Design Document
Smart Biometric Access Control System for Shared Living Environments

Jihao Li Mujia Li Shixuan Ma Denghan Xiong

April 8, 2026

Contents

1 Introduction

1.1 Problem Statement

Shared living environments such as university dormitories and shared apartments require an access control solution that is both secure and convenient. Traditional mechanisms such as mechanical keys and password-based systems suffer from multiple weaknesses. Keys can be lost, duplicated, or borrowed without authorization, and passwords can be forgotten, leaked, or shared. In addition, conventional locks provide no audit trail, making it impossible to determine who accessed a room and at what time.

Although commercial smart locks are available, many rely on smartphones or cloud-connected platforms, which either increase user burden or raise privacy concerns. High-end biometric access control systems exist, but their cost is often too high for large-scale deployment in student housing or low-cost rental properties. Therefore, there is a clear need for a low-cost, privacy-preserving, and locally processed access control system that can provide contactless authentication, access logging, and improved resistance to unauthorized entry.

This project proposes a smart biometric access control system built around a Raspberry Pi, a PIR sensor, a USB camera, and a local facial recognition pipeline. The system automatically detects a user approaching the door, captures a facial image, performs recognition locally, and unlocks an electronic door strike if the user is authorized. The system also records access events and supports additional security features such as liveness verification and coercion alert logic.

1.2 Solution Overview and Visual Aid

The proposed system is organized as a local biometric access platform. A PIR motion sensor first detects the presence of a person near the door and wakes the vision pipeline. A USB camera then captures image frames, which are preprocessed and passed to the face recognition module running on the Raspberry Pi. The recognition subsystem compares extracted facial features with templates stored in a local database. If a successful match is obtained and no security exception is triggered, the control logic activates a relay module to unlock the electronic strike for a limited time. In parallel, the user interface displays system status and the logging subsystem records all access attempts locally.

Figure 1: Prototype implementation of the smart biometric access control system, including the door-side sensing unit, keypad, electronic door lock, Raspberry Pi platform, and monitoring display.

1.3 High-Level Requirements List

To ensure the proposed design is technically meaningful and verifiable, the overall system shall satisfy the following high-level requirements:

(1) The system shall detect a person approaching within a range of 2.0 ± 0.2 meters and trigger the camera capture sequence within 500 ms of detection.

(2) The facial recognition subsystem shall achieve a false acceptance rate (FAR) below 0.1% and a false rejection rate (FRR) below 5% under standard indoor lighting conditions of 300–500 lux.

(3) The total latency from initial motion detection to relay activation shall not exceed 2.0 seconds.

(4) The system shall maintain local logging of all access attempts, including time, result, and recognized identity or unknown status.

(5) The system shall preserve user privacy by processing facial data locally without transmitting biometric information to external cloud services.

2 Design

2.1 Block Diagram

The system is divided into six main modules: Sensing, Control, Face Recognition Processing, Security Enhancement, User Interface and Management, and Power. The PIR sensor sends a GPIO wake-up signal to the Raspberry Pi, and the USB camera sends image data through USB. The face recognition module performs image preprocessing, feature extraction, and database comparison, then returns the authentication result to the control logic. The control logic drives the relay module and determines whether the door unlocks. The user interface displays the system state and recent events, while the management module stores logs and user data locally. The power subsystem provides regulated low-voltage power to all active components.

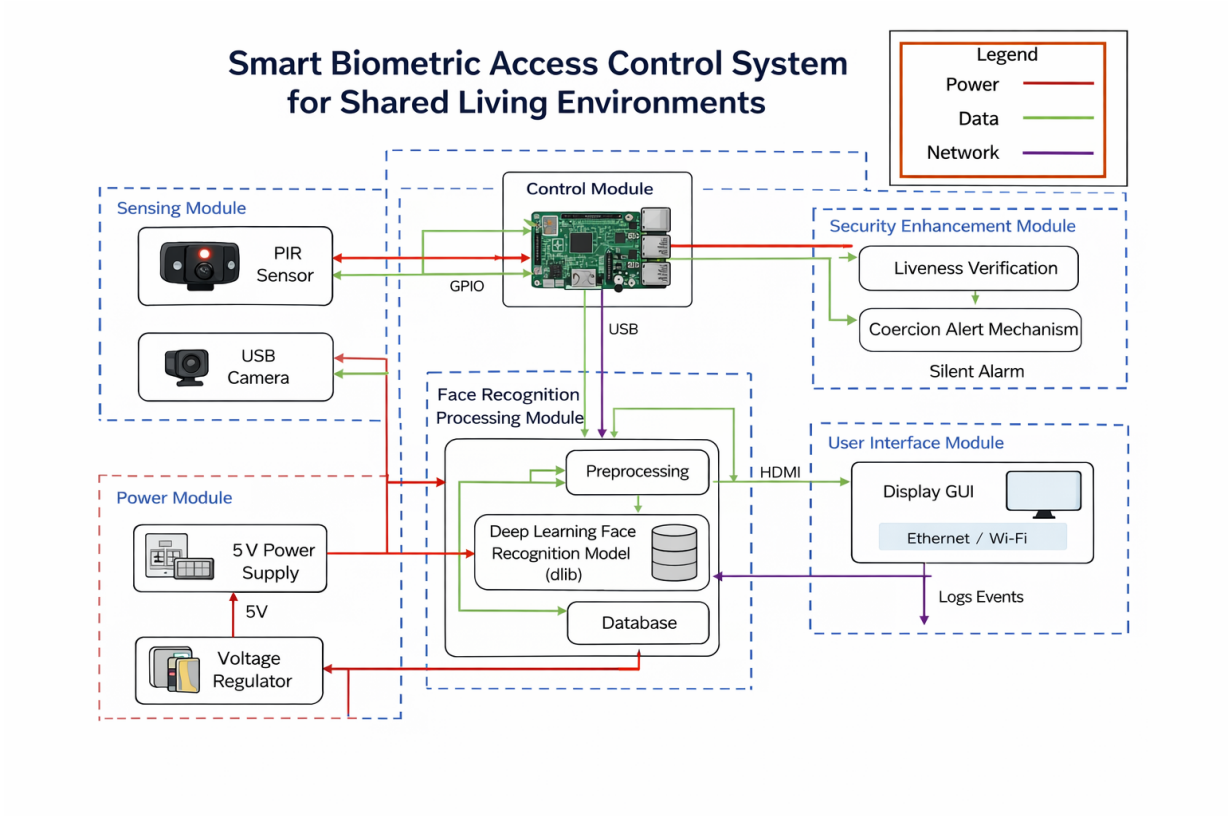

Figure 2: System block diagram of the smart biometric access control system for shared living environments. Red lines indicate power delivery, green lines indicate data flow, and purple lines indicate network or logging connections.

2.2 Subsystem 1: Hardware and Sensing

2.2.1 Function and Interaction

The Hardware and Sensing Subsystem serves as the physical interface between the environment and the digital controller. It includes the PIR motion sensor, USB camera, and relay-controlled lock actuation hardware. The PIR sensor continuously monitors for human motion near the door and generates a trigger signal to the Raspberry Pi through a GPIO pin. Once triggered, the camera captures image frames for downstream recognition. After the recognition subsystem returns a

successful authentication result, the Raspberry Pi asserts a GPIO output that activates the relay module. The relay closes the strike-lock circuit for a controlled duration, allowing the user to enter.

This subsystem interacts with the Control Subsystem through GPIO signaling, with the Recognition Subsystem through the camera data stream, and with the Power Subsystem through the regulated 5 V supply.

2.2.2 Design Rationale

A PIR sensor is used as a low-cost, low-power wake-up trigger so that the computationally expensive face recognition pipeline does not need to run continuously. A USB camera is selected because it offers higher image quality and easier driver support compared with low-end serial camera modules. An opto-isolated relay is chosen to electrically separate the low-voltage control logic from the higher-current electronic door strike.

2.2.3 Requirements

H1. The PIR sensor shall detect a human-sized moving heat source within 2.0 ± 0.2 m.

H2. The PIR output high level shall be at least 3.0 V and be recognized reliably by the Raspberry Pi GPIO.

H3. The camera shall provide the first valid frame within 400 ms after a PIR trigger.

H4. The camera shall support a minimum image resolution of 1280×720.

H5. The relay shall hold the door-unlock circuit closed for 5.0 ± 0.5 s after receiving an access-granted signal.

H6. The relay interface shall provide electrical isolation between the Raspberry Pi and the lock circuit.

2.2.4 Verification

Requirement	**Verification Procedure and Success Criterion**
H1. PIR detection range	Place a test subject at 1.5 m, 2.0 m, and 2.2 m from the sensor and record 20 trials at each distance. The sensor shall trigger in at least 19/20 trials at 2.0 m.
H2. PIR output level	Measure the PIR output with a multimeter or oscilloscope during motion events. The observed high level shall be ≥ 3.0 V.
H3. Camera startup latency	Timestamp the PIR interrupt and the first valid frame acquisition in software. The measured delay shall be ≤ 400 ms.
H4. Camera resolution	Query the camera frame format in software and verify that the resolution is at least 1280×720.

H5. Relay on-time	Issue an unlock command and measure the relay active duration with software timestamps or an oscilloscope. The relay shall remain active for 5.0 ± 0.5 s.
H6. Relay isolation	Inspect the relay module datasheet and verify opto-isolation and safe separation of control and actuation sides in the schematic.

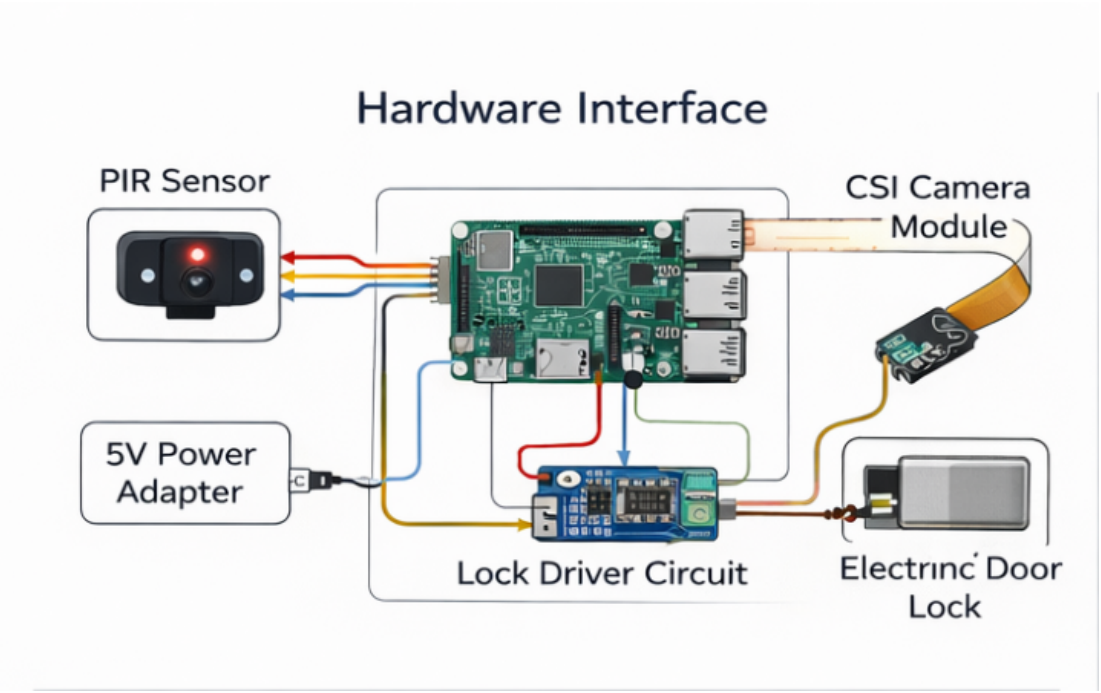

Figure 3: Hardware interface diagram showing the connections among the PIR sensor, camera module, Raspberry Pi controller, lock driver circuit, electronic door lock, and power adapter.

2.3 Subsystem 2: Image Processing and Recognition

2.3.1 Function and Interaction

The Image Processing and Recognition Subsystem is the computational core of the project. It converts raw camera frames into identity decisions. The subsystem first accepts image data from the USB camera, then performs preprocessing steps such as resizing, grayscale conversion, and histogram equalization. Next, the subsystem detects and localizes the face region, extracts facial features, and compares those features against authorized templates stored in a local database. It then outputs an identity result, confidence score, and access decision candidate to the Control Subsystem.

This subsystem interacts with the Hardware and Sensing Subsystem through the camera stream, with the UI and Management Subsystem through the user database and log database, and with the Security Enhancement Subsystem for liveness verification and exception handling. The Image Processing and Recognition Subsystem is the computational core of the project. It converts raw camera frames into identity decisions. The subsystem first accepts image data from the USB

camera, then performs preprocessing steps such as resizing, grayscale conversion, and histogram equalization. Next, the subsystem detects and localizes the face region, extracts facial features, and compares those features against authorized templates stored in a local database. It then outputs an identity result, confidence score, and access decision candidate to the Control Subsystem. This subsystem interacts with the Hardware and Sensing Subsystem through the camera stream, with the UI and Management Subsystem through the user database and log database, and with the Security Enhancement Subsystem for liveness verification and exception handling

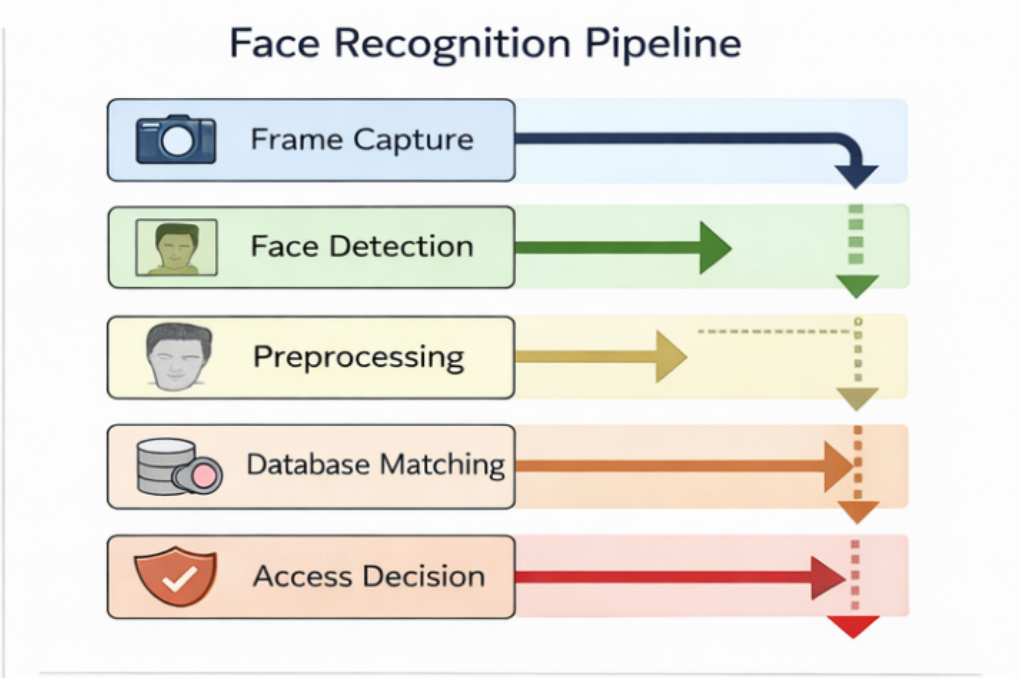

Figure 4: Software pipeline of the image processing and face recognition subsystem, including frame capture, face detection, preprocessing, database matching, and final access decision.

2.3.2 Design Rationale

The use of local face recognition avoids privacy issues associated with cloud-based biometric systems. Preprocessing is included to reduce sensitivity to indoor lighting variation. A single-face priority rule is used to limit computation and keep the system within real-time latency constraints. The comparison stage uses feature-vector similarity rather than raw image matching, which improves robustness and scalability.

2.3.3 Requirements

R1. The recognition subsystem shall produce an identity decision within 1.2 s after the first valid frame is captured under standard indoor lighting.

R2. The subsystem shall correctly identify authorized users with at least 95% accuracy under 300–500 lux lighting conditions.

R3. The subsystem shall achieve a false acceptance rate (FAR) below 0.1%.

R4. The subsystem shall achieve a false rejection rate (FRR) below 5%.

R5. The subsystem shall support a local database of at least 50 enrolled users without causing the total end-to-end unlock latency to exceed 2.0 s.

R6. The subsystem shall process only the dominant foreground face when multiple faces appear in the frame.

R7. The subsystem shall maintain recognition performance such that the accuracy difference between 300 lux and 500 lux is no more than 5 percentage points.

2.3.4 Verification

Requirement	**Verification Procedure and Success Criterion**
R1. Decision latency	Measure the elapsed time from the first valid frame acquisition to the final recognition result across 50 trials. The average shall be ≤ 1.2 s.
R2. Authorized-user accuracy	Test 5 authorized users, 20 trials each, under standard indoor lighting. At least 95 out of 100 attempts shall be correctly identified.
R3. FAR	Run at least 1000 unauthorized matching attempts using non-enrolled users. Fewer than 1 false acceptance shall occur.
R4. FRR	Repeat authorized-user trials and compute false rejections. The rate shall remain below 5%.
R5. Database scalability	Populate the database with at least 50 identities and repeat timing tests. The full access process shall remain under 2.0 s.
R6. Foreground-face filtering	Present two or more faces simultaneously and verify that only the largest or most central face is processed for authentication.
R7. Lighting robustness	Repeat recognition tests at 300 lux, 400 lux, and 500 lux. The accuracy difference across these conditions shall be no greater than 5 percentage points.

2.4 Subsystem 3: UI and System Management

2.4.1 Function and Interaction

The UI and System Management Subsystem ensures that the system is understandable, responsive, and maintainable. It includes the graphical user interface, multithreaded execution architecture, and local database services. The GUI displays system state, live video, recognized identity, access result, and recent logs. The software architecture separates the user interface thread from recognition and hardware I/O tasks so that the interface remains responsive during active authentication. The management layer stores authorized user profiles, face templates, and access logs in a local SQLite database.

This subsystem interacts with the Recognition Subsystem to obtain identity results, with the Control Subsystem to reflect lock state, and with the Security Enhancement Subsystem to display security warnings or silent alert conditions as needed.

2.4.2 Design Rationale

A dedicated GUI improves usability and also helps debugging and demonstration. Local logging is essential for accountability and management in shared living environments. Multithreading prevents interface freezing during expensive image-processing operations.

2.4.3 Requirements

U1. The GUI shall update access status within 200 ms after the recognition result is finalized.

U2. The GUI shall remain responsive during continuous operation, with no visible freeze longer than 500 ms.

U3. The system shall log 100% of access attempts with timestamp, result, and recognized identity or unknown status.

U4. The log database shall store at least 1000 entries locally.

U5. Database write time per access event shall be no more than 100 ms under normal operation.

2.4.4 Verification

Requirement	**Verification Procedure and Success Criterion**
U1. GUI update delay	Compare the recognition decision timestamp and the GUI status-change timestamp in software logs. The delay shall be ≤ 200 ms.
U2. GUI responsiveness	Operate the system continuously for 10 minutes while triggering multiple authentication events. No interface freeze longer than 500 ms shall be observed.
U3. Complete logging	Generate a set of successful and failed access attempts and verify that every attempt appears in the database.
U4. Log capacity	Insert at least 1000 entries and confirm correct storage and retrieval.
U5. Database write latency	Benchmark the time needed to write one access event to the database. The average write time shall be ≤ 100 ms.

2.5 Subsystem 4: Security Enhancement

2.5.1 Function and Interaction

The Security Enhancement Subsystem improves resistance against spoofing and forced access scenarios. It introduces a liveness verification stage and an optional secondary verification pathway. The liveness check aims to distinguish a real human face from printed photos or screen replays. If the recognition confidence is suspicious or liveness verification fails, the system may require a secondary action, such as a spoken passphrase or keypad confirmation. A coercion alert mechanism allows the user to silently flag a threat condition while preserving normal external behavior.

This subsystem receives candidate identities and confidence values from the Recognition Subsystem, sends final permission or deny decisions to the Control Subsystem, and writes security events into the Management Subsystem.

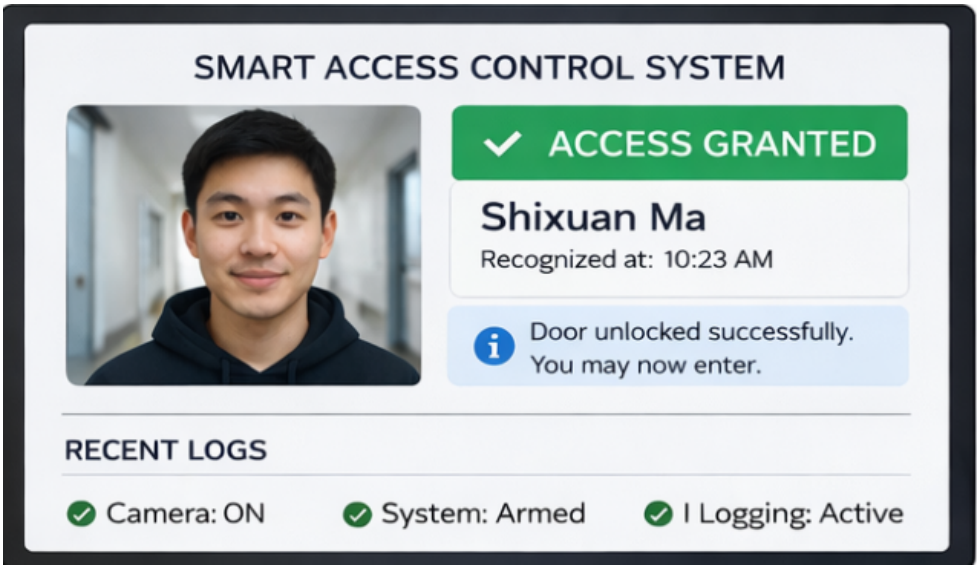

Figure 5: Graphical user interface of the system, showing recognized identity, access result, and recent system logs.

2.5.2 Design Rationale

Facial recognition alone is vulnerable to simple spoofing if no liveness check is used. In a real shared-living deployment, both convenience and safety matter, so it is beneficial to include a secondary path for suspicious cases rather than simply denying all uncertain inputs. A coercion alert function is particularly valuable because residential access scenarios may involve forced entry attempts.

2.5.3 Requirements

S1. The liveness verification module shall reject at least 90% of simple spoof attacks using printed photos or screen replays.

S2. The system shall invoke secondary verification within 500 ms after detecting a suspicious authentication event.

S3. The coercion alert mechanism shall generate a hidden log flag without visibly changing the outward interface behavior.

S4. The security enhancement features shall not increase normal successful-entry latency by more than 300 ms.

2.5.4 Verification

Requirement	**Verification Procedure and Success Criterion**

S1. Spoof rejection rate	Test the system with printed face photos and screen-replayed videos. At least 90% of spoof attempts shall be rejected.
S2. Secondary-verification response time	Trigger low-confidence or failed-liveness cases and measure the delay before secondary verification appears. The delay shall be ≤ 500 ms.
S3. Hidden coercion alert	Trigger the coercion path in a controlled test. The alert shall be recorded in the logs while the outward interface still appears normal.
S4. Latency overhead	Compare average access times with and without the security enhancement module enabled. The added delay shall be ≤ 300 ms.

2.6 Subsystem 5: Power

2.6.1 Function and Interaction

The Power Subsystem provides stable low-voltage power to all electronic modules, including the Raspberry Pi, camera, PIR sensor, relay module, and optional display. Because the Raspberry Pi and relay may both draw current peaks during authentication and unlocking, the power supply must remain stable and prevent brownout conditions.

2.6.2 Design Rationale

A regulated 5 V supply is necessary because the Raspberry Pi is sensitive to undervoltage, and unstable power may cause random resets or recognition failures. The power subsystem must also tolerate transient current demand caused by relay actuation and processor workload spikes.

2.6.3 Requirements

P1. The 5 V supply rail shall remain between 4.75 V and 5.25 V during all operating modes.

P2. The system shall not reboot, brown out, or throttle abnormally when the Raspberry Pi is under heavy CPU load and the relay is active.

P3. The peak-to-peak ripple on the 5 V rail shall remain below 100 mV during combined heavy-load operation.

P4. The full system shall operate continuously for at least 30 minutes during bench testing without thermal shutdown.

2.6.4 Verification

Requirement	**Verification Procedure and Success Criterion**
P1. Voltage regulation	Measure the 5 V rail with an oscilloscope during idle, recognition, and relay activation states. Voltage shall remain between 4.75 V and 5.25 V.

P2. System stability under load	Run a CPU stress test while repeatedly triggering the relay. The Raspberry Pi shall not reset or show undervoltage warnings.
P3. Output ripple	Measure the power rail ripple during peak load. Ripple shall be below 100 mV peak-to-peak.
P4. Thermal endurance	Run the system continuously for 30 minutes and monitor supply and processor temperatures. No shutdown or instability shall occur.

2.7 Supporting Material

To support the technical clarity of the design, the final document should include the following figures and diagrams:

- Overall system block diagram
- GPIO and relay interface schematic
- Camera and sensor wiring diagram
- Face recognition software pipeline flowchart
- GUI screenshot and status transitions
- Timing diagram from PIR trigger to door unlock
- Power regulation and distribution schematic

3 Tolerance Analysis

3.1 Critical Function Selection

The most critical subsystem function is the end-to-end authentication latency. This function directly determines whether the system meets the high-level requirement that the total delay from motion detection to relay activation must not exceed 2.0 seconds. If the latency is too large, the system will feel unresponsive and the user experience will degrade significantly even if recognition accuracy is acceptable.

3.2 Latency Model

The total latency is modeled as

$$T_{\text{total}} = T_{\text{PIR}} + T_{\text{camera}} + T_{\text{pre}} + T_{\text{extract}} + T_{\text{match}} + T_{\text{relay}},$$

where

- T_{PIR}: PIR detection and interrupt latency
- T_{camera}: camera wake-up and first valid frame latency
- T_{pre}: preprocessing time

- T_{extract}: face feature extraction time
- T_{match}: database matching time
- T_{relay}: GPIO output and relay response time

Based on expected operation on a Raspberry Pi 4B, a representative nominal estimate is:

$$T_{\text{PIR}} + T_{\text{camera}} \approx 500 \text{ ms},$$

$$T_{\text{pre}} \approx 50 \text{ ms},$$

$$T_{\text{extract}} \approx 800 \text{ ms},$$

$$T_{\text{match}} \approx 10 \text{ ms},$$

$$T_{\text{relay}} \approx 50 \text{ ms}.$$

Thus, the nominal latency is approximately

$$T_{\text{total}} \approx 500 + 50 + 800 + 10 + 50 = 1410 \text{ ms} = 1.41 \text{ s}.$$

This leaves a timing margin of

$$2.0 - 1.41 = 0.59 \text{ s}.$$

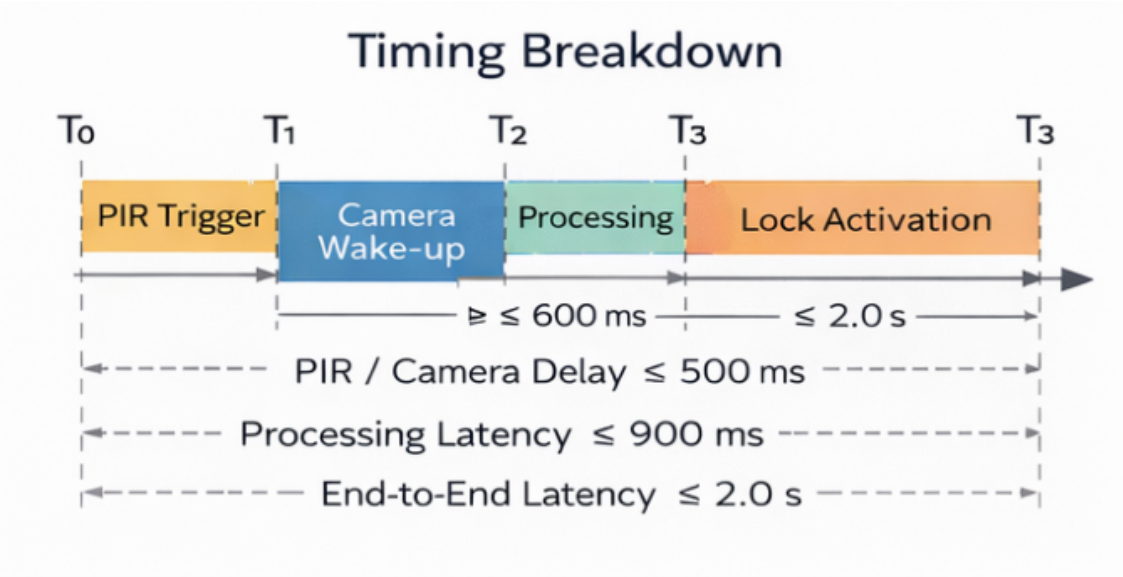

Figure 6: Timing breakdown of the end-to-end authentication process, from PIR trigger to final lock activation.

3.3 Worst-Case Tolerance Analysis

The dominant uncertainty sources are indoor lighting variation, multiple faces in the frame, and increased recognition time when the image contains more noise or blur. Under a conservative worst-case estimate:

$$T_{\text{pre}} = 80 \text{ ms}, \quad T_{\text{extract}} = 1100 \text{ ms}, \quad T_{\text{match}} = 20 \text{ ms}.$$

Assuming the PIR and camera stage remains near 500 ms and relay response remains 50 ms,

$$T_{\text{total,worst}} \approx 500 + 80 + 1100 + 20 + 50 = 1750 \text{ ms} = 1.75 \text{ s}.$$

Therefore, even under a pessimistic but realistic operating condition, the total delay remains below the 2.0 s requirement. The remaining tolerance is

$$2.0 - 1.75 = 0.25 \text{ s}.$$

3.4 Derived Design Constraints

To guarantee that the latency stays within specification, the following design constraints are imposed:

1. Only the largest or most central face in the frame will be processed for authentication.
2. Image resolution used for recognition will be capped to an efficient working size rather than full-resolution raw input.
3. If low-light conditions cause large preprocessing overhead, an auxiliary fill light can be triggered by the PIR sensor.
4. GUI updates and database writes will run in separate threads to avoid blocking the recognition pipeline.

3.5 Verification Plan

To validate the analysis, the total latency will be measured experimentally under multiple conditions:

- Standard lighting at 300 lux, 400 lux, and 500 lux
- Single-face and multi-face scenes
- Small and large enrolled-user databases

For each condition, at least 30 trials will be performed. The average, maximum, and standard deviation of T_{total} will be reported. The design will be considered verified if the maximum measured latency remains below 2.0 s and the average remains significantly below that threshold.

4 Cost

Part	**Qty**	**Unit Cost (RMB)**	**Total (RMB)**
Raspberry Pi 5 development kit (including display/accessories)	1	1282.81	1282.81
CSI camera module (OV564x)	1	24.63	24.63
HC-SR501 PIR motion sensor	1	5.74	5.74
Jumper wires and connectors	3 sets	3.84 / 3.65 / 3.65	11.14
Acrylic enclosure / camera bracket	1	12.81	12.81
Acrylic mirror / structural panel	1	80.00	80.00
Electronic door lock with driver module	1	26.00	26.00
Total			**1443.13**

Table 6: Estimated prototype cost based on purchased components.

5 Schedule

Week	Task
Week 1–2	Finalize system architecture, select components, and complete detailed block diagram and interface definitions.
Week 3–4	Integrate PIR sensor, camera, relay, and power supply with Raspberry Pi. Validate hardware I/O and unlock control.
Week 5–6	Implement image preprocessing, face detection, and baseline recognition pipeline.
Week 7–8	Integrate database management, user enrollment, and logging functions.
Week 9	Develop and refine GUI with live status display and event output.
Week 10	Add liveness verification and secondary verification / coercion alert logic.
Week 11	Perform subsystem-level verification against the requirements tables.
Week 12	Perform full-system integration, debugging, and optimization of latency and recognition accuracy.
Week 13	Complete final testing, generate plots/tables, and prepare final report and presentation.

Table 7: Proposed development schedule.

6 Ethics and Safety

6.1 Ethics

This project handles biometric information and therefore raises important ethical concerns. The design follows a privacy-preserving philosophy by ensuring that facial images, feature encodings, and logs are processed and stored locally. No biometric information is uploaded to cloud services or transmitted over external networks during normal operation. User awareness is also important; therefore, the installation should include clear notice that biometric recognition is being used.

Bias is another concern. Facial recognition systems may exhibit different performance across users with different skin tones, facial structures, or gender presentation. To reduce this risk, the system will be tested across a diverse set of users and thresholds will be evaluated to ensure that the error rates remain acceptable across different groups.

6.2 Safety

The system includes low-voltage logic electronics and an electronically controlled lock. Electrical safety will be addressed by using an opto-isolated relay interface and insulated enclosures. All wiring will be properly mounted and protected to prevent accidental shorting or damage.

From a physical safety standpoint, the system must not trap users during emergencies. Therefore, the lock design must remain compatible with building fire-code requirements, and a manual override must be available. Components such as the camera and PIR sensor must also be securely mounted to prevent falling hazards.

7 References

1. Raspberry Pi Foundation, *Raspberry Pi 4 Model B Product Brief*.
2. OpenCV Documentation, *Image Processing and Computer Vision Library*.
3. dlib Library Documentation, *Face Recognition and Landmark Detection*.
4. SQLite Documentation, *SQLite Database Engine*.
5. IEEE, *IEEE Standard for Ethically Aligned Design*.