Software Requirements Specification

Video-Based Fire Detection and Alarm System Software

Document Version 1 Date: 16/7/2023

Table of Contents:

1. Introduction

2. Purpose

3. Scope

4. System Overview

5. User Requirements

6. Functional Requirements

7. Non-functional Requirements

8. System Architecture

9. System Interfaces

10. Data Requirements

11. Security Requirements

12. Performance Requirements

13. Testing and Quality Assurance

14. Documentation

15. Maintenance and Support

16. Appendix

1. Introduction:

This Software Requirements Specification (SRS) document outlines the requirements for developing a video-based fire detection and alarm system software using Python. The software aims to detect fire in open areas or buildings and trigger alarms to alert occupants, providing emergency safety measures. This section provides an overview of the document and its intended audience.

2. Purpose:

The purpose of this document is to define the functional and non-functional requirements of the video-based fire detection and alarm system software. It serves as a guideline for the development team, stakeholders, and other involved parties, ensuring a common understanding of the system's goals, features, and constraints.

3. Scope:

The scope of the project includes the development of software that utilizes video feeds for fire detection in open areas or buildings. The software should detect fire-related events, trigger appropriate alarms to notify occupants, and provide emergency safety methods. The system will focus on real-time fire detection, alarm activation, and integration with existing fire safety infrastructure.

4. System Overview:

The video-based fire detection and alarm system software will process video inputs from cameras placed in the designated areas. It will analyze the video feeds using computer vision algorithms to detect fire-related events such as flames or smoke. Upon detecting a fire, the system will trigger alarms to notify and alert nearby occupants, facilitating their safety through emergency procedures.

5. User Requirements:

5.1 The system shall provide real-time fire detection in open areas or buildings through video analysis.

5.2 The software should trigger audio and visual alarms to alert occupants nearby upon detecting a fire event.

5.3 The system should integrate with existing fire safety infrastructure, such as sprinkler systems, to initiate fire suppression.

5.4 The software should support various communication methods for alerting occupants, including audio alerts, visual signals, and emergency notifications.

5.5 The system should provide clear and intuitive user interfaces for monitoring fire detection status, accessing live video feeds, and managing system settings.

6. Functional Requirements:

6.1 The software shall capture and process video feeds from designated cameras.

6.2 The system shall apply computer vision algorithms to detect fire-related events, including flames or smoke.

6.3 Upon confirming a fire event, the software shall trigger audio and visual alarms.

6.4 The system should support integration with fire suppression mechanisms to initiate fire suppression procedures.

6.5 The software should provide notifications and alerts to nearby occupants through audio alerts, visual signals, or emergency notifications.

7. Non-functional Requirements:

7.1 Performance:

7.1.1 The system should process video feeds in real-time for prompt fire detection.

7.1.2 The software should have low latency in detecting and triggering alarms.

7.2 Reliability and Availability:

7.2.1 The system should provide continuous monitoring and detection capabilities without frequent downtime.

7.2.2 The software should be reliable and recover gracefully from failures.

7.3 Scalability and Flexibility:

7.3.1 The software should be scalable to accommodate varying numbers of cameras and coverage areas.

7.3.2 The system should be flexible to support different camera types, resolutions, and video formats.

7.4 Security:

7.4.1 The software should ensure the security and privacy of video feeds and user data.

7.4.2 The system should incorporate appropriate measures to prevent unauthorized access and data breaches.

8. System Architecture:

The system architecture will consist of the following components:

- Video Input: Capture and process video feeds from designated cameras.

- Fire Detection Algorithms: Analyze video frames using computer vision algorithms for fire event detection.

- Alarm System: Trigger audio and visual alarms upon detecting a fire event.

- Integration with Fire Suppression Systems: Integrate with fire suppression mechanisms for initiating fire suppression.

- User Interface: Provide user-friendly interfaces for monitoring fire detection status, accessing video feeds, and managing system settings.

9. System Interfaces:

9.1 Hardware Interfaces:

- Cameras: Connect and capture video feeds from designated cameras.

- Alarms: Interface with audio and visual alarm systems for alarm triggering.

- Fire Suppression Systems: Integrate with fire suppression mechanisms for activation.

9.2 Software Interfaces:

- Database Systems: Store and retrieve necessary data, such as fire detection logs or configuration settings.

- Communication Interfaces: Facilitate notifications and alerts to occupants through various communication channels.

10. Data Requirements:

The software may require the following data:

- Video feeds from designated cameras for fire detection and analysis.

- User configurations and settings for customizing system behavior.

- Fire event logs and notifications for monitoring and analysis purposes.

11. Security Requirements:

11.1 Access Control: The system should enforce access control mechanisms to prevent unauthorized access to sensitive data and system functionality.

11.2 Data Encryption: The software should implement encryption algorithms for protecting video feeds and other sensitive data.

11.3 User Authentication: The system should support user authentication mechanisms to ensure only authorized access to the system.

12. Performance Requirements:

12.1 The system should be capable of processing video feeds in real-time for prompt fire detection.

12.2 The software should have low latency in detecting and triggering alarms upon a fire event.

12.3 The system should be able to handle and process video feeds from multiple cameras simultaneously.

13. Testing and Quality Assurance:

13.1 The software should undergo thorough testing to ensure accurate fire detection and reliable alarm triggering.

13.2 The system should be tested for performance, scalability, and compatibility with different camera types and configurations.

13.3 Quality assurance processes should be implemented to maintain software reliability, security, and stability.

14. Documentation:

14.1 User Manual: Provide a comprehensive user manual that explains the software's installation, configuration, and usage instructions.

14.2 Technical Documentation: Document the software architecture, system interfaces, algorithms used, and relevant technical details for future reference and maintenance.

15. Maintenance and Support:

15.1 The software should provide mechanisms for regular software updates and bug fixes.

15.2 Maintenance and support services should be available to address any issues or concerns promptly.

15.3 A helpdesk or support system should be established to assist users with any software-related queries or problems.

16. Appendix:

Include any additional supporting documents, references, or supplementary information related to the software development.

This Software Requirements Specification (SRS) document provides a comprehensive overview of the requirements for the video-based fire detection and alarm system software. It covers the purpose, scope, user requirements, functional and non-functional requirements, system architecture, interfaces, data requirements, security, performance, testing, documentation, and maintenance aspects of the software development project.