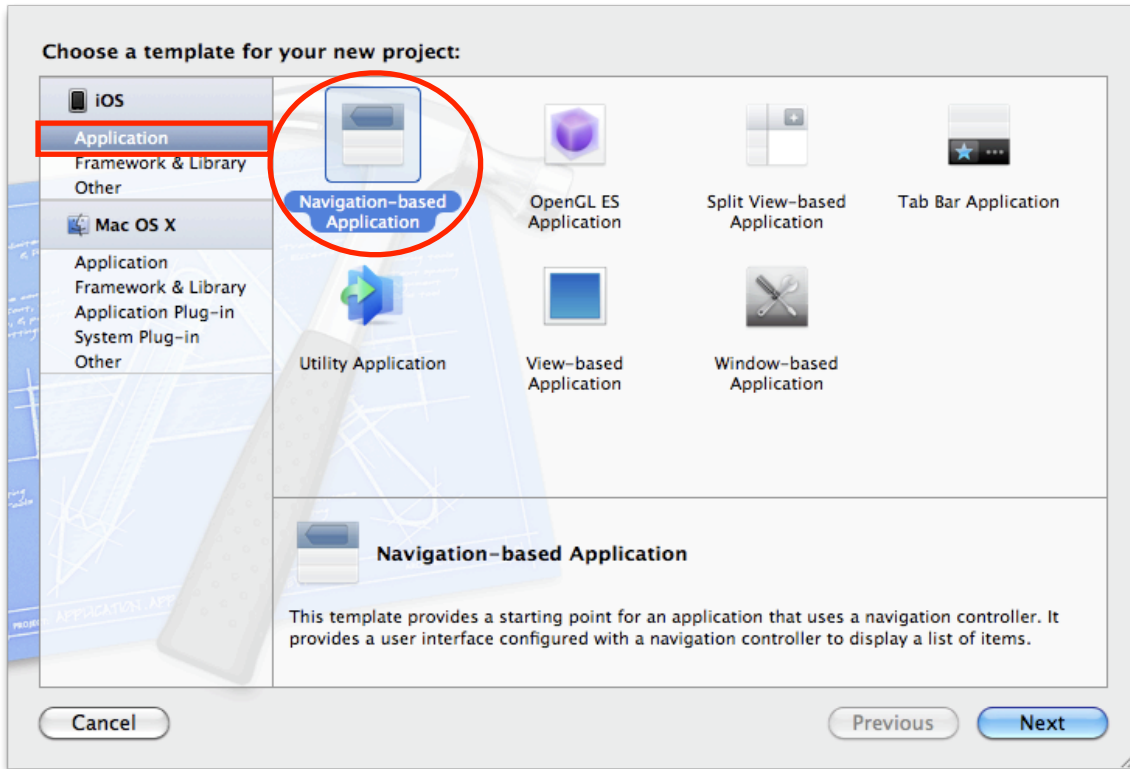
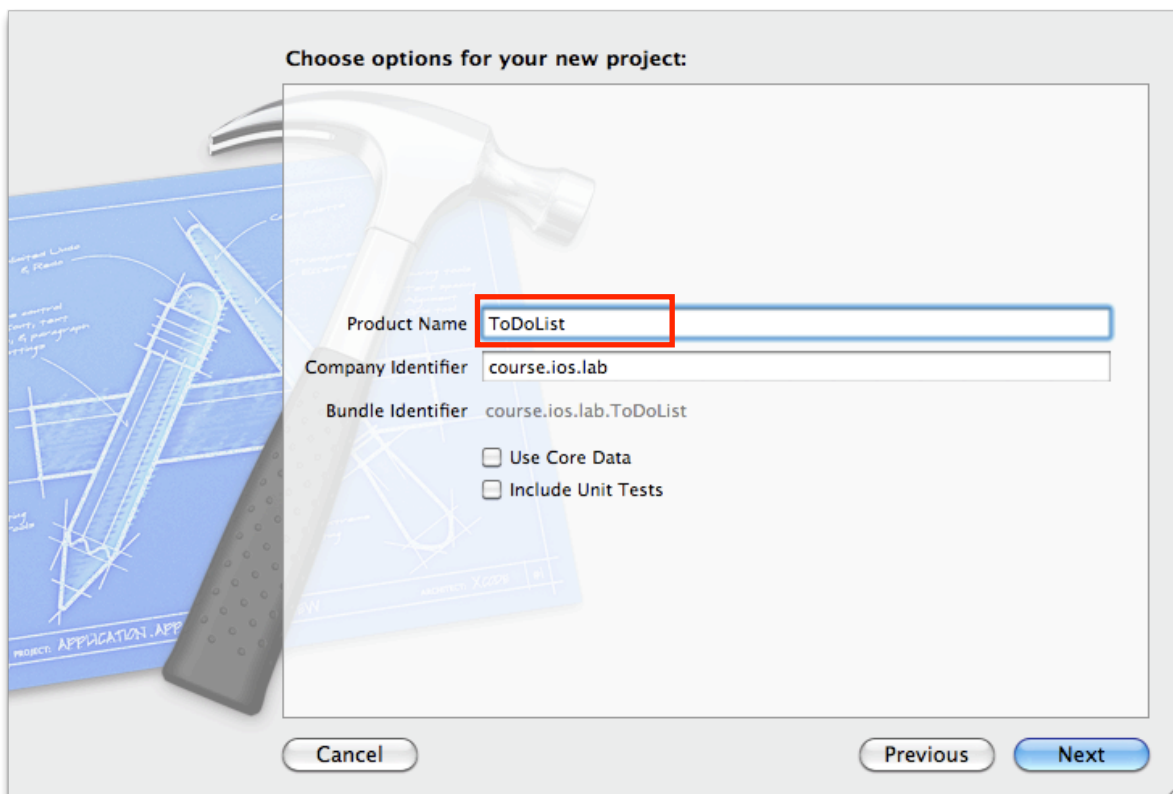


Lab ToDoList

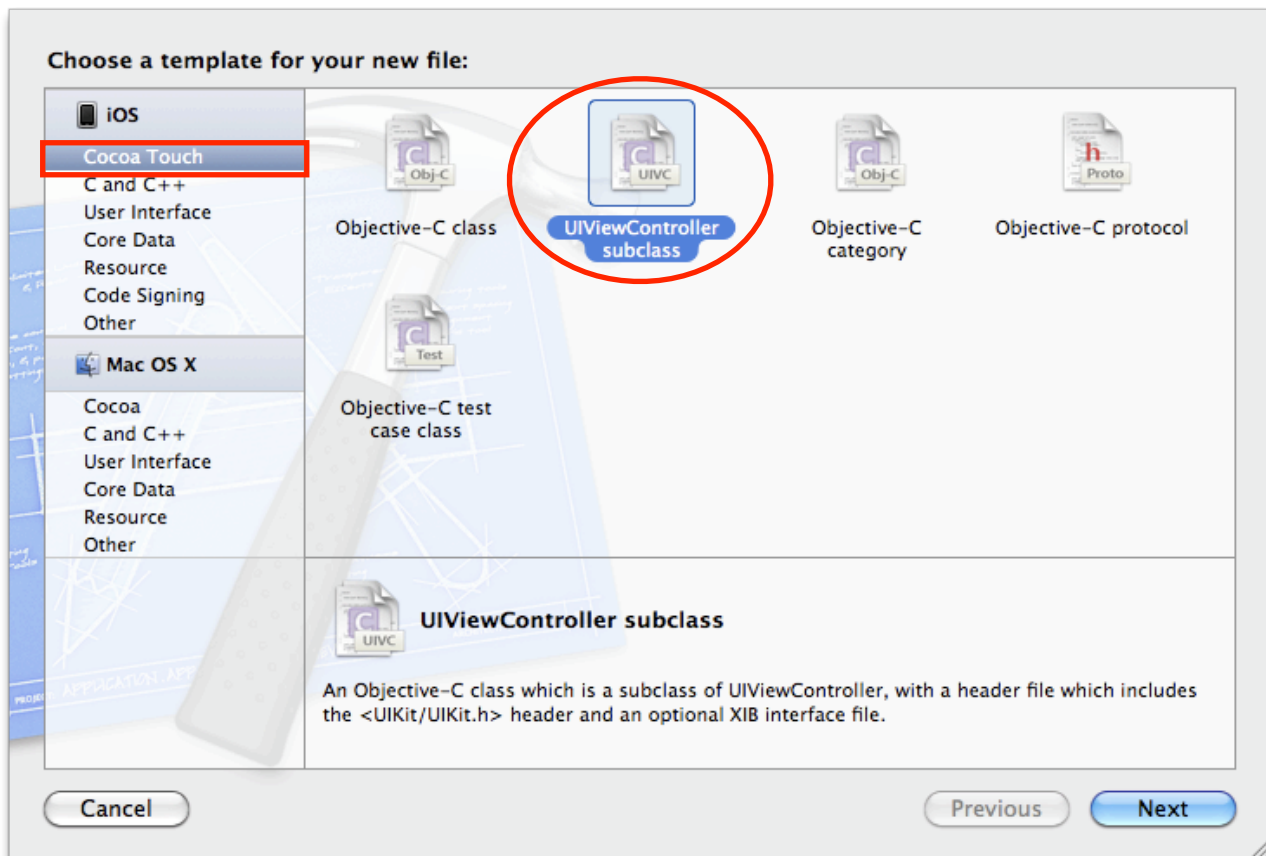
Step 1. 在File>New>New Project開啓一個新的project, 選擇 iOS 的 Application 目錄下的 Navigation-based application



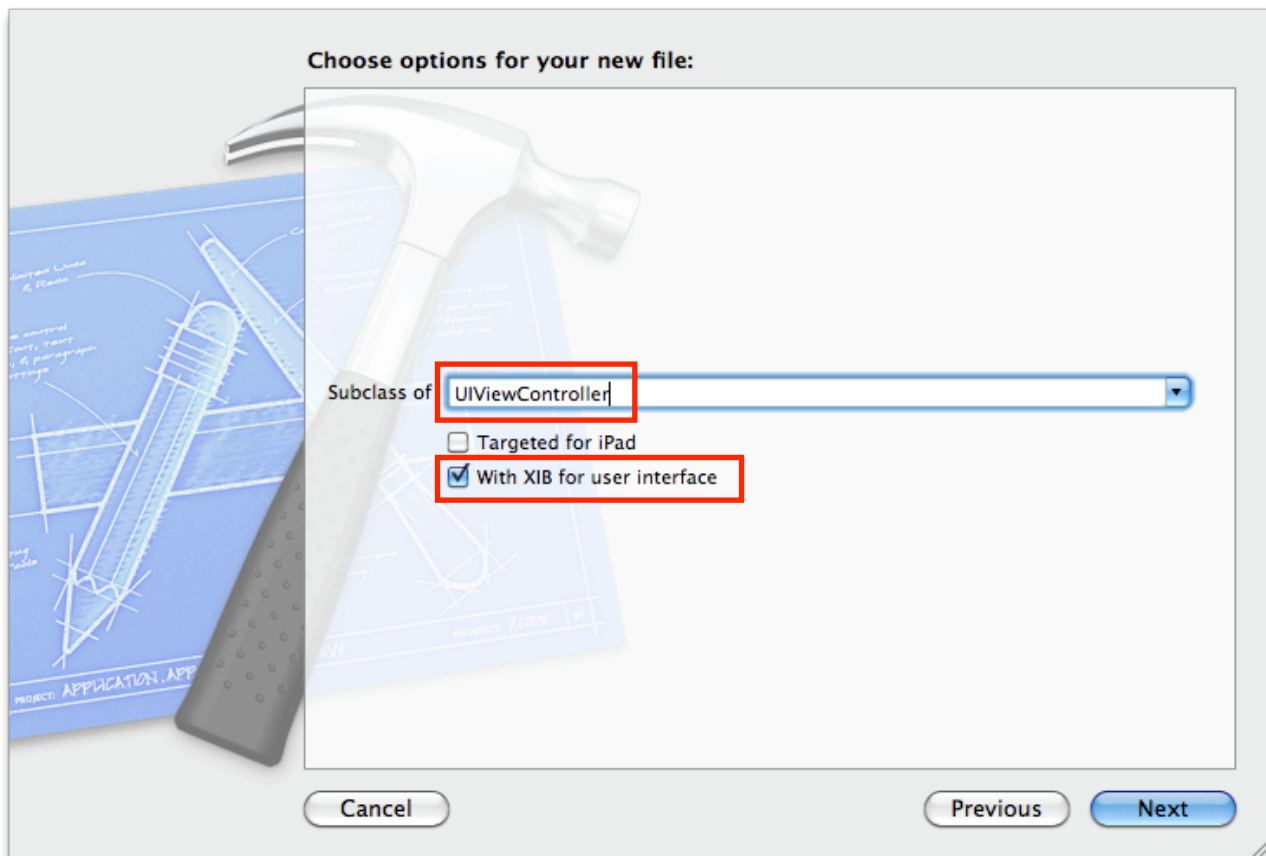
命名為 ToDoList



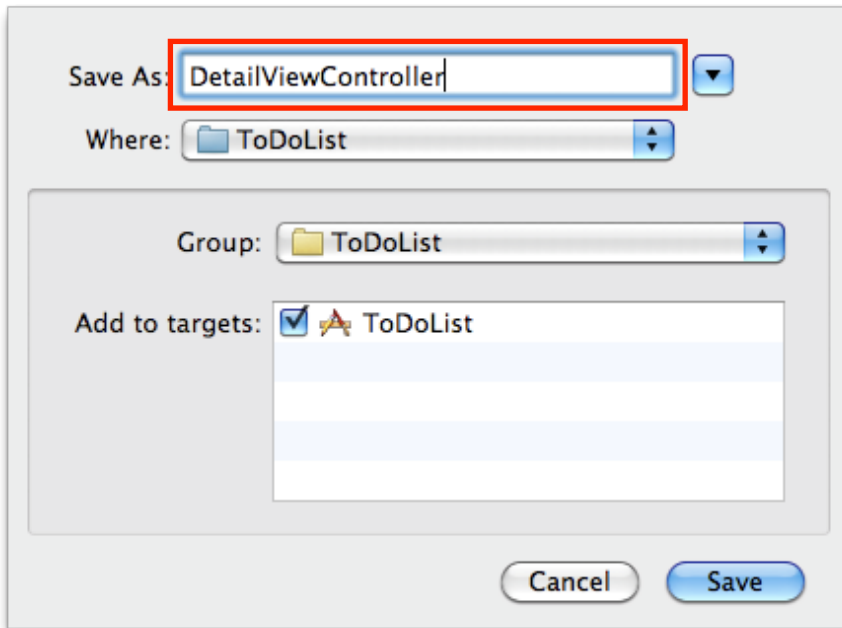
Step 2. 新增一個File,選擇 iOS 的 Cocoa Touch 目錄下的 UIViewController subclass



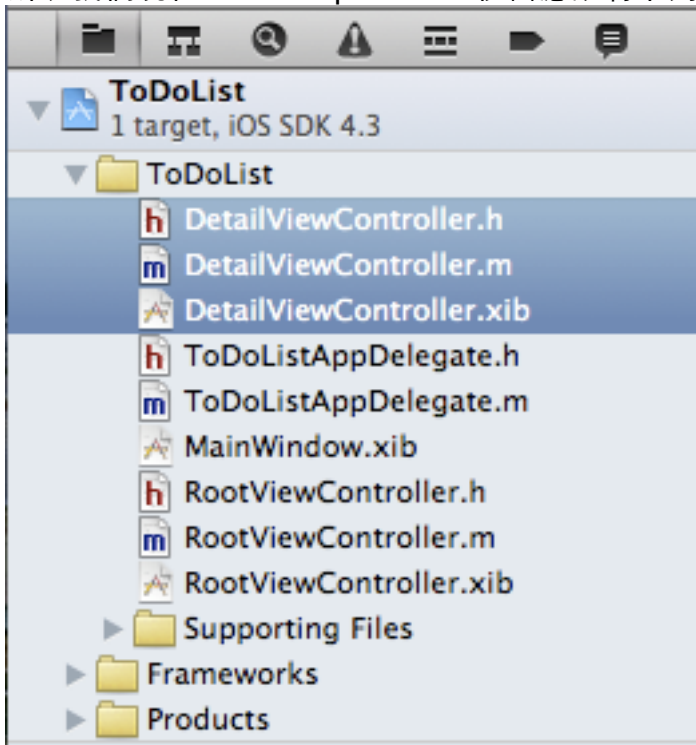
勾選 **With XIB for user interface** 來同時建立對應的xib檔



命名為DetailViewController



所以我們現在左邊Groups & Files視窗應該有下列檔案

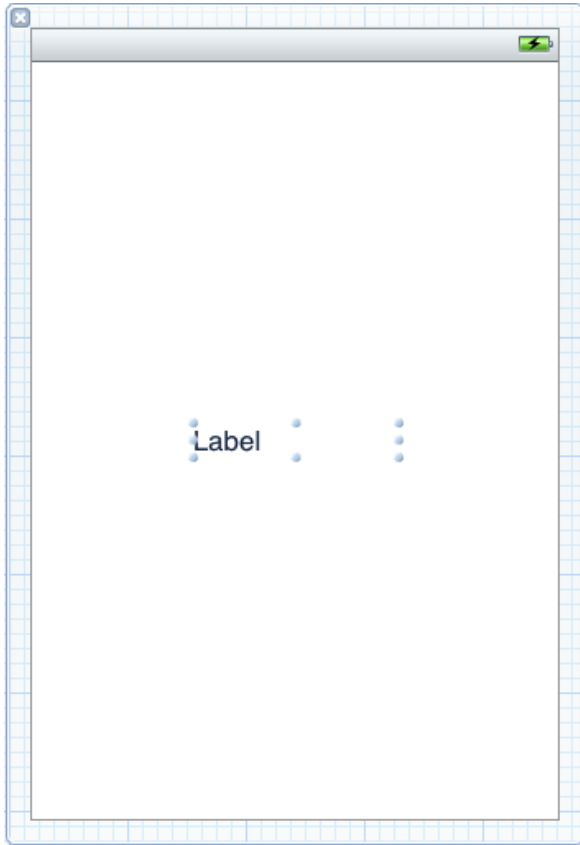


Step 3. 首先在剛剛新增的DetailViewController.h裡新增一個label和一個NSString來作為顯示detail資訊使用,並對這個NSString做@property來方便我們存取

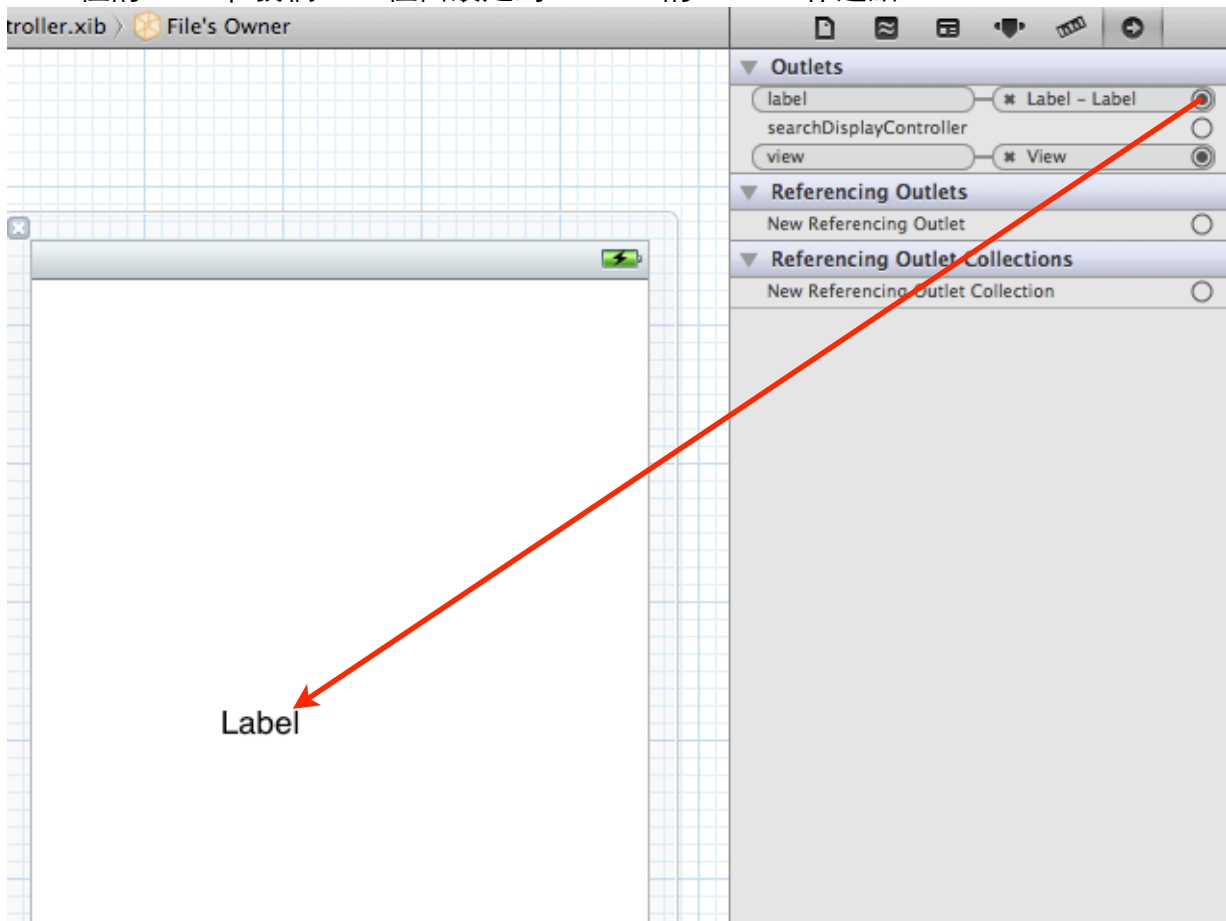
```
#import <UIKit/UIKit.h>

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

Step 4. 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



Step 5. 在DetailViewController.xib 裡點選 File's Owner, 開啓的Connections Inspector, 將View裡的Label和我們code裡面設定為IBOutlet的UILabel作連結



Step 6. 在DetailViewController.m 對 information 這個 NSString 作 synthesize,在 **ViewDidLoad** 這個method把label的text在loading這個view時就顯示information的字串

```
@synthesize information;
```

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view from its nib.
    label.text = self.information;
}
```

Step 7. 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
```

Step 8. 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];
}
```

Step 9. 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;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
(NSInteger)section
{
    if(section == 0)
        return [businessData count];
    if(section == 1)
        return [personalData count];
    else
        return 0;
}
```

Step 10. 同樣在 RootViewController.m 裡, 在 `// Configure the cell.` 下面加入對每個row的cell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;
}
```

Step 11. 接下來加入一個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";
}
```

Step 12. 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];
}
```

Step 13. Run (⌘+R)

