Mobile App Development Crash Course
Natasha Murashev, @natashatherobot
natasha@natashatherobot.com

GENERAL ASSEMBLY

Welcome To iOS Development!

You check your iPhone multiple times a day, and so do over **575 million** other people. This tutorial will help you get started on your journey to becoming an app maker.

We're going to start by building a Resume app. Imagine showing that to employers!

The Resume app will consist of four sections:

- **About Me**: Share your story.
- **Social Links**: Make it easy for others to follow you on LinkedIn, Twitter, Facebook, Dribbble, Github, and more.
- **Contact Me**: Make it easy for employers to contact you without exposing your email address.

Now that we got that out of the way, let's get started!

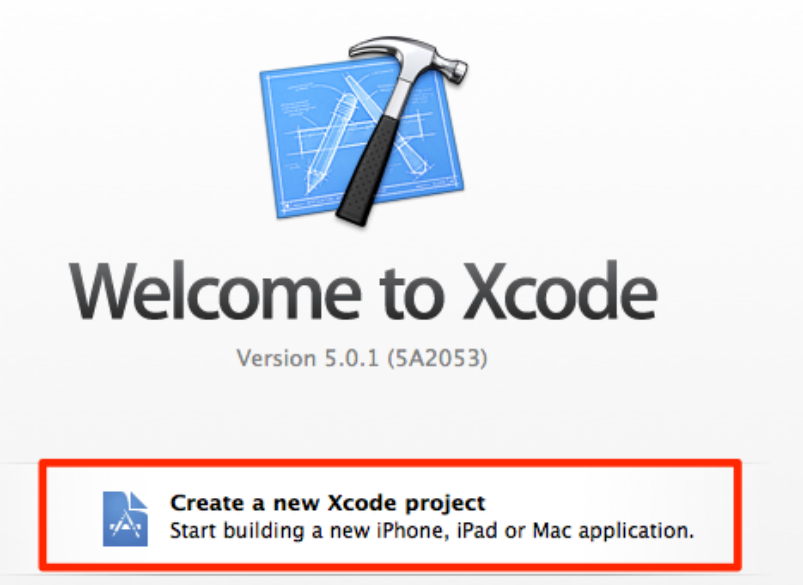
GENERAL ASSEMBLY

Getting Started

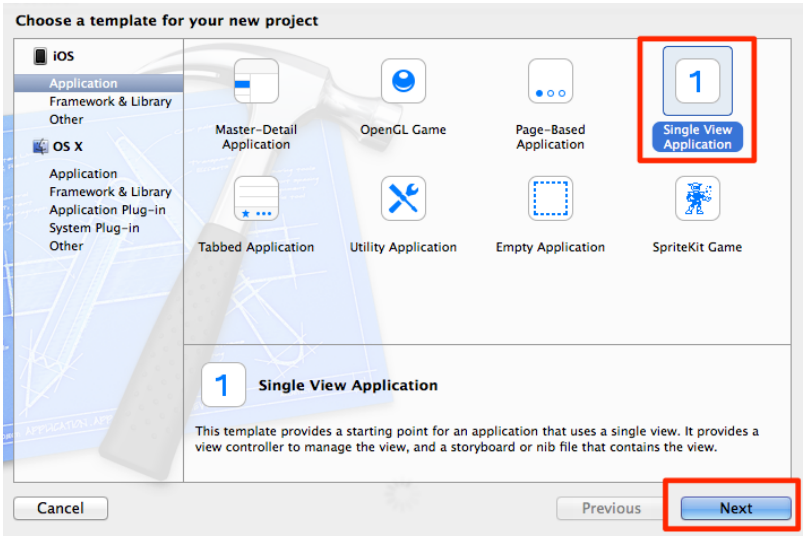
If you do not have XCode 5 install, install it first via this link.

Now, let's create our first project!

- Open XCode
- Select **Create a new Xcode project**

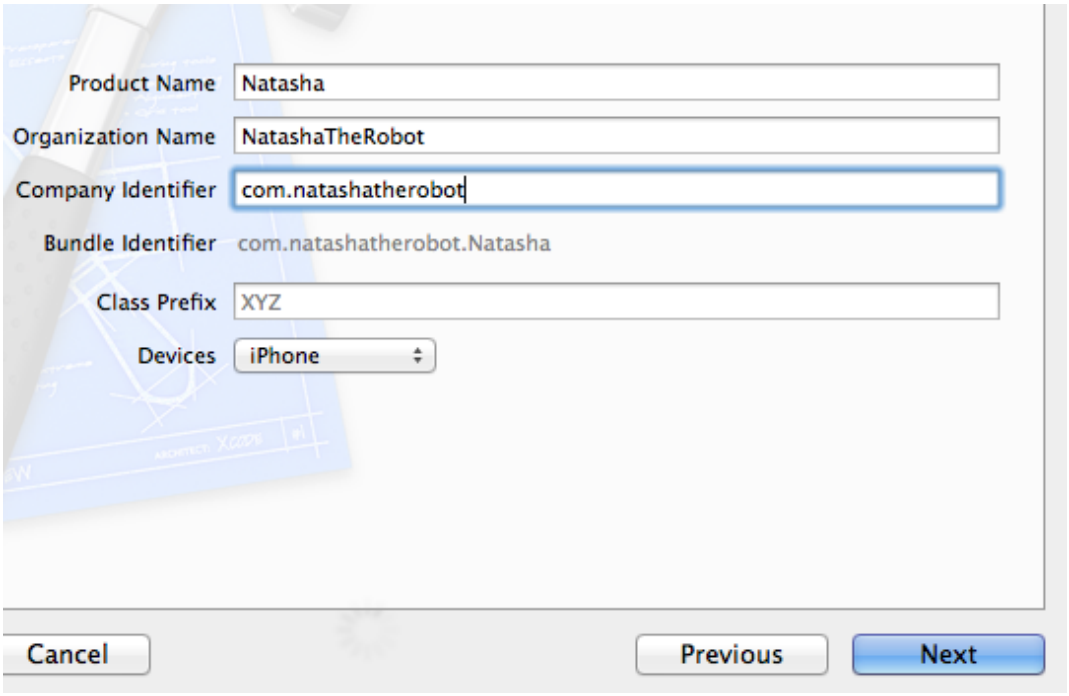


- Select **Single View Application** and click **Next**



GENERAL ASSEMBLY

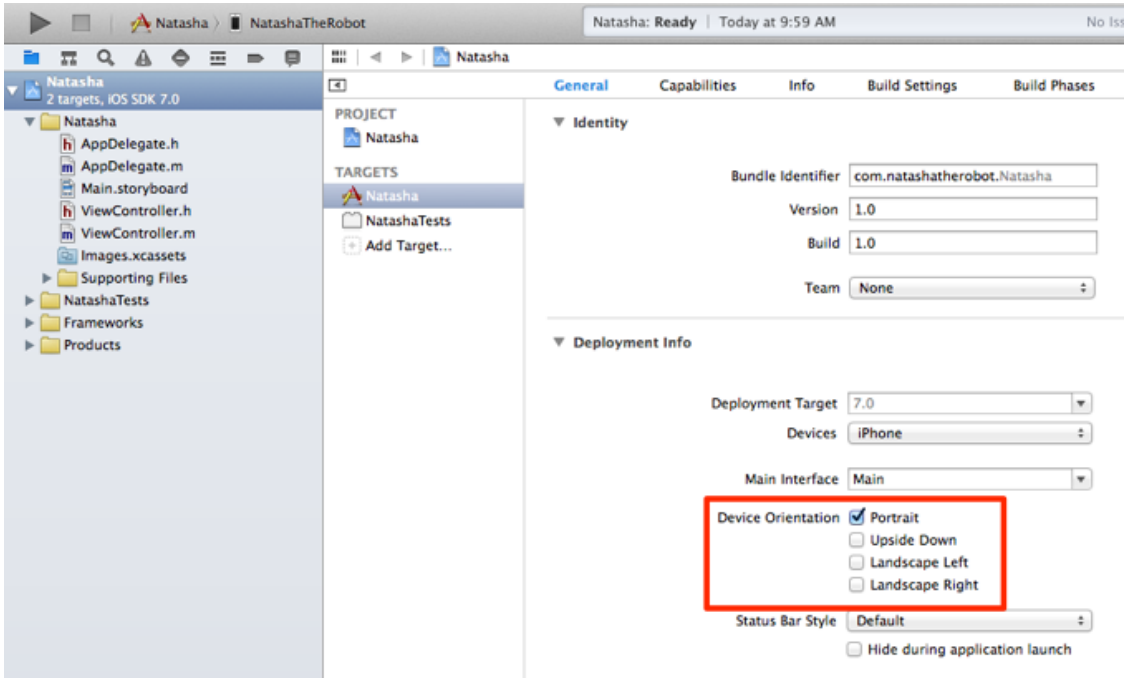
- Fill out your information and click Next:
  - **Product Name**: We're building your resume, so this is your name :)
  - **Organization Name**: This could be the name of your business, or your name
  - **Company Identifier**: This needs to be unique. The convention is to use your domain ending (e.g. com) followed by the domain name or your name, so **com.yourname**.
  - **Class Prefix**: You can **leave this blank** or add a 2-3 letter prefix. For my company, NatashaTheRobot, for example, the prefix would be NTR. This prefix will be added to all your project files by default, so let's leave it blank for this tutorial to avoid confusion.
  - **Devices**: Make sure iPhone is selected. In the future, you can make apps for iPad only by selecting the iPad option or for iPad and iPhone by selecting the Universal option.



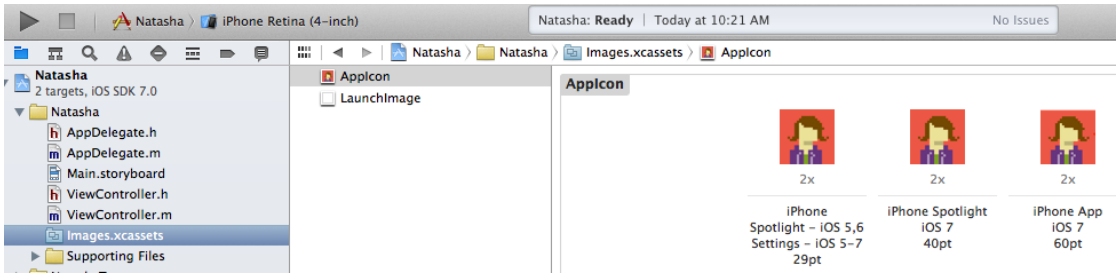
- Select a folder to save your project in and **Create**!

GENERAL ASSEMBLY

- Our app is only going to work in Portrait mode as a start, so **uncheck the Landscape Left and Landscape Right** options.



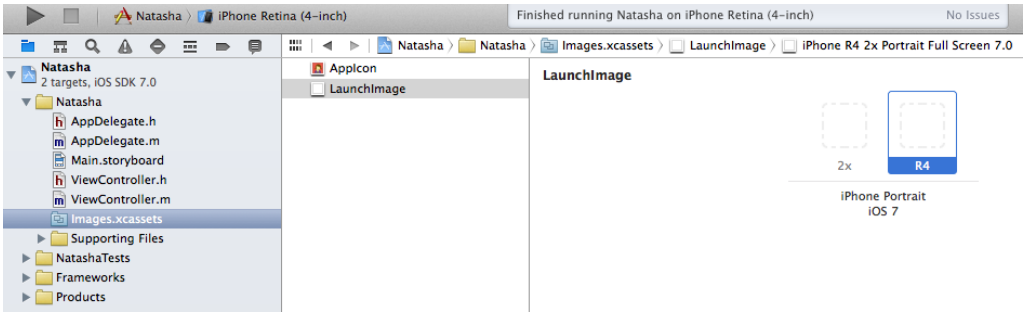
- Add an App icon!
  - Go to the **Images.xcassets** folder. This is where all your future project images will be managed from.
  - Select AppIcon
  - Drag in a photo of you at double the sizes (for retina displays). So your image needs to have a 120x120 version, an 80x80version, and a 58x58 version.



- Click the big Play button to run the application in the **iPhone Retina (4-inch)** simulator

- The project should successfully run, and all you'll see is a white screen.
- Use the **Command + Shift + H** keyboard shortcut to go to the simulator home screen, and see your app icon in action!

- **Homework**: The LaunchImage is the first image your users see when the app launches. Read the Launch Image section of Apple's iOS Human Interface Guidelines and add a LaunchImage to your app:

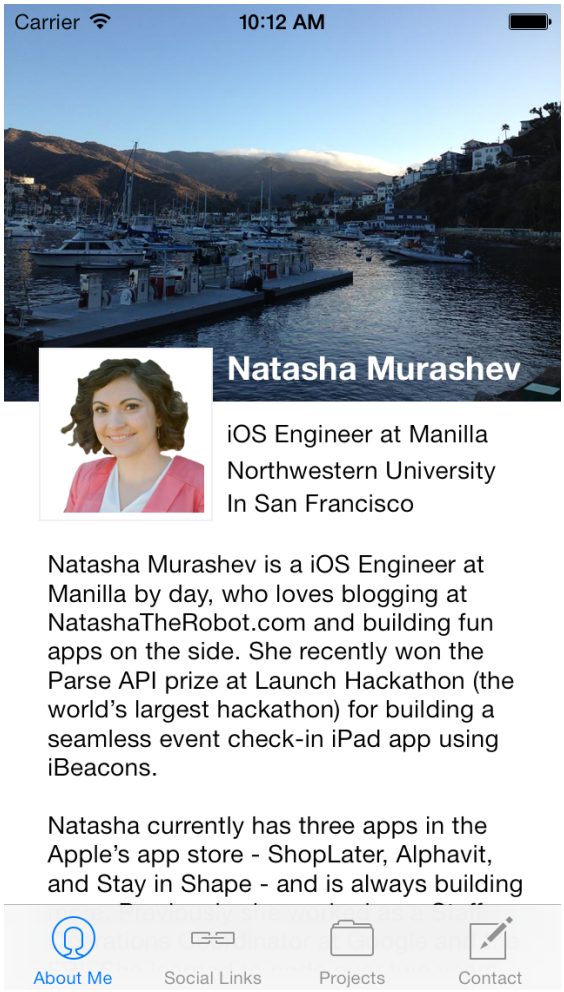


Congratulations, you've created your first iPhone project! Now onto the fun stuff ;)

GENERAL ASSEMBLY

About Me

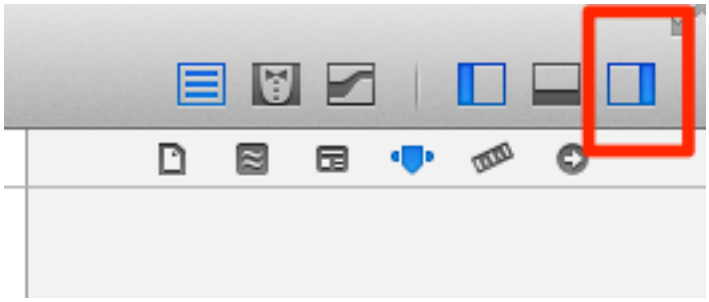
We're going to start by adding the about me tab first. The end result will look something like this:



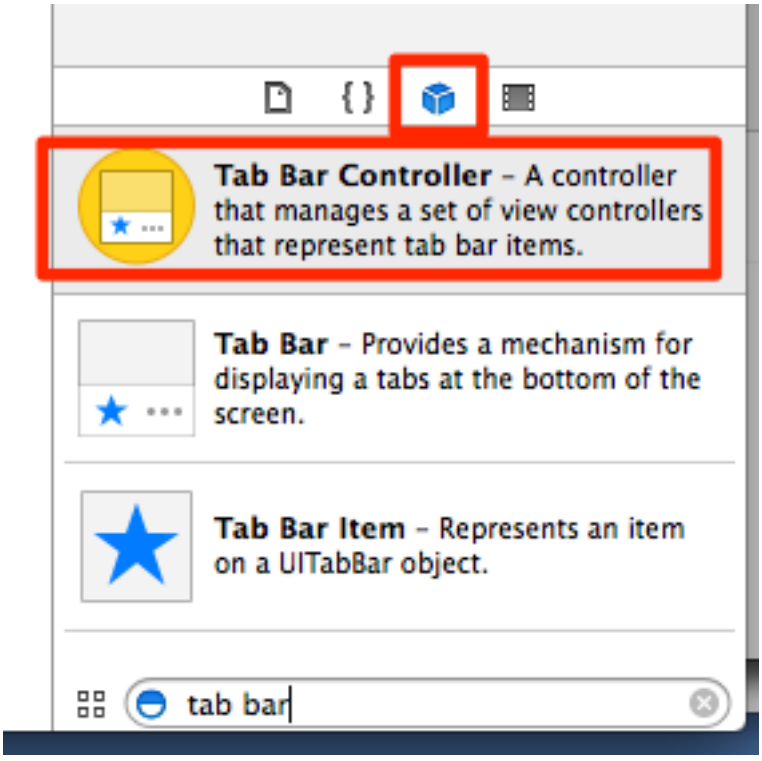
- Open the **Main.storyboard** file. This is where you're going to work on the visual design of your application, and you'll see a View Controller there.
- A View Controller is the logic that controls the user interface. For example, a button is a View, but when the user touches the button, the interaction that happens as a result is directed by the View Controller.
- **Homework**: Read more about View Controllers in the iOS Developer Library.
- Delete the View Controller that is already included in the StoryBoard. We're going to start the project with the Tab View Controller, since we'll have 4 tabs.

Mobile App Development Crash Course
Natasha Murashev, @natashatherobot
natasha@natashatherobot.com

- Open the right pane of XCode:



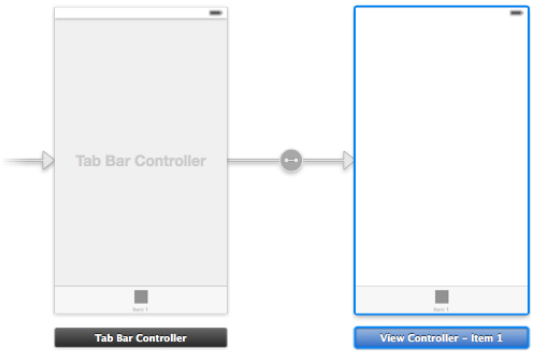
- Find the Tab Bar Controller



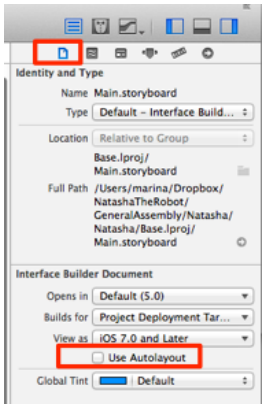
- Drag the Tab Bar Controller onto the Storyboard. By default, it comes with two View Controllers. For now, delete one of those two view controllers, so we could focus only on the About Me tab.

GENERAL ASSEMBLY

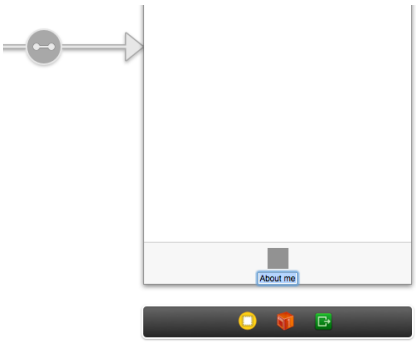
- Your Storyboard should now look like this:



- In the right-hand pane, find the Identity and Type Section, and **turn off Use Autlolayout** - we're not going to be working with AutoLayout in this workshop:

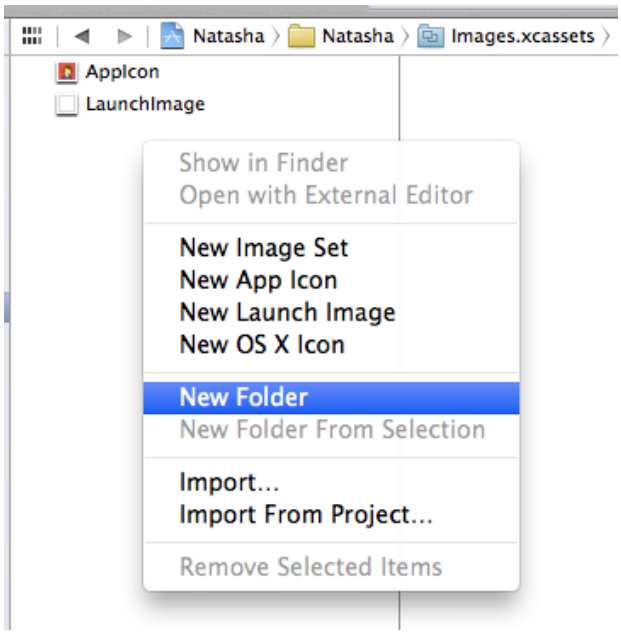


- Tap on the "Item 1" tab on your View Controller, and change the title to "About me"

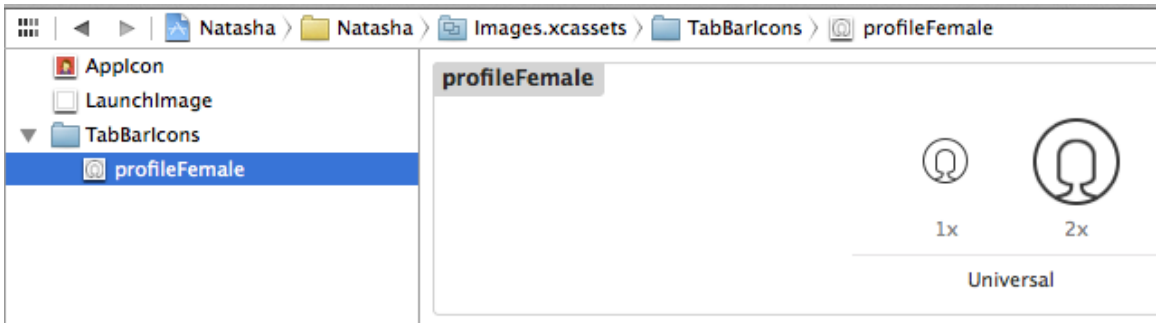


GENERAL ASSEMBLY

- In your Images.xcassets folder, add a TabBarIcons Folder (since we'll have four tabs):

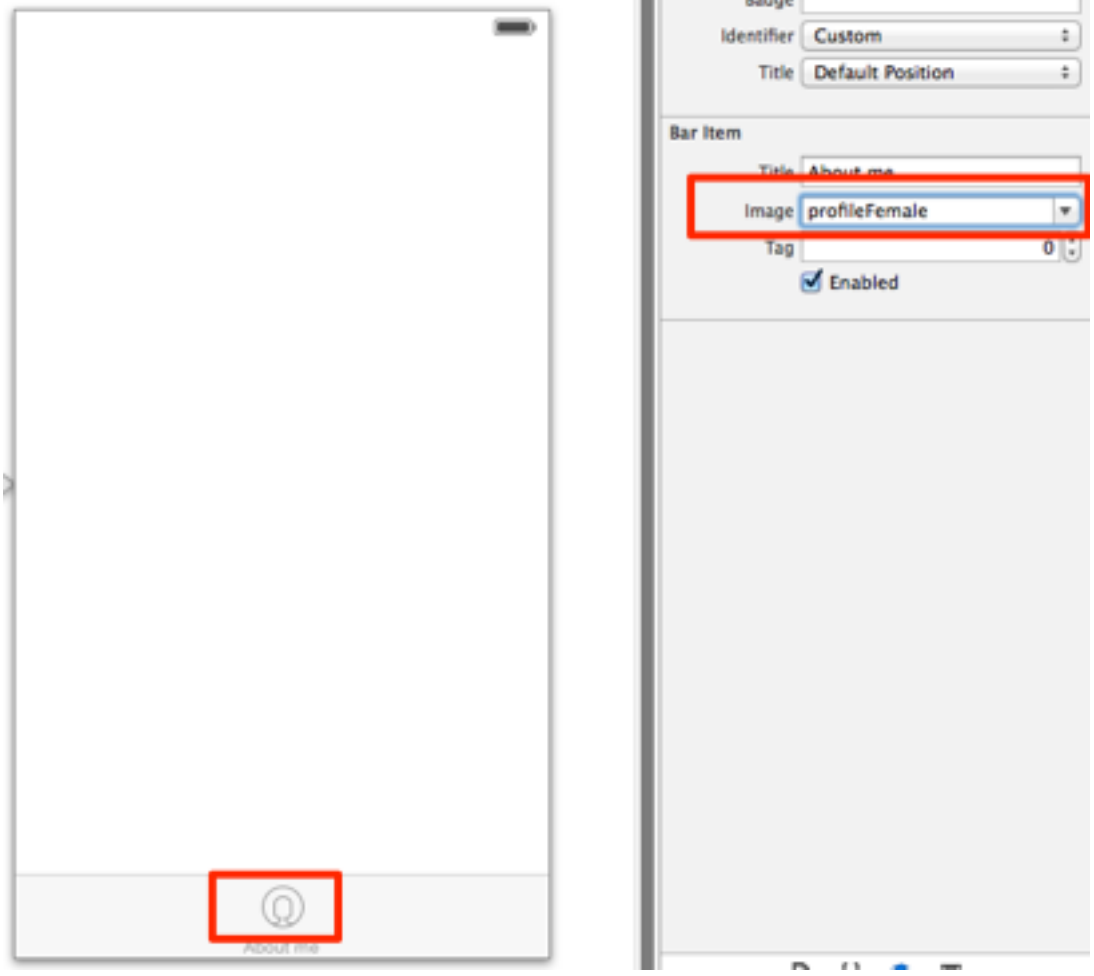


- Drag the provided male or female profile icon into the folder. This will populate one of the two necessary sizes of icons. Drag in the second one in the 2x or 1x position. The 2x icon image is needed for retina display:



- Go back to the Storyboard file.

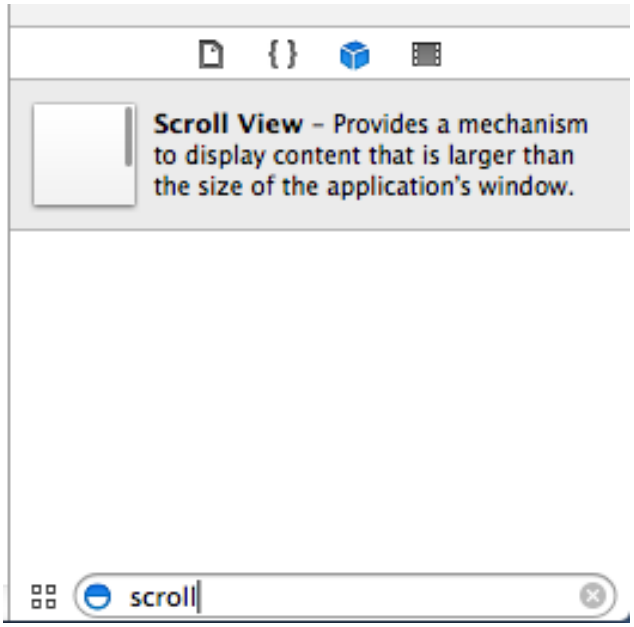
- Select the Tab Bar Icon on your About Me View Controller, and find the appropriate image name for the Icon. The correct image should now be displayed on the Tab Bar!



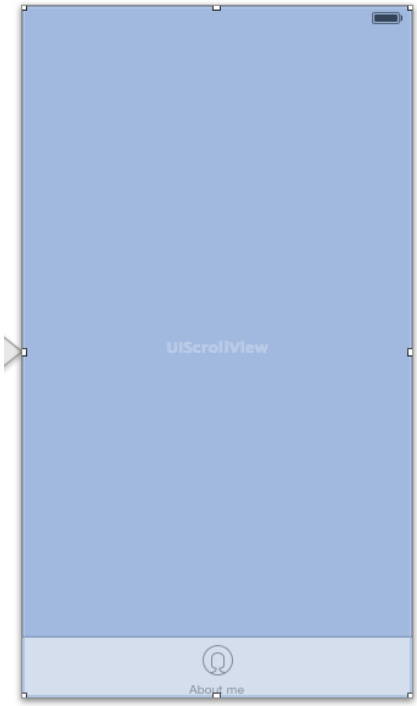
- Since we're going to have lot's of content on this page, we'll need to add a Scroll View (so the content is scrollable!).

GENERAL ASSEMBLY

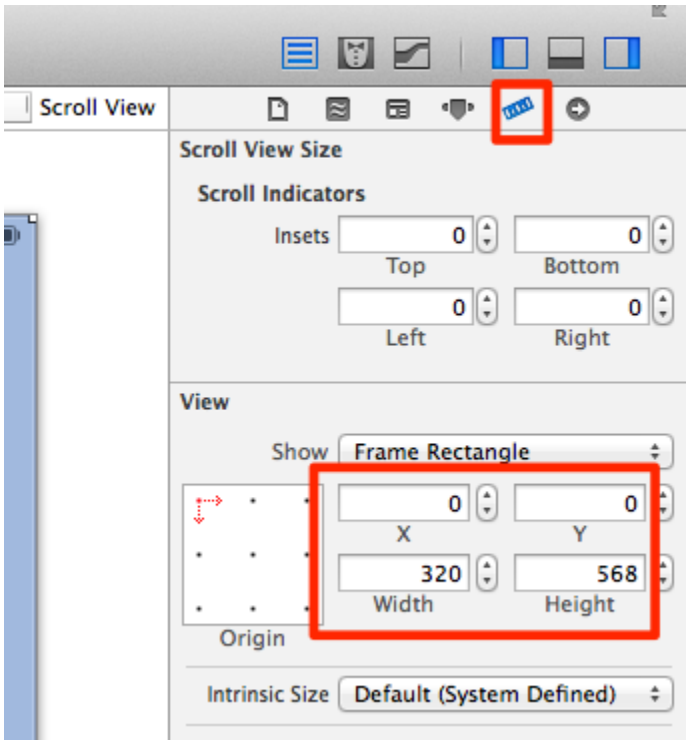
- Find the **Scroll View** in your right pane:



- Drag it onto your About Me View Controller



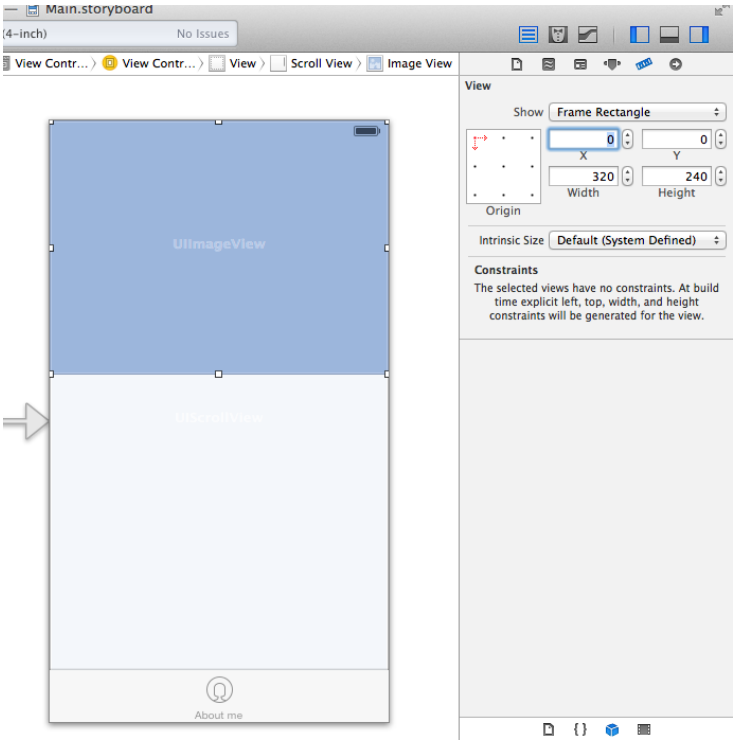
- Make sure it takes up the whole space of the View Controller. To check, go to the Ruler section of your Right hand pane, and make sure the Frame has the **position of 0, 0, with the width of 320 and height of 568**. If not, change it to that position and size.



- Find an **Image View** in the right pane - since you've done this already for the Tab Bar View Controller and the Scroll View, I won't put the exact instructions on how to do this. Ask me if you're having trouble!

GENERAL ASSEMBLY

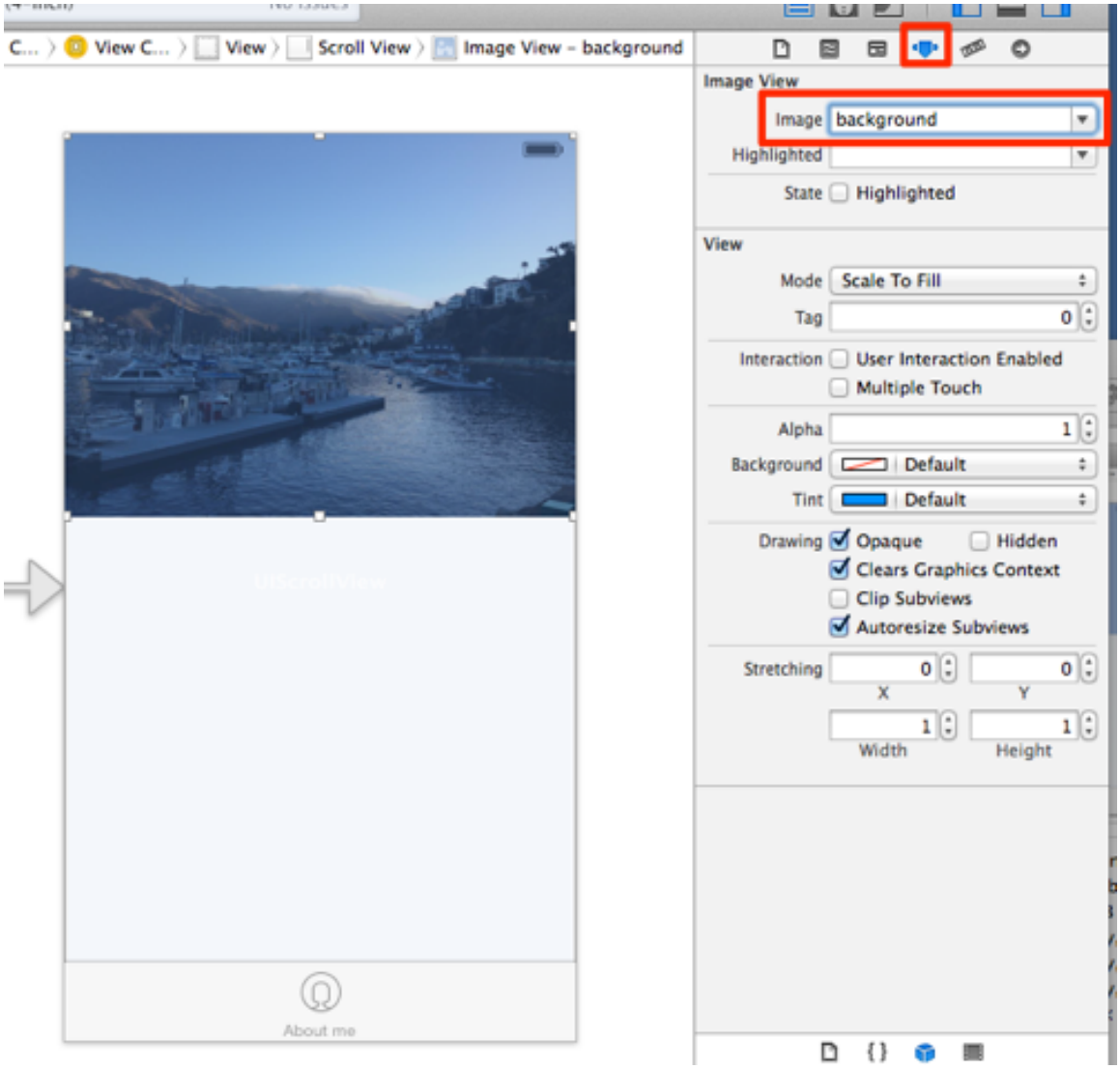
- Drag the Image View onto the Scroll View, at **position 0,0** with the **width of 320 and height 240**. It should look like this:



- In your Images.xcassets folder, add an About folder, and drag in a **320 x 240 image named background.png** into it. You'll also have to **add an image that is double the size 640x480 for retina display** - the convention is to **call this image background@2x.png**.
- In the Storybard, select the Image View on your About Me View Controller, and select the Tool Belt Icon on the right side pane.

GENERAL ASSEMBLY

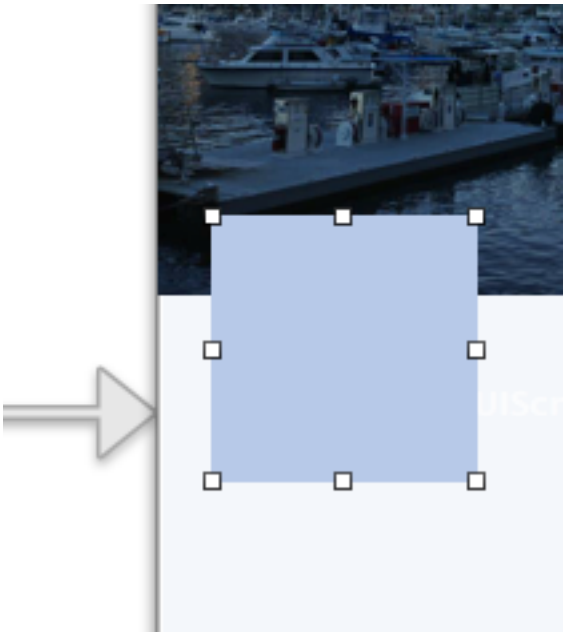
- Select the name of your background image in the Image View's image section. The image should show up on your About Me View Controller.



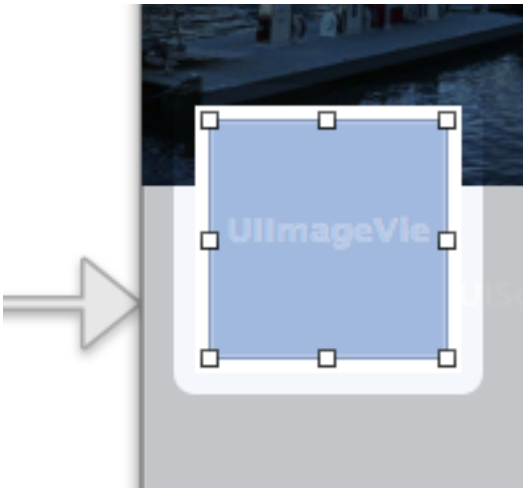
- Find a View in the right hand pane and drag it onto the ImageView - this is the border for the image of your photo.

View – Represents a rectangular region in which it draws and receives events.

- The View should be at **position 20, 210**, with the **height and width of 100**. Feel free to play with the position and height / width if you think it'll look better differently. You can always drag it around in your About Me View Controller to see what it'll look like!

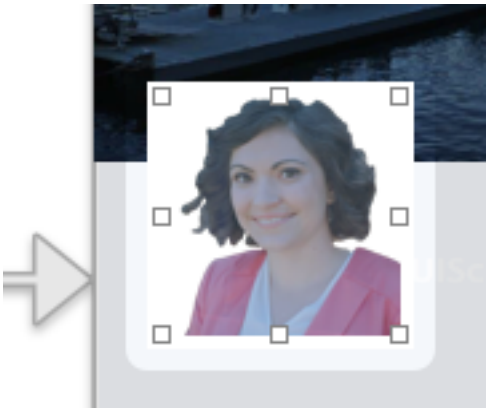


- Add an Image View **inside the View**, at **position 5,5**, (note that this position is relative to the container view we just added, not the entire view!), with **width and height 90, 90** leaving a 5px white border around the image.

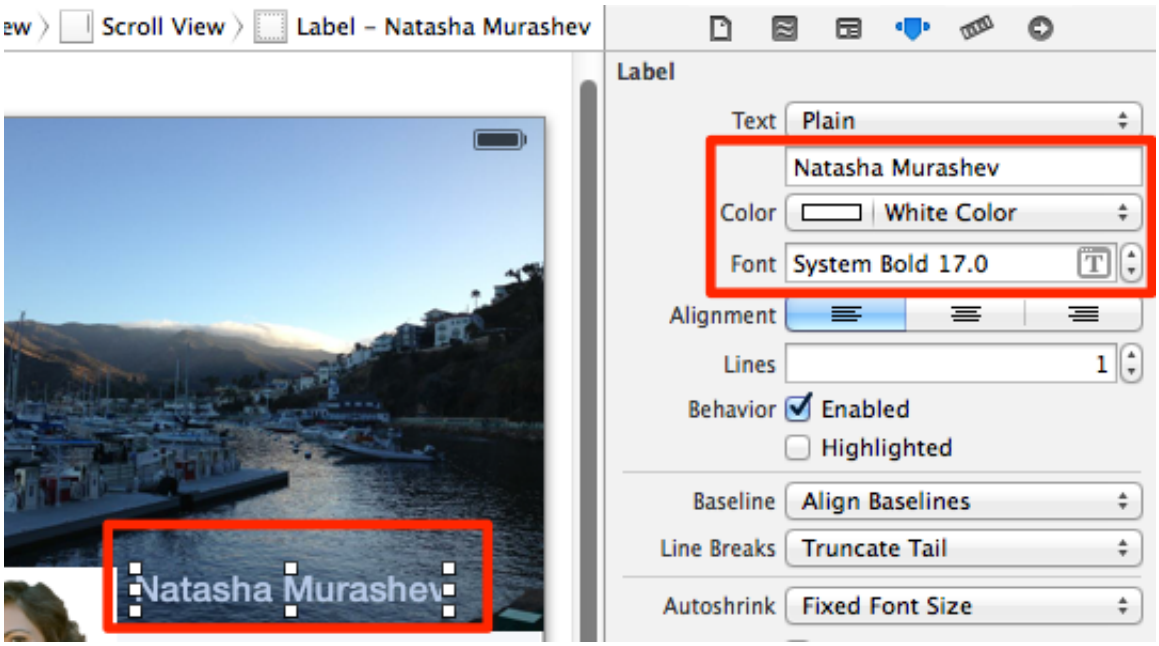


GENERAL ASSEMBLY

- Add an image called **profilePhoto.png** of yourself into the Images.xcassets folder, inside the About folder. There should be one image of **size 90x90** and one at **180x180** for retina display.
- In the storyboard, select the Tool Belt section of the right hand pane, and set the profilePhoto image to the the 90x90 Image View you just dragged on.

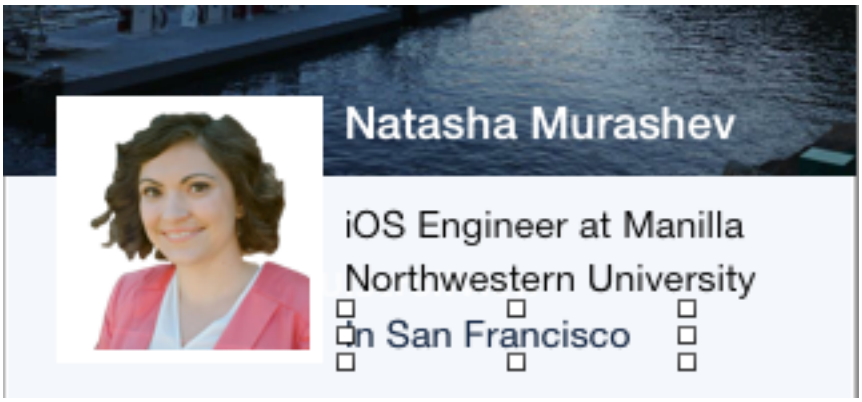


- Find a Label in the right hand pane, and drag it on to the background image, next to View with your profile photo at position 128, 210 (210 is the Y position of the View, so they will be top-aligned).
- In the Tool Belt section of the right hand pane, change the Text of the Label to your Name, change the text color to be White, and change the Font to be System Bold 17.0.

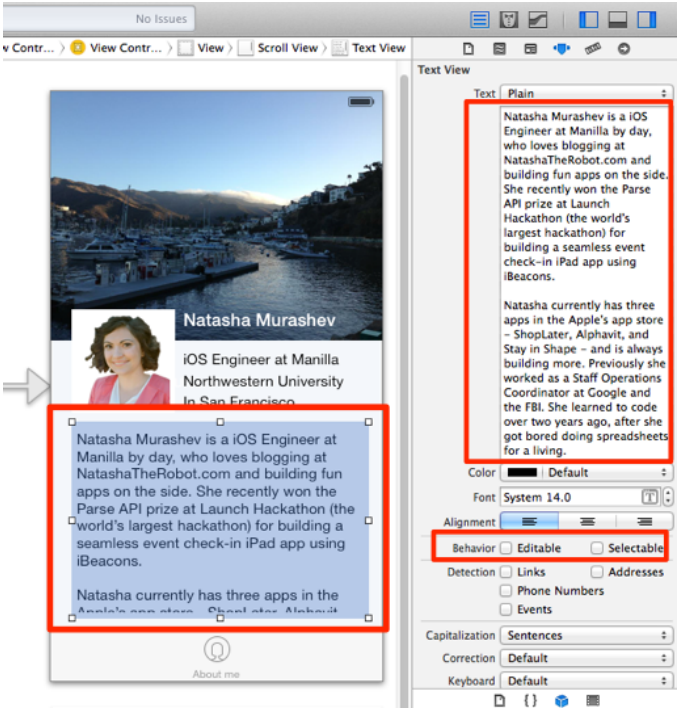


GENERAL ASSEMBLY

- Add three more labels with System 14.0 Font underneath your name label to add a quick summary about yourself.

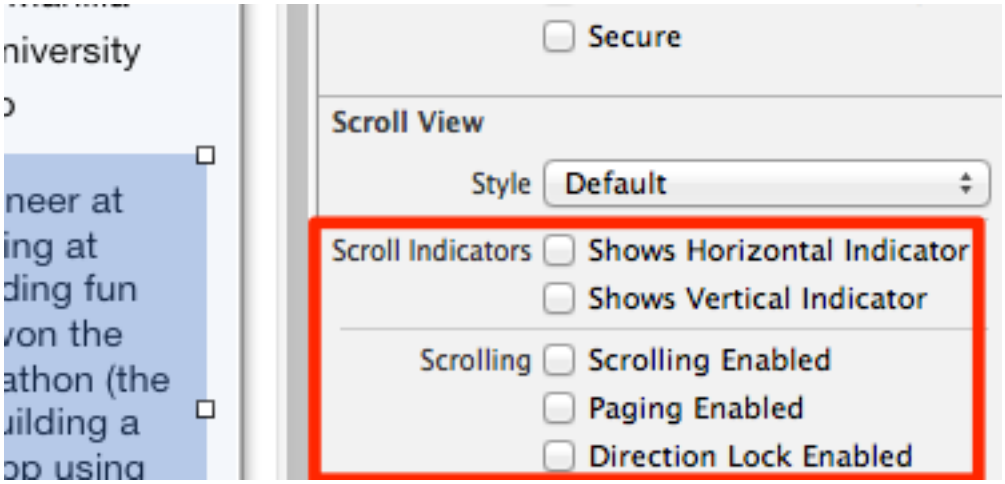


- **Homework:** Play around with the Label's Tool Belt options to see what they do! Try to align the text in the label to the center, change the background color of the label's view, add a shadow, change fonts, etc.
- Find a Text View and Drag it in under the View with your profile photo and the three labels. This is where you will write more about yourself.
- In the Text View's Tool Belt Section, write (or copy / paste) a summary about yourself. **Turn off the "Editable" and "Selectable" options**, so this behaves like text, not an editable / selectable text field.



Mobile App Development Crash Course
Natasha Murashev, @natashatherobot
natasha@natashatherobot.com

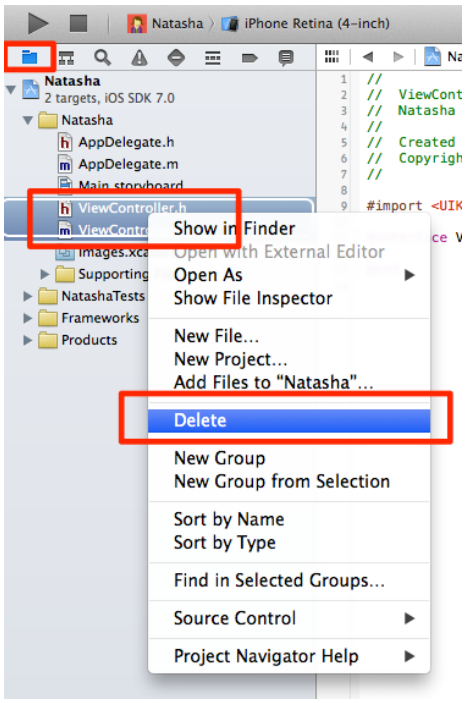
- In the Text View's Tool Belt Section, scroll down and **turn off scrolling** - we want the whole Scroll View to scroll, not the Text View individually:



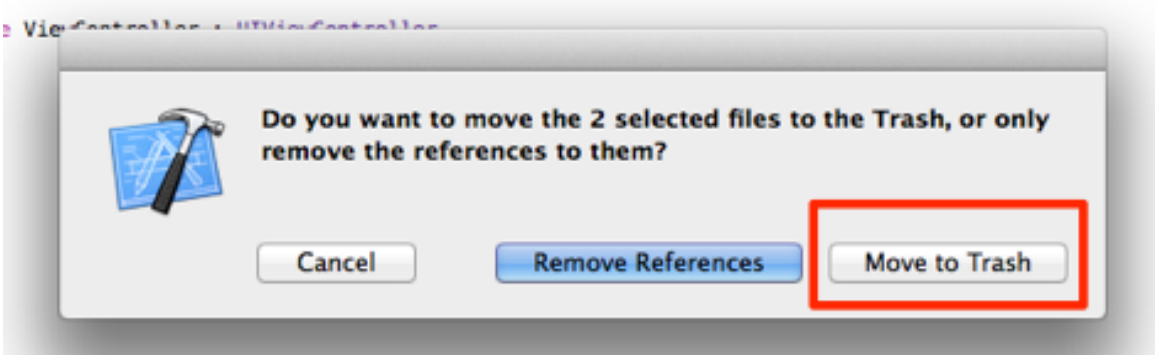
- In the Text View's Ruler section, expand the height of the Text View so all the text you have fits. The user will be able to scroll down to see all of it. For my two paragraphs, I put the height at 320.
- Run the project in your Simulator, and try scrolling. It doesn't work! That's because we have to specify how long the Scroll View is in our code throughout the ViewController.
- Find the Left Side Pane of Xcode. If you can't see it, you can find it by clicking the Left Pane Button on the top right:

- Find the Folder section.

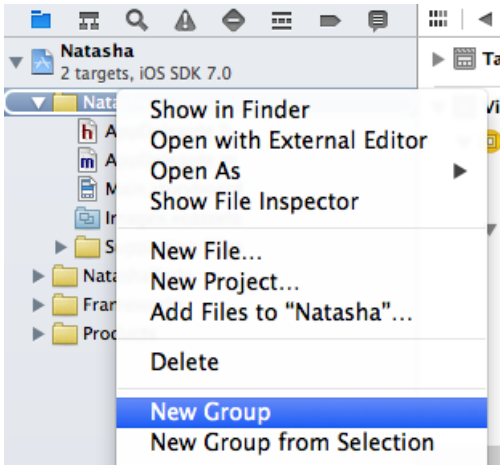
- The should be a system-generated ViewController.h and ViewController.m files. Delete these (we'll generate our own!):



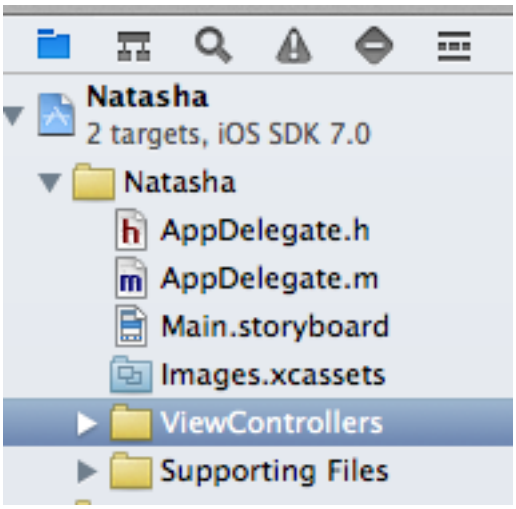
- Make sure to select the **Move to Trash** option when the delete pop-up comes up! If you just remove the reference, the file won't show up in your XCode project directory, but it'll still be there in the actual folder.



- Create a New Group called **ViewControllers** - this is where all your ViewControllers will be stored in the future.

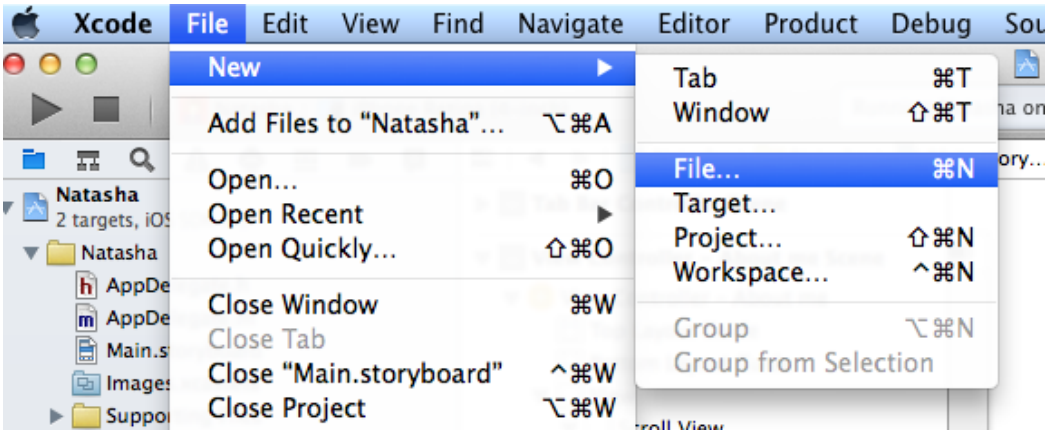


- You can drag it down just above the Supporting Files Group:

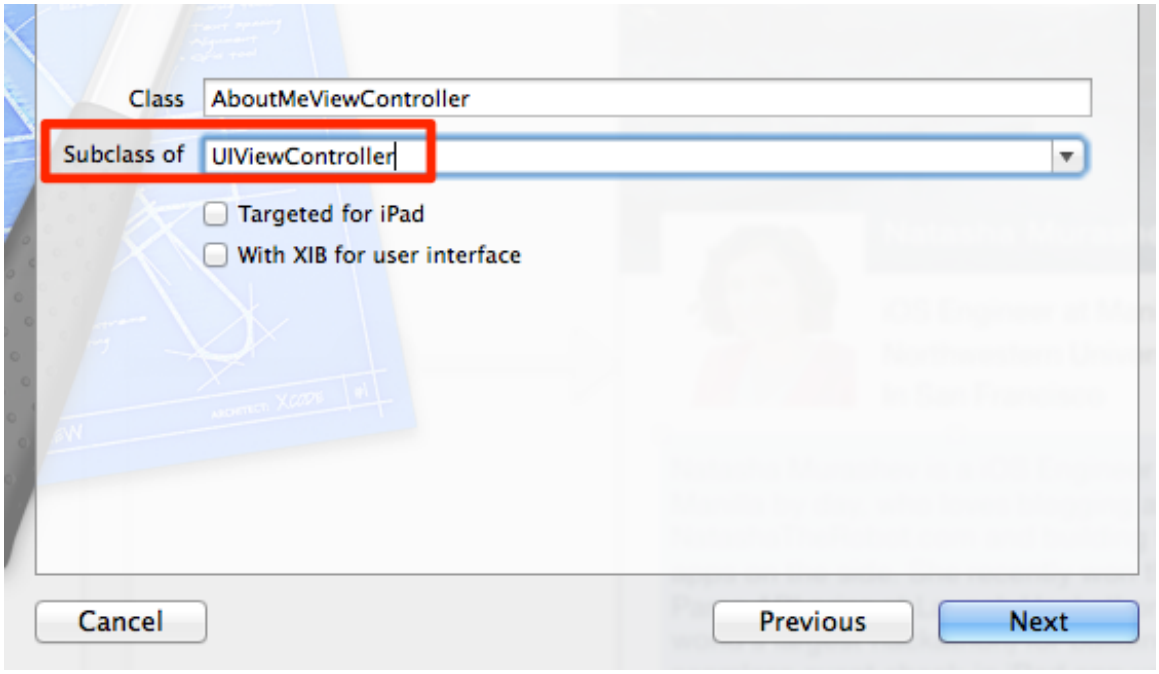


Mobile App Development Crash Course
Natasha Murashev, @natashatherobot
natasha@natashatherobot.com

- To create our AboutMeViewController, go to File -> New -> File…

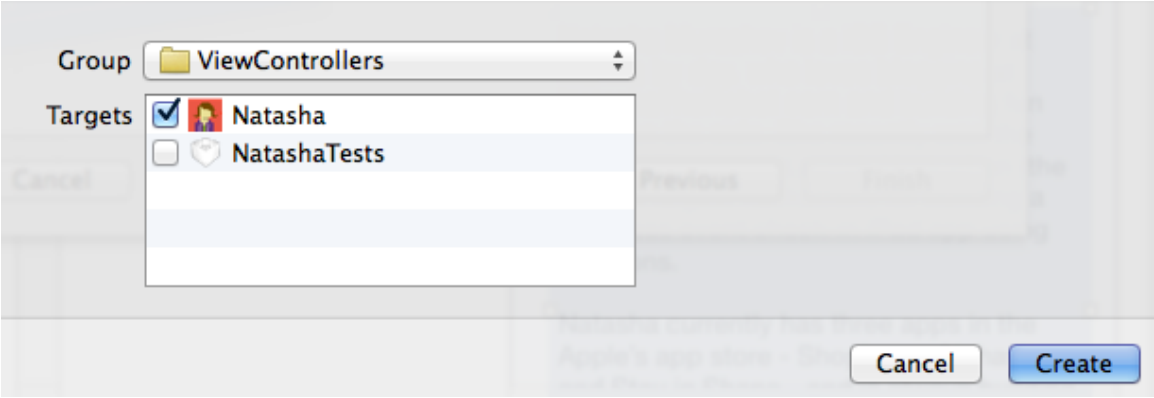


- Create an **AboutMeViewController**, which is going to be a **subclass of UIViewController**. That means that AboutMeViewController will inherit all the out-of-the-box methods that come with UIViewController.

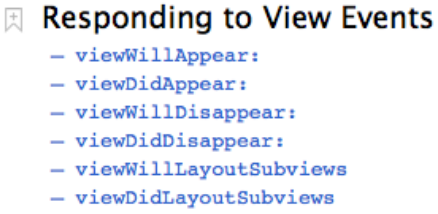


- **Click Next**.

- On the following screen, make sure that your new AboutMeViewController will be **in the ViewController's group**, and that it will be **in your project's target**:



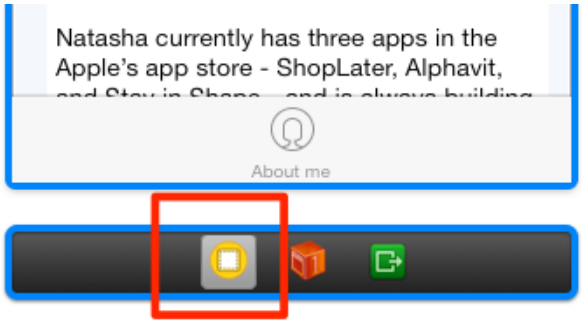
- Click **Create**.
- Open **AboutMeViewController.h**, and **Option Double-Click on the UIViewController** subclass. This will open the documentation for UIViewController. Scroll down to the **Creating a View Controller Using Nib Files section** and below to see all the available methods that come with the UIViewController! The one's you'll be using most are in the **Responding to View Events** section:



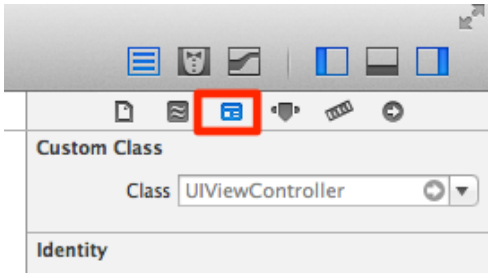
- Now, open the **AboutMeViewController.m** file. Delete all the methods besides viewDidLoad. viewDidLoad is the only method you'll need to over-write:

```
#import "AboutMeViewController.h"

@interface AboutMeViewController ()

@end

@implementation AboutMeViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
}

@end
```

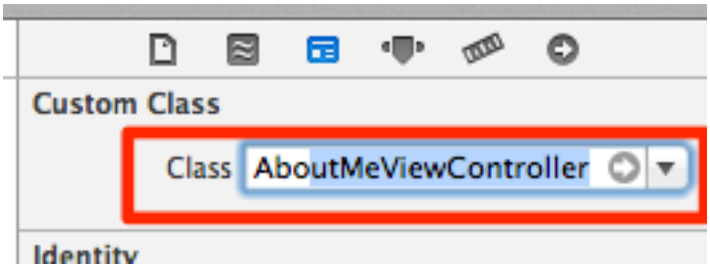
- On your Storyboard, select the AboutMeViewController by selecting the Yellow ViewController symbol underneath it:



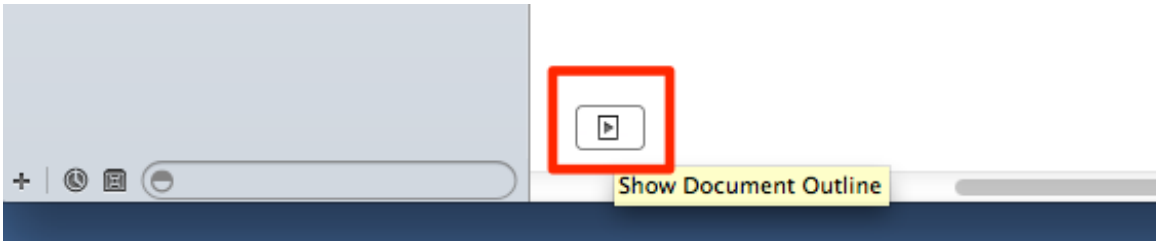
- In the right hand pane of the Storyboard, select the **Identity Section**:



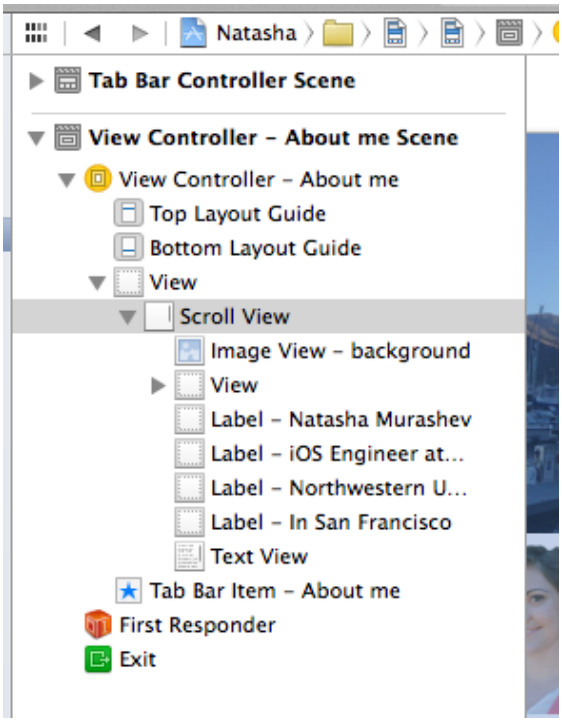
- Change the ViewController Class from UIViewController to AboutMeViewController. This links the storyboard's View that we just created with the AboutMeViewController!



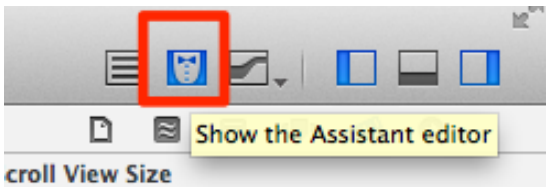
- On your Storyboard, make sure you have the **Document Outline open**. If you don't , you can open it by clicking the Open Document Outline button on the bottom left of your Storyboard:



- **Select the Scroll View** from the Document Outline:

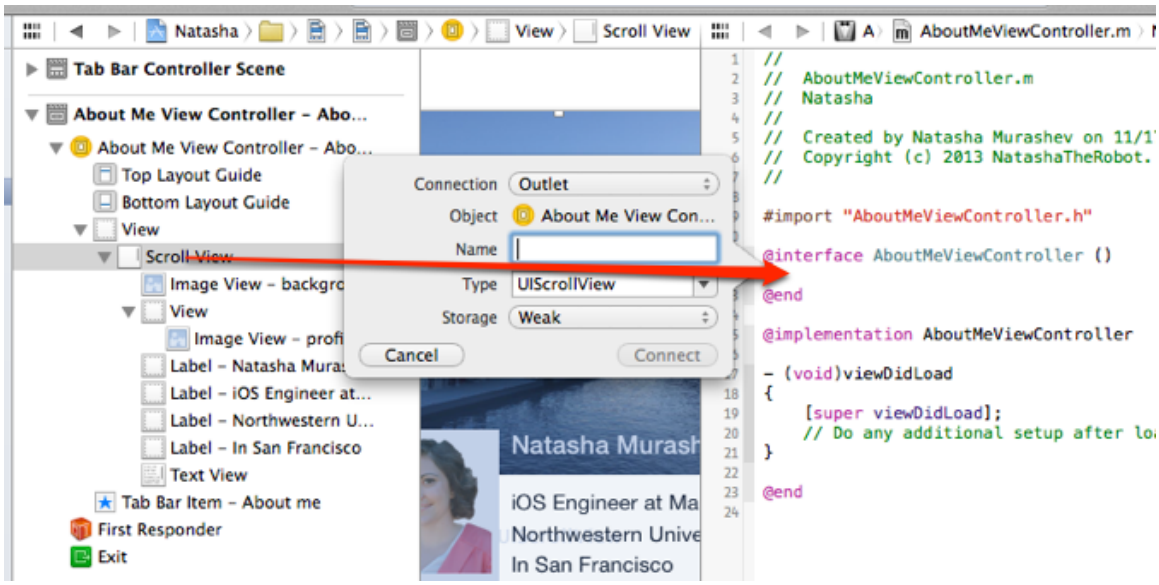


- In the Storyboard, select the AboutMeViewController, and **open the Assistant Editor**. The **AboutMeViewController.m file should be open** in the other editor.

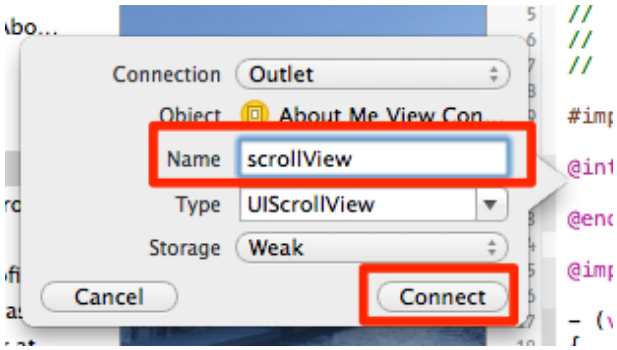


GENERAL ASSEMBLY

- **Control Drag** from the **Scroll View** in the Document Outline **into the @interface portion of the AboutMeViewController**. You'll get a pop-up like the one I have in below:



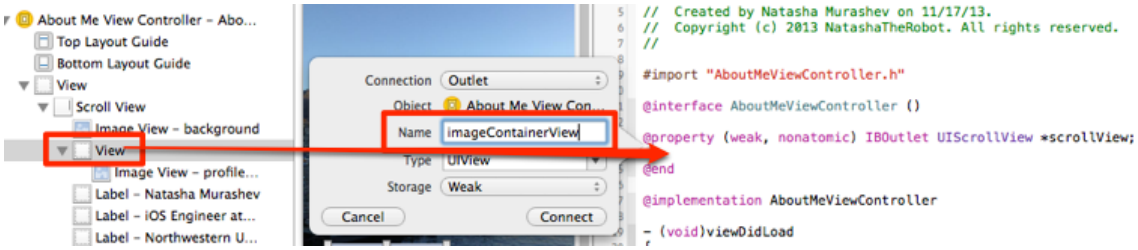
- Name your Scroll View outlet "**scrollView**" and click **Connect**



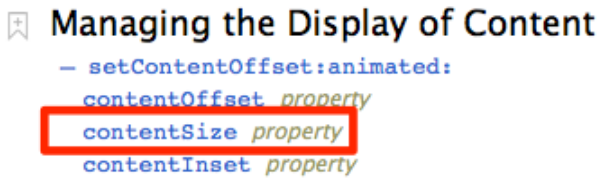
- Congratulations! You just connected a view to the storyboard to your ViewController, which means you can now programmatically modify it!

```
10
11  @interface AboutMeViewController ()
12
13  @property (weak, nonatomic) IBOutlet UIScrollView *scrollView;
14
15  @end
```

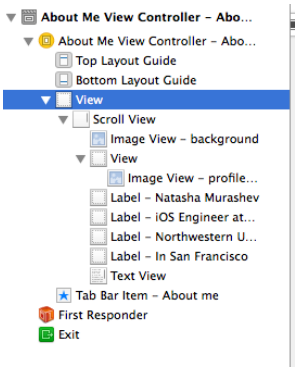
- Now, go through the same process (control drag from the Storyboard to the AboutMeViewController) to connect the View that holds your profile photo - we're going to programmatically add a border around it! Call the View **imageContainerView**.



- Open your AboutMeViewController.m file. Option **Double-Click on UIScrollView** to see the documentation for a UIScrollView. We're going to use the **UIScrollView's contentSize property** to increase the ScrollView's content size beyond what is visible, so scrolling is enabled.



- In your AboutMeViewController.m file, also **Option Double-Click on the UIView** to see the UIView's properties and methods. For our imageContainerView, we're going to use the **layer** property, which is responsible for drawing the border, to draw our border. We're also going to use the **main UIView's (the one that holds the ScrollView) frame property** to calculate the contentSize of the ScrollView.
- Back in the Storyboard, find the View that holds the ScrollView (you can close the Assistant Editor if you haven't already):



- The View that holds the Scroll View is a property of the UIViewController. Search for the UIViewController in the documentation again, and find the "Managing the View" section. That's where the view property is documented.
- The **Width of the Scroll View** should be the **same as the Width of the View** that holds it (it's **superview**). This is written in code as:

```
CGFloat width = self.view.frame.size.width;
```

- CGFloat is a fancy way of saying a decimal (e.g. 2.0 is a CGFloat).
- The Height of the Scroll View should be about the same as the Height of the SuperView plus about a fourth of that higher:

```
CGFloat height = self.view.frame.size.height + self.view.frame.size.height / 4;
```

- So the scrollView's contentSize, will be:

```
self.scrollView.contentSize = CGSizeMake(width, height);
```

- Add the above code in your ViewDidLoad method:

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    CGFloat width = self.view.frame.size.width;
    CGFloat height = self.view.frame.size.height + self.view.frame.size.height / 4;
    self.scrollView.contentSize = CGSizeMake(width, height);
}
```

- Run your Resume App, and try scrolling. It should work!
- To add a border to your imageContainerView, use the layer property to set the borderWidth to 1.0 pixels:

```
self.imageContainerView.layer.borderWidth = 1.0;
```

- In the documentation, look up UIColor to see the available properties and methods. We're going to use the groupTableViewBackgroundColor method for the color, and the CGColor property to convert the color to the Quartz color type:

```
self.imageContainerView.layer.borderColor = [UIColor groupTableViewBackgroundColor].CGColor;
```