# SMTE Practical assignment week 2

# Introduction

This week you are going to work with storyboards.

In this assignment you will create a tab bar controller, a modal dialog and a navigation controller.

You will also use IBAction and IBOutlet to link UI views to your controller classes.

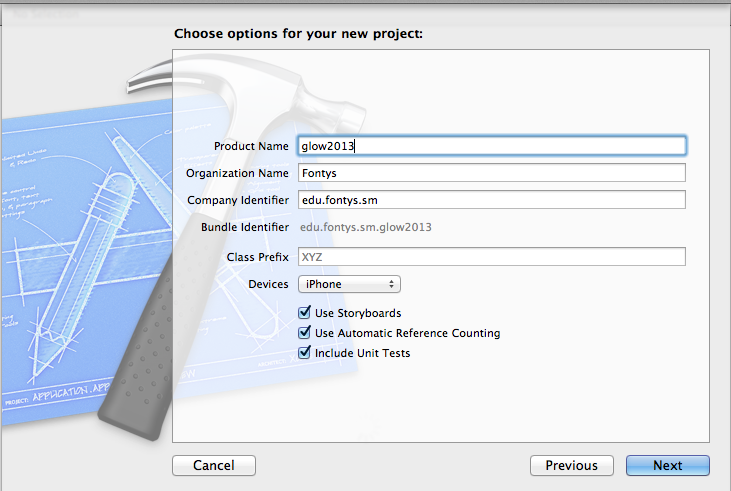
In this app you will work on the ‘Any App’™

Try to come up with a theme for this app, such as movies or TV series, or games. Anything will do.

## Step 1: New project.

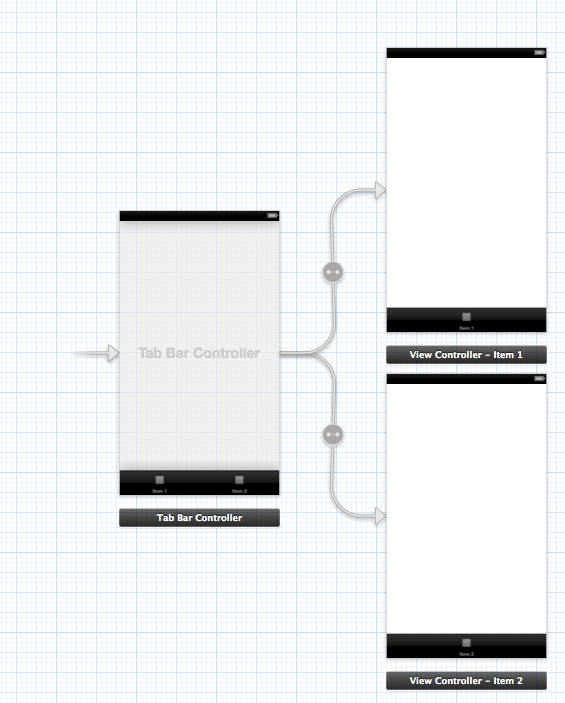
Create a new iOS application project and select ‘Single View application’ from the available presets. Select iPhone in the devices list and give the project a suitable name.

Use edu.fontys.sm as company identifier.



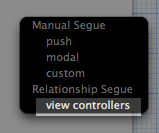
Select and open the main storyboard en remove the one view that is there. Add a tab bar controller to the storyboard.

You should now have something like this:

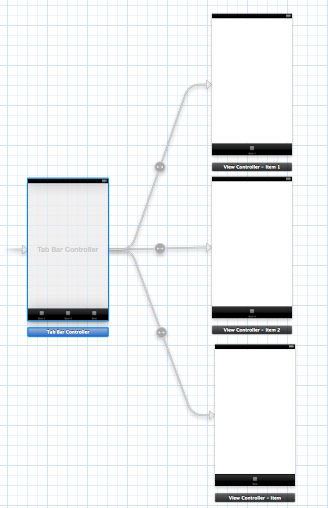


## Step 2: Add a tab

XCode was kind enough to create two tab views. However we want 3 tabs. In order to do this add a ViewController to the storyboard. To connect this view to the tabbar controller ctrl+ drag from the tabBar controller to the ViewController. Choose ‘View controllers’



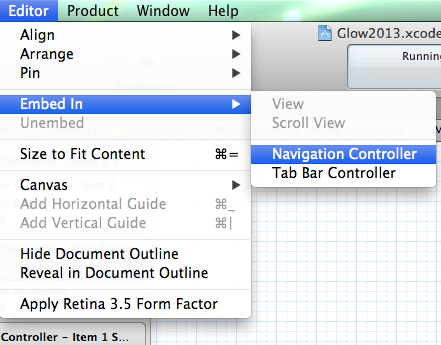
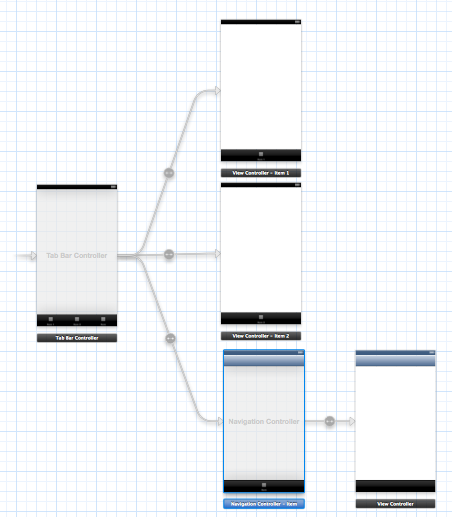
If done well you end up with this:



try running the application, you will see the above: 3 tab buttons.

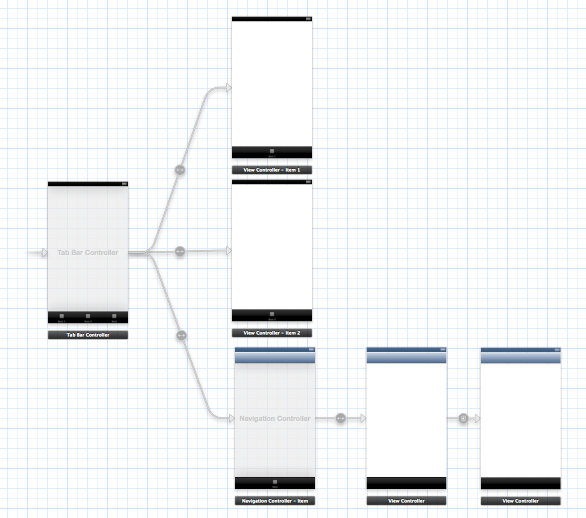
## Step 3: Special tab: navigation controller.

We are going to do something special with the third tab. Eventually it should display a table with information, but for now start simple. Select the view controller and select ‘Editor -> embed in -> Navigation controller’

 Which results in: 

Finally add another view controller and ctrl+ drag it to the third viewcontroller (Select manual segue, push).

You will end up with this:

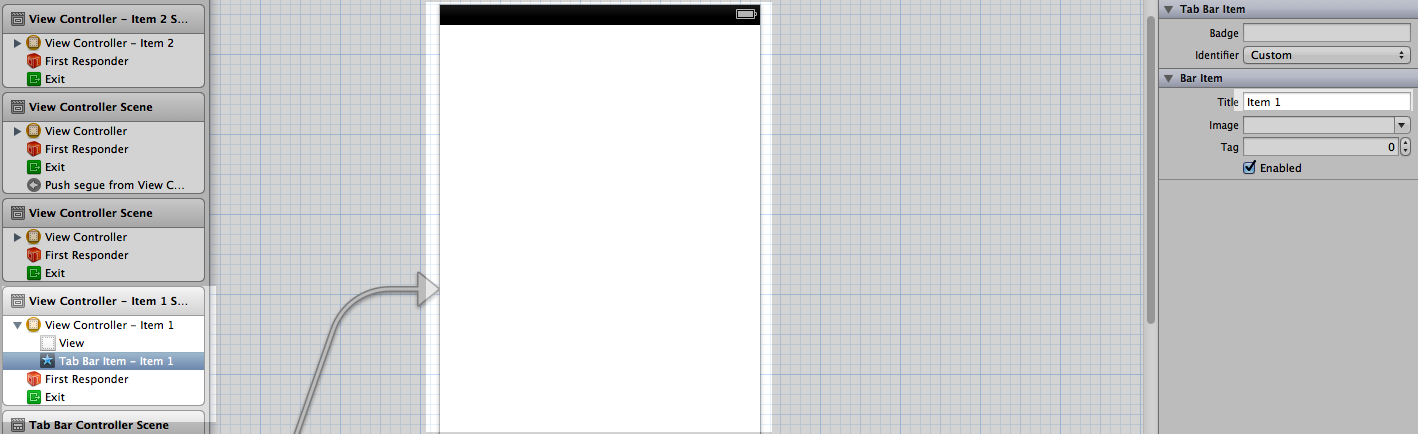


### Step 4: Fix the tab names

The tab bar uses some boring names like: Item 1, item2 etc. Let’s use some proper names:

* The first tab should give some general info, so ‘general’
* The second tab should say something about the creators, so ‘about’
* The third tab gives specific details, so ‘details.

You can change these names by selecting the viewcontroller, and select the TabBar item in the overview:



On the right side you can now give the proper name, and even select and icon if you want to.

### Step 5: Edit the first view (General)

The first view should display some general information. Feel free to add UI items as you see fit. Some text describing your app or even images. Be creative

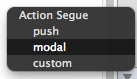
## Step 6 Edit the second view (About)

The second view should show some info about you, the creator of this app. Once again you are free to show whatever you want. There a few conditions: There has to be a button with text: ‘Easter egg’, a button with text ‘Show textfield’ and a text field. (Hint place the “show textfield” button above the textfield, you’ll find out why)

## Step 7 The easter egg view.

When the button is pressed another view should be shown to the user.

Add another viewcontroller to the storyboard. Ctrl+drag from the ‘Easter egg’ button to this view and choose ‘Modal

’ 

The easter egg should display something silly, like a riddle or a picture of an egg.

Add a button to it. Give it a text like ‘Close’. **Warning: Do not ctrl+drag a connection back to the previous view from the button.**

## Step 8 The details view

The details view should give some details about your app. A textview with text or some images for example. There should also be a button called ‘More details’

### Step 9 More details

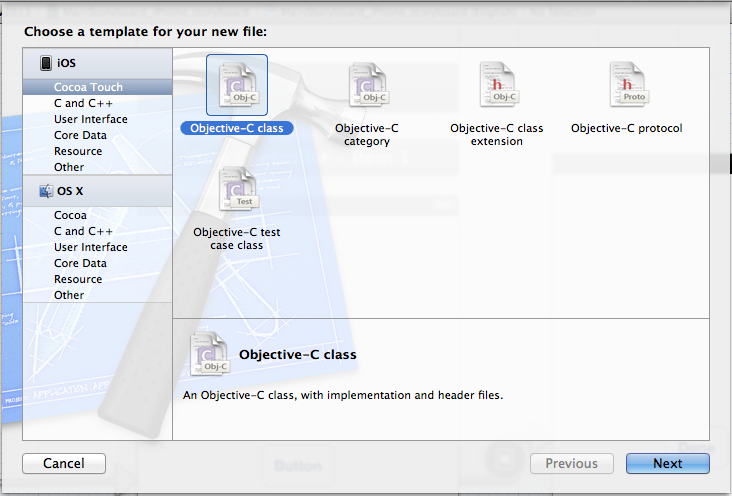
Create a new viewcontroller to the right of the details view. Connect it b ctrl+draggin from the ‘More details’ button to the new view.

As you can guess this view should display even more details. Do not add a button to go back, you won’t need it.

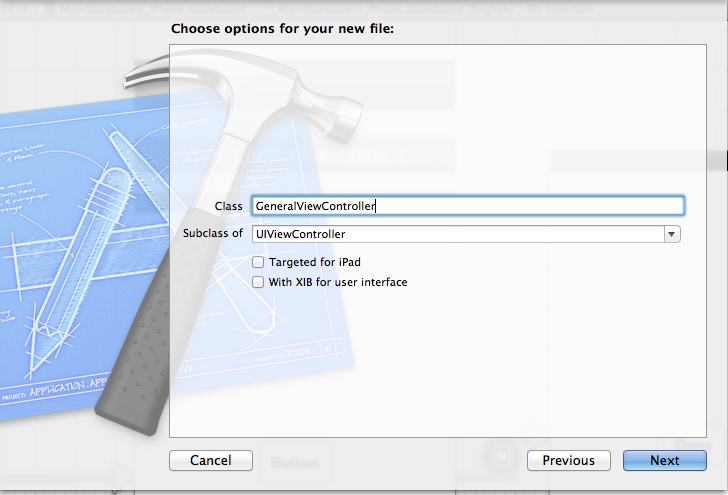
## Step 10: The view controllers

At this point the views are all created in the storyboard, but there is no code behind the views.

You need UIViewControllers. Use File -> new file -> Objective C Class.



Make sure that the class is a subclass of UIViewController



Create new UIViewControllers for each ViewController in the storyboard and give them a proper name:

- GeneralViewController

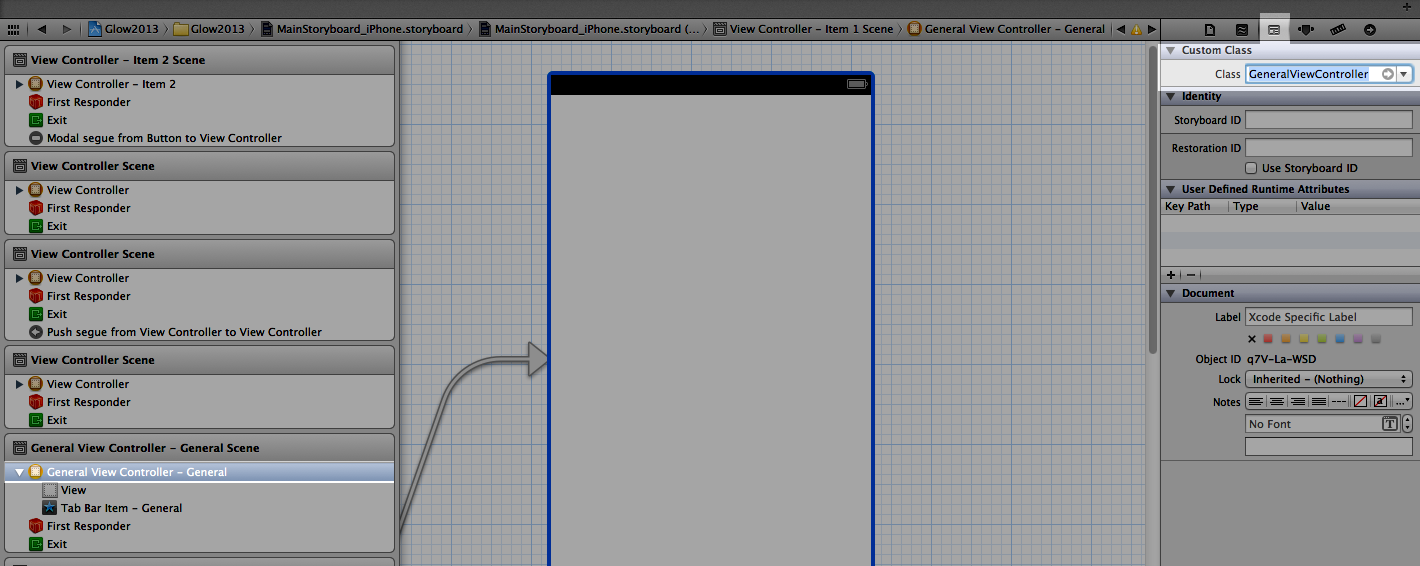
- AboutViewController

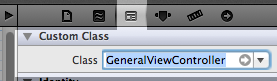
- EasterEggViewController

- DetailsViewController

- MoreDetailsViewController

Connect these classes to the views in the storyboard by setting the correct class in the identity inspector:





## Step 11: Connect the Actions and Outlets

In the about view and the easter egg view you’ve added a few buttons and a text field. Just adding them to the view won’t make them work.

Activate the assistant editor and ctrl+drag IBActions from the buttons to the controller.



Do the same for the IBOutlet of the textfield. Be sure to give these actions and outlets a proper name. For example easterEggButton etc.

### Step 12: Close the easter egg

In the generated IBAction for the close button on the Easter egg view add the following code.

[self dismissViewControllerAnimated:YES completion:nil];

this closes the modal dialog.

## Step 13 Show the text

In the about dialog add the following code to the IBAction that you created for the ‘Show textfield’ button.

//read the text from the IBOutlet. make sure you use the name you gave to the IBoutlet of the textfield

NSString\* selectedText = self.textField.text;

//Create an AlertView. This view can show a dialog.

UIAlertView \*alertView =

[[UIAlertView alloc] initWithTitle:@"Your text is:" message:selectedText delegate:nil cancelButtonTitle:@"Done" otherButtonTitles: nil];

//Actualy show the dialog by sending the 'show' message

[alertView show];

Warning: Make sure that you use you IBOutlet name. In the example above it’s ‘selectedText’

## Step 14: Done

Congratulations you finshed the ‘Any App’™ It doesn’t do much but you’ve learned a lot about storyboards. Now go right ahead and create the UI flow for your own app!