Test Plan

Andreea Șindrilaru
Fontys University of Applied Sciences
S3-CB-01

1. Why should we test?

Software that we develop must be checked carefully so that there are no bugs or errors in the software. In case there are issues, testing helps finding those in an early stage and fixing them before the product is delivered, increasing the quality of it.

The user test cases can be found within the Excel file from the "Documentation" folder.

2. What should we test?

First, we need to make sure that each important part of the code is integrated and working correctly. Therefore, unit testing is going to be done to test the behavior of the code.

Secondly, after unit testing is covered, we need to ensure that each component of the application is working fine. To test the behavior of the components, integration testing is going to be done.

The third type of testing is the system test. This is where we test the behavior of the whole system to ensure it is working correctly and respects the user stories and the defined requirements.

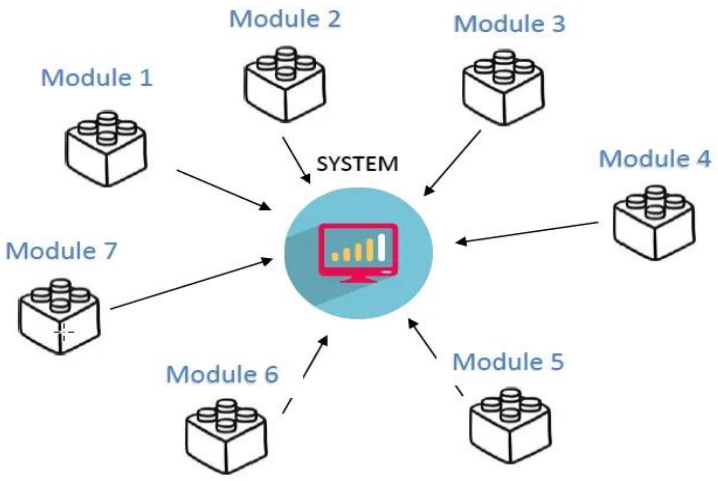
Finally, the last test to be done is acceptance test where it is determined if the system satisfies the acceptance criteria.

3. How should we test?

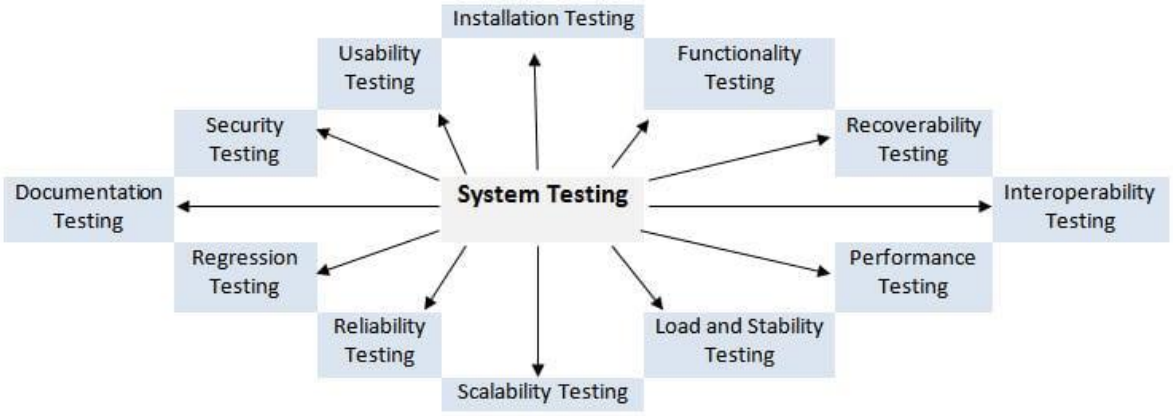
For unit testing it has been decided to use JUnit5 for testing the Data Access Layer, and Mockito for mocking the data needed for testing the Service Layer.

```
[Test]
public void TestMethod1()
{
    var calc = new Calculator();
    calc.ValidOperation = Calculator.Operation.Multiply;
    calc.ValidType = typeof(int);
    var result = calc.Multiply(-1, 3);
    Assert.AreEqual(result, -3);
    calc.ValidOperation = Calculator.Operation.Multiply;
    calc.ValidType = typeof(int);
    result = calc.Multiply(1, 3);
    Assert.IsTrue(result == 3);
    if (calc.ValidOperation == Calculator.Operation.Invalid)
    {
        throw new Exception("Operation should be valid");
    }
    calc.ValidOperation = Calculator.Operation.Multiply;
    calc.ValidType = typeof(int);
    result = calc.Multiply(10, 3);
    Assert.AreEqual(result, 30);
}
```

After finishing the unit testing part, we move to the integration part, where I will use newman, which is a command-line runner for collections made within Postman, to ensure all API endpoints are called correctly.



For the system tests I have to tests the entire system as mentioned above. Test cases to test the user stories and acceptance criteria are already made. For short, in this test I will check if all test cases have passed or failed.



System Testing - © www.SoftwareTestingHelp.com

The last test is acceptance testing, where Cypress will be used to make sure that each user story is completely fulfilled.