

# INEW 2332 Customer View Requirements

(2022.05.03)

Page 1 of 6

## Customer View

**All overall project wide requirements and standards apply.**

This is the outward facing application that an end user would use.

**\*\*\*Must be able to choose and purchase from a minimum of 25 unique and fully populated inventory items, not including variants, of individual inventory items stored on the SQL server\*\*\***

- Customers can browse all products without an account.
- Customers must have an account to purchase items.
- Customers can use a search feature to find available products and can filter their results based on categories.
  - Application must be designed with the assumption that there are thousands of inventory items available
- The customer can search for and interact with the product which will interface with the item details stored in the SQL server.
  - Product Name
  - Expanded description
  - Price
  - Quantity on hand
  - Examples include a customer viewing the different makes and models of a car, browsing menu items at a restaurant, browsing video games or looking at different action figures, various option packages and colors for a car, add or remove toppings from a pizza, or seeing multiple variants and cost of a particular figure.
- The user will be able to “purchase” items by placing the item(s) into a shopping cart (or equivalent) with the application keeping track of their choices.
  - All form level controls will be updated as items will be added or removed from the cart.
  - Multiple quantities of an individual inventory item will only have one entry in the shopping cart that displays the current quantity to be purchased
  - Database quantities are updated only after successful checkout.

# INew 2332 Customer View Requirements

(2022.05.03)

Page 2 of 6

- Customers can enter a known valid promotion code to receive a discount. No searches or browsing for codes. Users must enter promotion code in a textbox field. Example: **BOGO50**
- The customer must enter a simulated credit card number, card expiration date, and security code before completing the transaction.
  - Card is not accepted if past the expiration date (within 5 years of current year) or invalid card number format.
  - Example:
    - VISA/MasterCard 1234-5678-1234-5678
    - CCV: 123
    - Exp: 07/2026 or 07/26
- The shopping cart and final receipt/invoice upon checkout will display:
  - Line item:
    - Product Name
    - Item Price
    - Quantity
    - Line item total
  - Subtotal of all line items
  - *(Discount if applicable)*
    - Percentage, displayed as 10%, 20%, etc.
    - Actual discount amount in dollars and cents.  
USD Currency format (example \$12.34)
  - *(New subtotal after discount if applicable)*
    - New discount subtotal, represents previous subtotal minus discount amount. (Example: Subtotal \$10.00, Discount 10% (.10), New discounted subtotal is \$9.00)
  - Tax (8.25%) on subtotal
  - Final Purchase Amount
- Produce a printed HTML receipt/invoice when the user has “checked out” or completed the sale.
- After purchase, display the HTML receipt using the user’s default web browser.
- All produced HTML files will be stored in the user’s Documents\YourApplication folder. (Microsoft’s SpecialFolders)
- No Crystal Reports usage allowed for this application.

# INEW 2332 Customer View Requirements

(2022.05.03)

Page 3 of 6

## Example:

TSTC Bookstore			
Customer: Jenny J. Phone: 867-5309 Sales Receipt #123456			
Item	Item Price	Quantity	Total Price
Programming for Champs	\$15.25	x1	\$15.25
5 Hour Energy	\$1.93	x12	\$23.16
		Subtotal:	\$38.41
		Tax (8.25%):	\$3.17
		Total:	\$41.58
Thank you for your business!			

## Discount Example:

TSTC Bookstore			
Customer: Jenny J. Phone: 867-5309 Sales Receipt #123456			
Item	Item Price	Quantity	Total Price
Programming for Champs	\$15.25	x1	\$15.25
5 Hour Energy	\$1.93	x12	\$23.16
		Subtotal:	\$38.41
Discount		10%	-\$3.84)
		Discounted Subtotal	\$34.57
		Tax (8.25%):	\$2.85
		Total:	\$37.42
Thank you for your business!			

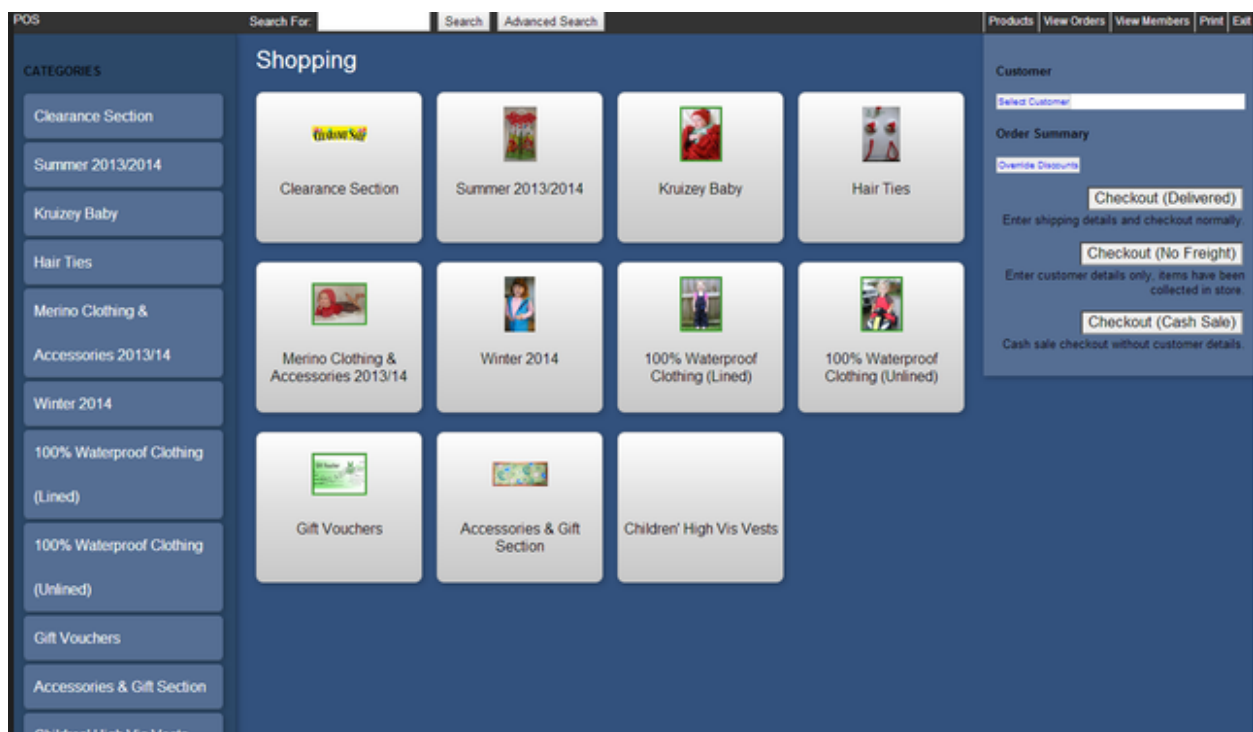
# INEW 2332 Customer View Requirements

(2022.05.03)

Page 4 of 6

- The above purchases will need to be stored in the SQL server database in the appropriate table as well with the appropriate invoice numbers and items purchased.
  - Stored data must be in Third Normal Form (3NF) with no duplicated data.
  - Invoice Number
  - Date/Time
  - and other Fields as Appropriate (while in 3NF)
- The Customer View will need to have specific, contextual help files available.

## Example Screen:



# INEW 2332 Customer View Requirements

(2022.05.03)

Page 5 of 6

## Database Usage

1. **Inventory** table holds all data related to each item in the database
  - a. Application must be able to accept data for every field in the database
  - b. Images must be unique for each item
2. **Category** table holds all data related to the types of items in the database
  - a. Each inventory item must belong to a category
  - b. Categories are used to help the user filter results within your application
  - c. Category can be further refined with the addition of a 'SubCategory' table linked to Category (by FK) if needed or preferred for the Project in order to provide appropriate Filtering/Search capabilities
3. **Orders** table holds data (in 3NF) that contains all the information for completed orders
  - a. No computed values will be stored
  - b. No duplicated Person or Inventory information
4. **OrderDetails** table holds "Line Item" details (in 3NF) for each order in the Orders table.
  - a. No computed values will be stored
  - b. No duplicated Person or Inventory information

# INEW 2332 Customer View Requirements

(2022.05.03)

Page 6 of 6

## Mandatory Classes

Separate code from data in a way that code resides in functions whose behavior does not depend on data that is somehow encapsulated in the function's context.

When we separate code and data, our programs benefit from:

1. Code can be reused in different contexts
2. Code can be tested in isolation
3. Systems tend to be less complex

When we separate code and data, it is straightforward to reuse code in different contexts.

**clsHTML** - All HTML receipts, reports, and invoices will be produced from this class.

- Will accept objects from the "Cart" such as a list or any other methodology to produce rows of data as .html files.
- clsHTML.cs will also handle all formatting through Cascading Style Sheets, to be produced "inline" within the same .html file.

**clsValidation (Appended)** - Handles input validation for controls or user input **before** reaching the server

- Validate TextBox input of required fields
  - Credit Card Number
  - Expiration Date
  - Quantity
  - ...etc.

**clsSQL** - Handles connections, INSERT/UPDATE/DELETE, and SELECT

- Schema names must be a class level constant.
- No query strings will be passed to the clsSQL.cs
- Efficient code that uses a minimal amount of methods/functions for connecting to the database. Connection strings exist in only one location (global variable or constant).
- Efficient code that uses a minimal amount of methods/functions for handling data manipulation. Appropriate methods for each of the DML statements.
- Efficient code that uses a minimal amount of methods/functions for querying the database. Methods usage can include populating application controls.
- Efficient code that uses a minimal amount of methods/functions for handling SQL Exceptions. SQL specific exception handling is better than generic handling.
- All SQL related strings (connections strings, query strings, etc) contained inside of the class. Pass parameters or references and use overloading.