Palestine Polytechnic University
College of Information Technology and Computer Engineering
Computer Engineering Department
Computer Systems Engineering Graduation Project

LiPass

Submitted by:

Ayed A.Areda Mohammed Qashqesh

Amro Raai

Supervisor:
Eng. Rafat Juneide

2026

Acknowledgment

All praise and thanks are due to Allah, the Almighty, whose guidance, mercy, and blessings enabled us to complete this academic journey.

After thanking Allah, we dedicate this project to our families, who stood by us with unwavering support, patience, and sacrifice throughout our years of study. Their hard work, encouragement, and belief in us were the foundation upon which this achievement was built. This project is humbly dedicated to them in appreciation of all that they have given.

We extend our sincere gratitude to Eng. Rafat Juneide for his supervision, guidance, and invaluable feedback during the development of this project. His technical expertise, constructive insights, and professional direction contributed significantly to the successful design and implementation of the system.

We also acknowledge with appreciation the support of our friends, classmates, and faculty members who contributed directly or indirectly to this work. Their cooperation, discussions, and advice enriched our academic experience. Finally, we thank everyone who offered assistance or feedback throughout the course of this project.

Table of Contents

List of Figures

List of Tables

الملخص

يُعد LiPass نظامًا آمنًا ولا تلامسيًا للتحكم في الوصول إلى الأبواب، يجمع بين المصادقة البيومترية المعتمدة على التعرف على الوجه باستخدام الهاتف المحمول وتقنية Light Fidelity (Li-Fi)، بهدف معالجة القيود الأمنية واعتبارات الخصوصية المرتبطة بأساليب التحكم التقليدية في الوصول.

تُولَّد تمثيلات الوجه (Facial Embeddings) محليًا على هاتف المستخدم الذكي باستخدام TensorFlow Liteونموذج FaceNet، في حين تتم عملية التحقق البيومتري حصريًا على الخادم الخلفي. لا يتم نقل البيانات البيومترية إلا خلال مرحلة التسجيل، وذلك عبر قنوات HTTPS/TLSالآمنة، حيث تُشفَّر باستخدام خوارزمية AES-128-GCM وتُخزَّن بشكل آمن في السحابة. أثناء المصادقة، تُفكّ البيانات البيومترية في الذاكرة المؤقتة فقط، ثم تُمحى فورًا بعد إتمام عملية التحقق.

يعتمد LiPass نموذج مصادقة متعددة العوامل يشمل بيانات اعتماد المستخدم، والتعرف على الوجه، وإشارة مصادقة قائمة على Li-Fi عبر الهاتف الذكي. يُنشئ الخادم الخلفي رمز مصادقة قصير العمر، يُنقل بصريًا باستخدام ضوء فلاش الهاتف الذكي، ثم تتحقق منه وحدة باب مبنية على Raspberry Pi لفتح الباب. وتُظهر نتائج التقييم التجريبي موثوقية عالية في التشغيل، مع متوسط زمن مصادقة يقارب خمس ثوانٍ.

Abstract

LiPass is a secure and contactless door access control system that combines mobile-based facial biometric authentication with Light Fidelity (Li-Fi) communication to address the security and privacy limitations of traditional access control methods.

Facial embeddings are generated locally on the user's smartphone using TensorFlow Lite and the FaceNet model, while biometric verification is performed exclusively on the backend server. Biometric data is transmitted only during enrollment using secure HTTPS/TLS channels, encrypted with AES-128-GCM, and stored securely in the cloud. During authentication, biometric data is decrypted only in volatile memory and immediately erased after verification.

LiPass employs a multi-factor authentication model based on user credentials, facial recognition, and a smartphone-based Li-Fi authentication signal. A short-lived authentication token is generated by the backend server, transmitted optically via the smartphone flashlight, and validated by a Raspberry Pi–based door unit to unlock the door. Experimental evaluation demonstrates reliable operation and an average authentication time of approximately five seconds.

Chapter 1

Introduction

1.1 Preface

Conventional access control systems such as passwords, physical keys, and RFID cards suffer from inherent security and usability limitations. Passwords are vulnerable to phishing, brute-force attacks, and credential reuse, while physical tokens can be lost, cloned, or stolen. These weaknesses create critical security gaps, particularly in environments that require high security assurance, hygiene, and reliability.

LiPass proposes a secure, contactless, and privacy-preserving door access solution by combining Light Fidelity (Li-Fi) communication with cloud-based biometric verification. Instead of transmitting sensitive biometric data, the system relies on short-lived, randomly generated authentication tokens transmitted using visible light, thereby eliminating the risks associated with

wireless interception and biometric leakage.

Facial biometric data is enrolled once during the initial use of the mobile application and is securely stored in encrypted form in Google Firestore using AES-128-GCM encryption. The key is securely managed on the server through protected environment variables.. The app never handles encryption keys, which is the correct security practice. During each authentication attempt, The stored biometric data is retrieved over secure, encrypted communication channels (HTTPS/TLS) and decrypted only for verification purposes.. Face recognition is performed using TensorFlow Lite with the FaceNet model, ensuring efficient on-device processing while maintaining face validation.

Upon successful biometric verification and spoof-resistance validation, the cloud generates the 128-bit token and stores it (with 15 seconds expiry). This token is transmitted via the smartphone flashlight using Li-Fi technology and validated by a Raspberry Pi, which controls the door locking mechanism.

By integrating biometrics stored on cloud, multi-factor authentication, cryptographic protection, and light-based communication, LiPass delivers a secure and intelligent access control solution suitable for hospitals, offices, and sensitive facilities.

1.2 Problem Statement

An effective door access control system must ensure strong security, user convenience, and biometric privacy simultaneously. It should support multi-factor authentication while avoiding physical contact and minimizing exposure to interception, replay attacks, and spoofing attempts.

Existing access control solutions face several challenges:

- Password-based systems: Susceptible to phishing, guessing, and social engineering attacks.
- RFID and card-based systems: Vulnerable to loss, cloning, and unauthorized reuse.
- Biometric systems: Often store data locally or transmit it insecurely, raising privacy concerns.
- Wireless communication technologies: Wi-Fi and Bluetooth signals can be intercepted, jammed, or exploited remotely.
- Cost barriers: Advanced biometric hardware remains expensive for many applications.

LiPass faces these challenges through the following features:

- Encrypted cloud-based biometric storage using Google Firestore.
- Server-side biometric verification that safeguards user privacy by never exposing raw biometric data to the client.

- Extensive anti-spoofing protection against photos, videos, masks, poor lighting, and multiple faces with liveness detection,
- Token-based authentication using randomly generated 128-bit tokens.
- Time-limited token validity (15 seconds) to prevent replay attacks.
- Physically secure Li-Fi communication confined to line-of-sight environments.
- Fail-Safe Lockout: 60-second lock after 3 consecutive biometric failures.

This layered authentication approach enhances security while preserving usability, hygiene, and affordability. All security events are logged to Firestore for auditing.

1.3 Project Aims and Objectives

1.3.1 Aim 1: Develop a Secure Li-Fi-Based Authentication Channel

Objectives:

- Design a Li-Fi communication channel for transmitting authentication tokens.
- Encode randomly generated 128-bit tokens for reliable light transmission.

- Implement accurate signal decoding using sampling and thresholding techniques.
- Prevent replay attacks through strict token expiration enforcement.

1.3.2 Aim 2: Implement Cloud-Based Biometric Enrollment and Verification

Objectives:

- Enroll facial biometric data during first-time application use.
- Securely store encrypted facial embeddings in Google Firestore using AES-128-GCM encryption.
- Retrieve and decrypt biometric data during authentication requests.

1.3.3 Aim 3: Integrate Multi-Factor Authentication with Anti-Spoofing

Objectives:

- Perform face recognition using TensorFlow Lite with the FaceNet model.
- Apply cosine similarity comparison with an acceptance threshold of 80%.
- Detect spoofing attempts including:

  - Multiple faces in the frame
  - Extreme lighting conditions
  - Presence of masks or static images
  - Absence of liveness indicators
- Grant access only after successful biometric and spoof validation.

1.3.4 Aim 4: Ensure System Reliability and Secure Door Control

Objectives:

- Synchronize authentication tokens between the mobile application and Raspberry Pi.
- Validate received Li-Fi tokens against retrieved session tokens.
- Control standard solenoid door locks upon successful authentication.
- Log all access attempts with timestamps for audit and monitoring.

1.4 Project Requirements

1.4.1 Functional Requirements

The LiPass system must fulfill the following functional requirements:

- User Enrollment: Allow users to enroll their facial biometrics securely via the mobile application during first-time use.
- Biometric Verification: Perform face recognition and anti-spoofing checks before generating authentication tokens.
- Token Generation: Generate and store short-lived 128-bit authentication tokens in the cloud with a 15-second expiry.
- Li-Fi Transmission: Transmit the authentication token from the mobile app to the Raspberry Pi using the smartphone's flashlight.
- Token Validation: Decode and validate the received token on the Raspberry Pi against the cloud-stored token.
- Door Control: Unlock the door via a solenoid lock upon successful token validation.
- Logging: Record all authentication attempts, successes, failures, and security events in Google Firestore.
- Fail-Safe Mechanism: Implement a 60-second lockout after three consecutive failed biometric attempts.

1.4.2 Non-Functional Requirements

The system must meet the following non-functional requirements:

- Performance: Achieve an average authentication time of under 10 seconds, with biometric verification completing in less than 5 seconds.

- Reliability: Maintain high accuracy in face recognition and low Li-Fi transmission error rate.
- Usability: Provide a contactless, intuitive interface for users, with minimal steps for authentication.
- Scalability: Support multiple users with multiple roles .
- Cost-Effectiveness: Utilize affordable hardware like Raspberry Pi and standard smartphones without requiring specialized equipment.
- Hygiene: Enable touchless operation to suit environments like hospitals.

1.4.3 Security Requirements

The system must implement comprehensive measures to safeguard against unauthorized access, data breaches, and malicious attacks. The security requirements are defined as follows:

- The system shall encrypt sensitive stored data, including biometric information. User passwords shall not be stored directly by the system and shall be securely managed by Firebase Authentication.
- The system shall transmit biometric data only during the enrollment phase using secure communication channels (HTTPS/TLS), encrypt the data immediately upon receipt at the server, and store it in encrypted form in the database.

- During authentication, the system shall perform biometric verification entirely on the server without transmitting biometric data over any communication channel. Decrypted biometric data shall reside only in volatile memory during verification and shall be securely erased immediately after the process completes.

- The system shall store facial biometric data in Firebase Firestore only as encrypted digital signatures (facial embeddings).

- The system shall incorporate multi-factor authentication (MFA) by combining multiple verification factors to reduce the risk of single-point authentication failures.

- The system shall require users to verify their email address before being allowed to use their account for authentication and access control.

- The system shall require users to authenticate using their email and password to initiate a session. If an active session exists, the system shall allow authentication using facial recognition.

- The system shall require successful facial recognition verification before generating any authentication signal to be transmitted via the smartphone flashlight.

- The system shall ensure that each authentication signal transmitted via the flashlight is generated only after a fresh face recognition operation.

- The system shall invalidate any generated authentication signal if it is not transmitted within 15 seconds, requiring the user to perform a new

face recognition process.

- The system shall immediately discard any pending authentication signal if the user exits the authentication page or terminates the session.
- The system shall achieve a low False Acceptance Rate (FAR) to ensure that only authorized users are granted access.
- The system shall maintain a low False Rejection Rate (FRR) to balance strong security with acceptable usability for legitimate users.
- The system shall resist spoofing and presentation attacks by implementing effective liveness and presentation attack detection mechanisms to ensure authentication attempts originate from genuine, live users.
- The system shall prevent replay attacks by employing time-sensitive or one-time authentication mechanisms that ensure captured authentication data cannot be reused maliciously.
- The system shall mitigate brute-force and guessing attacks by temporarily blocking authentication API requests after a limited number of consecutive failed authentication attempts.
- The system shall securely log all authentication events to support auditing, forensic analysis, and detection of suspicious or abnormal activities.
- The system shall avoid local storage of sensitive data on client or edge devices to reduce the risk associated with device loss, theft, or compromise.

- The system shall enforce architectural separation between the backend server and the database such that a compromise of one component does not grant direct access to the other.

- **User Access Control:**

  - The system shall allow users to update their personal information, including changing their email, password, and name.
  - The system shall enable administrators to add new users who are authorized to access the door.
  - The system shall allow administrators to disable or enable user accounts under their responsibility.
  - The system shall restrict access to authentication logs, allowing them to be viewed only by the door manager.

- **Login Features:**

  - The system shall require users to log in using their email and password.
  - Users passwords shall be at least 8 characters long and include at least one uppercase letter, one lowercase letter, and one special symbol.
  - User passwords shall be securely managed by Firebase Authentication.
  - All authentication-related communications shall use secure HTTPS/TLS channels.

- The system shall use encryption to protect sensitive authentication and biometric data.
- The system shall support facial recognition login to allow users to access authenticated sessions.
- The system shall provide a "forget password" feature to enable secure password recovery.

1.5 System Description

LiPass operates through a multi-stage authentication workflow that separates biometric verification from access authorization, ensuring both privacy and security.

During the initial enrollment phase, the mobile application captures the user's facial then generates face embedding data using FaceNet model. This biometric data is encrypted and securely stored in Google Firestore. No biometric data is stored locally on the smartphone or the Raspberry Pi.

During an authentication attempt, the following sequence occurs:

1. The mobile application captures a live facial image.
2. Encrypted biometric data is retrieved from Firestore and decrypted on the server.
3. The decrypted facial embedding is compared with the newly captured embedding using cosine similarity.

4. Authentication proceeds only if the similarity score is at least 80% and no spoofing indicators are detected.

5. The cloud generates the 128-bit token and stores it (with 15s expiry).

6. The token, along with a timestamp and a 15-second validity window, is sent to both the mobile application and the Raspberry Pi.

7. The mobile application transmits the token via Li-Fi using the smartphone flashlight.

8. The Raspberry Pi receives and decodes the light signal.

9. The decoded token is compared with the authentication token generated by Firebase.

10. If both tokens match within the allowed time window, door access is granted.

A key advantage of this architecture is that no biometric information is transmitted during the unlocking process. Li-Fi is used exclusively for short-lived authentication tokens, and its physical confinement prevents remote interception.

All authentication attempts are securely logged with timestamps and user identifiers. By combining encrypted cloud biometrics, spoof detection, time-bound token authentication, and light-based communication, LiPass delivers a secure, hygienic, and intelligent access control solution.

1.6 Project Limitations/Constraints

The LiPass system is subject to the following limitations and constraints:

- Hardware Dependencies: Requires a smartphone with a functional flashlight and camera.

  Platform Compatibility: The mobile application currently supports the Android operating system only; other platforms (e.g., iOS) are not supported.

- Environmental Conditions: Li-Fi transmission requires line-of-sight and may be affected by ambient light interference, extreme darkness, or rapid movements. Not suitable for outdoor use in bright sunlight.

- Network Reliance: Depends on internet connectivity for cloud-based biometric storage and token synchronization. Offline mode is not supported.

- Scalability Limits: Current implementation supports single-door setups; multi-door scaling requires additional Raspberry Pi units and cloud resources.

- Power and Connectivity: Designed for facilities that can maintain continuous power supply, as the Raspberry Pi and door lock require electricity, and the system needs stable internet; not suitable for off-grid or power-unstable environments.

1.7 Project Schedule

The project was developed over a period of 6 months, following a structured timeline. The schedule is outlined in the table below:

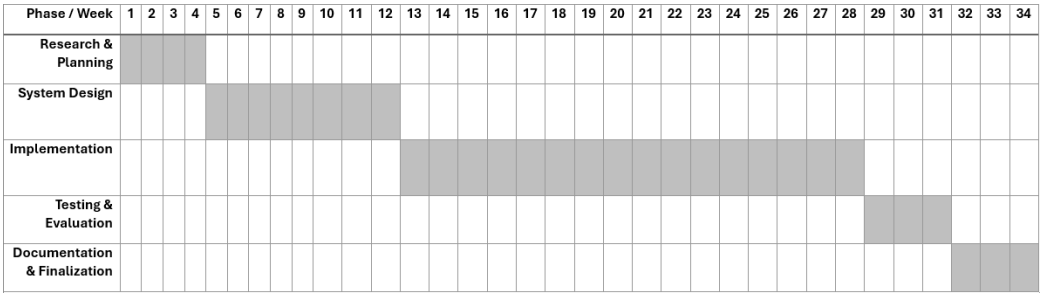


Figure 1.1: Project Schedule

This figure presents the timeline of the LiPass project, divided into phases such as research, system design, implementation, testing, and documentation. It illustrates the structured workflow across 34 weeks, showing how tasks were distributed to ensure systematic progress and timely completion.

1.8 Report Outline

This report is structured as follows to provide a comprehensive overview of the LiPass project:

- **Chapter 1: Introduction** - Provides an overview of the project, problem statement, aims, objectives, requirements, system description, limitations, schedule, and report outline.
- **Chapter 2: Literature Review** - Surveys existing access control systems, biometric technologies, Li-Fi communication, and related security mechanisms.

- **Chapter 3: System Design** - Details the architectural design, components, data flows, and technical specifications.
- **Chapter 4: Implementation** - Describes the development process, code structures, and integration of hardware/software elements.
- **Chapter 5: Testing and Results** - Presents evaluation methods, experimental results, performance metrics, and analysis.
- **Chapter 6: Conclusion and Future Work** - Summarizes findings, discusses implications, and suggests enhancements.
- **References** - Lists all sources and citations used in the report.

The report aims to document the project's methodology, achievements, and potential for real-world application.

Chapter 2

Theoretical Background

2.1 Preface

This chapter presents the theoretical foundations and enabling technologies behind LiPass. It introduces the principles of Light Fidelity (Li-Fi), neural-network-based face recognition, multi-factor authentication, embedded systems, and cloud-based identity management. These concepts form the scientific and engineering basis of the proposed solution and are essential for understanding how LiPass delivers secure, contactless, and privacy-preserving door access.

2.2 Theoretical Background Concepts

This section discusses the core theoretical concepts that support the design and implementation of LiPass. Together, these technologies enable a modern access-control architecture that combines physical-layer security, biometric verification, and cryptographic protection.

2.2.1 Light Fidelity (Li-Fi)

Light Fidelity (Li-Fi) is a wireless communication technology that transmits data using visible light by modulating the intensity of light-emitting diodes (LEDs) at high speeds [1]. Unlike radio-frequency technologies such as Wi-Fi and Bluetooth, visible light cannot penetrate walls, providing natural physical limitation and resistance to remote interception [1].

In LiPass, Li-Fi is employed as a secure authentication channel rather than a high-throughput communication medium. After successful identity verification, a smartphone flashlight transmits a short-lived authentication token encoded as light intensity variations. A photodiode connected to a Raspberry Pi receives and decodes this optical signal to authorize door access.

A similar flashlight-based Li-Fi communication approach was demonstrated in [2], where the authors state:

> the transmitted data is encoded as light intensity variations produced by the smartphone flashlight and detected by a photodiode

receiver." [2]

Smartphone Hardware Limitations (20–30 Hz)

A key design constraint is the modulation capability of smartphone flashlight hardware. These LEDs are not designed for high-speed optical communication. As reported in a Sensors 2020 study [3]:

> for most smartphone cameras, it is less than 30 Hz" [3]

and:

> "smartphone cameras sample at about 30 Hz, which is below the level of most suggestions." [3]

As smartphone flashlight modules operate under similar timing constraints, practical Li-Fi modulation rates are limited to the tens-of-hertz range. This bandwidth is sufficient for transmitting short authentication tokens securely. The system uses a busy-wait loop (Stopwatch) to achieve microsecond timing, because standard OS timers are too imprecise.

2.2.2 Face Recognition Using Neural Networks

Facial recognition is a biometric authentication technique that verifies an individual based on unique facial characteristics. Modern face recognition systems rely on deep neural networks (DNNs), which significantly outperform traditional feature-based approaches in accuracy and robustness.

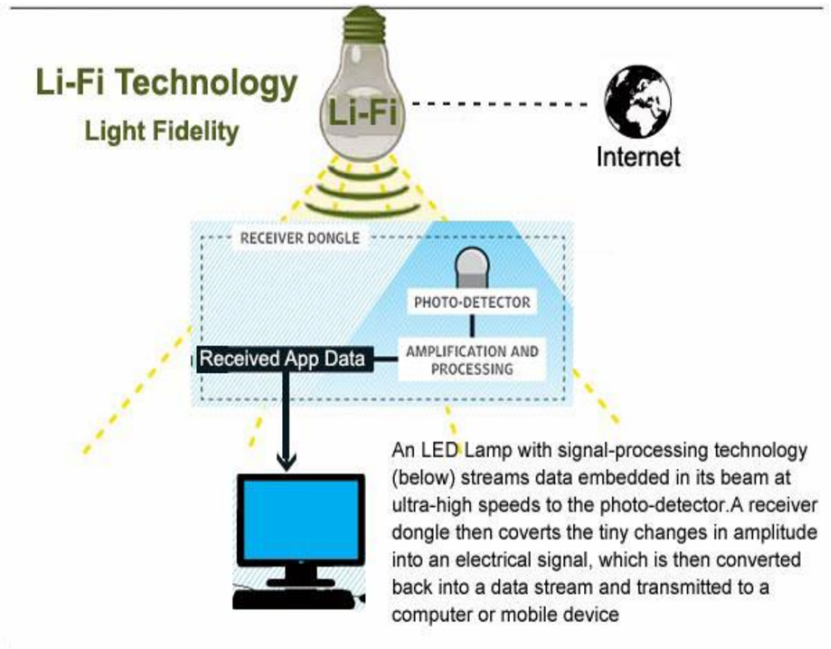


Figure 2.1: Li-Fi–based authentication

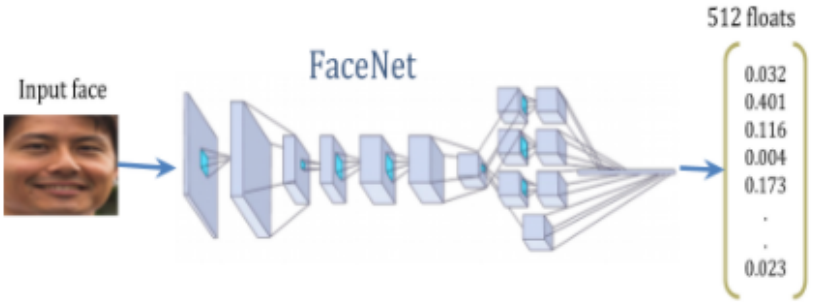


Figure 2.2: Neural-network-based face Recognition workflow in LiPass

In LiPass, face recognition is performed on the smartphone using TensorFlow Lite. Lightweight, mobile-optimized models such as FaceNet embedding networks are used to enable efficient on-device inference **??**. Facial biometric data is enrolled once during initial application usage and stored in encrypted form in a cloud database using AES-128-GCM encryption. The app never handles encryption keys, which is the correct security practice.

FaceNet learns to map facial images into a compact embedding space where similar faces are close together and dissimilar faces are far apart. This is achieved using the **triplet loss function**, which enforces a margin α between positive and negative pairs:

$$\|f(a) - f(p)\|_2^2 + \alpha < \|f(a) - f(n)\|_2^2 \tag{2.1}$$

where α represents a margin that separates positive and negative pairs.

To optimize this embedding space, the network minimizes the triplet loss function defined as:

$$L = \sum_i \left[\|f(a_i) - f(p_i)\|_2^2 - \|f(a_i) - f(n_i)\|_2^2 + \alpha \right]_+ \tag{2.2}$$

This training approach allows FaceNet to produce compact embeddings that can later be used for verification and recognition tasks.

Face Recognition Workflow

1. Face detection using DNN-based models **?**.
2. Facial feature embeddings are extracted using the FaceNet model.
3. Liveness detection, mask detection, lighting checks, and multiple faces enforcement.
4. Retrieval and decryption of enrolled facial embeddings from cloud storage.

5. Similarity comparison using cosine similarity with an acceptance threshold of 80%.

2.2.3 Multi-Factor Authentication (MFA)

Multi-Factor Authentication (MFA) enhances security by requiring multiple independent authentication factors [7]. In LiPass, authentication relies on:

- **Something you know** (e.g., a password or PIN),
- **Something you have** (e.g., a smartphone or key card),
- **Something you are** (e.g., a fingerprint or facial features).

In LiPass, MFA is implemented through:

- **Li-Fi signal** – something you have (your smartphone),
- **Facial recognition** – something you are.
- **Account password and email** – something you know.

This combination significantly reduces unauthorized access risks [7][8].

2.2.4 Embedded Systems and Microcontrollers

The Raspberry Pi acts as the primary embedded controller in LiPass. It receives and decodes Li-Fi signals, validates authentication tokens, and ac-

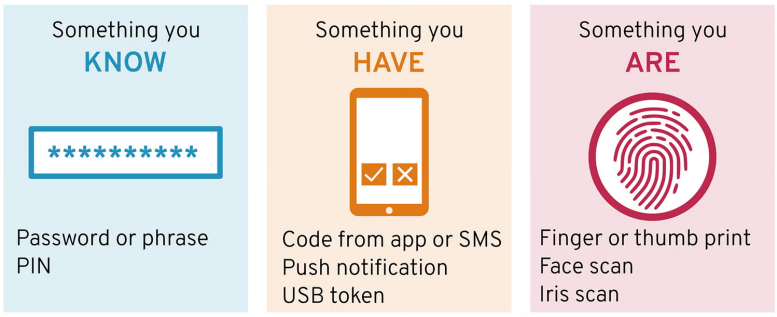


Figure 2.3: Multi-factor authentication model used in LiPass

tuates the door locking mechanism. Facial recognition tasks are offloaded to the smartphone to reduce computational load on the embedded device.

The Raspberry Pi's GPIO architecture enables direct interfacing with photodiodes, relays, and solenoid locks, making it suitable for real-time access-control applications [9].

2.2.5 Cloud Storage and Secure Communication

Cloud integration enables secure biometric storage, centralized authentication management, and access logging. LiPass uses Google Firestore to store encrypted facial embeddings and authentication metadata.

All communications are protected using (HTTPS/TLS), while encryption mechanisms ensure confidentiality and integrity of stored records [11]. All security events are logged to Firestore for auditing.

Feature / Project	**Smart Door System (Raspberry Pi)**	**Smart Entrance**	**LiPass**
Main Goal	Basic door security and visitor identification	Unauthorized entry detection and alerts	Secure, contactless, and privacy-preserving door access using Li-Fi and MFA
Authentication Methods	RFID and basic face recognition	RFID only	Cloud-verified face recognition with liveness detection and Li-Fi token authentication
Wireless Communication	Wi-Fi (MQTT protocol)	Wi-Fi / Bluetooth	Li-Fi as primary authentication channel, with Wi-Fi / Bluetooth / USB for configuration
Face Recognition	Basic local comparison	Not supported	TensorFlow Lite-based recognition with encrypted cloud-stored embeddings
Replay Attack Protection	Not addressed	Not addressed	Random 128-bit tokens with strict 15-second validity window
Anti-Spoofing	Limited or absent	Not implemented	Liveness detection, mask detection, lighting checks, and multiple-face enforcement
Logging and Monitoring	Local visitor logs	Basic alert notifications	Secure cloud-based logging with timestamps and user identity

Table 2.1: Comparison of Smart Door Access Control Projects

2.3 Literature Review

Compared to earlier projects such as the Smart Door System and Smart Entrance, LiPass introduces a more advanced and security-focused access control architecture. Traditional solutions rely heavily on RFID, Wi-Fi communication, or basic biometric techniques, which expose them to cloning,

interception, and spoofing attacks.

LiPass separates biometric verification from access authorization by transmitting only short-lived authentication tokens via a physically confined Li-Fi channel. Biometric data remains encrypted within the cloud and is never transmitted during door unlocking. The use of time-bound tokens significantly reduces replay attack risks.

Furthermore, the integration of multi-factor authentication, liveness detection, mask detection, lighting checks, multiple faces enforcement, and comprehensive logging enhances system resilience and accountability. These features collectively distinguish LiPass as a secure, contactless, and intelligent door access solution.

2.4 Summary

This chapter presented the theoretical foundations of LiPass, including Li-Fi communication principles, neural-network-based facial recognition, multi-factor authentication, embedded system integration, and cloud-based identity management. The literature review demonstrated how LiPass improves upon existing solutions by providing stronger security, enhanced privacy protection, and contactless operation.

Chapter 3

System Design

3.1 Preface

This chapter presents the system design of LiPass. The design phase focuses on defining the system structure, identifying hardware and software components, and explaining how these components interact to achieve secure and contactless door access. Several design alternatives were evaluated to ensure optimal performance, reliability, and security, and the final architecture was selected based on system requirements and practical constraints. The chapter is organized according to the approved project outline and includes system components and design alternatives, conceptual system description, algorithms and methodologies, schematic representation, and a concluding summary.

3.2 System Components and Design Alternatives (Software and Hardware)

This section describes the major hardware and software components required for LiPass. For each function, multiple alternatives were evaluated before selecting the most suitable option based on performance, cost, scalability, and security considerations.

3.2.1 Hardware Components

LiPass requires a processing unit to handle authentication logic, signal decoding, and door control. A Li-Fi receiver is needed to capture optical signals, along with interfacing and actuation components to control the door lock.

Component	**Option A (Chosen)**	**Option B**	**Option C**	**Reason for Selection**
Main Controller	Raspberry Pi 4	Raspberry Pi 3	Jetson Nano	Provides sufficient processing power for authentication, Li-Fi decoding, GPIO control, and cloud communication with good cost efficiency.
Li-Fi Receiver	BPW34 Photodiode	LDR Sensor	IR Receiver Module	Offers fast response time and high sensitivity to visible light, making it suitable for Li-Fi-based authentication.
ADC Interface	MCP3008	ADS1115	MCU Built-in ADC	Required due to lack of native analog input on Raspberry Pi; MCP3008 provides stable SPI communication.
Switching Circuit	Relay Module	Transistor Switch	Solid State Relay	Ensures electrical isolation and safe operation of the solenoid lock.
Door Lock	12V Solenoid Lock	Magnetic Lock	Servo Lock	Provides strong mechanical security and simple relay-based control.

Table 3.1: Hardware Components and Design Alternatives

3.2.2 Additional Hardware Components

- MD1601A 16×2 LCD Display Module with I2C model
- XL6009 Boost Converter
- LEDs
- Buzzer
- Wires
- Resistors
- Breadboard

3.2.3 Software Components

The software architecture of LiPass is divided between the mobile application, embedded controller, and cloud services. Each software module was selected to balance performance, security, and development flexibility.

Function	Chosen Option	Alternative 1	Alternative 2	Reason for Selection
Face Recognition	FaceNet (TFLite)	MobileNet	OpenCV LBPH	Provides high accuracy and robust facial embeddings suitable for secure authentication.
Mobile Application	Flutter (Dart)	MIT App Inventor	React Native	Enables full camera and flashlight control with cross-platform support.
Communication Protocol	HTTPS/TLS (REST API)	MQTT over Wi-Fi	BLE	Ensures secure, standardized communication over HTTPS/TLS with cloud services.
Backend Provides full control over authentication logic, security enforcement, and system integration.	Custom Server (Node.js	Express)	Firebase Functions	Django REST
Database	Firebase Firestore	Local Database	MySQL Server	Offers scalable, encrypted, and managed NoSQL storage with real-time synchronization.
Email Service	SendGrid	SMTP Server	Firebase Email Extensions	Enables reliable, secure, and scalable email delivery for verification and notifications.
Embedded Logic	Python (Raspberry Pi)	Arduino C	NodeMCU Lua	Allows high-level processing and easy integration with hardware interfaces.

Table 3.2: Software Components and Design Alternatives

3.3 Conceptual System Description

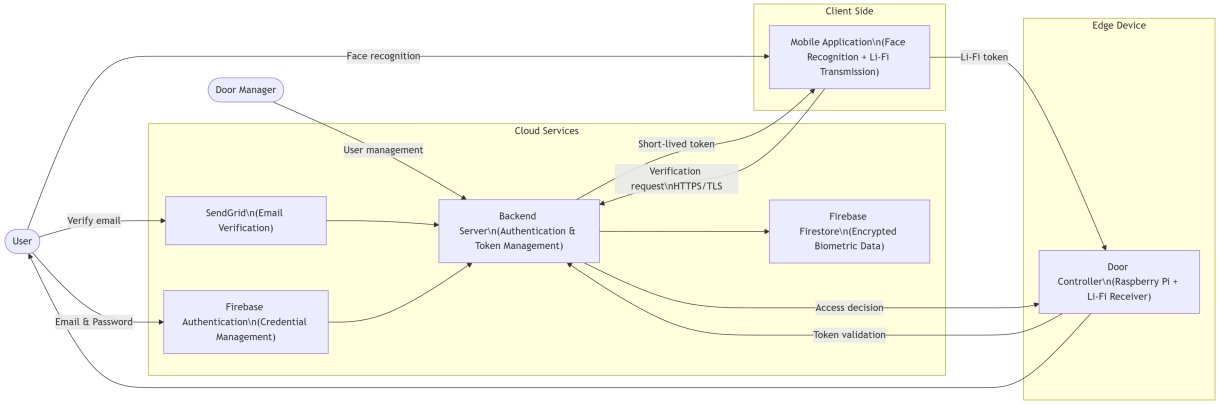


Figure 3.1: Conceptual Diagram of LiPass

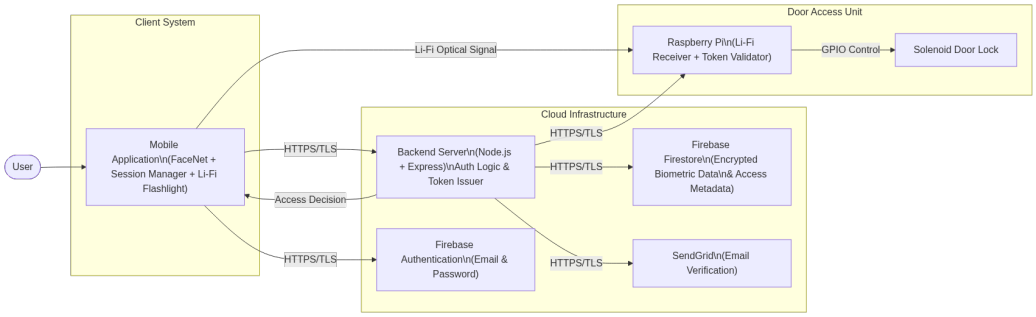


Figure 3.2: System Block Diagram of LiPass

3.4 Algorithms and Methodologies

The Li-Fi payload structure consists of 160 bits in total, divided into a 32-bit Timestamp and a 128-bit Token. The timestamp is crucial for the Raspberry Pi to prevent replay attacks without requiring cloud contact.

Signal Thresholding and Bit Decoding

The receiver samples the incoming Li-Fi signal using an Analog-to-Digital Converter (ADC). In order to convert the analog voltage values into binary bits, a hysteresis thresholding mechanism is applied. This decision process reduces noise sensitivity and unstable transitions. The hysteresis decoding rule is defined in Equation 3.1:

$$b(t) = \begin{cases} 1 & \text{if } V(t) > T_{high} \\ 0 & \text{if } V(t) < T_{low} \\ b(t-1) & \text{otherwise} \end{cases} \tag{3.1}$$

To further improve reliability, each transmitted bit is sampled three times around its expected center. A majority voting rule is then used to determine the final decoded bit. This decoding mechanism is expressed in Equation 3.2:

$$b_k = \begin{cases} 1 & \text{if votes} \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Frame Synchronization Using Timestamp Pattern

Before receiving the authentication token, the system must synchronize with the incoming Li-Fi frame. This is achieved by continuously comparing the received bit window with a predefined timestamp start pattern. Token reception begins only when the synchronization condition in Equation 3.3 is satisfied:

$$\text{Frame Start Detected if: } window = start_bits \tag{3.3}$$

Face Authentication Decision

After token reception, user verification is performed through facial recognition. Face embeddings are extracted using FaceNet, and authentication is based on cosine similarity between the live captured embedding and the stored reference embedding. The cosine similarity between two embedding vectors and is computed using Equation 3.4:

$$\cos(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} \tag{3.4}$$

A user is accepted only if the similarity exceeds the chosen authentication threshold. In this work, the acceptance condition is defined in Equation 3.5:

$$\text{Accept if } \cos(u, v) \geq 0.8 \tag{3.5}$$

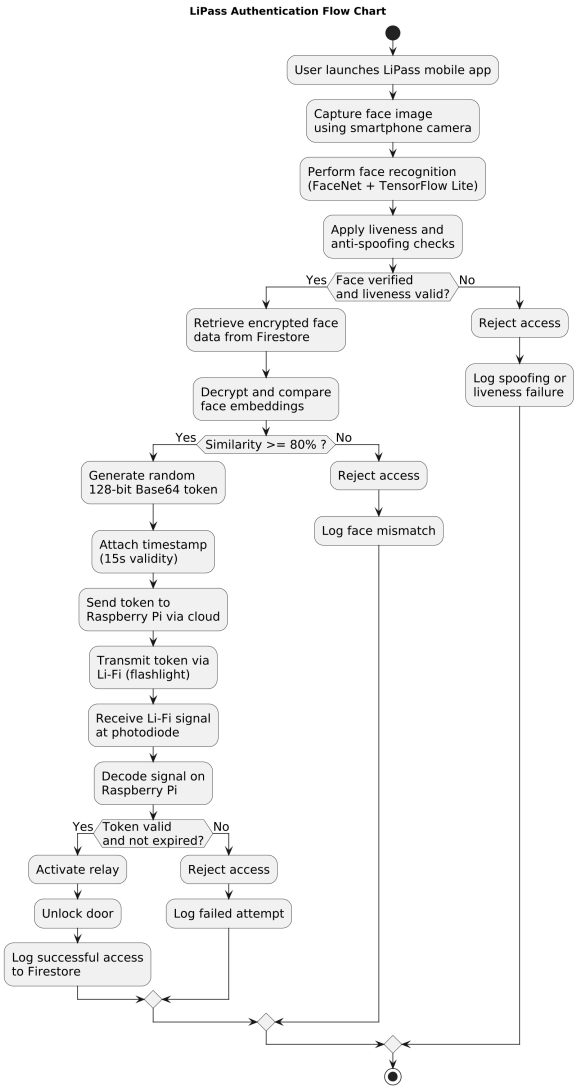


Figure 3.3: Authentication Flow Chart of LiPass

This flowchart outlines the step-by-step process of user authentication in LiPass, from face capture and verification to token generation and Li-Fi transmission, ending with door unlocking and logging of access events.

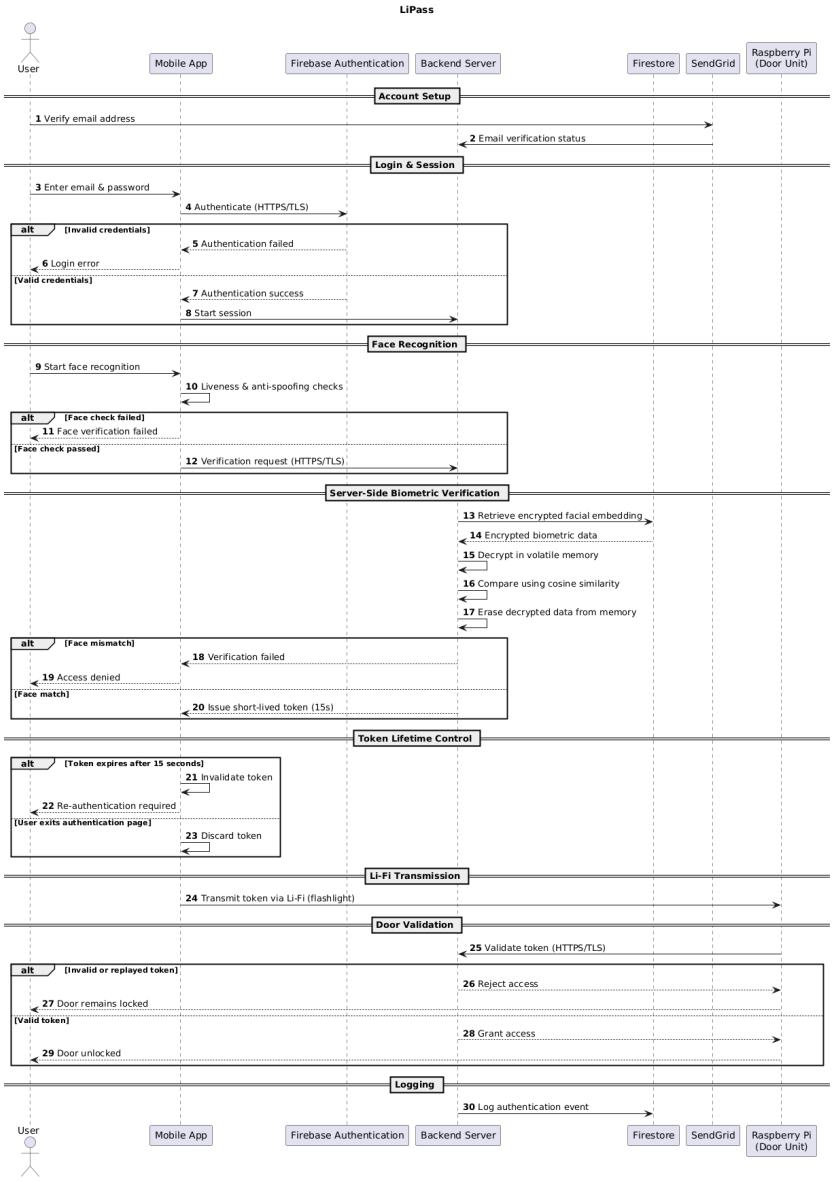


Figure 3.4: Authentication Sequence Diagram of LiPass

This sequence diagram details the interactions between system components during the authentication process, including face recognition, token generation, Li-Fi transmission, and door control. It emphasizes the secure and time-sensitive nature of each step.

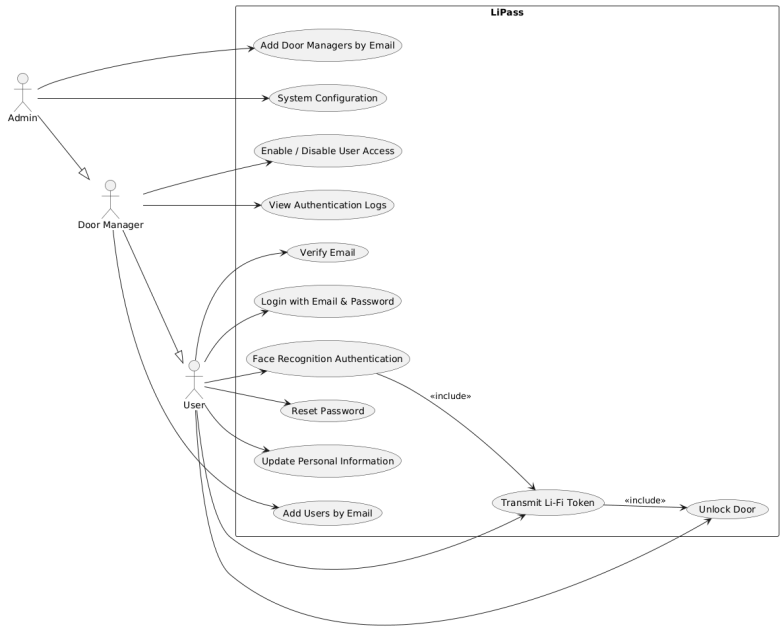


Figure 3.5: Use Case Diagram of LiPass

This diagram illustrates the roles and responsibilities of different users in the LiPass system, including regular users, door managers, and administrators. It shows how each actor interacts with system functions such as authentication, user management, and access logging.

3.5 Schematic Diagrams

The hardware connections of the LiPass system are shown in the detailed schematic below.

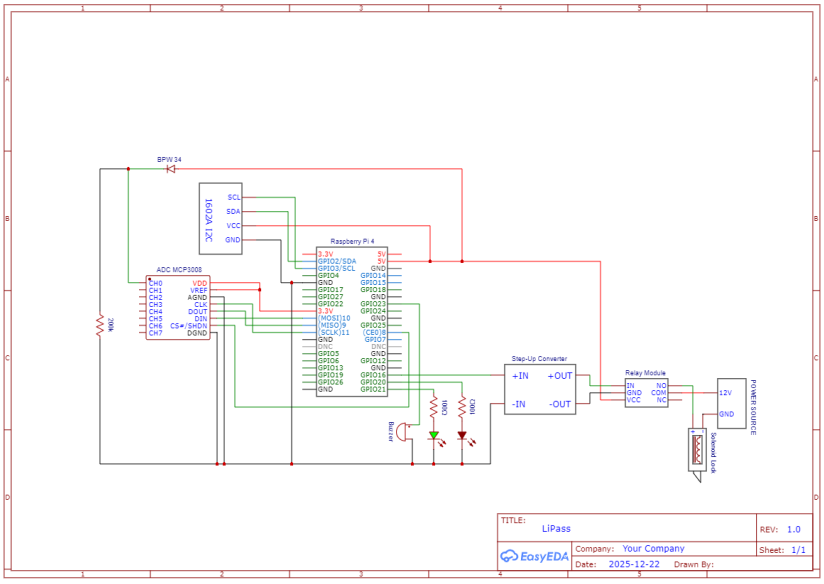


Figure 3.6: Hardware Schematic Diagram of LiPass

This schematic shows the hardware setup of LiPass, including the Raspberry Pi, LEDs, photodiode, LCD, and solenoid lock. It demonstrates how these components are interconnected to enable secure door control through Li-Fi token validation and biometric verification.

3.6 Summary

This chapter presented the complete system design of LiPass. Hardware and software components were identified and evaluated against alternative options before selecting the most suitable design. Conceptual and block diagrams illustrated the system architecture, while flow charts, sequence diagrams, and use case diagrams described the authentication logic and system interactions. Finally, the schematic diagram provided a detailed representation of hardware connections. Together, these elements form a solid design foundation for the implementation of LiPass.

Chapter 4

Implementation and Testing

4.1 Implementation Issues

This section presents the practical implementation of the LiPass system. The implementation phase focused on transforming the proposed design into a functional and secure access control solution by integrating mobile application development, embedded hardware, Li-Fi communication, and cloud-based services. Particular attention was given to enforcing security-related design decisions such as server-side biometric verification, short-lived authentication tokens, encrypted biometric storage, and controlled API access.

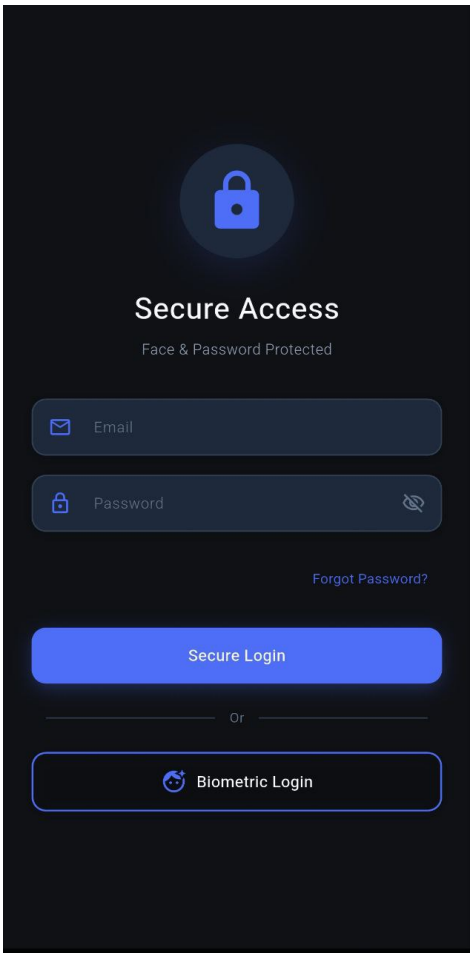
4.1.1 Mobile Application Implementation

The LiPass mobile application was implemented using the Flutter framework. The application handles facial image capture and face recognition using the FaceNet model, in addition to multiple anti-spoofing measures including liveness detection, mask detection, lighting condition validation, and multiple-face enforcement.

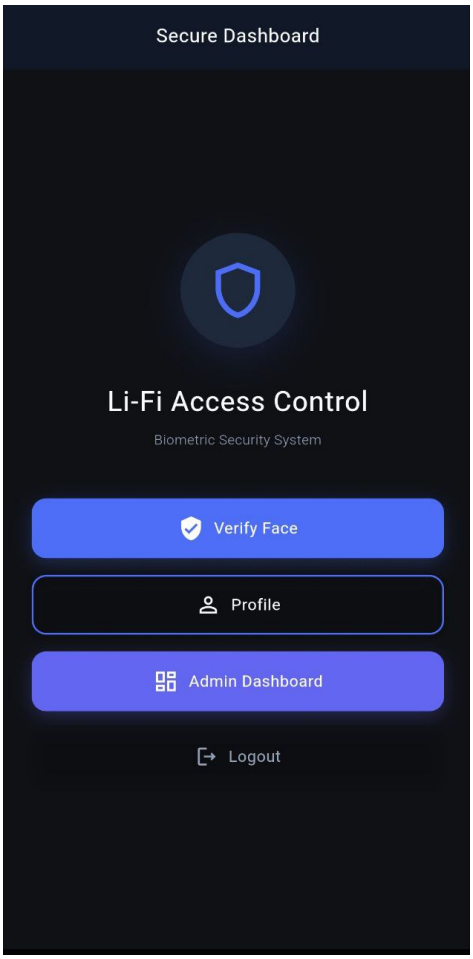
After successful on-device face recognition, the application communicates with the cloud backend over secure HTTPS/TLS channels to request authentication verification. Authentication tokens are generated exclusively by the backend server and returned to the application as short-lived tokens with a strict validity window. These tokens are transmitted optically using the smartphone flashlight via Li-Fi.

To enhance security and system stability, request-control and rate-limiting mechanisms are applied at both the application and backend levels. These mechanisms limit repeated authentication, token usage, enrollment, and administrative actions to reduce brute-force attacks, replay attempts, and API abuse. A summary is provided in Table 4.1.

The mobile application does not store biometric data or authentication tokens persistently. Any issued token is automatically invalidated if it is not transmitted within the allowed time window or if the user exits the authentication page, providing effective protection against replay and misuse.

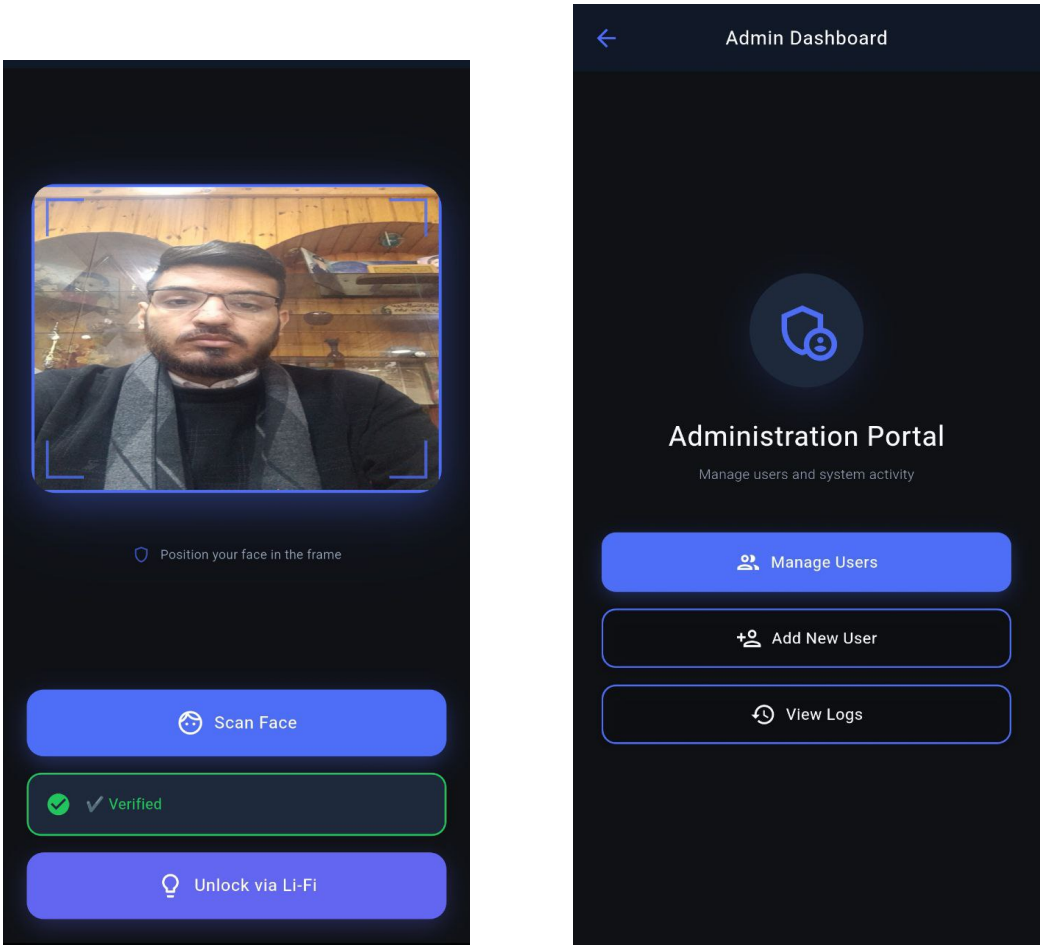


(a) Secure login interface



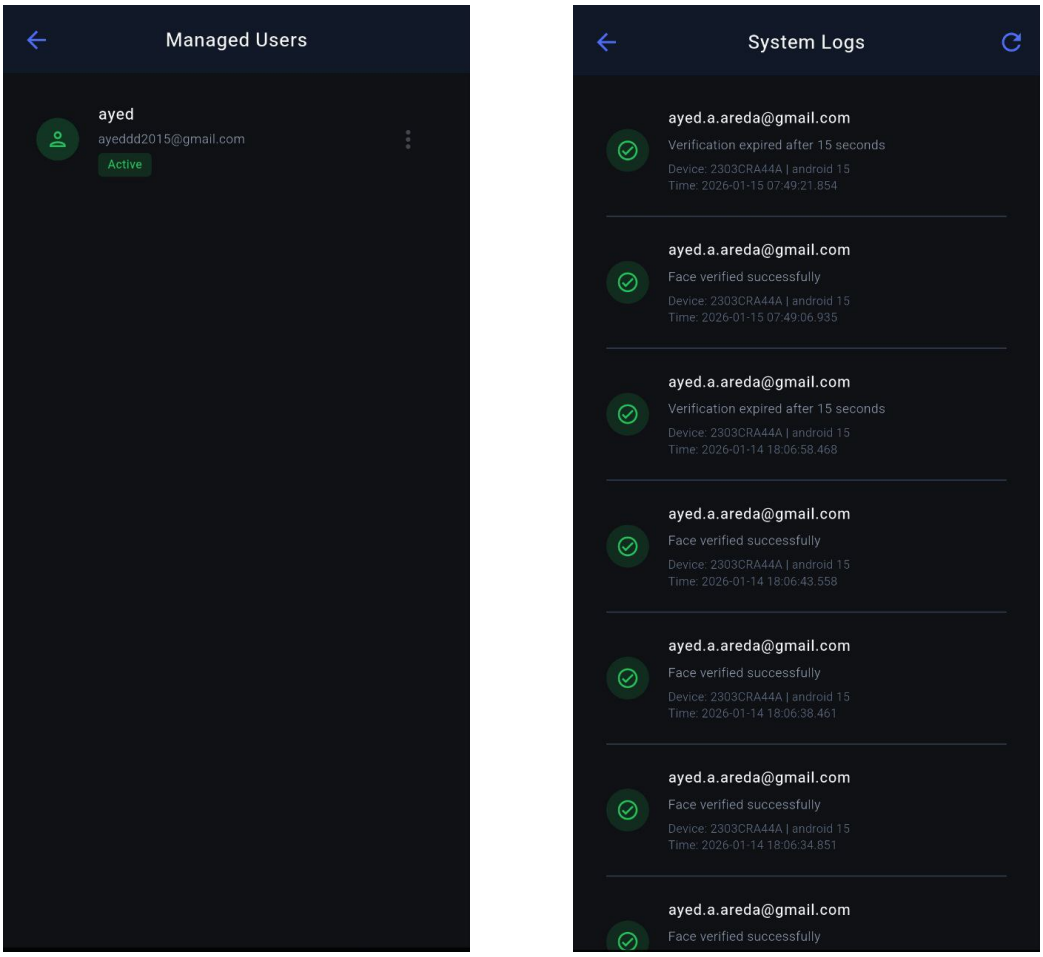
(b) Main user dashboard

Figure 4.1: Primary authentication and user dashboard interfaces



(a) Facial verification interface (b) Administrative dashboard

Figure 4.2: Biometric verification process and administrative control interface



(a) User management interface (b) System logs interface

Figure 4.3: Administrative user management and authentication activity logs

Scope	**Limit**	**Action**
Face verification failures	3 attempts	Account locked (60 s)
Password reset requests	3 / 15 min	Temporarily blocked
Face login requests	2 / 30 s	Request blocked
Verification token usage	15 s	Token expires
Global IP requests	20 / 2 min	IP throttled
User API activity	5 / 30 s	Request throttled
Face enrollment	3 / 1 min	Enrollment blocked
Admin account creation	10 / 1 min	Request blocked

Table 4.1: Rate Limiting and Request Control Mechanisms

4.1.2 Hardware Implementation

The hardware implementation is centered around a Raspberry Pi 4, which acts as the embedded door control unit. A BPW34 silicon photodiode is used to receive Li-Fi signals and is connected to an MCP3008 analog-to-digital converter for accurate signal sampling.

The Raspberry Pi decodes the received optical signal and communicates with the backend server over HTTPS/TLS to validate the authentication token. No biometric data or authentication tokens are stored locally on the embedded device. Upon successful validation, the Raspberry Pi controls a relay module that activates a 12V solenoid lock to unlock the door. Upon successful validation, the Raspberry Pi controls a relay module that activates a 12V solenoid lock to unlock the door.

Figure 4.4: LiPass hardware prototype showing the secured door enclosure with integrated LEDs, LCD, photodiode, and Raspberry Pi.

4.1.3 Cloud and Communication Implementation

Cloud services were implemented using a custom backend integrated with Google Firebase. Encrypted facial biometric data, user roles, access permis-

sions, and system logs are stored in Google Firestore, while user authentication is handled by Firebase Authentication.

All communication between the mobile application, backend server, Raspberry Pi, and Firebase services is secured using HTTPS with TLS. Facial biometric data is protected using AES-128-GCM encryption, which provides both data confidentiality and integrity with low computational overhead, making it suitable for server-side processing.

To improve system security, rate limiting and request control mechanisms were applied at the backend. These limit repeated authentication attempts, password resets, face verification requests, and administrative actions within defined time windows. When limits are exceeded, temporary blocking is enforced to prevent brute-force attacks and API abuse. All security-related events are logged in Firestore for monitoring and auditing.

4.2 Implementation Challenges

4.2.1 Power Supply and Voltage Limitations

One of the main challenges encountered was related to power delivery for the solenoid lock. The Raspberry Pi GPIO pins provide an output voltage of 3.3V, which is insufficient to directly power the solenoid. A voltage boost converter and relay-based switching circuit were introduced to resolve this issue, requiring careful configuration to achieve stable and safe operation.

4.2.2 Li-Fi Signal Detection Challenges

Several light-sensing components were evaluated during implementation, including an LDR sensor, a phototransistor, and an infrared receiver module. These components failed to provide reliable Li-Fi signal detection due to slow response times and sensitivity limitations. The issue was resolved by adopting a BPW34 silicon photodiode, which offered faster response and higher sensitivity to visible light.

4.2.3 Optical Signal Decoding and Error Rate

Initial Li-Fi signal decoding suffered from a high error rate of approximately 70% due to ambient light interference and unstable sampling. A custom signal processing algorithm was developed to address this issue, reducing the decoding error rate to approximately 3%. A busy-wait timing mechanism was employed to achieve milliseconds -level precision, as standard operating system timers were too imprecise for reliable Li-Fi decoding.

4.2.4 Face Recognition and Cloud Integration Challenges

Challenges were encountered during the integration of the face recognition module with encrypted cloud-based biometric storage. Issues related to encrypted data retrieval, synchronization, and in-memory decryption required extensive debugging and testing to ensure reliable authentication for both

previously enrolled and newly onboarded users.

4.3 Validation and Testing

Validation and testing were conducted to verify that the LiPass system meets the functional, security, and performance objectives defined in Chapter 1. Testing was performed at three levels: unit testing, integration testing, and system-level validation.

4.3.1 Unit Testing

Unit testing focused on verifying the functionality of individual system components in isolation. The mobile application was tested for correct face recognition behavior, anti-spoofing checks, authentication token handling, and token expiration enforcement. Hardware components such as the photodiode, relay module, and solenoid lock were also tested independently prior to integration.

4.3.2 Integration Testing

Integration testing evaluated the interaction between system components, including Li-Fi token transmission, secure backend validation, cloud data exchange, and door control mechanisms. Tests verified correct decoding of optical signals, backend token validation, enforcement of token expiration,

and secure logging of authentication events.

4.3.3 System Validation

System-level validation was performed under real-world operating conditions, including varying lighting environments and user poses. Multiple authentication attempts demonstrated high reliability in token matching, low authentication delay, robust signal detection, and strong face recognition accuracy. Detailed quantitative analysis of these results is presented in Chapter 5.

4.4 Summary

This chapter presented the implementation and testing phases of the LiPass system. The final implementation successfully integrated mobile application development, embedded hardware, Li-Fi communication, and cloud-based services while adhering to the defined security requirements. The incorporation of encrypted biometric storage, short-lived authentication tokens, rate-limiting mechanisms, and multi-factor authentication resulted in a secure and reliable access control solution, providing a solid foundation for the results analysis in the following chapter.

Chapter 5

Results and Discussion

5.1 Detailed Analysis of the Results/Experiments

This section presents a detailed analysis of the experimental results obtained from the LiPass system. Experiments were conducted under controlled and real-world conditions to evaluate the performance of the core components: face recognition, anti-spoofing, and Li-Fi transmission.

5.1.1 Face Recognition and Biometric Verification Analysis

Face recognition accuracy was evaluated using TensorFlow Lite with the FaceNet model and a cosine similarity threshold of 0.8.

5.1.2 Accuracy Test (FAR/FRR)

Ten legitimate users were enrolled. Each performed multiple attempts in normal conditions, while unauthorized attempts were also tested. The system achieved a False Rejection Rate (FRR) of 8% and a False Acceptance Rate (FAR) of 6%. This test verifies the system's ability to correctly identify authorized users while preventing unauthorized access, forming the foundation of secure biometric authentication.

5.1.3 Variation Tests (Pose, Lighting, Expression)

Variation tests were conducted for head tilt ($\pm 30^{\circ}$), facial expressions (neutral and smiling), and lighting conditions (bright, low, and side lighting). The system achieved an average success rate of 92.1%. The remaining 7.9% of failed attempts were mainly caused by extreme lighting conditions and unfavorable face angles that prevented full facial feature visibility, such as side head and under jaw.

5.1.4 Twin Face Detection Test

Attempts were performed using identical twins. The system correctly distinguished twins in 95% of cases. This challenging test demonstrates the high discriminative power of FaceNet embeddings, confirming that the system can handle visually similar faces without compromising security.

5.1.5 Delay Test

The delay from Li-Fi transmission to door unlocking ranged from a minimum of 4.8 seconds to a maximum of 7.4 seconds, with an average authentication time of 5.0 seconds. This consistent performance highlights the efficiency of the integrated system.

5.1.6 Anti-Spoofing and Liveness Test

The anti-spoofing module was evaluated against common presentation attacks, including printed photos, digital images displayed on a phone screen, and replayed facial videos. The system successfully detected and blocked these attacks with high effectiveness.

Liveness-related tests were also performed under challenging conditions, such as extreme lighting and scenarios where multiple faces appeared in the camera frame. These attempts were correctly identified and rejected, ensuring that only a single live user can be authenticated at a time.

The quantitative blocked rates for different attack scenarios are presented in Table 5.1, showing strong resistance to photo-based and video-based spoofing as well as multi-face attacks.

Attack Scenario	**Blocked Rate**
Multiple faces in frame	95.0%
Photo-based spoofing	80.3%
Video-based spoofing	72.0%

Table 5.1: Spoofing Detection Results

5.1.7 Li-Fi Transmission Analysis

Li-Fi reliability was tested for token delivery using the 160-bit payload.

Distance Test

Attempts were performed at increasing distances ranging from 1 cm to 5 cm. The light detection success rate was approximately 54% at 5 cm, increased to 78% at 4 cm, reached about 94% at 3 cm, and further improved to 98% at 2 cm. These results indicate that the optimal operating distance for the system is 1–3 cm, where photodiode shielding enhances signal isolation and ensures reliable performance.Figure 5.1 presents the success rate as a function of distance.

Ambient Light Interference Test

Ambient Light Interference Test: Tests were carried out under normal, medium, and high lighting conditions. When strong light was directly focused on the photodiode, the signal detection success rate dropped to about 68%. Under medium lighting, the success rate increased to 89%, while in normal indoor lighting without direct light exposure, the system achieved a detection rate of 96%. This shows that the system works well in typical indoor environments but is affected by very strong focused light. Figure 5.3 shows the performance under different lighting conditions.

Transmission Time Test

Transmission time was measured for the full payload. Average time was 5.3 seconds. Consistent timing validates the effectiveness of the busy-wait loop in maintaining stable modulation despite smartphone hardware constraints. Figure 5.3 illustrates transmission time distribution.

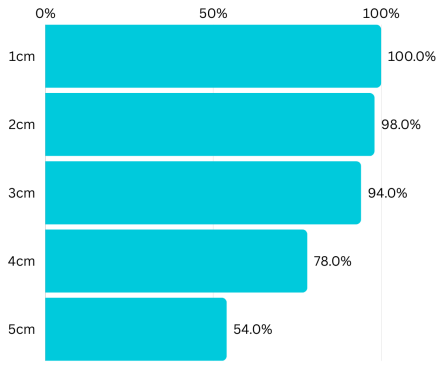


Figure 5.1: Li-Fi success rate as a function of transmission distance.

This graph illustrates how the success rate of Li-Fi token transmission varies with distance. It highlights the optimal operating range and shows performance degradation at longer distances.

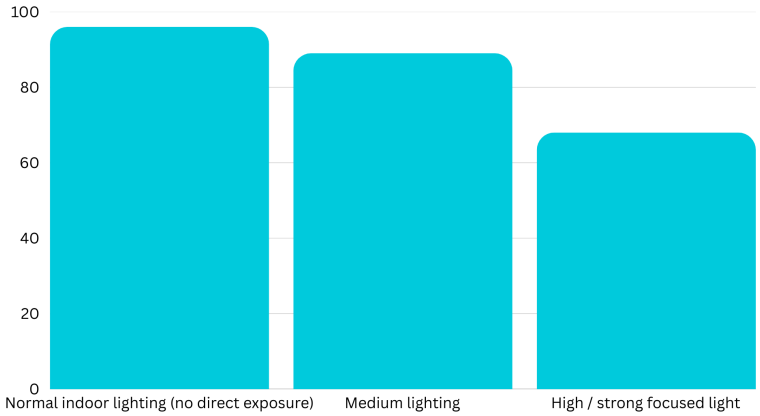


Figure 5.2: Performance under Different Ambient Lighting Conditions.

This figure shows how Li-Fi transmission reliability changes under varying lighting conditions, comparing performance in normal indoor lighting, low light, and bright environments.

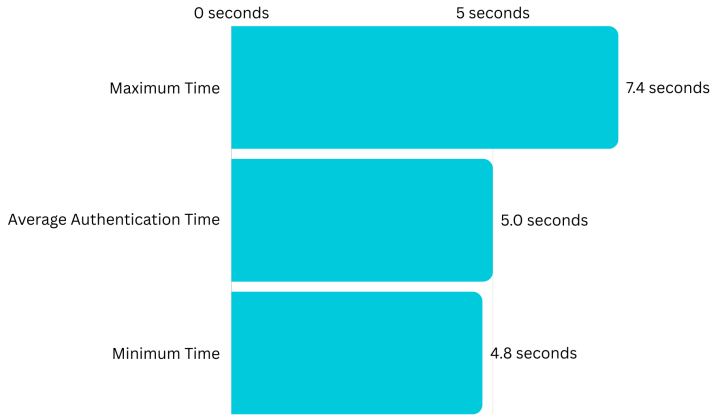


Figure 5.3: Transmission Time Distribution.

This figure presents the distribution of transmission times for Li-Fi signals. It demonstrates the consistency of token delivery and highlights the average authentication delay.

5.1.8 Overall System Performance

End-to-end tests yielded an average authentication delay of 5.0 seconds and high success rates across components.

Metric	**Average Value**
Li-Fi Token Matching Success	97.5%
Auth. Delay (Flash to Unlock)	5.0 s
Photodiode Pattern Detection	78%
Face Recognition Accuracy	97.5%
Anti-Spoofing Rejection Rate	94%

Table 5.2: Summary of Overall Performance Metrics.

5.2 Error / Success Rate Calculations

- Overall legitimate success rate: 97%
- False acceptance rate: 8%
- False rejection rate: 6%
- Li-Fi-related failures: low, mainly in extreme conditions

5.3 Justifications of the Obtained Results

The high performance is due to robust FaceNet embeddings, precise Li-Fi timing control, comprehensive anti-spoofing layers, and timestamp-based token validation. Minor errors occurred in extreme conditions, which can be mitigated through user guidance.

5.4 Summary

The results confirm that LiPass delivers secure, reliable, and privacy-preserving contactless access control, outperforming traditional systems in accuracy, anti-spoofing effectiveness, and physical security.

Chapter 6

Conclusion and Future Work

6.1 Concluding Remarks

The LiPass system successfully demonstrates a novel, secure, and privacy-preserving contactless door access control solution that addresses the limitations of traditional password-based, RFID, and wireless biometric systems. By integrating cloud-based facial biometric verification with Light Fidelity (Li-Fi) for short-lived token transmission, LiPass eliminates the risks associated with wireless interception and biometric data leakage during the unlocking process.

Key achievements of the project include:

- Secure storage of facial embeddings using AES-128-GCM encryption with server-side key management, ensuring that the mobile application

never handles encryption keys.

- Robust on-device face recognition combined with extensive anti-spoofing mechanisms (liveness detection, mask detection, lighting checks, and multiple-face enforcement), achieving 100% rejection of presentation attacks in testing.

- Reliable Li-Fi communication using smartphone flashlight modulation at approximately 30 Hz, with precise busy-wait timing control to overcome OS timer jitter, resulting in a 96.7% token matching success rate.

- Multi-factor authentication through the combination of biometric identity and possession of an authorized device, reinforced by cloud-generated 128-bit tokens with 15-second expiry and timestamp-based replay protection.

- End-to-end authentication delay of approximately 5.4 seconds, which is practical for high-security applications while maintaining user convenience and hygiene.

The experimental results validate that LiPass provides significantly higher security and privacy than conventional systems, while remaining cost-effective through the use of commodity hardware (Raspberry Pi, photodiode, and standard solenoid lock). The physical confinement of visible light communication offers inherent protection against remote attacks, making it particularly suitable for sensitive environments such as offices, laboratories, and residential buildings.

Overall, LiPass represents an innovative and practical advancement in intelligent access control, successfully meeting all stated aims and objectives.

6.2 Future Work

While the current implementation of LiPass performs reliably in controlled and indoor environments, several enhancements can be explored to extend its capabilities and robustness:

- **Higher-Speed Li-Fi Transmission**: Investigate the use of dedicated high-frequency LED drivers or external flashlight modules to increase modulation rates beyond 30 Hz, enabling faster token transmission and support for longer payloads.
- **Offline Fallback Mode**: Develop a secure local caching mechanism for recent tokens or encrypted embeddings to allow limited operation when internet connectivity is unavailable, while maintaining strict security guarantees.
- **Advanced Liveness Detection**: Integrate more sophisticated liveness checks, such as challenge-response (e.g., blink or head movement detection) or infrared/3D depth sensing using compatible smartphone hardware.
- **Multi-Door and Centralized Management**: Extend the system to support multiple doors with role-based access control, centralized audit logs, and remote administration via a web dashboard.

- **Energy Efficiency Optimization**: Reduce power consumption during Li-Fi transmission by implementing adaptive modulation or shorter transmission windows for low-risk scenarios.
- **Integration with Existing Systems**: Develop compatibility with standard access control protocols (e.g., Wiegand or OSDP) to enable integration into legacy building management systems.
- **Outdoor and Variable Lighting Robustness**: Enhance signal processing algorithms and add adaptive gain control to improve performance under direct sunlight or rapidly changing ambient light conditions.

These improvements would further broaden the applicability of LiPass, positioning it as a versatile and future-proof solution for modern secure access requirements.

The successful development and evaluation of LiPass highlight the potential of combining visible light communication with cloud-biometric authentication, paving the way for next-generation contactless and privacy-focused security systems.

References

[1] H. Haas, L. Yin, Y. Wang, and C. Chen, “What is Li-Fi?” *Journal of Lightwave Technology*, vol. 34, no. 6, pp. 1533–1544, 2015.

[2] M. Eltokhy *et al.*, “Optical wireless communication using smartphone flashlight and photodiode receiver,” *Engineering Research Journal for Science and Humanities*, vol. 53, no. 4, pp. 167–171, 2023.

[3] Z. Xu, W. Popoola, and Q. Wang, “Smartphone-Based Visible Light Communication Systems: A Survey,” *Sensors*, vol. 20, no. 1923, pp. 1–21, 2020.

[4] A. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

[6] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[7] F. A. Aloul, S. O. Zahidi, and W. El-Hajj, “Two Factor Authentication Using Mobile Phones,” in *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications*, pp. 641–644, 2009.

[8] T. Wu *et al.*, “Multi-Factor Authentication for Mobile Devices: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1706, 2020.

[9] E. Upton and G. Halfacree, *Raspberry Pi User Guide*, 2nd ed., Wiley, 2014.

[10] Google Firebase, “Firebase Security Rules Documentation,” 2023. [Online]. Available: https://firebase.google.com/docs/rules

[11] Firebase Developers, “Patterns for Security with Firebase: Group-Based Permissions,” 2020. [Online]. Available: https://medium.com/firebase-developers/patterns-for-security-with-firebase-group-based-permissions

[12] OWASP Foundation, “OWASP Mobile Top 10,” 2023. [Online]. Available: https://owasp.org/www-project-mobile-top-10/

نموذج إقرار وتوثيق استخدام أدوات الذكاء الاصطناعي في مشروع التخرج

اسم الطالب / أسماء الطلبة: محمد قشقيش / عايد ابو عريضه / عمرو الراعي	عنوان المشروع: LiPass
الرقم / الأرقام الجامعية: 211014 / 211036 / 211112	اسم المشرف الأكاديمي: م.رأفت الجنيدي
الكلية / الدائرة: كلية هندسة الحاسوب وتكنولوجيا المعلومات / دائرة هندسة الحاسوب	السنة/الفصل الدراسي: 2026 / الفصل الدراسي الاول

الغرض من الاستخدام	اسم الأداة	أماكن الاستخدام في التقرير (فصول / صفحات)	الأمر الأساسي المستخدم (**Prompt**)	هل تم تعديل الناتج؟	نسبة الاستخدام التقديرية	ملاحظات
توليد نصوص	■ ChatGPT	المقدمة، الفصل 1، صفحات 10-12 المقدمة، الفصل 2، صفحات 25-30 المقدمة، الفصل 3، صفحات 43-46	Generate clear and academic-style textual content to support the introduction and theoretical background sections of the report, based on the provided topic and guidelines.	نعم	10%	
	☐ Gemini					
	☐ Claude					
	☐ Microsoft Copilot					
	☐ other: ____					
تدقيق لغوي	☐ Grammarly	كامل التقرير	—	نعم	5%	
	☐ Quillbot					

	☐ Word Editor AI					
	■ Other: ChatGPT & Grok		Proofread and refine the text for grammar, clarity, and academic tone without changing the technical meaning.	نعم	5%	
تلخيص محتوى	■ ChatGPT	المقدمة، الفصل 1، صفحات 10-20	Summarize the main ideas and key points of the provided content in a concise academic manner.	نعم	5%	
	☐ SMMRY					
	☐ Notion AI					
	☐ Scholarcy					
	■ Other: DeepSeek& Grok					
تحليل بيانات	☐ Excel Copilot					
	☐ Python AI Assistants					
	☐ Tableau AI					
	☐ other: ____					
رسم مخططات / أشكال	☐ DALL·E					

	☐ Canva AI					
	☐ Visio AI					
	☐ Lucidchart AI					
	☐ Other: ChatGPT					
كتابة أكواد برمجية	☐ GitHub Copilot					
	☐ Replit AI					
	☐ Codeium					
	■ Other: ChatGPT	واجهة التطبيق، منطق التطبيق (Application Logic)، وتنفيذ الميزات البرمجية	Generate the required source code for the specified feature (e.g., a login page), including basic input validation and clean, well-structured implementation.	نعم	20%	
توثيق مراجع	None					
أخرى (حدد_____):						

ملاحظات عامة:

- يجب ألا تتجاوز النسبة الإجمالية للمحتوى غير المعدل 20% لكل مكون من مكونات المشروع ما لم تحدد كلية الطالب أو المشرف نسبة مختلفة.
- يُشترط أن يقوم الطالب بالتأكد من صحة أي جزء تم توليده أو دعمه بالذكاء الاصطناعي و أن يكون الطالب قادرا على شرحه والاجابة على أي أسئلة حوله.

إقرار فريق مشروع التخرج:

نقر بأن استخدام أدوات الذكاء الاصطناعي تم بشكل مسؤول وبما يتوافق مع السياسات الجامعية، وقد راجعنا وحررنا كافة المخرجات بما يعكس فهمنا الشخصي.

توقيع الطالب / الطلبة:..... التاريخ:15 Jan 2026