**Test Function Document**

|  |  |
| --- | --- |
| Project Name: Online Shopping Cart | |
| Automation Title: test\_checkout\_and\_payment | Version: 1.0 |
| Testing Phase: Phase 1 | Date of Test: 27 Nov 2024 |
| Module Name: Checkout process | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function Title: checkout\_and\_payment | | | | | Test Designed by: Zhiqi | | |
| Test Priority (Low/Medium/High): High | | | | | Test Designed Date: 27 Nov 2024 | | |
| Description: The checkout\_and\_payment function takes a disctionary object as parameter and then print all products, add product to cart, check out or log out based on user inputs. | | | | | Test Executed by: Zhiqi | | |
| Test Execution date: 27 Nov 2024 | | |
|  | | | | | | | |
| Pre-conditions:None | | | | | | | |
| Dependencies: get\_products, ShoppingCart, User, UserInterface.get\_user\_input, display\_products\_available\_for\_purchase, check\_cart, open, json.load, json.dump, logout, ShoppingCart.add\_item | | | | | | | |
|  | | | | | | | |
| S. No | Equivalence Class | Test case data | Expected Results | Actual Results | | Status (Pass/Fail) | Notes |
| 1 | Log out | User input of ‘l’.  .  .  . | Stub logout function is called once and the function terminates. | logout function is called once and the function terminates. | | Pass | Test name: test\_checkout\_and\_payment1 |
| 2 | Not logging out | User input of ‘l’, ‘l’.  .  .  . | Stub logout function is called twice. | logout function is called twice. | | Pass | The logout function returns False the first time and True the second time.  Test name: test\_checkout\_and\_payment2 |
| 3 | Display all of the products | User input of ‘d’, ‘l’ | Stub display\_products\_available\_for\_purchase function is called once. | display\_products\_available\_for\_purchase function is called once. | | Pass | Test name: test\_checkout\_and\_payment3 |
| 4 | Line break symbol in the input does not affect subsequent input | User input of ‘d\nc’, ‘l’ | Stub display\_products\_available\_for\_purchase function and logout function both are called once. | display\_products\_available\_for\_purchase function and logout function both are called once. | | Pass | Test name: test\_checkout\_and\_payment4 |
| 5 | Not checking out | User input of ‘c’, ‘l’ | Stub check\_cart is called once and mock\_open function is not called. | Stub check\_cart is called once and mock\_open function is not called. | | Pass | Stub check\_cart function returns false.  Test name: test\_checkout\_and\_payment5 |
| 6 | Normal check out | User input of ‘c’, ‘l’  Mock user.json of [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 150.0}]  Stub login\_info of {'username': 'Samantha', 'wallet': 148.0} | Mock\_dump is called once with [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 148.0}], mock\_file.return\_value, indent=2.  Stub check\_cart is called once and mock load is called once. | Mock\_dump is called once with [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 148.0}], mock\_file.return\_value, indent=2.  Stub check\_cart is called once and mock load is called once. | | Pass | Stub check\_cart function returns True.  Test name: test\_checkout\_and\_payment6 |
| 7 | Check out with 0 in wallet | User input of ‘c’, ‘l’  Mock user.json of [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 150.0}]  Stub login\_info of {'username': 'Samantha', 'wallet': 0.0} | Mock\_dump is called once with [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 0.0}], mock\_file.return\_value, indent=2.  Stub check\_cart is called once and mock load is called once. | Mock\_dump is called once with [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 0.0}], mock\_file.return\_value, indent=2.  Stub check\_cart is called once and mock load is called once. | | Pass | Stub check\_cart function returns True.  Test name: test\_checkout\_and\_payment7 |
| 8 | Check out with 0.2 in wallet | User input of ‘c’, ‘l’  Mock user.json of [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 150.0}]  Stub login\_info of {'username': 'Samantha', 'wallet': 0.2} | Mock\_dump is called once with [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 0.2}], mock\_file.return\_value, indent=2.  Stub check\_cart is called once and mock load is called once. | Mock\_dump is called once with [{"username": "Ramanathan","password": "Notaproblem23\*","wallet": 100.0}, {"username": "Samantha","password": "SecurePass123/^","wallet": 0.2}], mock\_file.return\_value, indent=2.  Stub check\_cart is called once and mock load is called once. | | Pass | Stub check\_cart function returns True. Test name: test\_checkout\_and\_payment8 |
| 9 | Invalid user input with letters | User input of ‘g\tsdf’, ‘l’ | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment9 |
| 10 | Invalid user input with float | User input of ’23.65’, ‘l’ | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment10 |
| 11 | Invalid user input with base 16 number | User input of ‘\x01’, ‘l’ | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment11 |
| 12 | Invalid user input with blank spaces | User input of ‘  ’, ‘l’ | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment12 |
| 13 | Invalid user input with list | User input of ‘[c]’, ‘l’ | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment13 |
| 14 | Product number that is larger than product list length | User input of ‘525’, ‘l’  Mock global\_products with [Product('Tomato', 1, 15)] | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment14 |
| 15 | Product number that is smaller than 1 | User input of ‘0’, ‘l’  Mock global\_products with [Product('Tomato', 1, 15)] | 'Invalid input. Please try again.\n' is printed. | 'Invalid input. Please try again.\n' is printed. | | Pass | Test name: test\_checkout\_and\_payment15 |
| 16 | Add one item to cart | User input of ‘1’, ‘l’  Mock global\_products with [Product('Tomato', 1, 15)] | Stub cart\_stub.add\_item is called once and ‘Tomato added to your cart.\n’ is printed | Stub cart\_stub.add\_item is called once and ‘Tomato added to your cart.\n’ is printed | | Pass | Test name: test\_checkout\_and\_payment16 |
| 17 | Add multiple of the same item of cart | User input of ‘1’, ‘1’, ‘1’, ‘l’  Mock global\_products with [Product('Tomato', 1, 15)] | Stub cart\_stub.add\_item is called three times and ‘Tomato added to your cart.\nTomato added to your cart.\nTomato added to your cart.\n’ is printed | Stub cart\_stub.add\_item is called three times and ‘Tomato added to your cart.\nTomato added to your cart.\nTomato added to your cart.\n’ is printed | | Pass | Test name: test\_checkout\_and\_payment17 |
| 18 | Add different items to cart | User input of ‘1’, ‘1’, ‘2’, ‘l’  Mock global\_products with [Product('Tomato', 1, 15), product2 = Product('Onion', 0.8, 20)] | Stub cart\_stub.add\_item is called three times and ‘Tomato added to your cart.\nTomato added to your cart.\nOnion added to your cart.\n’ is printed | Stub cart\_stub.add\_item is called three times and ‘Tomato added to your cart.\nTomato added to your cart.\nOnion added to your cart.\n’ is printed | | Pass | Test name: test\_checkout\_and\_payment18 |
| 19 | Add items to cart when it is out of stock | User input of ‘1’, ‘1’, ‘1’, ‘l’  Mock global\_products with [Product('Tomato', 1, 2)] | Stub cart\_stub.add\_item is called twice and ‘Tomato added to your cart.\nTomato added to your cart.\nSorry, Tomato is out of stock.\n’ is printed | Stub cart\_stub.add\_item is called twice and ‘Tomato added to your cart.\nTomato added to your cart.\nSorry, Tomato is out of stock.\n’ is printed | | Pass | Test name: test\_checkout\_and\_payment19 |
| 20 | Add item to cart when the product list is empty | User input of ‘1’, ‘l’  Mock global product with [] | Stub cart\_stub.add\_item is not called and ‘Invalid input. Please try again.\n’ is printed | Stub cart\_stub.add\_item is not called and ‘Invalid input. Please try again.\n’ is printed | | Pass | Test name: test\_checkout\_and\_payment20 |