Acceptance Tests

User Story Epic	User Story ID	Acceptance Test ID	Acceptance Tests	Critical		Test Result		Comments
				YES	NO	Accept	Reject	
User Management	1.1	1.1.1	Emily opens the system and sees a screen where she has an option to Sign Up	X				
		1.1.2	Emily wants to make a new account. She selects the button to sign up and is moved to a screen where she can create credentials	X				
		1.1.3	When she makes the account, there is confirmation of account creation		X			A confirmation is not critical for this use case, as long as the account is created and the user able to use the system.
		1.1.4	Emily is returned to the main screen where she can sign in with her newly created credentials	X				
		1.1.5	The sign in is successful	X				
	1.2	1.2.1	Emily is on the starting screen of the system and sees fields for her to enter account credentials	X				
		1.2.2	After entering her credentials (for a valid account), Emily selects to sign/log in to system	X				
		1.2.3	The sign in is successful and she enters the system	X				
	1.3	1.3.1	Emily is logged in to the system. She clicks a button to view saved items		X			
		1.3.2	She is shown a list of saved contributions made by her currently logged in account. The list items come with metadata		X			
	1.4	1.4.1	Emily is on the "Search" screen and has found the "Cisco Traffic Log" item for saved regex. She selects to "View info"		X			
		1.4.2	She is shown a screen which displays metadata about the regex, specifically including a "Created By" field. It shows which user saved the contribution.		X			
	1.5	1.5.1	Emily has a valid Telstra account which she is currently logged in with on her computer. She selects the button to "Sign In with Telstra SSO"		X			
		1.5.2	She is taken to the landing page of the system		X			
File Processing	2.1	2.1.1	Emily is on the landing page of the system. She selects to upload a log file	X				
		2.1.2	Emily is taken to a file explorer. She navigates to the data_source folder. Selects and then clicks open on the unstructured cisco_v1.log file	X				
		2.1.3	The cisco_v1.log file is opened and displayed on the screen in the system	X				
	2.2	2.2.1	Emily has a Cisco v4 Syslog file open and has built a regex for it. She selects to save that regex		X			
		2.2.2	Emily sees the popup and enters the details of the regex and its associated log. Vendor: Cisco, Log Type: Syslog, Version: v4, Documentation: https://cisco.com. She selects "Save"		X			
		2.2.3	Emily navigates to the "Search" page of the system. She types "Cisco v4" and clicks 'search by log file'.		X			
		2.2.4	Emily sees a list of results and sees the Cisco v4 log + its regex. She clicks to "view info" of this search result		X			
		2.2.5	She sees a window if regex metadata. She then selects to open the sample log file (which was the log file used at point of the regex's generation)		X			
	2.3	2.3.1	Emily is on the landing page of the system. She selects to upload a log file		X			

		2.3.2	Emily is taken to a file explorer. She navigates to the data_source folder. Selects and then clicks open on the structured costreesopendata.csv file		X			
		2.3.3	The costreesopendata.csv is opened and displayed on the screen in the system		X			
Normalising a File Format From New Vendor	3.1	3.1.1	Emily is currently building a regex for the Cisco v1 log file. She selects "Insert" next to the regex.	X				
		3.1.2	Emily is shown a screen for searching for a regex. From the "Recently Used" section, she selects the "IP Address" field.		X			
		3.1.3	She sees that the IP address regex is successfully appended to the main regex being built	X				
	3.2	3.2.1	Emily has built a regex in the regex builder screen and has a log file open. She is able to see the captured fields row by row because they are highlighted.	X				
	3.3	3.3.1	Emily is on the landing page of the system. She selects to "Upload a log file" and proceeds with user story 2.1 acceptance tests.		X			
		3.3.2	Emily sees an empty text field on the screen for writing regex into. She types a few characters into the textbox		X			
	3.4	3.4.1	Emily is in the regex builder screen. She conducts user story 3.1 acceptance tests.		X			
		3.4.2	Emily has made changes to the regex. She is able to see the results of the change (in this case, appended regex field) on the opened log file as per acceptance test 3.2.1.		X			
	3.5	3.5.1	Emily is currently building a regex for a Cisco v2 log file. She selects "Insert" next to the regex.		X			
		3.5.2	Emily is shown a screen for searching for a regex. In a search bar, she types "Cisco v1" and selects "search".		X			
		3.5.3	She sees the results of the search and selects the regex associated with Cisco v1 log files.		X			
		3.5.4	She sees that the entire regex from Cisco v1 is successfully appended to the regex currently being built		X			
	3.6	3.6.1	Emily is on the landing page of the system. She clicks "Search" on the top navigation bar		X			
		3.6.2	Emily types "Cisco" in the search box.		X			
		3.6.3	She can see the results of the search below. The results are a list of matching saved regular expressions.		X			
		3.6.4	She presses "View Info" besides the "Cisco System Log" result, and is then able to view the metadata associated with the regex for that log file		X			
	3.7	3.7.1	Emily has an unstructured log file open in the regex builder page. She highlights an IP address portion of text in the log file.		X			
		3.7.2	Emily sees a suggestion on screen based on the highlighted text. The suggestion says the highlighted text looks like an IP Address		X			
	3.8	3.8.1	Emily is on the landing page of the system. She selects "build a regex".		X			
		3.8.2	Emily sees the text field to type in a regular expression at the top of the screen. Outside the scope of the system, she copies a regex to the clipboard. She is able to paste this text into the bar.		X			
Comparison of new log file with existing regex	4.1	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder.					
		4.1.1	Emily clicks "Yes" on "Would You Like to Compare with Pre-Existing RegEx" and is rerouted to the RegEx search page and selects the Palo Alto Traffic Log.	X				
		4.1.2	When Emily clicks the "Continue to Build RegEx" button she is redirected to the live builder page	X				
		4.1.3	The previously selected regular expression is prefilled and the uploaded file showing in the text box	X				

	4.2	Prerequisite	Emily is logged in and has uploaded an unstructured log file. She has entered the Regex Builder page.					
		4.2.1	When Emily clicks "Yes" on "Would You Like to Compare with Pre-Existing RegEx", she is rerouted to the Search page	X				
		4.2.2	On the Search page, Emily can see all RegExs saved in the central repository	X				
	4.3	Prerequisite	Emily is logged in and has uploaded an unstructured log file. She has entered the Regex Builder page.					
		4.3.1	When Emily clicks on "Compare With Another Log", she is able to upload another file		X			
		4.3.2	When Emily selects a file, it is shown side by side with the original file		X			
	4.4	Prerequisite	Acceptance Tests for User Story 4.3 Complete					
		4.4.1	Emily clicks "Yes" on "Would You Like to Compare with Pre-Existing RegEx" and is rerouted to the RegEx search page		X			
		4.4.2	Emily selects the Palo Alto Traffic Log and sees it executed on both log files		X			
	4.5	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder.					
		4.5.1	Emily clicks "Yes" on "Would You Like to Compare with Pre-Existing RegEx" and is rerouted to the RegEx search page	X				
		4.5.2	Emily selects the Palo Alto Traffic Log	X				
		4.5.3	Emily Clicks on the Show Sample button, which brings up an overlay	X				
		4.5.4	The overlay shows the data sample that the existing RegEx was built open, complete with the fields that were captured	X				
	4.6	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder.					
		Prerequisite	Emily clicks "Yes" on "Would You Like to Compare with Pre-Existing RegEx" and is rerouted to the RegEx search page					
		Prerequisite	Emily selects the Palo Alto Traffic Log and sees it on the Regex Builder bar					
		4.6.1	Emily can see which fields were captured from each line	X				
		4.6.2	Emily can clearly see which lines were not captured at all in red	X				
		4.6.3	Emily can see how many lines were successfully captured	X				
	4.7	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder.					
		Prerequisite	Emily clicks "Yes" on "Would You Like to Compare with Pre-Existing RegEx" and is rerouted to the RegEx search page					
		4.7.1	Emily selects the Palo AltoTraffic Log regular expression. She is redirected to the RegEx builder. When she clicks the "View Documentation" button, she is redirected to the Palo Alto Log documentation site		X			
	4.8	Prerequisite	Emily is logged in and has uploaded a csv file and has entered the structured file page.					
		4.8.1	She clicks on the predict field names button. This should enable her to see predictions for each field in the structured file		X			
Saving Regular Expressions	5.1	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
		5.1.1	Emily writes a new regular expression into the bar	X				
		5.1.2	Emily presses the Save button, opening the Save overlay	X				
	5.2	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
		Prerequisite	Emily adds a new regular expression into the bar					

	5.2.1	Emily clicks on the save field button, opening the save overlay	X				
	5.2.2	Emily can see the Name text box, and adds the name of her regex into it	X				
	5.2.3	Emily presses save, saving the capture group	X				
5.3	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
	5.3.1	Emily presses the Insert button, redirecting her to the RegEx Search page	X				
	5.3.2	Emily adds "palo alto traffic log" into the search box. Emily should see the palo alto traffic log as the result	X				
5.4	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
	Prerequisite	Emily presses the Insert button, redirecting her to the RegEx Search page					
	5.4.1	Emily presses the log vendor button, showing Emily the log vendors stored in the system, such as Cisco and Linux	X				
5.5	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
	Prerequisite	Emily presses the Insert button, redirecting her to the RegEx Search page					
	5.5.1	Emily presses the log vendor button, showing Emily the log types stored in the system, such as Palo Alto Traffic Log	X				
5.6	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
	5.6.1	Emily adds a new field capture group into the bar	X				
	5.6.2	Emily presses the Save button, opening the Save overlay	X				
	5.6.3	Emily selects the "Capture Group" radio button	X				
	5.6.4	Emily adds a Field Capture Group Name into the provided textbox, and presses save. This should save the capture group	X				
5.7	Prerequisite	Continuing Test Case 5.1					
	5.7.1	Emily can successfully add a Log Vendor name into the provided text box	X				
	5.7.2	Emily can successfully add a Log Type name into the provided text box	X				
	5.7.3	Emily can successfully add a Log Version Number into the provided text box		X			
	5.7.4	Emily can add a Documentation Link in the provided Text Box		X			
5.8	Prerequisite	Continuing Test Case 5.7					
	5.8.1	Emily checks the "Save Sample File" box, which should save her sample file with her RegEx		X			
	5.8.2	Emily presses the Save button, saving the regular expression		X			
5.9	Prerequisite	Emily is logged in and has uploaded an unstructured file. She has entered the RegEx builder					
	Prerequisite	Emily presses the Insert button, redirecting her to the RegEx Search page					
	5.9.1	Emily adds a regular expression to the search box. This should filter all the regular expressions that partially match		X			