SOFTWARE TEST PLAN

for

Materials Ordering System

Release 1.0

Version 1.0 approved

Prepared by Baxter Halder, Claire Hall,
Kayla Jamar, Jennifer Olszyna

Contents

List of Figures

List of Tables

Revision History

Date	Description	Revised by
04/11/22	Initial draft	Data Store

1 INTRODUCTION

The Software test plan document explains the testing activities in order to deliver a quality product.The following sections explains details of the Data Store testing process.

1.1 Scope

The CS455 Monitor System software will monitor both the activity and health of all machines on a network. The system itself will be a combination of four sub-systems; Agents, a Monitoring Engine, a Data Store, and a Dashboard. The Engine and Data Store are hosted on the CS server. However, the Engine and Data Store will be portable to other server machines that run on a Unix-like OS. The Agents, installed on devices on the network, can run on both Windows and Unix-like systems. The Dashboard will display the status of the monitored machines. The Dashboard can be used on Windows or Ubuntu machines. Users will be able to view specific information such as CPU usage, memory usage, disk usage, and network usage of the monitored machines on the network from the Dashboard. This information passes through the Monitoring Engine to be validated, such as checking valid ranges for data, before being stored in the Data Store for the Dashboard to retrieve. Users will be able to view the specific machines the Agents are monitoring that are connected to the network in order to track down the cause of issues. Examples of these issues include a monitored machine being disconnected or unresponsive, a monitored machine's disk is close to or over capacity, a monitored machine's CPU usage is maxed out, which services should be running on the monitored machine that are not currently running, the network usage is less than the user expected for a monitored machine, or the monitored machine's memory usage is close to

capacity.

1.2 References

Standard	Reference
SEI CERT	SEI CERT Secure Coding Practices
Microsoft SDL	Microsoft SDL Secure Development Practices
SQL and PL/SQL Coding Standards	PL/SQL best practice Standards tips
Standard SQL Naming Conventions	Oracle naming standards tips
JSON	RFC 8259

Table 1.1: References

1.3 System Overview

The Software Test plan provides an organized blueprint on the process of testing the necessary requirements leading to a immaculate Data Store component. The document includes a summarized test schedule,responsibilities, and techniques used for preparing the testing trials. These trials will be assembled within the master test plan and will be be placed in certain levels based on their classification according to the integrity level scheme.

1.4 Organization

The team leader will acknowledge requirements and report progress of the project. The software engineer will proceed in developing those requirements using accurate programming and tools. Once requirements are met, the testing lead will be responsible for confirming that the requirements were encountered and efficient. If an error is detected, the testing lead will warn the following members and allow the testing engineer to record the

test results and provide recommendations to improve the software. The software quality assurance engineer will be aware of each of these phases for documentation purposes of risk and results.

1.5 Master test schedule

The first component listens to the Engine and then receives either error messages or JSON files containing information from the Agents. The second component is the database itself that will implemented as a MariaDB database hosted on the CS server. The database's schema is depicted in Figure 1.1 as an E/R diagram. It is assumed by the Data Store that the data is valid, so once the file is received the Data Store will parse out the information contained in the file and insert it into the *devices* entity or parse and insert the error message details into the *error_log* entity. A database entity is a thing, person, place, unit, object or any element data should be stored in the form of properties. The Data Store's database will be comprised of two entities. *devices*, and the *error_log*. Each entry in the *devices* entity represents a device with an Agent installed on it. The last component will receive confirmation from dashboard to send either queries or error messages. Error messages are processed the same way ones from the Engine are, and quires are transforms into SQL quires and processed. The specific test are shown below in the Master Test plan.

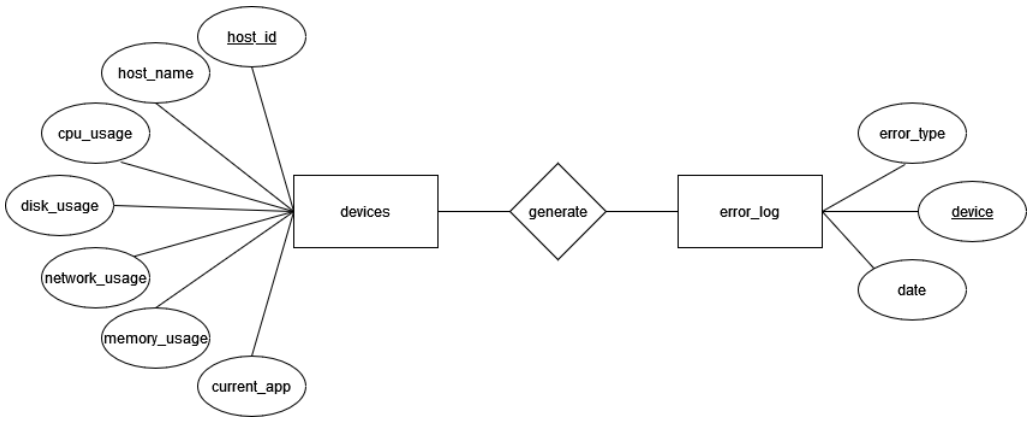

Figure 1.1: Database Design Depicted as an E/R Diagram

1.6 Software integrity levels

Table 1.2: Software integrity levels

Integrity level identifier	Description
Level 4	Catastrophic
Level 3	Critical
Level 2	Marginal
Level 1	Negligible

An integrity level scheme is used to provide a structured classification mechanism for denoting overall breadth and depth of testing for each testable portion of the system. Integrity levels may be applied to a variety of items including requirements, functions, classes or collections of functions, modules, subsystems, or whole systems. Table 1.2 shows the software integrity level scheme that will be used to denote the minimum required testing tasks for a testable unit. Each testable unit will be assigned one of the four integrity levels shown. The required amount of testing is dictated by the level

with Level 4 indicating exhaustive testing should be performed to ensure correctness and Level 1 testing indicating that basic functional is required. The scheme is drawn from the IEEE Standard for Software and System Test Document (IEEE 829-2008).

1.7 Resources summary

Functional PC's will be needed to access the following information. The PC will require a reliable server and browser for completing all tests. The standard Windows and Linux OS terminal is specifically required.Detailed information on tools,methods and metrics can be found in 1.3 table.

1.8 Responsibilities

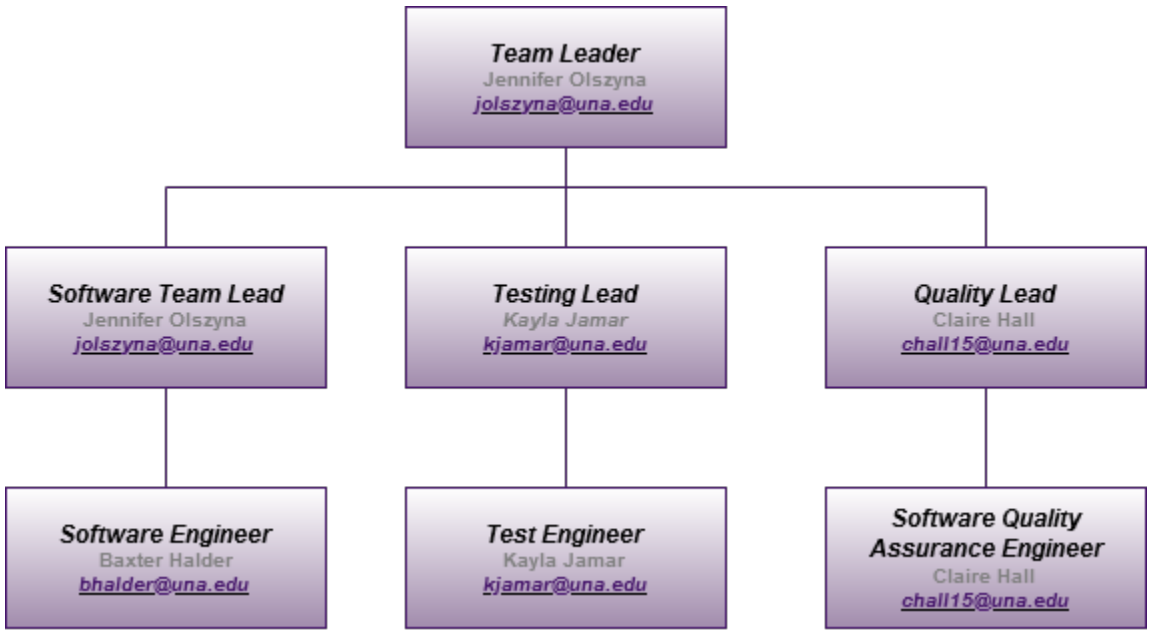

Figure 1.2: Responsibility Chart

1.9 Tools, techniques, methods, and metrics

Tool	**Purpose**
XAMMP	cross-platorm web server used for testing MYSQL
MariaDB	open-source GNU General Public License community-developed, commercially supported fork of the MySQL relational database management system

Table 1.3: Tools, techniques, methods, and metrics

2 MASTER TEST PLAN

2.1 Level 4 test plan

2.1.1 Traceability matrix

ID	Requirement	System Component	Approach
SR01	Engine can create and send a network message to the Data Store	Database Tables Storing Received Messages	Black box
SR02	Data store validates requests from Dashboard	Database Table Constraints	Analysis
SR06	Data Store sends data to the Dashboard	SQL Queries	Analysis
SR27	Data store can hold data for at least a year	Analyze Table, Check Table, Optimize Table, Repair Table, and Get Rows Count Queries	White box
QR7	System can be ported to other devices	Configuration File	Analysis

Table 2.1: Level 4 Matrix Table

2.1.2 Pass/Fail criteria

Level 4 must successfully complete all test cases.

2.1.3 Test deliverables

All test results will be recorded in excel spread sheet

2.2 Level 3 test plan

2.2.1 Traceability matrix

ID	Requirement	System Component	Approach
SR13	Users can view CPU usage	Device Table Storing CPU Usage, SQL Queries	White box
SR14	Users can view memory usage	Device Table Storing Memory Usage, SQL Queries	White box
SR15	Users can view disk usage	Device Table Storing Disk Usage, SQL Queries	White box
SR16	Users can view network usage	Device Table Storing Memory Network, Queries	White box
SR17	Users can view monitored services	Device Table Storing Monitored Service, SQL Queries	White box
SR28	Data can be archived or removed by users	Archive Database Tables, Insert and Delete Queries	White bpx

Table 2.2: Level 3 Matrix Table

2.2.2 Pass/Fail criteria

Level 3 must successfully complete all test cases.

2.2.3 Test deliverables

All test results will be recorded in the excel spreadsheet.

2.3 Level 2 test plan

2.3.1 Traceability matrix

ID	Requirement	System Component	Approach
SR11	Engine sends an error message to the Data Store when upon receiving invalid data	Error Log Table Storing Error messages	Black Box
SR26	Data Store is able to scale in size	Horizontal and Vertical Scaling	White box

Table 2.3: Level 2 Matrix Table

2.3.2 Approach

Black box testing will be used to make sure data store receive the invalid data message sent from the Engine.

2.3.3 Pass/Fail criteria

Level 2 test must successfully complete all test cases

2.3.4 Test deliverables

The Error Log table will store error messages in an Excel sheet.

2.4 Level 1 test plan

2.4.1 Traceability matrix

ID	Requirement	System Component	Approach
SR23	Data Store runs on Unix-like	CS Server (cs.csis.work)	item inspection
QR5	Document and follow a coding style	SEI CERT	item inspection

Table 2.4: Level 1 Matrix Table

2.4.2 Pass/Fail criteria

Level 1 must successfully complete all test cases.

2.4.3 Test deliverables

Data Store will run on the CS server, making sure document follow a coding style.

3 GENERAL

3.1 Quality assurance

Based on the test plan matrix level, all test must be passed. If all test is not passed then a diagnostic message will be shown, then the team will have to review what is needed for the product to pass the test. Nothing will appear or be visible to the user if the Data Store successfully runs.

3.2 Metrics

- **Level-4**: Engine create and send a network to Data Store, Data store validates request from Dashboard and then sends data to the Dashboard. Data store hold data for at least a year and system can be reported to other devices.
- **Level-3**: User can view CPU usage,memory usage, disk usage, network usage, monitored service, and the data can be archived or removed by the user.
- **Level-2**: Engine sends an error to the Data Store when upon receiving invalid data. Data store is able to scale in size.
- **Level-1**: Data Store running on Unix-like and document follow a coding style.

3.3 Test coverage

The software system will require all the test coverage Level 1,2,4 uses different test coverage, but level 3 uses the same test coverage.