**Test plan for Network Monitoring system**

Table of Contents

[1 Introduction](#_Toc515524393)

[1.1 Purpose](#_Toc515524394)

[1.2 Project Overview](#_Toc515524395)

[2 Scope](#_Toc515524396)

[2.1 In-Scope](#_Toc515524397)

[2.2 Out-of-Scope](#_Toc515524398)

[3 Testing Strategy](#_Toc515524399)

[3.1 Test Objectives](#_Toc515524400)

[3.2 Test Assumptions](#_Toc515524401)

[3.3 Data Approach](#_Toc515524402)

[3.4 Level of Testing](#_Toc515524403)

[3.5 Functional Testing](#_Toc515524405)

[3.6 User Acceptance Testing](#_Toc515524406)

[3.7 Regression Testing](#_Toc515524407)

[4 Execution Strategy](#_Toc515524408)

[4.1 Entry Criteria](#_Toc515524409)

[4.2 Exit criteria](#_Toc515524410)

[4.3 Validation and Defect Management](#_Toc515524411)

[5 Environment Requirements](#_Toc515524412)

[5.1 Test Environments](#_Toc515524413)

## 

1. **Introduction**
   1. **Purpose**

The purpose of a test plan for network monitoring is to ensure that the network monitoring system meets the organization's requirements, standards, and expectations for functionality, performance, security, and reliability. By following a structured test plan, organizations can identify and address issues early in the development lifecycle, minimize risks, and deliver a high-quality network monitoring solution that effectively meets the needs of users and stakeholders.

* 1. **Project Overview**

The network monitoring system is designed to provide comprehensive monitoring, analysis, and management of an organization's network infrastructure. It aims to ensure optimal performance, detect anomalies, and facilitate proactive troubleshooting

1. **Scope**

**2.1 In-scope**

Features to be tested.

* Real-time Network Monitoring
* Device Discovery
* Live Dashboards
* Bandwidth Monitoring
* Performance Analysis
* Network Topology Mapping
* Historical Performance Data
* Security Event Detection
* Log Analysis
* Network Configuration Management
* Configuration Backup
* Compliance Checks
* Alerts and Notifications
* Customizable Alerts
* Reporting and Analytics
* Pre-built Reports
* Trend Analysis
* Integration with Third-party Tools
* SIEM Integration
* Network Device Compatibility

**2.2 Out-of-Scope**

**3.** **Testing Strategy**

**3.1 Test Objectives**

* Performance Monitoring: Measure network performance metrics such as bandwidth usage, latency, and packet loss.
* Security Monitoring: Detect and prevent unauthorized access, intrusions, and abnormal behaviour.
* Availability Monitoring: Ensure network components are up and running, minimizing downtime.
* Traffic Analysis: Analyse network traffic patterns to identify trends, anomalies, and potential issues.
* Alerting and Notification: Set up alerts for critical events and thresholds to enable timely responses and interventions.

**3.2 Test Assumptions:**

Availability of Test Environment

The test environment, including hardware, software, and network configurations, is assumed to be available and properly configured for testing.

Access to Test Data

Test data required for network monitoring testing, such as network traffic captures, log files, and simulated events, are assumed to be available and accessible.

Test Tools and Resources

Test tools and resources necessary for conducting network monitoring testing, such as monitoring software, packet analysers, and performance testing tools, are assumed to be available and functional.

**3.3 Data Approach:**

* Valid test data
* Invalid test data
* Boundary test data
* Wrong data
* Absent data

**3.4 Levels of testing:**

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Description** | **Responsible Parties** |
| **Smoke Testing** | It is used to Determine if a new software build is ready for the next testing phase Smoke tests verify whether the most important features work as expected and that there are no showstopper issues in the build that can potentially lead to blocking the entire testing team. It helps in deciding if the build is flawed or not and hence, prevents the entire team from wasting time or resources. | **QA team** |
| **Functional Testing** | Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.  [**Unit testing**](https://www.geeksforgeeks.org/unit-testing-software-testing/) is the type of functional testing technique where the individual units or modules of the application are tested. It ensures that each module is working correctly.  [**Integration testing**](https://www.geeksforgeeks.org/software-engineering-integration-testing/)**,** combined individual units are tested as a group and expose the faults in the interaction between the integrated units.  [**System testing**](https://www.geeksforgeeks.org/system-testing/) is a type of software testing that is performed on the complete integrated system to evaluate the compliance of the system with the corresponding requirements.  [**Usability testing**](https://www.geeksforgeeks.org/usability-testing/) is done to measure how easy and user-friendly a software application is.  [**User acceptance testing**](https://www.geeksforgeeks.org/acceptance-testing-software-testing/) is done by the client to certify that the system meets the requirements and works as intended. It is the final phase of testing before the product release.  [**Regression testing**](https://www.geeksforgeeks.org/software-engineering-regression-testing/)is done to make sure that the code changes should not affect the existing functionality and the features of the application. It concentrates on whether all parts are working or not. |  |
| **Non-Functional Testing** | Non-Functional testing is essential for confirming the software’s reliability and functionality. The [Software Requirements Specification (SRS)](https://www.browserstack.com/guide/software-requirement-specifications-in-agile) serves as the basis for this software testing method, which enables quality assurance teams to check if the system complies with user requirements. Increasing the product’s [usability](https://www.browserstack.com/guide/what-is-usability-testing), effectiveness, [maintainability](https://www.browserstack.com/guide/maintainability-testing), and portability is the goal of non-functional testing. It aids in lowering the manufacturing risk associated with the product’s non-functional components.  [Performance testing](https://www.browserstack.com/guide/performance-testing) eliminates the causes of the software’s sluggish and constrained performance. The software’s reading speed should be as quick as possible. One must create a well-organized and precise specification about the desired speed for Performance Testing. Otherwise, it won’t be evident if the test is a success or a failure. Example: When 1000 users use an application simultaneously, the load time shouldn’t exceed 5 seconds.  **Tools Used:** JMeter |  |

## 3.5 Functional Testing

Features to be tested.

* Real-time Network Monitoring
* Device Discovery
* Live Dashboards
* Bandwidth Monitoring
* Performance Analysis
* Network Topology Mapping
* Historical Performance Data
* Security Event Detection
* Log Analysis
* Network Configuration Management
* Configuration Backup
* Compliance Checks
* Alerts and Notifications
* Customizable Alerts
* Reporting and Analytics
* Pre-built Reports
* Trend Analysis
* Integration with Third-party Tools
* SIEM Integration
* Network Device Compatibility

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Shweta |  | Tes QAt TesterManager |
| Sushma |  | Tes QAt Aad |

## 

## 3.6 User Acceptance Testing

Features to be tested.

* Real-time Network Monitoring
* Device Discovery
* Live Dashboards
* Bandwidth Monitoring
* Performance Analysis
* Network Topology Mapping
* Historical Performance Data
* Security Event Detection
* Log Analysis
* Network Configuration Management
* Configuration Backup
* Compliance Checks
* Alerts and Notifications
* Customizable Alerts
* Reporting and Analytics
* Pre-built Reports
* Trend Analysis
* Integration with Third-party Tools
* SIEM Integration
* Network Device Compatibility

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Sushma |  | Test ManQAger |
| Shweta |  | Test LeadQA |

## 

## 3.7 Regression Testing

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| s Shweta |  | Test Man Lead QA ager |
| Sushma |  | Test Lead Lead QA |
|  |  |  |

**4 Execution strategy**

**4.1 Entry criteria**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *Test environment(s) is available* |  |  |  |
| *Test data is available* |  |  |  |
| *Code has been merged successfully* |  |  |  |
| *Development has completed unit testing* |  |  |  |
| *Test scripts are completed, reviewed, and approved by the Project Team* |  |  |  |

**4.2 Exit criteria**

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *100% Test Scripts executed* |  |  |  |
| *90% pass rate of Test Scripts* |  |  |  |
| *No open Critical and High severity defects* |  |  |  |
| *All remaining defects are either cancelled or documented as Change Requests for a future release* |  |  |  |
| *All expected and actual results are captured and documented with the test script* |  |  |  |
| *All test metrics collected based on reports from daily and Weekly Status reports* |  |  |  |
| *All defects logged in Defect Tracker/Spreadsheet* |  |  |  |
| *Test environment cleanup completed and a new back up of the environment* |  |  |  |

**4.3 Validation and Defect Management**

Defects will be logged using a designated tracking tool (e.g., Jira).

Defect severity levels: Critical, Major, Minor.

Defect priorities: High, Medium, Low.

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| *1 (Critical)* | * *Functionality is blocked and no testing can proceed* * *Application/program/feature is unusable in the current state* |
| *2 (High)* | * *Functionality is not usable and there is no workaround but testing can proceed* |
| *3 (Medium)* | * *Functionality issues but there is workaround for achieving the desired functionality* |
| *4 (Low)* | * *Unclear error message or cosmetic error which has minimum impact on product use.* |

# **5. Environment Requirements:**

Test Environments:

Hardware, Software, and Network infrastructure required for testing.

Ensure that the test environment is isolated from production systems to avoid interference with live operations.