

中華大學資訊工程學系

=====

101學年度專題製作期末報告

製作 iPhone 手機軟體程式

QR code 與通訊錄合併應用

=====

我們發現近幾年發展出的 QR CODE，比一般商品的條碼，只能儲存少數的資訊，QR CODE 能儲存很多很多的訊息，不論中英日文還是數字，所以有鑑於他的方便性與發展性，我們希望透過名片上加印 QR CODE，只要把身上的智慧型手機哪出來快速掃描不用幾秒鐘，就能將資訊直接輸入進去通訊錄，這樣一來一定會省掉很多輸入的時間，充分利用到這兩大科技，不僅不失禮儀，也可以省下你更多的寶貴時間。

指導老師：

梁秋國

組員：

B09729014 江易霖

B09729030 謝鴻志

目錄

1. 摘要與簡介.....	3,4
1.1. 摘要.....	3
1.2. 簡介.....	4
2. 研究背景.....	5,6,7
2.1. 背景.....	5,6
2.2. 目標.....	7
3. 開發環境.....	8
4. 專題開發流程.....	9,10
4.1. 專題進度.....	9
4.2. 專題分工.....	10
5. 專題內容.....	11-22
5.1. 製作與流程.....	11-21
5.2. 程式畫面.....	22
6. 評估與展望.....	23,24
7.1. 評估.....	23
7.2. 展望.....	24
7. 參考文獻.....	25

摘要與簡介

摘要

科技的發展非常迅速，人們越來越繁忙，時常出外工作，時間的控管就變成一個很重要的課題。而我們以這個為出發點，想做一個可以在一些繁忙的事務裡可以縮短時間，再加上響應政府的節能減碳的環保政策，從這兩點的考慮，我們發覺在新認識對方，以及在談生意的時候都會有的東西：“名片”，想必大家身上一定有個幾張。但是在輸入的時候卻又顯的麻煩？所以我們又聯想到另一個東西：

“QRCode”。

“QRCode”，最近突然很火熱的一個東西，在電視上常聽到 **qrcode** 這個名詞，索性研究了一下，發現是可以使用在我們的構想中，這樣可以省去在手機輸入的時間，以及不必要的紙張浪費，在人手一機的情況下，業務人員帶上智慧型手機早就是普遍的現像，這樣既可以節省時間，又可以節省資源的情況下，我們心中的這個 **app** 就產生了。

“QRCode” + “名片” = “？”

我們繼續看下去。

簡介

“QRCode”，是二維條碼的一種，1994年由日本 **DENSO WAVE** 公司發明。**QR** 來自英文「**Quick Response**」的縮寫，即快速反應的意思，源自發明者希望 **QR** 碼可讓其內容快速被解碼。**QR** 碼最常見於日本，並為目前日本最流行的二維空間條碼。**QR** 碼比普通條碼可儲存更多資料，亦無需像普通條碼般在掃描時需直線對準掃描器。

QR 碼呈正方形，只有黑白兩色。在 **3** 個角落，印有較小，像「回」字的的正方圖案。這 **3** 個是幫助解碼軟體定位的圖案，使用者不需要對準，無論以任何角度掃描，資料仍可正確被讀取。

日本 **QR** 碼的標準 **JIS X 0510** 在 **1999** 年 **1** 月發佈，而其對應的 **ISO** 國際標準 **ISO/IEC18004**，則在 **2000** 年 **6** 月獲得批准。根據 **Denso Wave** 公司的網站資料，**QR** 碼是屬於開放式的標準，**QR** 碼的規格公開，雖由 **Denso Wave** 公司持有的專利權益，但不會被執行。

除了標準的 **QR** 碼之外，也存在一種稱為「微型 **QR** 碼」的格式，是 **QR** 碼標準的縮小版本，主要是為了無法處理較大型掃描的應用而設計。微型 **QR** 碼同樣有多種標準，最高可儲存 **35** 個字元。

QR 碼資料容量	
數字	最多 7,089 字元
字母	最多 4296 字元
二進位數 (8 bit)	最多 2,953 位元組
日文漢字／片假名	最多 1,817 字元(採用 Shift JIS)
中文漢字	最多 984 字元 (採用 UTF-8)
中文漢字	最多 1,800 字元 (採用 BIG5)

錯誤修正容量	
L 水平	7% 的字碼可被修正
M 水平	15% 的字碼可被修正
Q 水平	25% 的字碼可被修正
H 水平	30% 的字碼可被修正



研究背景

iOS :

iOS 是由蘋果公司開發的作業系統。最初是設計給 **iPhone** 使用，後來陸續套用到 **iPod touch**、**iPad** 以及 **Apple TV** 產品上。就像其基於的 **Mac OS X** 作業系統一樣，它也是以 **Darwin** 為基礎的。原本這個系統名為「**iPhone OS**」，直到 **2010** 年 **6 月 7 日 WWDC** 大會上宣布改名為「**iOS**」。

iOS 的系統架構：

分為四個層次：核心作業系統層（**the Core OS layer**）
核心服務層（**the Core Services layer**）
媒體層（**the Media layer**）
可輕觸層（**the Cocoa Touch layer**）
系統操作佔用大概 **1GiB** 的記憶體空間。

2007 年 **10 月 17 日**，史蒂夫·賈伯斯在一封張貼於蘋果公司網頁上的公開信上宣布軟體開發工具包。它將在 **2008** 年 **2 月** 提供給第三方開發商。軟體開發工具包於 **2008** 年 **3 月 6 日** 發布，並允許開發人員開發 **iPhone** 和 **iPod touch** 的應用程式，並對其進行測試，名為「**iPhone** 手機類比器」。然而，只有在付出了 **iPhone** 手機開發計劃的費用後，應用程式才能發布。自從 **Xcode 3.1** 發布以後，**Xcode** 就成為了 **iPhone** 軟體開發工具包的開發環境。

軟體開發工具包的歷史：

在 **2008** 年 **3 月 6 日**，**iPhone** 的軟體開發工具包在蘋果大會堂會議正式宣布。第一個 **Beta** 版本是 **iPhone SDK 1.2b1**（**build 5A147p**）它在發布後立即就能夠使用了，但是同時推出的 **App Store** 所需要的韌體更新則直到 **2008** 年 **7 月 11 日** 才發布。對於 **iPhone** 用戶，這個更新是免費的；然而，**iPod touch** 的用戶則需要為更新付出 **9.99** 美元。

SDK 的內容

由於 **iOS** 是從於 **Mac OS X** 核心演變而來，因此開發工具也是基於 **Xcode**。

該 **SDK** 可分類為下列幾大項：

觸控(**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

媒體

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

核心服務

網路

SQLite 嵌入式資料庫

地理位置(**GeoLocation**)

線程(**Threads**)

OS X 核心

TCP/IP 協議

套接字(**Sockets**)

電源管理

檔案系統(**File system**)

安全

和 **Xcode** 工具一樣，**SDK** 開發工具中包含的 **iPhone** 類比器用來在開發人員的電腦上類比 **iPhone** 的外觀和感覺。最初它被稱為阿斯類比器(**Aspen Simulator**)，它被重新命名於 **BETA2** 版中發布的 **SDK**。

請注意，**iPhone** 類比器並不是一個用於執行 **x86** 目標代碼的工具。

該 **SDK** 需要擁有英特爾處理器且執行 **Mac OS X Leopard** 系統的 **Mac** 才能使用。其他的作業系統，包括微軟的 **Windows** 作業系統和舊版本的 **Mac OS X** 都不被支援。

目標

在這人手一隻智慧型手機的時代，手機成了我們生活中密不可分的一部分，希望藉由這個**app**，可以省去在手機輸入的時間，以及不必要的紙張浪費。

開發環境

XCDOE :



1.Xcode 4(程式撰寫)

Xcode is Apple's powerful integrated development environment for creating great apps for Mac, iPhone, and iPad. Xcode includes the Instruments analysis tool, iOS Simulator, and the latest Mac OS X and iOS SDKs.

The Xcode interface seamlessly integrates code editing, UI design with Interface Builder, testing, and debugging, all within a single window. The embedded Apple LLVM compiler underlines coding mistakes as you type, and is even smart enough to fix the problems for you automatically.

2.Mac(虛擬測試) NTD 21000

2-1 Development Programmer -\$99(NTD 3300)

3.iPhone(實機測試) iPhone4 NTD 19000

=====

共花費**43300**。

專題開發流程

9-1月花了許多時間在**擬定題目**，以往嘗試過放大鏡、手電筒...等等不計其數的一些奇怪的想法，最後討論出的**qrcode+名片**也要歸功於老師所建議的。 功能:

在網路上尋找QRCode相關資料，幾經碰壁後找到了提供原始碼的一個網站，雖然提供出來，但是還是得實作一下，**製作QRCodeReader**。

在**AppLibrary**尋找有關**Addressbook**的資料，蘋果的網站上就像是msdn一樣，也是有提供sdk的使用方式以及類似的程式碼，我們採用了addressbook這個程式碼並加以修改，**製作Addressbook**。

尤於是Addressbook先寫出來，**兩個程式的合併**上卻出了很大的一個問題，一直出現沒有碰過的問題，在修改程式上花了許多的問題。最後經過老師的講解我們朝著分割、測試、合體的方式才正式的成功運作此APP。

最後再加上自己使用qrcode設計者提供的lib**寫成網頁(?)**。

以下是時間分布圖：

工作項目	9-1月	2月	3月	4月	5月	6月	7-8月
擬定題目							
尋找資料							
製作程式							
網頁(?)							
美化、除錯							

專題分工

江易霖：

查詢資料。

提供**mac**。

製作**QRCodeReader**。

合併程式。

實機測式。

指揮運作及表定時間。

謝鴻志：

提供**iPhone**。

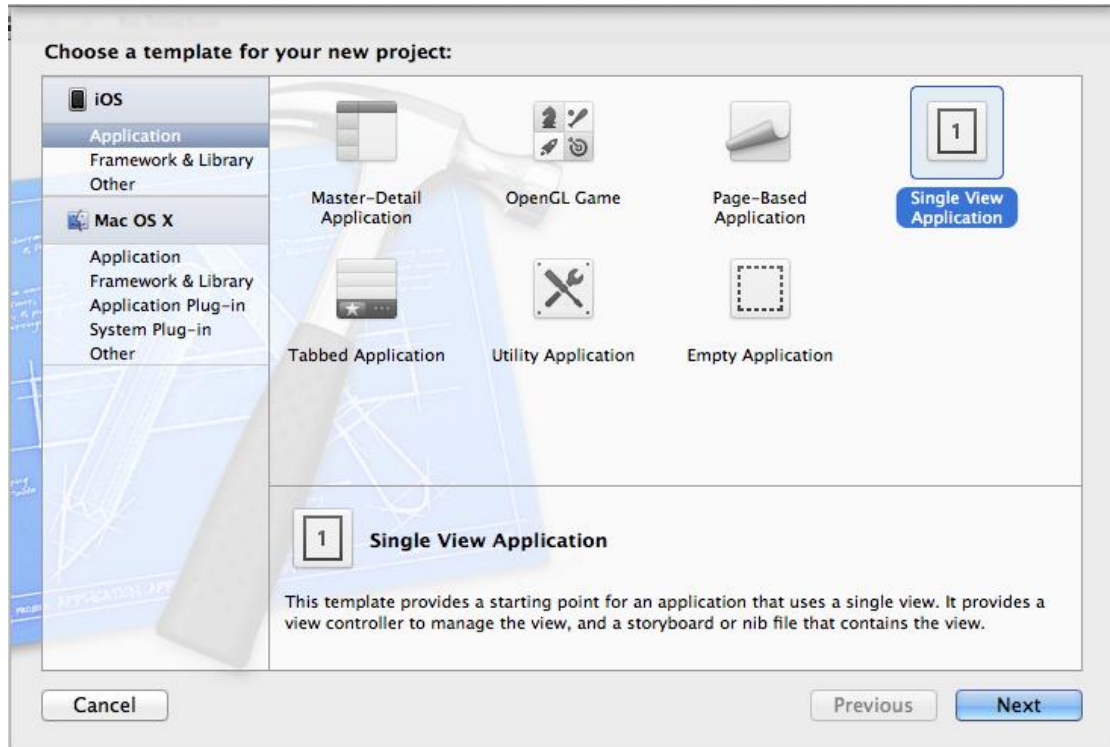
提供**Development Programmer**帳號

製作**Addressbook**。

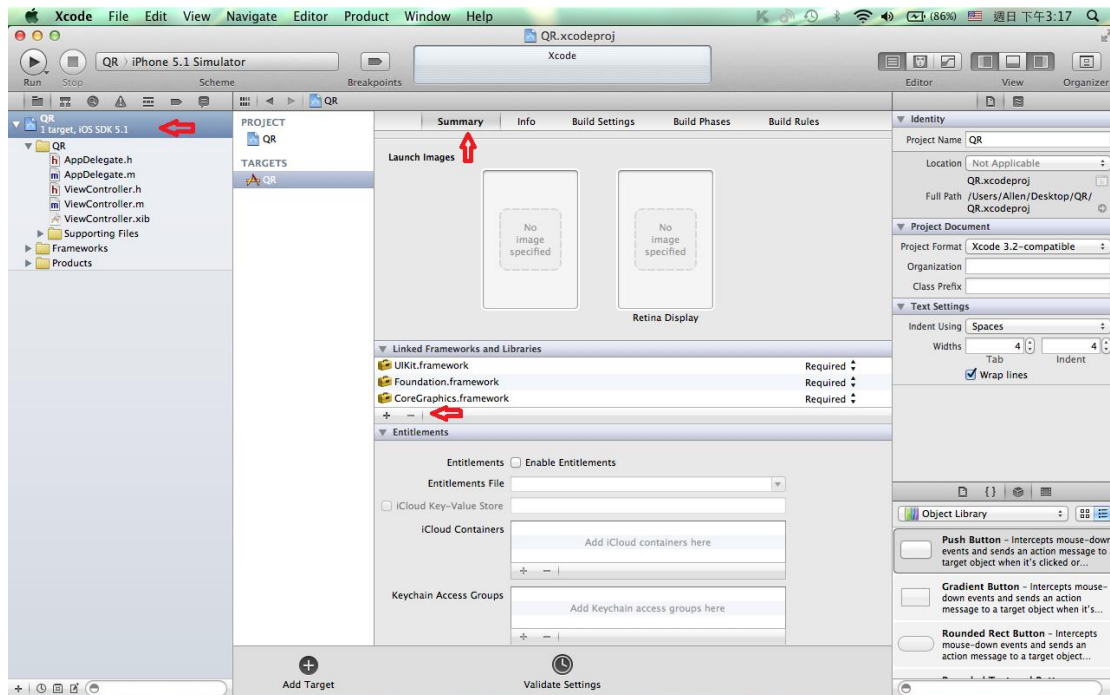
PTT製作、報告書製作、海報製作、上台報告。

專題內容

1. 首先建立一個 XCode 的檔案，並為之命名



2. 然後加入所需要的 Frameworks



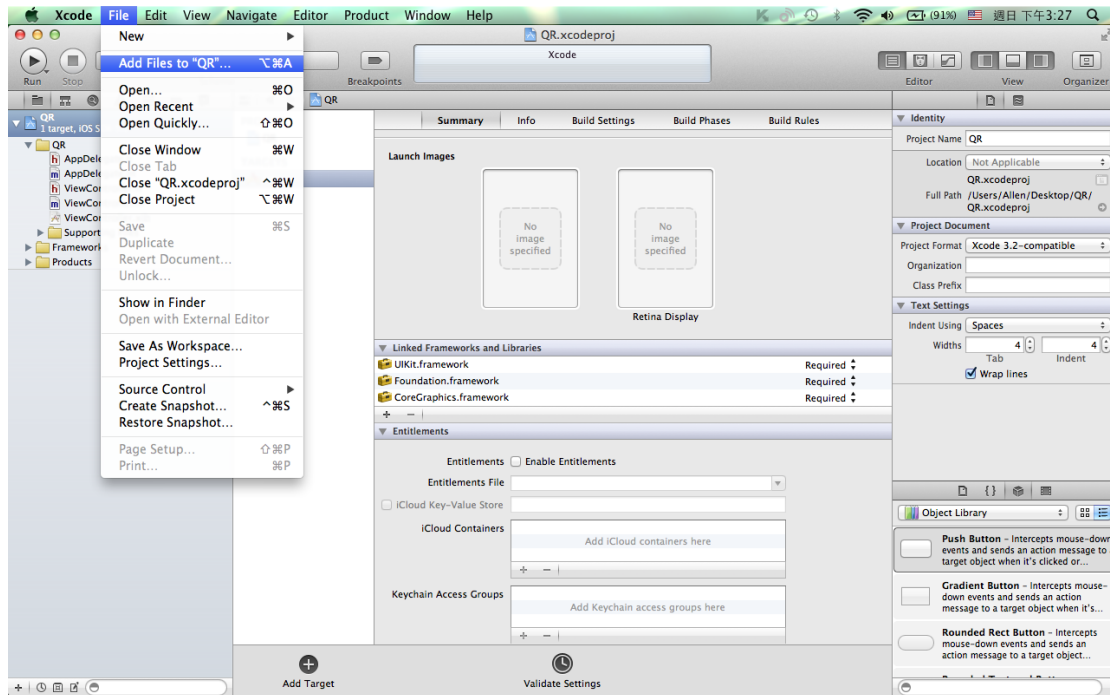
3. 加入下列所需要的 Frameworks

1. AudioToolbox.framework
2. CoreMedia.framework
3. QuartzCore.framework
4. SystemConfiguration.framework
5. libconv.dylib
6. AVFoundation.framework
7. CoreVideo.framework
8. AddressBook.framework
9. AddressBookUI.framework

4. 然後下載所需要的 ZBar.SDK

[http://sourceforge.net/projects/zbar/files/iPhoneSDK/ZBarSDK-1.2.dmg/download?
utm_expid=6384-3&utm_referrer=http%3A%2F%2Fzbar.sourceforge.net%2Fiphone%
2F](http://sourceforge.net/projects/zbar/files/iPhoneSDK/ZBarSDK-1.2.dmg/download?utm_expid=6384-3&utm_referrer=http%3A%2F%2Fzbar.sourceforge.net%2Fiphone%2F)

5. 加入 ZBar.SDK 到我們的檔案裡



6. 在 QuickContacts ViewController.h 寫入

```
#import <UIKit/UIKit.h>
#import <AddressBook/AddressBook.h> //加入AddressBook
#import <AddressBookUI/AddressBookUI.h> //加入AddressBookUI
#import "ZBarSDK.h" //加入ZBarSDK
```

//加入所使用的interface，並宣告成陣列使用

```
@interface QuickContactsViewController : UITableViewController <
ABPeoplePickerNavigationControllerDelegate,
    ZBarReaderDelegate>
```

```
{
    NSMutableArray *menuArray;
}
```

```
@property (nonatomic, retain) NSMutableArray *menuArray;
```

```
@property (nonatomic, retain) UIImagePickerController *imgPicker;
```

```
-(void)showPeoplePickerController;
```

```
-(void)showPersonViewController;
```

```
@end
```

7. 在 QuickContacts ViewController.m 寫入

```
#import "QuickContactsViewController.h"
#import <Foundation/Foundation.h>

enum TableRowSelected
{
    kUIDisplaysPickerRow = 0,
};

// Height for the Edit Unknown Contact row
#define kUIEditUnknownContactRowHeight 81.0

@implementation QuickContactsViewController
@synthesize menuArray;
@synthesize imgPicker;

#pragma mark Load views
// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Load data from the plist file
    NSString *plistPath = [[NSBundle mainBundle] pathForResource:@"Menu"
ofType:@"plist"];
    self.menuArray = [NSMutableArray arrayWithContentsOfFile:plistPath];
}

#pragma mark Unload views
- (void)viewDidUnload
{
    self.menuArray = nil;
}

#pragma mark Table view methods
```

```

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1; //[menuArray count];
}

// Customize the number of rows in the table view.
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    return 1;
}

// Customize the appearance of table view cells.
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *aCell = [tableView
    dequeueReusableCellWithIdentifier:CellIdentifier];
    if (aCell == nil)
    {
        // Make the Display Picker and Create New Contact rows look like buttons
        if (indexPath.section < 2)
        {
            aCell = [[[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] autorelease];
            aCell.textLabel.textAlignment = NSTextAlignmentCenter;
        }
        else
        {
            aCell = [[[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleSubtitle reuseIdentifier:CellIdentifier] autorelease];
            aCell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
            aCell.detailTextLabel.numberOfLines = 0;
            // Display descriptions for the Edit Unknown Contact and Display and Edit
            Contact rows
            aCell.detailTextLabel.text = [[menuArray objectAtIndex:indexPath.section]
            valueForKey:@"description"];
        }
    }
}

```

```

    }
}

aCell.textLabel.text = [[menuArray objectAtIndex:indexPath.section]
valueForKey:@"title"];

return aCell;
}

//電話
- (BOOL)addPhone:(ABRecordRef)person phone:(NSString*)phone
{
    ABMutableMultiValueRef multi =
ABMultiValueCreateMutable(kABMultiStringPropertyType);

    CFErrorRef anError = NULL;
    ABMultiValueIdentifier multivalueIdentifier;

    if (!ABMultiValueAddValueAndLabel(multi, (CFStringRef)phone,
kABPersonPhoneMobileLabel, &multivalueIdentifier)){
        CFRelease(multi);
        return NO;
    }

    if (!ABRecordSetValue(person, kABPersonPhoneProperty, multi, &anError)){
        CFRelease(multi);
        return NO;
    }

    CFRelease(multi);
    return YES;
}

- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath
*)indexPath
{
    [self showPeoplePickerController];
}

```



```

#pragma mark Show all contacts
// Called when users tap "Display Picker" in the application. Displays a list of contacts and
allows users to select a contact from that list.
// The application only shows the phone, email, and birthdate information of the selected
contact.
-(void)showPeoplePickerController
{
    ZBarReaderViewController *reader = [ZBarReaderViewController new];
    reader.readerDelegate = self;

    reader.readerView.torchMode = 0;

    ZBarImageScanner *scanner = reader.scanner;
    // TODO: (optional) additional reader configuration here

    // EXAMPLE: disable rarely used I2/5 to improve performance
    [scanner setSymbology: ZBAR_I25
                    config: ZBAR_CFG_ENABLE
                    to: 0];

    // present and release the controller
    [self presentViewController: reader
                        animated: YES];
    [reader release];

    //resultTextView.hidden=NO;
}

-(void) readerControllerDidFailToRead: (ZBarReaderController*) reader
                    withRetry: (BOOL) retry{
    NSLog(@"the image picker failing to read");
}

-(void) imagePickerController: (UIImagePickerController*) reader
didFinishPickingMediaWithInfo: (NSDictionary*) info
{

```

```

NSLog(@"the image picker is calling successfully %@",info);
// ADD: get the decode results
id<NSFastEnumeration> results = [info objectForKey: ZBarReaderControllerResults];
ZBarSymbol *symbol = nil;
NSString *hiddenData;
for(symbol in results)
    hiddenData=[NSString stringWithString:symbol.data];
NSLog(@"the symbols is the following %@",symbol.data);
// EXAMPLE: just grab the first barcode
// break;

// EXAMPLE: do something useful with the barcode data
//resultText.text = symbol.data;
//resultTextView.text=symbol.data;

NSLog(@"BARCODE= %@",symbol.data);

NSUserDefaults *storeData=[NSUserDefaults standardUserDefaults];
[storeData setObject:hiddenData forKey:@"CONSUMERID"];
NSLog(@"SYMBOL : %@",hiddenData);
//resultTextView.text=hiddenData;
[reader dismissModalViewControllerAnimated: NO];

//擷取“,”
/*-----*/

NSArray *stringArray = [hiddenData componentsSeparatedByString:@" "];//偵測,符號

NSLog(@"111%@",stringArray);
for (hiddenData in stringArray )
{
    ABRecordRef aContact = ABPersonCreate();
    CFErrorRef anError = NULL;
    //
    ABAddressBookRef iPhoneAddressBook = ABAddressBookCreate();

```

```

        NSString *first = [stringArray objectAtIndex:1];
ABRecordSetValue(aContact, kABPersonFirstNameProperty, first, &anError);
        NSString *last = [stringArray objectAtIndex:0];
ABRecordSetValue(aContact, kABPersonLastNameProperty, last, &anError);

        NSString *org = [stringArray objectAtIndex:2];
ABRecordSetValue(aContact, kABPersonOrganizationProperty, org,
&anError);

        NSString *phon = [stringArray objectAtIndex:3];
[self addPhone:aContact phone:phon];
        NSString *job = [stringArray objectAtIndex:4];
ABRecordSetValue(aContact, kABPersonJobTitleProperty, job, &anError);

ABAddressBookAddRecord(iPhoneAddressBook, aContact, &anError);

if (anError != NULL)
{

        NSLog(@"Danger Will Robinson! Danger!");

}

CFRelease(iPhoneAddressBook);

//郵件

NSString *mail = [stringArray objectAtIndex:5];
ABMultiValueRef email =
ABMultiValueCreateMutable(kABMultiStringPropertyType);

bool didAdd = ABMultiValueAddValueAndLabel(email, mail,
kABPersonHomePageLabel, NULL);

if (didAdd == YES)
{
        ABRecordSetValue(aContact, kABPersonEmailProperty, email,
&anError);

```

```

        if (anError == NULL)
        {
            ABUnknownPersonViewController *picker =
[[ABUnknownPersonViewController alloc] init];
            picker.displayedPerson = aContact;
            picker.allowsAddingToAddressBook = YES;

            picker.title = @"通訊錄";

            [self.navigationController pushViewController:picker
animated:YES];

            [picker release];
        }
        else
        {
            UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Error"

message:@"Could not create unknown user"

delegate:nil

cancelButtonTitle:@"Cancel"

otherButtonTitles:nil];

            [alert show];
            [alert release];
        }
        }
        CFRelease(email);
        CFRelease(aContact);
    }
}

#pragma mark ABPeoplePickerNavigationControllerDelegate methods

// 顯示聯絡人資訊

- (BOOL)peoplePickerNavigationController:(ABPeoplePickerNavigationController
*)peoplePicker shouldContinueAfterSelectingPerson:(ABRecordRef)person

```

```
{
```

```
    return YES;
```

```
}
```

```
#pragma mark ABUnknownPersonViewControllerDelegate methods
```

```
// Dismisses the picker when users are done creating a contact or adding the displayed person properties to an existing contact.
```

```
- (void)unknownPersonViewController:(ABUnknownPersonViewController
```

```
*)unknownPersonView didResolveToPerson:(ABRecordRef)person
```

```
{
```

```
    [self dismissModalViewControllerAnimated:YES];
```

```
}
```

```
#pragma mark Memory management
```

```
- (void)dealloc
```

```
{
```

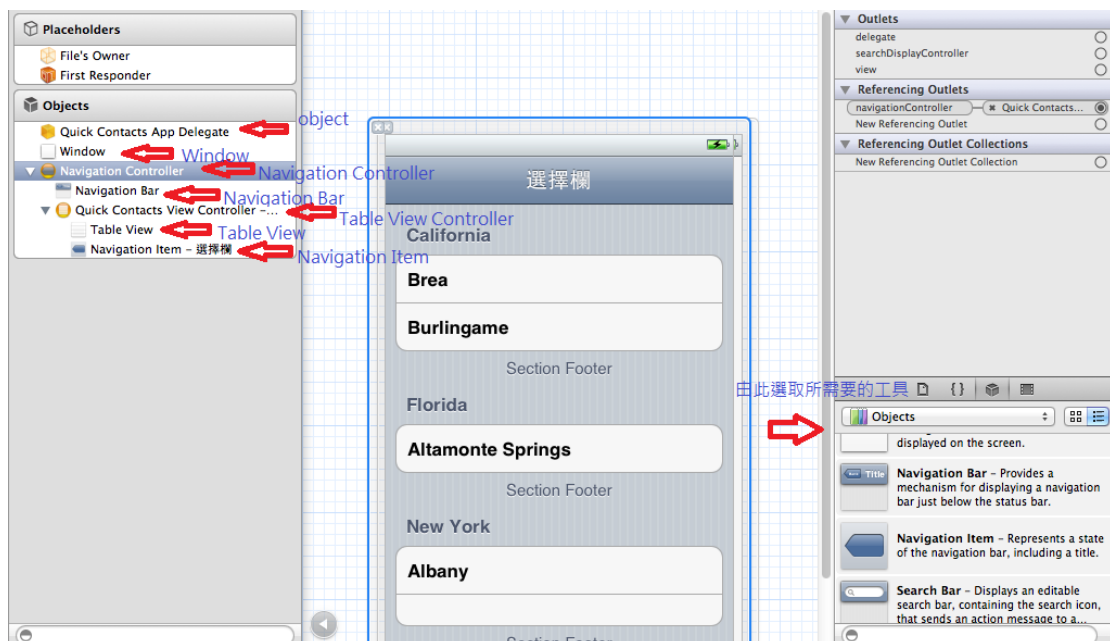
```
    [menuArray release];
```

```
    [super dealloc];
```

```
}
```

```
@end
```

8. 完成後在 QuickContacts ViewController.xib 裡面加入所需要的



程式畫面

點下 Display Picker，跳入相機畫面，照 QRCode 後，顯示結果，



評估與展望

評估：

在這個並不是很完整的**APP**中，希望給各位老師看到的是我們所要表達的，也就是讓生活更加方便、節省資源以及時間。從以往的專題發表中，並沒有找到有做**IOS**的專題，這對我們來說是一個挑戰，前無古人可以參考，也沒有人指導下，我們在寫程式以及找尋資料方面花了極大的時間，資料找到後，**Object-C**，這個並沒有教過的程式，雖然標榜著是“具有物件導向功能的**C**”，但是還是要從頭學起，裡面的物件、類別內容都跟以往學的**C**不太相同，所以在合併兩個寫好的程式之後，又花了一段時間才完成了現在這個版本，也許**BUG**很多，也許功能不完全，也請見諒。

展望：

我們打算再加入一些新功能；

比如：

shake (有點像是Line裡面的震動搜尋好友，而我們是靠著手搖手機的方式將自己所照好的，至做好資料可以靠這動作傳給對方)。

image(把自己照到的QRCode，儲存在通訊錄底下，這樣別人也可以使用此app直接照QRCode)。

make(可以自己就直接在手機上製作好屬於自己的QRCode，這樣就不必使用網頁了)。

參考文獻

addressbook:

- 蘋果官方網站的 sdklibrary
<http://developer.apple.com/library/ios/#documentation/ContactData/Conceptual/AddressBookProgrammingGuideforiPhone/Chapters/BasicObjects.html>

QRCodeScanner:

- qrcanner
<http://zbar.sourceforge.net/iphone/sdkdoc/>

所有技術支援：

- iOS SDK 4 初心者的學習殿堂
<http://www.drmaster.com.tw/Bookinfo.asp?BookID=PG30060>