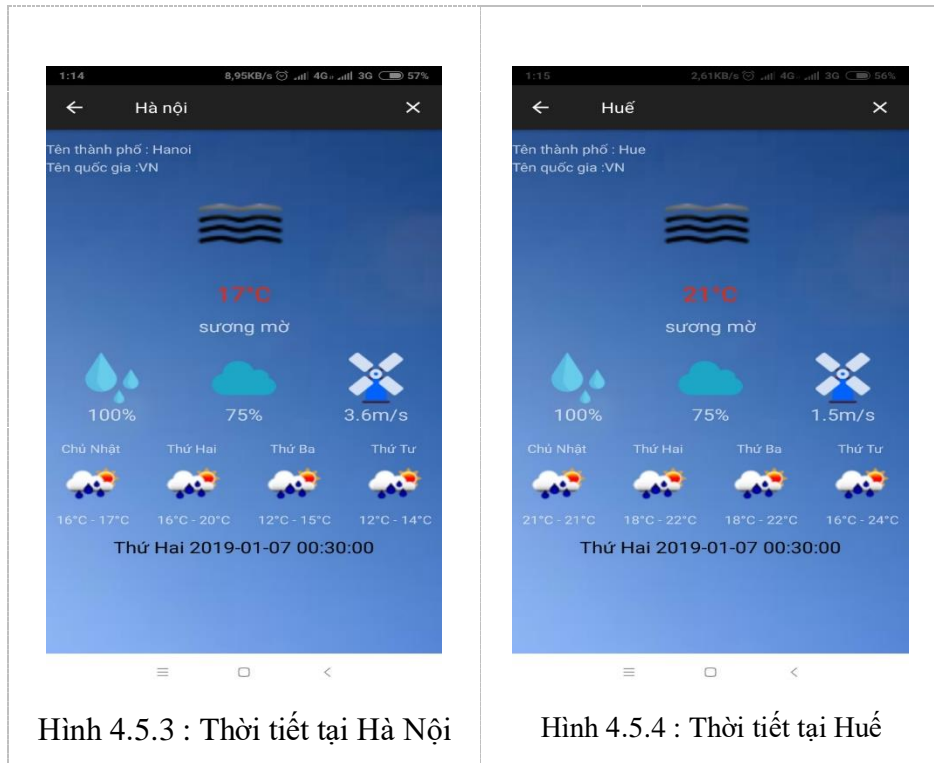
Hình 4.5.1: Giao diện màn hình chính


- Là chương trình lấy dữ liệu trên internet lên phần kiểm tra kết nối internet của thiết bị vô cùng quan trọng.

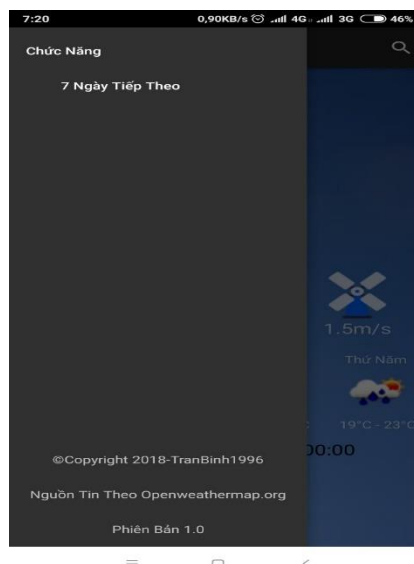


Hình 4.5.2: Màn hình hiển thị khi thiết bị không kết nối internet

- Người dùng có thể tìm kiếm tại các địa điểm khác nhau

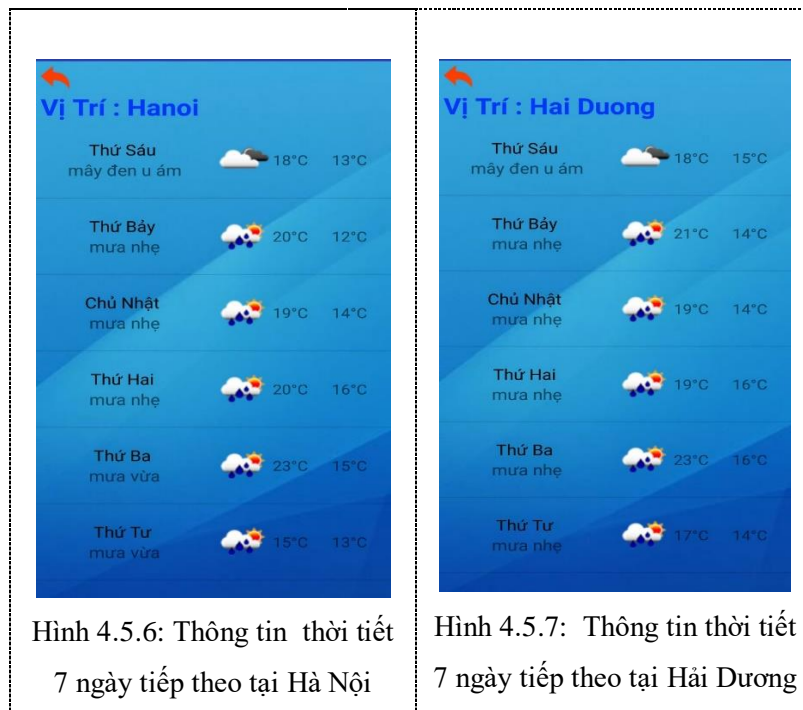


- Khi người dùng chọn vào biểu tượng  trên màn hình thì sẽ xuất hiện menu hiển thị chức năng xem thông tin thời tiết 7 ngày tiếp theo.



Hình 4.5.5: Khi người dùng vuốt sang bên cạnh

- Người dùng có thể truy cập xem thông tin thời tiết 7 ở địa điểm khác nhau.





## KẾT LUẬN

Sau một thời gian tìm hiểu đề tài “Xây dựng ứng dụng Android lấy thông tin dự báo thời tiết ” em đã thực hiện được đề tài của mình theo như dự kiến. Với mục đích là tìm hiểu về hệ điều hành Android, hiểu biết về ngôn ngữ lập trình mới cho điện thoại di động và tìm hiểu các ứng dụng của hệ điều hành Android trong thế giới di động ngày nay. Thông qua việc tìm hiểu và nắm bắt lý thuyết về hệ điều hành Android để bước đầu tìm hiểu ứng dụng dự báo thời tiết trên nền hệ điều hành Android cho điện thoại di động.

Trong quá trình tìm hiểu lý thuyết về hệ điều hành Android và tìm hiểu ứng dụng dự báo thời tiết cho thiết bị di động em đã thu được kết quả bước đầu như sau

### **Về mặt lý thuyết:**

- Sau quá trình tìm hiểu và nghiên cứu đề tài, em đã bước đầu hiểu về hệ điều hành Android chạy trên điện thoại. Biết và nắm vững quá trình tạo một file project trong lập trình trên nền Android.

- Hiểu và sử dụng được các lớp đối tượng, phương thức hỗ trợ lập trình trên nền hệ điều hành Android.

- Hiểu được cơ chế làm việc với JSON, weather map api để trích xuất dữ liệu từ JSON và đưa vào java class.

Do thời gian ngắn cùng với kiến thức của bản thân còn hạn chế nên em chưa tìm hiểu sâu về các kiến thức liên quan tới hệ điều hành Android nên chương trình vẫn còn rất nhiều hạn chế, em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và nhiều chức năng hơn.

### **Về mặt ứng dụng**

- Xem được thời tiết địa điểm hiện tại của thiết bị.
- Xem được thời tiết tại địa điểm nhập vào bất kỳ.
- Xem được thời tiết các ngày kế tiếp.

**Tuy nhiên vẫn còn những hạn chế:**

- Chưa hiển thị thời tiết theo thời gian thực.
- Ứng dụng còn ít dữ liệu, chưa đáp ứng đủ nhu cầu thông tin từ người sử dụng

**Hướng phát triển.**

Hoàn thiện và bổ sung thêm các tính năng mới:

- Cập nhật thời tiết theo thời gian thực.

Nghiên cứu và tìm hiểu sâu các kiến thức liên quan tới hệ điều hành Android để có thể xây dựng thêm một số ứng dụng khác trên hệ điều hành Android cho điện thoại di động như là: chat, gửi tin nhắn có hình MMS, gửi video, gửi file, nghe nhạc, chơi game, mail.

Trong quá trình thực hiện đề tài em đã cố gắng tìm hiểu tài liệu qua nhiều phương thức để tổng hợp kiến thức, học hỏi từ thầy cô và bạn bè để hoàn thành. Em xin chân thành cảm ơn các thầy cô cùng các bạn đã giúp đỡ em hoàn thành. Trong bài báo cáo còn có thiếu sót mong thầy cô và các bạn góp ý để em hoàn thiện hơn.

## TÀI LIỆU THAM KHẢO

- [1]. [https://vi.wikipedia.org/wiki/Android\\_\(hệ\\_điều\\_hành\)](https://vi.wikipedia.org/wiki/Android_(hệ_điều_hành))
- [2]. <http://genk.vn/mobile/lich-su-va-chang-duong-phat-trien-cua-android-qua-cac-phien-ban-phan-1-thoi-ky-dau-cua-android-20150608204249527.chn>
- [3]. <https://viblo.asia/p/kien-truc-cua-he-dieu-hanh-android-PaLGDYdaelX>
- [4]. <https://viblo.asia/p/android-buoc-tien-dai-cua-cach-mang-cong-nghe-DZrGNDOJkVB>
- [5]. [https://vi.wikipedia.org/wiki/Android\\_Studio](https://vi.wikipedia.org/wiki/Android_Studio)
- [6]. <http://giasutinhoc.vn/lap-trinh-android-co-ban/xu-ly-su-kien-trong-android-bai-3/>
- [7]. <https://laptrinhtuduy.wordpress.com/2014/04/27/activity-trong-android/>