

**Software Requirements**

**Specification**

**for**

**TEIM-Connect**

**Version 2.0 approved**

**Prepared by Team Cummins**

**CCOEW 16/08/2024**

**Software Requirements Specification for TEIM-Connect Page ii**

# Table of Contents

[**Table of Contents ii**](#_heading=h.gjdgxs)

[**Revision History ii**](#_heading=h.30j0zll)

[**1. Introduction 1**](#_heading=h.1fob9te)

[1.1 Purpose 1](#_heading=h.3znysh7)

[1.2 Document Conventions 1](#_heading=h.2et92p0)

[1.3 Intended Audience and Reading Suggestions 1](#_heading=h.tyjcwt)

[1.4 Product Scope 1](#_heading=h.3dy6vkm)

[1.5 References 1](#_heading=h.1t3h5sf)

[**2. Overall Description 2**](#_heading=h.4d34og8)

[2.1 Product Perspective 2](#_heading=h.2s8eyo1)

[2.2 Product Functions 2](#_heading=h.17dp8vu)

[2.3 User Classes and Characteristics 2](#_heading=h.3rdcrjn)

[2.4 Operating Environment 2](#_heading=h.26in1rg)

[2.5 Design and Implementation Constraints 2](#_heading=h.lnxbz9)

[2.6 User Documentation 2](#_heading=h.35nkun2)

[2.7 Assumptions and Dependencies 3](#_heading=h.1ksv4uv)

[**3. External Interface Requirements 3**](#_heading=h.44sinio)

[3.1 User Interfaces 3](#_heading=h.2jxsxqh)

[3.2 Hardware Interfaces 3](#_heading=h.z337ya)

[3.3 Software Interfaces 3](#_heading=h.3j2qqm3)

[3.4 Communications Interfaces 3](#_heading=h.1y810tw)

[**4. System Features 4**](#_heading=h.4i7ojhp)

[4.1 System Feature 1 4](#_heading=h.2xcytpi)

[4.2 System Feature 2 (and so on) 4](#_heading=h.1ci93xb)

[**5. Other Nonfunctional Requirements 4**](#_heading=h.3whwml4)

[5.1 Performance Requirements 4](#_heading=h.2bn6wsx)

[5.2 Safety Requirements 5](#_heading=h.qsh70q)

[5.3 Security Requirements 5](#_heading=h.3as4poj)

[5.4 Software Quality Attributes 5](#_heading=h.1pxezwc)

[5.5 Business Rules 5](#_heading=h.49x2ik5)

[**6. Other Requirements 5**](#_heading=h.2p2csry)

[**Appendix A: Glossary 5**](#_heading=h.147n2zr)

[**Appendix B: Analysis Models 5**](#_heading=h.3o7alnk)

[**Appendix C: To Be Determined List 6**](#_heading=h.23ckvvd)

# Revision History

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for developing an industrial safety alert system that receives inputs from CCTV cameras, uses AI to diagnose safety bypasses and missing steps during production, and generates notifications to prevent human fatalities and production losses. This document serves as a comprehensive guide for the development, implementation, and maintenance of the system, ensuring that all stakeholders have a clear understanding of the project’s objectives and requirements.

## 1.2 Document Conventions

**Terminology**:

* **System**: Refers to the industrial safety alert system being developed.
* **AI Model**: The artificial intelligence model responsible for diagnosing safety issues.
* **Notification**: Alerts generated by the system, including SMS, email, and siren/bell triggers.
* **CCTV**: Closed-Circuit Television cameras used for monitoring production areas.
* **Factual Report**: A detailed report generated by the system on any detected missing steps

## 1.3 Intended Audience and Reading Suggestions

**Intended Audience**:

* **Project Managers**: To ensure that the system meets project timelines and budget constraints.
* **System Architects**: To design the overall system architecture in accordance with the requirements.
* **Software Developers**: To develop the system according to the outlined specifications.
* **Quality Assurance Teams**: To validate the system against the requirements.
* **End Users**: To understand the system’s capabilities and limitations.
* **Safety Officers and Management**: To monitor and respond to safety alerts.

**Reading Suggestions**:

* **Project Managers**: Focus on the Purpose, System Features, and Non-functional Requirements sections.
* **System Architects**: Pay attention to the Overall Description and External Interface Requirements sections.
* **Software Developers**: Study the System Features and Interface Requirements in detail.
* **Quality Assurance Teams**: Review the System Attributes and Non-functional Requirements for testing guidelines.
* **End Users and Safety Officers**: Concentrate on the System Features and Factual Reports sections.

## 1.4 Product Scope

The industrial safety alert system aims to enhance safety and operational efficiency within industrial production environments by leveraging advanced AI and CCTV inputs. The system's primary functions include:

* **CCTV Input Analysis**: Monitor production areas in real-time, diagnosing potential safety bypasses or missing steps based on pre- defined safety limits.
* **AI-Powered Diagnosis**: An AI model processes CCTV footage to detect safety protocol violations and missing steps during the production phase.
* **Alert Generation**: The system generates real-time alerts via SMS, email, and physical sirens/bells to notify relevant personnel of potential safety hazards.
* **Reporting**: Automatically generates detailed reports on detected safety bypasses and missing steps, providing actionable insights to prevent future incidents.
* **Scalability**: Designed to be scalable and integrable with existing industrial safety practices and protocols, allowing for future expansion and customization.

## 1.5 References

# 

# 

# 

# 

# 2. Overall Description

## 2.1 Product Perspective

The system will be a web-based application using Django for backend ,UI verse for frontend and MongoDB as the database ,using inputs from CCTV cameras for real-time monitoring and alerting. The application will interact with hardware for triggering alarms and will be deployed in industrial environments.

## 2.2 Product Functions

Analyze real-time footage from CCTV cameras to detect safety violations.Notify stakeholders through SMS, email, and sirens/bells when safety limits are breached.Generate and manage reports of missing steps and safety issues.

## 2.3 User Classes and Characteristics

**Operators**:

* **Role**: Monitor the system’s dashboard, respond to alerts, and verify incidents.
* **Skills**: Basic understanding of system operation and safety procedures.
* **Interface Needs**: Real-time monitoring interface, alert response tools, and incident verification features.

**Administrators**:

* **Role**: Configure system settings, manage notifications, and set safety thresholds.
* **Skills**: Technical expertise in system configuration and understanding of safety parameters.
* **Interface Needs**: Configuration tools, alert management settings, and user management capabilities.

**Maintenance Personnel**:

* **Role**: Oversee system health, perform updates, and ensure system reliability.
* **Skills**: Technical knowledge of system maintenance and troubleshooting.
* **Interface Needs**: System health dashboard, maintenance tools, and update mechanisms.

## 2.4 Operating Environment

**Hardware**:

* **CCTV Cameras**: High-resolution cameras capable of streaming real-time footage.
* **Sirens/Bells**: Audio alert systems for immediate notification.
* **Server Hardware**: Servers to host the application, store data, and manage processing tasks.

**Software**:

* **Web Browsers**: Compatible with major browsers (e.g., Chrome, Firefox, Edge) for accessing the web-based application.
* **Technologies**: MongoDB, Django for building and running the application.
* **Operating Systems**: The application will be compatible with popular operating systems such as Linux and Windows.

## 2.5 Design and Implementation Constraints

**Inputs from Existing CCTV Hardware**:

* The system must be compatible with various brands and models of CCTV cameras used in the industrial environment.

**Compliance with Industrial Safety Standards**:

* The system must adhere to industry-specific safety regulations and standards to ensure it meets safety and legal requirements.

**Real-Time Processing Constraints**:

* The application must be optimized for real-time image processing to ensure timely detection and response to safety violations.

## 2.6 User Documentation

**User Manuals**:

* **Operators**: Detailed instructions on how to monitor the system, respond to alerts, and perform basic troubleshooting.
* **Administrators**: Guidelines for configuring system settings, managing notifications, and setting safety thresholds.

**Online Help System and FAQ**:

* An integrated help system with searchable FAQs, troubleshooting guides, and step-by-step instructions to assist users in navigating and using the system effectively.

## 2.7 Assumptions and Dependencies

**1. Availability of CCTV Cameras**:  
  
 The system assumes that CCTV cameras used in the industrial environment have sufficient resolution and frame rate to capture clear and usable footage for analysis.  
  
**2. Reliable Network Connectivity:** The system depends on stable and high-speed network connectivity to transmit real-time data from cameras to the application and ensure timely alerts.  
  
**3. Compliance with Regulations**:  
  
 The system assumes adherence to relevant industrial safety regulations and standards to ensure that it meets legal and safety requirements.  
  
**4. Integration Capabilities**:  
  
 The system depends on the ability to integrate with existing hardware for alarms and notifications, which may involve custom integration efforts depending on the hardware specifications.

# 

# 3. External Interface Requirements

## 1. User Interface (UI) Requirements:

### a. Dashboard Interface:

* Visual alerts (e.g., red flashing indicators) on the dashboard when safety violations or missing steps are detected.
* An overview panel showing the status of all monitored zones.
* Access to historical data, including video snippets and reports of safety incidents.
* User roles and permissions management, allowing different levels of access and control.
* A settings menu where administrators can define safety thresholds, update AI models, and configure notification preferences.

### b. Notification Interface:

* SMS and email notification templates customizable by the user.
* Acknowledge and dismiss alerts directly from the notification for authorized users.
* Links in notifications to view live feeds or review incident reports.

## 

## 2. Hardware Interface Requirements:

Requirement for running Chrome on devices:

**1.Processor (CPU):**A modern multi-core processor (e.g., Qualcomm Snapdragon, Apple A-series, or equivalent) is recommended to handle web app tasks smoothly, especially those involving complex JavaScript operations.

**2.Memory (RAM):  
Minimum:** 2 GB of RAM is sufficient for basic web apps with limited functionality.  
**Recommended:** 4 GB or more for a better experience, especially for web apps with heavy content, animations, or complex interactions.

**3.Storage: Minimum:** 16 GB of internal storage should be available to ensure the device can cache data, store necessary files, and manage temporary files. **Recommended:** 32 GB or more, especially if the web app handles large data sets or media files.

**4.Graphics (GPU):** A modern GPU that supports WebGL and hardware-accelerated graphics rendering. Most mid-range and high-end smartphones come with capable GPUs that can handle most web app graphics smoothly.

**5.Processing Unit:**  Integration with on-site or cloud-based servers capable of running AI algorithms for image processing and safety violation detection. Edge devices that can process data locally on-site to reduce latency and improve real-time responsiveness.

**6.Alerting Devices:**  Sirens/bells that can be triggered automatically when a safety violation is detected. Interface with programmable logic controllers (PLCs) or other industrial control systems to trigger physical alarms. Physical control panels with buttons for manual override or to acknowledge alarms.

## 3. Software Interface Requirements:

1. **AI Model Integration:**

- API endpoints or SDKs to integrate the AI model with the video feed for real-time analysis.

- Customizable safety parameters that the AI model uses to diagnose images, such as safety lines or required PPE (Personal Protective Equipment).

1. **Third-Party Software Integration:**

- Integration with existing industrial safety management software or ERP systems.

- API for integrating with SMS and email gateway services to send notifications.

- Interface with databases for storing and retrieving incident data, logs, and reports.

1. **Report Generation:**

- A module to generate and export reports in formats like PDF, CSV, or Excel.

- A logging system that tracks incidents, responses, and system operations for audit purposes.

## 4. Communication Interface Requirements:

1. **Network Protocols:**

* Support for standard network protocols (e.g., TCP/IP, HTTP/HTTPS) to ensure reliable communication between cameras, processing units, and user interfaces.
* Secure communication channels, such as SSL/TLS, to protect data transmitted between devices and servers.

2. **Notification Systems:**

* Interface with SMS gateways using protocols like SMPP for sending text message alerts.
* SMTP or other email protocols for sending alert emails.
* Webhooks or RESTful APIs to integrate with third-party services or systems for extended notification capabilities.
* Integration with Industrial Systems:
* Interfaces with SCADA systems or PLCs using industrial protocols (e.g., Modbus, OPC-UA) to trigger alarms or stop production lines in case of severe safety violations.
* Remote monitoring capabilities over a secure VPN or other encrypted channels to allow off-site personnel to access the system.

**5. Camera File Format Requirements:**

* **Function:** Ensure camera file formats support both low and high-resolution images.
* **Requirements:** The system should accept various file formats that can accommodate different resolution levels (e.g., JPEG, PNG for low resolution; RAW, TIFF for high resolution).

**6. Hardware Interface:**

* **Function:** Integration of cameras with machines and mobile devices.
* **Requirements:** Support for one camera per machine with provisions for mobile camera usage. Compatibility with a variety of industrial cameras and mobile devices.

**7. Siren Integration:**

* **Function:** Trigger industry sirens based on safety alerts.
* **Requirements:** Research and integrate appropriate signal inputs (e.g., dry contact, voltage input) to activate sirens. The system should allow configuration of these signals based on the type of siren used.

# 4. System Features

# 

## 4.1 Image and Video Analysis

## Function: Analyze CCTV footage to detect safety breaches, such as unauthorized proximity to machinery or missing production steps.

## Requirements: Real-time image processing, object detection algorithms.

## 4.2 Safety Alert System

## Function: Trigger alarms and send notifications when safety limits are breached.

## Requirements: Integration with alarm systems and communication services.

## 4.3 Notification System

## Function: Send alerts via SMS and email.

## Requirements: Configurable alert thresholds, communication service integration.

## 4.4 Reporting System

## Function: Generate and manage reports of safety violations and missed steps.

## Requirements: Report generation based on analyzed data, user-accessible reporting tools.

**4.5 Autoflush and Record Management:**

* **Function:** Automatically delete records older than six months with an option for manual override.
* **Requirements:** Provide a user-configurable setting to enable or disable the automatic deletion of records. Users should be able to specify custom retention periods.

**4.6 Notification System Integration with Messaging Apps:**

* **Function:** Send safety alerts and notifications via WhatsApp and Telegram.
* **Requirements:** Integration with WhatsApp and Telegram APIs. Customizable message templates for each platform, allowing personalized notifications based on alert severity (Yellow, Green, Red).

**4.7 Alert Templates and Customization:**

* **Function:** Generate and send customized alerts (Yellow, Green, Red) based on defined safety thresholds.
* **Requirements:** Allow administrators to define and customize alert templates based on different cut-off limits. Each alert level should contain precise, non-generalized information.

## 

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The performance requirements are crucial to ensure that the Industrial Safety Alert System operates effectively and responds in real-time to safety violations.

* **Real-Time Image Analysis:** The system must analyze images from CCTV cameras in real-time with a maximum processing delay of 2 seconds per image.
* **Scalability:** The system must support up to 50 simultaneous camera feeds without performance degradation.
* **Alert Latency:** The system must generate and send alerts (SMS, email, sirens) within 5 seconds of detecting a safety violation.
* **System Throughput**: The system must handle up to 10,000 alerts per day without performance issues.

## 5.2 Safety Requirements

Safety is a critical concern for both the system and its users. The following requirements ensure that the system operates without causing harm:

* **System Failures and Redundancy**:The system must include failover mechanisms to maintain operation if a primary component fails. This includes backup servers and redundant data storage.
* **Emergency Handling**:The system must have a manual override feature to disable alerts in case of false positives or emergency maintenance without compromising overall safety.
* **Regulatory Compliance**:The system must comply with industry safety standards and regulations such as ISO 45001 (Occupational Health and Safety Management Systems).
* **Physical Safety**:The system’s hardware components must be housed in secure enclosures to prevent physical tampering or damage.
* **Compliance with Data Protection Regulations**: The system must comply with data protection regulations such as GDPR for handling personal data.

## 5.3 Security Requirements

Security measures are essential to protect the system and the data it handles:

* **Data Encryption:** All data transmitted between CCTV cameras, the system, and notification channels must be encrypted using industry-standard protocols (e.g., TLS 1.2 or higher).
* **User Authentication:** The system must require multi-factor authentication (MFA) for access to configuration and administrative functions.
* **Access Control:** Role-based access control (RBAC) must be implemented to restrict access to different features based on user roles.

## 5.4 Software Quality Attributes

* **Availability:**The system must have an availability of 99.9%, ensuring it is operational and accessible nearly all the time.
* **Reliability:** The system must have a mean time between failures (MTBF) of at least 1,000 hours.
* **Maintainability:** The system must support modular updates and have comprehensive documentation for maintenance tasks.
* **Usability:** The user interface must be intuitive, with a maximum learning curve of 1 hour for new users.
* **Interoperability:** The system must be compatible with major CCTV camera brands and models, as well as various alert systems.

## 5.5 Business Rules

* **Role-Based Access**: Production Managers and Safety Officers can configure safety limits and thresholds, while System Administrators handle system configurations and maintenance.
* **Alert Management:** Alerts must be configured to be sent to predefined recipients based on the severity of the safety violation
* **Incident Review:** All detected incidents must be reviewed and approved by a Safety Officer before being recorded as resolved.

**5.6 Frontend and Backend Tech Stack:**

* **UI:** Utilize UIverse for the frontend design to ensure a responsive and user-friendly interface.
* **Backend:** Implement Django as the framework for the web application, ensuring robust and scalable backend architecture.

# 6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

# Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

# Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

# Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>