

The testing plan

This testing plan will be mostly based on integration tests. The integration tests will focus on interfacing with the web application in a web browser through a mouse and a keyboard and assuring that the specified inputs correspond to the desired outputs. To ease this process we will be using tools such as Selenium that allows us to automate such tests. With the help of VirtualBox the tests will be evaluated on all platforms. In every platform the tests will be executed against all major browsers that are popular for the particular OS. The coding team should be concerned about writing this tests and since this tests will be automated they can be easily evaluated at least once a day.

Stress tests will be conducted with Apache JMeter which is a graphical server performance testing tool, for both static and dynamic resources. It can be used to simulate a heavy load on a server to analyze overall performance under different load types.

We will also have an independent testing team that is not working on the current project, since they can bring an out of the box view on the whole system and test details that a programmer may easily omit.

We will provide 2 security experts John and Smith that will focus on the security part of the system like SQL injections, post/get modifications, JavaScript injections, ...

Tests will be evaluated in the order they are listed above. Automated integration tests will run probably once a day as programmers see fit, the testing team will be testing the system after every major update. After every major update the stress tests will also be done and John and Smith will try to find loopholes in the system before the final release.

Test design specification

Subscribe

Preconditions: The user must sign, find a subforum and click on "subscribe".

Outcome: If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Unsubscribe

Preconditions: The user must sign in, find a subforum, has to be already subscribed to that subforum and click on "unsubscribe".

Outcome: If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Create new discussions

Preconditions: The user must sign in, find a subforum, be subscribed to that subforum, click on "create new discussion", fill in "title" and "body" and click on "submit". "Body" must have more than 100 character and "title" more than 10.

Outcome: If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Comment

Preconditions: The user must sign in, find a subforum, be subscribed to that subforum, find an open discussion, click on "comment", insert the "body" and click on "submit". "Body" has to have more than 100 character

Outcome: If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Vote discussion

Preconditions: The user must sign in, find a subforum, be subscribed to the subforum, find a discussion, click on "vote positive" or "vote negative" button and the user has to leave a comment with his/her view on the matter.

Outcome: If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Vote comments

Preconditions: The user must sign in, find a subforum, be subscribed to the subforum, find a discussion, find a comment, click on "vote positive" or "vote negative" button and if the vote is negative the user has to leave a comment with an explanation.

Outcome: If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Hide a discussion

Preconditions: User must be a moderator, must click on "show reported content", choose a comment or discussion and click on "hide".

Outcome: Discussion/comment is hidden, only seen in a separate part of the subforum called "reported content". If all constraints are satisfied the user receives a positive notification else a negative one.

Priority=High

Filters

Preconditions: The user has to select a filter, write terms to search and click the "search" button.

Outcome: All future searches are restricted by the selected filter.

Priority=Medium

Tabs

Preconditions: The user has to select a subforum and select a tab "rising", "new" or "vote".

Outcome: Discussion are shown based on criteria selected.

Priority=Medium

Navigation

Preconditions: The user has to select a geographical region, repeat the process or choose a hot discussion that are being shown.

Outcome: Redirect to selected discussion.

Priority=Medium

Reporting inappropriate content

Preconditions: User must be sing-in, subscribed to the subforum, find a discussion/comment, click "report", fill in the "reason" and click on the "submit" button. The reason has to contain more than 20 characters.

Outcome: if enough reports, moderators are alerted and comment gets 'minimized'. Discussion loses ranking. The user receives a positive notification if the reason has more than 20 characters else return negative notification.

Priority=High

Test case specification

For brevity we assume that all tests that include signing-in, being subscribed, being a moderator and finding the right subforum/discussion/comment or any other page are already successfully tested. We will focus just on the edge cases for some tests and we will also skip tests that require only to click on a button and the outcome is just a positive notification.

ID: 1

Test name: Create new discussions

Assumptions: signed-in = true and subscribed = true

Sequence of actions:

System executes the following tests, and all cases match appropriate results.

body = "123" and title = "123" => negative notification

body = more than 100 characters , title = "" => negative notification

body = "" , title = more than 10 characters => negative notification

body = more than 100 characters , title = more than 10 characters => positive notification

Outcome: true

ID: 2

Test name: Comment

Assumptions: signed-in = true and subscribed = true

Sequence of actions:

System executes the following tests, and all cases match appropriate results.

body = "123" => negative notification

body = more than 100 characters => positive notification

Outcome: true

ID: 3

Test name: Vote discussion – comment required?

Assumptions: signed-in = true and subscribed = true

Sequence of actions:

System clicks on the "vote positive" button and checks if comment form shows up.

Outcome: true

ID: 4

Test name: Vote discussion – comment not required?

Assumptions: signed-in = true and subscribed = true

Sequence of actions:

System clicks on the "vote negative" button and checks if comment form shows up.

Outcome: true

ID: 5

Test name: Vote discussion – vote positive

Assumptions: signed-in = true and subscribed = true

Sequence of actions:

System clicks on the "vote positive" button and inserts a comment.

Outcome: positive notification

ID: 6

Test name: Vote discussion – vote negative comment required?

Assumptions: signed -in = true and subscribed = true

Sequence of actions:

The system clicks on the "vote negative" button and inserts a comment.

Outcome: positive notification

ID: 7

Test name: Vote comment – vote negative comment required?

Assumptions: signed -in = true and subscribed = true

Sequence of actions:

The system clicks on the "vote negative" button and inserts a comment.

Outcome: positive notification

ID: 8

Test name: Vote comment – comment required?

Assumptions: signed -in = true and subscribed = true

Sequence of actions:

The system clicks on the "vote negative" button and checks if comment form shows up.

Outcome: true

ID: 9

Test name: Vote comment – comment not required?

Assumptions: signed -in = true and subscribed = true

Sequence of actions:

The system clicks on the "vote positive" button and checks if comment form does not shows up.

Outcome: true

ID: 10

Test name: Filters - "religion"

Assumptions: None

Sequence of actions:

The system selects the filter "religion", inserts "atheism" into search input box and presses "search".

Extracts the contents of the dom elements returned by the web app and compares them with the result of the database query with those filters in it.

Outcome: true

ID: 11

Test name: Tabs - "rising"

Assumptions: None

Sequence of actions:

The system selects the tab "rising" and extracts the contents of the dom elements returned by the web app and compares them with the result of the database query with those filters in it.

Outcome: true

ID: 12**Test name:** Navigation - "Slovenija"**Assumptions:** None**Sequence of actions:**

The system selects the "Slovenija" region, it gets redirected and extracts the contents of the dom elements returned by the web app and compares them with the result of the database query with those geographical regions included in.

Outcome: true**ID: 13****Test name:** Reporting inappropriate content - more than 20 characters**Assumptions:** signed -in = true**Sequence of actions:**

The system clicks on the "report" button. Fills in the reason with more than 20 characters and clicks "submit".

Outcome: positive notification**ID: 14****Test name:** Reporting inappropriate content – alerted moderators**Assumptions:** signed -in = true**Sequence of actions:**

The system clicks on the "report" button. Fills in the reason with more than 20 characters and clicks "submit". Repeats the process 100x times and check the database if the moderators are notified.

Outcome: positive notification**ID: 15****Test name:** Reporting inappropriate content - less than 20 characters**Assumptions:** signed -in = true**Sequence of actions:**

The system clicks on the "report" button. Fills in the reason with less than 20 characters and clicks "submit".

Outcome: negative notification**ID: 16****Test name:** Reporting inappropriate content – Rank loss**Assumptions:** signed -in = true**Sequence of actions:**

The system clicks on the "report" button. Fills in the reason with more than 20 characters and clicks "submit". Checks the database if the reported content affected the rank of the topic and discussion.

Outcome: positive notification

ID: 17

Test name: Hide a discussion – hidden?

Assumptions: moderator = true

Sequence of actions:

The system clicks on the "show reported content" button, selects a discussion, and clicks on the "hide" button and checks the visibility of the comment.

Outcome: true

ID: 18

Test name: Hide a discussion – "reported" visible?

Assumptions: moderator = true

Sequence of actions:

The system clicks on the "show reported content" button, selects a discussion, and clicks on the "hide" button and checks if the comments is shown in the "reported" subforum.

Outcome: true

ID: 19

Test name: Hide a discussion – positive?

Assumptions: moderator = true

Sequence of actions:

The system clicks on the "show reported content" button, selects a discussion, and clicks on the "hide" button.

Outcome: positive notification

Test procedure

It is mostly described in the test plan.

Test item transmittal report

As we can see from the outputs of the integration tests above we can deduce that the system is in a solid state with all the major functionality working as specified in the specification. For the system to be ready for production the rest of the tests should be done and most importantly all the unit test should be "green" as well.