APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 94-113
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Access Control for Al-Iman Workshop using Facial Recognition System

Salah Aldeen Taha Qasim Al Wrafi[1] , Nurul Azma Abdullah[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: azma@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.006

Article Info

Received: 30 July 2024
Accepted: 16 October 2024
Available online: 15 December 2024

Keywords

Facial recognition, Raspberry Pi, Flutter app, unauthorized access, security system, real-time monitoring, Deep Learning, Google Firebase, proactive security, non-operational hours

Abstract

Unauthorized Access Alert System addresses security vulnerabilities during non-operational hours. It integrats face recognition technology with Raspberry Pi and a Flutter application for real-time monitoring and immediate alerts. Traditional security measures, such as manual surveillance, often fail to prevent unauthorized access, particularly when tools and equipment are left outside, making establishments like the Al-Iman Center, a car repair shop, susceptible to intrusions and potential threats. To mitigate these limitations, our system employs algorithms in facial recognition, seamlessly integrated with Raspberry Pi for processing and detection. The user-friendly Flutter application is linked to Google Firebase, enabling real-time monitoring and rapid alerts. Upon detecting intruders, the system captures images and alerts authorities instantly.Our methodology includes using advanced facial recognition technology to identify unauthorized individuals, Raspberry Pi for local processing, and a responsive Flutter application for real-time monitoring and alerts. This approach significantly enhances security, reduces response time, and safeguards assets during non-operational hours. The system's significance lies in its proactive prevention of security breaches, emphasizing advanced technology for safety and protection. It is applicable in various settings, including car care shops, offices, industrial facilities, or any establishment vulnerable to unauthorized access during non-operational hours, thereby bolstering security protocols and improving real-time monitoring to minimize risks and safeguard valuable assets.

1. Introduction

Security is crucial in various areas like surveillance, industrial applications, offices, and homes. Effective security systems that monitor and respond to potential threats are essential. Many existing security systems, such as CCTV, infrared detectors, ultrasonic detectors, and PIR sensors [1], are used both indoors and outdoors. However, these traditional systems often fail to prevent unauthorized access, especially during non-operational hours. For example, at the Al-Iman Workshop, a car care shop, there is a high risk of intrusion when tools and equipment are left outside, making the shop vulnerable to theft and other security threats. This situation shows the need for a more advanced and reliable security solution that can provide continuous monitoring and quick responses to potential dangers.

At Al-Iman Center, security after hours relies on traditional surveillance, mainly manual monitoring prone to errors and delays. There's no automated staff recognition, leaving the shop vulnerable. They heavily rely on physical locks and lack a dedicated alert system. This project targets a substantial security upgrade, using cutting-edge technology for quicker detection and alerts, aiming to fill these gaps.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

The security measures at Al-Iman Center face multiple challenges. Manual monitoring is costly, prone to human error, and lacks real-time threat response. Relying on physical locks leads to issues like lost keys or unauthorized copies, without any immediate alert system for unauthorized access. This project aims to counter these vulnerabilities by proposing an advanced automated security solution to bolster safety during non-operational hours.

The proposed comprehensive security system for Al-Iman Center during non-operational hours relies on a Raspberry Pi setup with a webcam and speaker for real-time monitoring and alerts. Using facial recognition, it identifies staff and detects unauthorized access, triggering alerts, and notifying admins via a Flutter app. All data will be securely stored in Google Firebase. This solution aims to proactively enhance security, reducing reliance on manual surveillance and improving response time to threats.

2. Literature Review

Literature Review explores past research on unauthorized access detection systems, studying methodologies and technologies used in security. This review aims to grasp the evolution, challenges, and solutions in this field, shaping the methodology for facial recognition-based access control at Al Iman Workshop.

2.1 Al-Iman Workshop

Al-Iman Center in Sana'a, Yemen, positioned near Al-Iman University, grapples with security challenges. Traditional methods like locks and gate personnel fall short against modern threats. The Center's push for upgraded security aligns with a global trend favoring advanced systems. Evolving technology drives this shift, highlighting the Center's commitment to robust security protocols. Unauthorized access poses severe risks to the center, threatening losses and staff safety. Reviewing past security risks emphasizes the urgency to upgrade. The center seeks to evolve its protocols, adopting modern technology to mitigate vulnerabilities, protect assets, and ensure premises and personnel safety.

2.2 Facial recognition

Facial recognition involves identifying and verifying individuals by analyzing their facial features. This complex process is broken down into several key steps, each handled by specific machine learning algorithms, ensuring accuracy and reliability.

2.2.1 Face Detection

The first step in facial recognition is face detection, where the system identifies the presence of a face within an image. Histograms of Oriented Gradients (HOG) [2] is a commonly used method for this purpose. The process begins by converting the image to grayscale, which simplifies the analysis by focusing on intensity changes rather than color information. Next, gradients, or changes in brightness, are calculated for each pixel in the image. These gradients highlight edges and other important features within the image. The gradients are then combined into cells of 16x16 pixels, creating a simplified representation of the image that captures essential facial features, such as the nose, eyes, and mouth contours. This HOG representation is robust against variations in lighting and facial expressions, making it a reliable method for detecting faces in diverse conditions [3]. Figure 25 in Appendix A illustrates the HOG process used for face detection. Once a face is detected, the system needs to accurately map out the facial features, which leads us to the next step: face landmark estimation.

2.2.2 Face Landmark Estimation

In face landmark estimation, the system identifies specific facial points, known as landmarks, which include key features such as the chin, eyes, and eyebrows. Typically, 68 landmarks are used to precisely map out the face's geometry. These landmarks are crucial for normalizing the face, as they allow the system to apply affine transformations, such as rotation and scaling. These transformations ensure that the facial features are consistently aligned across different images, regardless of the face's orientation or size. This step is vital for maintaining geometric integrity, which is essential for accurate face recognition [4]. Figure 25 in Appendix A demonstrates how landmarks are estimated on a face. After aligning the facial features, the system needs to convert this aligned face into a numerical representation, which brings us to face encoding.

2.2.3 Face Encoding

Once the facial features are aligned, the system proceeds to face encoding, where it converts the aligned face into a numerical representation. Deep Convolutional Neural Networks (CNN) are employed for this task due to their ability to capture complex patterns and features from the input data. The CNN processes the face image to extract 128 distinct measurements, known as embeddings, which represent the deep facial characteristics. These embeddings encapsulate unique features of the face, enabling the system to differentiate between different individuals. The training process for CNN involves the use of triplet loss, a technique that helps the network learn to distinguish between similar and different faces effectively. Triplet loss works by minimizing the distance between embeddings of the same person while maximizing the distance between embeddings of different individuals [5]. Figure 26 in Appendix A shows the encoding process and the resulting embeddings for a face. With the face now encoded into a unique set of measurements, the system can proceed to the final step: face identification.

2.2.4 Face Identification

A linear Support Vector Machine (SVM) classifier compares the face embeddings with a database of known embeddings. The classifier identifies the closest match within milliseconds, enabling real-time applications [5]. Figure 27 in Appendix A illustrates how the algorithm identifies and recognizes faces. It demonstrates the process of face recognition, detailing the steps the algorithm takes to detect and verify individual faces.

2.3 Related Work

This section provides a detailed analysis of three existing facial recognition systems that have been developed and implemented in real-world scenarios. By examining these systems, we can better understand the current state of facial recognition technology, its applications, and its limitations.

2.3.1 Facial Recognition Attendance System Using Python and OpenCV

The facial recognition system for gated communities aims to enhance security by distinguishing between residents and visitors. It uses the YOLO algorithm for real-time facial detection and the DLIB library for classification. Live images from an entrance gate camera are processed to differentiate individuals. The system identifies residents through trained datasets, storing their data to track traffic. [6] It complies with privacy laws and follows the Waterfall methodology for development. Leveraging tools like Kera's, Visual Studio, Python, Face Recognition (DLIB), and OpenCV, this system seeks to heighten security within gated communities, focusing on authentication and identification. [12] The YOLO algorithm's accurate real-time facial recognition enhances its effectiveness.

2.3.2 Facial Recognition at Gated Community

The gated community facial recognition project aims to enhance security by distinguishing residents from visitors using the YOLO algorithm for real-time detection and DLIB for classification. Live gate camera images are processed to differentiate individuals, with the system storing resident data for traffic tracking. Complying with privacy laws, it follows the Waterfall methodology and employs tools like Kera's, Visual Studio, Python, Face Recognition (DLIB), and OpenCV. [7] This technology aims to boost gated community security through reliable, real-time facial recognition using the YOLO algorithm.

2.3.3 Development of a secured room access system based on face recognition using Raspberry Pi and Android-based smartphone

The secured room access system relies on facial recognition using a webcam and Raspberry Pi with OpenCV. It employs the Haar Cascade Classifier for face detection. When a user is detected, their identity is checked against the database; if authorized, a door lock opens. Unrecognized users trigger a notification to a master user via an Android smartphone for verification. [8] The master user can remotely grant access if needed. Components include a webcam, Raspberry Pi, a solenoid door lock, and an Android smartphone with Telegram for communication. Diagrams outline the user identification process and system components. Experimental results on the Haar cascade classifier method's detection accuracy is provided. [9]

2.4 Comparison with the Existing Systems

In this section, we examine a comparison between the Existing Systems and the Proposed System for the Al-Iman Center. Table 1 shows the comparison of the existing related system with the proposed system

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

Table 1 *Comparison with the Existing Systems and the Proposed System*

Systems / Aspect	Facial Recognition Attendance System	Facial Recognition at Gated Community	Secured Room Access System	Proposed System for Al-Iman Center
Application Setting	Educational Institutions	Gated Community	Secured Rooms	Al-Iman Center during non-operational hours
Primary Objective	Efficient attendance monitoring	Enhanced security for residents	Secured access control	Unauthorized access detection & alerting
Language & Algorithm	Python, OpenCV	Python, YOLO, DLIB	Python, OpenCV, Facial recognition	Python, Facial recognition, Flutter
Hardware	Laptop	Raspberry Pi	Raspberry Pi	Raspberry Pi
Data Storage & Handling	Excel sheets Local	logs, Raspberry Pi storage	Database, Smartphone	Google Firebase
Scope of Recognition	Students, staff (up to 2000)	Residents and visitors	Room entrants	Access control
Accessibility & Portability	Portable system, email distribution	Raspberry Pi, Wi-Fi connectivity	Android smartphone, Telegram	Portable, Real-time alerts
User Interface	Flask User-friendly interface	Not specified	Not specified	Emphasis on user-friendly interface for admin

The proposed system for Al-Iman Center offers several significant advantages over existing facial recognition systems as in Table 1 above, particularly those used in gated communities or secured rooms. Unlike traditional systems that may only log access attempts or rely on manual monitoring, the Al-Iman system provides real-time facial recognition and streaming video capabilities directly to the admin's smartphone app. This allows administrators to monitor live video feeds, insert or delete recognized individuals remotely, and receive immediate alerts of unauthorized access attempts. Additionally, the system includes a screen monitor inside the shop for real-time viewing of detected unauthorized persons, enhancing situational awareness. The system is designed to operate 24/7, automatically activating unauthorized person detection and video streaming when the shop is closed. The automatic pop-up of the camera feed during non-operational hours allows for prompt response to potential threats. By integrating these advanced features, the Al-Iman system surpasses the limitations of traditional facial recognition systems, providing comprehensive and automated security coverage that significantly enhances the safety and monitoring capabilities of the shop.

3. Methodology

This chapter is all about giving you a big picture of how the Al-Iman Center project is set up and designed. It talks about the hardware and cloud systems being used and how they work together. It also explains the main idea behind the whole setup. To help you understand better, it uses block diagrams to show how everything fits together.

3.1 Agile Model

The Unauthorized Access Detection and Alert System for Al-Iman Center adopts an agile approach in its development, emphasizing rapid prototyping and detailed requirement gathering. This methodology helps identify critical concerns early, reducing risks throughout the development cycle. The Agile Development Life Cycle, shown in Figure 1, demonstrates the iterative nature of this approach.

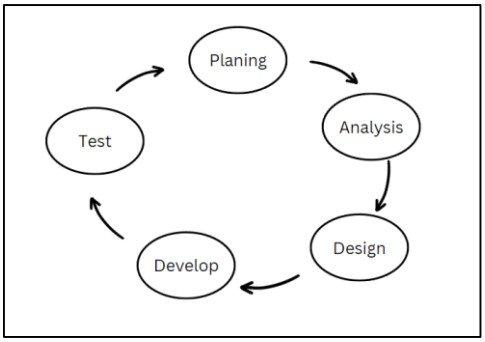

Fig. 1 *Agile Development Life Cycle [15]*

Figure 1 illustrates the iterative phases of the Agile Development Life Cycle: planning, analysis, design, development, and testing. Each phase is revisited as necessary to refine and enhance the system based on ongoing feedback and evolving requirements, ensuring a robust and adaptive security solution.

3.1.1 Planning Phase

The planning phase in developing the security system for Al-Iman Center is foundational, focusing on defining scope, goals, and strategies. Stakeholder engagement aids in understanding security needs, guiding functionality delineation. A detailed roadmap structures the development process, considering resource allocation and technology selection like Raspberry Pi, facial recognition, and Google Firebase. Defining measurable metrics ensures effective evaluation. This meticulous planning forms the core of the prototype methodology, ensuring an informed and purpose-driven approach for a robust security system.

3.1.2 Analysis Phase

During the analysis phase for the automated unauthorized access detection system at Al-Iman Center, extensive online meetings engaged stakeholders and conducted in-depth interviews. This approach revealed critical security vulnerabilities during non-operational hours, aligning closely with user needs. Functional requirements, including unauthorized individual detection and staff recognition, were identified alongside technical necessities like hardware and secure data storage. Integrating user requirements with rigorous analysis established a strong foundation for informed decision-making in design and implementation phases.

3.1.3 Design Phase

During the Design Phase for Al-Iman Center's security system, we're crafting a comprehensive blueprint integrating Raspberry Pi, Facial recognition, Google Firebase, and the admin app. We're ensuring seamless communication among system components and focusing on a user-friendly admin app design. Security measures like access controls and encryption are pivotal to safeguarding data. Our detailed plan includes testing strategies to ensure the system's functionality and security align with intended goals. This phase guides us in building a system that meets requirements while prioritizing security.

3.1.4 Development Phase

During the Implementation Phase, the security system blueprint is actualized. Software components, including facial recognition algorithms using Yolo and the admin app via Flutter, are developed. Simultaneously, hardware components like the Raspberry Pi, webcam, and speaker are set up and configured as per design specs. Seamless connectivity and functionality integration are emphasized, forming a strong system foundation. Additionally, the Google Firebase database is established for secure storage of captured images and system logs.

3.1.5 Testing Phase

In the Testing Phase, the system undergoes thorough evaluations to ensure it aligns with its design. Components are individually and interactively tested for functionality and security. User involvement ensures the system meets requirements, and stress tests assess reliability. Issues identified are opportunities for improvement, ensuring exceptional performance, security, and usability. Testing sessions are conducted via Google Meet for real-time collaboration and feedback, ensuring thorough validation and optimization.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

4. System Analysis and Design

This section details the analysis and design outcomes for the upcoming application. It covers functional and non-functional requirements, hardware and software needs, software specifications, and architectural design using an object-oriented approach. Visual representations are created using the Unified Modeling Language (UML).

4.1 System Requirements Analysis

System Requirement Analysis involves thoroughly examining acquired system data to understand its environment and improve effectiveness through computer-based solutions. This process determines the necessities for the system under development, using visual representations like Use Case, Sequence, Activity, and Class Diagrams.

4.1.1 Functional Requirements

Functional requirements outline the specific functionalities and capabilities that the system must possess to meet the user's operational needs. Table 2 lists the functional requirements.

Table 2 *Functional Requirements*

No	Module	Functions
1	Facial Recognition	Identify authorized staff members through facial recognition.
2	Unauthorized Access	Detect unauthorized individuals entering the premises after closing hours.
3	Alert System	Trigger alerts and capture images of unauthorized access attempts.
4	Notification System	Send notifications to the admin application upon unauthorized access detection
5	Admin Application	Allow admin to log in securely for system access.
6	Image Viewing	Provide the admin with the capability to view captured images of unauthorized individuals.
7	System Integration	Ensure seamless interaction between Raspberry Pi, Firebase, and the Admin Application.
8	Data Storage	Store images and logs securely in the Firebase database.
9	Register and Login Module	The system should allow the admin to login to the app in the first time. The system should allow the admin to change login credentials. The system should allow the admin to create a new administrator user. The system should display an error message when empty field is found. The system should allow the user to log into the system using the user_id and password that has been registered. The system should redirect to the desired dashboard based on the user role once the user successfully logged in. The system should show an error when user_id and password are wrong.

Table 2 defines what the system should do in terms of its operations, features, and interactions. They serve as the backbone for system development, detailing the core functionalities that users expect.

4.1.2 Non-Functional Requirements

Non-functional requirements define how the system should perform beyond its basic functions, covering qualities like performance, security, scalability, usability, and reliability, unlike functional requirements that detail system actions, these focus on the system's performance standards. As shown in Table 3 Non-Functional Requirements.

Table 3 *Non-Functional Requirements*

No	Requirements	Description
1.	Reliability	The system is designed to maintain continuous operation without crashing or experiencing unresponsiveness, except in cases of operating system errors.
2.	Availability	Throughout the year, the system will remain accessible, except for brief and pre-scheduled maintenance periods.
3.	Security and Privacy	Access to the system is always restricted to authenticated users.
4.	Usability	The user interfaces have been designed for ease of use, ensuring a straightforward interaction regardless of the user's technical expertise or background.

These non-functional requirements in Table 3 ensure that the Al-Iman Center's security system operates efficiently and effectively, providing reliable, accessible, and secure monitoring with user-friendly interfaces.

4.1.3 Integration of Components in the System

In this section, we'll explore the integrated components of the "Access Control Facial Recognition for Al-Iman Center on Raspberry Pi" system. The system includes several key components: Raspberry Pi 4, Raspberry Pi cameras, and a speaker. The figures in Appendix A illustrate the system's framework (Figure 27) and a flowchart detailing the facial recognition and access control processes (Figure 28). These visual aids offer a clear understanding of how the Raspberry Pi 4 interfaces with the cameras to capture and process images, and how the speaker is used for alert notifications. They demonstrate the connections and operational sequences within the Al-Iman Center framework, ensuring a cohesive and efficient security solution.

4.1.4 User Requirements Analysis

User requirement analysis identifies essential system functionalities for efficient task completion. Clients document these needs early in the validation process before device development. Table 4.3 details the specified system requirements by users.

Table 4 *User requirements*

No.	User Requirements (Administrative & User)
1.	The system will allow the administrator to log in and log out.
2.	The administrator will be able to add and delete staff.
3.	The administrator will be able to view and edit resident accounts.
4.	The administrator will be able to view the stored result of detected people
5.	The administrator will be able to save the stored result of detected people
6.	The administrator will be able to change his info.
7.	The administrator should be able to view all the cameras.
8.	The staff should be able to view all the cameras inside the shop

These user requirements in Table 4 ensure that the system provides comprehensive functionalities for administrators and staff, enhancing the overall efficiency and effectiveness of security management at the Al-Iman Center.

4.2 Analysis

This phase will contain the analysis progress for the system and who will use the developed system, in addition to the process and functions available to the user. The analysis phase shows what the system contains and performs.

4.2.1 System Architecture

Figure 2 shows the architecture of the system involves a combination of frontend, backend, and database components, which work together to deliver the system functionality. The front end communicates with the backend through APIs (application programming interfaces). The front end refers to the user interface of the system and the back end refers to the server-side components of the system, which handle tasks such as data storage, and processing by the host PostgreSQL server.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

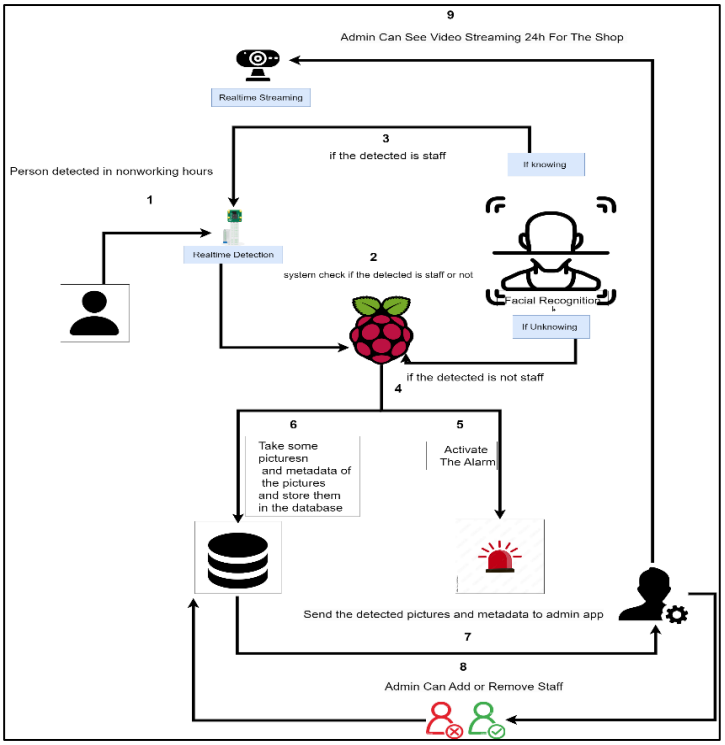

Fig.2 *System Architecture for Al-Iman Workshop*

4.2.2 Data Context Diagram (DFD CD)

A data Context Diagram illustrates the scope and boundaries of a system by depicting its interactions with external entities and the flow of data between them. Figure 3 will show the Data Context Diagram for the Access Control for the Al-Iman Workshop using the Facial Recognition System.

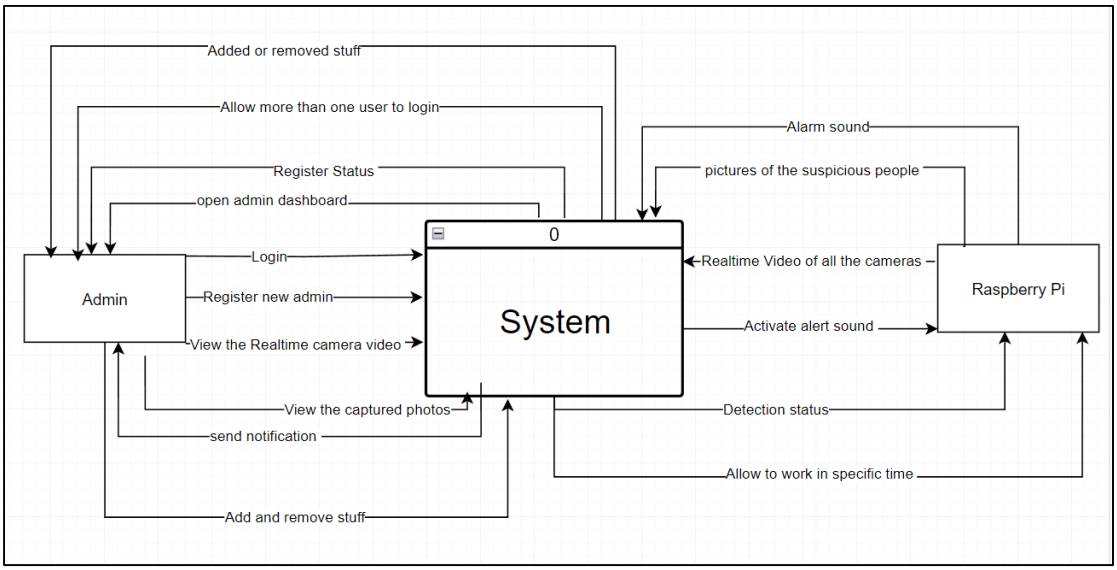

Fig. 3 *Data Context Diagram for Al Iman Workshop*

The diagram outlines how the Admin, System, and Raspberry Pi communicate and exchange data. The admin can log in, register new admins, and access the admin dashboard to manage staff and view real-time camera video. The System processes these inputs, allowing the Raspberry Pi to activate the alert sound, capture images of suspicious individuals, and manage detection statuses. The Raspberry Pi feeds real-time video and detection statuses back to the System, which then informs the Admin. This seamless interaction ensures efficient monitoring and immediate response to unauthorized access attempts.

4.2.3 Data Flow Diagram

A Level 0 Data Flow Diagram provides a high-level overview of a system's functionalities, showcasing the main processes and their interactions with external entities through data flow. Figure 4 will show the Data Flow Diagram for the system.

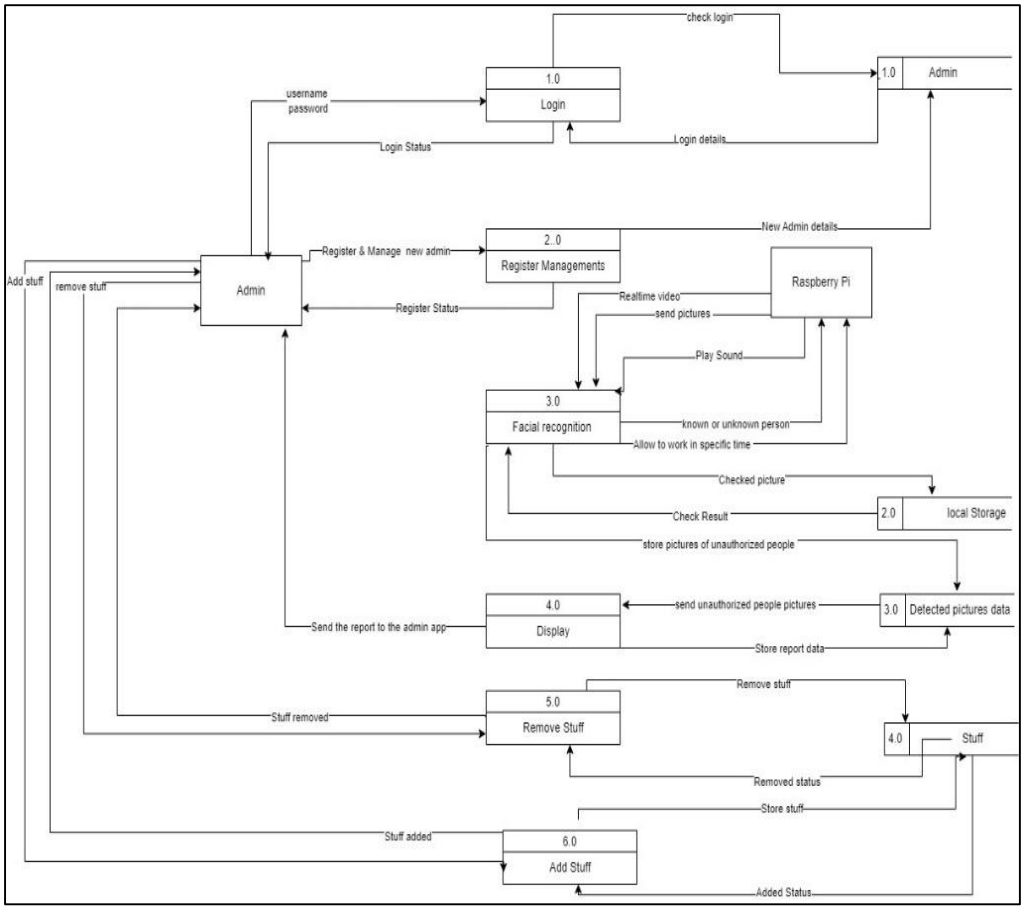

Fig. 4 *Data Flow Diagram Level 0 for Al-Iman Workshop*

The Data Flow Diagram (DFD) in Figure 4 illustrates how data moves through the security system for the Al-Iman workshop. It shows the processes of login, register management, facial recognition, display, and staff management. Admins log in and manage new admins, with the Raspberry Pi handling real-time video and facial recognition. Captured images and alerts are processed and displayed to the admin, allowing for real-time monitoring and response. This DFD provides a clear overview of the system's data interactions, ensuring efficient and effective security operations.

4.2.4 Entity Relation Diagram (ERD)

An entity relationship diagram is designed to outline the relationship between entities in Al Iman Workshop. Figure 5 shows the connection between entities in Al Iman Workshop System.

The Entity Relation Diagram (ERD) in Figure 5 shows the relationships between key entities in the Al-Iman Workshop security system. It includes Admins, who manage the system and staff, and Manage_new_staff, which stores new staff details. Detected_Photo captures images of unauthorized access, while Metadata provides additional information about these photos. View_Report links admins to the reports they can view. This diagram highlights the structured organization of the system's data and the interactions between these entities, ensuring efficient security management.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

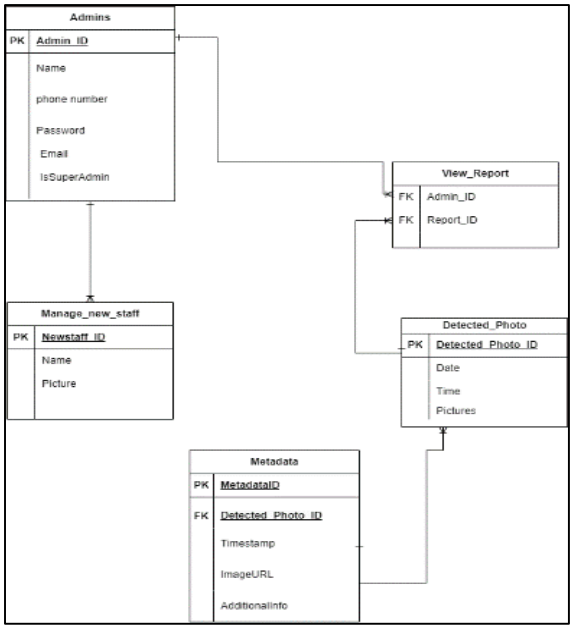

Fig. 5 *Entity Relation Diagram for Al-Iman Workshop*

4.3 Implementation of Security Module

The "Unauthorized Access Alert System" for Al-Iman Centre used a Raspberry Pi 4, a USB webcam, and a speaker. Python and OpenCV handled facial recognition, identifying authorized personnel, and detecting unauthorized access. A Flutter admin app enabled real-time monitoring, with Google Firebase securely storing data, ensuring an efficient and reliable security solution.

4.3.1 System Hardware

The "Unauthorized Access Alert System" uses a Raspberry Pi 4 with Raspbian OS, a USB webcam for image capture, and a USB speaker for alerts, ensuring robust monitoring and seamless software integration.

4.3.1.1 Database Connection

This section and the following Figure 6 demonstrate how to connect to a Firebase database using Firebase Admin SDK and Pyrebase.

```
Initialize Firebase Configuration:
    Set apiKey
    Set authDomain
    Set databaseURL
    Set projectId
    Set storageBucket
    Set serviceAccount path

Initialize Firebase Admin SDK:
    Load serviceAccount credentials
    Initialize Firebase Admin app with:
        databaseURL
        storageBucket

Initialize Firebase using Pyrebase:
    Initialize Pyrebase with configuration
    Get reference to Firebase storage
    Get reference to Firebase database
```

Fig. 6 *Database Connection (Firebase)*

The database connection used Google Firebase, with the Admin SDK and Pyrebase initialized for secure storage. Images and metadata of unauthorized access attempts were uploaded to Firebase Storage and Realtime Database, ensuring efficient, encrypted data handling and seamless integration with the system.

4.3.1.2 Download All Staff Image

The system includes a function to download all staff images from Firebase Storage. Figure 7 outlines the steps to download staff images from Firebase Storage.

```
Define download_images():
    List files in "faces" folder in Firebase Storage
    For each file in the list:
        Get file name
        Set local file path
        Try:
            Download file to local path
            Print success message
        Except Exception:
            Print error message
```

Fig. 7 *Download All Staff Images Function to Do Facial Recognition*

This function in figure 7 lists all files in the "faces" folder within Firebase Storage and downloads each image to a specified local directory. This ensures that the latest images are available for facial recognition, keeping the system updated with current staff photos.

4.3.1.3 Create Encoding File for the Staff

The system creates an encoding file for all staff images to enable efficient facial recognition. The following figure demonstrates the process of encoding face images and saving these encodings to a file for future use.

```
Define function findEncodings(imagesList):
    Initialize empty encodeList
    For each image in imagesList:
        Convert image color from BGR to RGB
        Get face encoding from image
        Append encoding to encodeList
    Return encodeList

Print "Encoding Started ..."
Call findEncodings with imgList and store result in encodeListKnown
Combine encodeListKnown with staffIds into encodeListKnownWithIds
Print "Encoding Complete"

Open file "EncodeFile.p" in write mode
Save encodeListKnownWithIds to file using pickle
Close file

Print "Loading Encode File ..."
Open file "EncodeFile.p" in read mode
Load encodeListKnownWithIds from file using pickle
Close file

Print staffIds
Print "Encode File Loaded"
```

Fig. 8 *Creating Encoding File for All Staff Faces*

This process in Figure 8 involves reading the downloaded images, converting them into a suitable format, and generating face encodings using the `face recognition` library. These encodings, along with the corresponding staff IDs, are saved into a pickle file. This encoded data is crucial for quickly and accurately identifying staff members, ensuring the system can recognize authorized personnel in real time.

4.3.1.4 Nonworking hours (12:00 am to 8:00 am)

The system operates from 12:00 AM to 8:00 AM to detect unauthorized access during nonworking hours. Figure 9 demonstrates the process that illustrates the code for time-based frame generation.

```
Function within_time_range():
    Get current date and time (now)
    Define start time as 12:00 am
    Define end time as 8:00 am
    If current time is between start time and end time:
        Return True
    Else:
        Return False
```

Fig. 9 *pseudocode for Time-Based Frame Generation*

The is_time_between function checks if the current time is within this range, activating the gen_frames function for video capture. This setup conserves resources by only monitoring during high-risk times. The facial recognition system automatically activates during nonworking hours (12:00 AM to 8:00 AM) to detect unauthorized access.

4.3.1.5 Facial Recognition

The system uses facial recognition to identify staff and detect unknown individuals, triggering alerts, capturing images, and logging metadata for any unauthorized access attempts.

4.3.1.5.1 Known (Staff) and Unknown

The system identifies and processes known staff members using facial recognition technology. When a face is detected, it is compared with the stored encodings of authorized personnel. If a match is found, the system labels the face with the corresponding staff member's name and draws a green rectangle around the face on the video feed, as demonstrated in Figure 10 and Figure 11. This allows for quick and accurate identification of authorized individuals, ensuring that staff members are recognized and can be distinguished from unauthorized persons.

```
Function recognize_faces():
    For each detected face in the frame:
        Extract facial features (face embedding)
        Compare face embedding with known encodings
        If match found:
            Continue to next face
        Else:
            Label face as "Unknown"
            Draw red rectangle around face
            Set unknown_detected flag to True
    Display annotated frame
```

Fig. 10 *The pseudocode for The Recognized Face As known or unknown*

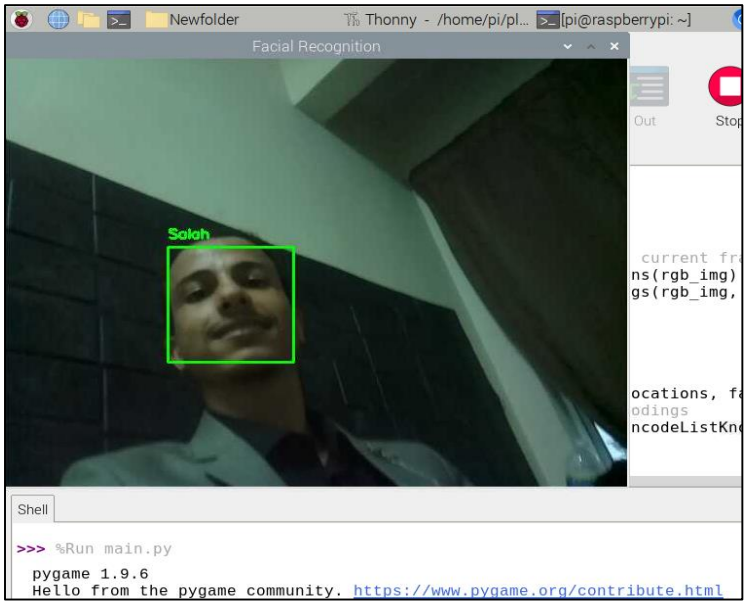

Fig. 11 *Camera Recognized Staff Face*

When the system encounters a face that does not match any of the stored encodings, it identifies the individual as unknown. Upon detecting an unauthorized person, the system initiates a series of actions to handle the unauthorized access attempt. This process includes labeling the face as "Unknown," drawing a red rectangle around the face on the video feed and performing additional steps such as playing an alarm, capturing images, and uploading them along with metadata to Firebase for further review figure 12 provides a detailed illustration of the discussed concept.

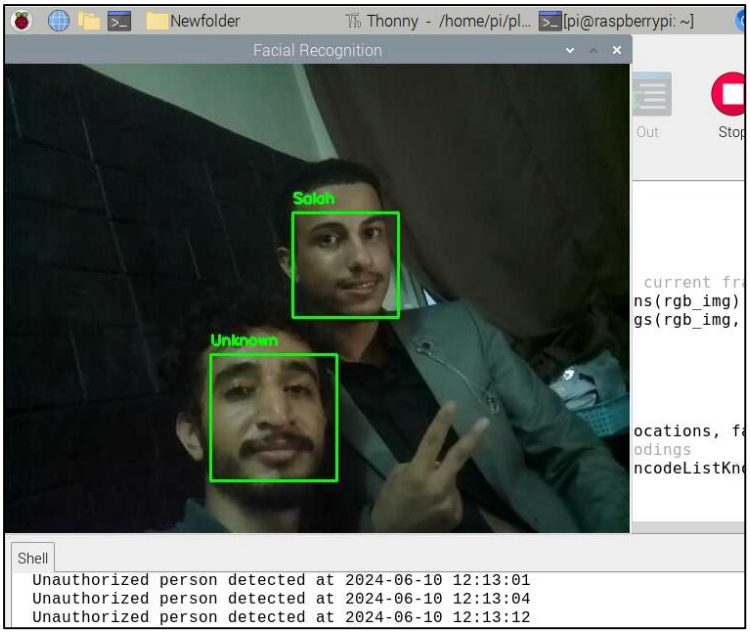

Fig. 12 *Camera Detected Unauthorized Person*

4.3.1.6 Take Pictures

The system captures multiple images of the unknown individual. These images are crucial for identifying and reviewing unauthorized access attempts later. Figure 13 provides a detailed illustration of the discussed concept

```
Function handle_unknown_detection():
    Get current date and time (current_time)
    Print "Unauthorized person detected at {current_time}"

    If unknown face detected and capture count < 3:
        Capture image
        Encode image as JPEG
        Create unique file name (img_name)
        Upload image to Firebase Storage
        Log event in Firebase Database with:
            timestamp
            additional_info
            image_url
        Increment capture count
```

Fig. 13 *Pseudocode for Capturing and Uploading Images of Unknown Faces to Firebase*

When an unknown face is detected, the system captures images and logs the event for security purposes. This process involves capturing the image, encoding it, uploading it to Firebase Storage, and logging the event details with a timestamp.

4.3.2 App Interface

The application interface is designed to provide a user-friendly experience for managing security at the Al-Iman Center. The following sections outline the key interfaces of the app:

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

4.3.2.1 Login Page

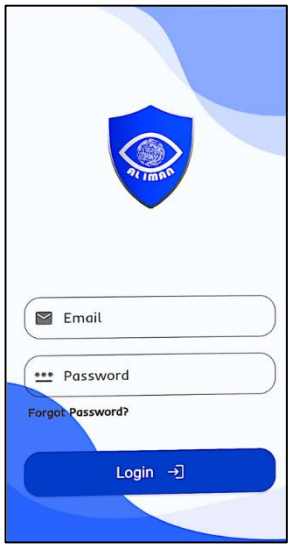

Fig. 14 Login In

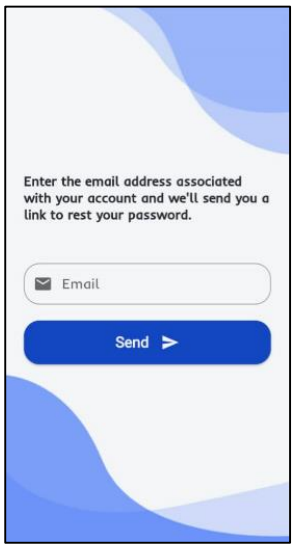

Fig. 15 Forget Password

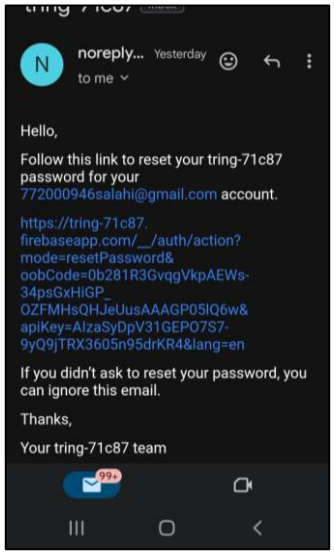

Fig. 16 Receive Email to Reset Password

Fig. 17 Create New Password

4.3.2.2 Home Page

The Al-Iman Center app's home page manages security features, including photo detection, staff management, and real-time video tracking. The "Detect People Photos" section uploads only images with faces. "Add Staff" and "Remove Staff" options help manage the staff database. Real-time video streaming allows live monitoring for comprehensive security

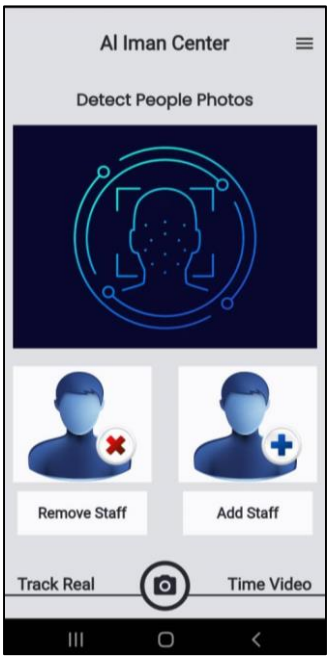

Fig. 18 Home Page Interface

Fig. 19 Home Page Interface

4.3.2.3 Detected People's Photos

The "Detected People's Photos" section displays events of unauthorized access attempts, each labeled with a unique event ID. Each event includes images, alerts, dates, and times of detection, providing a detailed log for administrators to review. This ensures that all unauthorized access attempts are documented and easily

accessible for security analysis and follow-up actions. Only images with detected faces are uploaded to the database to maintain the accuracy and relevance of the stored data. Figure 20 provides a detailed illustration of the detected people page.

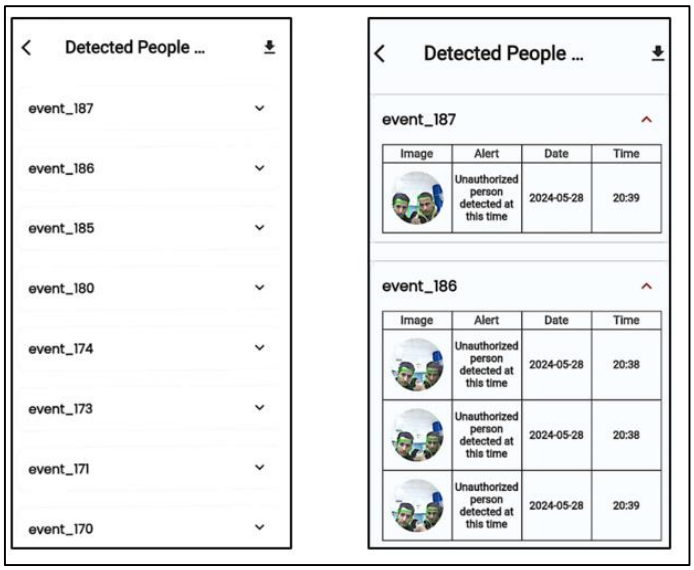

Fig. 20 Detected face & Add New Staff

4.3.2.4 Add New Staff

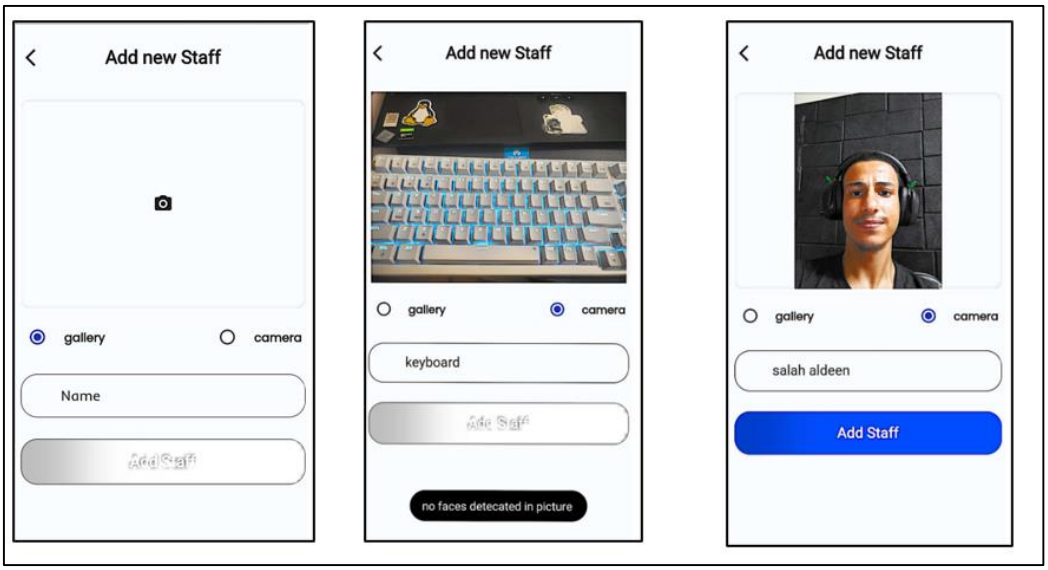

Fig. 21 Detected face & Add New Staff

Figure 21 The "Add Staff" interface allows administrators to add new staff members by uploading their photos either from the gallery or directly using the camera. The system ensures that only images with detected faces are accepted; if no face is detected in the uploaded photo, the image will not be uploaded to the database, ensuring data integrity and accuracy. Administrators also input the staff member's name, providing a complete and accurate record for the facial recognition system.

4.3.2.5 Remove Staff

The "Remove Staff" interface allows administrators to remove staff members from the system. This feature displays a list of current staff members, each with a delete icon next to their name and photo. Administrators can easily select and remove staff members, ensuring that the database remains up-to-date and accurately reflects active personnel. This functionality is crucial for maintaining the integrity and security of the facial recognition system. The following figure illustrates the page interface clearly.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

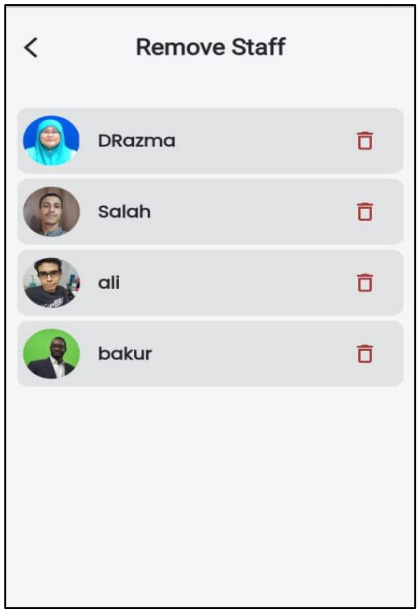

Fig. 22 Detected face & Add New Staff

4.3.2.6 Streaming Real-Time Video

The "Streaming Real-Time Video" interface allows administrators to monitor live video feeds from the security cameras. The following Figure snippet demonstrates this process:

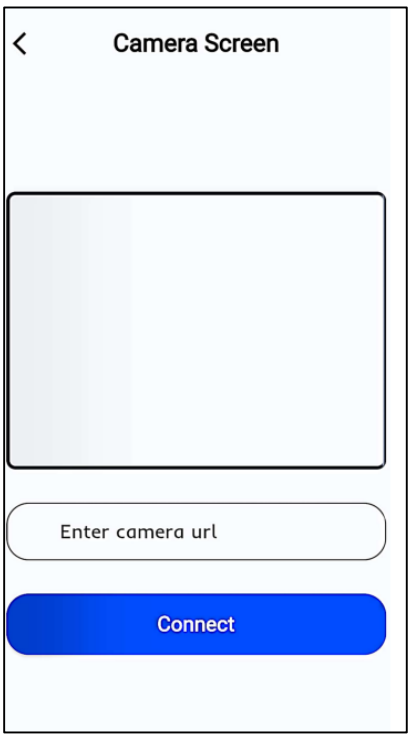

Fig. 23 Home Page Interface

```python
from flask import Flask, Response, render_template
import cv2

app = Flask(__name__)
camera = cv2.VideoCapture(0)


def gen_frames():
    # use 0 for web camera
    while True:
        success, frame = camera.read()  # read the camera frame
        if not success:
            break
        else:
            ret, buffer = cv2.imencode(".jpg", frame)
            frame = buffer.tobytes()
            yield (
                b"--frame\r\n" b"Content-Type: image/jpeg\r\n\r\n" + frame + b"\r\n"
            )  # concat frame one by one and show result


@app.route("/video_feed")
def video_feed():
    # Video streaming route. Put this in the src attribute of an img tag
    return Response(gen_frames(), mimetype="multipart/x-mixed-replace; boundary=frame")


@app.route("/")
def index():
    # Returns the main HTML page to show the video stream
    return render_template("index.html")


if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Fig. 24 Home Page Interface

Administrators can enter the camera URL and connect to the live stream, providing real-time surveillance of the premises. This feature ensures immediate response to unauthorized access attempts and continuous monitoring, enhancing the overall security management of the Al-Iman Center.

4.4 Testing

The test case results evaluate the functionality and security of the system through a structured test plan. The test plan includes functionality testing and security checks.

4.4.1 User Acceptance Form

The user acceptance form assesses the test cases of the suggested system from the user's perspective. Feedback from users, including PPA and employees, is gathered and analyzed to identify any issues or improvements needed, ensuring the system is practical and effective for real-world use. The user acceptance form is shown in Table 7. These test results ensure that the proposed system meets the functional and security requirements, and user feedback confirms its usability and effectiveness.

Table 7 Security Test Plan

No	Question	Result Dissatisfied 1 -7 Satisfied
Login Page		
1	Admin can log in without a problem	7
2	Admin can receive an error message if he inserted the wrong info.	7
3	Display message easy to understand	7
Admin Page		
4	Admin can add staff	7
5	Admin can remove staff	7
6	Admin can view the detected pictures	7
7	Admin can watch a Realtime camera video	6
8	Admin can add new administrator	7
9	Admin can logout	7
Raspberry Pi System		
10	The system can open the camera at a specific time	7
11	The system can detector staff and unauthorize people	7
12	The system can activate the alarm sound in the shop	7
13	The system can send the detected pictures to the admin app	7
Overall System		
14	The system works without a problem	6
15	The interface easy to understand	7
16	The system easy to understand	7

5. Conclusion

The security system designed for Al-Iman Center is a strong solution to protect the premises during non-operational hours. Using advanced technology like facial recognition, Raspberry Pi, and Google Firebase, the system quickly identifies and responds to unauthorized access. It shifts from passive to active monitoring by capturing intruders' images and alerting the admin app with a warning alart. The system has several advantages, such as real-time facial recognition, remote monitoring, and automatic activation during non-operational hours. Future improvements will include adding more security features like door access controls, smoke detection, and an application for staff attendance, as well as enhancing facial recognition accuracy and improving the user interface for easier use.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia for its support.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 94-113

Appendix A

Fig. 25 *Facial HOG Representation: Feature Capture [14].*

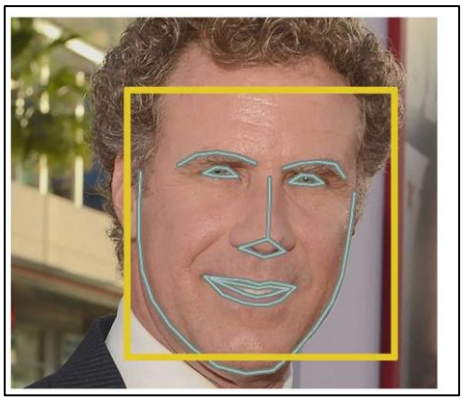

Fig.24 Facial *Landmark Alignment: Geometric Integrity Maintenance [14].*

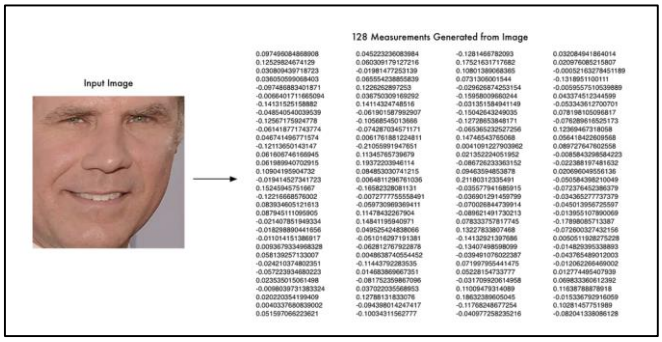

Fig. 25 *HOG Pattern Matching Process: Face Detection Alignment [13].*

Fig. 26 *HOG Pattern Matching Process: Face Detection Alignment [13].*

Appendix B

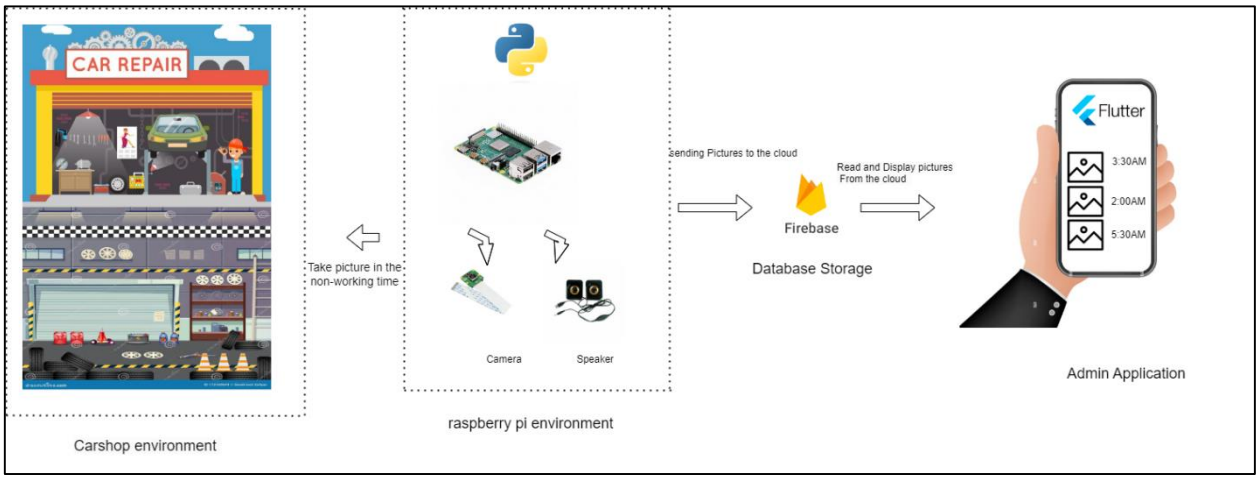

Fig. 27 *Project Framework for Al Iman Workshop*

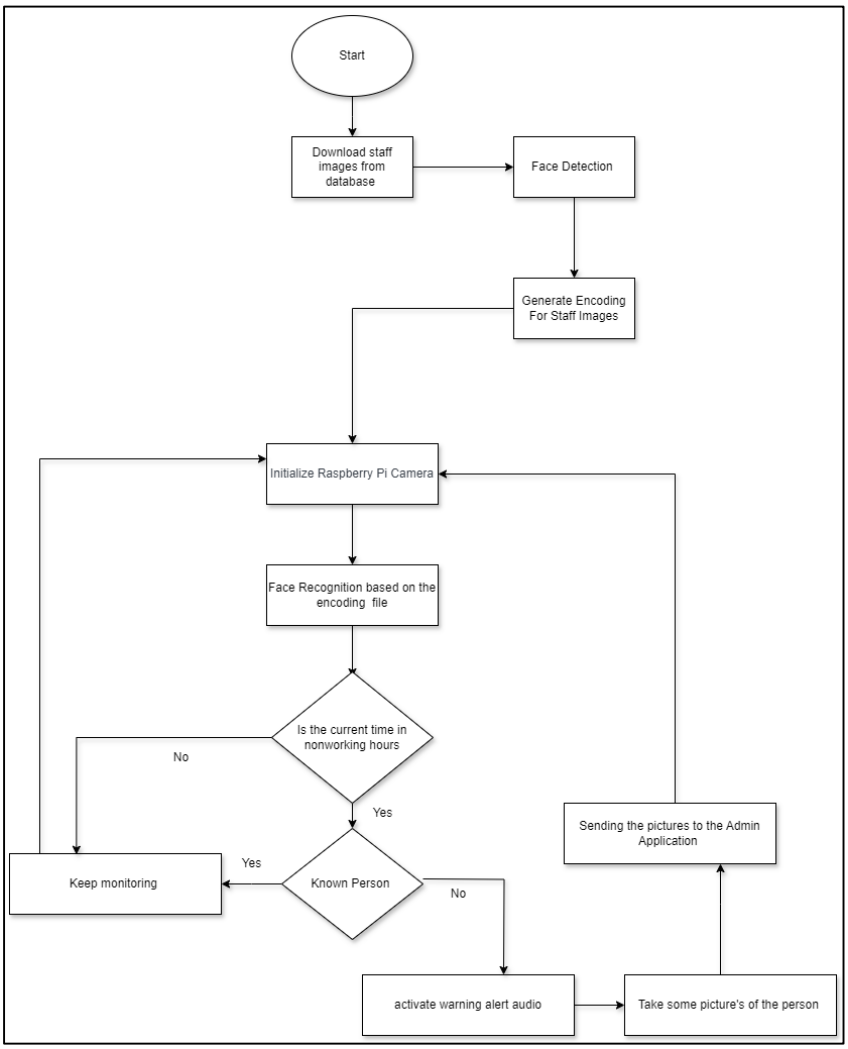

Fig. 28 *Project Framework Flowchart for Al Iman Workshop*

References

[1] ZHOU, J. Y. (2023). YOLO-Based Real Time Face Detection and Expression Recognition
[2] Das, B., & Halder, K. K. (2024, March). Face Recognition Using ESP32-Cam for Real-Time Tracking and Monitoring. In 2024 International Conference on Advances in Computing, Communication, and Informatics.
[3] Wang, Z., & Liu, T. (2022). Two-stage method based on triplet margin loss for pig face recognition. Computers and Electronics in Agriculture, 194, 106737.
[4] Dong, Y. (2023, November). Can machine recognize a long-missed old friend? A test to the FaceNet face recognition algorithm. Journal of Physics: Conference Series, 2634(1), 012055. IOP Publishing.
[5] Shendge, M. R., Patil, M. A., & Shendge, M. T. (2022). A Web-based Attendance System Using Face Recognition.
[6] Maurya, S., Kaur, B., & Rawat, P. (2023, July). Attendance management system using Python (Haar cascade and Open-CV). In International Conference on Green Energy, Computing and Intelligent Technology (GEn-CITy 2023) (Vol. 2023, pp. 438-444). IET.
[7] Anusha, P., Prasad, K. L., Kumar, G. R., Lydia, E. L., & Subbiah, V. (2020). "Facial Detection Implementation Using Principal Component Analysis (PCA)." 7(10), 1863–1872.
[8] Ali, A. S., & Hasan, D. S. (2023). An iot-based smart airport check-in system via three-factor authentication (3fa). Zanco Journal of Pure and Applied Sciences, 35(4), 1-13.
[9] Meddeb, H., Abdellaoui, Z., & Houaidi, F. (2023). Development of surveillance robot based on face recognition using Raspberry-PI and IOT. Microprocessors and Microsystems, 96, 104728.
[10] Allioui, H., & Mourdi, Y. (2023)). Exploring the full potential of IoT for better financial growth and stability: A comprehensive survey. Sensors, 23(19), 8015.
[11] Hsu, G. S. J., Wu, H. Y., Tsai, C. H., Yanushkevich, S., & Gavrilova, M. L. (2022). Masked face recognition from synthesis to reality. IEEE Access, 10, 37938-37952.
[12] Wang, Z., & Liu, T. (2022). Two-stage method based on triplet margin loss for pig face recognition. Computers and Electronics in Agriculture, 194, 106737.
[13] Thilaga, P. J. (2018). Modern Face Recognition with Deep Learning. SlideShare. Retrieved from https://www.slideshare.net/slideshow/modern-face-recognition-with-deep-learning/102557312
[14] VanderLaken, P. (2017). Facial Recognition Challenge. PaulVanderLaken.com. Retrieved from https://paulvanderlaken.com/2017/11/21/facial-recognition-challenge/
[15] Tutorialspoint. (n.d.). SDLC - Overview. Retrieved June 10, 2024, from https://www.tutorialspoint.com/sdlc/sdlc_overview.htm