

1.

How to test depends mainly on the requirements, usually described in the specification. Very important are such factors as risk level, purpose of tests, time, budget, software development life cycle phase etc. Tests should verify both functional and non-functional requirements, performed on various browsers and systems.

Suggested test types:

- Functional tests (functions to be performed by the software):
 - all validations
 - keyboard navigation
 - clearing the form
 - sending data to the server
 - correctness of error/information messages
- Non-functional tests (quality characteristics):
 - compatibility (with different browsers and operating systems)
 - usability (e.g. readability and usefulness of error messages)
 - load tests (with maximum amount of characters in the form fields)
 - reliability (e.g. database security)
 - correctness of GUI (including scalability, page refreshing)
- Tests related to changes (retests in case of testing after bugs fixing, regression tests in case of any modifications made in the software or in the environment)

Test scenario classes:

- verification of GUI (correct position and appearance of all specified elements, no language errors, mandatory fields marked with * sign)
- keyboard navigation tests
- verification if the limit of possible characters to enter into the fields is based on requirements
- specific fields validation
 - email: local-part@domain
 - age: numeric between 0 and 100
 - phone: numeric, length, country code if required
- verification of cleaning the form (whether the form fields are partially or completely cleaned after clicking on Clear button)
- verification of submitting data to the server
 - with all the mandatory fields filled in (with/without optional fields)
 - without all the mandatory fields (validation error, data not send to the server)
- verification of trimming leading and trailing spaces after clicking on Submit button (entering only whitespaces into mandatory fields is impossible)
- verification correctness of error/confirmation messages

Example of test cases: in the attached file sample_test_cases.xlsx

2.

Example bug report: in the attached file sample_bug_report.xlsx

3.

Automation:

For grouped and ordered selected test cases.

Selenium IDE (Mozilla Firefox plugin for older version of Firefox than 55.00)— record all the user actions (mouse clicks, data fillings, submissions etc.) - than it can be played back

or

Selenium WebDriver (API to communicate with different browsers by test scripts written in programming language)

Taking care of maintainability:

- include static analysis and reviews
- focus on testing important and unstable features
- avoid complex test scenarios
- clean code - simplicity and flexibility
- independant test scripts for testing independant features
- use proper, unambiguous identifiers
- write test reports and analyze what can be changed/improved during future testing
- update and archive testware, test environment and test infrastructure for later reuse