Final Report

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

(*Leave this page blank*)

Details of the Project

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

Group Number : Group 01

Group Name : Tech Titans

Submission Date : 2025/01/17

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	Dumindu.
H.L.I.N. Liyanaarachchi	ITT/2020/128	1390	
P.W.D.I.M. Rodrigo	ITT/2020/091	1357	
B.M.K.L. Bandaranayake	ITT/2020/121	1383	Niranga
K.R.N. Perera	ITT/2020/072	1343	Romes
A. Sanjayan	ITT/2020/097	1363	

Name : Ms. P.R.H.N.G. Thilakarathne

Designation : Lecturer

Department/ Unit/ Institute : Information & Communication Technology

Contact Details : +94 71 954 5073

Table of Contents

1 Introduction 1

1.1 Background of the project 1

1.2 Purpose and significance of the project 1

1.3 Scope of the project 1

1.4 Aim and objectives of the project 2

1.4.1 Aim 2

1.4.2 Objectives 2

1.5 System design approach 2

1.5.1 Software Development Process Model 2

1.5.2 Design Patterns 3

1.6 Project Work Plan 3

2 System Requirement 4

2.1 Functional Requirements of the Project 4

2.2 Non-functional Requirements 8

3 System Design 10

3.1 Architectural Design 10

3.1.1 System Architecture 10

3.1.2 Component design 11

3.1.3 Processes and interaction design 13

3.1.4 Tools, libraries, special algorithms and implementation environment 20

3.2 Interface Design 23

3.2.1 PACT (People, Activities, Contexts, Technologies) analysis of the system 23

3.2.2 Interfaces (software/hardware) of the system 25

6.1.1 Design tools, techniques, templates 28

6.2 Data Management 28

6.2.1 Design tools, techniques 28

6.2.2 Conceptual database design 29

6.2.3 Logical database design and schema refinement 31

6.2.4 Physical database design 32

6.3 Hardware Design (if available) 33

7 Testing and evaluation 35

7.1 Testing plan 35

8 Conclusion 39

8.1 Conclusions of the project 39
8.2 Lessons learned and skills earned 40
8.3 Recommendations for further improvements 40
9 References 41
10 Recommendation of supervisor(s) on the final Report 42
11 Viva presentation assessment team 43
12 Comments of the assessment team on viva presentation 43

List of Figures

Figure 1 3
Figure 2 11
Figure 3 12
Figure 4 13
Figure 5 14
Figure 6 15
Figure 7 15
Figure 8 16
Figure 9 17
Figure 10 18
Figure 11 19
Figure 12 25
Figure 13 25
Figure 14 26
Figure 15 26
Figure 16 26
Figure 17 27
Figure 18 27
Figure 19 30
Figure 20 31
Figure 21 31
Figure 22 32
Figure 23 33
Figure 24 34
Figure 25 34

List of Tables

Table 1 4
Table 2 4
Table 3 5
Table 4 5
Table 5 6
Table 6 6
Table 7 7
Table 8 7
Table 9 36
Table 10 37
Table 11 38
Table 12 39

1 Introduction

3.1 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.

3.2 Purpose and significance of the project

This project aims to address above mention 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.

3.3 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 storage recording on the cloud , and playback .
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.

3.4 Aim and objectives of the project

1.4.1 Aim

The aim of this project is to develop an intelligent indoor/outdoor surveillance camera system integrated with AI detection and programmable relay control, addressing the limitations of traditional CCTV systems. By leveraging IoT technology, this system aims to provide a cost-effective, user-friendly, and scalable solution for enhanced security and monitoring.

1.4.2 Objectives

1. Developing the ESP32-CAM WIFI IP Camera to integrate seamlessly with live streaming, recording, and relay control functionalities of the surveillance system.
2. Integrating AI-Powered Object Detection using YOLOv8 for accurate identification and classification of objects in real-time.
3. Enable Cloud-Based Live Streaming for remote access to the surveillance camera's live video feed with robust security measures.
4. Implementing Video Recording Functionality allowing users to review recordings from the mobile app.
5. Implementing Remote Physical Relay Control enabling users to manage external devices (e.g., lights, alarms) remotely via the mobile app.
6. Developing a Flutter-based Mobile Application with user-friendly interfaces for monitoring live video, receiving security alerts, and controlling connected devices.
7. Enhance Security Alerts and Notifications with intelligent mechanisms for analyzing events and providing context-aware alerts customizable by users.

3.5 System design approach

1.5.1 Software Development Process Model

For our "Intelligent Indoor/Outdoor Surveillance Camera with AI Detection and Programmable Relay Control" project, we use **SCRUM** models due to the iterative and flexible nature of development:

- **SCRUM**: SCRUM is an Agile framework with defined roles such as Product Owner, SCRUM Master, and Development Team. It uses sprints (2-4 weeks) to deliver incremental features. Regular meetings (stand-ups, sprint reviews) keep the team aligned and allow progress monitoring, as we are having meetings with our team and mentor.

1.5.2 Design Patterns

For our "Intelligent Indoor/Outdoor Surveillance Camera with AI Detection and Programmable Relay Control" project, we the design architecture of your system, you might use:

- **Model-View-Controller (MVC)**: This pattern is suitable for our mobile app. The **Model** represents the logic (e.g., AI detection and relay control), the **View** handles the UI (e.g., monitoring video, controlling devices), and the **Controller** manages the flow between them. This separation of concerns improves maintainability.
- **Client-Server Architecture**: Since our system involves live streaming and cloud-based access, a client-server model is apt. The **ESP32-CAM** (camera) serves as the server, streaming video to the mobile app (client) over the internet, with remote control features managed via the server. This design pattern supports scalable, secure remote surveillance.

3.6 Project Work Plan

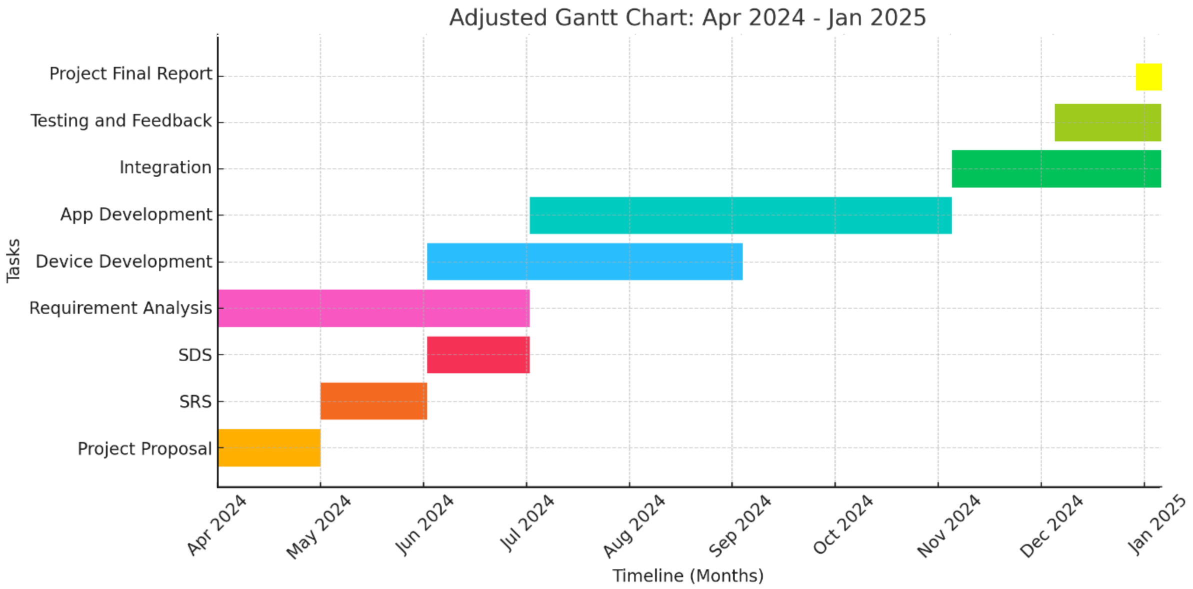


Figure 1

2 System Requirement

3.1 Functional Requirements of the Project

Priority Number	1
Function Name	The system must do 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 cloud server.
Output	Live video
Assumptions/ Constraints	Stable internet connection, compatible device, secure cloud server.

Table 1

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

Table 2

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 3

Priority Number	4
Function Name	The system must do video recording and playback.
Description	Allow users to review recorded footage.
Input	Camera feed, user commands
Process	Record video to 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 4

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 5

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 6

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 7

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.

Table 8

3.2 Non-functional Requirements

➢ 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.

➢ 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.

➢ Performance Requirements

➢ 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.

➢ Reliability Requirement:

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

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

➢ Portability Requirements

- Requirement: The system should be deployable on various hardware platforms without requiring significant modifications.

- Measurement: Successfully deploy and run the system on at least three different hardware configurations, including different camera models and processing units.

- 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.
- 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.
- 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.
- 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.
- 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.
- Legislative Requirements
- 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.

➢ 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.

➢ 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.

3 System Design

3.1 Architectural Design

3.1.1 System Architecture

The system architecture for the surveillance system is based on an integration of hardware and software components that function together to provide an intelligent, real-time security solution.

Overview:

The system includes the following key components

- **ESP32-CAM Module**: A low-power microcontroller with an integrated 2MP camera for video capture.

- **YOLOv8 Object Detection Model**: Embedded AI model used to identify and classify objects in real time.

- **Cloud Server**: Handles storage, cloud-based live streaming, and access control.

- **Mobile Application (Flutter)**: Allows users to interact with the system, view live feeds, control devices, and receive alerts.

Relay Module: Controls external devices such as lights or alarms based on input from the AI system or user.

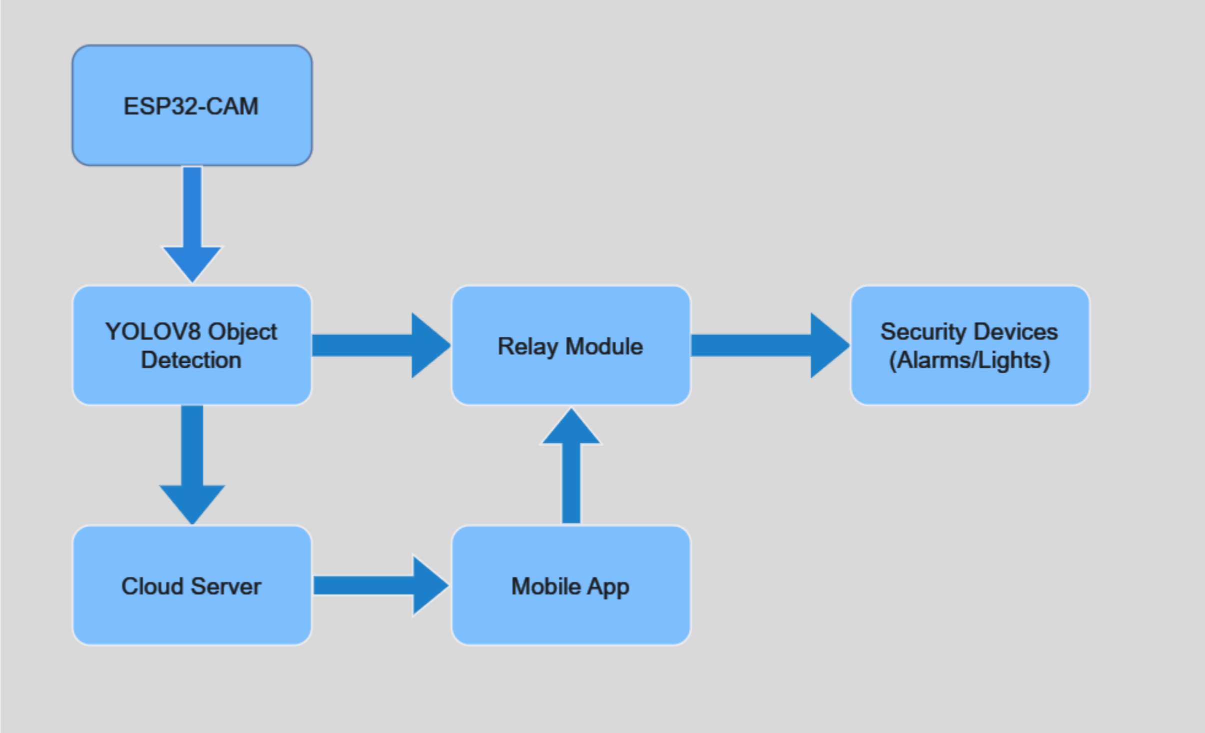


Figure 2

3.1.2 Component design

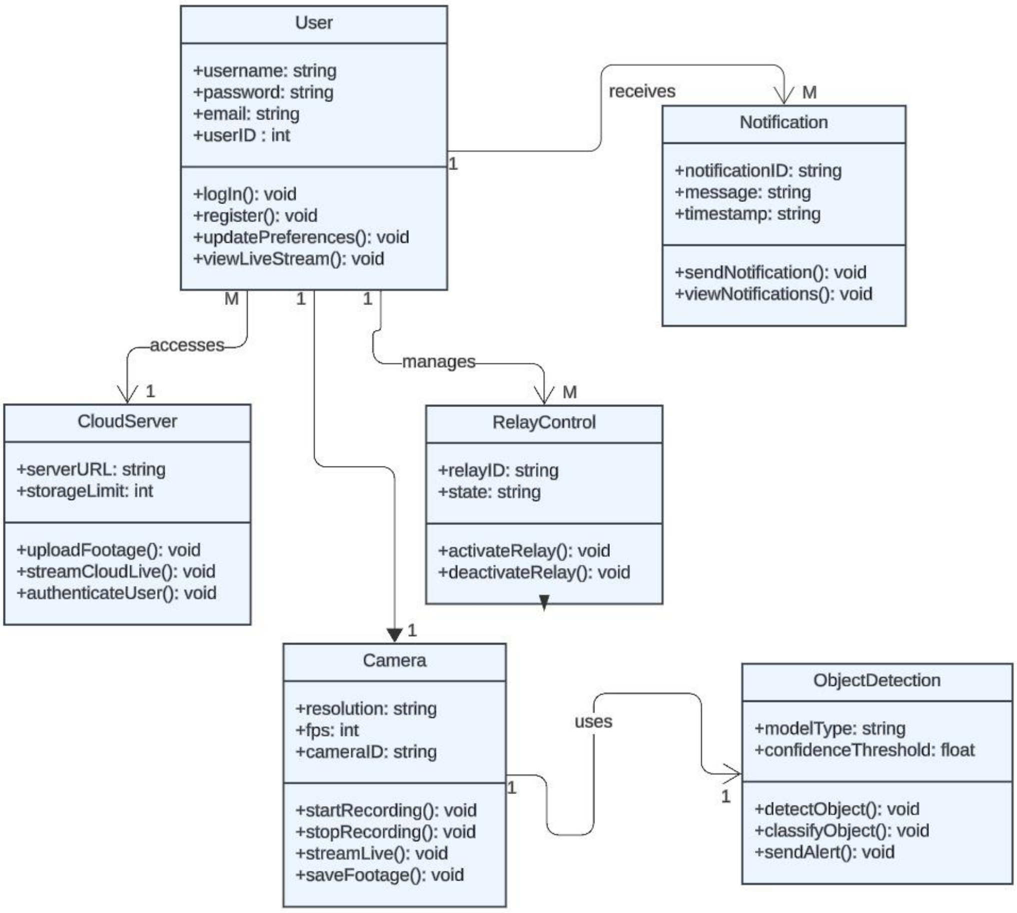


Figure 3

Description*:*

- The **Camera Class** handles video feed management, enabling live streaming or saving footage based on user commands.
- The **Object Detection Class** integrates the YOLOv8 AI model, running detection algorithms on captured video frames and sending alerts if an object is recognized.
- The **User Class** manages user authentication and settings, allowing users to access live streams, recorded footage, and security preferences.
- The **Relay Control Class** facilitates the management of external security devices such as lights or alarms, allowing remote activation or deactivation.
- The **Cloud Server Class** is responsible for storing video recordings and enabling cloud-based live streaming, along with secure user access.
- The **Notification Class** handles alert generation, enabling the system to send push notifications to the user's mobile app when a detection event occurs.

3.1.3 Processes and interaction design

1.

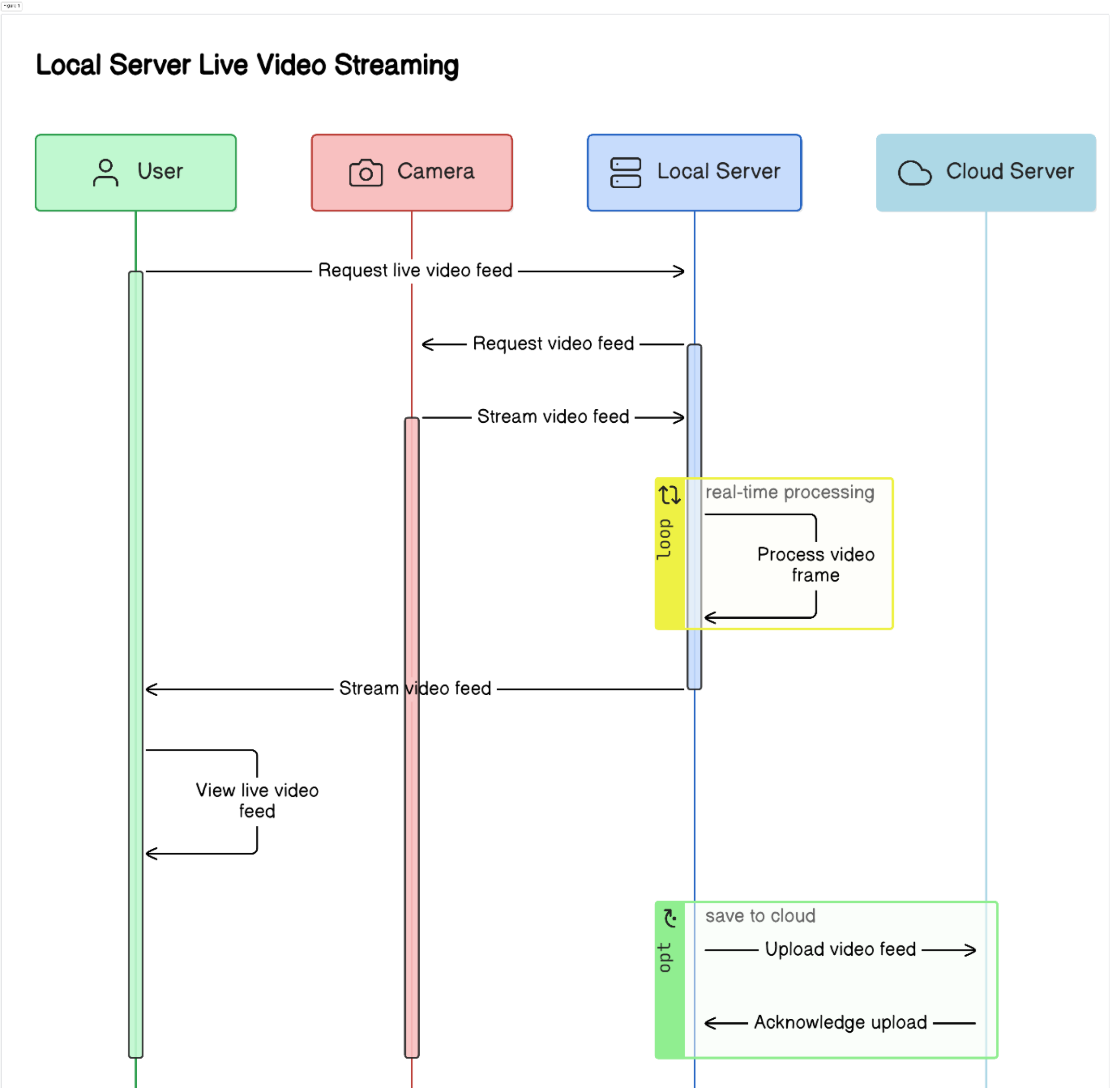


Figure 4

2.

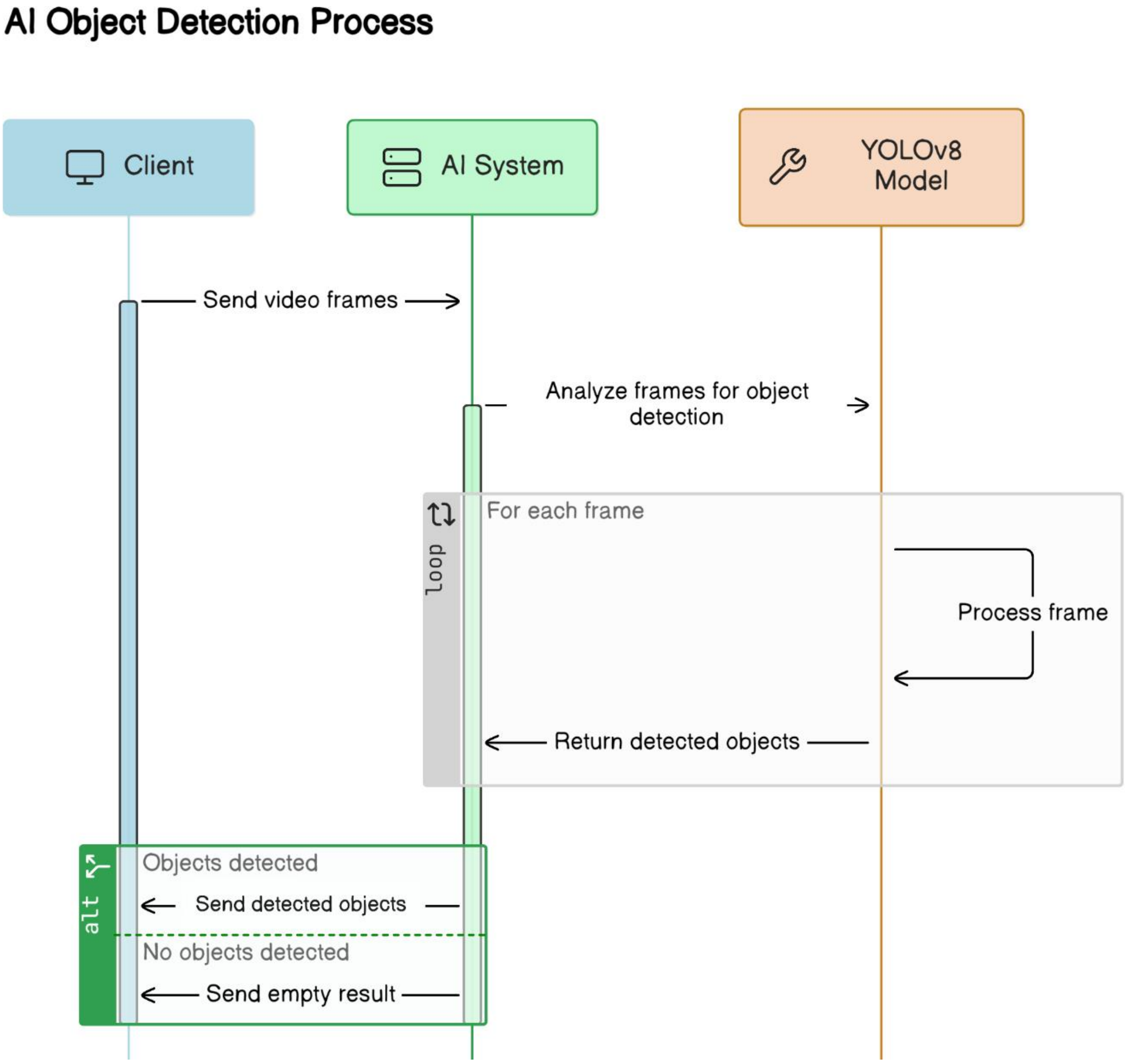

AI Object Detection Process
Client
AI System
YOLOv8 Model
Send video frames
Analyze frames for object detection
loop
For each frame
Process frame
Return detected objects
alt
Objects detected
Send detected objects
No objects detected
Send empty result

Figure 5

3

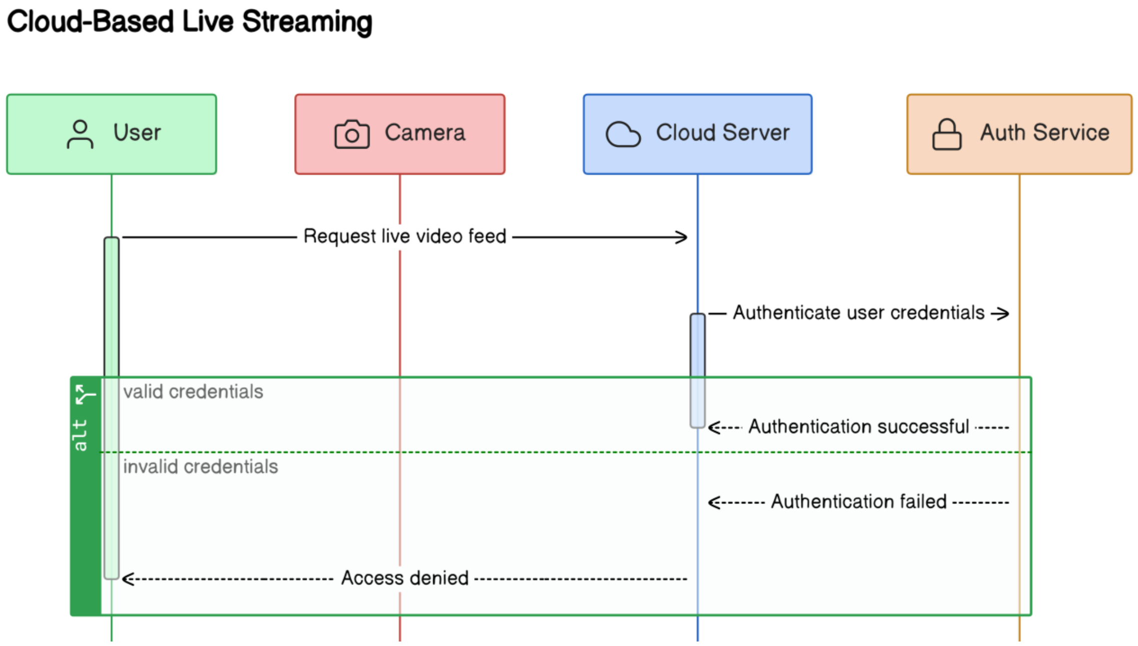


Figure 6

4.

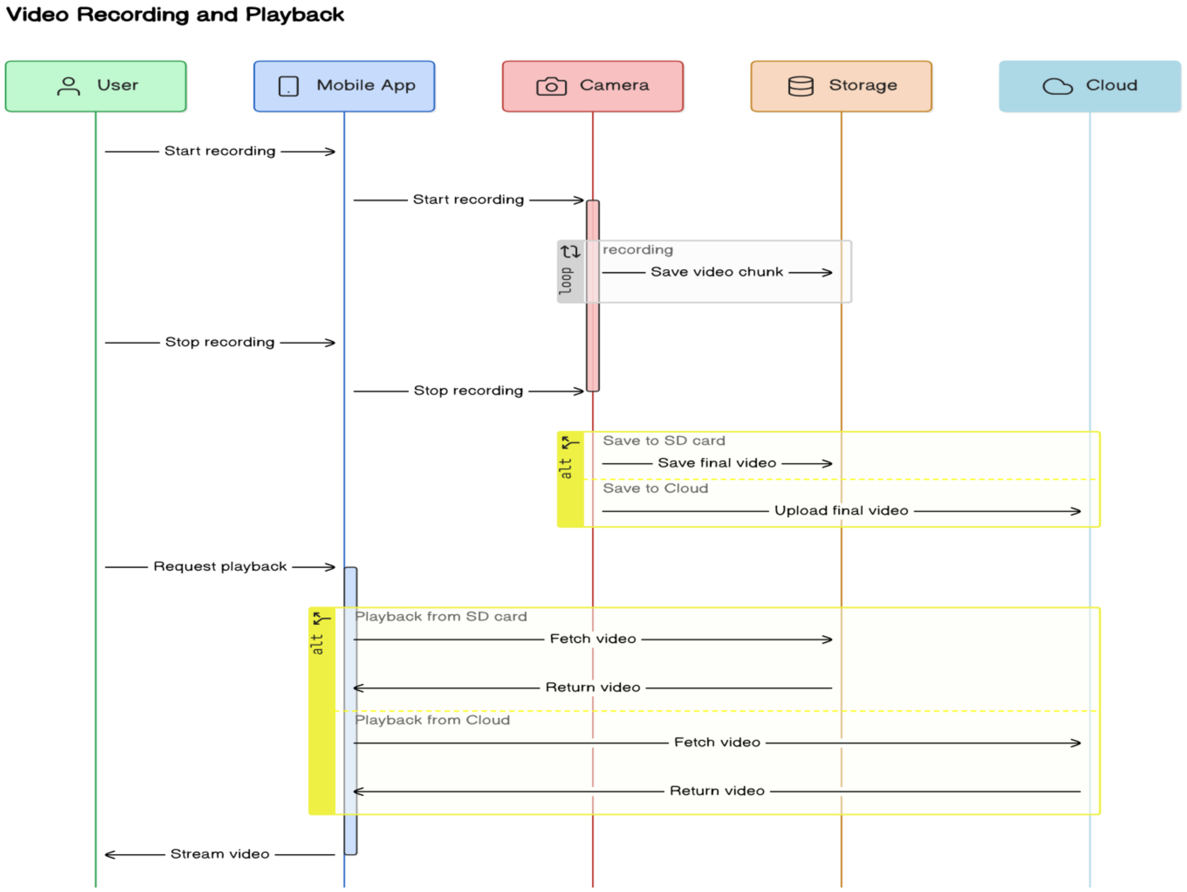


Figure 7

5.

Remote Physical Relay Control

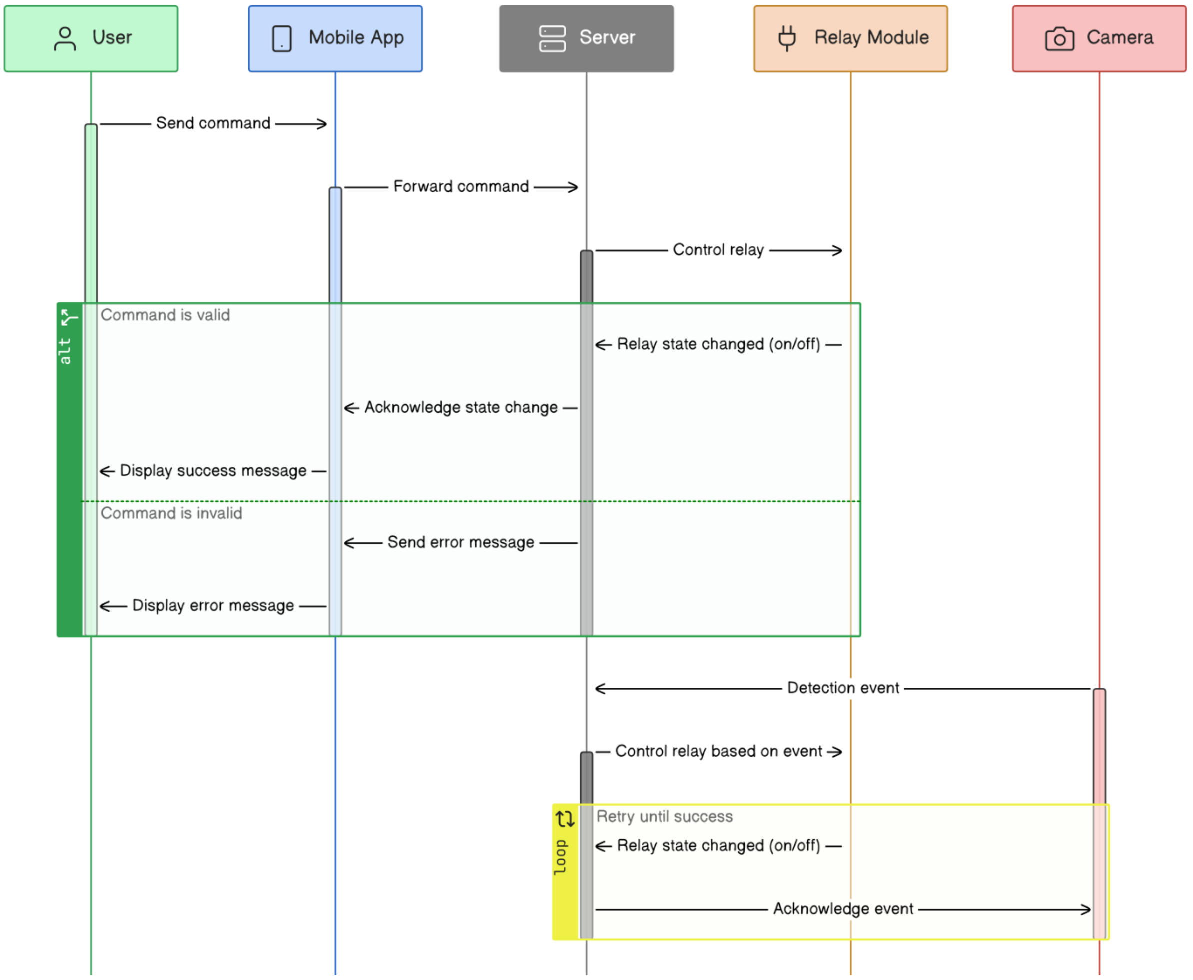


Figure 8

6.

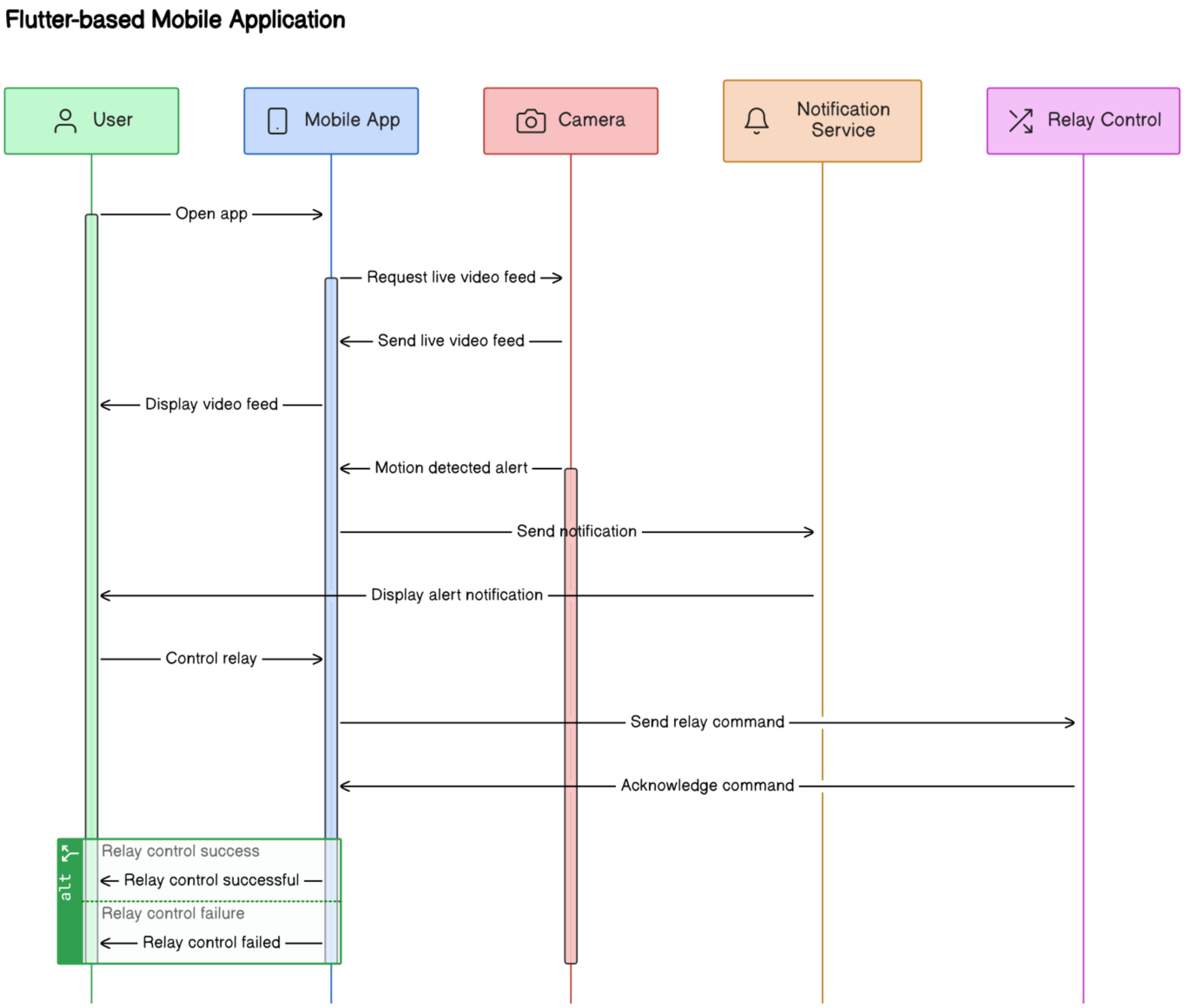

Flutter-based Mobile Application
User
Mobile App
Camera
Notification Service
Relay Control
Open app
Request live video feed
Send live video feed
Display video feed
Motion detected alert
Send notification
Display alert notification
Control relay
Send relay command
Acknowledge command
alt
Relay control success
Relay control successful
Relay control failure
Relay control failed

Figure 9

7.

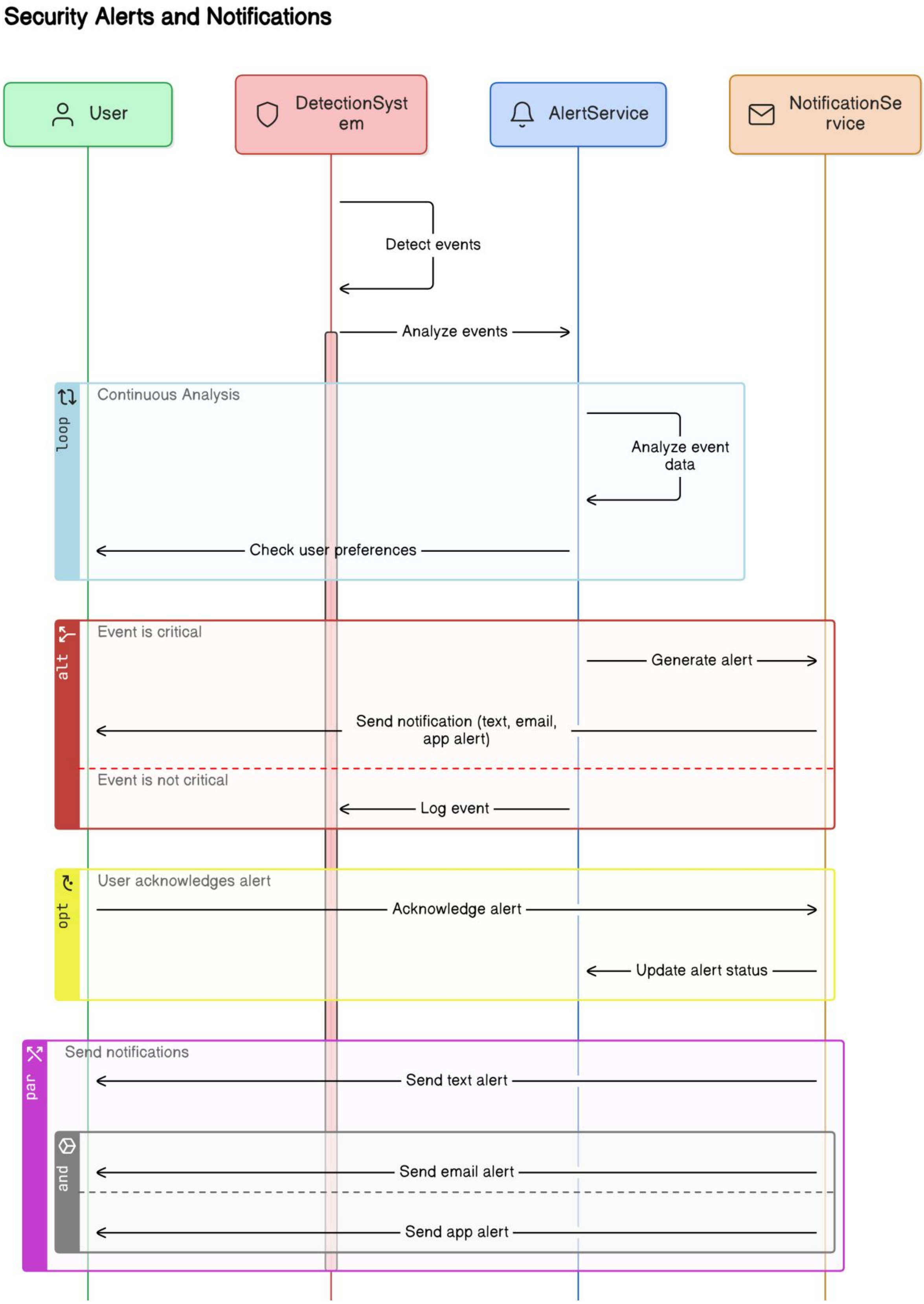

Security Alerts and Notifications
User
DetectionSystem
AlertService
NotificationService
Detect events
Analyze events
loop
Continuous Analysis
Analyze event data
Check user preferences
alt
Event is critical
Generate alert
Send notification (text, email, app alert)
Event is not critical
Log event
opt
User acknowledges alert
Acknowledge alert
Update alert status
par
Send notifications
Send text alert
and
Send email alert
Send app alert

Figure 10

8.

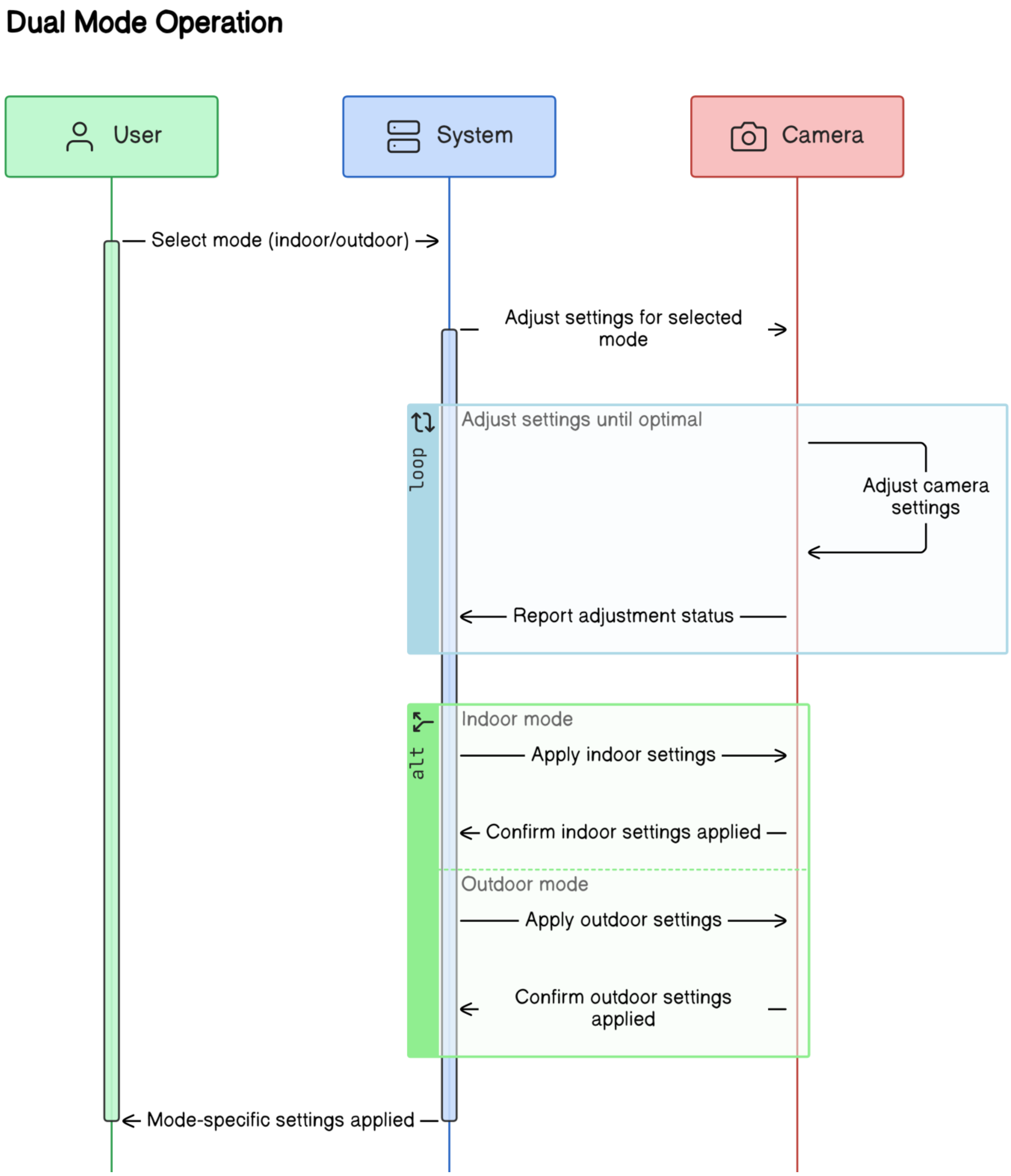

Dual Mode Operation
User
System
Camera
Select mode (indoor/outdoor)
Adjust settings for selected mode
loop
Adjust settings until optimal
Adjust camera settings
Report adjustment status
alt
Indoor mode
Apply indoor settings
Confirm indoor settings applied
Outdoor mode
Apply outdoor settings
Confirm outdoor settings applied
Mode-specific settings applied

Figure 11

3.1.4 Tools, libraries, special algorithms and implementation environment

ESP32-CAM WiFi IP Camera

- Hardware: ESP32-CAM Module (Espressif Systems)
  Reference: Espressif Official Documentation
- Firmware: Custom firmware
  Reference: ESP32-CAM Projects Guide
- **Libraries:**
  - ESP-IDF: Espressif IoT Development Framework
    Reference: ESP-IDF Documentation
  - Arduino IDE: Integrated Development Environment
    Reference: Arduino IDE

AI-Powered Object Detection

- Algorithm: YOLOv8 (You Only Look Once, version 8)
  Reference: YOLO Official Documentation
- **Libraries**
  - OpenCV: Computer vision library
    Reference: OpenCV Documentation
  - PyTorch: Deep learning framework
    Reference: PyTorch Documentation
  - Darknet: YOLO implementation
    Reference: Darknet GitHub
- **Tools**
  - Google Colab: Online notebook for running AI models
    Reference: Google Colab

Cloud-Based Live Streaming

- **Services:**
  - MUX Cloud infrastructure
    Reference: MUX Documentation
  - Firebase: Real-time database and authentication service
    Reference: Firebase Documentation
- **Libraries:**
  - FFmpeg: Multimedia framework for streaming
    Reference: FFmpeg Documentation
  - WebRTC: Real-time communication protocol
    Reference: WebRTC Documentation

Video Recording and Playback

- **Libraries**
  - FFmpeg: For video encoding and decoding
    Reference: FFmpeg Documentation
- **Tools**
  - Node.js: Backend service development
    Reference: Node.js Documentation

Programmable Physical Relay Control

- Hardware: Relay modules
  Reference: Relay Module Documentation
- **Libraries**
  - ESP32 GPIO Library: For controlling GPIO pins
    Reference: ESP32 GPIO Documentation
- **Tools**
  - Arduino IDE:
    Reference: Arduino IDE

Flutter Mobile Application

- Framework: Flutter
  Reference: Flutter Documentation
- **Libraries**
  - Flutter Blue: Bluetooth communication
    Reference: Flutter Blue Documentation
  - HTTP: API interaction library
    Reference: Flutter HTTP Documentation
- **Tools**
  - Android Studio: IDE
    Reference: Android Studio Documentation
  - Visual Studio Code: IDE
    Reference: VS Code Documentation

Security Alerts and Notifications

- **Libraries**
  - Firebase Cloud Messaging (FCM): Push notifications
    Reference: FCM Documentation
- **Tools**
  - Python: Backend development
    Reference: Python Documentation

Implementation Environment

- **Development Environment**
  - Local: For ESP32-CAM development and testing.
  - Cloud: AWS, Goggle or another cloud provider for hosting live streaming and data services.
- **Testing Environment**
  - Simulated: Use virtual environments and emulators for mobile app testing.
  - Physical: Deploy cameras and test in real indoor and outdoor scenarios.
- **Version Control**
  - Git: For source code management.
  - GitHub or GitLab: For repository hosting and collaboration.

3.2 Interface Design

3.2.1 PACT (People, Activities, Contexts, Technologies) analysis of the system

People

The primary users of the "Intelligent Indoor/Outdoor Surveillance Camera with AI Detection and Programmable Relay Control" system are homeowners, security personnel, and business owners. These users require an easy-to-use interface for real-time monitoring and control over surveillance feeds. Since the system is operated through a mobile app, the user interface must cater to different levels of technical proficiency. While homeowners may use the system for general security, security personnel might need more advanced functionalities like detailed notifications and remote device control.

Activities

The main activities include:

- **Monitoring live video feeds** from indoor and outdoor cameras.
- **Controlling external devices** such as alarms or lights through relay systems.
- **Receiving and managing security alerts**, such as notifications triggered by AI object detection.
- **Accessing recorded footage** from the mobile app, as well as starting and stopping video recordings. These activities ensure the user can respond quickly to threats and maintain security at all times.

Contexts

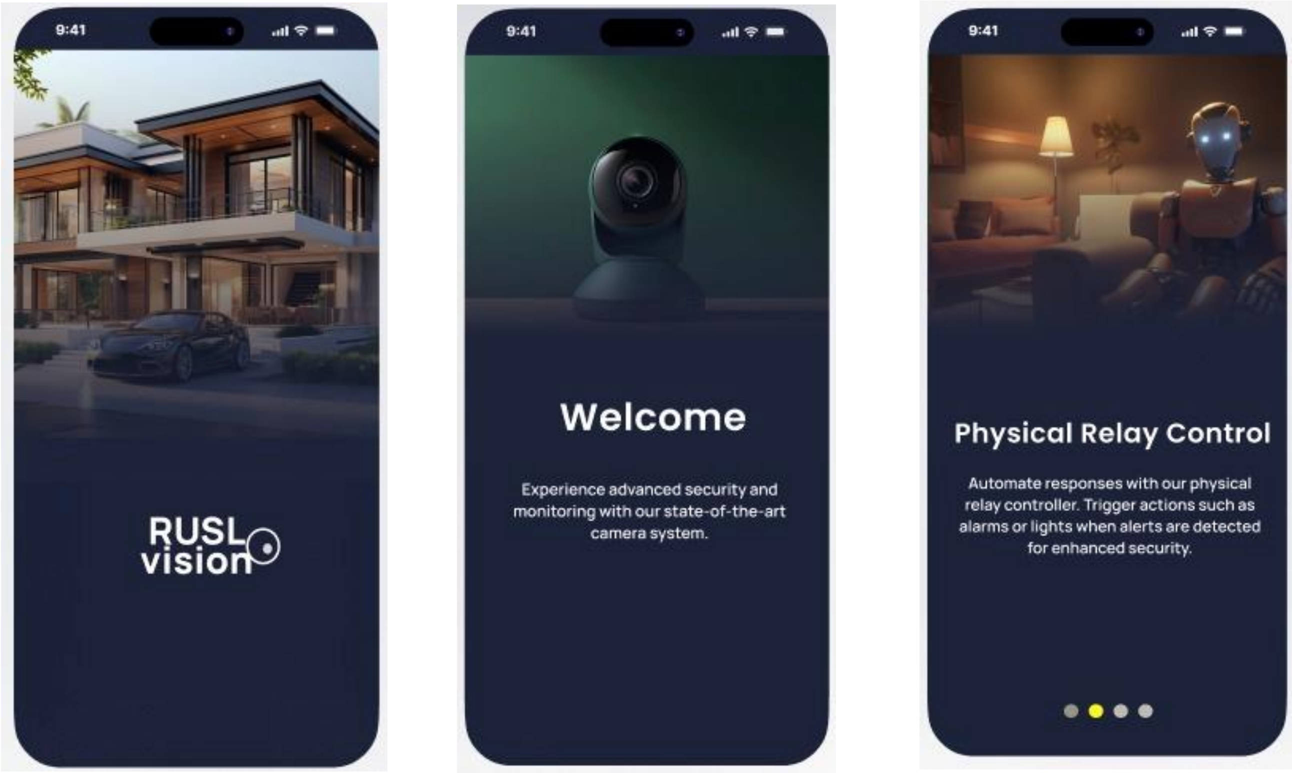
The system operates in diverse environments such as homes, offices, warehouses, and public places. It functions in both indoor and outdoor settings and must be capable of handling different lighting and weather conditions. The mobile app provides users with remote access to live video streams, security alerts, and controls for external devices, even when they are away from the monitored site. This makes the system ideal for situations where users need real-time updates and control over the surveillance environment.

Technologies

The core technologies involved in the system include:

- **ESP32-CAM module** for video capture and relay control integration.
- **YOLOv8** for AI-powered real-time object detection and classification.
- **Cloud-based services** for live streaming and data storage to ensure remote access and security.
- **Flutter-based mobile app** that serves as the user interface for video monitoring, receiving alerts, and controlling connected devices. The system uses Wi-Fi, RTSP streaming protocol, and programmable relays to achieve seamless hardware and software integration.

3.2.2 Interfaces (software/hardware) of the system)



9:41

RUSL vision

9:41

Welcome

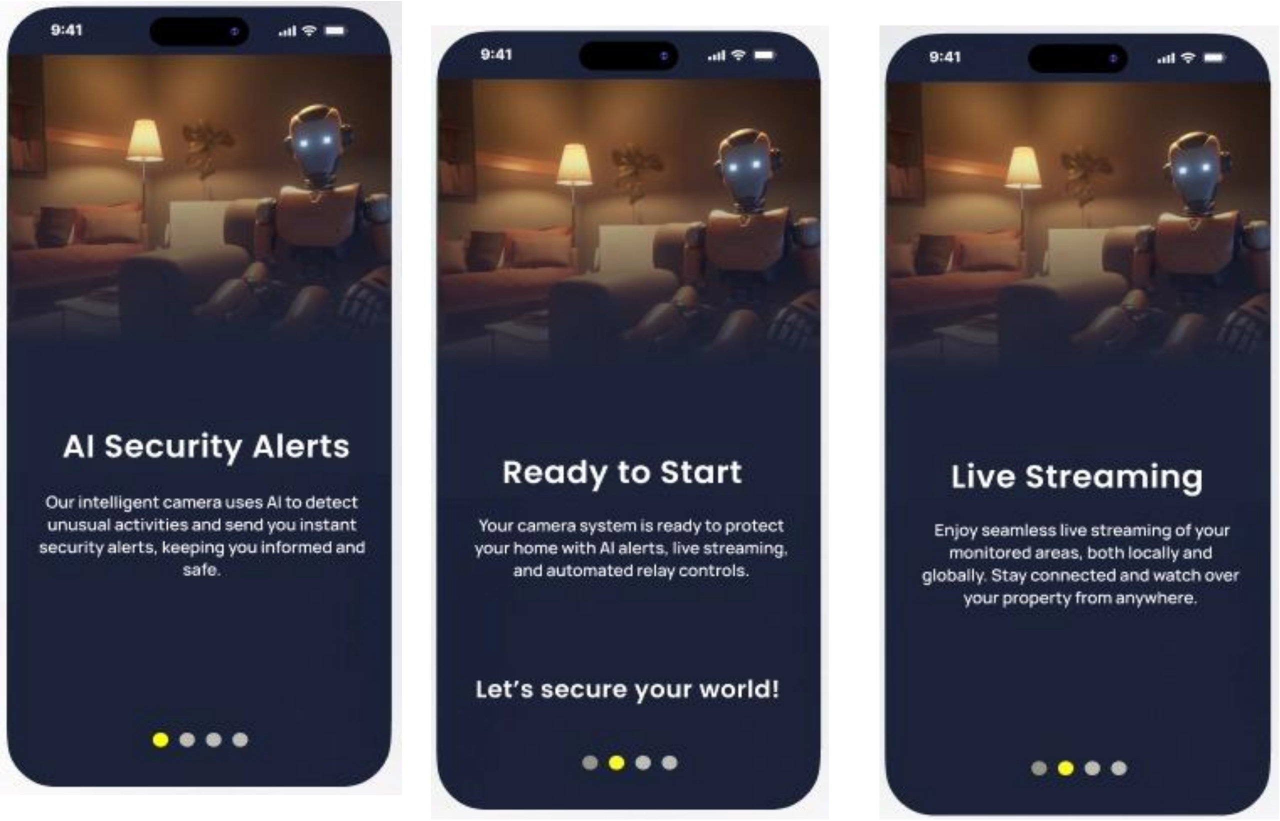
Experience advanced security and monitoring with our state-of-the-art camera system.

9:41

Physical Relay Control

Automate responses with our physical relay controller. Trigger actions such as alarms or lights when alerts are detected for enhanced security.

Figure 12



9:41

AI Security Alerts

Our intelligent camera uses AI to detect unusual activities and send you instant security alerts, keeping you informed and safe.

9:41

Ready to Start

Your camera system is ready to protect your home with AI alerts, live streaming, and automated relay controls.

Let's secure your world!

9:41

Live Streaming

Enjoy seamless live streaming of your monitored areas, both locally and globally. Stay connected and watch over your property from anywhere.

Figure 13

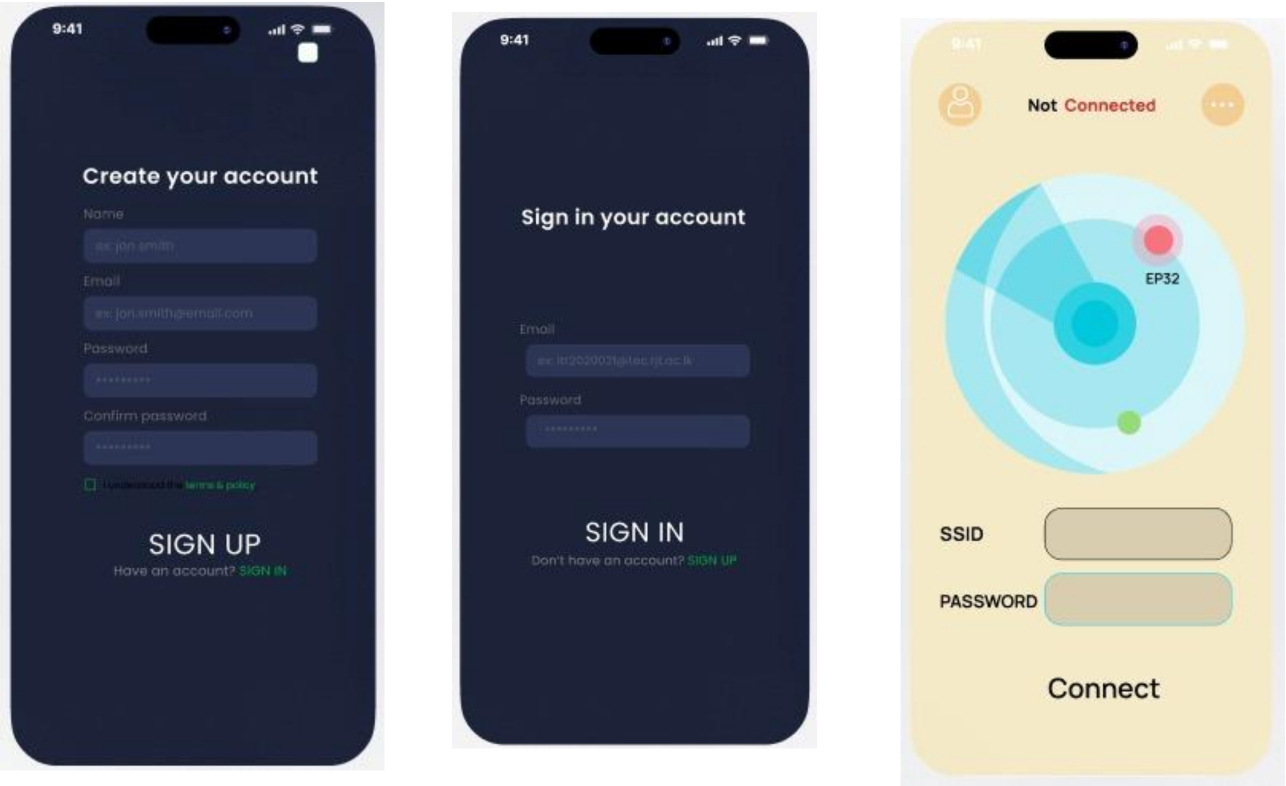


Figure 14

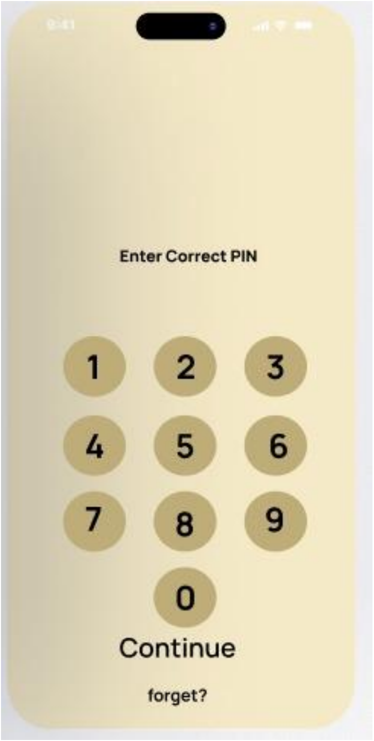
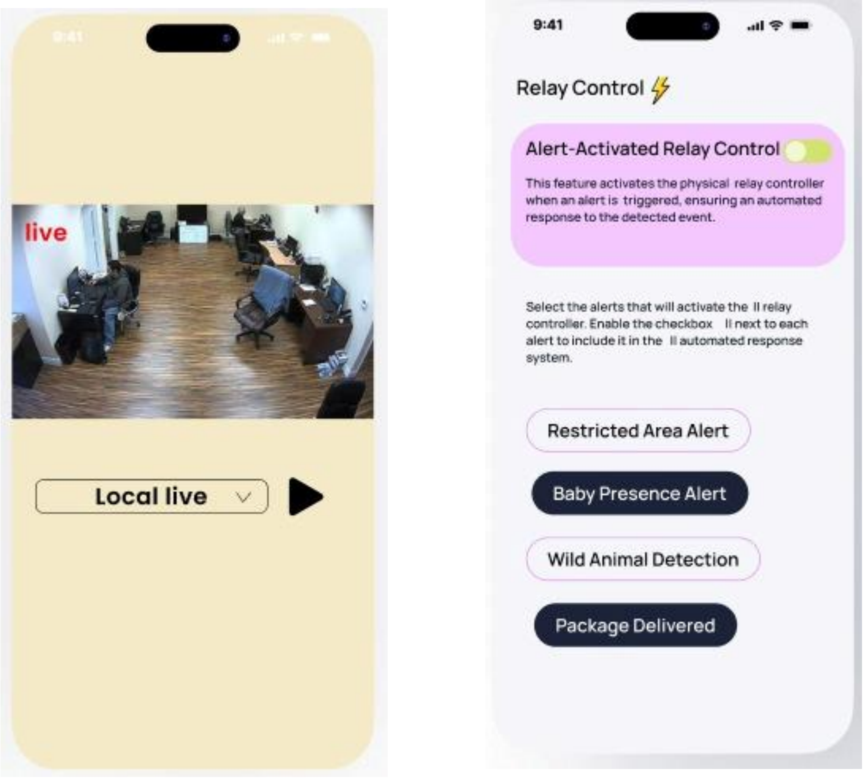


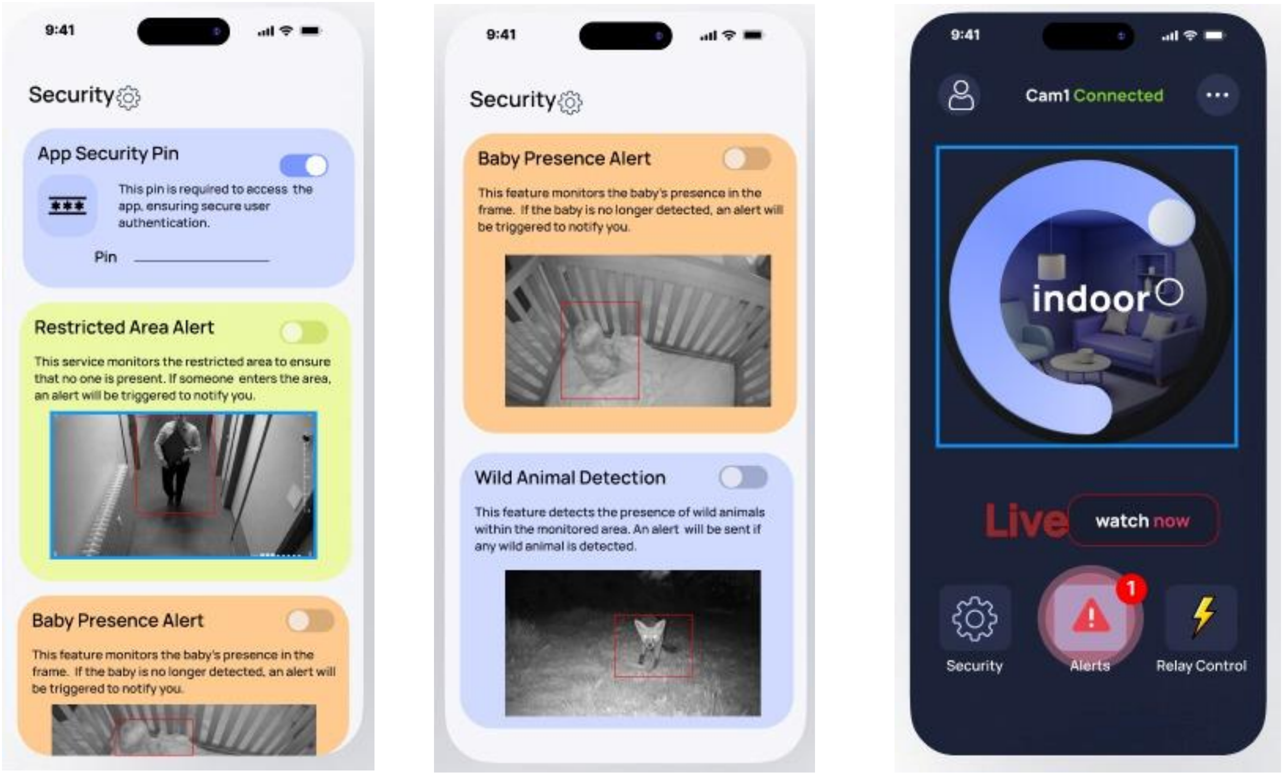
Figure 16



This feature activates the physical relay controller when an alert is triggered, ensuring an automated response to the detected event.

Select the alerts that will activate the II relay controller. Enable the checkbox II next to each alert to include it in the II automated response system.

Figure 15



This pin is required to access the app, ensuring secure user authentication.

This service monitors the restricted area to ensure that no one is present. If someone enters the area, an alert will be triggered to notify you.

This feature monitors the baby's presence in the frame. If the baby is no longer detected, an alert will be triggered to notify you.

This feature monitors the baby's presence in the frame. If the baby is no longer detected, an alert will be triggered to notify you.

This feature detects the presence of wild animals within the monitored area. An alert will be sent if any wild animal is detected.

Figure 17

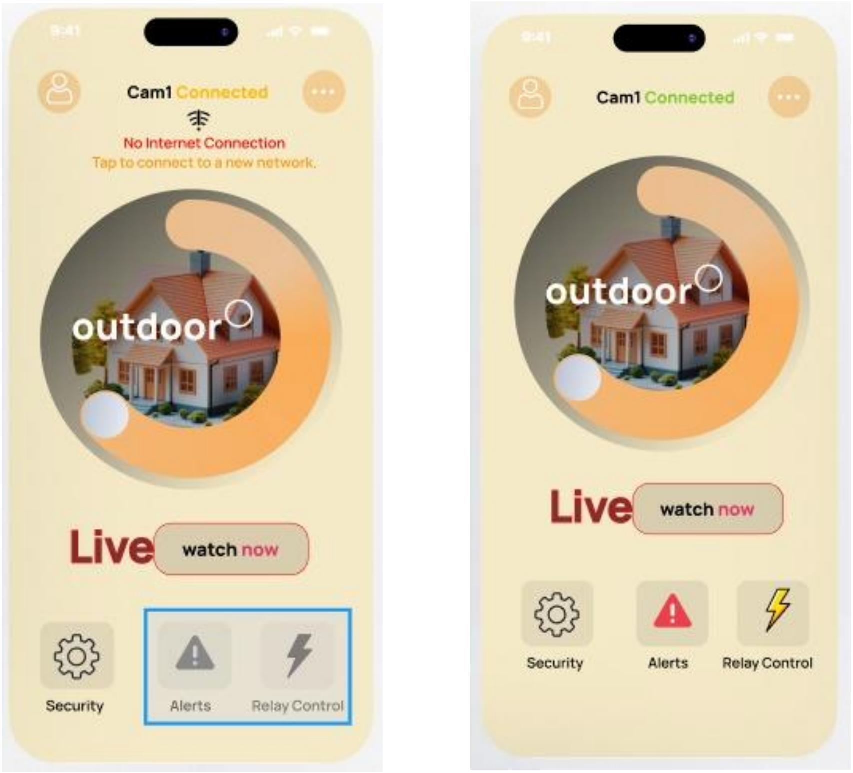


Figure 18

3.2.3 Design tools, techniques, templates

Design Tools

- **Figma/Adobe XD**: These tools were used for designing the user interface (UI) and user experience (UX) of the mobile application. They allowed for the creation of wireframes and prototypes before final implementation.
- **Arduino IDE**: Used to develop and program the ESP32-CAM for its core functionalities like live streaming, object detection, and relay control.
- **PyCharm/VS Code**: These development environments were used for programming AI algorithms, including YOLOv8 for object detection, as well as managing backend services.

Techniques

- **Wireframing and Prototyping**: Early-stage designs for the mobile app UI were created using wireframing techniques. Prototypes were developed to test usability and layout before finalizing the design.
- **Test-Driven Development (TDD)**: The AI detection system and other core features were developed using TDD to ensure the system functions as expected with a high level of accuracy.

Templates

- **Mobile App UI Templates**: Predefined UI components like buttons, sliders, and video panels were utilized to accelerate app development. These templates ensured consistency and efficiency in the design process.
- **Backend API Templates**: Standardized API templates were used to link the mobile app with cloud services and video streaming functionalities. These templates ensured a smooth integration between the front-end and back-end services.

3.3 Data Management

3.3.1 Design tools, techniques

Tools

- Database Management System (DBMS): MySQL was selected for its reliability, scalability, and robust support for relational data management, serving as the core for managing user, camera, recording, and stream data.

- ER Diagram Tool: draw.io was used to design Entity-Relationship (ER) diagrams, visually mapping out entities and their relationships.
- Development Environment: PyCharm/VS Code facilitated writing and testing SQL scripts and integrating the database with the system backend.
- Cloud Integration: AWS RDS or similar platforms hosted the database, ensuring secure, remote, and scalable access to surveillance data.

Techniques

- Entity-Relationship (ER) Modeling: Conceptualized entities like User, Camera, Recording, and Stream to establish relationships and attributes for a clear database structure.
- Normalization: The database was normalized to Third Normal Form (3NF) to minimize redundancy and ensure data integrity.
- Schema Refinement: Constraints like NOT NULL and UNIQUE, along with primary and foreign keys, ensured relational and data integrity.
- SQL Queries: SQL was used to create, manage, and query tables, supporting CRUD operations and system functionality.
- Cloud Integration: Hosting the database on cloud services achieved remote access and security, enabling seamless data retrieval from anywhere.

3.3.2 Conceptual database design

The conceptual design of the database was initially developed using an Entity-Relationship (ER) diagram. This stage defines the relationships and data flow between the four primary entities in the system: User, Camera, Recording and Stream. Each entity contains various attributes, and their relationships have been clearly outlined.

❖ User Entity

The User entity contains the following attributes.

- ➢ UserID (Primary Key)
- ➢ Username
- ➢ Password
- ➢ Email

❖ Camera Entity

The Camera entity manages information about the surveillance cameras, with these attributes.

- ➢ CameraID (Primary Key)
- ➢ IPAddress

❖ Recording Entity

The Recording entity stores details of recorded video files.

- ➢ RecordingID (Primary Key)
- ➢ Start_Time
- ➢ End_Time
- ➢ Storage Location

❖ Stream Entity

The Stream entity represents live streams captured by the cameras.

- StreamID (Primary Key)
- Start Time
- End Time
- Cloud URL

❖ Relationships

- A User can control multiple Cameras.
- A Camera can have multiple associated Recordings and Streams.
- A Camera can have one associated Streams.
- Recording and Stream entities are related through the Camera entity.

❖ Constraints

- Each User is uniquely identified by UserID.
- Each Camera must be assigned a uniquely identified CameraID.
- Each Recording must be assigned a uniquely identified RecordingID.
- Each Stream must be assigned a uniquely identified StreamID.

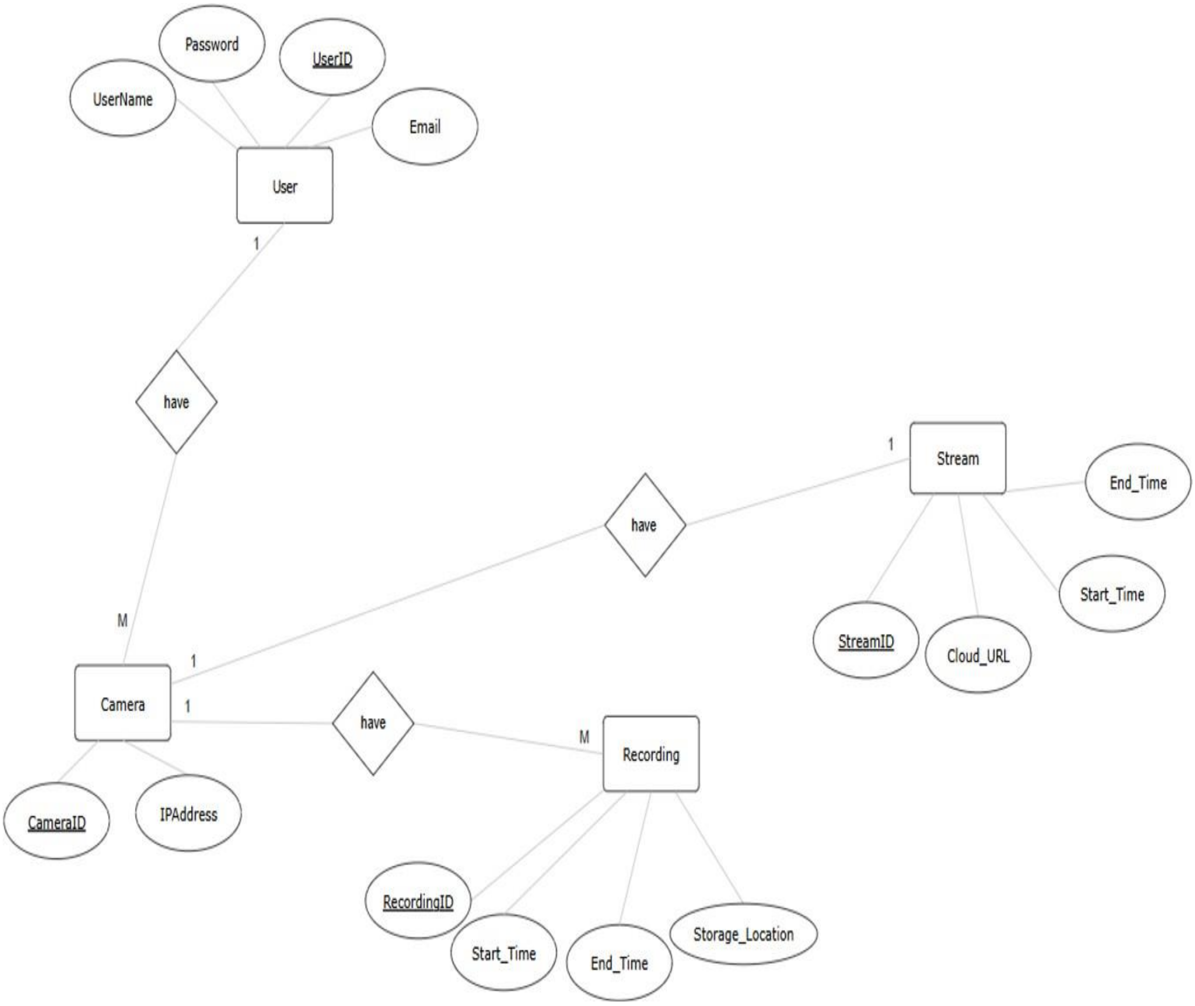


Figure 19

3.3.3 Logical database design and schema refinement

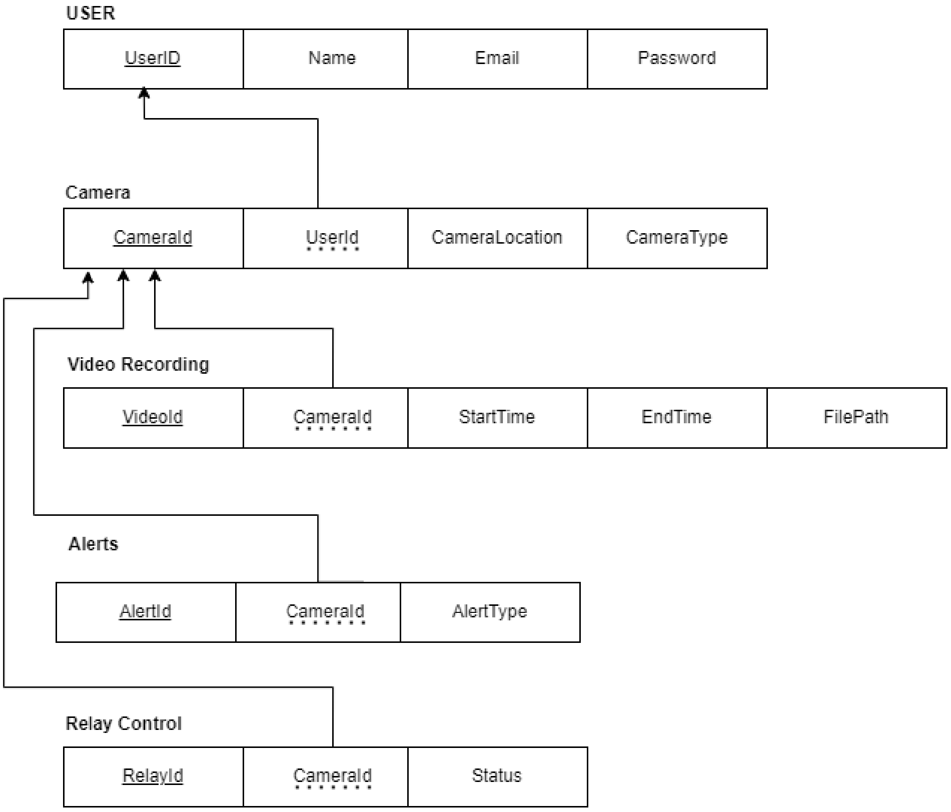


Figure 20

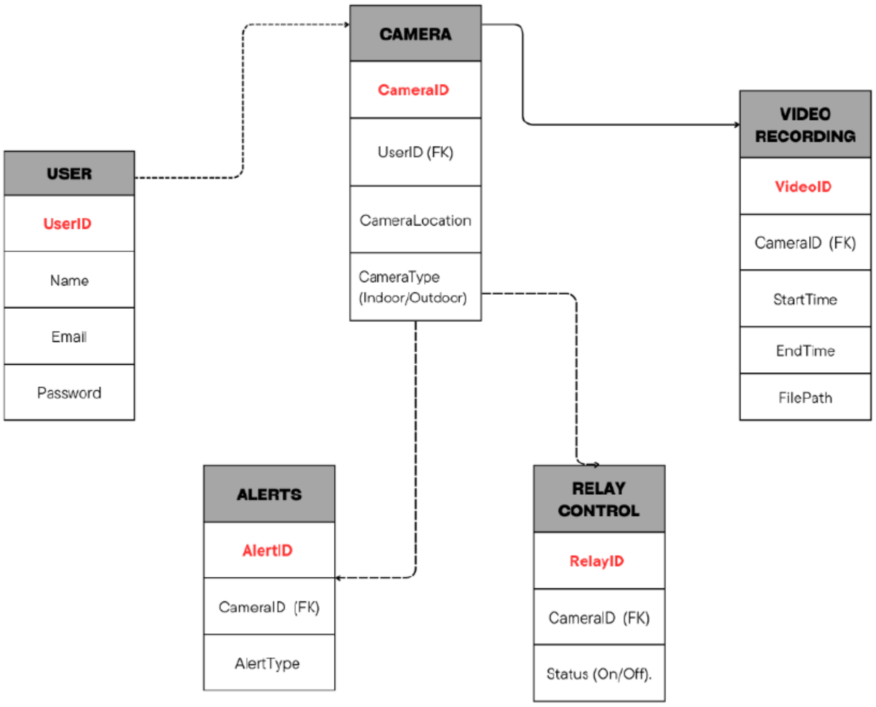
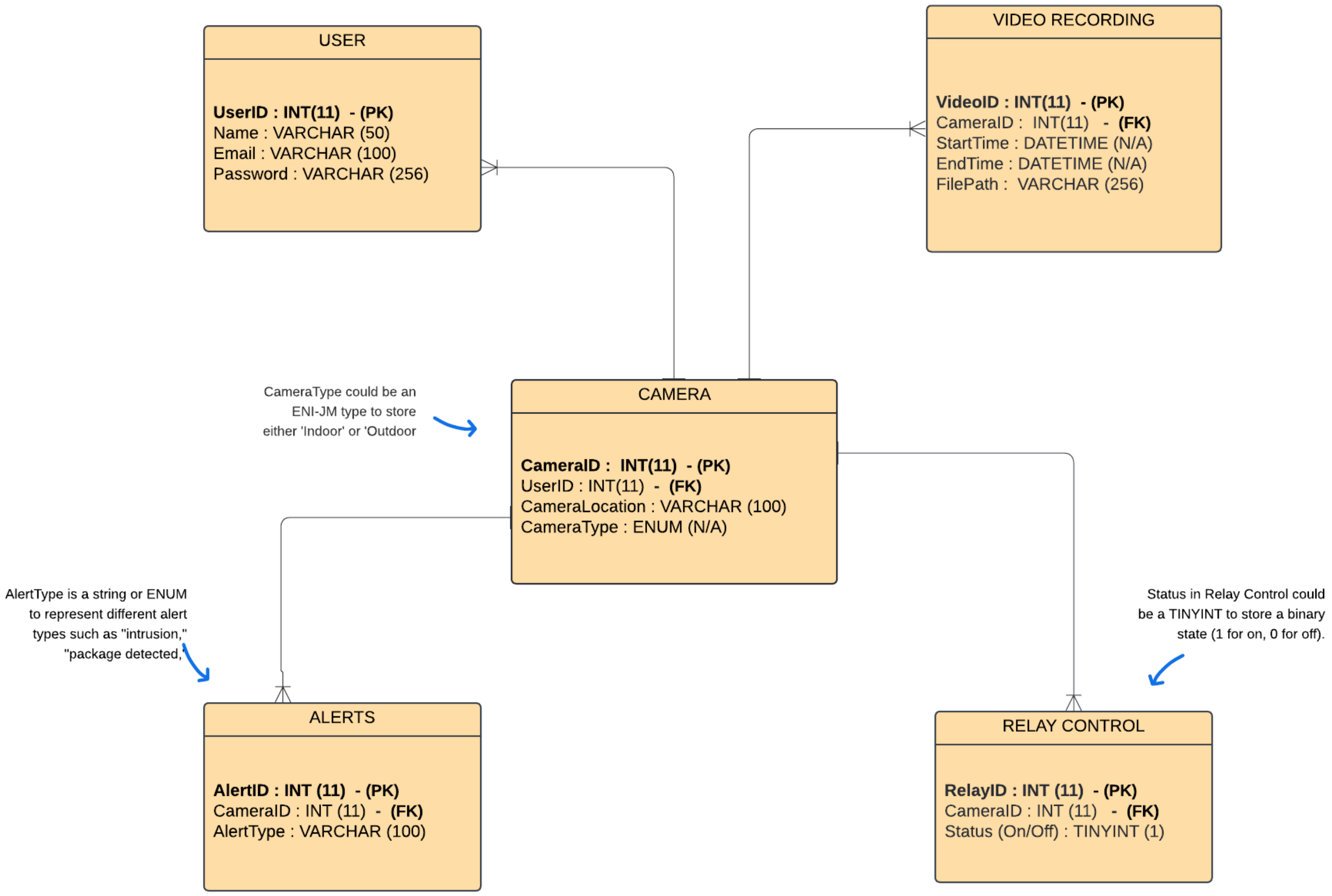


Figure 21

3.3.4 Physical database design


USER
UserID : INT(11) - (PK)
Name : VARCHAR (50)
Email : VARCHAR (100)
Password : VARCHAR (256)
VIDEO RECORDING
VideoID : INT(11) - (PK)
CameraID : INT(11) - (FK)
StartTime : DATETIME (N/A)
EndTime : DATETIME (N/A)
FilePath : VARCHAR (256)
CAMERA
CameraID : INT(11) - (PK)
UserID : INT(11) - (FK)
CameraLocation : VARCHAR (100)
CameraType : ENUM (N/A)
ALERTS
AlertID : INT (11) - (PK)
CameraID : INT (11) - (FK)
AlertType : VARCHAR (100)
RELAY CONTROL
RelayID : INT (11) - (PK)
CameraID : INT (11) - (FK)
Status (On/Off) : TINYINT (1)

CameraType could be an ENI-JM type to store either 'Indoor' or 'Outdoor

AlertType is a string or ENUM to represent different alert types such as "intrusion," "package detected,"

Status in Relay Control could be a TINYINT to store a binary state (1 for on, 0 for off).

Figure 22

3.5 Hardware Design

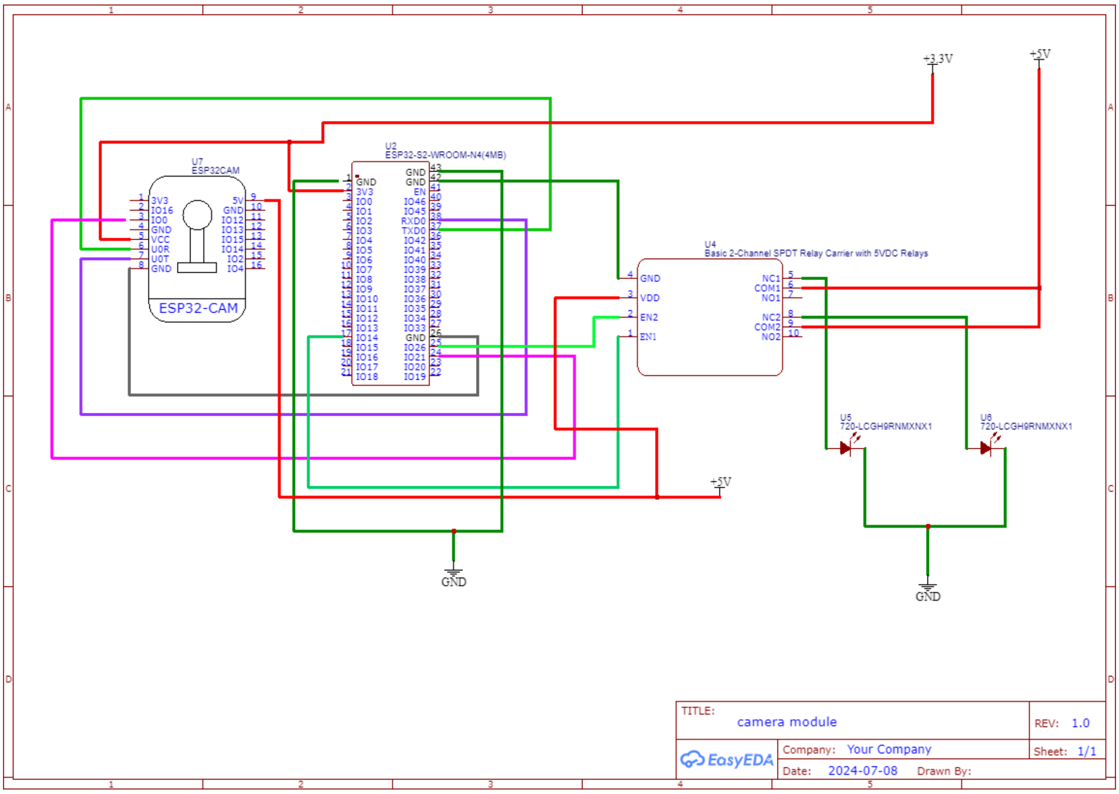


Figure 23

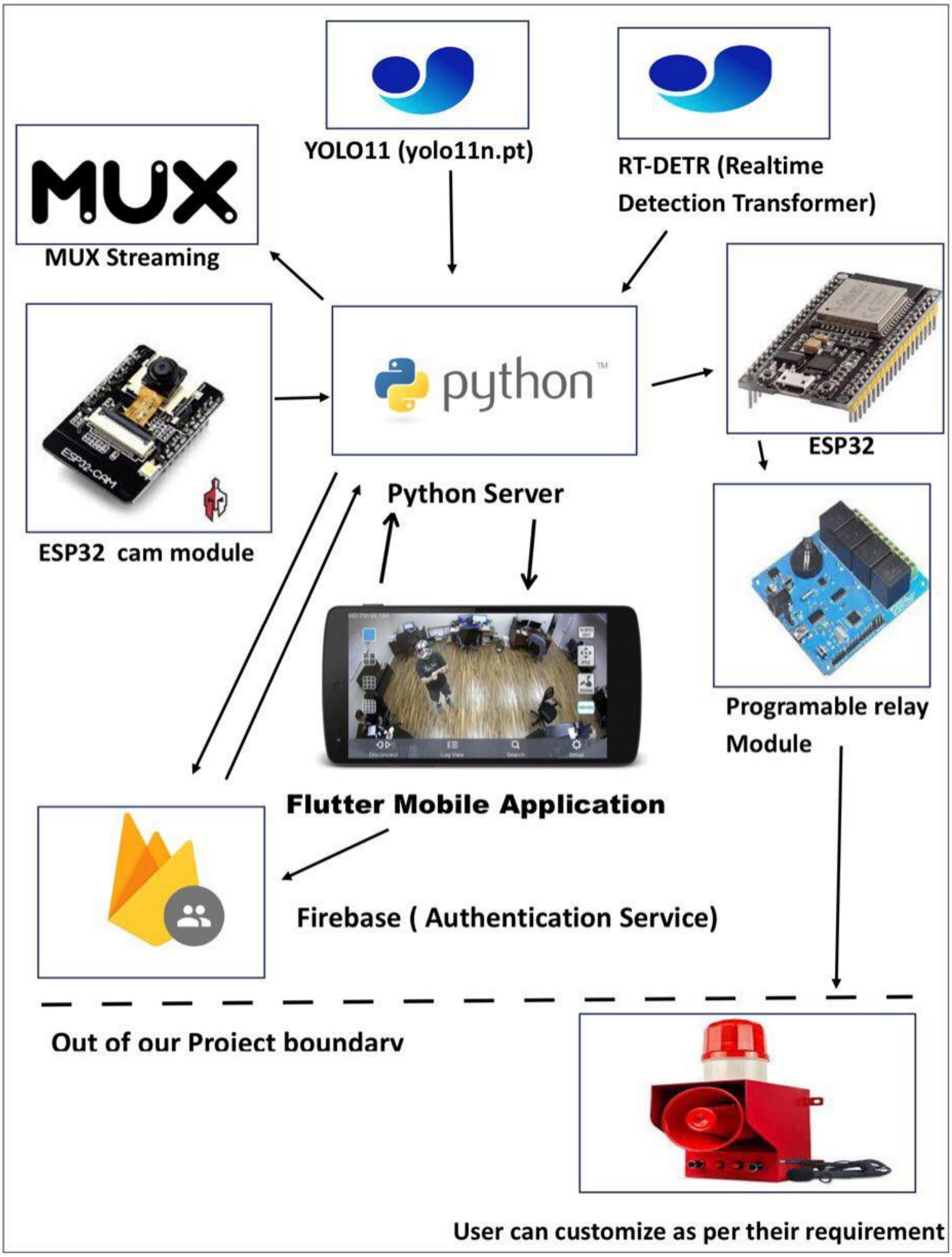

YOLO11 (yolo11n.pt)
RT-DETR (Realtime Detection Transformer)
MUX
MUX Streaming
python™
Python Server
ESP32
ESP32 cam module
Programable relay Module
Flutter Mobile Application
Firebase (Authentication Service)
Out of our Proiect boundarv
User can customize as per their requirement

Figure 24

4 Testing and evaluation

4.1 Testing plan

4.1Testing Plan

The testing plan outlines the methodology, costs, time frame, and tools/software used to ensure the functionality, performance, and reliability of the intelligent indoor/outdoor surveillance camera system.

Testing Methodology

The project will follow the following testing methodologies:

- Black-box testing: To verify the system's functionality without examining internal code.
- White-box testing: To validate code logic and flow.
- Regression testing: To ensure new changes do not affect existing functionality.
- Acceptance testing: To validate the system meets end-user requirements.

Costs Involved

The costs involved in the testing process are primarily related to:

- Software tools for testing (e.g., Selenium, JUnit).
- Hardware components for testing physical relay control.
- Cloud service subscriptions for live streaming and storage testing.
- Team member effort and time allocation.

Time Frame

The testing phase will span 4 weeks, divided into the following phases:

- Week 1: Unit Testing
- Week 2: Integration Testing
- Week 3: System Testing
- Week 4: Acceptance Testing and final evaluation.

Tools and Software

The following tools and software will be used during the testing process:

- Selenium: For automated testing of the mobile application.
- Postman: For API testing.
- JMeter: For performance testing.
- ESP32 test firmware: For relay control testing.

4.2 Testing

4.2.1 Unit Testing

Test scenario ID	Unit Test Scenario	Test case ID	UT-01
Test case description	Test camera streaming functionality.	Test priority	High
Pre-requisite	Camera module is functional.	Post-requisite	NA
Test execution steps:			

S No	Action	Input	Expected output	Actual output	Status	Note
1	Start stream	Start stream command	Live video feed starts	Livee Feed starts	Pass	
2	Stop stream	Stop stream command	Live video feed stops	Video stoping	Pass	
3	Click Loging button	Click the button	Login success	Login success	Pass	

Table 9

4.2.2 Integration Testing

Test scenario ID	Integration Test Scenario	Test case ID	IT-01
Test case description	Test interaction between camera and relay module.	Test priority	High
Pre-requisite	Camera and relay are connected to the system.	Post-requisite	NA
Test execution steps:			

S No	Action	Input	Expected output	Actual output	Status	Note
1	Start the live streaming feature and connect to the cloud server.	A video feed and valid cloud server credentials.	The system streams the video feed to the cloud server without interruptions	The system streams the video feed to the cloud server without interruptions	Pass	
2	Send the "Turn ON" command to the relay system remotely.	Command: "Turn ON"	The relay activates, and the connected system powers ON	The relay activates, and the connected system powers ON	Pass	
3	Send the "Turn OFF" command to the relay system remotely.	Command: "Turn OFF"	The relay deactivates, and the connected system powers OFF	The relay deactivates, and the connected system powers OFF	Pass	

Table 10

4.2.3 System Testing

Test scenario ID	System Test Scenario			Test case ID	ST-01	
Test case description	Test overall system functionality.			Test priority	High	
Pre-requisite	System is powered ON.			Post-requisite	NA	
Test execution steps:						
S No	Action	Input	Expected output	Actual output	Status	Note
1	Access live stream via mobile app	User login credentials	Live feed accessible	Feed accessible	Pass	
2	Toggle relay using app	ON/OFF command	Relay switches accordingly	Relay sswitched	Pass	
3	Activate the AI object detection	Video feed containing predefined objects	The system correctly identifies and labels all objects in the video feed	The system correctly identifies and labels all objects in the video feed	Pass	
4	Start recording, stop it, and play back the video.	Video input for recording and playback command.	Video records and plays back without interruptions or quality issues.	Video records and plays back without interruptions or quality issues.	Pass	
5	End to end sign up	Complete valide data	Redirect to login page	Redirect to login page	Pass	

Table 11

4.2.4. Acceptance Testing

Test scenario ID	Acceptance Test Scenario	Test case ID	AT-01
Test case description	Validate user login functionality.	Test priority	High
Pre-requisite	User account exists.	Post-requisite	User logs in successfully.

Test execution steps:

S No	Action	Input	Expected output	Actual output	Status	Note
1	Launch application	App URL	Login page displays	Login page displays	Pass	
2	Loging	Correct Username + Correct Password Wrong Username + Correct Password	Succefull login Error Massage Display	Succefull login Error Massage Display	Pass Pass	
3	Sign Up	Valid email, password, mandatory details.	The user is successfully registered	The user is successfully registered	Pass	

Table 12

4.3 Test results and conclusion of testing

The testing process confirmed that all modules functioned as expected, and the system meets the project requirements. Any identified issues were resolved, ensuring the system is reliable, efficient, and ready for deployment.

5 Conclusion

5.1 Conclusions of the project

The proposed project has successfully demonstrated the integration of AI-powered object detection and IoT technologies to revolutionize traditional surveillance systems. By combining the ESP32-CAM with YOLOv8 for real-time object detection, the system addresses common limitations of conventional CCTV systems, such as high costs, false alarms, and limited functionality. Cloud-based live streaming, recording, playback, and programmable relay control were implemented to provide enhanced monitoring capabilities. The Flutter-based mobile application further enhances user convenience by enabling remote access, intelligent notifications, and system control. This project sets a benchmark for future surveillance technologies, showcasing the potential of combining AI and IoT to tackle everyday security challenges effectively.

5.2 Lessons learned and skills earned

- **Technical Proficiency**: Gained hands-on experience with IoT devices such as the ESP32-CAM and the integration of AI models like YOLOv8 for real-time object detection.
- **Software Development Skills**: Improved skills in Flutter for mobile application development and FFmpeg for video stream relay and processing.
- **System Design and Implementation**: Learned how to design and implement scalable systems combining hardware and software components.
- **Problem-Solving Abilities**: Addressed challenges related to hardware limitations, network connectivity, and real-time video streaming.
- **Team Collaboration**: Strengthened teamwork and communication skills through collaborative project development.

5.3 Recommendations for further improvements

- **Enhanced AI Capabilities**: Incorporate advanced AI models for more accurate and diverse object detection capabilities, such as facial recognition or anomaly detection.
- **Motion Detection Integration**: Add motion sensors to enhance detection accuracy and reduce false positives.
- **Cloud Storage**: Integrate cloud storage options for long-term video recording and retrieval.
- **Edge AI Processing**: Optimize the system for edge AI processing on the ESP32-CAM or Raspberry Pi to reduce latency and improve performance.
- **Expanded Use Cases**: Adapt the system for industrial and agricultural applications, adding functionalities like environmental monitoring and equipment status tracking.
- **Energy Efficiency**: Implement power-saving features to make the system more sustainable for continuous operation.

4 References

- [1] A. Arkhipenko, "Multi-client MJPEG Streaming from ESP32," Hackster.io, Jan. 2024. [Online]. Available: https://www.hackster.io/anatoli-arkhipenko/multi-client-mjpeg-streaming-from-esp32-47768f. [Accessed: Jan. 2025].
- [2] A. Arkhipenko, "ESP32 MJPEG Multiclient ESPCAM Drivers," GitHub, Jan. 2024. [Online]. Available: https://github.com/arkhipenko/esp32-mjpeg-multiclient-espcam-drivers/tree/master/PlatformIO. [Accessed: Jan. 2025].
- [3] Espressif Systems, "ESP32 Series Datasheet," 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Accessed: Jan. 2025].
- [4] TensorFlow, "YOLOv8 Model Documentation," TensorFlow, 2023. [Online]. Available: https://www.tensorflow.org/models. [Accessed: Jan. 2025].
- [5] Google Developers, "Flutter Development Framework," Google, 2023. [Online]. Available: https://flutter.dev. [Accessed: Jan. 2025].

5 Recommendation of supervisor(s) on the final Report

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

Comments (if any):

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: 2025/02/14

6 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		

7 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	