

**TRƯỜNG CAO ĐẲNG NGHỀ ĐÀ NẴNG**

**KHOA CÔNG NGHIỆ THÔNG TIN**

**- - - 🙞 🕮** **🙜 - - -**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG ANDROID DỰ BÁO THỜI TIẾT**

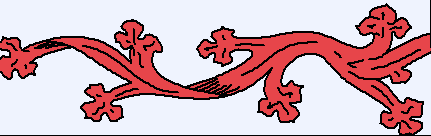
Giảng viên hướng dẫn : Trần Thị Hà Khuê

Sinh viên thực hiện : Trần Văn Công – Mai Quang Minh

Ngô Hoàng Anh

Lớp : 20CNTT1A

Mã Sinh Viên : 20C0751-20C0623



***Đà Nẵng, 01/ 2023***

**LỜI CẢM ƠN**

Qua thời gian học tập và nghiên cứu tại trường Cao Đẳng Nghề Đà Nẵng, em xin chân thành cảm ơn tới tất cả thầy cô giáo trong Khoa Công nghệ thông tin - trường Cao Đẳng Nghề Đà Nẵng, 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 cô Trần Thị Hà Khuê – Khoa Công nghệ thông tin - Trường Cao Đẳng Nghề Đà Nẵng. 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!

Mục Lục

[**CHƯƠNG 1: MỞ ĐẦU** 3](#_Toc123538226)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 5](#_Toc123538227)

[1.1. Giới Thiệu Hệ Điều Hành Android 5](#_Toc123538228)

[1.2. Sự Phát Triển Của Hệ Điều Hành Android 6](#_Toc123538229)

[1.3. Kiến Trúc Của Hệ Điều Hành Android 8](#_Toc123538230)

[1.4. Sơ lược về Android Studio 12](#_Toc123538231)

[1.5. Cài Đặt Android Studio 12](#_Toc123538232)

[1.6. Cấu Trúc Dự Án Android Studio 13](#_Toc123538233)

[1.9. Các Thành Phần Trong Một Ứng Dụng Android 20](#_Toc123538234)

[1.10. Tạo Giao Diện Chương Trình Trong Android Studio 22](#_Toc123538235)

[**CHƯƠNG 3: KỸ THUẬT XÂY DỰNG ỨNG DỤNG THỜI TIẾT** 27](#_Toc123538236)

[3.1. Tìm Hiều Về Dịch Vụ OpenWeatherMap 27](#_Toc123538237)

[3.2. Đăng Ký Và Kích Hoạt Sử Dụng Key API Openweathermap 28](#_Toc123538238)

[3.3. Thiết Kế Giao Diện Ứng Dụng Bằng Ngôn Ngữ XML 29](#_Toc123538239)

[CHƯƠNG 4: XÂY DỰNG APP DỰ BÁO THỜI TIẾT 38](#_Toc123538240)

[4.1. Phát Biểu Bài Toán 38](#_Toc123538241)

[4.2. Mô Hình Chức Năng 38](#_Toc123538242)

[4.3. Mô Hình Hoạt Động 38](#_Toc123538243)

[4.4. Thiết Kế Giao Diện Giao Chương Trình 39](#_Toc123538244)

[41](#_Toc123538245)

[43](#_Toc123538246)

[**CHƯƠNG 5: KẾT LUẬN** 49](#_Toc123538247)

# 

# **CHƯƠNG 1: MỞ ĐẦU**

1. **Tên đề tài:**

Xây dựng ứng dụng dự báo thời tiết trên ứng dụngAndroid

1. **Lý do chọn đề tài:**

* Đối với đời sống sinh hoạt:

+ Để dễ dàng sắp xếp lịch (đi chơi, đi thăm viếng...)

+ Để dễ dàng tận dụng (phơi đồ…)

* Đối với đời sống sản xuất:

+ Lên kế hoạch xử lí nông sản

+ Để dễ dàng theo dõi tình hình bệnh tình trên đồng ruộng (dự báo thời tiết nông vụ)

* Đối với an ninh, tình hình

+ Ứng biến kịp thời với thiên tai

+ Đảm bảo an toàn cho tàu thuyền, ngư dân đánh bắt.

Và một điều, một ý nghĩa rất quan trọng của Dự báo thời tiết đó chính là chứng tỏ sự tiến bộ của khoa học kĩ thuật

1. **Mục tiêu đề tài:**

Mong muốn độ chính xác của ứng dụng dự báo thời tiết là 90%

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

### 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

Định dạng Vban như trên: sz chữ 13, canh đều, thụt vào 1cm dong đầu tiên

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

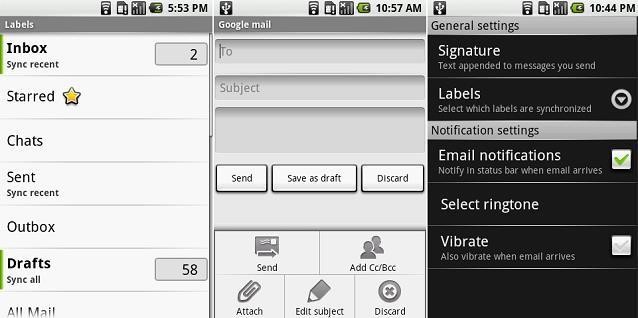
Thanh thông báo vuốt từ trên xuống (Notification bar): Ngay từ những ngày đầu tiên của Android, thanh thông báo này đã đánh dấu một bước quan trọng mà trước đây chưa hề có hệ điều hành nào làm được - đưa tất cả thông tin tin nhắn, tin thoại hoặc các cuộc gọi nhỡ chỉ với thao tác vuốt xuống như hình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

|  |  |
| --- | --- |
| Hình 1.2.3. Màn hình chính Android 1.0 | Hình 1.2.4. Màn hình chính Android 8.0 |

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 [2]. Giao diện gmail của phiên bản Android thời kỳ đầu như hình 1.2.5.



Hình 1.2.5. Gmail trên Android thời kì đầu

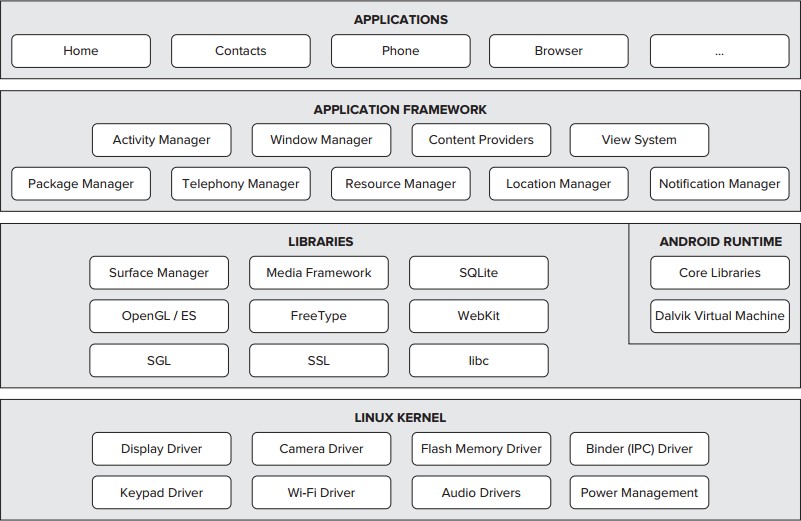
Kho ứng dụng Android: Thật khó có thể tưởng tượng một chiếc smartphone mà không hề có kho ứng dụng, nhưng vào thời điểm Android mới ra mắt, gần như không có bất kì điện thoại nào có kho ứng dụng nào được tích hợp và chính Android đã mở đầu cuộc cách mạng ứng dụng di động này. Android Market trên 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 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.

“Notifycation 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.

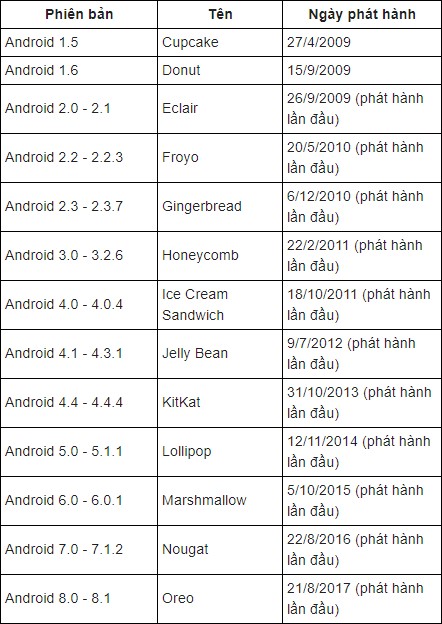
Tầng ứng dụng là tầng giao tiếp với người dùng, ví dụ: trên thiết bị Android như danh bạ, trình duyệt… mọi ứng dụng viết đều nằm trên tầng này. Giao diện hệ điều hành Android

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

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện.

Các phiên bản Android



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-piture,](https://quantrimang.com/huong-dan-su-dung-che-do-picture-in-picture-tren-android-oreo-140151) tự động điền mật khẩu trong ứng dụng, cải thiện thời lượng pin, Project Treble…

### 1.4. 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].

### 1.5. Cài Đặt Android Studio

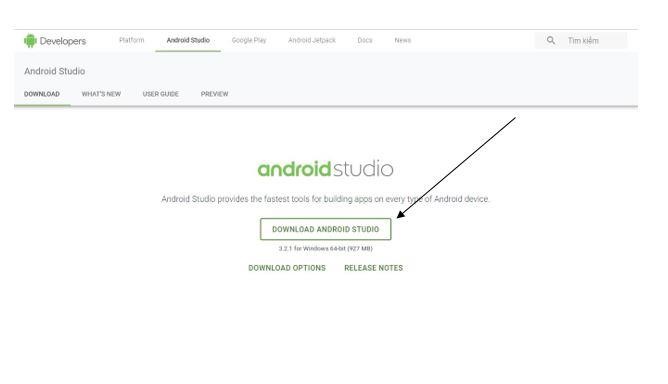
**1.5.1. Yêu Cầu Phần Cứng Máy Tính**

#### 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 thiếu 1366 x 768[6]

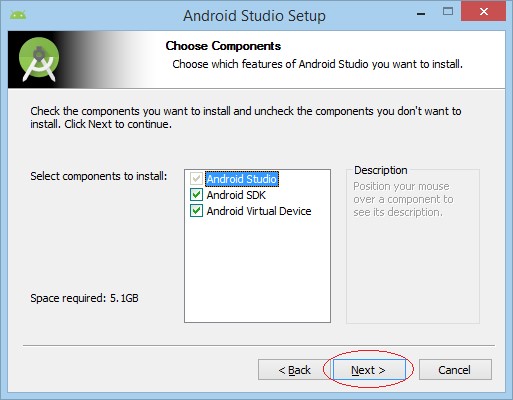
##### **1.5.2. Phần mềm Android Studio**

* Vào đường dẫn: “https://developer.android.com/studio/” để download bản mới nhất và tiến hành cài đặt như hình 1.5.1.1:



Hình 1.5.1.1. Trang download Android Studio

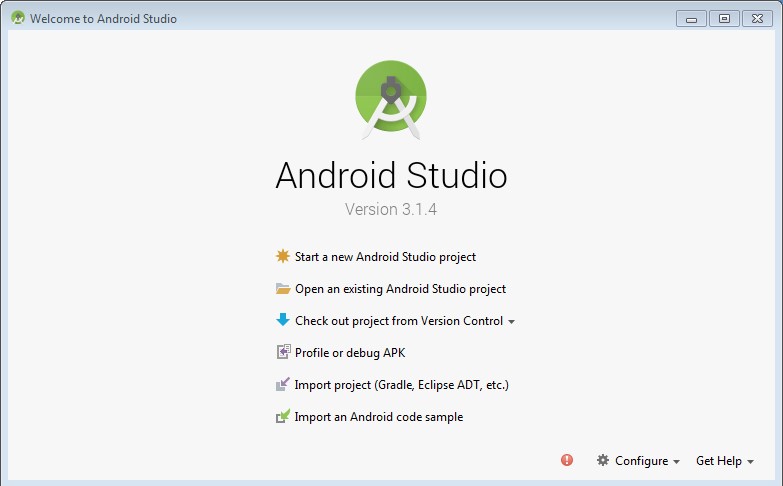
* Khi cài đặt chú ý chọn cả SDK và trình giả lập thiết bị Android Studio Device như hình 1.5.1.2.



Hình 1.5.1.2. 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 1.5.1.3:

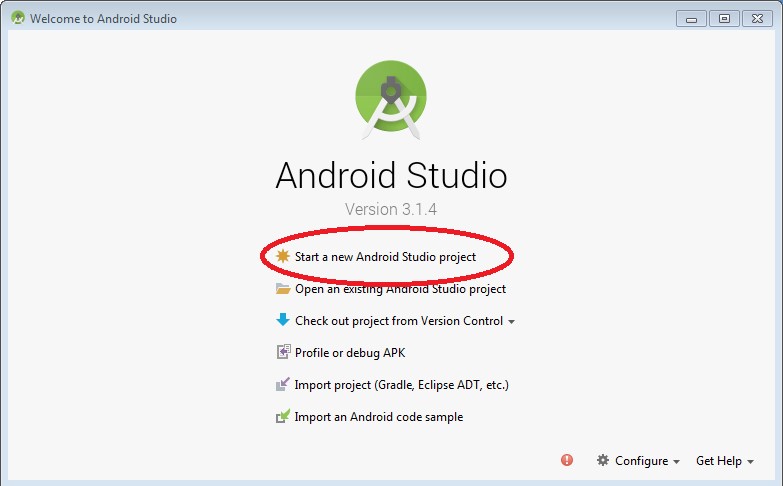


Hình 1.5.1.3. Giao diện cửa sổ Android Studio khi cài đặt hoàn tất

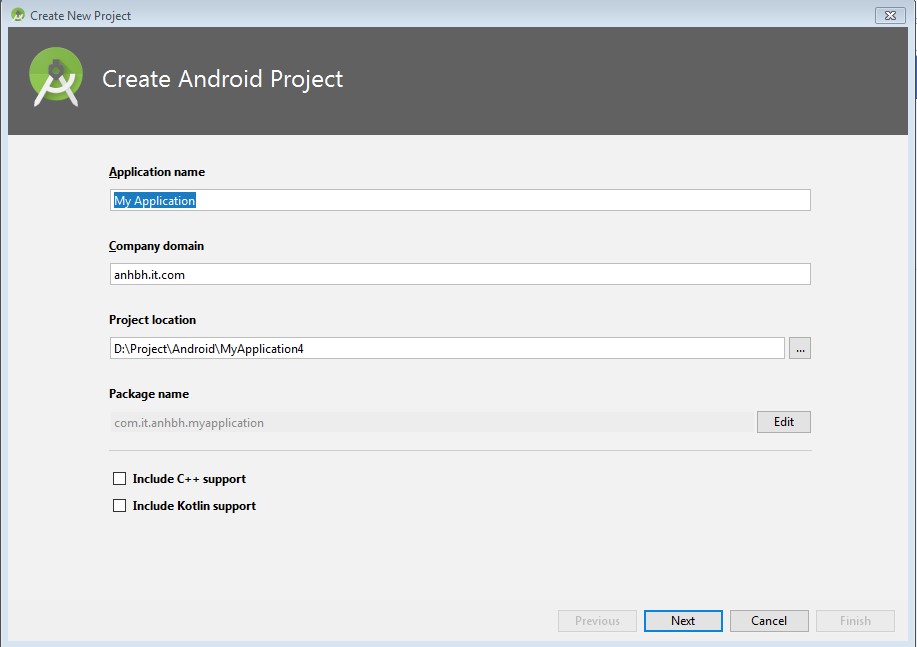
### 1.6. Cấu Trúc Dự Án Android Studio

##### 1.6.1. Tạo Mới Một Project

* Click vào dòng “Start a new Android Studio project” như hình 1.6.1.1.



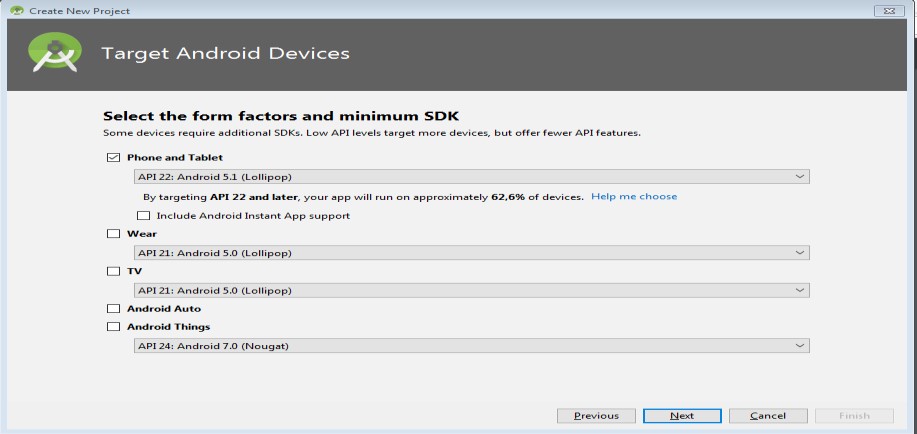
* Hình 1.6.1.1. Giao diện cửa sổ đầu tiên khi chọn Android Studio



Hình 1.6.1.2. Giao diện tạo project Android

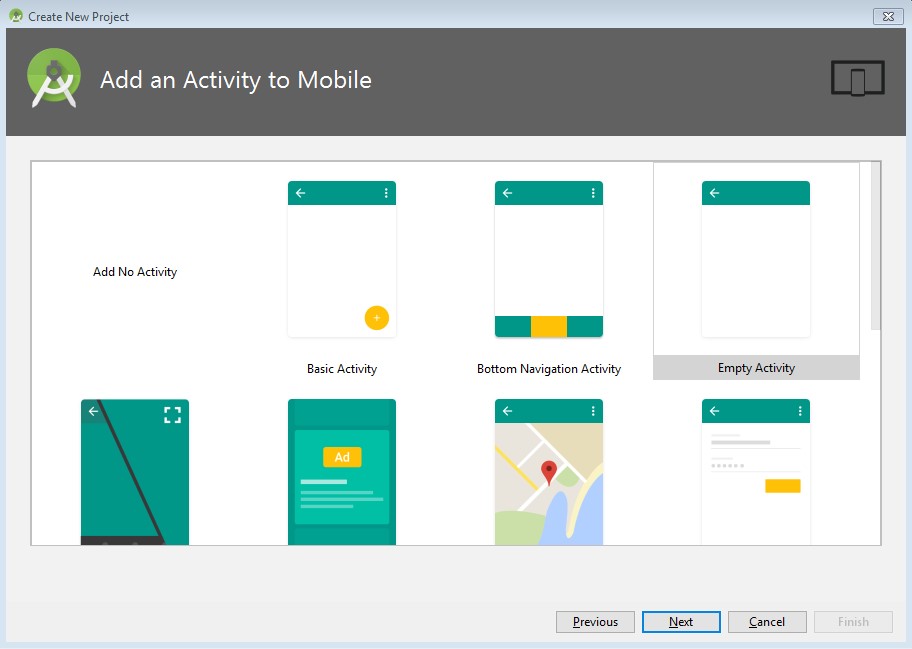
* Khi đó xuất hiện màn hình tạo project Android gồm các thành phần sau:
  + **Application Name**: Tên ứng dụng muốn đặt
  + **Company Domain**: Tên domain công ty, thường được dùng để kết hợp với tên Application để tạo thành Package (chú ý viết thường hết và có ít nhất 1 dấu chấm).
  + **Package name**: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

*1.7.2.* Cài Đặt Một Project



Hình 1.7.2.1. 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 1.7.2.2. Chọn kiểu màn hình cho ứng dụng

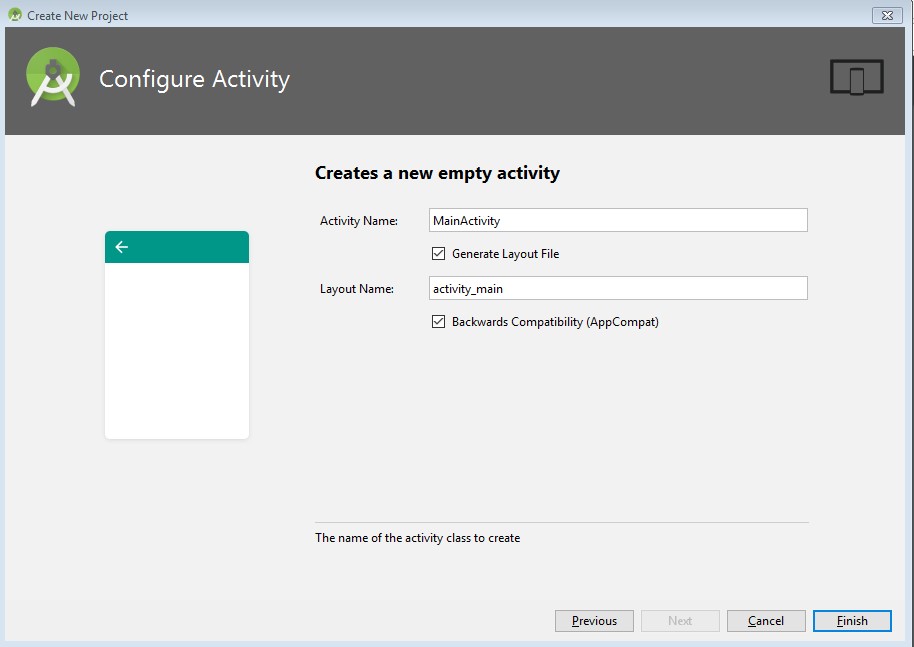
* Chọn **Empty Activity** rồi bấm Next

Hình 1.7

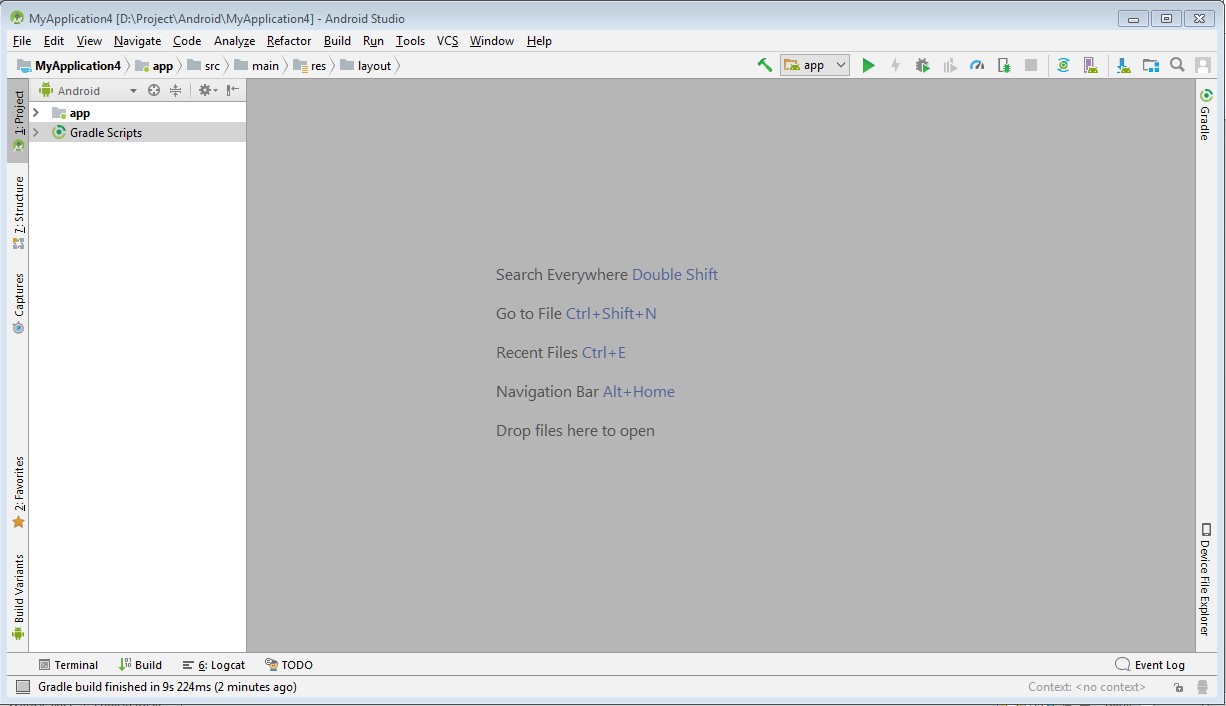
.2.3

.

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 1.7.2.4. Giao diện sau khi ra tạo thành công project

##### *1.8.3. Màn hình làm việc của dự án Android Studio*

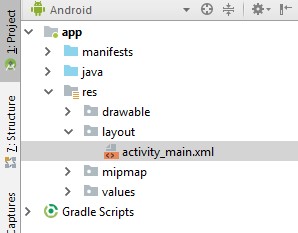
* Theo mặc định Android Studio hiển thị các files trong project theo góc nhìn Android. Góc nhìn này Android Studio sẽ tổ chức các files theo 3 module:
  + - **manifests**: chứa file AndroidManifest.xml.
    - **java**: chứa các file mã nguồn Java.
    - **res**: chứa tất cả các file layout, xml, giao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

*1.8*.3.1.



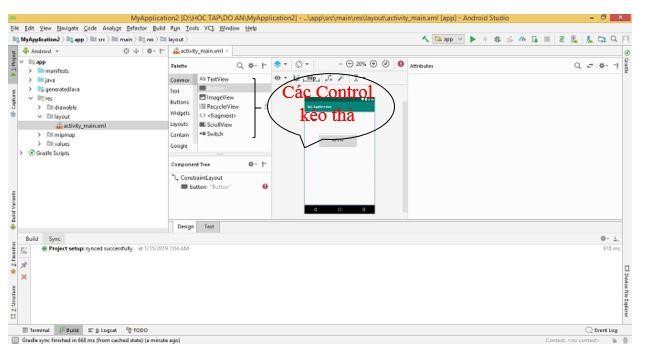
Hình *1.8*.3.1. Các vùng làm việc trong Android Studio

* Ở hình *1.8*.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).



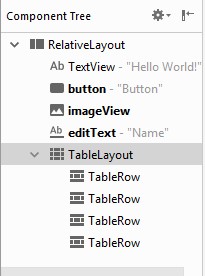
Hình *1.8*.3.2. Cấu trúc thư mục dưới dạng Android

* + **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 *1.8*.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 lập trên giao diện (vùng 4).



Hình *1.8*.3.4. Cấu trúc giao diện theo dạng cây

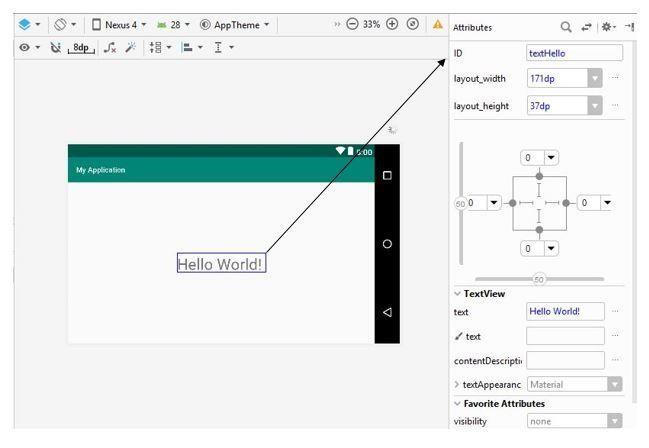
* + **Vùng 4** : Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiểu chính control.



Hình *1.8*.3.5. Vùng giao diện để các control kéo thả vào

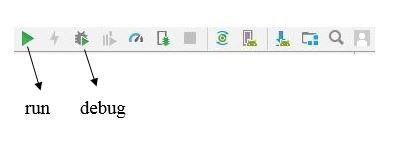
Vùng 4 ta có thể chọn cách hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉ control, lựa chọn loại thiết bị hiển thị….

* + **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 *1.8*.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.

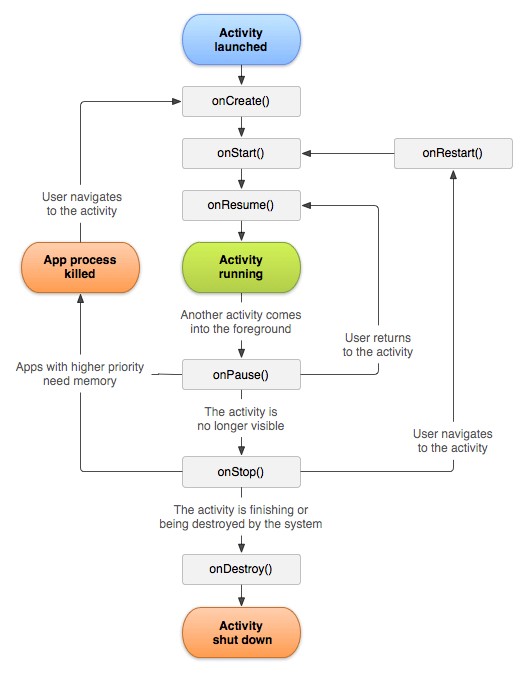


Hình *1.8*.3.7. Các nút quan trọng trên Android Studio

### 1.9. 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 1.9.1. 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 thế 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)

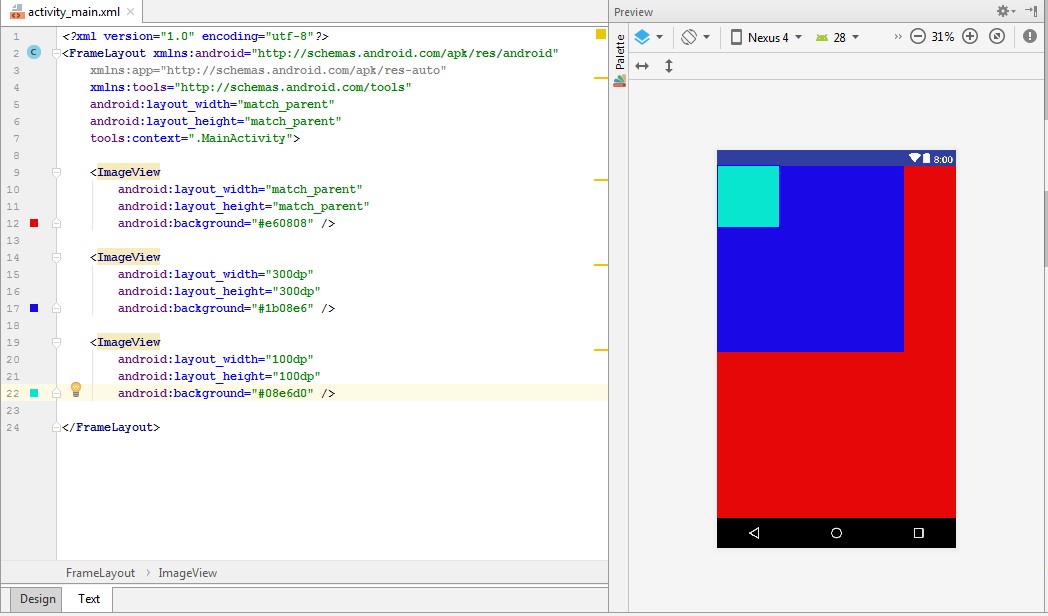
 **Lưu ý**: Activity, Service, Broadcast Receiver và Content Provider là những thành phần chính tạo nên ứng dụng Android do đó chúng phải được khai báo trong AndroidManifest.xml trước khi sử dụng

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

##### 1.10.1. 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 1.10.1.1.



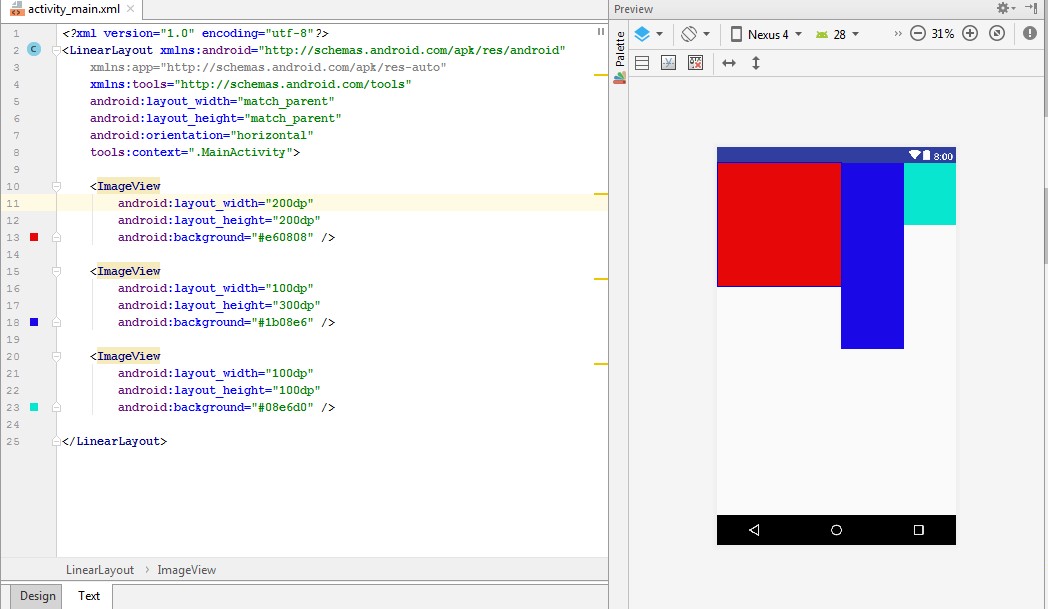
Hình 1.10.1.1. Giao diện sử dụng FrameLayout

* **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.

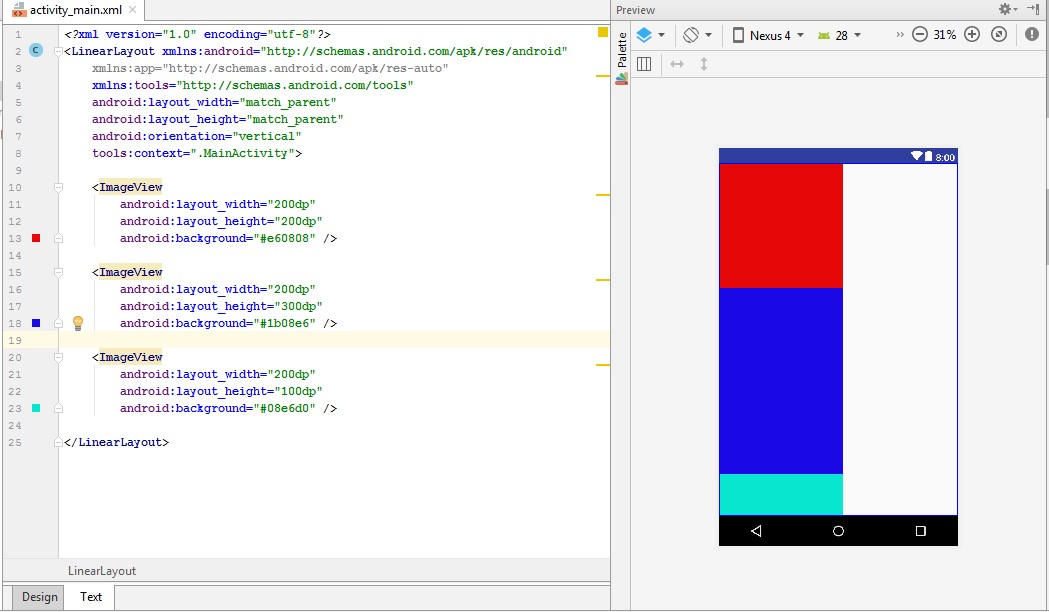
Sau đây là đoạn XML demo cách sử dụng layout:

* Theo hàng



Hình 1.10.1.2. Giao diện LinearLayout theo chiều ngang

* Theo cột

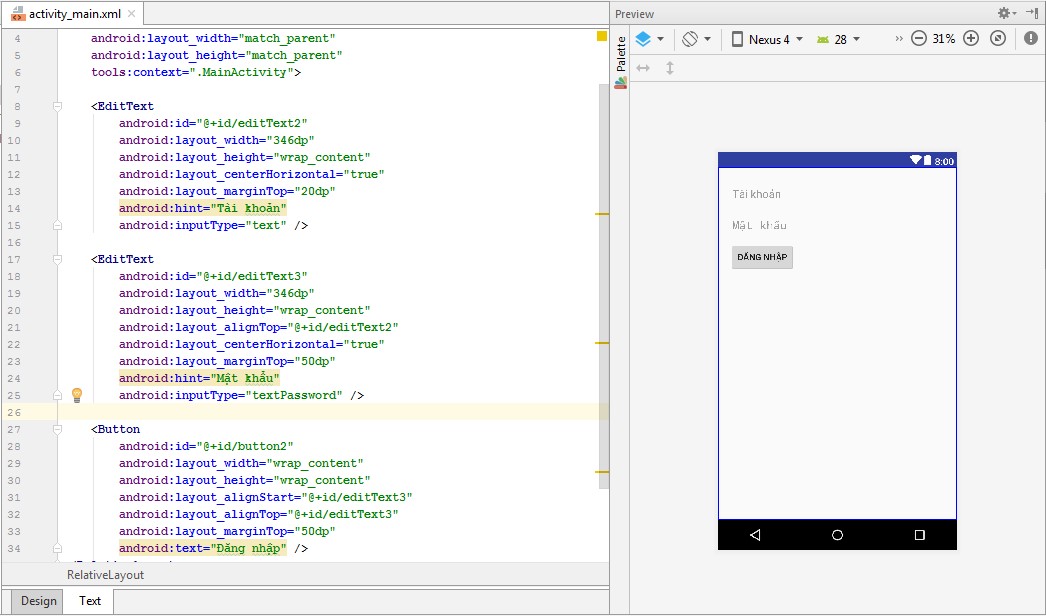


Hình 1.10.1.3. Giao diện LinearLayout theo chiều dọc

 **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 1.10.1.4. Giao diện RelativeLayout

##### 1.10.2. 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.

##### 1.10*.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()** | **OnClickListener()**  Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focuse trên bất ký widget như button, text, image vv. Chúng ta sẽ sử dụng phương onClick() để xử lý sự kiện. |
| onLongClick() | **OnLongClickListener()**  Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất ký widget như button, text, image vv. trong một hoặc nhiều giây. Chúng ta sẽ sử dụng |

|  |  |
| --- | --- |
|  | phương onLongClick() để xử lý sự kiện. |
| onFocusChange() | **OnFocusChangeListener()**  Sự kiện phát sinh khi widget mất focus. |
| onKey() | **OnFocusChangeListener()**  Sự kiện phát sinh khi người dùng focus trên widget và nhấn (presse) hoặc thả (release) một phím trên thiết bị. |
| onTouch() | **OnTouchListener()**  Sự kiện phát sinh khi người dùng nhấn phím, nhả phím, hoặc bất kỳ cử chỉ chuyển động trên màn hình. |
| onMenuItemClick() | **OnMenuItemClickListener()**  Sự kiện phát sinh khi người dùng chọn một mục trong menu. |
| onCreateContextMenu() | **OnCreateContextMenuItemListener()**  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 Hiều Về Dịch Vụ OpenWeatherMap

###### 3.1.1. Khái Niệm

[OpenWeatherMap](http://www.openweathermap.com/) 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 đ01B0ợc 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*

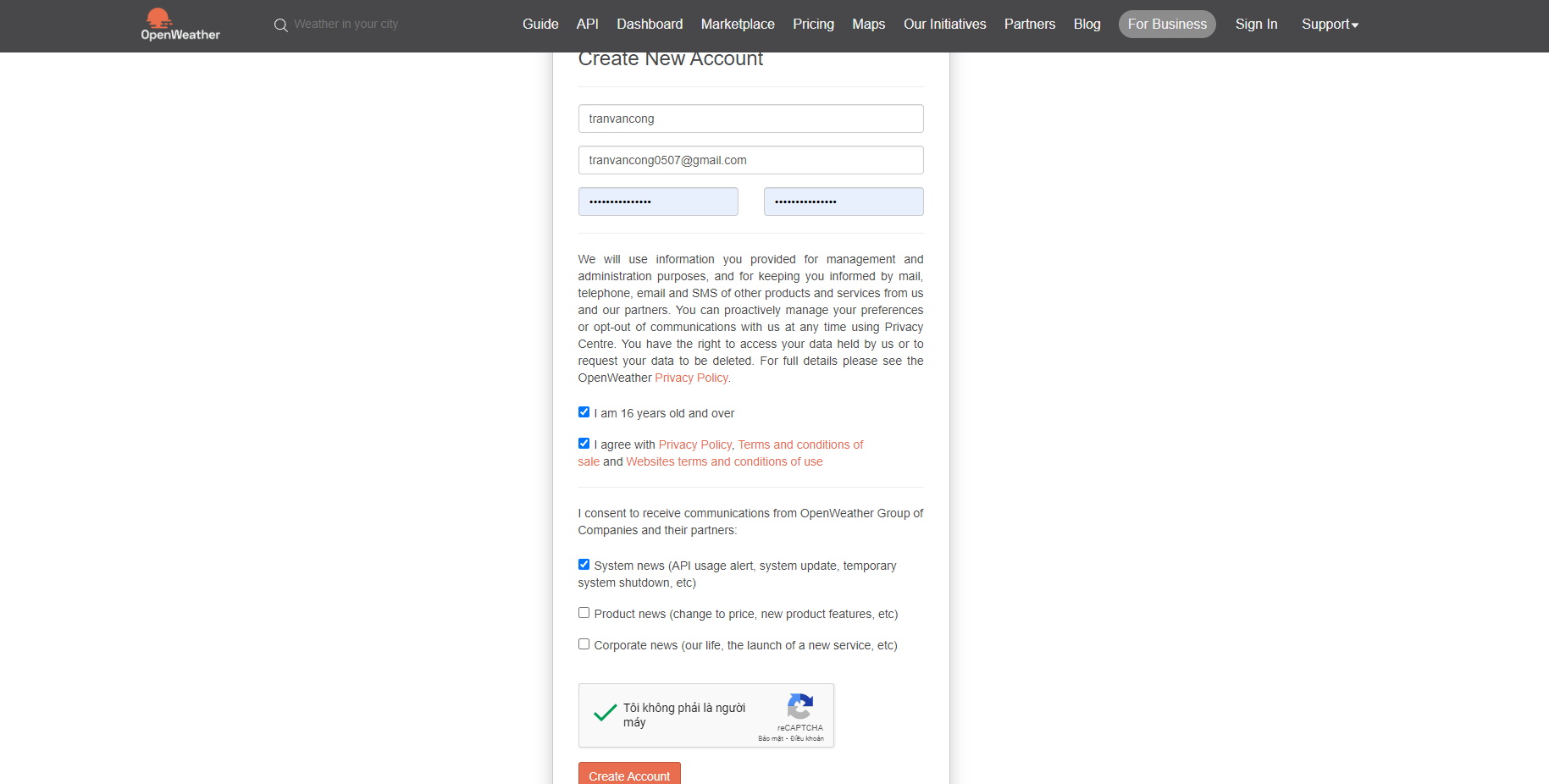
API OpenWeatherMap cung cấp dữ liệu lịch sử thời tiết cho hơn 37.000 thành phố. Dữ liệu lịch sử trong 1 tháng trước đó có sẵn trong tài khoản Starter, cho 1 năm trước trong tài khoản Medium và trong 5 năm trong History Bulk

### 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*

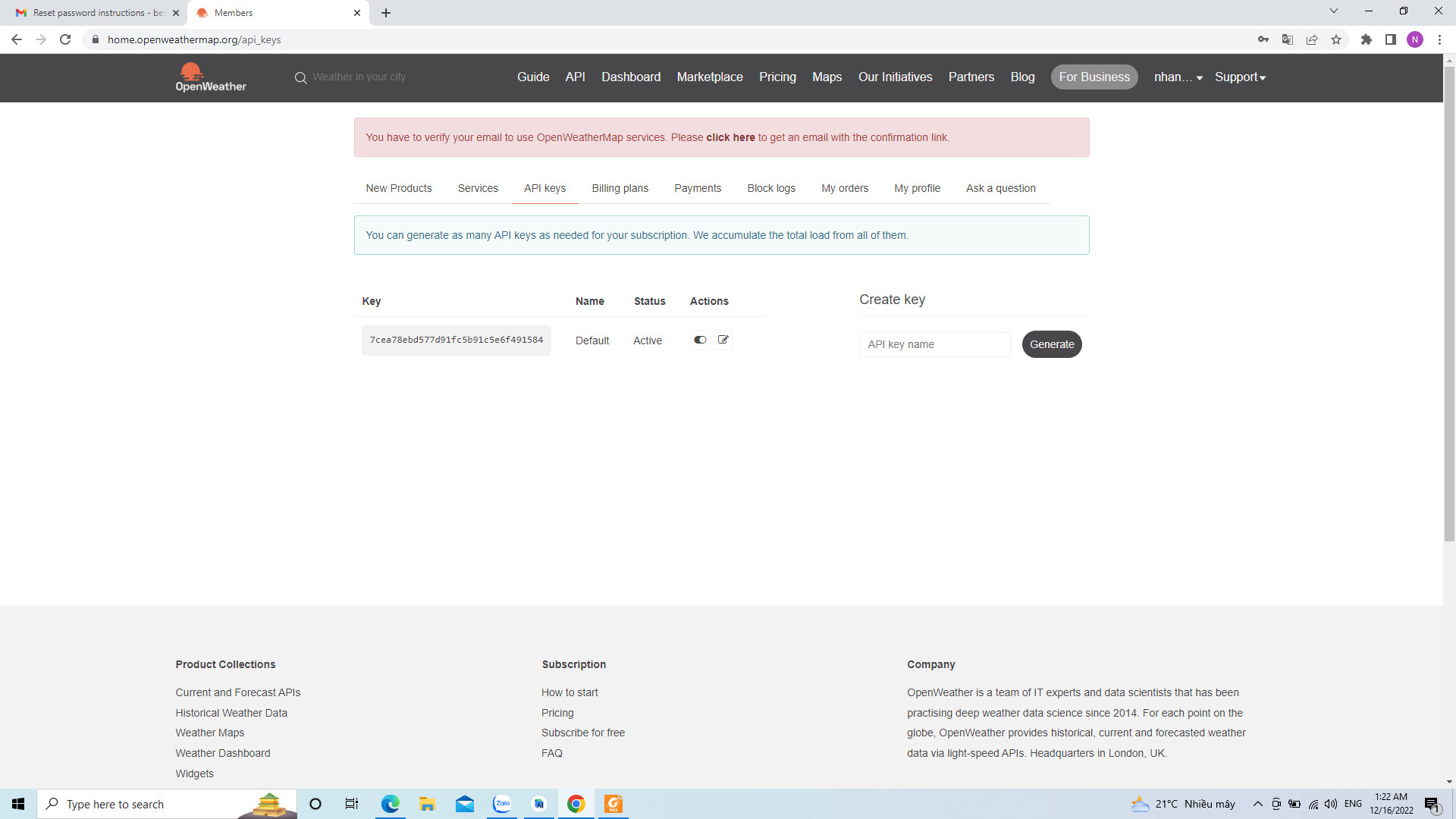
Ta truy cập đường link

<https://home.openweathermap.org/users/sign_up>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:

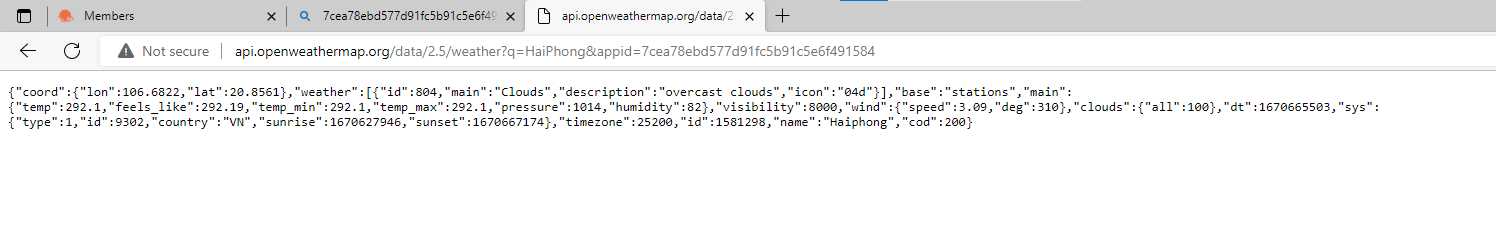


Hình 3.2.2. Trang chủ OpenWeatherMap

* 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=d 0e912badecb079b4f4658f6625cd097

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 hiện thị các thông tin như tên thành phố, quốc gia, trạng thái …..

Được thiết kế theo TextView và ImageView

activitiy\_scrolling.xml

|  |  |
| --- | --- |
| <LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <TextView  android:id="@+id/todayTemperature"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="0 °C"  android:textColor="@color/text\_light"  android:textSize="36sp"/>  <TextView  android:id="@+id/todayDescription"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="No data"  android:textColor="@color/text\_light"  android:textSize="16sp"/>  <TextView  android:id="@+id/todayWind"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  tools:text="Wind: 0 m/s"  android:textColor="@color/text\_light"  android:textSize="16sp"/>  <TextView  android:id="@+id/todayPressure"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Pressure: 0 hpa"  android:textColor="@color/text\_light"  android:textSize="16sp"/>  <TextView  android:id="@+id/todayHumidity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Humidity: 0 %"  android:textColor="@color/text\_light"  android:textSize="16sp"/> | <TextView  android:id="@+id/todaySunrise"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Sunrise: 00:00"  android:textColor="@color/text\_light"  android:textSize="16sp"/>  <TextView  android:id="@+id/todaySunset"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Sunset: 00:00"  android:textColor="@color/text\_light"  android:textSize="16sp"/>  <TextView  android:id="@+id/todayUvIndex"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Uv Index: Loading..."  android:textColor="@color/text\_light"  android:textSize="16sp"/>  </LinearLayout> |

* Tiếp đó tạo hiển thị thông tin thời tiết và thời gian cập nhật thời tiết, được hiển thị theo RecyclerView và TextView. **activitiy\_scorilling.xml**

|  |  |
| --- | --- |
| <LinearLayout  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_gravity="end"  android:layout\_weight="1"  android:orientation="vertical">  <LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:id="@+id/linearLayout\_tap\_for\_graphs"  android:layout\_gravity="end">  <TextView  android:id="@+id/todayIcon"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:gravity="center"  android:paddingRight="42dp"  android:textColor="@color/text\_light"  android:text=""  android:textSize="72sp"/>  <TextView  android:id="@+id/tapGraph"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:gravity="center"  android:paddingRight="42dp"  android:layout\_marginTop="-20dp"  android:textColor="@color/text\_light"  android:text=""  android:textSize="14sp" /> | </LinearLayout>  <TextView  android:id="@+id/lastUpdate"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_gravity="bottom|end"  android:gravity="bottom|end"  android:paddingRight="16dp"  android:textAlignment="viewEnd"  android:textSize="14sp"/>  </LinearLayout> |

* Tại giao diện hiển thị thông tin thời tiết

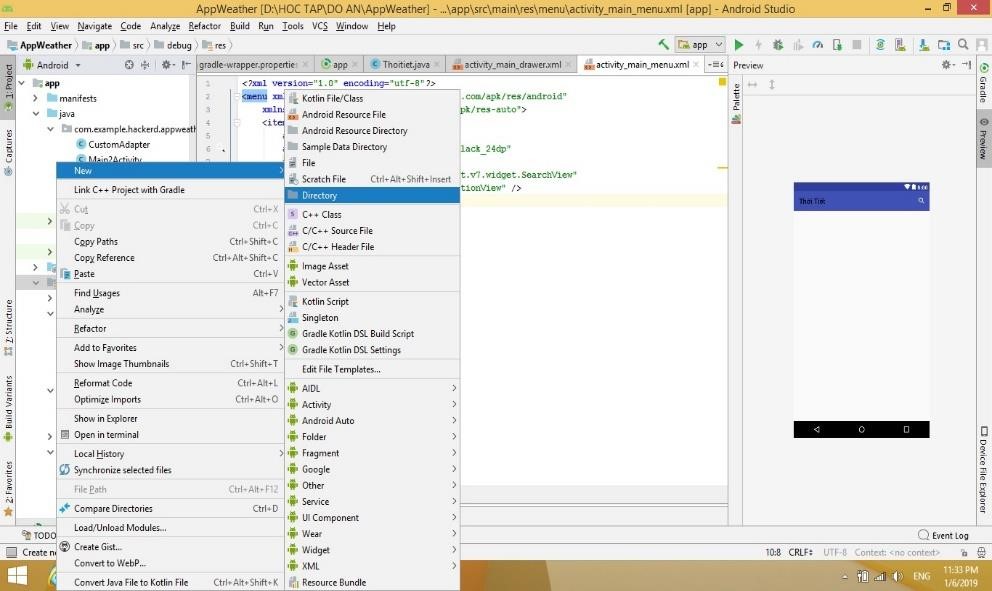
Được hiển thị theo TextView và ImageView

##### Extensive\_widget.xml

|  |
| --- |
| </LinearLayout>  <TextView  android:id="@+id/widgetDescription"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="No data"  android:textColor="@color/text\_light"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetCity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  tools:text="City name, AB"  android:textColor="@color/text\_light"  android:textSize="16sp"  android:textStyle="bold" />  <TextView  android:id="@+id/widgetWind"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  tools:text="Wind: 0 m/s"  android:textColor="@color/text\_light"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetPressure"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Pressure: 0 hpa"  android:textColor="@color/text\_light"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetHumidity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Humidity: 0 %"  android:textColor="@color/text\_light"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetSunrise"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Sunrise: 00:00"  android:textColor="@color/text\_light"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetSunset"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Sunset: 00:00"  android:textColor="@color/text\_light"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetLastUpdate"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="16sp" />  </LinearLayout> |

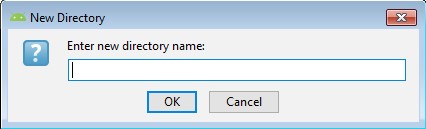
* Tạo menu menu tìm kiếm theo tên thành phố

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



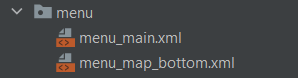
Hinh 3.3.1 : Tạo thư mục menu

Đặt tên thư mục cho menu



Hinh 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 menu\_main.



Hình 3.3.3 : Menu sau khi được tạo

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

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

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

tools:context=".activities.MainActivity">

</menu>

Giao diện tìm kiếm theo tên thành phố

<item

android:id="@+id/action\_search"

android:icon="@drawable/ic\_magnify\_white\_24dp"

android:title="@string/action\_search"

app:showAsAction="always" />

* Tiếp theo sẽ tiến hành tạo màn hình hiển thị thông tin dự báo thời tiết về 5 ngày tiếp theo

Được hiển thị theo ImageView, TextView, ListView ***time\_widget\_classic.xml***

|  |
| --- |
| <LinearLayout  android:id="@+id/textHolder"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_below="@id/date">  <TextView  android:id="@+id/widgetDescription"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Clear Sky"  android:textColor="@android:color/white"  android:textSize="16sp" />  <TextView  android:id="@+id/widgetCity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  tools:text="City name, AB"  android:textColor="@android:color/white"  android:textSize="16sp"  android:textStyle="bold" />    </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

##### list\_row.xml

|  |
| --- |
| <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  xmlns:tools="http://schemas.android.com/tools"  android:paddingBottom="16dp"  android:orientation="vertical">  <View  android:id="@+id/lineView"  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="?android:attr/listDivider" />  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="16dp"  android:layout\_marginTop="16dp"  android:orientation="vertical">  <TextView  android:id="@+id/itemDate"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="16:00"  android:textColor="?android:textColorPrimary"  android:textSize="16sp"  android:textStyle="bold" />  <TextView  android:id="@+id/itemDescription"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="No data"  android:textSize="16sp" />  <TextView  android:id="@+id/itemWind"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp"  tools:text="Wind: 0 m/s"  android:textSize="16sp" />  <TextView  android:id="@+id/itemPressure"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Pressure: 0 hpa"  android:textSize="16sp" />  <TextView  android:id="@+id/itemHumidity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="Humidity: 0 %"  android:textSize="16sp" />  </LinearLayout>  <TextView  android:id="@+id/itemIcon"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentEnd="true"  android:layout\_alignParentRight="true"  android:layout\_alignParentTop="true"  android:layout\_marginRight="16dp"  android:layout\_marginTop="16dp"  android:layout\_weight="1"  tools:text="o"  android:textSize="48sp" />  <TextView  android:id="@+id/itemTemperature"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  tools:text="0 °C"  android:textSize="22sp"  android:layout\_marginTop="8dp"  android:layout\_below="@+id/itemIcon"  android:layout\_alignRight="@+id/itemIcon"  android:layout\_alignEnd="@+id/itemIcon" />  </RelativeLayout> |

**PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

### CHƯƠNG 4: XÂY DỰNG APP DỰ BÁO THỜI TIẾT

### 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

Ứng dụn

g

Nhập địa

điểm

Xem Thông Tin Thời

Tiết Hiện Tại

Hiển Thị Thông Tin Thời

Tiết 5 Ngày Tiếp Theo

* 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 5 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 demo.



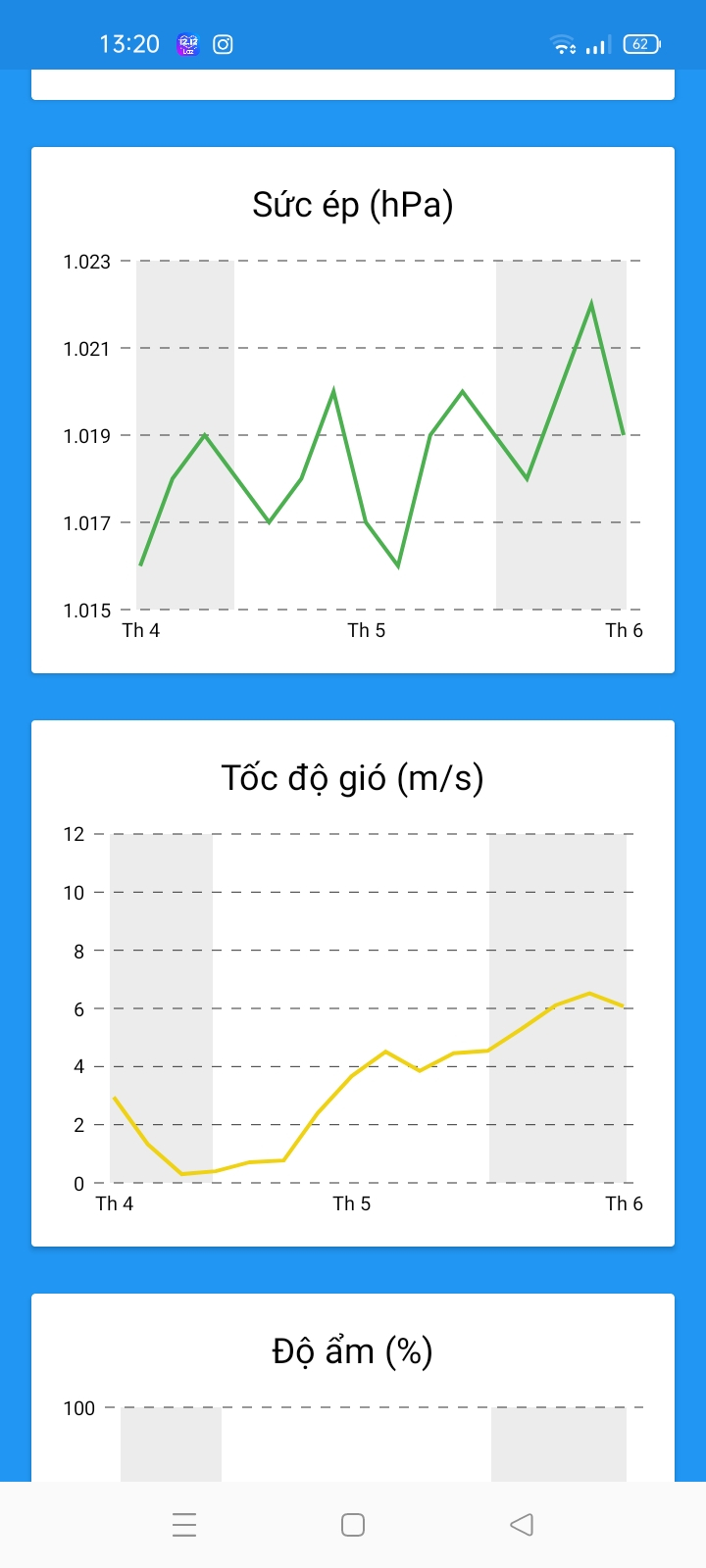
Hình 4.4.1. Thiết kế màn hình chính



Hình 4.4.2 Thiết kế thanh menu

### 

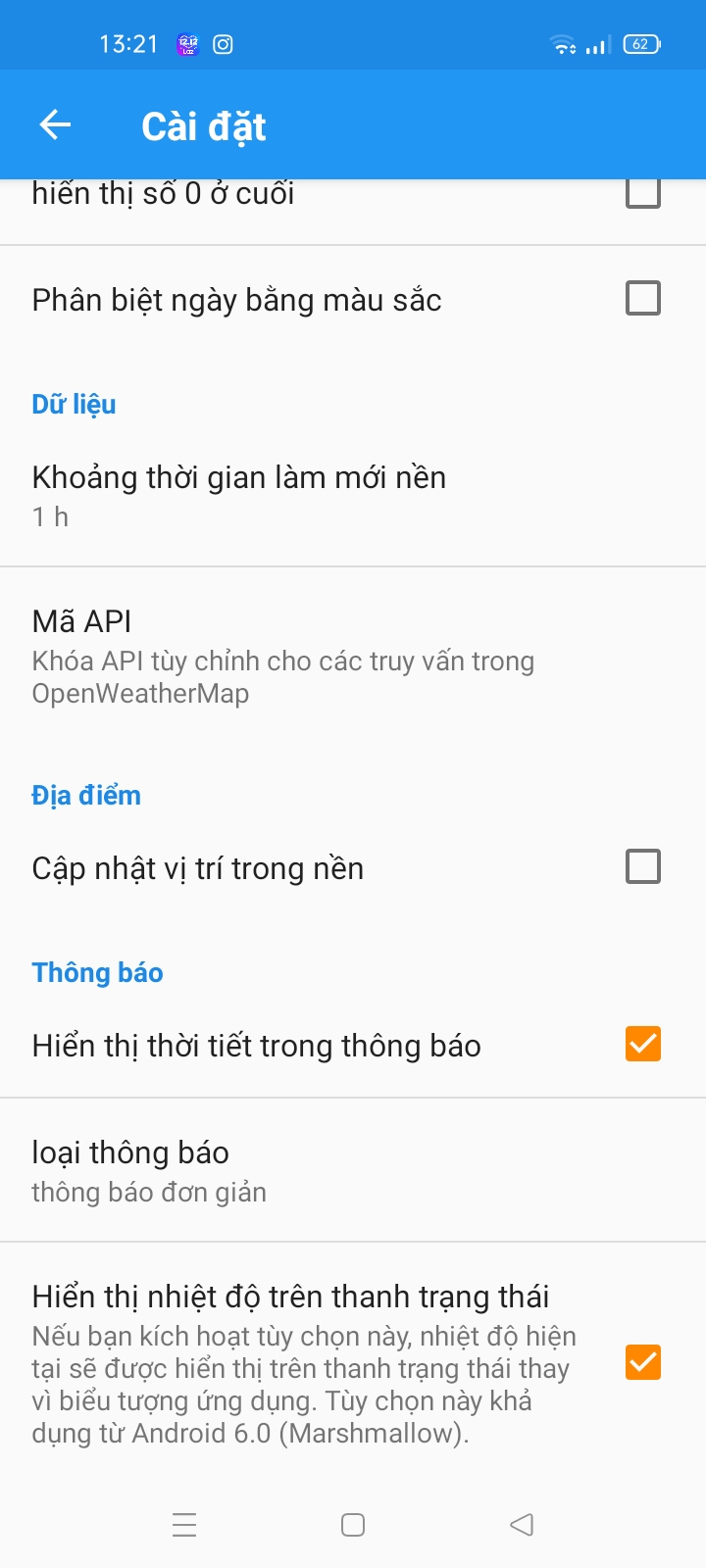
Hình 4.4.3. Thiết kế bản đồ thời tiết trong thanh menu

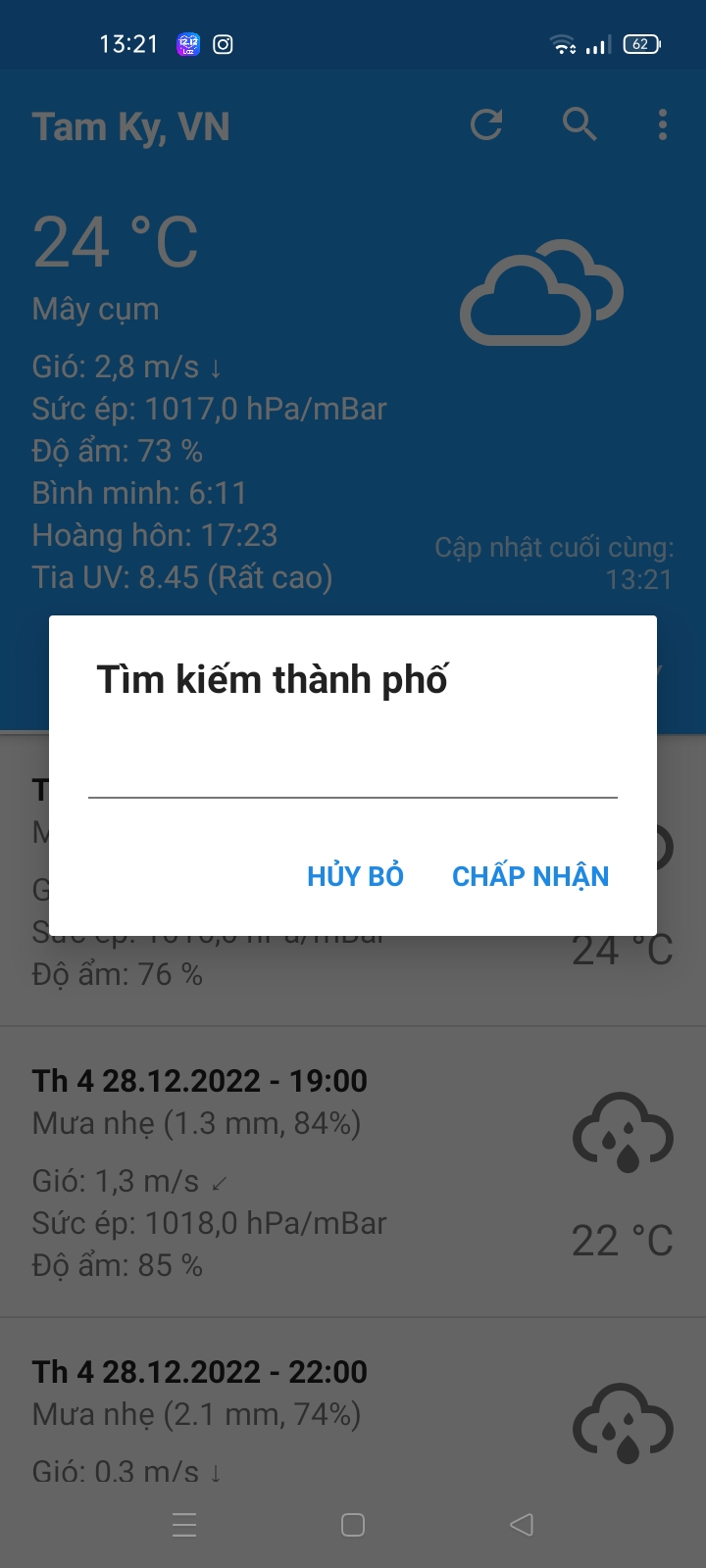
Hình 4.4.3. Thiết kế đồ họa trong thanh menu 1 ngày

### 

Hình 4.4.4. Thiết kế đồ họa trong thanh menu trong 5 ngày

Hình 4.4.5. Thiết kế cài đặt trong thanh menu



### 

Hình 4.4.7. Thiết kế thanh tìm kiếm thành phố

Hình 4.4.8. thời tiết của 2 thành phố



Hình 4.4.9. thời tiết ngày mai của đà nẵng



Hình 4.5 thời tiết 4 ngày sau ở hà nội

# **CHƯƠNG 5: KẾT LUẬN**

Kết quả đạt được

* Hoàn thành về cơ bản yêu cầu của đề tài là tìm hiểu, phân tích và thiết kế hệ thống cuả App
* Hoàn thành đúng tiến độ yêu cầu.
* Về mặt giao diện: App đơn giản, dễ thực hiện, đáp ứng cho người tiêu dung
* Đem lại thông tin thời tiết cho người dùng cần thiết
* Trải nghiệm người đùng đơn giản
* Tốc độ tải và khả năng xử lý tức thời.

Sau khi làm xong đồ án này, em biết đc những gì?????? Đó là kết quả đạt được

**HƯỚNG PHÁT CHUYỂN**

* Đẩy thông báo thời gian khi có mưa
* tự động lấy thông tin thời tiết đưới nền
* thông tin thời tiết cụ thể hơn

**TÀI LIỆU THAM KHẢO**

[1]. https://vi.wikipedia.org/wiki/Android\_(hệ\_điều\_ hành)

[2].http://genk.vn/mobile/lich-su-va-chang-duong-phat-trien-cua-android-quacac-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>

[6]. [http://giasutinhoc.vn/lap-trinh-android-co-ban/xu-ly-su-kien-trongandroid-bai-3/](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/