What's New in ResearchKit

Session 234

Bunny Laden Health Team
Yuan Zhu Health Team

9:41 AM 100%

DukeMedicine

Autism & Beyond

A Study of Young Children's Mental Health

Read Consent Document

Email Consent Document

Swipe to learn more

Join Study

Already Participating?

9:41 AM 100%

Back Viewing Instructions Next

Sit your child on your lap.

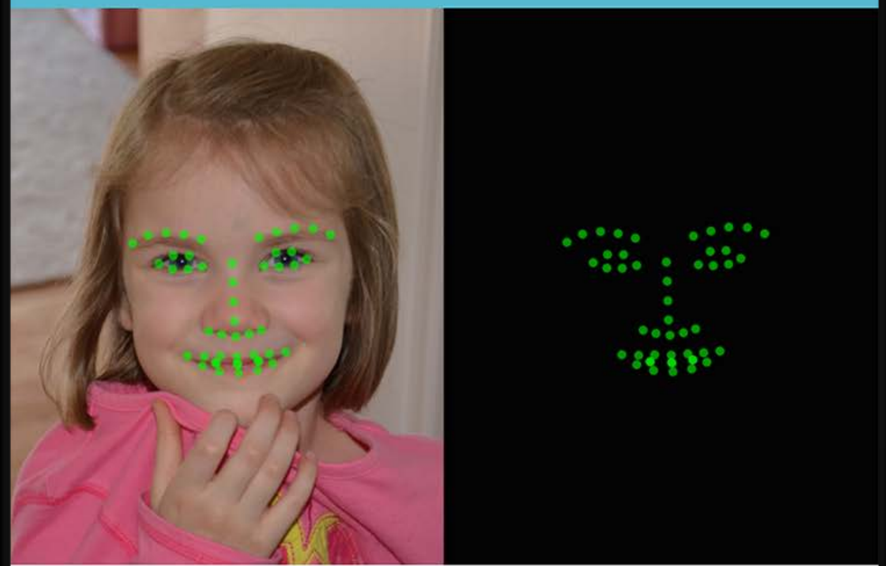
Remove pacifiers or other objects that may block your child's face. We can detect your child's facial expressions if they're wearing glasses but not if something is covering their mouth.

Try not to tell your child what to do. We want to see how your child acts on his or her own.

9:41 AM 100%

DukeMedicine

How it Works



While your child watches a video on your iPhone, we will record your child's face to measure emotion.

You can choose to share the video or only the facial expressions.

Join Study

Already Participating?

Agenda

New Features

Agenda

New Features

Open Source Community

Agenda

Recap

New Features

Open Source Community

Basic Components of Clinical Research

Basic Components of Clinical Research

Informed Consent

Basic Components of Clinical Research

Informed Consent

Survey Tasks

Basic Components of Clinical Research

Informed Consent

Survey Tasks

Active Tasks

Informed Consent

Autism & Beyond

Duke University Health System

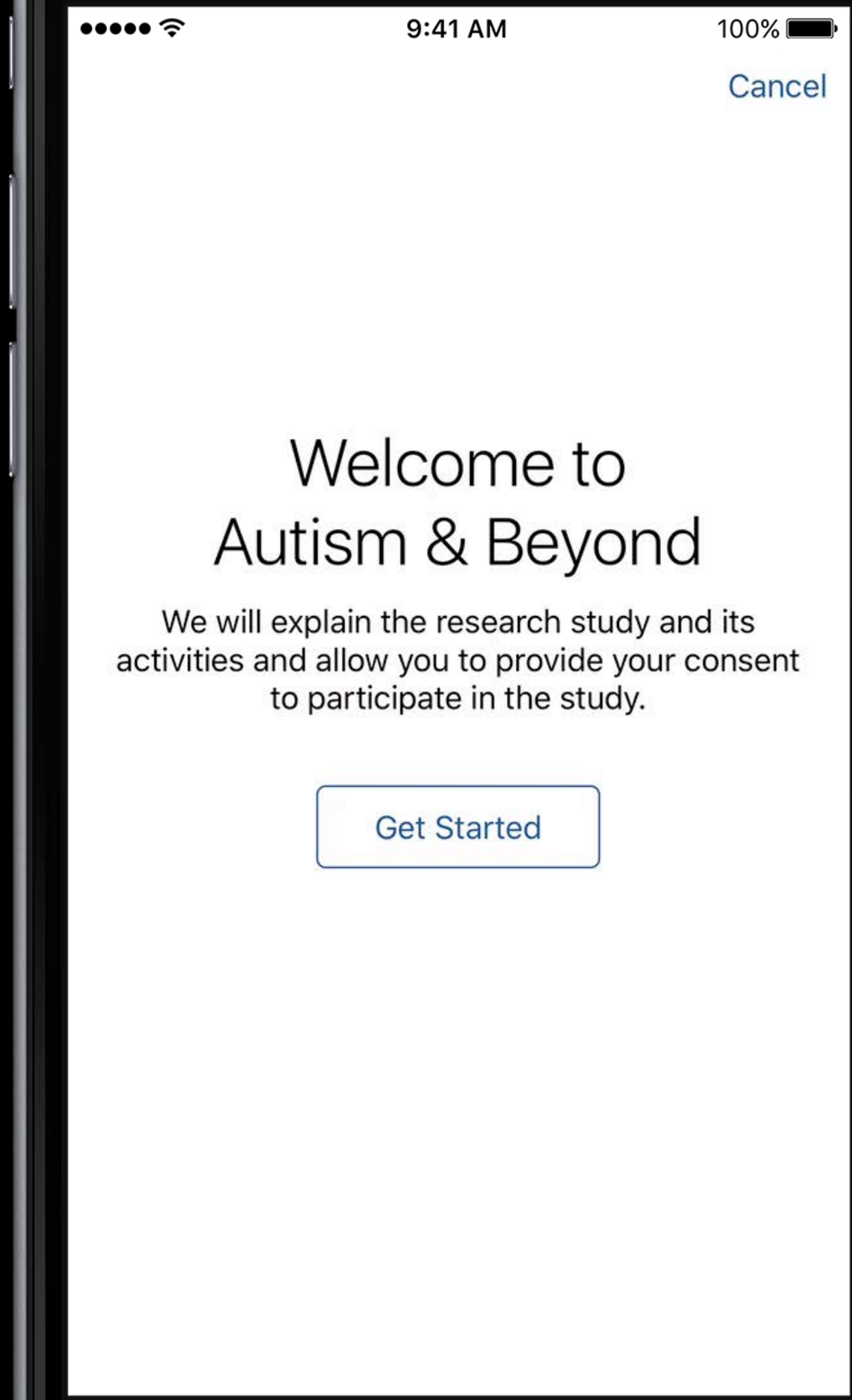
Consent to Participate in a Research Study
Autism & Beyond

Landmark Consent

SUMMARY

You are being asked to participate in the Autism & Beyond research study because you:
• Are an adult, at least 18 years old;
• Have a child under 6 years old (less than 72 months of age)
• You are the parent or legal guardian of this child;
• Speak and read English;
• Live in the United States of American; and,
• Have an iPhone you are willing to use to participate in this study.

Your participation in this study is entirely voluntary; research studies include only people who choose to take part. To be in a research study you must give your informed consent. The purpose

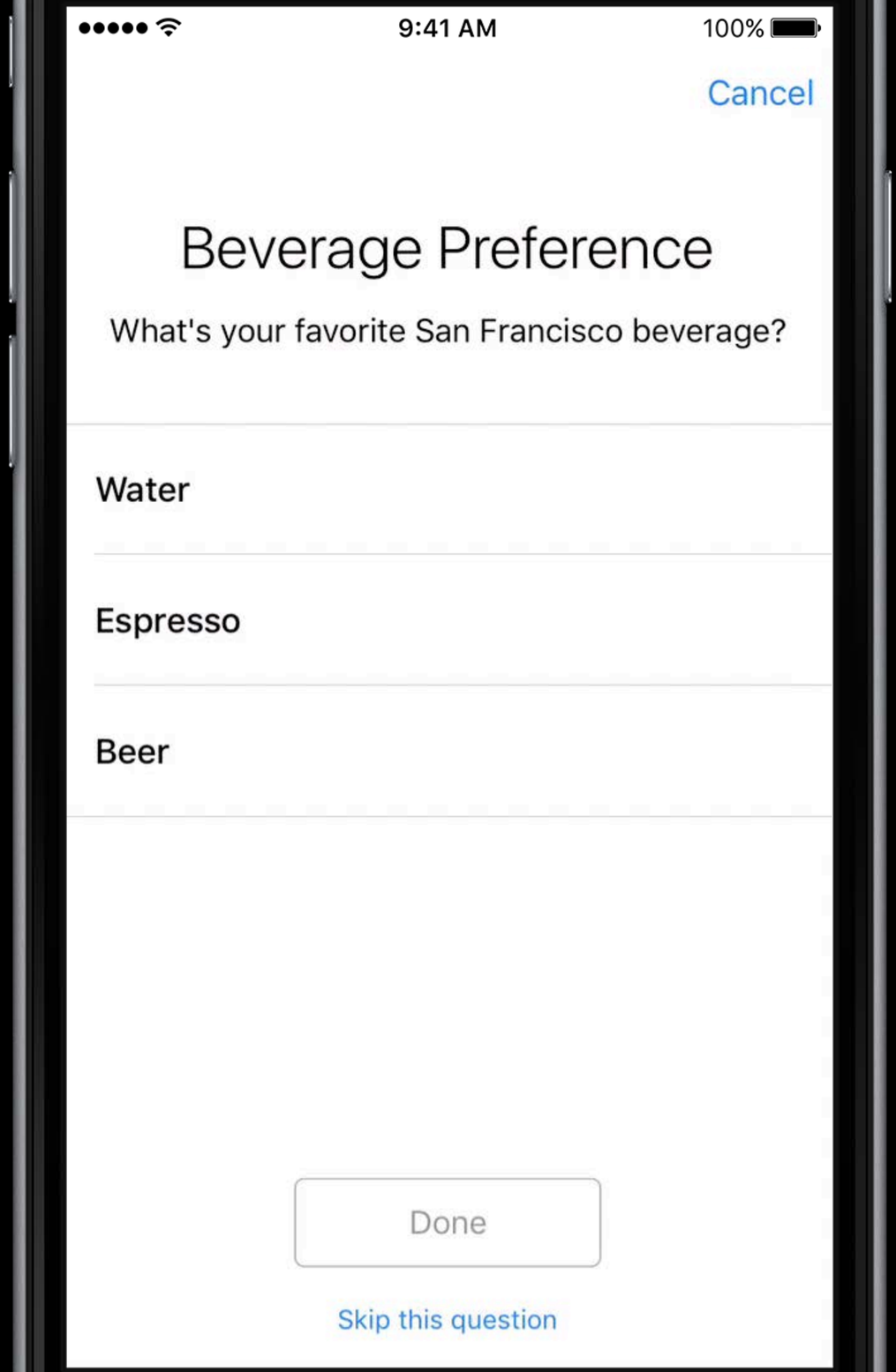

9:41 AM
100%
Cancel
Welcome to
Autism & Beyond
We will explain the research study and its activities and allow you to provide your consent to participate in the study.
Get Started

Survey Tasks

Patient Health Questionnaire (PHQ-9)

Patient Name: _____ **Date:** _____

	Not at all	Several days	More than half the days	Nearly every day
1. Over the *last 2 weeks*, how often have you been bothered by any of the following problems?				
a. Little interest or pleasure in doing things	☐	☐	☐	☐
b. Feeling down, depressed, or hopeless	☐	☐	☐	☐
c. Trouble falling/staying asleep, sleeping too much	☐	☐	☐	☐
d. Feeling tired or having little energy	☐	☐	☐	☐
e. Poor appetite or overeating	☐	☐	☐	☐
f. Feeling bad about yourself or that you are a failure or have let yourself or your family down	☐	☐	☐	☐
g. Trouble concentrating on things, such as reading the newspaper or watching television.	☐	☐	☐	☐
h. Moving or speaking so slowly that other people could have noticed. Or the opposite; being so fidgety or restless that you have been moving around a lot more than usual.	☐	☐	☐	☐
i. Thoughts that you would be better off dead or of hurting yourself in some way.	☐	☐	☐	☐
2. If you checked off any problem on this questionnaire so far, how difficult have these problems made it for you to do your work, take care of things at home, or get along with other people?	Not difficult at all	Somewhat difficult	Very difficult	Extremely difficult
	☐	☐	☐	☐


9:41 AM
100%
Cancel
Beverage Preference
What's your favorite San Francisco beverage?
Water
Espresso
Beer
Done
Skip this question

Active Tasks

9:41 AM

100%

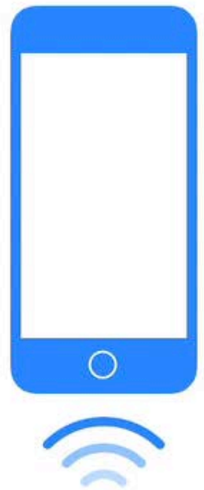
Step 1 of 5

Cancel

Voice

Everyone's voice has unique characteristics.

This activity evaluates your voice by recording it with the microphone at the bottom of your phone.



Next

9:41 AM 100%

Step 1 of 4 Cancel

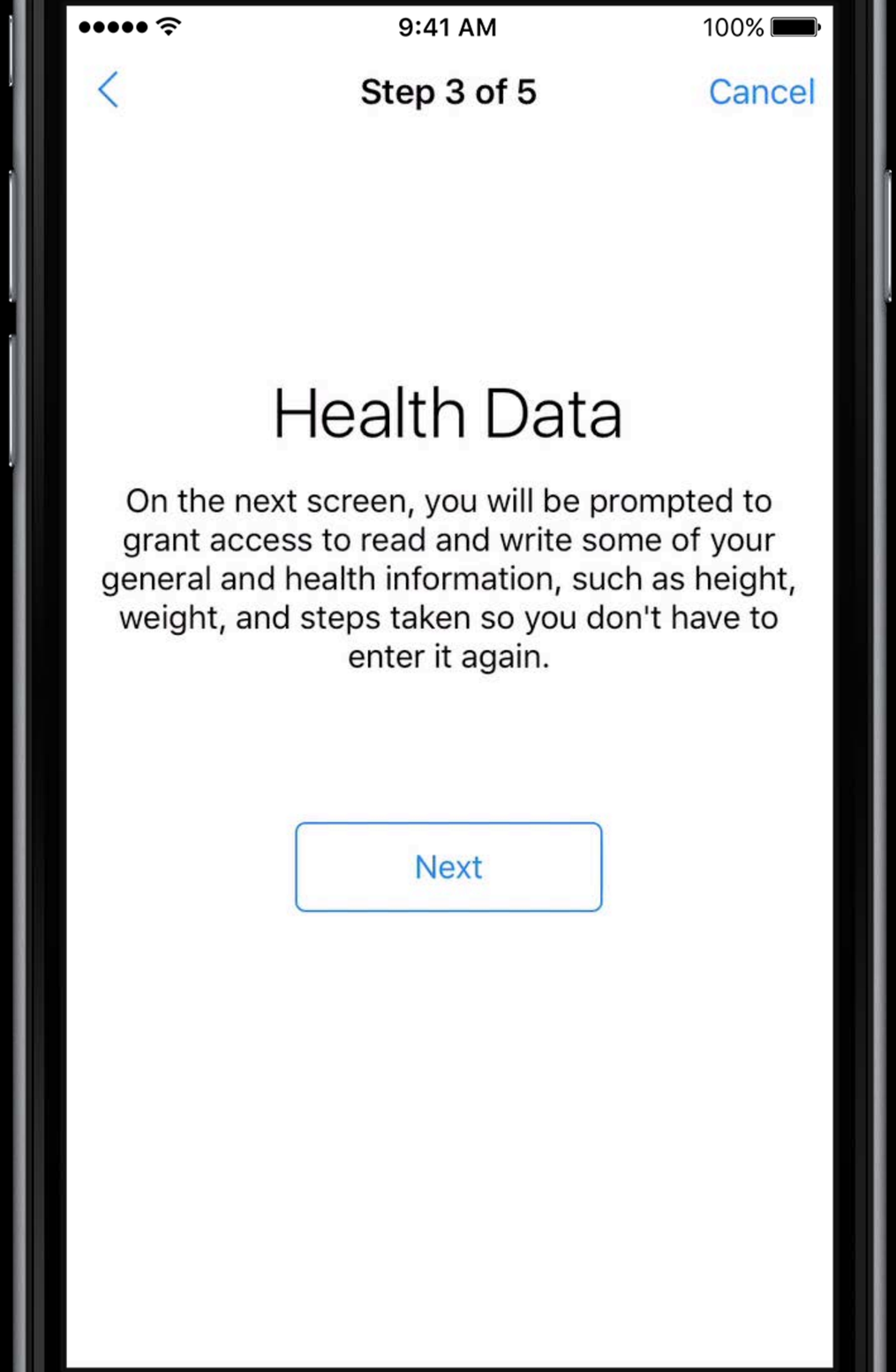
Tapping Speed

Finger tapping is a universal way to communicate.

This activity measures your tapping speed.

Next

Integration with HealthKit


9:41 AM
100%
Step 3 of 5
Cancel
Health Data
On the next screen, you will be prompted to grant access to read and write some of your general and health information, such as height, weight, and steps taken so you don't have to enter it again.
Next

Data Storage

Data Storage and ResearchKit

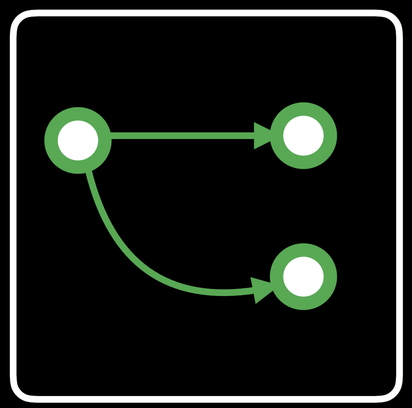
New Features

Yuan Zhu
Software Engineer

New Features

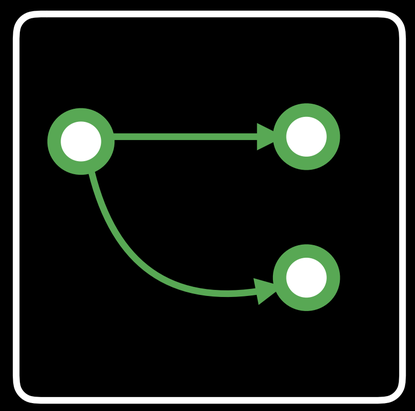
New Features

Branching Task



New Features

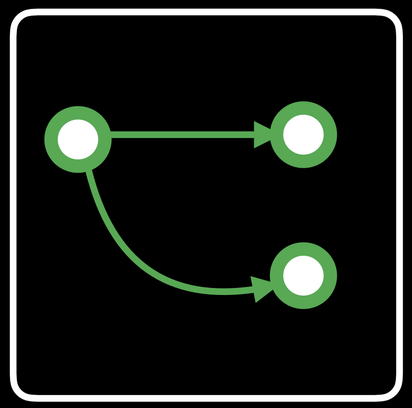
Branching Task



App Access

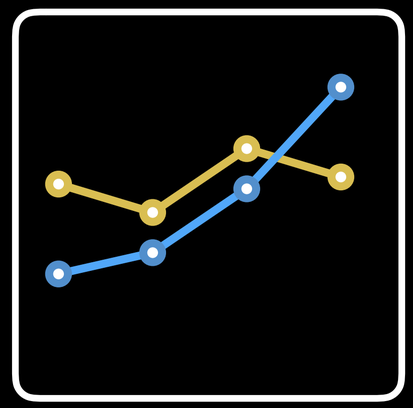
New Features

Branching Task



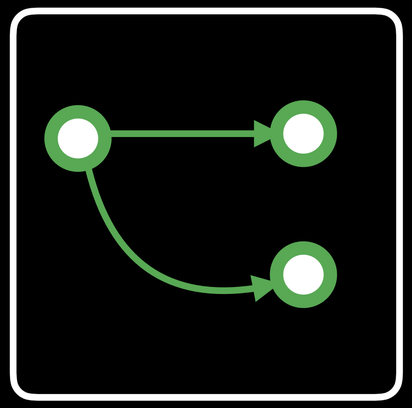
App Access

Charts



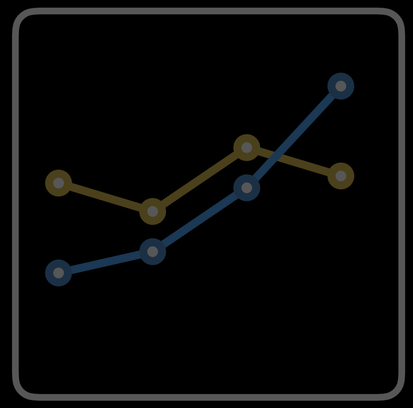
New Features

Branching Task



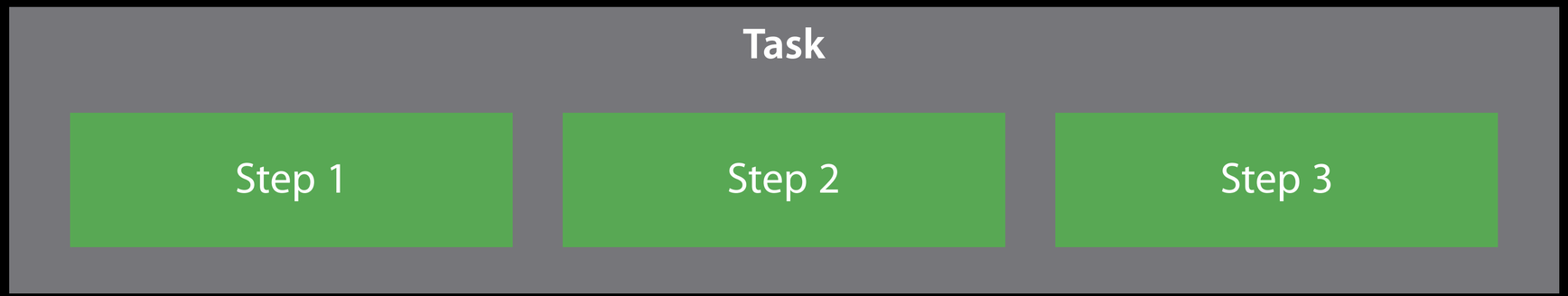
App Access

Charts



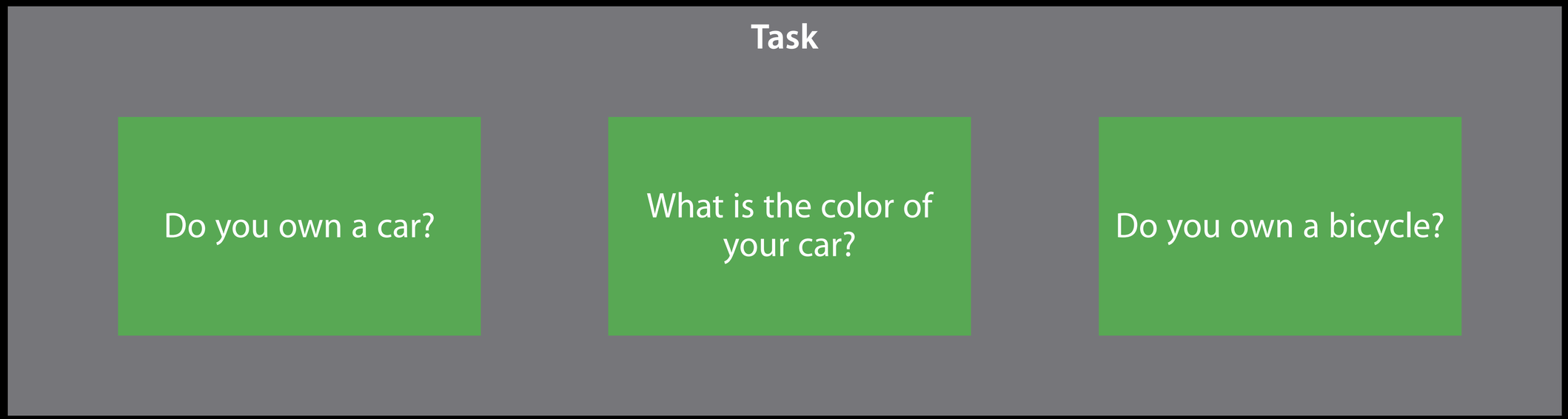
Branching Task

Task and step model



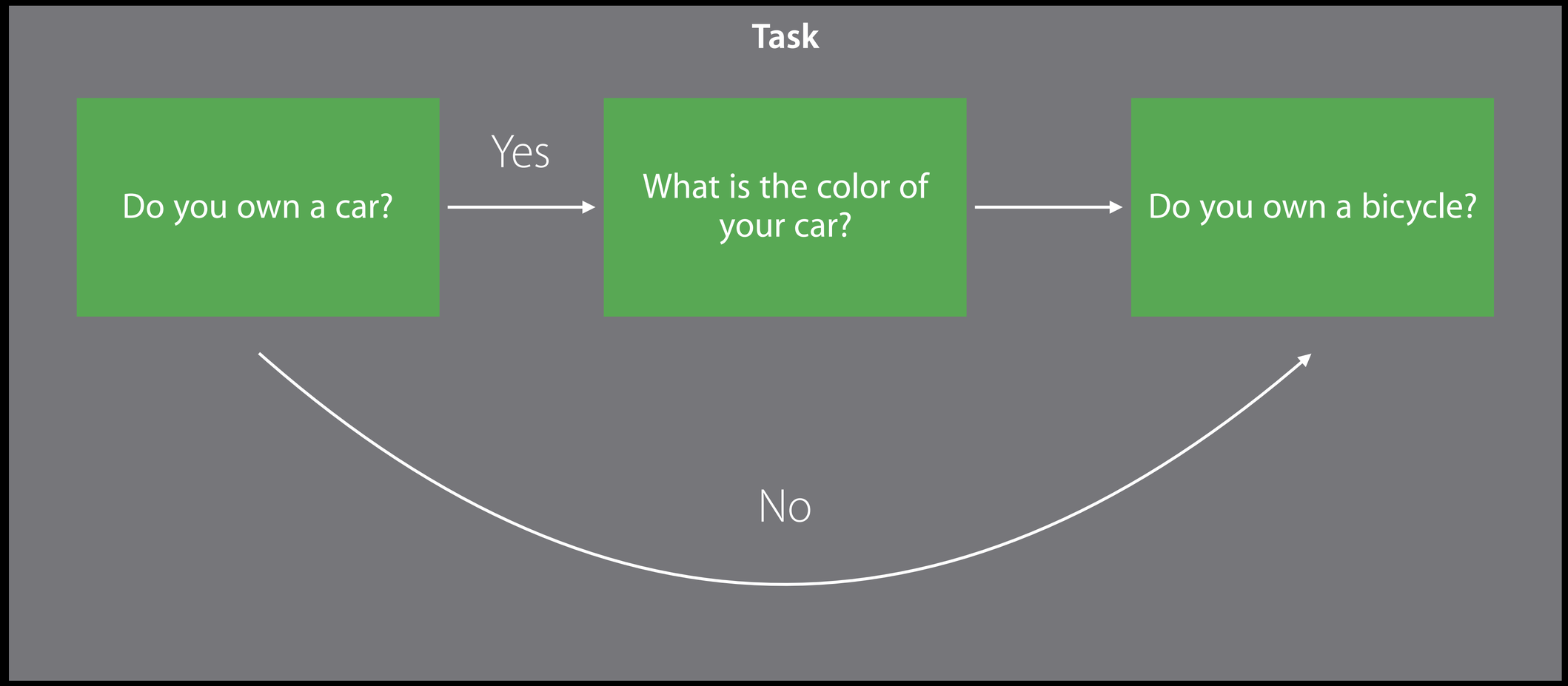
Branching Task

Example



Branching Task

Example

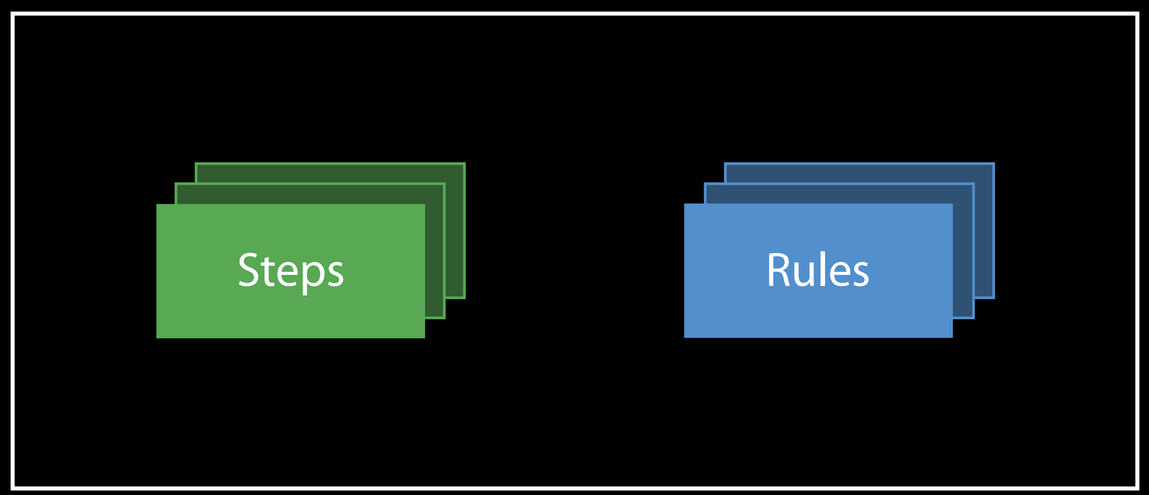


Branching Task

NEW

Branching Task

NEW



Navigable Task

Build task with steps

```
// Question steps
let ownCarStep = ORKQuestionStep(identifier: “own_a_car”, title: "Do you own a car?",
answer: ORKBooleanAnswerFormat())
let carColorStep = ORKQuestionStep(identifier: “car_color”, title: "What is the color of
your car?", answer: ORKTextAnswerFormat())
let ownBicycleStep = ORKQuestionStep(identifier: “own_a_bicycle”, title: "Do you own a
bicycle?", answer: ORKBooleanAnswerFormat())

// Form a task
let task = ORKNavigableOrderedTask(identifier: "questions", steps: [ownCarStep,
carColorStep, ownBicycleStep])
```

Navigable Task

Build task with steps

```
// Question steps
let ownCarStep = ORKQuestionStep(identifier: “own_a_car”, title: "Do you own a car?",
answer: ORKBooleanAnswerFormat())
let carColorStep = ORKQuestionStep(identifier: “car_color”, title: "What is the color of
your car?", answer: ORKTextAnswerFormat())
let ownBicycleStep = ORKQuestionStep(identifier: “own_a_bicycle”, title: "Do you own a
bicycle?", answer: ORKBooleanAnswerFormat())

// Form a task
let task = ORKNavigableOrderedTask(identifier: "questions", steps: [ownCarStep,
carColorStep, ownBicycleStep])
```

Navigable Task

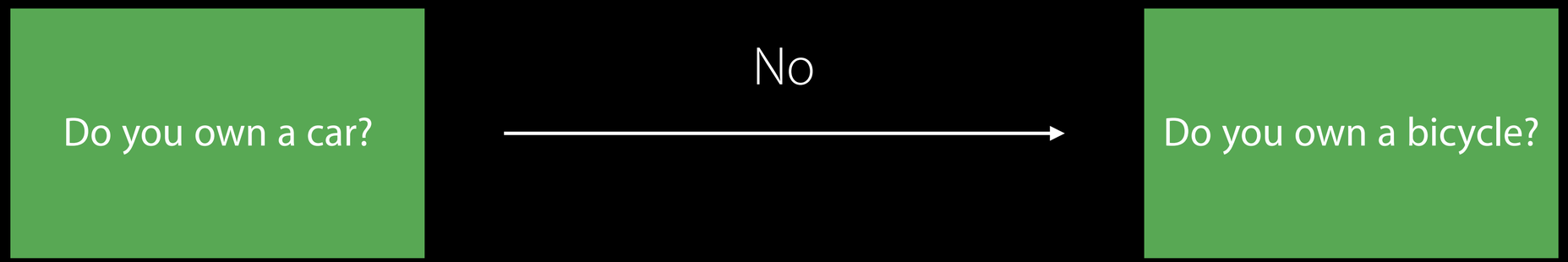
Build task with steps

```
// Question steps
let ownCarStep = ORKQuestionStep(identifier: “own_a_car”, title: "Do you own a car?",
answer: ORKBooleanAnswerFormat())
let carColorStep = ORKQuestionStep(identifier: “car_color”, title: "What is the color of
your car?", answer: ORKTextAnswerFormat())
let ownBicycleStep = ORKQuestionStep(identifier: “own_a_bicycle”, title: "Do you own a
bicycle?", answer: ORKBooleanAnswerFormat())

// Form a task
let task = ORKNavigableOrderedTask(identifier: "questions", steps: [ownCarStep,
carColorStep, ownBicycleStep])
```

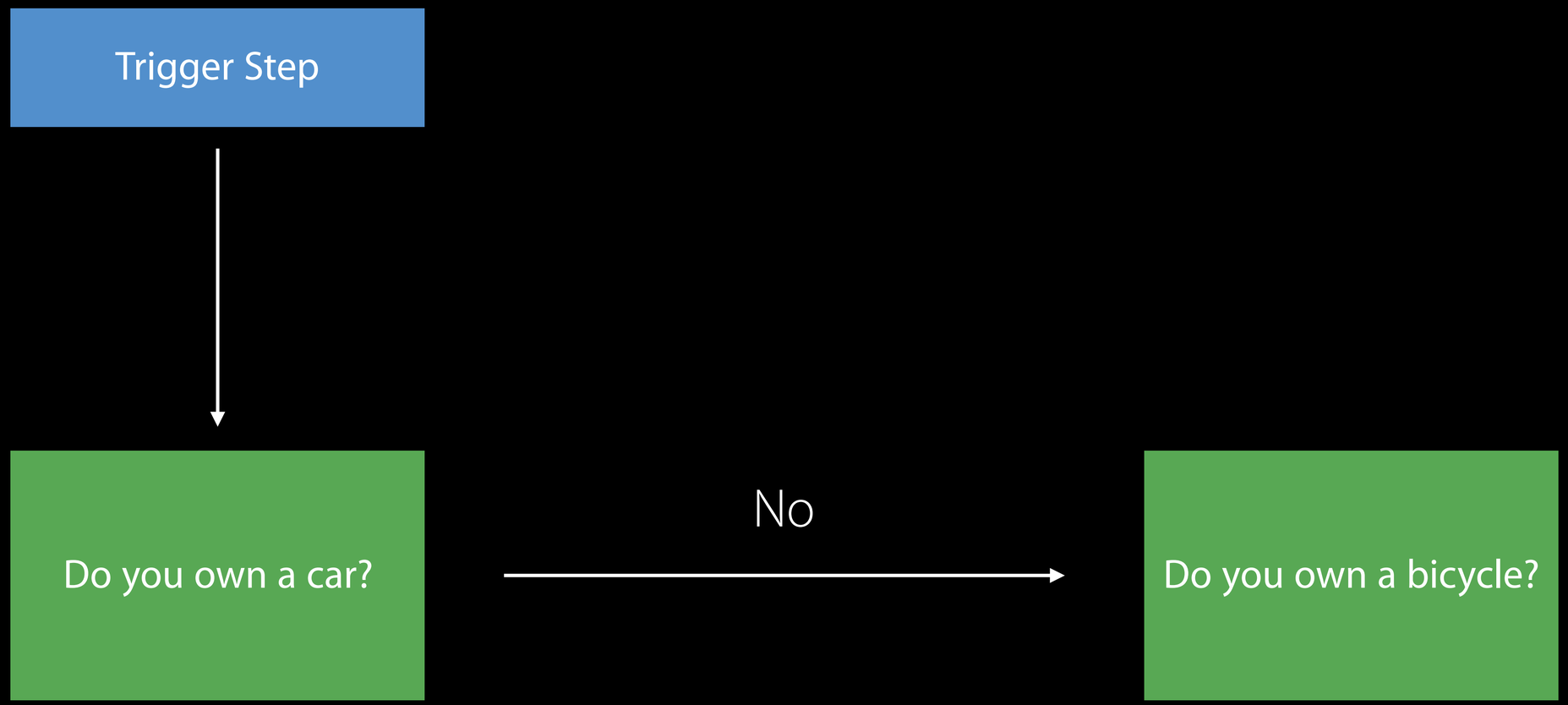
Branching Task

Navigation rule



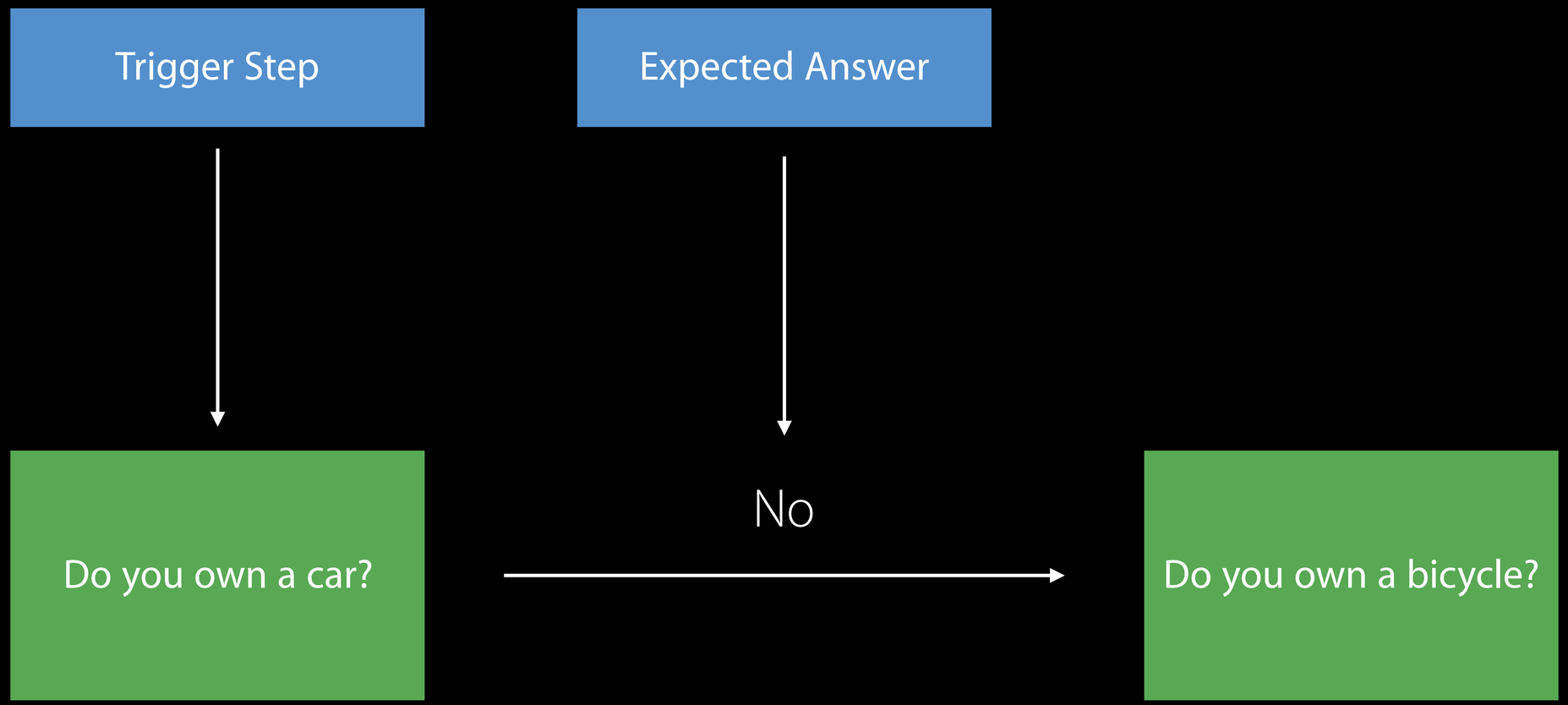
Branching Task

Navigation rule



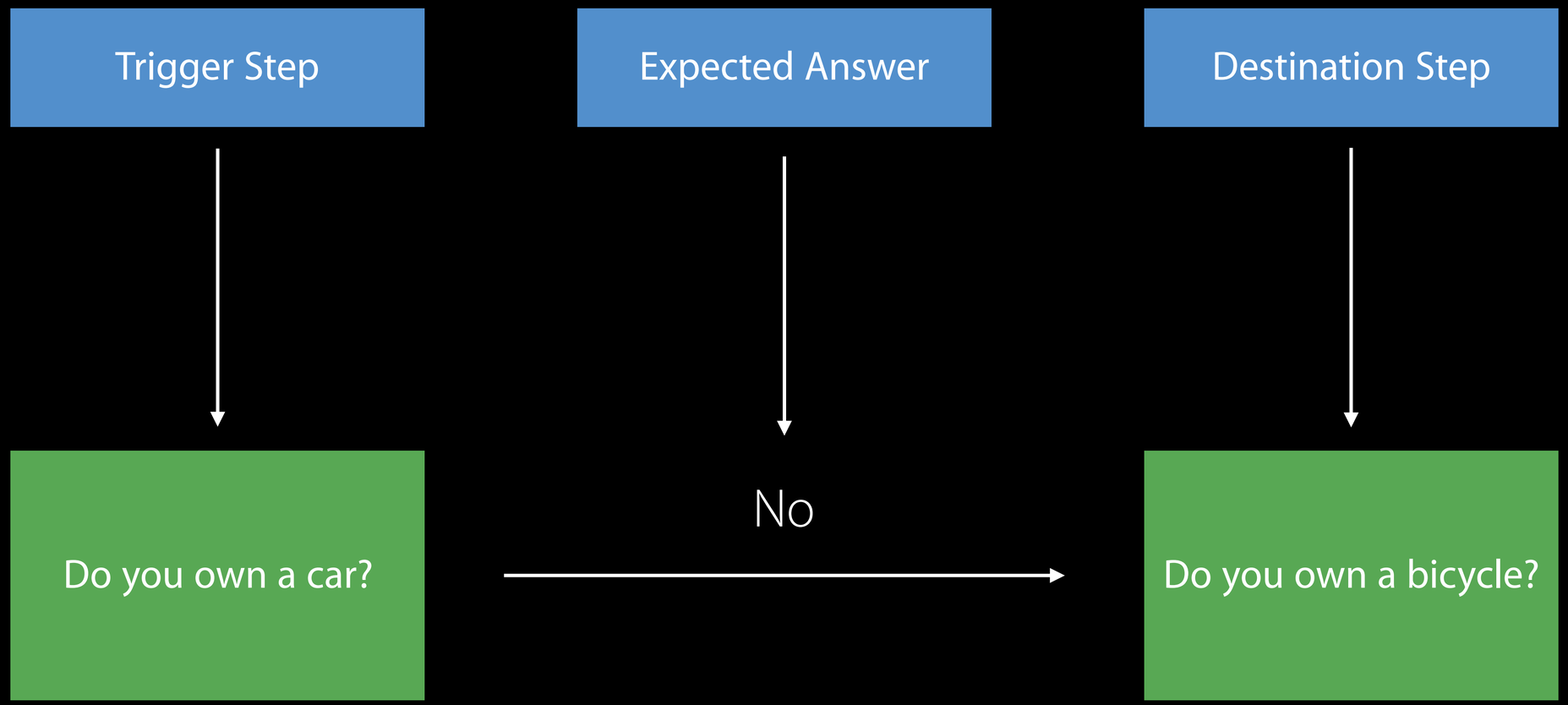
Branching Task

Navigation rule



Branching Task

Navigation rule



Branching Task

Navigation rule

```
let resultSelector = ORKResultSelector(resultIdentifier:ownCarStep.identifier)
let predicate = ORKResultPredicate.predicateForBooleanQuestionResult(with: resultSelector,
expectedAnswer: false)
let navigationRule =
ORKPredicateStepNavigationRule(resultPredicatesAndDestinationStepIdentifiers: [(predicate,
ownBicycleStep.identifier)])
task.setNavigationRule(navigationRule, forTriggerStepIdentifier: ownCarStep.identifier)
```

Branching Task

Navigation rule

Expected Answer

```
let resultSelector = ORKResultSelector(resultIdentifier:ownCarStep.identifier)
let predicate = ORKResultPredicate.predicateForBooleanQuestionResult(with: resultSelector,
expectedAnswer: false)
let navigationRule =
ORKPredicateStepNavigationRule(resultPredicatesAndDestinationStepIdentifiers: [(predicate,
ownBicycleStep.identifier)])
task.setNavigationRule(navigationRule, forTriggerStepIdentifier: ownCarStep.identifier)
```

Branching Task

Navigation rule

Expected Answer

Destination Step

```
let resultSelector = ORKResultSelector(resultIdentifier:ownCarStep.identifier)
let predicate = ORKResultPredicate.predicateForBooleanQuestionResult(with: resultSelector,
expectedAnswer: false)
let navigationRule =
ORKPredicateStepNavigationRule(resultPredicatesAndDestinationStepIdentifiers: [(predicate,
ownBicycleStep.identifier)])
task.setNavigationRule(navigationRule, forTriggerStepIdentifier: ownCarStep.identifier)
```

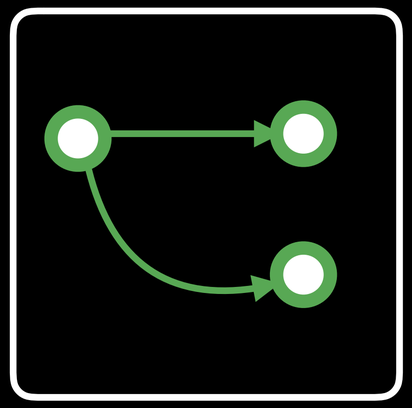
Branching Task

Navigation rule


```
let resultSelector = ORKResultSelector(resultIdentifier:ownCarStep.identifier)
let predicate = ORKResultPredicate.predicateForBooleanQuestionResult(with: resultSelector,
expectedAnswer: false)
let navigationRule =
ORKPredicateStepNavigationRule(resultPredicatesAndDestinationStepIdentifiers: [(predicate,
ownBicycleStep.identifier)])
task.setNavigationRule(navigationRule, forTriggerStepIdentifier: ownCarStep.identifier)
```

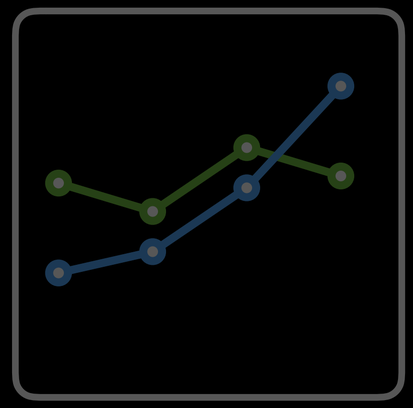
New Features

Branching Task



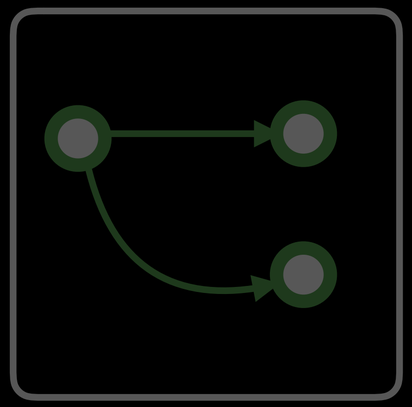
App Access

Charts



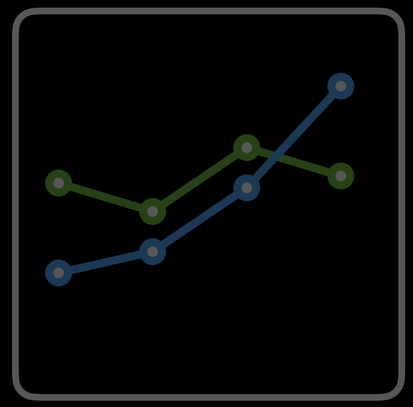
New Features

Branching Task



App Access

Charts

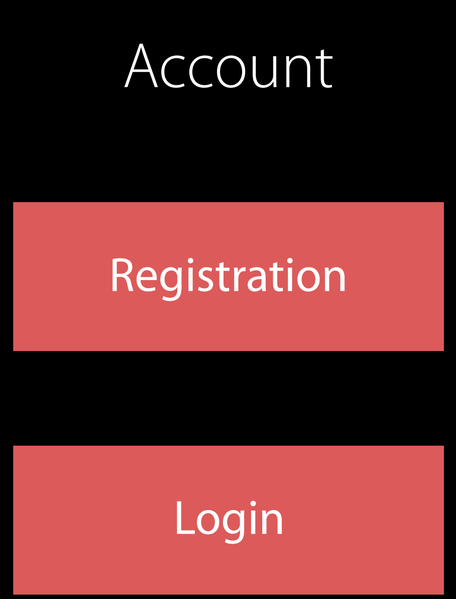


App Access

Overview

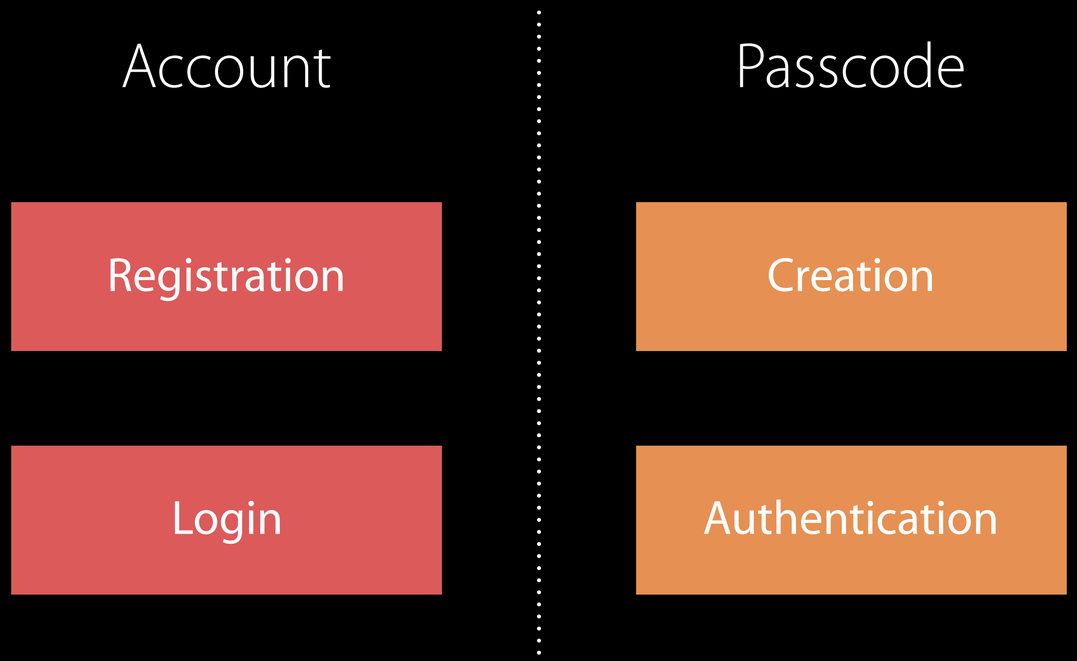
App Access

Overview



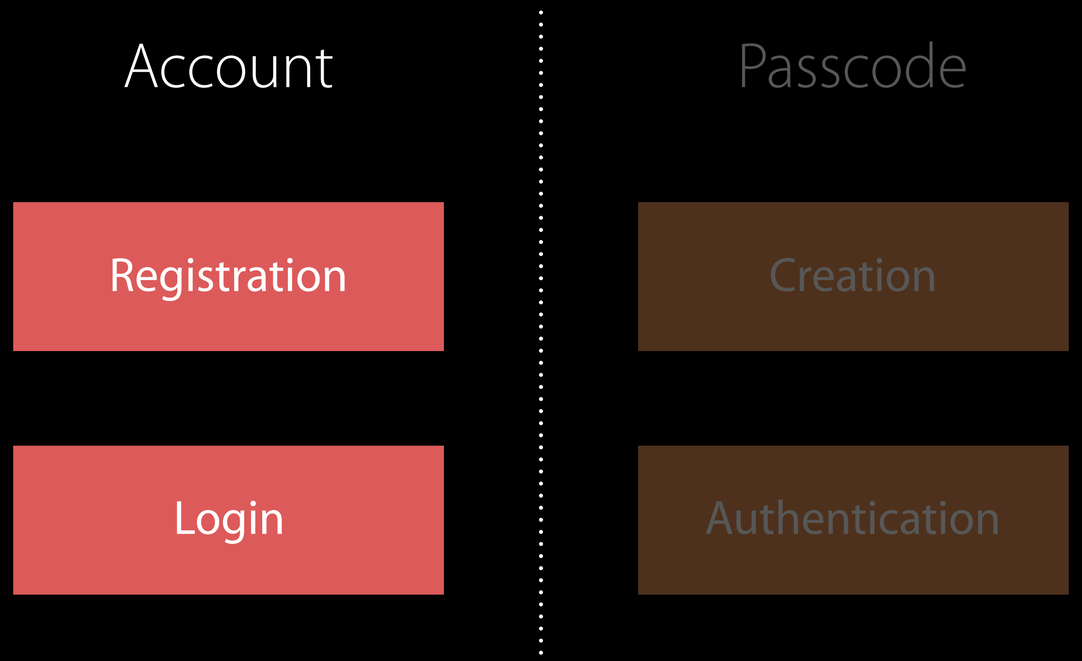
App Access

Overview



App Access

Overview



9:41 AM

100%

Cancel

Registration

Email jappleseed@example.com

Password Enter password

Confirm Enter password again

Next

App Access

Account registration

NEW

```
let registrationStep = ORKRegistrationStep(identifier: "registration", title:
"Registration", text: nil, options: [])
// Drop it into a task and present
```

App Access

Account registration

NEW

```
let registrationStep = ORKRegistrationStep(identifier: "registration", title:
"Registration", text: nil, options: [])
// Drop it into a task and present
```

```
// Retrieve the username and password
let username = (loginStepResult?.result(forIdentifier: ORKLoginFormItemIdentifierEmail) as!
ORKTextQuestionResult).answer

let password = (loginStepResult?.result(forIdentifier: ORKLoginFormItemIdentifierPassword)
as! ORKTextQuestionResult).answer

// Send to remote server ...
```

App Access

Account

Registration

Login

Passcode

Creation

Authentication

9:41 AM

100%

Cancel

User Login

Email jappleseed@example.com

Password Enter password

Login

Forgot password?

App Access

Account login

NEW

```
let loginStep = ORKLoginStep(
        identifier: "login",
        title: "User Login",
        text: "",
        loginViewControllerClass:LoginViewController.self)
```

App Access

Account login

NEW

```
let loginStep = ORKLoginStep(
        identifier: "login",
        title: "User Login",
        text: "",
        loginViewControllerClass:LoginViewController.self)
```

```
// Forgot password handling
class LoginViewController: ORKLoginStepViewController {
    override func forgotPasswordButtonTapped () {
     ...
    }
}
```

App Access

Account

Registration

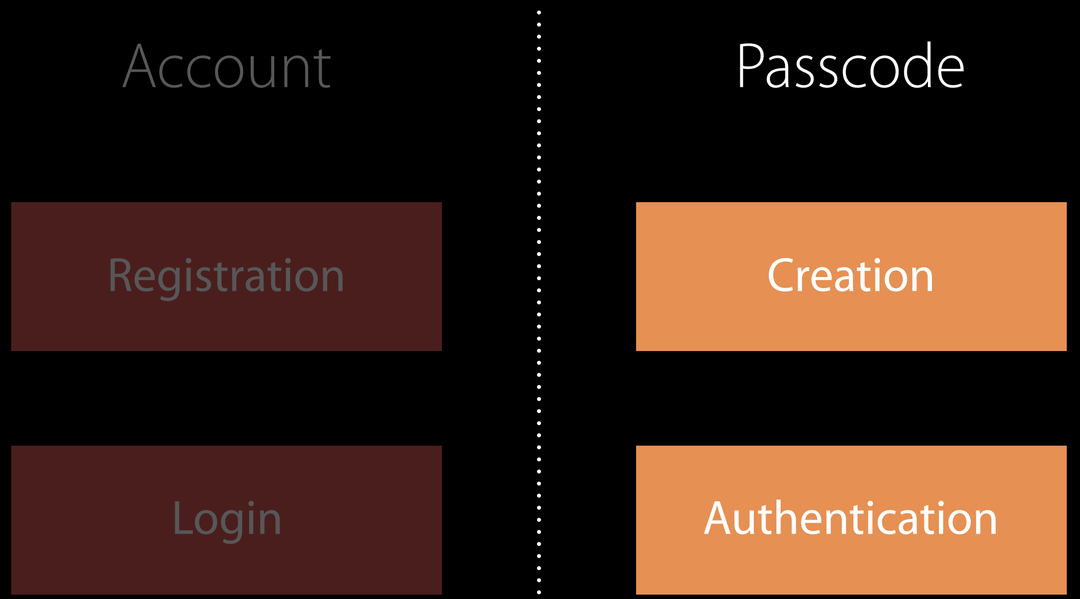
Login

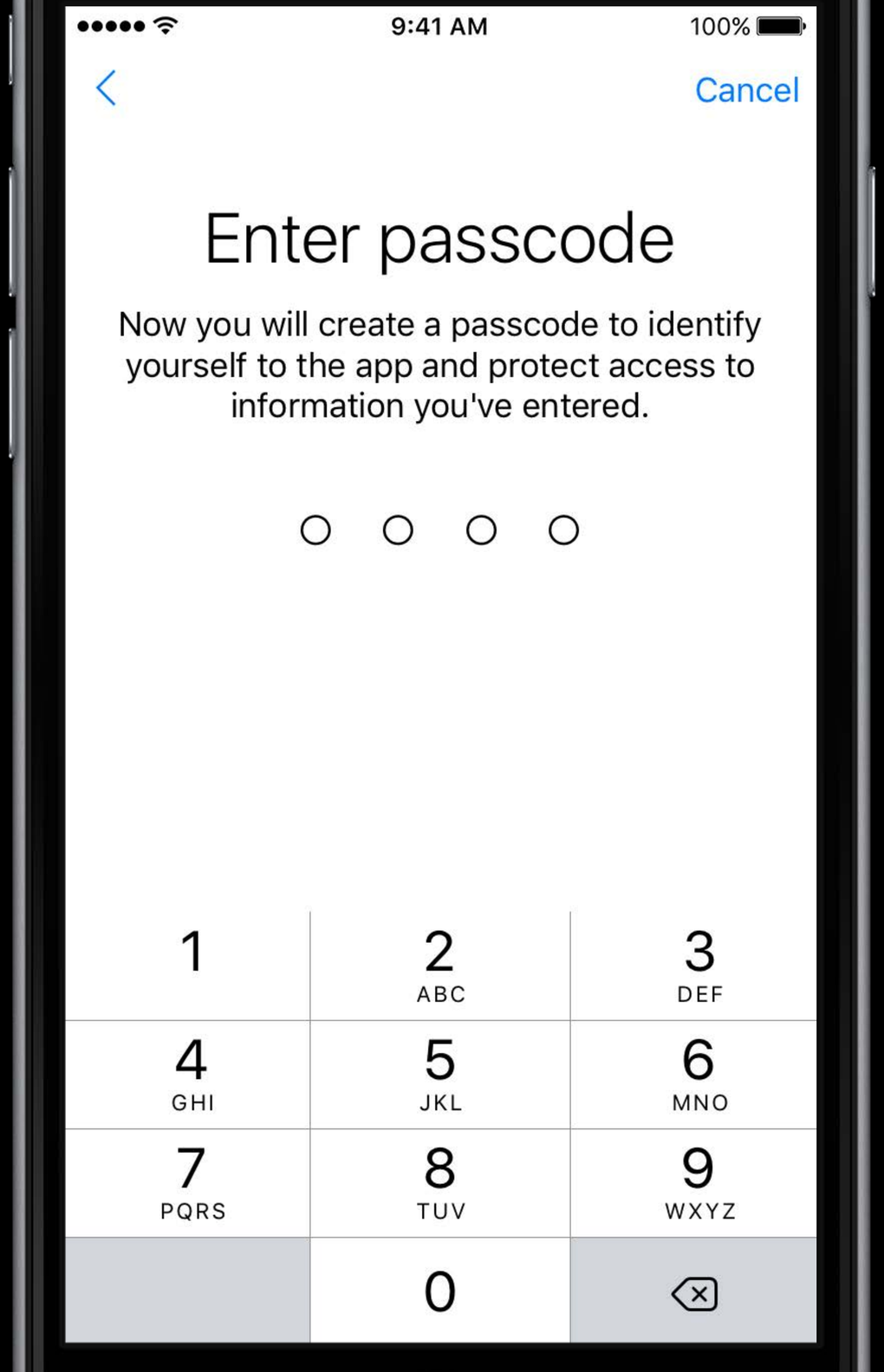
Passcode

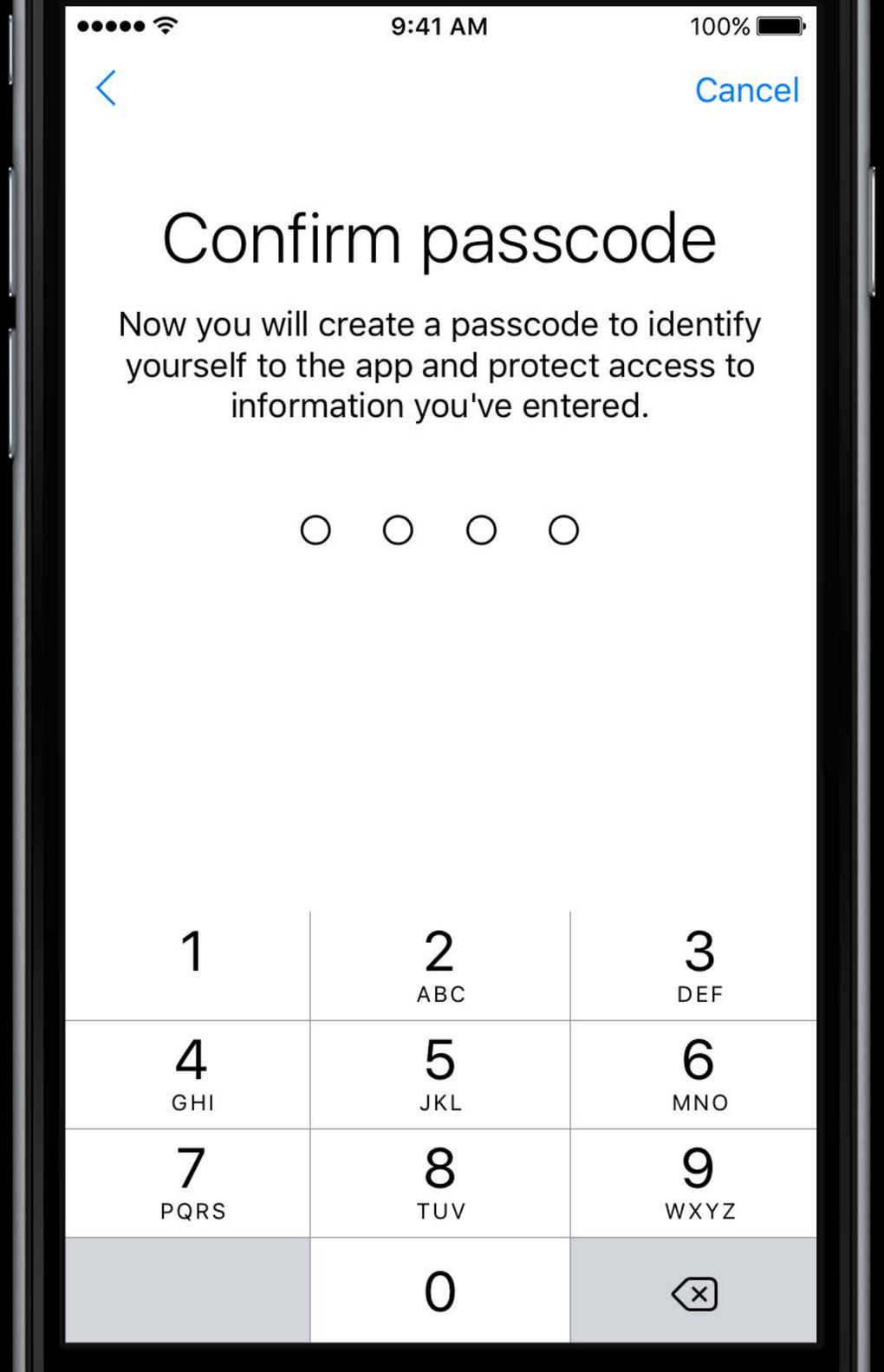
Creation

Authentication

App Access


Account
Passcode
Registration
Creation
Login
Authentication


9:41 AM
100%
Cancel
Enter passcode
Now you will create a passcode to identify yourself to the app and protect access to information you've entered.
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0


9:41 AM
100%
Cancel
Confirm passcode
Now you will create a passcode to identify yourself to the app and protect access to information you've entered.
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0

9:41 AM 100%

Cancel

Passcode saved

Touch ID for “WWDC2016”

Please authenticate with Touch ID

Cancel

App Access

Passcode creation

NEW

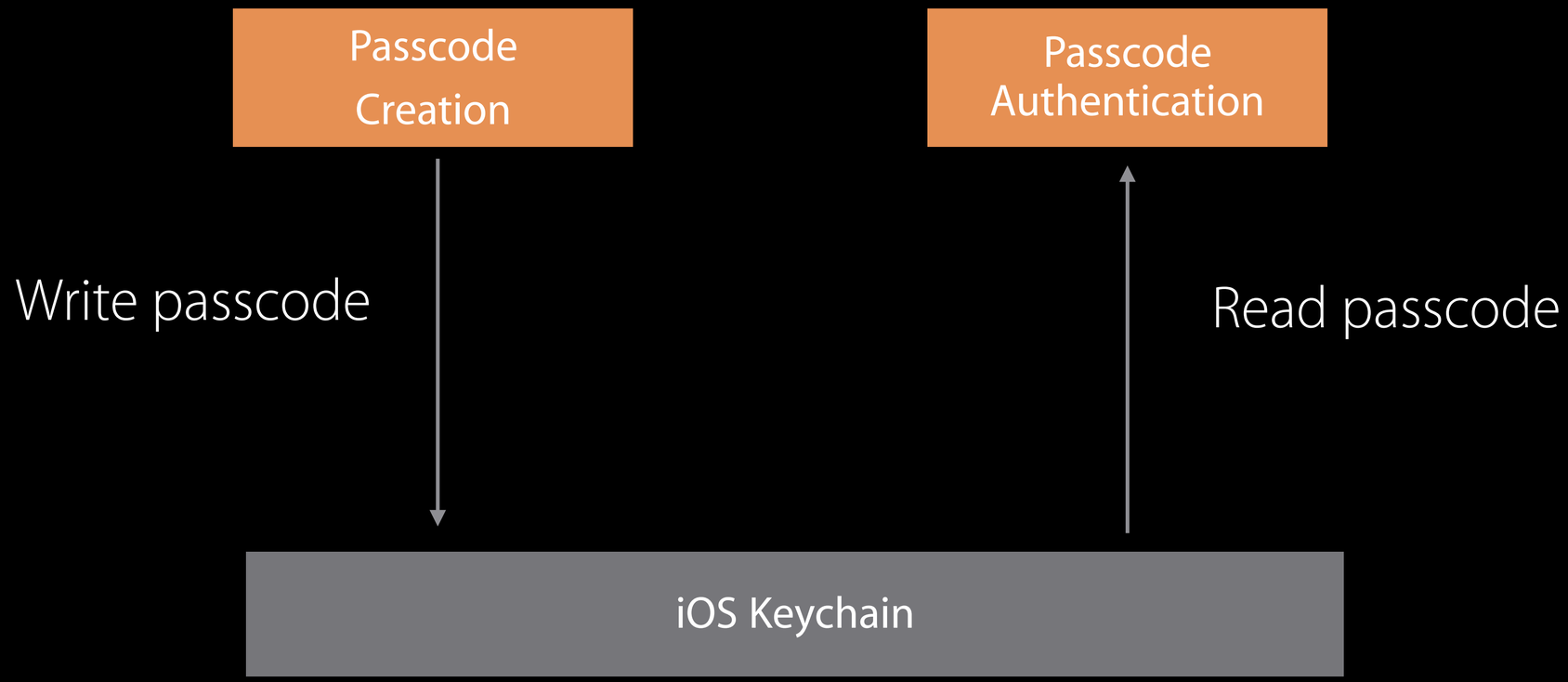
```
let passcodeStep = ORKPasscodeStep(identifier:"passcode_creation")

// Explain the purpose of setting up a passcode
passcodeStep.text = “Now you will create a passcode to identify yourself to the app and
protect access to the information you’ve entered.”
```

App Access

Store and retrieve passcode

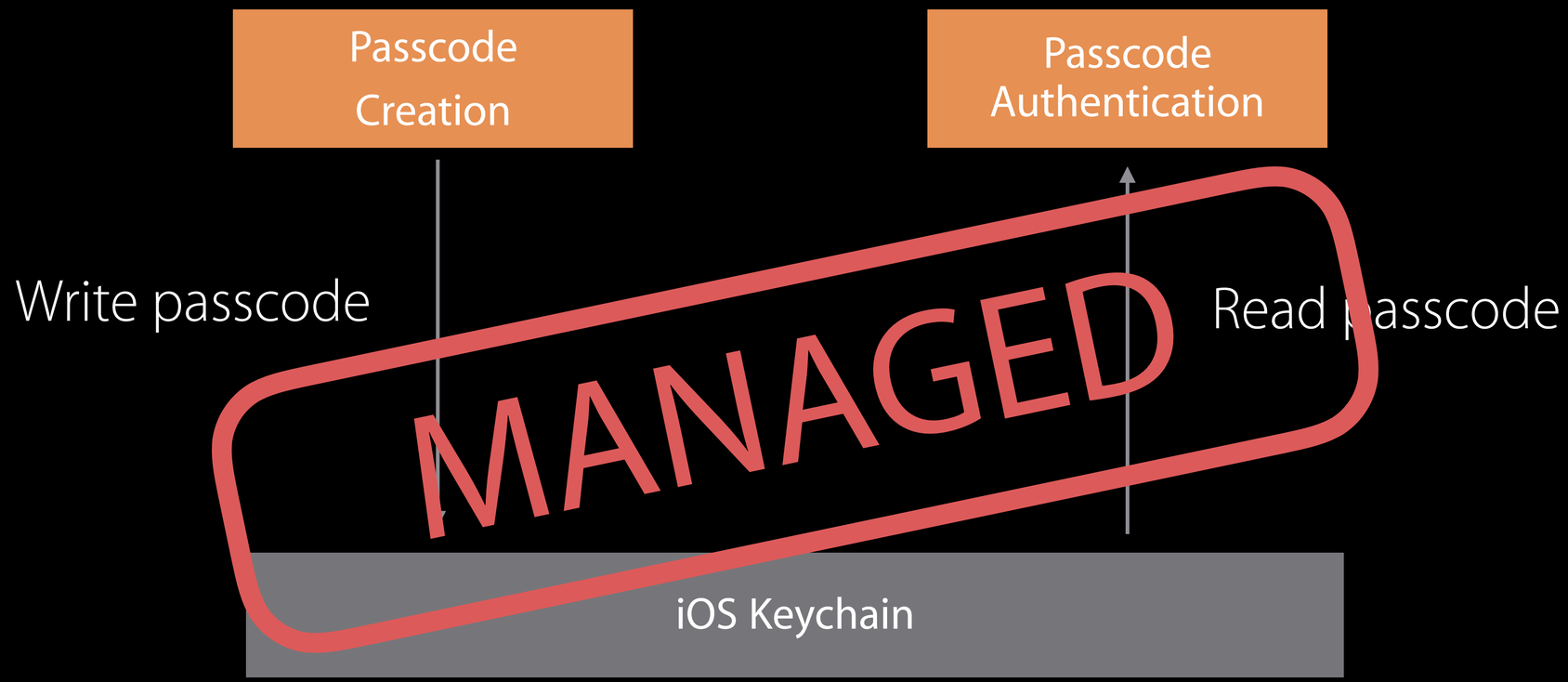
NEW



App Access

Store and retrieve passcode

NEW



9:41 AM 100%

Cancel

Enter passcode

Enter the 4 digits passcode

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
	0	⌫

App Access

Passcode authentication

NEW

```
let passcodeViewController =
ORKPasscodeViewController.passcodeAuthenticationViewController(withText: "Enter the 4
digits passcode", delegate: self)
```

App Access

Passcode authentication delegate

NEW

```
// Success
func passcodeViewControllerDidFinish(withSuccess viewController: UIViewController) {
    viewController.dismissViewControllerAnimated(true) {
        // Present the app functions
    }
}
// Fail
func passcodeViewControllerDidFailAuthentication(_ viewController: UIViewController) {
    // Show alert
}
// Cancel
func passcodeViewControllerDidCancel(_ viewController: UIViewController) {
    viewController.dismissViewControllerAnimated(true) {}
}
```

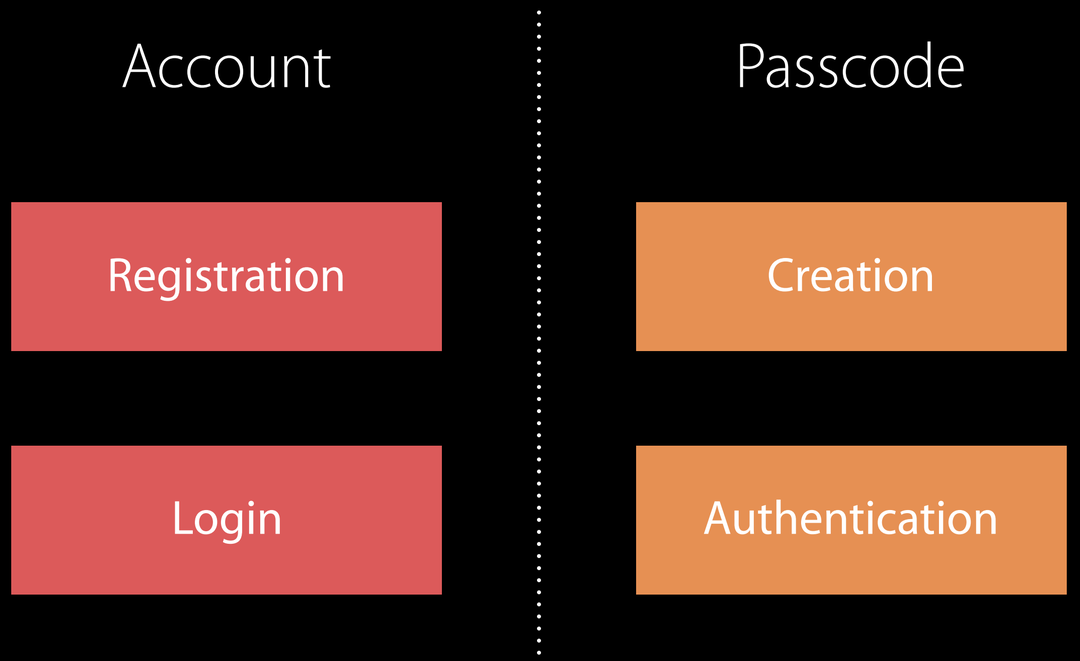
App Access

Passcode authentication delegate

NEW

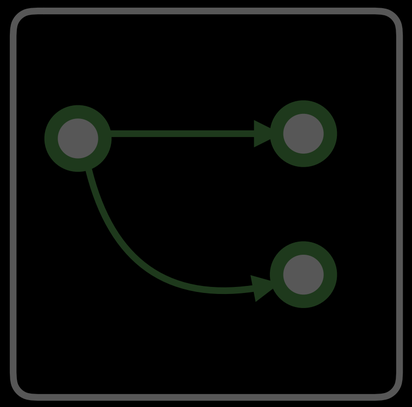
```
// Success
func passcodeViewControllerDidFinish(withSuccess viewController: UIViewController) {
    viewController.dismissViewControllerAnimated(true) {
        // Present the app functions
    }
}
// Fail
func passcodeViewControllerDidFailAuthentication(_ viewController: UIViewController) {
    // Show alert
}
// Cancel
func passcodeViewControllerDidCancel(_ viewController: UIViewController) {
    viewController.dismissViewControllerAnimated(true) {}
}
```

App Access


Account
Passcode
Registration
Creation
Login
Authentication

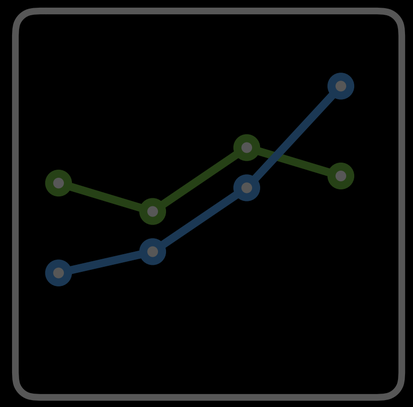
New Features

Branching Task



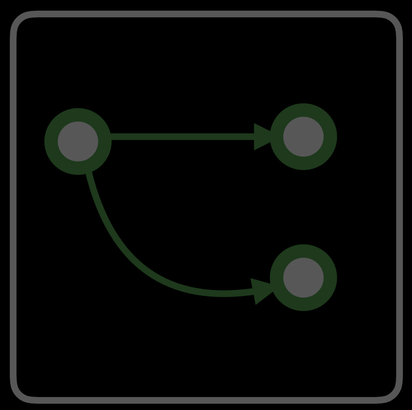
App Access

Charts



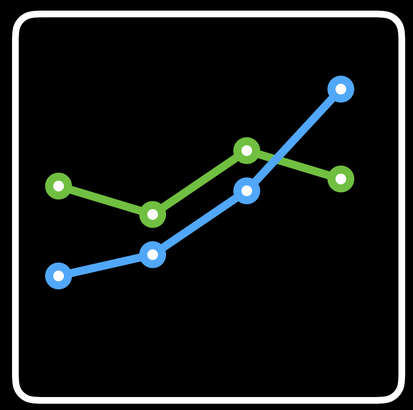
New Features

Branching Task



App Access

Charts



Charts

Display data in charts and graphs

Pie

Line

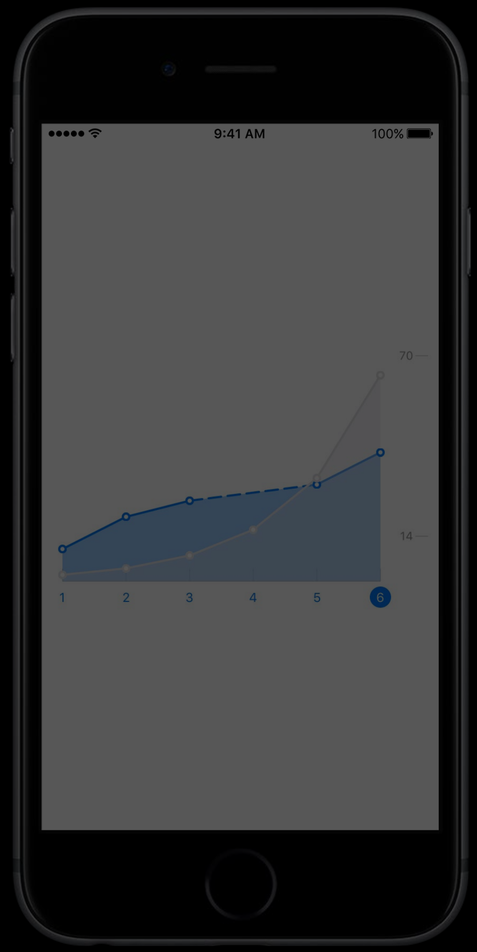
Discrete

Charts

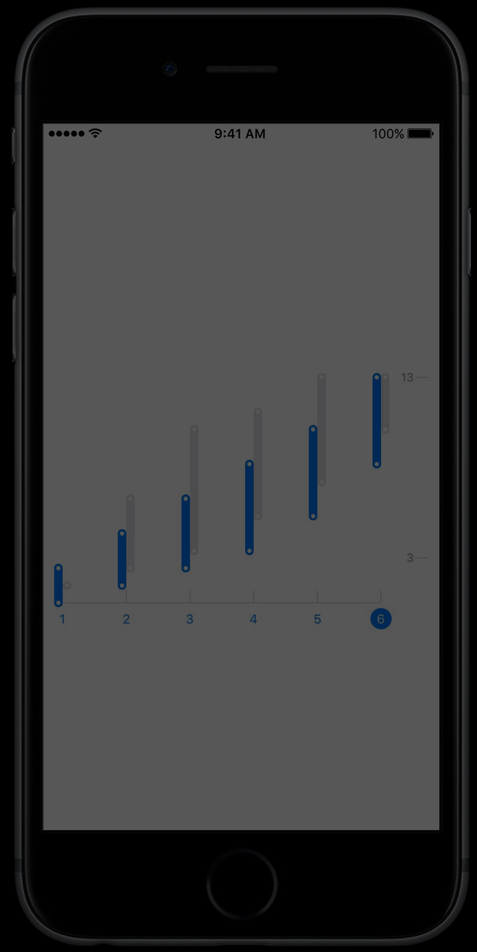
Display data in charts and graphs

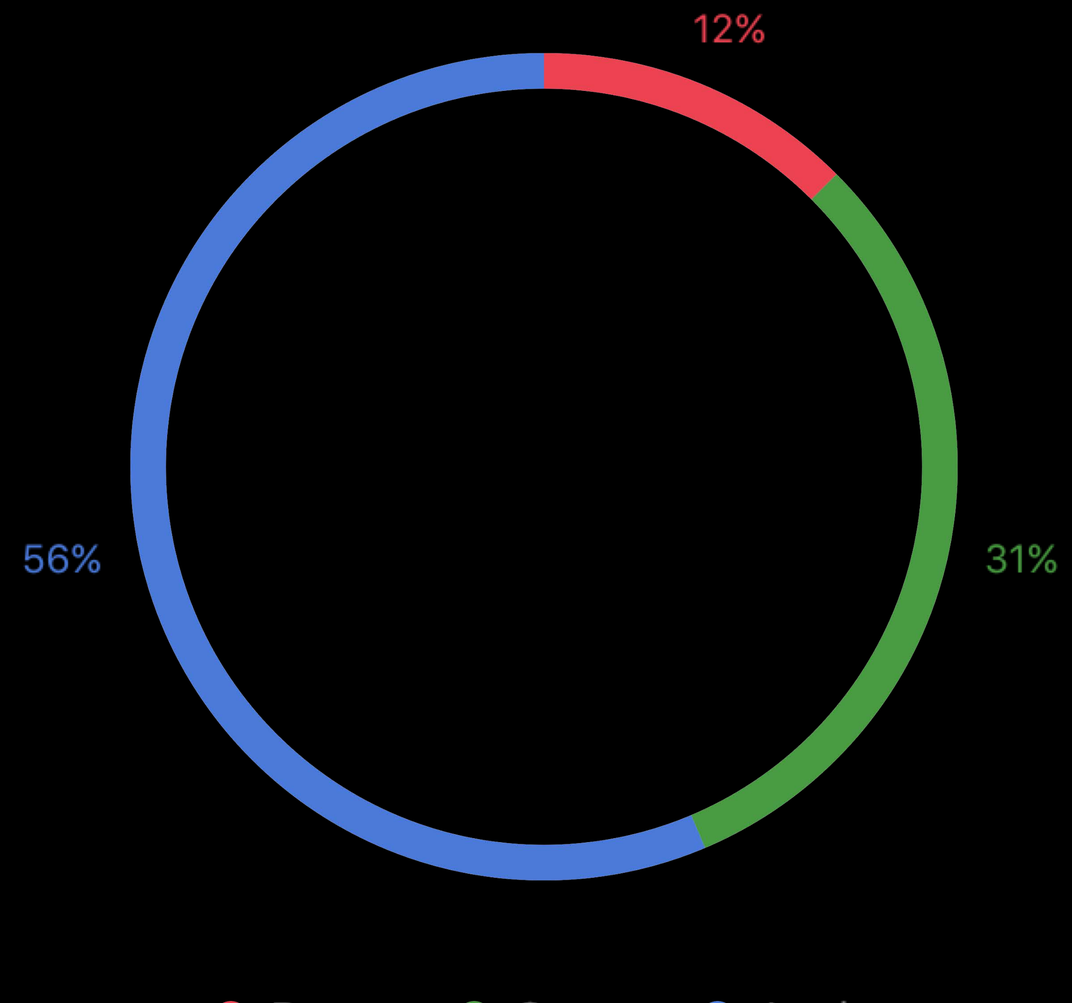
Pie

Line



Discrete




12%
56%
31%
Banana
Orange
Apple

```
// View
let pieChartView = ORKPieChartView(frame: CGRect(x:0, y:0, width:300, height:300))
let pieChartDataSource = PieChartDataSource()
pieChartView.dataSource = pieChartDataSource

// Data Source
class PieChartDataSource: NSObject, ORKPieChartViewDataSource {
    struct Segment {
        let title: String
        let value: Float
        let color: UIColor
    }
    // Properties
    let segments = [
        Segment(title: "Apple", value: 10.0, color: blueColor),
        Segment(title: "Orange", value: 10.0, color: greenColor),
        Segment(title: "Banana", value: 10.0, color: redColor),
    ]
```

```
// View
let pieChartView = ORKPieChartView(frame: CGRect(x:0, y:0, width:300, height:300))
let pieChartDataSource = PieChartDataSource()
pieChartView.dataSource = pieChartDataSource

// Data Source
class PieChartDataSource: NSObject, ORKPieChartViewDataSource {
    struct Segment {
        let title: String
        let value: Float
        let color: UIColor
    }
    // Properties
    let segments = [
        Segment(title: "Apple", value: 10.0, color: blueColor),
        Segment(title: "Orange", value: 10.0, color: greenColor),
        Segment(title: "Banana", value: 10.0, color: redColor),
    ]
```

```swift
// View
let pieChartView = ORKPieChartView(frame: CGRect(x:0, y:0, width:300, height:300))
let pieChartDataSource = PieChartDataSource()
pieChartView.dataSource = pieChartDataSource

// Data Source
class PieChartDataSource: NSObject, ORKPieChartViewDataSource {
    struct Segment {
        let title: String
        let value: Float
        let color: UIColor
    }
    // Properties
    let segments = [
        Segment(title: "Apple", value: 10.0, color: blueColor),
        Segment(title: "Orange", value: 10.0, color: greenColor),
        Segment(title: "Banana", value: 10.0, color: redColor),
    ]
```

```
// Number of Segments
func numberOfSegments(in pieChartView: ORKPieChartView) -> Int {
    return segments.count
}
// Value of each Segment
func pieChartView(_ pieChartView: ORKPieChartView, valueForSegmentAt index: Int) ->
CGFloat {
    return CGFloat(segments[index].value)
}
// Color for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, colorForSegmentAt index: Int) ->
UIColor {
    return segments[index].color
}
// Title for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, titleForSegmentAt index: Int) ->
String {
    return segments[index].title
}
```

```swift
// Number of Segments
func numberOfSegments(in pieChartView: ORKPieChartView) -> Int {
    return segments.count
}
// Value of each Segment
func pieChartView(_ pieChartView: ORKPieChartView, valueForSegmentAt index: Int) ->
CGFloat {
    return CGFloat(segments[index].value)
}
// Color for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, colorForSegmentAt index: Int) ->
UIColor {
    return segments[index].color
}
// Title for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, titleForSegmentAt index: Int) ->
String {
    return segments[index].title
}
```

```swift
// Number of Segments
func numberOfSegments(in pieChartView: ORKPieChartView) -> Int {
    return segments.count
}
// Value of each Segment
func pieChartView(_ pieChartView: ORKPieChartView, valueForSegmentAt index: Int) ->
CGFloat {
    return CGFloat(segments[index].value)
}
// Color for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, colorForSegmentAt index: Int) ->
UIColor {
    return segments[index].color
}
// Title for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, titleForSegmentAt index: Int) ->
String {
    return segments[index].title
}
```

```
// Number of Segments
func numberOfSegments(in pieChartView: ORKPieChartView) -> Int {
    return segments.count
}
// Value of each Segment
func pieChartView(_ pieChartView: ORKPieChartView, valueForSegmentAt index: Int) ->
CGFloat {
    return CGFloat(segments[index].value)
}
// Color for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, colorForSegmentAt index: Int) ->
UIColor {
    return segments[index].color
}
// Title for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, titleForSegmentAt index: Int) ->
String {
    return segments[index].title
}
```

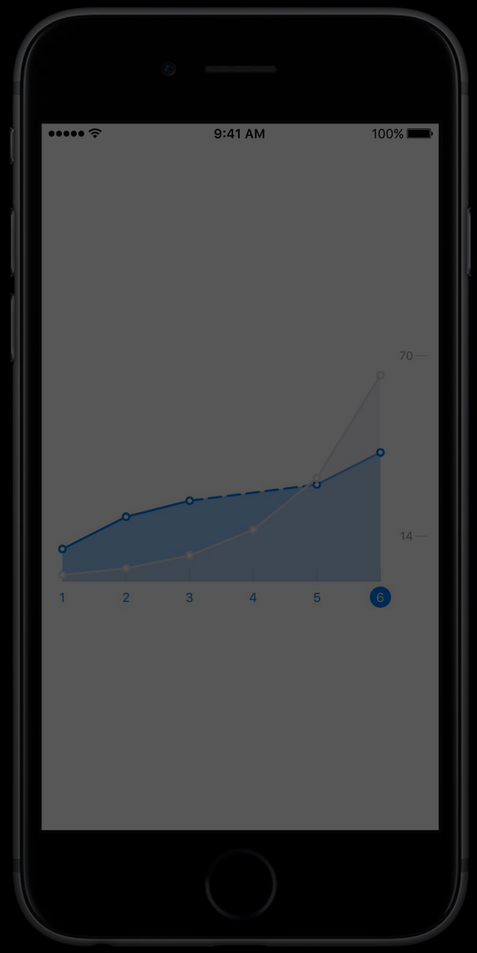
```
// Number of Segments
func numberOfSegments(in pieChartView: ORKPieChartView) -> Int {
    return segments.count
}
// Value of each Segment
func pieChartView(_ pieChartView: ORKPieChartView, valueForSegmentAt index: Int) ->
CGFloat {
    return CGFloat(segments[index].value)
}
// Color for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, colorForSegmentAt index: Int) ->
UIColor {
    return segments[index].color
}
// Title for each Segment
func pieChartView(_ pieChartView: ORKPieChartView, titleForSegmentAt index: Int) ->
String {
    return segments[index].title
}
```

Charts

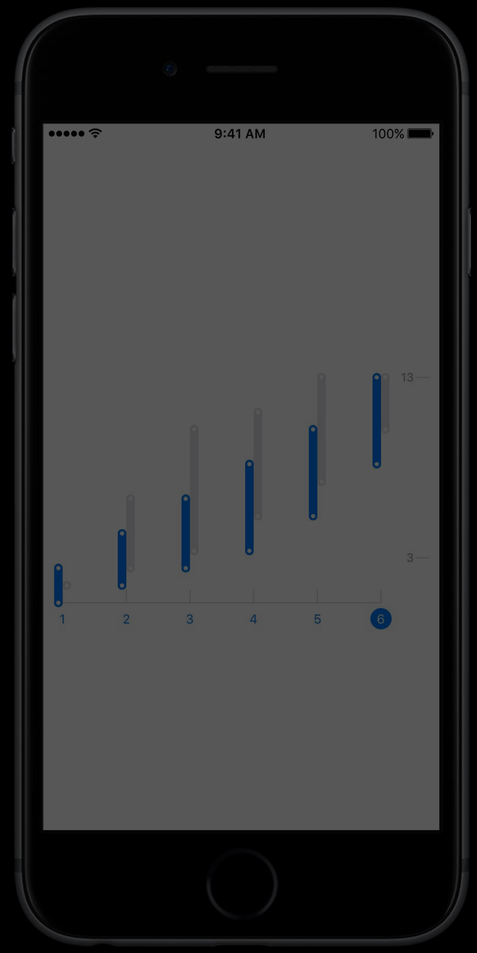
Display data in charts and graphs

Pie

Line



Discrete



Charts

Display data in charts and graphs

Pie

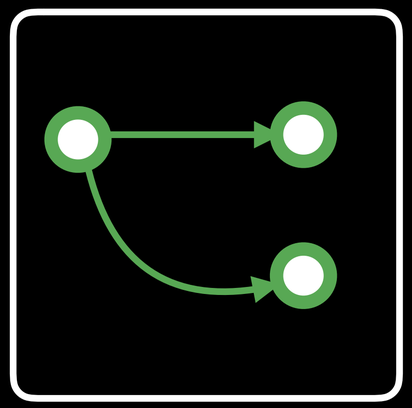
Line

Discrete

https://github.com/ResearchKit/ResearchKit

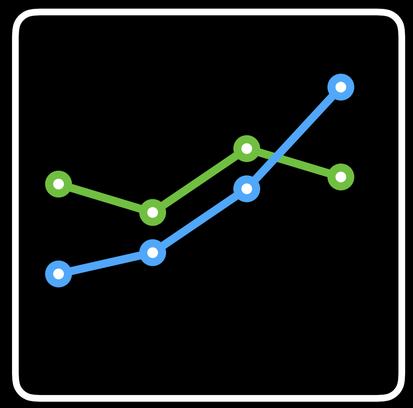
New Features

Branching Task



App Access

Charts



Contributing to ResearchKit

ResearchKit Community

https://github.com/researchkit/researchkit

Pick or open an issue

ResearchKit Community

https://github.com/researchkit/researchkit

Pick or open an issue

Submit a pull request

ResearchKit Community

https://github.com/researchkit/researchkit

Pick or open an issue

Submit a pull request

Work with review team

ResearchKit Community

https://github.com/researchkit/researchkit

Pick or open an issue

Submit a pull request

Work with review team

Changes get merged

ResearchKit Community

https://github.com/researchkit/researchkit

Pick or open an issue

Submit a pull request

Work with review team

Changes get merged

Branch for Convergence

ResearchKit Community

https://github.com/researchkit/researchkit

Pick or open an issue

Submit a pull request

Work with review team

Changes get merged

Branch for Convergence

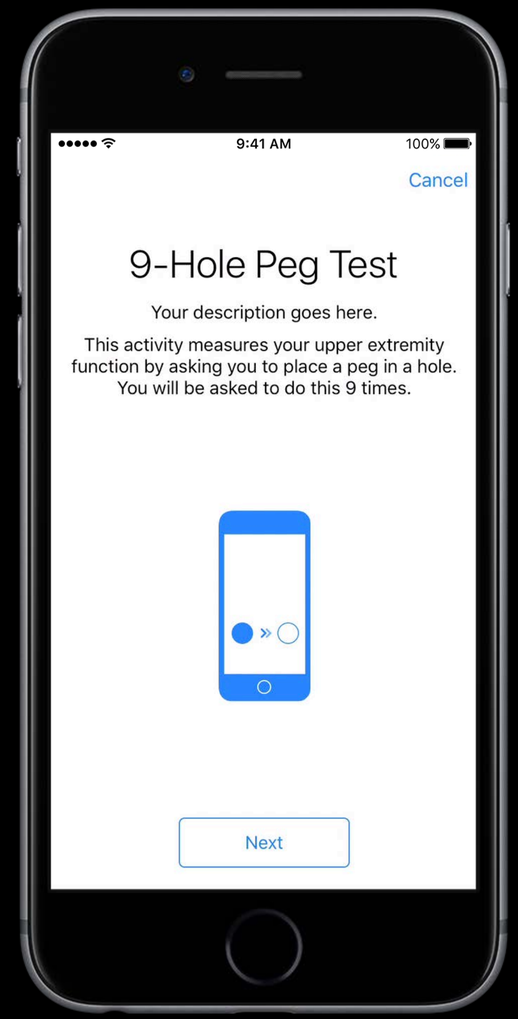
New stable version tagged

Contributions Are Important!

Active tasks

Contributions Are Important!

Active tasks



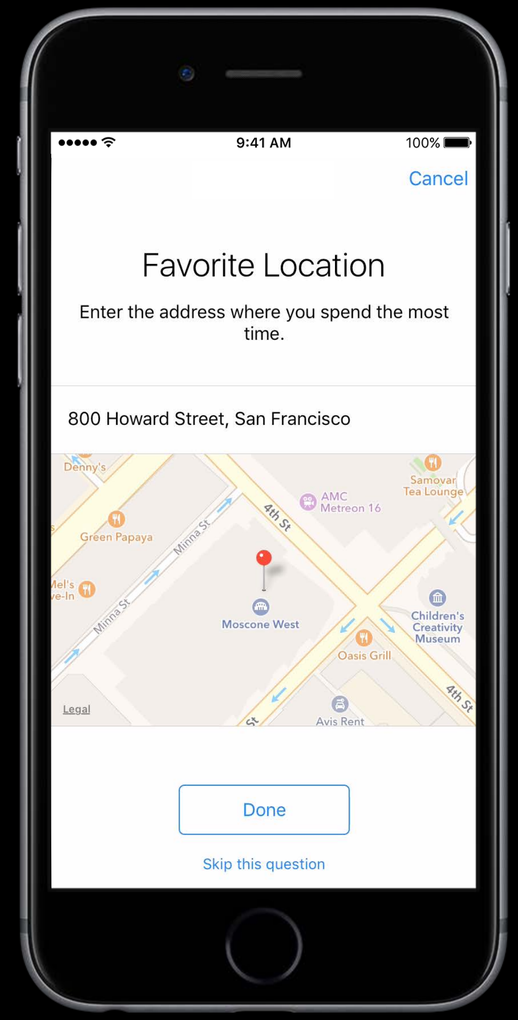
Contributions Are Important!

Active tasks

Contributions Are Important!

Active tasks

Answer formats

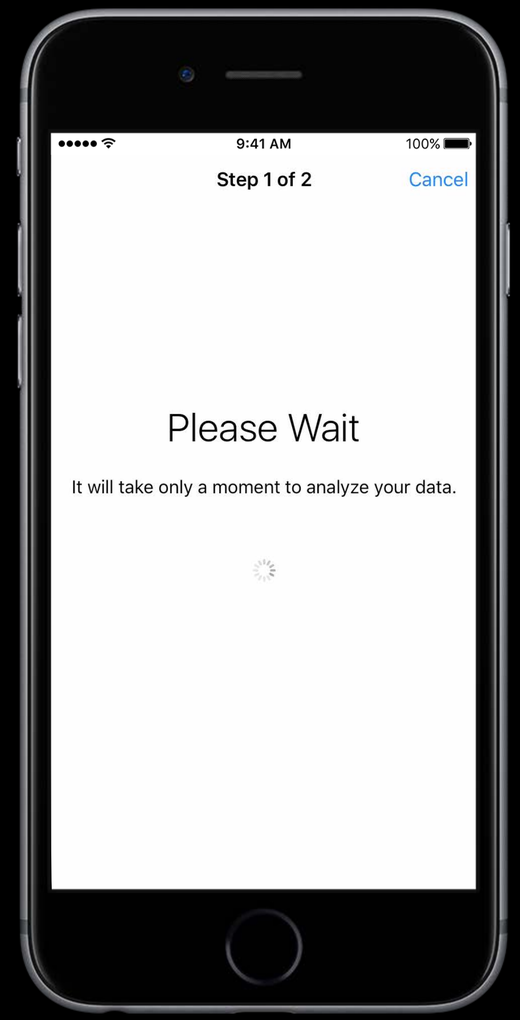


Contributions Are Important!

Active tasks

Answer formats

Steps



Contributions Are Important!

Active tasks

Answer formats

Steps

Modules

Contributions Are Important!

Active tasks

Answer formats

Steps

Modules

Contributions Are Important!

Active tasks

Answer formats

Steps

Modules

Data services

ResearchKit Tutorials

http://www.researchkit.org

How to Set up a ResearchKit Project

Advanced ResearchKit Project Setup

How to Create a ResearchKit Active Task

Learning ResearchKit

Accessing Heart Rate Data for Your ResearchKit Study

Research Apps

A research app lets iOS users participate in research studies from the convenience of their iOS devices. The predesigned screens and transitions available in Apple's open source ResearchKit project make it easy to create a beautiful research app that's customized for your study and enjoyable for people to use. To learn how to use ResearchKit to develop a research app for your study, see researchkit.org.

> IMPORTANT
>
> These guidelines are for informational purposes only and do not constitute legal advice. You should contact an attorney to obtain advice with respect to the development of a research app and any applicable laws.

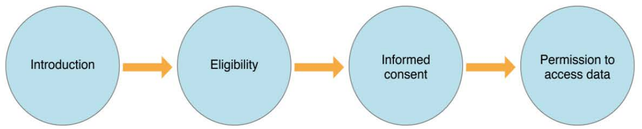
Typically, a research app groups customized ResearchKit screens and app-specific screens into sections that fit into three primary experiences:

- Onboarding
- Study-specific investigation
- Management items

Follow the guidelines for the sections that comprise each of these experiences so that you can design a research app that helps participants feel comfortable and stay engaged.

Onboarding

The onboarding experience consists of a series of sections that introduce the study to potential participants and allow you to get their consent. Participants don't typically revisit the onboarding sections after completing them. The onboarding experience includes the following sections:



Call to Action

https://github.com/researchkit/researchkit

Download the ResearchKit framework

Call to Action

https://github.com/researchkit/researchkit

Download the ResearchKit framework

Build and run the ORKSample app

Call to Action

https://github.com/researchkit/researchkit

Download the ResearchKit framework

Build and run the ORKSample app

Build and run the ORKCatalog app

Call to Action

https://github.com/researchkit/researchkit

Download the ResearchKit framework

Build and run the ORKSample app

Build and run the ORKCatalog app

Make a contribution

Call to Action

https://github.com/researchkit/researchkit

Download the ResearchKit framework

Build and run the ORKSample app

Build and run the ORKCatalog app

Make a contribution

ResearchKit is an open project—it will become what YOU make it

More Information

https://developer.apple.com/wwdc16/234

Related Sessions

Getting Started with CareKit	Pacific Heights	Friday 3:00PM

Labs

ResearchKit and CareKit Lab	Fort Mason	Friday 10:30AM
ResearchKit and CareKit Lab	Fort Mason	Friday 3:30PM

WWDC16