SOFTWARE TEST DESCRIPTION

(STD)

Asana website

Nidaa AbuSaleh

TABLE OF CONTENTS

INTRODUCTION

1. Purpose:

The purpose of this Software Test Description (STD) is to delineate the methodology, coverage, resources, and schedule for conducting comprehensive QA automation tests on the Asana website for both UI and API. The STD aims to ensure that the website's UI and API functionalities meet predetermined quality standards, with an emphasis on identifying and effectively addressing any potential defects or issues.

2.scope:

The scope of the Asana website for UI and API automation testing includes:

- Validating user account management, task and project collaboration, workflow automation, intuitive UI/UX, integration with external tools, data visualization, communication features, user feedback mechanisms, customization options and platform compatibility.

For API testing, it encompasses:

- Ensuring endpoint functionality, data accuracy, documentation clarity, response time, compatibility, authentication and authorization, data format support, and documentation examples.

OVERVIEW

1.Background:

Asana is a comprehensive task management and collaboration platform designed to streamline project coordination and organization. It serves as a centralized digital workspace where teams can plan, track, and manage their work effectively. The platform facilitates task assignment, progress tracking, and communication, allowing teams to collaborate seamlessly and automate workflows. Asana provides users with intuitive features and customizable options to create a dynamic and interactive workspace. Its user-friendly interface and robust functionalities empower teams to increase productivity and achieve their project goals collaboratively.
The Asana API website, accessible at https://developers.asana.com, offers developers and enthusiasts programmable access to a wide range of features and functionalities provided by Asana. This API enables integration with various third-party tools and services, allowing developers to build custom solutions and automate workflows. With the Asana API, developers can retrieve and manipulate task, project, and user data, facilitating seamless integration and extending the platform's capabilities for diverse use cases and applications.

2. Goals:

Ensuring Seamless Functionality:

- Striving for a website that operates without errors, ensuring all features and functionalities work as intended.
- Conducting thorough testing to identify and rectify any potential issues, aiming for a smooth and reliable user experience.

Cross-Browser Compatibility:

- Verifying and optimizing the Asana website for compatibility across various web browsers.
- Ensuring consistent performance and functionality regardless of the browser used by the end user.

Ensuring API Reliability and Functionality:

- Striving for an API that operates seamlessly, providing accurate and reliable data to developers.
- Conducting comprehensive testing to identify and rectify any potential issues, ensuring a robust and dependable API experience.

Cross-Platform Compatibility:

- Verifying and optimizing the Asana API for compatibility across different programming languages and environments.
- Ensuring consistent performance and functionality regardless of the development environment used by developers.

Data Accuracy and Consistency:

- Ensuring the accuracy and consistency of data retrieved from various API endpoints to support reliable research and development efforts.
- Providing developers with dependable and up-to-date information through the API.

3.Glossary asana Website:

- Task Board:
  A visual representation of tasks and projects organized on a digital board, enabling users to manage and track work progress effectively.

- Automation:
  Rules and triggers set up within Asana to automate repetitive tasks, streamline workflows, and enhance overall efficiency.
- Dashboard:
  Customizable visual displays providing an overview of key metrics, project progress, and data relevant to users' specific needs.
- Collaboration Space:
  The interactive area within Asana where team members communicate, share files, and collaborate on tasks and projects in real-time.
- Integration:
  Connections with external tools and services that enhance Asana's functionality, enabling users to incorporate additional features and data sources.
- Timeline View:
  A feature allowing users to visualize project timelines and dependencies, facilitating effective project planning and management.
- Endpoint:
  A specific URL within the Asana API that allows interaction with a particular resource or functionality.
- Request:
  An HTTP request made to an API endpoint to retrieve, create, update, or delete data.
- Response:
  The data returned by the API in response to a request, containing relevant information or error messages.
- Authentication:
  The process of verifying the identity of users or applications accessing the Asana API, ensuring secure access to protected resources.
- Authorization:
  Controls and permissions determining the actions that authenticated users or applications can perform within the Asana API.
- Documentation:
  Comprehensive guides and reference materials providing information on how to use the Asana API, including endpoint descriptions, request and response formats, and authentication methods.

Test Cases

1. Valid Login:

UI Test:

- Enter valid credentials (username and password) into the login form.
- Click on the login button.
- Verify that the user is redirected to the dashboard or landing page.
- Check that the user's profile information (name, profile picture) is displayed correctly.

API Test:

- Send a POST request to the login endpoint with valid credentials.
- Verify that the response status code is 200 (OK).
- Confirm that the response contains an authentication token or session identifier.

2. Contact Management:

UI Test:

- Navigate to the CRM section.
- Fill in the required fields to create a new contact.
- Save the contact details.
- Verify that the contact details are correctly displayed.

API Test:

- Send a POST request to the endpoint for creating a new contact with relevant data.
- Verify that the response status code is 201 (Created).
- Check that the newly created contact exists in the database.

3. Dark Mode:

UI Test:

- Toggle the dark mode switch in the settings or preferences menu.
- Verify that the interface changes to dark mode.
- Toggle the switch again to disable dark mode and revert to light mode.

API Test:

- No API test applicable for this UI-specific feature.

4. Changing Priority:

UI Test:

- Open a task and change its priority level from low to high.
- Verify that the priority change is reflected in the UI immediately.

API Test:

- Send a PATCH request to the endpoint for updating task priority.
- Verify that the response status code is 200 (OK).
- Confirm that the task's priority is updated in the database.

5. Deleting a Single Task:

UI Test:

- Select a task from the task list or details view and delete it.
- Verify that the task is removed from the UI instantly.

API Test:

- Send a DELETE request to the endpoint for deleting a task.
- Verify that the response status code is 204 (No Content).
- Check that the task is no longer present in the database.

6. Deleting All Tasks:

UI Test:

- Navigate to the option to delete all tasks.
- Confirm the deletion action.
- Verify that the task board is empty.

API Test:

- Send a DELETE request to the endpoint for bulk task deletion.
- Verify that the response status code is 204 (No Content).
- Check that all tasks are deleted from the database.

7. Undo Functionality:

UI Test:

- Perform an action (e.g., task deletion).
- Click on the undo button immediately.
- Verify that the previous action is reverted.

API Test:

- No API test applicable for this UI-specific feature.

8. Search Functionality:

UI Test:

- Enter a search query into the search bar.
- Verify that relevant results are displayed.
- Check for responsiveness when using multiple search criteria.

API Test:

- Send a GET request to the search endpoint with the search query as a parameter.
- Verify that the response status code is 200 (OK).
- Confirm that the response contains relevant search results.

9. Filtering Tasks:

UI Test:

- Apply filters (e.g., by priority, due date) to the task list.
- Verify that only tasks meeting the specified criteria are displayed.
- Check for responsiveness when applying multiple filters.

API Test:

- Send a GET request to the endpoint for filtering tasks with relevant parameters.
- Verify that the response status code is 200 (OK).
- Confirm that the response contains tasks meeting the specified criteria.

10. Exporting Data:

UI Test:

- Navigate to the option to export task or contact data.
- Choose the desired file format and export the data.
- Verify that the exported file contains accurate and complete information.

API Test:

- Send a GET request to the endpoint for exporting data with relevant parameters.
- Verify that the response status code is 200 (OK).
- Check that the exported file matches the data displayed in the UI.

Traceability table

Test Case Description	Test Steps	Test Data	Expected Results	Related Requirements
Valid Login	UI: Enter credentials, click login, verify redirection. API: Send POST request, verify response and token.	Valid credentials	Redirect to dashboard. Profile info displayed.	Authentication
Contact Management	UI: Navigate to CRM, create and save contact. API: Send POST request, verify contact creation.	Contact details (name, email)	Contact created with accurate details.	CRM Functionality
Dark Mode	UI: Toggle dark mode switch, verify mode change.	-	Dark mode enabled/disabled as expected.	User Preferences

Changing Priority	UI: Change task priority, verify UI update. API: Send PATCH request, verify priority update.	Task with low priority	UI reflects priority change accurately.	Task Management
Deleting a Single Task	UI: Select and delete task, verify removal. API: Send DELETE request, confirm deletion.	Task to be deleted	Task is removed from UI.	Task Management
Deleting All Tasks	UI: Navigate to delete all tasks, confirm deletion. API: Send DELETE request, confirm task absence.	-	Task board is empty after deletion.	Task Management
Undo Functionality	UI: Perform action, click undo, verify reversal.	-	Previous action is undone successfully.	-

Search Functionality	UI: Enter search query, verify results display. API: Send GET request with query, confirm search results.	Search query	Relevant search results are shown.	-
Filtering Tasks	UI: Apply filters, verify task display. API: Send GET request with filter params, confirm filtered tasks.	Filter parameters	Filtered tasks displayed accurately.	-
Exporting Data	UI: Navigate to export, choose format, verify data. API: Send GET request, verify exported data integrity.	Export format	Data in exported files matches the UI display.	-