Developing Applications for iOS

Lab 5:
Nearby Deals (1 of 6)

Radu Ionescu
raducu.ionescu@gmail.com
Faculty of Mathematics and Computer Science
University of Bucharest

Nearby Deals

Description:

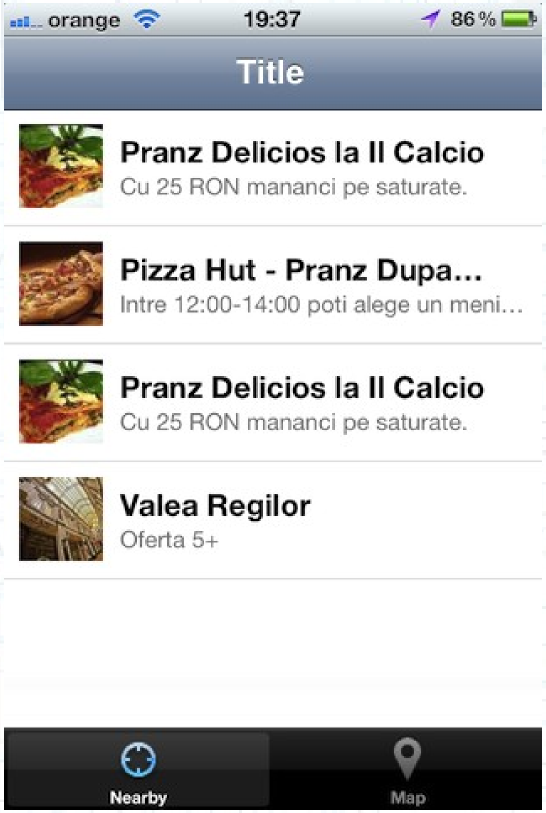
We are going to build a new application that will show deals from nearby restaurants and bars. The application will display the deals in two modes: a list view (using a `UITableViewController`) and a map view (using a `MKMapView`). We will request the deals from a server (www.geoadsplus.com to be more precise). We will use XML to communicate with this server. Note that XML and JSON are standard ways of communicating with a server.

We have to pass the device location (latitude, longitude) to the server so that it gives us nearby deals. Thus, we will need to use Location Services to determine the device location.

We will offer details about our deals. We are going to use a navigation controller to navigate between the list View and the details View.

The following screenshots are just a mock-up of the application that we are going to start building today. We will continue this app during the next 5 labs.

Nearby Deals



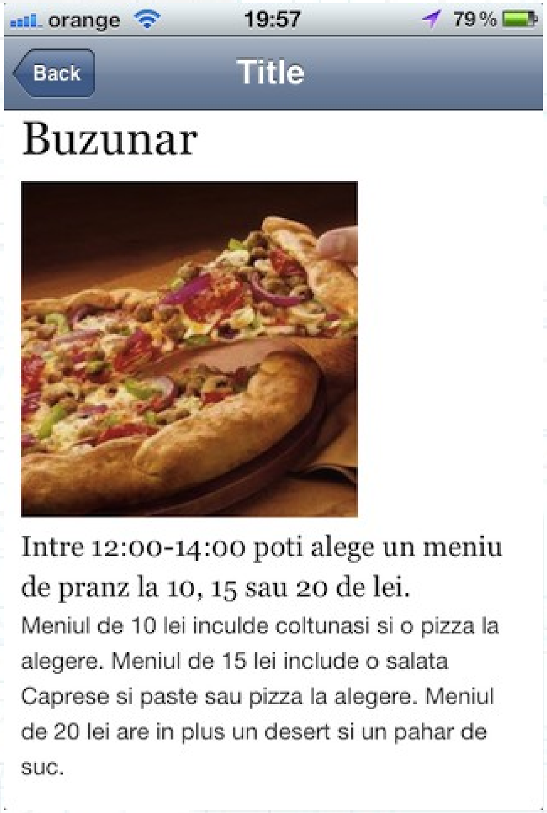
Title

Pranz Delicios la Il Calcio
Cu 25 RON mananci pe saturate.

Pizza Hut - Pranz Dupa...
Intre 12:00-14:00 poti alege un meni...

Pranz Delicios la Il Calcio
Cu 25 RON mananci pe saturate.

Valea Regilor
Oferta 5+

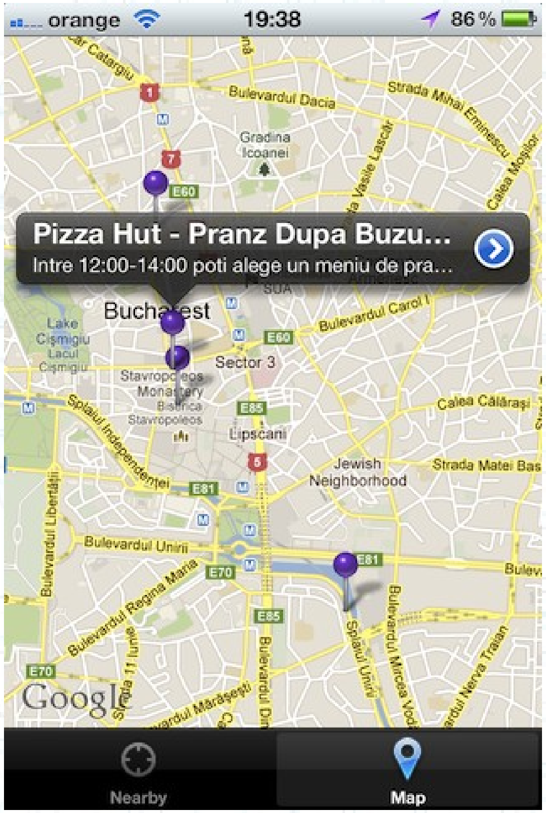


Title

Buzunar

Intre 12:00-14:00 poti alege un meniu de pranz la 10, 15 sau 20 de lei.

Meniul de 10 lei inculde coltunasi si o pizza la alegere. Meniul de 15 lei include o salata Caprese si paste sau pizza la alegere. Meniul de 20 lei are in plus un desert si un pahar de suc.



Pizza Hut - Pranz Dupa Buzu...
Intre 12:00-14:00 poti alege un meniu de pra...

Task 1

Task: Create a new application in Xcode called “NearbyDeals”.

1. Launch Xcode and select the “Create a new Xcode project” option. If you don't see the splash window, you should go to “File > New > New Project...” in Xcode menu.

2. Select the Single View Application template and click Next.

We are actually going to build a Tabbed Application (using a `UITabBarController`) as you can see on the previous slide, but we are going to do it from scratch so that you can learn how to create complex Storyboards yourself.

3. Type in “NearbyDeals” for the Product Name.

4. Enter “com.FMI.FirstName.LastName” for the Company Identifier. Notice how Bundle Identifier changes as you type. You should obtain something like “com.FMI.Radu.Ionescu.Calculator” as your bundle identifier.

5. Enter “NearbyDeals” as the Class Prefix for the classes this template is going to generate for us.

Task 1

Task: Create a new application in Xcode called “NearbyDeals”.

6. Select “iPhone” for Device Family.

7. Check “Use Storyboard”. We definitely want to use Storyboards for this app that will contain more Views. We will segue from one View to another using Storyboards.

8. Check “Use Automatic Reference Counting”.

9. We won't be creating Unit Tests for this application so we are going to leave the “Include Unit Tests” option unchecked.

10. Click Next.

11. Navigate to “~/Developer/Apps/” folder inside the home directory. If you want to keep your project for later use, please save it in a directory with your name like this: “~/Developer/Apps/<YourName>”.

12. Click Create to create your project directory inside the “~/Developer/Apps” folder.

Task 2

Task: Start building the app by creating its Storyboard.

1. Open up our MVC's View by clicking on MainStoryboard.storyboard in Project Navigator.

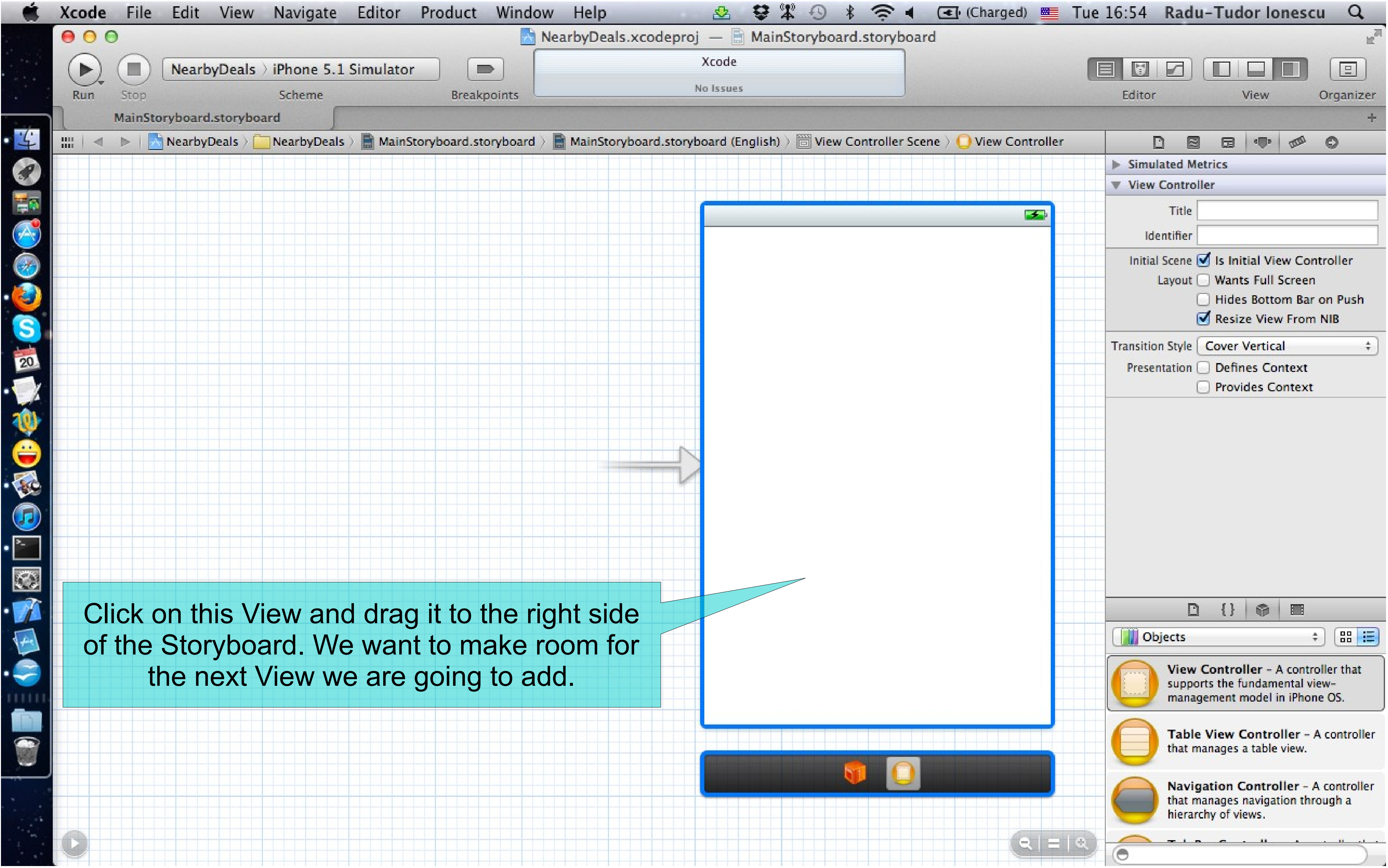
2. Hide the Document Outline if it's not already hidden.

3. We don’t need the Project Navigator at the far left either, so let’s hide it by using the “Hide or show the Navigator” button available on the Toolbar.

4. Bring up the Utilities area by clicking on the “Hide or show the Utilities” button that is also available on the Toolbar.

5. In Utilities area, click on the Object Library (it might already be selected). Some objects (those appropriate to dragging into your View) should appear in the Object Library.

6. Your Xcode project should be set up as in the next screenshot. We are now ready to create the Storyboard.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 16:54 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › View Controller Scene › View Controller
Simulated Metrics
View Controller
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style: Cover Vertical

Presentation □ Defines Context

□ Provides Context

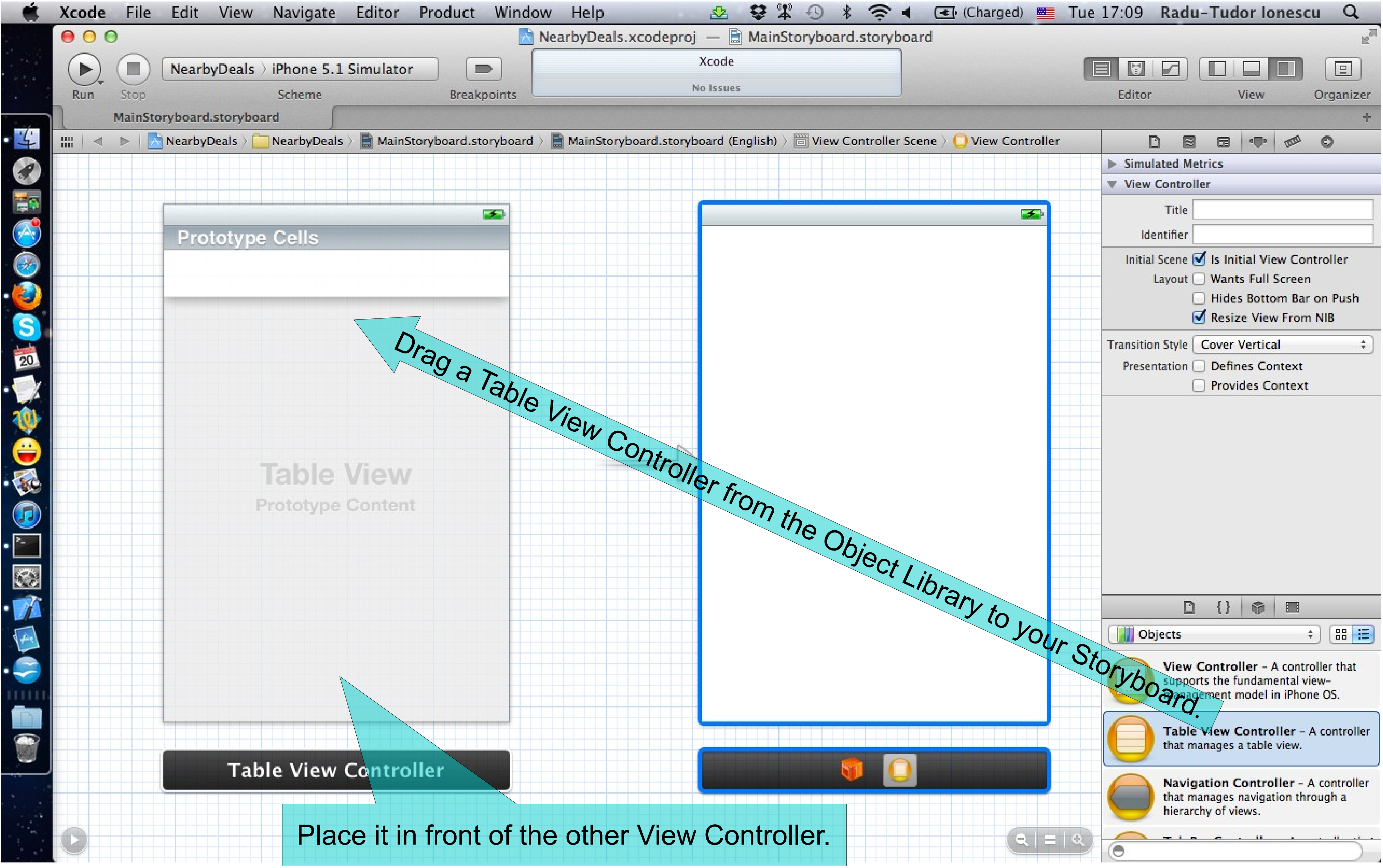
Click on this View and drag it to the right side of the Storyboard. We want to make room for the next View we are going to add.

Task 2

Task: Start building the app by creating its Storyboard.

7. We are going to add a Table View Controller to our Storyboard that will contain the list of nearby deals. The current View Controller will be used to present deal details. We are going to create a segue from the Table View Controller to this View Controller.

Follow the instructions from the next slides to learn how to do this.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 17:09 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › View Controller Scene › View Controller
Prototype Cells
Table View
Prototype Content
Table View Controller
Simulated Metrics
View Controller
Objects

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style: Cover Vertical

Presentation □ Defines Context

□ Provides Context

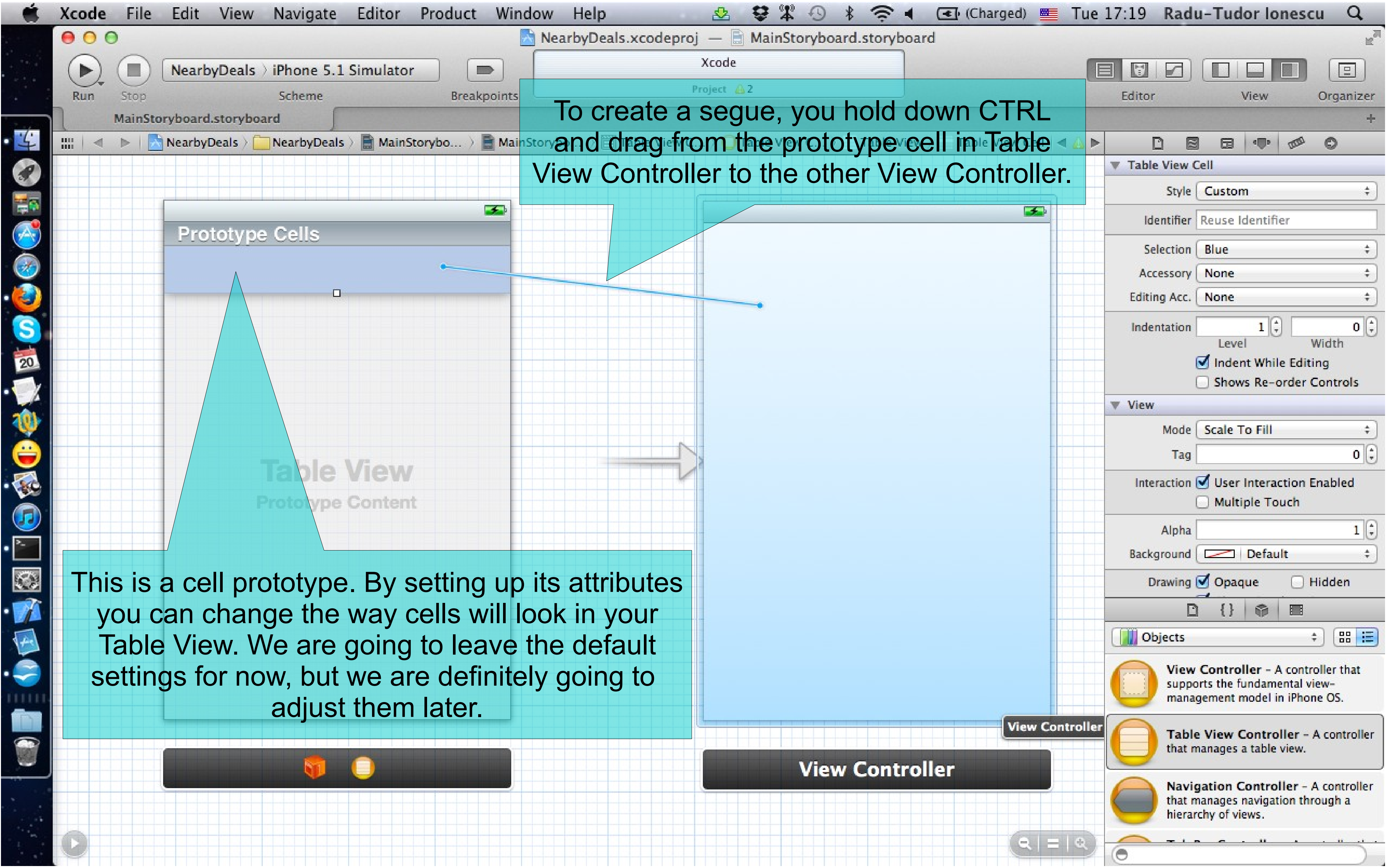
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Drag a Table View Controller from the Object Library to your Storyboard.

Place it in front of the other View Controller.



To create a segue, you hold down CTRL and drag from the prototype cell in Table View Controller to the other View Controller.

This is a cell prototype. By setting up its attributes you can change the way cells will look in your Table View. We are going to leave the default settings for now, but we are definitely going to adjust them later.

Table View Cell

Style: Custom

Identifier: Reuse Identifier

Selection: Blue

Accessory: None

Editing Acc.: None

Indentation: 1 Level, 0 Width

☑ Indent While Editing

□ Shows Re-order Controls

View

Mode: Scale To Fill

Tag: 0

Interaction: ☑ User Interaction Enabled

□ Multiple Touch

Alpha: 1

Background: Default

Drawing: ☑ Opaque □ Hidden

Objects

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

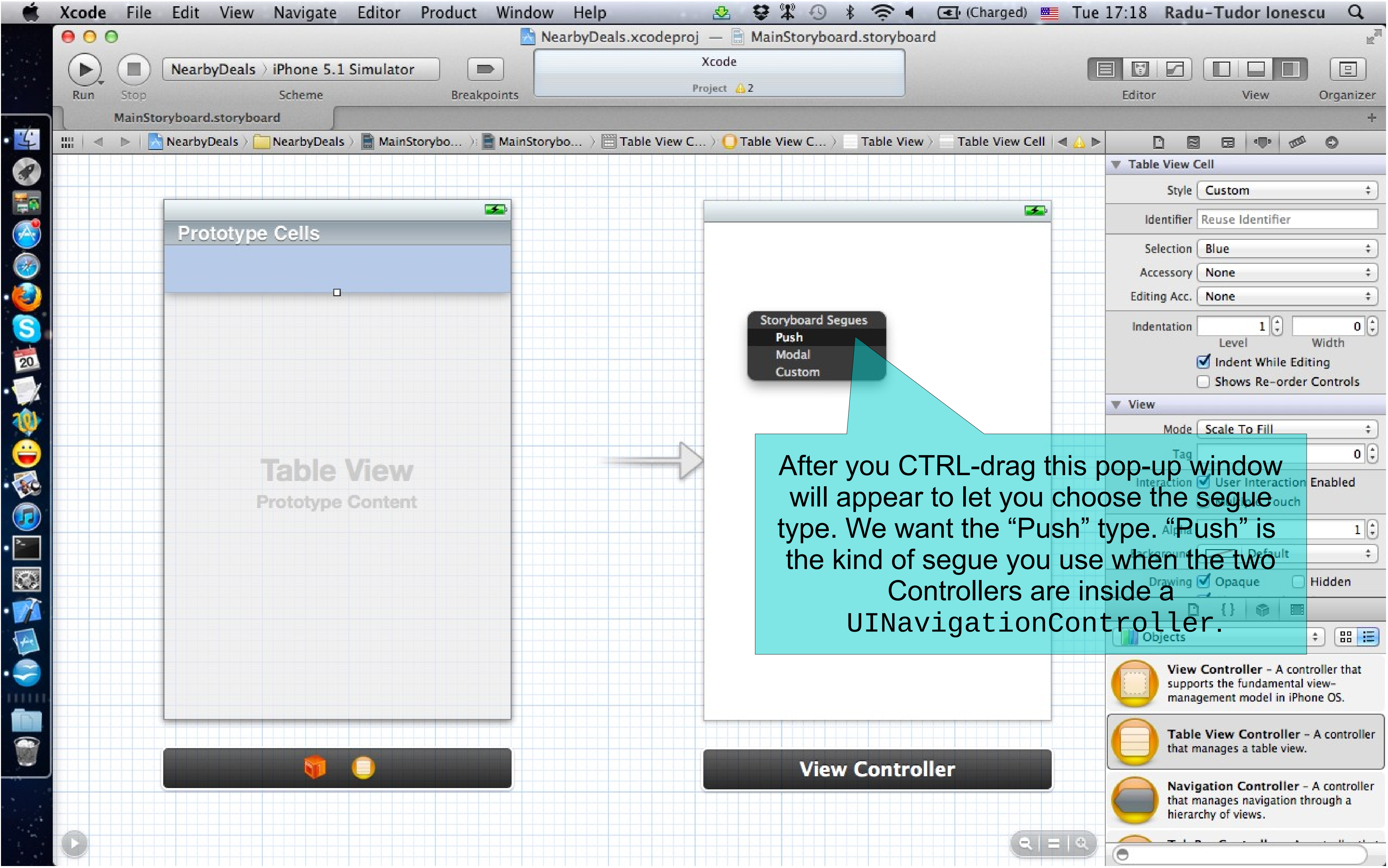

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 17:18 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStorybo... › MainStorybo... › Table View C... › Table View C... › Table View › Table View Cell
Prototype Cells
Table View
Prototype Content
Storyboard Segues
Push
Modal
Custom
View Controller
Objects

Table View Cell

Style: Custom

Identifier: Reuse Identifier

Selection: Blue

Accessory: None

Editing Acc.: None

Indentation: 1 Level, 0 Width

☑ Indent While Editing

□ Shows Re-order Controls

View

Mode: Scale To Fill

Tag: 0

Interaction: ☑ User Interaction Enabled

Alpha: 1

Background: Default

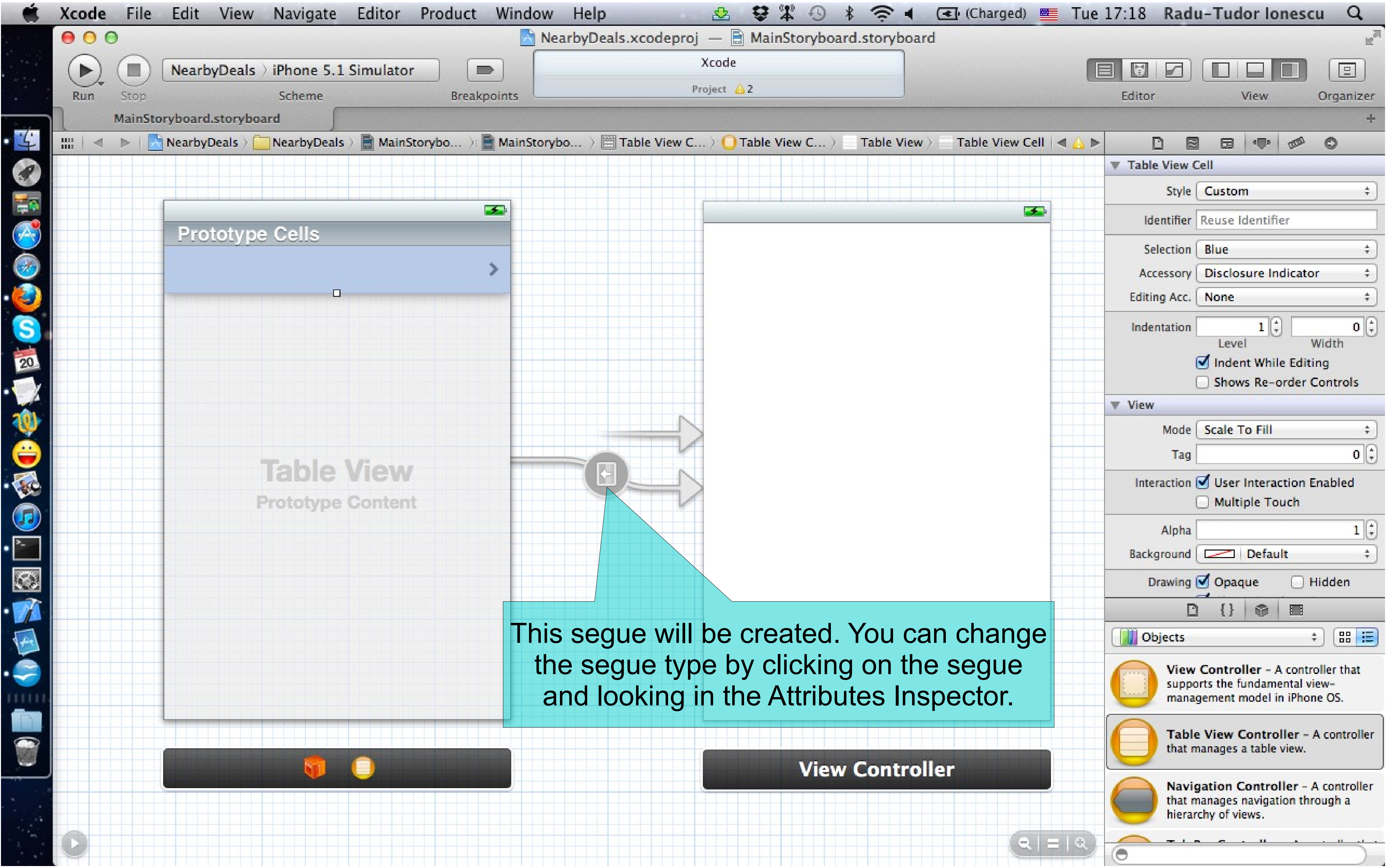
Drawing: ☑ Opaque □ Hidden

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

After you CTRL-drag this pop-up window will appear to let you choose the segue type. We want the “Push” type. “Push” is the kind of segue you use when the two Controllers are inside a `UINavigationController`.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 17:18 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStorybo... › MainStorybo... › Table View C... › Table View C... › Table View › Table View Cell
Prototype Cells
Table View
Prototype Content
View Controller
Objects

This segue will be created. You can change the segue type by clicking on the segue and looking in the Attributes Inspector.

Table View Cell

Style: Custom

Identifier: Reuse Identifier

Selection: Blue

Accessory: Disclosure Indicator

Editing Acc.: None

Indentation: 1 Level, 0 Width

☑ Indent While Editing

□ Shows Re-order Controls

View

Mode: Scale To Fill

Tag: 0

Interaction: ☑ User Interaction Enabled

□ Multiple Touch

Alpha: 1

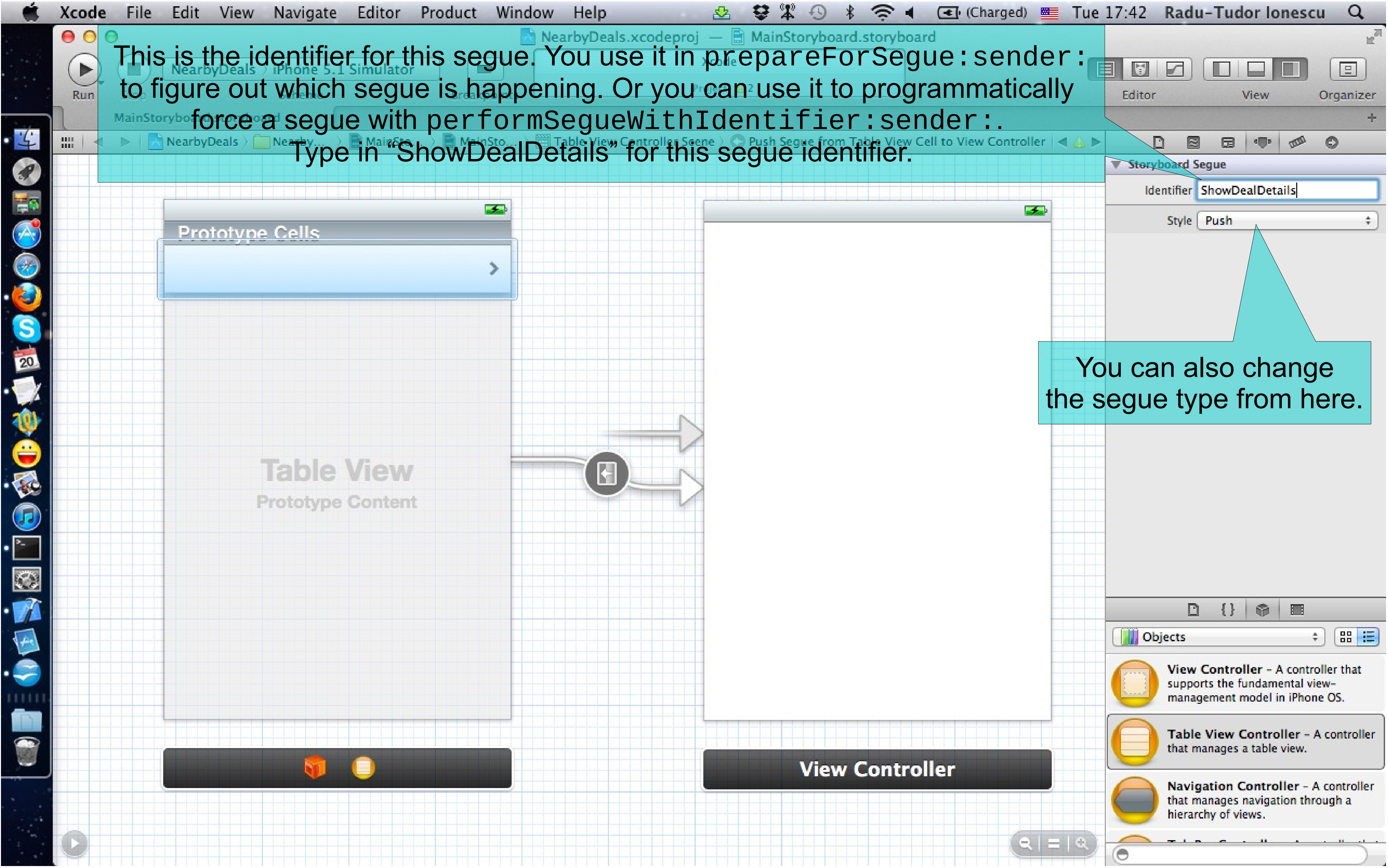
Background: Default

Drawing: ☑ Opaque □ Hidden

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.



This is the identifier for this segue. You use it in `prepareForSegue:sender:` to figure out which segue is happening. Or you can use it to programmatically force a segue with `performSegueWithIdentifier:sender:`.

Type in "ShowDealDetails" for this segue identifier.

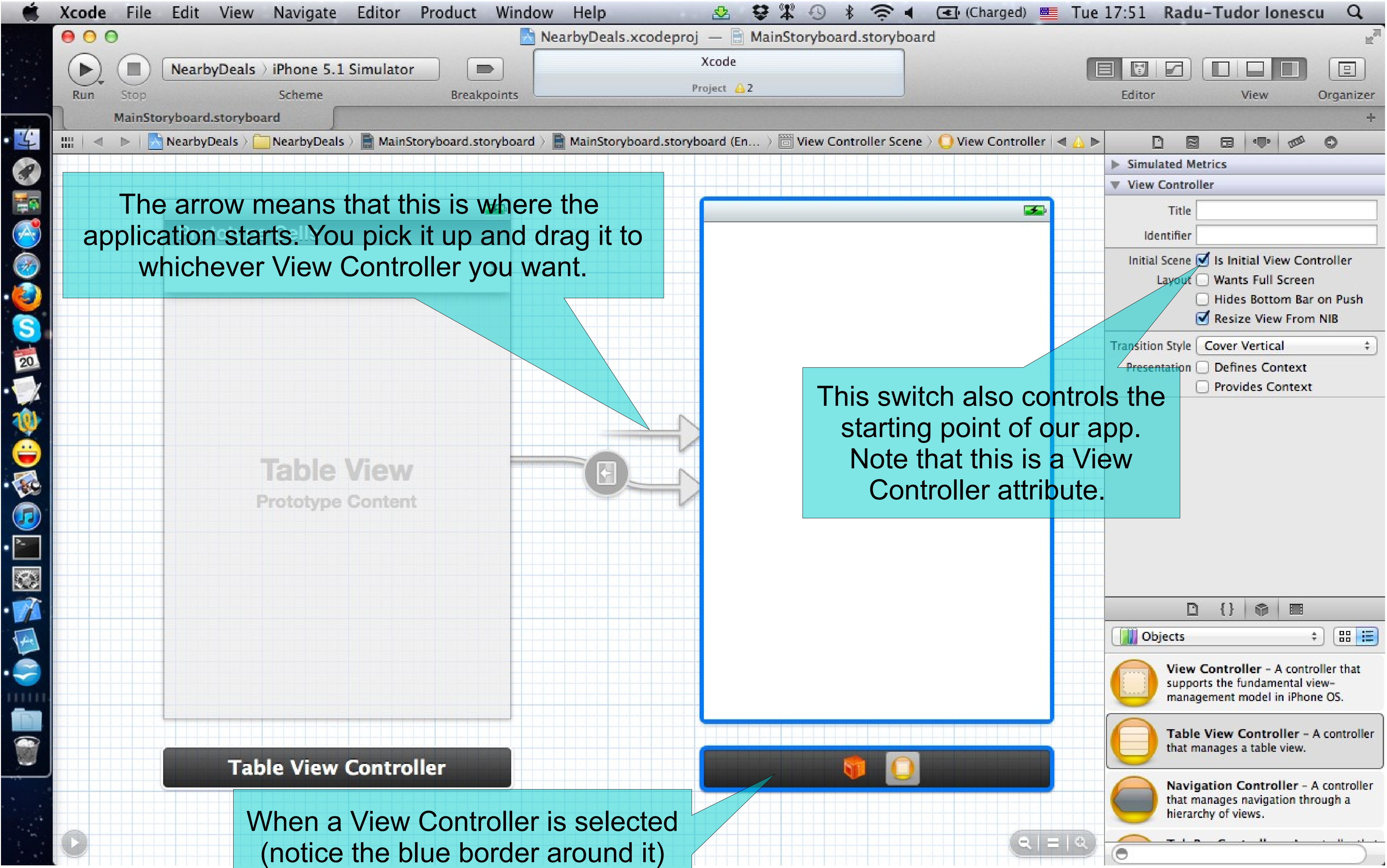
You can also change the segue type from here.

Storyboard Segue

Identifier: ShowDealDetails

Style: Push

Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
(Charged)
Tue 17:42
Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Editor
View
Organizer
NearbyDeals
Push Segue from Table View Cell to View Controller
Prototype Cells
Table View
Prototype Content
View Controller
Objects
View Controller - A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller - A controller that manages a table view.
Navigation Controller - A controller that manages navigation through a hierarchy of views.



The arrow means that this is where the application starts. You pick it up and drag it to whichever View Controller you want.

This switch also controls the starting point of our app. Note that this is a View Controller attribute.

When a View Controller is selected (notice the blue border around it) its attributes show up in Attributes Inspector.

NearbyDeals.xcodeproj — MainStoryboard.storyboard

View Controller

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style: Cover Vertical

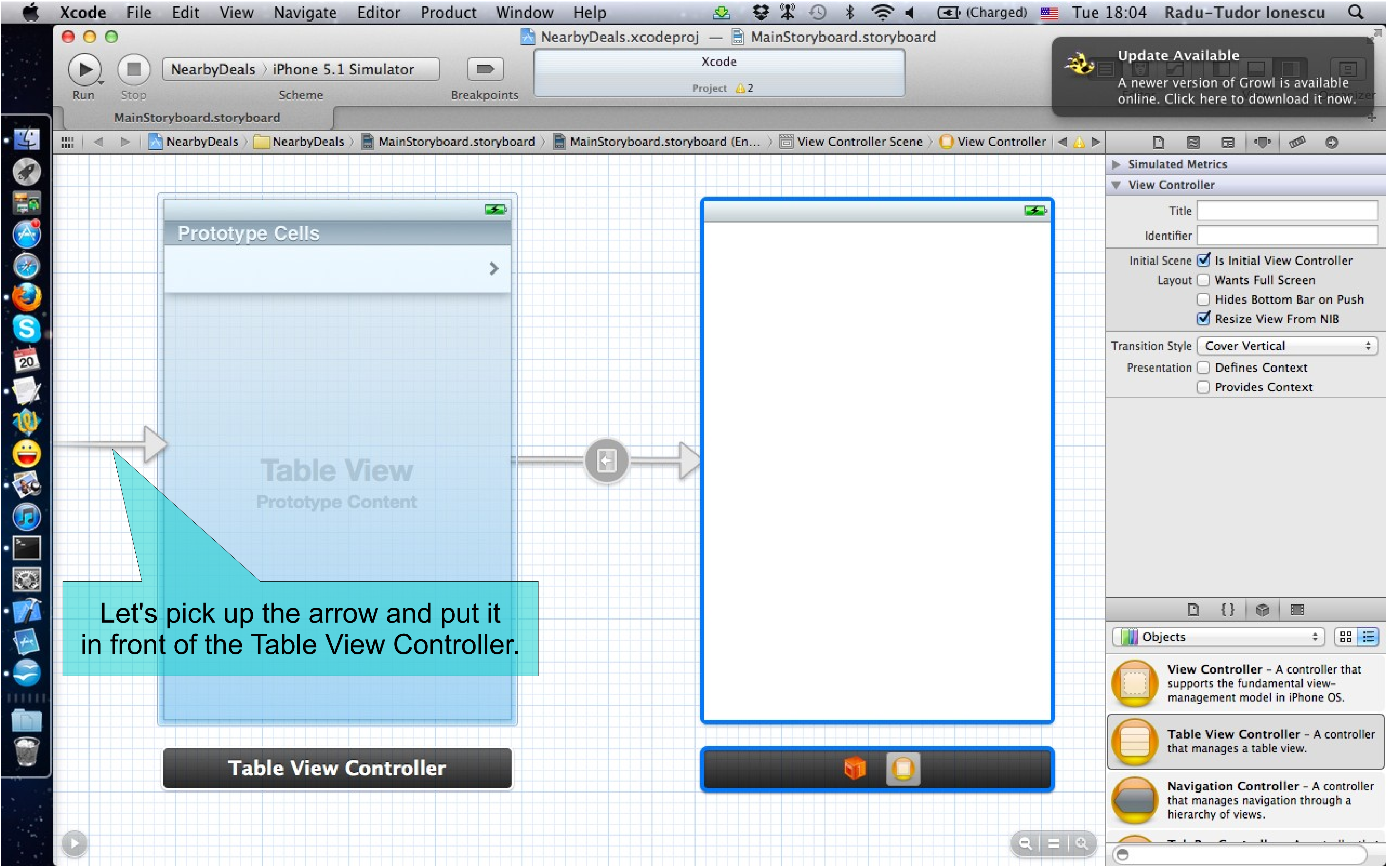
Presentation □ Defines Context

□ Provides Context

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 18:04 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
Project ⚠ 2
Update Available
A newer version of Growl is available online. Click here to download it now.
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (En… › View Controller Scene › View Controller
Prototype Cells
Table View
Prototype Content
Table View Controller
Objects

Simulated Metrics

View Controller

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style Cover Vertical

Presentation □ Defines Context

□ Provides Context

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

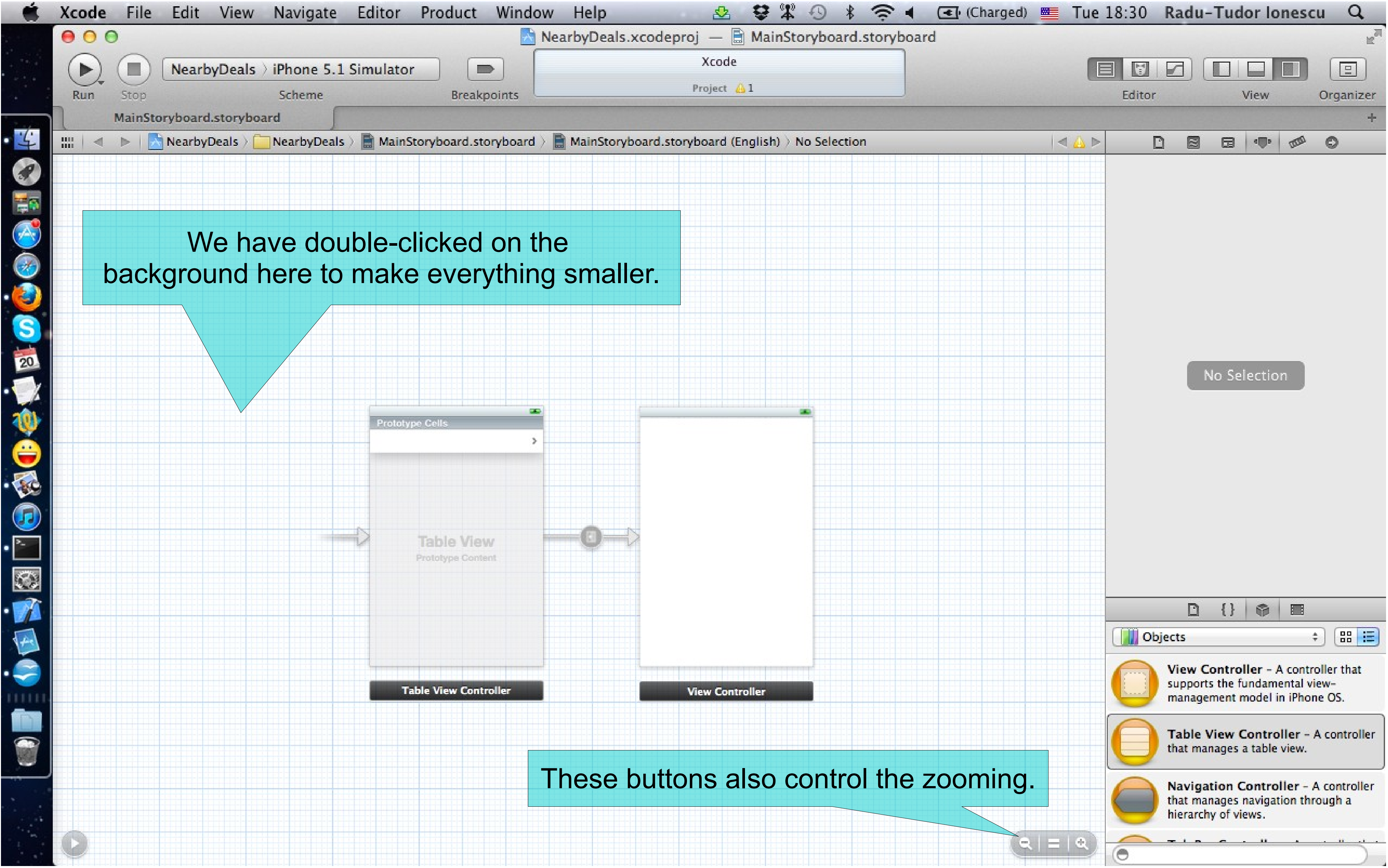
Let's pick up the arrow and put it in front of the Table View Controller.

Task 2

Task: Start building the app by creating its Storyboard.

8. We have created our first segue but there is a problem here. These View Controllers are not inside a `UINavigationController`. Push will do nothing in this case.

We have to embed our View Controllers inside a Navigation Controller. Follow the instructions from the next slides to learn how to do this step.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 18:30 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
Project ⚠ 1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals 〉 NearbyDeals 〉 MainStoryboard.storyboard 〉 MainStoryboard.storyboard (English) 〉 No Selection
Prototype Cells
Table View
Prototype Content
Table View Controller
View Controller
No Selection
Objects

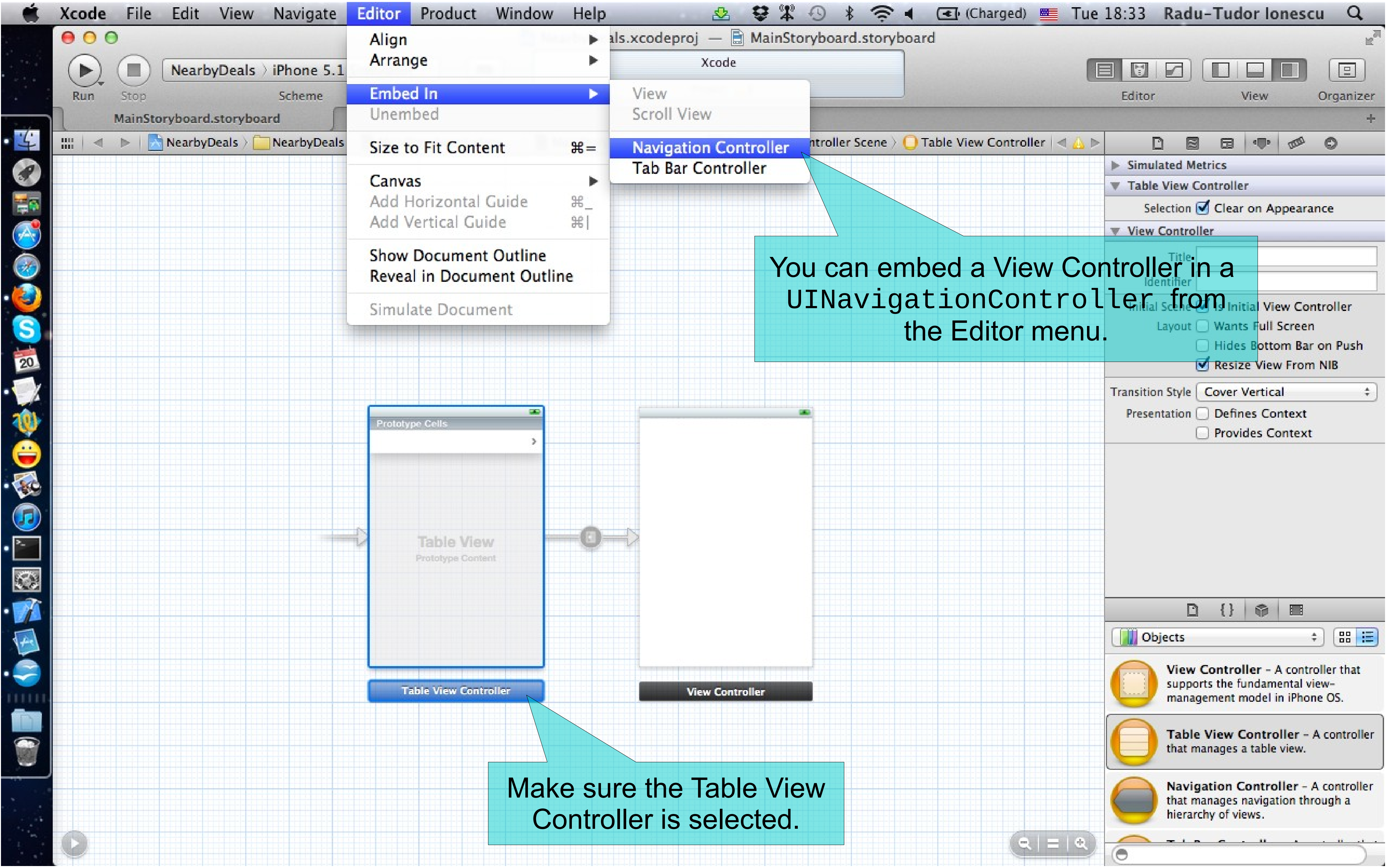
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

We have double-clicked on the background here to make everything smaller.

These buttons also control the zooming.



Selection ☑ Clear on Appearance

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style: Cover Vertical

Presentation □ Defines Context

□ Provides Context

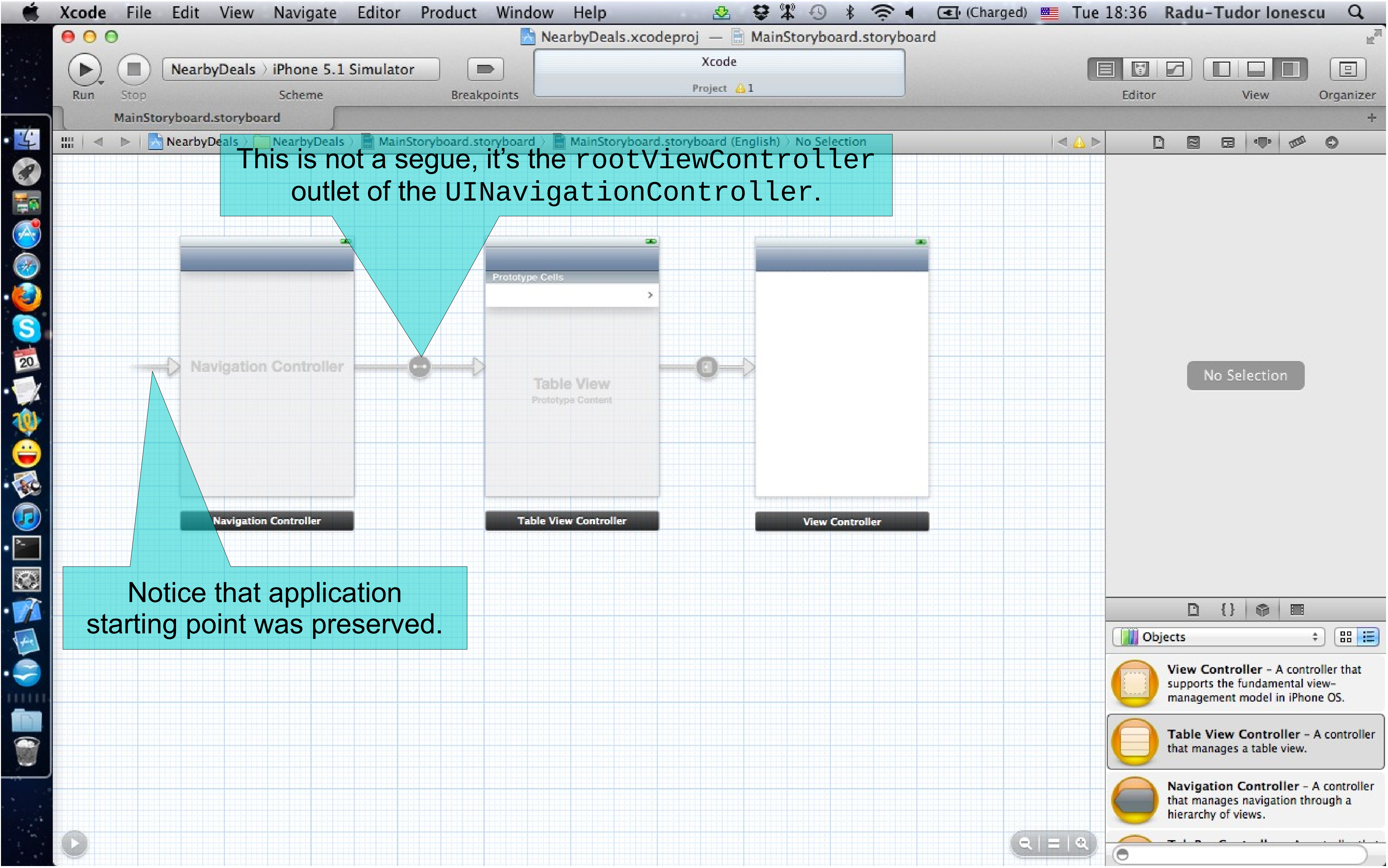
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

You can embed a View Controller in a `UINavigationController` from the Editor menu.

Make sure the Table View Controller is selected.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 18:36 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Xcode
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › No Selection
Navigation Controller
Prototype Cells
Table View
Prototype Content
Navigation Controller
Table View Controller
View Controller
No Selection
Objects

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

This is not a segue, it's the `rootViewController` outlet of the `UINavigationController`.

Notice that application starting point was preserved.

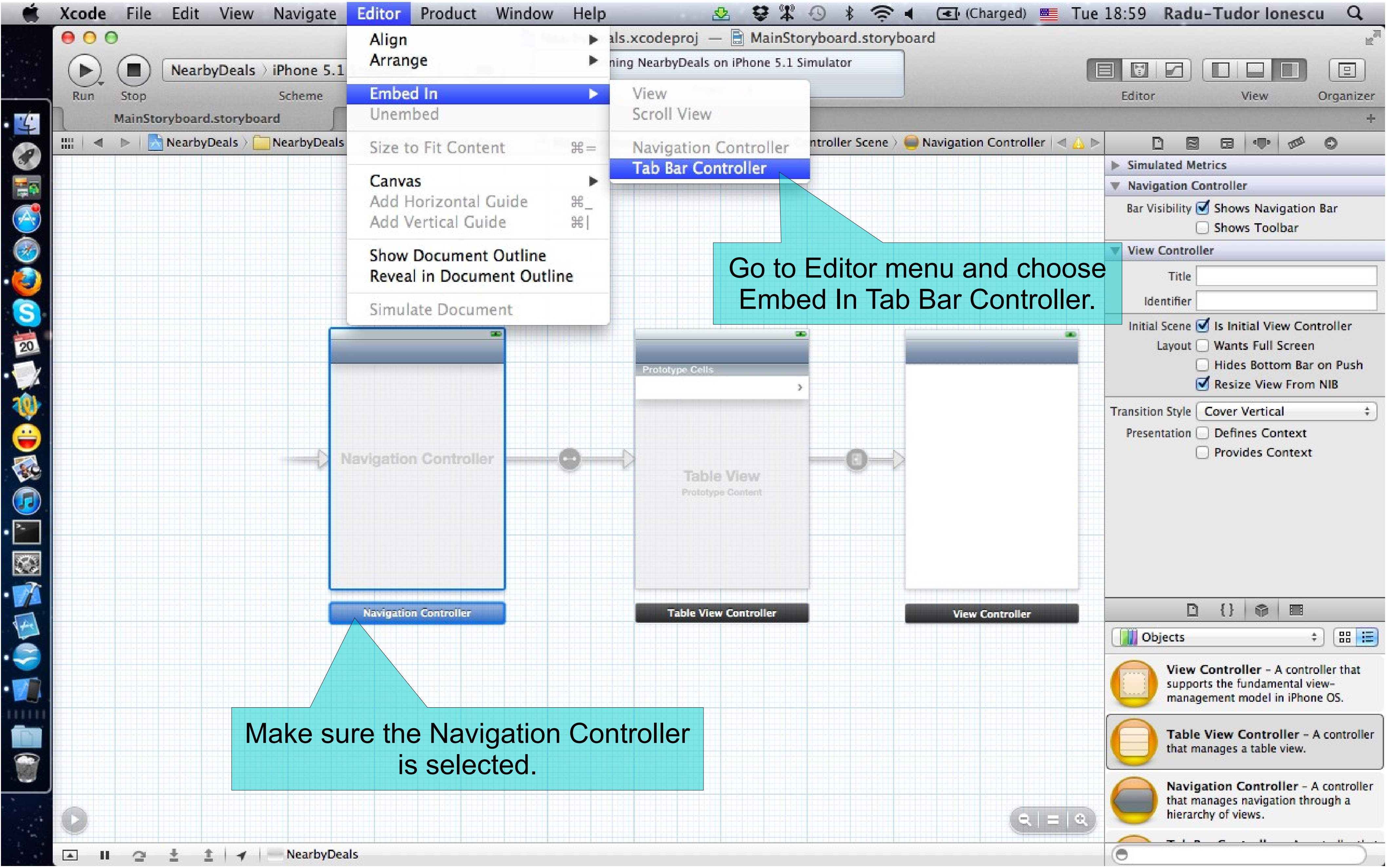
Task 2

Task: Start building the app by creating its Storyboard.

9. In a similar way, we will embed the Navigation Controller inside a Tab Bar Controller. Note that this is always the way to go (we never embed a Tab Bar Controller inside a Navigation Controller).

10. To complete the application Storyboard we will add another View Controller to the Tab Bar for the map view.

Follow the instructions from the next slides to learn how to do these steps.



Simulated Metrics

Navigation Controller

Bar Visibility ☑ Shows Navigation Bar

□ Shows Toolbar

View Controller

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style Cover Vertical

Presentation □ Defines Context

□ Provides Context

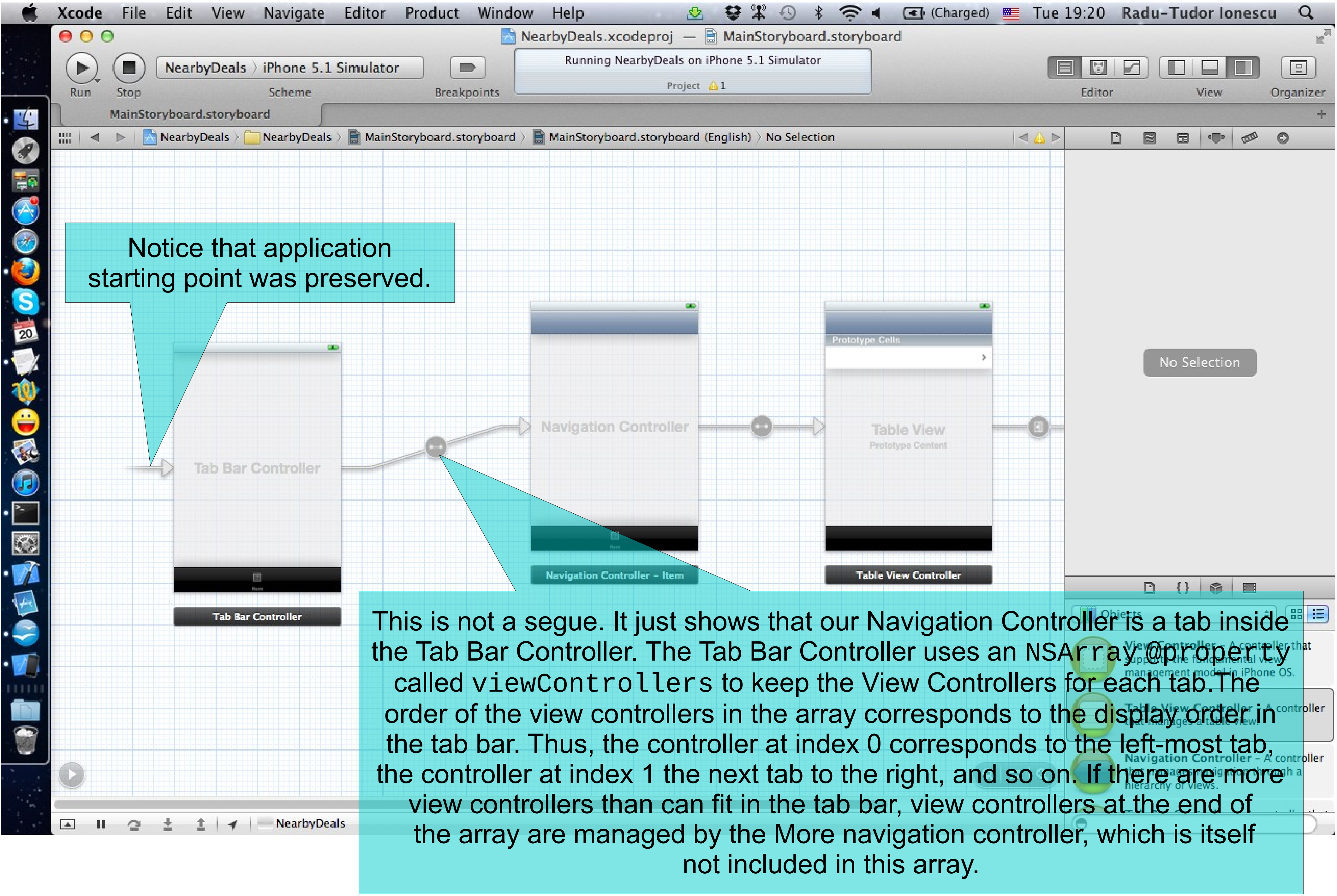
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

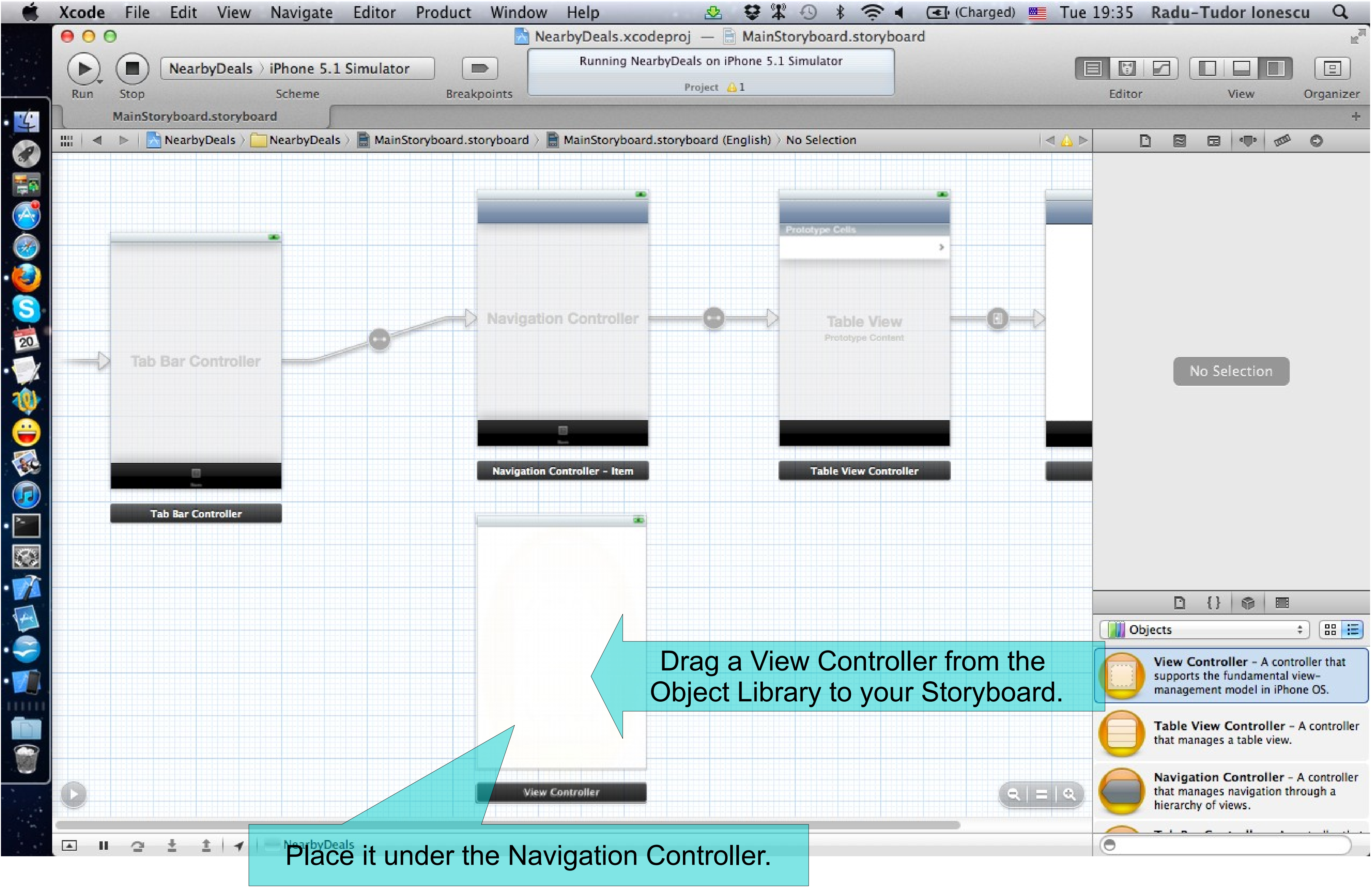
Go to Editor menu and choose Embed In Tab Bar Controller.

Make sure the Navigation Controller is selected.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 19:20 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › No Selection
Tab Bar Controller
Tab Bar Controller
Navigation Controller
Navigation Controller - Item
Prototype Cells
Table View
Prototype Content
Table View Controller
No Selection
Objects
View Controller - A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller - A controller that manages a table view.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
NearbyDeals

Notice that application starting point was preserved.

This is not a segue. It just shows that our Navigation Controller is a tab inside the Tab Bar Controller. The Tab Bar Controller uses an `NSArray` `@property` called `viewControllers` to keep the View Controllers for each tab.The order of the view controllers in the array corresponds to the display order in the tab bar. Thus, the controller at index 0 corresponds to the left-most tab, the controller at index 1 the next tab to the right, and so on. If there are more view controllers than can fit in the tab bar, view controllers at the end of the array are managed by the More navigation controller, which is itself not included in this array.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 19:35 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
Project ⚠ 1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › No Selection
Tab Bar Controller
Tab Bar Controller
Navigation Controller
Navigation Controller - Item
Table View
Prototype Content
Prototype Cells
Table View Controller
View Controller
No Selection
Objects
NearbyDeals

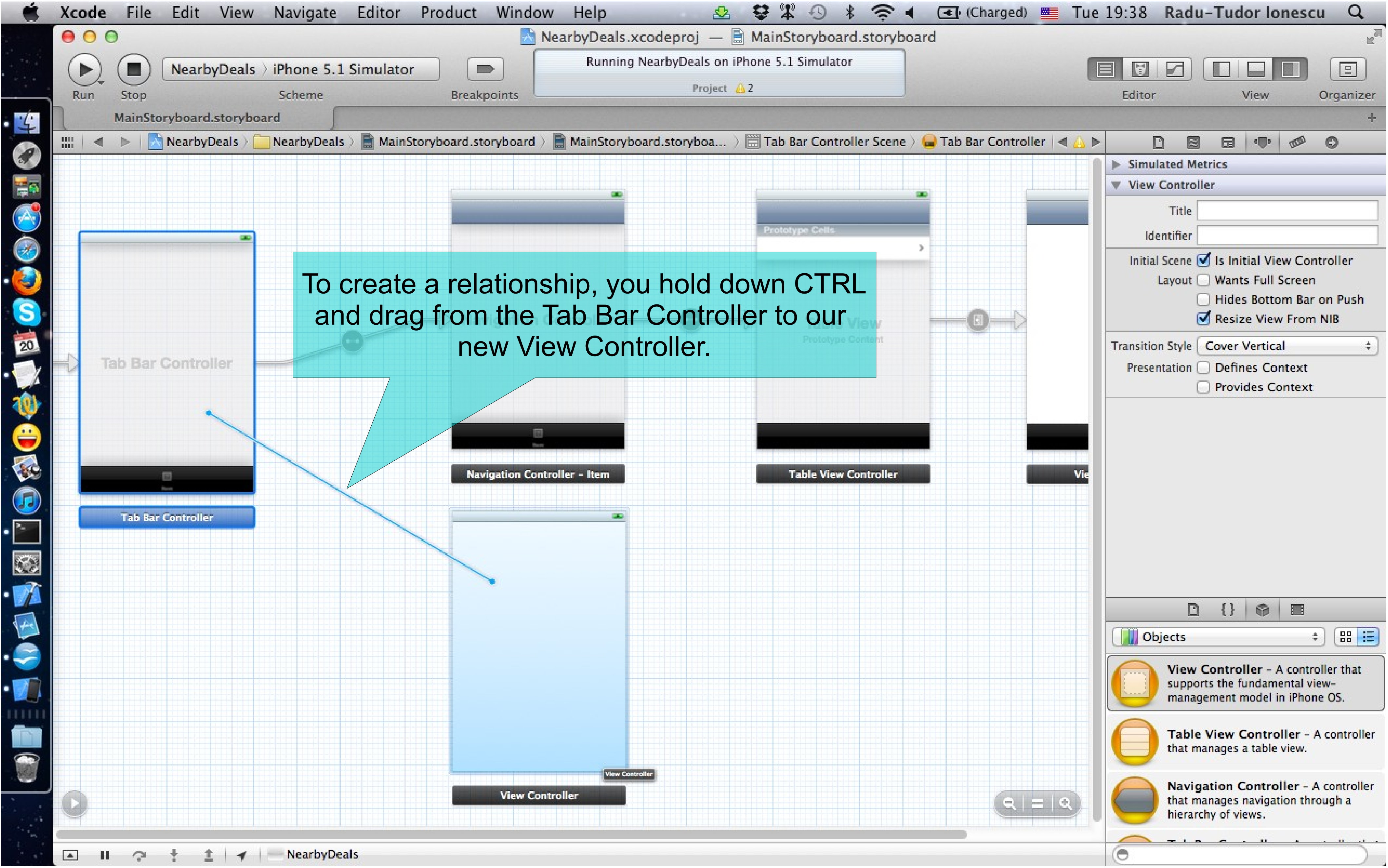
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Drag a View Controller from the Object Library to your Storyboard.

Place it under the Navigation Controller.



To create a relationship, you hold down CTRL and drag from the Tab Bar Controller to our new View Controller.

NearbyDeals.xcodeproj — MainStoryboard.storyboard

Running NearbyDeals on iPhone 5.1 Simulator

NearbyDeals 〉 NearbyDeals 〉 MainStoryboard.storyboard 〉 MainStoryboard.storyboa... 〉 Tab Bar Controller Scene 〉 Tab Bar Controller

▶ Simulated Metrics

▼ View Controller

Title

Identifier

Initial Scene ☑ Is Initial View Controller

Layout □ Wants Full Screen

□ Hides Bottom Bar on Push

☑ Resize View From NIB

Transition Style: Cover Vertical

Presentation □ Defines Context

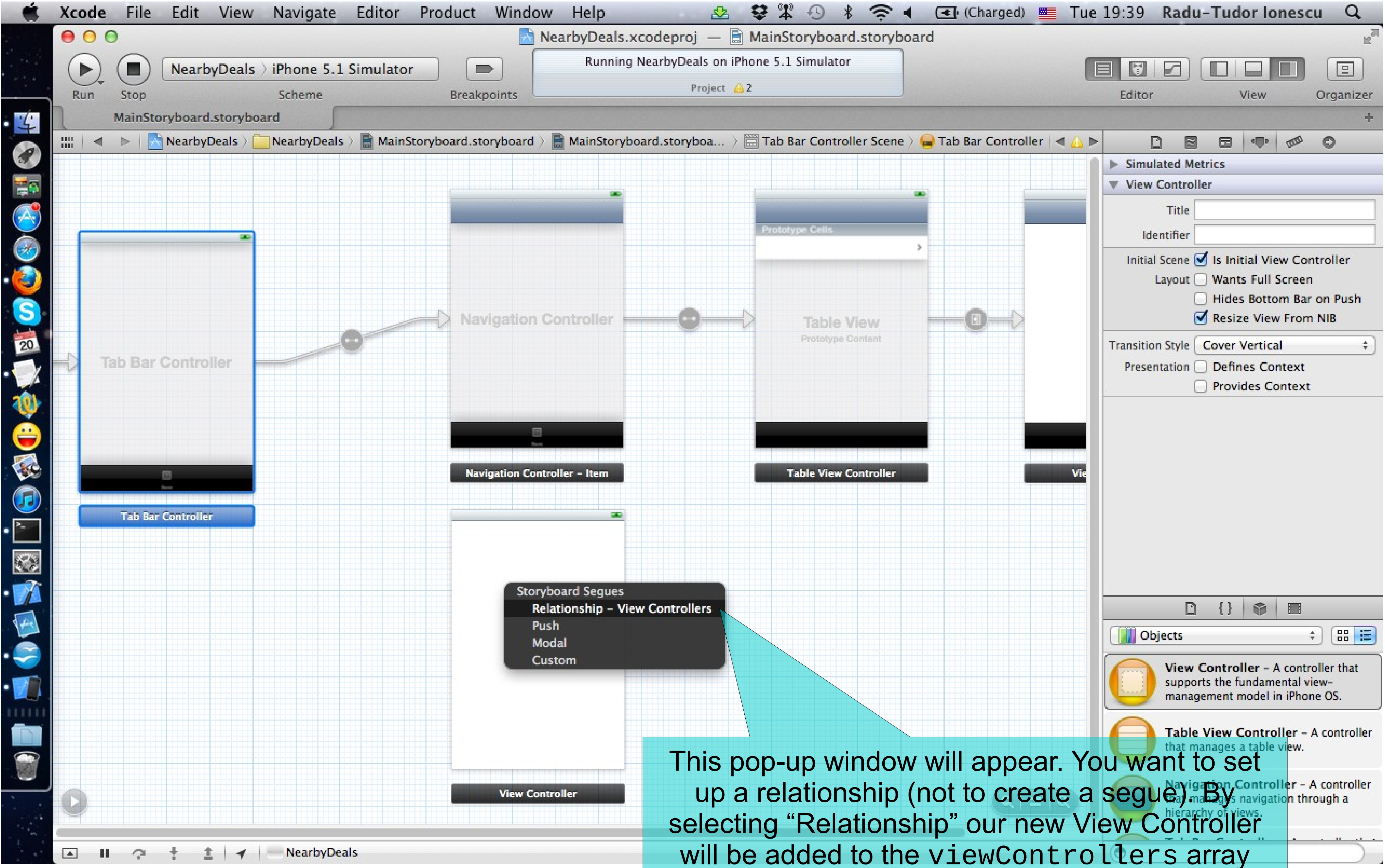
□ Provides Context

Objects

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

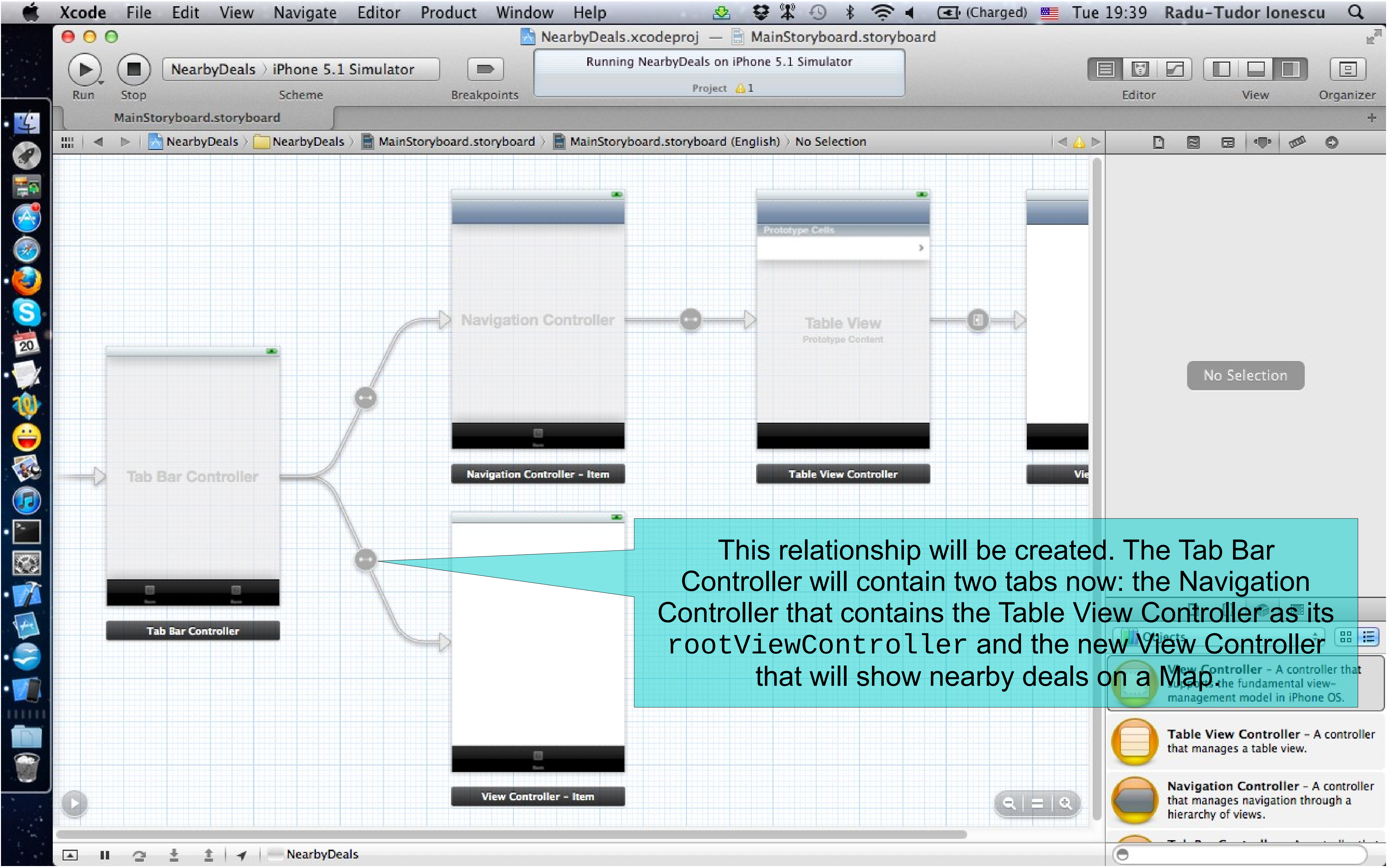
Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 19:39 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals iPhone 5.1 Simulator
Running NearbyDeals on iPhone 5.1 Simulator
Project ⚠2
MainStoryboard.storyboard
NearbyDeals NearbyDeals MainStoryboard.storyboard MainStoryboard.storyboa... Tab Bar Controller Scene Tab Bar Controller
Tab Bar Controller
Navigation Controller
Navigation Controller – Item
Prototype Cells
Table View
Prototype Content
Table View Controller
View Controller
Storyboard Segues
Relationship – View Controllers
Push
Modal
Custom
Simulated Metrics
View Controller
Title
Identifier
Initial Scene ☑ Is Initial View Controller
Layout □ Wants Full Screen
□ Hides Bottom Bar on Push
☑ Resize View From NIB
Transition Style Cover Vertical
Presentation □ Defines Context
□ Provides Context
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.
NearbyDeals

This pop-up window will appear. You want to set up a relationship (not to create a segue). By selecting “Relationship” our new View Controller will be added to the `viewControllers` array of the Tab Bar Controller.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 19:39 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals NearbyDeals MainStoryboard.storyboard MainStoryboard.storyboard (English) No Selection
Navigation Controller
Navigation Controller - Item
Prototype Cells
Table View
Prototype Content
Table View Controller
Tab Bar Controller
Tab Bar Controller
View Controller - Item
No Selection
Table View Controller - A controller that manages a table view.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
NearbyDeals

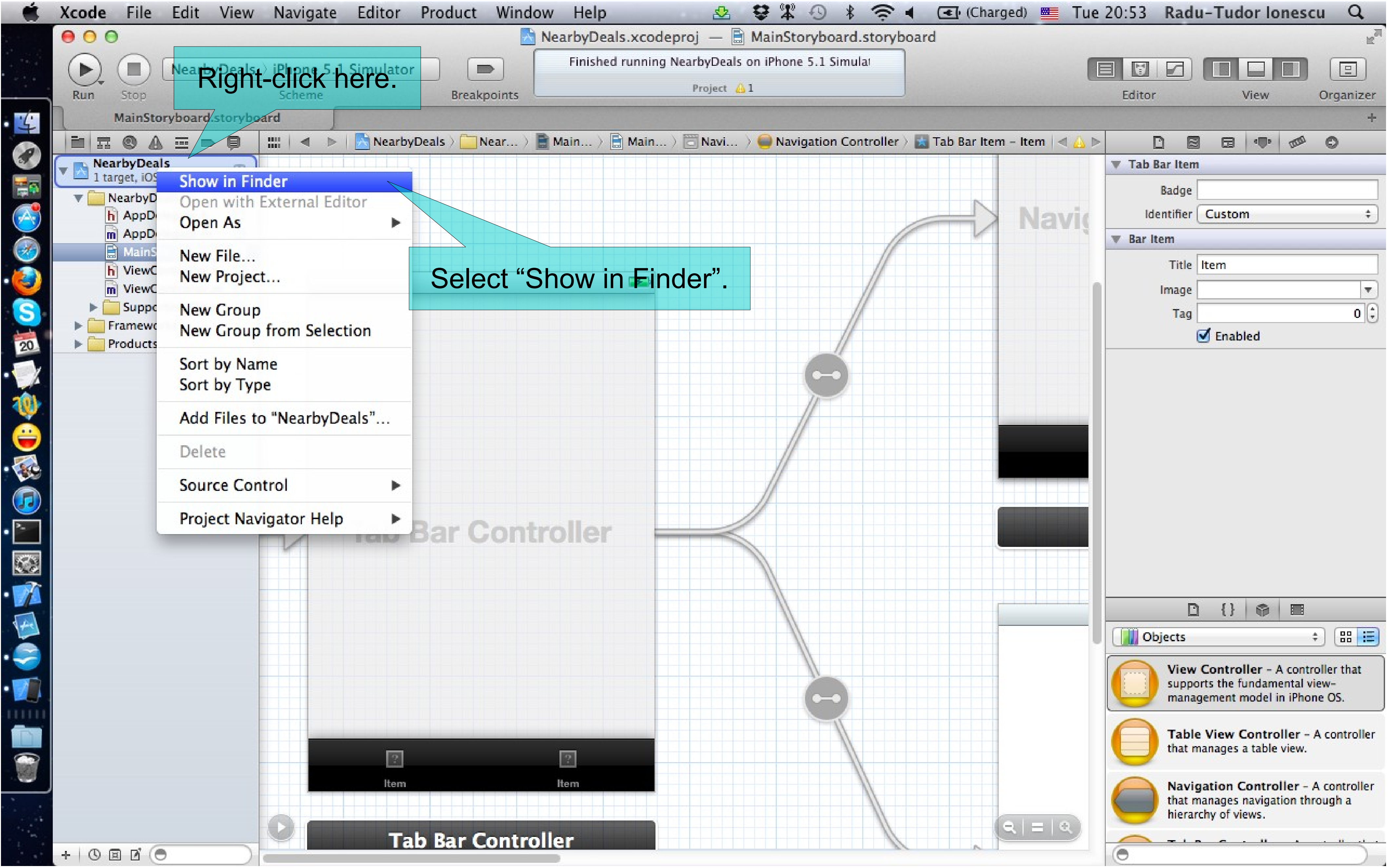
This relationship will be created. The Tab Bar Controller will contain two tabs now: the Navigation Controller that contains the Table View Controller as its `rootViewController` and the new View Controller that will show nearby deals on a Map.

Task 3

Task: Add tab icons for the two tabs of the application.

1. Open Project Navigator and right-click on the NearbyDeals Project.
2. Select the "Show in Finder" option.
3. In Finder create a new folder and name it "Images". We are going to use this folder to put images that we want to add to our Project. It is a good practice to keep a separate subfolder for this.
4. Copy and paste (using CMD + C and CMD + V, respectively) the "tab-icon-list.png" and "tab-icon-map.png" files to the "Images" subfolder. You might want to open another Finder window for this (use the CMD + N shortcut to do it).

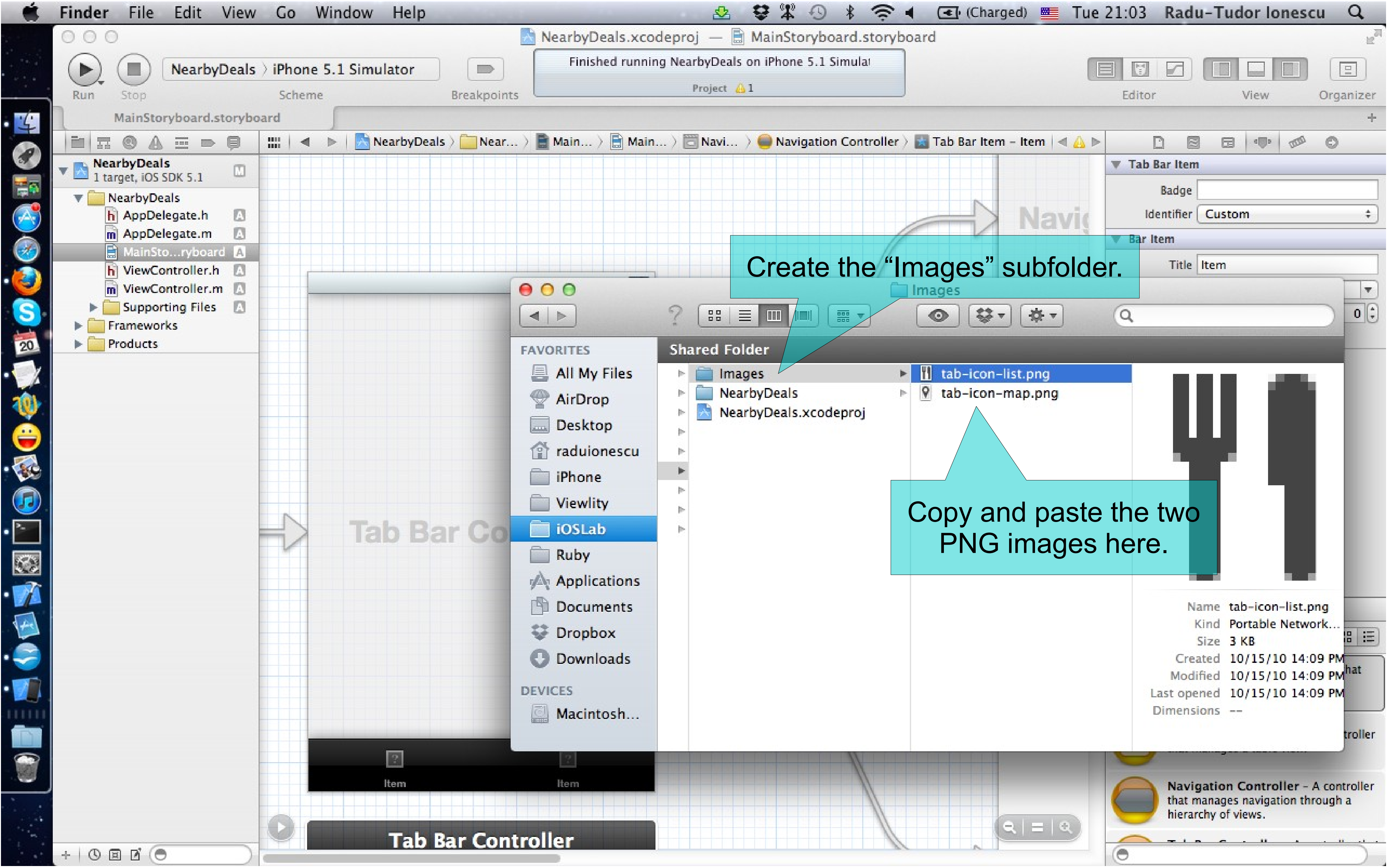
See the next screenshots for extra help.



Right-click here.

Select “Show in Finder”.

Xcode File Edit View Navigate Editor Product Window Help
Tue 20:53 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Finished running NearbyDeals on iPhone 5.1 Simulat
Project ⚠1
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals 〉 Near... 〉 Main... 〉 Main... 〉 Navi... 〉 Navigation Controller 〉 Tab Bar Item – Item
NearbyDeals
1 target, iOS
Show in Finder
Open with External Editor
Open As
New File...
New Project...
New Group
New Group from Selection
Sort by Name
Sort by Type
Add Files to "NearbyDeals"...
Delete
Source Control
Project Navigator Help
Tab Bar Controller
Item
Item
Tab Bar Item
Badge
Identifier Custom
Bar Item
Title Item
Image
Tag 0
Enabled
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.



Create the “Images” subfolder.

Copy and paste the two PNG images here.

Finished running NearbyDeals on iPhone 5.1 Simulat

Badge

Identifier: Custom

Title: Item

Name: tab-icon-list.png

Kind: Portable Network…

Size: 3 KB

Created: 10/15/10 14:09 PM

Modified: 10/15/10 14:09 PM

Last opened: 10/15/10 14:09 PM

Dimensions: --

Task 3

Task: Add tab icons for the two tabs of the application.

5. Close Finder and go back to Xcode. It's time to add the Images subfolder to our Project.

Right-click on the NearbyDeals Project and select the “Add Files to NearbyDeals ...” option.

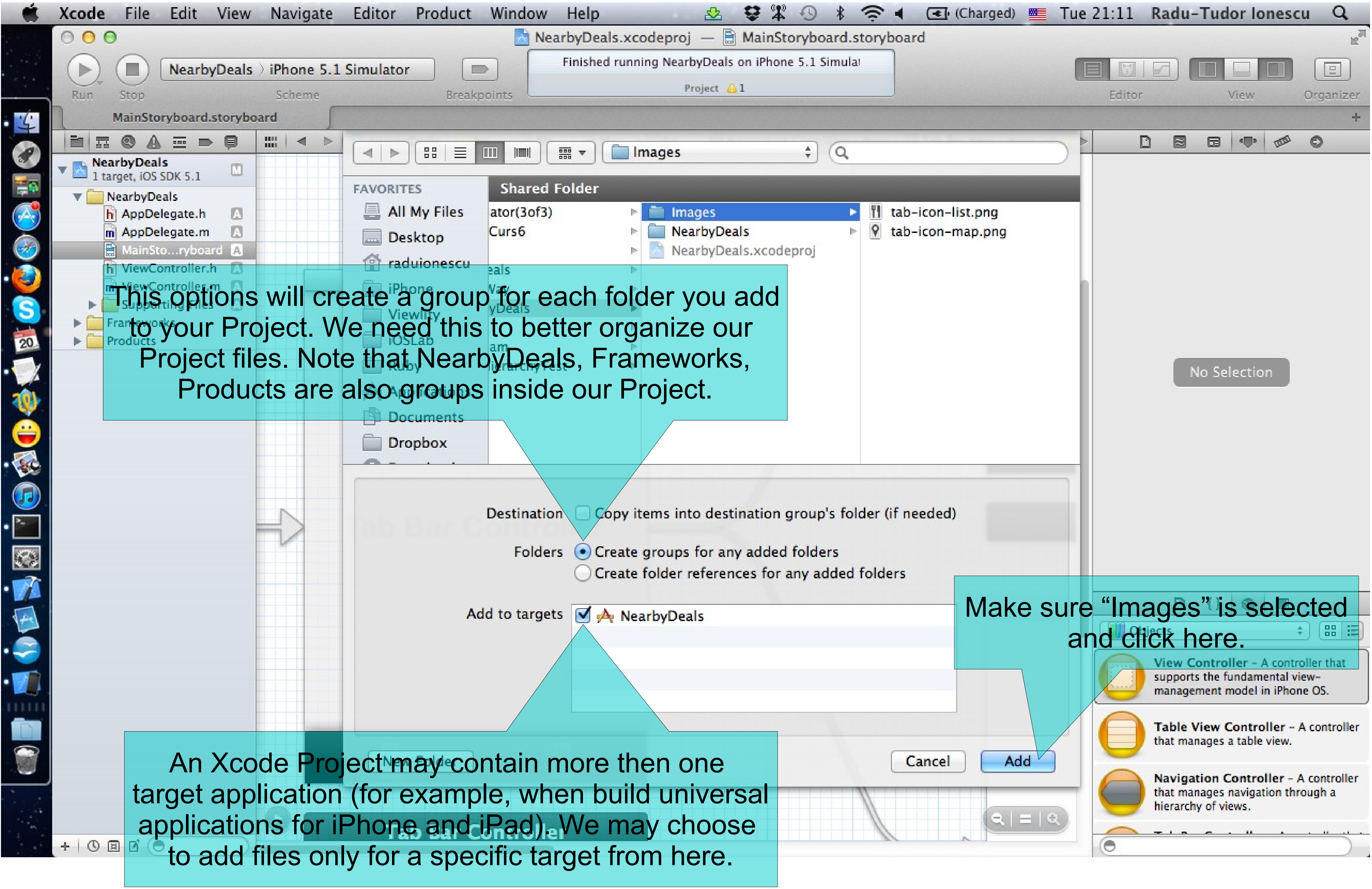
6. Search for the “Images” folder you've just created.

7. Make sure “Create groups for any added folders” is selected.

8. Click “Add” to add the “Images” folder to your project.

9. Make sure the “Images” folder appears in Project Navigator before you continue.

See the next screenshot for extra help.


Xcode File Edit View Navigate Editor Product Window Help
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Finished running NearbyDeals on iPhone 5.1 Simulat
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals
1 target, iOS SDK 5.1
AppDelegate.h
AppDelegate.m
MainSto...ryboard
ViewController.h
ViewController.m
Supporting Files
Frameworks
Products
Images
FAVORITES
All My Files
Desktop
raduionescu
iPhone
Viewlity
iOSLab
Ruby
Applications
Documents
Dropbox
Shared Folder
Images
NearbyDeals
NearbyDeals.xcodeproj
tab-icon-list.png
tab-icon-map.png
No Selection
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.
New Folder
Tab Bar Controller

Destination ☐ Copy items into destination group's folder (if needed)

Folders ● Create groups for any added folders
○ Create folder references for any added folders

Add to targets ☑ NearbyDeals

Cancel Add

This options will create a group for each folder you add to your Project. We need this to better organize our Project files. Note that NearbyDeals, Frameworks, Products are also groups inside our Project.

Make sure "Images" is selected and click here.

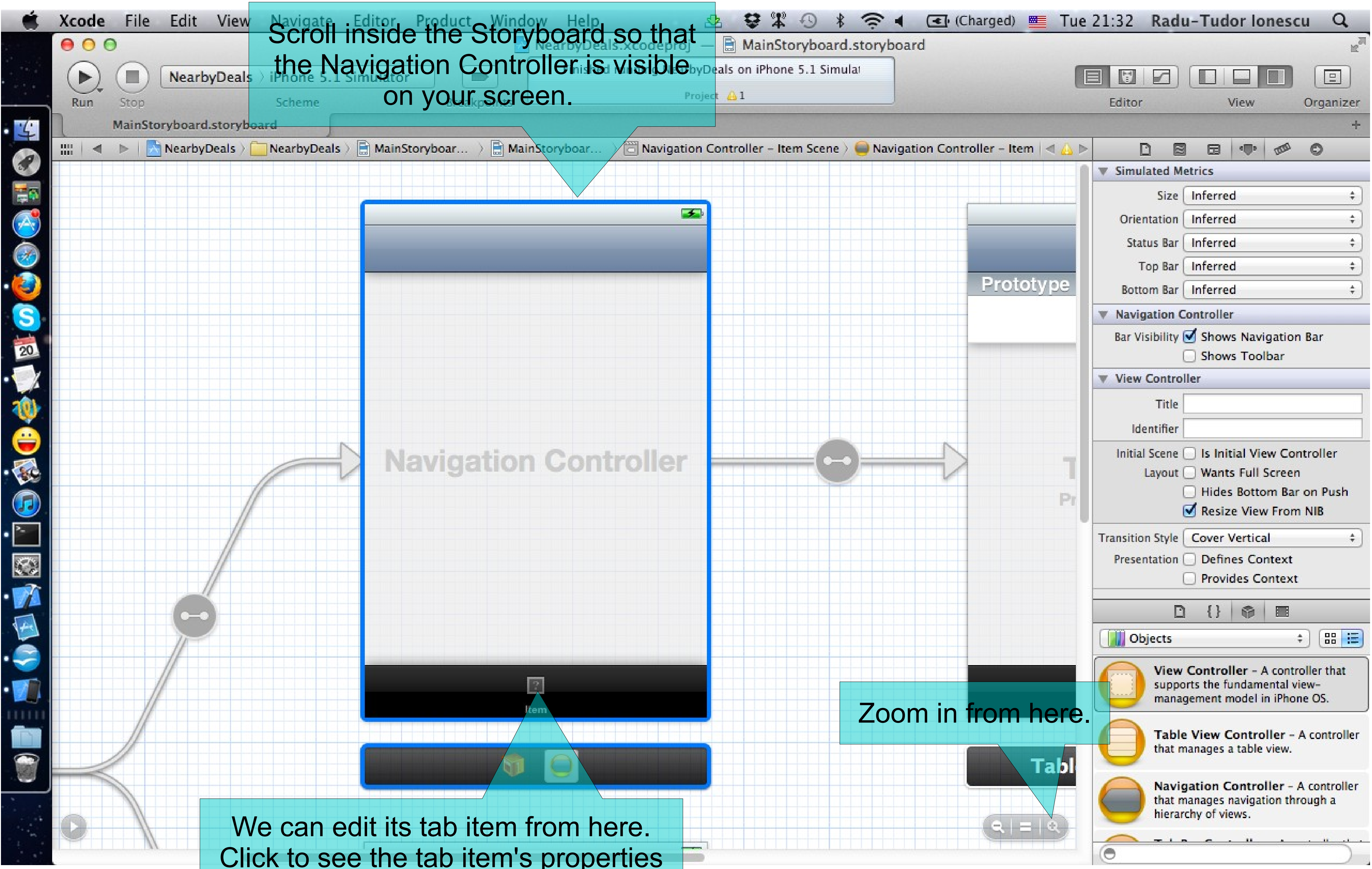
An Xcode Project may contain more then one target application (for example, when build universal applications for iPhone and iPad). We may choose to add files only for a specific target from here.

Task 3

Task: Add tab icons for the two tabs of the application.

10. Hide Project Navigator.

11. Continue with the steps from the following slides to add the tab icons and complete this task.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 21:32 Radu-Tudor Ionescu
Run
Stop
NearbyDeals iPhone 5.1 Simulator
Scheme
Breakpoints
MainStoryboard.storyboard
NearbyDeals on iPhone 5.1 Simulat
Project ⚠1
Editor
View
Organizer
NearbyDeals
MainStoryboar…
Navigation Controller – Item Scene
Navigation Controller – Item
Navigation Controller
Item
Prototype
Tabl
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

Simulated Metrics

- Size: Inferred
- Orientation: Inferred
- Status Bar: Inferred
- Top Bar: Inferred
- Bottom Bar: Inferred

Navigation Controller

- Bar Visibility: ☑ Shows Navigation Bar
- □ Shows Toolbar

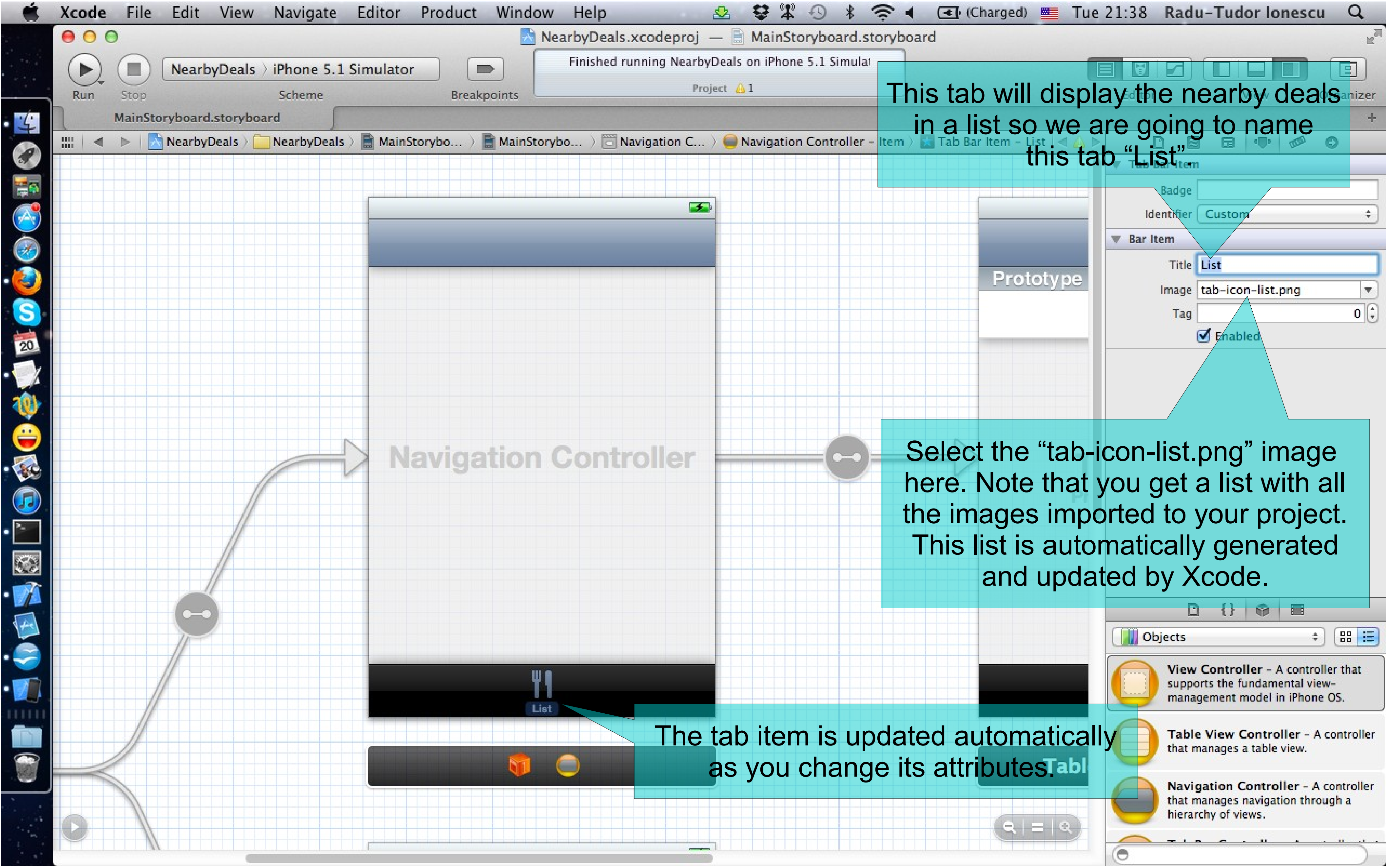
View Controller

- Title:
- Identifier:
- Initial Scene: □ Is Initial View Controller
- Layout: □ Wants Full Screen
- □ Hides Bottom Bar on Push
- ☑ Resize View From NIB
- Transition Style: Cover Vertical
- Presentation: □ Defines Context
- □ Provides Context

Scroll inside the Storyboard so that the Navigation Controller is visible on your screen.

Zoom in from here.

We can edit its tab item from here. Click to see the tab item's properties in Attributes Inspector.

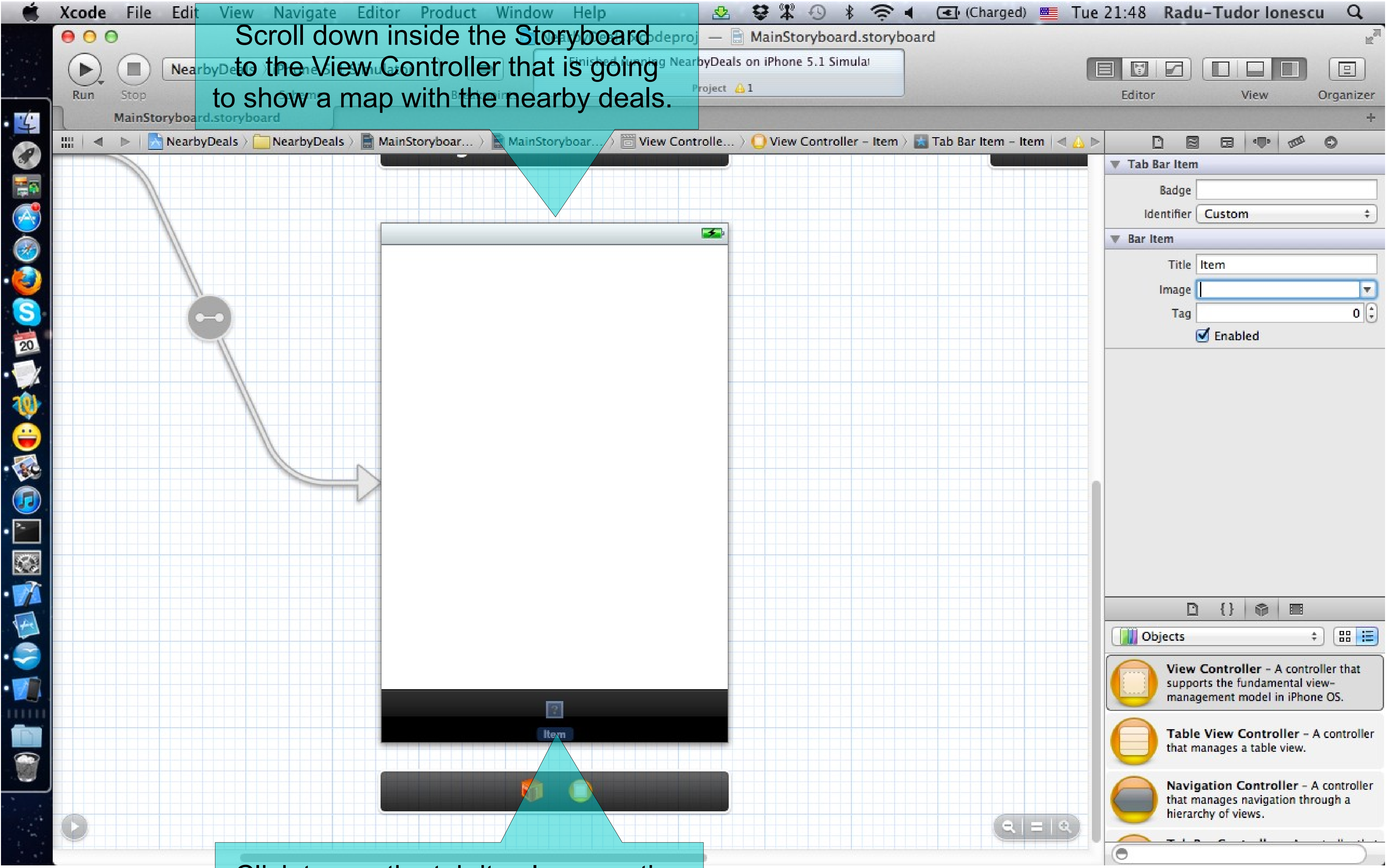


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 21:38 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Finished running NearbyDeals on iPhone 5.1 Simulat
Project ⚠1
MainStoryboard.storyboard
NearbyDeals 〉 NearbyDeals 〉 MainStorybo... 〉 MainStorybo... 〉 Navigation C... 〉 Navigation Controller – Item 〉 Tab Bar Item – List
Navigation Controller
Prototype
List
Tab Bar Item
Badge
Identifier Custom
Bar Item
Title List
Image tab-icon-list.png
Tag 0
☑ Enabled
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

This tab will display the nearby deals in a list so we are going to name this tab "List".

Select the "tab-icon-list.png" image here. Note that you get a list with all the images imported to your project. This list is automatically generated and updated by Xcode.

The tab item is updated automatically as you change its attributes.



Scroll down inside the Storyboard to the View Controller that is going to show a map with the nearby deals.

Tab Bar Item

Badge

Identifier Custom

Bar Item

Title Item

Image

Tag 0

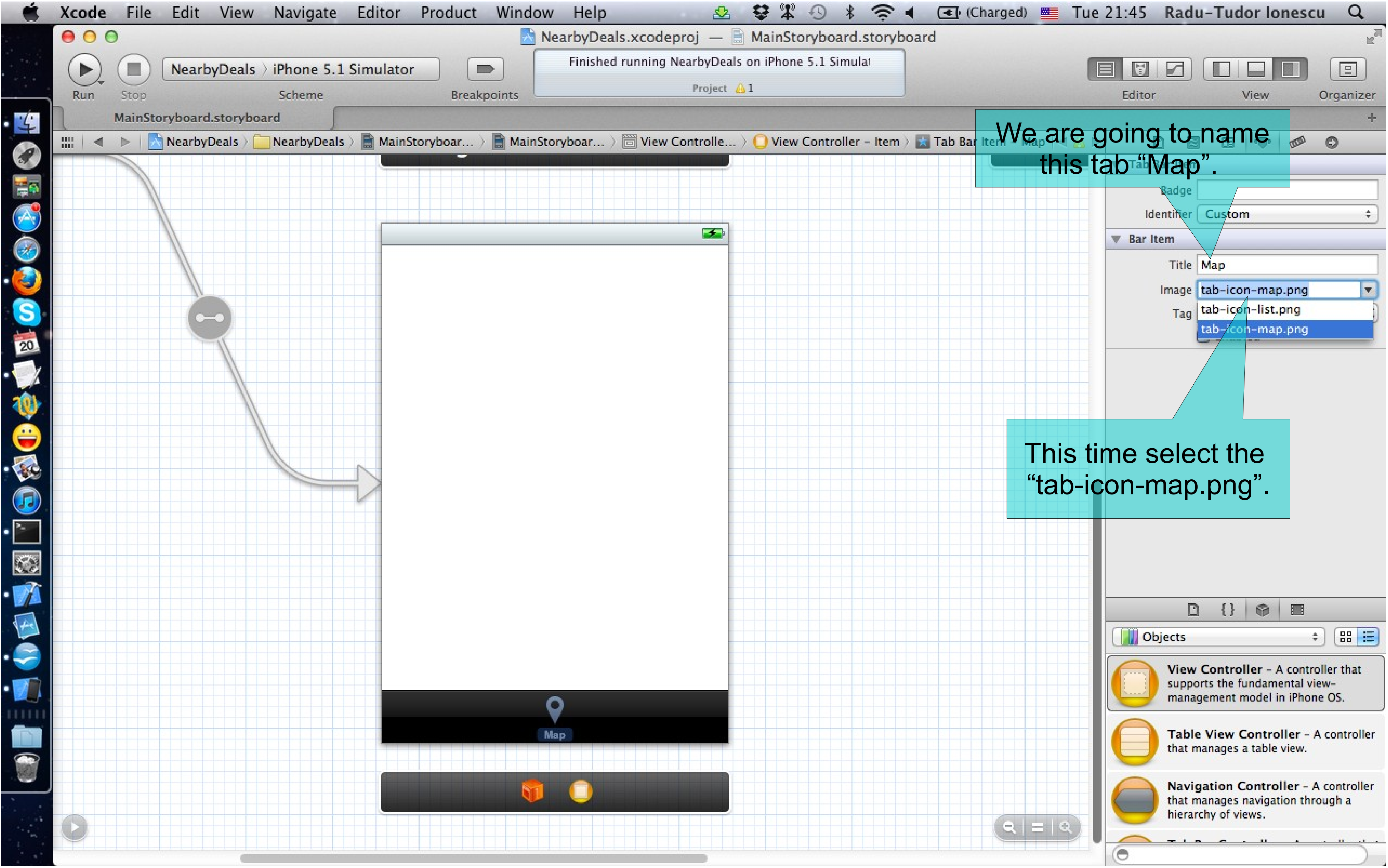
☑ Enabled

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Click to see the tab item's properties in Attributes Inspector.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 21:45 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals 〉 NearbyDeals 〉 MainStoryboar… 〉 MainStoryboar… 〉 View Controlle… 〉 View Controller – Item 〉 Tab Bar Item – Map
Badge
Bar Item
Map
Objects

Finished running NearbyDeals on iPhone 5.1 Simulat

Project ⚠ 1

Identifier: Custom

Title: Map

Image: tab-icon-map.png

Tag

tab-icon-list.png
tab-icon-map.png

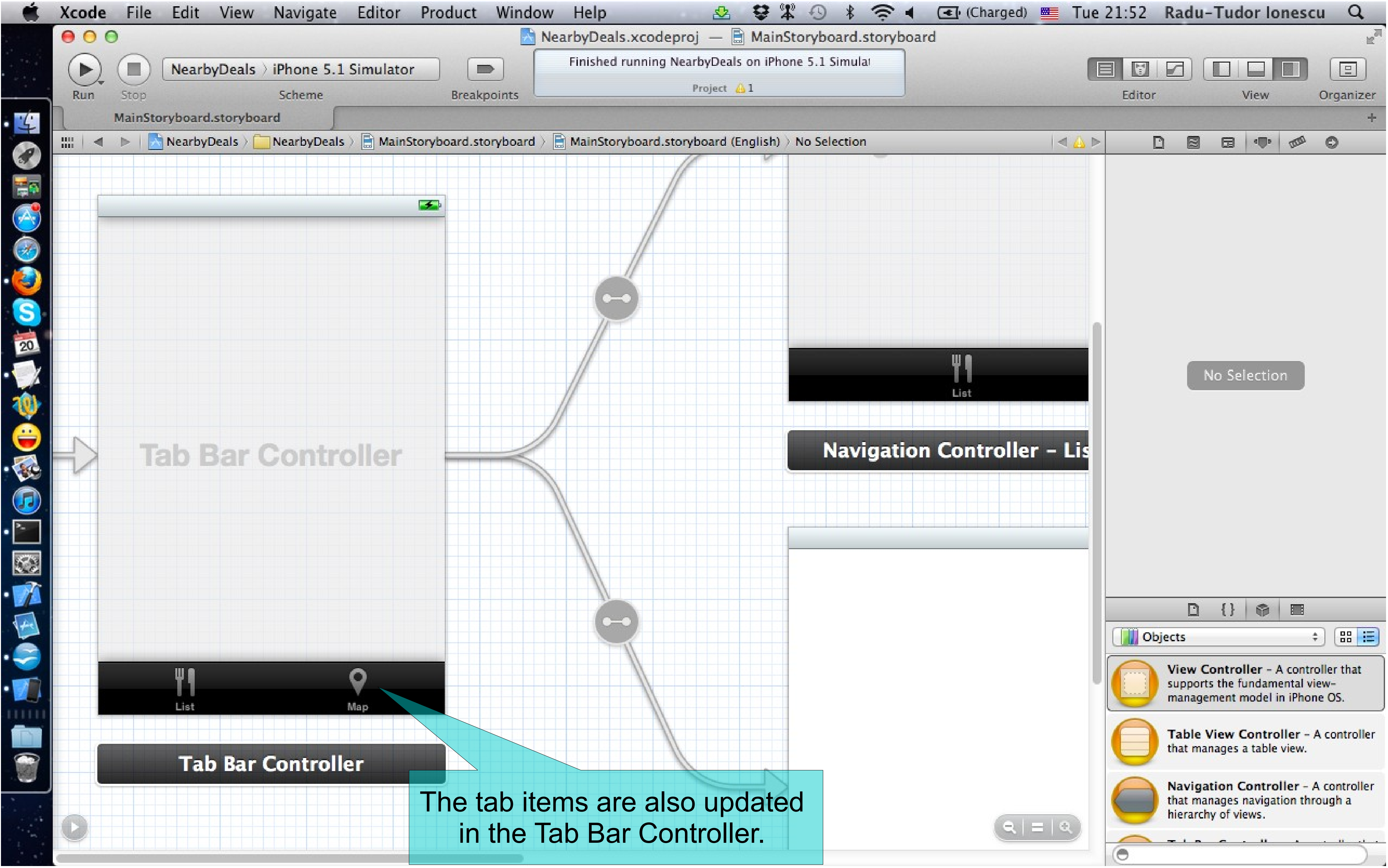
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

We are going to name this tab “Map”.

This time select the “tab-icon-map.png”.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 21:52 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals iPhone 5.1 Simulator
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals NearbyDeals MainStoryboard.storyboard MainStoryboard.storyboard (English) No Selection
Tab Bar Controller
List
Map
Tab Bar Controller
List
Navigation Controller – Lis
No Selection
Objects

Finished running NearbyDeals on iPhone 5.1 Simulat

Project ⚠1

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

The tab items are also updated in the Tab Bar Controller.

Task 4

Task: Configure the Table View Controller to show some mock-up data inside its Table View.

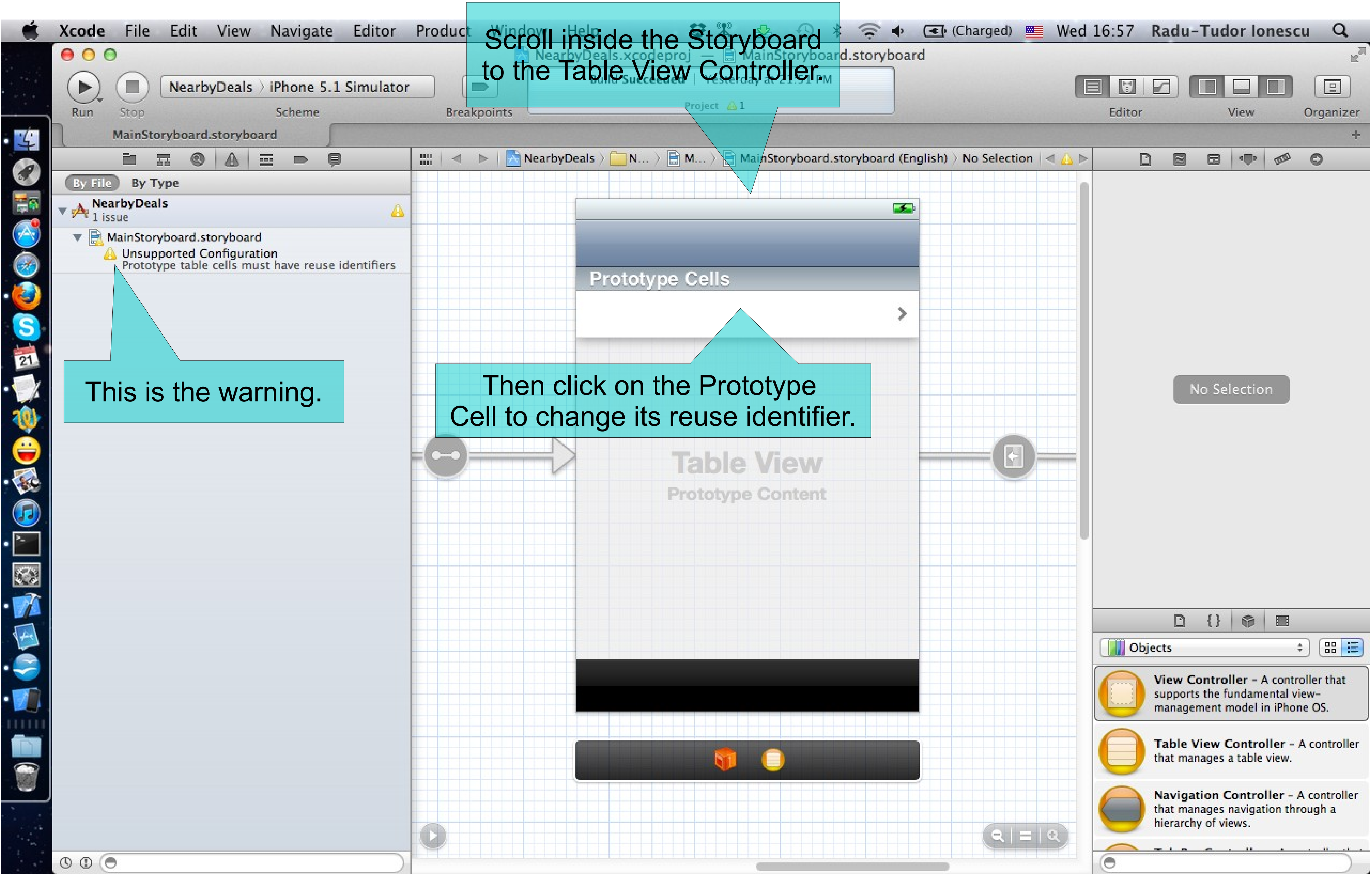
1. Note that there is a Xcode Warning that tells us that we need to set the reuse identifier of the Prototype Cell.

The reuse identifier is associated with a `UITableViewCell` object that the Table View's delegate creates with the intent to reuse it as the basis (for performance reasons) for multiple rows of a table view.

It is assigned to the cell object in the initializer method `initWithFrame:reuseIdentifier:` and cannot be changed thereafter. A `UITableView` object maintains a queue (or list) of the currently reusable cells, each with its own reuse identifier, and makes them available to the delegate.

The reuse identifier is just an `NSString` object that we can set up. It will be used to identify a type of cell.

Let's set the reuse identifier to "DealCell". See the next slides for help.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Wed 16:57 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 21:51 PM
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
By File
By Type
NearbyDeals › N… › M… › MainStoryboard.storyboard (English) › No Selection
Prototype Cells
Table View
Prototype Content
No Selection
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

NearbyDeals
1 issue

MainStoryboard.storyboard

⚠ Unsupported Configuration
Prototype table cells must have reuse identifiers

Scroll inside the Storyboard to the Table View Controller.

This is the warning.

Then click on the Prototype Cell to change its reuse identifier.

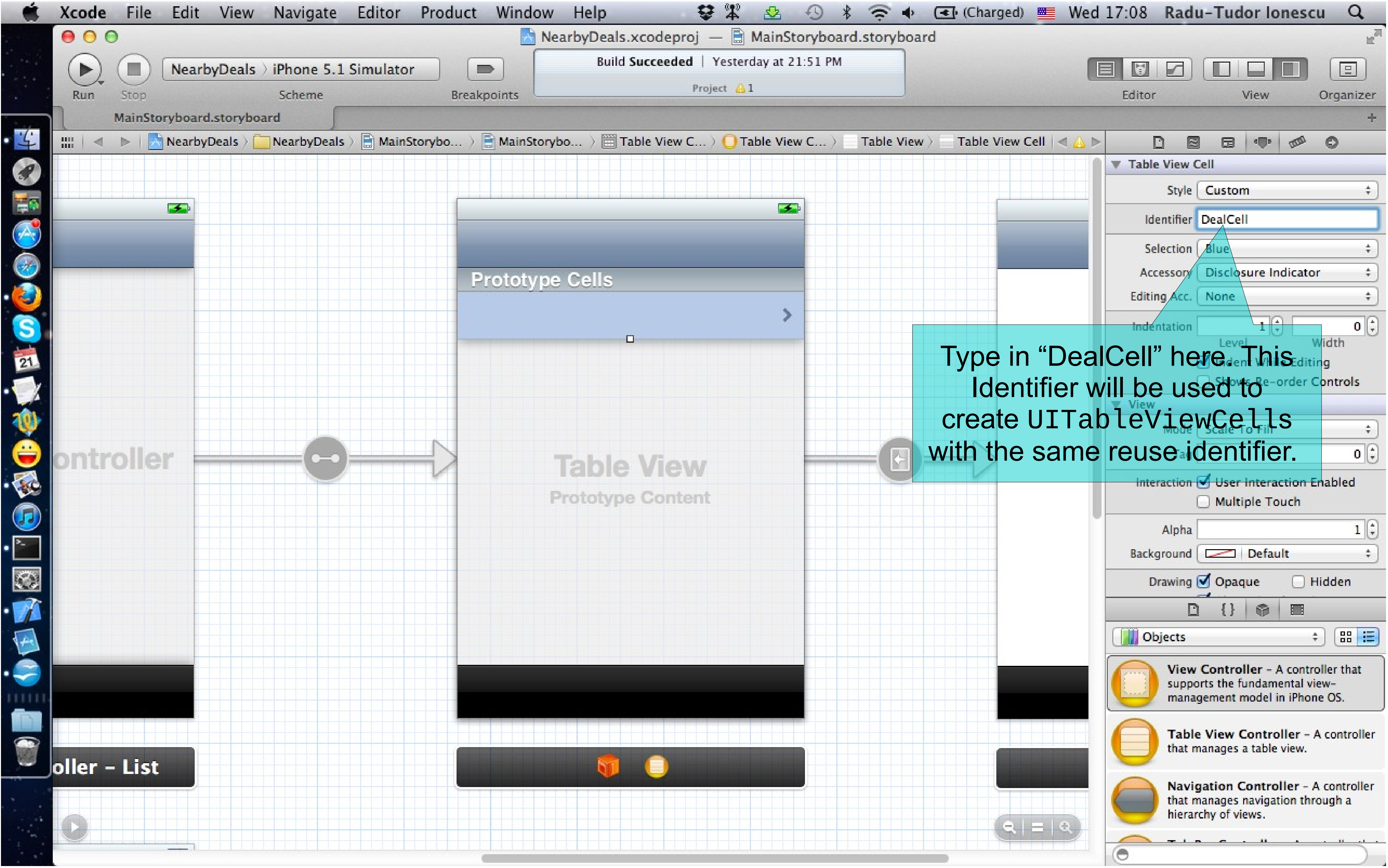

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Wed 17:08 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 21:51 PM
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals 〉 NearbyDeals 〉 MainStorybo... 〉 MainStorybo... 〉 Table View C... 〉 Table View C... 〉 Table View 〉 Table View Cell
Prototype Cells
Table View
Prototype Content
ontroller
oller – List
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

Table View Cell

Style: Custom

Identifier: DealCell

Selection: Blue

Accessory: Disclosure Indicator

Editing Acc.: None

Indentation: 1 (Level) 0 (Width)

☐ Indent While Editing

☐ Shows Re-order Controls

View

Mode: Scale To Fill

Tag: 0

Interaction: ☑ User Interaction Enabled

☐ Multiple Touch

Alpha: 1

Background: Default

Drawing: ☑ Opaque ☐ Hidden

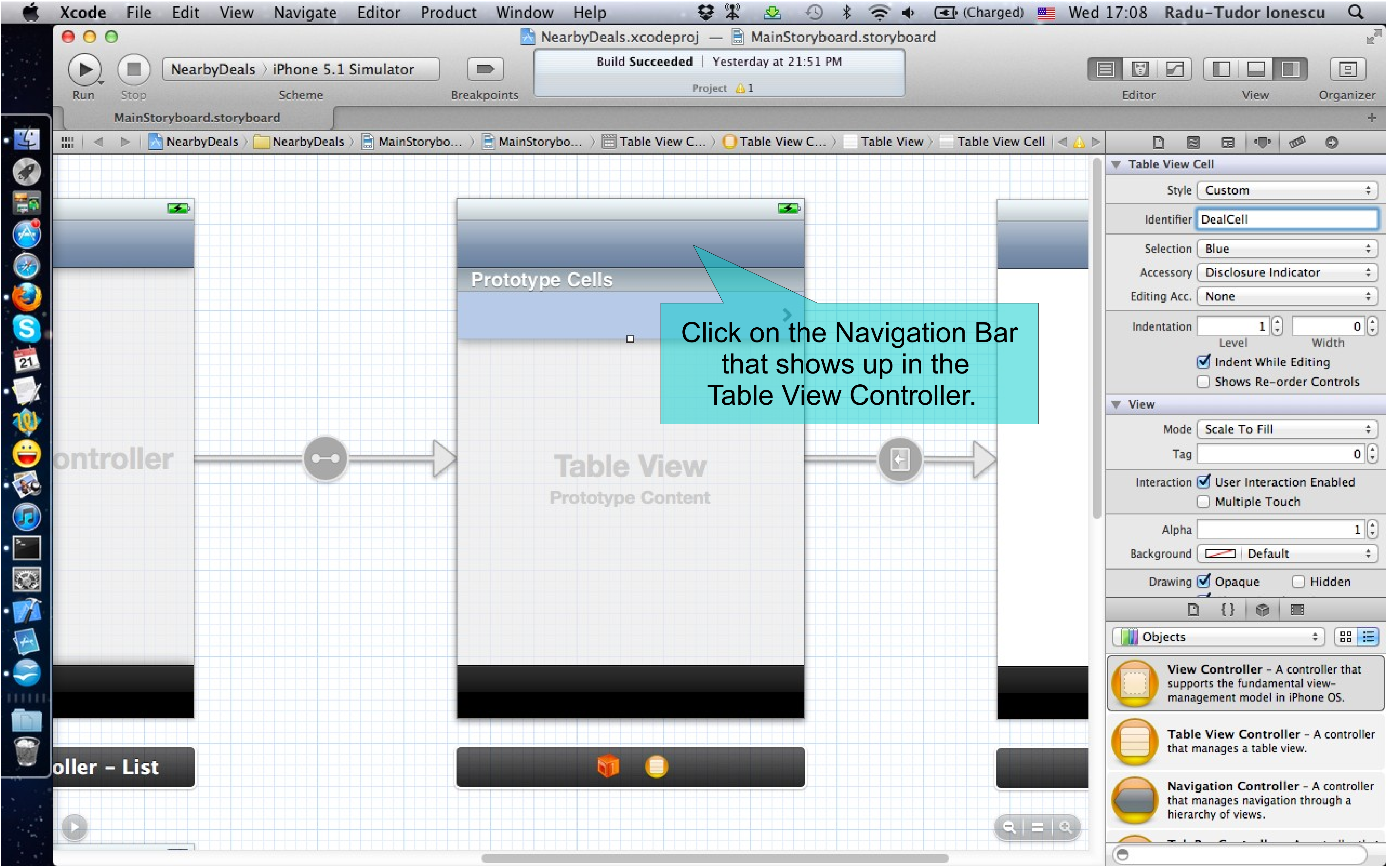
Type in “DealCell” here. This Identifier will be used to create `UITableViewCells` with the same reuse identifier.

Task 4

Task: Configure the Table View Controller to show some mock-up data inside its Table View.

2. We should set the Table View Controller title (that appears on the Navigation Bar). Note that each View Controller (including Table View Controllers) have a `navigationItem` that holds properties related to navigation.

We have to set the `navigationItem.title` property to “Nearby Deals”, but we are going to do this in Interface Builder. See the following slides to understand what needs to be done.



Click on the Navigation Bar that shows up in the Table View Controller.

NearbyDeals.xcodeproj — MainStoryboard.storyboard

Build Succeeded | Yesterday at 21:51 PM

Table View Cell

Style: Custom

Identifier: DealCell

Selection: Blue

Accessory: Disclosure Indicator

Editing Acc.: None

Indentation: 1 Level, 0 Width

☑ Indent While Editing

□ Shows Re-order Controls

View

Mode: Scale To Fill

Tag: 0

Interaction: ☑ User Interaction Enabled

□ Multiple Touch

Alpha: 1

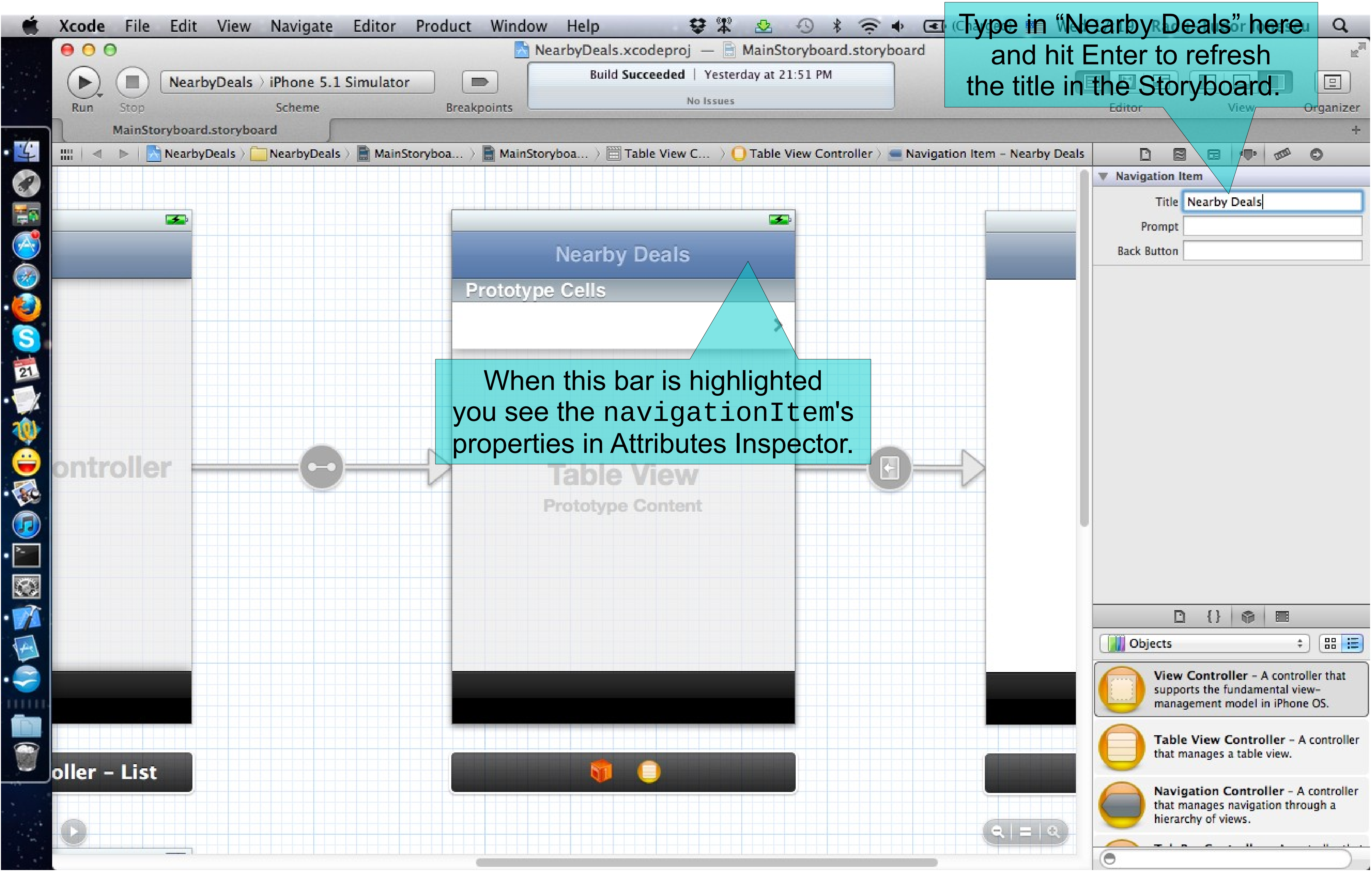
Background: Default

Drawing: ☑ Opaque □ Hidden

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.



Type in “Nearby Deals” here and hit Enter to refresh the title in the Storyboard.

When this bar is highlighted you see the `navigationItem`'s properties in Attributes Inspector.

NearbyDeals.xcodeproj — MainStoryboard.storyboard

Build Succeeded | Yesterday at 21:51 PM

No Issues

Navigation Item

Title: Nearby Deals

Prompt:

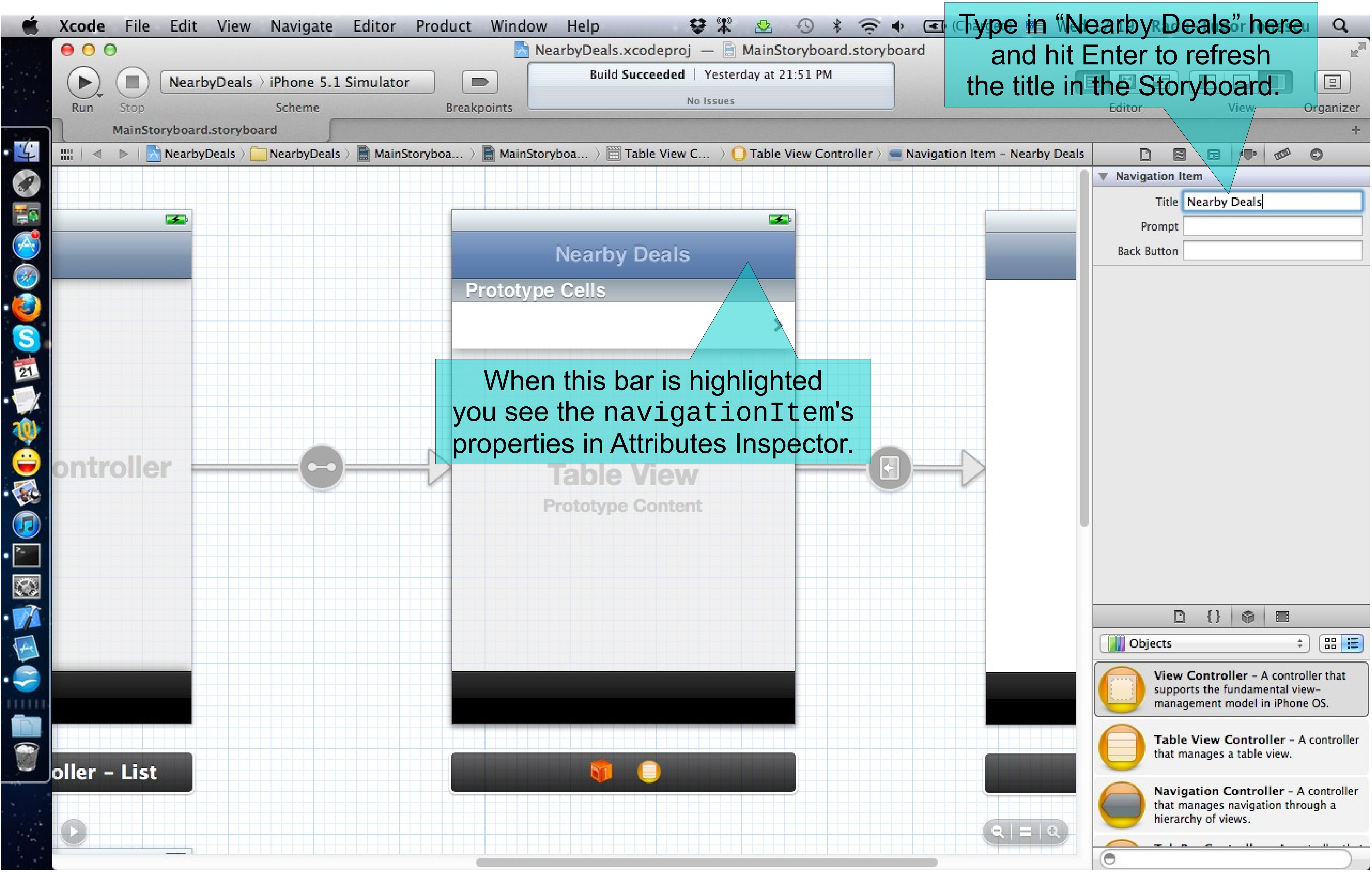
Back Button:

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals 〉 NearbyDeals 〉 MainStoryboa... 〉 MainStoryboa... 〉 Table View C... 〉 Table View Controller 〉 Navigation Item – Nearby Deals
Nearby Deals
Prototype Cells
Table View
Prototype Content
ontroller
oller – List
Objects



Xcode File Edit View Navigate Editor Product Window Help
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 21:51 PM
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboa… › MainStoryboa… › Table View C… › Table View Controller › Navigation Item – Nearby Deals
Nearby Deals
Prototype Cells
Table View
Prototype Content
ontroller
oller – List
Navigation Item
Prompt
Back Button
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

Type in “Nearby Deals” here and hit Enter to refresh the title in the Storyboard.

Title: Nearby Deals

When this bar is highlighted you see the `navigationItem`'s properties in Attributes Inspector.

Task 4

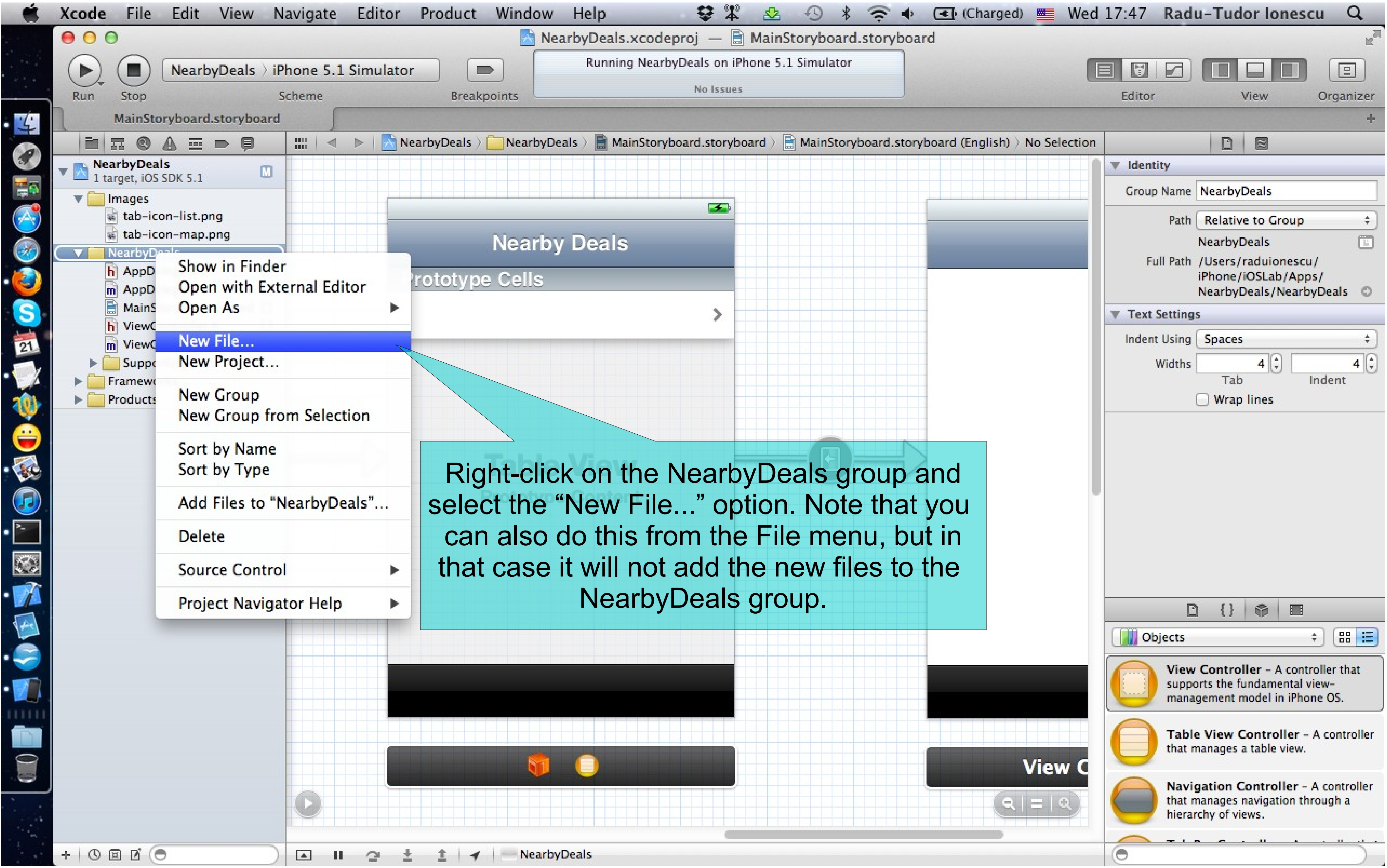
Task: Configure the Table View Controller to show some mock-up data inside its Table View.

3. Let's run the application and see how it looks by now. Notice that it has two tabs: one that shows an empty list of deals and another one that shows a white screen.

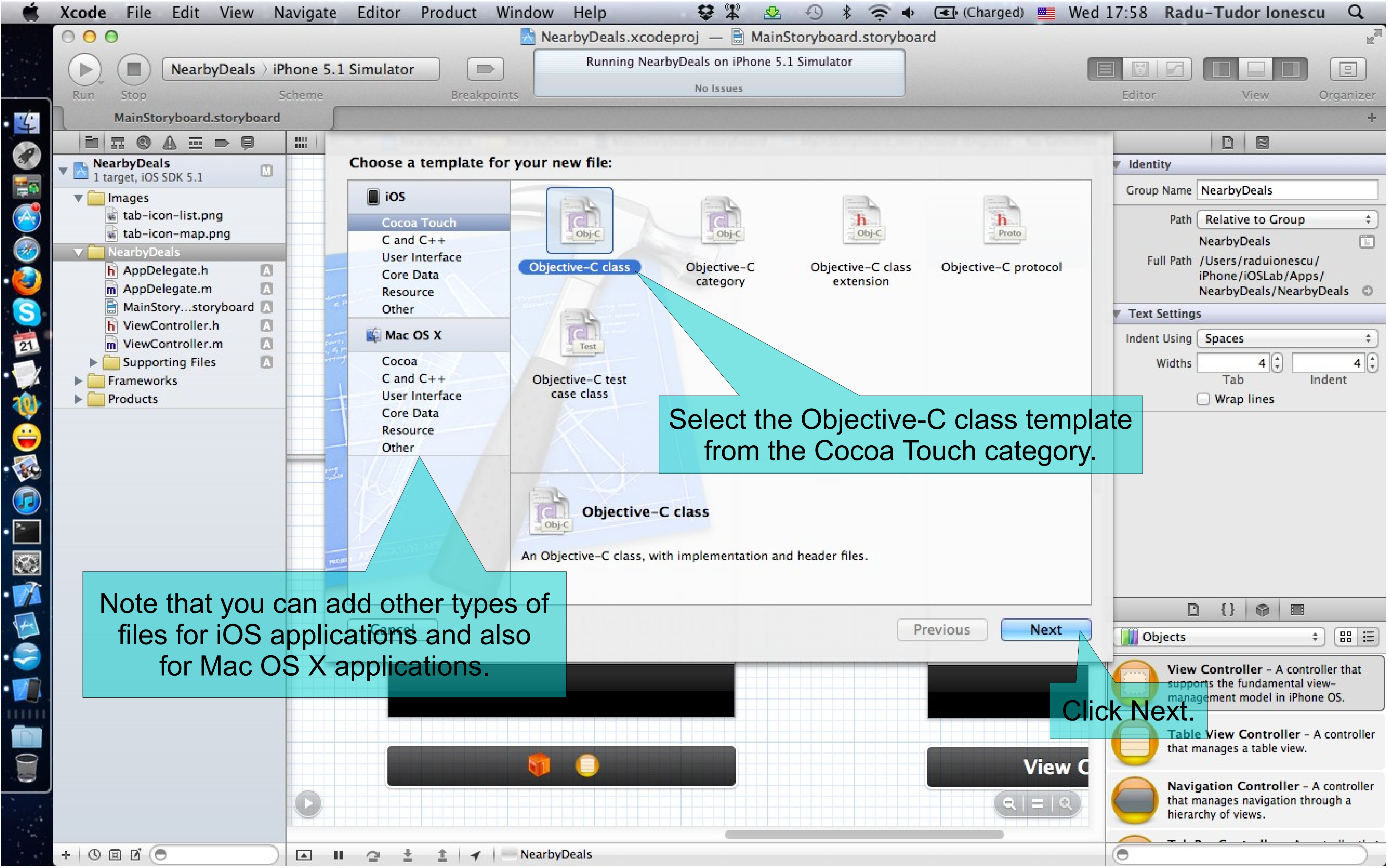
4. Stop running the application.

5. This all we can do from Interface Builder. Next we are going to have to write some code to show some mock-up data inside the Table View Controller. For this we need add a subclass of `UITableViewController` to our project and create a relationship between this subclass and the Table View Controller inside our Storyboard.

Let's open Project Navigator and continue with the following screenshots that guide you through adding a subclass of `UITableViewController`.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Wed 17:47 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › No Selection
NearbyDeals
1 target, iOS SDK 5.1
Images
tab-icon-list.png
tab-icon-map.png
NearbyDeals
AppD
AppD
MainS
ViewC
ViewC
Supp
Framew
Products
Show in Finder
Open with External Editor
Open As
New File...
New Project...
New Group
New Group from Selection
Sort by Name
Sort by Type
Add Files to "NearbyDeals"...
Delete
Source Control
Project Navigator Help
Nearby Deals
Prototype Cells
View C
NearbyDeals
Identity
Group Name NearbyDeals
Path Relative to Group
NearbyDeals
Full Path /Users/raduionescu/iPhone/iOSLab/Apps/NearbyDeals/NearbyDeals
Text Settings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines
Objects
View Controller - A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller - A controller that manages a table view.
Navigation Controller - A controller that manages navigation through a hierarchy of views.

Right-click on the NearbyDeals group and select the “New File...” option. Note that you can also do this from the File menu, but in that case it will not add the new files to the NearbyDeals group.



Choose a template for your new file:

Objective-C class

An Objective-C class, with implementation and header files.

Group Name: NearbyDeals

Path: Relative to Group

NearbyDeals

Full Path: /Users/raduionescu/iPhone/iOSLab/Apps/NearbyDeals/NearbyDeals

Indent Using: Spaces

Widths: 4 (Tab) 4 (Indent)

□ Wrap lines

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

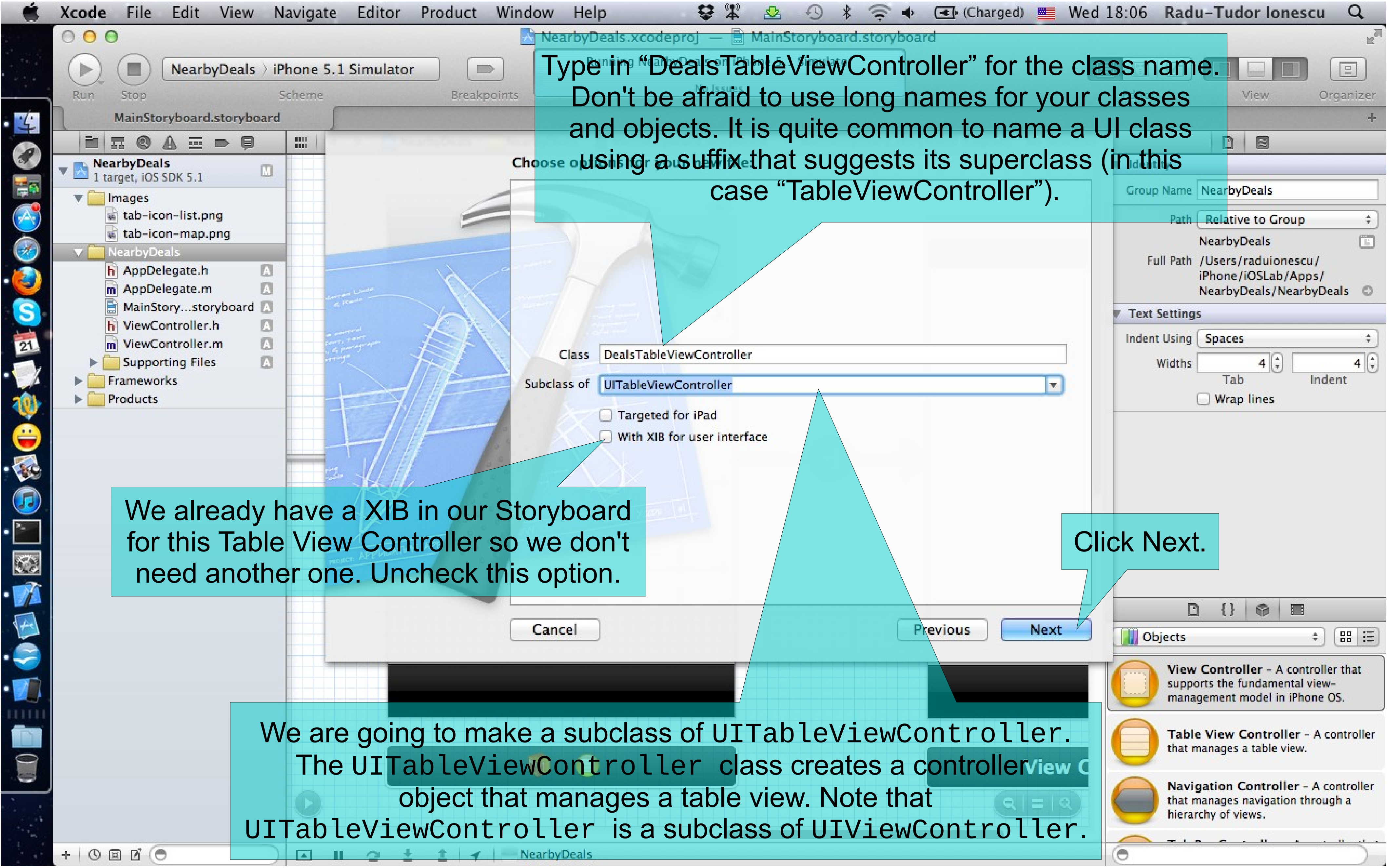
Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Select the Objective-C class template from the Cocoa Touch category.

Note that you can add other types of files for iOS applications and also for Mac OS X applications.

Click Next.



Xcode File Edit View Navigate Editor Product Window Help
Wed 18:06 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
View
Organizer
MainStoryboard.storyboard
NearbyDeals
1 target, iOS SDK 5.1
Images
tab-icon-list.png
tab-icon-map.png
NearbyDeals
AppDelegate.h
AppDelegate.m
MainStory...storyboard
ViewController.h
ViewController.m
Supporting Files
Frameworks
Products
Group Name NearbyDeals
Path Relative to Group
Full Path /Users/raduionescu/iPhone/iOSLab/Apps/NearbyDeals/NearbyDeals
Text Settings
Indent Using Spaces
Widths 4 Tab 4 Indent
Wrap lines
Objects
View Controller - A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller - A controller that manages a table view.
Navigation Controller - A controller that manages navigation through a hierarchy of views.

Choose options for your new file:

Class: DealsTableViewController

Subclass of: UITableViewController

☐ Targeted for iPad

☐ With XIB for user interface

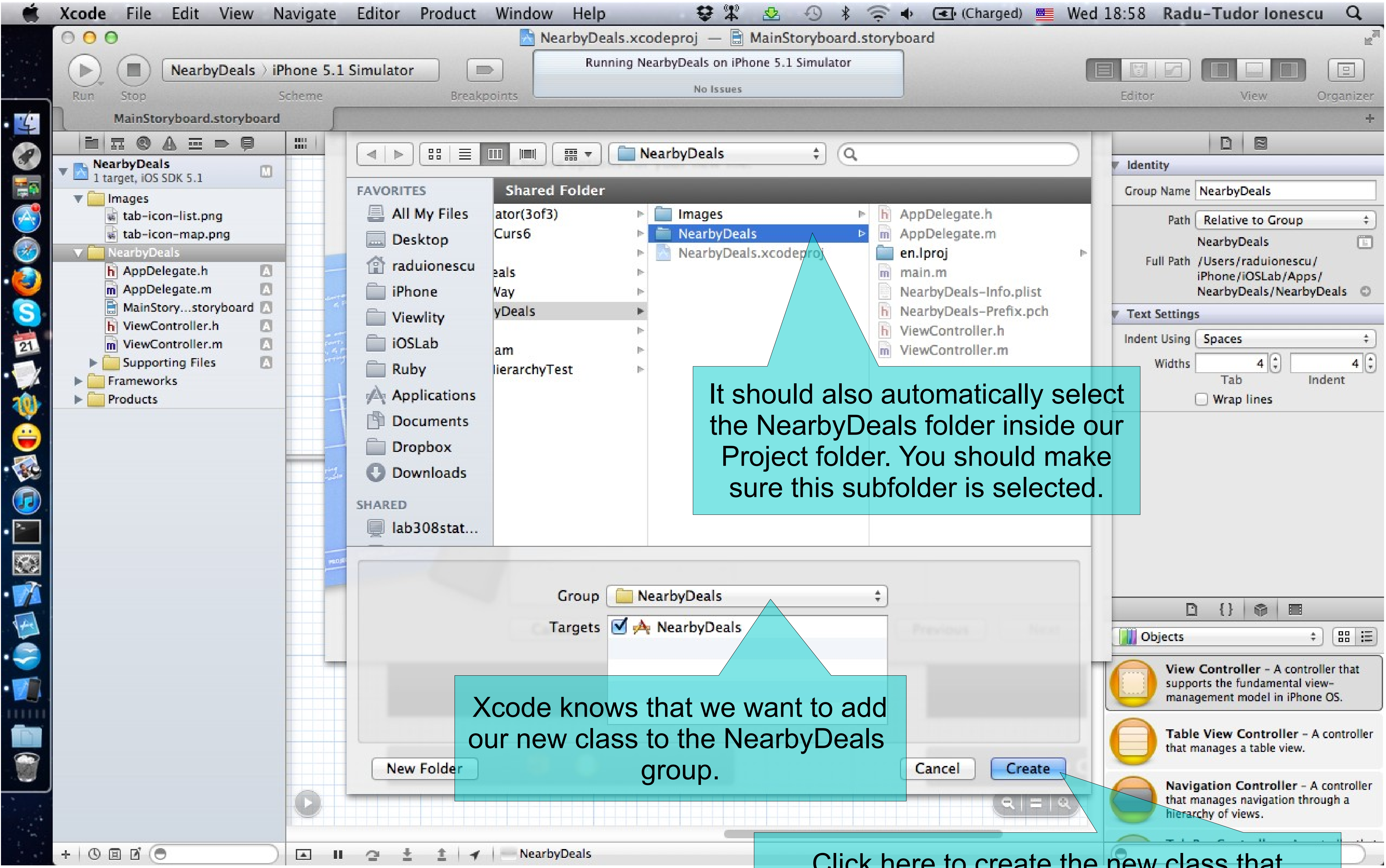
Cancel | Previous | Next

Type in “DealsTableViewController” for the class name. Don't be afraid to use long names for your classes and objects. It is quite common to name a UI class using a suffix that suggests its superclass (in this case “TableViewController”).

We already have a XIB in our Storyboard for this Table View Controller so we don't need another one. Uncheck this option.

Click Next.

We are going to make a subclass of `UITableViewController`. The `UITableViewController` class creates a controller object that manages a table view. Note that `UITableViewController` is a subclass of `UIViewController`.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Wed 18:58 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals › iPhone 5.1 Simulator
Running NearbyDeals on iPhone 5.1 Simulator
No Issues
MainStoryboard.storyboard
NearbyDeals
1 target, iOS SDK 5.1
Images
tab-icon-list.png
tab-icon-map.png
NearbyDeals
AppDelegate.h
AppDelegate.m
MainStory...storyboard
ViewController.h
ViewController.m
Supporting Files
Frameworks
Products
FAVORITES
All My Files
Desktop
raduionescu
iPhone
Viewlity
iOSLab
Ruby
Applications
Documents
Dropbox
Downloads
SHARED
lab308stat...
Shared Folder
Images
NearbyDeals
NearbyDeals.xcodeproj
AppDelegate.h
AppDelegate.m
en.lproj
main.m
NearbyDeals-Info.plist
NearbyDeals-Prefix.pch
ViewController.h
ViewController.m
New Folder
Cancel
Create
Identity
Text Settings
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.

Group: NearbyDeals

Targets: ☑ NearbyDeals

Group Name: NearbyDeals

Path: Relative to Group

Full Path: /Users/raduionescu/iPhone/iOSLab/Apps/NearbyDeals/NearbyDeals

Indent Using: Spaces

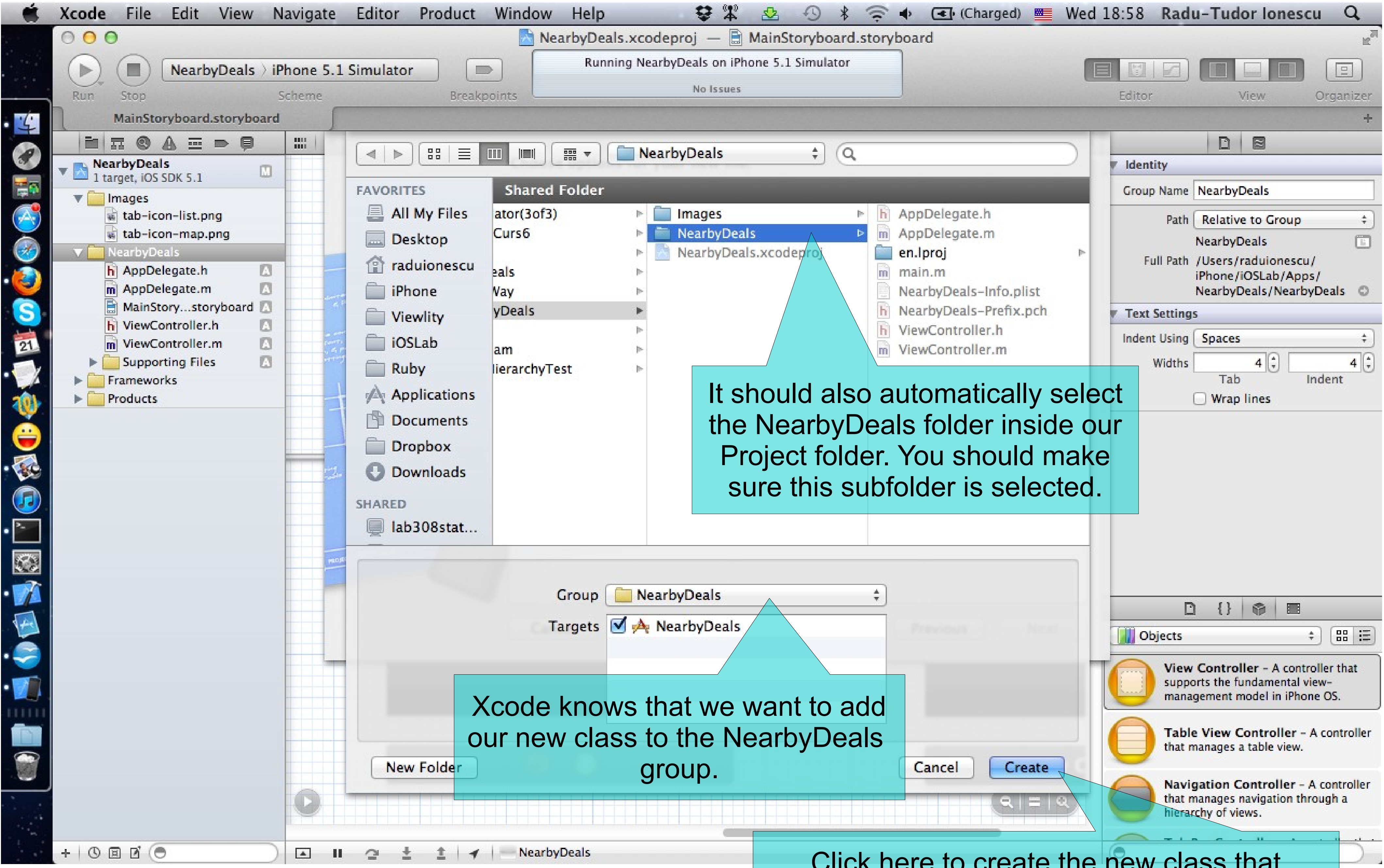
Widths: 4 Tab, 4 Indent

□ Wrap lines

It should also automatically select the NearbyDeals folder inside our Project folder. You should make sure this subfolder is selected.

Xcode knows that we want to add our new class to the NearbyDeals group.

Click here to create the new class that is a subclass of `UITableViewController`.



NearbyDeals

Group: NearbyDeals

Targets: ☑ NearbyDeals

Group Name: NearbyDeals

Path: Relative to Group
NearbyDeals

Full Path: /Users/raduionescu/iPhone/iOSLab/Apps/NearbyDeals/NearbyDeals

Indent Using: Spaces

Widths: 4 Tab, 4 Indent

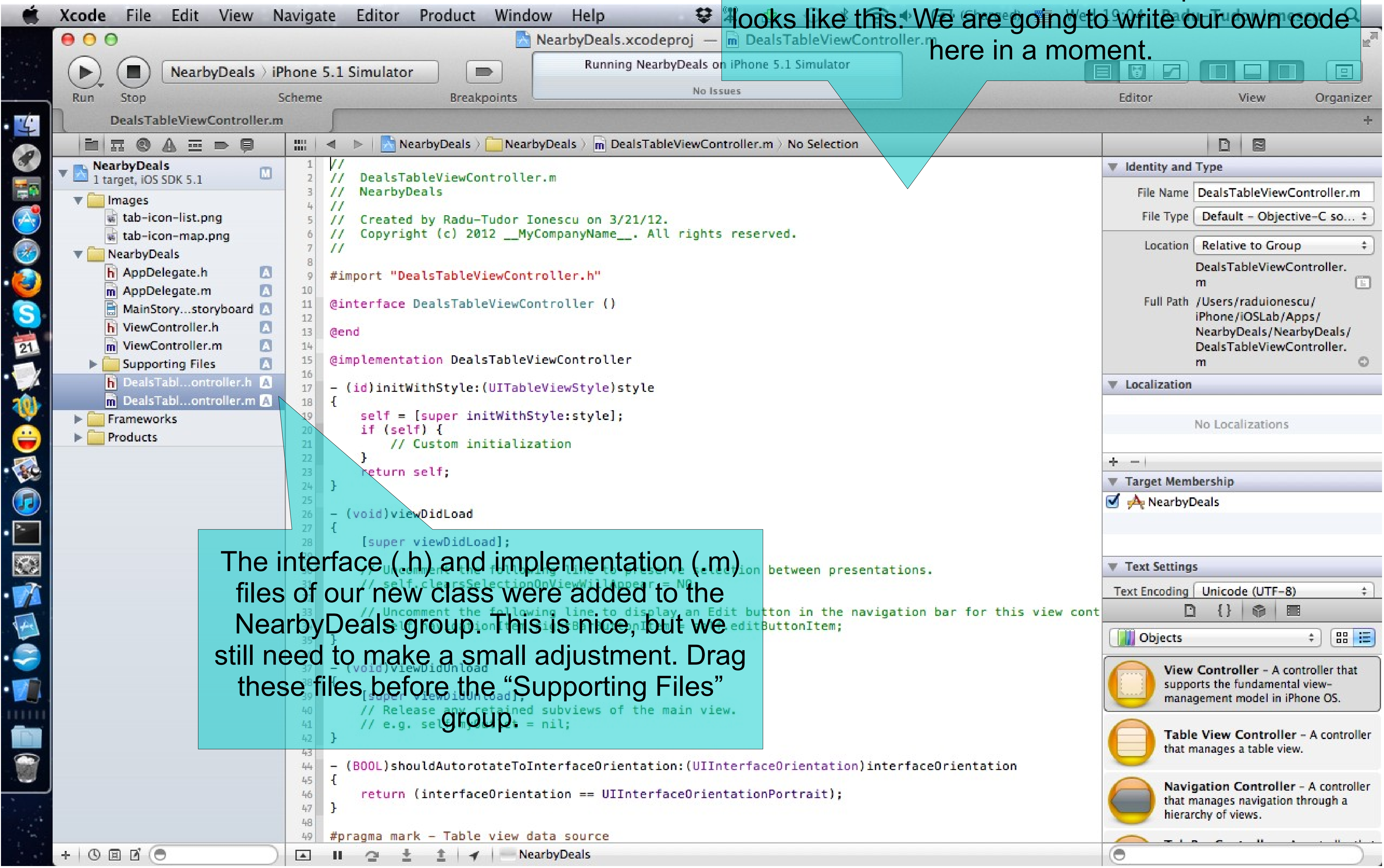
□ Wrap lines

It should also automatically select the NearbyDeals folder inside our Project folder. You should make sure this subfolder is selected.

Xcode knows that we want to add our new class to the NearbyDeals group.

Click here to create the new class that is a subclass of `UITableViewController`.

DealsTableViewController implementation looks like this. We are going to write our own code here in a moment.



```
//
//  DealsTableViewController.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/21/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealsTableViewController.h"

@interface DealsTableViewController ()

@end

@implementation DealsTableViewController

- (id)initWithStyle:(UITableViewStyle)style
{
    self = [super initWithStyle:style];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Uncomment the following line to preserve selection between presentations.
    // self.clearsSelectionOnViewWillAppear = NO;

    // Uncomment the following line to display an Edit button in the navigation bar for this view cont
    // self.navigationItem.rightBarButtonItem = self.editButtonItem;
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

#pragma mark - Table view data source
```

File Name: DealsTableViewController.m
File Type: Default - Objective-C so...
Location: Relative to Group
DealsTableViewController.m
Full Path: /Users/raduionescu/iPhone/iOSLab/Apps/NearbyDeals/NearbyDeals/DealsTableViewController.m
Text Encoding: Unicode (UTF-8)

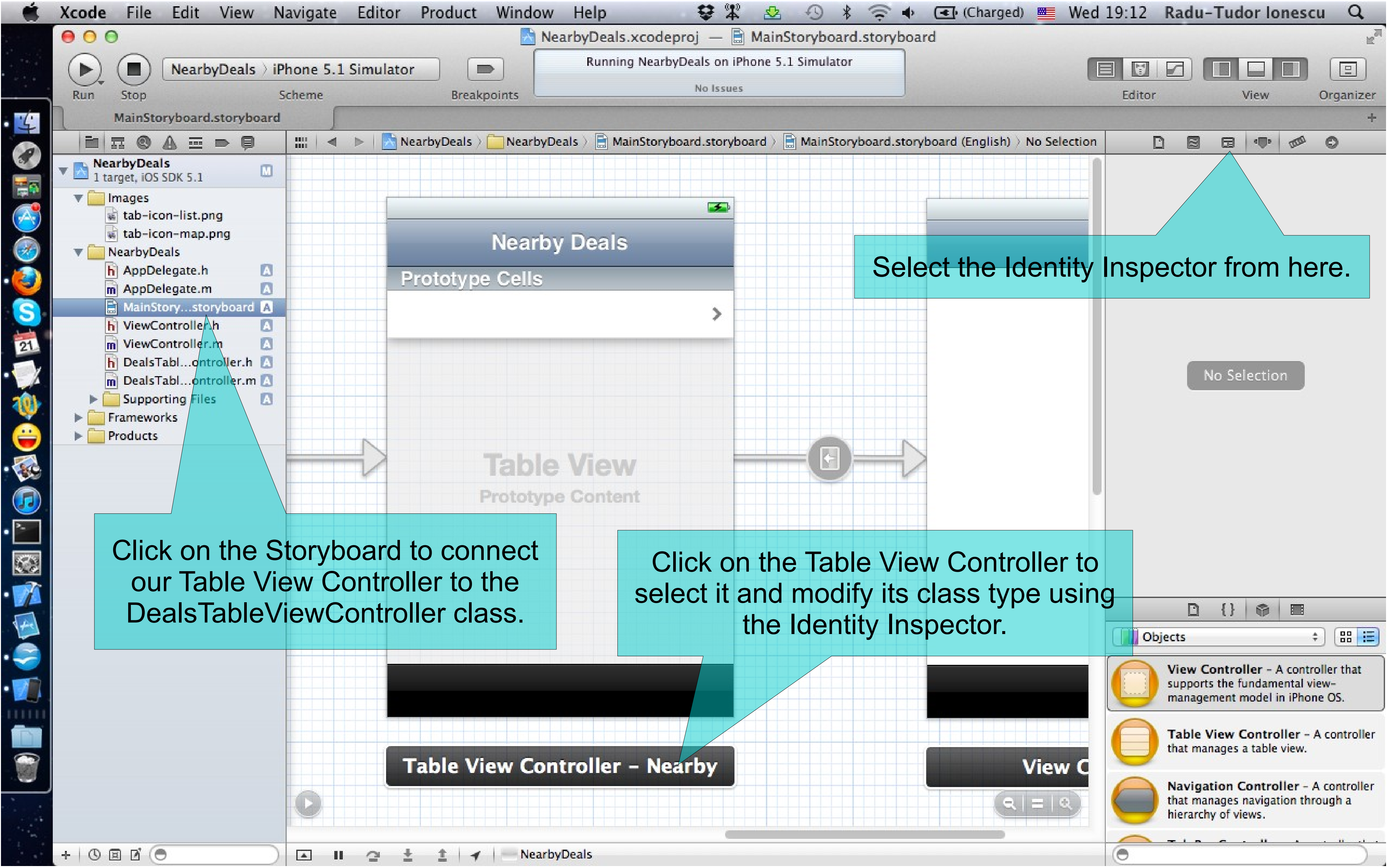
☑ NearbyDeals

View Controller - A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller - A controller that manages a table view.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

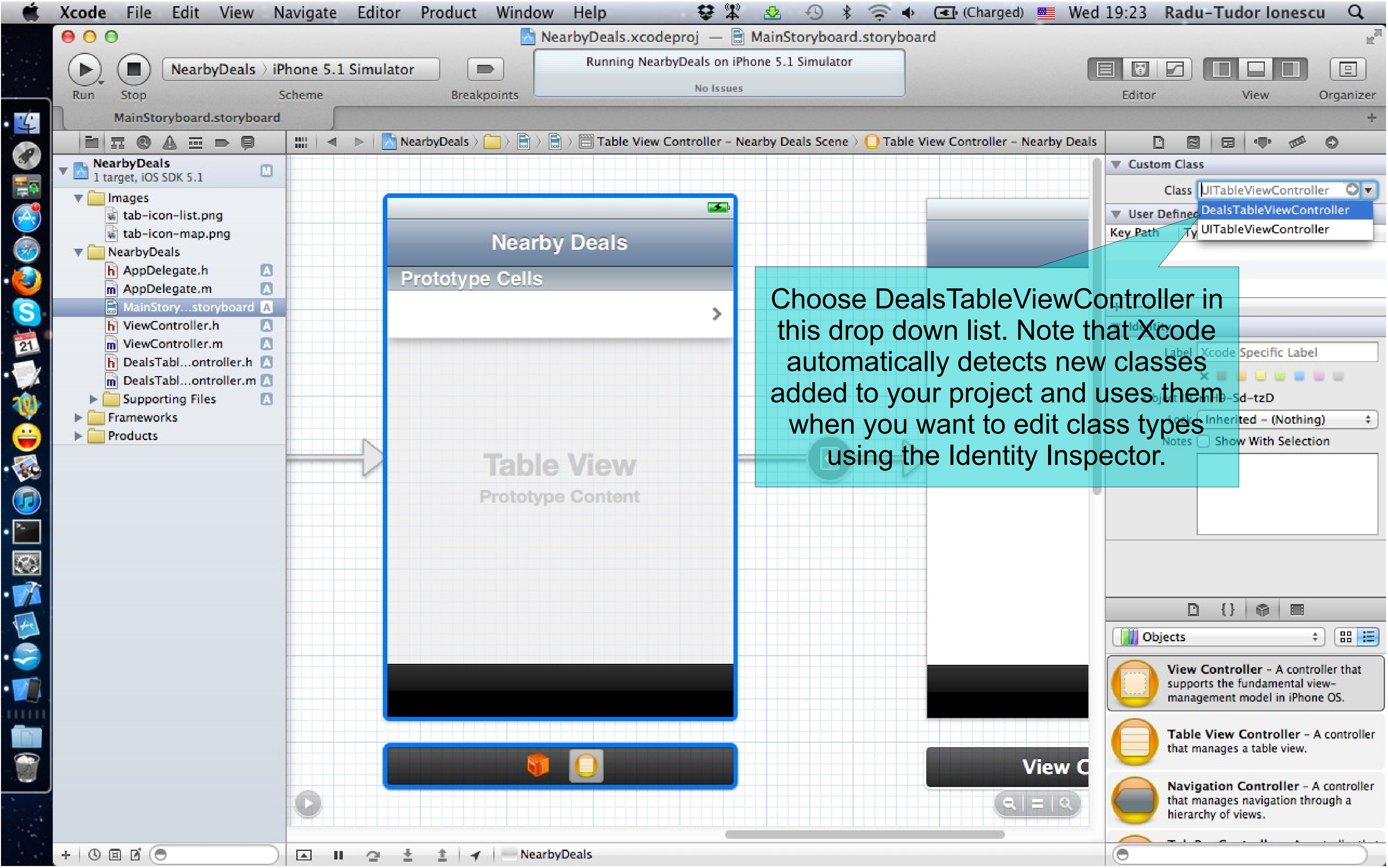
The interface (.h) and implementation (.m) files of our new class were added to the NearbyDeals group. This is nice, but we still need to make a small adjustment. Drag these files before the “Supporting Files” group.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Wed 19:12 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › NearbyDeals › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › No Selection
NearbyDeals
1 target, iOS SDK 5.1
Images
tab-icon-list.png
tab-icon-map.png
NearbyDeals
AppDelegate.h
AppDelegate.m
MainStory...storyboard
ViewController.h
ViewController.m
DealsTabl...ontroller.h
DealsTabl...ontroller.m
Supporting Files
Frameworks
Products
Nearby Deals
Prototype Cells
Table View
Prototype Content
Table View Controller – Nearby
View C
No Selection
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.
NearbyDeals

Select the Identity Inspector from here.

Click on the Storyboard to connect our Table View Controller to the DealsTableViewController class.

Click on the Table View Controller to select it and modify its class type using the Identity Inspector.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Wed 19:23 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › Table View Controller – Nearby Deals Scene › Table View Controller – Nearby Deals
NearbyDeals
1 target, iOS SDK 5.1
Images
tab-icon-list.png
tab-icon-map.png
NearbyDeals
AppDelegate.h
AppDelegate.m
MainStory...storyboard
ViewController.h
ViewController.m
DealsTabl...ontroller.h
DealsTabl...ontroller.m
Supporting Files
Frameworks
Products
Nearby Deals
Prototype Cells
Table View
Prototype Content
View C
Custom Class
Class UITableViewController
DealsTableViewController
UITableViewController
User Defined
Key Path
Label Xcode Specific Label
Inherited – (Nothing)
Show With Selection
Objects
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Navigation Controller – A controller that manages navigation through a hierarchy of views.
NearbyDeals

Choose DealsTableViewController in this drop down list. Note that Xcode automatically detects new classes added to your project and uses them when you want to edit class types using the Identity Inspector.

Task 4

Task: Configure the Table View Controller to show some mock-up data inside its Table View.

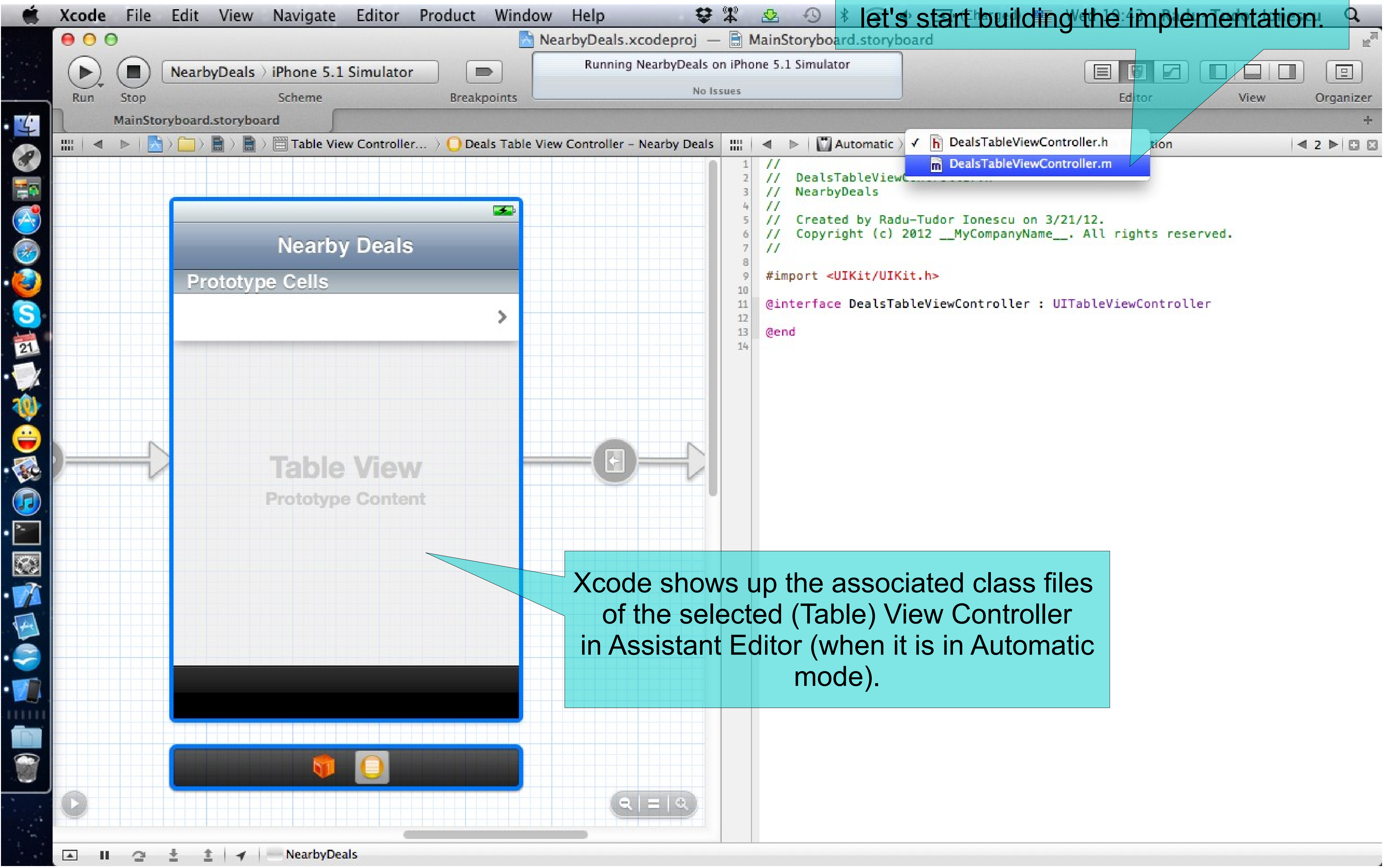
6. Close the Project Navigator and Utilities area to make room for the Assistant Editor.

7. We are going to modify the DealsTableViewController.m file. Open it in Assistant Editor. Note that when you select a View Controller in your Storyboard, Xcode will automatically select its class files in Assistant Editor.

We will add a very simple model to our Table View Controller that will hold the mock-up data that we want to present in our table. We are going to re-implement some of the Table View `dataSource` methods to present the data in our Table View.

The next slides will show you how to do this.

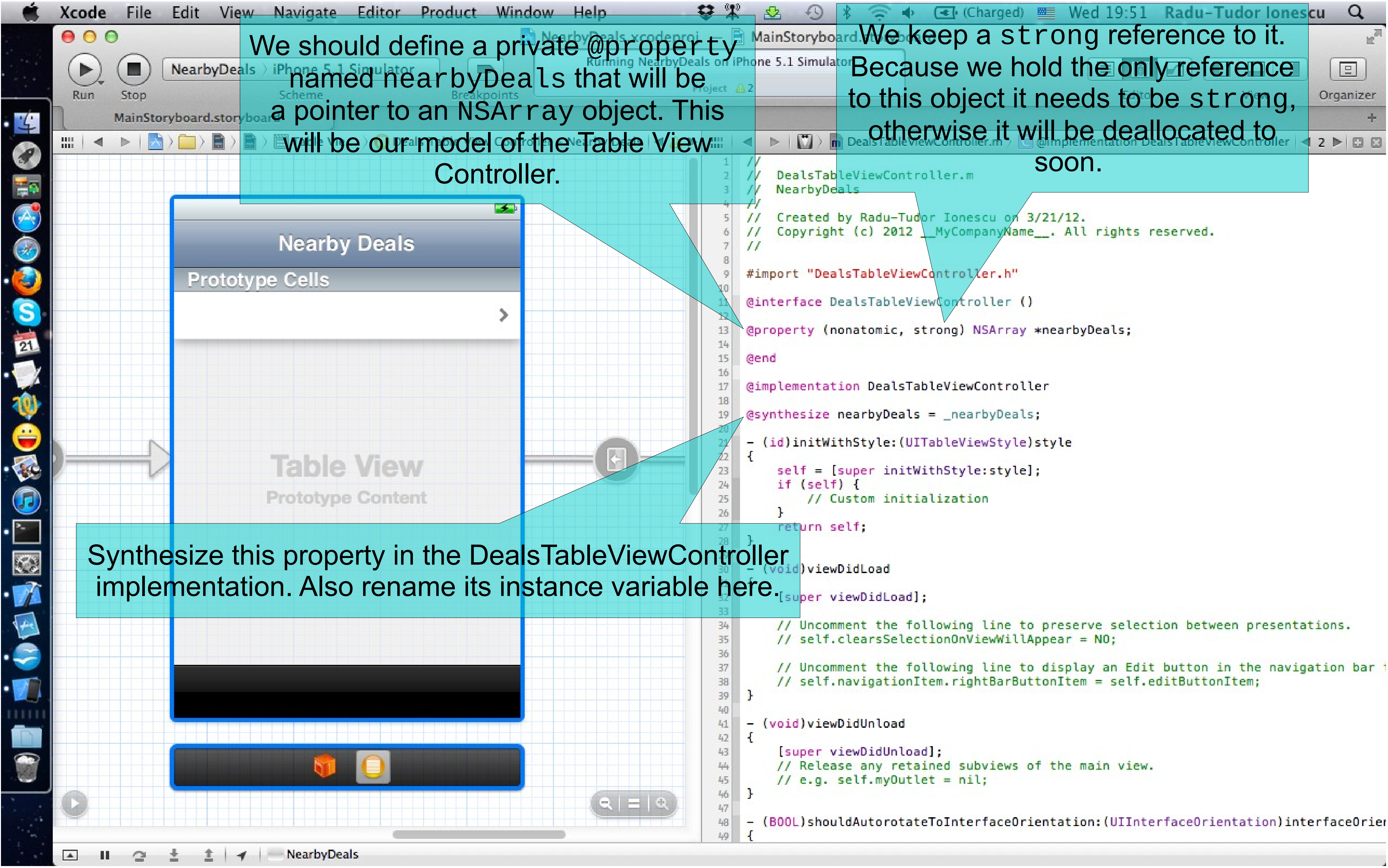
Select DealsTableViewController.m and let's start building the implementation.



Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
Table View Controller... 〉 Deals Table View Controller – Nearby Deals
Automatic 〉
✓ DealsTableViewController.h
DealsTableViewController.m
2
Nearby Deals
Prototype Cells
Table View
Prototype Content
NearbyDeals

```
//
//  DealsTableViewController.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/21/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface DealsTableViewController : UITableViewController

@end
```

Xcode shows up the associated class files of the selected (Table) View Controller in Assistant Editor (when it is in Automatic mode).

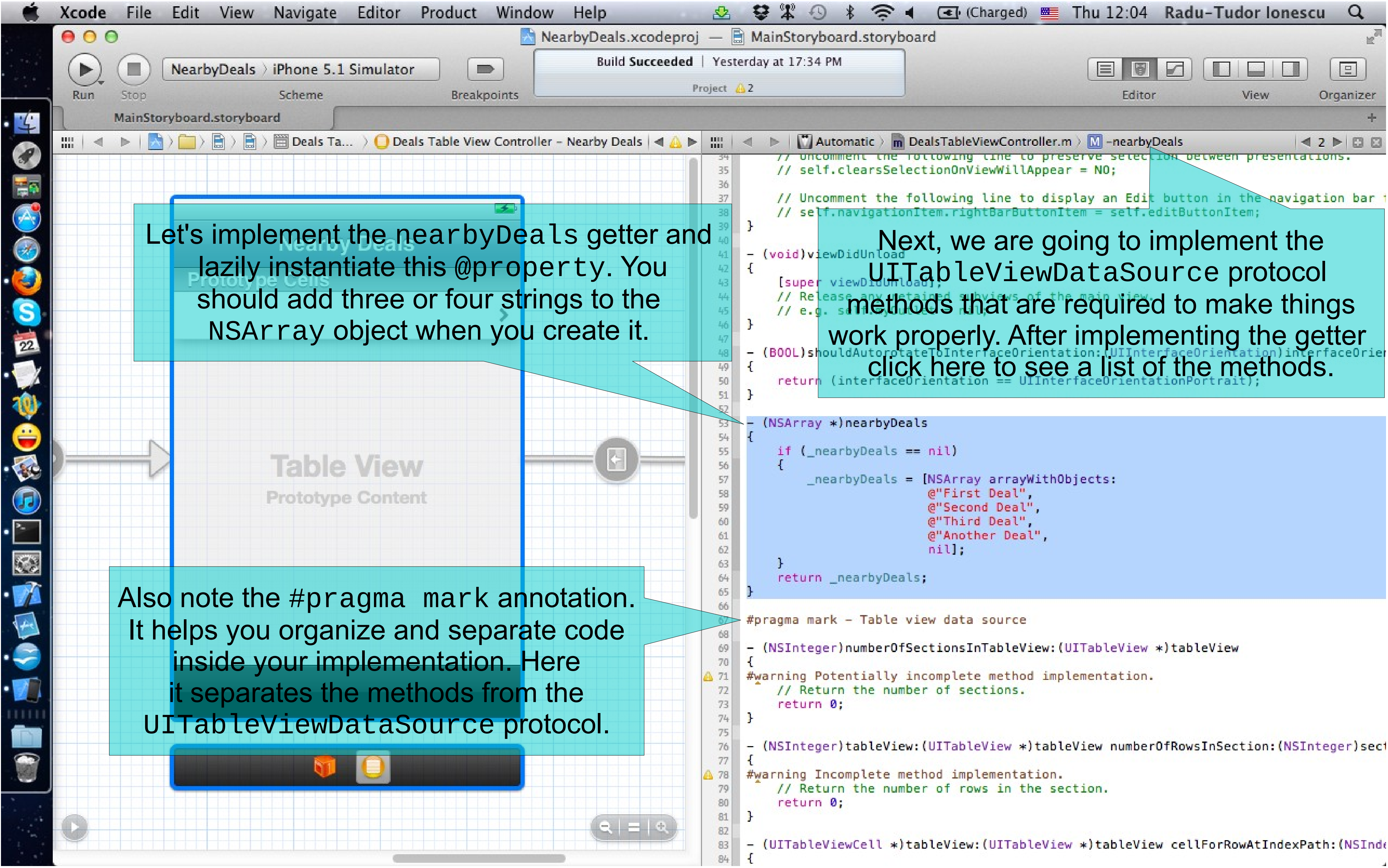


We should define a private @property named nearbyDeals that will be a pointer to an NSArray object. This will be our model of the Table View Controller.

We keep a strong reference to it. Because we hold the only reference to this object it needs to be strong, otherwise it will be deallocated to soon.

Synthesize this property in the DealsTableViewController implementation. Also rename its instance variable here.

```
//
//  DealsTableViewController.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/21/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealsTableViewController.h"

@interface DealsTableViewController ()

@property (nonatomic, strong) NSArray *nearbyDeals;

@end

@implementation DealsTableViewController

@synthesize nearbyDeals = _nearbyDeals;

- (id)initWithStyle:(UITableViewStyle)style
{
    self = [super initWithStyle:style];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad

    [super viewDidLoad];

    // Uncomment the following line to preserve selection between presentations.
    // self.clearsSelectionOnViewWillAppear = NO;

    // Uncomment the following line to display an Edit button in the navigation bar f
    // self.navigationItem.rightBarButtonItem = self.editButtonItem;
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrier
{
```

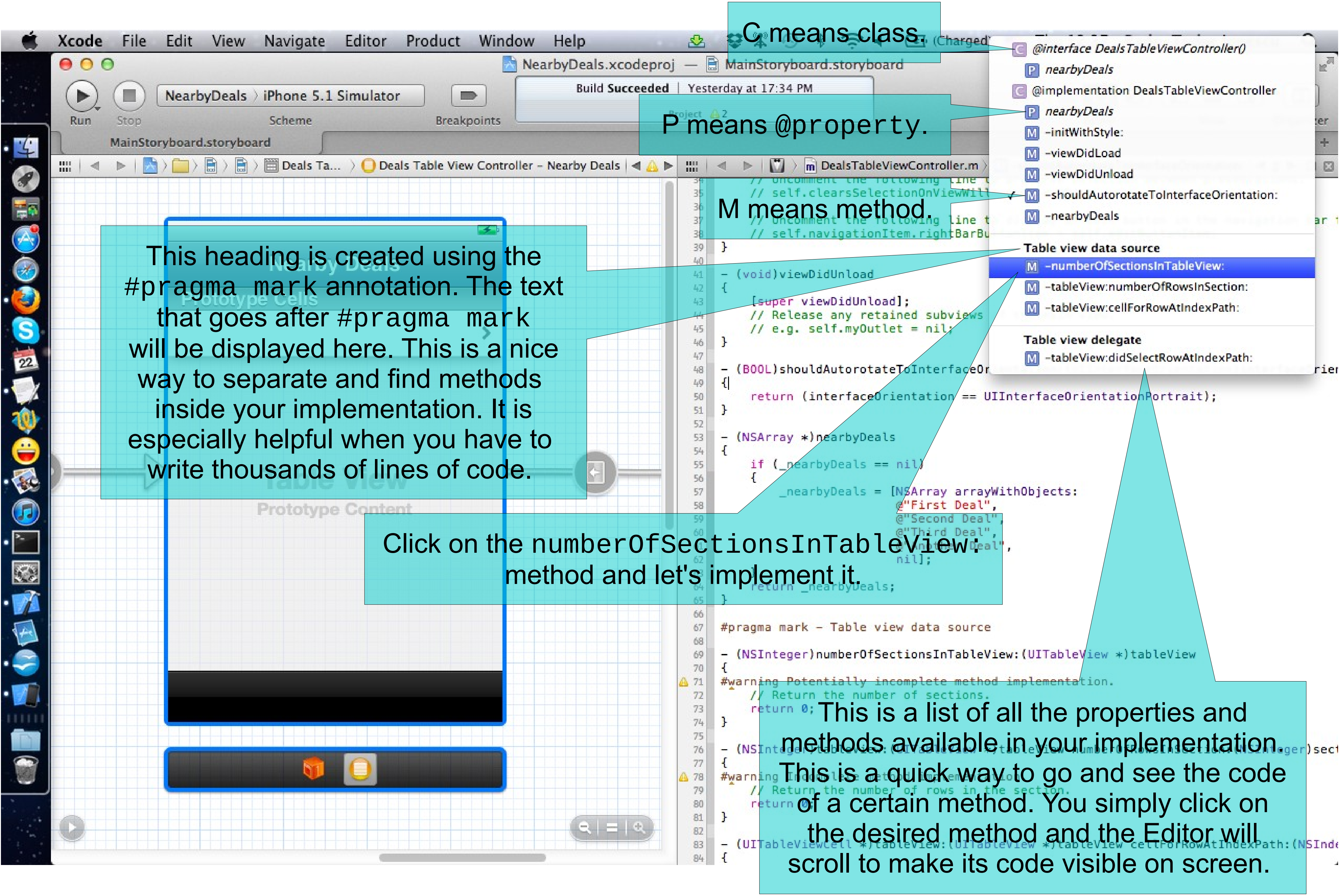

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 12:04 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 17:34 PM
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
Deals Ta... 〉 Deals Table View Controller – Nearby Deals
Automatic 〉 DealsTableViewController.m 〉 -nearbyDeals
2
Nearby Deals
Prototype Cells
Table View
Prototype Content

Let's implement the nearbyDeals getter and lazily instantiate this @property. You should add three or four strings to the NSArray object when you create it.

Next, we are going to implement the UITableViewDataSource protocol methods that are required to make things work properly. After implementing the getter click here to see a list of the methods.

Also note the #pragma mark annotation. It helps you organize and separate code inside your implementation. Here it separates the methods from the UITableViewDataSource protocol.

```
    // Uncomment the following line to preserve selection between presentations.
    // self.clearsSelectionOnViewWillAppear = NO;

    // Uncomment the following line to display an Edit button in the navigation bar f
    // self.navigationItem.rightBarButtonItem = self.editButtonItem;
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrie
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (NSArray *)nearbyDeals
{
    if (_nearbyDeals == nil)
    {
        _nearbyDeals = [NSArray arrayWithObjects:
                        @"First Deal",
                        @"Second Deal",
                        @"Third Deal",
                        @"Another Deal",
                        nil];
    }
    return _nearbyDeals;
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
#warning Potentially incomplete method implementation.
    // Return the number of sections.
    return 0;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)sect
{
#warning Incomplete method implementation.
    // Return the number of rows in the section.
    return 0;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSInde
{
```



Xcode File Edit View Navigate Editor Product Window Help
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 17:34 PM
MainStoryboard.storyboard
Deals Ta... › Deals Table View Controller – Nearby Deals
DealsTableViewController.m
Nearby Deals
Prototype Cells
Table View
Prototype Content
C @interface DealsTableViewController()
P nearbyDeals
C @implementation DealsTableViewController
P nearbyDeals
M -initWithStyle:
M -viewDidLoad
M -viewDidUnload
M -shouldAutorotateToInterfaceOrientation:
M -nearbyDeals
Table view data source
M -numberOfSectionsInTableView:
M -tableView:numberOfRowsInSection:
M -tableView:cellForRowAtIndexPath:
Table view delegate
M -tableView:didSelectRowAtIndexPath:

```
    // Uncomment the following line
    // self.clearsSelectionOnViewWill
    // Uncomment the following line t
    // self.navigationItem.rightBarBu
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOr
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (NSArray *)nearbyDeals
{
    if (_nearbyDeals == nil)
    {
        _nearbyDeals = [NSArray arrayWithObjects:
                        @"First Deal",
                        @"Second Deal",
                        @"Third Deal",
                        @"Another Deal",
                        nil];
    }
    return _nearbyDeals;
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
#warning Potentially incomplete method implementation.
    // Return the number of sections.
    return 0;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)sect
{
#warning Incomplete method implementation.
    // Return the number of rows in the section.
    return 0;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSInde
{
```

C means class.

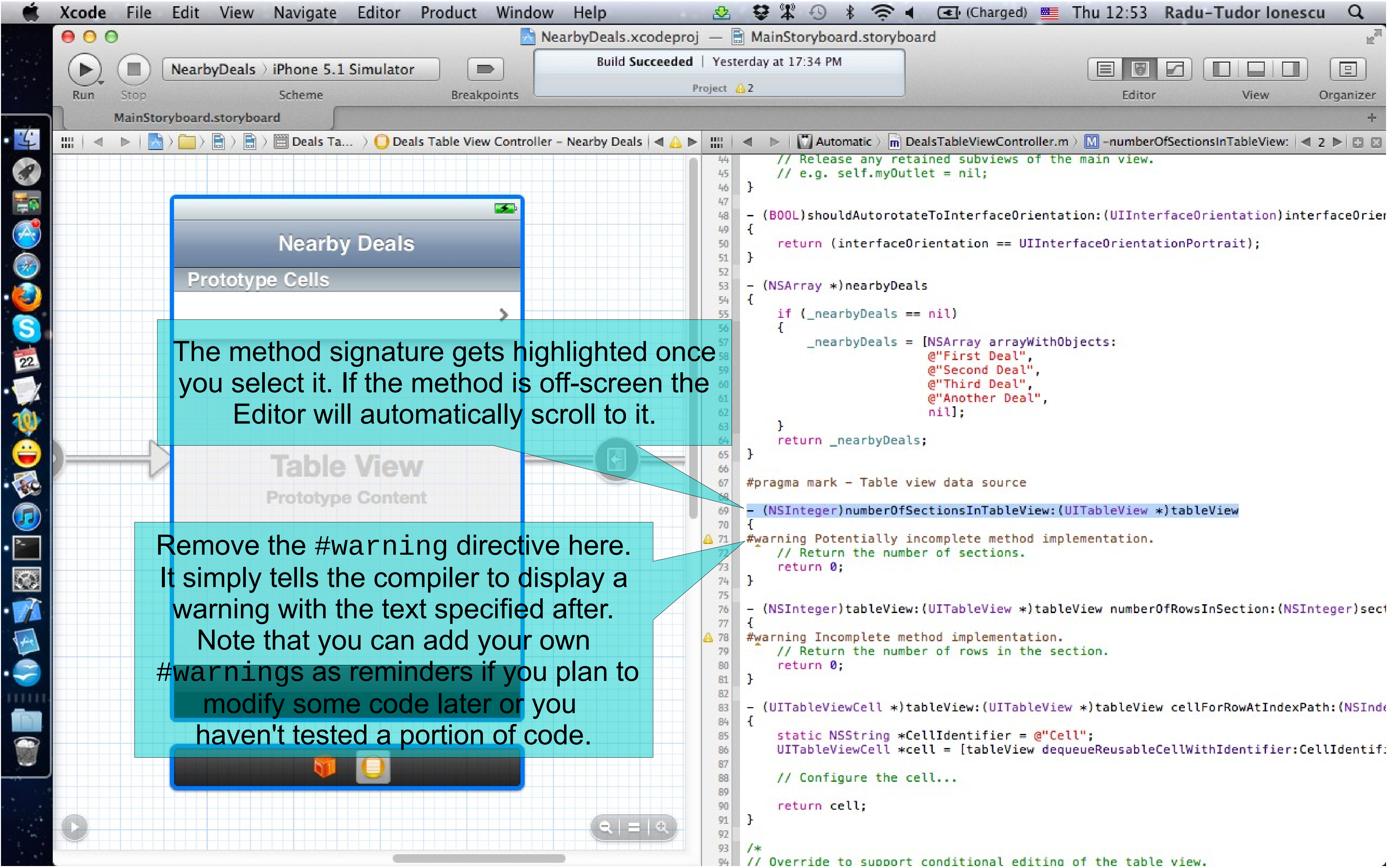
P means @property.

M means method.

This heading is created using the #pragma mark annotation. The text that goes after #pragma mark will be displayed here. This is a nice way to separate and find methods inside your implementation. It is especially helpful when you have to write thousands of lines of code.

Click on the numberOfSectionsInTableView: method and let's implement it.

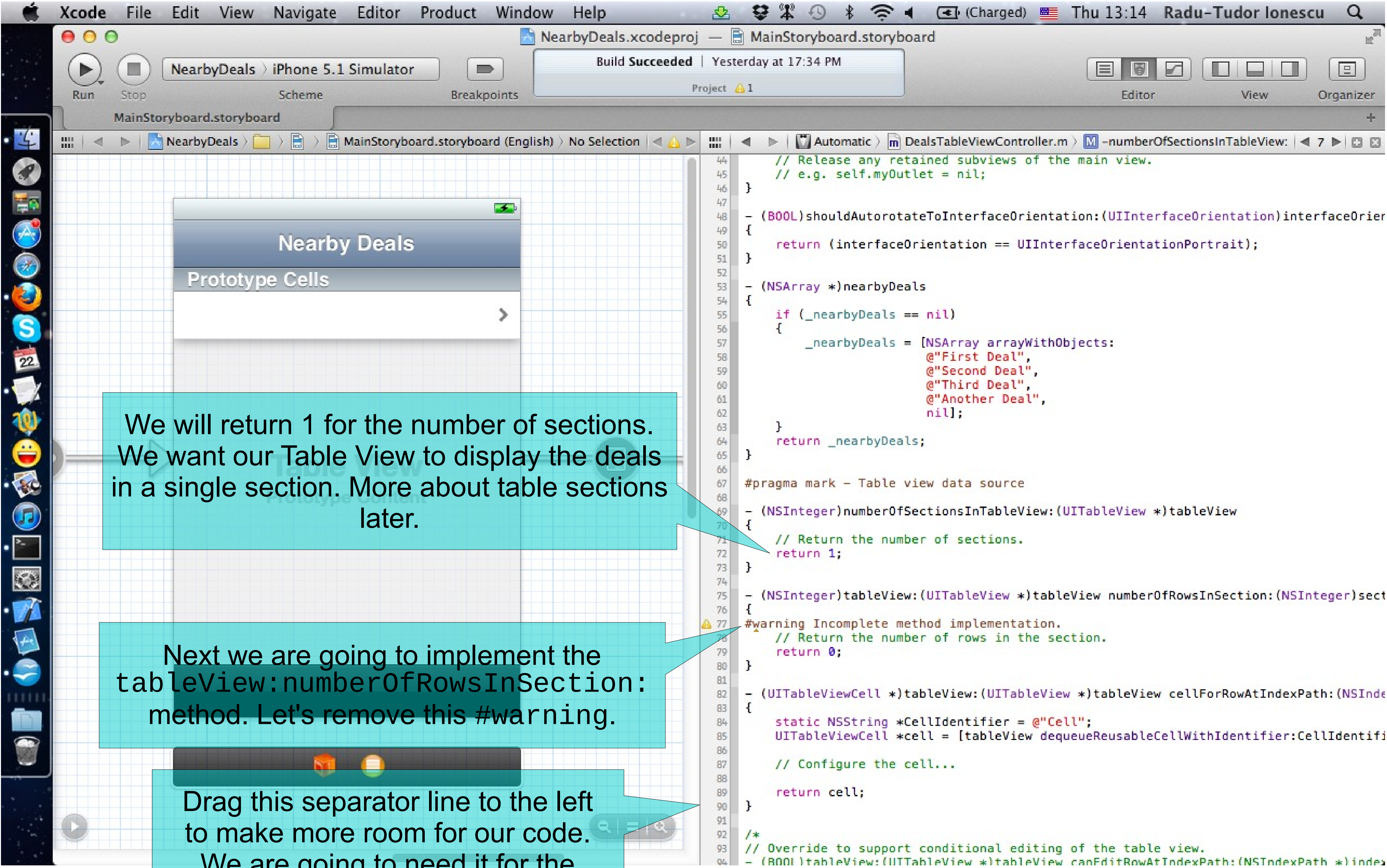
This is a list of all the properties and methods available in your implementation. This is a quick way to go and see the code of a certain method. You simply click on the desired method and the Editor will scroll to make its code visible on screen.


Xcode File Edit View Navigate Editor Product Window Help
(Charged)
Thu 12:53 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 17:34 PM
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
Deals Ta...
Deals Table View Controller – Nearby Deals
Automatic
DealsTableViewController.m
-numberOfSectionsInTableView:
2
Nearby Deals
Prototype Cells
Table View
Prototype Content

```
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrie
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (NSArray *)nearbyDeals
{
    if (_nearbyDeals == nil)
    {
        _nearbyDeals = [NSArray arrayWithObjects:
                        @"First Deal",
                        @"Second Deal",
                        @"Third Deal",
                        @"Another Deal",
                        nil];
    }
    return _nearbyDeals;
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
#warning Potentially incomplete method implementation.
    // Return the number of sections.
    return 0;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)sect
{
#warning Incomplete method implementation.
    // Return the number of rows in the section.
    return 0;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSInde
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentif

    // Configure the cell...

    return cell;
}

/*
// Override to support conditional editing of the table view.
```

The method signature gets highlighted once you select it. If the method is off-screen the Editor will automatically scroll to it.

Remove the `#warning` directive here. It simply tells the compiler to display a warning with the text specified after. Note that you can add your own `#warnings` as reminders if you plan to modify some code later or you haven't tested a portion of code.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 13:14 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 17:34 PM
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
NearbyDeals › MainStoryboard.storyboard (English) › No Selection
Automatic › DealsTableViewController.m › -numberOfSectionsInTableView:
7
Nearby Deals
Prototype Cells
Table View
Prototype Content

```
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrien
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (NSArray *)nearbyDeals
{
    if (_nearbyDeals == nil)
    {
        _nearbyDeals = [NSArray arrayWithObjects:
                        @"First Deal",
                        @"Second Deal",
                        @"Third Deal",
                        @"Another Deal",
                        nil];
    }
    return _nearbyDeals;
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    // Return the number of sections.
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)sect
{
#warning Incomplete method implementation.
    // Return the number of rows in the section.
    return 0;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSInde
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifi

    // Configure the cell...

    return cell;
}

/*
// Override to support conditional editing of the table view.
- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)inde
```

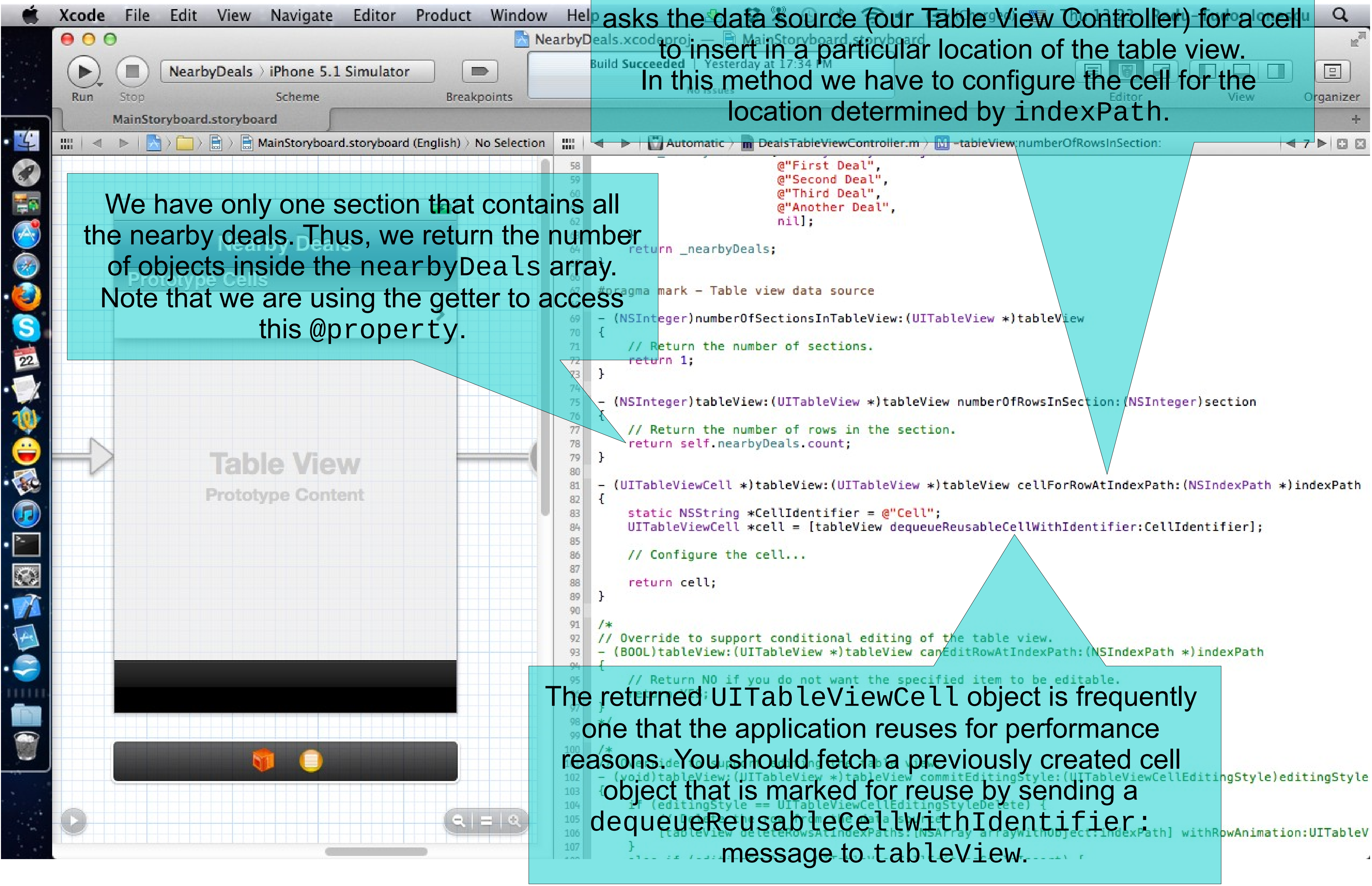
We will return 1 for the number of sections. We want our Table View to display the deals in a single section. More about table sections later.

Next we are going to implement the `tableView:numberOfRowsInSection:` method. Let's remove this `#warning`.

Drag this separator line to the left to make more room for our code. We are going to need it for the next two methods.

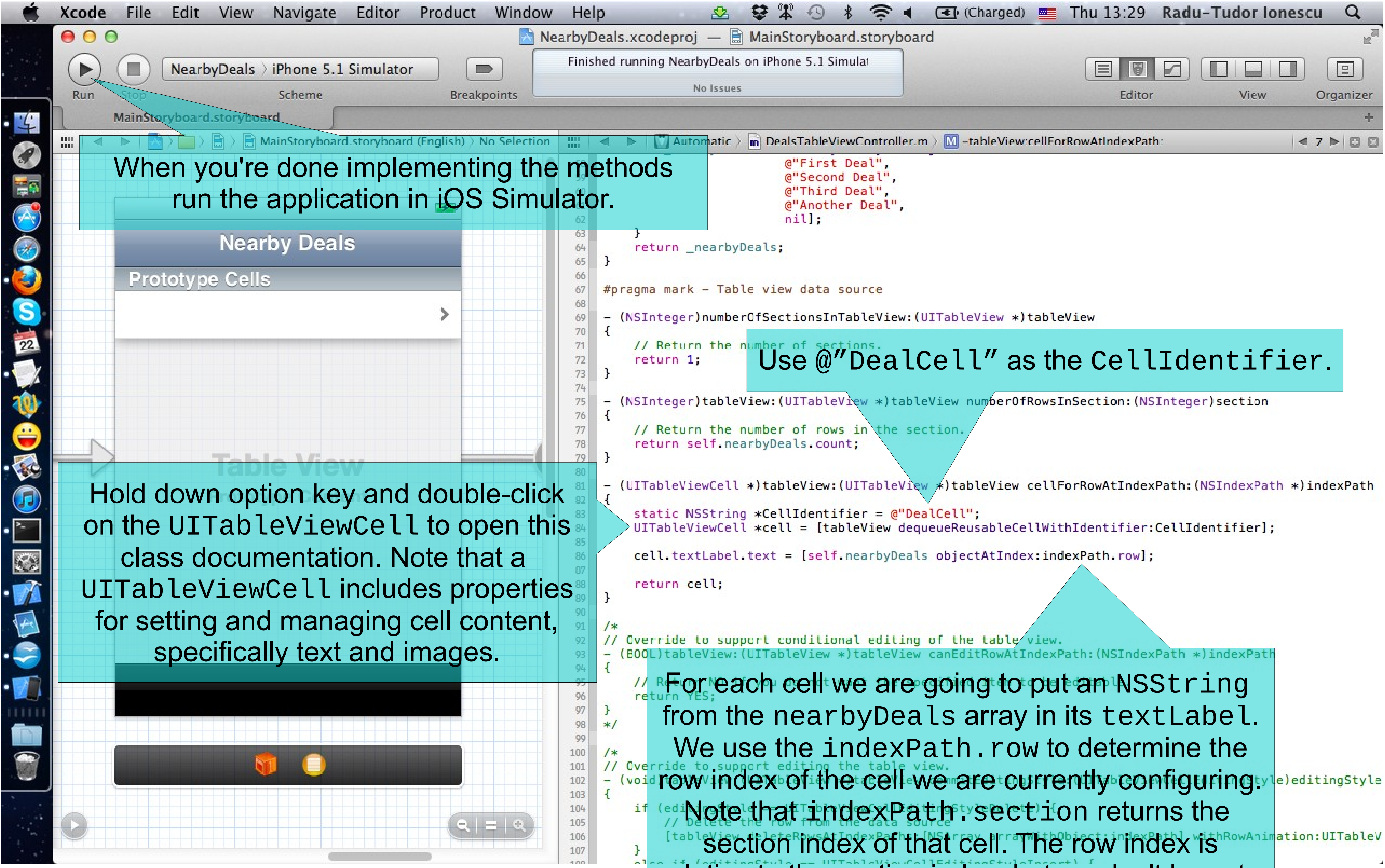
The `tableView:cellForRowAtIndexPath:` method asks the data source (our Table View Controller) for a cell to insert in a particular location of the table view. In this method we have to configure the cell for the location determined by `indexPath`.

We have only one section that contains all the nearby deals. Thus, we return the number of objects inside the `nearbyDeals` array. Note that we are using the getter to access this `@property`.


Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 17:34 PM
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
MainStoryboard.storyboard (English) › No Selection
Automatic › DealsTableViewController.m › -tableView:numberOfRowsInSection:
Nearby Deals
Prototype Cells
Table View
Prototype Content

```
                          @"First Deal",
                          @"Second Deal",
                          @"Third Deal",
                          @"Another Deal",
                          nil];
    }
    return _nearbyDeals;
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    // Return the number of sections.
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    // Return the number of rows in the section.
    return self.nearbyDeals.count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    // Configure the cell...

    return cell;
}

/*
// Override to support conditional editing of the table view.
- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath
{
    // Return NO if you do not want the specified item to be editable.
    return YES;
}
*/

/*
// Override to support editing the table view.
- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle
{
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        // Delete the row from the data source
        [tableView deleteRowsAtIndexPaths:[NSArray arrayWithObject:indexPath] withRowAnimation:UITableV
    }
```

The returned `UITableViewCell` object is frequently one that the application reuses for performance reasons. You should fetch a previously created cell object that is marked for reuse by sending a `dequeueReusableCellWithIdentifier:` message to `tableView`.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 13:29 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Finished running NearbyDeals on iPhone 5.1 Simulat
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
MainStoryboard.storyboard (English) › No Selection
Automatic › DealsTableViewController.m › -tableView:cellForRowAtIndexPath:
Nearby Deals
Prototype Cells
Table View

```
                        @"First Deal",
                        @"Second Deal",
                        @"Third Deal",
                        @"Another Deal",
                        nil];
    }
    return _nearbyDeals;
}

#pragma mark - Table view data source

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    // Return the number of sections.
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    // Return the number of rows in the section.
    return self.nearbyDeals.count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"DealCell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];

    cell.textLabel.text = [self.nearbyDeals objectAtIndex:indexPath.row];

    return cell;
}

/*
// Override to support conditional editing of the table view.
- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath
{
```

When you're done implementing the methods run the application in iOS Simulator.

Use @"DealCell" as the `CellIdentifier`.

Hold down option key and double-click on the `UITableViewCell` to open this class documentation. Note that a `UITableViewCell` includes properties for setting and managing cell content, specifically text and images.

For each cell we are going to put an `NSString` from the `nearbyDeals` array in its `textLabel`. We use the `indexPath.row` to determine the row index of the cell we are currently configuring. Note that `indexPath.section` returns the section index of that cell. The row index is relative to the section index, but we don't have to worry about this since we only have one section.

Assignment 1

Assignment: Add enough mock-up deals when creating the `nearbyDeals` array so that the Table View will have to use scrolling to display all its cells.

Hint: You should add more than 9 objects to the `NSArray` that is the Model of our Table View Controller.

Congratulations!