

Design Document

Methods and Tools in SW Development

I. Group Information

Group Number: 4

Group Member names/netIDs:

- Cooper Earls
 - <cce114>
- Milton Lewis
 - <ml1462>
- Allison Lyons
 - <acl523>
- Eleftherios Vganges
 - <epv15>
- Jake Wilson
 - <jbw622>

What classes are you going to have? Explain why.

<User List>

This class will handle all functions having to do with the User, including the User database.

<Inventory>

This class will have the functions used to display the store's inventory while inside the terminal.

<Shopping Cart>

This class will have the functions that have to do with the Shopping Cart.

<Orders>

This Class will handle the user's order history as well as some of the store's featured item generation.

<Item>

This class will be the other half of the Inventory class by managing items in the inventory.

II. Detailed Class Diagrams

Inventory
-movieTitle: string -director: string -gameTitle: string -designer: string -itemNum: int*
+movies() +games() +all() +popularItems(): void +discounted(): void

- Function 1 – Calls for movies in main
- Function 2 – Calls for games in main
- Function 3 – Calls for all movies and games
- Function 4 – Sorts items that are popular with customers and uses information from orders and shopping cart
- Function 5 – sorts items that are discounted

Shopping Cart
-cart: vector<Item>
+addGame() +addMovie +removeItem() +checkout() +viewCart()

Function 1 addGame() adds a game to the cart vector.

Function 2 addMovie() adds a movie to the cart vector.

Function 3 removeItem() removes an item from the cart vector.

Function 4 checkout() allows the user to checkout all items in the cart vector to buy.

Function 5 displays a list of all items in the cart to the user.

USER
-userName : char* - UserAddress : char* -UserEmail : char* -UserPayment : char*
+ getUsername() : char* +getUserAddress() : char* + getEmail() : char* + getUserPayment() : char + editUser() + deleteUser()

- Function 1: getUsername () - asks the user for the account user name
- Function 2: getUserAddress () - asks the user for the address for the account
- Function 3: getEmail () - ask the user for the email for the account
- Function 4: getUserPayment () - ask the user for the account's payment info
- Function 5: editUser () - allow the user to manipulate user account data.
- Function 6: deleteUser () - deletes the user's account

ORDERS
-OrderID: int*
+addOrder(): int* +removeOrder(): int* +listOrder():

- Variable 1: OrderID - a variable allowing ORDERS to temporarily store data.
- Function 1 : addOrder() will add an Order to a Users Order History
- Function 2: removeOrder() will remove an Order from a Users Order History
- Function 3: listOrder() will list out all the Orders under a Users Order History

ITEM
-itemcount: int* -itemID: int* -Discounted: vector<itemID> -HotItems: vector<itemID>
+lowerItemAmount(): void +raiseItemAmount(): void +genNewlyDiscounted() +genHotItems()

- Variable 1 – itemcount simply is the current item count of the item
- Variable 2 – to hold an itemID
- Variable 3 – a list of Discounted books for Inventory
- Variable 4 – a list of Hot Items for Inventory
- Function 1 – is called automatically when an Item is bought to go in and decrease the current count of said item
- Function 2 – the same as above but for in the scenario in which an items needs to be added back
- Function 3 – this will look for items that have been in the stock for a long time and assign them to the newly discounted list
- Function 4 – this will sort for items most recently bought and add them to the HotItems list

III. Menu Information

~~~Welcome to the E-Commerce Storefront~~~

1. Login
2. Create Acct
3. Exit -

~~~Menu~~~

1. Browse Inventory
2. View My User Info
3. View My Order History
4. View Cart
5. Logout – Goes back to Start

~~~Inventory~~~

1. Newly Discounted
2. Hot Items
3. View All Inventory
4. View All Games
5. View All Movies
6. Back

~~~My User Info~~~

- John Smith
140 Willabye Ave 54321
johnsmith@gmail.com
1. Edit User Info
 2. Edit Shipping Info
 3. Edit Payment Info
 4. Delete User – do this 2x (are you sure?)
 5. Back

~~~~~My Order History~~~~~

1. View Order History
2. Clear Order History
3. Back

~~~~Cart~~~~

Movie 1 \$19.99

Game 1 \$12.99

<- (View All Items in Cart)

Movie 2 \$12.31

Game 2 \$27.98

1. Checkout
2. Remove an Item from Cart
3. Save Cart
4. Back

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

Yes, the menu covers all the requirements

IV. Information Storage

How is your group storing information?

Include one of these lines of questioning based on your storage schema:

- If a database, what kind of database?
 - How many database tables will you have?
- If files, how will your files be format?
 - How many files are you going to use?

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

We will be using 4 SQL Databases.

There will be one for a User List, Item List, Cart List, and Order History.

User List will store User Name, Password, Email, Address, and payment info

Item List will store the Items name (game / movie), their type of item, genre of item, author/director, publisher/designer, ItemID, and Item Price

Cart List would store primary key, itemID (foreign key for each item in cart linked to respective row in item list)

Order History will have the primary key for that Order, a list of Item ID's in that Order, a foreign key of the UserID