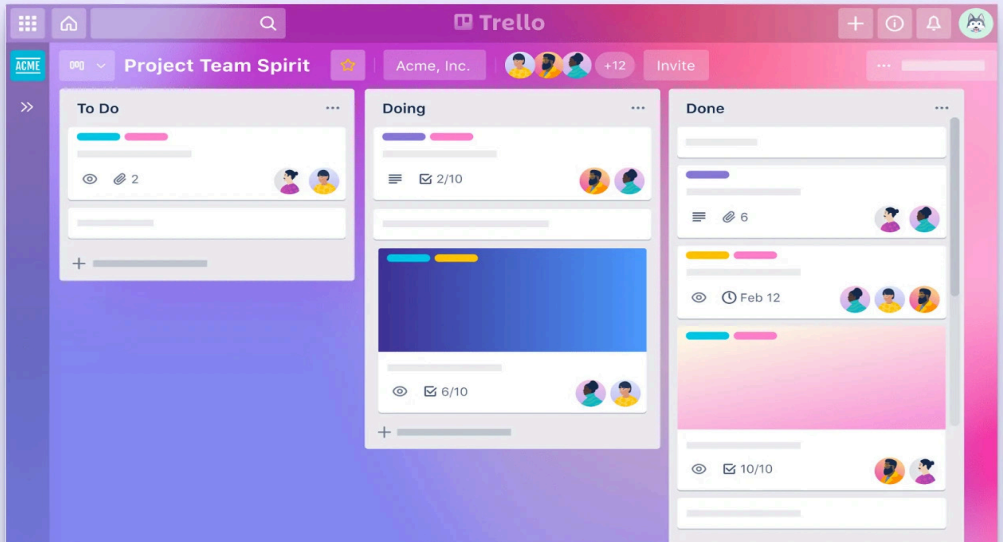
Software Test Plan

Trello

Tharaa Abu Saleh



CONTENTS

1. Introduction

a. Document overview

The Software Test Plan (STP) delineates the testing strategy, goals, and methodologies for Quality Assurance (QA) automation testing of the Trello platform. Trello is a renowned web-based application facilitating project management and collaboration through boards, lists, and cards.

b. Project overview

Trello serves as a pivotal tool for organizing tasks and projects, fostering collaboration among teams and individuals. The STP aims to ensure the seamless functionality of the Trello platform, providing users with an efficient and dependable project management experience. It delineates the testing scope, establishes testing objectives, and elucidates the testing approach to be employed.

c. project references

The purpose of this STP is to:

- Define the scope and objectives of QA automation testing for the Trello platform.
- Identify the roles and responsibilities of the testing team members.
- Outline the testing approach and methodologies to be employed.
- Specify the test environment, encompassing hardware, software, and network configurations pertinent to Trello automation testing.
- Define a timeline for the test design phase, ensuring timely completion of activities.

2. Test Strategy

a. Test Objectives

i. Validate the functionality of Trello UI elements and features.
ii. Verify the correctness and reliability of Trello API endpoints and responses.
iii. Ensure seamless integration and interoperability between UI and API layers.
iv. Detects and reports any inconsistencies or discrepancies between UI and API behaviours.

b. Test Assumption

i. Trello UI and API functionalities are expected to adhere to documented specifications and user expectations.
ii. The API responses are assumed to be consistent and reliable across different environments.
iii. The UI elements and API endpoints will maintain backward compatibility with existing features and integrations.
iv. Access to Trello boards, cards, and other resources is assumed to be consistent and uninterrupted during testing.

c. Test Types

i. UI Testing: Validate the visual elements, user interactions, and user experience of the Trello web interface.
ii. API Testing: Verify the functionality, performance, and security of Trello API endpoints.
iii. Integration Testing: Ensure seamless interaction and data flow between Trello UI and API layers.
iv. Regression Testing: Detect and prevent regressions in existing UI and API functionalities.
v. Compatibility Testing: Validate Trello UI and API across different browsers, devices, and platforms.

d. Test Approach

i. Automation: Utilise automation testing frameworks such as Selenium or Cypress for UI testing and tools like Postman or RestAssured for API testing.

ii. Parallel Testing: Execute UI and API tests in parallel to maximise test coverage and minimise test execution time.

iii. Continuous Integration: Integrate automated tests into the CI/CD pipeline to ensure frequent and reliable testing throughout the development lifecycle.

iv. Exploratory Testing: Supplement automation with exploratory testing to uncover usability issues and edge cases not covered by scripted tests.

e. Data Approach

i. Test Data Generation: Generate test data representing various Trello board configurations, cards, users, and interactions using tools or scripts.

ii. Data Management: Maintain separate test data sets for UI and API testing to ensure data isolation and integrity.

iii. Data Masking: Mask sensitive information in test data to comply with privacy regulations and security policies.

iv. Data Reusability: Reuse test data sets across different test scenarios to optimise testing efforts and ensure consistency.

f. Levels of Testing

The testing will be conducted at multiple levels, including:

- Unit Testing: Validate individual components and functions of the UI and API layers.
- Component Testing: Verify the integration and behaviour of UI components and API endpoints.
- System Testing: Testing the website as a whole to ensure it meets the specified requirements,end-to-end functionality and performance of Trello UI and API interactions.
- User Acceptance Testing (UAT): Testing with actual users to ensure the website meets their needs and expectations.

Test Deliverables

Deliverable Name	Author	Review
Test Plan	Test Lead	Project Manager
Functional and Non-Functional Test Cases and STD	Test Team	Test Lead
Bug Reports in Jira	Test Team	Test Lead and Developers Lead
STR	Test Lead	Project Manager

3. Execution Strategy

a. Entry and Exit Criteria

Entry Criteria for Trello UI and API automation testing:

1. Completion of test environment setup, including installation of necessary software and configuration of test data.
2. Availability of the latest build of Trello UI and API for testing.
3. Confirmation of test data readiness, including relevant test cases and datasets.
4. Completion of any prerequisite tasks or dependencies required for testing to commence.

Exit Criteria for Trello UI and API automation testing:

1. Successful execution of all planned test cases for both UI and API testing.
2. Adequate test coverage achieved based on predetermined metrics and targets.
3. Resolution of all critical defects impacting the stability or functionality of Trello UI and API.
4. Approval from stakeholders or project management for the completion of testing activities.

b. Validation and Defect Management

i. Validation Process:

1. Validate test results against expected outcomes and acceptance criteria.
2. Review logs, reports, and metrics to assess quality and reliability.
3. Seek feedback from stakeholders or end-users for usability validation.

ii. Defect Management:

1. Use a dedicated tracking system (e.g, Jira) to log and prioritise defects.
2. Assign severity and priority levels based on impact.
3. Implement systematic triaging, including root cause analysis.
4. Monitor resolution progress and verify fixes through retesting.
5. Communicate status updates to stakeholders regularly.

c. Defect tracking & Reporting

i. Defect Tracking:

1. Log defects promptly with detailed information.
2. Assign to responsible individuals or teams.
3. Track status throughout the defect lifecycle.

ii. Defect Reporting:

1. Generate regular defect reports with key metrics.
2. Include metrics like defect density and closure rate.
3. Provide reports tailored to stakeholders' needs.

4. SCHEDULE & ESTIMATION

Task	Objective	Expected Completion Date
Test Plan	Thorough test plan for validation.	19/03/2024
Test DesignTest Case and Development	Structured design, case creation, development.	20/03/2024
Test Execution phase 1: Sign In & Login	Ensure robust software functionality and performance.	20/03/2024
Test Execution phase 2: Board Creation	Successful Board Creation in Trello	21/03/2024
Tests	Functional and non functional tests	21/03/2024
Selenium grid - Parallel	Parallel test execution using Selenium Grid.	22/03/2024
Defect Management	Systematic tracking and resolution of defects.	23/03/2024
Test Report	Concise report detailing testing outcomes.	24/03/2024
presentation	Concise delivery of key information.	25/03/2024

5. TEST MANAGEMENT PROCESS

a. Test Management Tools

Utilize Trello as the primary test management tool for organizing and tracking UI and API automation testing tasks. Leverage Trello boards to represent different testing phases, such as test planning, test execution, and defect management. Utilize labels, checklists, and due dates to manage tasks effectively.

b. Role Expectations

Define clear roles and responsibilities within the test team. Assign a Test Lead responsible for overseeing the entire testing process, including test planning, task allocation, and coordination. Each team member should be assigned specific tasks related to UI and API automation testing, ensuring accountability and collaboration.

c. Project Management

Implement agile project management methodologies to facilitate iterative testing and quick feedback loops. Use techniques like Scrum or Kanban to manage the test backlog, plan sprints, and prioritize testing activities. Regular stand-up meetings and sprint reviews should be conducted to track progress and address any impediments.

d. Test Planning (Test Lead)

The Test Lead is responsible for creating a comprehensive test plan for UI and API automation testing. Define test objectives, scope, and entry/exit criteria for each testing phase. Identify test scenarios and prioritize them based on risk and business impact. Develop test strategies for UI and API automation, outlining tools, techniques, and environments required.

e. Test Team

The test team consists of automation engineers proficient in UI and API testing. Each team member should possess expertise in

relevant automation tools and frameworks such as Selenium for UI testing and Postman for API testing. Collaborate closely with developers, business analysts, and stakeholders to ensure alignment with project goals and requirements. Regular knowledge sharing sessions and skill development initiatives should be conducted to enhance the capabilities of the test team.

6. Test Environment

The test environment will mirror the production setup, aiming for parity in hardware, software, and network configurations.

7. Approvals

Name	Role	Signature
tzahi anidgar	Project Management	
tzahi anidgar	Test Lead	
-	Business Analyst	