

Test Plan
Debate.org/opinions

Prepared by:
Holly Liulka
3 March 2019

TABLE OF CONTENTS (TOC)

1. Introduction
2. Scope
3. Testing Strategy
4. Hardware Requirements
5. Features to Be Tested
6. Features Not to Be Tested
7. Resources/Roles & Responsibilities
8. Quality Control Metrics and Reports
9. Risks/Assumptions
10. Tools
11. Approvals

1. INTRODUCTION

This product is a web application which allows users to submit a yes or no opinion and write a short justification for why they hold that opinion. Users are also able to create their own topics for others to submit opinions on.

2. SCOPE

General:

QA efforts will be focused on verifying functionality of the “Opinions” section of Debate.org.

Testing activities will include

1. designing and executing test cases
2. reporting and following bugs
3. Generate QA reports per sprint

3. TESTING STRATEGY

Test cases will be executed using a variety of tools and techniques. Unit testing should be done by developers as they are creating features or during off sprints. Integration tests will be

created by QA's using Postman and run in the CI/CD tool with each commit using Newman. UI tests will be accomplished with both manual and automated testing strategies. Automation will be performed using Selenium WebDriver and will be used for smoke testing and regression testing on key features at the discretion of the team lead. Exploratory and remaining regression testing will be performed manually.

Functional Test Cases will be designed by QA Team for each User Story or for each requirement, depending on project documentation. A test case will contain:

- Test case ID: Number that identifies the TC
- Priority
- User Story: Number and Code of the User Story, or link to US if there are integrated management tools.
- Objective: Brief description of what the Test Case wants to achieve
- Users: Application role that should be used to execute the test case.
- Precondition: Situations needed to run the Test Case
- Steps: It describes the specific steps to run the Test Case
- Expected Result: it describes how the system should work running the correct steps
- Result / Reason: the possible states are: Pass, Fail, Can't Run, Not Apply
- Creator name
- Comments

Once the developer completes the code for a User Story and the corresponding code review was approved, the QA engineer will execute a smoke testing first in order to check that the main functionality works as requested. After this, the functional testing is performed. The test cases associated to the User Story related to the code complete in the QA environment are executed. For each test execution cycle and each executed TC QA engineers complete:

- Execution Date
- Tester Name: The person who will execute the Test Case
- ID bug: In case to identify a bug running the Test Case, the number of the bug is registered.

If the QA engineer finds any bug it will be reported in the bug tracking tool with all the necessary information and will be assigned to corresponding developer. If the QA engineer finds any necessary enhancement it will be reported to POs so they will decide if it will be implemented or rejected.

When developers finish fixing bugs, they will be marked as fixed and passed to QA engineer for validation on the QA environment. The QA engineer will validate the fixes and execute a Regression testing to verify the basic integrity of the application.

If the fix is correct, the bug will be closed. If a fix is incorrect, the bug will be reopened and resent to the assigned developer. The validation of an incorrect fix will continue until the bug is fixed or the acceptance criteria are reached.

When the code for sprint User Stories is finished, and tested on the QA environment, the QA engineer will execute an Integration testing. The purpose of this test is to ensure that

the functionality tested for each Story continues working as expected once it is integrated with previous one.

If the QA engineer finds any bug it will be reported and will follow the same steps for bugs defined above.

When a release is complete, the code related will be deployed to the QA environment and the QA engineer will execute a regression test. Manual test will be executed.

When the regression is completed, if it meets the Exit Criteria, all the code related to the sprint will be deployed to the Production environment and the QA engineer will execute a smoke test.

.....

4. HARDWARE REQUIREMENTS

Mac and/or Windows computer able to run popular browsers (i.e. chrome, firefox, safari, edge, etc.)

.....

5. FEATURES TO BE TESTED

Opinions section of Debate.org.

.....

6. FEATURES NOT TO BE TESTED

Other sections of Debate.org.

.....

7. RESOURCES/ROLES & RESPONSIBILITIES

QA Engineer - Holly Liulka

- Elaborate Test Plan.
- Define Test Cases.
- Run TC in order to ensure the quality of the developed product.
- Raise and track all bugs and issues.
- Generate QA reports and metrics.
- Participate in determining test cases to be automated.
- Automate test cases.

.....

8. QUALITY CONTROL METRICS AND REPORTS

For this project, the following QA metrics and reports will be generated and delivered to the team:

- Test Execution Status Report: Will be generated periodically during the sprint indicating (general metrics for internal use only):

- % of Test Cases executed
- Bugs by severity
- Bugs by Status
- Test Execution Summary Report: At the end of the execution of the TCs for each iteration, a summary status report will be generated indicating:
 - Sprint Stories Status
 - Sprint Test Cases execution summary by status (pass, fail, unexecuted)
 - Sprint Bugs by Severity
 - Total number of Bugs by Severity
 - Sprint Bugs by Status
 - Total number of Bugs by Status

9. RISKS/ASSUMPTIONS

None for the purposes of this project (examples could be lack of time or resources)

10. TOOLS

Activity	Tool
User story tracking	Jira
Bug Tracking	Jira
Test case management	Excel (testrail would be preferred)
Project documentation	Dropbox/Confluence
Web services testing	Postman
Automation	Selenium WebDriver w/Python

11. APPROVALS

Specify the names and titles of all the persons who must approve this plan. Provide space for the signatures and dates.

Name (In Capital Letters) Signature Date:

- 1.
- 2.
- 3.
- 4.