Unit testing of the login function

The objective of the login function is to login a user. If the user does not already exist he gets the option to register.

The login() function takes as input username and password entered by the user. Thus out of {int,float,string,list} all are technically invalid types, but get casted to string nonetheless.

We now turn to the valid inputs. The input domain can be divided into five equivalence classes (EC):

1) EC1: Username & Password are correct.
2) EC2: Username correct, Password wrong.
3) EC3: Username incorrect, no registration wanted by user.
4) EC4: Username incorrect, register with valid password.
5) EC5: Username incorrect, register with invalid password and retry until valid.

The coverage criteria we will use are:
For each EC, we need to test at least one input.

This leads to the following test requirements:
R1. If input is in EC1, Printout contains "Successfully logged in"
R2. If input is in EC2, Printout contains "Either username or password were incorrect"
R3. If input is in EC3, Printout contains "Username does not exists."
R4. If input is in EC4, Printout contains "Username does not exists." & "Successfully registered"
R5. If input is in EC5, Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"

These lead to the corresponding test cases:
TC1. Input= Userinputs:[“testuser”,”Test_Password”]
TC2. Input= Userinputs:["testuser", "wrongpassword"]
TC3. Input= Userinputs:["non_existing_user", "wrongpassword", "N"]
TC4. Input= Userinputs:["non_existing_user", " Test_Password", "N"]
TC5. Input= Userinputs:["non_existing_user", "password", "Y", "Correct_New_Password"]
TC6. Input= Userinputs:["non_existing_user", "password", "Y", "8Letter$"]
TC7. Input= Userinputs:["non_existing_user", "password", "Y", "password_missing_capital_letter", "Correct_New_Password"]
TC8. Input= Userinputs:["non_existing_user", "password", "Y", "PasswordMissingSpecialCharacter", "Correct_New_Password"]
TC9. Input= Userinputs:["non_existing_user", "password", "Y", "7L%tter", "Correct_New_Password"]
TC10. Input= Userinputs:["non_existing_user", "password", "Y", "password_missing_capital_letter", "PasswordMissingSpecialCharacter", "Pw$hort", "Correct_New_Password"]

In order to be able to execute the test cases, the fixtures json_dump_mock, registered_user and login_open_users_file_stub were written.

Unit testing of the logout function

The objective of the logout function is to check if the ShoppingCart is empty or not and then depending on that return True or False. In case the cart is not empty the user gets the option to clear the cart (and return True) or stay logged in (return False).

The polygon(nodes) function takes as input a ShoppingCart. Thus out of only objects of type ShoppingCart are invalid types.

We now turn to the valid inputs. The input domain can be divided into seven equivalence classes (EC):

1) EC1: Cart contains no items
2) EC2: Cart contains 1+ items, user wants to keep cart and not logout
3) EC3: Cart contains 1+ items, user wants to clear cart and logout

The coverage criteria we will use are:
For each EC, we need to test at least one input.

This leads to the following test requirements:
R1. If input is in EC1, Function returned True; Cart.items = 0;
R2. If input is in EC2, Function returned False; Cart.items != 0;
R3. If input is in EC3, Function returned True; Cart.items = 0;

These lead to the corresponding test cases:
TC1. Input= ShoppingCart without any products
TC2. Input= ShoppingCart with Product(name="Apple", price=5, units=10); UserInput: “n”;
TC3. Input= ShoppingCart with Product(name="Apple", price=5, units=10); UserInput: “No”;
TC4. Input= ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2); UserInput: “n”;
TC5. Input= ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2), Product(name="Orange", price=12, units=5); UserInput: “n”;
TC6. Input= ShoppingCart with Product(name="Apple", price=5, units=10); UserInput: “y”;
TC7. Input= ShoppingCart with Product(name="Apple", price=5, units=10); UserInput: “Yes”;
TC8. Input= ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2); UserInput: “y”;
TC9. Input= ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2), Product(name="Orange", price=12, units=5); UserInput: “y”;

In order to be able to execute the test cases, the fixtures cart_empty, cart_with_one_element and cart_with_two_elements were written.

Test Function Document

Project Name: A1_unit_testing_students	
Automation Title: test_login	Version: 1.0
Testing Phase: Phase 1	Date of Test: 23 nov 2023
Module Name: A1_unit_testing_students	

Function Title: login	Test Designed by: Jannik Recklies
Test Priority (Low/Medium/High): Medium	Test Designed Date: 23 Nov 2023
Description: The login function takes in user input (username, password) and checks for correctness by comparing to both inputs with file content of users.json. If username does not exist, it will further ask for registration. In that case the user can enter a password and will then be stored in the users.json file.	Test Executed by: Jannik Recklies Test Execution date: 23 nov 2023

Pre-conditions: Mock users.json and set content to [{"username": "testuser", "password": "Test_Password", "wallet": 0}]

Dependencies: None

S. No	Equivalence Class	Test case data	Expected Results	Actual Results	Status (Pass/ Fail)	Notes
1	EC1	Userinputs: [“testuser”, ”Test_Password”]	Printout contains "Successfully logged in"	Printout contains "Successfully logged in"	Pass	Entered username & password are correct. Test name: test_login_successful ()
2	EC2	Userinputs: ["testuser", "wrongpassword"]	Printout contains "Either username or password were incorrect"	Printout contains "Either username or password were incorrect"	Pass	Entered username correct, password incorrect. Test name: test_login_with_incorrect_password()
3	EC3	Userinputs: ["non_existing_user", "wrongpassword", "N"]	Printout contains "Username does not exists."	Printout contains "Username does not exists."	Pass	Username incorrect, Password exists, don’t register. Test name: test_login_with_non_existing_username_and_not_register()
4	EC3	Userinputs: ["non_existing_user", " Test_Password", "N"]	Printout contains "Username does	Printout contains "Username does	Pass	Username incorrect, Password exists, don’t register. Test name:

			not exists."	not exists."		test_login_with_non_existing_username_and_not_register()
5	EC4	Userinputs: ["non_existing_user", "password", "Y", "Correct_New_Password"]	Printout contains "Username does not exists." & "Successfully registered"	Printout contains "Username does not exists." & "Successfully registered"	Pass	Register correct on first try. Test name: test_login_failed_and_register_correct()
6	EC4	Userinputs: ["non_existing_user", "password", "Y", "8Letter$"]	Printout contains "Username does not exists." & "Successfully registered"	Printout contains "Username does not exists." & "Successfully registered"	Pass	Register correct on first try. Test name: test_login_failed_and_register_correct()
7	EC5	Userinputs: ["non_existing_user", "password", "Y", "password_missing_capital_letter", "Correct_New_Password"]	Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"	Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"	Pass	Register with non-accepted password and retry until correct. Test name: test_login_failed_and_retry_register()
8	EC5	Userinputs: ["non_existing_user", "password", "Y", "PasswordMissingSpecialCharacter",	Printout contains "Username does not exists."	Printout contains "Username does not exists."	Pass	Register with non-accepted password and retry until correct. Test name: test_login_failed_and_retry_register()

		"Correct_New_Password"]	& "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"	& "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"		
9	EC5	Userinputs: ["non_existing_user", "password", "Y", "7L%tter", "Correct_New_Password"]	Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"	Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol and be 8 characters long." & "Successfully registered"	Pass	Register with non-accepted password and retry until correct. Test name: test_login_failed_and_retry_register()
10	EC5	Userinputs: ["non_existing_user", "password", "Y", "password_missing_capital_letter", "PasswordMissingSpecialCharacter", "Pw$hort", "Correct_New_Password"]	Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol	Printout contains "Username does not exists." & "Password must have at least 1 capital letter, 1 special symbol	Pass	Register with non-accepted password and retry until correct. Test name: test_login_failed_and_retry_register()

			and be 8 characters long." & "Successfully registered"	and be 8 characters long." & "Successfully registered"		

Test Function Document

Project Name: A1_unit_testing_students	
Automation Title: test_logout	Version: 1.0
Testing Phase: Phase 1	Date of Test: 23 nov 2023
Module Name: A1_unit_testing_students	

Function Title: logout	Test Designed by: Jannik Recklies
Test Priority (Low/Medium/High): Medium	Test Designed Date: 23 Nov 2023
Description: The logout function takes a cart as input and returns true if the cart has no more products in it. In the case that the cart is not empty, the user gets asked if it should be emptied or not. If the user answers yes, the cart gets emptied, and the function returns true. Otherwise, the cart stays the same and the function returns false.	Test Executed by: Jannik Recklies Test Execution date: 23 nov 2023
Pre-conditions: None	
Dependencies: None	

S. No	Equivalence Class	Test case data	Expected Results	Actual Results	Status (Pass/Fail)	Notes
1	EC1	ShoppingCart without any products	Function returned True; Cart.items = 0	Function returned True; Cart.items = 0	Pass	Test name: test_logout_with_empty_cart()
2	EC2	ShoppingCart with Product(name="Apple", price=5, units=10) UserInput: “n”	Function returned False; Cart.items = 1	Function returned False; Cart.items = 1	Pass	Test name: test_cancel_logout_cart_with_ one_element()
3	EC2	ShoppingCart with Product(name="Apple", price=5, units=10) UserInput: “No”	Function returned False; Cart.items = 1	Function returned False; Cart.items = 1	Pass	Test name: test_cancel_logout_cart_with_ one_element()
4	EC2	ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2) UserInput: “n”	Function returned False; Cart.items = 2	Function returned False; Cart.items = 2	Pass	Test name: test_cancel_logout_cart _with_two_elements()

5	EC2	ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2), Product(name="Orange", price=12, units=5) UserInput: “n”	Function returned False; Cart.items = 3	Function returned False; Cart.items = 3	Pass	Test name: test_cancel_logout_cart_with_multiple_elements()
6	EC3	ShoppingCart with Product(name="Apple", price=5, units=10) UserInput: “y”	Function returned True; Cart.items = 0	Function returned True; Cart.items = 0	Pass	Test name: test_logout_clear_cart_with_one_element()
7	EC3	ShoppingCart with Product(name="Apple", price=5, units=10) UserInput: “Yes”	Function returned True; Cart.items = 0	Function returned True; Cart.items = 0”	Pass	Test name: test_logout_clear_cart_with_one_element()
8	EC3	ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2) UserInput: “y”	Function returned True; Cart.items = 0	Function returned True; Cart.items = 0	Pass	Test name: test_logout_clear_cart_with_two_elements
9	EC3	ShoppingCart with Product(name="Apple", price=5, units=10), Product(name="Banana", price=10, units=2), Product(name="Orange", price=12, units=5) UserInput: “y”	Function returned True; Cart.items = 0	Function returned True; Cart.items = 0	Pass	Test name: test_logout_clear_cart_with_multiple_elements()