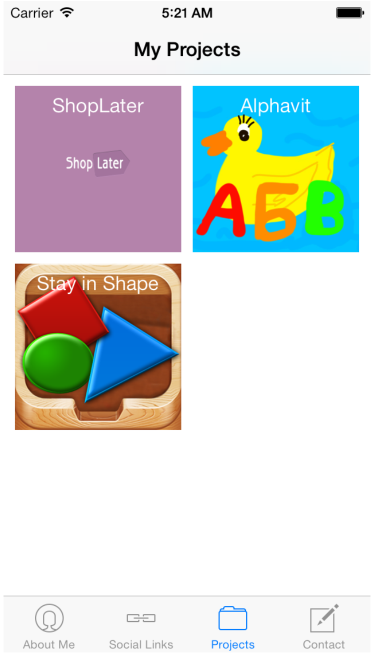
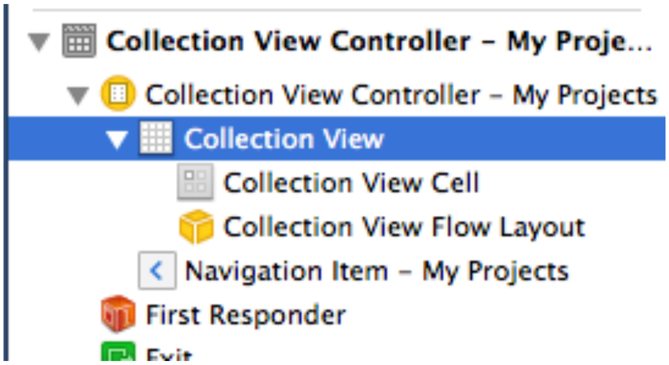
Projects

Your Resume app should for sure show off your Projects. So this is what we're going to make next!

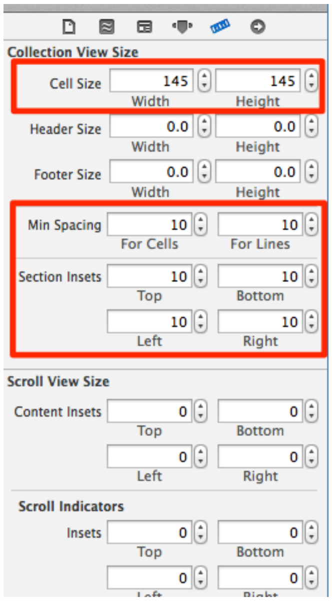


- In your **Storyboard**, drag on a new **Navigation Controller** and connect it to the **Tab Bar Controller** to create the Projects tab.
- Use the **projectsIcon** image provided in the Images folder for the Tab Bar Icon Image.
- **Rearrange** the order of the **Tab Bar Icons** on the **Tab Bar Controller** just by dragging the new Projects Icon to the left.
- Remove the TableViewController that came as part of your Projects Navigation Controller, can **connect the Projects Navigation Controller to a Collection View Controller**.
- Change the **title** of your Collection View Controller to **My Projects**.
- Select the Collection View:



```
▼ Collection View Controller – My Proje...
  ▼ Collection View Controller – My Projects
    ▼ Collection View
        Collection View Cell
        Collection View Flow Layout
      Navigation Item – My Projects
  First Responder
  Exit
```

- Go to the **Ruler Tab** on the right hand pane, and specify the **Size** of your Project Cells, and how much they should be inset. Here are my sizes, but feel free to make it any sizes you like:



Collection View Size

Cell Size: 145 Width, 145 Height

Header Size: 0.0 Width, 0.0 Height

Footer Size: 0.0 Width, 0.0 Height

Min Spacing: 10 For Cells, 10 For Lines

Section Insets: 10 Top, 10 Bottom, 10 Left, 10 Right

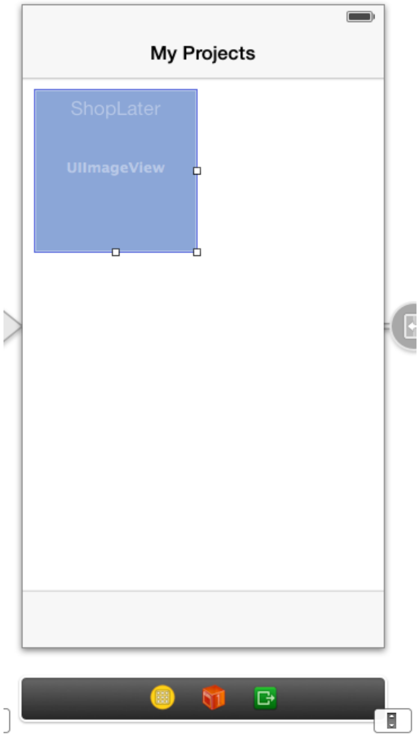
Scroll View Size

Content Insets: 0 Top, 0 Bottom, 0 Left, 0 Right

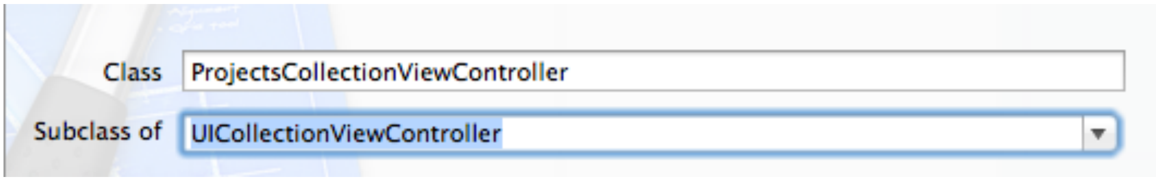
Scroll Indicators

Insets: 0 Top, 0 Bottom, 0 Left, 0 Right

- Change the **Background Color** of the **Collection View** to **White** (or any other color you like).
- Drag an **Image View** and a **Label** onto the **Collection View Project Cell**.

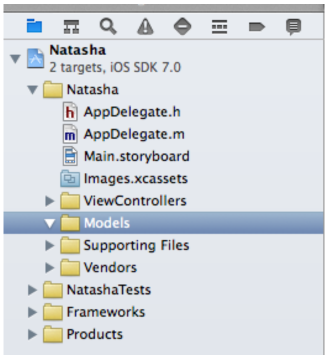


- Add a **Cell Identifier** called **projectCell** to your **Collection View Cell**.
- Create a **ProjectsCollectionViewController**, which is a subclass of **UICollectionViewController** to your View Controllers folder.

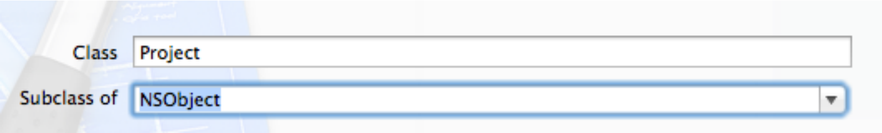


Class ProjectsCollectionViewController
Subclass of UICollectionViewController

- In your **Storyboard**, identify the Collection View Controller as the **ProjectsCollectionViewController**.
- In your Folder structure on the left hand pane, add a new Group called Models:

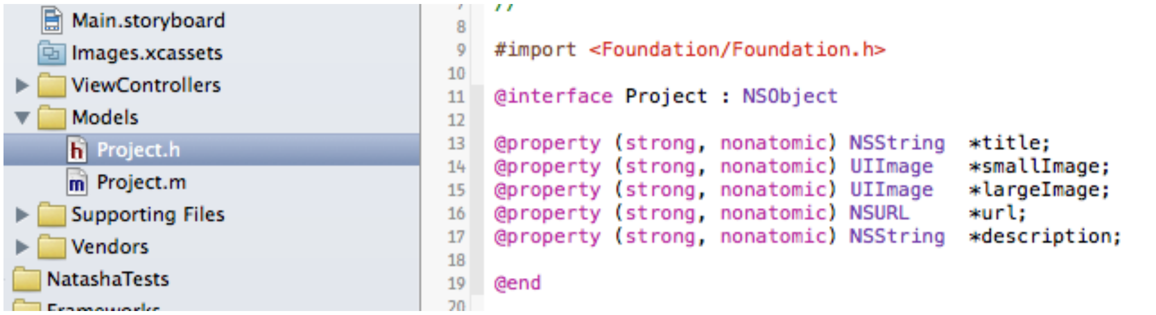


- We're going to create a **Project** model, which is a subclass of **NSObject**.



Class Project
Subclass of NSObject

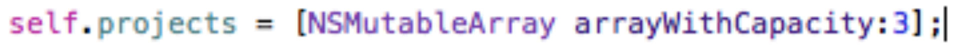
- In your Project.h file, add properties such as title, smallImage, largeImage, url, and description.



```
//

#import <Foundation/Foundation.h>

@interface Project : NSObject

@property (strong, nonatomic) NSString  *title;
@property (strong, nonatomic) UIImage   *smallImage;
@property (strong, nonatomic) UIImage   *largeImage;
@property (strong, nonatomic) NSURL     *url;
@property (strong, nonatomic) NSString  *description;

@end
```

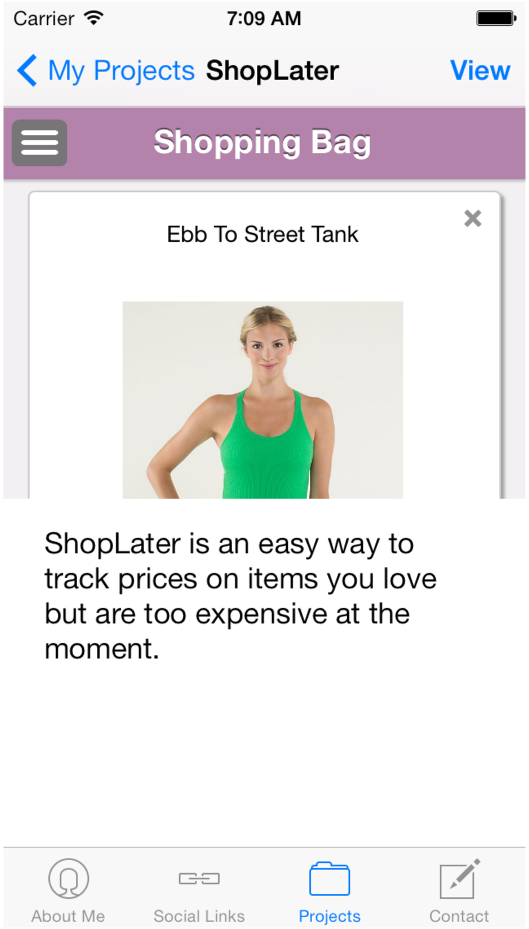
- **Import the Project.h** file into your **ProjectsCollectionViewController.m**.

- In your **ProjectsCollectionViewController**, add an **NSMutableArray property** called **projects**. This array will hold the Project Objects that you will create next.
- Look up the documentation for NSMutableArray. How is it different from an NSArray?
- Add a new **instance** method to setup your projects that **returns nothing**, called **setupProjects**.
- We first have to instantiate our NSMutableArray. Since you know **how many projects you'll be including in your Collection View**, use the arrayWithCapacity: class method to create an instance of NSMutableArray. I'll be including only three projects:



```
self.projects = [NSMutableArray arrayWithCapacity:3];
```

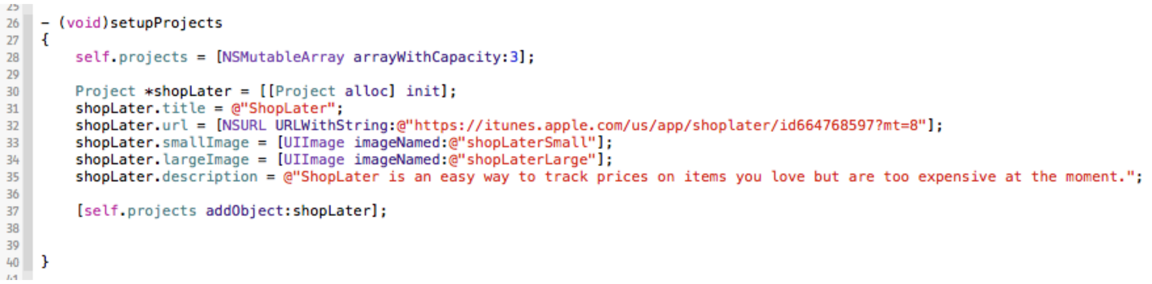
- Now, it's time to create your first Project. The **small image** is meant for the CollectionView Cell - **size 145 x 145** (don't forget to include 2x sizes!). When the user clicks on the cell, they'll be taken to a Project page similar to your About Me page. The Project page will include a **large image** - such as **320 x 240**, and have a **Description**. Feel free to go to the Storyboard and add a segue from the projectCell to this page, if you'd like a different layout or to play with different large image sizes.



Ebb To Street Tank

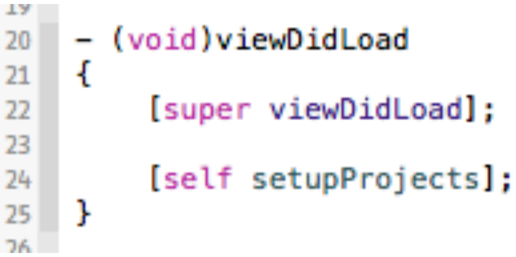
ShopLater is an easy way to track prices on items you love but are too expensive at the moment.

- Since a **Project** is a **subclass of NSObject**, look up NSObject in the documentation. We're going to use the **alloc** class method to **allocate memory** for the Project, and the **init** instance method to create an instance of it. The we'll set the project's properties. Once we've set the project's properties, we'll add it to our projects array.



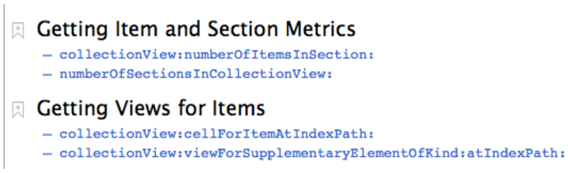
```
- (void)setupProjects
{
    self.projects = [NSMutableArray arrayWithCapacity:3];

    Project *shopLater = [[Project alloc] init];
    shopLater.title = @"ShopLater";
    shopLater.url = [NSURL URLWithString:@"https://itunes.apple.com/us/app/shoplater/id664768597?mt=8"];
    shopLater.smallImage = [UIImage imageNamed:@"shopLaterSmall"];
    shopLater.largeImage = [UIImage imageNamed:@"shopLaterLarge"];
    shopLater.description = @"ShopLater is an easy way to track prices on items you love but are too expensive at the moment.";

    [self.projects addObject:shopLater];


}
```

- Do the same thing for all your other projects! Make sure your small and large images are in the Images.xcassets folder.
- Call the **setupProjects** method in your viewDidLoad:



```
- (void)viewDidLoad
{
    [super viewDidLoad];

    [self setupProjects];
}
```

- Now that we have the projects array populated with Project objects, we're going to use it as our Collection View Data Source. Look up CollectionViewDataSource in the documentation.



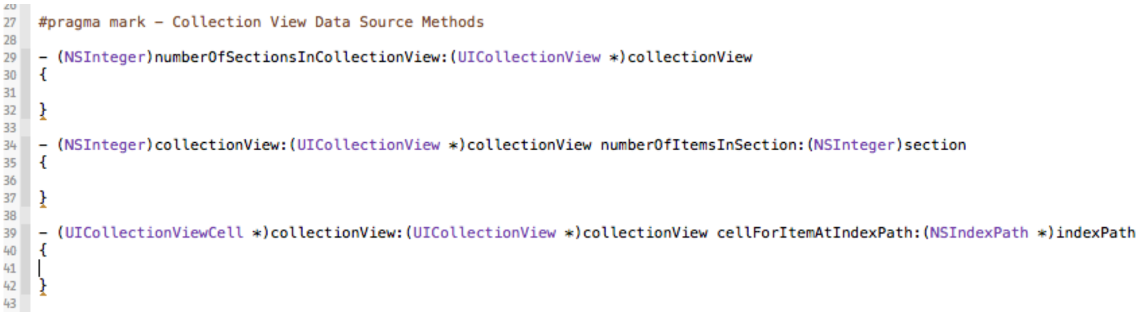
Getting Item and Section Metrics

- collectionView:numberOfItemsInSection:
- numberOfSectionsInCollectionView:

Getting Views for Items

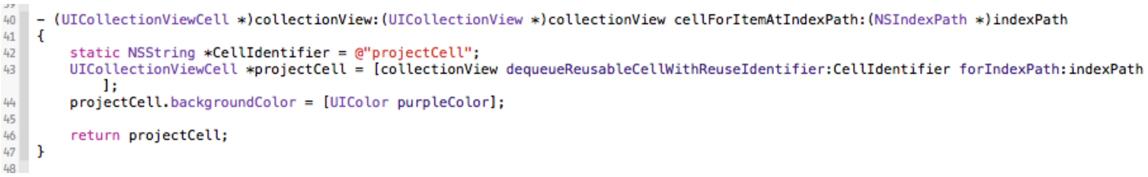
- collectionView:cellForItemAtIndexPath:
- collectionView:viewForSupplementaryElementOfKind:atIndexPath:

- We're going to implement the top three methods listed in the documentation.
- Click on each method, and copy it into your ProjectsCollectionViewController.m:



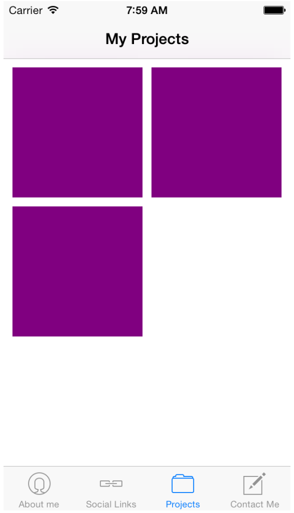
```
#pragma mark - Collection View Data Source Methods

- (NSInteger)numberOfSectionsInCollectionView:(UICollectionView *)collectionView
{

}

- (NSInteger)collectionView:(UICollectionView *)collectionView numberOfItemsInSection:(NSInteger)section
{

}

- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView cellForItemAtIndexPath:(NSIndexPath *)indexPath
{

}
```

- Look back to your SocialLinksTableView, and just like in your Table View Data Source Methods, fill out the number of sections and number of rows for your projects collection view using the projects array.
- For the collectionView:cellForItemAtIndexPath: method, just dequeue the cell and change it's background color for now:

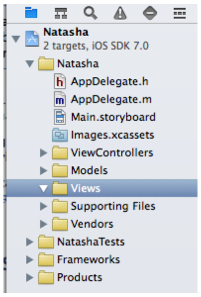


```
- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView cellForItemAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"projectCell";
    UICollectionViewCell *projectCell = [collectionView dequeueReusableCellWithReuseIdentifier:CellIdentifier forIndexPath:indexPath
        ];
    projectCell.backgroundColor = [UIColor purpleColor];

    return projectCell;
}
```

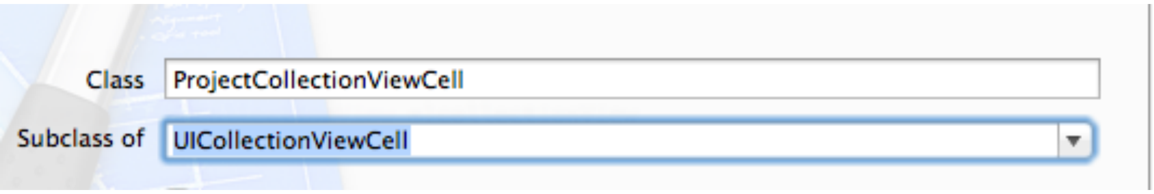
- Look in your SocialLinksTableView's tableView:cellForRowAtIndexPathMethod: - what are the differences / similarities between the collectionView:cellForItemAtIndexPath: method implementation?
- Run the app. You should see the each of your projects represented by a purple cell (or another color if you chose to make a background color something different). In my case, I have 3 projects, so I have 3 purple cells!



- Now we need to configure the cell. Since it's a custom cell, with a custom UIImageView and a custom UILabel, we're going to subclass the UICollectionViewCell so we can customize it!
- In your project directory, Create a new folder Group called Views:



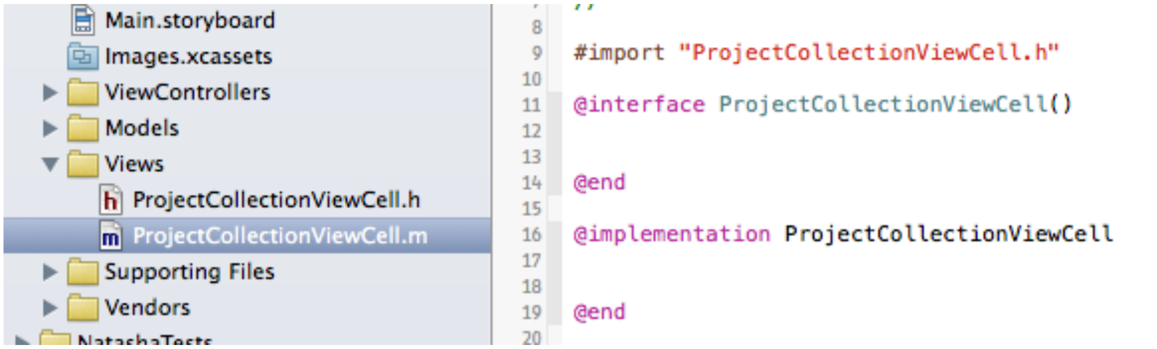
- Create a **ProjectCollectionViewCell**, which is a subclass of **UICollectionViewCell**:



Class: ProjectCollectionViewCell

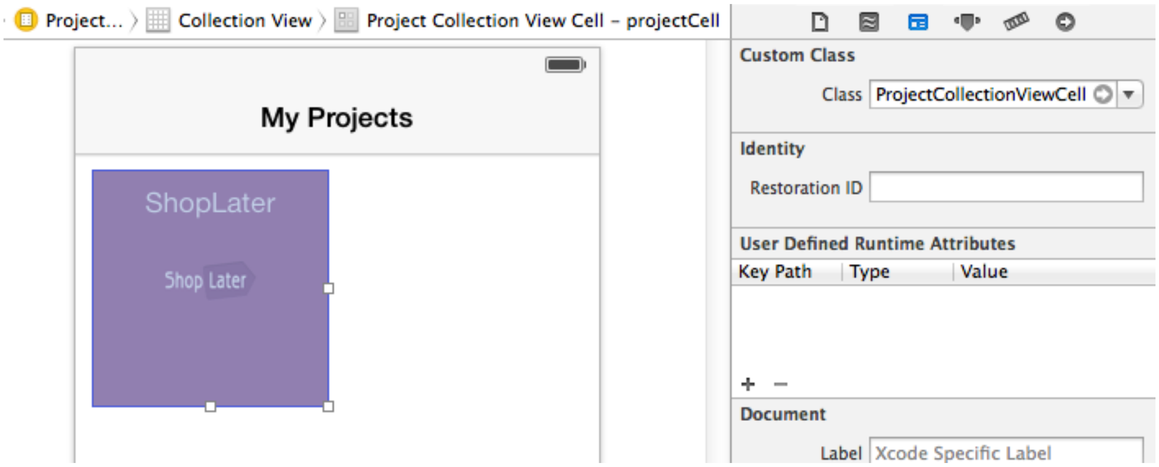
Subclass of: UICollectionViewCell

- Add an **@interface** section in the ProjectCollectionViewCell.m file. This is where we're going to connect the imageView and the UILabel for our cell from the Storyboard:



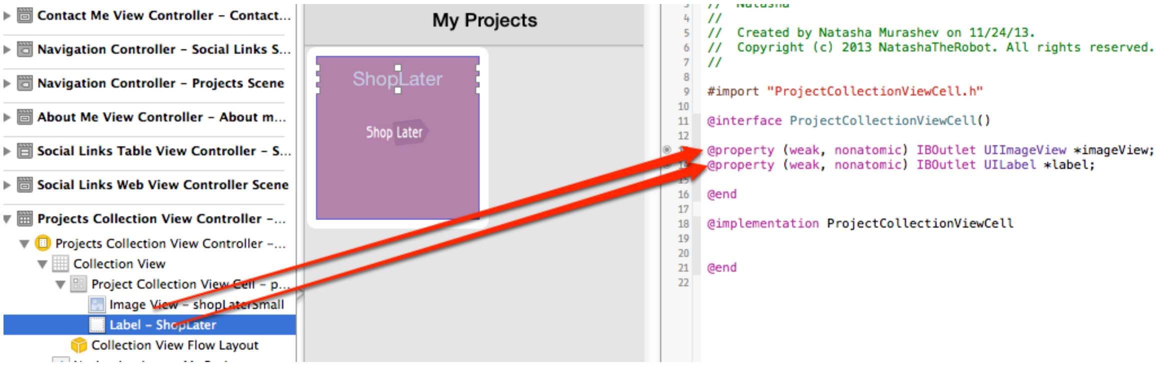
```
//

#import "ProjectCollectionViewCell.h"

@interface ProjectCollectionViewCell()


@end

@implementation ProjectCollectionViewCell


@end
```

- Go back to the Storyboard, and identify the projectCell as a ProjectCollectionViewCell:



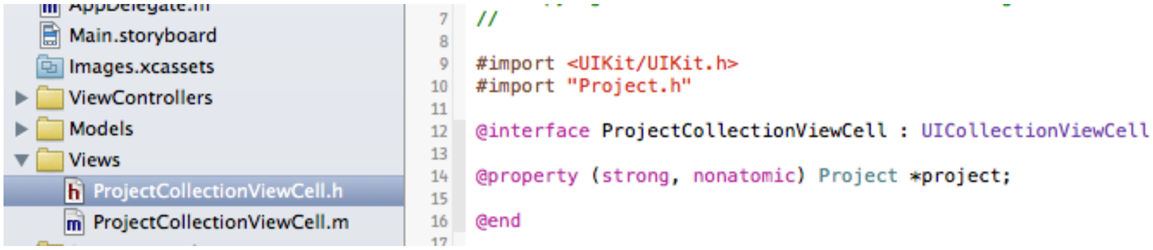
Class: ProjectCollectionViewCell

- Now, you can connect the UIImageView and the UILabel to the ProjectCollectionViewCell:


Contact Me View Controller – Contact…
Navigation Controller – Social Links S…
Navigation Controller – Projects Scene
About Me View Controller – About m…
Social Links Table View Controller – S…
Social Links Web View Controller Scene
Projects Collection View Controller –…
Projects Collection View Controller –…
Collection View
Project Collection View Cell – p…
Image View – shopLaterSmall
Label – ShopLater
Collection View Flow Layout
My Projects
ShopLater
Shop Later

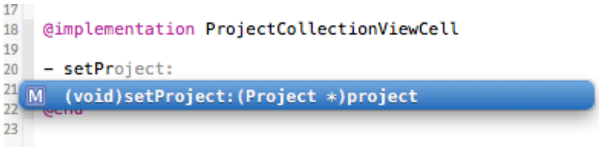
```
//  Natasha
//
//  Created by Natasha Murashev on 11/24/13.
//  Copyright (c) 2013 NatashaTheRobot. All rights reserved.
//

#import "ProjectCollectionViewCell.h"

@interface ProjectCollectionViewCell()

@property (weak, nonatomic) IBOutlet UIImageView *imageView;
@property (weak, nonatomic) IBOutlet UILabel *label;

@end

@implementation ProjectCollectionViewCell


@end
```

- In the ProjectCeoolectionViewCell.h, import the Project object, and make a project property. We're going to assign the project to each projectCell from the ProjectCollectionViewController:



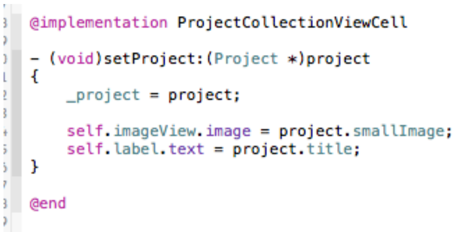
```
//

#import <UIKit/UIKit.h>
#import "Project.h"

@interface ProjectCollectionViewCell : UICollectionViewCell

@property (strong, nonatomic) Project *project;

@end
```

- When we assign the project to the cell, we also want to populate the cell's image and label. So we're going to override the project setter method (which was created for us by creating a property!).
- In your ProjectCollectionViewCell.m implementation section, start typing "-setProject" to get the setProject method associated with the project property:



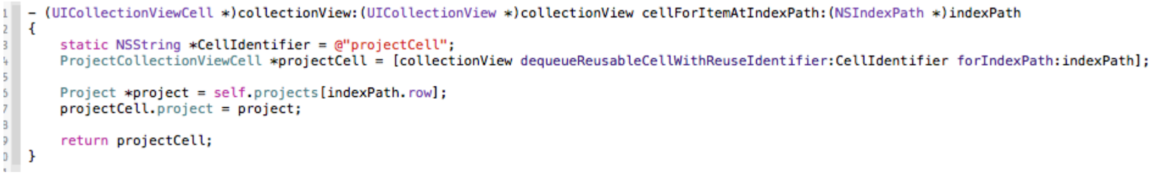
```
@implementation ProjectCollectionViewCell

- setProject:
M  (void)setProject:(Project *)project
@end
```

- Over ride the method to set the project instance variable and set the correct project cell image and title:



```
@implementation ProjectCollectionViewCell

- (void)setProject:(Project *)project
{
    _project = project;

    self.imageView.image = project.smallImage;
    self.label.text = project.title;
}

@end
```

- Now, back in our ProjectsCollectionViewController, we can **import** and **configure** the ProjectCollectionViewCell for each project:

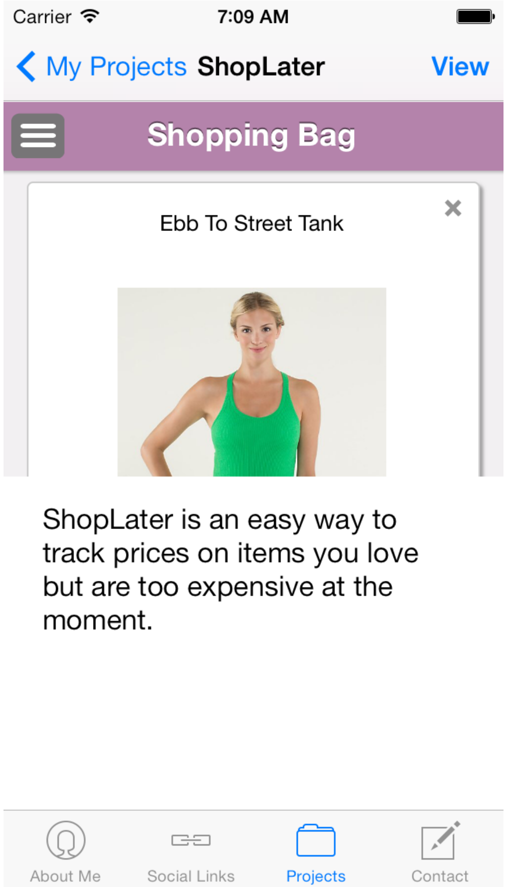


```
- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView cellForItemAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"projectCell";
    ProjectCollectionViewCell *projectCell = [collectionView dequeueReusableCellWithReuseIdentifier:CellIdentifier forIndexPath:indexPath];

    Project *project = self.projects[indexPath.row];
    projectCell.project = project;

    return projectCell;
}
```

- Run the app, and the projects will magically appear!
- **Challenge**:
  - Go back to your SocialLinksTableViewController.m. Do you see how your had several arrays for holding information about the social service? Create a SocialService model, similar to your Project model, which will have properties in your arrays - title, username, url.
  - Configure the SocialLinksTableViewController to use an array of SocialService models ad it’s main data source!

Project Details

When the user clicks on each project, have them go to a **ProjectDetailsViewController** with your project's **title, large image**, and **description**:



ShopLater is an easy way to track prices on items you love but are too expensive at the moment.

- Don't forget to user the **prepareForSegue:** method in your ProjectsCollectionViewController to pass the project from the ProjectsCollectionViewController to the ProjectDetailsViewController.
- When the user clicks the View button on the top right of the ProjectDetails, have them go to a ProjectWebViewController, where they can see the project page on the web.