Insights Report - 20250225.1809

This report provides insights into failed test cases and recommendations for improvement.

Overview

* The overall failure rate across all test executions was found to be 17.65% (3 of 17 test executions).
* 'High' severity errors accounted for 100% of the total errors (3 of 3 failed test executions).
* Test set [CRM:151](https://cloud.uipath.com/hpcghprtnym/HPCgTenant/testmanager_/CRM/testsets/CRM:151#) had a failure rate of 25% (1 out of 4 test set executions).
* Test set [CRM:8](https://cloud.uipath.com/hpcghprtnym/HPCgTenant/testmanager_/CRM/testsets/CRM:8#) had a failure rate of 33.33% (1 out of 3 test set executions).
* 'nestor demo-unattended' had 100% of the total test execution failures (3 of 3 failed test executions).
* 100% of total failed test executions were automated (3 of 3 failed test executions).

Top Failing Tests

* [CRM:7](https://cloud.uipath.com/hpcghprtnym/HPCgTenant/testmanager_/CRM/testcases/CRM:7?resultView=Chart#) – 'Funciones de Acceso Exitosas' had a failure rate of 28.57% (2 of 7 test executions).
* [CRM:76](https://cloud.uipath.com/hpcghprtnym/HPCgTenant/testmanager_/CRM/testcases/CRM:76?resultView=Chart#) – 'Credenciales de usuario Incorrectas' had a failure rate of 14.29% (1 of 7 test executions).

Common Errors

* Could not find the user-interface (UI) element required for the action.

Error Patterns

* UI/UX errors accounted for 100% of all failed test executions (3 of 3 failed test executions).

Recommendations

* Check for any dynamic content loading issues that might delay the rendering of the 'email-id' input field.
* Conduct a thorough review of the UI changes and update the test scripts accordingly to reflect the latest UI structure.
* Ensure that the 'email-id' input field is not being dynamically generated with different IDs in different sessions.
* Implement a fallback mechanism to locate the 'email-id' input field using alternative attributes like name or class.
* Implement a retry mechanism in the test scripts to handle transient issues with UI element loading.
* Leverage a self-healing test automation framework that can automatically update locators when UI changes are detected.
* Update the test scripts to use CSS selectors or XPath for locating the 'email-id' input field.
* Use a headless browser to speed up the test execution and reduce the chances of timing-related issues.
* Verify that the 'email-id' input field has a unique and consistent identifier across all environments.
* Add explicit waits in the test scripts to ensure the 'Sign In' button is fully loaded before interaction.
* Ensure the 'Sign In' button has a unique and consistent identifier across all environments.
* Ensure the 'Sign In' button is not hidden or disabled by any CSS or JavaScript during the test execution.
* Implement a custom script to validate the presence of critical UI elements before running the main test cases.
* Use a visual testing tool to verify the presence and appearance of the 'Sign In' button.
* Use AI-based element locators that adapt to minor changes in the UI structure.