Academy for iOS App Development

Dr. William C. Bulko

# Views and View Controllers

## Views

- *Views* are all of the individual elements that make up your user interface.
  - Buttons
  - Labels
  - Text Fields
  - etc.
- Views can contain other views.
  - Every view controller has a base view where other views can be added as subviews, creating a *view hierarchy*
- Any view can be hidden.  If it is, all of its subviews are also hidden.

## View Controllers

- *View Controllers* (VCs) are the objects in your iOS application that contain the code that coordinates communication between the data and view components.
- All VCs derive from the UIViewController class.
- All iOS applications have <u>at</u> <u>least</u> one VC, and <u>at</u> <u>least</u> one (but typically only one) window.

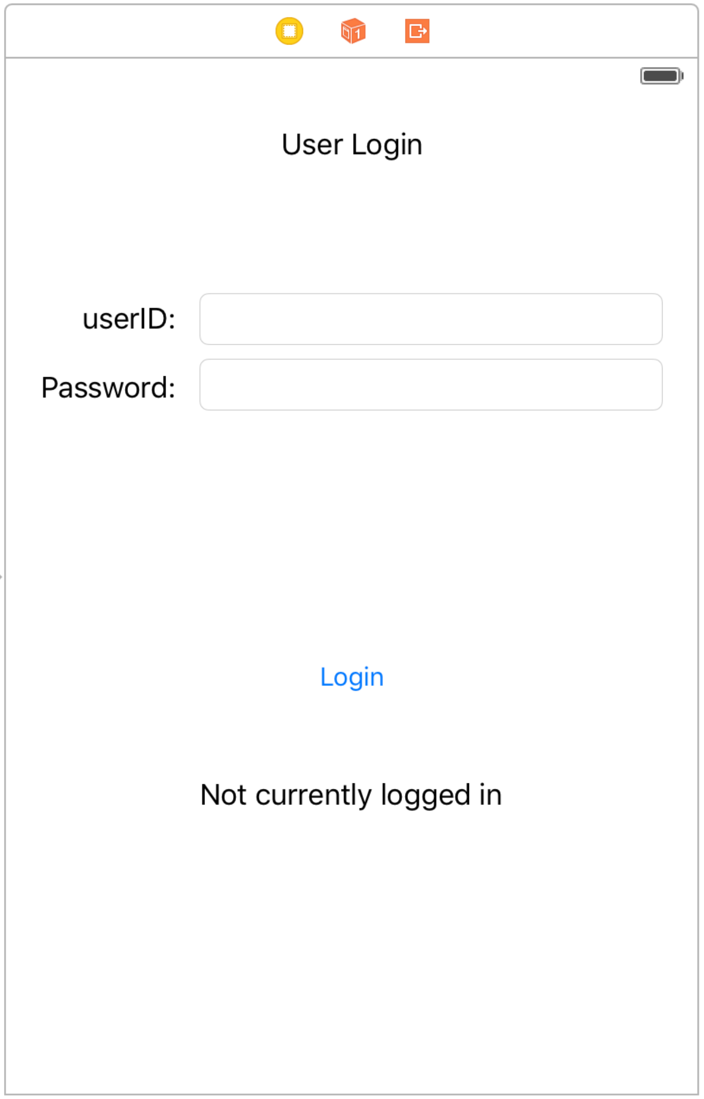
## Two General Categories of VCs

- Content VCs:  present content on the screen using a view or a group of views organization into a *view hierarchy*.
  - Ex. UIViewController
- Container VCs:
  - contain content owned by other VCs that are explicitly assigned to the container VC as its children
  - can be both a parent to other controllers and a child of another container
  - Ultimately, this combination of controllers establishes a *view controller hierarchy*.
  - Ex. UINavigationController

## Basic View Controller

Used to display any combination of views (widgets):

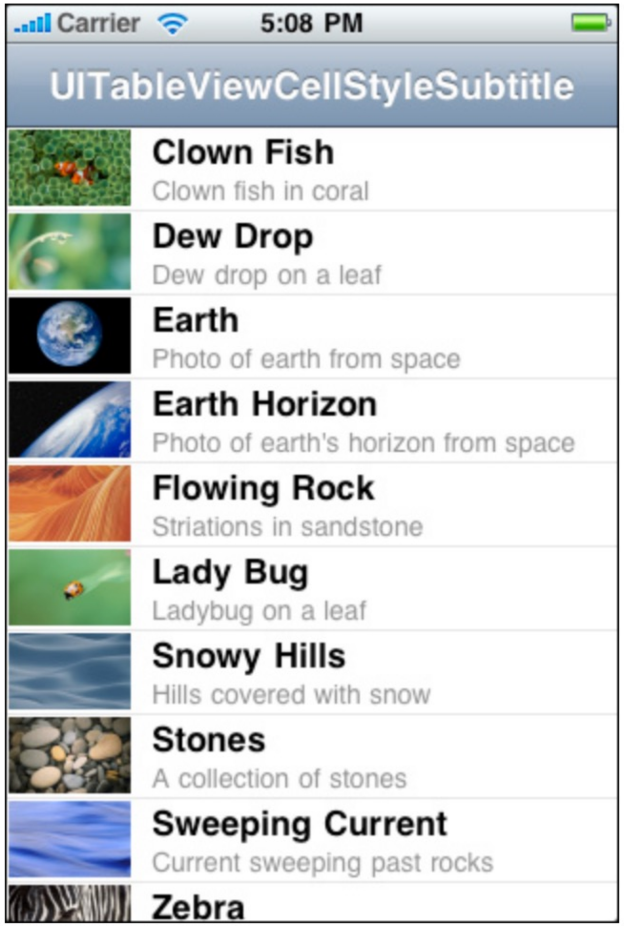
- Labels
- Buttons
- Text Fields
- etc.



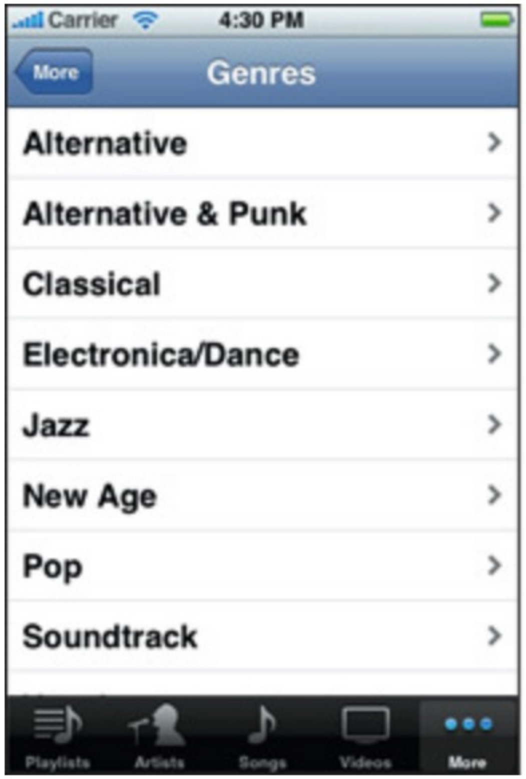
## Table View Controller

Used to display a list of things in tabular form.

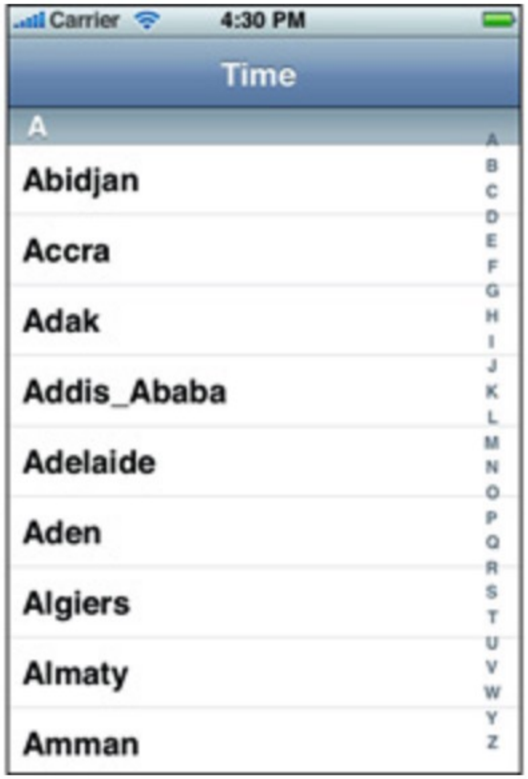
- Each item in the list is called a *table cell.*
- The default is to for all cells to have the same layout:  a single title and an optional image. However, you can write code to customize any cell.
- The view auto scrolls.
- There are many different flavors of Table VCs.



## Table View Controller

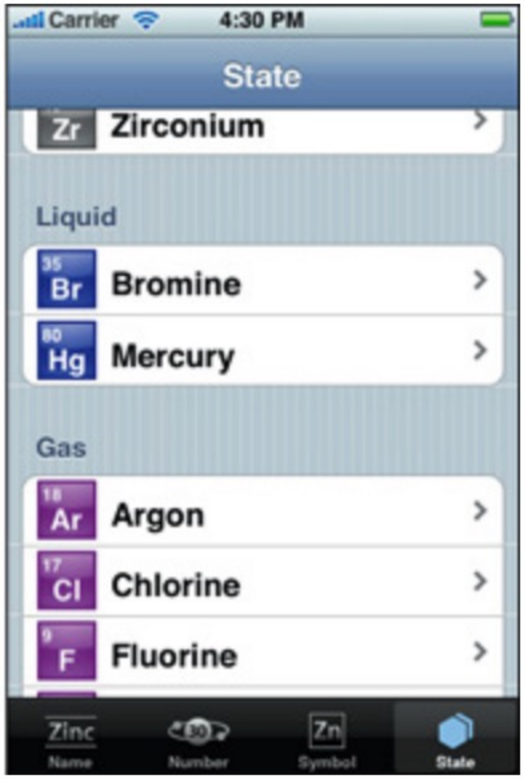


allows users to navigate through a hierarchical structure or to provide details

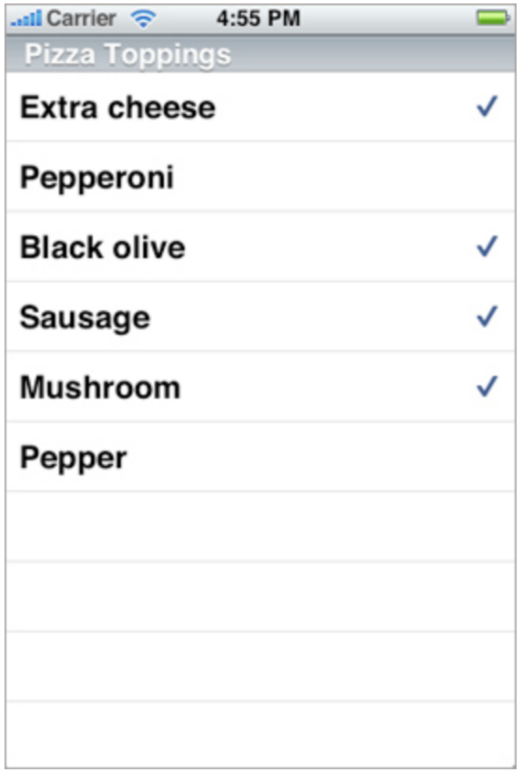


to present an indexed list of items

# Table View Controller



to display information in a visually distinct grouping

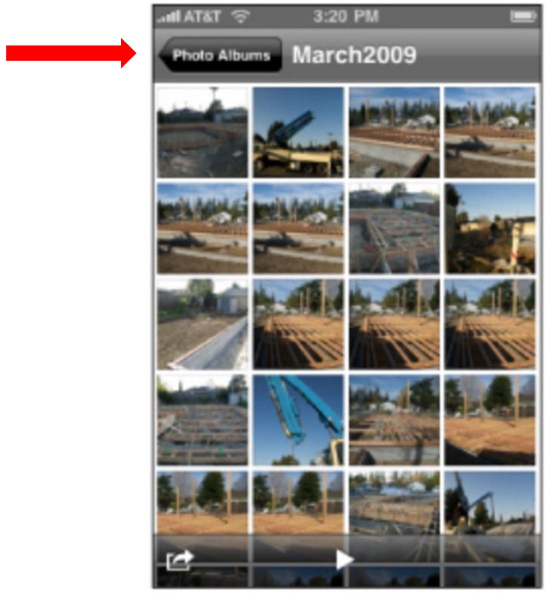


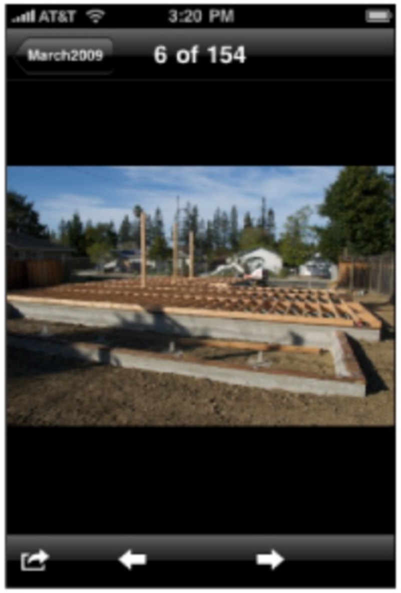
to allow selection of one or more entries in the table

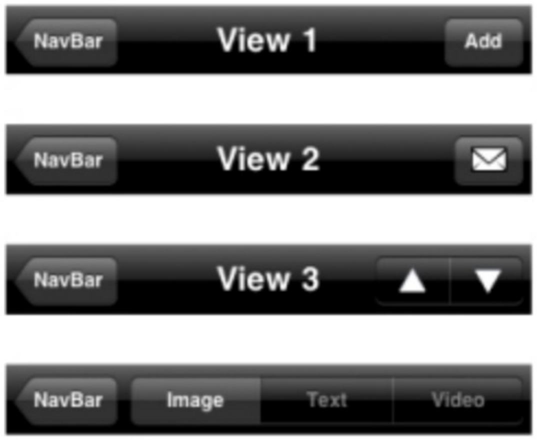
## Navigation View Controller

Used to coordinate navigating between one VC and another.

- Uses a portion of the top of the screen, where it provides an area for two optional buttons (top left and top right) and some text in the middle.





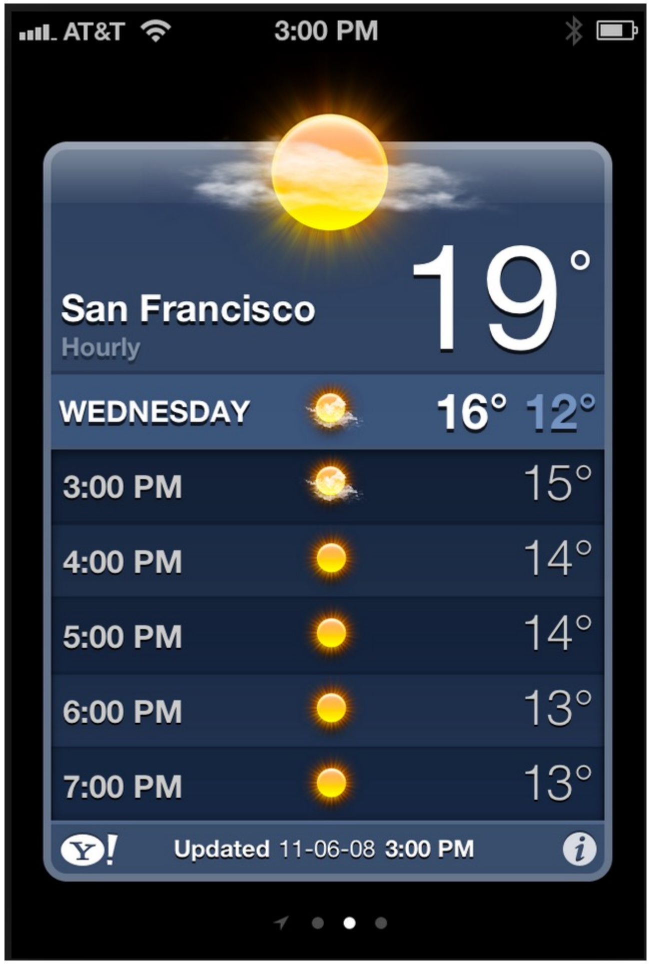


Uses a portion of the top of the screen, where it provides an area for two optional buttons (top left and top right) and some text in the middle.

As with most things, customizable

## Page View Controller

- Has a single view to display your content
- Provides a UI to simulate turning a page
- Has a row of dots that represent pages the user can navigate to
- Navigate via left or right swipes



San Francisco 19°
Hourly

| | |
|---|---|
| WEDNESDAY | 16° 12° |
| 3:00 PM | 15° |
| 4:00 PM | 14° |
| 5:00 PM | 14° |
| 6:00 PM | 13° |
| 7:00 PM | 13° |

Updated 11-06-08 3:00 PM

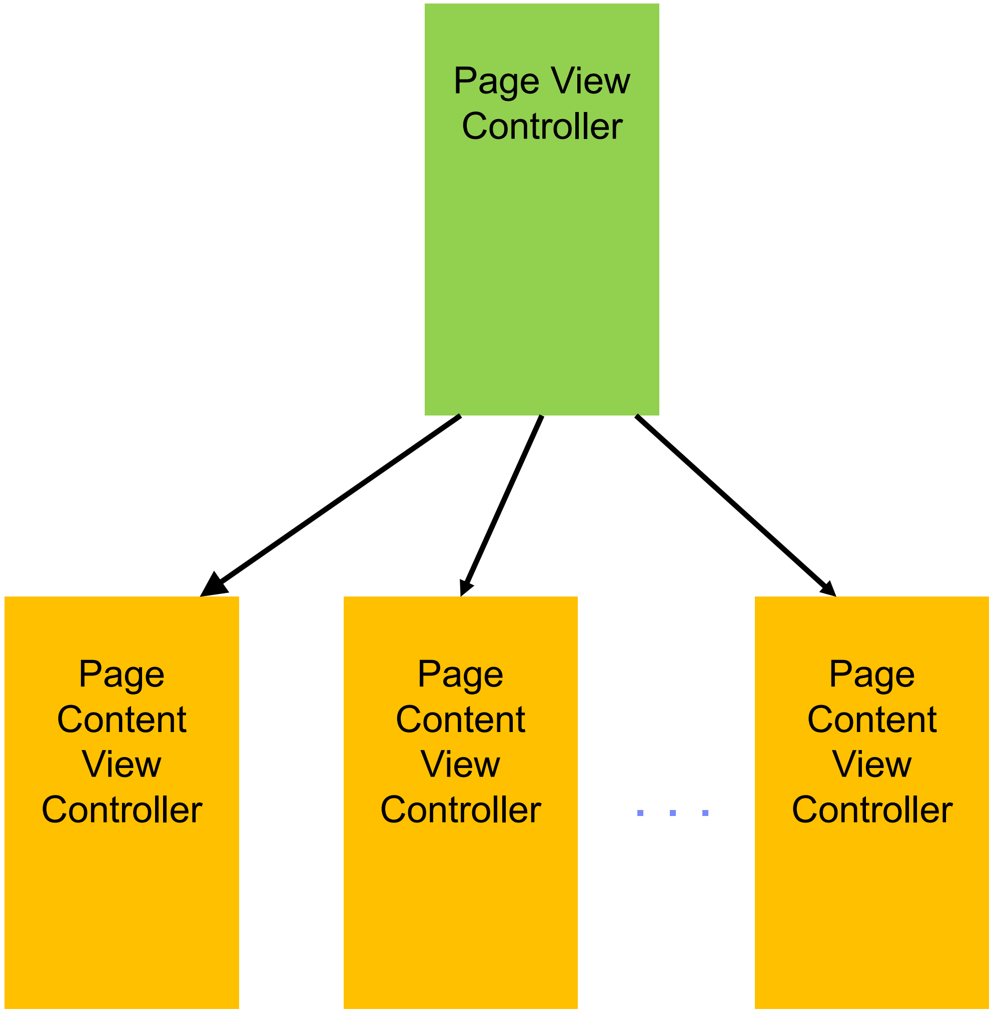
## Page View Controllers

- You can define:
  - The orientation of the page views:  vertical or horizontal
  - The transition style from page to page:  curl or scrolling
  - The location of the spine (only applicable to curl transitions)
  - The space between pages (only applicable to scrolling transitions) to define the inter-page spacing

## Page View Controllers (cont.)

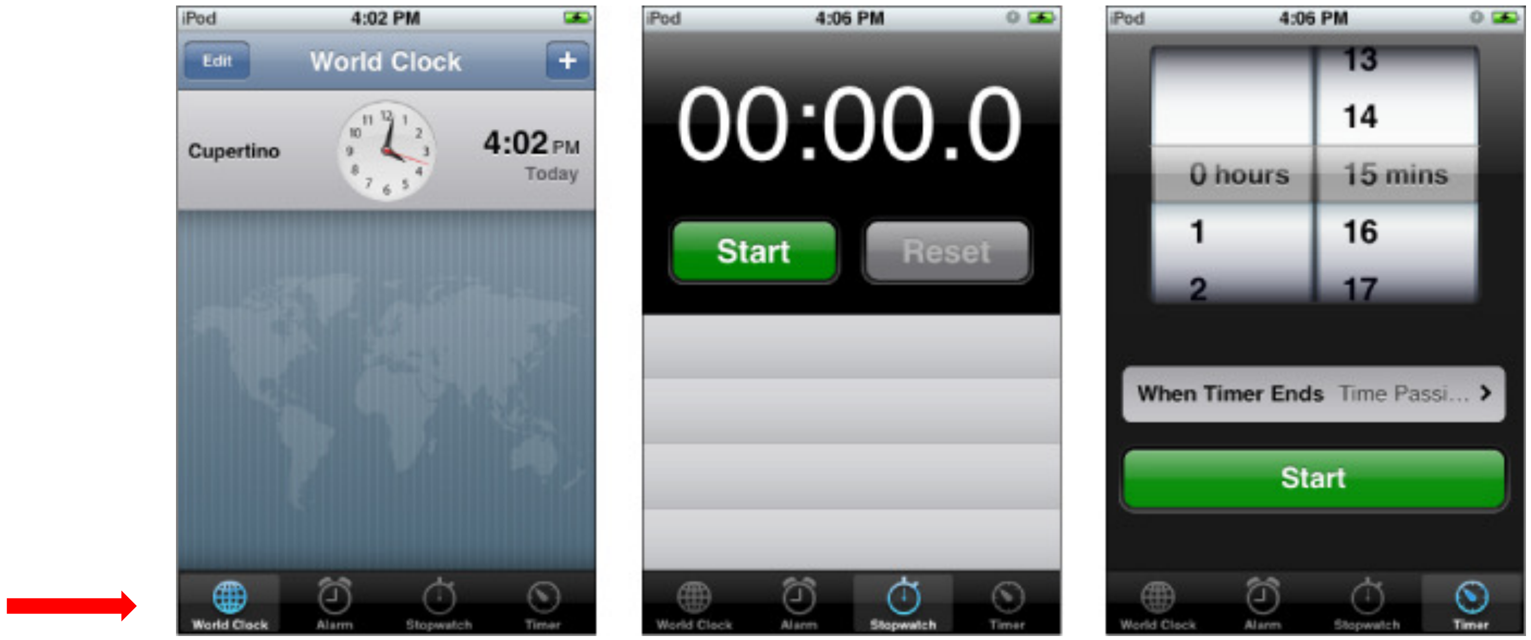
Classified as a *container* controller.

- Used to contain and manage multiple VCs shown in the app, as well as controlling the way one VC switches to another
- Each page is managed by its own VC


Page View Controller
Page Content View Controller
Page Content View Controller
. . .
Page Content View Controller

## Tab View Controller

- Has a row of tabs at the bottom of the screen
- Each button navigates to a different VC



Academy for iOS App Development

Dr. William C. Bulko

# Delegates and Segues

## Delegates

A *delegate* is a simple but powerful pattern in which one object acts on behalf of or in coordination with another object.

- The delegating object keeps a reference to another object (the delegate) and at the appropriate time, sends a message to it.  The main value of delegation is that it allows you to easily customize the behavior of several objects in one central object.
- There’s nothing that says you can’t have more than one delegate.

## Segue

A *segue* is a named transition between one part of the UI to another.  Its purpose is to make it easier to move from one VC to another.

- A segue is created in IB and code is written to make use of it.