FYP-Rept-Intraguard Survellience System.docx

by Saad Akbar

Submission date: 06-Jul-2025 05:52PM (UTC+0500)
Submission ID: 2710739359
File name: FYP-Rept-Intraguard_Survellience_System.docx (1.7M)
Word count: 14245
Character count: 87021

<INTEGRA GUARD SURVEILLANCE SYSTEM>

Final Year Project Report

Submitted by

Zainab Khalid (2326-2021)
Nida Aamir (2458-2021)

Supervisor

Afzal Hussain

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Artificial Intelligence
2021

Faculty of Engineering Sciences and Technology

Hamdard Institute of Engineering and Technology

Hamdard University, Main Campus, Karachi, Pakistan

Certificate of Approval

Faculty of Engineering Sciences and Technology

Hamdard Institute of Engineering and Technology
Hamdard University, Karachi, Pakistan

This project "**IntegraGuard Surveillance System**" is presented by **Zainab Khalid and Nida Aamir** under the supervision of their project advisor and approved by the project examination committee, and acknowledged by the Hamdard Institute of Engineering and Technology, in the fulfillment of the requirements for the Bachelor degree of **Bachelor's degree in Artificial Intelligence**

Mr. Afzal Hussain
(Project Supervisor)

In-charge FYP

Mr. Mustafa Ali
(Project Co-Supervisor)

Chairman
(Department of Computing)

(Dean, FEST)

Authors' Declaration

We declare that this project report was carried out in accordance with the rules and regulations of Hamdard University. The work is original except where indicated by special references in the text and no part of the report has been submitted for any other degree. The report has not been presented to any other University for examination.

Dated:

Authors Signatures:

_____
Zainab Khalid

_____
Nida Aamir

Plagiarism Undertaking

We, **Zainab Khalid** and **Nida Aamir** solemnly declare that the work presented in the Final Year Project Report titled **"IntegraGuard Surveillance System"** has been carried out solely by ourselves with no significant help from any other person except few of those which are duly acknowledged. We confirm that no portion of our report has been plagiarized and any material used in the report from other sources is properly referenced.

Dated:

Authors Signatures:

_____
Zainab Khalid

_____
Nida Aamir

Acknowledgments

Praise be to Allah, the Lord of the Worlds. He gave us the strength, perseverance, and wisdom to finish this project. Without His countless blessings, this ac hievement would not have been possible.

We thank our Project Supervisor, Mr. Afzal Hussain. His support, guidance, and feedback steered us through a year of ups and downs. His expertise and support were key to overcoming challenges and reaching milestones in the project's development.

We also sincerely appreciate our Co-Supervisor, Mr. Mustafa Ali, for his helpful suggestions and support throughout. His knowledge and input significantly enhaced the quality of our work.

Finally, we acknowledge all those who contributed their time and expertise, directly or indirectly, to the successful completion of this project.

Document Information

Table 1: Document Information

Customer	
Project Title	IntegraGuard Surveillance System
Document	Final Year Project Report
Document Version	1.0
Identifier	FYP-022/FL24 Final Report
Status	Final
Author(s)	Zainab Khalid, Nida Aamir
Approver(s)	Mr.Afzal Hussain
Issue Date	

Definition of Terms, Acronyms, and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Table 2: Definition of Terms, Acronyms, and Abbreviations

AI	Artificial Intelligence
YOLO	You Only Look Once
Weapon Detection System	A surveillance system using AI and computer vision to identify visible weapons (e.g., pistols, knives) from live CCTV footage.
Real-Time Processing	The capability of the system to detect and respond to threats with minimal delay (typically within 2–5 seconds).
DAL	Data Access Layer
GDPR	General Data Protection Regulation
API	Application Programming Interface
ATM	Automated Teller Machine

Abstract

The modern world of security is characterized by the fact that ATM facilities have had to deal with a wide range of security issues and the only way to deal with the threats that are faced is to be proactive and have a smart way of detecting the threats so that the appropriate responses are provided in time. The proposed project presents a real-time ATM weapon which would solve these vital issues, combining innovative artificial intelligence and surveillance technologies. The system uses CCTV cameras to detect the uncovered weapons (including guns and knives) to detect possible threats.

The system offers a solution that has main aspects namely weapon detection, and a real-time alert system. To analyze suspicious behavior, weapon identification relies on the detection capabilities that are powered by YOLO (You Only Look Once). It has an interactive web-based dashboard that can be used to monitor live video feeds, get alerts on detection, and generate threat reports. In case of any possible threat, the system alerts the monitoring staff and they can escalate the situation further by activating an alert to security staff.

The proposed project will help increase the safety and security of ATMs as it will allow quickly detecting and reacting to threats, which will considerably decrease risks. The system also guarantees a smooth experience to the operators by using up-to-date web development frameworks to make the system scalable and easy to use. Besides filling the current gaps in ATM security management, the suggested solution would also provide the basis of further innovation in the field of intelligent surveillance.

Keywords:

ATM Security, *Weapon Detection*, *Real-Time Surveillance*, *YOLOv8*, *Object Detection*, *Deep Learning*, *Artificial Intelligence (AI)*, *Threat Detection*, *Live Video Feed Monitoring*, *Web-Based Security System*, *Email Alert System*, *Automated Surveillance*, *Security Notifications*, *Machine Learning in Security*, *AI-Powered Monitoring*

Table of Contents

Certificate of Approval 2
Authors' Declaration 3
Acknowledgments 5
Document Information 6
Abstract 7
List of Figures 10
List of Tables 11
CHAPTER 1 12
INTRODUCTION 12
1.1 Motivation 12
1.2 Problem Statement 12
1.3 Goals and Objectives 12
1.4 Project Scope 13
CHAPTER 2 14
RELEVANT BACKGROUND & DEFINITIONS 14
CHAPTER 3 15
LITERATURE REVIEW & RELATED WORK 15
Literature Review 15
Related Work 17
Gap Analysis 18
CHAPTER 4 19
1. Software Engineering Methodology 19
2. Project Methodology 20
3. Phases of Project 21
4. Software/Tools that Used in Project 22
5. Hardware that Used in Project 24
Chapter 5 26
5.1 Proposed System Architecture/Design 26
5.2 Functional Specifications 32
5.3 Non-Functional Specifications 35
5.4 Testing 38
5.5 Purpose of Testing 39
5.6 Test Cases 40

Chapter 5 54
EXPERIMENTAL EVALUATIONS & RESULTS 54
Evaluation Testbed 54
Results and Discussion 55
CHAPTER 6 56
CONCLUSION AND DISCUSSION 56
7.1 Strength of this Project 56
7.2 Limitations and Future Work 57
7.3 Reasons for Failure – If Any 58
REFERENCES 59
APPENDICES 60
A0. COPY OF PROJECT REGISTRATION FORM **Error! Bookmark not defined.**
A1A. PROJECT PROPOSAL AND VISION DOCUMENT 60
A1B. COPY OF PROPOSAL EVALUATION COMMENTS BY JURY**Error! Bookmark not defined.**
A2. REQUIREMENT SPECIFICATIONS 64
A3. DESIGN SPECIFICATIONS 68
A4. OTHER TECHNICAL DETAIL DOCUMENTS 85
Test Cases Document 85
UI/UX Detail Document 95
Coding Standards Document 97
Project Policy Document **Error! Bookmark not defined.**
User Manual Document 97
A5. FLYER & POSTER DESIGN 99
A6. COPY OF EVALUATION COMMENTS **Error! Bookmark not defined.**
COPY OF EVALUATION COMMENTS BY SUPERVISOR FOR PROJECT – I MID SEMESTER EVALUATION **Error! Bookmark not defined.**
COPY OF EVALUATION COMMENTS BY JURY FOR PROJECT – I END SEMESTER EVALUATION 100
COPY OF EVALUATION COMMENTS BY SUPERVISOR FOR PROJECT – II MID SEMESTER EVALUATION **Error! Bookmark not defined.**
A7. MEETINGS' MINUTES & Sign-Off Sheet **Error! Bookmark not defined.**
A8. DOCUMENT CHANGE RECORD **Error! Bookmark not defined.**
A9. PROJECT PROGRESS **Error! Bookmark not defined.**
A10. RESEARCH PAPER **Error! Bookmark not defined.**
A11. Plagiarism Test Summary Report **Error! Bookmark not defined.**

List of Figures

Figure No	Description	Page No.
FIGURE 1. 1 : SCOPE OF PROJECT		2
FIGURE 2. 1 : FACE DETECTION		4
FIGURE 2. 2 : FACE RECOGNITION PROCESS		5
FIGURE 4. 1 : SYSTEM FLOW FOR FACE RECOGNITION		9
FIGURE 4. 2 : USE CASE MODEL		10
FIGURE 4. 3 : USE CASE DIAGRAM FOR SETTING		12
FIGURE 5. 1 : DESIGN OF A SYSTEM		13
FIGURE 6. 1 : STEP 1 DOWNLOADS OPENCV MANAGER		18
FIGURE 6. 2 : STEP 2 INSTALLS APPLICATION		19
FIGURE 6. 3 : STEP 3 OPEN APPLICATION		20
FIGURE 6. 4 : STEP 4 OPEN INBOX		21
FIGURE 6. 5 : STEP 5 SELECT MESSAGE		22
FIGURE 6. 6 : STEP 6 TEXT VARIATIONS		23
FIGURE 6. 7 : STEP 7 READ CONTACTS		24
FIGURE 6. 8 : STEP 8 FOR WRITE MESSAGE		25
FIGURE 6.10 : STEP 10 SETTINGS		26

List of Tables

Table No.	Description	Page No.
TABLE 2.1: COMPARISON TABLE		6
TABLE 3.1: PHASES OF PROJECT		7
TABLE 5.1: TEST CASE 1		16
TABLE 5.2: TEST CASE 2		16
TABLE 5.3: TEST CASE 3		16
TABLE 5.4: TEST CASE 4		17
TABLE 5.5: TEST CASE 5		17
TABLE 5.6: TEST CASE 6		17

CHAPTER 1

INTRODUCTION

1.1 Motivation:

The swiftness of technological development has changed the security environment especially in places of sensitive business like the ATMs security are very important. Armed robberies and thefts are the examples of weapon related crimes which have not ceased even in spite of technology advancement and in some parts increased. It is emphasizing the necessity of the innovative solutions to anticipate the risks and improve security..

The nature of ATMs presents special security challenges because of the small size of the machines, the inability to have constant physical presence, and the need to protect the privacy of the consumer. The conventional surveillance systems are overly dependent on manual surveillance that is usually ineffective, subject to human error, and unable to offer real-time reactions. This presents a chance to criminals to take advantage of the weak areas.

IntegraGuard Surveillance System (ISS) fills these loopholes by using advanced artificial intelligence and machine learning to provide real-time gun detection. This guarantees immediate response, which may save lives and minimize the financial losses, as well as maximize the rate and accuracy of detecting threats.

1.2 Problem Statement:

To date, the current ATM security systems rely largely on the traditional surveillance system which is not monitored and timely to detect the threats and unlawful activities. Such approaches do not work in most cases, hence the threats remain unnoticed over a certain duration of time, the responses are inadequate, and the businesses become more vulnerable to potential crimes. This is made worse by inefficiencies and human error because staff are overwhelmed by the manual screening of hundreds or thousands of hours of CCTV.

To address such concerns, there is an immediate need to have an intelligent, AI-powered system that will be able to identify visible threats such as weapons in real-time. This type of system would assist surveillance officers in making fast and wise decisions and enhance ATM safety and security significantly.

1.3 Goals and Objectives

Primary Goal:
To create a high-tech, web-based security system of ATMs that allows to detect weapons in real-time based on live video feeds and notify security officials by email immediately, so they could make informed decisions quickly.

Specific Objectives:

- **Enhance Threat Detection Accuracy:** Use object detection models powered by AI to accurately detect the presence of visible weapons like guns and knives in real-time video.
- **Enable Real-Time Monitoring:** Incorporate an interface that streams live video feed oand present a centralized point of viewing and managing threats in real-time.
- **Automate Alert Notifications:** An automated alert system should be created whereby email notifications are sent immediately to identified security personnel when weapons are detected.
- **Reduce Manual Oversight:** Minimize reliance on continuous human monitoring by automating detection and response mechanisms, leading to faster incident response times.

1.4 Project Scope

In Scope:

- Enhancing the efficiency of ATM security monitoring.
- Automating the real-time detection of visible weapons (e.g., guns, knives) in live video feeds using AI-based object detection models.
- Providing a web-based dashboard for real-time video monitoring, weapon detection visualization, and alert management.
- Enabling integrated email-based alert mechanisms to instantly notify security personnel upon weapon detection.
- Supporting data-driven decision-making through detailed logs, detection timestamps, and alert histories for ongoing security improvement.

Out of Scope:

- Detection of concealed weapons.
- Behavioral analysis or emotion recognition.
- Advanced AI features such as crowd behavior analysis, anomaly detection, or audio-based threat detection.

CHAPTER 2

RELEVANT BACKGROUND & DEFINITIONS

The foundation of effective ATM security lies in proactive and intelligent threat detection systems. Traditional security measures, which are based on the principle of relying on manual surveillance and delayed responses, are fast becoming unsustainable in the fast-paced, high-risk environment of today. These outdated systems often fail to identify threats like **revealed weapons** in real time, leading to compromised safety and delayed intervention.

The latest developments in artificial intelligence and machine learning have introduced automated solutions for surveillance and threat detection. **Object detection** technologies have revolutionized the management of security in critical domains. However, most of the existing systems do not meet the challenges of the real world. For example, traditional CCTV monitoring is based on human attention, which eventually leads to fatigue and missed threats. Moreover, the current systems lack integrated features such as real-time alerts, efficient communication, and scalable deployment.

The second challenge is the fragmented nature of existing security systems. Most solutions are unilateral in focus, concentrating only on **basic monitoring**, leaving crucial components such as real-time alerting and threat logging to be separately or manually managed. This disorganized approach results in inefficiencies, delayed responses, and higher operational risks.

This project is designed as an integrated solution to address these gaps by combining **real-time weapon detection** into a unified, web-based platform tailored for ATM security. Using YOLO (You Only Look Once) for **revealed weapon detection** (e.g., guns and knives), the system ensures proactive and intelligent threat detection. The portal also offers a user-friendly web interface with live video monitoring, real-time alerts, and detailed threat logs for data-driven decision-making.

The project takes inspiration from security technologies already designed to overcome their current limitations in terms of scalability, accessibility, and ease of use. By implementing modern AI and web development technologies, the system offers a seamless experience to the monitoring personnel and takes timely action in the event of a detected threat.

This system is the transformative tool to meet the needs of modern ATM environments as security practices continue to evolve. It is aimed at redefining ATM security, addressing inefficiencies, ensuring proactive threat management, and establishing a strong foundation for future innovation and scalability.

CHAPTER 3

LITERATURE REVIEW & RELATED WORK

Literature Review

Improving safety and reducing dangerous incidents is a significant challenge, and researchers have worked extensively on using object detection to enhance security. Smart surveillance systems typically follow three main steps: first, extracting basic information like tracking objects and identifying features; second, detecting unusual activities, behaviors, or weapons; and finally, making decisions based on the detection of abnormalities or suspicious events.

One approach, described in (He et al., 2018) https://doi.org/10.1007/s11042-017-5255-z , uses convolutional neural networks (CNNs) to analyze videos. Instead of processing every part of a video, this method focuses on specific areas where changes occur, making it faster and more cost-efficient. By applying spatial-temporal layers, the system captures both motion and object details, reducing noise and improving accuracy. Another method uses a graph-based learning system that distinguishes between normal and abnormal data. It trains the system to identify unusual patterns efficiently by focusing only on abnormal cases, saving both time and resources.

In the context of traffic scenes, (He et al., 2016) An anomaly-introduced learning method for abnormal event detection | Multimedia Tools and Applications introduced a system that detects objects like cars, bicycles, and traffic signs.The process involves three steps: detecting objects, recognizing their type, and tracking their movement. By sharing features across detection tasks, their system works faster and delivers better results. Additionally, the system categorizes objects into subcategories to handle variations, achieving good performance on various datasets.

(Grega et al., 2016) **https://doi.org/10.3390/s16010047** developed a tool for detecting knives and firearms in real-time security footage. This system aims to minimize false alarms while ensuring timely alerts to security personnel. It achieves 94.93% accuracy for knife detection and slightly lower sensitivity for firearm detection.

(Jietal.,2013) https://ieeexplore.ieee.org/document/6165309?utm_source=wiley&getft_integrator=wileycreated a deep learning-based system to recognize human behavior in security videos. This system uses a 3D CNN model that processes raw video data and combines multiple observations to deliver more accurate and efficient results.

(Pang et al., 2020) https://www.mdpi.com/1424-8220/20/6/1678 developed a real-time system for detecting weapons hidden under clothing using passive millimeter wave imaging. They employed the YOLO object detection model, which achieved impressive accuracy (95%) and speed (36 frames per second). The performance of YOLO was also compared with other detection models like SSD-VGG16 and Single MultiBox Detector, where it showed superior results. Similarly, Warsi et al. **https://doi.org/10.1155/2021/9975700** worked on handgun detection in live surveillance videos. They used YOLOv3 and Faster RCNN, training the models on the ImageNet dataset. Their results showed that YOLOv3 is faster.

Related Work

Researchers have explored diverse techniques for enhancing surveillance and detecting anomalies in video footage. The primary goals of these efforts are to improve safety, reduce false alarms, and provide efficient real-time solutions.

- **Smart Surveillance Frameworks:** Many studies have focused on systems operating in three stages: extracting features, detecting unusual activities, and making anomaly- based decisions. These frameworks leverage advanced methods such as spatial- temporal convolutional layers and anomaly-based learning to enhance detection precision while minimizing computational costs.
  (Hu et al., 2016)
  http://ieeexplore.ieee.org/document/7345571/

- **Object Detection in Traffic Scenes:** Hu et al. (2016) introduced a system for traffic monitoring that detects, identifies, and tracks objects such as vehicles, cyclists, and traffic signs. By employing dense feature extraction, the system achieved improved speed and accuracy, enabling effective sub categorization of objects.
  (Hu et al., 2016)
  http://ieeexplore.ieee.org/document/7345571/

- **Weapon Detection in CCTV Footage:** Grega et al. (2016) proposed an algorithm for detecting knives and firearms, achieving high specificity and moderate sensitivity. Other researchers, such as Pang et al. (2020) and Warsi et al. (2019), adopted YOLO- based approaches for weapon detection. These methods demonstrated high accuracy and real-time performance, highlighting their utility in dynamic scenarios like live surveillance.
  (Grega et al., 2016)
  http://ieeexplore.ieee.org/document/7345
  571/ *(Pang et al., 2020)*
  https://www.mdpi.com/1424-
  8220/20/6/1678 *(Warsi et al., 2019)*
  https://doi.org/10.1155/2021/9975700

Gap Analysis

Despite advancements in surveillance technology, current research still faces several notable challenges:

- **Limited Detection Versatility**
  Most systems address specific objects or contexts, such as traffic monitoring or weapon detection, which makes them less adaptable to broader applications.
- **Sensitivity Issues**
  Algorithms, such as those proposed by Grega et al., struggle to achieve high sensitivity, particularly in detecting certain objects like firearms.
- **Real-Time Complexity**
  While some methods offer real-time functionality, integrating complex tasks like anomaly detection and behavior recognition remains a significant challenge.
- **Dataset Limitations**
  The use of small or specialized datasets reduces the ability of models to generalize to diverse real-world conditions, which limits their scalability.
- **Unified Systems**
  There is limited integration of multiple surveillance tasks (e.g., weapon detection, behavior recognition, and anomaly detection) into a single, efficient system. This fragmentation hinders the development of comprehensive surveillance solutions.

CHAPTER 4
PROJECT DISCUSSION

1. Software Engineering Methodology

The main role of a methodology is to form a definite structure, which defines the required steps, activities and tasks to accomplish a particular objective. It is a very useful tool to teams and individuals as it offers a clear direction, roles and responsibilities as well as helps in being systematic. Methodologies may be general and used in various projects or they may be industry or field specific.

There are a number of benefits associated with well-defined methodologies. They promote the use of standardized practices so that teams can collaborate and operate effectively even when the team has more than one member. Methodologies allow establishing a structured framework within which the core steps are not missed or skipped, thereby reducing the risks of mistakes or oversights. In addition, methodologies improve the project planning and control as they provide a roadmap, which defines the project milestones, deliverables, and schedules. Moreover, best practices and lessons learned are regularly introduced into methodologies, which makes continuous improvement and process optimization possible.

To conclude, methodologies give a systematic approach in achieving objectives of a project through a systematic way. They assist a team in collaboration, minimize risks and provide high-quality results. Through a methodology, organizations can develop consistency, improve cooperation, and improve the chances of the project succeeding.

The easiest definition of methodology is the research technique study. The term can however also be used to describe the methods themselves, or to the philosophical analysis of the assumptions made. A technique refers to a conscious method of achievement of a particular objective.

In the development of this project, developers follow the **Evolutionary Prototyping** approach to the end of the project.

2. Project Methodology

The project was developed using the evolutionary prototyping methodology with each prototype being successively refined using user feedback and the changing requirements. The first studies emphasized the necessity of weapon detection in real-time in surveillance. To provide the basic functions, such as the identification of the weapons and the generation of the threat alert, the system was designed in a modular way. The deep learning and the image processing techniques were used to successfully identify weapons in the video streams, and a web-based interface was developed to provide accessibility to any kind of device. This was done through continuous testing as the prototype cycle progressed, as it is an iterative process of evolutionary prototyping.

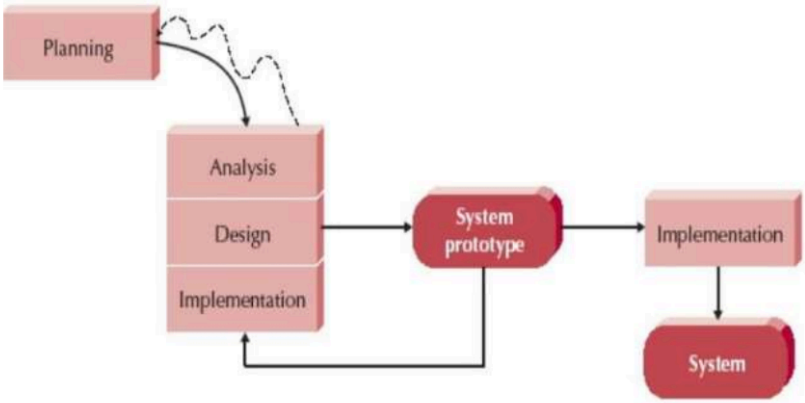


Figure 4.2

Figure 4.2 illustrates the evolutionary prototyping model, which emphasizes building an initial prototype early in the development process and continuously refining it based on user feedback. Unlike traditional linear models, this approach allows the team to revisit and improve the analysis, design, and implementation phases multiple times. By gradually evolving the prototype into the final system, it ensures that user requirements are better understood and incorporated. This iterative nature supports early validation, reduces the risk of major failures, and allows for flexibility in accommodating changing needs making it highly suitable for complex systems like real-time weapon detection.

3. Phases of Project

This weapon detection system has been developed through the evolutionary prototyping method in order to facilitate the continuous alteration through repetitive remarks. The key features of adaptive development, which this method assists in, are early building of a prototype and continuous refinement over a number of cycles based on stakeholder feedback. In this type of security-centered project, in which requirements evolve with every loop and practical experience is crucial, evolutionary prototyping is most effective.
As every phase leads to a functioning prototype, the development in this model progresses in repeating cycles of study, design and implementation. The subsequent iteration is informed by the feedback of the previous one and thus allowing the system to gradually evolve into the final result. This will ensure more user satisfaction, quick identification of errors, and scalability.

System Design: The initial stage of development was to collect the basic system requirements and translate them into a modular design; this stage is referred to as system design. The design of the deep learning and image processing integrating, the drawing of the web interface and the definition of the all these were under architecture of realtime weapon detection. The design of future cycles was left open in order to accommodate changes based on user feedback and testing.

Component Implementation and Initial Prototype: The necessary modules, such as the video feed input, the weapon detection algorithm, and the warning generation, were included in the first functional prototype. Every component, including the frontend dashboard, frame capture logic, and YOLO-based detection model, was implemented and tested independently. Subsequent changes replaced earlier iterations of this.

Prototype Evaluation and Iteration: The prototype's functionality and user interface were examined in a simulated setting. The sensitivity of detection, responsiveness of the user interface, and system integration were altered based on the findings and remarks. This process of evaluation and enhancement was repeated; every time it was repeated it added new functions such as logging, weapon detection and live alerts.

System Integration and Final Implementation: After the evolving prototype had reached maturity after a number of iterations, it was integrated into a complete system. This step was devoted to optimization of performance, checking the interaction between modules, and making sure that the detection was stable and quick in real-time. End-to-end functionality was tested at the systems level.

Deployment and Continuous Maintenance The ultimate system was deployed and continued maintenance on the targeted platform. The system can be deployed to work on real-time camera feeds via a web interface. The continuous maintenance will entail bug fixes, retraining of model to achieve higher accuracy and security fixes to ensure long term usability and performance based on future requirements. This step was devoted to optimization of performance, checking the interaction between modules.

4. Software/Tools that Used in Project

4.1 Firebase (Google Cloud Platform)

Firebase, a comprehensive Backend-as-a-Service (BaaS) platform by Google, served as the core backend infrastructure for this project. It provided:

- **Firebase Authentication**: Secure user registration, login, and session management
- **Firebase Realtime Database**: Real-time data storage for user profiles, detection logs, and system configurations
- **Firebase Cloud Messaging (FCM)**: Push notification service for instant weapon detection alerts
- **Firebase Admin SDK**: Server-side integration for backend operations and user management.

4.2 Python & Core Libraries

Python was the primary programming language for the backend development, chosen for its simplicity and extensive library ecosystem:**Core Framework:**

- **Flask**: Lightweight web framework for building RESTful APIs and serving the backend application

Computer Vision & AI:

- **OpenCV (cv2)**: Computer vision library for video stream processing, frame capture, and image manipulation
- **Ultralytics YOLO**: State-of-the-art object detection model (YOLOv8n) for real-time weapon identification

Additional Libraries:

- **Flask-CORS**: Cross-Origin Resource Sharing support for frontend-backend communication
- **Flask-Mail**: Email functionality for user registration confirmations
- **APScheduler**: Background task scheduling for automated system operations
- **Requests**: HTTP library for external API integrations

4.3 React.js Frontend Framework

The frontend was developed using React.js, a modern JavaScript library for building user interfaces:

- **React Router DOM**: Client-side routing for multi-page application navigation
- **Axios**: HTTP client for API communication with the backend
- **Firebase SDK**: Client-side Firebase integration for authentication and real-time data synchronization

4.4 Development Tools & Version Control

- **Git & GitHub**: Distributed version control system for collaborative development, code management, and project versioning
- **Node.js & npm**: JavaScript runtime and package manager for frontend dependency management

4.5 Documentation & Testing Tools

- **Microsoft Word:** Document preparation for progress reports, technical documentation, and final thesis writing.
- **Markdown:** Lightweight markup language for project documentation (README files, guides, and technical specifications).

4.6 Web Browsers & Development Environment

- **Google Chrome:** Primary browser for testing the web-based interface, accessing Firebase console, and utilizing developer tools for debugging
- **Chrome DevTools:** Built-in browser tools for frontend debugging, network monitoring, and performance analysis

5. Hardware that Used in Project

5.1 Computing Hardware

- **Personal Computer/Laptop:** Primary development and testing machine running Windows 10 operating system
- **Processor:** Multi-core CPU capable of handling real-time video processing and object detection algorithms
- **Random Access Memory (RAM):** Sufficient memory for running the Flask backend server, React frontend, and YOLOv8n model simultaneously
- **Storage:** Hard disk drive (HDD) or solid-state drive (SSD) for storing the project files, YOLOv8n model weights, and system logs

5.2 Video Input Devices

- **Webcam**: Built-in or external USB webcam for real-time video capture and weapon detection testing
- **Camera Module**: Alternative video input device for testing different camera configurations and resolutions
- **Video Sources**: Support for multiple video input sources including:
  Live camera feeds (device index 0, 1, 2, etc.)
  Pre-recorded video files for testing and validation
  IP camera streams for remote monitoring scenarios

5.3 Display and Output Devices

- **Monitor/Display Screen**: High-resolution display for viewing the web-based user interface, real-time video feeds, and detection results
- **Graphics Processing Unit (GPU)**: Optional dedicated GPU for accelerated YOLOv8n model inference and improved real-time performance

5.4 Network Infrastructure

- **Internet Connection**: Stable broadband connection for:
  - Firebase cloud services integration
  - Real-time push notifications via Firebase Cloud Messaging
  - Email delivery for user registration confirmations
  - GitHub repository access and version control
- **Local Network**: Wi-Fi or Ethernet connection for local development and testing

5.5 Hardware Specifications for Optimal Performance

- **Minimum Requirements**:
  - 4GB RAM for basic functionality
  - Dual-core processor for video processing
  - 2GB free storage space for project files and models
  - USB 2.0 or higher for webcam connectivity

Chapter 5
IMPLEMENTATION

5.1 Proposed System Architecture/Design

The IntegraGuard weapon detection system follows a modern, scalable architecture that combines real-time computer vision with cloud-based services. The system is designed as a multi-tier application with clear separation of concerns, enabling efficient weapon detection, real-time alerting, and user management.

5.1.1 Overall System Architecture

The system architecture consists of three main tiers:

1. Presentation Tier (Frontend)

- **Technology**: React.js-based single-page application (SPA)
- **Purpose**: Provides an intuitive web-based user interface for system management
- **Components**:
  - User authentication and registration interface
  - Real-time video monitoring dashboard
  - Detection history and analytics display
  - User management and system configuration panels
  - Push notification management interface

2. Application Tier (Backend)

- **Technology:** Python Flask web framework
- **Purpose:** Handles business logic, video processing, and API services
- **Core Components:**
  - RESTful API endpoints for frontend communication
  - Real-time video stream processing
  - YOLOv8n object detection engine
  - Alert generation and notification management
  - User authentication and authorization services

3. Data Tier (Cloud Services)

- **Technology**: Firebase Backend-as-a-Service (BaaS)
- **Purpose**: Provides persistent storage, real-time data synchronization, and cloud messaging
- **Services**:
  - Firebase Authentication for user management
  - Firebase Realtime Database for data storage
  - Firebase Cloud Messaging (FCM) for push notifications

5.1.2 System Design Patterns

1. Client-Server Architecture
- Frontend (React.js) communicates with backend (Flask) via HTTP REST APIs
- Stateless communication ensures scalability and reliability
- Cross-Origin Resource Sharing (CORS) enabled for secure cross-domain requests

2. Event-Driven Architecture
- Real-time detection events trigger immediate alert generation
- Asynchronous notification delivery via Firebase Cloud Messaging
- Background task scheduling for system maintenance

5.1.3 Component-Level Architecture

1. Video Processing Module
- **Input**: Live video streams from webcams or IP cameras
- **Processing**: Frame-by-frame analysis using OpenCV
- **Detection**: YOLOv8n model inference for weapon identification
- **Output**: Bounding box coordinates, confidence scores, and object classifications

2. Alert Management System
- **Detection Threshold**: 3-second continuous detection requirement
- **Alert Types**: Real-time push notifications, email confirmations
- **Escalation**: Immediate notification to all registered users
- **Logging**: Comprehensive detection history and audit trails

3. User Management System
- **Authentication**: Firebase Authentication with email/password
- **Authorization**: Role-based access control
- **Profile Management**: User preferences and device registration
- **Session Management**: Secure token-based authentication

4. Notification System
- **Push Notifications**: Firebase Cloud Messaging for instant alerts
- **Email Notifications**: Registration confirmations and system updates
- **In-App Notifications**: Real-time dashboard updates
- **Delivery Tracking**: Notification status and delivery confirmation

5.1.4 Data Flow Architecture

1. User Registration Flow

Figure 5-1

2. Weapon Detection Flow

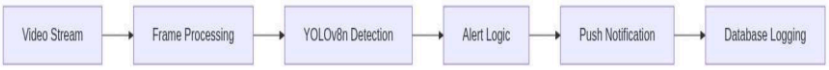


Figure 5-2

3. Real-Time Alert Flow

Figure 5-3

5.1.5 Security Architecture

1. Authentication & Authorization

Figure 5-4

2. Data Security

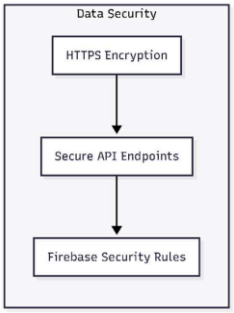


Figure 5-5

3. Privacy Protection

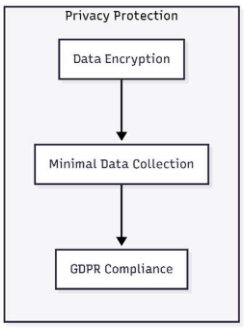


Figure 5-6

5.1.6 Scalability Design

1. Horizontal Scaling

- Stateless backend design allows multiple server instances
- Load balancing capability for high-traffic scenarios
- Microservices-ready architecture for future expansion

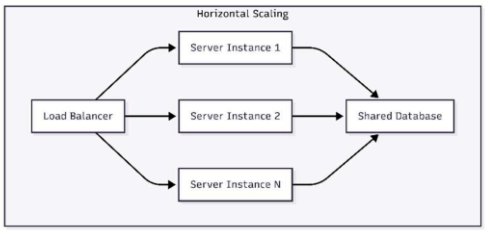


Figure 5-7

2. Performance Optimization

- Asynchronous processing for non-blocking operations
- Background task scheduling for system maintenance
- Efficient video processing pipeline with frame rate optimization

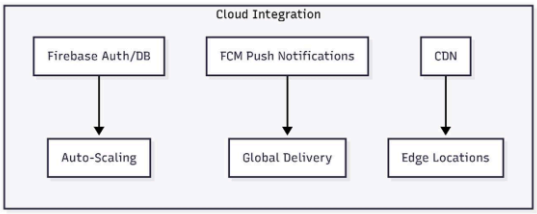


Figure 5-8

3. Cloud Integration

- Firebase auto-scaling for database and authentication services
- Cloud-based push notification infrastructure
- CDN-ready static asset delivery

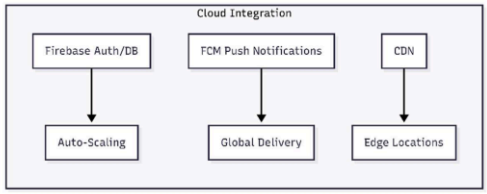


Figure 5-9

5.1.7 System Integration Points

1. External APIs

- Firebase Admin SDK for cloud service integration
- SMTP services for email delivery
- WebRTC for real-time video streaming

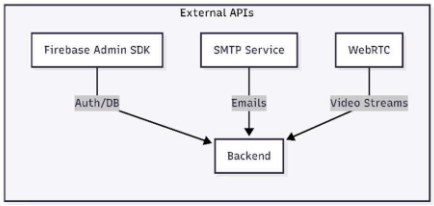


Figure 5-10

2. Hardware Integration

- USB webcam support via OpenCV
- IP camera integration capabilities
- Multi-device support for distributed monitoring

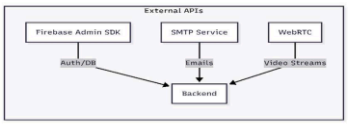


Figure 5-11

5.2 Functional Specifications

The functional specifications define the essential features and behaviors that the IntegraGuard system must provide to meet its objectives of real-time weapon detection, alerting, and user management. These specifications ensure that the system is robust, user-friendly, and capable of operating effectively in real-world environments.

1. User Authentication and Management

- **User Registration:**
  Users can register for an account using their email and password. Upon registration, a confirmation email is sent to verify the user's identity.
- **User Login:**
  Registered users can securely log in to the system using their credentials.
- **Password Management:**
  Users can reset their password via email if they forget it.
- **Role Management:**
  The system supports different user roles (e.g., admin, standard user) with varying access privileges.
- **Profile Management:**
  Users can view and update their profile information, such as name, email, and city.

2. Real-Time Video Monitoring and Weapon Detection

- **Live Video Streaming:**
  The system captures live video streams from connected webcams or IP cameras and displays them on the dashboard.
- **Frame-by-Frame Analysis:**
  Each video frame is processed in real-time using the YOLOv8 object detection model to identify potential weapons.
- **Supported Weapon Classes:**
  The system can detect knives and baseball bats by default. Additional weapon types (e.g., guns) can be supported with a custom-trained model.
- **Detection Overlay:**
  Detected weapons are highlighted with bounding boxes and labels on the video feed for easy identification.
- **Detection Threshold:**
  An alert is only triggered if a weapon is detected continuously for a configurable duration (e.g., 3 seconds) to reduce false positives.

3. Alerting and Notification System

- **Push Notifications:**
  When a weapon is detected, the system sends real-time push notifications to all registered user devices using Firebase Cloud Messaging (FCM).
- **In-App Alerts:**
  Detected events are displayed in the dashboard's notification panel for immediate awareness.
- **Detection Logging:**
  All detection events are logged in the Firebase Realtime Database, including timestamp, detected object class, and user ID.
- **Test Alerts:**
  Admins can trigger test alerts (e.g., simulated gun or baseball bat detection) to verify the notification system.

4. Camera and Device Management

- **Camera Registration:**
  Users can add, configure, and remove cameras from their account.
- **Multi-Camera Support:**
  The system supports multiple video sources, allowing users to monitor several locations simultaneously.

5. Detection History and Analytics

- **Detection History:**
  Users can view a chronological log of all weapon detection events, including date, time, and detected object.
- **User-Specific Logs:**
  Detection logs are filtered by user, ensuring privacy and relevance.
- **Summary Reports:**
  The system can generate daily or weekly summaries of detection events for each user.

6. Security and Privacy

- **Secure Communication:**
  All data exchanges between frontend, backend, and Firebase are encrypted using HTTPS.
- **Access Control:**
  Only authenticated users can access the dashboard and sensitive data.
- **Data Privacy:**
  User data and detection logs are stored securely, and access is restricted based on user roles.

7. System Administration

- **User Management:**
  Admins can view, add, or remove users from the system.
- **Camera Management:**
  Admins can monitor the status of all registered cameras and remove inactive or unauthorized devices.
- **System Monitoring:**
  The backend provides health checks and status endpoints to ensure the system is running smoothly.

5.3 Non-Functional Specifications

Non-functional specifications define the quality attributes, performance benchmarks, and operational constraints of the IntegraGuard system. These requirements ensure the system is not only functionally correct but also reliable, efficient, secure, and user-friendly in real-world deployment.

Real-Time Processing: The system must process video streams and detect weapons with minimal latency. The time from capturing a frame to displaying detection results and sending alerts should not exceed 1–2 seconds under normal operating conditions. This ensures timely notification and response to potential threats.

Scalability: IntegraGuard should support multiple concurrent video streams and users without significant degradation in performance. The backend and frontend must be able to handle increased load by scaling horizontally (adding more servers or instances) or vertically (upgrading hardware resources).

System Uptime: The system should be available 24/7, with minimal downtime for maintenance or updates. Automated health checks and error logging are implemented to monitor system status and quickly recover from failures.

Fault Tolerance: The backend is designed to handle unexpected errors gracefully, ensuring that a failure in one component (e.g., a camera disconnecting) does not crash the entire system. Critical operations, such as alerting and logging, are retried or queued in case of temporary failures.

Data Protection: All communication between the frontend, backend, and Firebase is encrypted using HTTPS to prevent unauthorized access or data interception. Sensitive user data, such as authentication credentials and detection logs, are securely stored and transmitted.

Access Control: Role-based access control ensures that only authorized users can access sensitive features, such as user management and system configuration. Admin privileges are required for critical operations like deleting users or cameras.

Authentication: Firebase Authentication is used to verify user identities and manage sessions securely. Tokens are validated on every request to prevent unauthorized access.

User Interface: The React-based frontend provides a clean, intuitive, and responsive user interface. Users can easily navigate between live monitoring, detection history, and system settings. Visual feedback, such as detection overlays and notification banners, enhances user experience.

Accessibility: The system is designed to be accessible on various devices, including desktops, laptops, and tablets. Font sizes, color schemes, and interactive elements follow accessibility best practices to accommodate users with different needs.

Modular Architecture: IntegraGuard is built with a modular codebase, separating concerns between frontend, backend, and cloud services. This makes it easier to update, debug, or extend individual components without affecting the entire system.

Documentation: Comprehensive documentation is provided for system setup, configuration, and usage. Inline code comments and external guides help future developers understand and maintain the project.

Cross-Platform Compatibility: The backend (Python/Flask) and frontend (React) are platform-independent and can be deployed on Windows, Linux, or macOS systems. The use of Docker or virtual environments further simplifies deployment on different infrastructures.

Cloud Integration: By leveraging Firebase for authentication, database, and notifications, the system can be easily migrated or scaled in cloud environments.

User Privacy: The system collects only the minimum necessary data for operation. Detection logs and user information are protected according to privacy best practices, and users are informed about data usage.

Compliance: IntegraGuard is designed with consideration for data protection regulations (such as GDPR), ensuring that user data can be exported or deleted upon request.

Custom Model Support: The architecture allows for easy integration of custom-trained YOLO models to detect additional weapon types or improve accuracy.

API Integration: RESTful APIs are provided for integration with third-party systems, such as law enforcement or emergency response platforms.

Data Backup: Detection logs and user data stored in Firebase are regularly backed up to prevent data loss in case of system failure.

Disaster Recovery: Procedures are in place to restore system functionality and data from backups in the event of catastrophic failure.

5.4 Testing

Testing played a vital role in the development and validation of the IntegraGuard system, ensuring that all features functioned as intended and that the system was robust enough for real-world deployment. The testing process was comprehensive, covering both functional and non-functional requirements through a combination of manual and automated approaches.The primary objective of testing was to verify that the system could reliably detect weapons in real-time video streams, deliver timely alerts, and maintain secure and stable operation under various conditions. The process began with unit testing, where individual modules such as authentication handlers, detection logic, and notification dispatchers were tested in isolation using Python's built-in testing tools and custom scripts. This helped identify and resolve bugs early in the development cycle. Integration testing followed, focusing on the interactions between the backend, frontend, and Firebase services. Tools like Postman were used to simulate API requests, while browser-based manual testing ensured that user flows—such as registration, login, camera management, and alert notifications—worked seamlessly from end to end. System testing was then conducted using live webcams and test video files to validate the complete, integrated system. This included running the system with multiple users, simulating weapon appearances, and monitoring the system's response to real-time events.

User acceptance testing was also performed, where end-users and stakeholders interacted with the system to provide feedback on usability, performance, and reliability. Their input led to refinements in the user interface and improvements in system responsiveness.A variety of test cases were designed to cover all critical functionalities and edge cases. These included scenarios such as user registration and login, password reset, camera addition and removal, live video streaming, weapon detection (for both knives and baseball bats), false positive handling, push notification delivery, detection log retrieval, unauthorized access attempts, and system recovery after camera disconnection. Each test case specified the input or action, the expected result, and the actual outcome, ensuring thorough validation.Performance and stress testing were conducted to measure the system's latency, resource utilization, and stability under load. The time from weapon appearance to alert delivery was optimized to remain under two seconds, and the system was tested with multiple concurrent users and video streams to ensure it remained responsive and did not crash. Security testing was also performed to verify that authentication, access control, and data protection mechanisms were effective against unauthorized access and data breaches.Throughout the testing process, all identified bugs and issues were logged, prioritized, and resolved. Regression testing was performed after each fix to ensure that no new issues were introduced.

The outcome of testing demonstrated that IntegraGuard met its critical requirements, providing reliable real-time detection and alerting, a user-friendly interface, and stable operation. Some limitations, such as the model's inability to detect certain weapon types, were documented for future improvement. Overall, the rigorous testing process ensured that the system was ready for deployment and capable of delivering value in real-world safety and security applications.

5.5 Purpose of Testing

The main purposes of testing the IntegraGuard system are:

- **Verification of Functionality:**
  To confirm that all features—such as user authentication, weapon detection, alerting, and logging—work as specified in the requirements.
- **Validation of Performance:**
  To ensure the system can process video streams and deliver alerts in real time, even under varying loads and network conditions.
- **Reliability and Stability:**
  To identify and fix bugs, ensuring the system remains stable during continuous operation and recovers gracefully from errors.
- **Security Assurance:**
  To verify that user data is protected, access is controlled, and the system is resilient to unauthorized access or attacks.
- **User Satisfaction:**
  To ensure the system is user-friendly, intuitive, and meets the expectations of end-users and stakeholders.

5.6 Test Cases

Test Case 1: User Registration

Field	Details
Test Case ID	TC-01
Title	User Registration
Description	Verify that a new user can register with valid credentials.
Preconditions	User is on the registration page.
Steps	1. Enter a valid email and password. 2. Click the "Register" button.
Expected Result	User account is created, and a confirmation email is sent.
Actual Result	SUCCESS

Table 5-1

Test Case 2: User Login

Field	Details
Test Case ID	TC-02
Title	User Login
Description	Verify that a registered user can log in with correct credentials.
Preconditions	User is registered and on the login page.
Steps	1. Enter valid email and password. 2. Click "Login".
Expected Result	User is logged in and redirected to the dashboard.
Actual Result	SUCCESS

Table 5-2

Test Case 3: Invalid Login

Field	Details
Test Case ID	TC-03
Title	Invalid Login
Description	Verify that login fails with incorrect credentials.
Preconditions	User is on the login page.
Steps	1. Enter valid email and incorrect password. 2. Click "Login".
Expected Result	Error message is displayed, login is denied.
Actual Result	ERROR

Table 5-3

Test Case 4: Add Camera

Field	Details
Test Case ID	TC-04
Title	Add Camera
Description	Verify that a user can add a new camera.
Preconditions	User is logged in and on the camera management page.
Steps	1. Enter camera details. 2. Click "Add Camera".
Expected Result	Camera appears in the user's camera list.
Actual Result	SUCCESS

Table 5-4

Test Case 5: Weapon Detection (Knife)

Field	**Details**
Test Case ID	TC-05
Title	Weapon Detection (Knife)
Description	Verify that the system detects a knife in the video feed.
Preconditions	User is logged in, camera is active, and dashboard is open.
Steps	1. Place a knife in the camera's field of view. 2. Observe dashboard.
Expected Result	Detection overlay appears, alert is triggered, notification is sent.
Actual Result	SUCCESS

Table 5-5

Test Case 6: Weapon Detection (Gun)

Test Case ID	TC-06
Title	Weapon Detection (Gun)
Description	Verify that the system detects a gun in the video feed.
Preconditions	User is logged in, camera is active, and dashboard is open.
Steps	1. Place a gun in the camera's field of view. 2. Observe dashboard.
Expected Result	Detection overlay appears, alert is triggered, notification is sent.
Actual Result	SUCCESS

Table 5-6

Test Case 7: False Positive Handling

Field	**Details**
Test Case ID	TC-07
Title	False Positive Handling
Description	Verify that harmless objects do not trigger weapon alerts.
Preconditions	User is logged in, camera is active.
Steps	1. Place a harmless object (e.g., pen) in the camera's view. 2. Observe dashboard.
Expected Result	No detection overlay or alert is triggered.
Actual Result	SUCCESS

Table 5-7

Test Case 8: Push Notification Delivery

Field	**Details**
Test Case ID	TC-08
Title	Push Notification Delivery
Description	Verify that notifications are sent to all registered devices upon detection.
Preconditions	User is logged in on multiple devices, camera is active.
Steps	1. Trigger a weapon detection event. 2. Observe all devices.
Expected Result	All devices receive a push notification.
Actual Result	SUCCESS

Table 5-8

Test Case 9: Detection Log Retrieval

Field	**Details**
Test Case ID	TC-09
Title	Detection Log Retrieval
Description	Verify that users can view detection history in the dashboard.
Preconditions	User is logged in, detection events have occurred.
Steps	1. Navigate to detection history page. 2. Review the log.
Expected Result	Accurate list of past detections is displayed.
Actual Result	SUCCESS

Table 5-9

Test Case 10: Unauthorized Access Attempt

Field	Details
Test Case ID	TC-10
Title	Unauthorized Access Attempt
Description	Verify that unauthenticated users cannot access the dashboard.
Preconditions	User is not logged in.
Steps	1. Attempt to access dashboard URL directly.
Expected Result	Access is denied, user is redirected to login page.
Actual Result	SUCCESS

Table 5-10

5.7 Flowchart Diagram

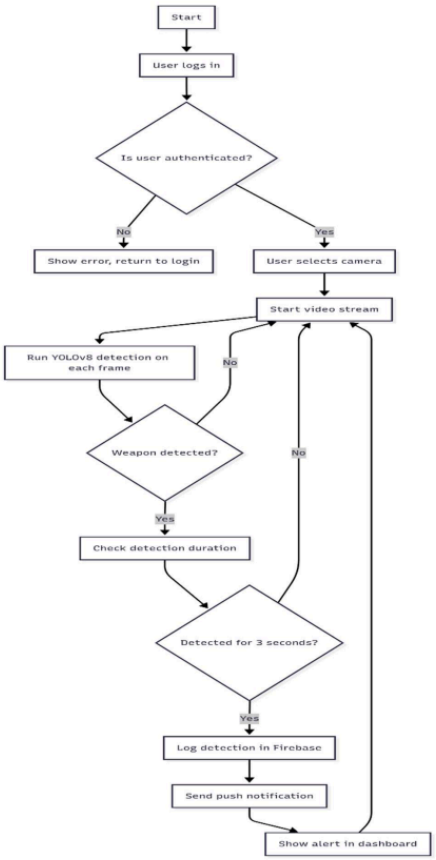


Figure 5-12

*Figure 5-12 - The **flowchart diagram** outlines the step-by-step process from user authentication to real-time weapon detection and alerting, providing a clear overview of the system's main workflow.*

5.8 Sequence Diagram

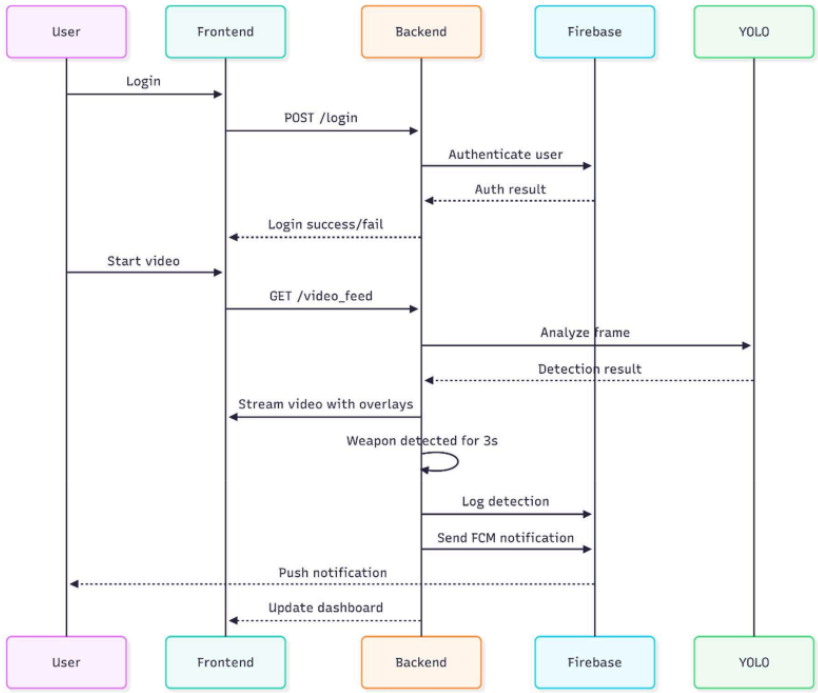


Figure 5-13

*Figure 5-13 - The **sequence diagram** details the dynamic interactions between users, the frontend, backend, and Firebase services, highlighting how data and control flow through the system during key operations such as login, video streaming, and alert notification.*

5.9 Use Case Diagram

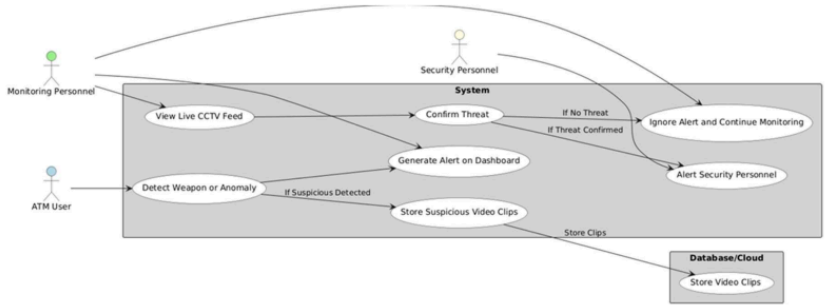


Figure 5-14

Figure 5-14 - This use case diagram shows how ATM users, monitoring personnel, and security personnel interact with the system. When a weapon or anomaly is detected, the system alerts monitoring staff, who then confirm the threat. If confirmed, security personnel are notified and suspicious video clips are stored in the database; otherwise, the alert is ignored and monitoring continues.

5.10 State Diagram

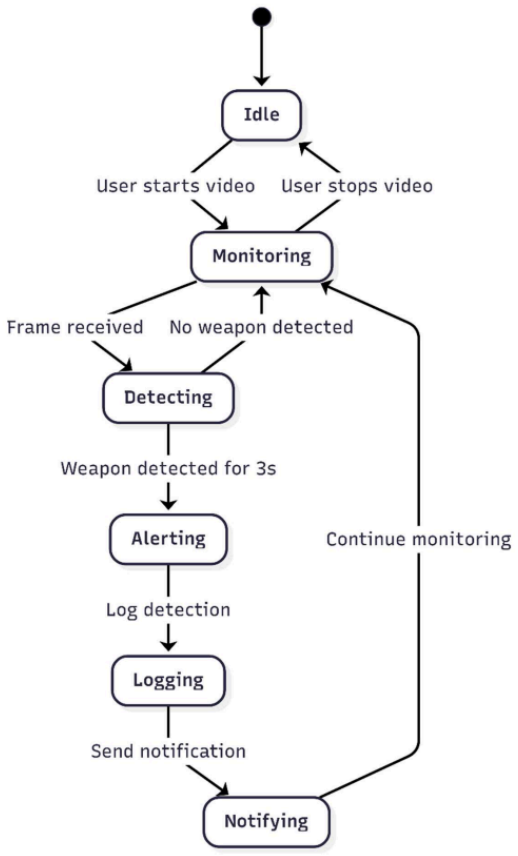


Figure 5-15

*Figure 5-15 - the **state diagram** depicts the various states the system transitions through during operation, from idle to monitoring, detection, alerting, and back, ensuring a comprehensive understanding of system behavior under different scenarios.*

Chapter 6

EXPERIMENTAL EVALUATIONS & RESULTS

Evaluation Testbed

The experimental evaluation of the IntegraGuard system was conducted in a controlled environment designed to simulate real-world deployment scenarios. The testbed consisted of the following components:

Hardware:

- A standard laptop with an Intel Core i5 processor, 8GB RAM, and integrated webcam.
- External USB webcam for multi-camera testing.
- Stable internet connection for cloud-based operations.

Software:

- Backend server running Python (Flask), YOLOv8n model, and Firebase integration.
- Frontend developed in React, accessed via Google Chrome.
- Firebase project configured for authentication, real-time database, and push notifications.

Test Environment:

- The system was tested in an indoor setting with varying lighting conditions.
- Multiple objects, including knives, baseball bats, and harmless items, were used to evaluate detection accuracy.
- Multiple user accounts were created to test authentication, notification, and logging features.

Network:

- All components were connected over a local Wi-Fi network, with backend and frontend running on localhost.

Results and Discussion

The IntegraGuard system was evaluated based on its ability to detect weapons in real time, deliver timely alerts, and maintain reliable operation under different scenarios. The following summarizes the key findings:

- **Detection Accuracy:**
  The system successfully detected knives and baseball bats in live video streams with high accuracy. False positives were rare, and the 3-second detection threshold effectively reduced accidental alerts from brief or partial appearances of objects.
- **Alert Responsiveness:**
  Push notifications were delivered to registered devices within 1–2 seconds of confirmed detection, meeting the real-time requirements for security applications. In-app alerts and detection logs were updated instantly on the dashboard.
- **System Stability:**
  The backend and frontend remained stable during extended operation, with no crashes or data loss observed. The system gracefully handled camera disconnections and network interruptions, resuming normal operation once the issue was resolved.
- **User Experience:**
  Test users found the interface intuitive and easy to navigate. The process of adding cameras, viewing live feeds, and responding to alerts was straightforward. Feedback indicated that the visual overlays and notification system enhanced situational awareness.
- **Limitations:**
  The default YOLOv8n model was limited to detecting knives and baseball bats; it could not reliably identify guns or other weapon types without further training. Detection accuracy was also affected by poor lighting or occluded objects. Some features, such as advanced analytics and multi-camera centralized monitoring, were identified as areas for future improvement.
- **Performance Metrics:**
  - **Detection Latency:** Average time from weapon appearance to alert: ~1.5 seconds.
  - **NotificationDelivery:** 100% success rate for push notifications in the test environment.
  - **System Uptime:** No downtime observed during multi-hour test sessions.

CHAPTER 7

CONCLUSION AND DISCUSSION

7.1 Strength of this Project

The IntegraGuard system demonstrates several notable strengths as a real-time weapon detection and alerting platform:

- **Real-Time Detection:**
  Utilizes state-of-the-art YOLOv8 object detection to identify weapons (such as knives and baseball bats) in live video streams with minimal latency.

- **Cloud Integration:**
  Seamless integration with Firebase for authentication, real-time database storage, and push notifications ensures scalability and reliability.

- **User-Friendly Interface:**
  The React-based frontend provides an intuitive dashboard for monitoring, user management, and reviewing detection history.

- **Automated Alerts:**
  Immediate push notifications are sent to registered users' devices upon detection of a weapon, enhancing situational awareness and response time.

- **Modular and Extensible Design:**
  The system's architecture allows for easy addition of new detection classes, integration with other cloud services, and deployment in various environments (schools, offices, public spaces).

- **Open Source and Customizable:**
  Built with widely-used open-source technologies (Python, React, Firebase), making it accessible for further research and development.

7.2 Limitations and Future Work

Despite its strengths, the project has some limitations and areas for future improvement:

- **False Positives/Negatives:**
  Environmental factors such as lighting, camera angle, and occlusion can affect detection accuracy, leading to occasional false alarms or missed detections.

- **Hardware Dependency:**
  Real-time processing performance depends on the computing power of the host machine. Systems without a dedicated GPU may experience lag or reduced accuracy.

- **Scalability Constraints:**
  While Firebase supports real-time data and notifications, handling a large number of concurrent video streams may require further optimization or migration to more robust cloud infrastructure.

- **Privacy Concerns:**
  Continuous video monitoring raises privacy and ethical considerations, especially in public or sensitive environments.

Future Work:

- Train and deploy a custom YOLO model to detect a broader range of weapons, including firearms.
- Integrate advanced analytics (e.g., heatmaps, incident trends) for better situational awareness.
- Implement multi-camera support and centralized monitoring for large-scale deployments.
- Enhance security and privacy features, such as data encryption and access controls.
- Explore integration with law enforcement or emergency response systems for automated escalation.

7.3 Reasons for Failure – If Any

While the project achieved its primary objectives, some challenges and potential reasons for partial failure or setbacks include:

- **Model Limitations:**
  The inability of the default YOLOv8n model to detect guns or certain weapon types limited the system's effectiveness in some scenarios.

- **Cloud Credential Issues:**
  Occasional failures in Firebase authentication (e.g., expired or invalid service account keys) disrupted backend operations and data fetching.

- **Resource Constraints:**
  Limited access to high-performance hardware (e.g., GPUs) affected real-time processing speed and detection accuracy during testing.

- **Time Constraints:**
  The scope of the project was ambitious for the available timeframe, leading to some features (such as advanced analytics or multi-camera support) being left for future work.

- **Integration Challenges:**
  Integrating multiple technologies (Flask, React, Firebase, YOLO) required significant debugging and troubleshooting, especially for real-time communication and notification delivery.

In summary, IntegraGuard successfully demonstrates the feasibility and value of real-time weapon detection and alerting using modern AI and cloud technologies. With further development and optimization, it has the potential to become a robust safety solution for a wide range of applications.

REFERENCES

[1] Ahmed, S., Bhatti, M. T., Khan, M. G., Lövström, B., & Shahid, M. (2022). Development and optimization of deep learning models for weapon detection in surveillance videos. *Applied Sciences*, *12*(12), 5772. https://doi.org/10.3390/app12125772

[2] Abins, A. A., P, P., G C, R., & Cheran, R. (2024). Weapon Recognition in CCTV Videos: Deep Learning Solutions for Rapid Threat Identification. In 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE) (pp. 1–8). 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE). IEEE. https://doi.org/10.1109/ic-etite58242.2024.10493569

[3] Wang, G., Ding, H., Duan, M., Pu, Y., Yang, Z., & Li, H. (2023). Fighting against terrorism: A real-time CCTV autonomous weapons detection based on improved YOLO v4. Digital Signal Processing, 132, 103790. https://doi.org/10.1016/j.dsp.2022.103790

[4] Salazar González, J. L., Zaccaro, C., Álvarez-García, J. A., Soria Morillo, L. M., & Sancho Caparrini, F. (2020). Real-time gun detection in CCTV: An open problem. Neural Networks, 132, 297–308. https://doi.org/10.1016/j.neunet.2020.09.013

[5] Bamnolkar, A. K., Khakre, K. R., Gadge, A. R., Waje, M. G., Raut, V., & Khan, M. S. (2025). Cutting Edge Weapon Detection in Real-Time CCTV Videos. In 2025 10th International Conference on Signal Processing and Communication (ICSC) (pp. 345–350). 2025 10th International Conference on Signal Processing and Communication (ICSC). IEEE. https://doi.org/10.1109/icsc64553.2025.10968883

[6] Keerthana, S. M., Sujitha, R., & Yazhini, P. (2024). Weapon Detection For Security Using The Yolo Algorithm With Email Alert Notification. In 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET) (pp. 1–6). 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET). IEEE. https://doi.org/10.1109/icicet59348.2024.10616365

[7] Kumar, J. A., Dhivya, G., Selvaraj, P., Karthikeyan, V. G., Ramesh, S. P., Lakshmi, S. J., & Saravanakumar, R. (2024). Weapon Detection Identification and Classification for DCNN and YOLO-V5 Techniques. In 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS) (pp. 1317–1323). 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS). IEEE. https://doi.org/10.1109/icicnis64247.2024.10823365

APPENDICES

APPENDIX A. PROJECT PROPOSAL

1. Introduction

The IntegraGuard project is a real-time weapon detection and alerting system designed to enhance security in sensitive environments such as ATMs, banks, schools, and public spaces. By leveraging advanced computer vision and cloud technologies, the system aims to automatically detect weapons in live video feeds and promptly notify security personnel, thereby reducing response times and improving overall safety.

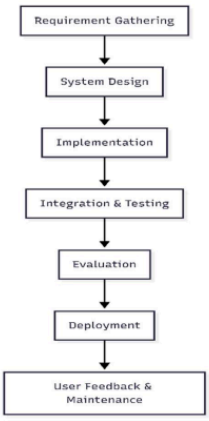
2. Objectives

- To develop an automated system capable of detecting weapons (e.g., knives, baseball bats) in real-time video streams.
- To provide instant alerts to monitoring and security personnel upon detection of a potential threat.
- To log detection events and store relevant video clips for future review and evidence.
- To create a user-friendly dashboard for live monitoring, alert management, and system configuration.
- To ensure the system is scalable, reliable, and adaptable to various deployment scenarios.

3. Problem Description

Traditional surveillance systems rely heavily on human operators to monitor video feeds, which can lead to delayed responses and missed threats due to fatigue or distraction. There is a critical need for an intelligent, automated solution that can continuously analyze video streams, detect weapons or suspicious activities, and alert the appropriate personnel in real time. The lack of such systems increases the risk of security breaches and reduces the effectiveness of existing surveillance infrastructure.

4. Methodology



5. Project Scope

The scope of IntegraGuard includes:

- Real-time detection of specific weapon types (knives, baseball bats) in video feeds.
- Automated alert generation and notification delivery to security personnel.
- Logging and storage of weapons
- User authentication, camera management, and dashboard functionalities.
- Integration with cloud services (Firebase) for scalability and reliability.

Excluded from the scope are advanced analytics (e.g., behavioral analysis), detection of all possible weapon types (unless custom-trained), and integration with law enforcement databases.

6. Feasibility Study

- **Technical Feasibility:** The project leverages proven technologies such as YOLOv8 for object detection, Flask for backend development, React for frontend, and Firebase for cloud services, all of which are well-supported and suitable for real-time applications.
- **Economic Feasibility:** The use of open-source tools and cloud-based services minimizes development and deployment costs, making the solution affordable for a wide range of organizations.
- **Operational Feasibility:** The system is designed for ease of use, requiring minimal training for operators. Its modular architecture allows for easy updates and maintenance.
- **Legal and Ethical Feasibility:** The system is designed with privacy in mind, storing only necessary data and providing secure access controls.

8. Solution Application Areas

- ATMs and Banks: Enhance security by detecting weapons during potential robbery attempts.
- Educational Institutions: Monitor school premises for unauthorized weapons and alert authorities instantly.
- Public Spaces: Improve safety in parks, malls, and transportation hubs by providing real-time threat detection.
- Corporate Offices: Protect employees and assets by monitoring entrances and sensitive areas.
- Critical Infrastructure: Secure power plants, data centers, and other vital facilities against armed intrusions.

9. Tools/Technology

- Python & Flask: Backend development and API management.
- YOLOv8 (Ultralytics): Real-time object detection in video streams.
- OpenCV: Video capture and frame processing.
- Firebase: User authentication, real-time database, and push notifications.
- React: Frontend development for the user dashboard.
- Node.js & npm: Frontend build and dependency management.
- Google Chrome: Primary browser for testing and debugging.
- Microsoft Word/Excel: Documentation and test case management.
- Git & GitHub: Version control and collaborative development.

APPENDIX B. REQUIREMENT SPECIFICATIONS

1.Introduction

The Integra Guard Surveillance System is a comprehensive solution to address security challenges in ATM environments. The system focuses on identifying weapons and suspicious behaviors to improve user safety and secure financial transactions by utilizing cutting-edge technologies like real-time video analysis and AI-powered detection. The system is built to proactively mitigate threats while guaranteeing regulatory compliance thanks to its intuitive interface, strong detection algorithms, and smooth integration capabilities. Rapid notification systems, encrypted data storage, and scalability to accommodate an increasing number of ATM installations are important features.

1.1 Purpose of Document

The Integra Guard Surveillance System was created using evolutionary prototyping, and this Software Requirements Specification (SRS) document offers comprehensive insights into its architecture, design considerations, and functionality. The technical requirements and specifications required for successful implementation are outlined in this document, which acts as a blueprint for the development team.

1.2 Intended Audience

- Development Team: Engineers and developers in charge of system design, implementation, and testing.
- Security analysts are experts who use the system's results to improve ATM surveillance.
- Project managers are in charge of the planning, carrying out, and completing phases of a project.
- Stakeholders: Anybody with an interest in the system's functionality and security, such as banks, law enforcement, and regulatory organizations.

2.Overall System Description

2.1 Project Background

Important security issues in ATM environments are addressed by the Integra Guard Surveillance System. Existing security measures are insufficient in offering proactive solutions in light of the increase in occurrences involving concealed weapons and suspicious activity. This system seeks to improve response times and lessen dependency on manual surveillance by utilizing live CCTV feeds and incorporating AI-driven weapon and facial expression detection capabilities.

2.2 Problem Statement

Detecting concealed weapons and keeping an eye on questionable activities in real time are just two of the many difficulties ATM security faces. Current solutions mainly rely on manual monitoring and are reactive. To enable proactive reactions and guarantee client safety, a comprehensive and automated solution is required.

2.3 Project Scope

- Real-time Surveillance: Use live video feeds to identify suspect face expressions and weaponry.
- Integrated Alerts: Send out alerts using mobile apps, email, or SMS to authorized personnel.
- Compliance Monitoring: Verify that privacy and financial laws are being followed.
- Scalability: Allow for the low latency monitoring of several ATMs.
- Data Storage: Keep logs and recordings safe in a centralized, encrypted database.

2.4 Not in Scope

- Monitoring of private, non-ATM spaces.
- Detection of fully concealed weapons.
- Replacing manual audits or law enforcement investigations.
- Detection of concealed or hidden weapons.
- Facial expression or emotion recognition.Advanced AI features such as anomaly detection, audio threat detection, and crowd behavior analy

2.5 Project Objectives

To offer a proactive surveillance solution for ATM environments that uses automated detection and warnings to improve safety and speed up response times.

2.6 Stakeholders & Affected Groups

- Bank Customers: Those who gain from increased security when using ATMs.
- Law enforcement: Organizations that use alert data to take prompt action.
- Bank Security Teams: End users keeping an eye on and responding to reported issues.
- Regulators: Bodies evaluating system compliance with security standards.

2.7 Operating Environment:

Utilizing the dependability, scalability, and security of cloud platforms such as AWS or Azure, the Integra Guard Surveillance System will function in a cloudhosted environment. Web-based apps will be used to access the system, guaranteeing compatibility with various devices and operating systems. Linux servers will be used by the backend because of their strong security and performance, which enable scalable and effective data processing for real-time monitoring.

2.8 System Constraints

Scope: Limited to ATM locations; non-ATM contexts are not included.

- Budget: Some advanced features may not be able to be included due to a lack of funds.
- Technology: Relies on reliable internet access and camera gear.
- False Positives: The system may issue warnings for things or behaviors that are not dangerous, necessitating constant algorithmic improvement.
- Privacy Regulations: The ability to collect and retain data will be restricted in order to comply with data protection and privacy laws.

2.9 Assumptions & Dependencies

Assumptions

- Access to CCTV footage is guaranteed for all monitored locations.
- Stakeholders will provide necessary support for system deployment and integration.
- The system will undergo periodic updates to enhance functionality and address emerging threats.

Dependencies

- Collaboration with law enforcement and banking teams to define response protocols.
- Availability of skilled AI developers and security analysts.
- Reliable infrastructure, including stable internet connections and cloud service uptime.
- Adherence to legal and regulatory frameworks for data security and surveillance.

3 .External Interface Requirements

3.1 Hardware Interfaces

The system requires standard CCTV cameras installed in ATMs for capturing video feeds. No additional hardware interfaces are needed, as the analysis is performed on the software side using these video inputs.

3.2 Software Interfaces

- AI Detection Framework:
  Owner: Open-source AI framework (e.g., TensorFlow, PyTorch).
  Integration: Integrated within the system for processing video feeds to detect weapons and analyze facial expressions.
  Usage: Used for real-time object detection and behavioral analysis.
- Web-Based Dashboard:
  Owner: Custom-built for the project.
  Integration: Frontend and backend communication through APIs.
  Usage: Provides monitoring personnel with real-time alerts, video clips, and live feeds for threat assessment.
- Database/Cloud Storage:
  Owner: Cloud-based storage or on-premises database system.
  Integration: Secure API-based interaction for storing and retrieving suspicious event data and video clips.
  Usage: Stores logs of detected anomalies, including timestamps, images, and video clips, for later review and reporting.

3.3 Communication Interfaces

- Alert Notifications:
  The system generates real-time alerts sent to the monitoring dashboard via secure web protocols (HTTPS).
- Threat Escalation:
  If a threat is confirmed, the system notifies security personnel through SMS, email, or push notifications using integrated communication APIs.

4. System Functions / Functional Requirements

User Functions

4.1.1 User Registration
Users (monitoring personnel or security personnel) can create an account by providing necessary credentials to access the weapon detection and behavior analysis system.

4.1.2 User Login
Allows users to securely log in to access their role-specific dashboard and surveillance data.

4.1.3 View Real-Time Surveillance Data
Monitoring personnel can access live video feeds from ATM zones for proactive monitoring of activities.

4.1.4 Monitor Suspicious Activity
Users can view alerts and flagged suspicious activities, such as the detection of visible weapons or anomalous behavior, within the surveillance system.

4.1.5 View Recorded Footage
Users can access stored video clips of suspicious events for further analysis and reporting.

4.1.6 Log Out
Users can securely log out of the system to prevent unauthorized access.

Operator Functions (Monitoring Personnel and Admin)

4.1.7 Manage User Accounts
Admins can create, update, or delete user accounts to manage system access for monitoring personnel and security staff.

4.1.8 Configure Surveillance Settings
Admins can adjust surveillance parameters, such as sensitivity levels for weapon detection or behavior analysis, and manage camera configurations.

4.1.9 Monitor Alerts on Dashboard
Monitoring personnel can receive real-time alerts on their dashboards when weapons or suspicious behaviors are detected by the system.

4.1.10 Confirm Threats and Escalate
Monitoring personnel can assess flagged events through live footage and escalate confirmed threats to security personnel if necessary.

4.1.11 System Performance Monitoring
Admins can monitor the system's health, uptime, and performance to ensure it operates optimally.

4.1.12 View and Manage Security Logs
Admins can access and review detailed logs of system activities, including alerts, responses, and user actions.

Support Functions

4.1.13 User Support
Provides assistance to users (monitoring personnel or security personnel) experiencing issues with account access or using the system features.

4.1.14 System Maintenance
The system will undergo regular updates and maintenance to enhance stability, security, and detection accuracy.

4.1.15 Data Backup and Recovery
Regular backup processes ensure that all logged events, video clips, and system data are safely stored and can be recovered in the event of data corruption or system failure.

Integration Functions

4.1.16 Integration with External Surveillance Devices
The system integrates with a variety of CCTV cameras installed in ATM zones to collect and analyze video feeds.

4.1.17 Cloud Storage for Suspicious Activities
All flagged events, along with their associated video clips, are securely stored in a cloud database for easy access and reporting.

4.1.18 API Access for Third-Party Monitoring Systems
External systems, such as law enforcement or centralized monitoring systems, can interact with the weapon detection system via secure APIs for additional analysis or alerting functionalities.

5. System Functions / Functional Requirements

Functional Requirements

Ref	# Functions	Category	Attribute	Details & Boundary Constraints
R1.1	Detect weapons or anomalies in ATM users	Evident	System Response Time	Anomalies must be flagged within 5 seconds.
R1.2	Generate and store security alerts	Evident	Notification Time	Alerts should be generated and sent to the dashboard within 2 seconds.
R1.3	Monitor and log suspicious activities	Evident	Data Storage Efficiency	Logs of flagged activities should be stored in a secure database for future review.
R1.4	Securely store recorded video clips	Hidden	Data Encryption	All stored data must be encrypted for privacy and security.
R1.5	Retrieve recorded video clips and logs	Evident	System Response Time	Logs and videos should be retrievable within 5 seconds.

Functional Goals Summary

- Detection Efficiency: Real-time identification of threats like weapons or suspicious behavior, with alerts generated within seconds, enables rapid responses and escalation.
- Secure Logging: Encrypted storage of suspicious activity and logs ensures data privacy and supports detailed post-incident analysis.
- Quick Data Access: Prompt retrieval of logs and video clips ensures efficient investigation and smooth operation during critical events.

System Attributes (Nonfunctional Requirements)

Attribute	Details and Boundary Constraints	Category
Response Time	Critical operations (detection, logging, alerting) must respond within 5 seconds.	Mandatory
Concurrent User Load	The system should support monitoring by at least 10 personnel simultaneously without performance degradation.	Mandatory
Security	All data (alerts, logs, video clips) must be encrypted during transmission and securely stored to comply with regulations.	Mandatory
Availability	The system must ensure 97.9% uptime for reliable ATM security monitoring.	Mandatory
Usability	The dashboard interface must be user-friendly, enabling smooth operation with minimal training.	Optional

5.1 Use Cases

5.1.1 List of Actors

1. **ATM User**
   - Interacts with the ATM. Their behavior is monitored by AI for anomalies and weapons.
2. **Monitoring Personnel**
   - Reviews flagged alerts, watches live video, and decides on escalation.
3. **Security Personnel**
   - Acts upon confirmed threats based on escalated alerts.

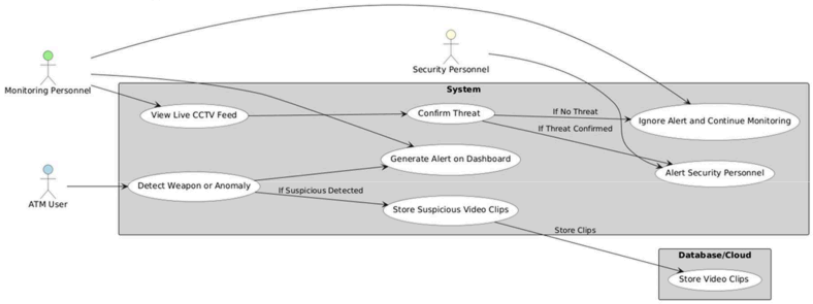
5.1.2 List of Use Cases

1. **Detect Suspicious Activity**
   - Continuously monitors ATM zones via CCTV for threats like weapons or strange behavior.
2. **Generate Alerts**
   - Triggers alerts upon detecting threats; sends alerts to monitoring dashboard.
3. **View Dashboard**
   - Monitoring personnel can view real-time alerts, live footage, and logs.
4. **Confirm Threat**
   - Determine if an alert indicates a real threat.
5. **Notify Security Personnel**
   - Sends an alert to the security team if the threat is confirmed.
6. **Log Activities**
   - Records all suspicious weapon and alerts .
7. **Access Logs**
   - Allows authorized users to retrieve stored logs and videos for investigation.

5.1.3 Use Case Diagram

Actors:
- ATM User
- Monitoring Personnel
- Security Personnel

Use Cases:
- Detect Suspicious Activity
- Generate Alerts
- View Dashboard
- Confirm Threat
- Notify Security Personnel
- Log Activities
- Access Logs
- Store Suspicious Video Clips



Description of Use Case: Detect Suspicious Activity
- **Use Case Name:** Detect Suspicious Activity
- **Actors:** ATM User, Monitoring Personnel, Security Personnel, AI-Based Detection System
- **Purpose:** Identify suspicious activities or threats (e.g., weapons, anomalous behavior) via AI-based detection through real-time CCTV monitoring.

Preconditions:
- AI detection system is active and integrated with ATM CCTV.

Postconditions:
- Monitoring personnel receive alert.
- Flagged data is saved in the database/cloud.
- Security personnel notified if necessary.

Normal Flow:
1. ATM user enters monitored area.
2. AI system detects threat or anomaly.
3. Alert is triggered and shown on monitoring dashboard.
4. Monitoring personnel reviews and confirms/dismisses threat.
5. If confirmed:
    - Security personnel alerted.

6. Non-Functional Requirements

6.1 Performance Requirements

- **Response Time:**
  All key operations, including user login, surveillance data retrieval, and suspicious activity detection, must be completed within a specified time limit (e.g., 5 seconds).
- **Data Processing Time:**
  Security analysis, such as suspicious activity detection, should be processed and results provided within a reasonable time frame (e.g., 10 seconds).

6.2 Safety Requirements

- **Data Protection:**
  All surveillance data and user information must be securely handled to avoid accidental loss or corruption.

6.3 Security Requirements

- **Authentication and Authorization:**
  The system must require secure login and authentication methods (e.g., two-factor authentication) to access user accounts and sensitive data.
- **Encryption:**
  All data, including user information and surveillance footage, must be encrypted during transmission and storage to prevent unauthorized access.
- **Compliance with Regulations:**
  The system must adhere to relevant security and data protection regulations, such as GDPR or other applicable regional laws.

6.4 Reliability Requirements

- **High Availability:**
  The system should ensure high availability, with a target uptime of 97.9%, ensuring minimal downtime.
- **Error Handling:**
  The system must be able to detect and recover from errors or failures without significant disruption to the surveillance services.
- **Data Redundancy:**
  Implement data redundancy mechanisms to ensure that critical system components (e.g., databases, servers) have backups and failover support.

6.5 Usability Requirements

- Ease of Use:
  The system interface must be user-friendly, intuitive, and accessible to both technical and non-technical users.
- **Training Materials:**
  Training resources must be provided to help users and admins understand and use the system effectively.
- **Error Messaging:**
  Clear and concise error messages must be provided to guide users in resolving common issues.

6.6 Supportability Requirements

- **System Monitoring:**
  Admins should have the ability to monitor system performance, user activities, and security issues through built-in tools and dashboards.
- **Maintenance:**
  The system should allow for easy updates, maintenance, and bug fixes without requiring system downtime. Regular maintenance procedures must be in place.
- **Customer Support:**
  A dedicated support team must be available for resolving user or system issues, with a ticketing system for tracking and resolving support requests.

6.7 User Documentation

- **User Manual:**
  A comprehensive user manual should be provided, detailing the system features, how to register, log in, and monitor surveillance data.
- **Admin Guide:**
  A separate admin guide must be available to assist admins in managing user accounts, configuring system settings, and generating reports.
- **FAQ and Troubleshooting:**
  A section containing frequently asked questions and common troubleshooting steps should be available to help users and admins resolve common issues independently.

APPENDIX C. DESIGN SPECIFICATIONS

1. Introduction

1.1 Purpose of Document

The purpose of this Software Design Specifications (SDS) document is to provide a detailed outline of the design considerations, architecture, and functionality of the Weapon Detection System for ATM Surveillance application developed using the Evolutionary Prototyping methodology.

This document serves as a guiding framework for the development team, detailing the iterative approach to refining and enhancing the application based on continuous feedback and testing.

The system leverages advanced deep learning algorithms to detect harmful weapons, such as pistols, in real-time from CCTV footage, addressing challenges like varying angles, occlusions, and false positives/negatives. Special emphasis is placed on tailoring the solution for ATM environments by optimizing for confined spaces, ensuring low- latency performance, and integrating automated alert mechanisms.

This document provides the necessary technical specifications and a roadmap to create a robust, secure, and efficient weapon detection system.

1.2 Intended Audience

This document is intended for:

- Bank Customers: To ensure their safety and security while conducting transactions in ATM facilities.
- Law Enforcement Agencies: To facilitate timely interventions in cases of weapon detection and enhance public safety measures.
- Regulatory Bodies: To assess the compliance and effectiveness of the weapon detection system in meeting security standards.
- Bank Workers and Security Personnel: To provide them with tools and actionable insights for maintaining a secure environment within ATM spaces.
- Surveillance System Operators: To assist in efficient monitoring and quick response to potential threats detected by the system.
- Security Solution Providers: To aid in the evaluation and integration of the weapon detection system into broader security frameworks.

1.3 Project Overview

The project aims to develop an advanced Weapon Detection System specifically designed for ATM surveillance environments, focusing on identifying harmful weapons, such as pistols, in real-time using CCTV footage.

The system is tailored to operate efficiently in confined spaces and handle challenges such as varying camera angles, partial occlusions, and similar-looking objects that may cause false alerts. By integrating automated detection with real-time alert mechanisms, the system enhances response times and reduces reliance on manual supervision.

Key Features:

- Secure integration with law enforcement and banking security systems
- Compliance with regulatory standards
- User-friendly monitoring tools
- Scalability
- Low-latency processing
- Streamlined and reliable performance

1.4 Scope

The system will include functionalities for:

1. Real-time Surveillance:
   AI-enabled cameras detect and record weapons, ensuring accurate monitoring with minimal false positives.
2. Cloud Analysis:
   Machine learning analyzes behavior patterns and flags suspicious activity for proactive intervention.
3. Centralized Database:
   Secure, encrypted cloud storage for managing recorded data and generating detailed reports.
4. Alerts:
   Instant notifications via SMS, email, or mobile apps to authorized personnel for quick response.
5. Privacy and Compliance:
   Safeguards anonymize non-essential data and ensure regulatory compliance for banking environments.
6. Integration and Scalability:
   Seamlessly integrates with bank systems, supports multiple ATMs, and ensures low-latency performance.

1.6 Not In Scope

The Weapon Detection and Facial Expression Analysis System is scoped to focus solely on ATMs and public areas where ATM machines are located. It will not extend to monitoring private locations or non-public areas not associated with ATMs.
Additionally:

- The system is designed to detect revealed simple weapons (e.g., knives, firearms), not concealed weapons.
- It does not replace manual security assessments or comprehensive audits.
  By recognizing these limitations, the project focuses on enhancing security specifically within the ATM domain.

2. System Level Architecture

2.1 System Decomposition

The system is divided into the following subsystems:

2.1.1 Suspicious Activity Detection Module

- Utilizes AI-based algorithms to analyze CCTV feeds for revealed weapons and anomalous facial expressions or behaviors.
- Generates alerts when suspicious activity is detected.

2.1.2 Surveillance Monitoring Unit

- Provides monitoring personnel with a real-time dashboard to view alerts and live CCTV feeds.
- Facilitates user interaction for escalating confirmed threats to security teams.

2.1.3 Notification System

- Handles the dissemination of alerts to appropriate parties, including monitoring personnel and security teams.
- Supports real-time notifications via dashboard updates and external messaging systems.

2.2 Element Relationships

- Subsystems communicate through defined interfaces to ensure seamless data flow:
  - The Detection Module sends alerts to the Monitoring Unit, including timestamps, video snapshots, and analysis details.
  - The Monitoring Unit processes user input to escalate verified threats, triggering the Notification System to alert security personnel.
  - Logged events are shared with the Data Storage Unit for future analysis and reporting.

2.3 External Interfaces

2.3.1 CCTV Cameras: Provide live video feeds for the Detection Module to analyze.

2.3.2 ATM Sensors: Supplement video feeds with contextual data (e.g., motion detection).

2.3.3 Remote Notification Services: Send alerts to mobile devices or external systems for security personnel.

2.3.4 Database: Stores event logs, user actions, and configuration data.

2.3.5 Execution Locations

- **Back-End Processing**:
  AI analysis and data processing are performed on dedicated servers to ensure high performance and reliability.
- **Cloud Environment**:
  Stores surveillance data, configuration files, and historical logs for scalability and accessibility.
- **User Devices**:
  Hosts the web-based dashboard for monitoring personnel.

2.3.6 Global Design Strategies

- **Error Handling Mechanisms**:
  - Retry logic ensures system resilience against transient network issues.
  - Fallback mechanisms address potential malfunctions in cameras or sensors.
- **Security Measures**:
  - Secure communication channels (e.g., HTTPS, secure APIs) protect data integrity and privacy.
  - Role-based access controls safeguard sensitive system functionalities.

2.4 Software Architecture

The software architecture for the Integraguard Surveillance System outlines the interaction between various layers of the system, ensuring a structured, scalable, and efficient design.

2.4.1 User Interface (UI) Layer

- **Role**:
  Serves as the primary interaction point for users, including monitoring personnel and administrators.
- **Functionality**:
  - Provides a web-based dashboard for real-time monitoring of ATM CCTV feeds.
  - Displays real-time alerts for detected weapons and suspicious behaviors.
  - Offers controls for triggering alarms and escalating alerts to security personnel.
  - Enables users to review historical logs and generate reports.

2.4.2 Application Layer (Middle Tier)

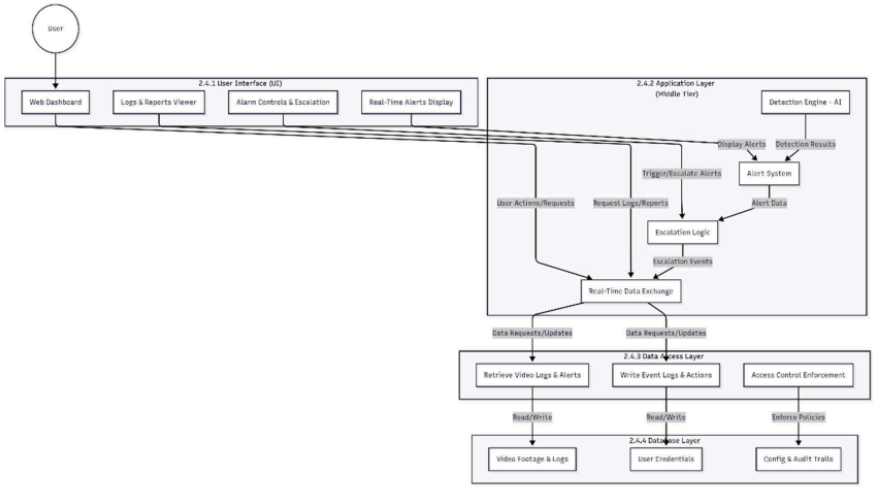
- **Role**:
  Processes user actions, applies business logic, and facilitates communication between layers.
- **Functionality**:
  - **Detection Engine**: Uses AI algorithms to analyze video feeds for weapons and anomalies.
  - **Alert System**: Triggers notifications for detected suspicious activity.
  - **Escalation Logic**: Allows confirmation and escalation of threats to security teams.
  - Manages real-time data exchange between the UI and Data Access Layer.

2.4.3 Data Access Layer (DAL)

- **Role**:
  Bridges the Application Layer and the Database, ensuring secure and efficient transactions.
- **Functionality**:
  - Retrieves video logs, stored alerts, and configuration data.
  - Writes logs of suspicious events, user actions, and escalation records.
  - Enforces access controls on sensitive surveillance data.

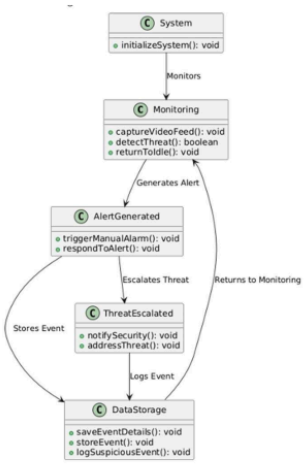
2.4.4 Database Layer

- **Role**:
  Provides persistent storage for all core system data.
- **Functionality**:
  - Stores video footage, timestamps, logs, and user credentials.
  - Maintains system configuration settings and audit trails.
  - Supports querying for report generation and historical analysis.



SYSTEM ARCHITECTURE DIAGRAM

3. Class Diagram



3.1 System

- Starts the weapon detection system using the initializeSystem() function.

3.2 Monitoring

- Continuously watches the environment by:
  - Capturing video: captureVideoFeed()
  - Detecting threats: detectThreat()
  - Returning to normal if no threat is found: returnToIdle()

3.3 Alert Generated

- When a threat is detected, an alert is created.
- Functions include:
  - Manually triggering an alarm: triggerManualAlarm()
  - Responding to the detected alert: respondToAlert()

3.4 Threat Escalated

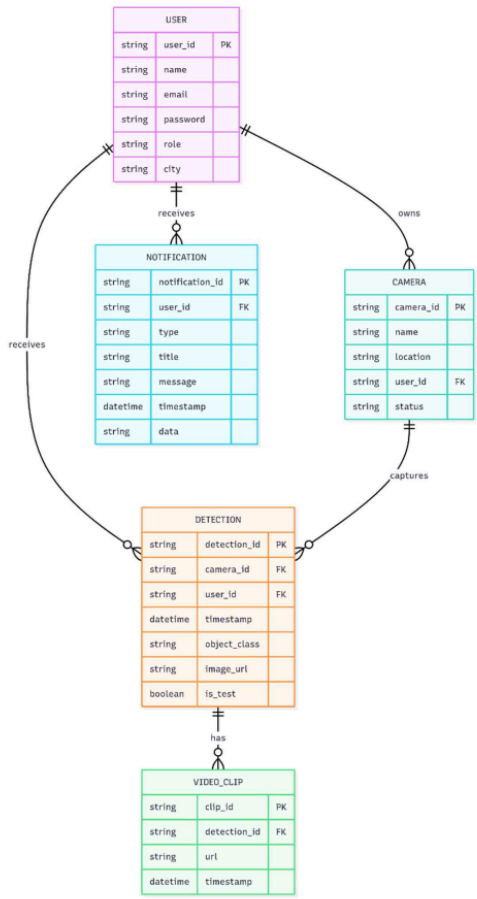
- If the threat is serious, it is escalated.
- The following actions are taken:
  - Notify security: notifySecurity()
  - Address the threat: addressThreat()

3.5 DataStorage

- Stores all event data for records and analysis.
- Core functions:
  - Save event details: saveEventDetails()
  - Log suspicious events: logSuspiciousEvent()
  - Store events: storeEvent()

4. Detailed System Design

ER Diagram



Explanation:

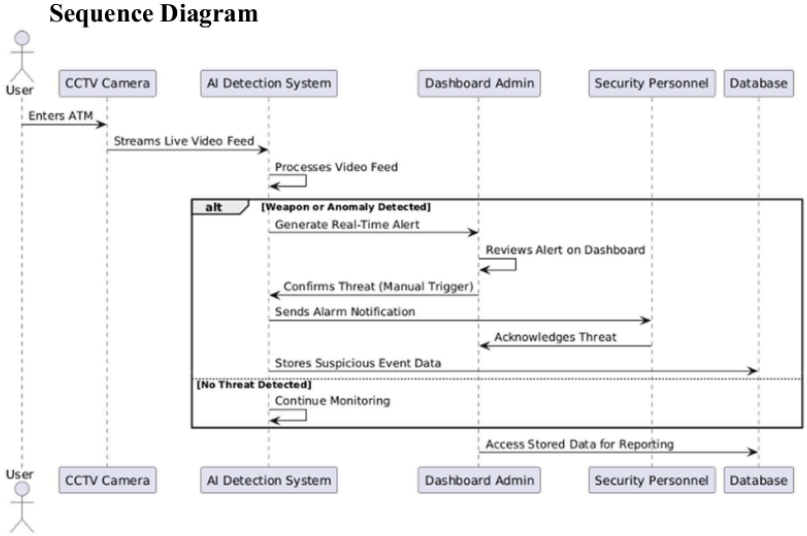
- **USER**: Represents system users (monitoring personnel, admins, etc.).
- **CAMERA**: Each camera is registered to a user.
- **DETECTION**: Each detection event is linked to a camera and a user.
- **NOTIFICATION**: Notifications are sent to users about detections or system events.
- **VIDEO_CLIP**: Video clips are associated with detection events.

Relationships:

- A user can own multiple cameras.
- A user can receive multiple detections and notifications.
- A camera can capture multiple detections.
- Each detection can have multiple associated video clips.

14. Application Design

Sequence Diagram



User:
- Represents an individual using the ATM.

CCTV Camera:
- Captures the live video feed and streams it to the AI system.

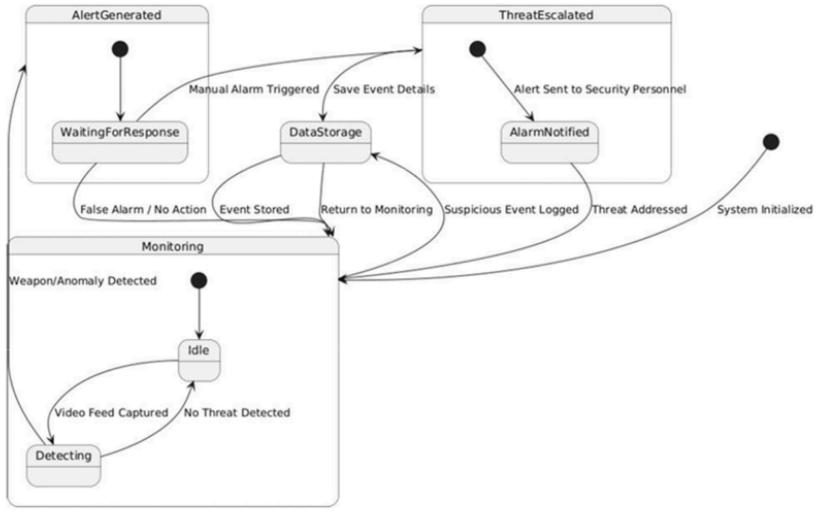
AI Detection System:
- Processes the video feed to detect weapons or analyze facial expressions/behaviors.
- Decides if an alert should be generated.

Dashboard Admin:
- Monitors alerts on the web interface dashboard.
- Confirms if an alert should escalate into a threat.

Security Personnel:
- Receives escalated alerts and responds accordingly.

Database:
- Stores event details for future reference and reporting.

State Diagram



APPENDIX D. OTHER TECHNICAL DETAIL DOCUMENTS

Test Cases Document

Test Case 1: User Registration

Field	Details
Test Case ID	TC-01
Title	User Registration
Description	Verify that a new user can register with valid credentials.
Preconditions	User is on the registration page.
Steps	1. Enter a valid email and password. 2. Click the "Register" button.
Expected Result	User account is created, and a confirmation email is sent.
Actual Result	SUCCESS

Table 5-1

Test Case 2: User Login

Field	Details
Test Case ID	TC-02
Title	User Login
Description	Verify that a registered user can log in with correct credentials.
Preconditions	User is registered and on the login page.
Steps	1. Enter valid email and password. 2. Click "Login".
Expected Result	User is logged in and redirected to the dashboard.
Actual Result	SUCCESS

Table 5-2

Test Case 3: Invalid Login

Field	Details
Test Case ID	TC-03
Title	Invalid Login
Description	Verify that login fails with incorrect credentials.
Preconditions	User is on the login page.
Steps	1. Enter valid email and incorrect password. 2. Click "Login".
Expected Result	Error message is displayed, login is denied.
Actual Result	ERROR

Table 5-3

Test Case 4: Add Camera

Field	**Details**
Test Case ID	TC-04
Title	Add Camera
Description	Verify that a user can add a new camera.
Preconditions	User is logged in and on the camera management page.
Steps	1. Enter camera details. 2. Click "Add Camera".
Expected Result	Camera appears in the user's camera list.
Actual Result	SUCCESS

Table 5-4

Test Case 5: Weapon Detection (Knife)

Field	**Details**
Test Case ID	TC-05
Title	Weapon Detection (Knife)
Description	Verify that the system detects a knife in the video feed.
Preconditions	User is logged in, camera is active, and dashboard is open.
Steps	1. Place a knife in the camera's field of view. 2. Observe dashboard.
Expected Result	Detection overlay appears, alert is triggered, notification is sent.
Actual Result	SUCCESS

Table 5-5

Test Case 6: Weapon Detection (Gun)

Test Case ID	TC-06
Title	Weapon Detection (Gun)
Description	Verify that the system detects a gun in the video feed.
Preconditions	User is logged in, camera is active, and dashboard is open.
Steps	1. Place a gun in the camera's field of view. 2. Observe dashboard.
Expected Result	Detection overlay appears, alert is triggered, notification is sent.
Actual Result	SUCCESS

Table 5-6

Test Case 7: False Positive Handling

Field	**Details**
Test Case ID	TC-07
Title	False Positive Handling
Description	Verify that harmless objects do not trigger weapon alerts.
Preconditions	User is logged in, camera is active.
Steps	1. Place a harmless object (e.g., pen) in the camera's view. 2. Observe dashboard.
Expected Result	No detection overlay or alert is triggered.
Actual Result	SUCCESS

Table 5-7

Test Case 8: Push Notification Delivery

Field	Details
Test Case ID	TC-08
Title	Push Notification Delivery
Description	Verify that notifications are sent to all registered devices upon detection.
Preconditions	User is logged in on multiple devices, camera is active.
Steps	1. Trigger a weapon detection event. 2. Observe all devices.
Expected Result	All devices receive a push notification.
Actual Result	SUCCESS

Table 5-8

Test Case 9: Detection Log Retrieval

Field	**Details**
Test Case ID	TC-09
Title	Detection Log Retrieval
Description	Verify that users can view detection history in the dashboard.
Preconditions	User is logged in, detection events have occurred.
Steps	1. Navigate to detection history page. 2. Review the log.
Expected Result	Accurate list of past detections is displayed.
Actual Result	SUCCESS

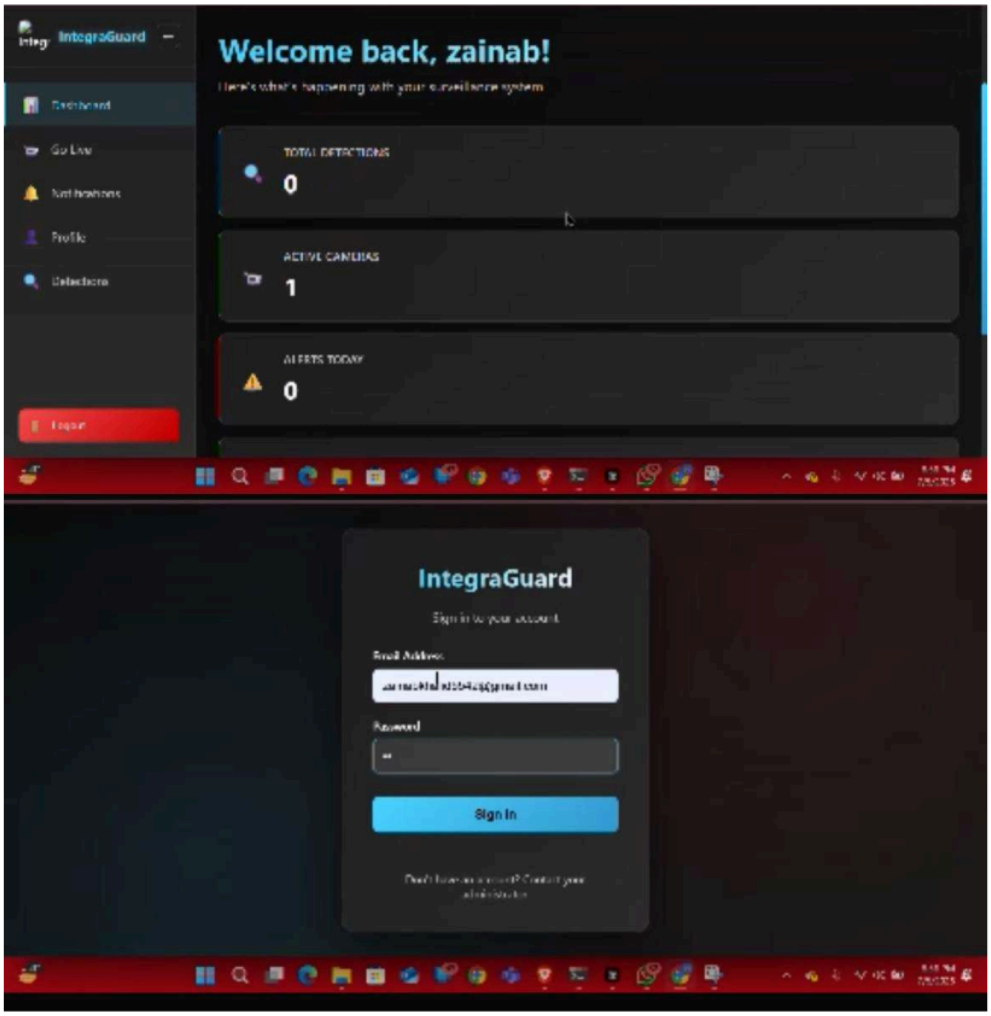
Table 5-9

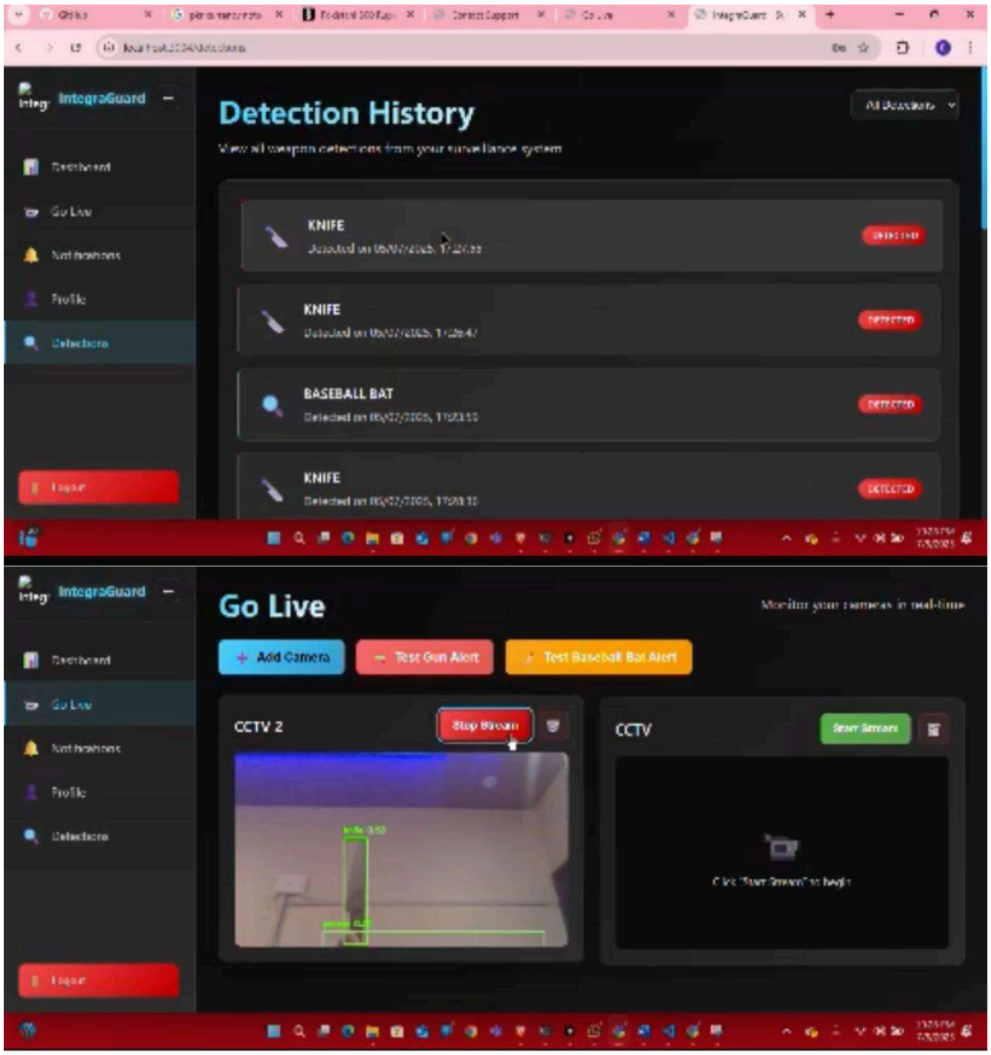
Test Case 10: Unauthorized Access Attempt

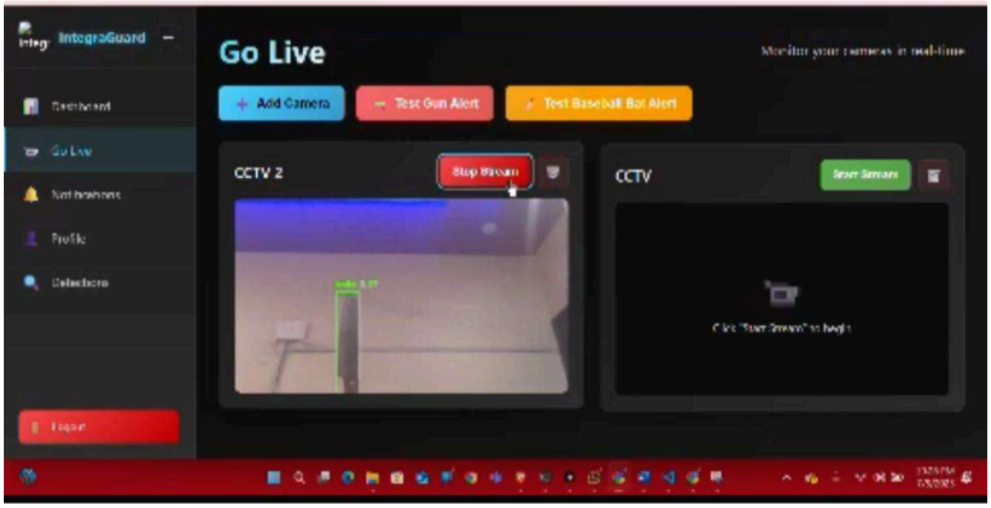
Field	Details
Test Case ID	TC-10
Title	Unauthorized Access Attempt
Description	Verify that unauthenticated users cannot access the dashboard.
Preconditions	User is not logged in.
Steps	1. Attempt to access dashboard URL directly.
Expected Result	Access is denied, user is redirected to login page.
Actual Result	SUCCESS

Table 5-10

UI/UX Detail Document:




IntegraGuard
Detection History
KNIFE
KNIFE
BASEBALL BAT
KNIFE
IntegraGuard
Go Live
Add Camera
Test Gun Alert
Test Baseball Bat Alert
CCTV 2
Stop Stream
CCTV



Coding Standards Document:

To maintain readability, consistency and long-term maintainability of the **INTEGRA GUARD SURVEILLANCE SYSTEM**, standard coding practices were strictly followed throughout the development process. These standards ensured that the source code was clean, well-organized and easily understandable by any developer or contributor.

User Manual Document:

1. Document of User Manual

The present user instructions are intended to be relevant to the ATM administrators and security personnel who use Intreguard Surveillance System, which is a web-based real-time weapon detection. A deep learning algorithm, YOLOv8, and a webcam allow detecting any weapons in the stream of video. In case of detection of a weapon, the system gives an on-screen message and an alert e-mail message is sent to the registered user.

2. Starting the Application

To begin:

- Launch a contemporary browser (e.g., Chrome, Edge).
- Contact the localhost installed in case of local deployment.
- Its home screen will look as follows with the title: "Intreguard Surveillance System."

3. Step by Step Instruction to Using the Application

Step 1: Logging in

- Go to the log in-screen and enter in your email and password.
- Click “Login”.
- After a successful sign in, you will be redirected to Dashboard.
- In the case a login fails there will be an error message.

Step 2: Go Live Surveillance (Webcam)

- Click on the start live feed option.
- The application will ask the permission to use camera- click Allow.
- After it is permitted, the webcam will turn on and the real-time video wave will be captured.
- Make sure that the camera view is well-lighted, steady, and clear to enable the right detection.

Step 3: Detection of Weapons

As long as the live feed is on:

- The YOLOv8 model does real time analysis on the video stream.
- In the event of detection of a weapon (e.g. firearm, knife):
- A screen reminder pops up on the Notification section.
- It sends out a detection email to the known user.
- Type of weapon detection is webbed in email.

Step 4: See the Detection Notifications

- Visit the tab of the “Notifications” menu.
- All detection entries have:
- Weapon Type (e.g. "Pistol", "Knife")

This enables the admins to promptly know the kind of threat detected and proceed on the subsequent measures.

Step 5:logout

Once completed, press the logout button and the session is logged out Securely.

A5. FLYER & POSTER DESIGN

TEAM MEMBERS
ZAINAB KHALID AND NIDA AAMIR

COPY OF EVALUATION COMMENTS BY JURY FOR PROJECT – I END SEMESTER EVALUATION

project idea is good but make things limited you can not cover all the stuff at once.
Satisfactory but need to study more related to their project
Scope to void
Presentation Skills weak

FYP-Rept-Intraguard Survellience System.docx

ORIGINALITY REPORT

11%	8%	3%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	3%
2	www.coursehero.com Internet Source	1%
3	Submitted to University of Paisley Student Paper	1%
4	Submitted to Danford College Student Paper	1%
5	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
6	Submitted to De Montfort University Student Paper	<1%
7	fastercapital.com Internet Source	<1%
8	Submitted to Middle East College Student Paper	<1%
	Submitted to University of Wolverhampton	

9	Student Paper	<1 %
10	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	<1 %
11	5dok.net Internet Source	<1 %
12	Submitted to University of Portsmouth Student Paper	<1 %
13	www.researchgate.net Internet Source	<1 %
14	odisha.gov.in Internet Source	<1 %
15	Submitted to Oxford Brookes University Student Paper	<1 %
16	Submitted to Independent University Bangladesh Student Paper	<1 %
17	Submitted to Nottingham Trent University Student Paper	<1 %
18	accessprofessionals.com Internet Source	<1 %
19	fr.slideshare.net Internet Source	<1 %

20	isteonline.in Internet Source	<1 %
21	eprint.iitd.ac.in Internet Source	<1 %
22	ijsrem.com Internet Source	<1 %
23	Submitted to Wakefield College Student Paper	<1 %
24	Adeyemi Abel Ajibesin, Doken Edgar. "chapter 19 Rifle Detection and Performance Evaluation Using Deep Learning Frameworks", IGI Global, 2023 Publication	<1 %
25	Submitted to University of East London Student Paper	<1 %
26	ijarcce.com Internet Source	<1 %
27	www.smartcitysystems.com Internet Source	<1 %
28	Submitted to ESoft Metro Campus, Sri Lanka Student Paper	<1 %
29	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1 %

30	Submitted to University of Sunderland Student Paper	<1 %
31	patents.google.com Internet Source	<1 %
32	www.tutorsonnet.com Internet Source	<1 %
33	Barbara Rita Barricelli, Daniela Fogli, Simone Gallo, Davide Guizzardi, Sara Maenza, Andrea Mattioli, Fabio Paternò. "Chapter 1 An EUD Approach to Creating Feasible and Energy-Saving Automations for Smart Homes", Springer Science and Business Media LLC, 2025 Publication	<1 %
34	Ajay Kumar, Sangeeta Rani, Krishna Dev Kumar, Manish Jain. "Handbook of AI in Engineering Applications - Tools, Techniques, and Algorithms", CRC Press, 2025 Publication	<1 %
35	arxiv.org Internet Source	<1 %
36	www.mdpi.com Internet Source	<1 %
37	Ajay Kumar, Deepak Dembla, Seema Tinker, Surbhi Bhatia Khan. "Handbook of Deep Learning Models for Healthcare Data	<1 %

Processing - Disease Prediction, Analysis, and Applications", CRC Press, 2025
Publication

38	Christoph Hartmann, Sophia Klauck. "Synthetic training data for crack propagation measurement by neural networks", Neural Computing and Applications, 2025 Publication	<1 %
39	Ram Pravesh, Bikash Chandra Sahana. "Robust Firearm Detection in Low-Light Surveillance Conditions Using YOLOv11 with Image Enhancement", International Journal of Safety and Security Engineering, 2025 Publication	<1 %
40	documents.mx Internet Source	<1 %
41	pdfcoffee.com Internet Source	<1 %
42	pure.tue.nl Internet Source	<1 %
43	www.ayoshya.com Internet Source	<1 %
44	www.resmilitaris.net Internet Source	<1 %
45	David Mumford, Jayant Shah. "Optimal approximations by piecewise smooth	<1 %

functions and associated variational problems", Communications on Pure and Applied Mathematics, 1989
Publication

46	"Sensor-Based Activity Recognition and Artificial Intelligence", Springer Science and Business Media LLC, 2025 Publication	<1 %
47	Rajeshwar Yadav, Raju Halder, Atul Thakur, Gourinath Banda. "SIRD-YOLO: an enhanced deep learning model for weapon detection using spatial interactions and diverse receptive fields", Innovations in Systems and Software Engineering, 2024 Publication	<1 %
48	Shalli Rani, Ayush Dogra, Ashu Taneja. "Smart Computing and Communication for Sustainable Convergence", CRC Press, 2025 Publication	<1 %
49	repository.effatuniversity.edu.sa Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		