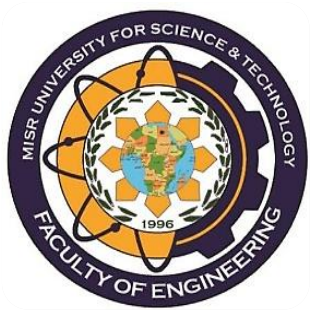


Misr University for Science and Technology
Faculty of Engineering
Department of Computer and Software

Secure computerized control system

based on AI with IoT compatibility to identify users.

BE. Final Project

Presented by: **Date:14/6/2025**

Adham Ahmed	**95630**
Salah Eldeen Yasser	**95675**
Bassel Ahmed	**95801**
Yousuf Ahmed	**97897**
Amr Khaled	**97910**
Ahmed Mostafa	**97924**

Supervised By **Dr. Abdelhameed Sharaf**

Misr University for Science and Technology
Faculty of Engineering
Department of Computer and Software

Secure computerized control system

based on AI with IoT compatibility to identify users.

BE. Final Project

Presented by: **Date:14/6/2025**

Adham Ahmed	**95630**
Salah Eldeen Yasser	**95675**
Bassel Ahmed	**95801**
Yousuf Ahmed	**97897**
Amr Khaled	**97910**
Ahmed Mostafa	**97924**

Supervised By **Dr. Abdelhameed Sharaf**

Acknowledgement

First and foremost, we would like to express our sincere gratitude to everyone who contributed to the successful completion of this project.

We extend our deepest appreciation to **our team members** for their dedication, collaboration, and hard work. Each member's efforts played a crucial role in overcoming challenges and achieving our goals.

We are especially grateful to our supervisor, **Dr. Abdelhameed Sharaf**, for his invaluable guidance, support, and constructive feedback throughout the project. His insights and encouragement were instrumental in refining our work. We also extend our sincere thanks to our mentor, **Eng. Nizar Hussein**, for his continuous support and technical advice, which greatly contributed to our project's development.

Thank you all for your support and contributions.

Special Thanks

We would like to express our deepest gratitude to the families of our team members, whose unwavering support and encouragement have been the cornerstone of our success throughout this graduation project. Your patience, understanding, and endless love provided us with the strength and motivation to overcome challenges and pursue our goals with determination. From late-night study sessions to moments of doubt, your belief in us never wavered, and for that, we are eternally grateful. This project is as much a testament to your sacrifices and support as it is to our hard work. Thank you for being our greatest cheerleaders and for making this journey possible.

Abstract

Traditional security systems often rely on single-factor authentication methods, such as keys, NFC cards, or keypads, which are susceptible to spoofing or unauthorized access. To address these vulnerabilities, we propose a versatile multi-factor authentication control unit that enhances security and usability.

Our system integrates face detection, fingerprint recognition, keypad access, and smartphone access via remote access with an easy-to-use GUI.

Users can configure the required authentication layers, ranging from one to three, to balance convenience and security. Phone access is prioritized for standalone authorization.

The system also includes live video communication feature, enabling admins to see the visitors remotely. With distinct user roles—admin and normal user—administrators can manage permissions, configure security settings, and promote users, while normal users have access-limited functionality.

This system provides a customizable and robust solution for modern access control needs with a simple GUI, combining advanced security features with user flexibility, and it can be applied to bank lockers, scientific high-security labs, security control rooms, and offices.

A smart door lock will serve as the target application for this system.

Table of Contents

Acknowledgement **2**

Special Thanks **2**

Abstract **3**

Table of Contents **4**

Table of figures **5**

Chapter 1 Introduction **7**

1.1 Problem Definition 7

1.2 Background 7

1.3 Project Significance 8

1.4 Challenges in Existing Systems 9

1.5 Project Objectives 9

1.6 Market Research 10

Chapter 2 Literature Review **13**

Chapter 3 Methodology **27**

3.1 Hardware 27

3.1.1 Hardware Components **27**

3.1.2 Hardware Circuits **39**

3.2 Intelligent Face Authentication System 56

3.2.1 AI Steps: **56**

3.2.2 Dataset Creation: **57**

3.3 System Software 64

3.3.1 GUI **64**

3.3.2 Back-end and System Integration **78**

3.4 Remote Access App 81

3.4.1. Software Configuration for Blynk **81**

3.4.2. Hardware Configuration for Blynk **86**

Chapter 4: Conclusion **87**

4.1 Limitations 87

4.2 Future Work 88

4.3 Final Remarks 89

References **90**

Table of figures

Figure 1.6.1 - Device 1 Outdoor view 10
Figure 1.6.2 - Device 1 Indoor view 10
Figure 1.6.3 - Device 2 Outdoor View 11
Figure 1.6.4 - Device 2 Indoor view 11
Figure 1.6.5 - Device 3 Outdoor View 11
Figure 1.6.6 - Device 3 Indoor View 11
Figure 1.6.7 - Device 4 Outdoor view 12
Figure 1.6.8 - Device 4 Indoor view 12
Figure 3.1.1.1.1 - Raspberry PI 5 27
Figure 3.1.1.1.2 - MPI3201 Touch Screen for Raspberry PI 28
Figure 3.1.1.2.1 – NodeMCU ESP8266 29
Figure 3.1.1.2.2 - 74HC4067 16-Channel Analog Digital Multiplexer Breakout Board 30
Figure 3.1.1.2.3 – SR501 PIR Sensor 30
Figure 3.1.1.3.1 - 12V 10A Power Supply 31
Figure 3.1.1.3.2 - 12.6V 10A 3S BMS 31
Figure 3.1.1.3.3 - 18650 5000mAh Li-ion Cell 32
Figure 3.1.1.3.4 - 12V 10A SPDT Relay 32
Figure 3.1.1.3.5 - 9-90V to 5V 3A Step Down converter 33
Figure 3.1.1.4.1 - 5V 10A SPDT Relay 34
Figure 3.1.1.4.2 - Solenoid Lock 34
Figure 3.1.1.5.1 - USB to 3.5mm Sound Card 35
Figure 3.1.1.5.2 – 1.5W 8Ohm Speaker 35
Figure 3.1.1.6.1 - Touch Sensor Module 36
Figure 3.1.1.6.2 - Raspberry PI Camera V1.3 37
Figure 3.1.1.6.3 - R301T Fingerprint Module 38
Figure 3.1.2.1.1 - Power System Block Diagram 39
Figure 3.1.2.1.2 - Power System Circuit Connection 40
Figure 3.1.2.1.3 - Power System Live Photo 40
Figure 3.1.2.2.1 - Main Unit Block Diagram 43
Figure 3.1.2.2.2 - Main Unit Live Photo 1 44
Figure 3.1.2.2.2 - Main Unit Live Photo 2 45
Figure 3.1.2.3.1 - ESP8266 Board Block Diagram 48
Figure 3.1.2.3.2 - ESP8266 Board Live Photo 1 49
Figure 3.1.2.3.3 - ESP8266 Board Live Photo 2 50
Figure 3.1.2.4.1 - Electric Lock Block Diagram 54
Figure 3.1.2.4.2 - Electric Lock Circuit Connection 54
Figure 3.2.1.4 - Flowchart of face recognition system 61
Figure 3.2.1.5 - Performance metrics for ArcFace 62
Figure 3.2.1.6 - Confusion matrix for ArcFace 62
Figure 3.2.1.1– table of comparison of different approaches of AI 63
Figure 3.3.1.1 GUI Home Screen 65
Figure 3.3.1.2 Admin Login Screen 66
Figure 3.3.1.3 Main Settings Screen 67
Figure 3.3.1.4 User Management Screen 68
Figure 3.3.1.5 Add New User Screen 69
Figure 3.3.1.6 Fingerprint Enrollment Screen 70

Figure 3.3.1.7 Face Enrollment Screen 71
Figure 3.3.1.8 Profile List Screen 72
Figure 3.3.1.9 Edit Profile Screen 73
Figure 3.3.1.9 Role Change Screen 74
Figure 3.3.1.10 Delete Profile Confirmation Screen 75
Figure 3.3.1.11 Device Settings Screen 76
Figure 3.3.1.12 Input Keypad Screen 77

Chapter 1 Introduction

1.1 Problem Definition

Traditional authorization systems, such as pin-number locks, lever locks, and deadbolt locks, have been in use for centuries and are a staple in ensuring security for residential and commercial properties. While these locks offer basic protection, they come with several significant limitations that can compromise their effectiveness in modern security environments.

One of the key issues with traditional locks is their vulnerability to common bypass techniques such as lock picking, bumping, and key duplication. These methods can allow unauthorized individuals to gain access to secured spaces, potentially leading to theft, property damage, or safety risks. Furthermore, traditional locks are also prone to wear and tear overtime, which may result in mechanical failures, causing difficulty in locking or unlocking doors.

Additionally, the reliance on physical keys presents its own set of challenges. Keys can be easily lost or stolen, creating security gaps if not immediately reported or replaced. In the case of multiple entry points in a building, managing and distributing numerous keys can become cumbersome and inefficient, leading to logistical difficulties.

In today's increasingly connected world, traditional authorization systems do not meet the demands of modern technological advancements. For instance, they lack integration with digital security systems, such as remote access control, smart home automation, or advanced monitoring capabilities. As a result, there is a need to explore alternative solutions that not only address these vulnerabilities but also meet the evolving expectations of security, convenience, and technological integration.

This research aims to analyze the limitations of traditional authorization systems, assess their impact on personal and public security, and propose modern alternatives that enhance both physical and digital security for various applications.

1.2 Background

Traditional authorization systems have played a pivotal role in safeguarding homes, businesses, and other secure facilities for centuries. The earliest forms of mechanical locks date back to ancient civilizations, with evidence of locks used by the Egyptians around 4,000 years ago. These early locks, constructed from wood and rudimentary metal components, were designed to prevent unauthorized access to private spaces. Over time, advancements in materials, design, and technology gave rise to the more sophisticated mechanical locks we use today.

The most used traditional locks are the pin-tumbler lock, lever locks, and deadbolt locks. The pin-tumbler lock, invented by Linus Yale Jr. in the mid-19th century, revolutionized security

by using a series of pins of varying lengths to prevent the lock from opening unless the correct key was inserted. Lever locks, commonly found in older doors, operate through a system of levers that must be lifted by a key to unlock the mechanism. Deadbolts, which are widely used in modern homes, are known for their strength and reliability, offering added security by requiring the turning of a bolt to engage or disengage the lock.

While these traditional locks have remained standard in residential and commercial security, their design has largely remained unchanged for decades. They provide basic security by acting as physical barriers to unauthorized entry. However, as society has evolved and technology has advanced, the limitations of traditional locks have become more apparent. The mechanisms within these locks can be manipulated through various bypass techniques such as lock picking, bumping, and key duplication. Furthermore, traditional locks are mechanical devices, making them prone to wear and tear, which can compromise their functionality over time.

In addition to their vulnerability to external manipulation, traditional locks often rely on physical keys, which introduce their own set of challenges. Keys can be lost, stolen, or duplicated, leading to security breaches if not properly managed. The inconvenience of managing many keys in larger buildings or complexes can also be a logistical nightmare, as it becomes increasingly difficult to track key distribution and ensure the right individuals have access to the necessary entry points.

Despite these limitations, traditional authorization systems continue to be widely used due to their affordability, simplicity, and general reliability. However, with the rise of smart home technologies and growing concerns about security breaches, there is a need to evaluate the viability of traditional locks in comparison to newer, more secure alternatives.

This background sets the stage for further exploration into the strengths and weaknesses of traditional authorization systems and the need for innovative security solutions that meet the demands of modern society.

1.3 Project Significance

Security and access control are fundamental aspects of modern infrastructure, ensuring the safety of individuals, assets, and sensitive data. Traditional authentication methods, such as keys, PIN codes, and NFC cards, are increasingly vulnerable to security breaches, making it essential to adopt more advanced and adaptive solutions.

To overcome these issues, we propose a multi-factor authentication smart control unit that integrates face detection, fingerprint recognition, keypad access, and smartphone connectivity. The system offers real-time video communication, configurable authentication layers, and role-based access management, ensuring both security and flexibility.

1.4 Challenges in Existing Systems

Despite advancements in security technology, many existing accesses control solutions suffer from key limitations:

1. **Single-Factor Authentication Risks** – Using only one authentication method increases the risk of unauthorized access through lost keys, stolen NFC cards, or hacked PIN codes.
2. **Lack of Multi-Factor Authentication (MFA)** – Many systems do not allow users to configure layered security, making them rigid and less adaptable to varying security needs.
3. **Limited Remote Access & Monitoring** – Most traditional systems do not provide real-time video communication, restricting the ability to verify visitors remotely.

1.5 Project Objectives

The development of a secure computerized control system based on Artificial Intelligence (AI) and compatible with the Internet of Things (IoT) to identify users is a critical advancement in modern security systems. With the increasing complexity and interconnectivity of today's digital landscape, traditional security measures, such as manual locks and basic authentication, are becoming insufficient to address the growing threats to privacy, property, and critical infrastructure.

This project aims to leverage cutting-edge AI algorithms and IoT integration to create a robust and adaptive security system that offers several key advantages:

1. Enhanced Security through AI-Powered User Identification:

Traditional security systems often rely on static methods like PINs, passwords, or physical keys, which can be easily bypassed or compromised. AI-driven user identification systems, such as biometric recognition (e.g., facial recognition, fingerprint scanning), provide a dynamic and personalized approach to authentication.

2. Scalability and Flexibility for Diverse Applications:

The secure computerized control system is highly scalable, making it suitable for a wide range of applications—from residential properties to large-scale commercial complexes and government buildings. Whether it's a multi-user home, office, or a high-security facility, the system can be configured to match the security requirements of any space.

3. Improved User Experience and Convenience:

One of the significant benefits of this system is the seamless and user-friendly experience it offers. Traditional locks and access systems often require the user to physically engage with a key or code, which can be inconvenient or problematic if keys are lost or forgotten. With AI and IoT, users can access their premises or devices without physical interaction, using methods

such as face recognition. This not only enhances convenience but also ensures quicker, hassle-free entry, especially in environments with high foot traffic, such as offices or apartment buildings.

4. Reduction of Human Error and Unauthorized Access:

The AI-driven system significantly reduces human error, such as improper key handling, forgotten codes, or negligence in locking doors. Moreover, it offers real-time monitoring of access patterns, enabling administrators to quickly identify suspicious activities or unauthorized attempts to breach the system. This proactive approach to security helps detect threats early, allowing for prompt corrective action.

1.6 Market Research

We have conducted research in the local market about our product to see what the features and designs are in the market and to find out who the manufacturers are and the manufacturing costs to develop an appropriate plan that will lead us to a product that is very close to what is available on the market.

Manufacturer	Cost (EGP)	Outdoor Picture	Indoor Picture
Cordless	**16900**	 *Figure 1.6.1 - Device 1 Outdoor view*	 *Figure 1.6.2 - Device 1 Indoor view*

Cordless	**16500**	 *Figure 1.6.3 - Device 2 Outdoor View*	 *Figure 1.6.4 - Device 2 Indoor view*
LZEN	**13500**	 *Figure 1.6.5 - Device 3 Outdoor View*	 *Figure 1.6.6 - Device 3 Indoor View*

Cordless	**4200**	 *Figure 1.6.7 - Device 4 Outdoor view*	 *Figure 1.6.8 - Device 4 Indoor view*

We found that:

- The most popular brands in the market are Cordless and Lzen.
- Prices range from 13500 to 18000 EGP.
- All products have remote access methods.
- It's common to use a motion sensor to save power.
- Most Products use Electrical Mortise Lock

Chapter 2 Literature Review

[1] IoT Assisted Fingerprint-Based Door Security System using Raspberry Pi 4

Date: April 2020
Journal: International Journal of Trend in Scientific Research and Development (IJTSRD) Volume 4 Issue 3
Approach:

This study presents an IoT-assisted fingerprint-based door security system using a Raspberry Pi 4 integrated with an R307 optical fingerprint sensor. The system is connected to an IoT network, enabling real-time email alerts in case of unauthorized access attempts.

Model and Devices:

1-Fingerprint Technology: **R307 optical fingerprint sensor** (used for capturing and verifying fingerprints).
2-Control Unit: **Raspberry Pi 4** (central processing unit for handling the fingerprint data).
3-Additional Devices: **GSM Module** (used for sending alerts), **Servo Motor** (for door operation).
4-IoT Integration: Real-time communication via the internet for sending notifications.
Key Features and Benefits:

1-IoT Integration: Provides real-time alerts and monitoring, improving security and convenience.
2-Efficiency: The system processes fingerprint data efficiently and ensures a smooth user experience.[1]

[2] Real Time Access Control Based on Face Recognition

Date: May 2023

Journal: 2015 International Conference on Network security & Computer Science (ICNSCS-15) Antalya (Türkiye)

Approach:
This project focuses on a hardware-based face recognition access control system using the Raspberry Pi. By employing Principal Component Analysis (PCA) for recognition and Haar-like features for face detection, the system addresses modern security challenges. The device includes essential hardware like a Raspberry Pi, Pi Camera, and a magnetic lock, providing an affordable and efficient solution for secure access management.

Model:

This device works by allowing the lock to open through facial recognition using Raspberry pi 3 B+ as well as Camera Pi 5MP as the core of their HW. The face recognition part is done through a combination of PCA and Haar-like features for real time applications and to decrease the computational load on Raspberry Pi.

Dataset: In this paper they made their own dataset with 5 positions and 10 images for a total of 50 positive images per person.

Accuracy: After training the model locally with variation in controlled parameters (Lighting, Poses, Number of photos) they found a 36% to 62% accuracy jump, which shows an issue with lighting and pose changes.

Hardware component and integration:

1-Raspberry Pi 3 B+: Is used for all the processing, identification and training of the model due to its high processing capacity, relatively low price and ease of use.

2-Camera Pi 5MP: can be connected to the Raspberry Pi through CSI which allows for Resolution 2592x1944 for image and 1080p30 and 720p60 for video.

3-Other Components: Such as transistors, resistors and a magnetic lock.

Key Features and Benefits:

1-Real time Use: PCA allows for quick identification of faces

2-Scalability: PCA allows for efficient Database usage

3-Local Processing: RPi 3 B+ allows for training without internet access.[2]

[3] Human Face Detection & Recognition Using Raspberry Pi

Date: (ICSESD-2017)

Journal: International Journal of Advanced Engineering, Management and Science

Approach:

The paper talks about a method of implementing quick real time face detection while keeping a high true positive rate. This is done through Raspberry Pi for processing and using python as their main language and testing their system across many standard face databases.

Model:

The system works by capturing an image using webcam then applying Pre-processing, face detection, ROI, Local binary pattern histogram, Machine learning predict sequentially on every captured image to detect where the face is.

Dataset: the system was tested on many standard databases such as At& T, Caltech, Indian Face Database, JAFFE, YALE B, Face94, Face95, Face96, Grimace.

Accuracy: Not mentioned in the paper since this is only detection not recognition.

Hardware Components and Integration:

1-Raspberry Pi 2 B: Used as the processing unit of the system for preprocessing and identification of faces in an image.

2-Camera module: Used to capture a video or image to pass to the Raspberry Pi.

3-Software: Used Python, OpenCV, Haar classifier and image processing algorithms.

Key Features and Benefits:

1-Hardware-Driven Efficiency: Uses Raspberry Pi with OpenCV for real-time face detection and recognition.
2-Advanced Algorithms: Incorporates Haar Classifiers and PCA for accurate detection and recognition.
3-Cost-Effective & Portable: Affordable and compact, ideal for diverse environments.
4-Versatile Applications: Supports security, attendance, and automation tasks effectively.
5-High Performance: Maintains excellent detection rates even with low-quality or noisy images.[3]

[4] Smart Door System using Face Recognition Based on Raspberry Pi

Date: December 2021

Journal: JURNAL INFOKUM, Volume 10, No.1

Approach:
This study demonstrates a hardware-based approach to a smart door system, leveraging facial recognition for access control. The Raspberry Pi serves as the core controller, integrated with a webcam and a solenoid lock. The system is designed to enhance security by replacing traditional alphanumeric passwords with biometric facial recognition. Tests revealed a high accuracy rate, emphasizing its viability for real-world applications.

Model:

This system works by obtaining an image from webcam then performing preprocessing, face detection, feature extraction and recognition then it allows or denies access. This is done through a Raspberry Pi for processing and Viola-Jones + Eigenface for recognition.

Dataset: the dataset here is 10 images per student for 14 students for 140 images.

Accuracy: after calculating the accuracy per person which ranged from 70% to 100% the average value of them was 94%.

Hardware Components and Integration:

Raspberry Pi: Mini-computer handling image processing, facial recognition, and control functions.

Webcam: Captures facial images for analysis.

Solenoid Lock: Controls the physical locking and unlocking of the door.

Buzzer: Alerts in case of unauthorized access attempts.

LCD Display: Provides feedback on the recognition status.

Key Features and Benefits:

Automated Smart Door: Keyless entry with a webcam, solenoid lock, and security alarm.
High Accuracy: Reliable 94% recognition success across varying conditions.
Convenience and Scalability: Simplifies access and adapts to various security applications.
Facial Recognition Security: Biometric access control using Raspberry Pi and Eigenface PCA technology.[4]

[5] Face Recognition for Smart Door Lock System Using Hierarchical Network

Date: October 8-9,2020

Journal: 2020 International Conference on Computational Intelligence (ICCI)

Approach:
The paper proposes a hierarchical network (HN) framework to improve face recognition accuracy and resilience. And demonstrates the application through a real-time prototype of a face recognition door lock system.

Model:

Face Recognition: Measures facial points for unobtrusive human identification.

Hierarchical Network (HN):

Two-tier recognition using ResNet101 and FaceNet.

Validates recognition results to minimize false positives.

Dataset: In this paper, they made their own dataset with 12 students and 50 images for each person 35 for training and 15 for testing each image is 3024×4032 JPEG format, mages were taken with different poses at different places under the different lighting conditions to ensure more accuracy for recognition

Accuracy: Enhanced accuracy (~87.36%) using the HN framework compared to a single-tier FaceNet approach (~72.43%).

Hardware component and integration:

Raspberry Pi 4 for system control Is used for all the processing, identification and training of the model due to its high processing capacity, relatively low price and ease of use.

Camera: Not mentioned

Servo motor for door actuation: Such as transistors, resistors and a magnetic lock.

Key Features and Benefits:

1- Highlight the importance of a two-tier recognition system to avoid spoofing.

2- Provides an implementation guide using Raspberry Pi and Servo motor

3- Suggests methods for integrating face recognition with additional security measures (e.g., email notifications).[5]

[6] RFID and Fingerprint Based Dual Security System: A Robust Secured Control to Access Through Door Lock Operation

Date: June 15, 2018

Journal: American Journal of Embedded Systems and Applications

Approach:

Design a robust, biometric-based door-locking system for enhanced security using fingerprint authentication and RFID where the lock opens on correct identification and does nothing for unauthorized individuals.

Model:

This system captures and verifies fingerprints against a stored database by utilizing optical or capacitance scanning methods for precision. Its main steps are Image Acquisition, Normalization, Thinning and Feature Extraction.

Dataset: This system does not rely on a dataset and instead functions on correlation between present users and previously stored filtered fingerprints.

Accuracy: no accuracy was submitted in this paper but a table showing that whenever either RFID or fingerprint matching failed the door did not open

Hardware Components and Integration:

1-Arduino Uno Board: Used for the processing and matching of RFID and fingerprint

2-Servo motor: Used to open the door upon granting access

3-RFID unit: Used to check the validity of RFID card for increased security.

4-Fingerprint unit: Used to scan fingerprints and output an image to the processing unit for enhancement and matching.

Key Features and Benefits:

1-Sequential authentication process involving a fingerprint sensor.

2-Strong security measures to prevent unauthorized access.

3-Automatic door control using a servo motor linked to the fingerprint system.[6]

[7] A Model of Secured ATM Pin Recovery with Face and Fingerprint Identification

Date: 28th October 2022
Journal: National Conference on Contemporary Research and Computer Intelligence 2022
Approach:

This paper addresses the problem of forgotten ATM PINs by integrating biometric solutions into the PIN recovery process. Uses facial recognition and fingerprint identification to verify user identity during PIN recovery.
Model:

The system integrates facial recognition, fingerprint matching, and PIN recovery for secure biometric authentication. Facial features are extracted using fine-tuned CNN models like ResNet-50, optimized with triplet loss, while fingerprint patterns are analyzed with Gabor filters and ridge mapping. An added security layer verifies biometrics before generating and encrypting temporary PINs for secure access.
Dataset: Leveraged datasets such as CASIA-WebFace or MS-Celeb-1M, providing labeled facial images which include diverse age groups, ethnicities, and lighting conditions for Face Recognition. and took high-resolution fingerprint scans from databases like NIST Special Database 4 as Fingerprint dataset. As well as Synthetic data pairs PINs with associated face and fingerprint biometrics to simulate ATM recovery scenarios.
Accuracy: Face and Fingerprint Recognition Combined recognition accuracy exceeds **98%**, ensuring low false rejection rates. PIN Recovery Success Rates of 99% in controlled tests, with minimal failure due to biometric mismatches. FAR: < 0.05% and FRR: < 0.5%.
Hardware Components and Integration:

1-Camera Modules: Capture high-quality facial images for authentication during PIN recovery.
2-Fingerprint Scanners: Serve as the second biometric input for verifying the user's identity.
3-Encryption and Processing Units: Encrypt biometric inputs (facial data, fingerprint scans) before transmitting to the bank's central database.
4-Touchscreen Displays: Facilitate user interaction during the PIN recovery process.
Key Features and Benefits:

1-Seamless PIN Recovery: Eliminates the need for bank visits or manual recovery processes.
2-Enhanced Fraud Prevention: Verifies identity using two unique biometric inputs, reducing the risk of impersonation.
3-User Convenience: Simplifies the process for users who forget their PINs, especially in emergencies.
4-Real-Time Security: Instant PIN issuance minimizes downtime and disruption for users. [7]

[8] Secure Access Microcontroller System Based on Fingerprint Template with Hyperchaotic Encryption

Date: 5 January 2023

Journal: Integration, the VLSI Journal

Approach:
This system employs a fingerprint-based biometric authentication mechanism enhanced with hyperchaotic encryption. It addresses the vulnerabilities of traditional biometric systems by combining chaos theory and cryptographic algorithms, ensuring that fingerprint is secure
Model:

Fingerprints are enrolled, and minutiae are extracted to create a digital template. During registration, the template is hashed using SHA-1, producing a 160-bit hash key. The hash key is combined with a personal symmetric key to generate an encrypted cryptogram using the hyperchaotic Lotka–Volterra map. For authentication, the encrypted fingerprint data is decrypted and matched against the stored template.
Dataset: The system uses fingerprint data captured using the Futronics FS83 fingerprint module. The fingerprint templates are digitized into 2,072 bytes per sample and stored locally or in the cloud after encryption.
Accuracy: The system underwent rigorous security analysis, including entropy testing and autocorrelation, ensuring cryptographic robustness. No explicit accuracy metrics (e.g., False Acceptance Rate or False Rejection Rate) for the biometric matching were reported, but the chaotic encryption's statistical properties ensured secure and reliable encryption.
Hardware Components and Integration:

1-Microcontroller: Freescale ColdFire MCF52259, chosen for its independent operation capabilities, flash memory, and GPIO ports.
2-Fingerprint Reader: Futronics FS83 module, which captures and processes fingerprint data.
3-LCD Display: A 20x4 matrix orbital LCD for user interaction.
4-Storage: Internal 512 KB flash memory for storing encrypted templates.
5-Development Environment: The system was programmed using Freescale CodeWarrior IDE with MQX real-time OS.
Key Features and Benefits:

Double Security Layer: Combines hyperchaotic encryption and SHA-1 hashing for enhanced security.
Efficient and Low-Cost: Designed to function with minimal hardware resources, making it cost-effective.
High Sensitivity: Utilizes hyperchaos for a large secret key space, ensuring high resistance to attacks.
Comprehensive Security Analysis: Includes entropy analysis, autocorrelation, and key sensitivity validation.
Applications: Suitable for restricted areas such as banks, offices, and secure facilities.[8]

[9] A Fingerprint & Pin Authentication to Enhance Security at The Automatic Teller Machines

Date: April-2017
Journal: International Journal of Scientific & Engineering Research, Volume 8, Issue 4
Approach:

The research proposes a multi-factor authentication system that integrates two security layers. A biometric fingerprint scan and a four-digit PIN for identity verification.
AES Algorithm Optimization: Introduces an optimized Advanced Encryption Standard (AES) algorithm to ensure secure communication between the ATM and the bank server.
Model:

This system provides security through AES encrypted fingerprint and PIN code for identity verification where the process of Training consists of fingerprint recognition model where features are extracted using minutiae-based algorithms and a classifier such as a SVM or a CNN is trained on pre-processed fingerprint data.
Dataset: Fingerprint Dataset sourced from public fingerprint databases such as FVC2004 or custom datasets collected from ATM users during trials. Typically includes a diverse set of fingerprint images with variations in quality, orientation, and pressure. While the PIN Dataset is Synthetic PIN combinations generated for testing encryption and matching algorithms. No actual user data is used to ensure privacy compliance.
Accuracy: Fingerprint Recognition Achieves 95-98% accuracy in correctly matching fingerprints under ideal conditions. While False Acceptance Rate (FAR): < 0.1% and False Rejection Rate (FRR): < 1%. Biometric + PIN verification reduces authentication errors, achieving an overall system reliability of 99% in trials.
Hardware component and integration:

1-Fingerprint Scanners: Capture and authenticate the unique fingerprint patterns of users.
2-AES Encryption Processor: Encrypts sensitive data (e.g., fingerprint and PIN)
3-Keypad and Display Interface: Enables users to input their PIN and interact with the system.
4-Secure Communication Module: Ensures encrypted data exchange between the ATM and the central server.
Key Features and Benefits:

Enhanced Security: Combines something the user knows (PIN) with something the user is (fingerprint), greatly reducing risks of identity theft, card skimming, or PIN observation. AES encryption protects against unauthorized access during data transmission.
Reliability: Fingerprints provide unique, immutable data, making them a secure alternative to traditional passwords and PINs.
Fraud Prevention: Addresses common ATM fraud issues like card cloning, PIN theft, and impersonation.[9]

[10] Enhanced Cloud Security Model Using Combined Multiple Biometric Features

Date: 28th October 2022
Journal: National Conference on Contemporary Research and Computer Intelligence 2022
Approach:

This paper introduces a hybrid biometric authentication model designed to protect sensitive cloud data by using multiple biometric inputs (e.g., fingerprint, facial recognition) to enhance identity verification while focusing on integrating biometric authentication with data encryption mechanisms for cloud security.
Model:

This biometric system employs specialized algorithms for feature extraction and fusion. Iris recognition uses an integrodifferential operator, while vein patterns are detected via repeated line tracking and maximum curvature methods. Facial features are mapped with models like FaceNet, and fingerprints are processed with CNNs and minutiae detection. Fusion of features is achieved using ensemble methods, trained with transfer learning in frameworks like TensorFlow.
Datasets: Facial Recognition Dataset is sourced from databases like LFW or custom collections of real-world face images. Includes a diverse range of images across lighting conditions, angles, and expressions. Fingerprint Dataset is made from Publicly available datasets (e.g., FVC series) or proprietary biometric databases. Includes clean, noisy, and distorted fingerprint samples for robust training. Combined Dataset Pairs of face and fingerprint data linked to a single user ID to test multi-modal biometric authentication.
Accuracy: Facial Recognition Accuracy was 90-95% under controlled conditions. And Fingerprint Recognition Accuracy was 95-98%. Multi-Modal System Accuracy was 99% accuracy by combining both biometric inputs, significantly reducing False Acceptance Rates (FAR < 0.05%).
Hardware Components and Integration:

1-Facial Recognition Cameras: Capture high-resolution images of users' faces for biometric authentication.
2-Fingerprint Scanners: Capture and match fingerprint patterns for user verification.
3-Data Encryption Hardware: Encrypts biometric data before uploading to the cloud storage.
4-Cloud Servers: Store encrypted biometric templates and authenticate user requests.
Key Features and Benefits:

1-Multi-Layer Security: Prevents unauthorized access by requiring two biometric inputs.
2-Enhanced Accuracy: Reduces false positives/negatives by using multiple biometric features.
3-Real-Time Access Control: Enables instantaneous validation of users trying to access cloud resources.
4-Scalability: Suitable for integration with various cloud platforms, supporting a wide range of users and devices.[10]

[11] Face Recognition Based Security System Using Raspberry Pi.

Date: 2021, JETIR July 2021, Volume 8, Issue 7
Journal: Journal of Emerging Technologies and Innovative Research (JETIR)
Approach:

The paper proposes an automatic door access system that leverages face detection and recognition for enhanced security. The system eliminates the need for physical keys, passwords, or cards by employing facial recognition as the primary authentication method. It uses Viola-Jones for face detection and Principal Component Analysis (PCA) for recognition, providing a seamless and efficient solution.
Model:

The system works by capturing the face image, detecting the facial region using Viola-Jones, and recognizing the individual by comparing features extracted via PCA to a stored database. Known individuals trigger door access, while unknown individuals activate an alarm.
Dataset: PCA efficiently reduces the dimensionality of the image data, enabling faster and accurate recognition. The dataset includes frontal facial images of authorized users.
Accuracy: The PCA-based approach can recognize images within one second but has limitations with non-frontal face orientations, potentially reducing accuracy in dynamic real-world scenarios.
Hardware Components and Integration:

1. Raspberry Pi: Serves as the central control unit, interfacing with the camera module and microcontroller.
2. Camera Module: Captures facial images for detection and recognition. Supports various resolutions (up to 12 MP for newer models).
3. Microcontroller: Controls door operations based on recognition results received from the Raspberry Pi.
4. Other Components: Includes an automated door lock and an alarm system for unauthorized access.
Functionality: The Raspberry Pi processes the face data, triggers the microcontroller to open or lock the door, and activates the alarm system when an unauthorized face is detected.
Key Features and Benefits:

1. Versatile Application: Suitable for residential and industrial security systems.
2. Fast and Reliable: Recognizes authorized faces within one second using PCA.
3. Scalable Storage: PCA efficiently handles multiple facial profiles in the database without significant performance degradation.
4. Real-Time Alerts: Immediate alarm activation for unknown individuals increases situational awareness.[11]

[12] Prevention of Unauthorized Door Access Using Face Recognition

Date: first posted online: 20 May 2020

Journal: International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016

Approach

The system integrates IoT with facial recognition technology to enhance security for door access control. It replaces traditional key, ID card, and password-based systems, addressing their vulnerabilities such as theft or forgery. This smart door system uses a Haar Cascade Classifier for face detection and Histogram of Oriented Gradients (HOG) for feature extraction. If an unauthorized individual attempts access, their image is captured and sent via Discord webhooks for remote monitoring.

Model:
Haar Cascade Classifier is trained using OpenCV tools ("opencv_createsamples" and "opencv_haartraining"). When a face is detected, its encoding is compared against a pre-trained dataset. If the face matches the Door unlocks via a servo motor but If unmatched the image is sent to a Discord server with a timestamp.
- **Dataset**: The system was trained using 1,000 positive images (with faces) and over 10,000 negative images (without faces), maintaining a 1:10 ratio for effective training. **Accuracy:** The combination of Haar Cascade and HOG ensures robust detection, though specific metrics are not detailed in the paper.
Hardware Components and Functionality:

1. **Raspberry Pi 3 Model B:** Acts as the processing unit, running OpenCV and managing door mechanisms.
2. **Camera:** Captures images for facial recognition.
3. **Servo Motor:** Controls the door lock mechanism, unlocking only for authorized users.
4. **Push Button:** Triggers the camera to capture an image.
5. **Internet Connection:** Enables communication with the Discord server for remote alerts.
6. **Software:** Uses Python, OpenCV, Raspbian OS, and Discord API for system operations.

Key Features and Benefits:

- **Enhanced Security:** Recognizes authorized users and alerts the admin about unauthorized access with captured images.
- **Remote Monitoring:** Uses Discord webhooks to send real-time intrusion alerts.
- **Low Cost and Maintenance**: Reduces dependency on expensive databases by leveraging Discord for storing intruder images.
- **Efficiency:** Rapid recognition enabled by pre-stored facial vector encodings.[12]

[13] A Prototype Model of an IoT-based Door System using Double-access Fingerprint Technique

Date: JUNE 2020

Journal: NIGERIAN JOURNAL OF TECHNOLOGICAL DEVELOPMENT, VOL. 17, NO.2,

Approach:

This paper presents the design and implementation of a **smart door lock system** using the **LLC751 optical fingerprint sensor** for biometric authentication. The system also integrates **GSM technology** to enable remote monitoring and notifications. The goal of the system is to enhance security by replacing traditional locks with a biometric solution that grants access only to registered users. The system works by scanning a user's fingerprint using the LLC751 optical sensor. If the fingerprint matches one stored in the system, it activates the **servo motor** to unlock the door. If an unauthorized access attempt is detected, the system triggers **GSM alerts**, notifying the owner of the breach attempt.

Model:

Components:
1-Fingerprint Sensor: **LLC751 optical fingerprint sensor** used for capturing and verifying fingerprints with high accuracy.
2-Control Unit: **ATmega328P microcontroller** processes the fingerprint data and controls the locking mechanism.
3-Communication Module: **GSM module** sends SMS alerts and enables remote control.
4-Locking Mechanism: **Servo motor** used for locking and unlocking the door.
Process:
The fingerprint is scanned using the LLC751 optical sensor and compared with pre-stored templates. If there is a match, the servo motor unlocks the door. Unauthorized access triggers an SMS alert sent via the GSM module.
Key Features and Benefits:

1-Enhanced Security: The **LLC751 optical fingerprint sensor** ensures precise and secure fingerprint verification with resistance to spoofing.

2-Remote Monitoring: The **GSM technology** allows real-time alerts, notifying the user of unauthorized access attempts.

3-Cost-Effective: The use of widely available and affordable components, like the **ATmega328P microcontroller**, ensures an accessible and scalable solution.[13]

[14] Finger Scanner: Embedding a Fingerprint Scanner in a Raspberry Pi

Date: 6 February 2016

Journal: Department of Computer Science and INSPIRES, University of Lleida

Approach:

This study presents a low-cost fingerprint biometric authentication system designed using the GT-511C1R optical fingerprint sensor and a PIC16F877A microcontroller. The system is intended for access control applications, such as smart door locks, providing an affordable and secure alternative to traditional key-based systems.

The fingerprint sensor captures and compares user fingerprints with stored templates, granting or denying access accordingly. The microcontroller serves as the central processing unit, controlling the locking mechanism based on authentication results.

Model:

Components:
1-Fingerprint Sensor: **GT-511C1R optical fingerprint sensor** for accurate and secure fingerprint recognition.
2-Microcontroller: **PIC16F877A** processes fingerprint data and controls the door lock mechanism.
3-Door Lock Mechanism: A servo motor operates the locking and unlocking system.
Process:
The system scans the user's fingerprint through the GT-511C1R sensor. The captured data is compared with pre-stored templates. If a match is found, the servo motor unlocks the door.
Key Features and Benefits:

1-High Security: The use of a **optical fingerprint sensor** enhances protection against spoofing and unauthorized access.
2-Cost-Effective Design: By leveraging low-cost components, the system offers affordability for small-scale and home applications.
3-Scalability: The modular design allows for the addition of features such as GSM alerts or integration with IoT platforms.[14]

Chapter 3 Methodology

3.1 Hardware

3.1.1 Hardware Components

In the development of a **secure computerized control system**, the selection of **hardware components** plays a crucial role in ensuring **efficiency, security, and seamless functionality**. Each component is carefully chosen to integrate **mechanical and electronic elements**, enabling a reliable and automated access control system.

By understanding the role and interaction of each component, we gain valuable insight into the **fusion of hardware and software**, making this smart locking mechanism a **technological innovation in modern security solutions**.

3.1.1.1 Main Unit Components

1. Raspberry PI 5 - 8 GB

Figure 3.1.1.1.1 - Raspberry PI 5

The **Raspberry Pi 5** is a **powerful single-board computer (SBC)** that serves as the **brain** of the smart door lock system. Equipped with a **quad-core processor, multiple USB ports, GPIO pins, and built-in Wi-Fi and Bluetooth**, it provides the necessary computing power to process authentication inputs such as **face recognition, fingerprint scanning, and keypad entry**.

In this project, the **Raspberry Pi 5** controls the whole system by sending signals through MQTT channels. When a control message is published, it triggers a **HIGH output**, activating the relay module to unlock the door. It also manages real-time monitoring, data logging, and mobile communication, making it a crucial component for automation and security.

With its **versatility, connectivity, and processing capabilities**, the **Raspberry Pi 5** is an ideal choice for integrating **IoT-based smart security solutions**.

2. Display

Figure 3.1.1.1.2 - MPI3201 Touch Screen for Raspberry PI

This **3.2-inch touch-screen LCD** is a compact **TFT LCD with a 320×240-pixel** resolution designed for **Raspberry Pi**, offering an **interactive interface** for user input and real-time system feedback. It features a **resistive touch screen** with an **XPT2046 controller** and driven by the **ILI9341 IC** via an **SPI interface**, that allows users to **navigate menus**, **enter passwords**, **manage users**, **monitor camera input**, and interact with the smart door lock system without the need for an external keyboard or mouse.

3.1.1.2 ESP8266 Board Components

1. NodeMCU ESP8266

Figure 3.1.1.2.1 – NodeMCU ESP8266

The **NodeMCU ESP8266** is an open-source IoT development board powered by the **ESP8266 Wi-Fi microcontroller**, featuring a **32-bit Tensilica LX106 processor at 80 MHz**, **128 KB RAM**, and **4 MB flash memory**. It offers built-in **802.11 b/g/n Wi-Fi, 17 GPIO pins** supporting **I2C, SPI, UART, and PWM**, and a single **ADC pin**. Programmable via Arduino IDE or Lua, is powered by USB or 5V external input, and is ideal for cost-effective projects like home automation and sensor networks.

In our system the **ESP8266** is used to provide more **modularity** to the system as it communicates with **the Raspberry PI using MQTT channels** to send logs and receive commands, it also is responsible for managing the **Touch Sensor Keypad, Bell Button, Fingerprint Module, PIR Sensor** and controlling the **Solenoid Lock**.

2. 74HC4067 16-Channel Analog Digital Multiplexer Breakout Board

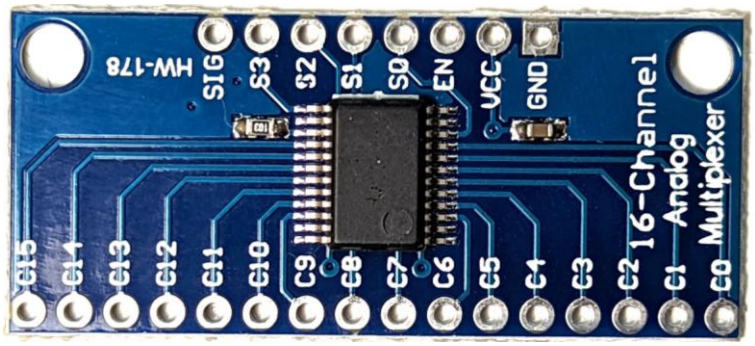


Figure 3.1.1.2.2 - 74HC4067 16-Channel Analog Digital Multiplexer Breakout Board

The **74HC4067** is a **16-Channel Analog/Digital Multiplexer Breakout Board**, based on the **CD74HC4067 IC**, functions as a bidirectional rotary switch, routing **one common pin (SIG)** to any of **16 channel pins (C0-C15)** for **analog or digital signals**. Operating between **2V and 6V**, it is compatible with **3.3V and 5V** systems like **ESP**, **Arduino** and **Raspberry Pi**. **Four address pins (S0-S3)** use **binary addressing** to select channels, allowing up to **16 sensors or devices** to connect to one microcontroller pin. With a low on-resistance of **60-70 ohms** and an **enable pin (EN)** for **channel control**, it's ideal for **sensor arrays, serial communication,** and **IoT prototyping**.

In our system the **74HC4067** is used to **extend** the number of **GPIO pin**s as the ESP8266 has only **10 usable GPIO pins**. By making use of this extension we created a **3x4 functional keypad** using **TP223 touch sensors** and also added **a bell button** for the sake of applying the **secure computerized control system** to create a **smart door lock**.

3. PIR Sensor

Figure 3.1.1.2.3 – SR501 PIR Sensor

The **HC-SR501 PIR** Sensor is a cost-effective motion detection module that senses infrared radiation changes from moving objects, ideal for security, lighting, and IoT applications. It detects motion within a **7-meter range and a 100°** conical angle, operating on **4.5V to 20V (typically 5V)**. The sensor offers single and repeatable trigger modes, adjustable via a jumper, with customizable sensitivity (3-7 meters) and delay (3-300 seconds) through potentiometers.

Featuring three pins (VCC, GND, OUT), it outputs a 3.3V HIGH signal upon motion detection, ensuring compatibility with microcontrollers **like Arduino and ESP8266.**

In our system it provides **motion detection feature** to prevent **excessive use of resources** and **exhaustion of the system**.

3.1.1.3 Power System Components

1. Power Supply

Figure 3.1.1.3.1 - 12V 10A Power Supply

This **12V 10A power supply** is a **high-efficiency power switching module** designed to provide a stable **DC 12V output** with a **10A current capacity**. It is ideal for powering electronic devices requiring a **reliable and continuous power source**.

This power supply is responsible for delivering **consistent 12V power** to components such as the **solenoid lock, Display, Camera, Speakers, Microphones, Fingerprint module, relay modules, Raspberry Pi (via a step-down converter), and all other peripherals**. It ensures that all hardware operates smoothly, supporting reliable security and automation functions.

2. BMS

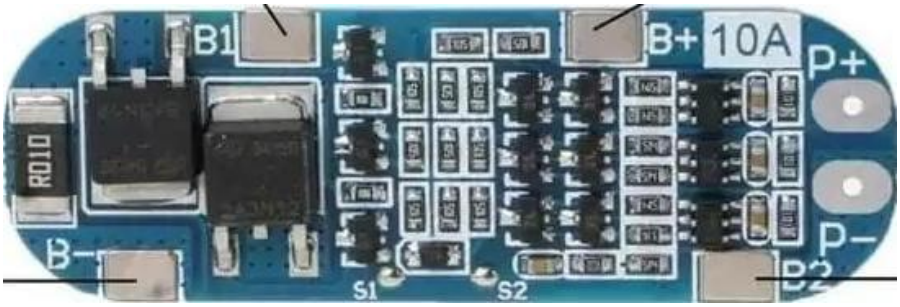


Figure 3.1.1.3.2 - 12.6V 10A 3S BMS

This **Battery Management System (BMS) module** is designed for **3-cell (3S) lithium-ion or lithium-polymer battery packs**, providing efficient **charging and protection**. It ensures safe

and reliable operation by preventing **overcharging, over-discharging, short circuits, and overcurrent issues**.

This BMS module manages the **power supply for backup battery operation**, ensuring that the system remains functional even during **power outages**. It safeguards the **battery pack** from damage, enhancing the reliability and longevity of the entire setup.

3. Li-ion Battery Cells

Figure 3.1.1.3.3 - 18650 5000mAh Li-ion Cell

These **18650 lithium-ion** battery cells is a **high-capacity rechargeable power source** with a **nominal voltage of 3.7V** and a **5000mAh capacity**. It is widely used in **portable electronics, power banks, flashlights, and embedded systems** due to its **high energy density, low self-discharge rate, and long cycle life**.

3 cells of this battery connected in series serve as a **backup power source**, ensuring uninterrupted operation during power outages. Combined with a **Battery Management System (BMS)**, it helps maintain system reliability and efficiency.

4. Relay

Figure 3.1.1.3.4 - 12V 10A SPDT Relay

This **12V 10A Single Pole Double Throw (SPDT) relay** is an electromechanical switching device used for controlling high-power circuits with low-power signals. It consists of a **coil and a set of contacts**, enabling it to toggle between two output states.

Key Features & Usefulness:

- **Voltage & Current Handling:** Operates at **12V DC** and can switch loads up to **10A**, making it suitable for various applications.
- **SPDT Configuration:** Provides both **normally open (NO) and normally closed (NC)** contacts, offering flexible control options.
- **Electrical Isolation:** Allows microcontrollers (such as Raspberry Pi or Arduino) to **safely switch high-power devices** like motors, lights, or door locks.
- **Reliability & Durability:** Commonly used in **automation, security systems, and smart home applications**.

This Relay is used as a switch between the primary power source and the backup emergency battery pack, which ensures that **the door remains operational even during power outages**, enhancing security and reliability.

5 .Step Down Converter

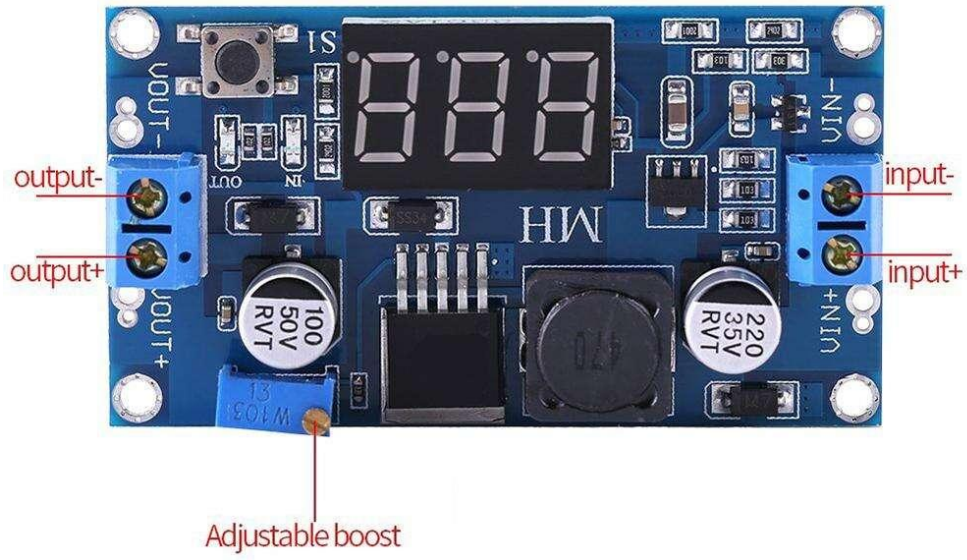


Figure 3.1.1.3.5 - 9-90V to 5V 3A Step Down converter

This **DC step-down module** is a **buck converter** that efficiently reduces an input voltage range of **9V to 90V** down to a stable **3.2V-35V output** with a maximum current of **3A**. It is commonly used in **embedded systems, IoT devices, and battery-powered circuits** to provide a reliable power source for **microcontrollers, sensors, and communication modules**.

In the **secure computerized control system**, this module is essential for **powering the Raspberry Pi 5, ESP8266 Board**. It converts the **12V power supply** to a **stable 5V** required by the Raspberry Pi and other low-voltage components, ensuring smooth and efficient operation.

3.1.1.4 Electric Lock System Components

1. Relay

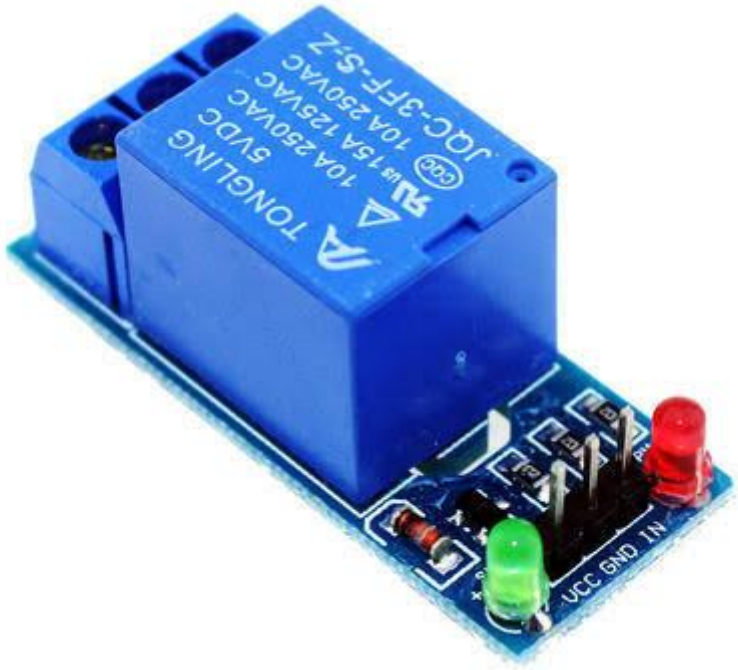


Figure 3.1.1.4.1 - 5V 10A SPDT Relay

This **Single Pole Double Throw (SPDT) relay** operates with a **5V DC coil** and can handle up to **10A of current**, making it suitable for switching high-power electrical devices using low-voltage signals. It features **five pins** for easy through-hole mounting and provides a reliable interface between low-power microcontrollers and high-power circuits.

This relay is used in the **smart door lock system** to **control the solenoid lock**. The ESP8266sends a **LOW or HIGH signal** to activate or deactivate the relay, allowing the **12V power supply** to engage the solenoid lock mechanism, ensuring secure and efficient access control.

2. Solenoid Lock / Electric Mortise Lock

Figure 3.1.1.4.2 - Solenoid Lock

The **Electromagnet Solenoid Lock 10mm 12V 600mA** is a compact locking mechanism designed for electronic access control systems. Operating at 12V DC and drawing 600mA of current, it features a lock pin that extends 10mm when activated. The solenoid is designed for intermittent use, with an unlocking time of approximately 1 second, making it suitable for applications requiring quick and temporary access control.

3.1.1.5 Two Way Audio System Components

1. Sound Card

Figure 3.1.1.5.1 - USB to 3.5mm Sound Card

The USB to 3.5mm Mic and Headphone Jack Stereo Headset Audio Adapter USB Sound Card 7.1. It features a USB interface that splits into dual 3.5mm jacks for stereo headphones and a mono microphone, supporting 7.1 channel audio for immersive sound. The adapter is plug-and-play, requiring no drivers, and is compatible with Linux, Windows, Mac, and other systems.

2. Speakers

Figure 3.1.1.5.2 – 1.5W 8Ohm Speaker

The speaker module is a compact, yet powerful audio output component designed for various embedded applications. In the smart door lock system, this speaker plays a crucial role in enabling real-time audio feedback, such as alarm signals, and communication with visitors. It ensures that users receive clear and loud sound notifications, enhancing the overall interactivity of the system.

This speaker features an impedance of **8Ω** and a **nominal power rating of 1.5W**, providing a balanced output with minimal harmonic distortion. Its **high sensitivity** ensures efficient sound reproduction, making it ideal for use in security systems where clarity and loudness are essential.

The speaker's **well-designed magnet and imported rubber coil** contribute to its durability and sound fidelity. With its capability to deliver **loud and clear audio**, the speaker enhances the smart lock system by providing effective communication between the user and the visitor.

3.1.1.6 Access Components

1. Touch Sensor Modules

Figure 3.1.1.6.1 - Touch Sensor Module

The **TTP223 Capacitive Touch Key Module** is a state-of-the-art capacitive touch sensor designed to replace traditional mechanical buttons with a more responsive and durable alternative. Built around the **TTP223 IC**, this module provides a seamless interface for human interaction through touch-based input. With a sensing range of approximately **5mm**, it detects human touch and switches its output between high and low states, offering configurable modes such as active-low or toggle operation.

This module plays a **crucial role in our smart door lock system**, where multiple TTP223 modules are **combined to form a capacitive keypad**. The keypad serves as an intuitive user interface for entering access codes, offering a modern alternative to physical buttons. Additionally, a single TTP223 module can be used as a **doorbell button**, allowing visitors to trigger a notification with a simple touch.

By leveraging capacitive sensing technology, this solution enhances both the durability and aesthetics of the smart lock system while ensuring smooth and reliable interaction.

2. Camera

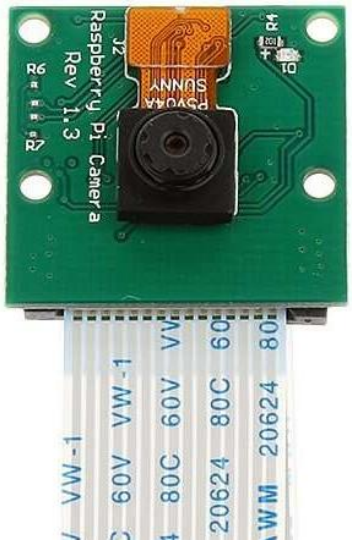


Figure 3.1.1.6.2 - Raspberry PI Camera V1.3

The Raspberry Pi Camera Module V1.3 is a high-quality imaging solution designed for Raspberry Pi boards, utilizing the **OmniVision OV5647** sensor. This 5-megapixel fixed-focus camera module can capture still images at a maximum resolution of 2592 × 1944 pixels and record high-definition video in 1080p at 30 frames per second.

Technical Specifications:

- **Image Sensor:** OmniVision OV5647, 1/4-inch format
- **Still Image Resolution:** 5 MP (2592 × 1944 pixels)
- **Video Recording:**
  - 1080p @ 30 fps
  - 720p @ 60 fps
  - 640 × 480 @ 60/90 fps
- **Interface:** 15-pin MIPI Camera Serial Interface (CSI)
- **Compatibility:** Supports Raspberry Pi Model A and Model B boards
- **Lens Type:** Fixed-focus lens
- **Data Transmission:** High-speed pixel data transfer via the CSI bus

The module connects to the Raspberry Pi via a **15-pin ribbon** cable, exclusively transmitting pixel data to the **BCM2835** processor. It is optimized for use with the Raspbian operating system, making it ideal for applications such as surveillance, facial recognition, and object detection.

2. Fingerprint Module

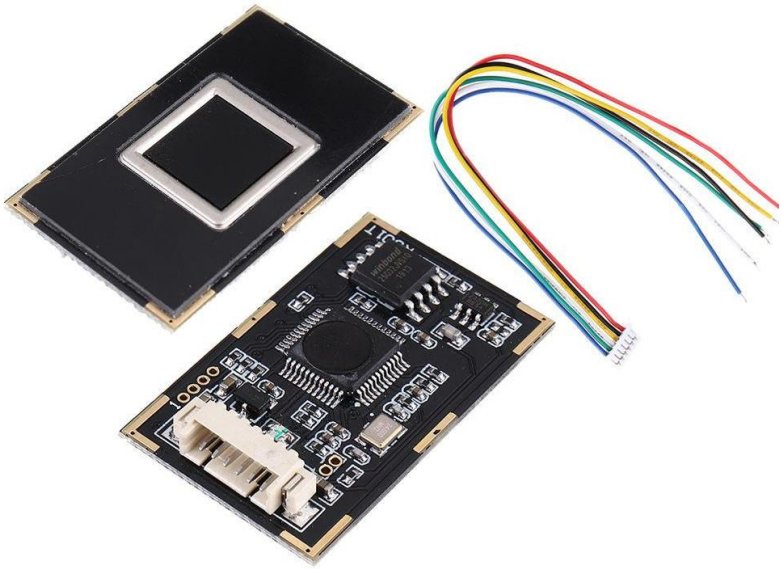


Figure 3.1.1.6.3 - R301T Fingerprint Module

The **R301T fingerprint module** is an optical fingerprint verification device with a **TTL interface**, designed to provide a high-security biometric access solution. With a **storage capacity of up to 3000 Image**, it ensures seamless authentication, making it ideal for access control systems, including smart locks, security doors, and digital safes.

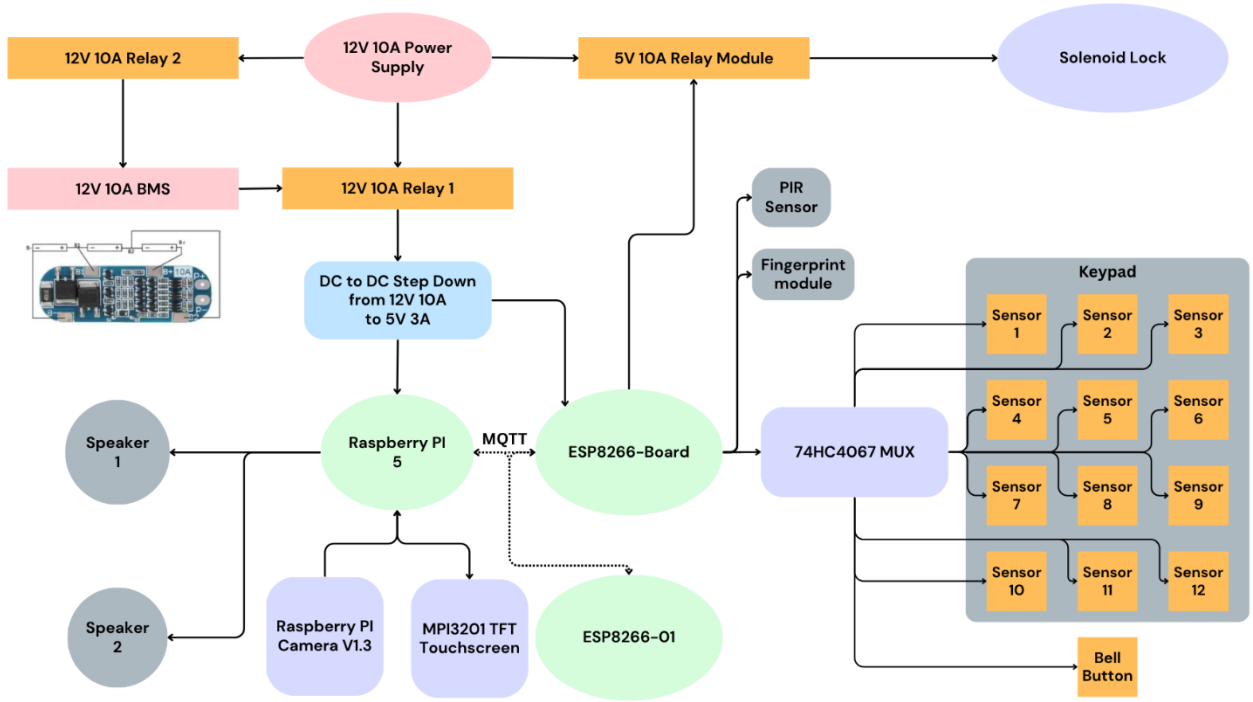
This module integrates **optical fingerprint recognition** with a **high-speed UART serial interface (57600 bps)** for efficient communication with microcontrollers like **Raspberry Pi, Arduino, and ESP8266**. The fingerprint sensor captures and processes images at **<50ms search time**, ensuring near-instant verification with **<0.3s response time**.

In our **secure computerized control system**, this module will serve as one of the primary access methods. The user's fingerprint is scanned and matched against stored templates, allowing the system to verify identity before granting access. The process follows these steps:

1. **Fingerprint Enrollment:** The user places a finger on the sensor, and the module captures the fingerprint image, extracts unique features, and stores it as a template.
2. **Authentication:** When the user attempts access, the fingerprint is scanned again and compared against stored templates.
3. **Verification Decision:** If a match is found, the system sends a signal to unlock the door. If authentication fails, access is denied.

The **R301T ensures reliability** in various conditions, including **wet, dry, and aged fingerprints**, making it a robust choice for secure, touch-based authentication in our **IoT-enabled smart lock system**.

3.1.2 Hardware Circuits



3.1.2.1 Power System Circuit

1. Requirements:

Design the **power circuit** with a rechargeable **12 V lithium batteries** for a system that uses **Raspberry PI 5**. The **220 V-power source** supplies the **Raspberry PI** with power and charges the batteries. When the power is **cut off**, the batteries supply the system instead.

2. Circuit Connection:

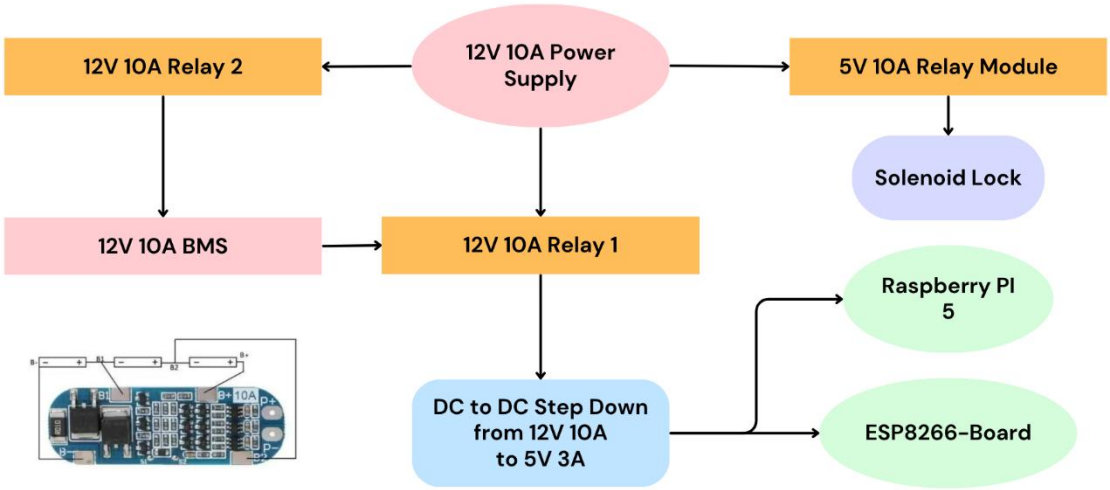


Figure 3.1.2.1.1 - Power System Block Diagram

3. Simulation:

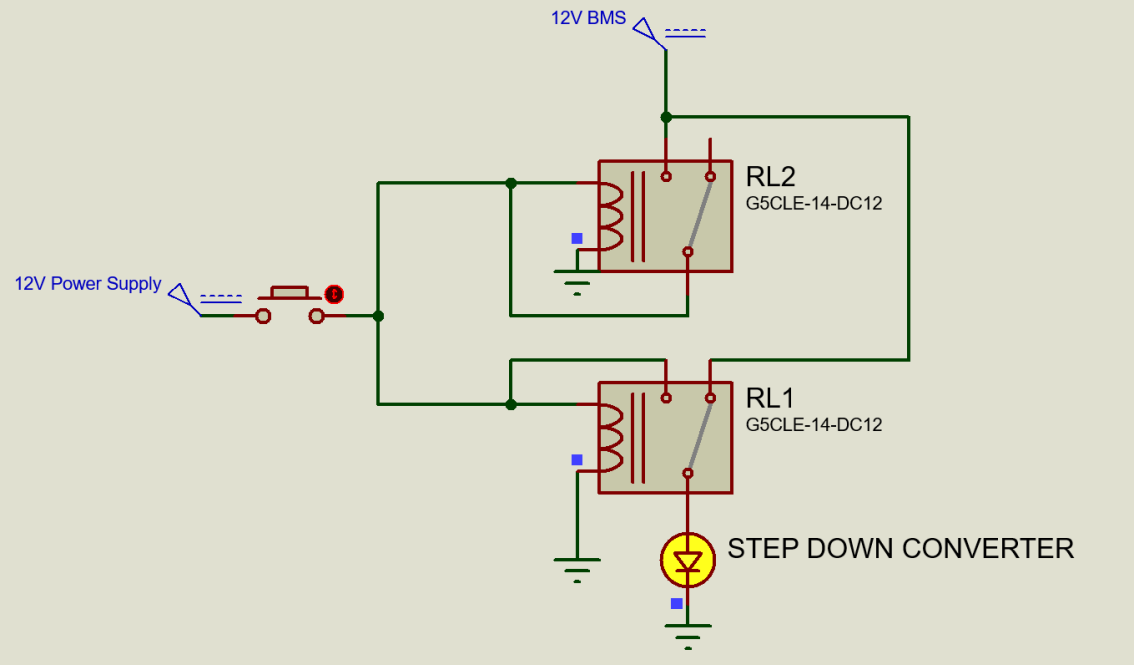


Figure 3.1.2.1.2 - Power System Circuit Connection

4. Live Photo:

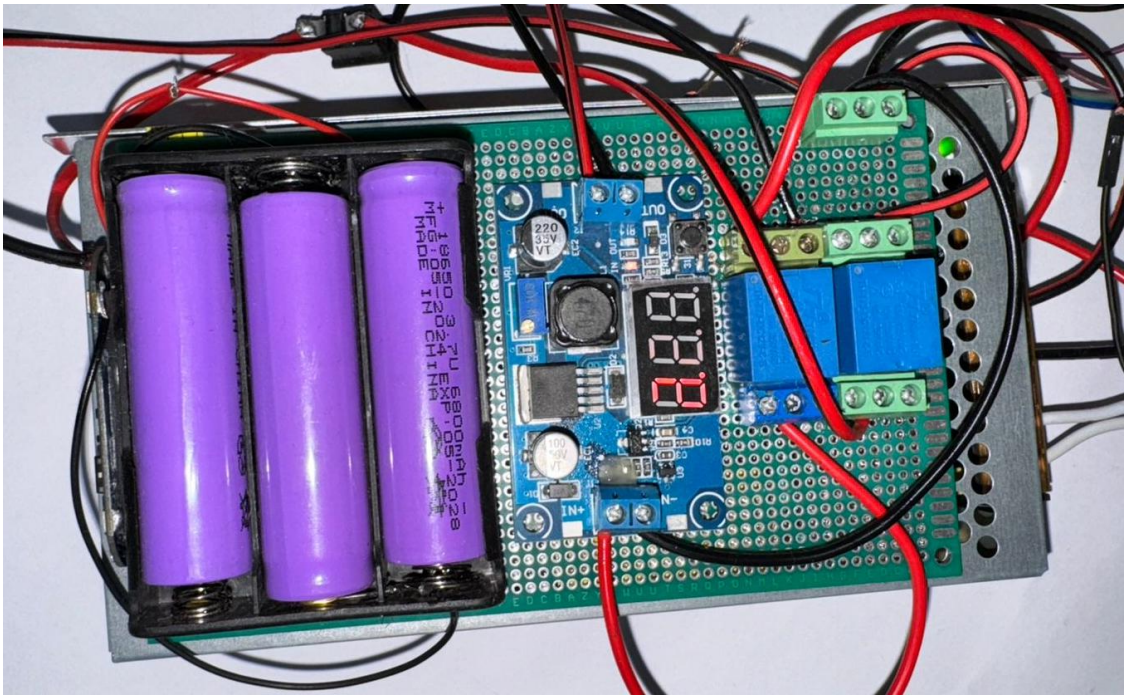


Figure 3.1.2.1.3 - Power System Live Photo

5. Breakdown:

The circuit diagram illustrates the structured power distribution for the smart door lock system. It ensures efficient power management and safe operation of all components involved in securing and automating door access.

1. **Power Source (12V 10A Power Supply):**
   The system is powered by a **12V 10A power supply**, which provides sufficient energy for all connected components.

2. **Battery Management System (12V 10A BMS):**
   A **Battery Management System (BMS)** is integrated to regulate power delivery, protect against overcharging, and ensure stable operation. This safeguards the connected hardware, extending its lifespan.

3. **Relay Module 1:**
   The **relay** acts as a switching mechanism, enabling controlled power distribution to different components, such as the locking mechanism or additional security modules.

4. **Voltage Step-Down Converter (DC to DC Step Down Module: 12V to 5V, 3A):**
   Since the Raspberry Pi 5 operates on **5V**, a **DC-DC step-down converter** is used to efficiently convert 12V power to a stable **5V 3A** output. This ensures proper functionality without overloading the Raspberry Pi.
5. **Raspberry Pi 5 (Core Controller):**
   The **Raspberry Pi 5** serves as the central control unit, managing authentication, facial recognition, fingerprint scanning, and keypad access while communicating with the mobile application for remote operation.

This structured power design ensures **reliable, efficient, and safe** operation of the smart door lock system, providing seamless integration of multiple security layers.

6. Causes For Success:

1. **Stable Voltage Conversion (12V → 5V)**
   - **The Raspberry Pi 5 requires a steady 5V DC supply.**
   - Since the **battery pack and adapter provide 12V**, a **DC-DC step-down (buck) converter** is necessary.
   - The **DC-DC converter must be capable of supplying at least 3A** to handle the Pi's peak power demands.
   - **Success Factor:** If the **buck converter delivers a stable 5V, the Pi will run smoothly** without undervoltage issues.
2. **Proper Relay Switching Between Power Sources**
   - **The relay ensures automatic switching** between the **12V adapter (primary power)** and **battery (backup power)**.
   - If **AC power** is available, the relay **connects the adapter** to the system.

- If AC power is lost, the relay **automatically switches to battery power** without delay.
- **Success Factor:** If the relay is correctly wired and fast enough, **the Pi won't experience power interruptions**.

3. Efficient Battery Management (BMS)

- The **BMS ensures safe charging, discharging, and balancing** of the **3S battery pack**.
- **Prevents over-discharge** (which could cause low voltage, shutting down the Raspberry Pi).
- **Success Factor:** A **well-functioning BMS** keeps the batteries healthy and ensures the Pi gets uninterrupted power.

4. Sufficient Battery Capacity & Current Supply

- The Raspberry Pi 5 **typically consumes 3A at 5V** (15W).
- The **18650-battery pack (3S, 6 cells total) must store enough energy** to run the Pi for an extended period during a power outage.
- **Success Factor:** If the battery **has enough capacity** (e.g. **6 x 2800mAh cells**) and **the BMS allows sufficient discharge current**, the Pi can run on battery power without issues.

5. Low-Resistance Connections & Proper Wiring

- **Thick enough wires** should be used for power connections to avoid voltage drops.
- **Secure soldering or connectors** prevent power interruptions.
- **Proper grounding** eliminates electrical noise that could cause instability.
- **Success Factor:** Well-made connections **ensure stable voltage delivery** to the Raspberry Pi.

3.1.2.2 Main RPI Unit

1. Circuit Connection:

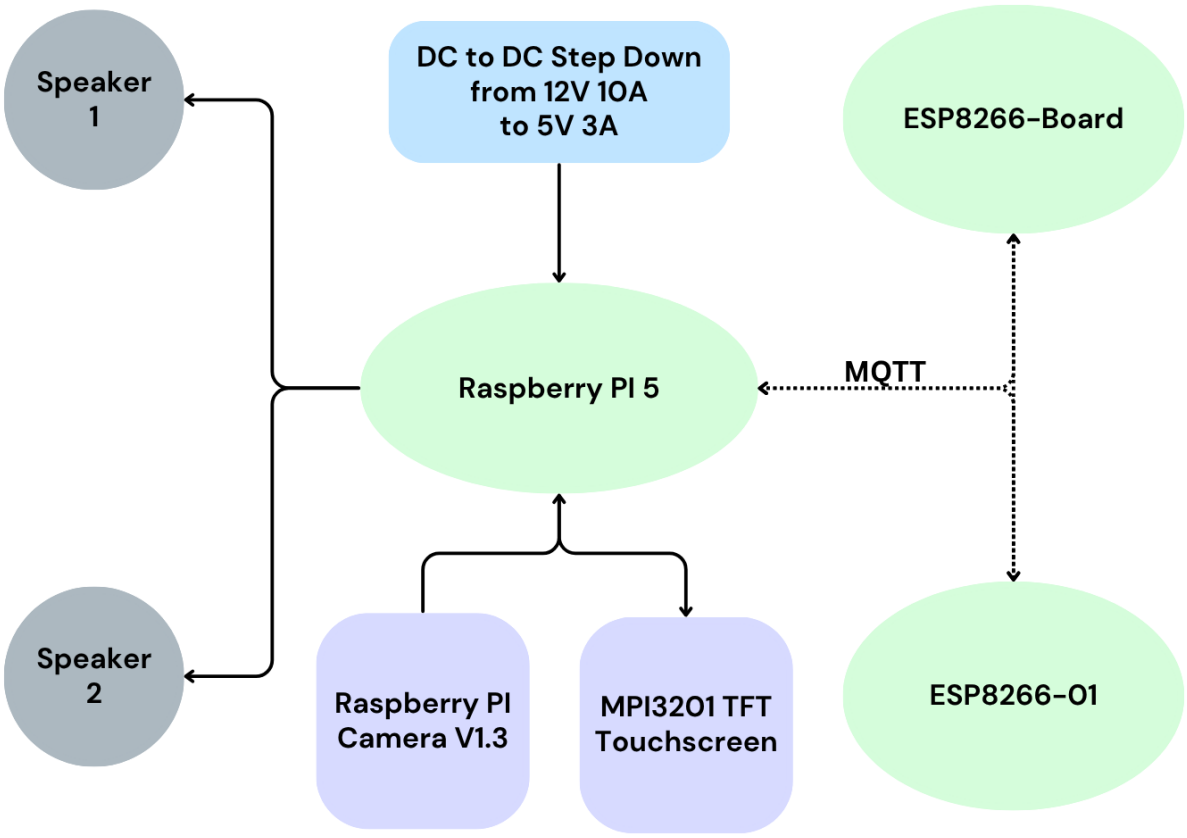


Figure 3.1.2.2.1 - Main Unit Block Diagram

2. Live Photos

Figure 3.1.2.2.2 - Main Unit Live Photo 1

Figure 3.1.2.2.2 - Main Unit Live Photo 2

3. System Requirements

The Raspberry Pi, Display, and Speaker system is designed to provide an interactive visual interface and an audible alert mechanism for a smart application, specifically triggering a doorbell sound when activated. Below are the specific hardware and software requirements to implement this subsystem.

4. Hardware Requirements

- **Raspberry Pi 5:** Serves as the central processing unit, running the Python-based application and hosting the MQTT broker.
- **Display**: A **3.2-inch touch display** to display **GUI** .
- **Speaker**: Provides an **audible doorbell ring** when the bell button is pressed, connected via **USB through a 3.5mm audio jack sound card**.
- **Power Supply**: **A 5V 3A USB-C** power adapter for the **Raspberry Pi**,.

5. Software Requirements

- **Raspberry Pi OS**: The latest Debian-based operating system for the Raspberry Pi, providing a stable environment for Python and system tools.
- **Python 3**: With version 3.7 or later, including the following libraries:
  - **paho-mqtt** for **MQTT communication**.
- **Mosquitto MQTT Broker**: Installed and configured on the Raspberry Pi to handle message passing.

6. Environmental Requirements

- **Stable Power Source**: Ensures uninterrupted operation of the Raspberry Pi, display, and speaker.
- **Controlled Temperature**: The **Raspberry Pi** should operate within **0°C to 50°C** to prevent overheating.
- **Acoustic Environment**: A setting where the doorbell sound can be clearly heard, avoiding excessive background noise.

7. Breakdown

The Raspberry Pi, Display, and Speaker system is a cohesive unit designed for visual status updates and audible alerts. Below is a detailed breakdown of its components:

1. **Raspberry Pi**
   - **Role**: Acts as the central hub, executing the Python script to manage MQTT communication, status updates, and doorbell triggering.
2. **Display**
   - **Purpose**: Provides a visual interface to display status information and alerts.
   - **Connection**: Interfaces via **SPI** with a resolution of **320x240 pixels**.
   - **Functionality**: Displays **GUI** and allows controls through **touch gestures**.
3. **Speaker**

- **Model**: **Generic 1.5W 8Ω speaker** for audible output.
- **Connection**: Linked to the **Sound Card 3.5mm jack**, amplified to produce a clear doorbell sound.
- **Role**: Emits a ringing sound when the **bell button is pressed**, serving as an audible alert.

8. How It Works

The Raspberry Pi, Display, and Speaker system operates as an interactive platform for status monitoring and doorbell alerts. Here's a step-by-step explanation of its operation:

1. **System Initialization**:
   - On boot, the Raspberry Pi loads Raspberry Pi OS and starts the **Mosquitto MQTT broker**.
   - The speaker is configured to produce a doorbell sound when triggered via MQTT.

2. **Doorbell Triggering**:
   - The system subscribes to the **door/bell MQTT topic**, which receives **"Bell pressed"** messages from an external device (e.g., an ESP8266).
   - Upon receiving a **"Bell pressed"** message, the Python script triggers the speaker to play a pre-defined doorbell sound (e.g., a WAV file or tone generated via GPIO).
   - The sound is amplified by **the Sound Card**, ensuring it is audible in the environment.

3. **MQTT Integration**:
   - The Python script connects to the local **Mosquitto broker** (localhost, port 1883) and subscribes to the door/bell topic.
   - When a **"Bell pressed"** message is received, it logs the event (e.g., to **/home/pi/Blynk/logs/door_logs.txt**) and triggers the speaker.
   - Status updates and logs can be forwarded to other devices via MQTT topics like **smartlock/status** and **smartlock/logs**.

4. **Operation Flow**:
   - When the bell button is pressed on the external device, it publishes **"Bell pressed"** to **door/bell**.
   - The Raspberry Pi receives this message, activates the speaker to ring, and updates the display with the event timestamp.
   - The alert continues until the sound file or tone completes (e.g., a 2-second ring), then resets to await the next trigger.

3.1.2.3 ESP8266 Board Circuit

1. Circuit Connection:

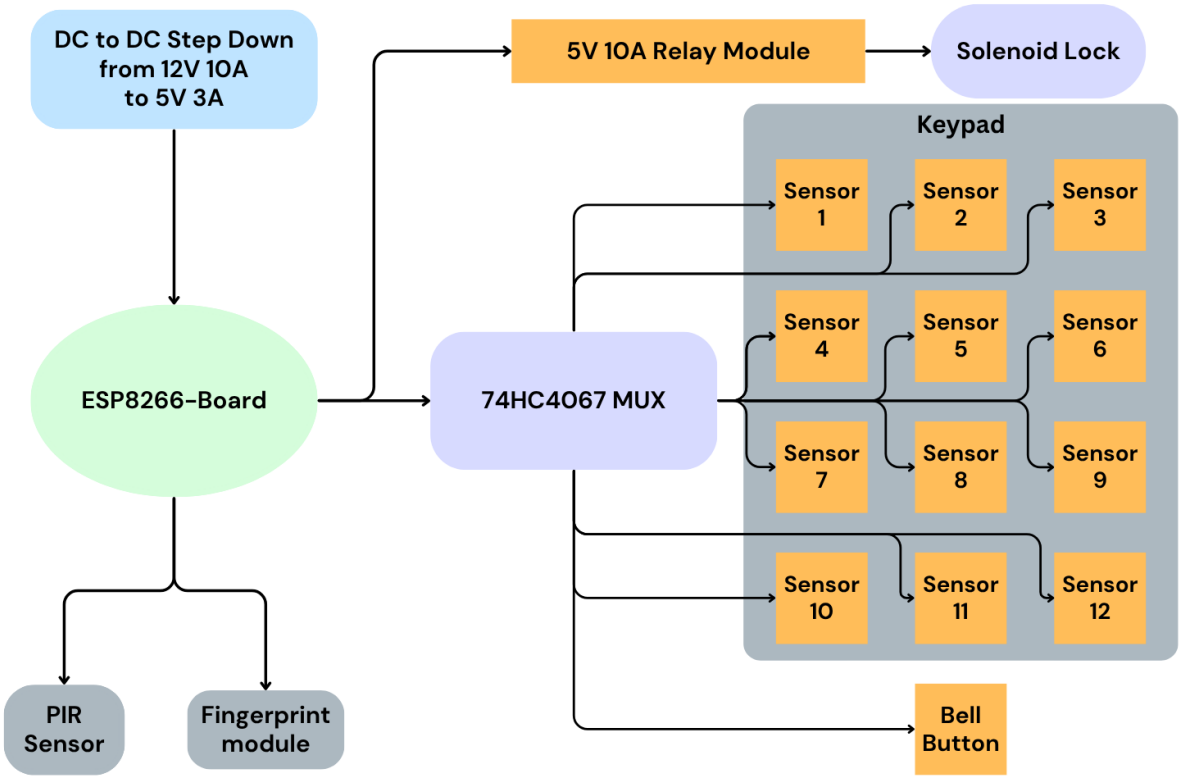


Figure 3.1.2.3.1 - ESP8266 Board Block Diagram

2. Live Photo

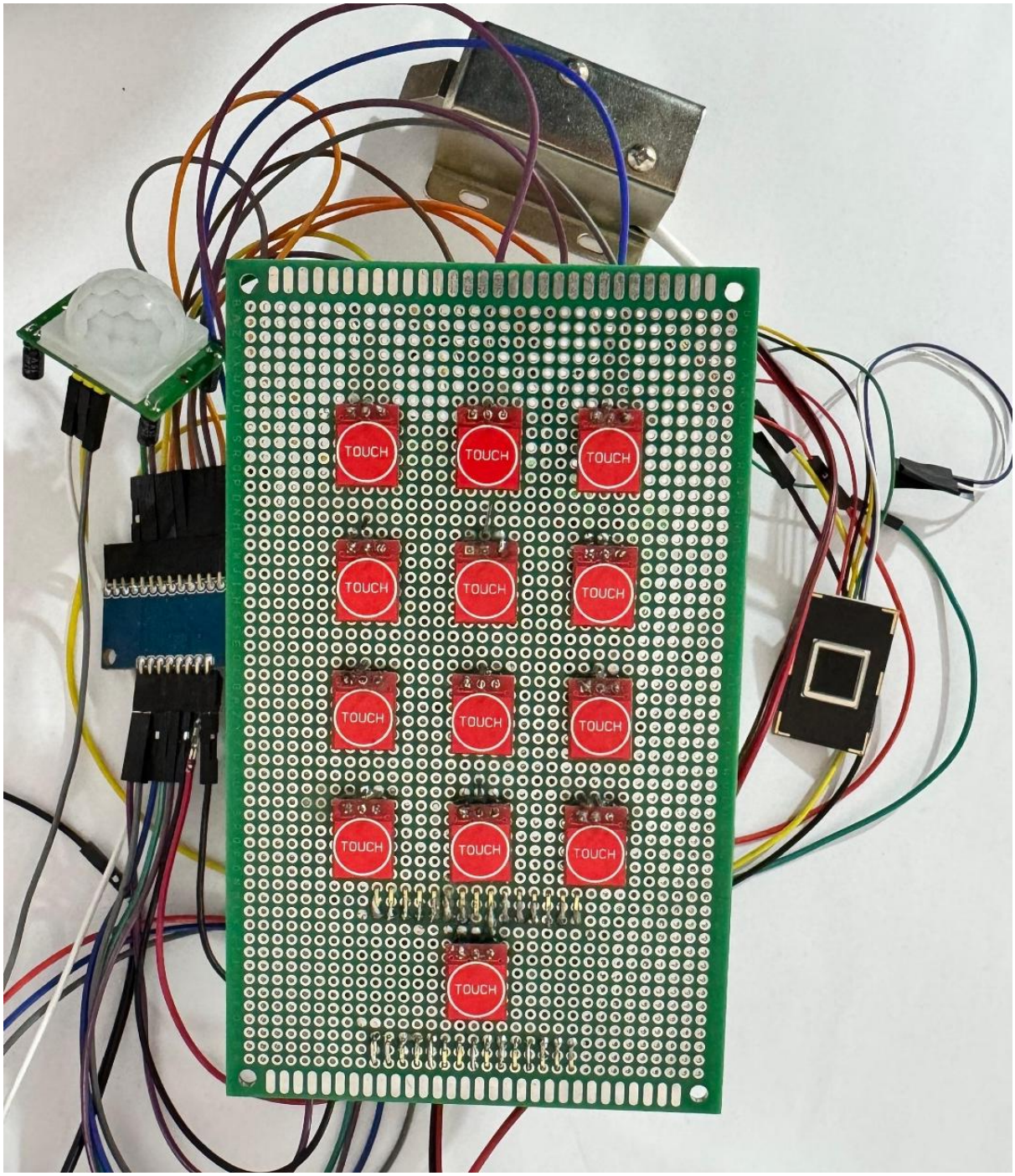


Figure 3.1.2.3.2 - ESP8266 Board Live Photo 1

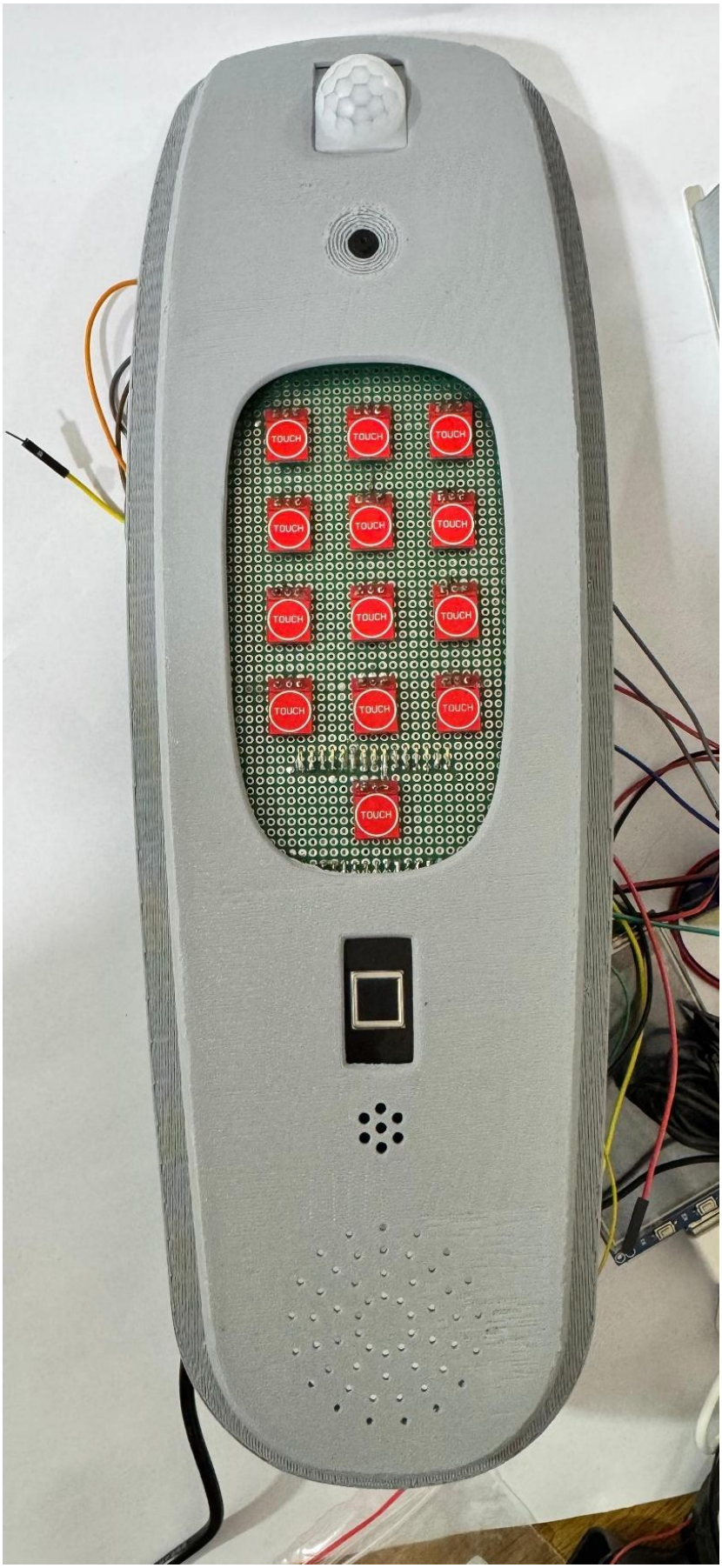


Figure 3.1.2.3.3 - ESP8266 Board Live Photo 2

3. System Requirements

Using the ESP8266 as a microcontroller to design a system to provide secure access control with remote monitoring capabilities. The following are the hardware and software requirements to implement this system:

4. Hardware Requirements

1. **ESP8266 Microcontroller:** Acts as the central controller, providing Wi-Fi connectivity and GPIO pins for interfacing with peripherals.
2. **12V Solenoid Lock:** A fail-secure solenoid that unlocks when powered, controlled via a relay.
3. **5V Active-High Relay Module:** Interfaces the ESP8266 (3.3V logic) with the 12V solenoid, activating when the input is HIGH.
4. **Fingerprint Sensor** (e.g., R307): For biometric authentication, connected via SoftwareSerial.
5. **74HC4067 16-Channel Analog Multiplexer:** Expands input capabilities to support a keypad (channels C0–C11) and a bell button (channel C12).
6. **PIR Motion Sensor:** Detects motion near the door for security monitoring.
7. **Push Button (Bell Button):** Connected via the multiplexer to trigger a doorbell event.
8. **Power Supply:**
    - **12V DC** supply for the **solenoid**.
    - **5V DC** supply for the **relay and ESP8266** (via a buck converter from 12V).

5. Software Requirements

The Following Libraries:

1. **ESP8266WiFi (built-in)**
2. **PubSubClient** for MQTT communication
3. **Adafruit_Fingerprint** for the fingerprint sensor
4. **SoftwareSerial** for serial communication with the fingerprint sensor

6. Breakdown

The system is a modular design with interconnected hardware and software components working together to secure and monitor a door. Below is a breakdown of the key components and their roles:

1. **ESP8266 Microcontroller**
    - **Role:** Central controller managing all peripherals, Wi-Fi connectivity, and MQTT communication.
    - **Pin Usage:**
        - **GPIO14 (D5):** Controls the solenoid lock via an active-high relay (LOCK_PIN).
        - **GPIO13 (D7) and GPIO15 (D8)**: SoftwareSerial for the fingerprint sensor (RX, TX).

- **GPIO16 (D0), GPIO5 (D1), GPIO4 (D2), GPIO0 (D3)**: Control pins **(S0–S3)** for the **74HC4067 multiplexer**.
- **A0 (ADC0, pin 17)**: Analog input **(signalPin)** for the multiplexer.
- **GPIO12 (D6)**: Input for the PIR motion sensor (PIR_PIN).

2. **Solenoid Lock and Relay Module**
   - **Solenoid Lock**: A 12V fail-secure solenoid that unlocks when powered.
   - **Relay Module**: A 5V active-high relay that activates (closes the circuit to power the solenoid) when **GPIO14 is HIGH**. It interfaces the **ESP8266's 3.3V logic** with the **12V solenoid.**
3. **Fingerprint Sensor**
   - **Model:** R301T, supporting up to 20 fingerprints.
   - **Connection:** Uses SoftwareSerial on **GPIO13 (RX) and GPIO15 (TX)** at **57600 baud**.
   - **Functionality:** Allows enrollment, deletion, and matching of fingerprints, with events published to **door/fingerprint**.
4. **74HC4067 Multiplexer**
   - **Purpose:** Expands the **ESP8266's limited GPIO pins** to handle multiple inputs.
   - **Channels:**
     - **C0–C11:** Keypad inputs, scanned to detect key presses (published to **door/keypad**).
     - **C12:** Bell button input, triggering a **"Bell pressed"** event on **door/bell** when **analogRead > 130**.
     - **Control: S0–S3** pins set the channel to read via binary selection.
5. **PIR Motion Sensor**
   - **Function:** Detects motion near the door, publishing **"Motion detected!"** or **"No motion."** to **door/motion**.
   - **Connection:** Connected to **GPIO12**, checked every **500ms** using millis() for timing.
6. **MQTT Communication**
   - **Broker:** Hosted on a Raspberry Pi
   - **Topics**:
     - **door/cmd:** Subscribed for commands like **unlock, toggle, enroll/<id>, delete/<id>**.
     - **door/lock:** Publishes lock status **("Unlocked", "Locked")**.
     - **door/fingerprint:** Publishes fingerprint events.
     - **door/keypad:** Publishes key presses (e.g., "1" for C0).
     - **door/bell:** Publishes bell button events **("Bell pressed")**.
     - **door/motion:** Publishes motion sensor status.
     - **door/status:** Publishes ESP8266 connection status **("ESP connected")**.

7. How It Works

The system operates as an integrated IoT solution for door access control and monitoring. Here's a step-by-step explanation of its operation:

1. **System Initialization:**
    - **On boot**, the **ESP8266** initializes its pins: **GPIO14 (solenoid)** is set LOW to keep the lock secure, **GPIO12 (PIR)** is set as input, and the multiplexer pins **(GPIO16, GPIO5, GPIO4, GPIO0)** are set as outputs.
    - **The fingerprint sensor** initializes via **SoftwareSerial** at **57600 baud,** verifying its presence.
    - The **ESP8266** attempts to connect to one of the available **Wi-Fi networks**, selecting the corresponding **MQTT server (e.g., "192.168.1.12").**
    - It connects to the **MQTT broker**, subscribes to **door/cmd**, and publishes **"ESP connected" to door/status.**
2. **In a Main Loop:**
    - **Fingerprint Authentication:** The **ESP8266** continuously scans for fingerprints using **getFingerprintID().** If a match is found, it publishes **"Fingerprint ID: #X"** to **door/fingerprint**. Fingerprint enrollment or deletion is triggered **via MQTT commands (e.g., enroll/5, delete/5).**
    - **Keypad Input:** The **74HC4067** multiplexer **scans channels C0–C11. If analogRead > 130** on any channel, the key number **(e.g., "1" for C0)** is published to door/keypad.
    - **Bell Button: Channel C12** is scanned for the bell button. If triggered, **"Bell pressed"** is published to **door/bell**, forwarded by the Raspberry Pi to **smartlock/bell**, triggering a **bell_pressed** event in Blynk.
    - **Motion Detection:** The **PIR** sensor on **GPIO12** is checked every **500ms**. Motion events are published to **door/motion ("Motion detected!" or "No motion.")**.
3. **MQTT Reconnection**: If disconnected, the **ESP8266** attempts to reconnect to the MQTT broker every **10 seconds.**
4. **Lock Control**:
    - **Unlock Command:** When an unlock command is received on **door/cmd** (e.g., from Blynk), **unlockDoor()** sets **GPIO14 HIGH for 5 seconds**, unlocking the solenoid, then **LOW** to lock it. Status updates **("Unlocked", "Locked")** are published to **door/lock** and forwarded to **smartlock/status** for Blynk.
    - **Toggle Command:** The toggle command switches the lock state **(HIGH/LOW)** using **Togglelock()**, updating lockstatus and publishing **"Toggled High" or "Toggled Low"** to **door/lock.**
5. **Logging**:
    - The Raspberry Pi logs all events (**lock status, bell presses, motion, fingerprin**t) to **/home/pi/Blynk/logs/door_logs.txt** and forwards them to **smartlock/logs** for display in Blynk's Terminal widget.

3.1.2.4 Solenoid Lock Circuit

1. Requirements:

Design a **swift lock system** that operates both **mechanically and electrically** by utilizing the **theory of a magnetic solenoid lock**. This system functions based on input signals received from the **ESP's GPIO pin**, which can either be **HIGH or LOW**. When the Raspberry Pi sends a **HIGH signal**, the solenoid lock is **activated**, generating a **magnetic field** that pulls the locking mechanism, allowing the door to **unlock**. Conversely, when the Raspberry Pi outputs a **LOW signal**, the magnetic field is **disengaged**, causing the lock to return to its original position, thereby **securing the door**.

2. Circuit Connection:

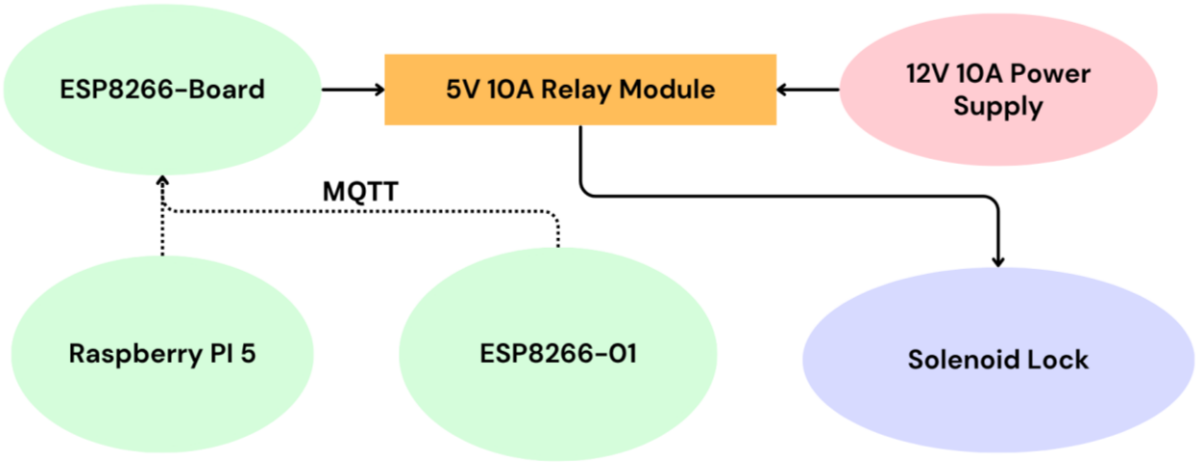


Figure 3.1.2.4.1 - Electric Lock Block Diagram

3. Simulation:

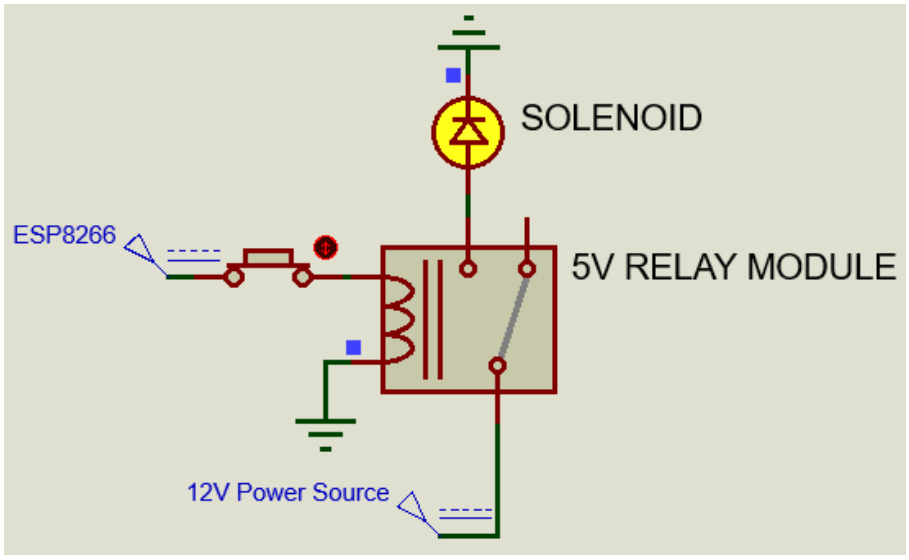


Figure 3.1.2.4.2 - Electric Lock Circuit Connection

4. Breakdown:

This circuit diagram represents the **core mechanism** behind the smart door lock system, ensuring secure and efficient access control through **ESP8266** automation.

1. **ESP8266 (Main Controller):**
   The **ESP8266** serves as the intelligent processing unit that handles authentication inputs such as facial recognition, fingerprint verification, or keypad entry. Based on authentication results, it sends control signals to the relay.

2. **Relay Module (Electronic Switch):**
   Acting as a bridge between the **ESP8266** and the high-power components, the **relay** enables safe switching of the **solenoid lock** using low-power signals from ESP8266. This prevents direct exposure of the microcontroller to high current.

3. **12V 10A Power Supply (Primary Energy Source):**
   The **12V 10A power supply** delivers sufficient electrical power to drive the solenoid lock while ensuring reliable operation.

4. **Solenoid Lock (Door Locking Mechanism):**
   The **solenoid lock** is the physical locking device that controls access to the door. When activated by the relay, it unlocks the door, allowing entry. Once deactivated, the lock secures the door again.

5. How It Works:

When a user attempts to unlock the door, the **ESP8266** processes authentication. If access is granted, it **triggers the relay**, allowing **12V power** to flow to the **solenoid lock**, unlocking the door. After a set duration, the relay deactivates, locking the door again.

This **automated and secure circuit design** integrates IoT technology with smart security, offering a **seamless, modern, and efficient** access control solution.

3.2 Intelligent Face Authentication System

3.2.1 AI Steps:

1. **Acquiring Image**:
   - Capturing an image using a camera or loading an image from a dataset.
   - The input image contains one or more faces.
2. **Face Detection**:
   - Detecting and locating the face(s) within the image.
   - A face detection algorithm (e.g., MTCNN, Haar cascades, or Res 10) identifies facial regions.
3. **Face Recognition**:
   - Extracting facial features and comparing them to a trained model.
   - Algorithms like LBPH, FaceNet, ArcFace …. and so on
4. **Person Identity**:
   - Matching the detected face with known individuals in the dataset.
   - If a match is found, the person's identity is confirmed; otherwise, the face is classified as unknown.

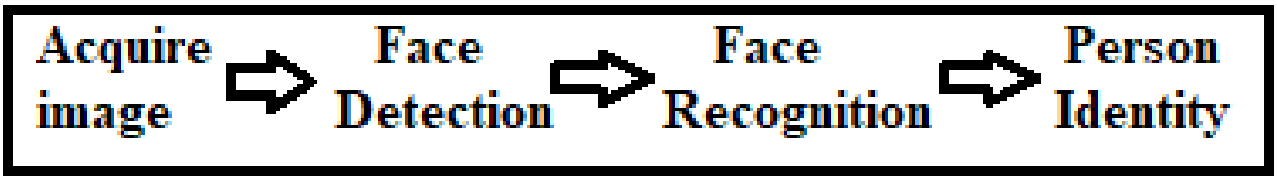


Figure 3.2.1 – AI Steps

3.2.2 Dataset Creation:

- **Overview**

The dataset creation process is a fundamental step in face recognition, as it involves collecting and preparing images of individuals for training the model. This process ensures that the system has enough data to learn and recognize faces accurately.

1. **Data Collection:**

- The system captures face images using a webcam or an existing image dataset.
- Users are prompted to enter their **User ID or Name** before capturing images.
- A dataset is created for each user, stored in a **structured directory format**.

✓ **Purpose:** Organizes face images into labeled directories for training.

2. **Face Detection & Cropping:**

- The **Res10 DNN** algorithm is used to detect faces in the captured image.
- Each detected face is **cropped** to focus only on the facial region.
- Images without a detected face are discarded.

✓ **Purpose:** Ensures only relevant face images are stored for recognition.

3. **Image Preprocessing**

- The cropped face images are **resized to a fixed size (200x200 pixels)** for consistency.
- Images are **RGB** which is suitable for Deep learning Algorithms.

✓ **Purpose:** Standardizes input images for training the face recognition model.

4. **Storing Images in Dataset**

- The processed images are **saved in the corresponding user's directory** with a unique filename.
- A counter is used to track the number of images collected per user (up to 200).

✓ **Purpose:** Creates an organized dataset that maps images to user identities.

5. **Completion & User Feedback**

- The dataset collection process runs until 200 images are collected or the user presses **Enter** to stop.
- The system provides feedback on the total number of images collected.

✓ **Purpose:** Ensures an adequate number of images are collected for training.

- **Structure of Training Dataset:**

/dataset

/1 → (image.1.1.jpg, image.1.2.jpg) # User with ID 1

/2 → (image.2.1.jpg, image.2.2.jpg) # User with ID 2

ex) image.UID.imgID.jpg

Figure 3.2.1.2 - Structure of Training Dataset

- **Testing Data:**

Structured similarly to the training dataset but with the addition of the unauthorized (-1) folder.

/dataset

/1 → (image.1.1.jpg, image.1.2.jpg) # User with ID 1

/2 → (image.2.1.jpg, image.2.2.jpg) # User with ID 2

/-1 → (unknown1.jpg, unknown2.jpg) # Unauthorized Users

Figure 3.2.1.3 – Structure of Testing Dataset

I. Face Detection:

- **Overview**

The face detection step is a critical component of the face recognition pipeline. It involves identifying and localizing faces in an image or video frame. This ensures that the model focuses only on the relevant region (the face) for further processing, such as feature extraction and recognition. The selected face detection model is Res10 SSD (Single Shot MultiBox Detector) with a Caffe backend, which offers high-speed, real-time performance and sufficient accuracy for frontal face detection tasks.

- **Face Detection Workflow Using Res10**

1. **Face Detection Process**
   - **Input**: A single image or video frame.
   - **Output**: A bounding box for each detected face.
2. **Image Preprocessing**
   - **Resizing**: The input image is resized to 300x300 pixels, as required by the Res10 model, while maintaining the aspect ratio.
   - **Mean Subtraction**: The model applies mean subtraction with specific values (typically [104.0, 177.0, 123.0]) to normalize the input image channels.
   - **Blob Conversion**: The preprocessed image is converted into a blob format suitable for the DNN module.
3. **Outputs of Res10 SSD**
   - **Bounding Box**: Coordinates of the detected face region (x, y, width, height), often filtered using a confidence threshold (e.g., 0.5) to exclude weak detections.
   - **Confidence Score**: A probability score indicating how likely the detected region contains a face.
   - **No Facial Landmarks**: Unlike MTCNN, Res10 SSD does not provide facial landmarks, so any alignment or landmark-based processing must be handled by an additional module if needed.

II. Face Recognition:

➢ Overview

Face recognition is the process of identifying or verifying the identity of a person from a detected face. In the context of this model, the **ArcFace** algorithm is used for face recognition. **ArcFace** is a state-of-the-art deep learning-based face recognition model that projects facial features onto a hypersphere and applies additive angular margin loss to enhance the discriminative power of embeddings. It is known for its **high accuracy and robustness** in real-time face recognition tasks.

1. **Face Recognition Workflow Using ArcFace:**

1. Face Detection & Preprocessing:

• The input frame is passed through a face detector (Res10 SSD in our case) to identify and crop the face region.
• The cropped face is resized and normalized to meet ArcFace input requirements.

2. Feature Extraction:

• The preprocessed face is passed through the ArcFace model, which extracts a 512-dimensional embedding vector.
• These embeddings are high-quality representations that capture the unique facial features.

3. Matching & Recognition:

• The cosine distance is calculated between the real-time embedding and pre-saved user embeddings.
• A threshold (e.g., 0.30–0.32) determines if the face belongs to a known user or is unknown.

4. Outlier Removal and Buffer Smoothing:

• To ensure accuracy, outlier embeddings are removed from the training set using standard deviation-based filtering.
• A frame buffer collects predictions over several frames (e.g., 5), and the most frequent label is selected for stability.

III. Flowchart:

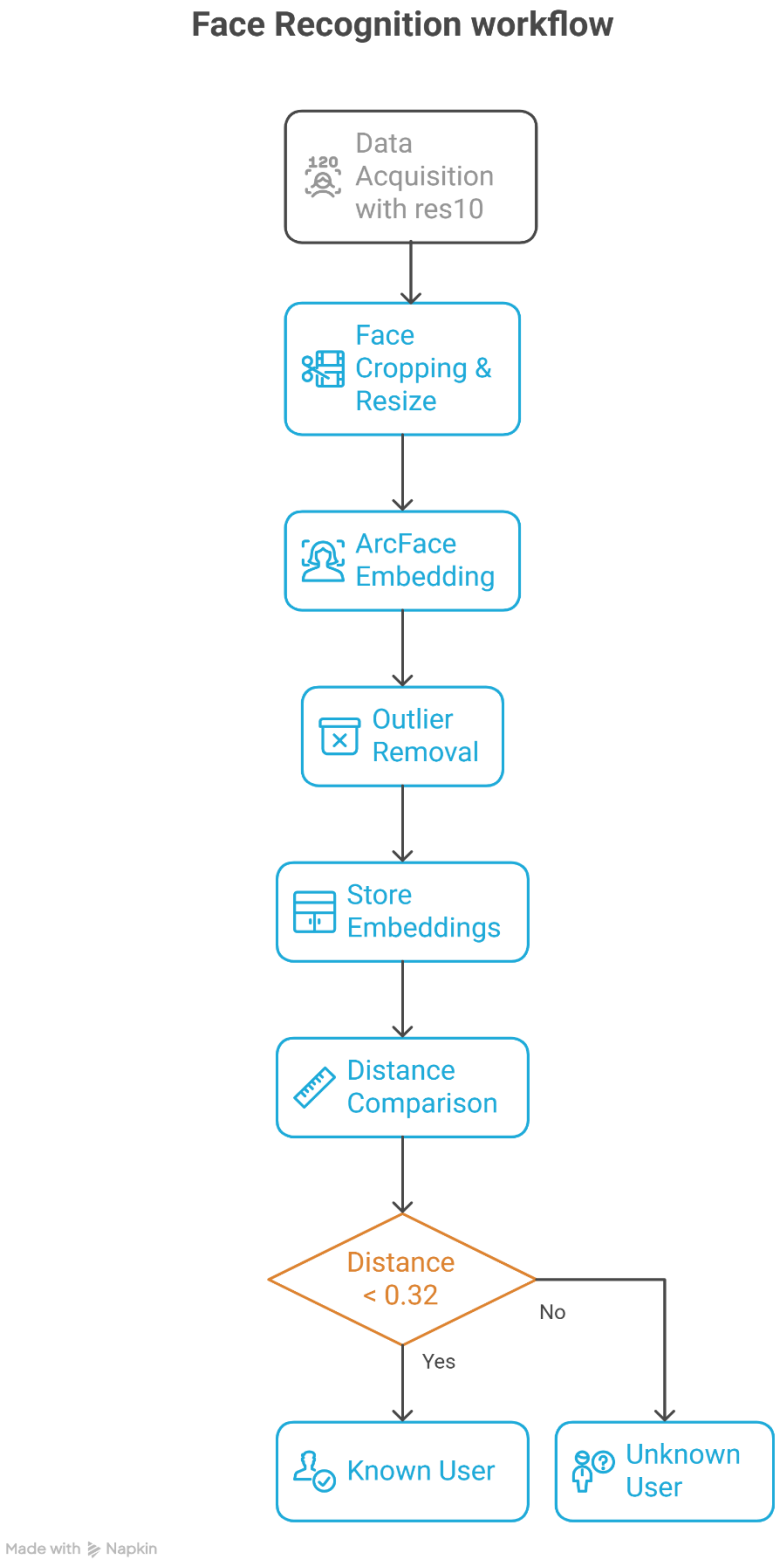


Figure 3.2.1.4 - Flowchart of face recognition system

IV. Evaluation:

at threshold= 0.32

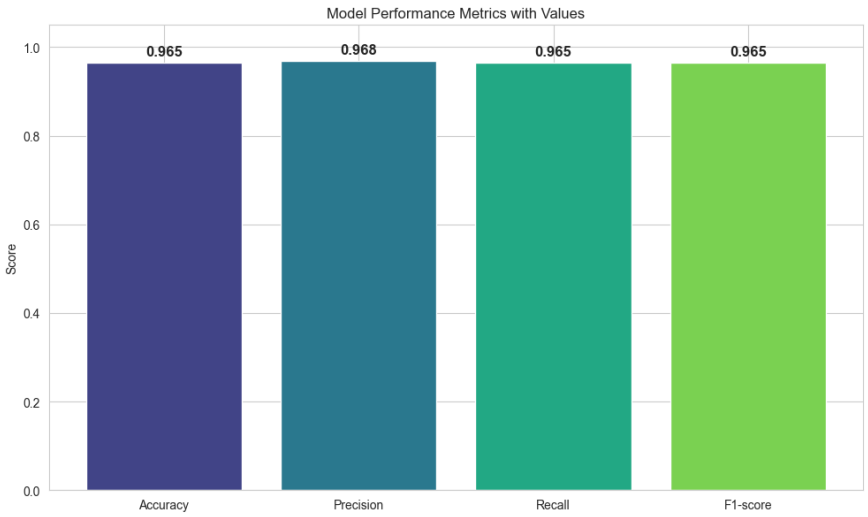


Figure 3.2.1.5 - Performance metrics for ArcFace

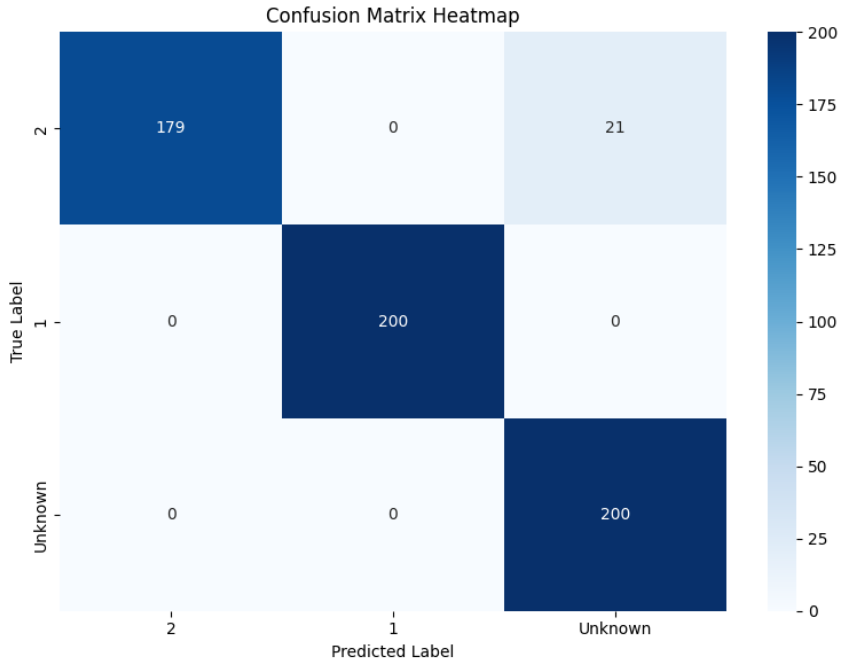


Figure 3.2.1.6 - Confusion matrix for ArcFace

V. Comparison of Different Approaches of AI progress

Model	Dataset Type	Accuracy (at the suitable threshold for real time reliability)	Real-Time Reliability	Notes
LBPH	Gray	~93%	Very Low	Basic, struggled with lighting variations
LBPH + SVM	Gray	~70%	Low	Minor improvement in **real time**, still unstable
FaceNet	RGB	~85%	Medium	Good embeddings but real-time thresholds hard to tune
ArcFace	RGB	96.5%	High	Stable, accurate, suitable for real-time applications

Figure 3.2.1.1– table of comparison of different approaches of AI

VI. Conclusion

ArcFace was selected because it consistently produced accurate and distinguishable face embeddings across different users, especially after outlier removal.

It also maintained smooth performance during real-time testing on modest hardware like Raspberry Pi 5.

3.3 System Software

3.3.1 GUI

- **Overview**

This GUI is a touchscreen interface designed for a Raspberry Pi–based smart door lock system. It enables user management and access configuration, allowing an admin to add, edit, or delete users and set authentication methods such as PIN, fingerprint, and face recognition. The interface also includes a live video feed feature to view the other side of the door, as well as a manual lock/unlock switch for controlling the door directly through the touchscreen. Unlocking the door itself is handled by hardware and backend logic, not through automatic authentication in the GUI. The GUI is used primarily for configuration, enrollment, and monitoring.

1. **Used Technologies**

- **Frontend Framework**: React.js
- **Component Library**: Custom-styled components, no external UI framework
- **Styling**: CSS modules tailored for a 240x320 touchscreen layout
- **UI Logic**: Built-in support for admin PIN verification, user profile management, and toggling multi-factor authentication settings
- **Integration**: Communicates with a FastAPI backend
- **Platform**: Designed to run on Raspberry Pi 5 with a physical touchscreen
- **Additional Features**:
    - T9-style keypad for name input
    - PIN keypad interface
    - Live camera view component

2. Screenshots & Interface Overview

Figure 3.3.1.1 GUI Home Screen

The home screen which allows you to turn the camera display on or off as well as access the settings.

Figure 3.3.1.2 Admin Login Screen

Upon trying to access the settings, the user is prompted to input the Admin pin in order to prevent unauthorized access.

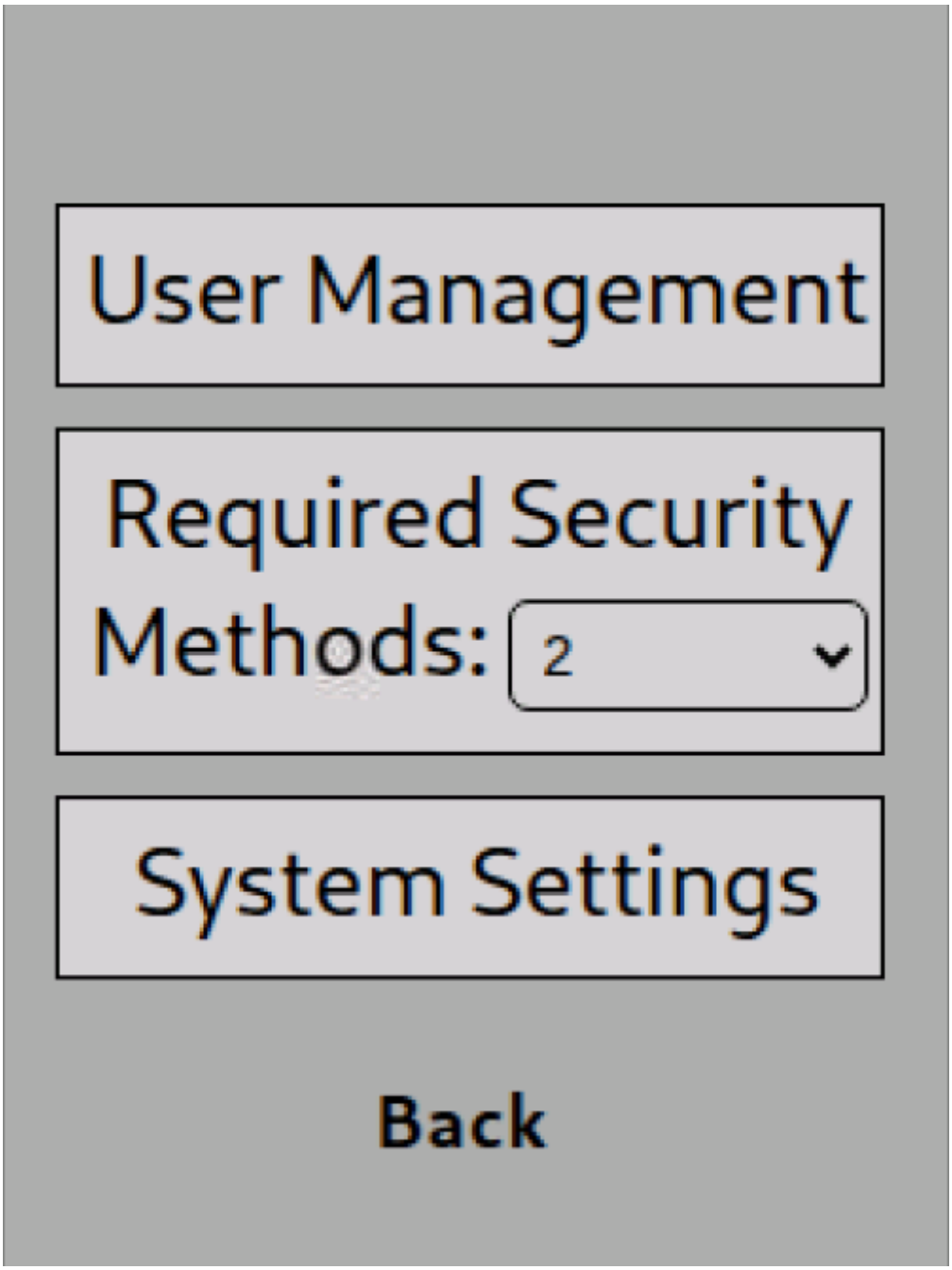


Figure 3.3.1.3 Main Settings Screen

The main settings screen, with buttons for additional settings as well as the option to set the number of Multi-Factor Authentication methods.

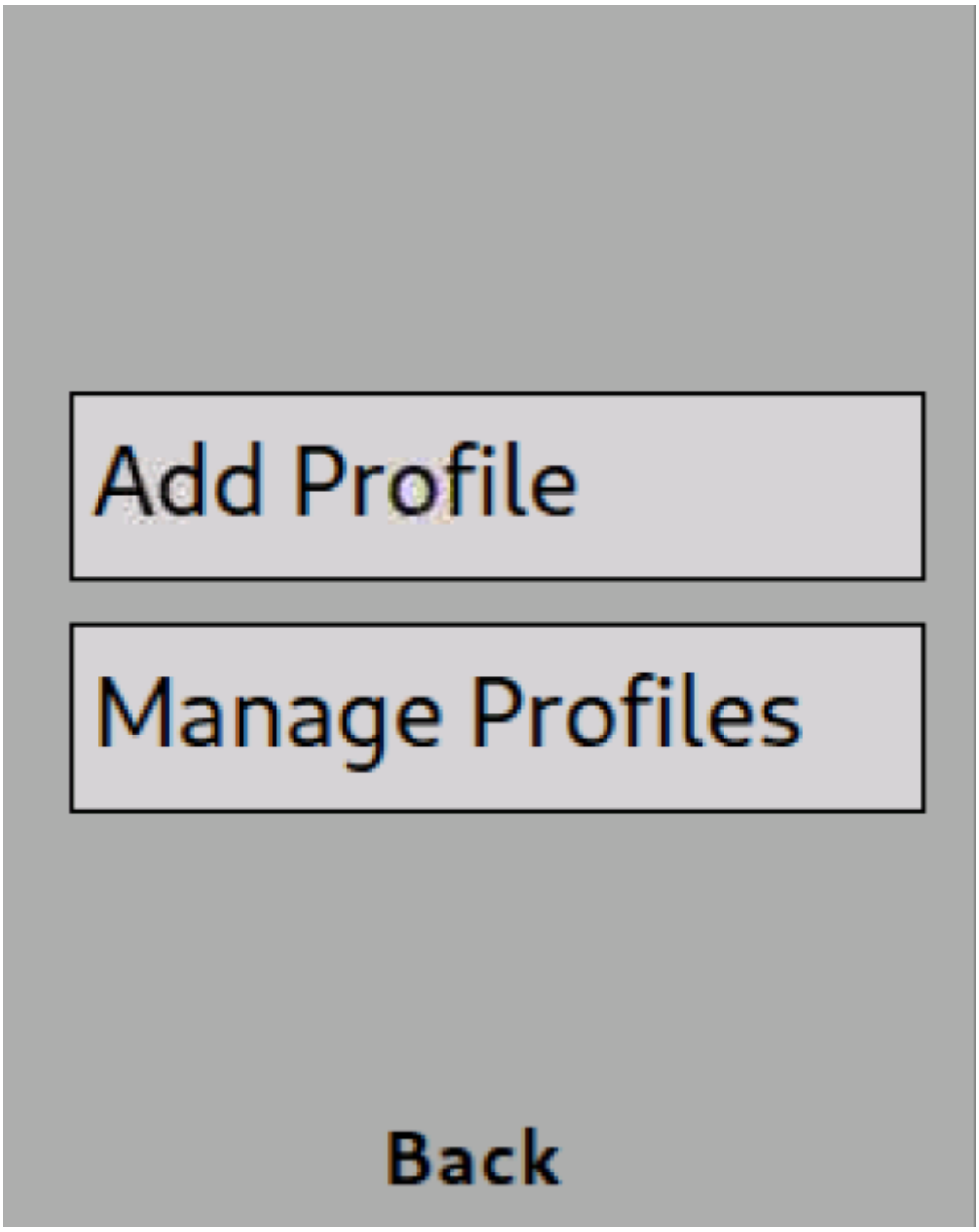


Figure 3.3.1.4 User Management Screen

The User Management screen, which contains the options to add or edit the profiles saved on the device.

Figure 3.3.1.5 Add New User Screen

The screen that lets you set the name and the role of the new profile.

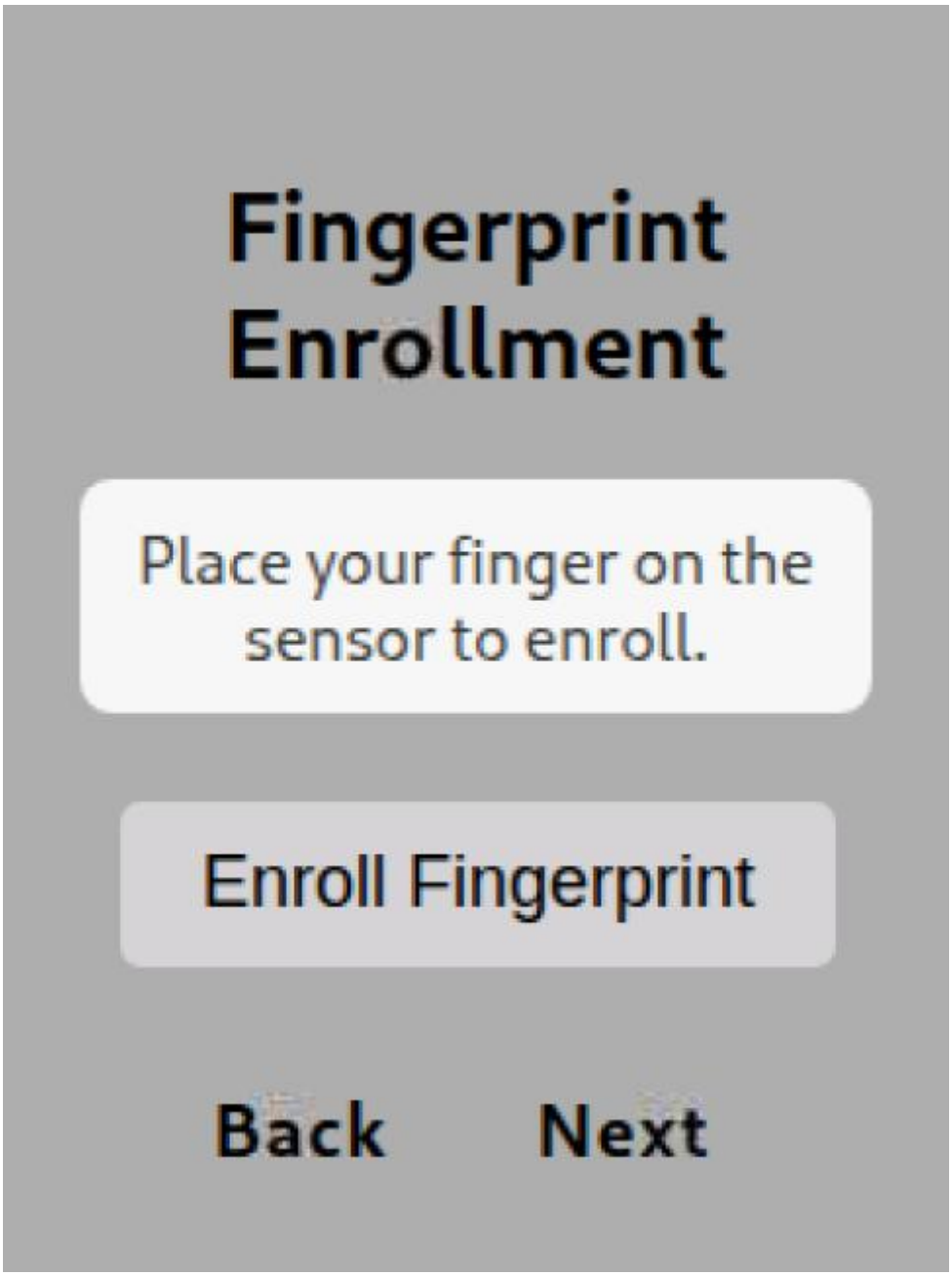


Figure 3.3.1.6 Fingerprint Enrollment Screen

The screen that lets you enroll a new fingerprint scan or edit an existing one.

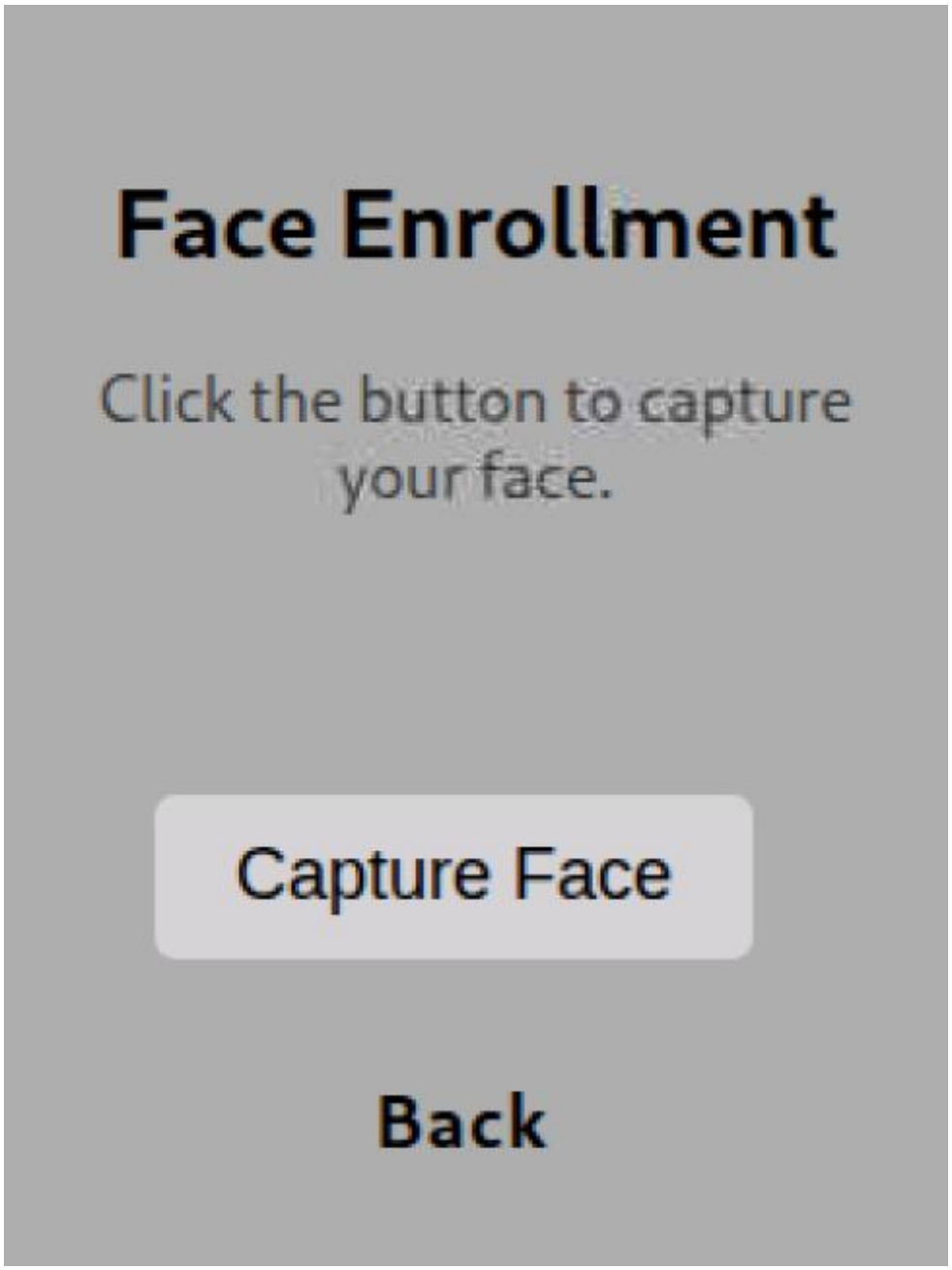


Figure 3.3.1.7 Face Enrollment Screen

The screen that lets you add a new face scan or edit an existing one.

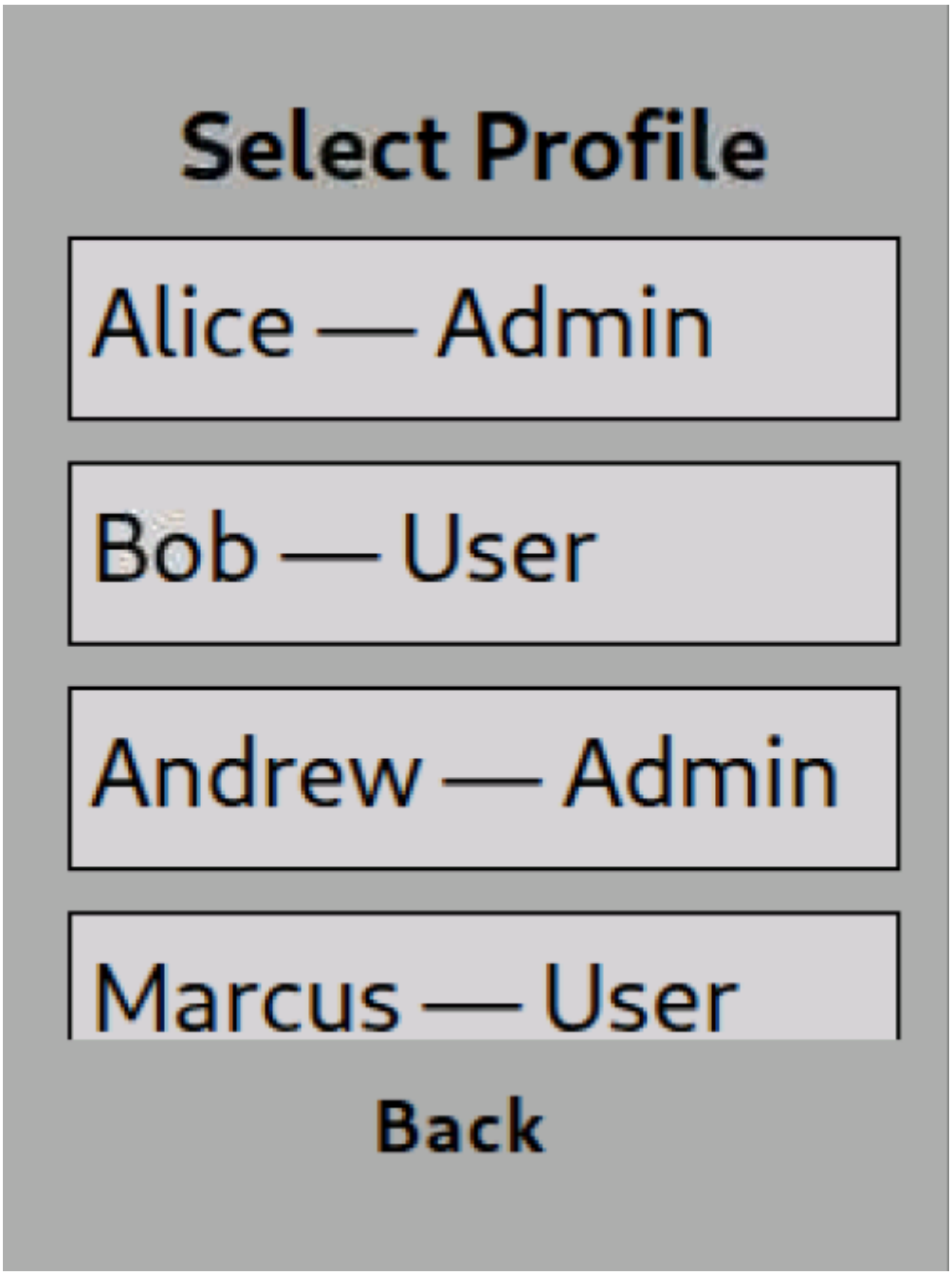


Figure 3.3.1.8 Profile List Screen

The scrollable screen that shows up when the “Manage Profiles” option is selected. It lists the profiles saved on the device and lets you choose which profile to edit.

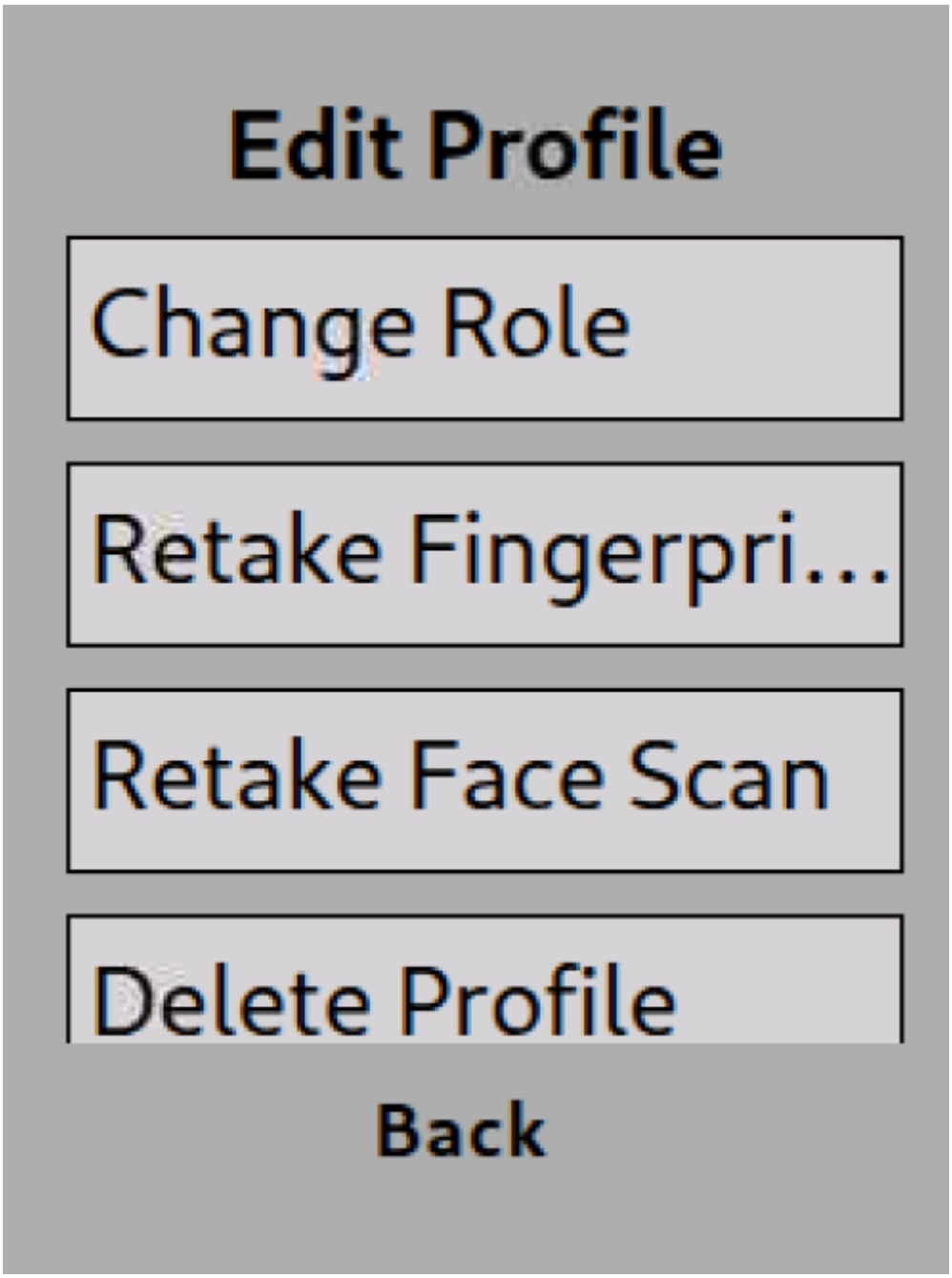


Figure 3.3.1.9 Edit Profile Screen

The scrollable screen that shows up when a profile is chosen to edit. It allows the user to change the role of the chosen profile, retake its fingerprint and face scan, and delete it as well. If the chosen profile is an admin then it allows the user to change the admin PIN, as well.

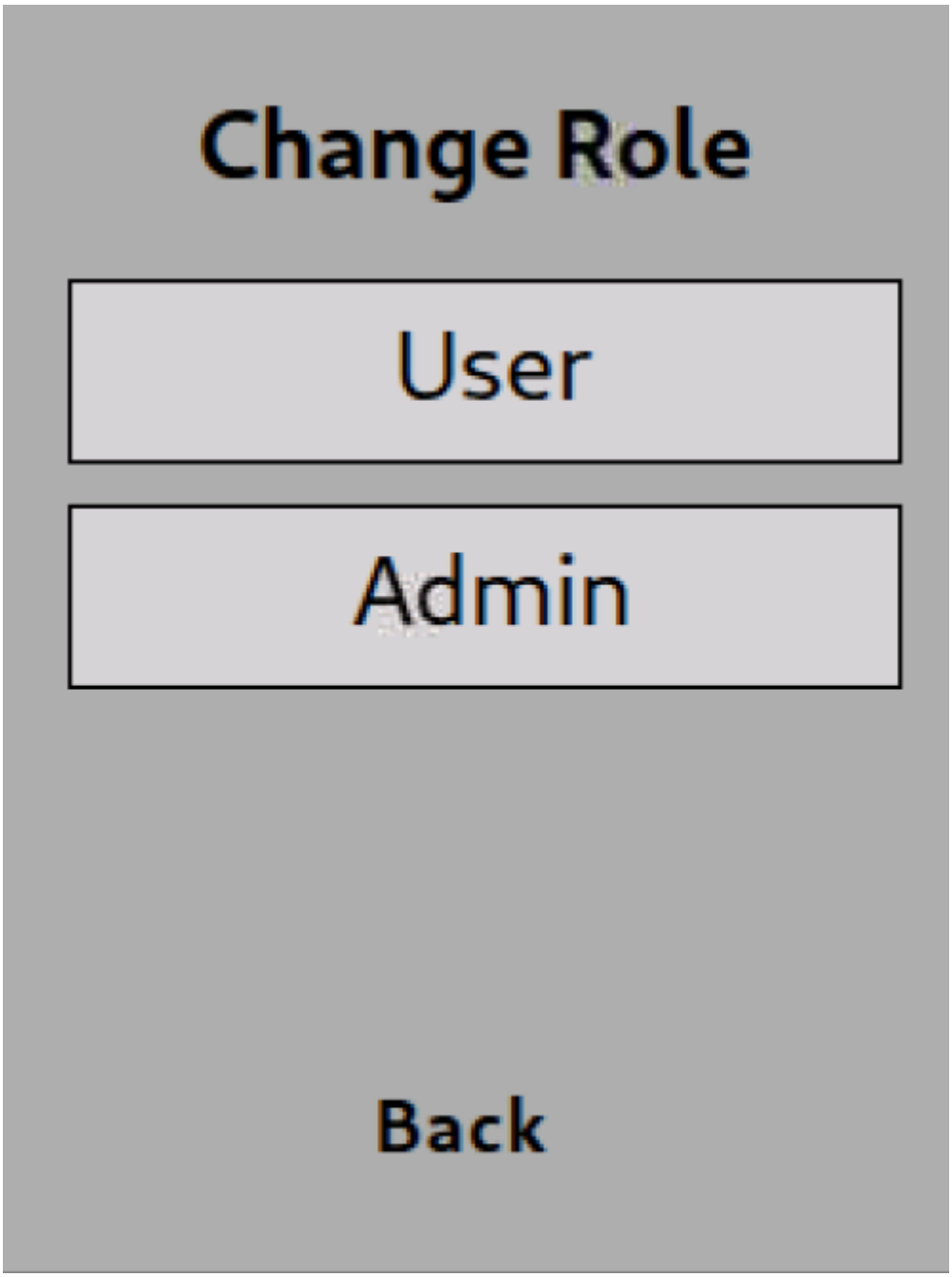


Figure 3.3.1.9 Role Change Screen

The screen that allows the user to change the role of the profile they are editing. Note that there must be at least one admin.

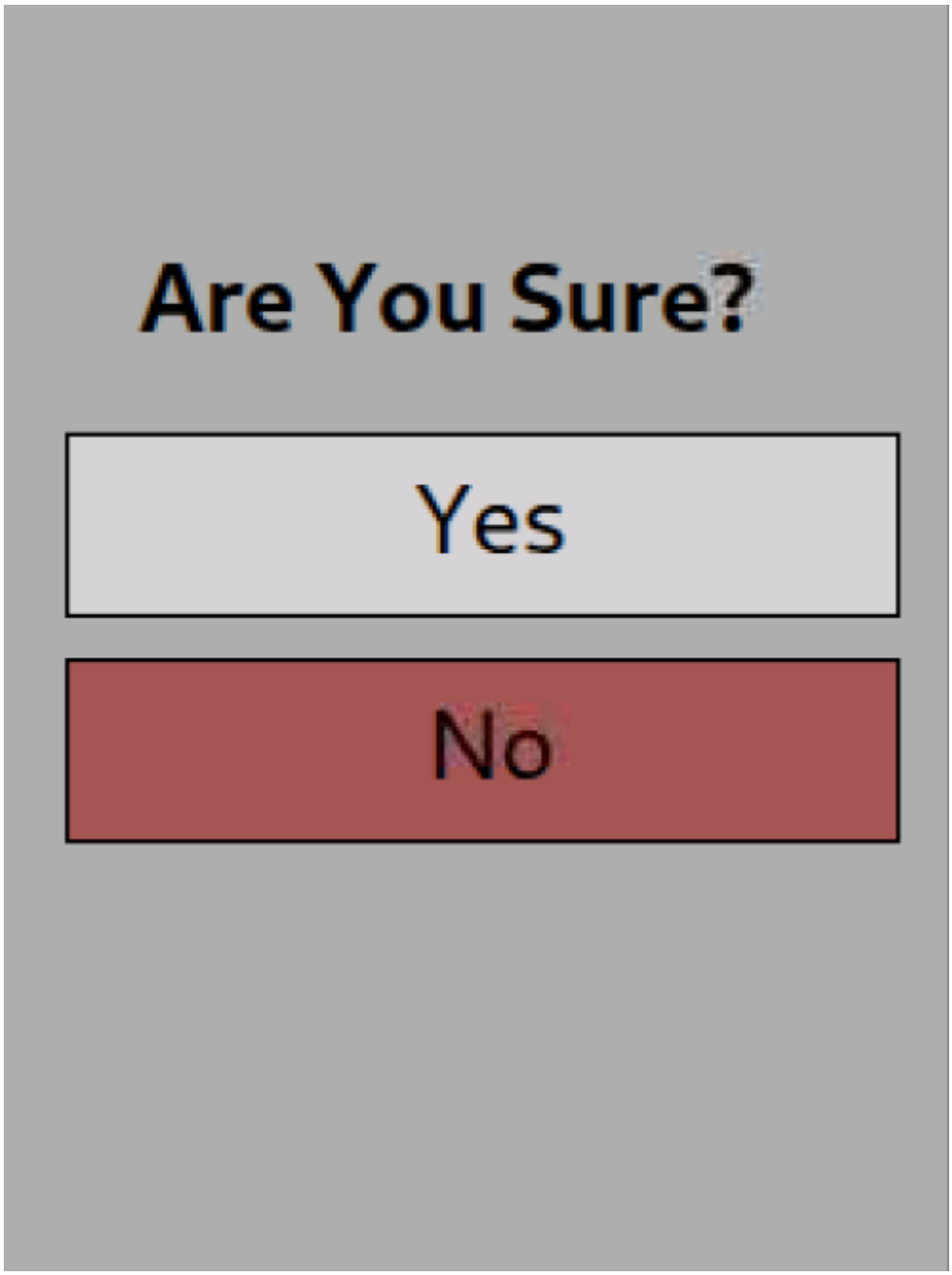


Figure 3.3.1.10 Delete Profile Confirmation Screen

The confirmation screen that shows up if the user tries to delete a profile.

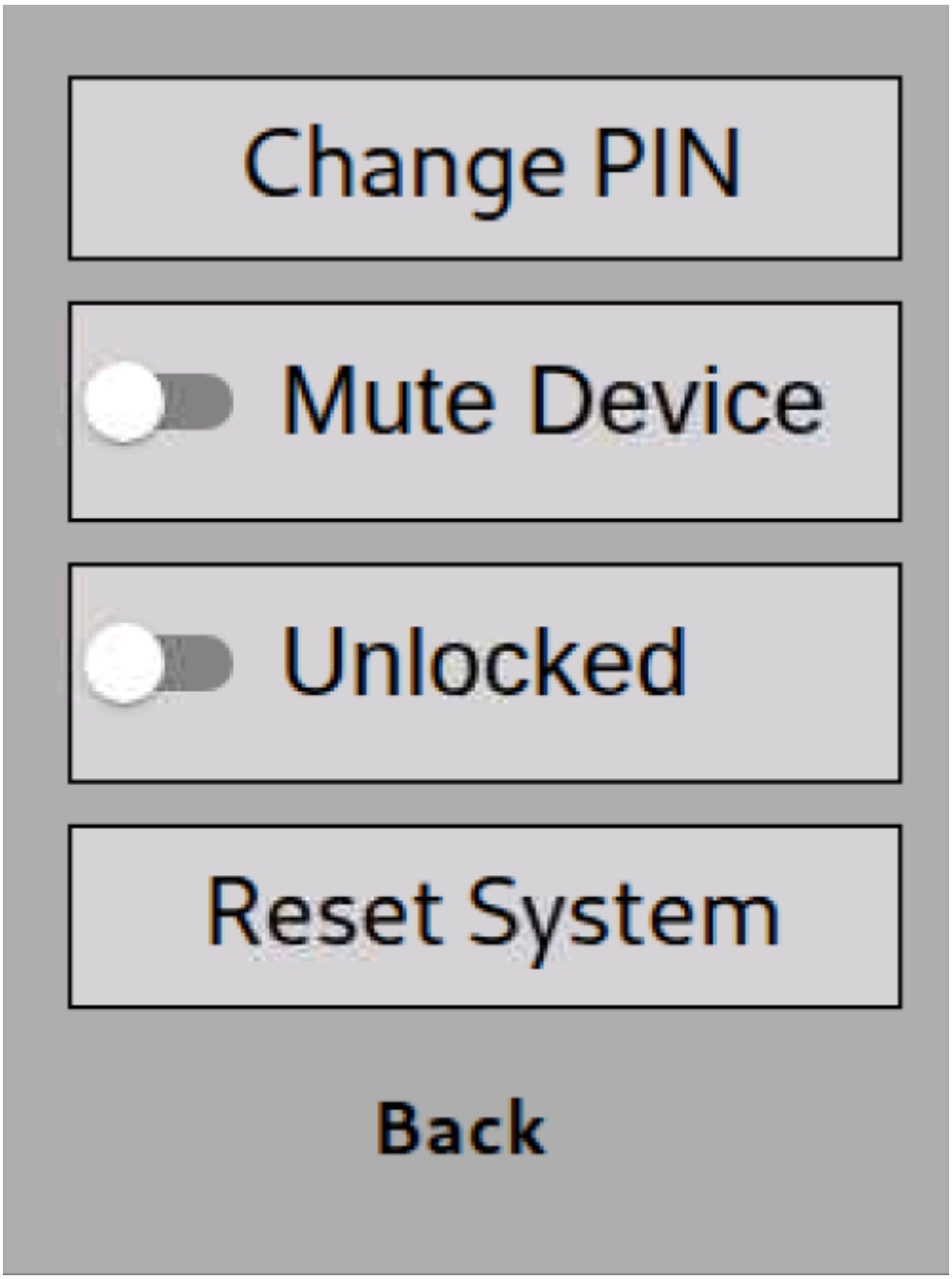


Figure 3.3.1.11 Device Settings Screen

The screen that shows up if the user presses the System Settings button. It lets the user change the device PIN, mute or unmute the system, lock or unlock the door, and reset the system which, upon confirmation, deletes all the profiles, sets the default settings, and prompts the user to create the profile of a new admin.

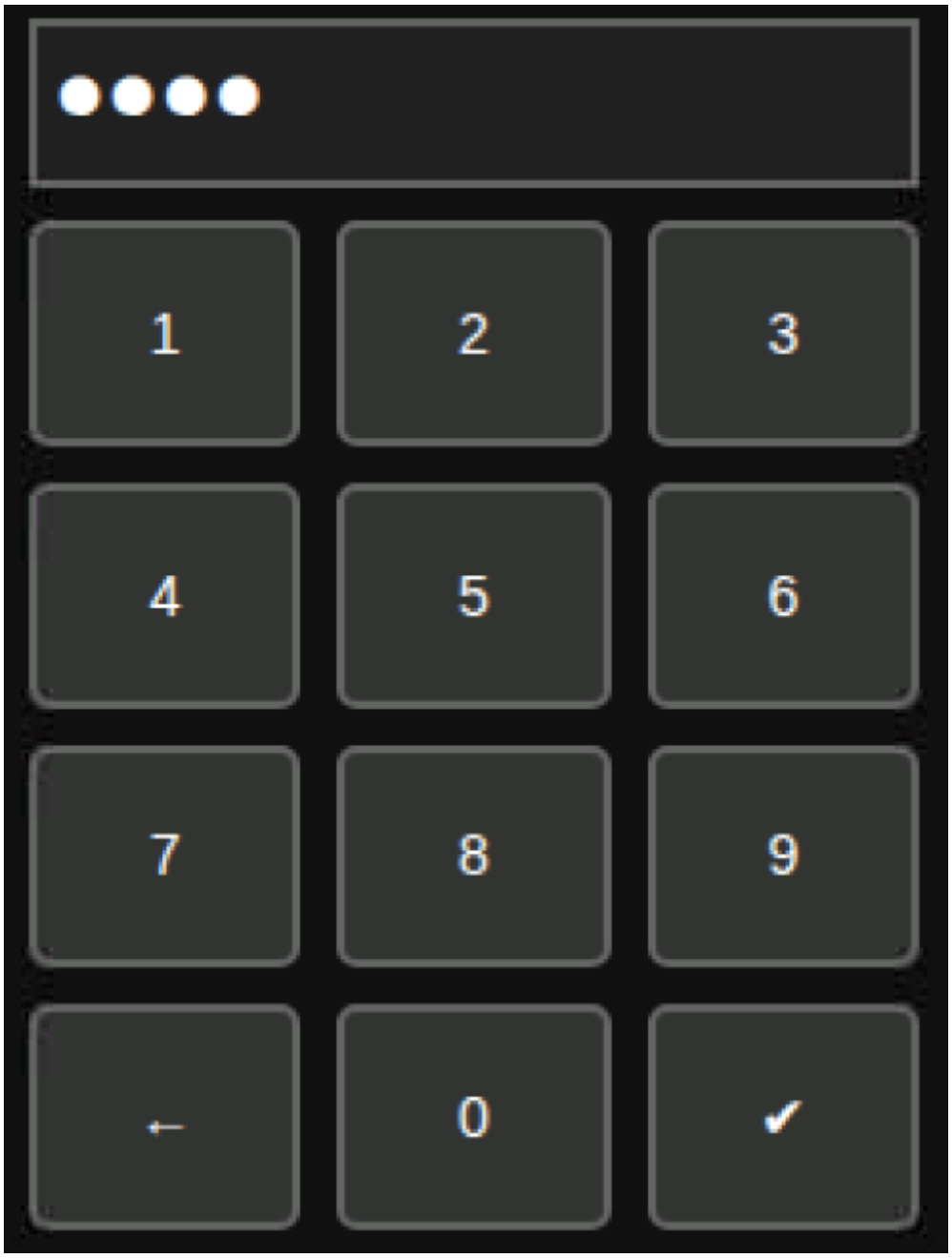


Figure 3.3.1.12 Input Keypad Screen

A keypad with a T9 layout that is used to change PINs, input PINs, and set names of new users. The same keypad is available with the appropriate letters similar to a normal T9 keypad.

3.3.2 Back-end and System Integration

- **Overview:**

This section documents the backend development and integration efforts for the Smart Lock project, a smart door lock system built on a Raspberry Pi 5, from April to May 2025. My contributions focused on developing a robust backend for fingerprint authentication using MQTT and integrating it with Blynk for remote control and monitoring. The section covers the project's file structure, backend architecture, fingerprint management, Blynk integration, challenges, and testing, complementing the team's hardware, frontend, and testing sections.

Backend Architecture

The backend runs on a Raspberry Pi 5 with Python 3.11, located in the /home/salah/doorLockGui/ directory. It leverages a Mosquitto MQTT broker for communication between the fingerprint sensor, Blynk, and other components. The backend is modular, with distinct functionalities for fingerprint management, logging, and doorbell handling.

Full File Structure and Descriptions

This is the full file structure in order with a brief description of each files use cases and integrations with other files

```
doorLockGui/
├── backend/
│   ├── core/              # Has main logic for entry access
│   ├── routes/                  # Has routes for FastAPI and GUI
│   ├── utils/             # Utility files for settings and user profiles
│   └── main.py              # Entry point for the backend Flask server.
│
├── Blynk/
│   ├── templates/         # Holds html files for WebRTC server
│   ├── blynk_code.py    # Handles Blynk MQTT and WebRTC Communication.
│   └── door_logs.txt      # Stores logs for fingerprint events, doorbell triggers, etc.
│
├── deepface_scripts/
│   ├── __pycache__/
│   ├── model/                   # Stores weights for Res10 model
│   ├── _init__.py           # Marks the model folder as a Python package.
│   ├── data_utils.py         # Prepares input images or data for model use.
│   ├── embed_utils.py      # Handles face embeddings generation.
│   ├── face_embeddings.pkl # Stores the embedded vectors of known faces.
│   ├── model_utils.py      # Utilities for Operating face recognition models.
```

```
│   ├── recognize_from_camera.py # Live face recognition from camera feed.
│   └── training_pipeline.py    # Script to train new face data into the system.
├── gui/
│   ├── build/                  # Compiled production-ready frontend.
│   ├── node_modules/           # React dependencies.
│   ├── public/                 # Public assets for the React app.
│   ├── src/                    # Source code for React GUI.
│   ├── .gitignore              # Git ignore rules.
│   ├── package-lock.json       # Version-locked dependency tree.
│   ├── package.json            # React app dependency and config metadata.
│   ├── README.md               # Frontend README file.
│   └── start_react.py          # Launches the React GUI
│
├── hardware/
│   ├── __pycache__/
│   ├── __init__.py             # Marks hardware folder as a package.
│   ├── aggregator.py           # Centralized logging and input aggregation module.
│   ├── bell_listener.py        # Listens for doorbell MQTT topic and triggers notifis.
│   ├── ding.wav                # Audio file played on doorbell press.
│   ├── fp_input.py             # MQTT-based control of fingerprint sensor
│   ├── fp_utils.py             # Utility functions supporting fingerprint logic.
│   ├── keypad_listener.py      # Listens to keypad input events.
│   ├── ToggleDoor.py           # toggles lock state through GPIO.
│   └── UnlockDoor.py           # Handles unlocking logic.
│
├── venv/                       # Python virtual environment for the project.
├── README.md                   # Root-level readme with setup/use instructions.
├── requirements.txt            # Python package dependencies (Flask, paho-mqtt, etc).
├── run.py                      # Launch script for the entire backend system.
└── camera_stream1.py           # Start MJPEG camera server
```

Project Modules

Fingerprint Management Module

- Purpose: Manages fingerprint sensor operations (enrollment, matching, deletion).
- Files: fp_input.py, aggregator.py, fp_utils.py
- Workflow:
  1. Launch Mosquitto on localhost.
  2. fp_input.py connects and handles MQTT commands.
  3. Aggregator logs results to door_logs.txt.
  4. Manual control via fp_input.py if needed.
- Special Conditions:
  - Sensor IDs limited to [1–20] to decrease MQTT delay.
  - Requires paho-mqtt 2.1.0 and Python 3.11.

2. Face Recognition Module

- Purpose: Manages face recognition pipeline operations (add user, recognize user from camera).
- Files training_pipeline.py, recognize_from_camera.py and the 3 util files.
- Workflow:
  1. Launch MJPEG server and start recognize_from_camera.
  2. Receive a command from PIR sensor through recognize or user management from GUI..
  3. Send result to aggregator which logs the result to doorlock.log.

3. Blynk Integration Module

- Purpose: Enables remote control and monitoring via Blynk mobile app.
- Files: blynk_code.py, aggregator.py
- Workflow:
  1. blynk_code.py connects and subscribes to topics.
  2. Monitors messages between ESP and ESP dev board to log activity.
  3. Logged events are saved to door_logs.txt which would be visible in APP.

4. Logging Module

- Purpose: Records operational events for troubleshooting.
- Files: aggregator.py, door_logs.txt, doorlock.log
- Workflow:
  1. Modules call log_event during operations.
  2. Log entries are written with timestamps.

3. Some logs are GUI visible while others aren't
- Special Conditions:
  - doorlock.log is in /tmp/ which is cleared every restart or 10 days to restrict file size

5. Doorbell Handling Module

- Purpose: Processes doorbell press events via MQTT.
- Files: bell_listener.py, aggregator.py
- Workflow:
  1. bell_listener.py listens on door/bell.
  2. On event, triggers a log and publishes "unlock"to the solenoid lock.

3.4 Remote Access App

3.4.1. Software Configuration for Blynk

1.Overview of Blynk IoT Platform

Blynk is a versatile IoT platform designed for building mobile applications for connected devices. It comprises a mobile app, a cloud server, and a microcontroller library, offering the following capabilities:

- **Customizable App Interface**: Features a drag-and-drop interface for creating tailored dashboards with widgets such as buttons, displays, and terminals.
- **Device Communication**: Facilitates seamless connectivity between IoT devices (e.g., ESP8266-01) and the Blynk cloud for efficient control and data exchange.
- **Event Management**: Supports real-time notifications and event triggers, such as alerts for doorbell presses or door status updates.
- **Cross-Platform Support**: Compatible with iOS and Android, ensuring accessibility across a wide range of devices.
- **Remote Access**: Enables users to lock or unlock the door and view live video feeds from any location with an internet connection.
- **Real-Time Notifications**: Delivers instant push notifications for events such as doorbell presses or door status changes, enhancing security and user awareness.
- **User-Friendly Interface**: Simplifies interaction compared to web interfaces or physical controls, making the system accessible to non-technical users.
- **Portability**: Leverages the ubiquity of smartphones, allowing users to manage the door lock on the go without additional hardware.

In this project, Blynk serves as the primary interface, hosting widgets for door control, status monitoring, event logs, and live video streaming.

2.Comparison of Blynk and Tuya

Feature	Blynk	Tuya
Customization	Highly flexible drag-and-drop interface for custom widget design.	Limited customization, tailored for pre-built smart devices.
Hardware Compatibility	Supports a wide range of hardware and protocols (e.g., MQTT, ESP8266).	Primarily designed for Tuya-specific devices, less flexible for custom hardware.
Integration	Seamless integration with ESP8266 and MQTT for custom IoT setups.	Requires proprietary hardware or additional setup for custom applications.
Cost	Free tier sufficient for small-scale projects; open ecosystem.	Often requires subscriptions for advanced features; proprietary ecosystem.
Development Speed	Rapid prototyping with minimal coding via drag-and-drop interface.	Slower setup for custom applications due to ecosystem restrictions.

Why Blynk is Preferred: Blynk's flexibility in customization and seamless integration with the ESP8266-01 and MQTT make it ideal for our custom smart door lock system. Its open ecosystem and rapid prototyping capabilities outperform Tuya's proprietary, less adaptable platform.

3. Features of the Blynk App for Smart Door Lock Control

The Blynk app is designed to provide comprehensive control and monitoring for the smart door lock system, incorporating the following features:

- **Door Lock/Unlock Control (V1)**: A button widget sends an unlock command via MQTT to the ESP8266 Board (authentication not yet implemented).
- **Door Status Monitoring (V2)**: A Value Display widget shows the door's current state ("Open" or "Closed") based on MQTT updates from the ESP8266 Board.
- **Event Log Display (V3)**: A Terminal widget logs events such as "Bell Button Pressed" or "Door Opened via App" received via MQTT.
- **Live Video Feed Streaming (V4)**: A WebView widget streams live video from the Raspberry Pi Camera Module 3 using MJPG-Streamer at `http://192.168.1.12:8080/stream.mjpg`.
- **Event Notifications**: Push notifications alert users to events like "Doorbell Pressed!" or "Door Opened!" via Blynk's event system.

4. Role of MQTT in the Smart Door Lock System

Overview of MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight, publish/subscribe messaging protocol optimized for IoT devices over TCP/IP networks.

Functionality in the Project

MQTT facilitates communication between the ESP8266-01 (Blynk bridge), the Raspberry Pi 5, and the ESP8266 Board. The workflow includes:

- The ESP8266-01 publishes unlock commands to the `door/cmd` topic.
- The ESP8266 Board subscribes to `door/cmd` to control the solenoid lock and publishes door status (`door/lock`) and bell status (`door/bell`).
- The Raspberry Pi 5 relays status updates to Blynk via `door/lock` and `door/bell` topics.

Advantages

- **Lightweight Protocol**: Minimal overhead ensures efficient communication for low-power devices like the ESP8266-01.
- **Scalable Architecture**: The publish/subscribe model supports multiple devices and topics, enabling future feature expansion.
- **Reliable Delivery**: Ensures consistent message delivery, even in unstable network conditions.

Disadvantages

- **Network Dependency**: Requires a stable Wi-Fi connection; disruptions halt communication (though local logs and status persist).
- **Security Risks**: Lacks built-in encryption, relying on local network security (future SSL integration recommended).

5. Role of MJPG-Streamer in the Smart Door Lock System

Overview of MJPG-Streamer

MJPG-Streamer is an open-source tool that streams Motion JPEG (MJPEG) video from a camera over a network, optimized for devices like the Raspberry Pi.

Functionality in the Project

MJPG-Streamer captures video from the Raspberry Pi Camera Module 3 and streams it to the Blynk app's WebView widget (V4) at `http://192.168.1.12:8080/stream.mjpg`. It is initiated via a placeholder function in the system code.

Advantages

- **Low-Latency Streaming**: Delivers real-time video with minimal delay, ideal for live monitoring.
- **Resource Efficiency**: Operates effectively on the Raspberry Pi 5 with moderate resource consumption.
- **Open-Source Flexibility**: Customizable and cost-free, aligning with the project's open ecosystem.

Disadvantages

- **Bandwidth Intensive**: High-resolution or frame-rate streams can strain network bandwidth.
- **Camera Dependency**: Requires a compatible camera module, with setup sensitive to hardware variations.
- **Limited Features**: Lacks advanced video processing capabilities (e.g., motion detection) out of the box.

6. Software Setup

Raspberry Pi 5

- **Operating System**:
  - Install Raspberry Pi OS (64-bit recommended):

```
sudo apt update
sudo apt full-upgrade
```

- **Dependencies**:
  - Install Python 3, pip, and libraries:

```
sudo apt update
sudo apt install python3-pip python3-dev
pip3 install paho-mqtt RPi.GPIO adafruit-
circuitpython
```

- **MJPG-Streamer**:
  - Install dependencies and build for ARM64:

```
sudo apt install libjpeg-dev libuvc-dev
git clone https://github.com/jacksonliam/mjpg-streamer.git
cd mjpg-streamer/mjpg-streamer-experimental
make
sudo make install
```

- **MQTT**:
  - Install Mosquitto:

```
sudo apt install mosquitto mosquitto-clients
```

ESP8266-01 (Blynk Bridge)

  - Install Arduino IDE with ESP8266 board support.
  - Install Blynk library via Library Manager.
  - Upload `esp8266_bridge.ino` using a USB-to-serial adapter.

ESP8266 Board

  - Install Arduino IDE with ESP8266 board support.
  - Upload `esp8266_lock_bell.ino` using a USB-to-serial adapter.

7. Blynk Configuration

Project Setup

- Create a new project in Blynk Console (`https://blynk.cloud`).
- Assign Template ID (e.g., `TMPL12345678`) and Template Name (e.g., "SmartLock").
- Update Auth Token (e.g., `BwWgrbXxUjQyaJZ6x2R4OVhNJZnhtxGy`) in `esp8266_bridge.ino`.

Widget Configuration

- **V1**: Button (Unlock, Switch, sends `1` to trigger unlock command).
- **V2**: Value Display (Door Status, displays "Open" or "Closed").
- **V3**: Terminal (Logs, displays events like "Bell Button Pressed").
- **V4**: WebView (Live Feed, URL: `http://192.168.1.12:8080/stream.mjpg`).

Event Configuration

- **bell_pressed**: Notification "Doorbell Pressed!" (Type: Info).
- **door_opened**: Notification "Door Opened!" (Type: Info).
- Enable push notifications with alert sounds in Blynk app settings.

3.4.2. Hardware Configuration for Blynk

1.Components

Raspberry Pi 5

- **Model**: Raspberry Pi 5 (4GB or 8GB RAM).
- **GPIO Pins**: GPIO 18 and GPIO 24 are unused, available for future enhancements.
- **Peripherals**:
  - USB sound card for speaker audio output.
  - Raspberry Pi Camera Module 3 for MJPG-Streamer video feed.
- **Power Supply**: 5V/3A USB-C with Power Delivery (PD) for stable operation.

ESP8266-01 (Blynk Bridge)

- **Model**: ESP8266-01, relaying commands and status between Blynk and Raspberry Pi 5 via MQTT over Wi-Fi.
- **Power**: Connected to Raspberry Pi USB

ESP8266 Board

- **Model**: ESP8266, controlling the solenoid lock and monitoring the bell button.
- **GPIO Pins**:
  - D1: Solenoid lock relay (output).
  - D2: Bell button (input, pull-up).
- **Connectivity**: Communicates with Raspberry Pi 5 via MQTT over Wi-Fi.

2. Purpose and Rationale for Using ESP8266-01 as a Blynk Bridge

The ESP8266-01 serves as a bridge between the Blynk app and the Raspberry Pi 5, selected for the following reasons:

- **Wi-Fi Connectivity**: Built-in Wi-Fi enables seamless communication with the Blynk cloud and MQTT-based interaction with the Raspberry Pi 5.
- **Compact Design**: Its small form factor ensures easy integration without adding bulk to the system.
- **Cost-Effectiveness**: Affordable hardware reduces project costs while maintaining functionality.
- **Blynk Compatibility**: Robust support for the Blynk library simplifies virtual pin (V1–V5) configuration for app interaction.
- **MQTT Support**: Acts as an MQTT client, relaying commands from Blynk to the Raspberry Pi 5 and receiving status updates.

Chapter 4: Conclusion

The Smart Door Lock project represents a significant leap forward in access control systems by integrating advanced artificial intelligence (AI) and Internet of Things (IoT) technologies to create a secure, flexible, and user-friendly solution. By incorporating multi-factor authentication—combining face detection (via the ArcFace algorithm with 96.5% accuracy), fingerprint recognition, keypad access, and smartphone-based remote control through the Blynk IoT platform—the system effectively addresses the vulnerabilities of traditional single-factor authentication methods, such as susceptibility to lock picking, key duplication, and unauthorized access. The modular hardware architecture, centered around the Raspberry Pi 5, ESP8266-01 (Blynk bridge), and ESP8266 Board, ensures scalability, cost-effectiveness, and seamless integration with IoT ecosystems. The intuitive React.js-based GUI, tailored for a 240x320 touchscreen, simplifies user management and system configuration, while the Blynk app enhances accessibility with real-time monitoring, event notifications, and live video streaming via MJPG-Streamer. MQTT's lightweight publish/subscribe model facilitates efficient communication between components, enabling reliable operation even with low-power devices.

The system's versatility makes it suitable for a wide range of applications, from residential homes to high-security laboratories and commercial offices. Its configurable authentication layers (one to three) allow users to balance security and convenience, while role-based access control (admin and normal user) enhances system management. Market research indicates that the system is competitively positioned, offering advanced features at a cost comparable to existing solutions (e.g., Cordless and Lzen products priced between 13,500–18,000 EGP). The use of open-source tools like MJPG-Streamer and Blynk's flexible platform further reduces development costs while maintaining robust functionality.

4.1 Limitations

Despite its strengths, the Smart Lock Symphony system has several limitations that impact its performance and scalability:

1. **Network Dependency**: The system relies heavily on a stable Wi-Fi connection for MQTT communication and Blynk app functionality. Network disruptions halt remote access, lock commands, and notifications, although local logs and status updates persist.
2. **Lack of Two-Way Audio**: The current implementation does not support real-time two-way audio communication, limiting interaction with visitors at the door.
3. **Security Gaps**: The absence of authentication (e.g., PIN or biometric verification) for unlocking via the Blynk app poses a risk of unauthorized access if the app is compromised. Additionally, MQTT's lack of built-in encryption relies on local network security, increasing vulnerability to interception.
4. **Unlock-Only Command**: The system currently supports only an unlock command via the `door/cmd` topic, with no corresponding lock command, limiting control flexibility.
5. **No Dedicated Door Status Sensor**: The system depends on the ESP8266 Board's signals for door status without an independent sensor (e.g., a magnetic switch), which may lead to inaccurate status reporting.

6. **Bandwidth Constraints for Video Streaming**: MJPG-Streamer's high-resolution video feeds can strain network bandwidth, potentially causing lag or reduced performance in low-bandwidth environments.
7. **Camera Dependency**: The video streaming functionality requires a compatible Raspberry Pi Camera Module 3, and setup is sensitive to hardware variations, which may complicate deployment.
8. **Limited Fingerprint Sensor Capacity**: The R301T fingerprint sensor supports only 20 fingerprints, which may be insufficient for large-scale deployments with many users.
9. **Lack of Advanced Video Processing**: MJPG-Streamer lacks built-in features like motion detection, requiring additional software for enhanced video analytics.

4.2 Future Work

To address these limitations and enhance the system's capabilities, the following improvements are proposed:

1. **Implement Authentication for Unlocking**: Introduce PIN or biometric authentication (e.g., fingerprint or face recognition) within the Blynk app to secure remote unlocking and prevent unauthorized access.
2. **Add Lock Command**: Extend MQTT functionality to include a lock command alongside the existing unlock command, providing full control over the door's state.
3. **Incorporate SSL/TLS for MQTT**: Implement secure communication protocols to encrypt MQTT messages, reducing the risk of interception and enhancing overall system security.
4. **Integrate Two-Way Audio**: Add a microphone and speaker to enable real-time audio communication with visitors, enhancing user interaction and monitoring capabilities.
5. **Add a Dedicated Door Status Sensor**: Incorporate a magnetic switch or similar sensor to independently verify the door's open/closed status, improving reliability and accuracy.
6. **Optimize Video Streaming**: Implement adaptive bitrate streaming or lower-resolution options for MJPG-Streamer to reduce bandwidth usage and improve performance in low-bandwidth environments.
7. **Enhance Fingerprint Sensor Capacity**: Upgrade to a fingerprint sensor with higher storage capacity or implement a cloud-based storage solution for biometric templates to support larger user bases.
8. **Improve Network Resilience**: Develop an offline mode or local fallback mechanism to maintain basic functionality (e.g., local authentication and logging) during network outages.
9. **Expand Multi-Factor Authentication Options**: Introduce additional authentication methods, such as RFID or voice recognition, to further enhance security and flexibility.
10. **Optimize MQTT Latency**: Fine-tune MQTT communication to reduce delays, particularly for fingerprint enrollment and deletion commands, by optimizing sensor ID ranges and network configurations.
11. **Add Manual Lock from Inside**: Implement a physical manual lock mechanism on the interior side of the door to prevent unauthorized entry, even if remote or electronic unlocking is attempted, enhancing physical security.

12. **Introduce Super User Role and Single Admin Restriction**: Add a "super user" role with elevated privileges (e.g., edit , add and remove users) while restricting the system to maintain only one admin at a time to ensure clear authority and prevent conflicts in administrative control.

4.3 Final Remarks

The Smart Door Lock project successfully demonstrates the potential of combining AI-driven biometric authentication with IoT technologies to create a robust, scalable, and user-friendly access control system. By addressing the shortcomings of traditional locks—such as vulnerability to bypass techniques and lack of remote access—it offers a modern solution tailored to the demands of connected environments. With the proposed enhancements, including improved security protocols, expanded functionality, and greater resilience, the system has the potential to become a comprehensive and competitive solution for smart security applications. This project lays a strong foundation for future innovations, contributing to safer, more connected, and technologically advanced environments.

References

1. IoT Assisted Fingerprint-Based Door Security System using Raspberry Pi 4, International Journal of Trend in Scientific Research and Development (IJTSRD) Volume 4 Issue 3, April 2020

2. Real Time Access Control Based on Face Recognition, 2015 International Conference on Network security & Computer Science (ICNSCS-15) Antalya (Türkiye), May 2023

3. Human Face Detection & Recognition Using Raspberry Pi, International Journal of Advanced Engineering, Management and Science, (ICSESD-2017)

4. Smart Door System using Face Recognition Based on Raspberry Pi, JURNAL INFOKUM, Volume 10, No.1, December 2021

5. Face Recognition for Smart Door Lock System Using Hierarchical Network, 2020 International Conference on Computational Intelligence (ICCI), October 8-9,2020

6. RFID and Fingerprint Based Dual Security System: A Robust Secured Control to Access Through Door Lock Operation, American Journal of Embedded Systems and Applications, June 15, 2018

7. A Model of Secured ATM Pin Recovery with Face and Fingerprint Identification, National Conference on Contemporary Research and Computer Intelligence 2022, 28th October 2022

8. Secure Access Microcontroller System Based on Fingerprint Template with Hyperchaotic Encryption, Integration, the VLSI Journal, 5 January 2023

9. A Fingerprint & Pin Authentication to Enhance Security at The Automatic Teller Machines, International Journal of Scientific & Engineering Research, Volume 8, Issue 4, April-2017

10. Enhanced Cloud Security Model Using Combined Multiple Biometric Features, National Conference on Contemporary Research and Computer Intelligence 2022, 28th October 2022

11. Face Recognition Based Security System Using Raspberry Pi, Journal of Emerging Technologies and Innovative Research Volume 8, Issue 7, July 2021

12. Prevention of Unauthorized Door Access Using Face Recognition, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016, 20 May 2020

13. A Prototype Model of an IoT-based Door System using Double-access Fingerprint Technique, NIGERIAN JOURNAL OF TECHNOLOGICAL DEVELOPMENT, VOL. 17, NO.2, JUNE 2020

14. Finger Scanner: Embedding a Fingerprint Scanner in a Raspberry Pi, Department of Computer Science and INSPIRES, University of Lleida, 6 February 2016