

Pseudocode

START

partial class ORDER : FORM

DECLARE:

list itemNames

list prices

list stockLeft

string userName

string userEmail

string userAddress

list chosenItemNames

list chosenItemAmount

list chosenItemPrices

decimal chosenItemsTotalPrice

function ORDER

 INITIALISE the components

 DISABLE gbItemList

 DISABLE NumericUpDown

function DOWNLOADITEMS

 TRY

 INIT directory = current directory("Items_Teas_Do_Not_Delete.txt")

 USING streamReader = new StreamReader(directory))

 INIT line = empty

 WHILE line != null

 tempArray = line.SPLIT('|')

 APPEND itemNames (tempArray[0])

 APPEND prices (tempArray[1])

```
APPEND stockLeft (tempArray[2])
```

```
itemNames.REMOVE(0);
```

```
prices.REMOVE(0);
```

```
stockLeft.REMOVE(0);
```

```
CATCH (FileNotFoundException)
```

```
SHOW MessageBox: "Couldn't locate the source file"
```

```
function BTNCREATE_CLICK(sender, e)
```

```
    INIT userName = INPUT
```

```
    INIT userEmail = INPUT
```

```
    INIT iserAddress = INPUT
```

```
    IF userName, userEmail, userAddress == EMPTY
```

```
        SHOW MessageBox: "please enter all details"
```

```
    ELSE
```

```
        ENABLE gblItemList
```

```
        CALL function DOWNLOADITEMS()
```

```
        FOREACH i in itemNames
```

```
            comBoxItem.Append(i)
```

```
function CBITEM_SELECTEDINDEXCHANGED(sender, e)
```

```
    INIT itemIndex = chosen comBoxItem index INPUT
```

```
    FOR i=0; i<prices.COUNT; i++
```

```
    IF i==itemIndex
```

```
        INIT lblPrice = $ + prices[i]
```

```
        INTI lblStockLeft = stockLeft[i]
```

```
        ENABLE NumericUpDown
```

```
        INIT nUDNoOfItems.Maximum = stockLeft[i]
```

```
function BTNADDTOCART_CLICK(sender, e)
```

IF nUDNoOfItems == 0

 SHOW MessageBox "The item amount is zero, please enter the amount"

ELSE IF chosenItemsNames.CONTAINS(comboBoxItem.SelectedItem))

 SHOW MessageBox "You already added this item to cart."

ELSE

 APPEND chosenItemsNames (comboBoxItem.SelectedItem);

 APPEND chosenItemsAmount (nUDNoOfItems)

 APPEND chosenItemsPrices (lblPrice * nUDNoOfItems);

 SHOW MessageBox: "The item {comboBoxItem.SelectedItem} was added to the cart"

function BTNVIEWCART_CLICK(sender, e)

 INIT chosenItemPrice = SUM OF chosenItemPrices

 INIT formaCart = new Cart

 SHOW formCart

partial class CART : FORM

 INITIALISE the components

function FORM2_LOAD(sender, e)

 INIT lblName = txtName

 INIT lblEmail = txtEmail

 INIT lblAddress = txtAddress

 IF chosenItemsName.Count > 0

 THEN

 FOR i=0; i<chosenItemsName.Count; i++

```

        APPEND lblItems (chosenItemsName[i])

        APPEND lbNumsRequired (chosenItemsAmount[i])

        APPEND lbTotalPrices (chosenItemsPrices[i])

    INIT lblTotal = $ + chosenItemsTotalPrice

ELSE

    PRINT error message "add items to cart"

function btnFinal_Click(sender, e)

    TRY

        SET directory = current directory + userName + current DateTime + .txt

        INIT txt = createText(directory)

        PRINT TO txt

            "Customer's name: {userName}"

            Customer's email: {userEmail}

            Customer's address: {userAddress}

            Invoice Date: {DateTime.Today}

            Item Name | No of Items Ordered | Total Item Price"

        FOR i=0; i<chosenItemsNames.Count; i++

            txt.WriteLine($"{Order.chosenItemsNames[i]} | {Order.chosenItemsAmount[i]} |
${Order.chosenItemsPrices[i]}")

            PRINT TO txt: "Total Price: {chosenItemsTotalPrice}"

        CLOSE txt

        SHOW MessageBox: "Order Placed! The invoice can be found here: {directory}"

    CATCH(IOException)

        SHOW MessageBox:"Error: Couldn't save the file"

END

```

Project documentation

This project is aiming to create a windows application form using C# to order items.

There are two forms – the Order form and the Cart form.

The Order form contents:

- Group Box for Customer Details:
 - Label and Text Box for Name
 - Label and Text Box for Email
 - Label and Text Box for Address
 - Button for Create Order
- Group Box for Item List:
 - Label and Combo Box for Item
 - 2 Labels for Item Price
 - 2 Labels for Items Left
 - Label and Numeric Up Down for No of Items Needed

The Order form is created to collect the information from the user (their name, email and address), which must not be null.

When the information has been fed, the Create Order button enables the 2nd part of the form – the Item information.

Here a user can choose from a list of available items, which in turn will show the item details – its price and stock left (also enabling the Numeric UpDown control). A user can choose the amount of the same item they'd like to add to the cart (with minimum being 1 and maximum – the amount left). A user can repeat the same action with other items.

When a user is done with adding items to the cart, they can press the View Cart button, which opens the 2nd form.

The Cart form contents:

- Label and List Box for Items
- Label and List Box for Nums Required
- Label and List Box for Total Prices
- Split Container:
 - 2 Labels for Name
 - 2 Labels for Email
 - 2 Labels for Address
 - 2 Labels for Total Price
 - Button for Place Order

The form illustrates the result of the actions in the previous form. It shows a list of all items added to the cart, the amount of each item required and the total price of each item times the amount. It also shows the customer's details and the total price for all items in the cart.

A user can then press the Place Order button, which will save all this information in .txt file and place it in the same directory as the application.

Functions:

The Order form.

At first the public static variables are declared inside the partial class Order : Form. This makes them available for use in different functions and forms.

public Order () initialises the components and disables the Item groubox, leaving only the first half of the form available at the start of the application. Also disables the Numeric UpDown.

private static void downloadItems(): the directory to the .txt file with the items' details is stored in the variable directory and read by the StreamReader.

The strings are split by the "|" symbol and line-by-line stored in 3 separate lists (every 0th element in the 1st list, every 1st element in the 2nd list and every 2nd element in the 3rd list).

All 0th elements are removed from all 3 lists as they contain the column titles (Item | Item Price | Stock Left).

The whole function is checked by try-catch to make sure the file can be found and read. If not, an error message is displayed.

private void btnCreate_Click(object sender, EventArgs e): when the Create Order button is clicked, the program stores the details provided by the user in 2 different variables. If no information was provided, an error message is displayed.

When details are stored, the Item groupbox gets enabled. The **downloadItems()** function gets called and combobox Item gets appended with the item names.

private void cbItem_SelectedIndexChanged(object sender, EventArgs e): when a user chooses an item from the combobox, it's index is stored in a variable.

The lists of prices and stock left are ran through and the corresponding indices are stored in 2 variables.

The Numeric UpDown is enabled. The maximum number for it is set to the stock left amount.

private void btnAddtoCart_Click(object sender, EventArgs e): If a user chooses zero when deciding on the amount of the same item added to the cart, an error message pops up.

Also, if a user tries to add the same item after already adding that to the cart, an error message pops up.

Else, all chosen items are stored to a list, and the corresponding amounts and total prices are stored to 2 lists.

private void btnViewCart_Click(object sender, EventArgs e): A total price for everything is calculated. The other form is opened.

The Cart form.

private void Form2_Load(object sender, EventArgs e): when the form is loaded, the customer's details are displayed in the form.

An error message is shown if a user didn't add any items to the cart, else all items come up in one list box, amounts in another and prices in the 3rd.

private void btnFinal_Click(object sender, EventArgs e): when the Place Order button is clicked, the program saves all the requested item's information in the base directory in the .txt format.

In case there is a problem with saving that, an error message is shown.