iPhone Apps Development using Objective C & Xcode (Lesson 6)

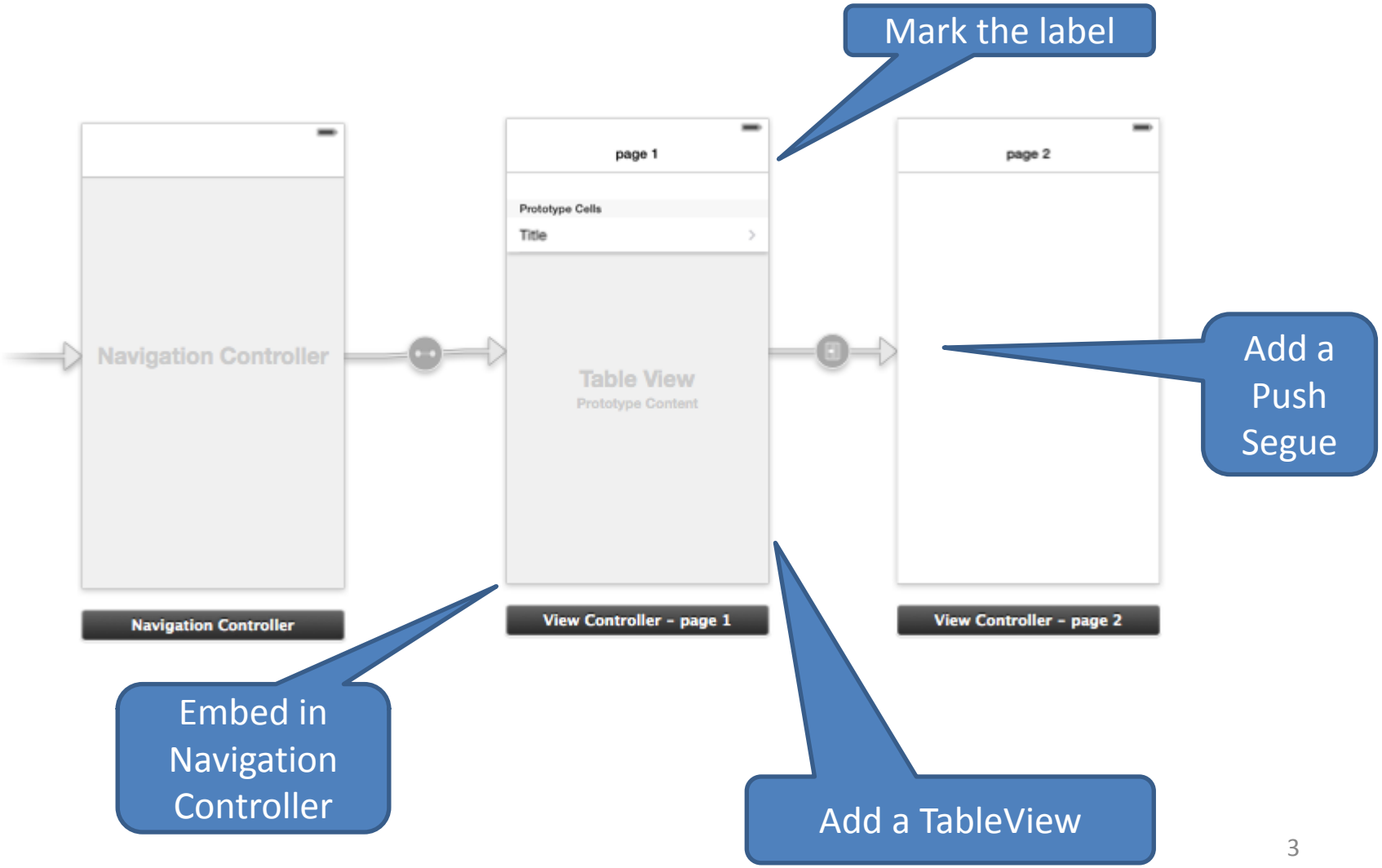
By Dannis Mok

TableView vs TableViewController

- TableView
  - Need to declare the protocol UITableViewDataSource and UITableViewDelegate in the ViewController header
  - Need to define both delegates of these protocol to be itself.
  - Need to define the required methods

- TableViewController
  - No need to declare the protocol
  - No need to define the delegates
  - The required method templates are listed.

Build a Master Detail Apps from Scratch using TableView



Build a Master Detail Apps from Scratch using TableView

```
//
//  ViewController.h
//  Master4
//
//  Created by Dannis Mok on 24/9/14.
//  Copyright (c) 2014 com.uec. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController <UITableViewDelegate, UITableViewDataSource>

@end
```

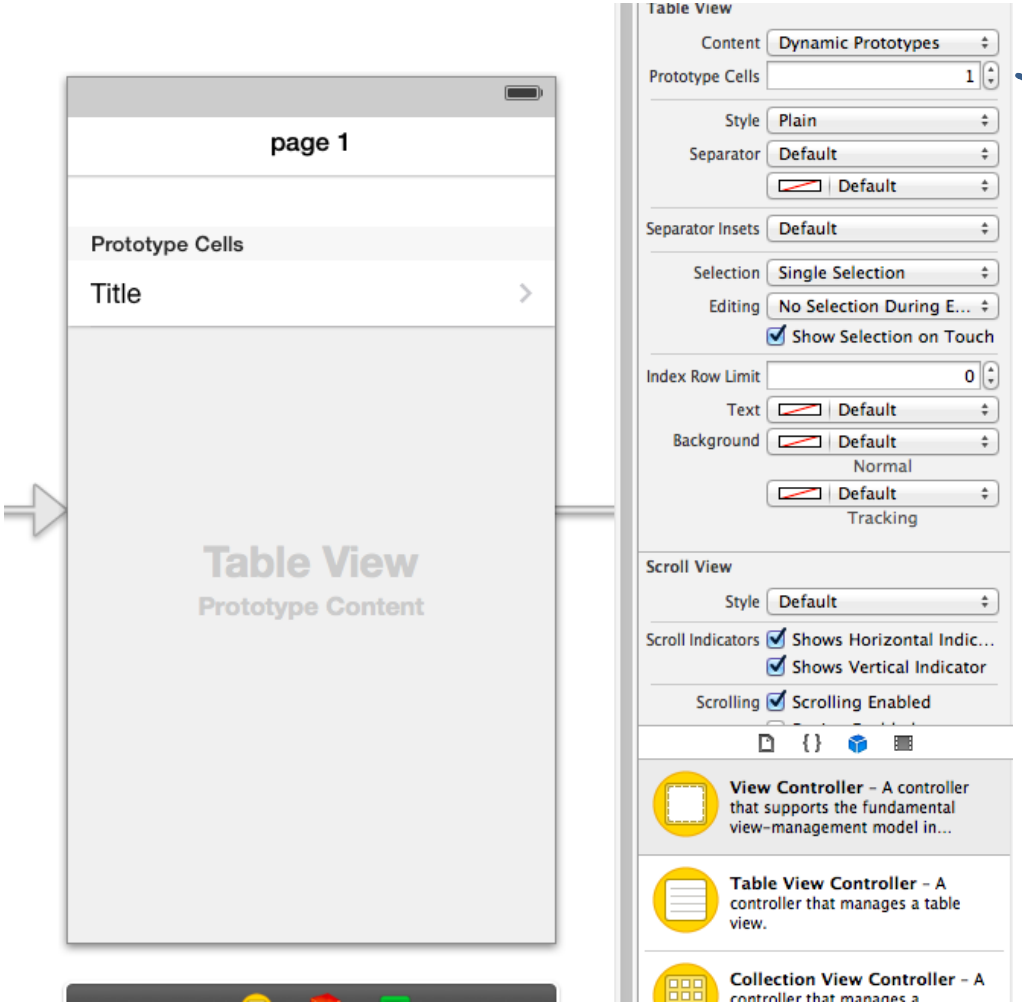
Add 2 protocols in the header

Build a Master Detail Apps from Scratch using TableView

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

    return 5;
}


- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)
    indexPath
{
   UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Cell" forIndexPath:
       indexPath];


    cell.textLabel.text = @"Testing";

    return cell;
}
```

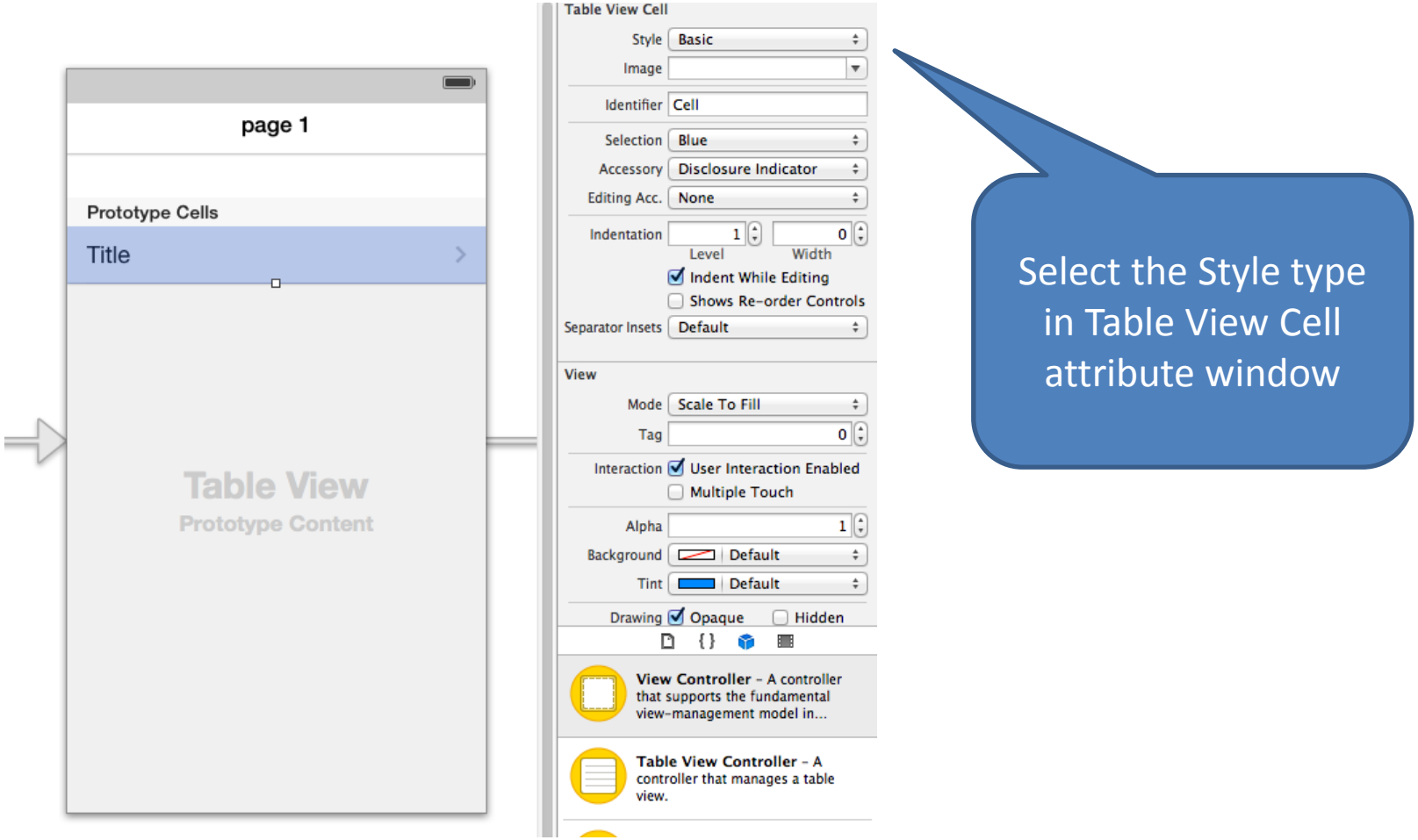
Add 2 required methods

Build a Master Detail Apps from Scratch using TableView

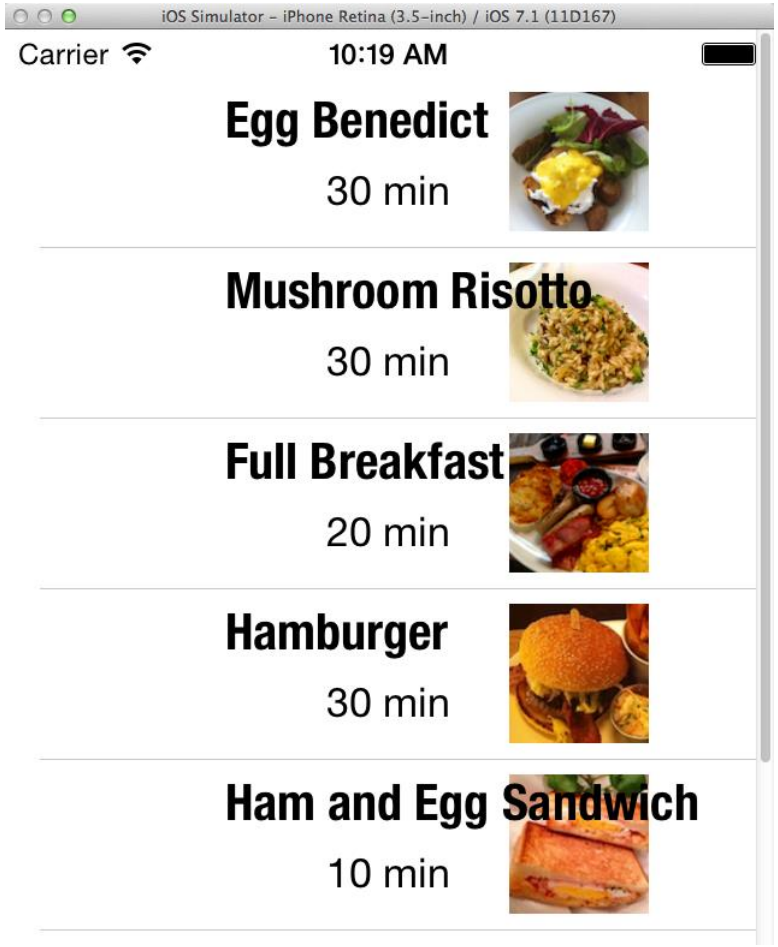


Make sure 1 is added here in TableView attribute window

Build a Master Detail Apps from Scratch using TableView



Custom made Cell



Custom made cell format must be supported by a custom UITableViewCell class

UITableViewCell

- Create a new Objective-C class file inherited from UITableViewCell. Define 2 UILabels and 1 UIImageView there

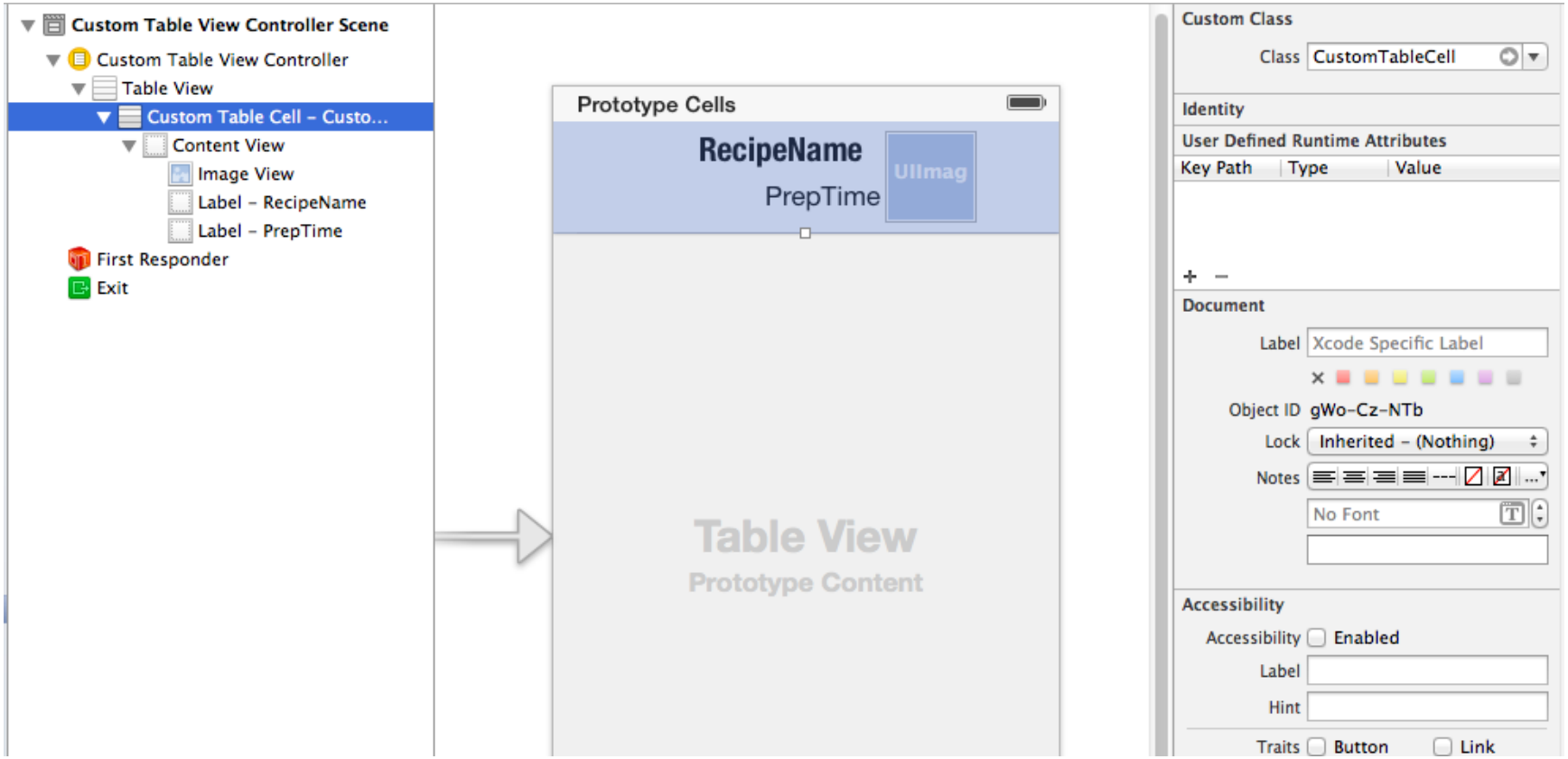
```
#import <UIKit/UIKit.h>

@interface CustomTableCell : UITableViewCell

@property (nonatomic, weak) IBOutlet UILabel *nameLabel;
@property (nonatomic, weak) IBOutlet UILabel *prepTimeLabel;
@property (nonatomic, weak) IBOutlet UIImageView *thumbnailImageView;

@end
```

Header file of the CustomTableCell

UITableViewCell

- Implementation file has no changes

```
#import "CustomTableCell.h"

@implementation CustomTableCell

- (id)initWithStyle:(UITableViewCellStyle)style reuseIdentifier:(NSString *)
    reuseIdentifier
{
    self = [super initWithStyle:style reuseIdentifier:reuseIdentifier];
    if (self) {
        // Initialization code
    }
    return self;
}

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

    // Configure the view for the selected state
}
```


Custom Table View Controller Scene
Custom Table View Controller
Table View
Custom Table Cell – Custo...
Content View
Image View
Label – RecipeName
Label – PrepTime
First Responder
Exit
Prototype Cells
RecipeName
PrepTime
UIImag
Table View
Prototype Content
Custom Class
Class CustomTableCell
Identity
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label
Xcode Specific Label
Object ID gWo-Cz-NTb
Lock
Inherited – (Nothing)
Notes
No Font
Accessibility
Accessibility
Enabled
Label
Hint
Traits
Button
Link

Use the CustomViewCell in the ViewController Class

```
#import "CustomTableViewController.h"
#import "CustomTableCell.h"

@interface CustomTableViewController ()

@end

@implementation CustomTableViewController
{
    NSArray *recipeNames;
    NSArray *recipeImages;
    NSArray *recipePrepTimes;
}
```

Have defined 3 arrays to save information

Use the CustomViewCell in the ViewController Class

Information are added into the array

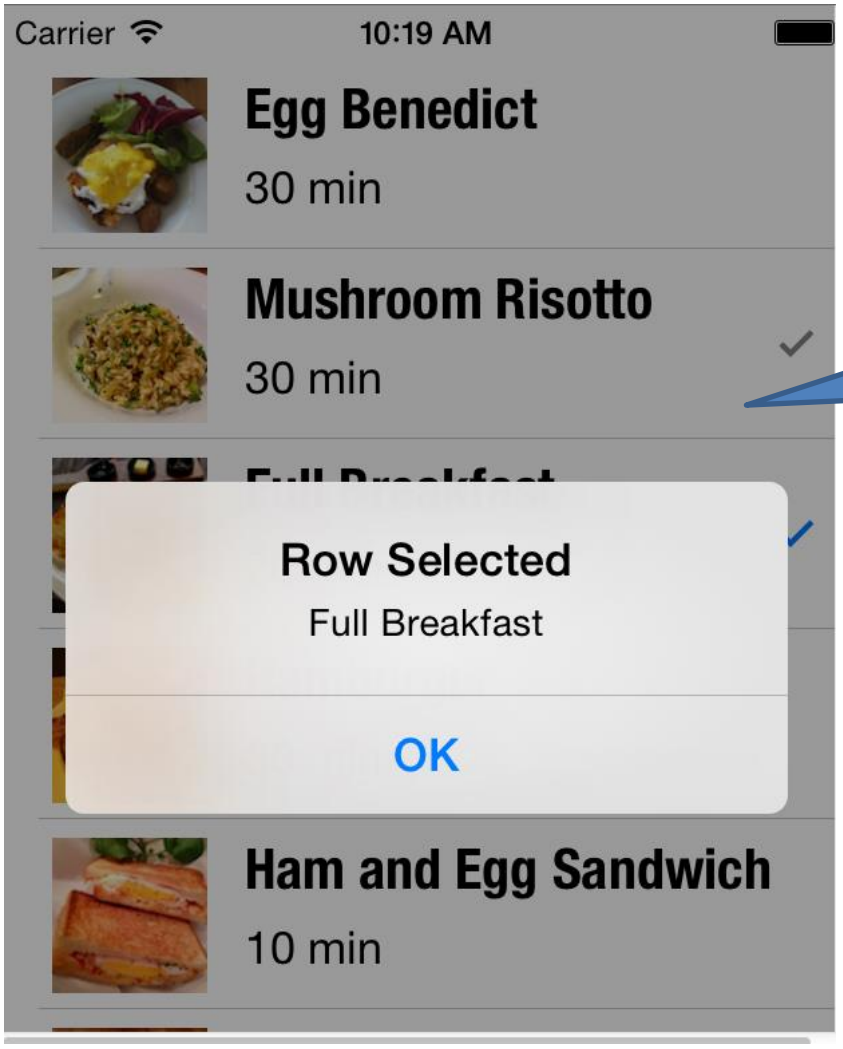
```
- (void)viewDidLoad
{
    [super viewDidLoad];

    recipeNames = [NSArray arrayWithObjects:@"Egg Benedict", @"Mushroom Risotto"
        , @"Full Breakfast", @"Hamburger", @"Ham and Egg Sandwich", @"Creme
        Brelee", @"White Chocolate Donut", @"Starbucks Coffee", @"Vegetable
        Curry", @"Instant Noodle with Egg", @"Noodle with BBQ Pork", @"Japanese
        Noodle with Pork", @"Green Tea", @"Thai Shrimp Cake", @"Angry Birds
        Cake", @"Ham and Cheese Panini", nil];

    recipeImages = [NSArray arrayWithObjects:@"egg_benedict.jpg",
        @"mushroom_risotto.jpg", @"full_breakfast.jpg", @"hamburger.jpg",
        @"ham_and_egg_sandwich.jpg", @"creme_brelee.jpg",
        @"white_chocolate_donut.jpg", @"starbucks_coffee.jpg",
        @"vegetable_curry.jpg", @"instant_noodle_with_egg.jpg",
        @"noodle_with_bbq_pork.jpg", @"japanese_noodle_with_pork.jpg",
        @"green_tea.jpg", @"thai_shrimp_cake.jpg", @"angry_birds_cake.jpg",
        @"ham_and_cheese_panini.jpg", nil];

    recipePrepTimes = [NSArray arrayWithObjects:@"30 min", @"30 min", @"20 min",
        @"30 min", @"10 min", @"1 hour", @"45 min", @"5 min", @"30 min", @"8
        min", @"20 min", @"20 min", @"5 min", @"1.5 hour", @"4 hours", @"10 min"
        , nil];
```

Use the CustomViewCell in the ViewController Class

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

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
    (NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"CustomTableCell";
    CustomTableCell *cell = (CustomTableCell *)[tableView
        dequeueReusableCellWithIdentifier:CellIdentifier];

    // Configure the cell...
    if (cell == nil) {
        cell = [[CustomTableCell alloc] initWithStyle:
            UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
    }

    // Display recipe in the table cell
    cell.nameLabel.text = [recipeNames objectAtIndex:indexPath.row];
    cell.thumbnailImageView.image = [UIImage imageNamed:[recipeImages
        objectAtIndex:indexPath.row]];
    cell.prepTimeLabel.text = [recipePrepTimes objectAtIndex:indexPath.row];

    return cell;
}
```

Use the CustomViewCell in the 2 required methods

Table Row Selection



Row can be selected to open the alert box

Table Row Selection

```
#import "CustomTableViewController.h"
#import "CustomTableCell.h"

@interface CustomTableViewController ()

@end

@implementation CustomTableViewController
{
    NSArray *recipeNames;
    NSArray *recipeImages;
    NSArray *recipePrepTimes;
    BOOL recipeChecked[16];
}
```

A boolean array is added to keep track of whether the row is clicked or not

Table Row Selection

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

    recipeNames = [NSArray arrayWithObjects:@"Egg Benedict", @"Mushroom Risotto",
        @"Full Breakfast", @"Hamburger", @"Ham and Egg Sandwich", @"Creme Brelee",
        @"White Chocolate Donut", @"Starbucks Coffee", @"Vegetable Curry",
        @"Instant Noodle with Egg", @"Noodle with BBQ Pork", @"Japanese Noodle
        with Pork", @"Green Tea", @"Thai Shrimp Cake", @"Angry Birds Cake", @"Ham
        and Cheese Panini", nil];

    recipeImages = [NSArray arrayWithObjects:@"egg_benedict.jpg",
        @"mushroom_risotto.jpg", @"full_breakfast.jpg", @"hamburger.jpg",
        @"ham_and_egg_sandwich.jpg", @"creme_brelee.jpg",
        @"white_chocolate_donut.jpg", @"starbucks_coffee.jpg",
        @"vegetable_curry.jpg", @"instant_noodle_with_egg.jpg",
        @"noodle_with_bbq_pork.jpg", @"japanese_noodle_with_pork.jpg",
        @"green_tea.jpg", @"thai_shrimp_cake.jpg", @"angry_birds_cake.jpg",
        @"ham_and_cheese_panini.jpg", nil];

    recipePrepTimes = [NSArray arrayWithObjects:@"30 min", @"30 min", @"20 min",
        @"30 min", @"10 min", @"1 hour", @"45 min", @"5 min", @"30 min", @"8 min",
        @"20 min", @"20 min", @"5 min", @"1.5 hour", @"4 hours", @"10 min", nil];

    [self.tableView setContentInset:UIEdgeInsetsMake(20, self.tableView.
        contentInset.left, self.tableView.contentInset.bottom, self.tableView.
        contentInset.right)];
}
```

Information are added into the array

Table Row Selection

```
#pragma mark -
#pragma mark Table view data source
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)
    section
{
    return [recipeNames count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
    (NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"CustomTableCell";
    CustomTableCell *cell = (CustomTableCell *)[tableView
        dequeueReusableCellWithIdentifier:CellIdentifier];

    // Configure the cell...
    if (cell == nil) {
        cell = [[CustomTableCell alloc] initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
    }

    // Display recipe in the table cell
    cell.nameLabel.text = [recipeNames objectAtIndex:indexPath.row];
    cell.thumbnailImageView.image = [UIImage imageNamed:[recipeImages
        objectAtIndex:indexPath.row]];
    cell.prepTimeLabel.text = [recipePrepTimes objectAtIndex:indexPath.row];

    if (recipeChecked[indexPath.row]) {
        cell.accessoryType = UITableViewCellAccessoryCheckmark;
    } else {
        cell.accessoryType = UITableViewCellAccessoryNone;
    }

    return cell;
}
```

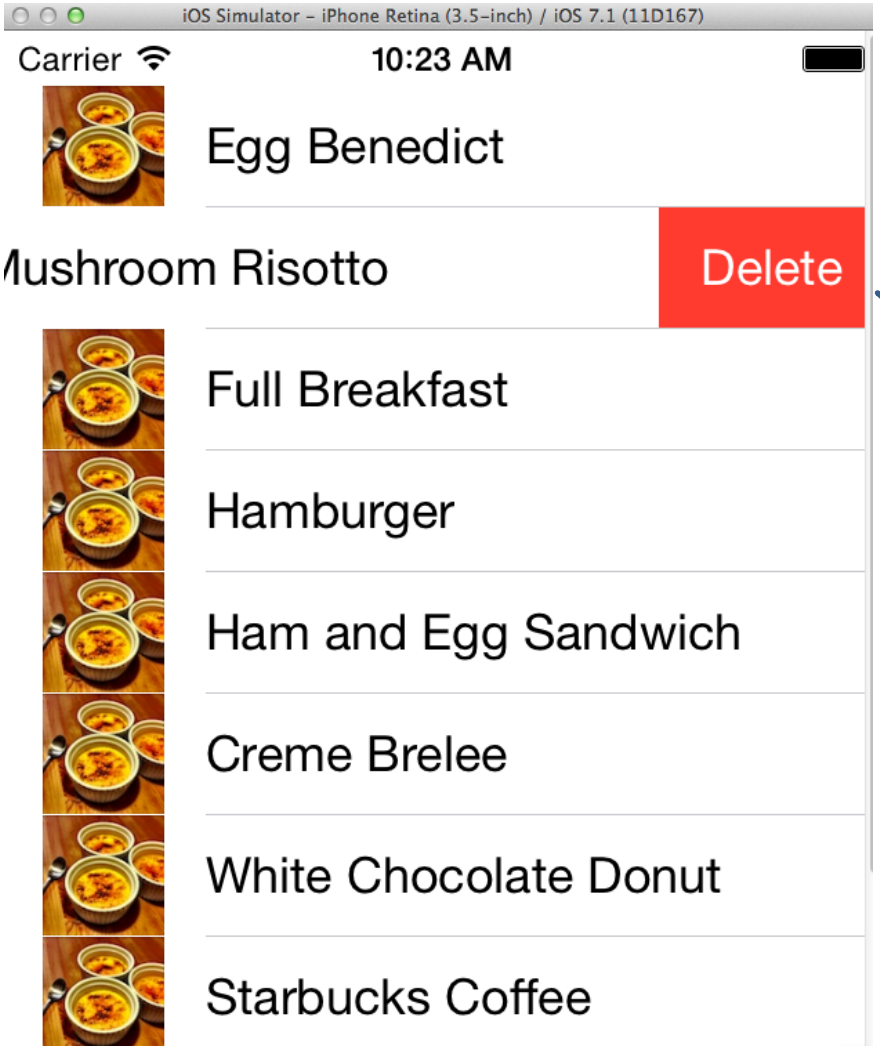
Different style will be applied when clicked

Table Row Selection

```
#pragma mark -
#pragma mark Table view delegate
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *
    )indexPath
{
    NSString* selectedRecipe = [recipeNames objectAtIndex:indexPath.row];
    UIAlertView *messageAlert = [[UIAlertView alloc]
                                 initWithTitle:@"Row Selected" message:
                                     selectedRecipe delegate:nil
                                     cancelButtonTitle:@"OK"
                                     otherButtonTitles:nil];

    // Display Alert Message
    [messageAlert show];

    UITableViewCell *cell = [tableView cellForRowAtIndexPath:indexPath];

    if (recipeChecked[indexPath.row]) {  // if the item is originally selected
        recipeChecked[indexPath.row] = NO;  // then remove the checkmark
        cell.accessoryType = UITableViewCellAccessoryNone;
    } else {                                // otherwise
        recipeChecked[indexPath.row] = YES; // add a checkmark
        cell.accessoryType = UITableViewCellAccessoryCheckmark;
    }

    [tableView deselectRowAtIndexPath:indexPath animated:YES];

}
```

Implement the didSelectRowAtIndexPath method

Table Row Deletion



TableRow can be made to delete

Table Row Deletion

```
#import "SimpleTableViewController.h"

@interface SimpleTableViewController ()

@end

@implementation SimpleTableViewController
{
    NSMutableArray *recipes;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Initialize table data
    recipes = [NSMutableArray arrayWithObjects:@"Egg Benedict", @"Mushroom Risotto", @"Full
        Breakfast", @"Hamburger", @"Ham and Egg Sandwich", @"Creme Brelee", @"White Chocolate
        Donut", @"Starbucks Coffee", @"Vegetable Curry", @"Instant Noodle with Egg", @"Noodle
        with BBQ Pork", @"Japanese Noodle with Pork", @"Green Tea", @"Thai Shrimp Cake", @"Angry
        Birds Cake", @"Ham and Cheese Panini", nil];
}
```

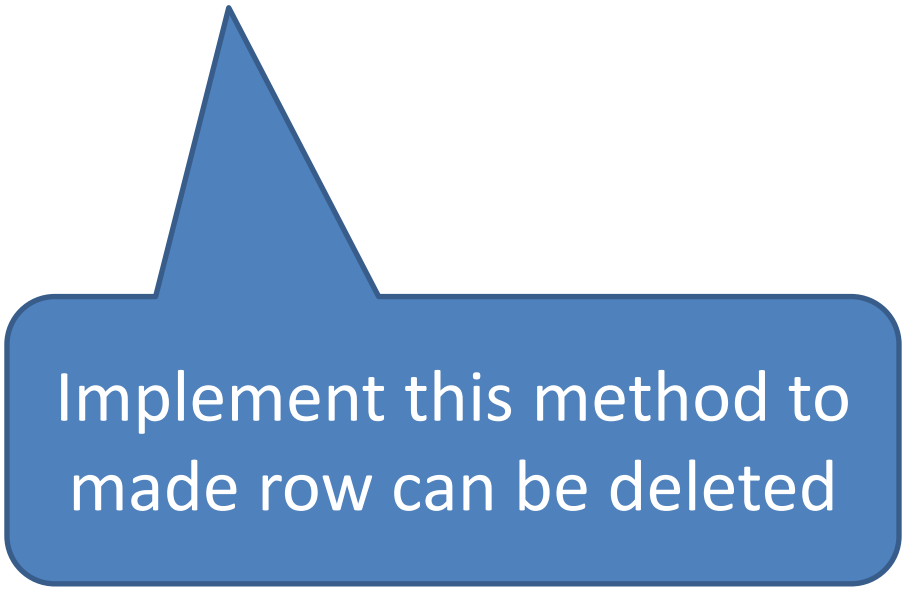
The array is mutable

Table Row Deletion

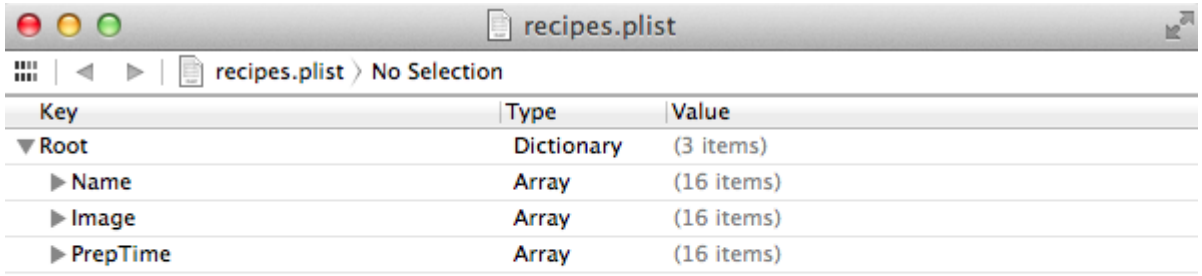
```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [recipes count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)
    indexPath
{
    static NSString *simpleTableIdentifier = @"SimpleTableCell";

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:simpleTableIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:simpleTableIdentifier];
    }

    cell.textLabel.text = [recipes objectAtIndex:indexPath.row];
    cell.imageView.image = [UIImage imageNamed:@"creme_brelee.jpg"];

    return cell;
}
```

Table Row Deletion

```
- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)
    editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath
{
    // Remove the row from data model
    [recipes removeObjectAtIndex:indexPath.row];

    // Request table view to reload
    [tableView reloadData];
}
```



Read Table Data from plist file



recipes.plist

recipes.plist 〉 No Selection

Key	Type	Value
Root	Dictionary	(3 items)
Name	Array	(16 items)
Item 0	String	Egg Benedict
Item 1	String	Mushroom Risotto
Item 2	String	Full Breakfast
Item 3	String	Hamburger
Item 4	String	Ham and Egg Sandwich
Item 5	String	Creme Brelee
Item 6	String	White Chocolate Donut
Item 7	String	Starbucks Coffee
Item 8	String	Vegetable Curry
Item 9	String	Instant Noodle with Egg
Item 10	String	Noodle with BBQ Pork
Item 11	String	Japanese Noodle with Pork
Item 12	String	Green Tea
Item 13	String	Thai Shrimp Cake
Item 14	String	Angry Birds Cake
Item 15	String	Ham and Cheese Panini
Image	Array	(16 items)
Item 0	String	egg_benedict.jpg
Item 1	String	mushroom_risotto.jpg
Item 2	String	full_breakfast.jpg
Item 3	String	hamburger.jpg
Item 4	String	ham_and_egg_sandwich.jpg
Item 5	String	creme_brelee.jpg
Item 6	String	white_chocolate_donut.jpg
Item 7	String	starbucks_coffee.jpg

Example plist file

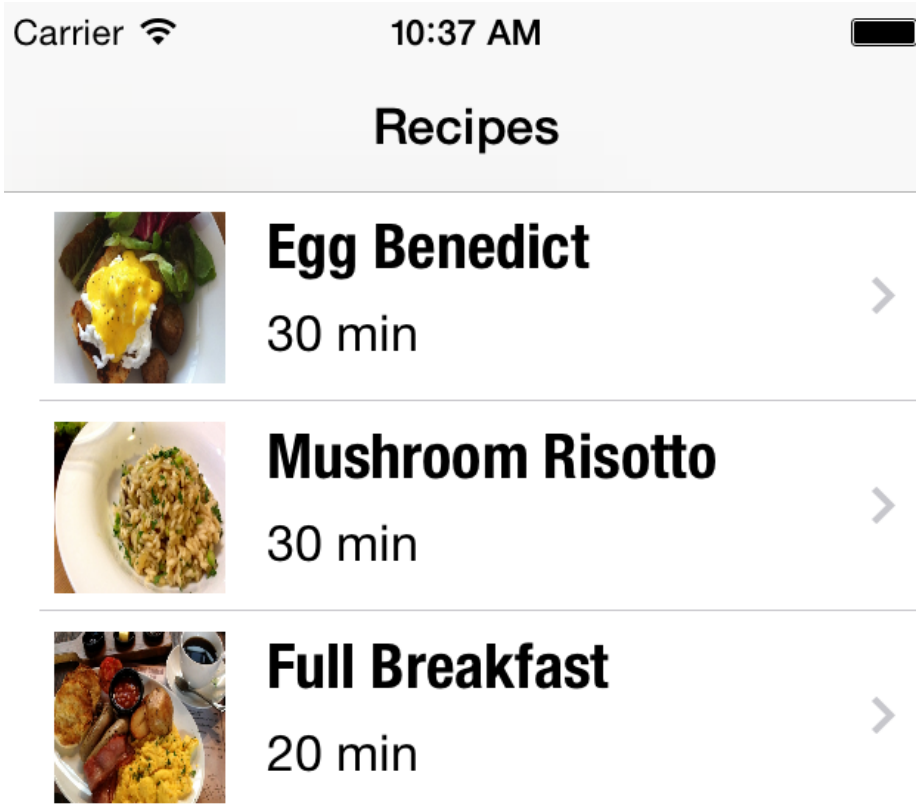
Read Table Data from plist file

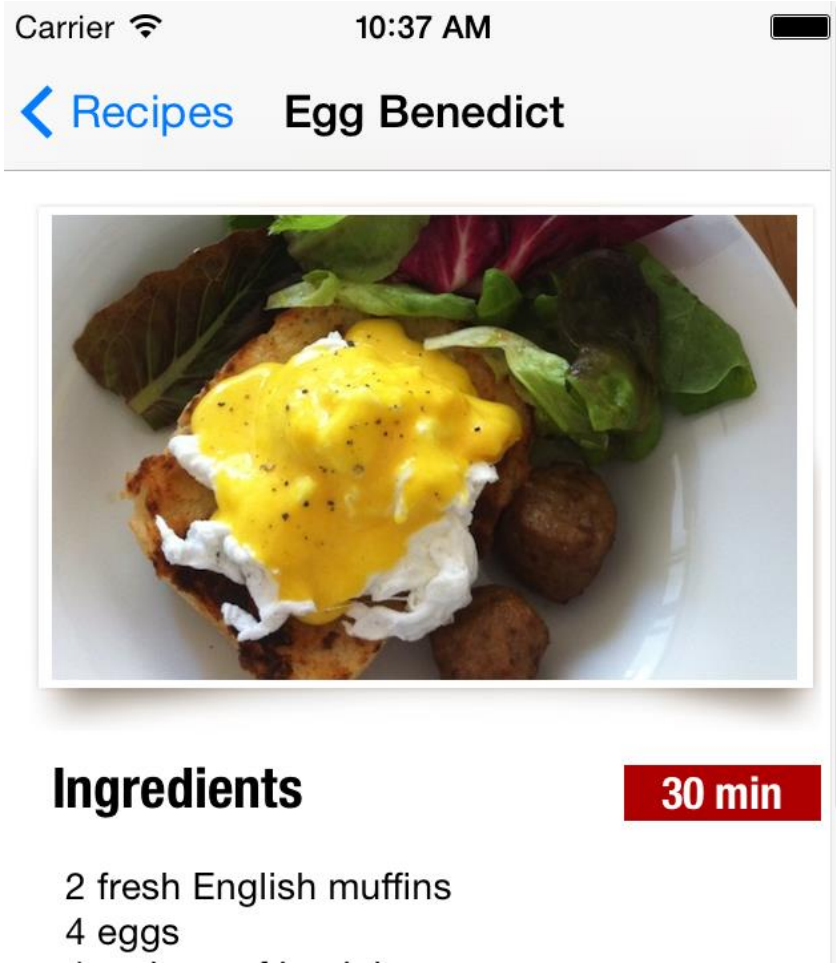
```
@interface CustomTableViewController ()

@end

@implementation CustomTableViewController
{
    NSArray *recipeNames;
    NSArray *recipeImages;
    NSArray *recipePrepTimes;
}


- (void)viewDidLoad
{
    [super viewDidLoad];

    // Find out the path of recipes.plist
    NSString *path = [[NSBundle mainBundle] pathForResource:@"recipes" ofType:@"plist"];

    // Load the file content and read the data into arrays
    NSDictionary *dict = [[NSDictionary alloc] initWithContentsOfFile:path];
    recipeNames = [dict objectForKey:@"Name"];
    recipeImages = [dict objectForKey:@"Image"];
    recipePrepTimes = [dict objectForKey:@"PrepTime"];

    [self.tableView setContentInset:UIEdgeInsetsMake(20, self.tableView.contentInset.left, self.
        tableView.contentInset.bottom, self.tableView.contentInset.right)];
}
```

Read Table Data from plist file

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

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)
    indexPath
{
    static NSString *CellIdentifier = @"CustomTableCell";
    CustomTableCell *cell = (CustomTableCell *)[tableView dequeueReusableCellWithIdentifier:
        CellIdentifier];

    // Configure the cell...
    if (cell == nil) {
        cell = [[CustomTableCell alloc] initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
    }

    // Display recipe in the table cell
    cell.nameLabel.text = [recipeNames objectAtIndex:indexPath.row];
    cell.thumbnailImageView.image = [UIImage imageNamed:[recipeImages objectAtIndex:indexPath.row
        ]];
    cell.prepTimeLabel.text = [recipePrepTimes objectAtIndex:indexPath.row];

    return cell;
}
```

Pass an object to Detail View





Pass an object to Detail View

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

@interface Recipe : NSObject

@property (nonatomic, strong) NSString *name; // name of recipe
@property (nonatomic, strong) NSString *prepTime; // preparation time
@property (nonatomic, strong) NSString *image; // image filename of recipe

@end
```

```
#import "Recipe.h"

@implementation Recipe

@end
```

Complex information can be saved in a custom made object and pass to the detail view

Pass an object to Detail View

```
#import "RecipeTableViewController.h"
#import "RecipeTableCell.h"
#import "RecipeDetailViewController.h"
#import "Recipe.h"

@interface RecipeTableViewController ()

@end

@implementation RecipeTableViewController
{
    NSArray *recipes;
}
```

Use in the header file

Pass an object to Detail View

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

    // Initialize the recipes array
    Recipe *recipe1 = [Recipe new];
    recipe1.name = @"Egg Benedict";
    recipe1.prepTime = @"30 min";
    recipe1.image = @"egg_benedict.jpg";

    Recipe *recipe2 = [Recipe new];
    recipe2.name = @"Mushroom Risotto";
    recipe2.prepTime = @"30 min";
    recipe2.image = @"mushroom_risotto.jpg";

    Recipe *recipe3 = [Recipe new];
    recipe3.name = @"Full Breakfast";
    recipe3.prepTime = @"20 min";
    recipe3.image = @"full_breakfast.jpg";
```

Creating a new object

Pass an object to Detail View

```
    Recipe *recipe15 = [Recipe new];
    recipe15.name = @"Angry Birds Cake";
    recipe15.prepTime = @"4 hours";
    recipe15.image = @"angry_birds_cake.jpg";

    Recipe *recipe16 = [Recipe new];
    recipe16.name = @"Ham and Cheese Panini";
    recipe16.prepTime = @"10 min";
    recipe16.image = @"ham_and_cheese_panini.jpg";

    recipes = [NSArray arrayWithObjects:recipe1, recipe2, recipe3, recipe4, recipe5, recipe6,
        recipe7, recipe8, recipe9, recipe10, recipe11, recipe12, recipe13, recipe14, recipe15,
        recipe16, nil];

}
```

Save the object into the array

Pass an object to Detail View

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)
    indexPath
{
    static NSString *CellIdentifier = @"CustomTableCell";
    RecipeTableCell *cell = (RecipeTableCell *)[tableView dequeueReusableCellWithIdentifier:
        CellIdentifier];

    // Configure the cell...
    if (cell == nil) {
        cell = [[RecipeTableCell alloc] initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:CellIdentifier];
    }


    Recipe *recipe = [recipes objectAtIndex:indexPath.row];
    cell.nameLabel.text = recipe.name;
    cell.thumbnailImageView.image = [UIImage imageNamed:recipe.image];
    cell.prepTimeLabel.text = recipe.prepTime;

    return cell;
}
```

Pass an object to Detail View

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    if ([segue.identifier isEqualToString:@"showRecipeDetail"]) {
        NSIndexPath *indexPath = [self.tableView indexPathForSelectedRow];
        RecipeDetailViewController *destViewController = segue.destinationViewController;

        Recipe *recipe = [recipes objectAtIndex:indexPath.row];
        destViewController.recipeName = recipe.name;
    }
}
```

Pass the object using prepareforSegue method

Pass an object to Detail View

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

@interface RecipeDetailViewController : UIViewController

@property (weak, nonatomic) IBOutlet UIImageView *recipeImageView;
@property (weak, nonatomic) IBOutlet UILabel *prepTimeLabel;
@property (weak, nonatomic) IBOutlet UITextView *ingredientsTextView;

@property (nonatomic, strong) Recipe *recipe;

@end
```

Use the object in the Controller header file

Pass an object to Detail View

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

    self.title = self.recipe.name;
    self.prepTimeLabel.text = self.recipe.prepTime;
    self.recipeImageView.image = [UIImage imageNamed:self.recipe.image];

    NSMutableString *ingredientsText = [NSMutableString string];
    for (NSString* ingredient in self.recipe.ingredients) {
        [ingredientsText appendFormat:@"%@\n", ingredient];
    }
    self.ingredientsTextView.text = ingredientsText;
}
```

Use the object in the Controller implementation file