Mobile Application Development
Aileen Pierce

DELEGATION

Delegation

- Delegation enables another object to help out with some of the work.
- A delegate is an object that takes responsibility for doing certain tasks for another object.
- Delegation is a common design pattern used in Cocoa Touch.
- Many UIKit classes allow customization of their behavior through delegation.

Protocols

- For a class to act as a delegate it needs to conform to a protocol.
- A protocol defines a set of methods a class must implement to conform to it.
  - A protocol can contain both required and optional methods
- That way objects of that class will be able to respond to those methods.

Protocols and Delegates

There are 3 steps in implementing a protocol:

1. Adopt the delegate protocol.
2. Implement the delegate methods.
3. Set the controller as the delegate

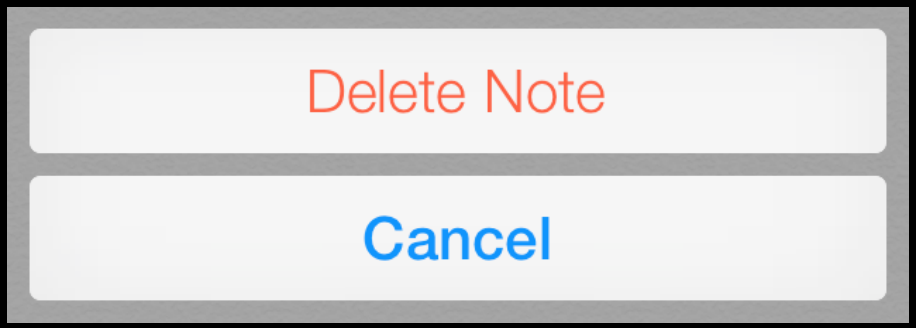
Text Field

- A text field is a single line area for entering text.
- UITextField class
- When tapped, the keyboard automatically slides up from the bottom of the screen.
- UITextFieldDelegate protocol

Alerts

- Alerts are primarily used to inform the user of something important or verify a destructive action.
- Alerts interrupt the user experience so use them sparingly.
- UIAlertController class
- preferredStyle: UIAlertControllerStyle.Alert

Action Sheets



- Action sheets display a list of 2 or more choices to the user when a toolbar button is tapped.
- Users are unable to continue until they chose one of the choices
- UIAlertController class
- preferredStyle: UIAlertControllerStyle.ActionSheet

Alert Controllers

- To present an alert or action sheet
  1. Create a UIAlertController object with the title, message and preferredStyle you want
  2. Define your UIAlertAction objects
  3. Add your UIAlertAction objects to your alert or action sheet object
  4. Call presentViewController() to present your alert or action sheet