

```

//
// ViewController.m
// Catalogue
//
// Created by kcfancher on 8/3/12.
// Copyright (c) 2012 kcfancher. All rights reserved.
//

#import "ViewController.h"
#import "Book.h"
#import "CD.h"
#import "ItemCell.h"
#import "DetailViewController.h"

@interface ViewController ()

@end

@implementation ViewController
@synthesize itemsTableView;
@synthesize items = _items;

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Instantiate items using NSMutableArray's class method array which is short for
    // [[NSMutableArray alloc] init]
    self.items = [NSMutableArray array];

    // We are adding some Book and CD instances to our items array
    [self.items addObject:[Book bookWithTitle:@"Objective-C" author:@"Keli" price:1.99
                                identificationNumber:100]];
    [self.items addObject:[Book bookWithTitle:@"Objective-C 2.0" author:@"Keli" price:1.99
                                identificationNumber:100]];

    [self.items addObject:[Book bookWithTitle:@"Cocoa Is Fun"
                                author:@"Keli C. Fancher"
                                price:2.99
                                identificationNumber:100]];

    [self.items addObject:[CD cdWithArtist:@"Keli C. Fancher"
                                label:@"Rhapsody in C, Objective-C"
                                price:0.00
                                identificationNumber:101]];

    [self.items addObject:[Book bookWithTitle:@"UITableView's are not so hard"
                                author:@"Steven Sikes"
                                price:2.99
                                identificationNumber:102]];

    [self.items addObject:[CD cdWithArtist:@"Aragorn"
                                label:@"I am King"
                                price:1.99
                                identificationNumber:103]];

    [self.items addObject:[Book bookWithTitle:@"iOS, Now What"
                                author:@"Yoda"
                                price:3.33
                                identificationNumber:104]];

    [self.items addObject:[CD cdWithArtist:@"Gandalf"
                                label:@"Magic, what magic?"
                                price:7.77
                                identificationNumber:105]];

    [self.items addObject:[Book bookWithTitle:@"Sandboxing"
                                author:@"Sam, yeah that Sam"
                                price:8.99
                                identificationNumber:106]];

    [self.items addObject:[CD cdWithArtist:@"Iluvatar"

```

```

        label:@"The Music is from me"
        price:99.99
        identificationNumber:107]];

[self.items addObject:[Book bookWithTitle:@"Objective-C 2.0"
        author:@"Stephen Kochan"
        price:3.99
        identificationNumber:108]];

[self.items addObject:[CD cdWithArtist:@"Tom Bombadil"
        label:@"Hey dol! merry dol!"
        price:9.99
        identificationNumber:109]];
}

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

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

// Here are the two required UITableViewDataSource Methods

// Every TableView can have a number of different sections
// An example is the letter heading in the Contacts application
// This method returns the number of rows that are in each section
// For an app with no sections this method just returns the total number of rows
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    // We want there to be one row for each item in our items array
    return self.items.count;
}

// tableView:cellForRowAtIndexPath returns a UITableViewCell that has been customized
// for each tableViewCell in our TableView (This method is called once per row/cell in our
// tableView)
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    // We first have to create a UITableViewCell instance
    // We use UITableView's instance method -(UITableViewCell *)dequeueReusableCellWithIdentifier
    // to
    // Return a TableViewCell that is memory efficient
    // We pass it the same string that we have used as an identifier in our Interface
    ItemCell *cell = [tableView dequeueReusableCellWithIdentifier:@"ItemCell"];

    // First we have to get the Item for the current row
    // IndexPath is an object that has both a section and a row property
    // Our catalogue stores both Book's and CD's - but since both are subclasses
    // of Item we can declare it universally as a Item
    Item *instance = [self.items objectAtIndex:indexPath.row];

    // Next we need to test if the Item is a Book or a CD to set the labels appropriately

    // If the Item is a Book we will use the Book's properties for the label
    if ([instance isKindOfClass:[Book class]]) {
        Book *book = (Book *) instance;
        cell.mainLabel.text = book.title;
        cell.smallLabel.text = book.author;
        cell.rightLabel.text = [NSString stringWithFormat:@"$%.2f", book.price];
        // Set the image property on the cells itemImage using book.png (book@2x.png will
        // automatically be used on retina displays)
        cell.itemImage.image = [UIImage imageNamed:@"book.png"];
    }
    // If the Item is a CD we will use the CD properties for the label

```

```

else if ([instance isKindOfClass:[CD class]]) {
    CD *cd = (CD *) instance;
    cell.mainLabel.text = cd.label;
    cell.smallLabel.text = cd.artist;
    cell.rightLabel.text = [NSString stringWithFormat:@"%$.2f",cd.price];
    // Set the image property on the cells itemImage using cd.png (cd@2x.png will automatically
    // be used on retina displays)
    cell.itemImage.image = [UIImage imageNamed:@"cd.png"];
}

// Return the customized cell
return cell;
}

// Prepare for segue is called everytime a segue occurs within our storyboard
// We want to check the identifiers of each segue to see which actual segue occurred
// We will then use this method to perform final changes that we need to make
// before we transition to
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    // Testing the segue's identifier to see if it is DetailViewSegue
    // If it is we are going to set the item property on the
    // new DetailViewController that will be loaded onto screen
    if ([segue.identifier isEqualToString:@"DetailViewSegue"]) {
        DetailViewController *detailVC = segue.destinationViewController;
        NSIndexPath *indexPath = [self.itemsTableView indexPathForSelectedRow];
        detailVC.item = [self.items objectAtIndex:indexPath.row];
        [self.itemsTableView deselectRowAtIndexPath:indexPath animated:YES];
    }
    // Testing the segue's identifier to see if it is AddViewSegue
    // If it is we are going to set the AddViewController's delegate
    // To be our self (we will respond to the delegate calls)
    else if ([segue.identifier isEqualToString:@"AddViewSegue"]) {
        AddViewController *addVC = (AddViewController *)[(UINavigationController *)segue.
            destinationViewController topViewController];
        addVC.delegate = self;
    }
}

// The first of AddViewControllerDelegate's methods
// this method will be called whenever the AddViewController's
// Cancel button has been pressed
// We are going to do nothing in this case
- (void)addViewControllerDidCancel:(AddViewController *)addVC
{
}

// The first of AddViewControllerDelegate's methods
// this method will be called whenever the AddViewController's
// Done button has been pressed
// In this case we are going to add the newItem to our
// items array, and present an alert to ask the user to Tweet the change
- (void)addViewController:(AddViewController *)addVC didSaveItem:(Item *)newItem
{
    // We add the newItem to our items array
    [self.items addObject:newItem];
    // We then have to call reloadData on tableview to tell it to refresh data
    [self.itemsTableView reloadData];

    // Create a UIAlertView to ask the user if they want to tweet about the new item
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"You added a new item"
        message:@"Would you like to tweet about your
        new item?"
        delegate:self
        cancelButtonTitle:@"No"
        otherButtonTitles:@"Yes", nil];

    [alert show];
}

// The UIAlertViewDelegate method that will be called when Yes or No button is pressed

```

```
- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex
{
    // if the button is No
    if (buttonIndex == 0) {
        // We do not have to do anything
    }
    // if the button is Yes
    else if (buttonIndex == 1) {
        // Create a new Tweet Compose View Controller instance
        TWTweetComposeViewController *tweetSheet = [[TWTweetComposeViewController alloc] init];
        // Set the initial text
        [tweetSheet setInitialText:@"Hey, check out our new items..."];
        // Display the new tweetSheet as a modal popup
        [self presentModalViewController:tweetSheet animated:YES];
    }
}

@end
```