Software Requirements Specification (SRS)

Project: Intelligent Incident & Log Management Platform

1. Introduction

This document describes the software requirements for the Intelligent Incident and Log Management Platform. The system is designed to collect, store, analyze, and group application logs into meaningful incidents, providing actionable insights for developers and operators. The platform is intended to simulate a real-world, production-grade backend system using only open-source technologies and free-tier services.

2. Overall Description

The platform ingests structured logs from multiple services through REST APIs. Logs are validated, normalized, stored in a relational database, and processed to detect anomalies and group related events into incidents. An optional AI-assisted summarization module generates human-readable explanations while enforcing strict validation and fallback mechanisms.

3. User Classes and Characteristics

1. System Administrator: Manages users, services, and configuration.
2. Developer: Views logs, investigates incidents, and analyzes root causes.
3. Viewer: Read-only access to logs and incident summaries.

4. Functional Requirements

1. The system shall provide REST APIs for log ingestion.
2. The system shall validate and normalize incoming log data.
3. The system shall store logs with indexed fields for efficient querying.
4. The system shall group related logs into incidents using correlation logic.
5. The system shall detect anomalies based on configurable thresholds.
6. The system shall generate incident summaries using AI or rule-based methods.
7. The system shall support authentication and role-based authorization.

5. Non-Functional Requirements

1. Performance: Log ingestion shall handle burst traffic without data loss.
2. Reliability: Partial failures shall not crash the system.
3. Security: APIs shall be protected using JWT-based authentication.
4. Scalability: The system shall support horizontal scaling at the service level.
5. Usability: APIs shall be documented using OpenAPI/Swagger.

6. External Interface Requirements

The system exposes RESTful APIs over HTTP/HTTPS. A minimal web-based dashboard provides log search, incident listing, and incident detail views. Database access is internal and not exposed directly.

7. Assumptions and Constraints

1. The system uses only open-source tools and free-tier services.
2. AI-assisted features rely on locally hosted language models when enabled.
3. The platform targets small to medium-scale applications for demonstration purposes.

8. Future Enhancements

1. Integration with message queues for high-throughput ingestion.
2. Advanced anomaly detection using machine learning models.
3. Multi-tenant support and advanced access controls.