Assessment Evaluation

Organization-Wide Alert System

For

CS 895 MSE Project

Department of Computer Science

Kansas State University

Submitted to

Dr. Mitch Neilsen

Dr. Torben Amtoft

Dr. Scott Deloach

Submitted by

Angela Hall

5/5/2019

1. **Introduction**

This document serves as the testing documentation for the Organization-Wide Alert System. It lists the test cases, details the tests performed, and records the results.
I chose to use manual black-box test procedures to test the functionality of the WebGUI, Rest API, and Client.
I chose JMeter to test the load capacity of the Server. JMeter is a Java-based testing tool that simulates user input to put the server under load. I tested the application running on a Raspberry Pi 3 with up to 1500 simultaneous requests.

2. **Manual Black Box Test Results Summary**

Below is a summary of the tests and results. Each test is detailed in the next section of this document.

Client Application

Test Case	Description	Result
TC#1	System Tray Icon	Pass
TC#2	Alert Window	Pass
TC#3	Settings	Pass

WebGUI

Test Case	Description	Result
TC#4	Login	Pass
TC#5	Trigger Alarms	Pass
TC#6	Customize Application	Pass
TC#7	Notification Table	Pass

Rest API

Test Case	Description	Result
TC#8	Get Alarms	Pass
TC#9	Get Titles	Pass
TC#10	Trigger Alarm with GET	Pass
TC#11	Stop Alarms	Pass

Database

Test Case	Description	Results
TC#12	Database	Pass

3. Detailed Test Results

3.1. Manual Testing

TC#1 System Tray Icon

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
System Tray Icon	System Tray Icon appears in all versions of Windows (10, 7, and XP) when program is launched	A single instance of the system tray icon  will appear near the system clock	As expected	Pass
System Tray Icon	When right clicked, the System Tray Icon will have the following Menu Items: 1. Each of the 4 alarms 2. Settings 3. Exit	The system tray icon menu will pop up when right-clicked, and all menu items will load and update correctly	As expected	Pass
System Tray Icon	Clicking on System tray icon and then one of the alarm menu items, will trigger alarm	Alarm windows will pop up with correct color and message Alarm status in DB will change from 0 to 1	As expected	Pass
System Tray Icon	Clicking on Settings will launch settings window	Settings window will appear on the left side of screen.	As expected	Pass

		Settings window will display correct settings with data from the Config file.		
System Tray Icon	Clicking on Exit on the system tray icon will cause the program to close	Program will terminate correctly	As expected	Pass

TC#2 Alert Window

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Alert Window	Alert Window will pop up when alarm is triggered	Window becomes visible, color will be correct color, message will be correct message	As expected	Pass
Alert Window	If more than one alarm is triggered, Alert window will “stack”	First window will be at y offset of 130. Second window will be directly above that, third window will be directly above second, etc. All windows will be the correct colors to correspond with their alerts. All windows line up vertically	As Expected	Pass
Alert Window	Alert window will be Always-on-top	Alert Window will not go to background if other applications are being used	As Expected	Pass

TC# 3 Settings

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Settings	Settings will create config file if one is not present	Config file will be created with correct IP address and hostname	As expected	Pass
Settings	Write button will update existing config file	Changes made in the settings window will be reflected in config file after the write button in settings is pressed	As expected	Pass
Settings	Read config file into the textfields of the settings window	Changes made to settings will reload from config file when program is closed and reopened	As expected	Pass
Settings	Password will encrypt before writing Config File	The password stored in the config file will not be plane text	As expected	Pass

TC#4 WebGUI

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Login	At the login screen a good password will allow user to access send alarm page	SendAlarm page will load after using good credentials to log in	As expected	Pass
Login	At the login screen, using a	Using bad credentials, user	As expected	Pass

	bad password, users will not be able to access any of the other pages	should be redirected to Invalid Login page		

TC#5 Trigger Alarms

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Trigger Alarms	Logged in to WebGUI Alarm 0 trigger alarm 0	Alarm's status will be changed to 1 in the database. Client will register Alarm is active and display alert window RestAPI resource will display alarm in active alarms list	As expected	Pass
Trigger Alarms	Logged in to WebGUI Alarm 1 trigger alarm 1	Alarm's status will be changed to 1 in the database. Client will register Alarm is active and display alert window RestAPI resource will display alarm in active alarms list	As expected	Pass
Trigger Alarms	Logged in to WebGUI Alarm 2 trigger alarm 2	Alarm's status will be changed to 1 in the database.	As expected	Pass

		Client will register Alarm is active and display alert window RestAPI resource will display alarm in active alarms lis		
Trigger Alarms	Logged in to WebGUI Alarm 3 trigger alarm 3	Alarm's status will be changed to 1 in the database. Client will register Alarm is active and display alert window RestAPI resource will display alarm in active alarms lis	As expected	Pass
Trigger Alarms	Logged in to WebGUI Alarm 4 trigger alarm 4	Alarm's status will be changed to 1 in the database. Client will register Alarm is active and display alert window RestAPI resource will display alarm in active alarms lis	As expected	Pass

TC# 6 Customize Application

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail

Customize Application	Changes made to the Alarm Titles will propagate across system when saved	Changes made to alarm titles on the Customize Application page will be immediately reflected on the SendAlarm Page, and also in the Systray Icon of the Java Client on all PCs	As expected	Pass
Customize Application	Users Added in the AddUsers section will be added	Users will appear in the Edit Users table Users will Appear in the database User passwords will be encrypted in database Using new Username and Password on login screen will result in successful login Using new username and password in the client settings will result in successful authentication.	As expected	Pass
Customize Application	Password reset link beside username will result in password reset for given user	User will be able to use new password to log into website User will be able to use new	As expected	Pass

		password to authenticate with the Java client		
Customize Application	Delete user link beside username will remove user from the system	User will be deleted from database User will not be able to log into the system with deleted username User will not be able to authenticate from the client with given username	As expected	Pass

TC#7 Notifications Table

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Notifications Table	Notifications table will display correct time, user, ip, and message	When alarm is triggered from client notifications table will display correct alarm, computer name, and IP When alarm is triggered from server notifications table will not display IP and will say that alarm was triggered from server	As expected	Pass
Notifications	Notifications	20 results will	As expected	Pass

Table	Table will paginate correctly	show on each page. Pagination buttons will display the preceding or next 20 results.		

TC# 8 Get Alarms

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Get Alarms	Get Alarms resource will respond to all correctly formed calls with a list of currently active alarms	Browser and CURL will show XML formatted list of active alarms	As expected	Pass

TC# 9 Get Titles

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Get Titles	Get Titles resource will respond to all correctly formed calls with a list of all current alarm titles	Browser and CURL will show XML formatted list of current titles	As expected	Pass

TC#10 Trigger Alarm with GET

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Trigger Alarm with GET	GET call created with appropriate headers will authenticate user and trigger alarm	Using JMeter to create GET call with username and password headers added, when activateAlarm resource is	As expected	Pass

		called alarm status in database will change to 1 and client will register change and display alert window		

TC#11 Stop Alarms

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Trigger Alarm with GET	GET call created with appropriate headers will authenticate user and stop alarm	Using JMeter to create GET call with username and password headers added, when StopAlarms resource is called, alarms status in database will change to 0 and client will register change and stop displaying alert window	As expected	Pass

TC#12 Database

Test Unit	Test Detail	Expected Result	Actual Result	Pass/Fail
Database	Computers Table is updated when new computer joins the system	Computers table will show the hostname and IP of the new computer	As expected	Pass

3.2. **JMeter Testing**

The server under test is a Raspberry Pi 3 running an Apache web server hosting

the Organization-Wide Alert System, with php 7 and maria DB installed.

I started the JMeter testing with 1 simulated user to get a baseline of the system when under light load. I then increased the load to 50 users and ran several tests. I then increased the load to 300 users at 3 loops (1500 total requests).

I ran one Test Plan that simulated clients requesting Get Alarms and Get Titles at the same time. This request does not involve a database update and simulates normal usage where the client makes these requests simultaneously, every 8 seconds.

I ran another Test Plan that simulated users triggering an alert. I again ran a baseline test with one user, increased to 50, and then increased to 300 at 3 loops (1500 total requests). My goal was to simulate an active shooter event where everyone in a facility attempts to trigger the alarm at the same time.

Summary Report
Test Plan #1 - Simulating client requests

# of Samples	Average (ms)	Min (ms)	Max (ms)	Std. Dev.	Error %
1	7	2	18	0	0
50	7	2	18	1.45	0
1500	10	2	86	5.11	0
1500	26	3	739	738.86	0

I got varying results for the first Test Plan, however, even the worst case results were very good. In most of the test iterations the latency remained stable at under 100 milliseconds. In the worst case results, latency reached .7 seconds. I have included the table and graph results of the best and worst test iterations for the first Test Plan.

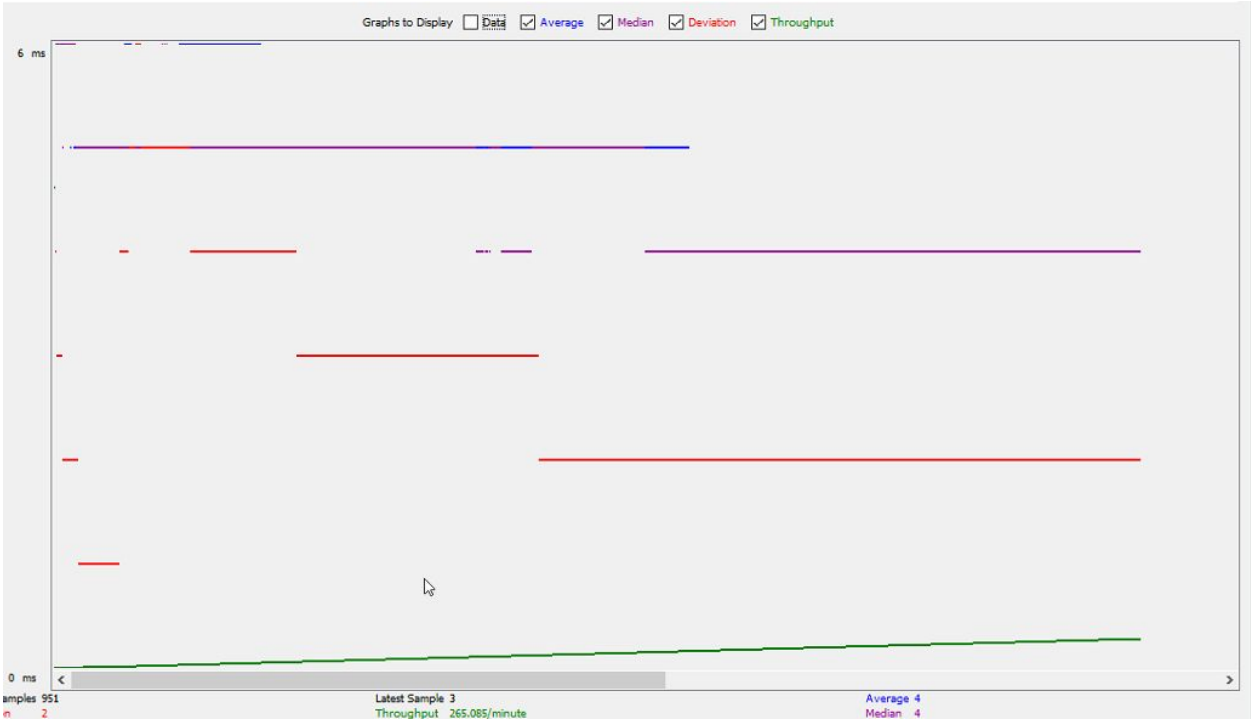
Fig 2.1 Best results - latency average remains consistently at 10ms

Fig 2.2 Worst results - latency average of 26 ms with a max of 738 ms

Test Plan #2 - Simulating Active Shooter event with 300 users

# of Samples	Avg (ms)	Min (ms)	Max (ms)	Std. Dev.	Error %
1	14	14	14	0	0
50	5	4	14	1.43	0
1500	4	3	41	2.25	0

I ran this Test Plan in 5 iterations each of 50 and 300 users. Results were nearly identical each time. Latency remained consistently low throughout the test. Alarms were successfully triggered by the GET requests and latency didn't increase appreciably.



4. **Overall Results**

The testing has been completed. All functionality of the Server, Client, Rest API, and Database have been confirmed to be working correctly.

The JMeter tests show that the Organization-Wide alert system running on a Raspberry Pi has sufficient resources to handle an organization of up 300 users quite easily and up to 1500 simultaneous requests without issues.

5. References

JMeter Tutorials: The Complete Free Training on JMeter (20+ Videos). (2019). Retrieved from https://www.softwaretestinghelp.com/jmeter-tutorials/

Jmeter.apache.org. (2019). *Apache JMeter - Apache JMeter™*. [online] Available at: https://jmeter.apache.org/ [Accessed 8 May 2019].

Karumuri, N. (2009). *KDD- Service Based Numerical Entity Search (KSNES).* Masters. Kansas State University.

Lastrapes, J. (2018). *RCRA Enforcement and Compliance History (REACH) System*. Masters. Kansas State University.