****

**COMSATS University Islamabad (CUI)**

Software Requirement Specification  
(SRS DOCUMENT)

for

**Mihawk**

Version 1.0

***By***

**Muhammad Usman Malik CIIT/FA21-BCS-072/ISB**

**Muhammad Hozefa Rauf CIIT/FA21-BCS-057/ISB**

**Hammad Ur Rehman CIIT/FA21-BCS-055/ISB**

***Supervisor*Mr. Qasim Malik**

*Bachelor of Science in Computer Science (2021-2025)*

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Supervised by

Mr. Qasim Malik

Signature**\_\_\_\_\_\_\_\_\_\_\_**

**Table of Contents**

[1. Introduction 6](#_Toc177824531)

[1.1 Purpose 6](#_Toc177824532)

[1.2 Scope 6](#_Toc177824533)

[1.3 Modules 6](#_Toc177824534)

[1.31 Module 1: Drone Control 6](#_Toc177824535)

[1.32 Module 2: Surveillance and Threat Detection 6](#_Toc177824536)

[1.33 Module 3: Alert Management 7](#_Toc177824537)

[1.34 Module 4: Blockchain Integration 7](#_Toc177824538)

[1.35 Module 5: User Management 7](#_Toc177824539)

[1.36 Module 6: Surveillance Monitoring Interface 7](#_Toc177824540)

[1.37 Module 7: Interactive Mapping and Location Visualization 8](#_Toc177824541)

[1.38 Module 8: Data Handling 8](#_Toc177824542)

[1.39 Module 9: Reporting and Analysis 8](#_Toc177824543)

[1.310 Module 10: Admin Dashboard 8](#_Toc177824544)

[1.311 Module 11: Raspberry Pi Integration 8](#_Toc177824545)

[1.4 Overview 8](#_Toc177824546)

[2. Overall Description 9](#_Toc177824547)

[2.1 Product Perspective 9](#_Toc177824548)

[2.2 User classes and characteristics 9](#_Toc177824549)

[2.3 Operating Environment 9](#_Toc177824550)

[2.4 Design and Implementation Constraints 10](#_Toc177824551)

[3. Requirement Identifying Technique 10](#_Toc177824552)

[3.1 Use Case 10](#_Toc177824553)

[3.1.1 Use Case Diagram 10](#_Toc177824554)

[3.2 Detailed Use Case 12](#_Toc177824555)

[3.2.1 UC-1: Register Account 12](#_Toc177824556)

[3.2.2 UC-2: User Login 13](#_Toc177824557)

[3.2.3 UC-3: Password Recovery 13](#_Toc177824558)

[3.2.4 UC-4: Logout Account 14](#_Toc177824559)

[3.2.4 UC-4: Initialize Drone for Surveillance 15](#_Toc177824560)

[3.2.5 UC-5: Drone Control 16](#_Toc177824561)

[3.2.6 UC-6: Real-Time Video Feed 16](#_Toc177824562)

[3.2.7 UC-7: Detect Threats in Surveillance Area 17](#_Toc177824563)

[3.2.8 UC-8: Send Alert Notification 17](#_Toc177824564)

[3.2.9 UC-9: Store Flagged Data on Blockchain 17](#_Toc177824565)

[3.2.10 UC-10: Retrieve Flagged Data from Blockchain 18](#_Toc177824566)

[3.2.11 UC-11: Conduct Pre-Flight System Check 18](#_Toc177824567)

[3.2.12 UC-12: Monitor Drone Battery Level 18](#_Toc177824568)

[3.2.13 UC-13: Assign Drone Operator Role 19](#_Toc177824569)

[3.2.14 UC-14: Remove User Account 19](#_Toc177824570)

[3.2.15 UC-15: User Activity Log Monitoring 20](#_Toc177824571)

[3.2.16 UC-16: User Role Modification 20](#_Toc177824572)

[3.2.17 UC-17: Send System Notifications to Users 20](#_Toc177824573)

[3.2.18 UC-18: Alert User on Drone Malfunction 21](#_Toc177824574)

[4. Functional Requirements 21](#_Toc177824575)

[4.1 Use Case Name: Register Account 21](#_Toc177824576)

[4.1.1 FR-1: RegisterAccount.UserRegistration 21](#_Toc177824577)

[4.1.2 FR-2: RegisterAccount.FieldValidation 22](#_Toc177824578)

[4.1.3 FR-3: RegisterAccount.UniqueEmailCheck 22](#_Toc177824579)

[4.1.4 FR-4: RegisterAccount.PasswordSecurity 22](#_Toc177824580)

[4.1.5 FR-5: RegisterAccount.ConfirmationEmail 23](#_Toc177824581)

[4.1.6 FR-5: RegisterAccount.StoreUserData 23](#_Toc177824582)

[4.1.7 FR-7: RegisterAccount.RegistrationErrorHandling 23](#_Toc177824583)

[4.2 Use Case Name: User Login 24](#_Toc177824584)

[4.2.1 FR-8: UserLogin.LoginForm 24](#_Toc177824585)

[4.2.2 FR-9: UserLogin.ValidateLoginDetails 24](#_Toc177824586)

[4.2.3 FR-10: UserLogin.PasswordEncryptionCheck 25](#_Toc177824587)

[4.2.4 FR-11: UserLogin.LoginAttemptsLimit 25](#_Toc177824588)

[4.2.5 FR-12: UserLogin.ErrorHandling 25](#_Toc177824589)

[4.2.6 FR-13: UserLogin.RememberMeOption 26](#_Toc177824590)

[4.2.7 FR-14: UserLogin.ForgotPassword 26](#_Toc177824591)

[4.2.8 FR-15: UserLogin.SessionManagement 26](#_Toc177824592)

[4.3 Use Case Name: Password Recovery 27](#_Toc177824593)

[4.3.1 FR-16: PasswordRecovery.RequestResetLink 27](#_Toc177824594)

[4.3.2 FR-17: PasswordRecovery.EmailVerification 27](#_Toc177824595)

[4.3.3 FR-18: PasswordRecovery.LinkExpiration 28](#_Toc177824596)

[4.4 Use Case Name: LogoutAccount 28](#_Toc177824597)

[4.4.1 FR-19: LogoutAccount.TerminateSession 28](#_Toc177824598)

[4.4.2 FR-20: LogoutAccount.RedirectToLogin 28](#_Toc177824599)

[4.5 Use Case Name: Drone Control 29](#_Toc177824600)

[4.5.1 FR-21: DroneControl.ActivateDrone 29](#_Toc177824601)

[4.5.2 FR-22: DroneControl.RealTimeFeedback 29](#_Toc177824602)

[4.5.3 FR-23: DroneControl.ManualControl 30](#_Toc177824603)

[4.5.4 FR-24: DroneControl.BatteryWarning 30](#_Toc177824604)

[4.5.5 FR-25: DroneControl.AutomaticLanding 30](#_Toc177824605)

[4.6 Use Case Name: Real Time Feed 31](#_Toc177824606)

[4.6.1 FR-26: RealTimeFeed.ActivateFeed 31](#_Toc177824607)

[4.6.2 FR-27: RealTimeFeed.StreamVideo 31](#_Toc177824608)

[4.6.3 FR-28: RealTimeFeed.StreamDroneData 31](#_Toc177824609)

[4.6.4 FR-29: RealTimeFeed.NotifyFeedDisruption 32](#_Toc177824610)

[4.6.5 FR-30: RealTimeFeed.ControlFeedQuality 32](#_Toc177824611)

[4.6.6 FR-31: RealTimeFeed.SaveVideoFeed 33](#_Toc177824612)

[4.7 Use Case Name: Detect Threats in Surveillance Area 33](#_Toc177824613)

[4.7.1 FR-32: ThreatDetection.InitializeSurveillance 33](#_Toc177824614)

[4.7.2 FR-33: ThreatDetection.MonitorArea 33](#_Toc177824615)

[4.7.3 FR-34: ThreatDetection.DetectMovementPatterns 34](#_Toc177824616)

[4.7.4 FR-35: ThreatDetection.ClassifyThreats 34](#_Toc177824617)

[4.7.5 FR-36: ThreatDetection.GenerateAlert 35](#_Toc177824618)

[4.7.6 FR-37: ThreatDetection.RecordThreatData 35](#_Toc177824619)

[4.8 Use Case Name: Send Alert Notification 35](#_Toc177824620)

[4.8.1 FR-38: AlertNotification.ConfigureAlertSettings 35](#_Toc177824621)

[4.8.2 FR-39: AlertNotification.GenerateRealTimeAlert 36](#_Toc177824622)

[4.8.3 FR-40: AlertNotification.MultipleRecipients 36](#_Toc177824623)

[4.8.4 FR-41: AlertNotification.DetailedAlertInformation 36](#_Toc177824624)

[4.8.5 FR-42: AlertNotification.ManageAlertHistory 37](#_Toc177824625)

[4.9 Use Case Name: Store Flagged Data on Blockchain 37](#_Toc177824626)

[4.9.1 FR-43: Blockchain.StoreFlaggedData 37](#_Toc177824627)

[4.9.2 FR-44: Blockchain.VerifyDataIntegrity 38](#_Toc177824628)

[4.9.3 FR-44: Blockchain.EncryptFlaggedData 38](#_Toc177824629)

[4.10 Use Case Name: Retrieve Flagged Data from Blockchain 38](#_Toc177824630)

[4.10.1 FR-45: Blockchain.RetrieveFlaggedData 38](#_Toc177824631)

[4.10.2 FR-46: Blockchain.VerifyUserPermissions 39](#_Toc177824632)

[4.10.3 FR-47: Blockchain.DecryptFlaggedData 39](#_Toc177824633)

[4.11 Use Case Name: Conduct Pre-Flight System Check 39](#_Toc177824634)

[4.11.1 FR-48: PreFlight.CheckBatteryLevel 39](#_Toc177824635)

[4.11.2 FR-49: PreFlight.VerifyCommunication 40](#_Toc177824636)

[4.11.3 FR-50: PreFlight.CheckWeatherConditions 40](#_Toc177824637)

[4.12 Use Case Name: Monitor Drone Battery Level 41](#_Toc177824638)

[4.12.1 FR-51: MonitorBattery.CheckInitialLevel 41](#_Toc177824639)

[4.12.2 FR-52: MonitorBattery.TrackInFlightLevel 41](#_Toc177824640)

[4.12.3 FR-53: MonitorBattery.SendLowBatteryAlert 41](#_Toc177824641)

[4.12.4 FR-54: MonitorBattery.PerformEmergencyLanding 42](#_Toc177824642)

[4.12.5 FR-55: MonitorBattery.LogBatteryUsage 42](#_Toc177824643)

[4.13 Use Case Name: Assign Drone Operator Role 43](#_Toc177824644)

[4.13.1 FR-56: AssignOperator.AssignRole 43](#_Toc177824645)

[4.13.2 FR-57: AssignOperator.SetOperatorPermissions 43](#_Toc177824646)

[4.13.3 FR-58: AssignOperator.RevokeOperatorRole 43](#_Toc177824647)

[4.13.4 FR-59: AssignOperator.NotifyUser 44](#_Toc177824648)

[4.14 Use Case Name: Remove User Account 44](#_Toc177824649)

[4.14.1 FR-60: RemoveAccount.DeleteUserAccount 44](#_Toc177824650)

[4.14.2 FR-61: RemoveAccount.ValidateAdminPermission 44](#_Toc177824651)

[4.14.3 FR-62: RemoveAccount.BackupUserData 45](#_Toc177824652)

[4.14.4 FR-63: RemoveAccount.NotifyUserAndAdmin 45](#_Toc177824653)

[4.14.5 FR-64: RemoveAccount.RevokeAccessImmediately 45](#_Toc177824654)

[4.15 Use Case Name: User Activity Log Monitoring 46](#_Toc177824655)

[4.15.1 FR-65: ActivityLog.TrackUserLogin 46](#_Toc177824656)

[4.15.2 FR-66: ActivityLog.AccessLogsByAdmin 46](#_Toc177824657)

[4.15.3 FR-67: ActivityLog.FlagSuspiciousActivity 47](#_Toc177824658)

[4.15.4 FR-68: ActivityLog.ExportLogs 47](#_Toc177824659)

[4.16 Use Case Name: User Activity Log Monitoring 47](#_Toc177824660)

[4.16.1 FR-69: RoleModification.AssignUserRole 47](#_Toc177824661)

[4.16.2 FR-70: RoleModification.ChangeUserRole 48](#_Toc177824662)

[4.16.3 FR-71: RoleModification.ViewUserRole 48](#_Toc177824663)

[4.17 Use Case Name: Send System Notifications to Users 49](#_Toc177824664)

[4.17.1 FR-72: Notification.SendNotification 49](#_Toc177824665)

[4.17.2 FR-73: Notification.NotificationHistory 49](#_Toc177824666)

[4.17.3 FR-74: Notification.AlertForCriticalEvents 49](#_Toc177824667)

[4.18 Use Case Name: Alert User on Drone Malfunction 50](#_Toc177824668)

[4.18.1 FR-75: Alert.UserDroneMalfunction 50](#_Toc177824669)

[4.18.2 FR-76: Alert.MalfunctionSeverityAssessment 50](#_Toc177824670)

[4.18.3 FR-76: Alert.MalfunctionDetails 50](#_Toc177824671)

[4.18.4 FR-77: Alert.HistoryLog 51](#_Toc177824672)

[5. Non-Functional Requirements 51](#_Toc177824673)

[5.1 Reliability 51](#_Toc177824674)

[5.2 Usability 52](#_Toc177824675)

[5.3 Performance 52](#_Toc177824676)

[5.4 Security 52](#_Toc177824677)

[6. External Interface Requirements 53](#_Toc177824678)

[6.1 User Interfaces Requirements 53](#_Toc177824679)

[6.2 Software interfaces 53](#_Toc177824680)

[6.3 Hardware interfaces 53](#_Toc177824681)

[6.4 Communications interfaces 54](#_Toc177824682)

[7. References 54](#_Toc177824683)

# Introduction

## Purpose

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.

## Scope

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.

## Modules

## Module 1: Drone Control

FE-1: Manual flight control system for drones to navigate on the routes.  
FE-2: Integrate weather data APIs to adjust flight plans based on weather conditions.  
FE-3: Real-time monitoring of drone status and location.  
FE-4: Provide controls for emergency procedures in case of unexpected situations during flight.

## Module 2: Surveillance and Threat Detection

FE-1: Implement real-time video processing algorithms to analyze surveillance footage for potential threats or anomalies.

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.

## Module 3: Alert Management

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.

## Module 4: Blockchain Integration

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.

## Module 5: User Management

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.

## Module 6: Surveillance Monitoring Interface

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.

## Module 7: Interactive Mapping and Location Visualization

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.

## Module 8: Data Handling

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.

## Module 9: Reporting and Analysis

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.

## Module 10: Admin Dashboard

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.

## Module 11: Raspberry Pi Integration

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.

.

## Overview

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

# Overall Description

## Product Perspective

**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.

## User classes and characteristics

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

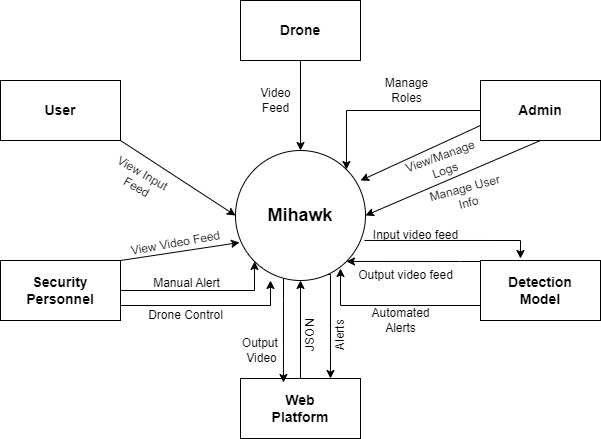


Figure 2.2.1 user classes and characteristic for Mihawk

|  |  |
| --- | --- |
| **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. |

## Operating Environment

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.

## Design and Implementation Constraints

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.

*.*

# Requirement Identifying Technique

## Use Case

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.

### Use Case Diagram

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.

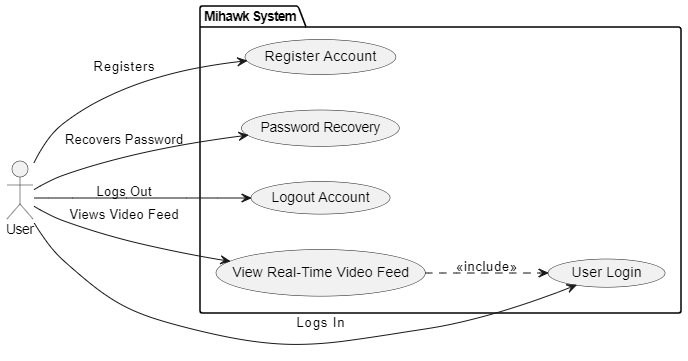


Figure 3.1.1.1 use cases of actor User

A diagram of a company

Description automatically generated

Figure 3.1.1.2 use cases of actor Admin

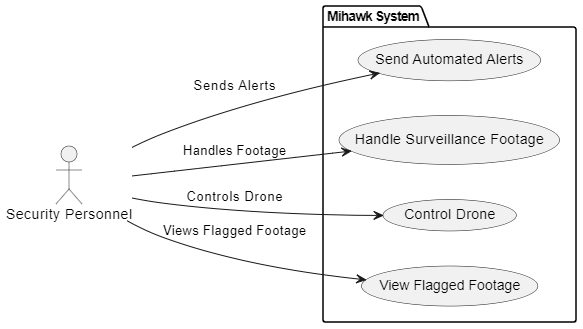


Figure 3.1.1.3 use cases of actor Security Personnelsy

## Detailed Use Case

### UC-1: Register Account

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: User |
| **Trigger:** | The User wants to register an account on the Mihawk system. |
| **Preconditions:** | PRE-1:User must have an active internet connection. |
| **Postconditions:** | POST-1. User is registered and information is stored.  POST-2. User gets a success message upon registration. |
| **Normal Flow:** | 1. User initiates the registration process. 2. User provides the following registration details: 3. Email Address 4. Password 5. Contact Number 6. User clicks on the "Register" button. 7. The system validates the provided information. 8. If the information is valid, the system stores the User's registration details and generates a unique User ID. 9. User receives a confirmation message stating that the account registration was successful. |
| **Alternative Flows:** | 1. After entering the email address, the system detects an incorrect format (e.g., missing '@' or domain name). 2. System prompts the user to re-enter the correct email.   The user corrects the email and proceeds. |
| **Exceptions:** | E1. In the Normal Flow, if any of the mandatory registration fields are missing or improperly filled out:   1. The system prompts the User to complete all the required fields. 2. User complies and adds the necessary information. 3. The registration process continues, and the User's account is registered.   E2. If the User provides unrealistic or invalid values for the registration fields:   1. The system does not proceed with the registration due to network error. 2. User is prompted to provide accurate and valid information. 3. If the User corrects the information, the registration process continues, and the account is registered.   E3. If the User's internet connection is poor or disconnected during the registration process:   1. User is informed that the registration cannot proceed. |
| **Business Rules** | BR-1: Registration requests must be processed within a reasonable timeframe.  BR-2: Users must have a stable and valid internet connection to complete registration. |
| **Assumptions:** | AS-1: Users have the necessary information and details to complete the registration.  AS-2: User-provided information is accurate and valid. |

### UC-2: User Login

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: User |
| **Trigger:** | The User wants to log into the Mihawk system. |
| **Preconditions:** | PRE-1: The User has successfully registered. |
| **Postconditions:** | |  | | --- | | POST-1: The user is logged into the system. |  |  | | --- | |  | |
| **Normal Flow:** | |  | | --- | |  |  |  | | --- | | 1. The user provides login credentials (email and password).  2. The system verifies the credentials.  3. If valid, the user is granted access to the system | |
| **Alternative Flows:** | 1. After entering the correct credentials, the system prompts for a second factor (e.g., a one-time password). 2. User enters the OTP received via email or SMS. 3. If valid, the user is granted access. |
| **Exceptions:** | E1. Incorrect login details:  1. The system notifies the user of the issue.  2. The user is prompted to re-enter valid information. |
| **Business Rules** | BR-1: User must have valid login credentials. |
| **Assumptions:** | AS-1: The user has valid credentials. |

### UC-3: Password Recovery

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: User |
| **Trigger:** | The User wants to reset their password while attempting to log into their account in the Mihawk application. |
| **Preconditions:** | PRE-1: User must have an active internet connection.  PRE-2: User must have a registered account on Mihawk web application.  PRE-3: The User is attempting to log into their account. |
| **Postconditions:** | POST-1: User's password is successfully reset, and they are granted access to their account. |
| **Normal Flow:** | 1. As part of the login process, the User clicks on the "Forgot Password?" link. 2. The system prompts the User to enter their registered email address. 3. User enters their email address and clicks on the "Submit" button. 4. The system verifies the provided email address. 5. If the email address is valid and associated with an existing account, the system generates a password reset link and sends it to the User's email address. 6. User receives an email containing the password reset link and instructions. 7. User clicks on the password reset link in the email. 8. The system opens a password reset page, allowing the User to create a new password. 9. User enters a new password and confirms it. 10. User clicks on the "Reset Password" button. 11. The system validates the new password and updates it in the User's account. 12. If the password reset is successful, the system grants the User access to their account. 13. User is directed to the application's Landing page. |
| **Alternative Flows:** | 1. If security questions are part of the password recovery process, the user is prompted to answer them. 2. If the answer is incorrect, the system prompts for a retry or initiates alternative recovery methods like email verification. |
| **Exceptions:** | E1. In the Normal Flow, if the User's internet connection is poor or disconnected during any step of the password reset process:   1. The system informs the User that the password reset cannot proceed due to network error.   E2. In the Normal Flow, if the User provides a invalid email , they will not get any password reset link. |
| **Business Rules** | BR-1: Users must have a registered account to use the password reset functionality.  BR-2: Password reset requests must be processed securely and in a timely manner.  BR-3: Users must have a stable and valid internet connection to complete the password reset process. |
| **Assumptions:** | AS-1: Users have previously registered an account in the Mihawk web application.  AS-2: User-provided details are accurate and valid. |

### 3.2.4 UC-4: Logout Account

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: User |
| **Trigger:** | The User wants to logout from Mihawk web application. |
| **Preconditions:** | PRE-1: User is currently logged in to the application.  PRE-2: User has an active internet connection. |
| **Postconditions:** | POST-1: User's session is terminated, and they are logged out of the application.  POST-2: User is redirected to the application's login screen.  POST-3: Any unsaved changes or incomplete actions within the application are discarded. |
| **Normal Flow:** | 1. User, while being logged in to the application, decides to log out for various reasons, such as completing their session, protecting their privacy. 2. User accesses the application's dashboard. 3. User selects the "Logout" or "Sign Out" option. 4. The application prompts the User to confirm their decision to log out. 5. User confirms to log out. 6. The application terminates the User's session, clearing their login credentials and any temporary data associated with the session. 7. User is immediately redirected to the login screen, where they can choose to log in again if needed. |
| **Alternate Flow:** | 4a. In step 4 of the Normal Flow, when the application prompts the User to confirm their decision to log out:   1. User decides to cancel the logout action. 2. User selects the "Cancel" option. 3. The application cancels the logout process and returns the User to their previous screen or action within the application. |
| **Exceptions:** | None |
| **Business rules:** | BR-1: The User must be logged in to initiate the logout process.  BR-2: The application should clear any session-related data and credentials upon successful logout to ensure the User's privacy and security.  BR-3: The User should be redirected to the login screen after logout to provide them with an option to log in again if needed. |
| **Assumptions:** | AS-1: Users are responsible for confirming their intention to log out to prevent accidental logouts.  AS-2: Users understand that any unsaved changes or actions within the application will be lost upon logout, and they accept this as part of the process. |

### UC-5: Initialize Drone for Surveillance

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Security Personnel |
| **Trigger:** | The user wants to initialize the drone for surveillance. |
| **Preconditions:** | PRE-1: The drone must be connected and ready |
| **Postconditions:** | POST-1: The drone is successfully initialized. |
| **Normal Flow:** | 1. The user initiates drone initialization.  2. The system runs diagnostics.  3. The system confirms the drone is ready for flight. |
| **Alternative Flows:** |  |
| **Exceptions:** | E1. Drone connection failure:  1. The system notifies the user. |
| **Business Rules** | BR-1: Initialization must take under 2 minutes. |
| **Assumptions:** | AS-1: The drone is functional and charged.  . |

### UC-6: Drone Control

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Security Personnel |
| **Trigger:** | The user wants to control the drone. |
| **Preconditions:** | PRE-1: The drone must be initialized. |
| **Postconditions:** | POST-1: The drone is controlled successfully. |
| **Normal Flow:** | 1. The user accesses the flight control interface.  2. The user navigates the drone manually. |
| **Alternative Flows:** |  |
| **Exceptions:** | E1. Signal loss:  1. The system attempts to reconnect to the drone. |
| **Business Rules** | BR-1: Drone must be controllable at all times. |
| **Assumptions:** | AS-1: The user has basic drone operation skills.  . |

### UC-7: Real-Time Video Feed

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Security Personnel |
| **Trigger:** | The user monitors the live video feed from the drone. |
| **Preconditions:** | PRE-1: The drone must be in flight. |
| **Postconditions:** | POST-1: The user views real-time video from the drone. |
| **Normal Flow:** | 1. The user clicks on “Live Feed.”  2. The system connects to the drone’s camera.  3. The user views the video. |
| **Alternative Flows:** |  |
| **Exceptions:** | |  | | --- | |  |  |  | | --- | |  |  |  | | --- | | E1. Camera failure:  1. The system alerts the user. | |
| **Business Rules** | BR-1: Live feed must have minimal delay. |
| **Assumptions:** | AS-1: The camera and internet connection are functional. |

### UC-8: Detect Threats in Surveillance Area

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Security Personnel |
| **Trigger:** | The system detects a threat using video analytics. |
| **Preconditions:** | |  | | --- | |  |  |  | | --- | | PRE-1: The drone is in flight. | |
| **Postconditions:** | POST-1: The system identifies and flags a threat. |
| **Normal Flow:** | 1. The system analyzes the video feed.  2. The system detects a suspicious object or activity.  3. The system flags the threat and alerts the user. |
| **Alternative Flows:** | Instead of relying on automatic detection, the user identifies a threat manually and flags it for the system. |
| **Exceptions:** | E1. False positives:  1. The system allows manual override of threat detection. |
| **Business Rules** | BR-1: Threat detection accuracy must exceed 90%. |
| **Assumptions:** | AS-1: The algorithms are properly trained.  . |

### UC-9: Send Alert Notification

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Security Personnel |
| **Trigger:** | The system sends an alert when a threat is detected. |
| **Preconditions:** | PRE-1: A threat must be detected. |
| **Postconditions:** | POST-1: The user receives an alert. |
| **Normal Flow:** | 1. The system flags a detected threat.  2. The system sends an alert to the user. |
| **Alternative Flows:** | If the system does not receive acknowledgment from the user after a certain time, it escalates the alert by notifying additional personnel or sending repeat notifications. |
| **Exceptions:** | E1. Alert delivery failure:  1. The system retries sending the alert. |
| **Business Rules** | BR-1: Alerts must be delivered within 30 seconds. |
| **Assumptions:** | AS-1: The user has configured alert settings.. |

### UC-10: Store Flagged Data on Blockchain

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The system stores flagged surveillance data on the blockchain. |
| **Preconditions:** | PRE-1: The drone is recording surveillance data. |
| **Postconditions:** | POST-1: Data is stored securely on the blockchain. |
| **Normal Flow:** | 1. The system encrypts the data.  2. The system stores data on the blockchain. |
| **Alternative Flows:** | 1. If the system is unable to connect to the blockchain, it stores the flagged data locally. 2. Once the connection is restored, the data is automatically uploaded to the blockchain. |
| **Exceptions:** | E1. Blockchain failure:  1. The system retries the storage operation. |
| **Business Rules** | BR-1: Data storage must be secure and immutable. |
| **Assumptions:** | AS-1: The blockchain network is operational.  . |

### UC-11: Retrieve Flagged Data from Blockchain

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The user retrieves stored data for review. |
| **Preconditions:** | PRE-1: The data must exist on the blockchain. |
| **Postconditions:** | POST-1: The user retrieves the data successfully. |
| **Normal Flow:** | 1. The user requests access to the stored data.  2. The system decrypts and retrieves the data from the blockchain. |
| **Alternative Flows:** | If full retrieval fails due to connection issues, the system retrieves partial data or metadata, allowing the user to access critical information. |
| **Exceptions:** | E1. Data retrieval failure:  1. The system retries retrieving the data. |
| **Business Rules** | BR-1: Data must remain secure during the retrieval process. |
| **Assumptions:** | AS-1: The user has appropriate access permissions.  . |

### UC-12: Conduct Pre-Flight System Check

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Drone Operator |
| **Trigger:** | The operator wants to perform a pre-flight check on the drone. |
| **Preconditions:** | PRE-1: The drone must be ready for flight. |
| **Postconditions:** | POST-1: The drone's system is confirmed functional before flight. |
| **Normal Flow:** | 1. The operator selects the option for a pre-flight check.  2. The system checks drone components (battery, sensors, communication).  3. The system provides a summary of the drone's status. |
| **Alternative Flows:** | If any components show warning signs (e.g., low battery or signal strength), the system suggests performing an extended diagnostic test for detailed insights. |
| **Exceptions:** | E1. Component failure detected:  1. The system prompts the operator to resolve the issue before flying. |
| **Business Rules** | BR-1: Pre-flight checks must be performed before each mission. |
| **Assumptions:** | AS-1: All system components are functional unless notified otherwise.  . |

### UC-13: Monitor Drone Battery Level

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Drone Operator |
| **Trigger:** | The operator wants to check the drone's battery level. |
| **Preconditions:** | PRE-1: The drone must be powered on and connected. |
| **Postconditions:** | POST-1: The battery status is displayed to the operator. |
| **Normal Flow:** | 1. The operator clicks on "Battery Status."  2. The system retrieves the current battery level and displays it. |
| **Alternative Flows:** |  |
| **Exceptions:** | E1. Connection failure:  1. The system attempts to reconnect to the drone. |
| **Business Rules** | BR-1: Battery levels should be monitored continuously during flight. |
| **Assumptions:** | AS-1: The battery is charged and working correctly. |

### UC-14: Assign Drone Operator Role

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The administrator wants to assign the "Drone Operator" role to a user. |
| **Preconditions:** | PRE-1: The user must have an active account. |
| **Postconditions:** | POST-1: The user receives "Drone Operator" permissions. |
| **Normal Flow:** | 1. The administrator selects a user from the list.  2. The system provides a list of available roles.  3. The administrator assigns the "Drone Operator" role. |
| **Alternative Flows:** |  |
| **Exceptions:** | E1. Invalid user selection:  1. The system prompts the administrator to select a valid user. |
| **Business Rules** | BR-1: Only users with admin privileges can assign roles. |
| **Assumptions:** | AS-1: The user has the necessary skills for the role. |

### UC-15: Remove User Account

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The administrator wants to remove a user account from the system. |
| **Preconditions:** | PRE-1: The user account must exist in the system. |
| **Postconditions:** | POST-1: The user account is deleted from the system. |
| **Normal Flow:** | 1. The administrator selects the user to be deleted.  2. The system confirms the action with the administrator.  3. The user account is deleted from the database. |
| **Alternative Flows:** | Instead of permanent deletion, the administrator has the option to deactivate the account temporarily, preserving user data for future reactivation. |
| **Exceptions:** | E1. User account deletion failure:  1. The system retries the deletion or alerts the administrator. |
| **Business Rules** | BR-1: Only users with admin rights can delete accounts. |
| **Assumptions:** |  |

### UC-16: User Activity Log Monitoring

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The administrator wants to view the activity logs of users. |
| **Preconditions:** | PRE-1: Users must be active in the system. |
| **Postconditions:** | POST-1: The administrator views user activity details. |
| **Normal Flow:** | 1. The administrator selects "Activity Log" from the menu.  2. The system displays a list of user activities, such as logins, patrols, etc. |
| **Alternative Flows:** | The administrator can apply filters (e.g., specific users or time ranges) to narrow down the log view to specific activities. |
| **Exceptions:** | E1. No recent user activity:  1. The system notifies the administrator that no logs are available. |
| **Business Rules** | BR-1: Activity logs should be updated in real time. |
| **Assumptions:** | AS-1: User activities are logged by the system.  . |

### UC-17: User Role Modification

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The administrator wants to modify a user’s role. |
| **Preconditions:** | PRE-1: The user must already have a role assigned. |
| **Postconditions:** | POST-1: The user’s role is successfully modified. |
| **Normal Flow:** | 1. The administrator selects the user.  2. The system displays the current role.  3. The administrator selects a new role and confirms the change. |
| **Alternative Flows:** |  |
| **Exceptions:** | E1. Role modification failure:  1. The system alerts the administrator. |
| **Business Rules** | BR-1: Only administrators can modify roles. |
| **Assumptions:** | AS-1: The administrator selects a valid role. |

### UC-18: Send System Notifications to Users

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: System Administrator |
| **Trigger:** | The administrator sends notifications to users regarding system updates. |
| **Preconditions:** | PRE-1: Users must have valid email addresses. |
| **Postconditions:** | POST-1: Users receive the notifications successfully. |
| **Normal Flow:** | 1. The administrator writes the notification message.  2. The system sends the notifications to all registered users. |
| **Alternative Flows:** | The system provides an option to schedule notifications for future delivery, allowing administrators to plan communications during off-peak times. |
| **Exceptions:** | E1. Notification delivery failure:  1. The system retries sending or logs the failure. |
| **Business Rules** | BR-1: Notifications must be sent within a defined timeframe. |
| **Assumptions:** | AS-1: The email system is functional. |

### UC-19: Alert User on Drone Malfunction

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Drone Operator |
| **Trigger:** | The system detects a drone malfunction during flight and alerts the operator. |
| **Preconditions:** | PRE-1: The drone must be malfunctioning. |
| **Postconditions:** | POST-1: The operator is alerted in real time. |
| **Normal Flow:** | 1. The system detects the malfunction.  2. The system alerts the operator via notification and sound. |
| **Alternative Flows:** | After detecting a malfunction, the system attempts to run basic troubleshooting steps and provides recommendations to the operator. |
| **Exceptions:** | E1. Notification failure:  1. The system retries sending the alert. |
| **Business Rules** | BR-1: Malfunctions must be reported immediately. |
| **Assumptions:** | AS-1: The system is capable of identifying malfunctions.. |

## Event Response Tables

### ERT-1: Drone Malfunction Detected

**Table 3.3.1: Event Response Table for Drone Malfunction Detected**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 1 | Drone detects a malfunction during flight | Drone health data (battery status, motor function, sensor data) | Drone operating under normal conditions | Drone sensors malfunction, causing an error in data reporting | System immediately sends an alert to the user with details of the malfunction and triggers the drone to land safely | System continues to operate as normal but logs the malfunction in the system database for future analysis | Drone attempts emergency landing, and system waits for user response |

### ERT-2: Threat Detected in Surveillance Area

**Table 3.3.2: Event Response Table for Threat Detected in Surveillance Area**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 2 | Drone's camera detects a potential threat in the surveillance area | Video feed from the drone, threat detection algorithm | Drone conducting regular surveillance | False positive in threat detection (e.g., movement of non-threatening objects) | The system captures and flags the detected threat, and an alert is sent to the user. The system stores this flagged data on the blockchain for further analysis | The system continues to monitor the area without any alert or flagged data being logged | Data on the potential threat is stored and user is alerted |

### ERT-3: Drone Battery Level Reaches Critical

**Table 3.3.3: Event Response Table for Drone Battery Level Reaches Critical**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 3 | Drone battery level drops below a critical threshold (e.g., 15%) | Battery percentage data from drone sensors | Drone is actively flying and performing surveillance | Sensor failure, inaccurate battery reporting | The system sends an alert to the user, and the drone is instructed to return to the base or land safely | The system continues the flight, but the battery level is logged, and the system keeps monitoring it for further degradation | Drone is in the process of returning to base or is safely landed |

### ERT-4: Operator Assignment/Change

**Table 3.3.4: Event Response Table for Operator Assignment/Change**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 4 | Drone operator is assigned or changed for a particular drone | Operator credentials, drone assignment data | Drone operator unassigned or previous operator assigned | Invalid operator credentials or authorization | The system updates the database with the new operator details and logs the change for future tracking | The system denies the request and prompts the user to check credentials or permissions | Drone is now assigned to the updated operator |

### ERT-5: Threat Alert Notification Sent

**Table 3.3.5: Event Response Table for Threat Alert Notification Sent**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 5 | System detects a threat and sends an alert notification to authorized users | |  | | --- | |  |  |  | | --- | | Detected threat data, user contact information (email, phone number) | | Surveillance feed being monitored but no threat detected yet | Alert notification fails due to communication channel error (e.g., email server down, SMS gateway failure) | The system successfully sends the notification to users via their preferred communication method (email, SMS, app notification) | The system retries the alert or selects an alternative communication method to notify the users | Users notified of the threat and response is logged |

### ERT-6: Pre-Flight System Check

**Table 3.3.6: Event Response Table for Pre-Flight System Check**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 6 | Drone performs a pre-flight system check before taking off | |  | | --- | |  |  |  | | --- | | Sensor data (battery, motor function, GPS status, camera functionality) | | Drone is on standby, ready for flight | One or more system components fail the check (e.g., GPS not functioning) | System passes the check and authorizes the drone for takeoff | System denies the takeoff request and logs the failed check component, alerting the operator | Drone is cleared for flight or kept grounded until issues are resolved |

### ERT-7: Malfunction Alert Notification Sent

**Table 3.3.7: Event Response Table for Malfunction Alert Notification Sent**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Event** | **Data Element** | **System State** | **Exception Condition** | **System Response** | | **System State** |
| **In case of yes** | **In case of no** |
| 7 | Malfunction occurs during the drone's operation | Error data from drone sensors, user contact details | Drone operating without issues | Communication failure in sending the alert | System sends an alert to the assigned operator with malfunction details and suggests actions (e.g., return to base) | System logs the malfunction, attempts to send the alert again through alternative communication methods | User notified of the malfunction and logs updated |

# Functional Requirements

## Use Case Name: Register Account

## FR-1: RegisterAccount.UserRegistration

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | RegisterAccount.UserRegistration |
| **Requirement** | The system shall allow Users to register an account by providing valid registration details (email address, password, and contact number) and shall store the information. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures that users can create an account and access the system |
| **Business Rule** | BR-2 : Users must have a stable internet connection to complete registration |
| **Dependencies** | - |
| **Priority** | High |

## FR-2: RegisterAccount.FieldValidation

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | RegisterAccount. FieldValidation |
| **Requirement** | The system shall validate the registration fields (email, password, contact number) before storing user details. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures valid information is provided to prevent errors and ensure data integrity |
| **Business Rule** | BR-1: Users must have valid login credentials. |
| **Dependencies** | FR-1 |
| **Priority** | High |

## FR-3: RegisterAccount.UniqueEmailCheck

|  |  |
| --- | --- |
| **Identifier** | FR-3 |
| **Title** | RegisterAccount.UniqueEmailCheck |
| **Requirement** | The system shall check if the provided email address is already registered. If it exists, the user shall be prompted to log in or use a different email address. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures that each email is unique and prevents duplicate accounts. |
| **Business Rule** | BR-3: Only one account can be associated with a single email address |
| **Dependencies** | FR-2 |
| **Priority** | High |

## FR-4: RegisterAccount.PasswordSecurity

|  |  |
| --- | --- |
| **Identifier** | FR-4 |
| **Title** | RegisterAccount.PasswordSecurity |
| **Requirement** | The system shall enforce password security standards by requiring that the password includes a minimum of 8 characters, one uppercase letter, one number, and one special character |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures that users create strong, secure passwords to protect their accounts |
| **Business Rule** | BR-4: Passwords must adhere to the specified security guidelines |
| **Dependencies** | FR-2 |
| **Priority** | High |

## FR-5: RegisterAccount.ConfirmationEmail

|  |  |
| --- | --- |
| **Identifier** | FR-5 |
| **Title** | RegisterAccount.ConfirmationEmail |
| **Requirement** | Upon successful registration, the system shall send a confirmation email to the user’s registered email address with an activation link to verify the account |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures the user’s email address is valid and that the account is correctly activated |
| **Business Rule** | BR-5: The registration is not complete until the user verifies their email address |
| **Dependencies** | FR-3 |
| **Priority** | Medium |

## FR-5: RegisterAccount.StoreUserData

|  |  |
| --- | --- |
| **Identifier** | FR-6 |
| **Title** | RegisterAccount.StoreUserData |
| **Requirement** | The system shall securely store the user’s registration information (email, password, contact number) in the database upon successful account creation. Passwords must be hashed |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures that user data is stored securely and is protected from unauthorized access |
| **Business Rule** | BR-6: Passwords must be encrypted using industry-standard hashing algorithms |
| **Dependencies** | FR-1, FR-2, FR-4 |
| **Priority** | High |

## FR-7: RegisterAccount.RegistrationErrorHandling

|  |  |
| --- | --- |
| **Identifier** | FR-7 |
| **Title** | RegisterAccount.RegistrationErrorHandling |
| **Requirement** | The system shall provide an error message if registration fails due to invalid input, system errors, or network issues, and guide the user to resolve the issue |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides feedback to the user when an error occurs during registration, ensuring that issues are addressed quickly |
| **Business Rule** | BR-7: The error message must be clear and informative, helping users understand the nature of the issue |
| **Dependencies** | FR-1, FR-2 |
| **Priority** | Medium |

## Use Case Name: User Login

## FR-8: UserLogin.LoginForm

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | UserLogin.LoginForm |
| **Requirement** | The user shall be able to access the login form by providing their registered email address and password |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides users with the ability to log into the Mihawk system |
| **Business Rule** | BR-1: Only registered users can access the login form |
| **Dependencies** | - |
| **Priority** | High |

## FR-9: UserLogin.ValidateLoginDetails

|  |  |
| --- | --- |
| **Identifier** | FR-9 |
| **Title** | UserLogin.ValidateLoginDetails |
| **Requirement** | The system shall validate the user's login details (email and password) to ensure they match the stored credentials. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures that only valid and registered users can log in |
| **Business Rule** | BR-2: The system must validate login credentials within 2 seconds of submission |
| **Dependencies** | FR-8 |
| **Priority** | High |

## FR-10: UserLogin.PasswordEncryptionCheck

|  |  |
| --- | --- |
| **Identifier** | FR-10 |
| **Title** | UserLogin.PasswordEncryptionCheck |
| **Requirement** | The system shall verify the password by comparing the user-entered password hash with the stored hash in the database |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Protects sensitive user data by ensuring password security during login |
| **Business Rule** | BR-3: Passwords must be hashed using secure algorithms |
| **Dependencies** | FR-9 |
| **Priority** | High |

## FR-11: UserLogin.LoginAttemptsLimit

|  |  |
| --- | --- |
| **Identifier** | FR-11 |
| **Title** | UserLogin.LoginAttemptsLimit |
| **Requirement** | The system shall limit the number of failed login attempts to 5. After exceeding this limit, the user's account shall be temporarily locked for 30 minutes |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Prevents brute-force attacks on user accounts |
| **Business Rule** | BR-4: Users must wait 30 minutes after 5 failed login attempts before they can try again |
| **Dependencies** | FR-9, FR-10 |
| **Priority** | High |

## FR-12: UserLogin.ErrorHandling

|  |  |
| --- | --- |
| **Identifier** | FR-12 |
| **Title** | UserLogin.ErrorHandling |
| **Requirement** | The system shall provide clear error messages if the login fails due to incorrect credentials or system errors, without revealing unnecessary information about the failure |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides users with informative feedback while maintaining security |
| **Business Rule** | BR-5: Error messages must not specify whether the email or password is incorrect to avoid security risks |
| **Dependencies** | FR-9, FR-10 |
| **Priority** | Medium |

## FR-13: UserLogin.RememberMeOption

|  |  |
| --- | --- |
| **Identifier** | FR-13 |
| **Title** | UserLogin.RememberMeOption |
| **Requirement** | The system shall provide a "Remember Me" option, allowing users to stay logged in for a predefined period unless they manually log out |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Enhances user experience by allowing quick access without frequent logins |
| **Business Rule** | BR-6: The "Remember Me" option should expire after 7 days |
| **Dependencies** | FR-9 |
| **Priority** | Medium |

## FR-14: UserLogin.ForgotPassword

|  |  |
| --- | --- |
| **Identifier** | FR-14 |
| **Title** | UserLogin.ForgotPassword |
| **Requirement** | The system shall provide a "Forgot Password" link on the login page, allowing users to reset their password by providing their registered email address |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Allows users to recover access if they forget their password |
| **Business Rule** | BR-7: Password reset links must expire after 24 hours |
| **Dependencies** | FR-8 |
| **Priority** | High |

## FR-15: UserLogin.SessionManagement

|  |  |
| --- | --- |
| **Identifier** | FR-15 |
| **Title** | UserLogin.SessionManagement |
| **Requirement** | The system shall manage user sessions securely by generating unique session tokens upon successful login and automatically logging users out after 15 minutes of inactivity |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures session security and prevents unauthorized access after periods of inactivity |
| **Business Rule** | BR-8: Session tokens must be securely stored and invalidated upon logout or inactivity |
| **Dependencies** | FR-9, FR-10 |
| **Priority** | High |

## Use Case Name: Password Recovery

## FR-16: PasswordRecovery.RequestResetLink

|  |  |
| --- | --- |
| **Identifier** | FR-16 |
| **Title** | PasswordRecovery.RequestResetLink |
| **Requirement** | The system shall allow users to request a password reset link by providing their registered email address |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Enables users to recover their password if forgotten |
| **Business Rule** | BR-1 : Users must have a registered account to use password reset functionality |
| **Dependencies** | FR-1 |
| **Priority** | High |

## FR-17: PasswordRecovery.EmailVerification

|  |  |
| --- | --- |
| **Identifier** | FR-17 |
| **Title** | PasswordRecovery.EmailVerification |
| **Requirement** | The system shall verify the provided email address against registered accounts before sending a password reset link |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures the reset link is sent only to users with valid accounts |
| **Business Rule** | BR-1 : Users must have a registered account to use password reset functionality |
| **Dependencies** | FR-9 |
| **Priority** | High |

## FR-18: PasswordRecovery.LinkExpiration

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | PasswordRecovery.LinkExpiration |
| **Requirement** | The password reset link sent to the user shall expire after a set time (e.g., 24 hours) |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides security by ensuring the reset link is time-bound |
| **Business Rule** | BR-2 : Password reset requests must be processed securely |
| **Dependencies** | FR-9 |
| **Priority** | Medium |

## Use Case Name: LogoutAccount

## FR-19: LogoutAccount.TerminateSession

|  |  |
| --- | --- |
| **Identifier** | FR-19 |
| **Title** | LogoutAccount.TerminateSession |
| **Requirement** | The system shall terminate the user session upon logout and clear session-related data. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures security by clearing sensitive information. |
| **Business Rule** | BR-2 : Session-related data should be cleared after logout |
| **Dependencies** | - |
| **Priority** | High |

## FR-20: LogoutAccount.RedirectToLogin

|  |  |
| --- | --- |
| **Identifier** | FR-20 |
| **Title** | LogoutAccount.RedirectToLogin |
| **Requirement** | The system shall redirect the user to the login screen after a successful logout |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides an option for users to log in again if needed |
| **Business Rule** | BR-3 : Users should be redirected to the login screen after logout |
| **Dependencies** | - |
| **Priority** | High |

## Use Case Name: Drone Control

## FR-21: DroneControl.ActivateDrone

|  |  |
| --- | --- |
| **Identifier** | FR-21 |
| **Title** | DroneControl. ActivateDrone |
| **Requirement** | The system shall allow the user to activate a drone after successful login. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Enables users to remotely control a drone, ensuring they are logged in before issuing command |
| **Business Rule** | BR-9: Only logged-in users can control drones |
| **Dependencies** | FR-9, FR-15 |
| **Priority** | High |

## FR-22: DroneControl.RealTimeFeedback

|  |  |
| --- | --- |
| **Identifier** | FR-22 |
| **Title** | DroneControl.RealTimeFeedback |
| **Requirement** | The system shall provide real-time feedback from the drone (e.g., camera feed, altitude, and battery status) while the drone is in operation. |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Real-time information ensures users can monitor drone activity and make informed control decisions |
| **Business Rule** | BR-10: Real-time data must be updated every 5 second |
| **Dependencies** | FR-21 |
| **Priority** | High |

## FR-23: DroneControl.ManualControl

|  |  |
| --- | --- |
| **Identifier** | FR-23 |
| **Title** | DroneControl.ManualCommands |
| **Requirement** | The system shall allow the user to send control commands (e.g., take off, land, change direction, adjust altitude) to the drone |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Allows the user to remotely operate the drone |
| **Business Rule** | BR-11: Manual control commands must respond within 1 second |
| **Dependencies** | FR-22 |
| **Priority** | High |

## FR-24: DroneControl.BatteryWarning

|  |  |
| --- | --- |
| **Identifier** | FR-24 |
| **Title** | DroneControl. BatteryWarning |
| **Requirement** | The system shall notify the user when the drone's battery falls below 20% |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Alerts the user to low battery levels to avoid drone failure |
| **Business Rule** | BR-12: The system must display a warning when the battery is below 20% |
| **Dependencies** | FR-22 |
| **Priority** | High |

## FR-25: DroneControl.AutomaticLanding

|  |  |
| --- | --- |
| **Identifier** | FR-25 |
| **Title** | DroneControl.AutomaticLanding |
| **Requirement** | The system shall automatically land the drone in case of a critical battery level or loss of connection with the controller |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures drone safety by handling emergencies automatically |
| **Business Rule** | BR-13: The system must trigger an automatic landing if the battery is below 10% or the connection is lost for more than 10 seconds. |
| **Dependencies** | FR-22, FR-23 |
| **Priority** | High |

## Use Case Name: Real Time Feed

## FR-26: RealTimeFeed.ActivateFeed

|  |  |
| --- | --- |
| **Identifier** | FR-26 |
| **Title** | RealTimeFeed.ActivateFeed |
| **Requirement** | The system shall allow the user to activate the real-time feed after successfully logging in and establishing a connection with the drone |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures only authorized users can access the live feed of the drone |
| **Business Rule** | BR-14: Only logged-in users with control access can view the real-time feed. |
| **Dependencies** | FR-9, FR-21 |
| **Priority** | High |

## FR-27: RealTimeFeed.StreamVideo

|  |  |
| --- | --- |
| **Identifier** | FR-27 |
| **Title** | RealTimeFeed.StreamVideo |
| **Requirement** | The system shall stream real-time video from the drone's camera to the user interface with minimal delay (less than 12 seconds) |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides real-time visual feedback for the user to monitor drone operations |
| **Business Rule** | BR-15: The video feed should not exceed a delay of 2 seconds |
| **Dependencies** | FR-26 |
| **Priority** | High |

## FR-28: RealTimeFeed.StreamDroneData

|  |  |
| --- | --- |
| **Identifier** | FR-28 |
| **Title** | RealTimeFeed.StreamDroneData |
| **Requirement** | The system shall display real-time drone data (e.g., altitude, speed coordinates, battery level) from the drone on the user interface alongside the video feed |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Provides essential telemetry data for effective drone operation and monitoring |
| **Business Rule** | BR-16: Drone data must update at least once per second |
| **Dependencies** | FR-27 |
| **Priority** | High |

## FR-29: RealTimeFeed.NotifyFeedDisruption

|  |  |
| --- | --- |
| **Identifier** | FR-29 |
| **Title** | RealTimeFeed.NotifyFeedDisruption |
| **Requirement** | The system shall notify the user if the real-time feed (video or sensor data) is disrupted or lost for more than 5 seconds |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Alerts the user to connection issues that could impact drone control and decision-making |
| **Business Rule** | BR-17: The system must provide an alert if the feed is interrupted for more than 5 seconds |
| **Dependencies** | FR-27, FR-28 |
| **Priority** | High |

## FR-30: RealTimeFeed.ControlFeedQuality

|  |  |
| --- | --- |
| **Identifier** | FR-30 |
| **Title** | RealTimeFeed.ControlFeedQuality |
| **Requirement** | The system shall allow the user to adjust the video quality of the real-time feed (e.g., low, medium, high) based on the available bandwidth |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Ensures optimal performance based on the user's internet connection |
| **Business Rule** | BR-18: The system should automatically adjust video quality if the bandwidth fluctuates. |
| **Dependencies** | FR-27, FR-28 |
| **Priority** | Medium |

## FR-31: RealTimeFeed.SaveVideoFeed

|  |  |
| --- | --- |
| **Identifier** | FR-31 |
| **Title** | RealTimeFeed.ActivateFeed |
| **Requirement** | The system shall allow the user to record and save the real-time video feed locally for later viewing or analysis |
| **Source** | Muhammad Hozefa Rauf |
| **Rationale** | Allows users to store video data for future reference or reporting purposes |
| **Business Rule** | BR-19: The system must notify the user when recording starts and stops |
| **Dependencies** | FR-27, FR-28 |
| **Priority** | low |

## Use Case Name: Detect Threats in Surveillance Area

## FR-32: ThreatDetection.InitializeSurveillance

|  |  |
| --- | --- |
| **Identifier** | FR-32 |
| **Title** | DroneControl.InitializeSurveillance |
| **Requirement** | The system shall allow the user to initialize threat detection after successfully logging in and establishing a connection with the drone in the surveillance area |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures that only authorized users can initiate threat detection operations |
| **Business Rule** | BR-20: The system must verify the user’s authorization before enabling threat detection |
| **Dependencies** | FR-9, FR-23 |
| **Priority** | High |

## FR-33: ThreatDetection.MonitorArea

|  |  |
| --- | --- |
| **Identifier** | FR-33 |
| **Title** | ThreatDetection.MonitorArea |
| **Requirement** | The system shall continuously monitor the designated surveillance area for any suspicious activities or objects using live video feed |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Enables the system to automatically scan the area for potential threats in real-time |
| **Business Rule** | BR-21: The drone must cover the entire designated area during surveillance without missing any zones |
| **Dependencies** | FR-26, FR-27 |
| **Priority** | High |

## FR-34: ThreatDetection.DetectMovementPatterns

|  |  |
| --- | --- |
| **Identifier** | FR-34 |
| **Title** | ThreatDetection.DetectMovementPatterns |
| **Requirement** | The system shall analyze real-time video and sensor data to detect abnormal movement patterns or unauthorized objects within the surveillance area |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures the system identifies and flags unusual activity that could indicate potential threat |
| **Business Rule** | BR-22: Movement patterns deviating from a predefined normal range should trigger a threat alert. |
| **Dependencies** | FR-33, FR-28 |
| **Priority** | High |

## FR-35: ThreatDetection.ClassifyThreats

|  |  |
| --- | --- |
| **Identifier** | FR-35 |
| **Title** | ThreatDetection.ClassifyThreats |
| **Requirement** | The system shall classify detected objects or activities based on predefined threat levels (e.g., low, medium, high) and flag them accordingly |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Provides context to detected threats, allowing the user to prioritize responses |
| **Business Rule** | BR-23: Each detected threat should be assigned a classification based on its level of danger or abnormality |
| **Dependencies** | FR-34 |
| **Priority** | Medium |

## FR-36: ThreatDetection.GenerateAlert

|  |  |
| --- | --- |
| **Identifier** | FR-36 |
| **Title** | ThreatDetection.GenerateAlert |
| **Requirement** | The system shall generate an alert (visual and audio) for the user when a potential threat is detected and classified |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Notifies the user of potential security issues in the surveillance area |
| **Business Rule** | BR-24: Alerts should be issued immediately upon detecting a threat |
| **Dependencies** | FR-35, FR-29 |
| **Priority** | High |

## FR-37: ThreatDetection.RecordThreatData

|  |  |
| --- | --- |
| **Identifier** | FR-37 |
| **Title** | ThreatDetection.RecordThreatData |
| **Requirement** | The system shall record video, sensor data, and threat classifications for all detected threats during the surveillance operation for later analysis |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Allows users to review threat incidents for post-surveillance analysis and reporting |
| **Business Rule** | BR-25: All threat data must be stored securely and accessible for review at any time |
| **Dependencies** | FR-36, FR-31 |
| **Priority** | Medium |

## Use Case Name: Send Alert Notification

## FR-38: AlertNotification.ConfigureAlertSettings

|  |  |
| --- | --- |
| **Identifier** | FR-38 |
| **Title** | AlertNotification.ConfigureAlertSettings |
| **Requirement** | The system shall allow the user to configure alert notification settings, including the type of alerts (e.g., email, SMS, push notifications) and alert priority levels |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Provides flexibility for users to receive alerts in their preferred method and ensure timely responses to critical threats |
| **Business Rule** | - |
| **Dependencies** | FR-9 |
| **Priority** | Medium |

## FR-39: AlertNotification.GenerateRealTimeAlert

|  |  |
| --- | --- |
| **Identifier** | FR-39 |
| **Title** | AlertNotification.GenerateRealTimeAlert |
| **Requirement** | The system shall generate and send real-time alert notifications to the user when a potential threat is detected by the drone or the surveillance system |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures immediate notification of threats so the user can take action swiftly. |
| **Business Rule** | - |
| **Dependencies** | FR-36 |
| **Priority** | High |

## FR-40: AlertNotification.MultipleRecipients

|  |  |
| --- | --- |
| **Identifier** | FR-40 |
| **Title** | AlertNotification.MultipleRecipients |
| **Requirement** | The system shall support sending alert notifications to multiple authorized recipients simultaneously, based on predefined settings |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Allows more than one user to be notified in critical situations, ensuring that appropriate personnel are informed |
| **Business Rule** | BR-26: Only authorized users should receive notifications based on their roles |
| **Dependencies** | FR-38 |
| **Priority** | Medium |

## FR-41: AlertNotification.DetailedAlertInformation

|  |  |
| --- | --- |
| **Identifier** | FR-41 |
| **Title** | AlertNotification.DetailedAlertInformation |
| **Requirement** | The system shall provide detailed information in the alert notification, including the location of the threat, time, and type of detected threat |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures users receive all necessary information in the alert to respond appropriately |
| **Business Rule** | - |
| **Dependencies** | FR-33, FR-34 |
| **Priority** | High |

## FR-42: AlertNotification.ManageAlertHistory

|  |  |
| --- | --- |
| **Identifier** | FR-42 |
| **Title** | AlertNotification.ManageAlertHistory |
| **Requirement** | The system shall record and store all sent alert notifications in the system, allowing users to review the history of alerts at any time |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Provides an audit trail and review capability for past threat incidents |
| **Business Rule** | - |
| **Dependencies** | FR-37 |
| **Priority** | Medium |

## Use Case Name: Store Flagged Data on Blockchain

## FR-43: Blockchain.StoreFlaggedData

|  |  |
| --- | --- |
| **Identifier** | FR-43 |
| **Title** | DroneHealth.StoreFlaggedData |
| **Requirement** | The system shall store flagged surveillance data on the blockchain, including metadata such as timestamp, location, and threat type |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures that flagged data is securely stored, tamper-proof, and can be traced back for auditing purposes |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | High |

## FR-44: Blockchain.VerifyDataIntegrity

|  |  |
| --- | --- |
| **Identifier** | FR-44 |
| **Title** | DroneHealth.VerifyDataIntegrity |
| **Requirement** | The system shall verify the integrity of the flagged data before storing it on the blockchain to ensure no unauthorized modifications have occurred |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures the data stored on the blockchain is accurate and unaltered |
| **Business Rule** | BR-27: Only verified data can be stored on the blockchain. |
| **Dependencies** | FR-43 |
| **Priority** | High |

## FR-44: Blockchain.EncryptFlaggedData

|  |  |
| --- | --- |
| **Identifier** | FR-44 |
| **Title** | DroneHealth.EncryptFlaggedData |
| **Requirement** | The system shall encrypt all flagged data before storing it on the blockchain to protect sensitive information from unauthorized access |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Enhances data security and prevents unauthorized access to sensitive information |
| **Business Rule** | - |
| **Dependencies** | FR-43 |
| **Priority** | High |

## Use Case Name: Retrieve Flagged Data from Blockchain

## FR-45: Blockchain.RetrieveFlaggedData

|  |  |
| --- | --- |
| **Identifier** | FR-45 |
| **Title** | Blockchain.RetrieveFlaggedData |
| **Requirement** | The system shall allow authorized users to retrieve flagged surveillance data from the blockchain by providing relevant search criteria, such as timestamp, location, or threat type |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures users can retrieve the necessary data for analysis or review when required |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | High |

## FR-46: Blockchain.VerifyUserPermissions

|  |  |
| --- | --- |
| **Identifier** | FR-21 |
| **Title** | Blockchain.VerifyUserPermissions |
| **Requirement** | The system shall verify the user’s access permissions before allowing retrieval of flagged data from the blockchain |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures that only authorized users can access sensitive data to maintain security and privacy |
| **Business Rule** | - |
| **Dependencies** | FR-45 |
| **Priority** | High |

## FR-47: Blockchain.DecryptFlaggedData

|  |  |
| --- | --- |
| **Identifier** | FR-47 |
| **Title** | Blockchain.DecryptFlaggedData |
| **Requirement** | The system shall decrypt flagged data retrieved from the blockchain before presenting it to the authorized user |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Allows users to access readable data while maintaining security for the stored information |
| **Business Rule** | - |
| **Dependencies** | FR-44 |
| **Priority** | High |

## Use Case Name: Conduct Pre-Flight System Check

## FR-48: PreFlight.CheckBatteryLevel

|  |  |
| --- | --- |
| **Identifier** | FR-48 |
| **Title** | PreFlight.CheckBatteryLevel |
| **Requirement** | The system shall check the drone’s battery level before initiating the flight to ensure sufficient power for the mission |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures that the drone has enough battery to complete the mission and return safely |
| **Business Rule** | BR-28: A minimum of 75% battery is required for long-distance missions |
| **Dependencies** | - |
| **Priority** | High |

## FR-49: PreFlight.VerifyCommunication

|  |  |
| --- | --- |
| **Identifier** | FR-49 |
| **Title** | PreFlight.VerifyCommunication |
| **Requirement** | The system shall verify that communication between the drone and the ground control system is established and stable before takeoff |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures uninterrupted communication during the flight, critical for safe drone operation. |
| **Business Rule** | BR-29: Communication link must be maintained at all times during the mission |
| **Dependencies** | FR-48 |
| **Priority** | High |

## FR-50: PreFlight.CheckWeatherConditions

|  |  |
| --- | --- |
| **Identifier** | FR-50 |
| **Title** | PreFlight.CheckWeatherConditions |
| **Requirement** | The system shall check the current weather conditions and notify the operator of any unsafe conditions (e.g., high wind, rain) before flight |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures the safety of the drone and mission by preventing flights in hazardous weather conditions |
| **Business Rule** | BR-28 |
| **Dependencies** | FR-49 |
| **Priority** | Medium |

## Use Case Name: Monitor Drone Battery Level

## FR-51: MonitorBattery.CheckInitialLevel

|  |  |
| --- | --- |
| **Identifier** | FR-51 |
| **Title** | MonitorBattery.CheckInitialLevel |
| **Requirement** | The system shall check the drone’s battery level before takeoff to ensure it meets the minimum threshold for the mission |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures the drone has adequate power to perform the mission without risk of battery depletion mid-flight |
| **Business Rule** | BR-28: A minimum of 75% battery is required for long-distance missions |
| **Dependencies** | FR-48 |
| **Priority** | High |

## FR-52: MonitorBattery.TrackInFlightLevel

|  |  |
| --- | --- |
| **Identifier** | FR-52 |
| **Title** | MonitorBattery.TrackInFlightLevel |
| **Requirement** | The system shall continuously track the drone's battery level during flight and display the real-time percentage to the operator |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Enables the operator to stay informed about the battery status and make informed decisions during flight. |
| **Business Rule** | - |
| **Dependencies** | FR-49 |
| **Priority** | High |

## FR-53: MonitorBattery.SendLowBatteryAlert

|  |  |
| --- | --- |
| **Identifier** | FR-53 |
| **Title** | MonitorBattery.SendLowBatteryAlert |
| **Requirement** | The system shall send an alert to the operator when the battery level falls below 25% to allow for safe landing procedures |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Prevents battery depletion during critical operations and ensures the drone has enough power to return to base |
| **Business Rule** | - |
| **Dependencies** | FR-52 |
| **Priority** | High |

## FR-54: MonitorBattery.PerformEmergencyLanding

|  |  |
| --- | --- |
| **Identifier** | FR-54 |
| **Title** | MonitorBattery.PerformEmergencyLanding |
| **Requirement** | The system shall automatically initiate an emergency landing sequence if the battery level drops below 10%, overriding any other flight commands. |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Ensures the safety of the drone by preventing complete battery failure during flight |
| **Business Rule** | BR-29: The drone must automatically land if the battery reaches critically low levels. |
| **Dependencies** | FR-53 |
| **Priority** | High |

## FR-55: MonitorBattery.LogBatteryUsage

|  |  |
| --- | --- |
| **Identifier** | FR-55 |
| **Title** | MonitorBattery.LogBatteryUsage |
| **Requirement** | The system shall log all battery usage data during each flight, including start and end levels, for post-flight analysis |
| **Source** | Hammad ur Rehamn |
| **Rationale** | Enables the operator to review battery performance and plan maintenance or replacements as necessary. |
| **Business Rule** | - |
| **Dependencies** | FR-52 |
| **Priority** | High |

## Use Case Name: Assign Drone Operator Role

## FR-56: AssignOperator.AssignRole

|  |  |
| --- | --- |
| **Identifier** | FR-56 |
| **Title** | AssignOperator.AssignRole |
| **Requirement** | The system shall allow an administrator to assign the "Drone Operator" role to a registered user by selecting their account from the user database |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures only authorized users are permitted to control drones |
| **Business Rule** | - |
| **Dependencies** | FR-1 |
| **Priority** | High |

## FR-57: AssignOperator.SetOperatorPermissions

|  |  |
| --- | --- |
| **Identifier** | FR-57 |
| **Title** | AssignOperator.SetOperatorPermissions |
| **Requirement** | The system shall automatically grant specific permissions to the user with the Drone Operator role, such as access to the drone control interface and flight data |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures the operator has the necessary access to perform drone control tasks |
| **Business Rule** | BR-30: Only Drone Operators can access the drone control panel and related features. |
| **Dependencies** | FR-56 |
| **Priority** | High |

## FR-58: AssignOperator.RevokeOperatorRole

|  |  |
| --- | --- |
| **Identifier** | FR-58 |
| **Title** | AssignOperator.RevokeOperatorRole |
| **Requirement** | The system shall allow an administrator to revoke the Drone Operator role from a user, removing their access to drone control functionalities |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures only authorized users are permitted to control drones |
| **Business Rule** | - |
| **Dependencies** | FR-56 |
| **Priority** | High |

## FR-59: AssignOperator.NotifyUser

|  |  |
| --- | --- |
| **Identifier** | FR-59 |
| **Title** | AssignOperator.AssignRole |
| **Requirement** | The system shall send a notification to the user when they have been assigned or revoked the Drone Operator role |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Keeps the user informed about their role and permissions within the system |
| **Business Rule** | BR-31: Notifications must be sent via both email and in-system alerts |
| **Dependencies** | FR-56 |
| **Priority** | Medium |

## Use Case Name: Remove User Account

## FR-60: RemoveAccount.DeleteUserAccount

|  |  |
| --- | --- |
| **Identifier** | FR-60 |
| **Title** | RemoveAccount.DeleteUserAccount |
| **Requirement** | The system shall allow an administrator to delete a registered user account permanently from the system |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures that administrators have the ability to manage user accounts and remove inactive or unauthorized users. |
| **Business Rule** | BR-32: Only administrators with specific privileges can delete user accounts |
| **Dependencies** | FR-1 |
| **Priority** | High |

## FR-61: RemoveAccount.ValidateAdminPermission

|  |  |
| --- | --- |
| **Identifier** | FR-61 |
| **Title** | RemoveAccount.ValidateAdminPermission |
| **Requirement** | The system shall validate the administrator’s permissions before allowing the deletion of a user account |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures that only authorized administrators can remove accounts, preventing unauthorized actions |
| **Business Rule** | - |
| **Dependencies** | FR-60 |
| **Priority** | High |

## FR-62: RemoveAccount.BackupUserData

|  |  |
| --- | --- |
| **Identifier** | FR-62 |
| **Title** | RemoveAccount.BackupUserData |
| **Requirement** | The system shall prompt the administrator to back up the user’s data (such as flagged data or logs) before permanently deleting the account. |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Allows for recovery of important information before account removal |
| **Business Rule** | - |
| **Dependencies** | FR-60, FR-43 |
| **Priority** | Medium |

## FR-63: RemoveAccount.NotifyUserAndAdmin

|  |  |
| --- | --- |
| **Identifier** | FR-63 |
| **Title** | RemoveAccount.NotifyUserAndAdmin |
| **Requirement** | The system shall send a notification to both the administrator and the user being removed, indicating the account removal process. |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures both parties are aware of the action, increasing transparency and allowing the user to contest the removal if necessary |
| **Business Rule** | - |
| **Dependencies** | FR-60 |
| **Priority** | Medium |

## FR-64: RemoveAccount.RevokeAccessImmediately

|  |  |
| --- | --- |
| **Identifier** | FR-64 |
| **Title** | RemoveAccount.RevokeAccessImmediately |
| **Requirement** | The system shall immediately revoke all access rights and login credentials associated with the user once the account is flagged for removal |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Prevents the user from accessing the system after the account removal request has been initiated |
| **Business Rule** | BR-33: User access must be disabled within 5 minutes of initiating the account removal process |
| **Dependencies** | FR-60 |
| **Priority** | High |

## Use Case Name: User Activity Log Monitoring

## FR-65: ActivityLog.TrackUserLogin

|  |  |
| --- | --- |
| **Identifier** | FR-65 |
| **Title** | ActivityLog.TrackUserLogin |
| **Requirement** | The system shall log every user login attempt, including successful and unsuccessful login attempts, with the timestamp |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Helps monitor system access and detect potential security breaches |
| **Business Rule** | - |
| **Dependencies** | FR-1, FR-9 |
| **Priority** | High |

## FR-66: ActivityLog.AccessLogsByAdmin

|  |  |
| --- | --- |
| **Identifier** | FR-66 |
| **Title** | ActivityLog.AccessLogsByAdmin |
| **Requirement** | The system shall allow administrators to access and review the user activity logs based on filters such as user ID, action type, and date range |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Provides administrators with the ability to track specific user activities for auditing purposes |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | High |

## FR-67: ActivityLog.FlagSuspiciousActivity

|  |  |
| --- | --- |
| **Identifier** | FR-67 |
| **Title** | ActivityLog.FlagSuspiciousActivity |
| **Requirement** | The system shall automatically flag suspicious activities, such as multiple failed login attempts or unauthorized access attempts, and notify the administrator |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Detects potential security threats or abnormal behaviors |
| **Business Rule** | BR-34: The system must send alerts to administrators within 5 minutes of detecting suspicious activity. |
| **Dependencies** | FR-65 |
| **Priority** | High |

## FR-68: ActivityLog.ExportLogs

|  |  |
| --- | --- |
| **Identifier** | FR-68 |
| **Title** | ActivityLog.ExportLogs |
| **Requirement** | The system shall allow administrators to export the user activity logs in common formats (e.g., CSV, PDF) for external analysis and archival purposes. |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Provides flexibility in sharing logs with external auditors or for compliance purposes |
| **Business Rule** | - |
| **Dependencies** | FR-66 |
| **Priority** | Medium |

## Use Case Name: User Activity Log Monitoring

## FR-69: RoleModification.AssignUserRole

|  |  |
| --- | --- |
| **Identifier** | FR-69 |
| **Title** | RoleModification.AssignUserRole |
| **Requirement** | The system shall allow administrators to assign specific roles (e.g., operator, manager, viewer) to new users upon account creation |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Allows the system to control user permissions and access based on roles |
| **Business Rule** | - |
| **Dependencies** | FR-1, FR-56 |
| **Priority** | High |

## FR-70: RoleModification.ChangeUserRole

|  |  |
| --- | --- |
| **Identifier** | FR-70 |
| **Title** | RoleModification.ChangeUserRole |
| **Requirement** | The system shall allow administrators to change the role of an existing user from one role to another, based on operational needs |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Provides flexibility in managing user access levels as responsibilities change |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | High |

## FR-71: RoleModification.ViewUserRole

|  |  |
| --- | --- |
| **Identifier** | FR-71 |
| **Title** | RoleModification.ViewUserRole |
| **Requirement** | The system shall allow users to view their assigned roles from the account settings, but only administrators can view the roles of other users |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Improves transparency for users regarding their access privileges |
| **Business Rule** | - |
| **Dependencies** | FR-66, FR-9 |
| **Priority** | Medium |

## Use Case Name: Send System Notifications to Users

## FR-72: Notification.SendNotification

|  |  |
| --- | --- |
| **Identifier** | FR-72 |
| **Title** | Notification.SendNotification |
| **Requirement** | The system shall send notifications to users regarding system updates, alerts, or important information based on user preferences |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures users are informed about critical updates and changes in the system |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | High |

## FR-73: Notification.NotificationHistory

|  |  |
| --- | --- |
| **Identifier** | FR-73 |
| **Title** | Notification.NotificationHistory |
| **Requirement** | The system shall maintain a history of sent notifications for each user, allowing them to view past notifications. |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Helps users keep track of important information and alerts they have received |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | Medium |

## FR-74: Notification.AlertForCriticalEvents

|  |  |
| --- | --- |
| **Identifier** | FR-74 |
| **Title** | Notification.AlertForCriticalEvents |
| **Requirement** | The system shall prioritize sending alerts for critical events (e.g., security threats, system failures) to ensure immediate user awareness |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures that users are promptly informed of situations requiring immediate attention |
| **Business Rule** | BR-35: Critical alerts must be sent within 1 minute of the event occurring |
| **Dependencies** | FR-72 |
| **Priority** | High |

## Use Case Name: Alert User on Drone Malfunction

## FR-75: Alert.UserDroneMalfunction

|  |  |
| --- | --- |
| **Identifier** | FR-75 |
| **Title** | Alert.UserDroneMalfunction |
| **Requirement** | The system shall automatically detect and alert users immediately when a drone malfunction occurs, providing details about the nature of the issue |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Ensures users are promptly informed of any malfunctions to take appropriate action |
| **Business Rule** | BR-36: Alerts must be generated within 30 seconds of detecting a malfunction |
| **Dependencies** | - |
| **Priority** | High |

## FR-76: Alert.MalfunctionSeverityAssessment

|  |  |
| --- | --- |
| **Identifier** | FR-76 |
| **Title** | Alert.MalfunctionSeverityAssessment |
| **Requirement** | The system shall assess the severity of the malfunction and categorize alerts as critical, warning, or informational |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Helps users prioritize their responses based on the urgency of the situation |
| **Business Rule** | - |
| **Dependencies** | FR-75 |
| **Priority** | High |

## FR-76: Alert.MalfunctionDetails

|  |  |
| --- | --- |
| **Identifier** | FR-76 |
| **Title** | Alert.MalfunctionDetails |
| **Requirement** | The system shall provide detailed information about the malfunction, including the drone ID, location, and a description of the issue. |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Provides users with the necessary context to understand and respond to the malfunction effectively |
| **Business Rule** | - |
| **Dependencies** | FR-75 |
| **Priority** | High |

## FR-77: Alert.HistoryLog

|  |  |
| --- | --- |
| **Identifier** | FR-77 |
| **Title** | Alert.HistoryLog |
| **Requirement** | The system shall maintain a log of all drone malfunction alerts sent to users, accessible for review and analysis |
| **Source** | Muhammad Usman Malik |
| **Rationale** | Allows users to track past issues and analyze patterns over time for better drone maintenance |
| **Business Rule** | - |
| **Dependencies** | - |
| **Priority** | Medium |

# Non-Functional Requirements

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.

## Reliability

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.

## Usability

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

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

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.

# External Interface Requirements

## User Interfaces Requirements

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.

## Software interfaces

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.

## Hardware interfaces

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.

## Communications interfaces

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.

# References

1. E. Androulaki et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains." *Proceedings of the EuroSys Conference*, vol. 17, pp. 1-15, Apr. 2018. Internet: https://www.eurosys.org/conferences/2023/androulaki.pdf [Sep. 17, 2024].
2. T. Martin et al. "DroneKit: A Python Library for Drones." *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3874-3881. Internet: <https://ieeexplore.ieee.org/document/7759818> [Sep. 17, 2024].
3. T. Martin et al. "DroneKit: A Python Library for Drones." *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3874-3881. Internet: <https://ieeexplore.ieee.org/document/7759818> [Sep. 17, 2024].
4. J. Doe. "Best Practices for Real-Time Video Streaming." Internet: https://www.streamingmedia.com/articles/readarticle.aspx?articleid=153742, Aug. 15, 2023 [Sep. 17, 2024].
5. NIST. "Advanced Encryption Standard (AES)." Internet: <https://csrc.nist.gov/publications/detail/fips/197/final>, May 2, 2001 [Sep. 17, 2024].