|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Requirement Specifications**  **Context-Aware Vision Anomaly Detection System for Educational Environments**  Version: 1.0   |  |  | | --- | --- | | Project Code | F24-17 | | Supervisor | Dr. Jawwad Shamsi | | Co Supervisor | Dr. Burhan Khan | | Project Team | |  |  | | --- | --- | | Sufiyaan Usmani | (21K-3195) | | Yousuf Ahmed Siddiqui | (21K-4594) | | Umer Tariq | (21K-3261) | | | Submission Date |  | |

[Instructions]

* No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
* All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
* This’ Instruction’ section should also be removed in final document.
* MS-Word Reviewing feature must be used to get the document reviewed by supervisors or co-supervisors.

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 1.0 | Sufiyaan Usmani |  | Document Created |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| Dr. Jawwad Shamsi | | Supervisor |
| Dr. Burhan Khan | | Co- Supervisor |
|  | | Shamsi School Director |

Document Sign-Off

[Following table will contain sign-off details of document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.

Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
| 1.0 | Dr. Jawwad Shamsi |  |
| 1.0 | Dr. Burhan Khan |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1. Introduction 7](#_Toc178130213)

[1.1. Purpose of Document 7](#_Toc178130214)

[1.2. Intended Audience 7](#_Toc178130215)  
1.3 Abbreviations ………………………………………………………………………………………...7

[1.4. Document Convention 7](#_Toc178130216)

[2. Overall System Description 8](#_Toc178130217)

[2.1. Project Background 8](#_Toc178130218)

[2.2. Project Scope 8](#_Toc178130219)

[2.3. Not In Scope 8](#_Toc178130220)

[2.4. Project Objectives 8](#_Toc178130221)

[2.5. Stakeholders 8](#_Toc178130222)

[2.6. Operating Environment 8](#_Toc178130223)

[2.7. System Constraints 8](#_Toc178130224)

[2.8. Assumptions & Dependencies 8](#_Toc178130225)

[3. External Interface Requirements 9](#_Toc178130226)

[3.1. Hardware Interfaces 9](#_Toc178130227)

[3.2. Software Interfaces 9](#_Toc178130228)

[3.3. Communications Interfaces 9](#_Toc178130229)

[4. Functional Requirements 10](#_Toc178130230)

[4.1. Functional Hierarchy 10](#_Toc178130231)

[4.2. Use Cases 10](#_Toc178130232)

[4.2.1. [Title of use case] 10](#_Toc178130233)

[5. Non-functional Requirements 11](#_Toc178130234)

[5.1. Performance Requirements 11](#_Toc178130235)

[5.2. Safety Requirements 11](#_Toc178130236)

[5.3. Security Requirements 11](#_Toc178130237)

[5.4. User Documentation 11](#_Toc178130238)

[6. References 12](#_Toc178130239)

[7. Appendices 13](#_Toc178130240)

1. Introduction
   1. Purpose of Document

This document serves as the Software Requirements Specification (SRS) for the development of an anomaly detection system tailored for educational environments. It describes the project’s objectives, scope, functional requirements, system architecture, and performance expectations. The purpose of this document is to ensure that all stakeholders have a clear understanding of the system’s intended functionalities and design requirements, enabling them to contribute effectively to its development, testing, and implementation. Additionally, it serves as a reference for maintaining, enhancing, and scaling the system in the future

* 1. Intended Audience

This document is intended for the following audience groups:

* **Project Team**: The developers and designers working on the project, including team members responsible for implementing, testing, and deploying the system.
* **Supervisor and Co-Supervisor**: Faculty members overseeing the project’s progress, providing guidance, feedback, and evaluation throughout the development process.
* **Jury**: Members of the evaluation committee responsible for assessing the project’s design, functionality, and impact as part of the final year project review.
* **Customer (School Administration)**: Representatives of the school or educational institution for which the system is being developed, including security and administrative staff who will utilize the anomaly detection system to enhance campus safety.

**1.3 Abbreviations**

[Describe the abbreviations use this document.]

* 1. Document Convention

This document adheres to the following formatting conventions to ensure clarity and consistency:

* **Font Type**: Times New Roman
* **Font Size**: Headings in size 16 (bold), Sub-headings in size 12 (bold), Body text in size 12 (regular)
* **Formatting Style**: Numbered sections and sub-sections for structured navigation; keywords and terms are italicized or bolded as needed for emphasis.

1. Overall System Description
   1. Project Background

In recent years, the demand for security and anomaly detection within educational environments has risen, creating a critical need for advanced surveillance systems. Traditional surveillance methods often rely on human observation or simple motion-detection algorithms, which are insufficient for capturing complex, context-dependent anomalies in real time. With the growth of computer vision and deep learning, opportunities have emerged to leverage context-aware computer vision systems to enhance scene understanding and improve the accuracy and robustness of object detection and activity recognition.

This project addresses the need to implement an advanced anomaly detection system that can interpret live video feeds from surveillance cameras within an educational setting. Leveraging advancements in deep learning, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the system will be capable of identifying anomalous patterns and alerting relevant authorities in real time. By integrating contextual information—such as spatial and temporal relationships between objects and activities—into the detection framework, the system will not only detect anomalies but also provide more interpretive insights that can help school authorities respond proactively to potential security risks.

The project is inspired by recent research emphasizing context-aware methodologies. Studies like "Context Understanding in Computer Vision: A Survey" by Wang and Zhu (2023) have highlighted the value of combining spatial and temporal context in anomaly detection and the potential of using edge computing for real-time processing. Additionally, frameworks like the "Context-Aware Co-Supervision for Accurate Object Detection" (Smith et al., 2022) and "Deep Learning Techniques for Context Understanding" (Lee et al., 2023) illustrate the benefits of context-driven detection accuracy and real-time integration. These advancements suggest that context-aware models, particularly in structured environments like schools, can significantly enhance surveillance capabilities.

By implementing an anomaly-detection system that uses context-aware computer vision, this project aims to address security challenges within educational institutions. It will enable monitoring of unusual activities, ensuring a safer environment for students, faculty, and staff. The project’s innovation lies in creating a scalable, interpretable, and highly accurate model that can adapt to diverse security-related anomalies, thereby reducing the burden on human monitoring and allowing authorities to focus on response and prevention.

* 1. Project Scope

The project scope includes developing a context-aware anomaly detection system specifically for educational environments, using a combination of contextual data and visual inputs to identify and alert relevant authorities about potential security risks in real-time. Key functionalities covered within this project are:

* **Data Gathering and Annotation**: Collecting and annotating video data from various locations, including classrooms, corridors, and other common areas, with contextual information on typical activities in these settings.
* **Model Selection and Experimentation**: Evaluating and selecting deep learning models to detect anomalies effectively.
* **Cloud Deployment and Scalability**: Hosting the final model on a cloud platform to ensure dynamic scalability for multiple camera integrations and high-performance processing.
* **Web Application and Dashboard**: Developing a NextJS-based web application with a real-time dashboard to display alerts and provide an interactive interface for administrators to monitor activity.
* **Data Communication via API**: Implementing API for efficient routing and data handling between various components, facilitating seamless interaction between the camera feeds, contextual data, and the cloud-based model.
  1. Not In Scope

The following functionalities are considered out of scope for the current project:

* **Integration with Advanced Security Systems**: The project will not integrate with existing advanced physical security systems, such as door locks, access controls, or alarm systems.
* **Development of Custom Hardware**: This project will not include the development or customization of hardware devices beyond the standard IoT devices used for local inference.
* **Long-Term Data Storage and Analysis**: Long-term storage of video data and extensive historical analysis of activity patterns are not part of the current scope. The system focuses on real-time detection and immediate alerts.
  1. Project Objectives

The primary objectives of this project are to address the unique security needs within educational environments through real-time, context-aware anomaly detection. By leveraging advanced computer vision techniques and contextual data, the project aims to:

* **Enhance Security Monitoring**: Provide automated and continuous surveillance across educational spaces, identifying unusual or potentially risky behavior without relying solely on human oversight.
* **Enable Proactive Responses**: Issue real-time alerts to relevant authorities when anomalies are detected, enabling timely interventions and mitigating potential security risks.
* **Improve Detection Accuracy with Context**: Utilize contextual information and deep learning to distinguish between benign and concerning activities, thereby reducing false alarms and focusing on genuinely suspicious behaviors.
* **Create an Interactive Monitoring Interface**: Develop a user-friendly web application that provides administrators with a clear, actionable interface to monitor live alerts, review recent activity, and respond promptly.
  1. Stakeholders

**Business Stakeholders**

1. **School Administration (Primary Customer)**  
   The main beneficiary and client of the project, responsible for security and operational oversight within the educational facility. They will use the system's monitoring and alert features to respond to security incidents as needed.
2. **Security Personnel**  
   Security staff will monitor the web application’s real-time dashboard and alerts to oversee activities across the campus. They will respond to potential security threats identified by the system.
3. **Supervisory and Academic Staff**  
   Faculty members and other staff may interact with the system in limited capacities, particularly when awareness of security features is necessary. Their feedback on normal campus activities may aid in refining detection accuracy.
4. **Project Supervisors and Jury**  
   This group includes the project supervisor, co-supervisor, and jury members who evaluate the project’s feasibility, effectiveness, and adherence to objectives. They provide ongoing feedback and guidance to ensure the project aligns with academic and technical standards.

**Technical Stakeholders**

**Project Development Team**  
The project development team is responsible for all technical work involved in building the system, including data gathering, machine learning model development, cloud deployment, and web application creation. The team’s roles encompass the following:

* **Data Gathering and Annotation**  
  Collecting and annotating data to train the machine learning model, ensuring accurate detection of contextual anomalies.
* **Model Development**  
  Designing, training, and optimizing deep learning models, particularly for video processing and sequential data handling, using CNNs and RNNs or LSTMs.
* **Cloud Deployment**  
  Implementing a Function-as-a-Service (FaaS) on a cloud platform to handle real-time data processing and scalability for live video feeds.
* **Backend and API Development**  
  Building APIs to enable smooth data communication between components and facilitate real-time monitoring and anomaly detection.
* **UI/UX Development**  
  Creating an intuitive web application and dashboard to allow the school administration and security personnel to interact with the system effectively.
  1. Operating Environment

The system will operate in a cloud-based environment to facilitate real-time monitoring, data processing, and alerting within a school campus. Cloud-based processing will ensure scalability and efficiency for large volumes of data. The following details describe the operating environment:

* Hardware Platform: The system will use network-connected cameras deployed throughout the school campus. These cameras will transmit video feeds directly to the cloud for processing. Additionally, standard computer systems will be available to the school administration and security personnel for accessing the monitoring dashboard.
* Operating System: The cloud environment will run on a compatible OS (e.g., Linux-based or Windows) to support the deployed model and web application.
* Network Environment: A reliable network connection is essential to continuously stream video data from the campus to the cloud. Secure internet access is also required for remote monitoring, model updates, and data synchronization.
* Other Software Components: The environment includes cloud-based deep learning models, a web application with a dashboard interface, and APIs. The front-end interface will utilize NextJS, and cloud services will be managed as Function-as-a-Service (FaaS) to ensure scalability and efficient processing of video data.
  1. System Constraints

The system’s design is influenced by various external constraints that impact its functionality, performance, and usability:

* **Software Constraints**:
  + The project relies on third-party cloud platforms that may impose data processing and bandwidth limitations.
  + API must support the required data transmission and latency levels for real-time monitoring.
  + The system’s compatibility with specific cloud platforms (e.g., AWS, Azure) may restrict choices for infrastructure and software integrations.
* **Hardware Constraints**:
  + The deployment environment may limit the number and types of cameras that can be installed due to budget or infrastructure limitations within the school campus.
* **Cultural Constraints**:
  + The system language will default to English, and consideration for multilingual support may be limited by time and resource constraints.
  + User interfaces must be intuitive and suitable for users with varying levels of technical proficiency among school staff.
* **Legal Constraints**:
  + Data privacy and security laws, such as the General Data Protection Regulation (GDPR) or local privacy regulations, will govern data collection, storage, and transmission, especially regarding video data.
  + Explicit consent and security protocols will be required to handle sensitive visual information.
* **Environmental Constraints**:
  + Data quality may be affected by fluctuating lighting and environmental conditions, impacting the system's accuracy and potentially necessitating additional processing for reliable detection.
* **User Constraints**:
  + The web application will be designed for non-technical users, necessitating a user-friendly and graphical interface for monitoring. Textual explanations will be minimized to simplify interaction.
  + The alert system must provide real-time visual indicators that are easy for school personnel to interpret quickly, requiring the interface to focus on visual feedback over complex textual notifications.
* **Off-the-Shelf Component Constraints**:
  + Some of the cloud-based platforms used may have proprietary constraints, such as limited model customization options or specific integration requirements, which may restrict some functionalities.
  + Existing annotation tools may have limited support for specific data types or formats, affecting the efficiency of the annotation process.
  1. Assumptions & Dependencies

**Assumptions**

1. **Stable Network Connectivity**: It is assumed that the school campus will have reliable and stable network connectivity to support continuous video streaming to the cloud for real-time processing. Temporary outages will be managed by local IoT devices for inference, but these outages should not be prolonged.
2. **Cloud Scalability**: The cloud platform used (AWS, Azure, etc.) will provide the necessary scalability to process video feeds and contextual data simultaneously, supporting the deployment model.
3. **Data Privacy Compliance**: The project assumes that all stakeholders, including school authorities, agree to abide by data privacy and security regulations, ensuring legal use of video and contextual data for monitoring purposes.
4. **High-Quality Data for Training**: It is assumed that the data collected from the campus, such as video feeds, will be of sufficient quality for model training and detection purposes. This includes clear video resolution and appropriate lighting to allow accurate analysis.
5. **Access to Required Hardware**: The necessary hardware (cameras, network routers) will be available and installed in all critical areas across the school campus without any physical restrictions or limitations.
6. **User Readiness for Training**: It is assumed that users (school staff and security personnel) will be provided with initial training on using the system and will possess a basic understanding of interacting with the monitoring dashboard.
7. **Budget Allocation**: The project assumes that sufficient budget is allocated not only for initial development and deployment but also for potential cloud hosting and maintenance costs.

**Dependencies**

1. **Cloud Provider Services**: The project depends on third-party cloud providers to host the system, perform model processing, and provide real-time analytics. The system’s functionality relies on the availability, uptime, and performance guarantees of the cloud platform.
2. **Video Data Storage Compliance**: The storage and handling of video data depend on adherence to privacy laws and institutional policies, which may require secure data encryption, restricted access, and periodic data purging.
3. **Access to Annotated Data**: The model's accuracy depends on a consistent supply of annotated data. Efficient and accurate data annotation tools or services are necessary to support data preparation for training the deep learning models.
4. **User Interface Compatibility**: The system relies on compatibility with the hardware (computers and mobile devices) used by the school administration and security staff to access the web application. Browser compatibility and device responsiveness are crucial for accessibility.
5. **Environmental Conditions**: System accuracy depends on external environmental factors, such as lighting and sound conditions, which may impact video quality and detection reliability.
6. **Legal and Regulatory Approval**: The system’s deployment depends on obtaining any required permissions and clearances from local authorities and school administration regarding the use of monitoring systems in educational environments.
7. **Security Measures**: To ensure data security, the system depends on strong cybersecurity measures on both the cloud platform and the local network to prevent unauthorized access or breaches, especially given the sensitive nature of school monitoring.

1. External Interface Requirements

**3.1. Hardware Interfaces**

* **Camera Feed Interface**:
  + **Device Type**: IP cameras installed at strategic points within the campus.
  + **Data Type**: High-definition video streams.
  + **Control Interaction**: The system will receive live video feeds from cameras, which will be processed both locally on IoT devices and on the cloud. Camera control, such as angle adjustment and resolution settings, may be managed remotely via the interface.
* **Admin Console Interface**:
  + **Device Type**: Desktop computers or tablets with access to the web-based dashboard.
  + **Data Type**: Live monitoring data, alert notifications, and system reports.
  + **Control Interaction**: Users will interact with the system through a web interface, enabling actions like acknowledging alerts, reviewing past events, and adjusting monitoring parameters.
* **Mobile Device Interface**:
  + **Supported Device Types**: Android smartphones and tablets.
  + **Data Type**: Notifications, alerts, and live monitoring data.
  + **Control Interaction**: Allows users to receive notifications, access event history, and view live feeds.

**3.2. Software Interfaces**

* **Databases**:
* **MySQL**:
  + **Purpose**: Storage of transactional data requiring ACID properties, such as user actions, events, and system logs.
  + **Data Items Exchanged**: Anomaly events, user information, and historical data logs.
  + **Communication**: SQL queries for CRUD operations, ensuring data integrity in critical operations.
* **MongoDB**:
  + **Purpose**: Storage of large volumes of non-relational data for analytics, such as aggregated event data and usage statistics.
  + **Data Items Exchanged**: Analytics data, patterns, and aggregated historical data.
  + **Communication**: JSON documents exchanged with MongoDB queries, supporting flexible storage and analysis of unstructured data.
* **Cloud Platform (AWS/Azure)**:
* **Purpose**: Scalable processing of video feeds using machine learning models.
* **Data Items Exchanged**: Video frames and analysis results.
* **Communication**: Data sent to cloud functions, which process the frames and return detection results to the application.
* **Services Needed**: Object storage, and security services (IAM).
* **Operating System**:
* **Environment**: Linux (Ubuntu 20.04) for servers and IoT devices.
* **Data Interactions**: System processes will manage network connectivity, file system access, and process controls.
* **Machine Learning Framework (TensorFlow/PyTorch)**:
* **Purpose**: Frameworks for running anomaly detection models.
* **Data Interactions**: Pre-processed video frames for inference, returning detection results.
* **Communication**: Integrated within cloud functions for real-time processing.
* **Mobile Application**
* **Purpose**: Provide remote access for users to monitor events, receive notifications, and manage system alerts on mobile devices.
* **API Integration**: Connects to system APIs for user authentication, data retrieval, and event updates.
* **Data Items Exchanged**: User credentials, event logs, live feed links, and push notifications.
* **Communication**: RESTful API for real-time data and notifications

**3.3. Communications Interfaces**

* **Network Communication Protocol**:
* **Protocol**: HTTP/HTTPS for web application, RTSP for video streaming from IP cameras.
* **Data Transfer**: Real-time video and alerts.
* **Security**: SSL/TLS encryption for data transmission.
* **Web Application Access**:
* **Requirement**: Admins access the interface via web browsers (Chrome, Firefox).
* **Format**: RESTful API and WebSocket for real-time updates.
* **Security**: HTTPS encryption with user authentication.
* **Mobile Communication**
* **Protocol**: HTTPS for secure API access from the mobile app.
* **Data Transfer**: Alerts, live feeds, and user authentication data.
* **Security**: SSL/TLS encryption, OAuth for secure login, and secure storage of credentials.
* **Email Alerts**:
* **Requirement**: Email notifications for critical anomalies.
* **Format**: HTML emails with event summaries and timestamps.
* **Security**: Secure SMTP with necessary authentication.

1. Functional Requirements
   1. Functional Hierarchy

[This section will give a big picture of overall system functionality. The main modules/features of system and their sub-functions will be described here in the form of a functional hierarchy so that, before getting into the use case, audience could grab the idea of overall system functions.]

* 1. Use Cases
     1. [Title of use case]

[Use Case Diagram]

[Use Case Description]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<Use case Id: name>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  <List of actors (external agents), indicating who initiated the use case> | | | | |
| **Feature:** <Feature from which the use case is driven> | | | | |
| **Pre-condition:** | | <List the assumptions required before this Use Case can be executed. > | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Numbered actions of the actors | | | Numbered description of system responses |
| **2.** |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a:**    **2a:** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Sequentially list conditions expected at the completion of the use case. | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |

1. Non-functional Requirements
   1. Performance Requirements

The system must meet the following performance characteristics to ensure effective monitoring and alert within a school campus environment:

* **Speed**: Real-time video feeds should be processed and analyzed within a few seconds to detect and report any anomalies without delay.
* **Precision**: The anomaly detection model must maintain a high accuracy rate to minimize false positives and negatives.
* **Concurrency**: The system should support multiple concurrent video streams simultaneously without performance degradation.
* **Capacity**: The cloud infrastructure must be scalable to handle increasing data volumes as new camera feeds are added to the system.
* **Reliability**: The system should operate with 99.9% uptime to ensure continuous monitoring, with immediate fallback options in case of failures.
  1. Safety Requirements

Safety precautions for the system include:

* **Data Protection**: All stored video data must be encrypted to prevent unauthorized access, ensuring that sensitive information about students and staff remain protected.
* **Compliance**: The system must comply with relevant safety regulations for video surveillance, including applicable educational privacy standards to avoid harm or misuse of the system.
* **Fail-Safe Mechanisms**: In the event of a system failure, automated alerts should notify the admin and security personnel to ensure any safety concerns are addressed promptly.
  1. Security Requirements

The security requirements for protecting the system and its data are as follows:

* **User Authentication**: Only authorized personnel with appropriate credentials can access the system’s monitoring and dashboard features.
* **Data Integrity**: Measures should be in place to prevent tampering with data, ensuring that all logged events and video streams are accurately recorded and verifiable.
* **Compliance with Policies**: The system must meet educational institution policies on data protection and surveillance, as well as any applicable certifications or regulations concerning student privacy.
  1. User Documentation

The following user documentation components will be provided to facilitate smooth operation and understanding of the system:

* **User Manual**: A comprehensive guide explaining system features, setup, usage, and troubleshooting for administrative and security staff.
* **Tutorials**: Step-by-step tutorials covering system setup, configuring alerts, monitoring video feeds, and responding to detected anomalies.
* **Technical Documentation**: For IT and development staff, detailing system architecture, deployment, and maintenance instructions.

1. References

[This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. (This section is like the bibliography in a published book).]

1. Appendices

[This section should include supporting detail that would be too distracting to include in the main body of the document.]