Silk Road Online Store
Deadline – 6

User Guide

The silk road online store is an online store that is developed by Ankit Gautam (2021518) and Anant Bhaskar (2021516) as a part of the final project for the course “Database Management Systems” at Indraprastha Institute of Information Technology during the winter semester 2023.

The online store is made with MySQL and utilises various functionalities. Python-CLI has been chosen as the interface. To browse through the project kindly find the “main.py” for the project and run it using the appropriate commands.

Functionalities of user:

1. Login or Register: A user can either choose to login or register on the online store. However, in order to use the store, the user must have either a existing account or should create a new one. The platform supports login for admin and user. Admin accounts cannot be created and already exists in the database.
2. Main Page: Upon successful login, user is greeted with various options for the online store. The user can choose to browse through the products, view cart, check his order status, view his profile or logout.
3. View products: The user can browse through a list of products that belong to different categories. If the user likes a particular product, they can add it to their cart by providing the product id and the quantity. Before adding a particular product to the cart of the user. Our code checks if there is enough quantity available for the product. If the quantity is not available. The user is asserted for the same.
4. View cart: The user can see what their cart currently holds. They can see the total amount and number of items in their cart.
5. Order status: The user can view the status of their currently placed orders and keep a track of their delivery. Payment details, Delivery partner name etc are available for the reference of customers after placing the order. Since, there was no way to provide tracking link because this is not deployed yet and is not associated with any real-life entities. We have assumed that customers will not need any such details, just the name of the delivery partner is enough for them.
6. View profile: The user can view and update various details such as email, password, address, phone number etc.

Functionalities of Admin:

1. View product sales: The admin can view sales done by a particular product. This would be beneficial in determining how well a particular product is doing on the market and provide information regarding how much quantity will maximise the profit, this can be achieved by a deep analysis of all the products that are currently listed on the online store.
2. View category sales: The admin can view sales done by a particular category. This would be beneficial in providing insights on how well a particular category is doing and can further contribute to increased brand collaborations leading to increases revenue.
3. View customer details: The customer can view the details for a particular customer.
4. Add a category: The admin can add as many categories as they want. This can be used to scale up the inventory for our online store.
5. Add a product: The admin can add as many products required. This can be used to scale up the inventory for our online store.
6. Add new delivery partner: The admin can add new delivery partners. Since, we have assumed that one delivery partner will take only one order. Therefore, it is very important to add more delivery partners.

Transactions

Conflicting Transactions:

1. **There are multiple customers trying to add a product with available quantity 1 into their respective carts.**

Transaction 1:

- Customer 1 tries to add the product in his cart.
- Our interface checks the available quantity of the product.
- Since, the quantity is available. It processes the quantity for customer 1 and updates the available quantity in the database.
- Product is added for customer 1 and now the quantity is zero for the product.

Transaction 2:

- Customer 2 tries to add the product in his cart.
- Our interface checks the available quantity of the product.
- Product is added for customer 2 and now the quantity is zero for the product.
- Product is added for customer 2 and now the quantity is zero for the product.

Assuming that these both transactions happen at the same time, when the transaction happens till step 2 both the customers are displayed the available quantity as 1. This transaction is likely to cause a conflict as only one of the two customers will get the demanded product. If transaction 1 executes before transaction 2, customer 1 will get the item on the other hand if transaction 2 executes before transaction 1, customer 2 will get the item.

2. **The admin terminates the account of a customer for some reason, while the customer is trying to update his or her details such as phone number or address.**

Transaction 1:

- A customer logs into his account.
- Customers goes to “view my profile” section and then tries to change a detail such as his “Phone number” or “address.”

Transaction 2:

- Admin wants to delete the account of the customer for some reason.
- Admin views the details of the customer and proceeds to delete the account.

Assuming that the customer is same in both transaction 1 and transaction 2, we can say that a conflict is likely to happen in this case. If transaction 1 executes before transaction 2, then the customer’s detail will get updated and the account with updated details will get deleted. This can also be interpreted as a non-conflict serializable schedule. However, if transaction 2 executes before transaction 1, then the customer’s detail will not be updated and all transactions will fail for that customer. This can be interpreted as a conflict serializable schedule. To resolve this conflict, we need to ensure that these transactions are executed serially. Serial transactions are the transactions that are executed in a particular sequence. MySQL uses a technique called locking in order to achieve this. Therefore, locking can be implemented to avoid such conflicts.

Non-conflicting Techniques:

1. **Updating total number of items and total amount in cart.**

The user adds multiple products to his cart and the system automatically performs a transaction that updates the total items and total amount in the cart. Following function has been achieved and implemented in our frontend application. There are no conflicts here because the cart belongs to the user itself and every user has a private instance of their cart to themselves.

2. **Placing an order for all the items in the cart and clearing the cart after placing the order.**

The user has added multiple products to his cart and now wants to place the order for all of them. He just needs to proceed towards the checkout option that has been placed in the “view my cart” option. After he chooses to checkout for all the items in the cart an order of all the items is placed and the cart is cleared for the user.

3. **Assigning a delivery partner for an order.**

After a order has been placed, order is now kept in the queue to assign to a delivery partner for the delivery processing. Since, our database assumes that one delivery partner will deliver only one order therefore, the database waits till there is a delivery partner available. Once it is available, it assigns a delivery partner for the order. Information for the delivery partner can be viewed by a user under “order status” section.

4. **Processing payment for the orders.**

A crucial step to placing order is making the payment for the same. We have simulated a dummy payment gateway that simulates the payment process of an online store. A user can specify the payment method and details for the same and proceed towards the order.