Test Plan For Trello Website

Version: 1.0

Created: 26/03/2024

Last Updated: 26/03/2024

Trello Website - Automation Testing

Prepared by : Ahmd Bdran

Table of Content

1. Introduction

Trello is a popular web-based project management application that utilizes boards, lists, and cards to organize tasks and collaborate with team members. It provides a flexible and intuitive interface for users to create and manage projects of any size or complexity. With features such as customizable boards, task assignments, due dates, labels, and comments, Trello facilitates efficient project planning and execution for teams across various industries.

The key functionalities of the Website include:

1.1. Board Creation and Management

Users can create boards to represent projects or workflows and customize them with lists and cards to organize tasks.

1.2. Task Creation and Tracking

Users can create tasks (cards) within boards, assign them to team members, set due dates, add labels, attachments, and checklists to track progress.

1.3. Collaboration and Communication

Trello enables seamless collaboration among team members through comments, mentions, and file attachments on cards, facilitating effective communication and information sharing.

2. Objectives

This test plan outlines the testing approach for the testing of Trello Website. It covers the following key areas:

1. Test Strategy: Defines the rules the test will be based on, including project timelines, objectives, and assumptions.
2. Execution Strategy: Describes how the tests will be performed, the process for identifying and reporting defects, and implementing fixes.
3. Test Management: Outlines the process for handling the logistics of the test, including communication, escalation procedures, risk mitigation, and team roster.

3. Test Tree

Functional tests

1. **Board Creation and Management**

    1.1. Verify board creation process:
    - 1.1.1 Create a new board with a random title
    - 1.1.2 Customize board settings (e.g., background color, visibility)

    1.2. Verify list and card management:
    - 1.2.1 Add lists to the board
    - 1.2.2 Create cards within lists
    - 1.2.3 Edit card details

2. **Task Creation and Tracking**

    2.1. Verify task creation process:
    - 2.1.1 Create a new task card with relevant details

    2.2. Verify task tracking:
    - 2.2.1 Add comments to cards

3. **Collaboration and Communication**

    3.1. Verify collaboration features:
    - 3.1.1 Add comments to cards
    - 3.1.2 Mention team members in comments
    - 3.1.3 invite member via email

Non-Functional tests

1. **Performance Testing**

1.1. Test system response under varying user loads:
- 1.1.1 Test with multiple concurrent get cards

1.2. Test system response under stress:
- 1.2.1 Test with high volumes of job applications and proposals
- 1.2.2 Test with heavy transaction loads during peak hours

1.3. Verify site load time:
- 1.3.1 Ensure the site loads in less than 5 seconds

2. Compatibility Testing

2.1. Test website compatibility across browsers:
- 2.1.1 Test on Google Chrome
- 2.1.2 Test on Mozilla Firefox

Localization and Globalization Testing

5.1. Verify multilingual support:
- 5.1.1 Change website language to hebrew

5.2. Test currency conversion:
- 5.2.1 Check currency display for different regions

4. Entry and Exit Criteria

4.1. Entry Criteria

- passing the sanity tests
- The requirement document should be available.
- Complete understanding of the application flow is required.
- The Test Plan Document should be ready.
- Test case/scripts are available
- Test environment is ready

4.2. Exit Criteria

- No critical test fail
- No high test fail

- No more than 3 medium test fail
- No more than 5 Low test fail
- The software meets all functional and non-functional requirements
- Approval from all relevant Product Manager has been obtained
- All defects and issues identified during testing have been resolved
- Documentation is complete and up-to-date

5. Test Strategy

5.1. Test Approach

The test strategy for the Trello website will focus on a comprehensive automation testing process, emphasizing both functional and non-functional aspects to ensure a seamless and user-friendly interface.

5.2. Test Categories

Functional Tests

- **Login Page**:Validate login functionality, including email/password login and social media login options.

- **Client Dashboard:**
  Test board creation, customization, and task management functionalities.
  Task Tracking: Verify task creation, assignment, status updates, and communication features.

Non-Functional Tests

Performance Testing: Evaluate system response, load handling, and site load time under varying conditions.

Compatibility Testing: Ensure the website performs consistently across different browsers and versions.

Non-Functional Tests

- **Website Compatibility**: The website's performance will be tested across a range of browsers to ensure consistent user experience.
- **Website Performance**

5.3. In Scope

Exploratory Testing: Initial testing phase focusing on key functionalities such as board creation, task management, and collaboration.
End-to-End Testing: Simulate user journeys from login to task completion, including collaboration and communication aspects.

5.4. Out of Scope

Performance Testing: Tests simulating extreme user loads and database overload conditions will not be conducted at this stage.
Localization and Globalization: Testing for support of multiple languages is not included in the current scope.
Recovery Testing: Session persistence and cart item retention after page reloads will be checked, but broader recovery testing will be deferred.

5.5. Test Execution

Test execution will be methodically documented, with automation test cases detailing the procedures, expected results, and observed outcomes to ensure thorough coverage.

6. Test Methodology

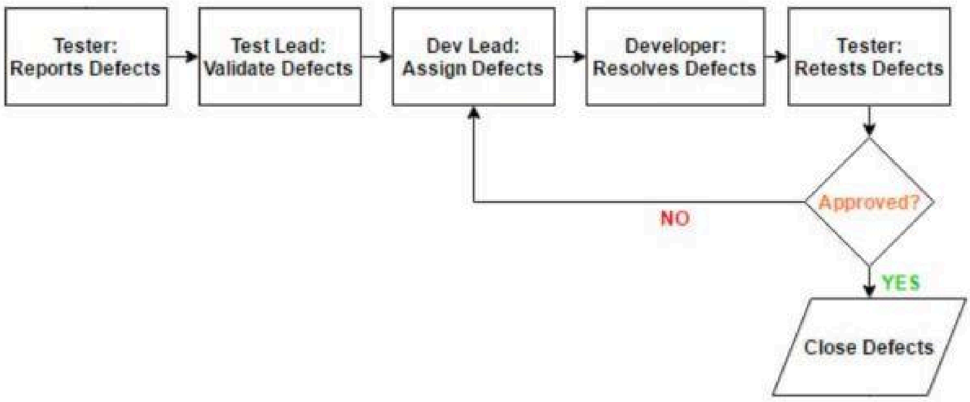
6.1. Validation and Defect Management

Defects found during the Testing will be categorized according to the bug-reporting tool “Jira” and the categories are:

Severity	Description
Critical	Bugs has the potential to crash the system
High	Bug significantly impacts program functionality
Medium	Bug prevents testing in specific areas of the product.
Low	Bug has minimal impact on product use .

Priority	Description
Critical	Bug needs immediate attention due to its severe impact.
High	Bug should be addressed promptly to restore functionality
Medium	Bug requires attention but can be managed within schedule .
Low	Bug can be deferred as it has minimal impact on workflow .

6.2. Defect tracking & ReportingDefect tracking & Reporting



6.3. Status of the Bug/Fault

- New: Just created bug
- Open: Opened bug and still not solved
- Rejected: Dev Lead rejected it (it's not bug)
- Fixed: SOLVED bug
- Closed: After bug is fixed, need close this bug
- Re-Open: Bug was been fixed and closed but appear again

6.4. Test Management Process

	Tool	**Comments**
Test Management	Test Tail	Test Cases Design
Bug Tracking	Jira/slack	Report bugs

6.5. **Assumptions for Test Execution**

- The test environment will remain stable and available throughout the test execution phase.
- Sufficient and representative test data will be available for executing test cases.
- Test resources, including testing tools and equipment, will be accessible and ready for use during test execution.
- Test preparation activities, such as test case design, documentation, and environment setup, will be completed before the test execution phase begins.
- Effective communication channels will be established and maintained throughout the test execution phase.

6.6. **Risk**

#	Risk	Impact	Trigger	Mitigation plan
1	Inadequate or unclear requirements.	High	can lead to incomplete test coverage.	Collaborate closely with Product managers to ensure comprehensive and clear requirements. Conduct regular reviews and validations.
2	Insufficient testing resources (human, hardware, or software) .	Medium	may lead to incomplete testing.	Plan resource needs in advance, prioritize testing tasks, and consider automation to optimize resource usage.
3	Testing in an environment that does not replicate the production environment.	High	lead to false positives or negatives.	ensure the test environment mirrors the production environment as closely as possible.
4	Poor communication among team members and Product Manager	medium	leading to ineffective testing.	Establish clear communication channels, conduct regular meetings, and document all communications to ensure everyone is on the same page.
5	Inadequate or unrealistic test data	medium	result in incomplete testing scenarios.	Develop a comprehensive test data strategy

7. Test Environments

- Operating System: Windows 11 OS Build 22631.3085
- Browser 1: Google Chrome Version 121.0.6167.140
    - All Extension are disabled
    - Cookies are allowed
- Browser 2: Firefox Version V 122.0
    - All Extension are disabled
    - Cookies are allowed

8. Time Table

Task	**Start Date**	**End Date**	**Comments**
STP Document	25.3.2024	25.3.2024	
STD Document	26.3.2024	26.3.2024	
Automation tests run	27.04.2024	27.04.2024	
STR Document	-	-	