

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



NIÊN LUẬN CƠ SỞ MẠNG MÁY
TÍNH VÀ TRUYỀN THÔNG

ĐỀ TÀI: Xây dựng ứng dụng mobile xem thời tiết
và chất lượng không khí sử dụng
Openweathermap.org APIs

Sinh viên thực hiện

Họ và tên sinh viên: Trần Thanh Hoà

MSSV: B190387

Lớp: DI19T9A2

Giảng viên hướng dẫn

TS. Trần Thị Tố Quyên

Học kỳ 2, năm học 2021-2022

MỤC LỤC

LỜI CẢM ƠN	4
LỜI MỞ ĐẦU	5
CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID	6
1.1 Giới thiệu hệ điều hành Android	6
1.2 Sự phát triển của hệ điều hành Android	7
1.3 Kiến trúc của hệ điều hành Android	10
CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI	15
2.1 Mô tả đề tài.....	15
2.2 Mục tiêu đề tài.....	15
2.3 Hướng giải quyết	16
CHƯƠNG 3: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO	17
3.1 Sơ lược về Android Studio	17
3.2 Cài đặt Android Studio	17
3.2.1 Yêu cầu phần cứng máy tính	17
3.2.2 Phần mềm Android Studio.....	17
3.3 Cấu trúc dự án Android Studio	19
3.3.1 Tạo mới một project	19
3.3.2 Màn hình làm việc của dự án Android Studio	22
3.2.3 Các thành phần trong một ứng dụng Android	24
3.2.4 Bắt và xử lý sự kiện trên giao diện	26
CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG THỜI TIẾT	28
4.1 Tìm hiểu về OpenWeatherMap.....	28
4.1.1 Các chức năng của API OpenWeatherMap	28

4.1.2 Đăng ký và kích hoạt sử dụng key API OpenWeatherMap	29
4.2 Thiết kế giao diện và xử lý các chức năng mà đề tài yêu cầu	34
4.2.1 Giao diện màn hình chính	34
4.2.2 Dự báo thời tiết trong 7 ngày tới ở mỗi ngày	38
4.2.3 Dự báo thời tiết trong 2 ngày tới ở mỗi giờ	41
4.2.4 Dự báo thời tiết hiện tại.....	44
4.2.5 Xem lịch sử thời tiết theo ngày (trong vòng 30 ngày).....	47
4.2.6 Xem lịch sử thời tiết trong vòng 30 ngày, tính trên mỗi giờ	50
4.3 Sơ đồ khối tổng thể của ứng dụng thời tiết	53
CHƯƠNG 5: KẾT LUẬN	55
5.1 Kết quả đạt được của đề tài.....	55
5.2 Hạn chế của đề tài.....	55
5.3 Hướng phát triển	55
TÀI LIỆU THAM KHẢO	57

LỜI CẢM ƠN

Qua thời gian học tập và nghiên cứu tại trường Đại học Cần Thơ, đầu tiên em xin chân thành cảm ơn sâu sắc tới thầy giáo GS.TS Hà Thanh Toàn hiệu trưởng trường Đại học Cần Thơ, 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 Cần Thơ, 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 niên luận cơ sở này.

Đặc biệt em xin chân thành cảm ơn cô TS. Trần Thị Tố Quyên - Khoa Công nghệ thông tin - Trường Đại học Cần Thơ. Trong thời gian làm vừa qua, cô 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 cô cũng như tất cả các bạn để bài niên luận cơ sở của em được hoàn thiện hơn. Em xin chân thành cảm ơn!

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. Adroid 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.

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.

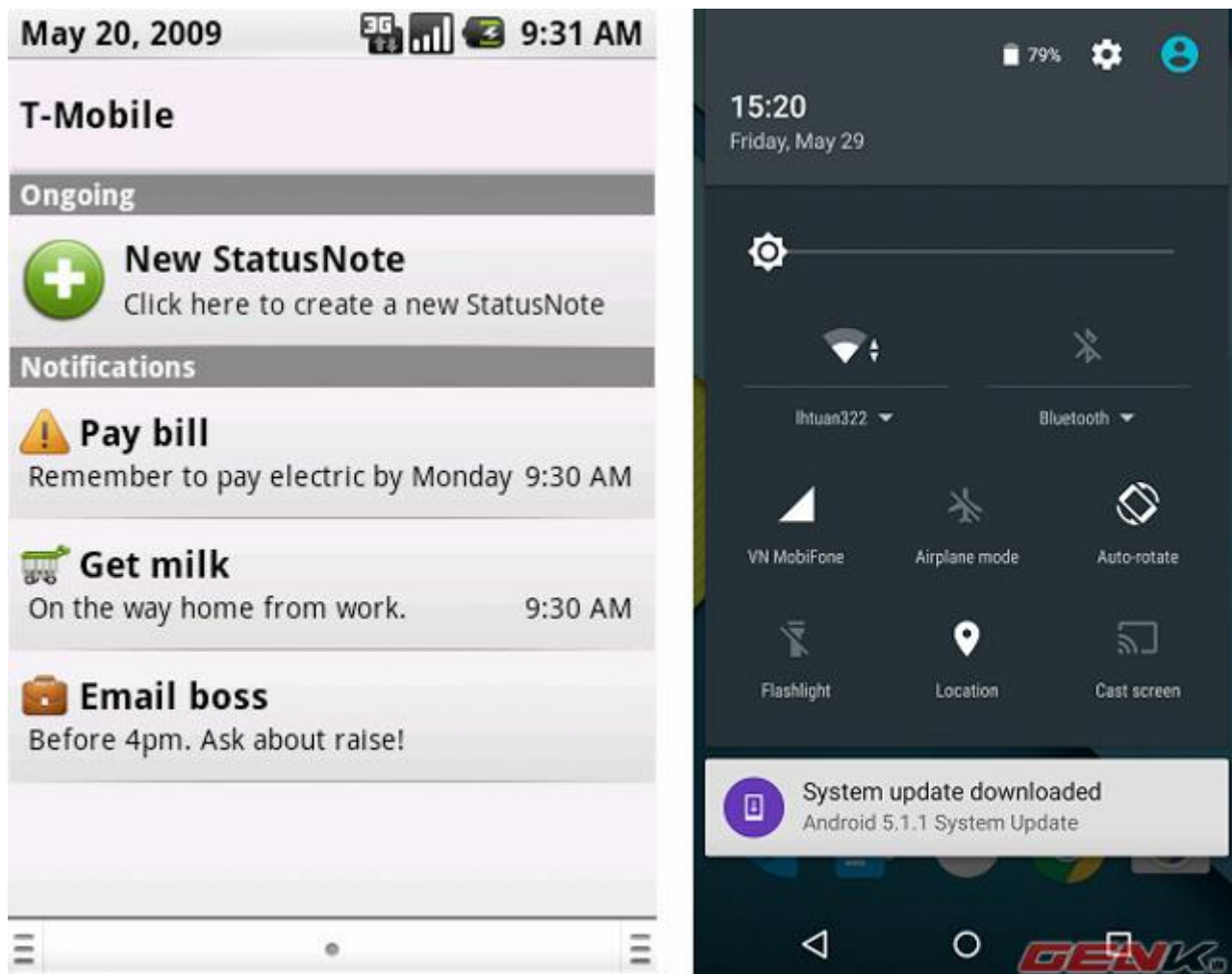
Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

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, Jellybean, Kitkat, Lollipop, Marshmallow, Nougat và bây giờ là Oreo.

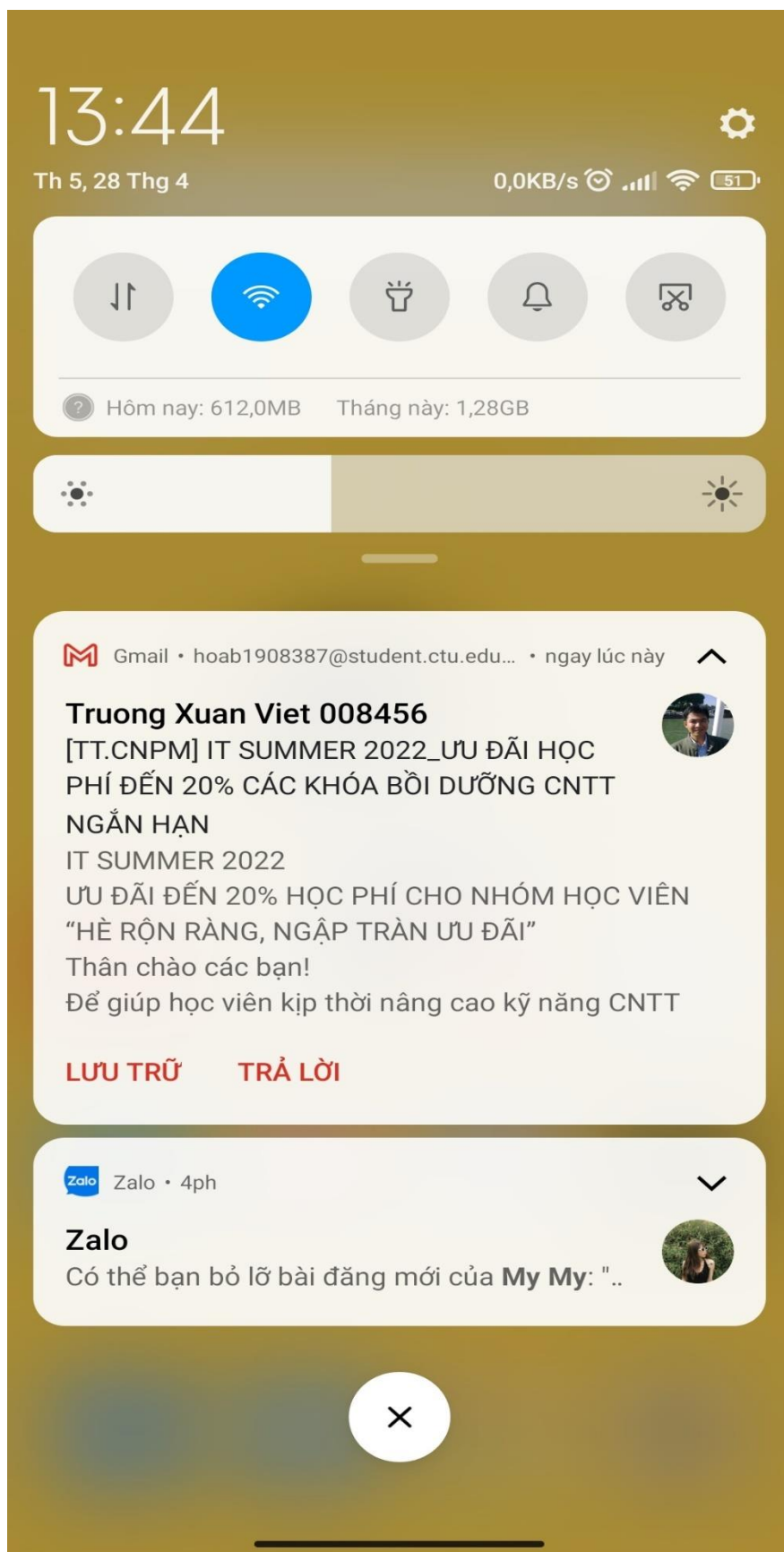
Kỷ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán ra tại Mỹ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng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

1.2 Sự phát triển của hệ điều hành Android

Thanh thông báo vuốt từ trên xuống (Notification bar): Ngay từ những ngày đầu tiên của Android, thanh thông báo này đã đánh dấu một bước quan trọng mà trước đây chưa hề có hệ điều hành nào làm được - đưa tất cả thông tin tin nhắn, tin thoại hoặc các cuộc gọi nhớ chỉ với thao tác vuốt xuống như **hình 1.2.1** và **hình 1.2.2**



Hình 3.2.1.1. Phiên bản hệ điều hành Android thời kì đầu



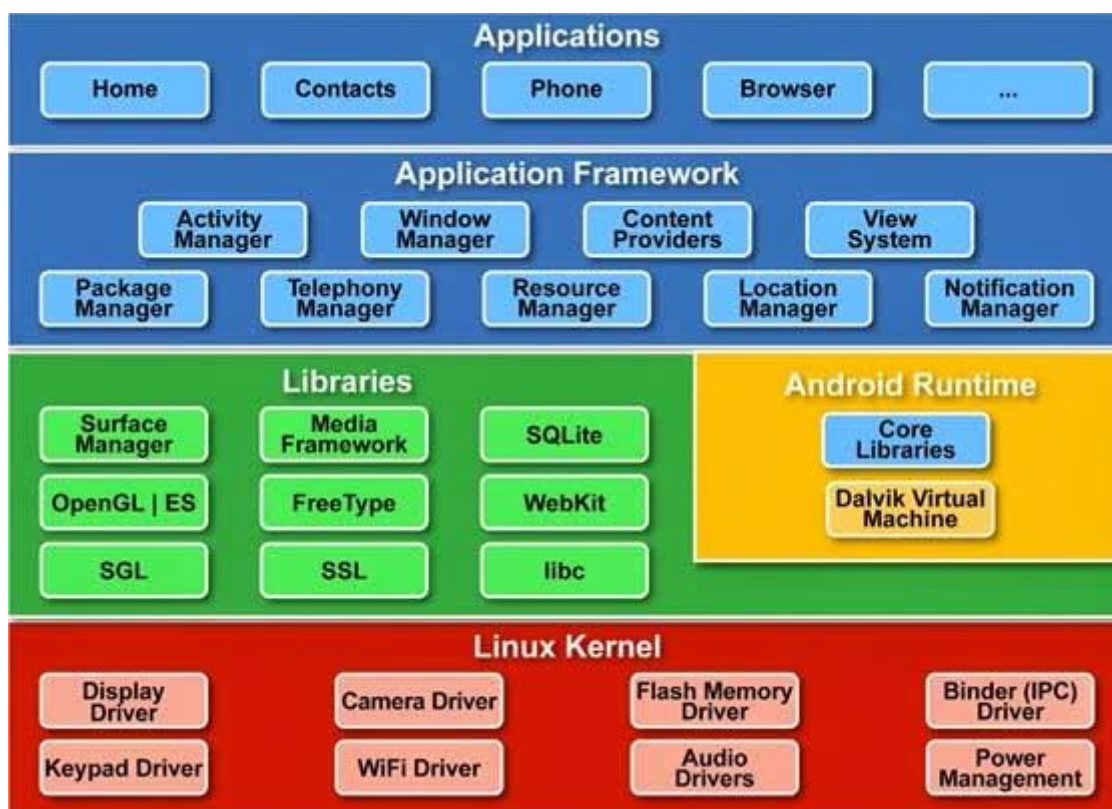
Hình 3.2.1.2 Phiên bản hệ điều hành hiện tại

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 3.2.1.3**



Hình 3.2.1.3 Màn hình chính Android 1.0 (trái) so với Android 5.1 (phải).

1.3 Kiến trúc của hệ điều hành Android



Hình 3.2.1.1 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.

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.
- 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.

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

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 3.2.1.2 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, ...

CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI

2.1 Mô tả đề tài

Openweathermap.org là một địa chỉ cung cấp các APIs từ miễn phí cho đến có phí giúp các nhà phát triển ứng dụng về thời tiết có thể truy xuất và lấy dữ liệu. Giới hạn của đề tài niên luận chỉ sử dụng các gói APIs miễn phí từ website này.

Ứng dụng được thiết kế gồm 2 thành phần chính:

a) Server: Đóng vai trò:

- Tương tác với openweathermap.org để lấy dữ liệu cần thiết và lưu vào cơ sở dữ liệu. Mục đích của việc lưu trữ lại dữ liệu là để sử dụng cho các report, history view, analysis sau này.
- Phát triển các APIs để ứng dụng mobile có thể liên hệ và lấy các dữ liệu cần thiết.

b) Mobile App: Phát triển giao diện chính để người dùng có thể sử dụng các chức năng:

- Xem thời tiết tại vị trí hiện tại của người dùng (nhiệt độ, tốc độ gió, mây, mưa)
- Cho phép người dùng lựa chọn địa điểm (thành phố/tỉnh) để xem thời tiết nơi đó
- Dự báo thời tiết trong 2 ngày tới ở mỗi giờ
- Dự báo thời tiết trong 7 ngày tới ở mỗi ngày
- Xem lịch sử nhiệt độ và mưa/nắng trong vòng 30 ngày, dữ liệu được tính trên mỗi giờ

2.2 Mục tiêu đề tài

Giúp sinh viên nắm vững được kiến trúc client-server, hiểu và triển khai được một ứng dụng mang tính thực tế. Thông qua đề tài sinh viên có khả năng tự phát triển một ứng dụng và triển khai lên môi trường thực tế phục vụ nhu cầu sử dụng của người dùng

2.3 Hướng giải quyết

- Phân tích API OpenWeatherMap để nắm được bản chất của API này
- Tiến hành lấy dữ liệu và đồ dữ liệu ra màn hình với giao diện tương ứng
- Đẩy dữ liệu lên server để lưu lại và tương tác khi cần thiết

CHƯƠNG 3: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO

3.1 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.

3.2 Cài đặt Android Studio

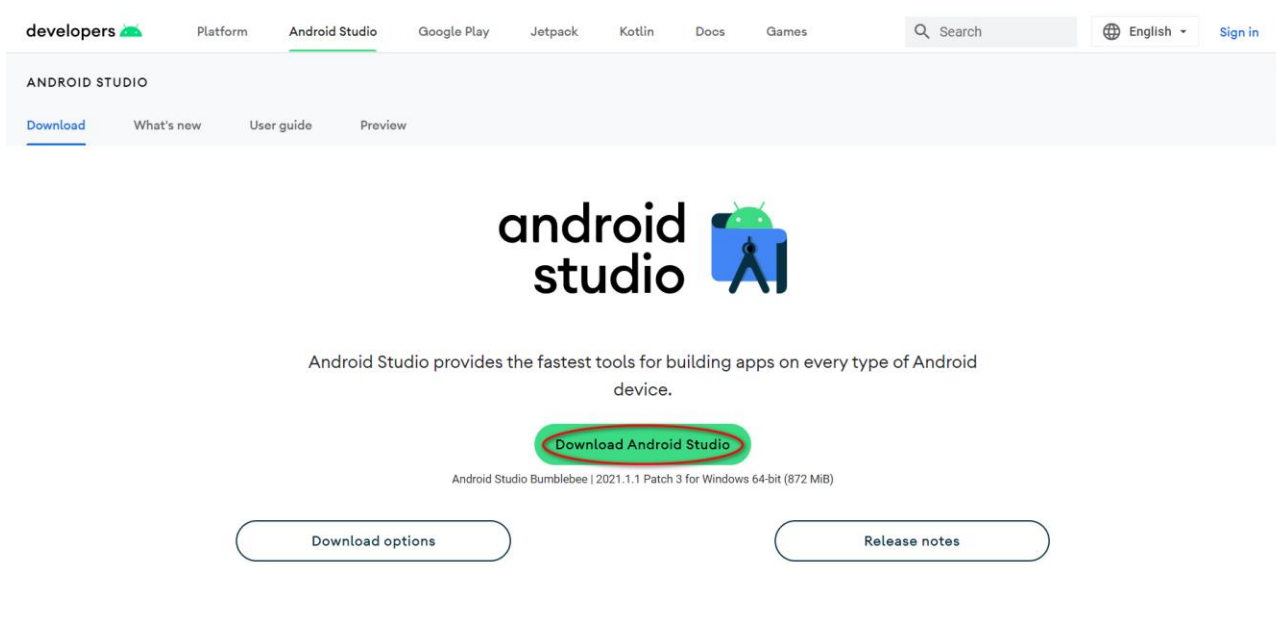
3.2.1 Yêu cầu phần cứng máy tính

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

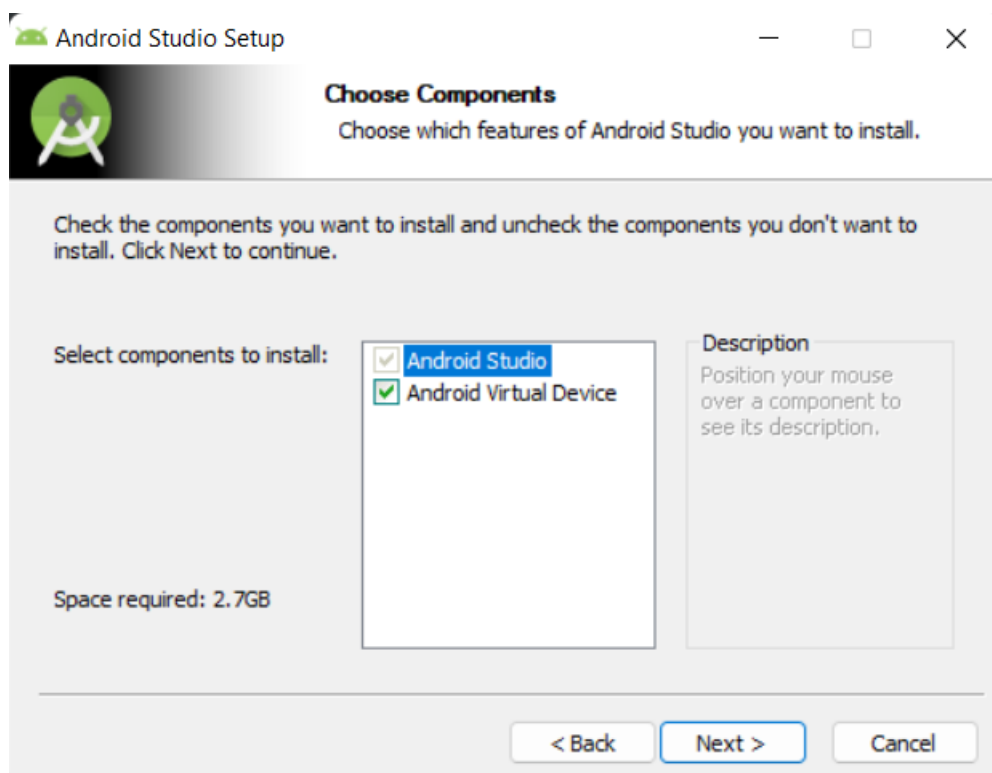
3.2.2 Phần mềm Android Studio

- Vào đường dẫn: “<https://developer.android.com/studio/>” để download bản mới nhất và tiến hành cài đặt như **hình 3.2.2.1**:



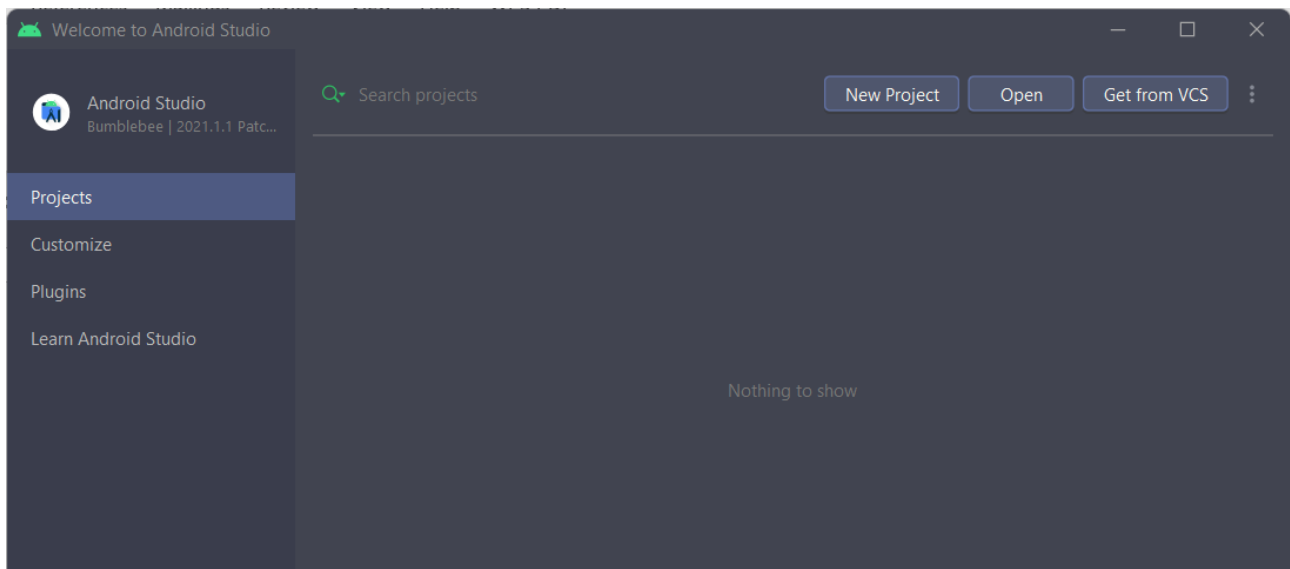
Hình 3.2.2.1 Trang download Android Studio

- Tiến hành cài đặt Android Studio



Hình 3.2.2.2 Giao diện cài đặt Android Studio

- Tiếp tục chọn next và agree cho đến khi hoàn tất.
- Khi việc cài đặt hoàn tất, chúng ta sẽ được như **hình 2.2.2.3**:

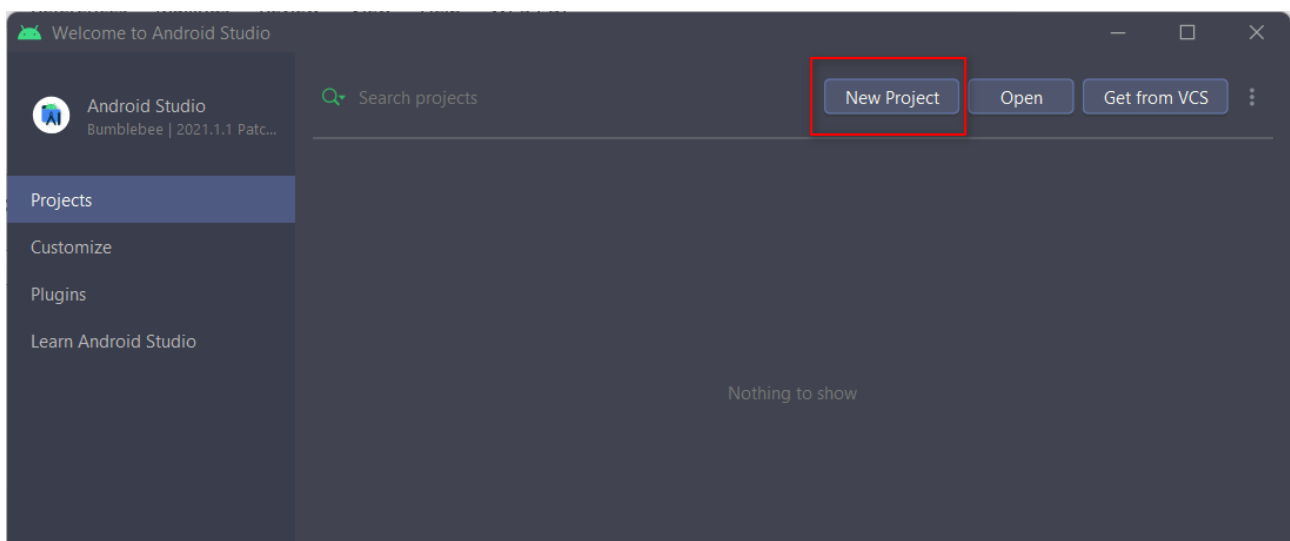


Hình 3.2.2.3 Giao diện cửa sổ Android Studio khi cài đặt hoàn tất

3.3 Cấu trúc dự án Android Studio

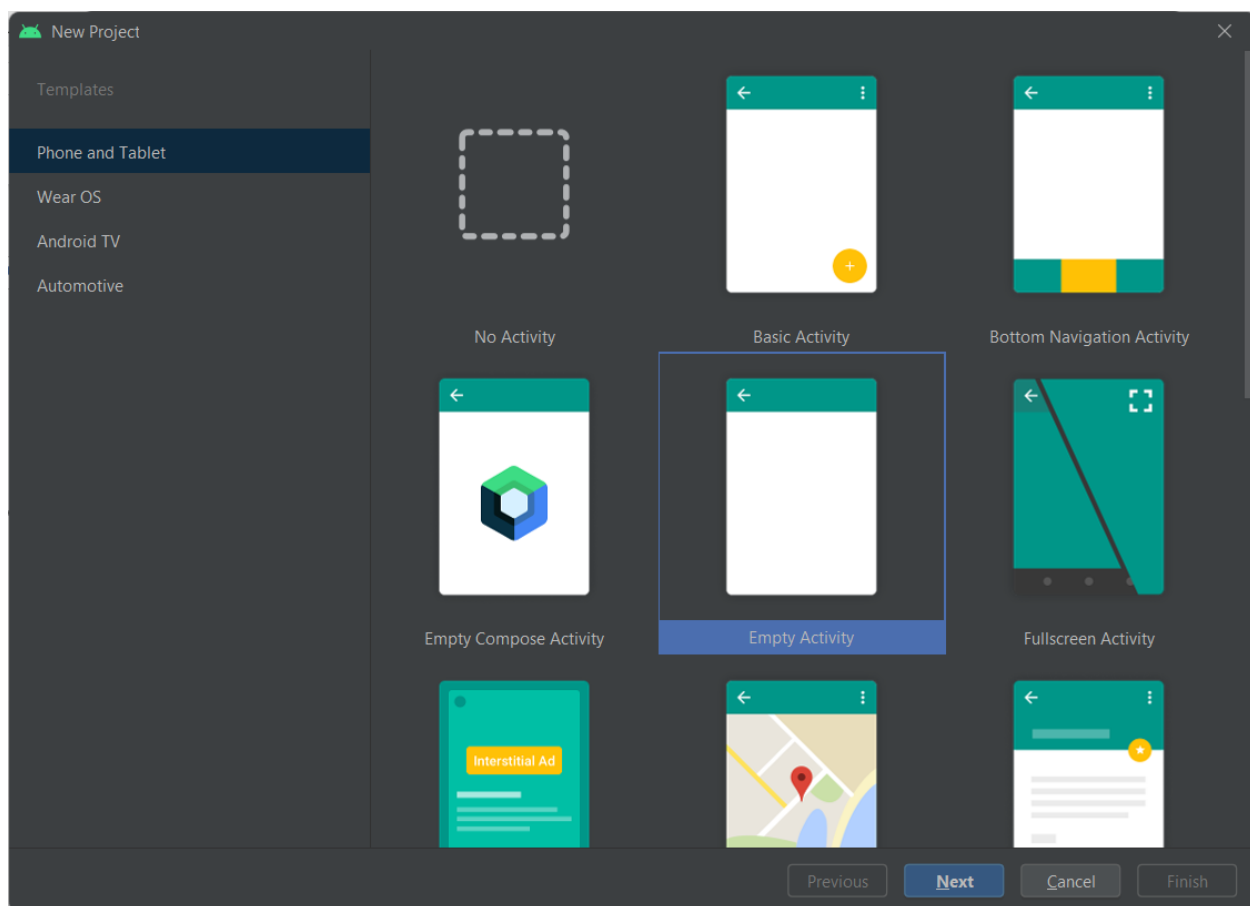
3.3.1 Tạo mới một project

Bước 1: Click vào dòng “New Project” như **hình 2.3.1.1**.



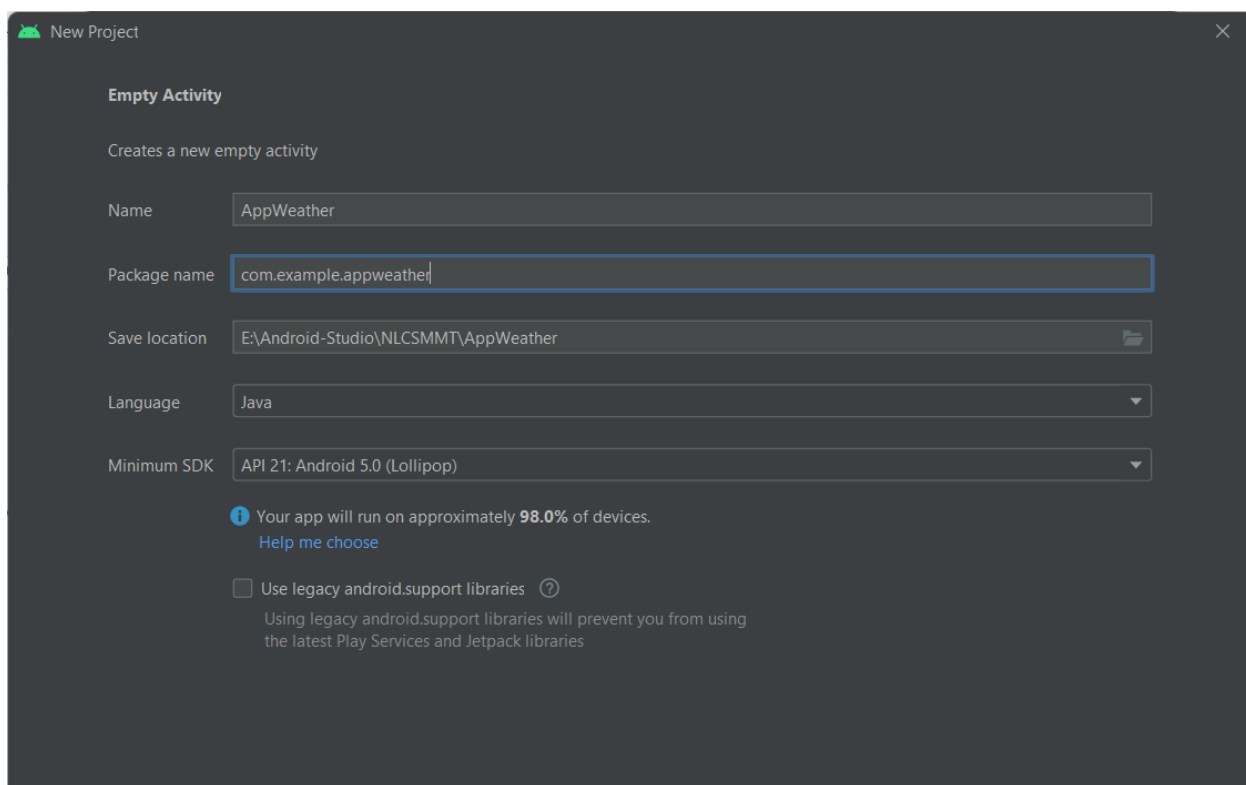
Hình 3.3.1.1 Giao diện cửa sổ đầu tiên khi chọn New Project

Bước 2: Chọn loại activity là **Empty Activity**, nhấn Next để tiếp tục



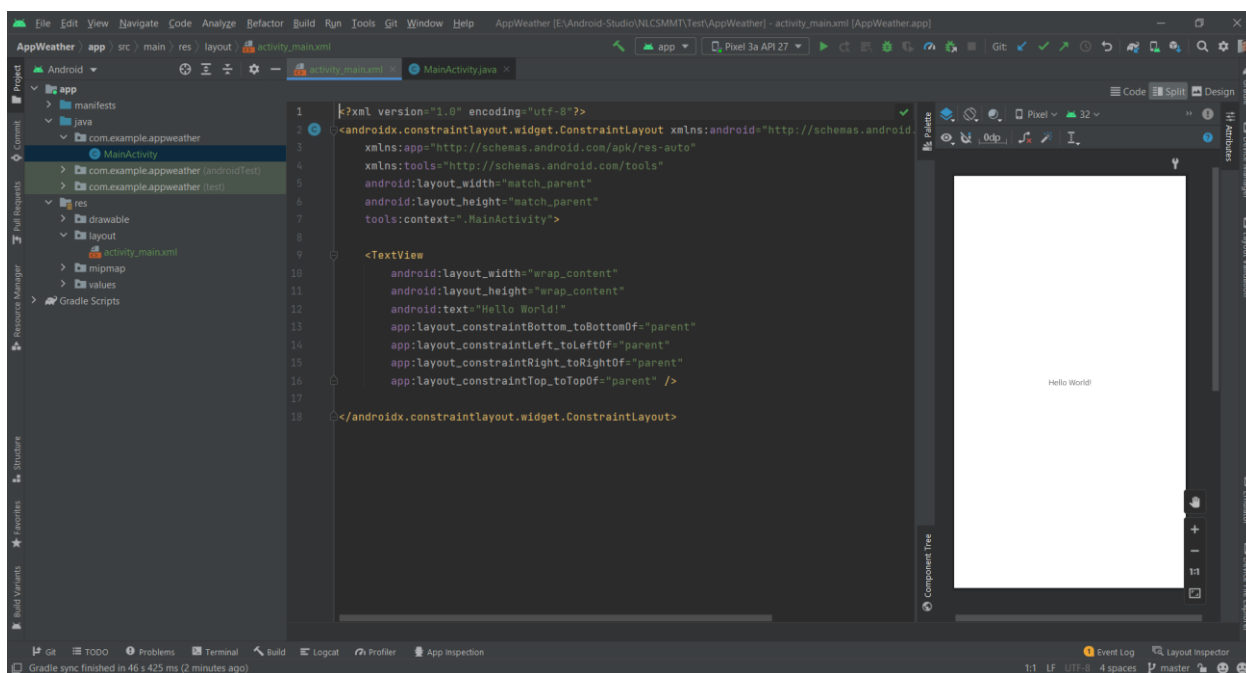
Hình 3.3.1.2 Chọn loại Activity

Bước 3: Nhập tên project, nơi lưu, ngôn ngữ code, và phiên bản hỗ trợ trên các thiết bị dành cho project

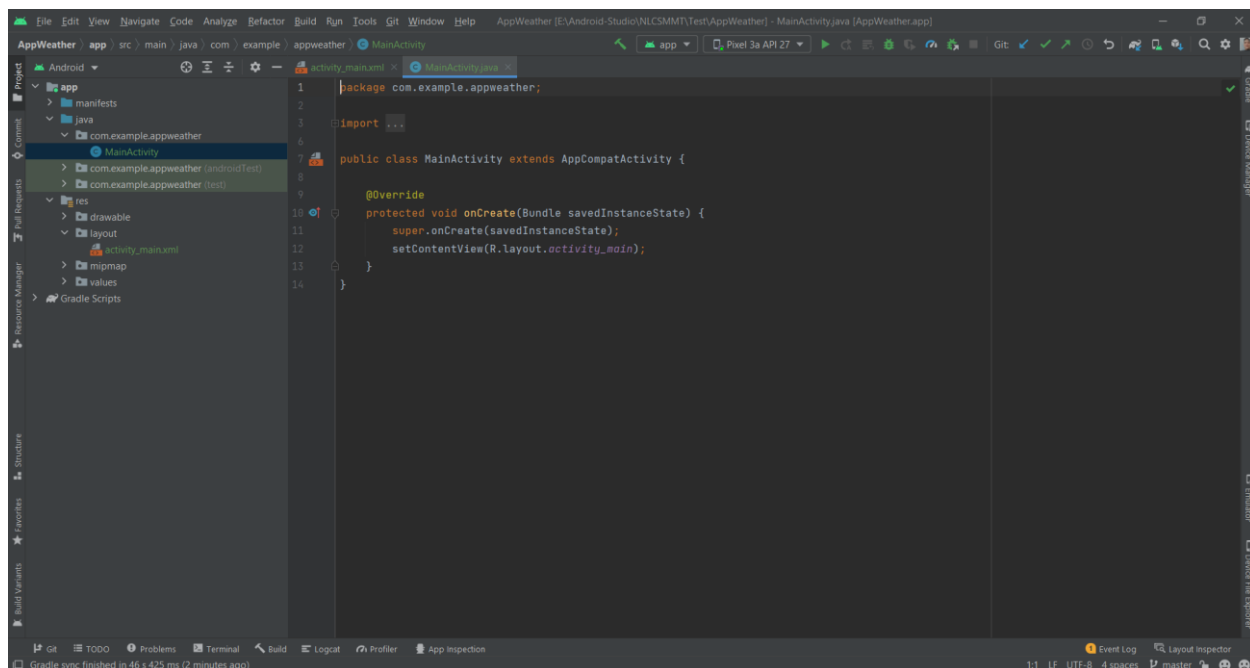


Hình 3.3.1.3 Các cấu hình ban đầu cho project

Bước 4: Sau khi cấu hình xong, nhấn Finished. Giao diện của project sau khi tạo thành công



Hình 3.3.1.4 Giao diện phần xml



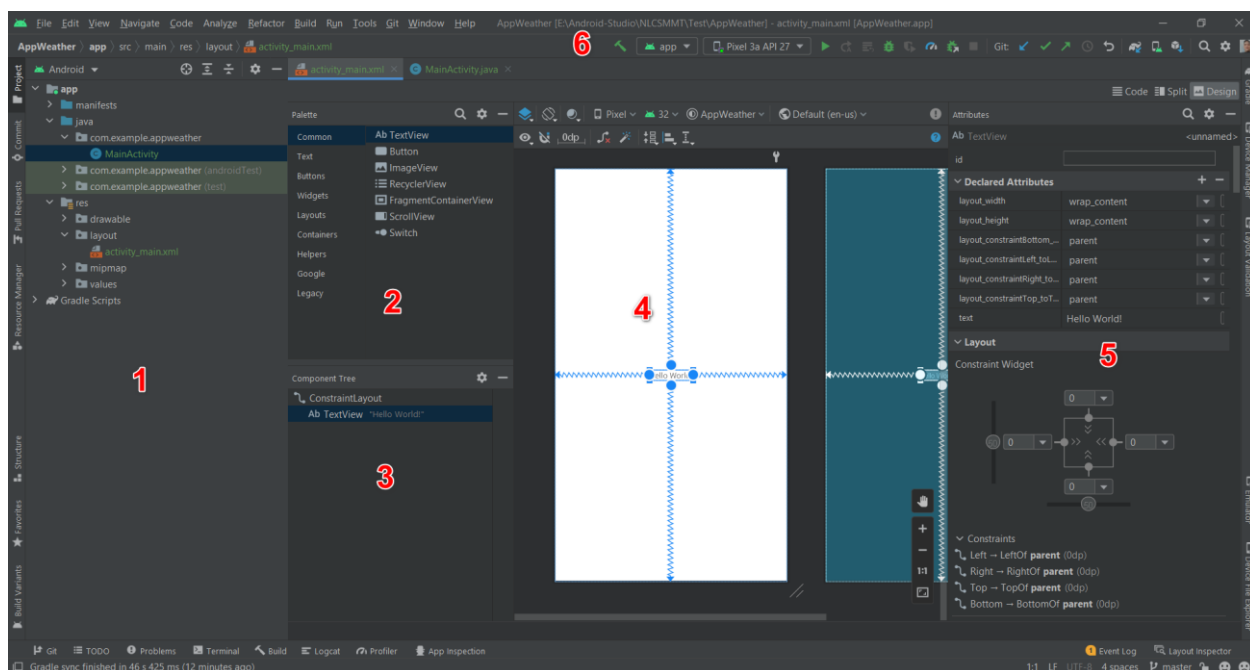
Hình 3.3.1.5 File dùng để viết code chương trình, cho project hoạt động

3.3.2 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:



Hình 3.3.2.1 Giao diện làm việc được chia thành 6 phần

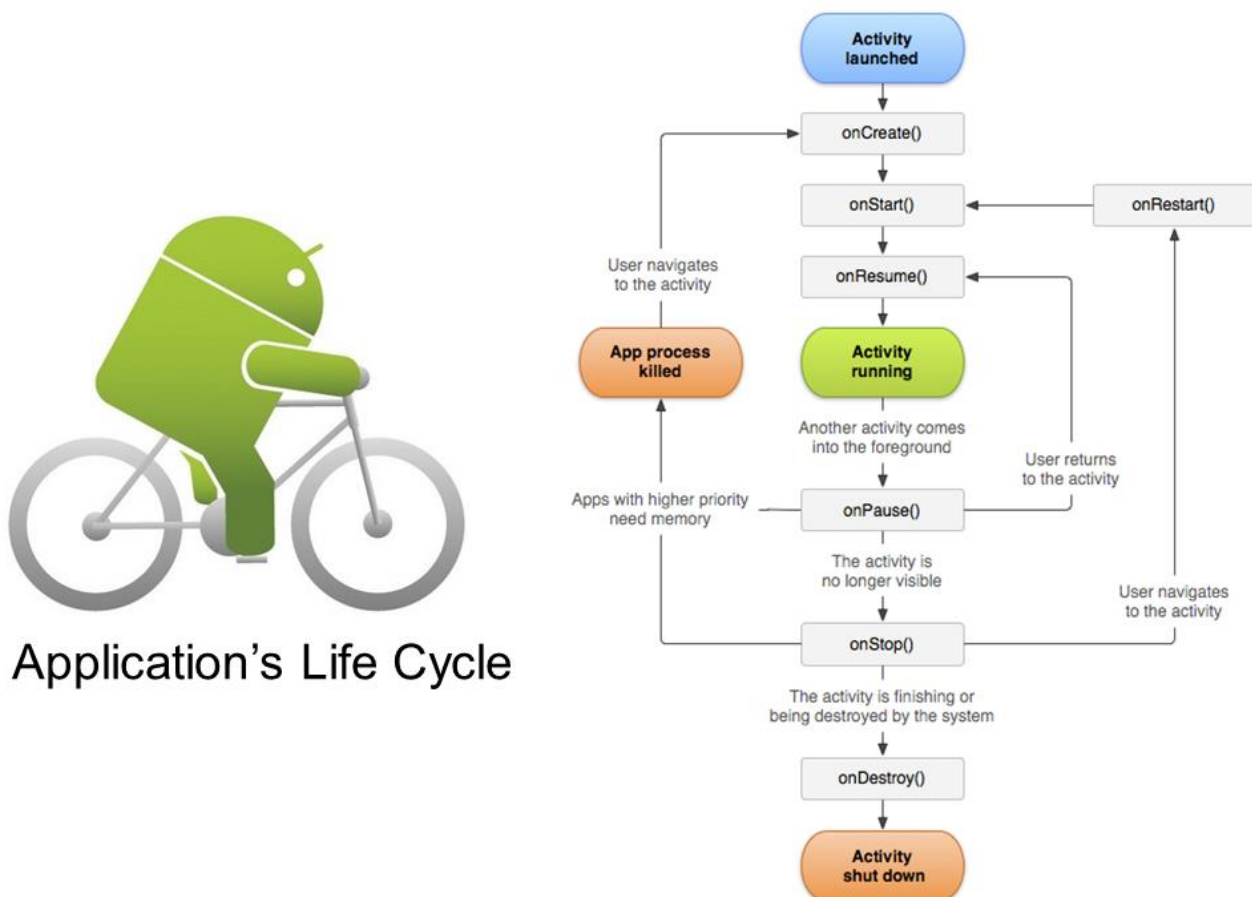
Ở hình 3.3.2.1 tạm thời chia làm 6 vùng làm việc mà lập trình viên thường tương tác:

- 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).
- 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.
- 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 lên trên giao diện (vùng 4).
- Vùng 4: Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiệu chỉnh control. Ta có thể chọn cách hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉnh 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
- Vùng 6: Là vùng các chức năng quan trọng thường dùng trong Android Studio.

3.2.3 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.

- a) **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.



Hình 3.2.3.1 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().
- b) 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.
- c) 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.
- d) 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.
- e) 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.

f) 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.

g) 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.

3.2.4 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,... Ta có thể nắm bắt những sự kiện trong chương trình và có những xử lý thích hợp theo yêu cầu.

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
---------------	---------------------

<i>onClick()</i>	<i>OnClickListener()</i> Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focuse trên bất kỳ widget như button, text, image vv. Chúng ta sẽ sử dụng phương onClick() để xử lý sự kiện
<i>onLongClick()</i>	<i>OnLongClickListener()</i> Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, trong một hoặc nhiều giây. Chúng ta sẽ sử dụng phương thức onLongClick() để xử lý sự kiện.
<i>onFocusChange()</i>	<i>OnFocusChangeListener()</i> Sự kiện phát sinh khi widget mất focus.
<i>onKey()</i>	<i>OnFocusChangeListener()</i> Sự kiện phát sinh khi người dùng focus trên widget và nhấn (presse) hoặc thả (release) một phím trên thiết bị.
<i>onTouch()</i>	<i>OnTouchListener()</i> Sự kiện phát sinh khi người dùng nhấn phím, nhả phím, hoặc bất kỳ cử chỉ chuyển động trên màn hình.
<i>onMenuItemClick()</i>	<i>OnMenuItemClickListener()</i> Sự kiện phát sinh khi người dùng chọn một mục trong menu.
<i>onCreateContextMenu()</i>	<i>OnCreateContextMenuItemClickListener()</i> Sự kiện phát sinh khi người dùng chọn một mục trong menu ngữ cảnh (Context Menu).

CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG THỜI TIẾT

4.1 Tìm hiểu về OpenWeatherMap

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

4.1.1 Các chức năng của API OpenWeatherMap

➤ *Thu thập dữ liệu thời tiết hiện tại và dự báo (Current & Forecast weather data collection)*

- a) Dữ liệu thời tiết hiện tại (Current Weather Data)
- b) Dự báo hàng giờ 4 ngày (Hourly Forecast 4 days)
- c) Dự báo hàng ngày 16 ngày (Daily Forecast 16 days)
- d) Dự báo khí hậu 30 ngày (Climatic Forecast 30 days)
- e) Tải xuống hàng loạt (Bulk Downloading)
- f) API bức xạ mặt trời (Solar Radiation API)
- g) Cảnh báo thời tiết toàn cầu (Global Weather Alerts)
- h) Dự báo 5 ngày / 3 giờ (5 Day / 3 Hour Forecast)
- i) API rủi ro đường bộ (Road Risk API)

➤ *Thu thập dữ liệu thời tiết lịch sử (Historical weather data collection)*

- a) API thời tiết lịch sử (Historical Weather API)
- b) API thời tiết lịch sử 40 năm theo dấu thời gian (Historical Weather API 40 years by timestamp)
- c) API thời tiết lịch sử kho lưu trữ đầy đủ 40 năm (Historical Weather API 40 years full archive)
- d) Lịch sử hàng loạt (History Bulk)
- e) Dự báo lịch sử hàng loạt (History Forecast Bulk)

- f) API dữ liệu thời tiết thống kê (Statistical Weather Data API)
- g) Các thông số tích lũy (Accumulated Parameters)
- h) Dữ liệu thời tiết lịch sử theo tiểu bang cho tất cả mã ZIP, Hoa Kỳ (Historical Weather Data by State for all ZIP codes, USA)

➤ **Bộ sưu tập bản đồ (Maps collection)**

- a) Bản đồ thời tiết 2.0 Bước 3 giờ (Weather Maps 2.0 3-hour step)
- b) Bản đồ thời tiết 2.0 Bước 1 giờ (Weather Maps 2.0 1-hour step)
- c) Bản đồ thời tiết 1.0 (Weather Maps 1.0)
- d) Bản đồ lượng mưa toàn cầu Dữ liệu lịch sử (Global Precipitation Maps Historical data)
- e) Dữ liệu lịch sử và dự báo của Bản đồ lượng mưa toàn cầu (Global Precipitation Maps Historical & Forecast data)
- f) Bản đồ cứu trợ (Relief Maps)

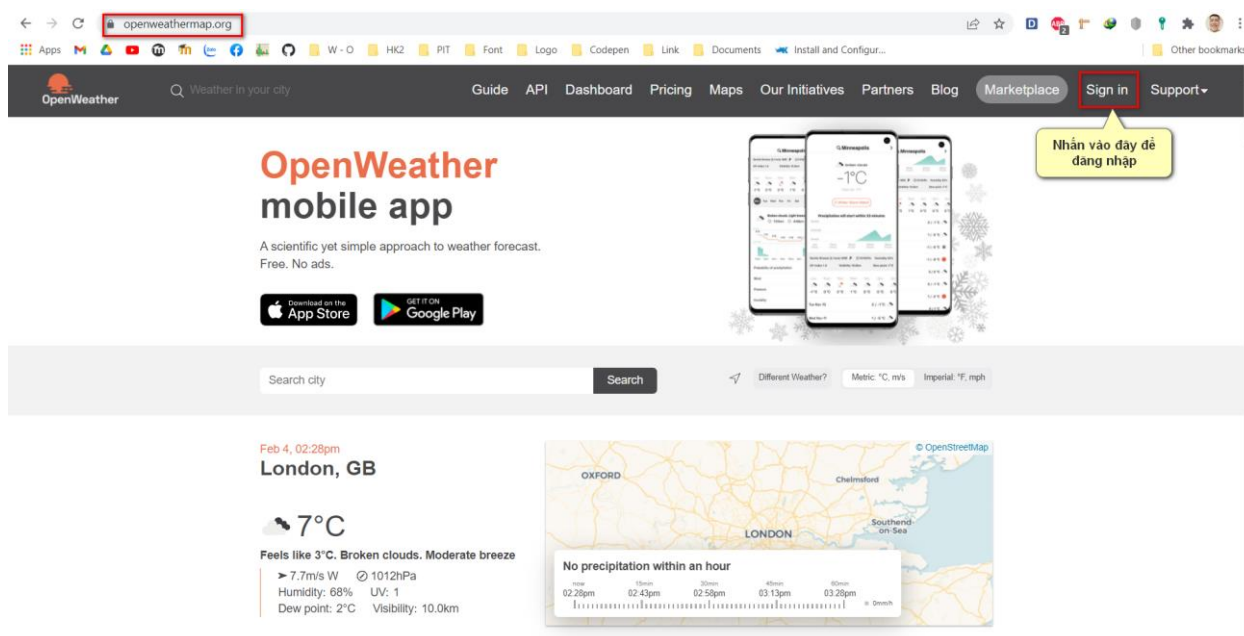
➤ **Bộ sưu tập API thời tiết khác**

- a) API ô nhiễm không khí (Air Pollution API)
- b) API mã hóa địa lý (Geocoding API)
- c) Trạm thời tiết (Weather Stations)
- d) Chỉ số UV (UV Index)
- e) Kích hoạt thời tiết (Weather Triggers)

4.1.2 Đăng ký và kích hoạt sử dụng key API OpenWeatherMap

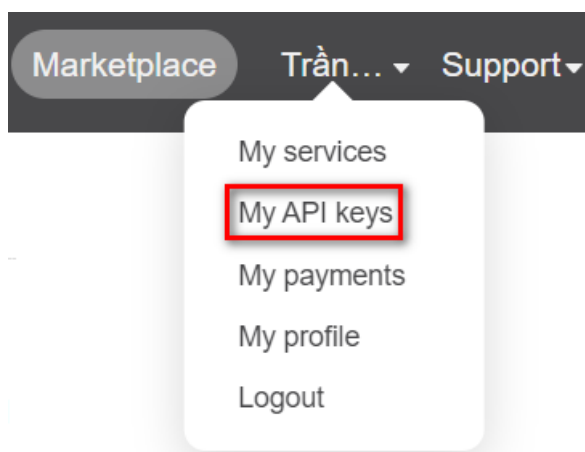
Bước 1: Vào trang chủ <https://openweathermap.org/>

Bước 2: Tạo tài khoản rồi đăng nhập vào trang chủ



Hình 4.1.2.1 Đăng nhập vào trang chủ OpenweatherMap

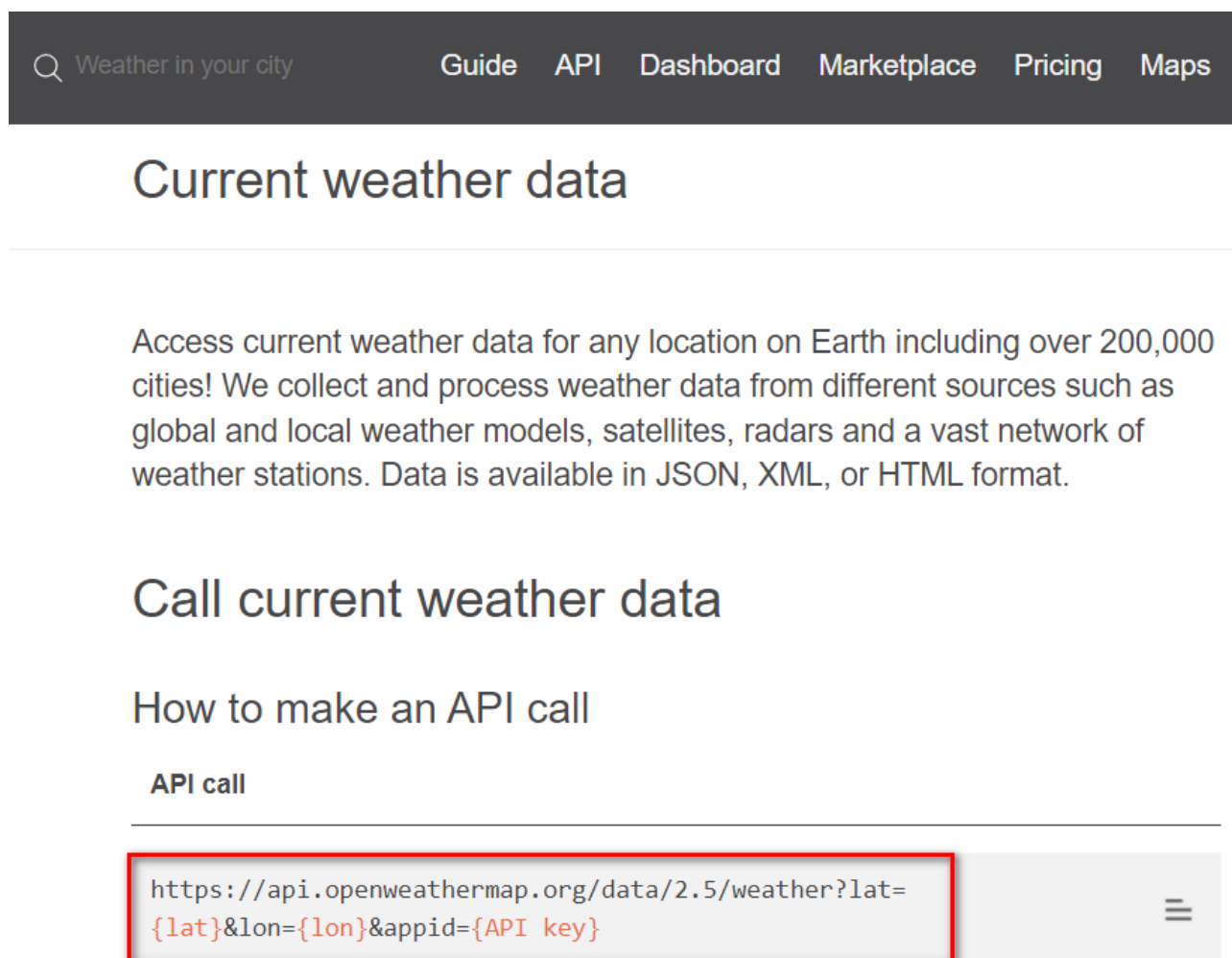
Bước 3: Sau khi đăng nhập thành công, vào mục My API keys để nhận key cá nhân



Hình 4.1.2.2 Đăng nhập thành công

Bước 4: Vào mục API và tùy theo từng yêu cầu mà lấy link API cho phù hợp

Cần sử dụng các API sau để hoàn thành ứng dụng thời tiết:

a) Current Weather Data (<https://openweathermap.org/current>)

Weather in your city Guide API Dashboard Marketplace Pricing Maps

Current weather data

Access current weather data for any location on Earth including over 200,000 cities! We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations. Data is available in JSON, XML, or HTML format.

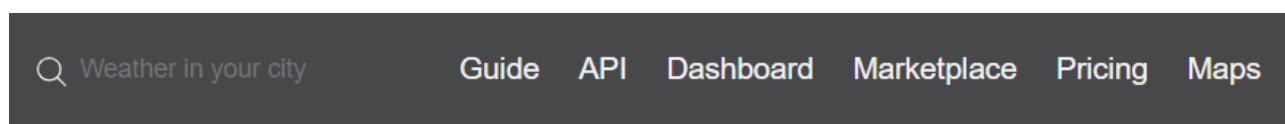
Call current weather data

How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
```

Hình 4.1.2.3 Current Weather Data

b) One Call API (<https://openweathermap.org/api/one-call-api>)

The One Call API 1.0 provides the following weather data for any geographical coordinates:

- Current weather
- Minute forecast for 1 hour
- Hourly forecast for 48 hours
- Daily forecast for 7 days
- National weather alerts
- Historical weather data for the previous 5 days

Current and forecast weather data

To get access to current weather, minute forecast for 1 hour, hourly forecast for 48 hours, daily forecast for 7 days and government weather alerts, please use this section of the documentation.

If you are interested in **historical weather data**, please read the "[Historical weather data](#)" section.

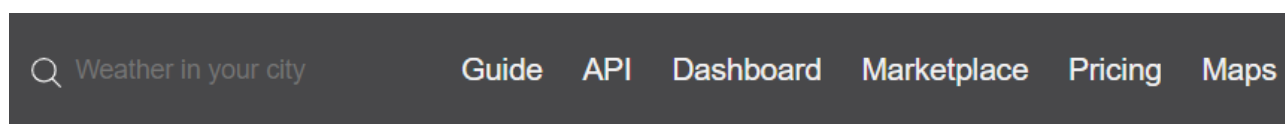
How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/onecall?lat={lat}&lon={lon}&exclude={part}&appid={API key}
```



Hình 4.1.2.4 One Call API

c) Air Pollution API (<https://openweathermap.org/api/air-pollution>)

Qualitative name	Index	Pollutant concentration in $\mu\text{g}/\text{m}^3$			
		NO ₂	PM ₁₀	O ₃	PM ₂₅ (optional)
Good	1	0-50	0-25	0-60	0-15
Fair	2	50-100	25-50	60-120	15-30
Moderate	3	100-200	50-90	120-180	30-55
Poor	4	200-400	90-180	180-240	55-110
Very Poor	5	>400	>180	>240	>110

Please follow [this link](#) to learn more about calculation of Air Quality index.

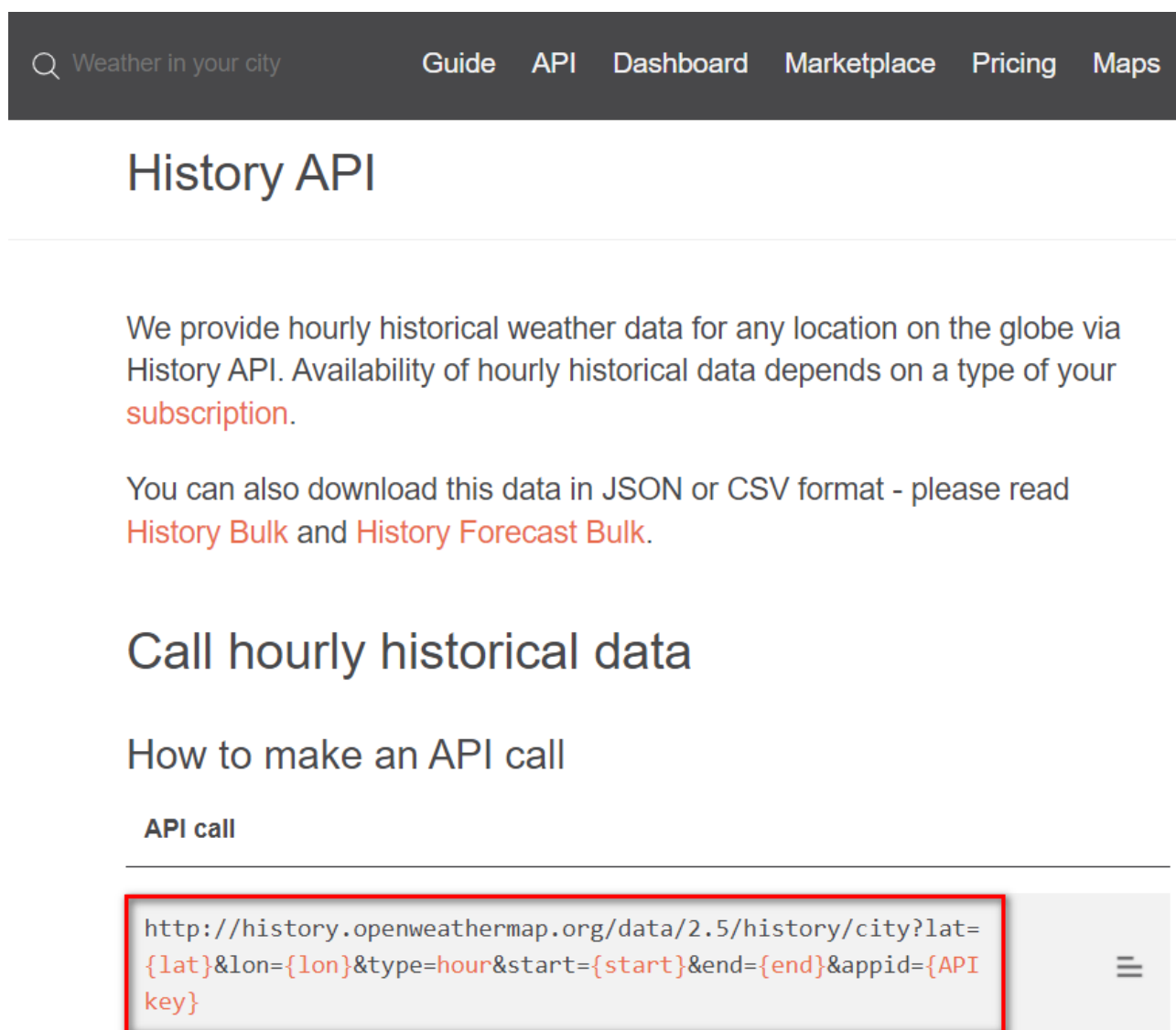
Current air pollution data

API call

```
http://api.openweathermap.org/data/2.5/air_pollution?lat={lat}&lon={lon}&appid={API key}
```



Hình 4.1.2.5 Air Pollution API

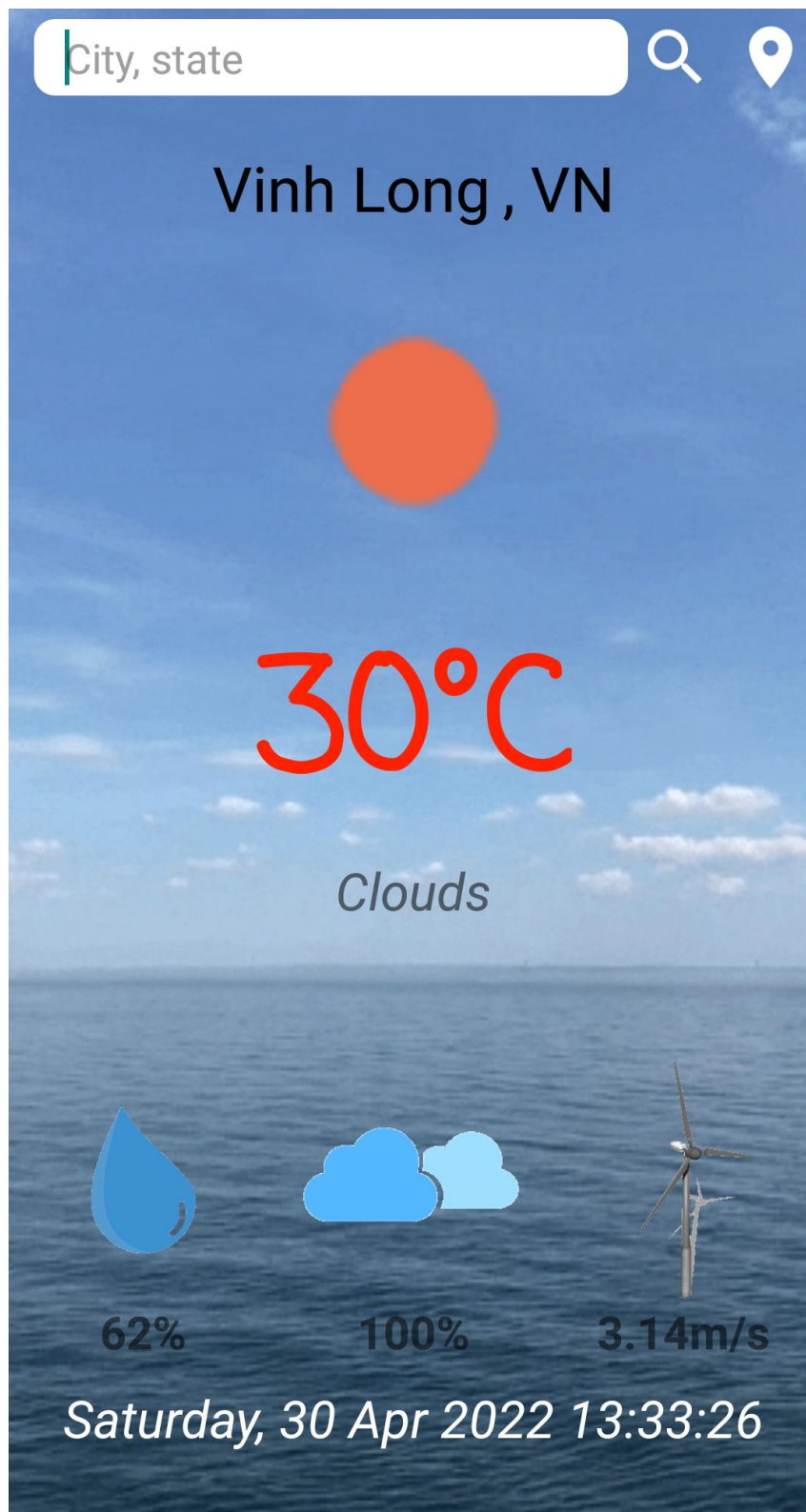
d) Historical Weather API (<https://openweathermap.org/history>)

Hình 4.1.2.6 Historical Weather API

4.2 Thiết kế giao diện và xử lý các chức năng mà đề tài yêu cầu**4.2.1 Giao diện màn hình chính****➤ Ý tưởng chung:**

- Thiết kế giao diện để hiển thị dữ liệu khi gọi từ API về màn hình ứng dụng
- Tạo ra các **Button** để khi bấm vào thì chuyển sang Activity khác tương ứng với từng chức năng khác nhau

- Sử dụng **Air Pollution API** (<https://openweathermap.org/api/air-pollution>) để lấy dữ liệu về Chất lượng không khí trong ngày, hiển thị với từng **TextView** tương ứng.



Hình 4.2.1.1 Giao diện màn hình chính _ 1



Hình 4.2.1.2 Giao diện màn hình chính _ 2



Hình 4.2.1.3 Giao diện màn hình chính _ 3

4.2.2 Dự báo thời tiết trong 7 ngày tới ở mỗi ngày

➤ Ý tưởng chung:

- Tạo activity có tên là **SevenDayWeather**, theo đó file xml tương ứng sẽ được tạo ra với tên là **activity_seven_day_weather**
- Tiến hành thiết kế giao diện trên **activity_seven_day_weather**

+ Do dữ liệu thời tiết trong vòng 7 ngày có thể sẽ được hiển thị ra màn hình ở dạng danh sách nên chúng ta cần tạo một **ListView** để hiển thị dữ liệu và có thể cuộn lên xuống được.

+ Để hiển thị được danh sách dữ liệu vào **ListView**, chúng ta cần phải thiết kế một **item** để đưa mỗi String vào một **View** con nằm trong **Listview**. Mỗi **item** đều có một layout riêng.

+ Khi sử dụng **Listview**, chúng ta cần tạo ra một **Adapter** để ánh xạ **item view** vừa tạo và **item** trong danh sách đầu vào.

➤ Xử lý chức năng:

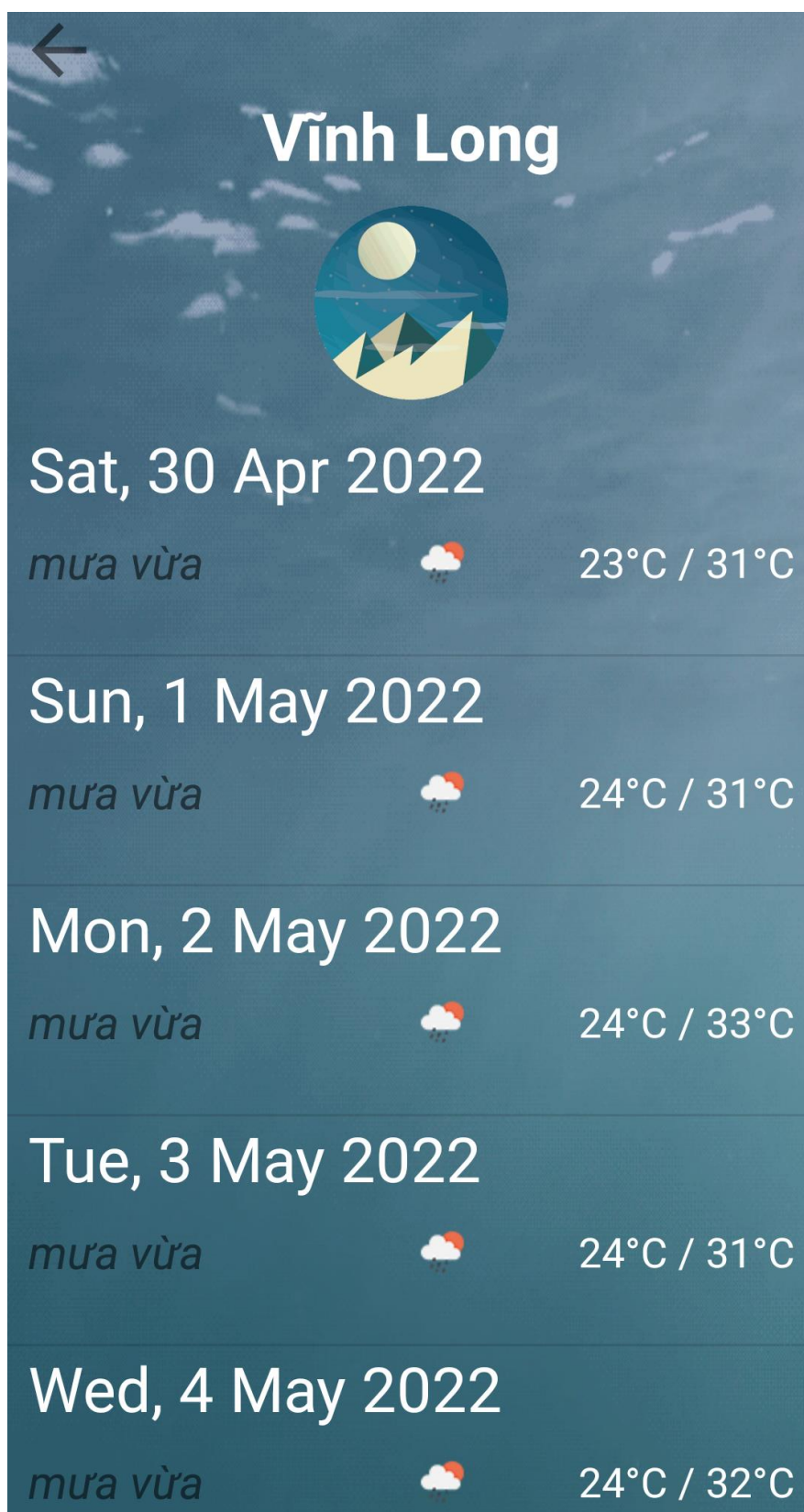
+ Việc gọi API sẽ được thông qua việc thực thi những **request** mà mình gửi đi, để làm được điều này cần sử dụng thư viện **Volley**

```
implementation 'com.android.volley:volley:1.2.1'
```

+ Sử dụng API **Current Weather Data** (<https://openweathermap.org/current>) để gọi về ứng dụng và lấy ra **lat**, **lon** và tên thành phố.

+ Tiếp theo, sử dụng **One Call API** (<https://openweathermap.org/api/one-call-api>) để gọi dữ liệu từ API này về và hiển thị ra màn hình. Dữ liệu sẽ được hiển thị dưới dạng danh sách dạng **Listview**.

➤ **Kết quả thực hiện:**



Hình 4.2.2.1 Dự báo thời tiết trong 7 ngày tới ở mỗi ngày _ 1



Hình 4.2.2.2 Dự báo thời tiết trong 7 ngày tới ở mỗi ngày _ 2

4.2.3 Dự báo thời tiết trong 2 ngày tới ở mỗi giờ

➤ Ý tưởng chung:

- Tạo activity có tên là **MainActivity**, theo đó file xml tương ứng sẽ được tạo ra với tên là **activity_main**
- Tiến hành thiết kế giao diện trên **activity_main**

+ Sử dụng **RecyclerView** để hiển thị dữ liệu theo dưới dạng danh sách ngang hoặc đứng **LinearLayoutManager**

+ Để hiển thị được danh sách dữ liệu vào **RecyclerView**, chúng ta cần phải thiết kế một **item** để đưa mỗi String vào một **View** con nằm trong **RecyclerView**. Mỗi **item** đều có một layout riêng, ở đây chúng ta sẽ sử dụng **CardView** để hiển thị dữ liệu.

+ Khi sử dụng **RecyclerView**, chúng ta cần tạo ra một **Adapter** để ánh xạ **item view** vừa tạo và **item** trong danh sách đầu vào.

➤ Xử lý chức năng:

+ Việc gọi API sẽ được thông qua việc thực thi những **request** mà mình gửi đi, để làm được điều này cần sử dụng thư viện **Volley**

```
implementation 'com.android.volley:volley:1.2.1'
```

+ Import các thư viện cần thiết để sử dụng **RecyclerView** và **CardView**:

```
// RecyclerView
implementation "androidx.recyclerview:recyclerview:1.2.1"
// Cardview
implementation "androidx.cardview:cardview:1.0.0"
```

+ Sử dụng API **Current Weather Data** (<https://openweathermap.org/current>) để gọi về ứng dụng và lấy ra **lat**, **lon** và tên thành phố.

+ Tiếp theo, sử dụng **One Call API** (<https://openweathermap.org/api/one-call-api>) để gọi dữ liệu từ API này về và hiển thị ra màn hình. Dữ liệu sẽ được hiển thị dưới dạng danh sách nằm ngang ở dạng **CardView**.

➤ **Kết quả thực hiện:**

Hình 4.2.3.1 Dự báo thời tiết trong 2 ngày tới ở mỗi giờ _ 1



Hình 4.2.3.2 Dự báo thời tiết trong 2 ngày tới ở mỗi giờ _ 2

4.2.4 Dự báo thời tiết hiện tại

➤ **Ý tưởng chung:**

- Tạo activity có tên là **CurrentWeather**, theo đó file xml tương ứng sẽ được tạo ra với tên là **activity_seven_day_weather**
- Tiến hành thiết kế giao diện trên **activity_current_weather**
- Xử lý chức năng:

+ Việc gọi API sẽ được thông qua việc thực thi những **request** mà mình gửi đi, để làm được điều này cần sử dụng thư viện **Volley**

```
implementation 'com.android.volley:volley:1.2.1'
```

+ Sử dụng API **Current Weather Data** (<https://openweathermap.org/current>) để gọi về ứng dụng và lấy ra **lat, lon** và tên thành phố.

+ Tiếp theo, sử dụng **One Call API** (<https://openweathermap.org/api/one-call-api>) để gọi dữ liệu từ API này về và hiển thị ra màn hình với từng **TextView** tương ứng.

➤ **Kết quả thực hiện:**



Hình 4.2.4.1 Dự báo thời tiết hiện tại _ 1

	Độ ẩm tương đối	62%
	Nhiệt độ khí quyển	22°C
	Mây	100%
	Chỉ số UV	9.58
	Tầm nhìn	10000m
	Tốc độ gió	3.14m/s
	Hướng gió	Tây Nam
	Gió giật	3.4km/h
	Lượng mưa	0mm
	Khả năng có tuyết rơi	0,0%

Hình 4.2.4.2 Dự báo thời tiết hiện tại _ 2

4.2.5 Xem lịch sử thời tiết theo ngày (trong vòng 30 ngày)

➤ Ý tưởng chung:

- Tạo activity có tên là **HistoryWeather**, theo đó file xml tương ứng sẽ được tạo ra với tên là **activity_history_weather**
- Tiến hành thiết kế giao diện trên **activity_history_weather**

+ Sử dụng **RecyclerView** để hiển thị dữ liệu theo dưới dạng danh sách ngang hoặc đứng **LinearLayoutManager**

+ Để hiển thị được danh sách dữ liệu vào **RecyclerView**, chúng ta cần phải thiết kế một **item** để đưa mỗi String vào một **View** con nằm trong **RecyclerView**. Mỗi **item** đều có một layout riêng, ở đây chúng ta sẽ sử dụng **CardView** để hiển thị dữ liệu.

+ Khi sử dụng **RecyclerView**, chúng ta cần tạo ra một **Adapter** để ánh xạ **item view** vừa tạo và **item** trong danh sách đầu vào.

➤ Xử lý chức năng:

+ Việc gọi API sẽ được thông qua việc thực thi những **request** mà mình gửi đi, để làm được điều này cần sử dụng thư viện **Volley**

```
implementation 'com.android.volley:volley:1.2.1'
```

+ Import các thư viện cần thiết để sử dụng **RecyclerView** và **CardView**:

```
// RecyclerView
implementation "androidx.recyclerview:recyclerview:1.2.1"
// Cardview
implementation "androidx.cardview:cardview:1.0.0"
```

+ Sử dụng API **Current Weather Data** (<https://openweathermap.org/current>) để gọi về ứng dụng và lấy ra **lat**, **lon** và tên thành phố.

+ Tiếp theo, sử dụng **Historical WeatherAPI** (<https://openweathermap.org/history>) để gọi dữ liệu từ API này về và hiển thị ra màn hình. Dữ liệu sẽ được hiển thị dưới dạng danh sách nằm dọc ở dạng **CardView**.

➤ **Kết quả thực hiện:**



Hình 4.2.5.1 Lịch sử thời tiết theo ngày _ 1



Hình 4.2.5.2 Lịch sử thời tiết theo ngày _ 2

4.2.6 Xem lịch sử thời tiết trong vòng 30 ngày, tính trên mỗi giờ

➤ Ý tưởng chung:

- Tạo activity có tên là **DetailHistoryWeather**, theo đó file xml tương ứng sẽ được tạo ra với tên là **activity_details_history_weather**
- Tiến hành thiết kế giao diện trên **activity_details_history_weather**

+ Sử dụng **RecyclerView** để hiển thị dữ liệu theo dưới dạng danh sách ngang hoặc đứng **LinearLayoutManager**

+ Để hiển thị được danh sách dữ liệu vào **RecyclerView**, chúng ta cần phải thiết kế một **item** để đưa mỗi String vào một **View** con nằm trong **RecyclerView**. Mỗi **item** đều có một layout riêng, ở đây chúng ta sẽ sử dụng **CardView** để hiển thị dữ liệu.

+ Khi sử dụng **RecyclerView**, chúng ta cần tạo ra một **Adapter** để ánh xạ **item view** vừa tạo và **item** trong danh sách đầu vào.

➤ Xử lý chức năng:

+ Việc gọi API sẽ được thông qua việc thực thi những **request** mà mình gửi đi, để làm được điều này cần sử dụng thư viện **Volley**

```
implementation 'com.android.volley:volley:1.2.1'
```

+ Import các thư viện cần thiết để sử dụng **RecyclerView** và **CardView**:

```
// RecyclerView
implementation "androidx.recyclerview:recyclerview:1.2.1"
// Cardview
implementation "androidx.cardview:cardview:1.0.0"
```

+ Sử dụng API **Current Weather Data** (<https://openweathermap.org/current>) để gọi về ứng dụng và lấy ra **lat**, **lon** và tên thành phố.

+ Tiếp theo, sử dụng **Historical WeatherAPI** (<https://openweathermap.org/history>) để gọi dữ liệu từ API này về và hiển thị ra màn hình. Dữ liệu sẽ được hiển thị dưới dạng danh sách nằm dọc ở dạng **CardView**.

➤ **Kết quả thực hiện:**

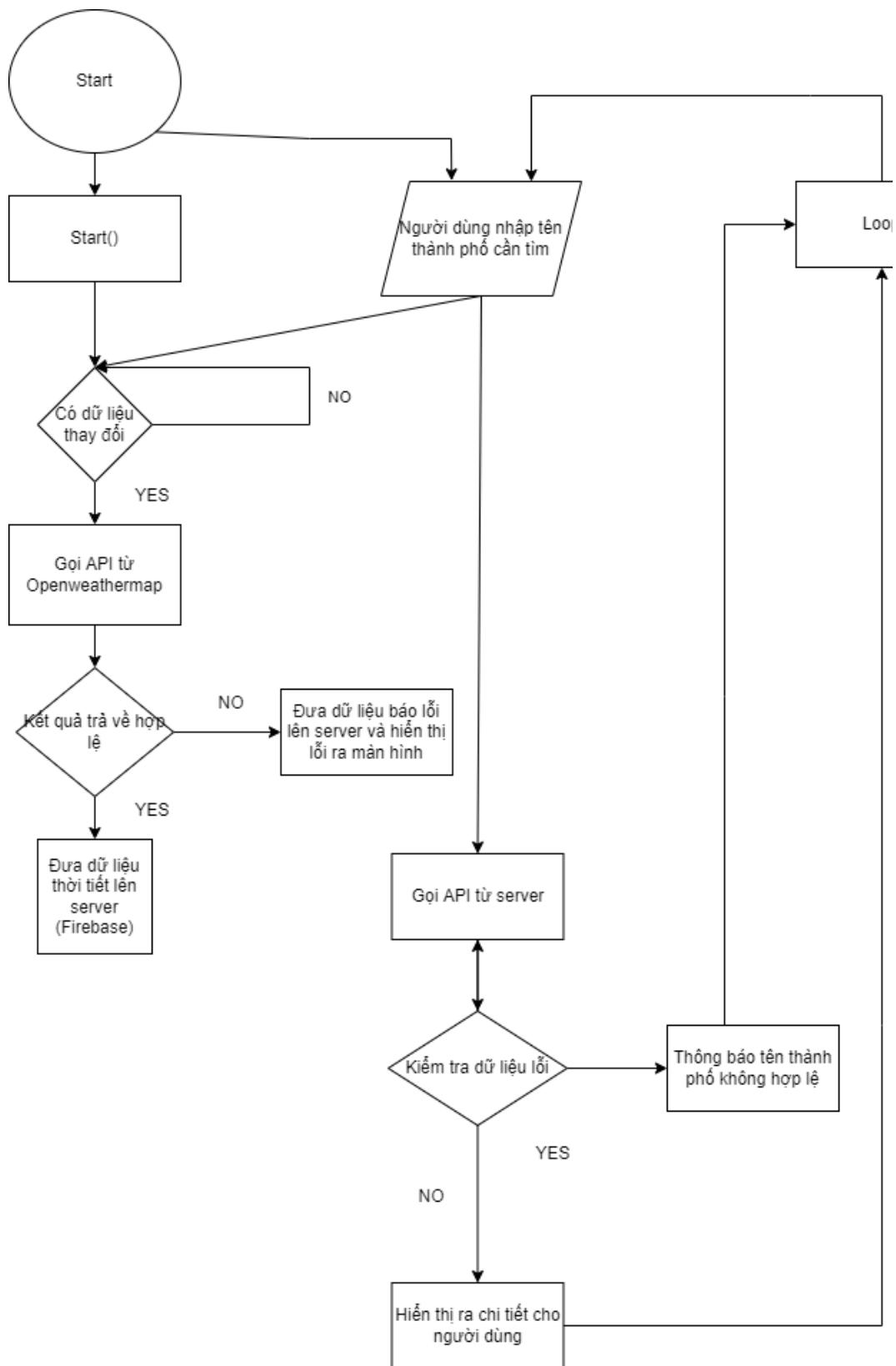


Hình 4.2.6.1 Lịch sử thời tiết trong vòng 30 ngày, tính trên mỗi giờ _ 1



Hình 4.2.6.2 Lịch sử thời tiết trong vòng 30 ngày, tính trên mỗi giờ _ 2

4.3 Sơ đồ khối tổng thể của ứng dụng thời tiết



Hình 4.2.6.1 Sơ đồ khối tổng thể của ứng dụng thời tiết

Trong sơ đồ này:

- Người dùng tiến hành khởi động ứng dụng lên.
- Sau đó nhập tỉnh/ thành phố cần tìm.
- Dữ liệu được nhập sẽ được kiểm tra xem có khớp với dữ liệu từ API của Openweathermap hay không.
- Nếu dữ liệu đúng, trả về kết quả ra màn hình; đồng thời đưa lên server (Firebase) để lưu trữ. Ngược lại, nếu dữ liệu không khớp, thông báo lỗi ra màn hình.
- Sau khi lưu trữ lên server, tiến hành gọi dữ liệu từ server về và hiển thị ra màn hình.
- Nếu trả đúng dữ liệu, in kết quả nhận được ra màn hình ứng dụng và thực hiện các chức năng cần thiết của ứng dụng đó. Ngược lại, nếu sai thì thông báo cho người dùng biết là tên thành phố không hợp lệ và yêu cầu người dùng nhập lại dữ liệu đúng.

CHƯƠNG 5: KẾT LUẬN

5.1 Kết quả đạt được của đề tài

Sau một thời gian tìm hiểu đề tài “*Xây dựng ứng dụng mobile xem thời tiết và chất lượng không khí sử dụng Openweathermap.org APIs*” 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:

- ✓ Xem được thời tiết địa điểm hiện tại của thiết bị.
- ✓ Cho phép người dùng lựa chọn địa điểm (tỉnh, thành phố) để xem thời tiết nơi đó.
- ✓ Dự báo thời tiết trong 7 ngày tới ở mỗi ngày.
- ✓ Dự báo thời tiết trong 2 ngày tới ở mỗi giờ.
- ✓ Xem chất lượng không khí với tỉnh, thành phố tương ứng.
- ✓ Xem lịch sử nhiệt độ và mưa/nắng trong vòng 30 ngày, dữ liệu được tính trên mỗi giờ.

5.2 Hạn chế của đề tài

- Phần lấy vị trí của người dùng khi đồ dữ liệu về ứng dụng còn bị chậm.
- Về phần chất lượng không khí, khá ổn nhưng chưa hiển thị chi tiết các khí trong các ngày tiếp theo.

5.3 Hướng phát triển

Hoàn thiện và bổ sung thêm các tính năng mới:

- Cập nhật thời tiết theo thời gian thực

- Chỉnh sửa lại ứng dụng để người dùng có thể xem chất lượng không khí trong ngày theo từng ngày và lịch sử của chất lượng không khí khi cần.

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] <https://genk.vn/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
- [6] <https://www.stdio.vn/android/event-listeners-trong-android-k117fa>
- [7] <https://laptrinhstuduy.wordpress.com/2014/04/27/activity-trong-android/>
- [8] https://en.wikipedia.org/wiki/Air_quality_index#CAQI