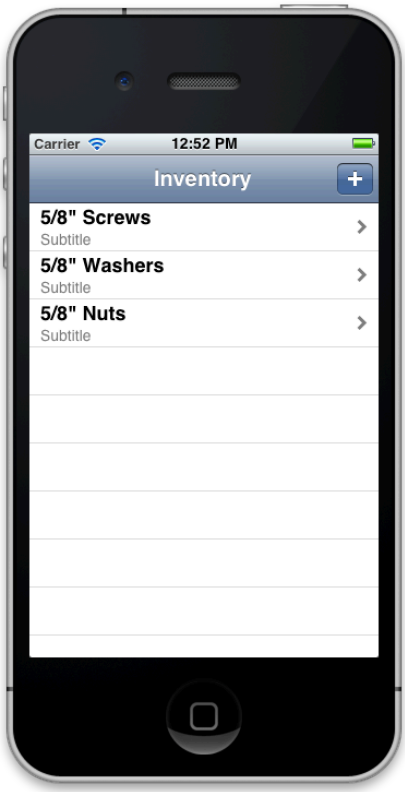
Assignment 3



Design an app to allow inventory tracking for a manufacturing company. The company buys parts in bulk and wants an app to help track how many items are in it's current inventory.

Use the list view from a UITableViewController and the UINavigationController to show a list of inventory parts. Each row will open a detail view which will display more information about the item. A user should be able to quickly add new parts using a numeric textfield and an add button.

Required Tasks

Create the following classes and view hierarchy.

1. UINavigationController

A UINavigationController will control the transitions between list view and the detail view screens. Create an instance and add your InventoryViewController as the root view controller.

2. InventoryViewController : UITableViewController

A view controller to show a list of items as UITableViewCell's in the inventory.

UITableViewCell

a. Indicate a tap action is available for the row by using the accessory type: `UITableViewCellAccessoryDisclosureIndicator`
b. Display the item name
c. Display the quantity

Actions

When a user taps on a row, the ItemViewController should be pushed onto the navigation controller.

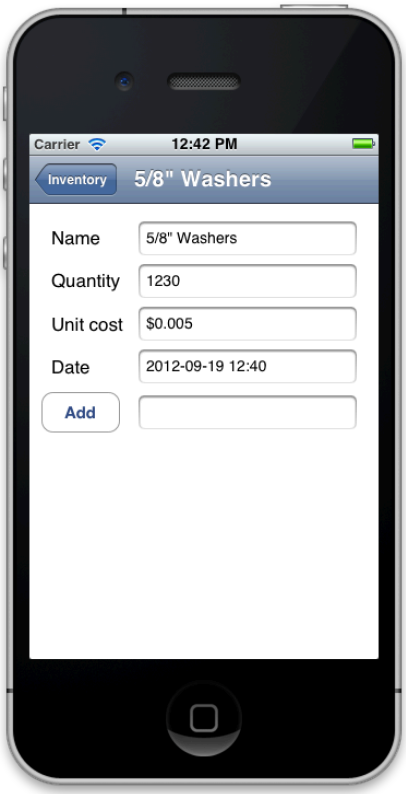
Add Button

Add a button to the navigation bar to add random inventory items. The item names can be fake and selected from an array that you provide.

Title

Set the title of the controller's navigationItem to @"Inventory"

3. ItemViewController : UIView



A view controller to display the details on an item.

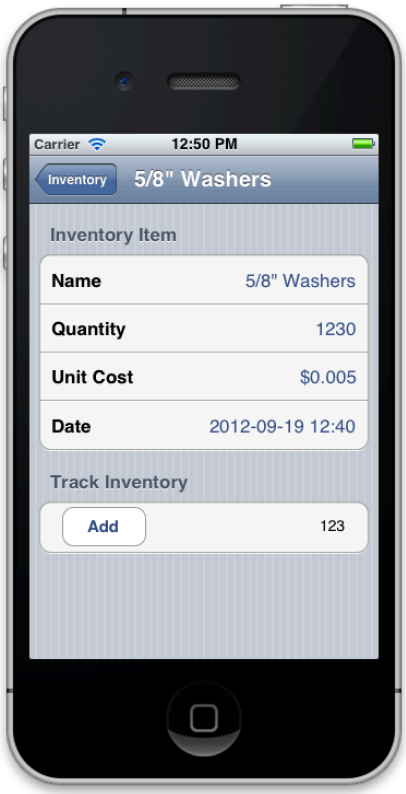
1. Display the *item name*
2. Display the *current quantity*
3. Display the *unit cost*
4. Display the last *inventory date*
5. *Add* button
6. *Add quantity* textfield

Actions

1. The user can edit the item name. The title for the controller's navigationItem should change.
2. The current quantity textfield is not enabled for user input.
3. The unit cost textfield is enabled for user input. Changes here should be reflected in the UITableView.
4. The date is not enabled for user input.
5. When the user pressed "Add" the following will occur:
   a. The items will be added to the current quantity total and the sum will be displayed.
   b. The Adding new quantity will update the quantity and inventory date.
   c. The Add UITextField will clear

4. Item : NSObject

- NSString *itemName;
- NSInteger currentQuantity;
- float unitCost;
- NSDate *lastInventoryDate;



5. Test Data

Create a method to create the following items and add it to the inventory. Set a random quantity and current time for each item on start.

```
– (void)addRandomTestData;
```

Sample Data

- 5/8" screws @ $0.03/screw
- 5/8" nuts @ $0.01/nut
- 5/8" washers @ $0.005/washer
- PCB boards @ $10/board

6. Add Inventory Item Button

Create an add button "+" to create new random inventory items.

a. Use the UIBarButtonItem and add it to your

InventoryViewController's navigation controller.
b. -(void)addRandomItem:(id)sender;

Hints

- Read *Table View Programming Guide for iOS*
- `[self.tableView reloadData]` will refresh the table view
- Use UIBarButtonItem `initWithBarButtonSystemItem:target:action:`
- You can pass a method selector using `@selector(addButtonPressed:)`
- In a UIViewController subclass use `self.navigationItem` to access and set left and right bar buttons on your UINavigationController.

Evaluation (100 points)

Assignments will lose points for the following:

- Build errors
- Build warnings.
- NSLog() print statements.
- Lack of Javadoc-style method comments.
  - Assume the reader is familiar with Objective-C. Explain what methods do in your header files and any complex logic in your implementation files.
- Incomplete or missing required specifications.
- Code style does not match "Coding Guidelines for Cocoa."

Grading

- Style/Documentation 30
- Organization 20
- Functionality 50

Extra Credit

1. Use a static cells to display the UITableViewController called ItemTableViewController (3 points)
   1. Create a MainStoryboard.storyboard file to setup static cells.
   2. Do not implement any UITableViewDataSource methods in ItemTableViewController. The interface you design is the data source.
   3. Set a view controller identifier and load it programmatically from storyboard.
2. Use an indexed list to allow quick navigation to data at sections. (5 points)
   1. Read *Table View Programming Guide for iOS* "Populating an Indexed List"
   2. Add *NSInteger sectionIndex* to your Item object.
3. Create a header view for the table to display the total number of inventory parts and the total value of all the parts. The view should be at the top of your UITableView. When adding new parts, the header view should reflect the changes. (2 points)
   1. Note: Changing the quantity or the cost of an item should be reflected in this header view.