

Software Requirements Specification

Skill Development Project III – ICT 3206

**Bachelor of Information and Communication Technology (Honors)**

Department of Information and Communication Technology Faculty of Technology

Rajarata University of Sri Lanka

# Details of the Project

**Project Title** : Intelligent Indoor OR Outdoor Surveillance Camera with AI Detection and Programmable Relay Control

**Group Number** : 01

**Group Name** : Tech Titans

**Submission Date** : 2024/07/10

# Details of the Group Members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Registration ID** | **Index No.** | **Signature** |
| E.M.S.M. Edirisooriya | ITT/2020/021 | 1297 |  |
| D.H.D. Prabhasha | ITT/2020/078 | 1348 |  |
| H.L.I.N. Liyanaarachchi | ITT/2020/128 | 1390 |  |
| P.W.D.I.M. Rodrigo | ITT/2020/091 | 1357 |  |
| B.K. Bandaranayake | ITT/2020/121 | 1383 |  |
| K.R.N. Perera | ITT/2020/072 | 1343 |  |
| A. Sanjayan | ITT/2020/097 | 1363 |  |

**Name** : Ms. Malki Jayawardhana

**Designation** : Lecturer

**Department/ Unit/ Institute** : Department of Information and Communication Technology

**Contact Details** : +94 71 270 9389

# Table of Contents

1. [Introduction 5](#_bookmark0)
   1. [Background of the project 5](#_bookmark1)
   2. [Purpose and significance of the project 5](#_bookmark2)
   3. [Scope of the project 5](#_bookmark3)
2. [Use cases of the project 6](#_bookmark4)
   1. [Actors 6](#_bookmark5)
   2. [Use cases 6](#_bookmark6)
   3. [Use case diagram 7](#_bookmark7)
   4. [Use case scenarios 8](#_bookmark9)
3. [Activity diagrams of the project 15](#_bookmark31)
4. [Functional Requirements of the Project 18](#_bookmark36)
5. [Non-functional Requirements 22](#_bookmark45)
6. [Data Design 24](#_bookmark46)
7. [Hardware Design 24](#_bookmark47)
8. [Recommendation of supervisor(s) on the document 25](#_bookmark50)
9. [Viva presentation assessment team 26](#_bookmark51)
10. [Comments of the assessment team on viva presentation 26](#_bookmark52)

**List of Figures**

[Figure 1 Use case diagram 7](#_bookmark8)

[Figure:2: Activity diagram for App 15](#_bookmark32)

[Figure:3: Activity diagrams for System 16](#_bookmark33)

[Figure 4 Activity diagrams for Object Detection and Provide Alerts 16](#_bookmark34)

[Figure 5 Activity diagrams for Sign-up & Log-in 17](#_bookmark35)

[Figure:6 ER Diagrams 24](#_bookmark48)

[Figure:7 Hardware Design 24](#_bookmark49)

**List of Tables**

[Table 1: Sign up or Register scenario 8](#_bookmark10)

[Table 2: Sign up or Register scenario 8](#_bookmark11)

[Table 3: Sign up or Register scenario 8](#_bookmark12)

[Table 4 :Sign up or Register scenario 9](#_bookmark13)

[Table 5: Sign up or Register scenario 9](#_bookmark14)

[Table 6: View Local Live Stream scenario 9](#_bookmark15)

[Table 7: View Local Live Stream scenario 10](#_bookmark16)

[Table 8: Receive Alerts scenario 10](#_bookmark17)

[Table 9: Receive Alerts scenario 10](#_bookmark18)

[Table 10: Receive Alerts scenario 11](#_bookmark19)

[Table 11: Object Detection and Provide Alerts scenario 11](#_bookmark20)

[Table 12: Object Detection and Provide Alerts scenario 11](#_bookmark21)

[Table 13: View Cloud Live Stream scenario 12](#_bookmark22)

[Table 14: View Cloud Live Stream scenario 12](#_bookmark23)

[Table 15: Get Response from Relay Module scenario 12](#_bookmark24)

[Table 16: Get Response from Relay Module scenario 13](#_bookmark25)

[Table 17:Select Indoor/Outdoor Mode scenario 13](#_bookmark26)

[Table 18: Select Indoor/Outdoor Mode scenario 13](#_bookmark27)

[Table 19: Select Security Parameters Using App scenario 14](#_bookmark28)

[Table 20: Select Security Parameters Using App scenario 14](#_bookmark29)

[Table 21: Select Security Parameters Using App scenario 14](#_bookmark30)

[Table 22 Functional Requirements live-streaming 18](#_bookmark37)

[Table 23 Functional Requirements object detection 18](#_bookmark38)

[Table 24 Functional Requirements cloud-based live streaming 19](#_bookmark39)

[Table 25 Functional Requirements video recording and playback 19](#_bookmark40)

[Table 26 Functional Requirements relay control 20](#_bookmark41)

[Table 27 Functional Requirements mobile application 20](#_bookmark42)

[Table 28 Functional Requirements security alerts and notifications 21](#_bookmark43)

[Table 29 Functional Requirements indoor and outdoor 21](#_bookmark44)

# Introduction

# Background of the project

Home security is becoming increasingly important as technology advances. Traditional CCTV surveillance systems face several challenges, such as high costs, complex installation, limited functionality, and accessibility issues. These older security camera systems often lack the flexibility and advanced features needed to handle today's security challenges effectively.

# Purpose and significance of the project

This project aims to address these issues by developing an intelligent indoor/outdoor surveillance camera system integrated with AI detection and programmable relay control. The system utilizes artificial intelligence, cloud technology, and a mobile app to offer a comprehensive security solution. By combining smart object detection, live video streaming, video recording, and relay control, this system provides excellent monitoring and control capabilities. The development of a mobile app using Flutter ensures that users can easily interact with and manage the system from their phones.

The significance of this project lies in its ability to provide a cost-effective, user-friendly, and scalable solution for enhanced security and monitoring. It aims to overcome the limitations of traditional CCTV systems by leveraging IoT technology, thereby delivering a reliable and user- friendly surveillance solution that meets today’s security needs.

# Scope of the project

The scope of this project encompasses the design, development, and implementation of an AI-powered ESP32-CAM WIFI IP camera surveillance system with advanced functionalities for indoor and outdoor monitoring. The primary components and functionalities to be developed include:

* + 1. **ESP32-CAM WIFI IP Camera**: Design and setup of the ESP32-CAM module, integrating a 2MP camera for video recording.
    2. **AI-Powered Object Detection**: Implementation of the YOLOv8 object detection algorithm to identify and classify objects in real-time, with optimization for efficient and accurate detection.
    3. **Cloud-Based Live Streaming**: Development of a secure cloud-based solution for streaming live video feeds.
    4. **Video Recording and Playback**: Functionality to start and stop video recordings from the mobile application, storage on the ESP32-CAM's SD card or the user's mobile device, and playback features within the mobile application.
    5. **Programmable Physical Relay Control**: Design and implementation of a dual relay module to control external devices (e.g., lights, alarms, locks), integrated within the mobile application for remote management.
    6. **Flutter Mobile Application**: Development of a cross-platform mobile application using Flutter, with user-friendly interfaces for monitoring live video feeds, receiving alerts, and controlling physical relays.
    7. **Security Alerts and Notifications**: Intelligent mechanisms for analyzing events and providing context-aware alerts, customizable by users.

By addressing these components, the project aims to deliver a robust, intelligent, and user- friendly surveillance solution that enhances security and provides convenient remote management capabilities. The successful implementation of this system will offer significant improvements over traditional surveillance systems, making it suitable for residential applications. The system is designed to operate in both indoor and outdoor modes, ensuring optimal performance and accuracy based on the specific environment where the camera is deployed.

# Use cases of the project

# Actors

Primary actors

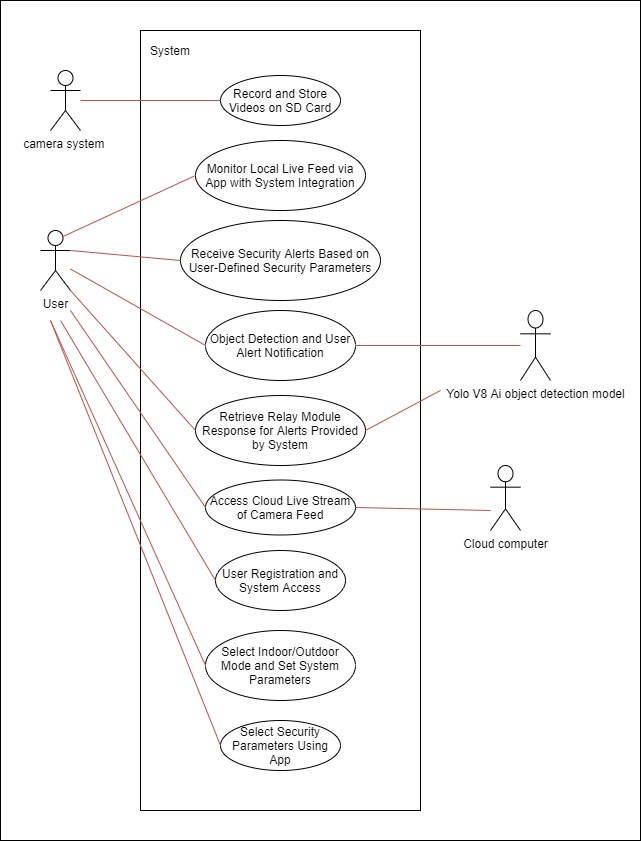
* + 1. User Scondary actors

1. Yolo V8 object detection model.
2. Cloud computer.

# Use cases

* + 1. Sign up or Register
    2. View Local Live Stream
    3. Make Recorded Videos to SD card
    4. Object Detection and provide Alerts
    5. Get response from relay module
    6. View cloud Live Stream
    7. Select indoor Outdoor mode
    8. Select Security Parameters Using App

# Use case diagram



*Figure 1 Use case diagram*

# Use case scenarios

1.

*Table 1: Sign up or Register scenario*

|  |  |
| --- | --- |
| Use case ID | UC01 |
| Use case name | Sign up or Register |
| Actor(s) | User |
| Pre-condition(s) | None |
| Activity descriptions | 1. User navigates to the registration page. 2. User fills in the registration form with necessary details (e.g., username, password, email). 3. User submits the form. 4. System validates the input. 5. System creates a new account and stores user details in the database. 6. System sends a confirmation email to the user (if applicable). |
| Post-condition(s) | User account is created and user is Authenticated |

## 1.E1.

*Table 2: Sign up or Register scenario*

|  |  |
| --- | --- |
| Use case ID | UC01E1 |
| Use case name | Sign up or Register |
| Actor(s) | User |
| Pre-condition(s) | Input validation fails |
| Activity descriptions | 1. System validates the input. 2. System detects invalid input. 3. System displays an error message and prompts user to correct the input. |
| Post-condition(s) | User is informed of the error and prompted to correct it. |

## 1.E2.

*Table 3: Sign up or Register scenario*

|  |  |
| --- | --- |
| Use case ID | UC01E2 |
| Use case name | Sign up or Register |
| Actor(s) | User |
| Pre-condition(s) | Email already exists |
| Activity descriptions | 1. System validates the input. 2. System detects that the email is already in use. 3. System displays an error message and prompts user to use a different email. |
| Post-condition(s) | User is informed of the duplicate email issue and prompted to use a different email. |

## E3.

*Table 4 :Sign up or Register scenario*

|  |  |
| --- | --- |
| Use case ID | UC01E3 |
| Use case name | Sign up or Register |
| Actor(s) | User |
| Pre-condition(s) | Username already exists |
| Activity descriptions | 1. System validates the input. 2. System detects that the username is already in use. 3. System displays an error message and prompts user to use a different username. |
| Post-condition(s) | User is informed of the duplicate username issue and prompted to use a different username. |

## E4.

*Table 5: Sign up or Register scenario*

|  |  |
| --- | --- |
| Use case ID | UC01E4 |
| Use case name | Sign up or Register |
| Actor(s) | User |
| Pre-condition(s) | User provides partial details |
| Activity descriptions | 1. User submits the form with partial details. 2. System prompts user to complete all required fields. 3. User completes the form and submits again. |
| Post-condition(s) | User completes the form and resubmits it. |

1. ​

*Table 6: View Local Live Stream scenario*

|  |  |
| --- | --- |
| Use case ID | UC02 |
| Use case name | View Local Live Stream |
| Actor(s) | User |
| Pre-condition(s) | User is logged in / Camera Connected |
| Activity descriptions | 1. User navigates to the live stream page. 2. User can see camera if available camera. 3. System attempts to establish a connection with the selected camera. 4. User can choose to stop the live stream at any time. 5. System stops the live stream and closes the connection. |
| Post-condition(s) | User views the live stream. |

## E1.

*Table 7: View Local Live Stream scenario*

|  |  |
| --- | --- |
| Use case ID | UC02E1 |
| Use case name | View Local Live Stream |
| Actor(s) | User |
| Pre-condition(s) | Connection fails |
| Activity descriptions | If connection fails, system displays an error message and prompts the user to try again. |
| Post-condition(s) | Display Error message and redirect to Home page |

3.

*Table 8: Receive Alerts scenario*

|  |  |
| --- | --- |
| Use case ID | UC03 |
| Use case name | Receive Alerts |
| Actor(s) | User |
| Pre-condition(s) | User is logged in and notifications are enabled |
| Activity descriptions | 1. User receives a push notification or email alert. 2. User taps on the notification to open the app. 3. System displays the alert details or relevant information. 4. User reads the alert message or information. |
| Post-condition(s) | User is informed about the alert and can take necessary actions. |

## E1

*Table 9: Receive Alerts scenario*

|  |  |
| --- | --- |
| Use case ID | UC03E1 |
| Use case name | Receive Alerts |
| Actor(s) | User |
| Pre-condition(s) | User receives multiple alerts |
| Activity descriptions | 1. User opens the app and sees a list of recent alerts. 2. System allows the user to view all recent alerts. |
| Post-condition(s) | User can view and manage multiple alerts received. |

## E2

*Table 10: Receive Alerts scenario*

|  |  |
| --- | --- |
| Use case ID | UC03E2 |
| Use case name | Receive Alerts |
| Actor(s) | User |
| Pre-condition(s) | User receives no alerts |
| Activity descriptions | 1. If no alerts are received within a specific time period, system displays a message indicating no new alerts. 2. User can refresh to check for new alerts. |
| Post-condition(s) | User is informed about the absence of new alerts. |

4

*Table 11: Object Detection and Provide Alerts scenario*

|  |  |
| --- | --- |
| Use case ID | UC04 |
| Use case name | Object Detection and Provide Alerts |
| Actor(s) | Yolo V8 object detection model. |
| Pre-condition(s) | System is operational and monitoring |
| Activity descriptions | 1. System detects an object using AI object detection. 2. System analyzes the detected object for relevance to alerts. 3. System generates and sends an alert to the user's device. |
| Post-condition(s) | User receives an alert about the detected object. |

## E01

*Table 12: Object Detection and Provide Alerts scenario*

|  |  |
| --- | --- |
| Use case ID | UC04E01 |
| Use case name | Object Detection and Provide Alerts |
| Actor(s) | Yolo V8 object detection model. |
| Pre-condition(s) | No object detected |
| Activity descriptions | 1. If no objects are detected within a monitoring period, system does not send an alert. 2. User continues monitoring or adjusts settings. |
| Post-condition(s) | System does not generate alerts when no objects are detected. |

1. ​

*Table 13: View Cloud Live Stream scenario*

|  |  |
| --- | --- |
| Use case ID | UC05 |
| Use case name | View Cloud Live Stream |
| Actor(s) | User / Cloud Computer |
| Pre-condition(s) | User is authenticated and has access |
| Activity descriptions | 1. User selects the option to view live stream from the cloud. 2. System retrieves live stream from cloud storage. 3. User views the live stream in real-time. |
| Post-condition(s) | User successfully views the live stream. |

## E01

*Table 14: View Cloud Live Stream scenario*

|  |  |
| --- | --- |
| Use case ID | UC05E01 |
| Use case name | View Cloud Live Stream |
| Actor(s) | User / Cloud Computer |
| Pre-condition(s) | Cloud storage is unavailable |
| Activity descriptions | 1. System displays an error message indicating the cloud service is unavailable. 2. User retries or accesses other features. |
| Post-condition(s) | User is informed about the unavailability of the cloud live stream service. |

6.

*Table 15: Get Response from Relay Module scenario*

|  |  |
| --- | --- |
| Use case ID | UC06 |
| Use case name | Get Response from Relay Module |
| Actor(s) | User/Cloud computer |
| Pre-condition(s) | System sends a command to the relay |
| Activity descriptions | 1. System sends a command to the relay module. 2. Relay module processes the command 3. Relay module sends a response back to the system. 4. System receives and processes the response. |
| Post-condition(s) | System successfully receives and processes the response from the relay module. |

## E01

*Table 16: Get Response from Relay Module scenario*

|  |  |
| --- | --- |
| Use case ID | UC06E01 |
| Use case name | Get Response from Relay Module |
| Actor(s) | User/Cloud computer |
| Pre-condition(s) | Relay module is not responding |
| Activity descriptions | 1. System sends a command to the relay module. 2. Relay module fails to respond. 3. System retries the command after a timeout. 4. Relay module still does not respond. 5. System logs an error and informs the user. |
| Post-condition(s) | User is informed about the relay module's unresponsiveness. |

1. ​

*Table 17:Select Indoor/Outdoor Mode scenario*

|  |  |
| --- | --- |
| Use case ID | UC07 |
| Use case name | Select Indoor/Outdoor Mode |
| Actor(s) | User |
| Pre-condition(s) | User changes their mind |
| Activity descriptions | 1. A1. User selects a mode. 2. User decides to change the mode before saving. 3. System allows the user to reselect the mode. 4. User selects the desired mode. 5. System saves the new selection. |
| Post-condition(s) | The user's updated selection is saved. |

## E01.

*Table 18: Select Indoor/Outdoor Mode scenario*

|  |  |
| --- | --- |
| Use case ID | UC07E01 |
| Use case name | Select Indoor/Outdoor Mode |
| Actor(s) | User |
| Pre-condition(s) | System error occurs |
| Activity descriptions | 1. User selects a mode. 2. System encounters an error while saving the settings. 3. System informs the user about the error. 4. User retries or exits the mode selection page. |
| Post-condition(s) | User is informed of the error and can retry or exit. |

8.

*Table 19: Select Security Parameters Using App scenario*

|  |  |
| --- | --- |
| Use case ID | UC08 |
| Use case name | Select Security Parameters Using App |
| Actor(s) | User |
| Pre-condition(s) | User is logged into the app |
| Activity descriptions | 1. User navigates to the security settings page. 2. System displays available security parameters. 3. User selects desired security parameters. 4. System saves the user's selections. 5. System confirms the changes to the user. |
| Post-condition(s) | Security parameters are successfully updated according to user's selection. |

## E01.

*Table 20: Select Security Parameters Using App scenario*

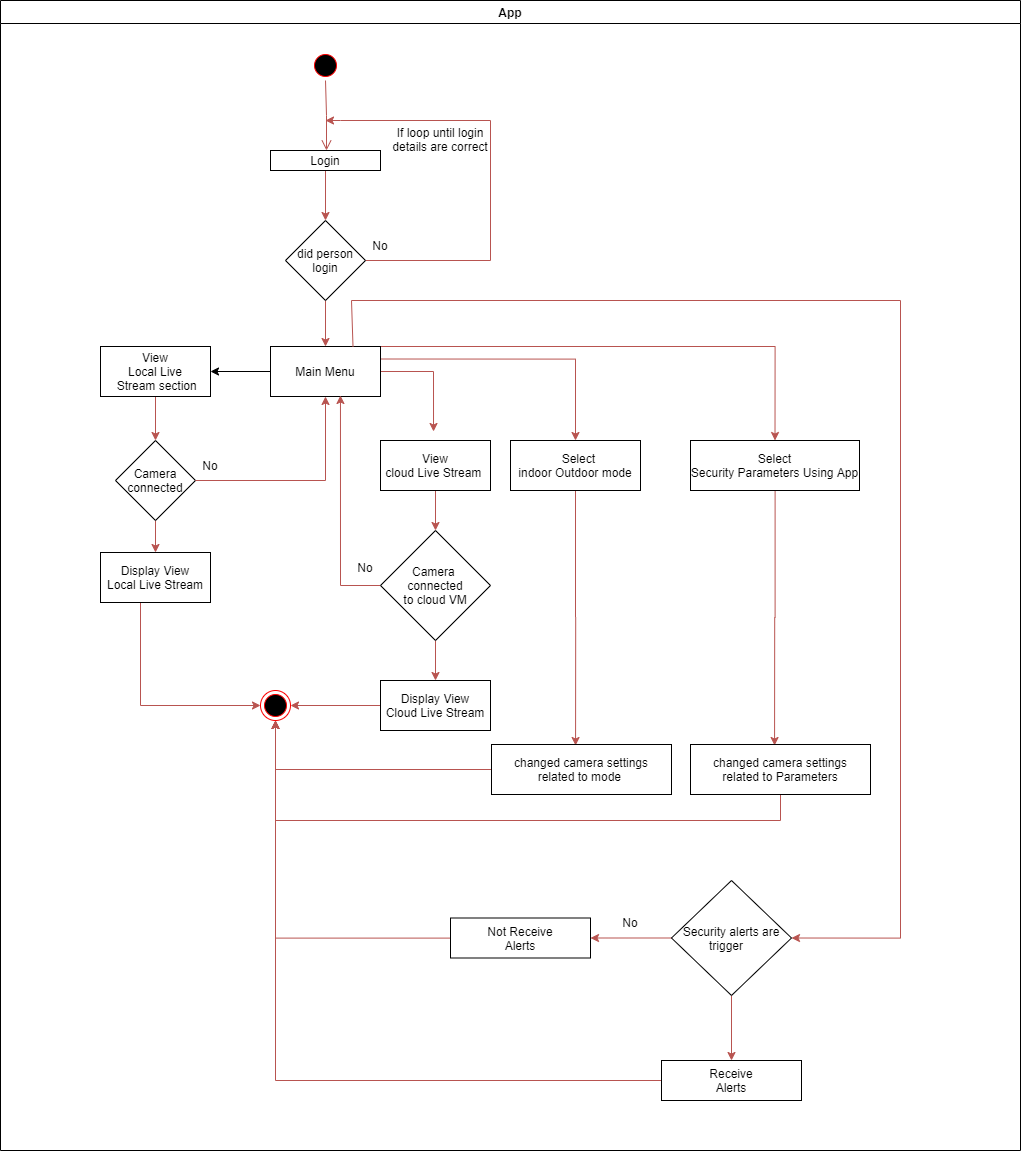
|  |  |
| --- | --- |
| Use case ID | UC08E01 |
| Use case name | Select Security Parameters Using App |
| Actor(s) | User |
| Pre-condition(s) | User selects an invalid parameter |
| Activity descriptions | 1. User selects a parameter. 2. System detects the parameter as invalid. 3. System prompts the user to select a valid parameter. |
| Post-condition(s) | User selects a valid security parameter. |

## 8.E02.

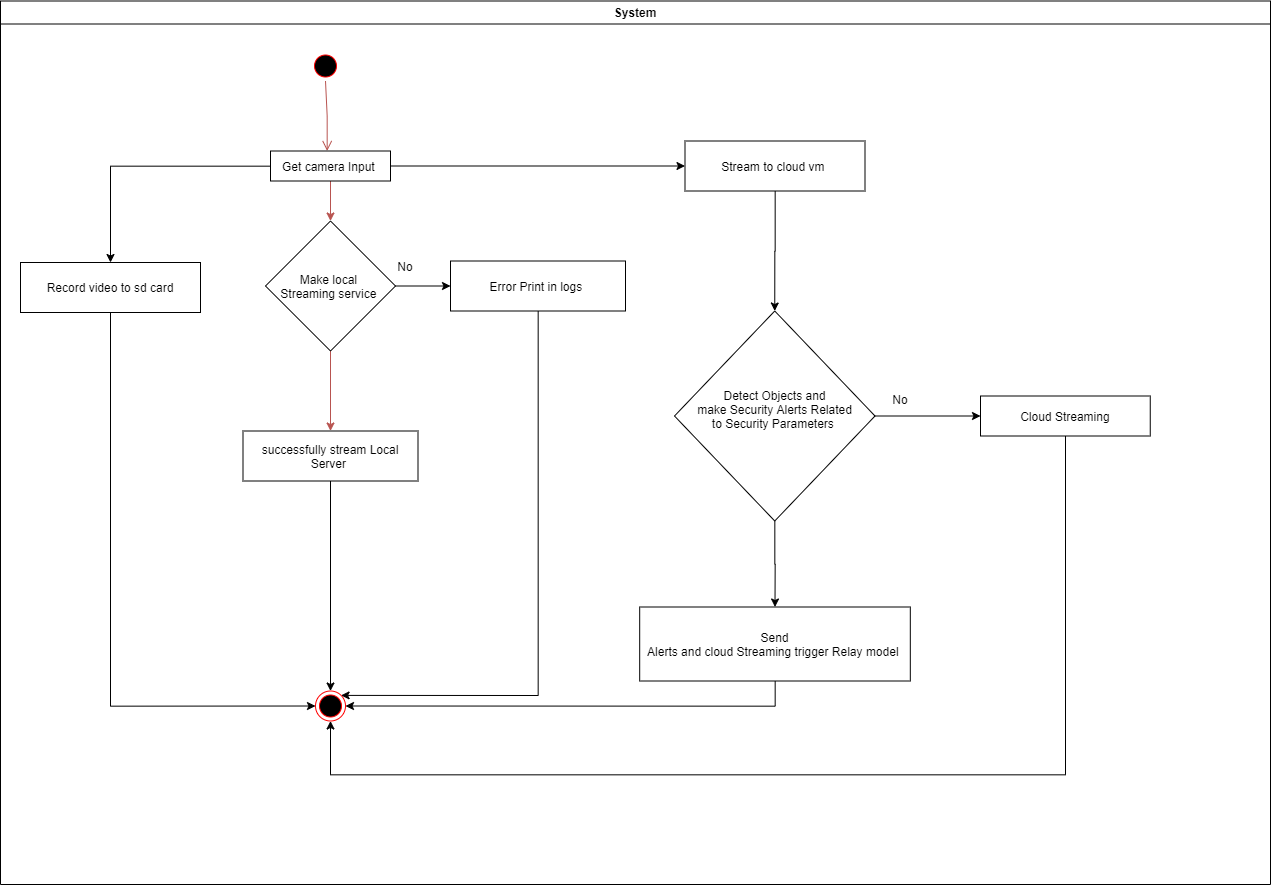
*Table 21: Select Security Parameters Using App scenario*

|  |  |
| --- | --- |
| Use case ID | UC08E02 |
| Use case name | Select Security Parameters Using App |
| Actor(s) | User |
| Pre-condition(s) | System error occurs |
| Activity descriptions | 1. User selects security parameters. 2. System encounters an error while saving the settings 3. System informs the user about the error. 4. User retries or exits the settings page. |
| Post-condition(s) | User is informed of the error and can retry or exit. |

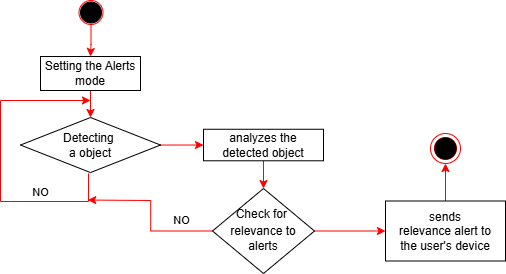
# Activity diagrams of the project



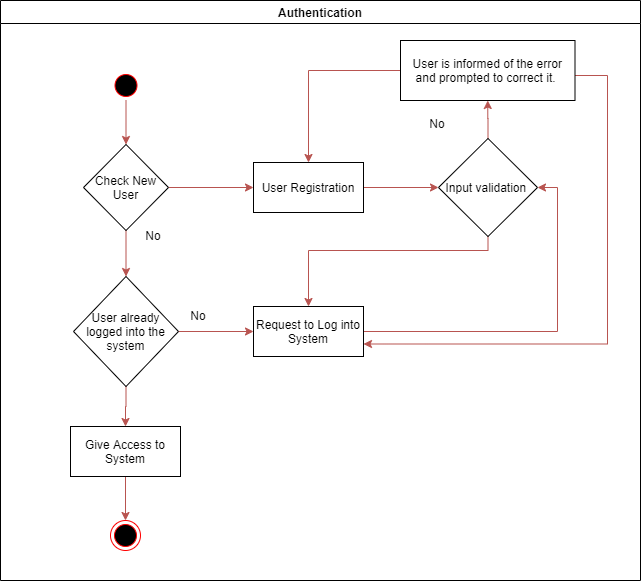
*Figure:2: Activity diagram for App*



*Figure:3: Activity diagrams for System*



*Figure 4 Activity diagrams for Object Detection and Provide Alerts*



*Figure 5 Activity diagrams for Sign-up & Log-in*

# Functional Requirements of the Project

*Table 22 Functional Requirements live-streaming*

|  |  |
| --- | --- |
| Priority Number | 1 |
| Function Name | The system must do local server live video streaming. |
| Description | Allow users to view live video feeds from the camera. |
| Input | Camera feed |
| Process | Stream video feed in real-time to the user's device through the local server. |
| Output | Live video |
| Assumptions/ Constraints | Stable internet connection, compatible device, secure cloud server. |

*Table 23 Functional Requirements object detection*

|  |  |
| --- | --- |
| Priority Number | 2 |
| Function Name | The system must do AI object detection. |
| Description | Detecting and identifying objects using AI. |
| Input | Video frames |
| Process | Analyse video frames using YOLOv8 for object detection. |
| Output | Identified objects |
| Assumptions/ Constraints | Sufficient processing power for real-time analysis, well-trained AI model. |

*Table 24 Functional Requirements cloud-based live streaming*

|  |  |
| --- | --- |
| Priority Number | 3 |
| Function Name | The system must do cloud-based live streaming. |
| Description | Enable remote access to the surveillance cameras live video feed with robust security measures. |
| Input | Camera feed, user credentials |
| Process | Securely stream live video feed to authorised users through a cloud server. |
| Output | Live video on the user’s device |
| Assumptions/ Constraints | Stable internet connection, cloud service availability, secure authentication mechanisms. |

*Table 25 Functional Requirements video recording and playback*

|  |  |
| --- | --- |
| Priority Number | 4 |
| Function Name | The system must do video recording and playback. |
| Description | Allow users to start and stop recordings from the mobile app and review recorded footage. |
| Input | Camera feed, user commands |
| Process | Record video to SD card or cloud storage and provide playback functionality. |
| Output | Video files, playback stream |
| Assumptions/ Constraints | Adequate storage capacity, compatible mobile device, stable internet connection for cloud storage. |

*Table 26 Functional Requirements relay control*

|  |  |
| --- | --- |
| Priority Number | 5 |
| Function Name | The system must do remote physical relay control. |
| Description | Enable users to manage external devices remotely via the mobile app. |
| Input | User commands, camera detection events |
| Process | Control relay modules based on user input or predefined automation rules. |
| Output | Relay state change (on/off) |
| Assumptions/ Constraints | Proper relay module setup, stable internet connection, compatible devices. |

*Table 27 Functional Requirements mobile application*

|  |  |
| --- | --- |
| Priority Number | 6 |
| Function Name | The system must have a Flutter-based mobile application. |
| Description | Provide user-friendly interfaces for monitoring live video, receiving alerts, and controlling devices. |
| Input | User interactions, camera feed, detection alerts |
| Process | Display video feed, send notifications, and manage relay controls through the app. |
| Output | User interface updates, notifications, relay commands |
| Assumptions/ Constraints | Cross-platform compatibility, responsive design, real-time updates. |

*Table 28 Functional Requirements security alerts and notifications*

|  |  |
| --- | --- |
| Priority Number | 7 |
| Function Name | The system must provide security alerts and notifications. |
| Description | Send intelligent alerts based on event analysis and user preferences. |
| Input | Detection events, user preferences |
| Process | Analyse detection events and send context-aware alerts to the user. |
| Output | Notifications (text, email, app alerts) |
| Assumptions/ Constraints | Accurate detection algorithm, customizable alert settings, stable internet connection. |

*Table 29 Functional Requirements indoor and outdoor*

|  |  |
| --- | --- |
| Priority Number | 8 |
| Function Name | The system must support dual mode operation (indoor and outdoor). |
| Description | Allow users to switch between indoor and outdoor modes for optimal performance. |
| Input | User commands |
| Process | Adjust camera settings and detection algorithms based on the selected mode. |
| Output | Mode-specific settings applied |
| Assumptions/ Constraints | Environmental adaptability, user interface for mode selection. |

# Non-functional Requirements

1. Usability Requirements

**Requirement:** The user interface for monitoring and controlling the surveillance system should be intuitive and easy to use.

**Measurement:** Achieve a usability score of at least 85% in user satisfaction surveys conducted with a sample group of target users.

1. Efficiency Requirements

**Requirement:** The system should efficiently process video feeds and detect motion in real- time.

**Measurement:** The time taken from motion detection to alert generation should be less than 1 second.

1. Space Requirement:

**Requirement:** The system should manage storage space efficiently for recorded video footage.

**Measurement:** Implement video compression algorithms to ensure that the system can store at least 30 days of footage within 1 TB of storage.

1. Reliability Requirement:

**Requirement:** The system should operate continuously and reliably without downtime.

**Measurement:** Achieve an uptime of 99.9% over a year.

1. Delivery Requirements

**Requirement:** The system should be delivered within the agreed project timeline.

**Measurement:** Complete the project with no more than a 5% deviation from the planned schedule.

1. Implementation Requirements

**Requirement:** The system should utilize open-source software components where feasible to reduce costs and promote transparency.

**Measurement:** At least 70% of the software stack should consist of open-source components.

1. Standard Requirements

**Requirement:** The system should comply with industry standards for surveillance systems, such as ONVIF for IP-based security products.

**Measurement:** Conduct tests to verify compliance with ONVIF standards and document the results.

1. Interoperability Requirements

**Requirement:** The system should integrate seamlessly with existing security and surveillance infrastructure.

**Measurement:** Test and confirm interoperability with at least three different existing security systems or platforms.

1. Ethical Requirements

**Requirement:** The system should ensure user privacy and data protection.

**Measurement:** Implement mechanisms for anonymizing faces in stored footage and conduct regular privacy audits to ensure compliance with privacy policies.

1. Privacy Requirements:

**Requirement:** The system must comply with relevant data protection laws and regulations, such as the General Data Protection Regulation (GDPR).

**Measurement:** Ensure data handling practices are compliant with GDPR, including user consent for data collection and the implementation of data protection measures.

1. Safety Requirements:

**Requirement:** The system should be safe to use and should not pose any harm to users or the environment.

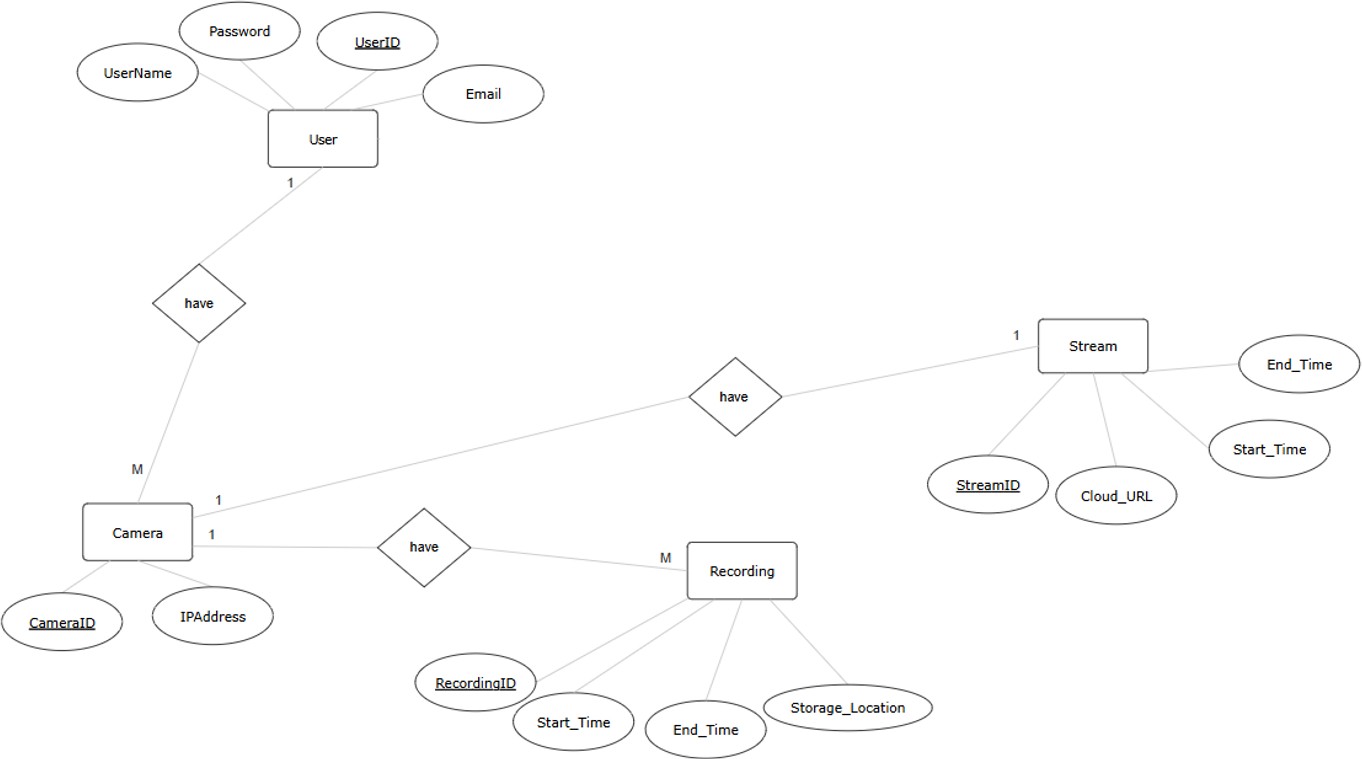
**Measurement:** Conduct safety assessments and obtain necessary certifications to confirm the system's safety.

1. Security Requirements:

**Requirement:** The system should be secure against unauthorized access and data breaches.

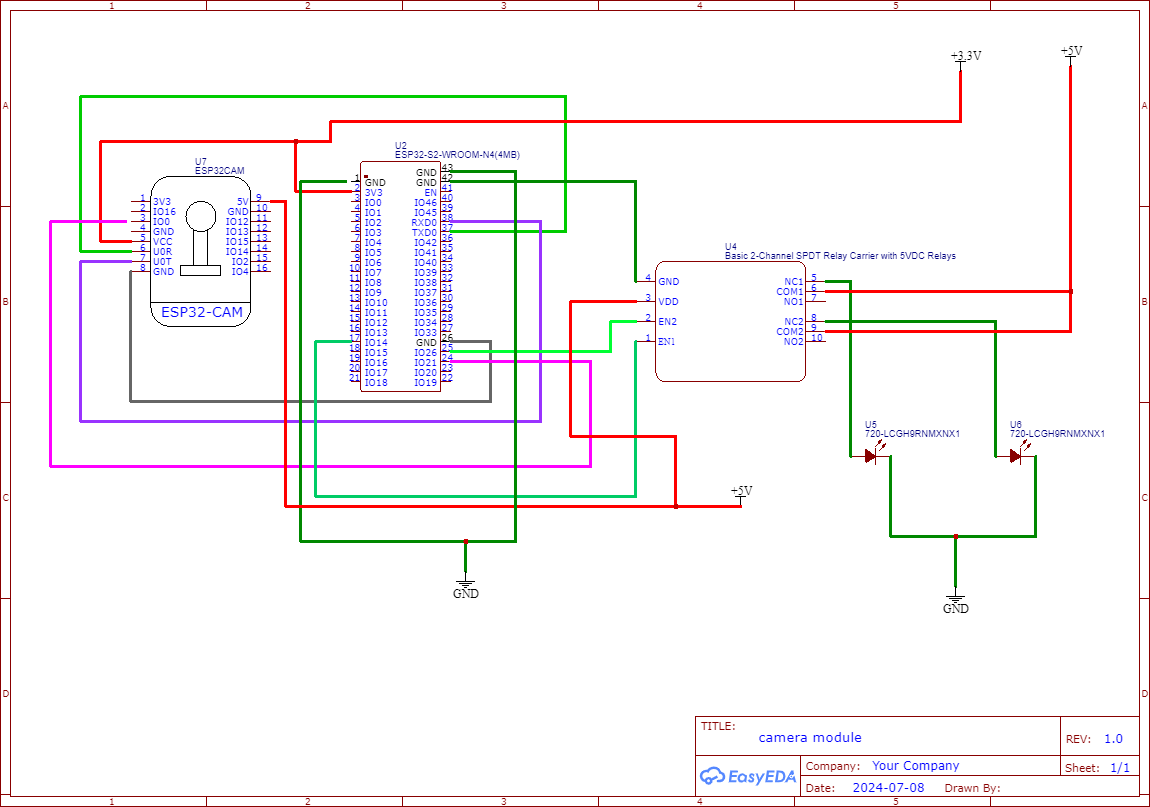
**Measurement:** Implement encryption, authentication, and regular security audits to safeguard the system against potential threats.

# Data Design



# Hardware Design

*Figure:6 ER Diagrams*



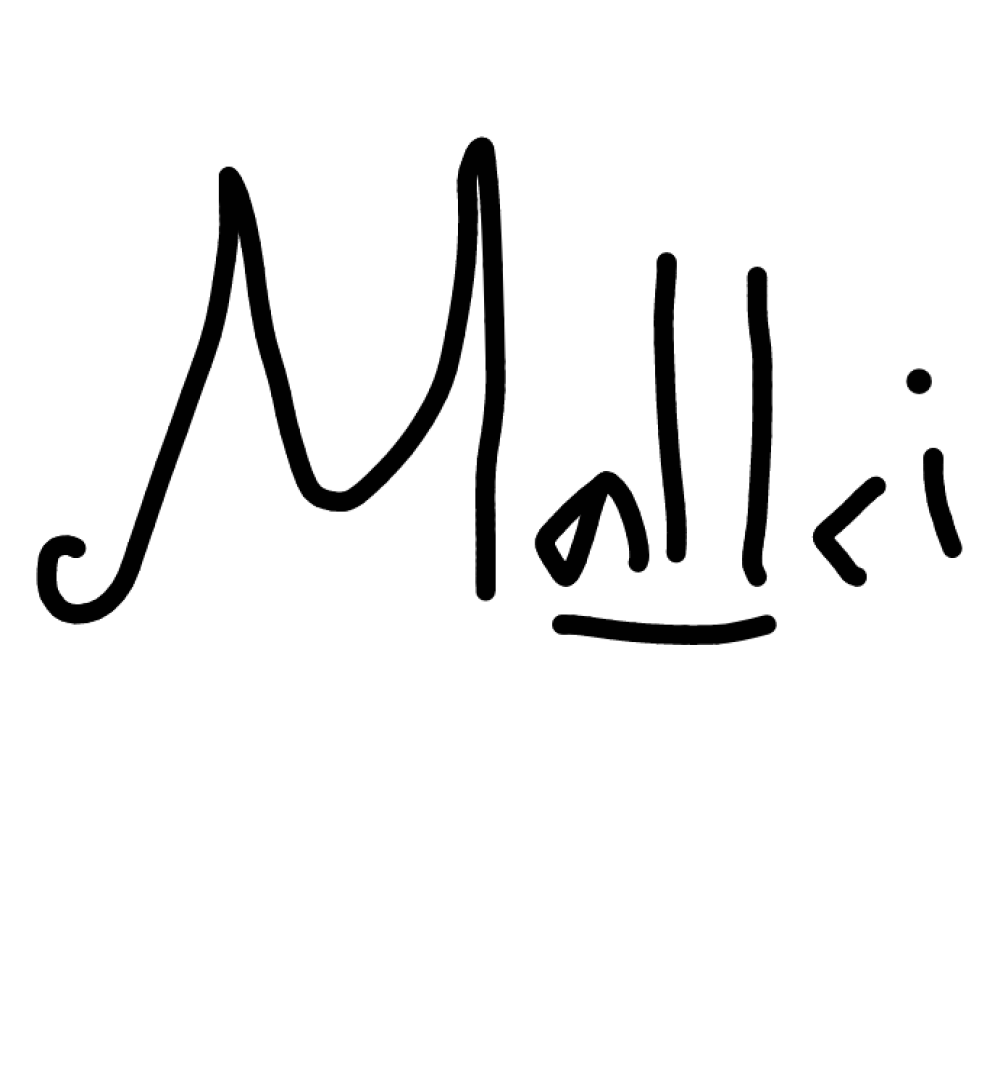
*Figure:7 Hardware Design*

# Recommendation of supervisor(s) on the document

***(This section should be filled by the supervisor(s)).***

# Comments (if any):

Document was recieved on 10.07.2024 and some modifications on the use case and activity diagrams were suggested

**I/We certify that, the student engaged continuously with me in developing the proposal and, I am confident that they are adequately competent to defend this viva.**

**Signature(s) of Supervisor(s):**

**Date:**

12.07.2024

# Viva presentation assessment team

***(This section should be filled by the department)***

# Date of viva presentation:

|  |  |  |
| --- | --- | --- |
| **Panel members** | **Name** | **Department / Institute** |
| **Chair** |  |  |
| **Member** |  |  |
| **Member** |  |  |
| **Member** |  |  |
| **Member** |  |  |

# Comments of the assessment team on viva presentation

***(This should be filled by the chair of the assessment panel. In case of revision or fail, needed revision or reasons to fail the viva presentation should be mentioned here)***

|  |  |
| --- | --- |
| **Result of the viva presentation** | **Excellent / Good / Pass with revisions / Fail** |
| **Score** |  |
| **Signature of the panel chair** |  |
| **Date** |  |