SRS_mihawk.docx

by Turnitin LLC

Submission date: 29-Sep-2024 04:16PM (UTC+0800)

Submission ID: 2468691014

File name: SRS_mihawk.docx (47.57K)

Word count: 2483

Character count: 15201

The purpose of the Mihawk project is to address the limitations of traditional surveillance systems, which often suffer from restricted coverage, delayed threat detection, and increase false alarms. By introducing an manual driven drone-based surveillance system, we aim to enhance the efficiency and effectiveness of security operations. Also the integration of blockchain technology ensures the integrity and security of the data collected, making it tamper-proof and readily accessible for analysis. The ultimate goal is to provide **law enforcement agencies**, **security personnel**, and **organizations** with a tool that offers improved surveillance capabilities, leading to quicker response times, less false alrarms and better protection of assets and people.

The aim of the Mihawk project is to create a cutting-edge drone surveillance system that addresses the challenges of modern security operations. The system will consist of several integrated modules, each designed to enhance surveillance efficiency and reliability. The manual flight control module enables users to navigate drones along the defined routes for precise surveillance coverage. The real-time video streaming module provides continuous live surveillance, allowing users to monitor activities as they happen. The advanced threat detection module will utilize sophisticated algorithms to identify potential security risks and respond promptly. Blockchain technology will be integrated for secure and transparent data storage, ensuring the integrity and confidentiality of flagged surveillance data. Additionally, user management features will allow administrators to control access and permissions, while customizable surveillance settings will provide flexibility in managing different monitoring scenarios. Tools for data analysis and reporting will enable users to gain insights from surveillance data and optimize security operations. This platform aims to offer a scalable, efficient, and comprehensive solution that enhances situational awareness and streamlines operational workflows in various security environments.

FE-2: Develop object detection algorithms to identify and classify objects of interest in the surveillance feed.

FE-3: Design threat identification algorithms to detect and categorize security threats based on detected objects or behaviors.

FE-4: Develop algorithms for crowd behavior analysis to detect suspicious activities in crowded areas.

FE-5: Customizable threat detection settings for specific security requirements and environments.

FE-1: Real-time alerts and notifications for security incidents or unusual activities.

FE-2: Priority-based alert categorization and escalation procedures for timely response.
FE-3: Integration with existing security systems for seamless alert management and coordination.
FE-1: Set up and configure a blockchain network for secure and tamper-proof storage of flagged surveillance data.
FE-2: Implement data encryption mechanisms to ensure the security and integrity to store surveillance data.
FE-1: User authentication and authorization mechanisms for secure access to the system.
FE-2: Role-based access control to restrict functionalities based on user roles and permissions.
FE-3: Enable user to edit profile information (password).
FE-4: Implement a secure password reset functionality where user can retrieve forgotten passwords through email verification or security questions.
FE-5: Audit trail functionality to track user activities and changes made to the system.
FE-6: Implement 2-Factor Authentication for better security.
FE-1: Design a user-friendly web interface for operators to monitor and control the surveillance system.
FE-2: Develop features to display live surveillance feed and playback recorded footage on the web application.
FE-3: Implement real-time status indicators for the network, battery levels, and connection status.

FE-1: Interactive maps and visualization tools for displaying drone locations and surveillance data.
FE-2: Provide search functionality for users to selectively view specific types of surveillance data or events.
FE-3: Implement zoom and pan functionality for detailed exploration of surveillance data and maps.
FE-4: Integrate real-time weather data overlays to visualize weather conditions and their impact on surveillance operations.
FE-5: Implement dynamic data filtering options to enable users to filter surveillance data by criteria like time, location, or activity type for focused analysis and monitoring.
FE-1: Develop mechanisms to efficiently store captured images or videos on the blockchain.
FE-2: Implement functionality to retrieve stored surveillance data from the blockchain for analysis or playback.
FE-3: Implement data compression techniques to reduce storage requirements without compromising quality.
FE-4: Integrate data lifecycle management policies to automatically archive or delete outdated surveillance data.
FE-1: Reporting functionalities to generate detailed reports on surveillance activities and incidents.
FE-2: Advanced analytics tools for deep dive analysis of surveillance data and trends.
FE-3: Export functionalities to save reports and analytics data in various formats for sharing and archival purposes.
FE-1: Design a user-friendly interface for administrators to monitor and manage the surveillance system efficiently.
FE-2: Enable administrators to manage user accounts, including creating, editing, and deleting user profiles, as well as assigning roles and permissions.

FE-3: Provide a comprehensive overview of system status, including drone fleet health, surveillance coverage, and alert summaries.
FE-4: Allow administrators to configure system settings, such as alert thresholds, and data retention policies.
FE-5: Design reporting tools to generate insights on system performance, alert trends, and operational metrics for analysis.

FE-1: Configure Raspberry Pi to stream real-time video from the drone's camera using the Real-Time Streaming Protocol (RTSP). This allows for live monitoring of surveillance feeds over the network.
FE-2: Enable integration of various camera modules and sensors with the Raspberry Pi, providing flexibility for different surveillance and data collection needs.
FE-3: Implement data compression techniques on the Raspberry Pi to ensure efficient transmission of high-quality video streams with minimal bandwidth usage.
FE-4: Integrate power management features to monitor and optimize the Raspberry Pi's energy consumption, ensuring it operates efficiently during drone flights.
FE-5: Use Raspberry Pi for real-time processing of surveillance data (e.g., object detection) to reduce latency before data is sent to the central system, improving threat detection capabilities.

.

The remainder of SRS document provides a detailed specification of Mihawk, outlining its functional and non-functional requirements. The document is organized into distinct modules, each focusing on a specific aspect of the software's functionality. This is followed by a section on non-functional requirements, encompassing aspects such as performance, security, usability, and maintainability. By following this modular structure, the document ensures clarity, organization, and ease of understanding for all stakeholders involved in the development process

Mihawk is an **innovative addition** to the landscape of surveillance systems, representing a significant advancement over traditional methods. Unlike conventional surveillance that often relies on stationary cameras or manual monitoring, Mihawk utilizes **advanced drone technology** to provide **dynamic and flexible coverage**. This evolution from static to mobile surveillance allows for **real-time monitoring** of expansive areas, addressing the limitations of fixed systems.

Figure 2.2.1 shows the user classes and characteristics for Mihawk : Drone Surveillance System

Figure 2.2.1 user classes and characteristic for Mihawk

User class	**Description**
Security Personnel	The primary users responsible for overseeing surveillance operations. They require real-time access to video feeds, threat alerts, and control over the drone's manual operations. The interface must be intuitive and provide easy access to critical functionalities like drone navigation, surveillance settings, and incident reporting.
System Administrators	Technical users tasked with system configuration, maintenance, and ensuring data integrity. They require access to system logs, user management tools, blockchain integration settings, and security protocols. A robust interface for configuring surveillance zones and managing drone flight paths is essential.
Law Enforcement Agencies	Users who leverage the system for public safety and crime prevention. They need features like evidence gathering, suspect tracking, and secure access to video archives. The system must provide reliable access to real-time feeds and historical data stored on the blockchain for analysis and legal purposes.
General Users	Property owners, event organizers, or individuals responsible for monitoring specific areas. They need a simplified interface that offers easy control over pre-defined surveillance routes, real-time video monitoring, and customizable alert settings, without requiring technical expertise.

OE-1: Mihawk is primarily designed to operate on desktop and laptop computers, as well as tablets with sufficient processing power, memory, and graphics capabilities to handle real-time video streaming and drone control interfaces.
OE-2: The system shall be compatible with the following operating systems: Windows 10 and 11, macOS 10.15 (Catalina) and later, and Linux distributions that support modern web technologies.
OE-3: Mihawk shall be accessible through modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari, ensuring cross-platform usability.
OE-4: The application is designed to function optimally with a minimum internet speed of 10 Mbps, to ensure smooth real-time video streaming and responsive drone control.
OE-5: While Mihawk can be used globally, the primary target users include security personnel, law enforcement agencies, and organizations in regions with strong infrastructure and internet connectivity.

CON-1: The system shall use the DroneKit SDK for drone control and navigation, ensuring flexibility and reliability in manual flight operations.
CON-2: The application must utilize blockchain technology (e.g., Ethereum or Hyperledger) for secure and tamper-proof data storage, ensuring the integrity of surveillance data.
CON-3: The application shall be developed using the Node.Js for the backend to ensure cross-platform compatibility and ease of deployment.
CON-4: All real-time video feeds and data must be securely transmitted and stored using AES-256 encryption to meet data security and privacy standards.
CON-5: The system must integrate with PostgreSQL or MySQL for managing user data and system logs, as these databases are highly scalable and support high volumes of data processing.

The first technique used in identifying requirements for the system is use cases. Mihawk has a lot of user involvement, and the user requirements are identified using this technique. The use case diagram representing the requirements of each user is given below. Detailed use cases are also present in the following sections.

Figure 3.1.1.1 shows the major use cases of the actor User , Figure 3.1.1.2 shows the major use cases of the actor Admin and Figure 3.1.1.3 shows the major use cases of the actor Security Personnel.

Figure 3.1.1.1 use cases of actor User

Figure 3.1.1.2 use cases of actor Admin

Figure 3.1.1.3 use cases of actor Security Personnelsy

Users can create an account on the Mihawk system, enabling them to access and utilize various platform features such as drone control, threat detection, and video monitoring. This process involves providing essential details like email and password, allowing users to securely log in and interact with the system.

This section specifies the non-functional requirements that the Mihawk Drone Surveillance System must meet to ensure a robust, efficient, and secure platform. These requirements include Reliability, Usability, Performance, and Security.

The Mihawk system must operate with minimal failures to ensure continuous surveillance operations. Reliability will be measured by the Mean Time Between Failures (MTBF). The system should have a downtime of no more than 5% over the course of a year. Additionally, error detection mechanisms shall be implemented to ensure any system failure is detected and corrected within 90 seconds.
REL-1: The system shall achieve a minimum of 95% uptime annually.
REL-2: In case of system failure, automated recovery procedures should restore operations within 1 minute and 30 seconds.
REL-3: Redundancy protocols should be in place to minimize the impact of hardware or software malfunctions.

The Mihawk system must provide an intuitive interface to ensure ease of use for operators and administrators. Users should be able to interact with the system effectively with minimal training. The following requirements address the system's usability:
USE-1: 90% of users shall be able to perform core functions such as drone navigation and real-time monitoring within 10 minutes of using the interface.
USE-2: The threat detection alerts shall be displayed in an easily understandable format with clear instructions for responding to threats.

USE-3: Users shall be able to view, manage, and export surveillance data within 3 clicks or less.
USE-4: Operators shall be able to switch between drone control and monitoring functions without noticeable delays or interruptions in workflow.

.

Performance requirements ensure that Mihawk operates efficiently in real-time environments. The system's responsiveness is critical for effective security operations.
PER-1: The drone's manual control inputs shall result in changes to flight movements within 1500 milliseconds.
PER-2: Video feeds shall have a latency of no more than 10 seconds during real-time monitoring.
PER-3: System-generated threat detection alerts should appear within 3 seconds of identifying suspicious activity.
PER-4: The system shall process and store flagged surveillance data on the blockchain within 15 seconds of being flagged by the user.
PER-5: Location-based data, such as drone position, should be updated on the interface every 5 second.

Security is paramount in the Mihawk Drone Surveillance System to ensure that all data and operations are protected from unauthorized access or tampering. The following requirements are aimed at safeguarding the system and its data.
SEC-1: All user login credentials and surveillance data shall be encrypted using AES-256 encryption.
SEC-2: Access to drone control and data retrieval operations shall be role-based and require two-factor authentication (2FA).
SEC-3: All communications between the drone, blockchain, and backend system must be conducted using secure HTTPS protocols.
SEC-4: The system shall automatically log out users after 15 minutes of inactivity to prevent unauthorized access.

The Mihawk Drone Surveillance System shall feature a user-friendly interface that supports ease of navigation and control. The interface will follow standard guidelines for accessibility and user experience.
UI-1: The color scheme shall use high-contrast colors to enhance visibility in various lighting conditions.
UI-2: The main dashboard shall prominently display critical functions such as real-time feed, drone controls, threat detection, and alerts.
UI-3: Icons for drone navigation, live feed, and alert management shall be large and clearly labeled to facilitate quick actions.
UI-4: An error message system shall be implemented to guide users through troubleshooting steps in case of failures.

UI-5: The user interface shall provide shortcut keys for frequently used actions such as zooming in on video feeds or switching drone views.

The Mihawk system must communicate seamlessly with other software components such as databases, blockchain networks, and third-party APIs.
SI-1: Mihawk shall interface with the DroneKit API to control drone navigation and obtain flight telemetry data.
SI-2: The system shall use a secure API to communicate with Ethereum-based blockchain platforms for storing flagged surveillance data.
SI-3: Mihawk shall integrate with PostgreSQL and MySQL databases for storing user information, logs, and other relevant system data.
SI-4: The system shall integrate with cloud storage services to archive large volumes of video footage and log files for future access.

The Mihawk system must interact with hardware components such as drones, cameras, and storage devices to fulfill its surveillance functions.
HI-1: The Mihawk system shall interface with drones via wireless communication protocols or via wired connection to receive video feed and control drone navigation.
HI-2: The system shall support USB or cloud storage devices for the export and backup of surveillance footage.
HI-3: The system shall interact with on-board drone cameras and sensors for data collection, including flight telemetry and battery levels.
Mihawk must ensure secure and reliable communication between its components, including drones, users, and back-end services.
CI-1: All communication between the drone and Mihawk's backend system shall be secured using HTTPS.
CI-2: The system shall support real-time notifications using web sockets to alert operators of detected threats or system status changes.
CI-3: The system shall utilize secure email and SMS protocols for sending notifications to users when critical events are detected.

SRS_mihawk.docx

ORIGINALITY REPORT

5%	2%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	2%
2	Submitted to Colorado Technical University Online Student Paper	1%
3	Submitted to University of Maryland, Global Campus Student Paper	<1%
4	Submitted to University of Alabama Student Paper	<1%
5	Submitted to Manipal University Student Paper	<1%
6	Daniel Birman, Kenneth J. Yang, Steven J. West, Bill Karsh, Yoni Browning, Joshua H. Siegle, Nicholas A. Steinmetz. "Pinpoint: trajectory planning for multi-probe electrophysiology and injections in an interactive web-based 3D environment", Cold Spring Harbor Laboratory, 2023 Publication	<1%

7	www.shure.com Internet Source	<1%
8	Ervin Varga. "Practical Data Science with Python 3", Springer Science and Business Media LLC, 2019 Publication	<1%
9	Takako Nakatani, Kazuaki Sato, Osamu Shigo. "Traverser: A method and a tool to extract user’s traverses within web systems from ontology", Intelligent Decision Technologies, 2023 Publication	<1%
10	scholarworks.sjsu.edu Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		