**Test Function Document**

|  |  |
| --- | --- |
| Project Name: Assignment 1 | |
| Automation Title: test\_checkout\_and\_payment | Version: 1.0 |
| Testing Phase: Phase 1 | Date of Test: 2023-11-23 |
| Module Name: checkout\_and\_payment.py | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function Title: checkoutAndPayment | | | | | Test Designed by: Sara Mosebach | | |
| Test Priority (Low/Medium/High): Medium | | | | | Test Designed Date: 2023-11-23 | | |
| Description: The objective of the checkoutAndPayment function is to manage the shopping, checkout and logout processes for the main program. The function initially prints out all available products and allows the user to select between three options: to check the cart), to logout or to add an item to the cart. Depending on the selected option, different operations is made and appropriate functions are called. | | | | | Test Executed by: Sara Mosebach | | |
| Test Execution date: 2023-11-23 | | |
|  | | | | | | | |
| Pre-conditions: None | | | | | | | |
| Dependencies: Yes, the functions check\_cart and logout | | | | | | | |
|  | | | | | | | |
| S. No | EC | Test case data | Expected Results | Actual Results | | Status (Pass/Fail) | Notes |
| 1 | EC1  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2)]  Input: ['l'] | Function prints:  '1. Ice cream - $10.0 - Units: 2' | Function prints:  '1. Ice cream - $10.0 - Units: 2' | | Pass | Test name: test\_printout\_one\_product  Stub functions:  logout\_stub1  Testing for a correctly written output when given one product |
| 2 | EC1/  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2), Product('Chocolate', 15, 5), Product('Popcorns', 8, 3) ]  Input: ['l'] | Function prints:  '1. Ice cream - $10.0 - Units: 2\n2. Chocolate - $15.0 - Units: 5\n3. Popcorns - $8.0 - Units: 3' | Function prints:  '1. Ice cream - $10.0 - Units: 2\n2. Chocolate - $15.0 - Units: 5\n3. Popcorns - $8.0 - Units: 3' | | Pass | Test name:  test\_printout\_multiple\_products  Stub functions:  logout\_stub1  Testing for a correctly written output when given multiple products |
| 3 | EC2  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['c', 'l'] | cart.retrieve\_item() == [[]]  (Cart contains no items) | cart.retrieve\_item() == [[]]  (Cart contains no items) | | Pass | Test name:  test\_check\_cart\_empty\_cart  Stub functions:  logout\_stub1  check\_cart\_stub1  Testing for an empty cart when choosing option c. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | EC2/  EC4 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2)]  Input: ['1', 'c', 'l'] | cart.retrieve\_item().get\_product() ==  [['Ice cream', 10.0, 2]] | cart.retrieve\_item().get\_product() ==  [['Ice cream', 10.0, 2]] | Pass | Test name: test\_check\_cart\_nonempty\_cart  Stub functions:  logout\_stub1  check\_cart\_stub2  Testing for correct items in the cart after adding an item and choosing option c. |
| 5 | EC2  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['c', 'l'] | User.name == 'Ramanathan'  User.wallet == 100 | User.name == 'Ramanathan'  User.wallet == 100 | Pass | Test name: test\_check\_cart\_user\_info  Stub functions:  logout\_stub1  check\_cart\_stub3  Testing for correct user information when choosing option c. |
| 6 | EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['l'] | Function prints:  'You have been logged out' | Function prints:  'You have been logged out' | Pass | Test name:  test\_printout\_logout\_confirmed  Stub functions:  logout\_stub1  Testing for a correctly written output when choosing option l. |
| 7 | EC2  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['c', 'l'] | json\_dump\_mock.assert\_called\_once() | json\_dump\_mock.assert\_called\_once() | Pass | Test name:  test\_call \_for\_dump  Stub functions:  logout\_stub1  open\_users\_file\_stub  json\_dump\_mock  check\_cart\_stub4  Testing for a call to the mock function json\_dump\_mock, simulating a call for the json.dump function. |
| 8 | EC2  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['c', 'l'] | json\_dump\_mock.assert\_called\_once\_with([{"username": "Ramanathan", "password": "Notaproblem23\*", "wallet": 100}], mock.ANY) | json\_dump\_mock.assert\_called\_once\_with([{"username": "Ramanathan", "password": "Notaproblem23\*", "wallet": 100}], mock.ANY) | Pass | Test name:  test\_nonupdated\_wallet\_after\_logout  Stub functions:  logout\_stub1  open\_users\_file\_stub  json\_dump\_mock  check\_cart\_stub4  Testing for a correct call to the mock function json\_dump\_mock, simulating an update call for the json.dump function. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | EC2  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2)]  Input: ['1', 'c', 'l'] | json\_dump\_mock.assert\_called\_once\_with([{"username": "Ramanathan", "password": "Notaproblem23\*", "wallet": 90}], mock.ANY) | json\_dump\_mock.assert\_called\_once\_with([{"username": "Ramanathan", "password": "Notaproblem23\*", "wallet": 90}], mock.ANY) | Pass | Test name:  test\_updated\_wallet\_one\_purcase  Stub functions:  logout\_stub1  open\_users\_file\_stub  json\_dump\_mock  check\_cart\_stub4  Testing for a correct call to the mock function json\_dump\_mock, simulating an update call for the json.dump function after a purcase. |
| 10 | EC2  /  EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2), Product('Chocolate', 15, 5), Product('Popcorns', 8, 3)]  Input: ['1', '2', '3', 'c', 'l'] | json\_dump\_mock.assert\_called\_once\_with([{"username": "Ramanathan", "password": "Notaproblem23\*", "wallet": 67}], mock.ANY) | json\_dump\_mock.assert\_called\_once\_with([{"username": "Ramanathan", "password": "Notaproblem23\*", "wallet": 67}], mock.ANY) | Pass | Test name:  test\_updated\_wallet\_multiple\_purcases  Stub functions:  logout\_stub1  open\_users\_file\_stub  json\_dump\_mock  check\_cart\_stub4  Testing for a correct call to the mock function json\_dump\_mock, simulating an update call for the json.dump function after multiple purcases. |
| 11 | EC3 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['l'] | Function prints:  'You have been logged out'  +  cart.retrieve\_item() == [[]] | Function prints:  'You have been logged out'  +  cart.retrieve\_item() == [[]] | Pass | Test name:  test\_logout\_empty\_cart  Stub functions:  logout\_stub2  Testing for an empty cart and correct output when choosing option l. |
| 12 | EC3  /  EC4 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2)]  Input: ['1', 'l', 'y'] | Function prints:  'You have been logged out'  +  cart.retrieve\_item().get\_product() == [['Ice cream', 10.0, 2]] | Function prints:  'You have been logged out'  +  cart.retrieve\_item().get\_product() == [['Ice cream', 10.0, 2]] | Pass | Test name:  test\_logout\_nonempty\_cart  Stub functions:  logout\_stub3  Testing for correct items in the cart after adding an item and correct output when choosing option l. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13 | EC3  /  EC4 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 2)]  Input: ['1', 'l', 'y'] | Function prints:  'Ice cream added to your cart' | Function prints:  'Ice cream added to your cart' | Pass | Test name:  test\_printout\_add\_item\_to\_cart  Stub functions:  logout\_stub1  Testing for a correct output after adding an item. |
| 14 | EC3  /  EC4 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: [Product('Ice cream', 10, 0)]  Input: ['1', 'l', 'y'] | Function prints:  'Sorry, Ice cream is out of stock.' | Function prints:  'Sorry, Ice cream is out of stock.' | Pass | Test name:  test\_printout\_add\_item\_to\_cart  Stub functions:  logout\_stub1  Testing for a correct output after adding an item that is out of stock. |
| 15 | EC3  /  EC5 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['a', 'l'] | Function prints:  'Invalid input, please try again.' | Function prints:  'Invalid input, please try again.' | Pass | Test name:  test\_printout\_choice\_other\_letter  Stub functions:  logout\_stub1  Testing for a correct output after selecting nonexistent letter as choice. |
| 16 | EC3  /  EC5 | User: {'username': 'Ramanathan', 'wallet': 100}  List of products: []  Input: ['5', 'l'] | Function prints:  'Invalid input, please try again.' | Function prints:  'Invalid input, please try again.' | Pass | Test name:  test\_printout\_choice\_other\_letter  Stub functions:  logout\_stub1  Testing for a correct output after selecting nonexistent letter as choice. |