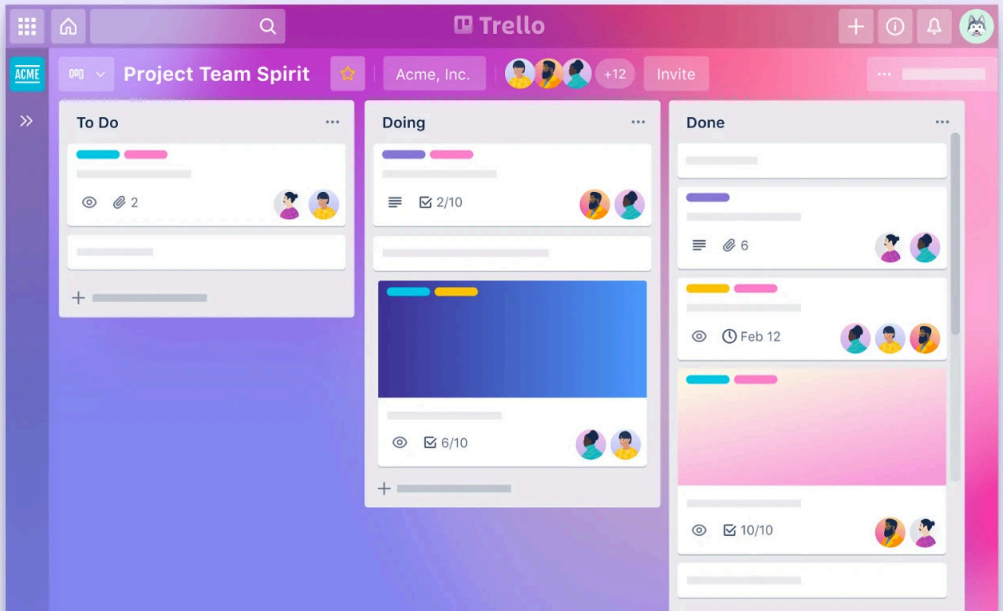
Software Test Design

Trello

Tharaa Abu Saleh



CONTENTS

1) Introduction

a) Document overview

The Software Test Design (STD) outlines the approach, objectives, and techniques for designing test cases and scenarios for the Trello platform. Test design is a critical phase in the software testing process, ensuring comprehensive coverage of functional and non-functional requirements.

b) Project overview

Effective test design is essential for ensuring the reliability and functionality of the Trello platform. The STD aims to establish a structured framework for creating test cases that validate the platform's features and behaviors. It defines the scope of test design, identifies test objectives, and outlines the methodologies and techniques to be utilized for crafting effective test scenarios.

c) project references

The purpose of this STP is to:

- Define the scope and objectives of test design for the Trello platform.
- Outline the approach and methodologies for crafting effective test scenarios and cases.
- Specify the tools and techniques to be utilized in the test design process tailored to Trello automation testing.
- Detail the test environment requirements essential for conducting test design activities effectively.
- Establish deliverables, including comprehensive test cases, scenarios, and test design documentation.
- Identify potential risks associated with test design and propose mitigation strategies.

2) Tests basic preparations

a) Hardware basic preparation

i) Assess the hardware requirements for running the automation testing framework and any necessary tools.

ii) Ensure that the hardware infrastructure meets the performance and compatibility needs of the automation testing environment.

iii) Procure any necessary hardware components such as servers, computers, or devices for running tests.

iv) Test the automation scripts on different hardware configurations to ensure compatibility and performance consistency.

b) Software basic preparation

i) Identify the software dependencies required for automation testing of Trello.

ii) Install and configure the necessary software components such as the automation testing framework, Trello API libraries, and any other supporting tools.

iii) Set up virtual environments or containers to isolate the testing environment if needed.

iv) Ensure that the software versions used in testing are compatible with the Trello API and any other integrated services.

v) Install and configure the automation testing framework or tools such as Selenium, Selenium Grid, and PyCharm for Trello automation testing.

vi) Web Browsers: Install the latest versions of popular web browsers (e.g., Chrome, Firefox, Safari) to ensure compatibility testing. IMDb should function correctly across different browsers.

vii) Documentation Tools: Use documentation tools (e.g., Microsoft Word, Google Docs) to create and maintain test plans, test cases, and other testing documentation.

c) Data preparation

i) Identify the types of data and scenarios relevant to Trello automation testing.

ii) Prepare test data representing various Trello board configurations, cards, users, and interactions.

iii) Cleanse and preprocess the test data to ensure its quality and integrity.

iv) Ensure that test data includes scenarios covering different user roles, permissions, and edge cases.

v) Establish data management processes for handling test data, including storage, backup, and versioning.

vi) Create data generation scripts or tools to automate the process of generating test data for Trello boards.

3) Test scenarios

a) Functional

i) Login

(1) Login with valid Trello Account

(2) Login with invalid Trello Account

ii) Trello Workspace

(1)Search

(a)Search for a board title

(2)Share

iii) Your Boards

(1) Create Board

(a) Add a list

(b) Add a card

(c) Edit dates to card

(d) Move a card to doing

(e) Remove list

(f) Remove card

(2) Calendar

(a) Adding date

(b) Change month

(3) Timeline

(a) Add a card

(b) Remove a card

(4) Filter

b) Non functional

i) personal information

(1) Change user name

(2) Update the bio

ii) Language & Region

(1) Change language

iii) Location

(1) Add location to card

iv) Change background

v) Templates

(1) Change templates board

4) Test cases

a) Test Case Title: Verify Successful Login with valid Trello Account

i) Preconditions:

(1) The user has a valid Trello account.

ii) Steps:

(1) Click on the "Login" button.
(2) Enter the valid Trello account email address.
(3) Click on the "Continue" button.
(4) Enter the corresponding valid password.
(5) Click on the "LogIn" button.
(6) Verify that the user is successfully logged in.
(7) Confirm that the user's profile information is displayed correctly.

Expected Result: User successfully signs in, is redirected to the Trello dashboard with accurate profile information, and can log out successfully.

b) Test Case Title:Verify Successful Board Creation in Trello

i) Preconditions:

(1) The user has access to Trello.
(2) User is logged in to their Trello account.

ii) Steps:

(1) Click on the "Create new board" button.
(2) Enter a title for the new board in the designated field.
(3) Provide a description for the board.
(4) Click on the "Create" button to initiate the board creation process.
(5) Verify that the new board is successfully created and displayed in the user's board list.
(6) Confirm that the title, description and privacy settings of the new board are accurate.
(7) Ensure that the user can interact with the board, adding lists, cards, and other elements as needed.

Expected Result: User successfully signs in, is redirected to the Trello dashboard with accurate profile information, and can log out successfully.

c) Test Case Title: Share a Board in Trello Workspace with UI and API Assertion

i) Preconditions:

(1) The user has access to Trello Workspace.

(2) User is logged in to their Trello account.

ii) Steps:

(1) UI Interaction:

(a) Navigate to the Trello Workspace.

(b) Locate the board that needs to be shared.

(c) Find the option to share the board with others.

(d) Click on the option to share the board.

(e) Enter the email addresses or usernames of the users to share the board with.

(f) Confirm the sharing action.

(2) API Assertion:

(a) Send a POST request to the Trello API endpoint to share the board.

(b) Provide the necessary parameters such as board ID and user details.

(c) Authenticate the API request using appropriate credentials or tokens.

(d) Verify that the response indicates successful sharing of the board.

Expected Result:

1. UI Interaction:
    a. The board is successfully shared with the specified users through the UI.
2. API Assertion:
    a. The API response confirms successful sharing of the board.
    b. If the assertions fail, an error or failure message is generated, indicating the discrepancies found.
    c. If the assertions pass, a success message is generated, confirming the correctness of the response attributes.

d) If the assertions pass, a success message is generated, confirming the correctness of the response attributes.Test Case Title: Search for a Board Title in Trello Workspace with UI and API Assertion
    i) Preconditions:
        (1) The user has access to Trello Workspace.
        (2) User is logged in to their Trello account.
    ii) Steps:
        (1) UI Interaction:
            (a) Navigate to the Trello Workspace.
            (b) Locate the search bar on the Trello dashboard.
            (c) Enter the desired board title in the search bar.
            (d) Initiate the search action by pressing the Enter key or clicking on the search button.
            (e) Verify that the search results are displayed on the UI.
        (2) API Assertion:
            (a) Send a GET request to the Trello API endpoint for searching boards.
            (b) Provide the search query parameter with the desired board title.
            (c) Authenticate the API request using appropriate credentials or tokens.
            (d) Verify that the response meets the expected criteria based on the assertions.

Expected Result:

3. UI Interaction:
    a. The search results for the specified board title are displayed on the UI.
4. API Assertion:
    a. If the assertions fail, an error or failure message is generated, indicating the discrepancies found, and bug in jira.
    b. If the assertions pass, a success message is generated, confirming the correctness of the response attributes.

e) Test Case Title: Remove List from Trello Workspace with UI and API Assertion
   i) Preconditions:
      (1) The user has access to Trello Workspace.
      (2) User is logged in to their Trello account.
      (3) There exists at least one board with lists and cards available.
   ii) Steps:
      (1) Remove List:
         (a) Navigate to the Trello Workspace.
         (b) Locate the board containing the list to be removed.
         (c) Identify the list to be removed.
         (d) Note down the name or identifier of the list for API assertion purposes.
         (e) Click on the list to open it.
      (2) UI Interaction:
         (a) Find the option to delete or archive the list.
         (b) Click on the option to delete or archive the list.
         (c) Confirm the action when prompted.
      (3) API Assertion:
         (a) Send a GET request to the Trello API endpoint to retrieve board information.
         (b) Implement assertions to validate that the removed list is no longer present in the board information retrieved from the API.

   Expected Result:

   5. UI Interaction:
      a. The list is successfully removed from the board on the Trello UI.
   6. API Assertion:
      a. The API request returns a successful response.

f) Test Case Title: Add Date to Calendar in Trello Workspace with UI and API Assertion
   i) Preconditions:
      (1) The user is logged in to Trello Workspace.
      (2) A board with a calendar feature exists.
   ii) Steps:
      (1) UI Interaction:
         (a) Navigate to the board with the calendar feature.
         (b) Access the calendar view.
         (c) Click on the date to which you want to add an event.
         (d) Enter event details such as title and time.
         (e) Save the event.
      (2) API Assertion:
         (a) Send a GET request to the Trello API to retrieve calendar events.
         (b) Verify that the added event is present in the API response.

Expected Result:

7. UI Interaction:
   a. The event is successfully added to the selected date on the calendar.
8. API Assertion:
   a. The API response contains the added event.

5) Requirements Traceability

ID	Main Requirements	Description	Ass. ID	Test Case	Purpose
1	logIn	Log in with a valid Trello account	TC_SI01	Verify Successful Login with valid Trello Account	Profile information displayed accurately.
2	CreateBoard	Create a new board in Trello	TC_SI02	Verify Successful Board Creation in Trello	Creation of a new board in Trello with accurate details and functionalities.
3	SearchBoard	Search for a board title in Trello Workspace	TC_SI03	Search for a Board Title in Trello	Search for a board title in Trello Workspace through both UI and API, with accurate display of results.
4	RemoveList	Remove a list from Trello Workspace	TC_SI04	Remove List from Trello Workspace with UI and API Assertion	Removal of a list from Trello Workspace via UI and API, maintaining consistency.
5	AddDate	Add date to calendar in Trello Workspace	TC_SI05	Add Date to Calendar in Trello	Addition of a date to the calendar in Trello Workspace through UI and API, ensuring accuracy in both methods.

6	UpdatePersonalInfo	Update personal information in Trello Workspace	TC_SI06	Update Personal Information in Trello Workspace with UI and API Assertion	Validate successful update of personal information in Trello Workspace via UI and API. ensuring consistency.
7	ShareBoard	Share a board in Trello Workspace	TC_SI07	Share a Board in Trello Workspace with UI and API Assertion	Sharing of a board in Trello Workspace through UI and API, maintaining consistency between methods.