**Test Function Document**

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function Title: test\_checkout | | | | Test Designed by: Edvin Eriksson | | | |
| Test Priority (Low/Medium/High): Medium | | | | Test Designed Date: 16 November 2023 | | | |
| Description: The objective of the checkout function is for when a user wants to checkout the products that are in the cart, resulting in the total amount of the products in the cart to be deducted from the user’s wallet. It also is important that the user gets notified of what happened with the checkout procedure given different scenarios. The units of the affected products should also be changed accordingly. | | | | Test Executed by: Edvin Eriksson | | | |
| Test Execution date: 16 November 2023 | | | |
|  | | | | | | | |
| Pre-conditions: None | | | | | | | |
| Dependencies: User, Product, ShoppingCart | | | | | | | |
|  | | | | | | | |
| S. No | Equivalence Class | Test case data | Expected Results | | Actual Results | Status (Pass/Fail) | Notes |
| 1 | VEC1 | An empty list of products  A user named Kim with a wallet worth 20.  An empty shopping cart. | Print out "Your basket is empty. Please add items before checking out." in the terminal  User’s wallet is still worth 20.  There are no items in the cart.  The list of products is empty. | | Print out "Your basket is empty. Please add items before checking out." in the terminal  User’s wallet is still worth 20.  There are no items in the cart.  The list of products is empty | Pass | Test name: test\_EC1()  Tests with an empty cart.  Uses a fixture new\_cart that is a ShoppingCart(). |
| 2 | VEC2 | A list of products with one product being an orange, priced at 10 and having 3 units in total.  A user named Kim with a wallet worth 5.  A shopping cart with one orange in it. | Print out " You don't have enough money to complete the purchase. Please try again!" in the terminal  User’s wallet is still worth 5.  There is one product in the cart.  The list of products still contains one product. | | Print out " You don't have enough money to complete the purchase. Please try again!" in the terminal  User’s wallet is still worth 5.  There is one product in the cart.  The list of products still contains one product. | Pass | Test name: test\_EC2()  Tests with insufficient wallet funds.  Uses a fixture new\_cart that is a ShoppingCart(). |
| 3 | VEC3 | A list of products with one product being an orange, priced at 10 and having 3 units in total.  A user named Kim with a wallet worth 100.  A shopping cart with one orange in it. | Print out " Thank you for your purchase, Kim! Your remaining balance is 90"” in the terminal.  User’s wallet is worth 90.  There are no items in the cart.  The list of products contains one product with 2 units left. | | Print out " Thank you for your purchase, Kim! Your remaining balance is 90"” in the terminal.  User’s wallet is worth 90.  There are no items in the cart.  The list of products contains one product with 2 units left. | Pass | Test name: test\_EC3  Tests with sufficient wallet funds and a product in cart  Uses a fixture new\_cart that is a ShoppingCart(). |
| 4 | VEC4 | A list of products with two products: an orange, priced at 10 and having 3 units in total, and an apple priced at 20 having 2 units in total.  A user named Kim with a wallet worth 100.  A shopping cart with one orange and one apple in it. | Print out " Thank you for your purchase, Kim! Your remaining balance is 70"” in the terminal.  User’s wallet is worth 70.  There are no items in the cart.  The list of products contains two products with 2 respectively 1 units/unit left. | | Print out " Thank you for your purchase, Kim! Your remaining balance is 70"” in the terminal.  User’s wallet is worth 70.  There are no items in the cart.  The list of products contains two products with 2 respectively 1 units/unit left. | Pass | Test name: test\_EC4()  Tests with sufficient wallet funds and multiple products in cart  Uses a fixture new\_cart that is a ShoppingCart(). |
| 5 | VEC5 | A list of products with one product being an orange, priced at 10 and having 1 unit in total.  A user named Kim with a wallet worth 100.  A shopping cart with two oranges in it. | User’s wallet is worth 80.  There are no items in the cart.  The list of products is empty. | | User’s wallet is worth 80.  There are no items in the cart.  The list of products is empty. | Pass | Test name: test\_EC5()  Tests with not enough available units than requested by the user  Uses a fixture new\_cart that is a ShoppingCart(). |
| 6 | VEC6 | A list of products with one product being an orange, priced at 10 and having 3 units in total.  A user named Kim with a wallet worth -100.  A shopping cart with one orange in it. | Print out " You don't have enough money to complete the purchase. Please try again!" in the terminal  User’s wallet is worth -100.  There is one product in the cart.  The list of products still contains one product with three units. | | Print out " You don't have enough money to complete the purchase. Please try again!" in the terminal  User’s wallet is worth -100.  There is one product in the cart.  The list of products still contains one product with three units. | Pass | Test name: test\_EC6()  Tests with negative wallet funds  Uses a fixture new\_cart that is a ShoppingCart(). |
| 7 | VEC7 | A list of products with one product being an orange, priced at -10 and having 3 units in total.  A user named Kim with a wallet worth 100.  A shopping cart with one orange in it. | Print out " Thank you for your purchase, Kim! Your remaining balance is 110"” in the terminal.  User’s wallet is worth 110.  There are no items in the cart.  The list of products contains one product with two units. | | Print out " Thank you for your purchase, Kim! Your remaining balance is 110"” in the terminal.  User’s wallet is worth 110.  There are no items in the cart.  The list of products contains one product with two units. | Pass | Test name: test\_EC7()  Tests with products in cart with negative price  Uses a fixture new\_cart that is a ShoppingCart(). |
| 8 | VEC8 | A list of products with one product being an orange, priced at 10 and having 1 unit in total.  A user named Kim with a wallet worth 100.  A shopping cart with one orange in it. | User’s wallet is worth 90.  The list of products is empty. | | User’s wallet is worth 90.  The list of products is empty. | Pass | Test name: test\_EC8()  Tests with sufficient funds and a product in cart but with only one unit left  Uses a fixture new\_cart that is a ShoppingCart(). |
| 9 | VEC9 | A list of products with one product being an orange, priced at 10 and having 3 units in total.  A user named Kim with a wallet worth 100.5.  A shopping cart with one orange in it. | User’s wallet is worth 90.5.  There are no items in the cart.  The list of products contains one product with two units. | | User’s wallet is worth 90.5.  There are no items in the cart.  The list of products contains one product with two units. | Pass | Test name: test\_EC9()  Tests with decimal balance in the wallet funds  Uses a fixture new\_cart that is a ShoppingCart(). |
| 10 | VEC10 | A list of products with one product being an orange, priced at 10 and having 0 unit in total.  A user named Kim with a wallet worth 100.  A shopping cart with one orange in it. | User’s wallet is worth 90.  There are no items in the cart.  The list of products contains one product with minus one unit. | | User’s wallet is worth 90.  There are no items in the cart.  The list of products contains one product with minus one unit. | Pass | Test name: test\_EC10()  Tests with product with no units  Uses a fixture new\_cart that is a ShoppingCart(). |