Developing Applications for iOS

Lab 8: Nearby Deals (4 of 6)

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

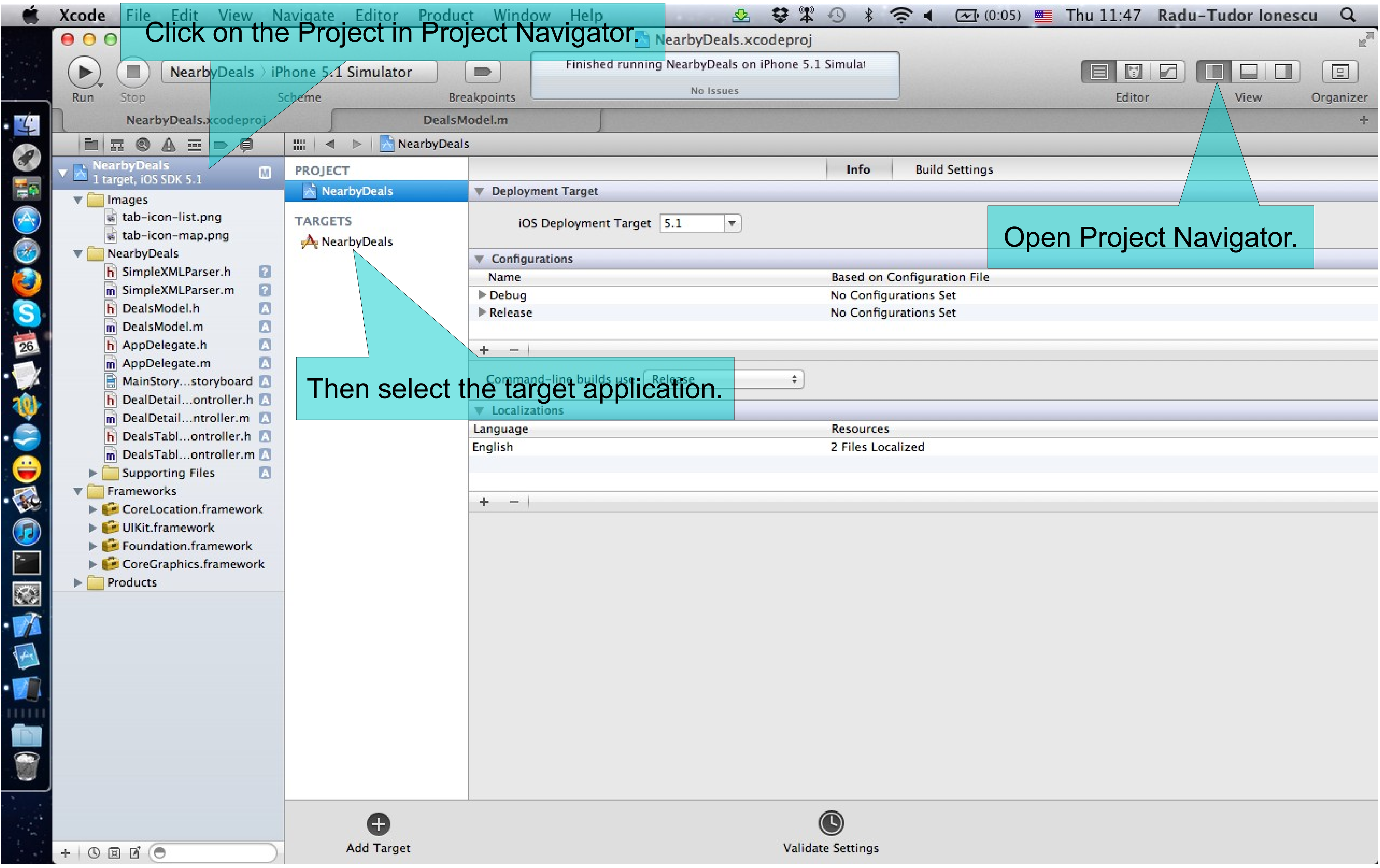
Task 1

Task: Add Locations Services as a required device capability.

1. Launch Xcode and go to “File > Open” and select the Xcode project (.xcodeproj) inside the “NearbyDeals(3of6)” folder.
2. Run the application in iOS Simulator and take a look over the application to remember what was done last time.
3. Stop running the application.
4. We will use Locations Services (through the Core Location framework that is already included in our Project) to determine the device location. The device location is needed when the application requests nearby deals from the GeoAds+ server.

Our application will not function (as the user expects) if the device location is not available. In other words, the device location is a required device capability for our application.

Follow the steps from the next slides to understand how to declare a required device capability.



Click on the Project in Project Navigator.

Open Project Navigator.

Then select the target application.

Deployment Target

iOS Deployment Target 5.1

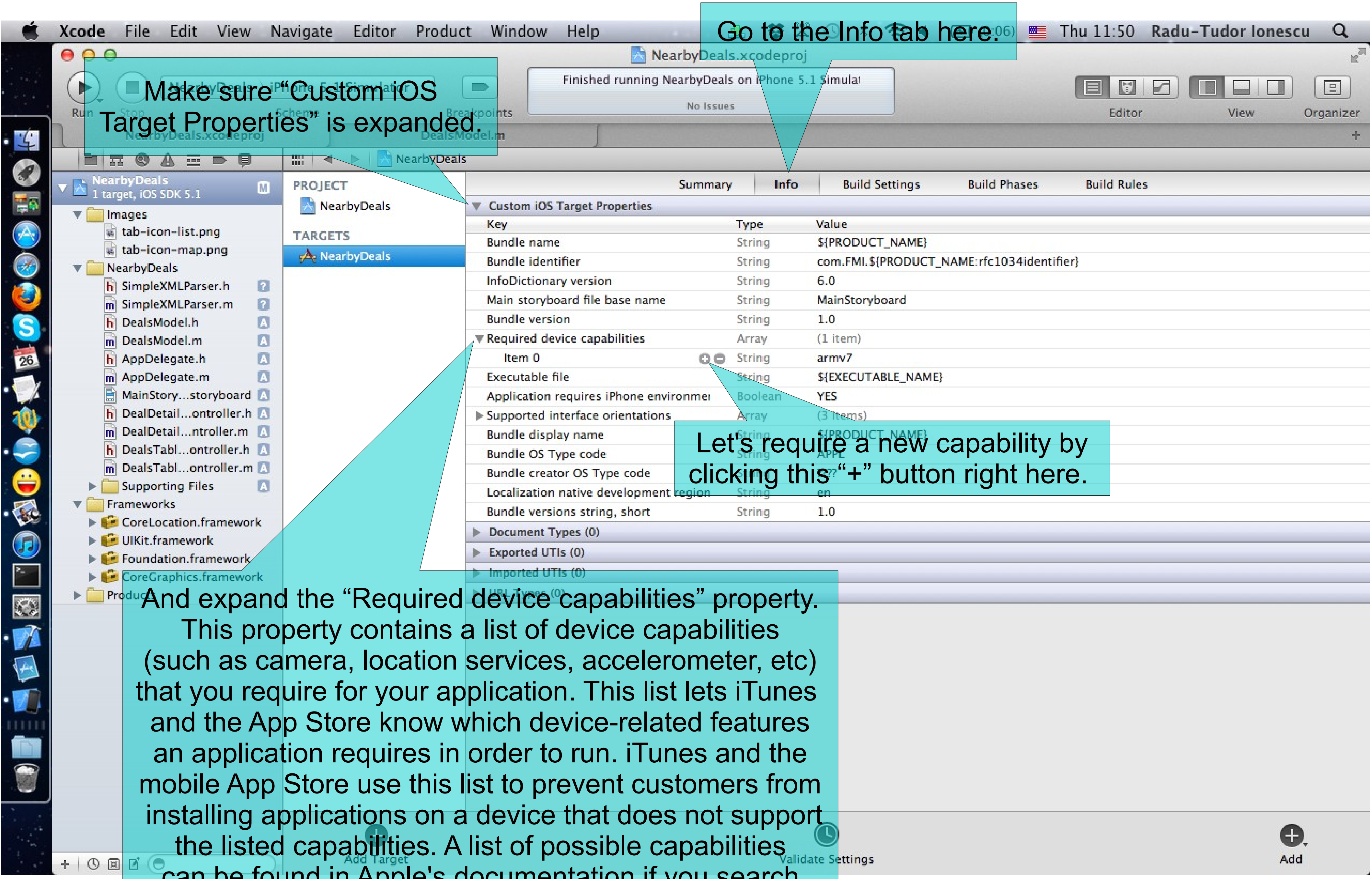
Configurations

Name	Based on Configuration File
Debug	No Configurations Set
Release	No Configurations Set

Command-line builds use: Release

Localizations

Language	Resources
English	2 Files Localized



Go to the Info tab here.

Make sure "Custom iOS Target Properties" is expanded.

Custom iOS Target Properties

Key	Type	Value
Bundle name	String	${PRODUCT_NAME}
Bundle identifier	String	com.FMI.${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	String	6.0
Main storyboard file base name	String	MainStoryboard
Bundle version	String	1.0
Required device capabilities	Array	(1 item)
Item 0	String	armv7
Executable file	String	${EXECUTABLE_NAME}
Application requires iPhone environment	Boolean	YES
Supported interface orientations	Array	(3 items)
Bundle display name	String	${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle creator OS Type code	String	????
Localization native development region	String	en
Bundle versions string, short	String	1.0

Document Types (0)

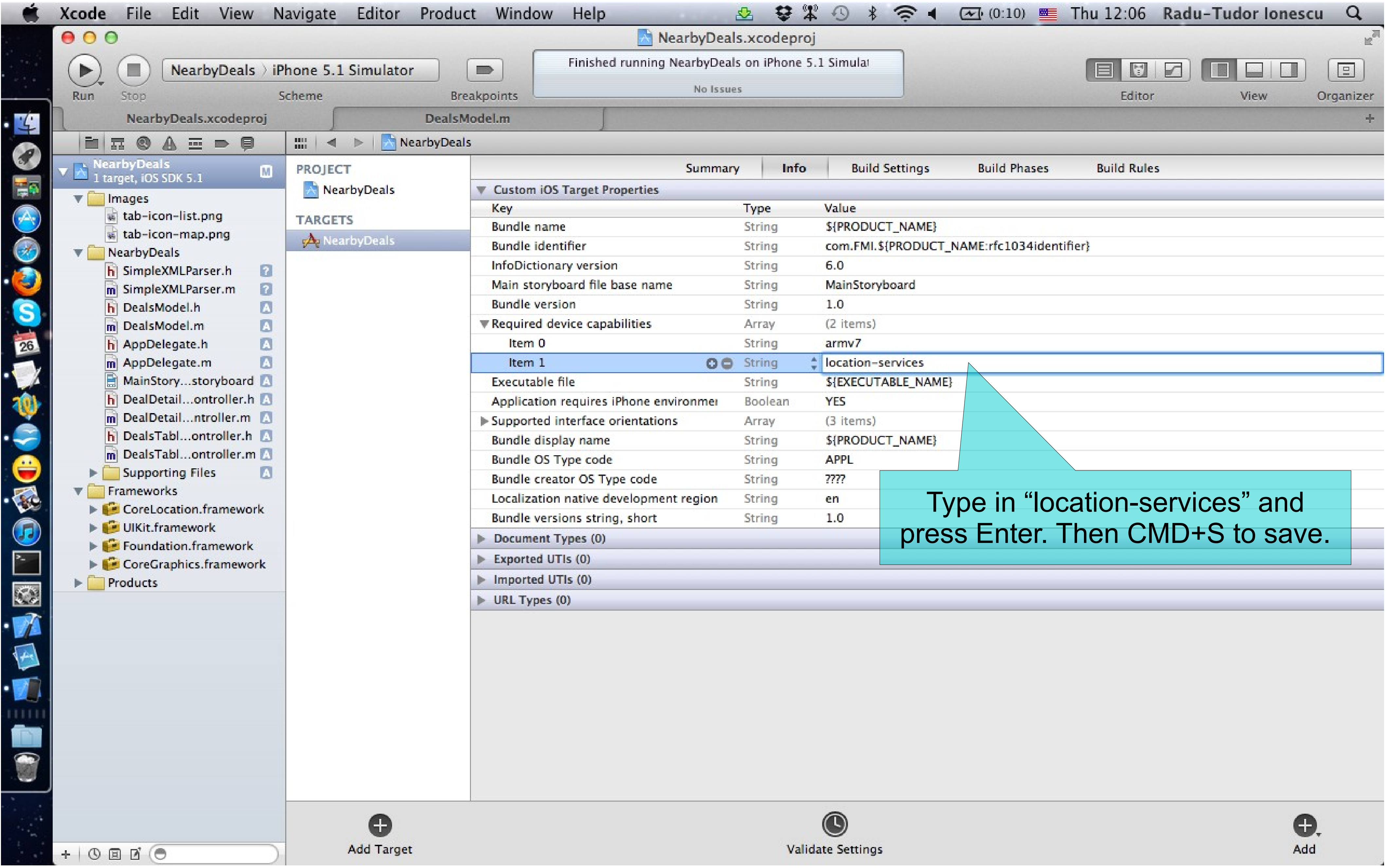
Exported UTIs (0)

Imported UTIs (0)

URL Types (0)

Let's require a new capability by clicking this "+" button right here.

And expand the "Required device capabilities" property. This property contains a list of device capabilities (such as camera, location services, accelerometer, etc) that you require for your application. This list lets iTunes and the App Store know which device-related features an application requires in order to run. iTunes and the mobile App Store use this list to prevent customers from installing applications on a device that does not support the listed capabilities. A list of possible capabilities can be found in Apple's documentation if you search for `UIRequiredDeviceCapabilities`.



Key	Type	Value
Bundle name	String	${PRODUCT_NAME}
Bundle identifier	String	com.FMI.${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	String	6.0
Main storyboard file base name	String	MainStoryboard
Bundle version	String	1.0
Required device capabilities	Array	(2 items)
Item 0	String	armv7
Item 1	String	location-services
Executable file	String	${EXECUTABLE_NAME}
Application requires iPhone environme	Boolean	YES
Supported interface orientations	Array	(3 items)
Bundle display name	String	${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle creator OS Type code	String	????
Localization native development region	String	en
Bundle versions string, short	String	1.0

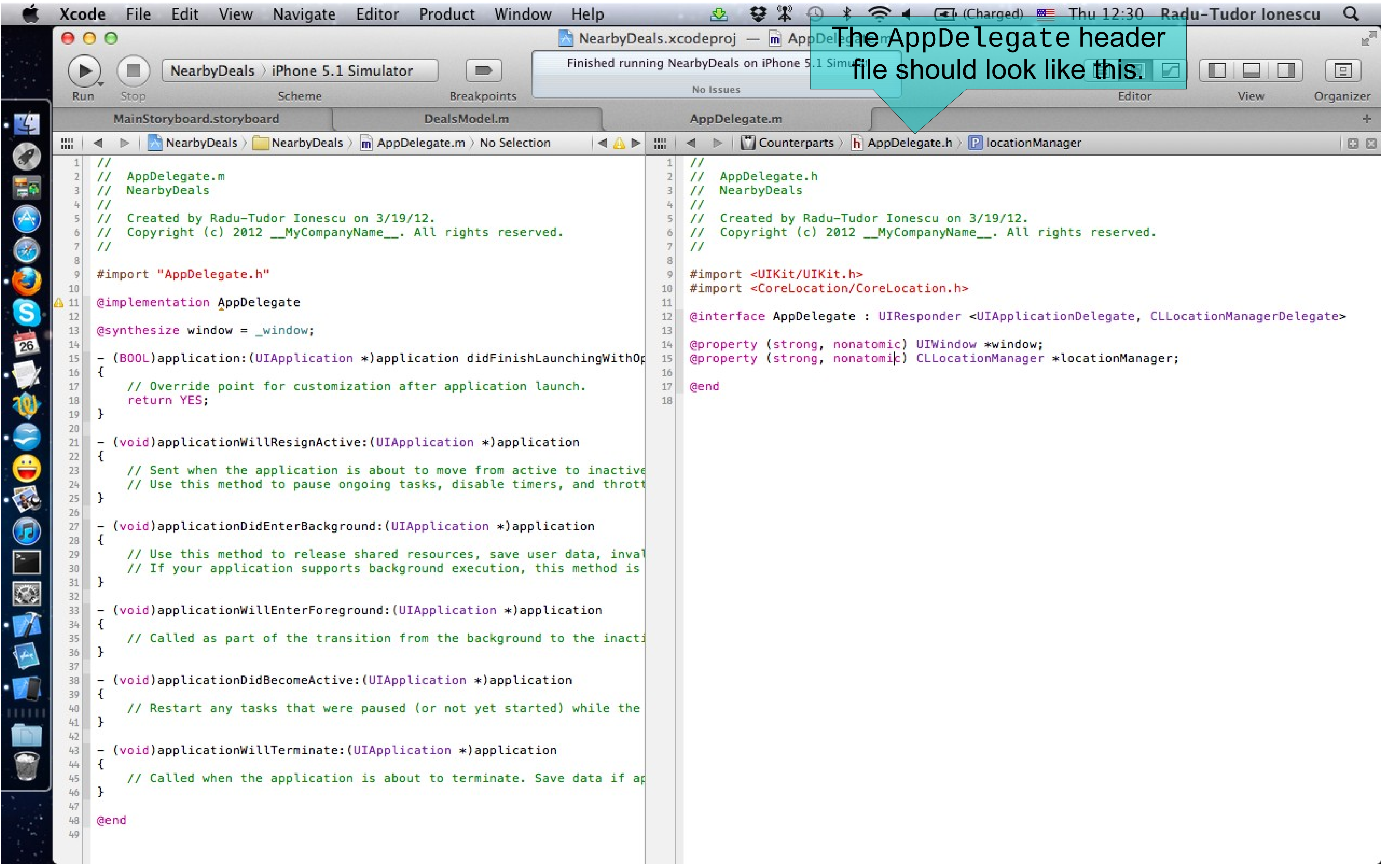
Type in “location-services” and press Enter. Then CMD+S to save.

Task 2

Task: Set up a `CLLocationManager` to receive location information in your application.

1. Create a new tab in Xcode (use the CMD+T shortcut keys).
2. Select AppDelegate.m in Project Navigator. Then close Project Navigator.
3. Open Assistant editor to have the AppDelegate.h header file on screen too.
4. Let's `#import` the Core Location framework into our `AppDelegate` header.
5. Add a new `@property` for the `CLLocationManager` that will get the device location for us.
6. Our application delegate will also be the `CLLocationManager` delegate, so let's declare that we implement the associated protocol (`CLLocationManagerDelegate`).

Look over the next screenshot to see how to do the above steps.


Xcode File Edit View Navigate Editor Product Window Help
Thu 12:30 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals › iPhone 5.1 Simulator
Finished running NearbyDeals on iPhone 5.1 Simulator
No Issues
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › No Selection
Counterparts › AppDelegate.h › locationManager

The AppDelegate header file should look like this.

```
//
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

@synthesize window = _window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOp
{
    // Override point for customization after application launch.
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive
    // Use this method to pause ongoing tasks, disable timers, and thrott
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, inval
    // If your application supports background execution, this method is
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inacti
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if ap
}

@end
```

```
//
//  AppDelegate.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate, CLLocationManagerDelegate>

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) CLLocationManager *locationManager;

@end
```

Task 2

Task: Set up a `CLLocationManager` to receive location information in your application.

7. In the `AppDelegate` implementation file, `#synthesize` the `locationManager` property and rename its instance variable by prefixing it with underscore.

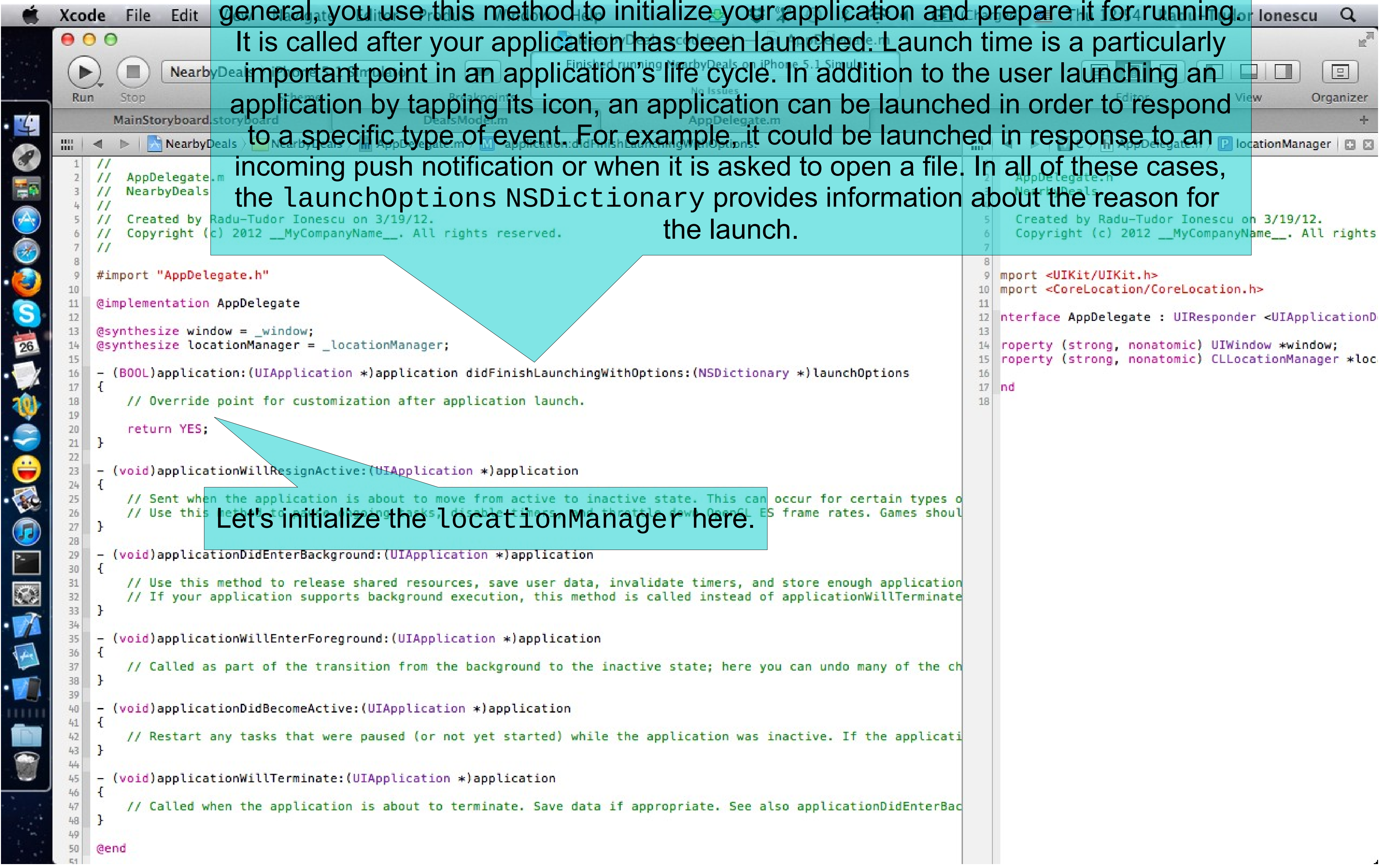
8. Similar to View Controllers, the application itself has a life cycle. The `UIApplicationDelegate` protocol declares methods that are implemented by the delegate of the singleton `UIApplication` object. These methods provide you with information about key events in an application's life cycle such as when it finished launching, when it is about to be terminated, when memory is low, and when important changes occur.

Let's initialize the `locationManager` when the application has finished launching and configure it to respond to other application events.

Look over the next screenshots to see how to do the above steps.

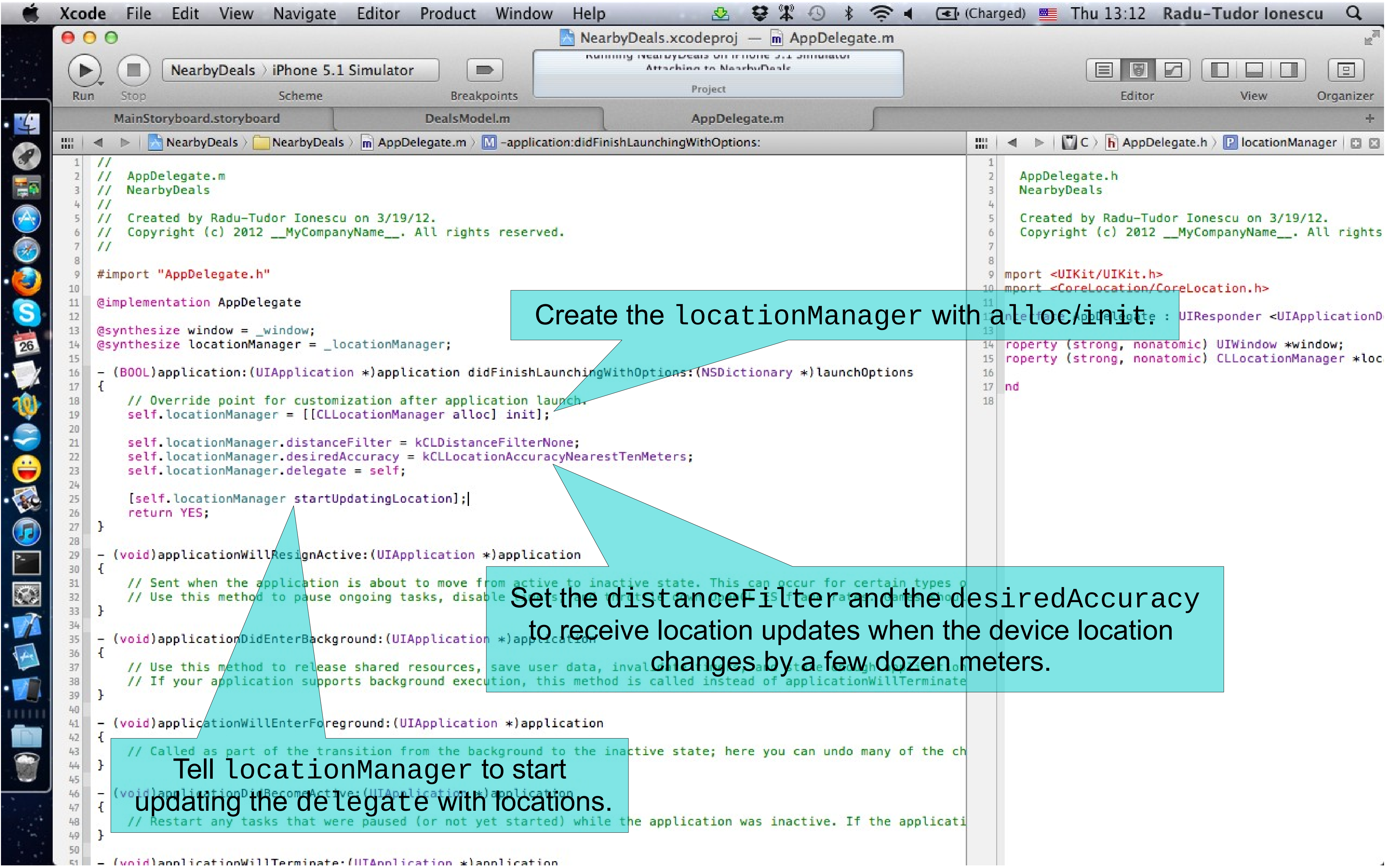
Let's implement the `application:didFinishLaunchingWithOptions` first. In general, you use this method to initialize your application and prepare it for running. It is called after your application has been launched. Launch time is a particularly important point in an application's life cycle. In addition to the user launching an application by tapping its icon, an application can be launched in order to respond to a specific type of event. For example, it could be launched in response to an incoming push notification or when it is asked to open a file. In all of these cases, the `launchOptions` `NSDictionary` provides information about the reason for the launch.

Let's initialize the `locationManager` here.



```
//
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

@synthesize window = _window;
@synthesize locationManager = _locationManager;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.

    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types o
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games shoul
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application
    // If your application supports background execution, this method is called instead of applicationWillTerminate
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the ch
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the applicati
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBac
}

@end
```

```
 AppDelegate.h
 NearbyDeals

 Created by Radu-Tudor Ionescu on 3/19/12.
 Copyright (c) 2012 __MyCompanyName__. All rights


mport <UIKit/UIKit.h>
mport <CoreLocation/CoreLocation.h>

nterface AppDelegate : UIResponder <UIApplicationD

roperty (strong, nonatomic) UIWindow *window;
roperty (strong, nonatomic) CLLocationManager *loc

nd
```


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 13:12 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme Breakpoints
Running NearbyDeals on iPhone 5.1 Simulator
Attaching to NearbyDeals
Project
Editor View Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › -application:didFinishLaunchingWithOptions:
C › AppDelegate.h › locationManager

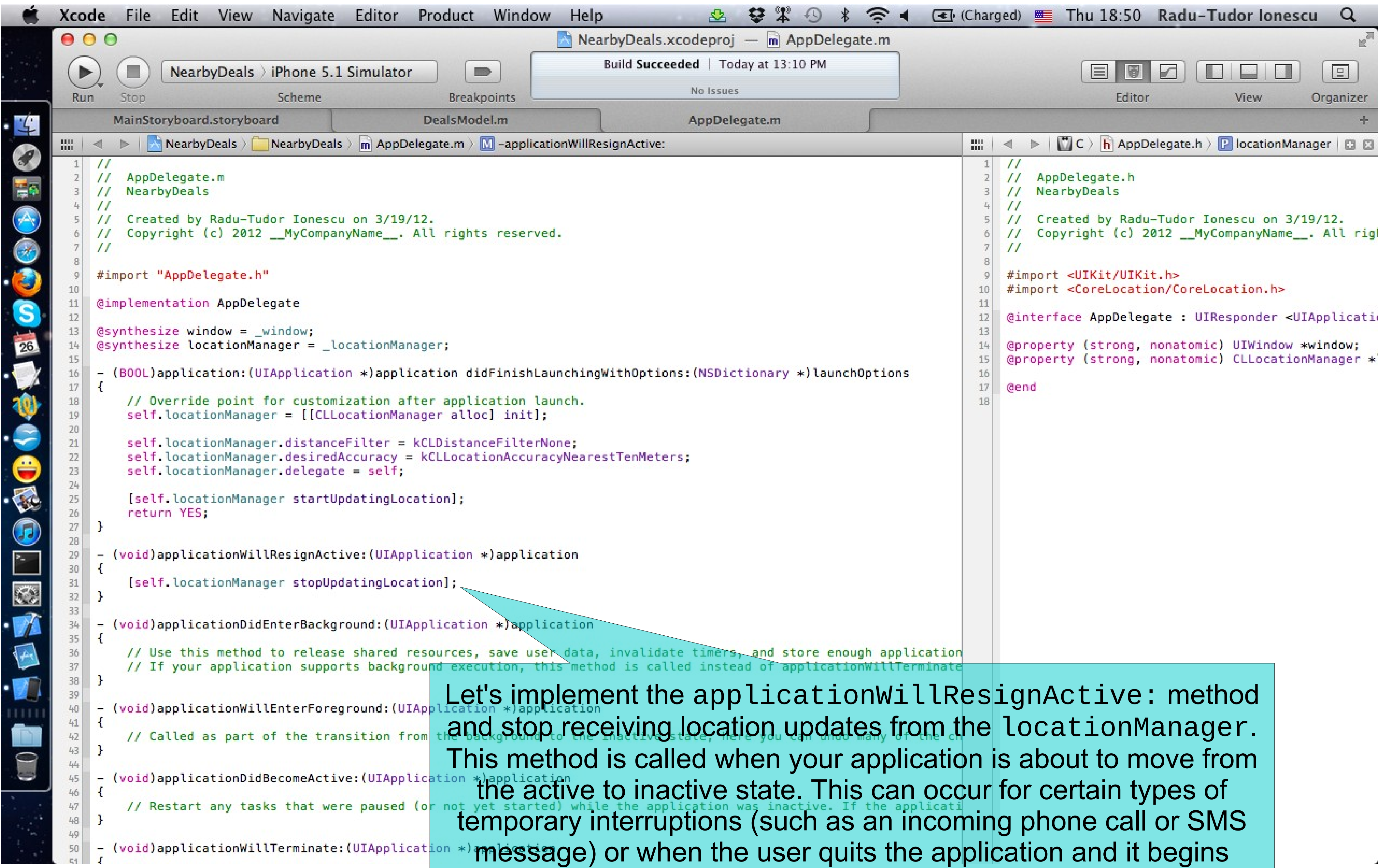
```
//
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

@synthesize window = _window;
@synthesize locationManager = _locationManager;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.locationManager = [[CLLocationManager alloc] init];

    self.locationManager.distanceFilter = kCLDistanceFilterNone;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
    self.locationManager.delegate = self;

    [self.locationManager startUpdatingLocation];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types o
    // Use this method to pause ongoing tasks, disable
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application
    // If your application supports background execution, this method is called instead of applicationWillTerminate
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the ch
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the applicati
}

- (void)applicationWillTerminate:(UIApplication *)application
```

```
  AppDelegate.h
  NearbyDeals

  Created by Radu-Tudor Ionescu on 3/19/12.
  Copyright (c) 2012 __MyCompanyName__. All rights


mport <UIKit/UIKit.h>
mport <CoreLocation/CoreLocation.h>

nterface AppDelegate : UIResponder <UIApplicationD

roperty (strong, nonatomic) UIWindow *window;
roperty (strong, nonatomic) CLLocationManager *loc

nd
```

Create the `locationManager` with `alloc/init`.

Set the `distanceFilter` and the `desiredAccuracy` to receive location updates when the device location changes by a few dozen meters.

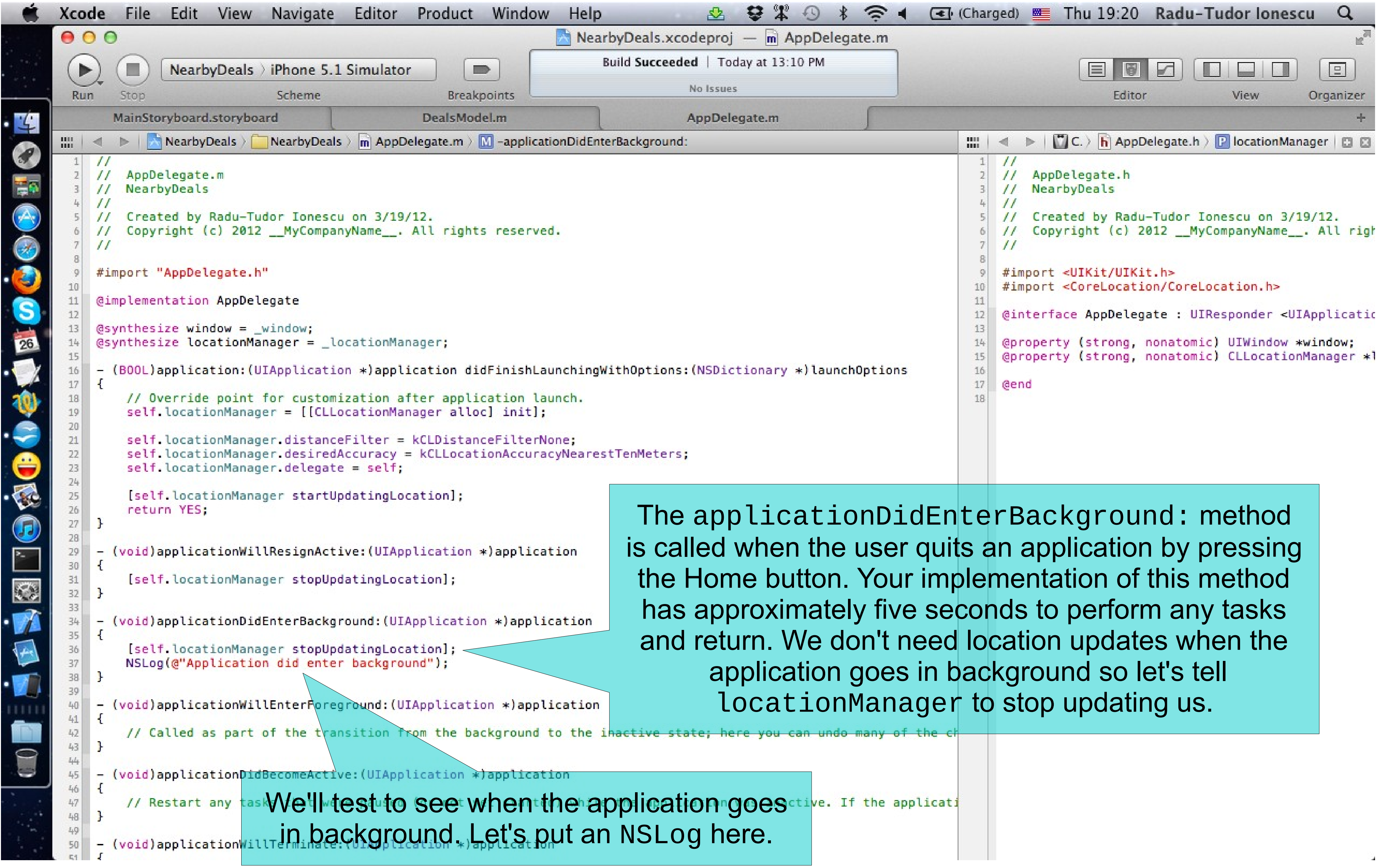
Tell `locationManager` to start updating the `delegate` with locations.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 18:50 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Today at 13:10 PM
No Issues
Editor View Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › -applicationWillResignActive:
C › AppDelegate.h › locationManager

```
//
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

@synthesize window = _window;
@synthesize locationManager = _locationManager;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.locationManager = [[CLLocationManager alloc] init];

    self.locationManager.distanceFilter = kCLDistanceFilterNone;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
    self.locationManager.delegate = self;

    [self.locationManager startUpdatingLocation];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application
    // If your application supports background execution, this method is called instead of applicationWillTerminate
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the ch
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the applicati
}

- (void)applicationWillTerminate:(UIApplication *)application
{
```

```
//
//  AppDelegate.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rig
//

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface AppDelegate : UIResponder <UIApplicati

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) CLLocationManager *

@end
```

Let's implement the `applicationWillResignActive:` method and stop receiving location updates from the `locationManager`. This method is called when your application is about to move from the active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.



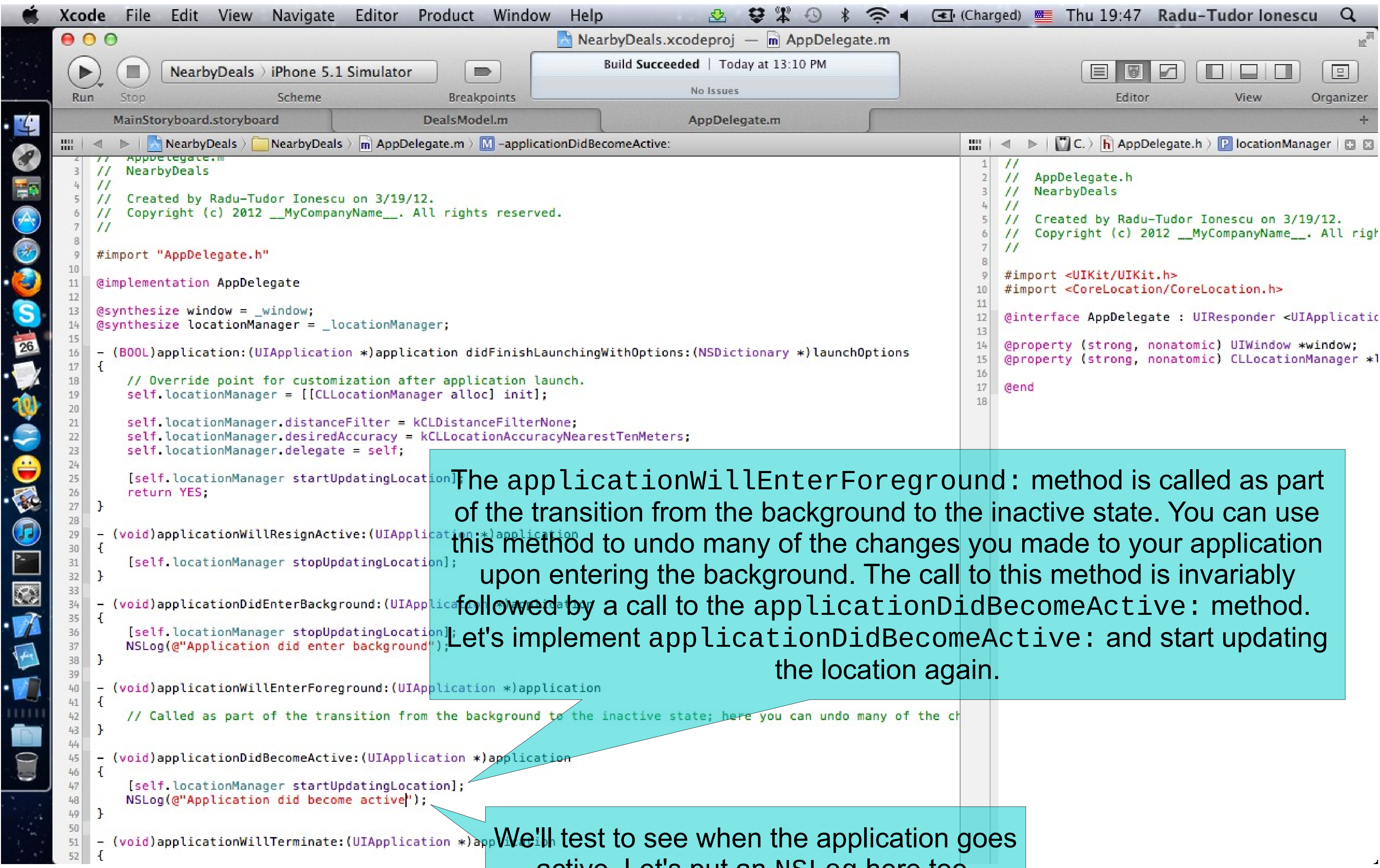
Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 19:20 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme Breakpoints
Build Succeeded | Today at 13:10 PM
No Issues
Editor View Organizer
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › -applicationDidEnterBackground:
C. › AppDelegate.h › locationManager

```
//
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

@synthesize window = _window;
@synthesize locationManager = _locationManager;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.locationManager = [[CLLocationManager alloc] init];

    self.locationManager.distanceFilter = kCLDistanceFilterNone;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
    self.locationManager.delegate = self;

    [self.locationManager startUpdatingLocation];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
    NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the ch
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the applicati
}

- (void)applicationWillTerminate:(UIApplication *)application
{
```

```
//
//  AppDelegate.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All righ
//

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface AppDelegate : UIResponder <UIApplicatio

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) CLLocationManager *l

@end
```

The `applicationDidEnterBackground:` method is called when the user quits an application by pressing the Home button. Your implementation of this method has approximately five seconds to perform any tasks and return. We don't need location updates when the application goes in background so let's tell `locationManager` to stop updating us.

We'll test to see when the application goes in background. Let's put an `NSLog` here.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 19:47 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop NearbyDeals iPhone 5.1 Simulator Scheme Breakpoints
Build Succeeded | Today at 13:10 PM
No Issues
Editor View Organizer
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals NearbyDeals AppDelegate.m -applicationDidBecomeActive:
C. AppDelegate.h locationManager

```
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

@synthesize window = _window;
@synthesize locationManager = _locationManager;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.locationManager = [[CLLocationManager alloc] init];

    self.locationManager.distanceFilter = kCLDistanceFilterNone;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
    self.locationManager.delegate = self;

    [self.locationManager startUpdatingLocation];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
    NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the ch
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    [self.locationManager startUpdatingLocation];
    NSLog(@"Application did become active");
}

- (void)applicationWillTerminate:(UIApplication *)application
{
```

```
//
//  AppDelegate.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All righ
//

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface AppDelegate : UIResponder <UIApplicati

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) CLLocationManager *l

@end
```

The `applicationWillEnterForeground:` method is called as part of the transition from the background to the inactive state. You can use this method to undo many of the changes you made to your application upon entering the background. The call to this method is invariably followed by a call to the `applicationDidBecomeActive:` method. Let's implement `applicationDidBecomeActive:` and start updating the location again.

We'll test to see when the application goes active. Let's put an `NSLog` here too.

Task 2

Task: Set up a `CLLocationManager` to receive location information in your application.

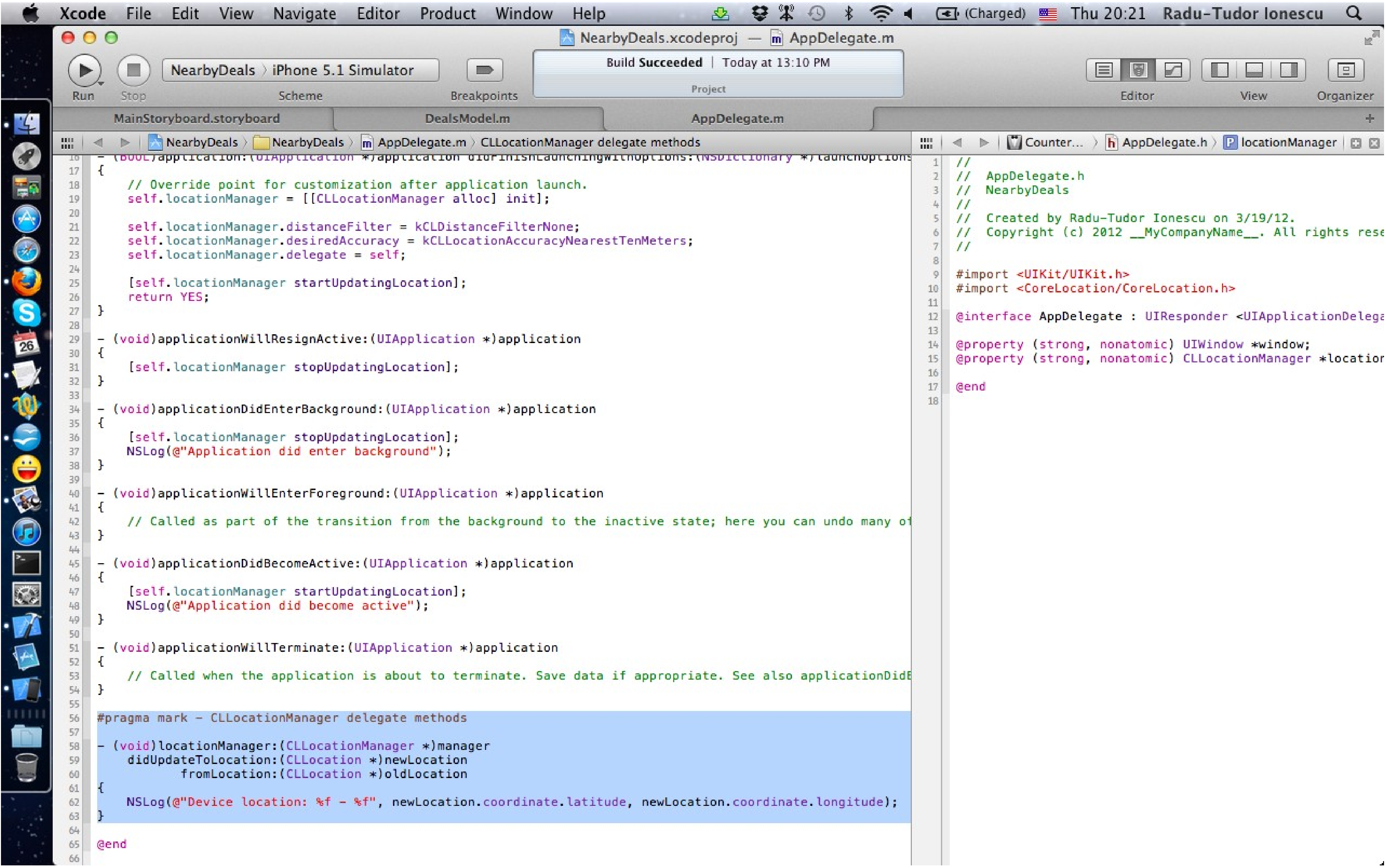
9. Next we should implement the `CLLocationManager`'s delegate methods.

Let's `#pragma mark` this section of code and put it at the end of the `AppDelegate` implementation block (right before `@end`).

10. The `delegate` object will get location updates when it receives the `locationManager:didUpdateToLocation:fromLocation:` message.

Implement this method and let's print the device location to the console using an `NSLog()`.

Next screenshot shows how to do this.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 20:21 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop Scheme Breakpoints
NearbyDeals › iPhone 5.1 Simulator
Project
Editor View Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › CLLocationManager delegate methods
Counter... › AppDelegate.h › locationManager

Build Succeeded | Today at 13:10 PM

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.locationManager = [[CLLocationManager alloc] init];

    self.locationManager.distanceFilter = kCLDistanceFilterNone;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
    self.locationManager.delegate = self;

    [self.locationManager startUpdatingLocation];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
    NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    [self.locationManager startUpdatingLocation];
    NSLog(@"Application did become active");
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidE
}

#pragma mark - CLLocationManager delegate methods

- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    NSLog(@"Device location: %f - %f", newLocation.coordinate.latitude, newLocation.coordinate.longitude);
}

@end
```

```
//
//  AppDelegate.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights rese
//

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface AppDelegate : UIResponder <UIApplicationDelega

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) CLLocationManager *locatior

@end
```

Task 2

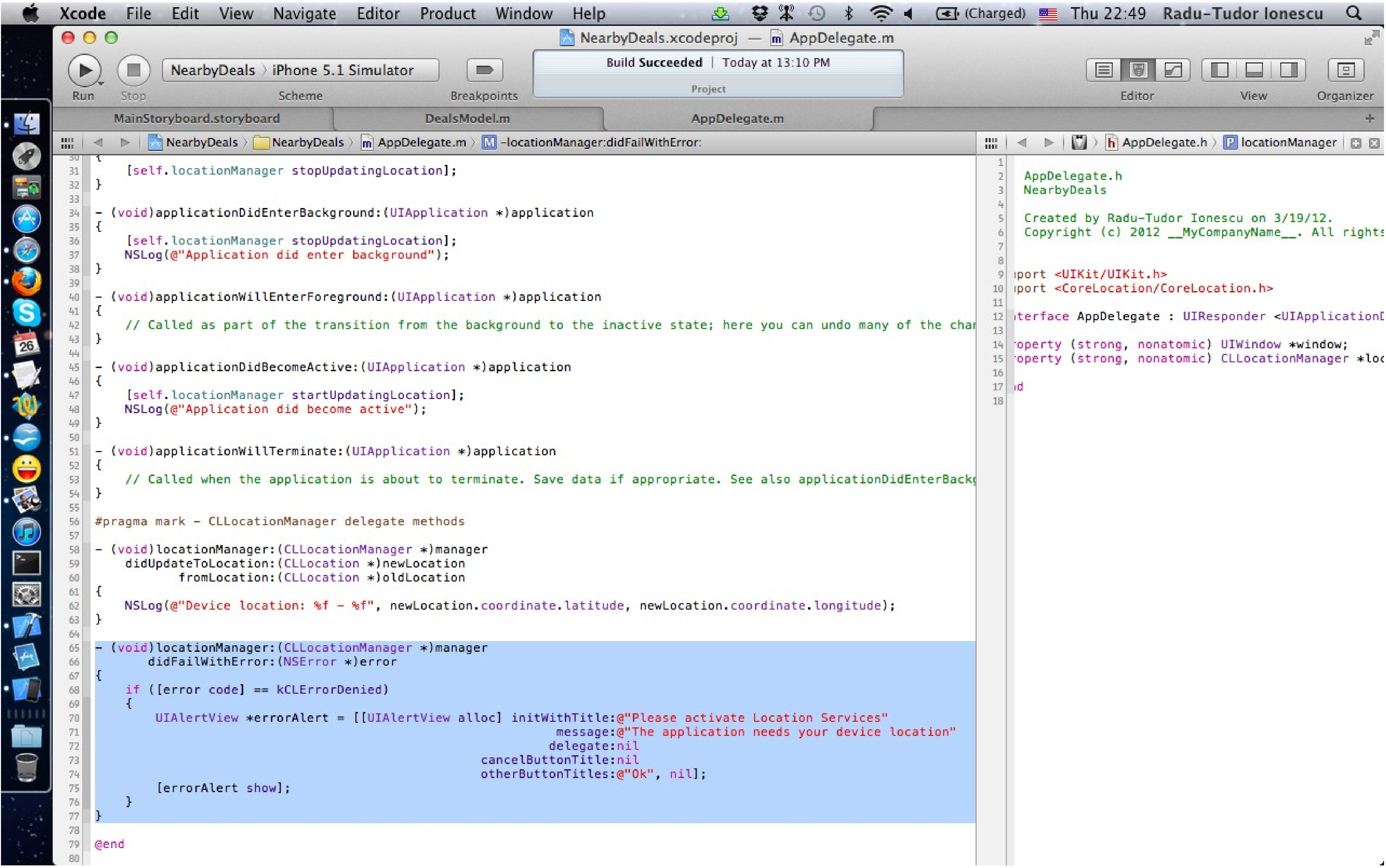
Task: Set up a `CLLocationManager` to receive location information in your application.

11. Your `delegate` object will get notified if the `locationManager` is unable to determine the device location by receiving the `locationManager:didFailWithError:` message.

If it reports a `kCLErrorLocationUnknown` error, we can simply ignore the error and wait for a new event (the `locationManager` keeps trying to get the device location).

If the user denies your application's use of the Location Services, this method reports a `kCLErrorDenied` error. In this case, it's best to let the user know our application can't function without this service. We will show up and `UIAlertView` with an appropriate message.

Look over the next screenshot to see how to implement this method.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 22:49 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme Breakpoints
Build Succeeded | Today at 13:10 PM
Project
Editor View Organizer
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › -locationManager:didFailWithError:
AppDelegate.h › locationManager

```
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
    NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the char
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    [self.locationManager startUpdatingLocation];
    NSLog(@"Application did become active");
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackg
}

#pragma mark - CLLocationManager delegate methods

- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    NSLog(@"Device location: %f - %f", newLocation.coordinate.latitude, newLocation.coordinate.longitude);
}

- (void)locationManager:(CLLocationManager *)manager
       didFailWithError:(NSError *)error
{
    if ([error code] == kCLErrorDenied)
    {
        UIAlertView *errorAlert = [[UIAlertView alloc] initWithTitle:@"Please activate Location Services"
                                                             message:@"The application needs your device location"
                                                            delegate:nil
                                                   cancelButtonTitle:nil
                                                   otherButtonTitles:@"Ok", nil];
        [errorAlert show];
    }
}

@end
```

```
 AppDelegate.h
 NearbyDeals

 Created by Radu-Tudor Ionescu on 3/19/12.
 Copyright (c) 2012 __MyCompanyName__. All rights


port <UIKit/UIKit.h>
port <CoreLocation/CoreLocation.h>

terface AppDelegate : UIResponder <UIApplication

operty (strong, nonatomic) UIWindow *window;
operty (strong, nonatomic) CLLocationManager *loc

d
```

Task 2

Task: Set up a `CLLocationManager` to receive location information in your application.

12. Run the application in iOS Simulator.

13. You will be asked to enable Location Services for this application. Deny this request to test what happens wit our application.

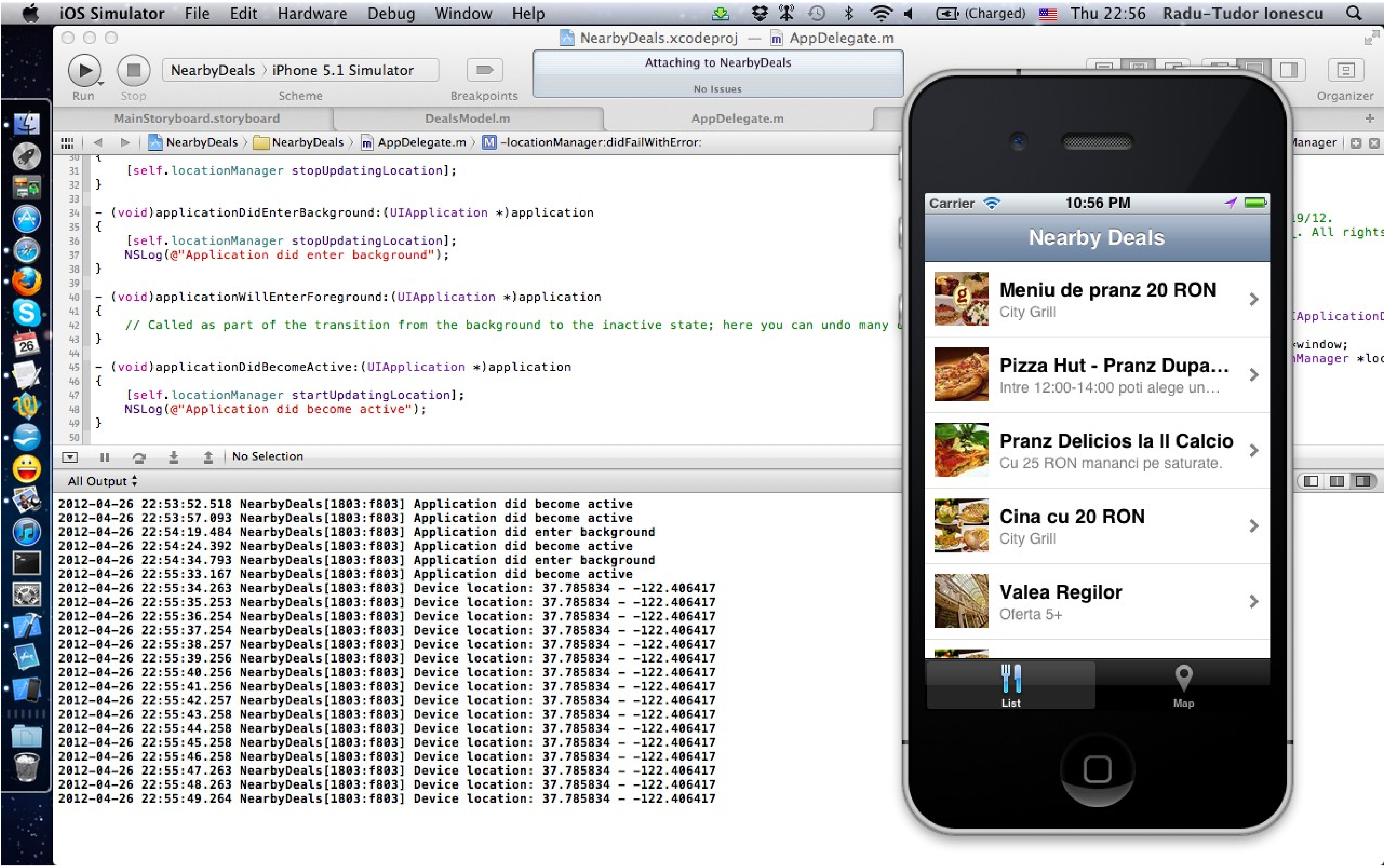
The error message should appear on screen. Click “Ok” to dismiss it.

14. Click on the Home button to put the application in background. Then open it up again. Notice the messages that appear on the console.

The error message should appear again. Click “Ok” to dismiss it.

15. Go to the Settings app and turn on Location Services for our application. Open the application again and look for the location messages in the console.

16. Stop running the application.


iOS Simulator File Edit Hardware Debug Window Help
(Charged) Thu 22:56 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop NearbyDeals iPhone 5.1 Simulator Scheme Breakpoints
Attaching to NearbyDeals
No Issues
Organizer
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals NearbyDeals AppDelegate.m -locationManager:didFailWithError:
No Selection
All Output
Carrier 10:56 PM
Nearby Deals
Meniu de pranz 20 RON
City Grill
Pizza Hut - Pranz Dupa…
Intre 12:00-14:00 poti alege un…
Pranz Delicios la Il Calcio
Cu 25 RON mananci pe saturate.
Cina cu 20 RON
City Grill
Valea Regilor
Oferta 5+
List
Map

```
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
   NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
   // Called as part of the transition from the background to the inactive state; here you can undo many
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    [self.locationManager startUpdatingLocation];
   NSLog(@"Application did become active");
}
```

```
2012-04-26 22:53:52.518 NearbyDeals[1803:f803] Application did become active
2012-04-26 22:53:57.093 NearbyDeals[1803:f803] Application did become active
2012-04-26 22:54:19.484 NearbyDeals[1803:f803] Application did enter background
2012-04-26 22:54:24.392 NearbyDeals[1803:f803] Application did become active
2012-04-26 22:54:34.793 NearbyDeals[1803:f803] Application did enter background
2012-04-26 22:55:33.167 NearbyDeals[1803:f803] Application did become active
2012-04-26 22:55:34.263 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:35.253 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:36.254 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:37.254 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:38.257 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:39.256 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:40.256 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:41.256 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:42.257 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:43.258 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:44.258 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:45.258 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:46.258 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:47.263 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:48.263 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
2012-04-26 22:55:49.264 NearbyDeals[1803:f803] Device location: 37.785834 - -122.406417
```

Task 2

Task: Set up a `CLLocationManager` to receive location information in your application.

17. Our application is now configured to receive location updates whenever the device location changes. The next thing to do is to save the current device location into our DealsModel.

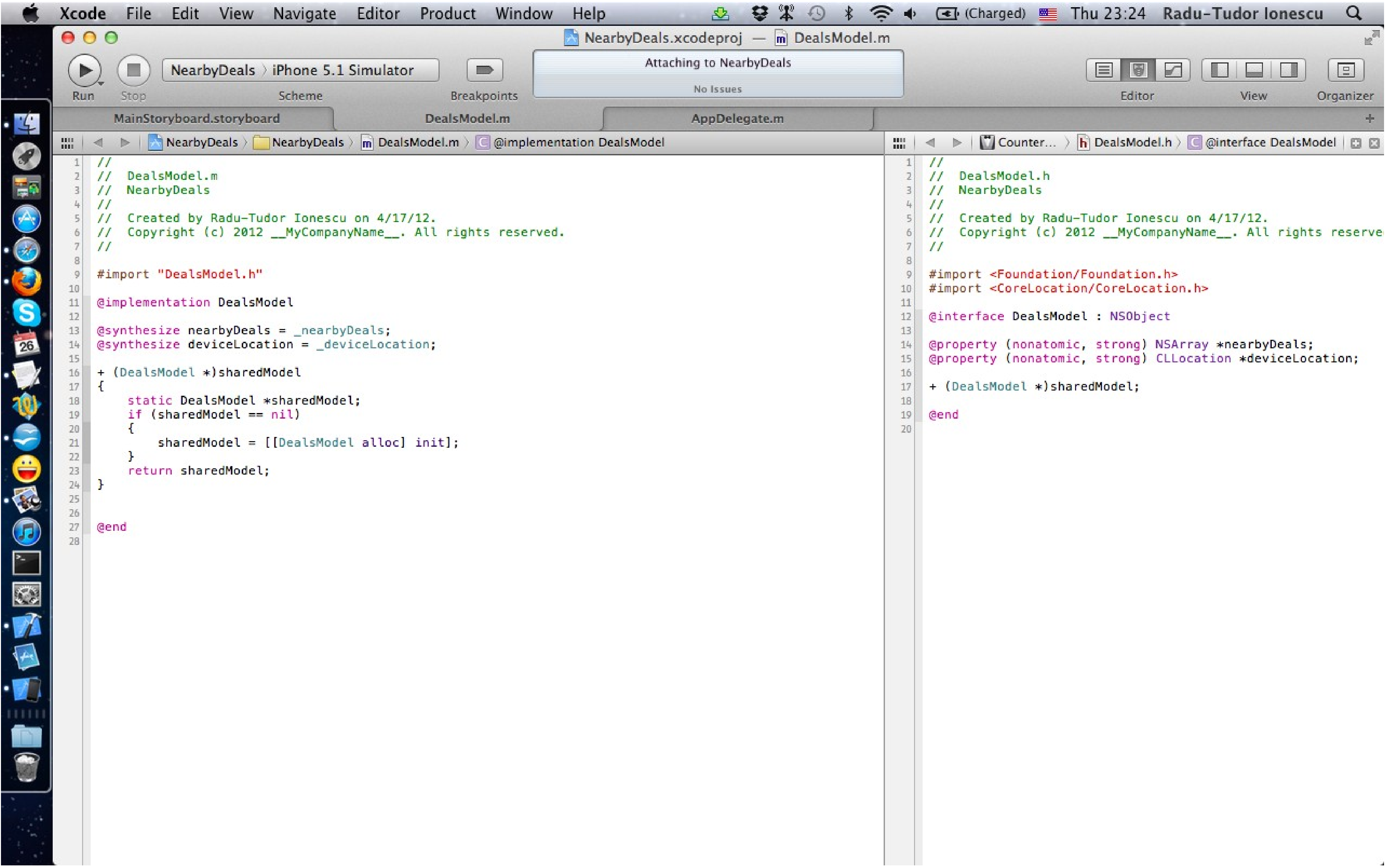
Go to the DealsModel.m tab in Xcode.

18. Let's add a `@property` called `deviceLocation` that will be a pointer to a `CLLocation` object. It has to be `strong` since no one else refers to it.

19. You will also have to `#import` the Core Location framework in the DealsModel header file.

20. As usual, `#synthesize` this property in the implementation file and rename its instance variable.

Look over the next slide for hints.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 23:24 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — DealsModel.m
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Attaching to NearbyDeals
No Issues
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › DealsModel.m › @implementation DealsModel
Counter... › DealsModel.h › @interface DealsModel

```
//
//  DealsModel.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/17/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealsModel.h"

@implementation DealsModel

@synthesize nearbyDeals = _nearbyDeals;
@synthesize deviceLocation = _deviceLocation;

+ (DealsModel *)sharedModel
{
    static DealsModel *sharedModel;
    if (sharedModel == nil)
    {
        sharedModel = [[DealsModel alloc] init];
    }
    return sharedModel;
}


@end
```

```
//
//  DealsModel.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/17/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserve
//

#import <Foundation/Foundation.h>
#import <CoreLocation/CoreLocation.h>

@interface DealsModel : NSObject

@property (nonatomic, strong) NSArray *nearbyDeals;
@property (nonatomic, strong) CLLocation *deviceLocation;

+ (DealsModel *)sharedModel;

@end
```

Task 2

Task: Set up a `CLLocationManager` to receive location information in your application.

21. We have to set the `deviceLocation` of the `sharedModel` when our `AppDelegate` receives location updates.

Go to the AppDelegate.m tab in Xcode.

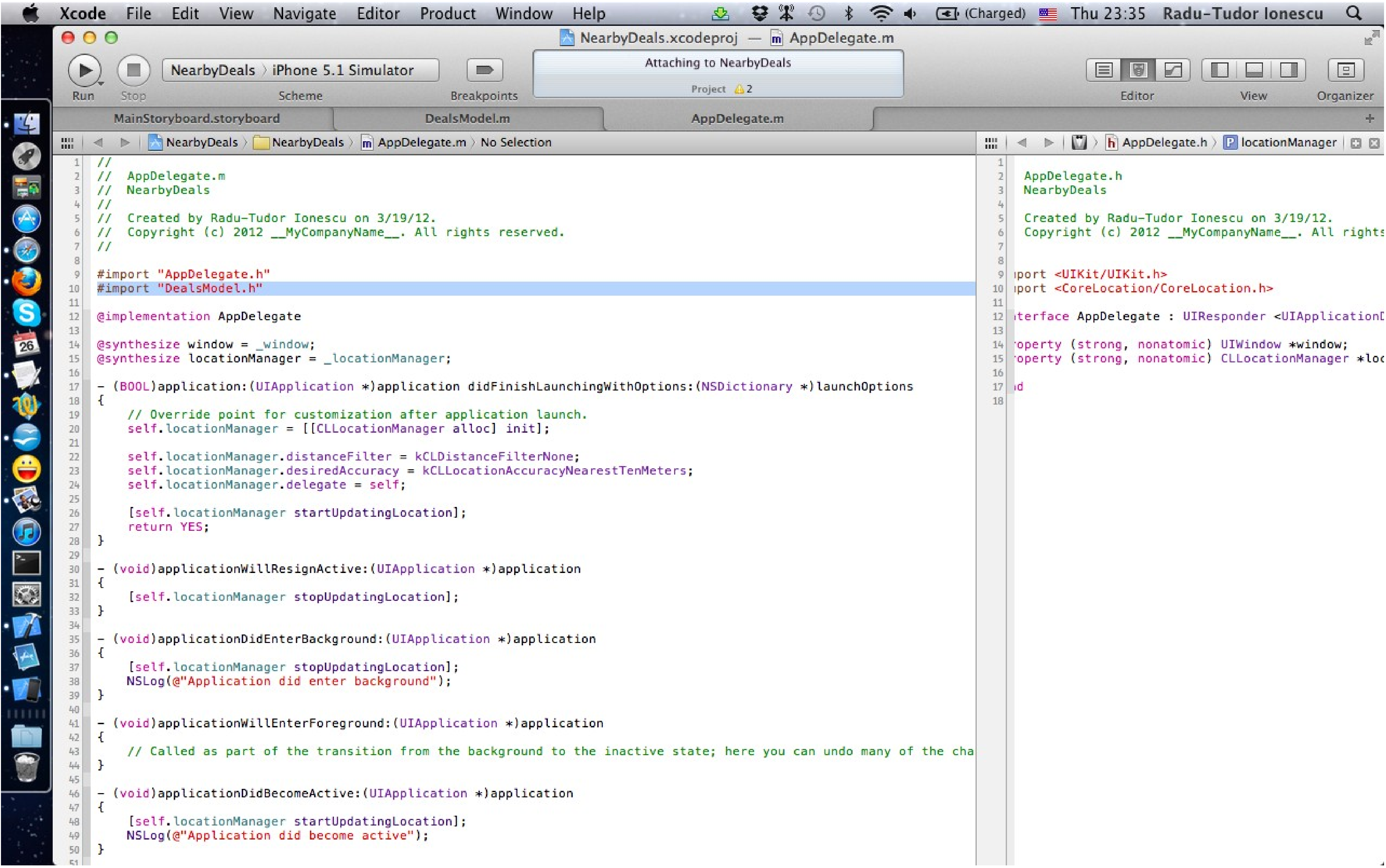
22. The first thing to do is to `#import` the “DealsModel.h” file into our `AppDelegate` implementation file.

23. Re-implement the `CLLocationManager's delegate` method `locationManager:didUpdateToLocation:fromLocation:` to save the device location into our `sharedModel`.

We will update the `deviceLocation` only when the `newLocation` is at least 100 meters away from the previous `deviceLocation` saved.

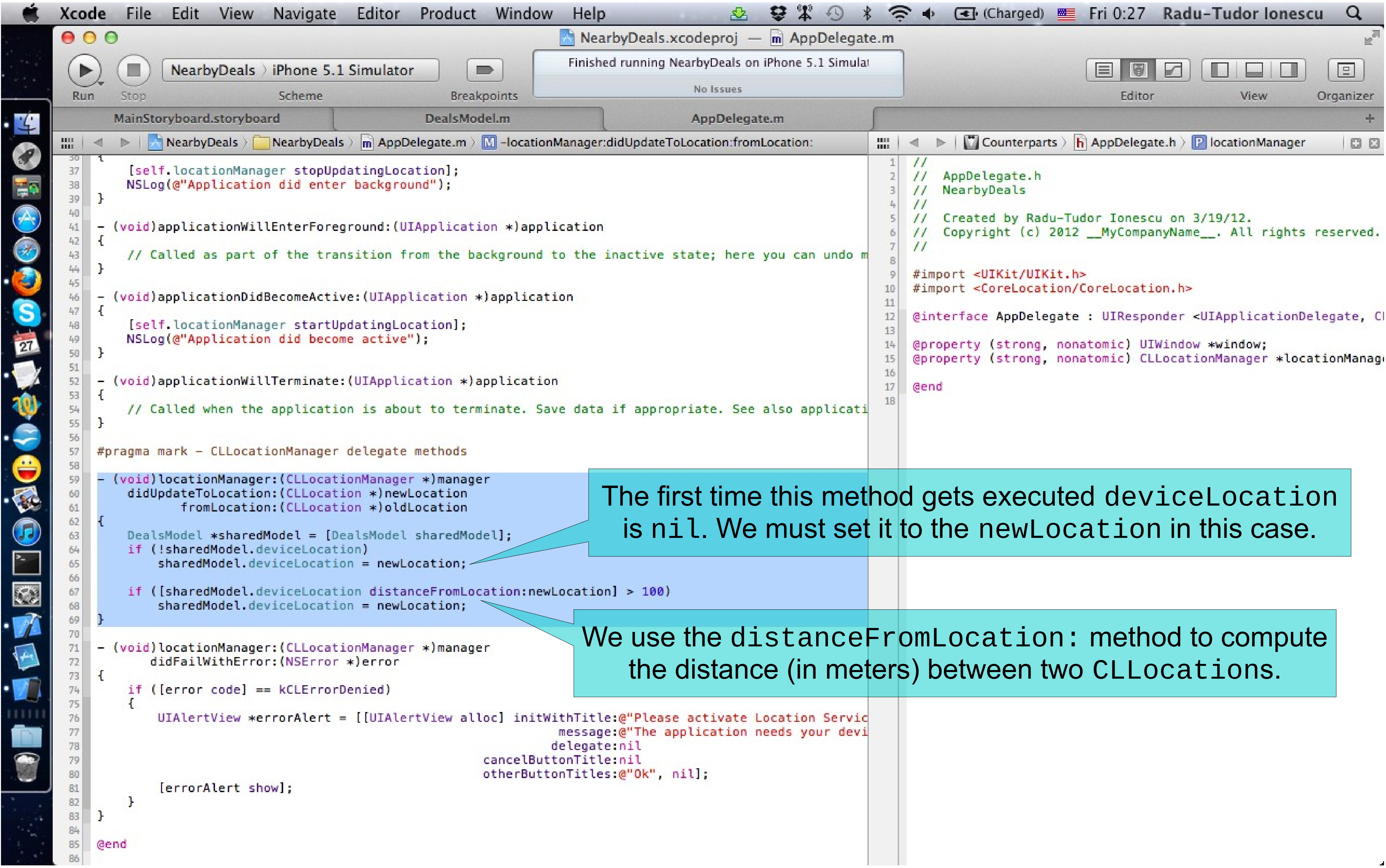
Also delete the `NSLog()` that was printing locations on the console.

Look over the next slides for hints.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Thu 23:35 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Attaching to NearbyDeals
Project ⚠2
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › No Selection
AppDelegate.h › locationManager

```objc
//
//  AppDelegate.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "AppDelegate.h"
#import "DealsModel.h"

@implementation AppDelegate

@synthesize window = _window;
@synthesize locationManager = _locationManager;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.locationManager = [[CLLocationManager alloc] init];

    self.locationManager.distanceFilter = kCLDistanceFilterNone;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
    self.locationManager.delegate = self;

    [self.locationManager startUpdatingLocation];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self.locationManager stopUpdatingLocation];
    NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the cha
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    [self.locationManager startUpdatingLocation];
    NSLog(@"Application did become active");
}
```

```objc

 AppDelegate.h
 NearbyDeals

 Created by Radu-Tudor Ionescu on 3/19/12.
 Copyright (c) 2012 __MyCompanyName__. All rights


iport <UIKit/UIKit.h>
iport <CoreLocation/CoreLocation.h>

iterface AppDelegate : UIResponder <UIApplicationD

roperty (strong, nonatomic) UIWindow *window;
roperty (strong, nonatomic) CLLocationManager *loc

id
```


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 0:27 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — AppDelegate.m
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals › iPhone 5.1 Simulator
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals › NearbyDeals › AppDelegate.m › -locationManager:didUpdateToLocation:fromLocation:
Counterparts › AppDelegate.h › locationManager

Finished running NearbyDeals on iPhone 5.1 Simulat
No Issues

```
{
    [self.locationManager stopUpdatingLocation];
    NSLog(@"Application did enter background");
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo m
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    [self.locationManager startUpdatingLocation];
    NSLog(@"Application did become active");
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate. See also applicati
}

#pragma mark - CLLocationManager delegate methods

- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    DealsModel *sharedModel = [DealsModel sharedModel];
    if (!sharedModel.deviceLocation)
        sharedModel.deviceLocation = newLocation;

    if ([sharedModel.deviceLocation distanceFromLocation:newLocation] > 100)
        sharedModel.deviceLocation = newLocation;
}

- (void)locationManager:(CLLocationManager *)manager
       didFailWithError:(NSError *)error
{
    if ([error code] == kCLErrorDenied)
    {
        UIAlertView *errorAlert = [[UIAlertView alloc] initWithTitle:@"Please activate Location Servic
                                                             message:@"The application needs your devi
                                                            delegate:nil
                                                   cancelButtonTitle:nil
                                                   otherButtonTitles:@"Ok", nil];
        [errorAlert show];
    }
}

@end
```

```
//
//  AppDelegate.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 3/19/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate, C

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) CLLocationManager *locationManag

@end
```

The first time this method gets executed `deviceLocation` is `nil`. We must set it to the `newLocation` in this case.

We use the `distanceFromLocation:` method to compute the distance (in meters) between two `CLLocations`.

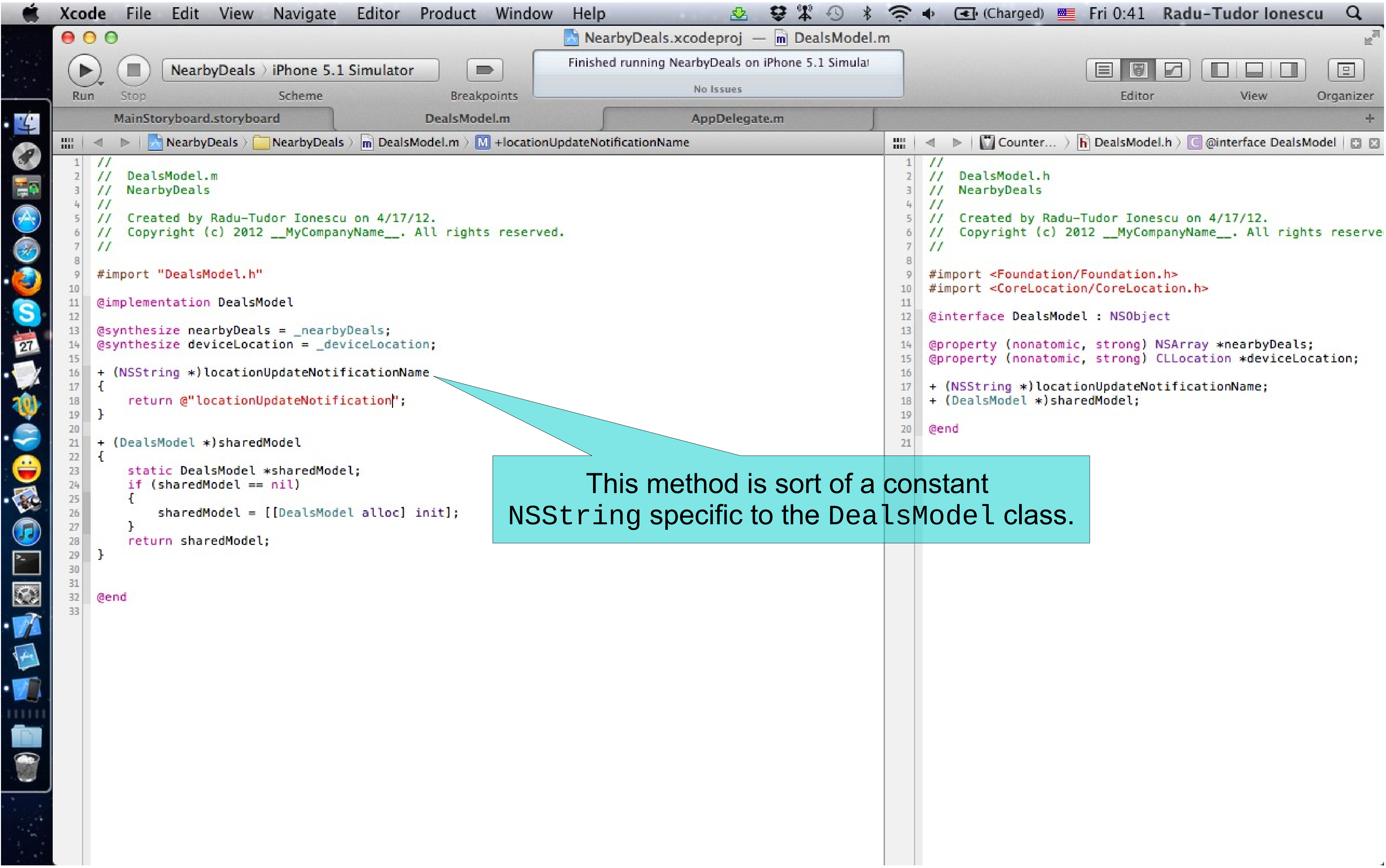
Task 3

Task: Configure the Deals Table View Controller to load deals near the device location.

1. We have saved the device location into our application's Model. We need to find a way to let the Deals Table View Controller know about the updated device locations. Recall the MVC design pattern. Using Notification Center and Key-Value Observation, the `sharedModel` can send notifications to its observers when a device location update occurs. We will add the Deals Table View Controller as an observer of the `sharedModel`. Upon receiving a notification our Table View Controller will make a request to the GeoAds+ API using the new device location.

First, go to the DealsModel.m tab in Xcode. Add a public class method that will return the notification name that will be sent when the `deviceLocation` setter gets called. Name this class method `locationUpdateNotificationName`. Implement it to return the `@”LocationUpdateNotification”` string.

Look over the next slide for hints.


Xcode File Edit View Navigate Editor Product Window Help
NearbyDeals.xcodeproj — DealsModel.m
NearbyDeals › iPhone 5.1 Simulator
Finished running NearbyDeals on iPhone 5.1 Simulat
No Issues
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › DealsModel.m › +locationUpdateNotificationName
Counter... › DealsModel.h › @interface DealsModel

```
//
//  DealsModel.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/17/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealsModel.h"

@implementation DealsModel

@synthesize nearbyDeals = _nearbyDeals;
@synthesize deviceLocation = _deviceLocation;

+ (NSString *)locationUpdateNotificationName
{
    return @"locationUpdateNotification";
}

+ (DealsModel *)sharedModel
{
    static DealsModel *sharedModel;
    if (sharedModel == nil)
    {
        sharedModel = [[DealsModel alloc] init];
    }
    return sharedModel;
}


@end
```

```
//
//  DealsModel.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/17/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserve
//

#import <Foundation/Foundation.h>
#import <CoreLocation/CoreLocation.h>

@interface DealsModel : NSObject

@property (nonatomic, strong) NSArray *nearbyDeals;
@property (nonatomic, strong) CLLocation *deviceLocation;

+ (NSString *)locationUpdateNotificationName;
+ (DealsModel *)sharedModel;

@end
```

This method is sort of a constant `NSString` specific to the `DealsModel` class.

Task 3

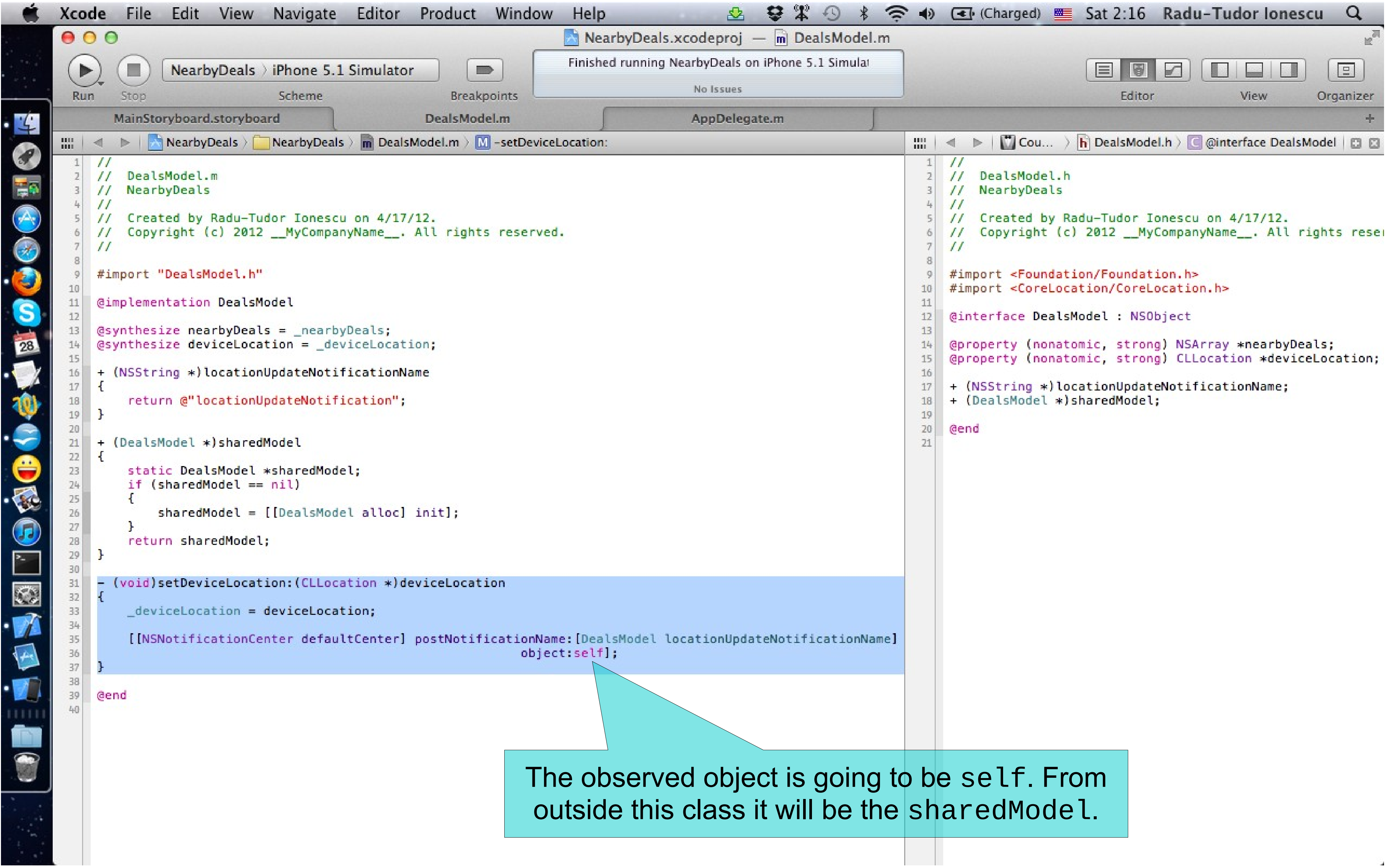
Task: Configure the Deals Table View Controller to load deals near the device location.

2. Whenever the `deviceLocation` is updated with a new value, we should post the notification to our Model's observers. To post the notification we have to implement the setter of this `@property`.

Besides setting the value of the associated instance variable, we send the `postNotificationName:object:` message to the default `NSNotificationCenter`.

Note that we will discuss notifications in detail in the last lecture.

Look over the next slide for help.



Xcode File Edit View Navigate Editor Product Window Help
Sat 2:16 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — DealsModel.m
Run Stop Scheme Breakpoints
NearbyDeals › iPhone 5.1 Simulator
Finished running NearbyDeals on iPhone 5.1 Simulat
No Issues
Editor View Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › DealsModel.m › -setDeviceLocation:
Cou... › DealsModel.h › @interface DealsModel

```
//
//  DealsModel.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/17/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealsModel.h"

@implementation DealsModel

@synthesize nearbyDeals = _nearbyDeals;
@synthesize deviceLocation = _deviceLocation;

+ (NSString *)locationUpdateNotificationName
{
    return @"locationUpdateNotification";
}

+ (DealsModel *)sharedModel
{
    static DealsModel *sharedModel;
    if (sharedModel == nil)
    {
        sharedModel = [[DealsModel alloc] init];
    }
    return sharedModel;
}

- (void)setDeviceLocation:(CLLocation *)deviceLocation
{
    _deviceLocation = deviceLocation;
    
    [[NSNotificationCenter defaultCenter] postNotificationName:[DealsModel locationUpdateNotificationName]
                                                        object:self];
}

@end
```

```
//
//  DealsModel.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/17/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights rese
//

#import <Foundation/Foundation.h>
#import <CoreLocation/CoreLocation.h>

@interface DealsModel : NSObject

@property (nonatomic, strong) NSArray *nearbyDeals;
@property (nonatomic, strong) CLLocation *deviceLocation;

+ (NSString *)locationUpdateNotificationName;
+ (DealsModel *)sharedModel;

@end
```

The observed object is going to be `self`. From outside this class it will be the `sharedModel`.

Task 3

Task: Configure the Deals Table View Controller to load deals near the device location.

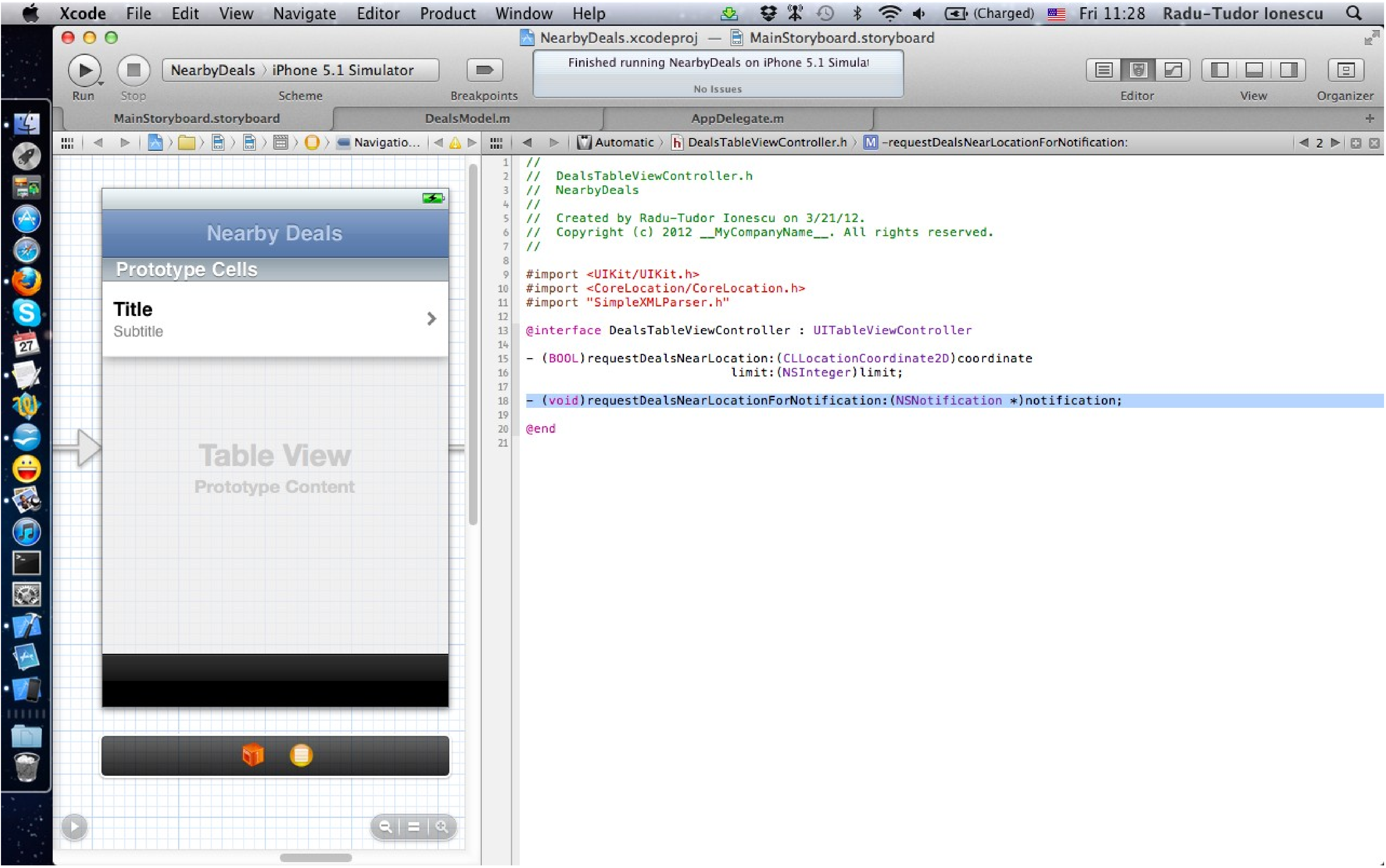
3. Switch to the MainStoryboard.storyboard tab in Xcode.

4. Select the Table View Controller from Interface Builder to see the associated class files in Assistant Editor.

Make sure the DealsTableViewController.h file is selected in Assistant Editor.

5. Declare a new instance method that will be executed when our Table View Controller receives the location update notification. Name this method `requestDealsNearLocationForNotification:`.

Look over the next slide for help.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 11:28 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
DealsModel.m
AppDelegate.m
Navigatio…
Automatic › DealsTableViewController.h › -requestDealsNearLocationForNotification:
Nearby Deals
Prototype Cells
Title
Subtitle
Table View
Prototype Content

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

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>
#import "SimpleXMLParser.h"

@interface DealsTableViewController : UITableViewController

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit;

- (void)requestDealsNearLocationForNotification:(NSNotification *)notification;

@end
```

Task 3

Task: Configure the Deals Table View Controller to load deals near the device location.

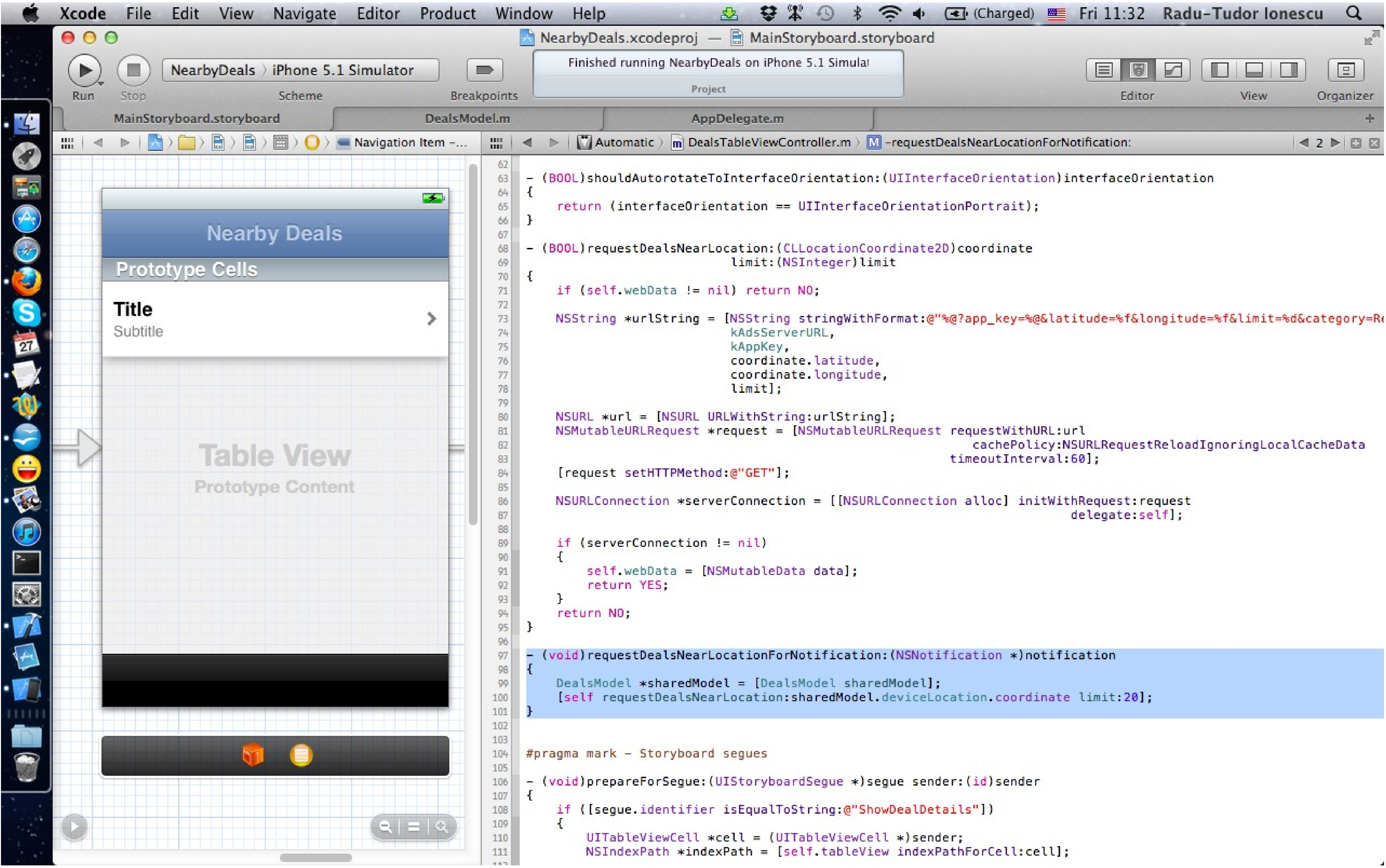
6. Select DealsTableViewController.m implementation file in Assistant Editor.

7. Implement `requestDealsNearLocationForNotification:`. This method should get the `deviceLocation` from the `sharedModel` of our application.

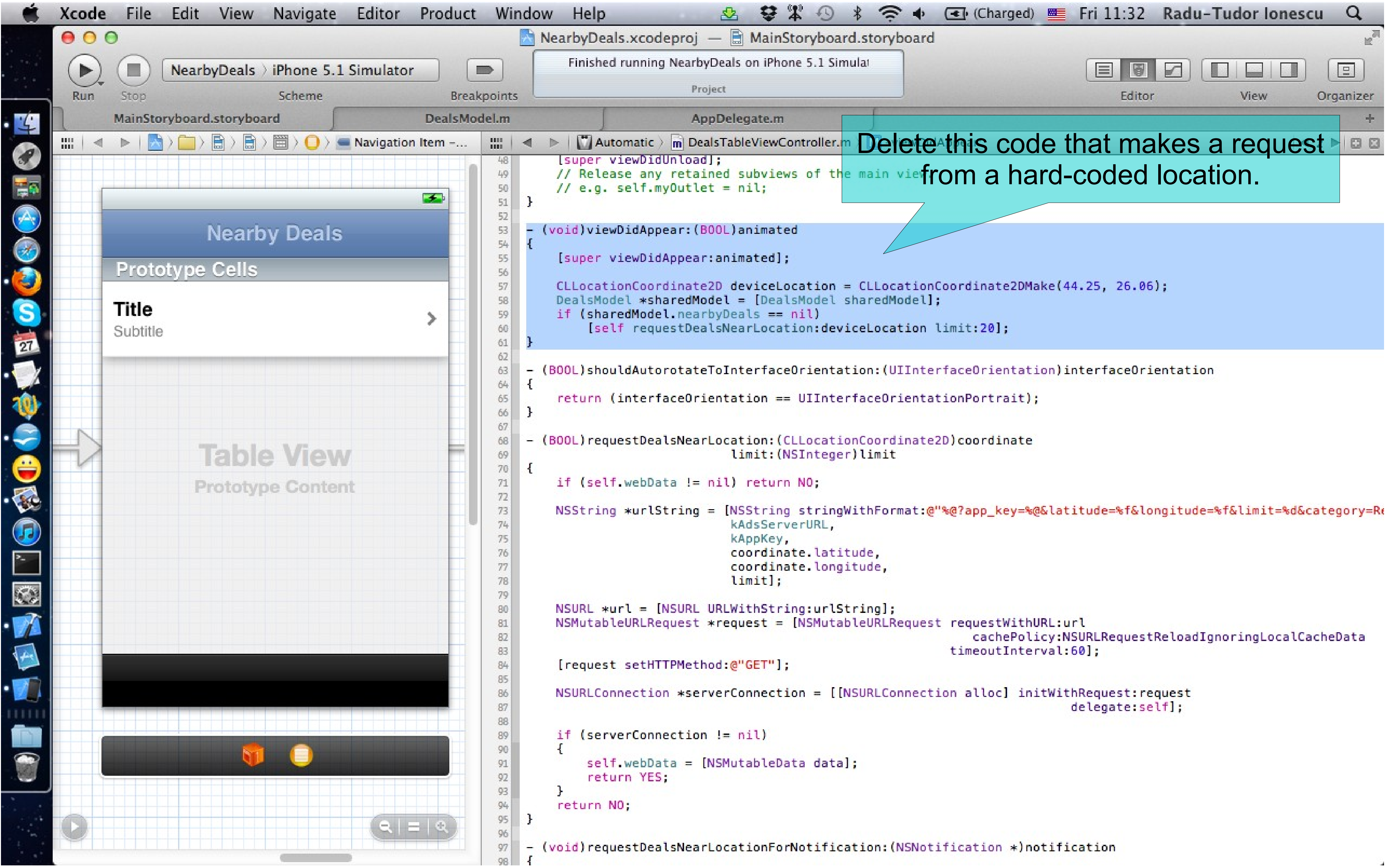
Then, it should send the `requestDealsNearLocation:limit:` message to `self` using the latitude and longitude provided by our our `sharedModel`.

8. The `requestDealsNearLocation:limit:` message from the `viewDidAppear:` method is no longer necessary. We can delete the `viewDidAppear:` implementation.

Look over the next slides for help.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 11:32 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
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
Navigation Item -...
Automatic › DealsTableViewController.m › -requestDealsNearLocationForNotification:
Nearby Deals
Prototype Cells
Title
Subtitle
Table View
Prototype Content

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

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit
{
    if (self.webData != nil) return NO;

    NSString *urlString = [NSString stringWithFormat:@"%@?app_key=%@&latitude=%f&longitude=%f&limit=%d&category=Re
                           kAdsServerURL,
                           kAppKey,
                           coordinate.latitude,
                           coordinate.longitude,
                           limit];

    NSURL *url = [NSURL URLWithString:urlString];
    NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url
                                                           cachePolicy:NSURLRequestReloadIgnoringLocalCacheData
                                                       timeoutInterval:60];
    [request setHTTPMethod:@"GET"];

    NSURLConnection *serverConnection = [[NSURLConnection alloc] initWithRequest:request
                                                                        delegate:self];

    if (serverConnection != nil)
    {
        self.webData = [NSMutableData data];
        return YES;
    }
    return NO;
}

- (void)requestDealsNearLocationForNotification:(NSNotification *)notification
{
    DealsModel *sharedModel = [DealsModel sharedModel];
    [self requestDealsNearLocation:sharedModel.deviceLocation.coordinate limit:20];
}


#pragma mark - Storyboard segues

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"ShowDealDetails"])
    {
        UITableViewCell *cell = (UITableViewCell *)sender;
        NSIndexPath *indexPath = [self.tableView indexPathForCell:cell];
```


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 11:32 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
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
Navigation Item -...
Automatic
DealsTableViewController.m
Nearby Deals
Prototype Cells
Title
Subtitle
Table View
Prototype Content

Delete this code that makes a request from a hard-coded location.

~~~
    [super viewDidUnload];
    // Release any retained subviews of the main vie
    // e.g. self.myOutlet = nil;
}

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];

    CLLocationCoordinate2D deviceLocation = CLLocationCoordinate2DMake(44.25, 26.06);
    DealsModel *sharedModel = [DealsModel sharedModel];
    if (sharedModel.nearbyDeals == nil)
        [self requestDealsNearLocation:deviceLocation limit:20];
}

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

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit
{
    if (self.webData != nil) return NO;

    NSString *urlString = [NSString stringWithFormat:@"%@?app_key=%@&latitude=%f&longitude=%f&limit=%d&category=Re
                           kAdsServerURL,
                           kAppKey,
                           coordinate.latitude,
                           coordinate.longitude,
                           limit];

    NSURL *url = [NSURL URLWithString:urlString];
    NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url
                                                           cachePolicy:NSURLRequestReloadIgnoringLocalCacheData
                                                       timeoutInterval:60];
    [request setHTTPMethod:@"GET"];

    NSURLConnection *serverConnection = [[NSURLConnection alloc] initWithRequest:request
                                                                        delegate:self];

    if (serverConnection != nil)
    {
        self.webData = [NSMutableData data];
        return YES;
    }
    return NO;
}

- (void)requestDealsNearLocationForNotification:(NSNotification *)notification
{
```]
~~~

Task 3

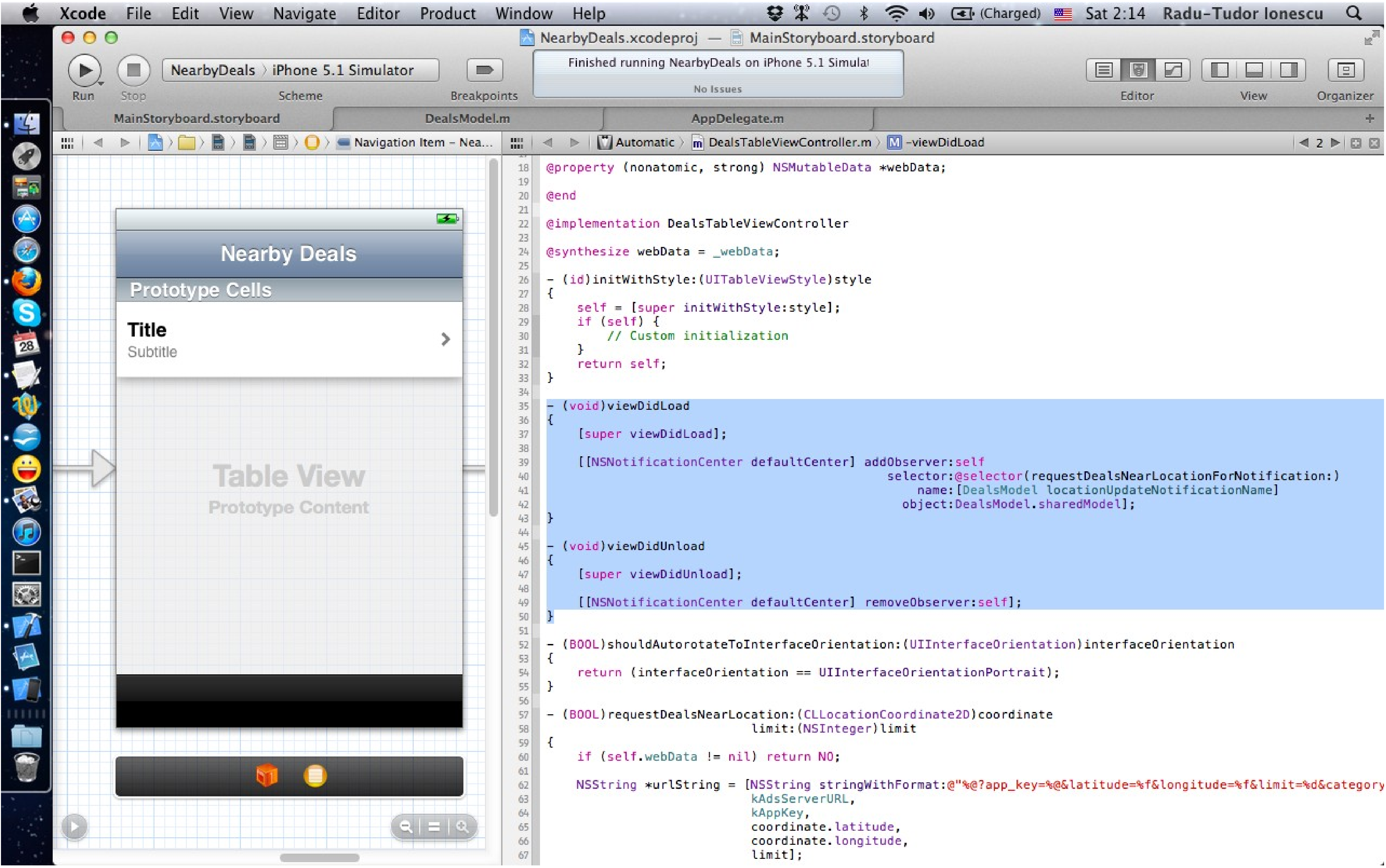
Task: Configure the Deals Table View Controller to load deals near the device location.

9. There is one more thing to do: to add the Deals Table View Controller as an observer for the `deviceLocation` object of our `sharedModel`.

As soon as the `viewDidLoads`, we can register the observer by sending the `addObserver:selector:name:object:` message to the default `NSNotificationCenter`.

10. It is our job to remove the observer before it gets deallocated. We will send the `removeObserver:` message to the `defaultCenter` in the `viewDidUnload` method.

Look over the next slide to see how to add implementation for these two methods.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Sat 2:14 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints
NearbyDeals › iPhone 5.1 Simulator
Finished running NearbyDeals on iPhone 5.1 Simula
No Issues
Editor View Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
Navigation Item – Nea...
Automatic › DealsTableViewController.m › -viewDidLoad
Nearby Deals
Prototype Cells
Title
Subtitle
Table View
Prototype Content

```
@property (nonatomic, strong) NSMutableData *webData;

@end

@implementation DealsTableViewController

@synthesize webData = _webData;

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

- (void)viewDidLoad
{
    [super viewDidLoad];

    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(requestDealsNearLocationForNotification:)
                                                 name:[DealsModel locationUpdateNotificationName]
                                               object:DealsModel.sharedModel];
}

- (void)viewDidUnload
{
    [super viewDidUnload];

    [[NSNotificationCenter defaultCenter] removeObserver:self];
}

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

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit
{
    if (self.webData != nil) return NO;

    NSString *urlString = [NSString stringWithFormat:@"%@?app_key=%@&latitude=%f&longitude=%f&limit=%d&category
                           kAdsServerURL,
                           kAppKey,
                           coordinate.latitude,
                           coordinate.longitude,
                           limit];
```

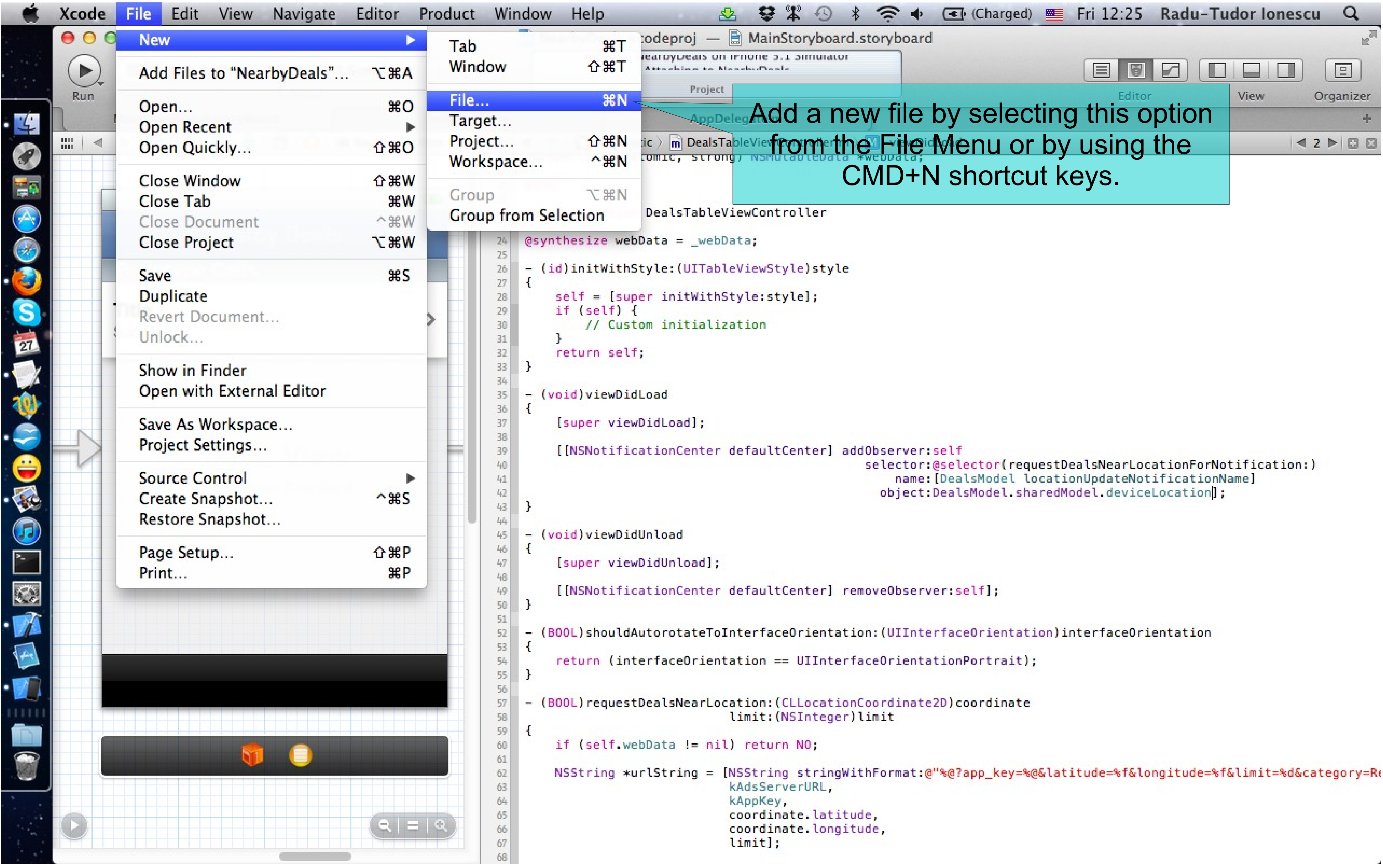
Task 3

Task: Configure the Deals Table View Controller to load deals near the device location.

11. In order to simulate a device location near Bucharest in iOS Simulator we are going to add a GPX file to our Project.

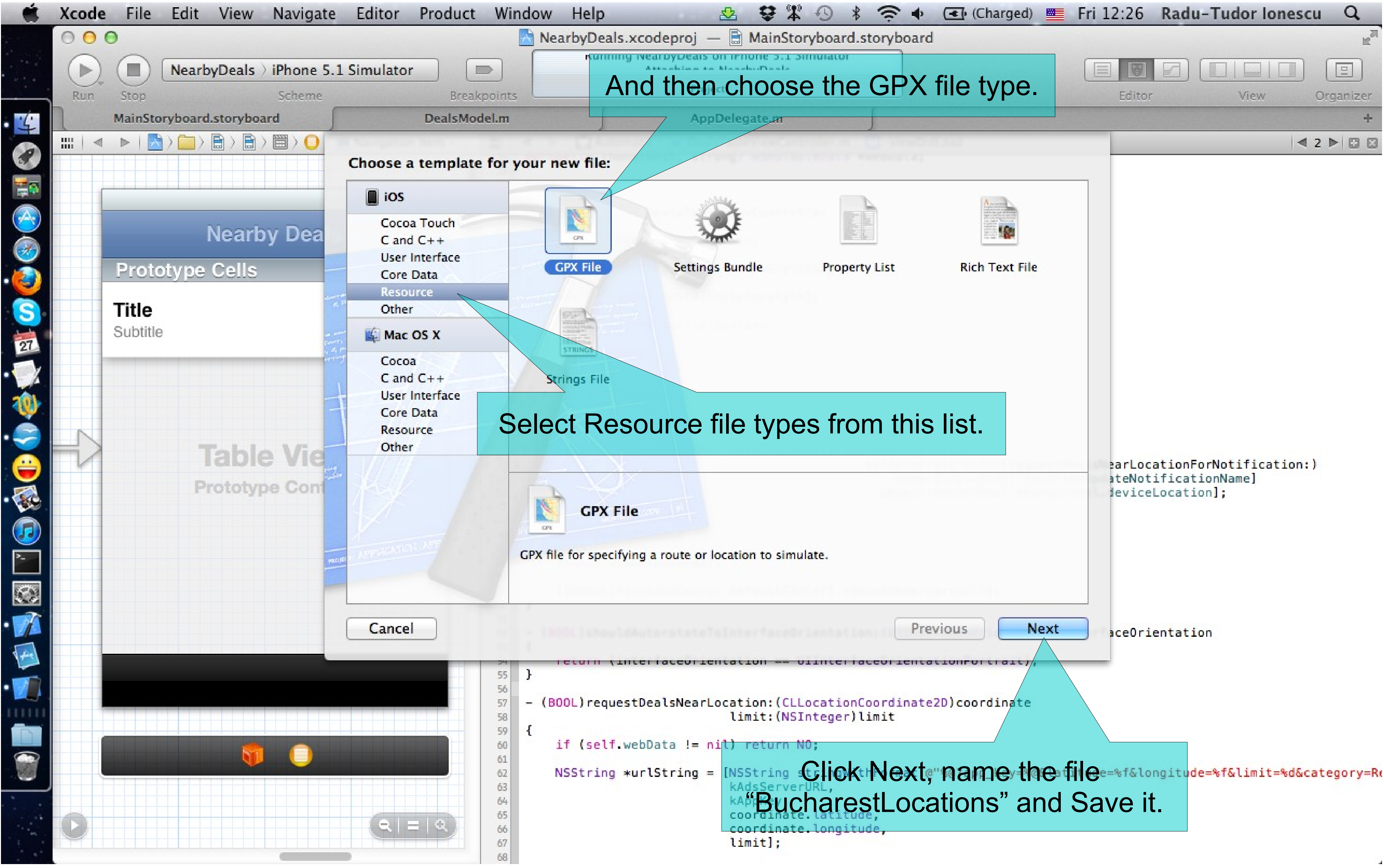
The GPX file will contain a few locations from Bucharest.

Follow the steps from the next slides to create this file and add it to our Project.



Add a new file by selecting this option from the File Menu or by using the CMD+N shortcut keys.

```
omic, strong) NSMutableData *webData;

DealsTableViewController
@synthesize webData = _webData;

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

- (void)viewDidLoad
{
    [super viewDidLoad];

    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(requestDealsNearLocationForNotification:)
                                                 name:[DealsModel locationUpdateNotificationName]
                                               object:DealsModel.sharedModel.deviceLocation];

}

- (void)viewDidUnload
{
    [super viewDidUnload];

    [[NSNotificationCenter defaultCenter] removeObserver:self];
}

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

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit
{
    if (self.webData != nil) return NO;

    NSString *urlString = [NSString stringWithFormat:@"%@?app_key=%@&latitude=%f&longitude=%f&limit=%d&category=Re
                           kAdsServerURL,
                           kAppKey,
                           coordinate.latitude,
                           coordinate.longitude,
                           limit];
```

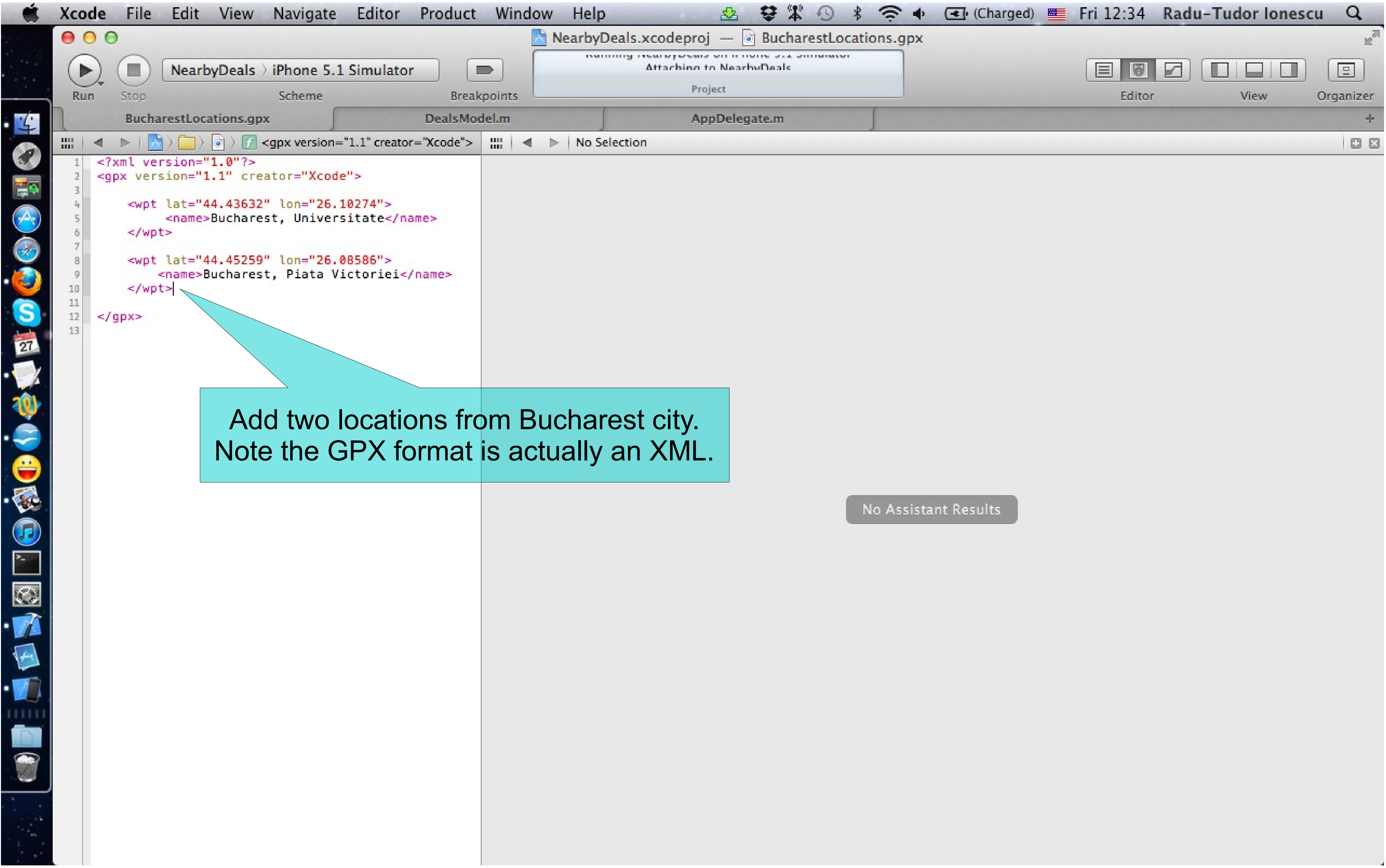


And then choose the GPX file type.

Select Resource file types from this list.

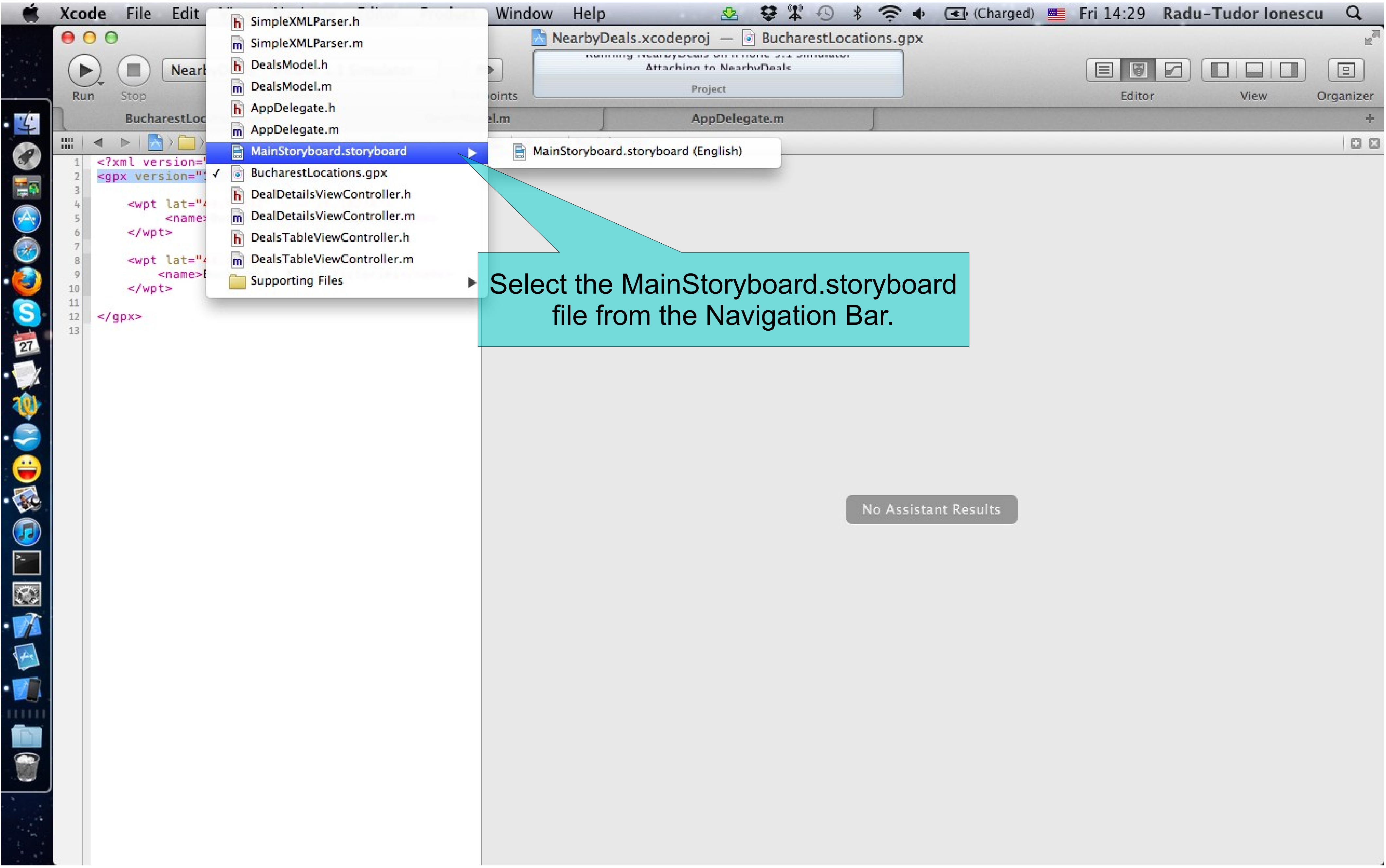
Click Next, name the file "BucharestLocations" and Save it.

```
- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit
{
    if (self.webData != nil) return NO;

    NSString *urlString = [NSString st
                           kAdsServerURL,
                           kApp
                           coordinate.latitude,
                           coordinate.longitude,
                           limit];
```



```
<?xml version="1.0"?>
<gpx version="1.1" creator="Xcode">

    <wpt lat="44.43632" lon="26.10274">
         <name>Bucharest, Universitate</name>
    </wpt>

    <wpt lat="44.45259" lon="26.08586">
        <name>Bucharest, Piata Victoriei</name>
    </wpt>

</gpx>
```

Add two locations from Bucharest city.
Note the GPX format is actually an XML.


SimpleXMLParser.h
SimpleXMLParser.m
DealsModel.h
DealsModel.m
AppDelegate.h
AppDelegate.m
MainStoryboard.storyboard
BucharestLocations.gpx
DealDetailsViewController.h
DealDetailsViewController.m
DealsTableViewController.h
DealsTableViewController.m
Supporting Files
MainStoryboard.storyboard (English)
NearbyDeals.xcodeproj — BucharestLocations.gpx
No Assistant Results

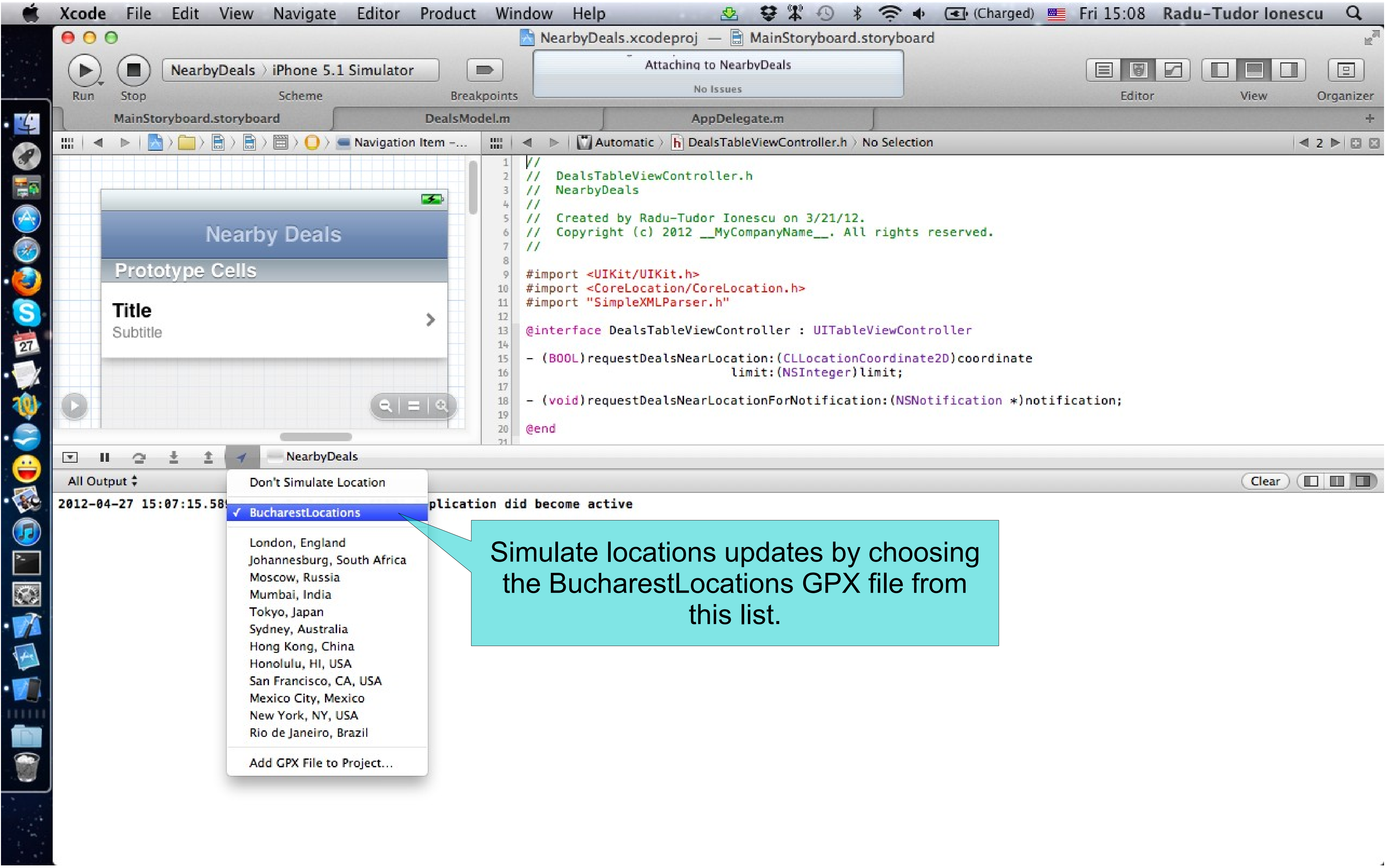
Select the MainStoryboard.storyboard file from the Navigation Bar.

Task 3

Task: Configure the Deals Table View Controller to load deals near the device location.

12. Run the application in iOS Simulator.

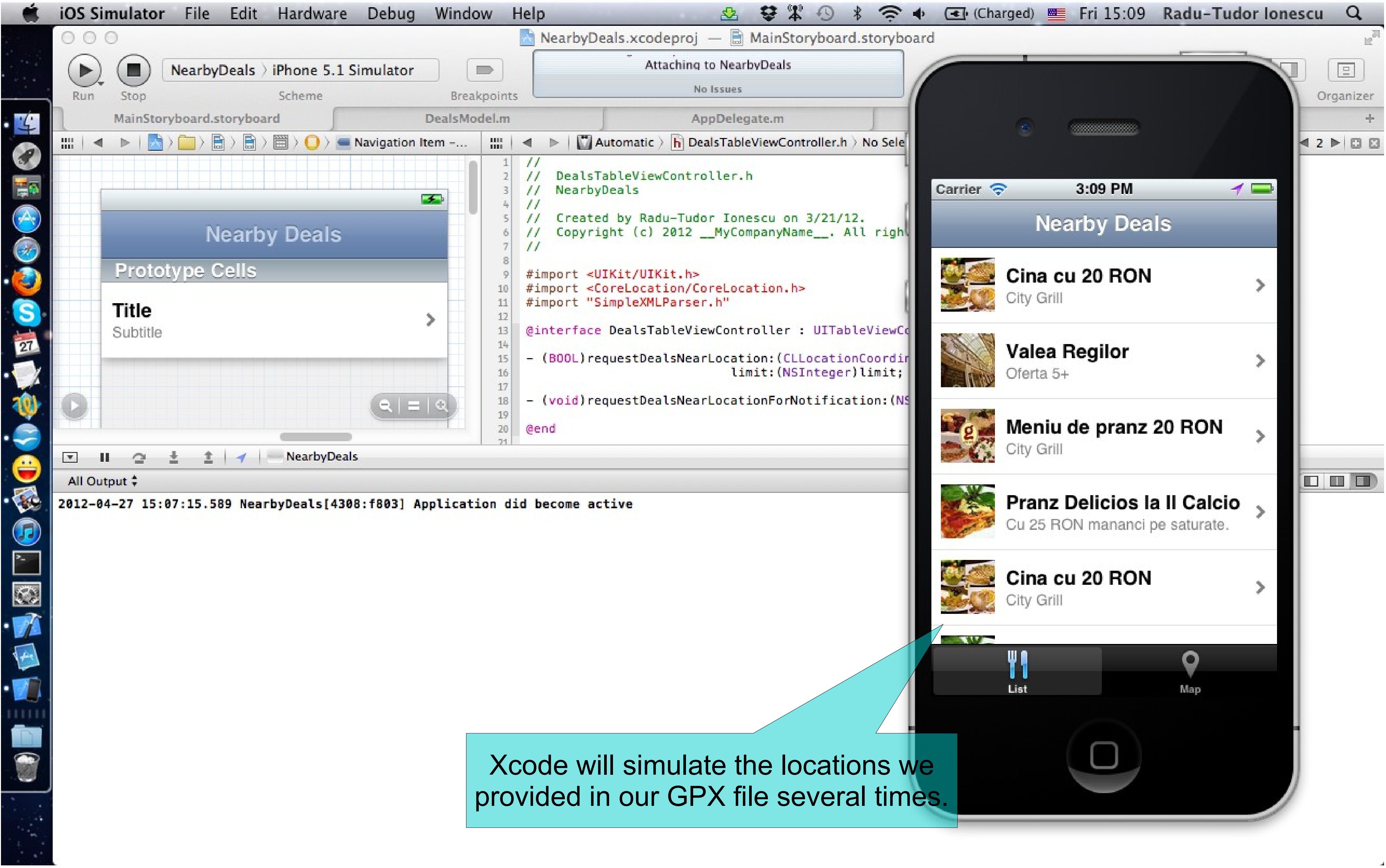
Follow the steps from the next slides to test location updates.



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

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>
#import "SimpleXMLParser.h"

@interface DealsTableViewController : UITableViewController

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit;

- (void)requestDealsNearLocationForNotification:(NSNotification *)notification;

@end
```

```
2012-04-27 15:07:15.58  plication did become active
```

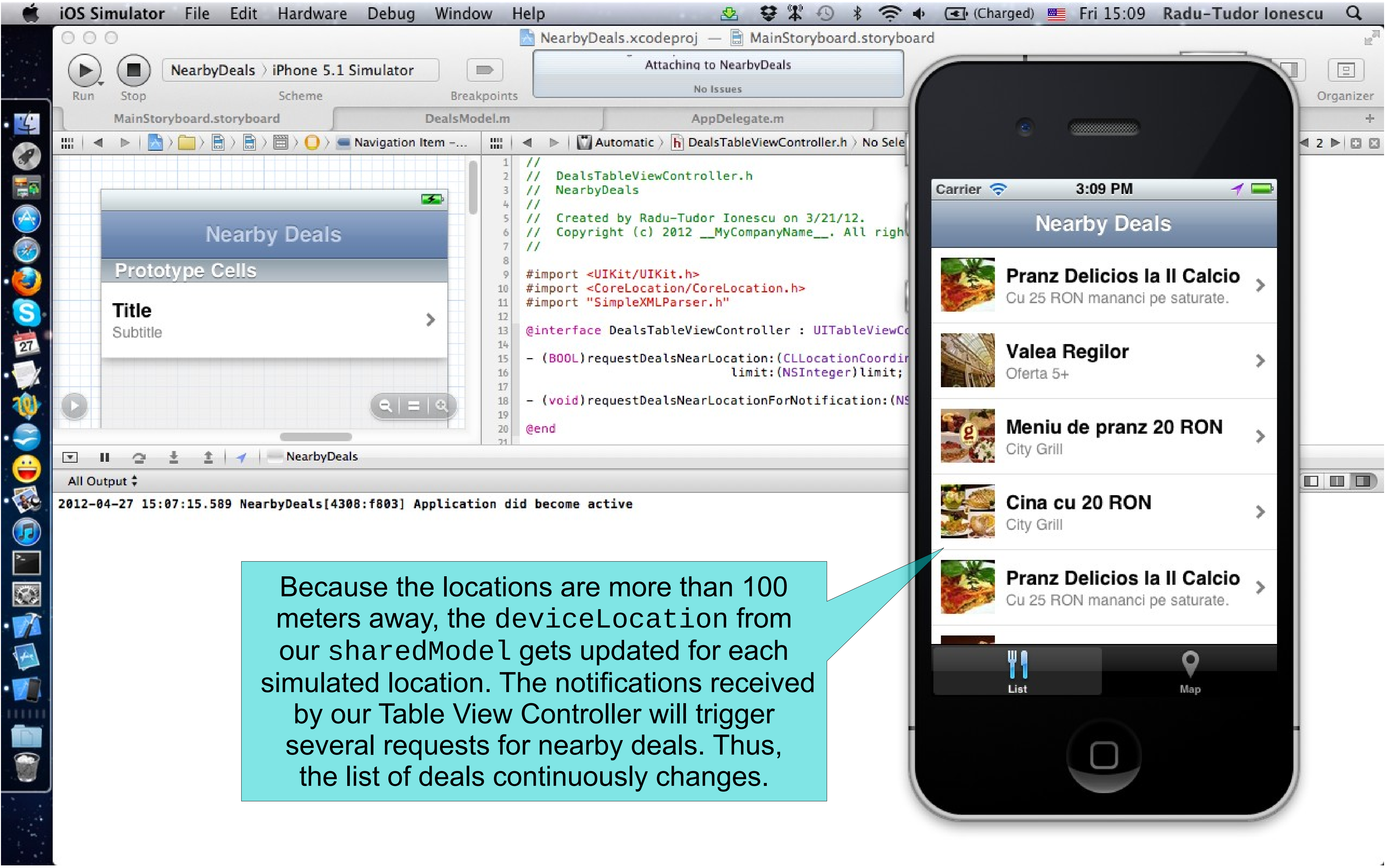
Simulate locations updates by choosing the BucharestLocations GPX file from this list.


iOS Simulator File Edit Hardware Debug Window Help
Fri 15:09 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints Organizer
NearbyDeals iPhone 5.1 Simulator
Attaching to NearbyDeals
No Issues
MainStoryboard.storyboard DealsModel.m AppDelegate.m
Navigation Item
Automatic DealsTableViewController.h No Sele
Nearby Deals
Prototype Cells
Title
Subtitle
All Output
Carrier 3:09 PM
Nearby Deals
Cina cu 20 RON
City Grill
Valea Regilor
Oferta 5+
Meniu de pranz 20 RON
City Grill
Pranz Delicios la Il Calcio
Cu 25 RON mananci pe saturate.
Cina cu 20 RON
City Grill
List
Map

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

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>
#import "SimpleXMLParser.h"

@interface DealsTableViewController : UITableViewCo

- (BOOL)requestDealsNearLocation:(CLLocationCoordir
                           limit:(NSInteger)limit;

- (void)requestDealsNearLocationForNotification:(NS

@end
```

```
2012-04-27 15:07:15.589 NearbyDeals[4308:f803] Application did become active
```

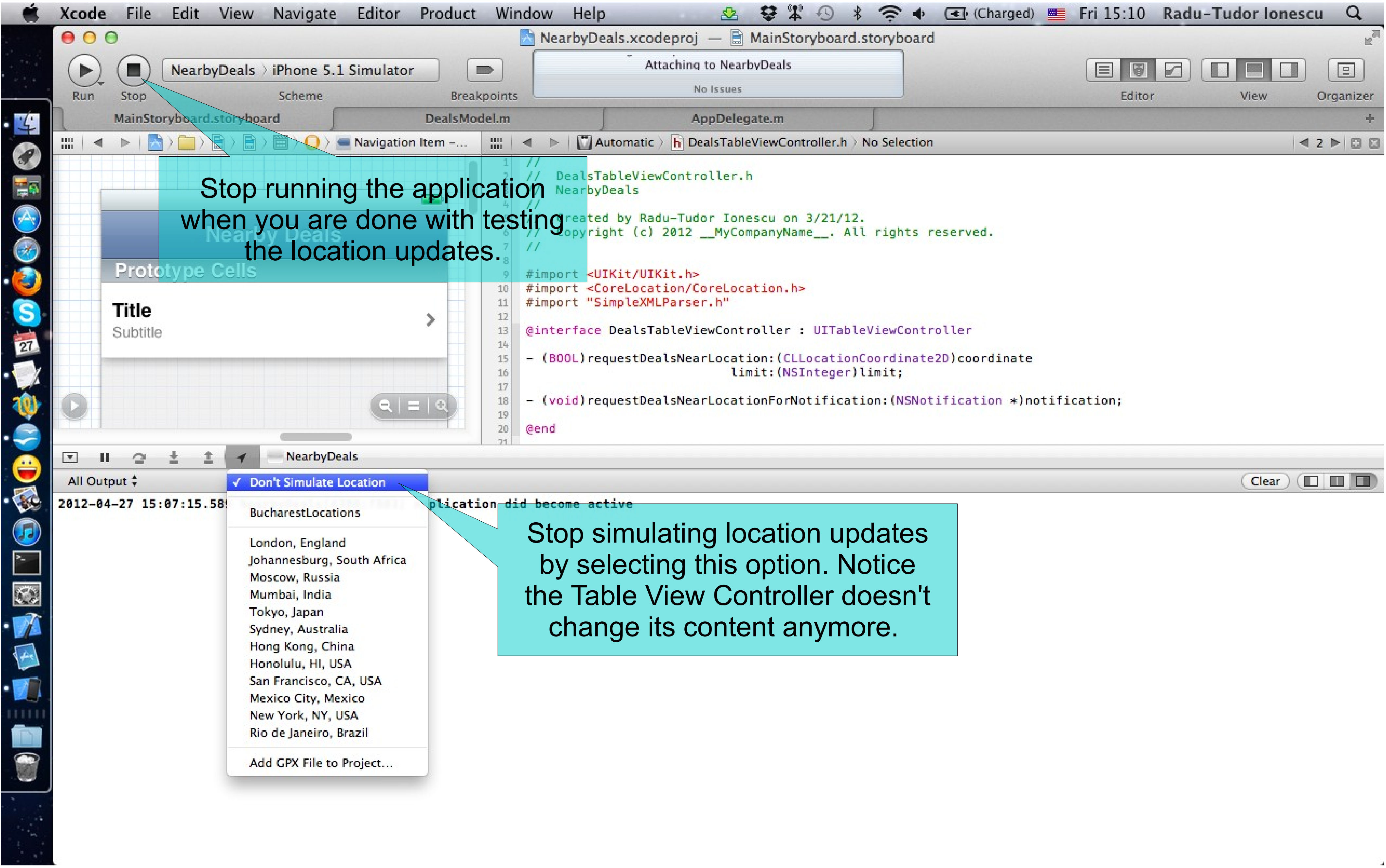
Xcode will simulate the locations we provided in our GPX file several times.


iOS Simulator File Edit Hardware Debug Window Help
Fri 15:09 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop NearbyDeals iPhone 5.1 Simulator Scheme Breakpoints
Attaching to NearbyDeals
No Issues
Organizer
MainStoryboard.storyboard DealsModel.m AppDelegate.m
Navigation Item -...
Automatic DealsTableViewController.h No Sele
Nearby Deals
Prototype Cells
Title
Subtitle
NearbyDeals
All Output
Carrier 3:09 PM
Nearby Deals
Pranz Delicios la Il Calcio
Cu 25 RON mananci pe saturate.
Valea Regilor
Oferta 5+
Meniu de pranz 20 RON
City Grill
Cina cu 20 RON
City Grill
Pranz Delicios la Il Calcio
Cu 25 RON mananci pe saturate.
List
Map

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

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>
#import "SimpleXMLParser.h"

@interface DealsTableViewController : UITableViewCo

- (BOOL)requestDealsNearLocation:(CLLocationCoordir
                          limit:(NSInteger)limit;

- (void)requestDealsNearLocationForNotification:(NS

@end
```

```
2012-04-27 15:07:15.589 NearbyDeals[4308:f803] Application did become active
```

Because the locations are more than 100 meters away, the `deviceLocation` from our `sharedModel` gets updated for each simulated location. The notifications received by our Table View Controller will trigger several requests for nearby deals. Thus, the list of deals continuously changes.



Stop running the application when you are done with testing the location updates.

Stop simulating location updates by selecting this option. Notice the Table View Controller doesn't change its content anymore.

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

#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>
#import "SimpleXMLParser.h"

@interface DealsTableViewController : UITableViewController

- (BOOL)requestDealsNearLocation:(CLLocationCoordinate2D)coordinate
                           limit:(NSInteger)limit;

- (void)requestDealsNearLocationForNotification:(NSNotification *)notification;

@end
```

```
2012-04-27 15:07:15.58 ... plication did become active
```

Task 4

Task: Add the Map Kit framework and create the map view.

1. The Deals Table View Controller is almost done. We are going to focus on creating the Map View of our application. The Map View will be annotated with pins for each nearby deal.

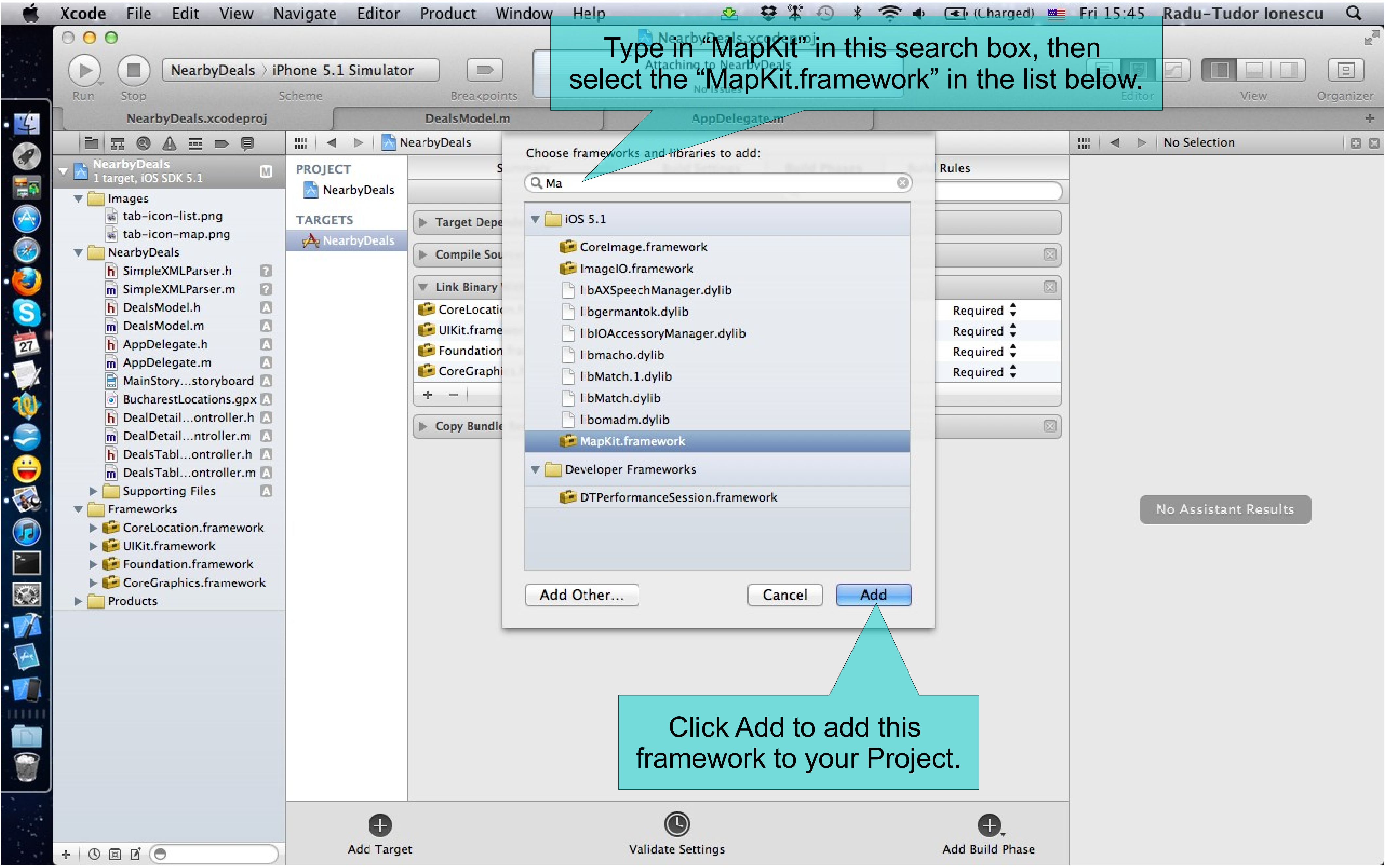
When the user selects a pin it will display a callout with deal information and a disclosure button to access deal details. The deal details will be presented in a Deal Details View Controller (we already have this View Controller).

Open Project Navigator and select the Project itself.

2. Select the Target application and make sure you are on the “Build Phases” tab.

3. Expand “Link Binary With Libraries” and click the “+” button to add a new library.

Continue with the steps from next slides.


Type in "MapKit" in this search box, then select the "MapKit.framework" in the list below.

Click Add to add this framework to your Project.

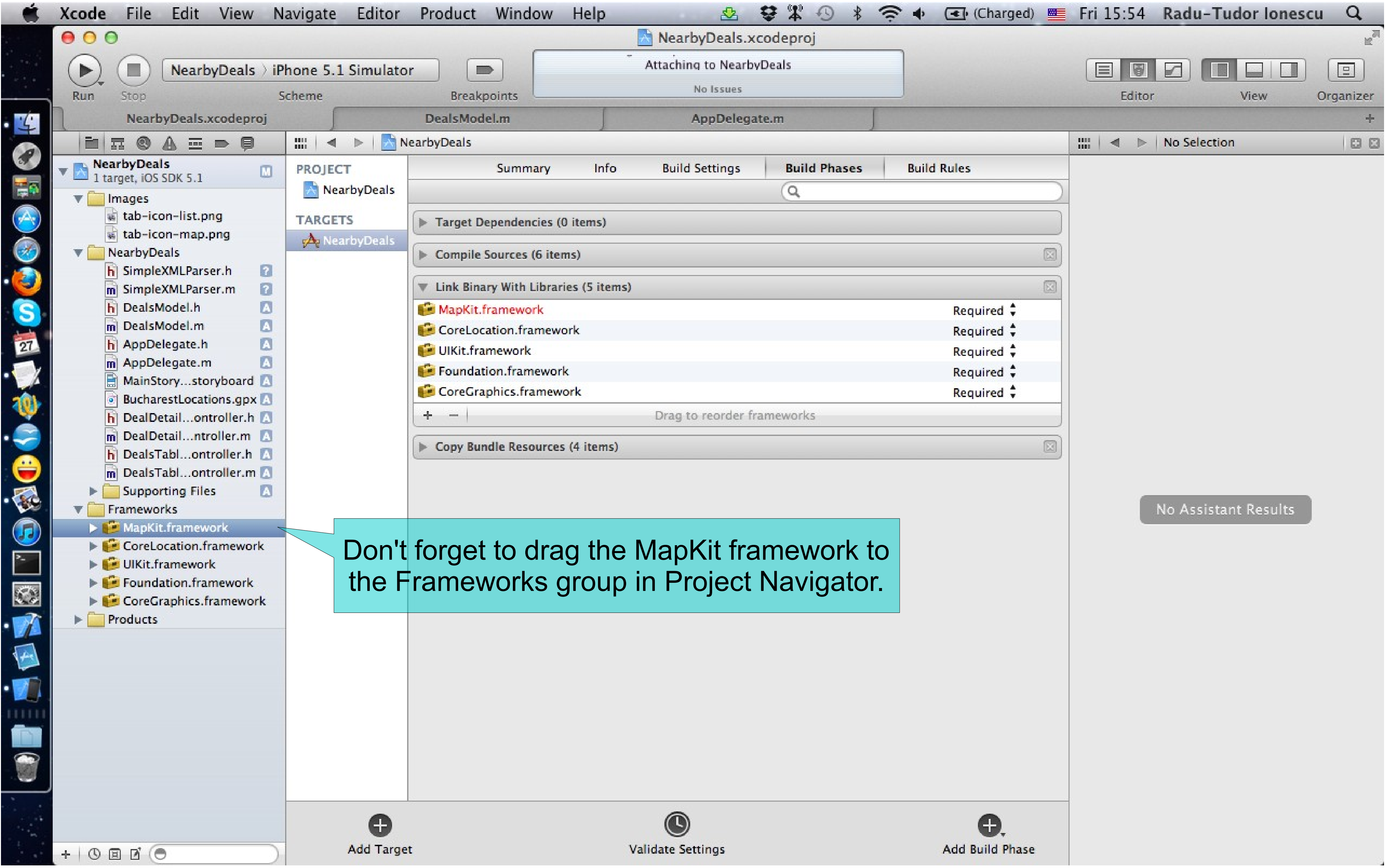
Choose frameworks and libraries to add:

Ma

iOS 5.1
- CoreImage.framework
- ImageIO.framework
- libAXSpeechManager.dylib
- libgermantok.dylib
- libIOAccessoryManager.dylib
- libmacho.dylib
- libMatch.1.dylib
- libMatch.dylib
- libomadm.dylib
- MapKit.framework

Developer Frameworks
- DTPerformanceSession.framework

Add Other...
Cancel
Add
Xcode File Edit View Navigate Editor Product Window Help
Fri 15:45 Radu-Tudor Ionescu
NearbyDeals › iPhone 5.1 Simulator
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals.xcodeproj DealsModel.m AppDelegate.m
PROJECT NearbyDeals TARGETS NearbyDeals
Target Dependencies
Compile Sources
Link Binary With Libraries
Copy Bundle Resources
Required
No Selection
No Assistant Results
Add Target Validate Settings Add Build Phase



Project navigator:

- NearbyDeals (1 target, iOS SDK 5.1)
  - Images
    - tab-icon-list.png
    - tab-icon-map.png
  - NearbyDeals
    - SimpleXMLParser.h
    - SimpleXMLParser.m
    - DealsModel.h
    - DealsModel.m
    - AppDelegate.h
    - AppDelegate.m
    - MainStory...storyboard
    - BucharestLocations.gpx
    - DealDetail...ontroller.h
    - DealDetail...ntroller.m
    - DealsTabl...ontroller.h
    - DealsTabl...ontroller.m
    - Supporting Files
  - Frameworks
    - MapKit.framework
    - CoreLocation.framework
    - UIKit.framework
    - Foundation.framework
    - CoreGraphics.framework
  - Products

PROJECT: NearbyDeals
TARGETS: NearbyDeals

Summary | Info | Build Settings | **Build Phases** | Build Rules

- Target Dependencies (0 items)
- Compile Sources (6 items)
- Link Binary With Libraries (5 items)

MapKit.framework	Required
CoreLocation.framework	Required
UIKit.framework	Required
Foundation.framework	Required
CoreGraphics.framework	Required

Drag to reorder frameworks

- Copy Bundle Resources (4 items)

Don't forget to drag the MapKit framework to the Frameworks group in Project Navigator.

Task 4

Task: Add the Map Kit framework and create the map view.

4. Let's add a class that conforms to the `MKAnnotation` protocol. The Map View annotations will be objects of this class.

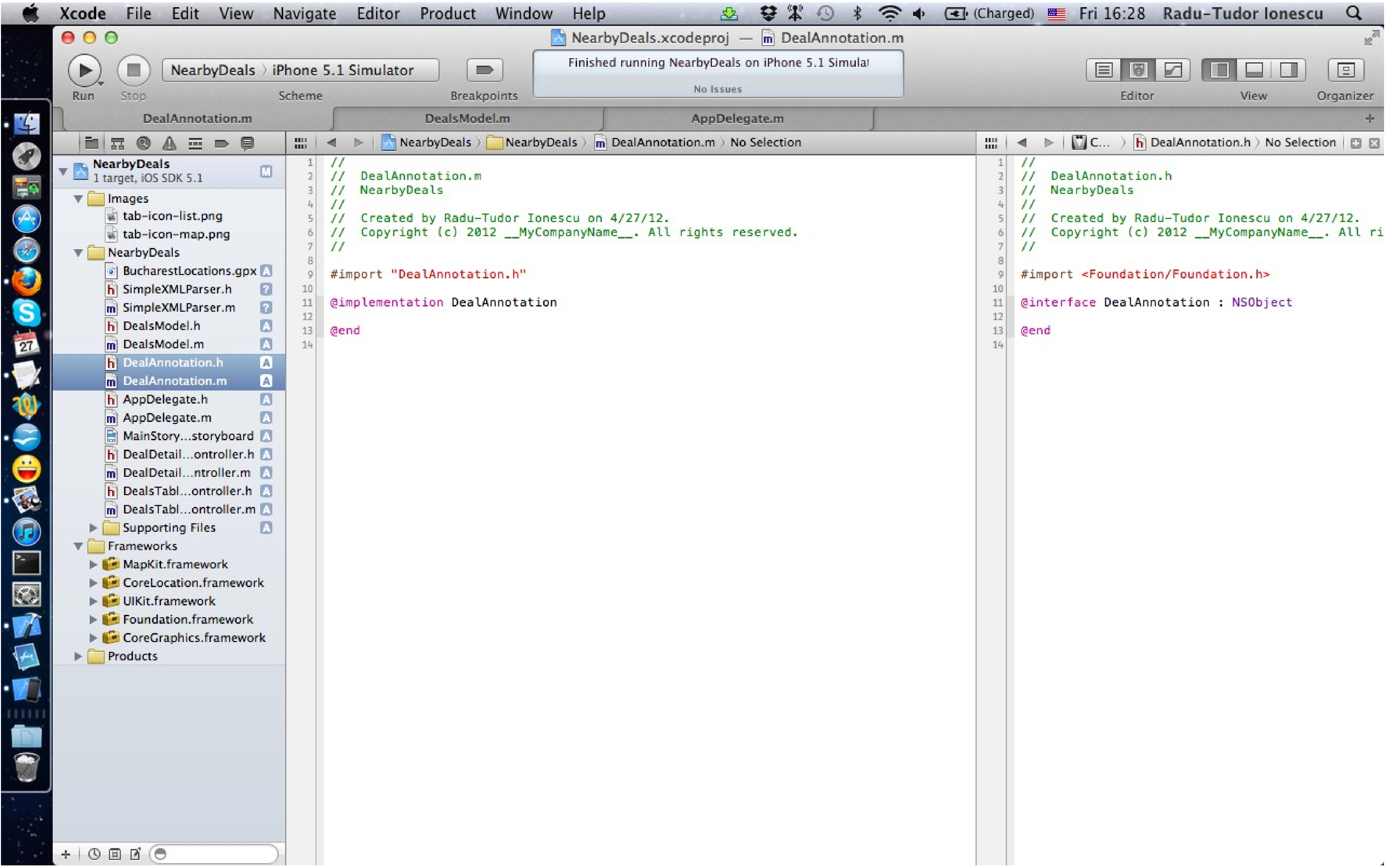
Right-click on the NearbyDeals group in Project Navigator and select the “New File...” option.

5. Choose the “Cocoa Touch > Objective-C class” template file and click Next.

6. Type in “DealAnnotation” for the class name and make it a subclass of `NSObject`.

7. Make sure the NearbyDeals subfolder in your Project folder is selected for the files location. Click Create.

8. It would be nice to organize your files in Project Navigator. Drag the DealAnnotation.h and DealAnnotation.m files above the DealsModel files. Make sure everything is set up as in the following screenshot before moving on.



```
//
//  DealAnnotation.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealAnnotation.h"

@implementation DealAnnotation

@end
```

```
//
//  DealAnnotation.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All ri
//

#import <Foundation/Foundation.h>

@interface DealAnnotation : NSObject

@end
```

Task 4

Task: Add the Map Kit framework and create the map view.

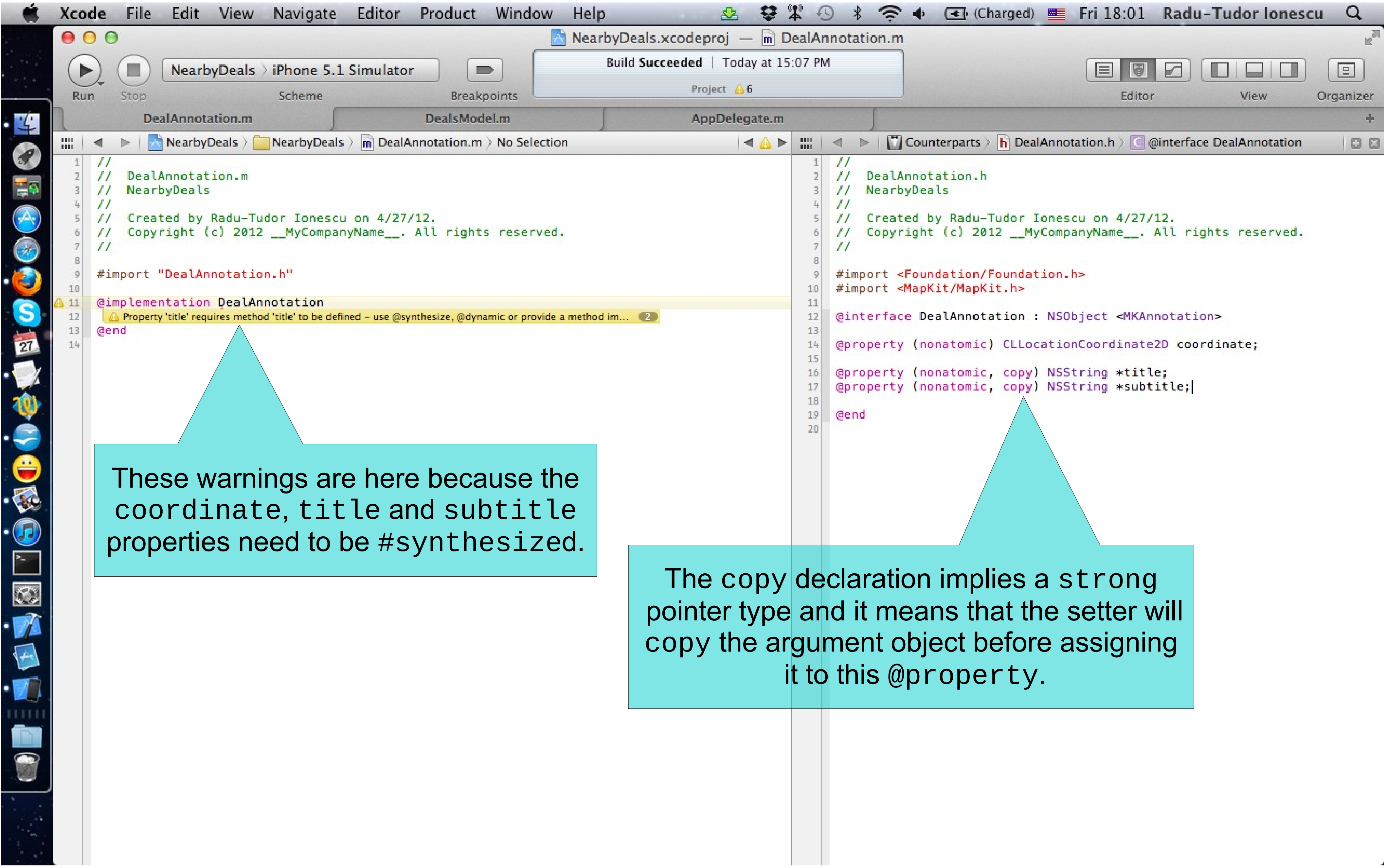
9. Next, let us implement the `DealAnnotation` class so that it conforms to the `MKAnnotation` protocol.

Hide Project Navigator to make more room.

10. The first thing to do is to `#import` the MapKit framework into our header file (the one that contains the `@interface` block) to know about the `MKAnnotation` protocol.

11. We declare the `MKAnnotation` protocol right after the superclass declaration. An object that adopts this protocol must implement the `coordinate` property. We are also going to implement the `title` and `subtitle` optional properties.

Follow the steps from the next slides to finish the `DealAnnotation` class implementation.



Xcode File Edit View Navigate Editor Product Window Help
NearbyDeals.xcodeproj — DealAnnotation.m
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Today at 15:07 PM
Project ⚠ 6
Editor
View
Organizer
DealAnnotation.m
DealsModel.m
AppDelegate.m
NearbyDeals › NearbyDeals › DealAnnotation.m › No Selection
Counterparts › DealAnnotation.h › @interface DealAnnotation

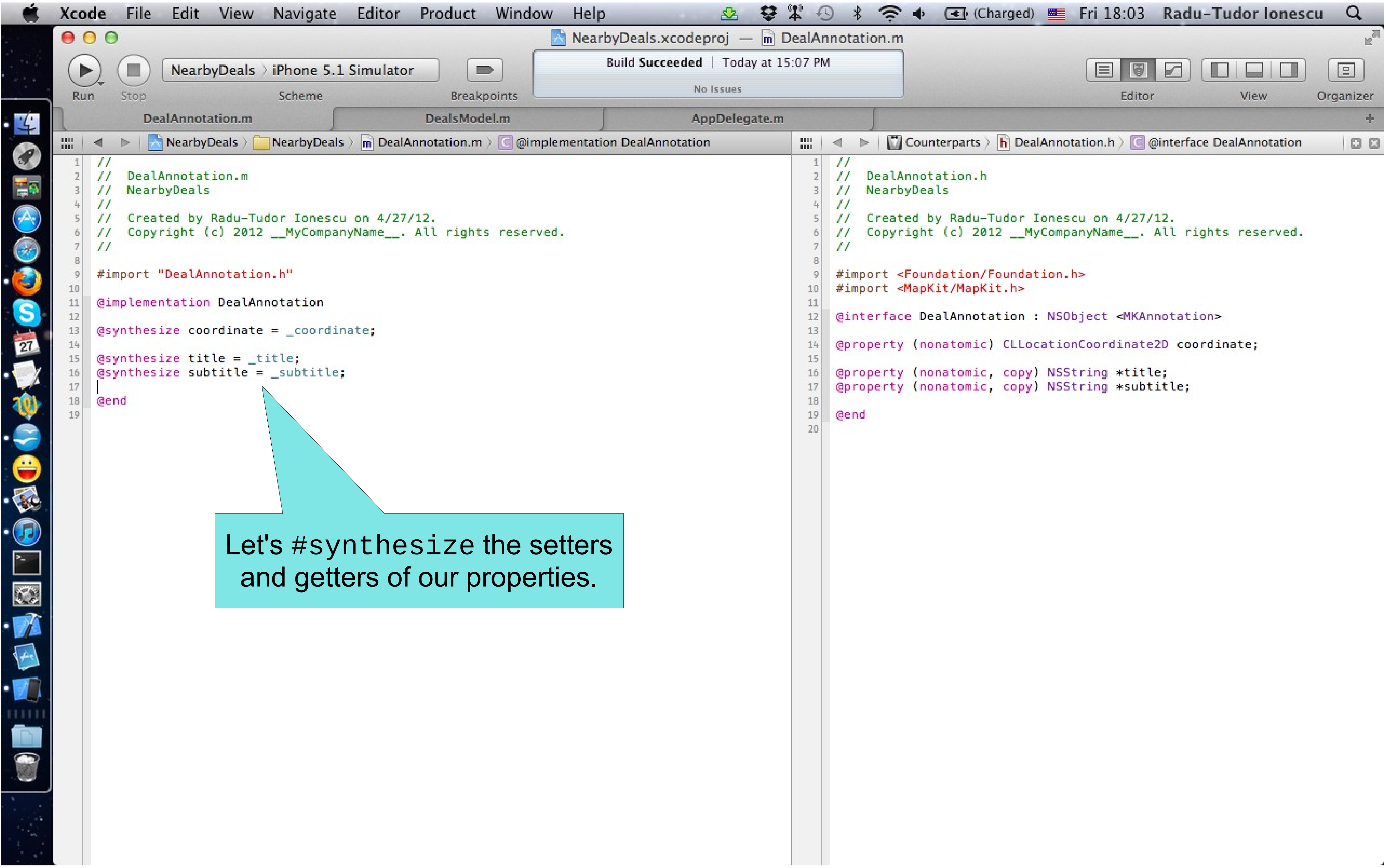
```objc
//
//  DealAnnotation.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealAnnotation.h"

@implementation DealAnnotation
@end
```

⚠ Property 'title' requires method 'title' to be defined - use @synthesize, @dynamic or provide a method im... 2

```objc
//
//  DealAnnotation.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <Foundation/Foundation.h>
#import <MapKit/MapKit.h>

@interface DealAnnotation : NSObject <MKAnnotation>

@property (nonatomic) CLLocationCoordinate2D coordinate;

@property (nonatomic, copy) NSString *title;
@property (nonatomic, copy) NSString *subtitle;

@end
```

These warnings are here because the `coordinate`, `title` and `subtitle` properties need to be `#synthesized`.

The `copy` declaration implies a `strong` pointer type and it means that the setter will copy the argument object before assigning it to this `@property`.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 18:03 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — DealAnnotation.m
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Build Succeeded | Today at 15:07 PM
No Issues
Editor
View
Organizer
DealAnnotation.m
DealsModel.m
AppDelegate.m
NearbyDeals 〉 NearbyDeals 〉 DealAnnotation.m 〉 @implementation DealAnnotation
Counterparts 〉 DealAnnotation.h 〉 @interface DealAnnotation

```objc
//
//  DealAnnotation.m
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "DealAnnotation.h"

@implementation DealAnnotation

@synthesize coordinate = _coordinate;

@synthesize title = _title;
@synthesize subtitle = _subtitle;

@end
```

```objc
//
//  DealAnnotation.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <Foundation/Foundation.h>
#import <MapKit/MapKit.h>

@interface DealAnnotation : NSObject <MKAnnotation>

@property (nonatomic) CLLocationCoordinate2D coordinate;

@property (nonatomic, copy) NSString *title;
@property (nonatomic, copy) NSString *subtitle;

@end
```

Let's `#synthesize` the setters and getters of our properties.

Task 4

Task: Add the Map Kit framework and create the map view.

12. Open Project Navigator and right-click on the NearbyDeals group to create a “New File...”.

13. Choose “Cocoa Touch > Objective-C class”. Click Next.

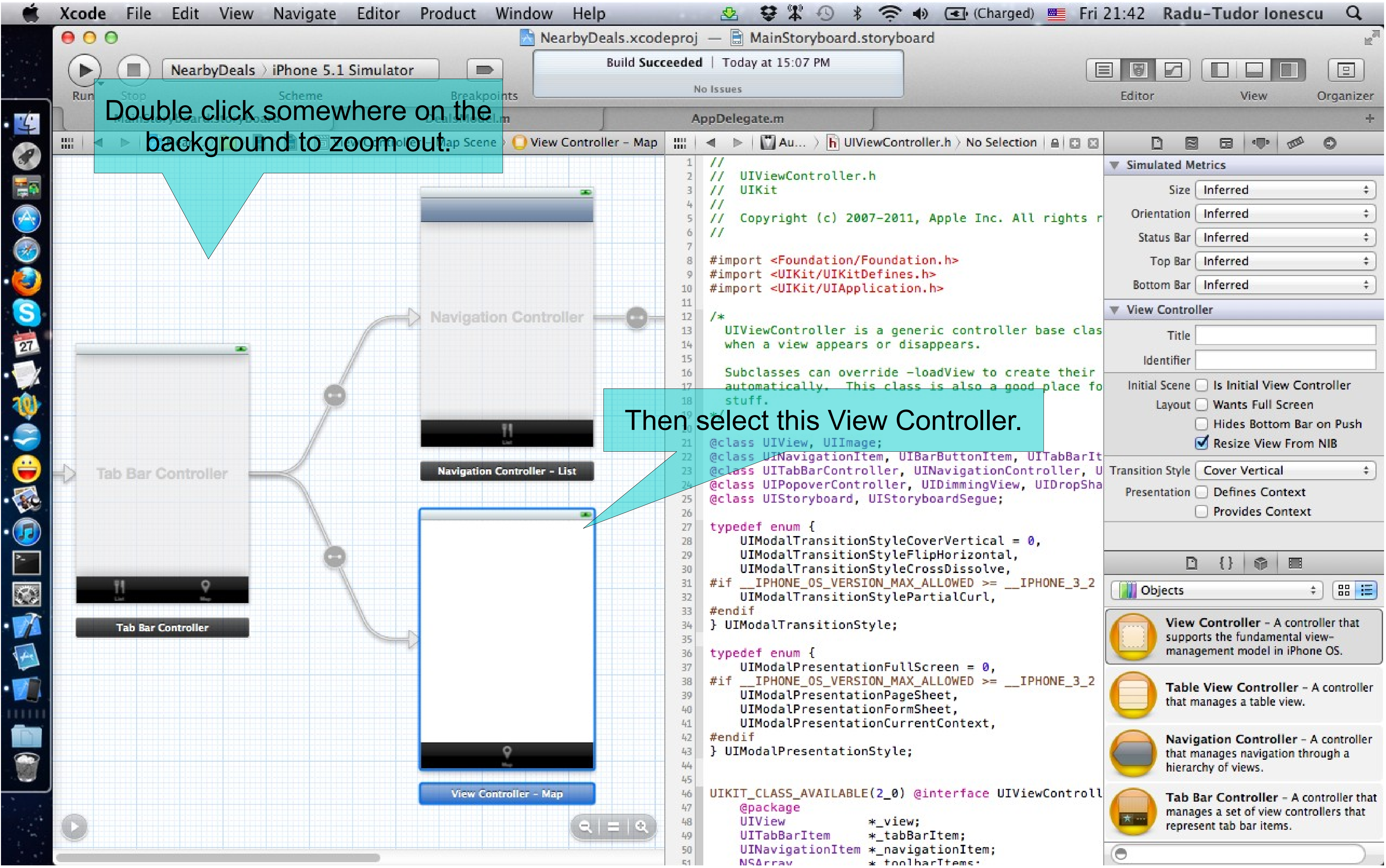
14. Name the class “MapViewController” and make it a subclass of `UIViewController`. Click Next.

15. Make sure the files location is the NearbyDeals subfolder and click Create.

16. Drag the new files right before the “Supporting Files” group in Project Navigator.

17. Select the MainStoryboard.storyboard file in Project Navigator. You can close Project Navigator to make room for what's next.

18. Open Utilities area and follow the steps from the next slides to associate the `MapViewController` class to the right View Controller in the storyboard file.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 21:42 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
NearbyDeals iPhone 5.1 Simulator
Build Succeeded | Today at 15:07 PM
No Issues
Run Stop Scheme Breakpoints
Editor View Organizer
AppDelegate.m
View Controller – Map
Au... UIViewController.h No Selection
Navigation Controller
Navigation Controller – List
Tab Bar Controller
Tab Bar Controller
View Controller – Map

Double click somewhere on the background to zoom out.

Then select this View Controller.

```
//
//  UIViewController.h
//  UIKit
//
//  Copyright (c) 2007-2011, Apple Inc. All rights r
//

#import <Foundation/Foundation.h>
#import <UIKit/UIKitDefines.h>
#import <UIKit/UIApplication.h>

/*
  UIViewController is a generic controller base clas
  when a view appears or disappears.

  Subclasses can override -loadView to create their
  automatically.  This class is also a good place fo
  stuff.
*/

@class UIView, UIImage;
@class UINavigationItem, UIBarButtonItem, UITabBarIt
@class UITabBarController, UINavigationController, U
@class UIPopoverController, UIDimmingView, UIDropSha
@class UIStoryboard, UIStoryboardSegue;

typedef enum {
    UIModalTransitionStyleCoverVertical = 0,
    UIModalTransitionStyleFlipHorizontal,
    UIModalTransitionStyleCrossDissolve,
#if __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_3_2
    UIModalTransitionStylePartialCurl,
#endif
} UIModalTransitionStyle;

typedef enum {
    UIModalPresentationFullScreen = 0,
#if __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_3_2
    UIModalPresentationPageSheet,
    UIModalPresentationFormSheet,
    UIModalPresentationCurrentContext,
#endif
} UIModalPresentationStyle;


UIKIT_CLASS_AVAILABLE(2_0) @interface UIViewControll
    @package
    UIView           *_view;
    UITabBarItem     *_tabBarItem;
    UINavigationItem *_navigationItem;
    NSArray          *_toolbarItems;
```

Simulated Metrics
Size: Inferred
Orientation: Inferred
Status Bar: Inferred
Top Bar: Inferred
Bottom Bar: Inferred

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

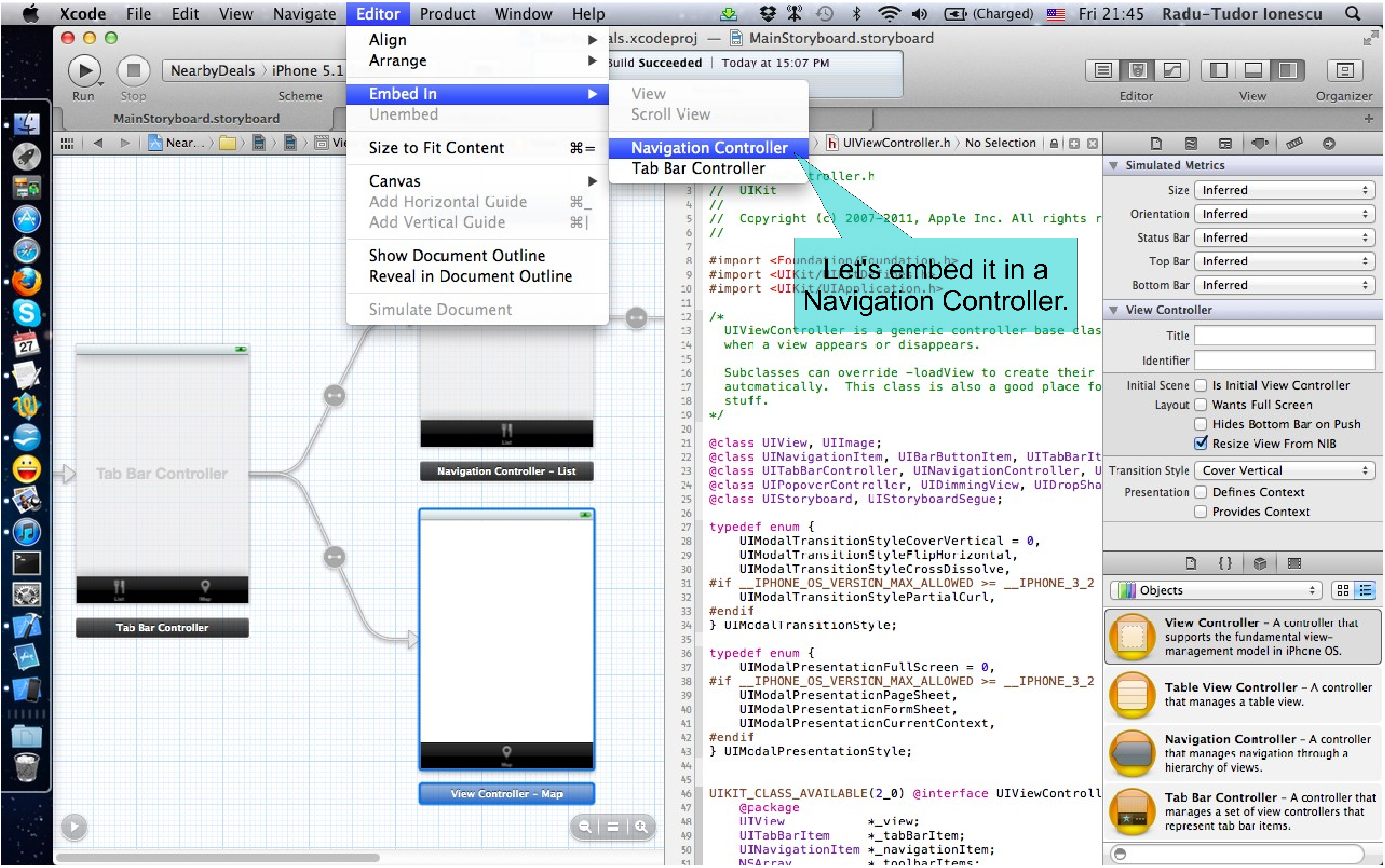
Objects

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.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

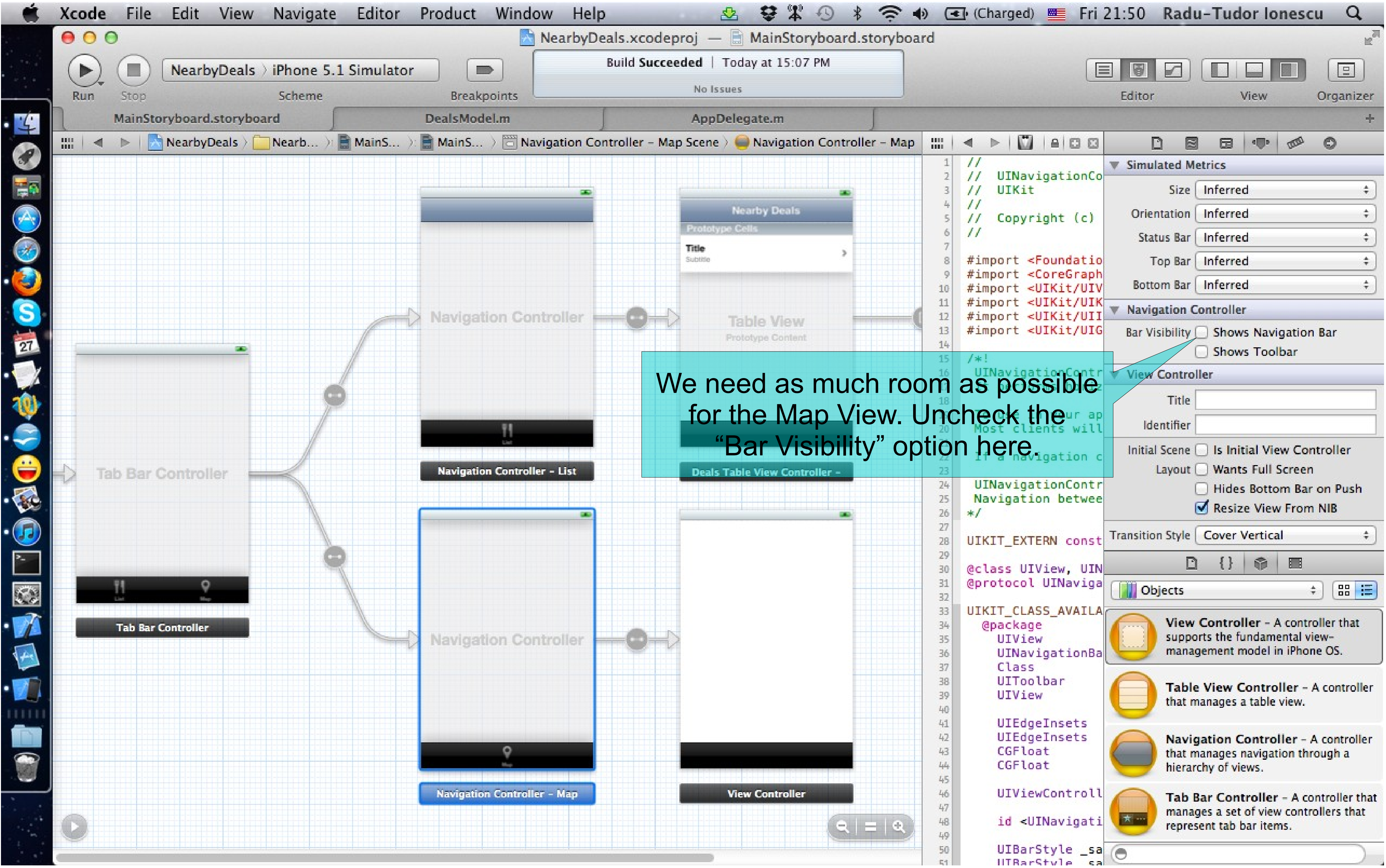

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 21:45 Radu-Tudor Ionescu
Run Stop NearbyDeals iPhone 5.1 Scheme
als.xcodeproj — MainStoryboard.storyboard
Build Succeeded | Today at 15:07 PM
Editor View Organizer
MainStoryboard.storyboard
Align
Arrange
Embed In
Unembed
Size to Fit Content ⌘=
Canvas
Add Horizontal Guide ⌘_
Add Vertical Guide ⌘|
Show Document Outline
Reveal in Document Outline
Simulate Document
View
Scroll View
Navigation Controller
Tab Bar Controller
UIViewController.h No Selection
Tab Bar Controller
Navigation Controller – List
View Controller – Map
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.
Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Let's embed it in a Navigation Controller.

```
troller.h
//  UIKit
//
//  Copyright (c) 2007-2011, Apple Inc. All rights r
//

#import <Foundation/Foundation.h>
#import <UIKit/UIKitDefines.h>
#import <UIKit/UIApplication.h>

/*
  UIViewController is a generic controller base clas
  when a view appears or disappears.

  Subclasses can override -loadView to create their
  automatically.  This class is also a good place fo
  stuff.
*/

@class UIView, UIImage;
@class UINavigationItem, UIBarButtonItem, UITabBarIt
@class UITabBarController, UINavigationController, U
@class UIPopoverController, UIDimmingView, UIDropSha
@class UIStoryboard, UIStoryboardSegue;

typedef enum {
    UIModalTransitionStyleCoverVertical = 0,
    UIModalTransitionStyleFlipHorizontal,
    UIModalTransitionStyleCrossDissolve,
#if __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_3_2
    UIModalTransitionStylePartialCurl,
#endif
} UIModalTransitionStyle;

typedef enum {
    UIModalPresentationFullScreen = 0,
#if __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_3_2
    UIModalPresentationPageSheet,
    UIModalPresentationFormSheet,
    UIModalPresentationCurrentContext,
#endif
} UIModalPresentationStyle;


UIKIT_CLASS_AVAILABLE(2_0) @interface UIViewControll
    @package
    UIView           *_view;
    UITabBarItem     *_tabBarItem;
    UINavigationItem *_navigationItem;
    NSArray          *_toolbarItems;
```

Simulated Metrics
Size: Inferred
Orientation: Inferred
Status Bar: Inferred
Top Bar: Inferred
Bottom Bar: Inferred

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


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 21:50 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme Breakpoints
Build Succeeded | Today at 15:07 PM
No Issues
Editor View Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › Nearb… › MainS… › MainS… › Navigation Controller – Map Scene › Navigation Controller – Map
Navigation Controller
Nearby Deals
Prototype Cells
Title
Subtitle
Table View
Prototype Content
Tab Bar Controller
Navigation Controller – List
Deals Table View Controller –
Navigation Controller
Navigation Controller – Map
View Controller

```
//
//  UINavigationCo
//  UIKit
//
//  Copyright (c)
//

#import <Foundatio
#import <CoreGraph
#import <UIKit/UIV
#import <UIKit/UIK
#import <UIKit/UII
#import <UIKit/UIG

/*!
 UINavigationContr

 Most clients will
 If a navigation c

 UINavigationContr
 Navigation betwee
*/

UIKIT_EXTERN const

@class UIView, UIN
@protocol UINaviga

UIKIT_CLASS_AVAILA
  @package
    UIView
    UINavigationBa
    Class
    UIToolbar
    UIView

    UIEdgeInsets
    UIEdgeInsets
    CGFloat
    CGFloat

    UIViewControll

    id <UINavigati

    UIBarStyle _sa
    UIBarStyle _sa
```

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

Objects

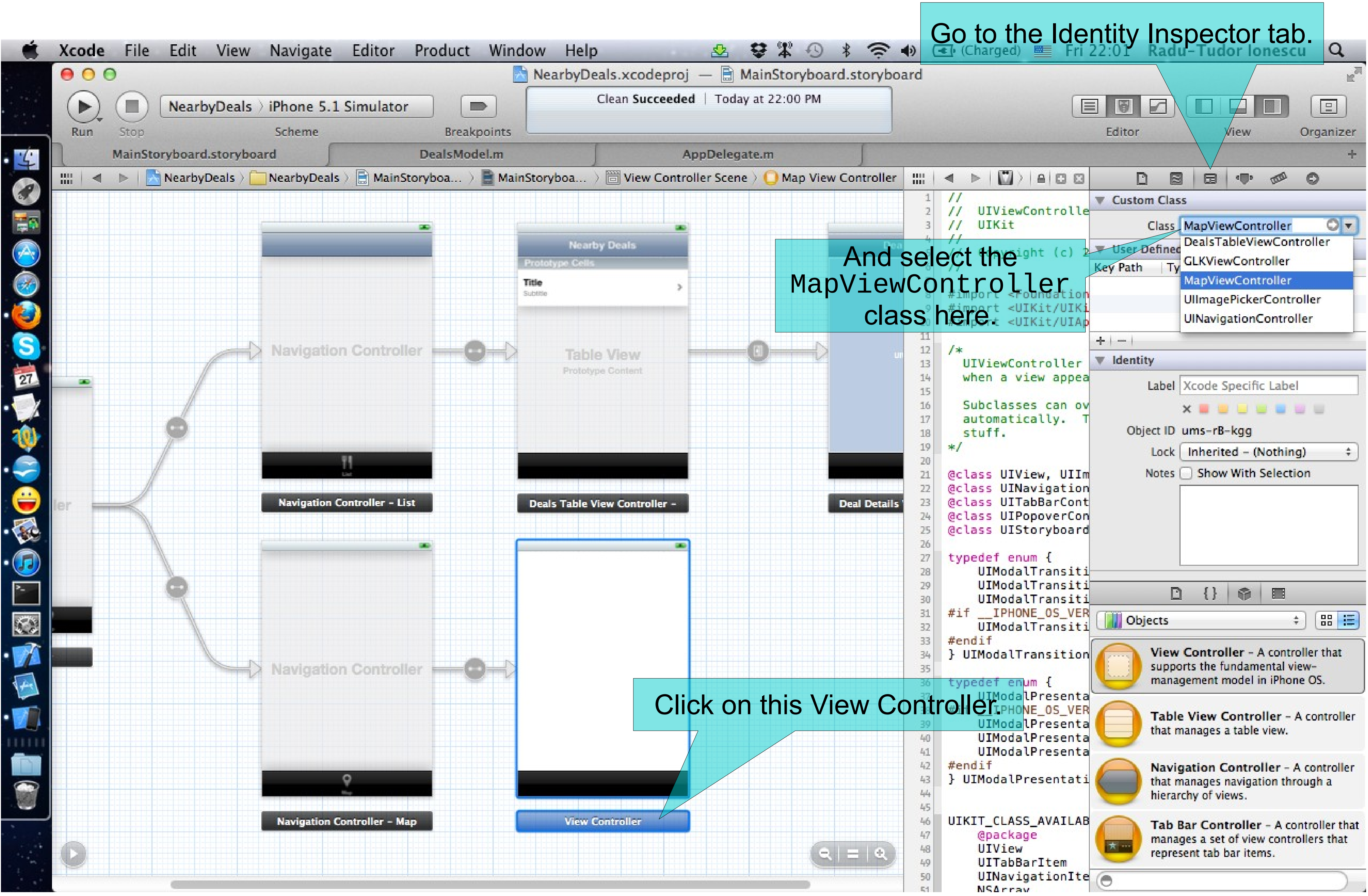
View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

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

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

We need as much room as possible for the Map View. Uncheck the "Bar Visibility" option here.


Xcode File Edit View Navigate Editor Product Window Help
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals › iPhone 5.1 Simulator
Clean Succeeded | Today at 22:00 PM
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals › NearbyDeals › MainStoryboa... › MainStoryboa... › View Controller Scene › Map View Controller
Navigation Controller
Navigation Controller – List
Nearby Deals
Prototype Cells
Title
Table View
Prototype Content
Deals Table View Controller –
Deal Details
Navigation Controller
Navigation Controller – Map
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.
Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Go to the Identity Inspector tab.

And select the MapViewController class here.

Click on this View Controller.

Custom Class

Class: MapViewController

- DealsTableViewController
- GLKViewController
- MapViewController
- UIImagePickerController
- UINavigationController

User Defined

Key Path | Ty

Identity

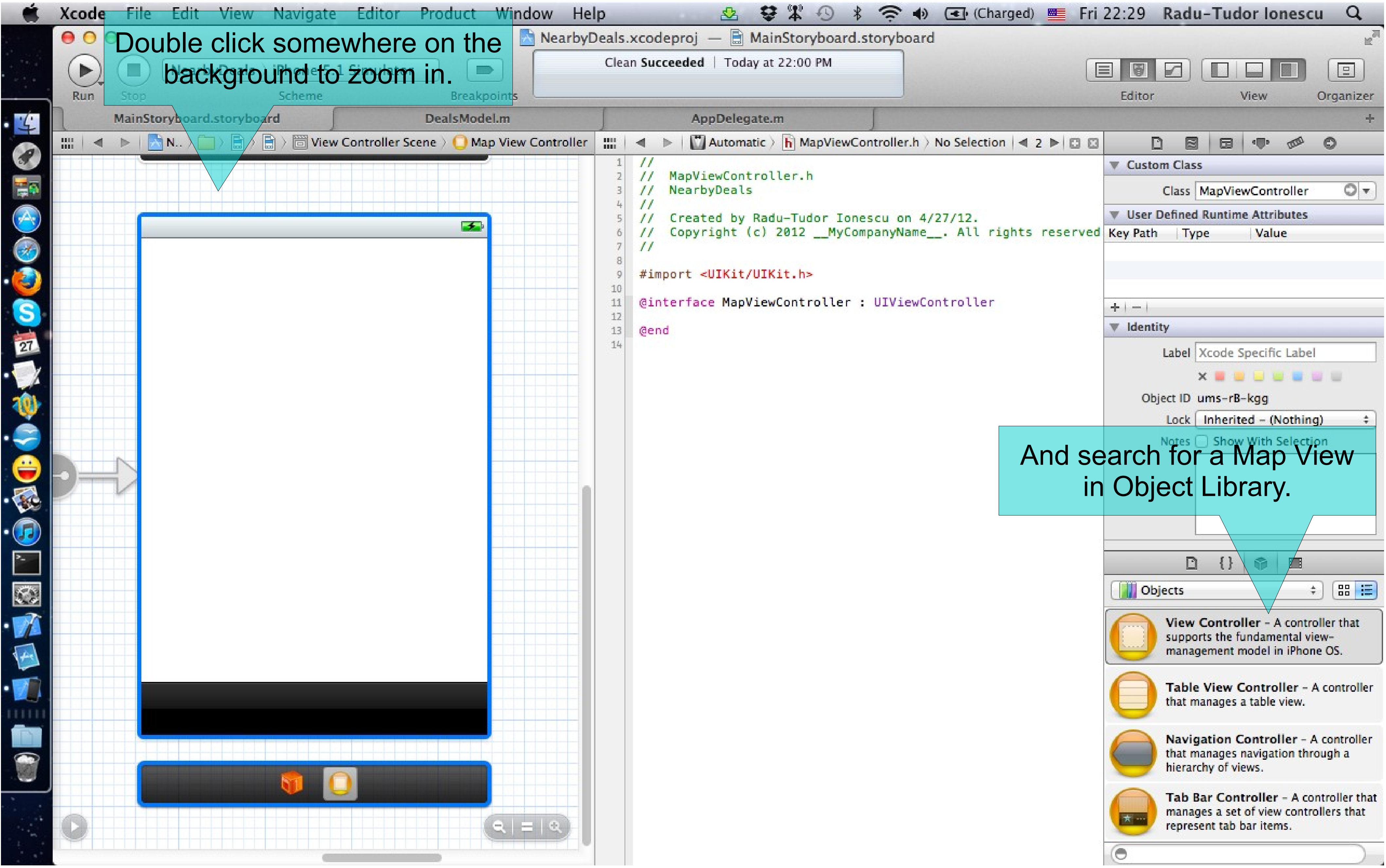
Label: Xcode Specific Label

Object ID: ums-rB-kgg

Lock: Inherited – (Nothing)

Notes: Show With Selection

```
//
//  UIViewControlle
//  UIKit
//
//  Copyright (c) 2
//
#import <Foundation
#import <UIKit/UIKi
#import <UIKit/UIAp

/*
  UIViewController
  when a view appea

  Subclasses can ov
  automatically.  T
  stuff.
*/

@class UIView, UIIm
@class UINavigation
@class UITabBarCont
@class UIPopoverCon
@class UIStoryboard

typedef enum {
    UIModalTransiti
    UIModalTransiti
    UIModalTransiti
#if __IPHONE_OS_VER
    UIModalTransiti
#endif
} UIModalTransition

typedef enum {
    UIModalPresenta
#if __IPHONE_OS_VER
    UIModalPresenta
    UIModalPresenta
    UIModalPresenta
#endif
} UIModalPresentati

UIKIT_CLASS_AVAILAB
    @package
    UIView
    UITabBarItem
    UINavigationIte
    NSArray
```



Double click somewhere on the background to zoom in.

And search for a Map View in Object Library.

```
//
//  MapViewController.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MapViewController : UIViewController

@end
```

Custom Class
Class: MapViewController

Identity
Label: Xcode Specific Label
Object ID: ums-rB-kgg
Lock: Inherited - (Nothing)

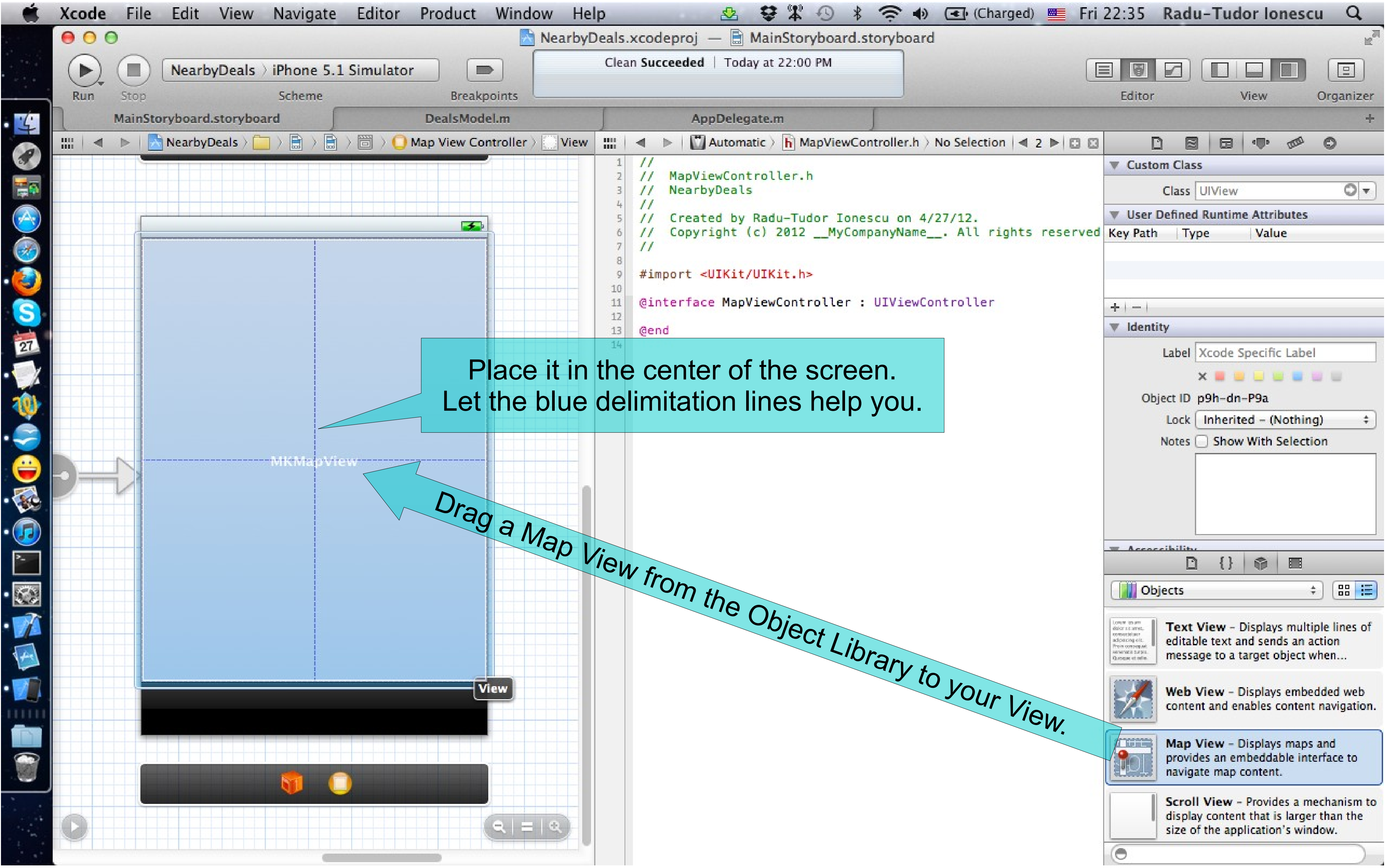
View Controller - A controller that supports the fundamental view-management model in iPhone OS.

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

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

Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.

Xcode File Edit View Navigate Editor Product Window Help
Fri 22:29 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Clean Succeeded | Today at 22:00 PM
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
View Controller Scene
Map View Controller
Automatic
MapViewController.h
No Selection
User Defined Runtime Attributes
Key Path
Type
Value
Objects

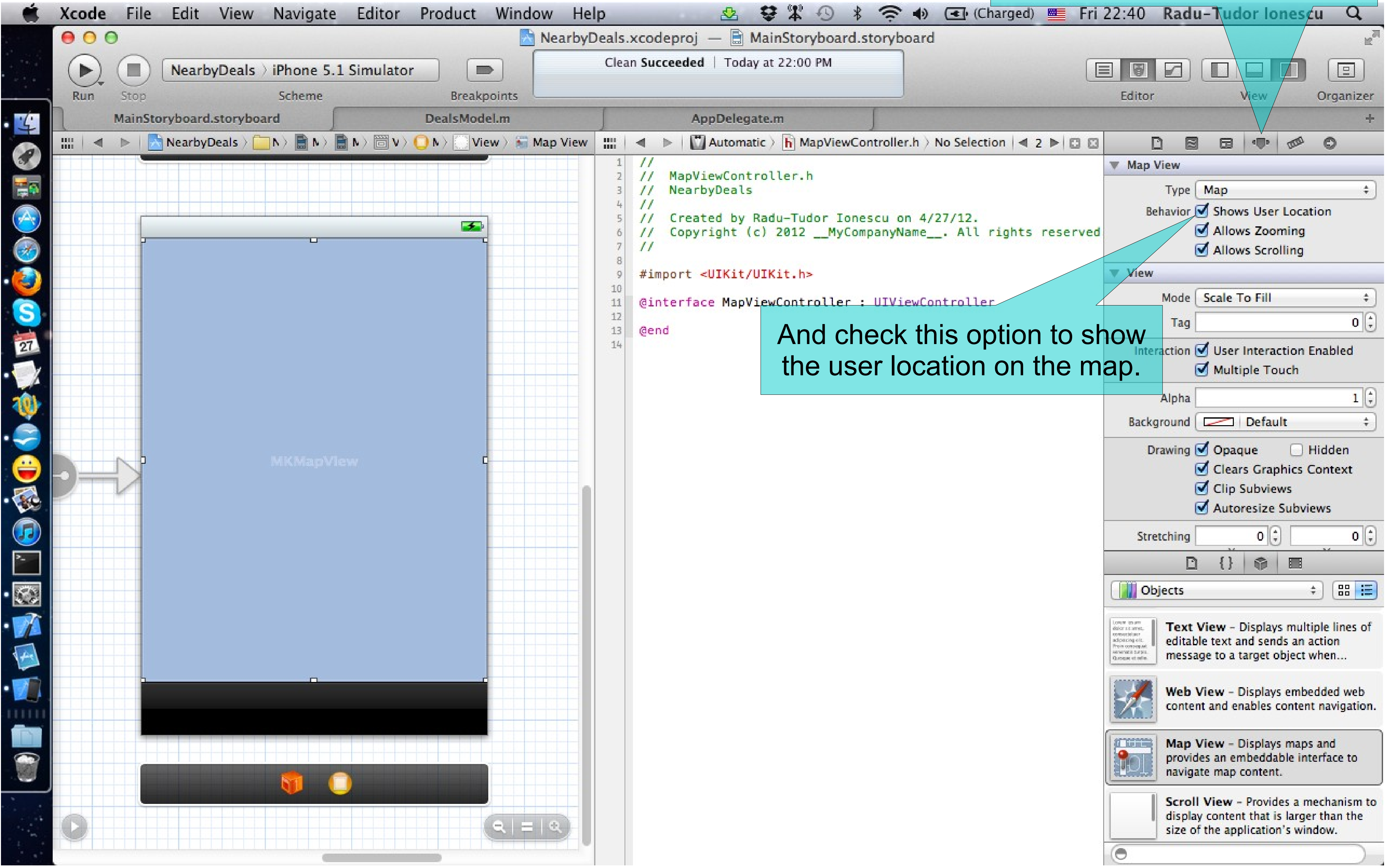

Xcode
File
Edit
View
Navigate
Editor
Product
Window
Help
(Charged)
Fri 22:35
Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals 〉 iPhone 5.1 Simulator
Scheme
Breakpoints
Clean Succeeded | Today at 22:00 PM
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals 〉 Map View Controller 〉 View
Automatic 〉 MapViewController.h 〉 No Selection
MKMapView
View
Custom Class
Class UIView
User Defined Runtime Attributes
Key Path
Type
Value
Identity
Label Xcode Specific Label
Object ID p9h-dn-P9a
Lock Inherited – (Nothing)
Notes Show With Selection
Objects
Text View – Displays multiple lines of editable text and sends an action message to a target object when...
Web View – Displays embedded web content and enables content navigation.
Map View – Displays maps and provides an embeddable interface to navigate map content.
Scroll View – Provides a mechanism to display content that is larger than the size of the application's window.

```
//
//  MapViewController.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved
//

#import <UIKit/UIKit.h>

@interface MapViewController : UIViewController

@end
```

Place it in the center of the screen.
Let the blue delimitation lines help you.

Drag a Map View from the Object Library to your View.

Choose Attributes Inspector.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 22:40 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › View › Map View
Automatic › MapViewController.h › No Selection
MKMapView
Objects

Clean Succeeded | Today at 22:00 PM

```
//
//  MapViewController.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved
//

#import <UIKit/UIKit.h>

@interface MapViewController : UIViewController

@end
```

Map View
- Type: Map
- Behavior: ✓ Shows User Location, ✓ Allows Zooming, ✓ Allows Scrolling

View
- Mode: Scale To Fill
- Tag: 0
- Interaction: ✓ User Interaction Enabled, ✓ Multiple Touch
- Alpha: 1
- Background: Default
- Drawing: ✓ Opaque, ☐ Hidden, ✓ Clears Graphics Context, ✓ Clip Subviews, ✓ Autoresize Subviews
- Stretching: 0 0

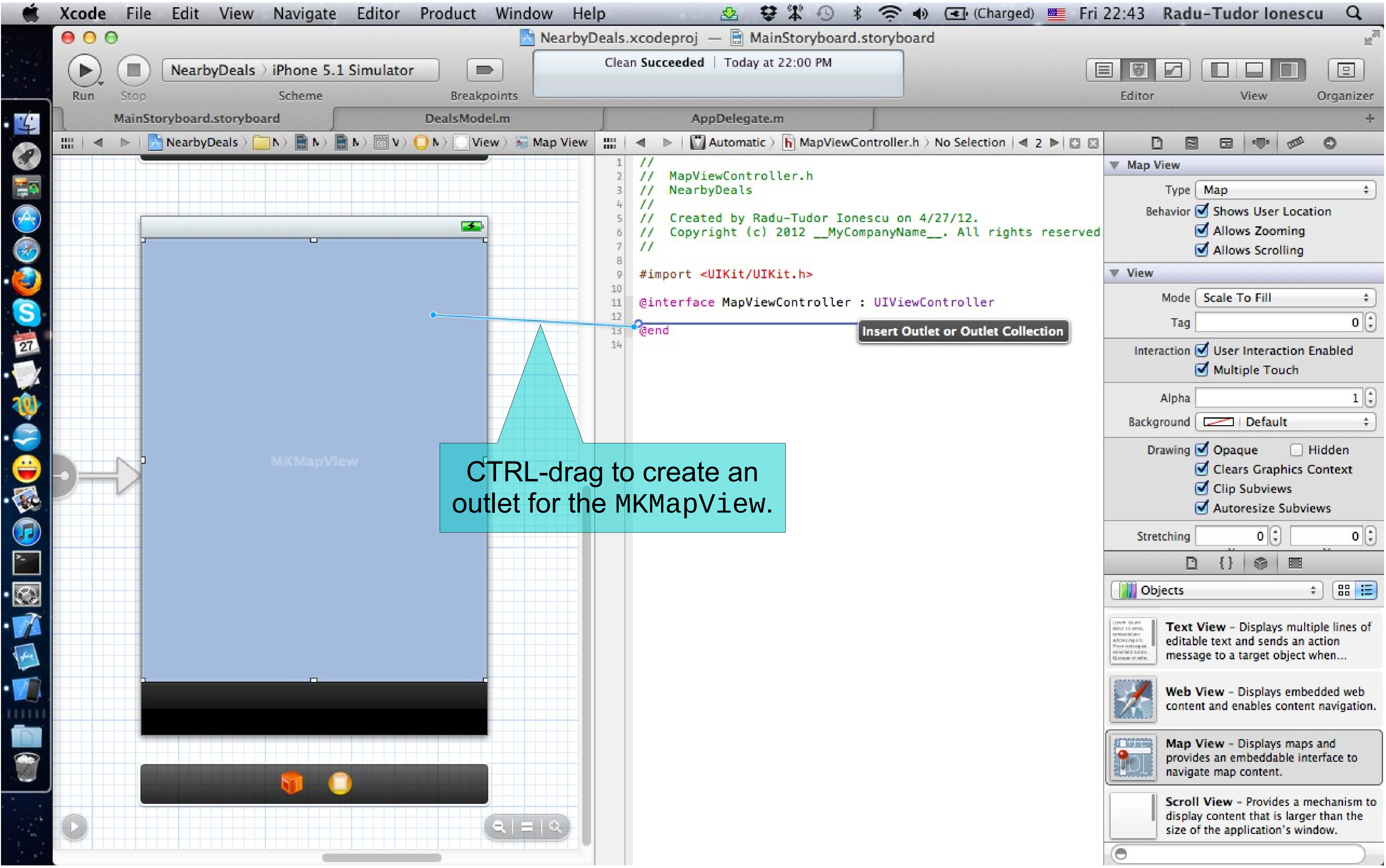
Text View – Displays multiple lines of editable text and sends an action message to a target object when...

Web View – Displays embedded web content and enables content navigation.

Map View – Displays maps and provides an embeddable interface to navigate map content.

Scroll View – Provides a mechanism to display content that is larger than the size of the application's window.

And check this option to show the user location on the map.


Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 22:43 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run
Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme
Breakpoints
Clean Succeeded | Today at 22:00 PM
Editor
View
Organizer
MainStoryboard.storyboard
DealsModel.m
AppDelegate.m
NearbyDeals › View › Map View
Automatic › MapViewController.h › No Selection
MKMapView
Insert Outlet or Outlet Collection
Objects

```
//
//  MapViewController.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved
//

#import <UIKit/UIKit.h>

@interface MapViewController : UIViewController

@end
```

Map View

Type: Map
Behavior: ✓ Shows User Location; ✓ Allows Zooming; ✓ Allows Scrolling

View

Mode: Scale To Fill
Tag: 0
Interaction: ✓ User Interaction Enabled; ✓ Multiple Touch
Alpha: 1
Background: Default
Drawing: ✓ Opaque; Hidden; ✓ Clears Graphics Context; ✓ Clip Subviews; ✓ Autoresize Subviews
Stretching: 0 0

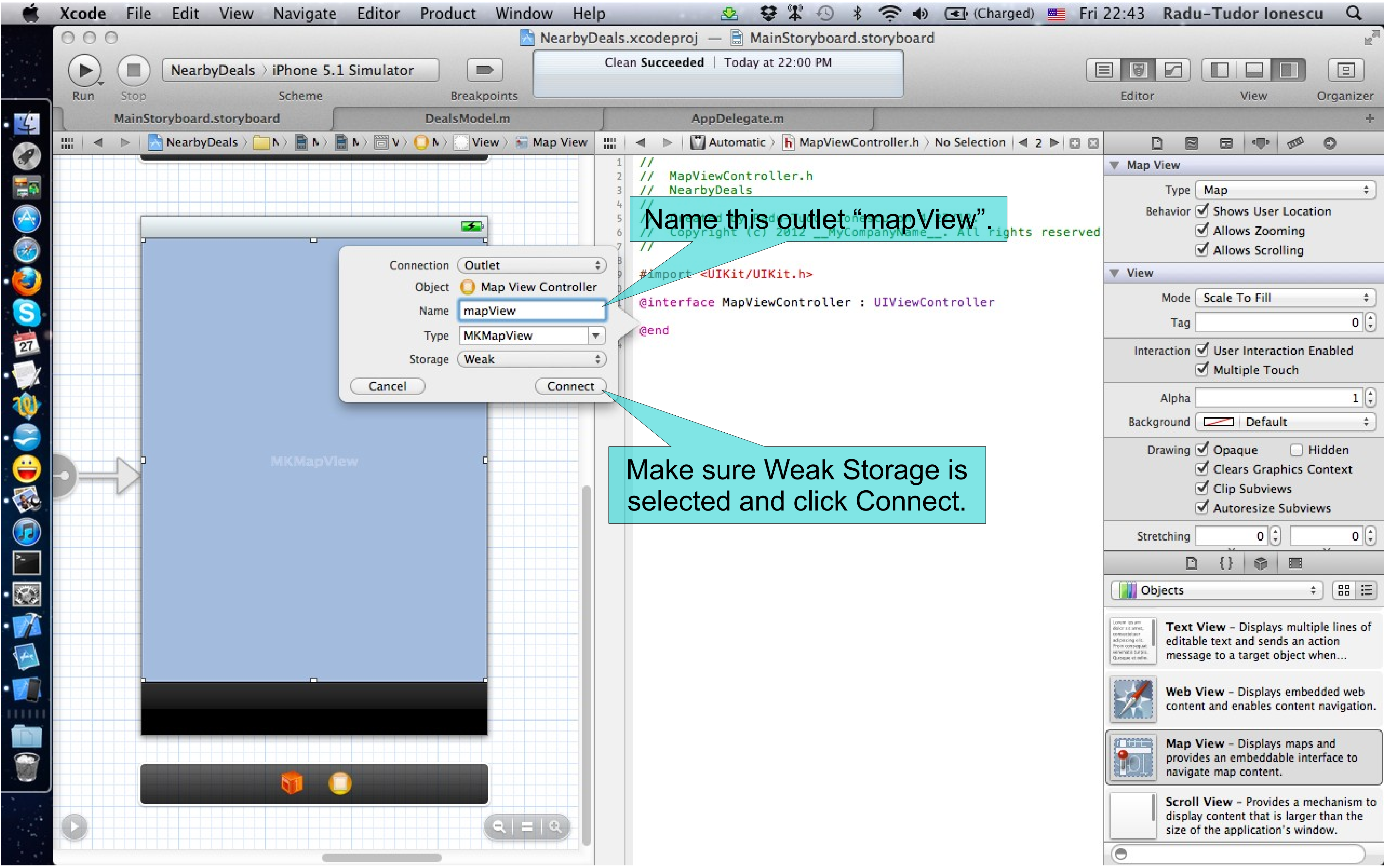
Text View – Displays multiple lines of editable text and sends an action message to a target object when...

Web View – Displays embedded web content and enables content navigation.

Map View – Displays maps and provides an embeddable interface to navigate map content.

Scroll View – Provides a mechanism to display content that is larger than the size of the application's window.

CTRL-drag to create an outlet for the `MKMapView`.



Name this outlet “mapView”.

Make sure Weak Storage is selected and click Connect.

```
//
//  MapViewController.h
//  NearbyDeals
//
//  Created by ...
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MapViewController : UIViewController

@end
```

Connection	Outlet
Object	Map View Controller
Name	mapView
Type	MKMapView
Storage	Weak

Cancel | Connect

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 22:43 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop Scheme Breakpoints Editor View Organizer
NearbyDeals › iPhone 5.1 Simulator
Clean Succeeded | Today at 22:00 PM
MainStoryboard.storyboard DealsModel.m AppDelegate.m
Automatic › MapViewController.h › No Selection
MKMapView

Map View
Type: Map
Behavior: ✓ Shows User Location ✓ Allows Zooming ✓ Allows Scrolling

View
Mode: Scale To Fill
Tag: 0
Interaction: ✓ User Interaction Enabled ✓ Multiple Touch
Alpha: 1
Background: Default
Drawing: ✓ Opaque ☐ Hidden ✓ Clears Graphics Context ✓ Clip Subviews ✓ Autoresize Subviews
Stretching: 0 0

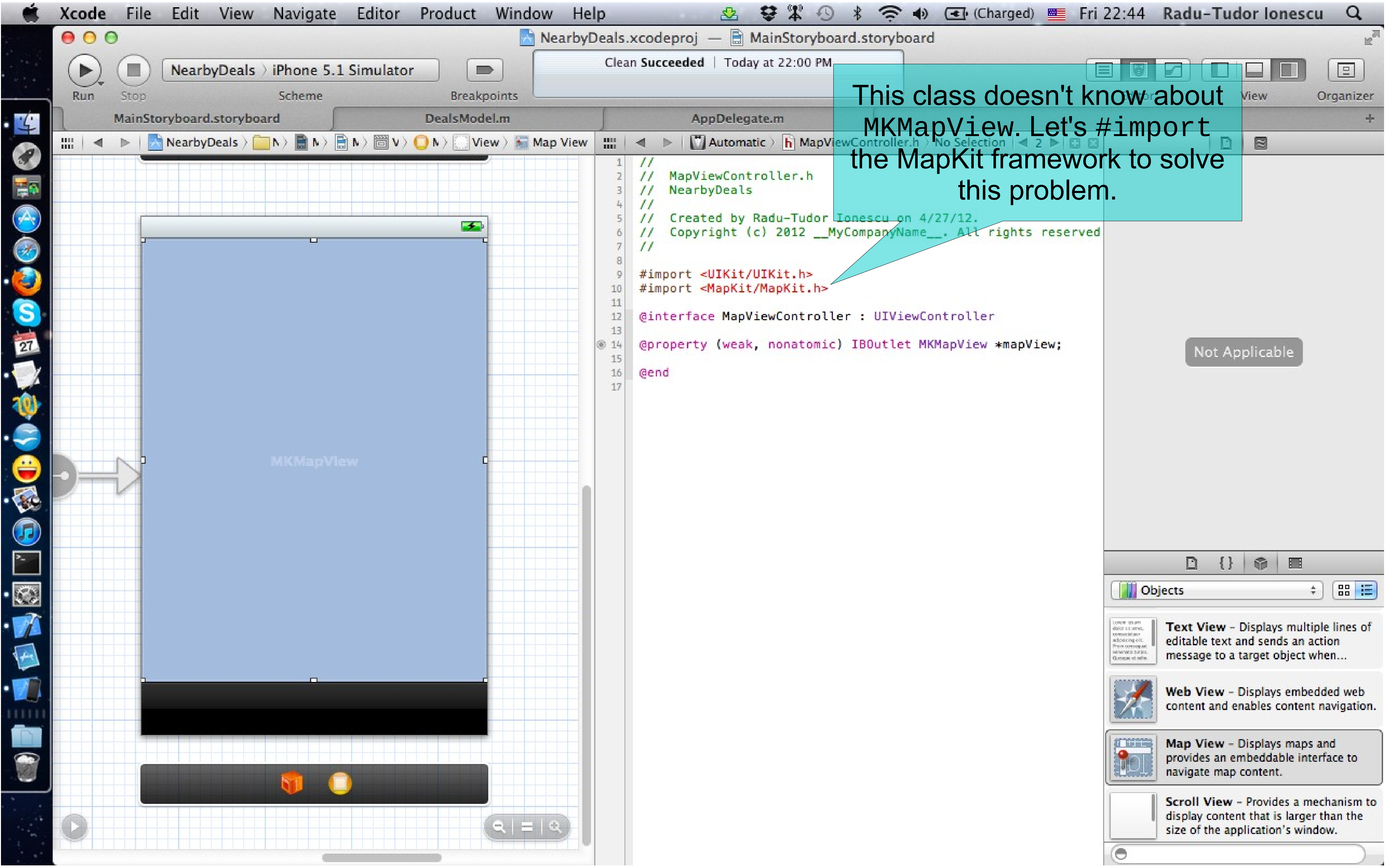
Objects

Text View – Displays multiple lines of editable text and sends an action message to a target object when...

Web View – Displays embedded web content and enables content navigation.

Map View – Displays maps and provides an embeddable interface to navigate map content.

Scroll View – Provides a mechanism to display content that is larger than the size of the application’s window.



This class doesn't know about MKMapView. Let's #import the MapKit framework to solve this problem.

```
//
//  MapViewController.h
//  NearbyDeals
//
//  Created by Radu-Tudor Ionescu on 4/27/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved
//

#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface MapViewController : UIViewController

@property (weak, nonatomic) IBOutlet MKMapView *mapView;

@end
```

Xcode File Edit View Navigate Editor Product Window Help
(Charged) Fri 22:44 Radu-Tudor Ionescu
NearbyDeals.xcodeproj — MainStoryboard.storyboard
Run Stop
NearbyDeals › iPhone 5.1 Simulator
Scheme Breakpoints
Clean Succeeded | Today at 22:00 PM
View Organizer
MainStoryboard.storyboard DealsModel.m AppDelegate.m
NearbyDeals View Map View
Automatic MapViewController.h No Selection
MKMapView
Not Applicable
Objects
Text View – Displays multiple lines of editable text and sends an action message to a target object when...
Web View – Displays embedded web content and enables content navigation.
Map View – Displays maps and provides an embeddable interface to navigate map content.
Scroll View – Provides a mechanism to display content that is larger than the size of the application's window.

Task 4

Task: Add the Map Kit framework and create the map view.

19. Select the MapViewController.m file in Assistant Editor.

20. Rename the instance variable of the `mapView` `@property` by prefixing it with underscore.

21. Test the application in iOS Simulator.

22. Simulate locations using the BucharestLocations GPX file.

23. Go on the second tab of the application (the one that displays the map). Find the user location marked by a rounded blue pin.

24. Stop running the application.

We are going to configure the Map View during the next lab. Try to solve the assignments for now.

Assignment 1

Assignment: Remove the `NSLogs` from the handler methods that get executed when application life cycle events occur.

Hint: These methods are in the AppDelegate.m file.

Assignment 2

Assignment: Add a getter for the `deviceLocation` `@property` of the `sharedModel` that will lazily instantiate it.

Also find and remove the unnecessary code from the `locationManager:didUpdateToLocation:fromLocation:`.

Hint: Use the `initWithLatitude:longitude:` to initialize the `CLLocation` object. Pass zero as the arguments of this method.

When the `CLLocationManager` updates us with a new location (especially the first time) we no longer have to test if the `deviceLocation` is `nil` since it gets lazily instantiated in its getter.

Assignment 3

Assignment: Add a refresh button on the right side of navigation bar of the Table View Controller. The refresh button action is to make a new request to the GeoAds+ server for nearby deals.

Hint: Drag and drop a `UIBarButtonItem` from Object Library. Set its Identifier to "Refresh" in Attributes Inspector. Create a new action for it and make a `requestDealsNearLocation:limit:` using the current device location.

Congratulations!