CS 329E
Elements of Mobile Computing

Fall 2017
University of Texas at Austin

Lecture 8

Agenda

- Alert Views
- Custom Table View Cells
- Homework 4

Alert Views

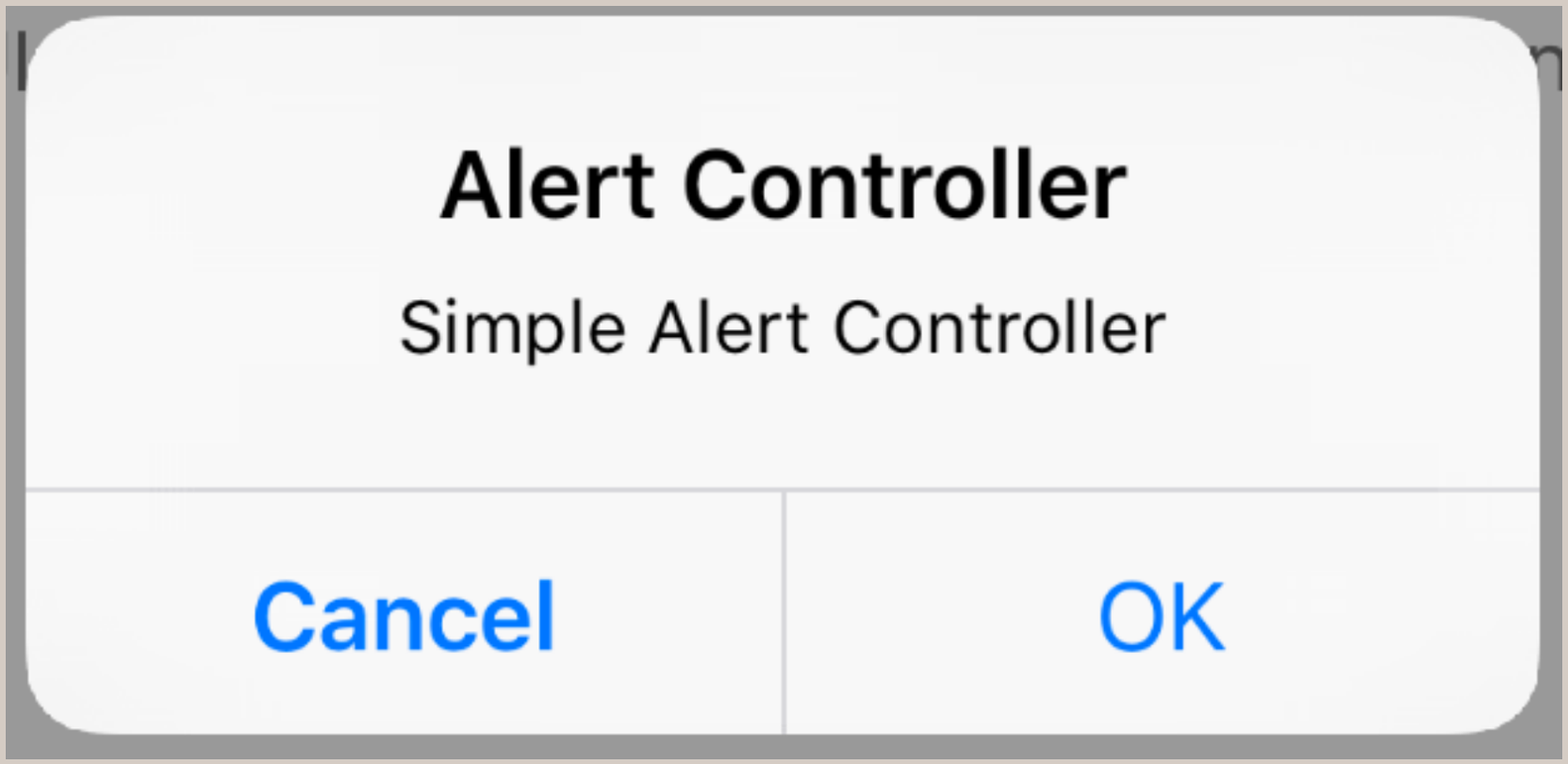
Alert Views

What are Alert Views?

- An easy way to display concise and informative information to the user
- *UIAlertView* and *UIActionSheet* were deprecated in iOS 8 and replaced by *UIAlertController*
- The kind of UI displayed in a UIAlertController (Alert, Action Sheet) is specified by the controller's *preferred style* when creating the controller
- You customize the UI by identifying what buttons or text fields you want to include

Alert Views

A simple alert:



Alert Views

Primary classes:

- *UIAlertController*
  - Displays an alert message to the user
- *UIAlertAction*
  - Represents an action that can be taken when tapping a button in an alert

You create a UIAlertController object first, then add as many UIAlertAction objects as necessary/desired; typically based on the number of buttons you want.

Alert Views

UIAlertAction object:

- A UIAlertAction object represents an action that can be taken when tapping a button in an alert
- You use this class to configure information about a single action:
  - Title to display in the button
  - Styling information
  - Handler to execute when the user taps the button
- After creating an alert action object, add it to a UIAlertController object before displaying the corresponding alert to the user

Alert Views

UIAlert*Controller*Style settings:

- The kind of alert you can create/display:
  - *Alert*
    - UI that displays over and grays out the current UI
  - *Action Sheet*
    - UI that slides up from the bottom of the screen and grays out the current UI

Alert Views

UIAlert*Action*Style settings:

- Defines the visual presentation of a button
  - *Default*
    - Apply the default style to the action's button
    - Normal text
  - *Cancel*
    - Apply a style that indicates the action cancels the operation and leaves things unchanged
    - Bolded text
    - Can only have <u>one</u> of these
      - App crashes if you define more than one
  - *Destructive*
    - Apply a style that indicates the action might change or delete data
    - Text color is red

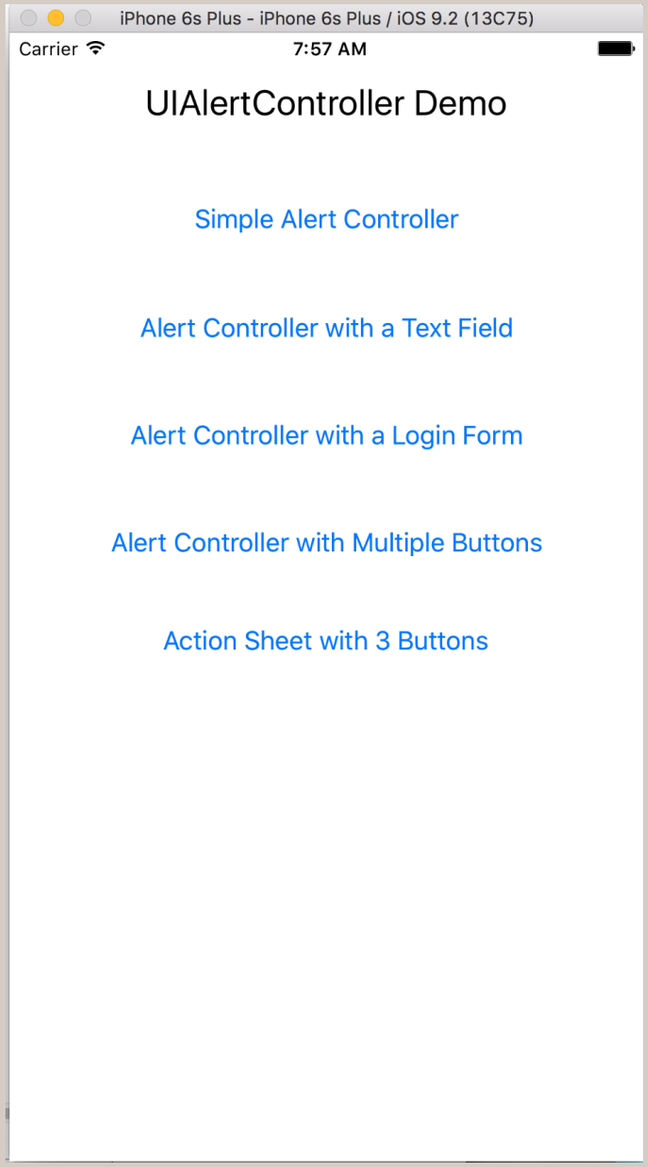
Alert Views

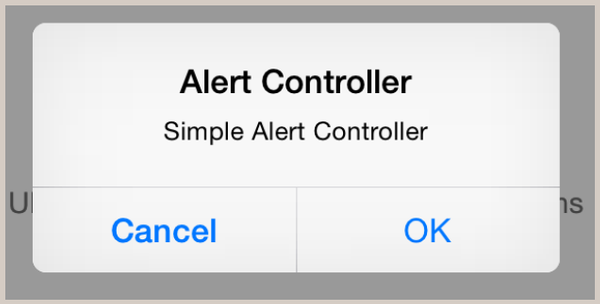
Steps to create and use an alert view controller:

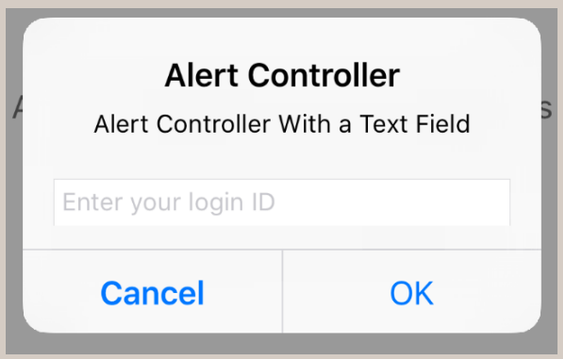
- Create a UIAlertController object
- Create some number of UIAlertAction objects and add them to the UIAlertController object
    - Each action object has a code block that defines the code to execute when the user selects the related button
- Present the UIAlertController

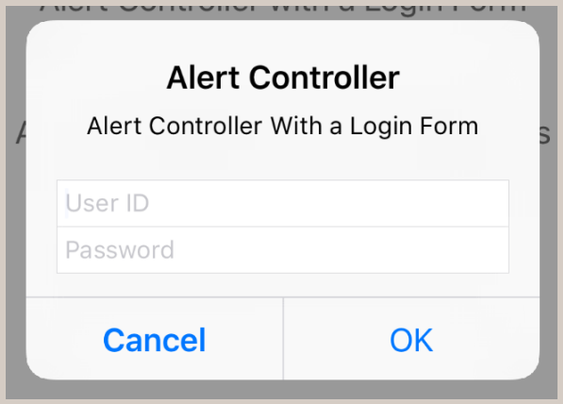
Alert Views

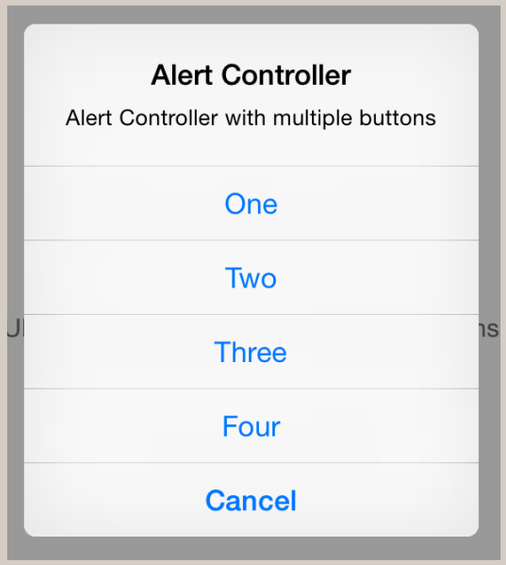
Demo - TestAlertController:

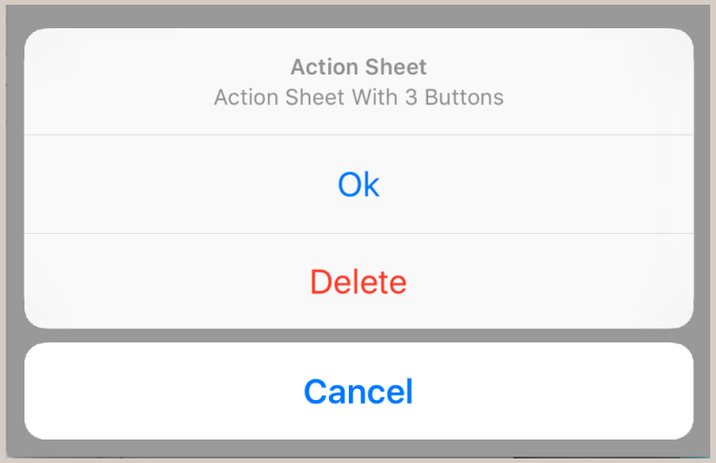








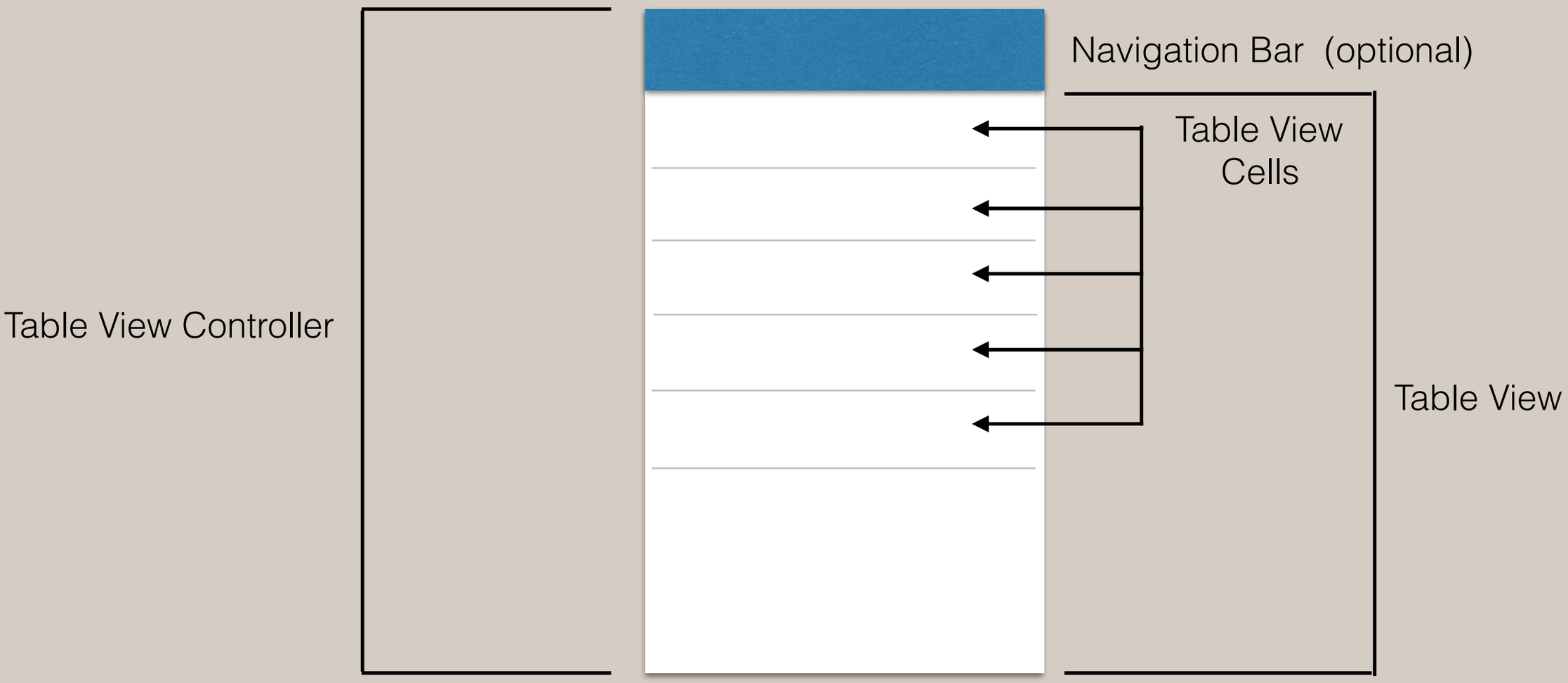




Custom Table View Cells

Custom Table View Cells

- A UITableViewController contains a UITableView.
- A UITableView contains *at least one* UITableViewCell.
- A UITableViewCell contains UI elements - Labels, Buttons, etc.



Custom Table View Cells

What is a custom table view cell?

- A custom table view cell is a table view cell that is <u>not</u> one of the pre-defined table view cell types

The pre-defined table view cell types are:

Basic

Right Detail

Left Detail

Subtitle

Custom Table View Cells

What kind of custom table view cell can you create?

- Any kind - simple to complex.
  - Which means pretty much every UI element is in play within the cell.
  - That said, you want to be reasonable since screen real estate is at a premium, even with ‘Plus’ phones.
    - Less of an issue if your target devices are iPads.
- If you have ‘a lot’ (subjective) in a table cell, you should consider creating a ‘detail’ view controller to navigate to when the user touches the table cell.

Custom Table View Cells

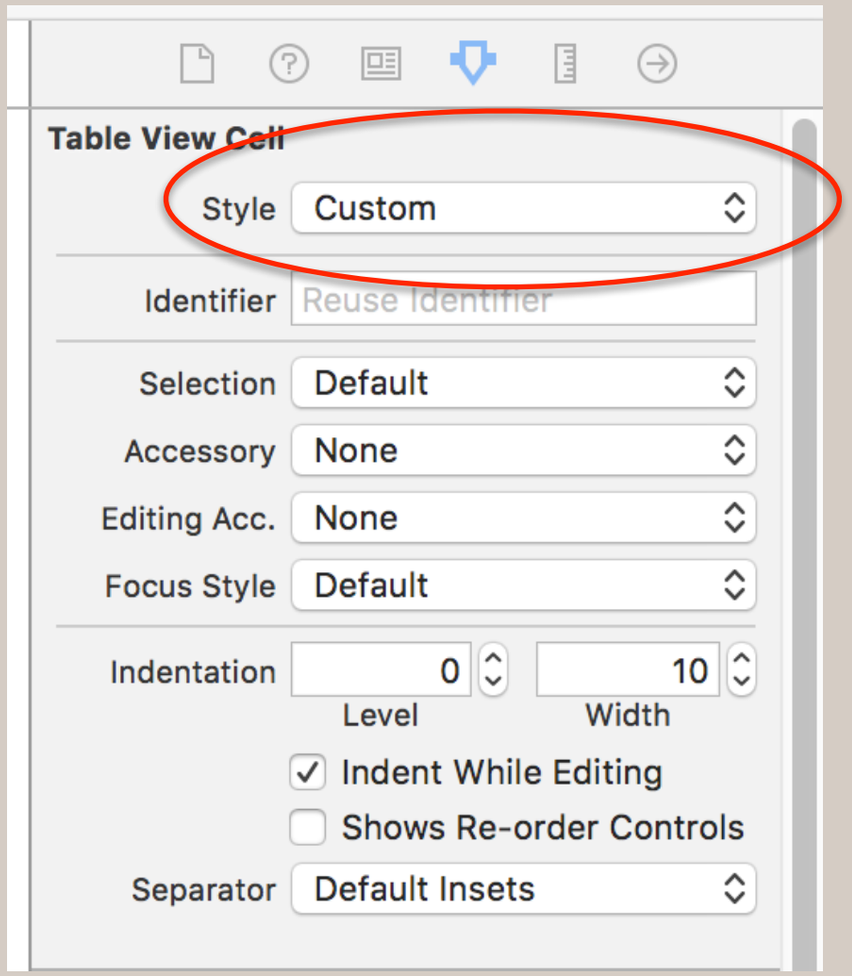
How do you define a custom table view cell?

- Select *Custom* as the table view cell style.
- Create a UITableViewCell-derived class.
- Associate the class you just created with the prototype cell in the storyboard.
- Drag-and-drop UI elements into the table view cell prototype.
- Create outlets for the UI elements in the prototype cell, in the UITableViewCell-derived class.
- Write code to make use of the UI elements in your code.

Custom Table View Cells

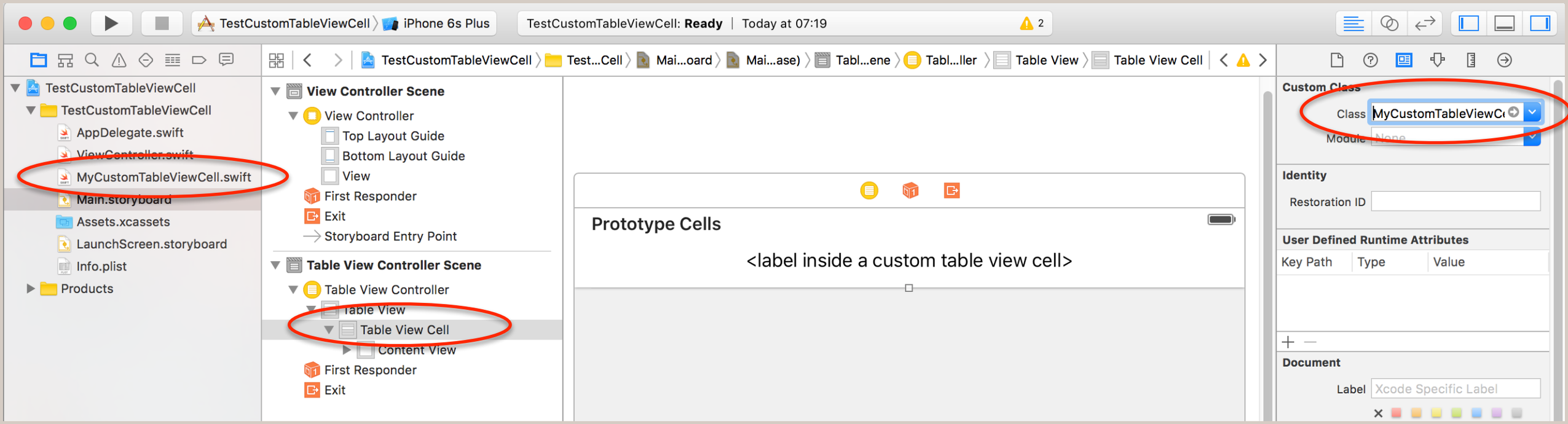
Select the Custom table view cell *style*.

- Which is the default anyway.



Custom Table View Cells

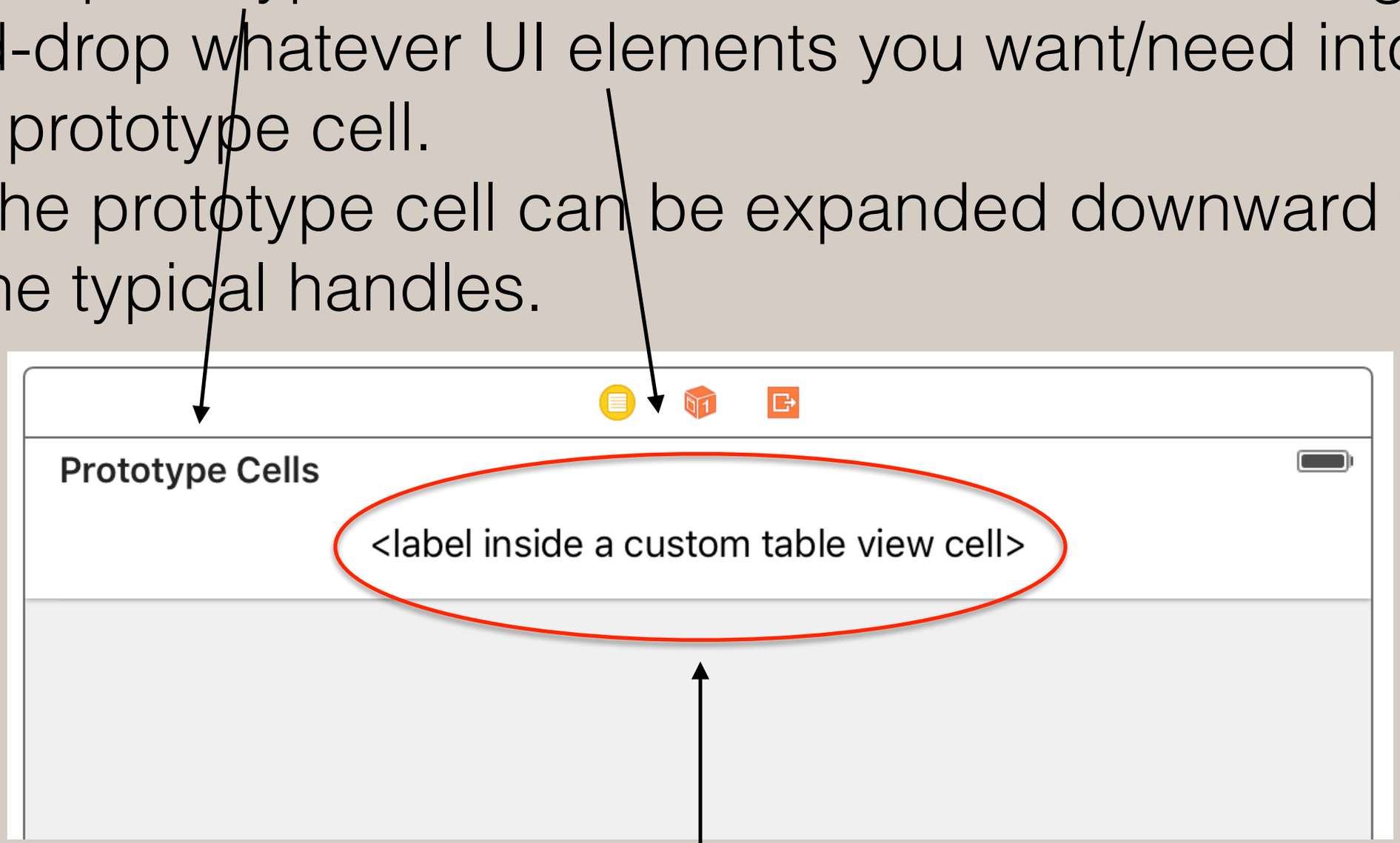
Associate your custom table view cell class with the prototype cell.



Custom Table View Cells

In the prototype cell of a table view controller, drag-and-drop whatever UI elements you want/need into the prototype cell.

- The prototype cell can be expanded downward via the typical handles.



In-Class Exercise

In-Class Exercise

Create an application with 2 custom table view cells.

We will alternate the custom table cells in the table view.

Homework 4

Homework 4

- Create an application that uses:
  - Navigation Controller
  - Table View Controller
  - Custom Table View Cells - 2
  - Alert Controller