

1204315 - Wireless Mobile Application Programming

Manasawee Kaenampornpan

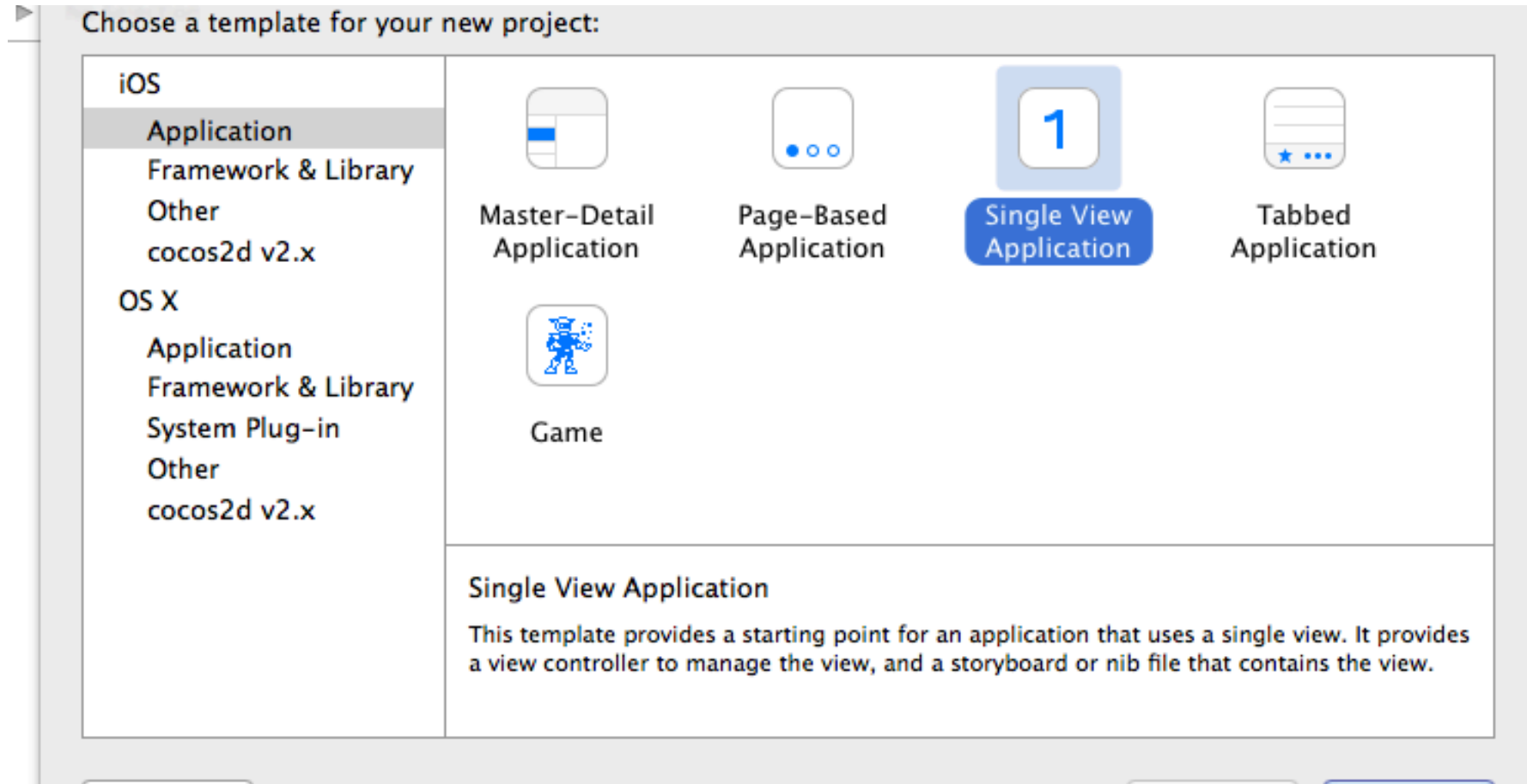
มนัสวี แก่นอำพรพันธ์

manasaweek@gmail.com

Maharakham University

Core Data

Create single view project



Check Use Core Data

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

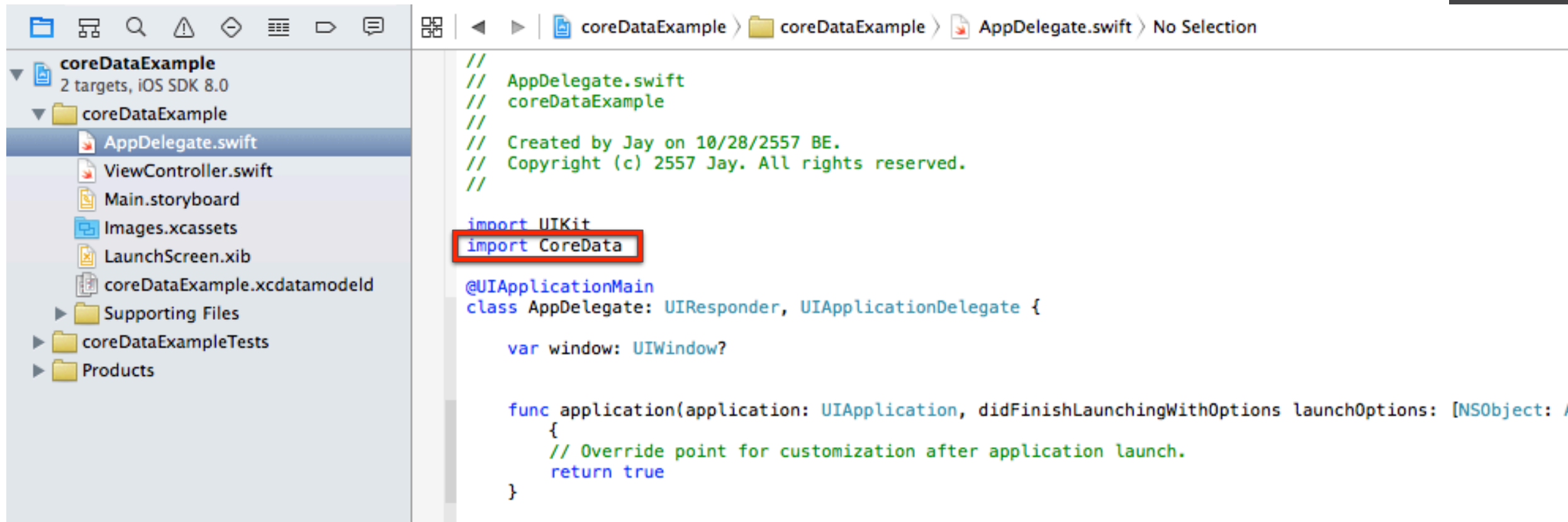
Bundle Identifier:

Language:

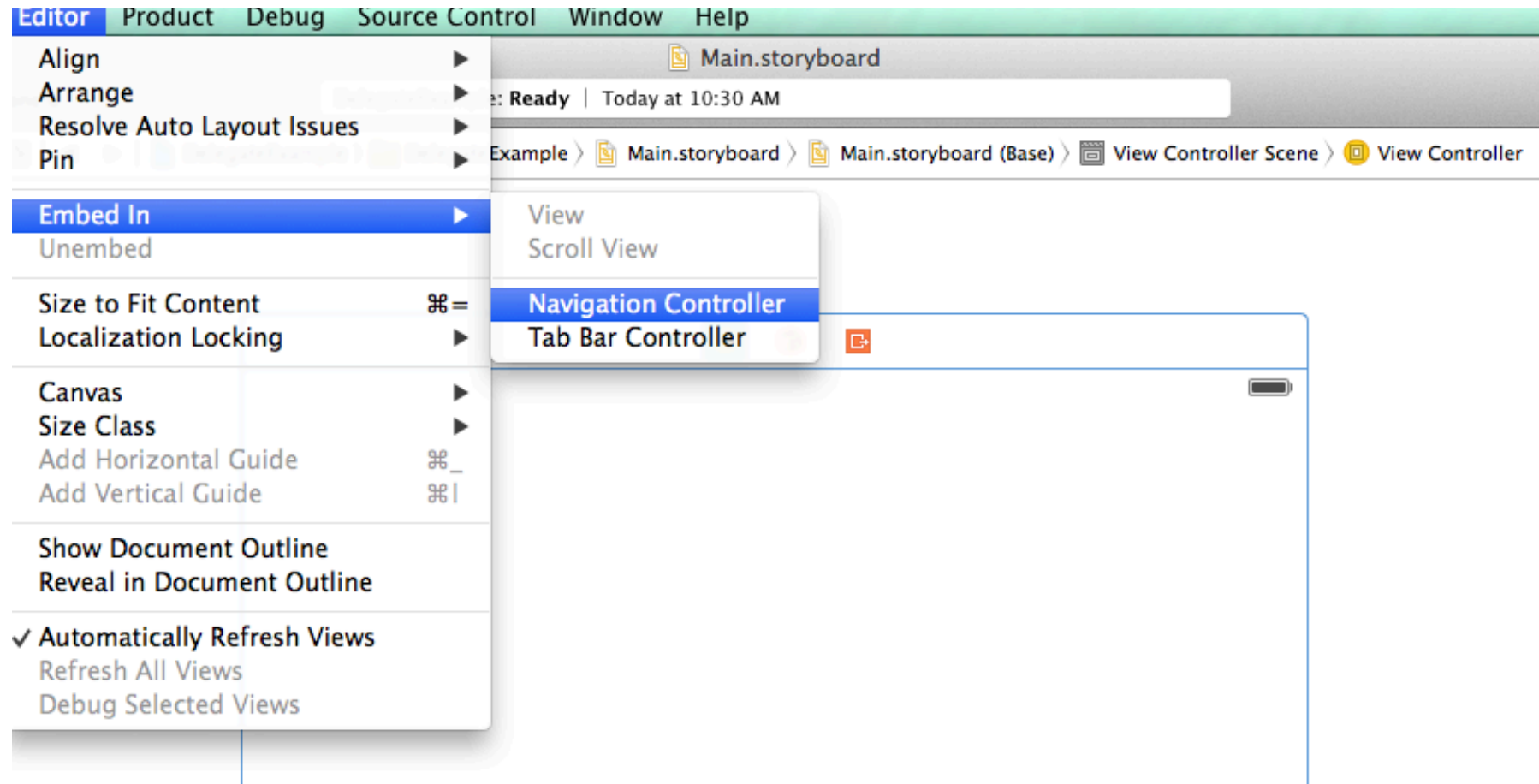
Devices:

☒ Use Core Data

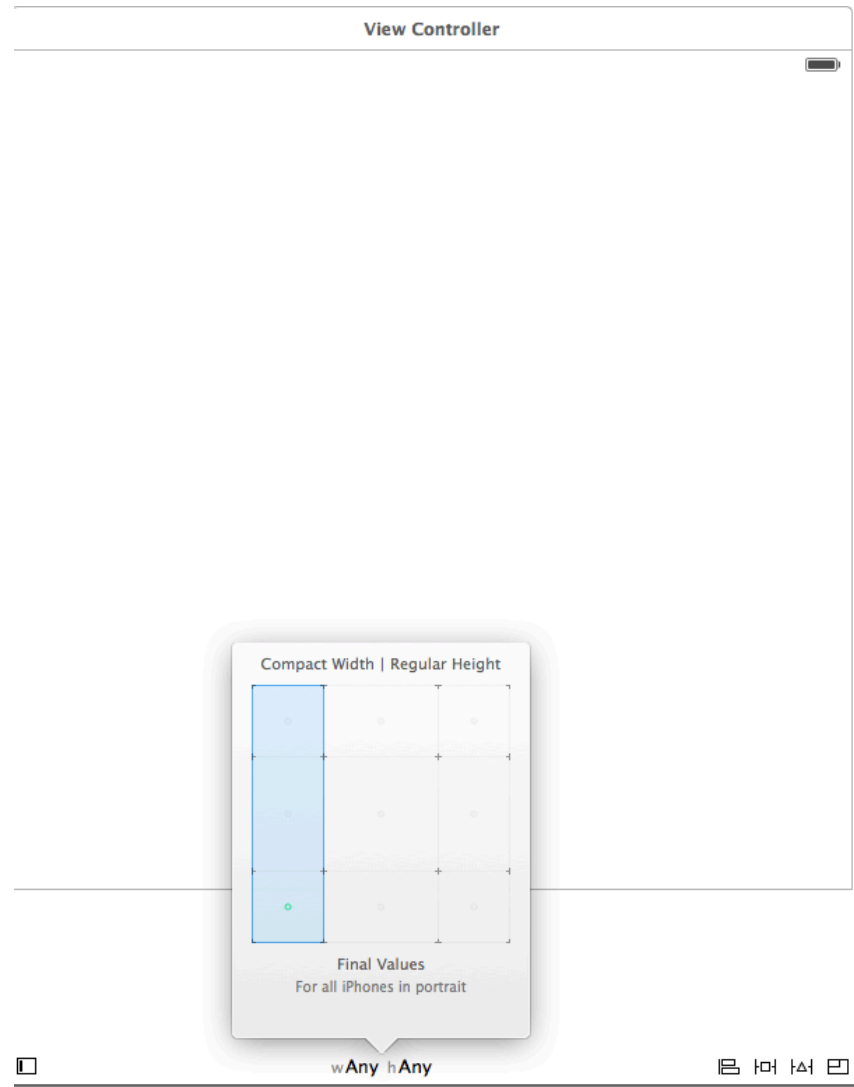
ใน AppDelegate.swift จะมี Core Data



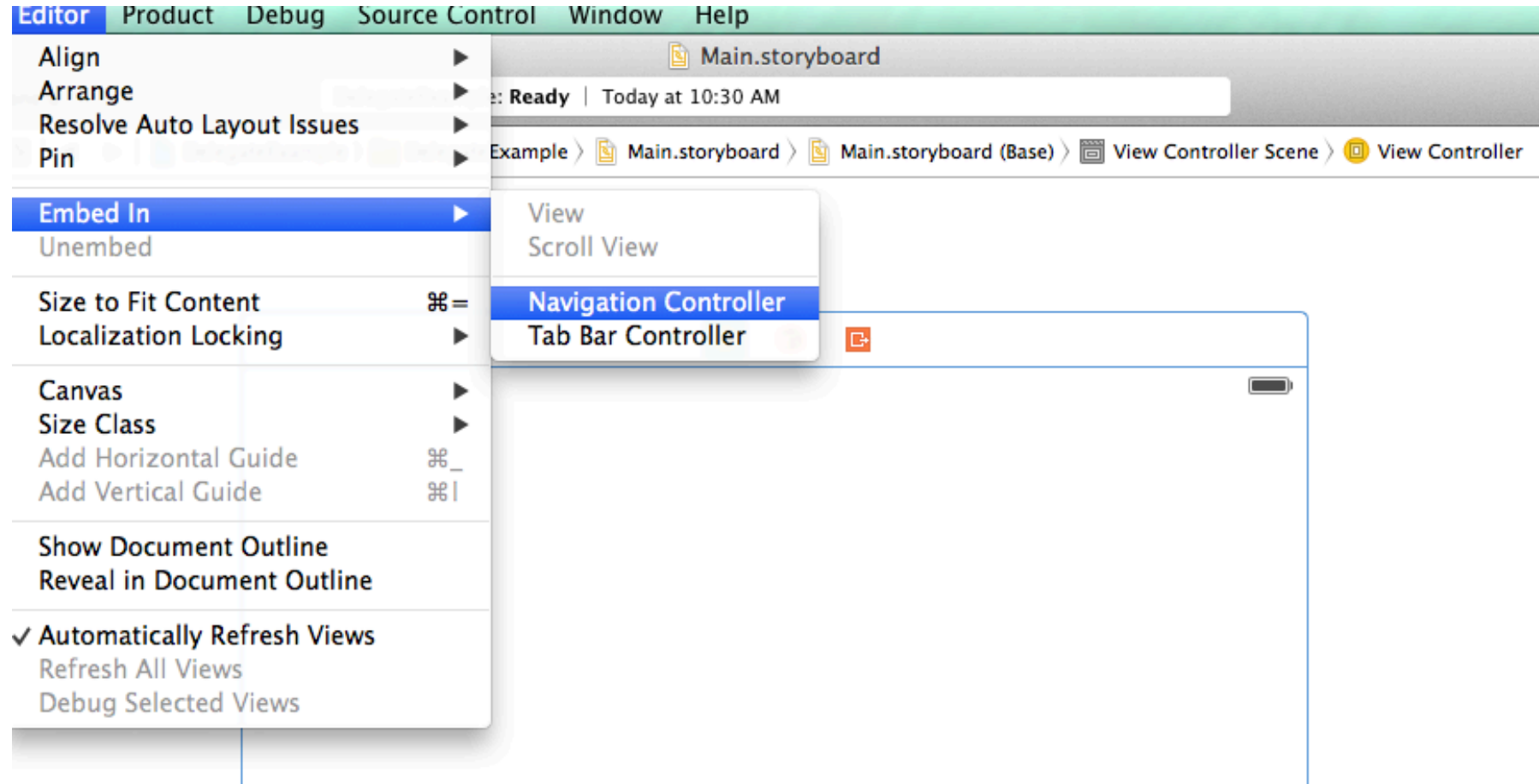
สร้าง app เพื่อเก็บรายการชื่อของ



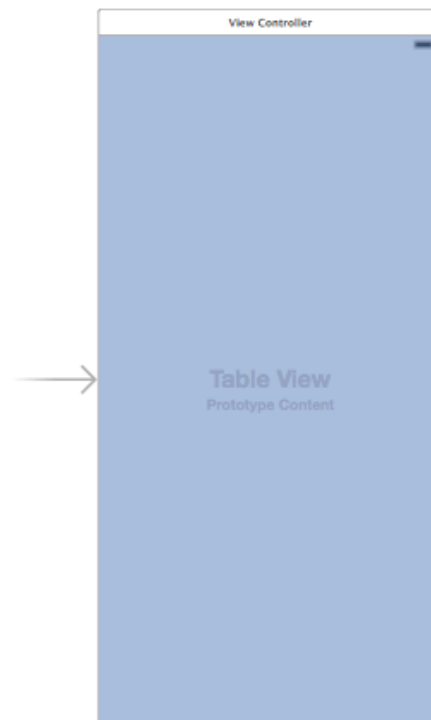
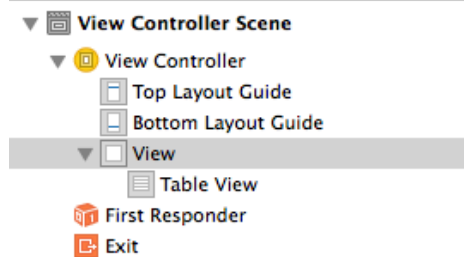
ตั้งค่า view Controller size: Regular Height และ Compact Width



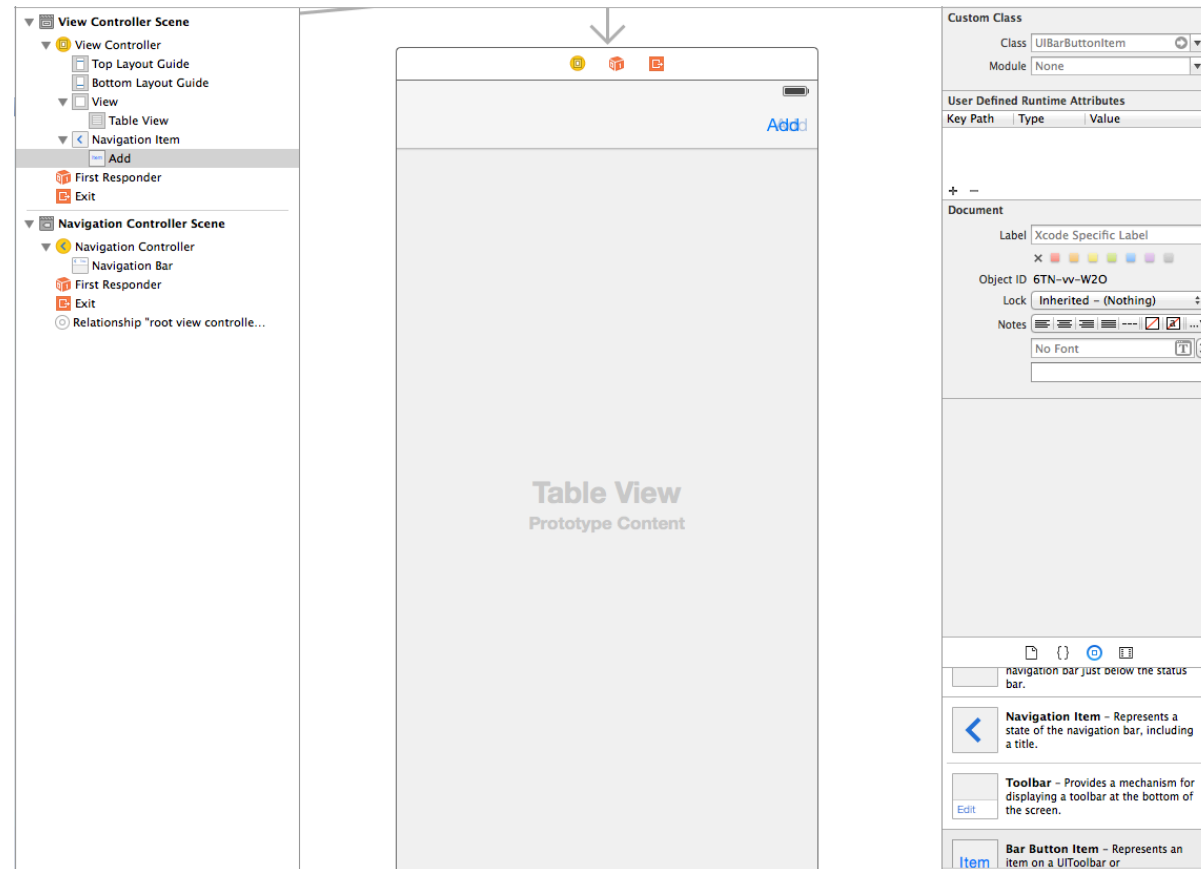
Embedded Navigation controller



เพิ่ม Table View



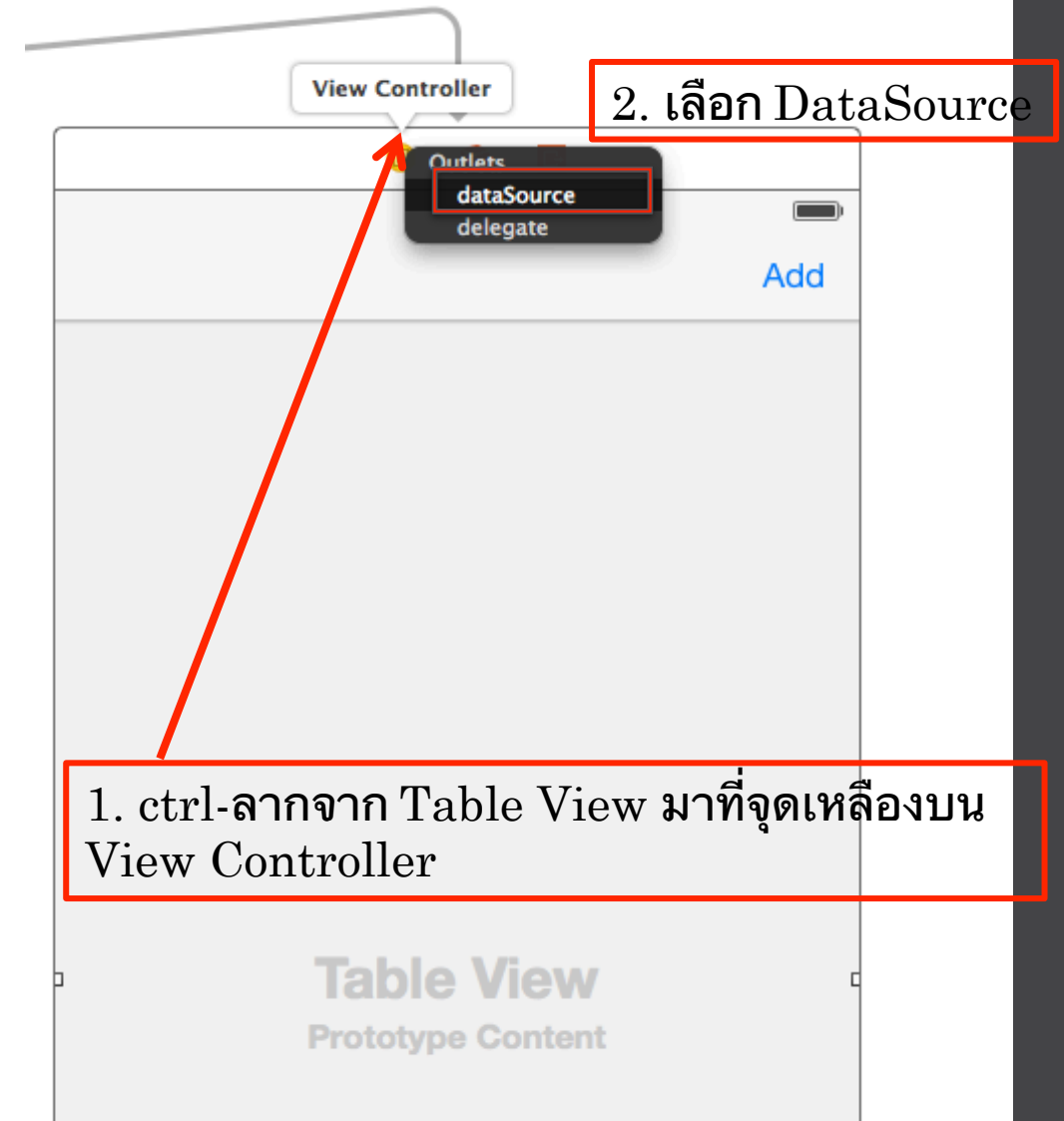
เพิ่ม Bar button Item -> Add



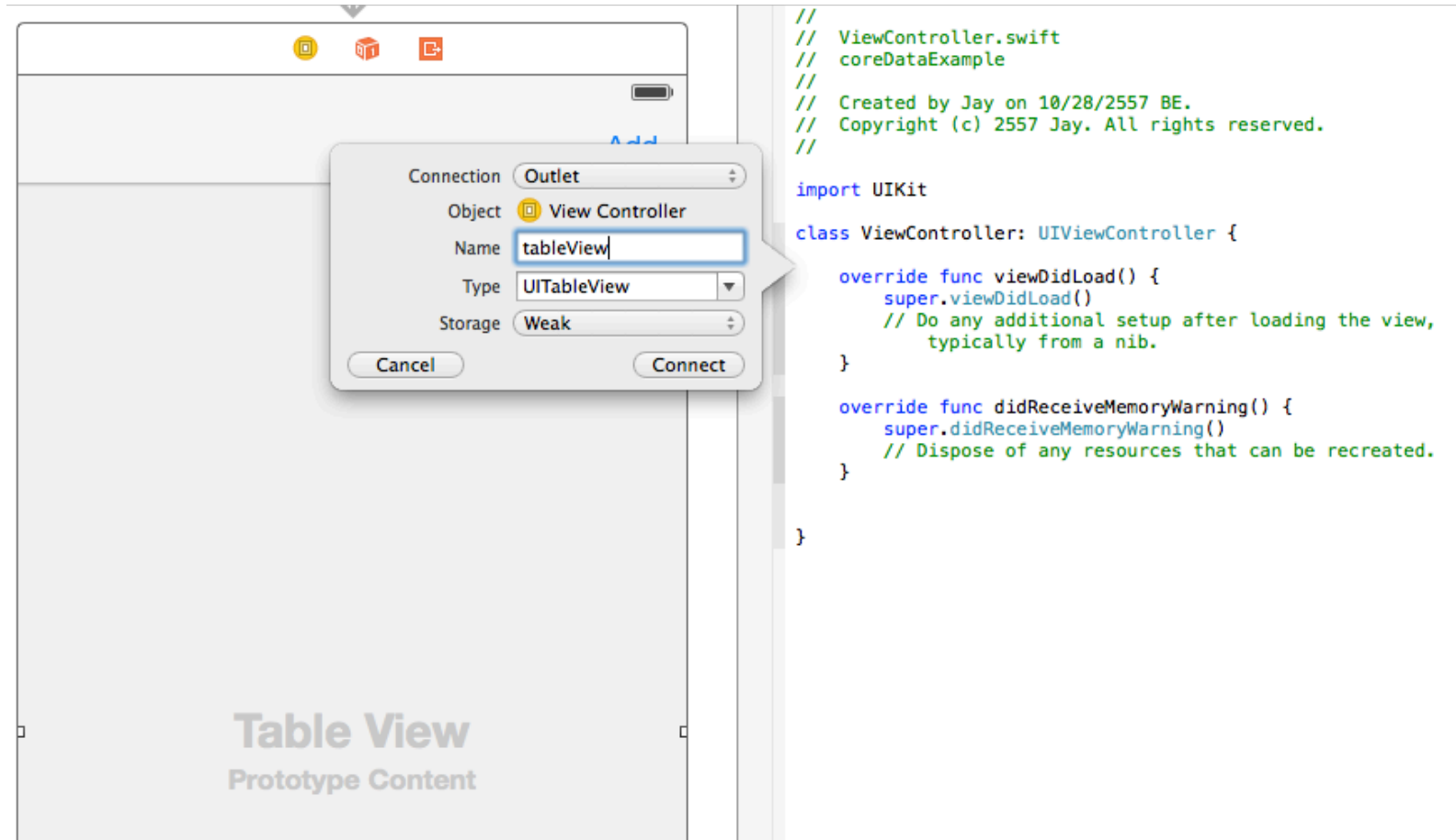
ตั้ง Table View ให้เป็น DataSource ของ View Controller

1. ctrl-ลากจาก Table View มาที่จุดเหลืองบน View Controller
2. เลือก DataSource

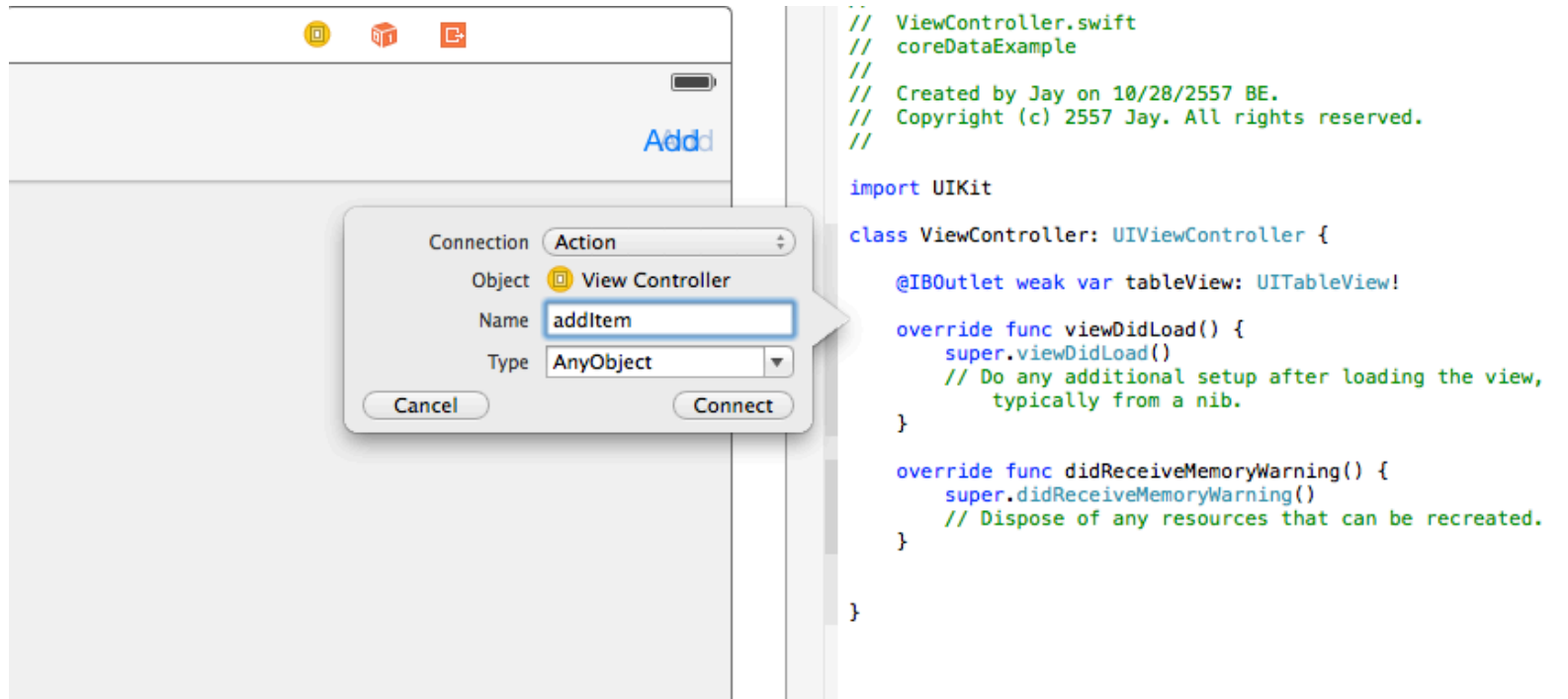
Note: ที่ไม่ได้เลือก delegate เพราะเวลากดที่ cell มันต้องเกิดอะไรขึ้นนั่นเอง



Connect TableView to viewController



Connect Button to viewController



เพิ่มตัวแปร `Array String` เพื่อเก็บรายการ
ของที่จะนำมาแสดงใน `TableView`

```
var items = [String]()
```

เพิ่มการลงทะเบียนของการแสดงข้อมูลใน cell
ของ TableView ลงใน ViewDidLoad

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    title = "\"Shopping List\""  
    tableView.registerClass(UITableViewCell.self,  
        forCellReuseIdentifier: "Cell")  
}
```

เพิ่ม protocol UITableViewDataSource

```
import UIKit

class ViewController: UIViewController,
    UITableViewDataSource {

    var items = [String]()

    @IBOutlet weak var tableView: UITableView!

    @IBAction func addItem(sender: AnyObject) {

    }
```


เพิ่ม func ของ UITableViewDataSource

```
// MARK: UITableViewDataSource
func tableView(tableView: UITableView,
    numberOfRowsInSection section: Int) -> Int {
    return items.count
}

func tableView(tableView: UITableView,
    cellForRowAtIndexPath
    indexPath: NSIndexPath) -> UITableViewCell {

    let cell =

        tableView.dequeueReusableCellWithIdentifier(
            "Cell")
        as UITableViewCell

    cell.textLabel!.text = items[indexPath.row]

    return cell
}
```

เพิ่มโค้ดการทำงานของ AddItem func

```
@IBAction func addItem(sender: AnyObject) {  
    var alert = UIAlertController(title: "New item",  
        message: "Add a new item",  
        preferredStyle: .Alert)  
  
    let saveAction = UIAlertAction(title: "Save",  
        style: .Default) { (action: UIAlertAction!) -> Void in  
        let textField = alert.textFields![0] as UITextField  
        self.items.append(textField.text)  
        self.tableView.reloadData()  
    }  
  
    let cancelAction = UIAlertAction(title: "Cancel",  
        style: .Default) { (action: UIAlertAction!) -> Void in  
    }  
  
    alert.addTextFieldWithConfigurationHandler {  
        (textField: UITextField!) -> Void in  
    }  
  
    alert.addAction(saveAction)  
    alert.addAction(cancelAction)  
  
    presentViewController(alert,  
        animated: true,  
        completion: nil)  
}
```

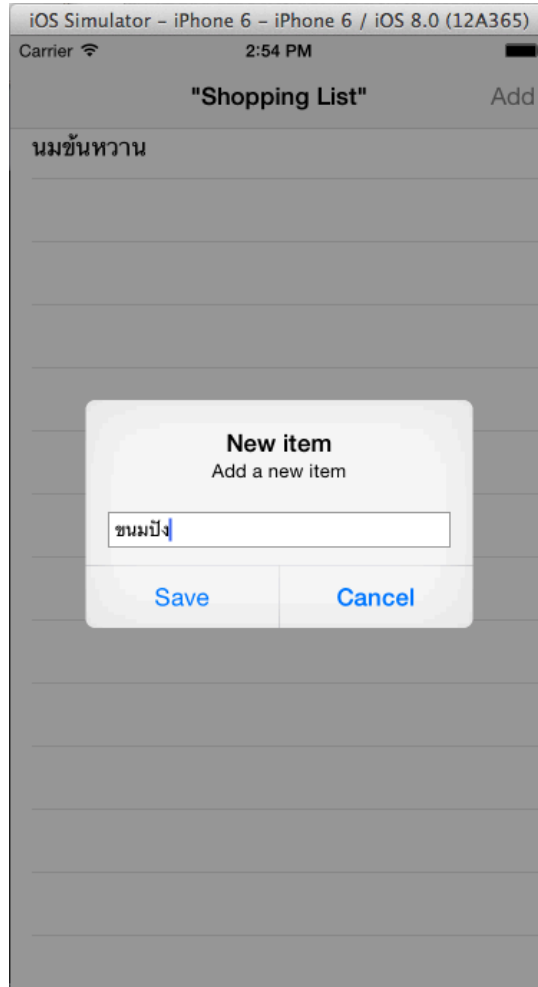
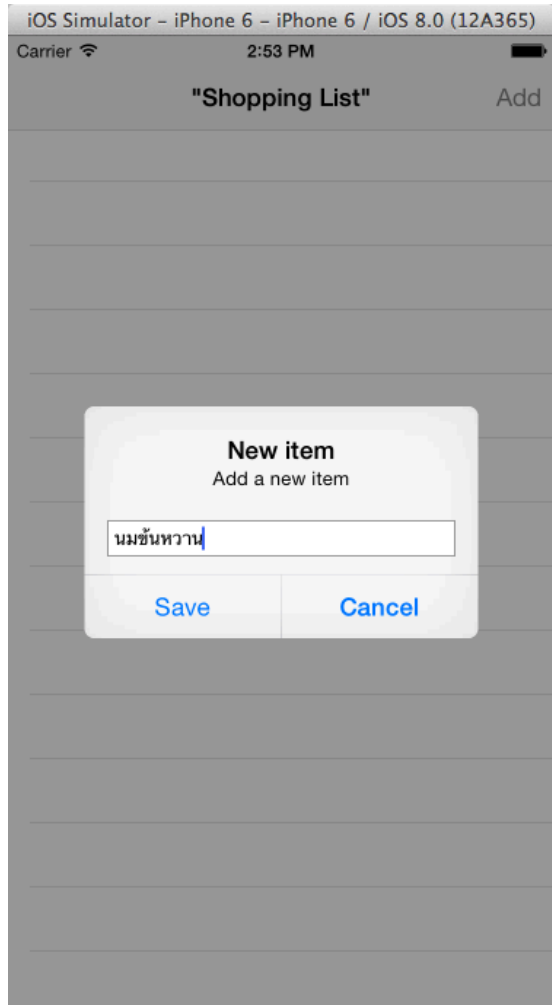
1. UIAlertController จะมี textField และปุ่ม Save, Cancel มันจะ popup ทุกครั้งที่กดปุ่ม Add

2. กดปุ่ม Save ก็จะเอาค่าใน textfield มาใส่ใน Array items

3. กดปุ่ม cancel ก็จะไม่เกิดอะไรขึ้น

4. เพิ่ม Action กับปุ่ม

Build and Run



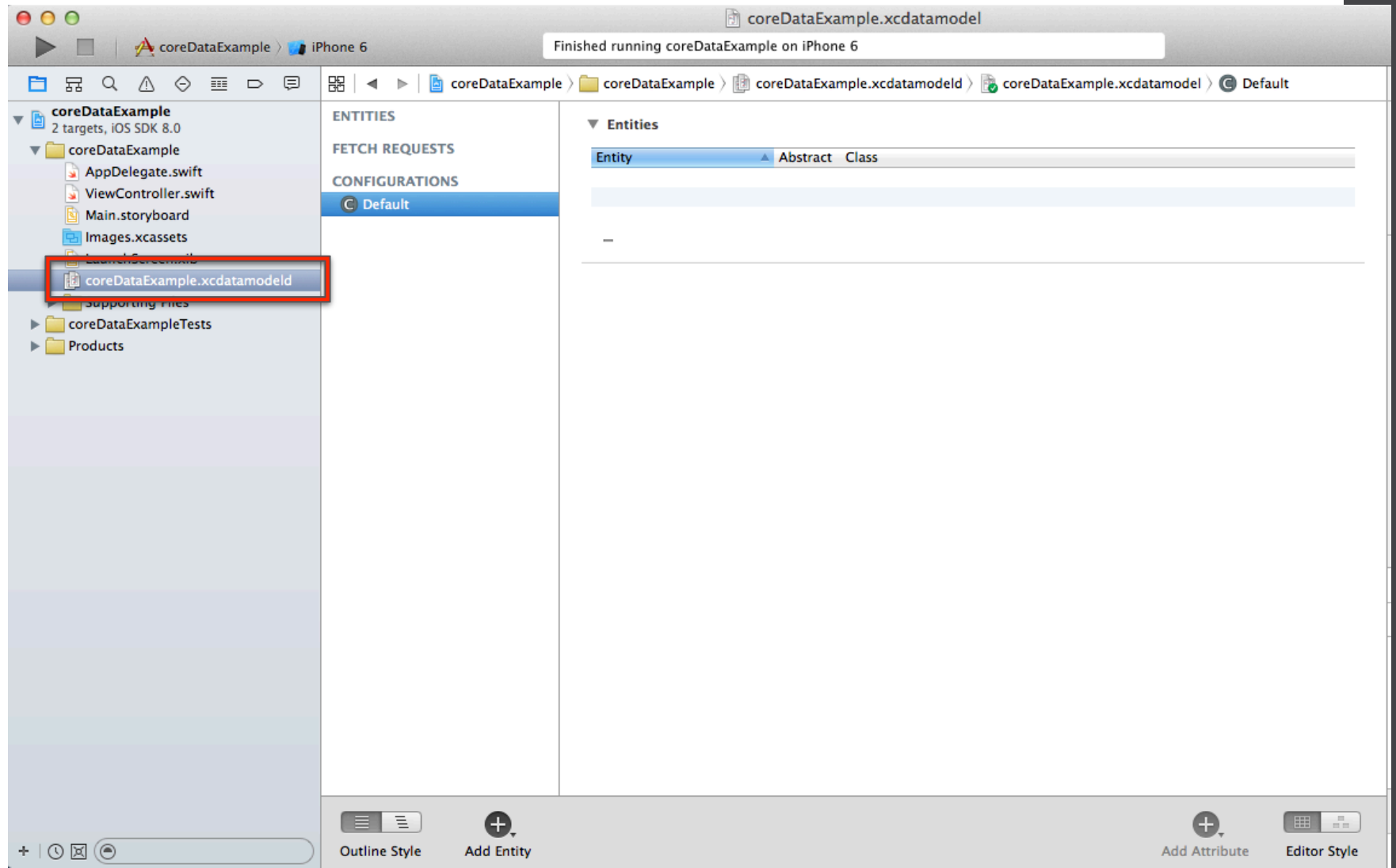
No Persistence

- ตอนนี้เพิ่มข้อมูลก็จะเพิ่มลงใน Array แต่เวลา Force quit หรือ reboot device รายการก็จะหายหมดเลย
- Core Data จะแก้ปัญหาข้อมูลหายเวลาเปิดแอปใหม่อีกครั้ง
- ลองกดปุ่ม Home (Shift+⌘+H) 2 ครั้งเพื่อ Kill app โดยลากแอปขึ้นข้างบนจอ
- แล้วกดเข้าแอปใหม่อีกที



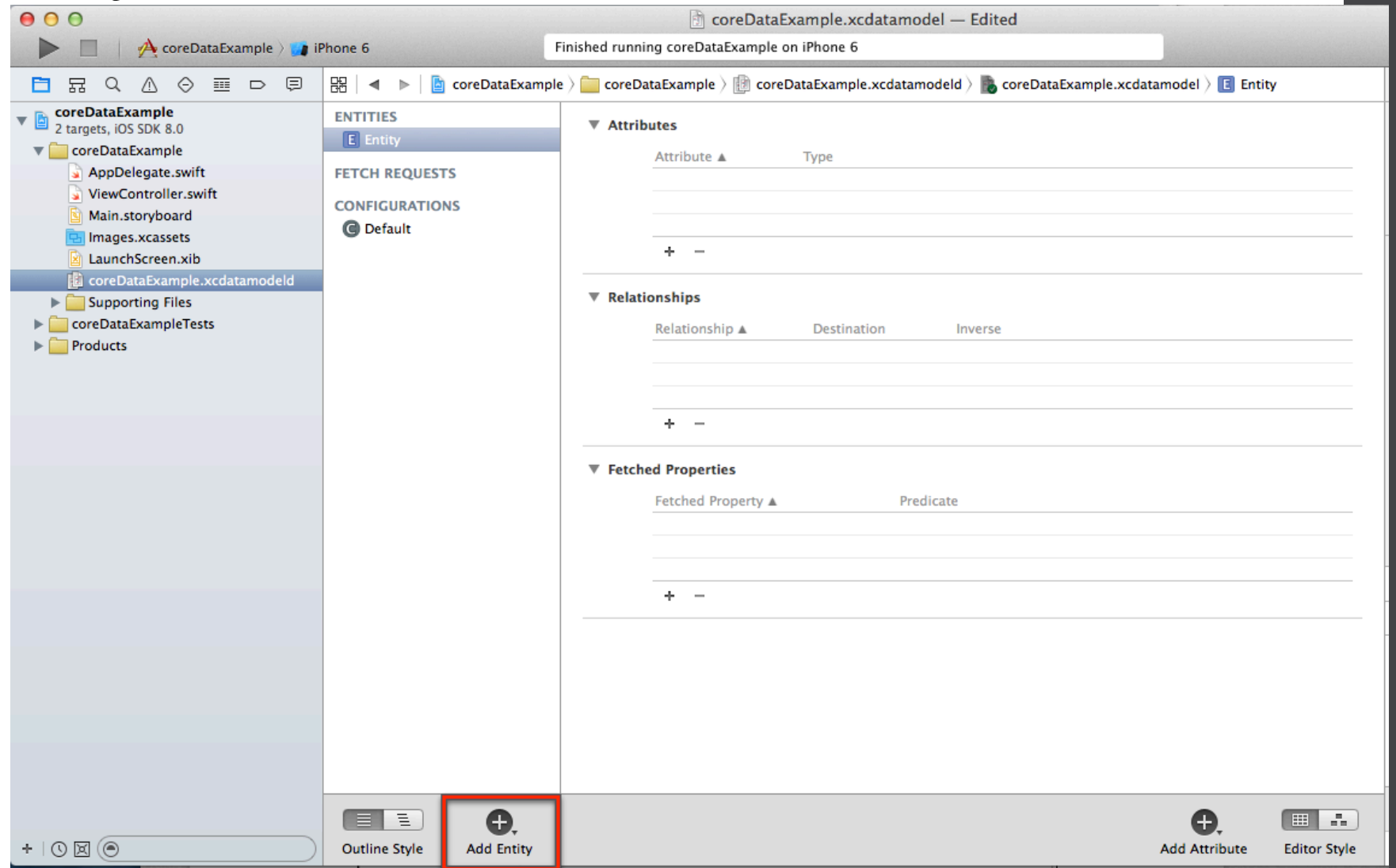
Design Core Data

- สร้าง manage object model
- มันใช้เหมือน SQLite database



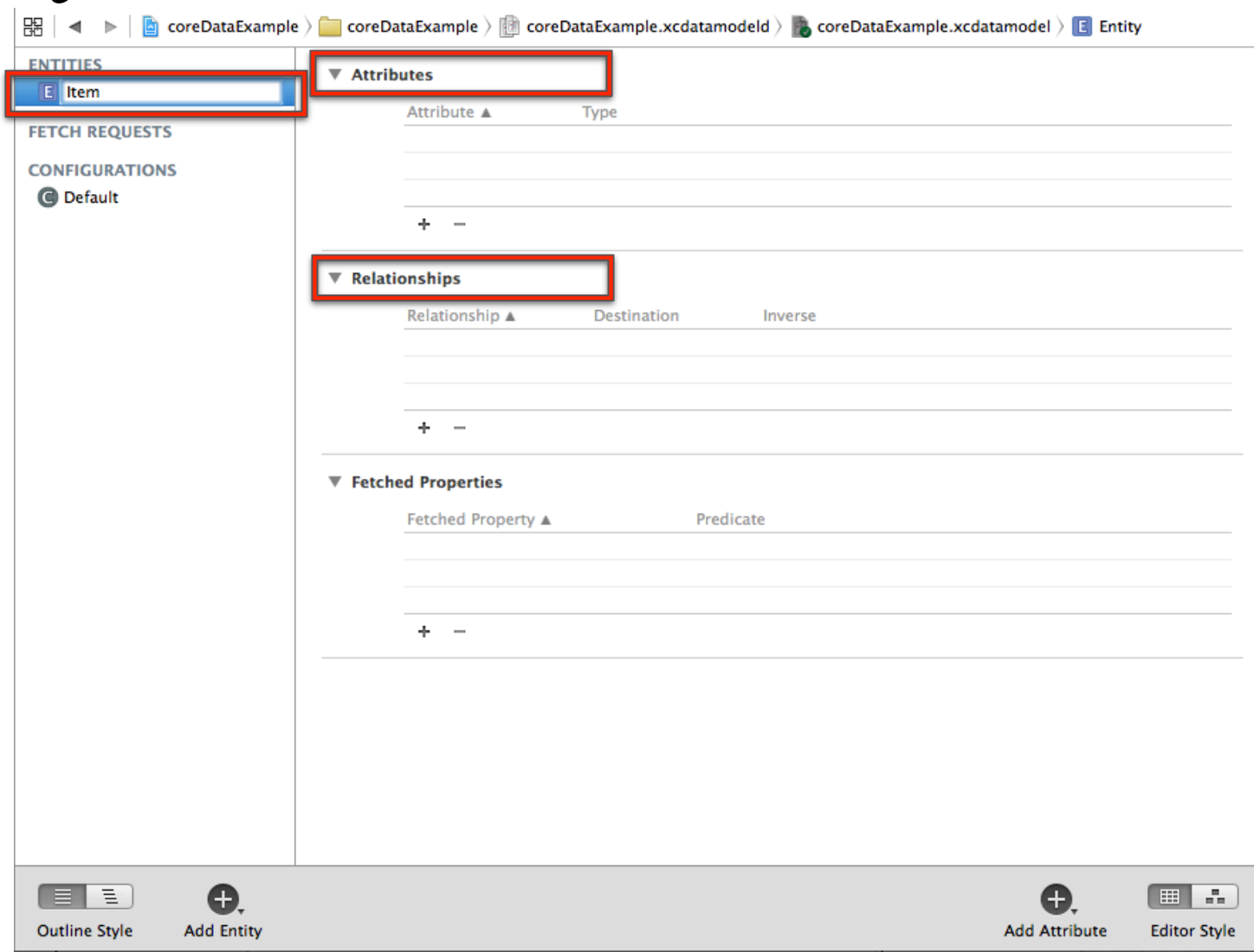
Add Entity

- กดเพิ่ม Entity ปุ่มล่างซ้าย



แก้ไข Entity

- แก้ชื่อ Entity เป็น Item
- Entity คือ Table
- Attribute คือ ข้อมูลที่จะเก็บใน Table
- Relationship คือ ความสัมพันธ์ระหว่าง Table



เพิ่ม Attribute: name, Type: String

ENTITIES

E Item

FETCH REQUESTS

CONFIGURATIONS

Default

▼ Attributes

Attribute ▲	Type
U name	✓ Undefined
	Integer 16
	Integer 32
	Integer 64
	Decimal
	Double
	Float
	String
	Boolean
	Date
	Binary Data
	Transformable

+ -

▼ Relationships

Relationship ▲	Inverse

+ -

▼ Fetched Properties

Fetch Property ▲	Predicate

+ -

แก้ตัวแปร Array String เพื่อเก็บรายการของที่จะนำมาแสดงใน TableView มาเป็น NSManagedObject

```
import UIKit  
import CoreData
```

```
class ViewController: UIViewController, UITableViewDataSource {  
  
    //var items = [String]()  
    var items = [NSManagedObject]()
```

แก้ func ของ UITableViewDataSource

```
//Replace both UITableViewDataSource methods
func tableView(tableView: UITableView,
  numberOfRowsInSection section: Int) -> Int {
    return items.count
}

func tableView(tableView: UITableView,
  cellForRowAtIndexPath
  indexPath: NSIndexPath) -> UITableViewCell {

    let cell =
    tableView.dequeueReusableCellWithIdentifier("Cell")
    as UITableViewCell

    let item = items[indexPath.row]
    cell.textLabel!.text = item.valueForKey("name") as String?

    return cell
}
```

เพื่อเอาชื่อมาจาก Attribute ของ
NSManagedObject เพราะมันไม่รู้จัก
Attribute ที่เราสร้างไว้ใน Data model

แก้ไขการทำงานของ AddItem func

```
let saveAction = UIAlertAction(title: "Save",  
    style: .Default) { (action: UIAlertAction!) -> Void in  
  
    let textField = alert.textFields![0] as UITextField  
    self.saveName(textField.text)  
    self.tableView.reloadData()  
}
```

1. กดปุ่ม Save ก็เรียกฟังก์ชัน saveName

เพิ่มโค้ด saveName func

```
func saveName(name: String) {  
    //1  
    let appDelegate =  
        UIApplication.sharedApplication().delegate as AppDelegate  
  
    let managedContext = appDelegate.managedObjectContext!  
  
    //2  
    let entity = NSEntityDescription.entityForName("Item",|  
        inManagedObjectContext:  
            managedContext)  
  
    let item = NSManagedObject(entity: entity!,  
        insertIntoManagedObjectContext:managedContext)  
  
    //3  
    item.setValue(name, forKey: "name")  
  
    //4  
    var error: NSError?  
    if !managedContext.save(&error) {  
        println("Could not save \(error), \(error?.userInfo)")  
    }  
  
    //5  
    items.append(item)  
}
```

1. สร้าง object ManagedObjectContext เพื่อ สร้าง reference มาที่ app delegate

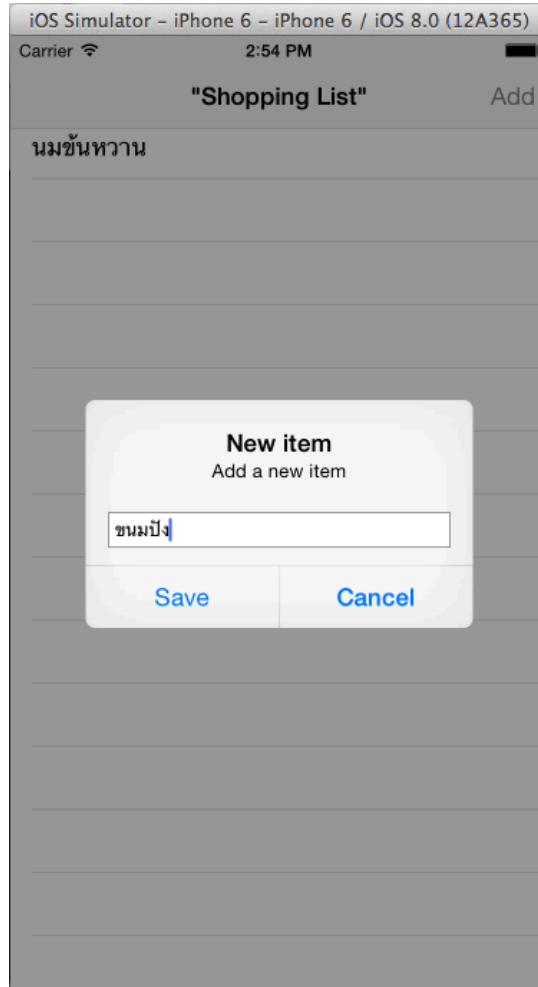
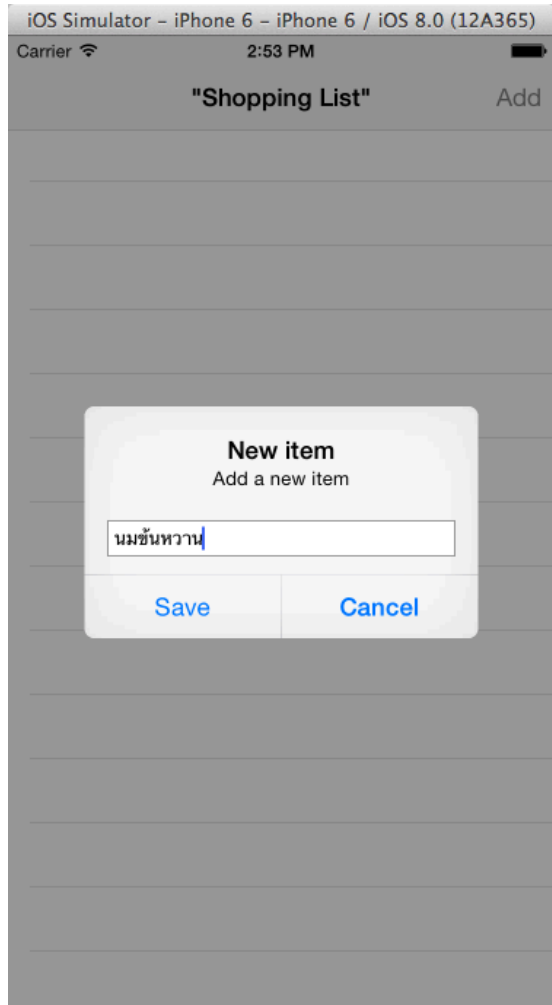
2. สร้าง entity เพื่อ link กับโค้ดตอน runtime

3. สร้าง name attribute โดยใช้ key-value coding ตามที่เราออกแบบไว้ใน data model

4. Commit การเปลี่ยนแปลงใน Item มี NSError เป็น pointer เมื่อมี error จะเช็ค

5. เพิ่มข้อมูลลงใน Core data ลงไปใน Items Array เพื่อแสดงใน table view

Build and Run



No Persistence

- ตอนนี้เพิ่มข้อมูลก็จะเพิ่มลงใน Array แต่เวลา Force quit หรือ reboot device รายการก็จะหายหมดเลย
- Core Data จะแก้ปัญหาข้อมูลหายเวลาเปิดแอปใหม่อีกครั้ง
- ลองกดปุ่ม Home (Shift+⌘+H) 2 ครั้งเพื่อ Kill app โดยลากแอปขึ้นข้างบนจอ
- แล้วกดเข้าแอปใหม่อีกที
- ยังว่างอยู่เหมือนเดิมเพราะข้อมูลอยู่ใน Core Data แต่เรายังไม่ได้ fetch มันออกมา



To Fetch data เพิ่มโค้ด viewDidLoad

```
override func viewWillAppear(animated: Bool) {
    super.viewWillAppear(animated)

    //1
    let appDelegate =
    UIApplication.sharedApplication().delegate as AppDelegate

    let managedContext = appDelegate.managedObjectContext!

    //2
    let fetchRequest = NSFetchRequest(entityName:"Item")

    //3
    var error: NSError?

    let fetchedResults =
    managedContext.executeFetchRequest(fetchRequest,
    error: &error) as [NSManagedObject]?

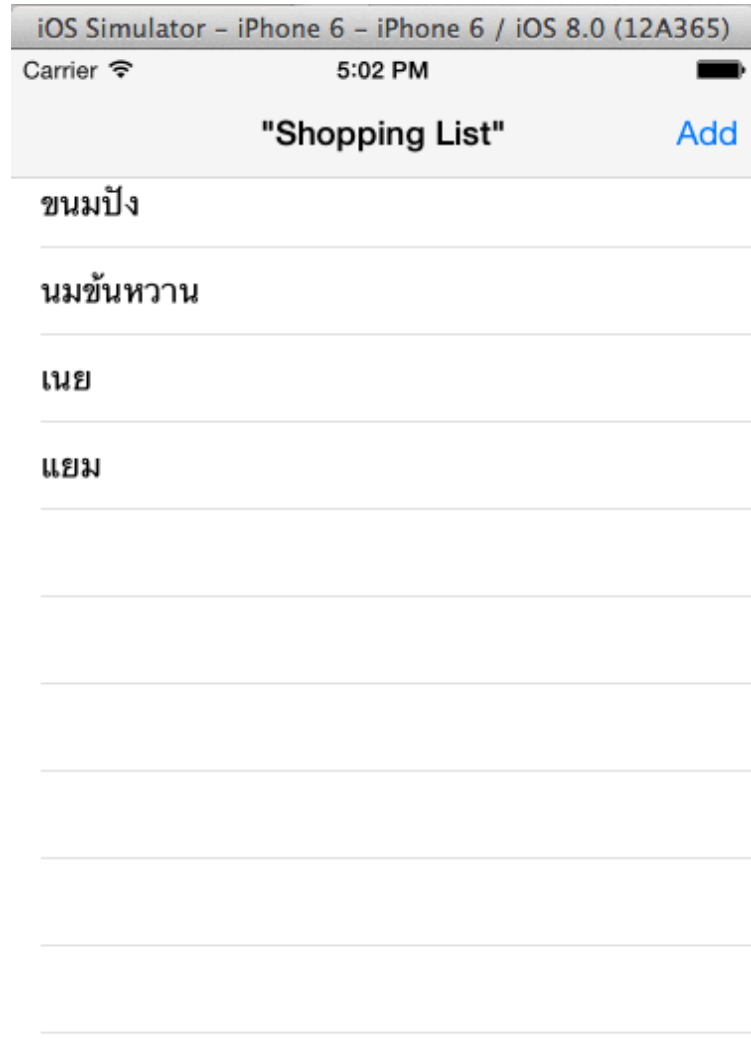
    if let results = fetchedResults {
        items = results
    } else {
        println("Could not fetch \(error), \(error!.userInfo)")
    }
}
```

1. สร้าง object ManagedObjectContext เพื่อ สร้าง reference มาที่ app delegate

2. Fetching data ดึงข้อมูลจาก object Item

3. ส่ง NSManagedObject Array ที่พบคืนมา

Build and Run



References

- <http://www.raywenderlich.com/85578/first-core-data-app-using-swift>