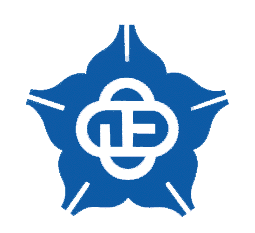
**國立中正大學通訊工程學系**

**Jo團去**

**電腦網路專題報告**

****

**專題成員**

**管韋傑**

**張喬彥**

**李宗諺**

**指導教授:侯廷昭 博士**

**指導助教:溫英佐 助教**

**中華民國 102年 6月**

**目錄**

[**第一章** **前言** 2](#_Toc360310934)

[1.1 概述 2](#_Toc360310935)

[1.2 報告架構 2](#_Toc360310936)

[**第二章** **全球定位系統GPS** 3](#_Toc360310937)

[2.1簡介 3](#_Toc360310938)

[2.2 系統元件 3](#_Toc360310939)

[2.3 定位原理 3](#_Toc360310940)

[2.4定位誤差來源 4](#_Toc360310941)

[**第三章** **Google Map定位** 6](#_Toc360310942)

[3.1 Google Map 定位 6](#_Toc360310943)

[3.2 GPS衛星定位 6](#_Toc360310944)

[3.3 WiFi定位 7](#_Toc360310945)

[**第四章** **Google App Engine** 9](#_Toc360310946)

[4.1 Google App Engine 簡介 9](#_Toc360310947)

[4.2 GAE運作原理 9](#_Toc360310948)

[4.3 NoSQL 10](#_Toc360310949)

[4.4 Eclipse開發環境 10](#_Toc360310950)

[**第五章** **Android 程式設計** 11](#_Toc360310951)

[5.1 Multithread programming 11](#_Toc360310952)

[5.2 Apache HTTP套件 13](#_Toc360310953)

[5.3 Android Google Map 14](#_Toc360310954)

[5.4 SQLite 17](#_Toc360310955)

[5.5 Sliding Menu 19](#_Toc360310956)

[5.6 Android Life Cycle 24](#_Toc360310957)

[**第六章** **系統架構與整合** 26](#_Toc360310958)

[6.1 系統架構 26](#_Toc360310959)

[6.2後端結構 27](#_Toc360310960)

[6.3前端結構 28](#_Toc360310961)

[**第七章** **結論與未來展望** 29](#_Toc360310962)

[**第八章** **參考資料** 30](#_Toc360310963)

1. **前言**

## 1.1 概述

全球定位系統使用已久，然而硬體和軟體的革新與結合，使得需求者可以利用各種產品來實現GPS的定位功能，並且加以融入日常的生活，我們為了能夠使定位系統更能結合進日常用途，便將此作為學習的過程的目標，然而這個目標需要一個媒介，而這個媒介又必須要能夠被廣泛使用且容易維護，於是在多方考量後，我們注意到了年輕族群日常使用最頻繁的，手機。

　　手機的系統大致可分為iOS、Windows Phone和Android，其中又以Android所提供的開發環境最適合廣大開發族群，所有的軟體資訊一切免費，並且還提供網頁能夠自行入門的訓練及youtube上免費的教學，網路上更有廣泛的開發者與使用者願意一同討論程式的革新與維護，在如此具有互動的平台裡面，可以使得我們在開發程式的過程中能夠快速地解決前人遇到的問題，又能將我們的想法快速實現。於是我們在這次的專題開發裡，選用了Google所提供的Android App以及Google Maps API作為主要的開發工具，決心設計一套能夠理清繁雜活動與事件的手機軟體，名為Jo團去。

## 1.2 報告架構

本篇專題報告中，依順序如下，第二章節報告GPS運作的原理，說明GPS的來由、目的和定位原理，第三章為Google Map的運作原理，除了使用GPS衛星定位以外，還有WiFi模式下的定位，第四章為本次專題所使用的GAE後台資料庫，說明它的運作原理，第五章為Android前端的程式設計，解釋本專題成品的運作原理，第六章為所有物件合成後的架構，前端與後端的互相溝通，第七章為本組最後的結論心得和未來的開發方向，最後第八章為本報告的參考資料。

1. **全球定位系統GPS**

## 2.1簡介

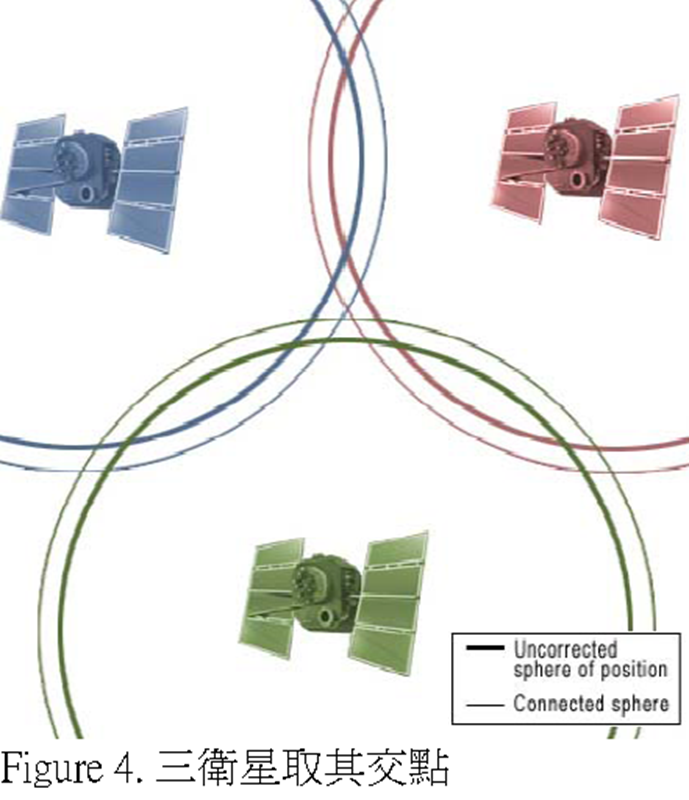
1973年由美國國防部開發，以軍事用途為優先並提供三度空間定位、定時功能有不受天候影響、系統功能連續不間斷、全球性等。具備低功率與低價位，可應用於多種設備上，例如汽車、手機、看護器。

## 2.2 系統元件

共有24顆衛星環繞於地球表面，分成6個軌道，每一個軌道有四顆衛星繞轉，地面有主控站負責所有的訊號規範，還有地面天線負責接發訊息和監測站負責監控衛星動態。

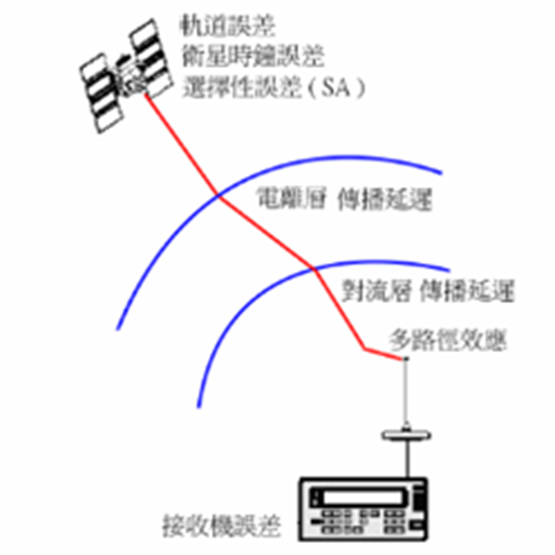
## 2.3 定位原理

一個地點的定位至少需要三顆衛星，利用彼此距離和時間差算出使用者與三顆衛星之間的距離，將衛星和接收端之間的距離作球面方程式，取其交點即為使用者的經緯度。第四顆衛星主要是作為時間修正的部份。



## 2.4定位誤差來源

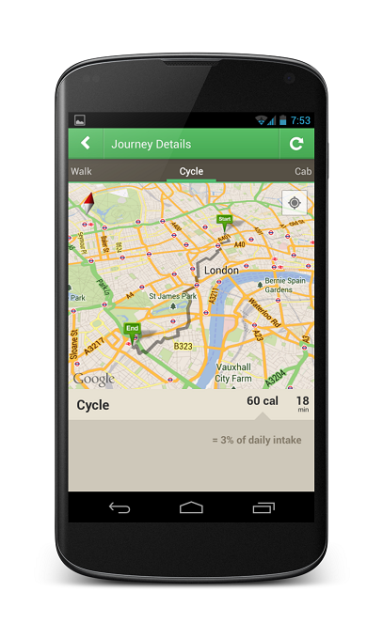
可分為:和**衛星有關**、**信號傳播有關**、**接收機有關種誤差**。第一種和衛星有關是衛星上的時間和地表不同步而產生的誤差，第二種信號傳播而產生的誤差，可能發生原因在於衛星到地表的路徑上，可能遭遇電磁波，產生路徑彎曲，或者是天氣氣候產生不同的結果，或者是建築物的障礙，影響了波的傳遞，最後接收機的誤差，可能是接收機本身出現故障而導致數據資料錯誤。



1. **Google Map定位**

## 3.1 Google Map 定位

Google 至今開發出許多友善的Web Service以及開發用API，使得開發者有更多的機會去學習與使用具創意且實用的功能，並能自行更改與更新開發，甚至開發結果優良也會分享至Google的平台上。舉凡Android這套系統便是如此。而這次專題實驗，我們應用到了Google所提供的Google Maps API也是一項具有創意且實用的技術。



## 3.2 GPS衛星定位

靠的是環繞在地球四周的數顆衛星發送訊號到地面，交叉連集出所在地的方法。從十幾年前開始出現汽車導航產品，就是使用這種技術。初次定位時常要花上一分鐘以上的時間，這也是目前汽車導航和智慧型手機都有提供的定位方式。當有遮蔽物阻擋來自天空的訊號時就會發生定位的困難。

　　Google Maps在定位時，與往常的GPS定位有很大的變化，由前面可以得知，當我們在使用GPS定位的時候是使用衛星訊號，因為參雜了人為的變數，所以在定位上會有不精準的結果，而且裝置必須得配置GPS的定位器才行使用。但Google Maps卻提供非GPS定位裝置也能精確定位的技術，只需要連上WIFI或是3G通訊，便能實現這個功能。

　Ｑ：然而Google Maps為何可以這樣定位呢？

**Ａ：答案便是因為WiFi定位功能**

## 3.3 WiFi定位

　　WiFi定位靠的是偵測附近周圍所有的無線網路基地台 (WiFi Access Point) 的 MAC Address (類似 10-78-D2-93-58-C2 這樣的格式)，去比對資料庫中該 MAC Address 的座標，交叉連集出所在地。此法尚須有網路連線做資料庫查詢才能完成定位。

　　但這其中便有一個問題點，要能建構如此龐大的ＷiFi Mac Address Data是如何做到的呢？答案是依靠了Google的街景車。



街景車除了拍街景之外，還實作了幾個工程：

1. 沿路蒐集所有公開的無線網路 MAC Address，與當時的經緯度一併記錄。

2. 根據拍下的街景來建立建築物 3D 模型資料。



但資料庫一定會面臨過期與需要更新的問題，這時候全世界有在使用行動裝置 + Google 定位程式的人，等於是在不知不覺中幫忙做這件事情。更新的方法是當 WiFi 與手機基地台定位或 GPS 定位同時開啟時，手持裝置藉由手機基地台定位或 GPS 定位這兩種方式可以獲得目前的座標，再透過 WiFi 搜尋到附近所有的 MAC Address，背後向 Google 的資料庫做更新。

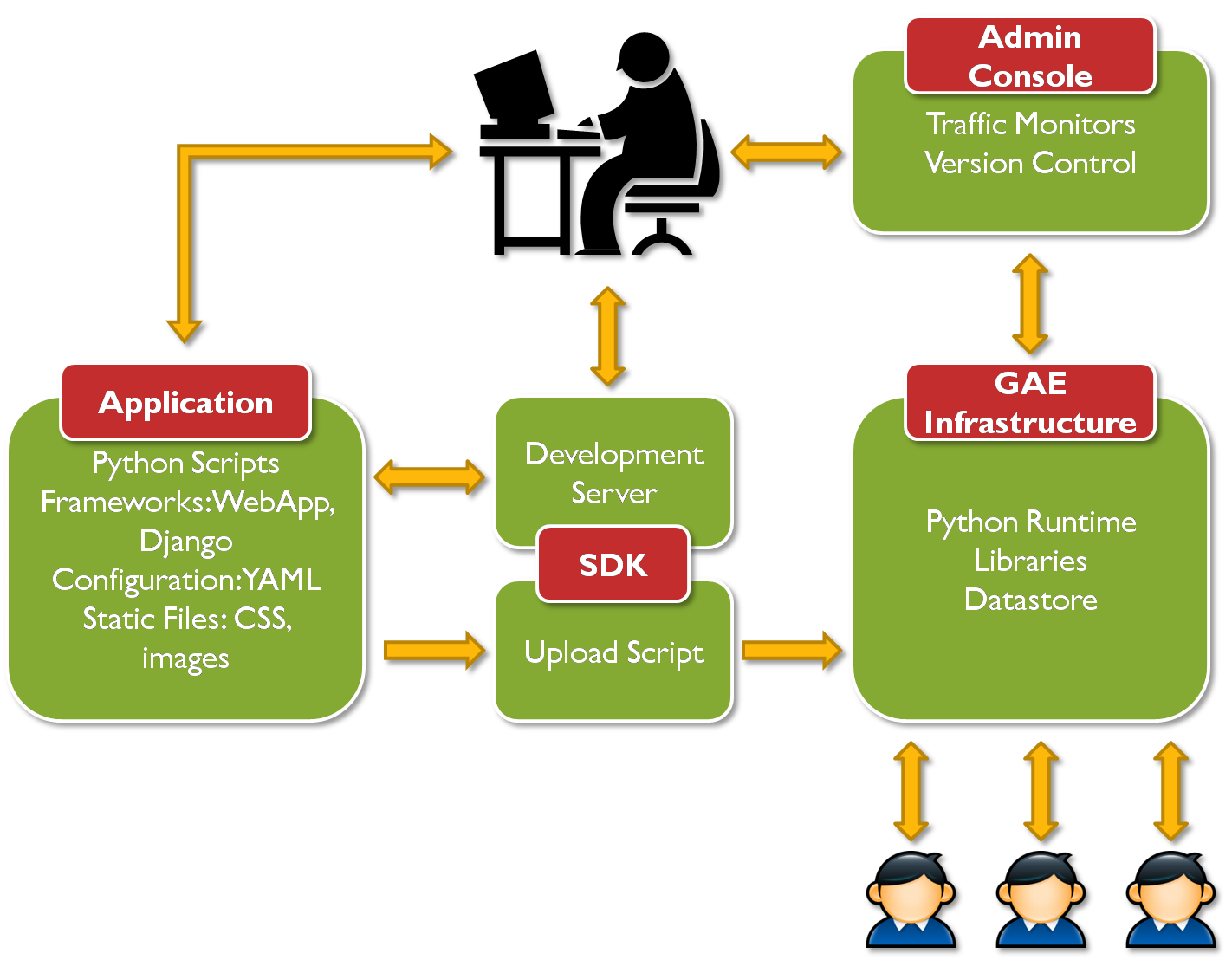
1. **Google App Engine**

## 4.1 Google App Engine 簡介

Google App Engine(簡稱GAE) 是由Google公司所提供的一項雲端運算技術，橫跨多台伺服器與資料中心，是一種分散式系統概念的應用程式，使用者註冊之後可以在有限的流量限制下免費使用，若想額外擴充流量才需要付費使用。

## 4.2 GAE運作原理

如下圖，開發者可在GAE上建構動態語言如:Python、Java(圖未標示)或靜態內容如:CSS、圖片，等Application，再經由Google所提供的SDK (Software Development Kit)，完成可供Client觀看的網站內容，開法者也可以從Google提供的Admin Console主控制台對其管理。



在本次專題中，我們使用Eclipse附掛GAE套件來開發，在安裝完Eclipse後，可至Google官方網址(goo.gl/8cJa4) 選擇Eclipse 版本進行安裝。本次專案，我們採用Java\JSP 開發GAE當作後端的資料庫應用。

## 4.3 NoSQL

GAE採用Google Big Table當作其資料庫，是一種NoSQL應用，而所謂NoSQL(Not only Structure Query Language)是使用**非**傳統的關聯式資料庫(RDBMS，Relational Database Management System)的資料庫，在GAE上，我們使用JDO(Java Data Object)的方式存取或讀取資料。

## 4.4 Eclipse開發環境

如右圖所示，專案主要可分為src與war兩大部分，src為前端網頁部分，我們使用JSP撰寫網頁外觀，首頁index.html也在此資料夾內，而war的部分則是Servlet，在GAE後端運行的Java code，可以運用物件導向程式設計來撰寫。

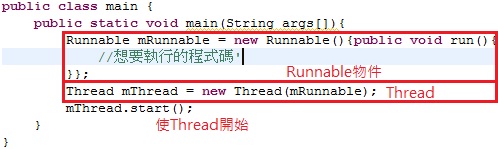
而後端的Servlet也可以和前端的JSP依靠GET和POST兩種方法來進行溝通，舉例來說，當使用者輸入帳號密碼時，可以從login.jsp送出POST表單到src裡的LoginServlet.java執行，來完成使用者的需求。

1. **Android 程式設計**

## 5.1 Multithread programming

1.Thread

在撰寫Android這種提供高使用者體驗的程式時，Multithread是不可或缺的coding技巧，因為總不能使用者在使用每個功能時，都需等此功能在Main thread跑完之後，再繼續使用，因此，什麼時候要開額外的thread，thread數量的掌控，每個thread之間都溝通、傳值便成為很重要的課題！

在Android裡是使用Java內建的thread來達成multithread，Java的使用及原理其實簡單：就是宣告一個thread物件，並在thread中加入Runnable的物件，並讓其開始，程式就會再額外開一個執行續並執行Runnable內的程式碼，此新的執行續會跟Main Thread快速的切換，以致使用者看起來所有功能是同步在進行的，可以大大提升使用者的體驗。****

2.Handler

然而在寫Android，單單會用Thread是不夠的，因為Android基於系統安全的機制，無法在Thread內更改系統的畫面，且若只用Thread便無法處理多個Thread要互相溝通、傳值的問題了，此只Java內建的一個物件也派上用場，就是Handler，Handler的使用方法也很簡單，便是在要接收訊息的Thread宣告一個Handler物件，在裡面的handleMessage function內放入想針對不同收到的值所做的反應，然後把Handler丟到其他的thread裡，當要傳訊息時執行sendmessage將寫好的message物件傳出，就完成一個簡單的傳值，若要更改主畫面的做法就是在Main Thread 的Handler裡放置要更改主畫面的code，thread裡傳值來通知執行，這種方法來實現。

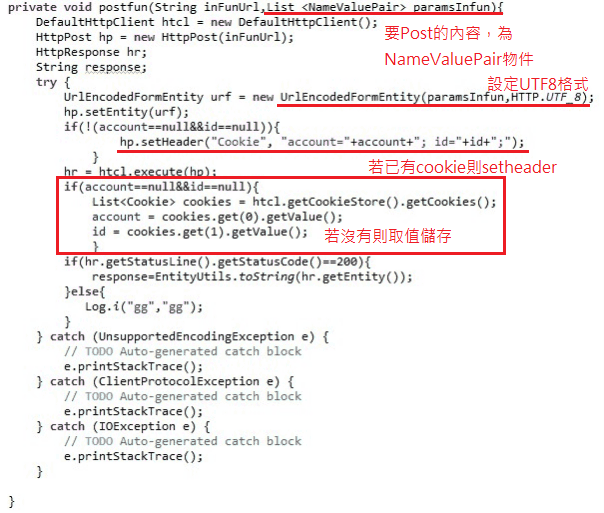
****

然而，雖然兩者的使用方式都很簡單，若在使用上沒有先很清楚的想好邏輯，便會造成很嚴重的錯誤，像是Deadlock的問題，便要Debug很久，另外Thread的數量掌控上也要有技巧，程式若是產生太多Thread可能會擠壓到Main thread造成使用者體驗不佳！

## 5.2 Apache HTTP套件

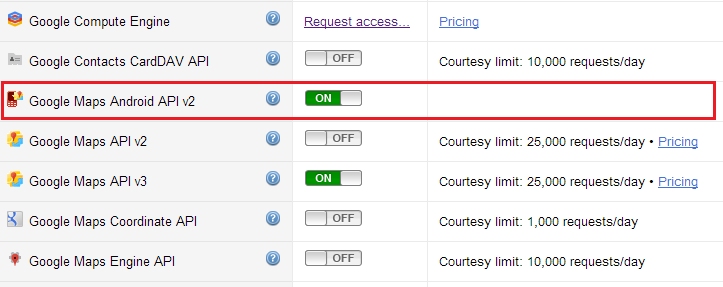
傳統撰寫網路程式通常都是用Socket programming，我們的程式當然也有用到網路功能，但因為我們是用GAE當Server，不支援使用socket等方法寫網路程式，另外我們也注意到了，現在越來越流行用http的溝通取代Socket programming，其優勢也很多，使用http套件進行application的溝通，能減低很多Server跟Client coding的負擔，傳值就到指定的網址使用HttpGet或是HttpPost，跟Server取值就到指定的網頁取值，不但減少了很多在connection內做判斷的步驟，邏輯上也變得清楚又鮮明很多！

而在認證的部分，則是在login之後將Server認證完封包的set-cookie header裡面的cookie內容記錄下來，之後每次向server不論發送post或是get的封包，只要先setheader將之前cookie的內容放進去，server便知道你的身分。

****

## 5.3 Android Google Map

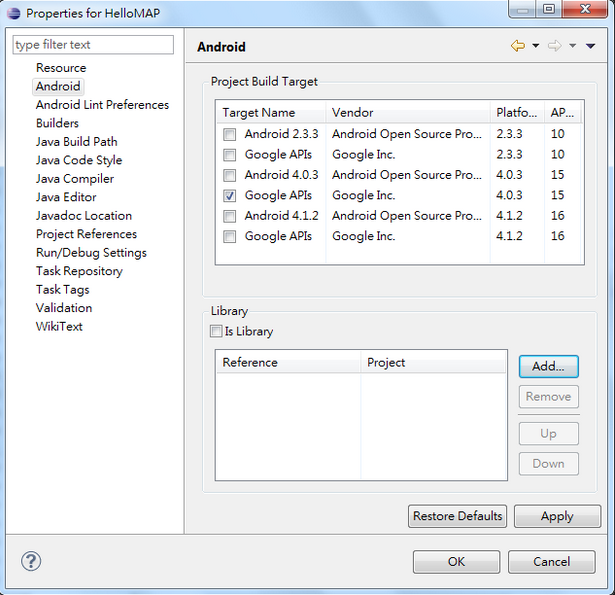
Google map是這次專題的主角，整個Project的重點就是在Google Map的呈現，Android上要使用Google map有兩種方法，一是在網頁上用Javascript寫Google Map，然後在手機端用一個Web view呈現，第二種方法是使用Android Google play service搭載的Google Map物件來使用，我們是使用第二種方法，而第二種方法的版本現在也來到了Version.2，要使用前要先到官方的API console申請API key

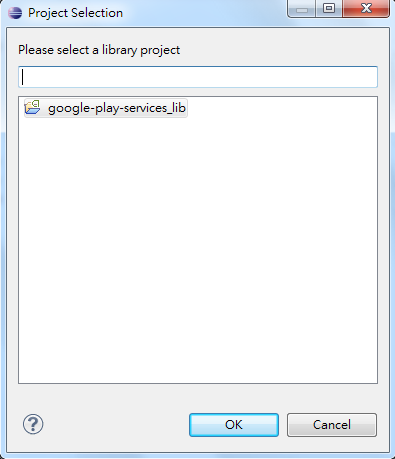
****

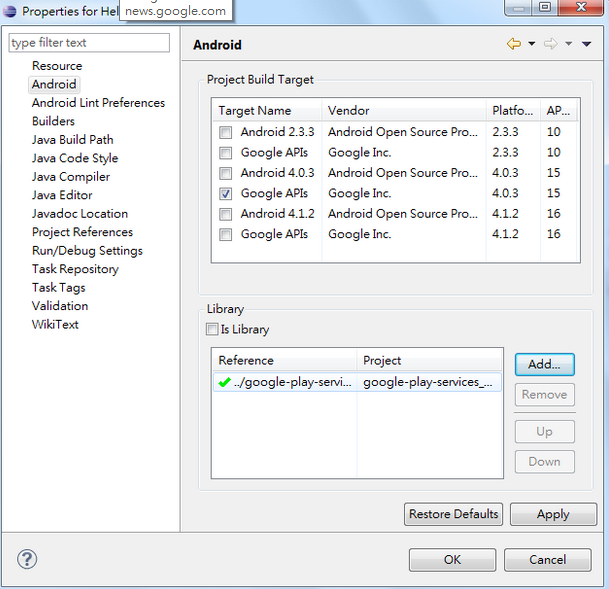
**然後在電腦command程式中輸入**keytool -list -v -keystore **"**這裡開Eclipse --> Windows --> Preferences --> Android -->Build 中找到Default debug keystore**"** -alias androiddebugkey -storepass android -keypass android

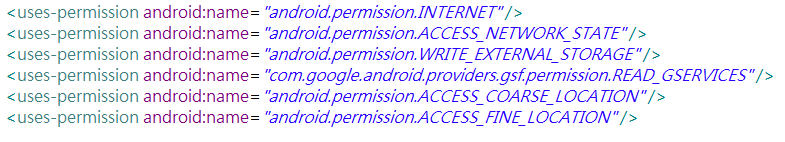
**指令便可取得SHA1的key，便可拿到Google API console裡去申請，然後因為不同台電腦所產生的key是不一樣的，要在不同的電腦上開發，也要申請不同的key。**

之後要利用Android ADT更新取得google-play-service取得後在eclipse的project裡inlcude此library，之後在程式內就可以使用Google map物件。

****

****

****

Google Map是一個Web service，在Android的使用方法是遷入Google Map的Fragment，裡頭描述好Web service的目標位置，並在AndroidManifest.xml裡描述好權限及key，便可在Android上控制Google Map。****

在主程式中利用findViewById將讓GoogleMap物件取得View上的Fragment並加以控制，其能對Google Map的修改空間非常大，包括：在地圖上畫軌跡，適用overlay覆蓋在地圖上改變地圖樣式，移動目前顯示的位置，改變單擊、雙擊與長按Google Map上所觸發的事件，用法之多讓我們能夠寫出千變萬化的Google Map應用。

## 5.4 SQLite

Android的程式內，若希望記住一些使用者的資料在本機內，選擇不外乎就是SQLite這個輕量的資料描述語言，它是以c編寫的，大部分的平台都有支援，並且執行的速度非常快，且非常大的優點就是，它是以檔案的方式呈現，使用者本地端不須額外建立一個database，只要程式內產生SQLite的檔案，每次程式開始時與此檔案溝通即可。

使用方式是寫一個物件繼承SQLiteDBHelper，並在onCreate裡寫好要創建的db以及table的內容，之後在其他class裡把這物件新增出來便可對它控制的database做一些select、insert或是delete的動作

****

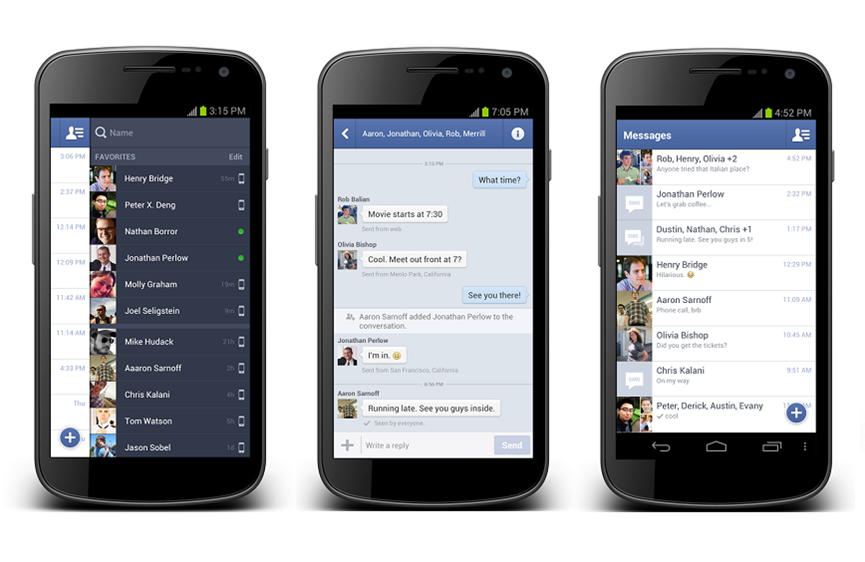
在使用SQLite之後程式Debug會比較麻煩，因為若像是table被drop掉了但是SQLite的檔案還在，程式在query table時便會直接crash，此時的解決辦法是進入Android設定內，找到寫的Project並清除資料，如此一來再開啟Project時就會重新onCreate一次。

****

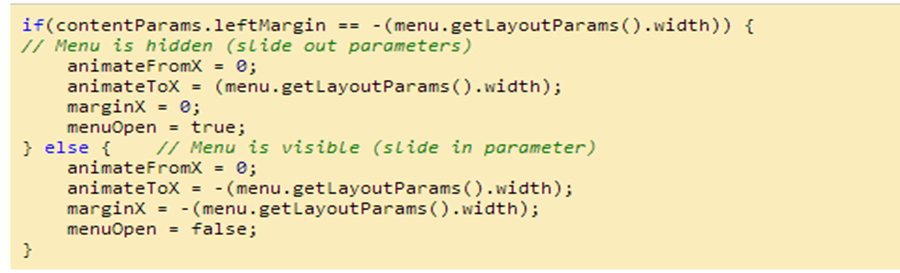
## 5.5 Sliding Menu

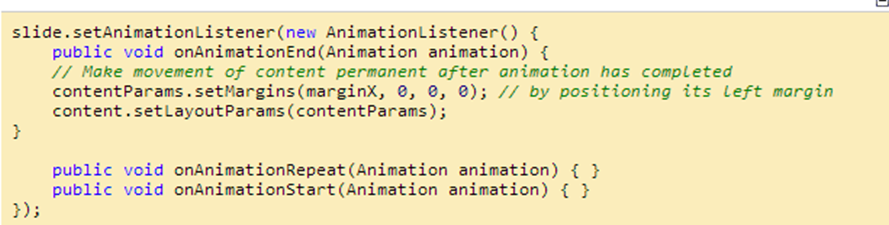
基礎的layout 呈現方式是經由不同的activity間去切換，也就是會彈出不同的頁面，每個想要新增如同Facebook一樣具有左滑又滑的頁面需要用到android 所提供的Fragment ，並在填充Fragment的時候利用動態的method “animation” 去連結。過程繁瑣，但大致上不複雜，

Facebook 頁面滑動的範例如下:



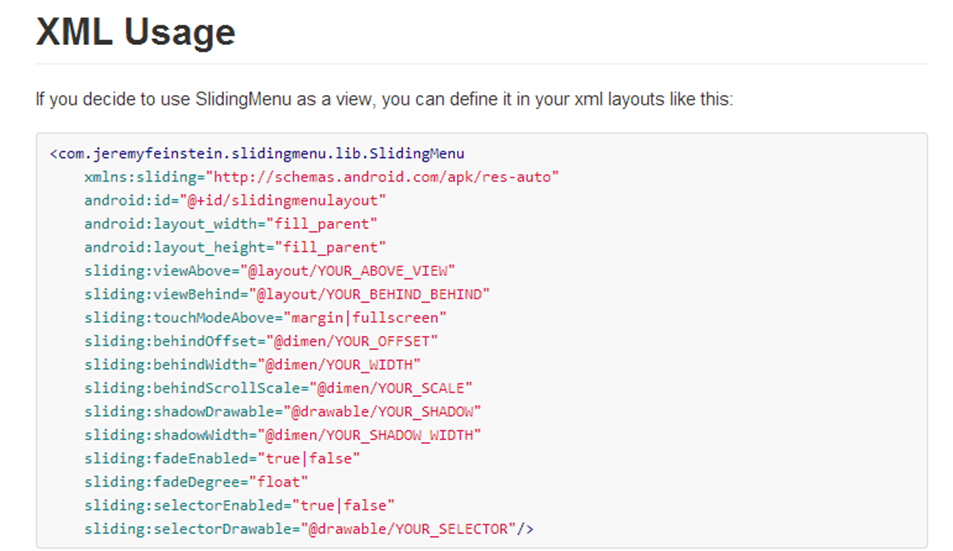
在實作的時候，原始做法如下:





上述的方法，只是個簡單的實作，若要更美觀的樣貌也可以import上架且實作好的Android Sliding Menu Library。我們這次使用到的library便是由jfeinstein10所製作提供的Sliding Menu。

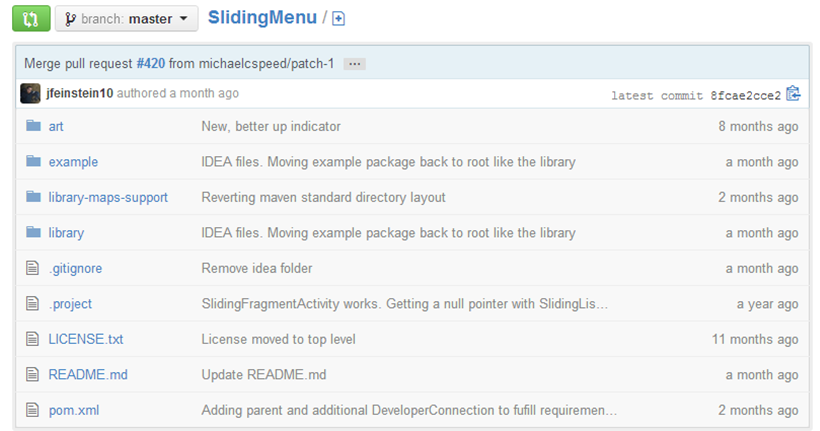




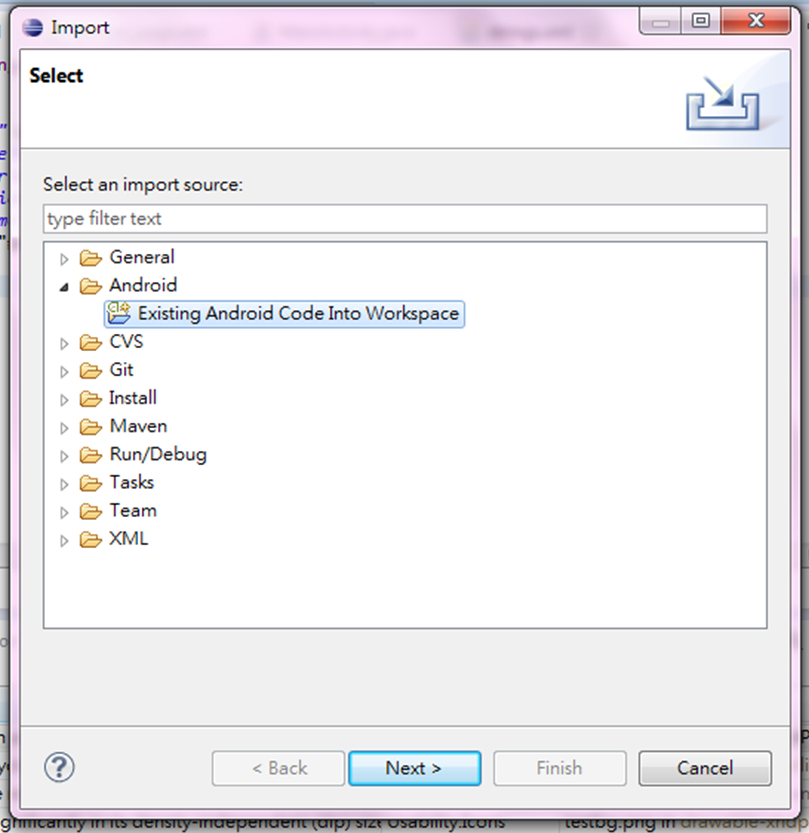
在實作的時候也是以物件的觀念去思考頁面，所以用法簡單，但是若對Android的程式設計不熟悉，可能會陷入自己心理的Bug，而對這套評價優質的Library有誤解，這是這次應用這個Library最常發生的事，譬如說他提供的左滑頁面和右滑頁面在設定上時，如果你在Main Activity 填充layout的方式是直接設定 menuLeft ＆ menuRight完後再填充頁面，結果最後只有menuRight 所負責的頁面顯示。但這其實是有敘述上的錯誤，因為我們在填充layout 的時候，沒有明確的傳送layout的指定頁面，而Sliding Menu 會自動幫你使用預設inflate 方式，所以才會產生這個結果。

以下是套用Sliding Menu Library 的用法，

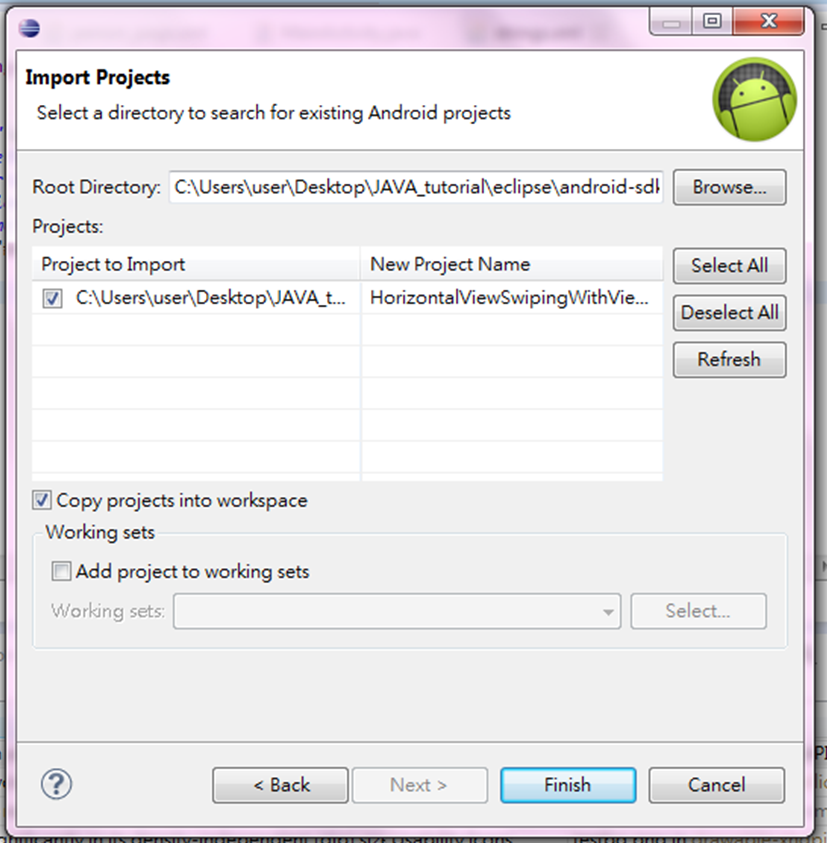
1. 下載 Sliding Menu Library <https://github.com/jfeinstein10/SlidingMenu>



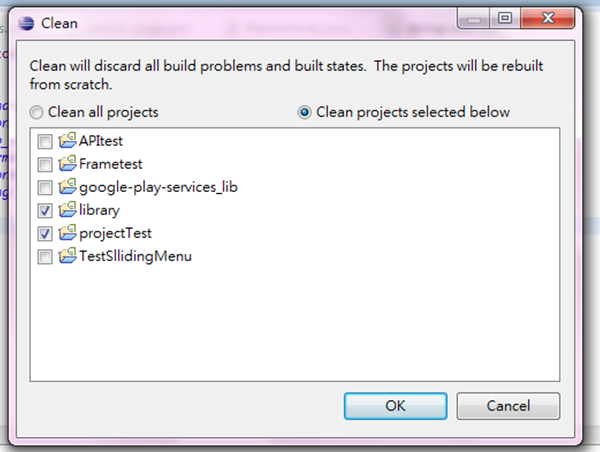
1. 開啟eclipse → import → Existing Android Code Into Workspace



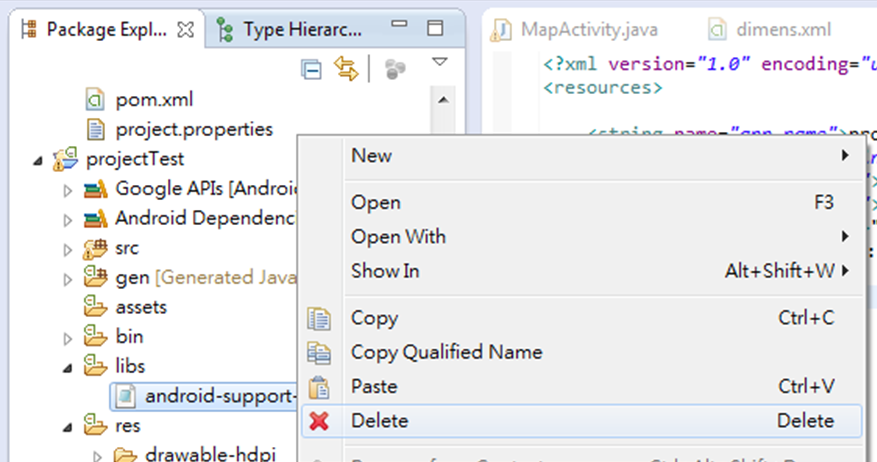
1. 找到Library 位置，記得勾選Copy Into Workspace.



1. 在eclipse 上面使用 project → clean 然後勾選你的project和Library.

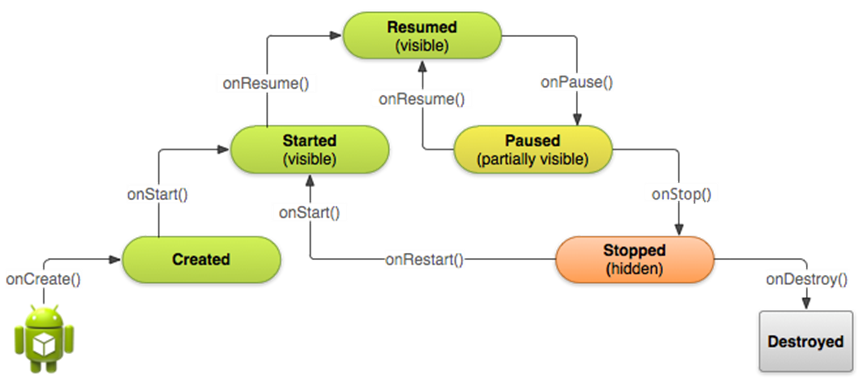


1. 刪除project下libs的android-support-v4.jar檔 (大部分人無法使用的原因是因為沒有刪除這個重複的lib，請務必檢查一下)

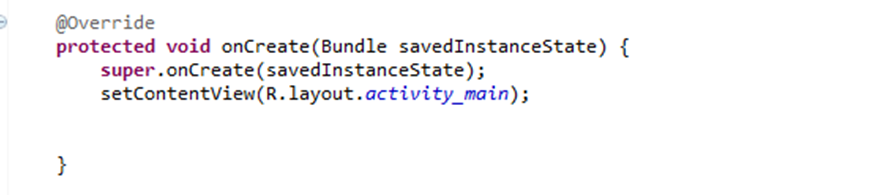


## 5.6 Android Life Cycle

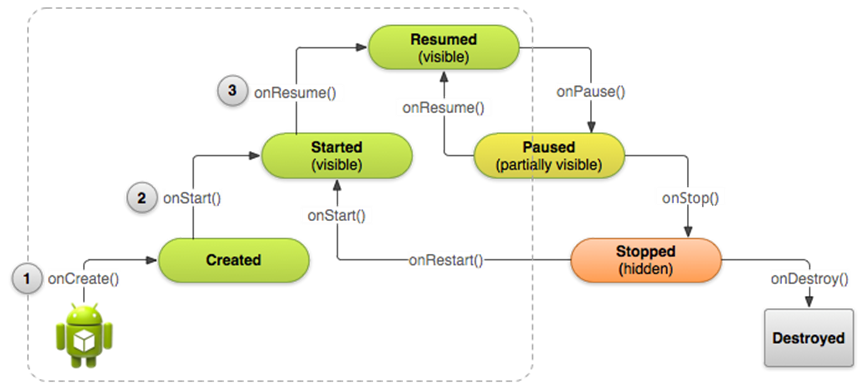
Android Life Cycle :



我們在撰寫android的code的時候，程式碼通常是打在onCreate()這個method裡。



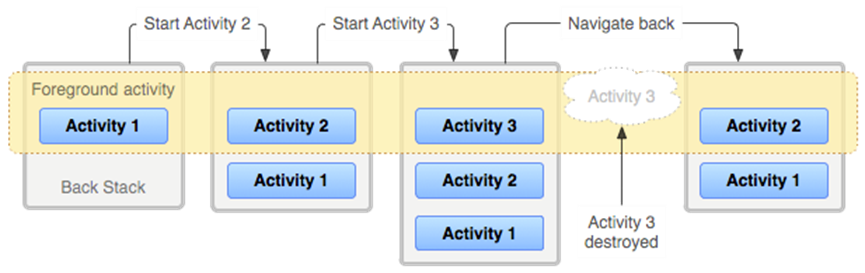
我們使用override，這個是因為在程式開始執行時，除了原本必須做的事情之外，還要外加那些程序，譬如說上述的setContentView()，便是指我的Program執行成process之後，我的第一個頁面要顯示activity\_main。



經過onCreate() 之後，程式會直接進入onStarted()和onResumed()，如果要對應到程式上的實作的話，其實看不出來，我們的畫面會直接設定好並運行，若要觀察這個部分的切換的話，可以使用Android提供的程式觀察(goo.gl/uOS9f)

　　比較值得注意的地方在於，我們這次有使用到Google Map，地圖上的標點和內容物，在切換不同頁面的時候(譬如說Activity間的切換)，必須要做BackStack的動作，因為onPused()到onStopped()的過程中並不會儲存你已經設定好的資料與操作，所以當我們從Google Map上標上了一些座標點的時候，不小心轉到螢幕的方向(ex:橫放到直放)，剛剛標定的點，便會全部消失。

以下是對於BackStack的圖示:

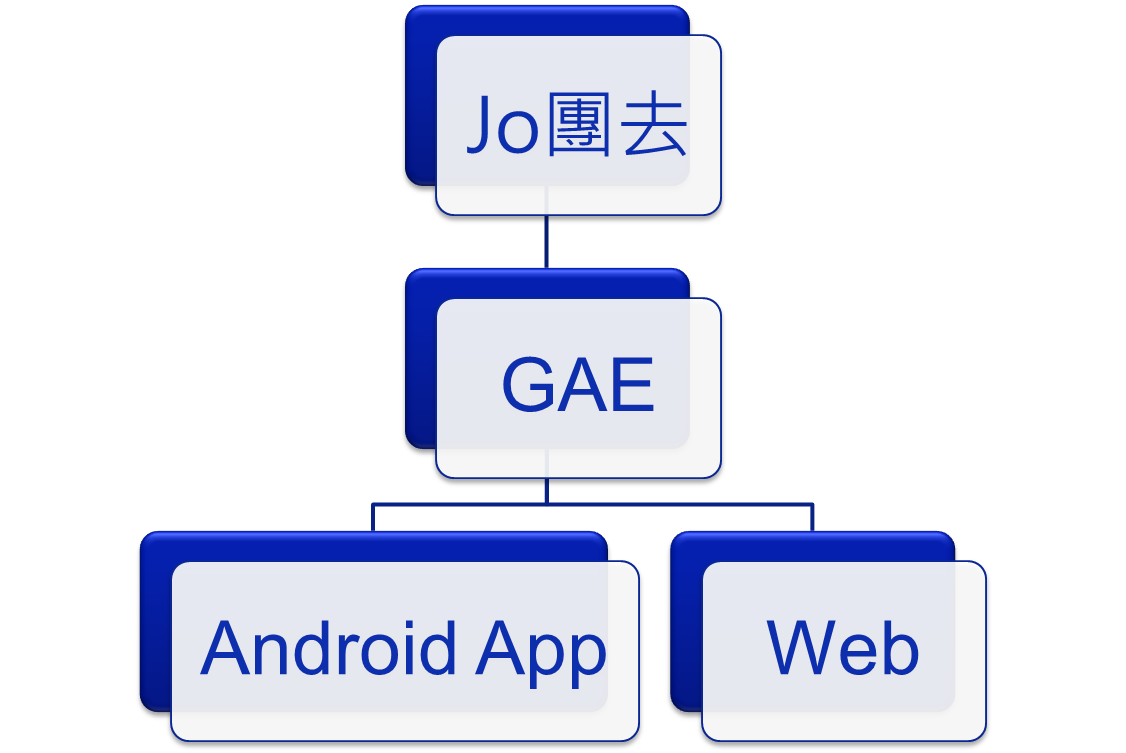


實作方法，使用的是onSaveInstanceState():

參考網址: <http://developer.android.com/reference/android/app/Activity.html>

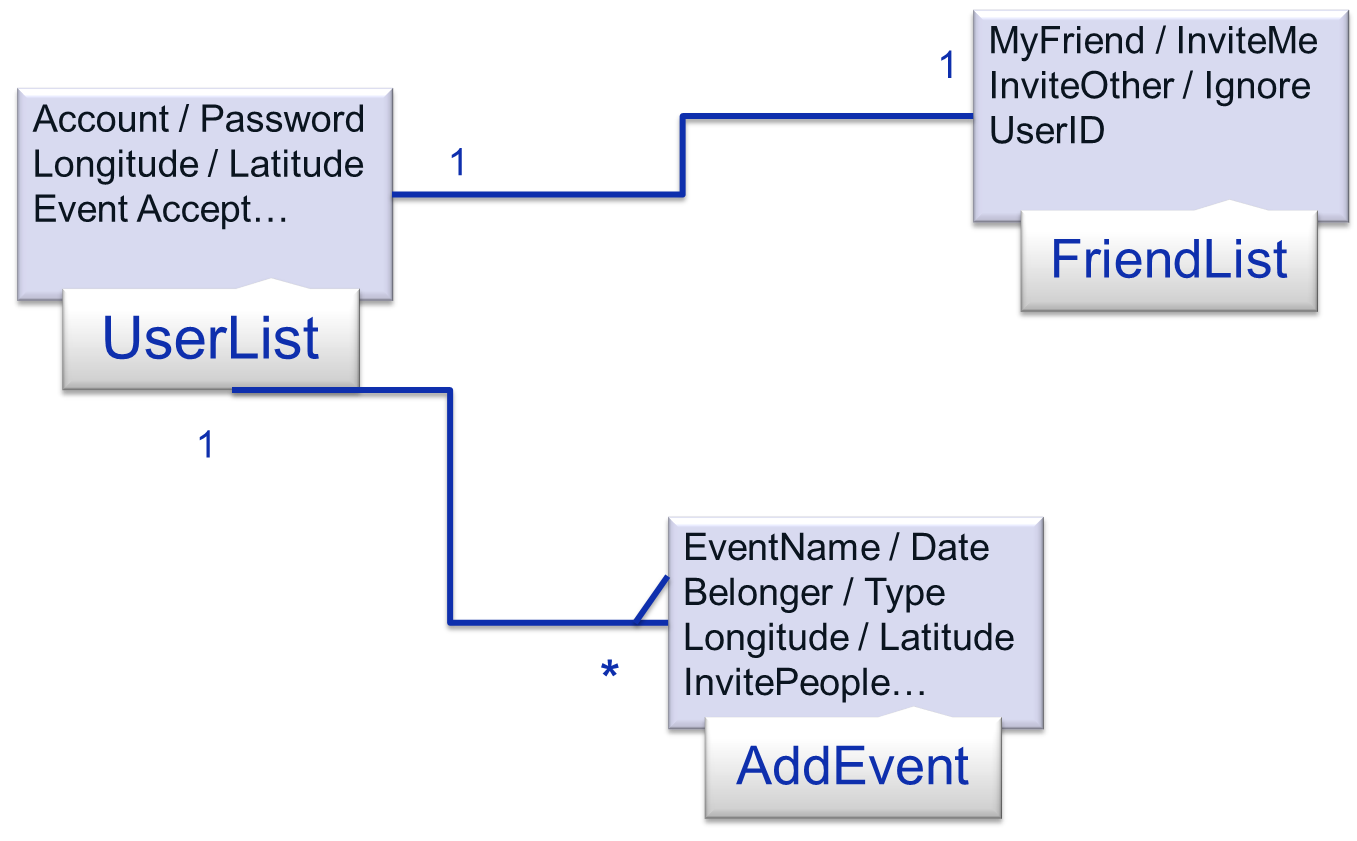
1. **系統架構與整合**

## 6.1 系統架構

如下圖，Jo團去App共分成後端的GAE和前端的Android和Web板的使用，後端GAE是我們用來存放使用者資訊、事件資料的Database，玵前端分為Android和Web，Android為我們現在所看到的App，另一個Web服務是用來模擬App的端口，因為在開發過程中，Android Programing和Database進度不同，所以為了開發GAE，先用了Web模擬App行為，如：送出表單，除此之外，Web服務也提供了管理員控制台，能夠追蹤每一筆資料的運作情況。 

## 6.2後端結構

在GAE這邊，我使用了三張Table存放所有資訊，分別是UserList存放使用者資訊、FriendList存放使用者的好友、和AddEvent存放主辦者和參加者的事件資訊，其中UserList的主鏈被FriendList存放在UserID這個Column中，且是一對一的關係，每一位使用者搭配一筆FriendList，另一方面，AddEvent中的Belonger也一樣被用來存放使用者主鏈，但在UserList中，Event Accept是為一個List用來存放答應要參加的Event，包刮自己辦的活動或是別人邀請的活動，把會參與的所有事件ID存放在此List中。



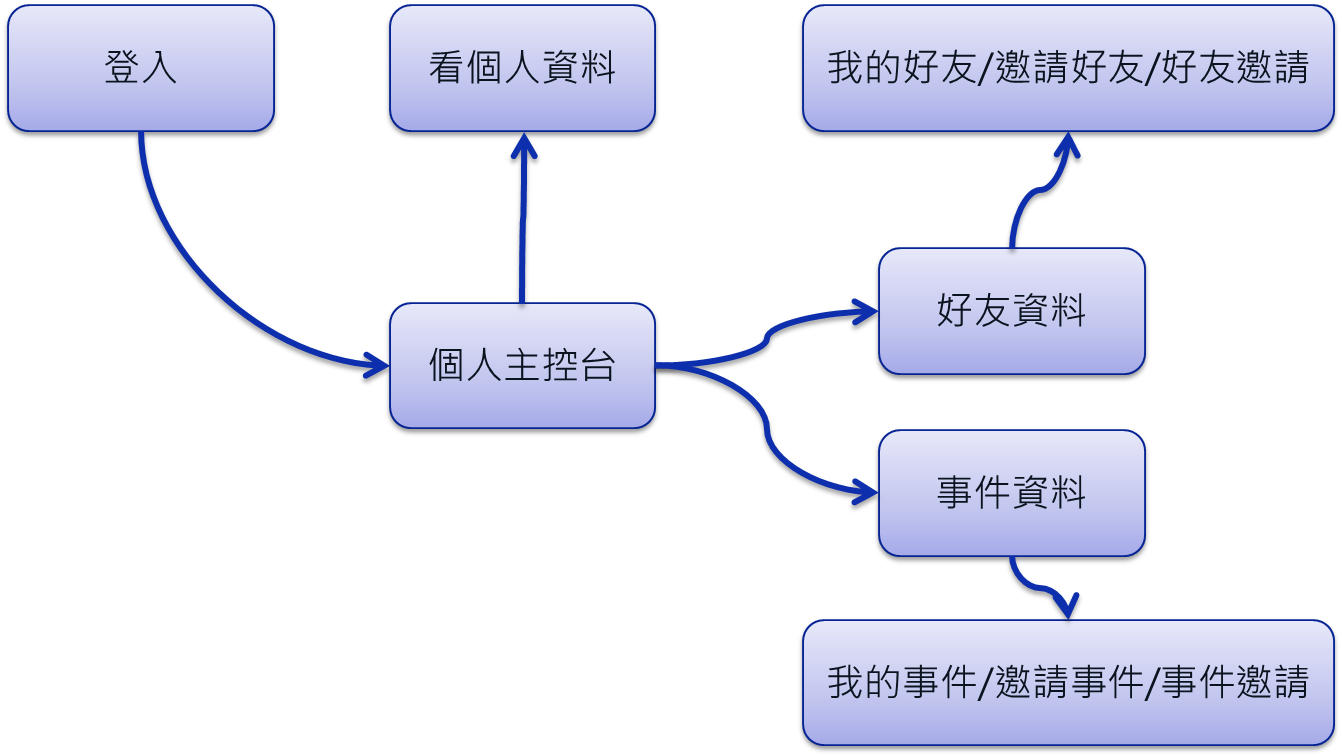
完成後端的資料庫後，我每筆資料我都建立了不同的Json Page讓Android能夠獲取資料，所謂Json指的是相對於XML比較輕量化的資料格式，因為具有統一的表現方式，所以可以讓很多前端Application辨識，如下圖是我們使用者的Json Page



辨識方法是一組變數的名稱會對應其值，如”Account”:”jason”所代表的意思即Java的String Account = “Jason”。

## 6.3前端結構

此為我們前端的架構簡圖，只要使用者登入，即進入主控台首頁，使用者可決定查看或修改自己的個人資料，或者進入好友選項，查看有無好友邀請或者邀請他人為好友，另外也可以進入事件資料，查看已參加活動或其他活動的邀請。



1. **結論與未來展望**

本次專題收穫很多，從最一開始完全對自己該準備的領域不了解，到能夠和組員一起溝通學習，建立程式的連結，我們學會了Android的程式撰寫，GAE的NoSQL資料規劃。但是在最後有crash的bug，我們了解自己所犯的錯誤，也就是太早規劃程式的視覺效果，而引用網路上別人的library，但卻沒有對別人的library做深入的了解，所以使用了很多有問題的程式碼，導致程式當機，資訊界的大師Knuth有一句名言：「premature optimization is the root of all evil」（過早最佳化是萬惡的根源），這是我們這次學到最深痛的教訓，我們應該先把最基本的功能寫好，再去追尋效能和視覺效果。

未來我們有下列幾點規劃:

1. **先將目前所遭遇到的bug移除**，目前程式還會不定時的crash，將程式穩定是第一步。
2. **結合gmail和Facebook**，要讓使用者在註冊一筆新的帳號密碼，可能會降低使用者的意願，如果能讓使用者依靠Google或Facebook當作認證會是最有效率的做法。
3. **如同instaWeather(APP)，可將Event直接貼到FB塗鴉牆**，輕鬆按下一個按鍵就能在貼到FB上塗鴉牆，是一個深具人性化的功能，並達到網路最重要的目的 - 分享。
4. **參考資料**
5. Google GAE tutorial  
   <https://developers.google.com/appengine/docs/java/>
6. JSP Tutorial  
   <http://www.jsptut.com/>
7. Global Positioning System – Wikipedia<http://en.wikipedia.org/wiki/Global_Positioning_System>