Swift

Jeremy Day
Steven Dao
Nicole Lazar

Overview

- Swift is primarily used to create applications for Apple’s iOS platform, as well as the OS X operating system.
- Swift is a multi-paradigm language. It features: the protocol-oriented programming paradigm, object-oriented programming, functional programming, imperative programming, and block structuring.
- Swift was introduced as an alternative to using Objective-C for iOS development. According to the June 2015 RedMonk Programming Language rankings, Objective-C is still more popular than swift (#10 vs #18), but Swift’s ranking is still impressive due to the fact it was only announced in June of last year.

History of Swift

- Swift was first announced by Apple at WWDC on June 2, 2014.
- Apple also released their manual on Swift, *The Swift Programming Language* at the same time.
- Swift left beta on September 9th, 2014. Several new iterations of the version have been released since then, with Swift 2.0 being announced at WWDC this year.
- Development of Swift was started in 2010 by Chris Lattner, but many engineers at Apple had a part in Swift's development.
- According to Chris Lattner, Swift draws ideas from "Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and far too many others to list."

Problem Domain

- Swift’s primary use is in building applications for iOS, which is the operating system on most Apple’s mobile devices including the iPhone and iPad. Programs written in Swift will also work on watchOS and tvOS, the operating systems for Apple Watch and Apple Tv, respectively.
- Swift started out as a proprietary language, but will transition into open source later this year. After its transition to open source, Swift applications will be able to run on OS X and Linux as well.

Language Design & Concepts

Objective-C Interoperability:

- Because Swift still has access to Objective-C libraries and APIs, the two languages can co-exist together.

Syntactic sugar:

- Use the `var` keyword to define a variable
- Use the `let` keyword to define a constant

Type Inference:

- Infers the type based on the value

Java	Swift

`String x = “Hello”;`	`var x = “Hello”`
	`var x: String = “Hello”`

Language Design & Concepts

Optionals:

- This new feature in Swift allows references or values to operate in a manner similar to the common pattern in C, where a pointer may refer to a value or may be null.
  - This implies that non-optional types cannot result in a null-pointer error; the compiler can ensure this is not possible.
- To indicate that a variable is optional, the question mark operator is placed after the variable name.
  - The exclamation point operator is used to force unwrap the optional, only if it is known to not be empty or null.

```
var age: Int? = nil
var height: Int? = 180
```

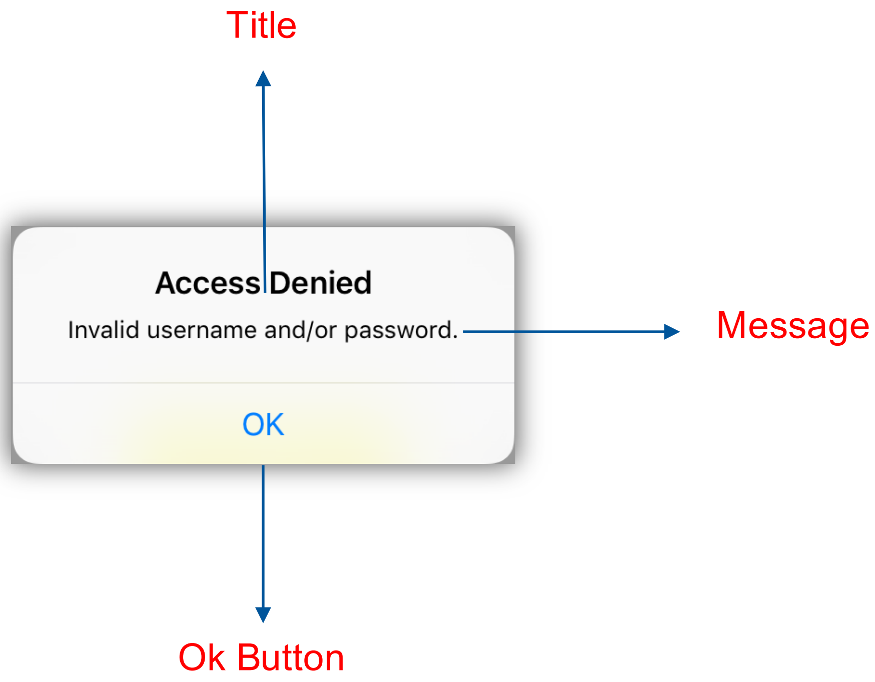
Language Design & Concepts

Protocols:

- Also a feature of Objective-C, protocols are used to ensure that a class implements a set of methods, just like interfaces do in other languages.

Tuples:

- Because Swift contains advanced types, such as tuples, it allows for functions to return more than one value.
- Swift also supports pattern matching in switch statements.



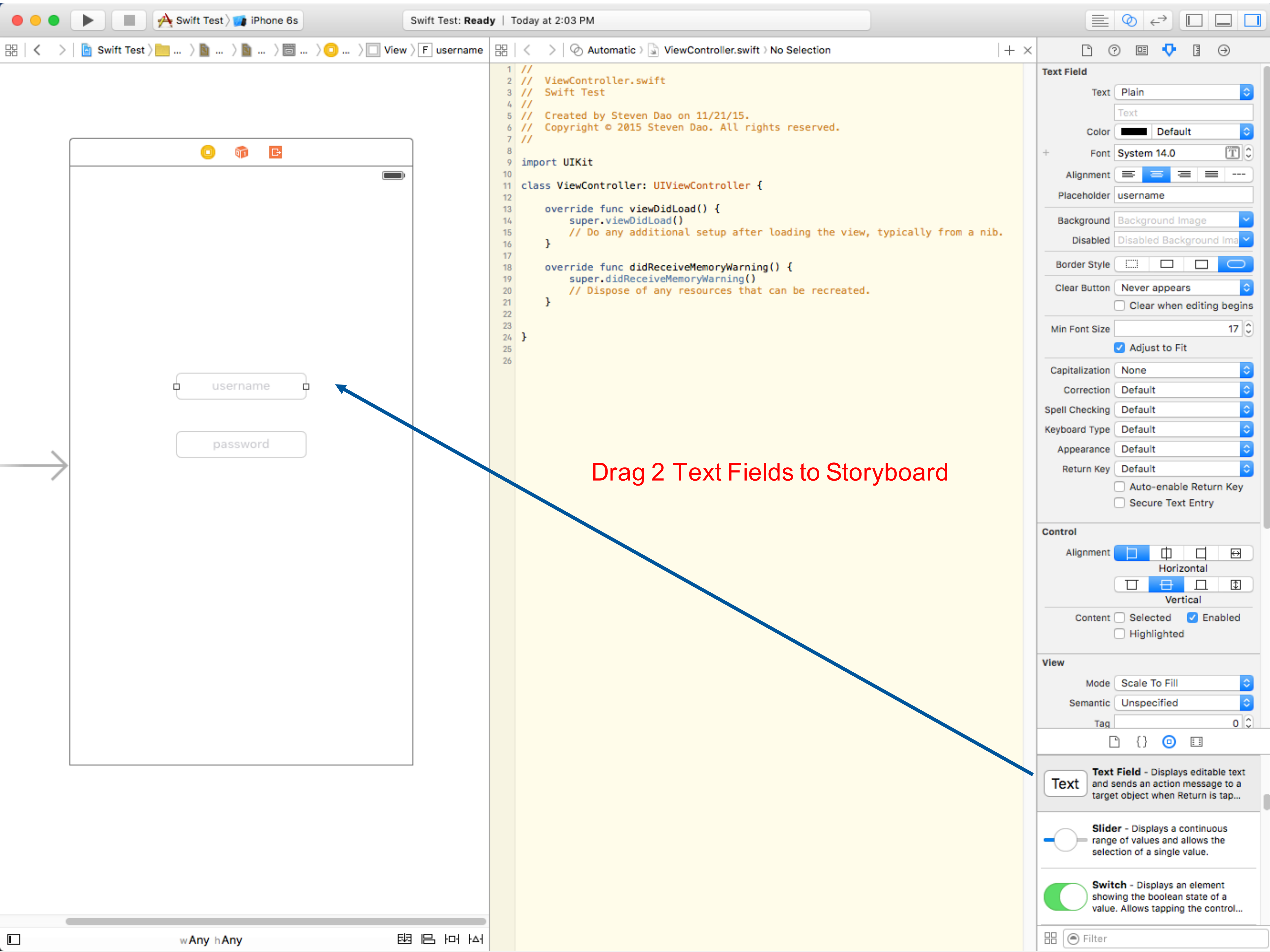
Title

Access Denied

Invalid username and/or password.

Message

Ok Button


Swift Test 〉 iPhone 6s
Swift Test: Ready | Today at 2:03 PM
Swift Test 〉 ... 〉 ... 〉 ... 〉 ... 〉 ... 〉 View 〉 F username
Automatic 〉 ViewController.swift 〉 No Selection
username
password
wAny hAny

```
//
//  ViewController.swift
//  Swift Test
//
//  Created by Steven Dao on 11/21/15.
//  Copyright © 2015 Steven Dao. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

Drag 2 Text Fields to Storyboard

Text Field

Text: Plain
Text
Color: Default
Font: System 14.0
Alignment
Placeholder: username
Background: Background Image
Disabled: Disabled Background Ima
Border Style
Clear Button: Never appears
□ Clear when editing begins
Min Font Size: 17
☑ Adjust to Fit
Capitalization: None
Correction: Default
Spell Checking: Default
Keyboard Type: Default
Appearance: Default
Return Key: Default
□ Auto-enable Return Key
□ Secure Text Entry

Control

Alignment: Horizontal
Vertical
Content: □ Selected ☑ Enabled
□ Highlighted

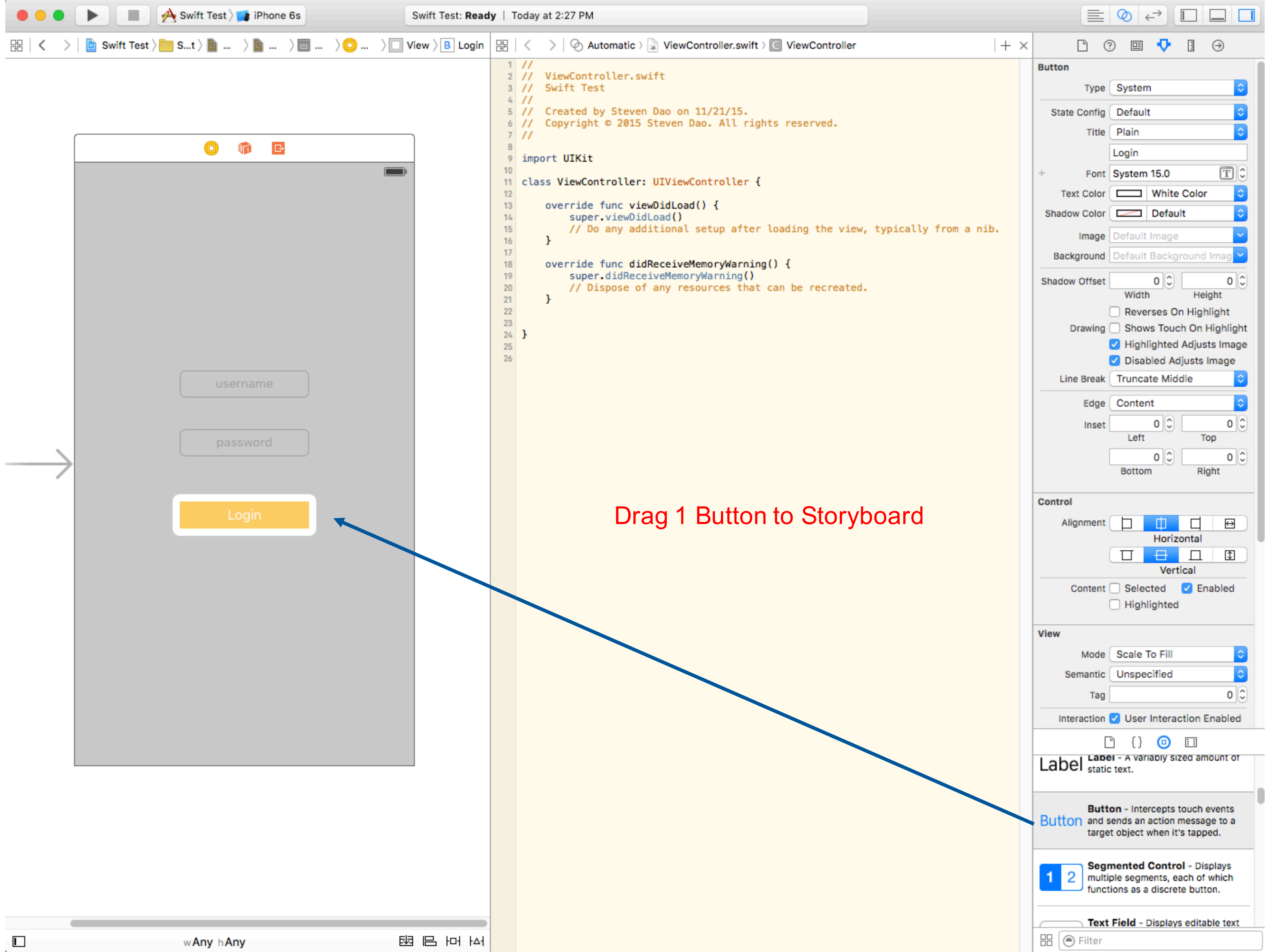
View

Mode: Scale To Fill
Semantic: Unspecified
Tag: 0

Text Field - Displays editable text and sends an action message to a target object when Return is tap...

Slider - Displays a continuous range of values and allows the selection of a single value.

Switch - Displays an element showing the boolean state of a value. Allows tapping the control...



Swift Test 〉 iPhone 6s
Swift Test: **Ready** | Today at 2:27 PM
Swift Test 〉 S...t 〉 ... 〉 ... 〉 ... 〉 ... 〉 View 〉 Login
Automatic 〉 ViewController.swift 〉 ViewController
username
password
Login
wAny hAny

```
//
//  ViewController.swift
//  Swift Test
//
//  Created by Steven Dao on 11/21/15.
//  Copyright © 2015 Steven Dao. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

Drag 1 Button to Storyboard

Button

Type: System
State Config: Default
Title: Plain
Login
Font: System 15.0
Text Color: White Color
Shadow Color: Default
Image: Default Image
Background: Default Background Imag
Shadow Offset: 0 Width, 0 Height
Drawing: □ Reverses On Highlight
□ Shows Touch On Highlight
☑ Highlighted Adjusts Image
☑ Disabled Adjusts Image
Line Break: Truncate Middle
Edge: Content
Inset: 0 Left, 0 Top, 0 Bottom, 0 Right

Control

Alignment: Horizontal, Vertical
Content: □ Selected ☑ Enabled
□ Highlighted

View

Mode: Scale To Fill
Semantic: Unspecified
Tag: 0
Interaction: ☑ User Interaction Enabled

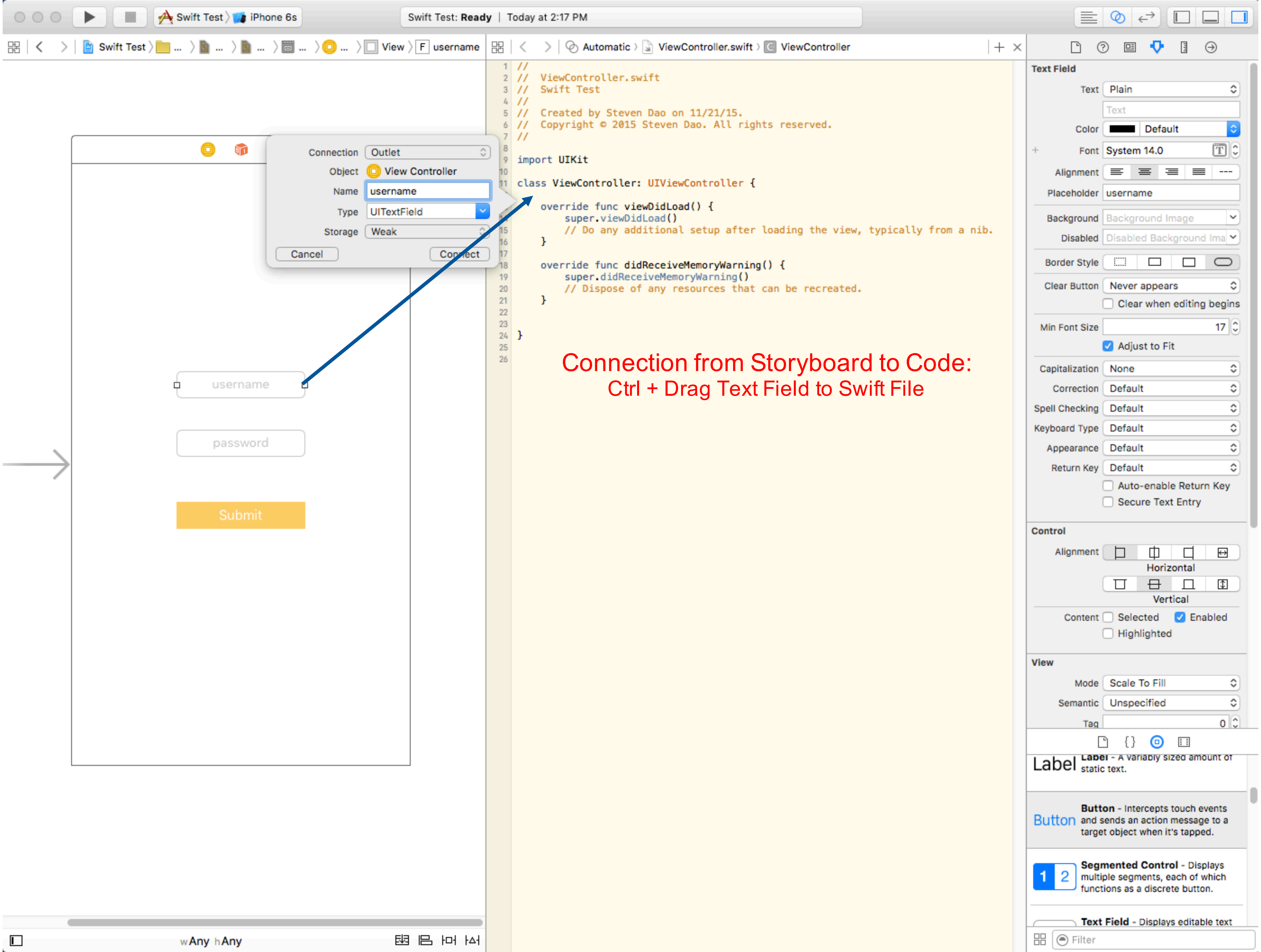
Label **Label** - A variably sized amount of static text.

Button **Button** - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 **Segmented Control** - Displays multiple segments, each of which functions as a discrete button.

Text Field - Displays editable text

Filter



Swift Test: **Ready** | Today at 2:17 PM

Swift Test 〉 iPhone 6s

Swift Test 〉 ... 〉 ... 〉 ... 〉 ... 〉 ... 〉 View 〉 F username

Automatic 〉 ViewController.swift 〉 C ViewController

Connection	Outlet
Object	View Controller
Name	username
Type	UITextField
Storage	Weak

Cancel Connect

```
//
//  ViewController.swift
//  Swift Test
//
//  Created by Steven Dao on 11/21/15.
//  Copyright © 2015 Steven Dao. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

Connection from Storyboard to Code:
Ctrl + Drag Text Field to Swift File

Text Field

Text	Plain
	Text
Color	Default
Font	System 14.0
Alignment	
Placeholder	username
Background	Background Image
Disabled	Disabled Background Ima
Border Style	
Clear Button	Never appears
	□ Clear when editing begins
Min Font Size	17
	☑ Adjust to Fit
Capitalization	None
Correction	Default
Spell Checking	Default
Keyboard Type	Default
Appearance	Default
Return Key	Default
	□ Auto-enable Return Key
	□ Secure Text Entry

Control

Alignment	Horizontal
	Vertical
Content	□ Selected ☑ Enabled
	□ Highlighted

View

Mode	Scale To Fill
Semantic	Unspecified
Tag	0

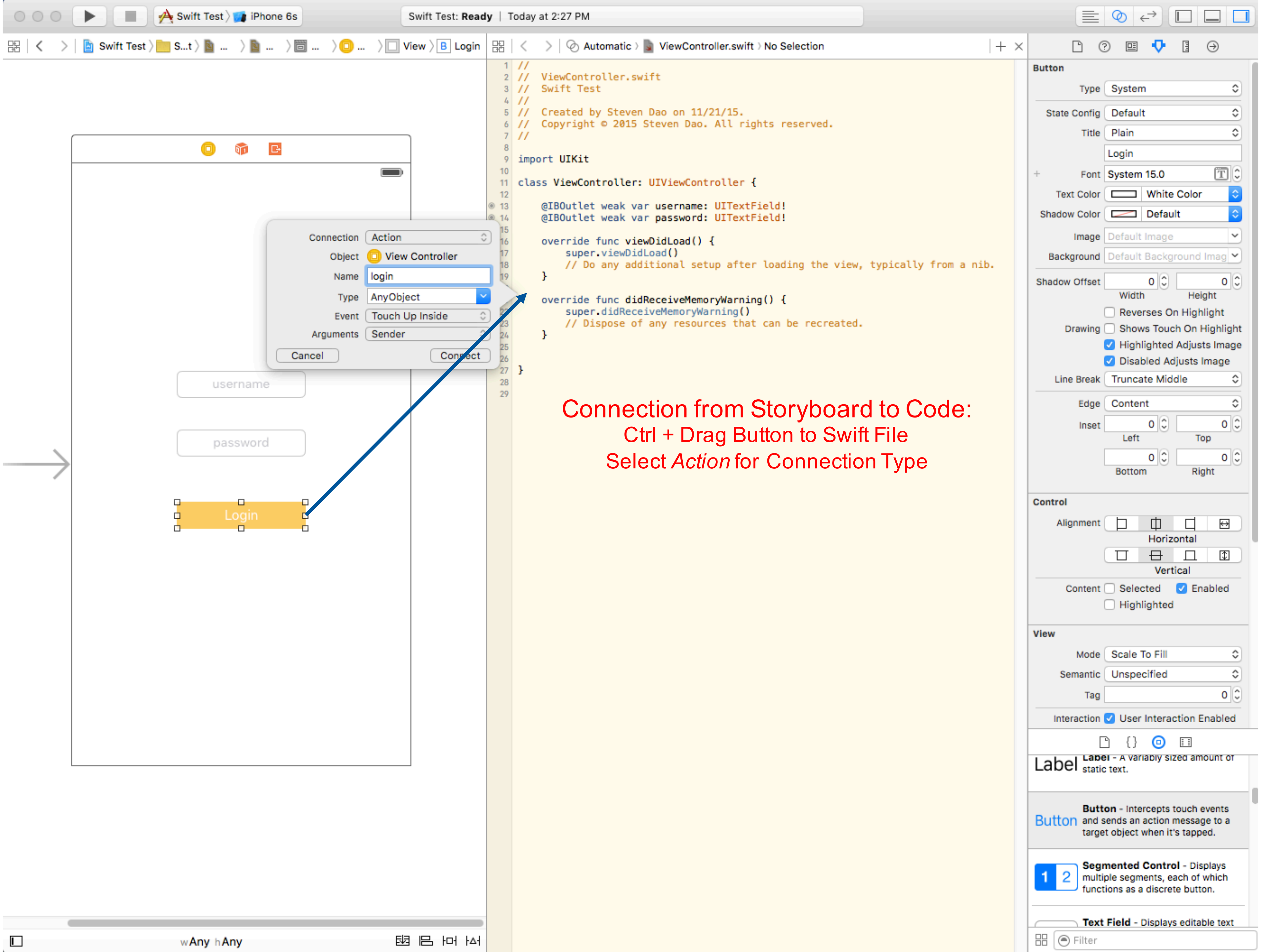
Label **Label** - A variably sized amount of static text.

Button **Button** - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 **Segmented Control** - Displays multiple segments, each of which functions as a discrete button.

Text Field - Displays editable text

Filter


Swift Test
iPhone 6s
Swift Test: Ready | Today at 2:27 PM
Swift Test › S...t › ... › ... › ... › ... › View › Login
Automatic › ViewController.swift › No Selection
username
password
Login
wAny hAny

```
//
//  ViewController.swift
//  Swift Test
//
//  Created by Steven Dao on 11/21/15.
//  Copyright © 2015 Steven Dao. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var username: UITextField!
    @IBOutlet weak var password: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

Connection: Action
Object: View Controller
Name: login
Type: AnyObject
Event: Touch Up Inside
Arguments: Sender

Cancel | Connect

Connection from Storyboard to Code:
Ctrl + Drag Button to Swift File
Select *Action* for Connection Type

Button

Type: System
State Config: Default
Title: Plain
Login
Font: System 15.0
Text Color: White Color
Shadow Color: Default
Image: Default Image
Background: Default Background Imag
Shadow Offset: 0 (Width) 0 (Height)
Drawing: □ Reverses On Highlight
□ Shows Touch On Highlight
☑ Highlighted Adjusts Image
☑ Disabled Adjusts Image
Line Break: Truncate Middle
Edge: Content
Inset: 0 (Left) 0 (Top)
0 (Bottom) 0 (Right)

Control

Alignment: Horizontal
Vertical
Content: □ Selected ☑ Enabled
□ Highlighted

View

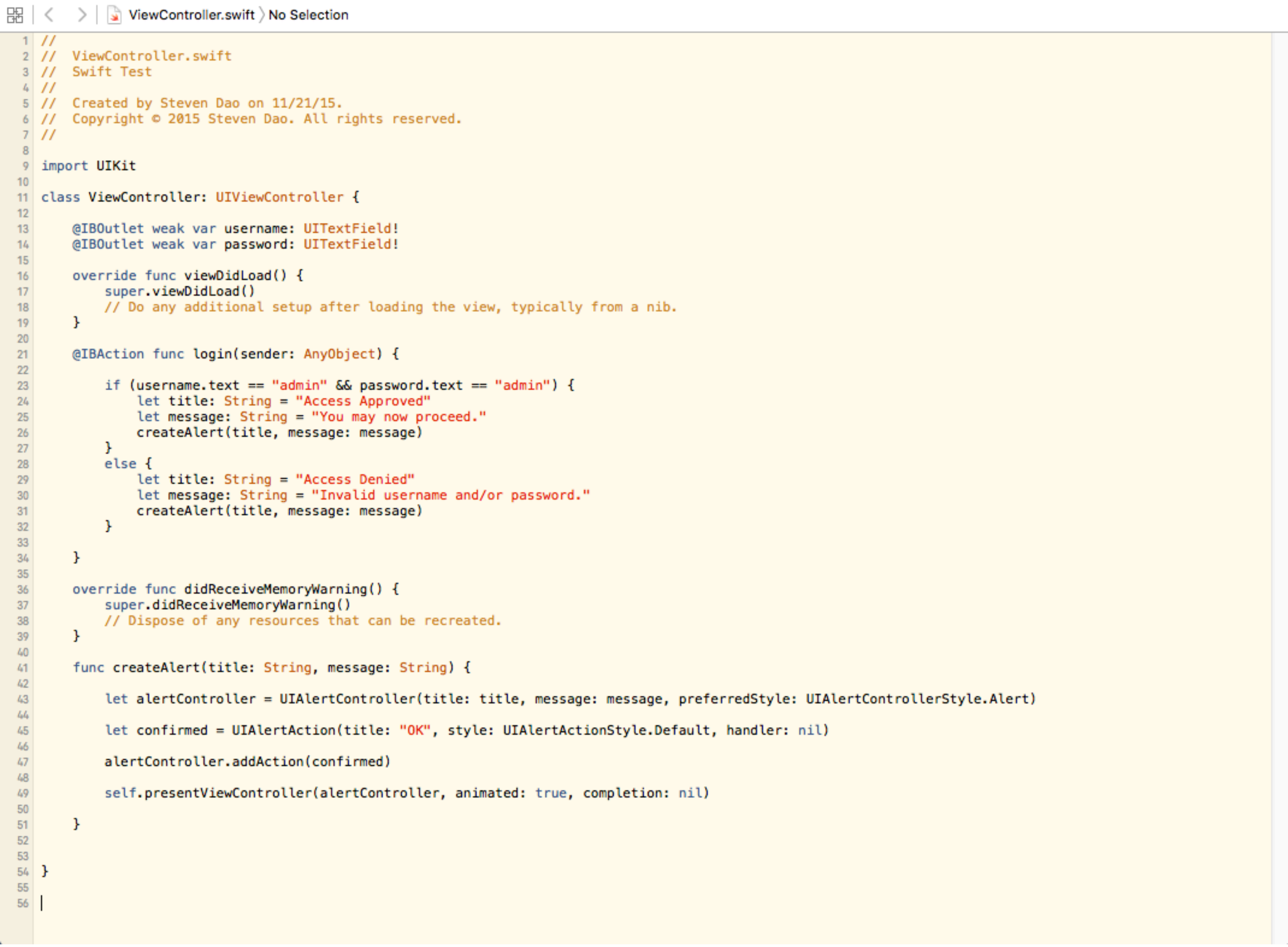
Mode: Scale To Fill
Semantic: Unspecified
Tag: 0
Interaction: ☑ User Interaction Enabled

Label **Label** - A variably sized amount of static text.

Button **Button** - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 **Segmented Control** - Displays multiple segments, each of which functions as a discrete button.

Text Field - Displays editable text



```swift
//
//  ViewController.swift
//  Swift Test
//
//  Created by Steven Dao on 11/21/15.
//  Copyright © 2015 Steven Dao. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var username: UITextField!
    @IBOutlet weak var password: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    @IBAction func login(sender: AnyObject) {

        if (username.text == "admin" && password.text == "admin") {
            let title: String = "Access Approved"
            let message: String = "You may now proceed."
            createAlert(title, message: message)
        }
        else {
            let title: String = "Access Denied"
            let message: String = "Invalid username and/or password."
            createAlert(title, message: message)
        }

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func createAlert(title: String, message: String) {

        let alertController = UIAlertController(title: title, message: message, preferredStyle: UIAlertControllerStyle.Alert)

        let confirmed = UIAlertAction(title: "OK", style: UIAlertActionStyle.Default, handler: nil)

        alertController.addAction(confirmed)

        self.presentViewController(alertController, animated: true, completion: nil)

    }


}

```

Comparison (Objective-C)

- Swift is easier to read

var x = 1 var y = "Hello”	var x: Int x = 2

  - No semicolons or parentheses for if-else statements

- Swift requires less code

Objective-C	Swift

NSString *person= @”John";
int age= 21;
NSString *s = [NSString stringWithFormat:@”%@ is %d years old", person, age];
	let person = ”John"
	let age = 21
	let s = “\(person) is \(age) years old.”

- Swift is faster and safer

  - Optional typing

References

- http://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html
- http://www.toptal.com/swift/from-objective-c-to-swift
- http://nondot.org/sabre/
- http://www.apple.com/live/2015-june-event/
- https://developer.apple.com/swift/
- https://redmonk.com/sogrady/2015/07/01/language-rankings-6-15/
- http://www.touch-code-magazine.com/swift-optionals-use-let/

Swift

Jeremy Day
Steven Dao
Nicole Lazar