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.