# Software Requirements Specification (SRS)

**Project Title:** System performance tracker (on a Windows system) – CPU Usage, Memory usage, HDD, WiFi, etc

**Version:** 1.0  
**Authors:** Karan M, K Adithya Shetty, Pavan Kumar G, Purandar Balasa  
**Date:** 18-08-2025  
**Status:** Draft

## Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Change summary | Approval |
| 1.0 | 18-08-2025 | Instructor | SRS with diagrams embedded |  |

## Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature / Email | Date |
| Course Coordinator |  |  |  |

## Table of Contents

1. Introduction  
2. Overall description  
3. External interfaces  
4. System features (detailed)  
5. Non-functional requirements (detailed)  
6. Quality attributes & Acceptance tests  
7. UML Use-Case Diagram  
8. Requirements Traceability Matrix (RTM)

## 1. Introduction

1.1 Purpose

This document is a Software Requirements Specification (SRS) for System performance tracker (on a Windows system) – CPU Usage, Memory usage, HDD, WiFi, etc system.

1.2 Scope  
The System Performance Tracker is a desktop-based application designed to monitor and analyze the performance of an operating system in real-time. The system will provide users with detailed information about key hardware and network resources to ensure efficient troubleshooting, system optimization, and effective resource utilization.

1.3 Audience  
Developers, QA Engineers, System Integrators, Maintenance Technicians, and Assessment Evaluators.

1.4 Definitions

**CPU:** Central Processing Unit

**HDD:** Hard Disk Drive

**PID:** Process Identifier - A unique number that identifies an active process.

**PPID:** Parent Process Identifier - The PID of the process that created the specified process.

**TTY:** Teletypewriter - A terminal device for user interaction.

**CLI:** Command Line Interface.

**TUI:** Terminal User Interface.

## 2. Overall description

2.1 Product perspective  
The System performance tracker

2.2 Major product functions (detailed)  
- CPU Utility  
- Virtual memory usage  
- Networking usages  
- State of processes  
- HDD Usage

- Processes Running currently

2.3 User roles and characteristics (expanded)

Role based access to Telemetry

* Standard users - Are exposed to aggregate system metrics , ensuring a minimal yet sufficient operational view.
* Privileged users - Granted access to granular instrumentation data. Could be accessed through root user privilege. Also allowed to manage processes.

* **Operating Systems**
* Any Linux based distro
* Unix based distros such as macOS

**- Runtime Environments**

-GCC/minGW - to compile and run c/c++

- Clang - alternate compiler with fast builds and modern tooling support

- Makefile - for build automation and dependency management

- CMake - Cross platform build system generator

2.5 Constraints  
Requires advanced privileges to view all performance metrics, constrained to a terminal UI over a desktop GUI, performance of project could vary based on operating systems.

## 3. External interface requirements

3.1 User interfaces  
Terminal UI: command line tool with multiple arguments passable.

System metrics are viewed in real-time and is refreshed dynamically every second.

3.2 Hardware interfaces  
System does not interact with hardware on a low level, hardware access is indirect, via the operating system resource log files.

3.3 Software interfaces  
- Operating systems interface: The tracker interacts with the systems API and libraries to monitor resources.

* Linux/Unix - /proc filesystem, systemcalls
* C - ncurses library (for the terminal user interface), OS-specific libraries to view resource monitoring

3.4 Communications  
- The system communicates with user via the terminal interface.

<< Make sure overall there are at least 15 FRs for overall project, 5 NFRs, 2 security objectives and 5 Security requirements>>

## 4. System features (detailed)

Each requirement below includes acceptance criteria and a reference test case. IDs follow SPT-F-###.

### 4.1 Authentication

Description: We can separate between the normal users and privileged users based on the privileges that the user account has. Privileged mode could only be accessed by the root user.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| SPT-F-001 | The system shall validate admin users via sudo. | Functional | High | Security | AC-SPT-F-001: Validate  Root user using sudo leads to an authenticated session. Test: TC-Auth-01 |  |
| SPT-F-002 | The system shall only allow access to restricted metrics (e.g., CPU freq scaling, network I/O stats) when run with root privileges. | Functional | High | PCI-DSS | AC-SPT-F-002: Attempt to run as normal user denies access to restricted metrics. Test: TC-Auth-02 | Enforced by OS |
| SPT-F-003 | If run without sufficient privileges, the system shall deny access and print a message: “Please run with sudo/admin rights.” | Functional | Medium | Usability | AC-SPT-F-003: Normal user receives clear error message. Test: TC-Auth-03 |  |
| SPT-F-004   |  | | --- | |  |  |  | | --- | |  | | The system shall log all privileged access attempts (successful or denied) to a file for auditing. | Functional | Medium | Security | Verify log entry is created on successful and failed privileged access attempts. | Depends on file system permissions |

### 4.2 Core Monitoring

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement (shall...) | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| SPT-F-005   |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  | | The system shall display real-time CPU usage per core. | Functional | High | Performance Monitoring   |  | | --- | |  |  |  | | --- | |  | | CPU usage per core updates in ≤1s. | Uses /proc/stat |
| |  | | --- | | SPT-F-006 |  |  | | --- | |  | | The system shall display real-time memory utilization (used, free, cached, swap).   |  | | --- | |  |  |  | | --- | |  | | Functional | High | Performance Monitoring | Memory stats refresh in ≤1s.  Test**:** TC-Mon-02 | Uses /proc/meminfo |
| |  | | --- | | SPT-F-007 |  |  | | --- | |  | | The system shall display disk usage and I/O statistics per mounted filesystem. | Functional | High | |  | | --- | | Performance Monitoring |  |  | | --- | |  | | AC-SPT-F-007: Disk usage & I/O stats visible per mount.  Test: TC-Mon-03 | Uses /proc/diskstats, df |
| |  | | --- | | SPT-F-008 |  |  | | --- | |  | | The system shall display network utilization (inbound/outbound packets, bandwidth). | Functional | High | Performance Monitoring | AC-SPT-F-008: Network traffic shown with correct values.  Test: TC-Mon-04 | Uses /proc/net/dev |

### 4.3 Process Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| |  | | --- | | SPT-F-009 |  |  | | --- | |  | | The system shall list running processes with PID, CPU %, and memory %. | Functional | High | Usability / Admin | AC-SPT-F-009: Process list includes PID, CPU %, and memory %.  Test: TC-Mon-05 | Uses /proc/[pid] |
| |  | | --- | | SPT-F-010 |  |  | | --- | |  | | The system shall allow sorting of processes by CPU %, memory %, or PID. | Functional | Medium | |  | | --- | | Usability / Admin |  |  | | --- | |  | | AC-SPT-F-010: Verify sorting works in all supported modes.  Test: TC-Mon-06   |  | | --- | |  | | Requires process table implementation |

### 4.4 Alerts & Thresholds

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| |  | | --- | | SPT-F-011 |  |  | | --- | |  | | The system shall trigger a warning when CPU utilization exceeds a configurable threshold (default: 85%). | Functional | High | Reliability / Admin | AC-SPT-F-011: Warning is triggered when CPU > threshold.  Test: TC-Alert-01 | Threshold configurable |
| SPT-F-012 | The system shall trigger a warning when available memory falls below a configurable threshold. | Functional | High | Reliability / Admin | AC-SPT-F-012: Warning triggered when memory < threshold.  Test: TC-Alert-02 | Threshold configurable |
| SPT-F-013 | The system shall allow the user to configure custom alert thresholds via a config file. | Functional | Medium | Usability | AC-SPT-F-013: Verify threshold config file updates affect monitoring behavior.  Test: TC-Alert-03 | Depends on config file parser |

### 4.5 Usability & Output

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| SPT-F-014 | The system shall provide both interactive (terminal UI) and non-interactive (CLI report) modes. | Functional | Medium | Usability | AC-SPT-F-014: Both interactive and CLI reporting available.  Test: TC-UI-01 | Interactive mode requires TTY |
| SPT-F-015 | The system shall support exporting monitoring data to a log file in CSV or JSON format. | Functional | Medium | Usability / Admin | AC-SPT-F-015: Verify CSV/JSON log file generated and matches observed values.  Test: TC-UI-02 | Depends on file system write access |

## 5. Non-functional requirements (detailed)

NFRs below are measurable and tied to test plans. IDs SPT-NF-###

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement | Category | Priority | Acceptance criteria / Measurement |
| SPT-NF-001 | The system shall update displayed real-time metrics (CPU, Memory, Disk, Network, Wi-Fi) with a maximum end-to-end latency of **1 second** under normal desktop load. | Performance | High | 95% of UI metric refreshes must render within 1.0 second over a 10-minute run.  Test Case: TC-NF-01 |
| SPT-NF-002 | The system shall not crash when monitored processes start/stop rapidly, and shall recover to a stable state within **5 seconds** after a transient fault (e.g., lost access to a metric device). | Reliability | High | After injected metric-read faults or rapid process churn, the app remains running and returns to normal updates within **5 seconds**.  **Test Case:** **TC-NF-02** |
| SPT-NF-003 | Access to privileged metrics (kernel counters, low-level NIC stats) shall require explicit administrative permission; the tool must display a clear prompt and refuse privileged reads unless run elevated. | Security | High | Privileged metrics are inaccessible to non-elevated users and produce a clear denial message when attempted.  **Test Case:** **TC-NF-03** |
| SPT-NF-004 | The system shall produce structured logs (JSON lines or similar) with timestamps, log level, component name, and unique event IDs. Logs shall rotate automatically and be retained per configurable policy (default: 90 days). | Operability / Audit | Medium | Logs are structured (JSON), include timestamp+level+eventID, rotate automatically, and retention policy (default 90 days) is enforced.  **Test Case:** **TC-NF-04** |
| SPT-NF-005 | The UI (desktop or terminal mode) shall be readable at common display sizes: support keyboard navigation, high-contrast theme, and scale correctly for 100% and 150% Windows scaling settings. | Usability / Accessibility | Medium | Keyboard navigation, high-contrast theme, and 100% & 150% DPI scaling produce no layout break and remain usable.  **Test Case:** **TC-NF-05** |

## 5.1. Security

## 5.1.1 Security Objectives

* Protect confidentiality of metric exports
* Ensure tamper-evident audit logging

## 5.1.2 Security Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement (shall...) | Type | Priority | Acceptance criteria / Test case ref |
| SPT-SR-001 | The system shall enforce role-based access control, restricting process termination to privileged users only. | Security | High | A standard user's attempt to kill a process owned by another user (including root) must fail. Test: TC-SR-01 |
| SPT-SR-002 | The system shall sanitize all user-provided input from the configuration file to prevent buffer overflows or injection attacks. | Security | High | Providing malformed or overly long strings in the config file does not cause a crash or exploitable behavior. Test: TC-SEC-02 |
| SPT-SR-003 | The system shall create an append-only audit log for all privileged actions (e.g., process termination). | Security | Medium | Privileged actions are logged with timestamp, user, action, and target PID. Log file permissions prevent modification by standard users. Test: TC-SEC-03 |
| SPT-SR-004 | The system's configuration file shall have default file permissions restricting write access to the owner (e.g., 644 or rw-r--r--). | Security | Medium | On first run, the generated config file has permissions that prevent modification by other non-root users. Test: TC-SEC-04 |
| SPT-SR-005 | The system shall not expose sensitive environment variables from other processes to a standard user. | Security | High | When viewing process details as a standard user, the process environment block is not accessible. Test: TC-SEC-05 |

## 6. Quality attributes & Acceptance tests

**Quality Attributes**

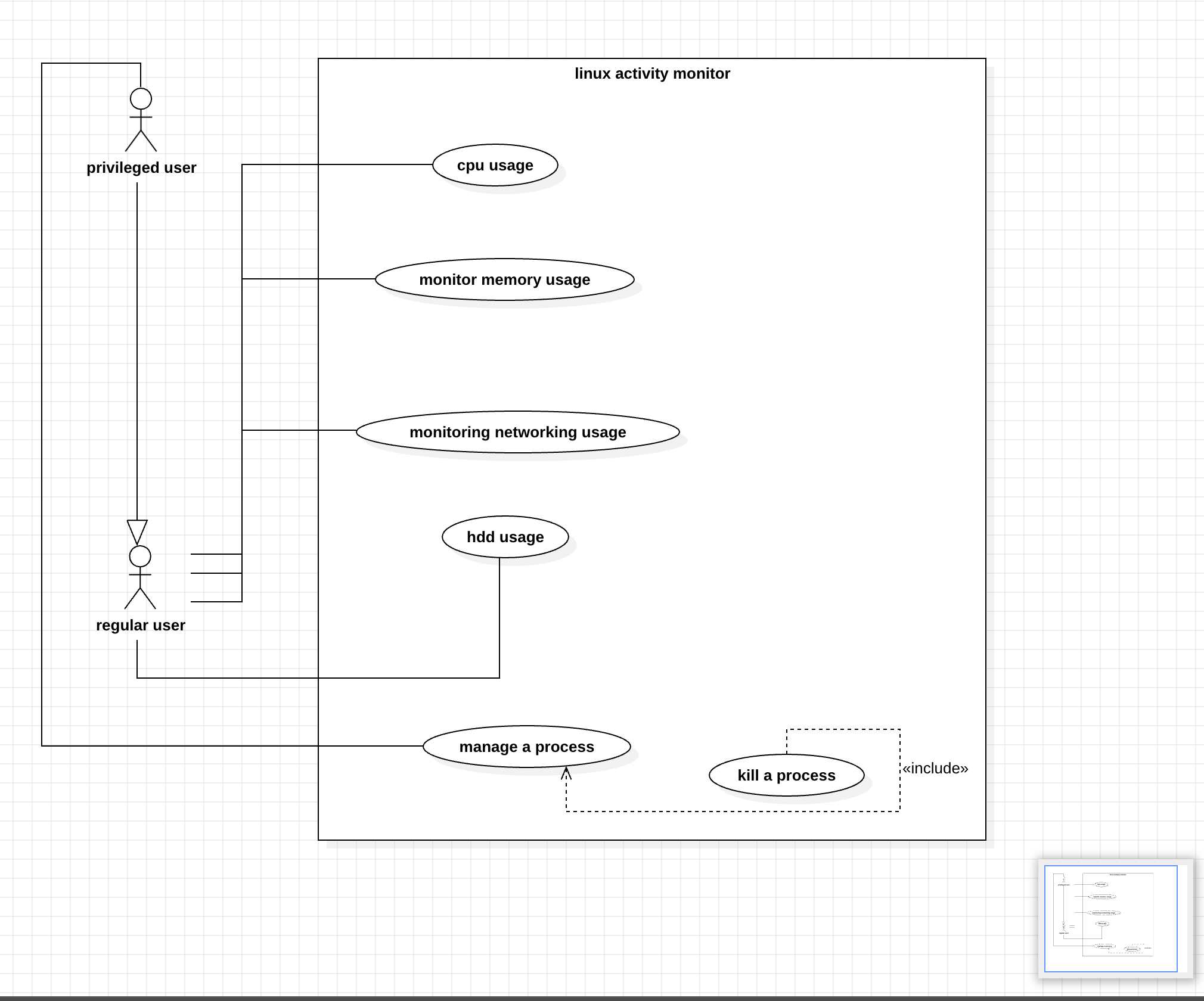
* Performance: Low overhead, near real-time updates.
* Usability: Clear, readable terminal output.
* Reliability: Handles process/CPU changes without crashing.
* Portability: Runs on major Linux distros.
* Security: Reveals advanced stats only with sudo.

**Acceptance Tests**

* CPU usage of tool ≤ 2%.
* Updates visible within 1s under load.
* Output remains aligned across terminals.
* No crashes when processes exit mid-refresh.
* Runs unmodified on Ubuntu, Fedora, Arch.
* Privileged info only visible as root.

## 7. System models and diagrams

## 7.1 UML Use-Case diagram



## 8. Requirements Traceability Matrix (RTM)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement short | Section ref / Design Spec | Module | Test case(s) | Status (N/P/A) | Comments |
| SPT-F-001 | Validate admin users via sudo | 4.1 Authentication | Auth Module | TC-F-01 | N | Root privilege check |
| SPT-F-002 | Restrict access to privileged metrics | 4.1 Authentication | Auth Module | TC-F-02 | N | OS enforced |
| SPT-F-003 | Deny insufficient privileges, show error | 4.1 Authentication | Auth Module | TC-F-03 | N | Usability message |
| SPT-F-004 | Log privileged access attempts | 4.1 Authentication | Logging Module | TC-F-04 | N | Requires file permissions |
| SPT-F-005 | Display real-time CPU usage per core | 4.2 Core Monitoring | CPU Monitor | TC-F-05 | N | Uses /proc/stat |
| SPT-F-006 | Display memory utilization | 4.2 Core Monitoring | Memory Monitor | TC-F-06 | N | Uses /proc/meminfo |
| SPT-F-007 | Display disk usage & I/O stats | 4.2 Core Monitoring | Disk Monitor | TC-F-07 | N | Uses /proc/diskstats |
| SPT-F-008 | Display network utilization | 4.2 Core Monitoring | Network Monitor | TC-F-08 | N | Uses /proc/net/dev |
| SPT-F-009 | List running processes (PID, CPU %, Mem %) | 4.3 Process Management | Process Manager | TC-F-09 | N | Uses /proc/[pid] |
| SPT-F-010 | Allow sorting processes | 4.3 Process Management | Process Manager | TC-F-10 | N | Sorting logic required |
| SPT-F-011 | Trigger CPU threshold warning | 4.4 Alerts & Thresholds | Alerts Module | TC-F-11 | N | Configurable threshold |
| SPT-F-012 | Trigger memory threshold warning | 4.4 Alerts & Thresholds | Alerts Module | TC-F-12 | N | Configurable threshold |
| SPT-F-013 | Configure custom thresholds via config | 4.4 Alerts & Thresholds | Config Parser | TC-F-13 | N | Depends on config file |
| SPT-F-014 | Provide interactive & non-interactive modes | 4.5 Usability & Output | UI Module | TC-F-14 | N | TTY required |
| SPT-F-015 | Export monitoring data (CSV/JSON) | 4.5 Usability & Output | Export/Logging Module | TC-F-15 | N | Depends on file system |
| SPT-NF-001 | Metrics update latency ≤1s | 5. NFRs | Performance Engine | TC-NF-01 | N | 95% updates within 1s |
| SPT-NF-002 | Recover within 5s after transient fault | 5. NFRs | Reliability Engine | TC-NF-02 | N | Fault injection test |
| SPT-NF-003 | Privileged metrics need admin rights | 5. NFRs | Security Layer | TC-NF-03 | N | Must deny unauthorized access |
| SPT-NF-004 | Structured JSON logs with rotation | 5. NFRs | Logging Module | TC-NF-04 | N | Default retention 90 days |
| SPT-NF-005 | UI accessibility & scaling | 5. NFRs | UI Module | TC-NF-05 | N | Test at 100% & 150% scaling |
| SPT-SR-001 | Role-based access for process termination | 5.1 Security | Process Manager + Auth | TC-SR-01 | N | Standard users restricted |
| SPT-SR-002 | Sanitize config input | 5.1 Security | Config Parser | TC-SR-02 | N | Prevent buffer overflow |
| SPT-SR-003 | Append-only audit logs | 5.1 Security | Logging Module | TC-SR-03 | N | Prevent log tampering |
| SPT-SR-004 | Secure default file permissions for config | 5.1 Security | Config/Setup | TC-SR-04 | N | Owner-only write access |
| SPT-SR-005 | No sensitive env vars for normal users | 5.1 Security | Process Manager | TC-SR-05 | N | Protect process details |