Developing Applications for iOS

Lecture 6:
Table Views

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

Content

- UITableView
- Creating Table View MVCs
- UITableViewDataSource
- UITableViewDelegate

UITableView

Very important class for displaying data in a table

- One-dimensional table.
- It’s a subclass of `UIScrollView`.
- Table can be a static or dynamic list of items.
- Lots and lots of customization via a `dataSource` protocol and a `delegate` protocol.
- Very efficient even with very large sets of data.

UITableView

Displaying multi-dimensional tables

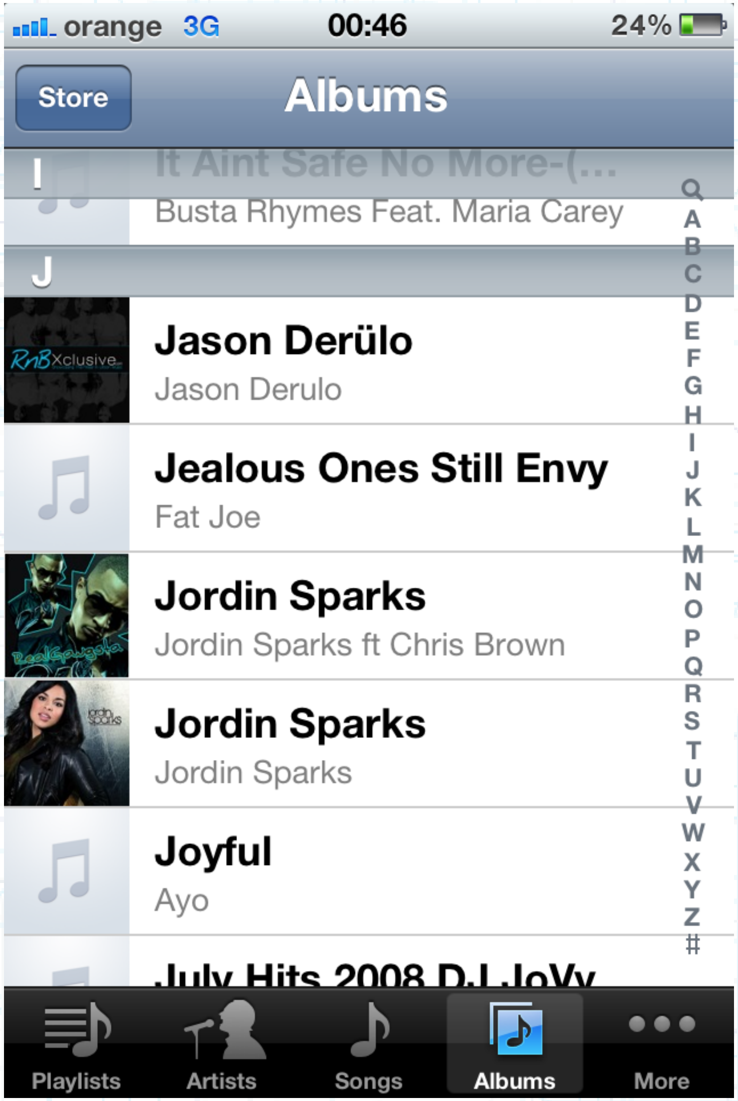
- Usually done via a `UINavigationController` containing multiple MVC’s where View is `UITableView`.
- Or, via the new `UICollectionView` in iOS 6.0. Collection views provide the same general function as table views, except that a collection view is able to support more layouts.
- Collection views support customizable layouts that can be used to implement multi-column grids, circular layouts, and many more.

Kinds of `UITableViews`

- Plain or Grouped.
- Static or Dynamic.
- Divided into sections or not.
- Different formats for each row in the table (including completely customized).

UITableView

UITableViewStylePlain



orange 3G 00:46 24%

Store

Albums

I

It Aint Safe No More-(...

Busta Rhymes Feat. Maria Carey

J

Jason Derülo

Jason Derulo

Jealous Ones Still Envy

Fat Joe

Jordin Sparks

Jordin Sparks ft Chris Brown

Jordin Sparks

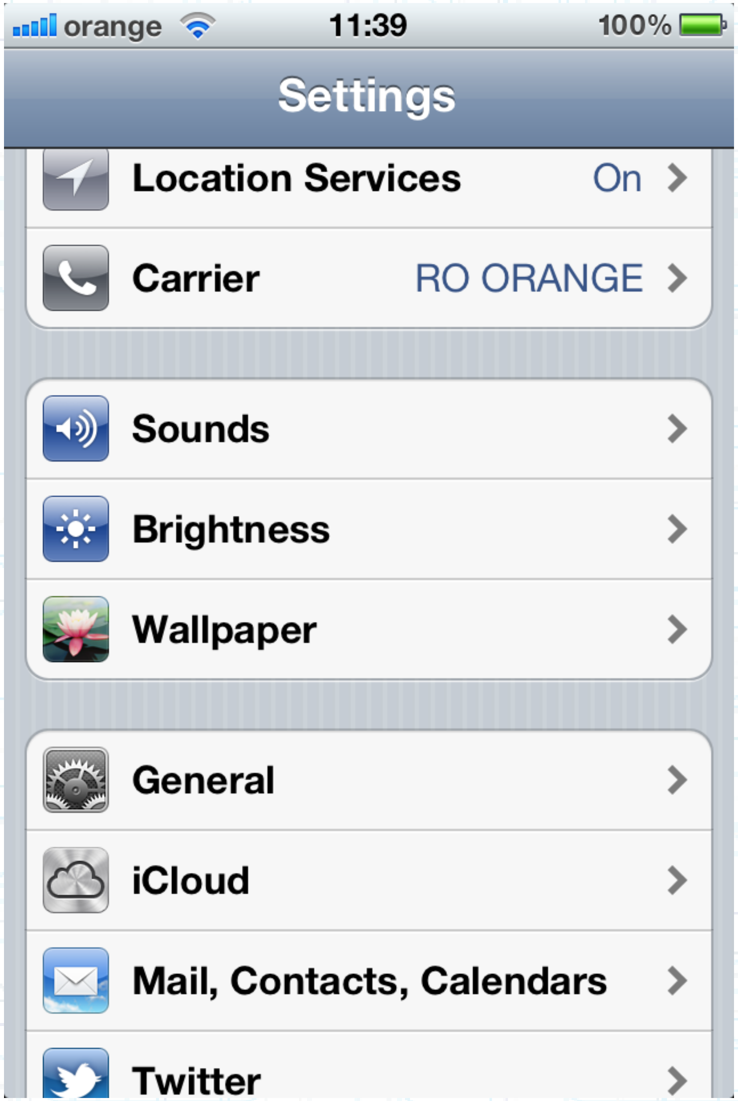
Jordin Sparks

Joyful

Ayo

July Hits 2008 DJ JoVy

UITableViewStyleGrouped



orange 11:39 100%

Settings

Location Services On

Carrier RO ORANGE

Sounds

Brightness

Wallpaper

General

iCloud

Mail, Contacts, Calendars

Twitter

UITableView

Plain Style

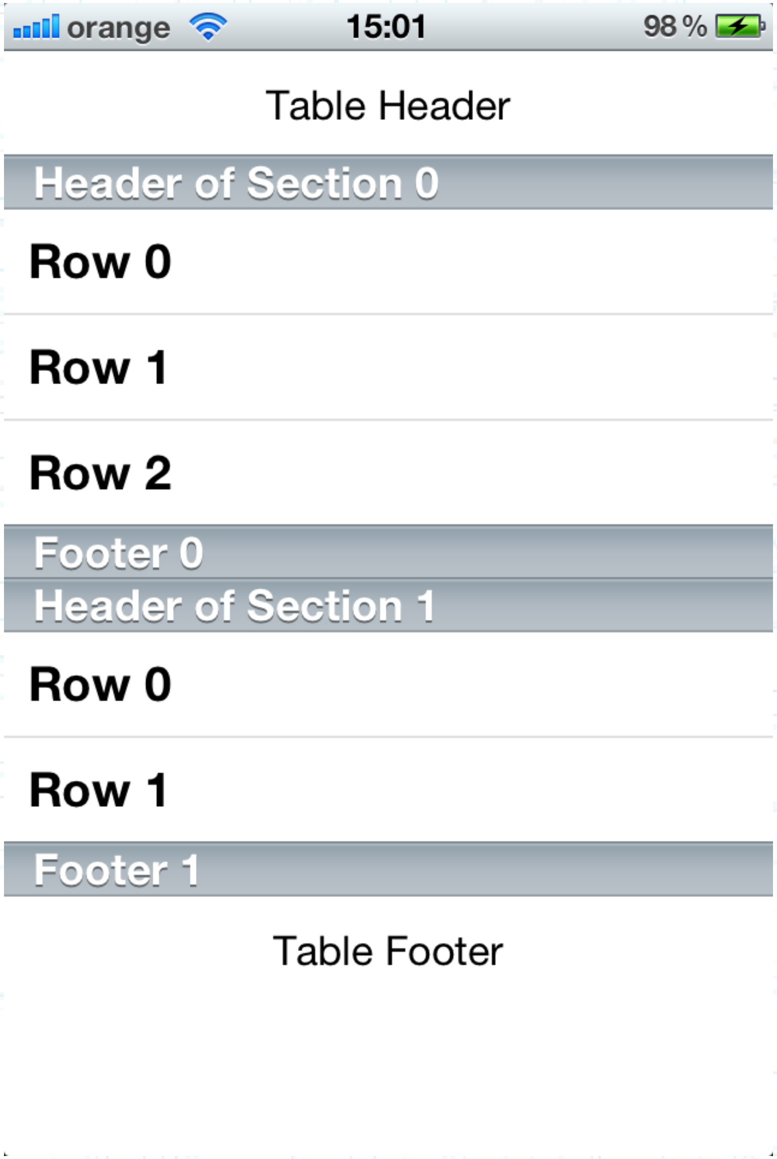


Table Header

Header of Section 0

Row 0

Row 1

Row 2

Footer 0

Header of Section 1

Row 0

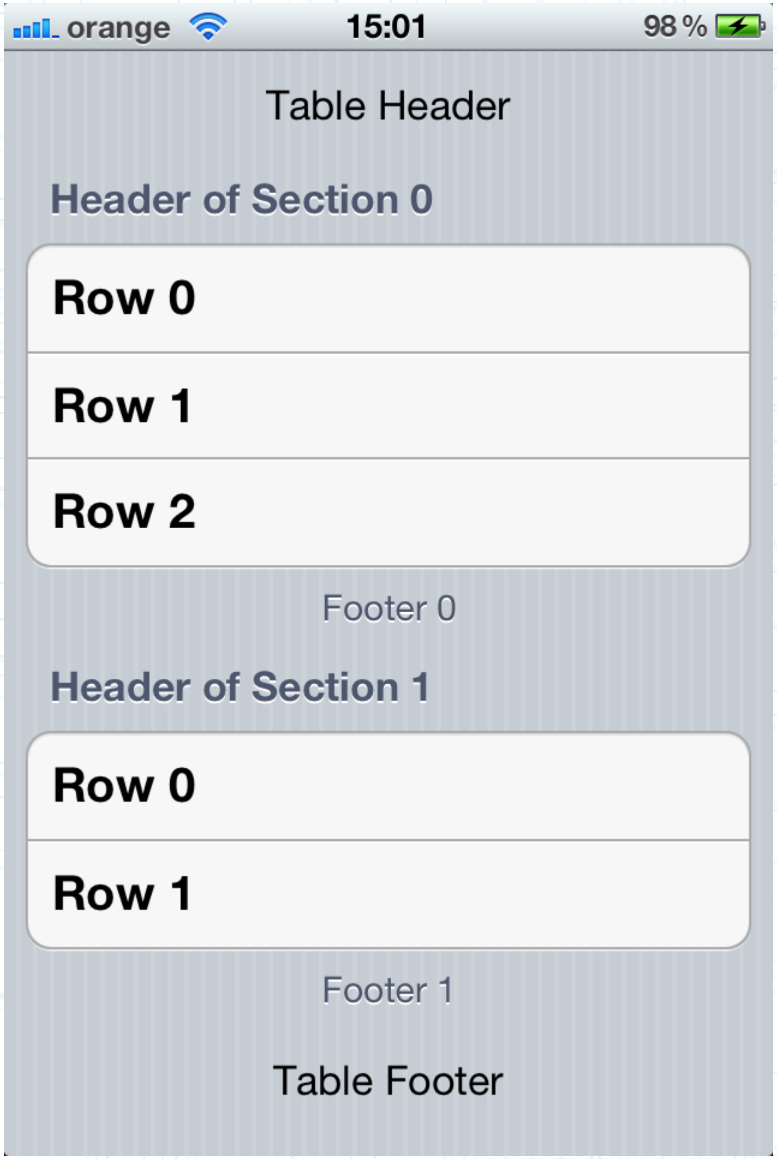
Row 1

Footer 1

Table Footer

UITableView

Grouped Style



orange 15:01 98 %

Table Header

Header of Section 0

Row 0

Row 1

Row 2

Footer 0

Header of Section 1

Row 0

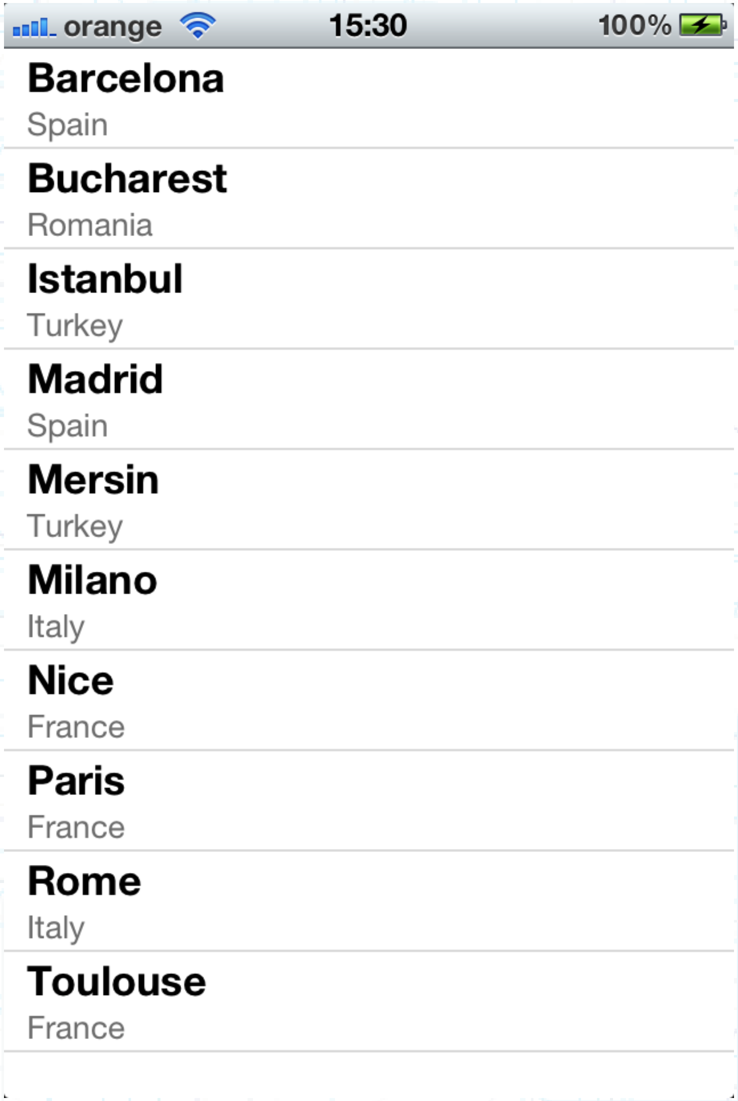
Row 1

Footer 1

Table Footer

UITableView

No Sections



Barcelona
Spain

Bucharest
Romania

Istanbul
Turkey

Madrid
Spain

Mersin
Turkey

Milano
Italy

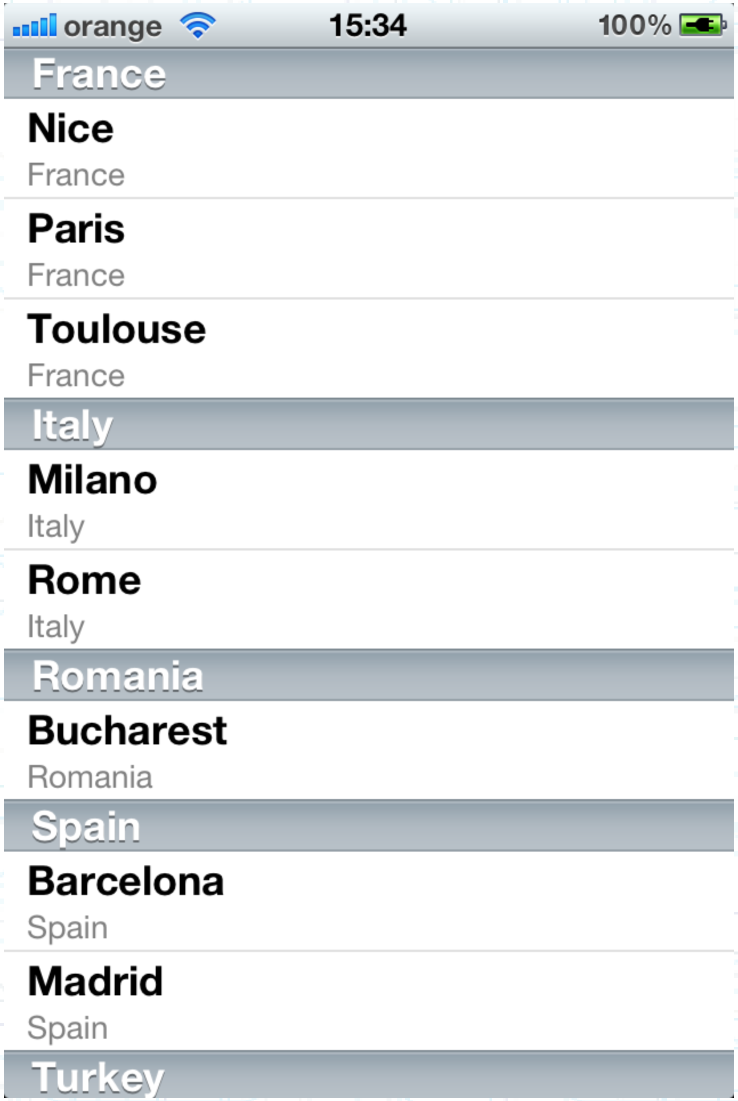
Nice
France

Paris
France

Rome
Italy

Toulouse
France

Sections



France

Nice
France

Paris
France

Toulouse
France

Italy

Milano
Italy

Rome
Italy

Romania

Bucharest
Romania

Spain

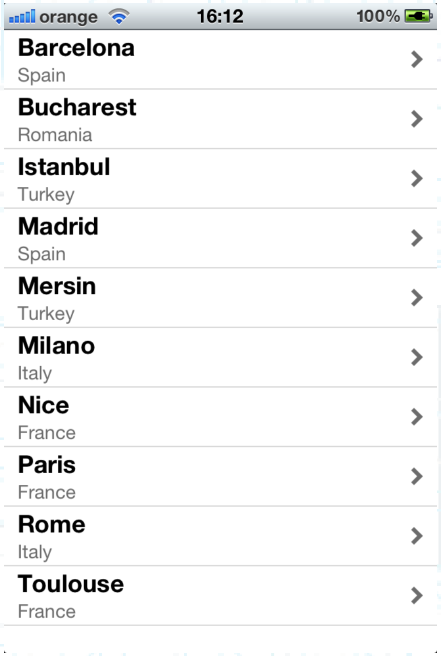
Barcelona
Spain

Madrid
Spain

Turkey

Cell Type

Subtitle
`UITableViewCellStyleSubtitle`



orange 16:12 100%

Barcelona
Spain

Bucharest
Romania

Istanbul
Turkey

Madrid
Spain

Mersin
Turkey

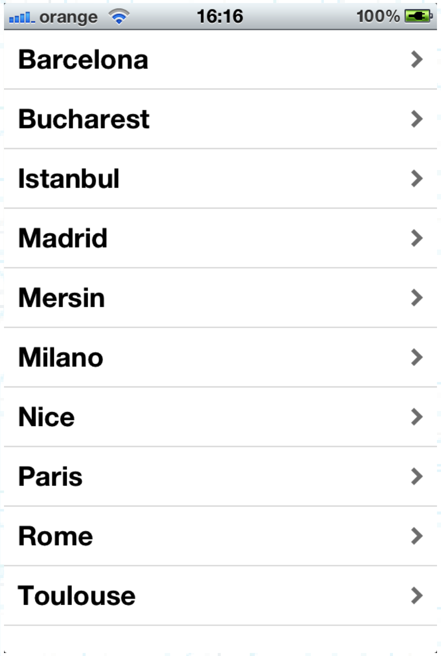
Milano
Italy

Nice
France

Paris
France

Rome
Italy

Toulouse
France



orange 16:16 100%

Barcelona

Bucharest

Istanbul

Madrid

Mersin

Milano

Nice

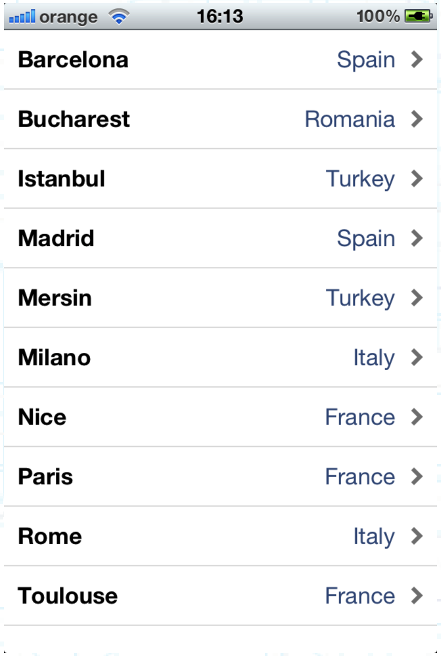
Paris

Rome

Toulouse

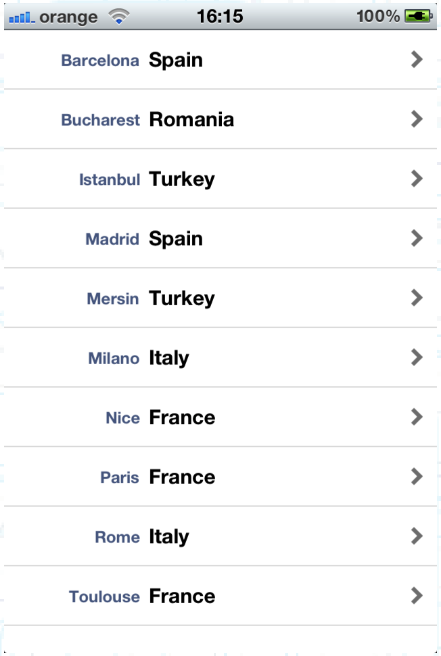
Basic
`UITableViewCellStyleDefault`

Right Detail
`UITableViewCellStyleValue1`



orange 16:13 100%

Barcelona	Spain
Bucharest	Romania
Istanbul	Turkey
Madrid	Spain
Mersin	Turkey
Milano	Italy
Nice	France
Paris	France
Rome	Italy
Toulouse	France



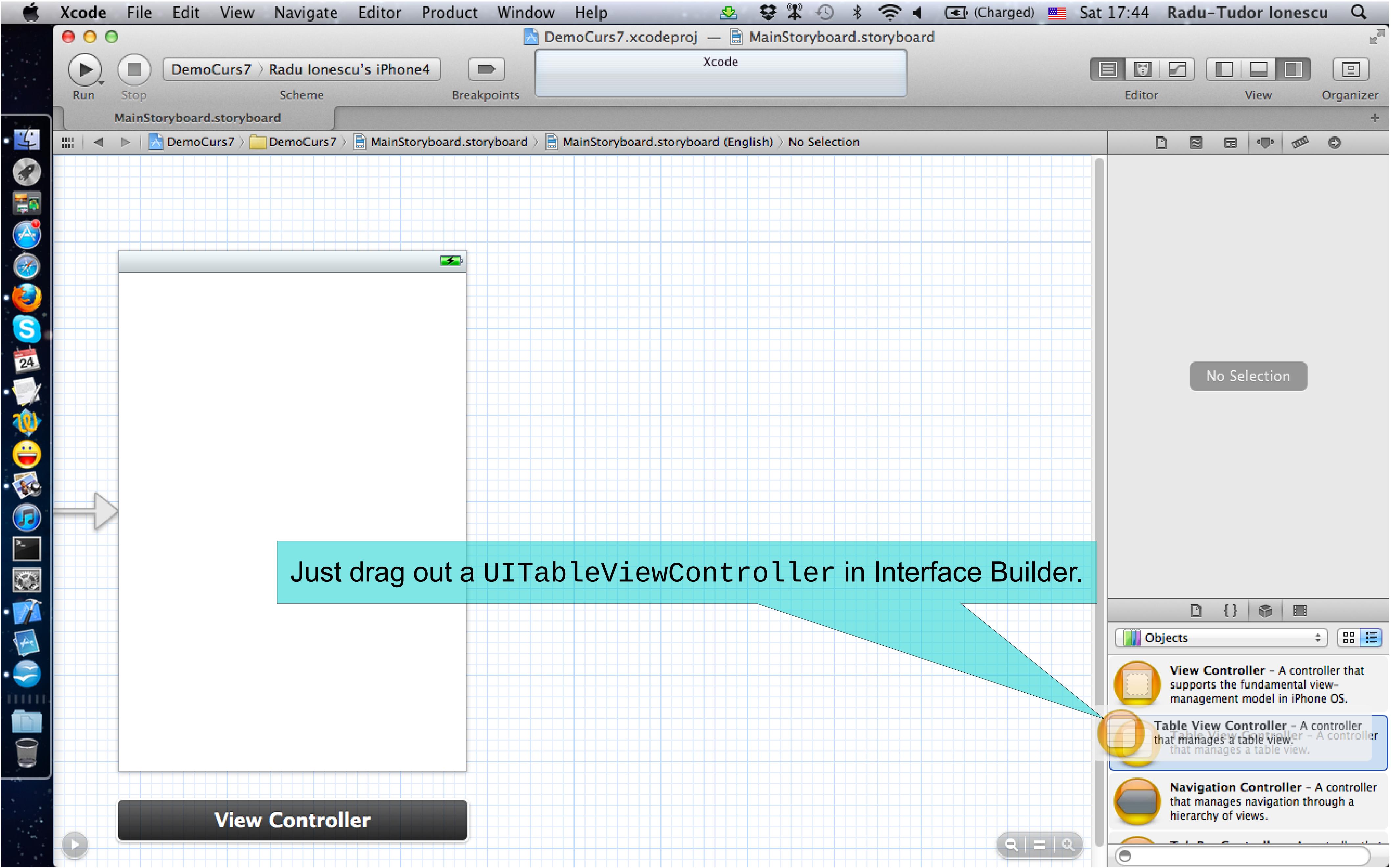
orange 16:15 100%

Barcelona	Spain
Bucharest	Romania
Istanbul	Turkey
Madrid	Spain
Mersin	Turkey
Milano	Italy
Nice	France
Paris	France
Rome	Italy
Toulouse	France

Left Detail
`UITableViewCellStyleValue2`

Creating Table View MVCs

`UITableViewController` is the iOS class used as the base class for MVC's that display `UITableViews`.

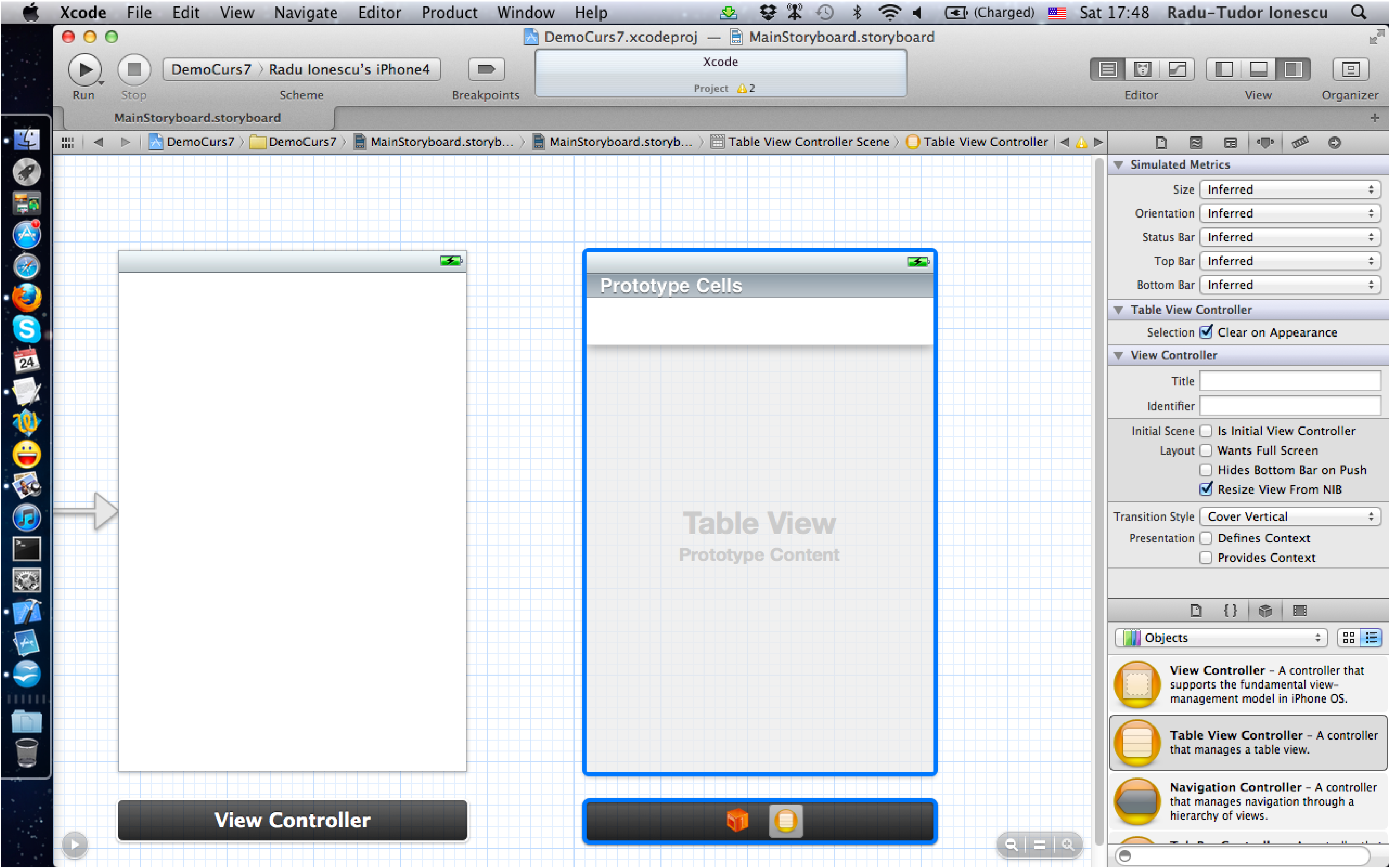


Just drag out a `UITableViewController` in Interface Builder.

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sat 17:44 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 〉 Radu Ionescu's iPhone4
Scheme
Breakpoints
Xcode
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 〉 DemoCurs7 〉 MainStoryboard.storyboard 〉 MainStoryboard.storyboard (English) 〉 No Selection
No Selection
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.

Creating Table View MVCs

`UITableViewController` is the iOS class used as the base class for MVC's that display `UITableViews`.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sat 17:48 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 〉 Radu Ionescu's iPhone4
Scheme
Breakpoints
Xcode
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 〉 DemoCurs7 〉 MainStoryboard.storyb... 〉 MainStoryboard.storyb... 〉 Table View Controller Scene 〉 Table View Controller
Prototype Cells
Table View
Prototype Content
View Controller

Simulated Metrics

Size: Inferred

Orientation: Inferred

Status Bar: Inferred

Top Bar: Inferred

Bottom Bar: Inferred

Table View Controller

Selection: ☑ Clear on Appearance

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

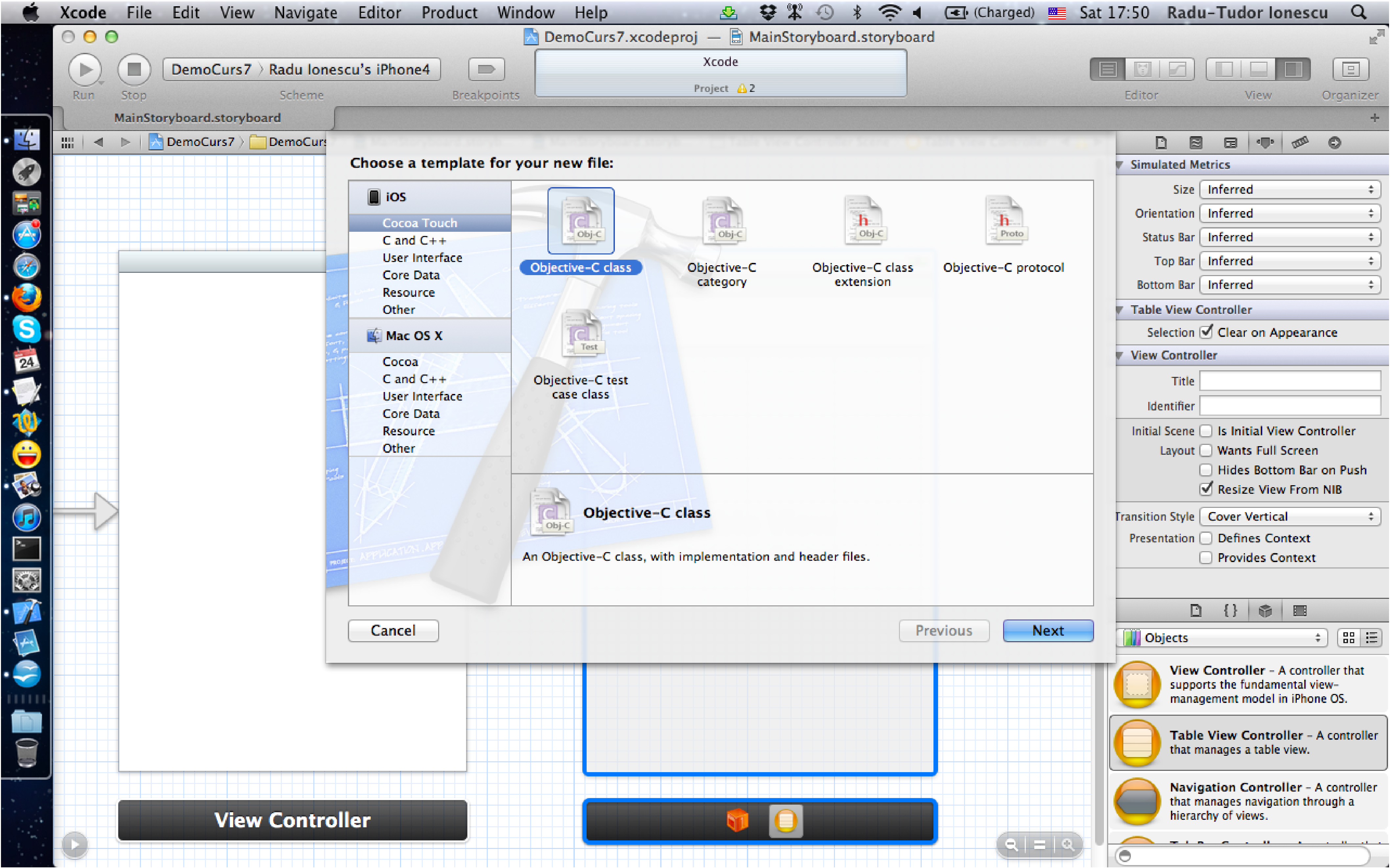
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Creating Table View MVCs

Choose "New File ..." from the File menu to create a custom subclass of `UITableViewController`.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sat 17:50 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 〉 Radu Ionescu's iPhone4
Scheme
Breakpoints
Xcode
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 〉 DemoCurs
View Controller

Choose a template for your new file:

Objective-C class

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

Simulated Metrics

Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Inferred

Table View Controller

Selection ☑ Clear on Appearance

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

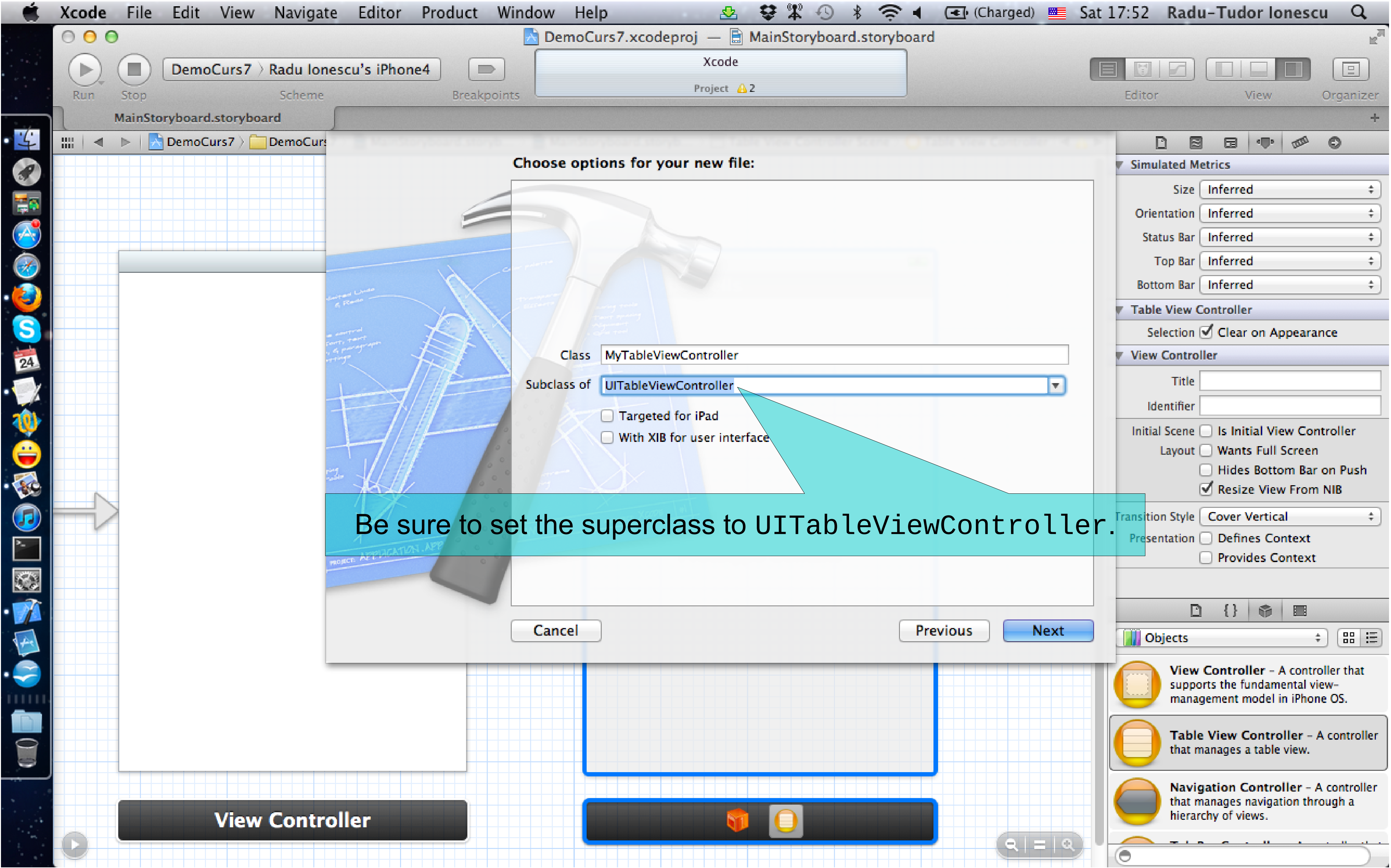
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Creating Table View MVCs

Choose “New File ...” from the File menu to create a custom subclass of `UITableViewController`.



Choose options for your new file:

Class: MyTableViewController

Subclass of: UITableViewController

□ Targeted for iPad

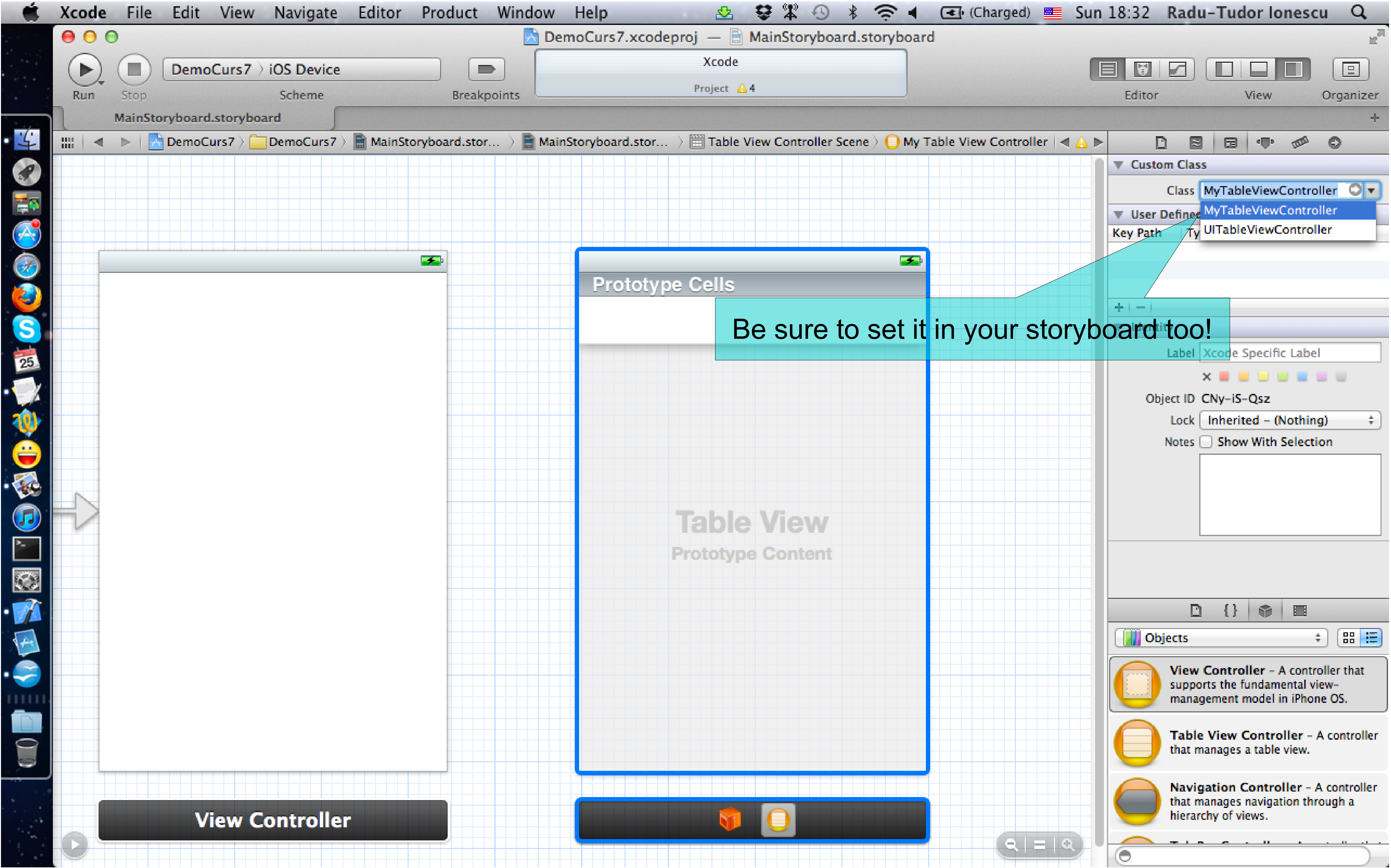
□ With XIB for user interface

Xcode File Edit View Navigate Editor Product Window Help
Sat 17:52 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 Radu Ionescu's iPhone4
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
Cancel
Previous
Next
Simulated Metrics
Size Inferred
Orientation Inferred
Status Bar Inferred
Top Bar Inferred
Bottom Bar Inferred
Table View Controller
Selection ☑ Clear on Appearance
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.
View Controller

Be sure to set the superclass to `UITableViewController`.

Creating Table View MVCs

Choose "New File ..." from the File menu to create a custom subclass of `UITableViewController`.

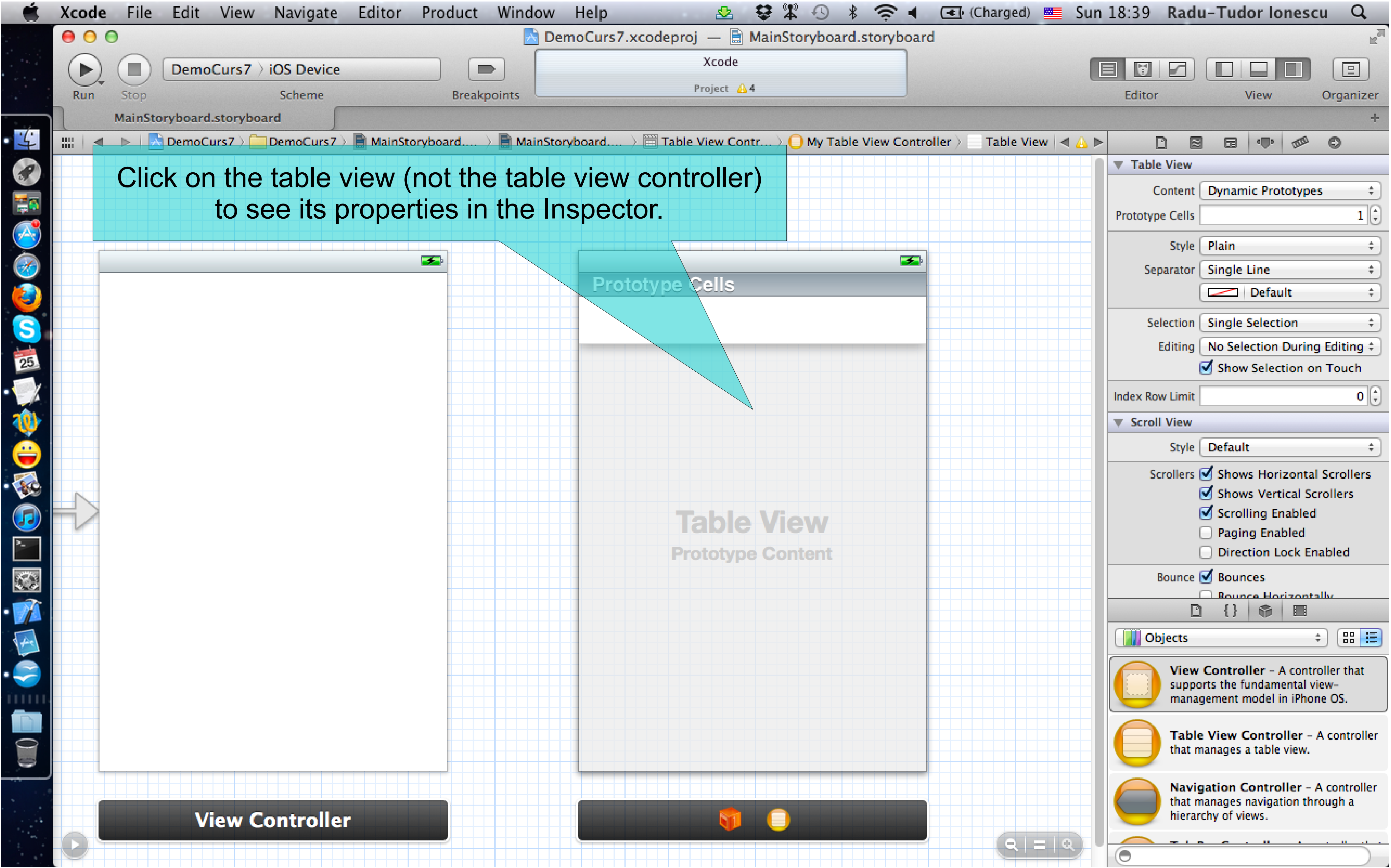


Be sure to set it in your storyboard too!

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 18:32 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 iOS Device
Scheme
Breakpoints
Xcode
Project 4
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 DemoCurs7 MainStoryboard.stor... MainStoryboard.stor... Table View Controller Scene My Table View Controller
Custom Class
Class MyTableViewController
MyTableViewController
UITableViewController
User Defined
Key Path
Label Xcode Specific Label
Object ID CNy-iS-Qsz
Lock Inherited - (Nothing)
Notes Show With Selection
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.

Creating Table View MVCs

You can customize both the look of the table view and its cells from Interface Builder.



Click on the table view (not the table view controller) to see its properties in the Inspector.

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 18:39 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠4
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStoryboard.... › MainStoryboard.... › Table View Contr... › My Table View Controller › Table View
Prototype Cells
Table View
Prototype Content
View Controller

Table View

- Content: Dynamic Prototypes
- Prototype Cells: 1
- Style: Plain
- Separator: Single Line
- Default
- Selection: Single Selection
- Editing: No Selection During Editing
- ☑ Show Selection on Touch
- Index Row Limit: 0

Scroll View

- Style: Default
- Scrollers: ☑ Shows Horizontal Scrollers
- ☑ Shows Vertical Scrollers
- ☑ Scrolling Enabled
- □ Paging Enabled
- □ Direction Lock Enabled
- Bounce: ☑ Bounces
- □ Bounce Horizontally

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.

Creating Table View MVCs

You can customize both the look of the table view and its cells from Interface Builder.

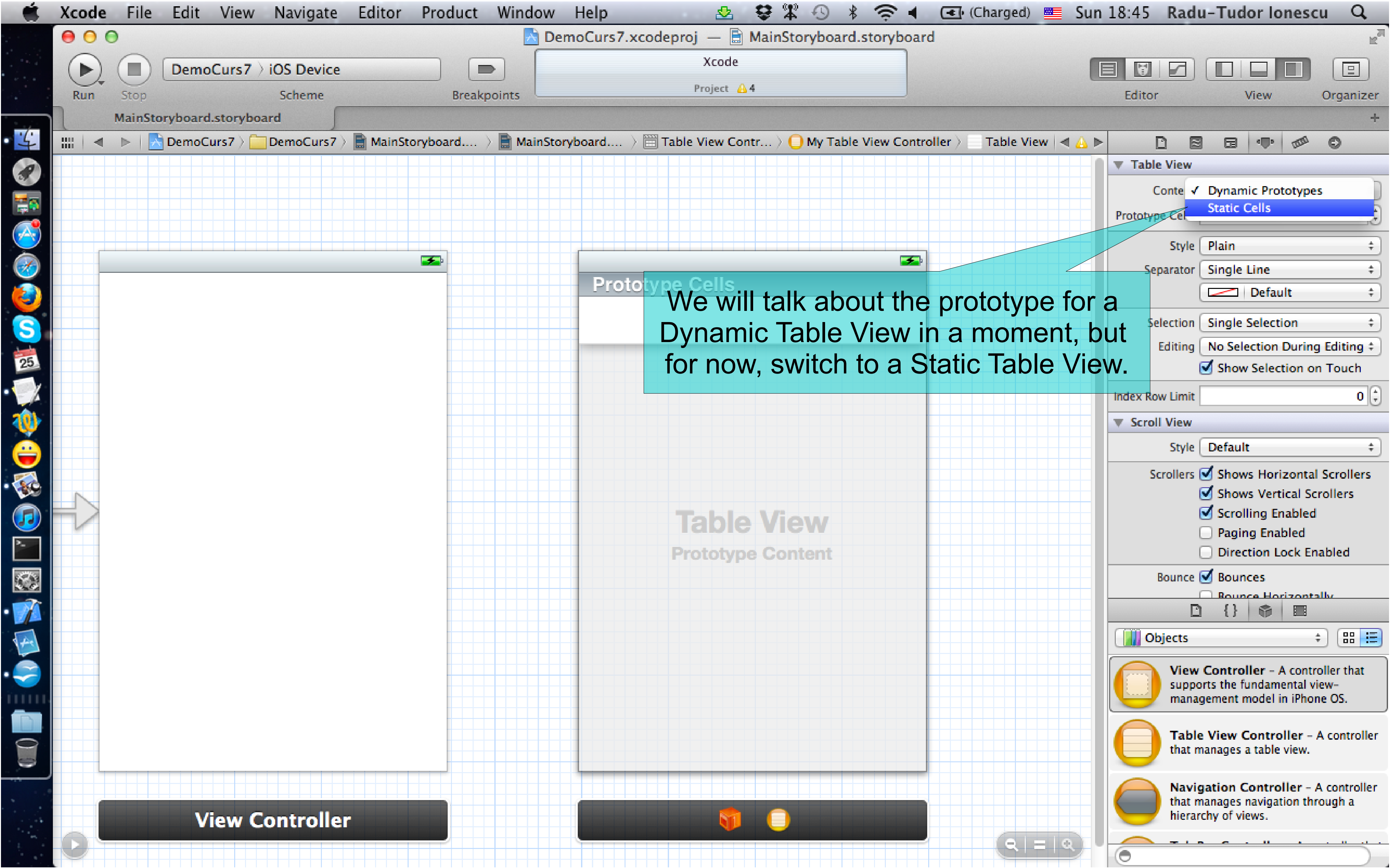


Table View

Content: ✓ Dynamic Prototypes / Static Cells

Prototype Cells

Style: Plain

Separator: Single Line

Default

Selection: Single Selection

Editing: No Selection During Editing

☑ Show Selection on Touch

Index Row Limit: 0

Scroll View

Style: Default

Scrollers: ☑ Shows Horizontal Scrollers

☑ Shows Vertical Scrollers

☑ Scrolling Enabled

□ Paging Enabled

□ Direction Lock Enabled

Bounce: ☑ Bounces

□ Bounce Horizontally

Objects

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

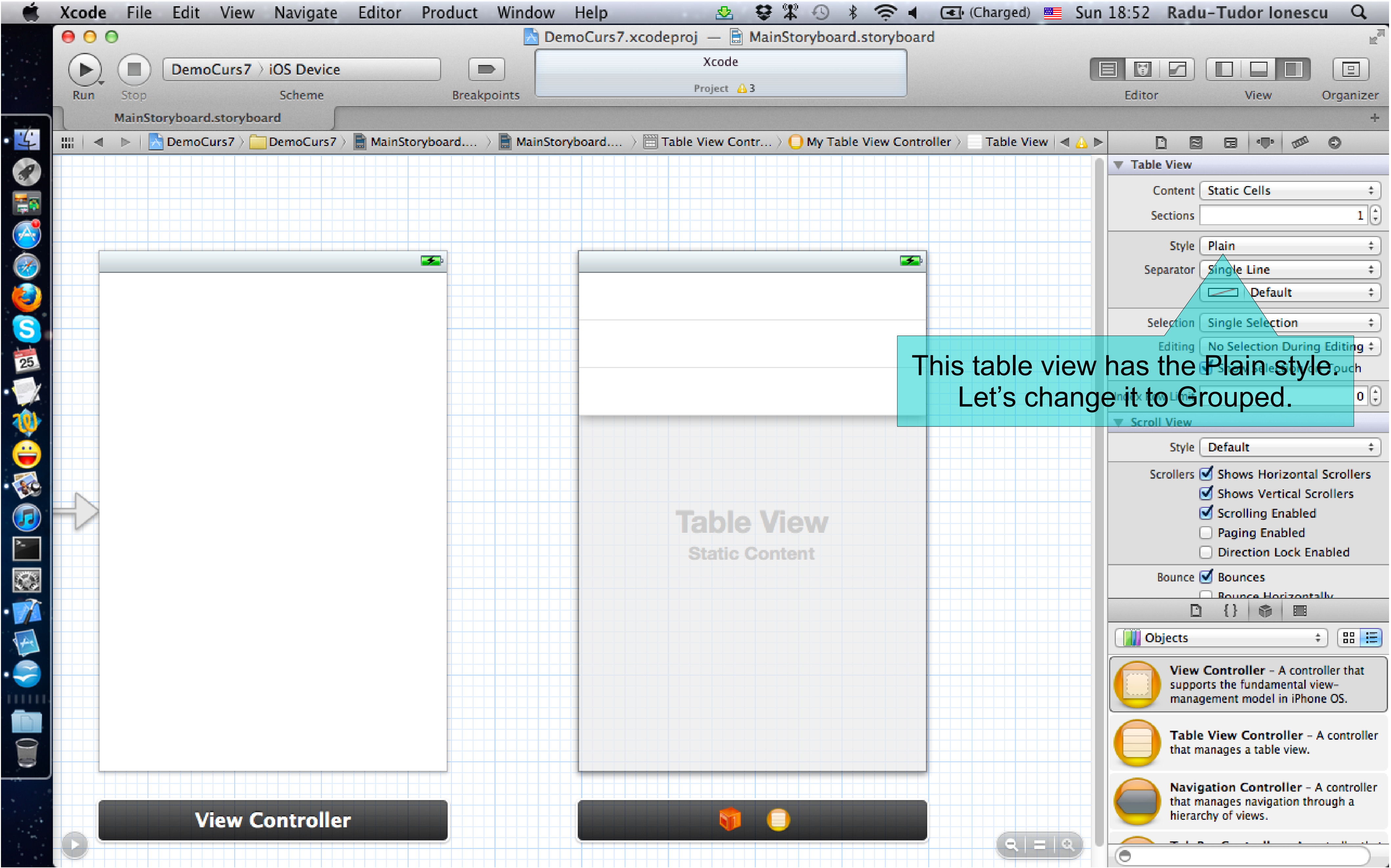
Table View Controller – A controller that manages a table view.

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

We will talk about the prototype for a Dynamic Table View in a moment, but for now, switch to a Static Table View.

Creating Table View MVCs

You can customize both the look of the table view and its cells from Interface Builder.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 18:52 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStoryboard.... › MainStoryboard.... › Table View Contr... › My Table View Controller › Table View
Table View
Static Content
View Controller

Table View

Content: Static Cells

Sections: 1

Style: Plain

Separator: Single Line

Default

Selection: Single Selection

Editing: No Selection During Editing

☑ Show Selection on Touch

Index Row Limit: 0

Scroll View

Style: Default

Scrollers: ☑ Shows Horizontal Scrollers

☑ Shows Vertical Scrollers

☑ Scrolling Enabled

☐ Paging Enabled

☐ Direction Lock Enabled

Bounce: ☑ Bounces

☐ Bounce Horizontally

Objects

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

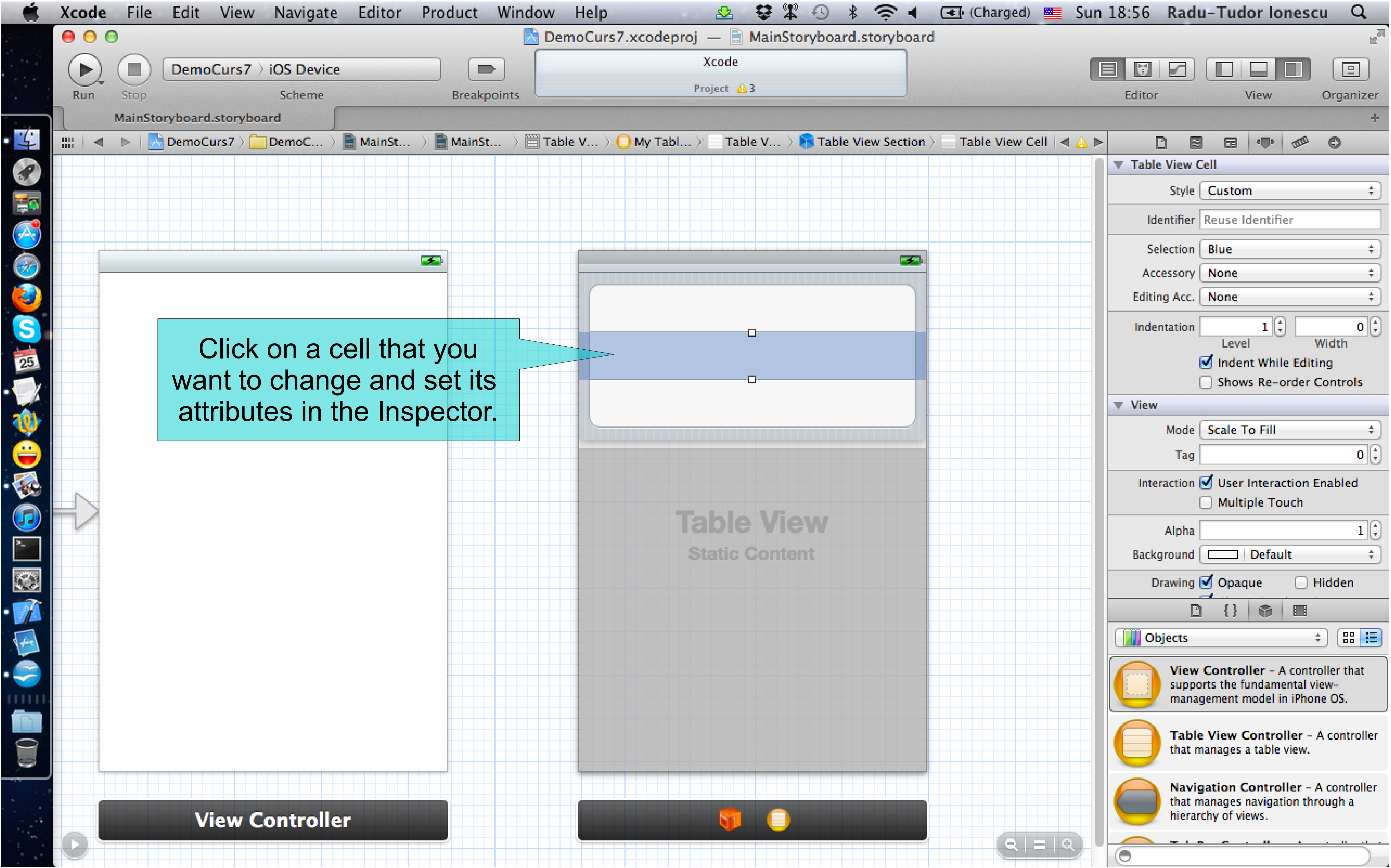
Table View Controller – A controller that manages a table view.

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

This table view has the Plain style. Let's change it to Grouped.

Creating Table View MVCs

And you can change the look of each cell as well.



Click on a cell that you want to change and set its attributes in the Inspector.

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 18:56 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC... › MainSt... › MainSt... › Table V... › My Tabl... › Table V... › Table View Section › Table View Cell
Table View
Static Content
View Controller

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

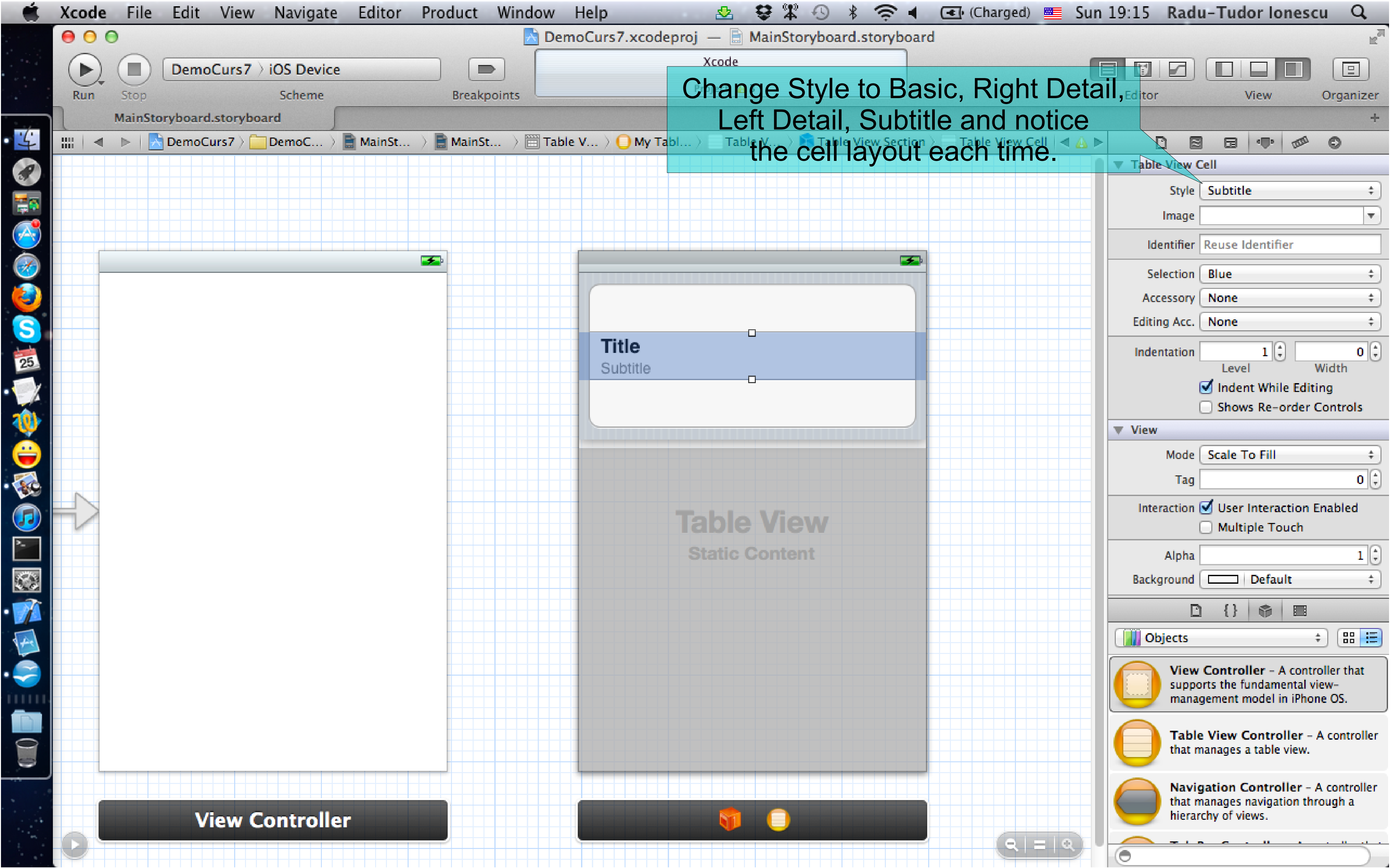
- Mode: Scale To Fill
- Tag: 0
- Interaction: ☑ User Interaction Enabled
- ☐ Multiple Touch
- Alpha: 1
- Background: Default
- Drawing: ☑ Opaque ☐ Hidden

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.

Creating Table View MVCs

And you can change the look of each cell as well.



Change Style to Basic, Right Detail, Left Detail, Subtitle and notice the cell layout each time.

Table View Cell

Style: Subtitle

Image:

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

Objects

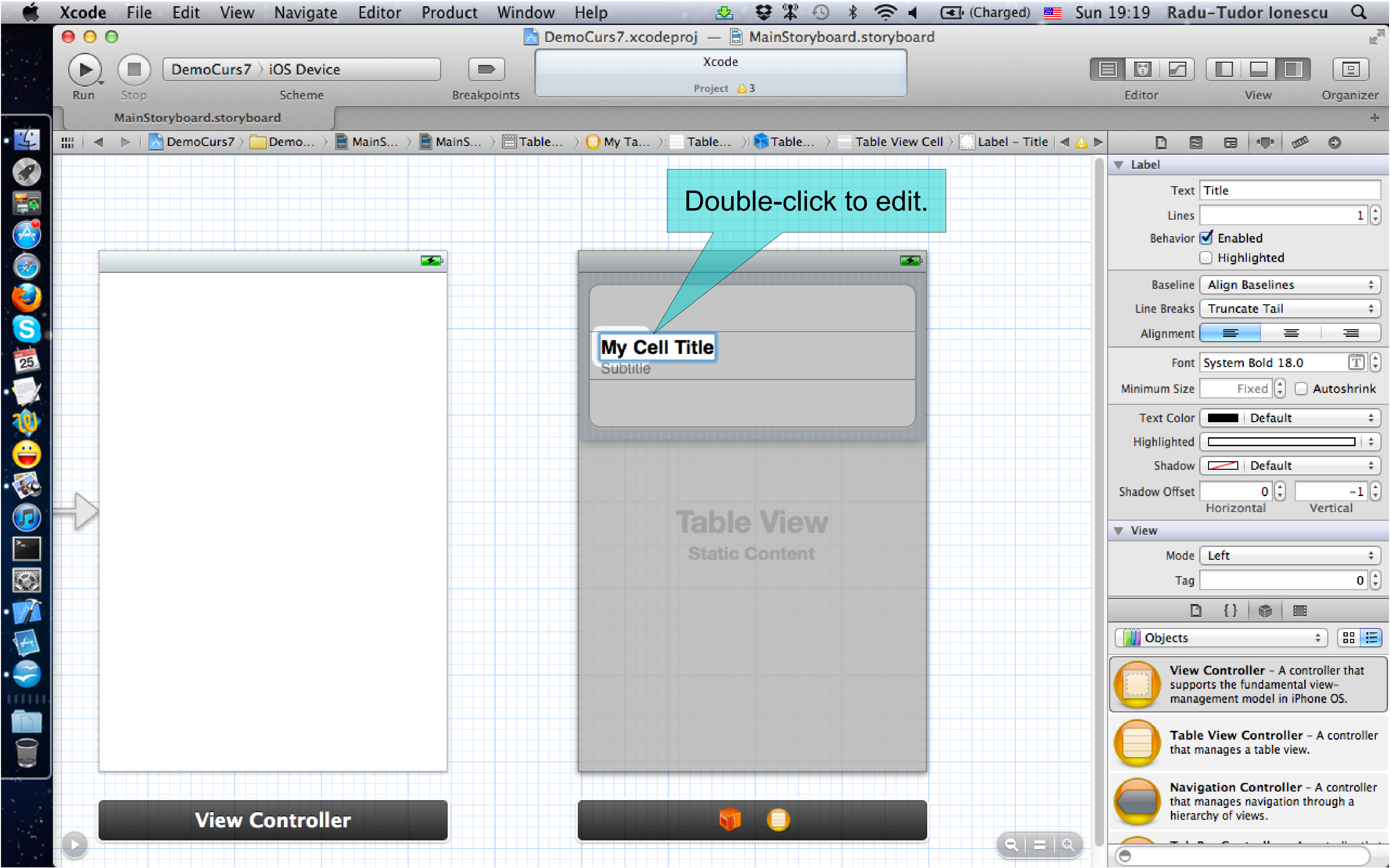
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Creating Table View MVCs

And you can change the look of each cell as well.



Double-click to edit.

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:19 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › Demo… › MainS… › MainS… › Table… › My Ta… › Table… › Table… › Table View Cell › Label – Title
My Cell Title
Subtitle
Table View
Static Content
View Controller

Label

Text: Title

Lines: 1

Behavior: ☑ Enabled

□ Highlighted

Baseline: Align Baselines

Line Breaks: Truncate Tail

Alignment

Font: System Bold 18.0

Minimum Size: Fixed □ Autoshrink

Text Color: Default

Highlighted

Shadow: Default

Shadow Offset: 0 Horizontal, -1 Vertical

View

Mode: Left

Tag: 0

Objects

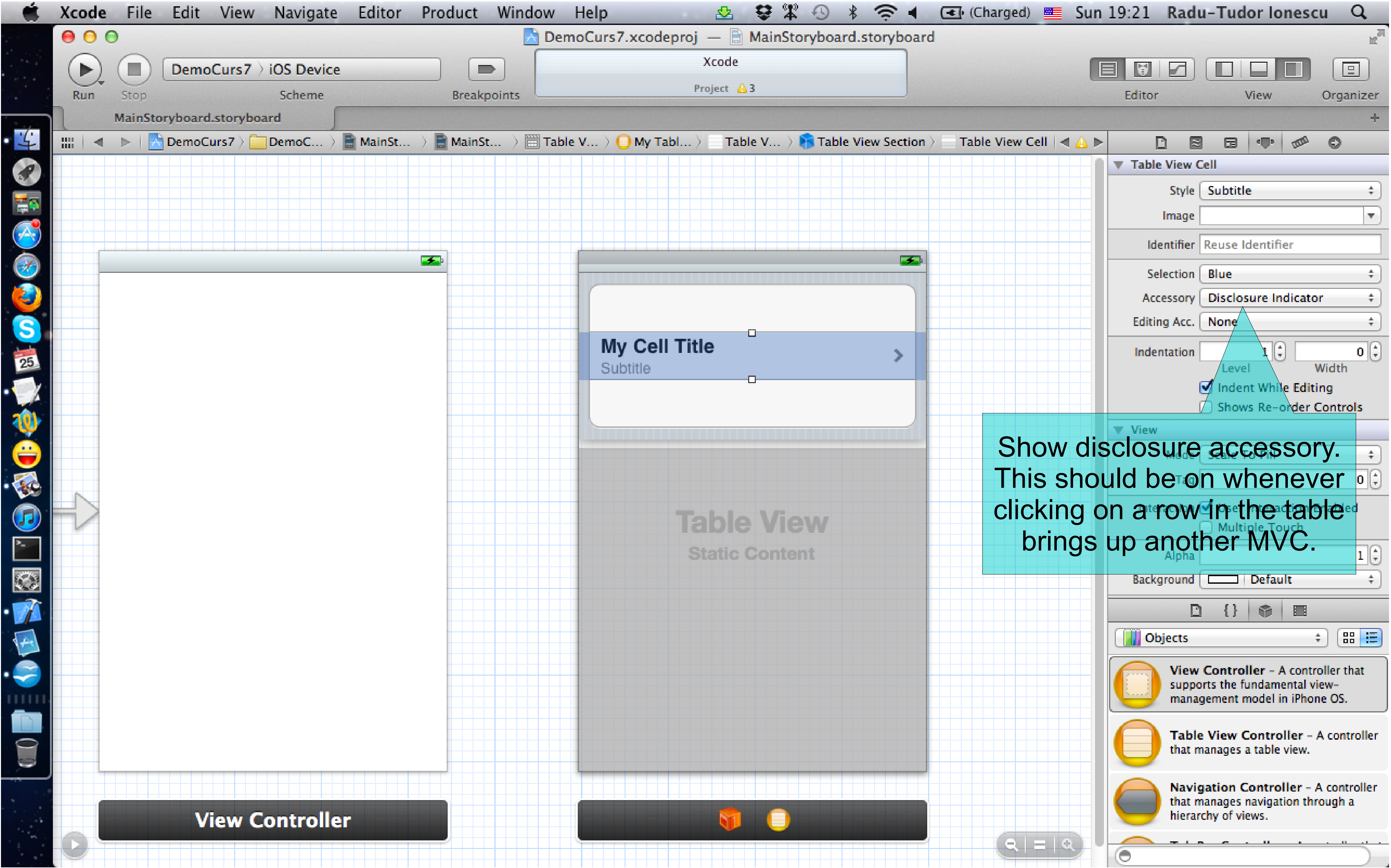
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Creating Table View MVCs

And you can change the look of each cell as well.



Style: Subtitle

Image:

Identifier: Reuse Identifier

Selection: Blue

Accessory: Disclosure Indicator

Editing Acc.: None

Indentation: 1 (Level), 0 (Width)

☑ Indent While Editing

□ Shows Re-order Controls

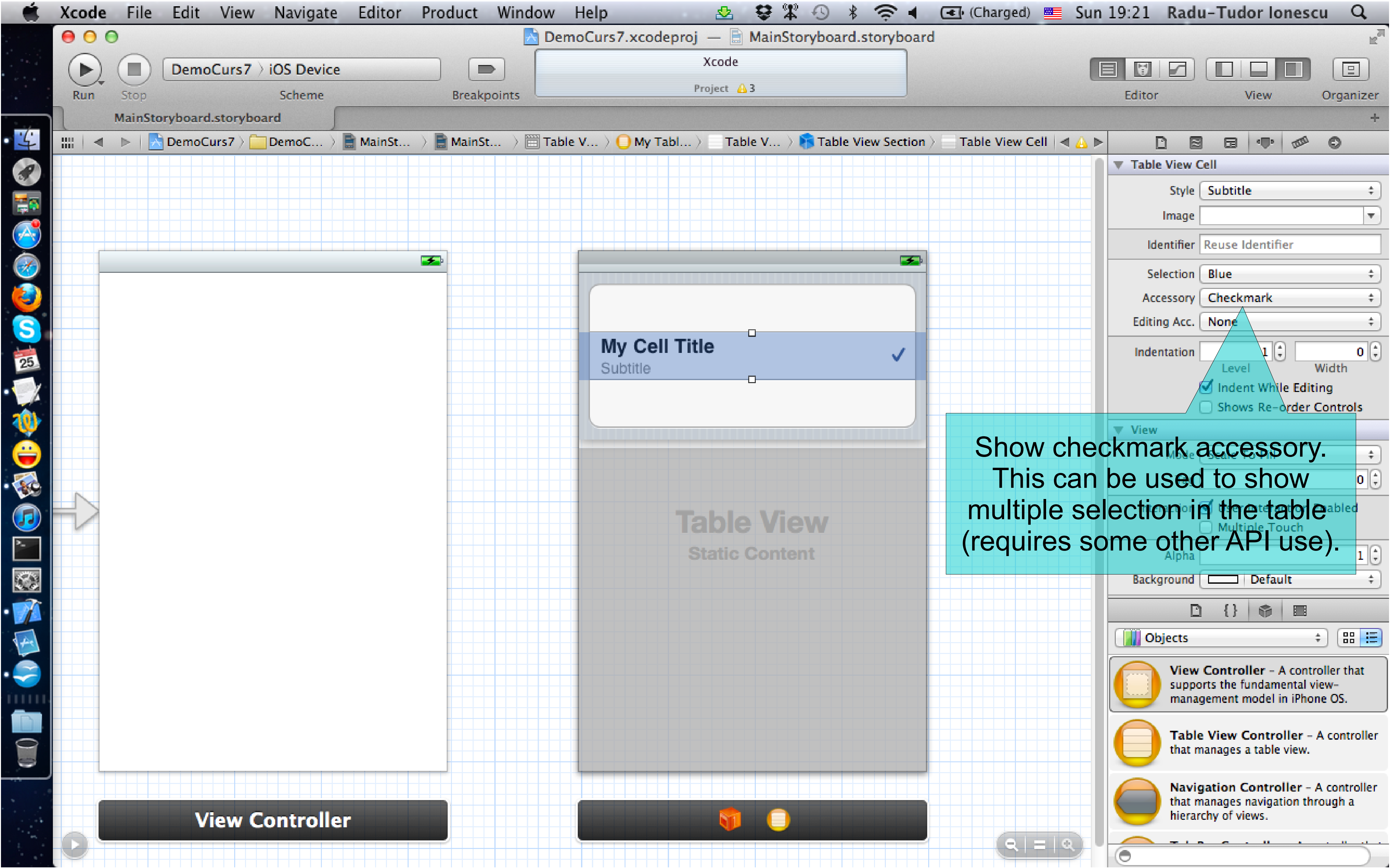
Background: Default

Show disclosure accessory. This should be on whenever clicking on a row in the table brings up another MVC.

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:21 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints
DemoCurs7 › iOS Device
Xcode
Project ⚠3
Editor View Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC... › MainSt... › MainSt... › Table V... › My Tabl... › Table V... › Table View Section › Table View Cell
Table View Cell
View
My Cell Title
Subtitle
Table View
Static 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.

Creating Table View MVCs

And you can change the look of each cell as well.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:21 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC... › MainSt... › MainSt... › Table V... › My Tabl... › Table V... › Table View Section › Table View Cell
My Cell Title
Subtitle
✓
Table View
Static Content
View Controller

Table View Cell

Style: Subtitle

Image:

Identifier: Reuse Identifier

Selection: Blue

Accessory: Checkmark

Editing Acc.: None

Indentation: Level 1, Width 0

☑ Indent While Editing

□ Shows Re-order Controls

View

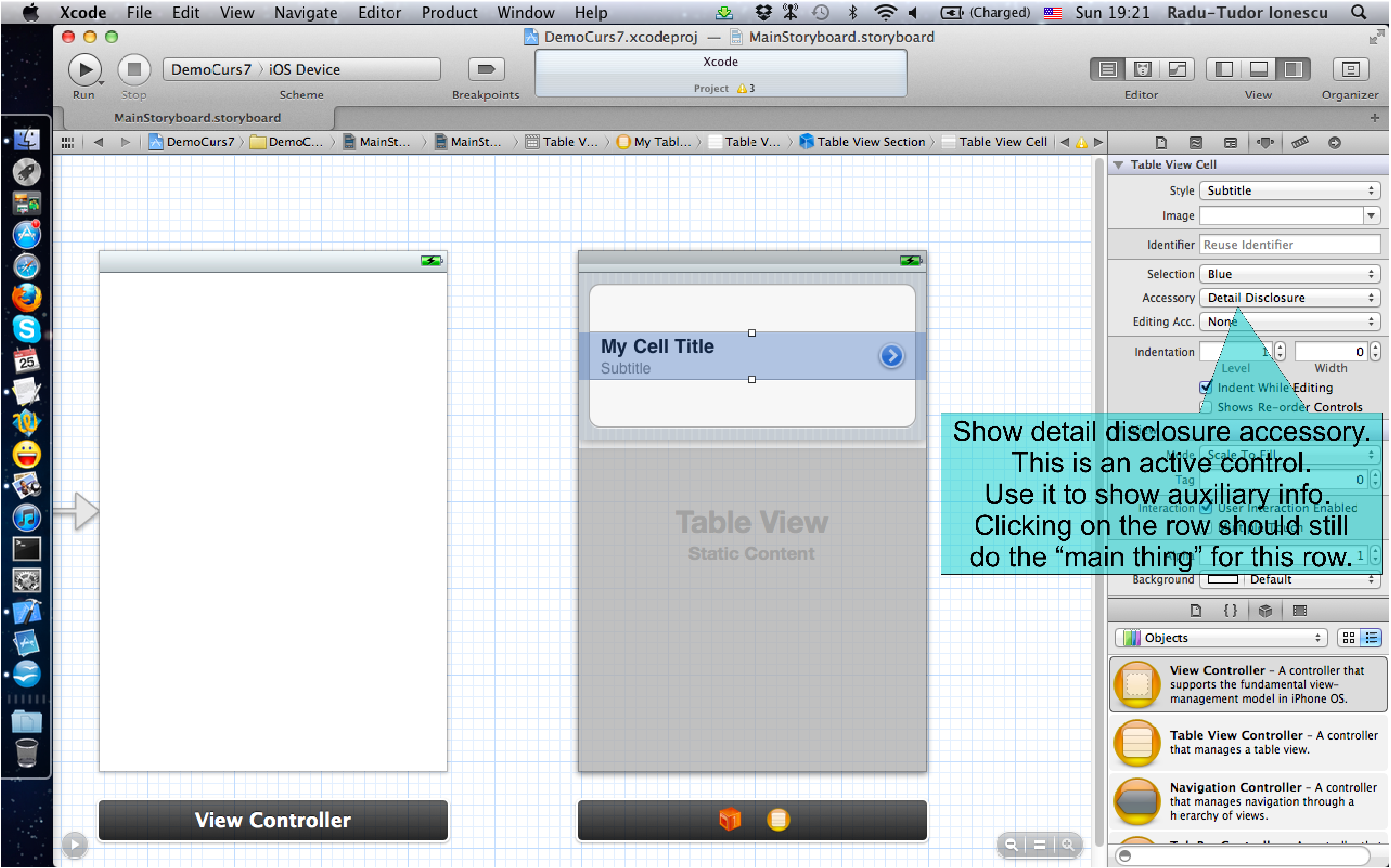
Background: Default

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.

Show checkmark accessory.
This can be used to show multiple selection in the table (requires some other API use).

Creating Table View MVCs

And you can change the look of each cell as well.



Xcode File Edit View Navigate Editor Product Window Help
Sun 19:21 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 iOS Device
Scheme
Breakpoints
Xcode
Project 3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 DemoC… MainSt… MainSt… Table V… My Tabl… Table V… Table View Section Table View Cell
My Cell Title
Subtitle
Table View
Static Content
View Controller
Objects

Table View Cell

Style	Subtitle
Image	
Identifier	Reuse Identifier
Selection	Blue
Accessory	Detail Disclosure
Editing Acc.	None
Indentation	1 Level, 0 Width

☑ Indent While Editing
□ Shows Re-order Controls

Background Default

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

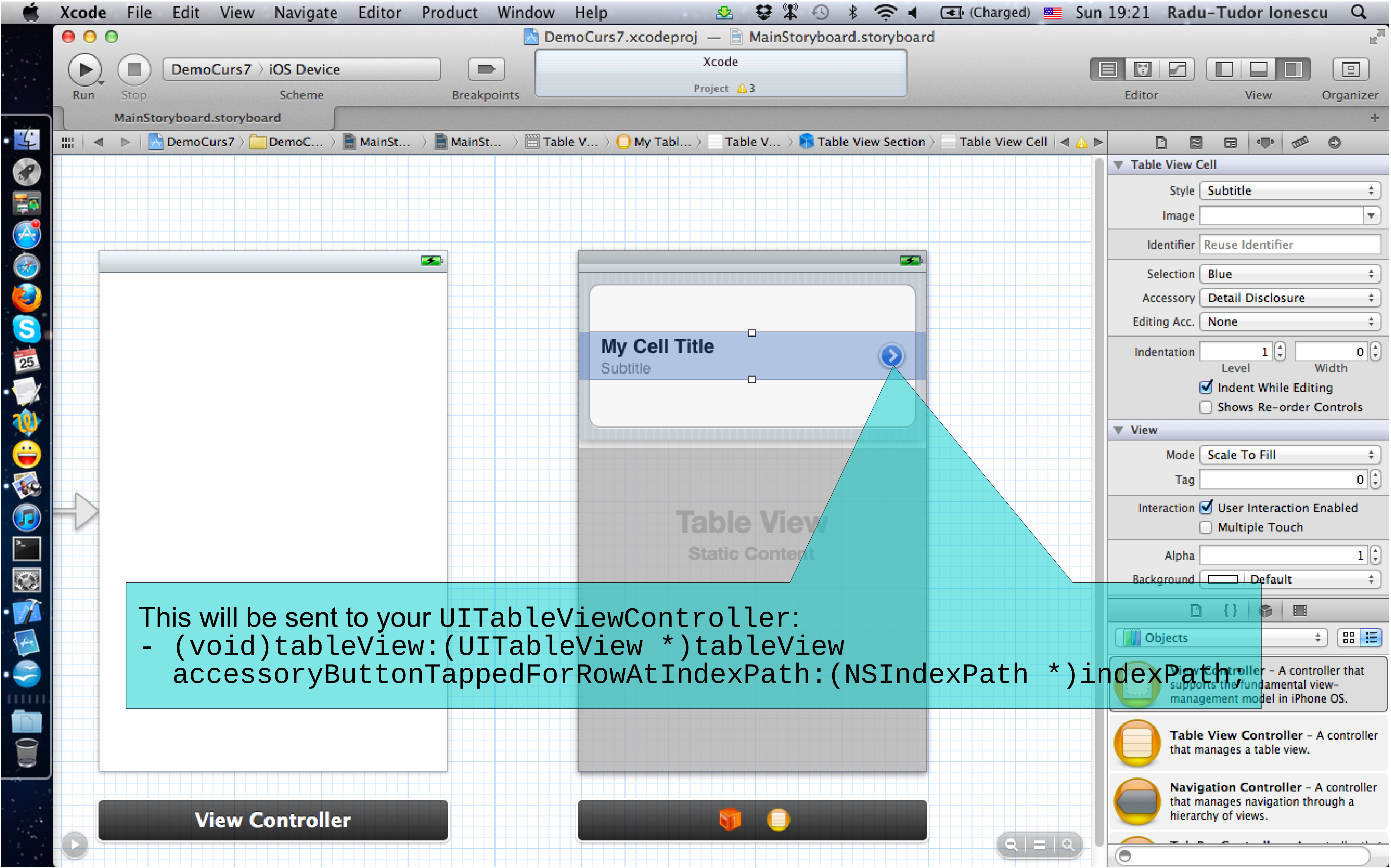
Table View Controller – A controller that manages a table view.

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

Show detail disclosure accessory.
This is an active control.
Use it to show auxiliary info.
Clicking on the row should still
do the “main thing” for this row.

Creating Table View MVCs

User taps on the blue detail disclosure below?

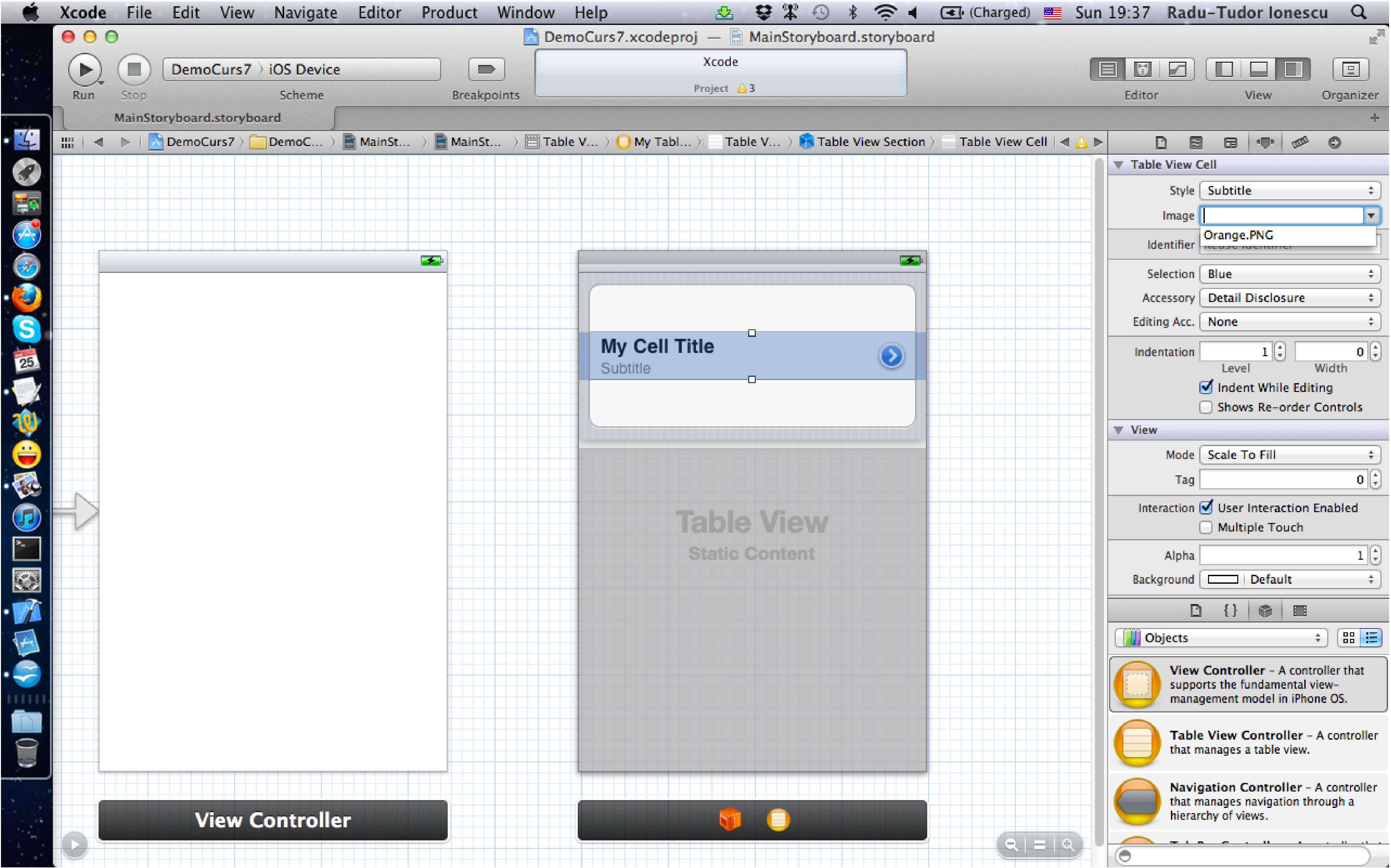

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:21 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC… › MainSt… › MainSt… › Table V… › My Tabl… › Table V… › Table View Section › Table View Cell
My Cell Title
Subtitle
Table View
Static Content
View Controller
Table View Cell
Style Subtitle
Image
Identifier Reuse Identifier
Selection Blue
Accessory Detail Disclosure
Editing Acc. None
Indentation 1 0
Level 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
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 will be sent to your `UITableViewController`:

```
- (void)tableView:(UITableView *)tableView
  accessoryButtonTappedForRowAtIndexPath:(NSIndexPath *)indexPath;
```

Creating Table View MVCs

Notice that some cell styles can have an image.
You can set this in the code as well (more in a moment on this).



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:37 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC... › MainSt... › MainSt... › Table V... › My Tabl... › Table V... › Table View Section › Table View Cell
My Cell Title
Subtitle
Table View
Static Content
View Controller
Objects

Table View Cell

Style: Subtitle

Image: Orange.PNG

Identifier: Reuse Identifier

Selection: Blue

Accessory: Detail Disclosure

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

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.

Creating Table View MVCs

Notice that some cell styles can have an image.
You can set this in the code as well (more in a moment on this).

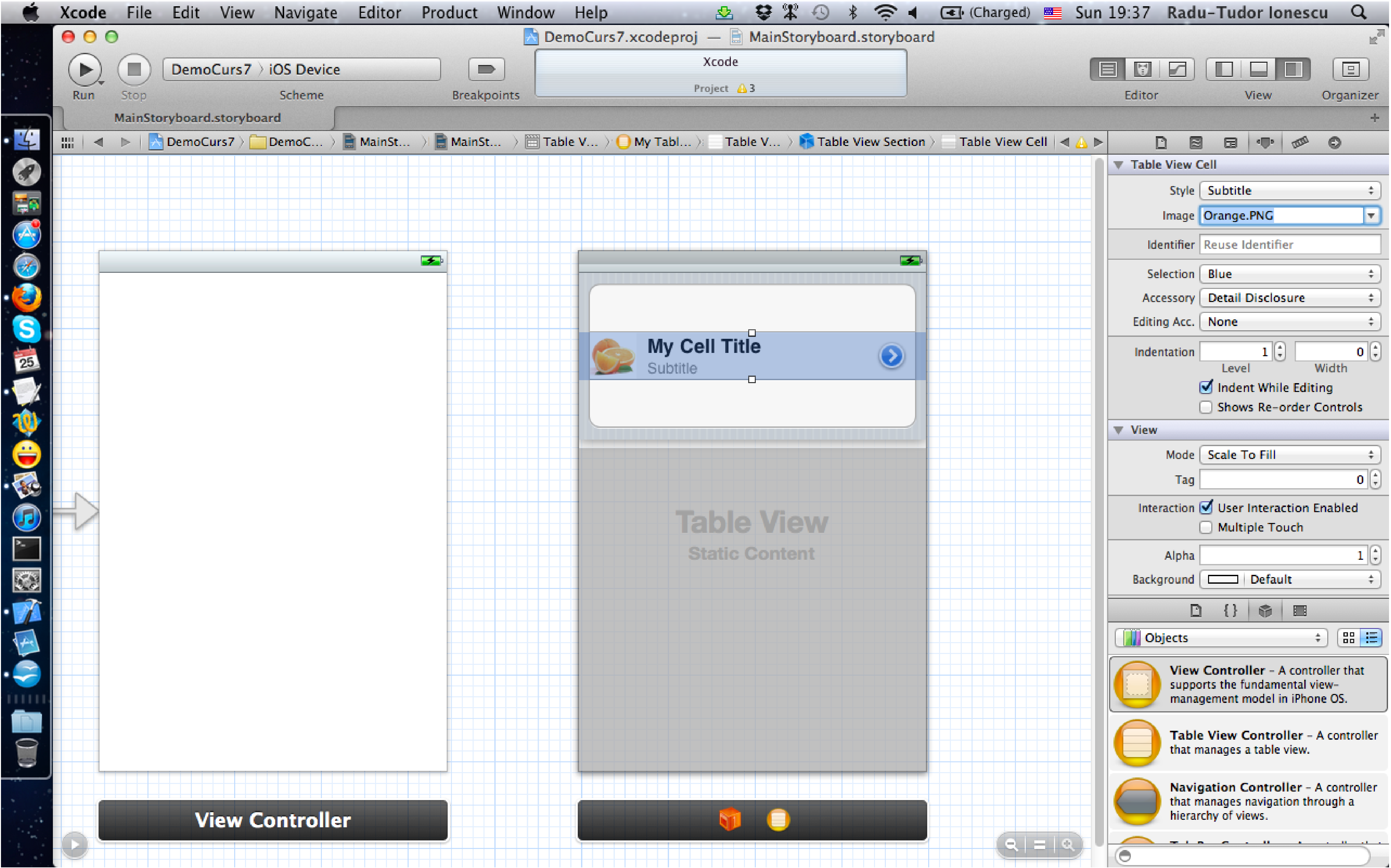


Table View Cell

Style: Subtitle
Image: Orange.PNG
Identifier: Reuse Identifier
Selection: Blue
Accessory: Detail Disclosure
Editing Acc.: None
Indentation: Level 1, Width 0
☑ Indent While Editing
□ Shows Re-order Controls

View

Mode: Scale To Fill
Tag: 0
Interaction: ☑ User Interaction Enabled
□ Multiple Touch
Alpha: 1
Background: Default

Objects

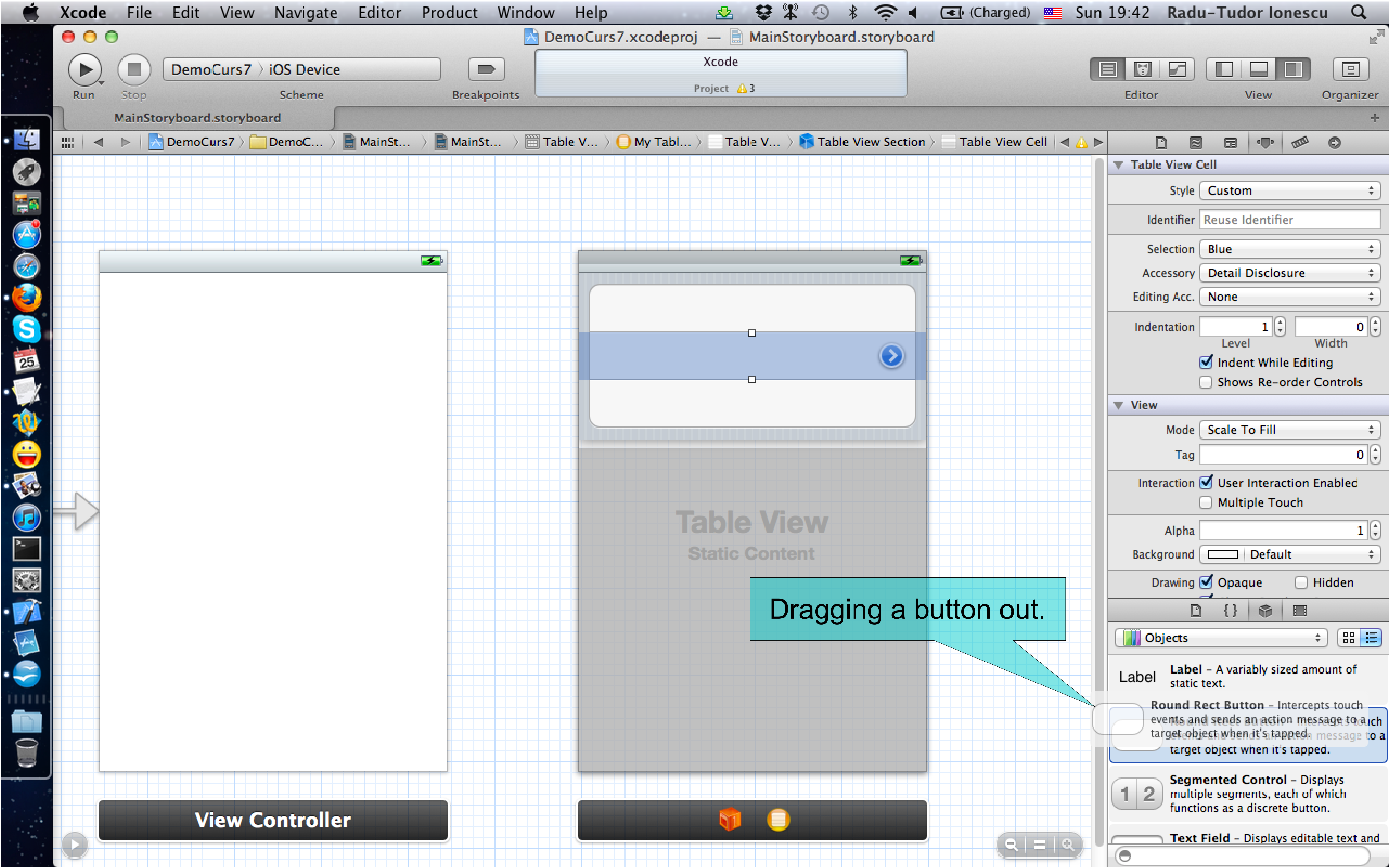
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Creating Table View MVCs

In the Custom style, you can drag out views and wire them up as outlets!



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:42 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iOS Device
Scheme
Breakpoints
Xcode
Project ⚠ 3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC… › MainSt… › MainSt… › Table V… › My Tabl… › Table V… › Table View Section › Table View Cell
Table View
Static Content
View Controller

Table View Cell

Style: Custom

Identifier: Reuse Identifier

Selection: Blue

Accessory: Detail Disclosure

Editing Acc.: None

Indentation: Level 1, Width 0

☑ 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

Label – A variably sized amount of static text.

Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.

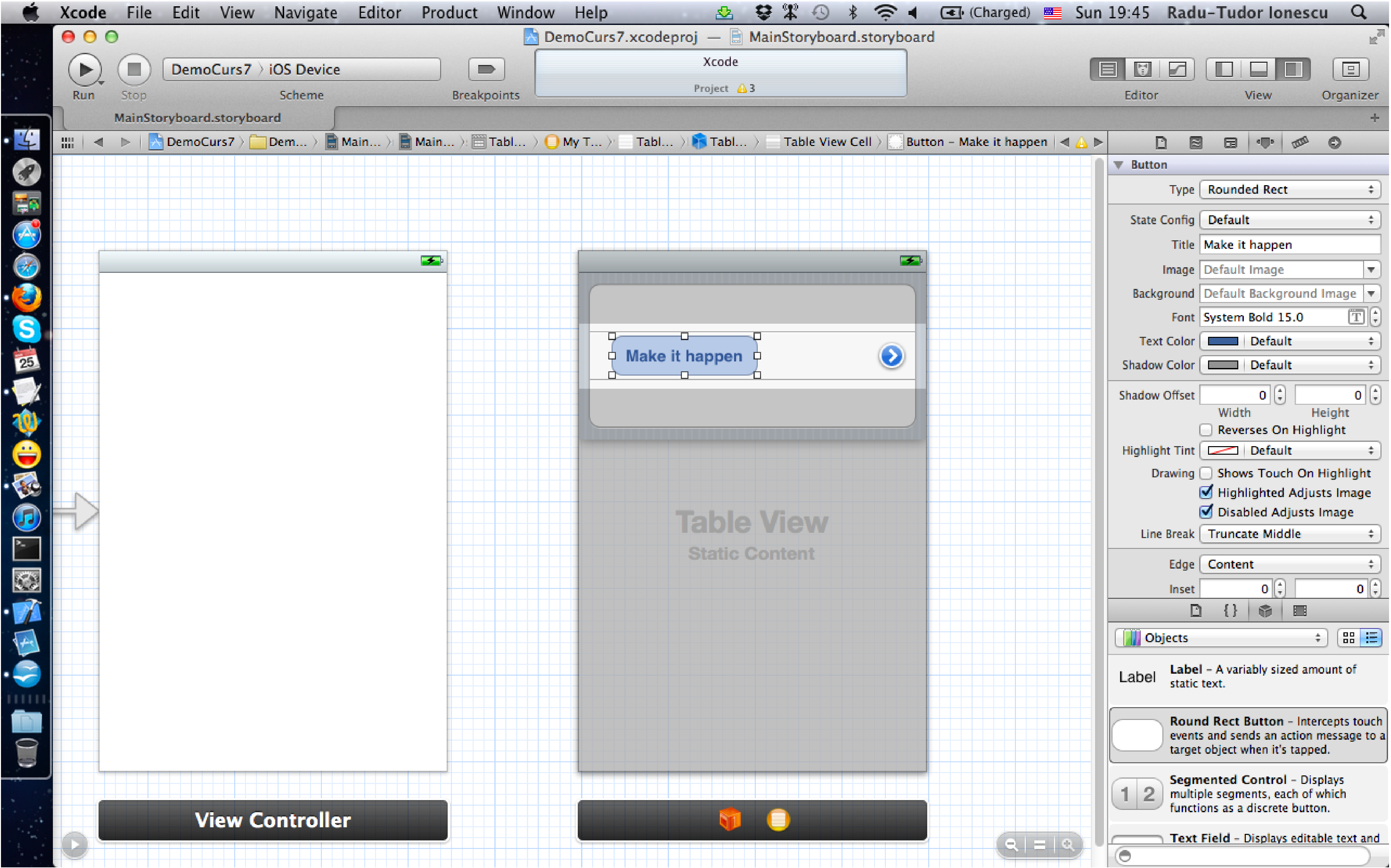
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Field – Displays editable text and

Dragging a button out.

Creating Table View MVCs

In the Custom style, you can drag out views and wire them up as outlets!



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sun 19:45 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 iOS Device
Scheme
Breakpoints
Xcode
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 Dem... Main... Main... Tabl... My T... Tabl... Tabl... Table View Cell Button – Make it happen
Make it happen
Table View
Static Content
View Controller

Button

Type: Rounded Rect
State Config: Default
Title: Make it happen
Image: Default Image
Background: Default Background Image
Font: System Bold 15.0
Text Color: Default
Shadow Color: Default
Shadow Offset: 0 Width, 0 Height
□ Reverses On Highlight
Highlight Tint: Default
Drawing: □ Shows Touch On Highlight
☑ Highlighted Adjusts Image
☑ Disabled Adjusts Image
Line Break: Truncate Middle
Edge: Content
Inset: 0 0

Objects

Label – A variably sized amount of static text.

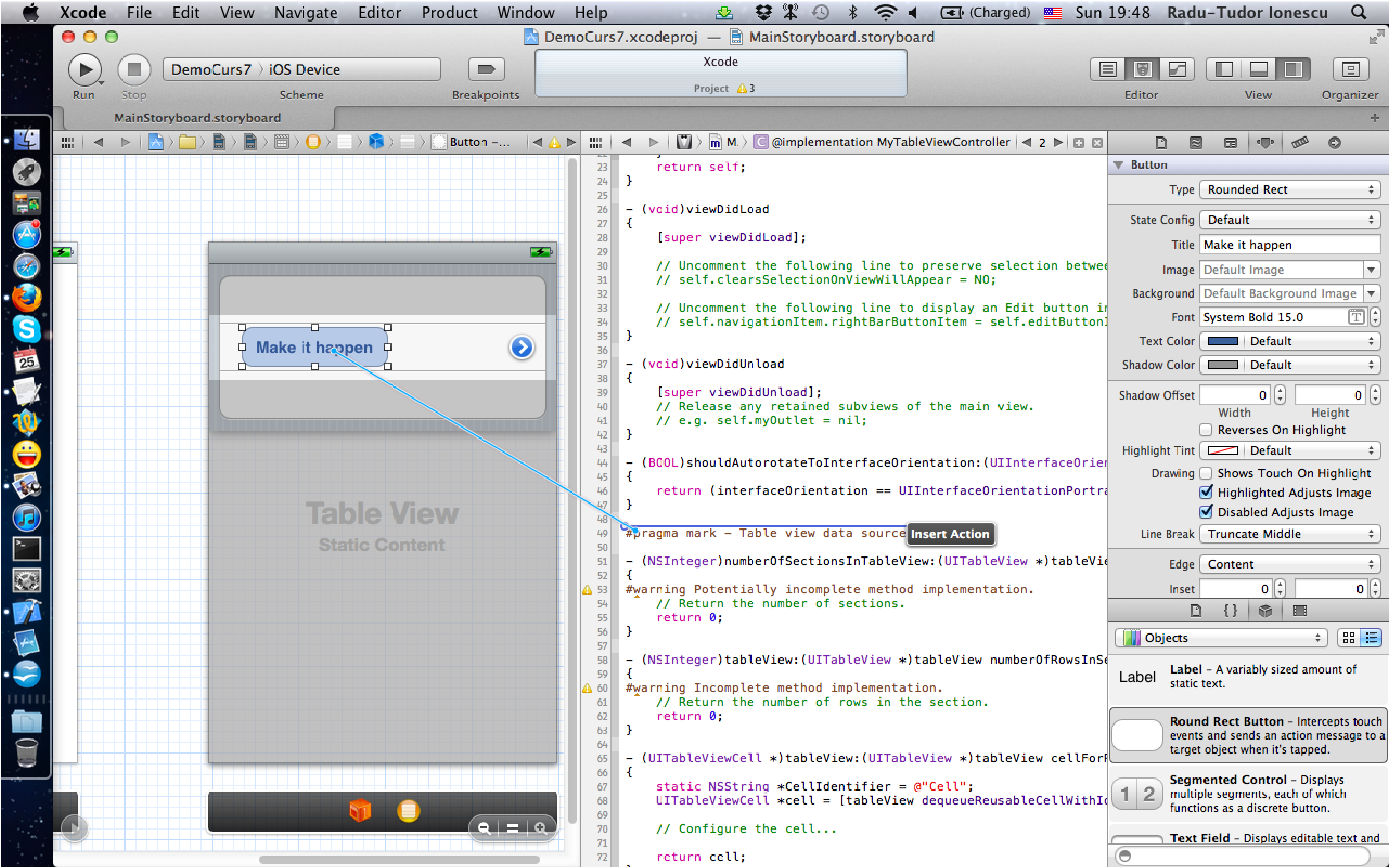
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Field – Displays editable text and

Creating Table View MVCs

In the Custom style, you can drag out views and wire them up as outlets!



```
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

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

    // Uncomment the following line to display an Edit button i
    // self.navigationItem.rightBarButtonItem = self.editButtonI
}

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

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrie
{
    return (interfaceOrientation == UIInterfaceOrientationPortra
}

#pragma mark - Table view data source

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

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

- (UITableViewCell *)tableView:(UITableView *)tableView cellForR
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithId

    // Configure the cell...

    return cell;
```

Button

Type: Rounded Rect
State Config: Default
Title: Make it happen
Image: Default Image
Background: Default Background Image
Font: System Bold 15.0
Text Color: Default
Shadow Color: Default
Shadow Offset: 0 (Width) 0 (Height)
□ Reverses On Highlight
Highlight Tint: Default
Drawing: □ Shows Touch On Highlight
☑ Highlighted Adjusts Image
☑ Disabled Adjusts Image
Line Break: Truncate Middle
Edge: Content
Inset: 0 0

Objects

Label – A variably sized amount of static text.

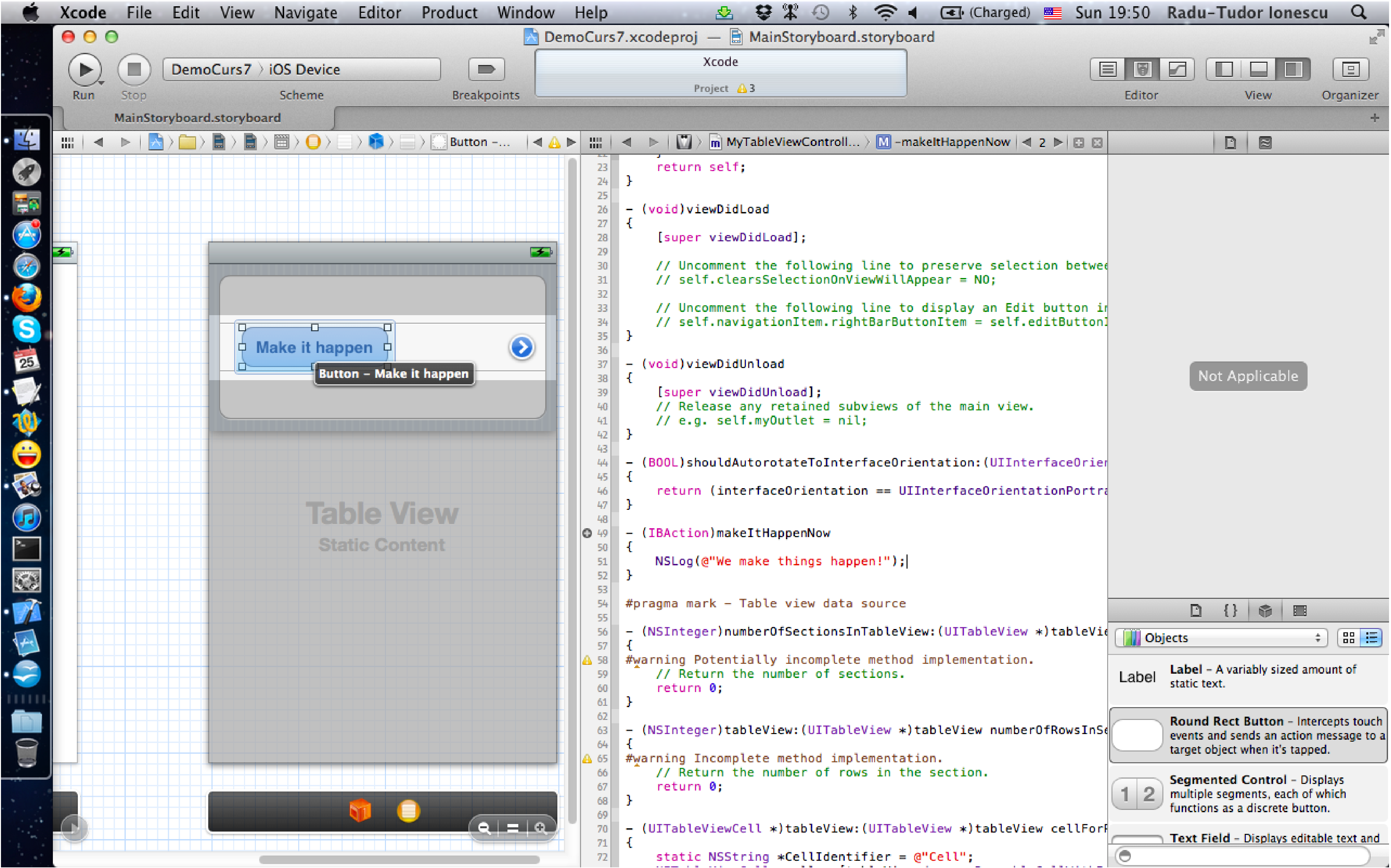
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Field – Displays editable text and

Creating Table View MVCs

In the Custom style, you can drag out views and wire them up as outlets!



```
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

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

    // Uncomment the following line to display an Edit button ir
    // self.navigationItem.rightBarButtonItem = self.editButtonI
}

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

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrier
{
    return (interfaceOrientation == UIInterfaceOrientationPortra
}

- (IBAction)makeItHappenNow
{
    NSLog(@"We make things happen!");
}

#pragma mark - Table view data source

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

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

- (UITableViewCell *)tableView:(UITableView *)tableView cellForR
{
    static NSString *CellIdentifier = @"Cell";
```

Label – A variably sized amount of static text.

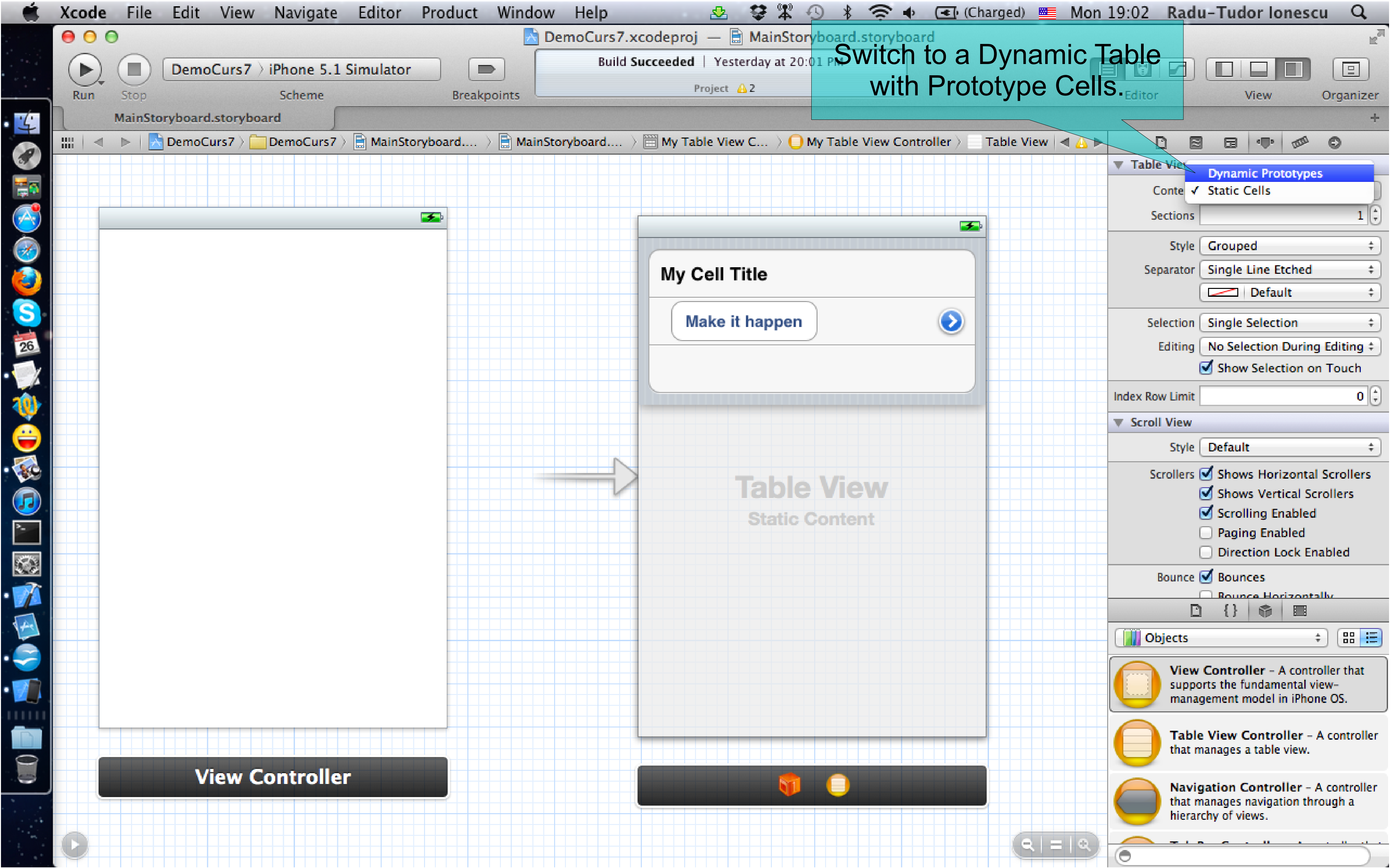
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Field – Displays editable text and

Creating Table View MVCs

All of the above examples were “static” cells (setup in the storyboard). If you switch to dynamic mode, then the cell you edit is a “prototype” for all cells in the list.



Switch to a Dynamic Table with Prototype Cells.

Table View
Content: Dynamic Prototypes / ✓ Static Cells
Sections: 1
Style: Grouped
Separator: Single Line Etched
Default
Selection: Single Selection
Editing: No Selection During Editing
☑ Show Selection on Touch
Index Row Limit: 0

Scroll View
Style: Default
Scrollers: ☑ Shows Horizontal Scrollers
☑ Shows Vertical Scrollers
☑ Scrolling Enabled
□ Paging Enabled
□ Direction Lock Enabled
Bounce: ☑ Bounces
□ Bounce Horizontally

Objects

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

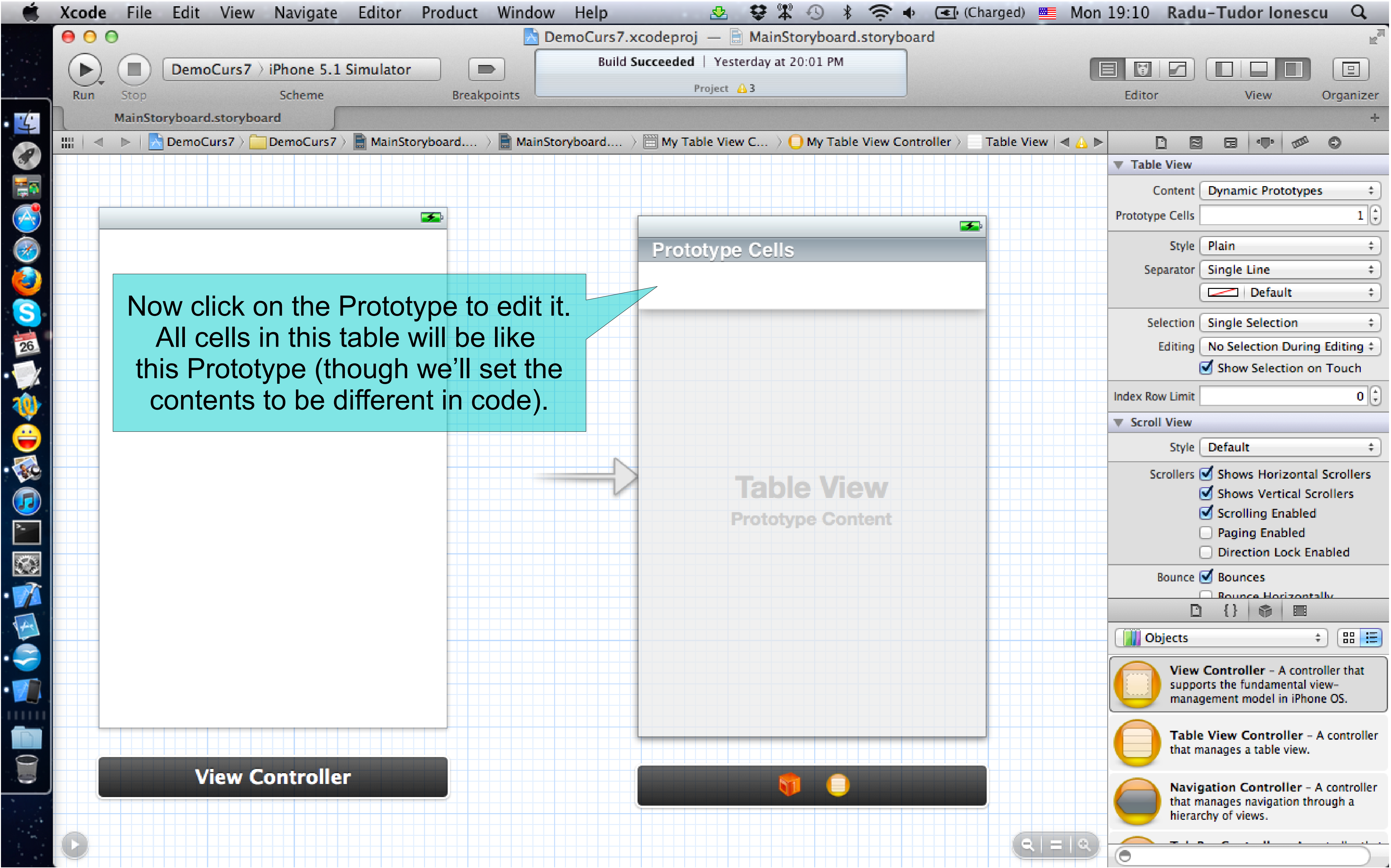
Table View Controller – A controller that manages a table view.

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

Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
(Charged)
Mon 19:02
Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 20:01 PM
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStoryboard.... › MainStoryboard.... › My Table View C... › My Table View Controller › Table View
My Cell Title
Make it happen
Table View
Static Content
View Controller

Creating Table View MVCs

All of the above examples were “static” cells (setup in the storyboard). If you switch to dynamic mode, then the cell you edit is a “prototype” for all cells in the list.



Now click on the Prototype to edit it. All cells in this table will be like this Prototype (though we’ll set the contents to be different in code).

Table View

Content: Dynamic Prototypes

Prototype Cells: 1

Style: Plain

Separator: Single Line

Default

Selection: Single Selection

Editing: No Selection During Editing

☑ Show Selection on Touch

Index Row Limit: 0

Scroll View

Style: Default

Scrollers: ☑ Shows Horizontal Scrollers

☑ Shows Vertical Scrollers

☑ Scrolling Enabled

☐ Paging Enabled

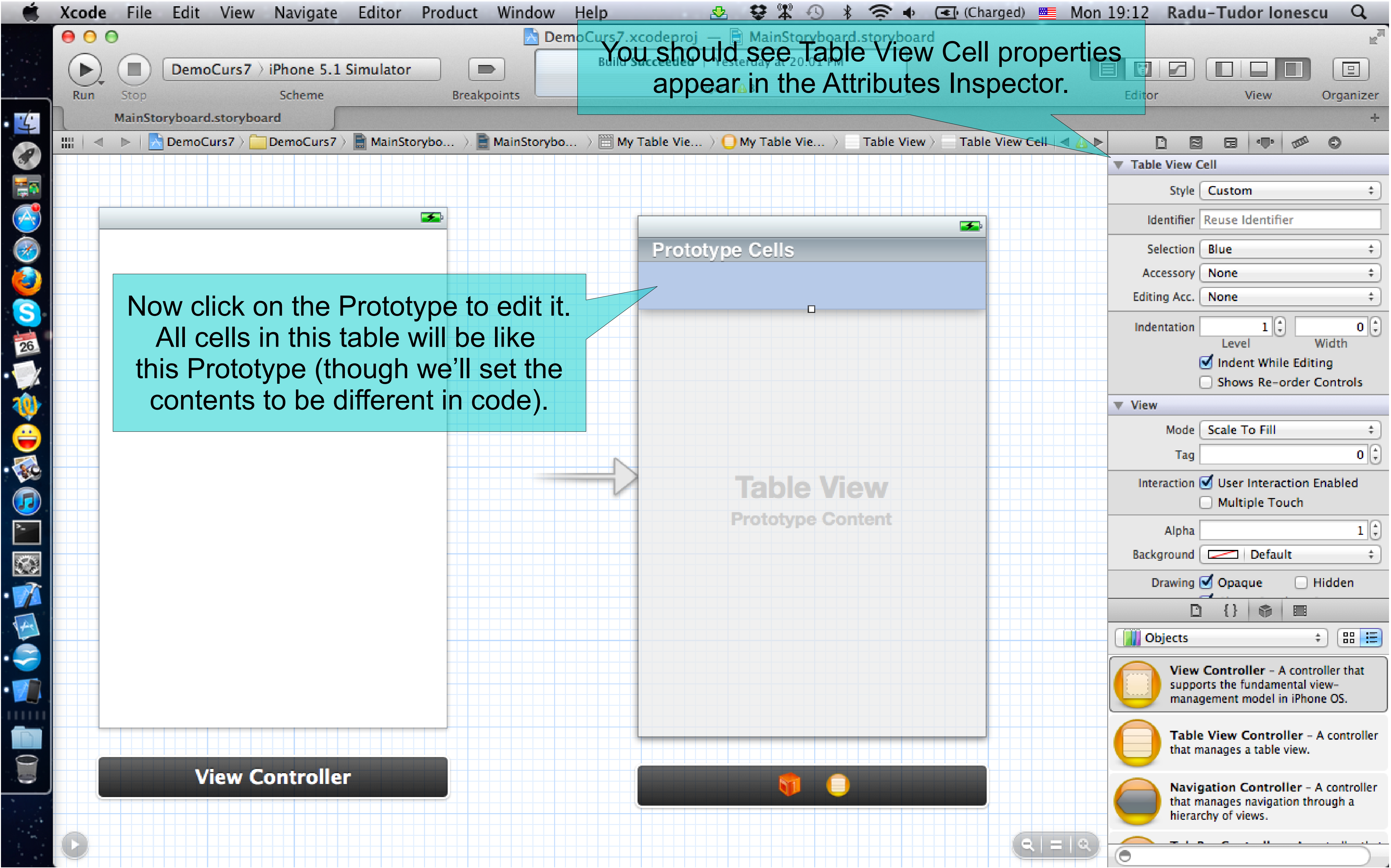
☐ Direction Lock Enabled

Bounce: ☑ Bounces

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Mon 19:10 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 20:01 PM
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStoryboard.... › MainStoryboard.... › My Table View C... › My Table View Controller › Table View
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.

Creating Table View MVCs

All of the above examples were “static” cells (setup in the storyboard). If you switch to dynamic mode, then the cell you edit is a “prototype” for all cells in the list.



You should see Table View Cell properties appear in the Attributes Inspector.

Now click on the Prototype to edit it. All cells in this table will be like this Prototype (though we’ll set the contents to be different in code).

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

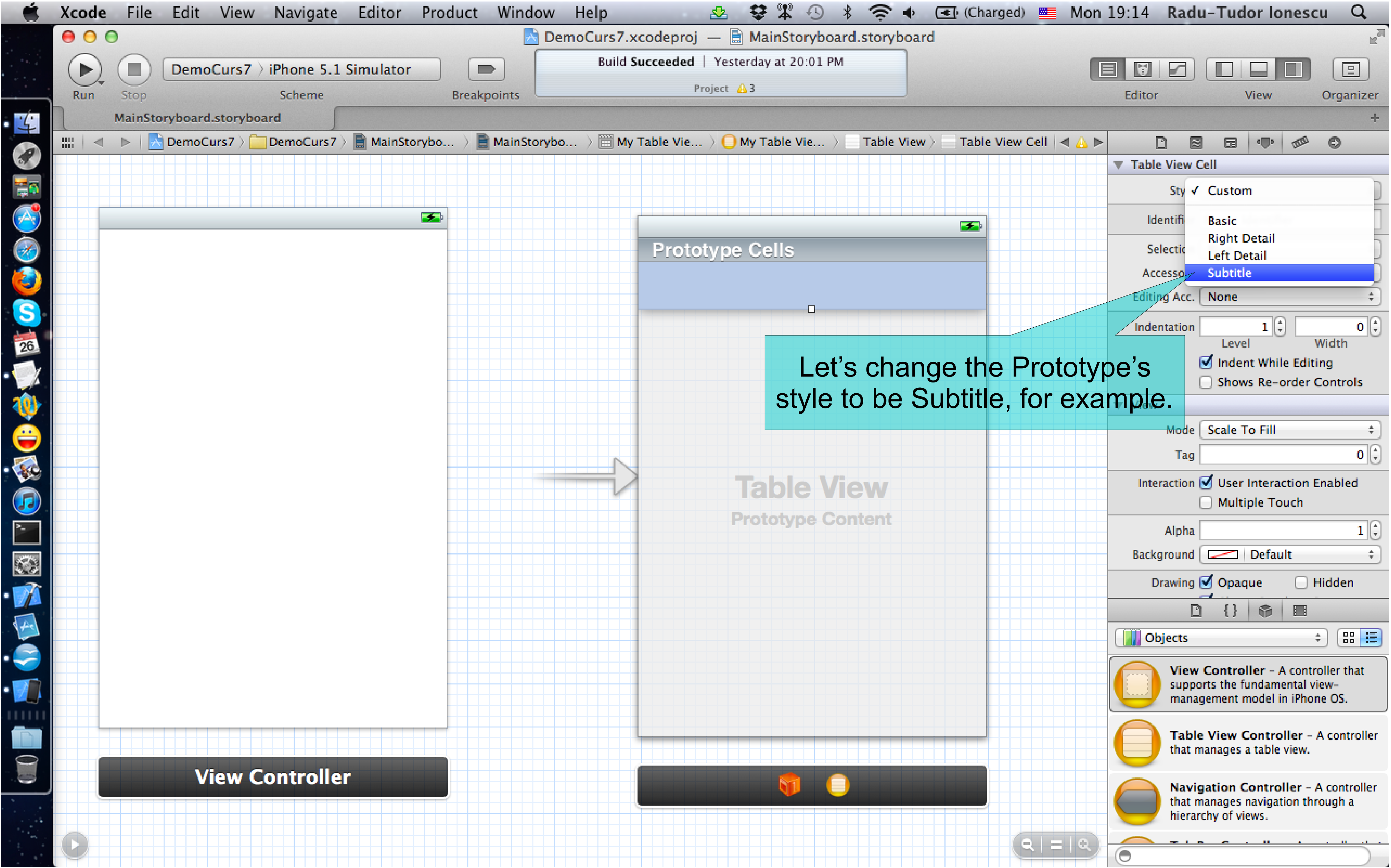
Objects

- **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)
Mon 19:12
Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 20:01 PM
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStorybo... › MainStorybo... › My Table Vie... › My Table Vie... › Table View › Table View Cell
Prototype Cells
Table View
Prototype Content
View Controller

Creating Table View MVCs

All of the above examples were “static” cells (setup in the storyboard). If you switch to dynamic mode, then the cell you edit is a “prototype” for all cells in the list.

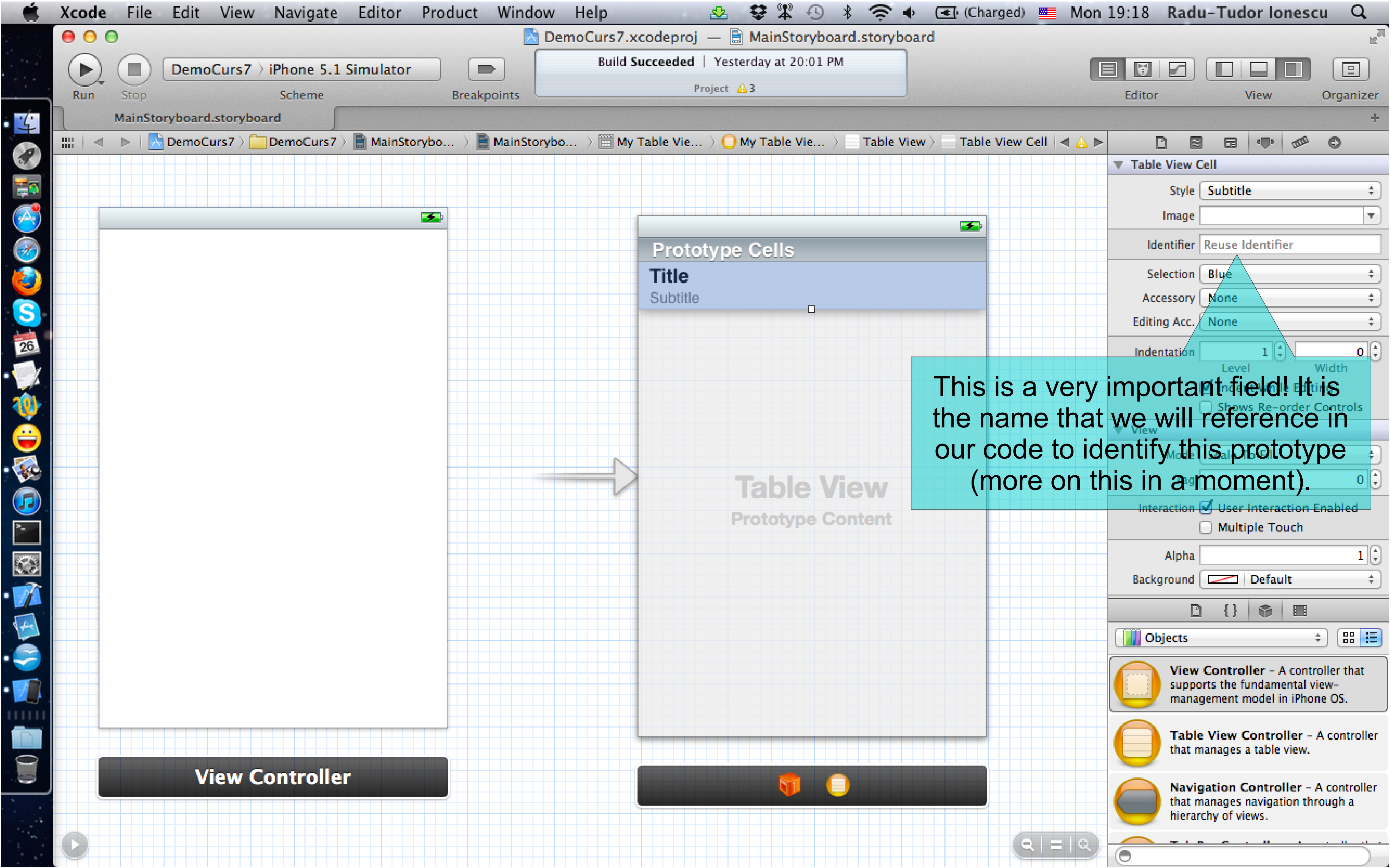


Let’s change the Prototype’s style to be Subtitle, for example.

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Mon 19:14 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Yesterday at 20:01 PM
Project ⚠3
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStorybo… › MainStorybo… › My Table Vie… › My Table Vie… › Table View › Table View Cell
Prototype Cells
Table View
Prototype Content
View Controller
Table View Cell
Style ✓ Custom
Basic
Right Detail
Left Detail
Subtitle
Identifier
Selection
Accessory
Editing Acc. None
Indentation 1 0
Level
Width
☑ Indent While Editing
☐ Shows Re-order Controls
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.

Creating Table View MVCs

All of the above examples were “static” cells (setup in the storyboard). If you switch to dynamic mode, then the cell you edit is a “prototype” for all cells in the list.



Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
(Charged)
Mon 19:18
Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoCurs7 › MainStorybo... › MainStorybo... › My Table Vie... › My Table Vie... › Table View › Table View Cell
View Controller
Prototype Cells
Title
Subtitle
Table View
Prototype Content
Table View Cell
View
Objects

Build Succeeded | Yesterday at 20:01 PM

Project ⚠ 3

Style: Subtitle

Image:

Identifier: Reuse Identifier

Selection: Blue

Accessory: None

Editing Acc.: None

Indentation: 1 (Level), 0 (Width)

☐ Shows Re-order Controls

Interaction: ☑ User Interaction Enabled

☐ Multiple Touch

Alpha: 1

Background: Default

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

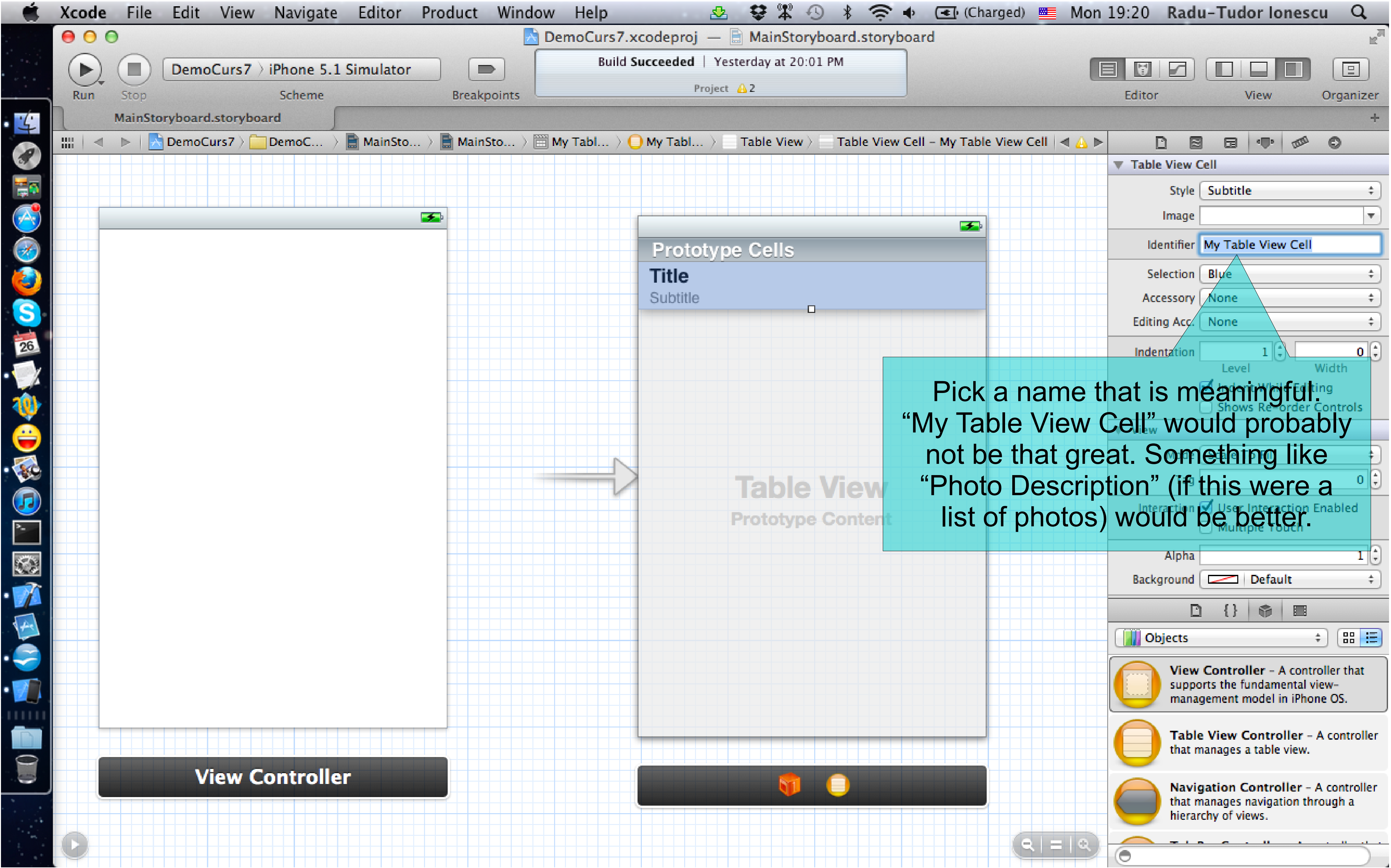
Table View Controller – A controller that manages a table view.

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

This is a very important field! It is the name that we will reference in our code to identify this prototype (more on this in a moment).

Creating Table View MVCs

All of the above examples were “static” cells (setup in the storyboard). If you switch to dynamic mode, then the cell you edit is a “prototype” for all cells in the list.



Build Succeeded | Yesterday at 20:01 PM

Project ⚠ 2

Style: Subtitle

Image:

Identifier: My Table View Cell

Selection: Blue

Accessory: None

Editing Acc.: None

Indentation: 1 Level, 0 Width

Alpha: 1

Background: Default

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.

Pick a name that is meaningful. “My Table View Cell” would probably not be that great. Something like “Photo Description” (if this were a list of photos) would be better.

UITableView Protocols

How do we connect to all this stuff in our code?

- A UITableView has two important @propertys: its delegate and its dataSource.
- The delegate is used to control how the table is displayed.
- The dataSource provides the data that is displayed inside the cells.
- Your UITableViewController is automatically set as the UITableView's delegate and dataSource.
- Your UITableViewController subclass will also have a property that points to the UITableView:

```
@property (nonatomic, strong) UITableView *tableView;
```

UITableView Protocols

- To be “dynamic”, we need to be the UITableView’s dataSource.
- Three important methods in this protocol:

  1. How many sections in the table?
  2. How many rows in each section?
  3. Give me a UIView to use to draw each cell at a given row in a given section.
- Let’s cover the last one first.

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
```

In a static table, you do not need to implement this method (though you can if you want to ignore what's in the storyboard).

```
}
```

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
           cellForRowAtIndexPath:(NSIndexPath *)indexPath
{




}
```

`NSIndexPath` is just an object with two important properties for use with `UITableView`: `row` and `section`.

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // get a cell to use (instance of UITableViewCell)









  // set @propertys on the cell to prepare it to display

}
```

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
           cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
   // get a cell to use (instance of UITableViewCell)
   UITableViewCell *cell;
   cell = [self.tableView
      dequeueReusableCellWithIdentifier:@“My Table View Cell”];
```

This MUST match what is in your storyboard if you want to use the prototype you defined there!

```
   // set @propertys on the cell to prepare it to display

}
```

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // get a cell to use (instance of UITableViewCell)
  UITableViewCell *cell;
  cell = [self.tableView
    dequeueReusableCellWithIdentifier:@“My Table View Cell”];
```

The cells in the table are actually reused. When one goes off-screen, it gets put into a “reuse pool”. The next time a cell is needed, one is grabbed from the reuse pool if available. If none is available, one will be put into the reuse pool if there’s a prototype in the storyboard. Otherwise this dequeue method will return `nil` (let’s deal with that next).

```
  // set @propertys on the cell to prepare it to display

}
```

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // get a cell to use (instance of UITableViewCell)
  UITableViewCell *cell;
  cell = [self.tableView
    dequeueReusableCellWithIdentifier:@“My Table View Cell”];

  if (!cell)
  {
    cell = [[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleSubtitle
          reuseIdentifier:@“My Table View Cell”];
  }

  // set @propertys on the cell to prepare it to display

}
```

UITableViewDataSource

How do we control what is drawn in each cell in a dynamic table?

- Each row is drawn by its own instance of `UITableViewCell`.
- Here is the `UITableViewDataSource` method to get that cell for a given row in a given section.

```
- (UITableViewCell *)tableView:(UITableView *)sender
           cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // get a cell to use (instance of UITableViewCell)
  UITableViewCell *cell;
  cell = [self.tableView
    dequeueReusableCellWithIdentifier:@“My Table View Cell”];

  if (!cell)
  {
    cell = [[UITableViewCell alloc]
                initWithStyle:UITableViewCellStyleSubtitle
              reuseIdentifier:@“My Table View Cell”];
  }
  cell.textLabel.text = [self dataForRow:indexPath.row
                               inSection:indexPath.section];
  return cell;
}
```

There are obviously other things you can do in the cell besides setting its text (detail text, image, accessory, etc).

UITableViewDataSource

How does a dynamic table know how many rows there are?

- And how many sections, too, of course?

```
- (NSInteger)numberOfSectionsInTableView:
                                  (UITableView *)sender;

- (NSInteger)tableView:(UITableView *)sender
 numberOfRowsInSection:(NSInteger)section;
```

- Number of sections is 1 by default. In other words, if you don't implement `numberOfSectionsInTableView:`, it will be 1.
- No default for number of rows in a section.
- This is a required method in this protocol (as is `tableView:cellForRowAtIndexPath:`).

What about a static table?

- Do not implement these `dataSource` methods for a static table.
- `UITableViewController` will take care of that for you.

UITableViewDataSource

There are a number of other methods in this protocol

- But we’re not going to cover all of them.
- They are mostly about getting the headers and footers for sections.
- And about dealing with editing the table (moving/deleting/inserting rows).

UITableViewDataSource

There are a number of other methods in this protocol

- Let us continue with our demo and see, for example, how can we delete rows from a Table View.
- We implement the following method to return YES to allow editing.

```
- (BOOL)tableView:(UITableView *)tableView
  canEditRowAtIndexPath:(NSIndexPath *)indexPath
{
    /* Return NO if you do not want
     * the specified item to be editable. */
    return YES;
}
```

UITableViewDataSource

There are a number of other methods in this protocol

- Let us continue with our demo and see, for example, how can we delete rows from a Table View.
- We delete the row from the Table View and also from the Model by implementing this method:

```
-  (void)tableView:(UITableView *)tableView
commitEditingStyle:(UITableViewCellEditingStyle)editingStyle
 forRowAtIndexPath:(NSIndexPath *)indexPath
{
 if (editingStyle == UITableViewCellEditingStyleDelete)
 {
  [self.tableData removeObjectAtIndex:indexPath.row];

  NSArray *indexPaths = [NSArray arrayWithObject:indexPath]
  [tableView deleteRowsAtIndexPaths:indexPaths
                   withRowAnimation:UITableViewRowAnimationFade];
 }
}
```

UITableViewDelegate

- All of the above was the UITableView’s dataSource.

  But UITableView has another protocol-driven delegate called its delegate.

- The delegate controls how the UITableView is displayed.

  Not what it displays (that’s the dataSource’s job).

- It is common for dataSource and delegate to be the same object.

  Usually the Controller of the MVC in which the UITableView is part of the (or is the entire) View.

- The delegate also lets you observe what the table view is doing.

  Especially responding to when the user selects a row.

  We often will use segues when this happens, but we can also track it directly.

Table View “Target/Action”

`UITableViewDelegate` method sent when row is selected

- This is sort of like table view “target/action”.
- You might use this to update a detail view in a split view if master is a table view

```
- (void)tableView:(UITableView *)sender
didSelectRowAtIndexPath:(NSIndexPath *)path
{
  /* go do something based on information about my
   * data structure corresponding to indexPath.row
   * in indexPath.section */
}
```

Table View “Target/Action”

Lots and lots of other delegate methods

- `will`/`did` methods for both selecting and deselecting rows.
- Providing `UIView` objects to draw section headers and footers.
- Handling editing rows (moving them around with touch gestures).
- `willBegin`/`didEnd` notifications for editing.
- Copying/pasting rows.

Table View Segues

You can segue when a row is touched, just like from a button. Segues will call `prepareForSegue:sender:` with the chosen `UITableViewCell` as sender.

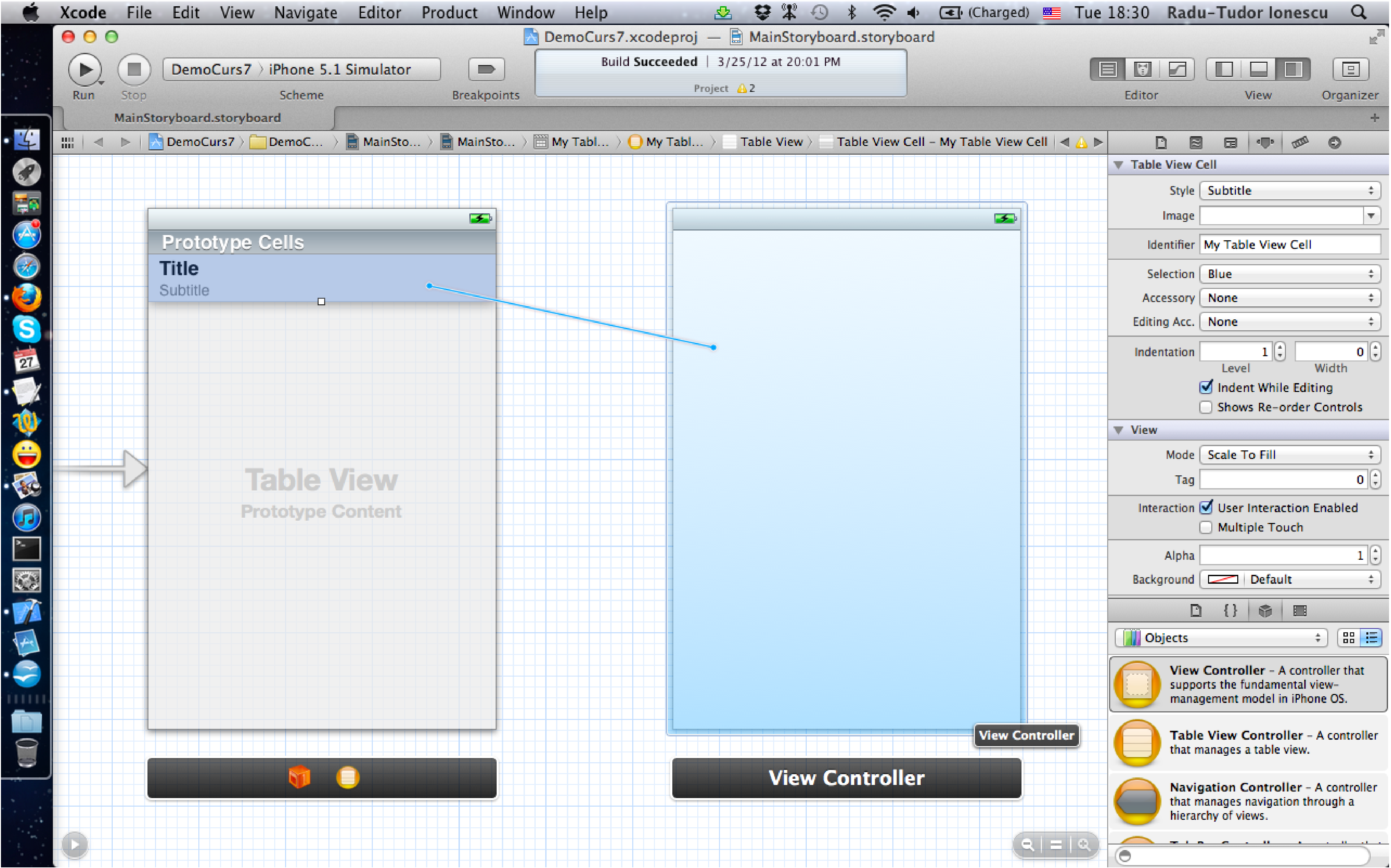

Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
(Charged)
Tue 18:30
Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | 3/25/12 at 20:01 PM
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC… › MainSto… › MainSto… › My Tabl… › My Tabl… › Table View › Table View Cell – My Table View Cell
Prototype Cells
Title
Subtitle
Table View
Prototype Content
View Controller
View Controller
Objects

Table View Cell

- Style: Subtitle
- Image:
- Identifier: My Table View Cell
- 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

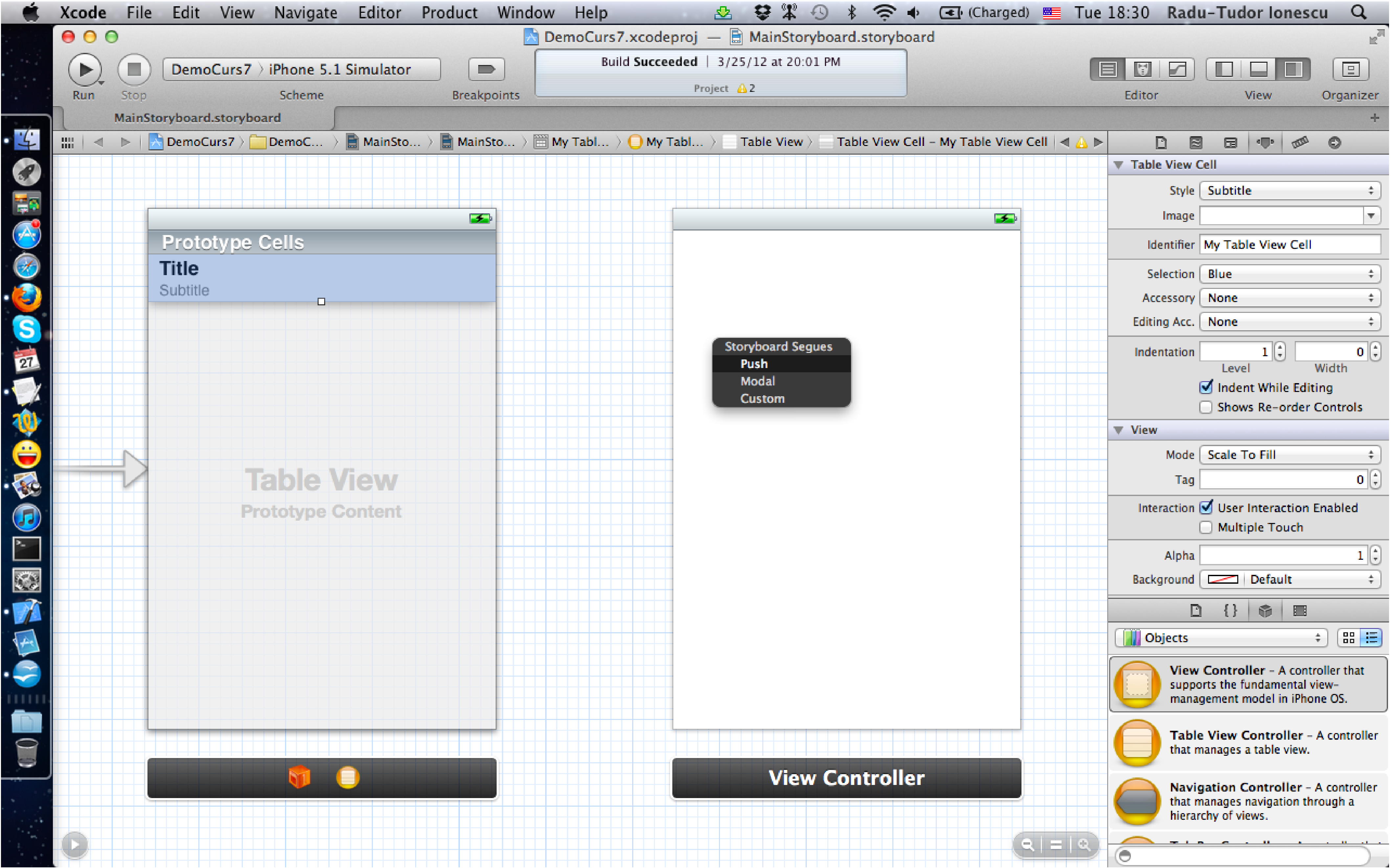
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Table View Segues

You can segue when a row is touched, just like from a button. Segues will call `prepareForSegue:sender:` with the chosen `UITableViewCell` as sender.



Build Succeeded | 3/25/12 at 20:01 PM

Table View Cell

Style: Subtitle

Image:

Identifier: My Table View Cell

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

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

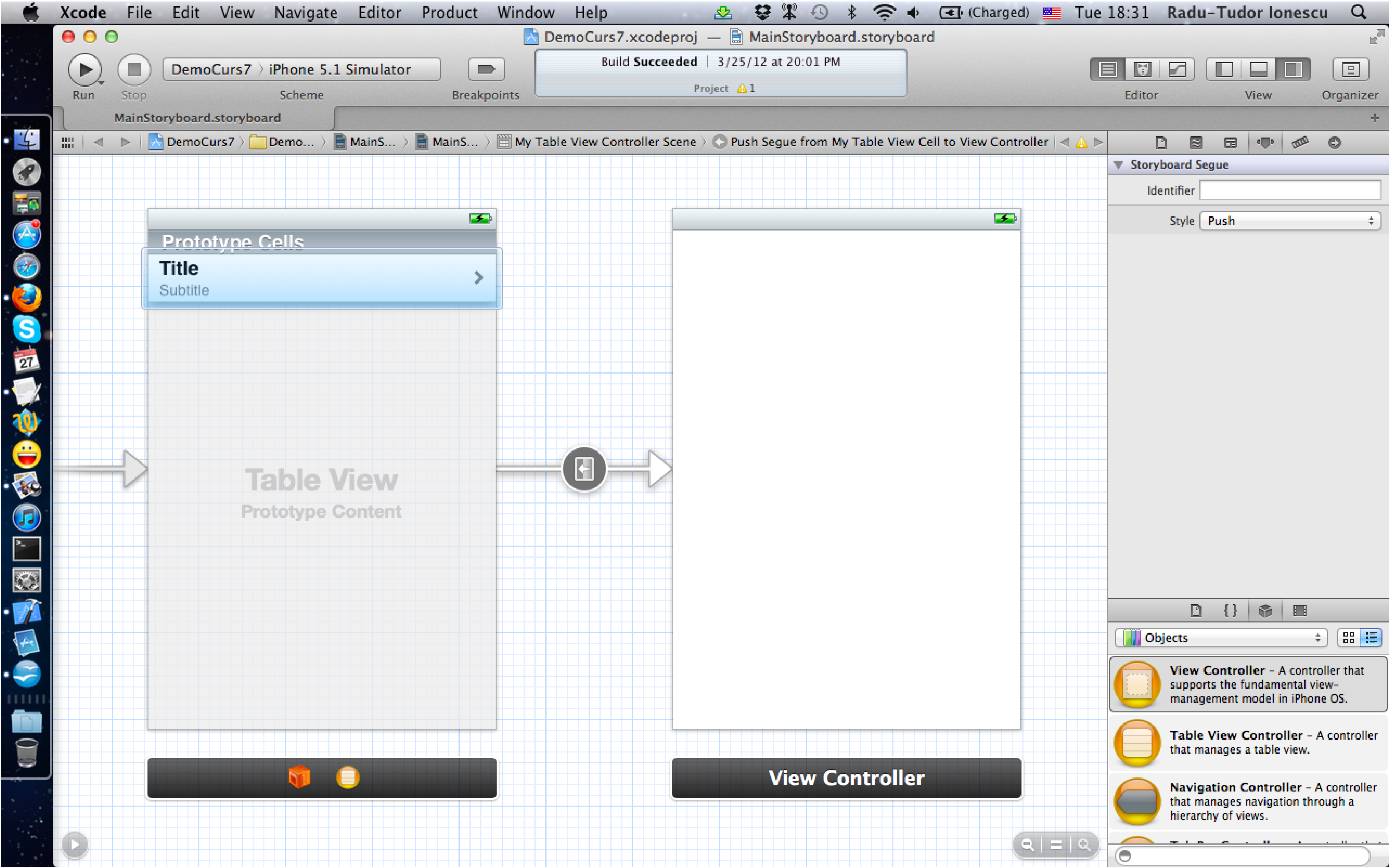
Table View Controller – A controller that manages a table view.

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

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 18:30 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › DemoC... › MainSto... › MainSto... › My Tabl... › My Tabl... › Table View › Table View Cell – My Table View Cell
Prototype Cells
Title
Subtitle
Table View
Prototype Content
Storyboard Segues
Push
Modal
Custom
View Controller
Objects

Table View Segues

You can segue when a row is touched, just like from a button. Segues will call `prepareForSegue:sender:` with the chosen `UITableViewCell` as sender.



Xcode File Edit View Navigate Editor Product Window Help
(Charged) Tue 18:31 Radu-Tudor Ionescu
DemoCurs7.xcodeproj — MainStoryboard.storyboard
Run
Stop
DemoCurs7 › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | 3/25/12 at 20:01 PM
Project ⚠1
Editor
View
Organizer
MainStoryboard.storyboard
DemoCurs7 › Demo… › MainS… › MainS… › My Table View Controller Scene › Push Segue from My Table View Cell to View Controller
Prototype Cells
Title
Subtitle
Table View
Prototype Content
View Controller
Objects

Storyboard Segue

Identifier:

Style: Push

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 Segues

- You can tailor whatever data the MVC needs to whichever cell was selected.
- This works whether dynamic or static.

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue
                             sender:(id)sender
{
    NSIndexPath *indexPath = [self.tableView
                           indexPathForCell:sender];

/* prepare segue.destinationController to display
 * based on information about my data structure
 * corresponding to indexPath.row
 * in indexPath.section */
}
```

UITableView

What if your Model changes?

- You can:

```
- (void)reloadData;
```

- Causes the table view to call `numberOfSectionsInTableView:` and `numberOfRowsInSection:` all over again and then `cellForRowAtIndexPath:` on each visible cell.
- Relatively heavy weight obviously, but if your entire data structure changes, that's what you need.
- If only part of your Model changes, there are lighter-weight reloaders, for example:

```
- (void)reloadRowsAtIndexPaths:(NSArray *)indexPaths
            withRowAnimation:
          (UITableViewRowAnimation)animationStyle;
```

UITableView

There are dozens of other methods in `UITableView`

- Setting headers and footers for the entire table.
- Controlling the look (separator style and color, default row height, etc).
- Getting cell information (cell for index path, index path for cell, visible cells, etc). Scrolling to a row.
- Selection management (allows multiple selection, getting the selected row, etc).
- Moving, inserting and deleting rows, etc.

Next Time

View Controller Lifecycle and UIKit:

- View Controller Lifecycle
- Image View
- Web View
- Scroll View