A One-Pager Software Requirements Specification (SRS) Format

Project Name: Fault Management

Project Overview:

- **Purpose:** Generate and distribute data integrity alerts with GUI configuration.
- **Target Audience:** Developers implementing the following requirements.
- **Scope:** This project only contains fault detection, isolation, and alerting of generic data input. It is meant to be integrated into a larger data platform.

Functional Requirements:

- Data Ingestion:
- FR1.1: The system shall support ingestion of CSV data in both batch and streaming modes.
- FR1.2: The system shall automatically identify and map column names from incoming CSV data.
- Event and Alarm Generation:
- FR2.1: The system shall allow users to define conditions for event generation based on specific cell values (e.g., Table X, Column C, Value > Threshold).
- FR2.2: The system shall support conditions based on calculated values such as Key Performance Indicators (KPIs).
- FR2.3: The system shall generate events when the specified conditions are met within a defined time window.
- FR2.4: The system shall generate alarms for events that require user attention and have associated clearing events.
- GUI Configuration:
- FR3.1: The system shall provide a graphical user interface (GUI) for configuring alert conditions, including selection of tables, columns, thresholds, and calculation formulas.
- FR3.2: The GUI shall allow users to specify delivery channels for alerts, including Slack, email, etc.
- FR3.3: The GUI shall display a section listing all current alarms with options to clear or snooze them.
-
- Alert Delivery:
- FR4.1: The system shall deliver alerts through multiple channels as configured by the user.
- FR4.2: Supported delivery channels shall include Slack, email, and other messaging platforms.
-
- Event and Alarm Management:
- FR5.1: The system shall log all generated events and alarms for audit and review purposes.
- FR5.2: The system shall associate alarms with their corresponding clearing events to manage their lifecycle.

Non-Functional Requirements:

- **Performance:** [Specify performance criteria, e.g., response time, throughput]
- **Security:** [Outline security measures or compliance standards]
- **Usability:** [Describe user interface guidelines or accessibility requirements]
- **Reliability:** [Define reliability metrics or fault tolerance requirements]
- **Maintainability:** [Specify maintainability criteria, e.g., code quality standards]

Performance:

NFR1: The system shall process incoming CSV data and generate alerts with a latency of no more than 2 seconds.

NFR2: The system shall support high-throughput data ingestion, capable of handling at least 10,000 data points per second.

Security:

NFR3: The system shall implement user authentication and role-based access control to secure configuration settings and data access.

NFR4: Data in transit and at rest shall be encrypted using industry-standard encryption protocols.

Usability:

NFR5: The GUI shall be intuitive and user-friendly, adhering to standard UI/UX design principles.

NFR6: The system shall provide tooltips and help documentation for all configurable options.

Reliability:

NFR7: The system shall achieve 99.9% uptime availability.

NFR8: The system shall ensure data integrity during ingestion and processing, with mechanisms to detect and recover from data corruption.

Maintainability:

NFR9: The codebase shall follow clean code principles and be well-documented to facilitate maintenance and future enhancements.

NFR10: The system shall include comprehensive logging and monitoring to assist in troubleshooting and performance tuning.

Assumptions and Dependencies:

- AD1: Incoming data will be in CSV format and structured with identifiable column names.
- AD2: The system depends on external data pipelines for CSV data ingestion.
- AD3: Alert delivery relies on third-party services like Slack APIs and email servers.
- AD4: Users have the necessary permissions and network access to configure and receive alerts through the specified channels.

Acceptance Criteria:

- AC1: Users can configure alert conditions via the GUI without requiring technical assistance.
- AC2: The system generates events and alarms accurately based on the configured conditions and time windows.
- AC3: Alerts are delivered promptly through all user-configured channels when conditions are met.

- AC4: Users can view, clear, and snooze alarms from the GUI effectively.
- AC5: The system handles both batch and streaming CSV data without data loss or significant processing delays.
- AC6: Security measures are validated through penetration testing, ensuring unauthorized access is prevented.

Additional Considerations:

- Future Expansion:
- Support for semi-structured and unstructured data formats may be added in subsequent versions.
- Integration with data visualization tools for more comprehensive data analysis.
- Compliance:
- Ensure compliance with data protection regulations like GDPR if handling personal data.
- Internationalization:
- Consider localization support for multiple languages in the GUI.