Software Requirements Specification

for

InfraVision

Version 1.0

Prepared by Deepam Ahuja, Kabir Panda, Siddarth Warrier

Manipal Academy of Higher Education

14th March 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Deepam Ahuja, Kabir Panda, Siddarth Warrier	March 14, 2025	Initial Draft	1.0

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the requirements for InfraVision V-1.0, a comprehensive systems monitoring toolkit. This document covers the entire scope of the application, including its monitoring capabilities, task automation features, incident management system, configuration management, user access tracking and load balancing functionality.

1.2 Document Conventions

This document follows IEEE 820-1998 standards for Software Requirements Specifications. They are categorized as follows:

a. *High Priority: Essential for the core functionality of software.*

b. *Medium Priority: Important features that enhance system performance.*

c. *Low Priority: Desirable features that are not critical to system operations.*

1.3 Intended Audience and Reading Suggestions

The document is intended for:

- *Developers: To understand the technical requirements and implementation details of the software.*
- *Project Managers: To plan project scope, timeline and resource allocation.*
- *Database Admins: To understand the data structure and relations.*
- *IT Operators: To understand system functionality and benefit.*
- *Testers: Create test cases and validation criteria.*

Reading Suggestions:

a. *Developers and DB Admins: Sections 3,4 and Appendix C*

b. *Project Managers: Sections 1,2 and 5*

c. *IT Operators: Sections 2.2, 4 and 5.1*

1.4 Product Scope

InfraVision is a solution in infrastructure management designed to address the challenges of modern IT infrastructure. It bridges critical technological gaps by providing an intelligent, proactive monitoring platform the offers:

a. *Multi-site infrastructure management capabilities.*

b. *Proactive system health monitoring system.*

c. *Automated routine administrative tasks.*

d. *Actionable insights for Information Technology teams.*

e. *Continuous compliance and security tracking.*

The system aims to streamline operational processes, reduce manual intervention, enhance software reliability, improve system operational efficiency and provide comprehensive monitoring capabilities across diverse IT Teams, Applications and Environments.

1.5 References

a. *IEEE Std. 820 – 1998, IEEE Recommended Practice for Software Requirements Specifications.*

b. *InfraVision Project Synopsis Document, Feb. 2025*

c. *InfraVision DB Normalization, Mar. 2025*

d. *OWASP Security Guidelines, 2025 Edition*

e. *Solarwinds Server and Application Monitor: Deployment and Administration, Nov. 2013*

2. Overall Description

2.1 Product Perspective

InfraVision is a self-contained software designed to address the fragmented nature of current infrastructure monitoring solutions. It can integrate with existing monitoring tools, server ticketing systems and configuration management databases, providing a unified platform that consolidates these functionalities into a single, cohesive unit in a packaged solution.

The system architecture follows a microservice approach with the following major components:

a. *Monitoring Engine*

b. *Task Automation Services*

c. *Incident Management System*

d. *Configuration Management Repo*

e. *User Access Control System*

These components interact through custom APIs and Endpoints and share a common normalized database schema.

2.2 Product Functions

InfraVision provides the following major functions:

a. Real Time Monitoring Dashboard

i. Monitoring CPU, Memory, Disk Usage with Health Indicators

ii. Display metrics via interactive and pleasant UI with updates.

b. Task Automation Toolkit

i. Automation of routine tasks including service restarts, patch deployment and backups.

ii. Log all automation actions executed for traceability and compliance.

c. Configuration Management

i. Stores and track server baseline configurations.

ii. Detect configuration drifts and provide one-click restoration.

d. Incident Management System

i. Provide a Ticketing System for logging, assigning and escalating incidents.

ii. Monitor Service Level Agreements with escalation alerts.

e. User Access and Privilege Tracker

i. Track user activities and flag suspicious behaviour.

ii. Generate compliance audit reports.

f. Role Based Access Control (RBAC)

i. Restrict sensitive operations to authorized users based on roles

ii. Enable role-specific dashboards and action logging

g. Load balancing

i. Monitoring and balance workloads across server groups.

ii. Track resource optimization metrics.

2.3 User Classes and Characteristics

The InfraVision system serves the following user classes:

a. System Admins:

- *Frequency of use: Daily, Continuous*
- *Technical Expertise: High*
- *Privilege Level: High*
- *Functions used: All system features.*
- *Priority: High*

b. IT Operations Staff:

- *Frequency of use: Daily*
- *Technical Expertise: Medium to High*
- *Privilege Level: Medium*
- *Functions used: Monitoring Dashboard, Incident management, Task automation*
- *Priority: High*

c. IT Managers:

- *Frequency of use: Weekly*
- *Technical expertise: Medium*
- *Privilege Level: Medium*
- *Functions used: Reporting, Compliance Verification, high-level monitoring*
- *Priority: Medium*

d. Security Analysts:

- *Frequency of use: As needed*
- *Technical expertise: High*
- *Privilege Level: Medium*
- *Functions used: User access tracking, suspicious activity monitoring*

- *Priority: Medium*

e. Auditors:

- *Frequency of use: Quarterly*
- *Technical expertise: Low to Medium*
- *Privilege Level: Low (READ-ONLY)*
- *Functions used: Audit Logs, Compliance reports*
- *Priority: Low*

2.4 Operating Environment

a. Server Environments which populate database:

- *OS: Linux (Ubuntu 20.04 +, CentOS 8 +), Windows Server 2019 +*
- *Hardware: Any -86-64 compatible server with min 8GB RAM, 4 CPU cores*
- *Storage: Minimum 100 GB SSD Storage*
- *Database: Oracle SQL+ 19c and above*

b. Network Environment:

- *Minimum Bandwidth: 10 Mbps*
- *Secure HTTPS Connections required.*
- *Supports for WebSocket connection for real time feed of server parameters*

c. Client Environments:

- *Modern web browsers*
- *Minium Screen Resolution 1280 * 800*
- *Windows (application packaged as binary to install before running)*

2.5 Design and Implementation Constraints

The following constraints affect the design and implementation of InfraVision

a. Technical Constraints:

- *Backend Development using .NET*
- *Front-End Development – Windows Form Application in VC#*
- *Database implementation using Oracle SQL +*
- *Current implementation will have Python to populate database for simulation and potential use of*

PostMan for testing development of APIs.
- *MicroServices architecture requirement.*
- *JavaScript framework React for future web application development.*

b. Security Constraints:
- *Data transmission must be encrypted.*
- *Password storage must use hashing algorithms*
- *Compliance with HIPAA and other relevant regulations.*

c. Performance Constraints:
- *Dashboard updated withing sub-seconds response times.*
- *Systems must support monitoring of at least 100 servers.*
- *Minimal impact on monitored servers (<0.5 % CPU overhead)*

d. Integration Constraints:
- *Provide standardized APIS for external system integration.*
- *Must support common monitoring protocols (SNMP/WMI)*

2.6 User Documentation

The following user documentation to be provided with InfraVision

a. ***Installation and Setup Guide****:*

- *System requirements*
- *Installation procedures*
- *Initial configuration steps*

b. ***Administrator Manual****:*

- *System architecture overview*
- *Configuration management procedures*
- *User management and role configuration*
- *Backup and recovery procedures*

c. ***User Guide****:*

- *Dashboard navigation and usage*
- *Incident management procedures*
- *Task automation workflows*
- *Reporting features*

d. ***API Documentation****:*

- *RESTful API endpoints*
- *Request/response formats*
- *Authentication requirements*
- *Code examples*

e. ***Online Help System****:*

- *Context-sensitive help within the application*
- *Searchable knowledge base*
- *Video tutorials for common tasks*

2.7 Assumptions and Dependencies

The following assumptions and dependencies affect the InfraVision project:

a. Assumptions:

- *Target servers have accessibility via standard network protocols.*
- *Sufficient network bandwidth between monitoring servers and targets.*
- *Users have familiarity with IT infrastructure concepts.*
- *Target environment has support for required Database System.*

b. Dependencies:

- *Oracle SQL+ 19c and above for database operations.*
- *Redux Libraries with VC# for frontend development.*
- *.NET for backend development.*
- *Python for populating databases with mock server data.*
- *JavaScript, HTML, CSS and model web browsers for future development of Web Application.*

3. External Interface Requirements

3.1 User Interfaces

InfraVision provides a Windows app user interface with the following characteristics:

a. Dashboard Interface:

- *Interactive, real-time dashboards with customizable widgets*
- *Color-coded system health indicators (green, yellow, red)*
- *Drill-down capabilities for detailed metrics*
- *Responsive design for various screen sizes.*

b. ***Navigation****:*

- *Left sidebar for main navigation categories*
- *Top bar for user settings, notifications, and search*
- *Breadcrumb navigation for current location*
- *Quick action buttons for common tasks.*

c. Data Visualization:

- *Line charts for performance trends.*
- *Heat maps for resource utilization.*
- *Status indicators for health service.*
- *Tabular data with sorting and filtering.*

d. Forms and inputs:

- *Validation for all user inputs.*
- *Autocomplete for common fields.*
- *Consistent error messaging.*
- *Confirmation dialog for critical actions.*

e. Accessibility:

- *Keyboard navigation support*
- *Screen reader compatibility*
- *Configurable color themes.*
- *Text Size adjustment options.*

3.2 Hardware Interfaces

InfraVision interacts with the following hardware interfaces:

a. Server Hardware Monitoring*:*

- *CPU usage tracking via native operating system APIs*
- *Memory utilization monitoring*
- *Storage capacity and performance metrics*

- *Network interface statistics*

b. Environmental Sensors:

- *Temperature Sensors*
- *Humidity Sensors*
- *Power Consumption metres*

c. Network Devices*:*

- *Router and switch monitoring via SNMP*
- *Firewall status monitoring*
- *Load balancer performance metrics*

3.3 Software Interfaces

InfraVision integrates with the following software interfaces:

a. ***Operating Systems****:*

- *Linux monitoring via SSH and system utilities*
- *Windows monitoring via WMI and PowerShell*
- *Virtualization platforms (VMware) via its respective API*

b. ***Databases****:*

- *Oracle SQL+ (primary system database)*
- *Support for monitoring other database systems (PostgreSQL)*

c. ***External Systems****:*

- *Email servers for notifications (SMTP)*
- *SMS gateways for urgent alerts*
- *Authentication services (LDAP, Active Directory)*
- *External ticketing systems (optional integration)*

d. ***Cloud Service:***

- *AWS CloudWatch integration*

3.4 Communications Interfaces

InfraVision utilizes the following communication interfaces:

a. ***Network Protocols****:*

- *HTTPS for secure web interface access*
- *WebSockets for real-time dashboard updates*
- *SSH for secure remote management*
- *SNMP for network device monitoring*

b. ***APIs****:*

- *RESTful API for external system integration*
- *JSON data format for API requests.*

- *OAuth 2.0 for API authentication*

c. ***Notification Channels****:*

- *Email notifications (SMTP)*
- *SMS alerts (via gateway APIs)*
- *Push notifications (web-based) (optional)*
- *Integration with messaging platforms (Slack)*

4. System Features

4.1 Real-Time Monitoring Dashboard

4.1.1 Description and Priority

The Real-Time Monitoring Dashboard provides a comprehensive view of infrastructure health and performance metrics. It displays key performance indicators (KPIs) with visual indicators of system health.

Priority: High

4.1.2 *Stimulus/Response Sequences*

- *User logs into the system and navigates to the dashboard*
- *System displays the default dashboard with configured widgets*
- *User selects a specific server or server group*
- *System updates the dashboard to show metrics for the selected target*
- *System automatically refreshes metrics at configured intervals*
- *System highlights metrics that exceed thresholds with color-coded indicators*
- *User clicks on a metric for detailed information*
- *System displays detailed charts and historical data for the selected metric*

4.1.3 *Functional Requirements*

- *RTM-1: The system shall collect and display real-time performance metrics including CPU usage, memory utilization, disk space, and network traffic.*
- *RTM2: The system shall provide visual indicators (green, yellow, red) for health status based on configurable thresholds.*
- *RTM-3: The system shall allow users to customize dashboards by adding, removing, and arranging widgets.*
- *RTM-4: The system shall support drill-down capabilities to view detailed metrics for specific components.*
- *RTM-5: The system shall maintain historical performance data for trend analysis.*
- *RTM-6: The system shall automatically refresh dashboard data at configurable intervals (default: 30 seconds).*
- *RTM-7: The system shall support filtering of displayed servers by location, group, or custom criteria.*

4.2 Task Automation Toolkit

4.2.1 Description and Priority

The Task Automation Toolkit allows users to create, schedule, and execute routine administrative tasks across the infrastructure without manual intervention. It provides a library of predefined scripts and the ability to create custom automation workflows.

Priority: High

4.2.2 Stimulus/Response Sequences

- *User navigates to the Task Automation section*
- *System displays available task types and previously configured automation scripts*
- *User selects a task type or creates a new custom script*
- *System presents configuration options for the selected task*
- *User configures task parameters, target servers, and schedule*
- *System validates the configuration and saves the task*
- *System executes the task according to the defined schedule or on-demand trigger*
- *System logs the execution details and results*
- *System notifies the user of task completion or failure*

4.2.3 Functional Requirements

- *TAT-1: The system shall provide a library of predefined automation scripts for common administrative tasks (service restarts, patch deployments, backups, log rotations).*
- *TAT-2: The system shall allow users to create custom automation scripts using a scripting interface.*
- *TAT-3: The system shall support scheduling of tasks with options for one-time execution, recurring schedules, and event-triggered execution.*
- *TAT-4: The system shall maintain a comprehensive execution log for all automated tasks including start time, end time, affected systems, and execution status.*
- *TAT-5: The system shall provide failure handling mechanisms with configurable retry options and notifications.*
- *TAT-6: The system shall support approval workflows for critical tasks before execution.*
- *TAT-7: The system shall allow task targeting by server groups, locations, or custom criteria.*

4.3 Incident Management System

4.3.1 ***Description and Priority***

The Incident Management System provides a structured approach to tracking, managing, and resolving infrastructure issues. It includes a ticketing system, SLA monitoring, and escalation workflows.

Priority: High

4.3.2 *Stimulus/Response Sequences*

1. *System detects an issue or user manually creates an incident*
2. *System generates a new incident record with a unique identifier*
3. *System assigns the incident based on configured rules or manual assignment*
4. *System notifies assigned personnel via configured channels*
5. *User updates incident status and adds comments as resolution progresses*
6. *System tracks time-to-resolution against defined SLAs*
7. *System escalates incidents that approach or exceed SLA thresholds*
8. *User resolves the incident and documents resolution steps*
9. *System maintains the incident record for historical reporting*

4.3.3 *Functional Requirements*

IMS-1: The system shall automatically create incidents based on monitoring alerts that exceed defined thresholds.
IMS-2: The system shall support manual incident creation with customizable fields for issue description, severity, category, and affected systems.
IMS-3: The system shall provide configurable incident assignment rules based on incident type, severity, and affected systems.
IMS-4: The system shall track incident status through a customizable workflow (e.g., New, Assigned, In Progress, Resolved, Closed).
IMS-5: The system shall support SLA monitoring with configurable thresholds for response time and resolution time based on incident severity.
IMS-6: The system shall provide automated escalation when incidents approach or exceed SLA thresholds.
IMS-7: The system shall maintain a complete audit trail of incident updates, status changes, and comments.
IMS-8: The system shall support linking related incidents and creating parent-child relationships for complex issues.

4.4 Configuration Management

4.4.1 Description and Priority

The Configuration Management feature stores and maintains baseline configurations for all managed servers, detects configuration drift, and enables one-click restoration of approved configurations.

Priority: Medium

4.4.2 Stimulus/Response Sequences

1. *System captures initial baseline configuration for a server*
2. *System stores the configuration in the configuration repository*
3. *User designates the configuration as an approved baseline*
4. *System periodically checks server configurations against the baseline*

5. *System detects and reports configuration drift when found*
6. *User reviews configuration differences*
7. *User approves changes or initiates configuration restoration*
8. *System applies the baseline configuration if restoration is selected*
9. *System logs all configuration changes with before/after details*

4.4.3 Functional Requirements

CFG-1: The system shall capture and store baseline configurations for all managed servers including operating system settings, installed software, and service configurations.

CFG-2: The system shall maintain version history for all configuration changes with timestamps and user attribution.

CFG-3: The system shall perform scheduled configuration compliance checks to detect unauthorized changes.

CFG-4: The system shall provide detailed comparison views showing differences between current and baseline configurations.

CFG-5: The system shall support one-click restoration of approved baseline configurations.

CFG-6: The system shall allow creation of configuration templates for standardized server deployments.

CFG-7: The system shall maintain a comprehensive audit trail of all configuration changes for compliance purposes.

4.5 User Access and Privilege Tracker

4.5.1 Description and Priority

The User Access and Privilege Tracker monitors user activities across the infrastructure, tracks access patterns, and identifies potentially suspicious behavior.

Priority: Medium

4.5.2 Stimulus/Response Sequences

1. *System logs user authentication events across the infrastructure*
2. *System records user actions including command execution and file access*
3. *System analyzes access patterns against historical behavior*
4. *System flags unusual or suspicious activities*
5. *System notifies security personnel of potential security concerns*
6. *User investigates flagged activities*
7. *System generates access reports for compliance and audit purposes*

4.5.3 Functional Requirements

UAT-1: The system shall track user authentication events including successful logins, failed login attempts, and privilege escalation.

UAT-2: The system shall record user activities including commands executed, files accessed, and configuration changes.

UAT-3: The system shall establish baseline access patterns for individual users and roles.

UAT-4: The system shall detect and flag anomalous access patterns that deviate from established baselines.

UAT-5: The system shall provide configurable alert thresholds for suspicious activities such as multiple failed login attempts, off-hours access, or unusual command execution.

UAT-6: The system shall generate comprehensive access reports for security and compliance purposes.

UAT-7: The system shall maintain user access logs for a configurable retention period (default: 1 year).

4.6 Role-Based Access Control

4.6.1 Description and Priority

The Role-Based Access Control (RBAC) system manages user permissions within InfraVision, restricting access to features and operations based on assigned roles.

Priority: High

4.6.2 Stimulus/Response Sequences

1. *Administrator creates role definitions with specific permissions*
2. *Administrator assigns roles to users*
3. *User logs into the system*
4. *System loads appropriate permissions based on user's assigned roles*
5. *System customizes the interface to show only authorized features*
6. *System permits or denies access to operations based on role permissions*
7. *System logs access attempts and permission violations*

4.6.3 Functional Requirements

RBAC-1: The system shall provide predefined roles with appropriate permission sets for common job functions (Administrator, Operator, Auditor, etc.).

RBAC-2: The system shall allow creation of custom roles with granular permission assignments.

RBAC-3: The system shall support assignment of multiple roles to individual users.

RBAC-4: The system shall enforce permission requirements for all system operations.

RBAC-5: The system shall customize the user interface to display only features and operations authorized for the user's roles.

RBAC-6: The system shall log all permission denials with user information and attempted operation.

RBAC-7: The system shall support temporary role assignments with automatic expiration.

4.7 Load Balancing

4.7.1 Description and Priority

The Load Balancing feature monitors resource utilization across server groups and implements load distribution strategies to optimize resource allocation.

Priority: Medium

4.7.2 Stimulus/Response Sequences

1. *System continuously monitors resource utilization across server groups*
2. *System detects imbalanced resource allocation*
3. *System evaluates load balancing options based on configured policies*
4. *System recommends or automatically implements load balancing actions*
5. *System executes the necessary commands to redistribute workloads*
6. *System records the before and after metrics for performance analysis*
7. *System notifies administrators of load balancing operations*

4.7.3 Functional Requirements

LB-1: The system shall monitor resource utilization metrics (CPU, memory, disk I/O, network) across defined server groups.

LB-2: The system shall detect resource imbalances based on configurable thresholds.

LB-3: The system shall support both advisory mode (recommendations only) and automatic mode for load balancing operations.

LB-4: The system shall implement load balancing strategies including workload migration, service redistribution, and traffic routing.

LB-5: The system shall record detailed metrics before and after load balancing operations for performance analysis.

LB-6: The system shall provide rollback capabilities for unsuccessful load balancing operations.

LB-7: The system shall maintain a history of load balancing events with complete execution details.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

a. *The system shall support monitoring of at least 1,000 servers simultaneously.*
b. *Dashboard updates shall occur within 1 second of data refresh.*
c. *The system shall process and display alerts within 5 seconds of threshold violation.*
d. *The monitoring engine shall consume less than 0.5% CPU and 3% memory on monitored systems.*
e. *The database shall support storage of at least 12 months of historical metrics with no degradation in query performance.*
f. *API endpoints shall respond within 300ms under normal load conditions.*
g. *The system shall maintain 99.9% uptime for core monitoring functions.*
h. *The system shall be capable of processing at least 10,000 metrics per second at peak load.*

5.2 Safety Requirements

a. *The system shall implement safeguards against malicious configuration changes that could impact system stability.*
b. *Automated tasks shall include validation checks to prevent undesired operations.*
c. *Critical system changes shall require confirmation and/or approval workflows.*
d. *The system shall maintain backups of configuration data before applying changes.*
e. *Load balancing operations shall include health checks before and after execution.*

5.3 Security Requirements

a. *All user passwords shall be stored using industry-standard hashing algorithms (bcrypt).*
b. *All data transmissions shall be encrypted using TLS 1.3 or later.*
c. *The system shall enforce strong password policies with configurable complexity requirements.*
d. *User sessions shall automatically expire after a configurable period of inactivity (default: 1 hour).*
e. *The system shall implement protection against common web vulnerabilities (SQL injection, XSS, CSRF).*
f. *The system shall support multi-factor authentication for administrative access.*
g. *The system shall maintain comprehensive audit logs of all security-relevant events.*
h. *The system shall support integration with enterprise identity management systems.*

5.4 Software Quality Attributes

a. ***Reliability**:*
- *The system shall achieve 99.9% uptime for core monitoring functions.*
- *The system shall include self-healing mechanisms for common failure scenarios.*

b. ***Maintainability**:*
- *The system shall follow a microservices architecture for component isolation.*
- *The system shall include comprehensive logging for troubleshooting.*
- *The system shall support configuration updates without service interruption.*

c. ***Scalability**:*
- *The system shall support horizontal scaling through addition of monitoring nodes.*
- *Database components shall support sharding for large-scale deployments.*
- *The system shall implement caching mechanisms for frequently accessed data.*

d. ***Usability**:*
- *The user interface shall follow consistent design patterns.*
- *Common tasks shall be completable in 3 clicks or fewer.*
- *The system shall provide contextual help for complex features.*
- *The system shall support keyboard shortcuts for common operations.*

e. *I**nteroperability**:*
- *The system shall provide standardized APIs for integration with external systems.*
- *The system shall support common monitoring protocols (SNMP, WMI, JMX).*
- *The system shall implement standard data exchange formats (JSON, XML).*

5.5 Business Rules

a. *Critical system alerts must be acknowledged within specified SLA timeframes based on severity.*

b. *Configuration changes to production systems must follow the approved change management process.*

c. *Access to sensitive monitoring data must be restricted based on user role and need-to-know principles.*

d. *System backup and recovery procedures must be tested regularly according to the defined schedule.*

e. *Security-related incidents must be escalated to the security team within 30 minutes of detection.*

6. Other Requirements

6.1 Database Requirements

InfraVision requires a normalized relational database schema as described in Appendix C. The database must support:

i. *Foreign key constraints to maintain referential integrity*
ii. *Transaction support for maintaining data consistency*
iii. *Indexing for optimized query performance*
iv. *Partitioning for efficient storage of historical data*
v. *Backup and recovery mechanisms*

6.2 Internationalization Requirements

i. *The system shall support localization of the user interface.*
ii. *Date and time formats shall adapt to the user's locale settings.*
iii. *The system shall support Unicode character encoding for international text.*

6.3 Legal and Compliance Requirements

i. *The system shall maintain audit trails in compliance with IT governance frameworks.*
ii. *The system shall support data retention policies in accordance with relevant regulations.*
iii. *The system shall include privacy controls for personally identifiable information.*

6.4 Installation and Deployment Requirements

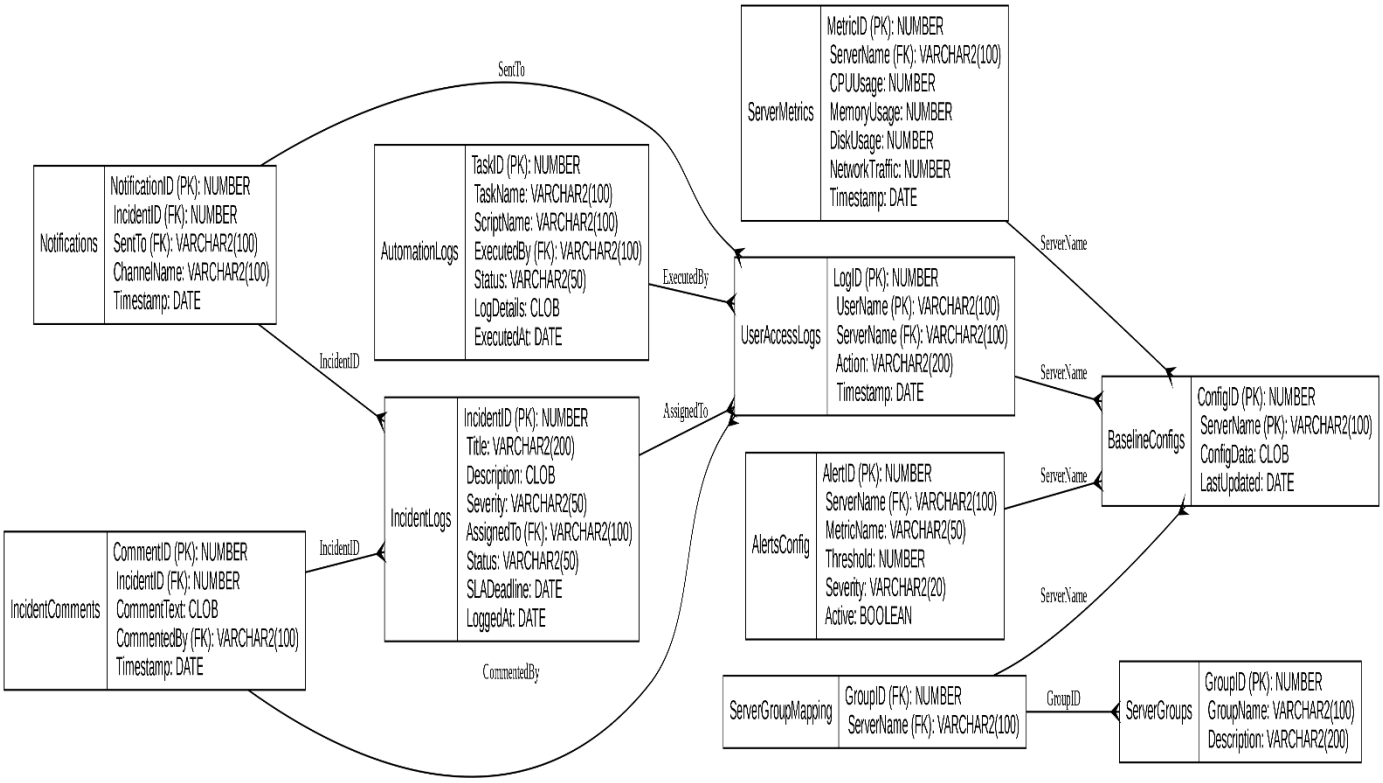
i. *The system shall provide automated installation scripts for supported platforms.*
ii. *The system shall support both on-premises and cloud deployment models.*
iii. *The system shall include migration tools for transitioning from legacy monitoring systems.*

Appendix A: Glossary

- *Alert: A notification generated when a monitored metric exceeds defined thresholds.*
- *Baseline Configuration: An approved, standard configuration for a server or application.*
- *Configuration Drift: Unauthorized or unintended changes from the baseline configuration.*
- *Incident: A record of a system issue requiring investigation and resolution.*
- *Load Balancing: The distribution of workloads across multiple computing resources.*
- *Metric: A measurable value related to system performance or health.*
- *Role-Based Access Control (RBAC): A method of restricting system access based on user roles.*
- *Service Level Agreement (SLA): A commitment to maintain specified performance levels.*
- *Threshold: A predefined limit for a metric that triggers alerts when exceeded.*

Appendix B: Analysis Models

Initial ER Diagram:



@TODO: Add the ER Diagram of Normalized Table

Appendix C: To Be Determined List

The following items require further clarification or definition and will be addressed in subsequent versions of this SRS document:

1. *Specific hardware requirements for the InfraVision server components beyond the minimum specifications*
2. *Detailed data retention policies for monitoring metrics and historical data*
3. *Thresholding values for default monitoring alerts across different server types*
4. *List of supported operating system versions for agent deployment*
5. *Integration specifications for third-party ticketing systems*
6. *Authentication method details for cloud service provider API access*
7. *Implementation approach for predictive maintenance algorithms mentioned in future enhancements*
8. *Performance benchmark methodology for measuring dashboard response times*
9. *Detailed security requirements for data at rest encryption*

10. *Specific browser compatibility requirements for mobile access*
11. *Recovery time objectives (RTO) and recovery point objectives (RPO) for the system*
12. *Detailed specifications for the historical trend analysis algorithms*
13. *Maximum number of concurrent users the system must support*
14. *Backup and disaster recovery procedures for the database components*
15. *Specific compliance framework requirements (e.g., which aspects of HIPAA must be supported)*