

Apple iOS 行動應用程式之開發 --

以校務行政系統為例

吳昌憲 陳建翔

國立虎尾科技大學資訊管理系

雲林縣虎尾鎮文化路 64 號

cswu@nfu.edu.tw

摘要

行動開發技術已經 IT 界上成為最受矚目的熱門話題之一，隨著通訊、嵌入式系統等技術的成熟，手機已經不再只是個通話工具，而是集合了通訊、網路、商務、影音、娛樂等多種功能於一體的高科技產物。在發展的過程中，手機戰場已經從硬體延伸到軟體與服務，智慧型手機的新價值在於行動市場的發展，這也是程式開發者的新機會。

本研究使用 Apple 公司的 iOS 系統，來做為智慧型手機發展系統開發的平台，強調校務系統可以達成便利性，利用該裝置的行動性建置一套屬於學生的個人助理系統，主要是可方便學生族群的使用者能夠管理自身相關的事務。

系統架構在 iOS 的裝置上，無論是 iPhone 或是 iPad 皆是使用 Objective-C 的原生程式碼進行開發應用，在本研究中最主要的目的為改善系統的方便性，讓使用者能隨時隨地，都能輕鬆使用。

關鍵字：智慧型手機、iOS、校務行政系統

第一章 緒論

1.1 研究背景

行動通訊服務產值隨著新興市場以 3G 服務等因素持續成長，2012 年全球行動通訊用戶約為 65 億戶，未來整體通訊產業的發展，仍以行動通訊服務為主。美國市場研究公司 Gartner 發佈報告稱，2012 年第二季度全球手機終端使用者銷量達到 4.19 億部，較 2011 年第二季度減少 2.3%。智慧手機銷量同比增長 42.7%，在全部手機銷量中占 36.7%。[6,8]

根據數據分析公司 Flurry 公布的資料，指出 iOS 和 Android 裝置在全球的接受度非常驚人，滲透速度之快超越史上所有的消費性科技，比 80 年代個人電腦快了 10 倍；比 90 年代的網際網路快兩倍；也比當紅的社群網站快三倍。[6,7]

五年之中，所謂的「智慧型裝置」正在迅速擴散，已超出典型的早期採用市場，如北美和西歐等，創造出一個真正的全球市場。Flurry 在這次的調查中使用了超過 20 萬個 app(約涵蓋了 90% 的 iOS 與 Android 裝置)的資料，估計在今年七月全球共有六億四千萬個 iOS 與 Android 裝置被使用。

下圖(圖 1)則是以「啟用的 iOS 與 Android 裝置」成長率來看。與去年相比，中國以 401% 排名第一，其次是智利的 279%，巴西以 220% 排名第三。我們可以發現前 10 名都是開發中國家。

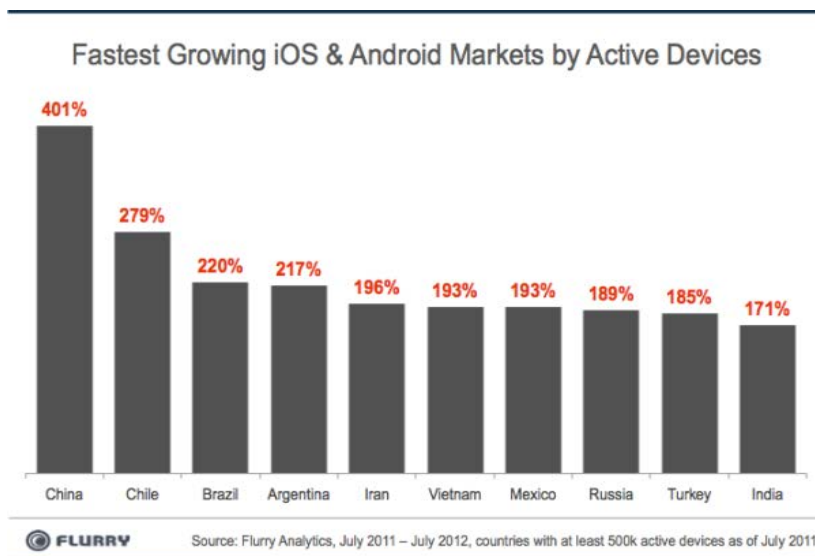


圖 1，資料來源：Flurry

除了成長最快速的國家，Flurry 也研究了滲透率最高的幾個國家，15 – 64 歲的人口中擁有 iOS 與 Android 裝置的比例，前五名分別是：

1. 新加坡：92%
2. 香港：87%
3. 瑞典：86%
4. 澳洲：79%
5. 美國：78%

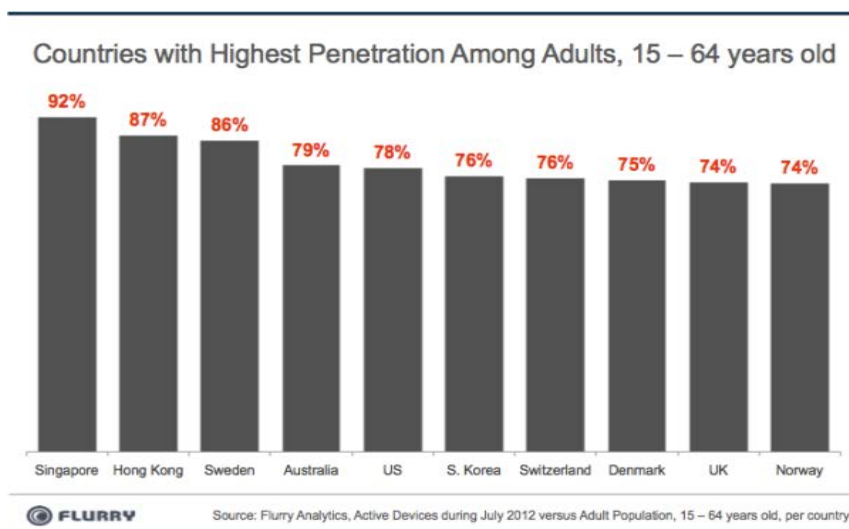


圖 2，資料來源：Flurry

無線通訊網路的成熟，電信營運商不再是以過往的方式來經營，而是透過無線通訊網路技術，提供使用者多元化和即時性的產品，讓使用者能在任何時間、任何地點都可以享受到便利的行動增值服務。[6]

1.2 研究動機與目的

iPhone 改變了一切。你可能沒注意到每家廠牌推出最新最酷的手機外觀和操作模式都像極了 iPhone，而不是他們在 iPhone 發售之前賣的那些手機。Apple 的 iPhone 廣告里展示著各種應用，iPhone 上市至今，已經改變了許多東西，在 iPhone 還沒出現之前，想要在小型設備上開發軟體是充滿風險的，在當時，每個設備有各自不同的特性，如螢幕分辨率、輸入技術等，以及不同的技術實現標準，迫使開發者們走上一條艱辛的「編寫一次，到處測試」之路。但在 iPhone 或 iPad 這種使用 iOS 系統的載具上，表現出高度良好的一致性和連貫性。原先故步自封的營運商們現在爭先恐後地開始營運他們自己的 App Store，就是害怕 iOS 那些酷炫的應用潛移默化地將他們的用戶帶走。

根據國家通訊傳播委員會（NCC）資料，如下圖(圖 3)所示，透過行動上網的用戶逐年上升，使用行動上網服務的總用戶數也達到 2070 萬戶，佔行動通信用戶比率 71.3%。由此可知，行動上網人口的增加將提升行動內容的需求，而豐富有趣的行動內容將會吸引更多的行動上網用戶。

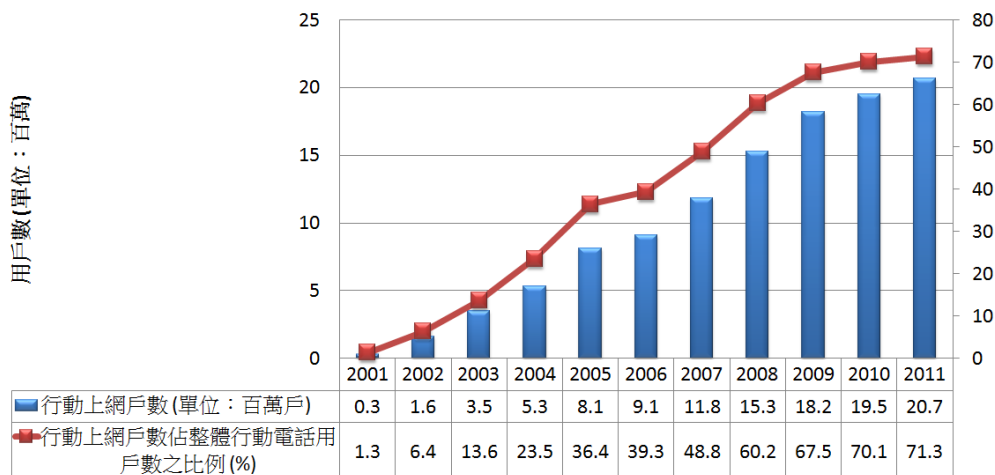


圖 3，資料來源：NCC 國家通訊傳播委員會，本研究整理

行動裝置已成為當今科技產業最火紅的產品之一，其中又以近年來突然崛起的智慧型行動裝置更為人所知，藉由實踐行動裝置應用程式的系統，並選擇以筆者的母校「吳鳳科技大學」的學生校務行政系統為例，創作出一套專屬於行動裝置上的應用。其中無論是請假或是選課，即使所在的工作場所無法使用電腦操作校務系統，也可透過這套專為行動裝置所開發的應用程式，在任何時間、任何場所都能享受到服務。

綜合以上所述，隨著國際上行動裝置的普及率的成長加速，要成為一個軟體開發者，科技史上沒有比現在更好的時機。

第二章 文獻探討

2.1 iOS 的歷史演進

iPhone近年來人氣爆紅，是由蘋果公司(Apple Inc.)所生產的行動裝置。自2007年6月29日發布後，手機的世界從此改變，也讓蘋果公司市值躍居全球第一，第一代iPhone發售以來，每年都有新的款式發售。基本上新機種的共同特徵是，它們都搭載了比前一代機種處理速度更快的CPU，更多的記憶體容量，以及畫面解析度提昇等功能。

iPhone是一種被稱為智慧型手機(Smart Phone)的行動裝置，兼具行動電話和PDA的功能。智慧型手機的特徵是可以像個人電腦一樣使用各式各樣的應用程式。2008年6月9日發布了iPhone 3G，對應3G(第三代行動通訊系統)，其CPU、記憶體影像處理功能和第一代相同，但搭載了GPS等新功能，且有8GB容量(黑色)和16GB容量(黑白兩色)等兩種選擇，iPhone系列正式開始支援WCDMA標準(3GPP)。

2009年6月8日，蘋果在全球開發者大會2009(WWDC 2009)上宣布了新一代的iPhone 3GS，S的意思表示速度，iPhone 3GS整體運行速度是iPhone 3G的2倍，硬體上也新增了錄影與自動對焦等功能。

除了iPhone之外，iPad也是相關的產品。2010年1月發布的iPad是比iPhone還大的平板電腦。因為它有比iPhone更大的畫面，所以在瀏覽網頁或是電子書時更加舒適，也能在比iPhone更具魄力的環境下玩遊戲。2010年6月7日，iPhone 4閃亮登場，推出了16GB和32GB兩種容量，並有黑白兩色可供選擇。iPhone與iPad共同搭載的作業系統被稱為iOS。以往這個系統曾經被稱為iPhone OS，不過從2010年6月釋出的4.0版本開始改稱iOS。

2011年3月2日蘋果公司發布第二代平板電腦iPad 2，主要面向是圖書、期刊、遊戲、影片、音樂等視聽媒體內容相關的平台，除了在硬體上有所改變之外，系統處理效能也優於初代iPad。2011年10月4日，蘋果公司在美國加州總部舉辦「Let's talk iPhone」，發表iPhone 4S，搭載iOS 5系統，具備了Siri語音助理在內的眾多新功能，以及為數不小的硬體設備增進。10月5日，賈伯斯逝世，也因此有人稱iPhone 4S就叫賈伯斯紀念機(iPhone 4S, iPhone for Steve)。

2012年3月7日，蘋果公司舉行第三代平板電腦發表會，並命名為「The New iPad」，在記者會中，執行長庫克(Tim Cook)談到「後PC世界」，在這個世界中，個人電腦不再是個人數位生活的中心，而第三代的iPad將會是「後PC世界」的主要貢獻者。2012年9月12日，新一代的iPhone 5降臨，全新一新的設計，顯示器的尺寸「加長」至4吋，但寬度仍和iPhone 4S一樣寬，因此一樣單手就能輕鬆使用。2012年10月23日蘋果公司推出首部設有7.85吋螢幕的iPad mini，它是iPad系列中體積最細小的型號。同日，在同一場發表會中，蘋果公司更是宣布了第四代平板電腦，其名為「iPad with Retina Display」，在蘋果網站上的官方中文名稱為「iPad 配備 Retina 顯示器」。^[5]

多家市場研究機構的調查顯示，Google雖然已將Android打造成最受歡迎的智慧手機操作系統，但

由於 Android 平台的碎片化、應用程式發布流程繁雜等問題，應用開發者正漸漸失去對這個平台的興趣，蘋果 iOS 仍然是他們的首選平台。

據行動軟體數據分析公司 Flurry 公布的最新調查結果，今年 1 月，相比於蘋果 iOS 平台，開發者為 Android 平台開發的應用程式數量低於去年同期水平。Flurry 調查發現，開發者為 Android 平台和 iOS 平台開發的新應用程式數量比例為 1:3，而在去年，此比例為 2:3。

2.2 為什麼選擇 iOS

智慧型手機的風潮吹啊吹的，就讓我們來聊聊當前的兩大強者，iOS 和 Android 吧，Android 的市佔率超越了 iOS，它真的有那麼好用嗎？

從使用者操作的角度來看，「觸控螢幕」會是習慣一堆按鍵的傳統手機的使用者，遇到的第一個關卡 — 「操作體驗」。iPhone 一貫使用觸控螢幕加上只有一個 Home 鍵的簡單設計，就如同賈柏斯說過的那句：「我討厭笨蛋，但我做的產品連笨蛋都會用。」直覺簡單的操控介面，為觸控螢幕手機帶來極佳的使用者體驗。這種設計方式也帶來市場上的模仿，目前市面上的觸控螢幕手機，大概就長那樣。

從機器更新的角度來看，在版本更新與擴充性上 Android 版本更新速度快，但壞就壞在 Android 作業系統更新頻率快。Android 以相對開放的授權，降低了各家廠商的進入門檻。然而，以手機市場來說，大概半年就是一個產品生命週期。在這麼短的產品生命週期中，如果你是手機廠商，會把資源投可以開創新商機的新機開發，還是已經賺不到錢的舊機升級？所以，即便新版本一直發佈，即便有良心的廠商，舊機也頂多獲得極少數的更新，新釋出的版本多半的情況是看得到吃不到。反觀來看，2010 年推出的第一代 iPad，到現在系統一樣能從 3.2 升到 4.2、5.0，甚至連 5.1 都沒問題。

從開發者的角度來看，同一款軟體，開發者能從 App Store 上賺到的錢是 Android Market 的 6 倍。如果你是軟體開發者，會選擇在螢幕尺寸單純、付費機制健全的 iOS 上開發軟體，還是螢幕尺寸眾多，並停止對台灣區付費 APP 銷售服務的 Android Market？

iOS 主力集中在安全性和設備管理性上，建立了一套整體的解決方案，反觀其他平台都將注意力集中在「超越」iOS 上，卻過於急躁與追求功利，而忽視了用戶需求。

2.3 iOS 5 的開發和以前有什麼不同？

在進行 iOS 應用程式開發時，總是會接觸到 Interface Builder(簡稱 IB)，IB 有著悠久的歷史，它從 1988 年起就存在，並被用來開發 NextSTEP、OpenSTEP、Mac OSX 以及現在的 iOS 應用程式，Interface Builder 認識兩種格式的副檔名：.nib 和 .xib。以 .xib 為副檔名的檔案可看成是 .nib 檔的 XML 版本。.xib 相對於 .nib 而言可說是比較新的，不過仍有開發人員把它們都看成 .nib 的檔案。

Storyboard 是 iOS 5 的新功能，它的出現讓人眼睛為之一亮，Storyboard 代替了歷史悠久的 NIB/XIB(其實 StoryBoard 還是來自於 NIB/XIB，不過開發人員已經不須再透過 NIB/XIB 進行開發了)，在 iOS 5 以前我們可以透過 Interface Builder 去編輯 View Controller 的樣式，而 StoryBoard 的導入直接的把整個 Workflow 也一併的加入，藉由 StoryBoard，可以看出整體應用程式的概括。

相較於 NIB/XIB，Storyboard 機制的優點是什麼？

1. 能夠減少很多跟 View 相關的程式碼
2. 能夠使 View 和 Controller 進一步解耦
3. 能夠優化程式的「頁面流程」，使程式的結構更清楚

要理解這些優點，我們要先對 NIB/XIB 有個基本的認識。NIB/XIB 是和 ViewController 相關聯的，ViewController 都有它所對應的 NIB/XIB。NIB/XIB 檔的作用是用來描述使用界面還有初始化的物件。NIB/XIB 中的配置其實都能透過程式碼的方式來實現，但是為了減少設置屬性還有編寫程式碼所浪費的時間，使用 Interface Builder 來繪製界面更能使開發人員能集中精神在開發上。

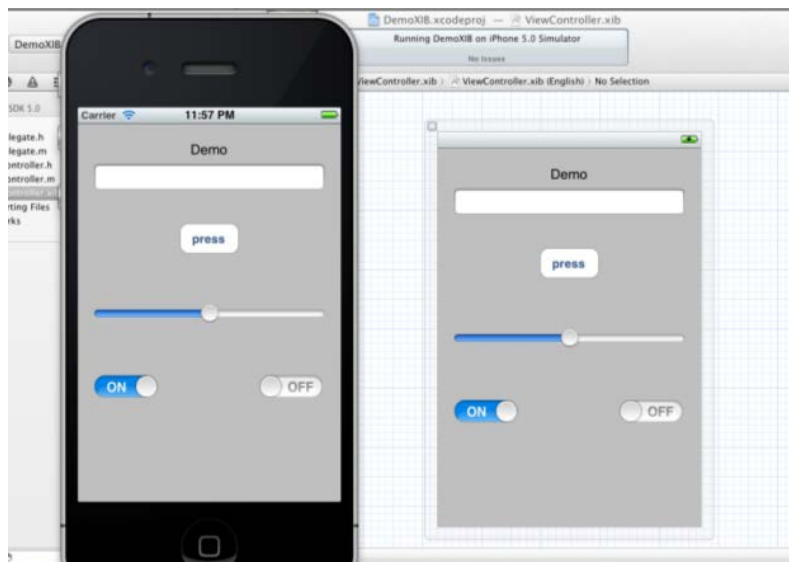


圖 4，XIB 檔的使用圖

而 StoryBoard 的出現強化的這方面的功能。NIB/XIB 檔是沒有辦法描述從一個 ViewController 到另一個 ViewController 之間的關係，只能透過程式碼的方式來控制動作。最常使用的 `-presentModalViewController:animated:` 以及 `-pushViewController:animated:` 兩個方法，這種使用程式碼來進行 ViewController 之間切換的行為，在 StoryBoard 中將成為歷史，取而代之的是 Segue 的出現。Segue 定義了 ViewController 之間溝通的橋樑。在 StoryBoard 裡，我們只需要把 ViewController 之間用 Segue 連起來就可以了。這就是上面所提到的第一個優點。

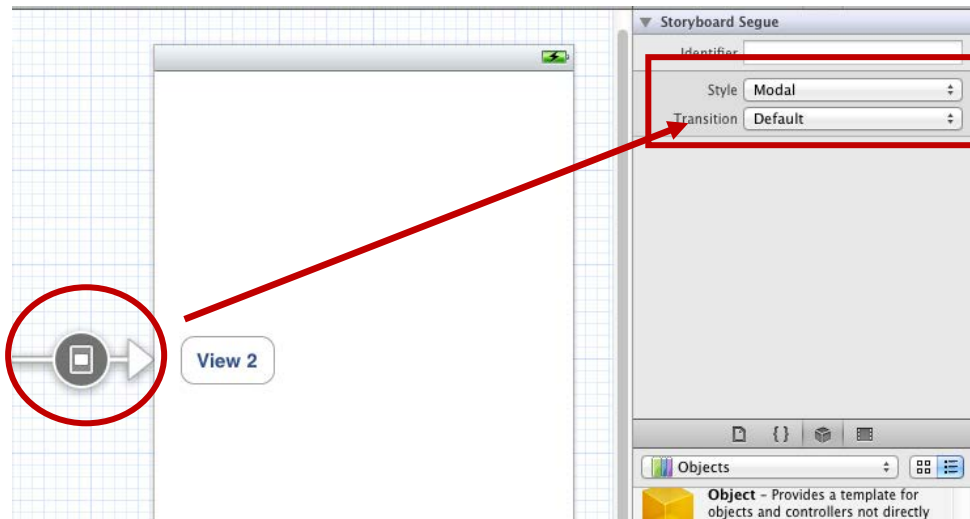


圖 5，Segue

要用好 StoryBoard 的機制，那麼必須嚴格遵守 MVC 原則。要讓 View 和 Controller 充分解耦，並且不同的 Controller 之間也要充分解耦，否則程式的邏輯就會亂了步調，很難理解。

就好比許多人喜歡直接把 AppDelegate 當成 ViewController 來使用，直接從 AppDelegate 中與 .xib 進行溝通。這是個非常不好的習慣，AppDelegate 的功用很單純，就是在處理 UIApplication，而不是負責使用者界面的處理。雖然 UIWindow 也是屬於 UIView 的子類別，但由 ViewController 來負責處理 View 才是正確的做法。

Xcode 的開發模版經常發生改變，特別是從 Xcode 4 發布之後，開始鼓勵使用 UIWindow 的 rootViewController 屬性來指定第一個視窗的 ViewController，以確保 AppDelegate 專注於它應該做的事情。在 Xcode 4.2 引進了 StoryBoard 之後，AppDelegate 已經不管 ViewController 的事情了，第一個視窗所使用的 ViewController 可以在 StoryBoard 中設置，應用程式的進入點就能從 StoryBoard 的設計圖上一目了然。這就是上面提到的第二個優點。

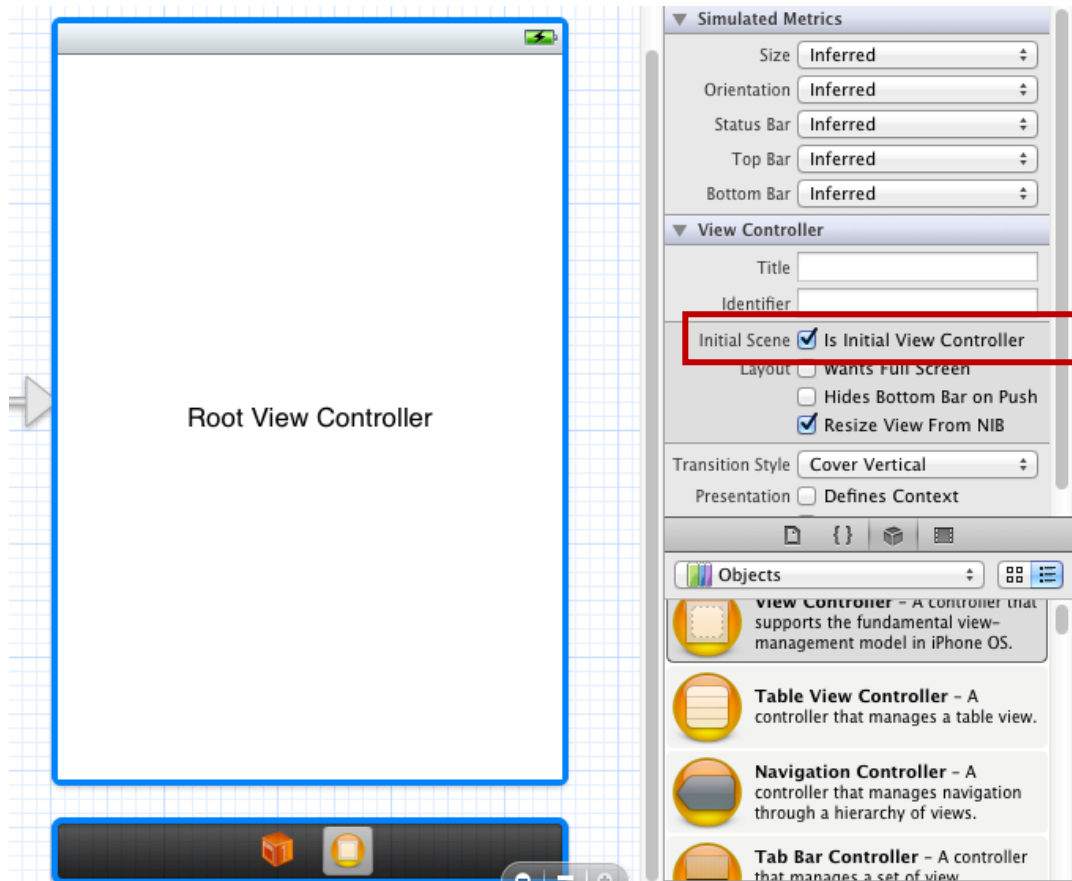


圖 6，畫面進入點

至於第三個優點，就是 StoryBoard 了。Storyboard 能夠包含一個程式所有的 ViewController 以及它們之間的連接。因此，Storyboard 甚至可以作為應用程式的「設計圖」來用了。理想情況下，在程式開發接近尾聲的時候，我們只需比對 StoryBoard 的「流程」和最初的设计「流程」，就是知道程式有沒有「走樣」了。

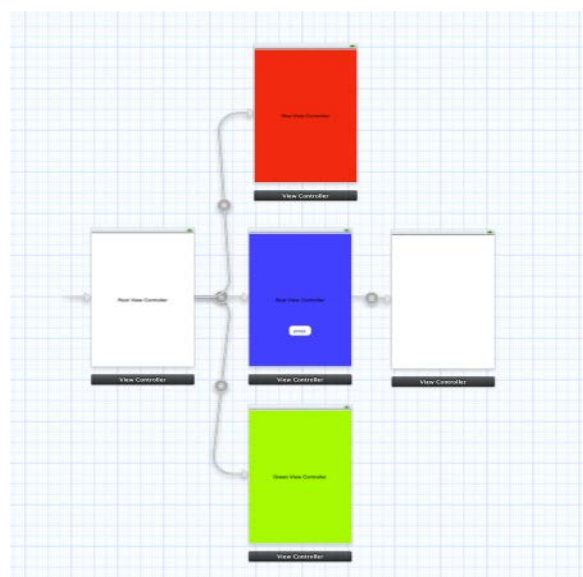


圖 7，Storyboard

說完了以上各項優點後，我們來看看從 NIB/XIB 到 StoryBoard 的轉移，有哪些理解和實踐上的改變

呢？

首先，自然是 `ViewController` 不再需要 `NIB/XIB` 了，以前在 `NIB/XIB` 上做連接 `Outlet` 和 `Action` 的操作都可以在 `Storyboard` 上完成了。

第二，獨立於 `ViewController` 的 `View` 是不能出現在 `Storyboard` 裡的，`View` 必須透過 `ViewController` 來管理，否則 `Xcode` 將會出現警告，這是相當忌諱，原則上我們不會希望它出現在開發專案中。

第三，`ViewController` 之間所使用的溝程式碼已經是歷史了，用 `Storyboard` 可以直接可視性地連接不同的 `ViewController`，透過 `Segue` 的強大功能，將過往的動作化繁為簡，更為輕鬆上手。

第四，`UIWindow` 物件的作用被進一步淡化，甚至可以這麼說：其實很多程式根本不需用到 `UIWindow` 物件。`AppDelegate` 也不再被鼓勵用來處理 `ViewController`，從 `Xcode 4.0` 開始，在新建專案(create new project) 時自動出現了 `RootViewController` 的 `.h` 和 `.m` 檔，處理 `ViewController` 將在此進行。

最後，寫優質的程式碼，嚴格遵守 `MVC` 設計模式，這樣不僅能夠讓你用好 `Storyboard`，也能幫助你理解 `Storyboard` 的原理。

`Storyboard` 是非常好的鼓勵 `MVC` 和程式碼解耦的手段，能夠讓開發人員寫出更加容易維護的程式。不過對於初學者來說，確實是個對理解力的小挑戰。但也用不著太擔心，一旦突破了理解障礙，你就會發現 `Storyboard` 也非常好用，就像最初理解 `NIB/XIB` 時，`Outlet` 和 `Action` 「拉來拉去」，看起來很神奇，理解後發現原來「拉來拉去」也不會困難到哪去。[4,9]

第三章 研究方法

3.1 自訂 Table View Controller

本系統最常採用的就是 `TableViewController`，那究竟要如何自行定義 `Table View` 呢？在 Xcode 所提供的 `Table View` 畫面如下圖(圖 8)所示：

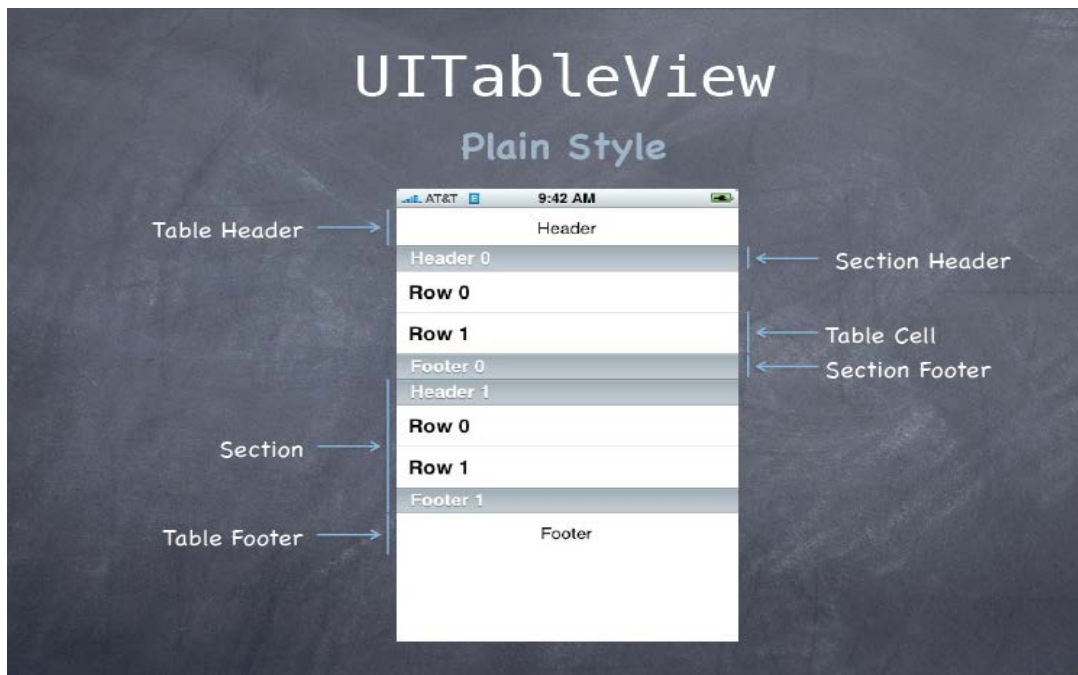


圖 8，圖片來源：Stanford, CS193P iPhone Application Development, Fall (2011-12)

在點擊 `Table Cell` 後請開啟 `Attribute Inspector`，可以發現 `Table Cell` 的 `Style`，除了內建的幾項外，其中還有一項是 `Custom`，如果要自定樣式就要從這裡著手。

首先要新增檔案，這邊我們要繼承的父類別要是 `UITableViewCell`，如下圖(圖 9)：

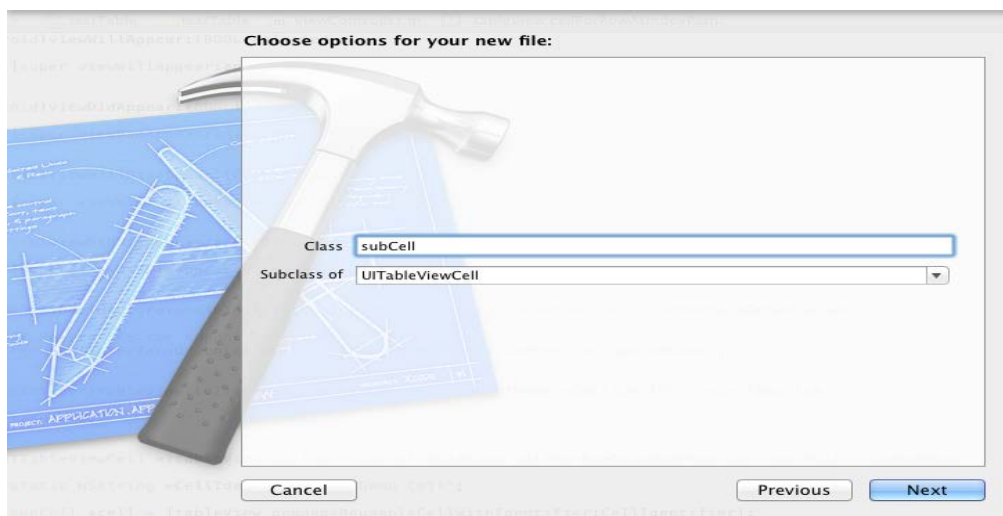


圖 9，繼承自 `UITableViewCell`

按下 Next 新增後，開啟 .storyboard 檔，目前 storyboard 中只有一個空白的 TableViewController，先選取其中的 Table Cell 後，然後按下 Alt + Command + 3 (Identity Inspector)。可以看到目前的 Table Cell (圖 10) 它是去繼承 UITableViewCell，而我們剛才所新增的檔案正巧也是繼承 UITableViewCell，因此可以將圖中的 Class 改成剛才新增的檔案。

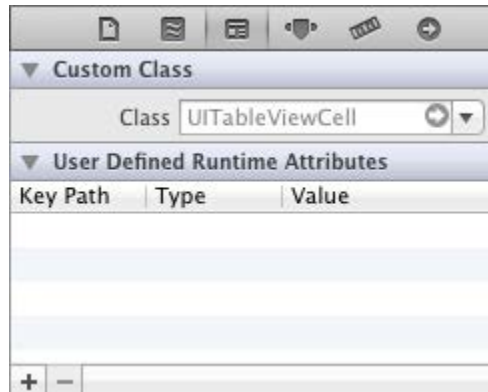


圖 10，更改 TableCell 所繼承的來源

接著回到畫面配置的地方來配置所想要的畫面，下面為示範圖，讀者們也可以依此為例製作出自己所想要的畫面。



圖 11，StroyBoard 中所見示意圖

再來請開啟剛才新增的 .h 檔，以此範例為主就是 subCell.h 檔。在上圖中共有「Demo1」和「Demo2」兩個 UILabel，因此須新增兩個 outlet。subCell.h 的總體程式碼如下：

```
#import <UIKit/UIKit.h>
```

```
@interface subCell : UITableViewCell
```

```
@property (nonatomic, strong) IBOutlet UILabel *demo1;
```

```
@property (nonatomic, strong) IBOutlet UILabel *demo2;
```

@end

有了 outlet 之後，再開啟 storyboard 檔，並點擊 Table Cell 後，按下 Alt + Command + 6 (Connections Inspector)，將 demo1 和 demo2 兩個 outlet 連接上。接著再按下 Alt + Command + 4 (Attribute Inspector)，給予這個 Cell 的 Identifier 一個名稱，在此示範為它命名為「Demo Cell」，如下圖(圖 12)所示：

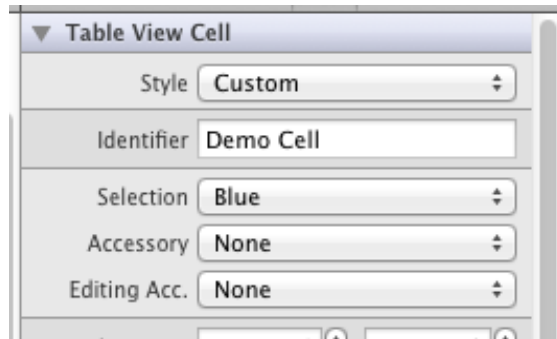


圖 12，Table Cell 的 Identifier

這時，我們的自行定義 Table View 就快完成了，接下來只要在目前這個 View 所屬 .m 檔中尋找下面這個方法並略微修改內容，要修改的部份是 CellIdentifier 的賦值以及 cell 的型態這兩個部份：

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Demo Cell";
    subCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[subCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
    }
    .....
}
```

下面是自定表格的完成圖，表格中 Label 的擺放位置是能自行放置的，而顯示的文字則是該 Cell 是所屬第幾個 row。

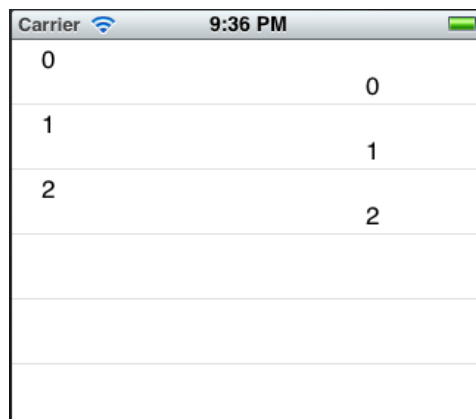


圖 13，完成圖

3.2 cookie 取得與刪除

iOS 中要如何使用 cookie 來進行 http 的請求？iOS 十分強大，不管你願不願意，每當你透過 `NSURLRequest` 訪問某個網頁時，它都會主動幫你記錄下這個網頁的 cookie，而且每當你再次拜訪這個網頁時，`NSURLRequest` 就會拿到先前所記錄下來的 cookie 去進行相關的請求。

那要怎麼查看 cookie 呢？我們可以透過以下方式：

```
NSHTTPCookieStorage *cookieJar = [NSHTTPCookieStorage sharedHTTPCookieStorage];
for (NSHTTPCookie *cookie in [cookieJar cookies]) {
    NSLog(@"cookie is %@", cookie);
}
```

此時你應該會得到類似下面這串東西：

```
cookie is <NSHTTPCookie version:0 name:"JSESSIONID"
value:"C2E750CDDAE50DCD329935967F998D0F" expiresDate:(null) created:2012-10-11 11:53:04
+0000 (3.71649e+08) sessionOnly:TRUE domain:"sais10.wfu.edu.tw" path:"/" isSecure:FALSE>
這個就是cookie了。
```

要怎麼清除cookie呢？使用以下方式：

```
NSHTTPCookieStorage *cookieJar = [NSHTTPCookieStorage sharedHTTPCookieStorage];
NSArray *tempArray = [NSArray arrayWithArray:[cookieJar cookies]];
for (id obj in tempArray) {
    [cookieJar deleteCookie:obj];
}
NSLog(@"cookie is %@", cookieJar);
```

這裡再次利用 `NSLog` 來查看，可以發現剛才還有一大串的 cookie，現在已經被清空了，如下所示：

```
cookie is <NSHTTPCookieStorage count:0 cookies:[]>
```

利用取得和清除 cookie 的方式，在登入登出的處理上就輕鬆多了。

第四章 預期成果

4.1 登入畫面

登入操作在系統開發上可算是最常見的介面了，它負責把關用戶是否有權限可使用該系統，唯有系統的會員才有資格可以進行操作，圖 14、圖 15 分別是 iPhone 與 iPad 兩種平台的登入畫面。



圖 14，這是登入的畫面，也是應用程式跑起來最初的畫面。

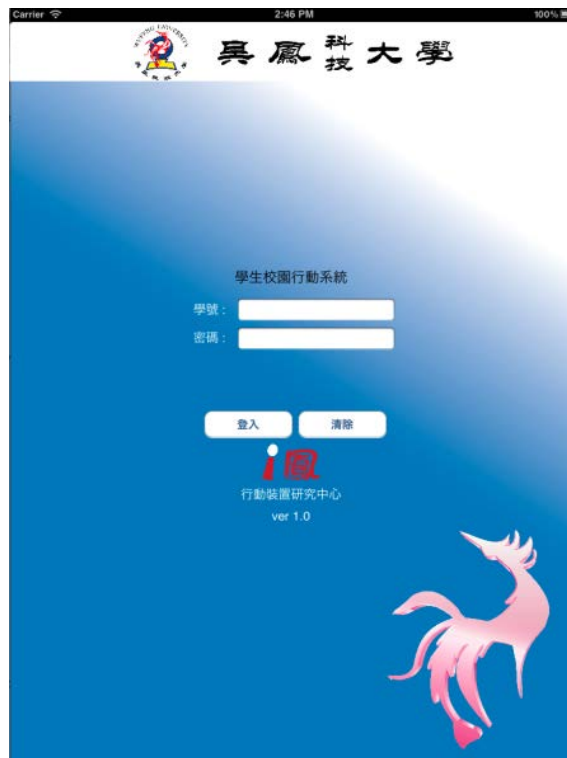


圖 15，iPad 的登入畫面

4.2 選單列

在本研究所開發的系統中，擷取了學生們較常使用的相關子系統，由於行動裝置的顯示畫面有限，無法如個人電腦上那麼多彩多姿，因此採用較平實的風格，運用 Table View 將這些子系統列出，供用戶點選。



圖 16，這是進去之後的選單列



圖 17，iPad 的選單列

4.3 選課系統

就學生的角度來看，選課是每個學期都會接觸到的功能，選課系統除了加退選之外，也可以在此查詢自己本學期所選的課程表。



圖 18，點了「選課系統」後，會出現下面的畫面，因為 iPhone 的畫面小，所以較長的文字無法全部顯示，加入了點擊表格後會進入詳細課程的效果。目前只有「查」的動作，所以「新增」和「退課」還沒有加入動作。



圖 19，iPad 畫面就顯得比較大了，因此不用再額外加入詳細課程的畫面。

4.3 校園公告

透過校園公告的系統，學生即使處在沒有電腦的場所，也可以藉由行動裝置取得最新的校園消息。



圖 20，再來是點了「校園公告」的畫面，會分成「校園新聞」、「校園活動」和「學術研討」三類。



圖 21，可以從新聞列表中選想要的那則新聞，iPhone 畫面較小，所以要使用 iPhone 特有的多點觸控功能，利用手指進行縮放，換到 iPad 上就沒這問題了。



圖 22，上圖就是 iPad 的畫面了，可以明顯看出來無論是標題還是內容都可以比較完整的顯示出來。



圖 23，新聞內容也有較大的顯示畫面

4.4 開發中子系統

除了以上系統之外，還有部份子系統尚在開發階段，例如下圖(圖 24)則是請假系統的示意圖，後續將會再繼續開發，請拭目以待。



圖 24，其他子系統正在開發中，以上是請假系統的展示圖

第五章 結論

本研究主要是針對在 Apple iOS 作業系統上，不借助任何第三方軟體，完全以原生程式碼的方式進行系統開發。該系統目前還在開發階段，其中雖以完成幾項子系統，但還未及正式上線地步。由於開發過程牽涉到個人資訊安全的部份，因此在與後端資料庫溝通上是請吳鳳科技大學的電算中心協助，筆者則是針對在前端系統開發上的過程。若從專案管理的角度來看，在開發過程中，時常發生兩邊進度不一致，導致開發時程拉長。

本文當中除了探討 iOS 相關架構外，也穿插了些許與開發 iOS 應用所要注意的事項及技巧，在常見的工具書中所描述的是相關的基礎用法，進階技巧與思路則很難詳細說盡，而本研究這些內容是筆者在整個開發過程中所學習到的經驗與技術，透過整理後再將它們以文字的形式分享出來。希望能對未來也是走上開發之路的讀者有所幫助。

參考文獻

1. 李亮,楊武,張永強,苟振興譯, Bill Dudney, Chris Adamson 著, iPhone SDK 3 開發指南, 電子工業出版社
2. 邵子卿,洪沛然譯, Jack Nutting, David Mark, Jeff LaMarche 著, 探索 iPhone 4 程式開發實戰, 基峰
3. 國家通訊傳播委員會 (NCC)
4. 網路資料: 淺談 iOS 5 的 StoryBoard, <http://cocoapps.blogspot.tw/2011/08/ios-5storyboard.html>
5. 網路資料: 維基百科「iOS」, <http://zh.wikipedia.org/wiki/IOS>
6. 網路資料: <http://www.inside.com.tw/2012/08/28/ios-and-android-adoption-explodes-internationally>
7. Flurry, About Flurry <http://www.flurry.com/index.html>
8. Gartner, About Gartner <http://www.gartner.com/technology/about.jsp>
9. Stanford University, CS193P iPhone Application Development, Fall (2011-12)