What's New in Accessibility

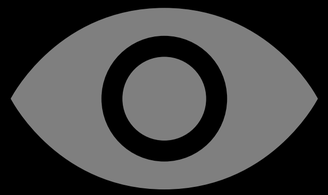
Session 215

Skylar Peterson, Software Engineer

Making technology usable by everyone

Cognitive

Motor

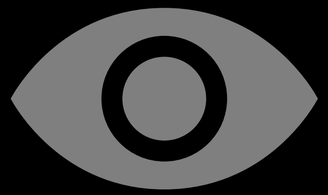


Vision

Hearing

Cognitive

Motor

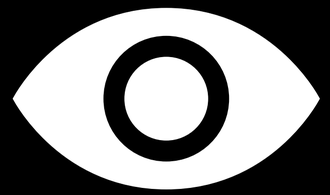


Vision

Hearing

Cognitive

Motor

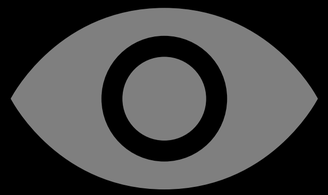


Vision

Hearing

Cognitive

Motor



Vision

Hearing

Over 1 Billion

People have a disability worldwide

http://www.who.int/disabilities/en/

1 in 7

People have a disability worldwide

http://www.who.int/disabilities/en/

VoiceOver Reduce transparency On/Off labels Grayscale Audio descriptions

Word prediction Mono audio AssistiveTouch Button shapes Speak screen

Gliding cursor speed Subtitles Captioning Closed captions

Switch control Larger text Speech Increase contrast

Invert colors Cursor color Dictation Siri Bold text

Larger cursor Zoom Reduce motion Hearing aids Sticky keys Gestures

Auto scanning Safari reader Guided access Slow Keys Touch accommodations


HumanWare
Brailliant

New assistive features

New assistive features

Auditing your app for accessibility

New assistive features

Auditing your app for accessibility

Accessibility API basics

New assistive features

Auditing your app for accessibility

Accessibility API basics

Beyond the basics

New assistive features

Auditing your app for accessibility

Accessibility API basics

Beyond the basics

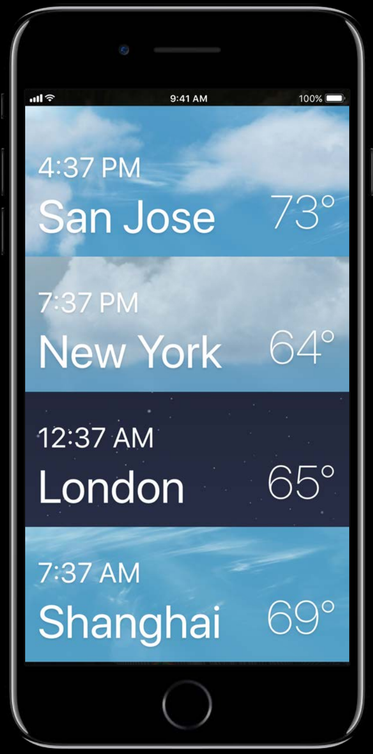
Drag and Drop accessibility

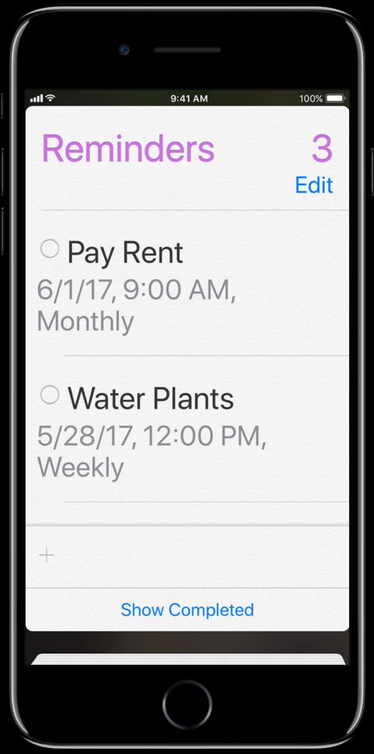
New Assistive Features

Text Detection

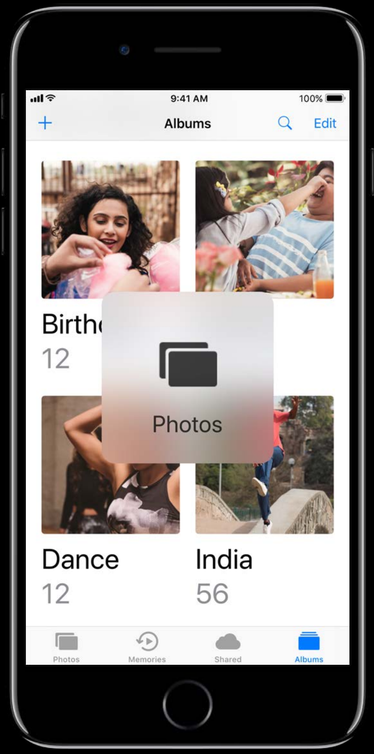
Improved Photo Descriptions

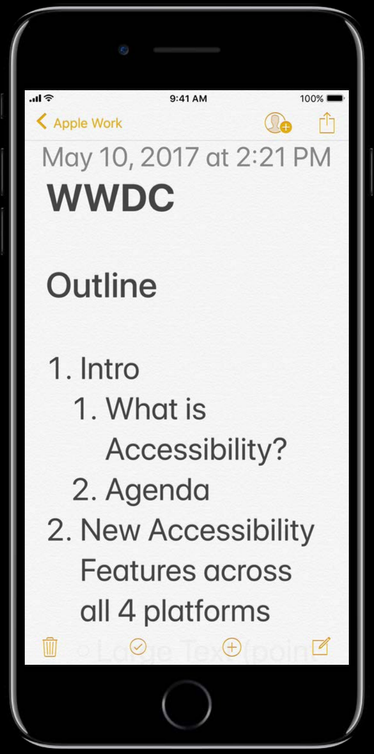
Large Text


9:41 AM
100%
4:37 PM
San Jose
73°
7:37 PM
New York
64°
12:37 AM
London
65°
7:37 AM
Shanghai
69°


9:41 AM
100%
Reminders
3
Edit
Pay Rent
6/1/17, 9:00 AM, Monthly
Water Plants
5/28/17, 12:00 PM, Weekly
Show Completed


9:41 AM
100%
9:41
Wed, Jun 7
Press home to unlock

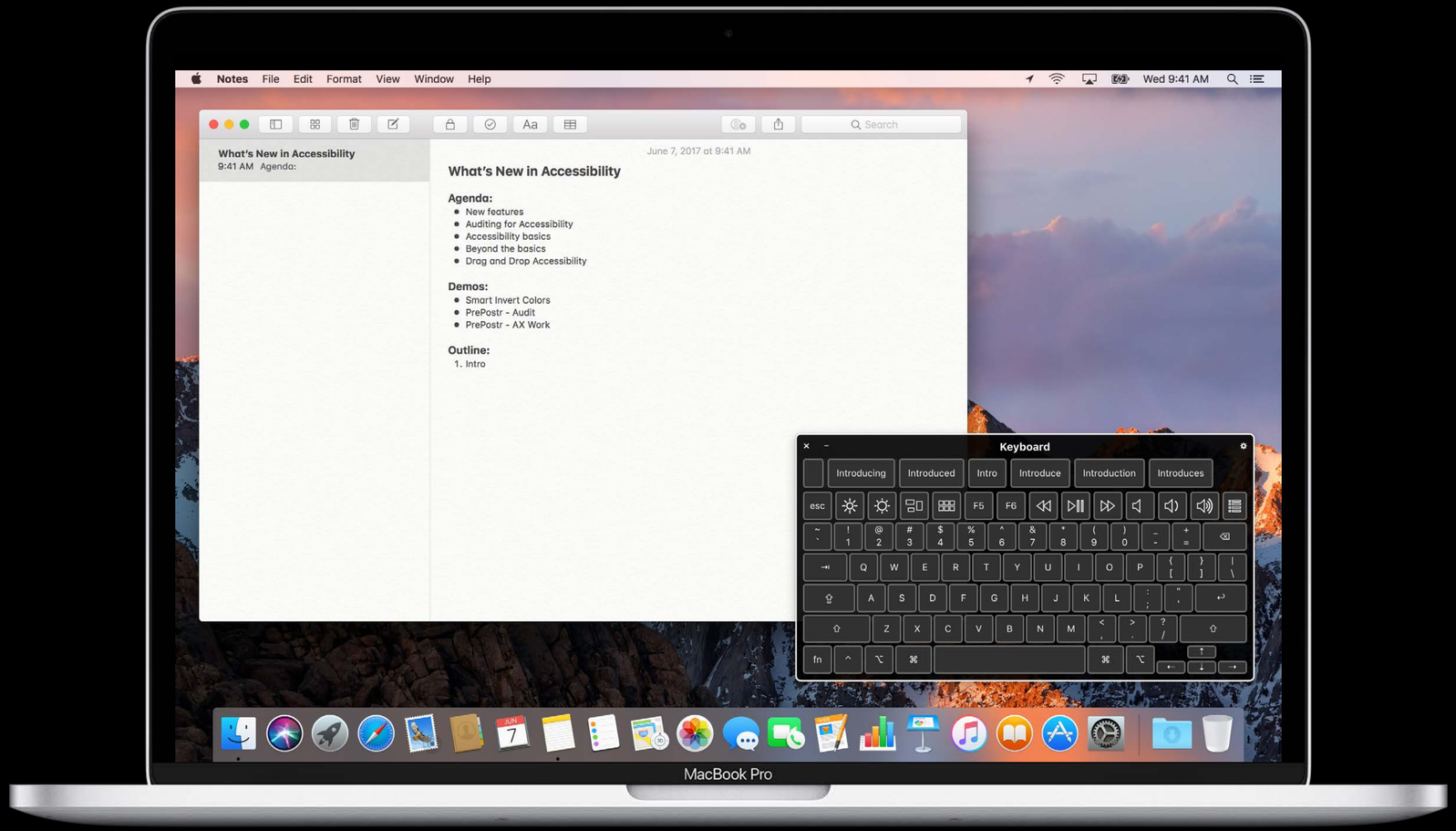

9:41 AM
100%
Albums
Edit
Photos
Dance
12
India
56
Photos
Memories
Shared
Albums


9:41 AM
100%
Apple Work
May 10, 2017 at 2:21 PM
WWDC
Outline
1. Intro
1. What is Accessibility?
2. Agenda
2. New Accessibility Features across all 4 platforms

Related Sessions

Design for Everyone	Hall 3	Thursday 11:00AM
Auto Layout Techniques in Interface Builder	Hall 3	Friday 9:00AM
Building Apps with Dynamic Type	Executive Ballroom	Friday 1:50PM

Accessibility Keyboard



Type to Siri


What can I help you with?
Type to Siri
I
Yeah
You
Finder File Edit View Go Window Help
Wed 9:41 AM
What can I help you with?
Type to Siri
MacBook Pro

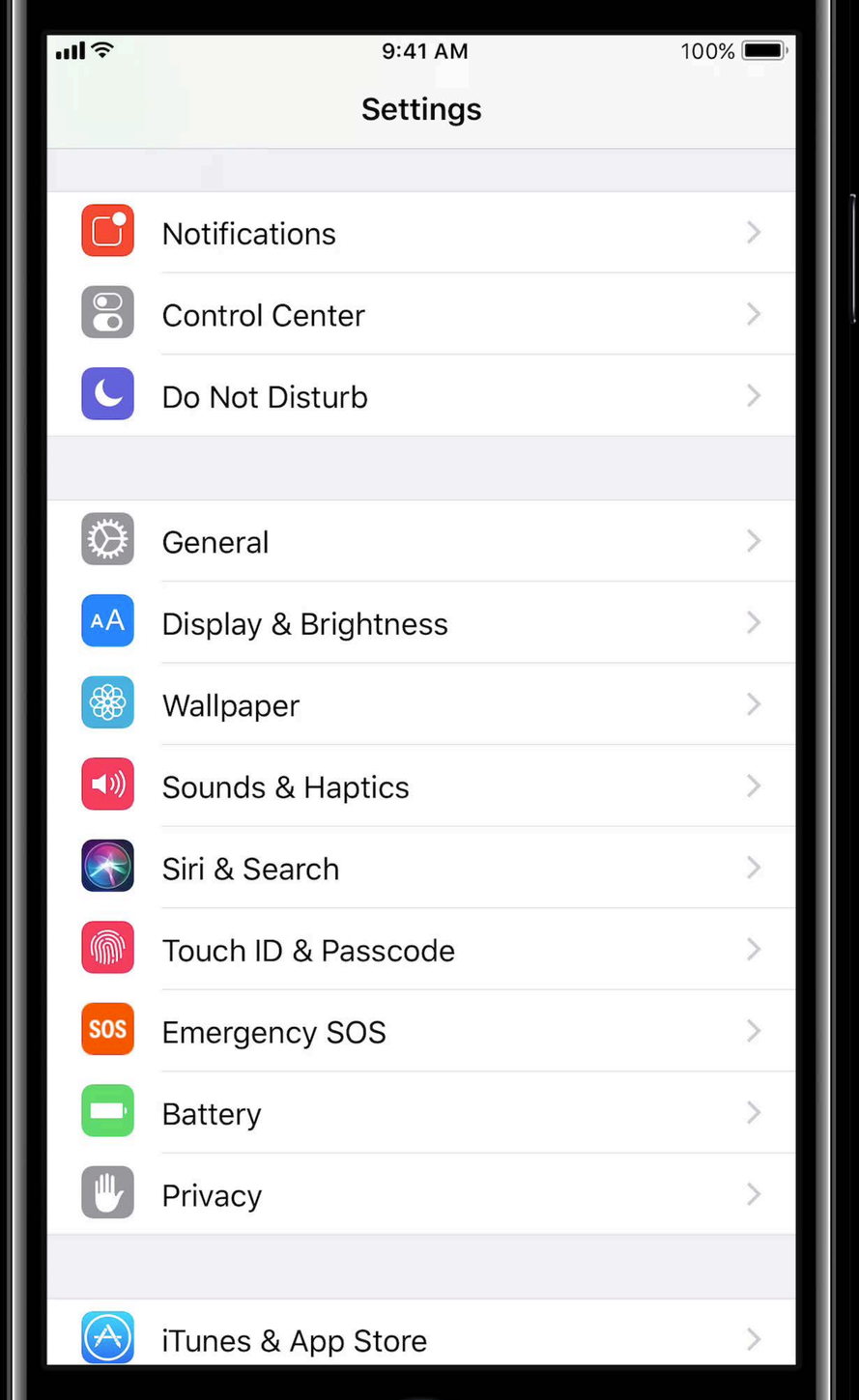
Smart Invert Colors

Demo

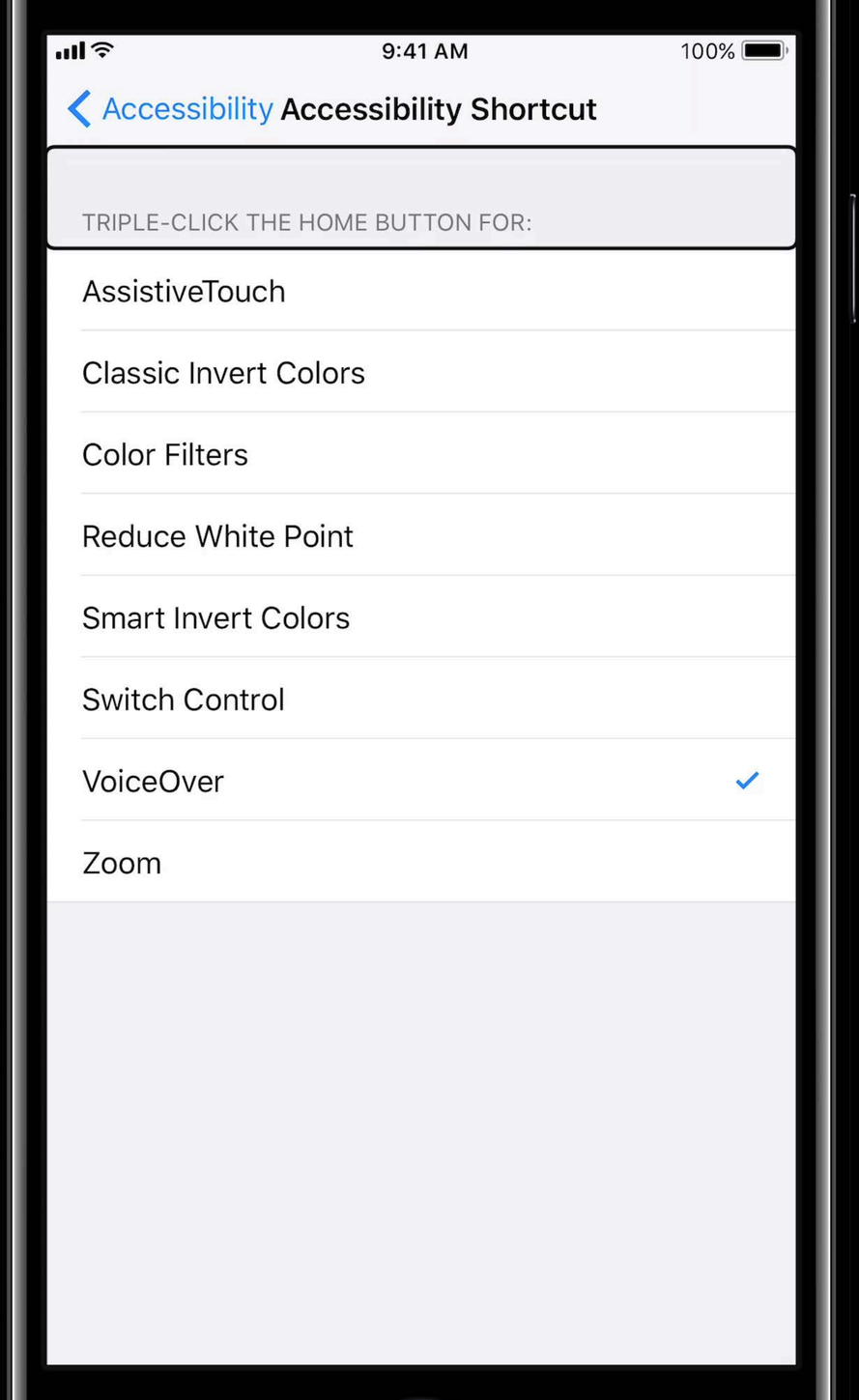
Smart Invert Colors

Auditing Your App for Accessibility

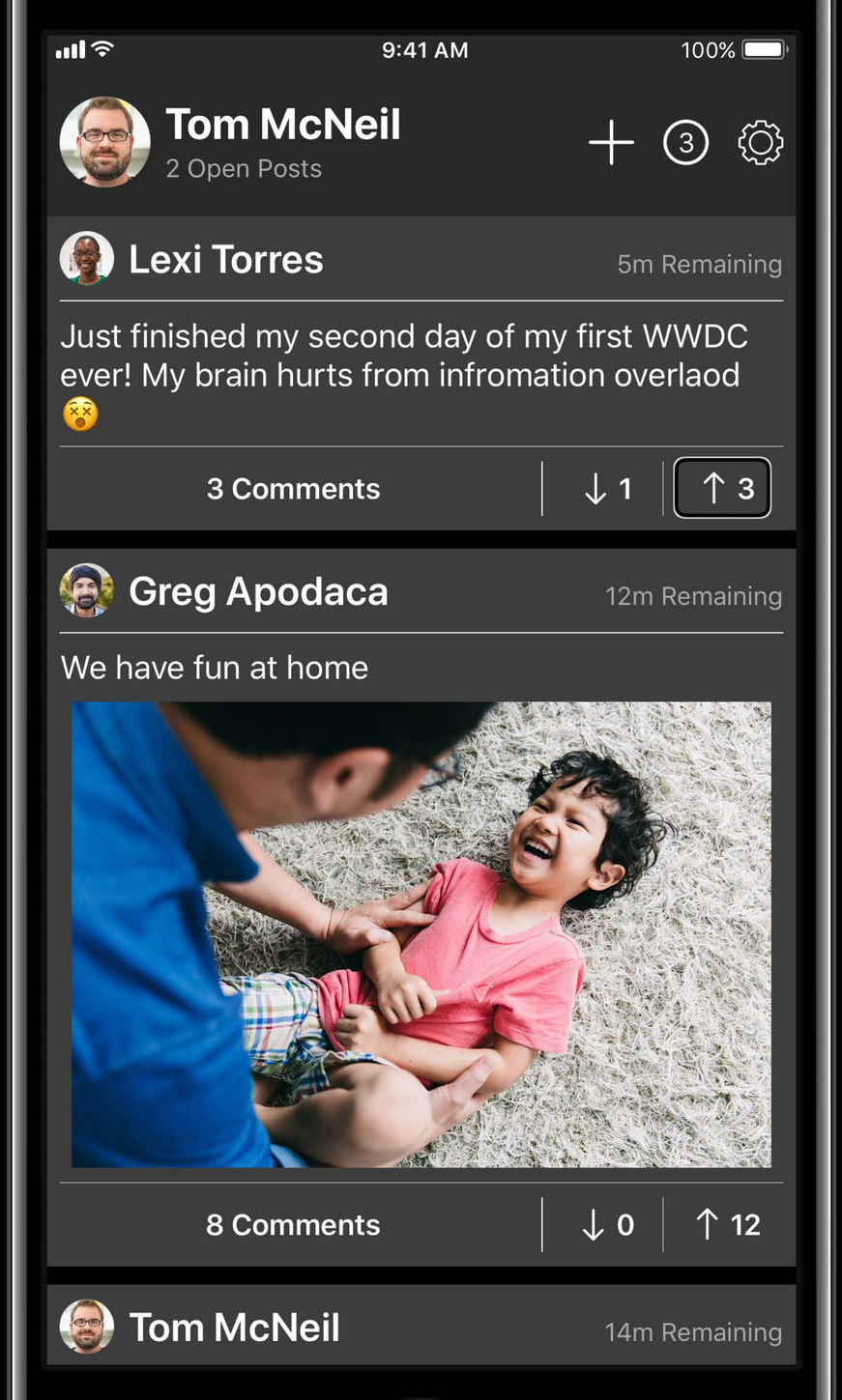
Auditing


9:41 AM
100%
Settings
Notifications
Control Center
Do Not Disturb
General
Display & Brightness
Wallpaper
Sounds & Haptics
Siri & Search
Touch ID & Passcode
Emergency SOS
Battery
Privacy
iTunes & App Store

Auditing


9:41 AM
100%
Accessibility
Accessibility Shortcut
TRIPLE-CLICK THE HOME BUTTON FOR:
AssistiveTouch
Classic Invert Colors
Color Filters
Reduce White Point
Smart Invert Colors
Switch Control
VoiceOver
Zoom

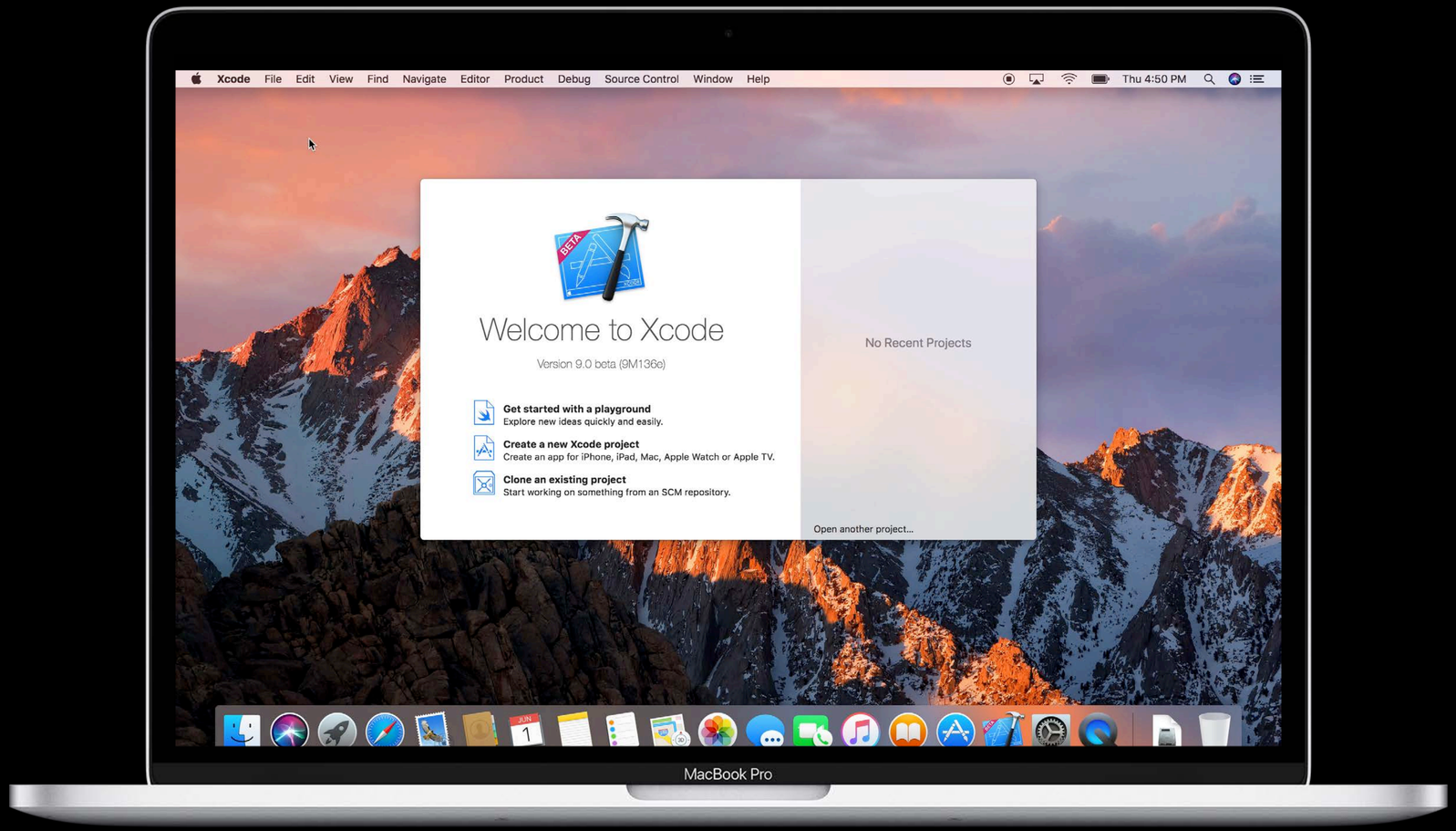
Auditing


9:41 AM
100%
Tom McNeil
2 Open Posts
3
Lexi Torres
5m Remaining
Just finished my second day of my first WWDC ever! My brain hurts from infromation overlaod
3 Comments
1
3
Greg Apodaca
12m Remaining
We have fun at home
8 Comments
0
12
Tom McNeil
14m Remaining

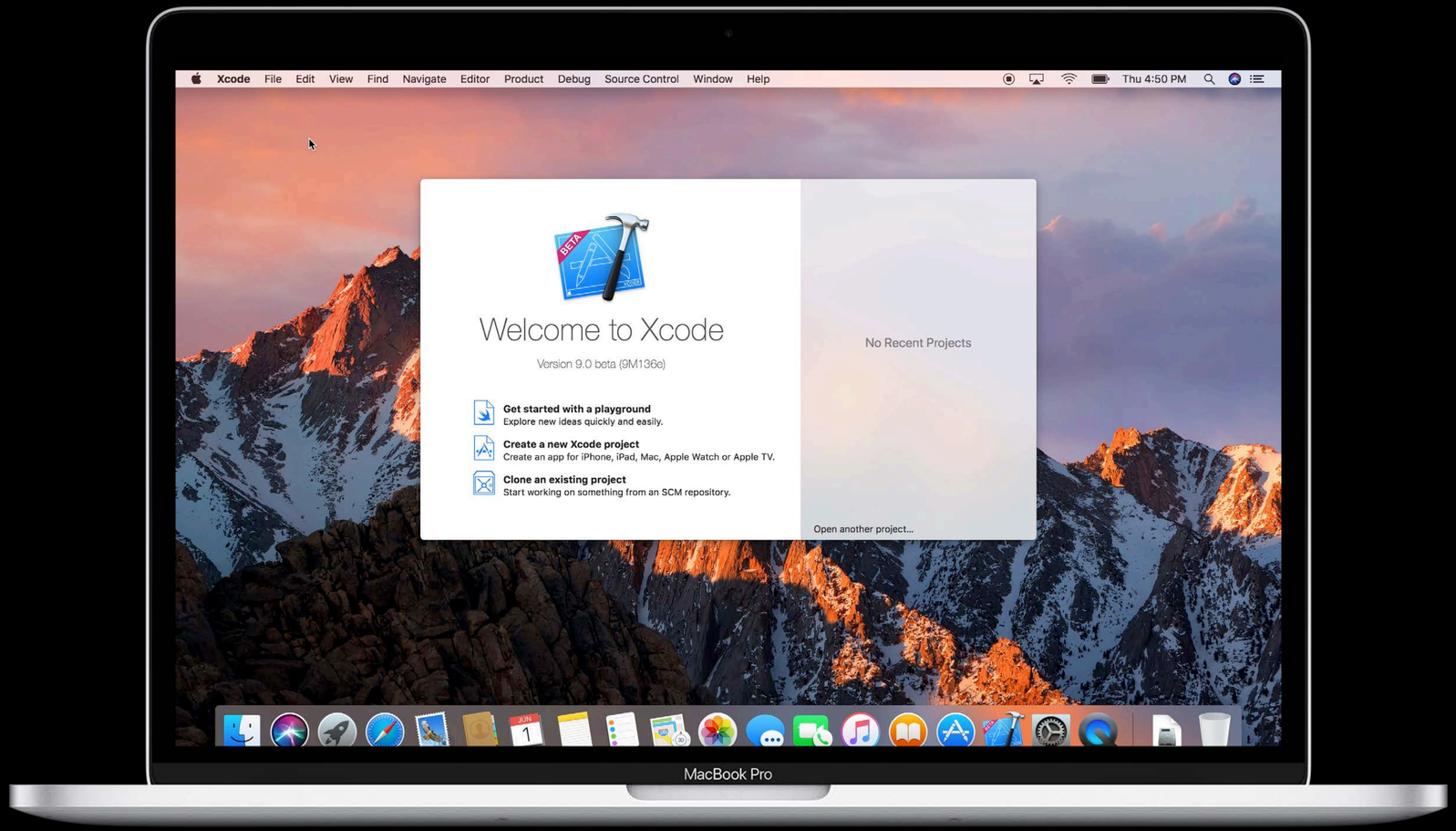
Auditing

"Up vote, 3 votes, Button"

Accessibility Inspector



Accessibility Inspector



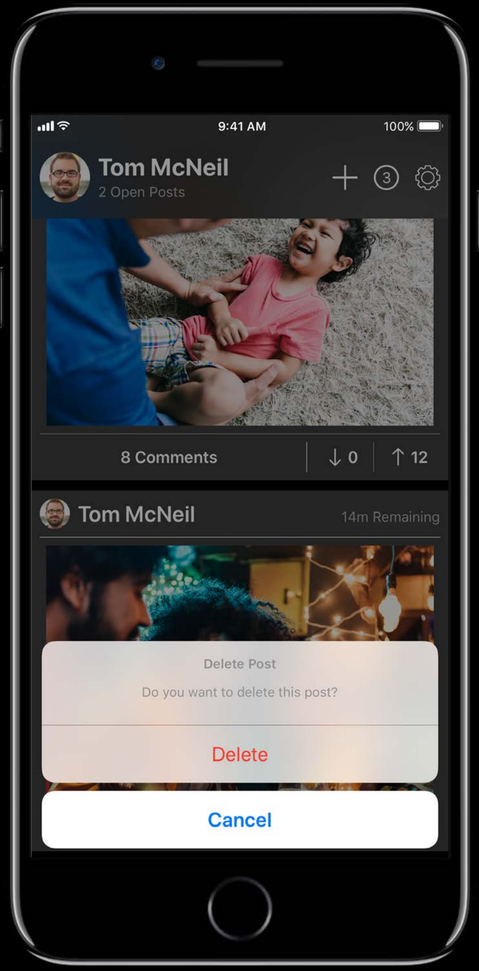
More Information

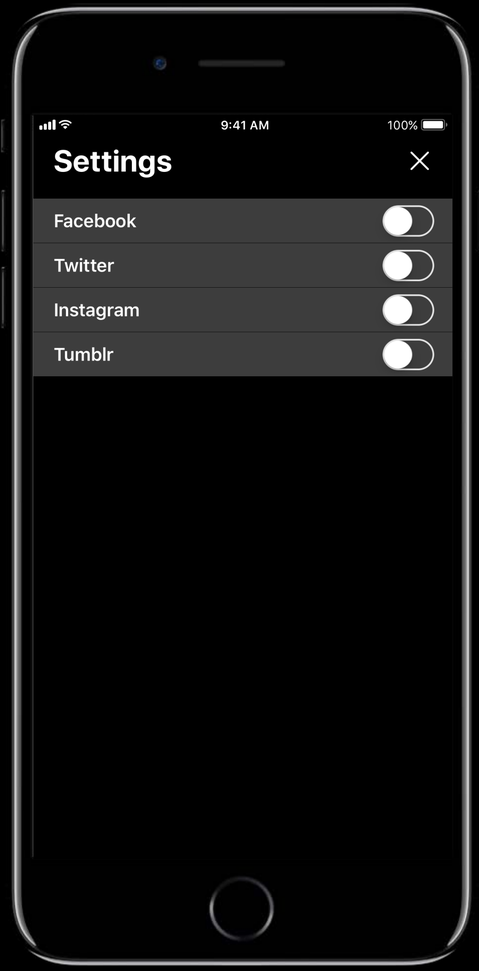
Auditing your Apps for Accessibility
https://developer.apple.com/videos/play/wwdc2016/407/

Demo

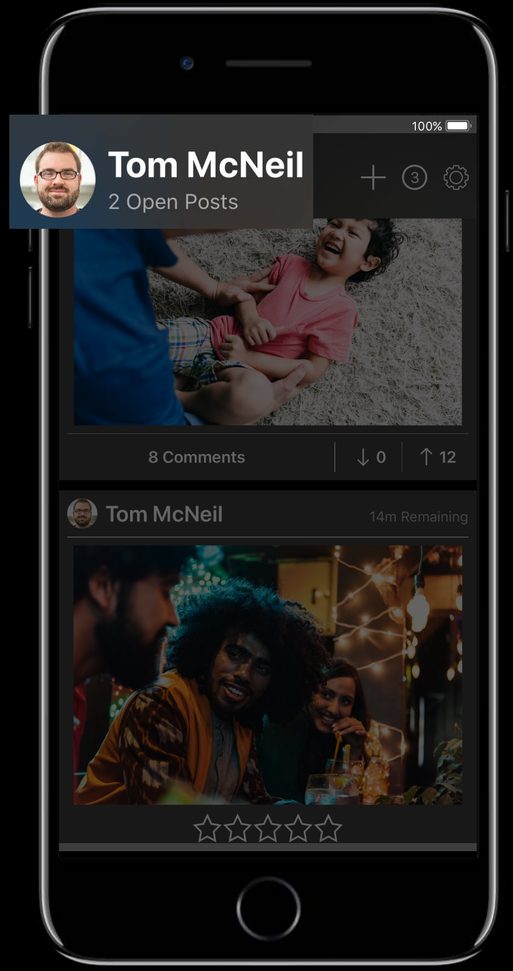
Auditing an app

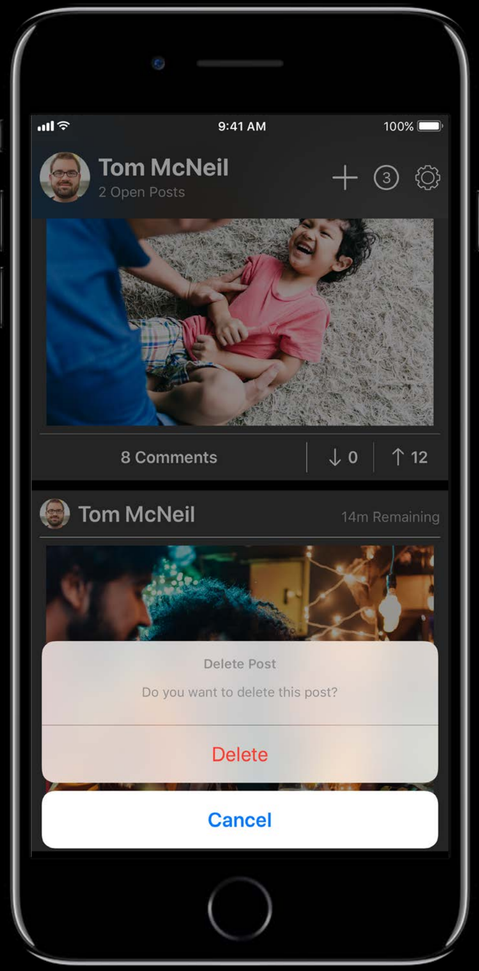
Audit Results

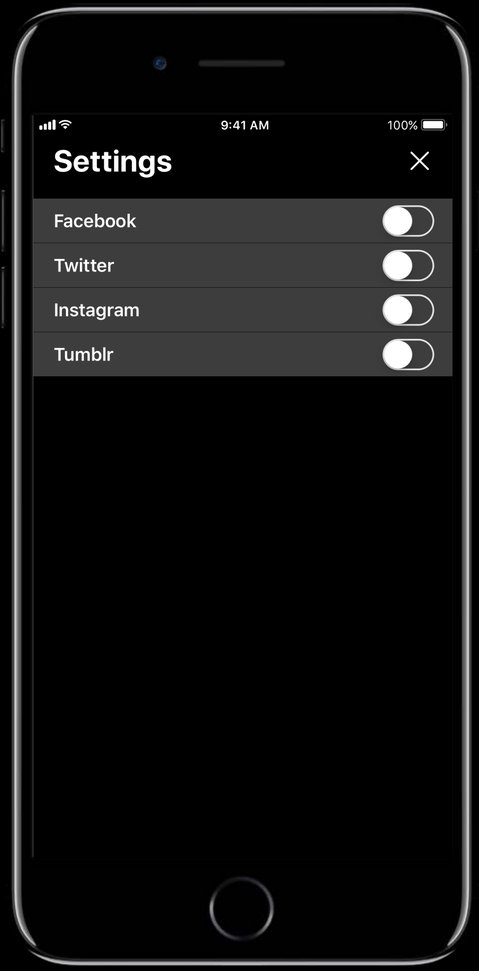




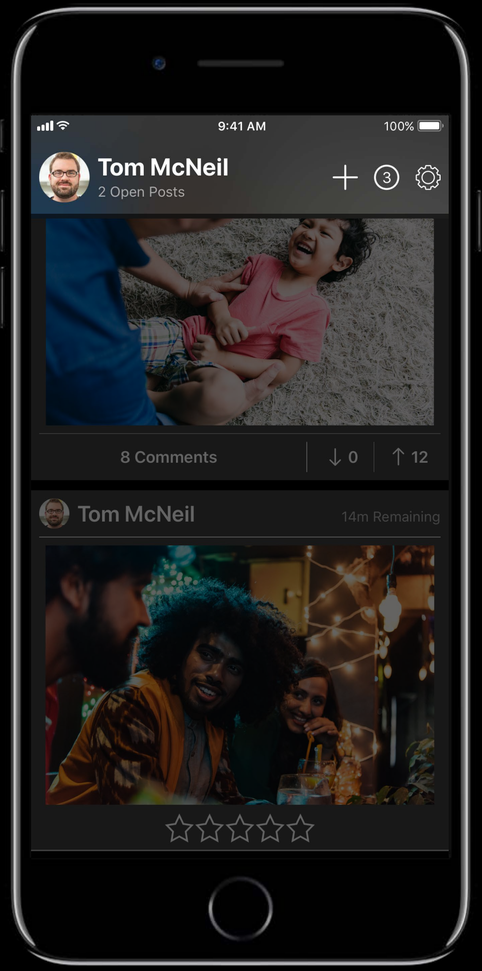
Audit Results

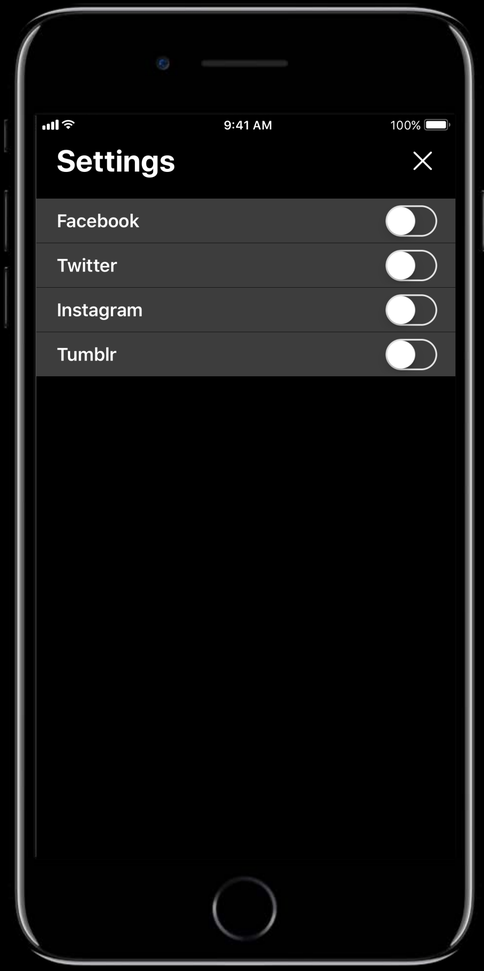




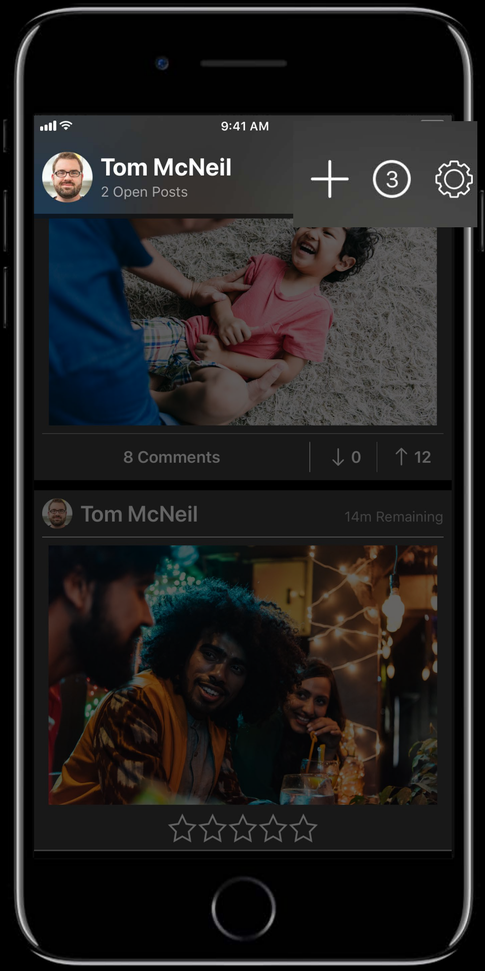


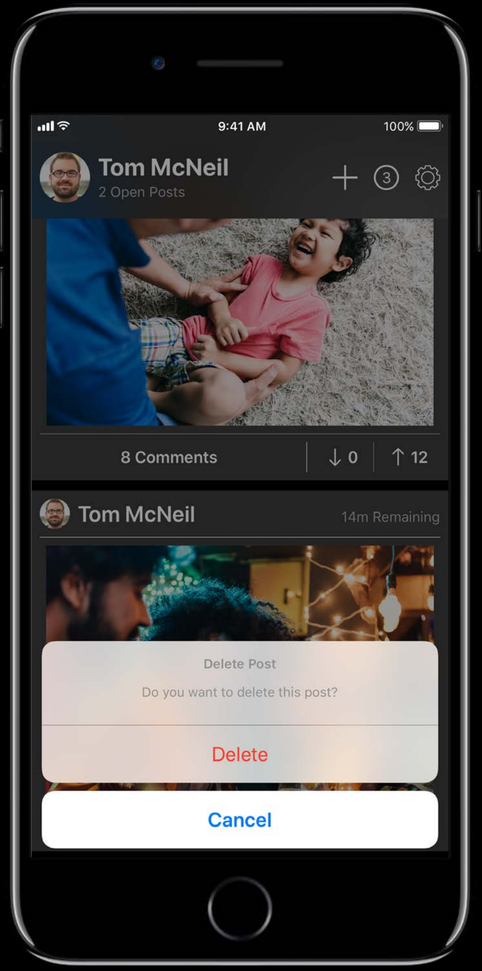
Audit Results

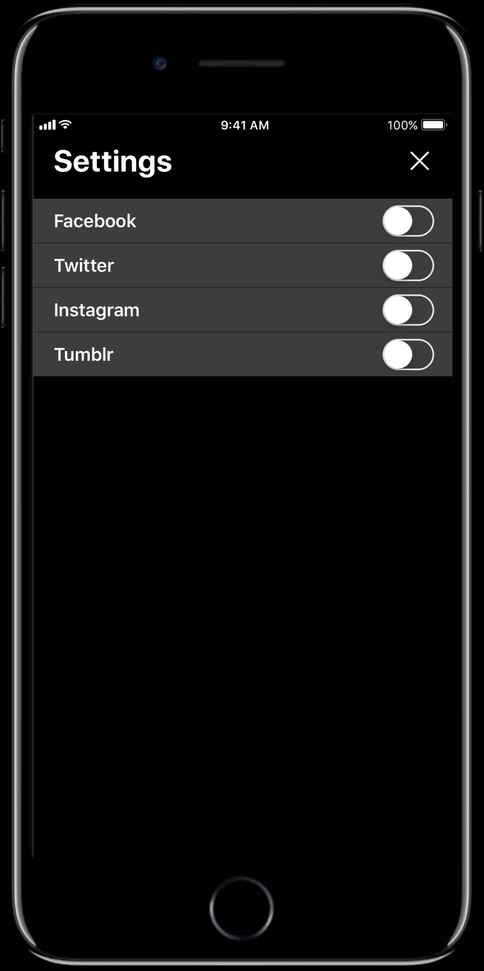




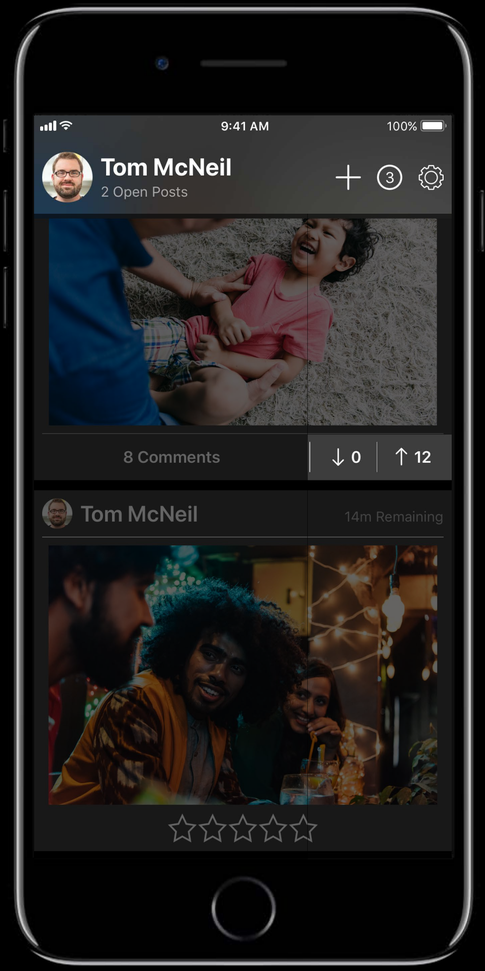
Audit Results

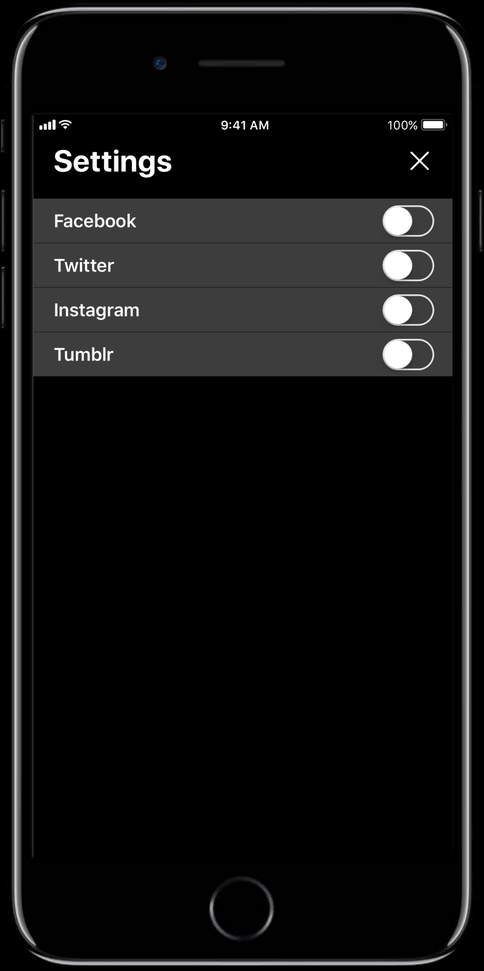




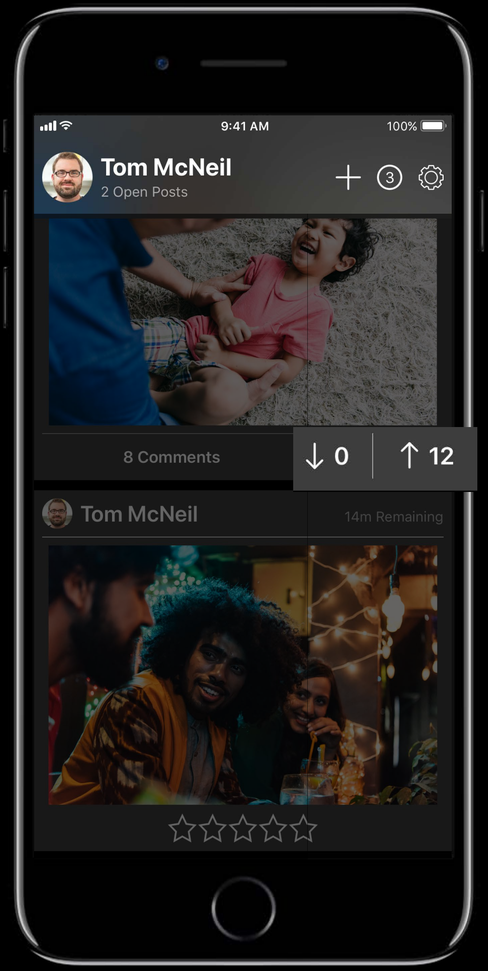


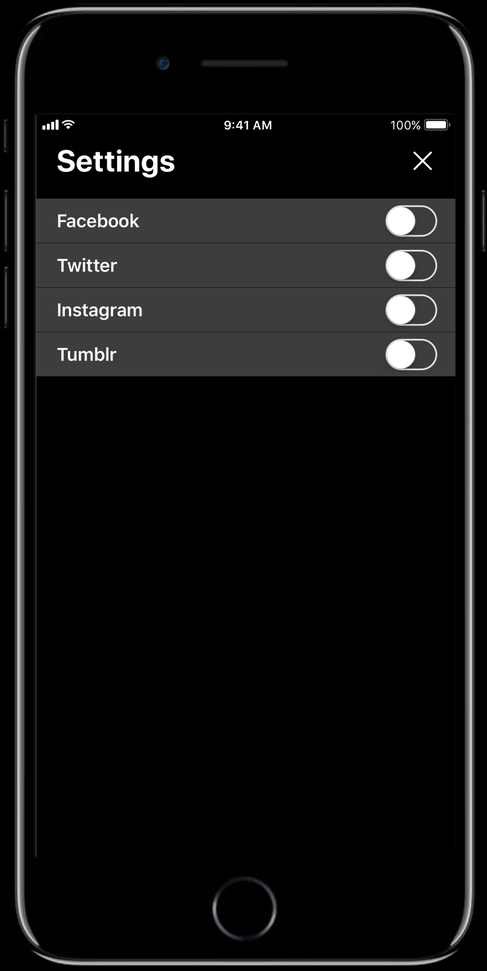
Audit Results



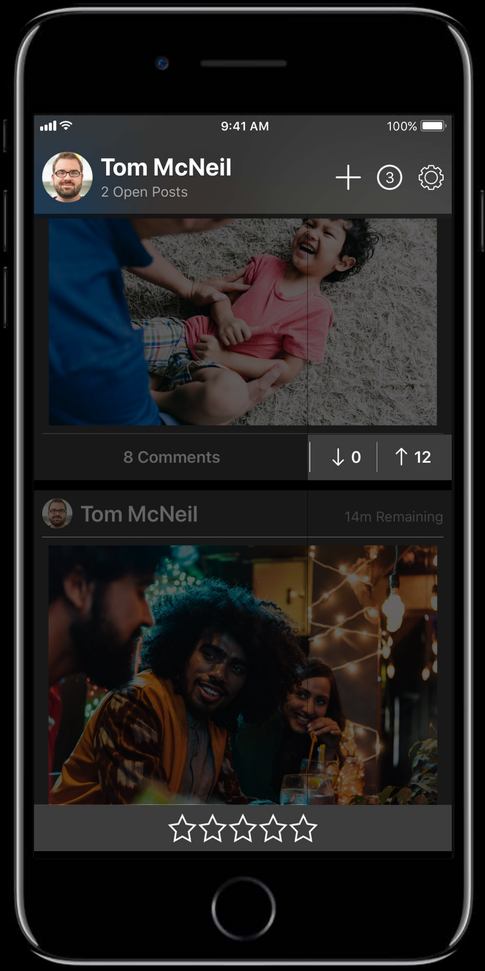


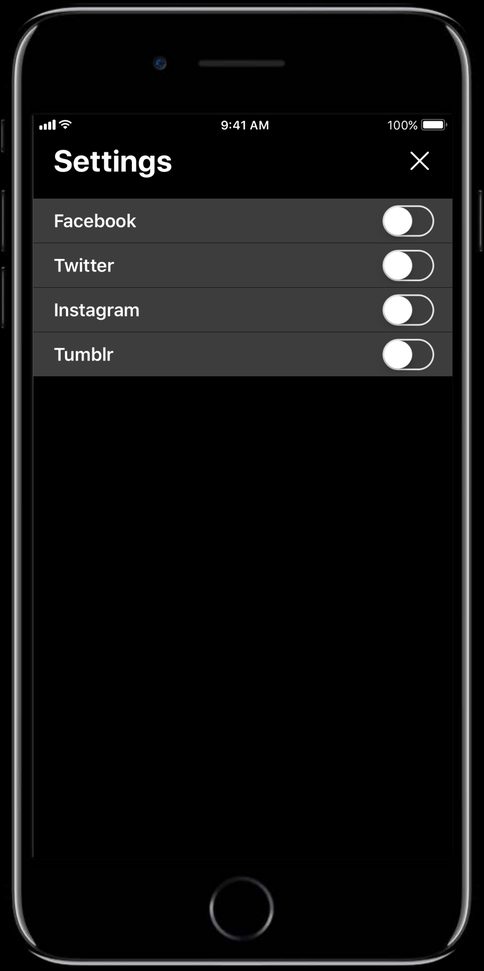
Audit Results



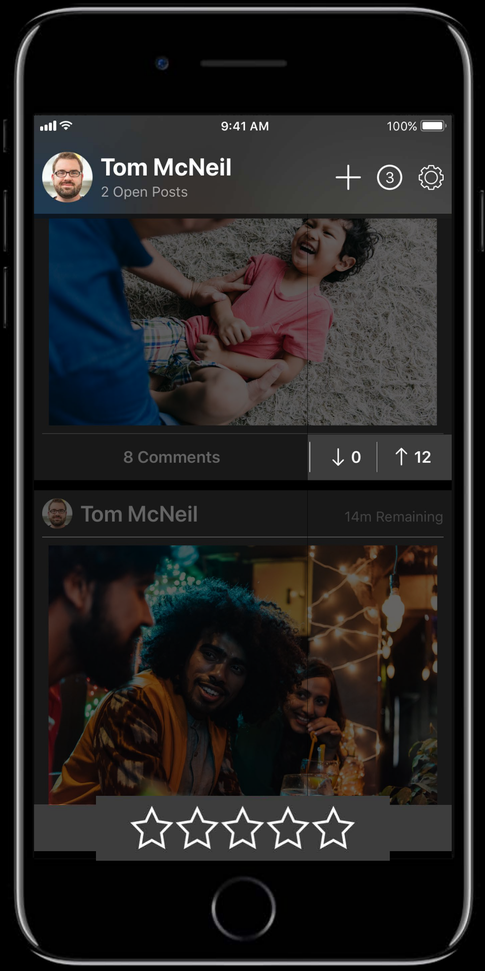


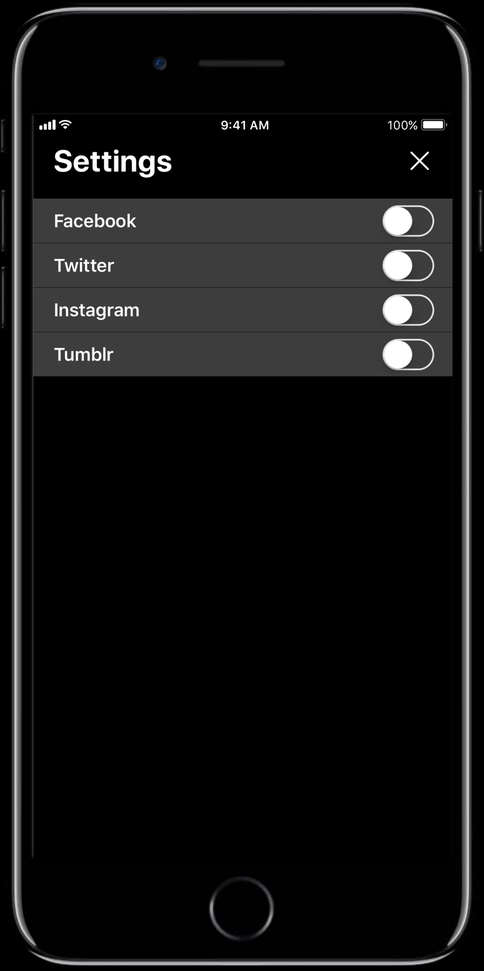
Audit Results



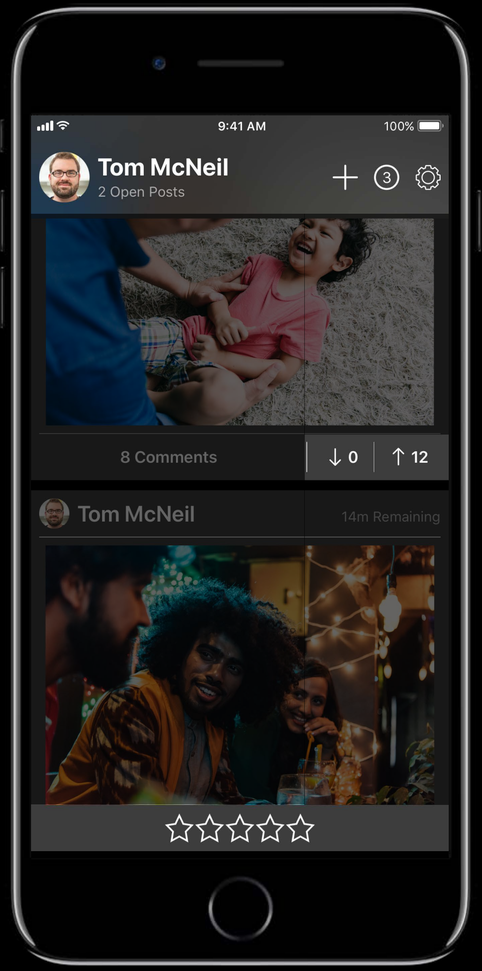


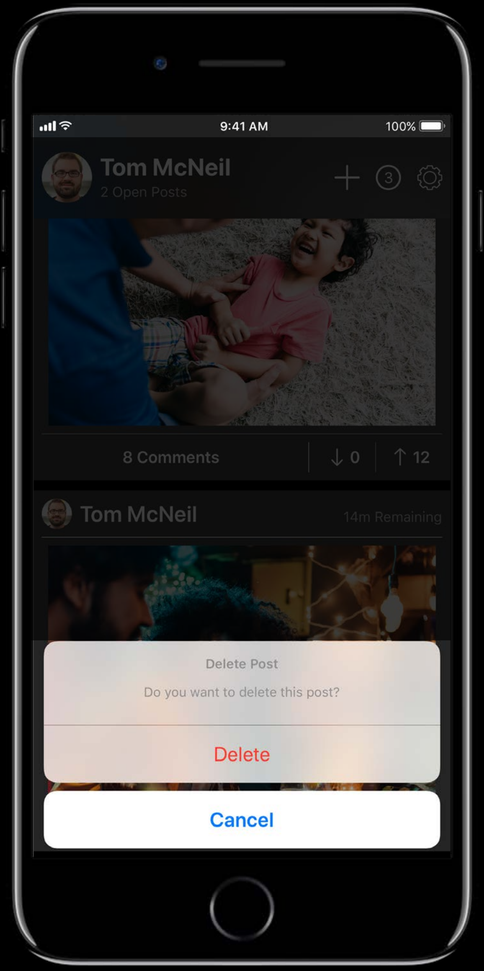
Audit Results

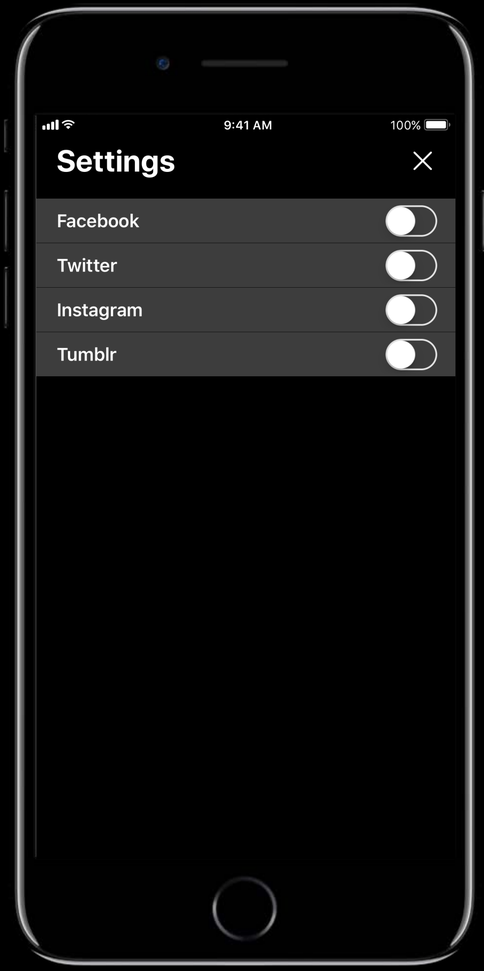




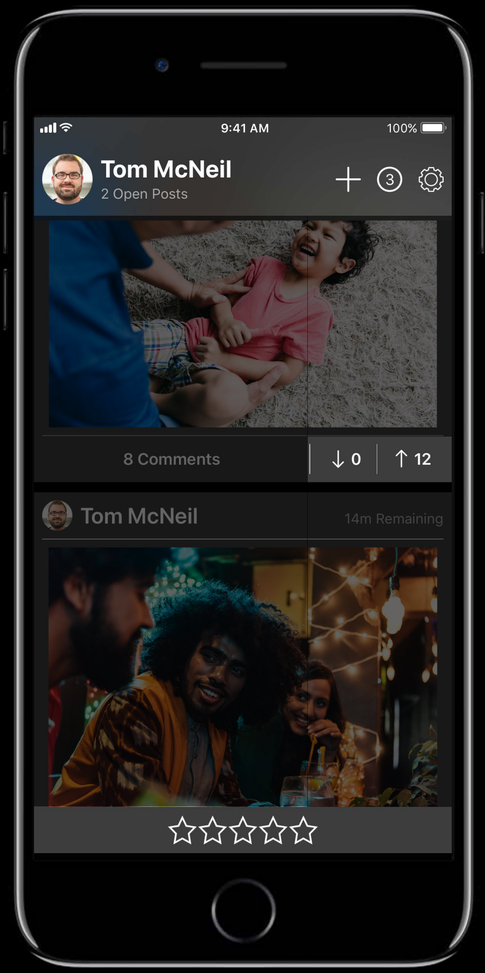
Audit Results

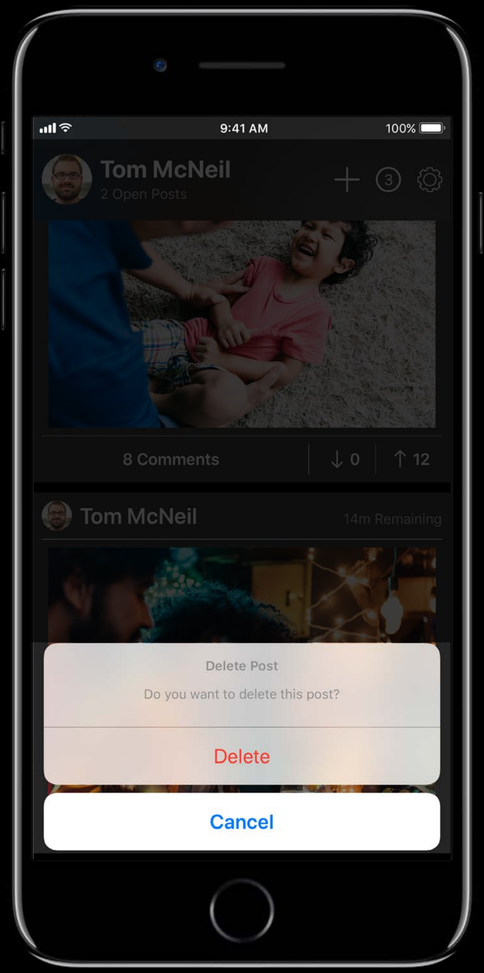


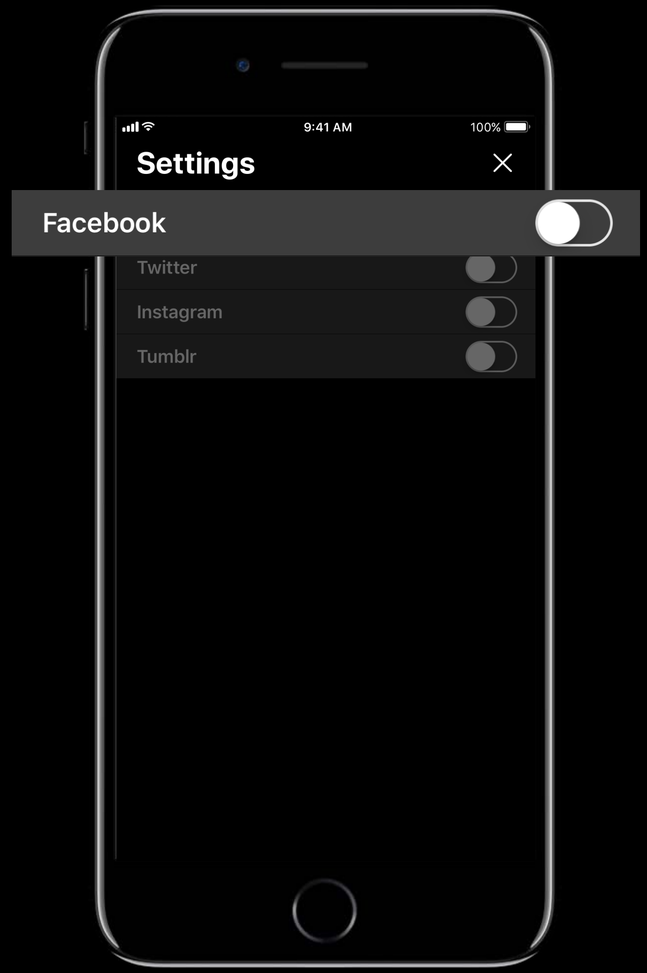




Audit Results







Accessibility API Basics

UIAccessibility Protocol

UIAccessibility Protocol

What are you?

UIAccessibility Protocol

What are you?

UIAccessibility Protocol

What are you?

Button

UIAccessibility Protocol

What are you?

Button

UIAccessibility Protocol

What are you?

Button

Who are you?

UIAccessibility Protocol

What are you?

Button

Who are you?

UIAccessibility Protocol

What are you?

Button

Who are you?

Notifications

UIAccessibility Protocol

What are you?

Button

Who are you?

Notifications

UIAccessibility Protocol

What are you?

Button

Who are you?

Notifications

What's your value?

UIAccessibility Protocol

What are you?

Button

Who are you?

Notifications

What's your value?

UIAccessibility Protocol

What are you?

Button

Who are you?

Notifications

What's your value?

3 Unread

UIAccessibility Protocol

control.accessibilityTraits

UIAccessibilityTraitButton

control.accessibilityLabel

“Notifications”

control.accessibilityValue

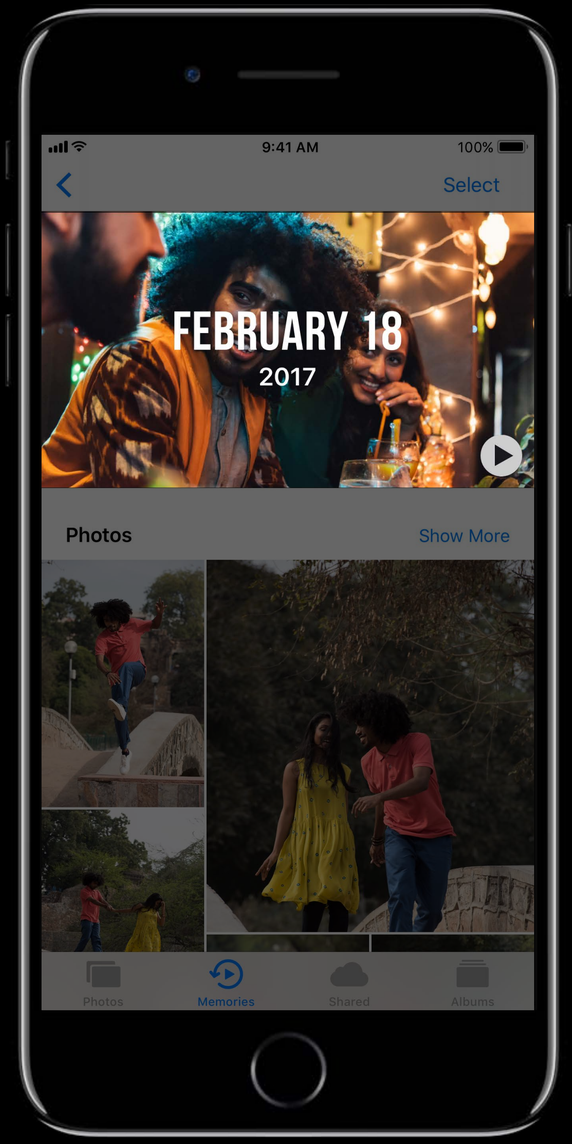
“3 Unread”

```
// UIAccessibility Basics

extension NSObject {
     open var isAccessibilityElement: Bool
     open var accessibilityLabel: NSString?
     open var accessibilityTraits: UIAccessibilityTraits
     open var accessibilityValue: NSString?
     open var accessibilityHint: NSString?
}
```

UIAccessibility

Basics

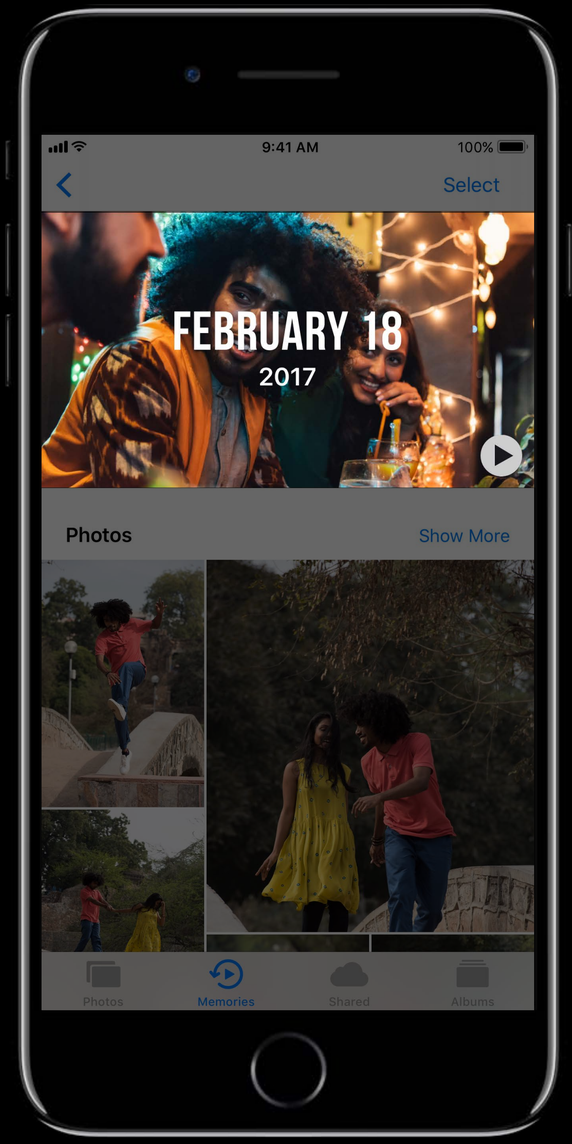
```
open var isAccessibilityElement: Bool

// Example
 memoryView.isAccessibilityElement = true
```



UIAccessibility

Basics

```
open var isAccessibilityElement: Bool

// Example
 memoryView.isAccessibilityElement = true
```

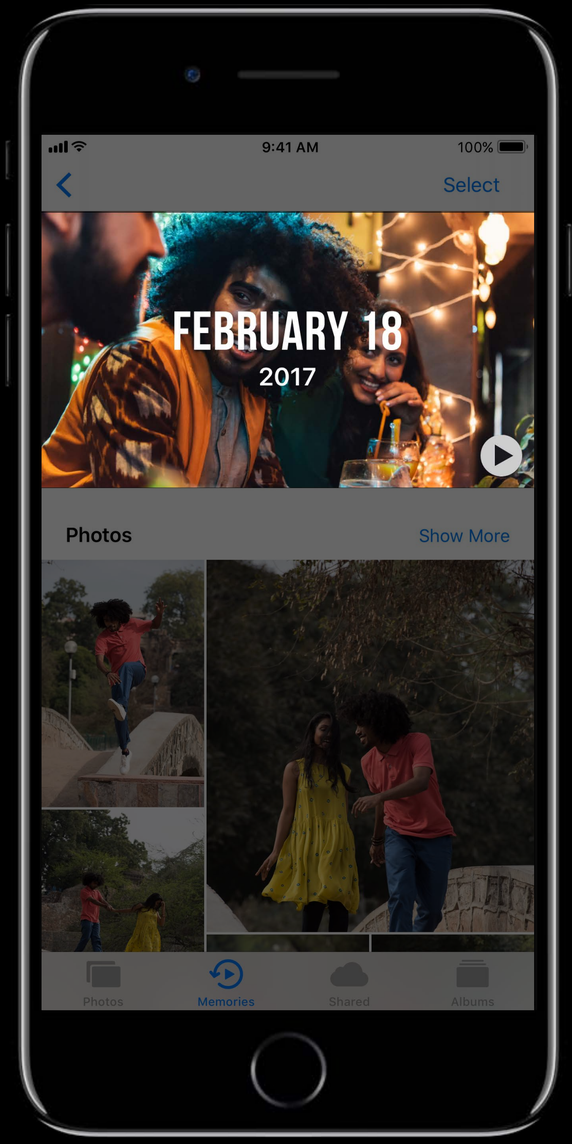


UIAccessibility

Basics

```
open var isAccessibilityElement: Bool
```

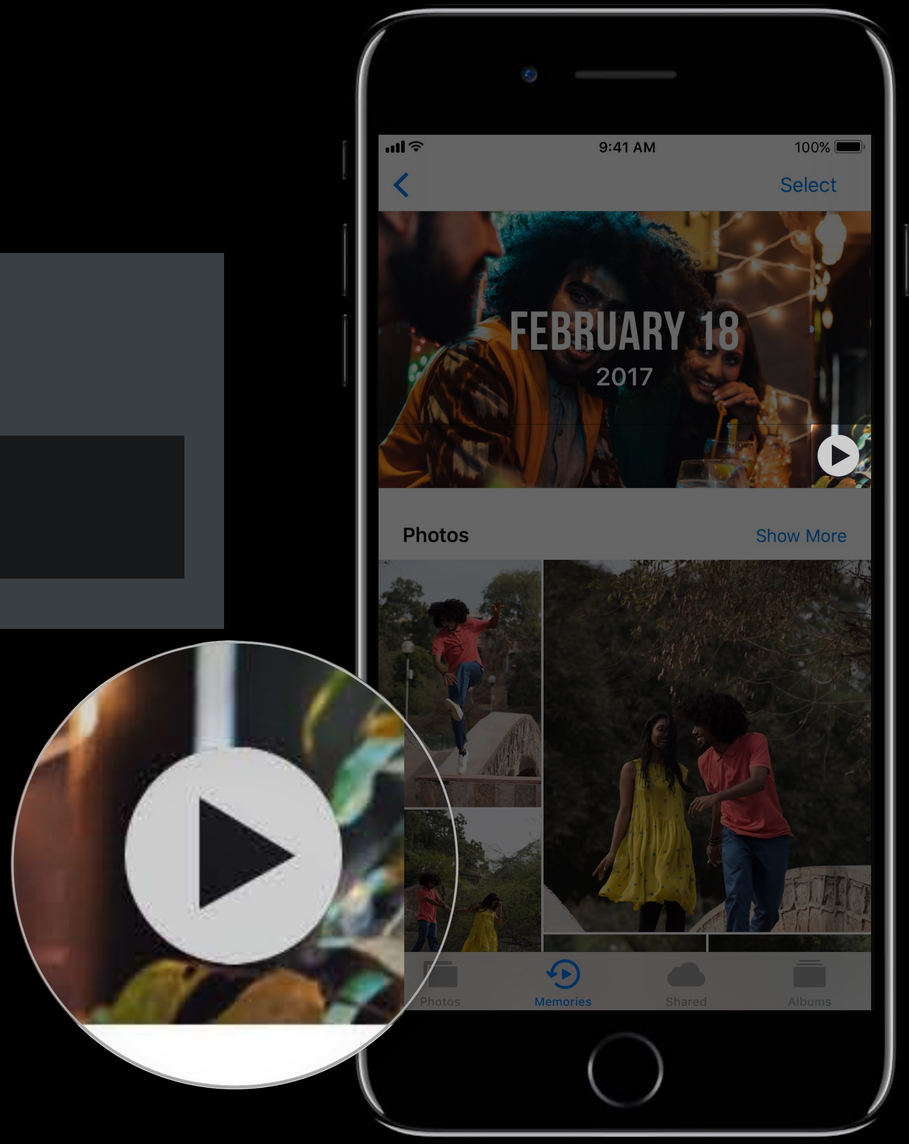
```
// Example
 memoryView.isAccessibilityElement = true
```



UIAccessibility

Basics

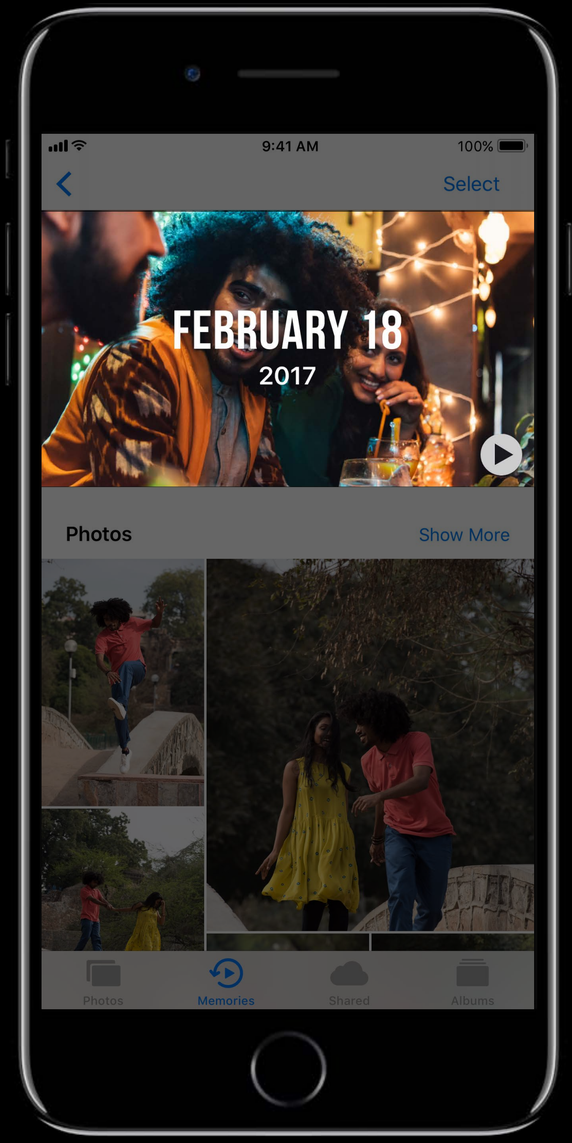
```
open var isAccessibilityElement: Bool

// Example
playButton.isAccessibilityElement = false
```



UIAccessibility

Basics

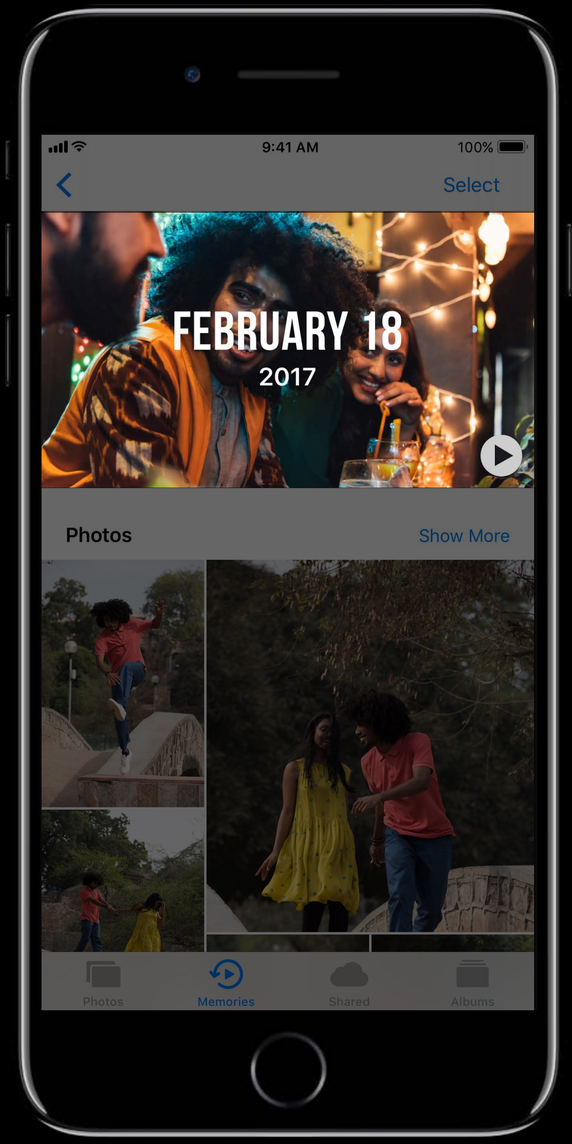
```
open var accessibilityLabel: NSString?

// Example
 memoryView.accessibilityLabel = “Memory, February 18 2017”
```



UIAccessibility

Basics

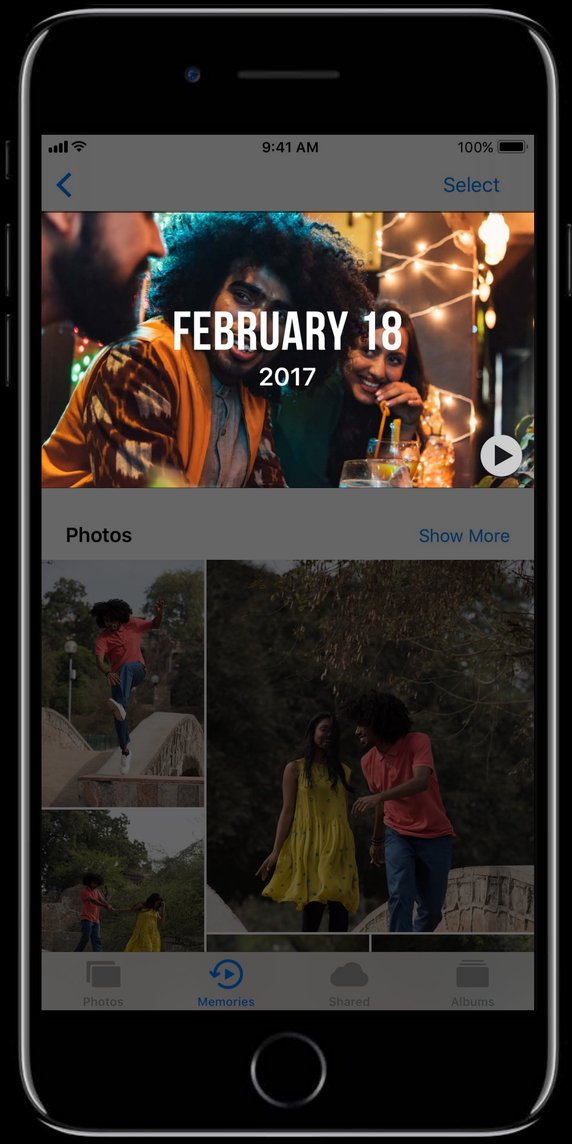
```
open var accessibilityLabel: NSString?

// Example
memoryView.accessibilityLabel = “Memory, February 18 2017”
```



UIAccessibility

Basics

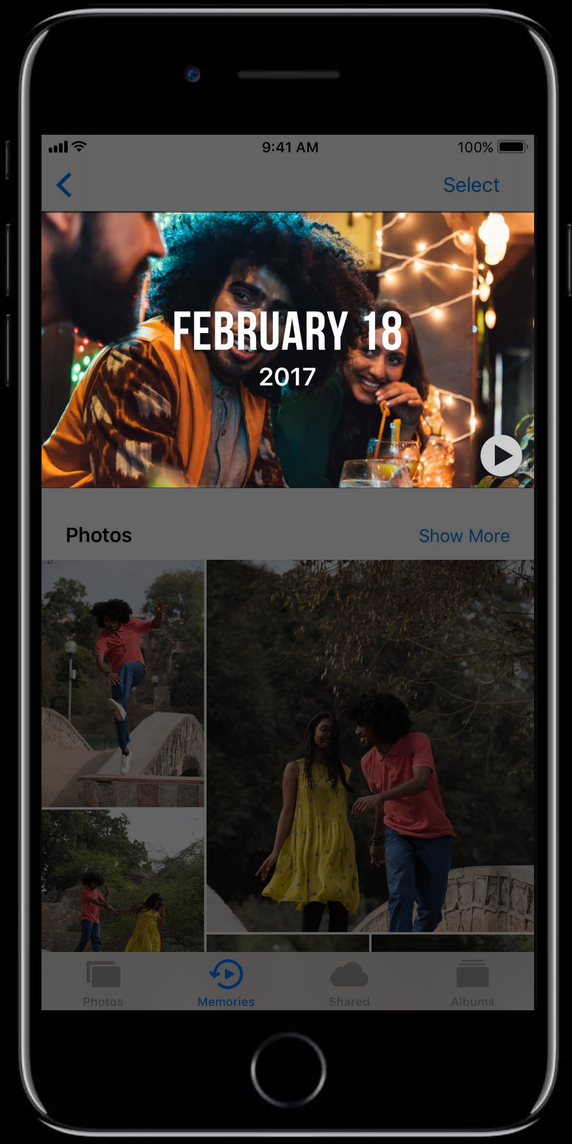
```
open var accessibilityLabel: NSString?

// Example
 memoryView.accessibilityLabel = “Memory, February 18 2017”
```



UIAccessibility

Basics

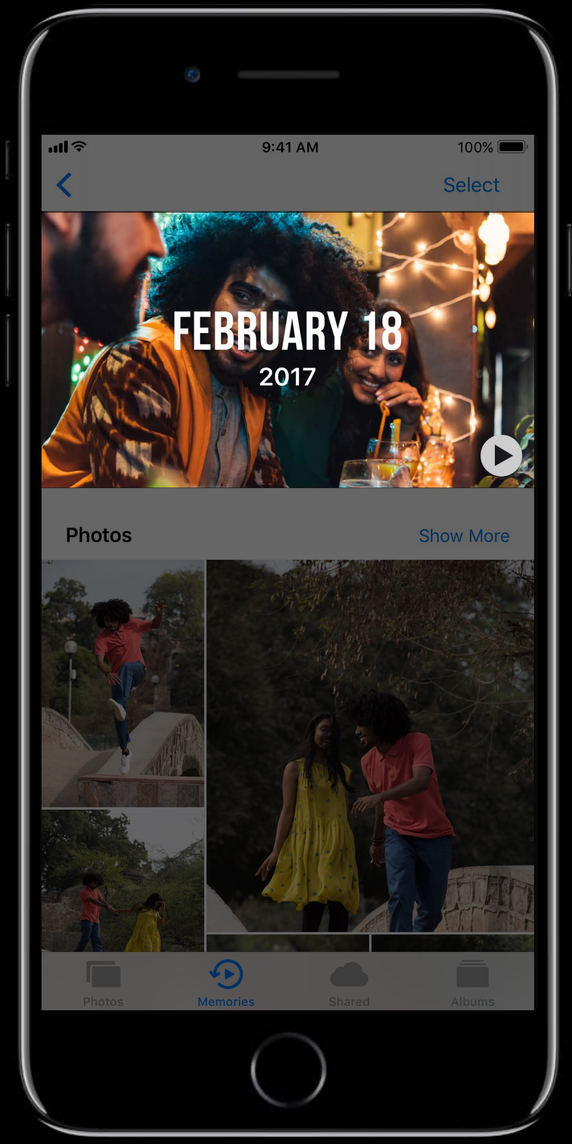
```
open var accessibilityTraits: UIAccessibilityTraits

// Example
memoryView.accessibilityTraits |= UIAccessibilityTraitButton
```



UIAccessibility

Basics

```
open var accessibilityTraits: UIAccessibilityTraits

// Example
memoryView.accessibilityTraits |= UIAccessibilityTraitButton
```

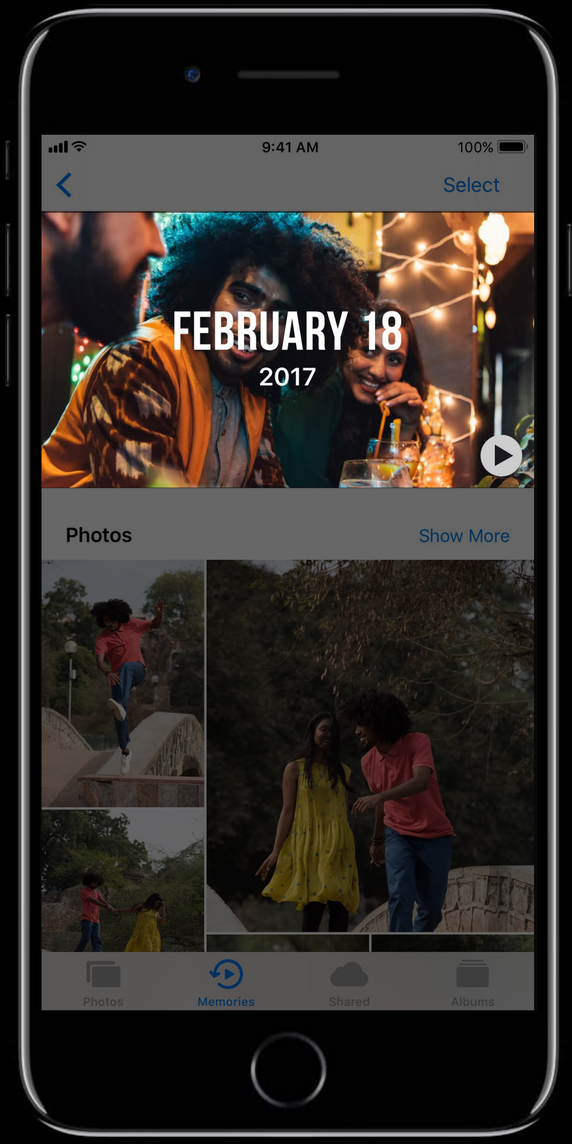


UIAccessibility

Basics

```
open var accessibilityTraits: UIAccessibilityTraits
```

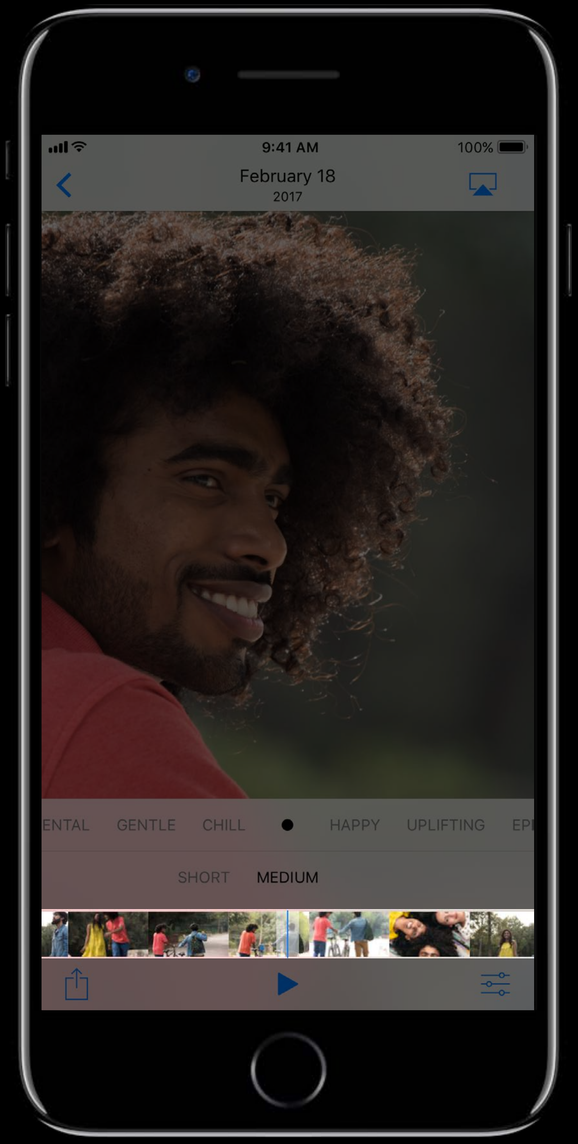
```
// Example
memoryView.accessibilityTraits |= UIAccessibilityTraitButton
```



UIAccessibility

Basics

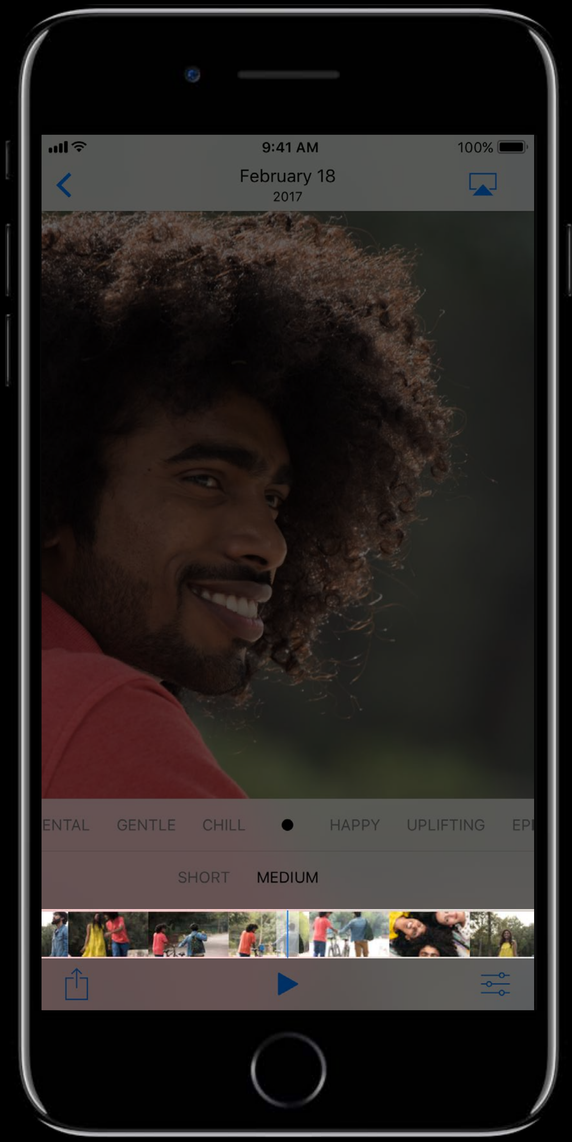
```
open var accessibilityValue: NSString?

// Example
 videoScrubber.accessibilityValue = "\(elapsedTime) seconds"
```



UIAccessibility

Basics

```
open var accessibilityValue: NSString?

// Example
 videoScrubber.accessibilityValue = "\(elapsedTime) seconds"
```

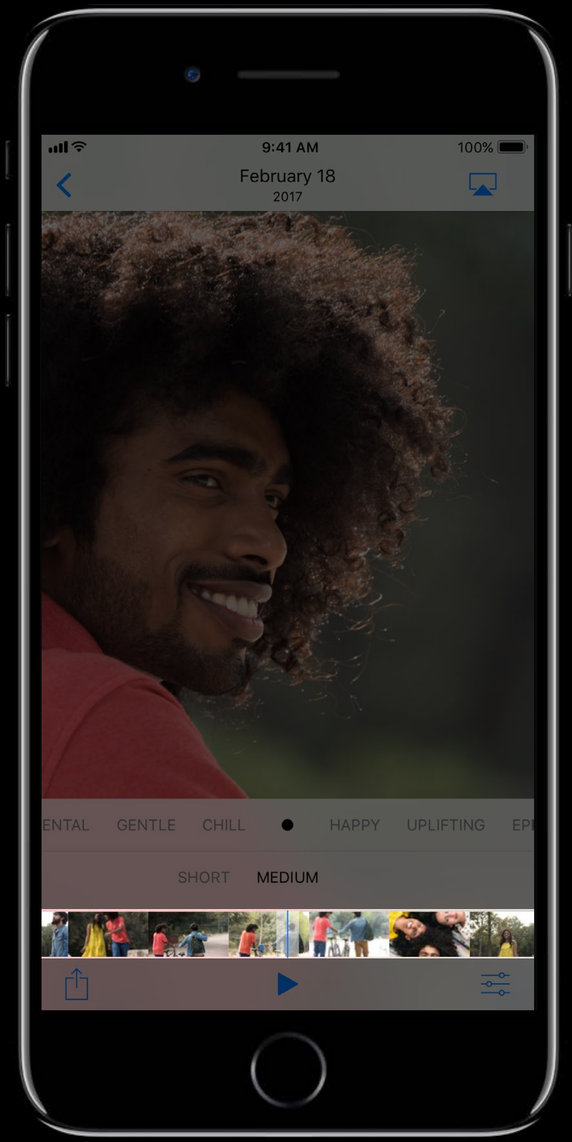


UIAccessibility

Basics

```
open var accessibilityValue: NSString?
```

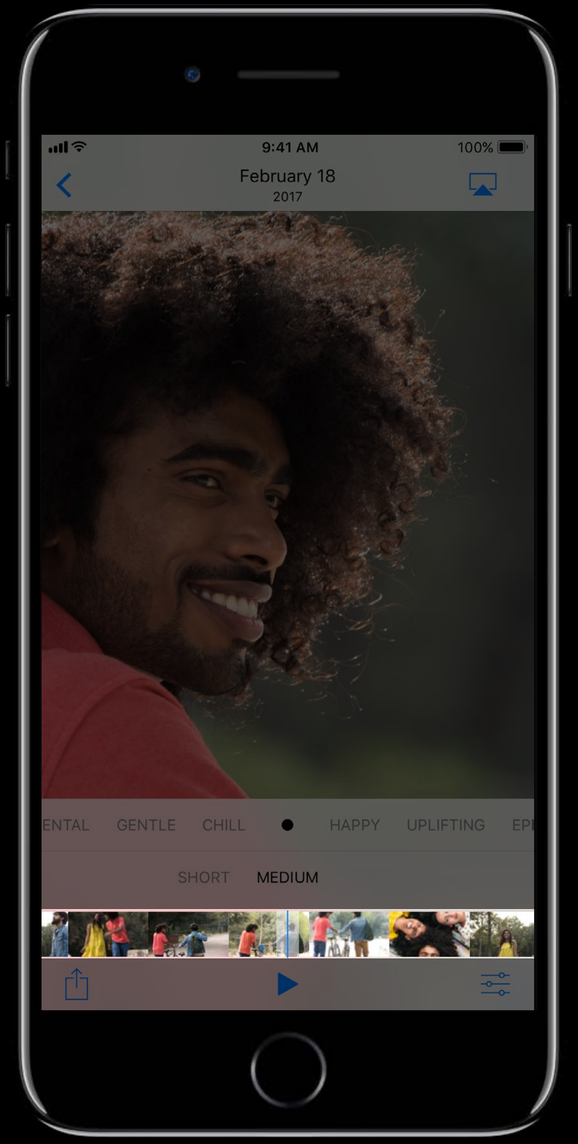
```
// Example
videoScrubber.accessibilityValue = "\(elapsedTime) seconds"
```



UIAccessibility

Basics

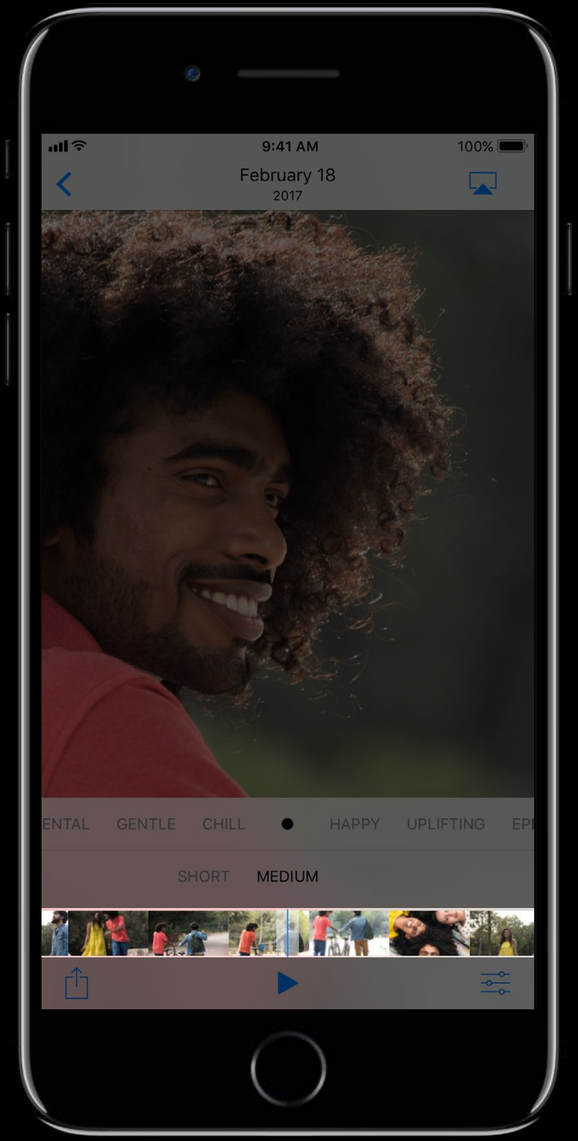
```
open var accessibilityHint: NSString?

// Example
 videoScrubber.accessibilityHint = “Swipe up or down with one
                                    finger to adjust the value“
```



UIAccessibility

Basics

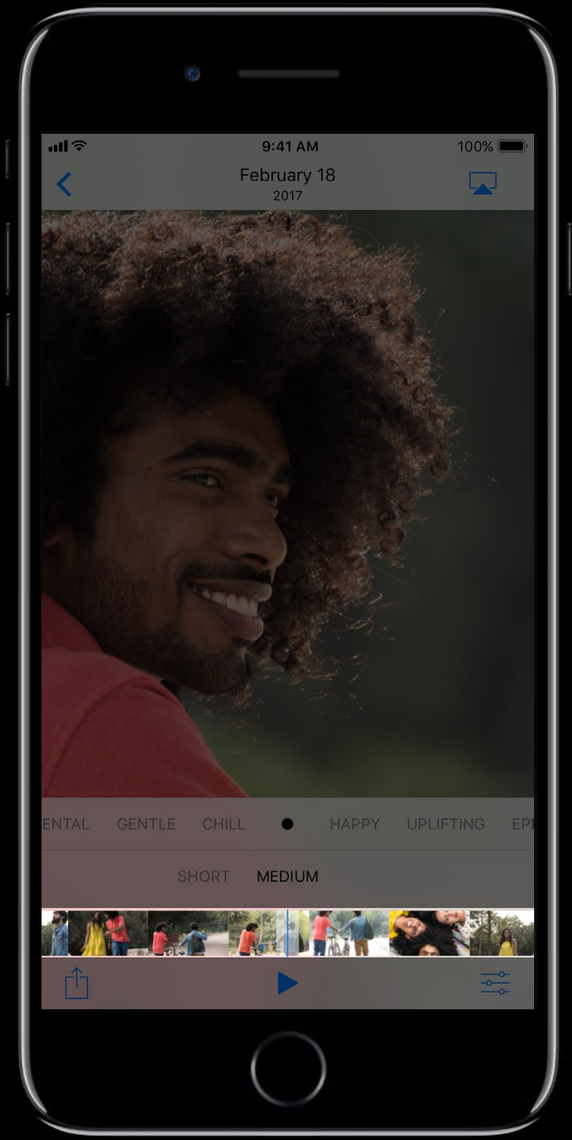
```
open var accessibilityHint: NSString?

// Example
 videoScrubber.accessibilityHint = “Swipe up or down with one
                                    finger to adjust the value“
```



UIAccessibility

Basics

```
open var accessibilityHint: NSString?

// Example
 videoScrubber.accessibilityHint = “Swipe up or down with one
                                    finger to adjust the value”
```



```
// Attributed Properties


extension NSObject {
     open var accessibilityAttributedLabel: NSAttributedString?
     open var accessibilityAttributedValue: NSAttributedString?
     open var accessibilityAttributedHint: NSAttributedString?
}
```

NEW

// Attributed Properties

NEW

```
extension NSObject {
    open var accessibilityAttributedLabel: NSAttributedString?
    open var accessibilityAttributedValue: NSAttributedString?
    open var accessibilityAttributedHint: NSAttributedString?
}
```

```
// Examples, see UIAccessibilityConstants.h for list of constants
```

NEW

```
let attributedLabel = NSAttributedString(string: “Bonjour”, attributes:
[ UIAccessibilitySpeechAttributeLanguage : “fr-FR” ]
imageView.accessibilityAttributedLabel = attributedLabel
```

```
let newValue = NSAttributedString(string: “50%”, attributes:
[ UIAccessibilitySpeechAttributeQueueAnnouncement : true ]
UIAccessibilityPostNotification(UIAccessibilityAnnouncementNotification, newValue)
```

// Examples, see UIAccessibilityConstants.h for list of constants

NEW

```
let attributedLabel = NSAttributedString(string: “Bonjour”, attributes:
[ UIAccessibilitySpeechAttributeLanguage : “fr-FR” ]
imageView.accessibilityAttributedLabel = attributedLabel
```

```
let newValue = NSAttributedString(string: “50%”, attributes:
[ UIAccessibilitySpeechAttributeQueueAnnouncement : true ]
UIAccessibilityPostNotification(UIAccessibilityAnnouncementNotification, newValue)
```

```
// Examples, see UIAccessibilityConstants.h for list of constants
```

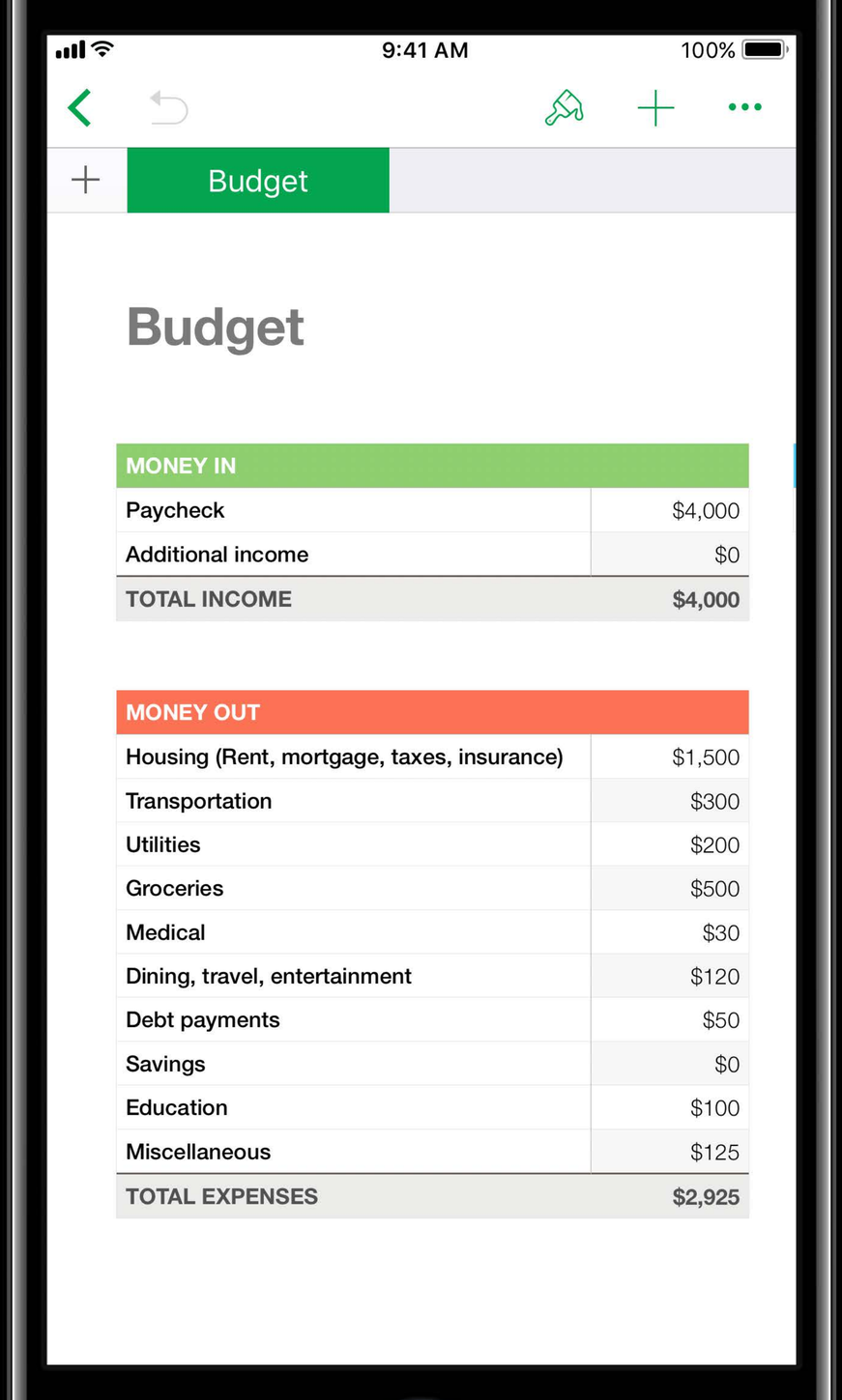
NEW

```
let attributedLabel = NSAttributedString(string: “Bonjour”, attributes:
[ UIAccessibilitySpeechAttributeLanguage : “fr-FR” ]
imageView.accessibilityAttributedLabel = attributedLabel
```

```
let newValue = NSAttributedString(string: “50%”, attributes:
[ UIAccessibilitySpeechAttributeQueueAnnouncement : true ]
UIAccessibilityPostNotification(UIAccessibilityAnnouncementNotification, newValue)
```

Containers

NEW



```
// Containers
```

NEW

```
extension NSObject {
    open var accessibilityContainerType: UIAccessibilityContainerType
}

public enum UIAccessibilityContainerType : Int {
     case none
     case dataTable //requires conformance to UIAccessibilityContainerDataTable protocol
     case list
     case landmark
}
```

// Containers

NEW

```
extension NSObject {
    open var accessibilityContainerType: UIAccessibilityContainerType
}
```

```
public enum UIAccessibilityContainerType : Int {
     case none
     case dataTable //requires conformance to UIAccessibilityContainerDataTable protocol
     case list
     case landmark
}
```

```
// Containers
```

NEW

```
extension NSObject {
    open var accessibilityContainerType: UIAccessibilityContainerType
}
```

```
public enum UIAccessibilityContainerType : Int {
     case none
     case dataTable //requires conformance to UIAccessibilityContainerDataTable protocol
     case list
     case landmark
}
```

Non-View Elements

```
open class UIAccessibilityElement : NSObject {
     public init(accessibilityContainer container: AnyObject)
}

extension NSObject {
     open var accessibilityElements: [AnyObject]?
}
```

```
open class UIAccessibilityElement : NSObject {
    public init(accessibilityContainer container: AnyObject)
}

extension NSObject {
    open var accessibilityElements: [AnyObject]?
}
```

```
open class UIAccessibilityElement : NSObject {
    public init(accessibilityContainer container: AnyObject)
}
```

```
extension NSObject {
    open var accessibilityElements: [AnyObject]?
}
```

9:41 AM 100%

Tom McNeil
2 Open Posts

+ 3

Lexi Torres 5m Remaining

Just finished my second day of my first WWDC ever! My brain hurts from infromation overlaod 😵

3 Comments ↓ 1 ↑ 3

Greg Apodaca 12m Remaining

We have fun at home

8 Comments ↓ 0 ↑ 12

Tom McNeil 14m Remaining

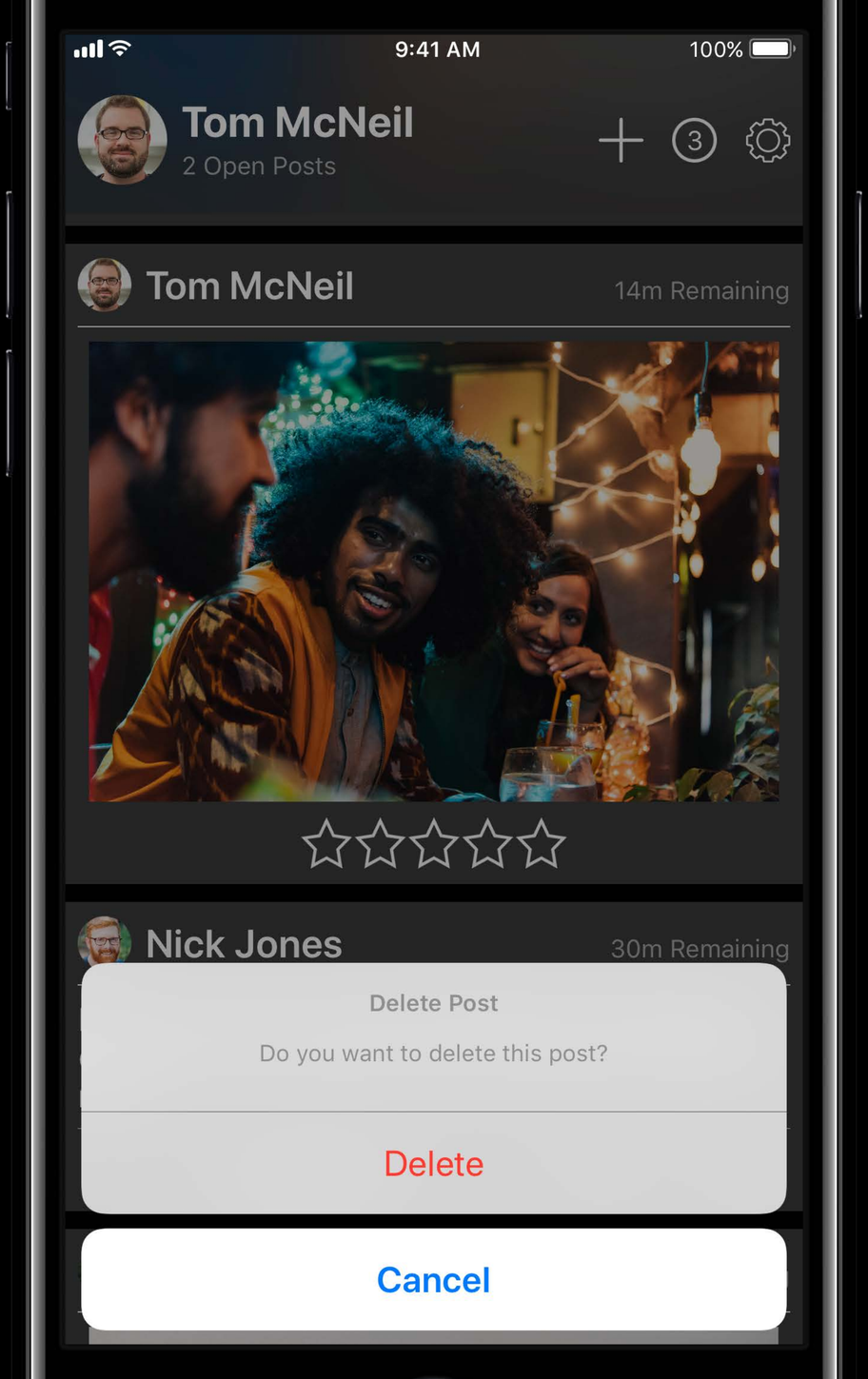
Demo

Using UIAccessibility basics

Beyond the Basics

Common problems and solutions

Custom Actions



```
// Custom Actions

extension NSObject {
    open var accessibilityCustomActions: [UIAccessibilityCustomAction]?
}

open class UIAccessibilityCustomAction : NSObject {
    open var name: String
    open var target: AnyObject?
    open var selector: Selector

    public init(name: String, target: Any?, selector: Selector)
}
```

```
// Custom Actions

extension NSObject {
    open var accessibilityCustomActions: [UIAccessibilityCustomAction]?
}

open class UIAccessibilityCustomAction : NSObject {
    open var name: String
    open var target: AnyObject?
    open var selector: Selector

    public init(name: String, target: Any?, selector: Selector)
}
```

```
// Custom Actions

extension NSObject {
    open var accessibilityCustomActions: [UIAccessibilityCustomAction]?
}
```

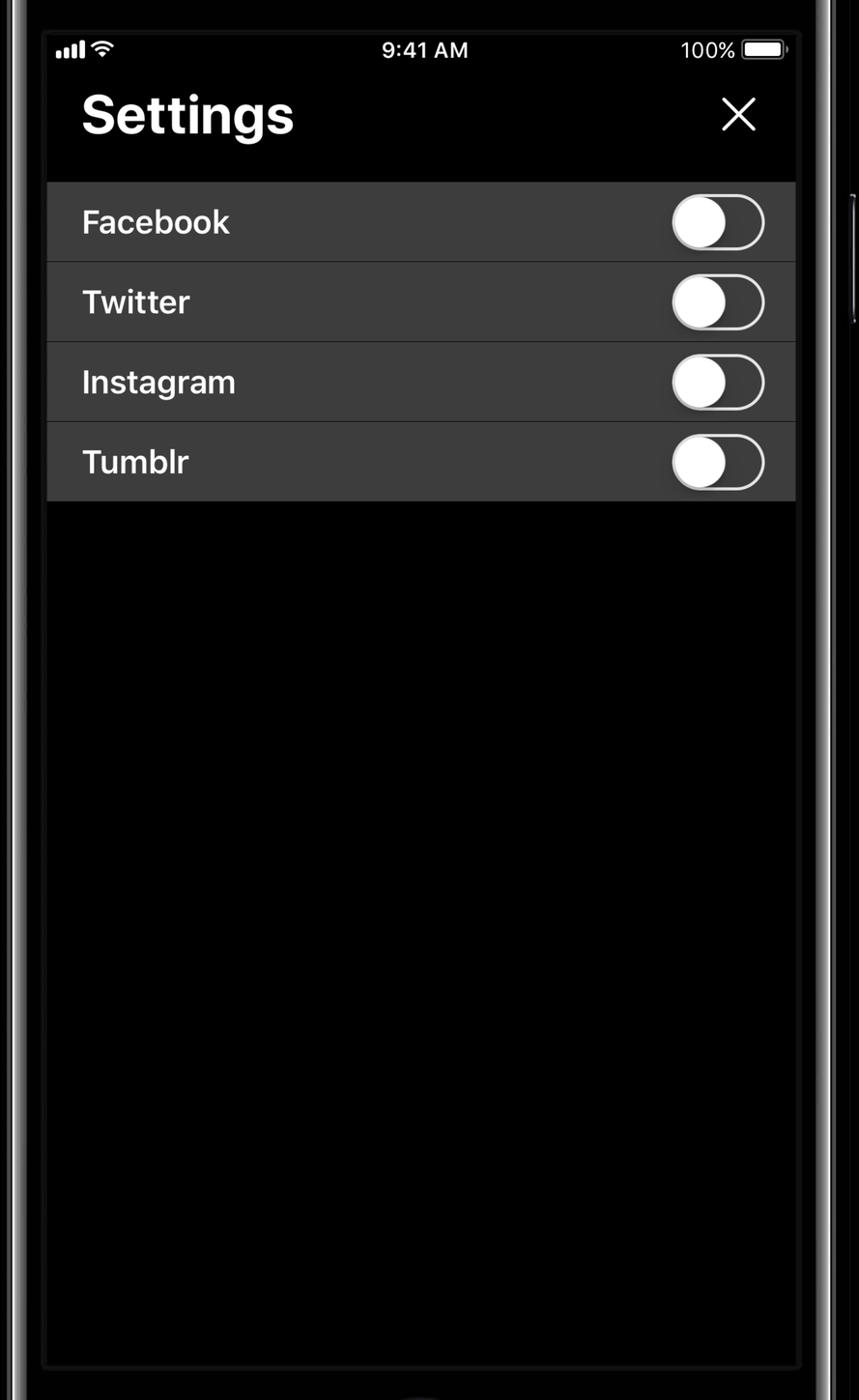
```
open class UIAccessibilityCustomAction : NSObject {
    open var name: String
    open var target: AnyObject?
    open var selector: Selector

    public init(name: String, target: Any?, selector: Selector)
}
```

```
// UIAccessibilityCustomAction Example

class StandardPostCollectionViewCell : UICollectionViewCell {

    func commonInit() {
        // Setup Code...
        let customAction = UIAccessibilityCustomAction(name: "Delete", target: self, selector:
                                                       #selector(deleteCellAction))
        accessibilityCustomActions = [customAction]
    }

    func deleteCellAction() -> Bool {
        // Code for initiating cell delete
    }
}
```

```
// UIAccessibilityCustomAction Example

class StandardPostCollectionViewCell : UICollectionViewCell {

    func commonInit() {
        // Setup Code...
        let customAction = UIAccessibilityCustomAction(name: "Delete", target: self, selector:
                                                       #selector(deleteCellAction))
        accessibilityCustomActions = [customAction]
    }

    func deleteCellAction() -> Bool {
        // Code for initiating cell delete
    }
}
```

```
// UIAccessibilityCustomAction Example

class StandardPostCollectionViewCell : UICollectionViewCell {

    func commonInit() {
        // Setup Code...
        let customAction = UIAccessibilityCustomAction(name: "Delete", target: self, selector:
                                                       #selector(deleteCellAction))
        accessibilityCustomActions = [customAction]
    }

    func deleteCellAction() -> Bool {
        // Code for initiating cell delete
    }
}
```

Default Activation



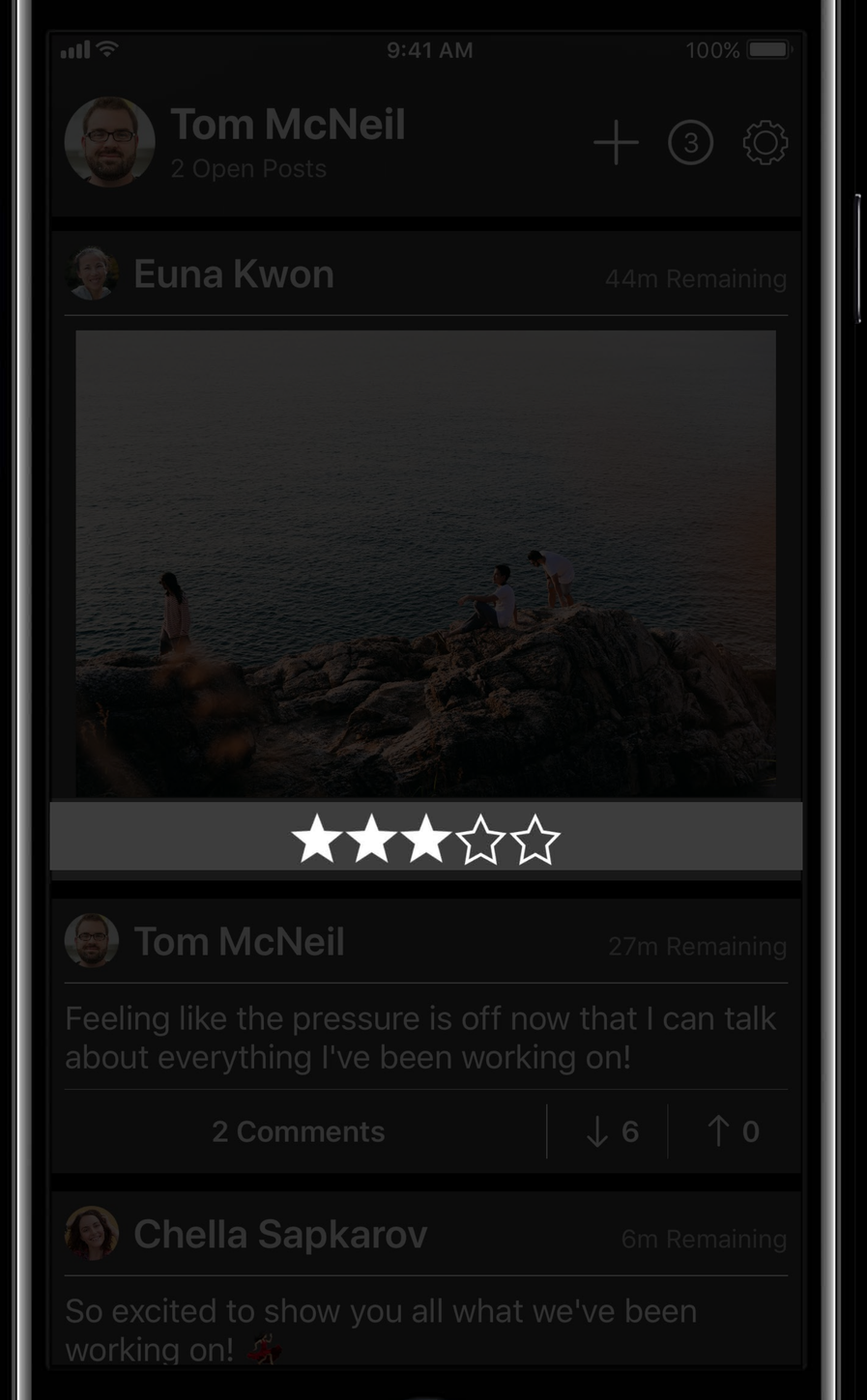
```
// Accessibility Activate

extension NSObject {
     open func accessibilityActivate() -> Bool // Boolean indicates success of activation
}
```

```
// Accessibility Activate

extension NSObject {
     open func accessibilityActivate() -> Bool // Boolean indicates success of activation
}
```

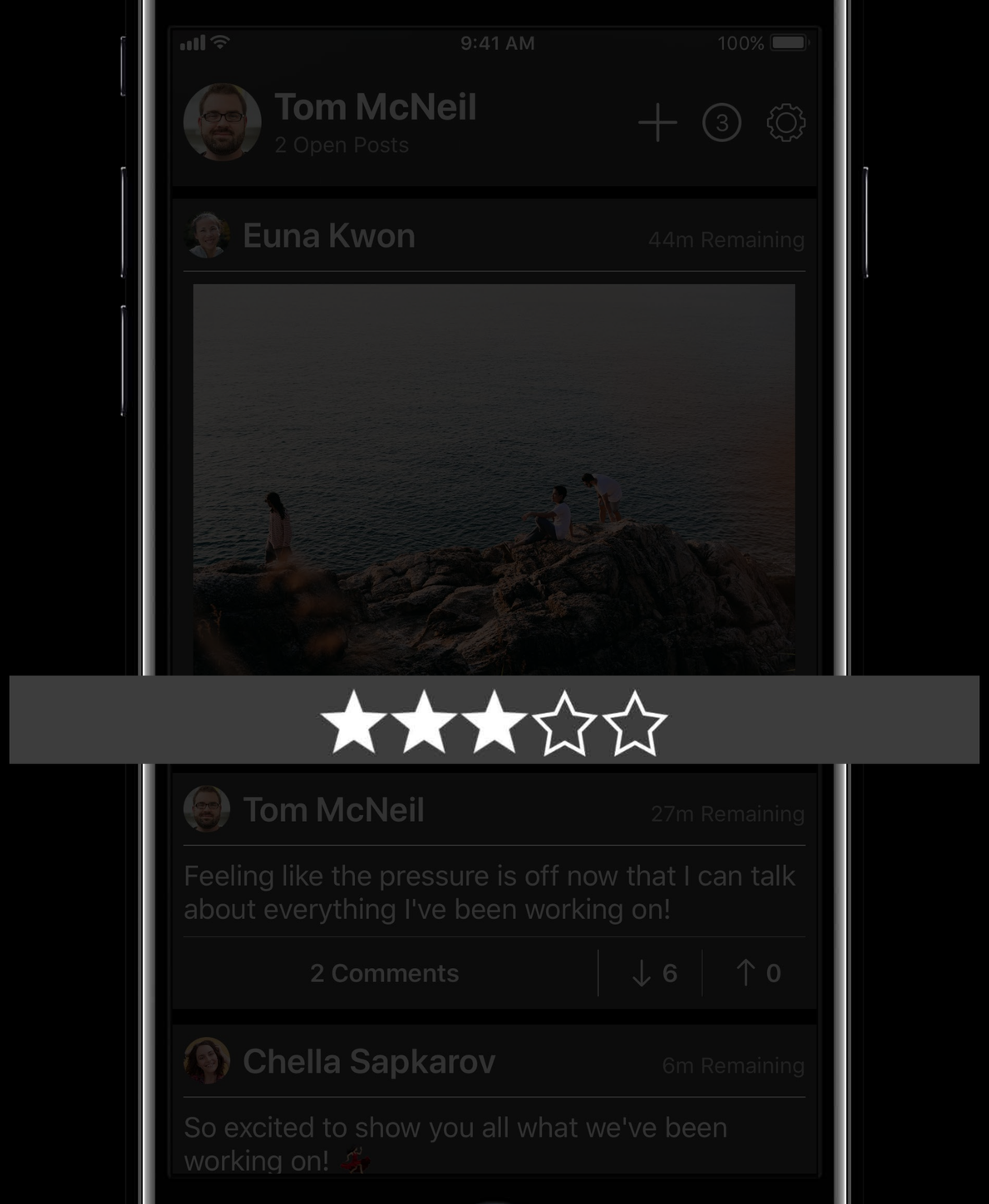
```
// Accessibility Activate Example

class SwitchTableViewCell : UITableViewCell {
     let switchView: UISwitch

     override func accessibilityActivate() -> Bool {
          switchView.setOn(!switchView.isOn, animated: true)
          return true
     }
}
```

```
// Accessibility Activate Example

class SwitchTableViewCell : UITableViewCell {
     let switchView: UISwitch

     override func accessibilityActivate() -> Bool {
          switchView.setOn(!switchView.isOn, animated: true)
          return true
     }
}
```

Adjustable


9:41 AM
100%
Tom McNeil
2 Open Posts
3
Euna Kwon
44m Remaining
Tom McNeil
27m Remaining
Feeling like the pressure is off now that I can talk about everything I've been working on!
2 Comments
6
0
Chella Sapkarov
6m Remaining
So excited to show you all what we've been working on!

Adjustable


9:41 AM
100%
Tom McNeil
2 Open Posts
3
Euna Kwon
44m Remaining
Tom McNeil
27m Remaining
Feeling like the pressure is off now that I can talk about everything I've been working on!
2 Comments
6
0
Chella Sapkarov
6m Remaining
So excited to show you all what we've been working on!

```
// Views That Increment and Decrement

extension NSObject {
    // Must have the UIAccessibilityTraitAdjustable
    open func accessibilityIncrement()
    open func accessibilityDecrement()
}
```

// Views That Increment and Decrement

```
extension NSObject {
    // Must have the UIAccessibilityTraitAdjustable
    open func accessibilityIncrement()
    open func accessibilityDecrement()
}
```

```
// Increment and Decrement Example
class StarsView : UIView {
    var numStars: Int = 0 {
        didSet {
            // Code for adjusting the star rating
        }
    }
    override var accessibilityTraits {
        get { return super.accessibilityTraits | UIAccessibilityTraitAdjustable }
        set { }
    }
    override func accessibilityIncrement() {
        numStars += 1
    }
    override func accessibilityDecrement() {
        numStars -= 1
    }
}
```

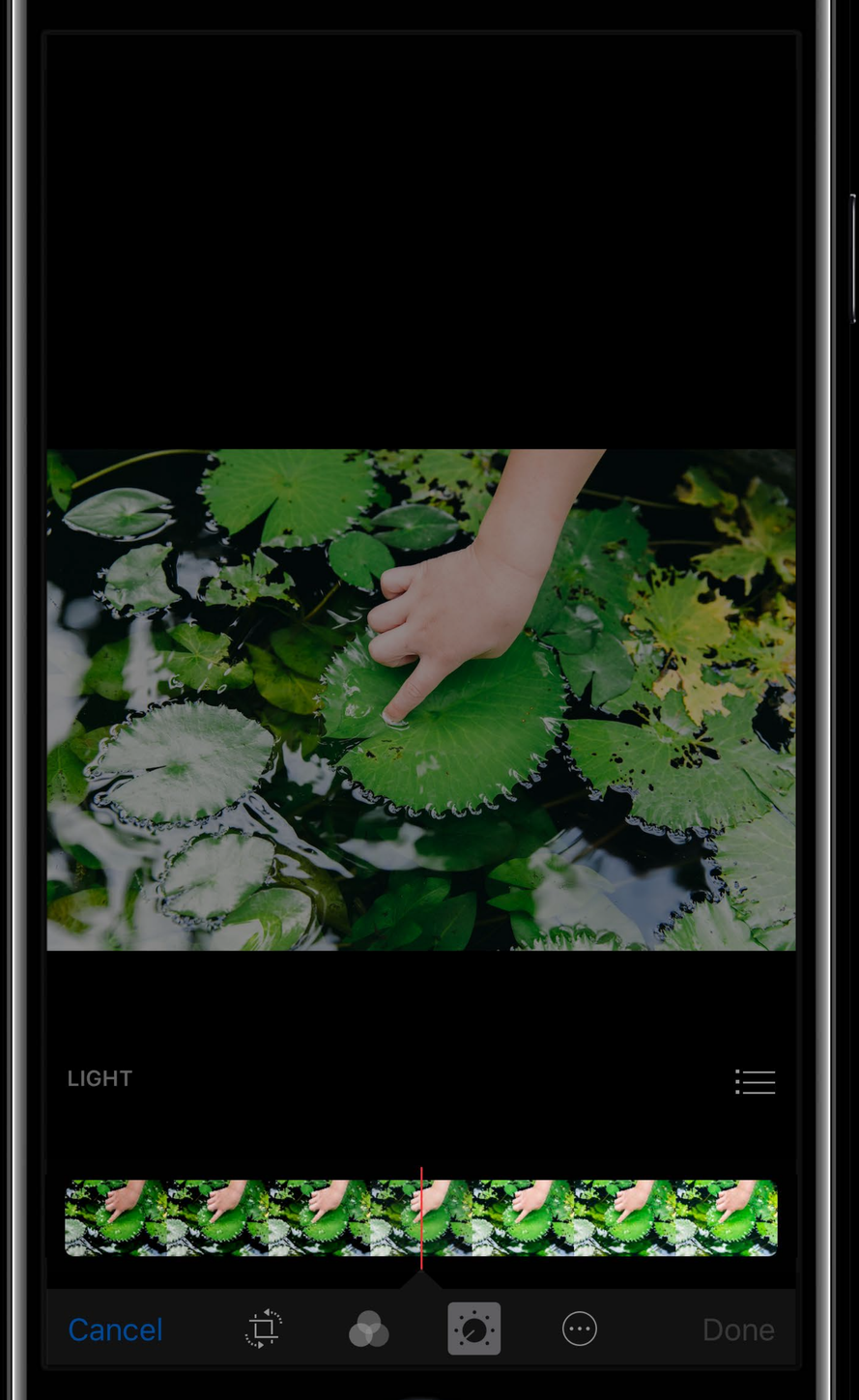
```
// Increment and Decrement Example
class StarsView : UIView {
    var numStars: Int = 0 {
        didSet {
            // Code for adjusting the star rating
        }
    }
    override var accessibilityTraits {
        get { return super.accessibilityTraits | UIAccessibilityTraitAdjustable }
        set { }
    }
    override func accessibilityIncrement() {
        numStars += 1
    }
    override func accessibilityDecrement() {
        numStars -= 1
    }
}
```

```
// Increment and Decrement Example
class StarsView : UIView {
    var numStars: Int = 0 {
        didSet {
            // Code for adjusting the star rating
        }
    }
    override var accessibilityTraits {
        get { return super.accessibilityTraits | UIAccessibilityTraitAdjustable }
        set { }
    }
    override func accessibilityIncrement() {
        numStars += 1
    }
    override func accessibilityDecrement() {
        numStars -= 1
    }
}
```

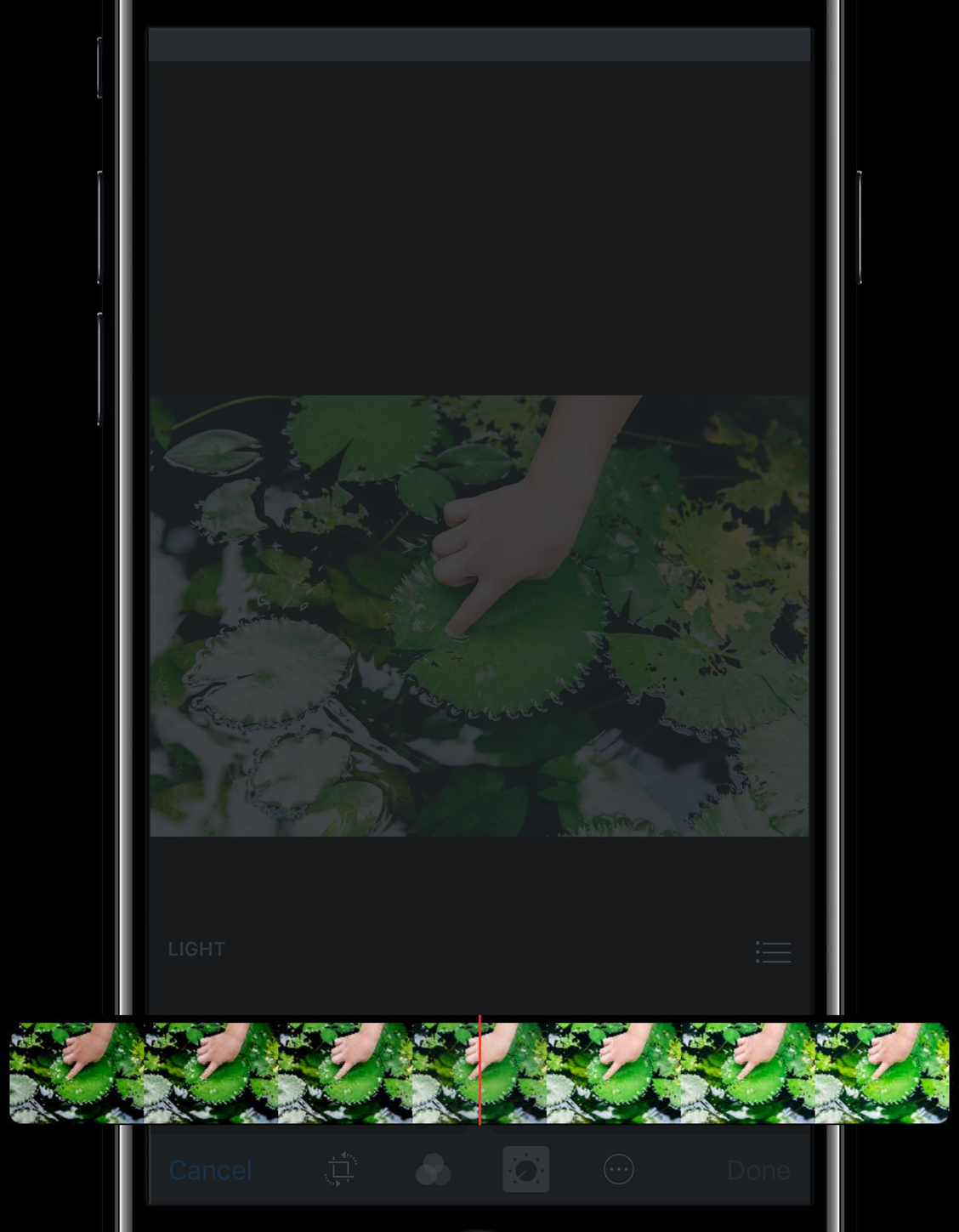
```
// Increment and Decrement Example
class StarsView : UIView {
    var numStars: Int = 0 {
        didSet {
            // Code for adjusting the star rating
        }
    }
    override var accessibilityTraits {
        get { return super.accessibilityTraits | UIAccessibilityTraitAdjustable }
        set { }
    }
    override func accessibilityIncrement() {
        numStars += 1
    }
    override func accessibilityDecrement() {
        numStars -= 1
    }
}
```

```
// Increment and Decrement Example
class StarsView : UIView {
    var numStars: Int = 0 {
        didSet {
            // Code for adjusting the star rating
        }
    }
    override var accessibilityTraits {
        get { return super.accessibilityTraits | UIAccessibilityTraitAdjustable }
        set { }
    }
    override func accessibilityIncrement() {
        numStars += 1
    }
    override func accessibilityDecrement() {
        numStars -= 1
    }
}
```

Pass Through



Pass Through



```
// Activation Point

extension NSObject {
     open var accessibilityActivationPoint: CGPoint
}
```

```
// Activation Point

extension NSObject {
     open var accessibilityActivationPoint: CGPoint
}
```

```
// Activation Point Example

class SliderView: UIView {

    let sliderNub: NubView

    override var accessibilityActivationPoint: CGPoint {
        get {
            return sliderNub.center
        }
        set { }
    }

}
```

```
// Activation Point Example

class SliderView: UIView {

    let sliderNub: NubView

    override var accessibilityActivationPoint: CGPoint {
        get {
            return sliderNub.center
        }
        set { }
    }

}
```

```
// Activation Point Example

class SliderView: UIView {

    let sliderNub: NubView

    override var accessibilityActivationPoint: CGPoint {
        get {
            return sliderNub.center
        }
        set { }
    }

}
```

Custom Scrolling

```swift
// Accessibility Scroll

extension NSObject {
    // Initiated with a 3-finger swipe
    open func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool
}

public enum UIAccessibilityScrollDirection : Int {
    case right
    case left
    case up
    case down
    case next
    case previous
}
```

```
// Accessibility Scroll

extension NSObject {
    // Initiated with a 3-finger swipe
    open func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool
}

public enum UIAccessibilityScrollDirection : Int {
    case right
    case left
    case up
    case down
    case next
    case previous
}
```

```
// Accessibility Scroll

extension NSObject {
    // Initiated with a 3-finger swipe
    open func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool
}
```

```
public enum UIAccessibilityScrollDirection : Int {
    case right
    case left
    case up
    case down
    case next
    case previous
}
```

```
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

```swift
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

```swift
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

```
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

```
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

```swift
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

```swift
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

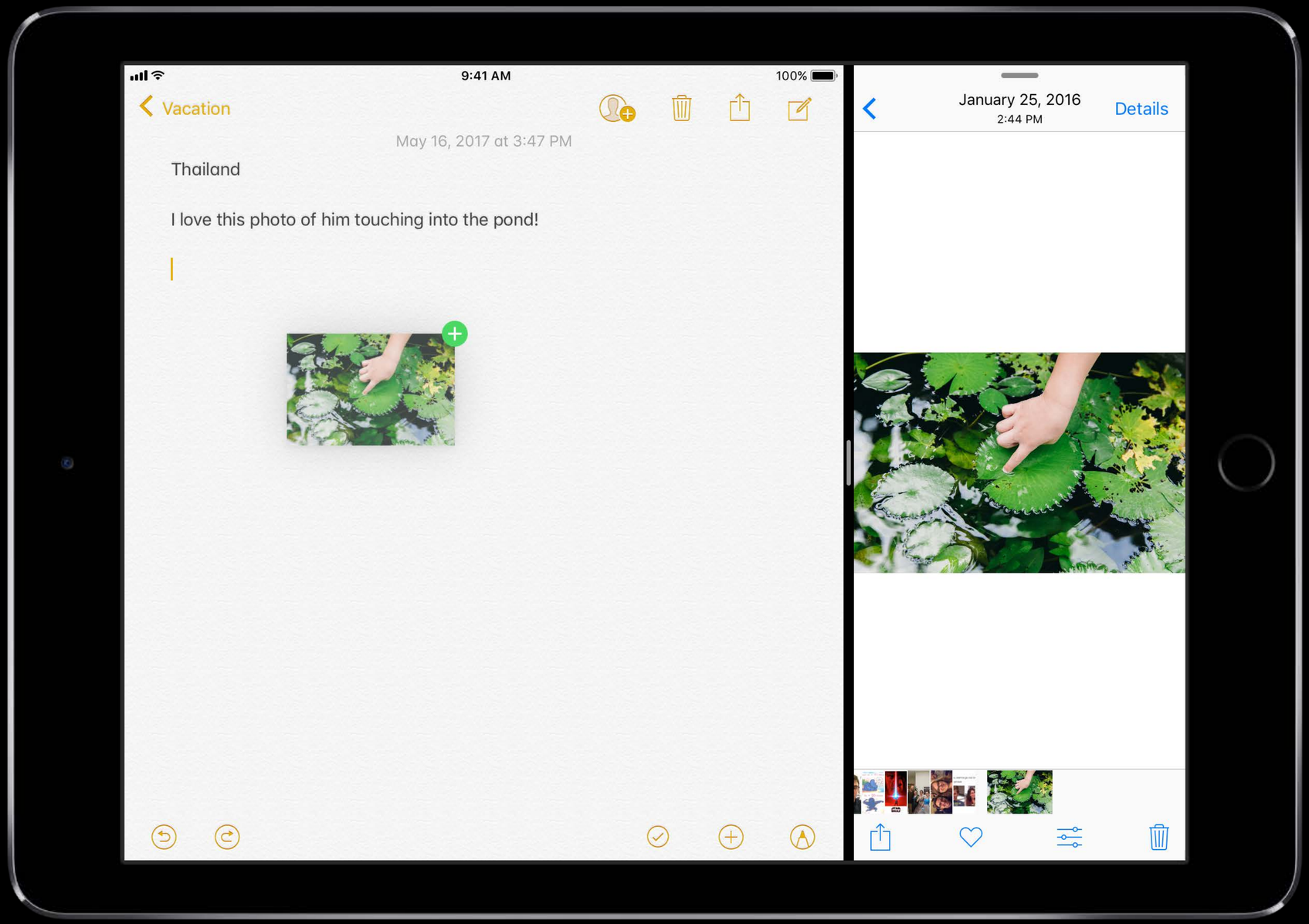
```
// Accessibility Scroll Example
class MyImageView : UIImageView {

    var canShowDetails: Bool = false

    func showDetails() {
        // Code for showing details
    }

    override func accessibilityScroll(_ direction: UIAccessibilityScrollDirection) -> Bool {
        if (canShowDetails && scrollDirection == .down) {
            showDetails()
            return true
        }
        return false
    }

}
```

More Information

https://developer.apple.com/sample-code/wwdc/2017/making-apps-accessible.zip

Drag and Drop Accessibility

NEW


9:41 AM
100%
Vacation
May 16, 2017 at 3:47 PM
Thailand
I love this photo of him touching into the pond!
January 25, 2016
2:44 PM
Details


NEW

NEW

Drag Sources

NEW

Drag Sources

Drop Points


9:41 AM
100%
Mail
Wednesday
24
Calendar
Photos
Camera
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
NEW


9:41 AM
100%
Mail
Wednesday
24
Calendar
Photos
Camera
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
NEW


9:41 AM
100%
Mail
Calendar
Wednesday
24
Photos
Camera
Maps
280
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
NEW


9:41 AM
100%
Mail
Wednesday
24
Calendar
Photos
Camera
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
NEW

```
// Drag and Drop Accessibility
```

NEW

```
extension NSObject {
    open var accessibilityDragSourceDescriptors: [UIAccessibilityLocationDescriptor]?
    open var accessibilityDropPointDescriptors: [UIAccessibilityLocationDescriptor]?
}

open class UIAccessibilityLocationDescriptor : NSObject {
    open var view: UIView
    open var point: CGPoint
    open var name: String
    open var attributedName: NSAttributedString

    public convenience init(name: String, view: UIView)
    public convenience init(name: String, point: CGPoint, in view: UIView)
    public init(attributedName: NSAttributedString, point: CGPoint, in view: UIView)
}
```

// Drag and Drop Accessibility

NEW

```
extension NSObject {
    open var accessibilityDragSourceDescriptors: [UIAccessibilityLocationDescriptor]?
    open var accessibilityDropPointDescriptors: [UIAccessibilityLocationDescriptor]?
}
```

```
open class UIAccessibilityLocationDescriptor : NSObject {
    open var view: UIView
    open var point: CGPoint
    open var name: String
    open var attributedName: NSAttributedString

    public convenience init(name: String, view: UIView)
    public convenience init(name: String, point: CGPoint, in view: UIView)
    public init(attributedName: NSAttributedString, point: CGPoint, in view: UIView)
}
```

```
// Drag and Drop Accessibility
```

NEW

```
extension NSObject {
    open var accessibilityDragSourceDescriptors: [UIAccessibilityLocationDescriptor]?
    open var accessibilityDropPointDescriptors: [UIAccessibilityLocationDescriptor]?
}

open class UIAccessibilityLocationDescriptor : NSObject {
    open var view: UIView
    open var point: CGPoint
    open var name: String
    open var attributedName: NSAttributedString

    public convenience init(name: String, view: UIView)
    public convenience init(name: String, point: CGPoint, in view: UIView)
    public init(attributedName: NSAttributedString, point: CGPoint, in view: UIView)
}
```

NEW

```
// Drag and Drop Example

class HomeView: UIView {

    func appView(at IndexPath: IndexPath) -> AppView {

        let appView = AppView()

        // Initialize app view

        let dragDescriptor = UIAccessibilityLocationDescriptor(name: “Drag \(appView.name)”,
                                point: appView.center, view: self)
        appView.accessibilityDragSourceDescriptors = [dragDescriptor]

        // Continued on next slide...
```

NEW

```
// Drag and Drop Example

class HomeView: UIView {

    func appView(at IndexPath: IndexPath) -> AppView {

        let appView = AppView()

        // Initialize app view

        let dragDescriptor = UIAccessibilityLocationDescriptor(name: "Drag \(appView.name)",
                                point: appView.center, view: self)
        appView.accessibilityDragSourceDescriptors = [dragDescriptor]

        // Continued on next slide...
```

NEW

```
// Drag and Drop Example

class HomeView: UIView {

    func appView(at IndexPath: IndexPath) -> AppView {

        let appView = AppView()

        // Initialize app view

        let dragDescriptor = UIAccessibilityLocationDescriptor(name: "Drag \(appView.name)",
                                point: appView.center, view: self)
        appView.accessibilityDragSourceDescriptors = [dragDescriptor]

        // Continued on next slide...
```

NEW

```
// Drag and Drop Example

class HomeView: UIView {

    func appView(at IndexPath: IndexPath) -> AppView {

        let appView = AppView()

        // Initialize app view

        let dragDescriptor = UIAccessibilityLocationDescriptor(name: “Drag \(appView.name)”,
                                point: appView.center, view: self)
        appView.accessibilityDragSourceDescriptors = [dragDescriptor]

        // Continued on next slide...
```

NEW

```
// Drag and Drop Example

class HomeView: UIView {

    func appView(at IndexPath: IndexPath) -> AppView {

        let appView = AppView()

        // Initialize app view

        let dragDescriptor = UIAccessibilityLocationDescriptor(name: "Drag \(appView.name)",
                                point: appView.center, view: self)
        appView.accessibilityDragSourceDescriptors = [dragDescriptor]

        // Continued on next slide...
```

NEW

```
// Drag and Drop Example, continued

        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                             \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                               \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                              \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

// Drag and Drop Example, continued

NEW

```
        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                             \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                               \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                              \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

NEW

```
// Drag and Drop Example, continued

        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                               \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                                \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                               \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

NEW

```
// Drag and Drop Example, continued

        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                               \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                                 \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                               \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

NEW

```
// Drag and Drop Example, continued

        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                             \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                               \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                              \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

NEW

```
// Drag and Drop Example, continued

        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                             \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                               \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                              \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

NEW

```
// Drag and Drop Example, continued

        let leftPoint = CGPoint(x: appView.frame.origin.x, y: appView.center.y)
        let rightPoint = CGPoint(x: appView.frame.origin.x + appView.frame.size.width,
                                 y: appView.center.y)

        let leftDescriptor = UIAccessibilityLocationDescriptor(name: "Drop left of
                             \(appView.name)", point: leftPoint, view: self)
        let folderDescriptor = UIAccessibilityLocationDescriptor(name: "Create folder with
                               \(appView.name)", point: appView.center, view: self)
        let rightDescriptor = UIAccessibilityLocationDescriptor(name: "Drop right of
                              \(appView.name)", point, rightPoint, view: self)

        appView.accessibilityDropPointDescriptors = [leftDescriptor, folderDescriptor,
                                                     rightDescriptor]
        return appView
    }
}
```

Summary

New features

Auditing

App accessibility

People


S

Thank you!

More Information

https://developer.apple.com/wwdc17/215

Related Sessions

Media and Gaming Accessibility	Grand Ballroom A	Wednesday 3:10PM
Design For Everyone	Hall 3	Thursday 11:00AM
Auto Layout Techniques in Interface Builder	Hall 3	Friday 9:00AM
Building Apps with Dynamic Type	Executive Ballroom	Friday 1:50PM

Labs

Accessibility Lab	Technology Lab J	Wed 4:10PM-6:00PM
Accessibility and Dynamic Type Lab	Technology Lab C	Fri 2:30PM-4:00PM
Accessibility Design by Appointment Lab	User Interface Design Lab B	Thur 9:00AM–6:00PM
Accessibility Design by Appointment Lab	User Interface Design Lab B	Fri 9:00AM–1:15PM

Other Events

Accessible Technology and Inclusive Design Get-Together	The Hub	Wednesday 6:30PM


WWDC17