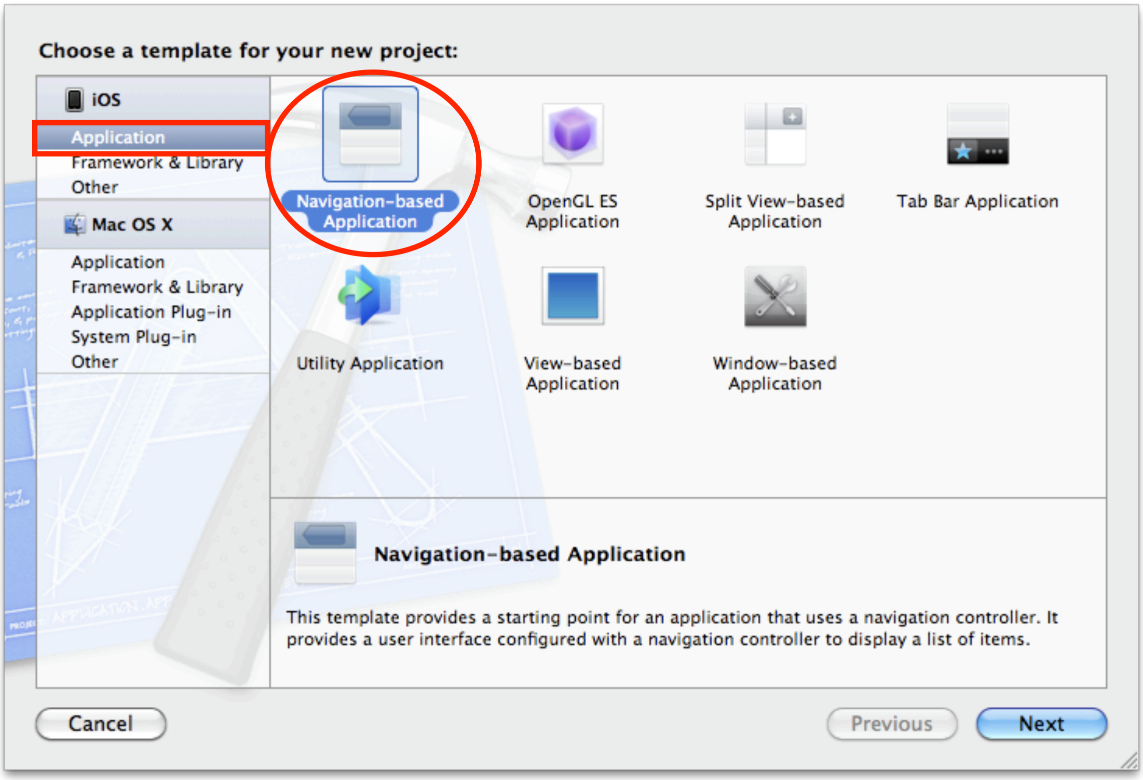
Lab SimpleSearch

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

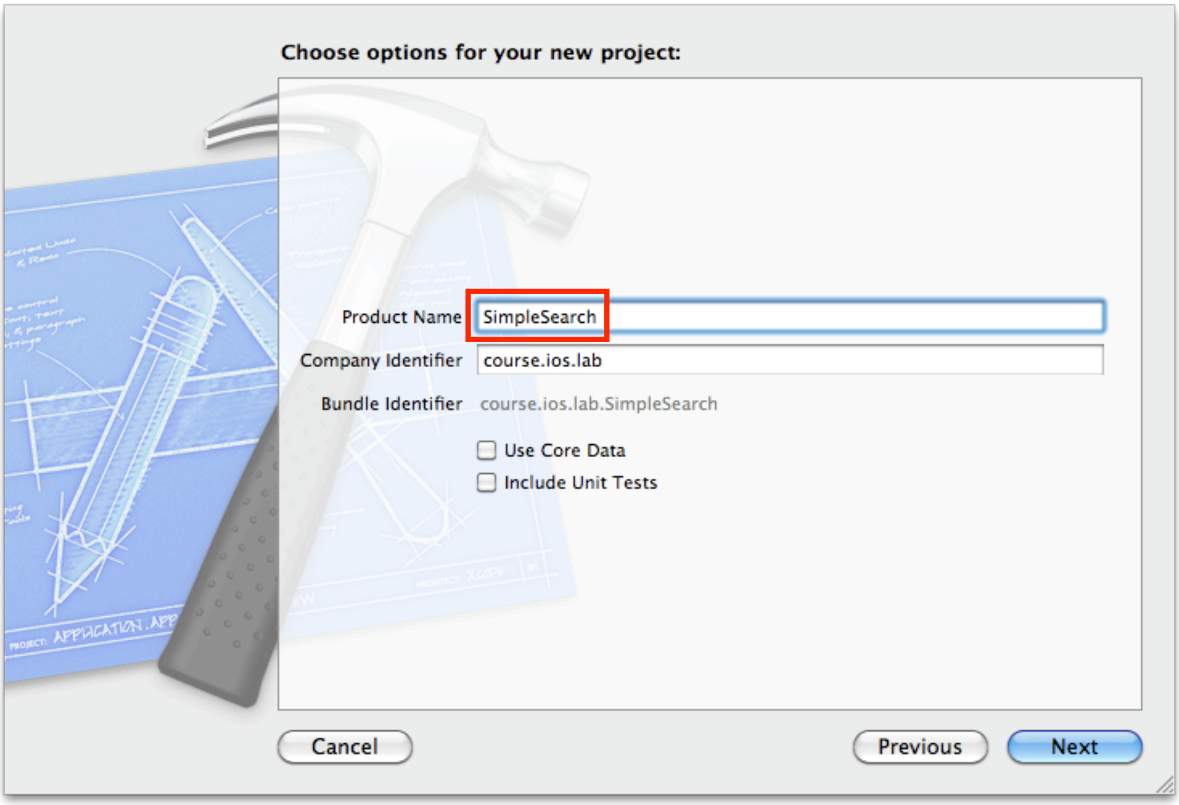


Choose a template for your new project:

Navigation-based Application

This template provides a starting point for an application that uses a navigation controller. It provides a user interface configured with a navigation controller to display a list of items.

命名為 SimpleSearch



Choose options for your new project:

Product Name: SimpleSearch

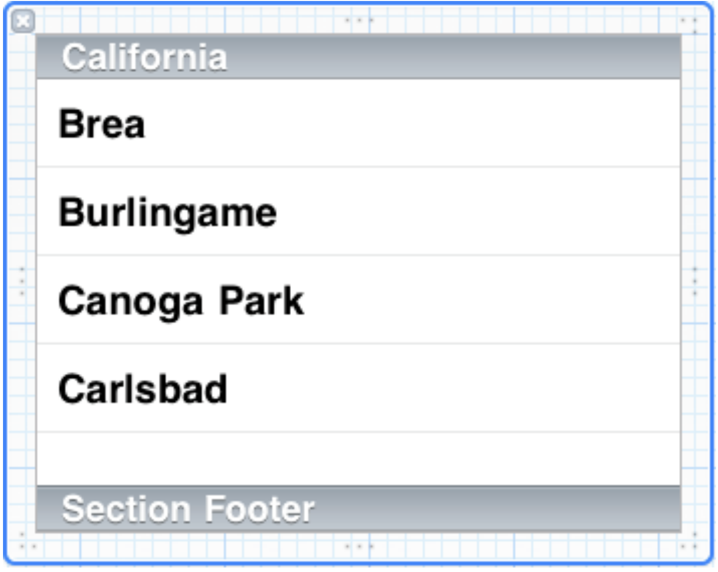
Company Identifier: course.ios.lab

Bundle Identifier: course.ios.lab.SimpleSearch

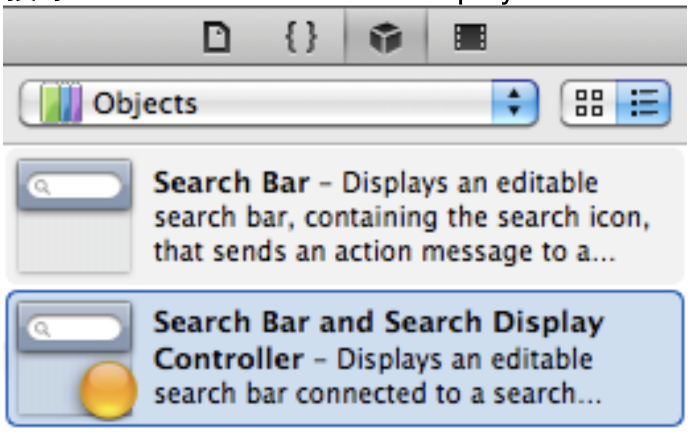
□ Use Core Data

□ Include Unit Tests

Step 2. 開啓RootViewController.xib,

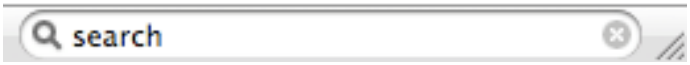


搜尋Search Bar and Search Display

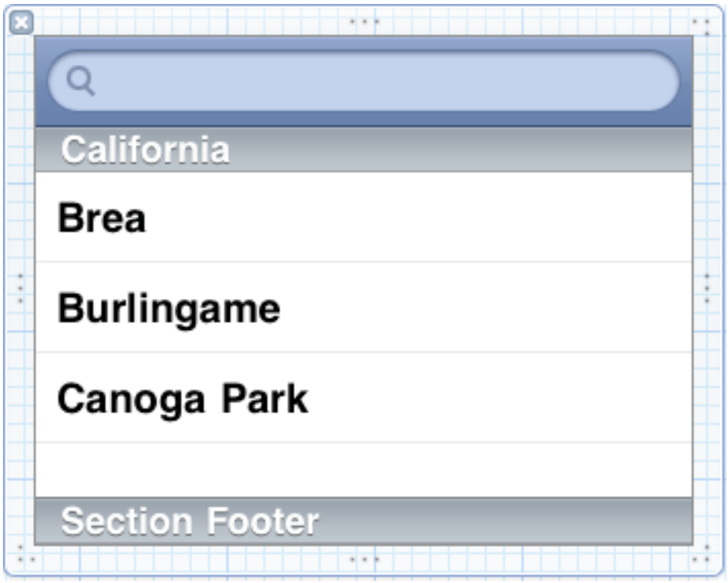


Search Bar – Displays an editable search bar, containing the search icon, that sends an action message to a...

Search Bar and Search Display Controller – Displays an editable search bar connected to a search...



然後加入到回來系統幫我們建好的TableView上面



Step 3. 開啓RootViewController.h, 在裡面加入以下程式,除了一個我們放置字串data的dataArray以外, 還有其他包括做search後找到的資料的fetchedArray, 照prefix分類的sections,以及其他的method.

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

@interface RootViewController : UITableViewController {
    NSArray * dataArray;
}

@property (retain) NSMutableArray * fetchedArray;
@property (retain) NSArray * sections;
- (void)filterContentForSearchText:(NSString*)searchText;
-(void) setUpSectionString;
-(NSInteger ) numberOfObjects:(NSArray *) objects inSection:(NSInteger)
section;
-(NSString *) titleForObjects:(NSArray *) data AtIndexPath:(NSIndexPath
*) indexPath;
@end
```

Step 3. 開啓RootViewController.m, 先synthesize fetchedArray和sections兩個Array, 並找到viewDidLoad() 初始化我們字串data的Array, 然後在 setUpSectionString() 再使用NSSet去設定依照Prefix排序的sections

```
#import "RootViewController.h"

@implementation RootViewController
@synthesize fetchedArray, sections;

- (void)viewDidLoad
{
     [super viewDidLoad];
     dataArray = [[NSArray alloc]
initWithObjects:@"one",@"two",@"three",@"four",@"five",@"six",@"seven",@
"eight",@"nine",@"ten",@"eleven",@"twelve",@"你好",@"你好嗎",@"我很好",
nil];
     self.fetchedArray = [NSMutableArray arrayWithCapacity:[dataArray
count]];
     [self setUpSectionString];
}

-(void) setUpSectionString {
    NSMutableSet * firstChars = [NSMutableSet set];
     [dataArray enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL
*stop) {
         [firstChars addObject:[obj substringToIndex:1]];
     }];
     self.sections = [NSArray array];
     self.sections = [[firstChars allObjects]
sortedArrayUsingSelector:@selector(compare:)];

}

- (NSComparisonResult)compare:(NSString *)otherObject {
     return [self compare:otherObject];
}
```

Step 4. 同樣在RootViewController.m裡, 找到 numberOfSectionsInTableView:(UITableView *)tableView 和 tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section 來回傳總共有幾個Sections並透過 numberOfObjects:(NSArray *) objects inSection:(NSInteger) section 將 dataArray 或是在search時將 fechedArray 裡的個數(Count)回傳給系統

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

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
(NSInteger)section
{
    if (tableView ==
self.searchDisplayController.searchResultsTableView)
    {
        return [self numberOfObjects:self.fetchedArray
inSection:section];
    }
    else
    {
        return [self numberOfObjects:dataArray inSection:section];
    }
}
```

```
-(NSInteger ) numberOfObjects:(NSArray *) objects inSection:(NSInteger)
section {
    __block NSInteger numbers = 0;
    [objects enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL
*stop) {
        if ([obj hasPrefix:[self.sections objectAtIndex:section]]) {
            numbers++;
        }
    }];
    return numbers;
}
```

Step 5. 同樣在 RootViewController.m 裡, 找到 `tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath` 裡 `// Configure the cell.` 下面加入把每個Cell的textLabel的text令為對應indexPath的dataArray或是在search mode時為fetchedArray

```
// Customize the appearance of table view cells.
- (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 (tableView ==
self.searchDisplayController.searchResultsTableView)
     {
        cell.textLabel.text = [self titleForObjects:self.fetchedArray
AtIndexPath:indexPath];
    }
     else
     {
        cell.textLabel.text= [self titleForObjects:dataArray
AtIndexPath:indexPath];
    }
    return cell;
}
```

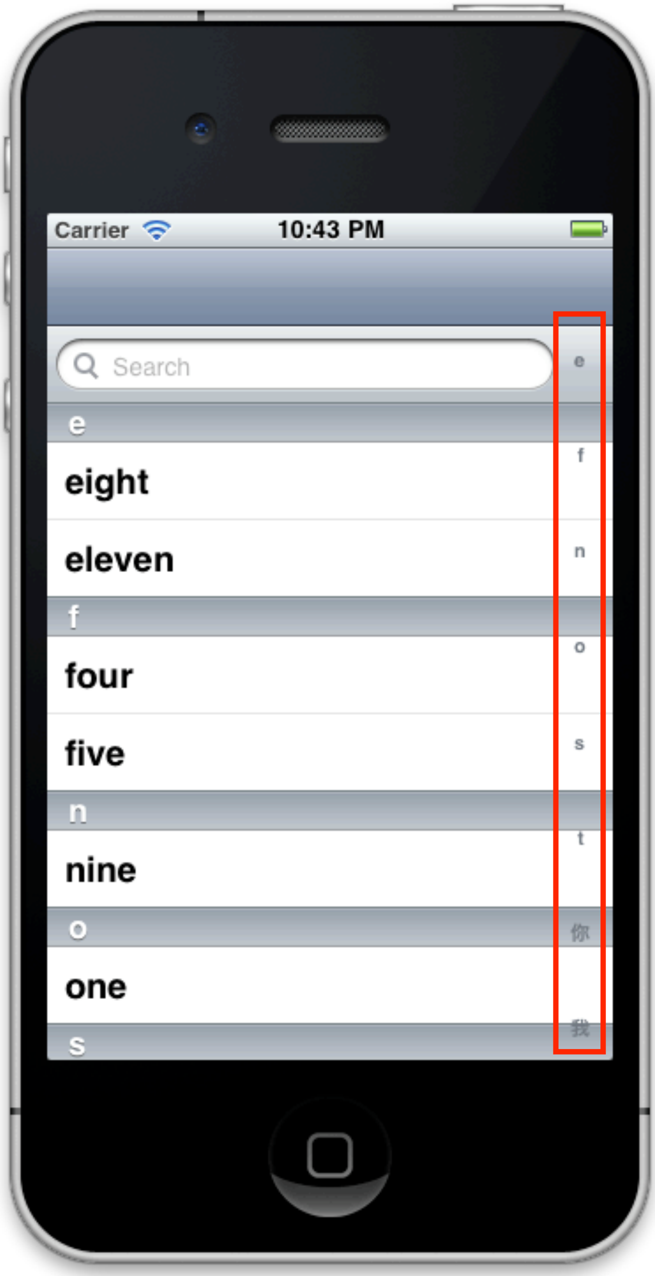
```
-(NSString *) titleForObjects:(NSArray *) data AtIndexPath:(NSIndexPath
*) indexPath {
    NSMutableArray * objects = [NSMutableArray array];
    [data enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL
*stop) {
        if ([obj hasPrefix:[self.sections
objectAtIndex:indexPath.section]]) {
            [objects addObject:obj];
        }
    }];
    return [objects objectAtIndex:indexPath.row];
}
```

Step 6. 同樣在 RootViewController.m 裡, 加入一個override method來設定各個section的title

```
- (NSString *)tableView:(UITableView *)tableView
titleForHeaderInSection:(NSInteger)section{
    return [self.sections objectAtIndex:section];
}
```

Step 7. 同樣在 RootViewController.m 裡, 加入一個override method來顯示如下面的效果去顯示sctionIndexTitles

```
- (NSArray *)sectionIndexTitlesForTableView:(UITableView *)tableView {
    return self.sections;
}
```

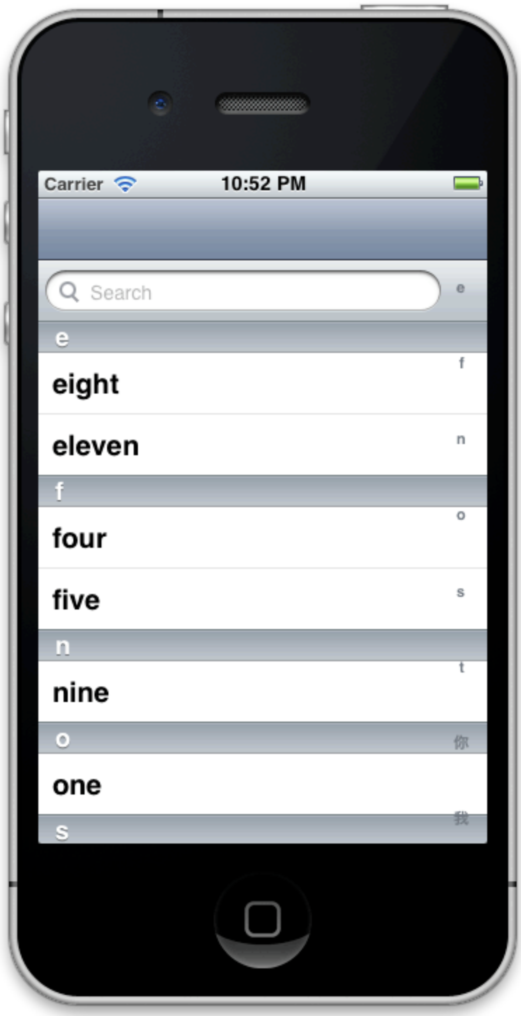


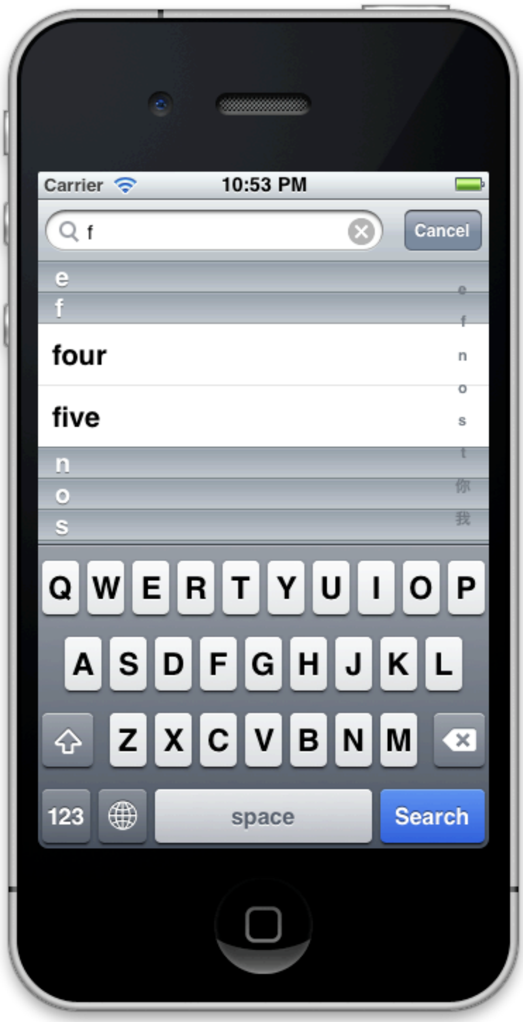
Step 8. 最後 同樣在 RootViewController.m 裡, 加入 searchDisplayController:(UISearchDisplayController *)controller shouldReloadTableForSearchString:(NSString *)searchString 去完成search後顯示搜尋到的結果,注意一定要 return **YES** 才會去reload search result的 TableView

```
- (BOOL)searchDisplayController:(UISearchDisplayController *)controller
shouldReloadTableForSearchString:(NSString *)searchString
{
    [self filterContentForSearchText:searchString];
    
    // Return YES to cause the search result table view to be reloaded.
    return YES;
}
```

```
- (void)filterContentForSearchText:(NSString*)searchText{
    [self.fetchedArray removeAllObjects];
    [dataArray enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL
*stop) {
        if ([obj hasPrefix:searchText]) {
            [self.fetchedArray addObject:obj];
        }
    }];
}
```

Step 9. Run (⌘+R)


Carrier
10:52 PM
Search
e
eight
eleven
f
four
five
n
nine
o
one
s
e
f
n
o
s
t
你
我


Carrier
10:53 PM
f
Cancel
e
f
four
five
n
o
s
e
f
n
o
s
t
你
我
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
Search