TYLER FORRESTER

CS162

ASSIGNMENT 1

DESIGN DOCUMENT

Contents

1 OVERVIEW

This design document will cover the basic use cases, classes and variables used in creating a list of variable length. Assignment 1 task to create a program which takes an items name, quantity, type of quantity and price and adds this to running list of items. The user can add items, display the entire list and delete items.

2 USE CASE CREATING A LIST

UC-1: Creating a List

Main Success Scenario

1. User enters program creating list (REQ-1-1)
2. User is displayed menu options to modify or view list (REQ-1-2)

Related Requirements	Name	Description	Notes
REQ-1-1	List Class is Initialized	Creates a dynamic array of 4 Item Object	
REQ-1-2	Displays Menu to User	Add Item, Delete Item, Display Shopping List	

3 USE CASE ADDING AN ITEM

UC-2: Adding an Item

Main Success Scenario

1. User selects a menu item to create new item (REQ-2-1)
2. User enters item name (REQ-2-2)
3. User enters type of units. (REQ-2-3)
4. User enters number of units to buy (REQ-2-4)
5. User enters price. (REQ-2-5)
6. User enters c to return to main menu. (REQ-2-6)

Related Requirements	Name	Description	Notes
REQ-2-1	User selects to add new item.	Creates Item Object	
REQ-2-2	Set Function for Item Name called	Function call validates input and allows spaces. Passes string to setItemName which stores the string in the variable itemName	
REQ-2-3	Set Function for unit type is called	Function call validates input and allows spaces. Passes string to setUnit which stores the string in the variable unit.	
REQ-2-4	Set Function for number to buy is called	Function call validates input and only allows numbers to be entered. Passes number to setUnit which stores the double in the variable numberToBuy.	
REQ-2-5	Set Function for price is called.	Function call validates input and only allows numbers to be entered. Passes number to setPrice which stores the double in the variable Price. Price is masked to only allow input in a XXXX.XX format.	
REQ 2-6	List Array size is checked.	Calls ListSize Check Function	
REQ 2-6a	If list is full, list is doubled in size.	doubleList Function is called when array is full.	
REQ 2-7	Item is added to List	addItem function is called in list. Adding item in the next available array spot.	
REQ 2-8	User is returned to main menu	Add Item Display Item Delete Item	

4 USE CASE DISPLAYING LIST

UC-3: Displaying the Item List

Main Success Scenario

1. User selects a menu item to display list (REQ-3-1)
2. List is generated in console(REQ-3-2)
3. User enters c to return to main menu. (REQ-2-3)

Related Requirements	Name	Description	Notes
REQ-3-1	User selects to menu item to display list	Calls displayList Function in class List.	
REQ-3-2a	Display List Function	Outputs position in list, item name, number to buy, unit, and price and on exit Calls totalCost Function which outputs totalCost of list. Informs user to press c to return to main menu.	
REQ-3-2b	totalCost Function	Multiplies the quantity by price adds to a running total.	
REQ-3-3	User returns to main menu	Displays: Add Item Display Item Delete Item menu options	

5 USE CASE DELETING ITEM

UC-4: Delete Item

Main Success Scenario

1. User selects a menu item to delete item(REQ-4-1)
2. List is generated in console(REQ-4-2)
3. User selects a number in list to delete item (REQ-4-3)
4. List is regenerated without item. (REQ-4-4)
5. User is asked if he want to delete another item (REQ-4-5)
6. User presses c to return to main menu. (REQ-4-6)

Related Requirements	Name	Description	Notes
REQ-4-1	User selects to menu item to Delete Item	calls displayList Function in class List.	

Related Requirements	Name	Description	Notes
REQ-4-2a	Display List Function	Outputs position in list, item name, number to buy, unit, and price on exit Calls totalCost Function which outputs totalCost of list. Informs user to press q to return to main menu. See Appendix for Output.	
REQ-4-3	Prompt User to Select Number of Item to Delete		
REQ-4-4	Calls DeleteItem	If number on list exists then deletes that item in the array and returns true. If the number does not exist in array. Returns False	
Req-4-5	User enters c to return to main menu		

Classes

Class-1: List

Function Names	Description	Variables Used	Parameters	Output
List Constructor	Creates a dynamic array of Items of size 4	Item[] itemList, int arrayEnd, size	Void	none
addItem	adds an Item to Item[] itemList	Item newItem; Item[] itemList, arrayEnd	Item newItem	none
checkList	Checks itemList for items, if full calls doubleList	arrayEnd size	Void	none
doubleList	Doubles List Size Copies contents of itemList to another array then double the size of itemList then points ItemList * to the copied array.	Item[] itemList , size, OldSize Item * itemList, Item * array2	Void	none

Function Names	Description	Variables Used	Parameters	Output
displayList	Displays List in Console: Prints out	arrayEnd Item[] itemList		
totalCost	Multiplies number to buy * price and then adds to a toal		Void	Total Cost of Shopping Cart
deleteItem	Deletes Item from itemList, changes size of Array to adjust	Integer position, Item[] itemList	position	The Item has been Deleted Bool return true or false.
Deconstructor	Deletes itemList on exit. Never Used			

Class-2: Item

Function Names	Description	Variables Used	Parameters	Output
Item Constructor	Default Constructor Initializes Item Class	None	None	none
Item Constructor	Override Constructor Initializes	itemName, unit, numberToBuy, Price.	itemName, unit, numberToBuy, Price.	

Function Names	Description	Variables Used	Parameters	Output
	Item Class			
setItemName	Sets Item Name	String itemName	String from user input	none
setUnit	Sets Unit	String Unit	String from user input	none
setNumberToBuy	Sets numberToBuy	Double numberToBuy	double from user input	none
setPrice	Sets price	double price	Double from user input	none
getItemName	Gets Item Name	String itemName		String itemName
getUnit	Gets Unit	String Unit		String Unit
getNumberToBuy	Gets numberToBuy	Double numberToBuy		Double numberToBuy
getPrice	getsPrice	double price		double price

Class-3: Main

Function Names	Description	Variables Used	Parameters	Output
displayMenu	Displays system Menu Example 6-14 in Gaddis			A printout list of menu choices
getChoice	Allows user to enter menu item Example 6-14 in Gaddis	Int choice	InputValid Object	choice

Function Names	Description	Variables Used	Parameters	Output
continueOn	Stops input until c is entered.		InputValid Object	“Printout to screen asking for c”

Class-4: InputValid

Function Names	Description	Variables Used	Parameters	Output
InputValid()	Constructor of InputValidation initializes input to “”	String input	none	
validateInt( )	Validates Positive Int by testing input for the digits.	String input Int myNumber Bool isNotNumber StringStream myStream	none	Positive integer

Function Names	Description	Variables Used	Parameters	Output
validateDouble()	Currently tests by taking an a double and comparing via stringstream to a string. Probably not working as intende.d	String Input Double myNumber Stringstream myStream	none	Double
validatePrice()	Tests for a Price in with the ending .DD. Where D is a digit and . is in the third to last position in the string.	Double myNumber Bool isNotPrice String input Stringstream myStream	None	Double in the form ".XX"
validateString()	Test string for "odd" characters such as backspace entered in console.	Bool isNotPrice String input Stringstream myStream	none	String with either alphabetic, numeric, punctuation or space characters
validateChar()	Currently unused and doesn't have return statement.	Char myChar input	none	none