Design Document
Methods and Tools in SW Development

1. Group Information

Group: Section 501

Group Members:

- Madison Bailey (mbb496)
- Jeffery Burns (jcb1251)
- Caleb Germany
- Molly Granger (mmg423)
- Spencer Hall (jsh278)
- Caleb Hickman-Thomas (cjh767)

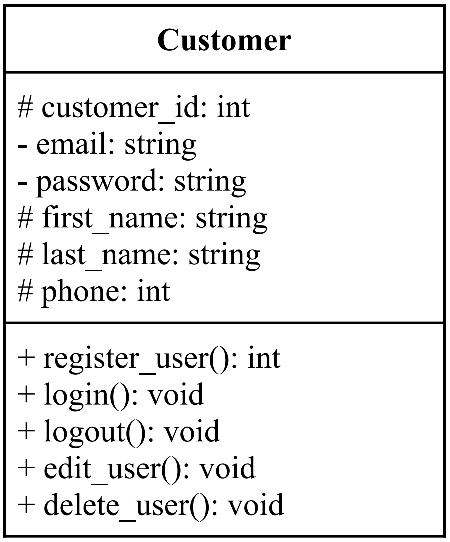
Class Overview:

Our group decided on four base classes: Customer, Product, Cart, and Store. We broke three of these classes down into one or two derived classes; we felt this would lead to easier management of their data, as well as aid in development and testing of the software and databases.

- **Customer** – *base* – will include only basic customer information
  - **Payment Info** – *derived* - will include everything required to make a purchase
  - **Shipping Info** – *derived* - will include everything required to ship an order
- **Product** – *base* – will include only basic product information shared by all product types
  - **Book** – *derived* – will include information related to book products
  - **Game** – *derived* – will include information related to video games products
- **Cart** – *base* – will be used to track items in customers' carts
  - **Order** – *derived* – will be used to track individual orders
- **Store** – *base* – will have information about the store for business and customer support

2. Detailed Class Diagrams

2.1. Customer



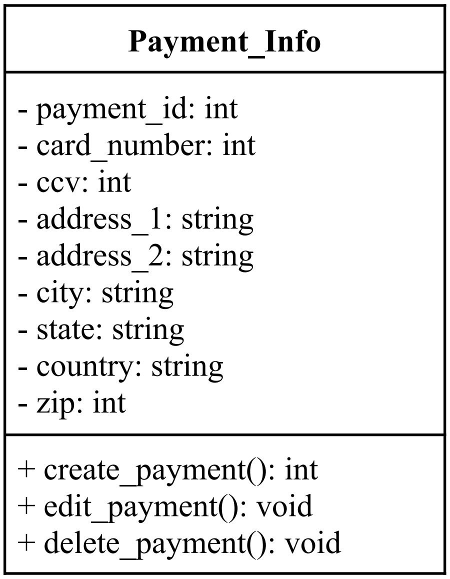
Customer

# customer_id: int
- email: string
- password: string
# first_name: string
# last_name: string
# phone: int

+ register_user(): int
+ login(): void
+ logout(): void
+ edit_user(): void
+ delete_user(): void

The following attributes are necessary in the Customer class which include user_id, email, password, first_name, last_name, and phone. These are for retaining basic user data that is utilized when making a purchase in the store and to differentiate each individual user. We also included functions to both register a new user and update the user information.

2.1.1. Payment Info



Payment_Info
- payment_id: int - card_number: int - ccv: int - address_1: string - address_2: string - city: string - state: string - country: string - zip: int
+ create_payment(): int + edit_payment(): void + delete_payment(): void

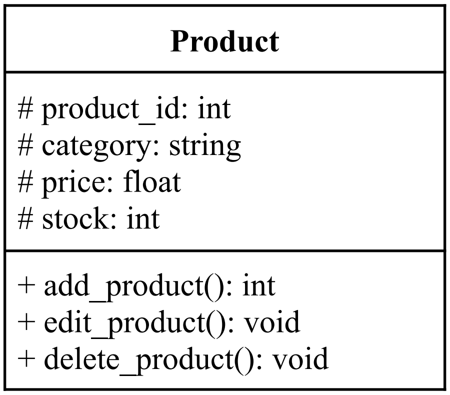
The following attributes are necessary in the Payment_Info class which includes payment_id, card_number, ccv, address, city, state, and zip. These serve the purpose of retaining basic payment information that is utilized when making a purchase in the store. In addition, we included the function to update your payment information to ensure that it is current and accurate.

2.1.2. Shipping Info

Shipping_Info
- shipping_id: int - address_1: string - address_2: string - city: string - state: string - country: string - zip: int
+ create_shipping(): int + edit_shipping(): void + delete_shipping(): void

The following attributes are necessary in the Shipping_Info class which includes shipping_id, address_1 and _2, city, state, country, and zip. These are for retaining basic shipping information that is utilized when making a purchase in the store. In addition, we included the function to edit your shipping information to ensure that it is up to date.

3. Product



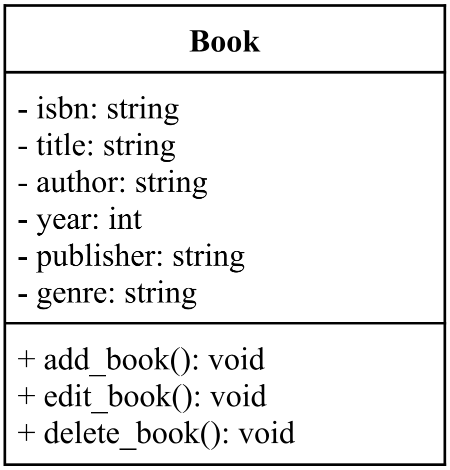
Product

# product_id: int
# category: string
# price: float
# stock: int

+ add_product(): int
+ edit_product(): void
+ delete_product(): void

This table will contain relevant information for each item in the store which includes the product_id to be able to identify the unique items along with the category, price and stock of the items. The functions that are used for this table will allow us to add or remove a quantity of an item and, in addition, update the price of an item.

3.1.1. Book



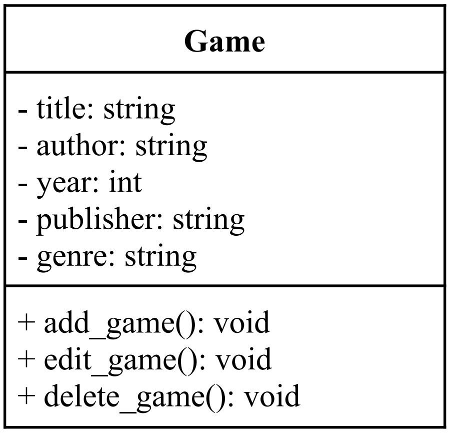
Book

- isbn: string
- title: string
- author: string
- year: int
- publisher: string
- genre: string

+ add_book(): void
+ edit_book(): void
+ delete_book(): void

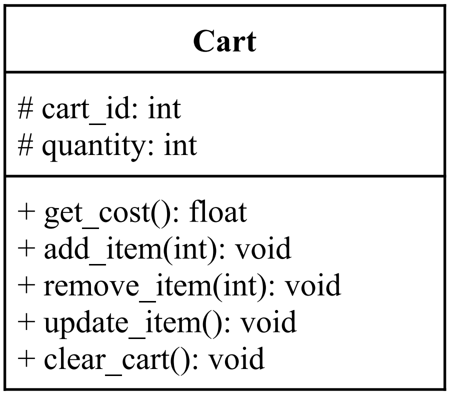
The following attributes are necessary in the Book class which includes isbn, title, author, year, publisher, and genre. These are here to store specific entries on books in the inventory, and will be available for viewing in the user’s cart. Functionality includes adding, removing, or editing a book from the inventory.

3.1.2. Game



The following attributes are necessary in the Game class which includes title, author, year, publisher, and genre. These are for showing the information about the game before adding it to the cart. Functions for this class include add_game, edit_game, and delete_game for inventory management.

4. Cart



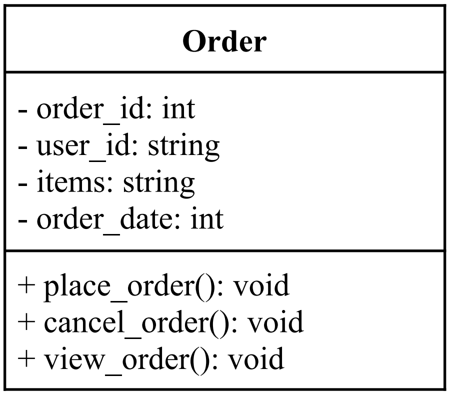
Cart

# cart_id: int
# quantity: int

+ get_cost(): float
+ add_item(int): void
+ remove_item(int): void
+ update_item(): void
+ clear_cart(): void

The attributes of the Cart class include cart_id and quantity. The cart_id serves as the unique identifier for each transaction. As for the functions within this table, we included the ability to add each item or remove it if needed. In addition, we considered that we will need the user to be able to see the total cost of their cart, the ability to update the amount of an item that they want, and the ability to clear the cart altogether.

4.1.1. Order



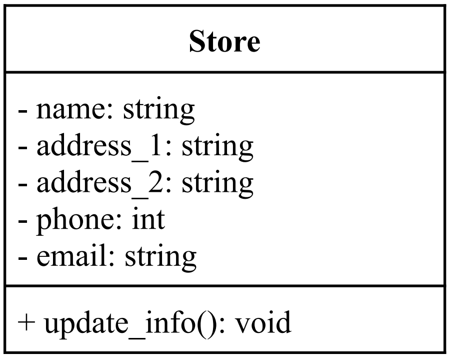
Order

- order_id: int
- user_id: string
- items: string
- order_date: int

+ place_order(): void
+ cancel_order(): void
+ view_order(): void

The Order table will include the attributes order_id, user_id, items, and order_date. Each order will be uniquely identified by an order_id, allowing operators to track the order details and the customer who placed it, while each user will have a unique identifier in the form of a user_id. The item attribute is necessary, as each order includes a number of different items to be purchased, while the order_date is utilized to track the date that each order was placed on. In addition, the table will include the functions place_order and cancel_order. These allow the user to place an order on the books and games and view the order contents. If the user becomes unhappy with their purchase, he/she also has the option to cancel it.

5. Store

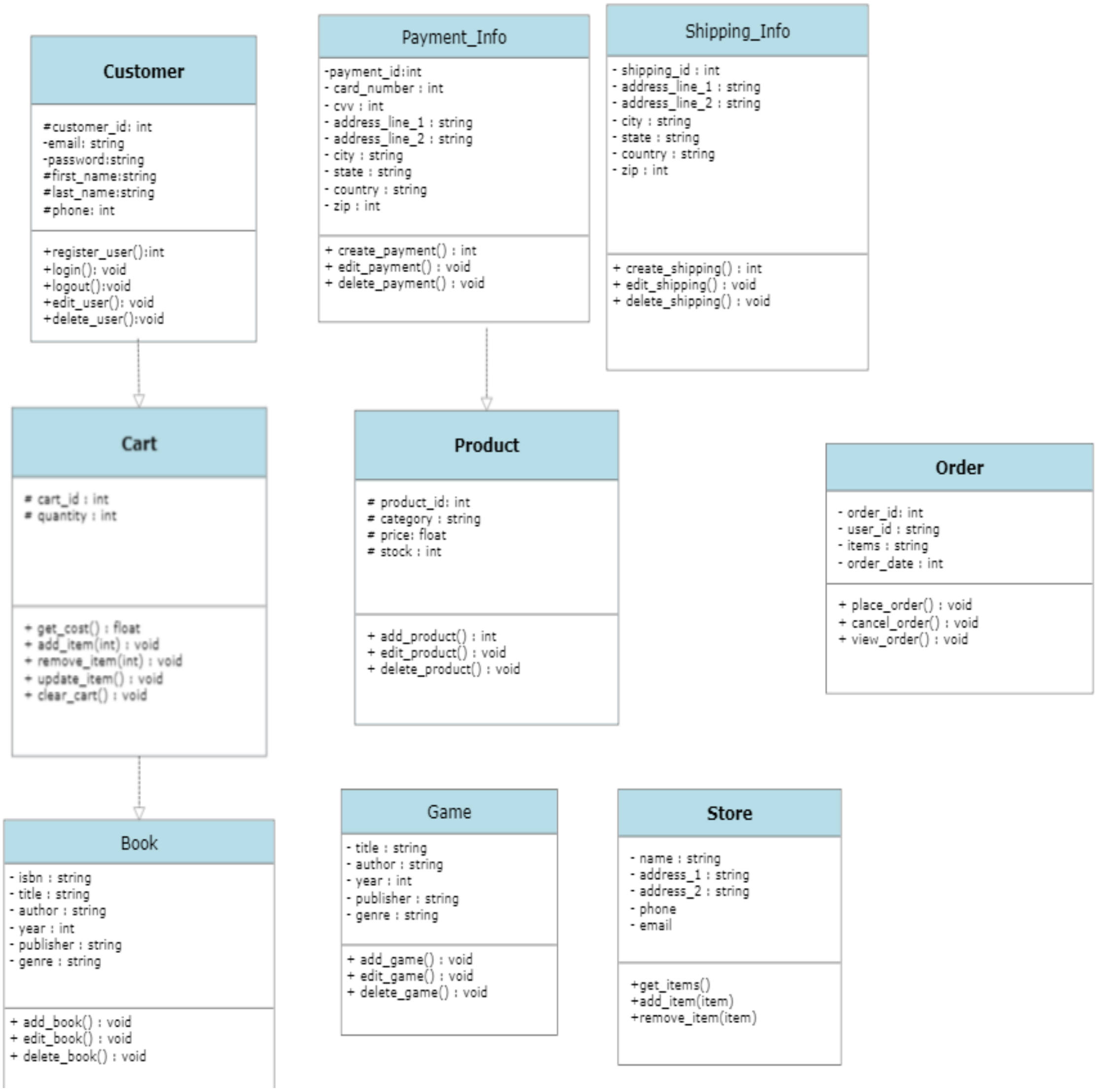


Store

- name: string
- address_1: string
- address_2: string
- phone: int
- email: string

+ update_info(): void

Within the store table we made sure to include the unique identifier for the store which is the StoreID. We also made sure to include basic details about the store for the user to view which includes the name, address_1 and _2, phone, and email. The function update_info is there in case the information describing the store needs to be changed.


Customer
#customer_id: int
-email: string
-password:string
#first_name:string
#last_name:string
#phone: int
+register_user():int
+login(): void
+logout():void
+edit_user(): void
+delete_user():void
Payment_Info
-payment_id:int
- card_number : int
- cvv : int
- address_line_1 : string
- address_line_2 : string
- city : string
- state : string
- country : string
- zip : int
+ create_payment() : int
+ edit_payment() : void
+ delete_payment() : void
Shipping_Info
- shipping_id : int
- address_line_1 : string
- address_line_2 : string
- city : string
- state : string
- country : string
- zip : int
+ create_shipping() : int
+ edit_shipping() : void
+ delete_shipping() : void
Cart
# cart_id : int
# quantity : int
+ get_cost() : float
+ add_item(int) : void
+ remove_item(int) : void
+ update_item() : void
+ clear_cart() : void
Product
# product_id: int
# category : string
# price: float
# stock : int
+ add_product() : int
+ edit_product() : void
+ delete_product() : void
Order
- order_id: int
- user_id : string
- items : string
- order_date : int
+ place_order() : void
+ cancel_order() : void
+ view_order() : void
Book
- isbn : string
- title : string
- author : string
- year : int
- publisher : string
- genre : string
+ add_book() : void
+ edit_book() : void
+ delete_book() : void
Game
- title : string
- author : string
- year : int
- publisher : string
- genre : string
+ add_game() : void
+ edit_game() : void
+ delete_game() : void
Store
- name : string
- address_1 : string
- address_2 : string
- phone
- email
+get_items()
+add_item(item)
+remove_item(item)

6. Menu Information

Before login: Welcome message: “Welcome to our online store! Please login or create an account to start!”

3 options:

- Create a new account
- Login
- Quit

After login: Different interactions are available to be made:
View items, Shopping Cart, Account Management, Order History, Logout

Items:
Item ID
Item name
Item description
Item price
Item quantity in stock
Item category

Shopping Cart:
View cart
Remove item from cart
Add item to cart
Checkout
Clear cart
Get total cost

Account Management/User:
Edit Shipping Information
Edit Payment Information
Order History
Delete Account

Order:
Order History
Place Order
Cancel Order
View Order

Store:

Adds items to the inventory

Removes items from the inventory

Logout:

Exit program: Logout: logs user out of their account

Logout message: “Thank you for shopping at our store!”

7. Information Storage

We will be storing the information in a database. In this particular case we will be using SQLite3 which is an open-source relational database management system (RDBMS). We will have a total of nine database tables, listed below:

Customer
- ***Customer ID *****
- Email
- Password
- First Name
- Last Name
- Phone

Payment Info
- ***Payment ID *****
- Card Number
- CVV
- Address Line 1
- Address Line 2
- City
- State
- Zip
- *User ID **

Shipping Info
- ***Shipping ID *****
- Address Line 1
- Address Line 2
- City
- State
- Zip
- *User ID **

Products
- ***Product ID *****
- Category
- Price
- Stock

Books
- ***ISBN *****
- Title
- Author

  - Year
  - Publisher
  - Genre
  - *Product ID* *
- Games
  - ***Title*** **
  - Publisher
  - Year
  - Genre
  - Platform
  - *Product ID* *
- Carts
  - ***Cart Item ID*** **
  - Quantity
  - *User ID* *
  - *Product ID* *
- Orders
  - ***Order Item ID*** **
  - Order ID
  - Total
  - Order Date
  - Arrival Date
  - *User ID* *
  - *Product ID* *
- Store
  - ***Store Name*** **
  - Address Line 1
  - Address Line 2
  - Phone
  - Email