CS193P - Lecture 7

iPhone Application Development

Navigation & Tab Bar Controllers

Announcements

- Assignment 3 is due tomorrow
- Paparazzi 1 is due on Wednesday February 3rd

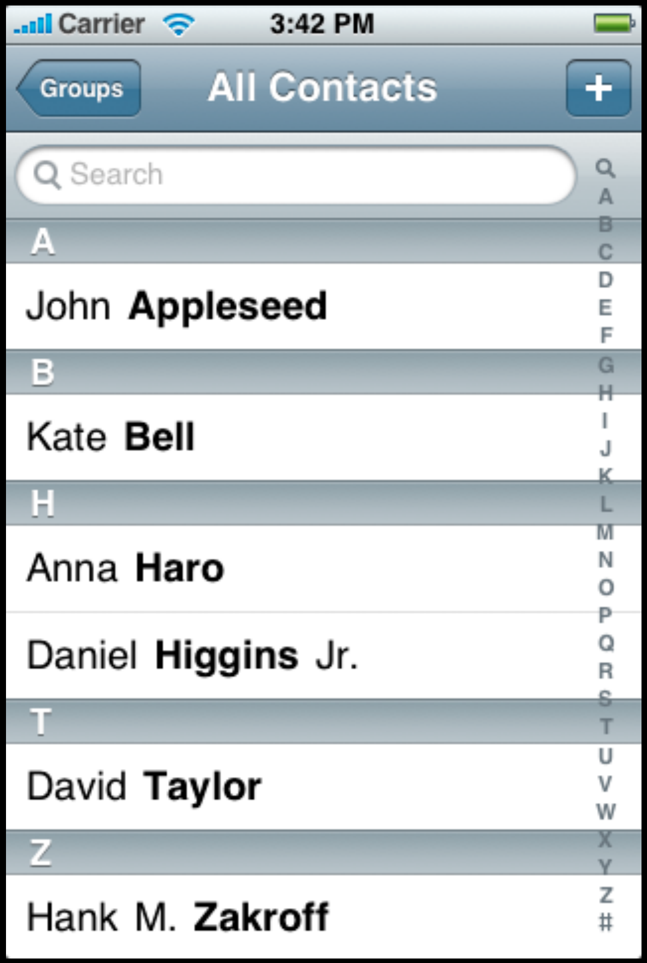
Today’s Topics

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

Navigation Controllers

UINavigationController

- Stack of view controllers
- Navigation bar



All Contacts

A

John **Appleseed**

B

Kate **Bell**

H

Anna **Haro**

Daniel **Higgins** Jr.

T

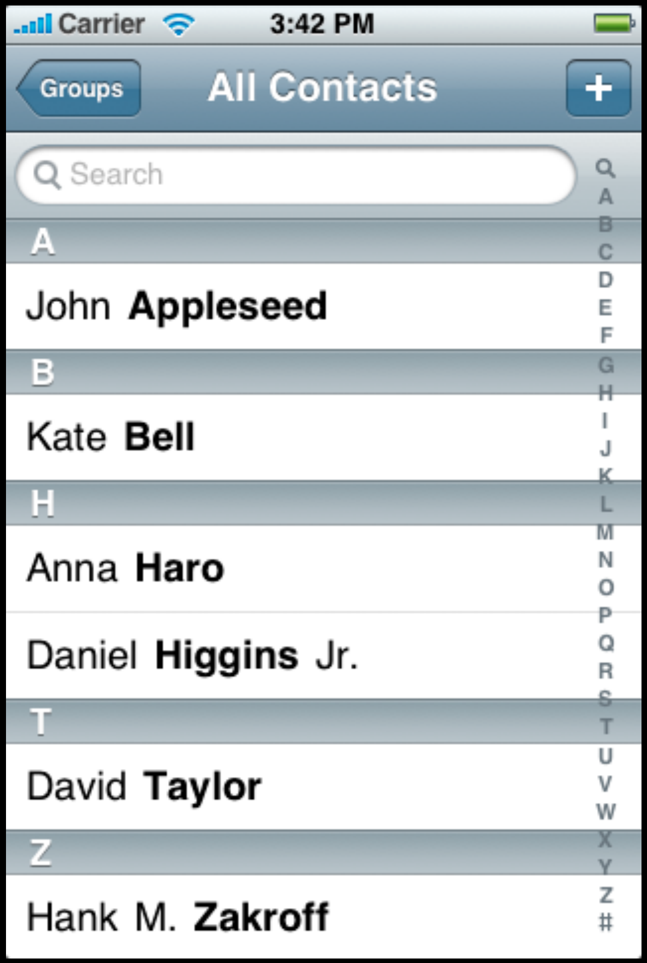
David **Taylor**

Z

Hank M. **Zakroff**

Navigation Controller

UINavigationController



A

John **Appleseed**

B

Kate **Bell**

H

Anna **Haro**

Daniel **Higgins** Jr.

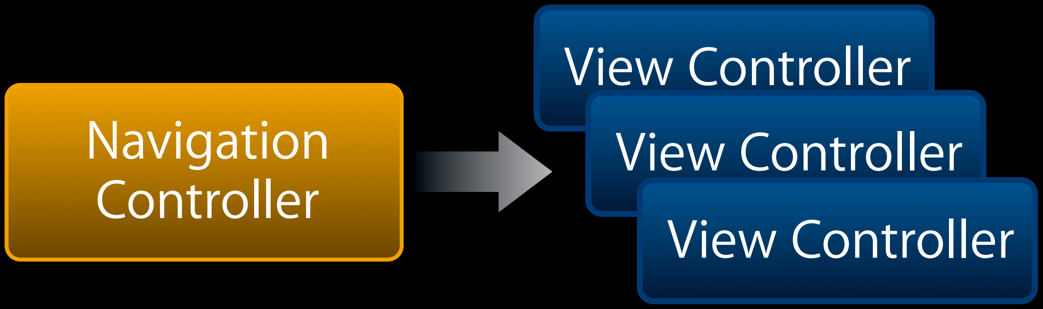
T

David **Taylor**

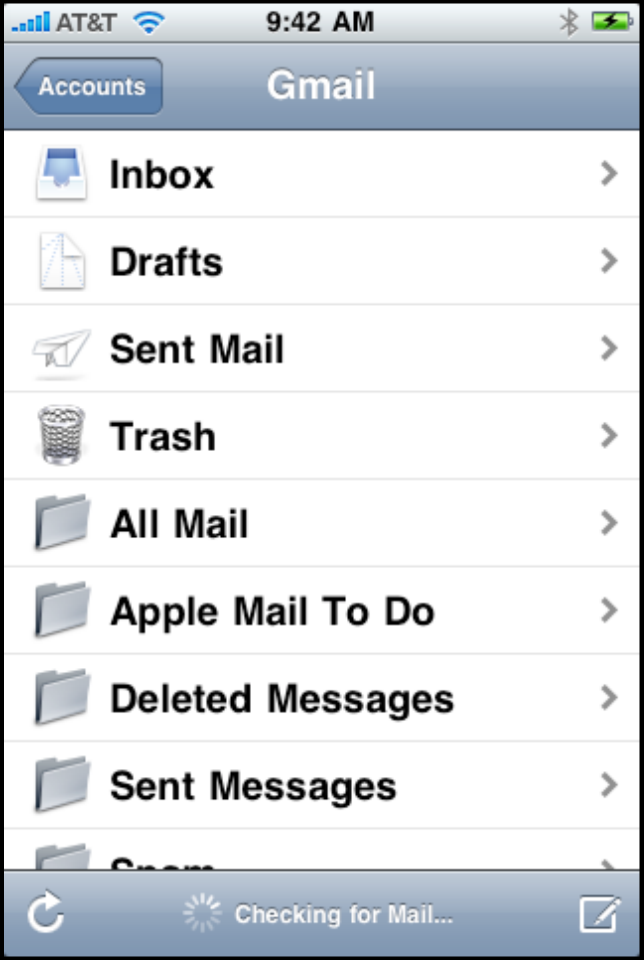
Z

Hank M. **Zakroff**

- Stack of view controllers
- Navigation bar



How It Fits Together


AT&T 9:42 AM

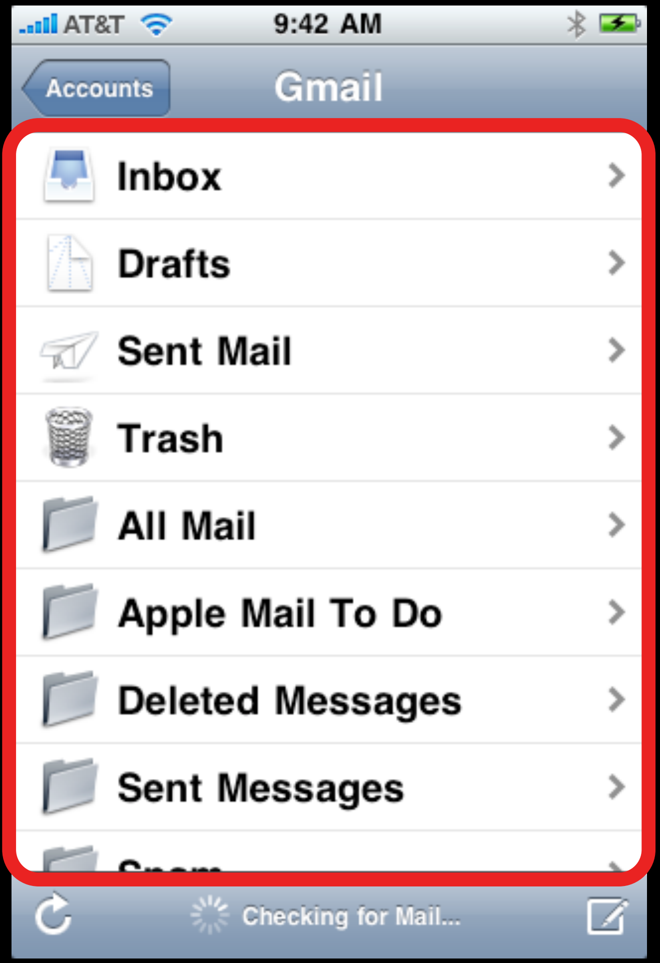
Accounts Gmail

- Inbox
- Drafts
- Sent Mail
- Trash
- All Mail
- Apple Mail To Do
- Deleted Messages
- Sent Messages
- Spam

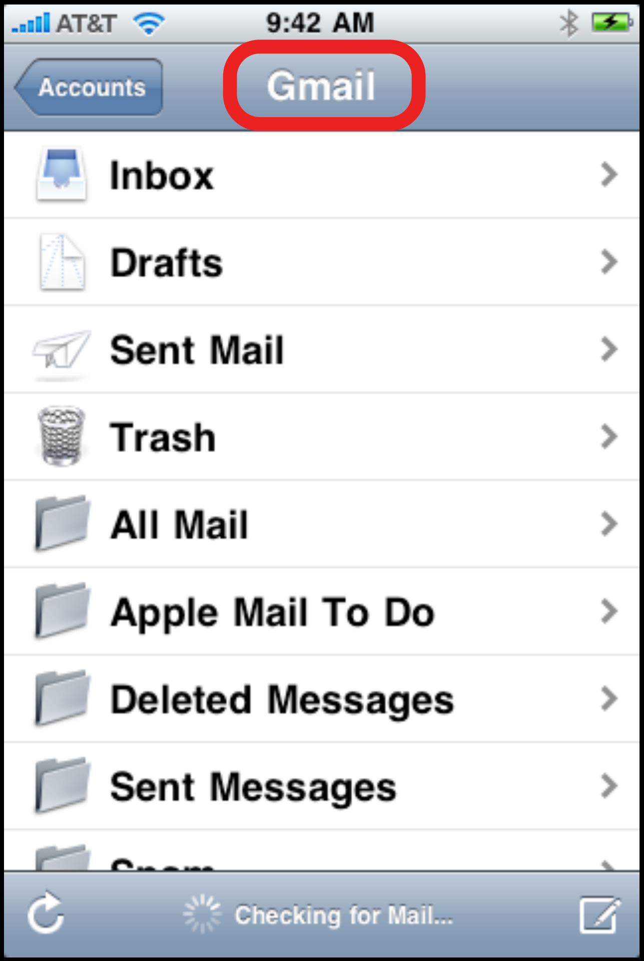
Checking for Mail...

How It Fits Together

- Top view controller's view

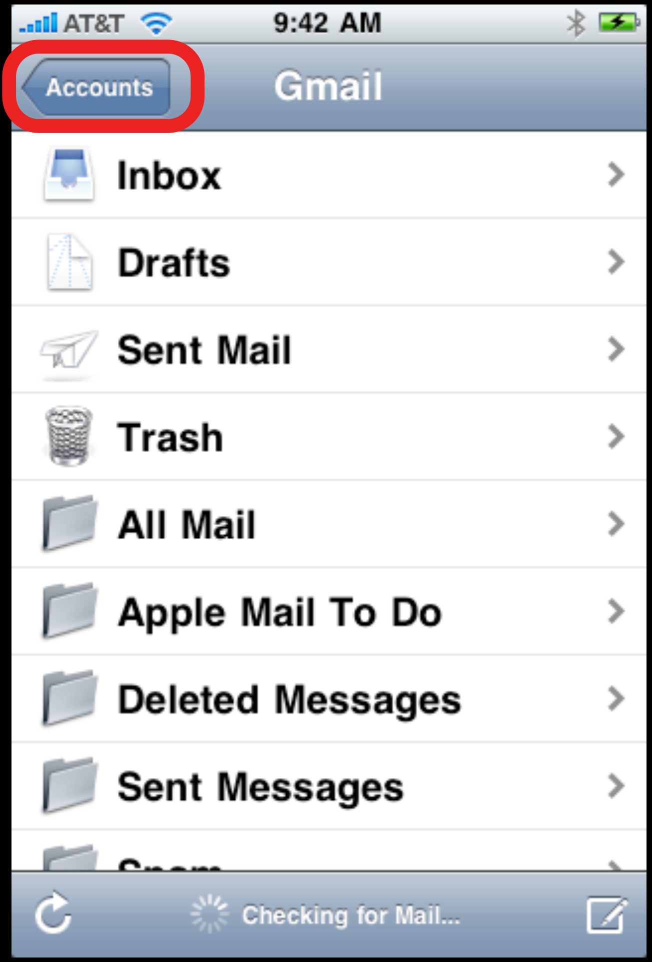


How It Fits Together



- Top view controller’s view
- Top view controller’s title

How It Fits Together



AT&T 9:42 AM

Accounts Gmail

Inbox

Drafts

Sent Mail

Trash

All Mail

Apple Mail To Do

Deleted Messages

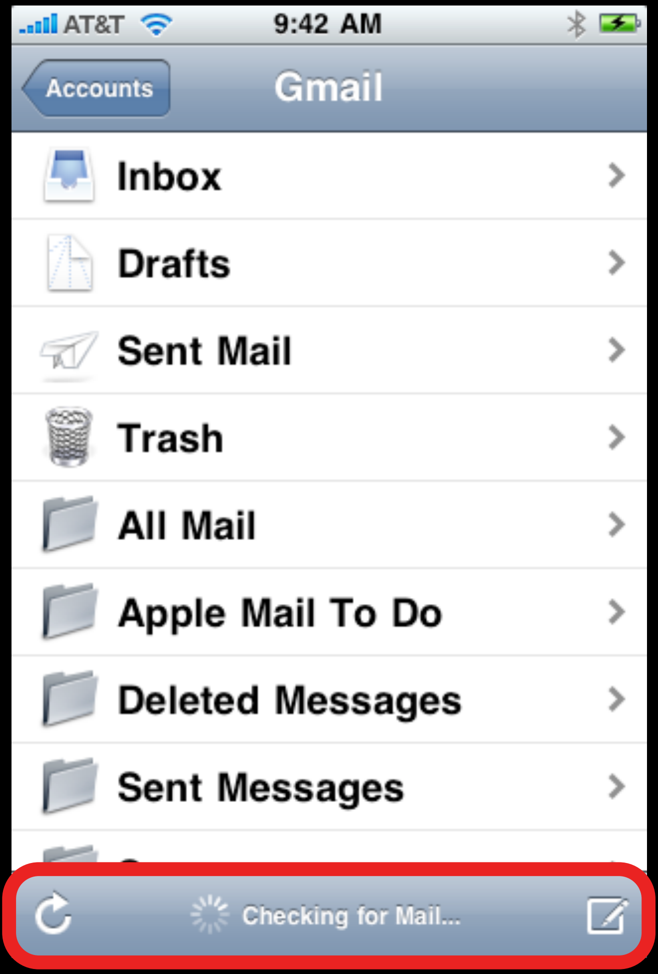
Sent Messages

Spam

Checking for Mail...

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

How It Fits Together



- Top view controller's view
- Top view controller's title
- Previous view controller's title
- Top view controller's toolbar items (iPhone OS 3.0)

Modifying the Navigation Stack

Modifying the Navigation Stack

- **Push** to add a view controller

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

Modifying the Navigation Stack

- **Push** to add a view controller

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

- **Pop** to remove a view controller

```
- (UIViewController *)popViewControllerAnimated:(BOOL)animated;
```

Modifying the Navigation Stack

- **Push** to add a view controller

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

- **Pop** to remove a view controller

```
- (UIViewController *)popViewControllerAnimated:(BOOL)animated;
```

- **Set** to change the entire stack of view controllers (iPhone OS 3.0)

```
- (void)setViewControllers:(NSArray *)viewControllers
                  animated:(BOOL)animated;
```

Pushing Your First View Controller

Pushing Your First View Controller

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







}
```

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];



}
```

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

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];
}
```

In Response to User Actions

- Push from within a view controller on the stack

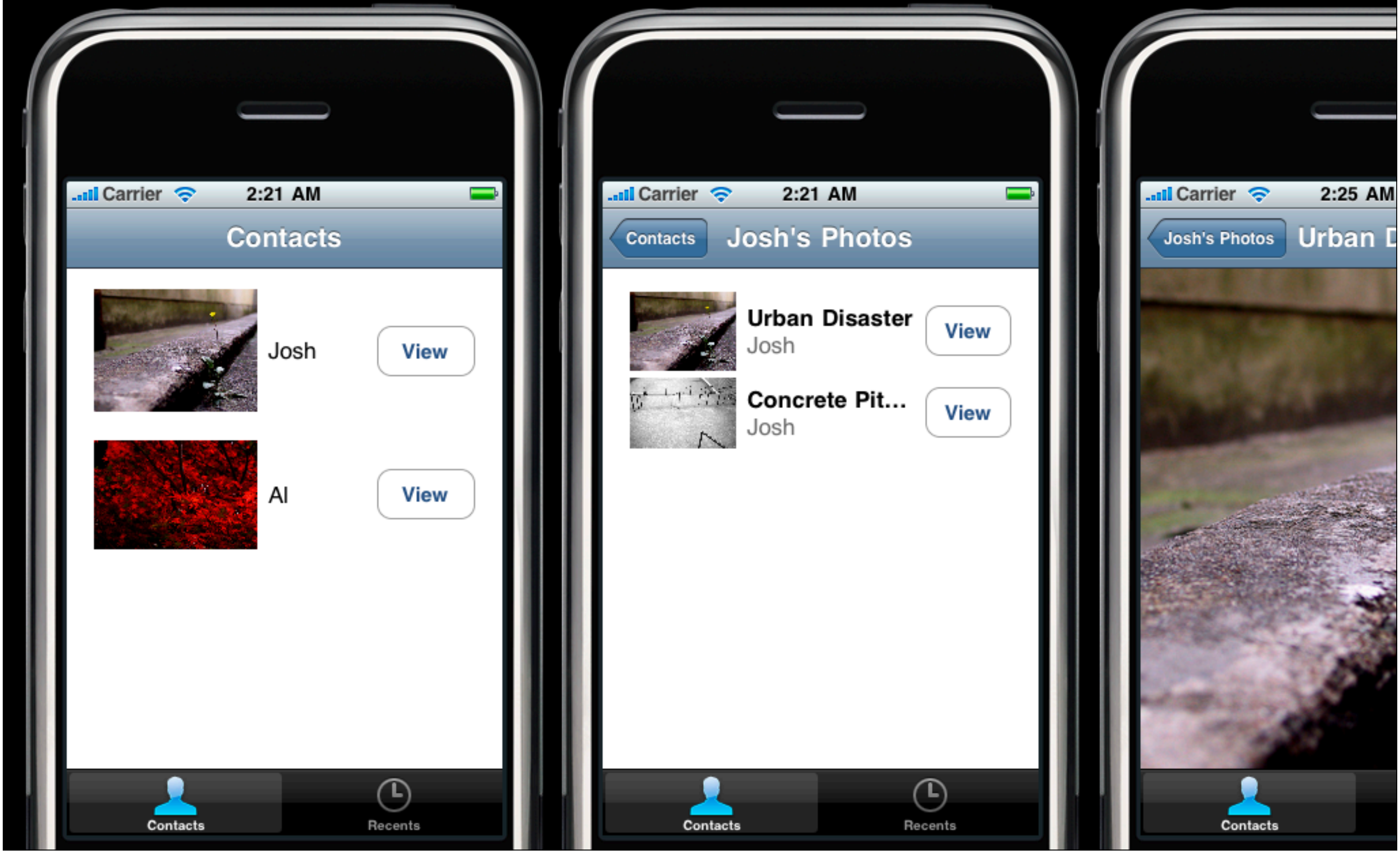
```
- (void)someAction:(id)sender
{
  // Potentially create another view controller
  UIViewController *viewController = ...;

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

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

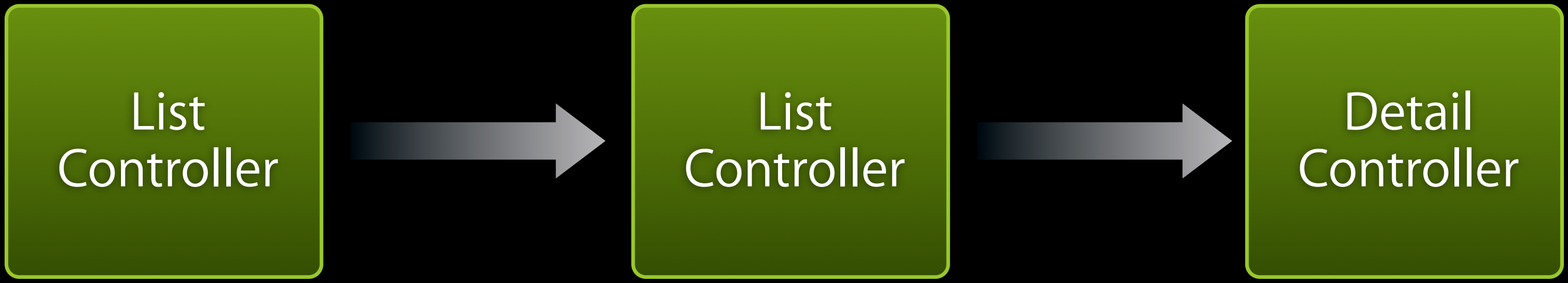
Demo:
Pushing & Popping

Application Data Flow

Paparazzi


Carrier
2:21 AM
Contacts
Josh
View
Al
View
Contacts
Recents
Carrier
2:21 AM
Contacts
Josh's Photos
Urban Disaster
Josh
View
Concrete Pit...
Josh
View
Contacts
Recents
Carrier
2:25 AM
Josh's Photos
Urban D
Contacts

A Controller for Each Screen



Connecting View Controllers

Connecting View Controllers

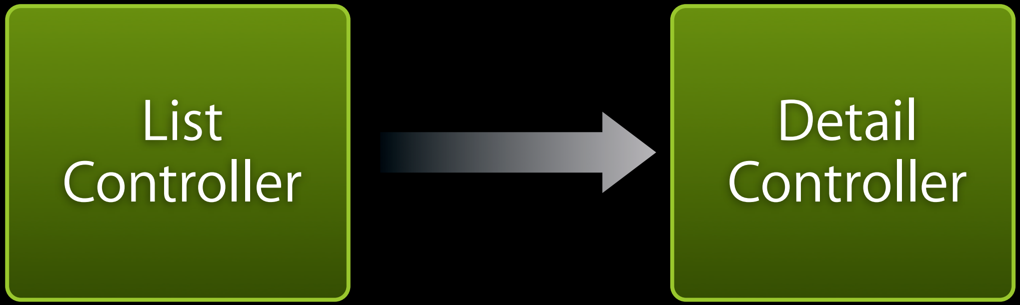
- Multiple view controllers may need to **share data**

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

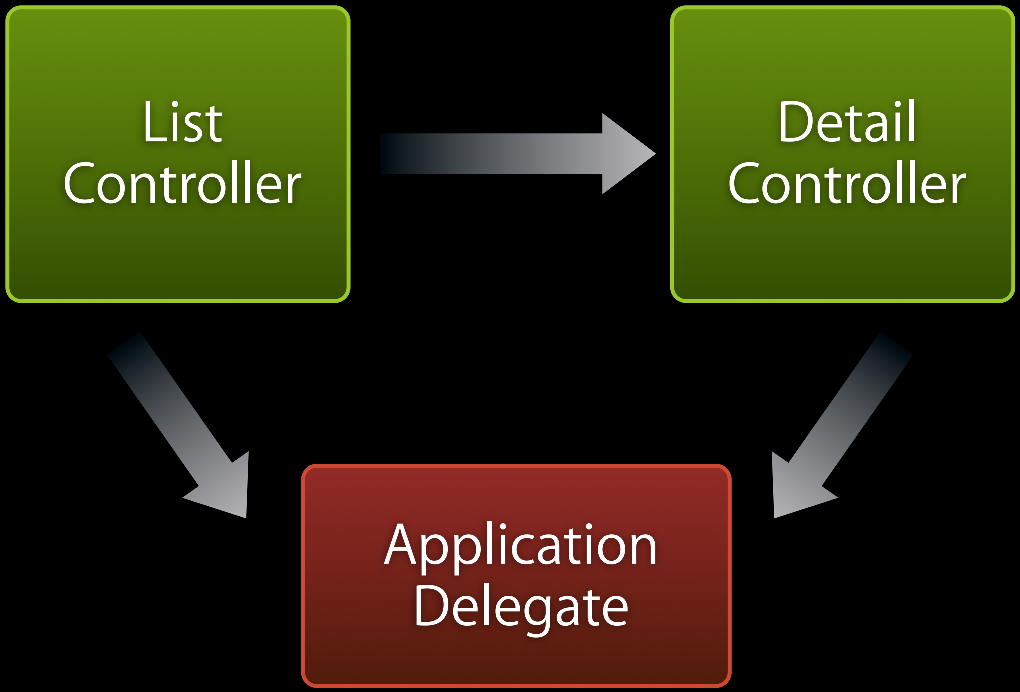
How Not To Share Data

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



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



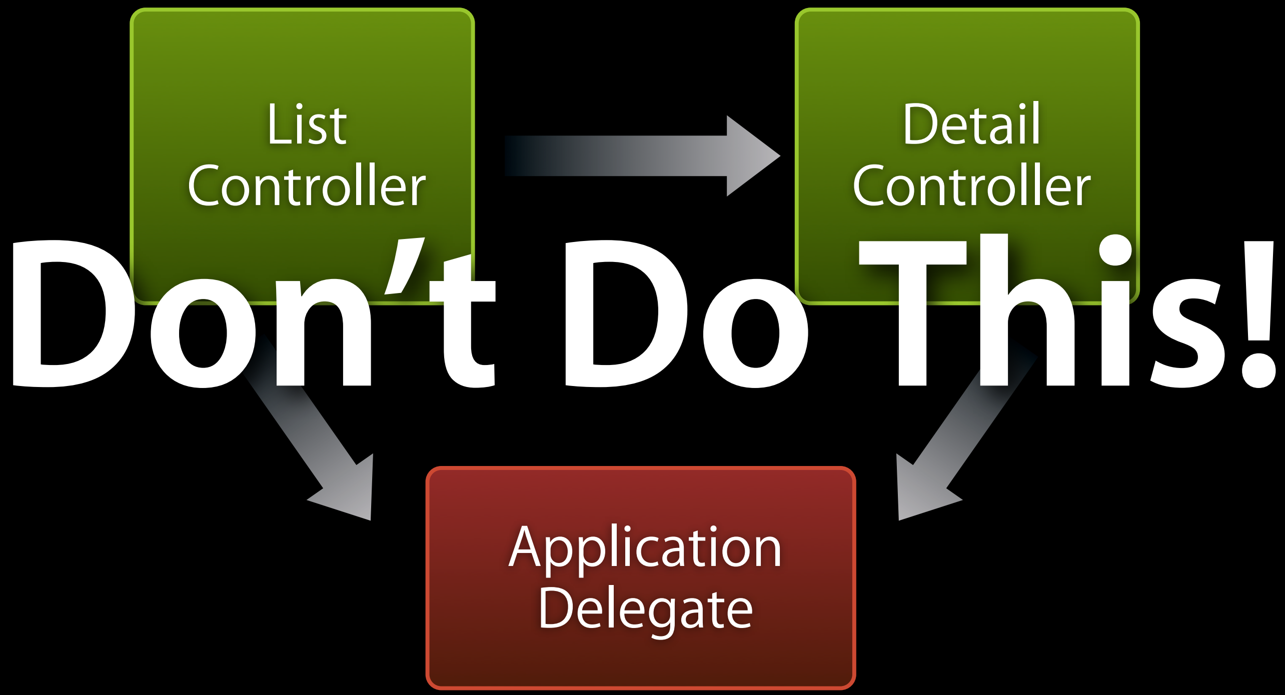
List Controller

Detail Controller

Application Delegate

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



List Controller

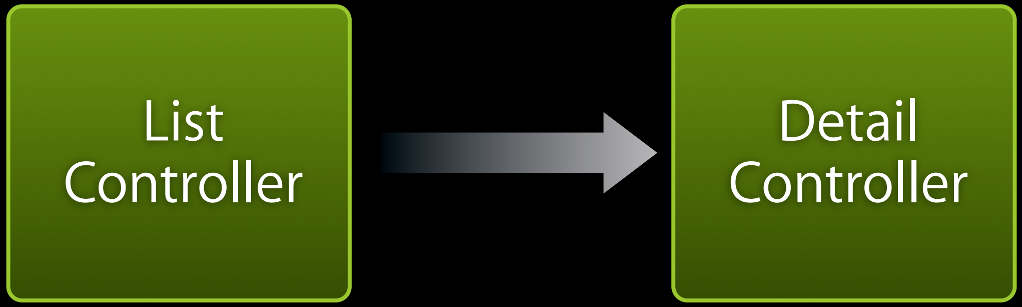
Detail Controller

Application Delegate

Don’t Do This!

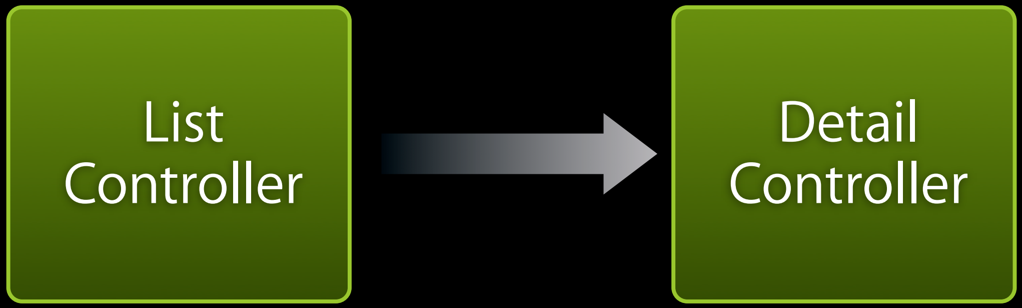
Best Practices for Data Flow

- Figure out **exactly** what needs to be communicated



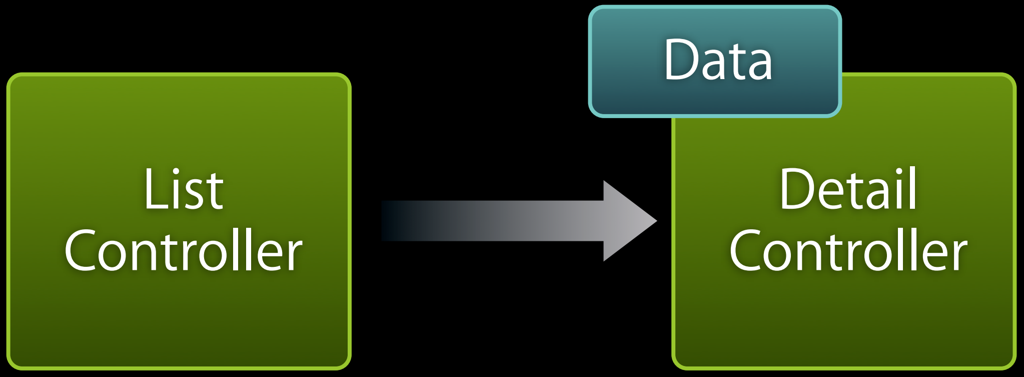
Best Practices for Data Flow

- Figure out **exactly** what needs to be communicated
- **Define input parameters** for your view controller



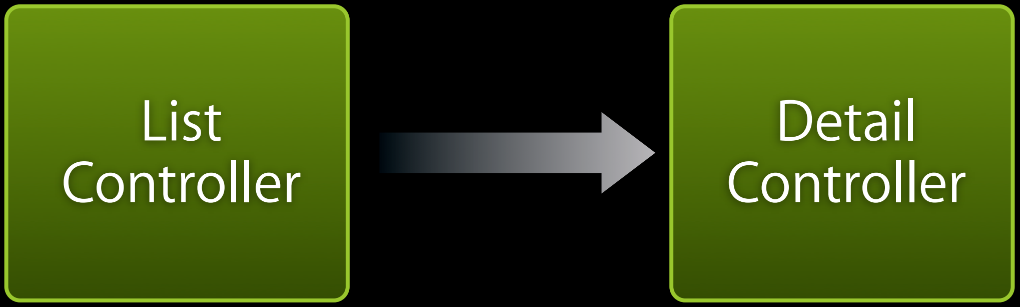
Best Practices for Data Flow

- Figure out **exactly** what needs to be communicated
- **Define input parameters** for your view controller



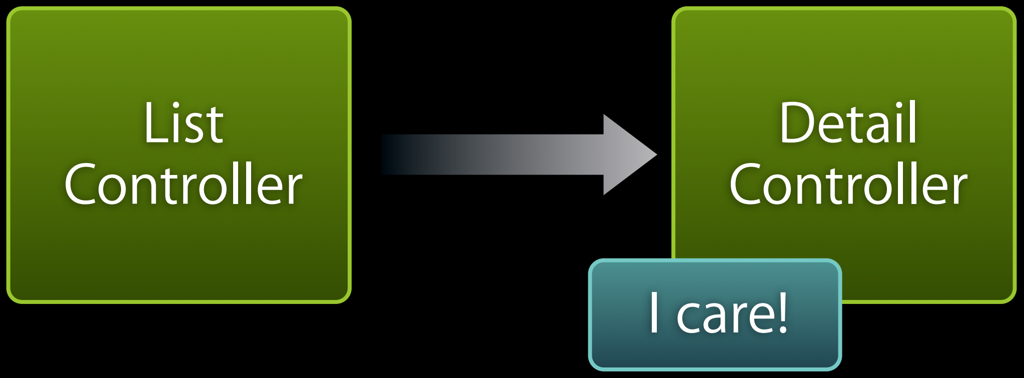
Best Practices for Data Flow

- 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)



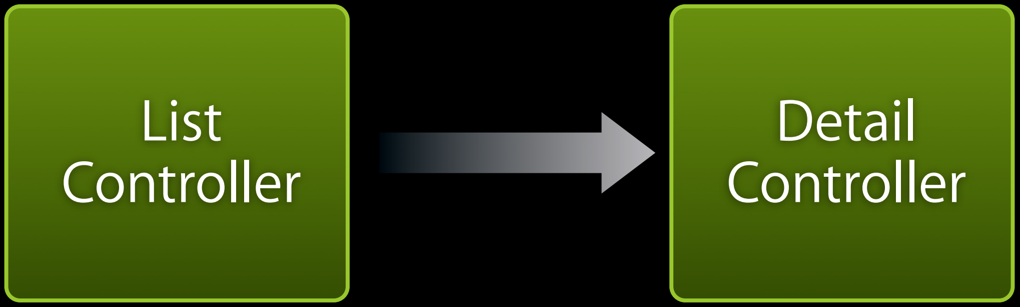
Best Practices for Data Flow

- 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)



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)



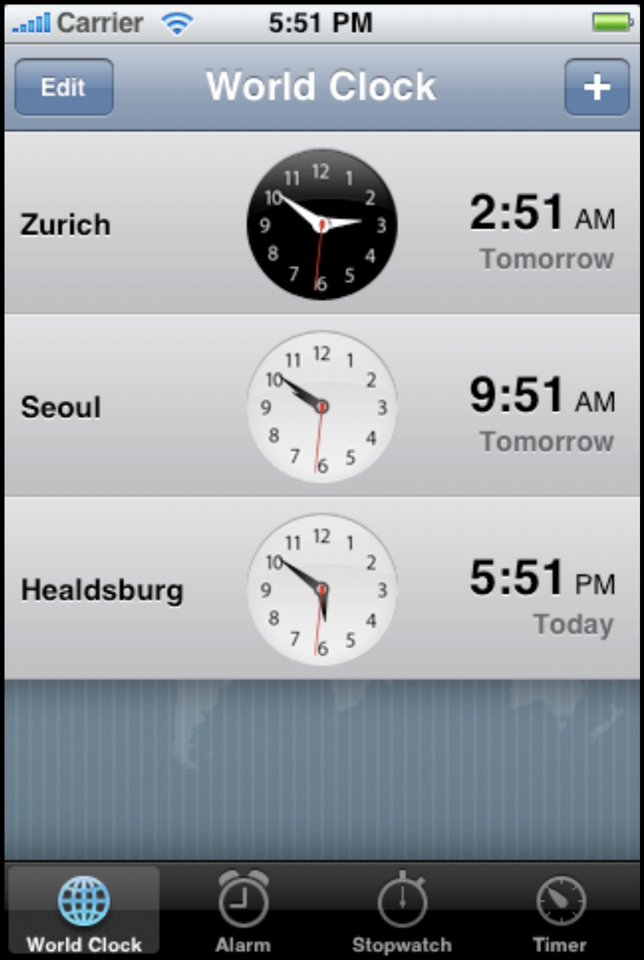
Example: UIImagePickerController

Demo:
Passing Data Along

Customizing Navigation

Customizing Navigation

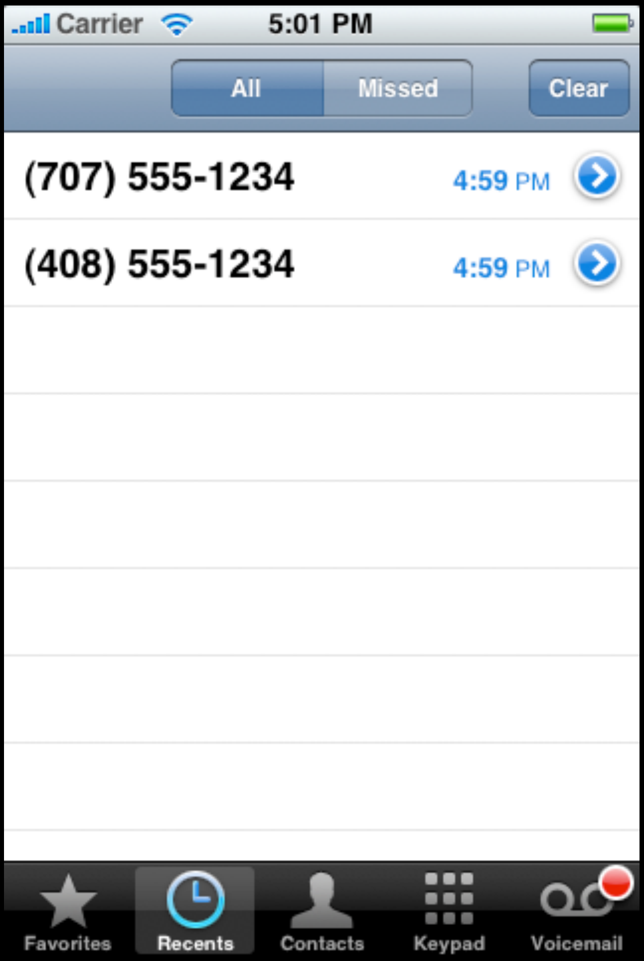
- Buttons or custom controls
- Interact with the entire screen



Zurich 2:51 AM Tomorrow

Seoul 9:51 AM Tomorrow

Healdsburg 5:51 PM Today

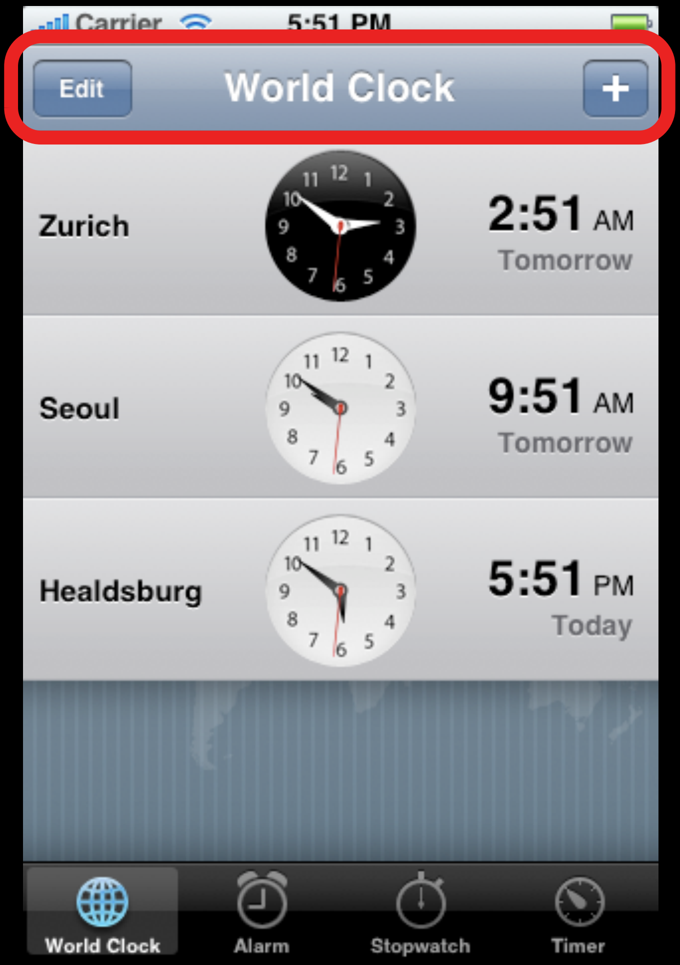


(707) 555-1234 4:59 PM

(408) 555-1234 4:59 PM

Customizing Navigation

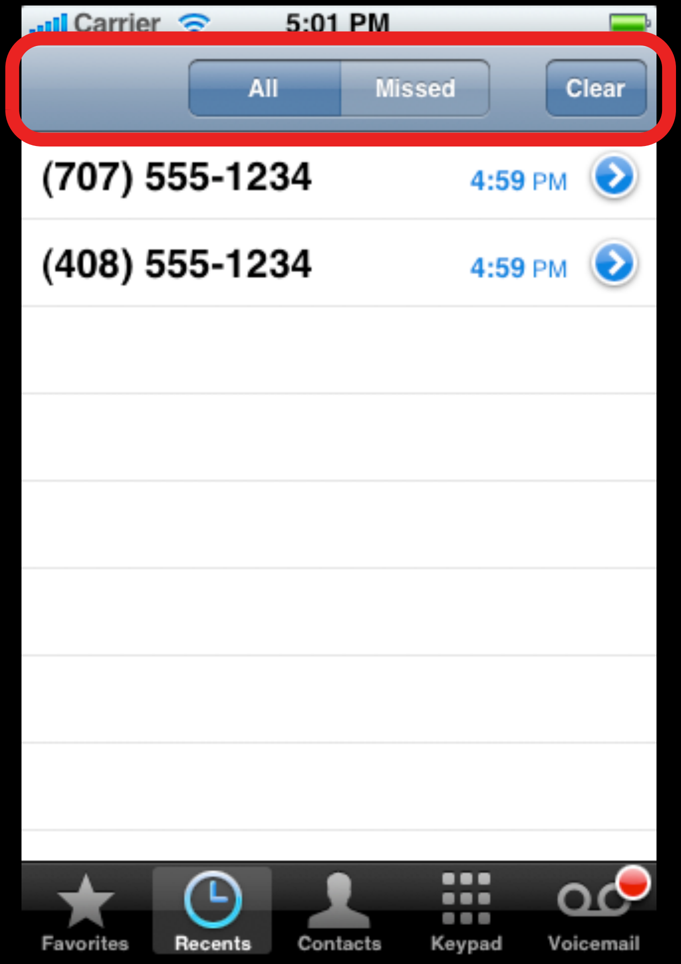
- Buttons or custom controls
- Interact with the entire screen



Zurich 2:51 AM Tomorrow

Seoul 9:51 AM Tomorrow

Healdsburg 5:51 PM Today



(707) 555-1234 4:59 PM

(408) 555-1234 4:59 PM

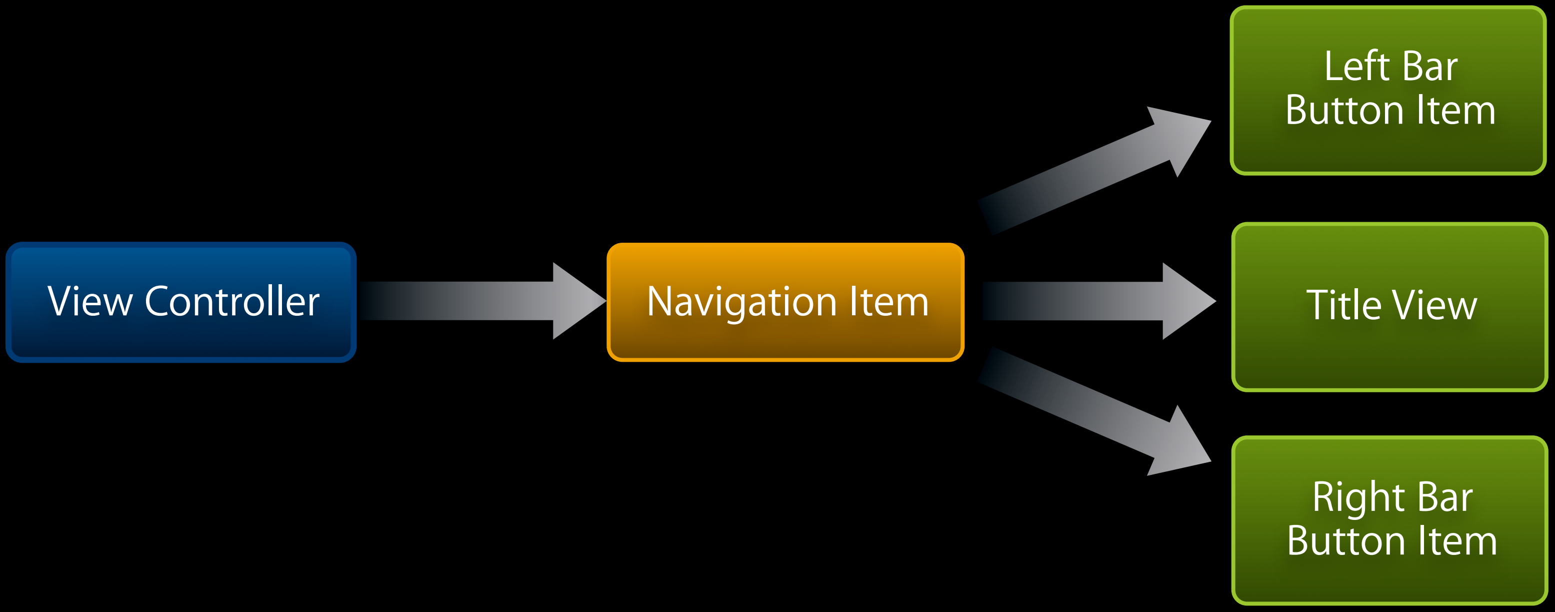
UINavigationItem

- Describes appearance of the navigation bar
    - Title string or custom title view
    - Left & right bar buttons
    - More properties defined in UINavigationBar.h

UINavigationItem

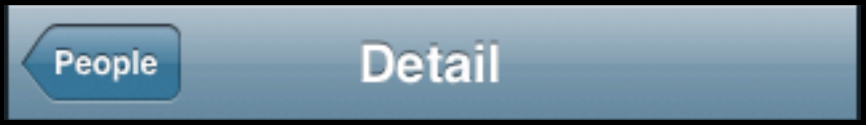
- Describes appearance of the navigation bar
  - Title string or custom title view
  - Left & right bar buttons
  - More properties defined in UINavigationBar.h
- **Every view controller has a navigation item** for customizing
  - Displayed when view controller is on **top of the stack**

Navigation Item Ownership



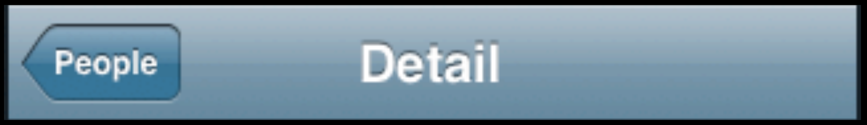
Displaying a Title

- 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



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



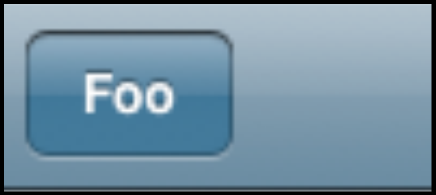
```
viewController.title = @“Detail”;
```

Left & Right Buttons

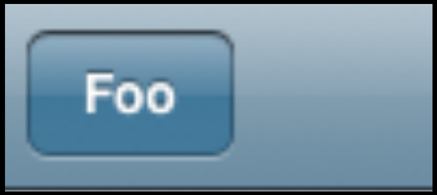
- UIBarButtonItem
    - Special object, defines appearance & behavior for items in navigation bars and toolbars
- Display a string, image or predefined system item
- Target + action (like a regular button)

Text Bar Button Item

Text Bar Button Item



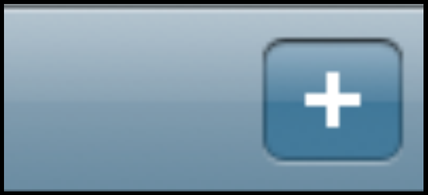
Text Bar Button Item



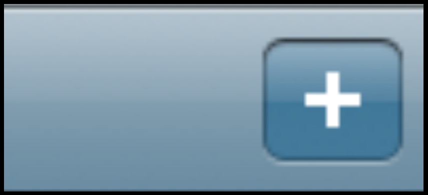
```
- (void)viewDidLoad
{
  UIBarButtonItem *fooButton = [[UIBarButtonItem alloc]
    initWithTitle:@"Foo”
    style:UIBarButtonItemStyleBordered
    target:self
    action:@selector(foo:)];

  self.navigationItem.leftBarButtonItem = fooButton;

  [fooButton release];
}
```

System Bar Button Item

System Bar Button Item



System Bar Button Item



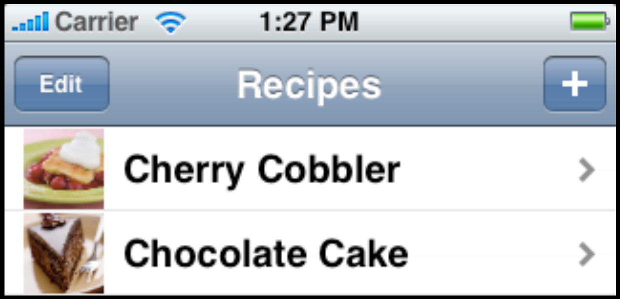
```
- (void)viewDidLoad
{
  UIBarButtonItem *addButton = [[UIBarButtonItem alloc]
    initWithBarButtonSystemItem:UIBarButtonSystemItemAdd
    style:UIBarButtonItemStyleBordered
    target:self
    action:@selector(add:)];

  self.navigationItem.rightBarButtonItem = addButton;

  [addButton release];
}
```

Edit/Done Button

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

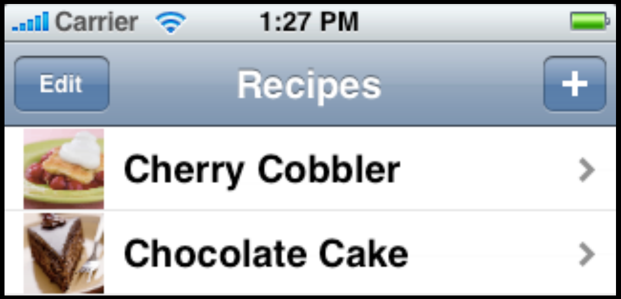
Edit/Done Button

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



Edit/Done Button

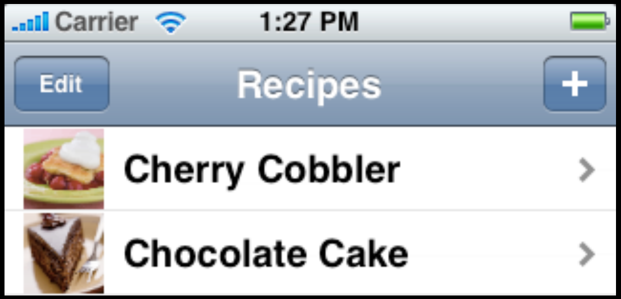
- Very common pattern
- Every view controller has one available
  - Target/action already set up



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

Edit/Done Button

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



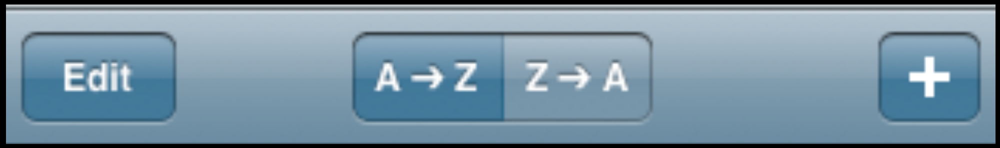
```
self.navigationItem.leftBarButtonItem = self.editButtonItem;

// 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

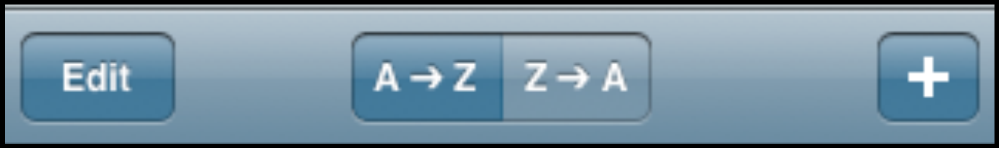
Custom Title View

- Arbitrary view in place of the title



Custom Title View

- Arbitrary view in place of the title



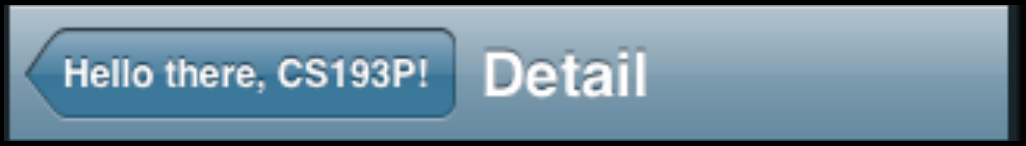
```
UISegmentedControl *segmentedControl = ...
self.navigationItem.titleView = segmentedControl;
[segmentedControl release];
```

Back Button

- Sometimes a shorter back button is needed

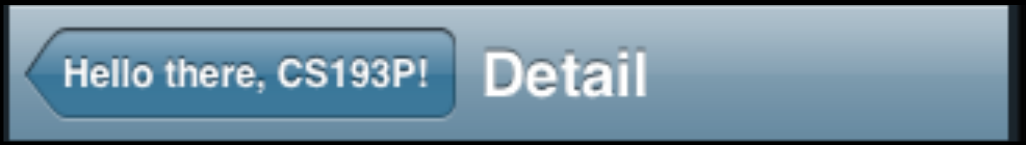
Back Button

- Sometimes a shorter back button is needed



Back Button

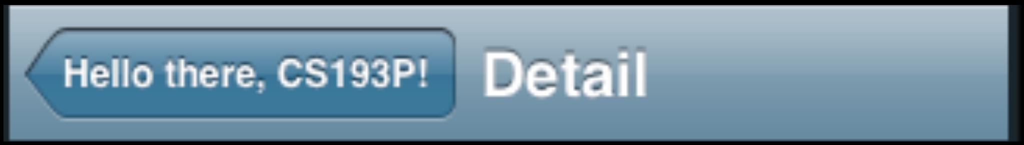
- Sometimes a shorter back button is needed



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

Back Button

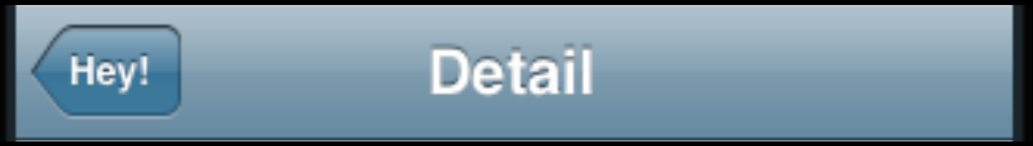
- Sometimes a shorter back button is needed



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

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

[heyButton release];
```

Back Button

- Sometimes a shorter back button is needed



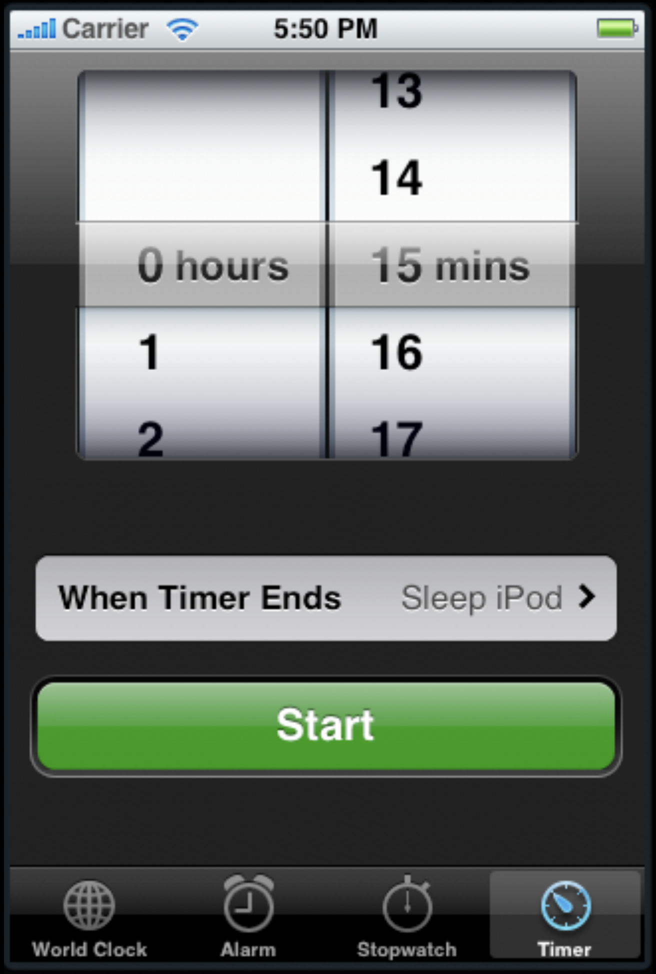
```
self.title = @“Hello there, CS193P!”;

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

[heyButton release];
```

Demo:
Customizing Buttons

Tab Bar Controllers

UITabBarController

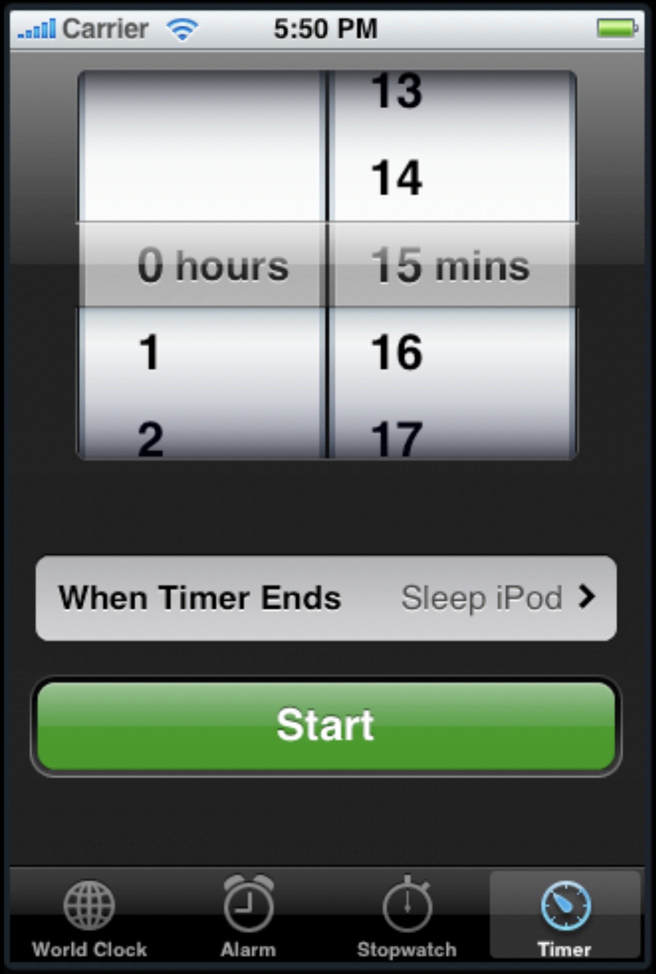
- Array of view controllers
- Tab bar



Tab Bar Controller

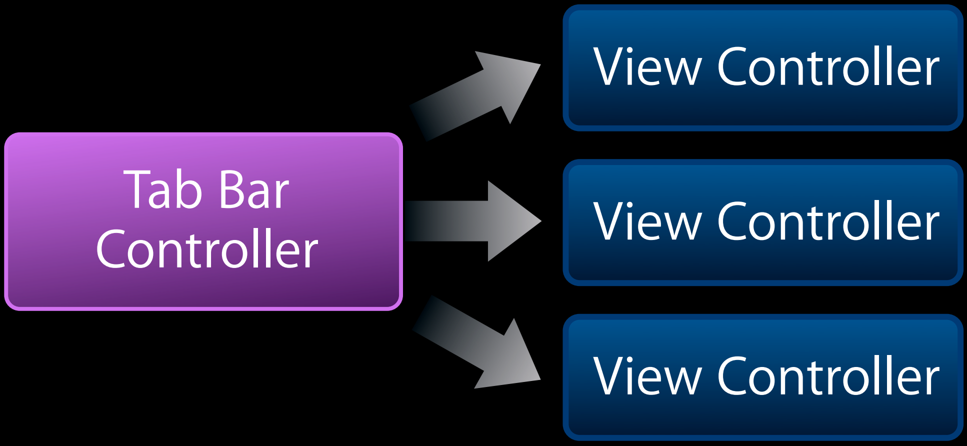
UITabBarController

- Array of view controllers
- Tab bar



0 hours 15 mins

When Timer Ends Sleep iPod ›



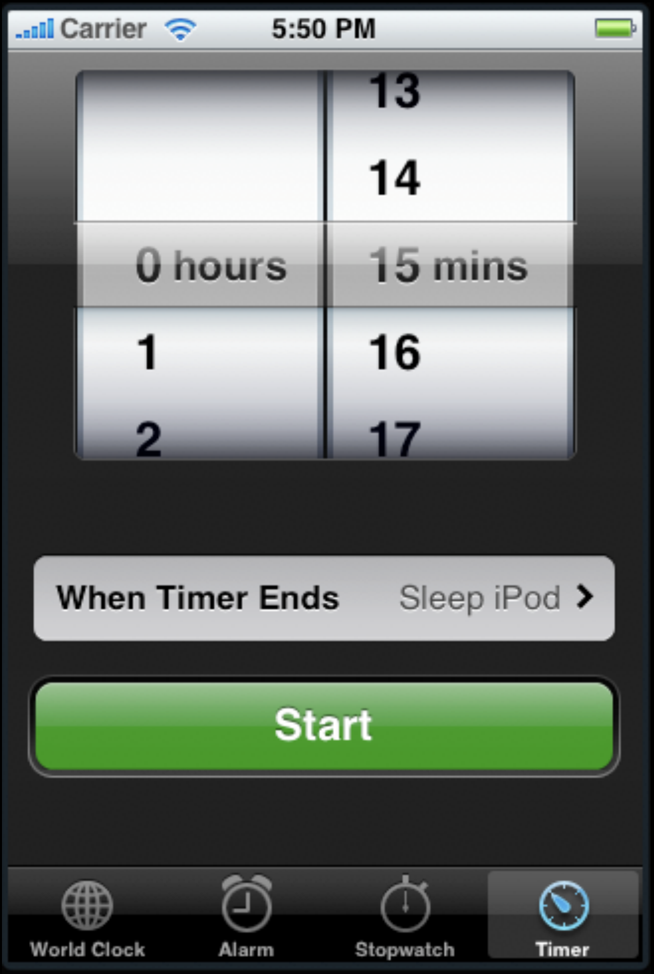
Tab Bar Controller

View Controller

View Controller

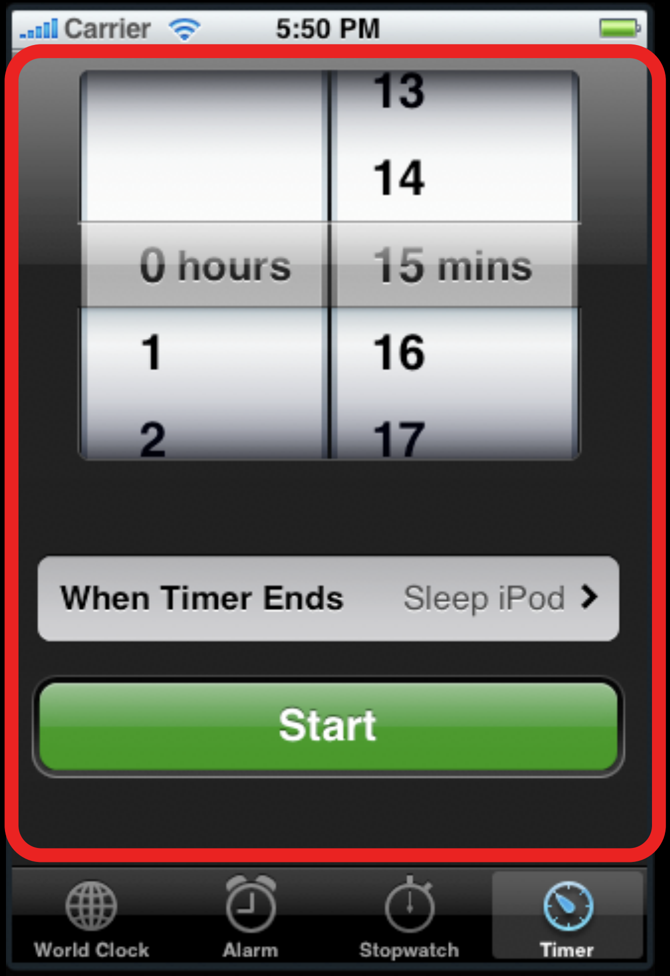
View Controller

How It Fits Together

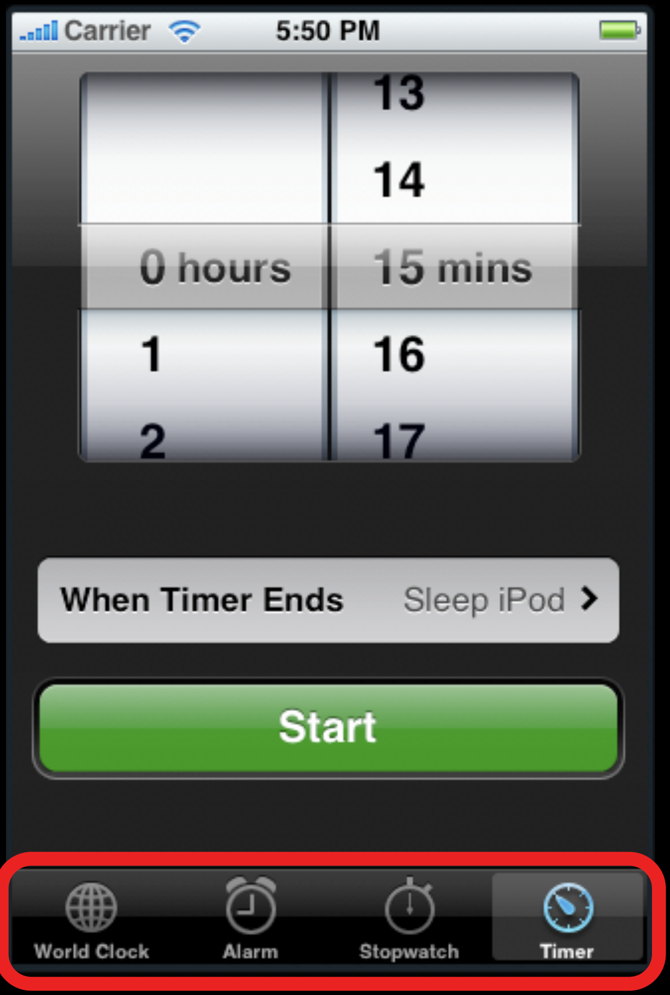


How It Fits Together

- Selected view controller's view



How It Fits Together



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

Setting Up a Tab Bar Controller

Setting Up a Tab Bar Controller

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







}
```

Setting Up a Tab Bar Controller

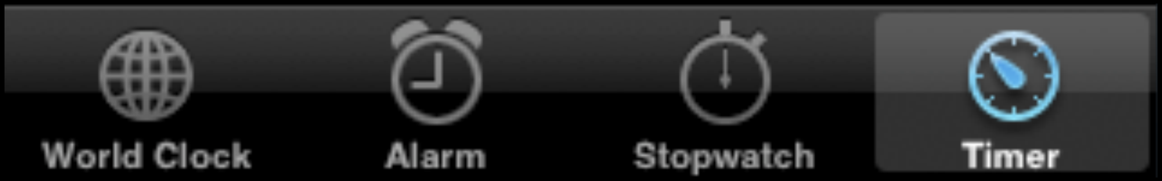
```
- (void)applicationDidFinishLaunching
    // Create a tab bar controller
    tabBarController = [[UITabBarController alloc] init];

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




}
```

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];
}
```

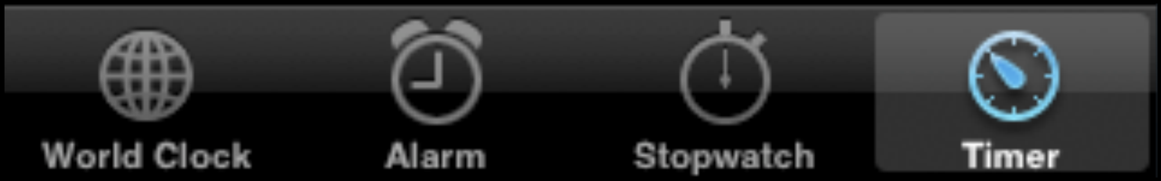
Tab Bar Appearance

- View controllers can define their appearance in the tab bar



Tab Bar Appearance

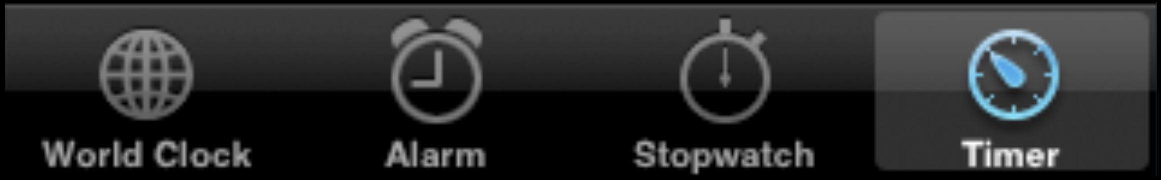
- View controllers can define their appearance in the tab bar



- UITabBarItem
  - Title + image or system item

Tab Bar Appearance

- View controllers can define their appearance in the tab bar



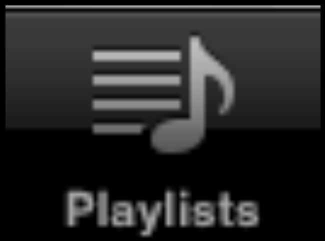
- UITabBarItem
  - Title + image or system item
- Each view controller comes with a tab bar item for customizing

Creating Tab Bar Items

- Title and image

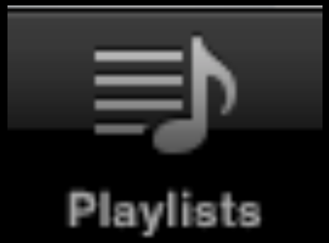
Creating Tab Bar Items

- Title and image



Creating Tab Bar Items

- Title and image



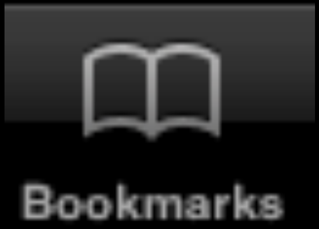
```
- (void)viewDidLoad
{
  UITabBarItem *item = [[UITabBarItem alloc]
                        initWithTitle:@“Playlists”
                        image:[UIImage imageNamed:@“music.png”]
                        tag:0];
  self.tabBarItem = item;
  [item release];
}
```

Creating Tab Bar Items

- System item

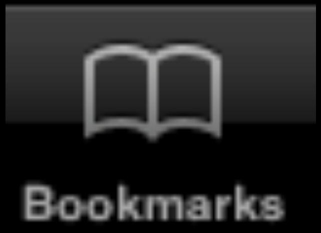
Creating Tab Bar Items

- System item



Creating Tab Bar Items

- System item

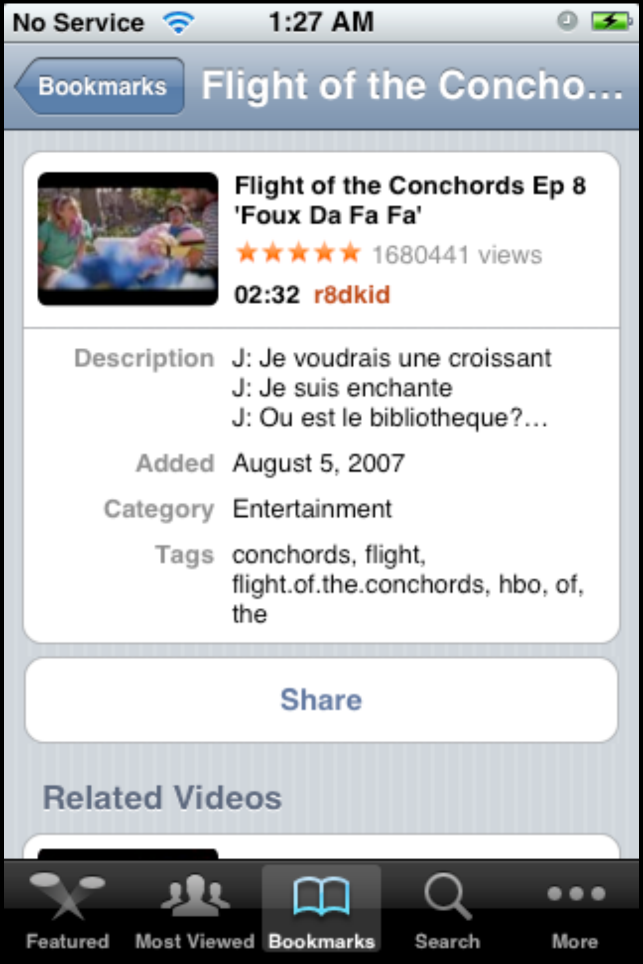


```
- (void)viewDidLoad
{
  UITabBarItem *item = [[UITabBarItem alloc]
                      initWithTabBarSystemItem:
                        UITabBarSystemItemBookmarks
                      tag:0]
  self.tabBarItem = item;
  [item release];
}
```

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?



Flight of the Conchords Ep 8 'Foux Da Fa Fa'

★★★★★ 1680441 views

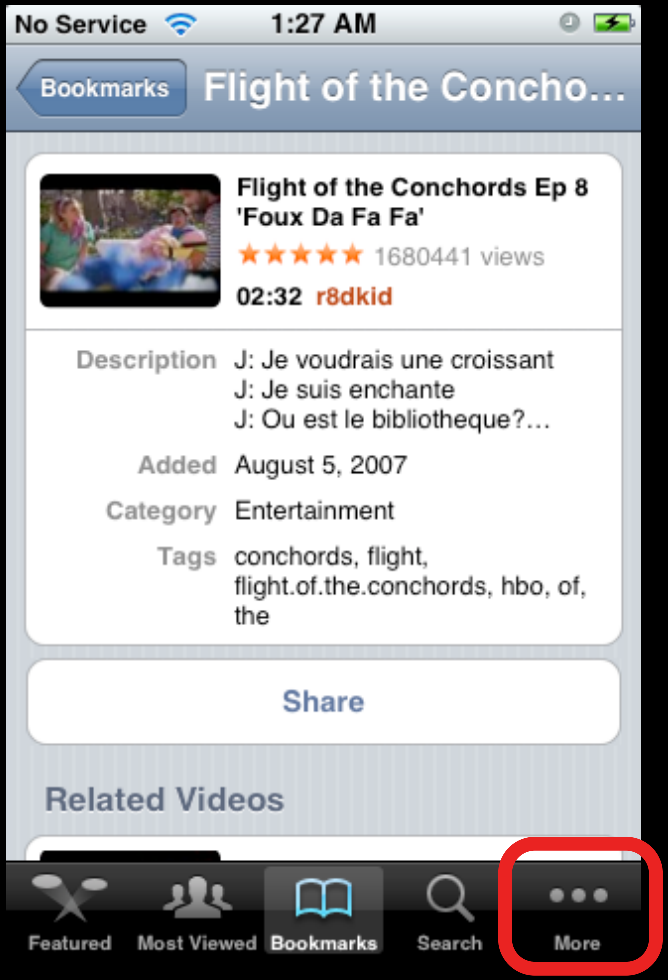
02:32 r8dkid

Description	J: Je voudrais une croissant J: Je suis enchante J: Ou est le bibliotheque?…
Added	August 5, 2007
Category	Entertainment
Tags	conchords, flight, flight.of.the.conchords, hbo, of, the

Related Videos

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



No Service 1:27 AM

Bookmarks Flight of the Concho...

Flight of the Conchords Ep 8 'Foux Da Fa Fa'
★★★★★ 1680441 views
02:32 r8dkid

Description J: Je voudrais une croissant
J: Je suis enchante
J: Ou est le bibliotheque?...

Added August 5, 2007

Category Entertainment

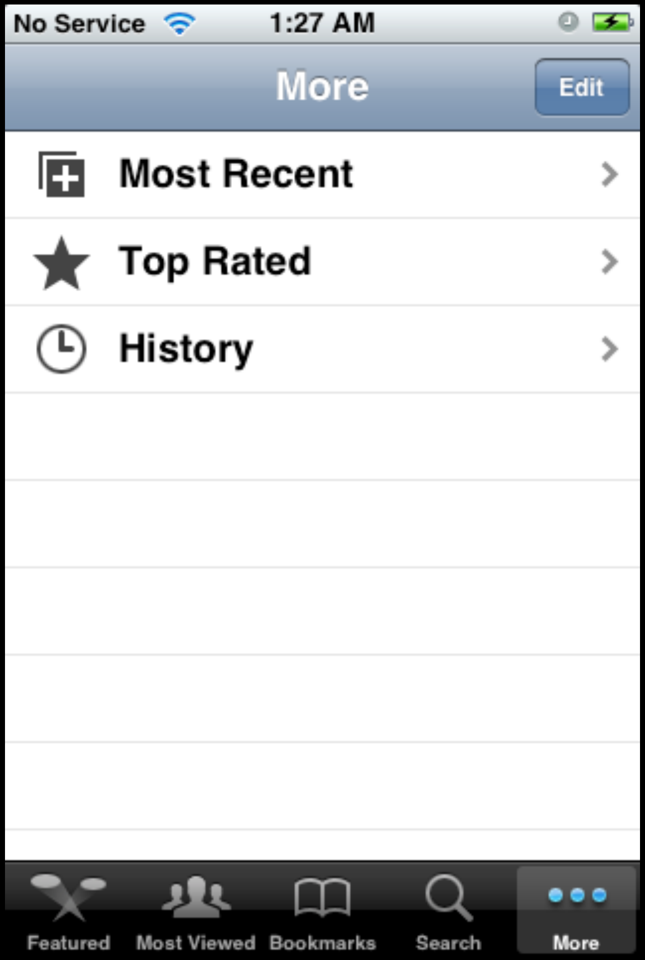
Tags conchords, flight, flight.of.the.conchords, hbo, of, the

Share

Related Videos

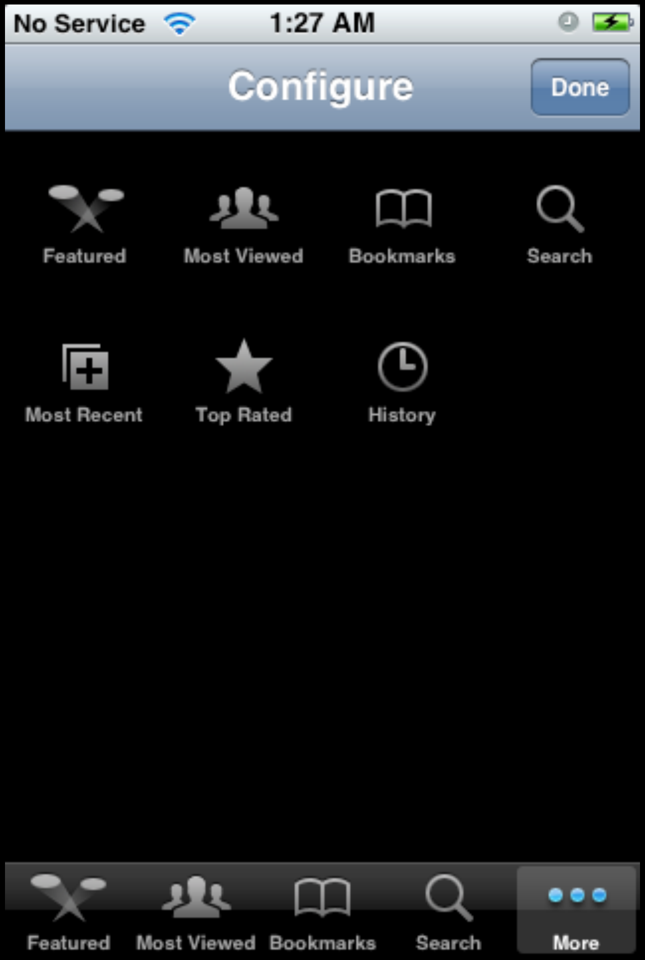
More View Controllers

- 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

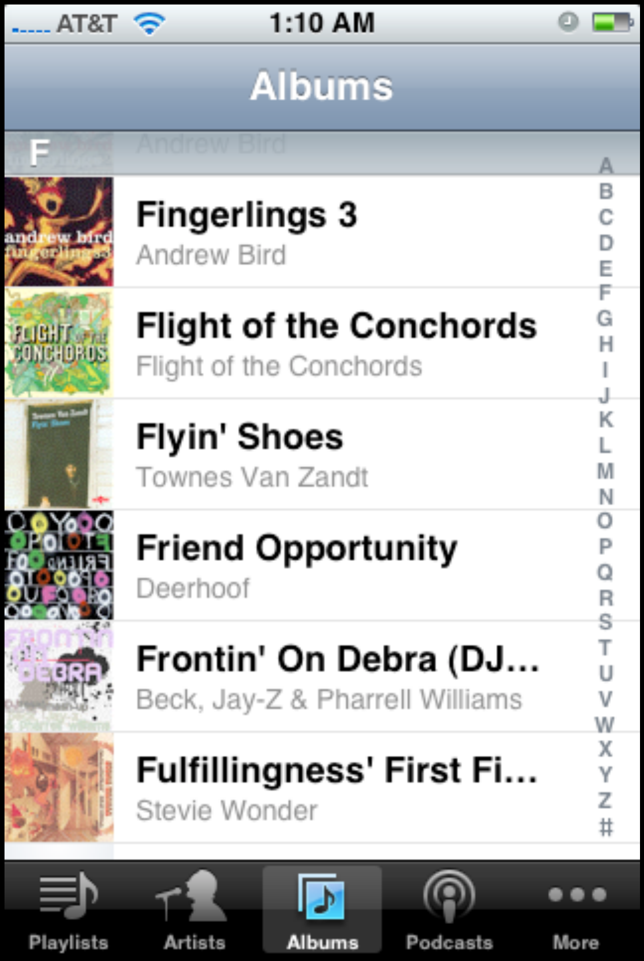
- What happens when a tab bar controller has too many view controllers to display at once?
  - "More" tab bar item displayed automatically
  - User can navigate to remaining view controllers
  - Customize order



Combining Approaches

Tab Bar + Navigation Controllers

Multiple parallel hierarchies



Fingerlings 3
Andrew Bird

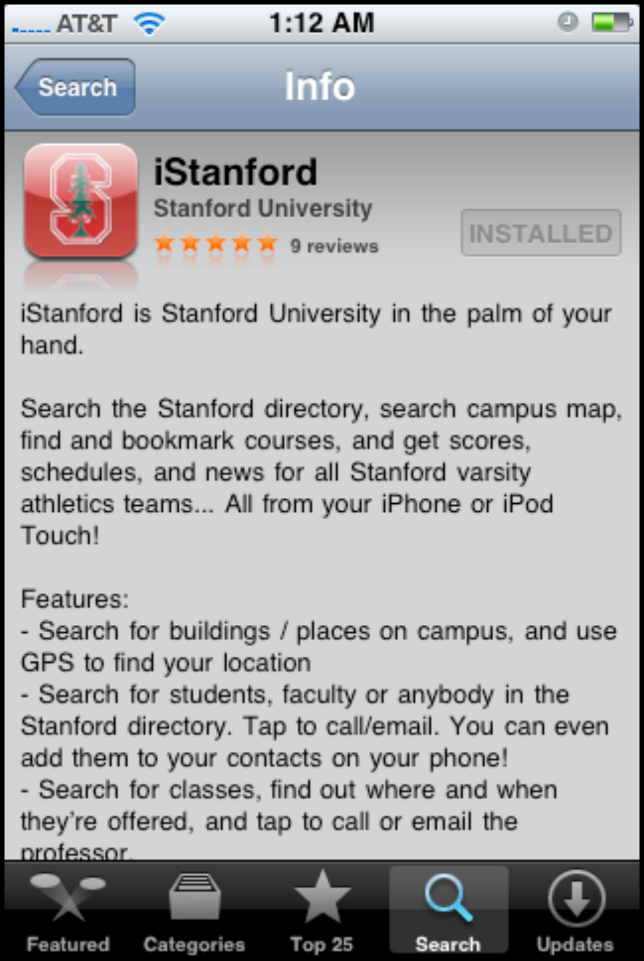
Flight of the Conchords
Flight of the Conchords

Flyin' Shoes
Townes Van Zandt

Friend Opportunity
Deerhoof

Frontin' On Debra (DJ...
Beck, Jay-Z & Pharrell Williams

Fulfillingness' First Fi...
Stevie Wonder



iStanford
Stanford University
★★★★★ 9 reviews

iStanford is Stanford University in the palm of your hand.

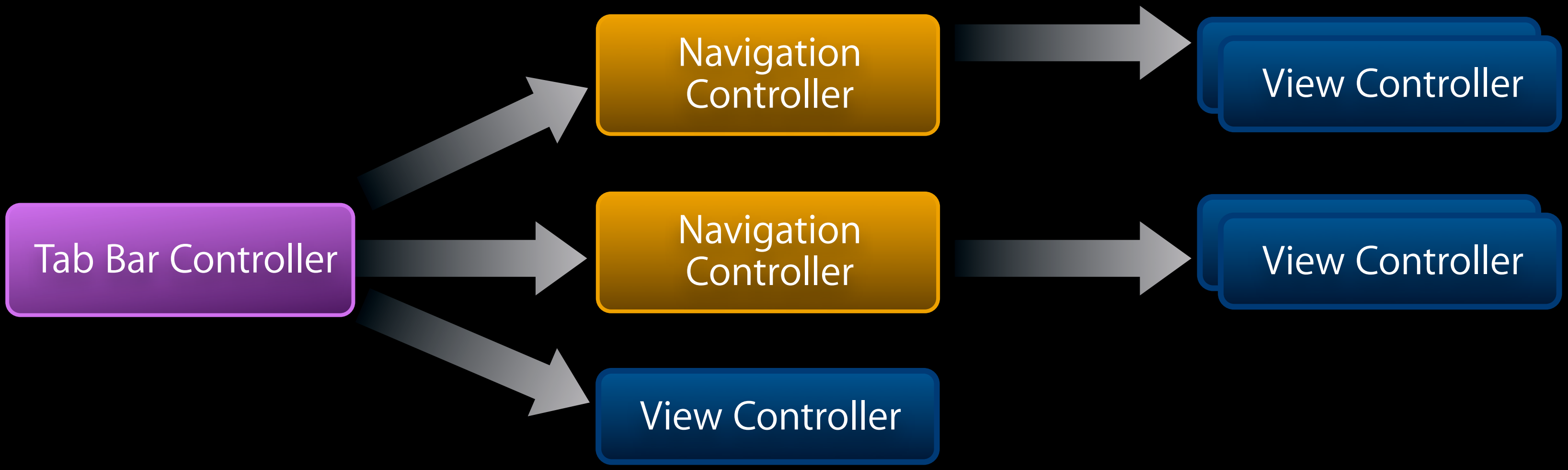
Search the Stanford directory, search campus map, find and bookmark courses, and get scores, schedules, and news for all Stanford varsity athletics teams... All from your iPhone or iPod Touch!

Features:
- Search for buildings / places on campus, and use GPS to find your location
- Search for students, faculty or anybody in the Stanford directory. Tap to call/email. You can even add them to your contacts on your phone!
- Search for classes, find out where and when they're offered, and tap to call or email the professor

Tab Bar + Navigation Controllers

Tab Bar Controller

Tab Bar + Navigation Controllers



Nesting Navigation Controllers

Nesting Navigation Controllers

- Create a tab bar controller

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

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];
```

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?