

國立勤益科技大學
資訊工程系研究所

碩士論文

雲端服務平台與 Android 整合系統之設計與開發

Design and Development of Cloud Service Platform

and an Android Application System

研究生：徐世威

指導教授：李隆財博士

中華民國一百零一年七月

國立勤益科技大學
研究所碩士班
論文口試委員會審定書

本校 資訊工程系碩士班 徐世威 君

所提論文 雲端服務平台與 Android 整合系統之設計與開發

合於碩士資格水準，業經本委員會評審認可。

論文口試委員會：

召集人：張一介

委員：張菡英

李隆財

指導教授：李隆財

所長：王圳木

中華民國一〇一年七月十三日

雲端服務平台與 Android 整合系統之設計與開發

研究生：徐世威
國立勤益科技大學
資訊工程系

指導教授：李隆財
國立勤益科技大學
資訊工程系

摘要

智慧行動裝置與雲端技術的結合是現今科技發展的一大趨勢，本文之雲端服務平台為一提供給各式服務、物品供應商之平台，各供應商無須自行建構硬體、軟體以及維護成本，只需透過此雲端平台建構其供應之項目，最後成果展現以餐飲店家為例，即是將其提供之食物、飲料等資訊上傳至此平台後，任何使用者可於任何時間，任何地點透過各種行動裝置或電腦，本論文以 Android 為例，接收資料，並提供使用者透過本文實作之 GAE 雲端技術進行外帶點餐、外送、訂位之服務。

本研究使用的雲端平台為 GAE 開發雲端應用程式，實作雲端運算與雲端資料儲存，並在供應商之操作介面使用應用網頁技術做管理，結構由網頁伺服器與資料庫伺服器構成。

Android 系統做為提供終端使用者瀏覽商品項目、訂購、外送、訂位之方便設備，所有資料都儲存在雲端，降低手機負荷，並可隨時取得店家最新資訊，也實作手機資料庫做為暫存最近存取資料，提供使

用者離線瀏覽功能。

本研究除了實作 GAE 雲端服務平台與 Android 系統間的整合系統外，並透過介紹本研究進行時架構設計的進程，提供一個專案從起頭開始進行應該有的流程參考，並附上本研究實作最後能取得的實驗數據與其它市面上已經在正式商業運轉系統間的比較。

關鍵詞：雲端服務、智慧型手機、Android、App、GAE。



Design and Development of Cloud Service Platform and an Android Application System

Student: Shih-Wei Hsu

Advisor: Lung-Tsai Li

Department of Computer Science and Information Engineering

National Chin-Yi University of Technology

Abstract

Combining the smart mobile device with the cloud service is a trend of modern Information technology. The cloud service platform described in this paper is a platform that provides many types of services and products. Therefore, the service providers do not need to build their own hardware such as the cloud server or ordering devices thus saving the hardware costs as well as the software maintenance costs by using the proposed cloud service platform. This paper examines a case study of food service in achievement display, in which the suppliers can use the cloud service platform to provide food and drinks to any user using a smart mobile device. This paper also examines a case using an Android Application running on smart mobile device to access data from the cloud service platform and provides services such as food and drink takeout, delivery, or reservation to users.

The cloud service platform developed in this paper is based on GAE technology to provide cloud service, cloud computing and the cloud storage. The server infrastructure is composed of a Web server and a datastore server.

The Web browser is used as an interface for food service providers to manage their products and process customer orders. The customer runs an Android App on smart mobile device to view menus, place orders, or make reservations. As the database is stored at the cloud, it is not necessary to store all the data in the mobile device. Only recent data that has been implemented is stored in the database of the mobile to enable the user to

view data offline and to reduce the network traffic to the cloud server, thus improving system performance.

This research not only implements a Cloud Service Platform and an Android App but also provides a development procedure reference guide so that any project can follow the development steps from the beginning to the end. The analysis results learned from this research are also provided in this paper.

Keywords: Cloud service, Smart Phone, Android, App, GAE



誌謝

從大學開始到碩士的求學生涯中，一路上都有指導教授李隆財老師的幫助與教導，提供我很好的學習環境以及發揮的空間，並在各方面給予關懷以及細心的指導。

還要感謝口試委員張蓺英教授在論文上給予許多寶貴的意見，更要感謝張一介總經理給予我這個研究的機會，並在論文研究的過程中不斷給予支持與教導，讓本論文有很好的開始跟完美的結束。另外也謝謝學弟們在求學生涯的陪伴，讓生活增添許多樂趣。

特別要感謝我的家人的栽培與無怨無悔為我的付出，以及女朋友嘉琪特地到台中陪伴我照顧我。希望接下來的職場生涯也能蒙神祝福，並充分發揮所長，回報所有愛我的人。

目錄

摘要	I
Abstract	III
目錄	VI
致謝	VII
圖目錄	VIII
表目錄	X
第一章 緒論	1
1.1 研究背景	1
1.2 研究動機和目的	2
1.3 文獻探討	4
1.4 章節概要	5
第二章 Google App Engine	6
2.1 雲端服務	6
2.2 Google App Engine 簡介	8
2.3 Google MapReduce	10
2.4 Google BigTable	11
第三章 Android	12
3.1 Android 簡介	12
3.2 Activity 的生命週期	13
第四章 系統規劃	16
4.1 GAE Service System	16
4.2 Android Service System	20
4.3 Android 使用者介面設計	22

第五章 系統實作與數據分析.....	28
5.1 雲端服務平台開發.....	28
5.2 Android 系統程式開發.....	29
5.3 資訊加解密機制.....	32
5.4 數據分析與比較.....	35
第六章 系統實測與介紹.....	39
6.1 GAE 雲端服務平台.....	39
6.2 Android 開發實測結果.....	42
第七章 結論及未來工作.....	50
參考文獻.....	51



圖目錄

圖 2.1 雲端服務三層	7
圖 2.2 Google App Engine 優勢	8
圖 3.1 Android 系統架構 [15]	13
圖 3.2 Activity 生命週期 [15]	15
圖 4.1 雲端服務平台與 Android 系統整合之架構	16
圖 4.2 GAE Service System Flowchart	18
圖 4.3 GAE Service System ER Model	19
圖 4.4 Android Service System flowchart	21
圖 4.5 Android Service System ER Model	22
圖 4.6 專案設計流程	24
圖 4.7 Android 使用者介面架構	25
圖 4.8 使用者介面架構部分圖一	26
圖 4.9 使用者介面架構部分圖二	27
圖 5.1 開發程式碼架構	29
圖 5.2 Android 開發程式碼架構	31
圖 5.3 個人識別式區塊加密方法	34
圖 5.4 個人識別式區塊解密方法	35
圖 5.5 下載流量測試分析結果	37
圖 5.6 上傳流量測試分析結果	37
圖 5.7 網路流量測試分析結果	38
圖 6.1 GAE 雲端服務平台登入畫面	39
圖 6.2 GAE 雲端服務平台最高管理者登入畫面	40
圖 6.3 GAE 雲端服務平台最高管理者管理合作店家畫面	40
圖 6.4 GAE 雲端服務平台店家管理者管理訂單畫面	41
圖 6.5 GAE 雲端服務平台店家管理者管理商品畫面	41
圖 6.6 GAE 雲端服務平台店家管理者新增商品畫面	42
圖 6.7 Android 店家列表畫面	43
圖 6.8 Android 店家列表分類篩選畫面	43
圖 6.9 Android 店家列表篩選後畫面	44
圖 6.10 Android 店家分店列表	44
圖 6.11 Android 下載圖檔中畫面	45
圖 6.12 Android 圖檔下載完畢畫面	46

圖 6.13 Android 商品詳細資訊畫面.....	46
圖 6.14 Android 商品訂購後列表畫面.....	47
圖 6.15 Android 購物車模式畫面.....	48
圖 6.16 Android 訂單紀錄列表畫面.....	48
圖 6.17 Android 訂單紀錄詳細資訊.....	49



表目錄

表 2.2 各家雲端運算比較..... 10



第一章 緒論

1.1 研究背景

餐飲業數位化的腳步隨著科技的進步也不斷推陳出新，大部分餐飲店幾乎都有自己的餐飲系統，但也僅限於店家 POS 系統。隨著智慧裝置的普及化，有越來越多餐飲店希望能讓顧客能使用數位裝置點餐，而自行著手研發；另一方面也有一些公司著手研發提供部分餐飲店特定服務之功能，其功能的限制性與系統之架構產生許多使用上的侷限。

本研究透過完整規劃之服務系統架構，期望將服務的侷限性縮到最小，能供應各個商家的各種服務；並透過雲端儲存使得使用者之 Android 系統僅需取得所需資料與最新資訊，以及雲端服務讓使用者透過 Android 系統即可隨時從雲端取得訂餐、外送、訂位之服務。

本論文研究重點有二：1 完整規劃一個雲端服務平台並使用 GAE (Google App Engine)實作，使所有商家都能提供其所需之服務、商品；2 透過 Android 系統能取得個人所需之服務、商品，同時系統使用不完全受限於 Android 連線能力，並且具有高流暢度及低系統資源消耗。

1.2 研究動機和目的

在現今科技蓬勃發展的時代，人們對於時間異常重視，若在重要時刻點餐、訂位必須排隊，甚至是打電話訂購也可能在忙碌中，這些現象都是每個人都會遇到的問題，現今雖然市面上可以看到有幾種軟體提供訂餐服務，卻狹隘的僅提供某種特定服務，大大的降低其實用性，系統也都是建構在純網路架構上，若遭遇網路不穩定或是無訊號狀態，則完全無法啟動程式。

目前市面上的 Android 服務程式雖說架構上似乎所有東西都是在雲端上，但卻是變相將手機作為瀏覽器呈現內容，如此一來只要網路有任何壅塞就會導致系統進入難以使用的狀況，即使網路暢通將資料轉送給內嵌瀏覽器引擎再轉而呈現導致高度互動的程式反應不順暢。

本論文希望改善這些狀況透過實作架構為 Android Cloud App 的程式，即是將必需的資訊留存於系統內，僅於資料更動時更新系統資料，如此一來既避免程式的依賴網路性，也能保有隨時最新的資訊，並且將大部分的資料放置雲端，降低 Android 系統資料負荷量，同時透過雲端服務在連線時可輕易透過雲端取得個人所需的服務，透過雲端運算將 mapping 與 sorting 資料在雲端完成，Android 即可輕易取得最新最符合個人需要之服務內容。

Android 系統於開發上不需要授權金，開發作業系統也可使用 Linux 與 Windows，在目前的市場趨勢看來，Android 系統之手機的銷售量已逐漸超越其他各式手機，在未來的發展開發潛力上相當高，因此選擇 Android 系統做開發。

在雲端系統的選用上有 Amazon Elastic Computing，Windows Azure 及 GAE (Google App Engine) 等是市面上的雲端系統龍頭，前兩者一個是開山龍頭，一個是電腦界龍頭，為何要選用 GAE 呢？原因也跟選用 Android 一樣，GAE 是可以免費使用的，具有一定額度的免費雲端空間、一定額度的流量、頻寬、存取數量等等，不過在學術研究上已經是非常足夠了，因此本篇論文採用 GAE 為例開發雲端系統完成本論文之研究。

本研究旨在探討雲端技術與行動裝置軟體技術之結合，透過實作雲端運算與雲端儲存，建構一個完整的雲端服務與終端使用系統。

透過雲端服務讓人們享受科技的便利，如何在 Android 系統上建構出一個流暢的雲端整合技術是本論文的重點。

1.3 文獻探討

因應智慧裝置普及的趨勢在市場上也很快有人開發 Android [1-2] 行動裝置上隨時隨地可連網取得的雲端服務技術[3-4]，綜觀這些 Android 應用程式幾乎都完全架構在 Web 2.0 技術上，有些行動裝置甚至直接顯示 Html 頁面，使用後可以發現其優劣：

優點：

1. 移植容易：直接透過 Android 顯示 Html 頁面使開發人員可以不須針對 Android 系統環境特性做開發，透過瀏覽器不論是電腦或智慧裝置上都可以直接瀏覽操作。
2. 資料隨時與伺服器同步：透過 Html 或是 Web 技術每次開啟應用程式都會直接跟伺服器取得資料，無須佔用行動裝置上的硬體空間且資料都是最新的。

缺點：

1. 操作便利與螢幕尺寸：在行動裝置上大部分都是小尺寸且長寬比例與電腦螢幕也不同，若直接顯示 Html 的頁面就顯得操作非常不便。
2. 網路依賴性：這些技術都是必須在網際網路具有連結的能力下

才能操作，只要訊號稍不佳就難以使用，使得這些應用能隨時隨地使用的雲端技術卻時常因為地點不佳難以使用。

1.4 章節概要

本篇論文主體架構共分成七個章節，各章節的編排與概要說明如下：第一章為緒論，描述研究背景以及研究動機和目的。第二章敘述本研究使用的 GAE (Google App Engine) 之相關技術探討。第三章敘述 Android 的相關技術探討。第四章介紹本篇整體研究系統架構、研究方式以及流程。第五章詳述系統實作上的程式方法架構，並提供數據分析結果驗證本篇研究之成果。第六章詳述並圖示本篇研究的總體成果。第七章為結論及未來工作。

第二章 Google App Engine

2.1 雲端服務

雲端一詞的由來是，工程師在繪製網路配置圖時會習慣將網際網路畫一朵雲來代替，而不論是雲、或是網際網路，都不是一個實體。自然界中的雲擁有循環、延展、靈活的特性，而網際網路的架構正符合這些特點，同樣的資源可以在不同的電腦之間循環流通；不論是 Scale-up 或是 Scale-out [5] 架構的雲端運算都具有其不同特性的延展性，網際網路的靈活性也更不用說了，需要時才取得，不需要時也不會佔用太多本地端資源量。

雲端服務簡單來說即是讓使用者隨時隨地能夠取得需要的服務，唯一所需的就是要連上網際網路。對公司企業來說可以省下許多硬體建構與維護的成本，對於一般使用者而言則不需擔心資料儲存、更新的問題，以及硬體速度需求的問題。

雲端技術架構如圖 2.1 所示可以包含三種雲端服務[6]：

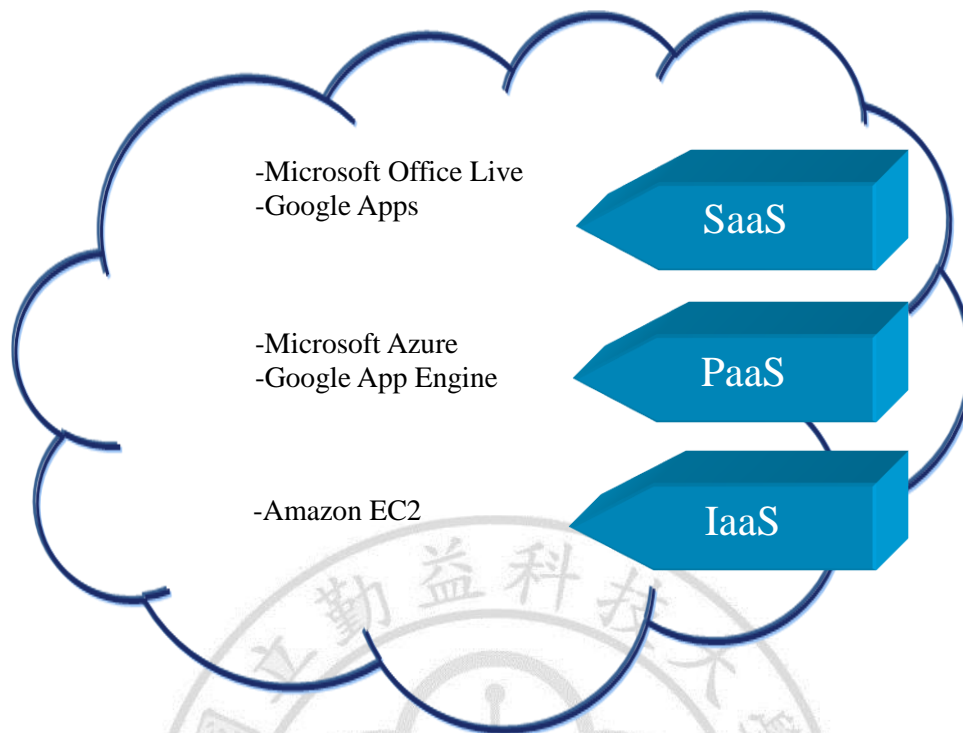


圖 2.1 雲端服務三層

SaaS (Software as a Service) 是可以透過雲端立即取得商業軟體、應用軟體的服務，同時取代以往使用者必須一次性購買軟體的方式，以有使用才需付費的方式，減少一般使用者的軟體支出開銷及軟體版本更新後又必須重新更換的麻煩。

PaaS (Platform as a Service) 以提供一個雲端軟體研發平台作為一種服務，使得開發者擁有一致的開發環境及雲端設備資源的 APIs，開發人員能夠透過這些 APIs 存取雲端設備的資源而無須去瞭解這些設備的細節，也提升了對 SaaS 的開發速度。

IaaS (Infrastructure as a Service) 以透過雲端提供資源設備為服務，如處理器、儲存設備、網路等等，讓使用者不需管理、維護以及更新硬體設備，可以任意建構、安裝所需的作業系統、應用程式。



圖 2.2 Google App Engine 優勢

2.2 Google App Engine 簡介

目前提供雲端運算技術平台的企業有 Microsoft、Amazon、Google 等，各自都擁有專有的核心技術以及不同的計價方式。各家雲端運算技術比較詳見表 2.2。但不論是哪一家企業所提供的雲端運算，最終都能替使用者或使用企業免去很多硬體設備或相關技術上的建構時間及經費。

Google App Engine [3-4][7-8]是一個可以建立並管理網路應用程式的一種 PaaS 雲端技術平台，並使用 Google 管理的資料管理中心作為讓我們開發及託管網路應用程式的雲端平台，而其毋需付費即可立即使用、無須自行搭建及管理資料中心、網路設備、依使用量付費並且讓開發者在一定限制內免費使用的特點，如圖 2.2 所示，也吸引很多開發者加入，同時也使得本論文進行時不需憂慮伺服器硬體設備等雜項因素即可專注於所探討的內容。



表 2.2 各家雲端運算比較

平台	Windows Azure	Google App Engine	Amazon EC2	Yahoo Application Platform
技術特性	整合不同裝置與網路服務	儲存與運算的水平擴充能力	可彈性配置的通用虛擬機器	儲存與運算的水平擴充能力
核心技術	Window Server 2008 與 Hypervisor 虛擬化技術	平行分散技術 MapReduce、BigTable 資料庫系統、GFS 檔案系統	Xen 虛擬化技術	平行分散技術 Hadoop、MapReduce、Hbase 資料庫、HDFS 檔案系統
支援的開發語言	.NET 語言 (IIS 7 支援語言)	Web Python, 未來會支援更多語言	企業可自行建置不同作業系統和平台的執行環境	PHP
支援的資料庫系統	SQL Service, 如資料表、檔案等。	BigTable 資料庫系統	提供 S3 儲存服務, 企業可自行建置所需資料庫系統	HBase 資料庫系統
計費方式	將按資源與服務等級計價	按使用的處理器時間、儲存空間與網路流量計價	按使用的處理器時間、儲存空間與網路流量計價	尚未公布

資料來源: 中華民國資訊軟體協會。雲端運算 Cloud Computing 的概念與應用。http://eblog.cisnet.org.tw/post/Cloud-Computing.aspx

2.3 Google MapReduce

MapReduce [9] 是 Google 的雲端技術核心之一，是一種用來平行處理大量資料的方法，MapReduce 方法如今已經廣泛被使用，不過最早

是由 Google 所提出。

在進行運算工作時，系統會將資料分割成一個個不同區塊，再透過大量機器進行 Map 運算工作同步處理與分析，經過 Map 工作處理分析的資料會使 Reduce 工作負擔大大降低，最後再經由 Reduce 工作進行合併後輸出完整結果。

2.4 Google BigTable

在雲端運算的普及下，作為雲端服務的提供者在資料庫部分也成為各種應用必須的項目之一。Google 所提供的資料庫架構也同樣為了能夠更合適的進行其核心技術的平行分散式計算而設計，一個 BigTable [10]的資料可以分布在數百甚至數千個伺服器中，因此高擴展性、高性能都是其特點。

BigTable 是一種分散式、長久儲存的多維度排序 map，map 有行關鍵字、列關鍵字及時間戳做為索引。列的索引與數值是任意字串，同時每一行或列的欄位不只能儲存單一的數值，也可以是一整個數值的集合與關聯的時間戳。BigTable 現在也廣泛使用在各個需要巨型資料庫的應用上如 Google Earth、Gmail、Youtube 等。

第三章 Android

3.1 Android 簡介

Android 是一種採用 Linux 作為核心的開放原始碼作業系統[11-14]，並為了能更合適於行動裝置做了修改及擴充。使用圖形化的介面作主要操作方式，提供使用者更直覺的操作模式。

如圖 3.1 [15]，主要可將 Android 分為四層架構，最底層使用 Linux 作為核心負責硬體驅動以及網路、電源管理等。往上一層提供系統執行所需的各種 Libraries，如存取顯示系統功能的 Surface Manager、關聯式資料庫系統 SQLite、網頁瀏覽器引擎 Web Kit 等等，其中分離了一塊 Android Runtime 區塊內有 Core Libraries 以及 Dalvik Virtual Machine，Android 透過這兩個元件更有效率的執行 JAVA 程式。Application Framework 層提供了開發者一個完整、統一的應用程式介面，如常見的控制項 Button、TextView 或是嵌入式瀏覽器等。最上層 Application 就是各種基本的 JAVA 應用程式，讓使用者一開始就有基本的應用程式使用，如瀏覽器、照相機等。

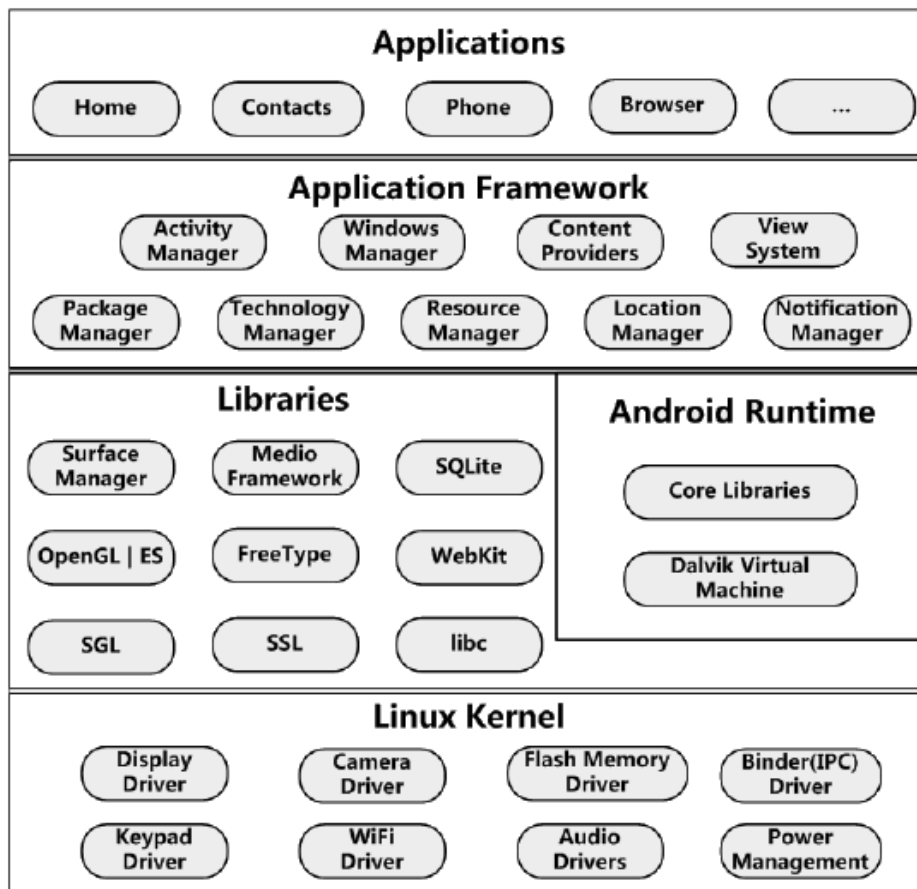


圖 3.1 Android 系統架構 [15]

3.2 Activity 的生命週期

要開發 Android 系統程式[16]就必須了解 Android 如何為智慧裝置作業系統管理有限記憶體而使用的生命週期機制，現今的作業系統為了提供使用者快速操作多種功能大多都使用多工處理的作業系統，可以同時執行不同程式，但在智慧行動裝置上不如 PC 般有大量的系統記憶體，當同時執行過多程式或是記憶體為正確釋放導致記憶體不足就會導致系統緩慢或是運作不正常。

一個啟動的 Activity 都將經過固定的生命週期如圖 3.2，依特性主要可以將其分成四種狀態：

Active/Running：當使用者啟動一個應用程式或一個 Activity，程式會處於 Active 狀態，從啟動開始會經過 onCreate、onStart 最後到 onResume 完成啟動。從圖 3.2 流程可以看到除了正常啟動 Activity 會進入此狀態外，還有可能從 onPause 及 onResume 會到 Active 狀態，就是在當前 Activity 可能是從其他地方被喚回或是重新取得焦點時。

Paused：Paused 狀態會在當前 Activity 暫時失去焦點時，不論是跳出小視窗導致當前 Activity 暗掉或是其他程式突然跳出使得當前 Activity 處於暫時無法操作的狀態就會觸發 onPause 進到 Paused 狀態。

Stopped：如果當前 Activity 將焦點的控制權給其他 Activity，當前 Activity 不再動作時就會觸發 onStop 進到 Stopped 狀態。

Dead/Destroyed：當 Activity 被終止或是處於 Stopped 的狀態下被系統自行結束以回收記憶體，就會觸發 onDestroy 進到此狀態

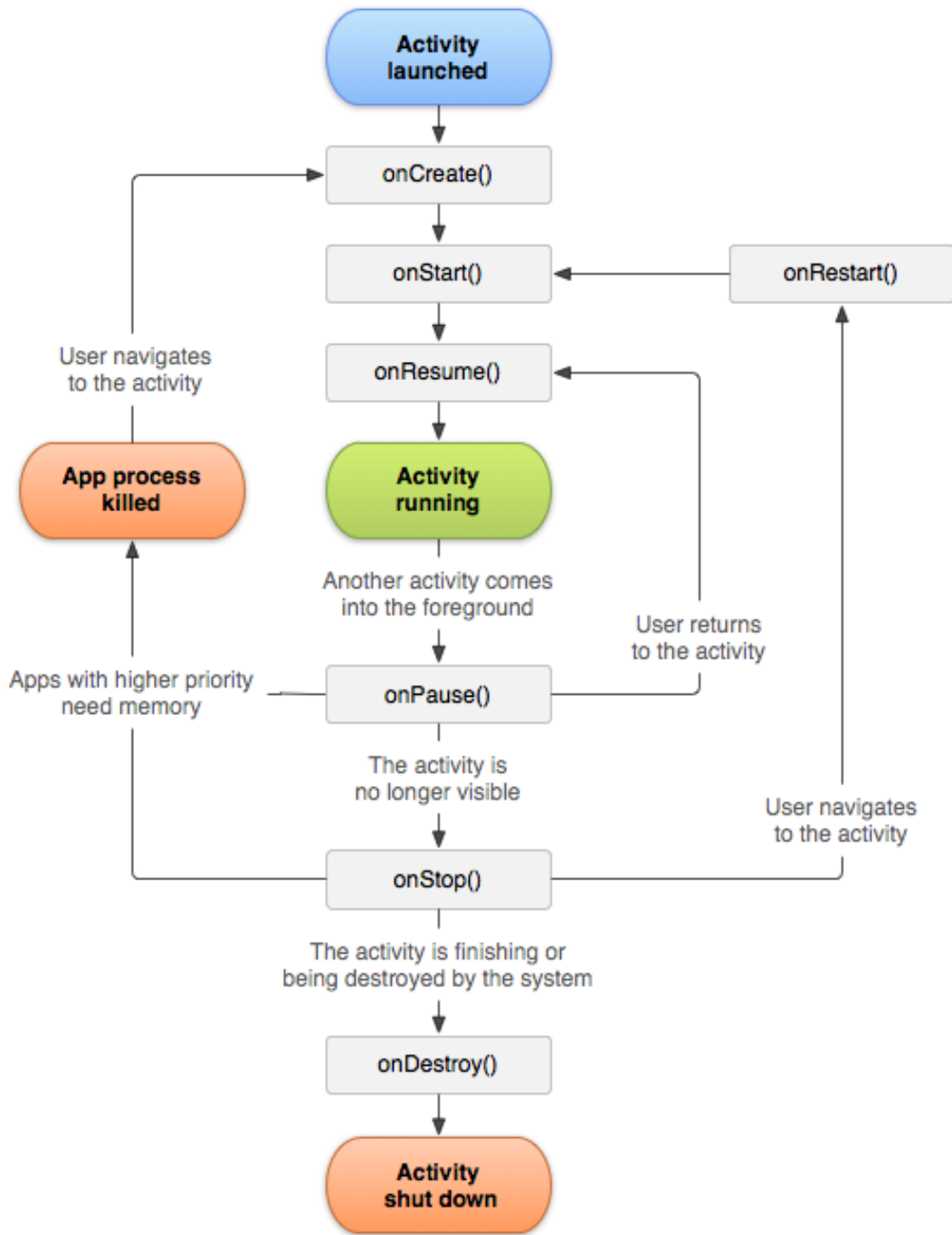


圖 3.2 Activity 生命週期 [15]

第四章 系統規劃

系統架構如圖 4.1 所示，餐飲店家可任意使用電腦或平板透過瀏覽器使用雲端服務平台建構自己的服務項目到 GAE 雲端伺服器，並且透過伺服器將資料儲存於雲端資料庫中，完成後 Android 端即可隨時到雲端取得最新服務資訊以及獲得所需要的服務。

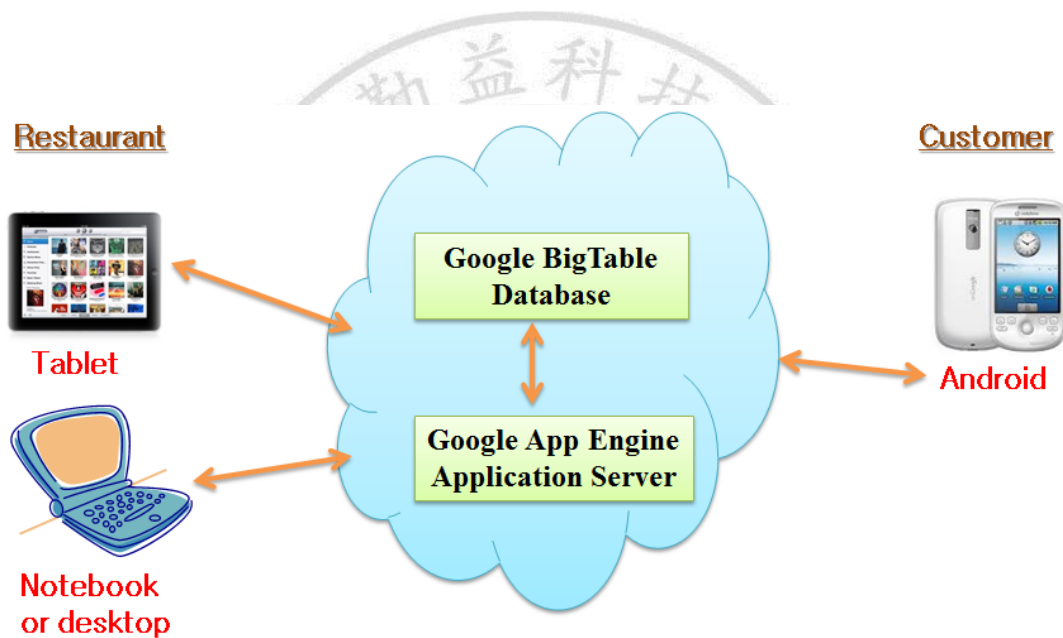


圖 4.1 雲端服務平台與 Android 系統整合之架構

4.1 GAE Service System

餐飲系統中雲端服務系統內部流程架構如圖 4.2，依功能可分為三部份：

1. **最高管理者** 具有最高權限，可以進行分配權限、增刪帳戶之功

能，當一個新的廠商加入時就必須透過最高管理者以最高權限分配管理店家所需的權限給該對應的帳戶。

2. **餐廳管理者** 可管理所屬的店家分店進行增刪或是編輯資訊，以及增刪、編輯要提供之服務、商品等大量資訊。

3. **客戶使用者** 具有瀏覽商品，取得商家服務等功能。

雲端服務系統為了詳實紀錄從管理紀錄、店家服務資訊到客戶點購紀錄等，因此利用 Entity Relationship Model [17] 規劃一個嚴謹的資料庫藍圖，如圖 4.3 的 ER Model 說明了管理者每個資料實體之間與其他實體間有一對一甚至多對多的關係。



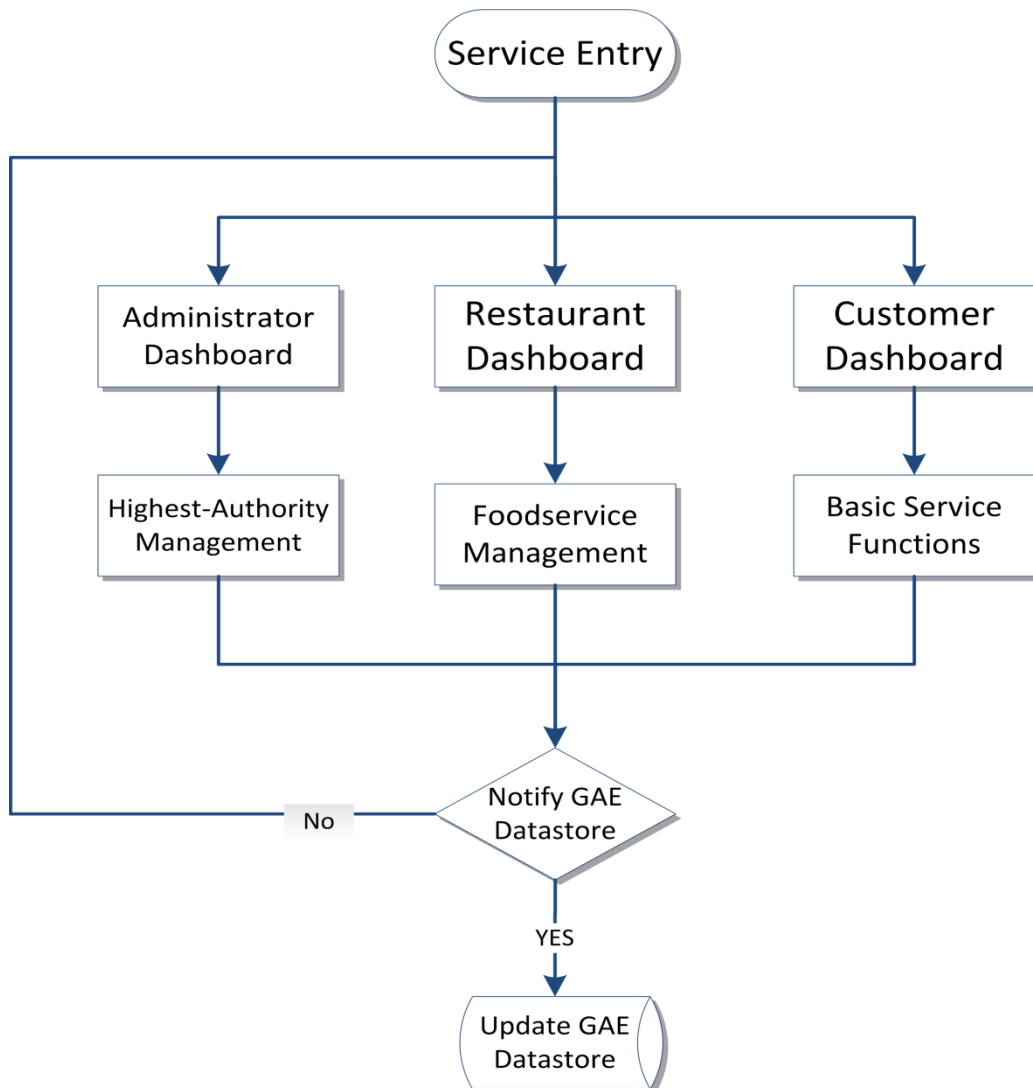


圖 4.2 GAE Service System Flowchart

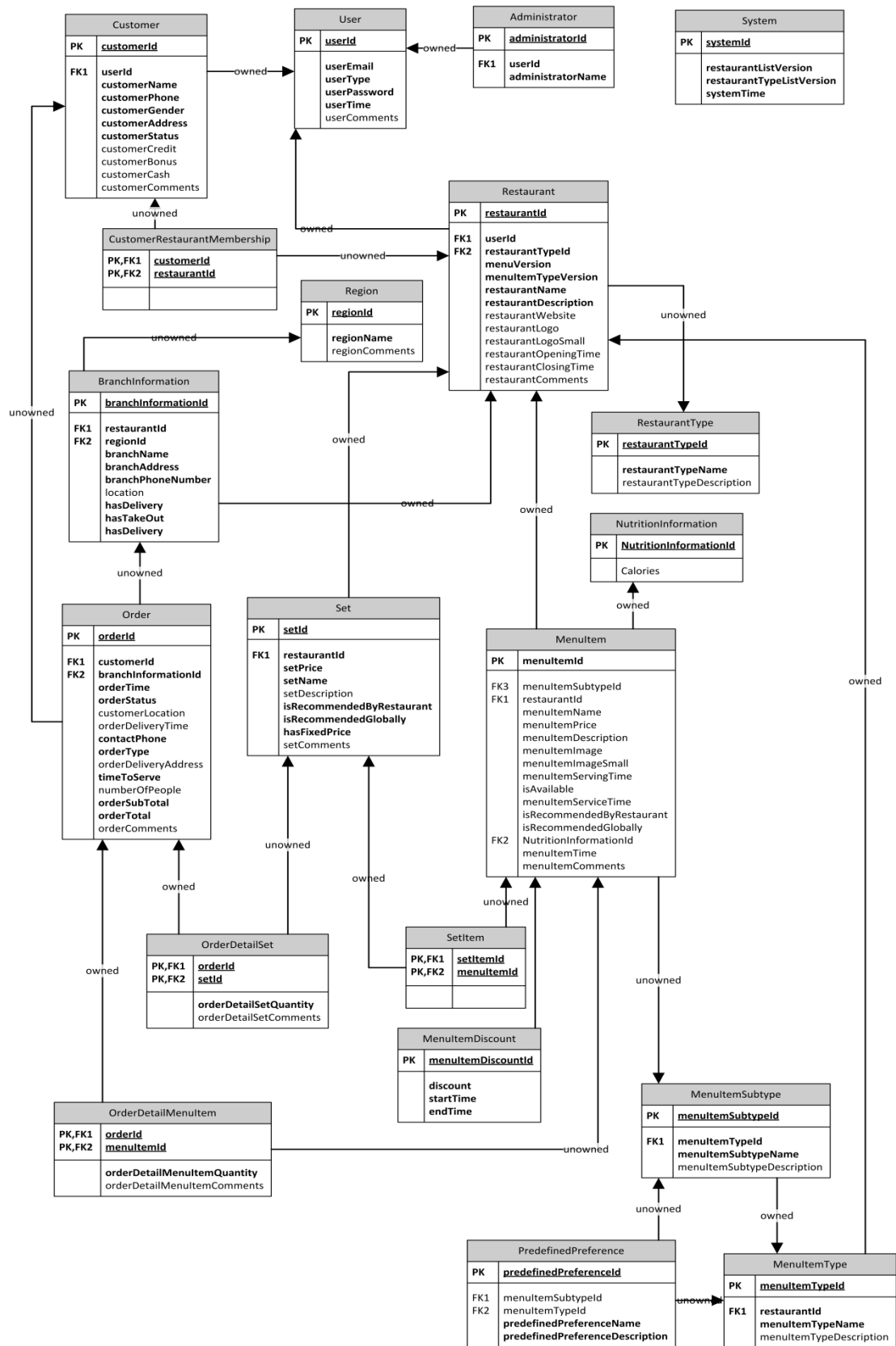


圖 4.3 GAE Service System ER Model

4.2 Android Service System

餐飲系統中 Android 系統介面功能流程如圖 4.4，除了一般使用者帳戶控制機制外，重點部分就在 Ordering 流程，為了提高系統流暢度並降低不需要的網路流量，在下載大量資料前系統都會先行比對 GAE 雲端資料庫系統版本與 Android SQLite 儲存資料之版本，若版本相同則直接顯示已儲存之資料，若版本不同才從 GAE 雲端取得新資料，並在最後向 GAE Server 送出服務要求前做一次版本檢核，避免任何可能的資料誤差狀況發生。



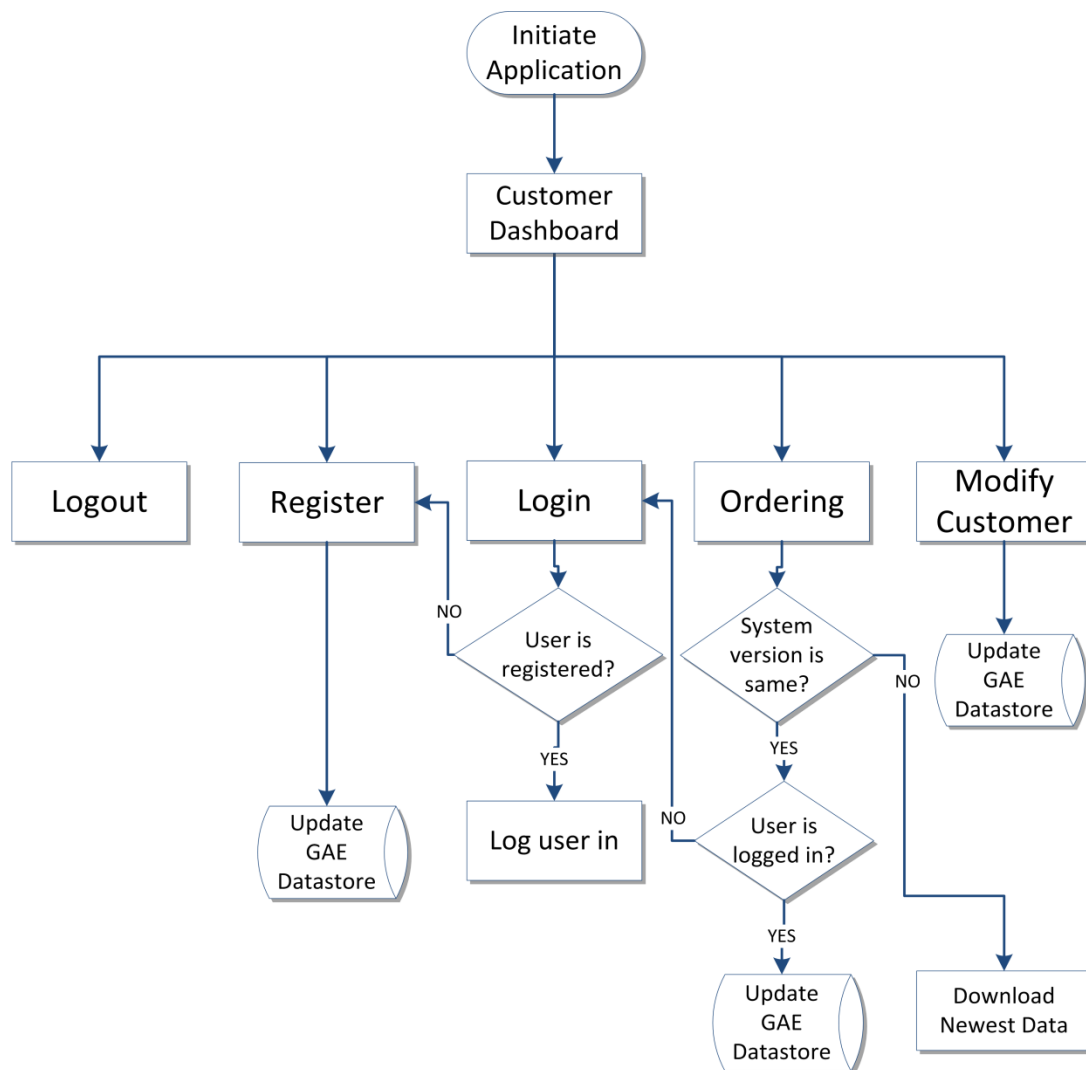


圖 4.4 Android Service System flowchart

為了使 Android 行動裝置容量需求盡可能輕量化，在 ER Model 的設計上也特地輕量化設計，如圖 4.5，Android 系統中只保留必要的資訊，除了降低所需網路流量，也降低所需佔用的資料空間。

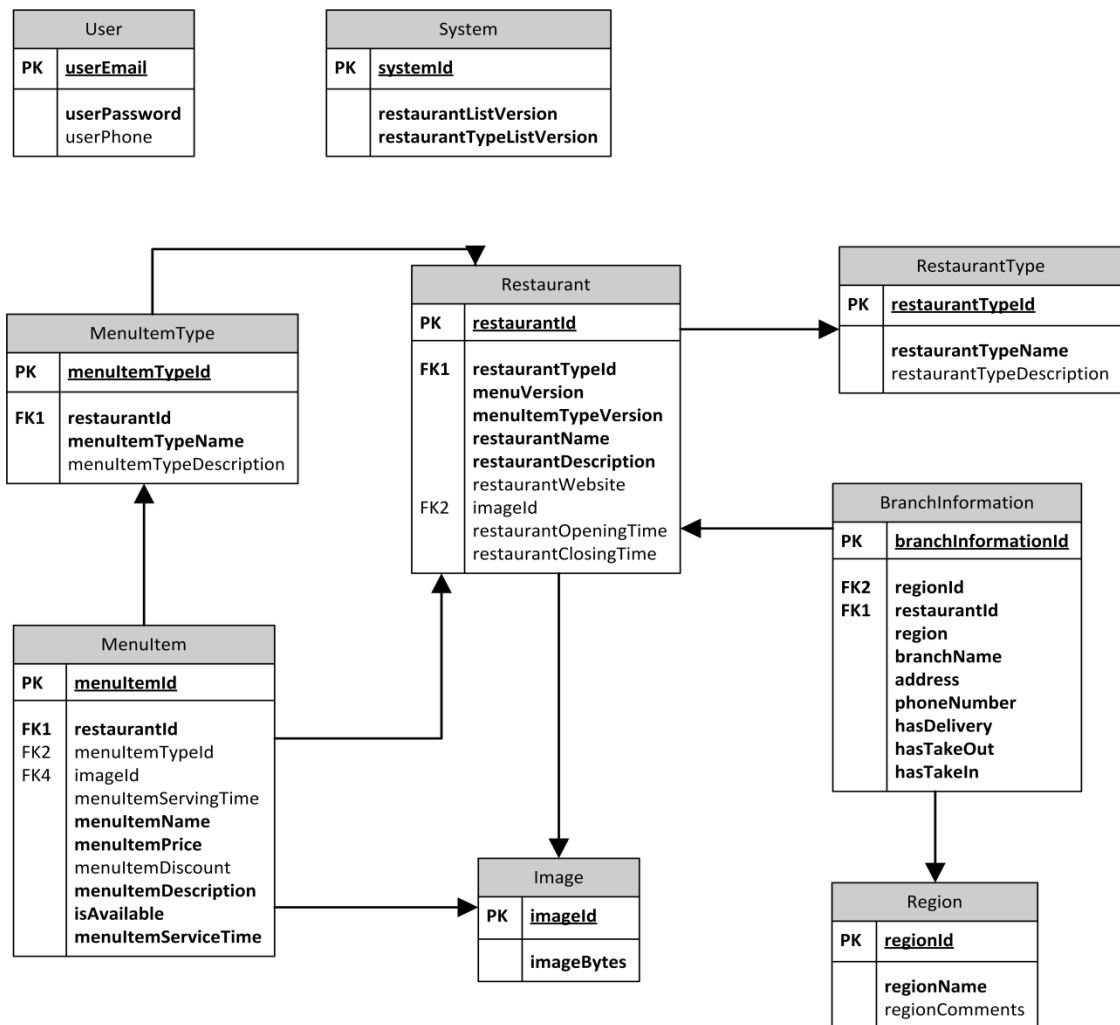


圖 4.5 Android Service System ER Model

4.3 Android 使用者介面設計

本篇研究在使用者介面設計上較注重 Android 端[18]，因為在實際應用上 Android 端也是廣大一般使用者的主要操作系統。手持裝置應用程式的操作方式與一般個人電腦的操作方式因畫面大小與控制方式不同，因此在使用者介面的設計上也不能像過去設計 PC 介面的方式進行。

一個好的使用者介面應該要注意以下四點：

1. 在有限的螢幕大小呈現清楚簡潔且足夠的資訊

手持裝置的螢幕大小一般都是直立式的，不但長寬比例不同，大小也比 PC 螢幕差的多，必須要在有限的畫面展現足夠的資訊，同時也不能讓畫面擠得雜亂無章。

2. 一致化的操作方式且要能符合使用者習慣

手持裝置在呈現不同資料時，都會切換不同頁面，但在許多不同畫面所帶給使用者的操作方式最好能夠一致，否則會造成使用者困惑，甚至造成使用者的錯誤操作。

3. 盡量讓使用者了解操作進度，同時不能影響操作過程

在手持裝置小畫面的操作過程中，常常將 PC 一個畫面能完成的工作必須分割成好幾個畫面完成。此時應該讓使用者知道當前進行到的步驟，讓使用者能掌控自己的進度，同時提醒的方式也不能影響使用者正常的操作。

4. 要能夠容忍錯誤的操作

目前智慧型裝置幾乎都是以觸控作為主要的操作方式，觸控方式雖然靈敏快速，但是有時不小心碰觸也會被判定為輸入控制行為，因此在操作上必須要有能夠避免這種誤觸行為的機制。如在重要選擇時

都要讓使用者最後再確認一次自己的選擇是否正確。

除了介面設計上的基本要注意的事情外，一個系統從設計到實現的過程，除了程式的功能架構及資料架構要設計外，使用者介面的架構設計也是跟著其他架構環環相扣的，一個完整設計的專案應該要具備這些流程，如圖 4.6。先分析這個專案需要那些功能，然後著手進行使用者介面的架構設計，最後再經由程式設計師評估再開始進行後續設計。



圖 4.6 專案設計流程

本篇研究的專案在 UI 設計架構上，主要使用一個 Master Page 搭配中間切換不同的 Content Page，如此一來可以在所有頁面擁有一致化的操控方式，提高使用者上手的容易度。如圖 4.7，在 Master Page 放上主要功能鈕，許多不會隨著不同頁面改變顯示及操作的部份就會放在這裡，底下為系統主要服務功能分支的進入點，左上角的返回及右上角的下一步按鈕則可能會隨著不同頁面稍微會有不同，但主要訴求

不變，Content Page 隨著使用者操作瀏覽不同的資訊就會改變切換不同的畫面如圖 4.8，圖 4.9 則為本研究部分實際設計使用者介面的過程。

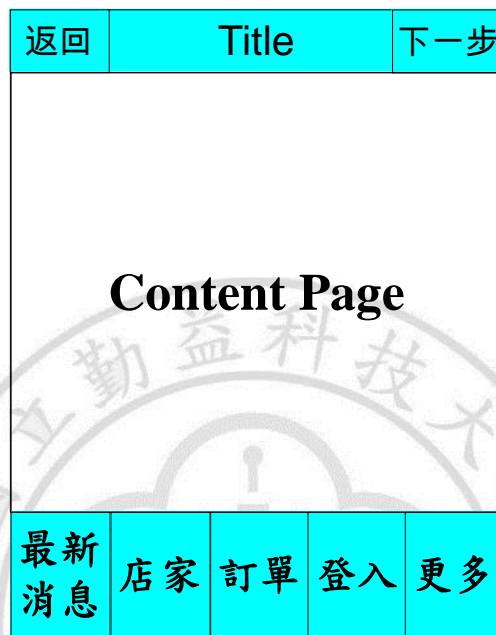


圖 4.7 Android 使用者介面架構

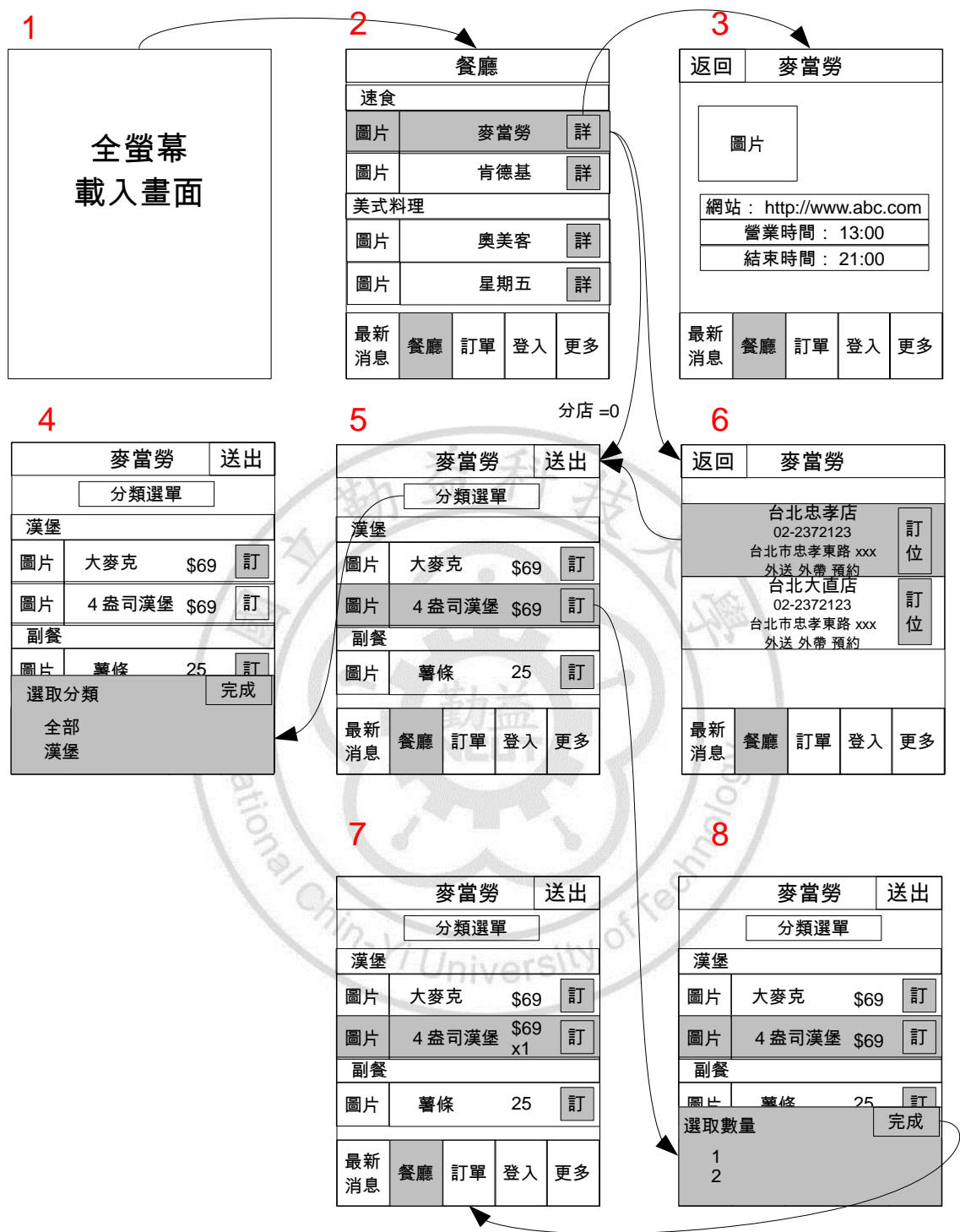


圖 4.8 使用者介面架構部分圖一



圖 4.9 使用者介面架構部分圖二

第五章 系統實作與數據分析

本節將介紹本篇實作的工具以及步驟過程。GAE 在開放支援 JAVA [19]開發環境後，同時也開放 Eclipse 的外掛程式整合，讓開發人員可以使用 JAVA 開發雲端應用程式，本篇也透過 Eclipse 以使用標準 JAVA 網頁技術的 JSP 來開發雲端伺服器，同時 Android 系統是以 JAVA 為中心開發的，也讓本文不需要花費多餘心力在額外的程式語言及開發工具上，同樣使用 Eclipse 工具以 JAVA 開發。

5.1 雲端服務平台開發

GAE 雲端服務平台的開發分成三部份，雲端程式、資料庫系統以及網頁呈現，程式流程以資料庫架構為主在 3.1 節已介紹。

如圖 5.1 紅色虛線部份，第一條線 datastore 為資料庫建構程式，將圖 4.3 的 ER Model 轉為程式邏輯撰寫，第二條線 webservices 撰寫提供給 Android 系統於雲端執行的程式，如取得菜單列表、送出點購菜單等，第三條線則是以 JSP 撰寫，提供給餐廳管理者於網頁上呈現的內容及程式。

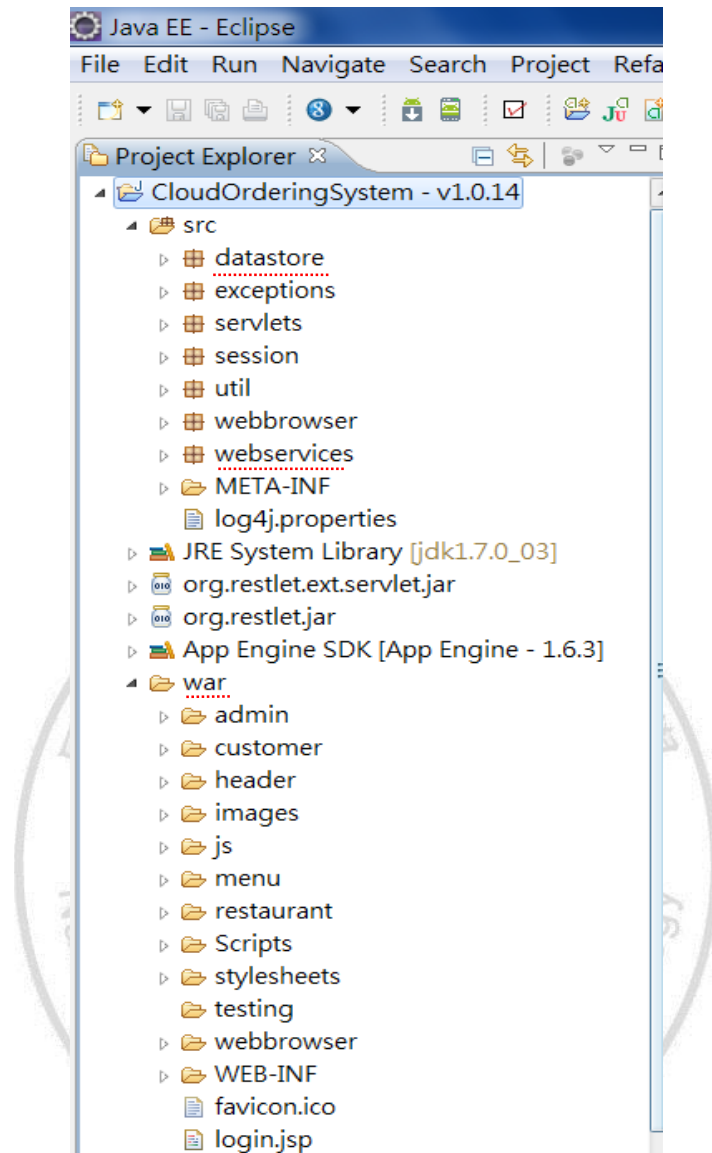


圖 5.1 開發程式碼架構

5.2 Android 系統程式開發

Android 專案開發中，一開始就必須選擇專案執行重要的核心，Android SDK 版本，選擇 SDK 版本也代表程式開發完成時目標在哪個

Android 版本上執行，幸運的是 Android 版本具有很好的向下相容性，Android 系統的功能都是累加的，新版本一定包含舊版本功能再加上新的功能，不過一旦使用到新版本的功能則無法順利運行在舊版本中。

因此儘管本研究開發時已經有 Android 2.3 以及 Android 4 的版本，也只需選擇在本研究開發上最低需求的版本 Android SDK 2.2 進行開發。

在本研究開發上有三個重要環節，GAE 雲端 Server 溝通、資料庫運作及畫面呈現與操控，如圖 5.2 紅色虛線部份由上而下分別為提供服務畫面呈現與操控的程式所在”smasrv.android.service”、與 GAE 雲端 Server 溝通之程式碼所在”smararv.service.communication”及 SQLite 資料庫運作程式碼所在”smasrv.service.sqlite”。

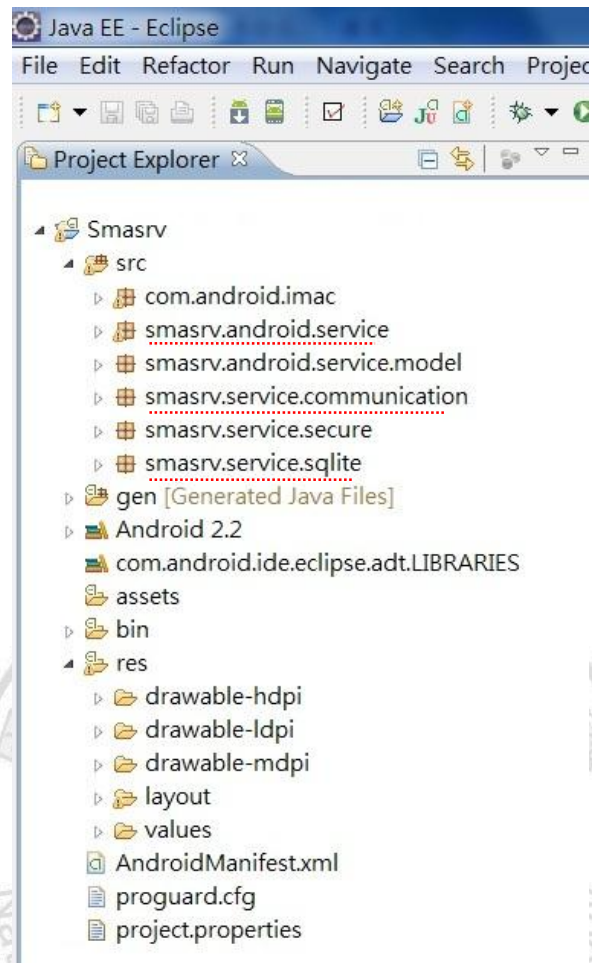


圖 5.2 Android 開發程式碼架構

程式運作流程則可參考圖 4.4，程式進入取得餐點服務時，會先運行 `smasrv.service.sqlite` 內程式碼取出 Android 端資料版本，接著運行 `smasrv.service.communication` 從 GAE 雲端 Server 取得最新版本號碼比對，若版本有差異則繼續運行其下的更新流程，此處所有與 GAE 雲端 Server 的通訊都以 JSON (JavaScript Object Notation) [20]的格式進行資料傳送與接收，比對版本及更新結束後運行 `smasrv.android.service` 其內

顯示與操控程式提供使用者取得需要的資訊與服務。

5.3 資訊加解密機制

由於本系統使用上會牽涉到使用者帳戶資料以及金流往來，因此在資訊安全上也是不可或缺的，本篇研究採用被廣泛使用的改良線性同餘法做虛擬亂數產生器的方法 [21]，並使用兩組獨特的 ID 作為其亂數種子產生原始密鑰，再使用對稱式區塊加密演算法對隱密資訊加密，最後使用無失真壓縮後，得到高度安全性的加密資料。由於採用對稱式區塊加密演算法以及無失真壓縮，解密的部分僅需反向操作加密過程即可得到原始資訊。

本篇研究使用到的兩組獨特 ID 為機械自動產生的唯一序列號，根據不同帳戶擁有者而不同，因此具有身份識別性以及不可竄改性，即使帳戶擁有者被竊取帳號及密碼，也會因為其中的唯一序列號不同而無法盜取身分使用。

5.3.1 密鑰產生方法

為了使本研究使用的對稱式加密方法具有好的抗盜用性，在密鑰的產生方式使用了兩種具有唯一性的序列號作為密鑰的產生元素：

1. Device ID：

本研究使用者對象為 Android 行動裝置用戶，而此種裝置具有唯

一的 Device ID，可透過 TelephonyManager.getDeviceId() 此 API 取得，且此 Device ID 跟隨著硬體設備因此具有即使行動裝置經過重置仍然不會被改變的特性，因此做為唯一性的序列號非常合適。

2. Application ID：

在使用者第一次註冊時會從伺服器產生一唯一且不重複的軟體序列號，作為該使用者密鑰產生的元素之一。

具有這兩個具有唯一性的序列號後，便個別將其設為我們虛擬亂數產生器的種子，產生兩組固定長度的二進制代碼，一組作為此亂數序列將插入的位置，另一組則作為將被嵌入明文的亂數序列。

5.3.1 加密機制

在取得兩組密鑰後，便將明文區塊依照第一組序列所代表的位置，插入第二組序列的內容，完成後將獲得一組被延伸的加密資料，我們再將這筆被延伸的加密資料做無失真壓縮，最後將獲得具有身份識別性的密文，如圖 5.3 所示。

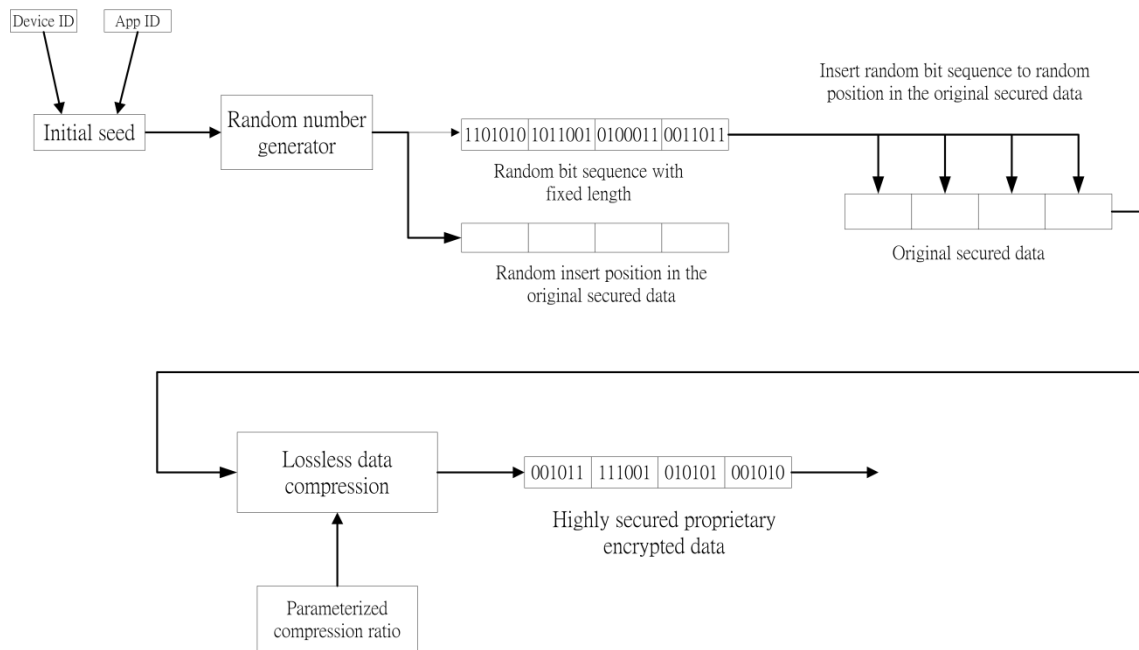


圖 5.3 個人識別式區塊加密方法

5.3.2 解密機制

如圖 5.4 所示，由於本篇採用對稱式區塊加密法，並且使用無失真壓縮，在伺服器取得密文後先對密文進行解壓縮動作即可取得經過原始經過延伸的加密資料，再使用該帳戶使用者的 Device ID 以及 Application ID 個別作為虛擬亂數產生器的種子產生兩組二進制代碼，此代碼就會與進行加密所產生的相同，便可利用一組代表位址，另一組代表插入的值這兩組代碼去將密文還原成明文。

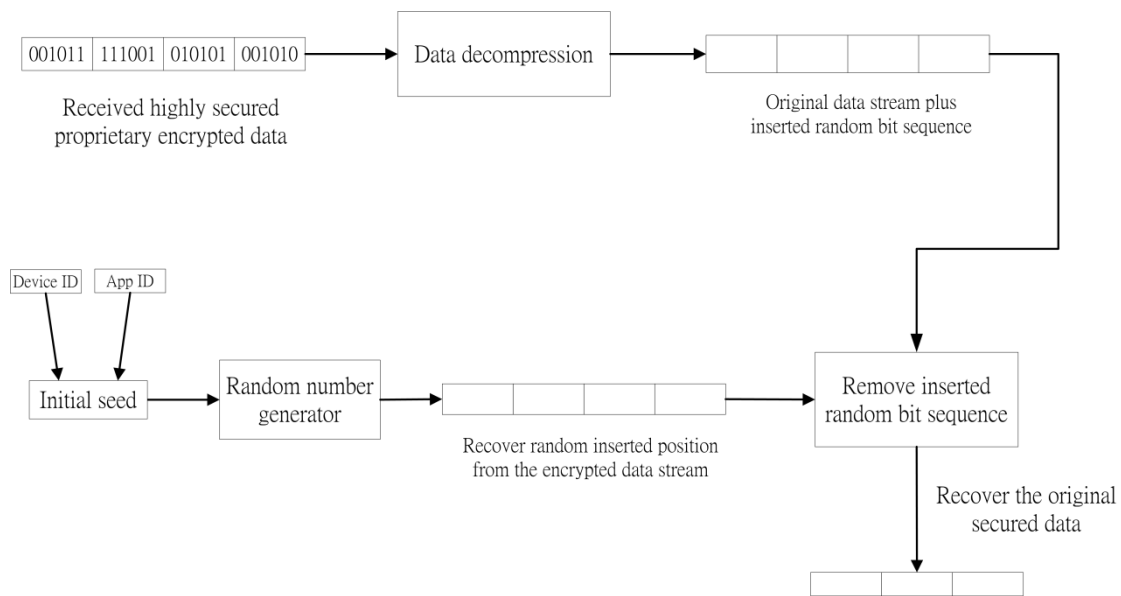


圖 5.4 個人識別式區塊解密方法

5.4 數據分析與比較

本研究希望在 Android 系統上建構出流暢的雲端整合系統，在本節將提出現有 Android 系統上提供類似服務技術的公司所研發的系統與本研究的程式系統效能上做比對，由於 Android 系統所能測試之數據來源有限，目前無法對他家系統測試程式運作流暢度，因此只測試佔用網路流量，額外的網路流量除了會影響系統效能，也會影響電量消耗甚至是使用者的網路數據費。

圖 5.5，5.6，5.7 為反覆使用包含 2626 外送外帶網 [1]、行動得來速 [2] 以及實作之系統各 10 次，並且 10 次內都進入同樣數家餐廳及瀏覽同樣 150 筆圖文資料所使用的網路流量。圖 5.5 為 10 次內所使用的

下載資料量，圖 5.6 為 10 次內所使用的上傳資料量，圖 5.7 為 10 次內總共消耗的網路資料量。

由圖 5.5 可看出 2626 外送外帶網 [20]在初次使用的流量是低於其他系統，但隨著每次使用流量也直線上升，而行動得來速 [21]與本篇研究成果同樣使用暫存資料方式，因此僅在第一次 Android 系統內無資料時做下載動作，之後使用都僅佔用非常低的下載量；圖 5.6 為使用之上傳資料量，2626 外送外帶網由於每次使用都必須要求資料因此上傳量仍舊是以同樣比例直線上升，行動得來速雖在第二次後上傳率有下降但卻同樣佔用許多上傳量，本篇研究成果於前幾節內容提到僅要求需要之資料版本資訊做比對，因此佔用網路量在第一次後就維持極低的消耗量，而由圖 5.7 可明顯從總流量統計結果看出本篇成果所消耗網路流量明顯低於其他系統。

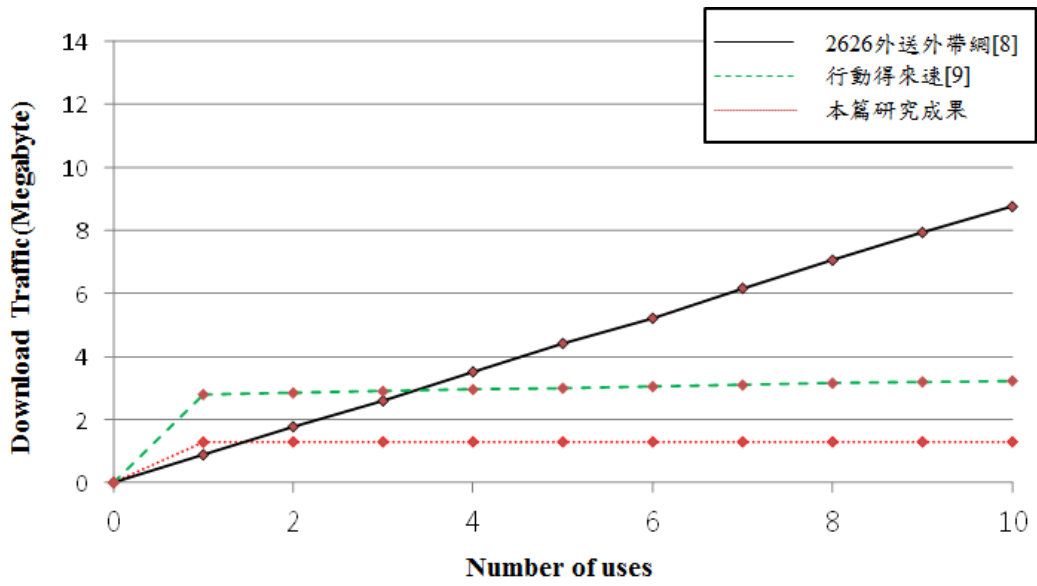


圖 5.5 下載流量測試分析結果

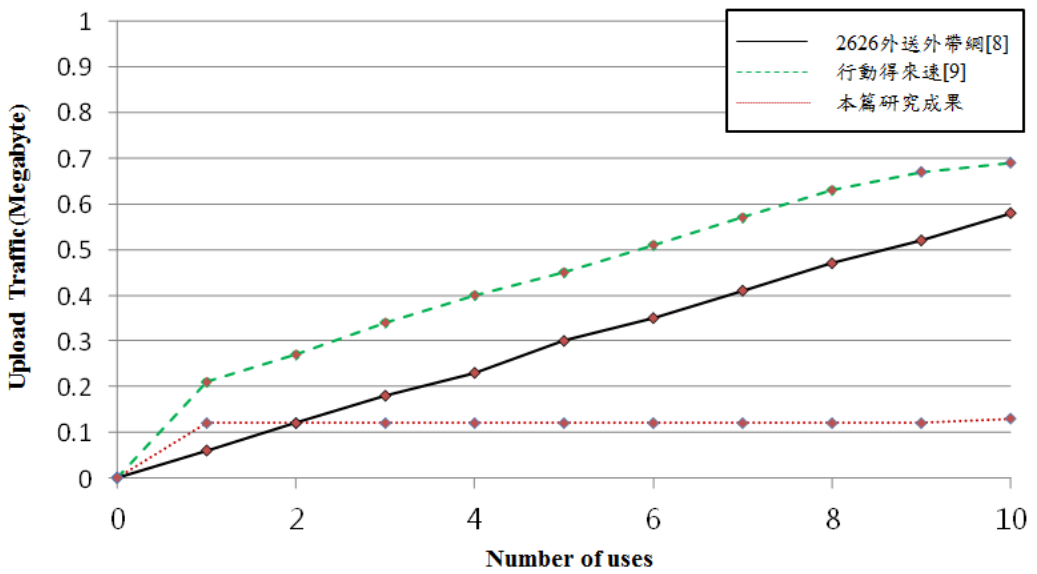


圖 5.6 上傳流量測試分析結果

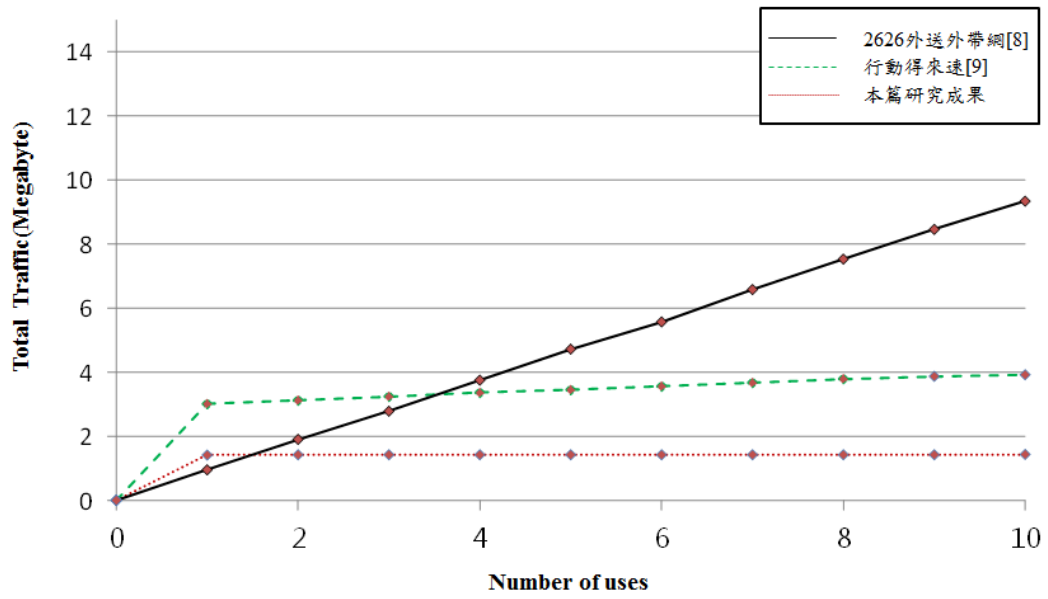


圖 5.7 網路流量測試分析結果



第六章 系統實測與介紹

在介紹各系統架構，及系統數據分析後，本章節將介紹實際測試各系統的運作以及功能，本系統於研究進行時同時也實際在市面上提供給廣大使用者試用，因此有些涉及私人隱私畫面會需經過處理。

6.1 GAE 雲端服務平台

圖 6.1 為本篇研究 GAE 雲端服務平台登入畫面，在章節 4.1 已介紹過，GAE 雲端服務平台架構依據權限將使用者分成 3 部分，此處登入後會依據登入帳號權限不同而進到不同的畫面。

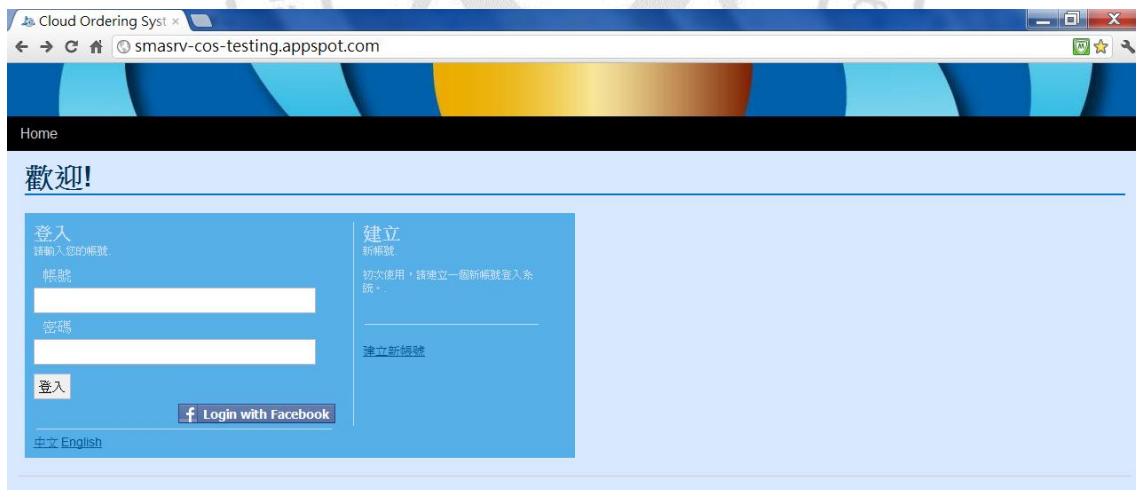


圖 6.1 GAE 雲端服務平台登入畫面

當登入為最高權限管理者時，如圖 6.2 可以編輯管理者帳號，圖

6.3 可以編輯合作店家之帳號。

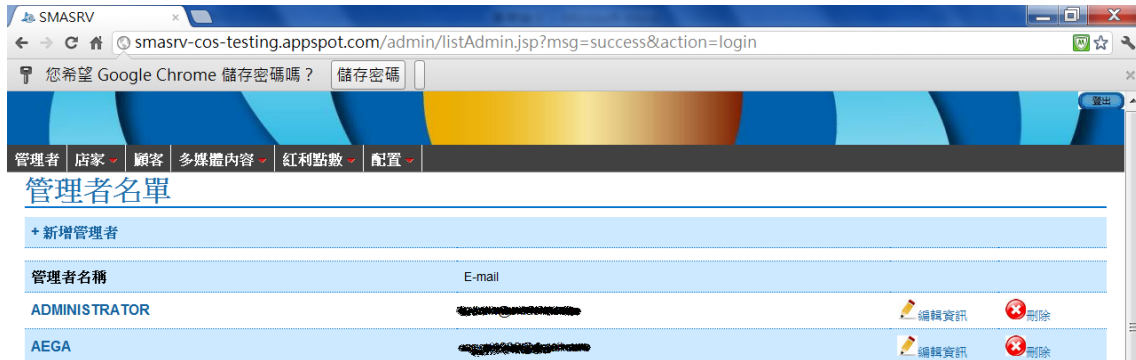


圖 6.2 GAE 雲端服務平台最高管理者登入畫面



圖 6.3 GAE 雲端服務平台最高管理者管理合作店家畫面

當伺服器判斷登入身分為店家管理者權限時，則會顯示店家相關管理功能的介面，如圖 6.4 為管理使用者對店家的訂單內容，圖 6.5 為店家管理提供的服務商品內容，圖 6.6 則是新增服務商品內容的介面，讓店家管理者建立商品的資訊到 GAE 雲端服務平台雲端資料庫，建立完成後便可立即在 Android 系統上瀏覽及取得。

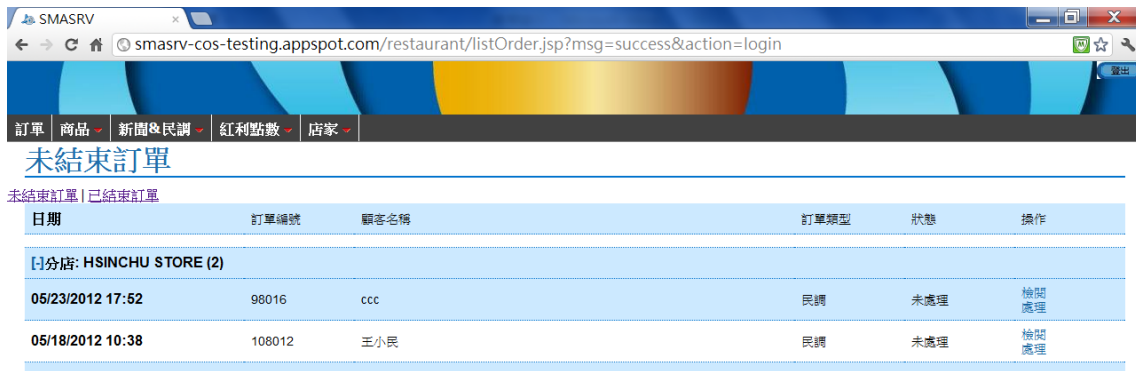


圖 6.4 GAE 雲端服務平台店家管理者管理訂單畫面

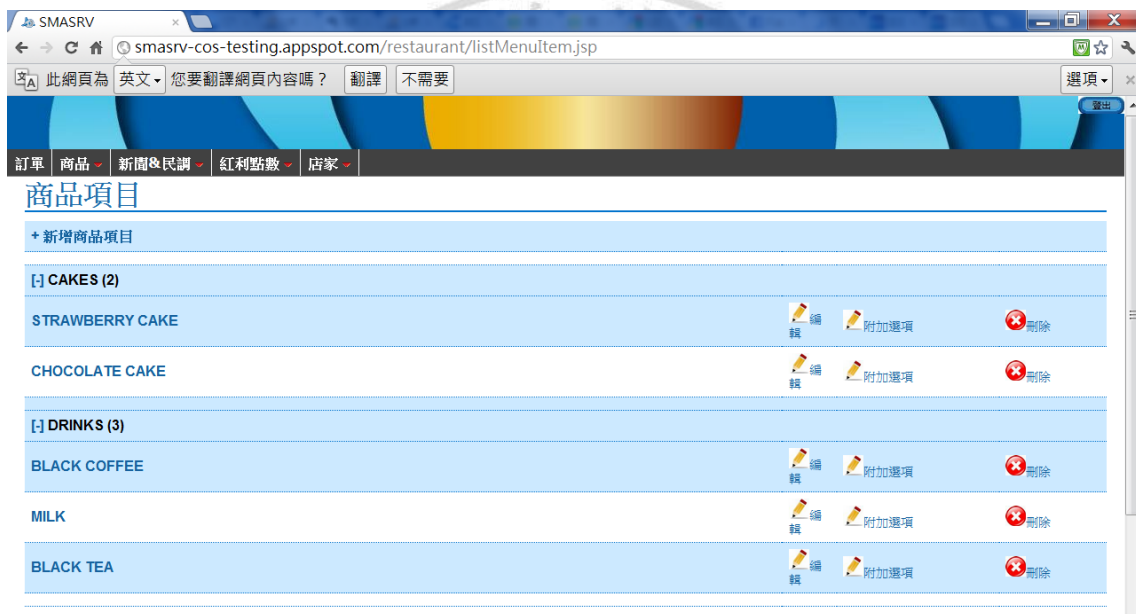


圖 6.5 GAE 雲端服務平台店家管理者管理商品畫面

The screenshot shows a web browser window with the URL `smasrv-cos-testing.appspot.com/restaurant/addMenuitem.jsp`. The page title is '商品項目' (Item List). The main content is a form titled '新增商品' (Add New Item). The form fields are as follows:

- 商品類型 ***: Cakes (dropdown menu)
- 商品名稱 ***: Blueberry Cake (text input)
- 價格 ***: 130 (text input)
- 折扣(0~1) 9折=0.1, 8折=0.2, 85折=0.15, 7折=0.3...**: 0.1 (text input)
- 描述 ***: 高甜度藍莓配上鬆軟可口的蛋糕 (text area)
- 商品圖片**: 選擇檔案 未選擇檔案 (file upload button)
- 準備時間 (分鐘)**: (empty text input)
- 供應中 ***: 有 無 (radio buttons)
- 供應時段 ***: 整天 (dropdown menu)
- 補充說明**: (empty text area)

At the bottom left, there is a checkbox labeled '繼續新增商品' which is checked. At the bottom right, there are two buttons: '關閉' (Close) and '儲存' (Save).

圖 6.6 GAE 雲端服務平台店家管理者新增商品畫面

6.2 Android 開發實測結果

本節介紹 Android 開發實測之成果，各圖示成果展現 Android 接收 GAE 雲端資料後顯示畫面。圖 6.7 為店家列表，為避免使用者因為店家過多造成難以瀏覽的狀況，上方使用分類篩選功能可以讓使用者僅瀏覽想看的類別項目如圖 6.8 所示，圖 6.9 為篩選過後畫面。



圖 6.7 Android 店家列表畫面



圖 6.8 Android 店家列表分類篩選畫面



圖 6.9 Android 店家列表篩選後畫面



圖 6.10 Android 店家分店列表

如圖 6.11 所示在進入條列式列表後，在取得文字資料後就會立即顯示，需要較多下載時間的圖檔另外使用線程分開下載，並暫時使用載入中的 Icon 代替，圖檔下載完畢後才刷新頁面顯示出來，如此一來與全部下載完畢的速度提升很多。圖 6.14 所示為商品訂購後列表立即顯示該商品訂購數量，幫助使用者在眾多商品列表中也可以清楚知道哪些商品買了多少。



圖 6.11 Android 下載圖檔中畫面



圖 6.12 Android 圖檔下載完畢畫面



圖 6.13 Android 商品詳細資訊畫面



圖 6.14 Android 商品訂購後列表畫面

在商品挑選結束後便進入一個購物車模式畫面，瀏覽該次購買了哪些項目以及金額總覽，同時也可以立即進行項目的編輯刪除，如圖 6.15 所示。在訂購流程結束後，便可在訂單紀錄裡觀察訂購物品的店家處理狀況或是瀏覽自己已經購入了哪些商品，如圖 6.16、圖 6.17 所示。



圖 6.15 Android 購物車模式畫面



圖 6.16 Android 訂單紀錄列表畫面



圖 6.17 Android 訂單紀錄詳細資訊



第七章 結論及未來工作

本文將伺服器端程式及資料庫建構在 GAE 上，展示出可以完成一個無須自行管理、維護硬體設備的雲端系統及資料中心。

透過實驗數據可以看出本篇強調結合本地端資料庫系統與雲端技術相互配合下的架構在 Android 系統上短期及長期使用上不會消耗大量網路頻寬，也將提昇系統速度及降低耗電量。

本文使用之行動裝置為 Android 系統，當前市場仍有很大的 iOS 使用者以及其他作業系統使用者，但其軟體架構與雲端服務卻不會因為行動裝置作業系統差異而無法實現。

未來將繼續擴大雲端平台所能應用的服務類型，並將 Android 系統架構朝智慧個人化邁進。

參考文獻

- [1] 2626 外送外帶網， https://play.google.com/store/apps/details?id=ctp.shop&feature=search_result
- [2] 行動得來速， https://play.google.com/store/apps/details?id=com.o2oist.jashami&feature=search_result#?t=W251bGwsMSwyLDEsImNvbS5vMm9pc3QuamFzaGFtaSJd
- [3] 李浩維，雲端運算與服務的研究與應用以「Google App Engine」為例，嶺東科技大學數位媒體設計研究所碩士論文，2010 一月。
- [4] 廖啟志，運用雲端儲存技術建構企業私有儲存雲，亞洲大學資訊工程系碩士論文，2010 六月。
- [5] Hao Yu; Moreira, J.E. Dube, P.; I-hsin Chung; Li Zhang, “Performance Studies of a WebSphere Application, Trade, in Scale-out and” IPDPS, IEEE, pp. 1-8, March 2007.
- [6] Geng Lin; Fu, D.; Jinzy Zhu; Dasmalchi, G., “Cloud Computing: IT as a Service”, IT Professional, IEEE, Vol. 11(2), pp. 10-13, March 2009.
- [7] Google.com,”Appengine,”2011, <http://code.google.com/appengine/>.
- [8] A. Bedra, “Getting Started with Google App Engine and Clojure,”Internet Computing, IEEE, Vol. 14(4), pp. 85-88, 2010.
- [9] J. Dean, S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, OSDI, pages 137–150, 2004.
- [10] Ramanathan, S., Goel, S., Alagumalai, S., “Comparison of Cloud database: Amazon's SimpleDB and Google's Bigtable”, ReTIS, pp. 165-168, Dec. 2011.
- [11] Jianye Liu; Jiankun Yu,”Research on Development of Android Applications”, Intelligent Networks and Intelligent Systems (ICINIS),

pp. 69-72, Nov. 2011.

- [12] 長高科技，” 行動裝置嵌入式系統與軟體” ，長高科技，2009。
- [13] 楊文誌， “Google Android 2 程式設計與應用” ，旗標出版，2009。
- [14] 余志龍、陳昱勛、鄭名傑、陳小鳳， “Google Android SDK 開發範例大全第三版” ，精誠資訊, 2011。
- [15] OL. Google Android Developers, Android Develop Guide, <http://developer.android.com/guide/topics/fundamentals.html>
- [16] 章詔博，以 Android 為平台的擴增實境導覽及大地尋寶系統，2011 七月。
- [17] Il-Yeol Song, Froehlich, K., “Entity-relationship modeling”, Potentials, IEEE, Vol.13(5), pp. 29-34, Jan 1995.
- [18] Maoqiang Song, Haiyan Song, Xiangling Fu, “Methodology of user interfaces design based on Android”, ICMT, IEEE, pp. 408-411, July 2011.
- [19] 袁葆宏、劉豐豪、吳建和， “精解 Java SE6 程式語言” ，旗標出版，2008。
- [20] Introducing JSON[EB/OL], <http://www.json.org>.
- [21] 陽欽舜，應用虛擬亂數於動態對稱式區塊加密機制之研究，2009 六月。