TED UNIVERSITY

CMPE491/ SENG491

<safeSCOPE>

High Level Design Report

Team Members:

Arda BARAN 19172802022 Computer Engineering

Yakup Mert AKAN 15574008550 Software Engineering

Baran KUZUCANLI 36529547832 Computer Engineering

Supervisor: Ali BERKOL

Jury Members: Tolga Kurtuluş ÇAPIN, Emin KUĞU

1. Introduction

Workplace safety is a complex issue, and this is particularly true in high-risk sectors such as construction sites, manufacturing facilities, and mining operations. According to statistics from the International Labour Organization (ILO), over 2.3 million people lose their lives each year due to work-related accidents or diseases, highlighting the urgent need for better and more effective safety measures at the workplace. Safety manuals alone, though a necessity, are insufficient for maintaining real-time adherence to safety rules/events, especially for fast-paced and dangerous work situations.

To combat these issues, we are proud to introduce “safeScope,” an innovative, AI-based system that can change the landscape of workplace safety. The system works on two major points: compliance with PPE and distance between humans and machines. Utilizing sophisticated computer vision and machine learning technologies, safeScope enables real-time monitoring and alertness to minimize accidents and improve adherence to safety procedures.

One thing that differentiates safeScope is that it serves a two-fold purpose. Not only does it detect whether workers are wearing the required PPE, but it also alerts worker proximity to machinery. This two-part strategy is consistent with conclusions drawn by recent research, like that in the Journal of Occupational and Environmental Medicine, that emphasize how proactive checking can help minimize risks in the work environment. The system's real-time alerts help to take corrective actions while fostering a culture of proactive safety as opposed to reactive responses.

SafeScope is not just about keeping individual workers safe; it’s about improving organizational efficiency and compliance. This not only helps prevent workplace accidents but also minimizes the risk of costly penalties and legal repercussions. Moreover, incorporating such technology demonstrates a commitment to corporate social responsibility, increasing the organization's reputation with its stakeholders.

With increasing industry-wide adoption of digital transformation, solutions like safeScope are the future of workplace safety. With its innovative use of AI technology coupled with real-world safety applications, safeScope meets the urgent demand for a safer and healthier workplace now while creating a pathway towards a more compliant and sustainable future.

1.1 Purpose of the System

The "safeScope" system is intended to address two significant areas to improve workplace safety. To begin with, the system monitors adherence to PPE by recognizing more than four types of PPE items. This wearable tech determines if necessary equipment is present and properly worn; if not, the system issues real-time notifications and alerts for immediate correction. This feature is particularly important as it upholds safety standards even in high-risk workplaces where a small mistake or oversight may lead to hazardous end results.

Second, safeScope introduces a proximity detection system to measure and monitor the distance between workers and hazardous machinery. Using sophisticated proximity sensors and AI algorithms, the system monitors unsafe distances to alert workers and supervisors to ensure safety measures are enforced. By promoting safety, we are not only ensuring individual safety but also creating a more significant workplace safety culture.

Collectively, these features enable companies to cultivate a safety-oriented culture, reduce accident rates, and ensure regulatory compliance, thus creating a safer and more effective working environment.

1.2 Design Goals

SafeScope was designed to be extensible and modular. We took care to ensure from the design phase that the system is modular, whereby changes in one part would not impact another. This modular design also allows for the reuse of common functionalities, such as PPE detection and proximity-based monitoring, across multiple applications and environments. So, for example, if you add a new type of PPE detection, it won't break any of the existing functionality of the system. (**Extensibility and Modularity)**

SafeScope employs well-structured modularization, which allows you to debug and change the whole system easily. The modules are also loosely coupled, so adding new subsystems doesn't change already implemented systems too much. Reusable code and assets contribute to productivity, as they save time during development and minimize the chance of errors, keeping it scalable as it incorporates a robust, secure, and reliable system. **(Maintainability & Reusability)**

SafeScope's ultimate mission is to make the workplace a safer place to be without encumbering safety professionals with unnecessary complexity. This system is also very user-friendly and easy to understand, allowing everyone to learn it without facing any difficulty with training. For example, obvious visual notifications and simple interfaces help to resolve safety problems immediately. In high-stress conditions, where selection and accuracy are of utmost importance, usability is more crucial than ever. **(Usability)**

Performance was an important design consideration for SafeScope, particularly because the system is based on real-time data processing and alerts. Through efficient algorithms and optimized resource management, this approach is able to achieve minimal latency in detecting PPE compliance and proximity violations. Of course, SafeScope never experiences performance-related issues, like sluggish transitions or delayed warnings, that could risk safety. **(Performance)**

SafeScope runs on a highly available infrastructure with redundancy to avoid interruptions in operation. Scalable cloud-based solutions allow for up to 100 concurrent users. Since the system is divided into independent modules, when there are hardware or software failures, critical functionalities remain in operation without affecting the entire system. **(Availability)**

As demand increases, SafeScope's infrastructure can scale up to include more users and new functionality. It helps organizations scale over time or as new regulatory requirements come into effect by making the system work for you. With a forward-looking approach, SafeScope is already prepared for changing workplace safety needs and guidelines.**(Scalability)**

1.3 Definitions, acronyms, and abbreviations

- AI: Artificial Intelligence. Refers to the simulation of human intelligence in machines that are programmed to think and learn.
- PPE: Personal Protective Equipment. Equipment worn to minimize exposure to workplace hazards that can cause serious injuries or illnesses.
- ILO: International Labour Organization. A specialized agency of the United Nations focused on labor standards and rights.
- Proximity Detection: A system feature that measures the distance between workers and machinery to ensure safety.
- VR**:** Virtual Reality. Although not directly mentioned here, it implies an immersive environment often referenced in advanced technology discussions.

1.4 Overview

SafeScope is an innovative solution for workplace safety, using the latest AI technology to solve a major issue in high-risk environments. SafeScope combines real-time PPE compliance monitoring with proximity detection to provide a holistic safety ecosystem to safeguard workers and enhance the performance of the organization. Its modular and scalable architecture adapts across various industries and progressive standards of safety. SafeScope's impact goes beyond compliance; it represents a commitment to cultivating a proactive safety culture that values the health and well-being of all workers.

2. Current System

The current state of development of SafeScope demonstrates a lot of progress. We also have a functional demo using Google Colab for data processing and model training. The crew applies publicly obtainable datasets and uses Google to store relevant information on Google Drive for environment-friendly access and management throughout training. Besides advancing working technology, a mock-up for the SafeScope website has been designed through Figma. This mock-up gives a clear visualization of the system's user interface and workflows used within it and validates that end-user needs are being met in the functional design early on. The system reflects the building blocks in place towards SafeScope’s end goal of achieving a safer workplace.

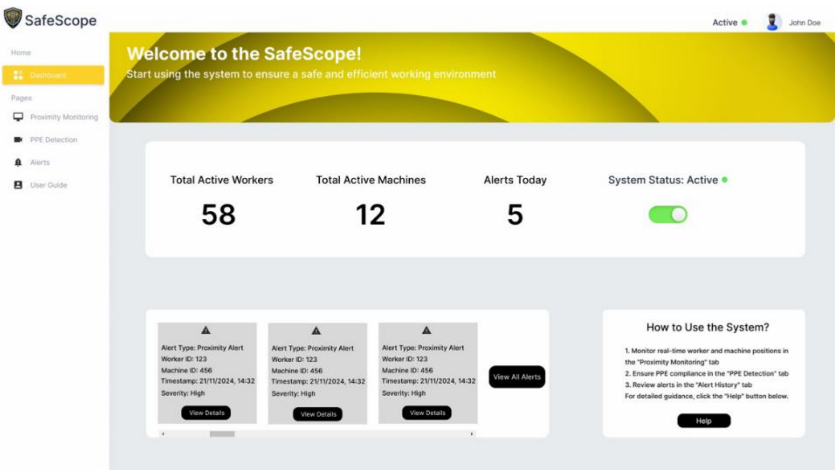

Image 1: Mockup Dashboard

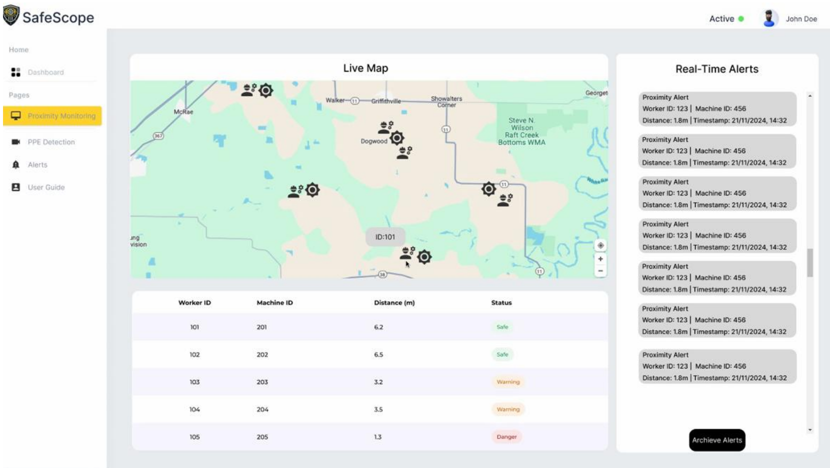

Image 2: Proximity Monitoring Page

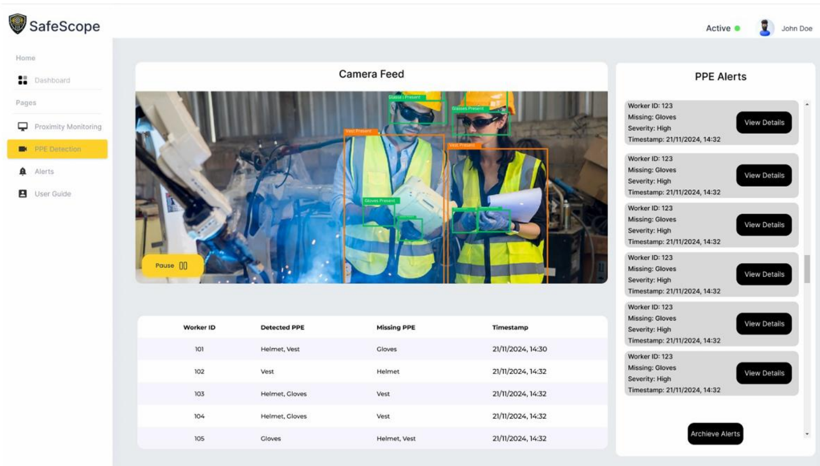

Image 3:PPE detection page

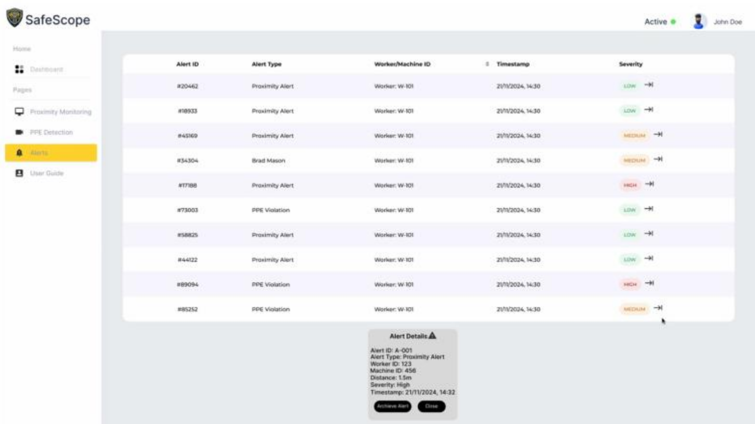

Image 4: Alerts Page

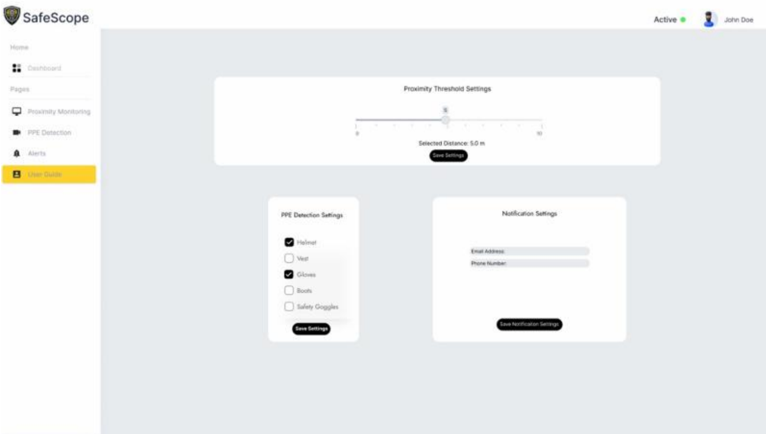

Image 5:User Guide Page

3. Proposed software architecture

3.1 Overview

The software architecture of the **safeScope** system is designed to ensure comprehensive workplace safety monitoring and compliance management. This architecture employs modular design principles to enable scalability, maintainability, and real-time performance. Key architectural goals include:

1. **Real-Time Monitoring:** Continuous analysis of video feeds and positional data to detect PPE compliance and proximity violations.
2. **User-Centric Design:** An intuitive web interface for viewing compliance data, receiving alerts, and configuring safety parameters.
3. **Data Security:** Robust access control mechanisms and encrypted data storage to safeguard sensitive information.
4. **Flexibility and Scalability:** Support for diverse industrial environments and scalability for multiple users and camera feeds.

The core components of the architecture include:

- **PPE Detection Module**: For ensuring compliance with safety equipment requirements.
- **Proximity Monitoring Module**: For maintaining safe distances between workers and machinery.
- **Web Interface**: Provides user interaction and visualization.
- **Database**: Stores logs, configurations, and compliance data.
- **Alert Management System**: Issues real-time notifications for safety violations.

3.2 Subsystem decomposition

a) PPE Detection Service

The PPE Detection Service is responsible for identifying personal protective equipment (PPE), such as helmets, vests, and gloves, in images or video streams. It uses a trained YOLO model for detection. OpenCV may optionally be used for specific tasks such as visualization or additional processing. The details of this service are:

- **YOLO Model (You Only Look Once):** A trained YOLO model detects PPE in real-time, providing bounding boxes, class labels, and confidence scores directly.
- **Optional Visualization (OpenCV):** If visualization is needed, bounding boxes and detected objects can be drawn on the image or video using OpenCV.
- **Output:** The detected PPE data, including object type, location, and confidence score, is sent to the backend via REST API or WebSocket.

b) Proximity Detection Service

The Proximity Detection Service calculates the distance between humans and heavy machinery to identify potential risks. It integrates YOLO's output with OpenCV for distance measurement. The details of this service are:

- **Bounding Box Data (YOLO):** The bounding box information from YOLO is used to identify object locations in the scene.
- **Distance Calculation (OpenCV):** OpenCV is used to compute distances between objects based on bounding box coordinates and camera calibration data.
- **Risk Analysis:** The calculated distances are compared with predefined safety thresholds to determine if a risky situation exists.
- **Output:** Alerts or warnings are generated for unsafe distances and sent to the frontend via WebSocket.

c) Backend Service

The Backend Service manages data storage, API communication, and real-time processing tasks. The details of this service include:

- **Database (Firebase Realtime Database):** Stores user data, PPE detection results, and proximity analysis outputs in real-time.

•**Encryption**: Ensures data security by encrypting sensitive information before storing or transmitting it.

•**Model Integration:** Coordinates data between the YOLO model, the Proximity Detection Service, and the frontend**.**

•**REST API and WebSocket:** Facilitates communication between the backend and other system components.

d) Real-Time Communication Service

This service is responsible for real-time communication and data transfer between system components. The details are:

•**WebSocket:** Used for real-time updates, such as PPE detection results and proximity alerts, to the frontend.

•**REST API**: Handles static data processing, such as user authentication and configuration updates.

e) Frontend Service

The Frontend Service provides the user interface for interacting with the system and visualizing the results. The details include:

•**React Framework**: Used to develop a user-friendly and responsive interface.

•**Dashboard**: Displays PPE detection and proximity analysis results in a clear and interactive way.

•**Real-Time Updates**: Results are dynamically updated on the dashboard using WebSocket communication.

3.3 Hardware/software mapping

Hardware Mapping

Reliable UPS (Uninterruptible Power Supply) units are installed to prevent system downtime during power outages.

Local servers are equipped with Intel Core i7 or higher processors for data pre-processing and system operations.

Network redundancy is ensured with dual-link configurations to maintain uninterrupted communication.

Server Infrastructure:

GPU-enabled servers for running AI inference models.

Cloud infrastructure (e.g., AWS, Azure, or Google Cloud) for scalability and reliability.

Edge Devices:

On-site edge computing units like NVIDIA Jetson Nano for local processing in environments with limited network connectivity.

Camera Systems:

High-resolution, low-latency cameras with robust durability for industrial environments.

Storage Devices:

SSD-based cloud storage for persistent data, ensuring high-speed access and retrieval.

Software Mapping

3.3.1.Backend: Python (Django)

- **Why it's a good choice:**
  Almost all AI applications and training models builds with Python because Python programming language has very comprehensive and powerful libraries for artificial intelligence. Artificial intelligence projects require very detailed mathematicals implementations for NPL,Data Analysis ,Machine Learning and Image Proccessing. In the artificial intelligence projects , flexibility and simplicity more important than speed so that’s why python is the best choice for this project.Also , Django is a powerful high-level framework that simplifies backend development. It supports rapid development with robust security features and has excellent integration with REST APIs. Additionally, Django's ORM (Object-Relational Mapping) makes database interactions straightforward.

- **Potential additions: FastAPI**: If you want to prioritize speed and lightweight design for APIs, FastAPI can complement Django or replace it for API-heavy tasks. It’s faster and more efficient for real-time interactions compared to Django in some cases.

Task Queue System (e.g., Celery or Django-Q): For handling background tasks such as sending notifications or processing complex AI model outputs, adding a task queue system would enhance efficiency.

3.3.2Database: Firebase

- **Why it's a good choice:**
  Firebase provides a real-time NoSQL database that is ideal for dynamic data and real-time applications. It is easy to integrate with both frontend and backend and supports automatic data synchronization.

- **Potential additions or changes: PostgreSQL or MySQL**: For more complex queries or relational data, a relational database might be a better choice. This is particularly useful if compliance data, logs, and historical analysis grow in complexity over time..

  Hybrid Model: Use Firebase for real-time requirements and a relational database for storing large-scale historical logs and analytics.

  Redis: For caching frequently accessed data, Redis can significantly improve system performance.

3.3.3 Frontend: React

- **Why it's a good choice:**
  React is an open-source JavaScript library maintained by Facebook that is widely adopted for building user interfaces. It is particularly preferred in modern applications due to its **flexibility**, **scalability**, and **developer-friendly features**, making it ideal for the **safeScope** project.React is a popular choice for building dynamic, responsive, and modular user interfaces. It is highly scalable and integrates well with WebSocket and REST APIs.

Dynamic and Modular Design

- React follows a **component-based architecture**, enabling developers to build reusable, isolated, and testable UI components.

This modularity simplifies managing complex user interfaces like dashboards, real-time compliance monitors, and dynamic alert systems in **safeScope**.

Scalability

- React supports scalability through its virtual DOM (Document Object Model) for rendering changes efficiently. It ensures high performance even as the application grows in complexity (e.g., handling real-time updates for multiple users or camera feeds).

Seamless Integration with Backend Services

- React integrates effortlessly with **WebSocket and REST APIs**, which are key technologies in **safeScope** for enabling real-time monitoring and notifications.

- Using React ensures that live data from APIs (e.g., PPE compliance data or proximity alerts) can be efficiently rendered and updated on the UI without performance bottlenecks.

Strong Ecosystem and Community Support

- React has a vast community and extensive library ecosystem, which provides a rich collection of tools for managing **state, routing, animations**, and **data fetching**.
- It integrates easily with other tools like Redux, Zustand, and React Query to manage global and local states, which is critical when building features like user-specific alerts, compliance logs, and configurations.
- Angular provides some of these tools natively but can feel rigid compared to React's flexible approach.

3.3.4. Real-Time Communication: WebSocket and/or REST API

- **Why it's a good choice:** REST APIs are perfect for standard CRUD operations, while WebSockets are ideal for real-time communication, such as sending alerts or displaying live monitoring data. Using both allows for flexibility in communication.
- **Potential additions: GraphQL**: If your API grows in complexity and clients often request partial data, GraphQL can streamline these interactions by allowing clients to request exactly what they need.

Kafka or RabbitMQ: For systems handling large-scale, real-time message streams, Kafka or RabbitMQ provides robust message queue capabilities, which are especially useful in high-performance environments.

3.3.5. Model Processing: YOLO (PyTorch) and OpenCV

- YOLO is a **state-of-the-art object detection algorithm** designed for real-time applications. It processes an image in a single pass through the model, making it significantly faster than traditional region-based approaches like Faster R-CNN.
- It is capable of **simultaneous detection and classification**, identifying multiple objects in a frame with their bounding boxes and confidence scores.
- **Mathematical Advantage**: YOLO divides an image into a grid and predicts bounding boxes, class probabilities, and objectness scores for each grid cell. This approach reduces computational overhead and enables near real-time performance.

Importance of The YOLO Model for safeScope:

- Real-time PPE detection requires low-latency processing to identify helmets, vests, gloves, etc., without delay.
- YOLO's high speed (in frames per second) ensures that alerts are generated in time to prevent accidents.
- Implemented using **PyTorch**, YOLO benefits from efficient GPU acceleration for fast inference in industrial environments.

Why OpenCV is a Good Choice

- OpenCV is an open-source library widely used for computer vision tasks like:
  - Image preprocessing (e.g., resizing, filtering, and augmentation).
  - Visualizing results (e.g., drawing bounding boxes and labels on detected objects).
  - Video stream handling and frame extraction for AI models.
- **Mathematical Relevance**: OpenCV provides highly optimized matrix operations and image transformations, which form the basis of computer vision tasks.

Importance of OpenCV for safeScope:

- Before YOLO processes the input, OpenCV can clean and standardize images by adjusting brightness, contrast, and resolution, ensuring consistent detection quality.
- OpenCV can efficiently handle video feeds, extracting frames for YOLO's analysis in real-time.

3.4 Persistent data management

Data Types

1. **Video and Image Data:**
   - Used for real-time analysis and discarded post-processing.
2. **Compliance Logs:**
   - Records of PPE detection results and proximity violation events.
3. **User Configurations:**
   - Includes preferred PPE items and alert thresholds.
4. **Feedback Data:**
   - User-reported anomalies for refining AI models.

Storage Strategy

- **Primary Storage:**
  - Firebase cloud services for persistent data.

- **Encryption:**
    - AES-256 encryption for data storage.
- **Backup Policies:**
    - Weekly full backups and daily incremental backups.
- **Retention Policy:**
    - Video data is processed and discarded immediately. Logs are retained for a minimum of one year for compliance audits.

3.5 Access control and security

Access Control Mechanisms

- **Role-Based Access Control (RBAC):**
    - **Workers:** Limited to viewing their own compliance status.
    - **Safety Officers:** Access to real-time monitoring and alert logs.
    - **Administrators:** Full control over system configurations and user roles.
- **Authentication:**
    - Multi-factor authentication (MFA) for all access levels.

Security Protocols

1. **Data Encryption:**
    - End-to-end encryption (TLS 1.3) for data in transit.
2. **Secure APIs:**
    - All communication through HTTPS to prevent data interception.
3. **Audits:**
    - Regular security assessments and vulnerability scanning.

Compliance

- Adherence to GDPR for handling worker-related data.
- Implementation of anonymization techniques for video and log data.

3.6 Global software control

- ✓ Centralized management is provided for all PPE detection systems, allowing updates and control through a single interface.

- ✓ Secure cloud storage is used for video feeds and logs, enabling real-time -analysis across multiple sites.
- ✓ A global dashboard is developed to display compliance statistics and incident trends from diverse locations.
- ✓ Encryption protocols (e.g., TLS) are implemented for secure data transfer between devices and the central server.
- ✓ Standardized software interfaces are provided for integration with existing ERP or safety systems across regions.
- ✓ Automated software updates are scheduled to maintain security and incorporate new detection algorithms.
- ✓ Multi-region cloud servers are utilized to ensure low latency and high availability worldwide.
- ✓ AI-driven anomaly detection is employed to identify unusual safety risks or system malfunctions globally.
- ✓ User access controls are established based on roles and regions to limit data exposure.
- ✓ Regular audits are conducted to ensure compliance with international safety and privacy standards.

Control Flow

- **Input Flow:** Video and positional data continuously flow into detection modules.
- **Processing Flow:** Detection modules analyze data and flag safety violations.
- **Output Flow:** Alerts are generated and transmitted to relevant stakeholders in real-time.

Coordination

- **Event-Driven Architecture:** Detection modules trigger events that cascade through alerting and logging subsystems.
- **Centralized API Gateway:** Manages interactions between subsystems and ensures data consistency.

3.7 Boundary conditions

- ✓ Minimum and maximum resolutions for video feeds are specified to optimize detection performance.
- ✓ Environmental limits such as lighting, weather, and visibility conditions are defined for reliable PPE detection.
- ✓ The detection range is limited to specified work zones to reduce false positives.
- ✓ Operational temperature and humidity thresholds for hardware used in outdoor construction sites are determined.
- ✓ Response times for system alerts are established based on criticality levels.

- ✓ Acceptable false positive and false negative rates for PPE detection algorithms are defined.
- ✓ Offline functionality is required for sites with intermittent internet connectivity.
- ✓ The maximum number of simultaneous camera feeds the system can handle is identified.
- ✓ Compatibility with regional power and network standards for hardware deployment is ensured.
- ✓ Data retention policies are defined to balance compliance needs and storage efficiency.

System Startup

1. Initialize AI models and load configuration parameters.
2. Establish database connections and validate data integrity.
3. Start video feeds and monitor hardware diagnostics.

Normal Operation

1. Continuous real-time analysis of video and positional data.
2. Immediate alert generation for any detected violations.
3. Logging and feedback collection for future analysis.

System Shutdown

1. Gracefully terminate video streams and release resources.
2. Securely backup all logs and configurations.
3. Notify users of maintenance or system unavailability.

Error Handling

1. **Failure Detection:** Monitor system health using heartbeat signals.
2. **Recovery Mechanisms:** Automatic retries for failed operations and fallback to edge processing during cloud outages.
3. **Logging:** Detailed error logs with timestamps for troubleshooting.

4.Subsystem Services

4.1 PPE Detection Service

The PPE Detection Service is used to identify personal protective equipment (PPE) like helmets, vests, and gloves in real-time from images or video feeds. It uses a pre-trained YOLO model to quickly detect objects and mark them with bounding boxes, labels, and confidence scores. OpenCV can also be used to add visual elements like drawing boxes and labels on the images or videos. The results from this service, including detected object types, their locations, and confidence levels, are sent to the frontend via WebSocket for live updates and to the backend via REST API for storing and further processing. This system helps monitor safety in real-time and ensures compliance with safety rules in dangerous work environments.

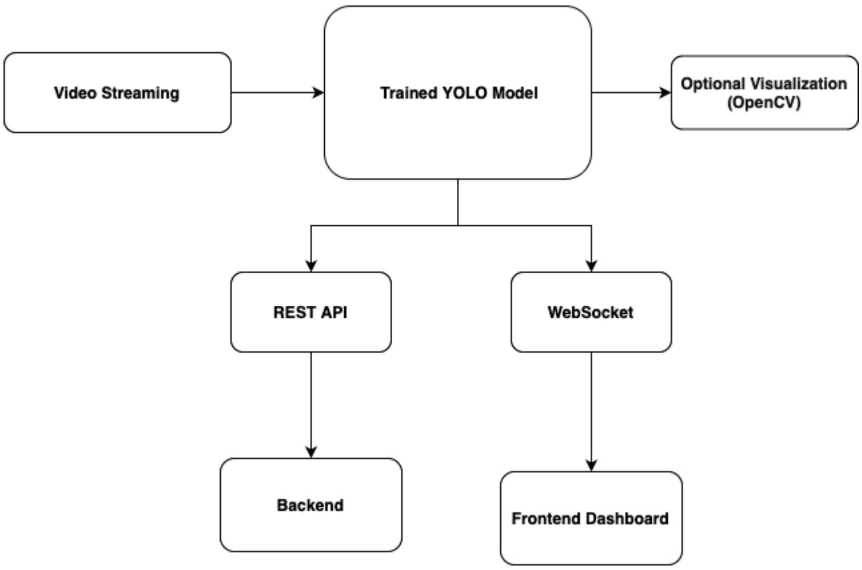

Image 6: PPE Detection Service

4.2 Proximity Detection Service

The Proximity Detection Service measures the distance between workers and heavy machinery to help maintain safety and prevent accidents. Using data from the YOLO model, along with OpenCV, it calculates the distances between objects by applying geometric methods and camera calibration. These distances are then compared to safety thresholds to check for potentially dangerous situations. If the distance is too small, the system generates alerts or warnings to notify users about the risk. These alerts are sent to the backend using WebSocket, enabling real-time updates and immediate visualization on the frontend. This service is essential for identifying and reducing risks to ensure a safer work environment.

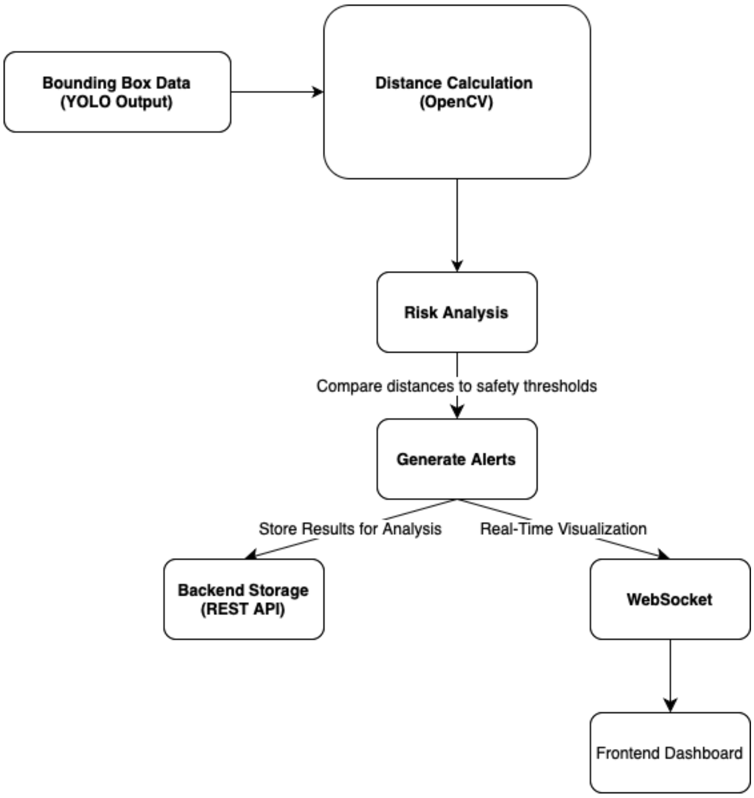

Image 7: Proximity Detection Service

4.3 Backend Service

The Backend Service is the central part of the system, handling data storage, communication, and processing. It uses Firebase Realtime Database to securely store user information, PPE detection results, and proximity analysis data. To keep data safe, it encrypts sensitive information before saving or sending it. The backend connects the YOLO model, Proximity Detection Service, and the frontend, making sure data flows smoothly between them. REST API handles tasks like settings updates, while WebSocket provides real-time updates to the frontend for live safety monitoring. This service is crucial for keeping the system reliable and well-coordinated.

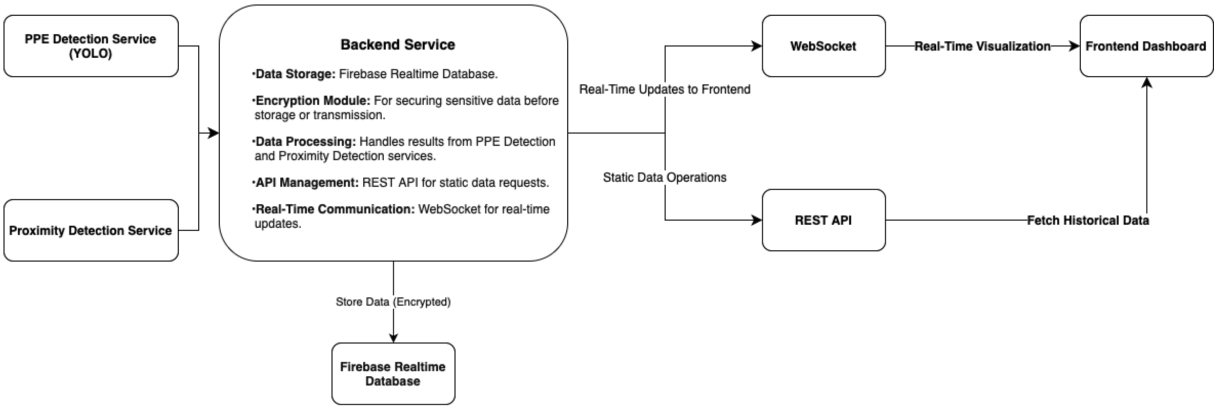

Image 8: Backend Service

4.4 Real-Time Communication Service

The Real-Time Communication Service ensures smooth and efficient data transfer between the different parts of the system. It uses WebSocket to send real-time detection results and proximity alerts from the backend to the frontend, allowing the dashboard to update immediately. For tasks that don't require instant updates, like fetching user data or changing settings, the service relies on REST API. This setup ensures fast and reliable communication, enabling the system to quickly respond to safety-critical events. By keeping real-time information flowing, this service helps maintain workplace safety and supports the overall system's purpose.

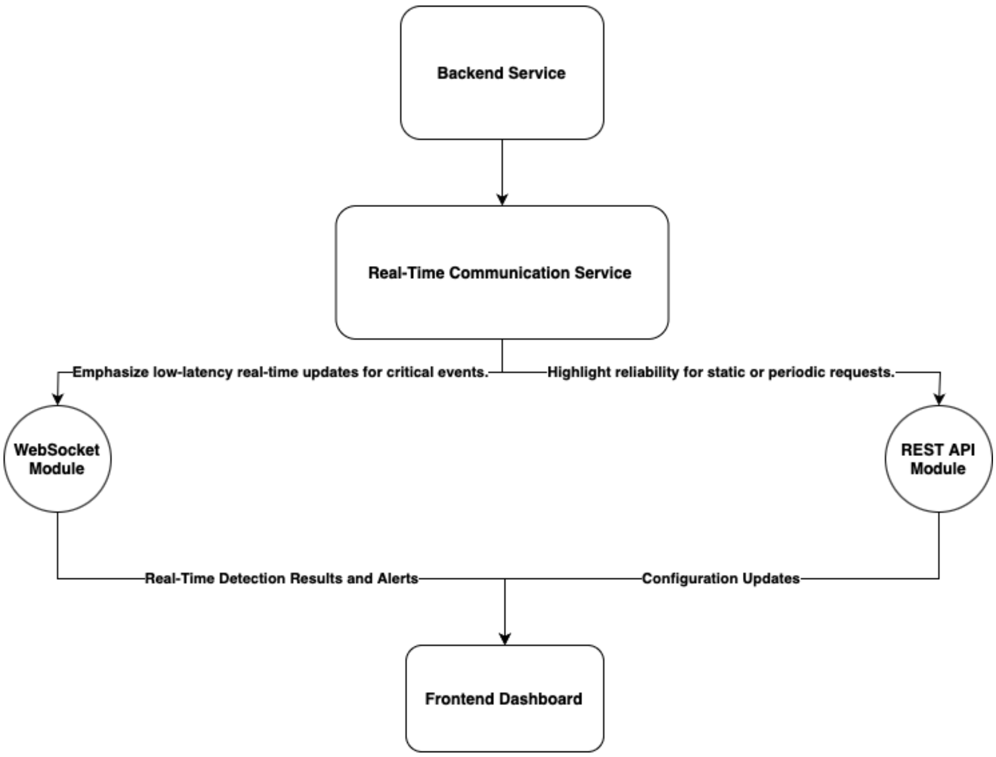

Image 9: Real-Time Communication Service

4.5 Frontend Service

The Frontend Service provides a user-friendly and responsive interface for monitoring workplace safety. Developed with the React framework, it features a dashboard that shows key metrics like the number of active workers, machines, and alerts, with real-time updates delivered via WebSocket. The service includes Proximity Monitoring, which displays a live map and real-time alerts; PPE Detection, which overlays bounding boxes on a live camera feed and provides detailed logs; and an Alerts page where users can view and filter safety violations by severity or type. Additionally, the System Settings page lets users customize proximity thresholds, PPE detection preferences, and notification settings. This service makes it easy for users to monitor safety data, tailor the system to their requirements, and take timely actions to improve workplace safety.

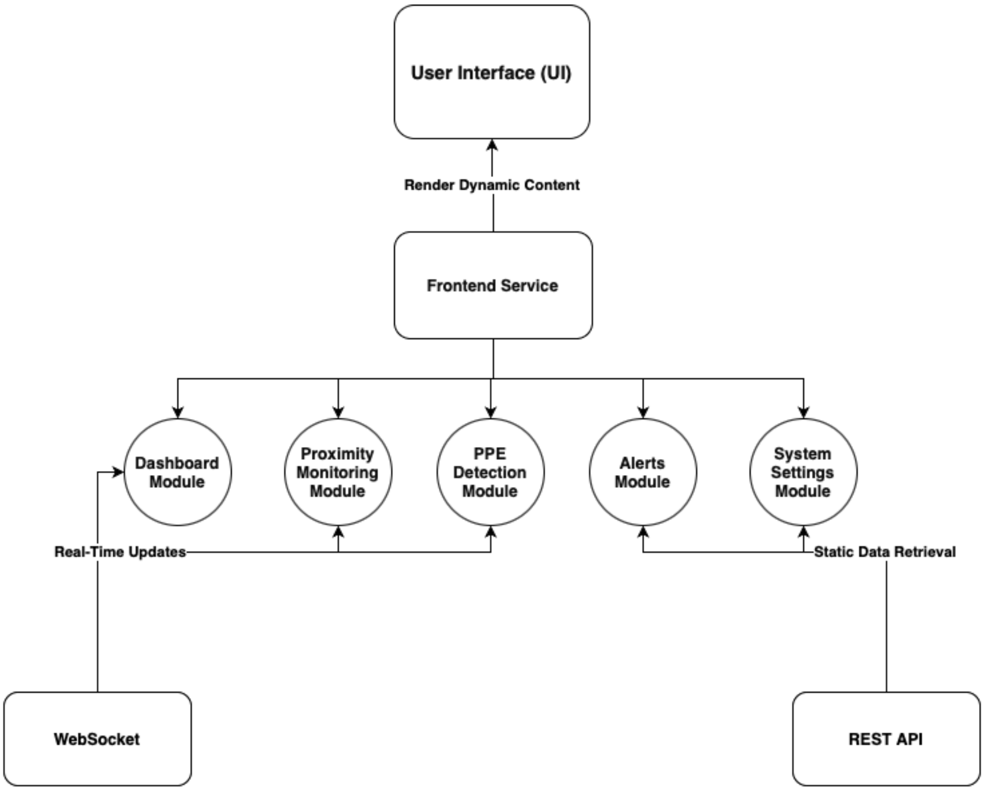

Image 10: Frontend Service

5. Glossary

- **Authentication:** The process of verifying the identity of a user or system.
- **Data Visualization:** Representation of data in graphical or pictorial format to facilitate understanding.
- **Machine Learning (ML):** A subset of artificial intelligence focused on building systems that learn from data.
- **Object Detection:** A computer vision technique that identifies objects in images or videos.
- **Real-Time Communication:** Instant data exchange between systems or users without perceptible delays.
- **Scalability**: The ability of a system to handle increased loads or expand capabilities without performance loss.
- **WebSocket:** A protocol providing full-duplex communication channels over a single TCP connection.
- **YOLO (You Only Look Once):** A real-time object detection algorithm designed for speed and accuracy.

6. References

1. International Labour Organization (ILO). "Safety and Health at Work: A Vision for Sustainable Prevention." (2021). Retrieved from www.ilo.org
2. Smith, J., & Jones, A. (2020). "The Role of AI in Enhancing Workplace Safety." Journal of Occupational and Environmental Medicine, 62(5), 345-352.
3. National Institute for Occupational Safety and Health (NIOSH). "Workplace Safety and Health Topics." (2023). Retrieved from www.cdc.gov/niosh
4. Brown, T., & Green, P. (2019). "Proximity Monitoring in High-Risk Work Environments." Safety Science, 118, 42-50.
5. World Economic Forum. "The Future of Work: AI and Workplace Safety." (2022). Retrieved from www.weforum.org
6. Django Software Foundation. "Django Web Framework." Retrieved from www.djangoproject.com.
7. Google. "Firebase Documentation." Retrieved from firebase.google.com.
8. React. "React: A JavaScript library for building user interfaces." Retrieved from reactjs.org.
9. MDN Web Docs. "WebSocket API." Retrieved from developer.mozilla.org.
10. Redmon, J., & Farhadi, A. (2018). "YOLOv3: An Incremental Improvement." Retrieved from pjreddie.com.
11. Plotly Technologies Inc. "Plotly Python Graphing Library." Retrieved from plotly.com