

Unit testing of checkout_and_payment

Checkout_and_payment functions similarly to a “game loop” which continuously prompts the user to make choices through input characters in the terminal. The main goals of testing this function is to ensure that user interactions do not crash the program, regardless of unexpected input, and to maintain consistent, predictable behaviour so the user always understands the state of the application. In other words, these tests are made to validate the stability of the function and to discover any eventual shortcomings.

The checkout_and_payment function only has one valid input type, a user object represented on the form {"username": __name__, "wallet": __balance__}. So since none of {int, float, string, list} is a valid input, 4 tests were created to make sure appropriate error handling exists, if the function ever were to receive one of these input types. That is only the parametric input though. The function operates based on the contents of these four data structures:

1. User Wallet Balance (from user argument)
2. Shopping Cart Contents (Shopping Cart Object is invoked when the function starts)
3. Product Inventory (A csv file containing a heap of different products, mocked as a smaller list in the test cases)
4. User Interactions ('d' for display products, 'c' for check cart etc)

These are divided into four equivalence classes (ECx) with different test scenarios as follows.

1. User Wallet Balance (EC1)

EC1.1 `user.wallet < cart.get_total_value()`, user has insufficient funds to perform the checkout

EC1.2 `user.wallet == cart.get_total_value()`, user has exactly the funds needed to perform the checkout

EC1.3 `user.wallet > cart.get_total_value()`, user has more funds than the total cost of the cart and can perform the checkout

EC1.4 user.wallet edge cases such as negative funds, extremely high funds(10^{50}), or extreme low funds 0.00000001

2. Shopping Cart Contents (EC2)

EC2.1 `cart.is_empty() == True`, cart is empty

EC2.2 `cart.is_empty() == False`, cart has items

3. Product Inventory (EC3)

EC3.1 Product is in stock

EC3.2 Product is out of stock

EC3.3 Product doesn't exist in the system

4. User Interactions

EC4.1 'd', displays available products

EC4.2 'c', displays your cart and prompts either for a product removal or checkout

EC4.3 'y', yes to a prompt

EC4.4 'n', no to a prompt

EC4.5 'l', attempts to log out

EC4.6 remove action is done by the sequence 'c' 'n' (no to checkout) 'y' (yes to remove) 'product number'