System Testing Plan

CSE 361 - Spring 2014

**Regression Testing System**

|  |  |
| --- | --- |
| Authors:  Alex Tobias  James Duin | Supervisor:  Dr. Greg Rothermel |

April 11, 2014

**Introduction**

The purpose of this document is to outline the procedures we went through in order to test our software system. It will outline all of the requirements that we tested, and the methods we used to test each individual requirement. We will also go through and specify the test cases we created, how each case can be executed, and which requirement the test case covers. Finally, at the end of the document, we will present a testing report, which will describe the results we obtained from running each of our test cases.

**Testing Requirements**

This section includes the testing requirements which will be referenced in the test case tables. This serves the purpose of tracing back requirements that a particular test case covers.

Test Req 1.1:

REQ-1.1: The system shall allow a user to save individual scripts in Sahi code via a Scripts page.

REQ-1.2: Previously saved scripts should be visible in a list on the Scripts page

REQ-1.3: The system shall allow a user to save a test schedule for a single Sahi script on IE9/Win7.

REQ-1.4: Previously saved schedules should be visible in a list on the Schedules page

REQ-1.5: The system shall execute the saved test schedule for IE9/Win7 and post the test results to the report page.

REQ-1.6: The system shall allow the user to navigate between the Scripts page, the Suites page, the Test Schedule page, and the Reports page via a header ribbon.

REQ-1.7: The system shall validate test script names, and they will not contain spaces

**Test Cases**

This section includes our test cases. The tables include information on how to execute the tests and the testing requirements covered by each test.

|  |  |
| --- | --- |
| **ID:** | **Webapp\_1** |
| **Title:** | **Save a Script** |
| **Requirements Coverage:** | **REQ-1.1, REQ-1.2, REQ-1.6** |
| **Steps:** | **1) Click the Scripts link in the header ribbon to navigate to the Scripts page.**  **2) Type in invalidLogin for the script name.**  **3) Set the script type as Sahi.**  **4) Copy the Sahi test script code and paste it into the Code text area.**  **5) Click Save.** |
| **Expected Results:** | **The Existing Scripts multi-select shows the newly created script. Furthermore a corresponding entry in the TestCases table in the database should be made for the newly saved script.** |

|  |  |
| --- | --- |
| **ID:** | **Webapp\_2** |
| **Title:** | **Save a Test Schedule** |
| **Requirements Coverage:** | **REQ-1.3, REQ-1.4, REQ-1.6** |
| **Steps:** | **Setup: An invalidLogin script must be in the database.**  **1) Click the Test Schedule link in the header ribbon to navigate to the TestSchedule page.**  **2) Type Invalid Login in the Test Schedule Name textbox.**  **3) Select IE9/Win7 in the Browser/OS dropdown.**  **4) Select live in the Environment dropdown.**  **5) Select Sunday in the Weekly Schedule dropdown.**  **6) Select 10:00 as the Time of day.**  **7) Click Save.** |
| **Expected Results:** | **The Existing Test Schedules multi-select shows the newly created Test Schedule. A corresponding entry in the TestSchedules table in the database should be made for the newly saved Test Schedule. A corresponding entry in the PendingTests table in the database should be made for the newly saved Test Schedule.** |

|  |  |
| --- | --- |
| **ID:** | **Webapp\_3** |
| **Title:** | **Validate Test Case Name** |
| **Requirements**  **Coverage:** | **REQ-1.7** |
| **Steps:** | **1) Navigate to the test case page**  **2) Create a test case and include a space into the script name**  **3) Save the test case** |
| **Expected Results:** | **The test case will not save, and an error message will pop saying the test case name cannot contain a space.** |

|  |  |
| --- | --- |
| **ID:** | **System\_1** |
| **Title:** | **Confirm a ‘success’ Report post, when script is executed directly from jenkins.** |
| **Requirements Coverage:** | **REQ-1.5, REQ-1.6** |
| **Steps:** | **Setup: A script that you know will have success must be saved in the database at least 10 minutes prior to the test to ensure jenkins has enough time to update the the backend configuration.**  **1) Manually click build now on the testrunner backend process that was created for the script.** |
| **Expected Results:** | **Refresh the Reports page and confirm that a pass entry was made for test case.** |

|  |  |
| --- | --- |
| **ID:** | **System\_2** |
| **Title:** | **Submit multiple tests within a minute.** |
| **Requirements Coverage:** | **REQ-1.3, REQ-1.6** |
| **Steps** | **Setup: Have the invalidLogin test script previously stored in the database.**  **1) Create and submit a test schedule using the test script**  **2) Create and submit a second test schedule using the same test script, set the time to be the an hour later than the schedule in step 1** |
| **Expected Results:** | **System should use the test schedule that was submitted second, and disregard the first schedule. Confirm that the configuration in the backend uses the time submitted in the second test schedule. Confirm that both pending test schedule submissions get cleared from the PendingTests table in the database.** |

|  |  |
| --- | --- |
| **ID:** | **System\_3** |
| **Title:** | **Confirm a ‘fail’ Report post, when script is executed directly from jenkins.** |
| **Requirements Coverage:** | **REQ-1.5** |
| **Steps:** | **Setup: A script that you know will have a failure must be saved in the database at least 10 minutes prior to the test to ensure jenkins has enough time to update the the backend configuration.**  **1) Manually click build now on the testrunner backend process that was created for the script.** |
| **Expected Results:** | **Refresh the Reports page and confirm that a fail entry was made for test case.** |

|  |  |
| --- | --- |
| **ID:** | **System\_4** |
| **Title:** | **Confirm the run time posted to the report page matches the run time in the Sahi logs.** |
| **Requirements Coverage:** | **REQ-1.5** |
| **Steps:** | **Setup: A script that you know will have success must be saved in the database at least 10 minutes prior to the test to ensure jenkins has enough time to update the the backend configuration.**  **1) Manually click build now on the testrunner backend process that was created for the script.**  **2) Look at the sahi logs and record the script time.** |
| **Expected Results:** | **Refresh the Reports page and confirm that the run time matches the the time reported in the Sahi logs.** |

|  |  |
| --- | --- |
| **ID:** | **System\_5** |
| **Title:** | **Confirm the date and time information posted to the report page matches the date and time that was recorded in the Sahi logs** |
| **Requirements Coverage:** | **REQ-1.5** |
| **Steps:** | **Setup: A script that you know will have a failure must be saved in the database at least 10 minutes prior to the test to ensure jenkins has enough time to update the the backend configuration.**  **1) Manually click build now on the testrunner backend process that was created for the script.**  **2) Open the Sahi logs and record the date and time that the script was run.** |
| **Expected Results:** | **Refresh the Reports page and confirm that the date and time column for the test run record matches the the time in the Sahi logs.** |

|  |  |
| --- | --- |
| **ID:** | **System\_6** |
| **Title:** | **Confirm the script runs on the desired Browser/OS combination.** |
| **Requirements Coverage:** | **REQ-1.5** |
| **Steps:** | **Setup: A script that you know will have success must be saved in the database at least 10 minutes prior to the test to ensure jenkins has enough time to update the the backend configuration. The script must be specified to run on IE9/Win7.**  **1) Manually click build now on the testrunner backend process that was created for the script.**  **2) Watch the script run on the VM.**  **3) Look at the sahi logs and record browser that the script was run against.** |
| **Expected Results:** | **Confirm that the script opened and ran in an IE\_9 browser window on the Win7 VM. Confirm that the browser that was reported in the Sahi logs was IE9.** |

**Test Report**

This section describes the results obtained from running the tests. The tables indicate whether the test passed or failed. If a failure did occur, a description of the failure that was observed is provided.

|  |  |
| --- | --- |
| **ID:** | **Webapp\_1** |
| **Title:** | **Save a Script** |
| **Expected Results:** | **The Existing Scripts multi-select shows the newly created script. Furthermore the a corresponding entry in the TestCases table should be made for the newly saved script.** |
| **Pass/Fail:** | **Pass** |

|  |  |
| --- | --- |
| **ID:** | **Webapp\_2** |
| **Title:** | **Save a Test Schedule** |
| **Expected Results:** | **The Existing Test Schedules multi-select shows the newly created Test Schedule. A corresponding entry in the TestSchedules table in the database should be made for the newly saved Test Schedule. A corresponding entry in the PendingTests table in the database should be made for the newly saved Test Schedule.** |
| **Pass/Fail:** | **Fail, the TestCaseName field was null in the record saved for the TestSchedules table in the database. The PendingTests table did have the expected entry.** |

|  |  |
| --- | --- |
| **ID:** | **Webapp\_3** |
| **Title:** | **Validate Test Case Name** |
| **Expected**  **Results:** | **The test case will not save, and an error message will pop saying the test case name cannot contain a space.** |
| **Pass/Fail:** | **Fail, the webapp still allows the user to include spaces in test script.** |

|  |  |
| --- | --- |
| **ID:** | **System\_1** |
| **Title:** | **Confirm a ‘success’ Report post, when script is executed directly from jenkins.** |
| **Expected Results:** | **Refresh the Reports page and confirm that a pass entry was made for test case.** |
| **Pass/Fail:** | **Pass** |

|  |  |
| --- | --- |
| **ID:** | **System\_2** |
| **Title:** | **Submit multiple tests within a minute.** |
| **Expected Results:** | **System should use the test schedule that was submitted second, and disregard the first schedule. Confirm that the configuration in the backend uses the time submitted in the second test schedule. Confirm that both pending test schedule submissions get cleared from the PendingTests table in the database.** |
| **Pass/Fail:** | **Pass** |

|  |  |
| --- | --- |
| **ID:** | **System\_3** |
| **Title:** | **Confirm a ‘fail’ Report post, when script is executed directly from jenkins.** |
| **Expected Results:** | **Refresh the Reports page and confirm that a fail entry was made for test case.** |
| **Pass/Fail:** | **Pass** |

|  |  |
| --- | --- |
| **ID:** | **System\_4** |
| **Title:** | **Confirm the run time posted to the report page matches the run time in the Sahi logs.** |
| **Expected Results:** | **Refresh the Reports page and confirm that the run time matches the the time reported in the Sahi logs.** |
| **Pass/Fail:** | **Pass** |

|  |  |
| --- | --- |
| **ID:** | **System\_5** |
| **Title:** | **Confirm the date and time information posted to the report page matches the date and time that was recorded in the Sahi logs** |
| **Expected Results:** | **Refresh the Reports page and confirm that the date and time column for the test run record matches the the time in the Sahi logs.** |
| **Pass/Fail:** | **Pass** |

|  |  |
| --- | --- |
| **ID:** | **System\_6** |
| **Title:** | **Confirm the script runs on the desired Browser/OS combination.** |
| **Expected Results:** | **Confirm that the script opened and ran in an IE\_9 browser window on the Win7 VM. Confirm that the browser that was reported in the Sahi logs was IE9.** |
| **Pass/Fail:** | **Pass** |