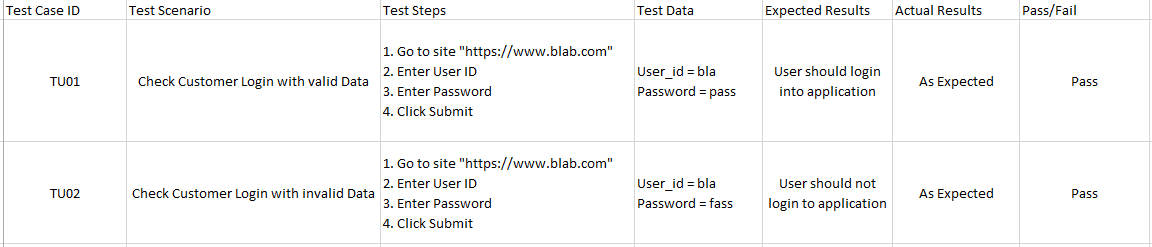
Format is



1. Test cases should be simple and transparent where concise and clear is best with use of assertive language
2. Create test Case with end user in mind where the ultimate goal of a project is to create cases which meets real users.
3. Avoid test case repetition by using pre conditions for reference
4. Do not assume functionality and features. Stick to specifications
5. Ensure 100% coverage where all software requirements are checked
6. Test cases should be identifiable while tracking defects
7. Implement testing techniques like BVA, EP, STA, EGT
8. Self-clean the test environment to pre-test state that should not render the environment unusable
9. Test cases should generate the same results every time it is run
10. Have the test cases reviewed by colleagues who may uncover defects i.e. Peer Review

Zain and Andrews notes

MTM is used to create and manage test cases

To access test for EV.Cloud

* Connect to team project
* Integration Team

Writing of test cases

* Test Case should be meaningful. Not long
* Add tags in square brackets (e.g. [AM][DB] Verify Forgot password link) in indicate product and are of the product
* Leave default value for state and priority (“New” and “2”)
  + If NO> “ not applicable” or “not automated”
* Area always pick integration
* Test steps should be clear enough for anyone to test it
* No more than 8 to 10 steps
* Database queries can be attached to test cases

Testing the steps

* Go to test tab in MTM
* Find the test case you want to run
* Click run
* Small box opens containing all of the test case steps
* You can pass some individual steps
* If one fails you can mark it as failed and give a reason

Environments

Dev Environment is unstable for testing

QA environment is suitable place for testing

Flow of Work

Developer creates a feature in their environment

Developer gives a PA check to tester

Developer pushes code to QA where functionality is tested

Types of testing

Acceptance testing: Testing against an acceptance criteria specified in user story.

Smoke/Sanity testing: Quick and dirty test to ensure all features still work after a change

Regression testing: Full end to end test of a product

Bugs

When you find a bug first attempt to reproduce it. Once you’ve done that report the bug and log it in TFS.

When logging a bug:

* Give it a clear title and make use of relevant tags of product and area
* Include the steps to reproduce the bug
* The results attributed to the bug
* The expected results that should have been achieved
* Under “Misc” any extra info
* Assign the bug to the referred developer
* The Severity should be recorded
* Environments
  + Dev/QA – Environment
  + Production: the real deal
  + Pre Production: Replica of actual production