Ryerson University

Department of Electrical & Computer Engineering
Faculty of Engineering & Architectural Science

Course Title:	Object Oriented Engineering Analysis and Design
Course Number:	COE 528
Semester/Year (e.g.F2016)	F2017

Instructor:	Olivia Das

Assignment/Lab Number:	Project
Assignment/Lab Title:	Analysis/Design/Implementation/Testing of a POS System

Submission Date:	November 26, 2017
Due Date:	November 26, 2017

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Gusthinna Wadu	Pasindu			
Nadarajah	Charles			
Silva	Hithanadura			

Reset Form

Problem Description:

The purpose of this project is to create a point of sale menu system for a customer to purchase items and a manger to overview the system. The items that will be for sale in the system includes fruits and vegetables. Each item has a certain quantity that is variable in the system. Initially the items will be read from a file that will include an initial quantity, but this quantity will change depending upon purchases and manager modification. The manager can modify the store inventory and add or remove coupons. The coupons will be used to apply discount to store items. When a customer is at the checkout level, he/she will have an option to enter their coupon number in which their cart total adjusted if a coupon is available. Once this is done, he/she cannot use the same coupon again (cart total won't change) but they may check out their cart with payment by credit or cash. If the latter option is selected, change will be given (if cash amount exceeds cart total.) At the end of each transaction, the transaction will be written to an array list. The array list will be used for the manager to view transactions. At the end of the program cycle, the list of transactions recorded will be written to a file.

Functional Requirements:

Use Case name	Login
Participating Actors	Customer, Manager
Flow of events	1. In the initial screen of the program, the option to login as Manager, Customer, or New Customer is offered. 2. If ‘Manager Login’ option is selected, User should sign in with Manager’s credentials. This will allow access to Manager functions such as modify inventory, view transactions etc. 3. If ‘Customer Login’ option is selected, User will be prompted to enter their unique credentials. Then the Customer will have access to view and buy store items. 4. If ‘New Customer Sign Up’ option is selected, the User will be prompted to enter a username and a password. After validating is no other Customer with same credentials, the New Customer is added to the existing Customer list and is presented with the POS.
Entry Condition	- All other users must be logged out
Exit Condition	- User is logged in - New Customer added to system (if ‘New Customer Sign Up’ option is selected)
Quality Requirements	A Manager User must always be created at the start of the system

Use Case name	Modify Inventory
Participating Actors	Manager
Flow of events	1) After logging in as Manager, the manager can modify the menu items in the inventory by changing two of its attributes; a) Add Item: Adding an item increases its quantity b) Remove Item: Removing an item decreases its quantity by the appropriate amount. Minimum quantity is 0 2) Any of the above two changes update the inventory(stored as an arraylist of type MenuItem) in the main POS system
Entry Condition	- Manager must be logged into system
Exit Condition	- Inventory for a specific menu item is increased/decreased
Quality Requirements	All menu items must exist previously. New menu items cannot be created

Use Case name	Payment
Participating Actors	Customer
Flow of events	1. Customer proceeds from the checkout stage to the payment stage. 2. If a coupon is presented at the checkout stage, applicable discounts are added. 3. If the customer chooses the cash option, the POS system calculates the correct amount of change by subtracting the discounted price from the presented cash. 4. If the customer chooses the credit card option, he/she is directed to the credit payment screen by the POS system. Here, the Credit Card number should be entered for the payment to go through. 5. After one of events 3 or 4 is completed, the transaction is considered completed and recorded in the POS system. This event updates the available menu items for next customer
Entry Condition	- Customer must be logged into system - Customer must have some menu items in basket
Exit Condition	- The payment is approved and the transaction is recorded - Remaining menu item quantities updated
Quality Requirements	Customer should choose either cash or credit, no other payment methods

Design:

As mentioned in the problem description, the rationale of this project was to simulate a POS (Point of Sale) program found in a typical food store/restaurant. It is implemented by 6 main modules. Namely:

1. Customer: A user of the program who buys certain items in the menu
2. Manager: User of the program who maintains the menu quantity, pricing, and transactions
3. MenuItem: Represents a tangible food with a name, price, and a quantity
4. Transaction: Called upon to handle the checkout process and payment of an order once the customer wishes to checkout
5. CashPayment: Simulates payment with cash, completed through a handing total
6. CreditPayment: Simulates payment with a credit card, completed through a credit card number

To implement the idea, elements from Design Patterns Bridge and State were incorporated to the design. The initial pattern is evident in the use of abstract class ‘Payment’ to implement the two payment types Cash and Credit Card. Use of method applyDiscount causes decoupling between the two types of payment, as these two classes do not require the comprehension of opposite class’s composition when applying the provided coupon and calculating the effective total price. Principles of State Pattern were used mainly in the design of ‘Customer’ and ‘Manager’ classes. During each sign-on in the program cycle, it changes its internal state into customer or a manger depending on the selection/credentials. The behaviour is changed accordingly to address state specific requests. As shown in both the Class and Use Case diagrams, this pattern’s localization property provides an easy and an efficient implementation of specific program functions depending on the type of user.

Testing:

Test Case ID:	Test Description	Whitebox/Blackbox
TC01-CreditCardConstructor	Testing the transaction constructor using credit card as payment method.	Whitebox
TC02-CashConstrucutor	Testing the transaction constructor using cash as payment method.	Whitebox
TC03-TransactionID	Testing the transaction class static variable.	Whitebox
TC04-GetCustomer	Testing the transaction class to get customer object.	Whitebox
TC05-GetMenuItemsBought	Testing the transaction class to get array list of menu items bought.	Whitebox
TC06-GetTotalNetPrice	Testing the transaction class get purchase total method. A double value will be returned.	Whitebox
TC07-GetPayment	Testing the transaction class to get the payment object.	Whitebox
TC08-toString	Testing the toString method to see if the output is correct.	Whitebox
TC09-repOk	Testing the Boolean repOk method to see if the test transaction object passes the repOk.	Whitebox
TC10-repOkCustomer	Testing the repOk method by setting the customer to null and observing if the repOk will fail.	Blackbox
TC11-repOkMenuItemsNull	Testing the repOk method by setting the menu item passed in the constructor to null. This test should fail.	Blackbox
TC12-repOkMenuEmpty	Testing the repOk method by setting the menu array list to be empty. This test should fail.	Blackbox
TC13-repOkNetPrice	Testing the repOk method by having the total net price equal to zero. This test should fail.	Blackbox

References:

Das, O. (2017). Modelling with UML [PDF file]. Retrieved from https://courses.ryerson.ca/d2l/le/content/134986/viewContent/1681158/View

Das, O. (2017). Design Patterns Part 2 [PDF file]. Retrieved from https://courses.ryerson.ca/d2l/le/content/134986/viewContent/1697454/View

Das, O. (2017). Design Patterns Part 3 [PDF file]. Retrieved from https://courses.ryerson.ca/d2l/le/content/134986/viewContent/1704266/View