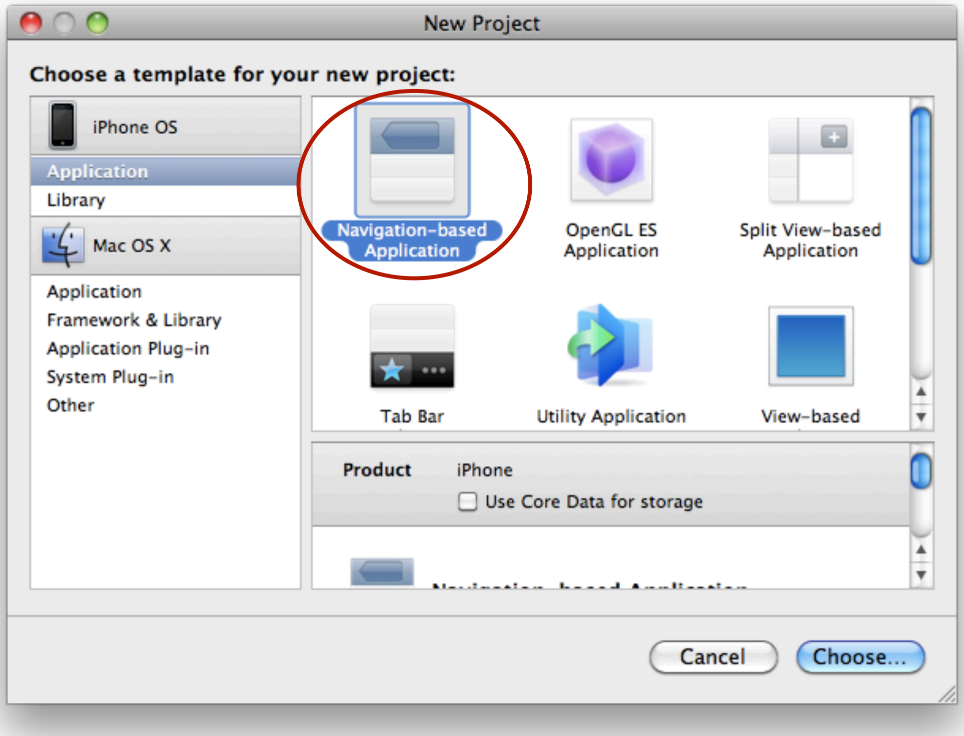
Lab ToDoList

Step1. 在File開啓一個新的project, 選擇 Navigation based application, 命名為 ToDoList

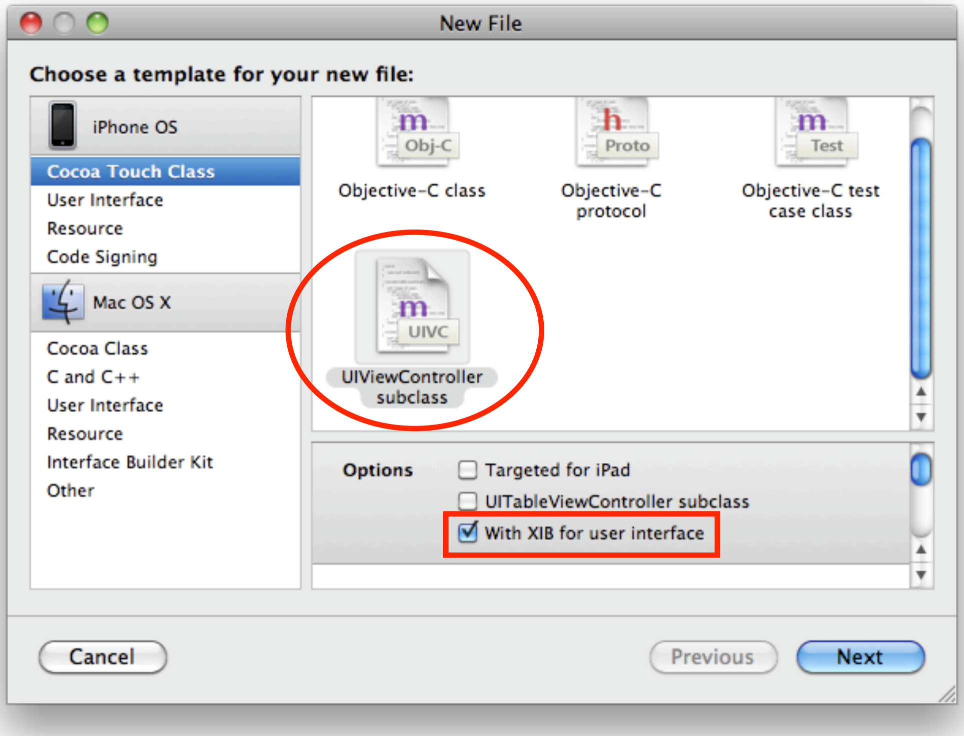


Choose a template for your new project:

Product iPhone

☐ Use Core Data for storage

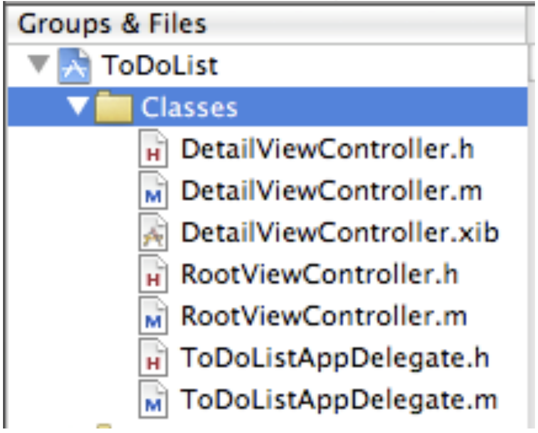
Step2. 新增一個File,選擇UIViewController subClass, 命名為DetailViewController, 同時勾選 With XIB for user interface 來同時建立對應的xib檔



Choose a template for your new file:

Options
☐ Targeted for iPad
☐ UITableViewController subclass
☑ With XIB for user interface

所以我們現在左邊Groups & Files視窗應該有下列檔案

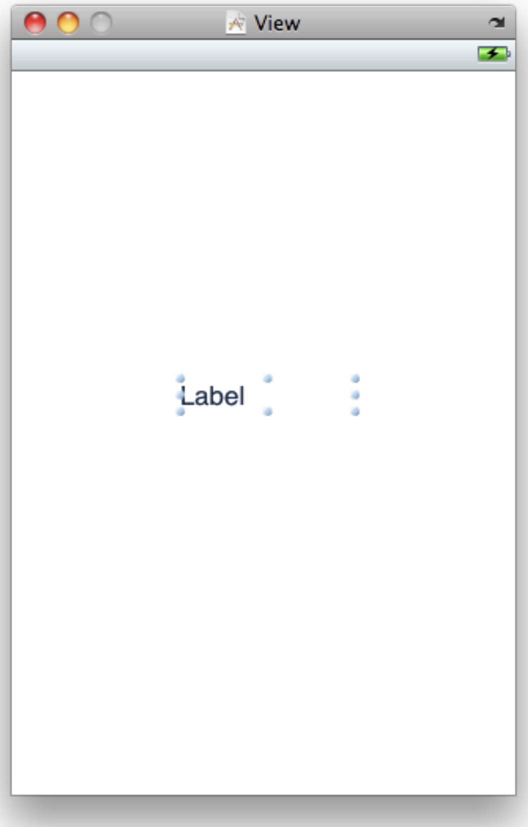


Groups & Files
- ToDoList
  - Classes
    - DetailViewController.h
    - DetailViewController.m
    - DetailViewController.xib
    - RootViewController.h
    - RootViewController.m
    - ToDoListAppDelegate.h
    - ToDoListAppDelegate.m

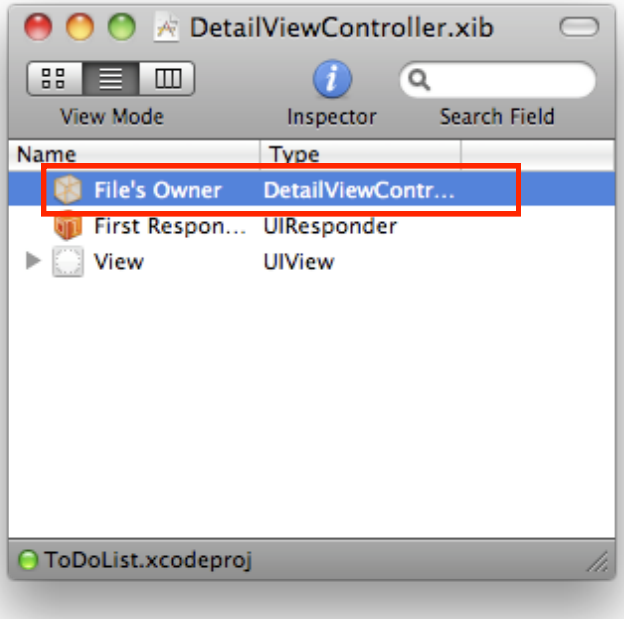
Step3. 首先在剛剛新增的DetailViewController.h裡新增一個label和一個NSString來作為顯示detail資訊使用,並對這個NSString做@property來方便我們存取

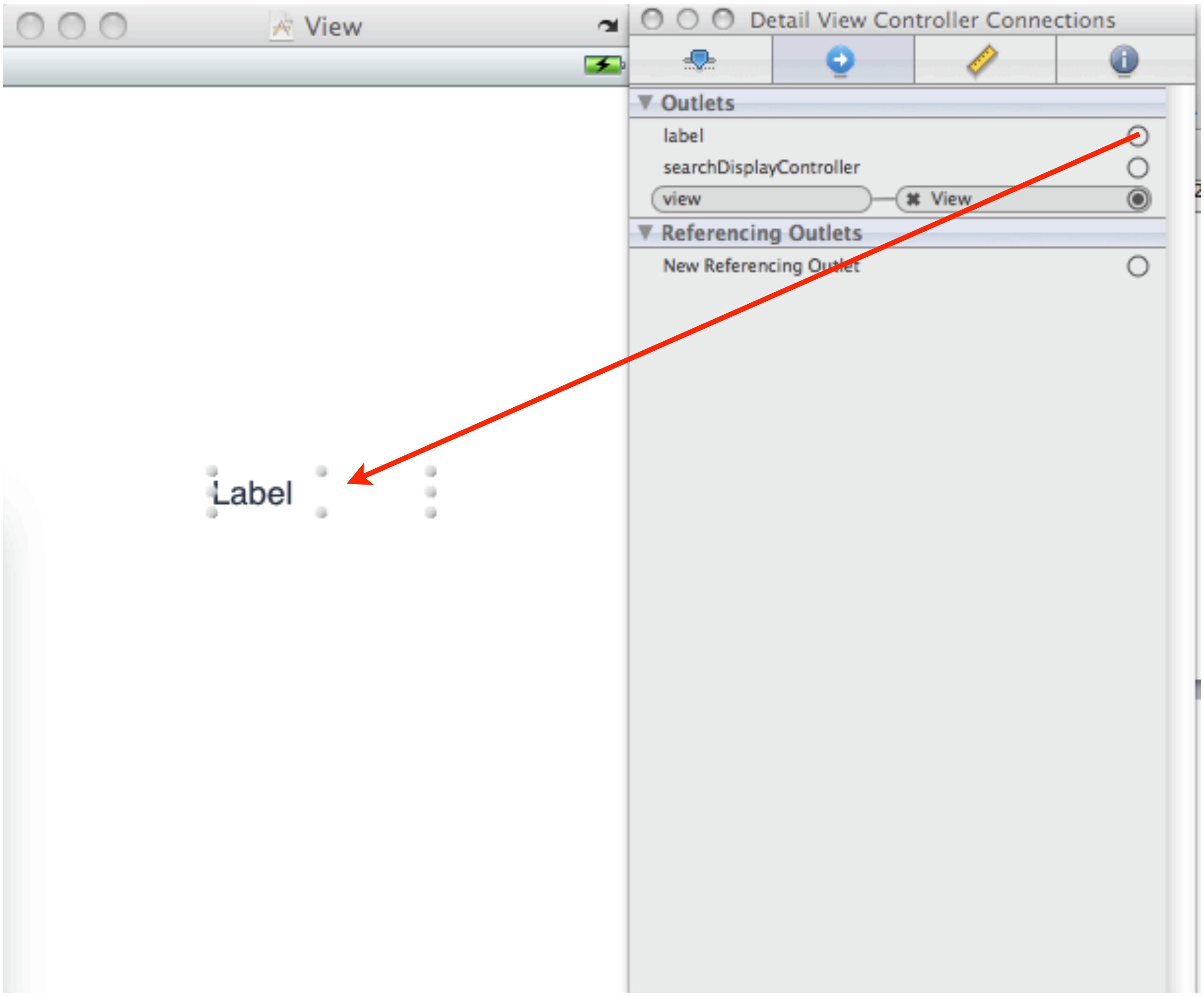
```
#import <UIKit/UIKit.h>

@interface DetailViewController : UIViewController {
    IBOutlet UILabel *label;
    NSString *sentName;
}
@property (retain) NSString *information;
@end
```

Step4. 開啓DetailViewController.xib,在View中加入一個label,並把label的frame加寬來符合之後可能不同長度字串的資訊



Step5. 在DetailViewController.xib主視窗點選File's Owner, 並在所顯示的Inspector裡選擇Connections這個tab,將裡面的label拉到View面剛剛新增的Label,將View和我們code裡面設定為IBOutlet的UILabel作連結,然後存檔.





Step6. 在DetailViewController.m對information這個NSString作synthesize,並將ViewDidLoad這個method的Mark去掉,把label的text在loading這個view時就顯示information的字串

```
@synthesize information;


// Implement viewDidLoad to do additional setup after loading the view,
typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    label.text = information;
}
```

Step7. 接著在RootViewController.h裡, 加入兩個我們顯示名稱和資料會使用到的Array

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

@interface RootViewController : UITableViewController {
    NSMutableArray* businessData;
    NSMutableArray* personalData;
}

@end
```

Step8. RootViewController.m裡, 我們先import DetailViewController.h,並將ViewDidLoad的Mark去掉,並將我們tableView的title和每個section裡面的row的顯示名稱命名.

```
#import "RootViewController.h"
#import "DetailViewController.h"


- (void)viewDidLoad {
    [super viewDidLoad];
    self.title = @"To Do List";
    businessData = [[NSMutableArray alloc] initWithObjects:@"Meeting",
@"Consulting", @"Make a call", nil];
    personalData = [[NSMutableArray alloc] initWithObjects:@"Party",
@"Shopping",nil];
    // Uncomment the following line to display an Edit button in the
navigation bar for this view controller.
    // self.navigationItem.rightBarButtonItem = self.editButtonItem;
}
```

Step9. 同樣在RootViewController.m裡, 把我們顯示的section和row所return的值依下列方式寫入.

```
// Customize the number of sections in the table view.
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return 2;
}


// Customize the number of rows in the table view.
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
(NSInteger)section {
    if(section == 0)
           return [businessData count];
    if(section == 1)
           return [personalData count];
    else
           return 0;
}
```

Step10. 同樣在RootViewController.m裡, 在//Configure the cell.下面加入對每個rowㄉㄚcell顯示的文字做設定,並加入一個DetailDisclosureButton.

```
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath {

    static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:CellIdentifier] autorelease];
    }

    // Configure the cell.
    if(indexPath.section == 0)
           cell.textLabel.text = [businessData
objectAtIndex:indexPath.row];
    else if(indexPath.section == 1)
           cell.textLabel.text = [personalData
objectAtIndex:indexPath.row];
    else
           cell.textLabel.text = nil;
    cell.accessoryType = UITableViewCellAccessoryDetailDisclosureButton;
    return cell;
}
```

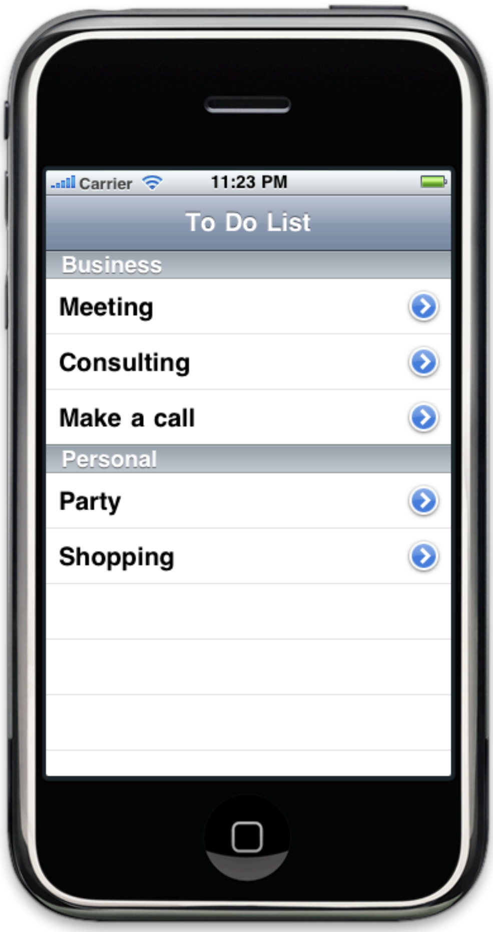
Step11. 接下來分別加入一個override method來設定兩個section的title

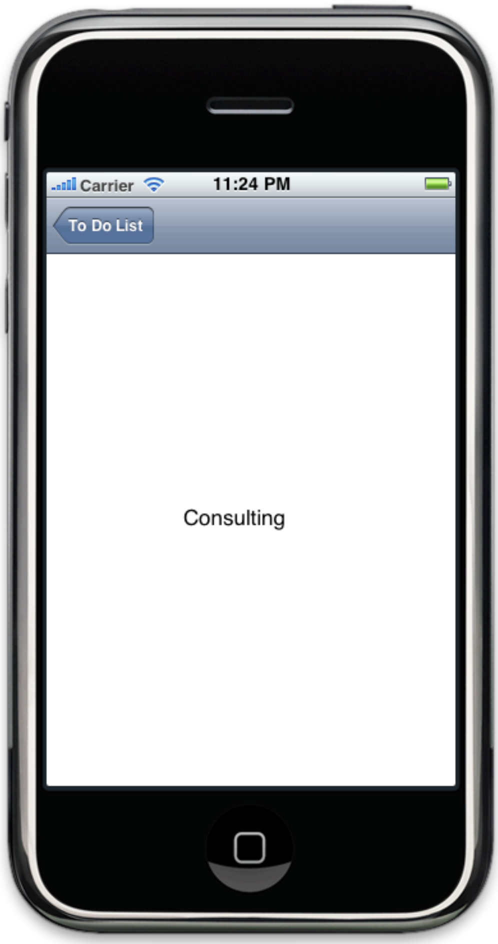
```
- (NSString *)tableView:(UITableView *)tableView
titleForHeaderInSection:(NSInteger)section
{
    if(section == 0)
          return @"Business";
    if(section == 1)
          return @"Personal";
    else
          return @"none";
}
```

Step12. 再加入一個override method使在點選每個row裡的DetailDisclosureButton之後,可以進入我們增加的View, 來顯示更多資訊,在此我們只將row裡的名稱放進顯示資訊的label裡.

```
- (void)tableView:(UITableView *)tableView
accessoryButtonTappedForRowWithIndexPath:(NSIndexPath *)indexPath {
    DetailViewController *detailViewController = [[DetailViewController
alloc] init];
    detailViewController.information = [tableView
cellForRowAtIndexPath:indexPath].textLabel.text;
    [self.navigationController pushViewController:detailViewController
animated:YES];
}
```

Step13. Build and Run


Carrier
11:23 PM
To Do List
Business
Meeting
Consulting
Make a call
Personal
Party
Shopping


Carrier
11:24 PM
To Do List
Consulting