

Lesson 2: iOS Human Interface Guidelines, Part 2

INTRODUCTION

This lesson continues the discussion of the Apple iOS Human Interface Guidelines, a document to help developers create a solid app founded on Apple's design principles.

LESSON OBJECTIVES

By the end of this lesson, the student will be able to:

1. Identify the function of the available iOS technologies.
2. Identify the content view that will appropriately display data given a design scenario.
3. List the views that are used primarily for interaction and are temporary.
4. Differentiate the use guidelines between a navigation bar, toolbar and a tab bar.
5. Demonstrate how to add a tab with an icon and title that is associated with a scene to the tab bar.
6. Create an application that uses the Tab Bar template.

LEARNING SEQUENCE

Required Reading	Read the following: <ul style="list-style-type: none">• Lesson 2: Human Interface Guidelines and Design
Resources	View the following: <ul style="list-style-type: none">• Xcode 5 Tutorial: iOS 7 Core Location: How To Get Current Location (13:45)• iOS7 App Icon Design in Photoshop (8:56) Other resources: iOS Human Interface Guidelines accessed through the Apple Developer website > iOS icon > iOS Human Interface Guidelines
Assignments	Complete the following: <ol style="list-style-type: none">1. Practice Exercise – Tab It2. Quiz 2



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](#).

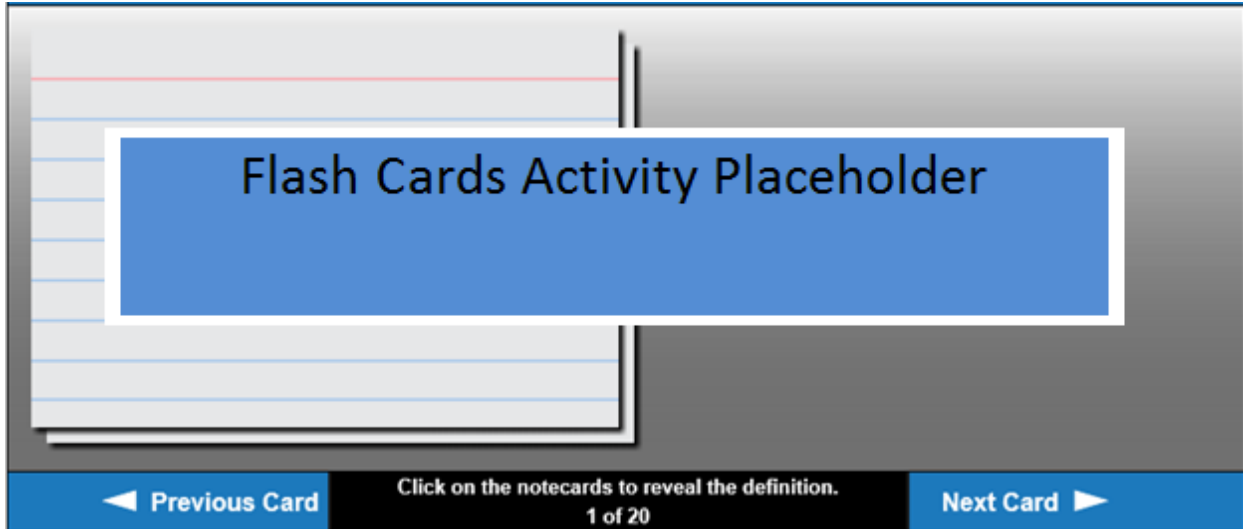
Authoring Organization: Collin College

Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

KEY TERMS

As you read your lesson, pay close attention to the [key terms and phrases](#) listed throughout the lesson. These terms and concepts are important to your understanding of the information provided in the lesson.



INSTRUCTION

Lesson 10 looked at the first two sections, UI Design Basics and Design Strategies, of iOS Human Interface Guidelines. This lesson will highlight concepts from the last three sections, iOS Technologies, UI Elements, and Icon and Image Design. Open the iOS Human Interface Guidelines document.

1. Browse to the [Apple Developer website](#).
2. Select the iOS Apps icon.
3. Click on the iOS Human Interface Guidelines link.

Keep this document open for reference throughout the lesson.

iOS Technologies

There are numerous iOS technologies that a developer can incorporate into an app that support common tasks that iOS device users expect. The following iOS technologies are listed alphabetically.

iAd Rich Media Ads

Revenue can be generated in apps by **iAds**. There are three different styles of ads that may appear in the app.

1. **Standard Banner:** The standard banner ad is a small area of the screen that displays the ad. It is usually at the bottom of the screen and remains present on the screen while the app is running.
2. **Medium Rectangle Banner:** This ad stays the same size (300pt x 250pt). Note that medium rectangle banners are available only with iPads that run iOS 6.0 or later.



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](#).

Authoring Organization: Collin College

Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

3. Full Screen Banners: The full screen banner takes up most of the screen but is only visible at specific times.

Make sure that ads can be seen in both portrait and landscape screen views.

iCloud

iCloud storage is used for user-generated content and user preferences. It is not a place where a database or the data source resides for the app. The iCloud usage in an app needs to be transparent to the user. The user should not have to click a button to send data to the iCloud. The app should be set up so that it automatically sends and receives information to/from iCloud. Another thing to keep in mind is that the iCloud space is limited. iCloud space is allocated for each iTunes account and is therefore a user space not a developer space. An app developer should plan for the possibility that cloud space is unavailable. It is possible for the user to turn off the iCloud globally so that nothing goes to iCloud, or they can turn the iCloud off for an individual app. The standard storage protocol is to save information locally on the device and also on iCloud, per user settings. Therefore, do not prompt the user in the app to choose the storage location for information generated in the app. It is also convention to warn the user before information is deleted.

In-App Purchases

The user can purchase digital content inside the app including an upgraded app version, subscriptions, and additional virtual content. **In-App purchases** use **Store Kit framework**. These purchasing options should be integrated into the app and should have simple titles and descriptions. It is important to note that the purchase confirmation alert should not be modified or removed.

Game Center

Game Center is a social gaming area that lets users organize multiplayer games, view leader boards and achievements, and add friends. The app should use the Game Center sign-in and user interface as opposed to creating a separate sign-in. Additionally, the users should be allowed to turn off voice chat if they choose to do so.

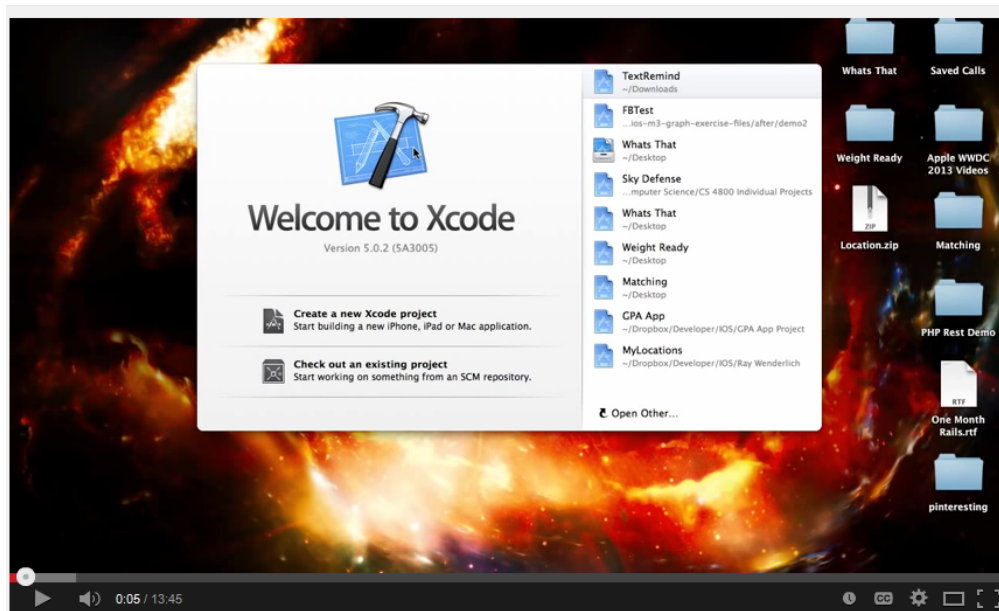
Location Services and Data Privacy

The location service allows the app to acquire the user's location based on cellular, Wi-Fi, or GPS information. This may benefit the user by finding a restaurant nearby or acquiring the time zone that he or she is in, for example. The location service may be turned on and off by the user. Additionally, the user must give an app permission to use the current location. It is beneficial for the user to be notified why the app needs to use location information. Therefore, only ask for the user's location if it is beneficial to the app function. Do not ask for the user's location if the location service is turned off globally.

Watch the video, *Xcode 5 Tutorial: iOS 7 Core Location: How To Get Current Location* (13:45) to learn how to use the Core Location framework to get the user's current location.



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).
Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands
Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)



Multitasking

Multitasking allows suspended apps to load quickly. This means that more than one app can be running, but there is only one app in the foreground. When an app is no longer in the foreground and the user picks a new app, the first app goes to the background. Developers need to plan for the app to stop and resume easily as well as enable the app to finish tasks in the background, if necessary. Another multitasking feature that needs to be incorporated into an app is the ability to support a phone status bar. If there is a phone call while the user is in an app, the user needs to be able to see the status bar for the call. Browse to the article, [Understanding iOS Status Bar Symbols](#), to see examples of status bar icons.

Developers also need to be aware of audio multitasking options for their app. Audio can either be set to run just while the app is in the foreground or it can be set to run while the app is in the background. For example, a radio or music app should allow the user to listen to audio while they are using other apps. This requires the audio to persist beyond the app being used in the foreground. Another aspect of audio that should be considered is sound clash when an app in the foreground is producing audio. Make sure the app meshes well with other apps, in terms of audio choices. Allow the user access to the media remote controls for audio.

Notification Center

The **notification center** is the location for gathered notifications. A **local notification** is from an app that may be in the foreground or may be in the background. A **push notification** is from an app remote server to all devices with that app. For example, in a sports app, all users with push notifications for



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](#).
Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands
Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

scoring updates will receive a push notification from that app's server. There are three different types of push notification styles.

1. **Banner:** A banner is a small transparent notification view that appears and disappears usually at the top of the screen.
2. **Alert:** An alert push notification is a standard alert view that may cause an app to pause in the foreground.
3. **Badge:** A badge notification is a small oval with a number over the app icon. This is a common push notification for unread emails or text messages.

Notifications should be used sparingly; the user should not be bombarded with unnecessary notifications.

Passbook

Passbook is a digital way of keeping track of physical tokens such as coupons, boarding passes, tickets, and membership cards. Keep the interface uncluttered. Put the company name in the logo text field and provide a white monochromatic logo for the best appearance. A rectangular barcode is recommended as opposed to a square barcode. A square barcode may crowd other fields either above or below the barcode. Go to [iOS Human Interface Guidelines > iOS Technologies > Passbook](#) and scroll down to see an example.

Printing

Wireless printing is available for iOS 4.2 and later. There should be an action button to allow the user to control when an item will be printed. The app should also incorporate different printing options for the material being printed. The printing option should not be made available in the app if the device does not support wireless printing.

Routing

A map can be used to display the different routes that users can choose from to get to their destination. **Routing** is available for iOS 6 or later. It provides transit options for a route that the user has selected beyond walking or driving, which are considered mapping. Routing gives step-by-step instructions for transit by bus, bike, or foot. The user should not have to re-enter the map because there should be visual and text instructions available with routing.

Social Media

When social media is being integrated into an app, it should be setup so that the user does not have to leave the app to interact with social media. Do not make the user sign in, if possible. A feature of the iPhone and iPad that makes this possible is the setting that allows the device to keep track of the social media passwords. This setting allows the app to access social media with permission needed from user without typing a username and password.

Other iOS Technologies

Additional iOS technologies include the following:



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](#).

Authoring Organization: Collin College

Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

- **VoiceOver:** increases accessibility for visually impaired users or users with learning challenges
- **Edit Menu:** gives users the opportunity to select, copy and paste
- **Undo and Redo:** shaking the device starts the Undo process and allows users to undo what was typed, redo previously undone typing, or cancel altogether
- **Keyboards and Input Views:** allows a developer to design a custom input view instead of using the system-provided onscreen keyboard

For more information on these iOS technologies, click to expand [IOS Human Interface Guidelines > iOS Technologies](#). Click on the technology to view more information on it.

UI Elements

UI elements are visual elements in the app that respond to user interaction.

Bars

There are four different bars that will be briefly discussed. More information on these and the other types of bars can be found in the iOS HIG document under the [UI Elements > Bars](#).

1. **Status Bar:** This bar is seen in the upper edge of the screen and contains only necessary information about the device and the current environment. Do not create a customized status bar. The user is dependent on the consistency that the status bar provides. Additionally, prevent any scrolling of text through the status bar.
2. **Navigation Bar:** The navigation bar is for navigation in a hierarchal information structured app. It also allows the management of screen content. The navigation bar usually appears at the top of the screen below the status bar.
3. **Toolbar:** This bar holds the controls related to views or objects. It is always on the bottom of the screen for the iPhone and either on the top or the bottom of the screen for the iPad. The toolbar allows the user access to actions in the current context.
4. **Tab Bar:** The tab bar is located on the bottom of the screen and provides access to different views or subtasks. The tab bar will display no more than five tabs at once.

Content Views

Several different content views will be highlighted in this section.

1. **Activity:** This is the content view for a custom or system service, for example, printing and social media.
2. **Collection View:** The collection content view is used to organize content in a custom layout.
3. **Container View Controller:** The container content view allows the management set of child views. This includes management of tab bar view controller, navigation controller, and split view controller.
4. **Image View:** This content view holds images or sets of animated images.
5. **Map View:** The map content view contains geographical information.



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](#).

Authoring Organization: Collin College

Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

6. Page View Controller: The page view handles multiple pages with either scrolling or page turning.
7. Popover: This is used for the iPad only and will pop up when a control is selected.
8. Scroll View: The scroll view is for content larger than view size.
9. Split-View Controller: This view is only used for the iPad and has two panels next to each other. One panel will be the master panel and the other will be the detail panel.
10. Table View: This view has a single column with lots of rows.
11. Text View: The text view displays multiple lines of text.
12. Web View: This view shows web pages and html content.

Read through iOS HIG document > UI Elements > Content Views for a thorough discussion of each content view due to the importance of this section.

Controls

A control is used to manipulate content, provide user input, navigate within an app, or execute an action. Controls allow a user to interact with an app. The following table presents a list of controls and their function.

Control	Function
Activity Indicator	The control that shows that activity is taking place
Date Picker	Shows components of date and time
Contact Add Button	Allows user to add existing contact to text field
Detail Disclosure Button	Gives more information about the functionality of an item
Info Button	Used for configuration information about an app
Label	Displays static text or text that the user is not able to edit.
Network Activity Indicator	Shows that network activity is occurring for more than a few seconds
Page Control	Shows users the number of open views by displaying a dot for each open view in an app
Picker	Allows the user to make a choice from a set of values, like a generic version of the date picker
Progress View	Shows progress of a task (downloading email, for example)
Refresh control	Gives users an indication that the content is being refreshed
Segmented Control	Like a button but has multiple segments that offer choices



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

Authoring Organization: Collin College

Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

Slider	Allows adjustments to be made within a range of allowed values (setting the brightness of the screen, for example)
Stepper	Used to make small adjustments to a value by providing a plus and minus sign
Switch	Allows a user to select On or Off
Text Field	Used to accept a small amount of input from the user

More detailed information on controls can be found by going to the iOS HIG document > UI Elements > Controls.

Temporary Views

Temporary views include alerts, action sheets, and modal views. An **alert** gives the user important information about an app or a device. An alert will contain text and one or more buttons.

An **action sheet** appears due to an action that the user performed and displays two or more buttons. An action sheet can provide a range of choices or it can get user confirmation before completing a task.

Modal views can occupy the entire screen or a portion of the screen. Modal views contain both text and the controls necessary to complete a task. A Cancel button is also included to dismiss the view completely.

Custom Icon and Image Creation Guidelines

The app icon makes an app stand out in the App Store, so every app needs an app icon and a launch image. An app may also need custom icons for specific content, functions, or modes in navigation. Information on app icons and design can be found in the iOS HIG document > Icon and Image Design and contains the following sections:

1. Icon and Image Sizes: Size specifications (in pixels) are given.
2. App Icon: View the best practices for app, document, and spotlight and settings icons.
3. Launch Images: This image displays when the app is starting up and gives a first impression about the app to the user.
4. Bar Button Icons: If an app includes a task that cannot be represented by one of the standard icons, make sure the custom icon is simple and not easily mistaken for one of the provided standard icons.
5. Newsstand Icons: An app can use the Newsstand Kit for subscription-based content (like magazines, for example).
6. Web Clip Icons: For a web app or a website, a web clip icon can be used on the user's Home screen.



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

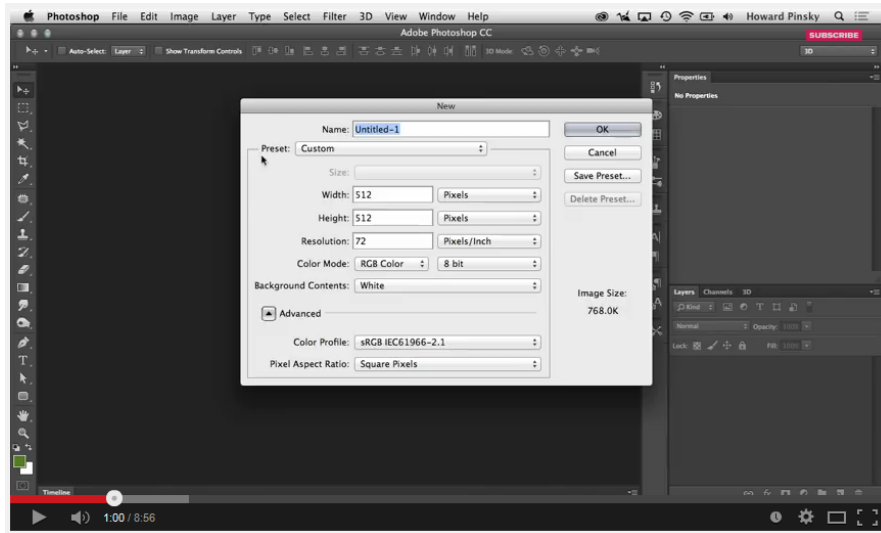
Authoring Organization: Collin College

Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands

Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)

7. Creating Resizable Images: Create resizable images to customize the background of UI elements like popovers, buttons, navigation bars, etc. An app will perform better if the developer provides resizable images for these elements.

Watch the video, *iOS7 App Icon Design in Photoshop* (8:56) for a discussion of tips and tricks for creating icons.



SUMMARY

This lesson completed the discussion of the iOS Human Interface Guidelines. Lessons 10 and 11 have only scratched the surface. A developer needs to be familiar with Apple's guidelines so that the app being designed can eventually be published in the App Store.

ASSIGNMENTS

1. Practice Exercise – Tab It
2. Quiz 2



This work by the National Information Security and Geospatial Technologies Consortium (NISGTC), and except where otherwise noted, is licensed under the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).
Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands
Copyright: © National Information Security, Geospatial Technologies Consortium (NISGTC)