








Assignment 4

Mobile Media

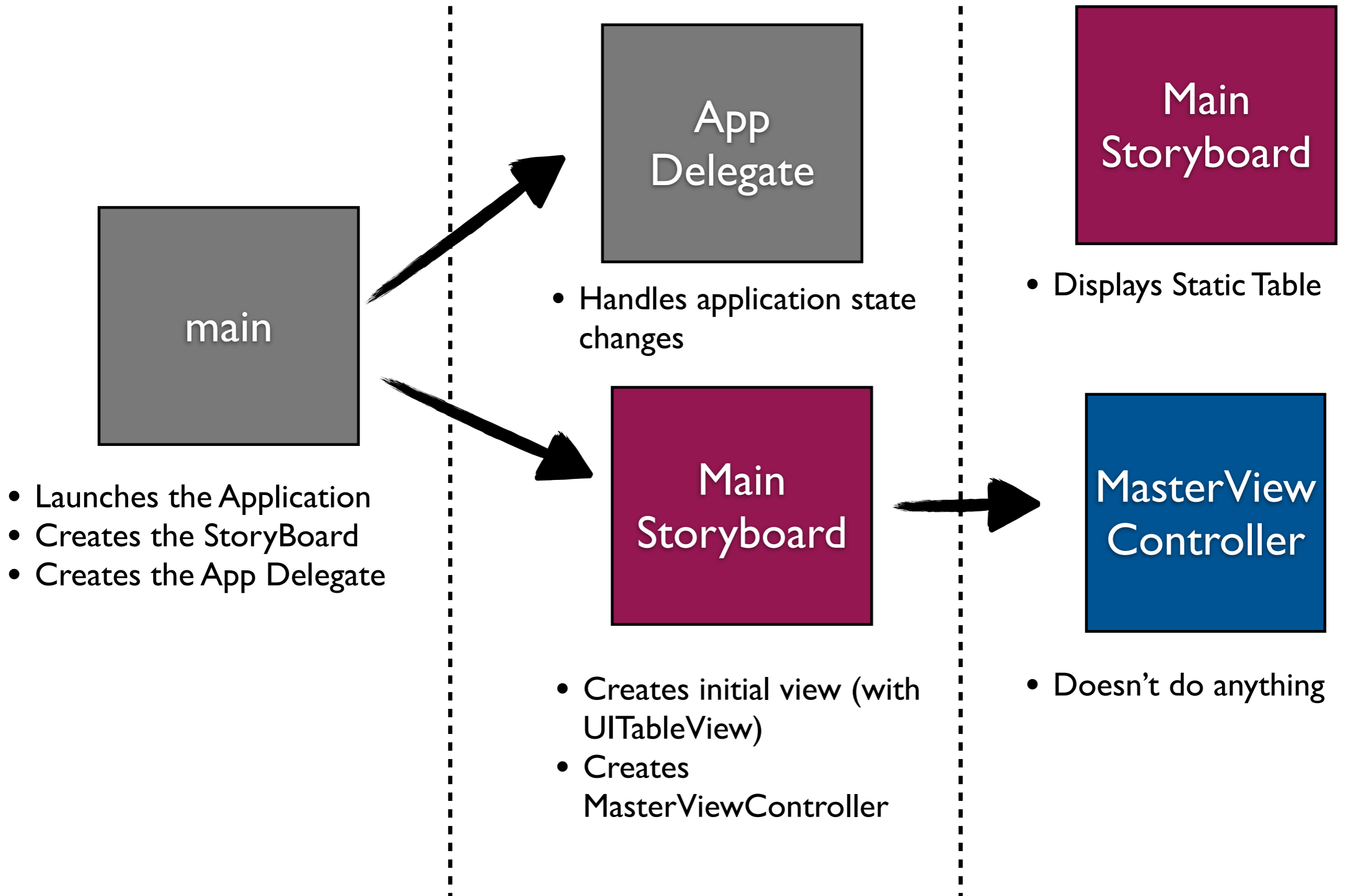
Copying has been disabled for this document.
Please check your code carefully as you type it in manually.

Where we are so far

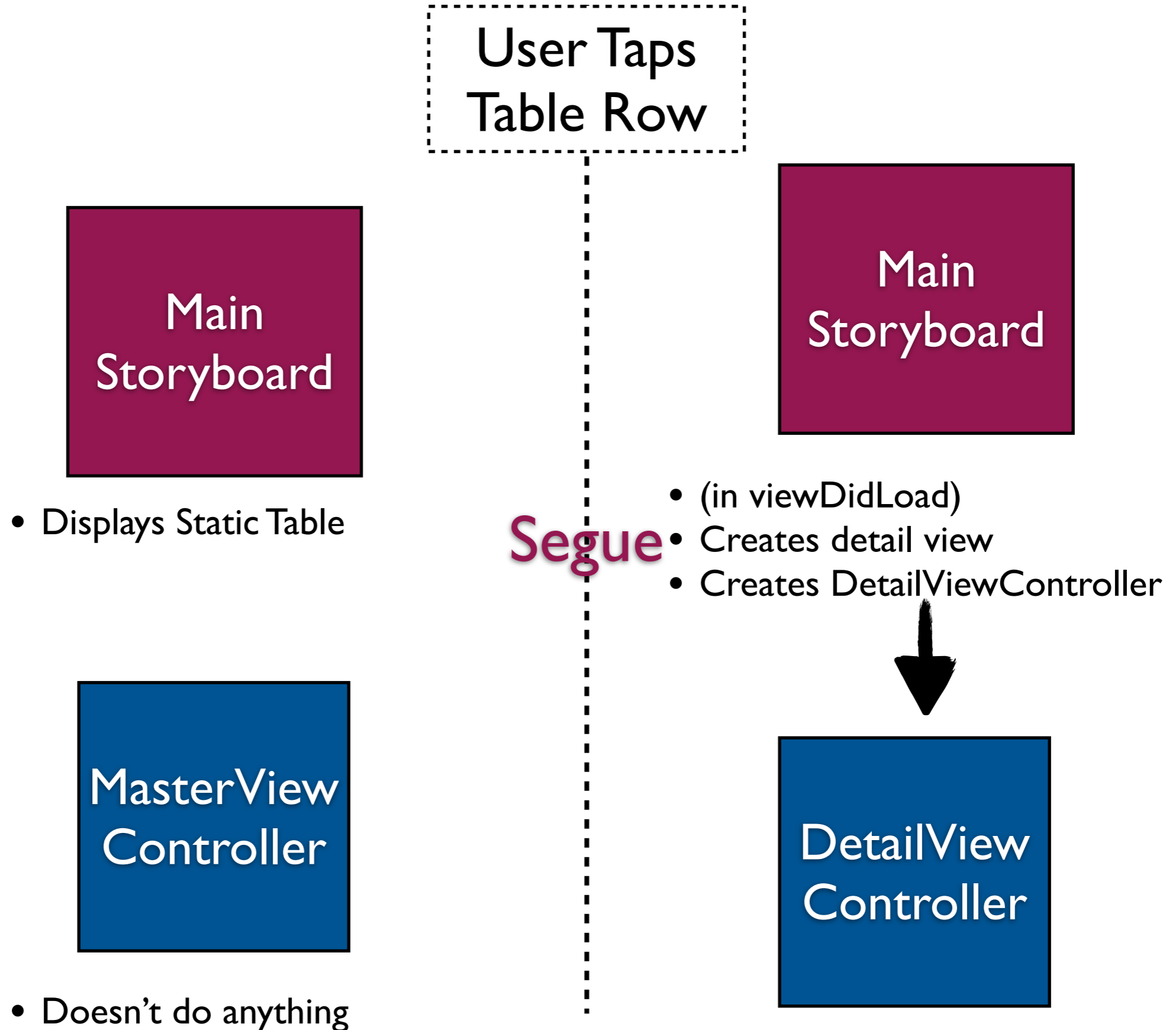
6 Classes, 1 Storyboard

2 Model	2 Controller	2 Program Execution	1 Storyboard
 <p>Restaurant</p>  <p>Review</p>	 <p>MasterView Controller</p>  <p>DetailView Controller</p>	 <p>App Delegate</p>  <p>main</p>	 <p>Main Storyboard</p>

Timeline (Part I)



Timeline (Part 2)



Timeline (Part 3)



- (in viewDidLoad)
- Creates a Restaurant object
- Assigns Restaurant to variable restaurant
- Sets restaurant properties
- Creates 5 Review objects
- Sets properties on each review
- Creates an array of Reviews
- Sets restaurant's reviews property to the array
- Asks the Restaurant for the Most Helpful Review



- (in mostHelpfulReview)
- Runs through the reviews array
- Asks each review for its helpfulPercentage
- Determines which review is the most helpful
- Returns the most helpful review to the DetailViewController

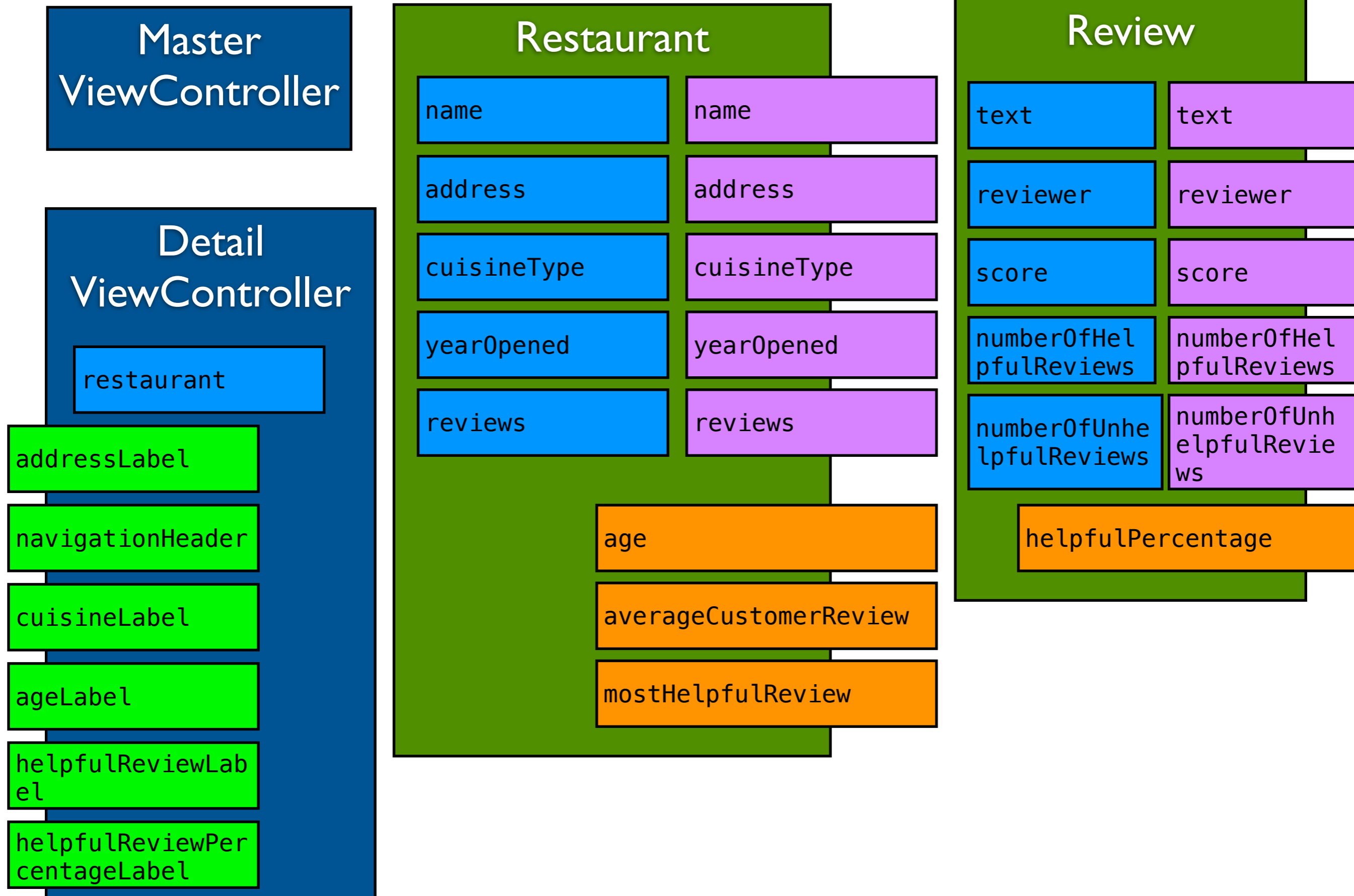


Timeline (Part 4)



- (in viewDidLoad)
- Sets the text of each of its labels to the Restaurant's properties
- Composes a new string out of the most helpful review
- Sets the string to the text of the review label

Instance Variables, Properties, IBOutlets, and Methods



What's next?

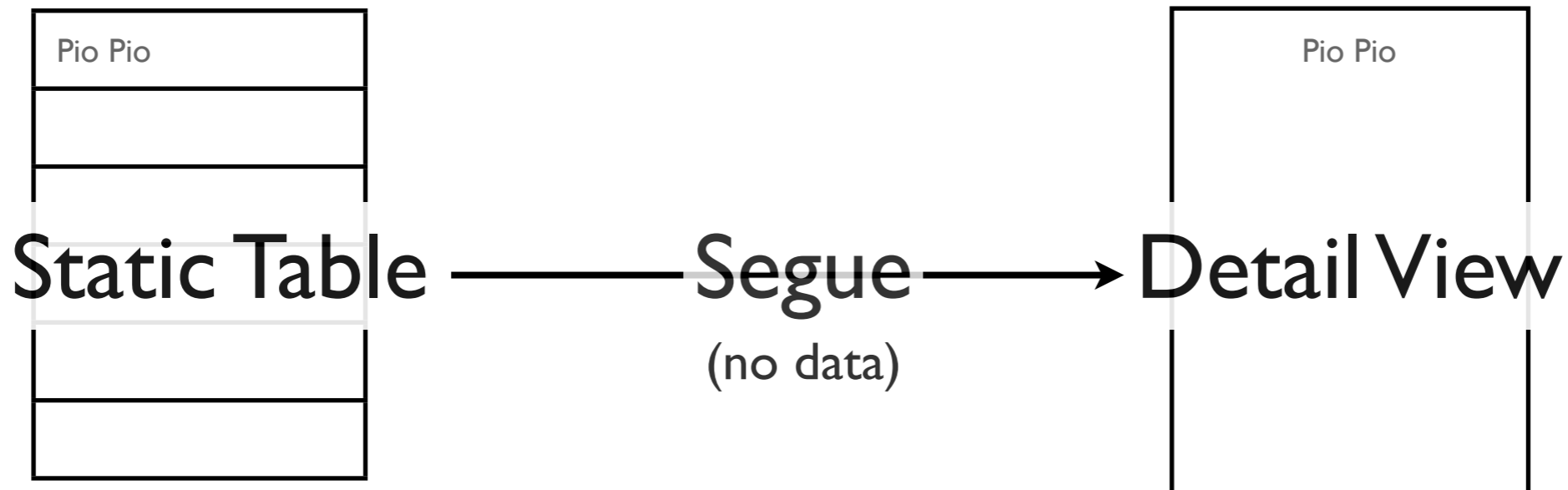
A working UITableView, multiple restaurants

MasterViewController

DetailViewController

Creates a restaurant & reviews

Shows details about restaurant on view

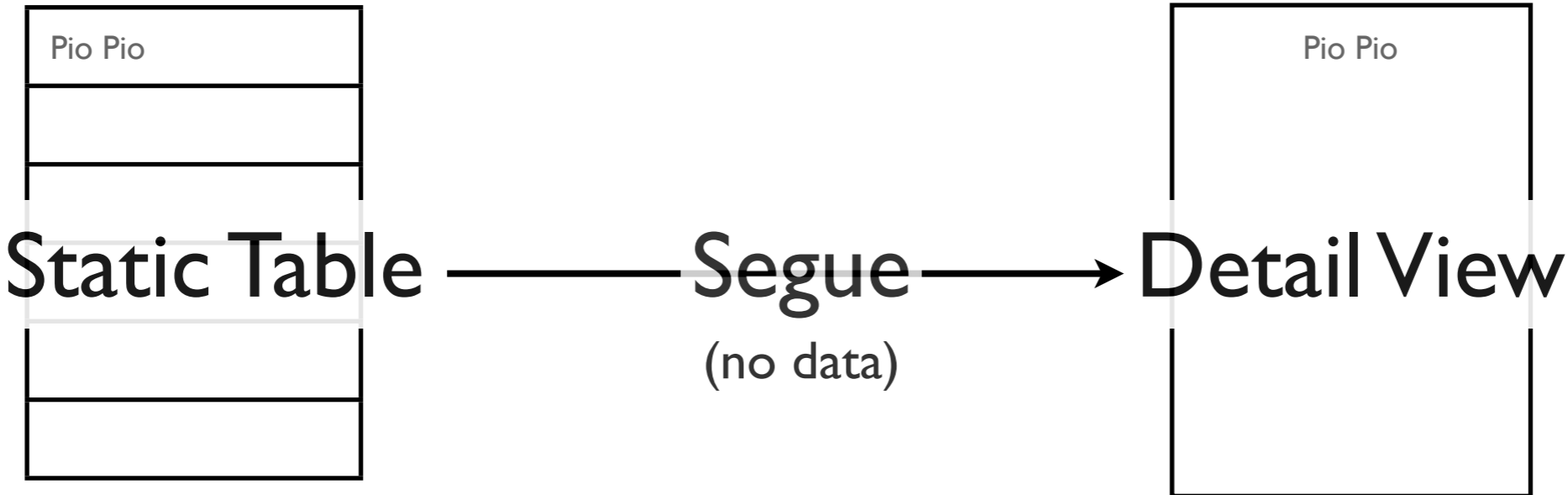


What's next?

A working UITableView, multiple restaurants

MasterViewController
Creates a restaurant & reviews

DetailViewController
Shows details about restaurant on view



What's next?

A working UITableView, multiple restaurants

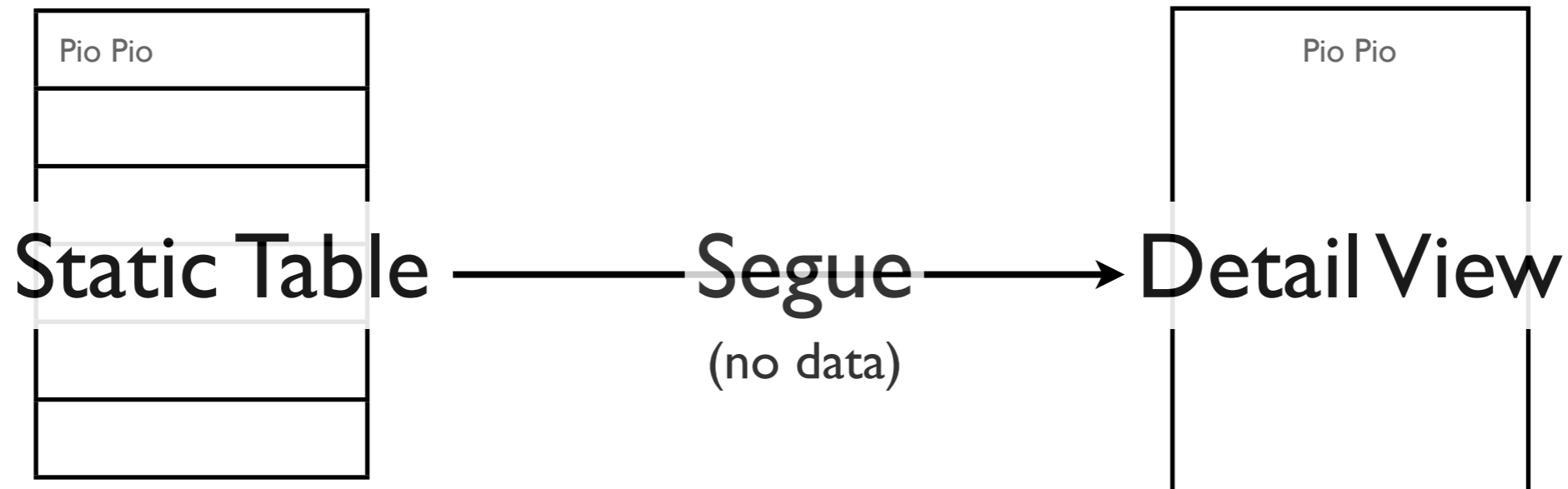
MasterViewController

Creates a restaurant & reviews

Populates table with restaurants

DetailViewController

Shows details about restaurant on view



What's next?

A working UITableView, multiple restaurants

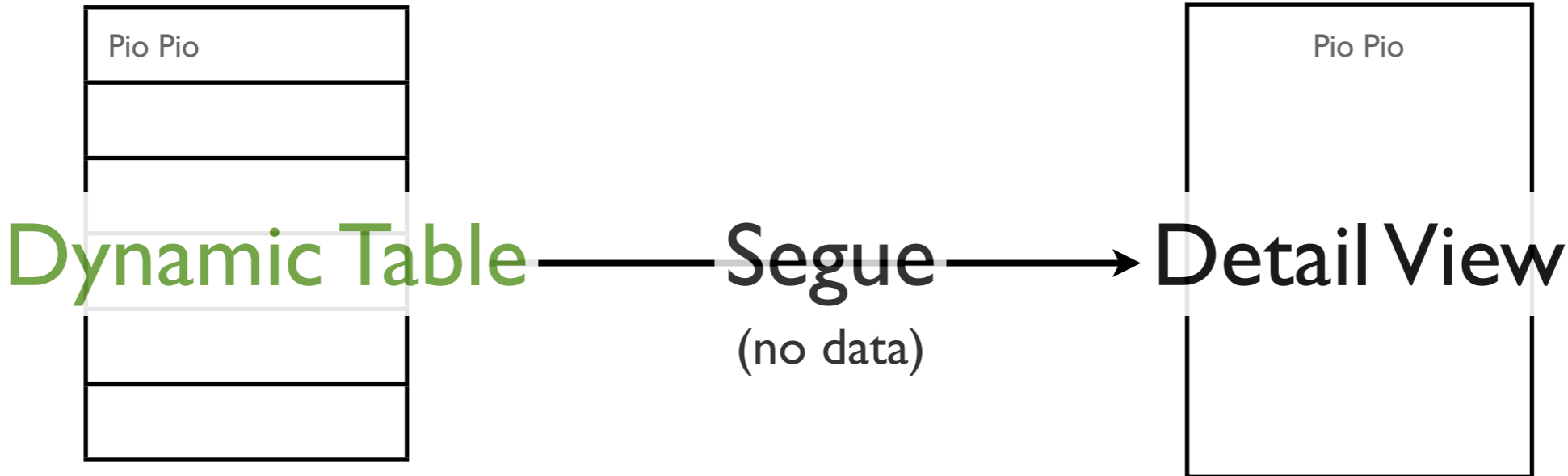
MasterViewController

Creates a restaurant & reviews

Populates table with restaurants

DetailViewController

Shows details about restaurant on view



What's next?

A working UITableView, multiple restaurants

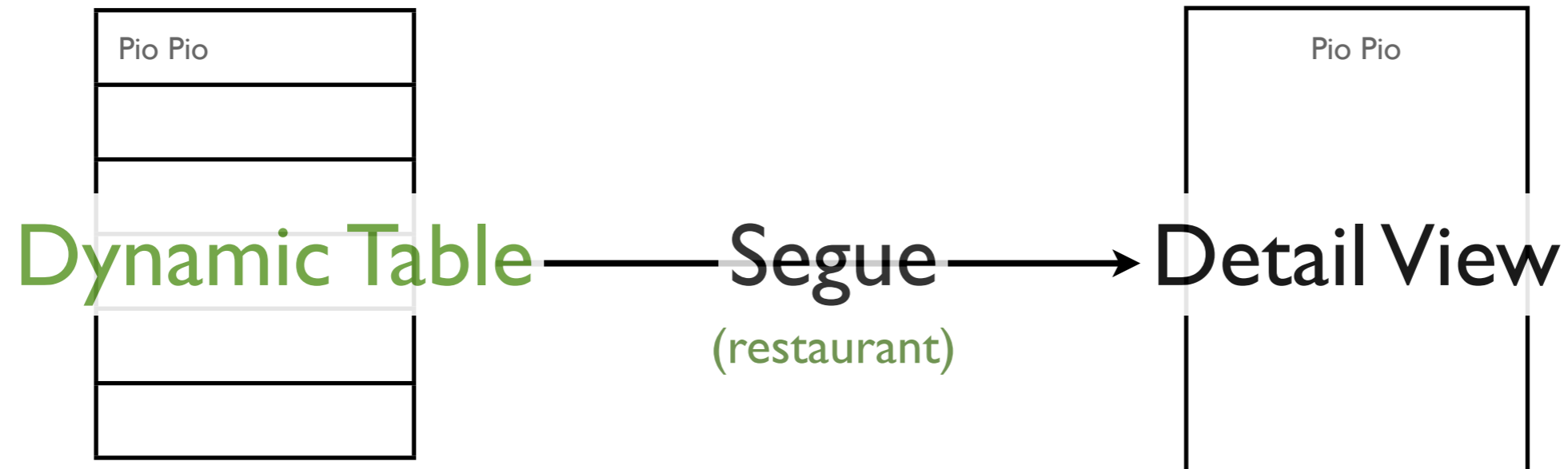
MasterViewController

Creates a restaurant & reviews

Populates table with restaurants

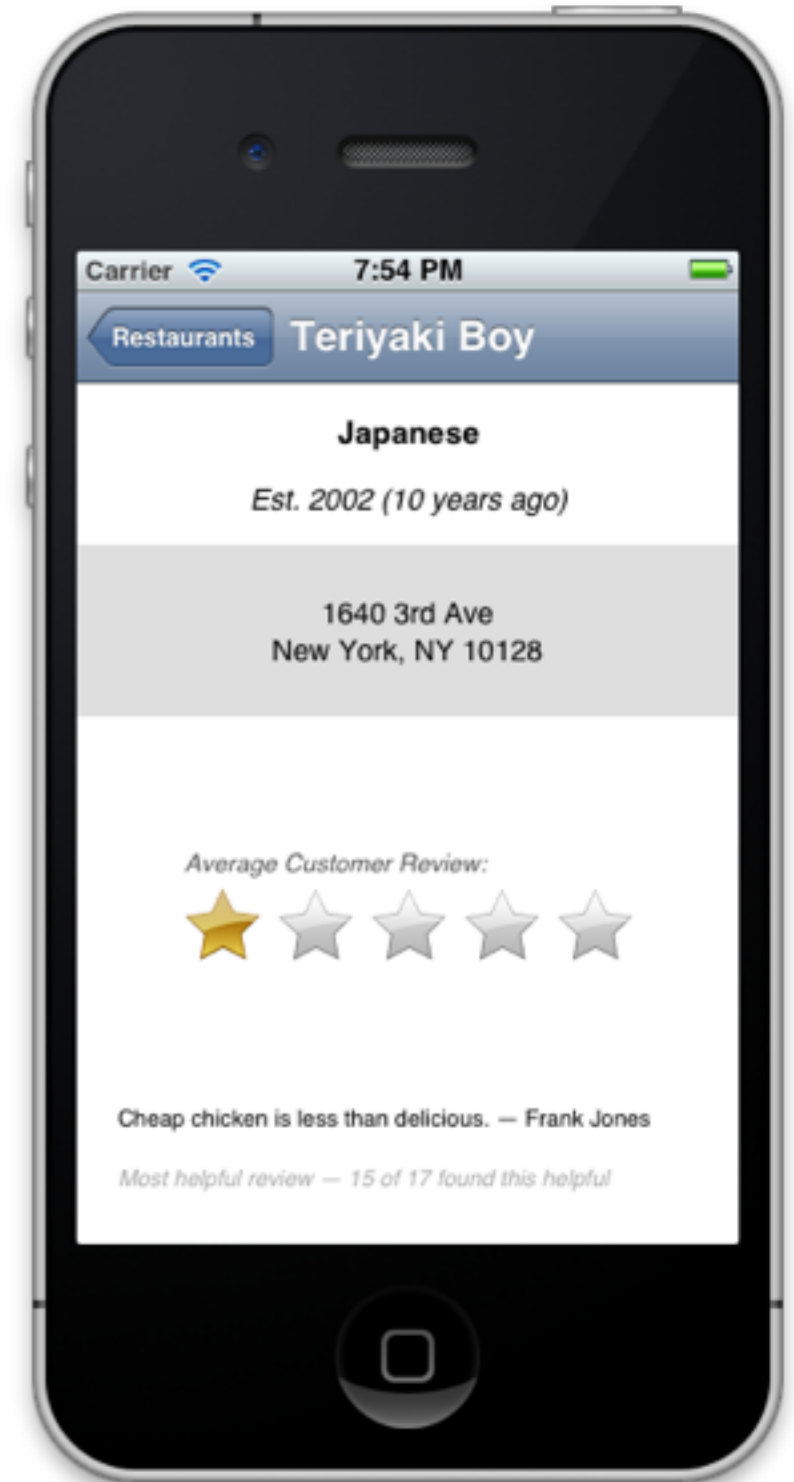
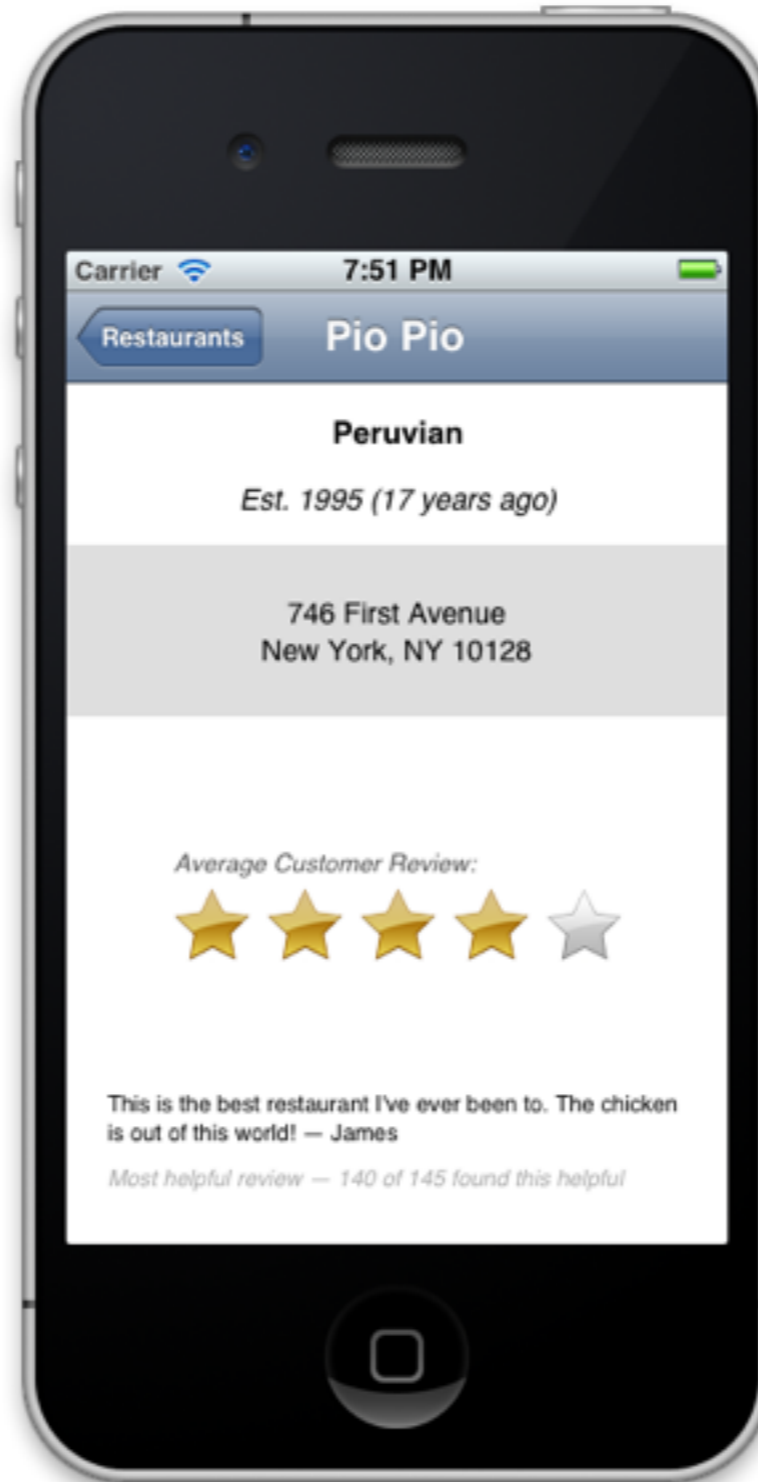
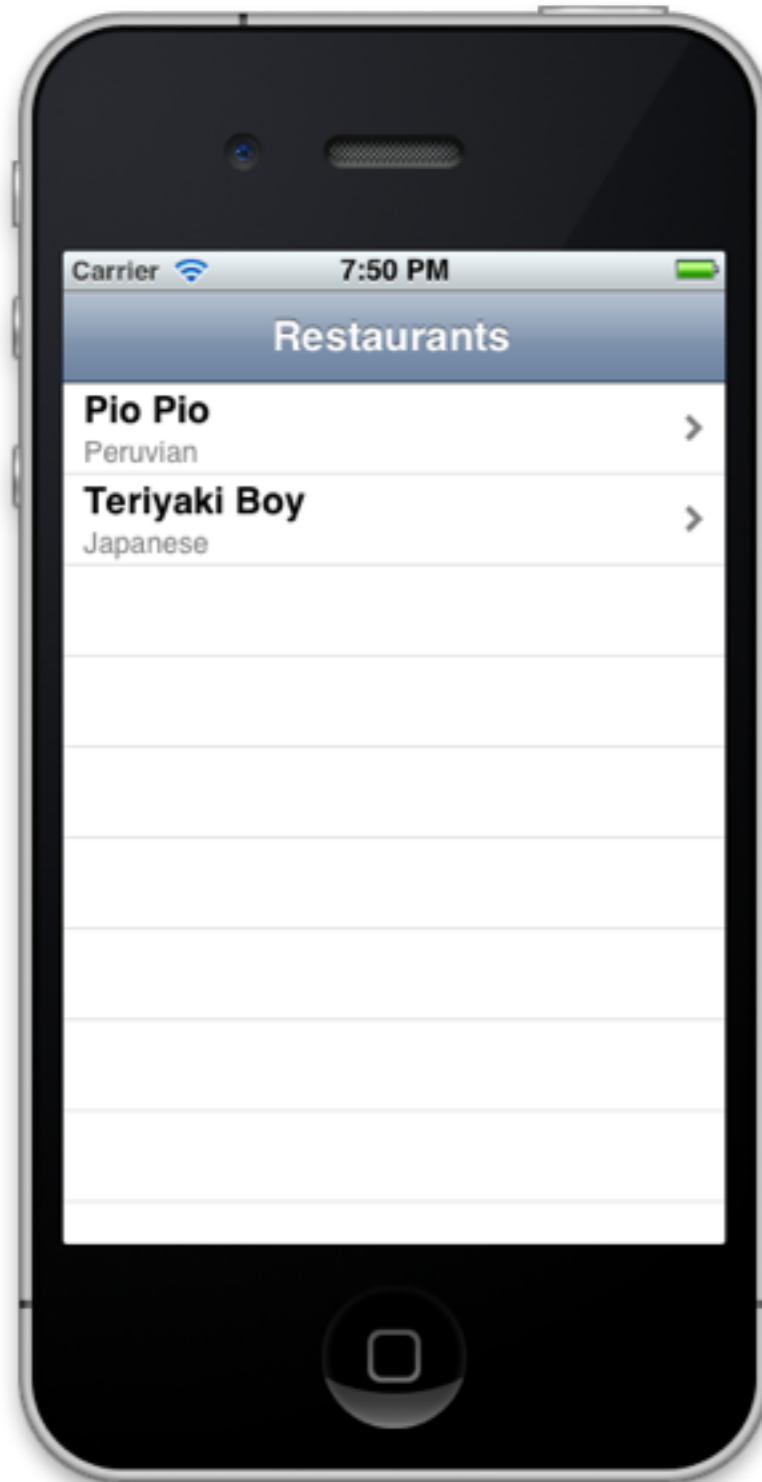
DetailViewController

Shows details about restaurant on view



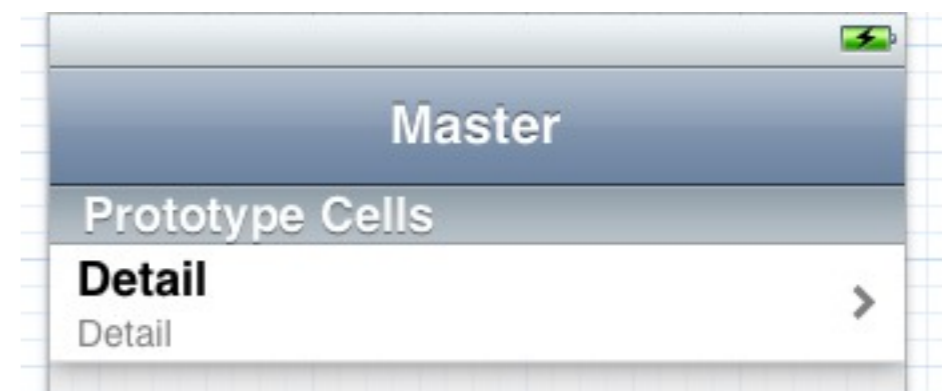
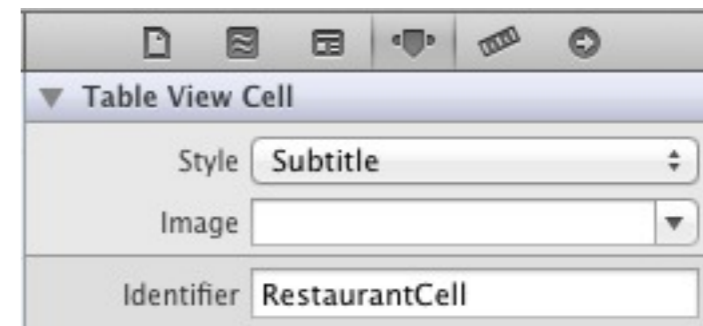
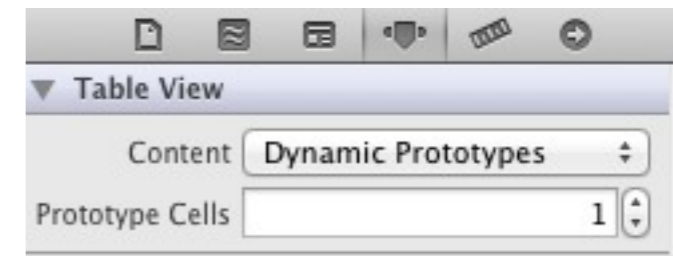
Finished Product

Table View, 2 restaurants



Task 1: Making the Table Dynamic

- We need to make the table dynamic to be able to populate it with any number of restaurants.
- In your storyboard, select your table view.
- Change the table type to “Dynamic Prototypes”
- Next, choose the Table Cell, and change its **Style** to “Subtitle”
- Finally, change the **Identifier** of the cell to “RestaurantCell”
- Your table should now look like the image on the right



How (Dynamic) Table Views Work

- UITableViews work via *Delegation*
- Rather than **telling** a table what to show, the table **asks** the controller what to show

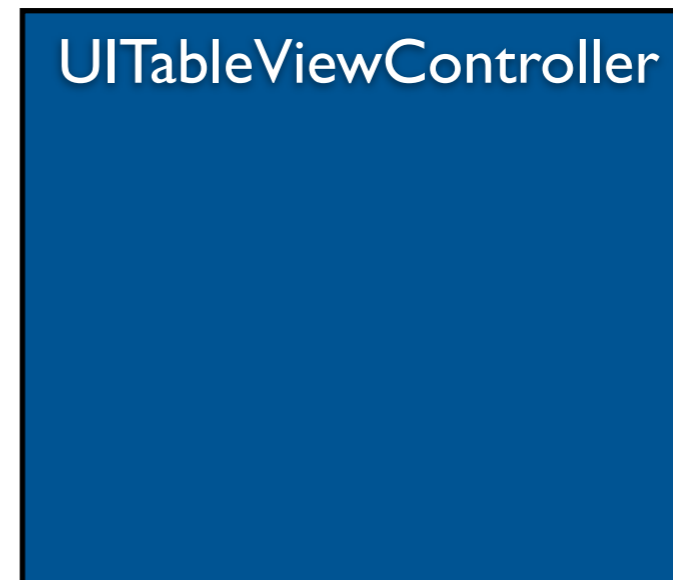
UITableView



How many cells should I
show in section 0?



UITableViewController

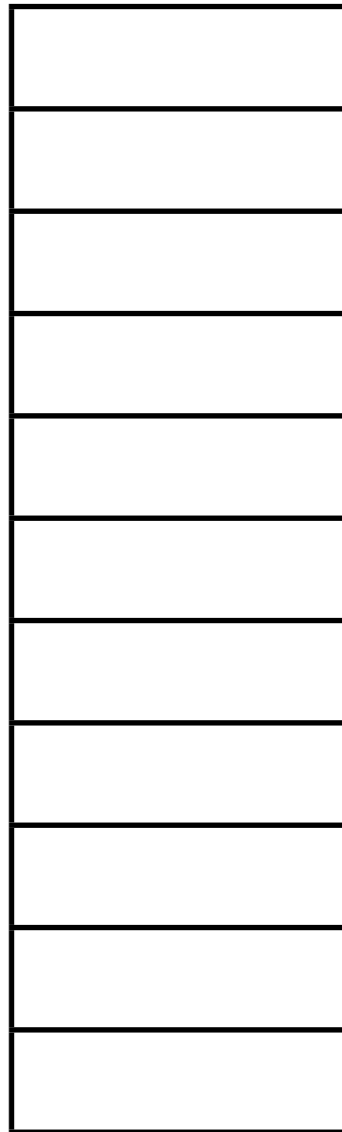


1 cell.



How (Dynamic) Table Views Work

UITableView



What cell should I show for section 0, row 0?



Hold on, I need a cell called "RestaurantCell".



Here you go, here's a cell called "RestaurantCell"



Ok, I've set the values on this cell. Show it please.



UITableViewController

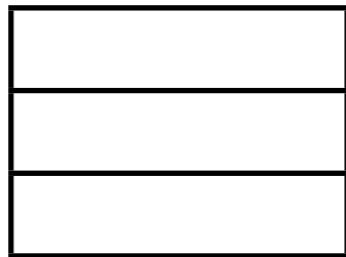


How (Dynamic) Table Views Work

- In our code, the Storyboard is already connected to our MasterViewController via 2 outlets: **dataSource** and **delegate**.
- These are both *Protocols*, which means they agree to provide certain methods to the view.
- MasterViewController is a **subclass** of UITableViewController, which conforms to these two protocols.
- In this case, we will be overriding 2 methods critical to making the table display anything.

How many cells should I show in section 0?

UITableView



MasterViewController
(UITableViewController)

```
-(int)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section  
{  
    return 1;  
}
```

```
-(int)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section  
{  
    return 1;  
}
```

What cell should I show for section 0, row 0?

UITableView



MasterViewController
(UITableViewController)

```
- (int) tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section  
{  
    return 1;  
}
```

```
- (UITableViewCell*) tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath  
{  
    NSString* cellIdentifier = @"RestaurantCell";  
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellIdentifier];  
    cell.textLabel.text = @"Pio Pio";  
    cell.detailTextLabel.text = @"Peruvian";  
    return cell;  
}
```

What cell should I show for section 0, row 0?

```
- (UITableViewCell*)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath
*)indexPath
{
    //This is the cell identifier we set in the Storyboard file
    NSString* cellIdentifier = @"RestaurantCell";

    //This line asks the tableView to "reuse" a cell. It will return a UITableViewCell with
    the identifier given.
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellIdentifier];

    //The cell type we selected in the storyboard supports a textLabel and a detailTextLabel.
    We'll set them both here.
    cell.textLabel.text = @"Pio Pio";
    cell.detailTextLabel.text = @"Peruvian";
    return cell;
}
```

Task 2

- Ensure your Assignment 3 runs
 - If you are having problems with assignment 3 still, please call me over, or ask your neighbor
- Implement in MasterViewController:

```
-(int)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
```

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

- Implement these as shown on the previous 2 slides.
- Build and run. You should have 1 cell, with “Pio Pio” and “Peruvian” in it.

Task 3:

Move the creation of the restaurant

- Now, we're going to move the creation of the Restaurant from the DetailViewController
- **Cut** all of your code from viewDidLoad in DetailViewController that deals with creating restaurants and reviews.
 - Do not cut where you set label text on your view.
- Paste this code inside of viewDidLoad in the MasterViewController.
- Add `#import "Restaurant.h"` to the top of MasterViewController.h
- You'll also need to **declare** restaurant as a local variable in the viewDidLoad method. It won't be a property of MasterViewController the same way it was a property of DetailViewController. **(Why not?)**
- Next, **Rename** the restaurant variable to "piopio" or whatever your restaurant is named. Remember, no spaces, and it should start with a lowercase letter.
- Build (Command-B). Ensure that you have no errors.

MasterViewController viewDidLoad: (what mine looks like)

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    Restaurant* piopio = [[Restaurant alloc] init];

    piopio.name = @"Pio Pio";
    piopio.address = @"746 First Avenue\nNew York, NY 10128";
    piopio.cuisineType = @"Peruvian";
    piopio.yearOpened = 1995;

    Review* pioReview1 = [[Review alloc] init];
    pioReview1.text = @"What fab-u-las chicken! We could eat it all day if we didn't have to
stop to drink sangria!";
    pioReview1.reviewer = @"The Addams";
    pioReview1.score = 5;
    pioReview1.numberOfHelpfulReviews = 19;
    pioReview1.numberOfUnhelpfulReviews = 8;
```

(continues, adds additional reviews)

Task 4

- Add a “restaurants” NSMutableArray* property to the MasterViewController
- Don't forget: Variable, @property, and @synthesize.
- alloc/init this in your viewDidLoad
- Add your restaurant to it.

Task 5

- Modify MasterViewController to return the actual number of restaurants in the array for the table.

```
-(int)tableView:(UITableView *)tableView numberOfRowsInSection:  
(NSInteger)section
```

How to populate a table with an Array

UITableView with one section

NSArray with 2 items

Restaurants	
Pio Pio Peruvian >	
Teriyaki Boy Japanese >	

IndexPath 0,0

IndexPath 0,1

Restaurant: Pio Pio
Restaurant: Teriyaki Boy

- A new object, NSIndexPath is a simple object that holds a **section** and a **row**
- Since we'll only be using one section in this example, we'll just have each cell of the UITableView's section 0 correspond with an item in the array.
- When the UITableView asks for its cell, we'll populate the values from the corresponding item in the NSArray.

Task 6

Modify MasterViewController

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

to find the Restaurant, and change the text of the `textLabel` and `detailTextLabel` to be the name and `cuisineType` from the restaurant.

MasterViewController Reference

```
- (UITableViewCell*)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    //This is the cell identifier we set in the Storyboard file
    NSString* cellIdentifier = @"RestaurantCell";

    //This line asks the tableView to "reuse" a cell. It will return a UITableViewCell with
the identifier given.
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellIdentifier];

    // We can get at the proper restaurant by asking for the object in the array at the
indexPath's row
    Restaurant* currentRestaurant = [restaurants objectAtIndex:indexPath.row];

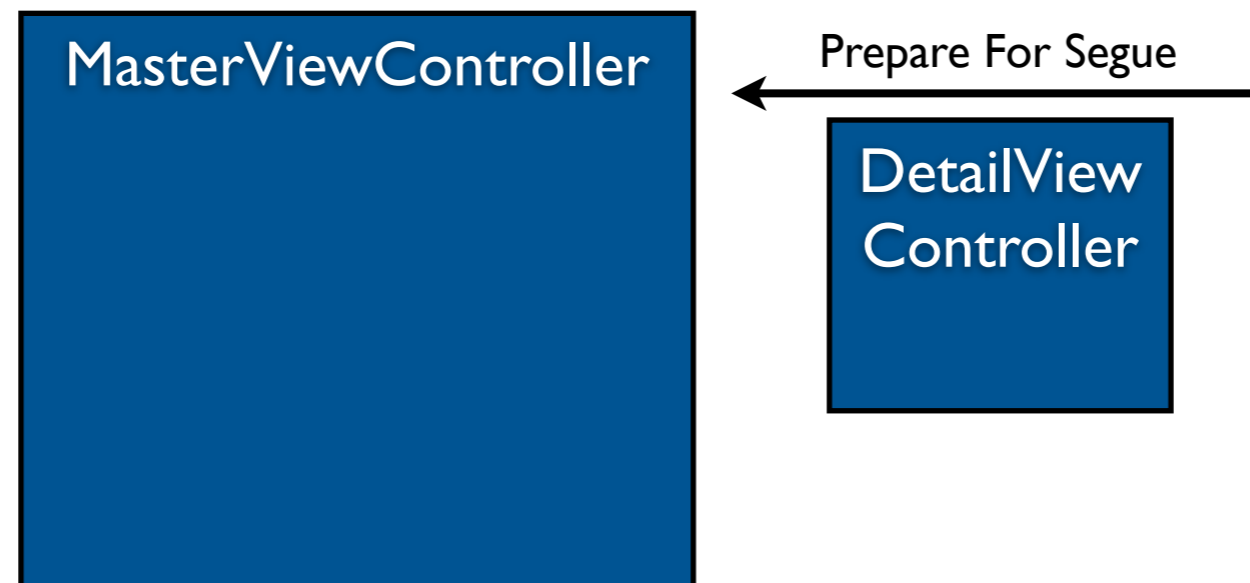
    //The cell type we selected in the storyboard supports a textLabel and a detailTextLabel.
We'll set them both here.
    cell.textLabel.text = currentRestaurant.name;
    cell.detailTextLabel.text = currentRestaurant.cuisineType;
    return cell;
}
```

Passing the Restaurant to the DetailView

- In the storyboard, we transition between views using a segue
- We need to pass the restaurant along with the segue
- Once the restaurant is set, your previous code for displaying a restaurant will work

Another Delegate

- When the segue transitions, it calls the initial view controller and passes the new view controller to the old one



- This allows the initial view controller to set properties on the new one

Prepare DetailViewController

- We need to access restaurant on DetailViewController, so we need to make it a property.

- Add the `@property` and `@synthesize` for restaurant

- And, since we'll be directly using DetailViewController from MasterViewController:

- Add `#import "DetailViewController.h"` to the top of MasterViewController.h

Task 7

Modify MasterViewController

```
-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
```

get the destination view controller

```
DetailViewController* detailVC = (DetailViewController*)  
[segue destinationViewController];
```

get the table view

```
UITableView* table = [self tableView];
```

get the selected row's index path from the table

```
NSIndexPath* indexPath = [table indexPathForSelectedRow];
```

get the current restaurant using the index path

```
Restaurant* currentRestaurant = [restaurants objectAtIndex:indexPath.row];
```

set the detail view controller's restaurant

```
detailVC.restaurant = currentRestaurant;
```

Test!

- Your detail view should now be populated with the restaurant created in the master view controller.

- Build, Run, Commit and Sync.
- Create a second restaurant in MasterViewController and add it to the restaurants array
- Run again, and see if your second restaurant appears