

逢 甲 大 學
資 訊 工 程 學 系
專 題 發 表 報 告

iPhone App：旅遊日記

指導教授：林明言

班 級：資訊四甲/資訊四乙

學 生：D9772585 吳佩玟

D9736085 陳鵬宇

D9772585 楊震麟

摘要

誰說寫一本旅遊日記一定要帶筆記本、相機、地圖這些東西，現在，你只要帶著你的 iPhone，打開「旅遊日記」就能隨意創造出專屬於自己的旅遊日記！旅遊日記包含了四項子功能，記事、相機、旅遊日曆、地圖。讓你凡走過必留下痕跡，只要手機隨手一拍，輕點螢幕把事情記錄下來，標記在地圖上，它們全部都會標記在旅遊日曆上，這些就是專屬於你的「旅遊日記」

目錄

摘要.....	2
目錄.....	3
圖目錄.....	7
表目錄.....	9
第一章 導論	10
1.1 專題概述	10
1.2 專題動機	10
1.3 專題目標	11
第二章 平台介紹	12
2.1 開發環境介紹	12
2.1.1 硬體需求	12
2.1.2 軟體需求	12
2.1.3 關於 Mac OSX	12
2.1.4 關於 iOS	12
2.2 開發工具與介紹	16
2.2.1 工具	16
2.2.2 SDK 介紹	19
2.2.3 iPhone 程序介紹	21

2.2.4	SQLite3 介紹	23
2.3	系統特色	23
2.3.1	旅遊日記特色	23
2.3.2	使用物件介紹	23
2.3.3	名詞定義	25
第三章 系統分析與設計		26
3.1	系統分析方法	26
3.1.1	瀑布分析法	26
3.1.2	雛形分析法	26
3.1.3	螺旋分析法	26
3.2	可行性分析	27
3.2.1	操作可行性	27
3.2.2	技術可行性	27
3.2.3	行程可行性	27
3.3	系統功能設計	28
3.3.1	記事子系統	28
3.3.2	相機子系統	28
3.3.3	旅遊日曆子系統	29
3.3.4	地圖子系統	29

第四章 系統架構	31
4.1 系統規格	31
4.2 系統架構	32
4.3 子系統規劃	33
4.4 資料庫規劃	34
4.4.1 relational schema	34
4.4.2 ER model	35
第五章 人機介面設計	36
5.1 HTA 圖	36
5.2 STD 圖	37
5.2.1 主畫面 STD	37
5.2.2 記事 STD	38
5.2.3 相機 STD	39
5.2.4 旅遊日曆 STD	40
5.2.5 地圖 STD	41
第六章 系統開發與實作	42
6.1 時程規劃	42
6.2 系統實作成品	43
6.2.1 記事	43

6.2.2	相機	46
6.2.3	旅遊日曆	51
6.2.4	地圖	53
<u>參考文獻</u>		<u>57</u>

圖目錄

圖 2.1 OS X Lion	12
圖 2.2 iOS 5 新功能	13
圖2.3 通知中心	13
圖2.4 iMessage	14
圖2.5 書報攤	14
圖2.6 提醒事項	15
圖2.7 相機	15
圖2.8 照片	16
圖2.9 Interface Builder	16
圖 2.10 New Project	17
圖 2.11 ButtonExample	17
圖2.12 Groups&Files	18
圖 2.13 .xib View	18
圖2.14 Cocoa Touch	19
圖2.15 Media	19
圖2.16 Core Services	20
圖2.17 Core OS	20
圖2.18 iPhone程序圖	21
圖 2.19 MVC	25
圖 3.1 瀑布分析圖	27
圖3.2 記事子系統 - a	29
圖3.3 記事子系統 - b	29
圖3.4 相機子系統	30
圖3.5 旅遊日曆子系統	30
圖3.6 地圖子系統 - a	31
圖3.7 地圖子系統 - b	31
圖3.8 地圖子系統 - c	31
圖 4.1 系統架構圖	32
圖 4.2 relational schema	34
圖 4.3 ER model	35
圖 5.1 HTA 圖	36
圖 5.2 STD - 主畫面	37
圖 5.3 STD - 記事	38
圖 5.4 STD - 相機	39
圖 5.5 STD - 旅遊日記	40

圖 5.6 STD - 地圖	41
圖 6.1 時程表	42
圖 6.2 記事 - a	43
圖 6.3 記事 - b	43
圖 6.4 記事 - c	44
圖 6.5 記事 - d	44
圖 6.6 記事 - e	45
圖 6.7 記事 - f	45
圖 6.8 記事 - g	46
圖 6.9 相機 - a	46
圖 6.10 相機 - b	47
圖 6.11 相機 - c	47
圖 6.12 相機 - d	48
圖 6.13 相機 - e	48
圖 6.14 相機 - f	49
圖 6.15 相機 - g	49
圖 6.16 相機 - h	50
圖 6.17 相機 - i	50
圖 6.18 旅遊日曆 - a	51
圖 6.19 旅遊日曆 - b	51
圖 6.20 旅遊日曆 - c	52
圖 6.21 旅遊日曆 - d	52
圖 6.22 地圖 - a	53
圖 6.23 地圖 - b	53
圖 6.24 地圖 - c	54
圖 6.25 地圖 - d	54
圖 6.26 地圖 - e	55
圖 6.27 地圖 - f	55
圖 6.28 地圖 - g	56
圖 6.29 地圖 - h	56

表目錄

表 4.1 記事屬性表	33
表 4.2 相簿資料夾屬性表	33
表 4.3 照片屬性表	33
表 4.4 大頭針文字屬性表	33
表 4.5 大頭針照片屬性表	33
表 4.6 地圖屬性表	33

第一章 導論

「旅遊日記」用手機輕鬆記錄生活，在每個特別的日子加上註記，用照片和簡短的文字讓生活深刻記錄，不會隨著時間容易淡忘，導航讓你更簡單地回到曾去過的地方。「凡走過必留下痕跡」用簡單的程式，幫你留下每個美麗的痕跡。

1.1 專題概述

當你今天和戀人旅行到「東京鐵塔」想的是如何記錄下這美麗的一刻，於是你會拿起單眼捕捉美麗的畫面，回到飯店看著照片，回憶今天所發生的所有事，在網站上發布日記，或者在身邊的行事本上紀錄。結束了一天的行程後，在睡前確認明天的行程後再行入睡。然後發現在行事曆上自己註記著下禮拜要到「北海道」出差，又在腦中苦苦回憶路線該怎麼前往，以往去過的記憶似乎有些模糊，於是搜尋地圖才能知道路線，記錄好路線規劃後還要期待自己不要迷路才好。

於是，如果單眼內建記事的功能，這樣就能在第一時間紀錄最精彩的每個片段；如果每個行程過後都能加上照片，用照片來記錄生活；如果地圖能夠聰明的記下曾去過的地方，讓你不再迷路，這樣該有多好。

我們將設計一個擁有四個子功能的應用程式，當你想到某個地點時，可以使用我們的導航搜尋功能，幫助你找到目的地，到達目的地後，可以開啟照片功能替這個地點標示上獨一無二的紀錄。同時也擁有記事功能，讓你方便記錄發生的事件，以及行事曆功能，幫你標註每天的行程，以及未來即將要做什麼，還可以將照片加在行事曆上，讓你更清楚在哪些天去過哪些地方做過哪些事。希望藉此四個功能，能讓生活更精采動人。

1.2 專題動機

為什麼單眼只會是單眼，不會告訴你去年拍的那張照片在哪拍得、怎麼去、發生過什麼？為什麼地圖明明說XX路要左轉，可是哪來的左轉？最後又是迷路問路又迷路…

於是我們想如果這些問題可以解決的話那該有多好，懂得理財規劃的人，喜歡用記帳本來清楚記錄每一天的花費與入帳，但是帳本卻不是隨時隨地輕鬆拿出來記錄，往往都是回到家再慢慢一筆一筆回想，或者是對照發票及收據來記錄等等。喜歡拍照攝影的人，喜歡每到一個新地點，就拿起相機，捕捉每一個動人的時刻。還有喜歡替拍完的照片加上一點屬於自己的特殊效果，但相機無法當下就修改的功能，通常還是必須回到電腦上在修改。旅遊者，他們享受生活，會把行程記錄在行事曆上，也喜歡使用地圖來幫助他們到每個未知的地方旅行。如果可以把這些特性都統合起來，成為一個應用程式，讓你可以更輕鬆的使用你想要使用的東西，用一個更簡易的方式來取代以往繁雜的動作，這樣就能夠達到一項東西擁有多項功能的目的。

1.3 專題目標

當手機應用程式，可以使生活的習慣更完美，你的生活就會更接近未來的雲端生活，一支手機可以發揮多大的效用，讓平常的隨身物品更輕便，隨走隨拍隨記，只要一個程式就能包辦所有功能。

打開[旅遊日記]的行事曆幫助你一目了然接下來的行程，如果那是一個地點還有照片幫助記憶，以及十二星座隨時提醒你該月份是否有哪些朋友是壽星。當你在外突發奇想的想到一個地方遊玩，旅遊日記擁有地圖當你的嚮導，地圖上的標記讓你清楚的知道曾去過的每個地方，同時可以將照片標記上去，讓你專屬的地圖更完美。有了這樣的地圖就能輕鬆回憶起曾經去過的地點，拍過什麼照片做過什麼事，這些就在這四個完美結合在一起的程式裡，那是我們希望帶給大家的[旅遊日記]。

第二章 平台介紹

2.1 開發環境介紹

2.1.1 軟體需求

作業系統：MAC OS X

↑原因：因為iPhone程式開發的環境有特殊限制，必須使用MAC他們的OS系統才能進行開發，所以使用這個系統。

資料庫：SQLite

↑原因：可跨平台開發，其特色是在程式語言內的直接API呼叫。這在消耗總量、延遲時間和整體簡單性上有積極的作用。

繪圖軟體：Photoimpact 11、Microsoft 小畫家

程式語言：Objective-C

↑原因：同作業系統的限制，其環境的開發語言限制必須使用Objective-C此語言，來進行開發。

2.1.2 硬體需求

CPU：AMD Athlon 166MHz 以上

記憶體：1GB 以上

↑原因：電腦主機的速度為一般的規格即可，但建議記憶體為1GB甚至更高，好讓電腦在運作系統和存取資料時會較為順暢。

硬碟：40GB 以上

I\O 設備：滑鼠、鍵盤、螢幕

2.1.3 關於Mac OSX

OS X LION — Macintosh最新的系統版本



圖2.1 OS X Lion

多年來，大家操作電腦的方式一成不變。點按、捲視、安裝、儲存。在OS X Lion中，Macintosh導入將改變你電腦使用習慣的全新功能，挑戰人們約定俗成的操作方式。

2.1.4 關於iOS

iOS 5 — Macintosh最新的iOS版本

iOS 5內含超過200種新功能，提供你的iPad、iPhone或iPod touch使用。以下列出幾項首選功能：



圖2.2 iOS 5 新功能



圖2.3 通知中心

- 不再打擾你。
- 輕掃即可讀取。
- 更加一目了然。





圖2.4 iMessage

- 發送訊息，不再只有文字。
- 那邊回覆時，這邊看得見。
- 這台裝置起個頭，換台裝置收個尾。



書報攤



圖2.5 書報攤

- 新聞新知總匯。
- 掌握最新時事。
- 訂閱更多。



圖2.6 提醒事項

- 看看接著要忙些什麼。
- 擬份清單。



圖2.7 相機

- 快門鍵。
- 從鎖定時螢幕進入。



照片



圖2.8 照片

- 裁切與旋轉。
- 螢幕上編輯。

2.2 開發工具與介紹

2.2.1 工具

方法：

- 以MVC的方法來進行開發
- Model - 紀錄程式所有的方法
- View - 設計給使用者的畫面
- Controller - 把model內的方法實作並且回饋給view使view完成Controller所給與的任務

技巧：

以物件導向的觀念為主軸來進行開發



圖 2.9 Interface Builder

點選Xcode

New 一個 Project:

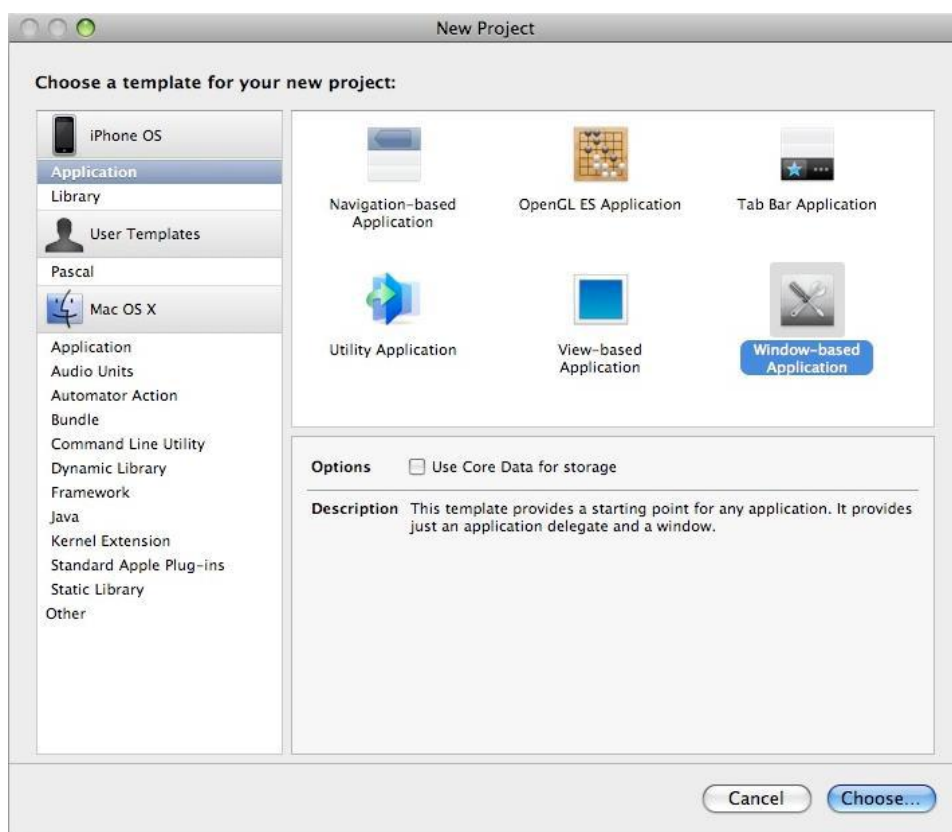


圖 2.10 New Project

產生新的 Project 如下

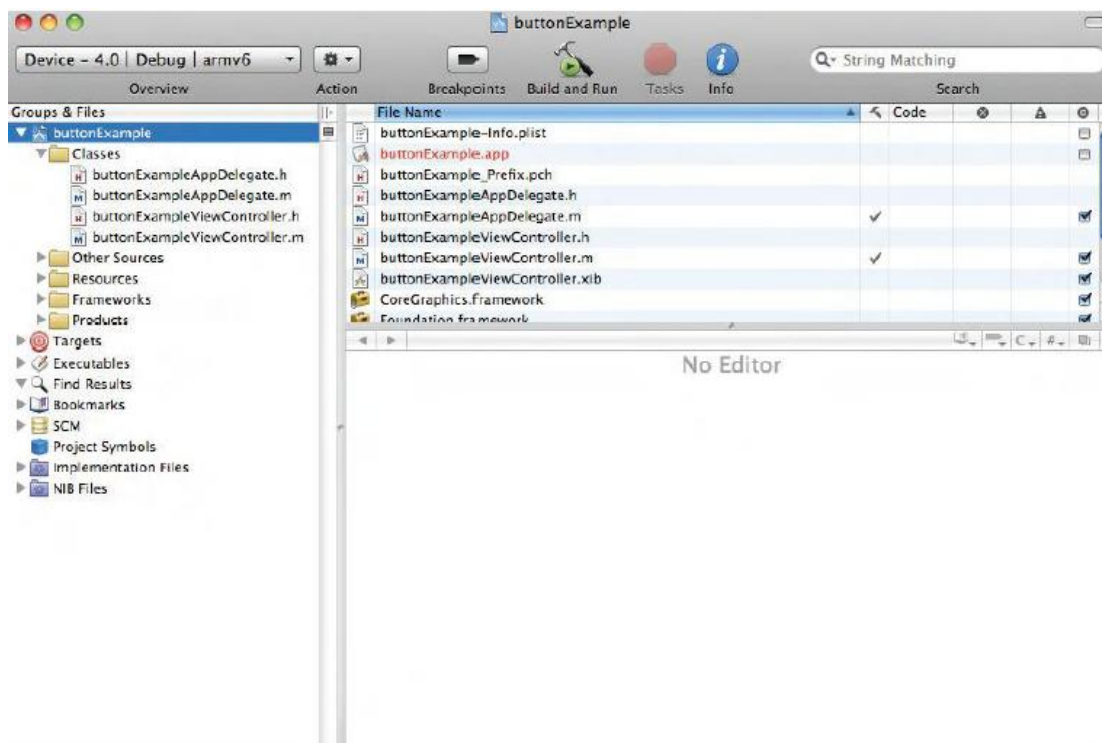


圖 2.11 ButtonExample

找到 MainWindow.xib 開啓，會交由 Interface Builder (IB)



圖 2.12 Groups&Files

點開 .xib 後 會出現以下視窗

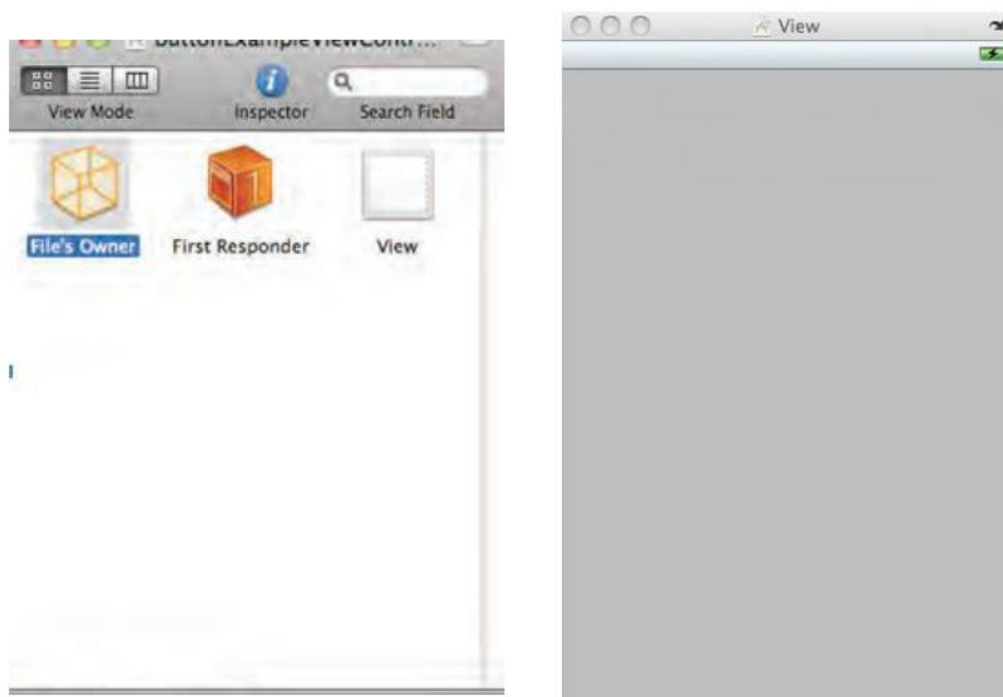


圖 2.13 .xib View

2.2.2 SDK 介紹

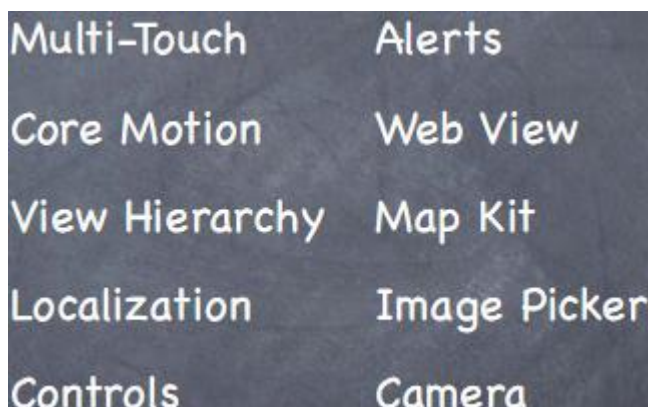
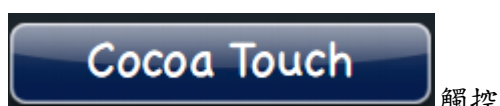


圖 2.14 Cocoa Touch

- 多點觸控事件和控制(Multi-touch events and controls)
- 加速支援(Accelerometer support)
- 檢視等級(View hierarchy)
- 在地化 (i18n) (Localization (i18n))
- 相機支援

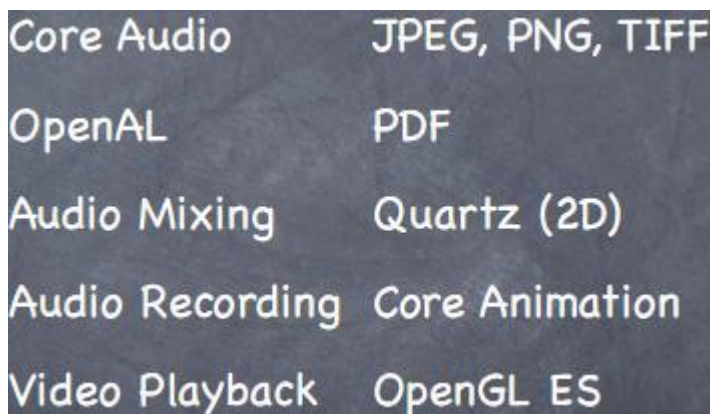


圖 2.15 Media

- OpenAL
- 混音及錄音(Audio mixing and recording)
- 視訊播放
- 圖像檔案格式(Image file formats)
- Quartz

- Core Animation
- OpenGL ES

Core Services 核心服務

Collections	Core Location
Address Book	Net Services
Networking	Threading
File Access	Preferences
SQLite	URL Utilities

圖 2.16 Core Services

- 網路
- SQLite 嵌入式資料庫
- 地理位置(GeoLocation)
- 線程(Threads)

Core OS OS X 核心

OSX Kernel	Power Management
Mach 3.0	Keychain Access
BSD	Certificates
Sockets	File System
Security	Bonjour

圖 2.17 Core OS

- TCP/IP 協議
- 套接字(Sockets)
- 電源管理
- 檔案系統(File system)
- 安全

2.2.3 iPhone 程序介紹

此節是介紹我們執行App時的一個生命週期，讓你對iPhone的運作方式能夠更加瞭解。首先我們先點選iPhone上的icon圖示，由程式的觀點來看，點選後程式會先執行，main函數。之後利用main函數去呼叫UIApplication函數，這個函數放在我們一開始創造專案時叫做MainAppDelegate裡面，它管理著手機的進入點，縮小在背景，或是如果離開程式在底層時的執行狀況。如下圖2.18所示：

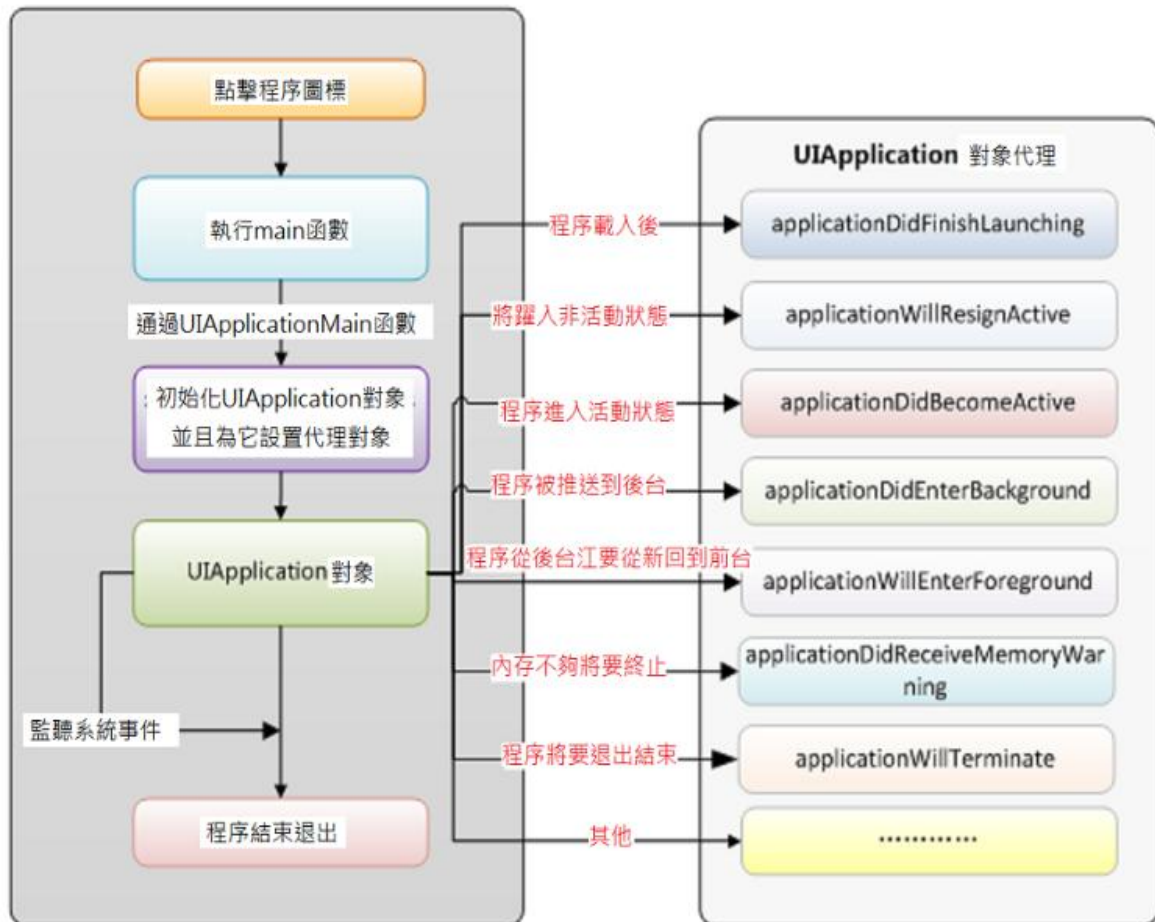


圖 2.18 iPhone 程序圖

iPhone 程序開發過程如上圖 2.18 所示，詳細函式說明如下列所示：

1、- (void)applicationWillResignActive:(UIApplication *)application

說明：當應用程序將要入非活動狀態執行，在此期間，應用程序不接收消息或事件，比如電話了

2、- (void)applicationDidBecomeActive:(UIApplication *)application

說明：當應用程序入活動狀態執行，這個剛好跟上面那個方法相反

3、- (void)applicationDidEnterBackground:(UIApplication *)application

說明：當程序被推送到後台的時候調用。所以要設置後台繼續運行，則在這個函數里面設置即可

4、- (void)applicationWillEnterForeground:(UIApplication *)application

說明：當程序從後台將要重新回到前台時候調用，這個剛好跟上面的那個方法相反。

5、- (void)applicationWillTerminate:(UIApplication *)application

說明：當程序將要退出是被調用，通常是用來保存數據和一些退出前的清理工作。這個需要設置 UIApplicationExitsOnSuspend 的鍵值。

6、- (void)applicationDidReceiveMemoryWarning:(UIApplication *)application

說明：iPhone 設備只有有限的內存，如果為應用程序分配了太多內存操作系統會終止應用程序的運行，在終止前會執行這個方法，通常可以在這裡進行內存清理工作防止程序被終止

7、- (void)applicationSignificantTimeChange:(UIApplication *)application

說明：當系統時間發生改變實執行

8、- (void)applicationDidFinishLaunching:(UIApplication *)application

說明：當程序載入後執行

9、- (void)application:(UIApplication *)application
willChangeStatusBarFrame:(CGRect)newStatusBarFrame

說明：當 StatusBar 框將要變化時執行

10、- (void)application:(UIApplication *)application
willChangeStatusBarOrientation:
(UIInterfaceOrientation)newStatusBarOrientation
duration:(NSTimeInterval)duration

說明：當 StatusBar 框方向將要變化時執行

11、- (BOOL)application:(UIApplication *)application
handleOpenURL:(NSURL *)url

說明：當通過 url 執行

12、- (void)application:(UIApplication *)application
didChangeStatusBarOrientation:(UIInterfaceOrientation)oldStatusBarOrientation

說明：當 StatusBar 框方向變化完成後執行

13、- (void)application:(UIApplication *)application
didChangeSetStatusBarFrame:(CGRect)oldStatusBarFrame

說明：當 StatusBar 框變化完成後執行

2.2.4 SQLite3 介紹

SQLite 是遵守 ACID 的關連式資料庫管理系統，它包含在一個相對小的 C 庫中。不像常見的用戶端/伺服器結構範例，SQLite 引擎不是個程式與之通訊的獨立行程，而是連線到程式中成為它的一個主要部分。所以主要的通訊協議是在程式語言內的直接 API 呼叫。這在消耗總量、延遲時間和整體簡

單性上有積極的作用。整個資料庫（定義、表、索引和資料本身）都在宿主主機上儲存在一個單一的檔案中。它的簡單的設計是透過在開始一個事務的時候鎖定整個資料檔案而完成的。

SQLite 優點：

1. 體積小（這裡指的是安裝所需的空間，實際的資料庫大小就看資料的多寡）、備份方便，如果只是要在系統上裝上 SQLite 的支援（libsqlite），只要幾百 KB 的大小，加上它是檔案型的資料庫，備份其實就是把整個檔案備份起來就好了。
2. 優異的效能。有 SQLite 的測試數據，在「需要經常讀取，但是寫入次數不高」的情況下，效能甚至超越了以速度聞名的 MySQL。
3. 眾多語言支援。常見的語言都支援 SQLite。
4. 跨平台。目前主流的作業系統都支援。Mac 甚至在作業系統 Mac OS X - Leopard 中讓 SQLite 成為預設的資料庫。

2.3 系統特色

2.3.1 旅遊日記特色

本軟件能幫你以最簡單的方式記錄生活中的一切細節。

它可以作為備忘錄。

它可以幫照片分資料夾。

它可以作為具有提醒功能的計劃表。

它可以在旅途中記錄你到過的每個地方。

它可以在第一時間記錄你的創作靈感。

它還可以讓你與朋友分享趣事，當然你也可以選擇保留隱私。

主要功能：

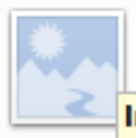
1. 記錄文字
2. 記錄圖片
3. 記錄每個事件的發生時間（或者設定計劃時間）
4. 記錄每個事件的發生地點
5. 按日期、類型、地點或者完成與未完成等分類瀏覽照片
6. 在地圖上用大頭針標示地址、照片、資訊

2.3.2 使用物件介紹

TableView: 實作表格實用的物件



imageView: 實作有關圖片時需要用的物件



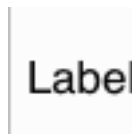
MapView:實作地圖時需要用到的物件



TextView:類似 TextField 可以輸入多行文字



Label:可變動大小的動態文字欄



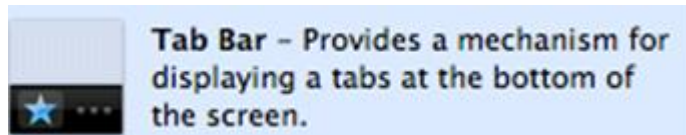
Button:可以觸發事件或是傳送訊息給目標物件



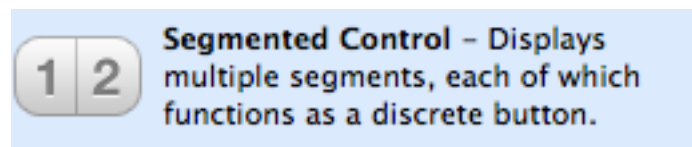
TextField:可編輯文字 可以傳送訊息給目標物件



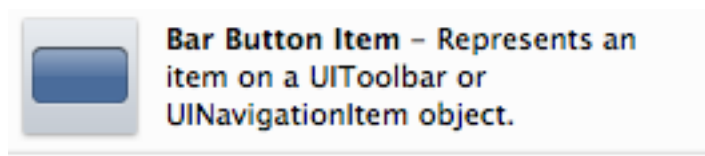
Tabbar:可以在一個畫面上實作多個分頁



Segmented Control:類似 TabBar 但是適用在同個畫面內 功能類似 Switch



Bar button Item:



2.3.3 名詞定義

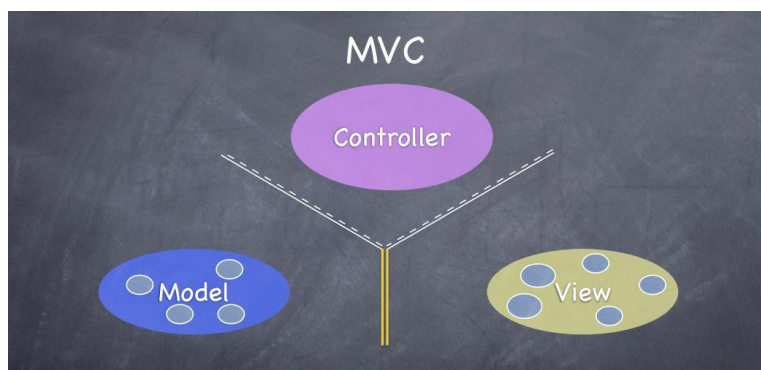


圖 2.19 MVC

系統軟體架構模式如上圖 2.19，詳細說明如下列所述：

Model = What your application is (but not how it is displayed)

Model 負責管理程式的資料以及狀態，同樣的 Model 在程式中的不同介面應該要能夠擁有一致的操作，以便重複利用。

Controller = How your Model is presented to the user (UI logic)

Controller 負責處理 Model 跟 View 中間的程式邏輯，使用者面對 View 進行資料的操作，而當 View 接受到這些指令，他就會把這些訊息傳遞給 Controller，Controller 在根據不同的 Model 進行操作。

View = Your Controller's minions

View 負責把 Model 呈現到使用者介面，作為使用者操作資料的介面。它本身不處理資料相關的程式邏輯，所以如果即便套用上不同的 Model，理論上同一份 View 都能夠搭配顯示。

第三章 系統分析與設計

系統分析其最主要的目的是系統撰寫者了解使用者對系統的需求以及操作上的要求，進而以文件描述需求的詳細內容，並且清楚仔細地定義出每個子系統所需要的功能。在開發過程中系統分析是很重要的一環，也是一個系統重要的開始步驟，要想設計出完善的系統，就必須要有個對系統撰寫者最合適的方法，如果缺乏有效率的系統分析方法，這個系統很有可能就無法達到預期的理想狀態。在此將討論下列三種系統分析的方法：

3.1 系統分析方法

- (1) 瀑布分析法
- (2) 雛形分析法
- (3) 螺旋分析法

3.1.1 瀑布分析法

瀑布分析法是由上至下的流程，當其中一個階層的工作完成時，才能進行下一個階層的工作，而如果發現前一階段的工作出現錯誤不夠完整時，則往回至前個階層重新工作，通常會把系統測試和除錯放在最後的階段，對於非常大型的系統而言，容易會在實作過程中發生錯誤或出現瑕疵而沒在當下發現並糾正，增加了開發的成本。主要工作如下：

- 1. 詳細了解使用者所碰到的問題以及發生問題的原因。
- 2. 定義系統目的和工作內容。
- 3. 確認系統的功能和要求。
- 4. 確認系統的潛在風險和限制要求。
- 5. 決定系統的範圍。
- 6. 找出所有的可行性方案，對其進行研究，找出最佳的方案，並分析此最佳方案所帶來的成本效益。
- 7. 撰寫可行性報告及專案計劃。

3.1.2 雛形分析法

由系統撰寫者和使用者一起確認系統的目的和需求以及所有細節的功能，對使用者的使用介面以及最適合使用者的操作流程很要求，系統撰寫者得以快速設計的方式擬定出一套系統的雛形，讓使用者可以測試操作進而發現不足的功能或者是功能上哪裡還需要調整，一直重覆的進行快速設計跟使用者測試的步驟，直至使用者覺得滿意為止。雛形分析法的優點如下：

- 1. 使用者加入定義系統功能的過程，這樣可以在發現不當之處時隨時提醒，有種為使用者量身訂做的感覺。
- 2. 此分析方法的成本較瀑布分析法低，所以開發所需的時間自然也不會那麼長。
- 3. 此分析法讓使用者得以驗證其邏輯，技術系統規格書可藉由驗證工作內容，建立未來改良版本的藍圖。

4. 通常此分析方法會搭配較省時省力的開發工具來實作。

3.1.3螺旋分析法

螺旋分析法結合了瀑布分析法以及雛形分析法的性質，由系統中的核心功能階層建構雛形，與一般雛型最大的不同點在於，它是一個真正功能完善的系統，而不是像半成品功能不完整的系統。它會藉由不同的輸入格式，畫面顯示以及輸出的報告，進而引出使用者的需求。主要工作如下：

1. 資料蒐集過後，建立專案計劃。
2. 對計劃階段中所有解決方式執行風險分析以及各個風險項目的認定。
3. 藉由使用者測誦之後，進行下一版的開發。
4. 使用者對其呈現的結果進行測誦且加以評估。

開發的方法沒有絕對的好或壞，每種方法都有其優點、缺點以及其適用的範疇，通常依照個人的使用方向來決定。在經過審慎思考後瀑布分析法擁有符合我們小組情形的幾個特點。

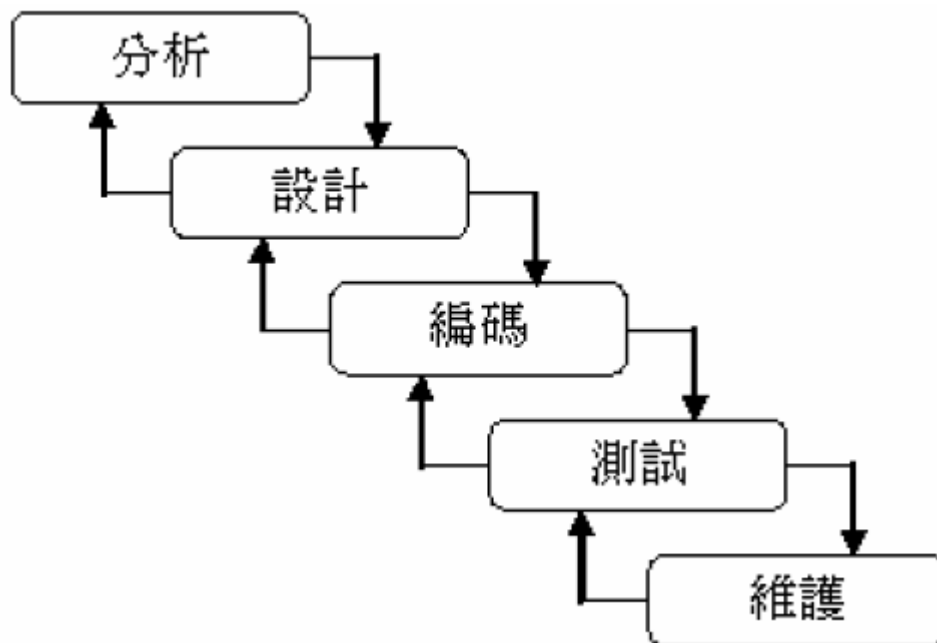


圖3.1 瀑布分析圖

瀑布分析如上圖3.1所示，詳細說明如下：

1. 分割數個小區塊來進行實作：在專題製作剛開始的時候往往只擁有基本的想法及目標，而並不是完整的開發計劃，所以採納這種循序漸進的方法可以在開發同時思考接下來的行動。
2. 前個階層完成才能繼續下個階層：因為這是我們第一次製作大型的專題，所以我們無法處理如此大型系統發生錯誤的風險，因此分成多階層的實作，在任何階段下如果發現錯誤，就可以立即回至前階段進行修改，有效地降低系統的風險。

3. 階段的清楚劃分：不同階段進行不同的功能，進而可以在分工上很容易的分配工作，如有必要進行修正時也很容易地知道問題發生的階段為何，因此在系統管理上相當方便。

3.2 可行性分析

本章節將針對操作之可行性，技術開發的可行性，以及最為重要的時程規劃之可行性，以下將以此小節方面一一詳述。

一個系統之可行性，決定了此系統開發過程的成功與否，旅遊日記系統的可行性分析也是相同，為了避免開發過程中，出現初步可避免或是無法修正的開發問題，因此擬定其可行性分析之研究。

3.2.1 操作可行性

到目前為止，還較少有系統整合不同的功能，像是相片、地圖、記事、行事曆等…都是以單一功能的出現，但是這些又是出門玩時不可或缺的功能，當我們要使用時又要打開不同的 app 來使用這些功能便略顯麻煩，為了改善這種問題，於是我們便將各種功能整合為一個 app 讓使用者可以在出門玩時只需打開一個 app，不用在因為不同的功能去使用不同的 app。

3.2.2 技術可行性

本專題採用平台為 Apple iPhone，使用技術以 apple 開發之 cocca touch framework 以及 O B J E C T I V E - C 為主，而 iPhone 具有方便、好操作、可以隨身攜帶，以及由 Apple 把關產品品質，因此可以放心設備的品質不會有太大的錯誤發生。

Objective-C

簡言之，是為 C 語言的另一種語言，因此以過往課程所學之 C 語言基礎，加上 objective C 之語法特性，熟悉此語言並不是非常的難。

Cocca touch Framework

由於 apple 開發 ios 之 framework A P I 之過程便將部份 M V C 架構 API 內建其中，因此在開發過程，只需理解 A P I 之生命週期與開發模式便可。

3.2.3 行程可行性

考量時程分析的可行性，會與其系統之範圍有關，若無妥善規劃開發時程，會發生開發範圍擴大之危險，而作為一個專題的系統考量，旅遊日記實作範圍定於前端 iPhone 界面操作。因此，時程考量便可以減到最少；此外，透過訂定實作範圍也可以更加確定系統的實作需求，以及實作的細緻程度。

3.3 系統功能設計

3.3.1 記事子系統

記事資訊子系統，是旅遊日記一個重要的功能之一，此系統幫助我們記住甚麼時候需要去辦什麼事情。當我們輸入標題跟時間時，去更新資料庫，並將資料顯示於 view 上，讓你一目了然。詳細資料庫設計如下圖 3.1 圖 3.2

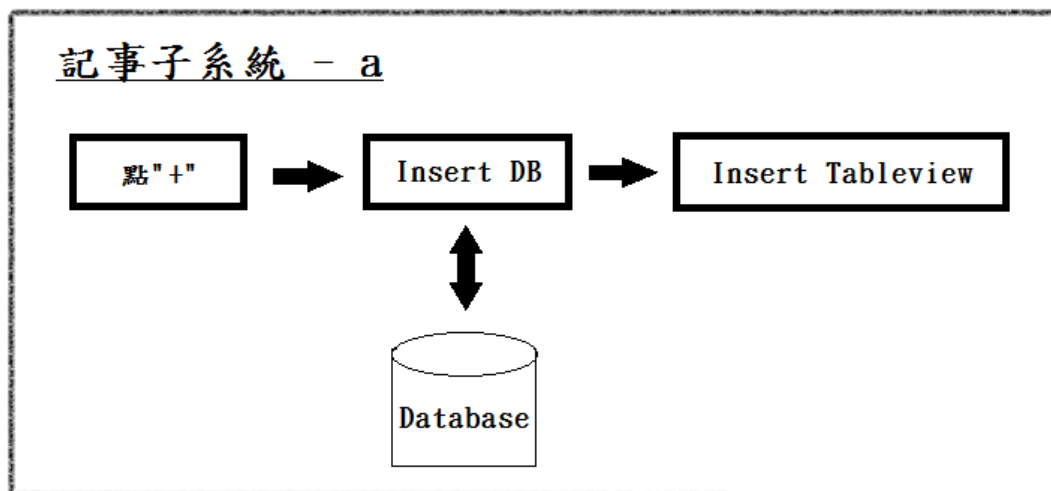


圖 3.2 記事子系統 - a

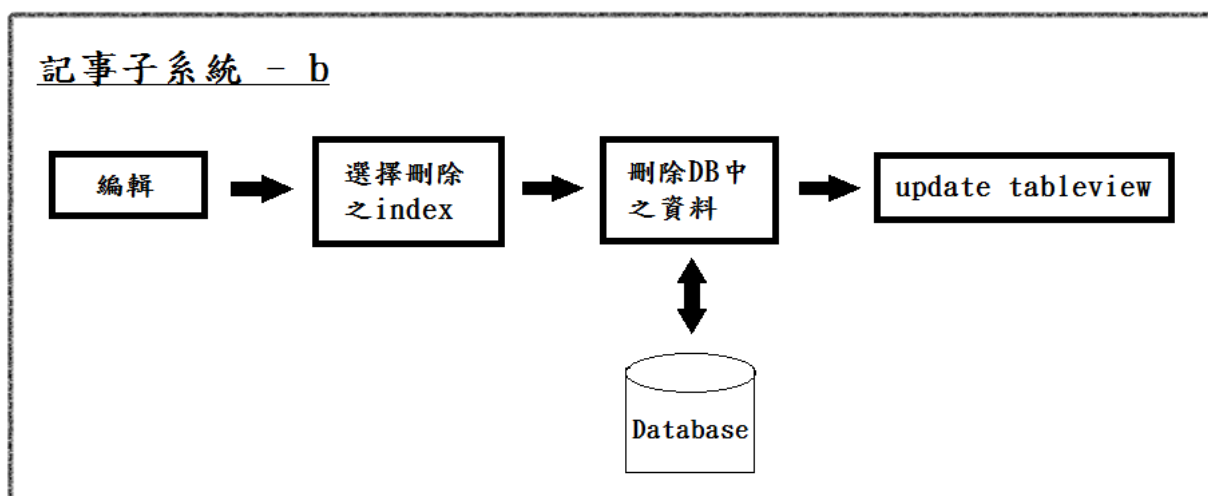


圖 3.3 記事子系統 - b

3.3.2 相機子系統

相機子系統這個功能，是為了讓使用者不管走到哪，都能夠隨時拍下自己出門玩的相片，以及瀏覽自己出門玩拍下的照片所設置的功能，讓你達到隨走隨拍的方便性。詳細資料庫設計如下圖 3.3

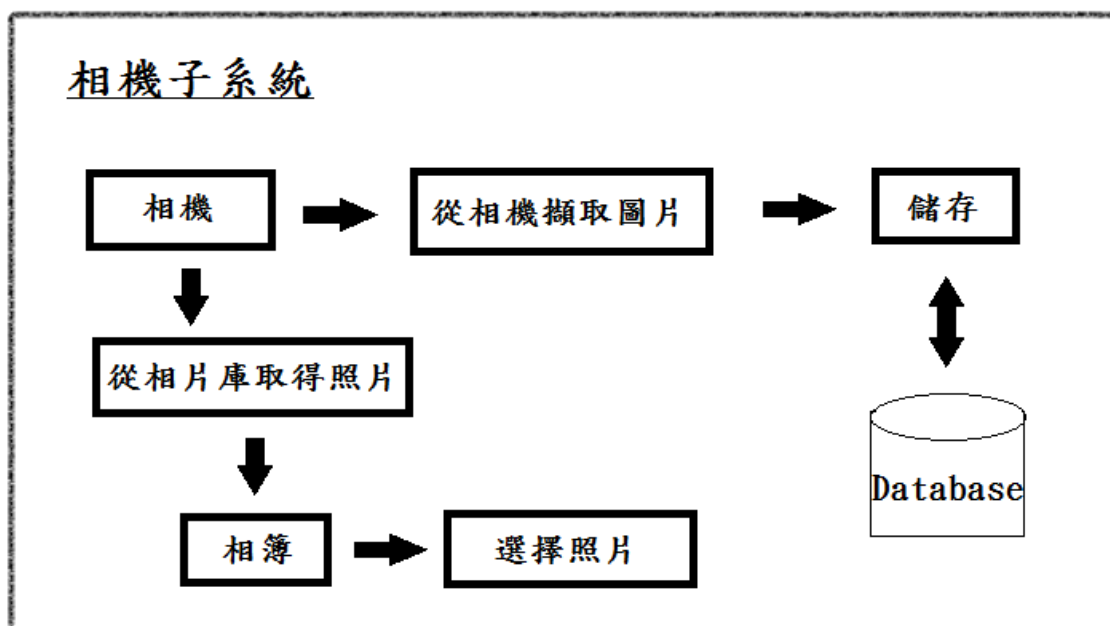


圖 3.4 相機子系統

3.3.3 旅遊日曆子系統

旅遊日記子系統，能夠快速瀏覽過去發生的事件，使用日曆的方式呈現更加一目了然，還可以藉由選日期快速跳到該天檢視該天的情境。詳細資料庫設計如下圖 3.5

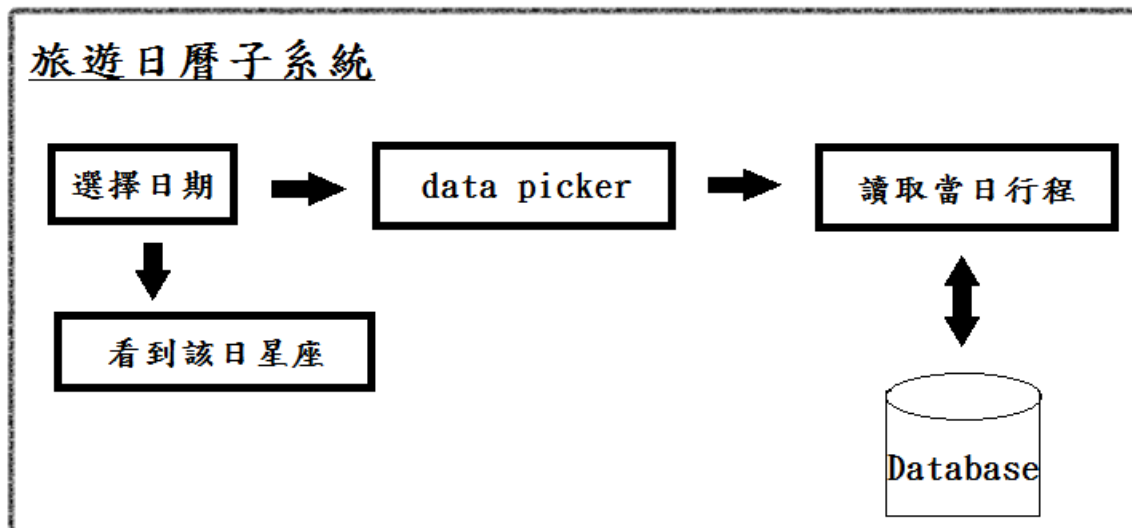


圖 3.5 旅遊日曆子系統

3.3.4 地圖子系統

地圖子系統，包含了可以設定要用哪種模式來看地圖，有標準模式、衛星模式跟混合模式，並且能夠更換大頭針的圖片，快速讓我們回想起曾在這個地點做了什麼樣的事情。還能使用註記的方式，讓我們記得曾在該處做了什麼事情等，並隨時更新資料庫去保存資料。詳細資料庫設計如下圖 3.6

圖 3.7 圖 3.8

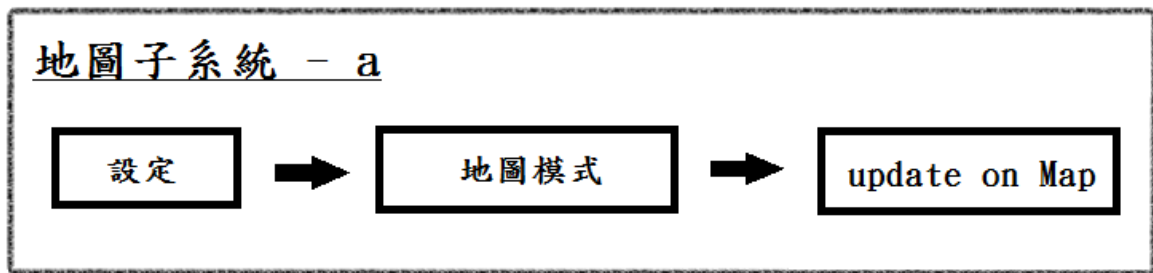


圖 3.6 地圖子系統 - a

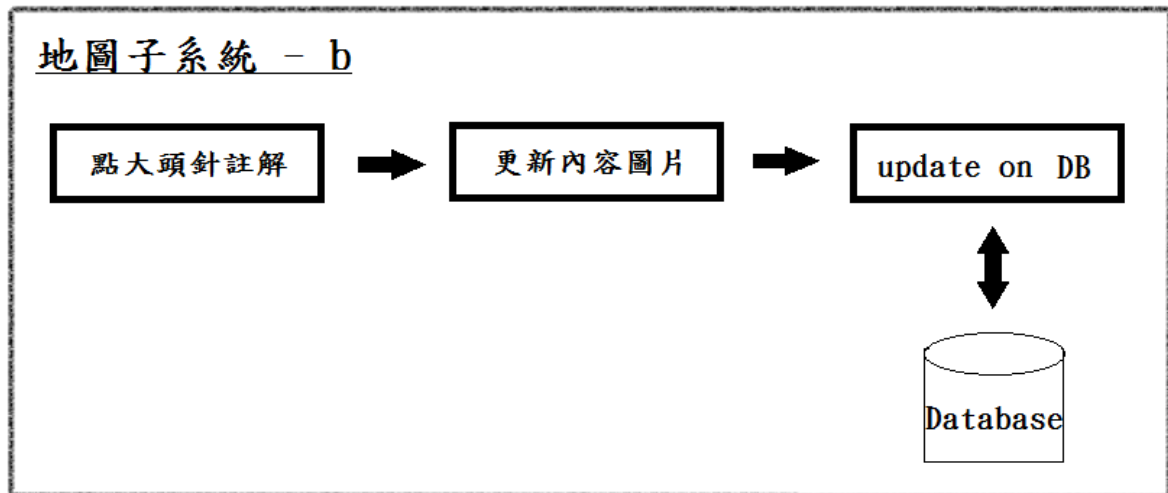


圖 3.7 地圖子系統 - b

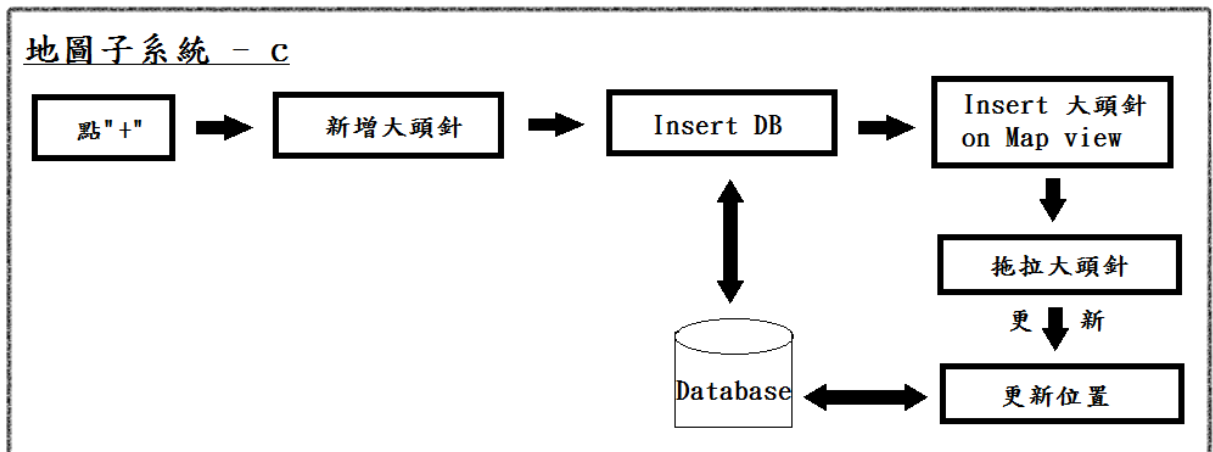


圖 3.8 地圖子系統 - c

第四章 系統架構

4.1 系統規格

- 名稱: 旅遊日記
- Category: Lifestyle
- Released: Nov18, 2011
- Version: 1.0.4
- Language: 繁體中文
- 適用: iPhone、iPod Touch, iOS 4以上環境

4.2 系統架構

旅遊日記主要有四項功能，包括記事、拍照、行事曆、地圖。

- 記事 - 隨時寫上每張照片的故事。
- 拍照 - 隨手隨拍。
- 旅遊日曆 - 在每個旅遊的日子標上照片及註記。
- 地圖 - 凡走過必留下痕跡，地圖大頭針。

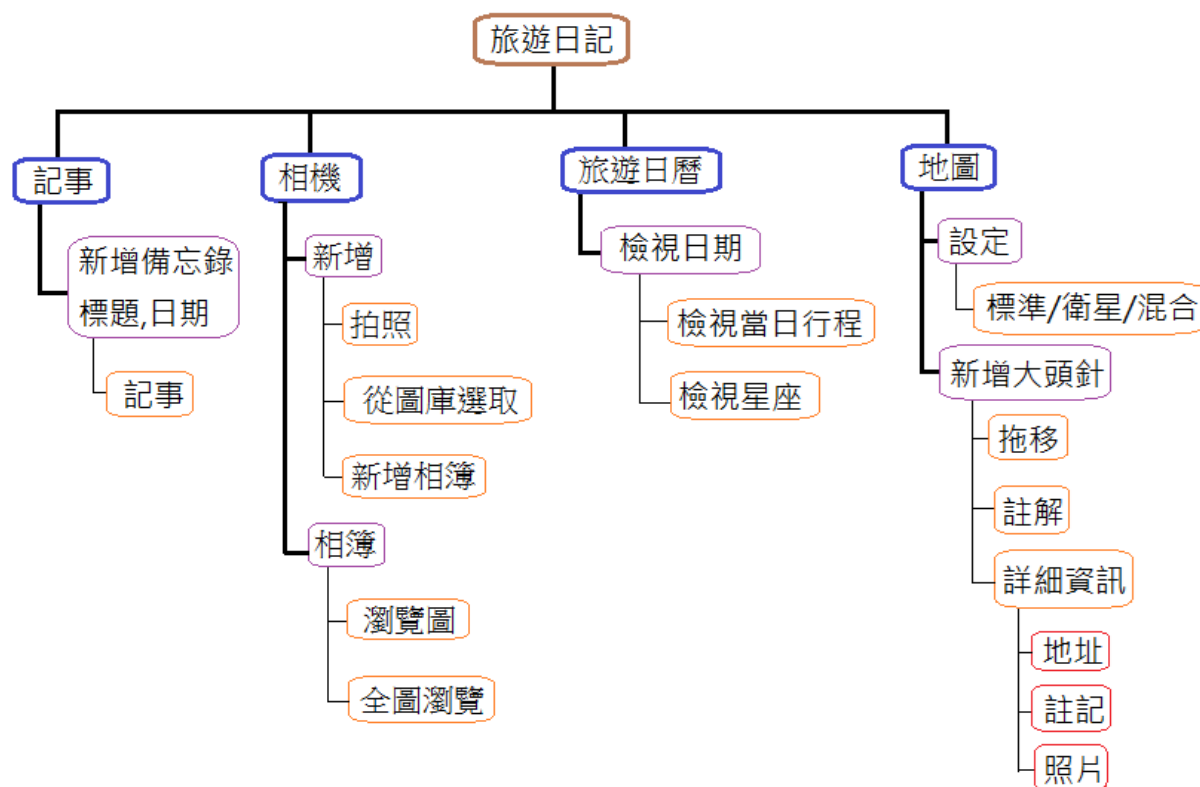


圖 4.1 系統架構圖

4.3 子系統規劃

Stocks	symbol	VARCHAR(50)	table 之標題
	Information	VARCHAR(1000)	紀錄備忘錄文字欄
	year	INT(4)	年
	month	INT(4)	月
	day	Int(4)	日

表 4.1 記事屬性表

subtable	thetitle	VARCHAR(50)	相簿資料夾名稱
	thesubtitle	VARCHAR(50)	相簿創建日期

表 4.2 相簿資料夾屬性表

photobook	Image	BloB	照片
	thetitle	VARCHAR	資料夾名稱
	thedata	VARCHAR	照片日期

表 4.3 照片屬性表

setText	<u>thetitle</u>	VARCHAR(20)	關聯Map 上之 thetitle
	text	VARCHAR(1000)	紀錄備忘錄文字欄

表 4.4 大頭針文字屬性表

IMAGES	<u>URL</u>	VARCHAR	關連Map 上之 thetitle
	IMAGE	BloB	顯示該大頭針的圖片

表 4.5 大頭針照片屬性表

Map	thetitle	VARCHAR(100)	大頭針主標題（關聯用）
	thesubtitle	VARCHAR(100)	紀錄地址 or 經緯度
	latitude	VARCHAR(20)	紀錄緯度
	longtitude	VARCHAR(20)	紀錄經度

表 4.6 地圖屬性表

4.4 資料庫規劃

4.4.1 relational schema

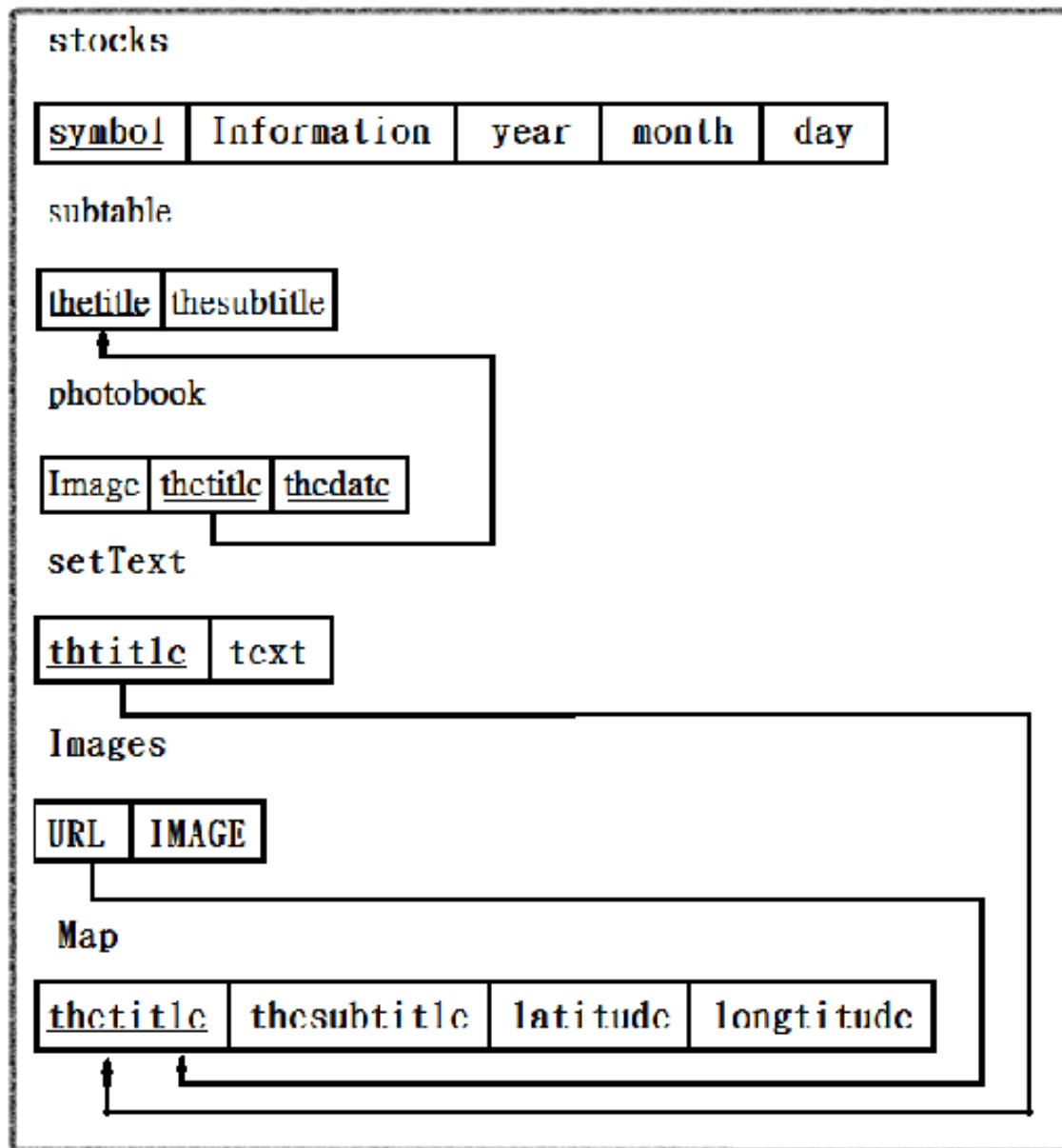


圖 4.2 relational schema

4.4.2 ER model

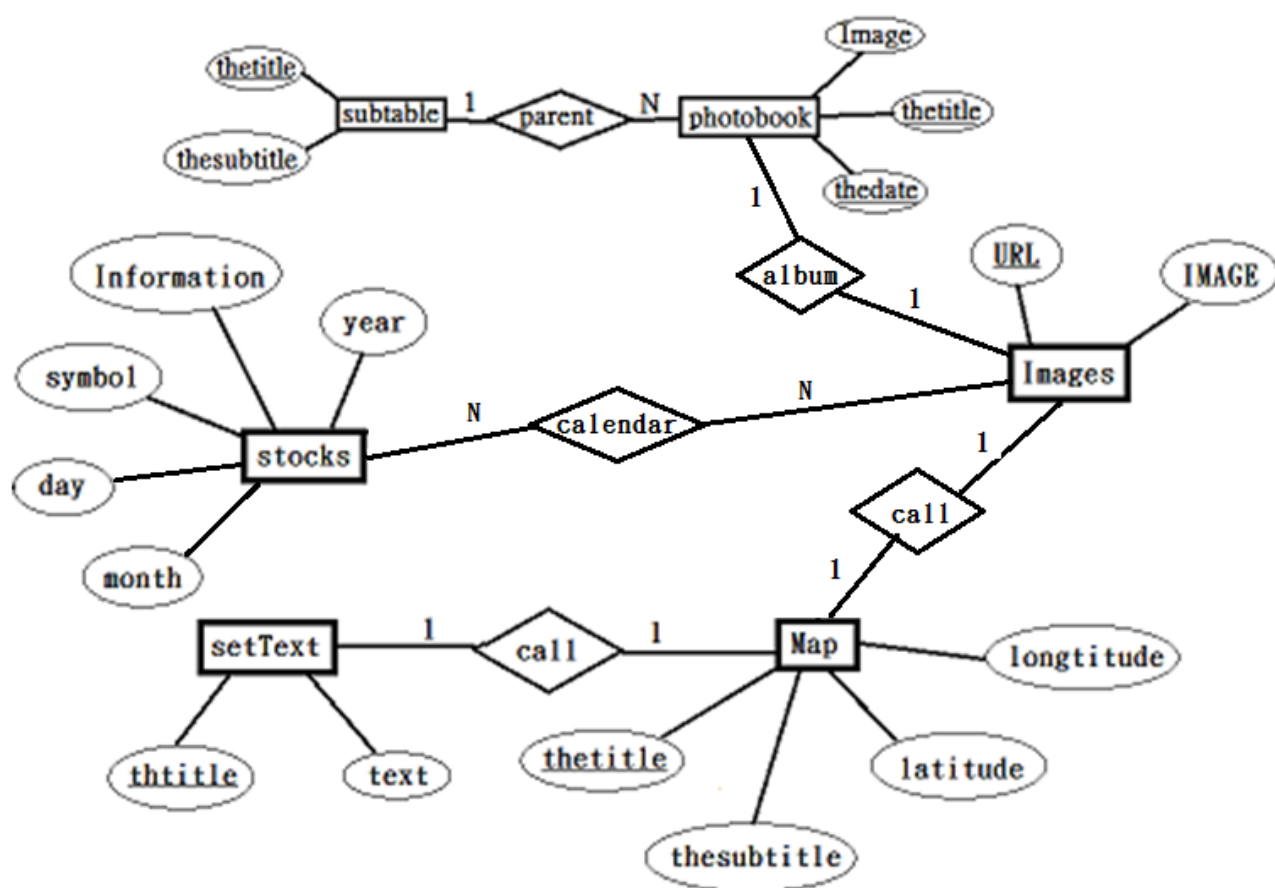


圖 4.3 ER model

第五章 人機介面設計

5.1 HTA 圖

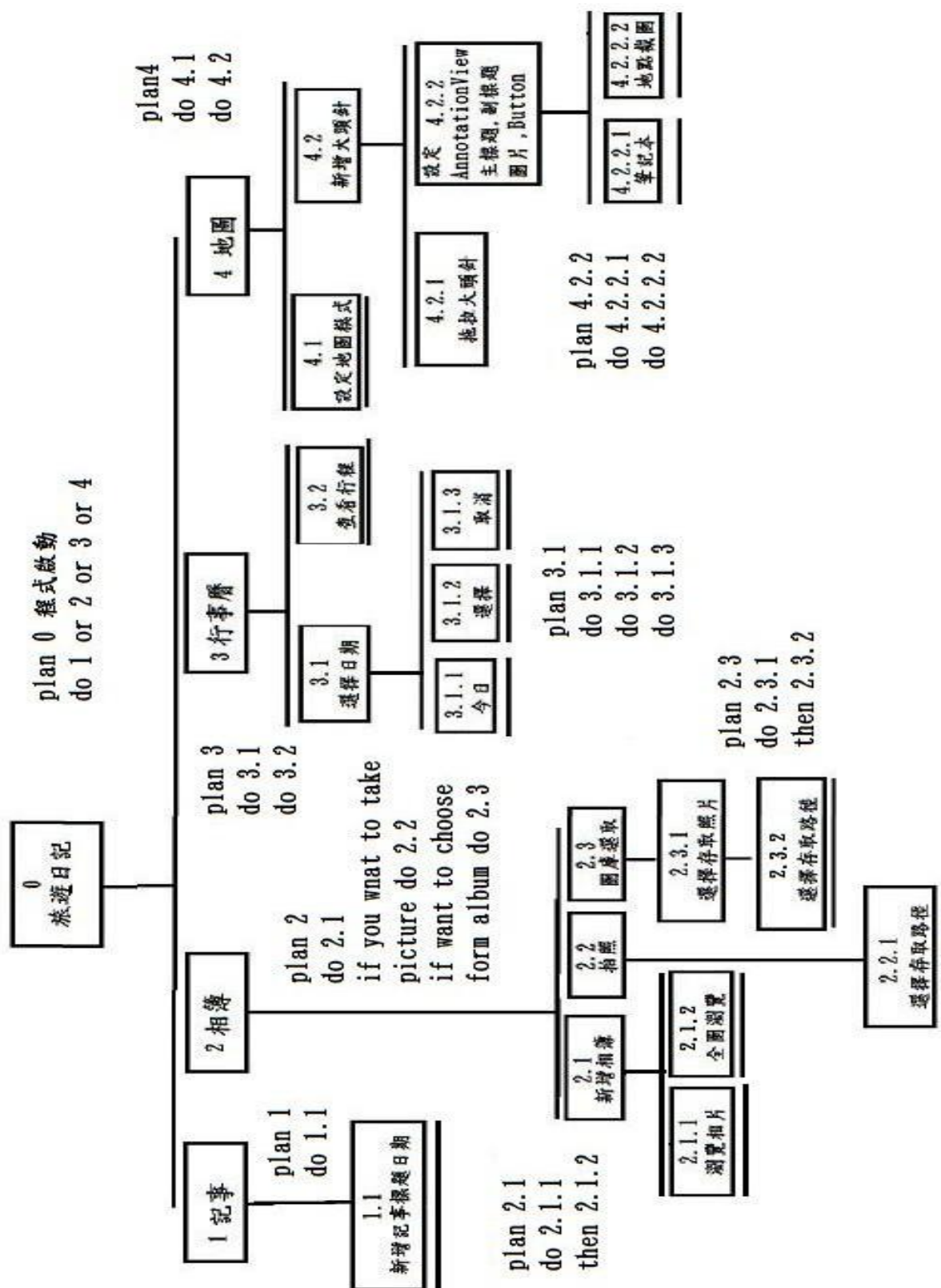


圖 5.1 HTA 圖

5.2 STD 圖

5.2.1 主畫面 STD

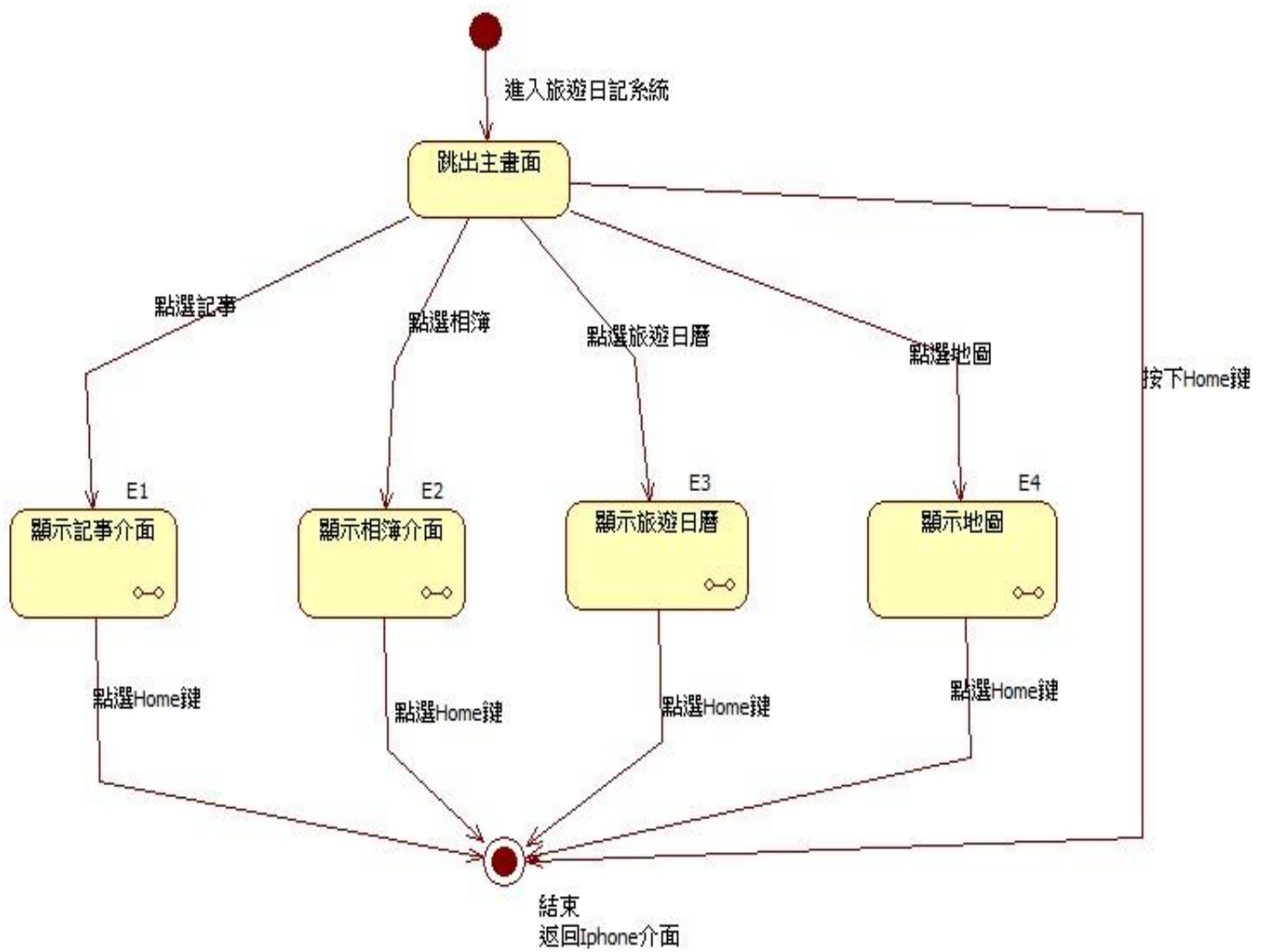


圖 5.2 STD – 主畫面

功能描述：我們將主畫面分成四個部份 其中包括 記事, 相簿, 旅遊日曆還有地圖
四個按鈕

5.2.2 記事 STD

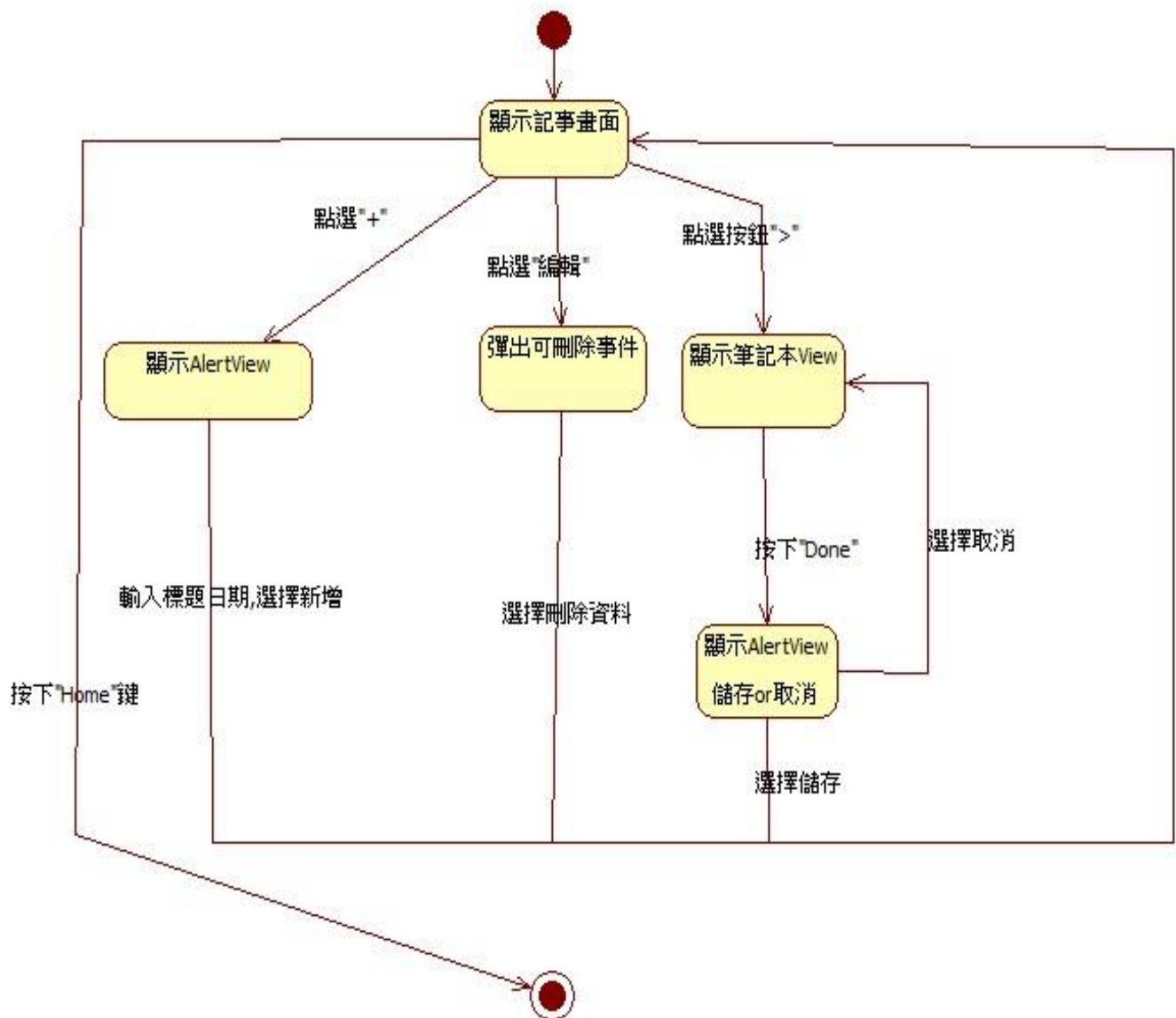


圖 5.3 STD – 記事

功能描述：如果停留在記事主畫面我們可以做新增事件，刪除的項目，還有進入筆記本做更詳細的設定

5.2.3 相機 STD

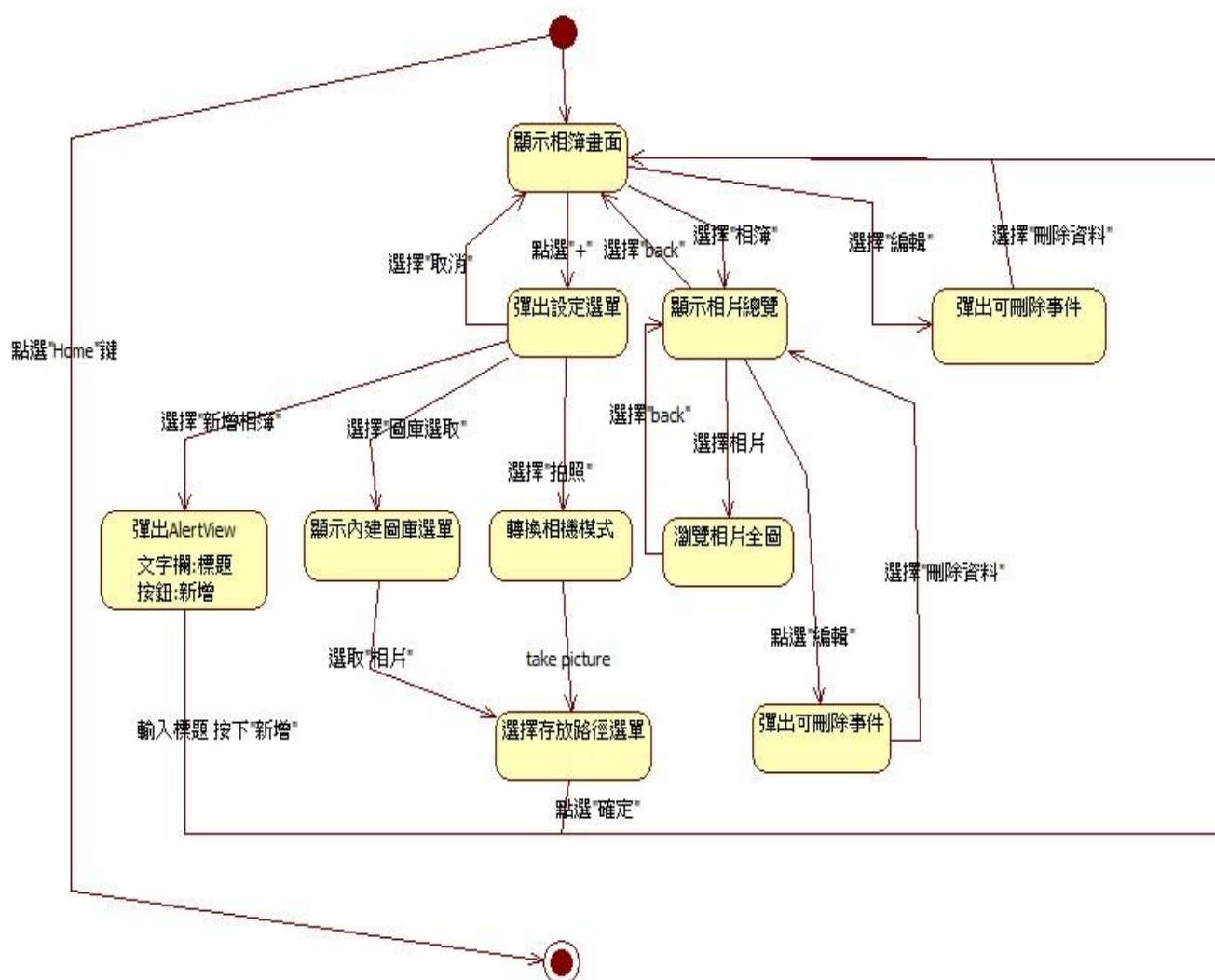


圖 5.4 STD-相機

功能描述：這是我們相簿的主畫面，按下新增按鈕我們會跳出一個表單讓我們選擇我們要做的動作，如果我們選擇拍照，程式會幫我們開啓相機拍完後可以讓我們選擇要存放的資料夾如果選擇由圖酷選取，程式會幫我們開啓內建圖庫去選擇相片然後在放入要分類的路徑，而新增相簿就是新增一個新的目錄讓我們去存放我們選擇的照片此功能可以全圖瀏覽相簿，刪除整個相簿，刪除單張相片

5.2.4 旅遊日曆 STD

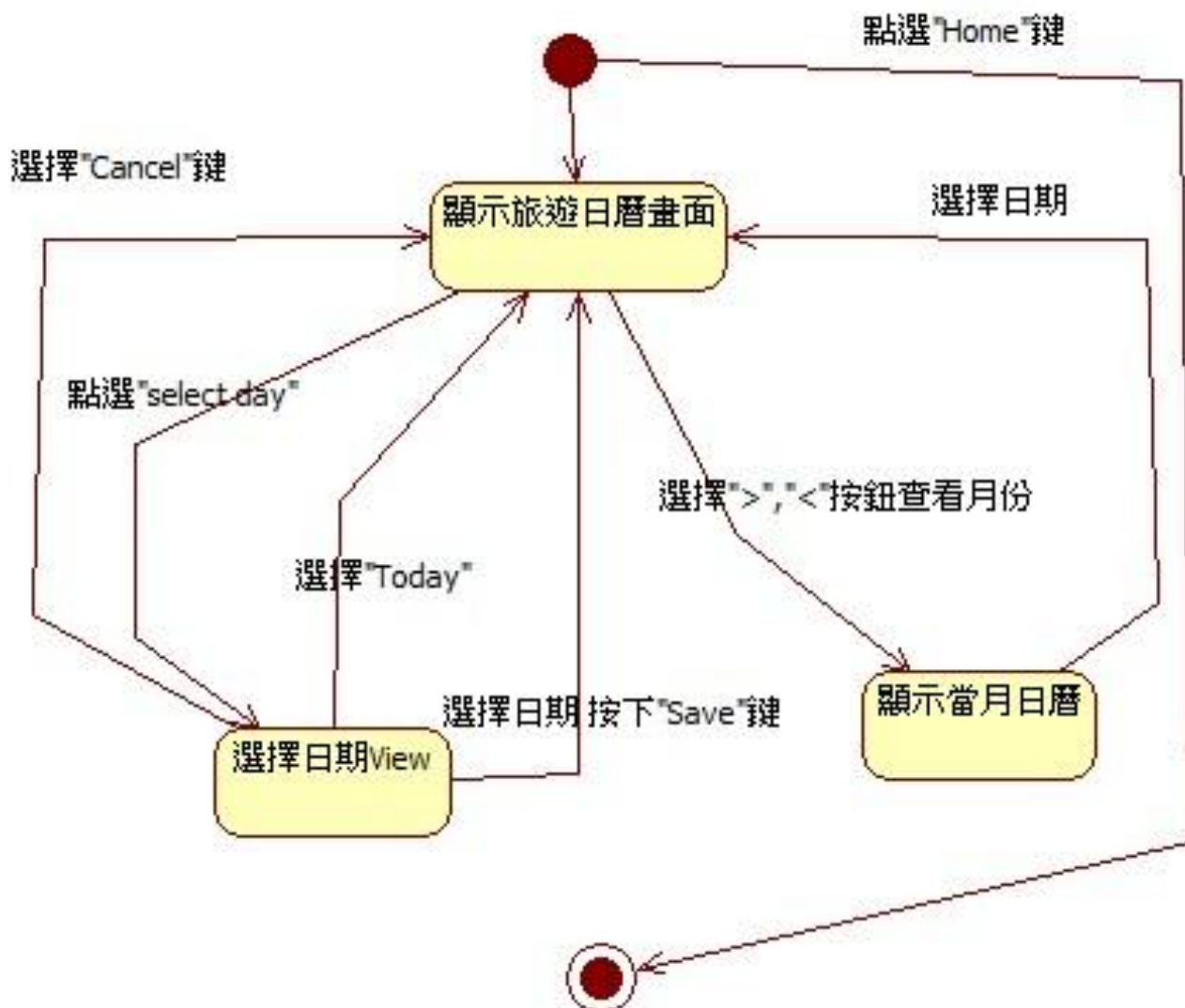


圖 5.5 STD – 旅遊日曆

功能描述：旅遊日曆主畫面可以讓我們做隔月的檢視行程, 當日有行程的話我們會在日期下方坐標點的動作提醒使用者這天有事情要辦理, 此功能也有直接跳躍到某天的功能, 藉由選取想要的日期來檢視當日的行程

5.2.5 地圖 STD

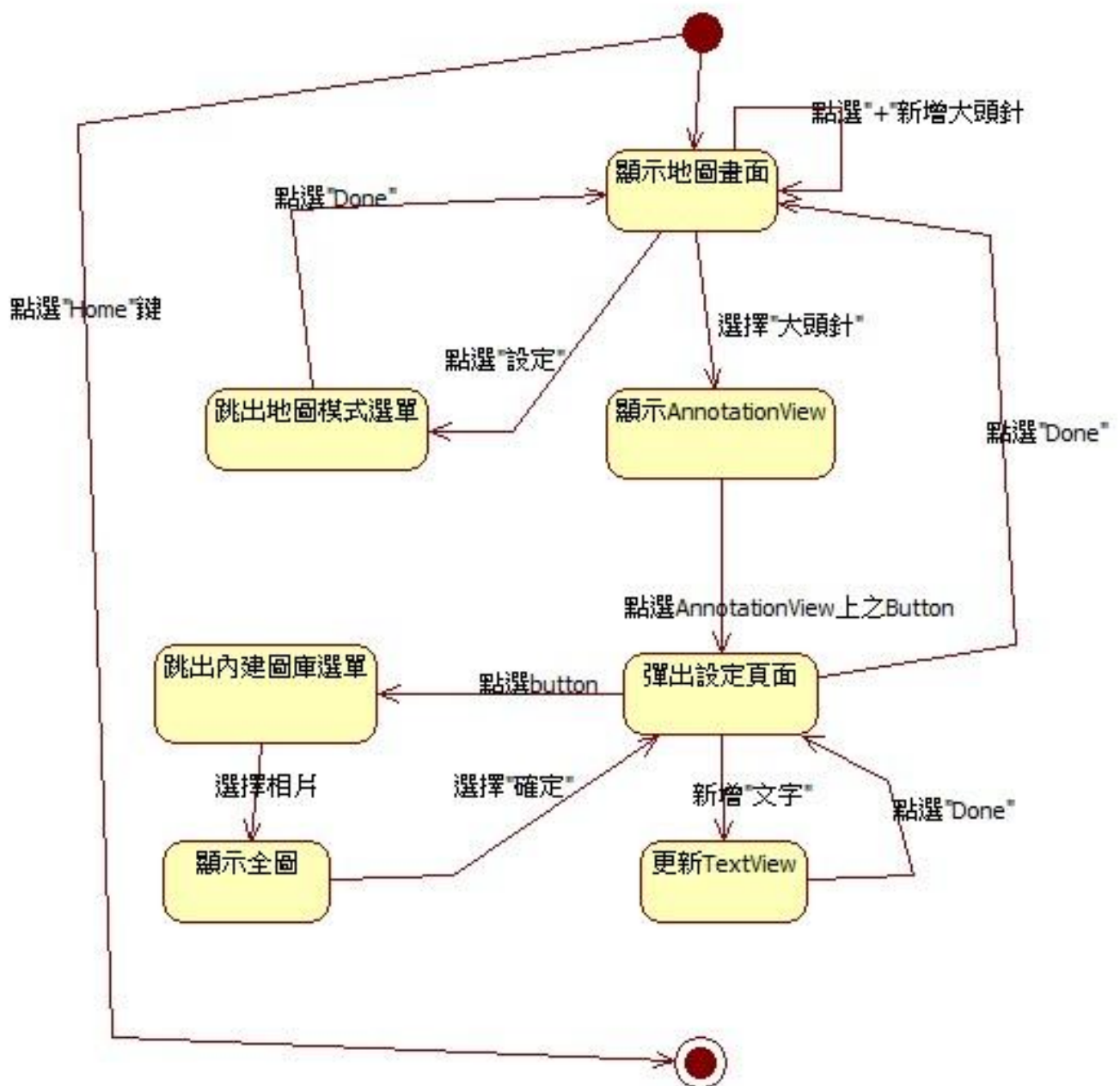


圖 5.6 STD - 地圖

功能描述：打開地圖主畫面一開始開啓MapView然後對目前位置坐定位，如果有移動就更新目前位置計算經緯度，按下新增來新增一個大頭針，當他拖移時重新計算經緯度，大頭針上有個詳細設定的按鈕讓我們可以對此地點做內文紀錄或是拍張照片，讓他顯現在AnnotationView上面如果點選設定的話可以切換地圖的模式總共有三種模式可以選，混合，標準，跟衛星模式的地圖

第六章 系統開發與實作

6.1 時程規劃

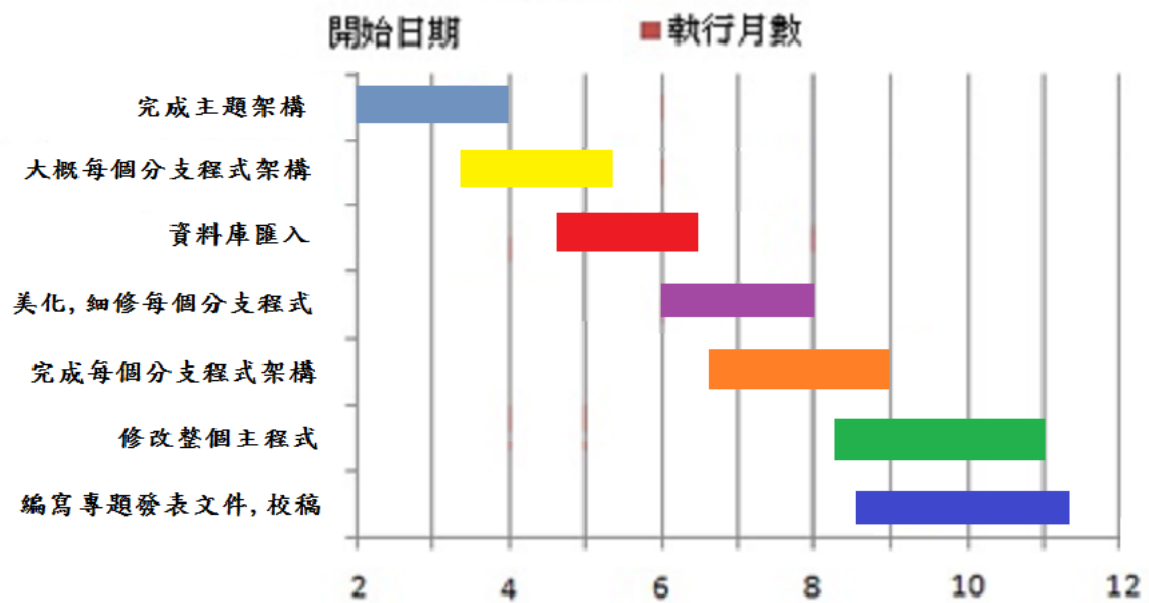
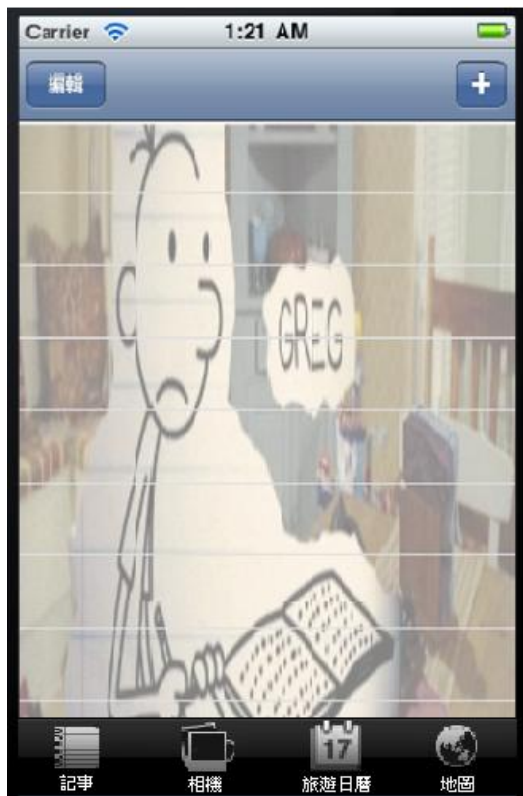


圖 6.1 時程表

- 2 ~ 4 月：構思專題方向，完成主體架構。
- 3/x ~ 5/x：將主題細分，建構每個子程式。
- 4/x ~ 6/x：將寫好的程式匯入資料庫。
- 6 ~ 8 月：美化、debug。
- 6/x ~ 9 月：完成每個子程式架構。
- 8/x ~ 11 月：修改整份程式。
- 8/x ~ 11/x：編寫文件。

6.2 系統實作成品

6.2.1 記事



記事的初始畫面

圖 6.2 記事 - a



點選+後的
新增畫面

圖 6.3 記事 - b

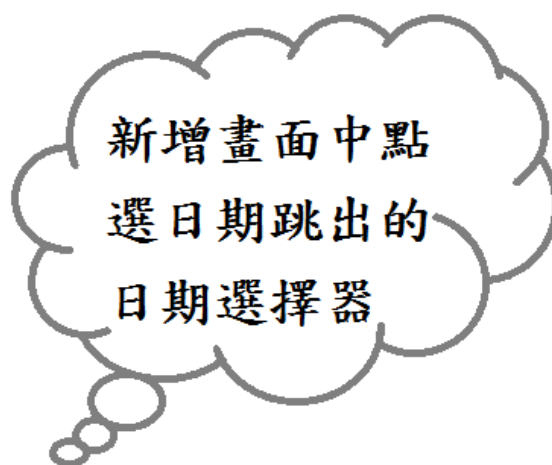


圖 6.4 記事 - c

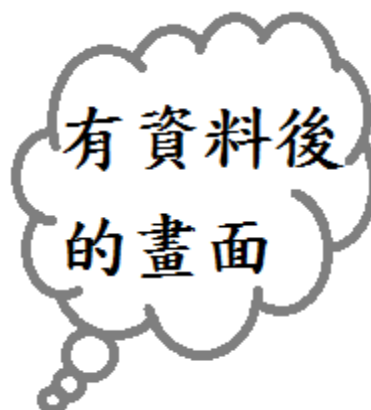


圖 6.5 記事 - d



手指往左右
兩邊拉，
可以出現
delete

圖 6.6 記事 - e



點擊記事
> 跳至可
以輸入資
料的畫面

圖 6.7 記事 - f



點選Done
選擇是否
要儲存

圖 6.8 記事 - g

6.2.2 相機



相機的
初始畫面

圖 6.9 相機 - a



圖 6.10 相機 - b

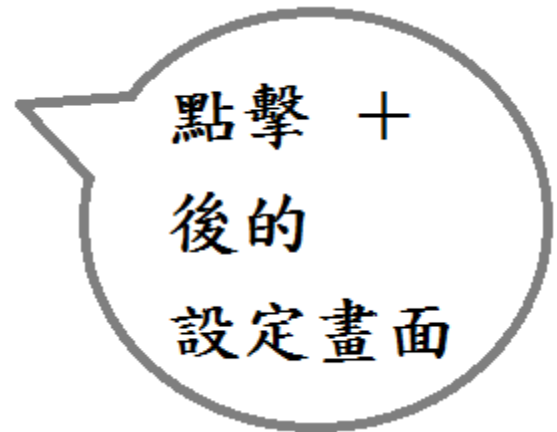


圖 6.11 相機 - c

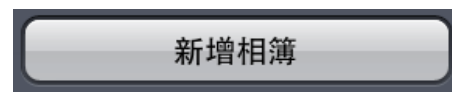




圖 6.12 相機 - d

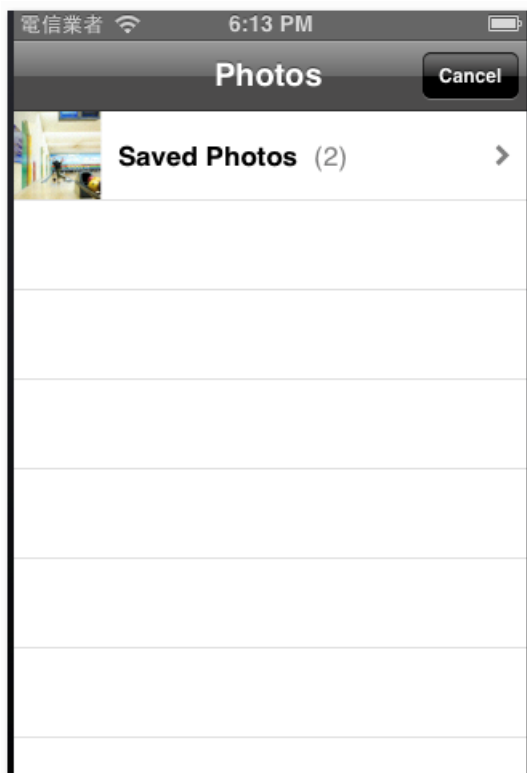
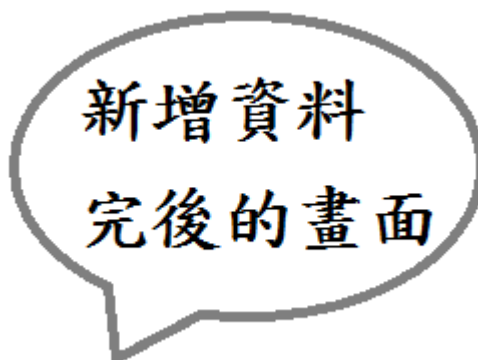
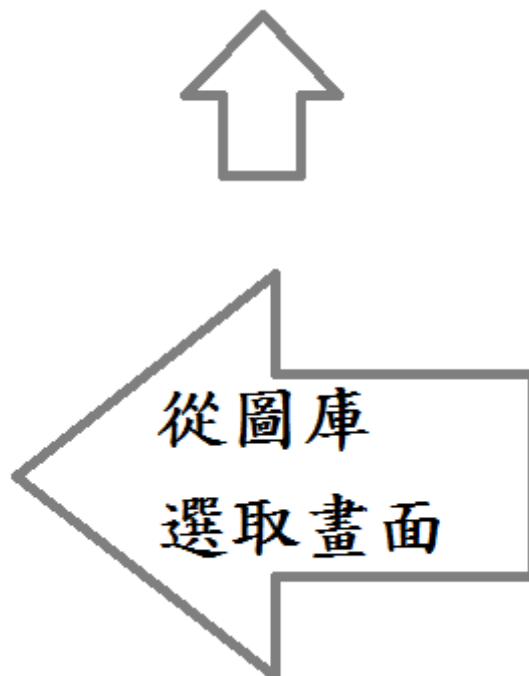
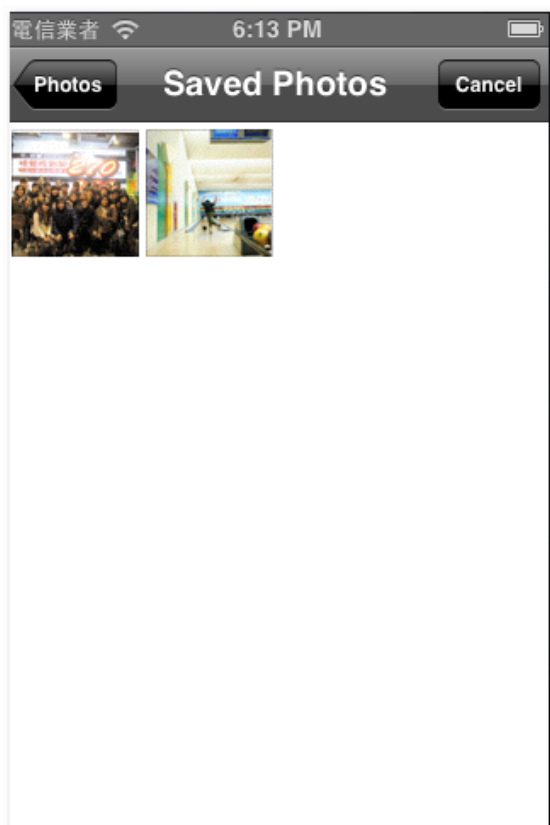


圖 6.13 相機 - e





圖庫畫面

圖 6.14 相機 - f



圖庫點選
照片後
之畫面

圖 6.15 相機 - g

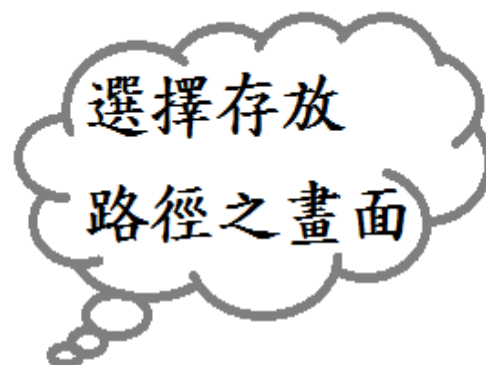


圖 6.16 相機 - h

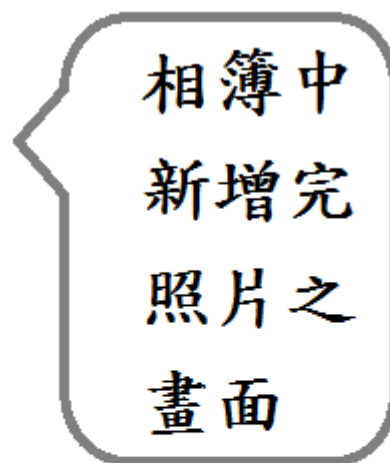
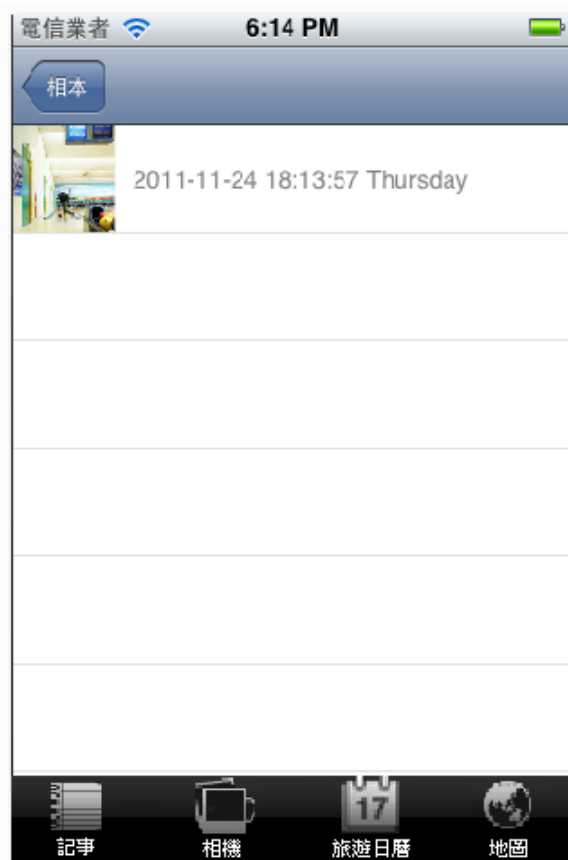


圖 6.17 相機 - i

6.2.3 旅遊日曆



行事曆
初始畫面

圖 6.18 旅遊日曆 - a



選擇完
日期之畫面

圖 6.19 旅遊日曆 - b



點擊

select day

跳出日期

選擇器

圖 6.20 旅遊日曆 - c



顯示

不同日期

的行程

圖 5.21 旅遊日曆 - d

5.2.4 地圖

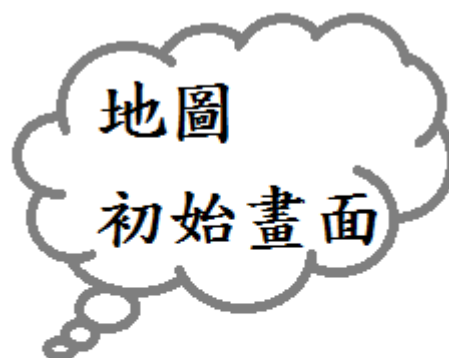


圖 5.22 地圖 - a

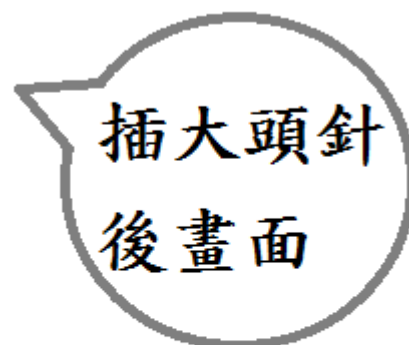


圖 5.23 地圖 - b



大頭針
的詳細
資料畫面

圖 5.24 地圖 - c



點選圖片
的選擇
模式

圖 5.25 地圖 - d



新增完
資料畫面

圖 5.26 地圖 - e



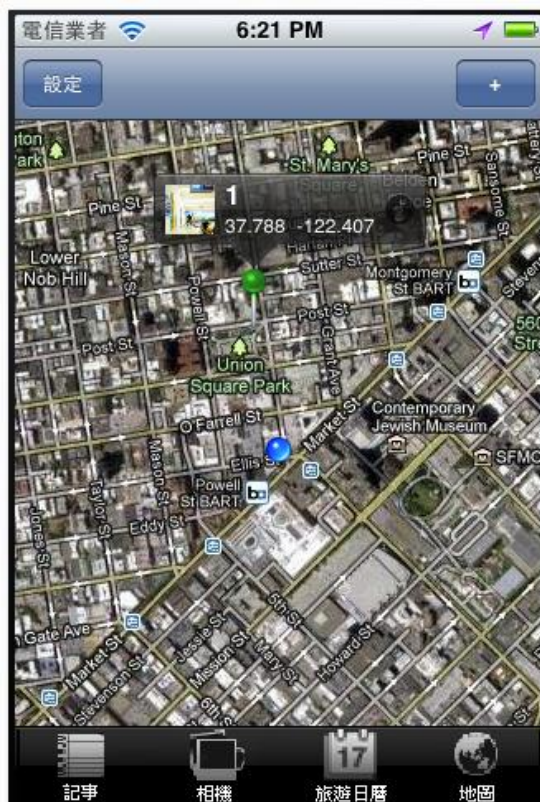
點選地圖
主畫面上
設定之
畫面

圖 5.27 地圖 - f



點選衛星
之後畫面

圖 5.28 地圖 - g



點選混合
之後畫面

圖 5.29 地圖 - h

參考文獻

[1] iPhone SDK 3 App 程式設計

作者：鄭斐文, 譯者：許郁文

出版社：旗標

[2]探索iPhone程式開發實戰

David Mark, Jack Nutting, Jeff LaMarche 著、邵子卿、洪沛然 譯

出版商:基峰

[3]iPhone程式設計範例經典

Paul Deitel, Harvey Deitel, Abbey Deitel, Eric Kern, Michael Mor 著、楊仁和 譯

出版商:基峰

[4] iPhone 創意程式設計家, 2/e (適用 SDK 3、SDK 4)

作者：林柏全

出版商:松崗

[5] iPhone SDK 應用程式開發實務

作者：鶴□賢吾

出版商:悅知

[6] 掌握 iPhone SDK 程式開發技巧

Maher Ali 著、羅友志 譯

出版商:上奇科技

[7]iDevTW - iPhone/iPad iOS開發者論壇

<http://dev.iphonetw.net/>

[8]CocoaChina - 蘋果開發中文站

<http://www.cocoachina.com/>

[9]CocoaControls

<http://cocoacontrols.com/>

[10]Furance iOS 程式設計中文學習網站
<http://furnacedigital.blogspot.com/>

[11]iPhone Dev SDK Forum
<http://www.iphonedevsdk.com/forum/>