**Giới thiệu hệ điều hành Android**

Android là một hệ điều hành mở dành cho thiết bị di động dựa trên một phiên bản được chỉnh sửa của Linux. Hệ điều hành này ban đầu được phát triển bởi một công ty cùng tên là Android Inc. Vào năm 2005, với chiến lược thâm nhập vào thị trường di động đầy tiềm năng, Google đã mua lại Android và tham gia vào việc phát triển và hoàn thiện nó.

Google muốn Android trở thành một hệ điều hành mở và hoàn toàn miễn phí, tuy nhiên, hầu hết mã nguồn của Android đều được cho ra mắt dưới bản quyền Apache dành cho mã nguồn mở, vì vậy bất cứ ai muốn sử dụng Android phải tải về toàn bộ mã nguồn của Android. Hơn thế nữa, các nhà sản xuất (đặc biệt là nhà sản xuất phần cứng) có thể thêm vào phần mở rộng của họ vào Android và tùy biến nó để trở nên khác biệt so với sản phẩm của hãng khác. Kiểu mẫu phát triển như vậy đã khiến cho Android trở nên rất hấp dẫn và thu hút nhiều nhà sản xuất. Điều này đặc biệt có ý nghĩa đối với những công ty đang bị ảnh hưởng bởi hiện tượng iPhone của Apple, một sản phẩm rất thành công tạo nên một cuộc cách mạng trong ngành công nghiệp Smartphone bấy giờ. Trong số những công ty đó, có Motorola và Sony Ericsson, những công ty cũng nhiều năm phát triển hệ điều hành di động của riêng mình. Khi iPhone ra mắt, những công ty đó phải tìm nhiều cách để có thể làm hồi sinh những sản phẩm của mình. Và họ đã tìm ra câu trả lời, chính là Android. Họ quyết định tiếp tục phát triển phần cứng của riêng mình, và sử dụng Android như là hệ điều hành.

Một trong những thuận lợi khi lựa chọn Android, đó là nó thay đổi cách tiếp cận của sự phát triển ứng dụng. Các nhà phát triển (Developer) chỉ phải phát triển ứng dụng cho Android, và các ứng dụng của họ sẽ có thể chạy trên vô số các thiết bị, miễn là sử dụng hệ điều hành Android. Trong thế giới Smartphone, các ứng dụng là phần quan trọng nhất trong dây chuyền thành công của sản phẩm. Các nhà sản xuất thiết bị coi Android là hi vọng lớn nhất của họ để có thể cạnh tranh với iPhone của Apple, một thiết bị đã có quá nhiều những ứng dụng lớn.

1. **Các phiên bản Android**

Kể từ khi phát hành chính thức, Google đã cho ra mắt rất nhiều phiên bản khác nhau của Android, có thể kể đến thời điểm hiện tại những phiên bản đã được ra mắt như dưới đây:

|  |  |
| --- | --- |
| Phiên bản | Tên mã |
| 1.1 |  |
| 1.5 | Cupcake |
| 1.6 | Donut |
| 2.0/ 2.1 | Éclair |
| 2.2 | Froyo |
| 2.3 | Gingerbread |
| 3.0/ 3.1/ 3.2 | Honeycomb |
| 4.0 | Ice Cream Sandwich |
| 4.1 | Jelly Bean |
| 4.2 | Jelly Bean |
| 4.3 | Jelly Bean |
| 4.4 | KitKat |

Kể từ phiên bản 1.5, các phiên bản Android lần lượt được đặt tên theo các loại bánh, theo thứ tự của bảng chữ cái. Phiên bản mới nhất vừa được ra mắt là phiên bản 4.4 với tên mã là KitKat.

Phiên bản KitKat ra mắt vào tháng 11/2013, từng bước hiện thực ước muốn của Google, đó là phổ biến hệ điều hành phiên bản mới nhất cho các dòng điện thoại có cấu hình ở mức trung bình. Google cho biết phiên bản KitKat sẽ có thể điều tiết lượng RAM tùy thuộc vào phần cứng của thiết bị, và KitKat có thể chạy trên những thiết bị có RAM là 512MB.

1. **Các tính năng của Android**

Vì Android là mã nguồn mở và hoàn toàn miễn phí đối với các nhà sản xuất để có thể tùy biến, nên không có một tùy chỉnh cấu hình phần cứng và phần mềm cố định nào. Tuy nhiên, bản thân Android hỗ trợ các tính năng sau:

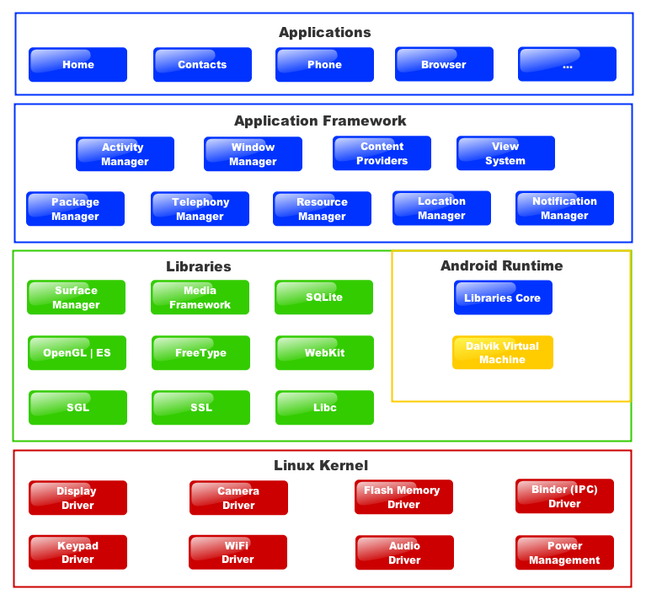
* Lưu trữ: Sử dụng SQLite để lưu trữ dữ liệu.
* Kết nối: Hỗ trợ GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (bao gồm A2DP và AVRCP), Wi-Fi, LTE và WiMAX.
* Tin nhắn: Hỗ trợ MMS và SMS.
* Trình duyệt: Dựa trên mã nguồn mở WebKit, cùng với Engine V8 JavaScript của Chrome.
* Hỗ trợ đa phương tiện: Hỗ trợ các chuẩn media: H.263, H.264 (ở định dạng 3GP hoặc MP4), MPEG-4 SP, AMR, AMR-WB (ở định dạng 3GP), AAC, HE-AAC (ở định dạng MP4 hoặc 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF và BMP.
* Hỗ trợ phần cứng: Cảm biến gia tốc (Accelerometer Sensor), Camera, La bàn số (Digital Compass), Cảm biến tiệm cận (Proximity Sensor) và GPS.
* Đa chạm: Hỗ trợ màn hình đa chạm.
* Đa nhiệm: Hỗ trợ những ứng dụng đa nhiệm.
* Hỗ trợ Flash: Android 2.3 hỗ trợ Flash 10.1.
* Phát sóng (Tethering): Hỗ trợ chia sẻ kết nối Internet như là một trạm phát không dây (Wireless hotspot).

1. **Cấu trúc của Android**

Để có thể hiểu cách thức vận hành của Android, ta sẽ cùng tìm hiểu các lớp (layers) mà tạo nên hệ điều hành Android.

Hệ điều hành Android được chia thành năm phần với bốn lớp chính:

* **Linux Kernel**: đây là kernel (nhân) mà Android dựa trên. Lớp này bao gồm toàn bộ những drivers thiết bị cấp thấp cho các thành phần phần cứng đa dạng của thiết bị Android.
* **Libraries**: Lớp này bao gồm toàn bộ code mà cung cấp những tính năng chính của Android. Ví dụ thư viện SQLite cung cấp hỗ trợ cơ sở dữ liệu mà nhờ đó ứng dụng có thể sử dụng để lưu trữ dữ liệu, hoặc thư viện WebKit cung cấp những tính năng để duyệt web.
* **Android runtime**: Ở cùng lớp với Libraries, Android runtime cung cấp một bộ thư viện lõi mà giúp các nhà phát triển có thể viết ứng dụng Android bằng ngôn ngữ lập trình Java. Android runtime còn bao gồm máy ảo Dalvik, giúp cho mỗi ứng dụng Android có thể chạy một process riêng bằng một Instance của máy ảo Dalvik (các ứng dụng Android được biên dịch thành các file thực thi Dalvik). Dalvik là một máy ảo được thiết kế đặc biệt cho Android và được tối ưu cho các thiết bị dùng pin với bộ nhớ và CPU hạn chế.
* **Android Framework**: Chỉ ra những khả năng của hệ điều hành Android, từ đó các nhà phát triển Android có thể sử dụng chúng trong những ứng dụng của mính.
* **Applications**: Ở lớp trên cùng này, ta có thể thấy những ứng dụng đi kèm với thiết bị Android (Danh bạ, Trình duyệt, vv) cũng như những ứng dụng ta download và cài đặt từ Market. Tất cả những ứng dụng mà ta viết đều nằm ở lớp này.



1. **The Android Market**

Giống như đã nói ở trên, yếu tố chính quyết định đến sự thành công của một nền tảng di động chính là những ứng dụng hỗ trợ nó. Từ sự thành công của iPhone có thể thấy rất rõ ràng rằng những ứng dụng có vai trò vô cùng quan trọng trong việc quyết định một nền tảng mới sẽ tồn tại hay bị diệt vong. Thêm vào đó, việc làm cho những ứng dụng này đến với những người dùng phổ thông là hết sức quan trọng.

Vì vậy, vào tháng 8/2008, Google thông báo sẽ mở ra Android Market, một kho ứng dụng online cho những thiết bị Android, và mở cửa vào tháng 10/2008. Sử dụng ứng dụng Market được cài đặt sẵn, người dùng có thể download những ứng dụng của bên thứ ba một cách đơn giản. Market hỗ trợ cả ứng dụng miễn phí và trả phí, mặc dù một vài ứng dụng trả phí chỉ dành cho những người dùng ở những quốc gia nhất định vì lí do pháp lí.

1. **Những công cụ cần thiết**

Để có thể phát triển ứng dụng Android, ta cần download những công cụ cần thiết và các bộ SDK.

Để lập trình Android, ta có thể sử dụng một máy tính chạy hệ điều hành Windows, MAC hoặc Linux. Tất cả những công cụ đều miễn phí và có thể download trực tiếp từ Web.

Trước khi download Android SDK, ta cần cài đặt Java JDK, vì Android SDK sử dụng Java SE Development Kit (JDK). Ta có thể download dễ dàng trên trang chủ của Oracle: www.oracle.com/technetwork/java/javase/downloads/index.html

Tiếp theo sau khi đã download và cài đặt JDK, ta sẽ download và cài đặt phần mềm quan trọng nhất, chính là Android SDK. Android SDK bao gồm debugger, các thư viện, bộ mô phỏng emulator, các tài liệu, code mẫu và các bài hướng dẫn.

Ta có thể download Android SDK trên website dành cho nhà phát triển của Android: http://developer.android.com/sdk/index.html

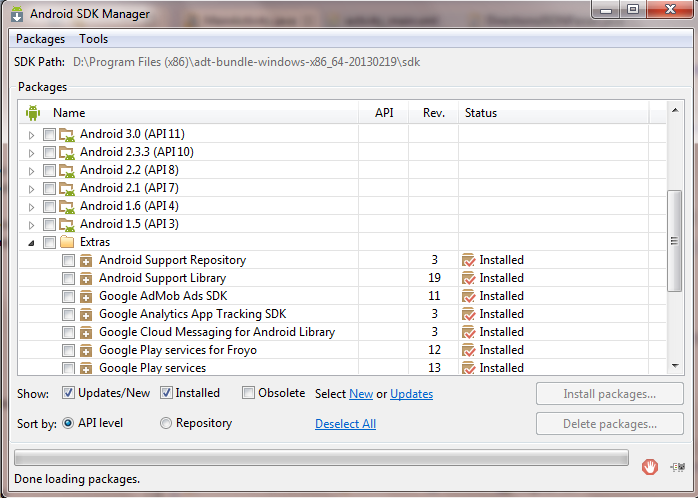
Sau khi download xong Android SDK, ta tiến hành cài đặt, lưu ý khi chọn địa chỉ lưu bộ SDK.

1. **Tùy chình Android SDK Manager**

Android SDK Manager là một phần mềm quản lý các phiên bản Android SDK hiện tại được cài đặt trong máy. Khi khởi động nó sẽ cho thấy một danh sách các mục và các mục đó đã được cài đặt hay chưa.

Đánh dấu vào các mục ta cần download, thường thì bộ SDK download từ ở trên sẽ chỉ gồm Platform phiên bản Android mới nhất, nếu muốn lập trình trên những phiên bản Android thấp hơn, ta phải chọn download các Platform còn thiếu từ SDK Manager.

Đề tài luận văn là “Lập trình xe tự hành dựa trên Google Maps API”, nên phần quan trọng không thể thiếu là bộ SDK phải hỗ trợ Google Maps API. API này nằm trong Google Play Service, nên ta phải download mục này thì mới có thể lập trình được.

Hình ảnh của Android SDK Manager:

1. **Eclipse**

Bước tiếp theo là cài đặt môi trường phát triển tích hợp (IDE) để phát triển ứng dụng Android. Và Google khuyến khích sử dụng Eclipse, một IDE hỗ trợ lập trình nhiều ngôn ngữ phổ biến như Java, Ada, C, C++, COBOL, Python, …

Để lập trình Android, ta nên download Eclipse IDE for Java EE Developers. Ta phải chọn hệ điều hành phù hợp để download phiên bản Eclipse phù hợp cho mình. Trong trường hợp này là Windows 64 bit.

Các bước download và cài đặt khá đơn giản. Chỉ cần giải nén file download được là có thể sử dụng được Eclipse bằng cách mở file thực thi eclipse.exe.

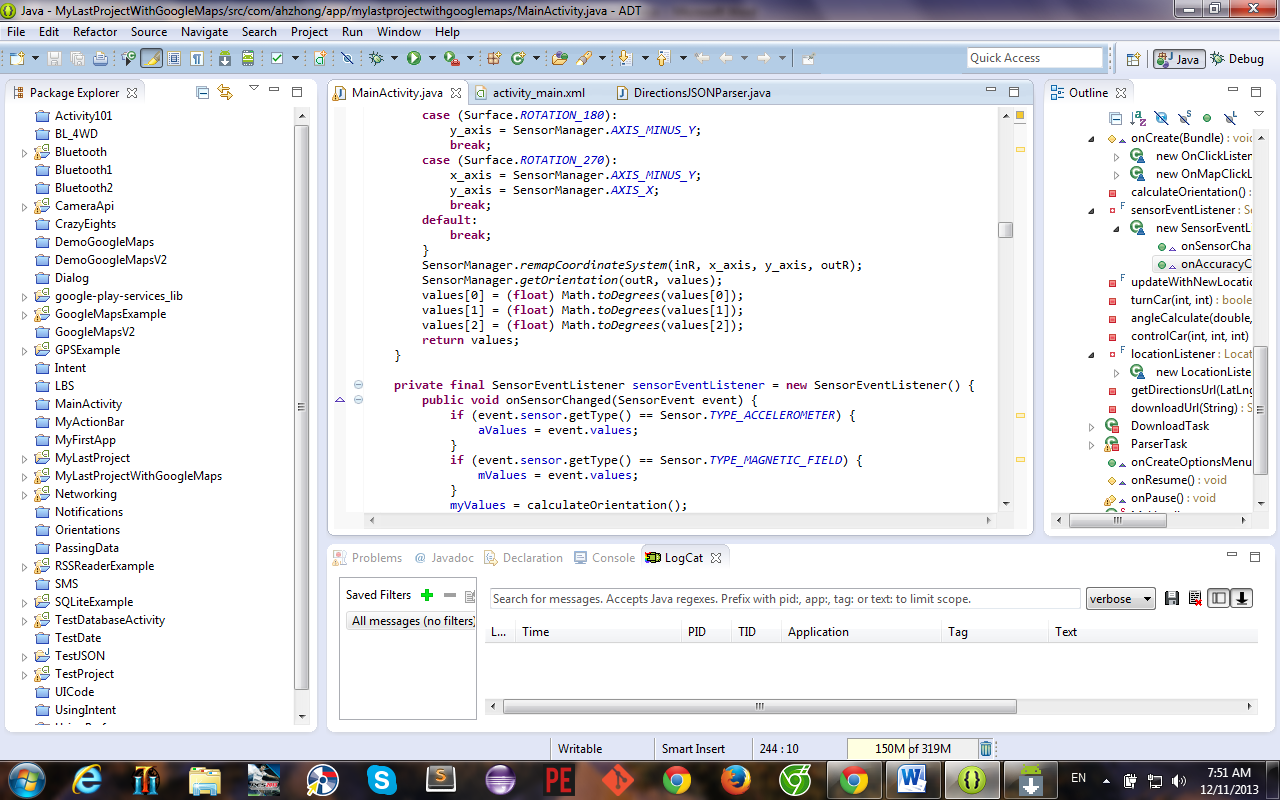
Đối với những bản Android SDK sau này (từ 4.2 trở đi), thì Google đã giúp tích hợp Eclipse vào bộ Android SDK, khi chúng ta giải nén Android SDK thì cũng có sẵn một thư mục Eclipse, ta không cần phải download thêm ở ngoài nữa.

Thêm vào đó, việc tích hợp sẵn Eclipse vào bộ cài đặt Android SDK cũng giảm nhiều công sức cho người dùng, vì phiên bản Eclipse này đã được cài đặt sẵn phần mở rộng ADT, mà ngày xưa người dùng phải tự cài đặt thêm khi cài đặt Eclipse.

Tiện ích mở rộng ADT sẽ giúp chúng ta:

* Tạo project Android mới.
* Tiếp cận với các công cụ để tiếp cận bộ Emulator hoặc thiết bị Android của chúng ta.
* Biên dịch và debug chương trình Android.
* Export ứng dụng Android (file APK).
* Tạo chứng chỉ số cho chữ kí mã hóa cho file APK.

Hình ảnh của Eclipse:

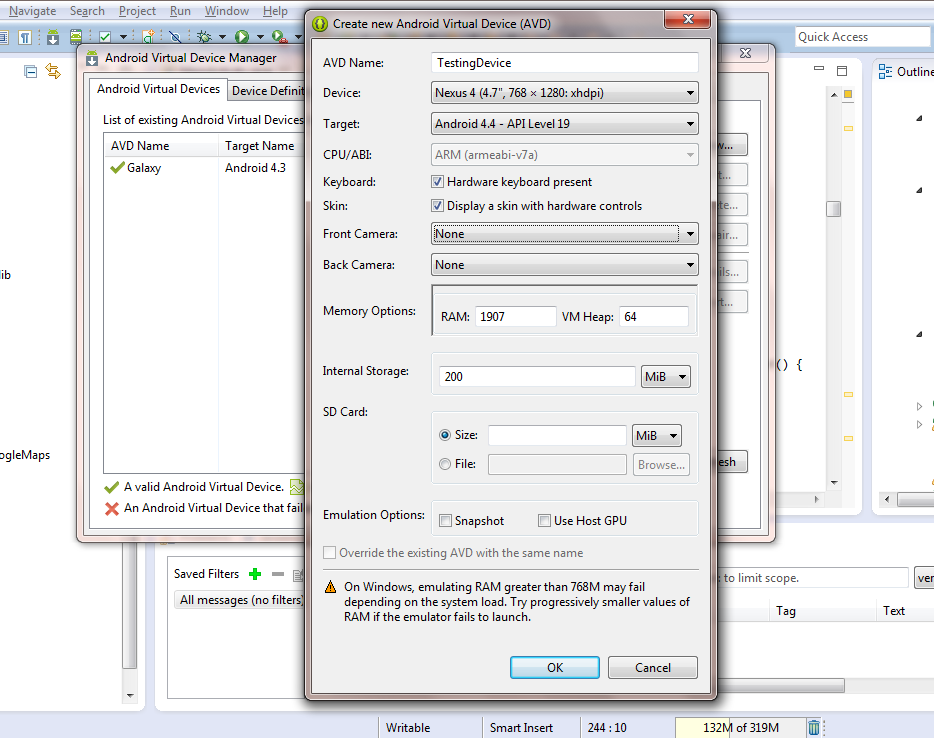


1. **Tạo thiết bị ảo Android (Android Virtual Devices – AVD)**

Một phần không thể thiếu của bộ công cụ lập trình ứng dụng Android, chính là thiết bị ảo Android – AVD. AVD được Google tích hợp vào bộ Android SDK nhằm giúp người dùng có thể chạy thử các ứng dụng của họ. AVD có thể mô phỏng hoạt động của một thiết bị thực. Mỗi AVD có một cấu hình phần cứng, file hệ thống cũng như bộ nhớ (vd như SD card).

Ta có thể tạo nhiều AVD để có thể chạy thử chương trình trên nhiều cấu hình phần cứng và phần mềm khác nhau. Việc chạy thử này rất quan trọng để kiểm tra ứng dụng có hoạt động đúng trên nhiều thiết bị có cấu hình khác nhau hay không.

Trong Eclipse, ta tạo AVD bằng cách vào Window -> AVD Manager, chọn New ở cửa sổ Android Virtual Device Manager, cửa sổ mới hiện lên sẽ cho ta tùy chình AVD.



Như ở hình ví dụ ở trên, ta đang tạo một AVD với tên là TestingDevice, chọn mô phỏng điện thoại Nexus 4, hệ điều hành là Android 4.4, các thông số RAM và bộ nhớ trong được set như trên hình.

Khi lập trình ứng dụng thì ta nên tạo nhiều Android với nhiều cấu hình phần cứng và phần mềm khác nhau để kiểm tra sự tương thích và hoạt động của ứng dụng.

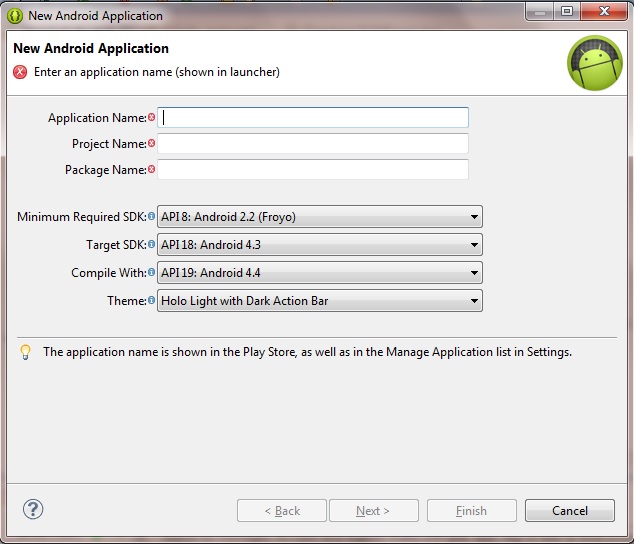
Tuy nhiên, AVD cũng có một vài nhược điểm:

* Vì là một thiết bị ảo Android, nên dẫn đến AVD chạy không được mượt như thiết bị thực, thậm chí đáp ứng rất chậm, gây khó khăn trong việc chạy thử ứng dụng.
* AVD không thể mô phỏng những ứng dụng đòi hỏi sử dụng phần cứng của điện thoại, ví dụ như Bluetooth, Sensor, …

Trong phạm vi của luận văn, rõ ràng không thể sử dụng AVD để mô phỏng, do luận văn sử dụng nhiều tính năng phần cứng của một thiết bị thực như Bluetooth và các Sensor, vì vậy chỉ có thể chạy trên một thiết bị thực mới có thể kiểm tra được ứng dụng.

1. **Tạo một ứng dụng Android trong Eclipse**

Trong Eclipse, mở File -> New -> Android Application Project.



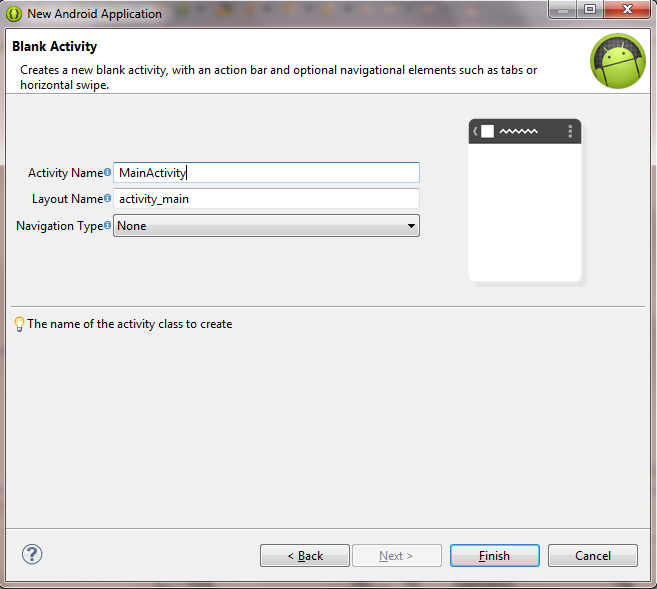
Sau đó nhập các field yêu cầu:

* Application Name: tên của ứng dụng, là tên mà người dùng nhìn thấy, nên có thể có khoảng trắng và mô tả ứng dụng.
* Project Name: tên của project trong Eclipse, và bắt buộc đặt tên theo quy định của một project trong Eclipse (không được có khoảng trắng và các kí tự đặc biệt).
* Package Name: tên của package chứa các Class của project. Trong phạm vi luận văn này, package name còn dùng để lấy API key mà Google cung cấp cho các ứng dụng sử dụng Google Maps API.
* Minimum Required SDK: nền tảng Android thấp nhất mà ứng dụng hỗ trợ.
* Target SDK: là nền tảng Android mà ứng dụng hướng đến, có thể coi là nền tảng Android cao nhất mà ứng dụng hỗ trợ.
* Compile with: là nền tảng Android dùng để biên dịch ứng dụng, thường chọn bằng với Target SDK.

Sau khi thiết lập xong ta chọn Next, ở hai bước tiếp theo là các công đoạn lựa chọn các tùy chọn (như workspace, icon), ta có thể để mặc định hai bước này và bấm Next để đến bước tiếp theo.

Bước tiếp theo là Create Activity, Activity có thể hiểu là một Class để định nghĩa hành vi của ứng dụng khi tương tác với người dùng, có thể coi đơn giản Activity là một cửa sổ để người dùng có thể tương tác thông qua các đối tượng. Ở mục tiếp theo bên dưới sẽ nói thêm về Activity trong một ứng dụng Android.

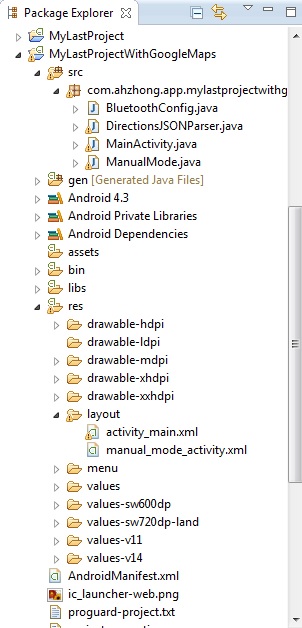
Cửa sổ này ta có thể để mặc định dấu check ở Blank Activity, nghĩa là Eclipse sẽ tạo cho ta một Activity trống. Ta bấm Next qua bước tiếp theo.



Ở cửa sổ tiếp theo, ta sẽ điền vào hai field:

* Activity Name: tên của Activity, thông thường trong một chương trình Android sẽ có rất nhiều Activity khác nhau, tương ứng với những cửa sổ mà người dùng tương tác, Activity Name nên đặt theo behavior của nó (ví dụ như Activity của mode manual nên đặt tên là ManualModeActivity).
* Layout Name: vì Activity định nghĩa các Class dùng để tương tác với người dùng, nên ứng với mỗi Activity sẽ có một Layout dùng để mô tả giao diện cho nó, ta cũng nên đặt tên Layout có liên quan đến Activity mà nó mô tả.

1. **Cấu trúc của một ứng dụng Android**

Chúng ta sẽ cùng xem xét những file và folder được tạo ra trong quá trình phát triển ứng dụng Android:

* Src: là thư mục chứa các file mã nguồn .java cho project, ta sẽ thấy các file này nằm trong package mà ta đặt tên lúc đầu, ngoài các Activity ra ta có thể tạo thêm nhiều Class thông thường khác để định nghĩa các Config cần thiết cho project.
* Gen: Chứa file R.java, một file tạo bởi bộ biên dịch để tham chiếu đến tất cả tài nguyên trong project. Ta không nên chỉnh sửa file này, vì tất cả các tài nguyên sẽ được tự động biên dịch vào file này và ta có thể tham chiếu đến nó.
* Android 4.3: chứa file android.jar, chứa toàn bộ các thư viện cần thiết cho ứng dụng Android.
* Assets: chứa những asset dùng bởi ứng dụng, như HTML, database hoặc file text, …
* Bin: folder này chứa các file được build bởi ADT (Android Development Tool), và đây cũng là nơi tạo ra file .apk (Android Package). File .apk là file thực thi của Android, nó chứa mọi thứ cần để chạy được ứng dụng Android. Có thể coi file .apk tương đương với file .exe của hệ điều hành Windows.
* Res: folder này chứa tất cả tài nguyên (resources) của project. Folder này có một vài sub-folder như drawable - <resolution> dùng để chứa các file ảnh, layout dùng để chứa các file layout, values dùng để chứa các giá trị (như string, array, …)
* AndroidManifest.xml: đây là file khai báo của project. Ta sẽ phải khai báo toàn bộ các permission (ví dụ như Internet permission, Bluetooth permission, …) và khai báo toàn bộ các Activity dùng trong chương trình.

Hình ảnh ví dụ của AndroidManifest.xml



Trong một file AndroidManifest.xml, trước hết vì đây là định dạng file XML, nên nó sẽ có cấu trúc bao gồm các thẻ <element></element>, và bên trong có những element khác lồng vào để định nghĩa các element lớn hơn.

* AndroidManifest.xml trước hết khai báo Package Name của project đang thực hiện trong thuộc tính package của thẻ <manifest>.
* Tiếp theo là khai báo mã phiên bản (version code), khai báo này nhằm mục đích để xác định xem một ứng dụng đã được cài sẵn có cần nâng cấp hay không.
* Tiếp đến là tên phiên bản (version name), đây là tên được hiển thị tới người dùng, thường đặt theo cú pháp là <major>.<minor>.<point>.
* Tiếp đến là thuộc tính minSdkVersion của thẻ <uses-sdk>, dùng để khai báo phiên bản Android thấp nhất mà ứng dụng hỗ trợ.
* Ứng dụng sẽ lấy ảnh ic\_launcher.png nằm ở thư mục drawables để làm icon.
* Trong ví dụ này, chương trình chỉ có một Activity nên ta chỉ khai báo một Activity này trong file AndroidManifest.xml

Khi ta khai báo trong file Manifest và tạo các folder, các file, thì lập tức Eclipse sẽ tự động build file R.java. Ta cùng xem ví dụ một file R.java như bên dưới:



Giới thiệu Activity và Process LifeCycle

Trong các chương trình Android, các màn hình mà người dùng tương tác sẽ được coi là các Activity. Một Activity là có thể hiểu như một chương trình đơn mà người dùng tập trung vào. Trong chương trình Android, các Activity khác so với các Class thông thường khác. Các Class nói chung, có thể bao gồm các Class định nghĩa các đối tượng để sử dụng trong chương trình. Trong số đó, có các Class đặc biệt mà ta đã định nghĩa là các Activity như ở trên. Một Activity thường có mục đích chính là để tương tác với người dùng. Vì vậy Android cung cấp cho chúng ta một Class Activity để ta có thể thừa kế và tùy biến cho các Activity của riêng mình.

Một trong những điều Class Activity mang lại cho chúng ta, là nó sẽ cung cấp hàm để tạo nên một cửa sổ cho người dùng, từ cửa sổ đó chúng ta có thể tùy biến Giao Diện Người Dùng (User Interface – UI). Hàm mà Class Activity cung cấp cho chúng ta là hàm “setContentView(View)”. Trong đó View là một Class được định nghĩa sẵn của hệ điều hành Android, mô tả thuộc tính cho các đối tượng mà người dùng sẽ tương tác (có thể kể ra như Button, TextView, EditText, …).

Các Activity thường được hiển thị lên màn hình theo kích thước full-screen để có thể dễ dàng tương tác với người dùng, ta hoàn toàn có thể thay đổi các thể hiện mặc định này:

* Dạng màn hình động (floating window)
* Dạng embedded (được nhúng bên trong những Activity khác)