CS193P - Lecture 7

iPhone Application Development

Navigation & Tab Bar Controllers

Announcements

- Assignment 3 was due last night at 11:59 PM
- Presence 1 is due on Tuesday 4/28

Announcements

- Next Monday, 4/27
  - Table Views, Scroll Views and Presence 2
  - Guest speaker: Jason Beaver, UIKit Engineer

Announcements

- This Friday: “Preparing Your App for the App Store”
- Next Friday: Loren Brichter of Tweetie (http://www.atebits.com)

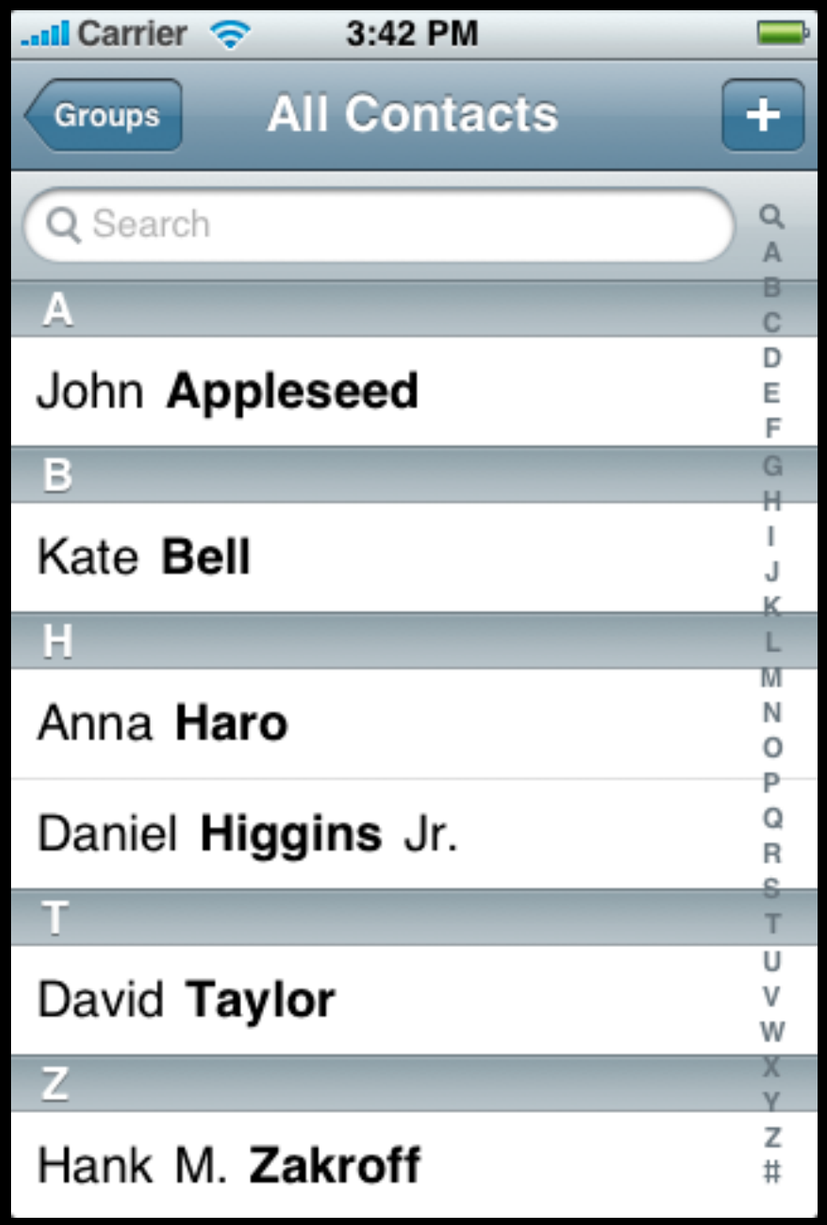
Today's Topics

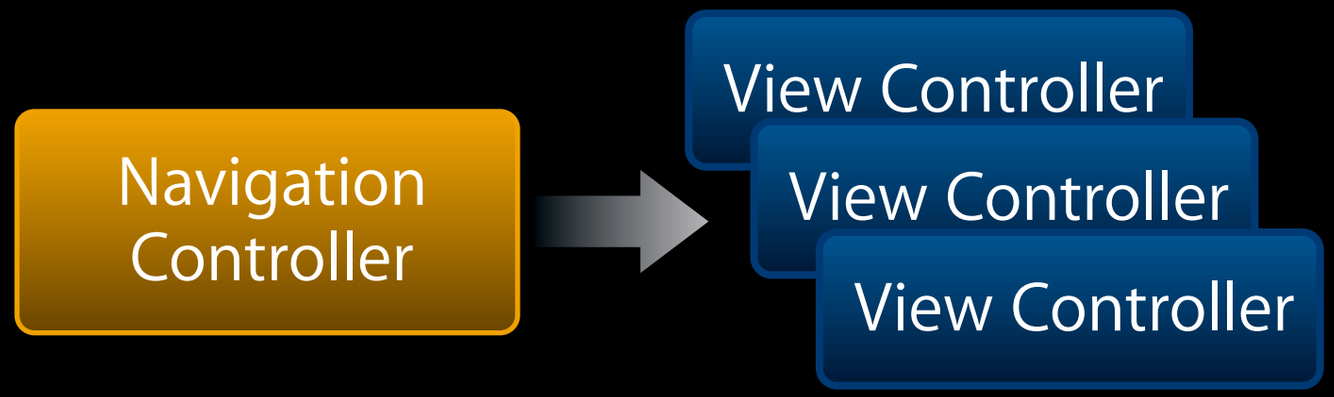
- Navigation Controllers
- Application Data Flow
- Customizing Navigation
- Tab Bar Controllers
- Combining Approaches

Navigation Controllers

UINavigationController

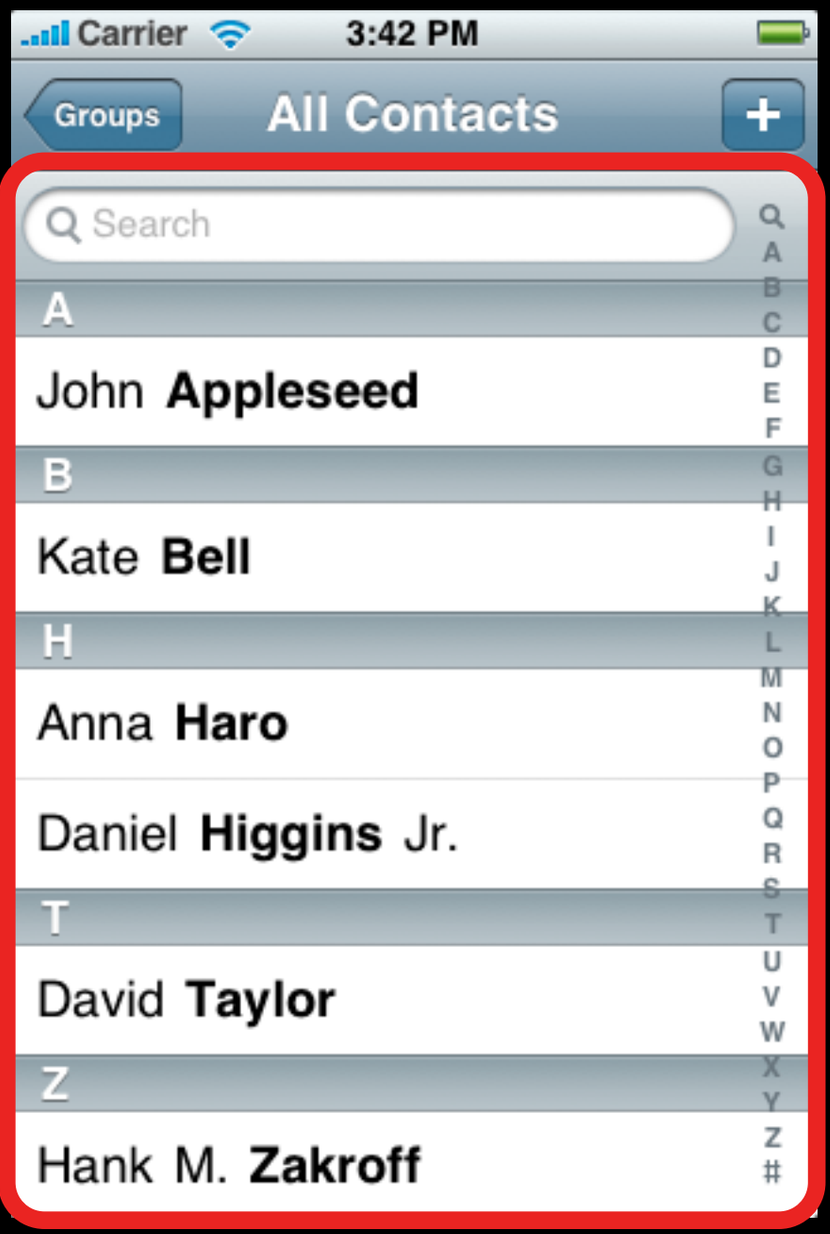
- Stack of view controllers
- Navigation bar





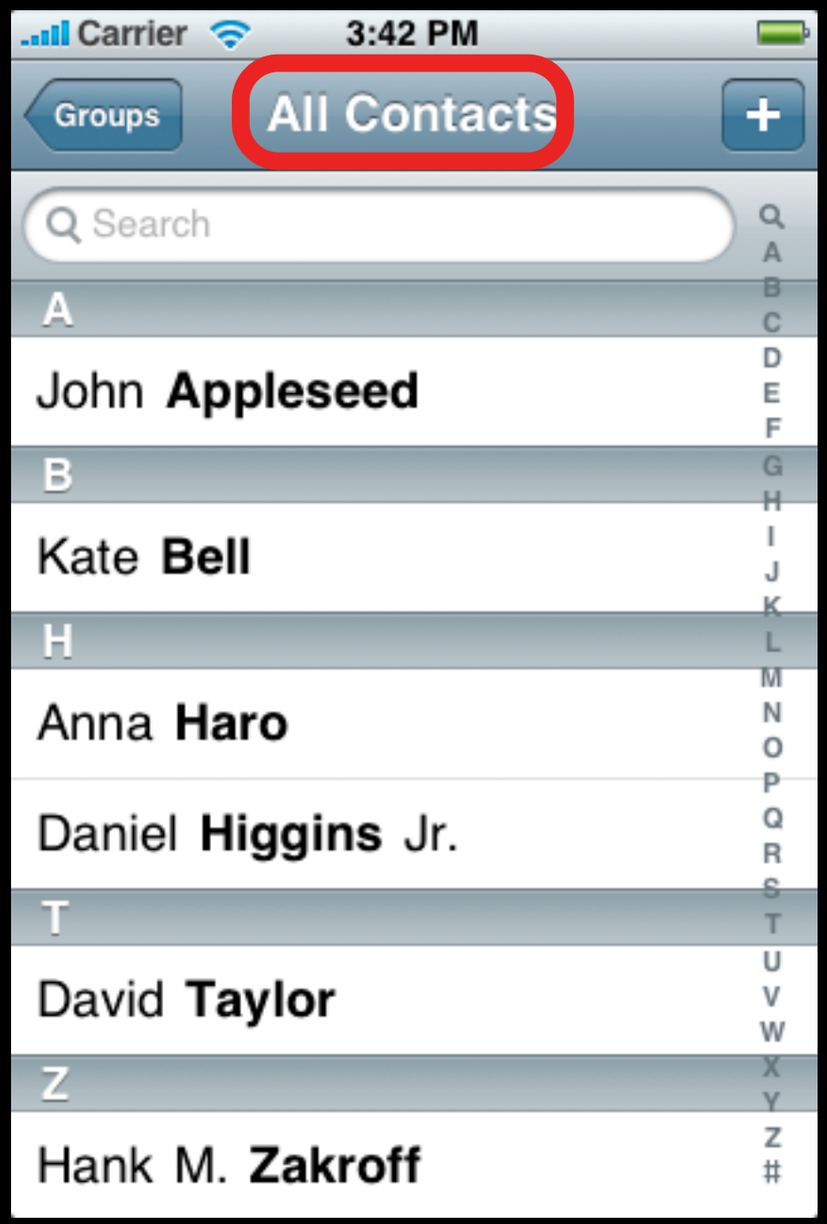
How It Fits Together

- Top view controller's view

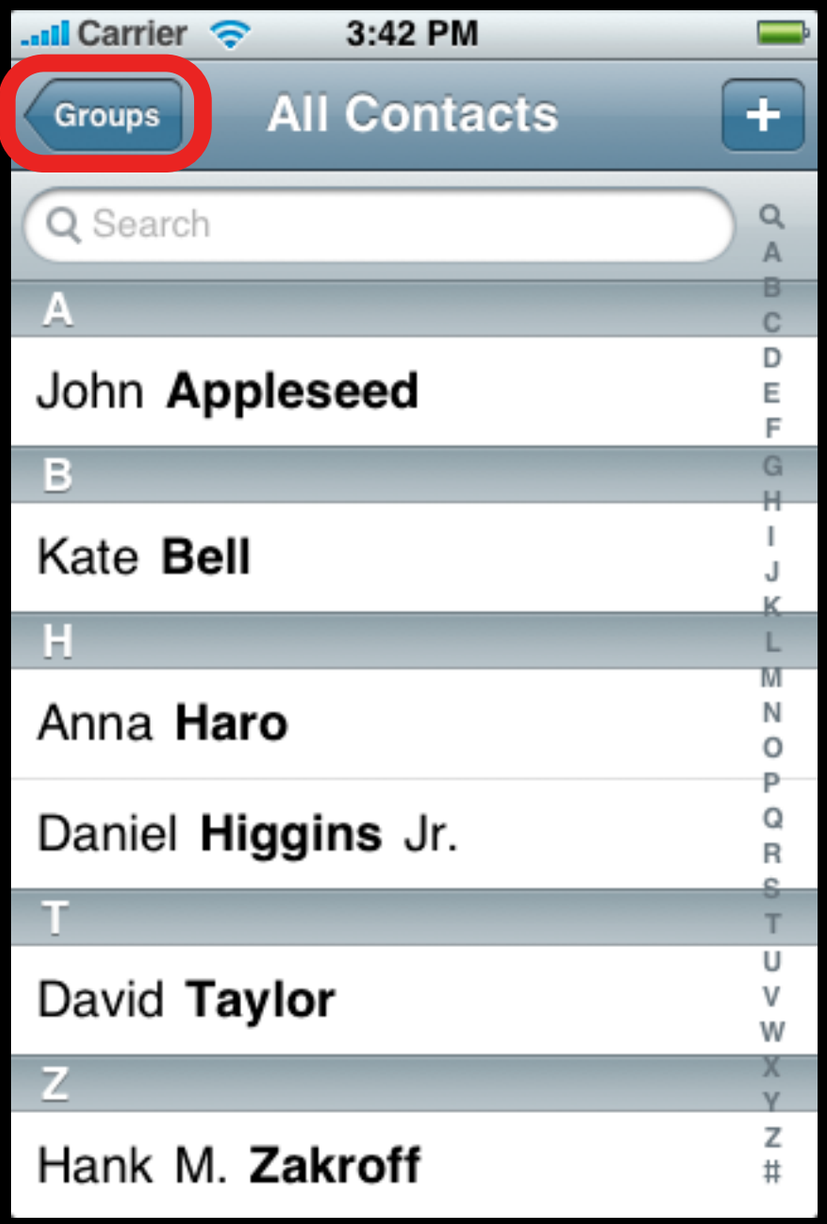


How It Fits Together

- Top view controller's view
- Top view controller's title



How It Fits Together



- Top view controller's view
- Top view controller's title
- Previous view controller's title

Modifying the Navigation Stack

- **Push** to add a view controller

```
- (void)pushViewController:(UIViewController *)viewController
                  animated:(BOOL)animated;
```

- **Pop** to remove a view controller

```
- (void)popViewControllerAnimated:(BOOL)animated;
```

Pushing Your First View Controller

```
- (void)applicationDidFinishLaunching
  // Create a navigation controller
  navController = [[UINavigationController alloc] init];

  // Push the first view controller on the stack
  [navController pushViewController:firstViewController
                           animated:NO];

  // Add the navigation controller's view to the window
  [window addSubview:navController.view];
}
```

In Response to User Actions

- Push from within a view controller on the stack

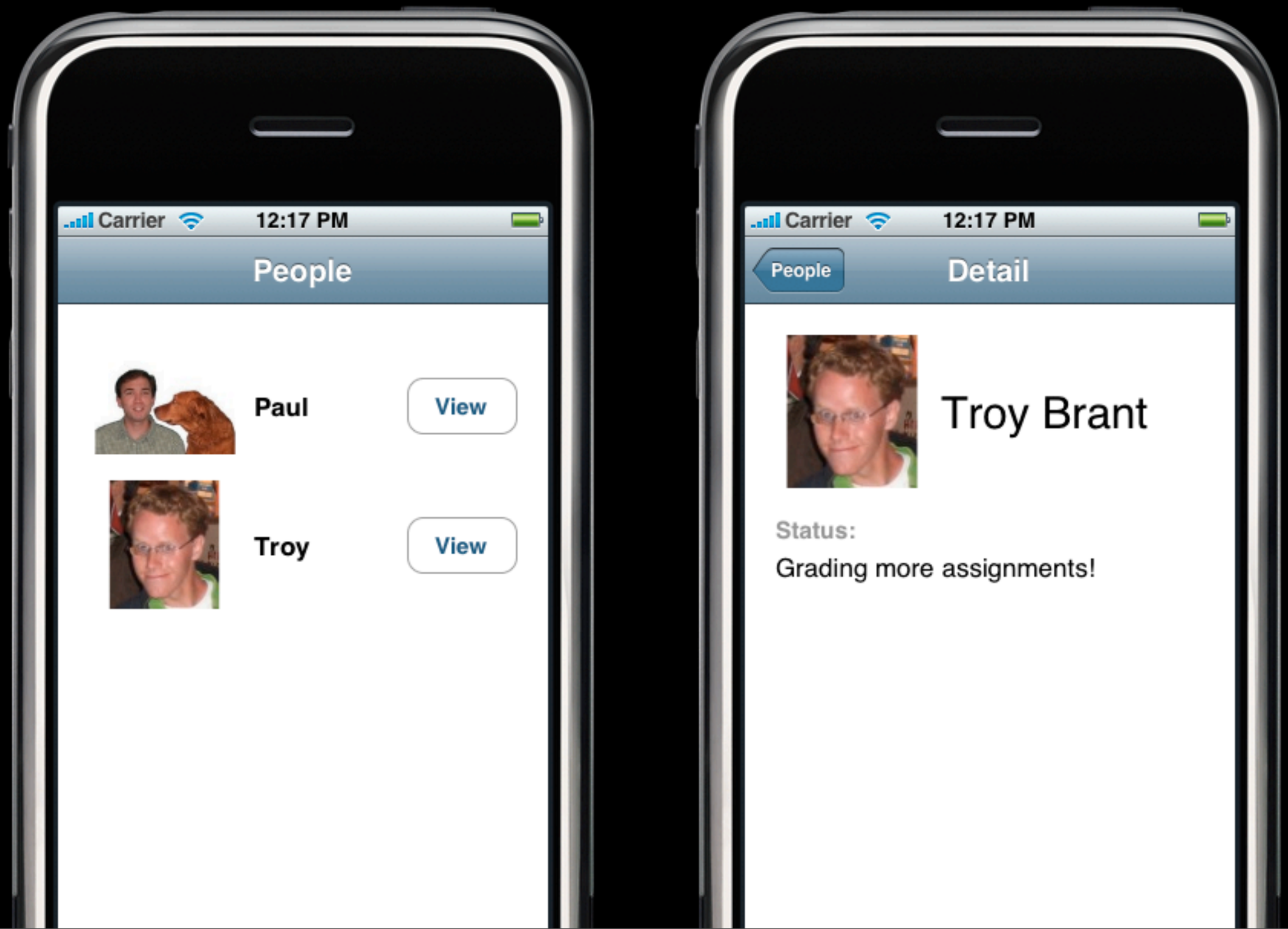
```
- (void)someAction:(id)sender
{
    // Potentially create another view controller
    UIViewController *viewController = ...;

    [self.navigationController pushViewController:viewController
                                         animated:YES];
}
```

- Almost **never** call pop directly!
  - Automatically invoked by the back button

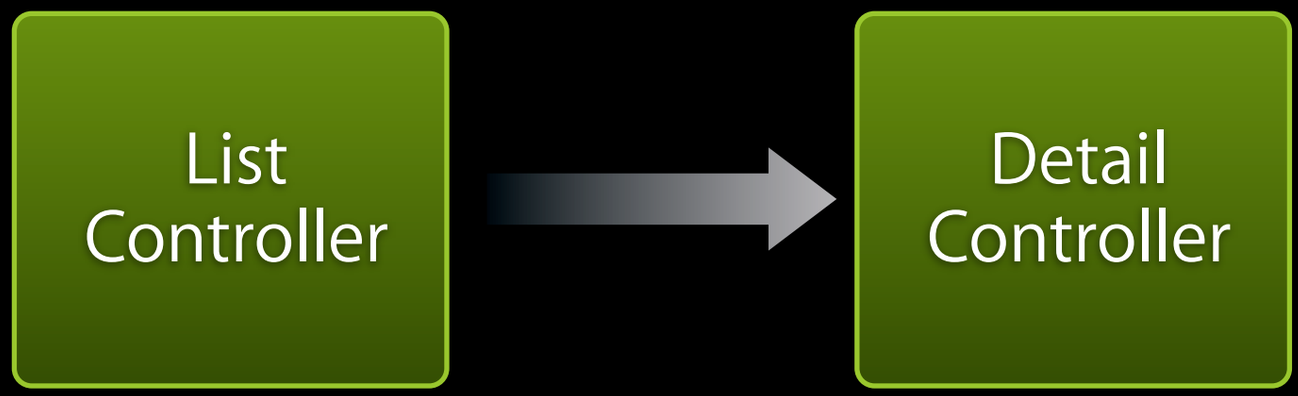
Demo:
Pushing & Popping

Application Data Flow

Presence


Carrier
12:17 PM
People
Paul
View
Troy
View
Carrier
12:17 PM
People
Detail
Troy Brant
Status:
Grading more assignments!

A Controller for Each Screen

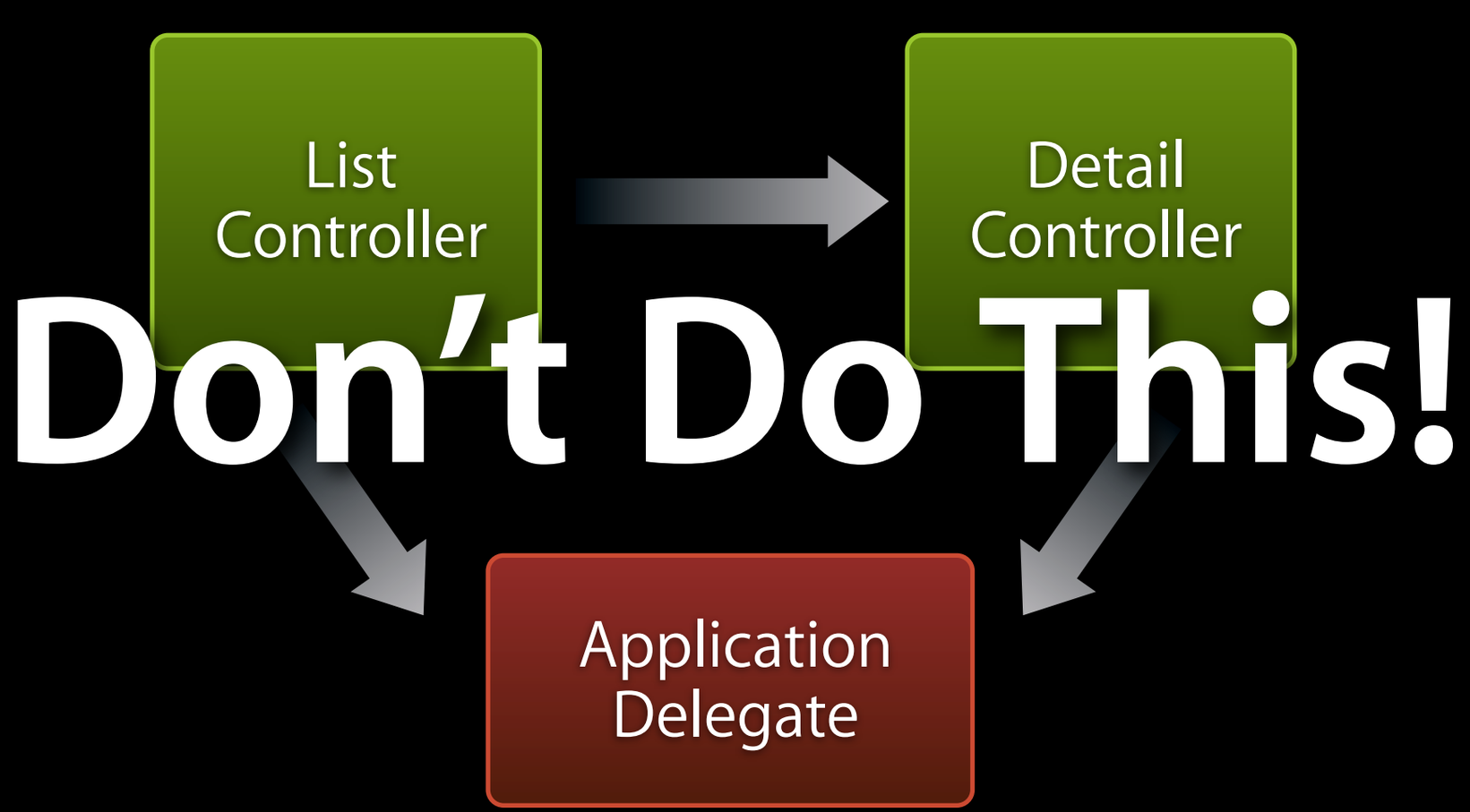


Connecting View Controllers

- Multiple view controllers may need to **share data**
- One may need to know about what another is doing
  - Watch for added, removed or edited data
  - Other interesting events

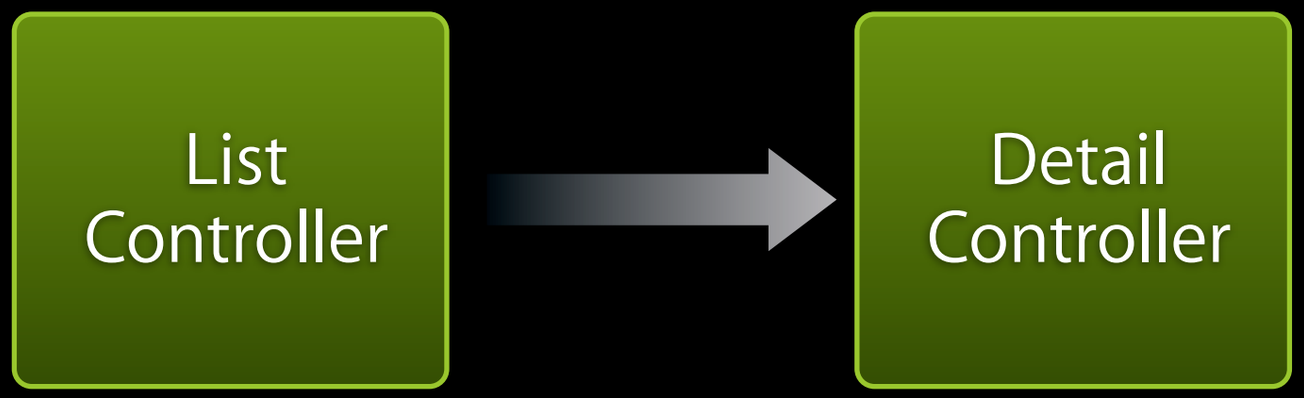
How Not To Share Data

- Global variables or singletons
  - This includes your **application delegate**!
- Direct dependencies make your code less reusable
  - And more difficult to debug & test



Best Practices for Data Flow

- Figure out **exactly** what needs to be communicated
- **Define input parameters** for your view controller
- For communicating back up the hierarchy, **use loose coupling**
  - Define a generic interface for observers (like delegation)



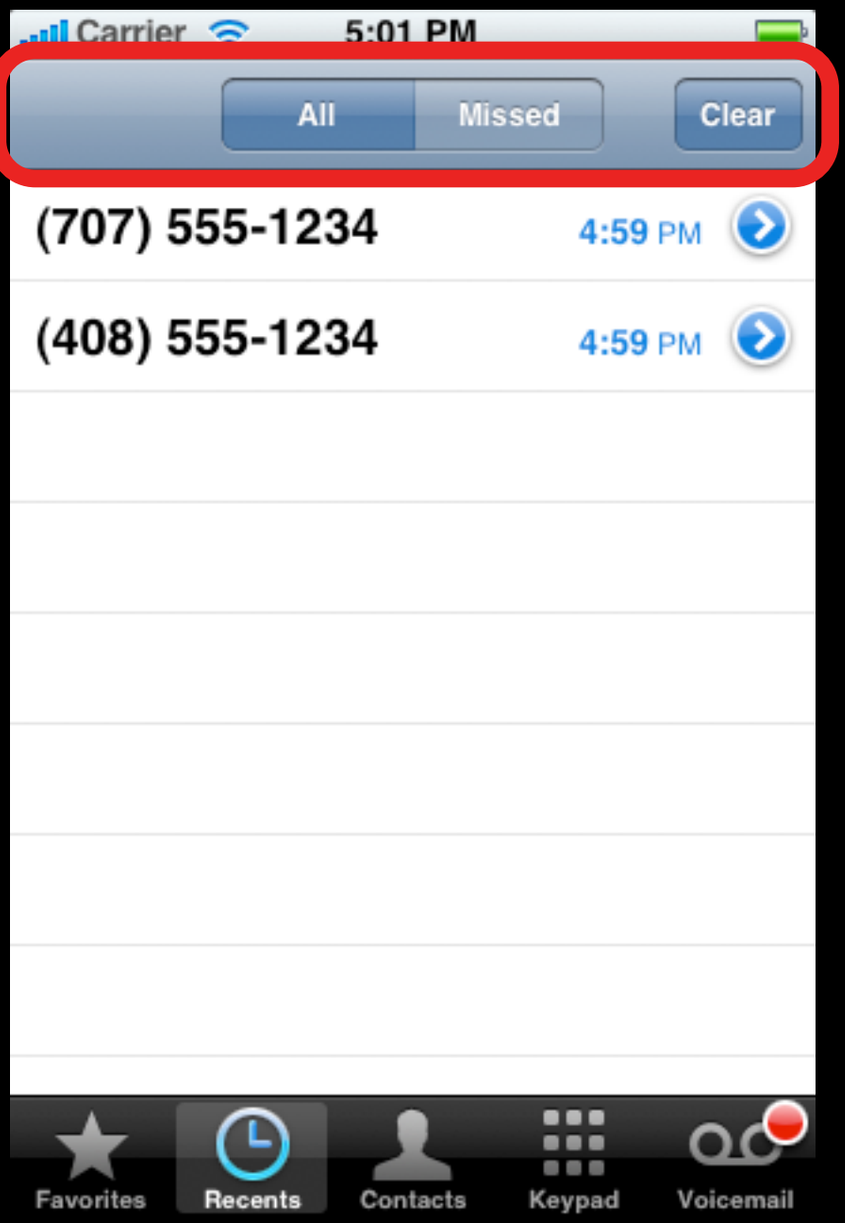
Example: UIImagePickerController

Demo:
Passing Data Along

Customizing Navigation

Customizing Navigation

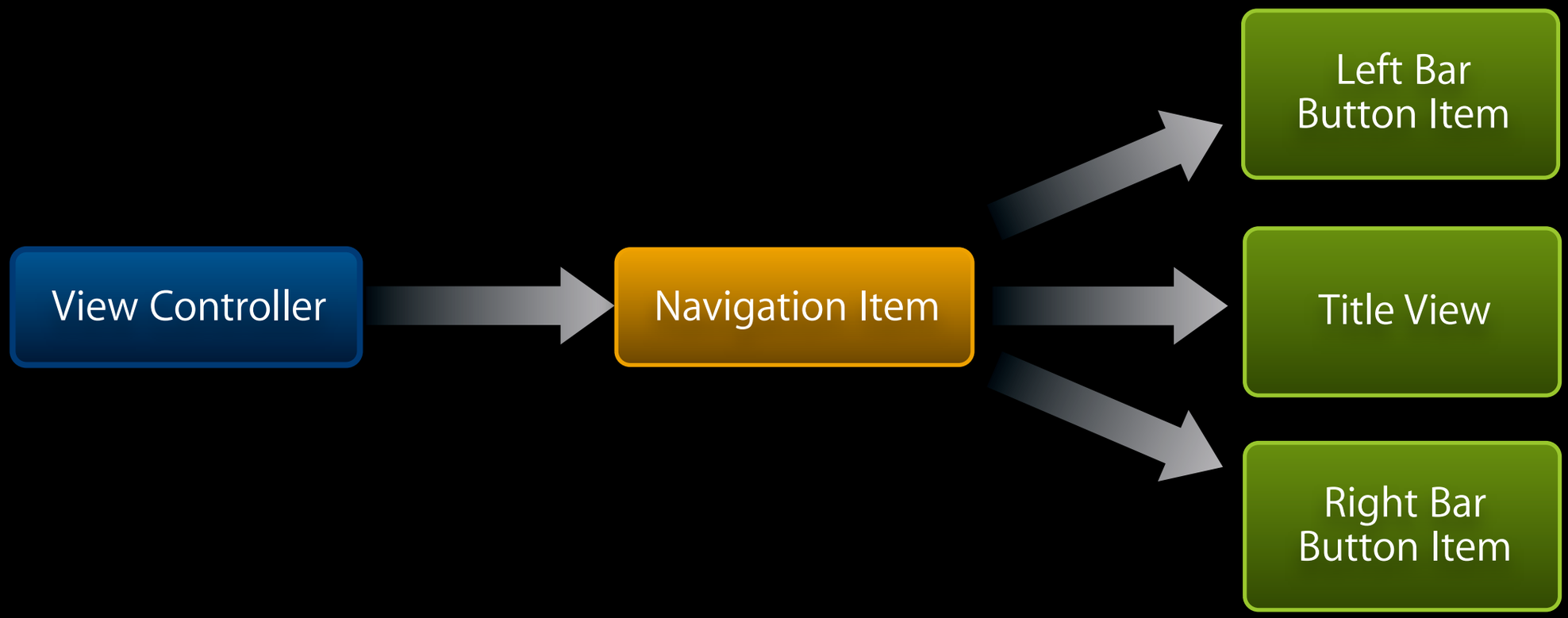
- Buttons or custom controls
- Interact with the entire screen



UINavigationItem

- Describes appearance of the navigation bar
  - Title string or custom title view
  - Left & right bar buttons
  - More properties defined in UINavigationBar.h
- **Every view controller has a navigation item** for customizing
  - Displayed when view controller is on **top of the stack**

Navigation Item Ownership



Displaying a Title

- UIViewController already has a title property
  - `@property(nonatomic,copy) NSString *title;`
- Navigation item inherits automatically
  - Previous view controller's title is displayed in back button


```
viewController.title = @“Detail”;
```

Left & Right Buttons

- UIBarButtonItem
  - Special object, defines appearance & behavior for items in navigation bars and toolbars
- Display a string, image or predefined system item
- Target + action (like a regular button)

Text Bar Button Item

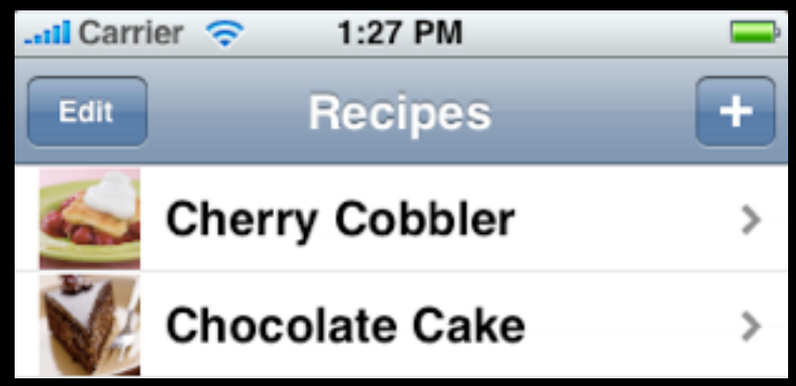

```
- (void)viewDidLoad
{
   UIBarButtonItem *fooButton = [[UIBarButtonItem alloc]
      initWithTitle:@"Foo”
      style:UIBarButtonItemStyleBordered
      target:self
      action:@selector(foo:)];

   self.navigationItem.leftBarButtonItem = fooButton;

   [fooButton release];
}
```

System Bar Button Item

```
- (void)viewDidLoad
{
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc]
        initWithBarButtonSystemItem:UIBarButtonSystemItemAdd
        style:UIBarButtonItemStyleBordered
        target:self
        action:@selector(add:)];

    self.navigationItem.rightBarButtonItem = addButton;

    [addButton release];
}
```

Edit/Done Button

- Very common pattern
- Every view controller has one available
  - Target/action already set up




```
self.navigationItem.leftBarButtonItem = self.editButtonItem;
```

```
// Called when the user toggles the edit/done button
- (void)setEditing:(BOOL)editing animated:(BOOL)animated
{
   // Update appearance of views
}
```

Custom Title View

- Arbitrary view in place of the title


```
UISegmentedControl *segmentedControl = ...
self.navigationItem.titleView = segmentedControl;
[segmentedControl release];
```

Back Button

- Sometimes a shorter back button is needed


```
self.title = @“Hello there, CS193P!”;
```

Back Button

- Sometimes a shorter back button is needed

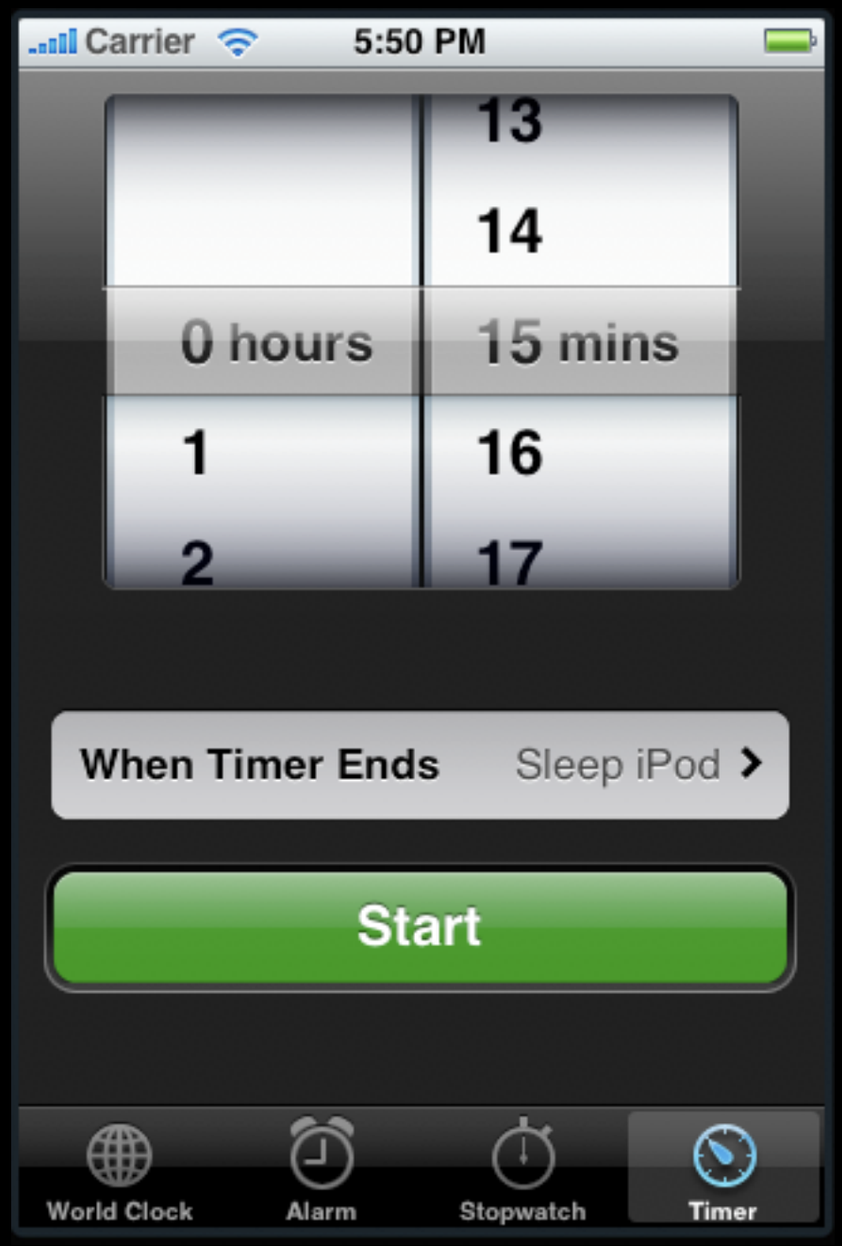

```
self.title = @“Hello there, CS193P!”;

UIBarButtonItem *heyButton = [[UIBarButtonItem alloc]
                               initWithTitle:@“Hey!”
                               ...];
self.navigationItem.backButtonItem = heyButton;

[heyButton release];
```

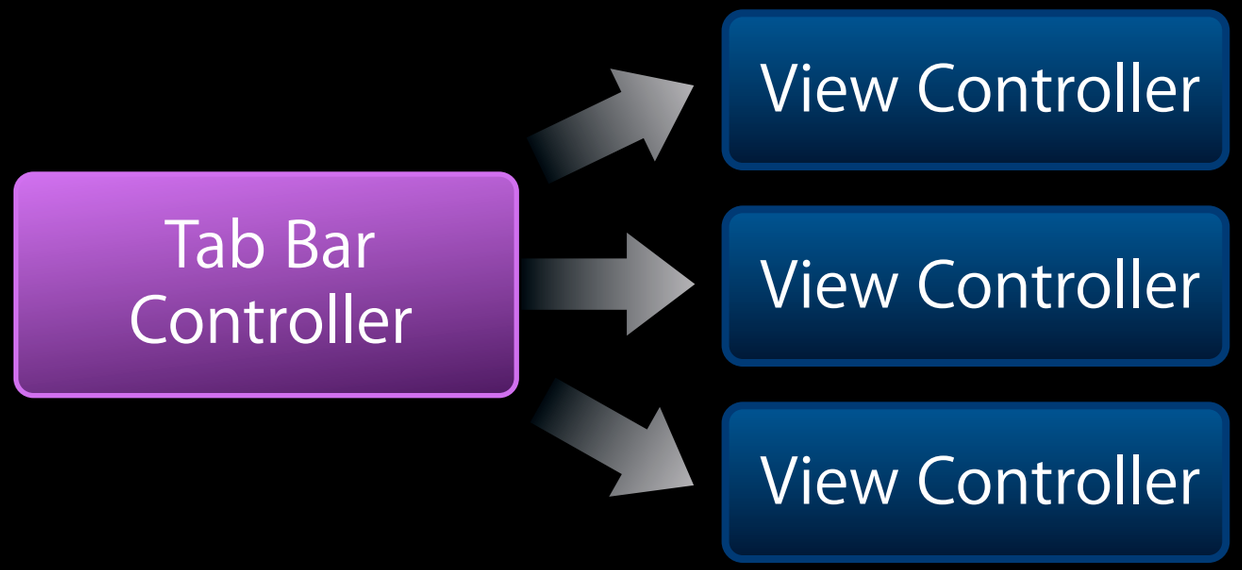
Demo:
Customizing Buttons

Tab Bar Controllers

UITabBarController

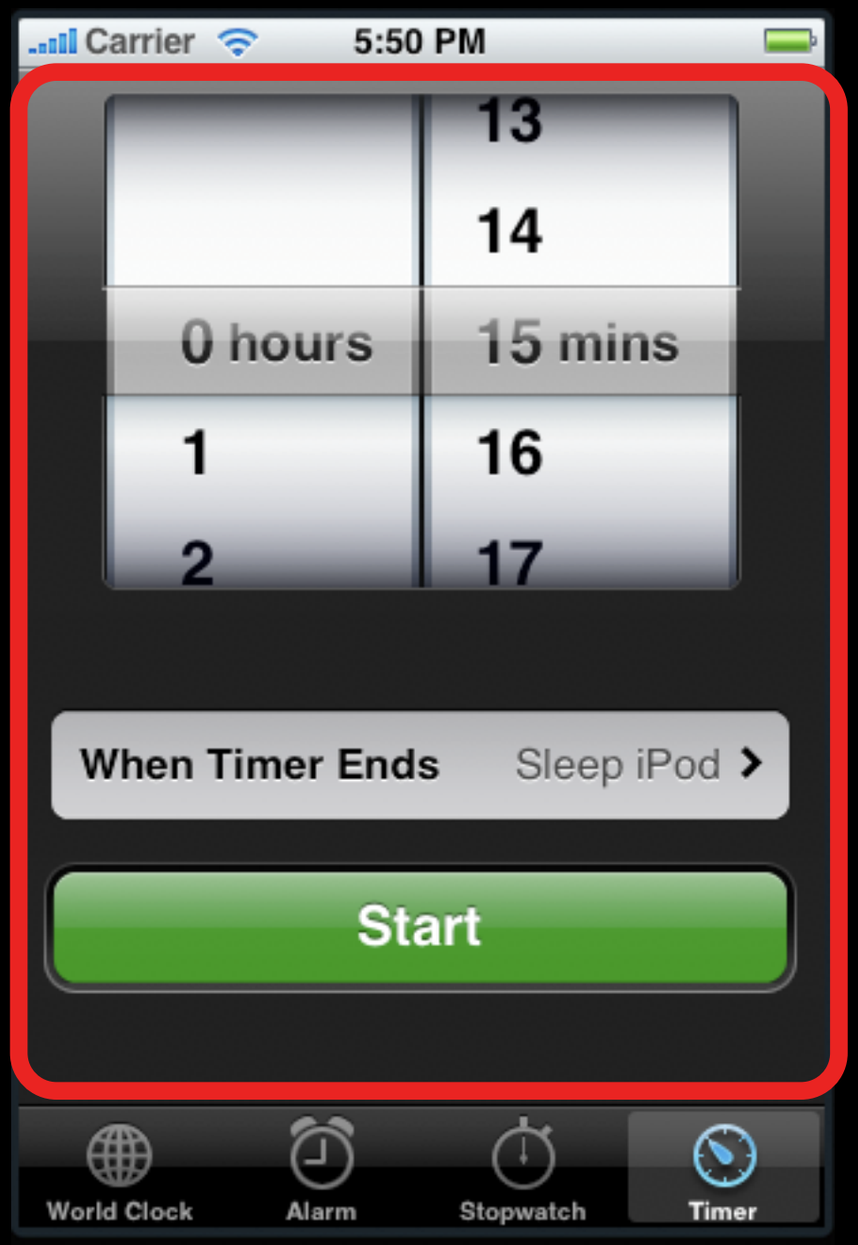
- Array of view controllers
- Tab bar



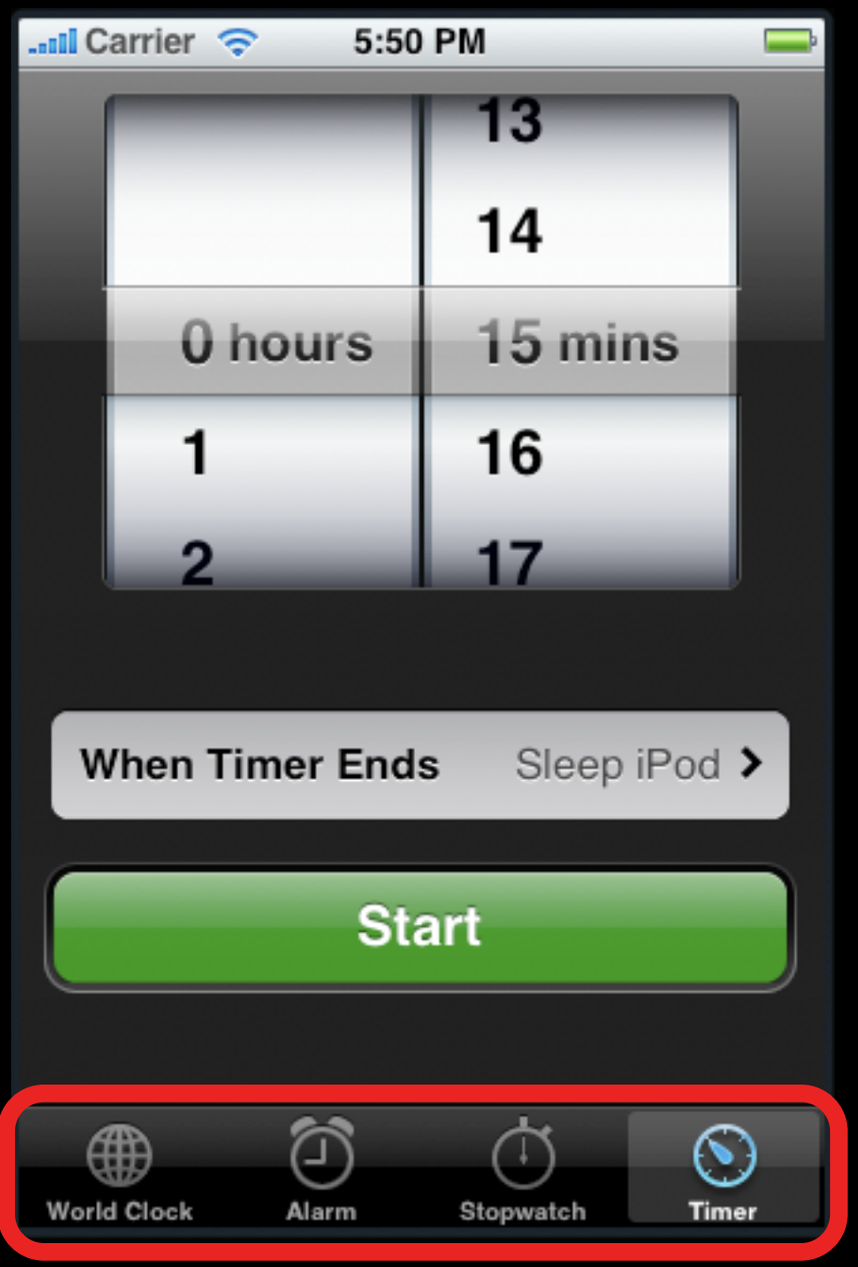


How It Fits Together

- Selected view controller's view



How It Fits Together



- Selected view controller's view
- All view controllers' titles

Setting Up a Tab Bar Controller

```
- (void)applicationDidFinishLaunching
    // Create a tab bar controller
    tabBarController = [[UITabBarController alloc] init];

    // Set the array of view controllers
    tabBarController.viewControllers = myViewControllers;

    // Add the tab bar controller's view to the window
    [window addSubview:tabBarController.view];
}
```

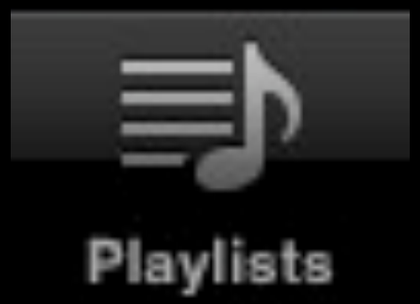
Tab Bar Appearance

- View controllers can define their appearance in the tab bar

- UITabBarItem
  - Title + image or system item
- Each view controller comes with a tab bar item for customizing

Creating Tab Bar Items

- Title and image




```
- (void)viewDidLoad
{
  self.tabBarItem = [[UITabBarItem alloc]
                     initWithTitle:@“Playlists”
                     image:[UIImage imageNamed:@“music.png”]
                     tag:0]
}
```

Creating Tab Bar Items

- System item


```
- (void)viewDidLoad
{
  self.tabBarItem = [[UITabBarItem alloc]
                      initWithTabBarSystemItem:
                        UITabBarSystemItemBookmarks
                      tag:0]
}
```

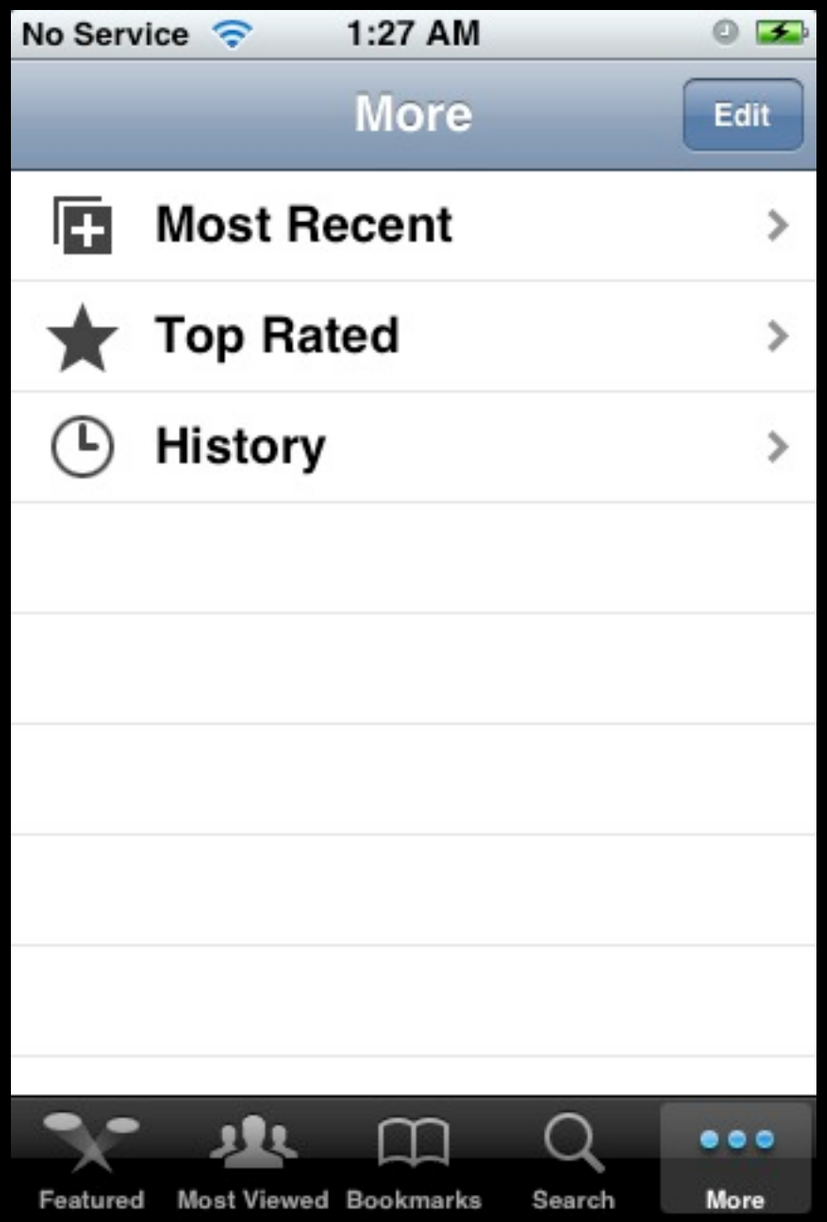
Demo:
Using a Tab Bar Controller

More View Controllers

- What happens when a tab bar controller has too many view controllers to display at once?
  - "More" tab bar item displayed automatically

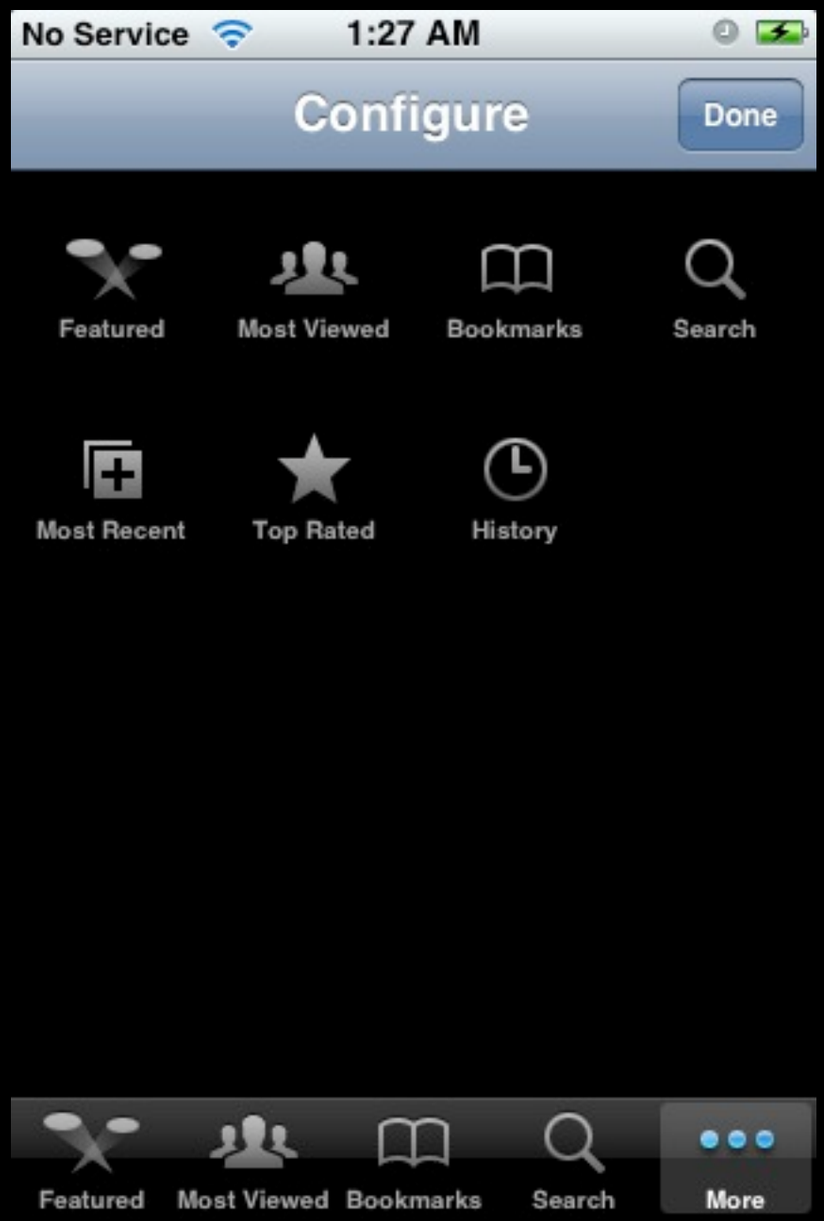
More View Controllers

- What happens when a tab bar controller has too many view controllers to display at once?
  - "More" tab bar item displayed automatically
  - User can navigate to remaining view controllers



More View Controllers

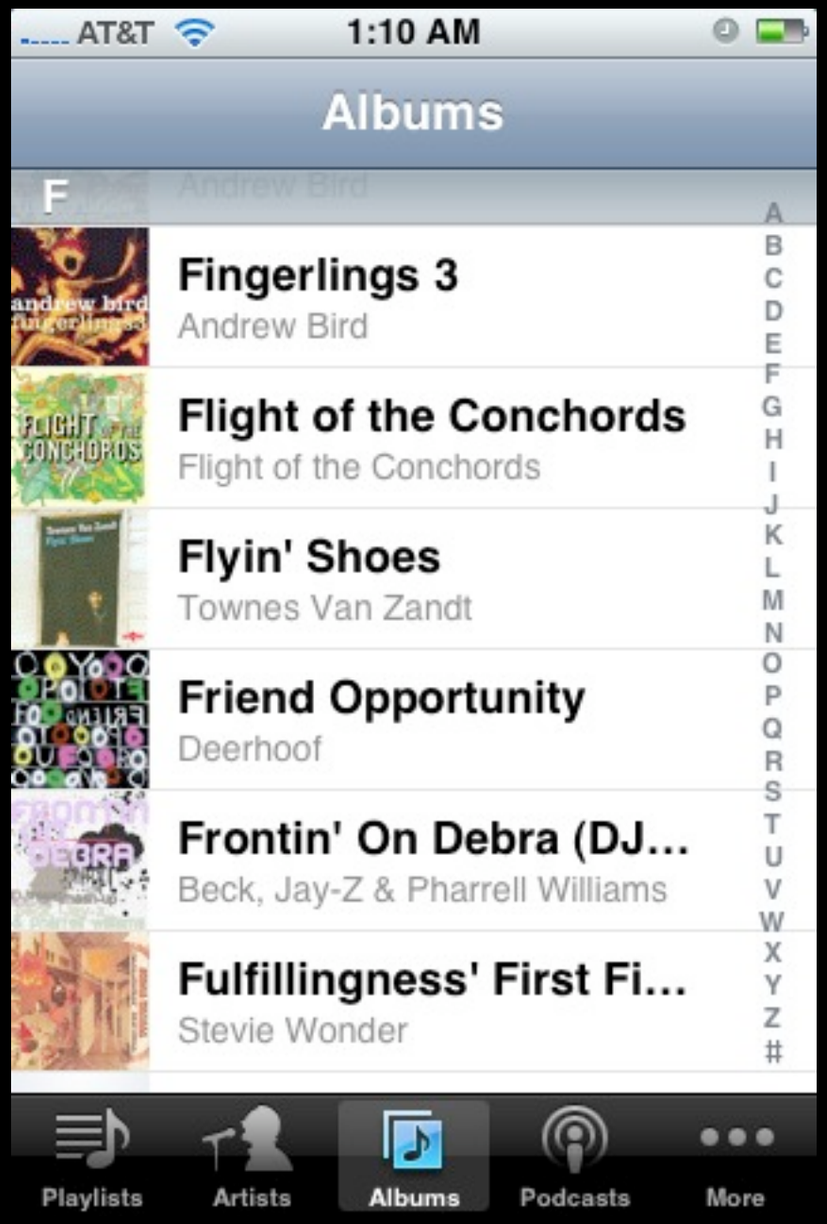
- What happens when a tab bar controller has too many view controllers to display at once?
  - "More" tab bar item displayed automatically
  - User can navigate to remaining view controllers
  - Customize order



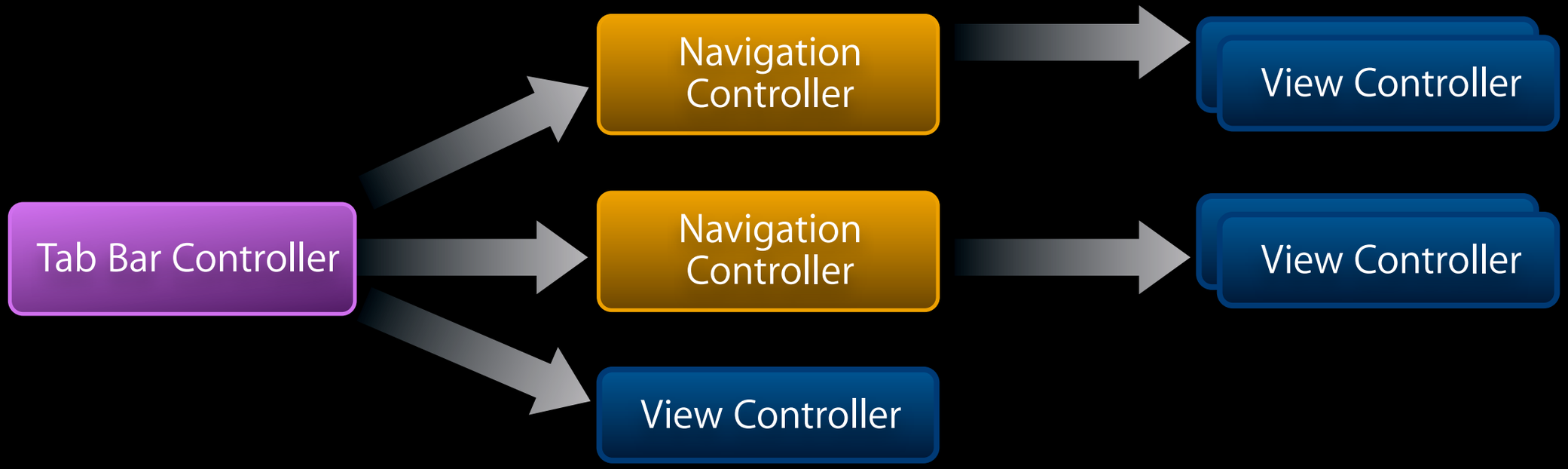
Combining Approaches

Tab Bar + Navigation Controllers

Multiple parallel hierarchies



Tab Bar + Navigation Controllers



Nesting Navigation Controllers

- Create a tab bar controller

```
tabBarController = [[UITabBarController alloc] init];
```

- Create each navigation controller

```
navController = [[UINavigationController alloc] init];
[navController pushViewController:firstViewController
                         animated:NO];
```

- Add them to the tab bar controller

```
tabBarController.viewControllers = [NSArray arrayWithObjects:
                                    navController,
                                    anotherNavController,
                                    someViewController,
                                    nil];
```

Questions?