Name: Christine Ebeo

Student Number: N01455114

Course: ITE-5334 IOS App Dev

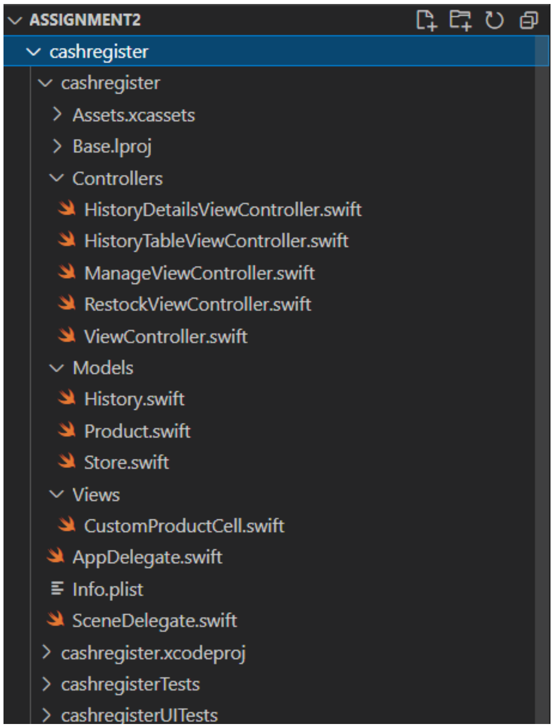
Assignment 2

Video URL (Youtube): https://youtu.be/C4hKfagOtcU

Video URL (Google drive): https://drive.google.com/file/d/1ztlI7AJm6OvT3ZxDvPb6pwrCVeF0dc3Z/view?usp=sharing

Github URL: https://github.com/tinebeo/ite5334-ios-app-dev/tree/main/Assignment2/cashregister

Folder structure:



```
ASSIGNMENT2
  cashregister
    cashregister
      Assets.xcassets
      Base.lproj
      Controllers
        HistoryDetailsViewController.swift
        HistoryTableViewController.swift
        ManageViewController.swift
        RestockViewController.swift
        ViewController.swift
      Models
        History.swift
        Product.swift
        Store.swift
      Views
        CustomProductCell.swift
      AppDelegate.swift
      Info.plist
      SceneDelegate.swift
    cashregister.xcodeproj
    cashregisterTests
    cashregisterUITests
```

Controllers:

HistoryDetailsViewController

```
import UIKit

class HistoryDetailsViewController: UIViewController {
    var history : History?

    @IBOutlet weak var histoyLabel: UILabel!
    @IBOutlet weak var historyQuantity: UILabel!
    @IBOutlet weak var historypurchaseDate: UILabel!
    @IBOutlet weak var historyTotal: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.

        // update title
        self.title = history!.name

        // update historyDetails
        histoyLabel.text! = "\(history!.name)\n"
        historyQuantity.text! = "\(String(history!.quantity))\n"
        historypurchaseDate.text! = "\(history!.purchaseDate)\n"
        historyTotal.text! = "Total amount: \(String(history!.totalPrice))"

    }

}
```

HistoryTableViewController

```
import UIKit

class HistoryTableViewController: UITableViewController {

    // main View controller assigns this value
    var store : Store?

    override func viewDidLoad() {
        super.viewDidLoad()

    }

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSection
section: Int) -> Int {
        return store!.getHistoryCount()
    }


    override func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {

        let cell = tableView.dequeueReusableCell(withIdentifier: "historyCell",
for: indexPath) as! CustomProductCell

        cell.productLbl.text = store?.getHistory(idx: indexPath.row).name
        cell.productQty.text = String(store!.getHistory(idx:
indexPath.row).quantity)

        return cell
    }

    override func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {

        // item selected, prepare segue
        performSegue(withIdentifier: "historyDetails", sender: indexPath)

    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?)
    {
        // there's only one destination, directly set to that controller
        let s = segue.destination as? HistoryDetailsViewController
        s?.history = (store?.getHistory(idx: (sender as! IndexPath).row))!

    }

}
```

ManageViewController

```
import UIKit

class ManageViewController: UIViewController {

    // main View controller assigns this value
    var store : Store?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }



    override func prepare(for segue: UIStoryboardSegue, sender: Any?)
    {

        if (segue.identifier == "history") {
            let s = (segue.destination as? HistoryTableViewController)!
            s.store = store;

        } else  {
            let s = (segue.destination as? RestockViewController)!
            s.store = store;
        }

    }

}
```

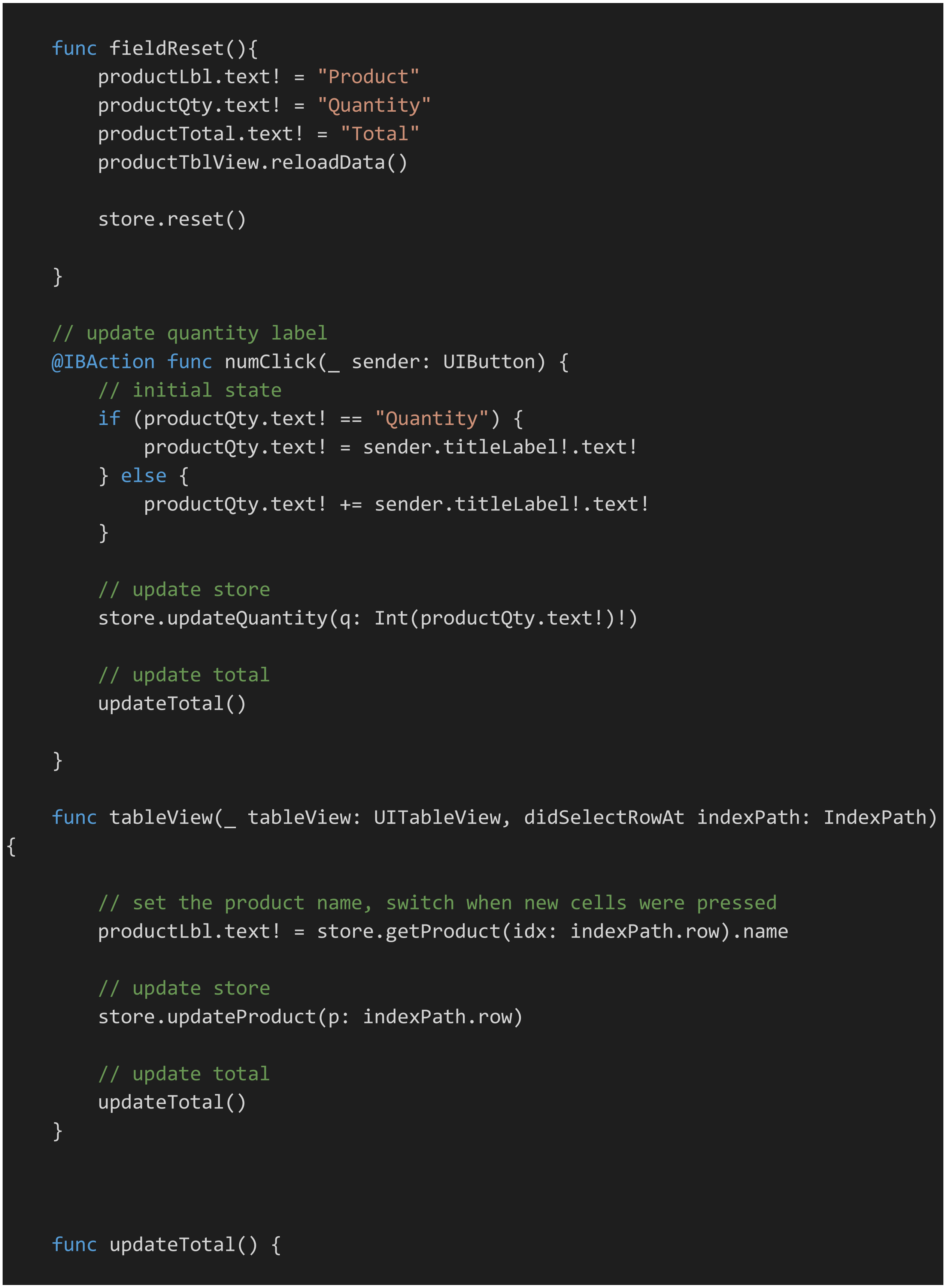
RestockViewController

```
import UIKit

class RestockViewController: UIViewController,
                             UITableViewDelegate,
                             UITableViewDataSource  {

    var store : Store?
    var productIdx : Int?
    @IBOutlet weak var restockTblView: UITableView!
    @IBOutlet weak var restockAmount: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        restockTblView.delegate = self
        restockTblView.dataSource = self
    }

    func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }
    // number of rows in table view
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int)
-> Int {
        return store!.getProductCount()
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -
> UITableViewCell {

        let cell = tableView.dequeueReusableCell(withIdentifier: "restockCell",
for: indexPath) as! CustomProductCell

        cell.productLbl.text = store!.getProduct(idx: indexPath.row).name
        cell.productQty.text = String(store!.getProduct(idx:
indexPath.row).quantity)

        return cell
    }

    // functionality logic

    @IBAction func cancelClick(_ sender: UIButton) {
        if let nav = self.navigationController {
            nav.popViewController(animated: true)

        } else {
            self.dismiss(animated: true, completion: nil)

        }
    }
    @IBAction func restockClick(_ sender: Any) {

        if (Int(restockAmount.text!) == nil
            || productIdx == nil) {
            let alert = UIAlertController(title: "Error!", message: "An invalid
restock was made. Verify product and quantity.", preferredStyle: .alert)

            let action = UIAlertAction(title: "OK", style: .cancel,handler: nil)

            alert.addAction(action)
            present(alert, animated: true, completion: nil)
        } else {
            store!.getProduct(idx: productIdx!).quantity =
Int(restockAmount.text!)!
```

```
            restockTblView.reloadData()

            // reset selections
            restockAmount.text! = "Enter new quantity"
            productIdx = nil
        }

    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
{

        // product to updated selected
        productIdx = indexPath.row


    }

}
```

ViewController

```
import UIKit

class ViewController: UIViewController ,
                      UITableViewDelegate,
                      UITableViewDataSource {

    @IBOutlet weak var productLbl: UILabel!
    @IBOutlet weak var productQty: UILabel!
    @IBOutlet weak var productTotal: UILabel!
    @IBOutlet weak var productTblView: UITableView!

    // store holds the products
    var store : Store = Store()

    func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    // number of rows in table view
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int)
-> Int {
        return store.getProductCount()
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -
> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "productCell",
for: indexPath) as! CustomProductCell

        cell.productLbl.text = store.getProduct(idx: indexPath.row).name
        cell.productQty.text = String(store.getProduct(idx:
indexPath.row).quantity)
        cell.productPrice.text = String(store.getProduct(idx:
indexPath.row).price)

        return cell
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?)
    {
        // there's only one destination, directly set to that controller
        let s = segue.destination as? ManageViewController
        s?.store = store;

    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        productTblView.delegate = self
        productTblView.dataSource = self


    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        productTblView.reloadData()
    }

    // functionality logic below

    // extra functionality to clear inputs
    @IBAction func resetClick(_ sender: Any) {

        fieldReset()
    }
```



```
    func fieldReset(){
        productLbl.text! = "Product"
        productQty.text! = "Quantity"
        productTotal.text! = "Total"
        productTblView.reloadData()

        store.reset()

    }

    // update quantity label
    @IBAction func numClick(_ sender: UIButton) {
        // initial state
        if (productQty.text! == "Quantity") {
            productQty.text! = sender.titleLabel!.text!
        } else {
            productQty.text! += sender.titleLabel!.text!
        }

        // update store
        store.updateQuantity(q: Int(productQty.text!)!)

        // update total
        updateTotal()

    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
{

        // set the product name, switch when new cells were pressed
        productLbl.text! = store.getProduct(idx: indexPath.row).name

        // update store
        store.updateProduct(p: indexPath.row)

        // update total
        updateTotal()
    }



    func updateTotal() {
        let total = store.getTotal()

        // if total == 0, either product/quantity is missing
        // or user typed a 0 quantity
        if (total > 0) {
            productTotal.text! = String(total)
        }

    }

    @IBAction func buyClick(_ sender: UIButton) {

        // if sucessful purchase
        if (store.buy()) {
            // update the tableview
            productTblView.reloadData()
            fieldReset()

        } else {
            let alert = UIAlertController(title: "Error!", message: "An invalid purchase was made. Verify product and quantity.", preferredStyle: .alert)

            let action = UIAlertAction(title: "OK", style: .cancel,handler: nil)

            alert.addAction(action)
            present(alert, animated: true, completion: nil)

        }

    }
}
```

Models:

History

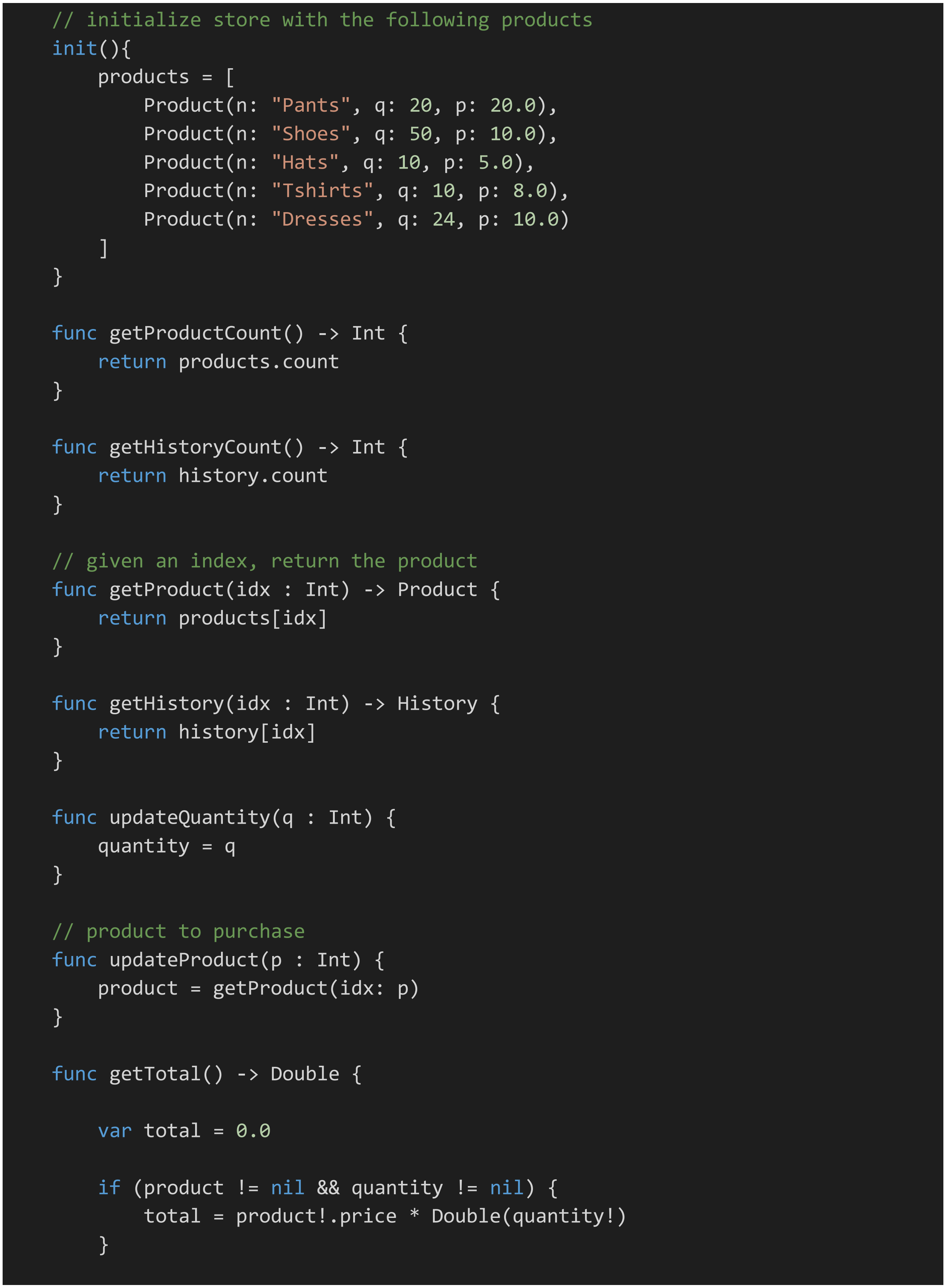
```
import Foundation

class History {

    var name : String;
    var quantity : Int;
    var totalPrice : Double;
    var purchaseDate : String;

    init(n: String, q: Int, t: Double) {
        name = n
        quantity = q
        totalPrice = t

        let dateFormatter = DateFormatter()
        dateFormatter.dateFormat = "MM/dd/yyyy HH:mm:ss a"
        purchaseDate = dateFormatter.string(from: Date())

        //purchaseDate = Date.now
    }

}
```

Product

```
import Foundation

class Product {
    var name : String
    var quantity : Int
    var price : Double

    init(n: String, q: Int, p: Double) {
        name = n
        quantity = q
        price = p
    }
}
```

Store

```
import Foundation

class Store {

    var products : [Product] = []
    var history : [History] = []

    // current selected Product and quantity
    var product : Product?
    var quantity : Int?
```



```swift
    // initialize store with the following products
    init(){
        products = [
            Product(n: "Pants", q: 20, p: 20.0),
            Product(n: "Shoes", q: 50, p: 10.0),
            Product(n: "Hats", q: 10, p: 5.0),
            Product(n: "Tshirts", q: 10, p: 8.0),
            Product(n: "Dresses", q: 24, p: 10.0)
        ]
    }

    func getProductCount() -> Int {
        return products.count
    }

    func getHistoryCount() -> Int {
        return history.count
    }

    // given an index, return the product
    func getProduct(idx : Int) -> Product {
        return products[idx]
    }

    func getHistory(idx : Int) -> History {
        return history[idx]
    }

    func updateQuantity(q : Int) {
        quantity = q
    }

    // product to purchase
    func updateProduct(p : Int) {
        product = getProduct(idx: p)
    }

    func getTotal() -> Double {

        var total = 0.0

        if (product != nil && quantity != nil) {
            total = product!.price * Double(quantity!)
        }
        return total
    }

    func buy() -> Bool {

        var purchaseStatus = false;

        if (product != nil && quantity != nil
            && product!.quantity >= quantity!) {

            purchaseStatus = true

            // add purchase to history
            history.append(History(
                n: product!.name,
                q: quantity!,
                t: product!.price * Double(quantity!)
            ))

            // update product quantity
            product!.quantity -= quantity!

        }

        return purchaseStatus

    }

    func reset() {
        product = nil
        quantity = nil
    }
}
```

Views:

CustomProductCell

```
import UIKit

class CustomProductCell: UITableViewCell {

    @IBOutlet weak var productLbl: UILabel!
    @IBOutlet weak var productQty: UILabel!
    @IBOutlet weak var productPrice: UILabel!
    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }

}
```