Software Requirements Specification (SRS)

NetSight

Version 1.0

Instructor: Prof. Jayprakash Lalchandani

Course: IT632 Software Engineering

Prepared by

Group ID : - 13

Sr. No.	Name	Student ID
1	Kenil Dhola	202512029
2	Harshvardhansinh Sarvaiya	202512047
3	Hardik Vachhani	202512072
4	Vivek Savaliya	202512083

Table of Contents

1 Introduction

NetSight is an intelligent network discovery and management solution that can be used to gain a comprehensive level of visibility into network infrastructures. The system can automatically scan and discover all devices that are connected to an organization's network, including routers, switches, firewalls, servers, access points, IoT devices, and other end-user devices.

Using automated network discovery protocols like SNMP, ICMP, and ARP, along with its real-time monitoring and visualization capabilities, the NetSight system allows network administrators to have a precise and accurate representation of network topology.It can be used to improve network management efficiency and security, along with other network management activities.

1.1 Document Purpose

The objective of this Functional Requirements Document (FRD) is to outline the detailed functional and non-functional requirements of the NetSight system. The document will be a formal reference point for the developers, testers, project stakeholders, and system administrators who will be associated with the implementation and deployment of the system.

The FRD will present a systematic description of:

- System Features and Functionalities
- User Roles and Permissions
- Interface Requirements
- Data Management Requirements

The document will ensure that there is a common understanding of the system's objective, scope, and limitations among all the stakeholders.

1.2 Product Scope

NetSight is a centralized web-based network monitoring and discovery solution that automatically identifies and registers devices within a network infrastructure. The system assigns a unique identifier to each detected device and categorizes it based on its role, type, and operational characteristics.

The platform enables administrators to:

- Automatically scan IP ranges and discover devices
- Maintain real-time topology mapping
- Track device status (online/offline)
- Store and manage device metadata (IP address, MAC address, OS, vendor, firmware version)
- Detect new or suspicious devices

- Generate alerts and reports
- Manage users using Role-Based Access Control (RBAC)

Key Benefits of NetSight Include:

- Automated network discovery and device identification
- Real-time topology visualization
- Improved network security through anomaly detection
- Centralized device metadata management
- Proactive monitoring and alerting
- Scalable architecture for enterprise environments

NetSight is designed for organizations seeking enhanced visibility, security, and control over their network infrastructure while minimizing manual network management efforts.

1.3 Intended Audience

- **Development Team:** To understand and implement the functional and non-functional requirements of NetSight.
- **Project Manager:** To plan, manage resources, and monitor project progress.
- **Client / Stakeholders:** To ensure the system aligns with organizational goals and network security needs.
- **Quality Assurance Team:** To design test plans and verify system compliance.
- **System / Network Administrators:** To understand deployment, configuration, and operational features.
- **Professor / Course Instructor:** To evaluate the project documentation and technical completeness.

1.4 Functional Requirements

F1 – User Authentication

- System shall support Login and Signup.
- System shall issue JWT token upon authentication.
- System shall expire tokens after configurable time (default: 1 hour).

F2 – Role-Based Access Control

- System shall support roles: Admin, Network Engineer, Viewer.
- Admin shall manage users and system configuration.
- Network Engineer shall view metrics and troubleshoot.

- Viewer shall have read-only access.

F3 – Network Discovery

- System shall automatically scan network IP range.
- System shall identify device type using SNMP OID.
- Each device shall have unique ID.
- System shall refresh discovery every 15 minutes.

F4 – Device Registration

- System shall store:
    - IP Address
    - MAC Address
    - Device Type
    - Vendor
    - Status

F5 – Topology Visualization

- System shall display devices as nodes.
- System shall display connections as edges.
- Status colors:
    - Green – Healthy
    - Yellow – Warning
    - Red – Critical

F6 – Real-Time Monitoring

System shall collect:

- Latency (ms)
- Packet Loss (%)
- Bandwidth utilization (%)
- Uptime (%)
- CPU Usage (%)

- Memory Usage (%)

Collection Frequency: Configurable (default 60 seconds)

F7 – Alert Management

- System shall allow threshold configuration.
- System shall trigger alerts if threshold exceeded for 3 consecutive cycles.
- Alert channels:
    - Email
    - Web Notification
    - Slack Webhook

F8 – Failure Prediction

- System shall analyze historical data.
- System shall apply anomaly detection model.
- System shall display risk score (0–100%).
- Prediction accuracy ≥ 80%.

F9 – Audit Logs

- System shall log user activities.
- Log retention period: 180 days.
- Logs must be immutable.

F10 – Report Generation

- System shall generate PDF and CSV reports.
- Report duration selectable (daily, weekly, monthly).

1.5 Non-Functional Requirements

- **Performance:** Dashboard loads within 3 seconds; network scan of up to 1000 devices completes within 5 minutes; API response time ≤ 500 ms.
- **Security & Privacy:** Secure authentication using JWT; HTTPS (TLS 1.3); password hashing (bcrypt); Role-Based Access Control (RBAC); SNMPv3 support; OWASP-compliant security practices.

- **Maintainability:** Modular microservice-based architecture; well-documented REST APIs; version-controlled codebase; CI/CD integration support.
- **Usability:** Responsive web interface; intuitive dashboard with real-time visualization; search and filtering capabilities.
- **Reliability:** Minimum 99.5% system uptime; automated retry mechanisms for failed scans; daily database backups.
- **Scalability:** Supports up to 10,000 devices; horizontal scaling supported via load balancer; cloud-ready deployment.

1.6 Architecture Overview

- **Frontend:** React.js (Web Dashboard with real-time topology visualization)
- **Backend:** Node.js / Express.js (RESTful API services)
- **Network Discovery Engine:** SNMP, ICMP, ARP scanning modules
- **Database:** MongoDB (device metadata, users, alerts); Time-series for monitoring logs
- **Authentication & Security:** JWT-based authentication; RBAC implementation
- **Notification System:** SMTP Email service
- **API Layer:** REST API with JSON data exchange; versioned endpoints (/api/v1/)

1.7 Future Enhancements

- Mobile Application
- Advanced AI predictive models
- Integration with SIEM tools
- Automated remediation system

1.8 User Stories for NetSight

ID	User Role	Wants To	So That
US001	Visitor	register using email and password	securely access the NetSight platform
US002	Admin / Network Admin	log in securely	access authorized system features
US003	Admin	create and manage user accounts	control system access
US004	Admin	assign roles to users (Admin / Network Admin)	enforce RBAC policies
US005	Admin	configure IP range for network scanning	discover devices within specific network segments
US006	System	scan network using SNMP, ICMP, and ARP	detect connected devices automatically
US007	System	assign unique ID to each discovered device	maintain structured device tracking
US008	System	categorize devices by type (Router, Switch, Server, IoT)	simplify network management
US009	Network Administrator	view real-time network topology	understand network structure visually
US010	Network Administrator	check device online/offline status	quickly identify outages
US011	Network Administrator	view detailed device metadata	analyze device information effectively
US012	Admin	receive alerts for newly connected devices	verify authorized access
US013	Admin	receive alerts for suspicious or unknown devices	enhance network security
US014	Admin	configure alert thresholds	detect abnormal network behavior
US015	Network Administrator	search and filter devices	quickly locate specific devices
US016	Admin	generate network summary reports	evaluate network performance

ID	User Role	Wants To	So That
US017	Admin	export reports in PDF/CSV format	share reports with management
US018	Network Administrator	view historical device activity	analyze past network events
US019	Admin	view audit logs	track system changes and user activities
US020	System	update topology automatically when device status changes	maintain accurate network representation
US021	Admin	enable/disable scanning schedules	control network discovery operations
US022	Network Administrator	view alert history	monitor recurring network issues
US023	Admin	integrate system with external tools via REST API	enable third-party system integration
US024	Network Administrator	refresh device status manually	verify real-time device availability
US025	Admin	configure notification channels (Email)	ensure timely alert delivery

1.9 NetSight – Project Plan

Overview

- **Project Duration:** Approximately 8–9 weeks
- **Development Methodology:** Agile (Scrum-based with 1-week sprints)
- **Team Structure:** Frontend Developer, Backend Developer(s), Database Engineer, QA

Phase 1: Requirement Analysis & Planning (Week 1)

Deliverables:

- Finalized Software Requirements Specification (SRS)
- Defined User Stories and Use Case Model
- Technology Stack Finalization

- Project Timeline & Sprint Planning

Tasks:

- Stakeholder meetings and requirement gathering
- Define functional and non-functional requirements
- Finalize user roles (Admin, Network Administrator)
- Setup GitHub repository and project tools
- Prepare initial project roadmap

Phase 2: System Design (Week 2)

Deliverables:

- System Architecture Diagram
- Database Schema Design
- API Design Documentation
- UI Wireframes

Tasks:

- Design overall system architecture
- Define REST API endpoints
- Design device metadata schema
- Create DFD Level 0 and Level 1 diagrams
- Prepare UI dashboard mockups

Phase 3: Core Development – Authentication & RBAC (Week 3)

Deliverables:

- User Registration & Login Module
- JWT Authentication Implementation
- Role-Based Access Control (RBAC)

Tasks:

- Implement secure login system
- Configure role permissions
- Implement password hashing and token validation

Phase 4: Network Discovery Module (Week 4)

Deliverables:

- SNMP, ICMP, ARP Scanning Module
- Device Detection Engine
- Unique Device ID Assignment

Tasks:

- Implement network scanning logic
- Integrate SNMP libraries
- Store discovered devices in database
- Validate device metadata extraction

Phase 5: Topology & Monitoring Module (Week 5)

Deliverables:

- Real-time Topology Visualization
- Online/Offline Status Tracking
- Device Metadata Dashboard

Tasks:

- Develop topology graph rendering
- Implement real-time status updates
- Integrate WebSocket (if required)

Phase 6: Alert & Reporting System (Week 6 - 7)

Deliverables:

- Alert Generation System
- Email Notification Integration
- Network Report Generation (PDF/CSV)

Tasks:

- Configure alert rules
- Implement notification module
- Develop report export functionality

Phase 7: Testing & Optimization (Week 7)

Deliverables:

- Test Cases & Test Reports
- Performance Optimization
- Security Hardening

Tasks:

- Unit Testing
- Integration Testing
- Performance testing
- Fix bugs and optimize queries

Phase 8: Deployment & Documentation (Week 8–9)

Deliverables:

- Deployment on Cloud / Linux Server
- Final Documentation Submission
- Project Demonstration

Tasks:

- Dockerization & server deployment
- Configure HTTPS
- Prepare presentation and demo
- Final review and submission

Tools & Technologies

Layer	Technology
Frontend (Web Dashboard)	React.js, HTML5, CSS3, Chart.js ,React Flow, Axios, TypeScript , Socket.IO (Client)
Backend (Core System & APIs)	Node.js, Express.js, Socket.IO, Webhooks
Network Monitoring & Data Collection	ICMP (Ping), TCP Probes, SNMP, Traceroute, net-ping, net-snmp, node-cron
Database & Storage	MongoDB (topology, users, alerts, logs),
AI & Intelligent Analysis	Python, FastAPI, Pandas, NumPy, Scikit-learn (anomaly detection), Prophet / LSTM (optional), LLM API (OpenAI or local)
Alerting & Notification	Backend alert rules, Nodemailer (Email), Slack / Microsoft Teams Webhooks, In-app notifications
Authentication & Security	JWT, Role-Based Access Control (Admin / Network Engineer / Viewer), HTTPS, Rate Limiting, API Key Protection, Audit Logs
Tools / Frameworks / Dev Tools	GitHub, VS Code, Postman, Draw.io (UML Diagrams)

Timeline Overview – NetSight

Phase	Duration (Weeks)	Sprint(s)
Requirement Analysis & Planning	1	Sprint 1
System Design	1	Sprint 2
Authentication & RBAC Development	1	Sprint 3
Network Discovery Module	1	Sprint 4
Topology & Monitoring Module	1	Sprint 5
Alert & Reporting System	2	Sprint 6–7
Testing & Optimization	1	Sprint 7–8
Deployment & Documentation	2	Sprint 8–9
MVP Launch	1	Sprint 9

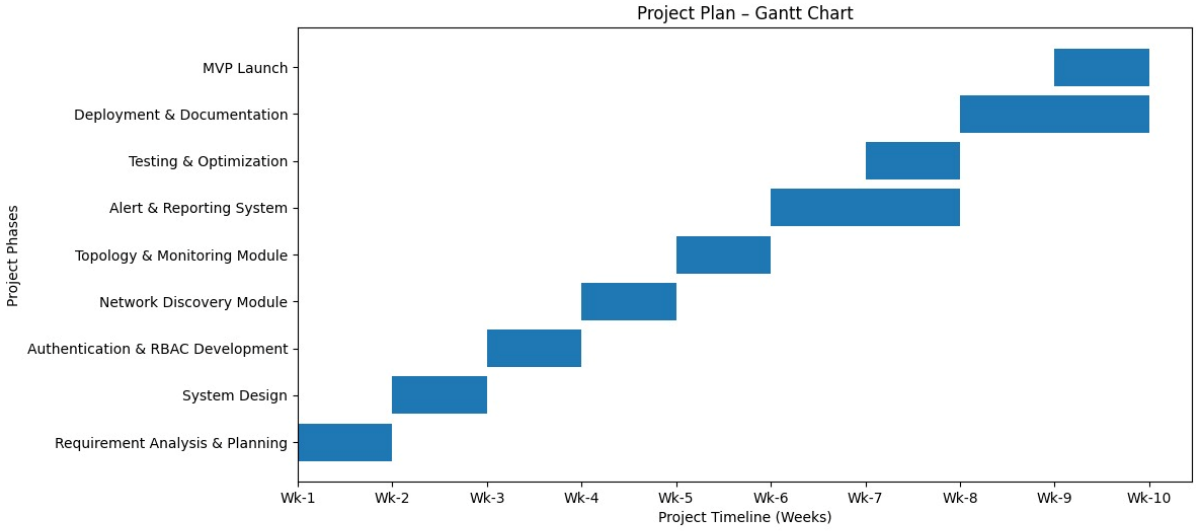

Figure 1: Project Plan - Gantt Chart