

Project 1 (ReqKeeper) Black Box Test Plan

Document Author(s): Prem K Subedi

Date: Summer 2017

Introduction

ReqKeeper program that involves requirements management systems which uses the idea of Finite State Machine (FSM) and implements state pattern algorithm to update the requirement state from one to another. The black box tests involves the testing of valid file format, tests whether the file exists or cannot be found, tests the valid commands as an inputs based on the current state for the requirements to get updated. Running the RequirementsTrackerGUI class and the .xml file containing reqs list starts the program.

Test ID	Description	Expected Results	Actual Results
Test 1: InvalidFileFormat Author: Prem Subedi	Preconditions: None Run RequirementsTrackerGUI Click Load, file browser opens, then choose the test-files/test-files and then select file named invalid.txt instead of .xml files to read. Check results Close GUI.	A pop-up dialogue box stating "Unable to load requirements file" is displayed.	A pop-up dialogue box stating "Unable to load requirements file" is displayed.
Test 2: requirementStates Author: Prem Subedi	Preconditions: None Run RequirementsTrackerGUI Click File and then click Load choose test- files from the browser, then choose filename "exp_req_all.xml" Click Load . Check results. Close GUI.	GUI view shows the requirements with reqId in all six different states, submitted, working, accepted, completed, verified and rejected states.	GUI view shows the requirements with reqId in all six different states, submitted, working, accepted, completed, verified and rejected states.
Test 3: invalidRequirement Id Author: Prem Subedi	Preconditions: None Click Add New Requirement Enter the requirement	1 Requirement is listed with it's requirement Id of 0.	1 Requirement is listed with it's requirement Id of 0.

	<p>summary as “System should change the requirement from submitted to accepted when user clicks Accept” and test Id as “SubmittedToAccepted”</p> <p>Click Add Requirement to List.</p> <p>Check results. Close GUI.</p>		
<p>Test 4: Save in validFile</p> <p>Author: Prem Subedi</p>	<p>Preconditions: Test 2 has passed. Click Save. Choose one of the proper file format or create a new file in the browser to save the requirements with .xml format.</p> <p>Check results. Close GUI.</p>	File is successfully written or saved with the requirements.	File is successfully written or saved with the requirements.
<p>Test 5: rejectedToSubmitted</p> <p>Author: Prem Subedi</p>	<p>Preconditions: Test 2 has passed.</p> <p>Select the requirement that is in rejected state from the list of requirements. Click Revise. Check result. Close GUI</p>	The requirement in the rejected state is updated to the submitted state.	The requirement in the rejected state is updated to the submitted state.

<p>Test 6: submittedToAccepted</p> <p>Author: Prem Subedi</p>	<p>Preconditions: Test 2 has passed.</p> <p>Select the requirement that is in submitted state from the list of requirements. Give the priority of 1. Click Accept.</p> <p>Check result. Close GUI.</p>	<p>The requirement in the submitted state is updated to the accepted state.</p>	<p>The requirement in the submitted state is updated to the accepted state.</p>
<p>Test 7: invalidCommand</p>	<p>Preconditions: Test 2 has passed.</p> <p>Select the requirement that is in submitted state from the list of requirements. Give the priority of 0. Click Accept.</p>	<p>A pop-up dialogue box stating, "Invalid command", is displayed.</p>	<p>A pop-up dialogue box stating, "Invalid command", is displayed.</p>

Document Revision History

Date	Author	Change Description
Summer 2017	Prem Subedi	Completed Black box test plan for Project 1 part 2.