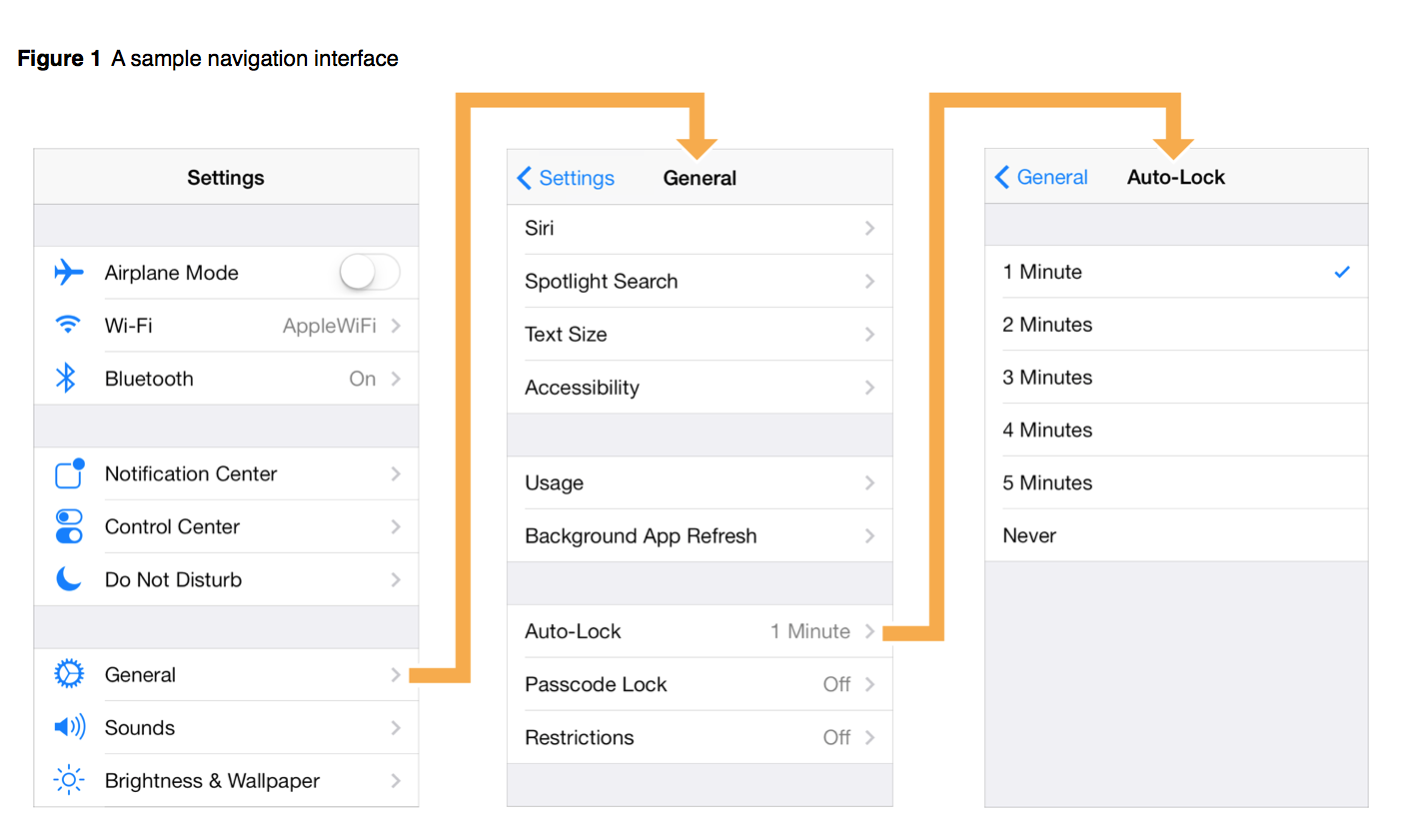
UINavigationController

The UINavigationController class implements a specialized view controller that manages the navigation of hierarchical content.

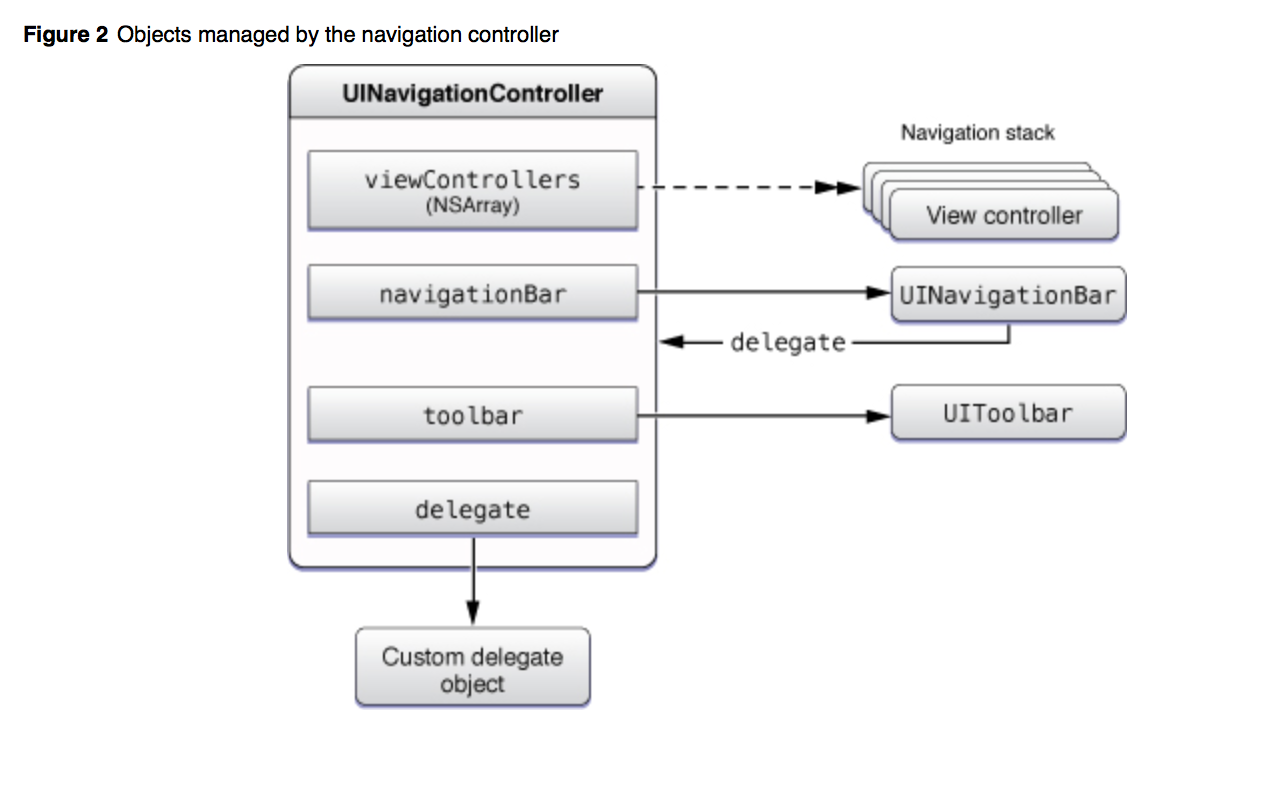


Use of Navigation Bar

Navigation bar contains the navigation buttons of a navigation controller, which is a stack of view controllers which can be pushed and popped. Title on the navigation bar is the title of the current view controller.

Navigation Controller Views

A navigation controller is a container view controller—that is, it embeds the content of other view controllers inside of itself. You access a navigation controller’s view from its [view](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html#//apple_ref/occ/instp/UIViewController/view) property. This view incorporates the navigation bar, an optional toolbar, and the content view corresponding to the topmost view controller.



XIB stands for the XML Interface Builder.

Interface Builder is a software application which allows you to develop Graphical User Interface with the help of Cocoa and carbon. The generated files are either stored as NIB or XIB files. These files are copied into the app bundle and loaded at run time to provide the user interface for the application. XIB files were introduced in 2007 with Leopard (Xcode 3.0).

Apple introduced the concept of "storyboarding" in iOS5 SDK to simplify and better manage screens in your app. You can still use the .xib way of development.

Pre-storyboard, each UIViewController had an associated .xib with it. Storyboard achieves two things:

* .storyboard is essentially one single file for all your screens in the app and it shows the flow of the screens. You can add segues/transitions between screens, this way. So, this minimizes the boilerplate code required to manage multiple screens.

Minimizes the overall no. of files in an app.

**Storyboard**

Storyboards are an exciting feature first introduced way back in iOS 5 that save you a lot of time building user interfaces for your apps.

Storyboards have a number of advantages:

* You can visually lay out all your view controllers in “scenes” and describe the connections between them. With a storyboard you have a better conceptual overview of all the scenes in your app.
* Storyboards can describe the transitions between the various scenes. These transitions are called “segues” and you create them by connecting your view controllers right in the storyboard. Thanks to segues you need less code to take care of your UI.
* Storyboards make working with table views a lot easier with prototype cells and static cells features. You can design your table views almost completely in the storyboard editor, cutting down on the amount of code you have to

Execution States for Apps

App states

