Design Document

Methods and Tools in SW Development

**I. Group Information**

Group Number: 1

Group Member names/netIDs:

* Caden Austin
  + CBA169
* Caleb Byers
  + CTB388
* Darcie Dalafave
  + DAD514
* Justin Pettiss
  + JP2973
* Marcus Bridgman
  + MB3668

What classes are you going to have? Explain why.

Customer – will have customer information

Address – Address Information

Payment Info – Payment info information

Order – Historical orders from a user

Shopping Cart – Stores the items temporarily for a user

Item – Parent class for book and movie

Book – Book Class (one of the items for purchase)

Movie – Movie Class (one of the items for purchase)

**II. Detailed Class Diagrams**

|  |
| --- |
| **Customer** |
| -id: int  +name: string  +shippingAddress: Address  +paymentInfo: PaymentInfo  -email: string  -passwordHash: string  +shoppingCarts: List<ShoppingCart>  +orders: List<Order> |
| + Customer(email, password, name)  + getID(): int  + getEmail(): int  + authenticateCustomer(password): boolean  + changePasssword(previousPassword, newPassword)  + addShoppingCart(ShoppingCart)  + removeShoppingCart(ShoppingCart)  + addOrder(Order)  + deleteAccount(password): bool  - checkPassword(password): bool |

* Customer – Constructor that takes a name, email, and password. Will hash the password for storing using SHA256
* getID – Returns the id
* getEmail – Returns the customers email
* authenticateCustomer – Authenticates user by hashing their plaintext password and comparing it to the password hash; returns a boolean of if the user was successfully authenticated
* changePassword – Takes the current password and verifies it’s correct then changes the users password to the provided one
* addShoppingCart – Takes a shopping cart instance and adds it to the list
* removeShoppingCart – Removes a shopping cart instance from the list if currently on the in the shoppingCarts list
* addOrder – Adds an order instance to the orders list
* deleteAccount – Verifies the password input, then cascading delete all the orders and shopping carts the user has and deletes the user after they’ve been deleted
* checkPassword – Checks the user’s password to see if matches the password hash

|  |
| --- |
| **Address** |
| +addressLineOne: string  +addressLineTwo: string  +city: string  +state: string  +zip: int |
| + Address() |

* Address – Constructor for address, initializes data to None

|  |
| --- |
| **PaymentInfo** |
| +cardNumber: string  +nameOnCard: string  +expirationDateMonth: int  +expirationDateYear: int  +cvv: int  +billingAddress: Address |
| + PaymentInfo()  + getExpirationDate(): Dict<string, int> |

* PayementInfo – Constructor for Payment Info, initializes data to None
* getExpirationDate – Returns dict of ints with [“month”: month, “year”: year] format

|  |
| --- |
| **Order** |
| -orderDate: Date  +deliveryAddress: Address  +paymentInfo: PaymentInfo  +items: dict<item, int> |
| + Order() |

* Order – Constructor for order, initializes data to None

|  |
| --- |
| **Shopping Cart** |
| +items: dict<item, int>  +subtotal: float  +total: float |
| + ShoppingCart()  + setItems(item, quantity): boolean  + removeItem(ite): boolean  + checkout() |

* ShoppingCart – Constructor for Shopping Cart, initializes data to None
* setItem – Add item to items dict, quantity defaults to 1, recalculates subtotal/total
* removeItem – Remove item from items dict, recalculates subtotal/total
* checkout – Deletes the users cart and moves the necessary information to orders

|  |
| --- |
| **Item** |
| +quantity: int  +title: string  +description: string  +genre: string  +price: float |
| +Item() |

* Item – Constructor for item, initializes data to None, shouldn’t be called unless movie or book is inheriting

|  |
| --- |
| **Book** |
| +isbn: string  +author: string  +publisher: string |
| + Book() |

* Book – Constructor for book, initializes data to None, inherits from item
* Getters – Returns the value stored in the class instance
* Setters – Sets the value stored in the class instance

|  |
| --- |
| **Movie** |
| +director: string  +leadingActor: string |
| +Movie() |

* Movie – Constructor for movie, initializes data to None, inherits from item

**III. Menu Information**

Before login:

* Login
* Create Account
* Exit Program

After login:

* Manage Account
  + Go Back
  + Change Name
  + Change Password
    - Prompt user for current and new password
  + Change/Add Shipping Address
    - Prompt with user a form for address information
  + Change/Add Payment Info
    - Prompt user with payment details
      * Prompt user with with an address form for the billing address
  + View Previous Orders
    - Display orders in formatted list
  + Delete Account
    - Warn user before action and have a password confirmation
* Store Front
  + Go back
  + Display all books
    - Go back
    - Add Items to Cart
  + Display all movies
    - Go back
    - Add Items to Cart
  + Filter search books and movies
    - Go back
    - Add Items to Cart
  + Search by title books and movies
    - Go back
    - Add Items to Cart
* Cart Information
  + Go back
  + View Cart
  + Remove Item from Cart
  + Checkout Cart
* Logout
* 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?

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

* If a database, what kind of database? SQLite
  + How many database tables will you have? 8

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

*Sample Answer:*

* *Customer*
  + *id - PK*
  + *name*
  + *email*
  + *password\_hash*
  + *shipping\_address\_id – FK*
  + *payment\_information\_id – FK*
* *ShoppingCart*
  + *cart\_id - PK*
  + *user\_id – FK*
  + *subtotal*
  + *total*
* *CartItem*
  + *id – PK*
  + *cart\_id – FK*
  + *item\_id – FK*
  + *sub\_total*
  + *quantity*
* *InventoryItem*
  + *item\_id – PK*
  + *quantity*
  + *title*
  + *description*
  + *genre*
  + *price*
  + *item\_type (either ‘M’ or ‘B’)*
  + *item\_reference – FK (Will exist in movie or book table depending on itemtype flag)*
* *Movie*
  + *id - PK*
  + *director*
  + *leading\_actor*
* *Book*
  + *id - PK*
  + *author*
  + *publisher*
  + *isbn*
* *Address*
  + *address\_id - PK*
  + *street\_one*
  + *street\_two*
  + *city*
  + *state*
  + *zip*
* *PaymentInfo*
  + *payment\_id – PK*
  + *billing\_address\_id - FK*
  + *card\_number*
  + *cvv*
  + *expiration*