FUNCTIONAL TESTING

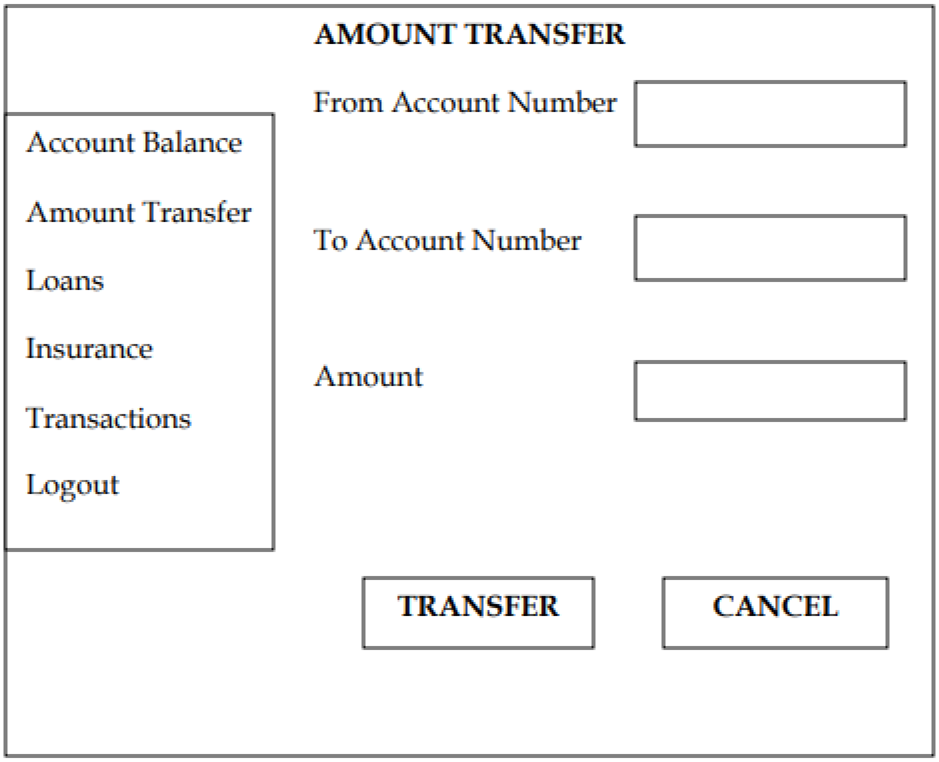
This is the type of Black Box Testing. Also called component testing. Testing each and every component thoroughly (rigorously) against requirement specifications is known as functional testing.

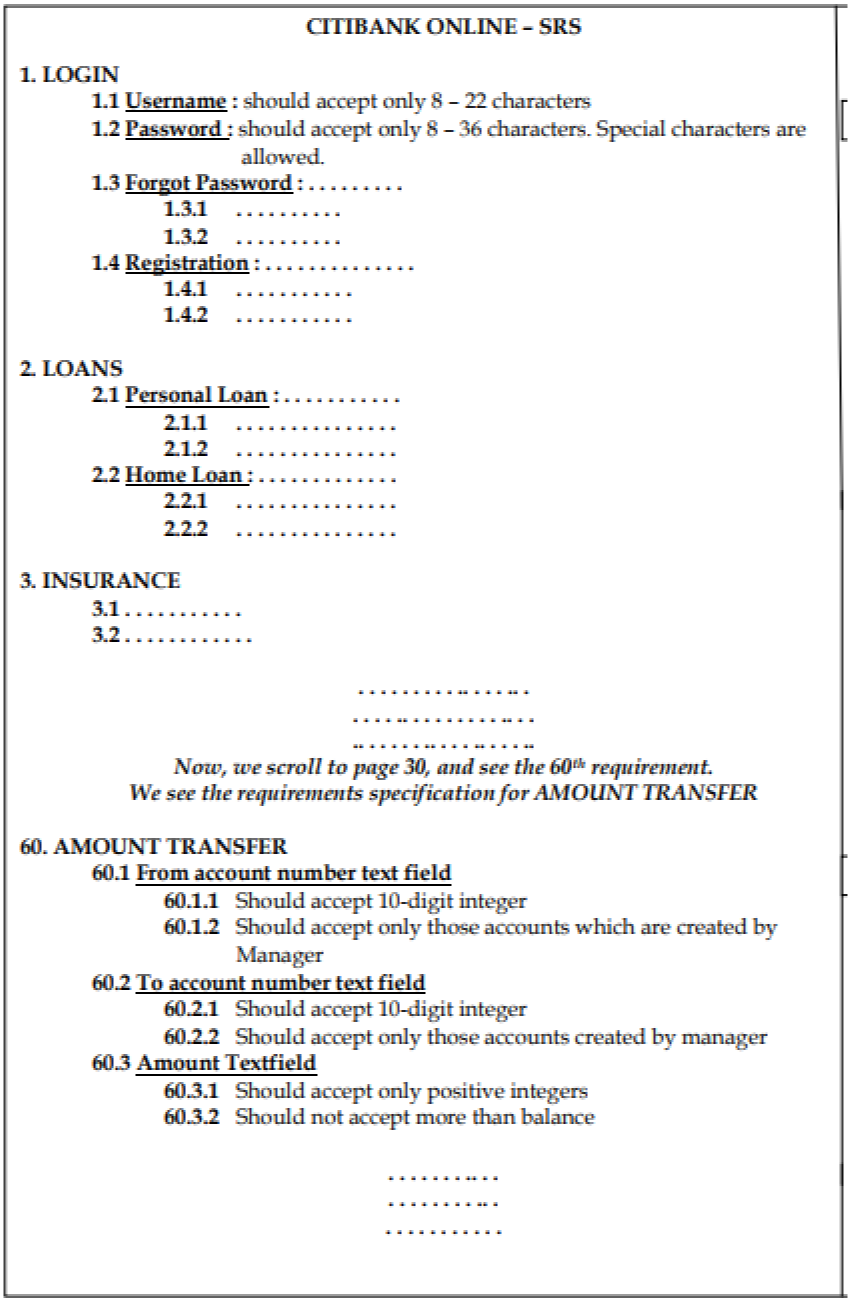
Functional testing verifies that the software performs its stated functions in a way that the users expect.

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements

For example, let us consider that ibank wants a s/w for banking purposes, and it asks the company Iflex to develop this s/w. The s/w is something as shown below. When the user clicks his valid username and enters his password, then he is taken into the homepage. Once inside the homepage, he clicks on the amount transfer and the below page is displayed. He enters his valid account number and then the account number to which the money is to be transferred. He then enters the necessary amount and clicks on transfer. The amount must be transferred to the other account number.

Now in black box testing, the test engineer tests the s/w against requirements and checks if the s/w is working correctly as per requirements


AMOUNT TRANSFER
Account Balance
Amount Transfer
Loans
Insurance
Transactions
Logout
From Account Number
To Account Number
Amount
TRANSFER
CANCEL



CITIBANK ONLINE – SRS

1. LOGIN

1.1 **Username** : should accept only 8 – 22 characters

1.2 **Password :** should accept only 8 – 36 characters. Special characters are allowed.

1.3 **Forgot Password** : . . . .

1.3.1 . . . .

1.3.2 . . . .

1.4 **Registration** : . . . .

1.4.1 . . . .

1.4.2 . . . .

2. LOANS

2.1 **Personal Loan** : . . . .

2.1.1 . . . .

2.1.2 . . . .

2.2 **Home Loan :** . . . .

2.2.1 . . . .

2.2.2 . . . .

3. INSURANCE

3.1 . . . .

3.2 . . . .

. . . .

. . . .

. . . .

Now, we scroll to page 30, and see the 60th requirement.
We see the requirements specification for AMOUNT TRANSFER

60. AMOUNT TRANSFER

60.1 **From account number text field**

60.1.1 Should accept 10-digit integer

60.1.2 Should accept only those accounts which are created by Manager

60.2 **To account number text field**

60.2.1 Should accept 10-digit integer

60.2.2 Should accept only those accounts created by manager

60.3 **Amount Textfield**

60.3.1 Should accept only positive integers

60.3.2 Should not accept more than balance

. . . .

. . . .

. . . .

This is how the requirements given by the client looks like (figure below). It is usually a word document file. Let us consider that Citibank gives a 80pg SRS in MS-WORD format. The test engineer then looks at the requirements and correspondingly checks the s/w.

Now the test engineer does all possible tests on the 2 account numbers. Now, he proceeds with the testing of Amount transfer. These are the following tests he conducts for testing the amount field, He enters the following data in the amount field,

a) – 100
b) 100
c)100.50
d) Hundred rupees only
e) 100 blank space 0
f) 100
g)0.001

For all the above cases except for f) , it should throw an error message. If it doesn‟t throw, then there is a bug in the s/w and the s/w must be sent to the development team to repair the defect

Thus, during testing, we must remember the following points

a) We must always start testing the application with the valid data. In the above example for amount transfer, we see that we have entered the valid data 100 only in the 6th test. This should not be done, because if the valid data itself is not taken correctly, then we need not have to waste our time checking for the invalid data

b) If the application works for valid data, only then we must start testing for invalid data

c) If the application is not working for 1 of the invalid values, then we can continue testing for all the other invalid values and then submit the test report of all the defects for invalid values.

d) In testing, we should not assume or propose requirements. If we have any queries, talk to the one who knows the requirements very well and clarify the queries.

e) We must not do over-testing (testing for all possible junk values) or under-testing (testing for only 1 or 2 values). We must only try to optimize testing (testing for only the necessary values- both invalid and valid data).

f) We must do both positive testing (testing for valid data) and negative testing (testing for invalid data).

The characteristics of a good requirement are,

Complete – the requirement is fully stated in 1 place with no missing information

Consistent – the requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation

Traceable – the requirement meets all or part of a business need as stated by stakeholders and authoritatively documented

Mandatory – the requirement represents a stakeholder defined characteristic, the absence of which will result in a deficiency that cannot be ameliorated

Important points:

1. **Whenever you have given an application to test, what is the 1st value you will test with ?**
   Ans: Valid Range

2. **The process of functional testing involves a series of tests: Smoke, Sanity, Integration, Regression, Interface, System and finally User Acceptance Testing. Tests are conducted on each feature of the software to determine its behavior, using a combination of inputs simulating normal operating conditions, and deliberate anomalies and errors.**

3. **What are the important steps that are covered in Functional testing?**
   - Understanding the Requirement document specification and clearing the doubts and queries in the form of review comments.
   - After analysis, the requirement specification tester will make a plan
   - After planning the tests, the tester will write the test cases with respect to the requirement specification by keeping in mind all the scenarios should be considered for all the cases.
   - Identifying the test inputs and requesting the test data that is required to execute the test cases as well as to check the functionality of the application.
   - Determine the actual result as per the input values to be tested.
   - Execute the test cases that determine whether application behavior is as expected or any defect has occurred.

- Compare the actual result and the computed result to find out the actual outcome.
-

4. **What to test in functional testing? Explain**
    - Basic Usability: Functional Testing involves the usability testing of the system. It checks whether a user can navigate freely without any difficulty through screens.
    - Accessibility: Functional testing test the accessibility of the function.
    - Mainline function: It focuses on testing the main feature.
    - Error Condition: Functional testing is used to check the error condition. It checks whether the error message is displayed.