

Design Document

Methods and Tools in SW Development

I. Group Information

Group Number: 20

Group Member names/netIDs:

- Joshua Rowland
 - jbr390
- Jason Knight
 - jk1834
- Gabriel Becker
 - gjb125
- Brooke Burton
 - bb2468

Item:

_____ We need an item class that holds information about specific items that can be viewed, edited, and checked out.

User:

_____ The user class is needed to hold references to their own shopping carts, usernames and passwords.

Inventory:

_____ Need an inventory class to hold the categories and lists of items, along with all of their attributes.

Shopping Cart:

_____ This class is needed to hold a list of items. Also enables the user to add and remove from the cart at will.

II. Detailed Class Diagrams

Item
<u>Attributes:</u> <ul style="list-style-type: none">- name: string- category: string- price: float
<u>Functions:</u> <ul style="list-style-type: none">+ Item(name: string, category: string, price: float)+ ~Item()+ get_name(): string+ get_category(): string+ set_price(price: float): void+ get_price(): float

- Item(name: string, category: string): Constructor
- ~Item(): Destructor
- get_name(): Gets the name of the item
- get_category(): Gets the category of the item
- set_price(price: float): Sets the price of the item
- get_price(): Gets the price of the item

User
<u>Attributes:</u> <ul style="list-style-type: none"> - cart: ShoppingCart - username: string - name: string - password: string - cardnum: int - address: string
<u>Functions:</u> <ul style="list-style-type: none"> + User() + ~User() + login(username: string, password: string): void + logout(): void + get_shipping_info(): string + set_shipping_info(address: string): void + get_payment_info(): int + set_payment_info(cardnum: int): void + delete_account(): void + checkout(): void

- User(): Constructor
- ~User(): Destructor
- login(username: string, password: string): Logs a user in
- logout(): Logs a user out
- get_shipping_info(): Gets the address associated with the user
- set_shipping_info(address: string): Sets the address associated with the user
- get_payment_info(): Gets the card number associated with the user
- set_payment_info(cardnum: int): Sets the card number associated with the user
- delete_account(): Deletes a user's account
- checkout(): Checks the user out (removes all items from their shopping cart and removes the item from stock)

ShoppingCart
<u>Attributes:</u> - cart: Item[0..*]
<u>Functions:</u> + ShoppingCart() + ~ShoppingCart() + add(item: Item): void + remove(item: Item): void + remove_all(): void + view(): Item[0..*]

- ShoppingCart(): Constructor.
- ~ShoppingCart(): Destructor.
- add(): Adds an item to the shopping cart.
- remove(): Removes an item from the shopping cart.
- remove_all(): Clears the shopping cart.
- view(): Prints the entire shopping cart.

Inventory
<u>Attributes:</u> - stock: [0..*]
<u>Functions:</u> + Inventory() + ~Inventory() + add_stock(item: Item, amount: int): void + remove_stock(item: Item, amount: int): void + view_stock(): Item[0..*] + view_category(category: string): Item[0..*] + update(): void

- Inventory(): Constructor.
- ~Inventory(): Destructor.
- add_stock(): Adds items to the stock list.
- remove_stock(): Removes items from the stock list.
- view_stock(): Prints the entire stock list.
- view_category(): Prints all items from the stock list in the given category.
- update(): Reads from file and updates stock accordingly.

III. Menu Information

Before login:

- Login
- Create Account
- Exit Program

After login:

- Logout
- Cart Information
 - Go back
 - View Cart
 - Remove Item from Cart
 - Check out
- View all Items
- View Items by Category
- View Order history
- Edit Shipping Info
- Edit Payment Info
- Delete Account
- Add Item to Cart
- Exit Program

Does your menu cover all requirements given? If not, explain why certain requirements don't have a distinct menu option?

Yes!

IV. Information Storage

How is your group storing information?

We are storing all of our information in files. We will primarily be working with three files, Users, Stock, and Previous Orders.

What information are you going to store in each (table / file depending on schema)?

- User
 - Username
 - Password
 - First Name
 - Last Name
 - Address
 - Payment Info
- Stock
 - Categories
 - Item Name
 - Item Amount
 - Item Price
- PreviousOrders
 - Username
 - List of Previous Items Checked Out with a Timestamp