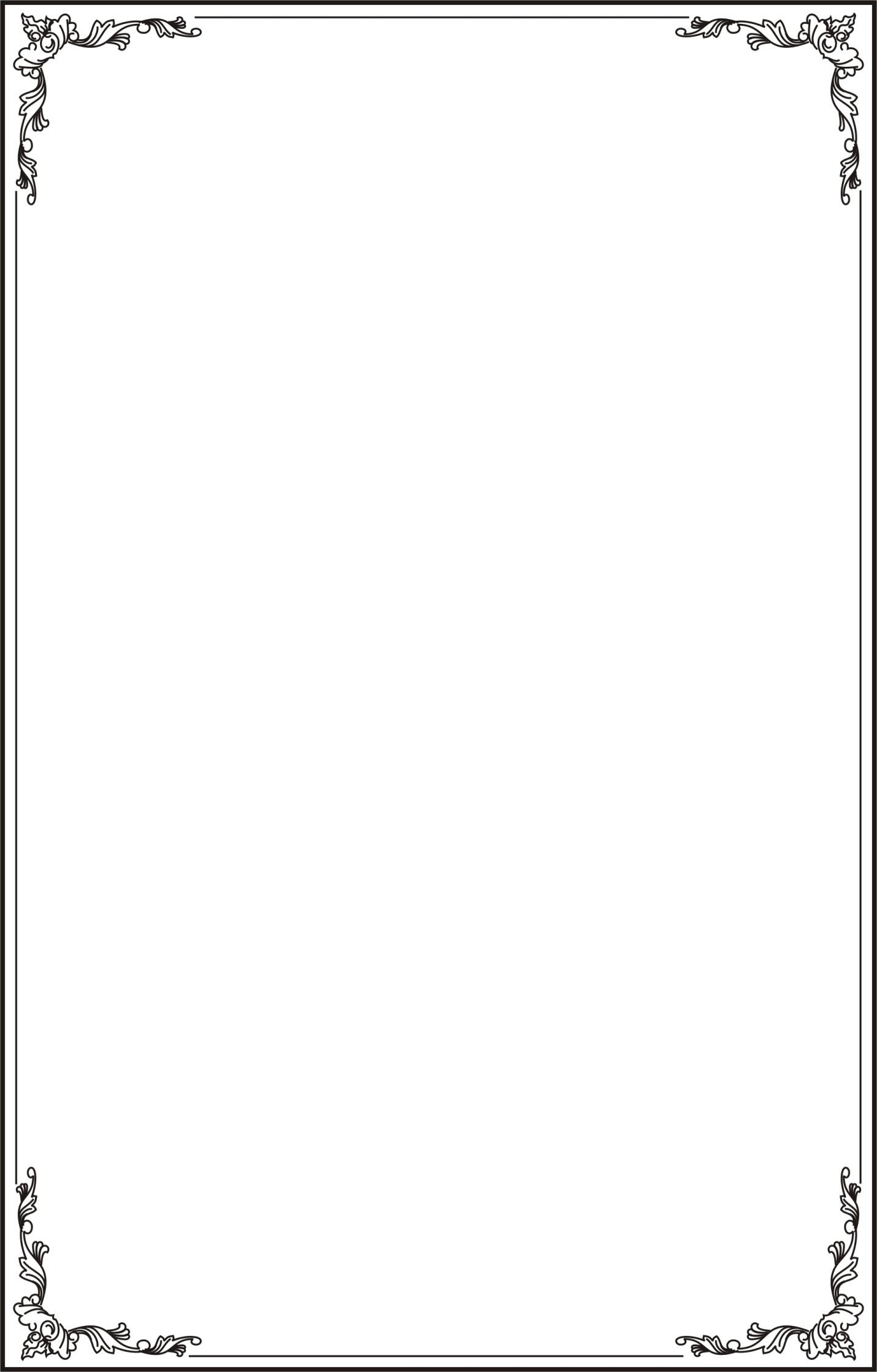
****

**Bộ Công Thương**

**Trường Đại Học Công Nghiệp Thực Phẩm TP.HCM**

**🙡🙡⚫🙣🙣**

**Môn học: Lập Trình Di Động**

**Đề Tài: Quản lý địa điểm kết hợp Google Map**

**GVHD: Th.s Vũ Đình Ái**

**Danh sách nhóm 12:**

1. **Phạm Văn Hiệp 2001160224**
2. **Nguyễn Huỳnh Bá Huy 2001160250**
3. **Nguyễn Ngọc Tuấn Khôi 2001160259**

***-* TP.HCM, 11/2019 *-***

**Bộ Công Thương**

**Trường Đại Học Công Nghiệp Thực Phẩm TP.HCM**

**🙡🙡⚫🙣🙣**

**Môn học: Lập Trình Di Động**

**Đề Tài: Quản lý địa điểm kết hợp Google Map**

**GVHD: Th.s Vũ Đình Ái**

**Danh sách nhóm 12:**

1. **Phạm Văn Hiệp 2001160224**
2. **Nguyễn Huỳnh Bá Huy 2001160250**
3. **Nguyễn Ngọc Tuấn Khôi 2001160259**

***-* TP.HCM, 11/2019 *-***

# LỜI MỞ ĐẦU

Như chúng ta đều biết, ngày nay khi ngành công nghệ thông tin đang phát triển một cách mạnh mẽ. Các sản phẩm công nghệ cùng với các ứng dụng, phần mềm hỗ trợ ngày càng phổ biến và được áp dụng tất cả các lĩnh vực. Chúng đã chở thành những công cụ đắc lực hỗ trợ cho các công việc của con người trở nên thận tiện, dễ dàng, nhanh chóng và chính xác hơn rất nhiều. Việc tìm kiếm, tra cứu thông tin cũng trở nên đơn giản, nhanh chóng hơn nhờ các công cụ hỗ trợ như Google, Bing, Yahoo... Thì chúng ta không thể không nhắc tới Google Maps ứng dụng cho phép tìm kiếm địa điểm phổ biến và mạnh mẽ nhất hiện này.

Google Map là một bản đồ vệ tinh được Google phát triển với mục đích thay thế bản đồ giấy truyền thống. Giờ đây chỉ cần một chiến smartphone đã cài đặt Google Map và được kết nối internet, bạn có thể đi khắp mọi nơi mà không sợ bị lạc đường cùng không cần phải hỏi han bất kỳ ai cả, đó chính là sự tiện lợi mà Google Map đang mang lại.

Bên cạnh những tính năng nổi bật của Google Map như là:

* Lưu địa chỉ nhà và nơi làm việc.
* Chia sẻ bản đồ, chia sẻ vị trí hiện tại của bản thân.
* Tìm kiếm đường đi tốt nhất để đến một địa điểm một cách nhanh chóng.
* Giao diện thân thiện, dễ sử dụng, hiển thị các tuyến đường một các trực quan.
* Khả năng nhận diện giọng nói tốt.
* Tích hợp Google Map vào website.
* Đánh dấu vị trí của doanh nghiệp trên Google Map.

Chính vì những tính năng nổi bật trên nên chúng em chọn đề tài xây dựng ứng dụng quản lý địa điểm kết hợp Google Map để có thể giúc cho người sử dụng có thể dễ dàng quản lý các địa điểm, tìm kiếm thông tin của các địa điểm, đồng thời tìm được đường đi ngắn nhất từ vị trí của người dùng đến địa điểm được chọn.

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc26342257)

[Chương 1: Cơ Sở Lý Thuyết 5](#_Toc26342258)

[**I.** **Lịch sử của hệ điều hành Android** 5](#_Toc26342259)

[**I.** **Khái niệm hệ điều hành Android** 5](#_Toc26342260)

[**II.** **Sự ra đời của hệ điều hành Android** 6](#_Toc26342261)

[**III.** **Các phiên bản Android** 7](#_Toc26342262)

[**II.** **Các thành phần cấu thành nên một ứng dụng Android** 11](#_Toc26342263)

[1. Activity 11](#_Toc26342264)

[2. Intent 14](#_Toc26342265)

[3. Service 16](#_Toc26342266)

[4. Content Provider 19](#_Toc26342267)

[5. View 20](#_Toc26342268)

[Chương 2: Triển Khai Đồ Án 22](#_Toc26342269)

[**I.** **Cách lấy dữ liệu về địa điểm từ trang Foody** 22](#_Toc26342270)

[II. **Google Maps API key là gì và cách lấy** 27](#_Toc26342271)

[Google Maps API là gì? 27](#_Toc26342272)

[Cách lấy Google Maps API key? 27](#_Toc26342273)

[**III.** **AsyncTask** 36](#_Toc26342274)

[**1)** **Khái niệm** 36](#_Toc26342275)

[**2)** **Cách triển khai AsyncTask trong Android** 36](#_Toc26342276)

[Chương 3: Demo và kết luận 38](#_Toc26342277)

[**I.** **Giao diện và chức năng của giao diện** 38](#_Toc26342278)

[II. Kết Luận 46](#_Toc26342279)

[Tài Liệu Tham Khảo 47](#_Toc26342280)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Logo Android 5](#_Toc26340518)

[Hình 2: Các phiên bản Android 7](file:///C:\Users\Asus\Downloads\LTDD_UngDungKetHopGoogleMaps.docx#_Toc26340519)

[Hình 3:Lược đồ vòng đời của 1 activity 13](#_Toc26340520)

[Hình 4:Sử dụng Intent để trao đổi thông tin giữ hai chương trình 15](#_Toc26340521)

[Hình 5: Lược đồ vòng đời của 1 Service 18](#_Toc26340522)

[Hình 6: Ví dụ mô hình dữ iệu thông tin số điện thoại trong máy 20](file:///C:\Users\Asus\Downloads\LTDD_UngDungKetHopGoogleMaps.docx#_Toc26340523)

[Hình 7: Chọn loại địa điểm mà ta muốn lấy dữ liệu 22](#_Toc26340524)

[Hình 8: Tab Netword 23](#_Toc26340525)

[Hình 9: Chọn địa điểm muốn lấy dữ liệu 23](#_Toc26340526)

[Hình 10: Lấy dữ liệu dạng JSON 24](#_Toc26340527)

[Hình 11: Chuyển dữ liệu sang trang https://jsoneditoronline.org/ 25](#_Toc26340528)

[Hình 12: Chuyển đổi dữ liệu 25](#_Toc26340529)

[Hình 13: Chọn lọc dữ liệu muốn lấy 26](#_Toc26340530)

[Hình 14: Giao diện trang quản lý các nền tảng đang đuợc cung cấp bởi Google 28](#_Toc26340531)

[Hình 15: Tạo Project mới trong Google Cloud Platform 29](#_Toc26340532)

[Hình 16: Tạo project mới hoặc chọn project có sẵn 29](#_Toc26340533)

[Hình 17: Khi bạn chưa đặt tên sẽ có 1 tên tự tạo sẵn như hình trên 30](#_Toc26340534)

[Hình 18: Chọn menu APIs & Services và chọn Library 30](#_Toc26340535)

[Hình 19: Thanh công cụ tìm kiếm Google Cloud Platform 31](#_Toc26340536)

[Hình 20: Chọn vào Maps SDK for Android 31](#_Toc26340537)

[Hình 21: click vào nút ENABLE để bật thư viện Google Maps API lên 32](#_Toc26340538)

[Hình 22: Chọn vào tab Credentials 32](#_Toc26340539)

[Hình 23: Click chọn vào nút Create credentials 33](#_Toc26340540)

[Hình 24: Click chọn vào <tên API key> 34](#_Toc26340541)

[Hình 25: Chọn vào lựa chọn Android apps 34](#_Toc26340542)

[Hình 26: Chọn vào ADD AN ITEM 35](#_Toc26340543)

[Hình 27: Nhập vào tên Package project của bạn và chuỗi SHA-1 35](#_Toc26340544)

[Hình 28: Các bước sử dụng Asynctask 36](file:///C:\Users\Asus\Downloads\LTDD_UngDungKetHopGoogleMaps.docx#_Toc26340545)

[Hình 29: Giao diện loading 38](#_Toc26340546)

[Hình 30: Giao diện chính 39](file:///C:\Users\Asus\Downloads\LTDD_UngDungKetHopGoogleMaps.docx#_Toc26340547)

[Hình 31: Thanh tìm kiếm 40](file:///C:\Users\Asus\Downloads\LTDD_UngDungKetHopGoogleMaps.docx#_Toc26340548)

[Hình 32: Thanh Scroll View 41](file:///C:\Users\Asus\Downloads\LTDD_UngDungKetHopGoogleMaps.docx#_Toc26340549)

[Hình 33: Danh sách địa điểm lọc theo loại 42](#_Toc26340550)

[Hình 34: Vị trí của địa điểm được chọn 43](#_Toc26340551)

[Hình 35: Tìm đường từ vị trí hiện tại đến địa điểm 44](#_Toc26340552)

[Hình 36: Lịch sử tìm kiếm 45](#_Toc26340553)

**DANH MỤC VIẾT TẮT**

|  |  |
| --- | --- |
| **GPS** | **Global Positioning System**: là hệ thống xác định vị trí dựa trên vị trí của các [vệ tinh nhân tạo](https://vi.wikipedia.org/wiki/V%E1%BB%87_tinh) |
| **CDMA** | **Code Division Multiple Access**: nghĩa là đa truy nhập (đa người dùng) phân chia theo [mã](https://vi.wikipedia.org/w/index.php?title=M%C3%A3&action=edit&redlink=1). |
| **HTTP** | **HyperText Markup Language**:  là một [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) được thiết kế ra để tạo nên các [trang web](https://vi.wikipedia.org/wiki/Website) |
| **API** | **Application Programming Interface**:  là một [giao diện](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng |

# Chương 1: Cơ Sở Lý Thuyết

1. **Lịch sử của hệ điều hành Android**
2. **Khái niệm 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 được chính Google mua lại vào năm 2005.

[](http://genknews.genkcdn.vn/k:2015/android-1-0-1433777469803/lich-su-va-chang-duong-phat-trien-cua-android-qua-cac-phien-ban-phan-1-thoi-ky-dau-cua-android.png)

Hình 1: Logo Android

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.

Cho đến hiện tại, Google đã bán ra 200.000 thiết bị Android mỗi ngày và Play Store của họ vượt mức 90.000 ứng dụng. 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.

1. **Sự ra đời của hệ điều hành Android**

Vào tháng 10/2003, trước khi thuật ngữ “điện thoại thông minh” được hầu hết công chúng sử dụng và vài năm trước khi Apple công bố iPhone đầu tiên và hệ điều hành iOS, công ty Android Inc được thành lập ở Palo Alto, California. Bốn người sáng lập là Rich Miner, Nick Sears, Chris White và Andy Rubin. Vào thời điểm thành lập, ông Rubin nói rằng Android Inc sẽ phát triển “thiết bị di động thông minh hơn về vị trí và sở thích của chủ sở hữu”.

Năm 2005, chương lớn tiếp theo trong [lịch sử](https://viettimes.vn/tags/IGzhu4tjaCBz4but/lich-su.html) của Android được thực hiện khi Google mua lại công ty gốc. Ông Andy Rubin và các thành viên sáng lập khác vẫn tiếp tục phát triển hệ điều hành dưới quyền chủ sở hữu mới của họ. Quyết định này được đưa ra để sử dụng Linux làm nền tảng cho hệ điều hành Android và điều đó cũng có nghĩa là Android sẽ được cung cấp miễn phí cho các nhà sản xuất điện thoại di động của bên thứ ba. Google và nhóm Android cảm thấy công ty có thể kiếm tiền với các dịch vụ khác sử dụng hệ điều hành, bao gồm cả ứng dụng.

Hệ điều hành Android được chính thức ra mắt từ năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở. Chiếc điện thoại Android đầu tiên được bán vào năm 2008.

1. **Các phiên bản Android**

*[](http://genknews.genkcdn.vn/k:2015/android-versions-by-jared-rummler-with-border-1433777517012/lich-su-va-chang-duong-phat-trien-cua-android-qua-cac-phien-ban-phan-1-thoi-ky-dau-cua-android.png)*

Hình 2: Các phiên bản Android

* Android 1.5 Cupcake

Tên công khai chính thức đầu tiên của Android đã không xuất hiện cho đến phiên bản 1.5 Cupcake vào tháng 4/2009. Nó đã bổ sung khá nhiều tính năng và cải tiến mới so với hai phiên bản công khai đầu tiên, như khả năng tải video lên YouTube, cách hiển thị màn hình của điện thoại tự động xoay sang đúng vị trí và hỗ trợ bàn phím của bên thứ ba.

* Android 1.6 Donut

Google đã nhanh chóng tung ra Android 1.6 Donut trong tháng 9/2009. Một số tính năng mới hỗ trợ cho các nhà mạng sử dụng mạng CDMA. Điều này giúp đưa điện thoại Android được bán trên toàn thế giới.

Các tính năng khác bao gồm việc giới thiệu Quick Search Box và nhanh chóng chuyển đổi giữa Máy ảnh, Máy quay và Thư viện để sắp xếp trải nghiệm chụp media. Donut cũng đã giới thiệu tiện ích Power Control để quản lý Wifi, Bluetooth, GPS,…

* Android 2.0-2.1 Eclair

Vào tháng 10/2009, khoảng một năm sau khi tung ra Android 1.0, Google đã phát hành phiên bản 2.0 của hệ điều hành với tên mã chính thức là Eclair. Phiên bản này là phiên bản đầu tiên bổ sung hỗ trợ Text-to-Speech và cũng giới thiệu các hình nền động, hỗ trợ nhiều tài khoản và điều hướng của Google Maps trong số nhiều tính năng và cải tiến mới khác.

* Android 2.2 Froyo

Ra mắt vào tháng 5/2010, [Android 2.2 Froyo](https://viettimes.vn/tags/IEFuZHJvaWQgMi4yIEZyb3lv/android-22-froyo.html) (yogurt đông lạnh) đã chính thức ra mắt. Điện thoại thông minh có cài đặt Froyo có thể tận dụng một số tính năng mới, bao gồm chức năng điểm phát sóng di động Wi-Fi, đẩy thông báo qua dịch vụ C2DM của Android Cloud, hỗ trợ flash và nhiều hơn thế nữa.

* Android 2.3 Gingerbread

Android 2.3 Gingerbread, được tung ra vào tháng 9/2010, hiện là phiên bản cũ nhất của hệ điều hành mà Google vẫn đang liệt kê trong trang cập nhật phiên bản nền tảng hàng tháng. Tính đến ngày 13/9/2017, Google chỉ ra rằng chỉ 0,6 phần trăm của tất cả các thiết bị Android hiện đang chạy một số phiên bản của [Gingerbread](https://viettimes.vn/tags/IEdpbmdlcmJyZWFk/gingerbread.html).

* Android 3.0 Honeycomb

Honeycomb được Google phát hành chỉ để cài đặt trên máy tính bảng và các thiết bị di động khác có màn hình lớn hơn so với điện thoại thông minh hiện tại. Honeycomb lần đầu tiên được giới thiệu vào tháng 2/2011, cùng với máy tính bảng Motolola Xoom đầu tiên và bao gồm các tính năng như giao diện người dùng được thiết kế lại dành riêng cho màn hình lớn cùng với thanh thông báo được đặt ở phía dưới màn hình của máy tính bảng.

* Android 4.0 Ice Cream Sandwich

Được phát hành vào tháng 10/2011, phiên bản Ice Cream Sandwich của Android đã mang một số tính năng mới cho người dùng. Nó kết hợp nhiều tính năng của phiên bản Honeycomb dành cho máy tính bảng với Gingerbread theo định hướng smartphone. Nó cũng bao gồm một “khay yêu thích” trên màn hình chủ, cùng với sự hỗ trợ đầu tiên để mở khóa điện thoại bằng cách chụp ảnh khuôn mặt của chủ sở hữu. Loại hỗ trợ đăng nhập sinh trắc học đã phát triển và cải thiện đáng kể kể từ đó.

* Android 4.1Jelly Bean

Kỷ nguyên Jelly Bean của Android bắt đầu vào tháng 6/2012 với việc phát hành Android 4.1. Một số tính năng mới bao gồm thông báo mới cho thấy nhiều nút nội dung cùng với sự hỗ trợ đầy đủ cho phiên bản trình duyệt web Chrome của Android. Google Now cũng đã xuất hiện trên Search và “Project Butter” được giới thiệu để tăng tốc độ hoạt hình và cải thiện khả năng phản ứng cảm ứng của Android. Màn hình, chụp ảnh HDR và Miracast cũng được hỗ trợ.

* Android 4.4 KitKat

KitKat không có nhiều các tính năng mới, nhưng nó đã có một điều mà thực sự đã giúp mở rộng thị trường Android tổng thể. Nó đã được tối ưu hóa để chạy trên điện thoại thông minh có ít nhất 512 MB RAM. Điều này cho phép các nhà sản xuất điện thoại để có được phiên bản mới nhất của Android và cài đặt nó trên điện thoại rẻ hơn nhiều.

* Android 5.0 Lollipop

Được ra mắt lần đầu vào mùa thu năm 2014, Android 5.0 Lollipop là một bước thay đổi lớn trong giao diện tổng thể của hệ điều hành. Đây là phiên bản đầu tiên của hệ điều hành sử dụng ngôn ngữ Material Design của Google, sử dụng các hiệu ứng ánh sáng và bóng tối tự do. UI cũng có một số thay đổi khác cho Lollipop, bao gồm một thanh điều hướng mới, thông báo phong phú cho màn hình khóa và nhiều hơn nữa.

Bản cập nhật Android 5.1 đã có một số thay đổi. Các tính năng này bao gồm hỗ trợ chính thức cho các cuộc gọi thoại đôi, cuộc gọi thoại HD và bảo vệ thiết bị ngay cả sau khi khôi phục cài đặt gốc.

* Android 6.0 Marshmallow

Android 6.0 Marshmallow được phát hành vào mùa thu năm 2015. Nó bao gồm các tính năng như một ngăn kéo ứng dụng di chuyển theo chiều dọc, Google Now trên Tap, hỗ trợ mở khóa sinh trắc học, hỗ trợ USB Type-C, Android Pay, và nhiều hơn nữa.

* Android 7.0 Nougat

Phiên bản 7.0 của hệ điều hành di động của Google được ra mắt vào mùa thu năm 2016. Trước khi Nougat được tiết lộ “Android N” đã được Google gọi nội bộ là “New York Cheesecake”. Một số tính năng mới của Nougat bao gồm các chức năng đa tác vụ tốt hơn cho số lượng ngày càng tăng của điện thoại thông minh có màn hình lớn, chẳng hạn như chế độ chia màn hình cùng với việc chuyển đổi nhanh giữa các ứng dụng.

* Android 8.0 Oreo

Vào tháng 3/2017, Google chính thức công bố và phát hành bản xem trước dành cho nhà phát triển đầu tiên dành cho Android O, còn được gọi là Android 8.0.Đây là lần thứ hai Google chọn một tên thương hiệu cho Android (Oreo thuộc sở hữu của Nabisco). Android Oreo bao gồm nhiều thay đổi trực quan trong menu Cài đặt, cùng với hỗ trợ chế độ ảnh-trong-ảnh, các kênh thông báo, API tự động điền mới để quản lý mật khẩu và điền dữ liệu tốt hơn.

* Android P

Google đã giới thiệu bản xem trước dành cho nhà phát triển đầu tiên của bản cập nhật [Android](https://viettimes.vn/tags/QW5kcm9pZA==/android.html) chính tiếp theo vào ngày 7/3. Hiện tại, nó chỉ được gọi là "Android P" và phiên bản xem trước đầu tiên chỉ hoạt động trên điện thoại thông minh Pixel của Google. Một số tính năng trong bản phát hành xem trước đầu tiên bao gồm hỗ trợ định dạng video HDR và HEIF, điều hướng trong Google Maps, cải tiến về bảo mật và hỗ trợ chính thức cho máy ảnh và cảm biến “tai thỏ”.

1. **Các thành phần cấu thành nên một ứng dụng Android**

### ****Activity****

**Định nghĩa Activity**

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

Có 2 phương thức mà gần như mọi lớp con của Activity đều phải hiện thực:

* **onCreate(Bundle)** - Nơi khởi tạo activity. Quan trọng hơn, đây chính người lập trình gọi setContentView(int) kèm theo layout để thể hiện UI của riêng mình. Đồng thời còn có findViewById(int) giúp gọi các widget (buttons, text boxes, labels,..) để dùng trong UI.
* **onPause()** - Nơi giải quyết sự kiện người dùng rời khỏi activity. Mọi dữ liệu được người dùng tạo ra tới thời điểm này cần phải được lưu vào ContentProvider.

**Vòng đời của một activity**

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

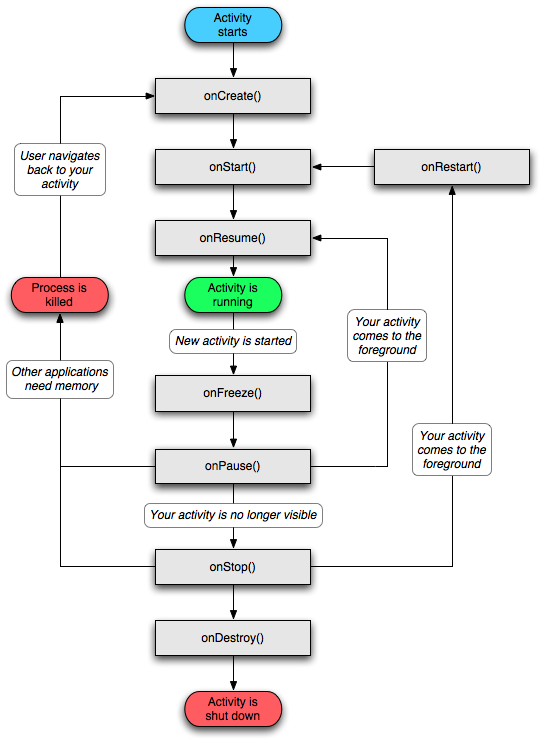
Một Activity có 4 trạng thái:

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

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

**Stopped:** Khi 1 activity bị che khuất hoàn toàn bởi 1 activity khác. Các stopped activity đa phần bị hệ thống bắt chấm dứt để dành chỗ cho các tiến trình khác.

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



Hình 3:Lược đồ vòng đời của 1 activity

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

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

**Vòng đời thấy được (Visible Lifetime):** Diễn ra từ khi gọi onStart() và kéo dài tới khi gọi onStop(). Ở vòng đời này, activity được hiển thị trên màn hinh nhưng có thế không tương tác với người dùng ở trên nền. Các phương thức onStart(0 và onStop() có thể được gọi nhiều lần.

**Vòng đời trên nền (Foreground Lifetime):** Diễn ra từ khi gọi onResume (0 và kéo dài tới khi gọi onPause(). Ở vòng đời này, activity nằm trên mọi activity khác và tương tác được với người dùng. 1 activity có thể liên tục thay đổi giữa 2 trạng thái paused và resumed, chẳng hạn khi thiết bị sleep hay 1 intent mới được đưa tới.

### ****Intent****

**Định nghĩa Intent:**

**Intent** là một mô tả trừu tượng của một hành động được thực thi. Nó đại diện cho một hành động đi kèm với một ngữ cảnh xác định. Với **Intent** thì có nhiều hành động và nhiều component (Một thể hiện của một class java dùng để thực thi các hành động được đặc tả trong **Intent**) dành cho **Intent** của Android hơn là so với HTTP verbs (POST, GET) và nguồn tài nguyên (hình ảnh, web page) của giao thức HTTP, tuy nhiên khái niệm vẫn tương tự nhau.

**Intent** được sử dụng với phương thức **startActivity()** để mở một **Activity**, và dùng với **broadcastIntent** để gởi nó đến bất kì **BroadcastReceiver** liên quan nào, và dùng với [**startService(Intent)**](http://developer.android.com/reference/android/content/Context.html#startService%28android.content.Intent%29)**,** **bindService(Intent, ServiceConnection, int)** để giao tiếp với các **Service** chạy dưới nền.

Intent cung cấp một chức năng cho phép kết nối hai chương trình khác nhau trong quá trình thực thi (runtime) (Cung cấp khả năng cho phép hai chương trình khác nhau giao tiếp với nhau). Chức năng quan trọng và được sử dụng nhiều nhất của một **Intent**  là mở một **Activity**, nơi mà nó có thểđược dùng như một vật kết nối các Activity lại với nhau (Truyền thông tin giữa hai Activity khác nhau).

[](https://expressmagazine.net/sites/default/files/imageDevelopment/android_1288.png)

Hình 4:Sử dụng Intent để trao đổi thông tin giữ hai chương trình

Thành phần chính của Intent bao gồm:

**Action:** Xác định hành động sẽ được thực thi, các hành động này có thể là: ACTION\_VIEW, ACTION\_EDIT, ACTION\_MAIN…

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

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

**Category:** thông tin chi tiết về hành động được thực thi, ví dụ như **CATEGORY\_LAUNCHER** có nghĩa là nó sẽ xuất hiện trong Launcher như ứng dụng có mức level cao (top-level application),  trong khi **CATEGORY\_ALTERNATIVE** chứa thông tin danh sách các hành động thay thế mà người dùng có thể thực thi trên phần dữ liệu mà **Intent** cung cấp.

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

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

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

### ****Service****

**Định nghĩa Service**

Một service là một thành phần của ứng dụng, thể hiện mong muốn ứng  dụng thực hiện các hành động trong khi không tương tác với người dùng hoặc cung cấp chức năng cho các ứng dụng khác sử dụng. Nói một cách đơn giản, service là các tác vụ (task) chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó.  Mỗi class Service phải chứa thẻ <service> được khai báo trong file **AndroidManifext.xml**. Services có thể được bắt đầu bởi Context.startService() và Context.bindservice()

Cũng như các đối tượng của ứng dụng khác, services chạy trên luồng (Thread) của tiến trình chính. Có nghĩa là nếu service của bạn dự định chạy các hành vi có cường độ lớn hoặc các hành vi có thể gây nghẽn mạch, nó sẽ tự sinh ra luồng (Thread) của chính nó để làm thực hiện các tác vụ được đặc tả trong Service.

**Cần lưu ý:**

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

- Service không phải là một luồng (thread).

- Service có chứa 2 đặc trưng cơ bản sau:

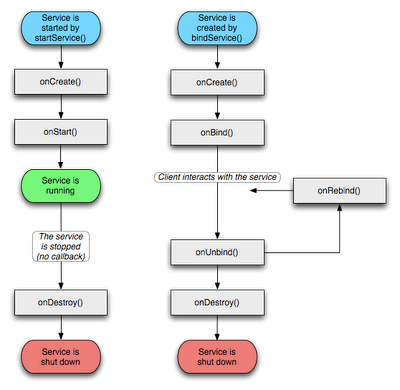
Là một chức năng để ứng dụng thông báo với hệ thống về việc mà nó muốn thực hiện ở phía dưới nền (ngay cả khi người dùng không tương tác trực tiếp tới ứng dụng). Điều này tương ứng với việc gọi phương thức Context.startservice(), nó sẽ yêu cầu hệ thống lên lịch cho service để chạy cho đến khi bị chính service hay người nào đó dừng nó lại.

Là một chức năng cho phép ứng dụng để cung cấp các chức năng của nó cho các ứng dụng khác. Điều đó tương ứng với việc gọi Context.bindService(), cho phép một long-standing connection được tạo ra để service có thể tương tác với nó.

Khi một Service được tạo ra, việc hệ thống cần làm là tạo ra thành phần và gọi hàm onCreate() và bất kỳ lệnh callback thích hợp nào trên luồng chình. Nó phụ thuộc vào Service thi hành hành vi thích hợp nào, chẳng hạn tạo luồng thứ hai để thực hiện tiến trình.

**Vòng đời của 1 Service**

Có 2 lý do để một service được chạy bởi hệ thống. Bằng cách gọi phương thức Context.startService() và Context.bindService(). Nếu người dùng gọi phương thức Context.startService() thì hệ thống sẽ nhận về một đối tượng **service** (tạo mới nó và gọi hàm onCreate() nếu cần thiết) và sau đó lại gọi phương thức onStartCommand(Intent, int ,int) với các biến được cung cấp bởi client. Service mới được tạo ra này sẽ chạy cho tới khi có lệnh gọi phương thức Context.stopService() hoặc stopSelf(). Dù cho một service có được start bao nhiêu lần đi nữa thì nó sẽ bị tắt ngay khi phương thức Context.stopService() hoặc phương thức stopSelf() được gọi.Service có thể dùng phương thức stopSelf(int) để đảm bảo rằng service sẽ không được tắt cho đến khi Intent đã khởi tạo nó được xử lý.



Hình 5: Lược đồ vòng đời của 1 Service

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

**BroadcastReceiver**

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

Có hai lớp broadcast quan trọng cần xét đến:

Broadcasts bình thường (Normal broadcasts): Được gửi bởi  context.sendBroadcast tới các receivers hoàn toàn không theo 1 trật tự nào, dẫn tới đôi khi tồn tại cùng 1 lúc nhiều receivers khác nhau. Tuy nhanh nhưng không thể trao đổi kết quả nhận về giữa cac1 receivers với nhau.

Broadcasts có thứ tự (Ordered broadcasts) - Được gửi bởi Context.sendOrderedBroadcast() tới từng receiver một. Như vậy, kết quả nhận về của 1 receiver có thể trở thành input của 1 receiver khác, hay có thể ngay lập tức dừng việc broadcast tới các receiver khác. Trật tự chạy các receivers được quyết định bởi thuộc tính android:priority trong intent-filter tương ứng.

### ****Content Provider****

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

Android cung cấp sẵn content providers cho 1 số kiểu dữ liệu thông dụng như âm thanh, video, thông tin sổ điện thoại,... Người lập trình cũng có thể tự tạo ra các class con (subclass) của Content Provider để lưu trữ kiểu dữ liệu của riêng mình.

**Cách thức hoạt động của Content Provider:**

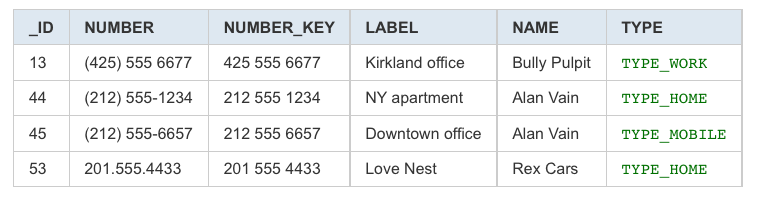
Mọi content provider, tuy có thể khác nhau về cách thức lưu trữ dữ liệu, đều sử dụng chung 1 interface để tìm và trả về dữ liệu, bao gồm việc thêm, bớt và sửa dữ liệu. Việc này được thực hiện thông qua các đối tượng ContentResolver khi gọi getContentResolver()

ContentResolver cr = getContentResolver();

**Mô hình dữ liệu**

Content providers sử dụng dữ liệu dưới dạng 1 table đơn giản, trong đó mỗi

hàng là 1 record và mỗi cột là dữ liệu của 1 kiểu nhất định.



Hình 6: Ví dụ mô hình dữ iệu thông tin số điện thoại trong máy

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

URI (Uniform Resource Identifier)

URI là dãy các ký tự string được dùng để:

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

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

### ****View****

View là thành phần cơ bản để xây dựng giao diện người dùng cho 1 ứng dụng [Android](https://expressmagazine.net/taxonomy/term/479). View là 1 lớp căn bản của widgets (widgets được dùng để tạo các nút nhấn, text fields,...).

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

* Một số Views thường dùng:

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

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

**Button:** Hiển thị 1 nút nhấn.

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

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

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

* Một số ViewGroups thường dùng:

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

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

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

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

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

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

# Chương 2: Triển Khai Đồ Án

1. **Cách lấy dữ liệu về địa điểm từ trang Foody**

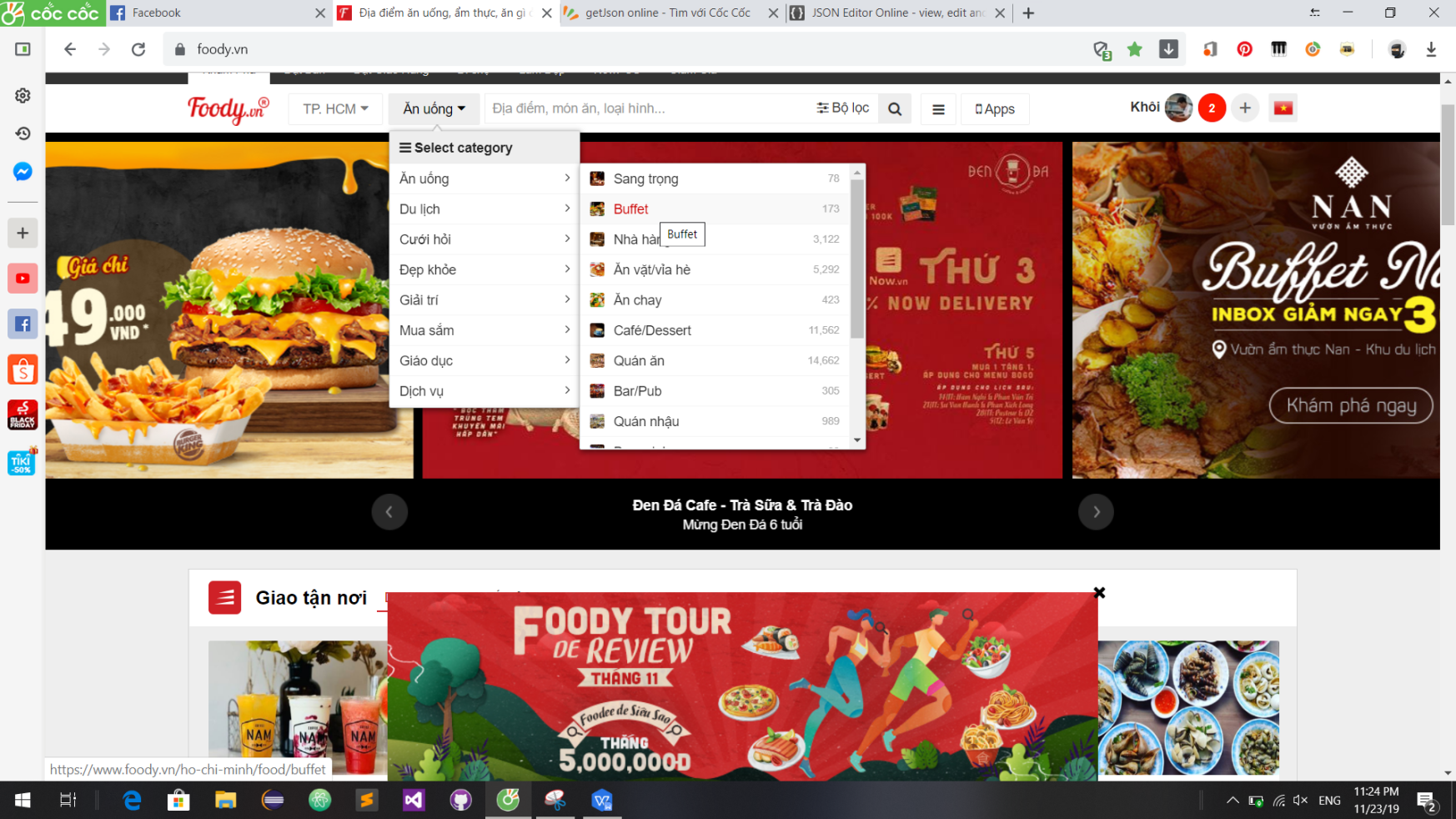
Dưới đây, nhóm sẽ giới thiệu 1 cách trong nhiều cách để lấy dữ liệu về các địa điểm của trang foody theo từng loại.

Hai trang web cần quan tâm trong suốt quá trình này là

1. <https://www.foody.vn/>
2. <https://jsoneditoronline.org/>

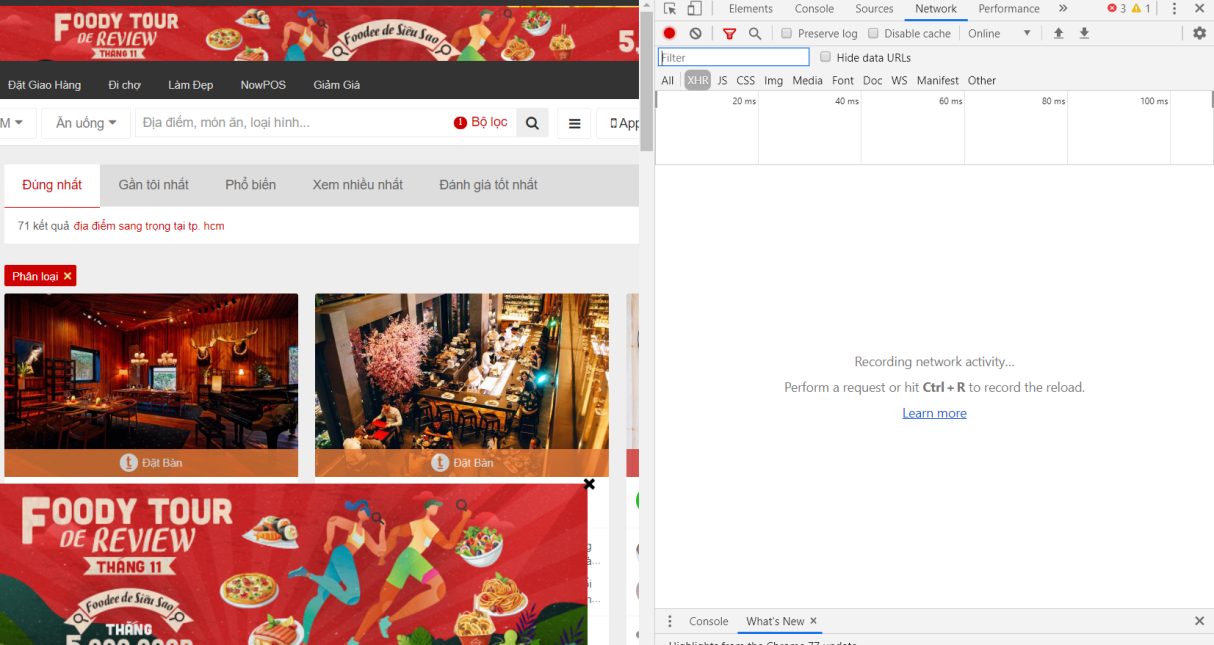
Dữ liệu ta lấy được từ trang web thứ 1 sẽ là dạng json ta sẽ dùng hỗ trợ của trang web thứ 2 để chuyển về dạng khác (mà ở đây là mảng đối tượng) để dễ lưu trữ và “làm sạch” dữ liệu.

**Bước 1:** Chọn loại địa điểm mà ta muốn lấy dữ liệu



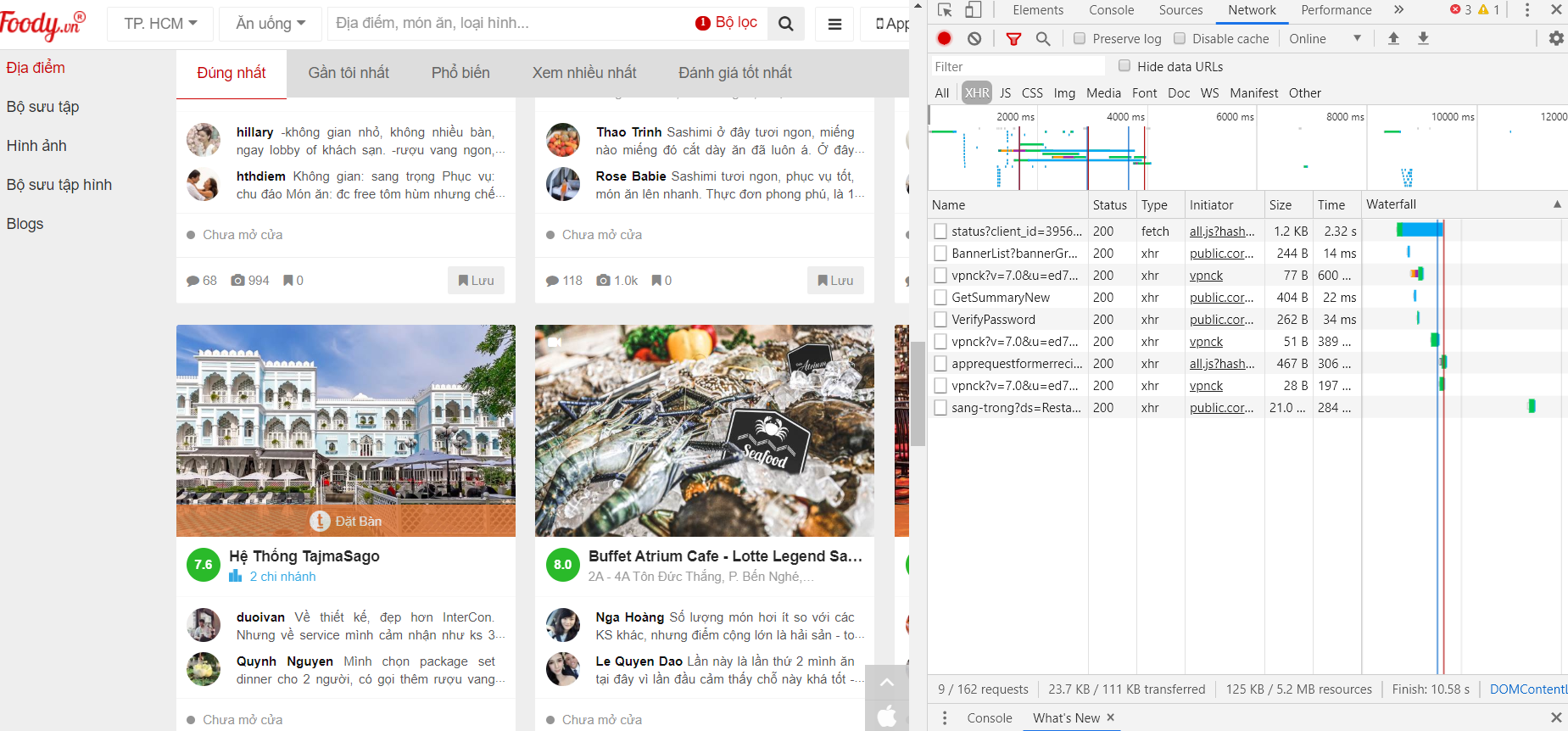
Hình 7: Chọn loại địa điểm mà ta muốn lấy dữ liệu

**Bước 2:** Bấm vào Kiểm tra để xem nguồn trang và chuyển sang tab Network



Hình 8: Tab Netword

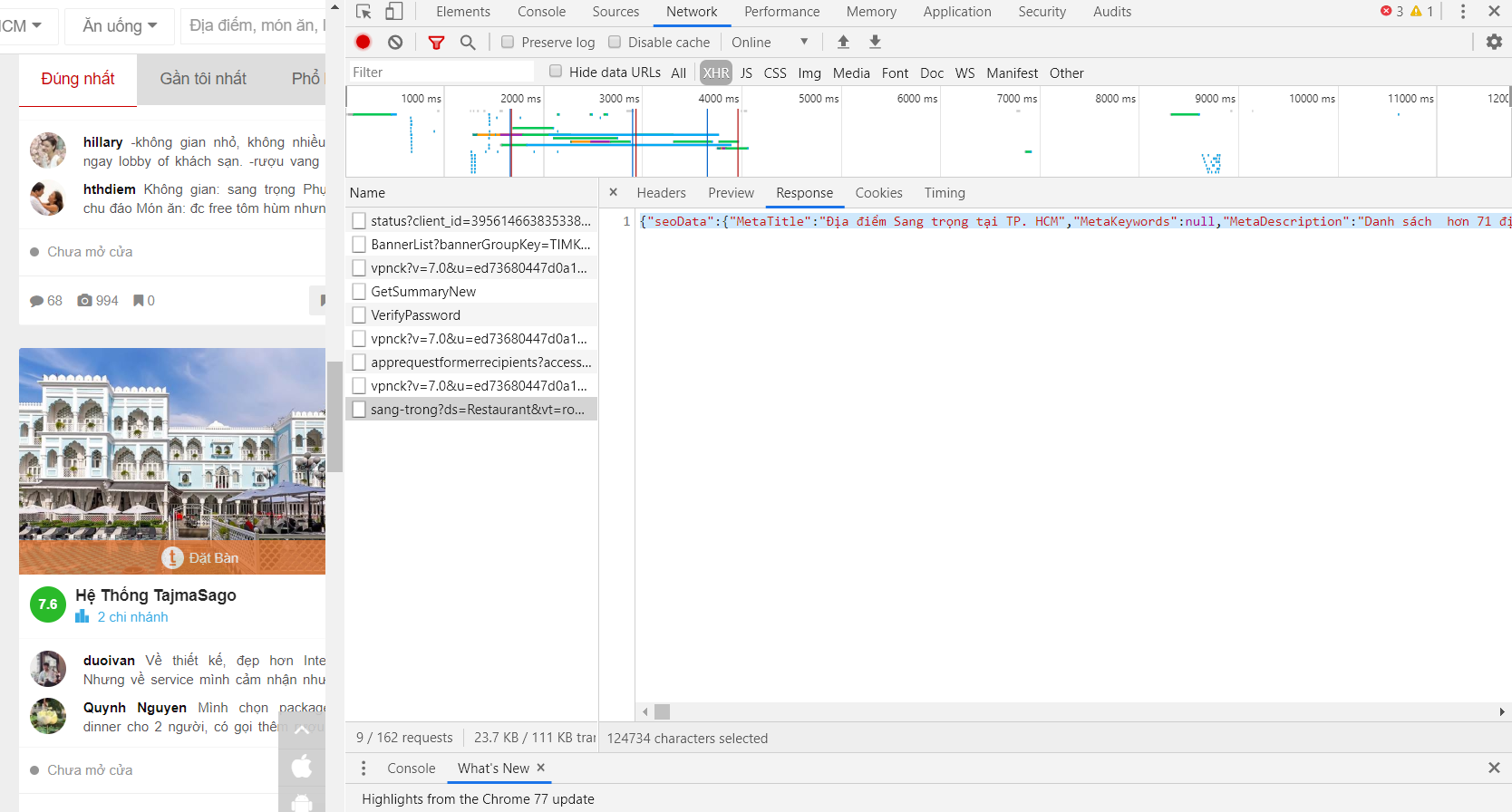
**Bước 3:** Ta để nguyên phần Kiểm tra như thế và bắt đầu kéo chuột trang web xuống dưới cho đến khi nào bên tab Network xuất hiện tên loại địa điểm mình chọn như hình bên dưới (chú ý chỗ sang-trong?ds=….)



Hình 9: Chọn địa điểm muốn lấy dữ liệu

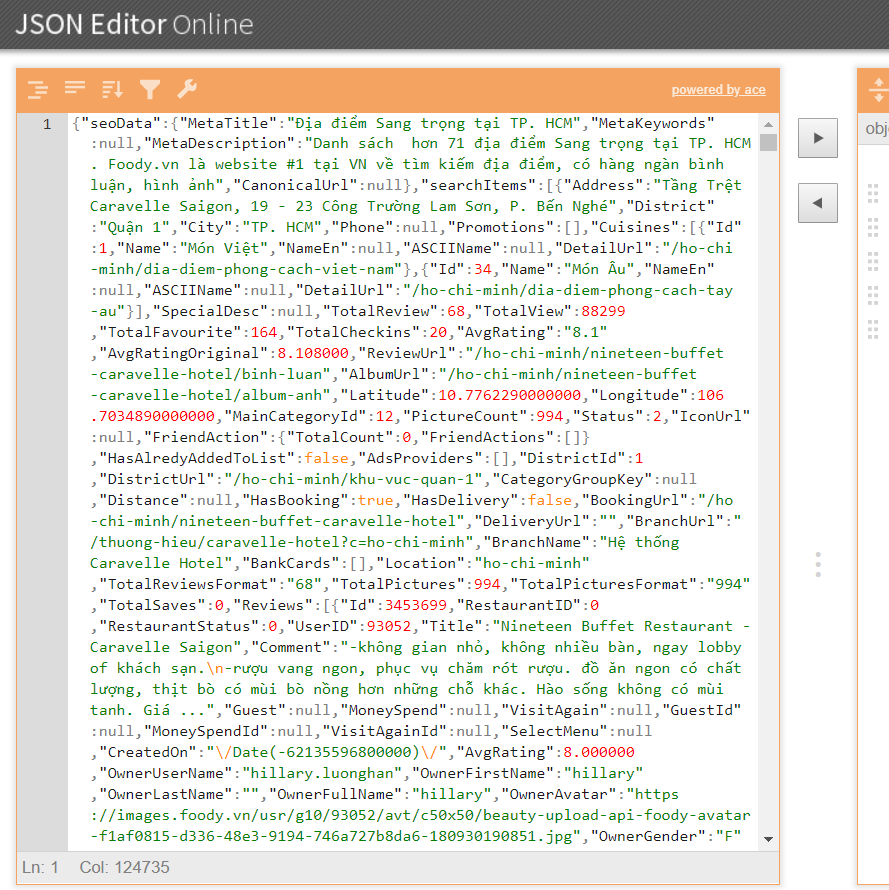
*(chú ý chỗ sang-trong?ds=….)*

**Bước 4:** Click vào dòng đó vừa xuất hiện đó, trong tab Respone (phản hồi) chính là dữ liệu dạng json mà trang web foody trả về cho ta (Khi ta kéo trang web sẽ gọi 1 **Request Method:** GET để lấy dữ liệu về hiện thị)



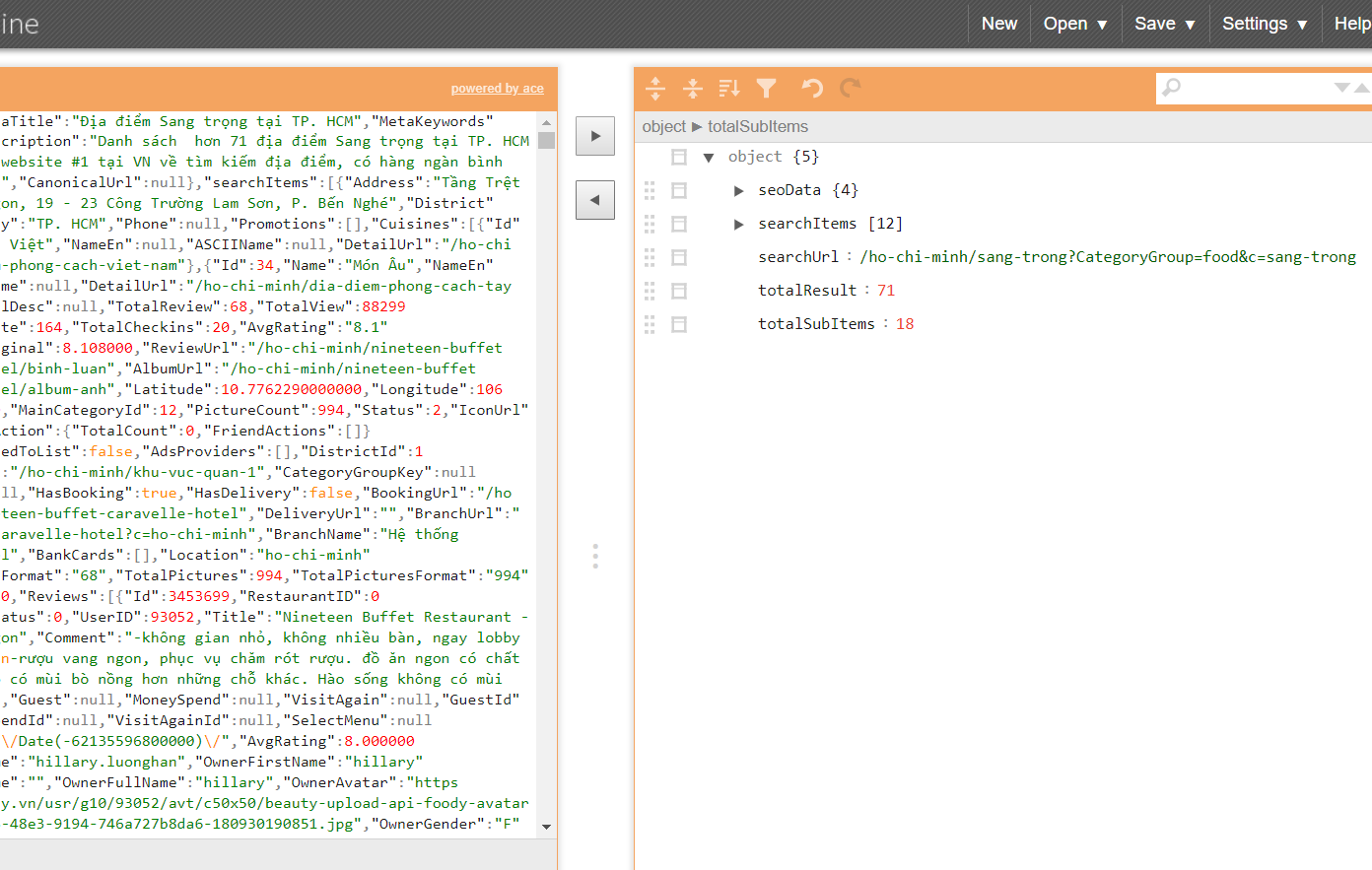
Hình 10: Lấy dữ liệu dạng JSON

**Bước 5:** Bạn sao chép hết tập Json trong tab Respone. Vào trang web thứ 2 và copy vào nội dung của khung bên trái của trang web.

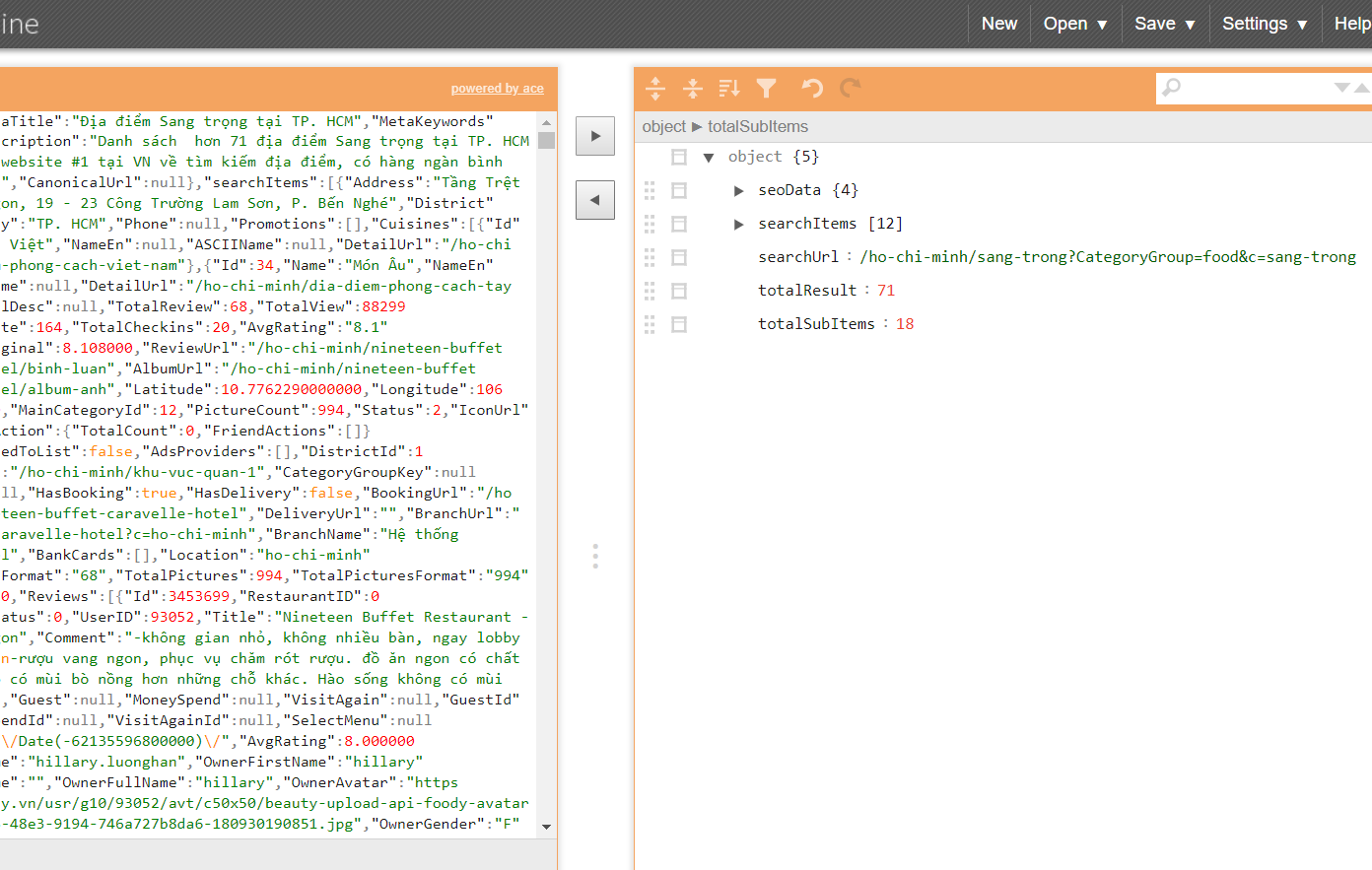


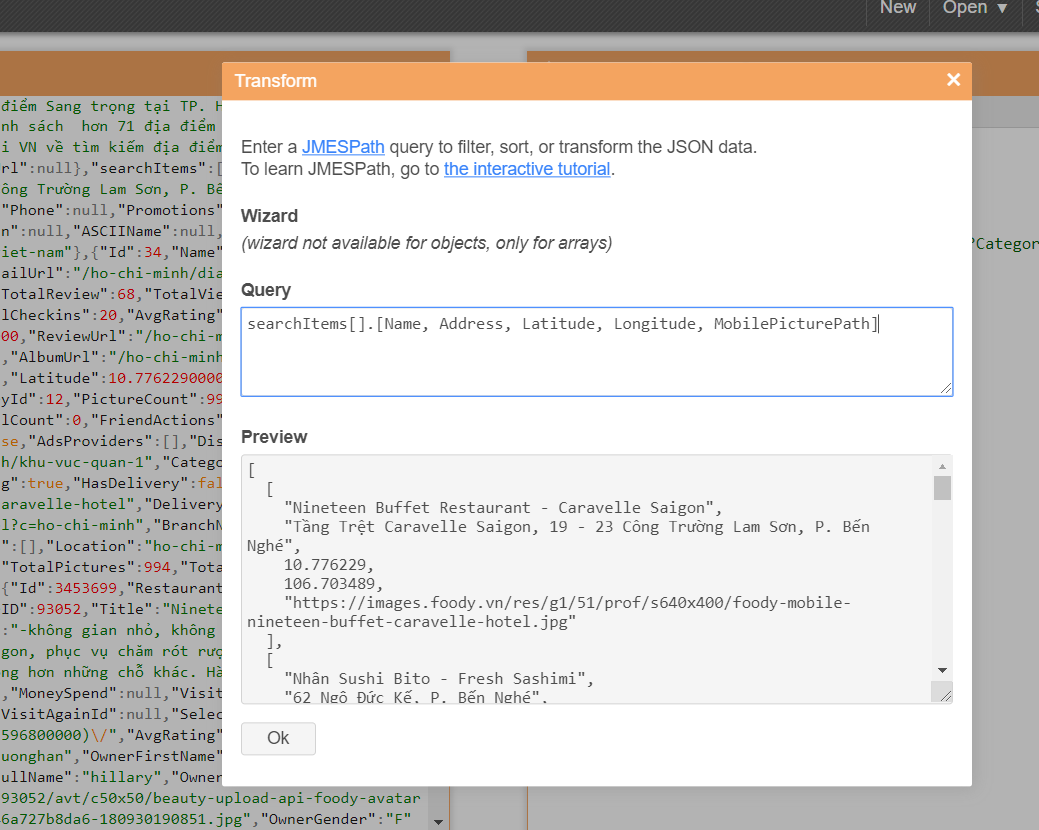
Hình 11: Chuyển dữ liệu sang trang https://jsoneditoronline.org/

Sau đó click vào button có hình mũi tên hướng sang phải để chuyển đổi dữ liệu Json từ khung bên trái sang bên phải. Kết quả thực hiện như hình dưới



Hình 12: Chuyển đổi dữ liệu

**Bước 6:** Ta lọc dữ liệu khung bên phải bằng cách click vào  và nhập điều kiện lọc dưới dạng chuỗi  [JMESPath](http://jmespath.org/) query vào khung thứ nhất và ta sẽ thu được dữ liệu được lọc theo mong muốn ở khung thứ hai như hình dưới.



Hình 13: Chọn lọc dữ liệu muốn lấy

**Bước 7:** Sao chép dữ liệu có được ở bước 6 vào file txt và tổ chức cho phù hợp.

*Ta sẽ lặp lại các bước trên để có đủ dữ liệu dùng cho project.*

1. **Google Maps API key là gì và cách lấy**

## Google Maps API là gì?

Google cho phép chúng ta sử dụng miễn phí dịch vụ bản đồ Google Maps có điều kiện. Và tùy thuộc vào lĩnh vực ngành nghề: Sinh viên, Lập trình viên, Chuyên gia marketing. Google sẽ có những điều kiện khác nhau để giới hạn phạm vi sử dụng. Google sẽ không tính phí.

Để **tích hợp Google Maps vào prohject** của mình. Bạn cần phải xác minh với Google. Việc xác minh này đã được Google đơn giản hóa quy trình. Thay vì bạn phải cung cấp tài khoản và mật khẩu như cách truyền thống. Giờ đây! Chúng ta chỉ cần 1 đoạn mã duy nhất gọi là **Google Maps API key**. Đoạn mã này sẽ được gửi lên Google để xác thực. Nếu hợp lệ Google sẽ gửi về dữ liệu cần tìm cho chúng ta. Đơn giản, nhanh chóng, bảo mật.

## Cách lấy Google Maps API key?

Trong quá trình quản trị website. Hẳn sẽ có đôi lần bạn gặp tình trạng bản đồ Google Maps không hiển thị. Làm thế nào để khắc phục tình trạng trên? Lý do hay gặp phải nhất là. Google Maps API key chưa chính xác. Google Maps API key chưa tạo. Hoặc bạn đã tạo, nhưng tên miền đang sử dụng chưa được cấp phép. Dưới đây là các bước hướng dẫn chi tiết. Cách để tạo một Google Maps API key mà mình thường dùng. Để triển khai dự án cho khách hàng của [Mua Theme](https://muatheme.vn/). Bạn cùng làm theo mình nhé.

**Bước 1: Sử dụng tài khoản Google cũ hoặc đăng ký mới**

Bạn cần có tài khoản Google.

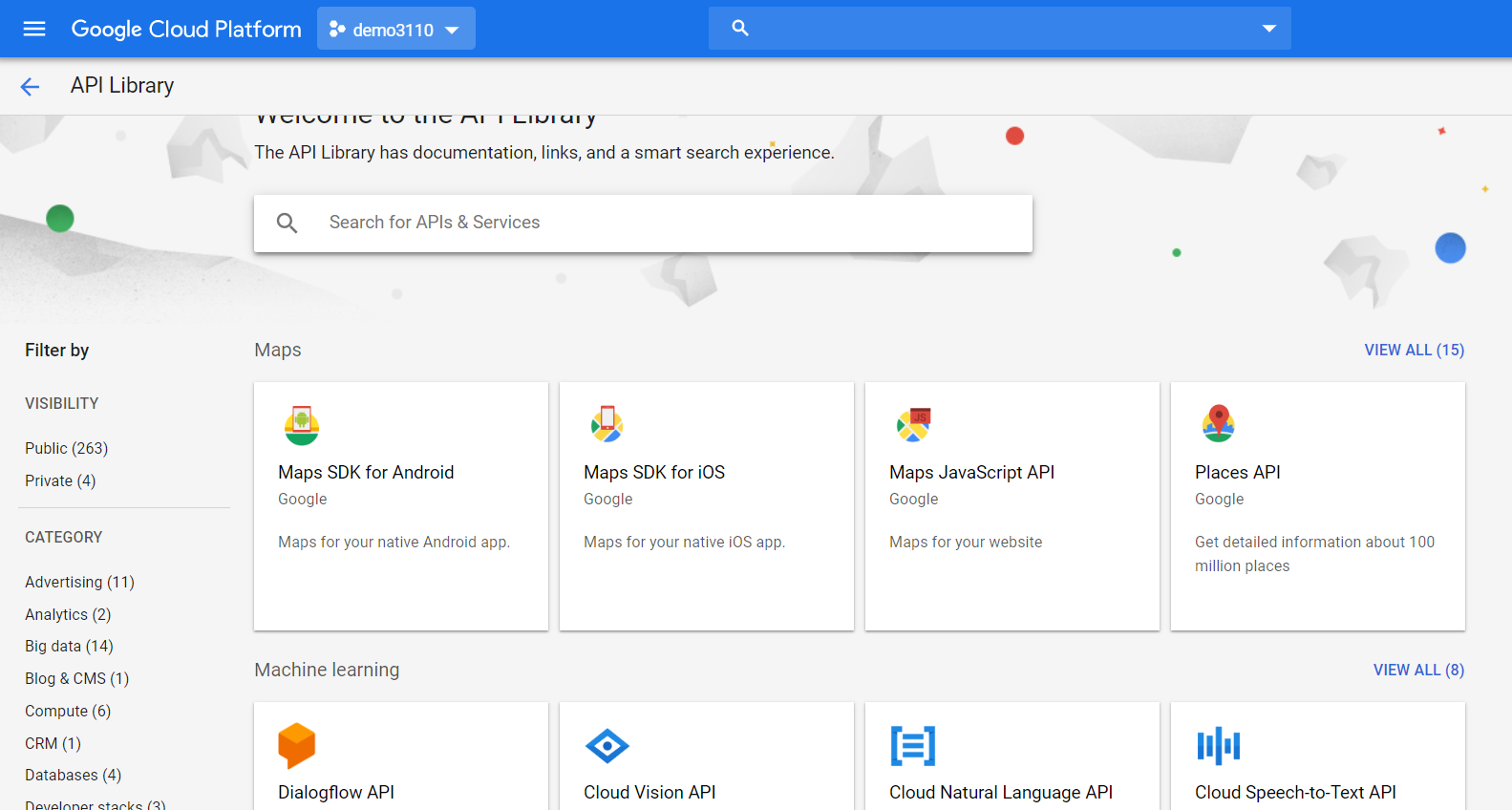
**Bước 2: Đăng nhập vào tài khoản Google**

Bạn tiếp tục vào đường dẫn:

[**https://console.cloud.google.com/apis/library?hl=en**](https://console.cloud.google.com/apis/library?hl=en), nếu Google yêu cầu bạn đăng nhập, thì bạn hãy sử dụng thông tin tài khoản ở bước 1 để đăng nhập và tiếp tục bước 3 nhé.

**Bước 3: Đăng nhập thành công**

Sau khi xác nhận thông tin tài khoản thành công. Bạn sẽ được Google chuyển hướng đến trang **Getting started** có giao diện như ảnh dưới.

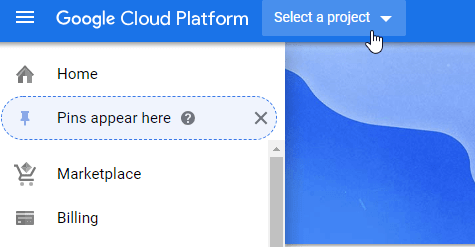


Hình 14: Giao diện trang quản lý các nền tảng đang đuợc cung cấp bởi Google

Ở đây, Google hỗ trợ rất nhiều các APIs khác nhau, phục vụ nhu cầu công việc của từng người.

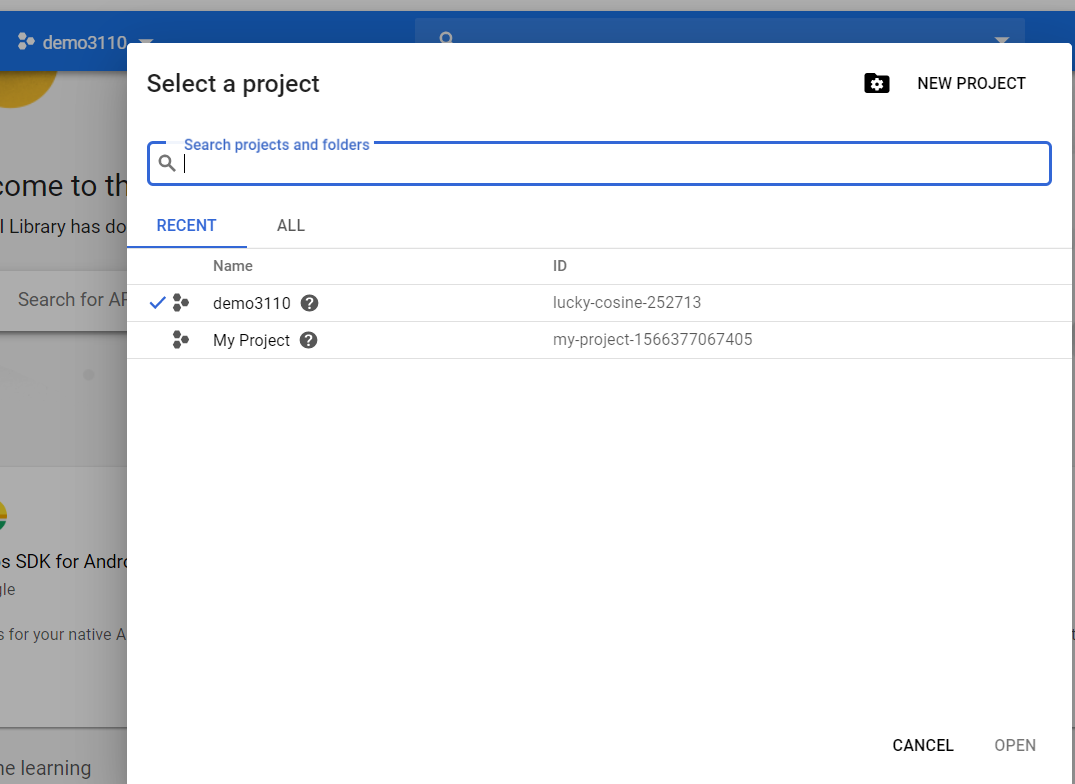
**Bước 4: Tạo dự án mới trên Google Cloud Platform**

Bạn hãy Click vào **Select a project** để chọn một project có sẵn hoặc tạo một project mới



Hình 15: Tạo Project mới trong Google Cloud Platform

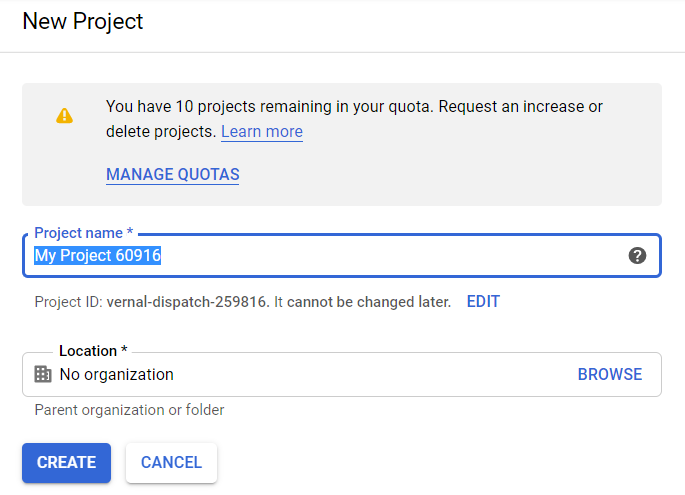
Nếu bạn đã có project trước đó rồi, bạn có thể bỏ qua phần này và chuyển tới bước 7.



Hình 16: Tạo project mới hoặc chọn project có sẵn

Tạo project mới hoặc chọn project có sẵn

Chọn **NEW PROJECT** để tạo 1 project mới.

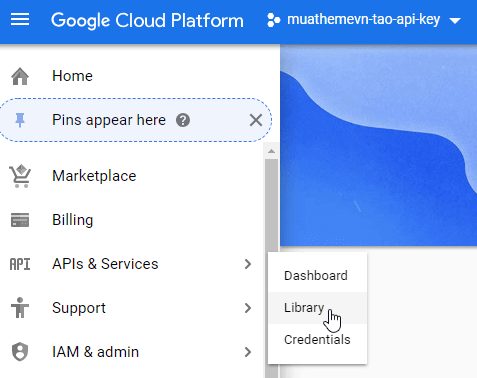


Hình 17: Khi bạn chưa đặt tên sẽ có 1 tên tự tạo sẵn như hình trên

Bạn có thể xóa và đặt tên cho project mới, sau đó bấm nút **CREATE** để khởi tạo project

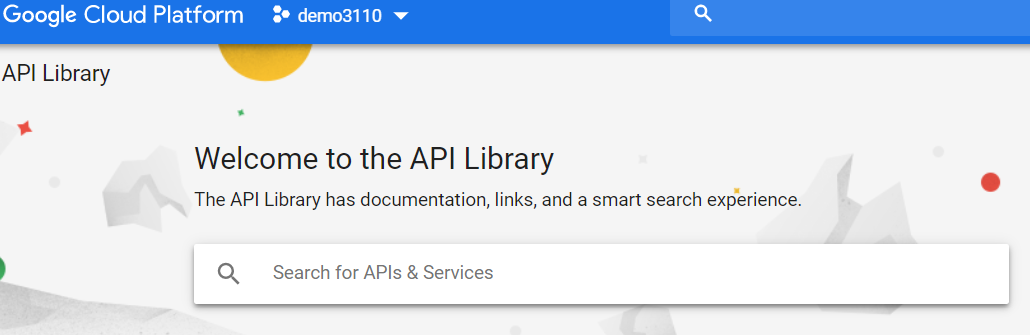
**Bước 5: Vào trang quản lý thư viện API để tìm API cho app**

- C1: Bạn rê chuột vào menu APIs & Services và click chọn menu Library



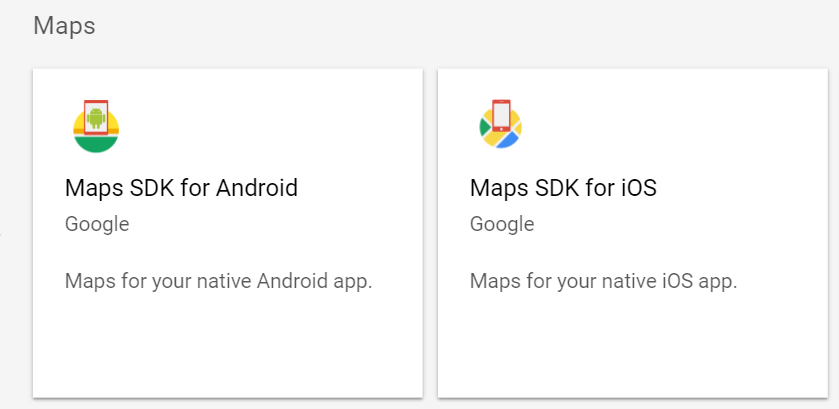
Hình 18: Chọn menu APIs & Services và chọn Library

- C2: Dùng thanh công cụ tìm kiếm, để tìm Google Maps API cho nhanh.



Hình 19: Thanh công cụ tìm kiếm Google Cloud Platform

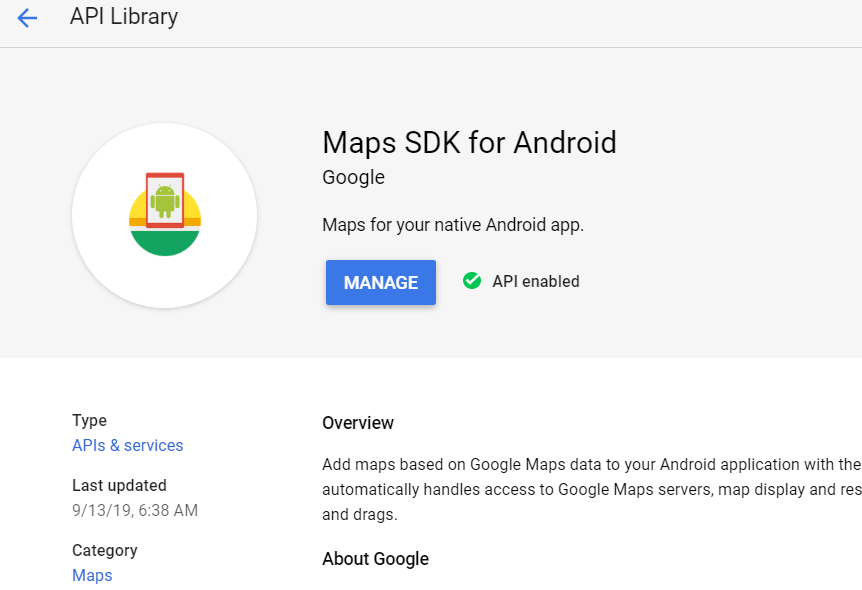
Click chọn Maps SDK for Android (nếu làm trên iOS thì chọn for iOS)



Hình 20: Chọn vào Maps SDK for Android

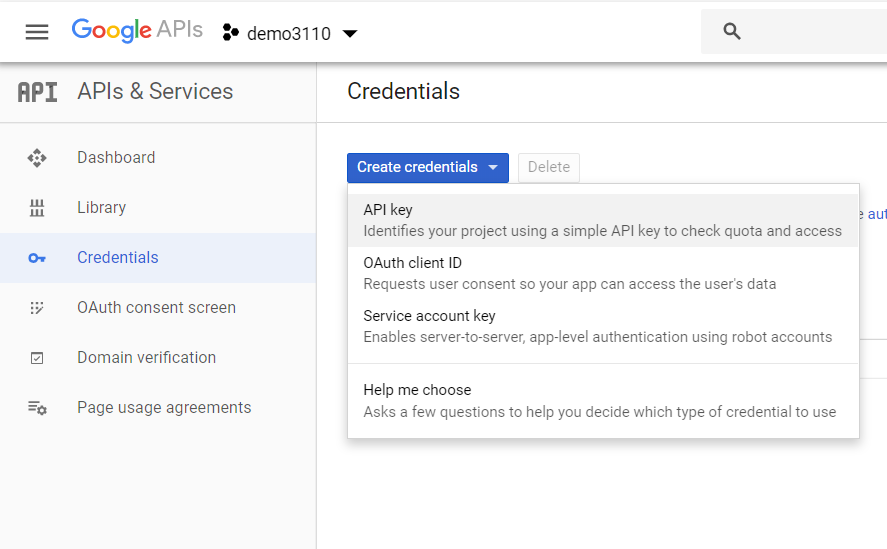
**Bước 6: Bật Google Maps API**

Bạn click vào nút **ENABLE** để bật thư viện Google Maps API lên.



Hình 21: click vào nút ENABLE để bật thư viện Google Maps API lên

Nếu bật Google Maps API thành công sẽ thấy giao diện trên hình. Bạn sẽ được chuyển hướng tới giao diện sau.

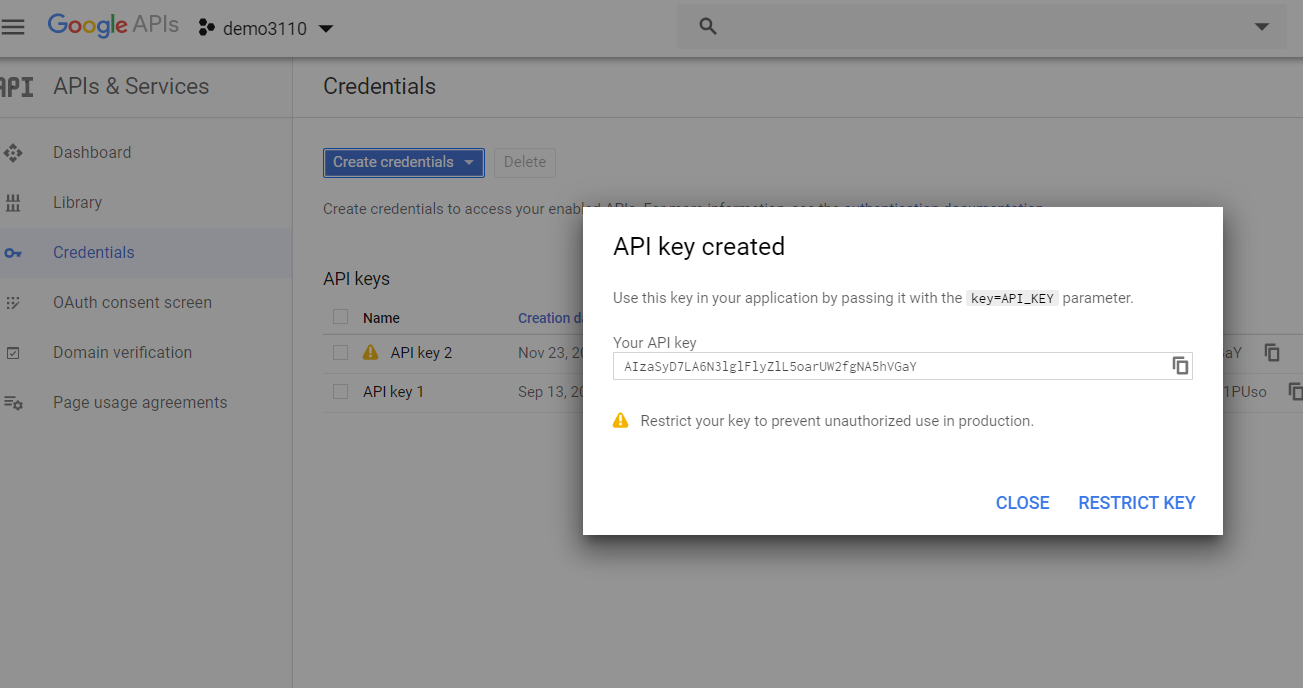


Hình 22: Chọn vào tab Credentials

Bạn tiếp tục Click chọn vào tab **Credentials**.

**Bước 7: Bắt đầu tạo Google Maps API key**

Tiếp theo, bạn Click chọn vào nút **Create credentials** và chọn **API key.** Sau khi Click vào **API key**, Google sẽ tự động tạo ra 1 đoạn mã ngẫu nhiên cho bạn.

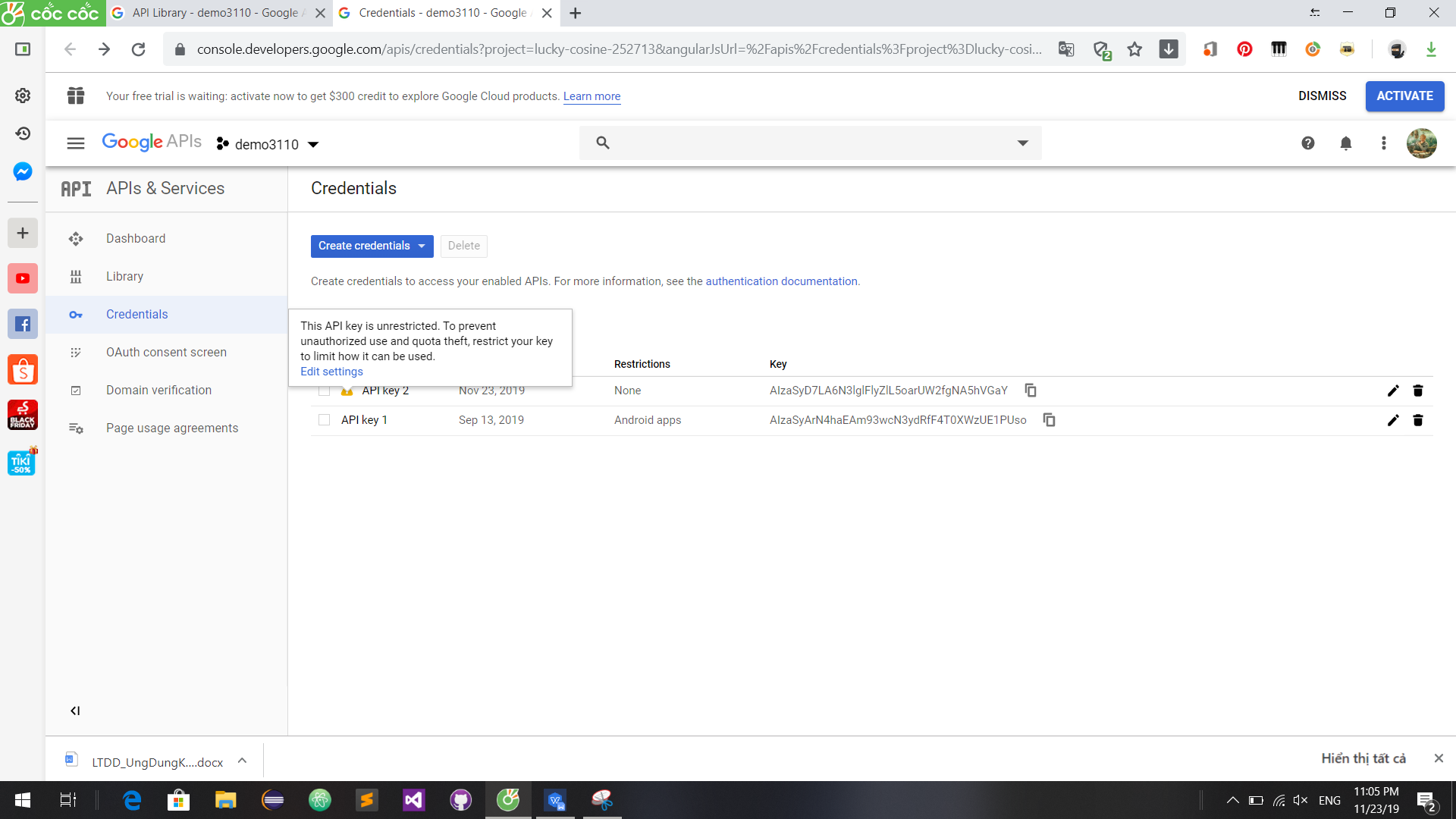


Hình 23: Click chọn vào nút Create credentials

Vậy là cuối cùng bạn cũng có được một mã **Google Maps API key** cho riêng mình rồi. Hãy lưu API key lại và đừng quên, nếu bạn không muốn phải làm lại bước 1.

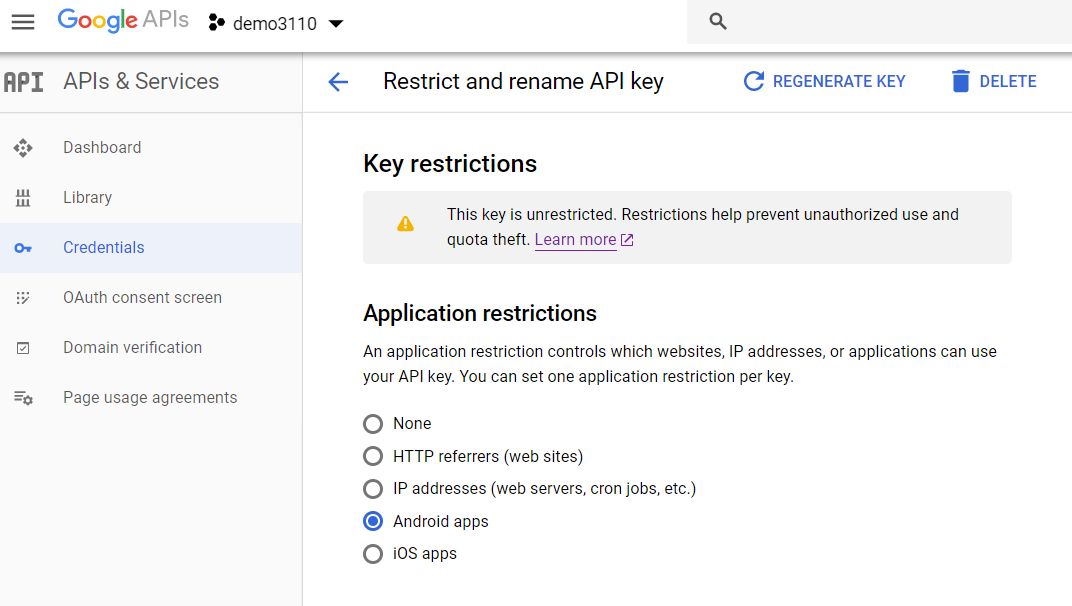
**Bước 8: Cấp quyền sử dụng Maps cho project android**

Bạn có để ý thấy biểu tượng hình tam giác màu vàng không? Google cảnh báo rằng: đoạn mã **Google Maps API key** vừa được khởi tạo của bạn chưa được an toàn. Bạn cần giới hạn nó để không bị sử dụng trái phép, vượt quá giới hạn cho phép.



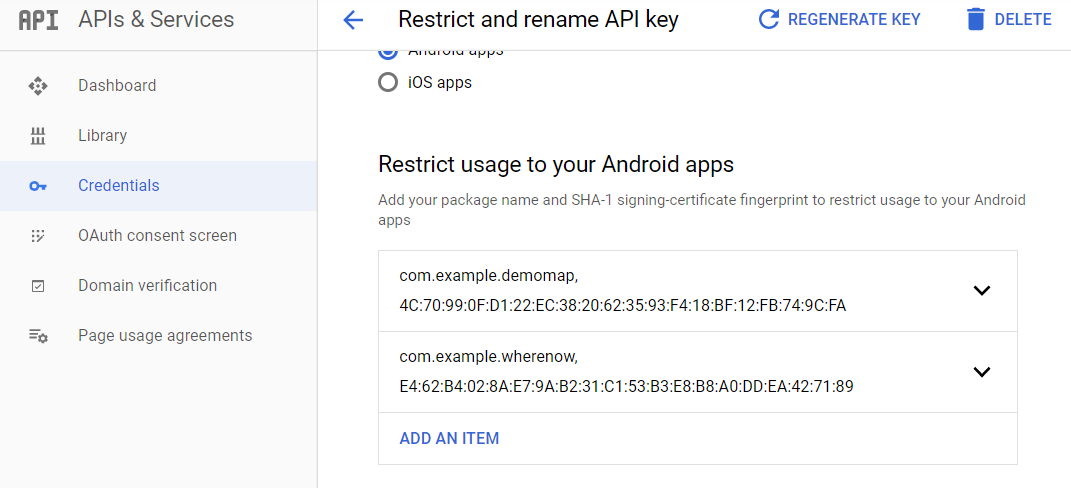
Hình 24: Click chọn vào <tên API key>

Để khắc phục vấn đề này bạn cần click chọn vào <tên API key> vừa tạo ra để thiết lập cho API.



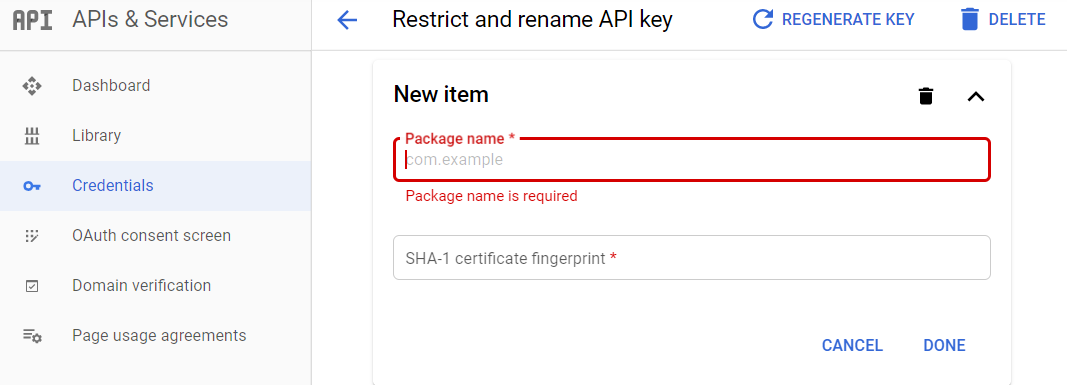
Hình 25: Chọn vào lựa chọn Android apps

Bạn tick chọn vào lựa chọn **Android apps**. Nhưng hiện giờ project của bạn vẫn chưa xài được key. Bạn cần chọn vào **ADD AN ITEM**



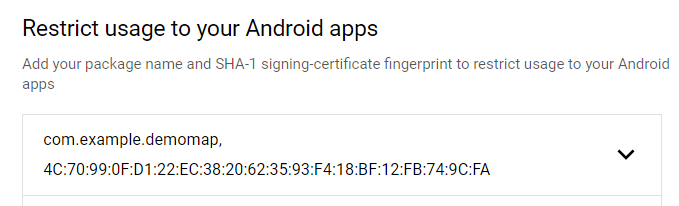
Hình 26: Chọn vào ADD AN ITEM

Tại đây bạn nhập vào tên Package project của bạn và chuỗi SHA-1 (trong **values > googlemaps\_api.xml**) tự sinh khi bạn tạo 1 project Google map mới vào theo yêu cầu.



Hình 27: Nhập vào tên Package project của bạn và chuỗi SHA-1

Sau đó bạn click vào **Done** để hoàn thành. Dưới là kết quả khi ta đã thành công việc cấp quyền sử dụng Goggle Maps API cho project.



1. **AsyncTask**
2. **Khái niệm**

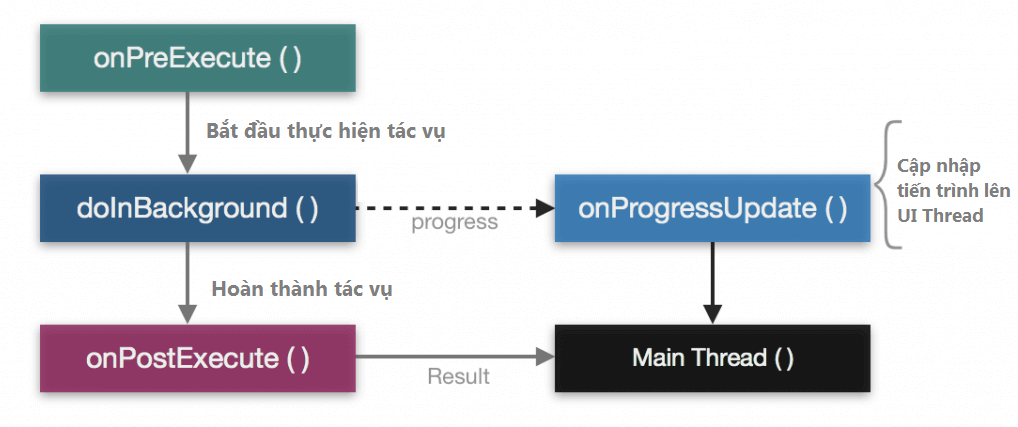
AsyncTask là một lớp trừu tượng được cung cấp bởi Android, giúp chúng ta sử dụng các thread UI đúng. Lớp này cho phép chúng tôi thực hiện các hoạt động dài / nền và hiển thị kết quả của nó trên thread UI mà không cần phải thao tác đề.

1. **Cách triển khai AsyncTask trong Android**

Bản chất **Asynctask** gồm có 4 bước:

**Bước 1:**  on PreExecute()

Được thực hiện trước khi bắt đầu thực hiện tác vụ. Hàm được gọi trước phương thức doInBackground() và được gọi trên UI thread.

Thông thường, hàm này được dùng để hiển thị thanh progressbar thông báo cho người dùng biết tác vụ bắt đầu thực hiện

Hình 28: Các bước sử dụng Asynctask

**Bước 2**: doInBackgroud()

Tất cả code mà cần thời gian thực hiện sẽ được đặt trong hàm này.  Vì hàm này được thực hiện ở một thread hoàn toàn riêng biệt với UI thread nên bạn không được phép cập nhật giao diện ở đây.

Để có thể cập nhập giao diện khi tác vụ đang thực hiện. Ví dụ như cập nhập trạng thái % file đã download được, chúng ta sẽ phải sử dụng đến hàm bên dưới onProgressUpdate()

**Bước 3:** onProgressUpdate()

Hàm này được gọi khi trong hàm doInBackgroud()

gọi đến hàm publishProgress()

**Bước 4**: onPostExecute()

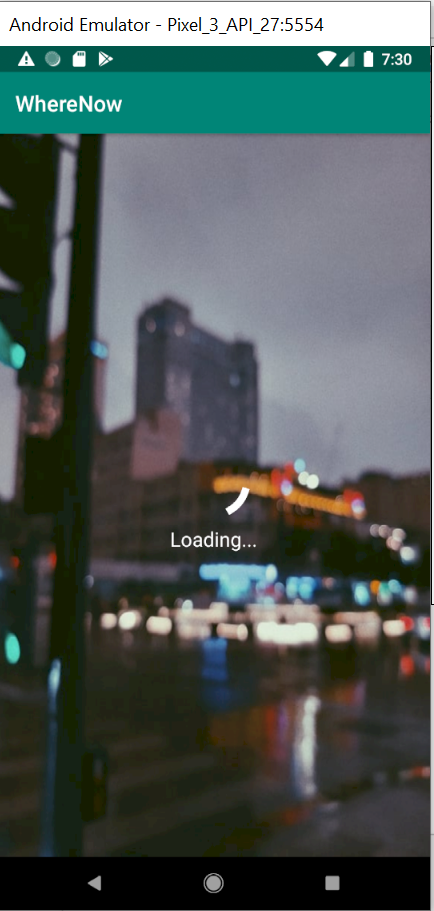
Hàm này được gọi khi doInBackground hàm thành công việc. Kết quả của doInBackground() sẽ được trả cho hàm này để hiển thị lên giao diện người dùng.

Trong quá trình Asynctask thực hiện tác vụ, bạn hoàn toàn có thể tạm dừng bất kể lúc nào mà không cần phải đợi AsyncTask làm xong. Đơn giản là bạn gọi hàm cancel(boolean)

# Chương 3: Demo và kết luận

1. **Giao diện và chức năng của giao diện**

* **Giao diện loading**



Hình 29: Giao diện loading

* **Giao diện chính của phần mềm**

Hình 30: Giao diện chính

Ở giao diện này sẽ hiển thị một giao diện bản đồ và gắn marker thể hiện vị trí của bản thân

****Thanh tìm kiếm với chức năng tìm kiếm vị trí

Hình 31: Thanh tìm kiếm

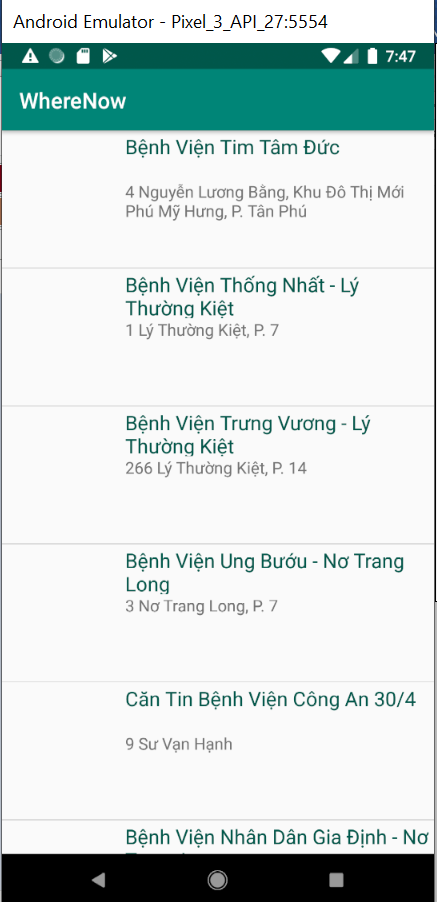
Dấu 3 gạch dùng để mở giao diện lịch sử tìm kiếm

****Thanh Scroll View chứa các Button thể hiện loại địa điểm. Bạn có thể Click vào để tìm kiếm các địa điểm theo ý muốn

Hình 32: Thanh Scroll View

* **Giao diện hiển thi danh sách địa điểm khi bạn click vào 1 button trên thanh Scroll View. Mỗi địa điểm sẽ hiển thị tên, địa chỉ và hình ảnh của địa điểm**

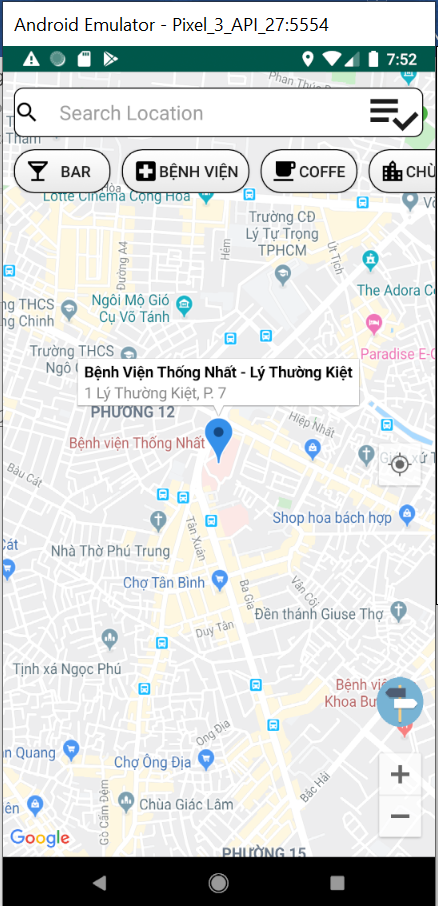
VD: ở đây chúng ta chọn vào button Bệnh Viện



Hình 33: Danh sách địa điểm lọc theo loại

* **Giao diện khi bạn chọn vào một địa điểm bất kỳ trong danh sách địa điểm**

Khi chọn vào một địa điểm thì sẽ gắn 1 marker tại vị trí của địa điểm đó trên bản đồ. Đồng thời hiển thị mô tả tên địa điểm và vị trí của địa điểm



Hình 34: Vị trí của địa điểm được chọn

# 

* **Giao diện khi ấn vào nút chỉ đường**

Phần mềm sẽ tìm ra đường gần nhất từ vị trí của bản thân đến địa điểm bạn chọn

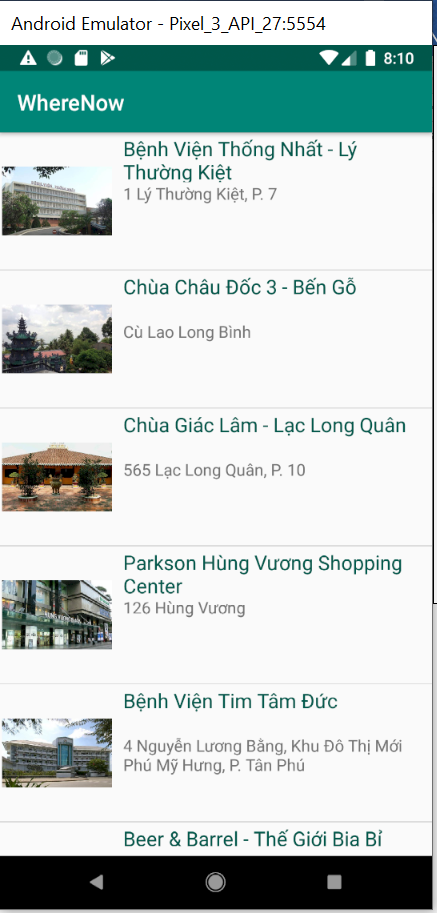
****

Hình 35: Tìm đường từ vị trí hiện tại đến địa điểm

# 

* **Giao diện lịch sử tìm kiếm**

Các địa điểm bạn đã tìm kiếm trước đó sẽ được lưu lại trong cơ sở dữ liệu và hiển thị tại đây



Hình 36: Lịch sử tìm kiếm

# 

## Kết Luận

Sau một thời gian tìm hiểu về android và đề tài quản lý địa điểm kết hợp google map. Chúng em đã thực hiện được đề tài theo như dự kiến. Với mục đính là tìm hiểu về hệ điều hành android , hiểu biết về ngôn ngữ lập trình 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 các ứng dụng android để xây dựng ứng dụng quản lý địa điểm kết hợp android trên hệ điều hành android cho thiết bị di động. Sau quá trình tìm hiểu và nghiên cứu đề tài, chúng em đã bước đầu hiểu được về hệ điều hành androi trên thiết bị điện thoai. Viết và nắm vứng quá trình tạo và sử dụng 1 file project trong android. Hiểu và sử dụng được các đối tượng API của Google, phương thức hỗ trợ lập trình trên android. Hiểu được cơ chế làm việc với SQLite để lấy và cập nhật dữ liệu cho ứng dụng. Do thời gian ngắn cùng với kiến thức bản thân còn hạn chế, nên chúng em chưa tìm hiểu sâu về kiến thức tới hệ điều hành android nên phần mềm còn nhiều hạn chế, chúng em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và nhiều chức năng hơn.

# Tài Liệu Tham Khảo

Tài liệu tiếng việt:

1. Slide bài giảng Lập trình trên thiết bị di động

Tài liệu tiếng anh:

1. Android Notes For Professionals

Tài liệu website:

1. <https://mona.media/google-maps/>
2. https://viettimes.vn/kham-pha-lich-su-cua-he-dieu-hanh-android-nguon-goc-bieu-tuong-va-ten-goi-toan-keo-banh-297481.html