

Function 1: Cart checkout

White box testing method: Mutation Testing

Mutation type	Mutation	pass/fail
decision mutation	change == to != if len(shopping_cart.cart) != 0:	fail
decision mutation	change == to != if len(payment_methods) != 0:	fail
decision mutation	change != to == if choice.lower() == "yes":	fail
value mutation	add 1 if len(shopping_cart.cart) == 1:	fail
value mutation	add 1 if len(payment_methods) == 1:	fail
statement mutation	delete line user_cart_total = shopping_cart.cart_total_price()	fail (error)
statement mutation	delete line card_number = payment_methods[0]['credit_card']	fail (error)
statement mutation	delete line shopping_cart.cancel_order() (ensure in test cart remains full)	fail

Function 2: Add to cart

White box testing method: Statement Coverage

Statement line	Inputs	Test	Output
20	item = {"item_name": "Salad", "price": 8.99} restaurant = "Nonexistent Restaurant" quantity = 1	T1: The restaurant is not in the system	pass

21	item = {"item_name": "Salad", "price": 8.99} restaurant = "Nonexistent Restaurant" quantity = 1		
24	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = -1	T2: Invalid item quantity	pass
25	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = -1		
27	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = 1		
28	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = 1	T3: Valid restaurant, Valid item quantity, Restaurant is in cart	pass
29	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = 1	T4: Valid restaurant, Valid item quantity, Restaurant is in cart, item is already in cart (just increase its quantity)	pass
31	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = 1		
32	item = {"item_name": "Cheesy Morning Fries", "price": 6.59} restaurant = "Bunrise Burgers" quantity = 1		
33	item = {"item_name": "Classic Bunrise Burger", "price": 8.79} restaurant = "Bunrise Burgers" quantity = 2	T5: Valid restaurant, Valid item quantity, Restaurant is in cart, item is not already in cart(add item to cart)	pass
35-38	item = {"item_name": "Classic		

	Bunrise Burger", "price": 8.79} restaurant = "Bunrise Burgers" quantity = 2		
39	item = {"item_name": "Classic Bunrise Burger", "price": 8.79} restaurant = "Bunrise Burgers" quantity = 2		
42-49	item = {"item_name": "Lobster Tortellini", "price": 18.0} restaurant = "Ocean Restaurant" quantity = 1	T6: Valid restaurant, Valid item quantity, Restaurant is not in cart, add it along with the item	pass
50	item = {"item_name": "Lobster Tortellini", "price": 18.0} restaurant = "Ocean Restaurant" quantity = 1		

For all mutation tests, they either failed (invalid assertions), or had errors (syntax errors from mutations), so they were all “killed”.

For all statement coverage tests, the statements are all covered through each test, and they all pass, implying all statements in the function are run.