Java E-Commerce Console Application

This project « 9offa.tn »is a simplified demonstration of an e-commerce application developed in Java, operating through the command line (console). It has been designed to showcase fundamental concepts of object-oriented programming, data manipulation, and basic operations of an e-commerce system.

Mohammed Achref Hemissi

1.User Authentication:

- Users (customer, admin) can log in , and log out.
- A customer can register.
- Role-based access control implemented (customer, admin).
- Access restrictions based on user roles.
- Customers have the ability to manage their account information conveniently. By selecting "3. Visit my account," customers can access options to modify their credit card details, change their password, or opt to remove their account entirely. This functionality provides customers with control over their account settings, ensuring a seamless and personalized user experience.

2.Product Management

- I have implemented an abstract class named "Product" from which various categories inherit their specific functionalities. The features encompass adding (a method in the Crud Class), viewing, updating, and deleting products (both methods are within the Product class itself).

- Distinct user types possess their unique methods for viewing products (both methods are within the Article class). Additionally, only the admin user has the privilege to delete and add products to the system.

- It's important to note that the Product class doesn't directly represent an article; instead, it acts as a base class providing foundational functionality for various categories of articles to inherit and extend upon.

3.The Basket

- Customers have the ability to view, add, update, and remove items in their shopping basket. To manage their basket, customers can follow these steps:

1. Choose "3. Visit my account" and then select "1. Manage your Basket.“

2. From there, they can view their current orders and choose to update or delete each order individually.

- To address this issue, I implemented the method "UpdateRemoveBasket()" in the Customer class. This method allows customers to modify or delete items within their basket, providing them with greater control and flexibility over their orders.

4. Order Processing

- To enhance the transition from the cart to order completion, users can select "4. Complete my orders." Following this selection, users will be prompted to input their credit card's CVV. The system will then verify the availability of sufficient funds and the quantity of items in stock. If both conditions are met, the purchase will be processed successfully. However, if there's an insufficiency in funds or insufficient stock, an error message will be displayed to notify the user about the issue.
- Once the purchase is completed, the order details will be stored in the system. Customers can view their past orders by selecting "3. Visit my account" followed by "5. Your Previous Orders." These details will be stored and organized within an Order class, providing customers with easy access to their order history for reference purposes.

5. Inventory Management

- Automatic inventory updates post-purchase and updating of information .
- The admin can see the articles that their stock is under or equal to 5 .

7. Dynamic Search and Filtering

- **Search by Name:**

Both customers and admins can search for products using full or partial names. This feature allows users to find products even with incomplete information.

- **Filtering by Category:**

Users (both customers and admins) can filter products based on their respective categories. This filtering mechanism aids in narrowing down search results for a more targeted product selection.

- To align the project closer to real-world functionalities, I've integrated the ability to purchase a product and add a review during the search process. When searching for a product using any available method, users can select the desired product by its corresponding number. Upon selection, users will be presented with the option to either proceed with purchasing the product or to add a review for it.

- This enhancement provides users with a streamlined experience where they can seamlessly navigate the search process, select a product of interest, and then choose between making a purchase or leaving a review for the selected item.

8.Payment Processing:

- Basic simulation of payment transactions.
- Supports checkout and order confirmation.

9.User Reviews and Ratings

- Certainly! Users have the capability to leave reviews, comprising both comments and ratings, for the products they have purchased or interacted with. Additionally, the system provides a feature that aggregates these individual ratings to generate an overall rating for each product. This aggregated rating is then displayed alongside the product information, allowing other users to quickly gauge the collective feedback and opinions regarding a specific product. Overall, this functionality enhances the user experience by providing valuable insights and feedback from the community about various products within the system.

10.Product Recommendations

To enhance user experience, the system analyzes the user's browsing behavior and history to offer personalized product recommendations. It utilizes this information to suggest relevant products aligned with the user's preferences.

Moreover, the system's main page displays the last 5 products that the customer has browsed. This feature provides quick access to recently viewed items, allowing users to easily revisit or consider these products during their current browsing session. This personalized approach aims to assist users in discovering products of interest based on their recent activity within the system.

Contribution and Feedback

- We welcome you to explore the codebase and engage with the application. Your feedback, suggestions, and contributions are highly valued to improve the functionality and usability of this project.