**Test Function Document**

**1)Module : get\_product**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name: Online Shopping Cart | | | | | | | | |
| Automation Title: Get Products Function Testing | | | | Version: 1.0 | | | | |
| Testing Phase: Functional Testing | | | | Date of Test: 28/11/2024 | | | | |
| Module Name: Online Shopping Cart | | | | | | | | |
| Function Title: get\_product | | | | | Test Designed by: Babitha M K | | | | |
| Test Priority (Low/Medium/High): High | | | | | Test Designed Date: 27/11/2024 | | | | |
| Description: The get\_products() function reads a CSV file containing product details (name, price, units) and returns a list of Product objects, handling various input validations and file parsing scenarios. | | | | | Test Executed by: Babitha M K | | | | |
| Test Execution date: 28/11/2024 | | | | |
|  | | | | | | | | | |
| Pre-conditions: 1) The products.csv file must be available.  2) The Product class should be functional to create product objects.  3) The get\_products() function must read and parse the CSV file correctly | | | | | | | | | |
| Dependencies: Product class must be functional to store product information (name, price,units  The mock\_open function must be used to mock file reading for testing. | | | | | | | | | |
|  | | | | | | | | | |
| S. No | Equivalence Class | Test case data | Expected Results | | | Actual Results | Status (Pass/Fail) | Notes | |
| 1 | IEC1 | Invalid input: integer (1) | Raises TypeError | | | TypeError | Pass | Tests invalid input type (integer) for the function | |
| 2 | IEC2 | Invalid input: float (55.8) | Raises TypeError | | | TypeError | Pass | Tests invalid input type (float) for the function. | |
| 3 | IEC3 | Invalid input: string ("abc.csv") | Raises FileNotFoundError | | | |  | | --- | | FileNotFoundError |  |  | | --- | |  | | Pass | Tests invalid input type (string) for file path. | |
| 4 | IEC4 | Invalid input: list (["abc.csv", "def.csv"]) | Raises TypeError | | | TypeError | Pass | Tests invalid input type (list) for the file path. | |
| 5 | VEC1 | Valid file (products.csv) | Returns 71 products with valid data | | | 71 products, valid data | Pass | Tests loading from a valid CSV file and checks product data. | |
| 6 | VEC2 | Empty CSV file | Returns an empty list | | | Empty list | Pass | Tests handling of empty CSV file. | |
| 7 | VEC3 | Malformed CSV (missing columns) | Raises KeyError | | | KeyError | Pass | Tests malformed CSV file with missing columns. | |
| 8 | VEC4 | Malformed CSV (missing data) | Raises KeyError | | | KeyError | Pass | Tests malformed CSV file with missing data (empty cells). | |
| 9 | VEC5 | Invalid data types (e.g., "three" for price) | Raises KeyError | | | KeyError | Pass | Tests handling of invalid data types in CSV (non-numeric). | |
| 10 | VEC6 | Special characters in product names | Loads products with special characters | | | Special characters loaded | Pass | Tests how special characters in product names are handled. | |
| 11 | VEC7 | Whitespace in product names | Strips whitespace from product names | | | Stripped whitespace | Pass | Tests how the function handles leading and trailing whitespace. | |
| 12 | VEC8 | Case-sensitive product names | Loads products with case-sensitive names | | | Case-sensitive names loaded | Pass | Tests case-sensitivity of product names. | |
| 13 | VEC9 | |  | | --- | |  |  |  | | --- | | Extra fields in CSV | | Ignores extra fields and loads valid products | | | Extra fields ignored | Pass | Tests how extra fields in the CSV are handled. | |
| 14 | VEC10 | Valid file with correct data | Returns a list of products with correct attributes | | | Correct attributes loaded | Pass | Tests loading valid products and ensuring correctness in data. | |

**2)Module : checkout**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name: Online Shopping Cart | | | | | | | | |
| Automation Title: Checkout function testing | | | | Version: 1.0 | | | | |
| Testing Phase: Functional Testing | | | | Date of Test: 28/11/2024 | | | | |
| Module Name: Online Shopping Cart | | | | | | | | |
| Function Title: checkout() | | | | | Test Designed by: Babitha M K | | | | |
| Test Priority (Low/Medium/High): High | | | | | Test Designed Date: 27/11/2024 | | | | |
| Description: The checkout function processes a user's shopping cart by calculating the total cost of the selected products and updating the user's wallet balance. If the user has sufficient funds, the cart is cleared, otherwise, the transaction is aborted, and the cart remains intact. | | | | | Test Executed by: Babitha M K | | | | |
| Test Execution date: 28/11/2024 | | | | |
|  | | | | | | | | | |
| Pre-conditions: 1) The user and shopping cart classes must be functional and initialized.  2) The checkout function should process a valid cart and user.  3) The Product class must be functional to store product information (name, price, units).  4) The user wallet balance should be initialized.  5) The checkout function must reduce the wallet balance by the total cost of products in the cart if there is sufficient balance.  6 The cart must be emptied after a successful checkout. | | | | | | | | | |
| Dependencies: 1) Product class: Must create product objects with name, price, and units.  2) ShoppingCart class: Handles adding products, calculating total cost, and checking if the cart is empty.  3) User class: Manages wallet balance and ensures the user has enough funds for checkout.  4) File handling (if used): Mocking or reading data from files, such as products or user details, for testing. | | | | | | | | | |
|  | | | | | | | | | |
| S. No | Equivalence Class | Test case data | Expected Results | | | Actual Results | Status (Pass/Fail) | Notes | |
| 1 | IEC1 | Invalid input: negative wallet balance (-10.0) | Cart is not cleared, wallet balance remains -10.0 | | | -10.0, Cart not cleared | Pass | Tests handling of negative wallet balance | |
| 2 | IEC2 | Invalid input: zero price and quantity for product | Cart is cleared, wallet balance remains unchanged | | | Wallet remains 100.0, Cart is empty | Pass | Tests zero price and quantity product handling | |
| 3 | IEC3 | Invalid input: empty cart | Wallet remains unchanged, cart is already empty | | | |  | | --- | |  |   Wallet remains 100.0, Cart is empty | Pass | Tests behavior when cart is empty | |
| 4 | VEC1 | |  | | --- | |  |  |  | | --- | | Valid input: user with sufficient balance (100.0) and products in cart | | |  | | --- | |  |  |  | | --- | |  |   Wallet balance decreases by total cost, cart is emptied | | | Wallet becomes 80.0, Cart is empty | Pass | Tests successful checkout with sufficient balance | |
| 5 | VEC2 | Valid input: single product checkout | Wallet decreases correctly, cart is emptied | | | Wallet becomes 85.0, Cart is empty | Pass | Tests checkout for a single product | |
| 6 | VEC3 | Valid input: multiple products in cart with sufficient balance | Wallet decreases by total price, cart is emptied | | | Wallet becomes 115.0, Cart is empty | Pass | Tests checkout with multiple products | |
| 7 | VEC4 | Valid input: wallet balance exactly matching total cost | Wallet balance becomes zero, cart is emptied | | | Wallet becomes 0.0, Cart is empty | Pass | Tests exact wallet balance matching total cost | |
| 8 | VEC5 | Valid input: large quantity of a single product | Wallet decreases accordingly, cart is emptied | | | Wallet becomes 400.0, Cart is empty | Pass | Tests handling of large quantity product | |
| 9 | VEC6 | Valid input: zero wallet balance | Cart is not cleared, wallet remains zero | | | Wallet remains 0.0, Cart is not emptied | Pass | Tests behavior with zero wallet balance | |
| 10 | VEC7 | Insufficient balance with expensive product | Wallet remains unchanged, cart is not cleared | | | Wallet remains 50.0, Cart is not emptied | Pass | Tests insufficient balance with expensive product | |
| 11 | VEC8 | Valid input: exact product price with exact wallet balance | Wallet becomes zero, cart is emptied | | | Wallet becomes 0.0, Cart is empty | Pass | Tests behavior when exact product price matches wallet balance | |
| 12 | VEC9 | |  | | --- | |  |  |  | | --- | |  |   Valid input: zero price product | Wallet remains unchanged, cart is emptied | | | Wallet remains 100.0, Cart is empty | Pass | Tests zero price product handling | |
| 13 | VEC10 | Valid input: multiple products with exact total match to wallet balance | Wallet balance becomes zero, cart is emptied | | | Wallet becomes 100.0, Cart is empty | Pass | Tests multiple products with exact total match | |
| 14 | VEC11 | Valid input: zero wallet balance with expensive product | Wallet remains zero, cart is not emptied | | | Wallet remains 0.0, Cart is not emptied | Pass | Tests checkout with insufficient balance to afford the product | |

|  |  |
| --- | --- |
| Project Name: Online Shopping Cart | |
| Automation Title: Check cart Function Testing | Version: 1.0 |
| Testing Phase: Functional Testing | Date of Test: 28/11/2024 |
| Module Name: Online Shopping Cart | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function Title: check\_cart | | | | Test Designed by: Babitha M K | | | |
| Test Priority (Low/Medium/High): High | | | | Test Designed Date: 27/11/2024 | | | |
| Description: The check\_cart function manages the checkout process for a shopping cart, displaying the contents of the cart and prompting the user to either proceed with checkout or remove an item. It ensures that the user input is validated, and handles edge cases like empty carts, invalid selections. | | | | Test Executed by: Babitha M K | | | |
| Test Execution date: 28/11/2024 | | | |
|  | | | | | | | |
| Pre-conditions: 1) User must be logged in  2) Shopping cart must have products to process | | | | | | | |
| Dependencies: Product class and ShoppingCart class must be functional  UserInterface methods must be available | | | | | | | |
|  | | | | | | | |
| S. No | Equivalence Class | Test case data | Expected Results | | Actual Results | Status (Pass/Fail) | Notes |
| 1 | VEC1 | User with valid balance, cart with valid products | Wallet reduced by total price, cart empty | | Wallet reduced by total price, cart empty | Pass | Valid checkout for user with sufficient balance |
| 2 | VEC2 | Single product, sufficient wallet balance | Wallet reduced by product price, cart empty | | Wallet reduced by product price, cart empty | Pass | Test case with a single product and sufficient balance |
| 3 | VEC3 | Multiple products, sufficient wallet balance | Wallet reduced by total product price, cart empty | | Wallet reduced by total product price, cart empty | Pass | Test case with multiple products and sufficient balance |
| 4 | VEC4 | Wallet balance exactly equal to product price | Wallet reduced to zero, cart empty | | Wallet reduced to zero, cart empty | Pass | Wallet balance matches total cost of products |
| 5 | VEC5 | Large quantity of single product | Wallet reduced by total price for large quantity, cart empty | | Wallet reduced by total price for large quantity, cart empty | Pass | Test case for handling large quantities |
| 6 | VEC6 | Empty cart checkout | Wallet remains unchanged, cart empty | | Wallet remains unchanged, cart empty | Pass | Cart is empty, no change in wallet balance |
| 7 | IEC1 | |  | | --- | |  |  |  | | --- | | Insufficient balance for expensive product | | Wallet unchanged, cart not empty | | Wallet unchanged, cart not empty | Pass | |  | | --- | |  |  |  | | --- | | Insufficient balance prevents checkout | |
| 8 | VEC7 | Exact wallet balance matches total product price | Wallet reduced to zero, cart empty | | Wallet reduced to zero, cart empty | Pass | Total price exactly matches wallet balance |
| 9 | IEC2 | Product with zero price | Wallet unchanged, cart empty | | Wallet unchanged, cart empty | Pass | Product with zero price does not affect wallet |
| 10 | VEC8 | Multiple products, exact total matches wallet balance | Wallet reduced by total cost, cart empty | | Wallet reduced by total cost, cart empty | Pass | Exact match between wallet and total cost |
| 11 | IEC3 | Negative wallet balance | Wallet balance remains negative, cart not empty | | Wallet balance remains negative, cart not empty | Pass | Negative wallet balance doesn't prevent adding items to cart |
| 12 | IEC4 | Zero price and quantity | Wallet unchanged, cart empty | | Wallet unchanged, cart empty | Pass | Invalid product (zero price and quantity) handled correctly |
| 13 | IEC5 | Empty cart checkout | Wallet remains unchanged, cart empty | | Wallet remains unchanged, cart empty | Pass | No effect when cart is empty during checkout |
| 14 | IEC6 | Small wallet balance with expensive product | Wallet unchanged, cart not empty | | Wallet unchanged, cart not empty | Pass | Insufficient balance does not clear cart |