CSE 564 Project Report Number 5

Team 16

Team Member Names:

1. Aditya Pant
2. Ameya Shahu
3. Lalit Arvid Balaji
4. Pravalika Mukkiri

Table of Contents

Table of Contents

1. Problem

1.1. Problem Overview

- Government Agencies
  - Border Control
    - Concerns related to illegal border crossings and smuggling activities.
    - Struggle to monitor large areas for potential encroachments.
  - Law Enforcement
    - Challenges in monitoring and managing public events and gatherings.
- Private Enterprises
  - Industrial Facilities
    - Safety concerns in high-risk industrial environments
    - Desire for privacy and protection against unauthorized access
  - Sensitive Research Facilities
    - Protection of intellectual property from surveillance
    - Mitigation of potential threats to research and development activities

1.2. Previous Solutions and Why They Are Unsatisfactory

Previous Solutions and their Limitations

- Traditional Surveillance Systems: CCTV cameras
  - Limited Scalability:
    - Inability to cover vast areas or properties.
  - Blind Spots:
    - Challenges in monitoring certain terrains or locations
- Human Patrols
  - Resource Intensiveness:
    - High costs associated with continuous human presence.
  - Human Error:
    - Risk of missing encroachments due to fatigue or oversight
  - Inability to cover extensive areas continuously.

Current Attempts and Their Drawbacks

- Fixed-Position Drone Surveillance
  - Limited Mobility:
    - Drones stationed at fixed locations, limiting coverage.
  - Dependence on Operator Skills:
    - Operator skills impact effective monitoring
  - Limited Integration
    - Challenges in integrating with existing security infrastructure.
  - Regulatory Restrictions
    - Legal and regulatory hurdles in deploying drone systems

2. Requirements

2.1. Autonomous Drones

2.1.1. Formal statement of the requirement

- The system must employ autonomous drones with sensor capabilities for real-time monitoring.

2.1.2. Source

- Handbook of Unmanned Aerial Vehicles. Springer [1].

2.2. Real time Surveillance

2.2.1. Formal statement of the requirement

- Any unlawful movements are reported to the central control system by drones.
- Security personnel need to respond when the control system triggers an alarm.

2.2.2. Source

- Handbook of Unmanned Aerial Vehicles. Springer [1].

2.3. Data Processing in Real-time

2.3.1. Formal statement of the requirement

- Implement AI-powered detection algorithms to analyze data for potential encroachments and minimize false positives.
- A maximum 500 millisecond delay cannot be allowed in the real-time processing of data.
- Data from up to 10 drones must be processed simultaneously by the system.

2.3.2. Source

- Pattern Recognition and Machine Learning. Springer [1].

2.4. Programmable Routes

2.4.1. Formal statement of the requirement

- Drones should support programmable routes for comprehensive coverage of designated areas.
- Scheduled monitoring: At regular intervals, the drones are intended to fly over the entire industrial zone.
- Demand monitoring: To provide an additional layer of security, one can instruct the drones to watch a certain section of the industrial zone.

2.4.2. Source

- Onboard IMU and Monocular Vision-based Control for Aggressive Quadrotor Flight. In Robotics: Science and Systems (RSS)[1].

2.5. Real-time Data Sharing

2.5.1. Formal statement of the requirement

- Enable real-time data sharing with law enforcement agencies and relevant authorities.
- Drones tracking the encroached creature should communicate real-time footage of its movements.

2.5.2. Source

- Dictionary of Computer Science, Engineering, and Technology [1].

2.6. Remote Operation Capabilities

2.6.1. Formal statement of the requirement

- Ensure the capability for remote operation to respond to encroachments in challenging or hazardous environments.

2.6.2. Source

- Industrial Communication Technology Handbook [1].

2.7. Load Balancing

2.7.1. Formal statement of the requirement

- Processing information from every drone at once shouldn't take longer than 100 ms.
- The system should split up the data processing tasks when multiple drones (up to 10) are working at once to maintain performance.

2.7.2. Source:

- Towards Traffic Load Balancing in Drone-assisted Communications for IoT[1]

2.8. Interface

2.8.1. Formal statement of the requirement

- Users' devices need to have user-friendly interfaces that make it possible for them to control and oversee the activities of drones.
- The drone's location in the event of an invasion should be visible to users on the interface, along with live footage captured by the drone's cameras.

2.9. User Access Control

2.9.1. Formal statement of the requirement

- To allow different users to have different levels of access and control over the system's capabilities.
- role-based access control ought to be included in the user-friendly control interface.

2.9.2. Source

- Role-based_access_control2

2.10. Data Encryption and Privacy Measures

2.10.1. Formal statement of the requirement

- Implement robust data encryption and privacy measures to protect sensitive information.
- Security procedures such as authentication and encryption should be used when exporting surveillance data to cloud computing platforms or external storage to avoid unwanted access.

2.10.2. Source

- Algorithms and Architectures for Parallel Processing [3].

[1] Fan, Qiang, and Nirwan Ansari. "Towards traffic load balancing in drone-assisted communications for IoT." IEEE Internet of Things Journal 6.2 (2018): 3633-3640.

[2] https://en.wikipedia.org/wiki/Role-based_access_control

3. Architecture

3.1. Overview of the Architectural Views

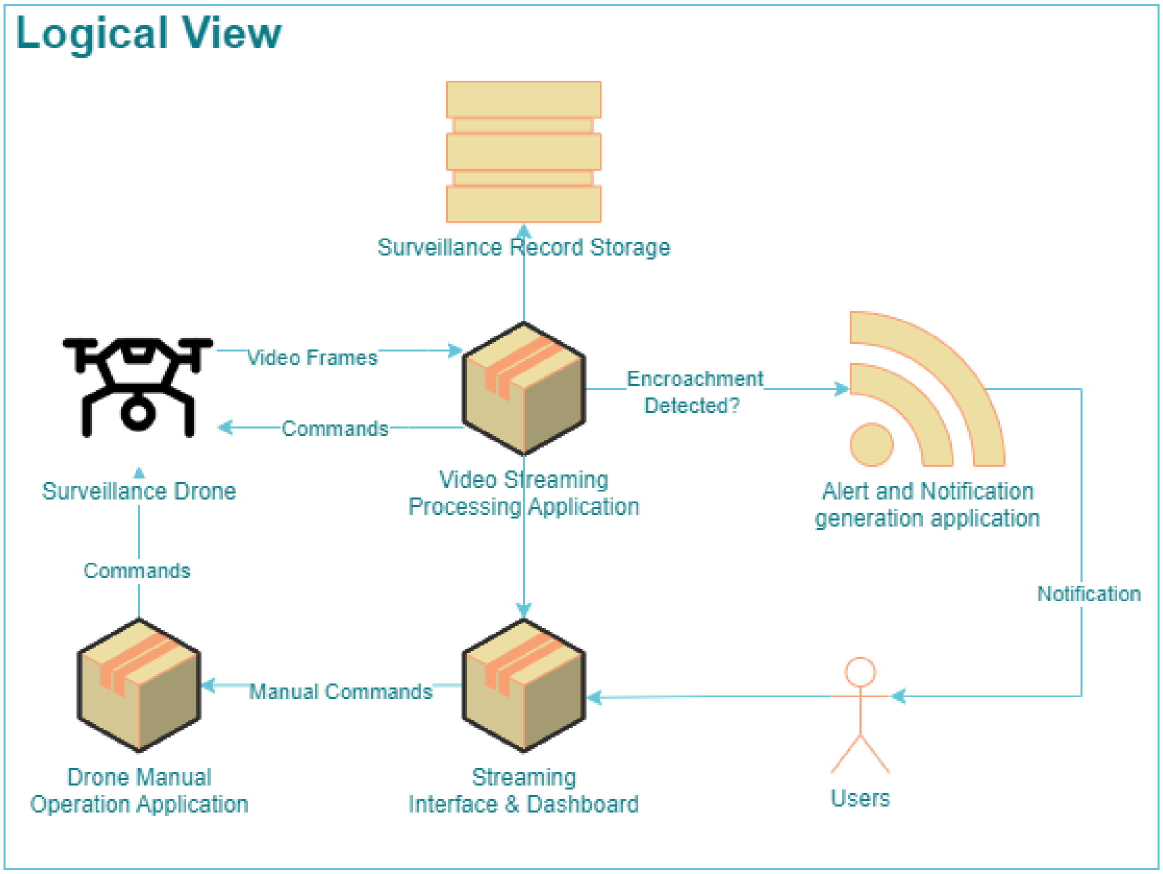
- Within the architectural framework of our drone-based encroachment detection system, each view assumes a critical role in establishing a resilient and adaptable solution.
- The Logical View elucidates the core components responsible for user interfaces, camera video processing, intrusion alert generation, and data storage.
- The Process View delineates the dynamic interactions between system components, encompassing drone operations, flight planning, and camera video processing.
- The Physical View shifts the focus to the system's hardware, including drones, databases, control, ensuring that the physical architecture aligns with environmental requirements.
- The Deployment View spotlights the technological underpinnings, such as the software programming languages employed detection in camera video, and database management.
- The system's versatility comes to the fore in Scenarios View, where it adeptly manages diverse scenarios.
- Also including user interactions with operators, efficient flight scheduling between charging intervals, and rapid alert systems for encroachment detection.
- These architectural perspectives collaborate harmoniously, crafting a well-rounded system designed to cater.
- As well as to a broad spectrum of applications while upholding efficiency, security, and responsiveness.

3.2. Logical View

3.2.1. Introduction and Rationale

- For incursion detection, our surveillance system combines drones, cameras, and deep learning.
- The logical approach emphasizes the software components, user interfaces, and alerting systems that structure the operation of the system.
- It also allows for live streaming and archiving of surveillance footage, allowing for real-time monitoring and historical study of encroachment.

3.2.2. Architectural Model


Logical View
Surveillance Record Storage
Video Frames
Commands
Encroachment Detected?
Surveillance Drone
Video Streaming Processing Application
Alert and Notification generation application
Commands
Notification
Manual Commands
Drone Manual Operation Application
Streaming Interface & Dashboard
Users

3.2.3. Streaming Interface and Dashboard

- The interface offers live video streams from drones' cameras for real-time area monitoring, accompanied by user-customizable camera views, geographic positioning, and navigation.
- Additionally, it supports video review and in-depth analysis for post-event investigative work.

3.2.4. Drone Operation Application

- The system enables prompt responses to intrusion alerts through user-directed drone actions and incorporates safety measures like geofencing.
- It offers robust access control, seamless machine learning integration for intrusion detection, and a live telemetry dashboard for real-time drone performance data.

3.2.5. Surveillance Drone

- The system leverages GPS tracking for precise drone navigation and location data.
- It ensures real-time data transmission, automatic alert responses, and remote manual drone control with live camera feed for dynamic surveillance adjustments.

3.2.6. Alert Application

- The application offers an alert dashboard providing real-time grouped notifications for quick situational analysis, with alert information accessible, including intrusion type, timestamp, coordinates, and media.
- It supports various notification channels, tracks alert history for trend analysis, and integrates a map overlay for enhanced situational awareness and precise event location display.

3.2.7. Surveillance Records

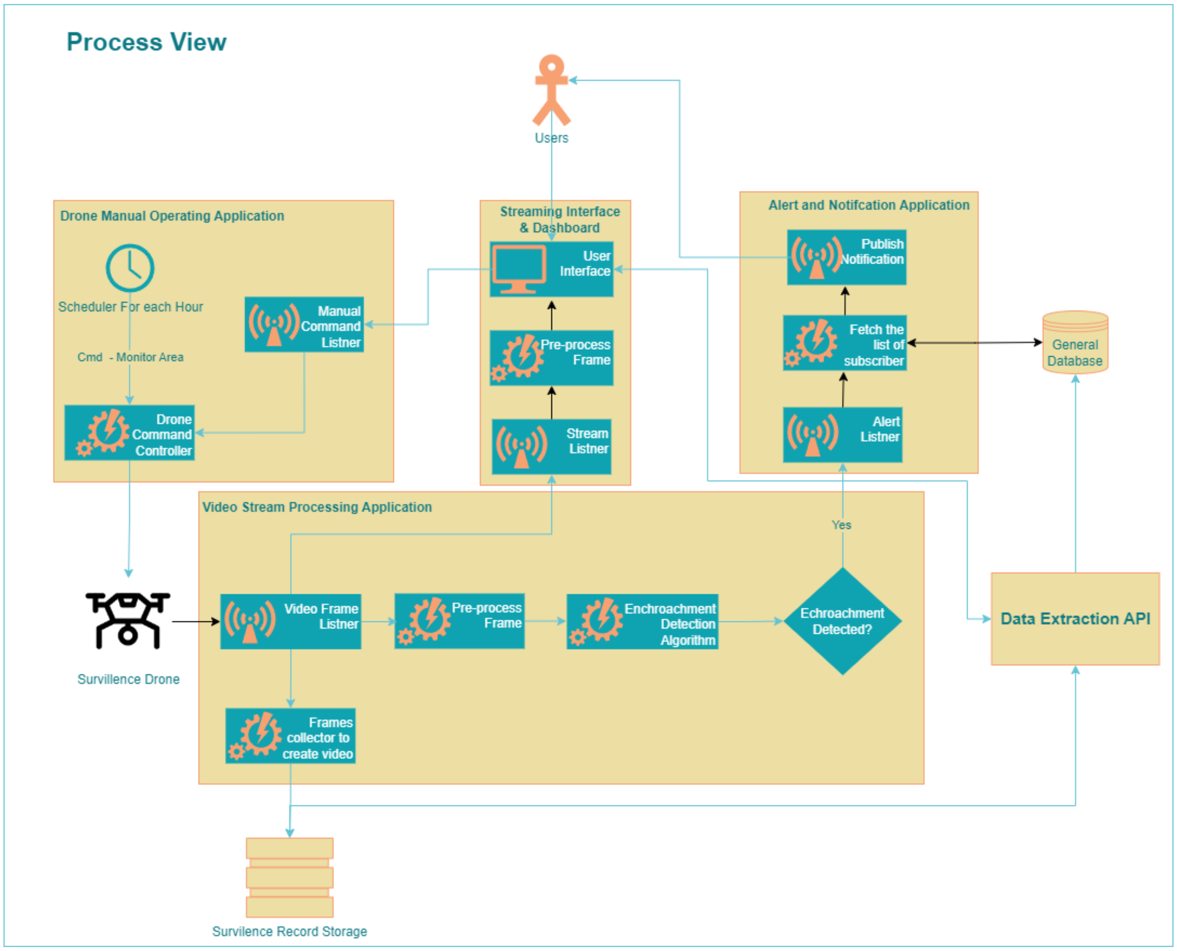
- The database manages user profiles, ensuring authorized access with contact details and access permissions.
- It maintains a substantial archive of timestamped, geotagged camera footage, historical encroachment records, and enhancing system reliability and historical analysis capabilities.

3.3. Process View

3.3.1. Introduction and Rationale

- The system's three primary components are the Drone Operating Application for real-time control, scheduled patrols, and intrusion warnings.
- The Live Streaming Interface manages camera access, ensures data security, and collects real-time drone data.
- The Alert Application records intrusion details for later analysis, sends out alerts promptly, analyzes camera data, and detects intrusions.

3.3.2. Architectural Model



3.3.3. Drone Data Monitoring

- To keep operators informed and in control, the app collects real-time data from drones, such as GPS information, height, speed, and more.

- By providing waypoints and flight information in advance, users can make routine patrols easier with pre-planned drone routes.

3.3.4. Video Stream Processing

- The system receives live video feeds from drones, processing them in real-time to detect encroachments through deep learning algorithms.
- It will generate alert if encroachment is detected to ensure timely responses to potential threats.

3.3.5. Alert Management

- The system efficiently manages alerts by grouping them based on severity and location, allowing for rapid situational analysis.
- Alerts are promptly delivered through various channels such as email, SMS, and push notifications, ensuring responsible personnel are notified for swift responses.

3.3.6. User Instruction Processing and Live Streaming

- The Live Streaming Interface ensures controlled access to live camera feeds based on user roles, while collecting real-time video and data from nearby drones.
- It prioritizes data privacy through encryption and robust security measures to prevent unauthorized access or manipulation of camera feeds.

3.3.7. Drone Scheduling and Operation

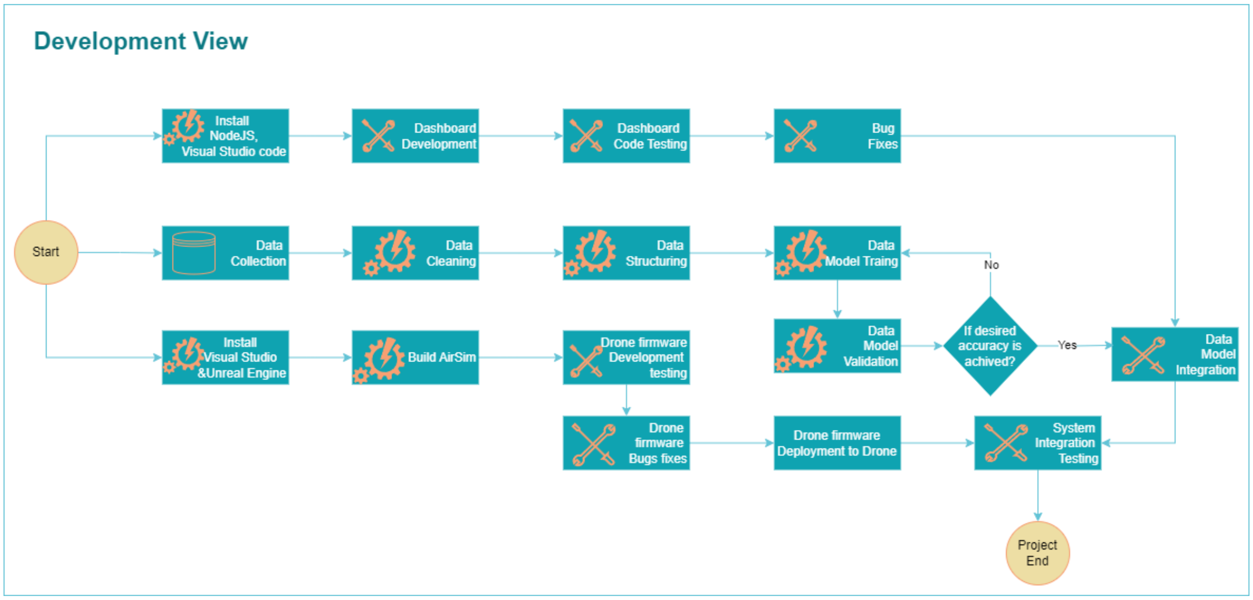
- The Drone Operating Application not only controls drone operations but also schedules routine monitoring patrols, ensuring hourly surveillance for enhanced security and threat detection.

3.4. Development View

3.4.1. Introduction and Rationale

- The Development View in the drone-based intrusion detection system's design gives insight into the software development perspective.
- It entails the selection of programming languages, the incorporation of machine learning models, and the development of a unified software infrastructure that enables real-time data processing, alert production, and responsive surveillance operations.
- This viewpoint assures that software components are efficiently planned, built, and maintained to serve the system's mission-critical functions.
- The development view can be segmented into 3 parts -
- Training Deep Learning model to identify encroachment.
- Simulation and configuration of drone flight patterns.
- Dashboard creation and setting up communication channels to drones.

3.4.2. Architectural Model


Development View
Start
Install NodeJS, Visual Studio code
Dashboard Development
Dashboard Code Testing
Bug Fixes
Data Collection
Data Cleaning
Data Structuring
Data Model Traing
No
Install Visual Studio &Unreal Engine
Build AirSim
Drone firmware Development testing
Data Model Validation
If desired accuracy is achived?
Yes
Data Model Integration
Drone firmware Bugs fixes
Drone firmware Deployment to Drone
System Integration Testing
Project End

3.4.3. User Dashboard Development

- NodeJS is employed for web-app interface development, enabling user instructions and drone operation configuration.
- Communication channels with the application server connect to all drones for data exchange, while the application server serves as a central hub for user commands and database connectivity, allowing access to real-time drone data and historical records.

3.4.4. Encroachment Detection Model Training

- Python is the language of choice for training the YOLO8 model, enabling object detection in drone footage, and extracting object positions.
- An algorithm is designed to calculate object distance and provide drone motion directives, enhancing the system's ability to track and follow detected objects effectively leveraging CV2.

3.4.5. Drone Simulation Setup and Firmware Development

- The Webots Drone Simulation Setup facilitates realistic testing and development of drone operations in various environments, ensuring system readiness.
- Drone firmware development is essential to customize drone behavior, optimizing performance and responsiveness for real-world encroachment detection scenarios.

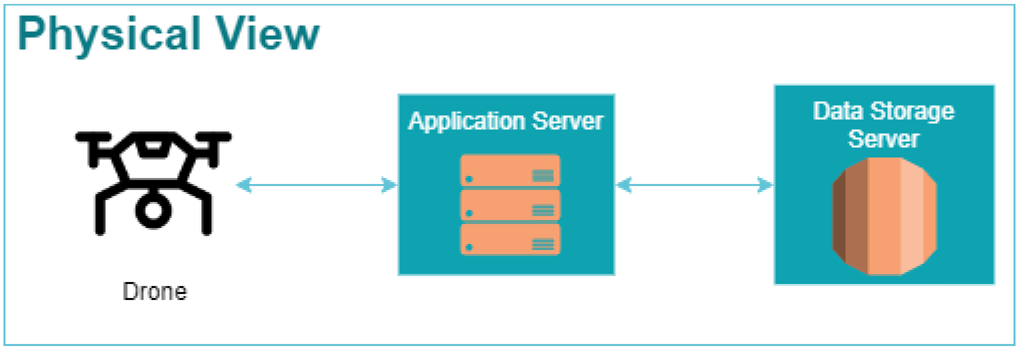
3.5. Physical View

3.5.1. Introduction and Rationale

- The Physical View of the drone-based encroachment detection system encompasses the hardware and infrastructure aspects.
- It involves the deployment of drones with specific capabilities and sensors, along with the selection of cameras for high-quality footage capture.
- The control network plays a crucial role in facilitating communication and data transmission.
- This view ensures that the hardware components are robust and resilient to withstand varying environmental conditions.

- It also includes considerations such as power supply and charging stations to maintain uninterrupted surveillance and swift threat response capabilities.

3.5.2. Architectural Model



3.5.3. Drone

- The drone conducts encroachment monitoring by flying in designated areas, with options for scheduled monitoring at intervals or demand-driven focused surveillance.
- It adapts to adverse weather conditions to ensure operational effectiveness and security objectives.

3.5.4. Bare Metal Application Server

- The Application server receives user dashboard requests, communicates instructions to drones, and collects live drone data.
- It streams real-time information to the user dashboard, encrypts and transfers data to the Data Storage server, and accesses past data from the Data Storage server, decrypting and delivering it to the user dashboard upon request.

3.5.5. Database and Data Storage Server

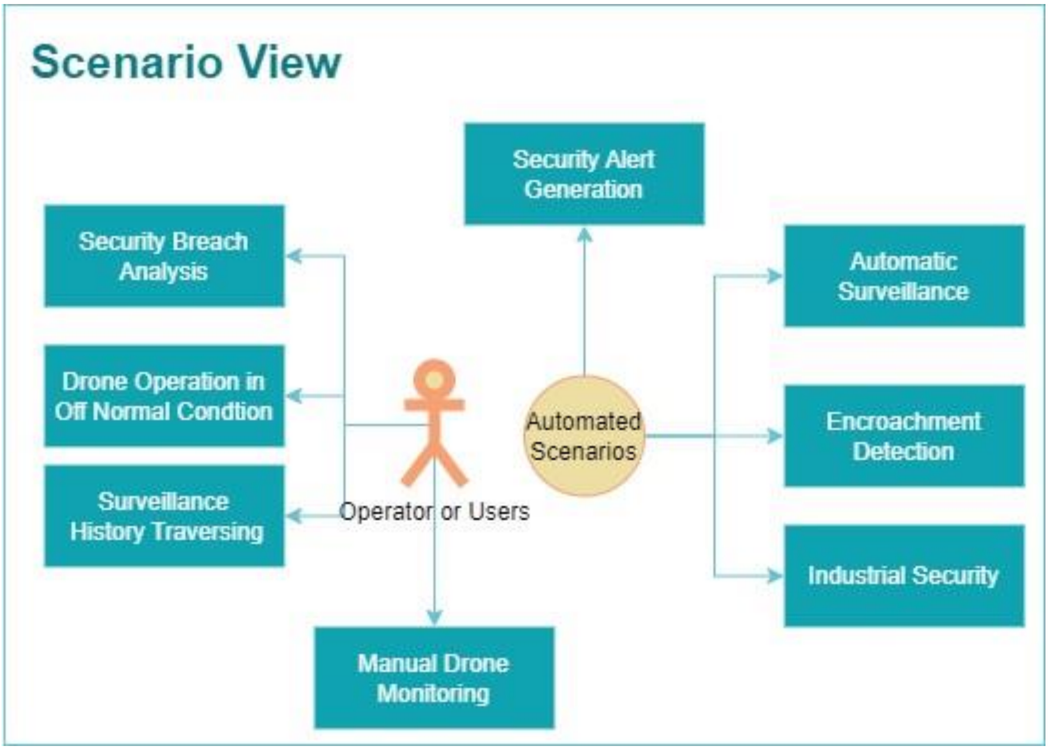
- The Data Storage Server receives encrypted drone data from the Application server, stores, and indexes it for efficient access.
- It retrieves stored data upon request, automatically managing data retention by removing older records after a configured time, enhancing storage efficiency.

3.6. Scenario View

3.6.1. Introduction and Rationale

- The Situation A look inside the drone-based incursion detection system reveals information about its flexibility in real-world circumstances.
- It includes user interactions with operators, scheduled monitoring, and real-time warning in the event of an incursion.
- The perspective displays the system's efficiency in a variety of settings, ranging from industrial sites to rural landscapes.
- It demonstrates the system's adaptability and response to changing operating needs, allowing it to be used in a variety of applications.

3.6.2. Architectural Model



3.6.3. Automated Scenarios

- The system automatically detects encroachments in camera footage, generating alerts for designated personnel.
- Continuous surveillance is carried out autonomously, monitoring for encroachments in real-time without manual intervention.
- Advanced data models ensure highly accurate and precise detection of encroachments, enhancing security for industrial applications with state-of-the-art measures.

3.6.4. User Driven Manual Scenarios

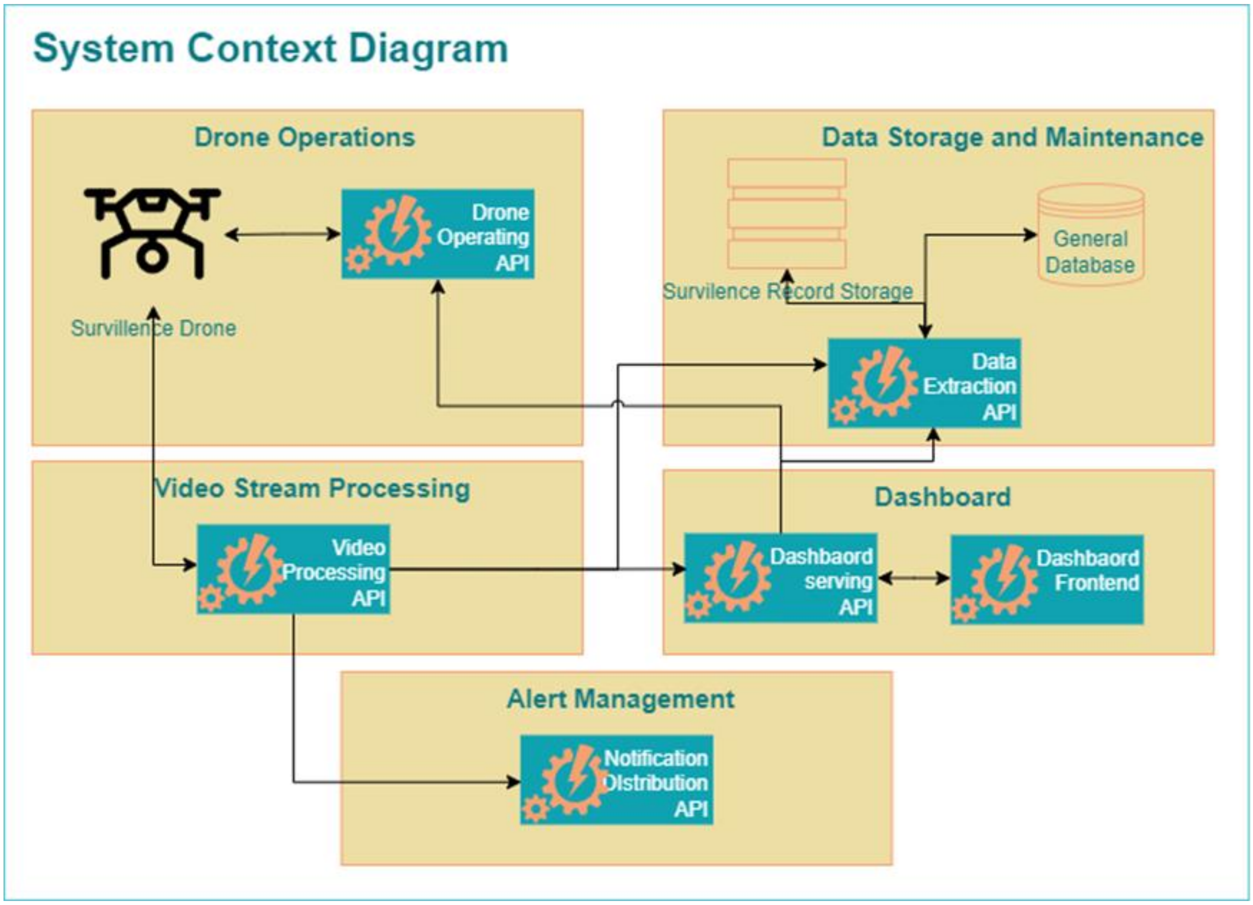
- Users can seamlessly review camera footage in real-time or retrospectively, offering detailed examination through zoom and navigation controls.
- Drones exhibit reliable performance even in challenging weather conditions, including rain, extreme heat, and low battery situations.
- Access to one month of stored footage empowers users for retrospective review and in-depth analysis of surveillance history.

4. Detailed Design

4.1. Overview

- The drone-based encroachment detection system is intended to improve security and surveillance in restricted or sensitive areas with unmanned aerial vehicles (UAVs), often known as drones.
- Drones are fitted with specific sensors and a dedicated operating program monitor and identify unwanted entries into secured zones in real time.
- By notifying security professionals of threats and incursions, this technology aids in the protection of essential infrastructure, private assets, and public areas.

4.2. System Context Diagram



4.3. Drone Operations

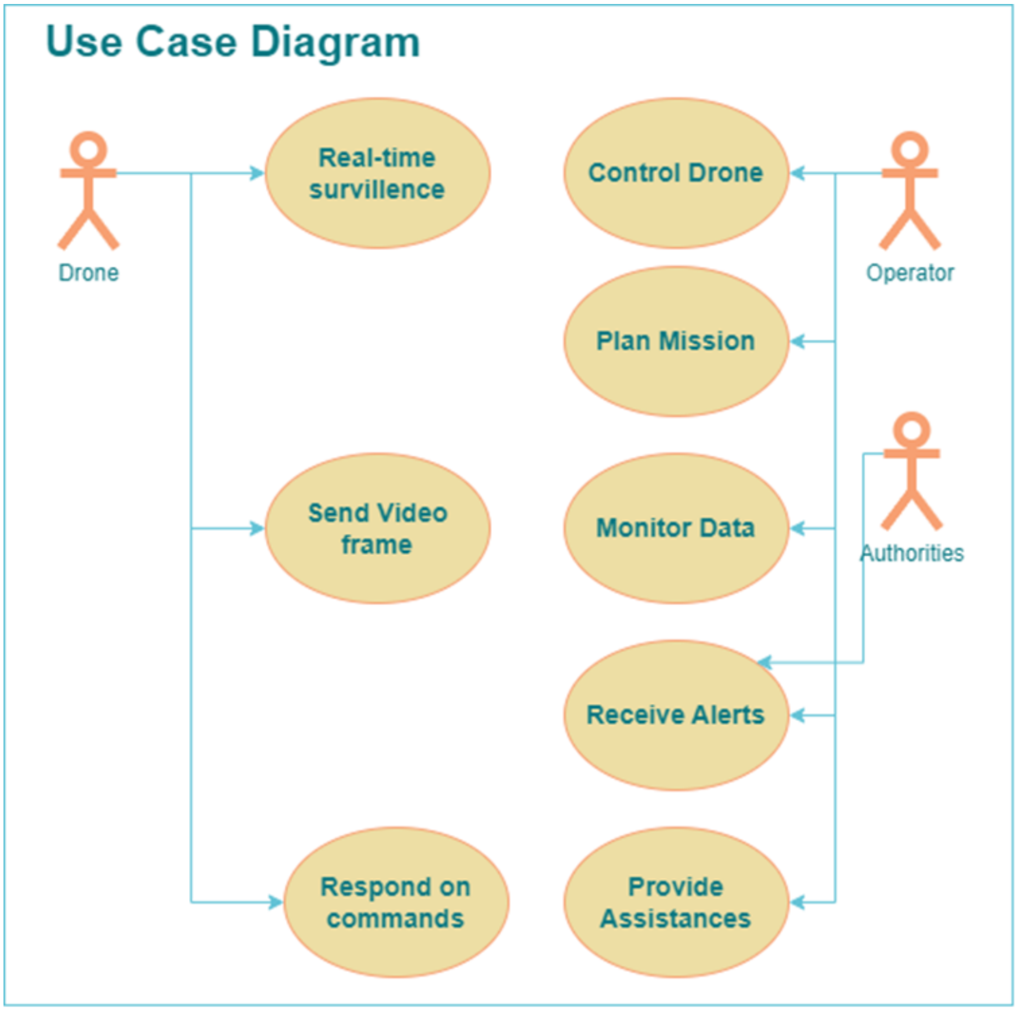
4.3.1. High-Level Overview

- A drone is a highly maneuverable, remote-controlled, or autonomous aircraft outfitted with specialized sensors for visual and environmental data collection for surveillance purposes.
- The software that acts as the primary interface for operators to manage and monitor drone operations is known as the drone operating application.
- It allows operators to control the drone's motions, modify sensor settings, and get real-time data, allowing them to respond to encroachments more quickly.

4.3.2. Technical Details

- **Real-Time Surveillance and Data Collection** - Drones fitted with modern sensors, such as high-resolution cameras, and other environmental sensors, may collect real-time data.
- This information can be useful in detecting and analyzing potential encroachments, security breaches, or safety problems.
- The capacity of the drone to visit difficult-to-reach or hazardous regions makes it crucial for getting timely information.
- **Integration with Alert Systems** - When odd behavior or encroachments are noticed, the drone operating application is integrated with alert systems, allowing it to send alarms and notifications.
- This integration guarantees that contemporary issues are addressed in a timely manner.
- **Scalability and Coverage** - Drones can cover enormous regions fast and effectively, making them very scalable for a variety of deployment situations.
- They can fly over large areas of land and infrastructure, offering a complete picture that would be difficult to acquire with fixed surveillance systems or staffed patrols.
- This scalability is vital for monitoring broad perimeters, critical infrastructure, and disaster zones.
- **Adaptability and Flexibility** - The drone and the operating application can be tailored to suit specific use cases.
- They can adapt to different environmental conditions, security requirements, and operational needs, making them highly flexible for a wide range of applications.

4.3.3. Design Element User Case Diagram



4.3.3.1. Use Case 1: Real-Time Surveillance

Use Case ID:	001		
Use Case Name:	Real-Time Surveillance		
Traceability:			
Created By:	Ameya Shahu	Last Updated By:	10/19/2023
Date Created:	10/19/2023	Date Last Updated:	10/19/2023

Actor:	Drone
Description:	This use case describes the process by which the drone conducts real-time surveillance over a designated area to detect any encroachments.
Preconditions:	• The drone is fully charged and operational. • The drone's cameras and sensors are functional.

	• The flight path has been predefined and uploaded to the drone's system.
Postconditions:	• The drone has completed the surveillance mission. • The surveillance data has been recorded and is available for review.
Primary Pathway:	• The operator initiates the surveillance mission. • The drone follows the predefined flight path, capturing real-time video and sensory data. • The drone processes and streams data to the operator's dashboard. • The operator monitors the data feed for any encroachments. • The drone completes the surveillance loop and returns to the starting position.
Alternate Pathways:	• If an encroachment is detected, the drone sends an alert to the operator and continues to monitor the area. • If the drone encounters a flight or mechanical issue, it returns to base immediately.
Exception Pathways:	• In case of a low battery or signal loss, the drone initiates an emergency landing protocol.

4.3.3.2. Use Case 2: Control Drone

Use Case ID:	002		
Use Case Name:	Control Drone		
Traceability:			
Created By:	Ameya Shahu	Last Updated By:	10/19/2023
Date Created:	10/19/2023	Date Last Updated:	10/19/2023

Actor:	Drone Operator
Description:	This use case details the operator's interaction with the drone to initiate, control, and manage its operations during a mission.
Preconditions:	• The drone is operational and in communication range. • The operator is trained and authorized to control the drone.
Postconditions:	• The drone has been successfully controlled and managed by the operator.
Primary Pathway:	• The operator sends a command to initiate the drone's operations. • The drone receives the command and executes the start-up sequence. • The operator continues to send control commands during the drone's flight.

Alternate Pathways:	• The operator sends a command to land the drone in case of an emergency.
Exception Pathways:	• Communication failure between the operator and the drone.

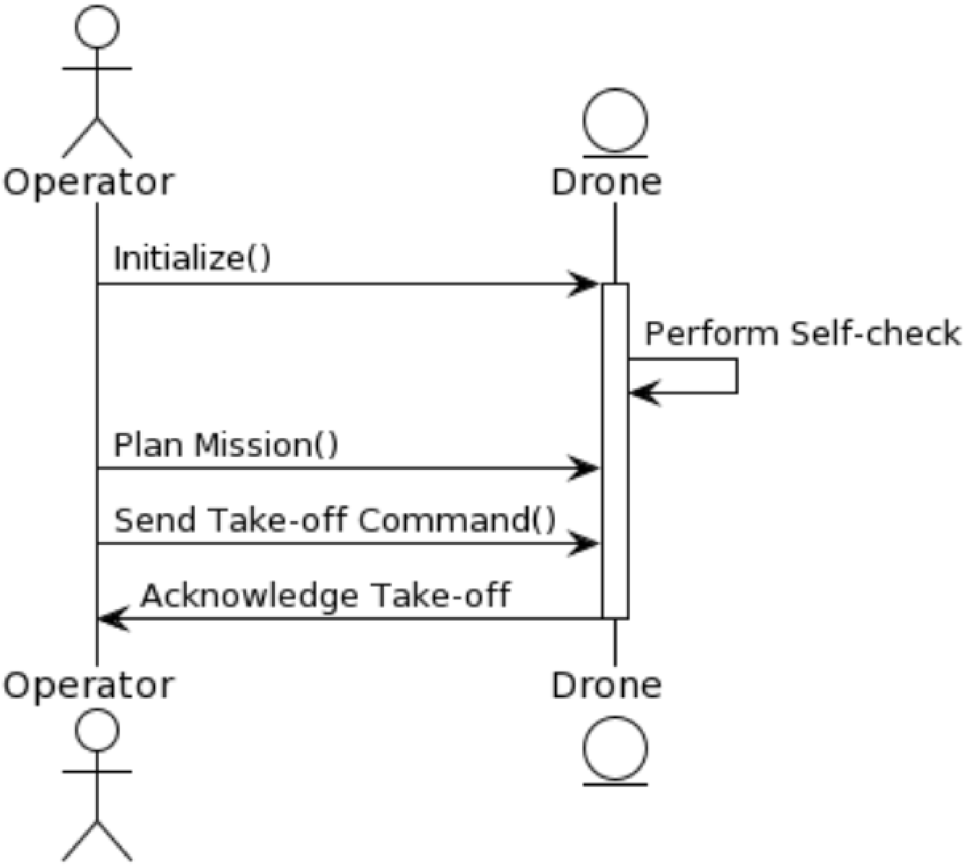
4.3.3.3. Use Case 3: Plan Mission

Use Case ID:	003		
Use Case Name:	Plan Mission		
Traceability:			
Created By:	Ameya Shahu	Last Updated By:	10/19/2023
Date Created:	10/19/2023	Date Last Updated:	10/19/2023

Actor:	Operator
Description:	This use case describes the operator's process to plan and upload a mission for the drone to execute, including route, altitude, and behavior.
Preconditions:	• The operator has access to the mission planning tools. • The drone is available and ready to receive mission parameters.
Postconditions:	• The drone has a complete and executable mission plan.
Primary Pathway:	• The operator accesses the mission planning tool. • The operator inputs the necessary parameters for the mission. • The mission is uploaded to the drone. • The drone acknowledges receipt of the mission plan.
Alternate Pathways:	• The operator adjusts the mission plan in response to real-time events or data.
Exception Pathways:	• Mission planning tool fails to upload the plan to the drone.

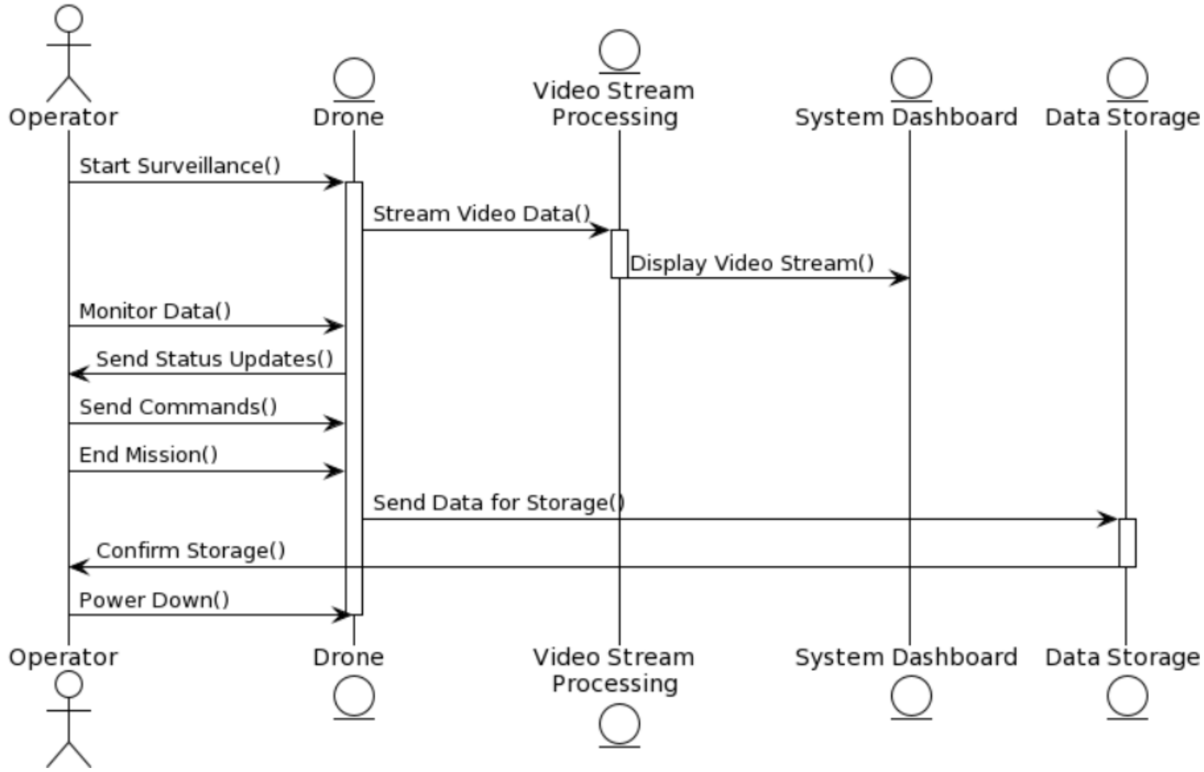
4.3.4. Design Element Class Diagram

4.3.4.1. Sequence Diagram 1: Mission Planning and Launch



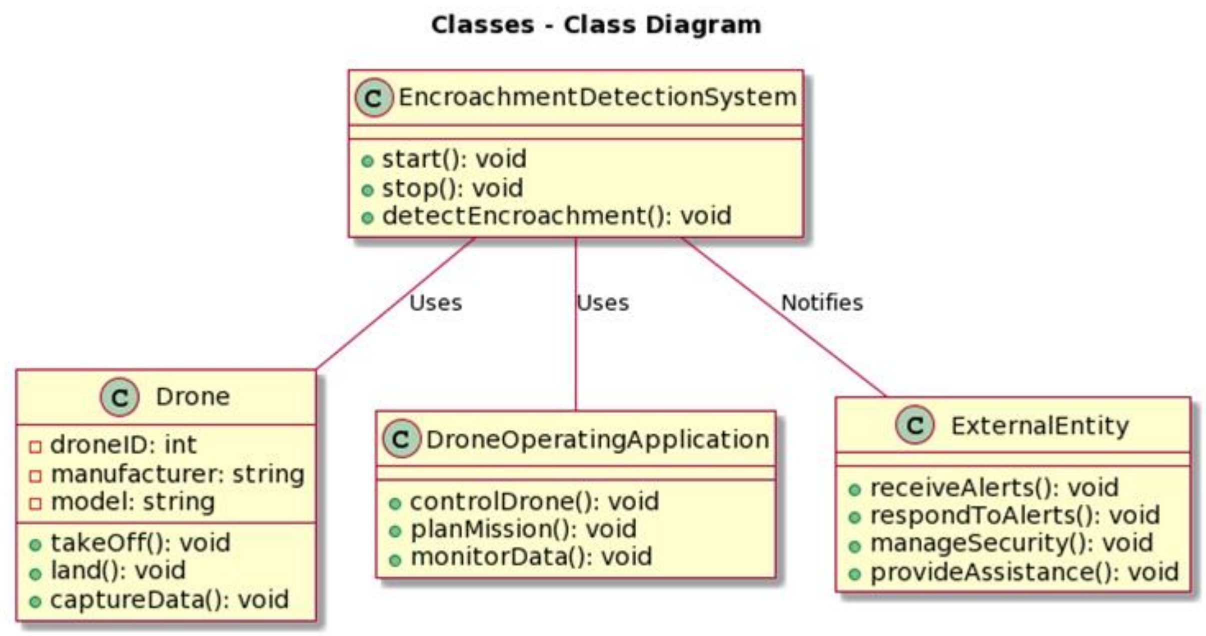
- The "Operator" starts the sequence by initializing the "Drone".
- The "Drone" performs a self-check to ensure all systems are functional.
- The "Operator" then plans the mission, inputting parameters such as the area to survey, altitude, and flight path.
- After planning, the "Operator" sends a take-off command to the "Drone".
- The "Drone" acknowledges the take-off and begins the mission.

4.3.4.2. Sequence Diagram 2: Surveillance and Data Handling



- The "Operator" commands the "Drone" to start surveillance.
- The "Drone" streams video data to the "Video Stream Processing" system.
- The "Video Processing" system sends the video stream to be displayed on the "System Dashboard".
- The "Operator" monitors data and may send various commands to the "Drone" during the mission.
- Upon mission completion, the "Drone" sends the gathered data to "Data Storage" for safekeeping and future analysis.
- The "Storage" system confirms the storage of data to the "Operator".
- Finally, the "Operator" powers down the "Drone", concluding the operation.

4.3.5. Other Relevant UML Diagrams



4.4. Video Stream Processing Module

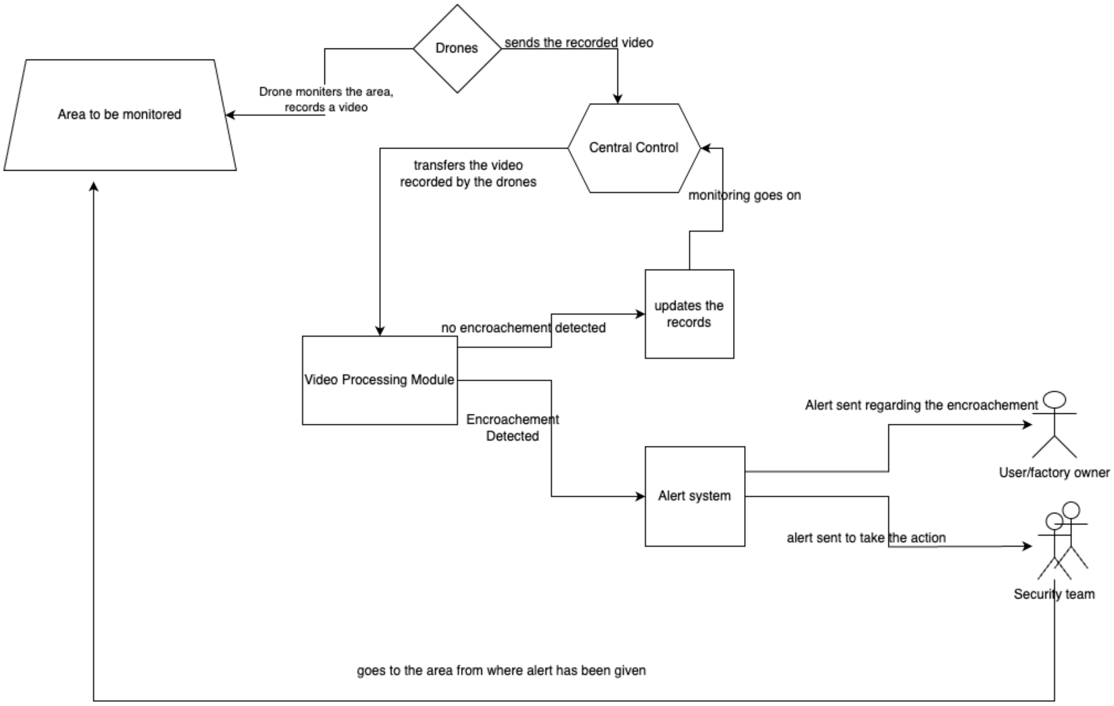
4.4.1. High-Level Overview

- The video processing component of our system is essential for real-time analysis of drone footage to detect encroachments in the industrial area.
- Video Processing Module is an important part of our system since it allows for real-time processing of drone data for incursion identification in industrial regions.
- It makes use of image processing and computer vision techniques to accurately identify potential security concerns.

4.4.2. Technical Details

- The module communicates with drones that are actively monitoring the industrial area, processing video data in real-time using the YOLO algorithm.
- It communicates with external entities such as the Central Control System and External Storage.
- The design prioritizes quick and accurate operation, which is critical for the system's intrusion detection capabilities.

4.4.3. Design Element User Case Diagram


Drones
sends the recorded video
Area to be monitored
Drone moniters the area, records a video
Central Control
transfers the video recorded by the drones
monitoring goes on
updates the records
no encroachement detected
Video Processing Module
Encroachement Detected
Alert system
Alert sent regarding the encroachement
User/factory owner
alert sent to take the action
Security team
goes to the area from where alert has been given

4.4.3.1. Use Case 1: Video Surveillance Processing

Use Case ID:	001		
Use Case Name:	Video Surveillance Processing		
Traceability:			
Created By:	Pravalika Mukkiri	Last Updated By:	11/19/2023
Date Created:	11/19/2023	Date Last Updated:	11/19/2023

Actor:	• Drone, Security Team, User/Factory Owner •
Description:	• This use case represents the normal operation of the system during video surveillance. It involves capturing video data, processing it for encroachments, triggering alarms if threats are detected, and notifying relevant parties.
Preconditions:	• Drones actively surveilling, video data available.
Postconditions:	• Encroachment detection results generated, alerts sent, processed data archived.
Primary Pathway:	• Drone captures video → Sent to central control system → Processed for encroachments → Alarm triggered if detected → Alerts delivered → Encroachment data archived.

Alternate Pathways:	• In case of poor video quality or incomplete data, the system may request retransmission of video data from the drone.
Exception Pathways:	• If the central control system is unavailable, the system may store the video data locally and retry transmission once connectivity is restored.

4.4.3.2. Use Case 2: Scheduled Maintenance

Use Case ID:	002		
Use Case Name:	Scheduled Maintenance		
Traceability:			
Created By:	Pravalika Mukkiri	Last Updated By:	11/19/2023
Date Created:	11/19/2023	Date Last Updated:	11/19/2023

Actor:	Maintenance Team
Description:	This use case represents the scheduled maintenance of the system to ensure its continued functionality.
Preconditions:	• Scheduled maintenance time reached.
Postconditions:	• Maintenance tasks completed, system operational.
Primary Pathway:	• Maintenance team notified → System enters maintenance mode → Scheduled tasks performed → System exits maintenance mode
Alternate Pathways:	• If a critical issue is identified during maintenance, the system may remain in maintenance mode until the issue is resolved.
Exception Pathways:	• If the maintenance team encounters difficulties, the system may roll back to the previous operational state to avoid disruptions.

4.4.3.3. Use Case 3: Emergency Shutdown

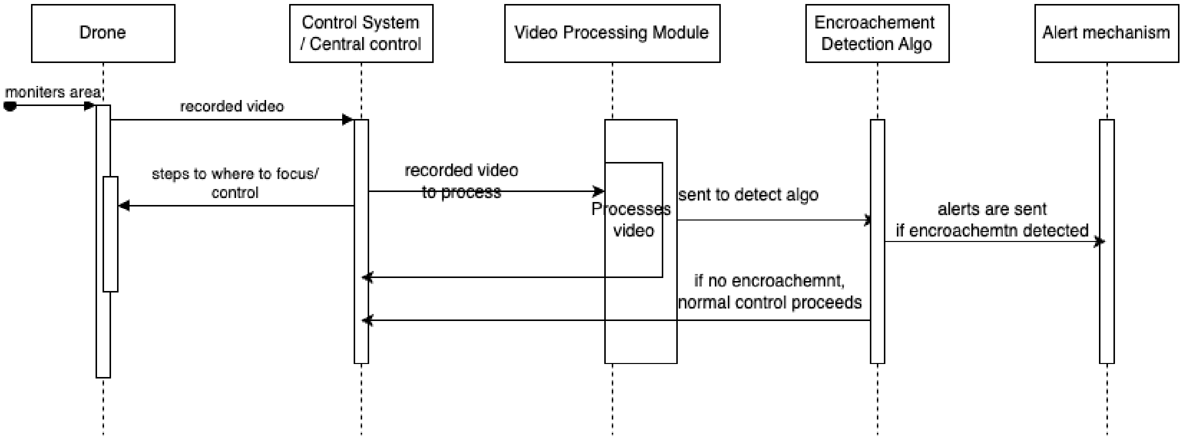
Use Case ID:	003		
Use Case Name:	Emergency Shutdown		
Traceability:			
Created By:	Pravalika Mukkiri	Last Updated By:	11/19/2023
Date Created:	11/19/2023	Date Last Updated:	11/19/2023

Actor:	Emergency Operator

Description:	This use case represents the emergency shutdown of the system in response to a critical situation. It involves an emergency operator triggering the shutdown, ceasing normal operations, and activating emergency protocols.
Preconditions:	• Emergency detected.
Postconditions:	• System safely shut down; emergency protocols initiated.
Primary Pathway:	• Emergency operator triggers shutdown → Central control system ceases normal operation → Emergency protocols activated → System safely shuts down.
Alternate Pathways:	• If the emergency is resolved before the shutdown is complete, the system may abort it.
Exception Pathways:	• If the shutdown process encounters errors, the system may initiate a safe mode to prevent further complications.

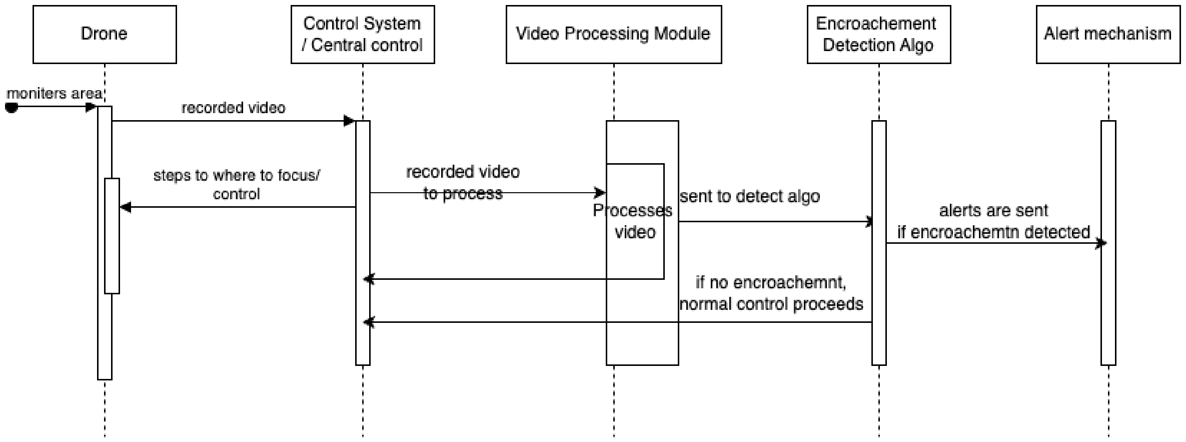
4.4.4. Design Element Class Diagram

4.4.4.1. Sequence Diagram 1


Drone
Control System / Central control
Video Processing Module
Encroachement Detection Algo
Alert mechanism
moniters area
recorded video
steps to where to focus/ control
recorded video to process
Processes video
sent to detect algo
alerts are sent if encroachemtn detected
if no encroachemnt, normal control proceeds

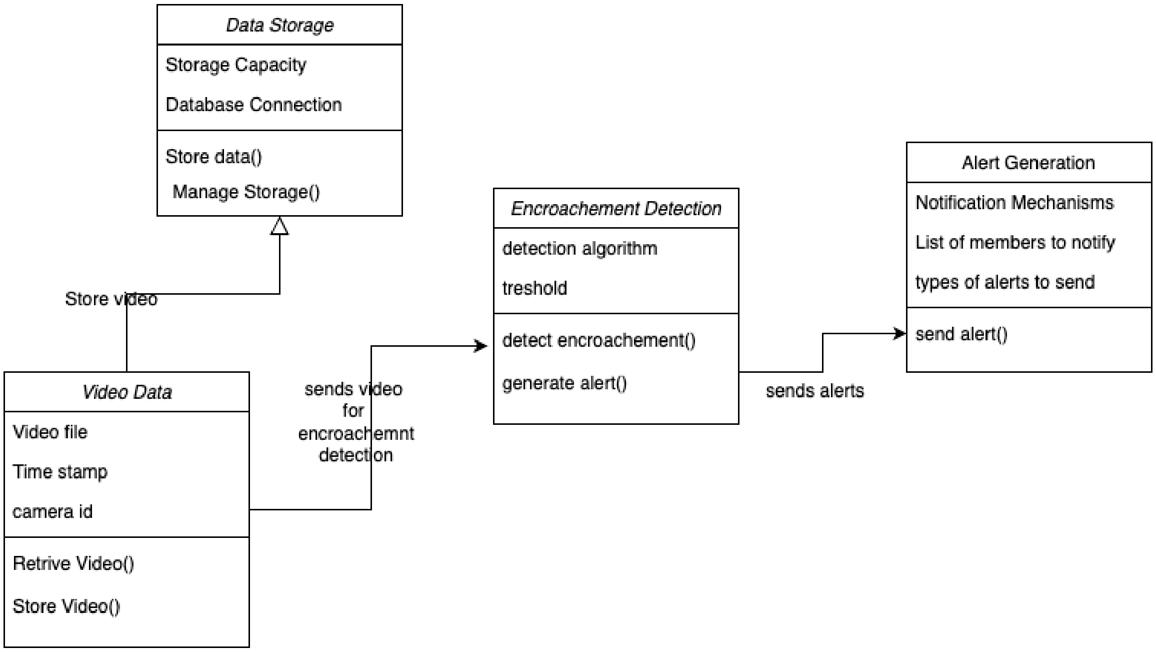
- This sequence diagram outlines the dynamic interactions within the Video Processing Module during a typical threat detection scenario.
- The drone captures video data, which is subsequently delivered to the central control system for processing.
- The video data is routed through the Video Data class, where it is checked for invasion. When an encroachment is detected, the Alert Generation class is engaged, which sends notifications to the central control system or security personnel.
- Simultaneously, the Data Storage class maintains the storage of processed data, assuring its future accessibility.

4.4.4.2. Sequence Diagram 2


Drone
Control System / Central control
Video Processing Module
Encroachement Detection Algo
Alert mechanism
moniters area
recorded video
steps to where to focus/ control
recorded video to process
Processes video
sent to detect algo
alerts are sent if encroachemtn detected
if no encroachemnt, normal control proceeds

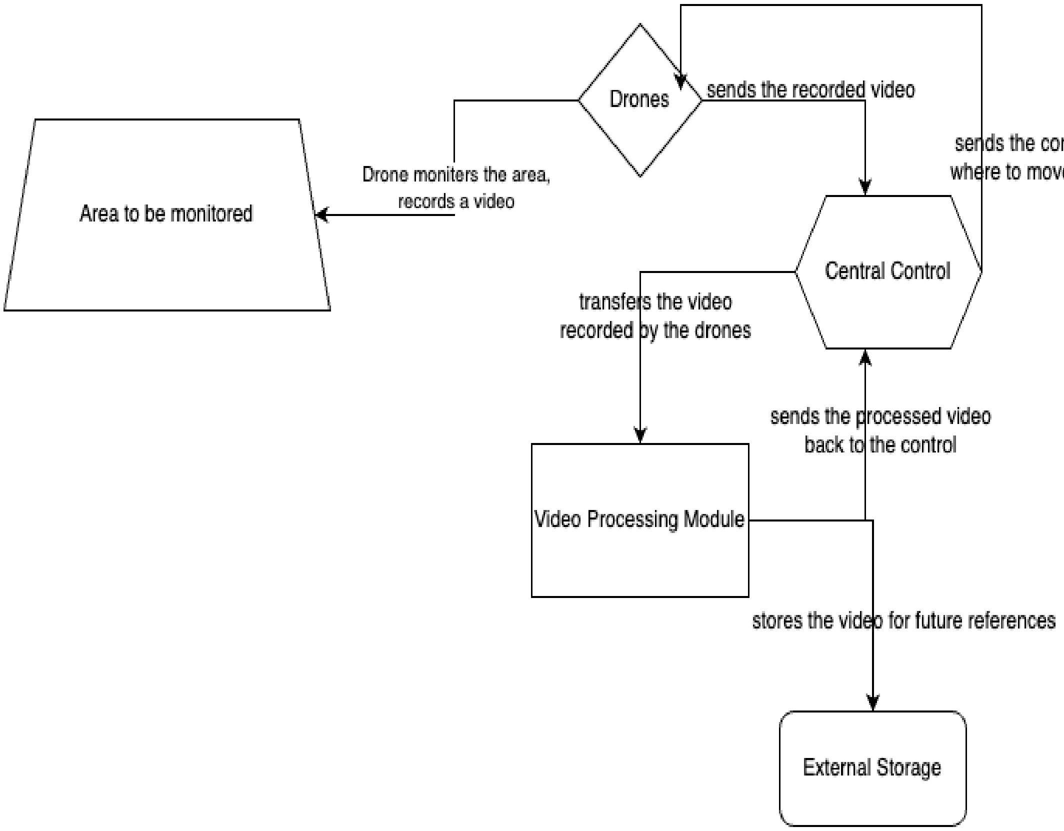
- This flow diagram shows interactions occurring when video data is retrieved and stored in the Video Processing Module.
- The Video Data class retrieves raw video data, including properties such as Timestamp, Video File, and Camera ID, to begin the process.
- The video data is subsequently stored using the "StoreVideo" method, which interacts with the Data Storage class.
- The Data Storage class provides effective storage capacity and data retention management using methods such as "StoreData" and "ManageStorage."
- The sequence demonstrates how classes work together seamlessly to handle video data from retrieval to storage.

4.4.5. Other Relevant UML Diagrams



- This diagram illustrates how the classes interact with each other and provides a blueprint for designing the system.

- Visual depiction at a high level.
- Describes the connections between the Video Processing Module and external entities or important system components.


Drones
sends the recorded video
Area to be monitored
Drone moniters the area, records a video
sends the control to where to move/focus
Central Control
transfers the video recorded by the drones
sends the processed video back to the control
Video Processing Module
stores the video for future references
External Storage

- An overview of the Video Processing Module's role in the system design.
- External entities' interactions are illustrated.
- The impact of external components on the module is emphasized, and vice versa.

4.5. Dashboard

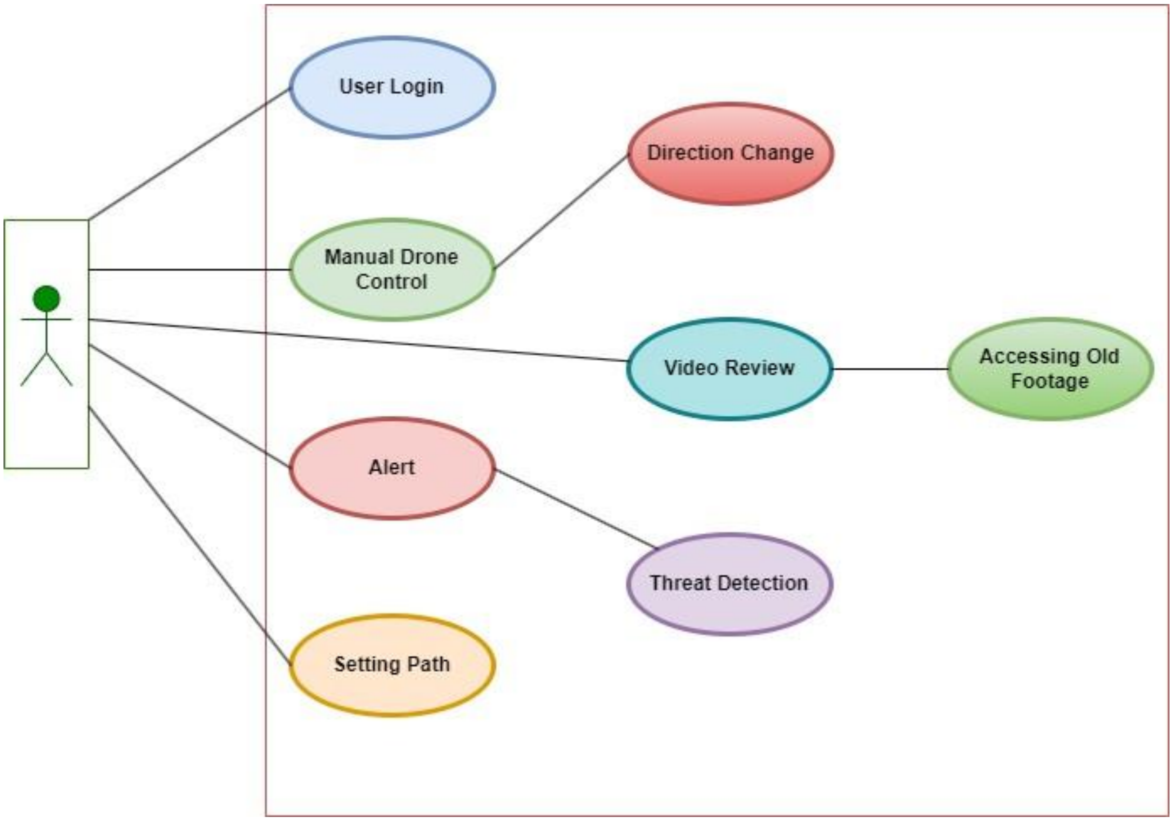
4.5.1. High-Level Overview

- The user needs an application to send commands and receive information about the system's main functionality.
- Dashboard enables the user to control the drone, view footage and be informed about the intrusion if any.

4.5.2. Technical Details

- Live Stream Dashboard: This sub-component provides the user with different live footage from the camera of the drone.
- Drone Manual Control: When the user wants to manually control the drone, have drone information or draw set drone paths, this provides the user with the controls.
- Alert Display: If a drone finds an intrusion, then the user is notified using this API.
- Reviewing Earlier Footage: The footage from the drone is stored to be viewed later own using the dashboard with the part of the application.

4.5.3. Design Element User Case Diagram



4.5.3.1. Use Case 1: User Login

Use Case ID:	001		
Use Case Name:	User Login		
Traceability:			
Created By:	Aditya Pant	Last Updated By:	11/19/2023
Date Created:	11/19/2023	Date Last Updated:	11/19/2023

Actor:	Operator
Description:	This use case entails users logging in and logging out
Preconditions:	• The operator has the username and password
Postconditions:	• The user ID and password match and login is successful.
Primary Pathway:	• The operator goes to the dashboard where they put their ID and password to login.
Alternate Pathways:	• None

Exception Pathways:	- The user ID and password mismatch.

4.5.3.2. Use Case 2: Manual Drone Control

Use Case ID:	002		
Use Case Name:	Manual Drone Control		
Traceability:			
Created By:	Aditya Pant	Last Updated By:	11/19/2023
Date Created:	11/19/2023	Date Last Updated:	11/19/2023

Actor:	Operator
Description:	This use case describes the operator's process of controlling the drone.
Preconditions:	• Logged into the system as the operator. • Knowledge about controlling drones1.
Postconditions:	• Drone is maneuvered in the right path.
Primary Pathway:	• The operator logs into the dashboard using user ID and password. • The operator uses the Manual Drone Controller to maneuver the drone.
Alternate Pathways:	• Use the physical drone controller instead of the software one.
Exception Pathways:	• The controller loses contact with the drone.

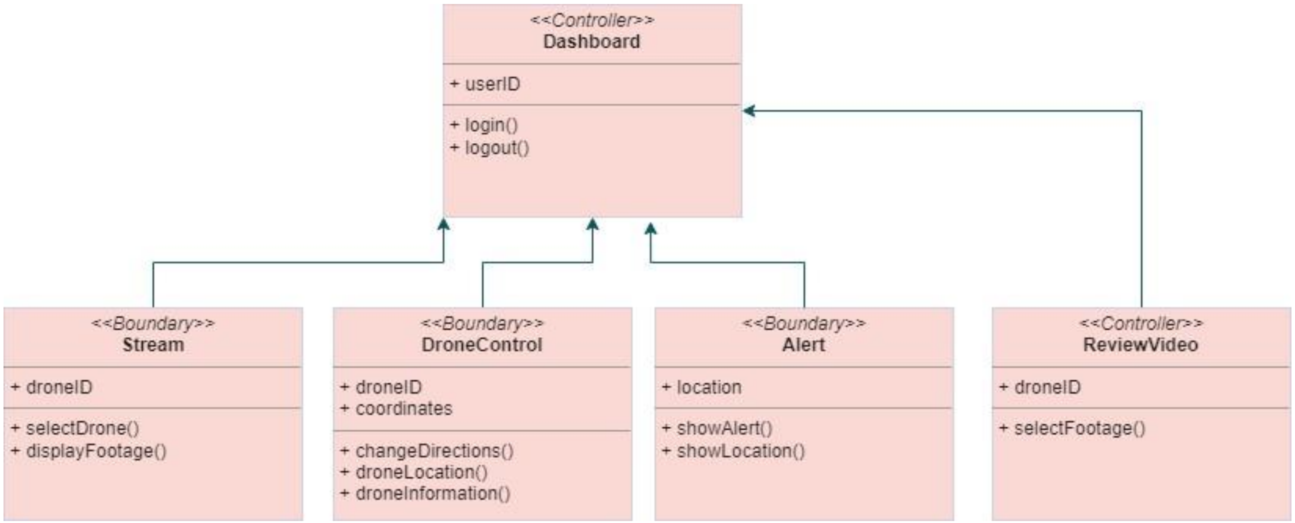
4.5.3.3. Use Case 3: Alert

Use Case ID:	003		
Use Case Name:	Alert		
Traceability:			
Created By:	Aditya Pant	Last Updated By:	11/19/2023
Date Created:	11/19/2023	Date Last Updated:	11/19/2023

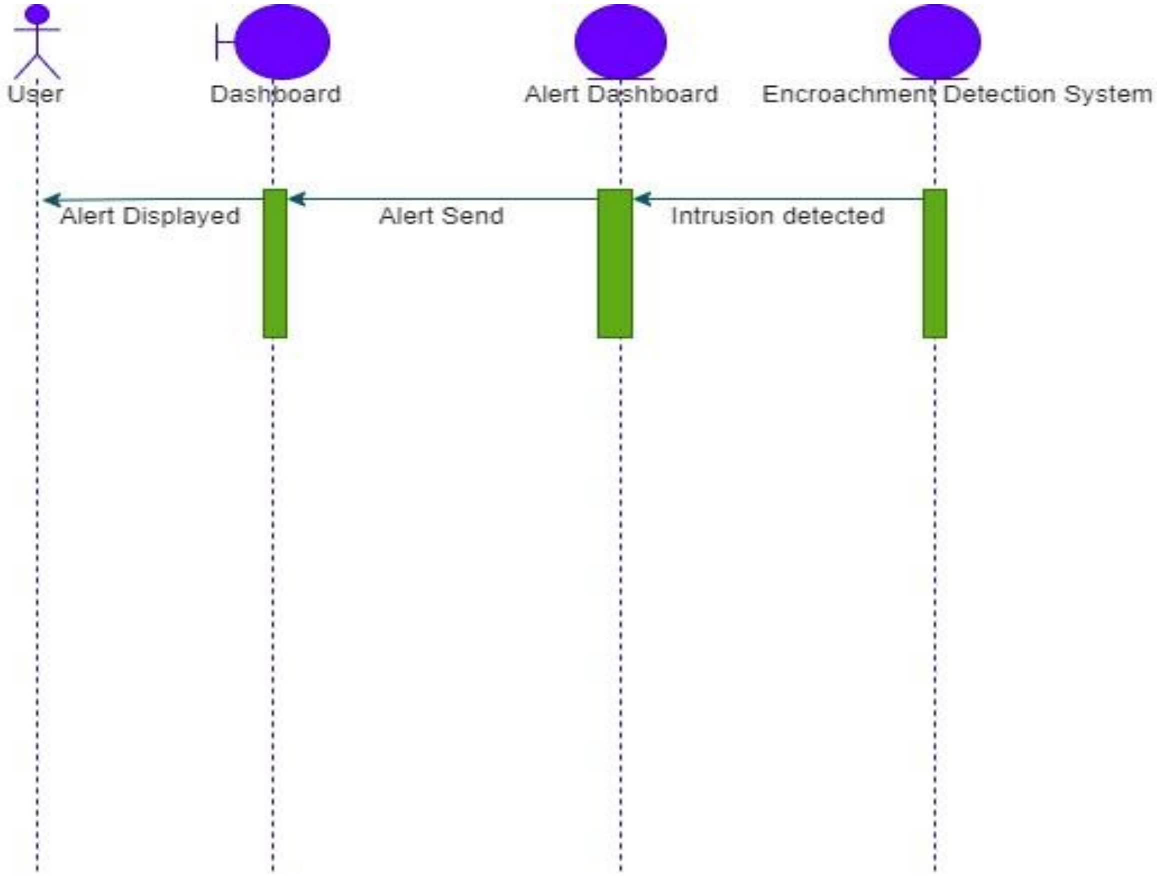
Actor:	Operator
Description:	This use case describes the operator's process of receiving alerts.
Preconditions:	• Logged into the system as the operator.

Postconditions:	- Requirement actions taken against the intrusion.
Primary Pathway:	- The operator logs into the dashboard using user ID and password. - The operator watches out for an alert in the dashboard.
Alternate Pathways:	- The intrusion is not detected, and the alert is not issued.
Exception Pathways:	- The alert malfunctions and does not reach all required users.

4.5.4. Design Element Class Diagram

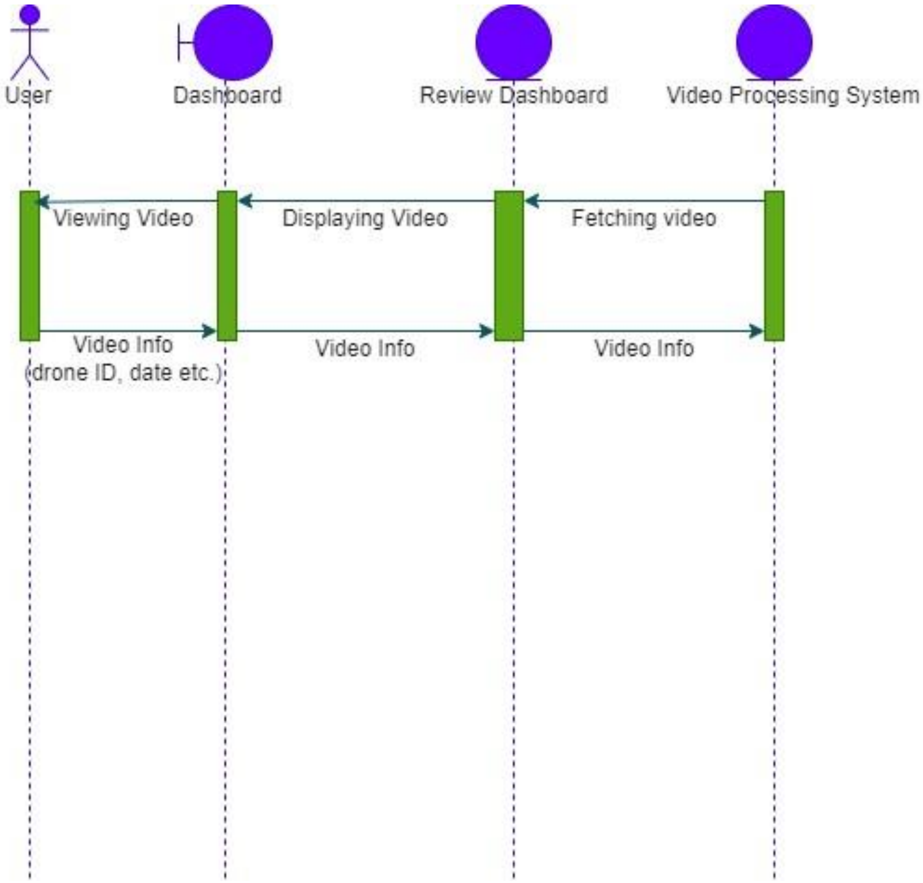


4.5.4.1. Sequence Diagram 1



- Drone surveillance is constantly on going.
- If the drone spots the intrusion, then the alert dashboard is informed.
- The alert dashboard sends the alert to the user who can view it and has valuable information like coordinates of the intrusion and follow the intrusion.

4.5.4.2. Sequence Diagram 2



- For viewing the video, a request containing video date, drone ID and others is sent from the dashboard.
- The dashboard sends this information to the Review API that collates the information and sends it to the Video Processing System.
- Appropriate video is fetched and displayed to the user.

4.6. Alert Management System

4.6.1. High-Level Overview

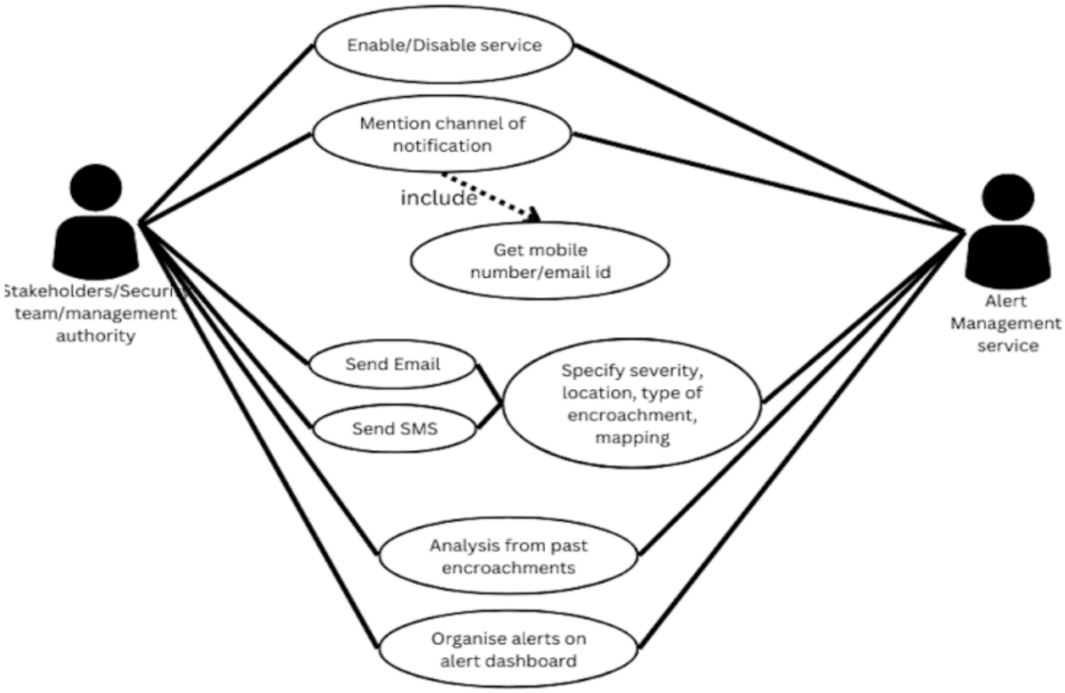
- With the help of real-time dashboards and map overlays, the Alert API effectively gathers and classifies drone-generated alerts, facilitating their quick distribution and improving situational awareness.
- It is important because it makes it easier for stakeholders to coordinate smoothly and respond quickly, which leads to better decision-making.

4.6.2. Technical Details

- Alert Aggregation and Categorization: The API is responsible for gathering drone-generated alerts and classifying them according to the kind, location, and intensity of encroachment that was found.
- Notification and Alert Dissemination: It oversees the alerts' delivery to response teams or pertinent stakeholders through a variety of channels, including email, SMS, push alerts, and more.
- Analysis: The API makes it easier to analyze trends and generate reports that provide information about encroachment-related patterns and trends.

- Provision of a Real-time Alert Dashboard: It drives a dashboard that shows information about each alert in real-time, facilitating prompt situational analysis and well-informed decision-making.
- Integrating with mapping services allows the API to overlay alerts on a map, giving users precise location information and a visual representation of the alerts for improved situational awareness.

4.6.3. Design Element User Case Diagram


Enable/Disable service
Mention channel of notification
include
Get mobile number/email id
Stakeholders/Securi team/management authority
Alert Management service
Send Email
Send SMS
Specify severity, location, type of encroachment, mapping
Analysis from past encroachments
Organise alerts on alert dashboard

4.6.3.1. Use Case 1: Alert Configuration and Setup

Use Case ID:	401		
Use Case Name:	Alert Configuration and Setup		
Traceability:			
Created By:	Lalit Arvind Balaji	Last Updated By:	Lalit Arvind Balaji
Date Created:	11/19/23	Date Last Updated:	11/19/23

Actor:	Admin User
Description:	Setting up profile in notification interface to obtain notifications in case any encroachment occurs
Preconditions:	User is authorized personnel in industry
Postconditions:	Interface obtains user’s preference and notification channel details
Primary Pathway:	• User enables notifications • User chooses preferred notification channel: SMS/E-Mail

	• User provides details for chosen notification channel: Mobile no./E-Mail ID
Alternate Pathways:	• User disables notifications
Exception Pathways:	• Provided E-Mail ID/Phone no does not exist

4.6.3.2. Use Case 2: Alert Notification

Use Case ID:	402		
Use Case Name:	Alert Notification		
Traceability:			
Created By:	Lalit Arvind Balaji	Last Updated By:	Lalit Arvind Balaji
Date Created:	11/19/23	Date Last Updated:	11/19/23

Actor:	Alert Subscribed Users
Description:	Alerts are sent to Security team members, and management authorities who have enabled notification channels and provided details through their chosen notification channel.
Preconditions:	• Authority has enabled notification • Authority has provided correct details during sign up • Encroachment has been detected by drones
Postconditions:	• Alert notification has been sent to authority through their preferred notification channel
Primary Pathway:	• Encroachment has been detected by drones and alerted from Video Processing API • Details about the encroachment are obtained: Severity, Location and type of encroachment from the Drone Firmware API • Alerts with all details are sent through SMS/Email to all registered users
Alternate Pathways:	• No notification is sent when no encroachment is detected
Exception Pathways:	Alert was not sent to registered users due to network issue

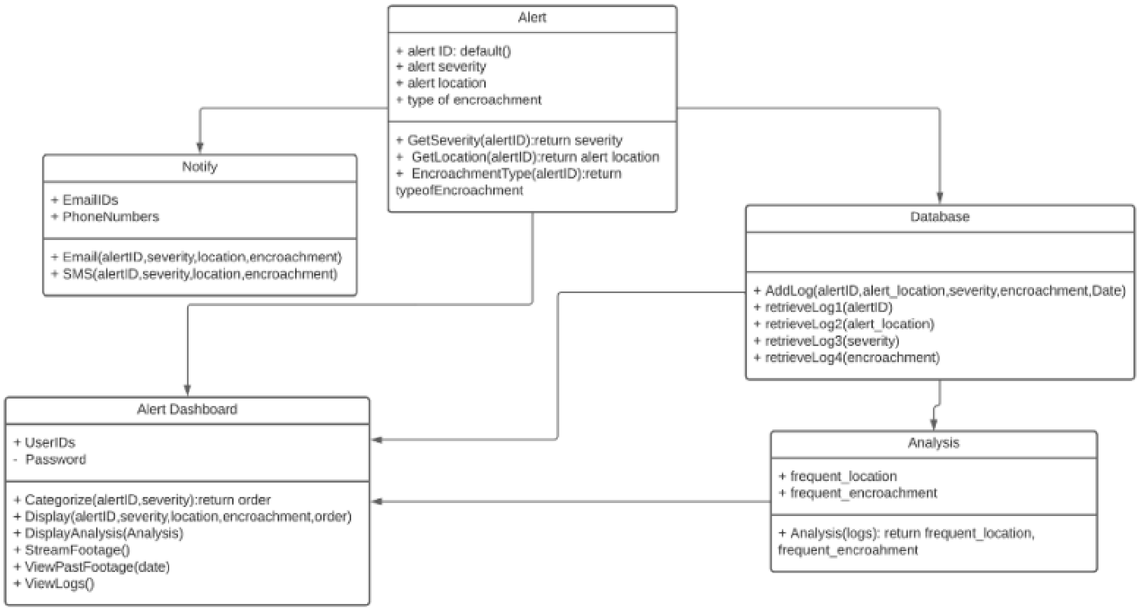
4.6.3.3. Use Case3: Alert Analysis

Use Case ID:	403		
Use Case Name:	Analysis		
Traceability:			
Created By:	Lalit Arvind Balaji	Last Updated By:	Lalit Arvind Balaji
Date Created:	11/19/23	Date Last Updated:	11/19/23

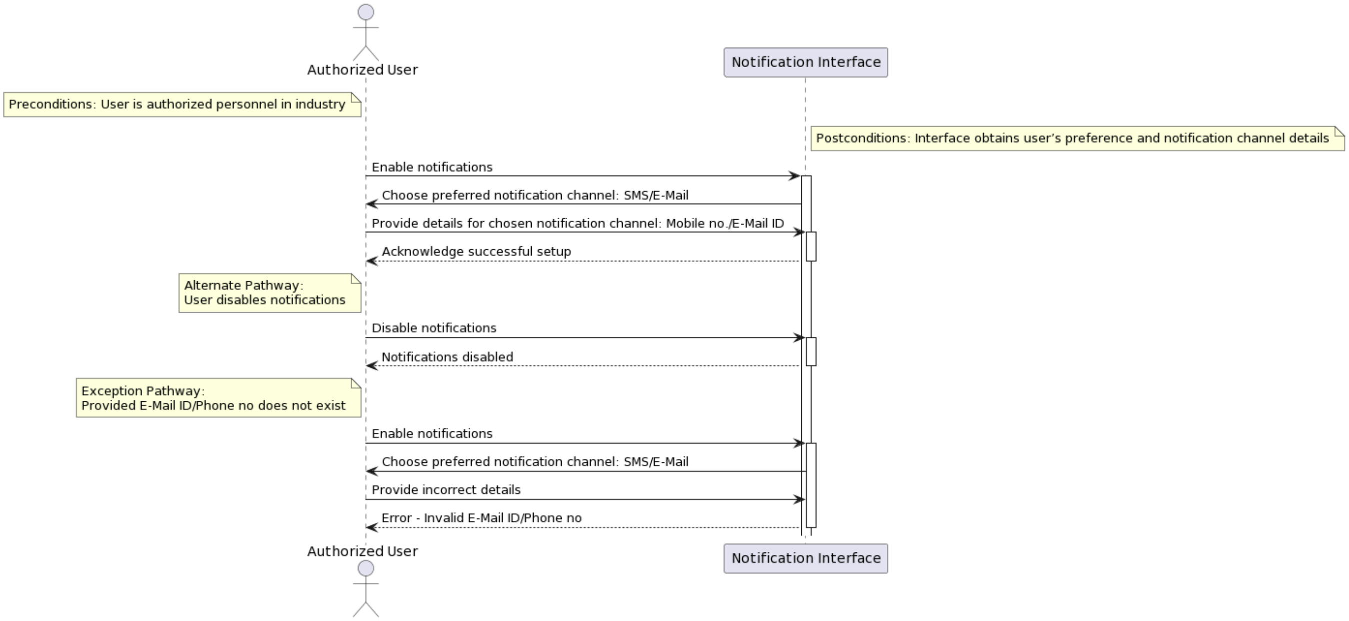
Actor:	Authorized Admin User
Description:	Retrieves logs from database and makes analysis on vulnerabilities on current security system, the location where security needs to be tightened etc.

Preconditions:	• The database contains previous records of encroachment
Postconditions:	• An analysis on past encroachment records
Primary Pathway:	• Retrieves logs from the database • Algorithm extracts prominent common factors in encroachment logs • Analysis is consolidated and sent to user dashboard for the management to take appropriate decisions
Alternate Pathways:	• None
Exception Pathways:	• No previous records leads to no analysis made

4.6.4. Design Element Class Diagram

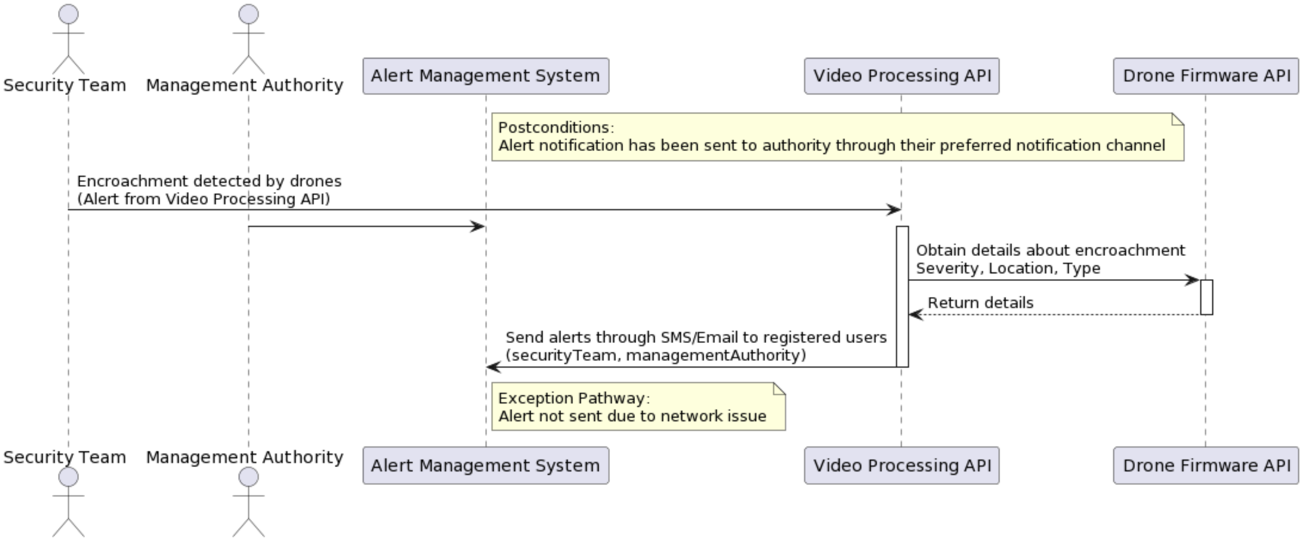

Alert
+ alert ID: default()
+ alert severity
+ alert location
+ type of encroachment
+ GetSeverity(alertID):return severity
+ GetLocation(alertID):return alert location
+ EncroachmentType(alertID):return typeofEncroachment
Notify
+ EmailIDs
+ PhoneNumbers
+ Email(alertID,severity,location,encroachment)
+ SMS(alertID,severity,location,encroachment)
Database
+ AddLog(alertID,alert_location,severity,encroachment,Date)
+ retrieveLog1(alertID)
+ retrieveLog2(alert_location)
+ retrieveLog3(severity)
+ retrieveLog4(encroachment)
Alert Dashboard
+ UserIDs
- Password
+ Categorize(alertID,severity):return order
+ Display(alertID,severity,location,encroachment,order)
+ DisplayAnalysis(Analysis)
+ StreamFootage()
+ ViewPastFootage(date)
+ ViewLogs()
Analysis
+ frequent_location
+ frequent_encroachment
+ Analysis(logs): return frequent_location, frequent_encroahment

4.6.4.1. Sequence Diagram 1


Authorized User
Notification Interface
Preconditions: User is authorized personnel in industry
Postconditions: Interface obtains user's preference and notification channel details
Enable notifications
Choose preferred notification channel: SMS/E-Mail
Provide details for chosen notification channel: Mobile no./E-Mail ID
Acknowledge successful setup
Alternate Pathway:
User disables notifications
Disable notifications
Notifications disabled
Exception Pathway:
Provided E-Mail ID/Phone no does not exist
Enable notifications
Choose preferred notification channel: SMS/E-Mail
Provide incorrect details
Error - Invalid E-Mail ID/Phone no
Authorized User
Notification Interface

- The diagram depicts the interaction between an authorized user and a notification interface.
- The user can enable or disable notifications.
- If the user enables notifications, they must choose a notification channel and provide the necessary details.
- If the user disables notifications, the notifications will be disabled.
- If the user provides incorrect details, the notification interface will display an error message.

4.6.4.2. Sequence Diagram 2


Security Team
Management Authority
Alert Management System
Video Processing API
Drone Firmware API
Postconditions:
Alert notification has been sent to authority through their preferred notification channel
Encroachment detected by drones
(Alert from Video Processing API)
Obtain details about encroachment
Severity, Location, Type
Return details
Send alerts through SMS/Email to registered users
(securityTeam, managementAuthority)
Exception Pathway:
Alert not sent due to network issue
Security Team
Management Authority
Alert Management System
Video Processing API
Drone Firmware API

- The diagram shows how the security team management authority uses the video processing app and drone firmware app to detect and respond to encroachments.
- When the video processing app detects encroachment, it sends an alert to the security team management authority.
- The security team management authority then obtains details about the encroachment and sends an alert to registered users.

5. Risks Addressed

5.1. Overview

- Our project poses several major hazards that must be carefully considered.
- Drone performance in bad weather is the major priority, with a prototype assuring dependability.
- Another problem is correctly identifying recognized items in an industrial setting without false alarms, which is solved by dynamic threshold modifications.
- Data caching and a retention strategy are used to manage storage capacity risk.
- Encryption and secure protocols ensure system integrity and data safety in cyberspace.
- Team members are working hard to address each risk to the system's efficacy and dependability in various operational circumstances.

5.2. Drone Performance Degradation or Getting Lost

- **Description of the risk**
- The drone's performance may decline in adverse weather conditions, leading to uncontrollable flight or deviation from its course.
- Severe weather, such as hurricanes or snowstorms, poses a significant threat to invasion detection efficiency.
- **Significance**
- This risk directly impacts the system's essential functionality, making it a top priority concern.
- **Description of how the prototype mitigated or avoided the risk**
- Our prototype is being created and intensively tested in inclement weather.
- Flight stability, sensor precision, and communication dependability are all tested.
- Iterative adjustments will be made to improve drone performance in difficult scenarios.
- Data analysis insights will inform prototype improvements that are in line with performance goals.
- Gained knowledge and improvements will be effortlessly implemented into production drones, lowering the danger of degraded surveillance during severe weather.

5.3. Cybersecurity Vulnerabilities

- **Description of the risk**
- The danger comprises potential cybersecurity weaknesses in the system, which might expose drone operations, data, and communication channels to malicious activity.
- **Significance**
- The potential compromise of sensitive data, disruption of drone performance, or unauthorized access to the system, all of which pose concerns to overall security and privacy, is significant.
- **Description of how the prototype mitigated or avoided the risk**
- This danger is being actively addressed by dedicated cybersecurity specialists, including Priya Sharma.
- To protect data transmission, the prototype includes encryption methods and secure connection protocols.
- Cybersecurity audits and updates will be performed on a regular basis to identify and patch potential vulnerabilities, maintaining a strong defense against cyber threats.

5.4. Drone Identifying Known Entity as Encroachment

- **Description of the risk**
- Drones may mistakenly identify entities known as encroachers, such as industrial staff, resulting in false alarms.

- False alarms can cause confusion, making it difficult for the security team to respond efficiently.
- **Significance**
- This danger has a direct impact on the system's trustworthiness and the responsiveness of the security team.
- **Description of how the prototype mitigated or avoided the risk**
- Based on environmental circumstances and the intensity of the incursion, the system will dynamically alter encroachment detection limits.
- The goal of this dynamic modification is to eliminate false alarms and improve the system's ability to discern between genuine alerts and erroneous signals.

6. Prototype

6.1. Overview of the Prototype

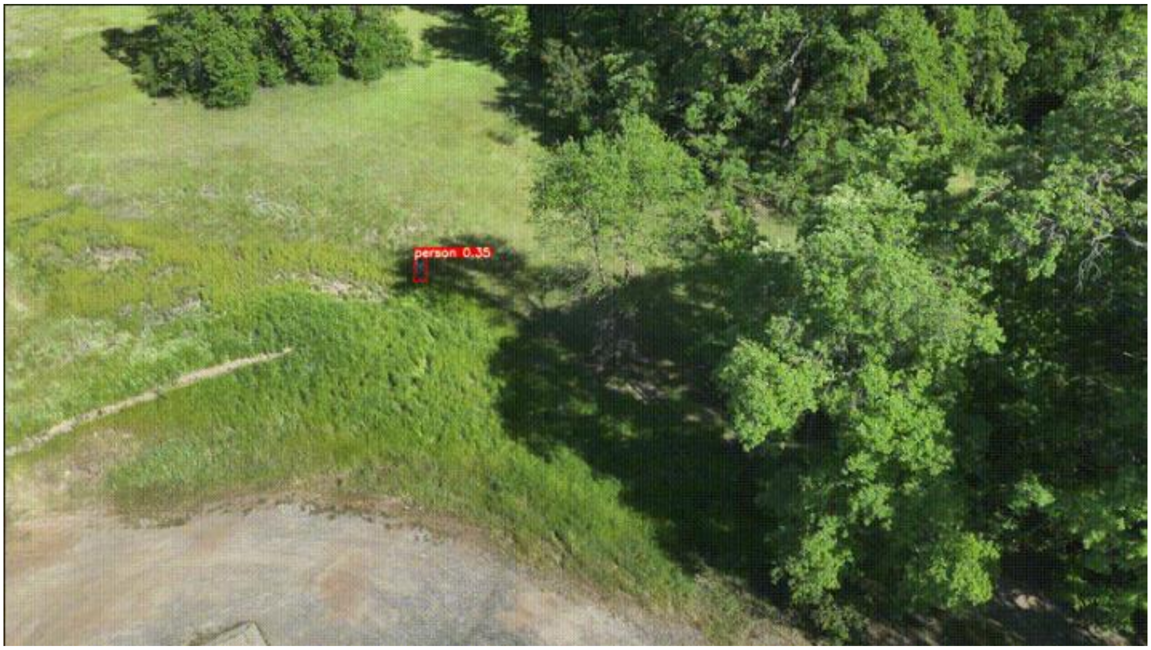
- This prototype presents a drone-based encroachment detection system.
- The system integrates drone programming, the Webots simulator for realistic testing, and the YOLO (You Only Look Once) algorithm for real-time object detection.
- Its primary aim is to identify unauthorized or unexpected entities in a designated area and initiate appropriate alerting mechanisms.

6.2. Major Functions of the Prototype

- **Real-time Encroachment Detection** - Utilizes the YOLO algorithm for detecting encroachments in the monitored area.
- **Alert Generation and Notification** - Sends notifications/alerts when an encroachment is detected.
- **Scheduled Drone-Based Monitoring** - Automated scheduling for regular surveillance

6.3. Major Function 1: Real-time Encroachment Detection

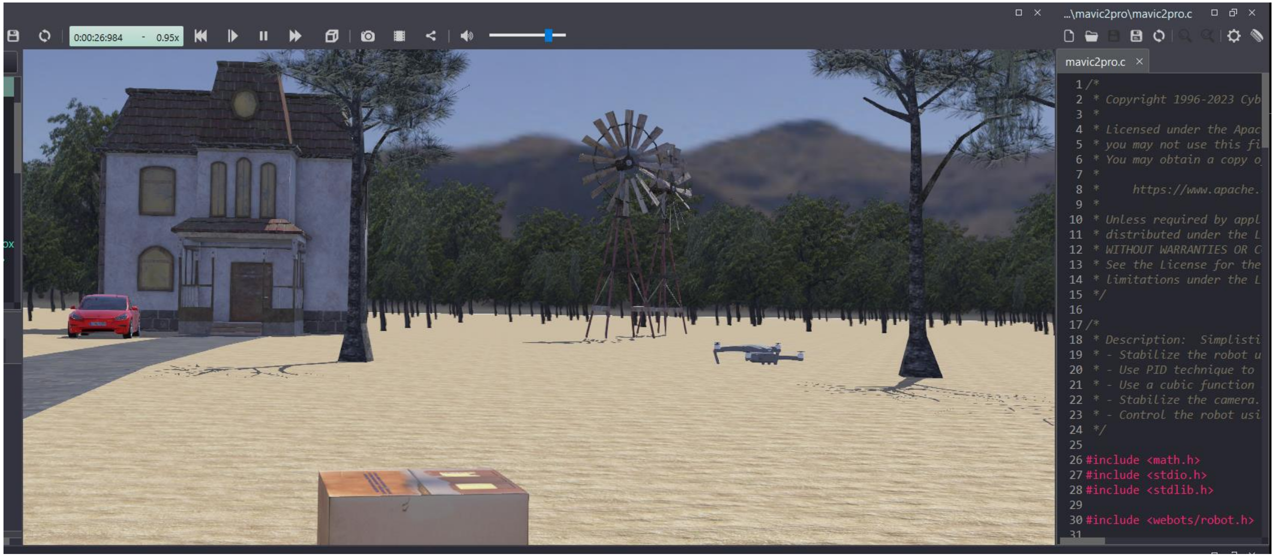
- **Description of the function**
- The drone uses the YOLO algorithm to analyze the surveillance area in real-time.
- It follows a predefined flight path, capturing images and feeding them into the YOLO algorithm for object identification and encroachment detection.
- **Screenshots showing the function in use**



- **How these screenshots show prototype has mitigated or avoided the risk**
- Illustrates the system's ability to accurately identify potential encroachments in real-time, demonstrating its effectiveness in risk mitigation.

6.4. Major Function 2: Scheduled Drone-Based Monitoring

- **Description of the function**
- This function allows for the drone to be programmed for regular, automated surveillance missions.
- The drone can be scheduled to patrol at specific times, covering predefined areas, ensuring continuous monitoring without manual intervention.
- **Screenshots showing the function in use**



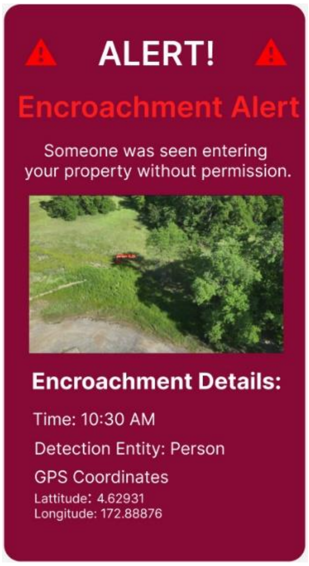
```
/*
 * Copyright 1996-2023 Cyb
 *
 * Licensed under the Apac
 * you may not use this fi
 * You may obtain a copy o
 *
 *     https://www.apache.
 *
 * Unless required by appl
 * distributed under the L
 * WITHOUT WARRANTIES OR C
 * See the License for the
 * limitations under the L
 */

/*
 * Description:  Simplisti
 * - Stabilize the robot u
 * - Use PID technique to
 * - Use a cubic function
 * - Stabilize the camera.
 * - Control the robot usi
 */

#include <math.h>
#include <stdio.h>
#include <stdlib.h>

#include <webots/robot.h>
```

- **How these screenshots show prototype has mitigated or avoided the risk**
- Illustrates how regular, automated monitoring ensures consistent surveillance, reducing the risk of undetected encroachments.

6.5. Major Function 3: Alert Generation and Notification

- **Description of the function**
- Triggers an alert protocol upon detecting an encroachment. Notifications are sent to relevant authorities with details like location, time, and nature of the encroachment.
- **Screenshots showing the function in use**

Prototype



⚠ ALERT! ⚠

Encroachment Alert

Someone was seen entering your property without permission.

Encroachment Details:

Time: 10:30 AM

Detection Entity: Person

GPS Coordinates

Lattitude: 4.62931
Longitude: 172.88876

- **How these screenshots show prototype has mitigated or avoided the risk**
- Demonstrates the system's responsiveness and efficiency in alerting, enabling quick response to potential threats.

7. Conclusion

7.1. Real time Encroachment Detection

7.1.1. Model Training for Encroachment Detection

- Screenshots of a grassy area with trees show the system's precision in identifying things in real time, proving its efficacy in risk prevention.
- The prototype successfully employs the YOLO algorithm for real-time intrusion detection, demonstrating its capacity to recognize possible hazards quickly.
- The incorporation of the YOLO algorithm ensures that the system can scan and analyze photos effectively during drone monitoring, reducing the danger of delayed or inaccurate invasion detection.

7.1.2. Model Integration with Drone

- The prototype's main feature of real-time intrusion detection is critical in improving the overall security of the system.
- The real-time analysis of the YOLO algorithm helps to the system's proactive nature, lowering the danger of delayed response to possible threats.
- The screenshots demonstrate how the prototype successfully mitigated the risk associated with delayed or inaccurate intrusion detection.

7.2. Alert Generation and Notification

7.2.1. Detail Item 1 Description

- The prototype includes an alert creation and notification mechanism that notifies relevant authorities as soon as an encroachment is detected.
- Screenshots of the alert creation process demonstrate the system's quickness and efficiency in contacting authorities, hence improving the overall security protocol.
- The alerting mechanism reduces the danger of delayed communication by allowing for prompt response to possible threats and reducing the effect of security incidents.

7.2.2. Detail Item 2 Description

- The Alert Generation and Notification function makes a substantial contribution to the risk mitigation strategy of the prototype.
- The capacity of the system to provide comprehensive information about the invasion, such as location and timing, improves situational awareness and aids in effective decision-making.
- The screenshots mentioned above provide visual evidence that the prototype efficiently handles the risk associated with inadequate or delayed alerting measures.

7.3. Items for future consideration

- Improve the drone's adaptability to a variety of environmental situations, such as severe temperatures, changing light conditions, and difficult terrain.
- To eliminate false positives, refine machine learning algorithms such as the YOLO algorithm for more accurate and context-aware object detection.
- Consider incorporating additional sensors, such as thermal cameras or advanced LiDAR systems, to supplement visual data and improve danger detection.
- Examine the viability of coordinating numerous drones in a fleet to improve observation and reaction capabilities.

- Examine the system's scalability for wider areas or many sites, taking into account communication, data processing, and system responsiveness concerns.
- Put in place strong cybersecurity measures to secure drone operations, data transmission, and storage systems against cyber threats and assaults.
- Optimize the drone's operating interface.

Appendix A: Credit Sheet

Team Member Name	Contributions
Aditya Pant	Detailed Design Conclusion
Ameya Shahu	Requirements Architecture Detail Design Prototype Conclusion
Lalit Arvind Balaji	Problem Requirements Detailed Design – Alert Management System
Pravalika Mukkiri	Risks Addressed Detailed Design Conclusion