ResearchKit and CareKit Reimagined

Srinath Tupil Muralidharan, Software Engineer
Community
Community
ResearchKit
NEW APPS
ResearchKit & CareKit

MONDAY, JUNE 3
TODAY
MONDAY, JUNE 3
TODAY

NEW APPS
ResearchKit & CareKit

[App icons]

WebMD
FDA
NOVARTIS

[Other app icons]
TODAY

MONDAY, JUNE 3

NEW APPS

ResearchKit & CareKit

FocalView
ResearchKit & CareKit
TODAY

MONDAY, JUNE 3

NEW APPS

ResearchKit & CareKit

MyStudies
MONDAY, JUNE 3

TODAY

NEW APPS

ResearchKit & CareKit
USING MOBILE TECHNOLOGY TO ENGAGE SEXUAL AND GENDER MINORITIES IN CLINICAL RESEARCH
PRIDE STUDY
THE HEALTHY PREGNANCY RESEARCH PROGRAM: TRANSFORMING PREGNANCY RESEARCH THROUGH A RESEARCHKIT APP

PRIDE STUDY

HEALTHY PREGNANCY
PERSONAL TOUCH: IMPROVING TOUCHSCREEN USABILITY BY PERSONALIZING ACCESSIBILITY SETTINGS BASED ON INDIVIDUAL USER’S TOUCHSCREEN INTERACTION

PRIDE STUDY

HEALTHY PREGNANCY

PERSONAL TOUCH
Investigator Support Pilot

Apple Watch
Limited Grant Program
Investigator Support Pilot

Simple Submission Process

COMING SOON
Redesigned Website
ResearchKit
UI Updates

Active Tasks
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- [ ] Insomnia
- [ ] Obstructive Sleep Apnea (OSA)
- [ ] Restless Leg Syndrome (RLS)
- [ ] Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- [ ] Coffee required
- [ ] Tea required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- Insomnia
- Obstructive Sleep Apnea (OSA)
- Restless Leg Syndrome (RLS)
- Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee  required
- Tea  required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- Insomnia
- Obstructive Sleep Apnea (OSA)
- Restless Leg Syndrome (RLS)
- Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee: required
- Tea: required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- Insomnia
- Obstructive Sleep Apnea (OSA)
- Restless Leg Syndrome (RLS)
- Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

Coffee  required
Tea  required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- Insomnia
- Obstructive Sleep Apnea (OSA)
- Restless Leg Syndrome (RLS)
- Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

Coffee: required
Tea: required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- Insomnia
- Obstructive Sleep Apnea (OSA)
- Restless Leg Syndrome (RLS)
- Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee
- Tea

Learn More Button
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- [✓] Insomnia
- [ ] Obstructive Sleep Apnea (OSA)
- [ ] Restless Leg Syndrome (RLS)
- [ ] Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee: required
- Tea: required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- [x] Insomnia
- [ ] Obstructive Sleep Apnea (OSA)
- [ ] Restless Leg Syndrome (RLS)
- [ ] Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee: required
- Tea: required
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- **Insomnia**
- Obstructive Sleep Apnea (OSA)
- Restless Leg Syndrome (RLS)
- Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee  **required**
- Tea  **required**
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

Insomnia:
• This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

Obstructive Sleep Apnea (OSA):
• Causes breathing to stop and start during sleep as the result of the throat muscles not relaxing during sleep.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

**Insomnia:**
This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

**Obstructive Sleep Apnea (OSA):**
Causes breathing to stop and start during sleep as a result of the throat muscles relaxing.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

Insomnia:
- This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

Obstructive Sleep Apnea (OSA):
- Causes breathing to stop and start during sleep as the result of the throat muscles collapsing or relaxing.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

Insomnia:
- This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

Obstructive Sleep Apnea (OSA):
- Causes breathing to stop and start during sleep as the result of the throat muscles relaxing and blocking the passageway to the lungs.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

Insomnia:
- This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

Obstructive Sleep Apnea (OSA):
- Causes breathing to stop and start during sleep as the result of the throat muscles relaxing and blocking the airway.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

Insomnia:
- This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

Obstructive Sleep Apnea (OSA):
- Causes breathing to stop and start during sleep as the result of the throat muscles relaxing and blocking the airway at irregular intervals.

Restless Leg Syndrome (RLS):
- Neurological disorder that patients often describe as a tingling sensation in the legs, often relieved by moving or stretching.

Narcolepsy:
- Excessive daytime sleepiness despite adequate evening sleep.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

- **Insomnia:**
  - This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

- **Obstructive Sleep Apnea (OSA):**
  - Causes breathing to stop and start during sleep as the result of the throat muscles relaxing and blocking the airway at irregular intervals.

- **Restless Leg Syndrome (RLS):**
  - Neurological disorder that patients often describe as a tingling sensation in the legs, often relieved by moving or stretching.

- **Narcolepsy:**
  - Excessive daytime sleepiness despite adequate evening sleep.
import ResearchKit

let learnMoreStep = ORKLearnMoreStep(identifier: "learnMoreStep")
import ResearchKit

let learnMoreStep = ORKLearnMoreStep(identifier: "learnMoreStep")
learnMoreStep.image = UIImage(named: "Sleep")
import ResearchKit

let learnMoreStep = ORKLearnMoreStep(identifier: "learnMoreStep")
learnMoreStep.image = UIImage(named: "Sleep")
learnMoreStep.title = "Why Your Sleep Matters"
import ResearchKit

let learnMoreStep = ORKLearnMoreStep(identifier: "learnMoreStep")
learnMoreStep.image = UIImage(named: "Sleep")
learnMoreStep.title = "Why Your Sleep Matters"
learnMoreStep.text = "Sleep is an important measure of your overall health and well-being."
import ResearchKit

let learnMoreStep = ORKLearnMoreStep(identifier: "learnMoreStep")
learnMoreStep.image = UIImage(named: "Sleep")
learnMoreStep.title = "Why Your Sleep Matters"
learnMoreStep.text = "Sleep is an important measure of your overall health and well-being."

let bodyItem1 = ORKBodyItem(text: "Narcolepsy:",
    detailText: "Excessive daytime sleepiness.",
    style: .bulletPoint)
import ResearchKit

let learnMoreStep = ORKLearnMoreStep(identifier: "learnMoreStep")
learnMoreStep.image = UIImage(named: "Sleep")
learnMoreStep.title = "Why Your Sleep Matters"
learnMoreStep.text = "Sleep is an important measure of your overall health and well-being."

let bodyItem1 = ORKBodyItem(text: "Narcolepsy:",
    detailText: "Excessive daytime sleepiness.",
    style: .bulletPoint)

learnMoreStep.bodyItems = [bodyItem1]
// create a learnMoreItem by passing in a learnMoreStep
let learnMoreItem = ORKLearnMoreItem(step: learnMoreStep)
// create a learnMoreItem by passing in a learnMoreStep
let learnMoreItem = ORKLearnMoreItem(step: learnMoreStep)

// group one or more answer formats under a section
let section = ORKFormItem(sectionTitle: "Do you have any of the following sleep disorders?", detailText: "For more detailed instructions tap the info button.", learnMoreItem: learnMoreItem, showsProgress: true)
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

**Insomnia:**
- This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

**Obstructive Sleep Apnea (OSA):**
- Causes breathing to stop and start during sleep as the result of the throat muscles relaxing and blocking the airway at irregular intervals.

**Restless Leg Syndrome (RLS):**
- Neurological disorder that patients often describe as a tingling sensation in the legs, often relieved by moving or stretching.

**Narcolepsy:**
- Excessive daytime sleepiness despite adequate evening sleep.
Why Your Sleep Matters

Sleep is an important measure of your overall health and well-being. It is important to understand if you have been diagnosed with or experience any of the following conditions that could have other impacts to your overall health.

Sleep Disorder Descriptions:

Insomnia:
This condition refers to the inability to initiate sleep, waking up frequently during the night or waking up too early each morning.

Obstructive Sleep Apnea (OSA):
Causes breathing to stop and start during sleep as the result of the throat muscles relaxing and blocking the airway at irregular intervals.

Restless Leg Syndrome (RLS):
Neurological disorder that patients often describe as a tingling sensation in the legs, often relieved by moving or stretching.

Narcolepsy:
Excessive daytime sleepiness despite adequate evening sleep.
Let's Talk About Your Sleep

QUESTION 1 OF 3
Do you have any of the following sleep disorders?
For more detailed descriptions tap the info button.

- [ ] Insomnia
- [ ] Obstructive Sleep Apnea (OSA)
- [x] Restless Leg Syndrome (RLS)
- [ ] Narcolepsy

QUESTION 2 OF 3
How many caffeinated drinks do you consume each day?
Type the number of drinks you consume on a typical day for each of the following caffeinated beverages.

- Coffee  required
- Tea  required
UI Updates

Active Tasks
Vision
Vision

Hearing
Vision

Hearing

Speech
Visual Acuity
Landolt C & Tumbling E
Visual Acuity
Landolt C & Tumbling E

Contrast Sensitivity
Landolt C & Gabor Patch
Contrast Sensitivity

Visual Acuity
Landolt C & Tumbling E

Contrast Sensitivity
Landolt C & Gabor Patch
Visual Acuity — Landolt C
Novartis FocalView Contribution

Landolt C Acuity Task
This task collects data to better understand your visual acuity.

During this task, a Landolt C will appear on the screen in various orientations and sizes for a few seconds.

- Learn more about Landolt C.

Each time this happens, use your finger to align the dial with the opening gap in the C and tap the 'Next' button.

Get Started
Cancel
Visual Acuity — Landolt C
Novartis FocalView Contribution

Landolt C Acuity Task
This task collects data to better understand your visual acuity.

During this task, a Landolt C will appear on the screen in various orientations and sizes for a few seconds.
Learn more about Landolt C.

Each time this happens, use your finger to align the dial with the opening gap in the C and tap the 'Next' button.

Get Started
Cancel
Visual Acuity — Landolt C
Novartis FocalView Contribution

Landolt C Stimulus
• Fixed contrast
• Decreasing size
• Random orientations
Visual Acuity — Landolt C
Novartis FocalView Contribution

Landolt C Stimulus
• Fixed contrast
• Decreasing size
• Random orientations

Result
• VAR (visual acuity rating)
import ResearchKit

// create a landoltC acuity step
let step = ORKLandoltCStep(identifier: "LandoltCAcuity")
import ResearchKit

// create a landoltC acuity step
let step = ORKLandoltCStep(identifier: "LandoltCAcuity")
step.testType = .acuity
import ResearchKit

// create a landoltC acuity step
let step = ORKLandoltCStep(identifier: "LandoltCAcuity")
step.testType = .acuity

// create an active task by passing in the step
let task = ORKOrderedTask(identifier: "LandoltCAcuityTask", steps: [step])
import ResearchKit

// create a landoltC acuity step
let step = ORKLandoltCStep(identifier: "LandoltCAcuity")
step.testType = .acuity

// create an active task by passing in the step
let task = ORKOrderedTask(identifier: "LandoltCAcuityTask", steps: [step])

let taskVC = ORKTaskViewController(task: task, taskRun: UUID())
import ResearchKit

// create a landoltC acuity step
let step = ORKLandoltCStep(identifier: "LandoltCAcuity")
step.testType = .acuity

// create an active task by passing in the step
let task = ORKOrderedTask(identifier: "LandoltCAcuityTask", steps: [step])

let taskVC = ORKTaskViewController(task: task, taskRun: UUID())
present(taskVC, animated: true, completion: nil)
Visual Acuity — Tumbling E

COMING SOON
Visual Acuity — Tumbling E

COMING SOON

Tumbling E
This task collects data to better understand your visual acuity.

- Hold the device 15-17 inches from your face and focus your attention on the screen.
- During this task, an E will appear on the screen in various orientations and sizes.

Each time you see the E, use your finger to swipe into the arms of the Snellen E (as shown in the image above).

Get Started

Cancel
Visual Acuity — Tumbling E

Tumbling E Stimulus
• Leverages TrueDepth camera
• Fixed contrast
• Decreasing size
• Random orientations

COMING SOON
Visual Acuity — Tumbling E

Tumbling E Stimulus
• Leverages TrueDepth camera
• Fixed contrast
• Decreasing size
• Random orientations

Result
• logMAR (minimum angle of resolution)
import ResearchKit

// create a tumblingE acuity step
let step = ORKTumblingEStep(identifier: "TumblingEAcuity")
import ResearchKit

// create a tumblingE acuity step
let step = ORKTumblingEStep(identifier: "TumblingEAcuity")

// configure min and max viewing distance as determined by the TrueDepth camera
step.minViewingDistance = Measurement(value: 38.0, unit: UnitLength.centimeters)
step.maxViewingDistance = Measurement(value: 42.0, unit: UnitLength.centimeters)
import ResearchKit

// create a tumblingE acuity step
let step = ORKTumblingEStep(identifier: "TumblingEAcuity")

// configure min and max viewing distance as determined by the TrueDepth camera
step.minViewingDistance = Measurement(value: 38.0, unit: UnitLength.centimeters)
step.maxViewingDistance = Measurement(value: 42.0, unit: UnitLength.centimeters)

// create an active task by passing in the step
let task = ORKOrderedTask(identifier: "TumblingEAcuityTask", steps: [step])
let taskVC = ORKTaskViewController(task: task, taskRun: UUID())
present(taskVC, animated: true, completion: nil)
Visual Acuity
Landolt C & Tumbling E

Contrast Sensitivity
Landolt C & Gabor Patch
Visual Acuity
Landolt C & Tumbling E

Contrast Sensitivity
Landolt C & Gabor Patch
Contrast Sensitivity — Landolt C
Novartis FocalView Contribution

Landolt C Contrast Sensitivity
This task collects data to better understand your visual contrast sensitivity.

- During this task, a Landolt C will appear on the screen in various orientations and contrasts for a few seconds. Learn more about Landolt C.
- Each time this happens, use your finger to align the dial with the opening gap in the C and tap the ‘Next’ button.

Get Started
Cancel
Contrast Sensitivity — Landolt C
Novartis FocalView Contribution

Landolt C Contrast Sensitivity

This task collects data to better understand your visual contrast sensitivity.

During this task, a Landolt C will appear on the screen in various orientations and contrasts for a few seconds.

Learn more about Landolt C.
Each time this happens, use your finger to align the dial with the opening gap in the C and tap the 'Next' button.

Get Started
Cancel
Contrast Sensitivity — Landolt C
Novartis FocalView Contribution

Landolt C Stimulus
• Fixed size
• Decreasing contrast
• Random orientations
Contrast Sensitivity — Landolt C
Novartis FocalView Contribution

Landolt C Stimulus
• Fixed size
• Decreasing contrast
• Random orientations

Result
• VAR (Visual Acuity Rating)
import ResearchKit

// create a landoltC peak contrast sensitivity step
let step = ORKLandoltCStep(identifier: "ContrastSensitivity")
step.testType = .contrastSensitivity
import ResearchKit

// create a landoltC peak contrast sensitivity step
let step = ORKLandoltCStep(identifier: "ContrastSensitivity")
step.testType = .contrastSensitivity

// create an active task by passing in the step
let task = ORKOrderedTask(identifier: "ContrastSensitivityTask", steps: [step])
let taskVC = ORKTaskViewController(task: task, taskRun: UUID())

present(taskVC, animated: true, completion: nil)
Contrast Sensitivity Function

Gabor Patch
Contrast Sensitivity Function
Gabor Patch
Contrast Sensitivity Function

Gabor Patch
Contrast Sensitivity Function

COMING SOON

Contrast Sensitivity Function

This task collects data to better understand your visual health.

- Hold the device 15-17 inches from your face and focus your attention on the screen.

During this task, a gabor patch will appear on the screen. Each gabor patch will have a tilt that points to the left or right. Learn more about gabor patches.

Tap the button to indicate which direction the tilt appeared. You must complete this step before the next image appears.

Get Started

Cancel
Contrast Sensitivity Function

This task collects data to better understand your visual health.

• Hold the device 15-17 inches from your face and focus your attention on the screen.

During this task, a gabor patch will appear on the screen. Each gabor patch will have a tilt that points to the left or right. Learn more about gabor patches.

• Tap the button to indicate which direction the tilt appeared. You must complete this step before the next image appears.

Get Started
Contrast Sensitivity Function

Gabor Patch Stimulus
• Leverages TrueDepth camera
• Adaptive algorithm varies spatial frequency
• Random orientation and position
Contrast Sensitivity Function

Gabor Patch Stimulus

• Leverages TrueDepth camera
• Adaptive algorithm varies spatial frequency
• Random orientation and position

Result

• Sensitivity vs Spatial Frequency curve
import ResearchKit

// create a CSF (Contrast Sensitivity Function) step
let step = ORKCSFStep(identifier: "CSF")
import ResearchKit

// create a CSF (Contrast Sensitivity Function) step
let step = ORKCSFStep(identifier: "CSF")

// configure min and max viewing distance as determined by the TrueDepth camera
step.minViewingDistance = Measurement(value: 38.0, unit: UnitLength.centimeters)
step.maxViewingDistance = Measurement(value: 42.0, unit: UnitLength.centimeters)
import ResearchKit

// create a CSF (Contrast Sensitivity Function) step
let step = ORKCSFStep(identifier: "CSF")

// configure min and max viewing distance as determined by the TrueDepth camera
step.minViewingDistance = Measurement(value: 38.0, unit: UnitLength.centimeters)
step.maxViewingDistance = Measurement(value: 42.0, unit: UnitLength.centimeters)

// create an active task by passing in the step
let task = ORKOOrderedTask(identifier: "CSFTask", steps: [step])
let taskVC = ORKTaskViewController(task: task, taskRun: UUID())
present(taskVC, animated: true, completion: nil)
Vision
Hearing
Speech
Hearing Task Updates

Tone Audiometry
This activity identifies hearing threshold levels of an individual in dB HL scale.

Use headphones
- Please use EarPods or AirPods for accurate results

Adjust volume
- Set your device to maximum value for best results

Listen carefully
- A subtle tone will play for a short duration on either the left or right channel

Tap the button
- When you hear a tone, tap the button on the screen

Get Started
Hearing Task Updates

Algorithm
• Updates to calibration data
• Adaptive starting level
Hearing Task Updates

Algorithm
• Updates to calibration data
• Adaptive starting level

Result
• Write to HealthKit

Tone Audiometry
This activity identifies hearing threshold levels of an individual in dB HL scale.

Use headphones
• Please use EarPods or AirPods for accurate results

Adjust volume
• Set your device to maximum value for best results

Listen carefully
• A subtle tone will play for a short duration on either the left or right channel

Tap the button
• When you hear a tone, tap the button on the screen

Get Started
Cancel
Hearing Task Updates

Algorithm
• Updates to calibration data
• Adaptive starting level

Result
• Write to HealthKit
Speech Task Updates

Speech Recognition
This activity records your speech with the microphone at the bottom of the device, and translates it to text.

- **Start Recording**
  Click the button when you are ready to speak

- **Stop Recording**
  Tap the button again once you are done speaking

- **Edit Transcript**
  Edit the generated transcript if necessary

Next
Speech Task Updates

SFTranscription
• Speaking rate
• Average pause duration
Speech Task Updates

SFTranscription
• Speaking rate
• Average pause duration

SFVoiceAnalytics
• Jitter
• Shimmer
• Pitch
• Voicing
Speech Task Updates

SFTranscription
• Speaking rate
• Average pause duration

SFVoiceAnalytics
• Jitter
• Shimmer
• Pitch
• Voicing

Advances in Speech Recognition

WWDC 2019
CareKit
June 2016

CareKit

Care Contents

Hamstring Stretch: 5 mins

Pain: Lower Back

Blood Glucose: After dinner

Insights

Threshold Alerts

Pain: High pain level.

Insights

Medication Adherence: Your medication adherence was 0% last week.

Dr. Maria Ruiz
Physician

Contact Info

Phone: 888-555-5512

Email: mruiz2@mac.com

Send reports
2017
UI Updates
2019 WWDC TODAY CareKit 2.0
CareKit UI
Naive UI Views

CareKit Store
On-Device Database
Doxylamine
7:30 AM
import CareKitUI

let taskView = OCKSimpleTaskView()
import CareKitUI

let taskView = OCKSimpleTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "7:30 AM"
import CareKitUI

let taskView = OCKSimpleTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "7:30 AM"
import CareKitUI

let taskView =
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "7:30 AM"
import CareKitUI

let taskView =
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "7:30 AM"
import CareKitUI

let taskView = OCKInstructionsTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "7:30 AM"
import CareKitUI

let taskView = OCKInstructionsTaskView()

taskView.headerView.titleLabel.text = "Doxylamine"

taskView.headerView.detailLabel.text = "7:30 AM"

Take the tablet with a full glass of water.

Mark as Completed
import CareKitUI

let taskView = OCKInstructionsTaskView()

taskView.headerView.titleLabel.text = "Doxylamine"

taskView.headerView.detailLabel.text = "7:30 AM"

taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKInstructionsTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "7:30 AM"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView =
    taskView.headerView.titleLabel.text = "Doxylamine"
    taskView.headerView.detailLabel.text = "7:30 AM"
    taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView =
    taskView.headerView.titleLabel.text = "Doxylamine"
    taskView.headerView.detailLabel.text = "1 event remaining today"
    taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKGridTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "1 event remaining today"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKGridTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "1 event remaining today"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKGridTaskView()

taskView.headerView.titleLabel.text = "Doxylamine"

taskView.headerView.detailLabel.text = "1 event remaining today"

taskView.instructionsLabel.text = "Take the tablet with a full glass of water."

taskView.collectionView.dataSource = self
import CareKitUI

let taskView =
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "1 event remaining today"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
taskView.collectionView.dataSource = self
import CareKitUI

let taskView =
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "1 event remaining today"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKChecklistTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "1 event remaining today"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKChecklistTaskView()

taskView.headerView.titleLabel.text = "Doxylamine"

taskView.headerView.detailLabel.text = "1 event remaining today"

taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
import CareKitUI

let taskView = OCKChecklistTaskView()
taskView.headerView.titleLabel.text = "Doxylamine"
taskView.headerView.detailLabel.text = "1 event remaining today"
taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
taskView.appendItem(withTitle: "2:00 PM")
import CareKitUI

let taskView =
    taskView.headerView.titleLabel.text = "Doxylamine"
    taskView.headerView.detailLabel.text = "1 event remaining today"
    taskView.instructionsLabel.text = "Take the tablet with a full glass of water."
    taskView.appendItem(withTitle: "2:00 PM")
import CareKitUI

let taskView =
taskView.headerView.titleLabel.text = "Track your Nausea"
taskView.headerView.detailLabel.text = "Anytime today"
taskView.instructionsLabel.text = "Tap the button below anytime you experience nausea."
import CareKitUI

let taskView = OCKSimpleLogTaskView()
taskView.headerView.titleLabel.text = "Track your Nausea"
taskView.headerView.detailLabel.text = "Anytime today"
taskView.instructionsLabel.text = "Tap the button below anytime you experience nausea."
import CareKitUI

let taskView = OCKSimpleLogTaskView()
taskView.headerView.titleLabel.text = "Track your Nausea"
taskView.headerView.detailLabel.text = "Anytime today"
taskView.instructionsLabel.text = "Tap the button below anytime you experience nausea."
import CareKitUI

let taskView = OCKSimpleLogTaskView()

    taskView.headerView.titleLabel.text = "Track your Nausea"
    taskView.headerView.detailLabel.text = "Anytime today"
    taskView.instructionsLabel.text = "Tap the button below anytime you experience nausea."

    taskView.appendItem(withTitle: "9:10 PM", detail: "Value logged")
import CareKitUI

let chartView = OCKCartesianChartView(type:.bar)
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]
import CareKitUI

let chartView = OCKCartesianChartView(type: .bar)
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]
import CareKitUI

let chartView = OCKCartesianChartView(type:)
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]
import CareKitUI

let chartView = OCKCartesianChartView()
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]
import CareKitUI

let chartView = OCKCartesianChartView(type:.scatter)
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]
import CareKitUI

let chartView = OCKCartesianChartView(type:
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]

import CareKitUI

let chartView = OCKCartesianChartView(type:
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]
import CareKitUI

let chartView = OCKCartesianChartView(type:.line)
chartView.headerView.titleLabel.text = "Trends"
chartView.headerView.detailLabel.text = "Nausea occurrences"
chartView.dataSeries = [OCKDataSeries(values: [5, 4, 7, 9, 2, 3, 1], title: "Nausea")]

import CareKitUI

let contactView = OCKContactCardView()
contactView.headerView.titleLabel.text = "Matthew Reiff"
contactView.headerView.detailLabel.text = "Family Practice"
contactView.headerView.iconImageView.image = UIImage(named: "matthew")
contactView.instructionsLabel.text = "Dr. Reiff is an OBGYN with 13 years of experience."
contactView.callButton.setTitle("Call", for: .normal)
CareKit UI
Naive UI Views

CareKit Store
On-Device Database
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")

// create an instance of the Patient entity
let patient = OCKPatient(identifier: "patient", givenName: "Jane", familyName: "Appleseed")
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")

// create an instance of the Patient entity
let patient = OCKPatient(identifier: "patient", givenName: "Jane", familyName: "Appleseed")

// add the newly created Patient entity to the store asynchronously
store.addPatient(patient) { result in
  switch result {
  case .success(let patient):
    print("ingested patient with id - \(patient.localDatabaseID)")
  case .failure(let error):
    print("ingestion failed with reason - \(error.localizedDescription)")
  }
}
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")

// create an instance of the Patient entity
let patient = OCKPatient(identifier: "patient", givenName: "Jane", familyName: "Appleseed")

// add the newly created Patient entity to the store asynchronously
store.addPatient(patient) { result in

    switch result {
    case .success(let patient):
        print("ingested patient with id - \(patient.localDatabaseID)")
    case .failure(let error):
        print("ingestion failed with reason - \(error.localizedDescription)")
    }
}
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")

// query patient with identifier asynchronously from the CareKitStore
store.fetchPatient(withIdentifier: "patient") { result in
  switch result {
  case .success(let patient):
    // create a new CarePlan that is associated with the patient
    let carePlan = OCKCarePlan(identifier: "carePlan",
                               title: "Healthy Pregnancy",
                               patientID: patient.localDatabaseID)

    // add CarePlan to the CareKitStore
    store.addCarePlan(carePlan)
  case .failure(let error):
    print("ingestion failed with reason - \\n          (error.localizedDescription)")
  }
}
import CareKitStore

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")

// query patient with identifier asynchronously from the CareKitStore
store.fetchPatient(withIdentifier: "patient") { result in
    switch result {
    case .success(let patient):
        // create a new CarePlan that is associated with the patient
        let carePlan = OCKCarePlan(identifier: "carePlan",
                                   title: "Healthy Pregnancy",
                                   patientID: patient.localDatabaseID)

        // add CarePlan to the CareKitStore
        store.addCarePlan(carePlan)
    case .failure(let error):
        print("ingestion failed with reason - \(error.localizedDescription)")
    }
}
import CareKitStore

let store = OCKStore(identifier: "store")
import CareKitStore

let store = OCKStore(identifier: "store")

// schedule a medication to be taken everyday at 7am
let breakfastTime = DateComponents(year: 2019, month: 6, day: 5, hour: 7, minute: 0)
let breakfastDose = OCKSchedule.dailyAtTime(start: breakfastTime)
import CareKitStore

let store = OCKStore(identifier: "store")

// schedule a medication to be taken everyday at 7am
let breakfastTime = DateComponents(year: 2019, month: 6, day: 5, hour: 7, minute: 0)
let breakfastDose = OCKSchedule.dailyAtTime(start: breakfastTime)

// schedule a medication to be taken every other day at 12pm
let lunchTime = DateComponents(year: 2019, month: 6, day: 5, hour: 12, minute: 0)
let lunchDose = OCKScheduleElement(start: lunchTime, interval: DateComponents(day: 2))
import CareKitStore

let store = OCKStore(identifier: "store")

// schedule a medication to be taken everyday at 7am
let breakfastTime = DateComponents(year: 2019, month: 6, day: 5, hour: 7, minute: 0)
let breakfastDose = OCKSchedule.dailyAtTime(start: breakfastTime)

// schedule a medication to be taken every other day at 12pm
let lunchTime = DateComponents(year: 2019, month: 6, day: 5, hour: 12, minute: 0)
let lunchDose = OCKScheduleElement(start: lunchTime, interval: DateComponents(day: 2))

// compose these schedules together
let schedule = OCKSchedule(composing: [breakfastDose, lunchDose])
import CareKitStore

let store = OCKStore(identifier: "store")

// schedule a medication to be taken everyday at 7am
let breakfastTime = DateComponents(year: 2019, month: 6, day: 5, hour: 7, minute: 0)
let breakfastDose = OCKSchedule.dailyAtTime(start: breakfastTime)

// schedule a medication to be taken every other day at 12pm
let lunchTime = DateComponents(year: 2019, month: 6, day: 5, hour: 12, minute: 0)
let lunchDose = OCKScheduleElement(start: lunchTime, interval: DateComponents(day: 2))

// compose these schedules together
let schedule = OCKSchedule(composing: [breakfastDose, lunchDose])

// create a new task to take Ibuprofen medication with the composed schedule
var task = OCKTask(identifier: "medicationTask", title: "Doxylamine", schedule: schedule)
task.instructions = "Take the tablet with a full glass of water."
import CareKitStore

let store = OCKStore(identifier: "store")
import CareKitStore

let store = OCKStore(identifier: "store")

// query elements that were generated in the past week
let startDate = Calendar.current.date(byAdding: .day, value: -7, to: Date())
let query = OCKInsightQuery(startDate: startDate, endDate: Date())
import CareKitStore

let store = OCKStore(identifier: "store")

// query elements that were generated in the past week
let startDate = Calendar.current.date(byAdding: .day, value: -7, to: Date())
let query = OCKInsightQuery(startDate: startDate, endDate: Date())
store.fetchInsights(forTask: "medicationTask", query: query, dailyAggregator: { events in
    // this block provides all events generated on a daily basis and should return a Double
    var dailyDosage = 0.0
    for event in events {
        if let outcome = event.outcome {
            dailyDosage += outcome.values?.first?.doubleValue ?? 0.0
        }
    }
    return dailyDosage
}, completion: { values in
    // use the computed values for further analysis or for charting
})
import CareKitStore

let store = OCKStore(identifier: "store")

// query elements that were generated in the past week
let startDate = Calendar.current.date(byAdding: .day, value: -7, to: Date())
let query = OCKInsightQuery(startDate: startDate, endDate: Date())
store.fetchInsights(forTask: "medicationTask", query: query, dailyAggregator: {
    events in
        // this block provides all events generated on a daily basis and should return a Double
        var dailyDosage = 0.0
        for event in events {
            if let outcome = event.outcome {
                dailyDosage += outcome.values?.first?.doubleValue ?? 0.0
            }
        }
        return dailyDosage
    }, completion: {
        values in
            // use the computed values for further analysis or for charting
    }}
import CareKitStore

let store = OCKStore(identifier: "store")

// query elements that were generated in the past week
let startDate = Calendar.current.date(byAdding: .day, value: -7, to: Date())
let query = OCKInsightQuery(startDate: startDate, endDate: Date())

store.fetchInsights(forTask: "medicationTask", query: query, dailyAggregator: { events in
    // this block provides all events generated on a daily basis and should return a Double
    var dailyDosage = 0.0
    for event in events {
        if let outcome = event.outcome {
            dailyDosage += outcome.values?.first?.doubleValue ?? 0.0
        }
    }
    return dailyDosage
}, completion: { values in
    // use the computed values for further analysis or for charting
})
Care Plan
- Attributes
- Relationships
  - carePlans
- Schedule
  - Attributes
  - Relationships
- Outcome
  - Attributes
  - Relationships
- Value
  - Attributes
  - Relationships
- Patient
  - Attributes
  - Relationships
  - carePlans
- Contact
  - Attributes
  - Relationships
- Task
  - Attributes
  - Relationships
  - schedules
  - outcomes
- Contacts
  - Attributes
  - Relationships
  - tasks
Synchronizer

Leverages Combine
Synchronizer Database

Synchronizer
Leverages Combine

Database
CoreData/Other

Doxylamine
0 events remaining today
7:30 AM
2:00 PM
Take the tablet with a full glass of water.

Doxylamine
1 event remaining today
7:30 AM
2:00 PM
Take the tablet with a full glass of water.

Nausea vs. Doxylamine Intake
This week

Track your Nausea
Anytime today
Tap the button below anytime you experience nausea.
I'm Feeling Nauseous
9:10 PM Value logged
Synchronizer Database

Synchronizer Leverages Combine

Database CoreData/Other

Doxylamine
0 events remaining today

7:30 AM
2:00 PM
Take the tablet with a full glass of water.

Doxylamine
1 event remaining today

7:30 AM
2:00 PM
Take the tablet with a full glass of water.

Nausea vs. Doxylamine Intake
This week

Track your Nausea
Anytime today

Tap the button below anytime you experience nausea.
// importing CareKit also imports CareKitUI and CareKitStore
import CareKit
// importing CareKit also imports CareKitUI and CareKitStore
import CareKit

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")
// importing CareKit also imports CareKitUI and CareKitStore
import CareKit

// create an instance of the CareKitStore with a unique identifier
let store = OCKStore(identifier: "store")

// instantiate a SynchronizedStoreManager by passing in the store
let manager = OCKSynchronizedStoreManager(wrapping: store)
import CareKit

let store = OCKStore(identifier: "store")

let manager = OCKSynchronizedStoreManager(wrapping: store)

let taskListViewController = OCKTaskListViewController(storeManager: manager)

present(taskListViewController, animated: true, completion: nil)
CareKit
CareKit

ResearchKit
Available on GitHub!

**ResearchKit**
[github.com/researchkit/researchkit](https://github.com/researchkit/researchkit)

**CareKit**
[github.com/carekit-apple/CareKit/](https://github.com/carekit-apple/CareKit/)
2019

WWDC

TODAY
Please contribute!
## More Information

developer.apple.com/wwdc19/217

<table>
<thead>
<tr>
<th>Event</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health and Fitness Technologies Lab</td>
<td>Wednesday, 3:00</td>
</tr>
<tr>
<td>Introducing Combine and Advances in Foundation</td>
<td>Thursday, 10:00</td>
</tr>
<tr>
<td>ResearchKit and CareKit Lab</td>
<td>Thursday, 12:00</td>
</tr>
</tbody>
</table>