

致理技術學院

商務科技管理系

實務專題報告

Android 手機/平板電腦商家資訊系統

學生：張竣傑(19833106)

陳國霖(19833120)

林孟杰(19833145)

林偉民(19833146)

周明輝(19833147)

本成果報告書經審查及口試合格特此證明

指導老師：_____

中華民國 101 年 12 月

CTM 實務專題研究授權書

本授權書所授權之實務專題研究為 張竣傑、陳國霖、林孟杰、林偉民、周明輝 共 5 人，在致理技術學院商務科技管理系101學年度第 1 學期完成商管實務專題。

商管實務專題名稱：Android 手機/平板電腦商家資訊系統

同意 不同意

本組同學共 5 人，皆同意著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。

上述同意與不同意之欄位若未鉤選，該組同學皆同意視同授權。

指導教授姓名：

專題生簽名：

學號：

專題生簽名：

學號：

專題生簽名：

學號：

專題生簽名：

學號：

專題生簽名：

學號：

中華民國 年 月 日

誌謝

這一年來專題之所以能順利完成，首先要感謝我們的指導老師因為有老師長久以來的無私教誨以及耐心的指導，讓我們能夠更加快速地進入程式設計以及資料庫的領域中；從一開始對於程式的懵懵懂懂，專題的目標也不明確，到現在各組員能確實分工，並且規畫出一個完整的專題，老師總能在最關鍵的時刻點出最重要的問題，並且讓我們解決問題的所在；另一方面，老師不只是指導我們做專題，對於人生的規劃也會提出良好的建議，或是在生活大小事的方面也會給予我們很大的幫助，讓我們不只是在學業方面，在生活中也讓我們受益良多，真的是非常的感恩。

論文寫作期間，還要感謝各個系上的老師，從大一到大三為我們打好基礎，可以讓我們在專題加以應用；也感謝系姊以及系上的資源，在忙碌中還可以為我們處理很多的問題；最後也感謝各個組員的努力，每個禮拜總能撥空一起前來討論專題事項，並且總能完成每周規定的進度，才能讓我們的專題如期、如實、如質的完成，相信這一段美好的旅程，都會永留在我們的心中。

陳國霖、張竣傑、周明輝、林偉民、林孟杰

摘要

從古自今，地圖一直在人類文明中扮演重要的腳色，從以前航海時代使用的手繪地圖，到近代社會的紙本地圖，一直到最新科技的 Google Map，隨著科技的進步，地圖也一直在演進，讓我想起電視廣告的一句話「科技來自於人性」，所以我們也一直在思考，思考如何可以讓 Google Map 的使用者更方便的使用，於是我們決定將 Google Map 結合 MySQL 資料庫，做出一個利用 Google Map 顯示商家資訊系統的 App，利用這一個 App 使用者不僅可以知道自己的位子，三五好友逛街之餘，也可以利用這個 App 搜尋附近的商家與美食，科技持續進步，智慧型行動裝置發展可以說是飛速成長，到了現在幾乎是人手一機，我們希望可以利用這個趨勢，讓我們一個好的 App 可以讓更多人使用。

關鍵詞：hiPage、Android、Google Map、MySQL、GPS 定位、Google Maps API、資料擷取

目錄

授權書.....	ii
誌謝.....	iii
摘要.....	iv
目錄.....	v
圖目錄.....	vii
表目錄.....	viii
第一章 緒論.....	1
第一節 研究背景.....	1
第二節 研究動機.....	2
第三節 研究目的.....	3
第四節 研究流程.....	3
第五節 預期效益.....	4
第二章 理論基礎及文獻探討.....	5
第一節 Google Map 系統介紹.....	5
壹、Google Map 簡介.....	5
貳、Google Map 應用.....	7
第二節 Android 介紹.....	10
壹、Android 簡介.....	10
貳、Android 特性.....	10
參、Android 市占率.....	11
肆、Android 架構.....	12
第三節 MySQL 資料庫介紹.....	14
壹、MySQL 簡介.....	14
貳、MySQL 特性.....	14
參、MySQL 應用.....	15
肆、MySQL 架構.....	16

第三章	系統功能與特色.....	21
第四章	使用對象.....	22
第五章	開發工具與建構流程.....	23
第六章	軟體架構.....	25
第七章	系統畫面.....	28
第八章	結論與未來發展.....	33
參考文獻	34
附錄 A	系統程式碼.....	36

圖目錄

圖 1-1 研究流程圖.....	3
圖 2-1 IPeen 愛評美食旅遊網.....	8
圖 2-2 台灣美女地理雜誌.....	8
圖 2-3 客運即時動態查詢.....	9
圖 2-4 市占率比例圖.....	11
圖 2-5 Android 架構圖.....	12
圖 2-6 MYSQL 架構圖.....	16
圖 6-1 軟體架構圖.....	25
圖 7-1 軟體安裝檔畫面.....	28
圖 7-2 系統畫面.....	29
圖 7-3 系統畫面 醫院.....	30
圖 7-4 系統畫面 醫院.....	30
圖 7-5 系統畫面 加油站.....	31
圖 7-6 系統畫面 加油站.....	31
圖 7-7 系統畫面 餐廳.....	32
圖 7-8 系統畫面 餐廳.....	32

表目錄

表 2-1 Android 特性.....	10
表 2-2 MYSQ 使用的通訊協定.....	17

第一章 緒論

第一節、 研究背景

隨著科技日新月異的發展，許多發明與科技結合後演變出來的新型發明，使我們的生活更加便利，更讓我們能夠在第一時間精準的找到我們所需要的資訊，漸漸的兩我們生活密不可分。

近年來隨著行動裝置與行動裝置的應用程式發明，電子地圖可說是越來越火紅了，電子地圖是把普通地圖或地形地物等現狀資料，以數位化的方式記錄、處理後存放進數位化地圖資料庫在與衛星定位後以年為單位更新。

不像是傳統的紙本地圖只有街道名稱跟方位，電子地圖有提供街景、規畫路線、定位與導航，目前台灣地區陸上電子地圖資料庫，有道路、鐵路、機場、河流、湖泊、水庫、設施、港口、火車站、橋樑、都市界線及都會區等資料，還可以放上許多的地標、風景點、餐飲、娛樂、加油站、文教、醫院、金融機構等等之生活資訊，讓使用者能快速尋找到餐廳、旅館及其他服務地點、及所在位置的詳細街道地圖。

第二節、 研究動機

自 21 世紀以來，行動電話有了重大的改革，手機不再只是通訊的工具，他有了更多的功能，演變至今「智慧型手機」出現了，他功能的強大，也讓我們思考，我們該如何讓智慧型手機更智慧，更貼近我們的生活。

在時下的生活，我們總是喜歡三五好友聚聚，或是穿梭在大街小巷之內到處尋找美食，尋找新的驚奇，而在智慧型手機出現之後，我們更是希望把找到的驚嘆號，分享給網路上的好朋友之到，無時無刻總是希望可以讓大家知道，所以我們開始思考，不管是在尋找美食，或是互相分享位子，都可以讓這一切變得更簡單。隨著電腦技術的發展，越來越多的人開始依賴電腦，當 Google Map 的推出後，相關的地圖應用推陳出新，所以我們決定結合 Google Map 與 Android 平台，並且使用 MySQL 資料庫，製作出商家資訊系統。

第三節、 研究目的

現在人們對於休閒娛樂可說是越來越重視了，以往人們只能在紙本地圖上找到大概的方位，但是電子地圖可以提供精準的位置如果還是找不到店家的話可以用導航功能，可以快速又正確的指引到目的地還能夠顯示店家名稱、電話、營業事項、營業時間等等的店家資訊，店家也可以自行提供照片與優惠資訊來吸引消費者。

第四節、 研究流程

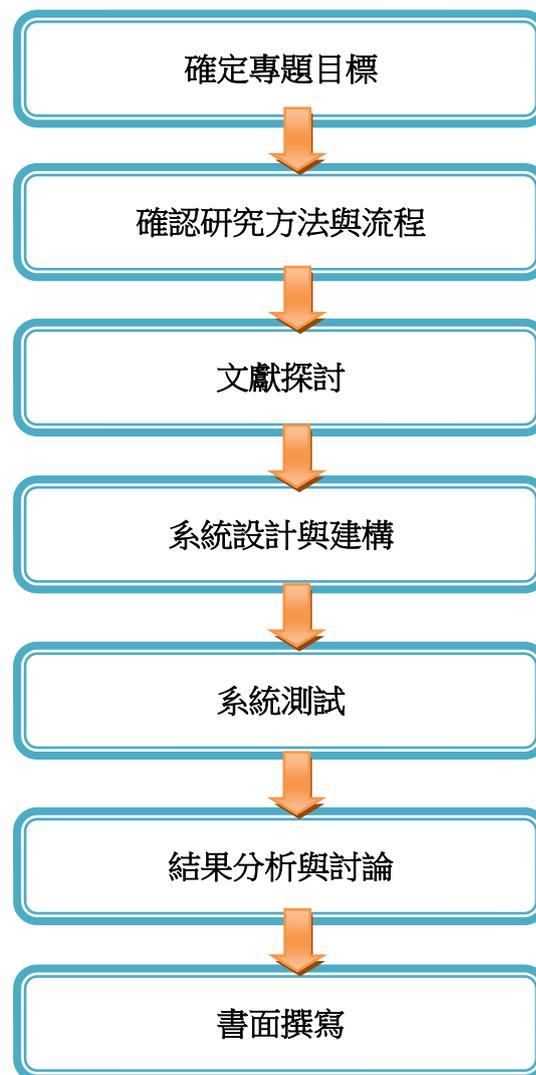


圖 1-1 研究流程圖

第五節、 預期效益

我們想要使用 Google Map 做出一個以商家資訊查詢的地圖應用程式，程式結合智慧手機普及、對於電子地圖的依賴與使用者喜歡用智慧手機進行拍攝記錄並上傳至社群網站的特性並且有助於將店家的知名度曝光而去吸引更多消費者以及提供多元服務取得使用者的青睞。

1. 建構一個完善的美食資料庫，台灣各地美食應有盡有。
2. 用戶能即時分享自己喜歡的美食並且透過 Google Map 精確快速的找到店家的位置。
3. 不僅是在手機上可以使用這個系統，在網頁上也能同時使用這個美食系統。

第二章 理論基礎及文獻探討

第一節、 Google Map 系統介紹

壹、Google Map 簡介

Google 地圖（英語：Google Maps），前稱 Google Local，是 Google 公司向全球提供的電子地圖服務，包括局部詳細的衛星照片。能提供三種視圖：一是向量地圖（傳統地圖），可提供政區和交通以及商業訊息；二是不同解析度的衛星照片（俯視圖或 45° 影像，跟 Google 地球上的衛星照片基本一樣）；三是地形圖，可以用以顯示地形和等高線。

Google 地圖中的向量檢視目前採用將向量資料按照製圖樣式預先渲染成影像並切割為瓦片（tiles），瀏覽器取得這些瓦片並按照其相互位置關聯在瀏覽內拼接成整體地圖的方法。此方案使得瀏覽器內使用者看到的地圖其實是柵格形式，道路、水體、建築、行政區等地理物件並不能被選擇互動，因此應用功能受到一定限制。2011 年 10 月，Google Maps 推出 WebGL 版本實作版本，試驗在客戶端即時渲染取自服務器的向量資料的技術方案。這是 Google 地圖發行以來在地圖應用架構上最大的變化。

它的姊妹產品是 Google 地球，一個電腦應用程式，在 3D 地球模型上提供街景和更多的衛星視圖及 GPS 定位（付費版本）的功能，但沒有前述的向量視圖和地形視圖功能。由於 Google 地圖是一項電子地圖服務，因而擁有比紙質地圖更多的優越性，特別是近年來流動裝置的智慧化，網絡速度的提升，更是讓 Google 地圖與人們的生活產生更為緊密的聯繫，因而倍受人們的讚譽。

爭議

- 2010年3月，香港版本使用了簡體中文及漢語拼音，引起香港網民的強烈反彈。
- 2010年4月17日起，Google地圖對全中文用戶，將全世界地名含街道名均改為簡體中文表示，引起香港及台灣用戶不滿。
- 2010年4月23日起，中國版Google地圖中大量的香港地名改用音譯，例如赤鱸角機場變成「切克拉普科克艾爾波特」。
- 2010年5月2日，Google地圖將中華民國宜蘭縣屬地的龜山島歸類為「日本沖繩縣」領土。

歷史

- 2005年6月20日，Google Maps的覆蓋範圍，從原先的美國、英國及加拿大，擴大至全球。
- 2005年9月，Google和北京圖盟科技有限公司（Mapabc）合作出版中國大陸的Google本地。後來，Google本地重新命名為Google地圖。台灣的圖資則是由勤崴科技所提供，並於2007年10月11日推出台灣版地圖。
- 目前Google Maps的美國、英國、加拿大、日本、法國等地區的服務已經是完成階段，亦在全球多國開通了「街景」（Street View）服務，使用者可以透過由Google 甲蟲車在街道上拍到的影像檢視街景（例如巴士站、大廈和商鋪等），但是該服務卻有侵犯個人私隱的嫌疑，街邊路人的動作可以看得一清二楚。
- Google地圖的香港版本於2008年7月推出，可以讓使用者搜尋餐廳，亦可以建立自己的地圖，與其他使用者分享有關的照片和評論。其他功能包括路線查詢，可以顯示兩個地點之間的距離和行車時間。

- 由 2011 年 10 月 7 日 開始 Google 地圖新增 3D 鳥瞰功能。

貳、Google Map 應用

(一) Platial 社群網站¹

Platial.com 一個利用 Google Map 的創新網站，透過地圖故事帶大家全球走透透，將住家附近的資料全都標示在 Google Map 上，要來訪的親友隨時上網查詢。還可將地圖印出來，不會再迷路。而上面標示的好玩地點也與日俱增，愈來愈豐富，是個以地點為主的社群網站，與 facebook 等無太大差異，就僅僅只多了一個特色——利用地圖。讓人們在網站上「存」(save) 一個地點，製作一個地圖，貼上一個標示 (mark)，分享對這個「地點」的情感。

(二) IPeen 愛評美食旅遊網²

將附近的吃喝玩樂的景點標記，並連結店家資訊，也可直接撥打店家電話。
結合

了 goole map 與無名小站的應用如下圖:

¹參考資料: <http://66jacky.blogspot.com/2006/06/platialcom.html>

²資料來源: <http://www.ipeen.com.tw/map/>

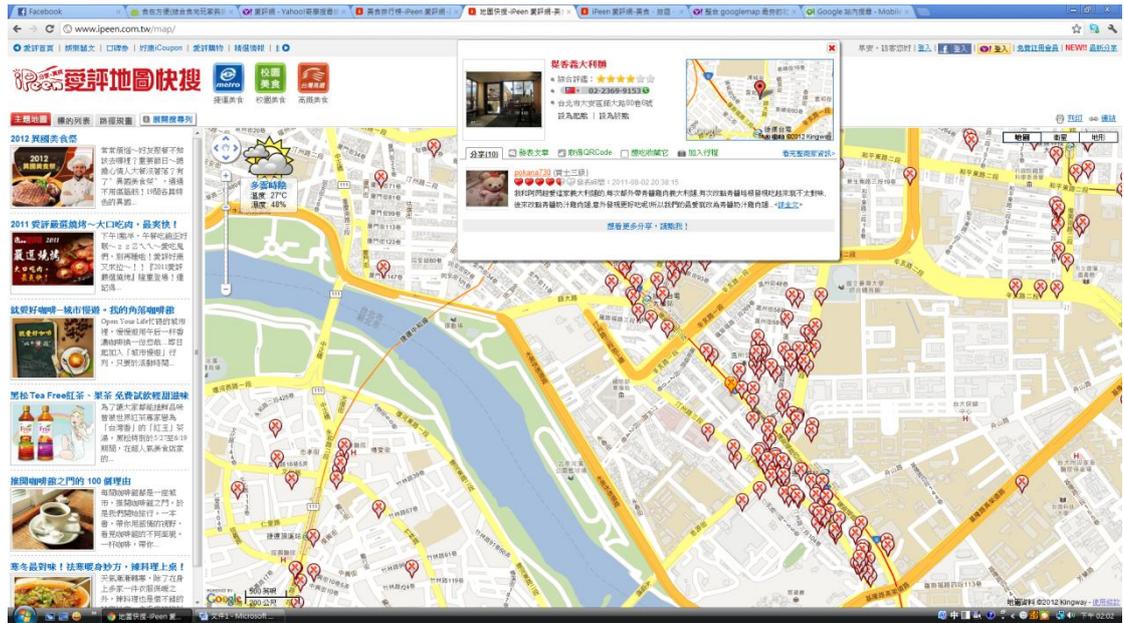


圖 2-1 IPeen 愛評美食旅遊網³

(三)台灣美女地理雜誌⁴

結合了 goole map 與無名小站的應用如下圖：

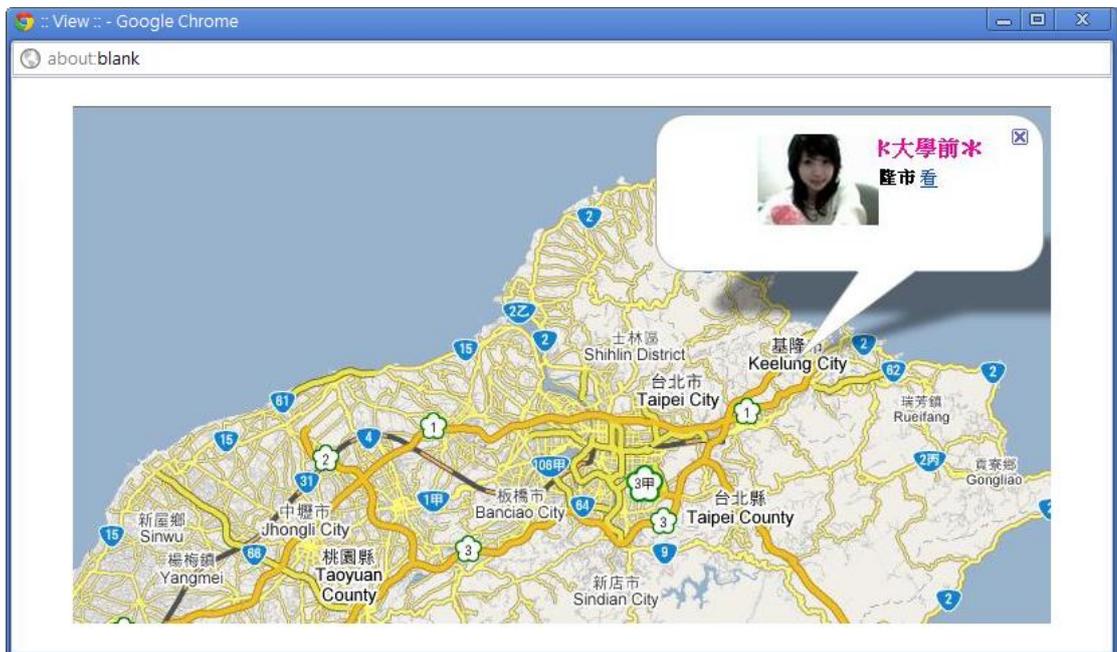


圖 2-2 台灣美女地理雜誌

(四)客運即時動態查詢⁵

³參考資料: <http://66jacky.blogspot.com/2006/06/platialcom.html>

⁴來源: <http://ssorc.tw/rewrite.php/read-514.html>

透過 google map 的應用客運業者可以將站名標記，顯示車輛的訊息



圖 2-3 客運即時動態查詢

⁵來源: <http://124.199.77.90/bus/Dybus.aspx>

第二節、Android 介紹

壹、Android 簡介

Android 是 Google 於 2007 年 11 月推出的開放性嵌入式作業系統平台，它提供了一個包含作業系統、中間層及應用程式的軟體堆疊架構。透過不同作業系統業者所提供的軟體開發工具包(Software Development Kit, SDK,) 以及高階的程式語言(如 Java 或 Objective-C)，讓應用程式開發者可以完全不需要瞭解手機硬體的內部構造與軟體語言便能自行發揮創意巧思去設計開發可安裝於智慧型手機之中的應用程式。

貳、Android 特性

Free	平台提供從底層作業系統到上層應用程式介面所有軟體，使用這個平台不用繳納任何授權費用。
SQLite	整合羽量及資料庫管理系統
多媒體支援	支援常見的音樂和視訊，及各圖片格式
內嵌瀏覽器	基於開放原始碼的 webkit 引擎
完備的開發環境	設備模擬器、除錯工具、記憶體、用於 Eclipse 套件
在應用程式溝通無界限	經由簡單的宣告，應用程式可以存取或呼叫其他應用程式的功能

表 2-1 Android 特性

參、Android 的市占率

在 2010 年第三季度末時，Android 已經佔據了美國市場的 21.4% 的份額。截止至 2011 年 6 月，Google 表示，每日通過 Google 服務器啟用的 Android 設備達到了 55 萬部，並且以每週 4.4% 的速度增長。2011 年 8 月 1 日，Canalys 的數據顯示，Android 已佔據美國 48% 的智慧型手機市場佔有率。2011 年 10 月 13 日，Google 表示，全球市場上有 1.9 億部 Android 設備通過 Google 認證，而在 2011 年 11 月 16 日，全球市場上已有 2 億部 Android 設備通過 Google 認證。但是，搭載 Android 作業系統的平板電腦卻只有 1.9% 的份額，只有 380 萬 Android 蜂巢平板被賣出，遠遠低於蘋果公司的 iPad 的銷量。2012 年 6 月，Google 在 2012 Google I/O 大會上表示全球市場上有 4 億部 Android 設備被啟動，每天啟動一百萬台

智慧型手機平台Q2市占率		
平台	代表廠商	市占率(%)
Android	摩托羅拉、宏達電、三星、索愛、LG	43.4
Symbian	諾基亞	22.1
iOS	蘋果	18.2
RIM	RIM	11.7
Bada	三星	1.9
微軟	宏達電、三星	1.6

註：其他作業系統約占1%
資料來源：Gartner
方巧文/製表

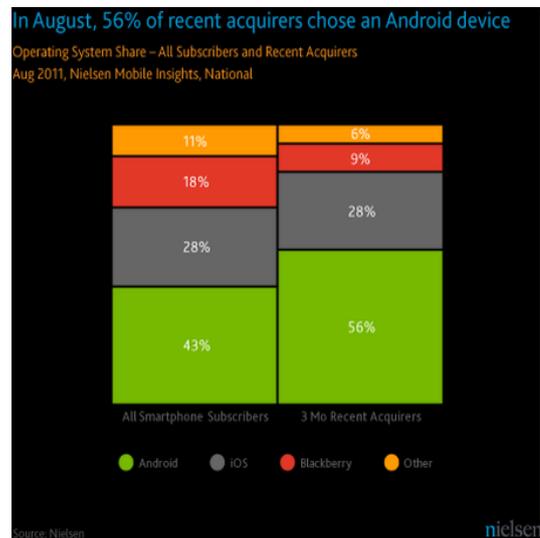


圖 2-4 市占率比例圖

肆、Android 架構

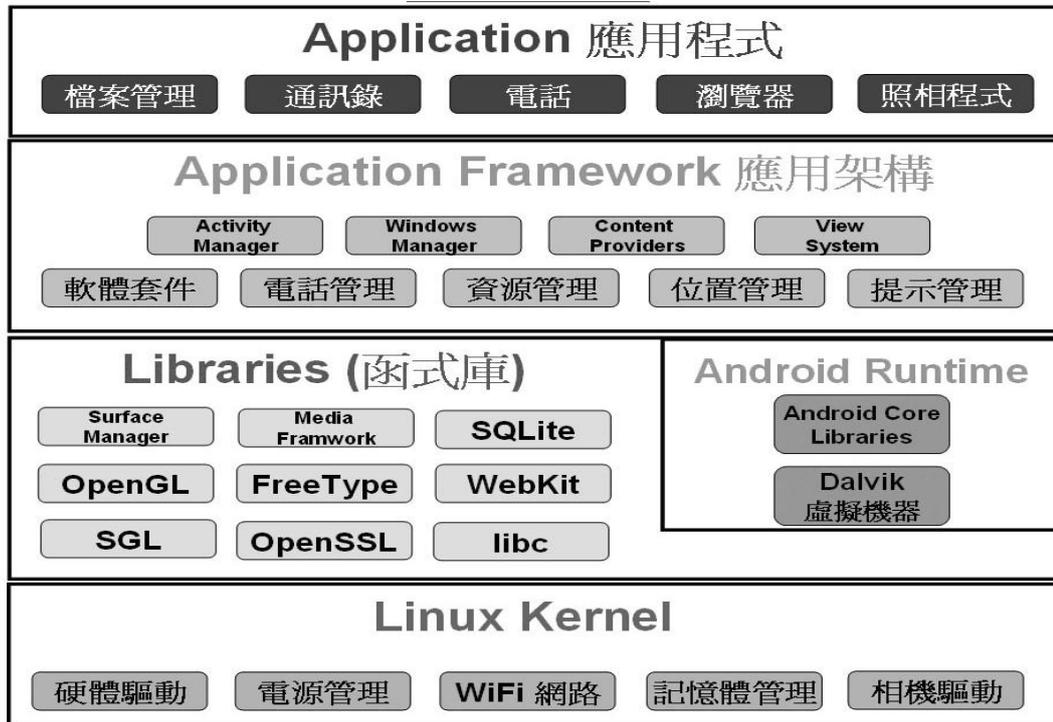


圖 2-5 Android 架構圖

(一)應用程式

- 以 Java 程式編寫
- 包含各種資源文件（放置在 res 目錄中）
- 編譯後會生成一個 APK 包。
- 內建 聯絡人 (Contact)，電話 (Phone)，瀏覽器 (Browser) 等應用。

(二)應用程式框架

- 為了節省開發者撰寫應用程式的時間、加速應用程式開發的速度，所以在應用程式與內部系統環境之間設置了一個讓程式更簡易溝通的界面，它稱之為應用程式框架(Application framework)
- 應用程式的開發者來說，他得以透過 Java 程式呼叫應用程式框架所提供的應用程式界面(Application Programming Interface, API)，來利用更底層函式庫(Libraries)提供的功能。
- 提供應用程式所需的軟體元件。

(三)函式庫

- 多媒體框架 (MediaFramework)：影音播放與錄製。
- SGL：2D 圖像引擎。(google 自己寫的)
- SSL：即 Secure Socket Layer 位於 TCP/IP 協議與各種應用層協議之間,為數據通訊提供安全支持。
- OpenGL ES 1.0：本部分提供了對 3D 圖像的支持。
- 界面管理工具 (Surface Management)：本部分提供了對管理顯示子系統等功能。
- SQLite：一個通用的嵌入式數據庫
- WebKit：網路瀏覽器的核心
- FreeType：字型函式庫。
- libc：C runtime 函式庫。

(四)Android 執行時期環境

- 顯示驅動(Display Driver):常用基於 Linux 的圖像緩衝(Frame Buffer) 驅動。
- Flash 內存驅動 (Flash Memory Driver)
- 照相機驅動(Camera Driver):常用基於 Linux 的 v4l(Video for Linux) 驅動。
- 音頻驅動 (Audio Driver)：常用基於 ALSA (Advanced Linux Sound Architecture，高級 Linux 聲音) 驅動。
- WiFi 驅動 (Wifi Driver)：基於 IEEE 802.11 標準的驅動程式
- 鍵盤驅動 (KeyBoard Driver)
- 藍牙驅動 (Bluetooth Driver)
- Binder IPC 驅動：Android 一個特殊的驅動程式，具有單獨的設備節點，提供程式之間通訊的功能。

第三節、MySQL 資料庫介紹

壹、MySQL 簡介

MySQL是一個開放原始碼的關聯式資料庫管理系統，原開發者為瑞典的MySQL AB公司，該公司於2008年被昇陽微系統（Sun Microsystems）收購。2009年，甲骨文公司（Oracle）收購昇陽微系統公司，MySQL成為Oracle旗下產品。

MySQL在過去由於效能高、成本低、可靠性好，已經成為最流行的開源資料庫，因此被廣泛地應用在Internet上的中小型網站中。隨著MySQL的不斷成熟，它也逐漸用於更多大規模網站和應用，比如維基百科、Google和Facebook等網站。非常流行的開源軟體組合LAMP中的「M」指的就是MySQL。

貳、MySQL特性

- 使用 C 和 C++編寫，並使用了多種編譯器進行測試，保證源代碼的可移植性。
- 支援 AIX、BSDi、FreeBSD、HP-UX、Linux、Mac OS、Novell NetWare、NetBSD、OpenBSD、OS/2 Wrap、Solaris、Windows 等多種作業系統。
- 為多種編程語言提供了 API。這些編程語言包括 C、C++、C#、VB.NET、Delphi、Eiffel、Java、Perl、PHP、Python、Ruby 和 Tcl 等。
- 支援多線程，充分利用 CPU 資源，支援多用戶。
- 優化的 SQL 查詢演算法，有效地提高查詢速度。
- 既能夠作為一個單獨的應用程式應用在客戶端服務器網路環境中，也能夠作為一個庫而嵌入到其他的軟體中。
- 提供多語言支援，常見的編碼如中文的 GB 2312、BIG5，日文的 Shift JIS 等都可以用作數據表名和數據列名。

- 提供 TCP/IP、ODBC 和 JDBC 等多種資料庫連接途徑。
- 提供用於管理、檢查、優化資料庫操作的管理工具。
- 可以處理擁有上千萬條記錄的大型資料庫。

參、MYSQL 應用

與其他的大型資料庫例如 Oracle、IBM DB2、SQL Server 等相比，MySQL 自有它的不足之處，如規模小、功能有限等，但是這絲毫也沒有減少它受歡迎的程度。對於一般的個人使用者和中小型企業來說，MySQL 提供的功能已經綽綽有餘，而且由於 MySQL 是開放原始碼軟體，因此可以大大降低總體擁有成本。

目前 Internet 上流行的網站構架方式是 LAMP

(Linux Apache MySQL PHP)，即是用 Linux 作為作業系統，Apache 作為 Web 服務器，MySQL 作為資料庫，PHP (部分網站也使用 Perl 或 Python) 作為服務器端腳本解釋器。由於這四個軟體都是開放原始碼軟體，因此使用這種方式不用花一分錢就可以建立起一個穩定、免費的網站系統。

MySQL 加 PHP 的配對在網際網路上的應用相比 LAMP 來說更為常見，並獲得了「動態配對」(Dynamic Duo) 的雅號，大部分 Blog 網站基於的 WordPress 系統主要運用 MySQL 加 PHP 的配對。除了 LAMP 之外，用於 Solaris、Windows 和 Mac 上的網站構架也分別被稱為 SAMP、WAMP 和 MAMP。

維基百科所使用的 Mediawiki 維基引擎採用 PHP 語言寫成，並以 MySQL 作為其資料庫管理系統。

肆、MYSQL 架構

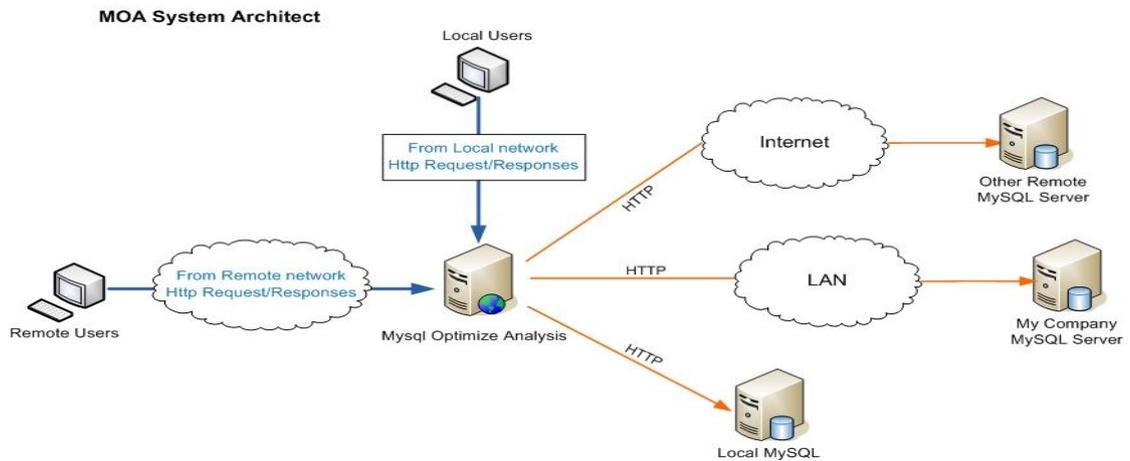


圖 2-6 MYSQL 架構圖

採用 Client/Server 架構

由於 MySQL 採用了 Client/Server 架構，因此 MySQL 的 Server 端與 Client 端不一定要在同一台機器上，例如 Server 端可以是網路上的某台 Linux 機器，而 Client 端則是位在另外一台 Windows 機器上，其概念類似於 Web Server 與 Browser 之間的關係。

MySQL 主要可以分為三大部份

1. MySQL Server：MySQL 資料庫系統的 Server 端，也就是 mysqld 這支程式執行時所在的機器。
2. Client Program：MySQL 資料庫系統的 Client 端，有了 Client 端你才可以與 MySQL Server 進行互動，例如連線至 MySQL Server 新增、刪除資料或是建立新的資料表等等。
3. Non-Client Utility Program：一些用來協助管理 MySQL Server 的工具程式，例如 myisamchk 可用來修復受損的資料表。

MySQL 所使用的通訊協定

既然 MySQL 是使用 Client/Server 架構，那麼當 Client 端需要與 Server 端溝通時便需要遵循特定的通訊協定，下表便是 MySQL 所使用的通訊協定。

通訊協定	支援的連線類型	支援的作業系統
TCP/IP	Local, Remote	ALL
Unix Socket File	Local	Unix-Like OS
Named Pipe	Local	Windows
Share Memory	Local	Windows

表2-2 MYSQ使用的通訊協定

若是 MySQL 的 Client 端與 Server 端皆位在同一台機器上，那麼在作業系統支援的前提下，Client 端可以使用以上所提到的所有通訊協定與 MySQL Server 進行溝通。但若 Client 端與 Server 端各別位在不同的機器上，則 Client 只能夠透過 TCP/IP 與 MySQL Server 進行溝通。

二階層處理模式

當 MySQL Server 收到來自 Client 端送來的 SQL Statement 後，會分成二個階層來處理。

1. 第一個階層－SQL 解析與最佳化：MySQL Server 會先解析所收到的 SQL Statement 並對它進行最佳化，以得到較高的執行效率。在這個階層中並不會實際去存取儲存在資料庫中的資料。
2. 第二個階層－由各種 Storage Engine 組成：Storage Engine 可以視為用來管理資料表的軟體模組，由它來負責實際存取儲存在資料表中的資料。

MySQL 如何使用磁碟空間

- 在 MySQL Server 安裝完畢後，相關的檔案將會放置在 MySQL 的 Data Directory 中，在 Data Directory 中大致有以下的資料。
- 資料庫目錄：MySQL Server 允許你同時管理多個資料庫，而每一個資料庫都會分別在 Data Directory 中建立同名的目錄。例如若是 MySQL 的 Data Directory 為 /var/lib/mysql，則當你建立一個名字為 slime 的新資料庫時，在系統便會自動在 /var/lib/mysql 目錄下建立一個名為 slime 的新資料夾(/var/lib/mysql/slime)。
- 資料表的格式檔(*.frm)：用來記錄資料表的實際結構，此檔案會位在資料庫目錄中。例如若在 slime 這個資料庫中建立一個名為 first_home 的資料表，則在 /var/lib/mysql/slime 目錄下，系統便會自動產生 slime.frm 這個檔案。
- 資料檔(*.MYD)與索引檔(*.MYI)：承上例，若 slime 這個資料表使用的 Storage Engine 為 MyISAM，則系統會自動在資料庫目錄下建立 slime.MYD 與 slime.MYI 這兩個檔案。slime.MYD 用來實際儲存 slime 這個資料表所包含的資料；slime.MYI 則是用來儲存 slime 這個資料表的索引(Index)。註：MyISAM 中的 ISAM 指的是：Indexed Sequential Access Method。
- InnoDB Storage Engine 所使用的 tablespace 與 Log File：與 MyISAM Storage Engine 不同的是，InnoDB 會將所有的資料庫與資料表儲存在單一(或是多個)檔案中，預設是 ibdata1 這個檔案；而 transaction 所需使用的 Log File 則預設為 ib_logfile0 與 ib_logfile1。上述三個檔案預設皆位在 /var/lib/mysql 目錄下(MySQL 的 Data Directory)。

- MySQL Server 本身的 Log File 與 Status File：在這些檔案中記錄了 Server 所執行的各種操作與狀態，主要用來進行 Data Recovery、Replication 或是分析 Server 的運作效率，以及檢測各種可能的問題。

MySQL 如何使用記憶體

- Thread Handlers：MySQL Server 是 Multi-Threaded，一個來自 Client 端的連線即代表一個 Thread，為了避免 Client 端在大量建立與終止連線時所造成的 overhead，MySQL Server 會將 Thread 快取起來，以減少所需建立與終止的 Thread 數量。
- Memory Storage Engine：將資料表建立在記憶體之中(而非檔案系統)以加快存取效率，當 MySQL Server 重新啟動時該資料表也將會消失。
- Server 建立的暫時性資料表：當 MySQL Server 在處理 Query 時會在記憶體內建立暫時性資料表，但若是該資料表的大小超過 tmp_table_size 所設定的上限，則 Server 會改在檔案系統上建立暫時性資料表並將 Created_tmp_disk_tables 的值加一。

<Server 所使用的各種 Buffer >

- Grant Table Buffer：用來存放 Grant Table 的 Buffer，Grant Table 中記錄了使用者的各項權限，在每次執行 client 端送過來的指令前，MySQL Server 必須查核 grant table 以確定該使用者是否具備相對應的權限。為了加快查核的動作，MySQL Server 會將 grant table 快取在 memory 中。

- **Key Buffer**：用來存放 MyISAM table 的索引(Index)，以避免每次存取索引資料時(例如排序)都要重新從磁碟中讀取資料，但要注意的是 MySQL 只會對資料表的索引進行快取，而不會對資料表的內容進行快取。資料表內容的快取是交由作業系統來處理，而不是由 MySQL 負責。
- **Table Cache**：用來存放資料表的 descriptors，以減低檔案重覆開啟與關閉的機率，降低系統負載。註：可以把資料表的 descriptors 看做是資料表在檔案系統上實際對應到的檔案。
- **Query Cache**：用來存放前一次的 Query Result，以加快重覆執行某個 Query 時的效率。
- **Host Cache**：用來存放 DNS 名稱解析的結果，以降低不必要的 DNS Query。
- **InnoDB Storage Engine 所使用的 Log**：主要用來記錄目前的 transaction 狀態，可用以進行 roll back 或是 recommit。

第三章 系統功能與特色

我們開發的 Google Map 商家資訊查詢系統是建構在 Android 平台上的智慧型手機，使用者開啟應用程式時開始觸發 Global Positioning System(全球衛星定位系統)顯示使用者所在位子，經由網際網路連接資料庫載入商家資訊，使用者可以在界面的左手邊設定要查詢的商家類別，選定店家後可以查看店家基本資料還有提供街景和規劃路線，如果需要定位的話只需點商家基本資料裡的電話就會自動幫您撥出。

本專題所建置的商家資訊查詢系統包含有六個元件：使用者介面、Excel、資料網頁、影像辨識系統、資料庫、地址轉經緯度。

使用者介面：使用者經此介面開起 Google Map，再由衛星定位把資料庫相對應經緯度的店家資訊顯示。

Excel：使用 Excel 來存取網頁資料。

資料網頁：資料由 Web66 黃頁與中華電信黃頁使用 Excel 來存取。

影像辨識系統：從中華電信黃頁下載來的資料部分是圖片檔，這時就用影像辨識系統可以快速的將資料轉換成文字檔儲取。

資料庫：使用 MySQL 資料庫查詢對應的經緯度，並將查詢到的商家資訊顯示與使用者介面。

地址轉經緯度：在地圖上標記需要經緯度要用到 Google Map API(Google Map API 每天地址轉經緯度上限只有 2500 筆)

第四章 使用對象

市面上的地圖應用程式比比皆是，可是市面上多數的地圖程式都是小分類或單一性的，因此系統能提供較多分類項目，使用者就不需要去尋找它所需要的地圖類型，對於想找餐廳聚餐、旅遊景點卻又沒有意見時就可以使用到或著是要尋找高度 CP 值的餐廳、旅遊景點時就可以參考評價，還是到一個人生地不熟的地方想樣找加油站、飯店經由分類可以迅速的查詢到相關資訊；此外需擁有智慧型手機 Android2.2 以上的使用者。

第五章 開發工具與建構流程

我們使用—Eclipse，是著名的跨平台的自由整合式開發環境（IDE）。最初主要用來 Java 語言開發，目前亦有人透過外掛程式使其作為 C++、Python、PHP 等其他語言的開發工具。

Eclipse 的本身只是一個框架平台，但是眾多外掛程式的支援，使得 Eclipse 擁有較佳的靈活性。許多軟體開發商以 Eclipse 為框架開發自己的 IDE。

本專題的系統建構流程如下所述：

- (1) 安裝 Java—首先透過 <http://www.oracle.com/technology/index.html> 此網頁來下載 JAVA SE(標準版本)，安裝在 C 磁碟的 JDK160 目錄，變數名稱設回 JAVA HOME。
- (2) 安裝 Android SDK—連上 <http://developer.android.com/sdk/index.html>，再依照作業系統下載對應檔案，藉由此工具可以在電腦上模擬 Android 手機環境，安裝完後 Android 系統的偵錯橋梁 ADB(Android Debug Bridge)會重新啟動。
- (3) 安裝 Eclipse—<http://www.eclipse.org/downloads> 下載 Eclipse 整合開發環境，設定 Android SDK 的路徑。
- (4) 安裝 Android ADT plug in—安裝下載網站設定致 Eclipse 中，透過此系統連結 Android SDK 及 Eclipse。
- (5) 設定 Android 手機模擬器—在電腦中的 Eclipse 中使用 AVD 新增手機模擬器，可安裝多個模擬器，每個模擬器可有不同版本。

- (6) Google Map 的使用，取得 Map API Key，打開 CMD 命令列視窗，使用 JDK 內建工具產生 MD5 指紋，打開瀏覽器輸入網址：
<http://code.google.com/android/maps-api-signup.html>，輸入取得的 MD5 認證指紋。
- (7) 程式撰寫－在 Eclipse 編輯器中撰寫程式，也要將前一項所取得的 google map 相關程式碼輸入。
- (8) 在手機模擬器測試程式－在 Eclipse 中透過偵錯 ADB 來呼叫 AVD(Android Debug Bridge) 模擬器執行程式。
- (9) 連接手機測試程式－將手機設定在 USB 偵錯模式後，安裝 USB 驅動程式，接著連接電腦測試程式。
- (10) Build Apk 程式安裝檔－利用 Eclipse 下的 ADT 選單，在電腦上將程式碼經過 JAVA 編譯，再加上其他輔助檔壓縮形成 APK 程式安裝檔。
- (11) 於手機安裝 APK 檔－將壓縮過的檔案寄至手機的 Gmail 帳號，便可在手機中開啟 APK 程式安裝檔加以安裝。

第六章 軟體架構

本專題所開發的手機/平板電腦之商家資訊系統應用程式的軟體架構包含 Android 程式開發、與資料庫連線建立與取得資料這三個部分。

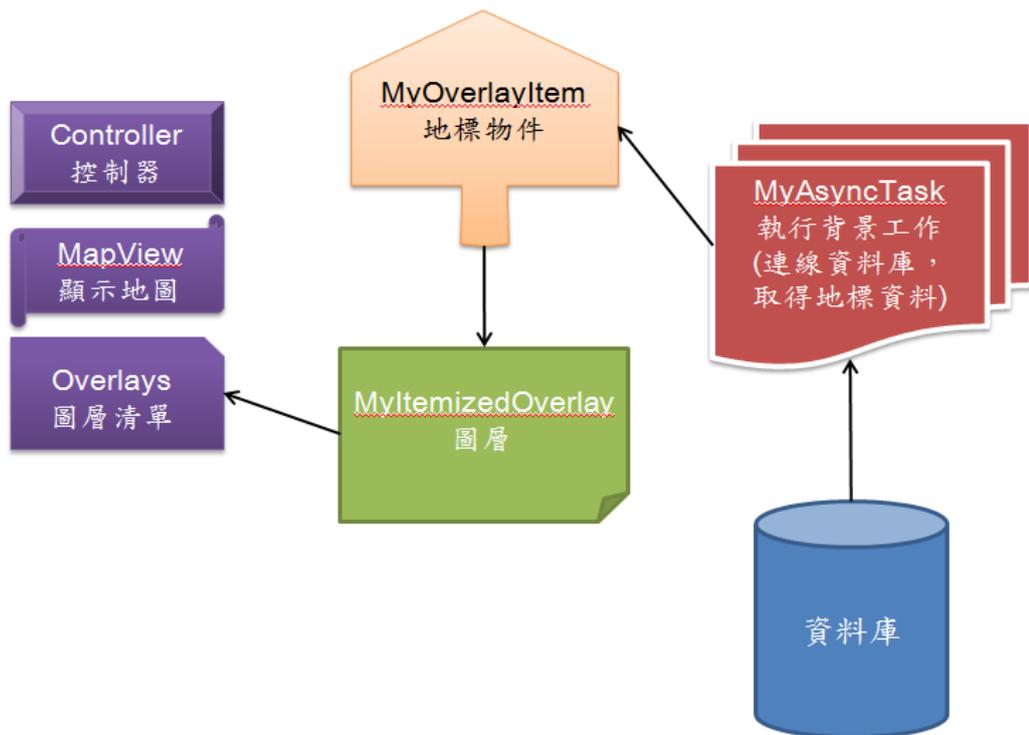


圖 6-1 軟體架構圖

1. **Android** 程式開發，首先 **ParkingActivity.java** 是本系統主要的程式內容，我們在裡面設了四個類別定義，第一個為 **ParkingActivity** 類別定義，它是 **Activity** 行為定義，繼承了 **MapActivity** 因為要顯示地圖，裡面使用到的函數有：**LocationManager** 來取得現在所在位置、**showMyLocation** 動畫顯示現在所在位置以及對於位置各種狀況都能即時處理的能力設置等等；第二個為 **MyAsyncTask** 類別定義，它是背景執行程式的行為定義，繼承了 **AsyncTask** 因為要讓背景工作與主執行緒分開執行，使用到的函數有：**doInBackground** 執行背景工作、**onPostExecute** 在背景工作開始執行前先執行來建立資料庫連線、**onProgressUpdate** 當 **doInBackground** 呼叫 **publishProgress** 時負責更新進度、**onPostExecute** 當背景工作執行完後執行負責告知執行結果等等。**doInBackground** 的功能是從資料庫取得地標資料，將每筆地標資料當作是 **myOverlayItem** 地標資訊物件，並將其加到 **myItemizedOverlay** 圖層，每取得一筆資料，呼叫 **publishProgress** 在主執行緒中更新進度，工作執行完畢後回傳工作執行結果字串，系統接著會觸發 **onPostExecute**；第三個為 **MyOverlayItem** 類別定義，功能為自訂地標資訊物件而地標物件最終需要加到圖層上，它繼承 **OverlayItem** 且 **OverlayItem** 只能儲存兩個字串資訊，內有的函數為：初始化函數、取得電話與各種分類等等，商家名稱與地址可利用 **OverlayItem** 原本就提供的兩個字串儲存；最後一個類別定義為 **MyItemizedOverlay** 類別定義，此類別為自訂圖層物件類別而圖層最終需要加到圖層清單上，繼承了 **ItemizedOverlay**，所包含的函數為：初始化函數、加上地標物件、取得地標陣列(清單)、地標陣列(清單)的大小以及 **onTap** 當某個地標被點選時執行等等，**onTap** 須根據被點選地標的索引，取得地標物件，然後將地標物件的資訊，顯示於訊息視窗的文字標籤中(**TextView**)中。

2. 顯示 google 地圖，Activity 必須繼承 MapActivity，畫面定義檔需加入 MapView 宣告，並寫出地圖金鑰及宣告要使用的 google map 函數庫，如果要利用網路或衛星地位，需宣告適當權限，利用 MapView 設定使用內建放大縮小工具、找出地圖控制器(MapController)、控制地圖放大倍數、控制地圖中心位置、找出圖層清單、將地標物件加到圖層上、將圖層加到圖層清單上等等。
3. 資料庫連線建立與取得資料，首先要取得驅動程式資訊，再來是建立連線(新建連線物件)利用連線物件建立指令物件，利用指令物件執行 select 指令，將 select 指令執行的結果暫存於 recordset 物件，利用 recordset 物件將資料逐筆取出處理(會在回圈之中)。

第七章 系統畫面

第一步驟 從 Gmail 下載 apk 安裝檔

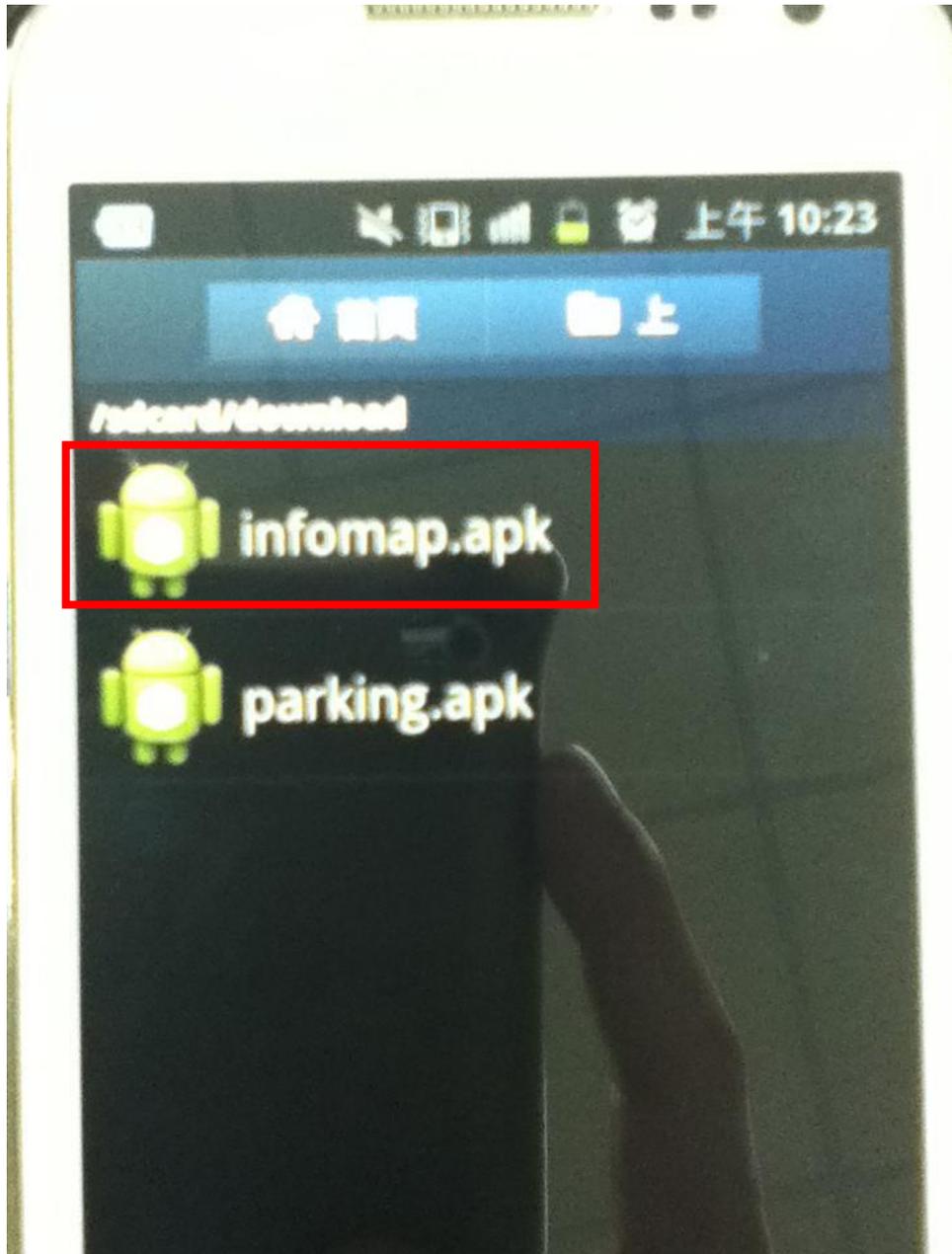


圖 7-1 軟體安裝檔畫面

第二步驟 安裝商家資訊查詢系統到手機/平板電腦上

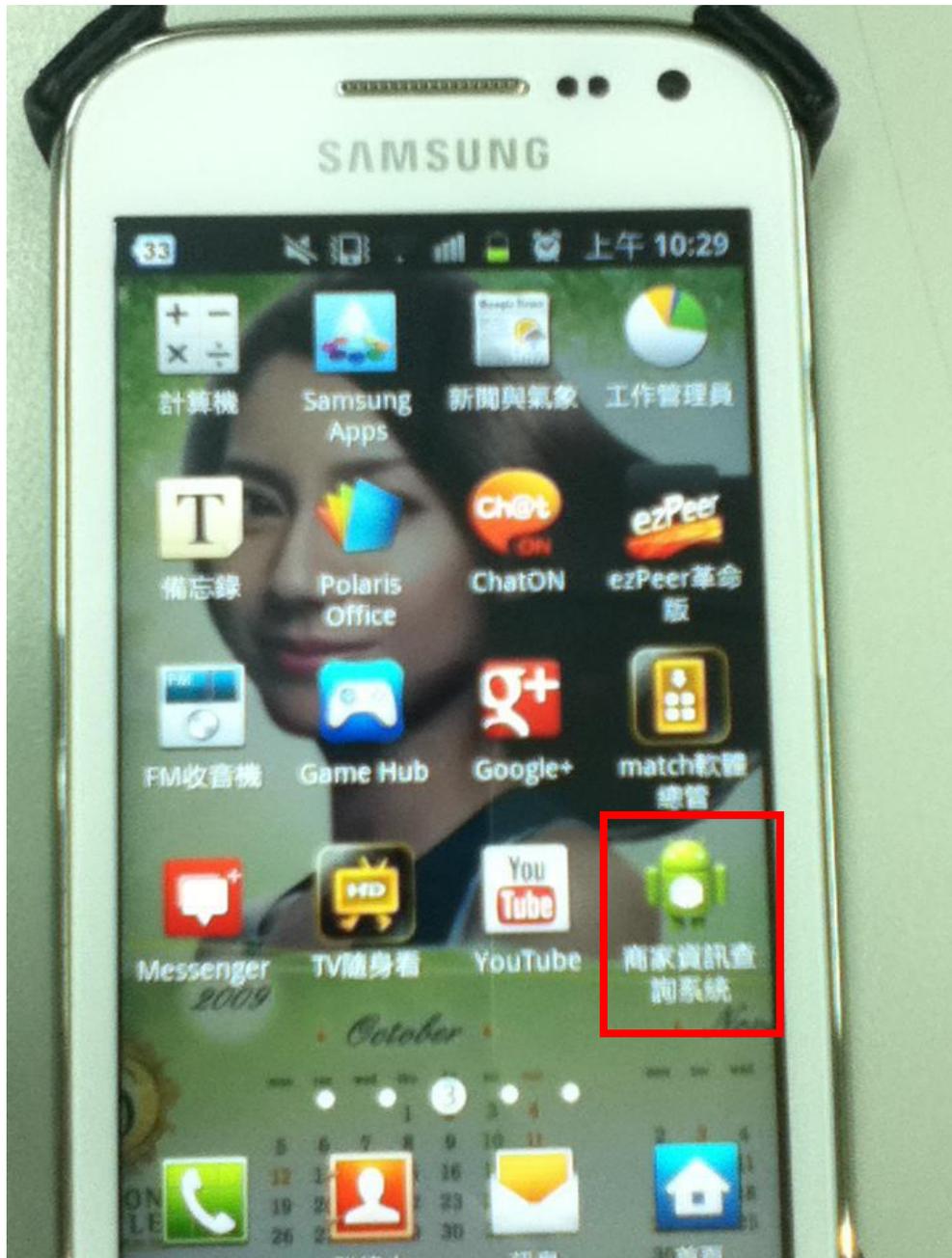


圖 7-2 系統畫面

第三步驟 查詢此區域醫院診所位置並顯示其中一所的資訊

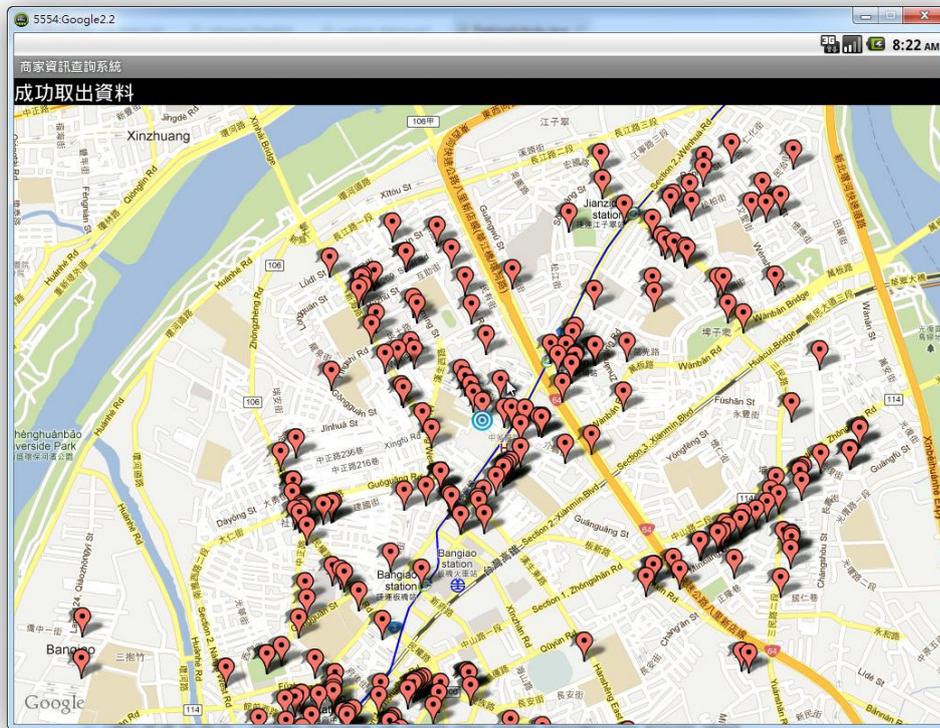


圖 7-3 系統畫面 醫院

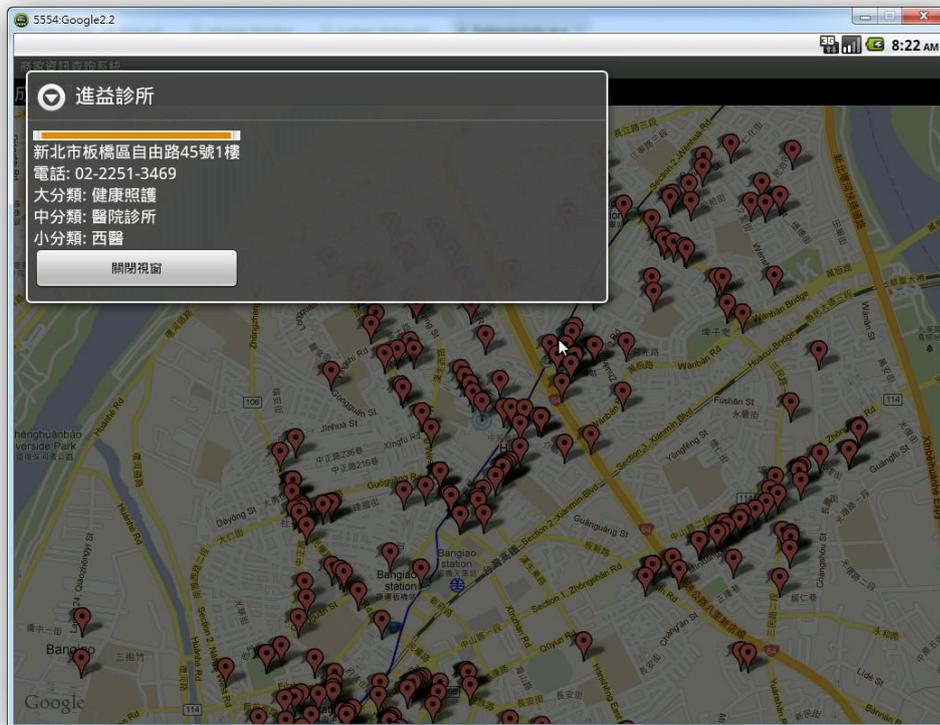


圖 7-4 系統畫面 醫院

第四步驟 查詢此區域加油站位置並顯示其中一間的資訊

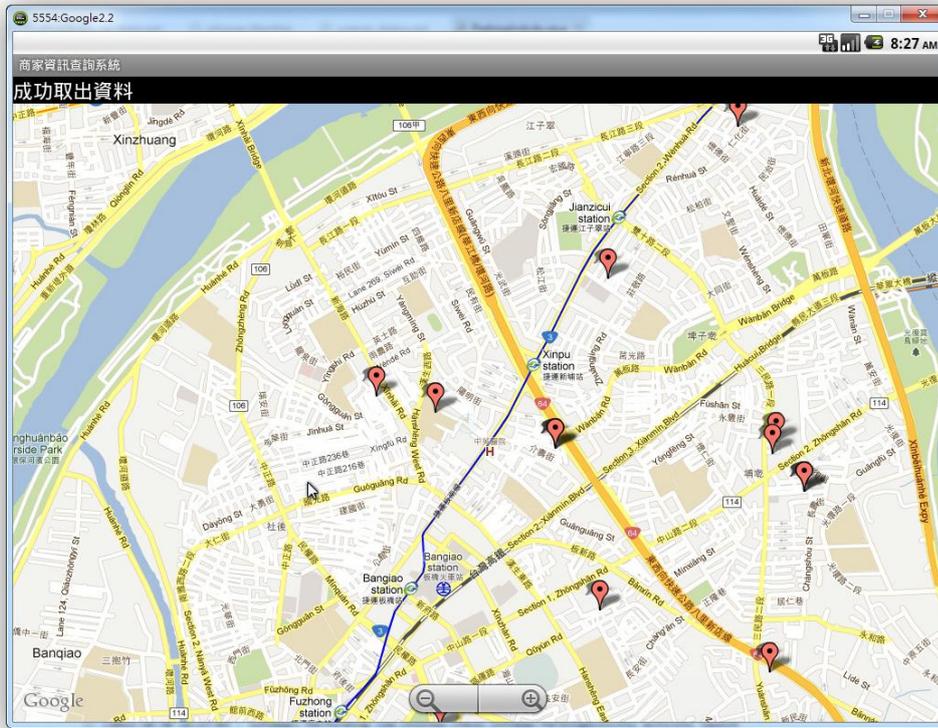


圖 7-5 系統畫面 加油站



圖 7-6 系統畫面 加油站

第五步驟 查詢此區域美食餐廳位置並顯示其中一家的資訊

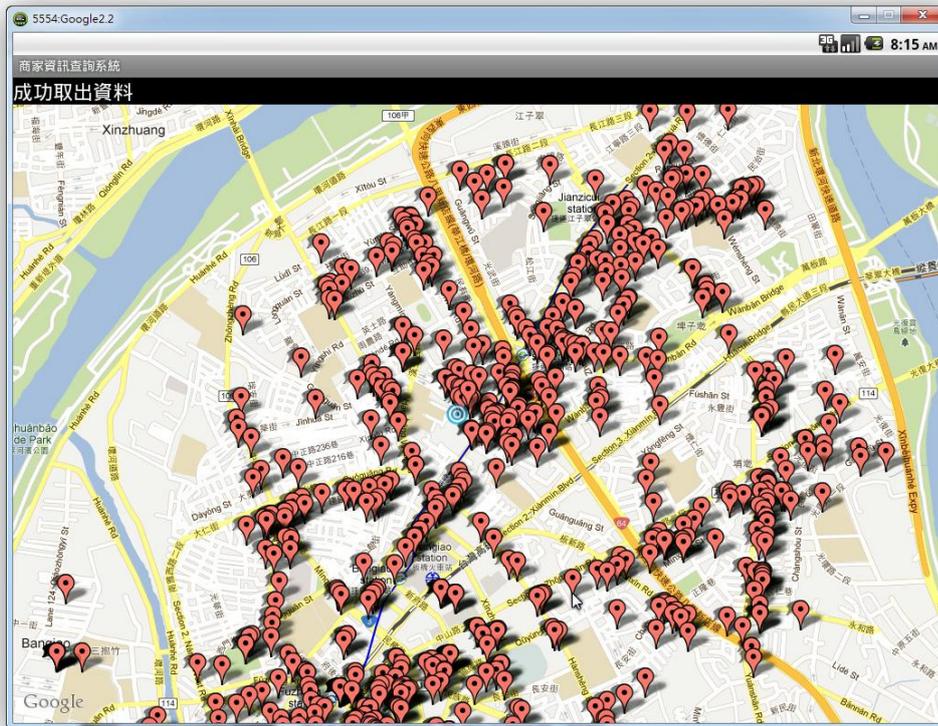


圖 7-7 系統畫面 餐廳

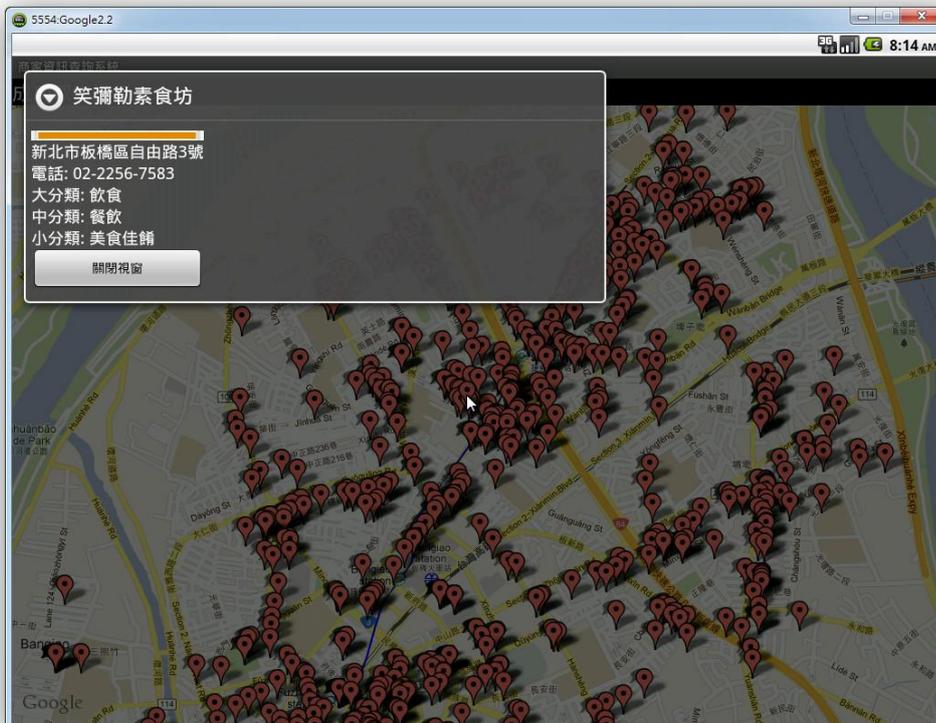


圖 7-8 系統畫面 餐廳

第八章 結論與未來發展

由於 Apple 原始碼沒有開放，只有 Apple 可以生產相較之下 Android 系統的開放吸引眾多廠商使用例如: SUMSUNG、hTc、SONY...，就手機的性能與價格來說 Apple 的手機都是偏單一性且高效能、價位也是偏高的，相反的SUMSUNG 跟 hTc 都是偏機海戰術提供許多機型有高效能也有中低效能來給使用者挑選符合的類型、價錢也隨者性能高低而有所變化，SONY 的話是有獨特的設計與專利在 Android 這部分也佔有一定的比例。Android 的用戶一定越來越多這樣可以快速的擴展 Android 的應用程式的下載與使用率，當應用程式符合使用者的需求與實用性將會由使用者以一傳十、十傳百的速度進行時這將是一個很成功的應用程式。為來是想整合 Facebook 的打卡這樣可以跟你的親朋好友一起分享心情，與停車場系統當你想要去某間餐廳或某個景點時卻找不到停車位時這時他就會幫你尋找離你最近的停車場並顯示還有多少空位，可以幫你省去不必要的麻煩與時間。

參考文獻

中文文獻

1. 吳亞峰、索依娜，2011，學會 Android 應用開發的 18 堂關鍵基礎課程，新北市，博碩文化股份有限公司
2. 柯芊邑，高麒茹，梁耕輔，許揚揚，2011，條碼辨識與商品資訊查詢系統，致理技術學院
3. 吳建斌，蕭凱若，2003，MySQL 資料庫教學手冊，基峰資訊股份有限公司
4. 董士偉、王威翔、洪才庭、林明璋，2010，Android 2 SDK 開發入門與應用，松崗資產管理股份有限公司。
5. 佘志龍、陳昱勛、鄭名傑、陳小鳳、郭秩均，2010，Google Android SDK 開發範例大全，悅知文化。
6. 編委會，2010，Google API 大全-程式設計、開發、實例，松崗資產管理股份有限公司。
7. 江寬、龔小鵬，2008，Google API 開發詳解 Google Map 與 Google Earth 雙劍合璧，松崗資產管理股份有限公司。
8. 孫惠民，2005，Java5+Eclipse 視窗應用程式開發-實作進銷，博碩文化股份有限公司。
9. 謝文川、黃光徹、趙士元，2005，藍芽技術理論與實作，基峰資訊股份有限公司。
10. 蓋索林，2005，Google ! Android 2 手機應用程式設計入門第三版，松崗資產管理股份有限公司。

英文文獻

1. Paul Dubois, Stefan Hinz, Carsten Pedersen, 2005, “MySQL 5.0 Certification Study Guide.”
2. J. Z. Dong, 2008, “Construct of Cell Phone Global Positioning System with Software Model” Department of Electrical Engineering National Ilan University Master Thesis.
3. Gasolin, 2009, “Google! Android mobile phone application programming second edition of Getting Started” Sung Gang Asset Management Corp. Limited,.

網站文獻

1. Android, <http://zh.wikipedia.org/zh-hant/Android>
2. Android, <http://www.android.com/>
3. Android SDK, <http://developer.android.com/sdk/index.html>
4. Eclipse, <http://www.eclipse.org/>
5. Google API, <http://code.google.com/intl/zh-TW/apis/maps/>

附錄 A 系統程式碼

宣告匯入需要使用到的函式庫

```
package com.chihlee.chkuo.infomap;
```

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.List;
```

```
import com.google.android.maps.GeoPoint;  
import com.google.android.maps.ItemizedOverlay;  
import com.google.android.maps.MapActivity;  
import com.google.android.maps.MapController;  
import com.google.android.maps.MapView;  
import com.google.android.maps.MyLocationOverlay;  
import com.google.android.maps.Overlay;  
import com.google.android.maps.OverlayItem;  
import com.google.android.maps.Projection;
```

```
import android.app.AlertDialog;  
import android.app.Dialog;  
import android.app.AlertDialog.Builder;  
import android.content.Context;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Paint.Align;  
import android.graphics.Point;  
import android.graphics.drawable.ColorDrawable;  
import android.graphics.drawable.Drawable;  
import android.location.Location;
```

```
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.view.Display;
import android.view.LayoutInflater;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ZoomControls;
```

此為主要程式內容，先宣告各種需要使用到的變數，如地圖、控制器、圖層清單、所在位置、商家的地標、商家的標籤、自製的對話方塊等等，接下來依照大中小分類輸入指定的資訊，使程式到資料庫抓取指定的商家資訊，並且顯是目前執行進度，從資料庫抓出來的資料，將用地標標示在 ItemizedOverlay 之上，當我們點擊地標時要顯示 AlertDialog 這個自製的對話方塊，上面顯示的資訊有電話、地址、各分類。

並且對於各種執行結果是否成功正確做出判斷，如：地圖顯示、GPS 定位、資料庫連結等等。

```
public class ParkingActivity extends MapActivity implements LocationListener
{
    MapView mapView; //google 地圖
    MapController mapController; //mapView 的地圖控制器
    List<Overlay> mapOverlays; //mapView 的圖層清單

    LocationManager locationManager; //用來定位現在所在位置

    ImageView mapMarker; //用來顯示現在位置的地標(與動畫)
    Animation myAlphaAnimation; //現在位置的地標動畫

    GeoPoint geoPoint;
    View layout;
    AlertDialog alertDialog; //自製 alerdialog
    AlertDialog.Builder builder; //alerdialog 的 builder

    Builder alert;

    //class MyAsyncDBTask extends AsyncTask<String, Integer, String>{
class MyAsyncDBTask extends AsyncTask<String, String, String>{
    Boolean bError;
    String errorString;

    //mysql JDBC 連結物件宣告
    Connection con = null;
    Statement stat, statCount = null;
    ResultSets, rsCount = null;
    String sqlstr;
```

```

        //自訂 ItemizedOverlay 及 OverlayItem
MyItemizedOverlay myItemizedOverlay;
MyOverlayItem myOverlayItem1;

TextView infoTV;

@Override
protected String doInBackground(String... myDummyStrs) {
    int recordCount, progressInt;
    int count;
    if (!bError) {
        try {
            stat = con.createStatement();
            //stat.executeQuery("SELECT lat, lont, name, address, phone,
laddclass, maddclass, saddclass from coinfo where laddclass='行車運輸' and
maddclass='海陸空運輸服務' and saddclass='停車場' and city='新北市' and
area='板橋區' and lat>0");
            stat.executeQuery("SELECT lat, lont, storename, address, phone,
total, available, price from testing where mcat='美食佳餚' and address like '%板
橋%' and lat> 0");
            rs = stat.getResultSet();
            rs.last();
            recordCount = rs.getRow();
            rs.beforeFirst();
            count = 0;

            while (rs.next()) {
                longlat = rs.getLong("lat");
                longlont = rs.getLong("lont");
                GeoPointmysqlGeoPoint = new GeoPoint ((int) lat, (int)
lont);

                String storename = rs.getString("storename");
                String address = rs.getString("address");
                String phone = rs.getString("phone");
                String total = "飲食";
                String available = "餐飲";
                String price = "美食佳餚";

```

```

        //if (phone !=null) {
        //} else {
        //    phone = "(02)27516600";
        //}
        //XXXXXXXXXXXXXXXXXXXXXXXXX 測試先刪除
        myOverlayItem1 = new MyOverlayItem(mysqlGeoPoint,
storename, address, phone, total, available, price);
        myItemizedOverlay.addOverlay(myOverlayItem1);
        //XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        count++;
        progressInt = (int) (((float) count/(float) recordCount) * 100);
        //publishProgress("總資料筆數:" + Integer.toString(recordCount)
+ ";已處理資料筆數:" +Integer.toString(count));
        publishProgress("總資料筆數:" + Integer.toString(recordCount)
+ ";已處理資料百分比:" +Integer.toString(progressInt) + "%");
        //publishProgress(((int) count/recordCount * 100);

    }

    safelyCloseDB();
    errorString = "成功取出資料";

    //myItemizedOverlay.addOverlay(myOverlayItem1);
    //XXXXXXX
    }
    catch (SQLException x) {
    errorString = "SQLException - Select";

    }

    }
    returnerrorString;
}

@Override
protected void onPreExecute(){

```

```

        bError = false;
        infoTV = (TextView) findViewById(R.id.infoTV);

Drawabledrawable = getResources().getDrawable(R.drawable.androidmarker);
        if (drawable != null) {

                drawable.setBounds(0,0,drawable.getIntrinsicWidth(),drawable.getIntrinsic
Height());
                myItemizedOverlay = new MyItemizedOverlay(drawable,
ParkingActivity.this);

                try {
                        Class.forName("com.mysql.jdbc.Driver");
                        con =
DriverManager.getConnection("jdbc:mysql://123.193.186.197:3306/parking?us
er=android&password=android");
                        if (con!=null) {
                                errorString="連線建立成功";

                                } else {
                                        errorString="連線建立失敗";
                                }
                        }
                catch (ClassNotFoundException e) {
                        bError = true;
                        errorString = "ClassNotFoundException - Connection ";
                        safelyCloseDB();
                }
                catch (SQLException x){
                        bError = true;
                        errorString = "SQLException - Connection";
                        safelyCloseDB();
                }
        } else {
                bError = true;
                errorString = "地標製作 drawable 失敗";
        }
}

```

```

infoTV.setText(errorString);
    super.onPreExecute();

}

@Override
protected void onPostExecute(String myResultStr){
    MapView mapView1;
    List<Overlay> mapOverlays1; //mapView 的圖層清單
    super.onPostExecute(myResultStr);
    infoTV = (TextView) findViewById(R.id.infoTV);
    infoTV.setText(myResultStr);
    if (myResultStr=="成功取出資料") {
        mapView1 = (MapView) findViewById(R.id.mapView);
        mapOverlays1 = mapView1.getOverlays();
        mapOverlays1.add(myItemizedOverlay);
    } else {
        infoTV.setText(myResultStr);
    }
}

@Override
//protected void onProgressUpdate(Integer... myProgressCodes) {
protected void onProgressUpdate(String... myProgressCodes) {
    super.onProgressUpdate(myProgressCodes);
    infoTV = (TextView) findViewById(R.id.infoTV);
    //infoTV.setText("載入資料
中..." +Integer.toString(myProgressCodes[0]) + "%");
    infoTV.setText(myProgressCodes[0]);

}

private void safelyCloseDB(){
    try {
        if (rs!=null) {
            rs.close();
            rs=null;
        }
    }
}

```

```

        if (rsCount!=null) {
            rsCount.close();
            rsCount=null;
        }

        if (stat!=null) {
            stat.close();
            stat=null;
        }

        if (statCount!=null) {
            statCount.close();
            statCount=null;
        }

        if (con!=null) {
            con.close();
            con=null;
        }
    }
    catch (SQLException x) {
        bError = true;
    }
}
}

```

```

class MyItemizedOverlay extends ItemizedOverlay<MyOverlayItem> {
    private ArrayList<MyOverlayItem> mOverlays = new
    ArrayList<MyOverlayItem>();
    Context mContext;
    Drawable myMarker;

    public MyItemizedOverlay(Drawable defaultMarker, Context context) {
        super(defaultMarker);
        mContext = context;
        myMarker = defaultMarker;
    }
}

```

```

        // TODO Auto-generated constructor stub
    }

    public void addOverlay(MyOverlayItem overlay) {
        mOverlays.add(overlay);
        populate();
    }

    @Override
    protected MyOverlayItem createItem(int i) {
        // TODO Auto-generated method stub
        return mOverlays.get(i);
    }

    @Override
    public int size() {
        // TODO Auto-generated method stub
        return mOverlays.size();
    }
    /*
    @Override
    public void draw(Canvas canvas, MapView mapView, boolean shadow) {
        Projection projection = mapView.getProjection();
        //Bitmap markerImage =
        BitmapFactory.decodeResource(mContext.getResources(), myMarker);

        for (int index = size() - 1; index >= 0; index--) {
            MyOverlayItem myOverlayItem = getItem(index);
            String available = myOverlayItem.getMaddclass();
            Point point = projection.toPixels(myOverlayItem.getPoint(), null);
            Paint paintCircle = new Paint();
            paintCircle.setARGB(10, 0, 0, 0);
            canvas.drawCircle(point.x, point.y, 25, paintCircle);
            Paint paintText = new Paint();
            paintText.setColor(Color.RED);
            paintText.setTextSize(25);
            paintText.setTextAlign(Align.CENTER);

```

```

        canvas.drawText(available, point.x, point.y, paintText);
        //Paint paintBitmap = new Paint();
        //canvas.drawBitmap(markerImage,
point.x-markerImage.getWidth()/2, point.y-markerImage.getHeight()/2, null);

    }
    //super.draw(canvas, mapView, shadow);
    //boundCenter(myMarker);
}
*/

@Override
protected boolean onTap(int index) {
    TextView nameTV, addressTV, phoneTV, laddTV, maddTV, saddTV;
    MyOverlayItem item = mOverlays.get(index);

    //nameTV = (TextView) layout.findViewById(R.id.textView1);

    addressTV = (TextView) layout.findViewById(R.id.textView1);
    addressTV.setText(item.getSnippet());
    phoneTV = (TextView) layout.findViewById(R.id.textView2);
    phoneTV.setText("電話: "+item.getPhone());
    laddTV = (TextView) layout.findViewById(R.id.textView3);
    laddTV.setText("大分類: "+item.getLaddclass());
    maddTV = (TextView) layout.findViewById(R.id.textView4);
    maddTV.setText("中分類: "+item.getMaddclass());
    saddTV = (TextView) layout.findViewById(R.id.textView5);
    saddTV.setText("小分類: "+item.getSaddclass());

    //顯示 AlertDialog
    alertDialog.setTitle(item.getTitle());
    alertDialog.show();
    return true;
}
}

class MyOverlayItem extends OverlayItem {

```

```

private String phone, laddclass, maddclass, saddclass;
public MyOverlayItem(GeoPoint point, String title, String snippet, String
phoneStr, String laddclassStr,
                    String maddclassStr, String saddclassStr) {
    super(point, title, snippet);
        //this.name = nameStr;
        //this.address = addressStr;
    this.phone = phoneStr;
    this.laddclass = laddclassStr;
    this.maddclass = maddclassStr;
    this.saddclass = saddclassStr;
        // TODO Auto-generated constructor stub
    }

    public String getPhone(){
    return phone;
    }

    public String getLaddclass(){
    return laddclass;
    }

    public String getMaddclass(){
    return maddclass;
    }

    public String getSaddclass(){
    return saddclass;
    }

    }

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    }

```

```

MyAsyncDBTaskdbTask = new MyAsyncDBTask();
dbTask.execute("start");

alert = new AlertDialog.Builder(this);

mapView = (MapView) findViewById(R.id.mapView);
mapController = mapView.getController();
mapOverlays = mapView.getOverlays();

mapController.setZoom(16);
mapView.setBuiltInZoomControls(true);

        mapMarker = new ImageView(this);
        mapMarker.setImageResource(R.drawable.mapmarker);
        mapMarker.setAdjustViewBounds(true);
        mapMarker.setScaleType(ImageView.ScaleType.FIT_XY);

        myAlphaAnimation = new AlphaAnimation(0,1);
        myAlphaAnimation.setFillBefore(false);
        myAlphaAnimation.setFillAfter(true);
        myAlphaAnimation.setDuration(500);
        myAlphaAnimation.setRepeatCount(Animation.INFINITE);
        myAlphaAnimation.setRepeatMode(Animation.REVERSE);

        myAlphaAnimation.setInterpolator(AnimationUtils.loadInterpolator(this,a
ndroid.R.anim.accelerate_decelerate_interpolator));

        myLocationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

        //使用 inflater 將 xml 畫面定義轉成一個 View (變數名稱為 layout)
        LayoutInflaterinflater = (LayoutInflater)
this.getSystemService(LAYOUT_INFLATER_SERVICE);
        layout = inflater.inflate(R.layout.custom_dialog, null);

        //利用 AlertDialog 的 Builder 來根據 layout 生成一個對話視窗
(alertDialog)

```

```

builder = new AlertDialog.Builder(this);
builder.setView(layout);
alertDialog = builder.create();

//設定 alertDialog 的顯示位置
alertDialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
WindowManager.LayoutParams WMLP =
alertDialog.getWindow().getAttributes();
    // display 可以求出螢幕大小
    Display display = getWindowManager().getDefaultDisplay();
    //alertdialog 位置若是 0,0 表示顯示在螢幕中央
WMLP.x = -display.getWidth()/2;    //x position
WMLP.y = -display.getHeight()/2;  //y position
alertDialog.getWindow().setAttributes(WMLP);

    //好像沒作用，設背景為透明
    //alertDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(0));

    //geoPoint = new GeoPoint(25022000,121465000);

    //myOverlayItem1 = new MyOverlayItem(geoPoint, "致理技術學院", "就
在捷運站附近", "0933720476", "學校", "北區", "技術學院");
    //myItemizedOverlay.addOverlay(myOverlayItem1);

initiateLocationService();
//mapOverlays.add(myItemizedOverlay);

}

public void closeDialog(View view) {
    alertDialog.dismiss();
}

```

```

private void initiateLocationService() {
    String locationProvider = LocationManager.GPS_PROVIDER;
    Location lastKnownLocation =
myLocationManager.getLastKnownLocation(locationProvider);
    if (lastKnownLocation == null) {
        locationProvider = LocationManager.NETWORK_PROVIDER;
        lastKnownLocation =
myLocationManager.getLastKnownLocation(locationProvider);
    }
    if (lastKnownLocation != null) {
        showMyLocation(lastKnownLocation);
    } else {
        //alert.setTitle("偵錯");
        //alert.setMessage("定位完全失敗");
        //alert.show();

        GeoPointmyGeoPoint = new GeoPoint(25021002, 121465198);
        mapView.removeAllViews();

mapController.animateTo(myGeoPoint);
mapController.setCenter(myGeoPoint);

mapView.addView(mapMarker, new
MapView.LayoutParams(MapView.LayoutParams.WRAP_CONTENT,
MapView.LayoutParams.WRAP_CONTENT, myGeoPoint,
MapView.LayoutParams.CENTER));
        mapMarker.startAnimation(myAlphaAnimation);
    }
}

private void showMyLocation(Location location) {
    //GeoPointmyGeoPoint = new GeoPoint((int) (location.getLatitude() *
1E6), (int) (location.getLongitude() * 1E6));

```

```

        // for testing , remove correct location with school location
        GeoPointmyGeoPoint = new GeoPoint(25021002, 121465198);
        mapView.removeAllViews();

        mapController.animateTo(myGeoPoint);
        mapController.setCenter(myGeoPoint);

        mapView.addView(mapMarker, new
        MapView.LayoutParams(MapView.LayoutParams.WRAP_CONTENT,
        MapView.LayoutParams.WRAP_CONTENT, myGeoPoint,
        MapView.LayoutParams.CENTER));
            mapMarker.startAnimation(myAlphaAnimation);

    }

    @Override
    protected void onResume() {
        super.onResume();

        //showParkingInfo();

    }

    @Override
    protected void onPause() {
        super.onPause();

        //myLocationOverlay.disableCompass();
        //myLocationOverlay.disableMyLocation();

    }

```

```
@Override
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}

public void onLocationChanged(Location location) {
    showMyLocation(location);
    // TODO Auto-generated method stub(

}

public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub

}

public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub

}

public void onStatusChanged(String provider, int status, Bundle extras) {

    // TODO Auto-generated method stub

}

}
```