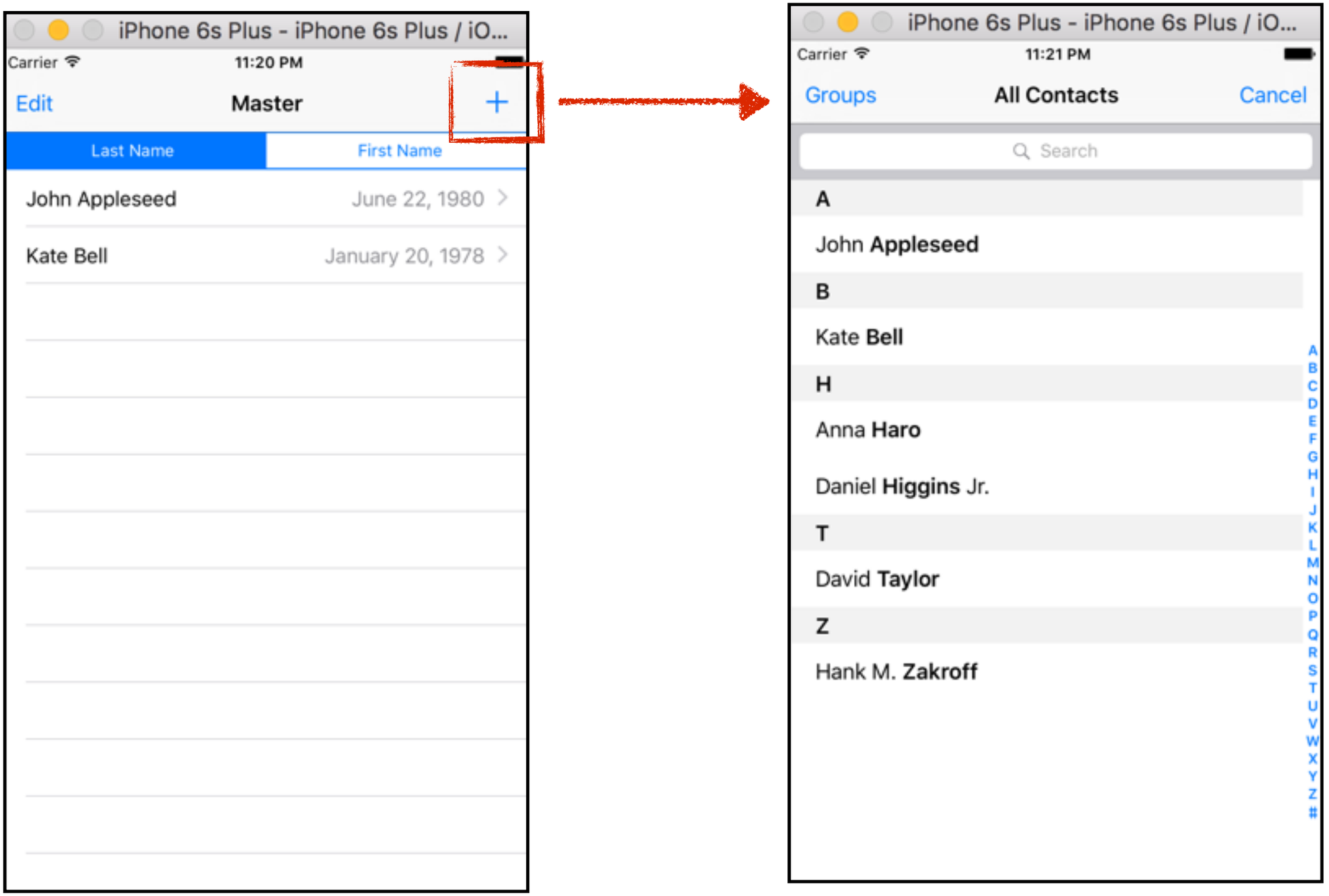
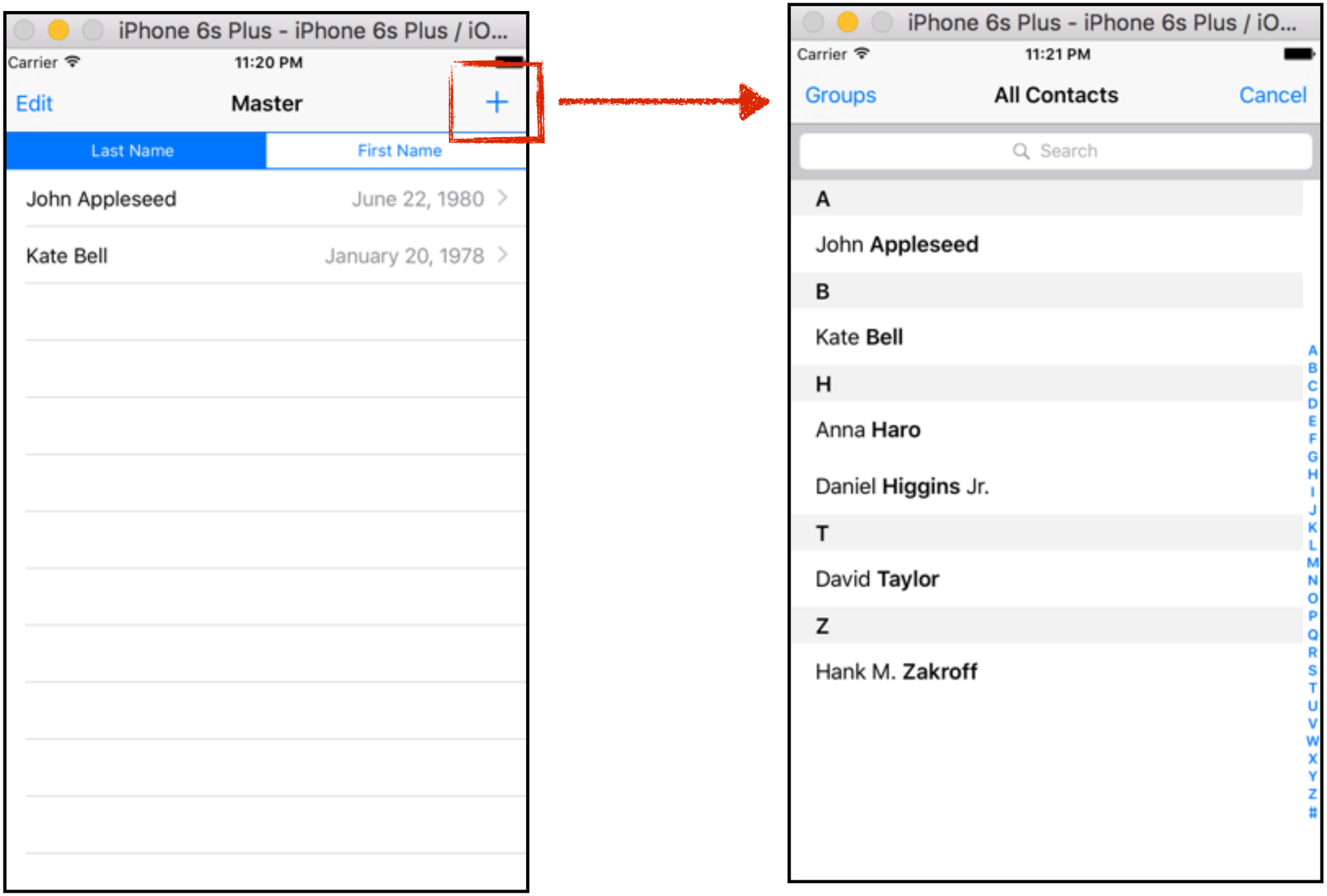
We ❤ Swift

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Workshop :: Birthday



Using Core data



Workshop :: Add Unit test

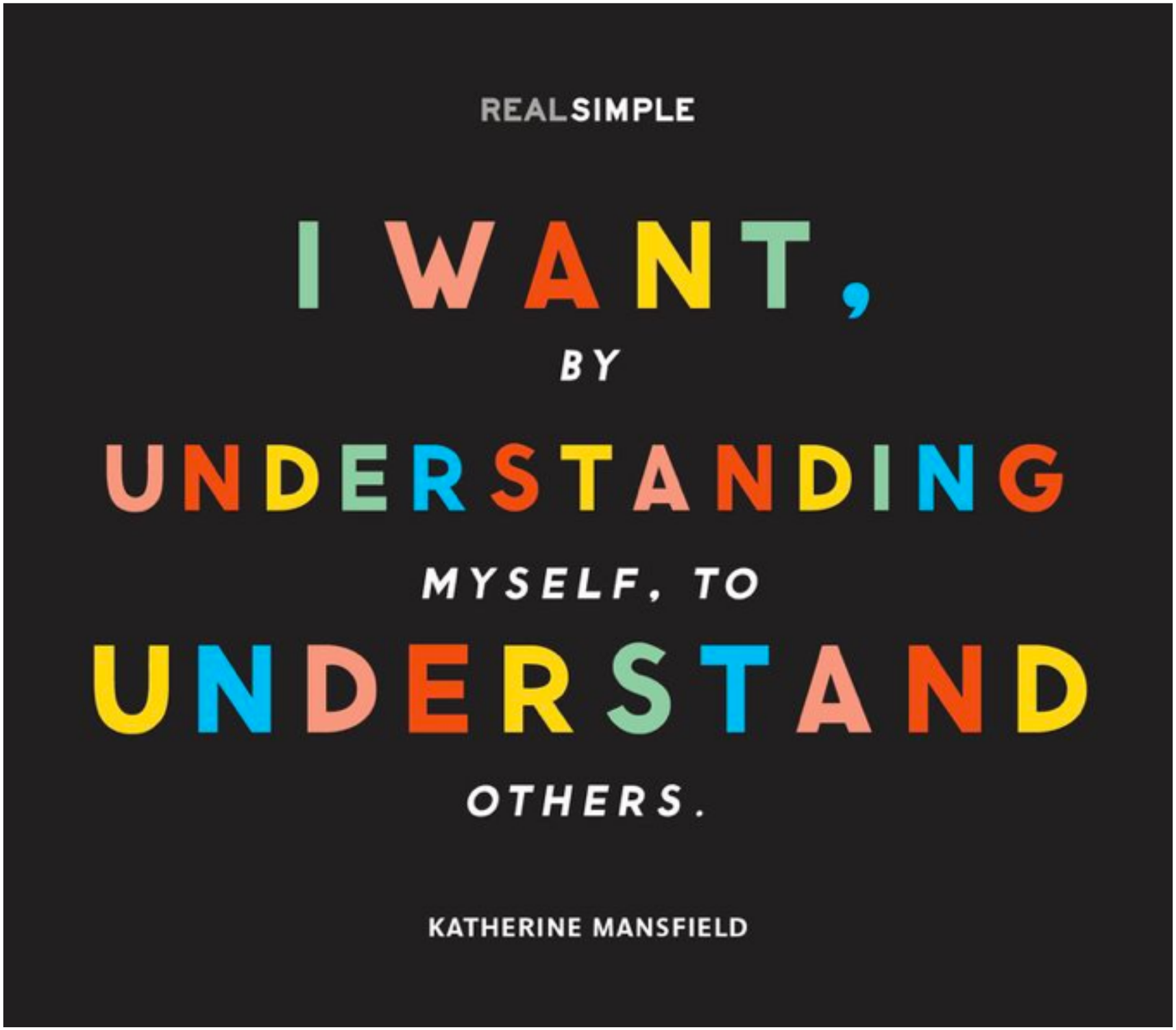
บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

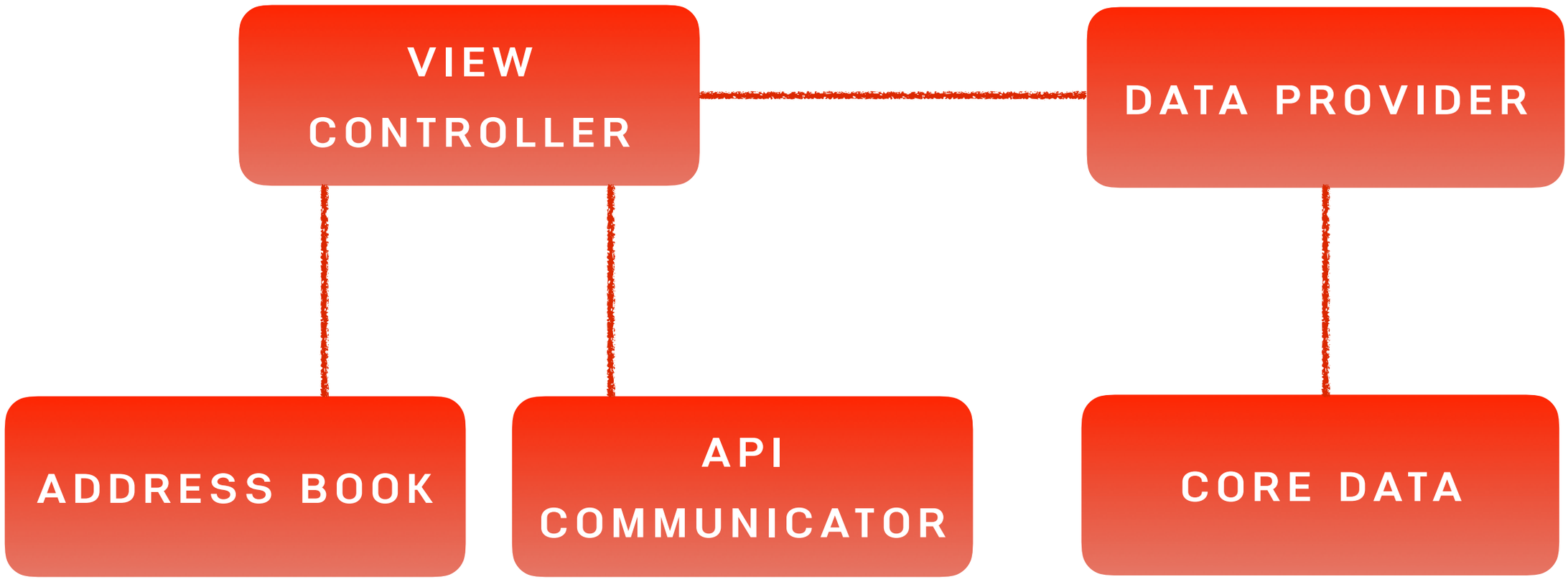
Pros

- More confidence
- Fast feedback
- Focus
- Regression
- Refactoring
- Modularity
- Documentation

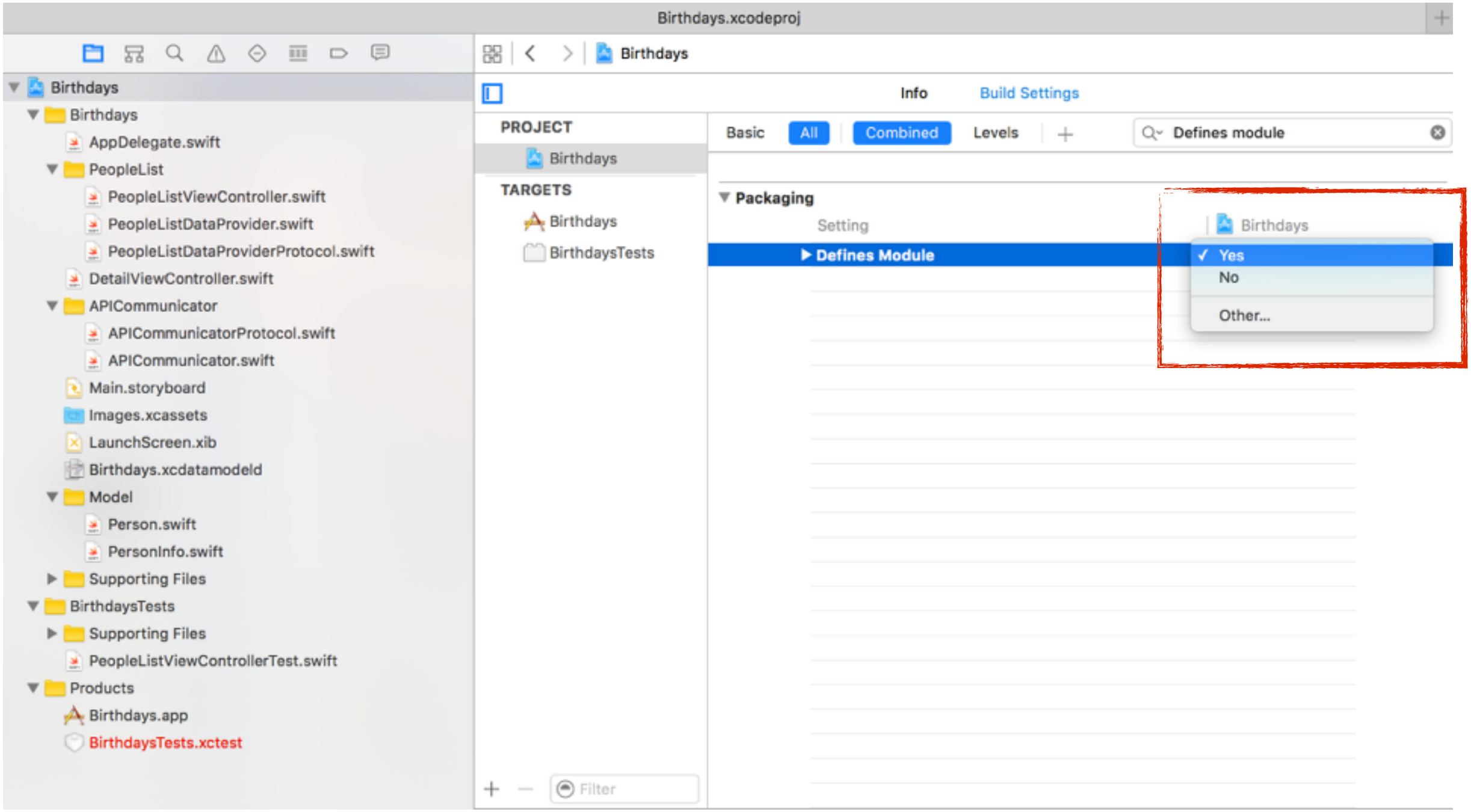
Cons

- More code
- Hard to maintain
- Writing tests take time
- No silver bullets

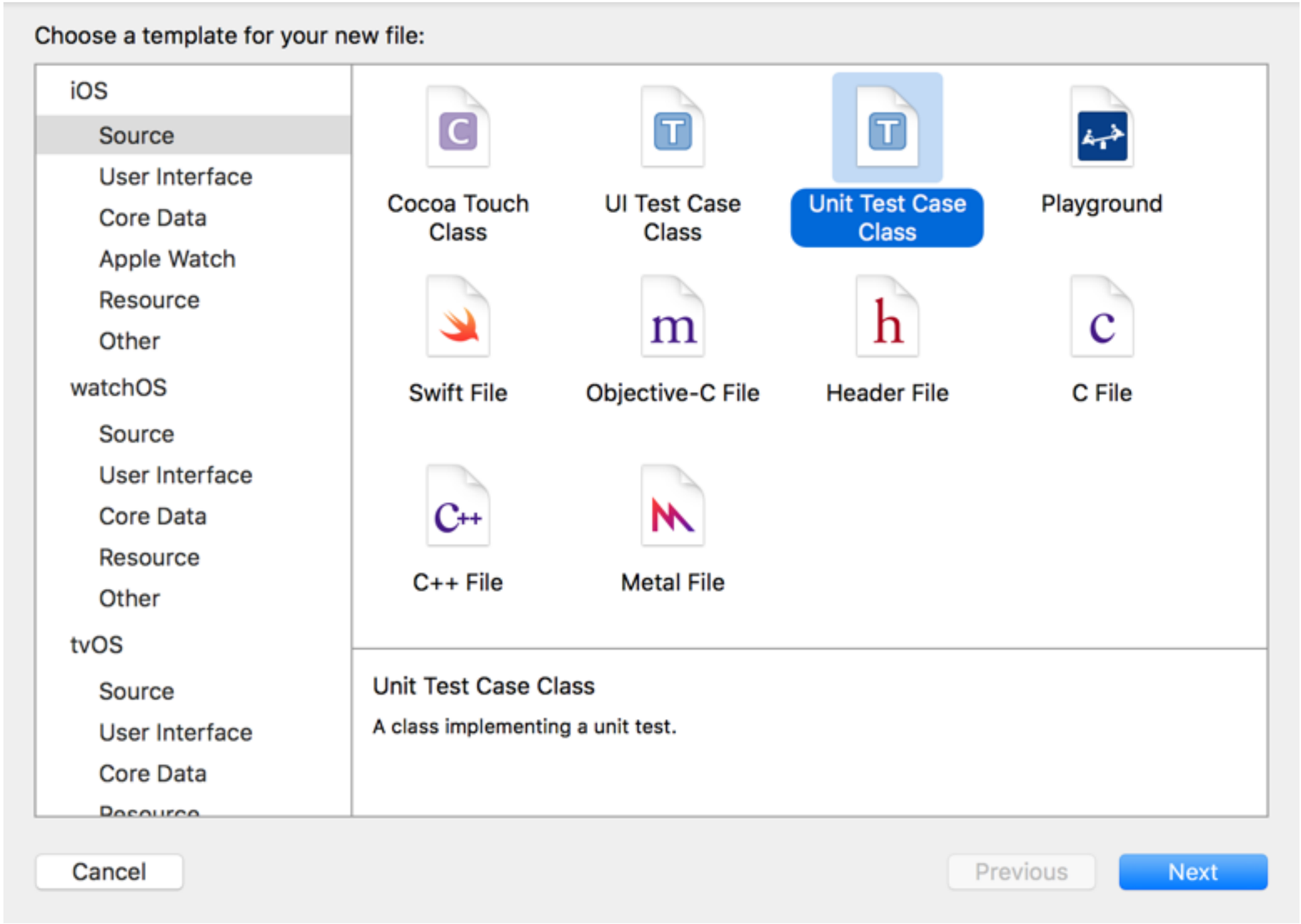



VIEW CONTROLLER
DATA PROVIDER
ADDRESS BOOK
API COMMUNICATOR
CORE DATA

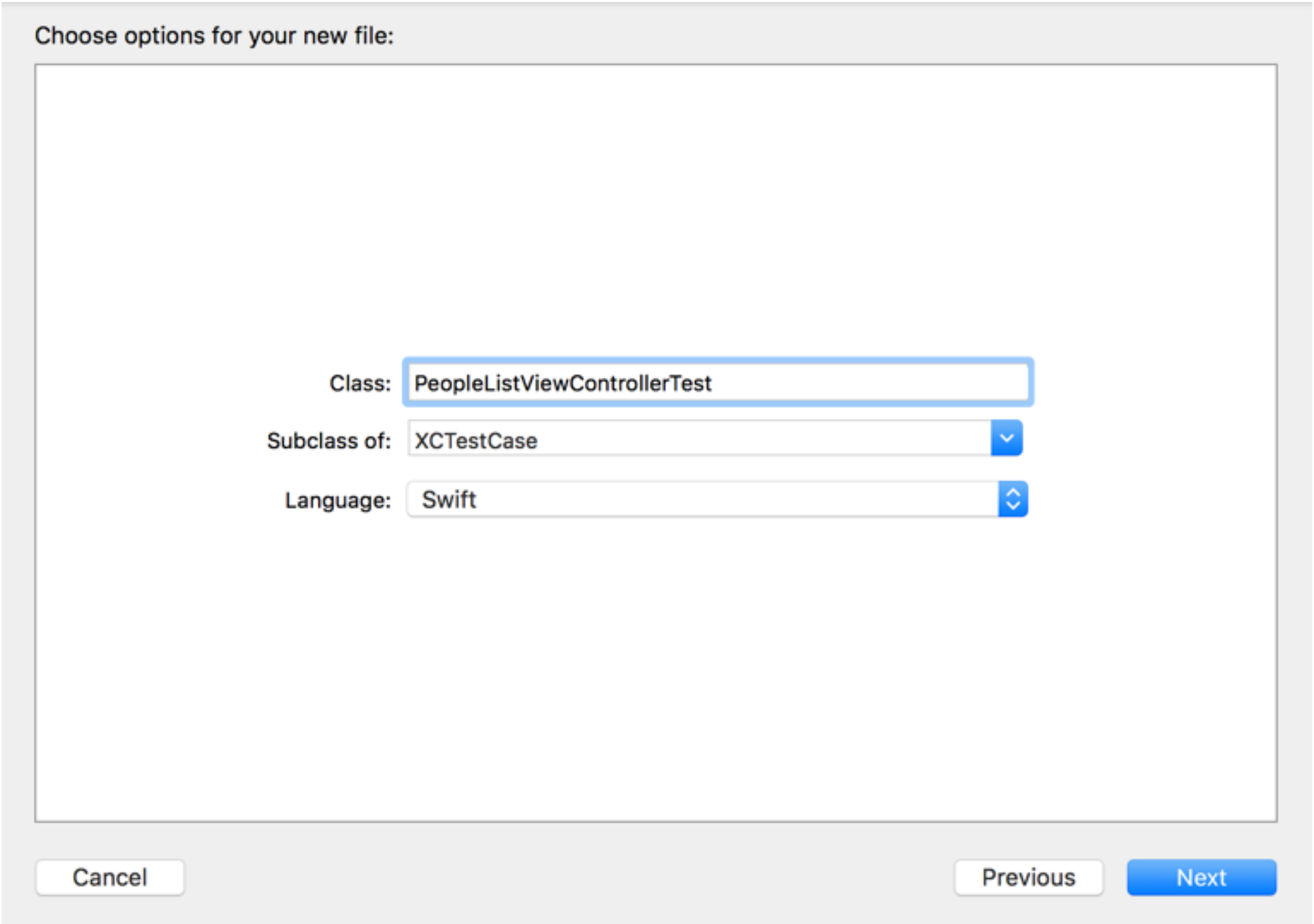
Prepare App for Testing



Create Test case class



Create Test case class



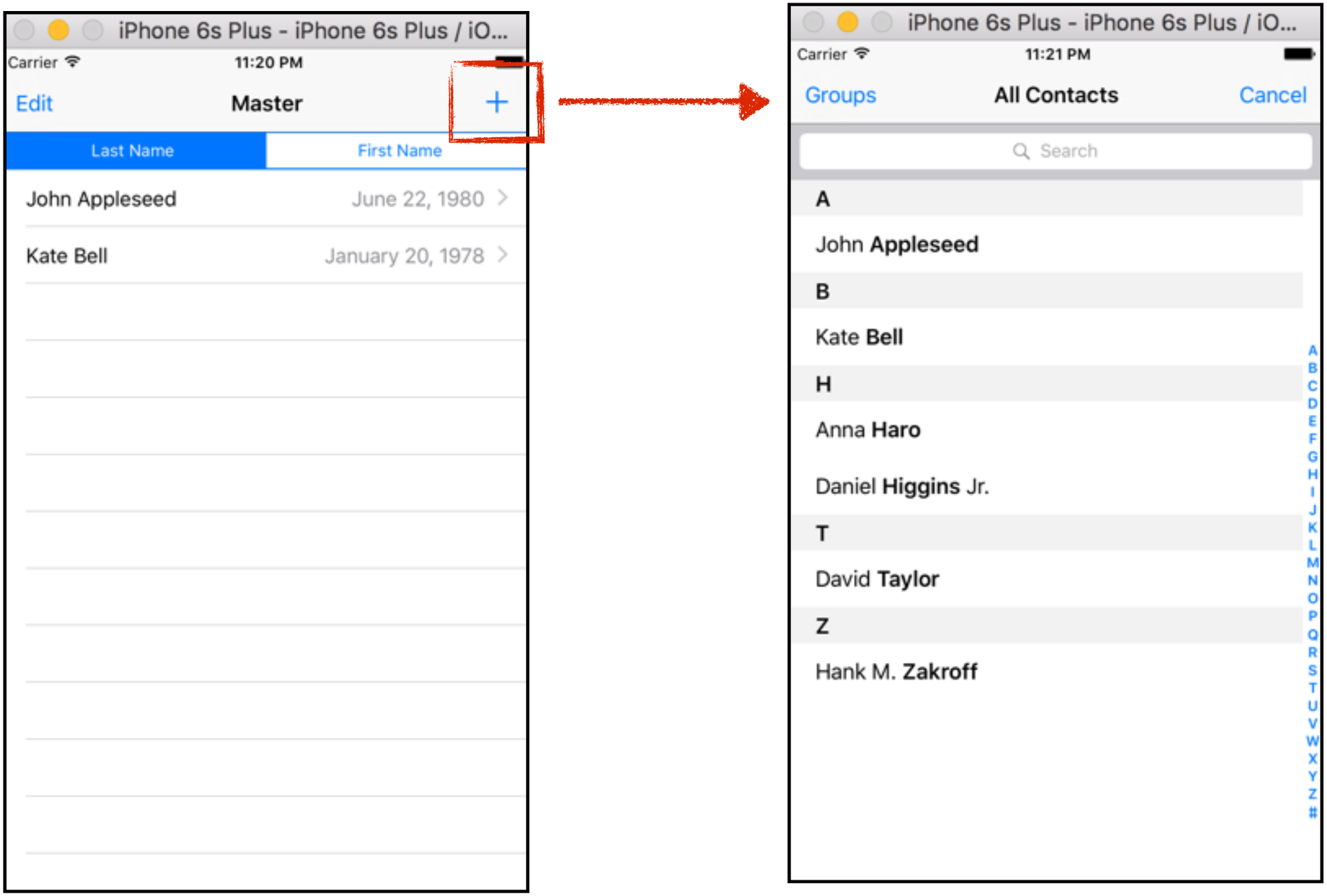
Create Test case class

```
import UIKit
import XCTest
@testable import Birthdays

class PeopleListViewControllerTest: XCTestCase {

    func testExample() {
        XCTAssert(true, "PASS")
    }


}
```

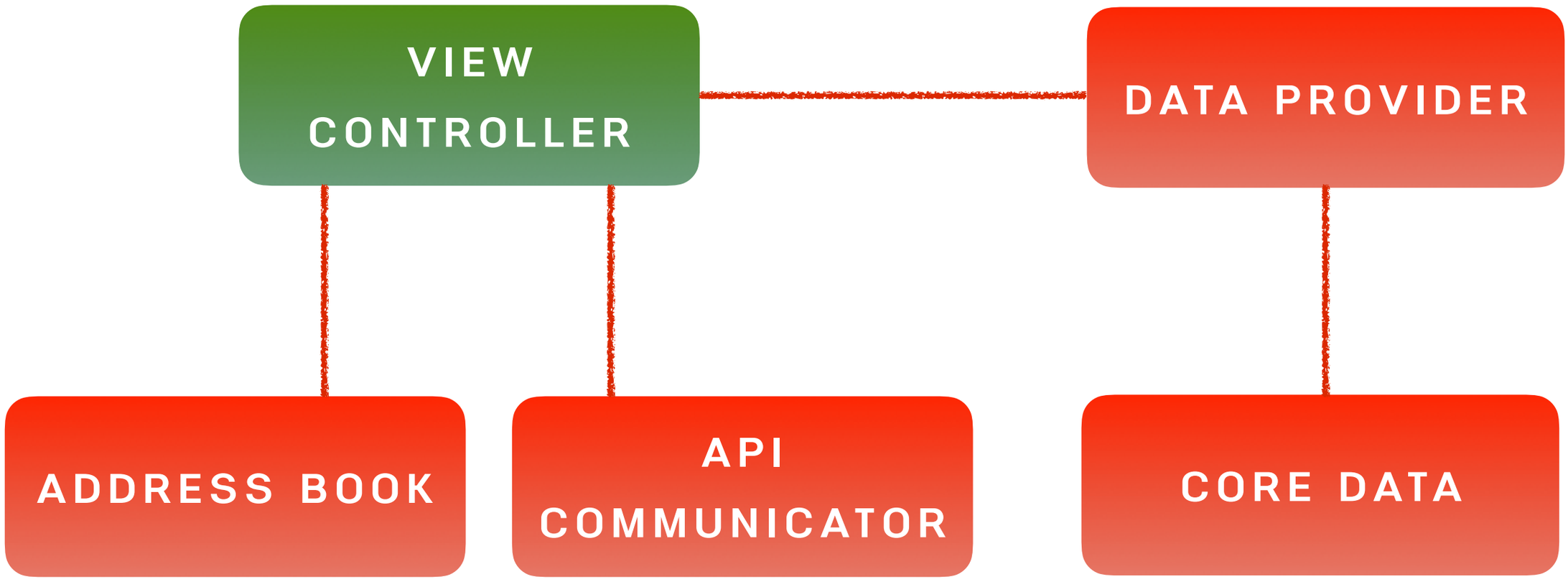
ต้องการทดสอบอะไรบ้าง ?

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Using Core data



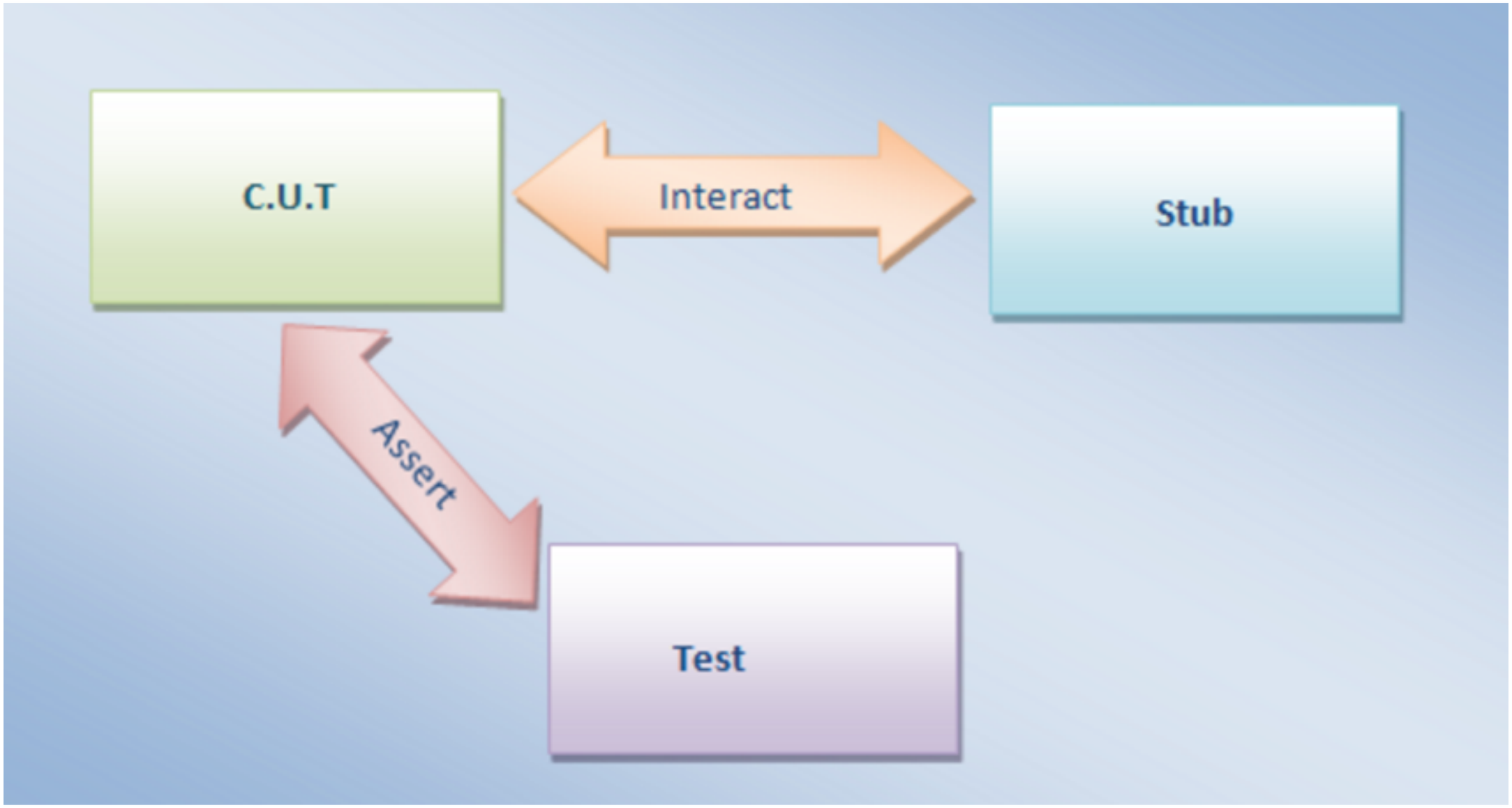
PeopleListViewController



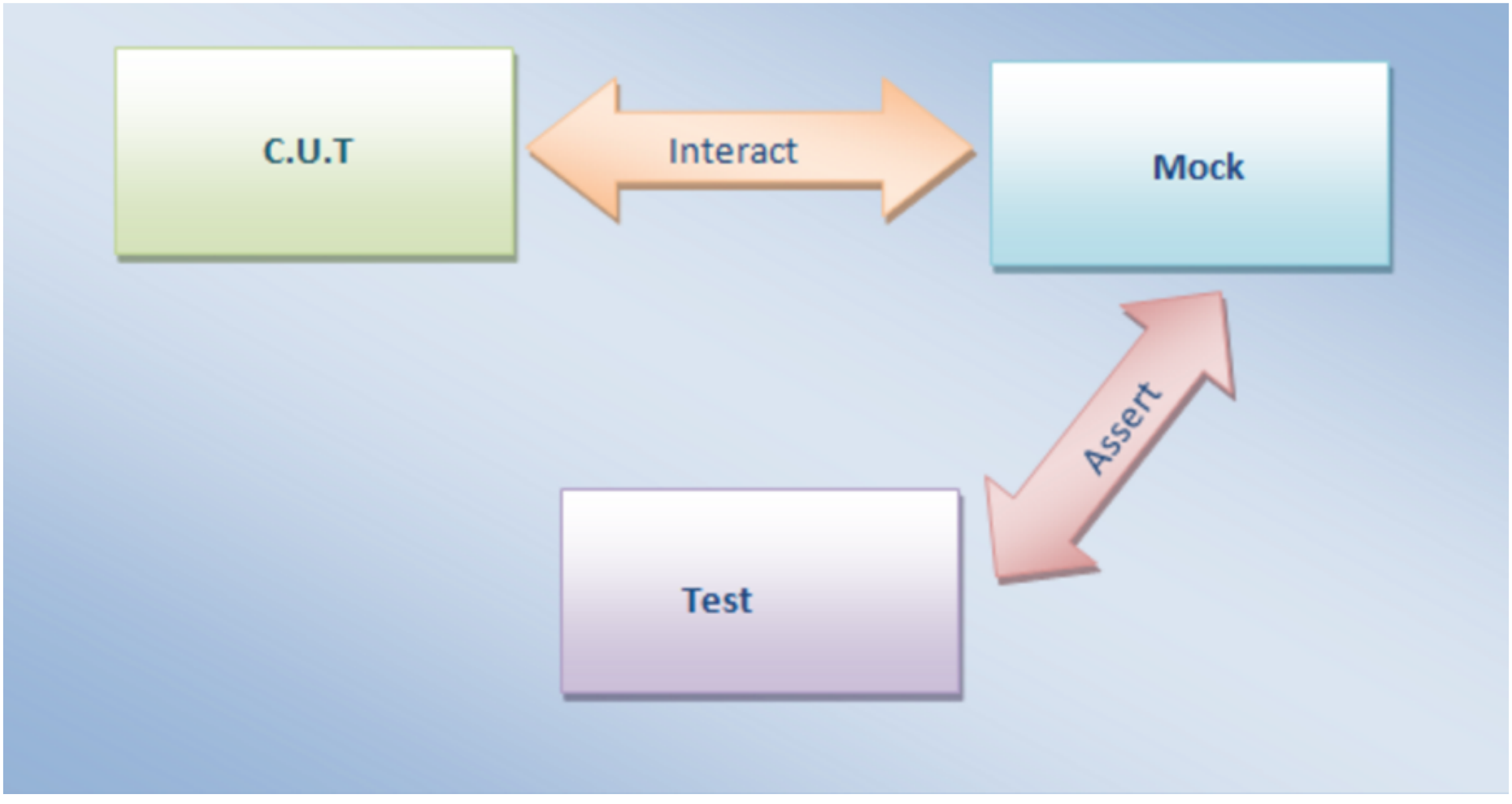
PeopleListViewController

- Add person button
- Tableview and Data Provider
- Call addPerson after adding
- Fetching people from API
- Sending people to API

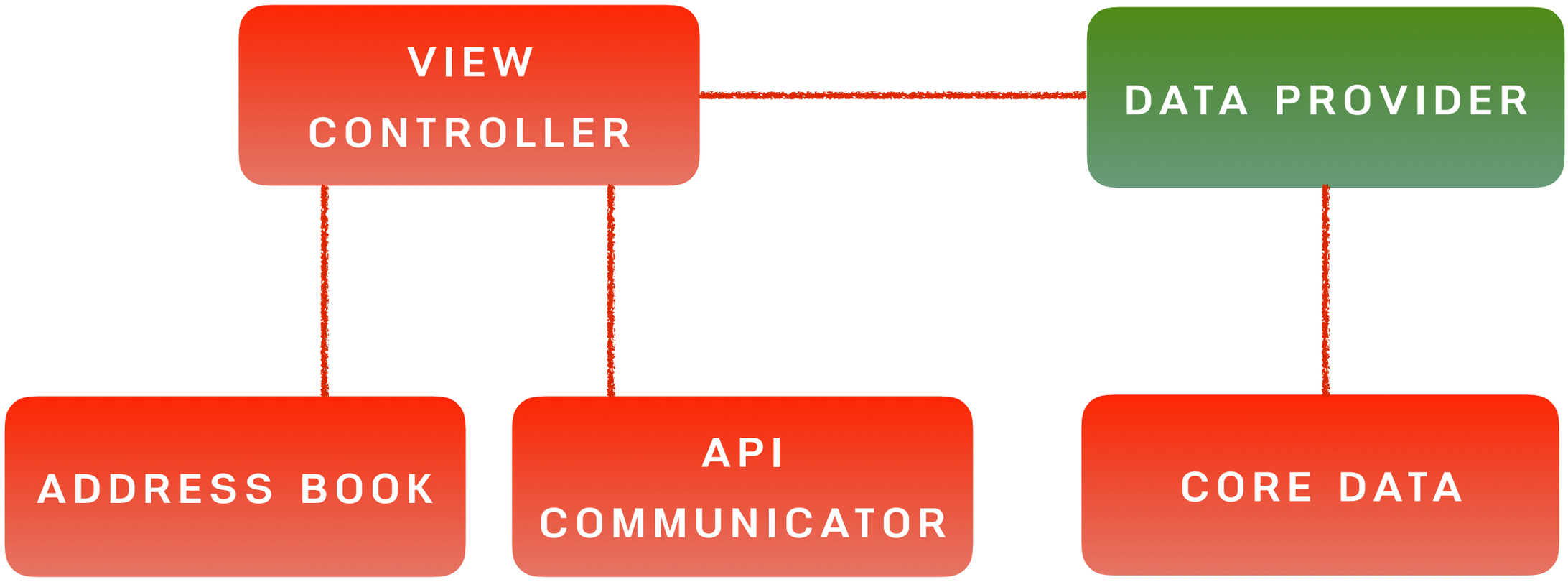
Stub



Mock



PeopleListDataProvider



PeopleListDataProvider

- Data in persistence

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

1. Add person button

Create View Controller

```
class PeopleListViewControllerTests: XCTestCase {

    var viewController: PeopleListViewController!

    override func setUp() {
        super.setUp()
        viewController = UIStoryboard(name: "Main", bundle: nil).
            instantiateViewControllerWithIdentifier("PeopleListViewController") as!
            PeopleListViewController

    }

    override func tearDown() {
                super.tearDown()
    }

    func testAddPersonShouldBeSet() {
        let _ = viewController.view
        let addPersonButton = viewController.navigationItem.rightBarButtonItem

        XCTAssertNotNil(addPersonButton, "Should not null")
    }
```

Create Test Case

```
class PeopleListViewControllerTests: XCTestCase {

    var viewController: PeopleListViewController!

    override func setUp() {
        super.setUp()
        viewController = UIStoryboard(name: "Main", bundle: nil).
            instantiateViewControllerWithIdentifier("PeopleListViewController") as!
            PeopleListViewController
    }

    override func tearDown() {
                super.tearDown()
    }
```

```
    func testAddPersonShouldBeSet() {
        let _ = viewController.view
        let addPersonButton = viewController.navigationItem.rightBarButtonItem

        XCTAssertNotNil(addPersonButton, "Should not null")
    }
```

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

See error and fix !!

```
11  override public func viewDidLoad() {
12    super.viewDidLoad()
13    self.navigationItem.leftBarButtonItem = self.editButtonItem()
14
15    let addButton = UIBarButtonItem(barButtonSystemItem: .Add, target: self, action: #selec
          (PeopleListViewController.addPerson))
16    self.navigationItem.rightBarButtonItem = addButton
17
18    assert(dataProvider != nil, "dataProvider is not allowed to be nil at this point")
19    tableView.dataSource = dataProvider
20    dataProvider?.tableView = tableView
21  }
22
23  func addPerson() {
24    let picker = ABPeoplePickerNavigationController()
25    picker.peoplePickerDelegate = self
26    presentViewController(picker, animated: true, completion: nil)
27  }
```

Thread 1: EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP,

Mock DataProvider

```
class MockDataProvider: PeopleListDataProvider {
    var addPersonGotCalled = false

    override func addPerson(personInfo: PersonInfo) { addPersonGotCalled =
        true }
    override func fetch() { }
    override func tableView(tableView: UITableView, numberOfRowsInSection
        section: Int) -> Int { return 1 }
    override func tableView(tableView: UITableView, cellForRowAtIndexPath
        indexPath: NSIndexPath) -> UITableViewCell { return
        UITableViewCell() }
}
```

Update Test Case

```
func testAddPersonShouldBeSet() {
    viewController.dataProvider = MockDataProvider()

    let _ = viewController.view

    let addPersonButton = viewController.navigationItem.
        rightBarButtonItem

    XCTAssertNotNil(addPersonButton, "Should not null")
    XCTAssertEqual(addPersonButton?.action, #selector
        (PeopleListViewController.addPerson))
}
```

2. Set table view to Data Provider

Create Test Case

```
func testSetTableViewToDataProviderAfterLoadingView() {
    //Arrange
    let mockProvider = MockDataProvider()
    viewController.dataProvider = mockProvider

    //Act
    XCTAssertNil(mockProvider.tableView, "Before loading the tableview should be null")
    let _ = viewController.view

    //Assert
    XCTAssertNotNil(mockProvider.tableView, "The tableview should be set")
    XCTAssertEqual(mockProvider.tableView, viewController.tableView)

}
```

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

3. Call addPerson after adding

Create Test Case

```
func testCalledAddPersonOfDataProviderAfterAddingPerson() {
    //Arrange
    let mockProvider = MockDataProvider()
    viewController.dataProvider = mockProvider

    //Act
    let record: ABRecord = ABPersonCreate().takeRetainedValue()
    viewController.peoplePickerNavigationController(
        ABPeoplePickerNavigationController(), didSelectPerson: record)

    //Assert
    XCTAssert(mockProvider.addPersonGotCalled,
              "addPerson should have been called")
}
```

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

See error and fix !!

```
8
9   public init(firstName: String, lastName: String, birthday: NSDate) {
10    self.firstName = firstName
11    self.lastName = lastName
12    self.birthday = birthday
13  }
14
15  init(abRecord: ABRecord) {
16    self.firstName = ABRecordCopyValue(abRecord,
          kABPersonFirstNameProperty).takeRetainedValue() as! String
17    self.lastName = ABRecordCopyValue(abRecord, kABPersonLastNameProperty).t
          String
18    self.birthday = ABRecordCopyValue(abRecord, kABPersonBirthdayProperty).t
          NSDate
19  }
20 }
```

Thread 1: EXC_BAD_INSTRUCTION (co

Update Test Case

```
func testCalledAddPersonOfDataProviderAfterAddingPerson() {
    //Arrange
    let mockProvider = MockDataProvider()
    viewController.dataProvider = mockProvider

    //Act
    let record: ABRecord = ABPersonCreate().takeRetainedValue()
    ABRecordSetValue(record, kABPersonFirstNameProperty, "MockFirstname", nil)
    ABRecordSetValue(record, kABPersonLastNameProperty, "MockLastname", nil)
    ABRecordSetValue(record, kABPersonBirthdayProperty, NSDate(), nil)
    viewController.peoplePickerNavigationController(
        ABPeoplePickerNavigationController(), didSelectPerson: record)

    //Assert
    XCTAssert(mockProvider.addPersonGotCalled,
              "addPerson should have been called")
}
```

4. Fetching people from APIs

Create Mock of API

```
class MockAPICommunicator: APICommunicatorProtocol {
    var allPersonInfo = [PersonInfo]()
    var postPersonGotCalled = false

    func getPeople() -> (NSError?, [PersonInfo]?) {
        return (nil, allPersonInfo)
    }

    func postPerson(personInfo: PersonInfo) -> NSError? {
        postPersonGotCalled = true
        return nil
    }
}
```

Create Test Case

```
func testFetchingPeopleFromAPIShouldBeCalledAddPerson() {
    //Arrange
    let mockProvider = MockDataProvider()
    viewController.dataProvider = mockProvider

    let mockAPI = MockAPICommunicator()
    viewController.communicator = mockAPI

    //Act
    viewController.fetchPeopleFromAPI()

    //Assert
    XCTAssertTrue(mockProvider.addPersonGotCalled,
                  "addPerson should have been called")
}
```

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

See error and fix !!

```
71
72  func testFetchingPeopleFromAPIShouldBeCalledAddPerson() {
73      //Arrange
74      let mockProvider = MockDataProvider()
75      viewController.dataProvider = mockProvider
76
77      //Act
78      viewController.fetchPeopleFromAPI()
79
80          //Assert
81          XCTAssertTrue(mockProvider.addPersonGotCalled,        XCTAssertTrue failed
82                        "addPerson should have been called")
83      }
84
85  }
```

Update Test Case

```
func testFetchingPeopleFromAPIShouldBeCalledAddPerson() {
    //Arrange
    let mockProvider = MockDataProvider()
    viewController.dataProvider = mockProvider

    let mockAPI = MockAPICommunicator()
    mockAPI.allPersonInfo =
        [PersonInfo(firstName: "Firstname", lastName: "Lastname", birthday: NSDate())]
    viewController.communicator = mockAPI

    //Act
    viewController.fetchPeopleFromAPI()

    //Assert
    XCTAssertTrue(mockProvider.addPersonGotCalled,
                  "addPerson should have been called")
}
```

5. Sending people to API

Create Test Case

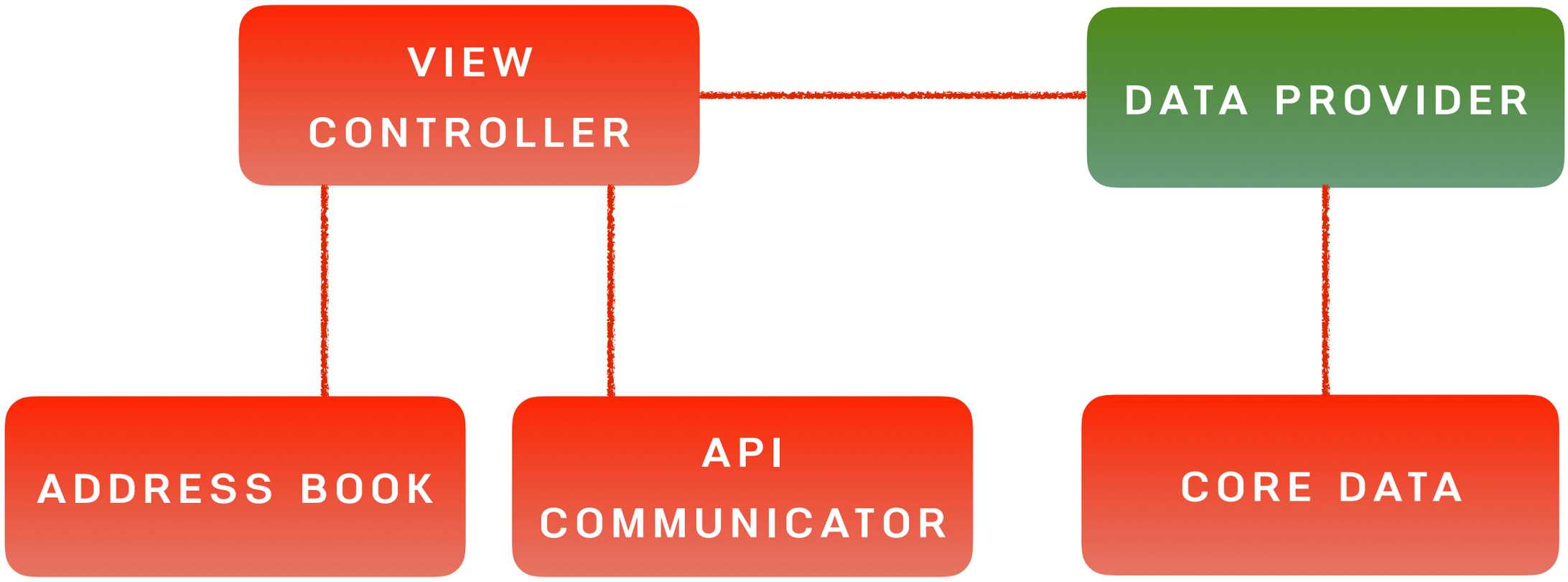
```
func testSendingPeopleToAPIShouldBeCalledPostPerson() {
    //Arrange
    let mockProvider = MockDataProvider()
    viewController.dataProvider = mockProvider

    let mockAPI = MockAPICommunicator()
    viewController.communicator = mockAPI

    //Act
    viewController.sendPersonToAPI(PersonInfo(firstName: "Firstname",
        lastName: "Lastname", birthday: NSDate()))


    //Assert
    XCTAssertTrue(mockAPI.postPersonGotCalled,
                  "postPerson should have been called")
}
```

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

PeopleListDataProvider

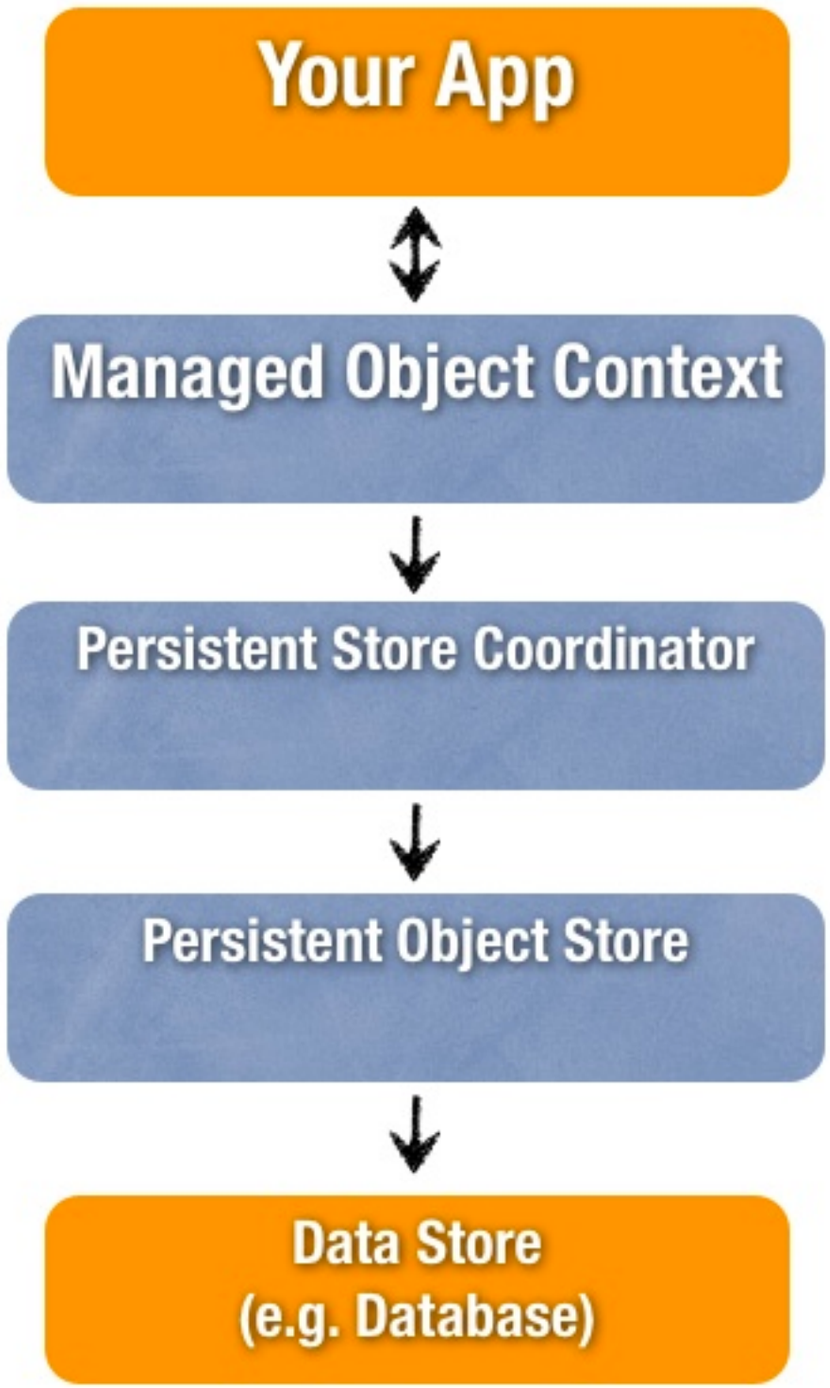
- Data in persistence

PeopleListDataProvider



How to create a persistent store in memory of device ?

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่



Fake Persistence Store

```
class PeopleListDataProviderTests: XCTestCase {

    var managedObjectContext: NSManagedObjectContext!
    var storeCoordinator: NSPersistentStoreCoordinator!
    var managedObjectModel: NSManagedObjectModel!
    var store: NSPersistentStore!

    var dataProvider: PeopleListDataProvider!
    var viewController: PeopleListViewController!
```

Fake Persistence Store

```
override func setUp() {
    super.setUp()

    managedObjectModel = NSManagedObjectModel.
        mergedModelFromBundles(nil)!
    storeCoordinator = NSPersistentStoreCoordinator
        (managedObjectModel: managedObjectModel)
    store = try? storeCoordinator.addPersistentStoreWithType
        (NSInMemoryStoreType, configuration: nil, URL: nil,
        options: nil)
    managedObjectContext = NSManagedObjectContext()
    managedObjectContext.persistentStoreCoordinator =
        storeCoordinator

    dataProvider = PeopleListDataProvider()
    dataProvider.managedObjectContext = managedObjectContext
}
```

Setup data

```
viewController = UIStoryboard(name: "Main", bundle: nil).
    instantiateViewControllerWithIdentifier("PeopleListViewController") as!
    PeopleListViewController

viewController.dataProvider = dataProvider

tableView = viewController.tableView

testRecord = PersonInfo(firstName: "FirstName",
                        lastName: "LastName",
                        birthday: NSDate())
```

Remove data

```
override func tearDown() {
    managedObjectContext = nil
    _ = try? storeCoordinator.removePersistentStore(store)
    super.tearDown()
}
```

Create persistence store

```
func testSetupDataSuccessfully() {
    XCTAssertNotNil(store, "Setup persistence store")
}
```

Add one person

```
func testOnePersonInThePersistantStoreResultsInOneRow() {
    //Act
    dataProvider.addPerson(testRecord)

    //Assert
    XCTAssertEqual(tableView.dataSource!.tableView(
                   tableView, numberOfRowsInSection: 0),1,
                   "After adding one person number of rows should be 1")
}
```

Show full name

```
func testPersonCellShowsFullName() {
    //Act
    dataProvider.addPerson(testRecord)

    //Assert
    let cell = tableView.dataSource!.tableView(
        tableView,
        cellForRowAtIndexPath: NSIndexPath(forRow: 0, inSection: 0)) as UITableViewCell

    XCTAssertEqual(cell.textLabel!.text!, "FirstName LastName", "Full name is not as
        expected")
}
```