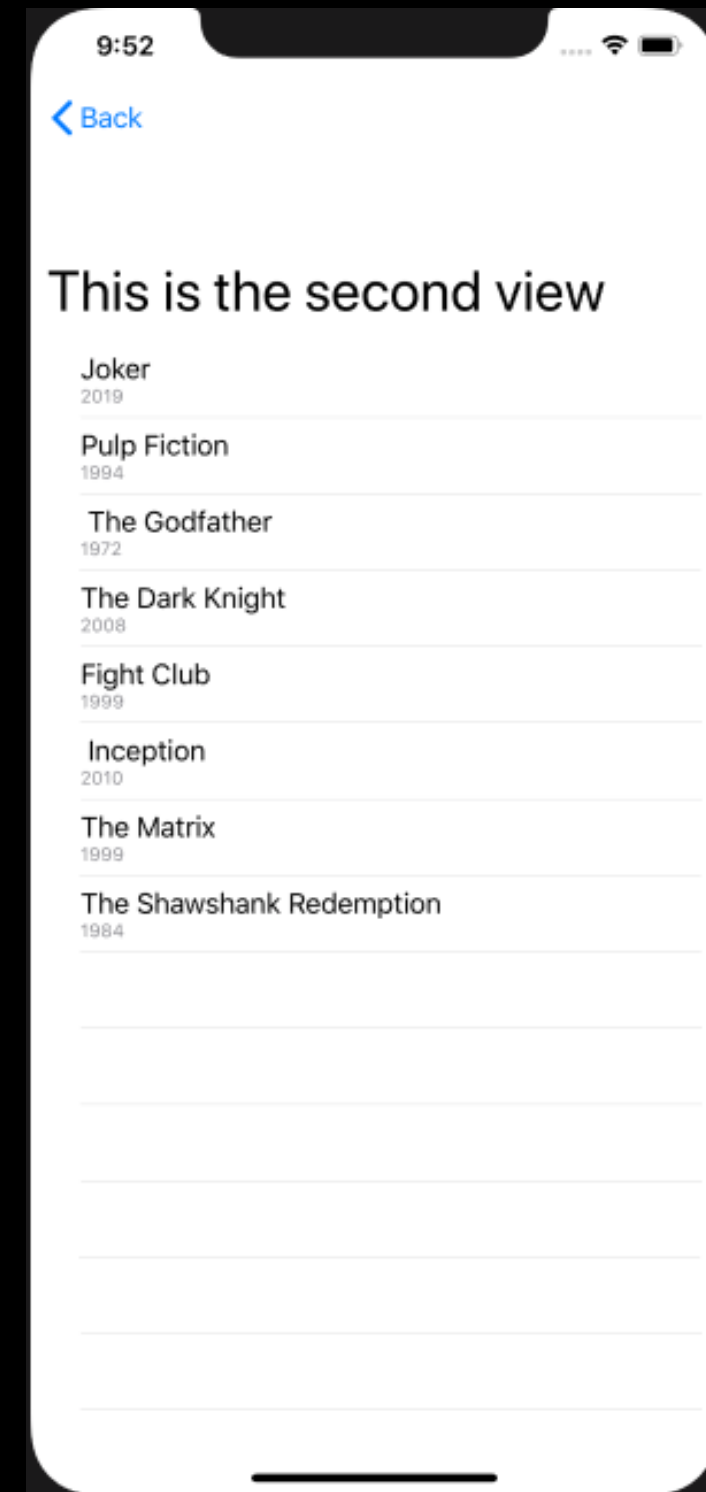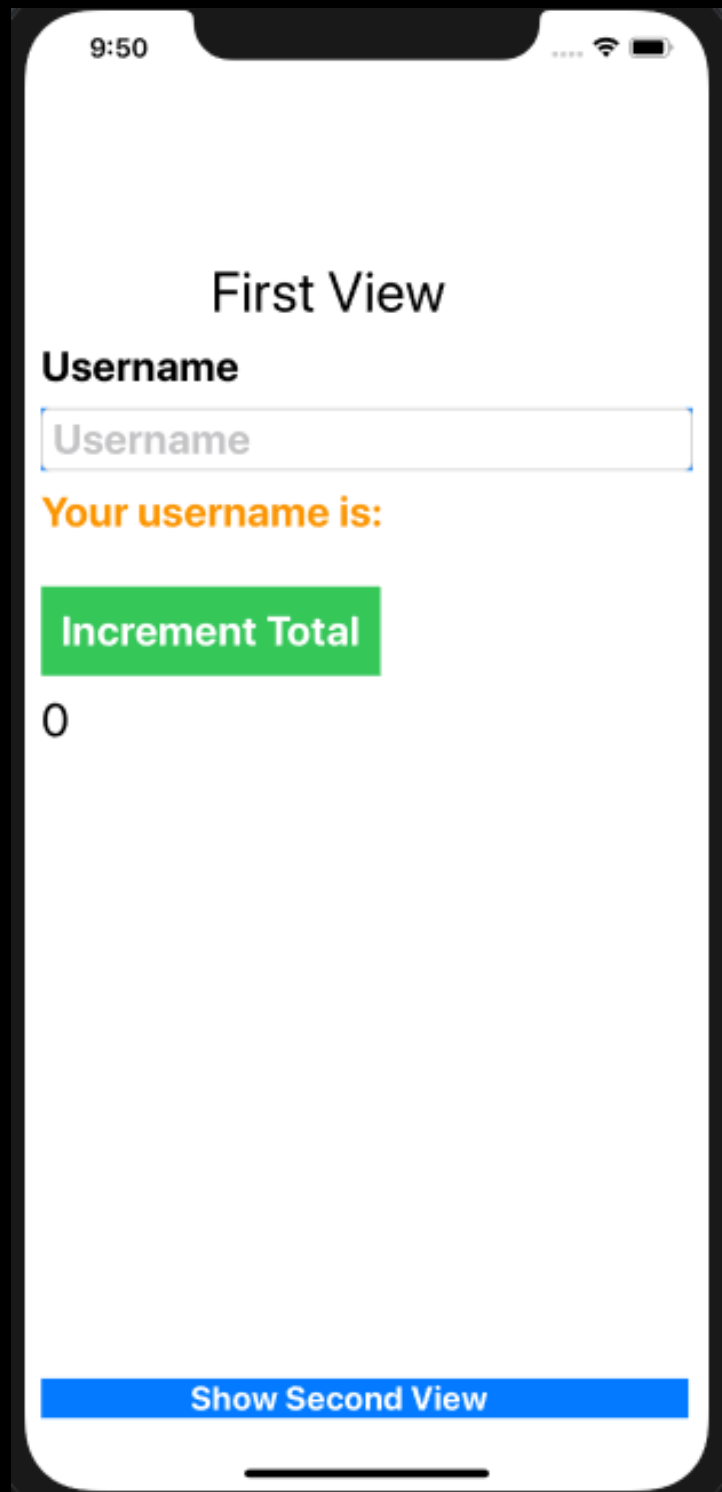# Application's first, and secondary  screens

# Test scenarios

```swift
class MartinezXCUIAsmokeTest: MartinezBaseXCUITestHelper {

    func testIncrementTotal() {

        given("I launch the App I validate all first view UI element appear") {
            incrementScreenElementValidation()
        }

        when("I enter the default username for testing") {
            typeDefaultUsernanme()
        }

        then("I increment the number to three") {
            incrementNumberTest()
        }

        and("I validate that the number has been increased") {
            defaultDataPersist()
        }
    }

    func testNavigateToSeconview() {

        given("I'm on the First View") {
            firstViewValidation()
        }

        when("I navigate to the second view") {
            navigateToSecondView()
        }

        then("I validate that the list of movies appear") {
            validateMoviesInTableView()
        }

        then("I navigate back to the first view") {
            navigateBackToFirstView()
        }
    }
}
```
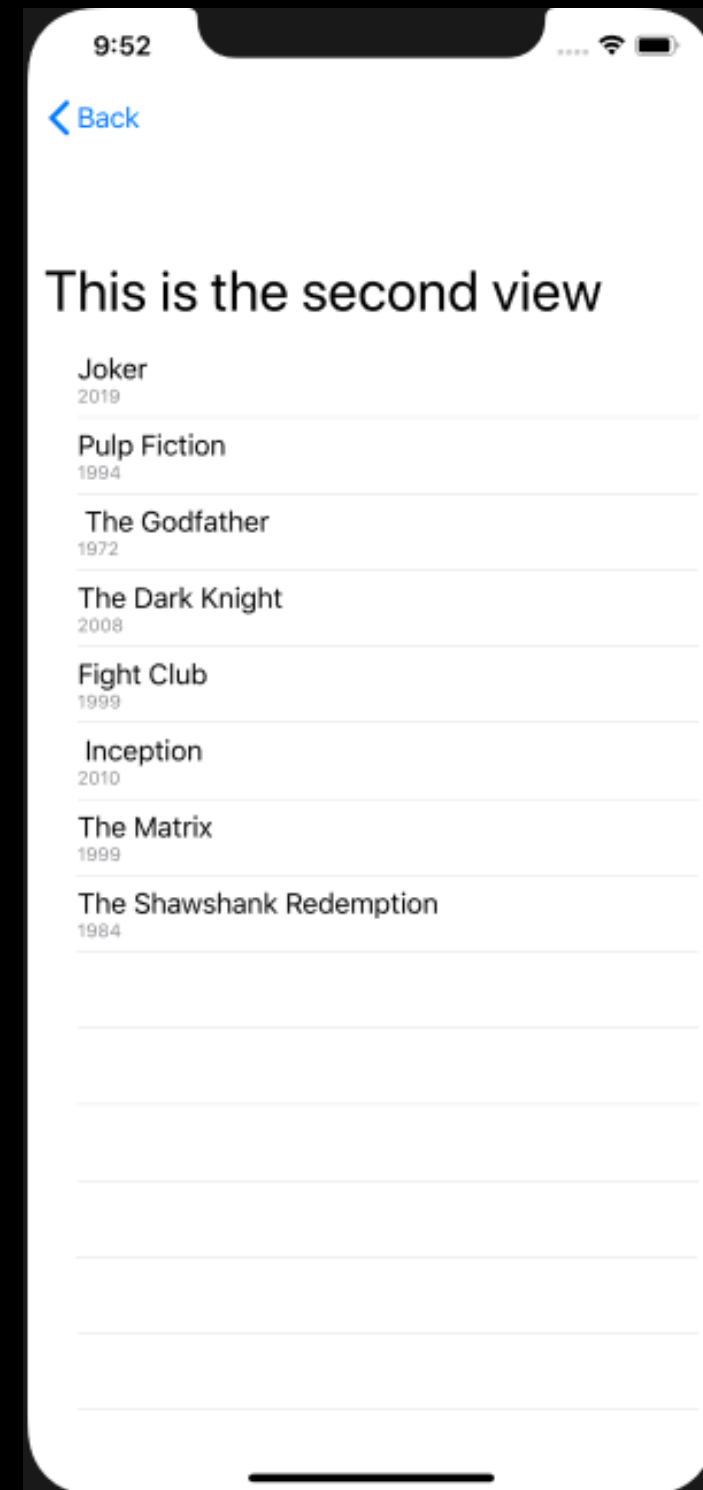
**First scenario**
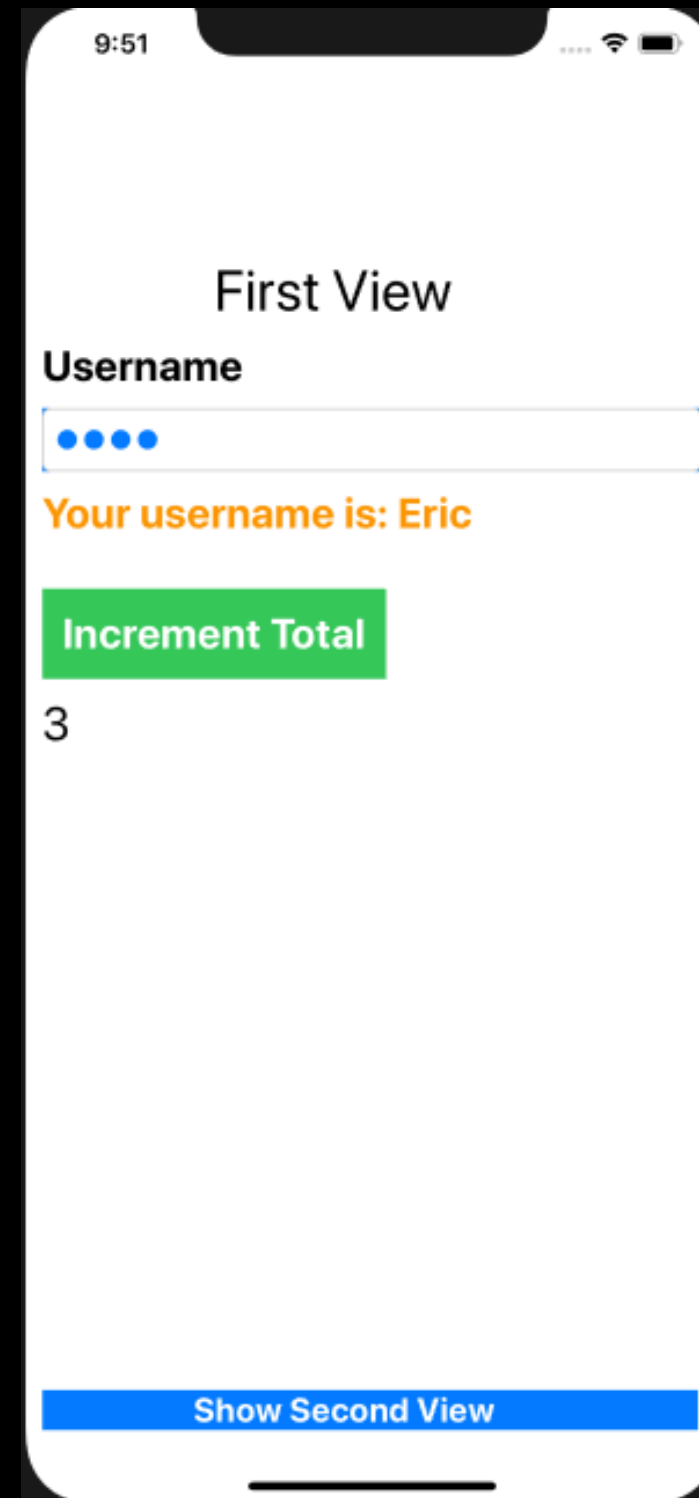
# First view test step definitions

```swift
5  //  Created by Eric Martinez on 1/17/20.
6  //  Copyright © 2020 emobile. All rights reserved.
7  //
8
9  import Foundation
10 import XCTest
11
12 extension XCTestCase {
13     func typeDefaultUsernanme() {
14         application.tapfirstViewTextField(.userName)
15         application.typeText(FirstVIewPageHelper.defaultNameEntry.rawValue)
16         application.buttons["Return"].tap() // Soft keyboard event needs its own helper
17         assertTrueStaticTexts(text: FirstVIewPageHelper.defaultNameEntryResult.rawValue)
18     }
19
20     func incrementScreenElementValidation() {
21         assertTrueStaticTexts(text: FirstVIewPageHelper.defaultNumber.rawValue)
22         assertTrueButtons(name: FirstVIewPageHelper.incrementTotal.rawValue)
23     }
24
25     func incrementNumberTest() {
26         application.tapfirstViewButton(.incrementTotal)
27         application.tapfirstViewButton(.incrementTotal)
28         application.tapfirstViewButton(.incrementTotal)
29         waiting(for: application.staticTexts["3"])
30     }
31
32     func defaultDataPersist() {
33         application.tapfirstViewButton(.showSecondView)
34         self.waitForElementToAppear(NavigationBarHelper.navBarBackButton.navBarButton(in: application.self)!)
35         application.tapNavBarButton(.navBarBackButton)
36         assertTrueStaticTexts(text: FirstVIewPageHelper.defaultNameEntryResult.rawValue)
37         waiting(for: application.staticTexts["3"])
38     }
39 }
```

# First View page helper

```
10
11  enum FirstVIewPageHelper: String {
12      case viewTitle = "                    First View"
13      case showSecondView = "                        Show Second View                  "
14      case incrementTotal = "Increment Total"
15      case defaultNumber = "0"
16      case userName = "Username"
17      case defaultNameEntry = "Eric"
18      case defaultNameEntryResult = "Your username is: Eric"
19
20      func firstViewStaticTexts(in application: XCUIApplication) -> XCUIElement? {
21          return application.staticTexts[self.rawValue]
22      }
23
24      func firstViewButton(in application: XCUIApplication) -> XCUIElement? {
25          return application.buttons[self.rawValue]
26      }
27
28      func firstViewTextField(in application: XCUIApplication) -> XCUIElement? {
29          return application.secureTextFields[self.rawValue]
30      }
31  }
32
33  extension XCUIApplication {
34      func tapfirstViewStaticTexts(_ firstViewStaticTexts: FirstVIewPageHelper) {
35          guard let element = firstViewStaticTexts.firstViewStaticTexts(in: self) else {
36              XCTFail("failed to tap static text in Sign In screen")
37              return
38          }
39          element.tap()
40      }
41
42      func tapfirstViewButton(_ firstViewButton: FirstVIewPageHelper) {
43          guard let element = firstViewButton.firstViewButton(in: self) else {
44              XCTFail("failed to tap button in Sign In screen")
45              return
46          }
47          element.tap()
48      }
49
50      func tapfirstViewTextField(_ firstViewTextField: FirstVIewPageHelper) {
51          guard let element = firstViewTextField.firstViewTextField(in: self) else {
52              XCTFail("failed to tap text field in Sign In screen")
53              return
54          }
55          element.tap()
56      }
57  }
58
```

**Navigate to Second view**

# NavBar Helper

```swift
//
//  NavigationBarHelper.swift
//  SampleUiTest
//
//  Created by Eric Martinez on 1/18/20.
//  Copyright © 2020 emobile. All rights reserved.
//

import Foundation
import XCTest

enum NavigationBarHelper: String {

    case navBarBackButton = "Back"

    func navBarButton(in application: XCUIApplication) -> XCUIElement? {
        return application.buttons[self.rawValue]
    }
}

extension XCUIApplication {

    func tapNavBarButton(_ navBarButton: NavigationBarHelper) {
        guard let element = navBarButton.navBarButton(in: self) else {
            XCTFail("failed to tap Navigation bar button")
            return
        }
        element.tap()
    }
}
```

# Second view page helper

```swift
enum SecondViewPageHelper: String {
    case secondViewTitle = "This is the second view"
    case propertyJoker = "Joker"
    case propertyJokerRelease = "2019"

    func secondViewStaticText(in application: XCUIApplication) -> XCUIElement? {
        return application.staticTexts[self.rawValue]
    }

    func secondViewTableStaticText(in application: XCUIApplication) -> XCUIElement? {
        return application.tables.staticTexts[self.rawValue]
    }
}

extension XCUIApplication {

    func tapSecondViewStaticText(_ secondViewStaticText: SecondViewPageHelper) {
        guard let element = secondViewStaticText.secondViewStaticText(in: self) else {
            XCTFail("failed to tap Static text")
            return
        }
        element.tap()
    }

    func findSecondViewTableStaticText(_ secondViewTableStaticText: SecondViewPageHelper) {
        guard let element = secondViewTableStaticText.secondViewTableStaticText(in: self) else {
            XCTFail("failed to find Table Static text")
            return
        }
        element.tap()
    }
}

func assertTrueTablesStaticText(_ movieInTable: String) {
    XCTAssertTrue(application.tables.staticTexts[movieInTable].exists)
}

func movieListInTable() {
    let moviesList = ["Joker", "Pulp Fiction", " The Godfather ", "The Dark Knight ",
                      "Fight Club", " Inception", "The Matrix ", "The Shawshank Redemption "]
    print(moviesList)
    for movie in moviesList {
        assertTrueTablesStaticText(movie)
    }
}
```

# Second view test step definitions

```swift
4   //
5   //  Created by Eric Martinez on 1/18/20.
6   //  Copyright © 2020 emobile. All rights reserved.
7   //
8
9   import Foundation
10  import XCTest
11
12  extension XCTestCase {
13
14      func firstViewValidation() {
15          self.waitForElementToAppear(FirstVIewPageHelper.showSecondView.firstViewButton(in: application.self)!)
16      }
17
18      func navigateToSecondView() {
19          application.tapfirstViewButton(.showSecondView)
20          self.waitForElementToAppear(NavigationBarHelper.navBarBackButton.navBarButton(in: application.self)!)
21          assertTrueStaticTexts(text: SecondViewPageHelper.secondViewTitle.rawValue)
22      }
23
24      func validateMoviesInTableView() {
25          movieListInTable()
26          application.findSecondViewTableStaticText(.propertyJoker)
27          application.findSecondViewTableStaticText(.propertyJokerRelease)
28      }
29
30      func navigateBackToFirstView() {
31          application.tapNavBarButton(.navBarBackButton)
32          self.waitForElementToAppear(FirstVIewPageHelper.showSecondView.firstViewButton(in: application.self)!)
33      }
34  }
35
```

# HTML Report

# HTML report test journey screenshots

# Directory tree



```
▼ 📁 SampleUiTest
    ▼ 📁 StepDefinitions
        📄 TestIncrementNumber.swift
        📄 NavigateToSecondVIew.swift
    ▼ 📁 PageHelpers
        📄 SecondVIewPageHelper.swift
        📄 NavigationBarHelper.swift
        📄 FirstVIewPageHelper.swift
    ▼ 📁 Support
        📄 SampleXCUITestBaseClass.swift
        📄 XCUITest+GerkinsFormatters.swift
        📄 TestWaitHelper.swift
        📄 AssertionHelper.swift
        📄 SampleXCUITests-Bridging-Header.h
    📄 MartinezXCUIAsmokeTest.swift
```

# Makefile terminal commands

```
94  test-ui:
95      rm -rf TestResults
96      $(XCODEBUILDCMD) test -scheme "SampleUiTest" \
97       -sdk iphonesimulator \
98       -destination '${SIMULATOR_UITEST}' \
99       -resultBundlePath TestResults
100
101 ui-test-report:
102     xcrun simctl shutdown all
103     xchtmlreport -r TestResults
104     echo "Your Test Report test report is launching on our default browser"
105     open index.html
106
```