A logo of a software company

Description automatically generated

**​**

**[CUSTOMER NAME – SYSTEM NAME]**

**​Requirement Specifications**

**​For <<App Name>>**

**​Vietnam, May  2024**

**​Approval Page**

​The endorsement on this document by authorized [Customer Name] representative indicates [Customer Name] and FPT’s agreement on the “[Project Name] Requirement Specifications” document.

​

|  |  |  |  |
| --- | --- | --- | --- |
| **​Prepared  by :** | ​ (FPT)  ​Business Analyst | ​Signature: | ​  ​\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **​** | ​ | ​Date: | ​  ​\_\_\_\_  /\_\_\_\_   / \_\_\_\_ |
| **​** | **​** | ​ | ​ |
| **​** | **​** | ​ | ​ |
| **​Reviewed  by :** | ​(FPT)  ​Project Manager | ​Signature: | ​  ​\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **​** | **​** | ​Date: | ​  ​\_\_\_\_  /\_\_\_\_   / \_\_\_\_ |
| **​** | **​** | ​ | ​ |
| **​Supported by:** | ​ (Customer Name)  ​ | ​Signature: | ​  ​\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **​** | ​ | ​Date: | ​  ​\_\_\_\_  /\_\_\_\_   / \_\_\_\_ |
| **​** | ​ | ​ | ​ |
| **​** | ​ | ​ | ​ |
| **​Approved by :** | ​ (Customer Name)  **​** | ​Signature: | ​  ​\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **​** | **​** | ​Date: | ​  ​\_\_\_\_  /\_\_\_\_   / \_\_\_\_ |

# Introduction

* 1. Template
  2. Overview

# Software Requirements Document (SRD) for Sausage Production Quality Control Software

## Objective

Develop a robust, scalable, and secure quality control software system for sausage production that integrates real-time data processing, anomaly detection, traceability, and reporting with configurable management tools.

## Components and Requirements

### 1. Data Integration

* **Objective**: To create a unified data pipeline for real-time ingestion and distribution of manufacturing data.
* **Requirements**:
  + Use Apache Kafka for the data pipeline to handle varied data volumes.
  + Develop source connectors for different data types and systems, standardizing data formats.
  + Ensure parallel capture off live systems to avoid downtime.
  + Implement lambda architecture with change data capture for syncing back to Kafka for data consistency.

### 2. Real-Time Quality Control Processing

* **Objective**: To process and analyze data in real-time for identifying defects and quality issues.
* **Requirements**:
  + Evaluate and select a stream processing engine (Apache Spark Structured Streaming or Apache Flink).
  + Implement statistical models, rules, and machine learning algorithms for anomaly detection.
  + Design flexible interfaces and APIs for easy configuration and management of detection models by quality experts.

### 3. Security Architecture

* **Objective**: To ensure the security of the system through authentication, access controls, data encryption, and compliance with regulatory standards.
* **Requirements**:
  + Integrate with enterprise identity providers using SAML or OIDC for single sign-on.
  + Implement role-based access control (RBAC) policies.
  + Encrypt all data in transit and at rest.
  + Centralize audit logging and monitoring for traceability and anomaly detection.
  + Design the system to meet ISO 27001, NIST 800-171, and other relevant compliance standards.

### 4. Traceability and Reporting

* **Objective**: To provide end-to-end traceability from raw materials to finished products and generate real-time quality KPI dashboards and reports.
* **Requirements**:
  + Develop a data model linking raw materials, production processes, and finished products.
  + Implement an auditing application for traceability.
  + Construct customizable real-time dashboards and reports for different user roles.

### 5. Configuration and Management Tools

* **Objective**: To offer intuitive tools for quality teams to configure and manage detection rules and models.
* **Requirements**:
  + Develop a desktop application and lower-level APIs for rule and model configuration.
  + Ensure the system includes version control, audit trails, and testing environments.
  + Incorporate features for full auditing of configuration changes and rule simulation before production deployment.

### 6. Non-Functional Requirements

* **Objective**: To achieve worldwide scale, high availability, disaster recovery, and optimal performance.
* **Requirements**:
  + Deploy a multi-region, multi-cloud architecture for data pipeline and processing tiers.
  + Ensure high availability and disaster recovery using cloud services.
  + Comply with data residency laws and performance targets for real-time monitoring.

### Next Steps

* Formalize architecture into design documents.
* Establish core development teams for each workstream.
* Prioritize prototyping and proof of concepts.
* Engage in iterative requirements gathering with stakeholder teams.

This document outlines the high-level requirements for the sausage production quality control software, capturing the essence of the technical planning and architecture review discussions. It will serve as a foundational guide for the development teams as they move forward with the project. A screenshot of a computer screen

Description automatically generated

# 2. High-level requirements

## 2.1 Object Relationship Diagram

### **Object Description:**

Data Objects

|  |  |  |
| --- | --- | --- |
| Item # | Object | Description |
| 1 | Data Pipeline | A unified real-time system for data ingestion and distribution using Apache Kafka. |
| 2 | Source Connectors | Custom connectors for integrating various data types and systems into the data pipeline. |
| 3 | Lambda Architecture | A data-processing architecture designed for handling massive quantities of data by using a combination of batch and stream-processing methods. |
| 4 | Stream Processing Engine | A core component for real-time data processing and analysis, selected from Apache Spark Structured Streaming or Apache Flink. |
| 5 | Anomaly Detection Models | Statistical models, rules, and machine learning algorithms for identifying defects and quality issues in real-time. |
| 6 | Security Architecture | The framework ensuring system security through authentication, access control, data encryption, and compliance. |
| 7 | Identity Providers | External services integrated for single sign-on, using SAML or OIDC protocols. |
| 8 | RBAC Policies | Role-based access control policies for managing system access and permissions. |
| 9 | Data Model | A structured representation linking raw materials, production processes, and finished products for traceability. |
| 10 | Auditing Application | A specialized application developed for tracking and maintaining traceability across the production process. |
| 11 | Dashboards and Reports | Customizable real-time visualizations and reports for monitoring quality KPIs. |
| 12 | Configuration Tools | Desktop applications and APIs designed for configuring and managing detection rules and models. |
| 13 | Multi-Region Architecture | A deployment strategy ensuring worldwide scale, high availability, and disaster recovery. |
| 14 | Cloud Services | Services utilized for achieving high availability, disaster recovery, and compliance with data residency laws. |

Actor Objects

|  |  |  |
| --- | --- | --- |
| Item # | Object | Description |
| 1 | Sensor | Interfaces with the data integration component to provide real-time manufacturing data. |
| 2 | Worker | Interacts with the configuration and management tools for rule and model adjustments. |
| 3 | Customer | Uses the traceability and reporting features to verify product quality and traceability. |
| 4 | QC Specialist | Utilizes real-time quality control processing and configuration tools for quality assurance. |
| 5 | Supervisor | Oversees the entire quality control process, using reports and dashboards for insights. |
| 6 | System Administrator | Manages the security architecture, ensuring system integrity and compliance. |
| 7 | Data Analyst | Works with the traceability and reporting component to generate and interpret reports. |
| 8 | IT Developer | Engages in system development, especially in data integration and processing components. |
| 9 | Compliance Officer | Ensures the software meets all regulatory compliance standards mentioned. |
| 10 | External Auditor | Uses auditing and reporting features for external verification and compliance checks. |

External System Objects

|  |  |  |
| --- | --- | --- |
| Item # | Object | Description |
| 1 | Apache Kafka | A distributed streaming platform used for building the real-time data pipeline. |
| 2 | Stream Processing Engine (Apache Spark Structured Streaming/Apache Flink) | Software platforms for processing and analyzing streaming data in real-time. |
| 3 | Enterprise Identity Providers (SAML/OIDC) | Systems for managing user identities and enabling single sign-on (SSO) across applications. |
| 4 | Role-Based Access Control (RBAC) Policies | A method for restricting system access to authorized users based on their role. |
| 5 | Encryption Technologies | Technologies used to encrypt data in transit and at rest to ensure confidentiality. |
| 6 | Audit Logging and Monitoring Systems | Systems for tracking and recording user activities and system changes for security and compliance. |
| 7 | Compliance Standards (ISO 27001, NIST 800-171) | International and national standards for managing information security. |
| 8 | Auditing Application | A software application designed for tracking and verifying data for traceability. |
| 9 | Desktop Application and APIs | Software tools for configuring and managing detection rules and models. |
| 10 | Cloud Services (Multi-region, Multi-cloud Architecture) | Cloud computing services for hosting data and applications, ensuring high availability and disaster recovery. |
| 11 | Data Residency Laws Compliance | Legal requirements for storing data within specific jurisdictions. |

### 2.2 Workflow Generated User Workflow:

Given the requirements and actor interactions outlined in the Software Requirements Document (SRD) for Sausage Production Quality Control Software, the user workflow can be detailed as follows:

### Workflow Overview

#### Stage 1: System Setup and Configuration

1. **System Administrator**:
   * Initializes the software system, setting up server and database requirements.
   * Configures the Apache Kafka data pipeline, ensuring secure connectivity to data sources.
   * Implements security measures, including data encryption and access controls, integrating with enterprise identity providers for SSO.
2. **IT Developer**:
   * Develops source connectors for real-time data ingestion from various systems, standardizing data formats.
   * Selects and configures the stream processing engine (Apache Spark Structured Streaming or Apache Flink) for real-time data processing.
   * Implements lambda architecture for data consistency.
3. **QC Specialist and Worker**:
   * Use configuration tools to define initial quality control rules and models.
   * Engage in initial testing of detection algorithms with simulated data.

#### Stage 2: Real-Time Quality Control Processing

1. **Sensor**:
   * Begins transmitting real-time manufacturing data to the system via Apache Kafka.
2. **System** (Automated Process):
   * Processes incoming data streams using configured statistical models and machine learning algorithms to detect anomalies.
   * Alerts are generated for detected quality issues and are logged for traceability.
3. **QC Specialist**:
   * Reviews alerts and refines detection models and rules as necessary.
   * Collaborates with the **Worker** to adjust manufacturing processes based on quality insights.

#### Stage 3: Traceability and Reporting

1. **Data Analyst**:
   * Utilizes the traceability component to link raw materials, production processes, and finished products.
   * Generates real-time quality KPI dashboards and detailed reports for analysis.
2. **Supervisor**:
   * Monitors dashboards and reports for overall quality control insights.
   * Makes strategic decisions based on quality trends and reports.

#### Stage 4: Compliance and Audit

1. **Compliance Officer**:
   * Reviews system configurations and audit logs to ensure compliance with ISO 27001, NIST 800-171, and other standards.
   * Works with the **System Administrator** to address any compliance gaps.
2. **External Auditor**:
   * Conducts external audits using the system’s auditing and reporting features.
   * Verifies compliance with regulatory standards and industry best practices.

#### Stage 5: Continuous Improvement and Management

1. **All Users**:
   * Engage in continuous feedback loops to identify system improvements and enhancements.
   * Update and refine quality control models and rules based on new data and insights.
2. **IT Developer and System Administrator**:
   * Implement system updates and upgrades, ensuring high availability and disaster recovery capabilities.
   * Monitor system performance, making adjustments as needed to meet real-time monitoring performance targets.

### Conditional Flows

* **If** a quality issue is identified, **then** the QC Specialist and Worker are alerted to take corrective action, and the issue is logged for traceability.
* **If** a compliance gap is identified, **then** the System Administrator and Compliance Officer collaborate to address the issue, ensuring continued compliance.
* **If** performance targets are not met, **then** the IT Developer optimizes data processing components and infrastructure to improve system performance.

This workflow integrates the requirements and interactions of various actors to ensure the effective operation of the sausage production quality control software, from system setup and real-time processing to compliance and continuous improvement.

## 2.3 State Transition

### Generated State Transitions:

Based on the requirements plan and data objects defined in the Software Requirements Document for Sausage Production Quality Control Software, the state transition steps for each component can be outlined as follows:

### 1. Data Integration

**Initial State**: Data is disparate and unstandardized across various sources.

**Transition Steps**:

1. **Implement Apache Kafka**: Set up the Kafka cluster to serve as the backbone of the data pipeline.
2. **Develop Source Connectors**: Create connectors for each data source type to ingest data into Kafka, ensuring data is standardized into a uniform format.
3. **Enable Parallel Capture**: Configure the system to capture data in parallel from live systems, minimizing downtime.
4. **Apply Lambda Architecture**: Integrate batch and stream processing to ensure data consistency and real-time processing capabilities.

**Final State**: A unified, real-time data pipeline that standardizes and distributes manufacturing data efficiently.

### 2. Real-Time Quality Control Processing

**Initial State**: Data is not being analyzed in real-time for quality control.

**Transition Steps**:

1. **Select Stream Processing Engine**: Choose between Apache Spark Structured Streaming or Apache Flink based on performance and scalability requirements.
2. **Implement Detection Models**: Develop and integrate statistical models, rules, and machine learning algorithms for anomaly detection within the selected stream processing engine.
3. **Design Configuration Interfaces**: Create flexible interfaces and APIs for quality experts to manage and configure detection models.

**Final State**: A system capable of real-time data processing and analysis for immediate quality control actions.

### 3. Security Architecture

**Initial State**: The system lacks comprehensive security measures.

**Transition Steps**:

1. **Integrate Identity Providers**: Set up SAML or OIDC integration for single sign-on capabilities.
2. **Implement RBAC Policies**: Define and enforce role-based access control policies to manage system permissions.
3. **Encrypt Data**: Ensure all data in transit and at rest is encrypted.
4. **Centralize Logging and Monitoring**: Establish a system for audit logging and security monitoring.
5. **Compliance Alignment**: Design the system to adhere to ISO 27001, NIST 800-171, and other standards.

**Final State**: A secure, compliant, and monitored quality control software system.

### 4. Traceability and Reporting

**Initial State**: Limited traceability and reporting capabilities.

**Transition Steps**:

1. **Develop Data Model**: Construct a data model that links raw materials, production processes, and finished products.
2. **Implement Auditing Application**: Develop an application specifically for maintaining traceability throughout the production process.
3. **Create Dashboards and Reports**: Build customizable dashboards and reports for real-time monitoring of quality KPIs.

**Final State**: Enhanced end-to-end traceability with robust reporting tools for quality monitoring.

### 5. Configuration and Management Tools

**Initial State**: No dedicated tools for configuring detection rules and models.

**Transition Steps**:

1. **Develop Configuration Tools**: Create a desktop application and APIs for rule and model configuration.
2. **Ensure Version Control and Testing**: Incorporate version control, audit trails, and testing environments within the configuration tools.
3. **Rule Simulation and Auditing**: Add features for simulating rule changes and fully auditing configuration adjustments before deployment.

**Final State**: Intuitive and comprehensive tools for quality teams to manage detection rules and models effectively.

### 6. Non-Functional Requirements

**Initial State**: The system is localized and not optimized for global scale or disaster recovery.

**Transition Steps**:

1. **Deploy Multi-Region Architecture**: Establish a multi-cloud, multi-region deployment to ensure global scalability and availability.
2. **Leverage Cloud Services**: Use cloud services for high availability, disaster recovery, and compliance with data residency laws.
3. **Performance Optimization**: Continuously monitor and optimize the system for meeting real-time monitoring performance targets.

**Final State**: A scalable, highly available, and compliant system capable of worldwide operations.

These state transition steps provide a roadmap for developing the Sausage Production Quality Control Software from its initial planning stages to a fully operational state, ensuring all requirements are systematically addressed.

## 2.4 Use Case

## 2.5 Permission Matrix

# 3. Use Case Specifications