**Test Function Document**

|  |  |
| --- | --- |
| Project Name: Assignment 1 Testing | |
| Automation Title: test\_check\_cart | Version: 1.0 |
| Testing Phase: Phase 1 | Date of Test: 16 November 2023 |
| Module Name: checkout\_and\_payment.py | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function Title: test\_check\_cart | | | | Test Designed by: Edvin Eriksson | | | |
| Test Priority (Low/Medium/High): Medium | | | | Test Designed Date: 16 November 2023 | | | |
| Description: The objective of the check\_cart function is for when a user wants to see all the products that are in the cart, but also given the option to checkout if desired. | | | | Test Executed by: Edvin Eriksson | | | |
| Test Execution date: 16 November 2023 | | | |
|  | | | | | | | |
| Pre-conditions: None | | | | | | | |
| Dependencies: checkout, User, Product, ShoppingCart | | | | | | | |
|  | | | | | | | |
| S. No | Equivalence Class | Test case data | Expected Results | | Actual Results | Status (Pass/Fail) | Notes |
| 1 | VEC1 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=3)]  User(name="Kim", wallet='100')  new\_cart.add\_item(product\_list[0])  user\_input “n” | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | Pass | Test name: test\_EC1()  Tests when a product in cart, but with no checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 2 | VEC2 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=3)]  User(name="Kim", wallet='100')  new\_cart.add\_item(product\_list[0])  user\_input “y” | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | | result = check\_cart(user, new\_cart)  result == None  checkout\_stub called once. | Pass | Test name: test\_EC2()  Tests when a product in cart with checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 3 | VEC3 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=3), Product(name='Apple', price='2', units='10')]  User(name='Kim', wallet=100)  new\_cart.add\_item(product\_list[0])  new\_cart.add\_item(product\_list[1])  user\_input “y” | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | Pass | Test name: test\_EC3  Tests when multiple products in cart with checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 4 | VEC4 | checkout\_stub,  User(name='Kim', wallet=100)  user\_input “y” | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | Pass | Test name: test\_EC4()  Tests when empty cart with checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 5 | VEC5 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=3), Product(name='Apple', price='2', units='10')]  User(name='Kim', wallet=100)  new\_cart.add\_item(product\_list[0])  new\_cart.add\_item(product\_list[1])  user\_input “n” | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | Pass | Test name: test\_EC5()  Tests when multiple products in cart, but with no checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 6 | VEC6 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=3)]  User(name='Kim', wallet=100)  new\_cart.add\_item(product\_list[0])  user\_input “t” | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | Pass | Test name: test\_EC6()  Test with product in cart, but wrong input to checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 7 | VEC7 | checkout\_stub,  User(name='Kim', wallet=100)  user\_input “n” | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | Pass | Test name: test\_EC7()  Tests when empty cart, but with no checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 8 | VEC8 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=1)]  User(name='Kim', wallet=0)  new\_cart.add\_item(product\_list[0])  user\_input “n” | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | | result = check\_cart(user, new\_cart)  result == False  checkout\_stub not called. | Pass | Test name: test\_EC8()  Tests with insufficient wallet, with product in cart but no checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 9 | VEC9 | checkout\_stub,  product\_list = [Product(name='Orange', price=-10, units=3)]  User(name='Kim', wallet=100)  new\_cart.add\_item(product\_list[0])  user\_input “y” | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | Pass | Test name: test\_EC9()  Tests with product in cart with negative price with checkout  Uses a fixture new\_cart that is a ShoppingCart(). |
| 10 | VEC10 | checkout\_stub,  product\_list = [Product(name='Orange', price=10, units=-1)]  User(name='Kim', wallet=100)  new\_cart.add\_item(product\_list[0])  user\_input “y” | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | | result = check\_cart(user, new\_cart)  assert result == None  checkout\_stub called once. | Pass | Test name: test\_EC10()  Tests when product in cart with negative units, with checkout  Uses a fixture new\_cart that is a ShoppingCart(). |