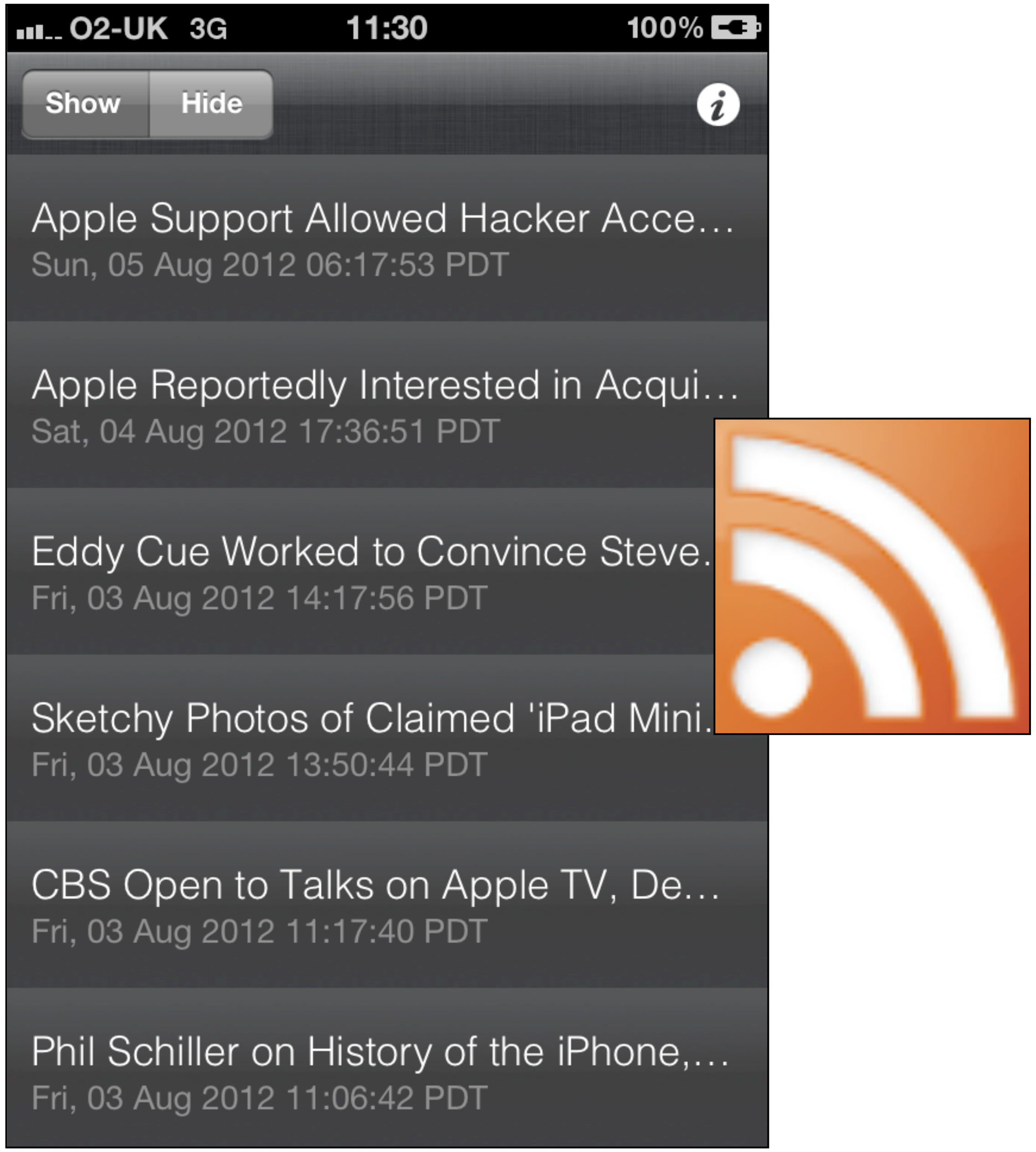
IMPLEMENTATION GUIDE

iRSS - iPhone



Apple Support Allowed Hacker Acce…
Sun, 05 Aug 2012 06:17:53 PDT

Apple Reportedly Interested in Acqui…
Sat, 04 Aug 2012 17:36:51 PDT

Eddy Cue Worked to Convince Steve…
Fri, 03 Aug 2012 14:17:56 PDT

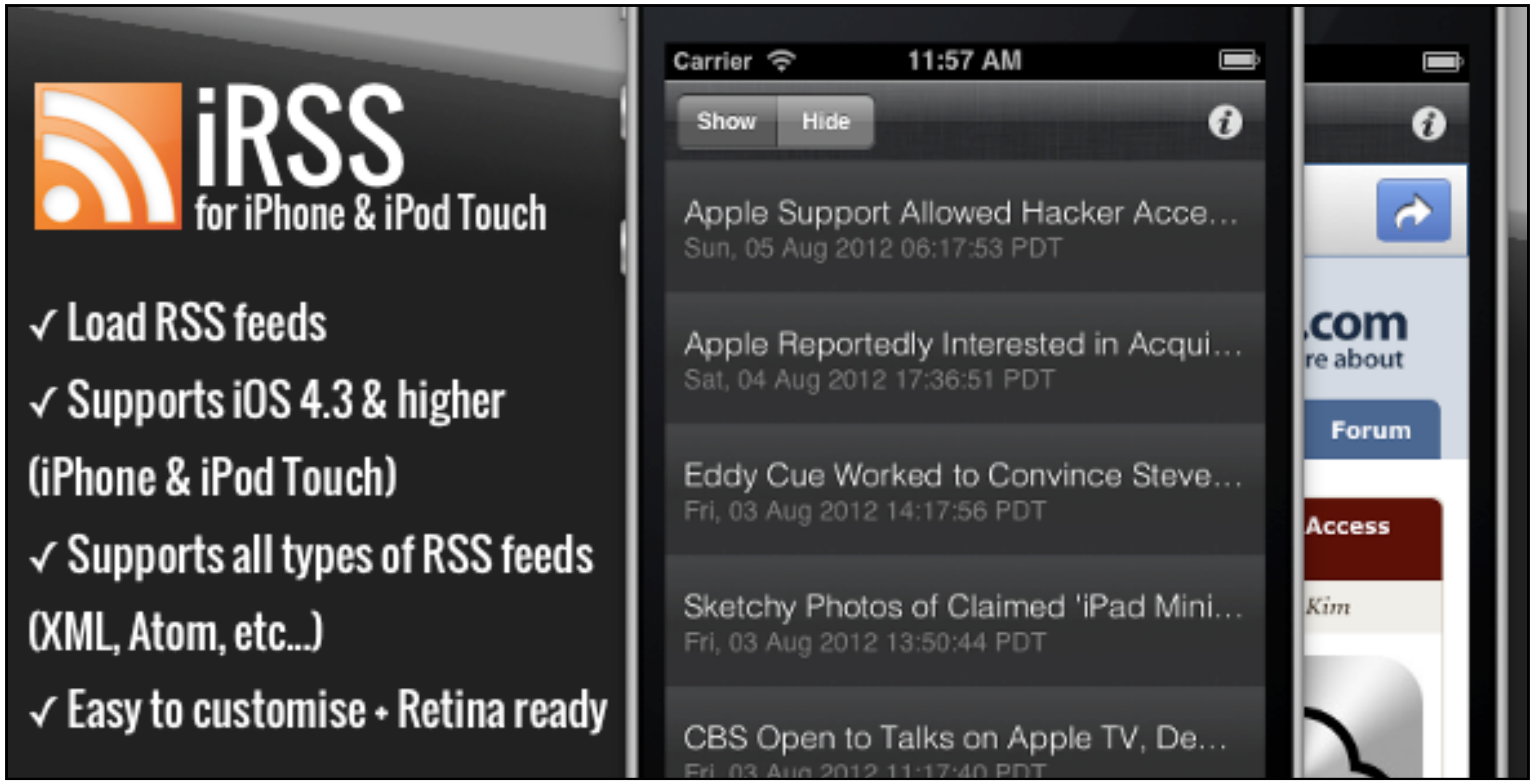
Sketchy Photos of Claimed 'iPad Mini…
Fri, 03 Aug 2012 13:50:44 PDT

CBS Open to Talks on Apple TV, De…
Fri, 03 Aug 2012 11:17:40 PDT

Phil Schiller on History of the iPhone,…
Fri, 03 Aug 2012 11:06:42 PDT

WELCOME

Implement iRSS into your existing iPhone apps is 3 steps.



✓ Load RSS feeds
✓ Supports iOS 4.3 & higher (iPhone & iPod Touch)
✓ Supports all types of RSS feeds (XML, Atom, etc...)
✓ Easy to customise + Retina ready

Apple Support Allowed Hacker Acce...
Sun, 05 Aug 2012 06:17:53 PDT

Apple Reportedly Interested in Acqui...
Sat, 04 Aug 2012 17:36:51 PDT

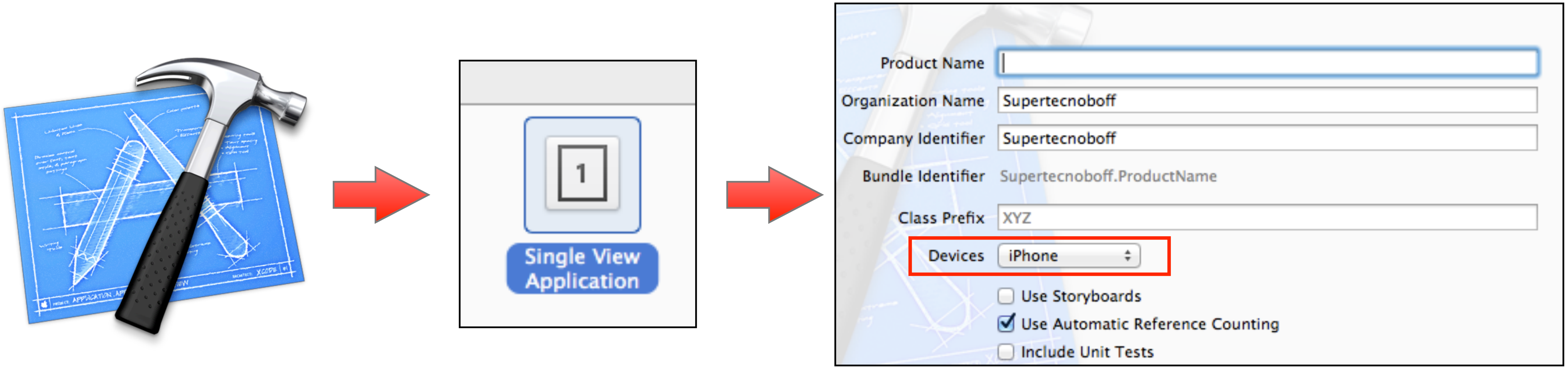
Eddy Cue Worked to Convince Steve...
Fri, 03 Aug 2012 14:17:56 PDT

Sketchy Photos of Claimed 'iPad Mini...
Fri, 03 Aug 2012 13:50:44 PDT

CBS Open to Talks on Apple TV, De...
Fri, 03 Aug 2012 11:17:40 PDT

It is really easy to implement iRSS into your existing iPhone applications and this guide is going to show you how. This guide will provide you with 3 simple steps which you have to perform in order to implement iRSS into your existing iPhone applications. If you need anymore help, please send me an email: falkzone@icloud.com

Step One



Product Name
Organization Name Supertecnoboff
Company Identifier Supertecnoboff
Bundle Identifier Supertecnoboff.ProductName
Class Prefix XYZ
Devices iPhone

□ Use Storyboards
☑ Use Automatic Reference Counting
□ Include Unit Tests

Open Xcode and start a new project. Select “Single View Application” and click next. Then type in your app name and make sure the “Device Family: option is set to “iPhone”. Click next and save your Xcode project.

Now we are going to start with the coding. In all Xcode projects we generally have two types of code files - a “header” file (which is the .h file) and the “implementation” file (which is the .m file).

Lets start with the header file. Now by default, Xcode will call this file “ViewController.h” - select that and we can being coding our header file.

Step One

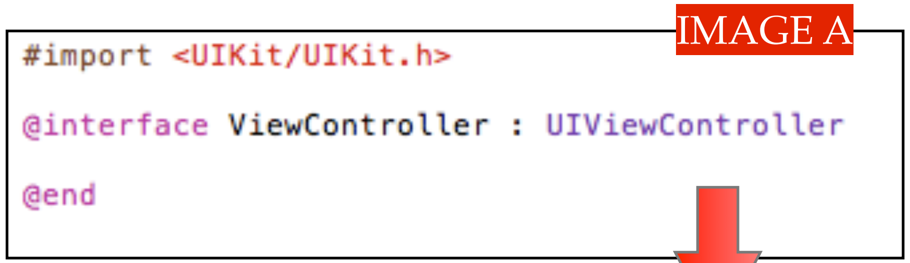


IMAGE A

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

@interface ViewController : UIViewController

@end
```

This is what your header file will look like to begin with (Image A). You need to copy and paste the code provided in ".h code" file (provided in the iRSS "Code" folder) into your header file so that your header file will look like this: (Image B).

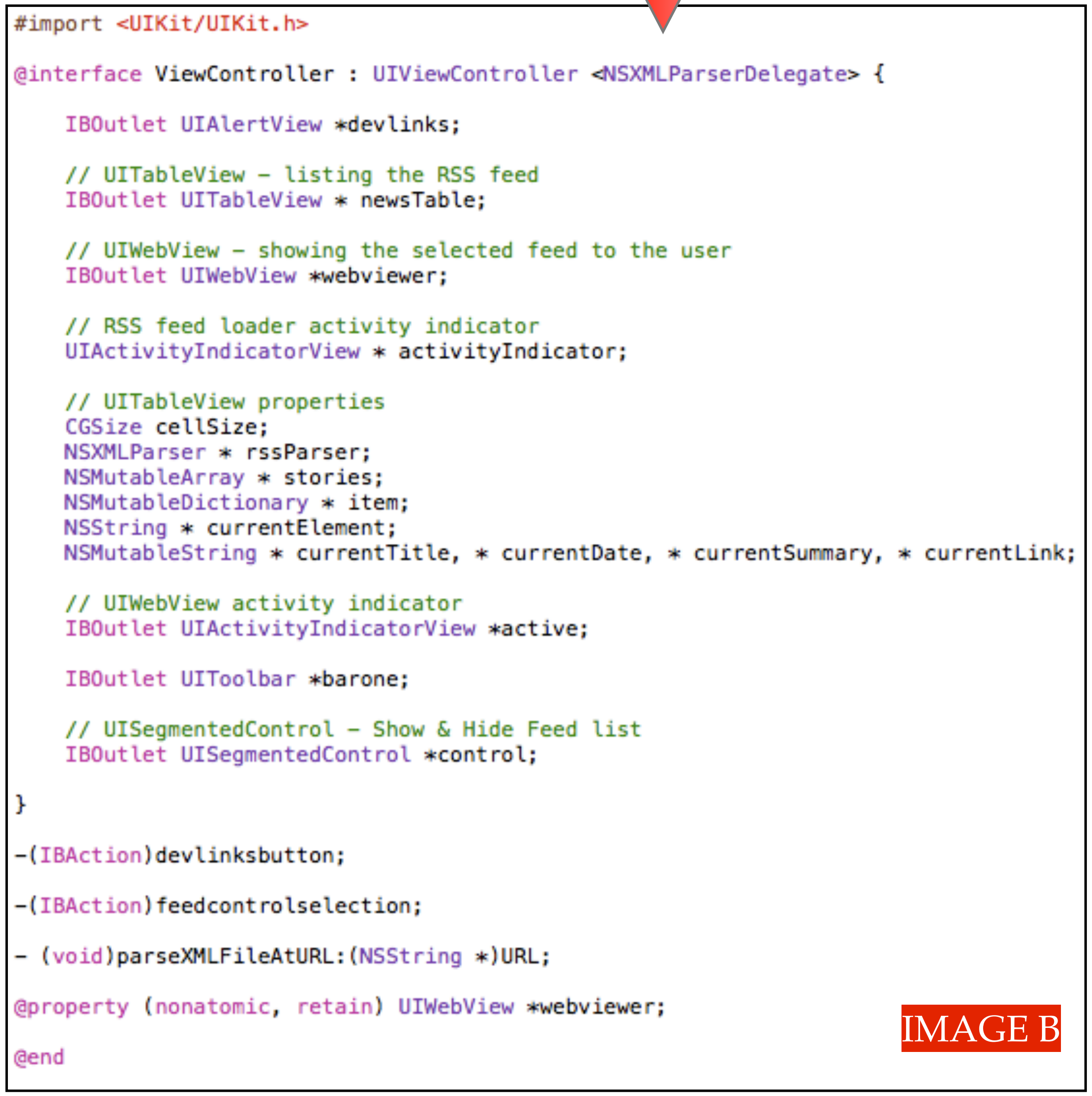


IMAGE B

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

@interface ViewController : UIViewController <NSXMLParserDelegate> {

    IBOutlet UIAlertView *devlinks;

    // UITableView - listing the RSS feed
    IBOutlet UITableView * newsTable;

    // UIWebView - showing the selected feed to the user
    IBOutlet UIWebView *webviewer;

    // RSS feed loader activity indicator
    UIActivityIndicatorView * activityIndicator;

    // UITableView properties
    CGSize cellSize;
    NSXMLParser * rssParser;
    NSMutableArray * stories;
    NSMutableDictionary * item;
    NSString * currentElement;
    NSMutableString * currentTitle, * currentDate, * currentSummary, * currentLink;

    // UIWebView activity indicator
    IBOutlet UIActivityIndicatorView *active;

    IBOutlet UIToolbar *barone;

    // UISegmentedControl - Show & Hide Feed list
    IBOutlet UISegmentedControl *control;

}

-(IBAction)devlinksbutton;

-(IBAction)feedcontrolselection;

- (void)parseXMLFileAtURL:(NSString *)URL;

@property (nonatomic, retain) UIWebView *webviewer;

@end
```

Once you have done that you can move onto the coding for the implementation file (.m file).

Step One

This is what your implementation file will look like to begin with (Image A). You need to delete the code currently in the implementation file and replace it with the code provided in ".m code" file (provided in the iRSS "Code" folder) into your implementation file so that your implementation file will look like this: (Image B).

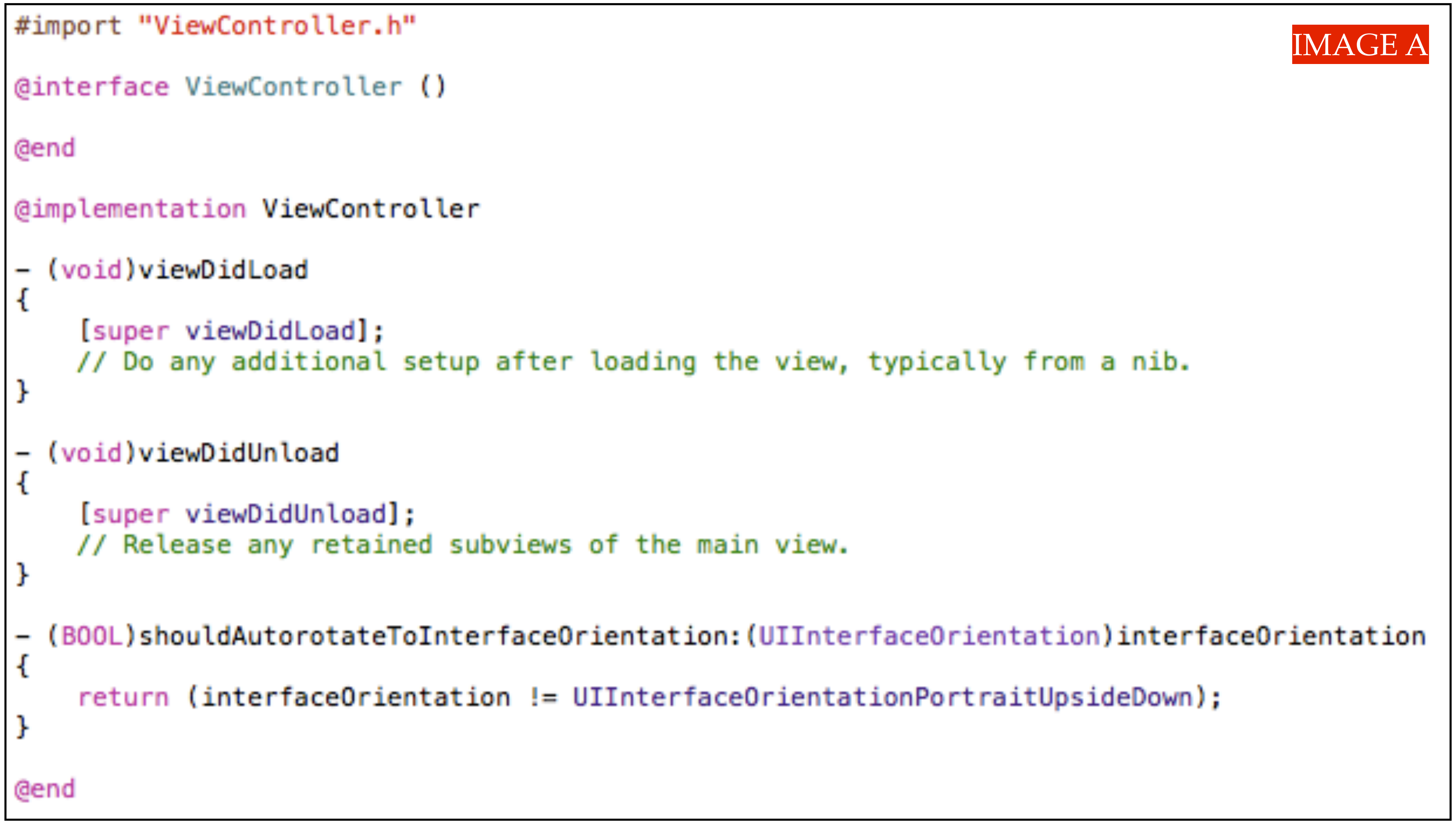


IMAGE A

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
}

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

@end
```

P.T.O for Image B -->

Step One

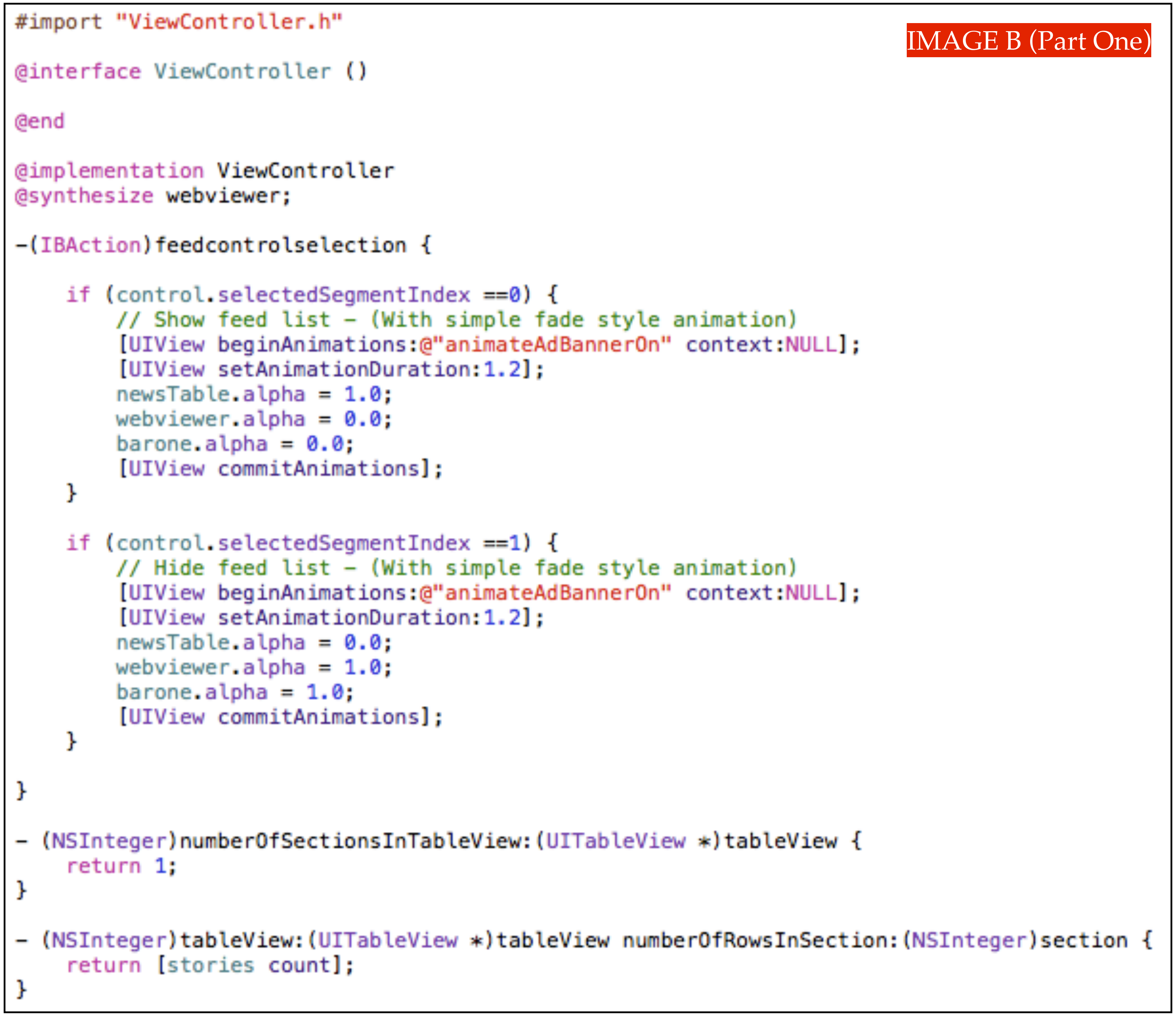
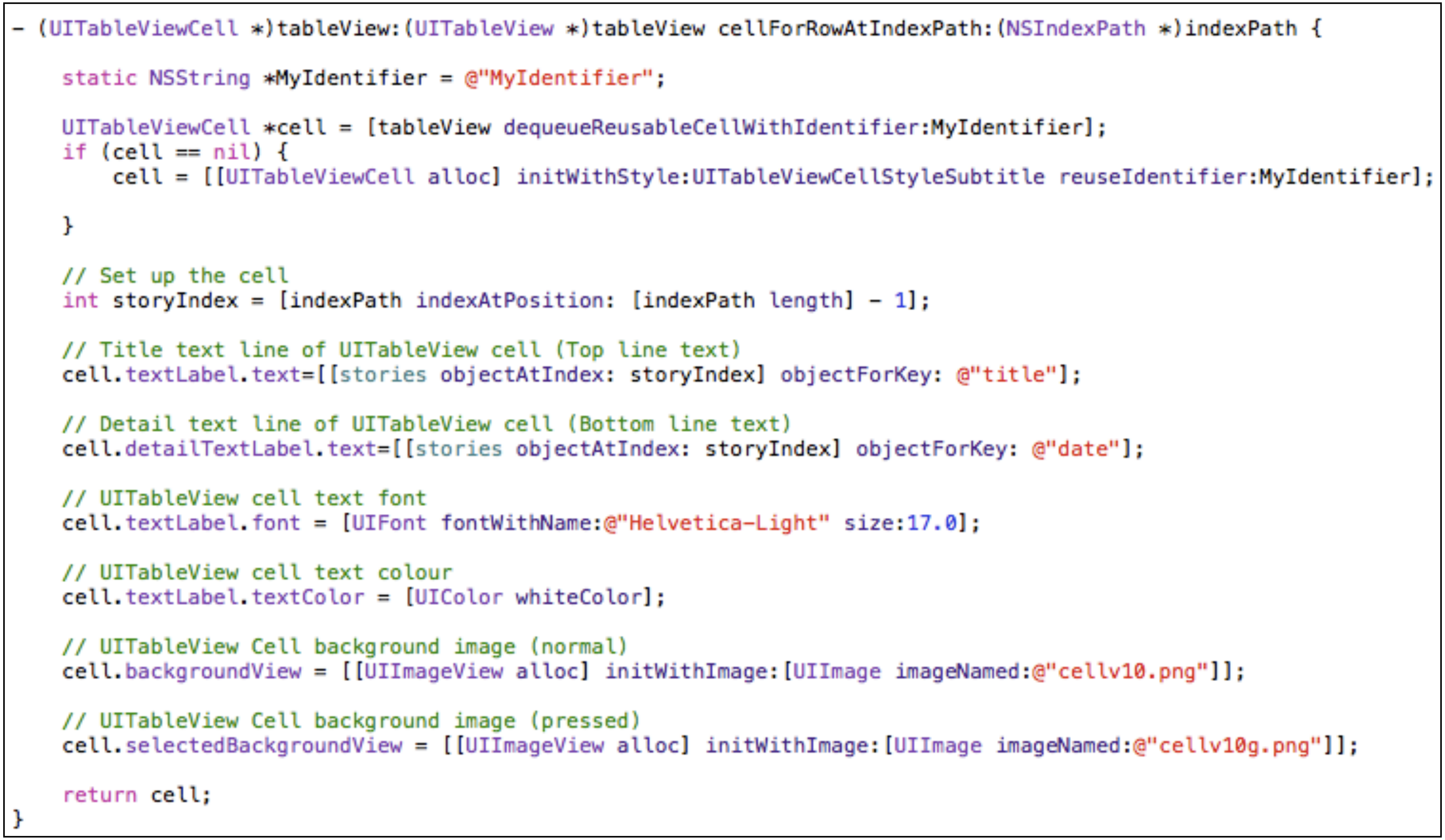


IMAGE B (Part One)

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController
@synthesize webviewer;

-(IBAction)feedcontrolselection {

    if (control.selectedSegmentIndex ==0) {
        // Show feed list - (With simple fade style animation)
        [UIView beginAnimations:@"animateAdBannerOn" context:NULL];
        [UIView setAnimationDuration:1.2];
        newsTable.alpha = 1.0;
        webviewer.alpha = 0.0;
        barone.alpha = 0.0;
        [UIView commitAnimations];
    }

    if (control.selectedSegmentIndex ==1) {
        // Hide feed list - (With simple fade style animation)
        [UIView beginAnimations:@"animateAdBannerOn" context:NULL];
        [UIView setAnimationDuration:1.2];
        newsTable.alpha = 0.0;
        webviewer.alpha = 1.0;
        barone.alpha = 1.0;
        [UIView commitAnimations];
    }

}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
    return [stories count];
}
```

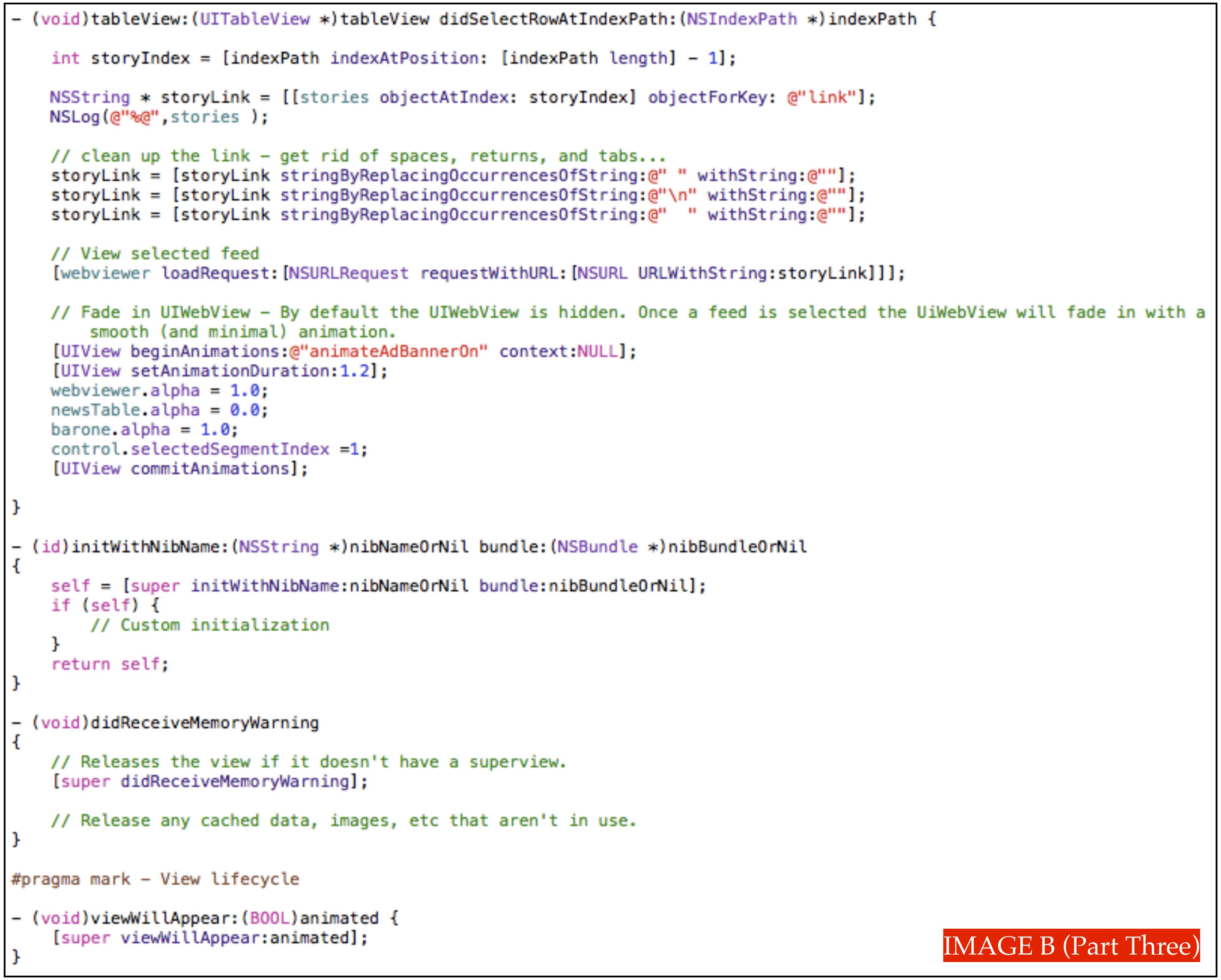
Step One

IMAGE B (Part Two)



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

    static NSString *MyIdentifier = @"MyIdentifier";

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:MyIdentifier];
    if (cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle reuseIdentifier:MyIdentifier];

    }

    // Set up the cell
    int storyIndex = [indexPath indexAtPosition: [indexPath length] - 1];

    // Title text line of UITableView cell (Top line text)
    cell.textLabel.text=[[stories objectAtIndex: storyIndex] objectForKey: @"title"];

    // Detail text line of UITableView cell (Bottom line text)
    cell.detailTextLabel.text=[[stories objectAtIndex: storyIndex] objectForKey: @"date"];

    // UITableView cell text font
    cell.textLabel.font = [UIFont fontWithName:@"Helvetica-Light" size:17.0];

    // UITableView cell text colour
    cell.textLabel.textColor = [UIColor whiteColor];

    // UITableView Cell background image (normal)
    cell.backgroundView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"cellv10.png"]];

    // UITableView Cell background image (pressed)
    cell.selectedBackgroundView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"cellv10g.png"]];

    return cell;
}
```

Step One



```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {

    int storyIndex = [indexPath indexAtPosition: [indexPath length] - 1];

    NSString * storyLink = [[stories objectAtIndex: storyIndex] objectForKey: @"link"];
    NSLog(@"%@",stories );

    // clean up the link - get rid of spaces, returns, and tabs...
    storyLink = [storyLink stringByReplacingOccurrencesOfString:@" " withString:@""];
    storyLink = [storyLink stringByReplacingOccurrencesOfString:@"\n" withString:@""];
    storyLink = [storyLink stringByReplacingOccurrencesOfString:@"  " withString:@""];

    // View selected feed
    [webviewer loadRequest:[NSURLRequest requestWithURL:[NSURL URLWithString:storyLink]]];

    // Fade in UIWebView - By default the UIWebView is hidden. Once a feed is selected the UiWebView will fade in with a
        smooth (and minimal) animation.
    [UIView beginAnimations:@"animateAdBannerOn" context:NULL];
    [UIView setAnimationDuration:1.2];
    webviewer.alpha = 1.0;
    newsTable.alpha = 0.0;
    barone.alpha = 1.0;
    control.selectedSegmentIndex =1;
    [UIView commitAnimations];

}

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

#pragma mark - View lifecycle

- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
}
```

IMAGE B (Part Three)

Step One

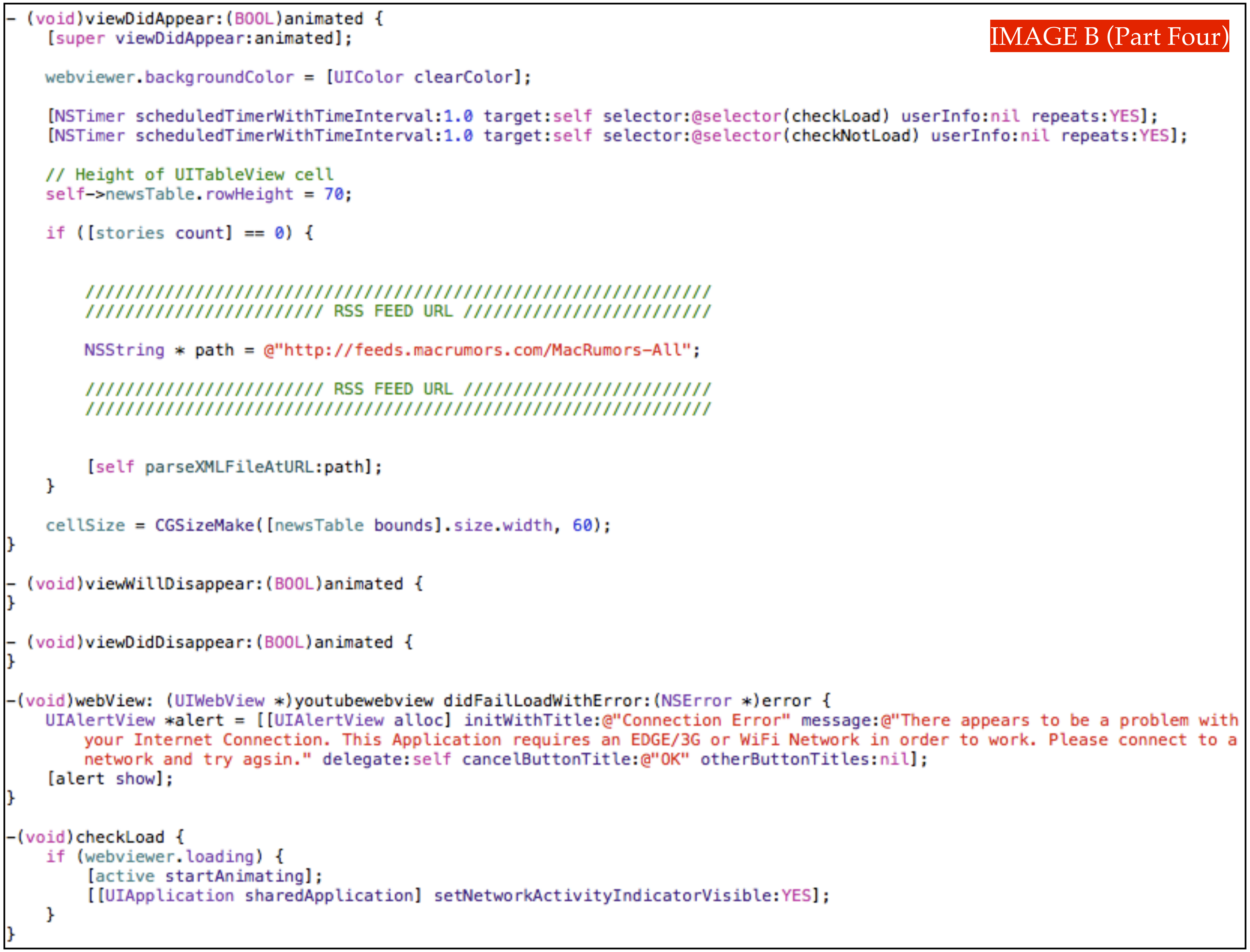
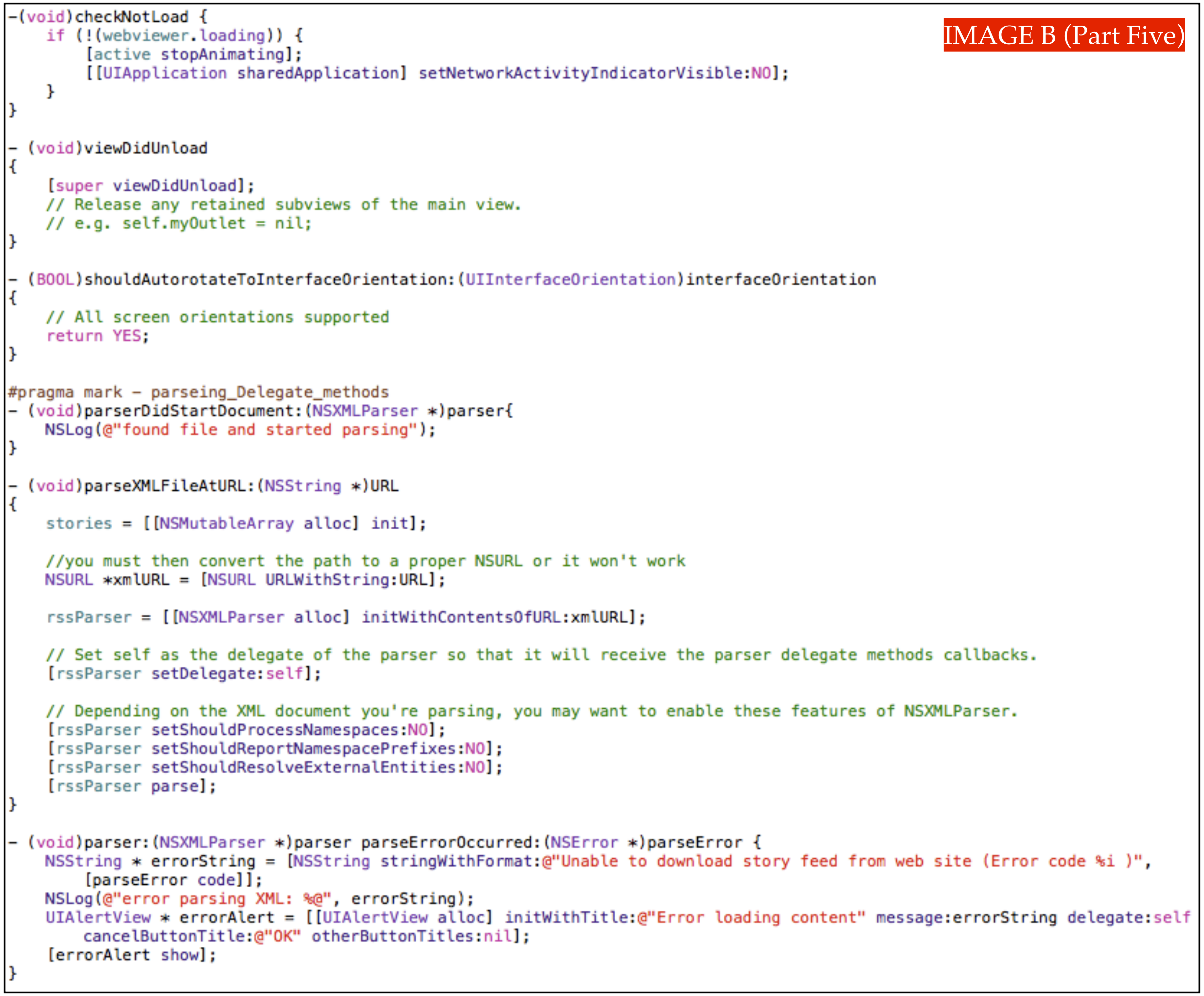


IMAGE B (Part Four)

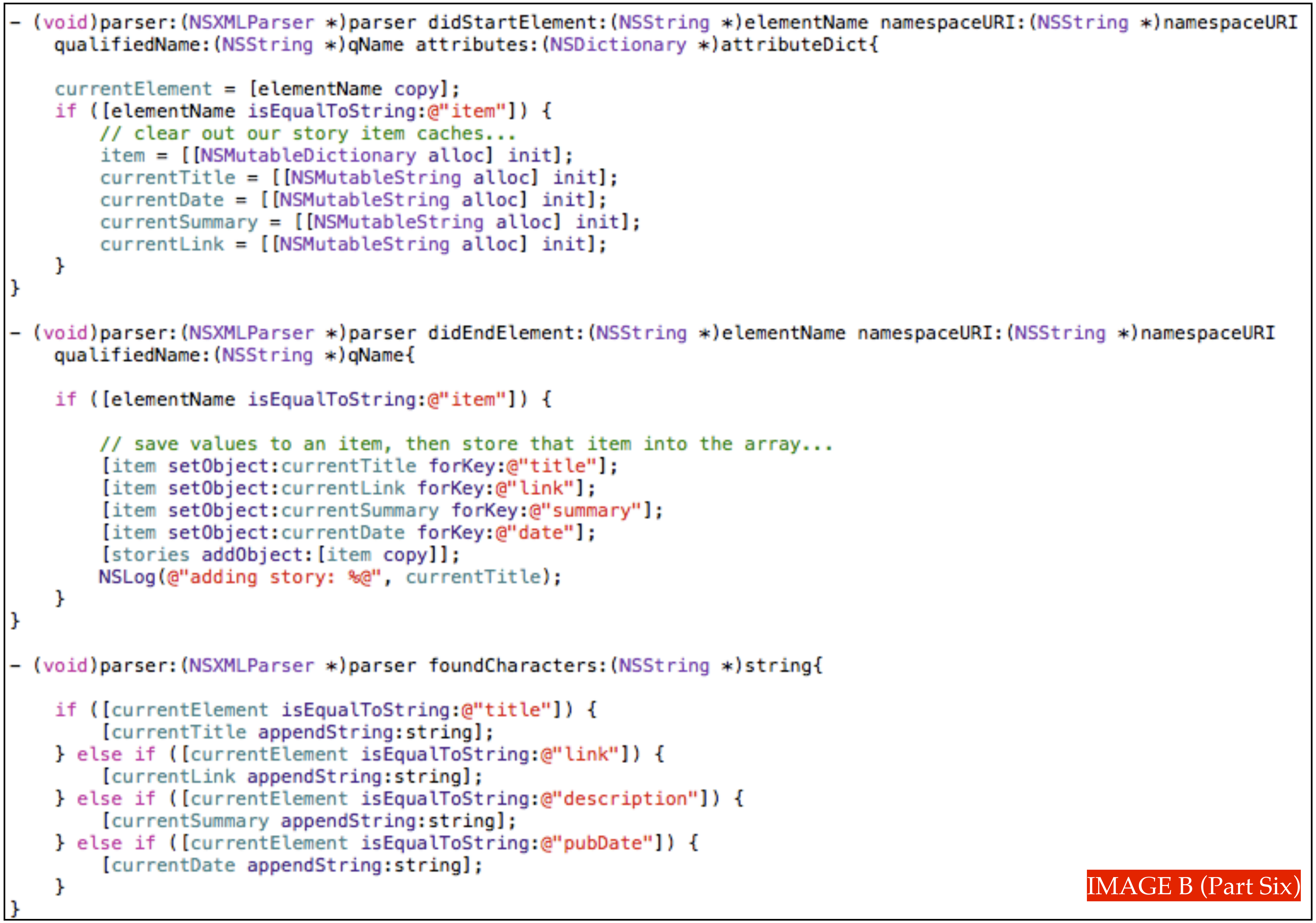
```
- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];

    webviewer.backgroundColor = [UIColor clearColor];

    [NSTimer scheduledTimerWithTimeInterval:1.0 target:self selector:@selector(checkLoad) userInfo:nil repeats:YES];
    [NSTimer scheduledTimerWithTimeInterval:1.0 target:self selector:@selector(checkNotLoad) userInfo:nil repeats:YES];

    // Height of UITableView cell
    self->newsTable.rowHeight = 70;

    if ([stories count] == 0) {


        ///////////////////////////////////////////////////////////////
        //////////////////////// RSS FEED URL ////////////////////////

        NSString * path = @"http://feeds.macrumors.com/MacRumors-All";

        //////////////////////// RSS FEED URL ////////////////////////
        ///////////////////////////////////////////////////////////////


        [self parseXMLFileAtURL:path];
    }

    cellSize = CGSizeMake([newsTable bounds].size.width, 60);
}

- (void)viewWillDisappear:(BOOL)animated {
}

- (void)viewDidDisappear:(BOOL)animated {
}

-(void)webView: (UIWebView *)youtubewebview didFailLoadWithError:(NSError *)error {
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Connection Error" message:@"There appears to be a problem with
        your Internet Connection. This Application requires an EDGE/3G or WiFi Network in order to work. Please connect to a
        network and try agsin." delegate:self cancelButtonTitle:@"OK" otherButtonTitles:nil];
    [alert show];
}

-(void)checkLoad {
    if (webviewer.loading) {
        [active startAnimating];
        [[UIApplication sharedApplication] setNetworkActivityIndicatorVisible:YES];
    }
}
```

Step One



```
-(void)checkNotLoad {
    if (!(webviewer.loading)) {
        [active stopAnimating];
        [[UIApplication sharedApplication] setNetworkActivityIndicatorVisible:NO];
    }
}

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

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    // All screen orientations supported
    return YES;
}

#pragma mark - parseing_Delegate_methods
- (void)parserDidStartDocument:(NSXMLParser *)parser{
    NSLog(@"found file and started parsing");
}

- (void)parseXMLFileAtURL:(NSString *)URL
{

    stories = [[NSMutableArray alloc] init];

    //you must then convert the path to a proper NSURL or it won't work
    NSURL *xmlURL = [NSURL URLWithString:URL];

    rssParser = [[NSXMLParser alloc] initWithContentsOfURL:xmlURL];

    // Set self as the delegate of the parser so that it will receive the parser delegate methods callbacks.
    [rssParser setDelegate:self];

    // Depending on the XML document you're parsing, you may want to enable these features of NSXMLParser.
    [rssParser setShouldProcessNamespaces:NO];
    [rssParser setShouldReportNamespacePrefixes:NO];
    [rssParser setShouldResolveExternalEntities:NO];
    [rssParser parse];
}

- (void)parser:(NSXMLParser *)parser parseErrorOccurred:(NSError *)parseError {
    NSString * errorString = [NSString stringWithFormat:@"Unable to download story feed from web site (Error code %i )",
        [parseError code]];
    NSLog(@"error parsing XML: %@", errorString);
    UIAlertView * errorAlert = [[UIAlertView alloc] initWithTitle:@"Error loading content" message:errorString delegate:self
        cancelButtonTitle:@"OK" otherButtonTitles:nil];
    [errorAlert show];
}
```

Step One



```
- (void)parser:(NSXMLParser *)parser didStartElement:(NSString *)elementName namespaceURI:(NSString *)namespaceURI
    qualifiedName:(NSString *)qName attributes:(NSDictionary *)attributeDict{

    currentElement = [elementName copy];
    if ([elementName isEqualToString:@"item"]) {
        // clear out our story item caches...
        item = [[NSMutableDictionary alloc] init];
        currentTitle = [[NSMutableString alloc] init];
        currentDate = [[NSMutableString alloc] init];
        currentSummary = [[NSMutableString alloc] init];
        currentLink = [[NSMutableString alloc] init];
    }
}

- (void)parser:(NSXMLParser *)parser didEndElement:(NSString *)elementName namespaceURI:(NSString *)namespaceURI
    qualifiedName:(NSString *)qName{

    if ([elementName isEqualToString:@"item"]) {

        // save values to an item, then store that item into the array...
        [item setObject:currentTitle forKey:@"title"];
        [item setObject:currentLink forKey:@"link"];
        [item setObject:currentSummary forKey:@"summary"];
        [item setObject:currentDate forKey:@"date"];
        [stories addObject:[item copy]];
        NSLog(@"adding story: %@", currentTitle);
    }
}

- (void)parser:(NSXMLParser *)parser foundCharacters:(NSString *)string{

    if ([currentElement isEqualToString:@"title"]) {
        [currentTitle appendString:string];
    } else if ([currentElement isEqualToString:@"link"]) {
        [currentLink appendString:string];
    } else if ([currentElement isEqualToString:@"description"]) {
        [currentSummary appendString:string];
    } else if ([currentElement isEqualToString:@"pubDate"]) {
        [currentDate appendString:string];
    }
}
```

IMAGE B (Part Six)

Step One

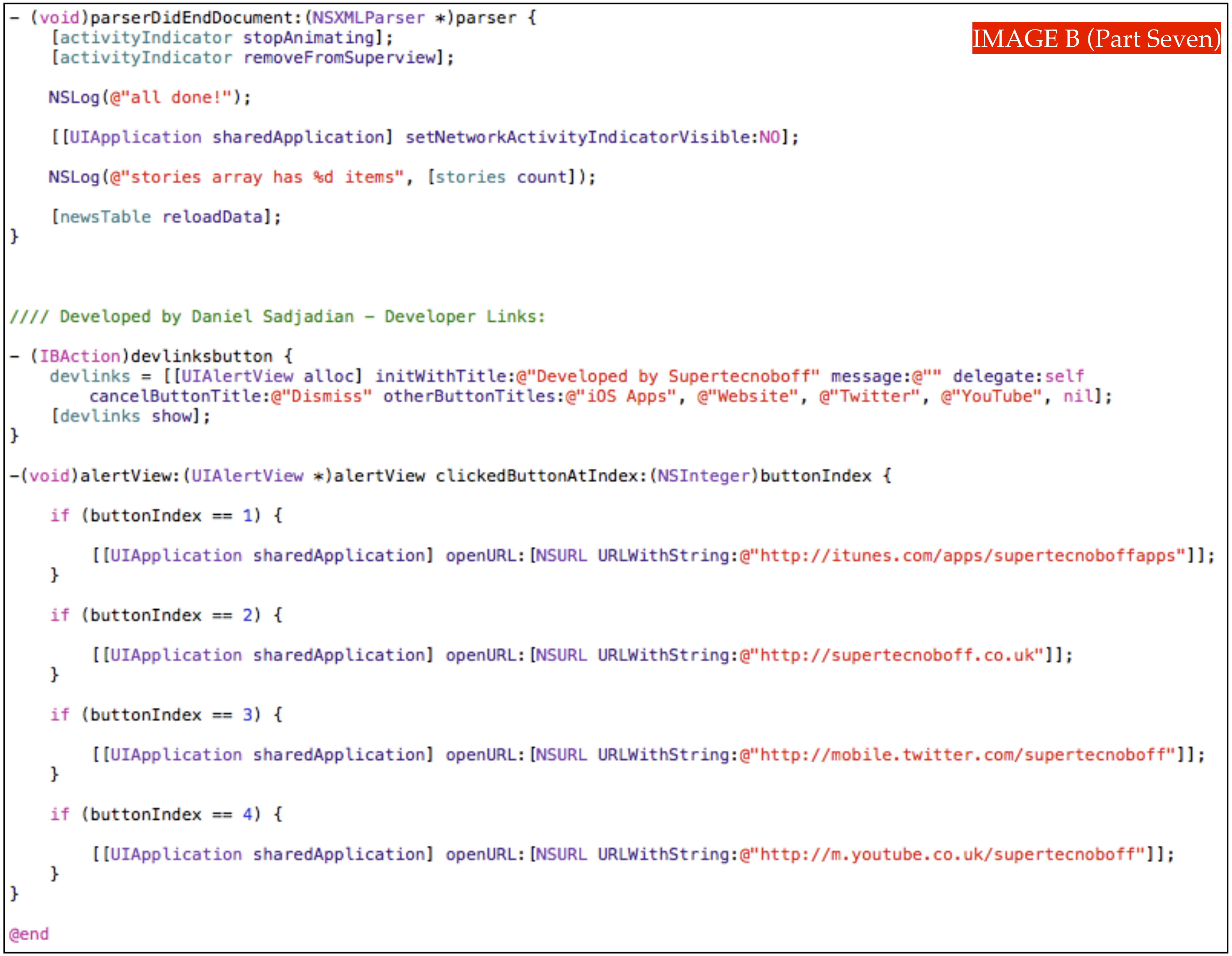


IMAGE B (Part Seven)

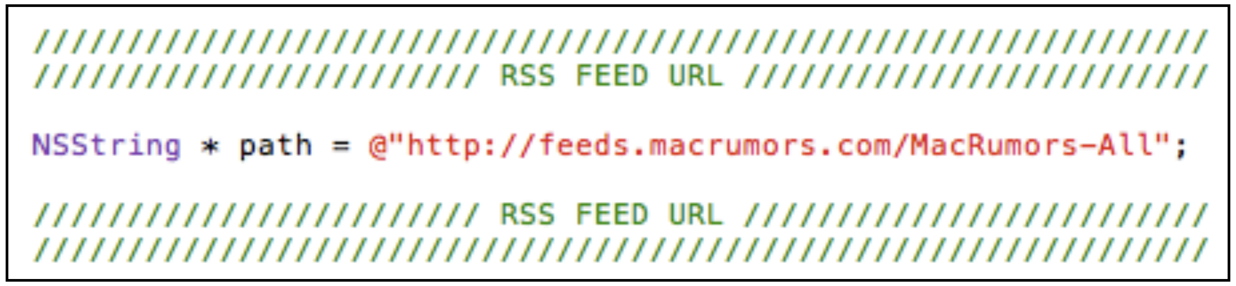
```
- (void)parserDidEndDocument:(NSXMLParser *)parser {
    [activityIndicator stopAnimating];
    [activityIndicator removeFromSuperview];

    NSLog(@"all done!");

    [[UIApplication sharedApplication] setNetworkActivityIndicatorVisible:NO];

    NSLog(@"stories array has %d items", [stories count]);

    [newsTable reloadData];
}



//// Developed by Daniel Sadjadian - Developer Links:

- (IBAction)devlinksbutton {
    devlinks = [[UIAlertView alloc] initWithTitle:@"Developed by Supertecnoboff" message:@"" delegate:self
        cancelButtonTitle:@"Dismiss" otherButtonTitles:@"iOS Apps", @"Website", @"Twitter", @"YouTube", nil];
    [devlinks show];
}

-(void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex {

    if (buttonIndex == 1) {

        [[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"http://itunes.com/apps/supertecnoboffapps"]];
    }

    if (buttonIndex == 2) {

        [[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"http://supertecnoboff.co.uk"]];
    }

    if (buttonIndex == 3) {

        [[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"http://mobile.twitter.com/supertecnoboff"]];
    }

    if (buttonIndex == 4) {

        [[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"http://m.youtube.co.uk/supertecnoboff"]];
    }
}

@end
```

Step One (almost done)

Now we need to set a few things:

- The RSS feed URL you want to load
- The UITableView appearance

If you scroll down in your ViewController.m file you'll find a section of code in which you can add your RSS feed URL:



```
///////////////////////////////////////////////////////////////
//////////////////////// RSS FEED URL /////////////////////////

NSString * path = @"http://feeds.macrumors.com/MacRumors-All";

//////////////////////// RSS FEED URL /////////////////////////
///////////////////////////////////////////////////////////////
```

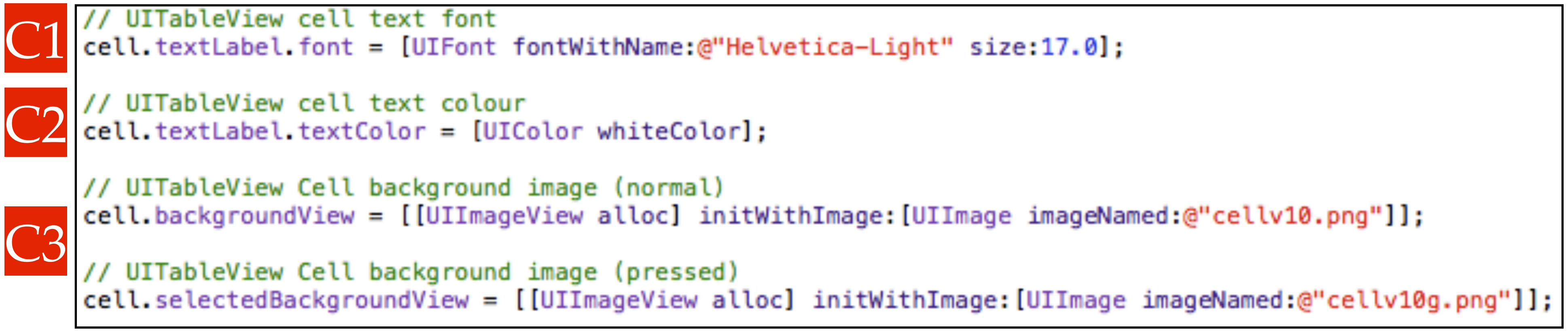
Replace the current URL with your chosen RSS feed URL.

Customizing the UITableView cells:

You may have noticed that the UITableView cells don't look like the normal white cells you find in standard UITableView; this is because I have customized them by removing the default white background and replacing it with my own images. I also changed the text font and and size. You can do this too and it's really easy.

Step One (almost done)

Find the section of code which looks like this (in your ViewController.m file):

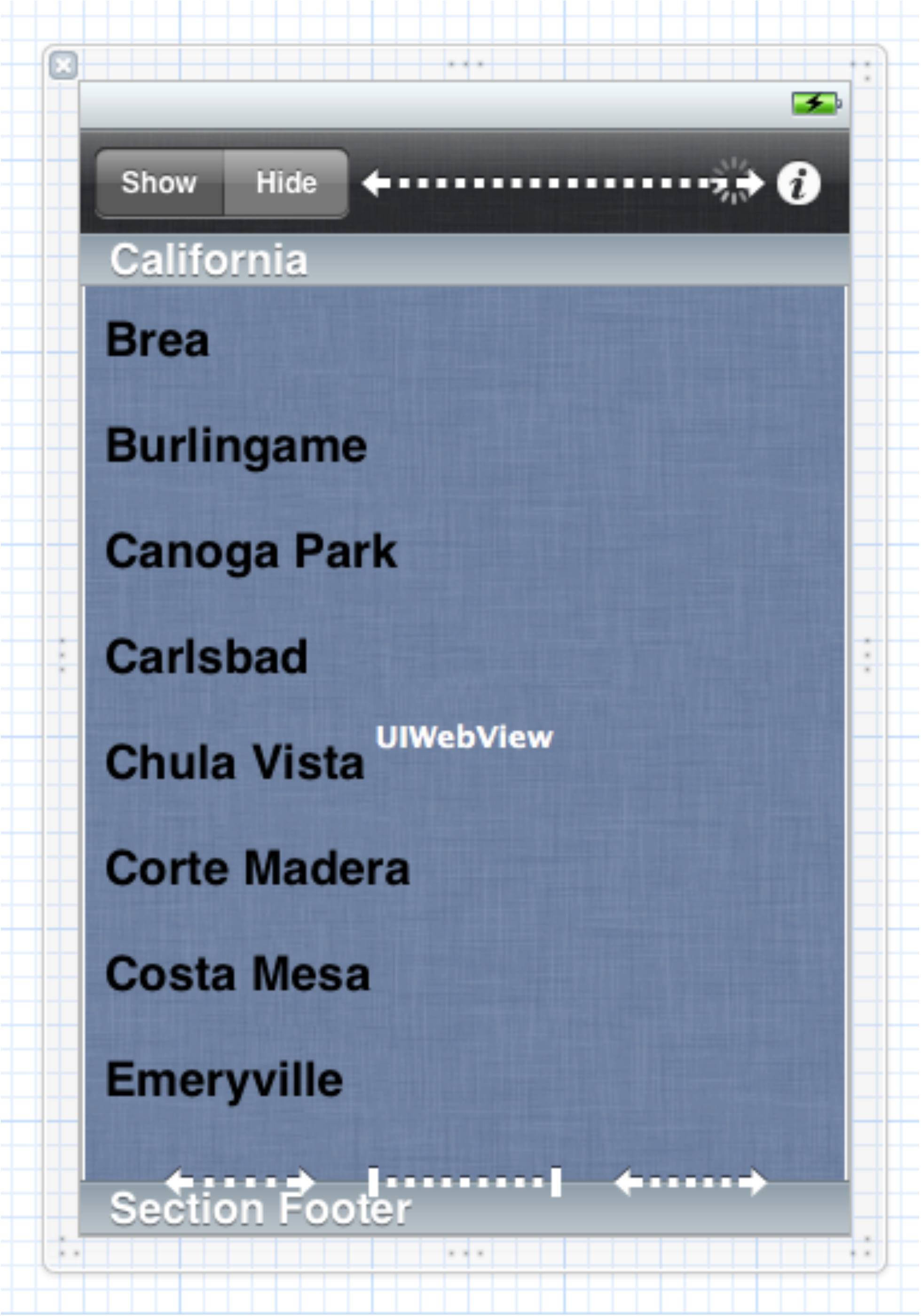


```
// UITableView cell text font
cell.textLabel.font = [UIFont fontWithName:@"Helvetica-Light" size:17.0];

// UITableView cell text colour
cell.textLabel.textColor = [UIColor whiteColor];

// UITableView Cell background image (normal)
cell.backgroundView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"cellv10.png"]];

// UITableView Cell background image (pressed)
cell.selectedBackgroundView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"cellv10g.png"]];
```

You can see that each line of code here has a different task. (C1) - The top line sets the font (and size) of the text which appears in each UITableView cell. You can customize the font to what ever you want. (C2) - The next line of code below that sets the font colour, once again you can change this to what ever you want. (C3) - The next two lines of code below that control the background image of each UITableView cell. Change the image names to you own custom images; or if you wish to use the standard white background of the UITableView, simply delete the bottom lines of code.

And now we can move on to the User Interface design. (Step Two) - YAY :D

Step Two



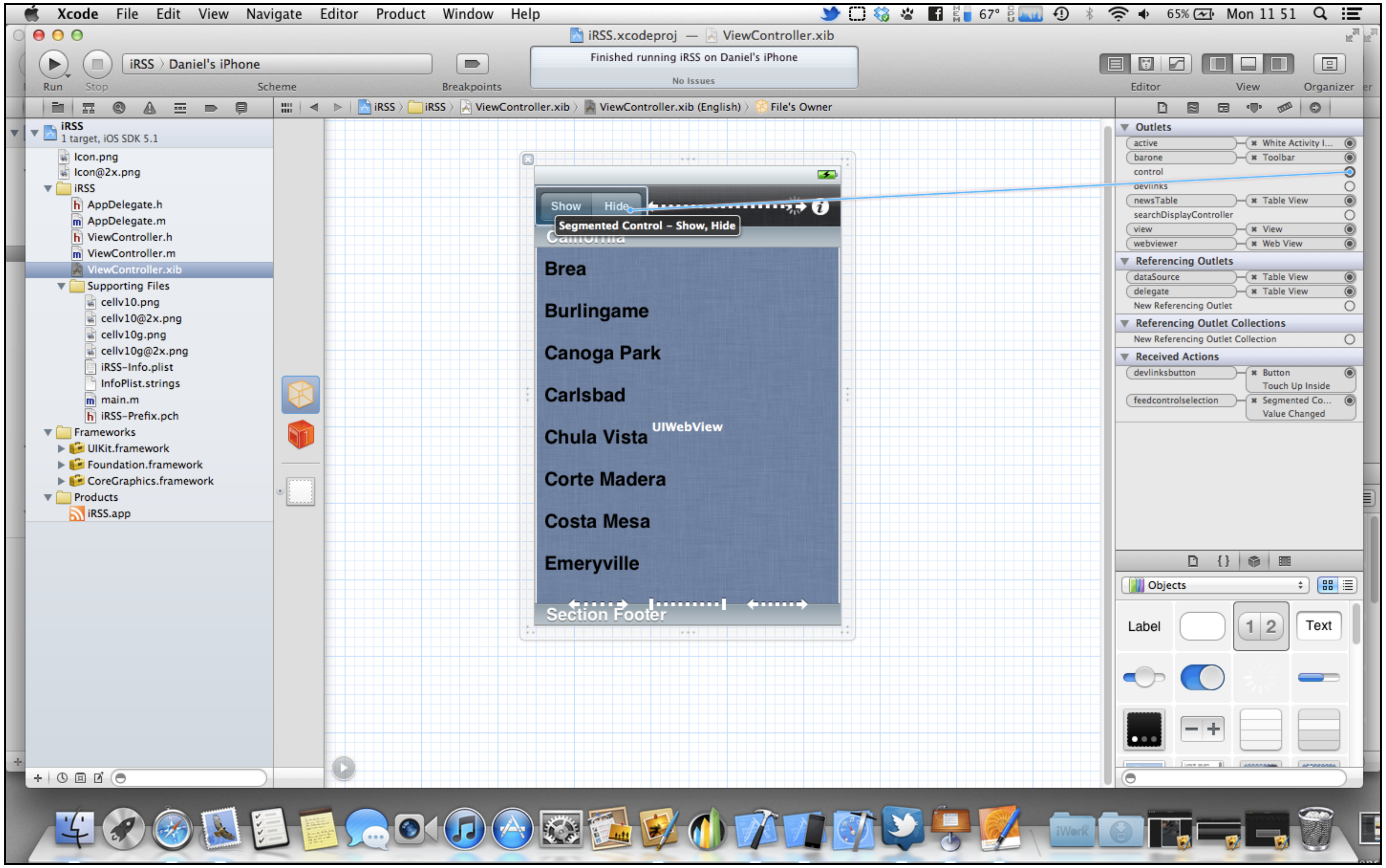
We have now successfully coded our RSS application. We need to now work on the User Interface. Click “ViewController.xib” and we can begin designing our interface.

Start by dragging in a UIWebView and a UIToolBar. Put the UIToolBar at the top of the page. Then drag in a UITableView.

Now and drag in a UISegmentedControl and place it on your UIToolBar. Then drag in a UIActivityIndicator and place it where ever you want.

Now open the connections tab and we can start connecting our code to our User Interface elements.

Step Two



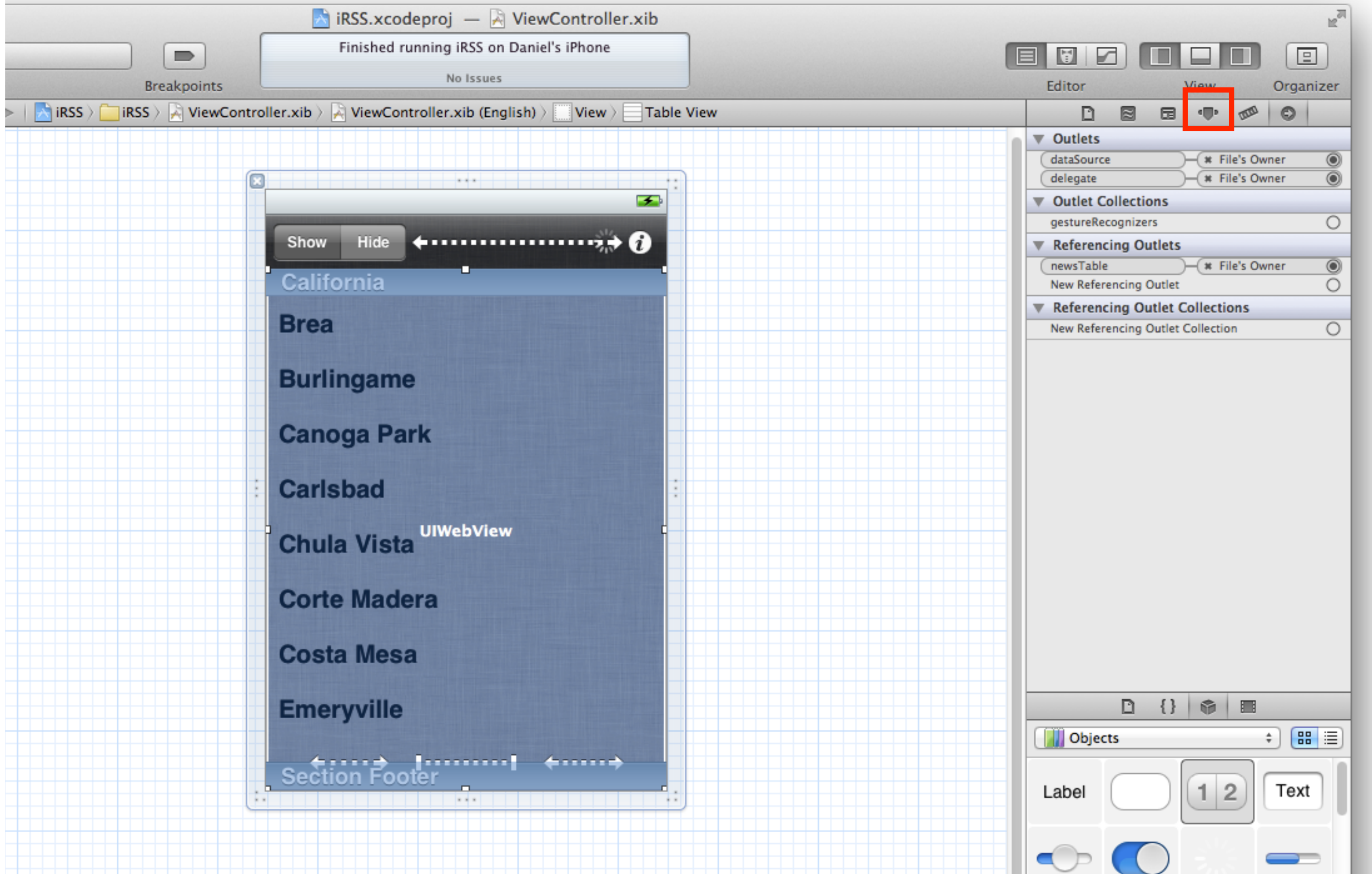
Now select the "Files Owner" and then make sure the connections tab is open. Now connect "active" to your UIActivityIndicator, connect "control" to your UISegmentedControl, connect "newsTable" to your UITableView and connect "feedcontrolselection" to your UISegmentedControl. (Once you connect "feedcontrolselection" to your UISegmentedControl, a menu will pop up; make sure you select "ValueChanged" on this pop up menu).

Step Two

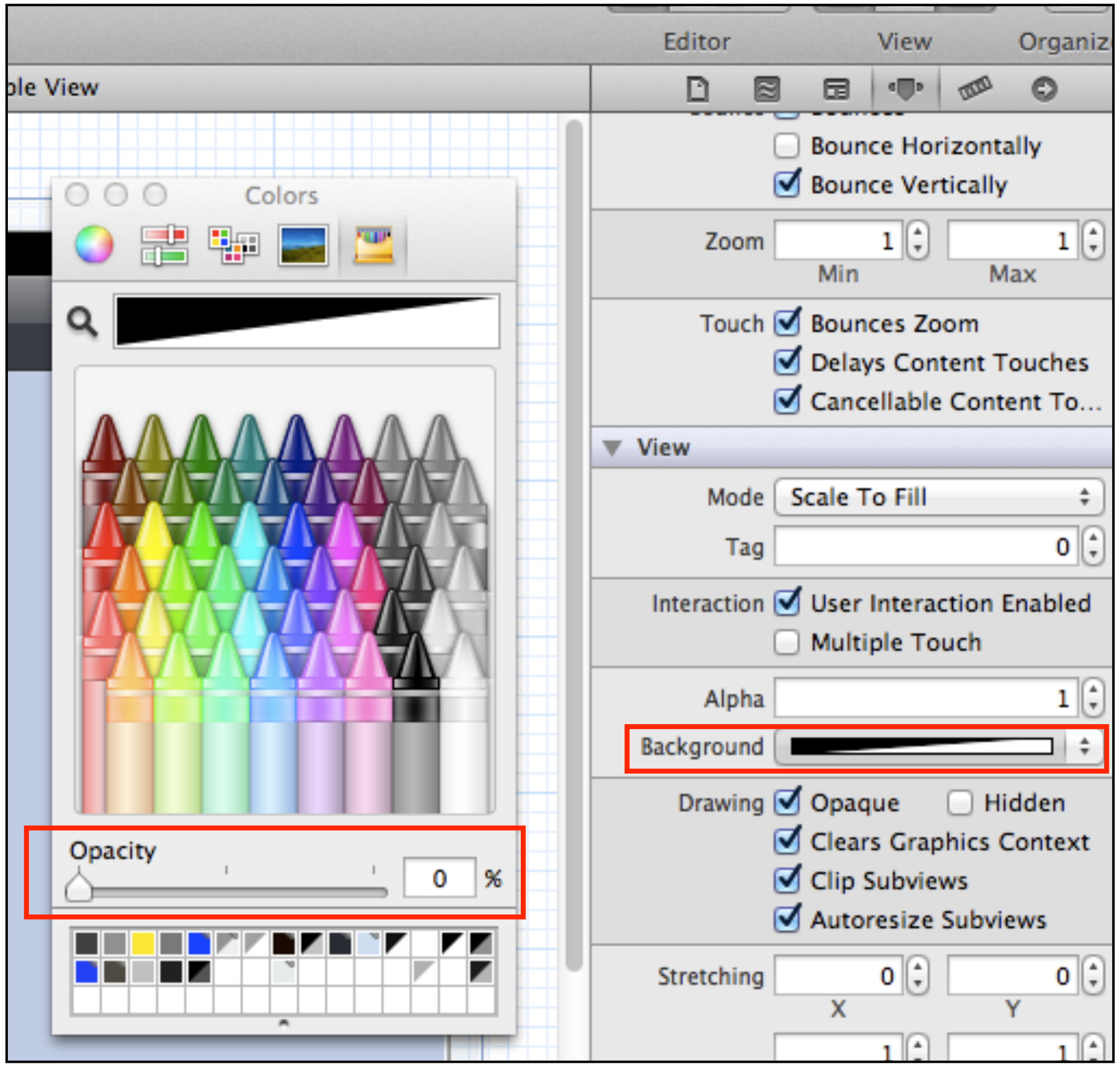
And finally select your UITableView and then press and hold the "ctrl" button on your keyboard and connect it to the "Files Owner", a menu will now pop up, select "dateSource", then repeat this step and select "delegate".

Now remember earlier on in this guide we talked about customizing the UITableView cell; well we now need to make a few changes to the User Interface accordingly. So, if you decided that you want the standard white UITableView cells then you can forget this bit and simply build and run the app and your done :D - However, if you decided that you want your own background images for your UITableView cells then you need to continue with this step.

So click on the UITableView and then open the "Attributes Inspector":



Step Two



☐ Bounce Horizontally
☑ Bounce Vertically

Zoom 1 1
Min Max

Touch ☑ Bounces Zoom
☑ Delays Content Touches
☑ Cancellable Content To...

View

Mode Scale To Fill

Tag 0

Interaction ☑ User Interaction Enabled
☐ Multiple Touch

Alpha 1

Background

Drawing ☑ Opaque ☐ Hidden
☑ Clears Graphics Context
☑ Clip Subviews
☑ Autoresize Subviews

Stretching 0 0
X Y
1 1

Opacity 0 %

Now if you decided that you do want your own custom images as the background for your UITableView cells then you need to change the background colour of the UITableView to transparent so that your background images can be seen. Simply click on the background box and a colour palette will pop up. Select any colour you want (it doesn't which one, just select one) and the drag the "Opacity" slider to 0%. This will make the background of your UITableView cells transparent.

And thats it. Now the background of your UITableView cells will be filled with the image you stated in the "ViewController.m" file.

And thats it. We can move on to step 3 :D

Step Three

So if you did decide to implement your own background images for the UITableView cells, now is the time to import them into your Xcode project.

Once you have done that, you can give yourself a round of applause and build and Run the app :D

Enjoy :D