Beginning Carekit Development

Develop CareKit Applications Using Swift

Christopher Baxter

Apress®

Beginning CareKit Development

Develop CareKit Applications Using Swift

Christopher Baxter

Apress®

Beginning CareKit Development

Christopher Baxter
North Yorkshire
United Kingdom

ISBN-13 (pbk): 978-1-4842-2225-6 ISBN-13 (electronic): 978-1-4842-2226-3
DOI 10.1007/978-1-4842-2226-3

Library of Congress Control Number: 2016959390

Copyright © 2016 by Christopher Baxter

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Lead Editor: Aaron Black
Technical Reviewer: Idriss Juhoor
Editorial Board: Steve Anglin, Pramila Balan, Laura Berendson, Aaron Black, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham, Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing
Coordinating Editor: Jessica Vakili
Copy Editor: Corbin Collins
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text are available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

I'd like to dedicate this book to my gorgeous wife, Melanie,
and my amazing children, Jason and Holly, who mean everything to me.

Contents at a Glance

Contents

About the Author

Christopher Baxter has vast experience in creating mobile apps, and has been the lead iOS engineer and architect on more than 50 apps in a wide variety of industries. With over 26 years experience in software development, Chris has been working with iOS since its first publication in 2008, as well as with the Android platform and Windows Phone. He is also the founder and director of a mobile consultancy based in the UK. He can be reached via his consultancy business at `www.catalystmobile.co`.

About the Technical Reviewer

Idriss Juhoor is a world-travelling software engineer from a small island in the middle of the Indian Ocean. He's worked for both small startups and large companies in different parts of the globe and now focuses on mobile healthcare. When he's not writing health apps, he's connecting stuff to his phone using solder and Bluetooth chips. You can find him on twitter: `@foiegras33`.

Acknowledgments

This book would not be possible without the existence of Apple and its new language Swift, which inspired me to write it.

I'd like to express my gratitude to Apress for publishing this book and to the following editors, who have put a lot of energy into making this a great book: Aaron Black, Jessica Vakii, and James Markham.

I'd also like to thank Idriss Juhoor, who has been the technical reviewer for the book. He is a keen advocate of applying good engineering practices to software projects and is involved in the digital healthcare sector.

Lastly, I'd like to thank my close friend Tom Gleeson who set the barrier for success so high I had to raise my game and write a book.

Introduction

Welcome to *Beginning CareKit Development*. My goal is to provide a practical guide for developers to create CareKit-based applications using the Swift language.

I've started with the basics, using a step-by-step approach to learning all aspects of creating a CareKit iOS application that could serve as the basis for a digital patient Care Plan. You'll see the key modules and concepts of CareKit, starting off by installing and building the open source framework.

Examples within demonstrate how to customize CareKit modules and integrate them with other frameworks, such as ResearchKit and HealthKit, and how to extend the application with Today extensions and an Apple Watch app.

By the end of the book you'll to be able to fully utilize CareKit for your own personal Care Plans. This is the future of patient care: health-tracking apps that put patients in control of their day-to-day care.

CHAPTER 1

Getting Started

This chapter introduces you to Apple's CareKit. After some background on CareKit's base classes and modules and the example app, we'll then move on to gain an understanding of how the framework is organized, the architecture of the CareKit framework, and the anatomy and key modules provided within the framework, along with some best practices for working with it.

Understanding the Core Elements of CareKit

CareKit was first introduced by Apple at a media event in March 2016. It's an open source framework that enables developers to build apps that "empower people to take on an active role in their care." iPhone apps that support this framework allow users to track their ongoing condition, symptoms, and medication to get an overall wider view of their health and share this with their care team or personal contacts. CareKit can support a wide range of care plans—from managing chronic illnesses to recovery programs after injury or surgery, and general care plans to improve health.

The CareKit framework was released as open source on April 29, 2016, and is accompanied by four example applications. The following apps are all available on the App Store and showcase the core features of CareKit, demonstrating real-world digital patient care:

- *One Drop* is for managing diabetes. It helps you track your food and medication intake, as well as activity. There's an Apple Watch app, too.
- *Start* covers the monitoring, treatment, and medication of depression, helping to diagnose mental health problems and track progress of the treatments.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-1-4842-2226-3_1) contains supplementary material, which is available to authorized users.

© Christopher Baxter 2016
C. Baxter, *Beginning CareKit Development*, DOI 10.1007/978-1-4842-2226-3_1

- *Glow Nurture* is a pregnancy tracker. It helps you track all the important milestones within a pregnancy, such as due dates, doctor's appointments, and so on, and also allows you to enter symptoms and measurements such as weight.
- *Glow Baby* is made by the same company as Glow Nurture, taking up the mantle after the baby is born. It covers breastfeeding, sleep, feeding, and diaper cycles.

You can see that the four sample applications are very different from each other, although they share the same underlying anatomy and structure that all CareKit apps do. In some cases, the CareKit integration is just one part of the care plan, which might include a broader set of features.

CareKit applications can be customized beyond the basic appearance of the standard module controllers provided within CareKit. We talk more about this in Chapter 4.

Apple has open sourced CareKit, and the source code comes with one example application called OCKSample, which demonstrates all the key models within CareKit.

Links to the source code, documentation, and other information can be found on `www.apple.com/researchkit/` and `www.carekit.org`. The source code is hosted on Github at `https://github.com/carekit-apple`.

Fundamentally, CareKit manages various scenes for scheduling patient activities, monitoring treatment, and providing feedback to the patient and their connections. You can find an overview of these modules and the key data classes in the official documentation. If you've already read the documentation, you may want to skip to the "CareKit Framework Architecture" section.

Framework Organization

There are six modules in CareKit. Four relate to providing the user interface, and two are for managing data.

User interface modules:

- Care Card
- Symptom and Measurement Tracker
- Insights
- Connect

Data modules:

- Care Plan Store
- Documents Exporter

As you will see, generally most CareKit classes are easily recognizable as they are prefixed with `OCK`. We can now take a closer look at each module.

User Interface Modules

CareKit provides a number of ViewController-derived objects that take care of loading the appropriate data and presenting it to the user. Each ViewController interacts with the Care Plan Store and various key data objects that represent the care plan.

Care Card

The Care Card manages intervention activities that a user needs to perform as part of the treatment for their condition. The Care Card is a scene managed by the OCKCareCardViewController object and presents the intervention activities to the user. Intervention activities are basically scheduled tasks that the user must perform as part of their treatment—for example, taking medication three times a day.

You can read more about the Care Card scene and what it's used for in the official documentation. Chazpter 4 covers creating, presenting, and interacting with the Care Card view.

Figure 1-1 shows a typical Care Card ViewController.

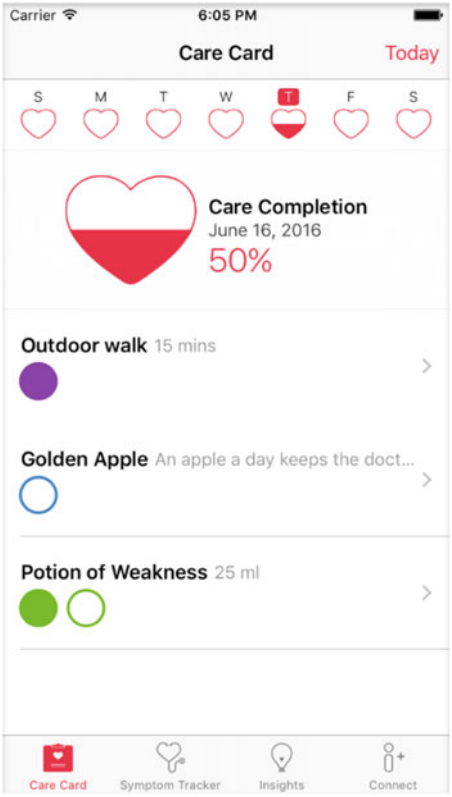

Figure 1-1. *Care Card ViewController*

Figure 1-2 shows the detail view for a specific intervention activity.

Figure 1-2. *Care Card detail view*

Symptom and Measurement Tracker

The Symptom and Measurement Tracker manages activities that are used to evaluate the effectiveness of the treatments. There are two types of these activities:

- *Subjective activities* allow users to record symptoms like their mood or pain scales. You as the developer can implement your own tasks to record these symptoms or integrate with existing tasks provided through ResearchKit.
- *Objective activities* are measurements that can be entered manually or recorded from devices or even HealthKit—for example, blood pressure. The Symptom and Measurement Tracker scene is managed by the OCKSymptomTrackerViewController.