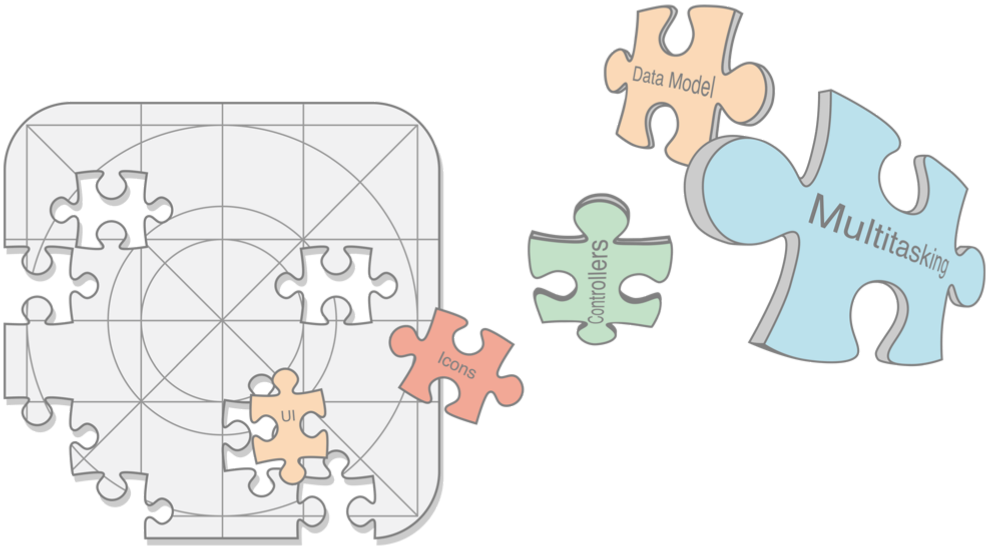
About iOS App Architecture

Apps need to work with the iOS to ensure that they deliver a great user experience. Beyond just a good design for your app's design and user interface, a great user experience encompasses many other factors. Users expect iOS apps to be fast and responsive while expecting the app to use as little power as possible. Apps need to support all of the latest iOS devices while still appearing as if the app was tailored for the current device. Implementing all of these behaviors can seem daunting at first but iOS provides the help you need to make it happen.



This document highlights the core behaviors that make your app work well on iOS. You might not implement every feature described in this document but you should consider these features for every project you create.

Note: Development of iOS apps requires an Intel-based Macintosh computer with the iOS SDK installed. For information about how to get the iOS SDK, go to the iOS Dev Center.

At a Glance

When you are ready to take your ideas and turn them into an app, you need to understand the interactions that occur between the system and your app.

Apps Are Expected to Support Key Features

The system expects every app to have some specific resources and configuration data, such as an app icon and information about the capabilities of the app. Xcode provides some information with every new project but you must supply resource files and you must make sure the information in your project is correct before submitting your app.

Relevant Chapter: Expected App Behaviors

Apps Follow Well-Defined Execution Paths

From the time the user launches an app to the time it quits, apps follow a well-defined execution path. During the life of an app, it can transition between foreground and background execution, it can be terminated and relaunched, and it can go to sleep temporarily. Each time it transitions to a new state, the expectations for the app change. A foreground app can do almost anything but background apps must do as little as possible. You use the state transitions to adjust your app's behaviors accordingly.

Relevant Chapter: The App Life Cycle, Strategies for Handling App State Transitions

Apps Must Run Efficiently in a Multitasking Environment

Battery life is important for users, as is performance, responsiveness, and a great user experience. Minimizing your app's usage of the battery ensures that the user can run your app all day without having to recharge the device, but launching and being ready to run quickly are also important. The iOS multitasking implementation offers good battery life without sacrificing the responsiveness and user experience that users expect, but the implementation requires apps to adopt system-provided behaviors.

Relevant Chapters: Background Execution, Strategies for Handling App State Transitions

Communication Between Apps Follows Specific Pathways

For security, iOS apps run in a sandbox and have limited interactions with other apps. When you want to communicate with other apps on the system, there are specific ways to do so.

Relevant Chapter: Inter-App Communication

Performance Tuning is Important for Apps

Every task performed by an app has a power cost associated with it. Apps that drain the user's battery create a negative user experience and are more likely to be deleted than those that appear to run for days on a single charge. So be aware of the cost of different operations and take advantage of power-saving measures offered by the system.

Relevant Chapter: Performance Tips

How to Use This Document

This document is not a beginner's guide to creating iOS apps. It is for developers who are ready to polish their app before putting it in the App Store. Use this document as a guide to understanding how your app interacts with the system and what it must do to make those interactions happen smoothly.

Prerequisites

This document provides detailed information about iOS app architecture and shows you how to implement many app-level features. This book assumes that you have already installed the iOS SDK, configured your development environment, and understand the basics of creating and implementing an app in Xcode.

If you are new to iOS app development, read *Start Developing iOS Apps (Swift)*. That document offers a step-by-step introduction to the development process to help you get up to speed quickly. It also includes a hands-on tutorial that walks you through the app-creation process from start to finish, showing you how to create a simple app and get it running quickly.

See Also

If you are learning about iOS, read *iOS Technology Overview* to learn about the technologies and features you can incorporate into your iOS apps.