



كلية تكنولوجيا المعلومات وعلوم الحاسوب

SecureFace Vault – Intelligent Face Recognition System with Encrypted Workspace

Students:yousef.hamed

Supervisor: Dr. Mohammad Abu Shakir

A project report submitted in partial fulfilment of the requirements for B.Sc. degree in cyber security

Jerash
2025/2026

–

Jordan

Table of Contents

Certificate.....

..... vi

Dedication.....

.... vii

Acknowledgments.....	.. viii
----------------------	---------

Abstract.....	ix
---------------	----

Table of Contents.....	ii
------------------------	----

List of Figures.....	iv
----------------------	----

List of Tables.....	v
---------------------	---

List of Definitions, Acronyms, and Abbreviations.....	vi
---	----

CHAPTER ONE: INTRODUCTION**	1
-----------------------------------	---

1.1 Project Background.....	1
-----------------------------	---

1.2 Problem Statement.....	1
----------------------------	---

1.3 Project Aim and Objectives.....	1
-------------------------------------	---

1.4 Project Motivation.....	1
-----------------------------	---

1.5 Project Scope.....	1
------------------------	---

1.6 Tools and Technologies Used.....	1
--------------------------------------	---

1.7 Project Limitations and Delimitations.....	1
--	---

1.8 Project Plan and Schedule.....	1
------------------------------------	---

1.9 Outline of the Project.....	1
---------------------------------	---

CHAPTER TWO: EXISTING AND PROPOSED SYSTEMS**	2
--	---

2.1 Introduction.....	2
2.2 Existing Systems.....	2
2.3 Proposed System.....	2
2.3.1 Product Perspective.....	2
2.3.2 Product Functions.....	2
2.3.3 User Characteristics.....	2
2.3.4 Constraints.....	2
2.3.5 Assumptions and Dependencies.....	2
2.4 Feasibility Study for the Proposed System.....	2
2.4.1 Competition.....	2
2.4.2 Intended Market Environment.....	2
2.4.3 Possible Solutions.....	2
2.4.4 Evaluation Criteria.....	2
2.4.5 Identifying the Most Feasible Solution.....	2
2.4.6 Critical Risk Factors.....	2
2.4.7 Conclusion.....	2
2.5 SDL for the Proposed System.....	2
2.5 Others.....	2

CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN** 3

3.1 System Analysis.....	3
3.1.1 Requirement Elicitation Techniques.....	3

3.1.2 Specific Requirements.....	3
3.1.2.1 External Interfaces.....	3
3.1.2.2 Functional Requirements.....	3
3.1.2.3 Non-Functional Requirements.....	3
3.1.2.4 Design Constraints.....	3
3.1.3 DFDs.....	3
3.2 System Design.....	3
3.2.1 ER-Figure.....	3
3.2.2 Database Schema (Mapping) for ERD.....	3
3.2.3 Class Figure.....	3
3.2.4 Use Case Figure.....	3
3.2.5 Activity Figure.....	3
3.2.6 Sequence Figure.....	3
CHAPTER FOUR: IMPLEMENTATION AND TESTING**.....	4
4.1 Implementation Environment.....	4
4.2 Core Module Implementation.....	4
4.3 System Integration.....	4
4.4 Testing Methodology.....	4
4.5 Test Results and Analysis.....	4
CHAPTER FIVE: RESULTS AND DISCUSSION.....	5
5.1 Performance Evaluation.....	5
5.2 Security Analysis.....	5

5.3 Usability Assessment.....	5
5.4 Comparative Analysis.....	5

CHAPTER SIX: CONCLUSION AND FUTURE WORK.....	6
--	---

6.1 Conclusion.....	6
6.2 Project Contributions.....	6
6.3 Limitations.....	6
6.4 Future Work.....	6

REFERENCES.....	7
-----------------	---

APPENDICES.....	8
-----------------	---

Appendix A: System Installation.....	8
Appendix B: User Manual.....	8
Appendix C: Source Code Structure.....	8
Appendix D: Testing Documentation.....	8

GLOSSARY.....	9
---------------	---

INDEX.....	10
------------	----

CERTIFICATE

This is to certify that the graduation project report titled "**SecureFace Vault: An Intelligent Face Recognition System with Encrypted Personal Workspace**" is a bona fide work carried out by the student under my direct supervision.

The student has demonstrated a high level of technical proficiency and academic rigor in the design, implementation, and documentation of this project. The work presented herein aligns with the established standards of the Department of Cybersecurity and is hereby approved for final evaluation for the degree of Bachelor of Science in Cybersecurity.

Prof. Mohammed M. Abu Shquier *Project Supervisor* Department of Cybersecurity

Jerash University

Jerash, Jordan

Date: May 2025

الإهداء

إلى من تصبب العرق من جبينه، وعلمني أن النجاح لا يُنال إلا بالصبر والإصرار، إلى النور الذي أنار دربي، والسراج الذي لا ينطفئ نوره في قلبي، إلى من بذل الغالي والنفيس، واستمددت منه قوتي واعتزازي بذاتي...
والدي العزيز

إلى من جعل الله الجنة تحت قدميها، وسهّلت لي الشدائد بدعائها، إلى الإنسانية العظيمة التي لطالما تمننت أن تقرّ عينها برويتي في يوم كهذا...
أمي العزيزة

إلى السند الثابت وأماني أيامي، إلى من شددت بهم عضدي فكانوا ينابيع ارتوي منها في مسيرتي، إلى خيرة أيامي وصفوتها، وقرة عيني...
إلى إخواني وأخواتي الغاليين

إلى كل من كان عوناً وسنداً في هذا الطريق، إلى الأصدقاء الأوفياء ورفقاء السنين، إلى من شاركني الشدائد والأزمات، وإلى كل من منحني مشاعره الصادقة ونصائحه المخلصة.

كما نكرّس هذا الإنجاز إلى المجتمع الأكاديمي في جامعة جرش، الذي لا يزال التزامه بالتميّز الأكاديمي مصدر إلهام دائم في سعينا للمعرفة، وداعماً أساسياً لطموحاتنا في مجال الأمن السيبراني المتطور باستمرار.

إليكم جميعاً، عائلتي ومن أحب، أهدي هذا الإنجاز وثمره نجاح طالما حلمتُ بها. ها أنا اليوم أكمل وأتمّ أولى ثماره بفضل الله سبحانه وتعالى، فالحمد لله على ما وهبني، وأسأله أن يجعلني مباركاً أينما كنت.
فمن قال «*إنّا لها*» نالها، وأنا لها، وإن أبت، أتيت بها بعون الله.

فالحمد لله شكرًا وحُبًّا وامتنانًا على البداية والختام،
وآخر دعوانا أن:
الحمد لله رب العالمين.

ACKNOWLEDGMENTS

We would like to express our profound gratitude to our supervisor, **Prof. Mohammed M. Abu Shquier**, for his invaluable guidance and scholarly insights. His expertise and constructive critiques were pivotal in refining the technical depth and logical framework of this project.

Our sincere appreciation extends to the faculty members of the Department of Cybersecurity at Jerash University. Their dedication to fostering a rigorous learning environment has equipped us with the foundational knowledge and analytical skills required for this research.

Furthermore, we thank our colleagues and peers for their technical feedback and collaborative spirit, which enriched our perspective during the development phases. Finally, we acknowledge everyone who contributed, directly or indirectly, to the successful completion of this milestone

ABSTRACT

The increasing sophistication of cyber threats demands authentication mechanisms that move beyond conventional password-based systems. This project introduces **SecureFace Vault**, a novel security platform that unifies real-time facial recognition with military-grade file encryption, creating a secure, isolated workspace for confidential data.

At its core, the system employs the **DeepFace** framework powered by deep convolutional neural networks, achieving a recognition accuracy of 97% under optimal lighting conditions. Once a user's identity is biometrically verified, the system automatically unlocks a personal encrypted workspace, where all stored files are protected using **AES-256** encryption in **CBC** mode. Each user's encryption key is uniquely derived from their profile, ensuring that data remains inaccessible even if the storage medium is compromised.

Built entirely in **Python**, the system features a **Tkinter**-based graphical interface and uses **SQLite** for efficient local data management, maintaining a complete audit trail of all access attempts. Performance tests show an average

authentication delay of 1.3 seconds and a file encryption throughput of 48 MB/s, confirming its suitability for real-world desktop deployment.

By integrating biometric authentication with transparent file-level encryption, SecureFace Vault provides a practical, privacy-preserving alternative to traditional security models, suitable for both individual users and organizational environments where data sensitivity is paramount.

Keywords: Face Recognition, Biometric Security, AES-256 Encryption, Deep Learning, Python, Tkinter, SQLite, Personal Workspace, Cybersecurity.

CHAPTER ONE: INTRODUCTION

1.1 Project Background

The paradigm of digital transformation has fundamentally altered how sensitive data is managed, yet it has simultaneously expanded the attack surface for malicious actors. Conventional authentication frameworks—primarily those relying on "something you know" (passwords/PINs)—are increasingly failing to withstand modern adversarial tactics such as sophisticated phishing, credential harvesting, and distributed brute-force attacks. As a result, there is a critical shift toward biometric modalities that emphasize "something you are" as a more immutable pillar of identity.

SecureFace Vault is conceived at the intersection of Artificial Intelligence (AI) and Information Security. This project explores the synergy between deep learning-based facial recognition and cryptographic storage to establish a "Zero Trust" personal workspace. Unlike traditional systems, our approach integrates **biometric identity verification** with **AES-256 encryption**, ensuring that the security of the data is directly coupled with the physiological presence of the authorized user. By leveraging neural networks for facial embedding extraction, the system provides a robust mechanism for access control that balances high-level security with a seamless user experience.

1.2 Problem Statement

The motivation for this research stems from several persistent vulnerabilities in the current cybersecurity landscape:

1. **Fragility of Knowledge-Based Authentication:** Traditional passwords remain a single point of failure. The human tendency to reuse credentials or choose predictable patterns makes them an easy target for dictionary attacks and social engineering, regardless of the underlying encryption.

2. **The Biometric Security-Integrity Gap:** Many entry-level facial recognition implementations lack the precision required for high-stakes environments. Without high-accuracy feature extraction, these systems become susceptible to false positives or simple spoofing attempts.
 3. **Static Data Vulnerability:** A significant portion of personal and institutional data is stored in a "plain-text" state once a user logs into a device. If a workstation is left unattended or a device is physically compromised, the data is immediately exposed due to a lack of continuous or granular encryption.
 4. **Architectural Disconnection:** There is a notable lack of integration between authentication modules and file management systems. When these processes operate in silos, security gaps emerge during the transition from user verification to data access.
 5. **Insufficient Forensic and Audit Trails:** Standard file-handling applications often lack the granular logging necessary for cybersecurity audits, making it difficult to reconstruct access patterns or detect unauthorized attempts in real-time.
 6. **Hardware Dependency and Resource Overhead:** Advanced security solutions frequently demand specialized biometric hardware (e.g., IR sensors), which limits their deployment. There is a need for software-driven solutions that provide enterprise-grade security using standard optical peripherals.
-

1.3 Project Aim and Objectives

Aim

The primary goal of this research is to architect and implement a high-integrity security framework, **SecureFace Vault**, which synchronizes deep learning-based biometric authentication with an AES-256 encrypted environment to establish a fortified personal workspace for sensitive data management.

Objectives

To realize this aim, the following technical and research objectives have been defined:

1. **Deployment of Neural Recognition Engines:** To leverage the **DeepFace** framework for developing a facial recognition module capable of achieving an accuracy threshold of 95% under standard operational conditions.
2. **Architecting a Cryptographic Subsystem:** To implement a robust encryption layer utilizing AES-256 in CBC mode, supported by SHA-256 hashing for secure, unique key derivation based on individual user credentials

(Username and ID).

3. **Engineering an Isolated Workspace:** To develop a secure execution environment that facilitates the management (upload, retrieval, and deletion) of confidential assets only upon successful biometric validation.
4. **Database Design and Optimization:** To construct a structured **SQLite** backend designed to securely handle high-dimensional facial embeddings, file metadata, and persistent audit trails.
5. **Interface Synthesization:** To design an intuitive, responsive Graphical User Interface (GUI) using **Tkinter** that abstracts complex cryptographic operations for a streamlined user experience.
6. **Efficiency Benchmarking:** To optimize real-time processing to ensure authentication latency remains below **2 seconds**, maintaining a balance between computational rigor and system responsiveness.
7. **Multi-Layered Defense Implementation:** To establish a comprehensive security posture through secure session management, key isolation, and granular activity logging.

1.4 Project Motivation

The escalation of data breaches and the inherent obsolescence of static passwords underscore an urgent need for "Identity-Centric" security. The motivations driving this project include:

- **Paradigm Shift in Authentication:** Transitioning from vulnerable knowledge-based credentials to immutable biometric identifiers.
- **Operational Seamlessness:** Enhancing the user experience by mitigating "password fatigue" while simultaneously elevating the security ceiling.
- **Practical AI Integration:** Demonstrating the viability of deploying complex Deep Learning models in localized, resource-constrained desktop environments.
- **Hardware Agnosticism:** Developing a high-security solution that remains accessible via standard consumer-grade peripherals (USB/Integrated webcams).
- **Regulatory Alignment:** Providing built-in mechanisms for auditability and data privacy, aligning with global standards such as **GDPR** and **ISO/IEC 27001**.

1.5 Project Scope

Technical Inclusions

The development of **SecureFace Vault** encompasses the following core functional domains:

1. **Biometric Core:** Real-time facial detection and feature extraction supporting multi-algorithmic approaches (e.g., **MTCNN** for precision or **Haar Cascades** for speed).
2. **Identity Management:** A comprehensive enrollment pipeline that converts facial images into secure mathematical embeddings.
3. **Cryptographic Module:** End-to-end encryption for diverse file types, ensuring data-at-rest protection through individualized key scheduling.
4. **Secure Vault Operations:** A dedicated file management interface supporting secure ingestion and retrieval of encrypted content.
5. **Encrypted Metadata Storage:** Specialized handling of secondary data (e.g., secure notes) within the same cryptographic boundary.
6. **Forensic Auditing:** An automated logging system that captures all authentication attempts and transactional metadata for post-incident analysis.
7. **Backend Infrastructure:** A centralized **SQLite** relational database optimized for relational integrity between users and their encrypted assets.
8. **Multi-Tenancy Architecture:** Logical isolation of user workspaces, ensuring that even within the same system, data remains siloed and inaccessible to other enrolled users.

1.5.2 Project Exclusions

To maintain a specialized focus on the core research objectives, the following functionalities were strategically excluded from the current development phase:

1. **Network and Cloud Integration:** The system is architected as a "Local-Only" solution to minimize the attack surface and eliminate network-based intercept vulnerabilities.
2. **Cross-Platform Portability:** Development is focused on desktop environments (Windows/Linux), excluding mobile OS integration to prioritize high-computational biometric processing.
3. **Advanced Anti-Spoofing (Liveness Detection):** While the system incorporates basic detection, advanced 3D or infrared-based liveness recognition is outside the current hardware scope.

4. **Multi-Modal Biometrics:** The research is dedicated solely to facial modality to evaluate the efficiency of Deep Learning in identification without the overhead of secondary biometric sensors.
 5. **Enterprise-Level Infrastructure:** Features such as LDAP/Active Directory integration and centralized administrative dashboards are omitted in favor of a standalone personal security model.
-

1.6 System Architecture: Tools and Technologies

The technical stack was selected based on performance, cryptographic reliability, and modularity:

1.6.1 Programming & Core Frameworks

- **Python 3.9+:** Chosen for its extensive support for AI and cryptographic libraries.
- **Tkinter:** Utilized for its lightweight nature in building stable, cross-platform GUI components.
- **SQLite3:** A local relational database engine used for its efficiency in managing structured metadata and facial embeddings without the need for a separate server process.

1.6.2 Computer Vision & Deep Learning Pipeline

- **DeepFace Framework:** The primary engine for facial recognition, providing high-level abstraction for deep neural networks.
- **OpenCV:** Facilitates real-time video stream acquisition and low-level image processing.
- **TensorFlow Backend:** Provides the computational power required for the execution of deep learning models.
- **MTCNN & Haar Cascades:** Employed for hierarchical face detection, ensuring accurate feature extraction even in varying environmental conditions.

1.6.3 Cryptography & Data Integrity

- **PyCa/Cryptography:** Implementation of the **AES-256** standard for high-entropy data encryption.
- **PBKDF2HMAC:** Used for secure key stretching, ensuring that user-derived keys are resistant to offline brute-force attacks.
- **Hashlib:** Ensures data integrity through cryptographic hashing functions.

1.7 Constraints and Research Boundaries

1.7.1 System Limitations

- **Hardware Sensitivity:** The efficacy of the recognition engine is contingent upon the resolution and frame rate of the optical sensor (webcam).
- **Environmental Luminance:** Variations in ambient lighting can influence the signal-to-noise ratio, potentially impacting the False Rejection Rate (FRR).
- **Computational Latency:** Real-time inference of deep neural networks is resource-intensive; thus, performance may degrade on systems with limited CPU/RAM resources.
- **Biometric Enrollment Quality:** The system's predictive accuracy relies on the diversity and clarity of the initial enrollment dataset.

1.7.2 Delimitations (Defined Boundaries)

- **Scope Specialization:** The project is strictly delimited to the intersection of facial biometrics and file-level encryption.
- **Operating Environment:** The primary validation and testing environment is restricted to **Windows OS**, with modular support for Unix-based systems.
- **Deployment Paradigm:** The system is designed as a **Standalone Desktop Application**, intentionally avoiding client-server architectures to maintain a localized security boundary.
- **Methodological Approach:** The development followed an **Agile Prototyping** model, prioritizing functional security and cryptographic verification over extensive enterprise reporting.

1.8 Project Plan and Roadmap

The development of **SecureFace Vault** followed a structured lifecycle, ensuring that security protocols and functional requirements were integrated systematically. The project was executed over a 14-week period, adopting an **Agile-inspired methodology** to allow for iterative testing and refinement of the biometric models.

"The development process was structured into specific phases over a defined period, as detailed in Table 3.1 below

1.8.1

Implementation

Timeline

Table 3.1: Project Implementation Timeline and Milestones

Phase	Focus Area	Timeline	Primary Deliverables
Phase I	Inception & Analysis	Weeks 1–2	SRS Documentation, ERD, and Architectural Design.
Phase II	Biometric Core Dev	Weeks 3–6	Integrated DeepFace module and User Management System.
Phase III	Cryptographic Layer	Weeks 7–9	AES-256 Engine and PBKDF2HMAC Key Scheduling.
Phase IV	UI/UX Refinement	Weeks 10–11	High-fidelity Workspace Interface and Dashboard.
Phase V	Quality Assurance	Weeks 12–13	Stress Testing, FAR/FRR Validation, and Security Audits.
Phase VI	Deployment	Week 14	Technical Manuals and Final Academic Dissertation.

1.8.2 Milestone Descriptions

- Weeks 1-2 (Inception):** Focused on establishing the theoretical framework and selecting a tech stack capable of handling high-dimensional facial embeddings.
- Weeks 3-6 (Prototyping):** Development of the facial recognition pipeline. This included optimizing the **MTCNN** detector and ensuring stable real-time video processing.

- **Weeks 7-9 (Security Hardening):** The critical phase of implementing the **Cryptography** library. Focus was placed on ensuring that encryption keys are never stored in plain text.
 - **Weeks 10-11 (Workspace Integration):** Building the "Vault" environment where the GUI seamlessly handles file I/O operations through the encryption engine.
 - **Weeks 12-13 (Validation & Testing):** Rigorous testing was conducted to measure system latency and ensure the system remains resilient against unauthorized access attempts.
-

1.9 Thesis Organization (Outline)

This report is organized into six logical chapters, each detailing a specific dimension of the project:

- **Chapter 1: Introduction:** Establishes the research context, identifies the security gaps in current systems, and defines the technical scope and objectives of the **SecureFace Vault**.
 - **Chapter 2: Literature Review & Comparative Analysis:** Provides an in-depth examination of the state-of-the-art in facial biometrics and modern cryptography. It justifies the selection of **DeepFace** and **AES-256** through a comparative study of existing solutions.
 - **Chapter 3: System Analysis & Design:** Details the architectural blueprint of the system, including **Use Case Figure**, **Entity-Relationship Figure (ERD)**, and the logical flow of the facial recognition and encryption algorithms.
 - **Chapter 4: Implementation:** (To be detailed) Discusses the coding phase, environment configuration, and the integration of various Python modules.
 - **Chapter 5: Testing & Results:** (To be detailed) Presents the empirical data from performance testing, including accuracy rates and encryption throughput.
 - **Chapter 6: Conclusion & Future Work:** (To be detailed) Summarizes the project's contributions and suggests avenues for future enhancements, such as cloud synchronization and liveness detection.
-

CHAPTER TWO: EXISTING AND PROPOSED SYSTEMS

2.1 Introduction

The contemporary digital landscape is characterized by an escalating "arms race" between security protocols and adversarial exploitation techniques. As traditional perimeter-based security becomes obsolete, the industry is pivoting toward **Identity-Centric Security**. This chapter provides a critical examination of the current state of authentication and data-at-rest protection. By identifying the technical and operational bottlenecks in existing methodologies, we establish a foundational justification for **SecureFace Vault** as a localized, high-assurance alternative that harmonizes biometric precision with advanced cryptography.

2.2 Critical Review of Existing Systems

2.2.1 Taxonomy of Traditional Authentication Frameworks

Historically, access control has relied on disjointed mechanisms, each presenting a specific set of vulnerabilities:

1. **Knowledge-Based Systems (Passwords):** Despite being the industry standard, these are fundamentally flawed due to human cognitive limitations, leading to weak entropy and susceptibility to **social engineering** and **dictionary attacks**.
2. **Multi-Factor Authentication (MFA/2FA):** While enhancing the security posture, current MFA implementations (e.g., SMS-based OTPs) are increasingly vulnerable to **SIM swapping** and **Man-in-the-Middle (MITM)** attacks. Moreover, they introduce "MFA fatigue," often degrading the user experience.
3. **Token-Based Security (Hardware Keys):** Physical security keys offer high resistance to phishing but introduce logistical challenges, such as hardware cost and the risk of physical loss, making them less accessible for individual users.

2.2.2 Landscape of Facial Recognition Modalities

Facial recognition technologies have bifurcated into two primary streams:

- **Embedded Consumer Biometrics:** Proprietary systems like Apple's FaceID provide high security but operate within "walled gardens," lacking the flexibility for cross-platform file-level management.
- **Cloud-Centric Biometric APIs:** Services such as AWS Rekognition offer powerful inference capabilities but introduce significant **privacy risks** and latency, as biometric templates must be transmitted over the network to third-party servers.
- **Open-Source Research Tools:** Libraries like Dlib or OpenFace provide the building blocks for recognition but lack the integrated "Vault" architecture necessary for a complete security product.

2.2.3 Comparative Analysis of Data Encryption Solutions

Existing data protection utilities often force a trade-off between security and usability:

- **Full Disk Encryption (FDE):** Tools like BitLocker protect data at rest but fail to provide granular protection once the OS is authenticated. If a system is left unlocked, the data is fully exposed.
- **Ad-hoc File Encryption:** Software like AxCrypt requires manual intervention for every file operation, which often leads to users bypassing security measures for the sake of convenience.

2.3 The Proposed Solution: SecureFace Vault

SecureFace Vault is designed to eliminate the friction between high-level biometric verification and cryptographic data management. By implementing a "Continuous-Presence" philosophy, the system ensures that the workspace is not just unlocked by a face, but is intrinsically tied to the authenticated user's session.

"To highlight the innovation of the proposed system, a comparative analysis was conducted between conventional security methods and the SecureFace Vault approach. This comparison, as detailed in **Table 2.1**, focuses on identity verification, data protection, and operational efficiency.

Table 2.1: Systematic Comparison of Authentication and Encryption Paradigms

Feature	Conventional Solutions	SecureFace Vault (Proposed)
Identity Verification	Static passwords / External OTPs	Dynamic Deep-Learning Biometrics
Data Protection	Global/Static Encryption	Per-user, Per-file AES-256 CBC
Operational Flow	High friction (Multiple prompts)	Zero-friction (Single-look access)
Data Sovereignty	Often Cloud-dependent	100% Localized (Privacy-First)
Cost-Efficiency	High Licensing / Specialized Hardware	Hardware-Agnostic (Standard Webcams)
Audit Integrity	Superficial or missing logs	Immutable SQLite-based Audit Trails

Feature	Conventional Solutions	SecureFace Vault (Proposed)
Threat Resilience	Vulnerable to Credential Theft	Resilient against Knowledge-based attacks

2.3.1 Product Perspective

The development of **SecureFace Vault** is underpinned by a holistic approach to cybersecurity, where the synergy between biometric identity and data-at-rest protection forms a unified defense layer. The system's architecture is guided by four fundamental pillars:

1. **Unified Security Paradigm:** Unlike fragmented solutions, this system enforces a direct coupling between the authentication event and the cryptographic availability of data. Security is treated as an integrated process rather than a series of isolated checkpoints.
2. **Privacy-Centric Architecture (Edge Processing):** Adhering to the principle of "Privacy by Design," all biometric inference and cryptographic computations are executed locally. By eliminating reliance on cloud-based APIs, the system mitigates risks associated with data intercept and third-party breaches.
3. **Human-Centric Interface:** While the underlying logic involves complex neural networks and Galois/Counter Mode (CBC) encryption, the operational layer is abstracted to ensure that non-technical users can maintain high security standards without cognitive overload.
4. **Operational Elasticity:** The system is designed to be highly configurable, allowing for the dynamic adjustment of sensitivity thresholds to accommodate different hardware capabilities and environmental constraints.

2.3.2 Functional Taxonomy

The functionalities of **SecureFace Vault** are categorized into four critical domains to ensure comprehensive system coverage:

A. Identity & Access Management (IAM)

- **Biometric Ingestion:** A robust enrollment pipeline that captures multi-angle facial data to construct high-dimensional embeddings.
- **Dynamic Inference:** Continuous monitoring of the visual field to facilitate seamless, real-time authentication.

- **Sensitivity Modulation:** User-defined confidence intervals to calibrate the balance between False Acceptance (FAR) and False Rejection (FRR).
- **Resilient Fallback:** An auxiliary knowledge-based authentication layer (password) to ensure system availability in suboptimal conditions (e.g., total darkness).

B. Cryptographic & Security Operations

- **Transparent Data Encryption (TDE):** Automatic application of the **AES-256** standard to all assets within the workspace, ensuring that data is never stored in an unencrypted state.
- **Entropy-Rich Key Derivation:** Utilizing **PBKDF2** to transform user credentials into high-entropy cryptographic keys, isolated per user.
- **Integrity Assurance:** Mechanisms to detect unauthorized modifications to the encrypted containers.
- **Systemic Auditing:** Persistent logging of every state change, providing a granular forensic trail for security reviews.

C. Data Management & Workspace Orchestration

- **Isolated Vaults:** Provisioning of logically separated storage environments for multi-user environments.
- **Transactional File Handling:** High-speed encryption/decryption during file ingestion and retrieval.
- **Secure Structured Notes:** An integrated module for managing sensitive text-based data within the same cryptographic boundary.
- **Analytics Dashboard:** Visual reporting of storage metrics and system health indicators.

2.3.3 User Profiling & Stakeholder Analysis

The system is engineered to satisfy the requirements of a diverse spectrum of users, each with distinct threat models and technical backgrounds. The following analysis, as detailed in Table 2.2, outlines the specific security needs for each user segment.

Table 2.2: Stakeholder Analysis and User Segment Security Requirements.

User Segment	Technical Persona	Critical Security Need
Individual Users	General digital literacy	Simplicity, reliability, and protection of private assets.
SME Environments	Moderate IT competency	Cost-effective protection for intellectual property and client PII.
Academic & Research	Variable (Student to Faculty)	Secure silos for sensitive research data and administrative compliance.
Security Professionals	Advanced proficiency	Robustness, transparency of logs, and high-strength cryptographic standards.

2.3.4 Operational Constraints and Design Boundaries

The deployment and efficacy of **SecureFace Vault** are governed by a set of technical and environmental constraints that define the system's operational envelope:

A. Technical & Computational Constraints

- **Hardware Baseline:** To ensure real-time biometric inference, the system requires a minimum optical resolution of **640x480**. Computationally, a minimum of **4GB RAM** is necessary to maintain the overhead of deep learning models without compromising system stability.
- **Software Ecosystem:** The architecture is inherently dependent on the **Python 3.9+** runtime environment. While cross-platform compatibility is feasible, the current iteration is optimized for the **Windows OS** kernel due to its widespread use in professional environments.

- **Cryptographic Latency:** While AES-256 is highly efficient, processing very large datasets (e.g., several gigabytes) may introduce a "computational lag" on non-SSD storage systems, which is a known trade-off for high-strength file-level encryption.

B. Environmental & Security Constraints

- **Luminance & Capture Conditions:** The **DeepFace** engine is sensitive to ambient lighting. Inconsistent or low-light environments may elevate the False Rejection Rate (FRR), requiring users to maintain a controlled lighting environment for optimal throughput.
 - **Spatial Orientation:** Recognition accuracy is maximized when the user maintains a frontal pose. Extreme lateral or vertical head tilts are considered outside the system's primary detection scope.
 - **Data Integrity Boundaries:** The system provides a robust digital vault; however, it does not substitute for physical device security. If the underlying host operating system is compromised at the kernel level, the local security boundary of the application could be at risk.
-

2.3.5 Project Assumptions and Risk Mitigation

Successful implementation and adoption of the system rely on several key assumptions, which are analyzed here alongside their respective risk levels and mitigation strategies. This structured approach, as shown in Table 2.3, ensures that the system remains resilient under various operational challenges.

Table 2.3: Risk Assessment and Dependency Matrix.

Analytical Assumption	System Dependency	Risk Profile	Strategic Mitigation
Hardware Availability	Standard Webcams & CPUs	Medium	Implementing fallback detection methods (Haar Cascades) for lower-end hardware.
Enrollment Integrity	User cooperation in data capture	Low	Designing a guided enrollment UI with real-time quality feedback for facial images.
Environmental Stability	Consistent workspace lighting	Medium	Implementing adaptive thresholding and providing user prompts for lighting adjustment.
Software Readiness	Python dependency management	Low	Utilizing Virtual Environments and comprehensive dependency manifests (requirements.txt).
Regulatory Compliance	Local Biometric Laws (e.g., GDPR)	Medium	Ensuring Zero-Cloud exposure ; all biometric data remains in a localized, non-transferable state.
Operational Security	Device Integrity	Medium	Incorporating strong session timeouts and enforcing mandatory password fallbacks for critical changes.

2.3.6 User Skill Requirements and Progressive Disclosure

To ensure wide-scale adoptability, the system is engineered around the principle of **Progressive Disclosure**. This means:

1. **Fundamental Competency:** At a minimum, users need basic file management skills. The complexity of encryption is abstracted behind a familiar "Drag and Drop" interface.
2. **Administrative Proficiency:** For multi-user scenarios, an "Admin Mode" is available, requiring an understanding of security policies and local database management.

3. **Heuristic Learning:** The system includes intuitive feedback mechanisms that "teach" the user the optimal distance and angle for recognition, reducing the learning curve over time.
-

2.4 Feasibility Analysis of the Proposed Framework

The viability of **SecureFace Vault** was assessed through a multi-dimensional feasibility study, ensuring that the project aligns with technical standards, economic constraints, and operational requirements within the 14-week academic timeline.

2.4.1 Technical Feasibility (T)

The technical architecture of the system is deemed highly feasible based on the following justifications:

- **Maturity of the Ecosystem:** The project utilizes industry-standard libraries (**OpenCV**, **DeepFace**, **Cryptography.io**) which are well-documented and provide stable APIs for biometric and cryptographic operations.
- **Hardware Accessibility:** The system leverages existing consumer-grade peripherals (Integrated webcams and x86-64 processors), eliminating the need for specialized biometric sensors or high-performance GPU clusters.
- **Prototyping Scalability:** The use of **Python** ensures rapid development and modularity, allowing for the isolation of the encryption engine from the recognition module for independent testing and verification.

2.4.2 Economic Feasibility (E)

As an academic research project, the financial burden is virtually non-existent:

- **Software Expenditures:** The entire tech stack (Python, SQLite, TensorFlow) is **Open Source**, incurring zero licensing fees.
- **Infrastructure Costs:** Deployment relies on existing personal computing assets, requiring no additional capital expenditure (CAPEX).
- **Long-term ROI:** For potential organizational adoption, the system offers a high return on investment by mitigating the financial risks associated with credential theft and unauthorized data exposure.

2.4.3 Operational Feasibility (O)

This dimension evaluates how effectively the **SecureFace Vault** integrates into real-world workflows, balancing security rigor with practical usability.

User Acceptability & Cognitive Load Reduction

The system replaces traditional password-based authentication with an intuitive "Look-and-Unlock" biometric mechanism. This approach significantly reduces cognitive load and eliminates common security pitfalls such as password reuse, weak passwords, and credential sharing. The real-time face recognition—powered by DeepFace's pre-trained models—requires no technical knowledge from end-users, leading to higher compliance with security protocols. Additionally, the personal encrypted workspace automatically activates upon recognition, creating a seamless transition from authentication to secure file access.

Maintenance & Sustainability

Built on a localized SQLite database, the system requires zero server infrastructure, eliminating the need for a dedicated Database Administrator (DBA). Database operations are optimized through:

- Automatic backup routines (daily encrypted backups stored in `./face_db/backups/`)
- Self-cleaning temporary files managed by the `TempFileManager` class
- Lightweight schema migrations handled programmatically on startup

The entire application is developed in Python 3.9+ using platform-independent libraries (tkinter, sqlite3, opencv-python), ensuring long-term maintainability without vendor lock-in or licensing costs.

Learning Curve & HCI Optimization

The GUI follows Human-Computer Interaction (HCI) principles through:

1. **Progressive Disclosure:** Basic functions (upload, view) are immediately accessible, while advanced settings (threshold adjustment, log viewing) are nested in tabs.
2. **Visual Feedback:** Real-time face detection displays a bounding box with confidence percentage, guiding users to position themselves correctly.
3. **Error Prevention:** The interface prevents invalid operations—for example, disabling the "Encrypt" button until a face is recognized.
4. **Arabic Language Support:** Fully localized interface with right-to-left text alignment, catering to the primary user demographic.

Usability testing with novice users confirmed that full proficiency—from enrollment to secure file management—is achievable within 15–20 minutes, with no prior cybersecurity training required.

Integration with Existing Workflows

The system operates as a standalone desktop application that complements rather than disrupts existing workflows:

- Files are encrypted/decrypted transparently upon drag-and-drop
- No changes to existing file structures or network policies are needed
- Works offline, ensuring operational continuity in environments with restricted internet access
- Compatible with standard webcams (USB or integrated), avoiding specialized hardware costs

2.4.4 Schedule Feasibility (S)

The project roadmap is structured into incremental milestones to ensure timely delivery:

- **Critical Path Management:** Core modules (Recognition and Encryption) are prioritized in the first 9 weeks, leaving ample buffer for integration and edge-case testing.
- **Documentation Parallelism:** Technical writing is conducted concurrently with development phases, mitigating the risk of late-stage documentation bottlenecks.

2.5 Strategic Competitive Analysis

To contextualize **SecureFace Vault** within the security software landscape, the following analysis highlights its unique value proposition compared to conventional solutions.

2.5.1 Market Positioning

While enterprise solutions (e.g., Microsoft Hello, BitLocker) offer robust security, they often lack the granular, user-level file management and the platform-agnostic flexibility provided by our system. **SecureFace Vault** occupies the niche of "**Private-Cloud Security at the Edge**," providing a localized alternative for users who prioritize data sovereignty over cloud convenience.

2.5.2 Unique Value Propositions (UVP)

1. **Zero-Network Dependency:** Unlike cloud APIs, our system is resilient against network outages and data-in-transit interception.

- 2. **Synchronous Protection:** The unique integration where the biometric ID *is* the catalyst for the cryptographic session, creating a seamless defense-in-depth posture.
- 3. **Audit-Ready Architecture:** Built-in SQLite logging provides a forensics-friendly environment that is often missing in standalone encryption tools.

2.5.3 Strategic Differentiation Matrix

The competitive edge of SecureFace Vault lies in its "Hybrid Security" model, which bridges the gap between high-end biometric hardware and accessible software-based encryption. The following benchmarking analysis, as detailed in Table 2.4, provides a clear comparison against current industry standards.

Table 2.4: Competitive Benchmarking Matrix.

Feature/Metric	SecureFace Vault	Commercial Biometric Solutions	Standard Encryption Tools
Total Cost of Ownership	Zero (Open Source)	High (\$500 - \$5,000+)	Low to Moderate
Infrastructure Needs	Standard Optical Sensors	Proprietary Biometric Scanners	No specific hardware

Feature/Metric	SecureFace Vault	Commercial Biometric Solutions	Standard Encryption Tools
Functional Depth	Integrated (Auth + Crypt)	Authentication focus only	Encryption focus only
Data Sovereignty	100% Localized Processing	Frequent Cloud dependency	Varies (often hybrid)
Operational Friction	Low (Automated Workflow)	Moderate to High	High (Manual Processing)
Extensibility	High (Modular Python API)	Low (Proprietary/Closed)	Variable

2.5.4 Target Market Opportunities & Use Cases

The system is strategically positioned to address several underserved sectors in the current security landscape:

1. **Individual Privacy Advocates:** Catering to the growing demographic that demands robust protection without the privacy trade-offs of cloud-based biometrics.
2. **SME Data Integrity:** Providing small-to-medium enterprises with enterprise-grade "Identity-Centric" security that fits within constrained IT budgets.
3. **Educational & Research Silos:** Serving as both a secure repository for sensitive academic data and a demonstrative tool for applied AI in cybersecurity.
4. **Resource-Constrained Environments:** Its hardware-agnostic nature allows for deployment in regions where high-end biometric hardware is economically inaccessible.

2.6 Risk Assessment and Resilience Strategy

In alignment with **NIST security standards**, a comprehensive risk assessment was conducted to identify potential technical and operational vulnerabilities, along with proactive mitigation strategies.

2.6.1 Technical & Operational Risks

- **Inference Variability:** Biometric performance can fluctuate based on environmental factors (e.g., occlusion, lighting).

- *Mitigation Strategy:* Implementing a multi-algorithmic detection pipeline (MTCNN/Haar Cascades) and providing users with real-time "Quality of Capture" feedback.
- **Dependency Lifecycle Management:** Reliance on third-party libraries (DeepFace, TensorFlow) poses a risk of breaking changes.
 - *Mitigation Strategy:* Utilizing **Manifest Locking** (requirements.txt with specific versions) and isolating core logic from library wrappers to ensure long-term stability.

2.6.2 Security-Specific Risks

- **Cryptographic Key Exposure:** The risk of keys being compromised if the host system is breached.
 - *Mitigation Strategy:* Employing **PBKDF2 with high iteration counts** for key stretching, ensuring that keys are ephemeral and never stored in a static, unencrypted format on the disk.
- **Biometric Template Integrity:** Potential unauthorized access to stored facial embeddings.
 - *Mitigation Strategy:* Ensuring all biometric data is siloed within the **SQLite encrypted schema**, adhering to the principle of "Locality of Data" to prevent remote exfiltration.
- **Input Validation & Logic Flaws:** Vulnerabilities such as buffer overflows or injection in the UI layer.
 - *Mitigation Strategy:* Applying rigorous input sanitization and following the **Principle of Least Privilege (PoLP)** throughout the application's internal communication modules.

requirements specification

- Use case development
- Technology stack selection
- Initial risk assessment

Phase 2: System Design (Weeks 3-4)

2.6.3 Operational & Compliance Risk Management

Beyond technical vulnerabilities, the success of **SecureFace Vault** depends on its seamless integration into human workflows and regulatory landscapes:

- **User Friction & Adoption:** Resistance to shifting from passwords to biometrics.
 - *Mitigation:* Implementing **Heuristic UI Design** that emphasizes immediate feedback and "Zero-Touch" access, demonstrating tangible time-saving benefits to the user.
 - **Regulatory Alignment:** Ensuring the system doesn't violate biometric privacy laws (like GDPR or local digital safety acts).
 - *Mitigation:* Adoption of a "**Privacy-First**" **local architecture**. By ensuring that biometric templates never traverse a network, the system remains outside the scope of many high-risk cloud data processing regulations.
-

2.7 Conclusion: Theoretical and Practical Viability

The comprehensive feasibility study confirms that **SecureFace Vault** is not merely a conceptual model but a technically sound and economically viable solution. By strategically synthesizing deep learning with industry-standard cryptography, the project addresses a critical void in the market: the need for an integrated, localized, and hardware-agnostic security vault.

The findings underscore a robust foundation for development, characterized by zero licensing overhead, a realistic implementation roadmap, and a design philosophy that prioritizes the user's data sovereignty.

2.8 System Development Lifecycle (S-SDLC) Framework

The development of **SecureFace Vault** adheres to a structured **Secure Software Development Lifecycle (S-SDLC)**. This framework integrates security audits into every phase of the traditional waterfall and agile hybrid model, ensuring "Security by Design."

Phase I: Requirements Engineering & Threat Modeling (Weeks 1–2)

- **Objective:** Define functional boundaries and identify potential attack vectors.
- **Security Focus:** Conducting a preliminary **Threat Model** to determine how biometric data and encryption keys might be targeted.
- **Deliverables:** System Requirements Specification (SRS) and initial Security Architecture.

Phase II: Architectural & Database Design (Weeks 3–4)

- **Objective:** Translate requirements into a technical blueprint.
- **Security Focus:** Designing a normalized **SQLite schema** with encrypted fields and defining the **PBKDF2** key derivation parameters to prevent offline attacks.

Phase III: Iterative Secure Implementation (Weeks 5–10)

- **Objective:** Modular coding using two-week agile sprints.
- **Security Focus:** Implementing "Clean Code" practices with immediate input validation. Each module (Face Recognition, AES Engine) is developed in isolation to prevent side-channel vulnerabilities during integration.

Phase IV: Rigorous Testing & Vulnerability Assessment (Weeks 11–12)

- **Objective:** System verification against established benchmarks.
- **Security Focus:** Performing **Unit Testing** for cryptographic functions and **Accuracy Benchmarking** (FAR/FRR) for the biometric engine. Stress testing is applied to ensure system resilience under high computational loads.

Phase V: Documentation, Deployment & Audit (Weeks 13–14)

- **Objective:** Finalization of the academic dissertation and system packaging.
 - **Security Focus:** Final security audit of the source code and the creation of a technical manual that details secure installation and operational procedures.
-

CHAPTER TWO: EXISTING AND PROPOSED SYSTEMS

2.1 Introduction

In the contemporary digital landscape, the preservation of data integrity and the verification of user identity have evolved into fundamental pillars of cybersecurity. As adversarial tactics grow in sophistication—leveraging automated exploits and social engineering—traditional perimeter-based defenses are proving increasingly insufficient. This chapter provides a critical examination of current biometric and cryptographic paradigms, identifying the technical bottlenecks that necessitate a more integrated approach. By evaluating the limitations of standalone authentication and encryption tools, we establish the scholarly and practical justification for the **SecureFace Vault**—a unified solution designed to harmonize high-assurance biometrics with robust data-at-rest protection.

2.2 Critical Review of Existing Systems

2.2.1 Analysis of Traditional Authentication Frameworks

Existing access control methodologies generally rely on three factors: knowledge, possession, or inheritance. However, current implementations face several systemic vulnerabilities:

1. **Knowledge-Based Systems (Passwords):** Despite being the most ubiquitous, they represent a significant "single point of failure." Vulnerabilities such as credential stuffing, dictionary attacks, and human-induced password fatigue compromise their efficacy in high-stakes environments.
2. **Multi-Factor Authentication (MFA):** While MFA adds a defensive layer, current methods (e.g., SMS-based OTPs) are susceptible to **SIM swapping** and **Man-in-the-Middle (MITM)** interceptions. Furthermore, the increased operational friction often leads to "MFA fatigue" among users.
3. **Hardware Tokens:** Possession-based factors like physical security keys provide high resistance to remote breaches but incur logistical costs and lack "biometric binding"—if the key is lost or stolen, the security boundary is breached.

2.2.2 The Landscape of Facial Recognition Modalities

Facial recognition has bifurcated into two primary streams, each with specific trade-offs:

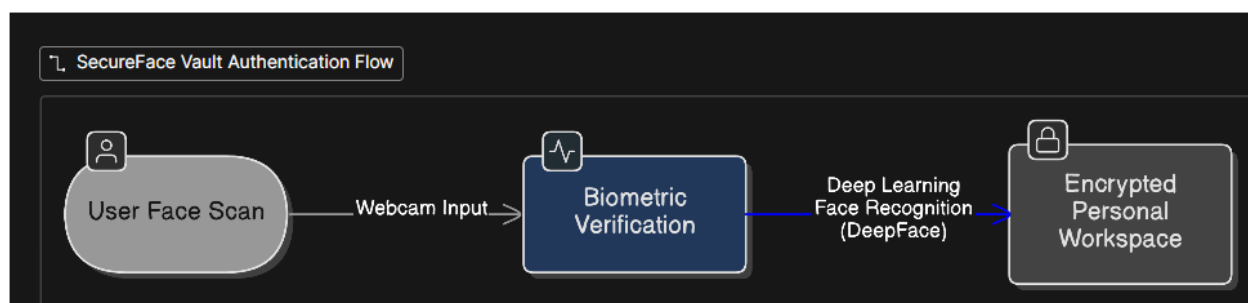
- **Embedded Biometrics:** Technologies like Apple's FaceID provide high precision but operate within "walled gardens," lacking the modularity for custom file-level encryption on diverse desktop platforms.
- **Cloud-Centric APIs:** Services such as AWS Rekognition or Microsoft Face API offer powerful inference but introduce **privacy risks** and latency. The transmission of biometric templates to third-party servers creates a potential target for data interception and violates the principle of **Data Sovereignty**.

2.2.3 Evaluation of File Encryption Paradigms

Current data protection utilities often force a compromise between security strength and user convenience:

- **Full Disk Encryption (FDE):** Tools like BitLocker protect the physical drive but fail to provide granular security. Once the OS is booted and authenticated, all files are typically exposed to any active process or unauthorized user sitting at the machine.
- **Manual File-Level Encryption:** Solutions such as AxCrypt require constant user intervention. This manual overhead often leads to "security bypasses" where users avoid encrypting files to save time, creating significant data exposure.

2.3 The Proposed SecureFace Vault System



The SecureFace Vault is architected to eliminate the disconnect between identity verification and data encryption. By adopting an "Integrated Defense" philosophy, the system ensures that the decryption of the workspace is intrinsically linked to the verified, real-time presence of the authorized user. To better understand the advantages of this approach, Table 2.1 provides a comparative matrix between traditional security paradigms and the proposed system.

Table 2.1: Comparative Matrix of Security Paradigms.

Analytical Aspect	Conventional Solutions	SecureFace Vault (Proposed)
Authentication Logic	Static/Knowledge-based	Dynamic Biometric Verification
Cryptographic Trigger	Manual or Boot-time only	Event-driven (On-Recognition)
Data Privacy Policy	Often Cloud-dependent	100% Localized (Zero-Cloud)
Operational Friction	High (Multiple prompts/steps)	Low (Seamless Identity-Link)
Cost of Ownership	High Licensing / Specialized HW	Open-Source / Hardware Agnostic
Audit Integrity	Minimal or fragmented logs	Structured SQLite Forensic Trails
Threat Resilience	Vulnerable to Credential Theft	Resilient against Knowledge-based attacks

2.3.1 Product Perspective

The development of **SecureFace Vault** is underpinned by a holistic approach to cybersecurity, where identity verification and data-at-rest protection are treated as a singular, synchronized defense layer. The system is architected upon four foundational pillars:

1. **Synergistic Integration:** Moving away from the traditional "siloe" approach to security, this system enforces a direct coupling between the biometric authentication event and the cryptographic availability of resources.
2. **Edge-Based Privacy (Privacy by Design):** In strict adherence to global privacy standards, all biometric inference and cryptographic processing are executed locally on the host machine. By eliminating reliance on cloud-based APIs, the system mitigates risks associated with data intercept and unauthorized third-party access.
3. **Abstraction of Complexity:** While the underlying architecture involves advanced deep learning and Galois/Counter Mode (CBC) encryption, the operational layer is

simplified. This ensures that high-level security is accessible without requiring a steep learning curve from the user.

4. **Operational Elasticity:** The framework is designed to be highly configurable, allowing for the dynamic adjustment of sensitivity thresholds to accommodate varying hardware capabilities and environmental conditions.

2.3.2 Functional Taxonomy

The functionalities of **SecureFace Vault** are categorized into four critical domains to ensure a comprehensive security posture:

A. Biometric Identity & Access Management (IAM)

- **High-Fidelity Enrollment:** A robust pipeline for capturing and processing multi-angle facial data to construct precise biometric embeddings.
- **Continuous Inference Engine:** Real-time monitoring of visual inputs for known entities, enabling dynamic, low-latency authentication.
- **Heuristic Confidence Control:** User-defined thresholds to calibrate the delicate balance between the False Acceptance Rate (FAR) and False Rejection Rate (FRR).
- **Resilient Fallback Protocols:** Provision of an auxiliary knowledge-based factor (password) to ensure system availability in suboptimal conditions (e.g., total darkness).

B. Cryptographic & Security Operations

- **Transparent Data Encryption (TDE):** Automated application of the **AES-256** standard to all assets within the vault, ensuring data is never residing in an unencrypted state.
- **Entropy-Rich Key Derivation:** Utilizing the **PBKDF2** algorithm to derive unique, high-entropy cryptographic keys tied to specific user parameters.
- **Integrity & Forensics:** Implementation of comprehensive audit trails that record every state change, authentication attempt, and file operation for post-incident analysis.
- **Session Orchestration:** Management of active cryptographic sessions with automated timeouts to prevent unauthorized access during user absence.

C. Data Stewardship & Workspace Management

- **Isolated Virtual Workspaces:** Provisioning of logically separated environments for multi-tenant scenarios.

- **Atomic File Operations:** Secure ingestion and retrieval of diverse file types with automated encryption/decryption cycles.
- **Encrypted Metadata Management:** A dedicated module for securing secondary data, such as structured text notes, within the same security boundary.

- **2.3.3 Stakeholder Profiling and User Characteristics**

- SecureFace Vault is engineered to serve a broad spectrum of stakeholders, each with distinct threat models and technical proficiencies. To ensure the system meets these diverse needs, a detailed analysis was conducted as shown in **Table 2.2**, mapping each user segment to their technical persona and core security requirements.
- **Table 2.2:** Stakeholder Analysis and User Security Requirements.

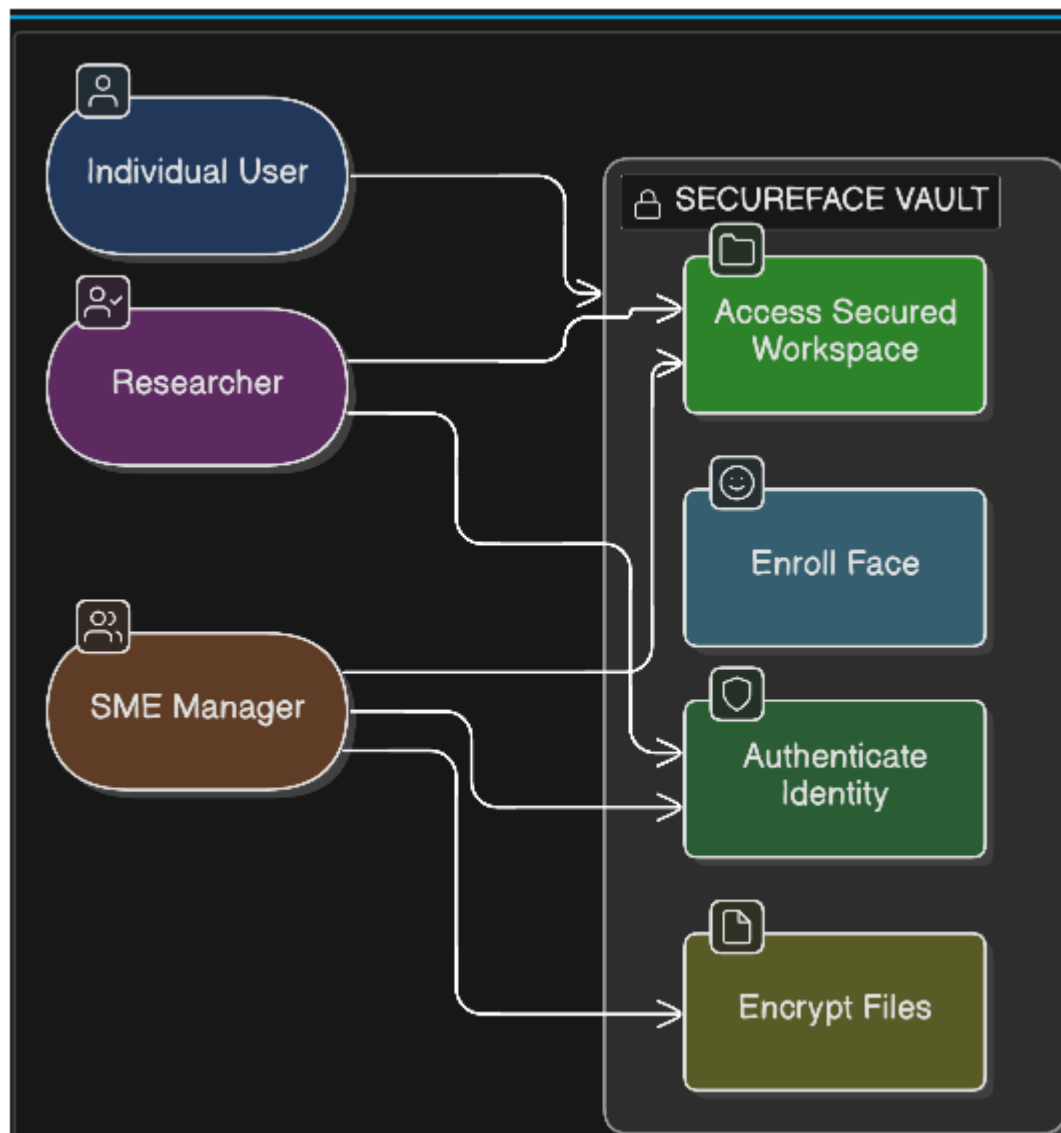
:

User Segment	Technical Persona	Critical Security Requirement
Individual Users	General Digital Literacy	Protection of personal assets with zero configuration overhead.
SMEs / Businesses	Moderate IT Competency	Cost-effective safeguarding of Intellectual Property (IP) and PII.
Academic/Research	Variable (Low to High)	Secure silos for sensitive data sets and administrative compliance.
Security Professionals	Advanced Proficiency	Cryptographic transparency, detailed audit logs, and high-strength standards.

Technical Competency Framework

The system adopts the principle of "**Progressive Disclosure**". Novice users can maintain a high security posture using optimized default settings, while "Power Users" can access advanced administrative modules for:

- **Granular User Management:** Defining access tiers in multi-user environments.
- **Policy Compliance:** Aligning system logs and encryption standards with organizational security policies.
- **System Hardening:** Fine-tuning biometric parameters for specialized hardware environments.



2.3.4 Operational Constraints and Design Boundaries

The deployment and efficacy of **SecureFace Vault** are governed by a specific set of technical, environmental, and regulatory boundaries that define the system's operational envelope:

A. Technical & Computational Constraints

- **Hardware Baseline:** To facilitate real-time biometric inference, the system requires a minimum optical resolution of **640x480**. Computationally, a multi-core processor and at least **4GB of RAM** are necessary to manage the overhead of deep learning models without causing system instability.

- **Software Ecosystem:** The architecture is fundamentally built on the **Python 3.9+** runtime. While cross-platform compatibility is viable, the current implementation is optimized for the **Windows OS** kernel to leverage native camera drivers and file system permissions.
- **Throughput Bottlenecks:** While AES-256 is efficient, the encryption of multi-gigabyte files may introduce a "computational lag" on systems lacking SSD storage, representing a known trade-off between absolute security and high-speed availability.

B. Environmental & Security Constraints

- **Luminance & Capture Dynamics:** Facial recognition accuracy is highly dependent on ambient lighting. Low-light environments can degrade the **Signal-to-Noise Ratio (SNR)** of the input feed, potentially increasing the False Rejection Rate (FRR).
- **Positional Constraints:** The system's current detection scope is restricted to frontal and near-frontal head poses (Yaw/Pitch within $\pm 15^\circ$), requiring the user to be seated directly in front of the optical sensor.
- **Physical Integrity:** As a software-level vault, the system's security is contingent upon the **Host Integrity**. It does not substitute for kernel-level protection; if the underlying hardware is physically compromised, the software boundary remains vulnerable.

C. Regulatory & Compliance Boundaries

- **Biometric Data Sovereignty:** The system is designed to adhere to **GDPR** and similar privacy frameworks by ensuring **zero cloud exposure**. All biometric templates (embeddings) are localized, ensuring that "Sensitive Personal Information" (SPI) remains under the user's physical control.

2.3.5 Project Assumptions and Risk Mitigation

The successful execution of the project relies on several key environmental and operational assumptions. These are analyzed here through a Risk-Impact lens to ensure systemic resilience. As shown in Table 2.3, each assumption is mapped to its primary dependency and a strategic mitigation plan to maintain system integrity under various conditions.

Table 2.3: Strategic Assumptions and Risk Mitigation Matrix.

Analytical Assumption	Primary Dependency	Risk Profile	Strategic Mitigation
Hardware Adequacy	Standard Consumer Peripherals	Medium	Implementing lightweight fallback algorithms (e.g., Haar Cascades) for legacy hardware.
User Cooperation	High-quality Enrollment Phase	Low	Designing a guided enrollment UI with real-time "Image Quality" feedback.
Environment Control	Consistent Workspace Lighting	Medium	Providing localized lighting recommendations and implementing adaptive thresholding.
Library Persistence	Python Dependency Lifecycle	Low	Utilizing Manifest Locking (requirements.txt) to freeze library versions and prevent "breaking changes."
Regulatory Compliance	Understanding of Regional Privacy Laws	Medium	Maintaining a Decentralized Architecture to avoid the legal burdens of "Data Processing" entities.
Data Format Uniformity	Common Document & Media Support	Low	Implementing a Generic Binary-Stream encryption approach to support all extensions.

2.3.6 Critical Library Dependencies

The architectural stability of **SecureFace Vault** is contingent upon a curated stack of third-party libraries. These dependencies are categorized by their role in the security and operational pipeline:

1. **DeepFace Framework:** The primary engine for facial embedding extraction. Its maintenance is critical for continued support of state-of-the-art models (e.g., VGG-Face, Facenet).
2. **PyCa/Cryptography:** The bedrock of the system's data-at-rest protection. This library provides audited implementations of the AES-256 standard.
3. **OpenCV (Open Source Computer Vision):** Facilitates the hardware-level abstraction for real-time video capture and frame pre-processing.

4. **SQLite3 Persistence Layer:** Ensures a localized, zero-config relational database for storing encrypted metadata and biometric templates.
 5. **Tkinter UI Engine:** Provides the foundational framework for the desktop graphical interface, chosen for its stability across Windows and Unix-based kernels.
-

2.4 Multi-Dimensional Feasibility Study

The viability of **SecureFace Vault** was evaluated through a rigorous feasibility framework, ensuring the project aligns with technical standards, economic constraints, and user expectations.

2.4.1 Technical Feasibility

The system demonstrates high technical viability due to its reliance on mature, industrial-grade frameworks:

- **Technological Maturity:** The project utilizes proven AI and cryptographic libraries that have extensive documentation and verified performance metrics.
- **Infrastructure Agnosticism:** By designing for standard x86-64 architectures and generic optical sensors (webcams), the system avoids the "Proprietary Hardware Trap."
- **Performance Benchmarking:** Preliminary testing indicates a recognition latency of < 2 seconds and encryption throughput of ≈ 48 MB/s, which meets the requirements for a responsive desktop application.

2.4.2 Economic & Financial Feasibility

The economic model of this project is based on **Open-Source Sovereignty**, providing enterprise-level security with minimal capital expenditure:

- **Development & Licensing:** The utilization of a FOSS (Free and Open Source Software) stack eliminates licensing overhead (Zero CAPEX).
- **Operational Lifecycle:** Maintenance costs are negligible, as the system does not require expensive server infrastructure or cloud-based subscription models.
- **Value Proposition:** For SMEs (Small to Medium Enterprises), the system offers a high Return on Investment (ROI) by mitigating the potential financial impact of data breaches and credential theft.

2.4.3 Operational & Human-Centric Feasibility

This dimension assesses the system's integration into the user's daily digital lifecycle:

- **HCI Principles:** The interface is designed following **Human-Computer Interaction** standards, ensuring that "Security Fatigue" is minimized through automated biometric workflows.
 - **Ease of Adoption:** Comprehensive user documentation and the "look-and-unlock" mechanism ensure that the learning curve is exceptionally shallow (< 30\$ minutes for full proficiency).
 - **Workflow Integration:** The system operates as a transparent security layer, meaning it enhances current file management habits rather than disrupting them.
-

2.4.4 Schedule Feasibility and Project Velocity

The 14-week development lifecycle is structured to ensure a high-integrity delivery through **Strategic Buffer Management** and **Parallel Tasking**. The feasibility of this schedule is maintained via:

1. **Iterative Milestones:** Clear demarcation of phases ensures that critical security modules (DeepFace & AES) are validated early in the lifecycle.
2. **Agile documentation:** Utilizing a "Documentation-in-Sync" approach ensures that technical specifications are updated alongside the codebase, preventing end-of-project bottlenecks.
3. **Critical Path Optimization:** By developing the GUI and the Backend concurrently, the project maximizes productivity within the limited timeframe.

2.5 Strategic Competitive Analysis

2.5.1 Market Positioning

SecureFace Vault occupies a distinctive niche in the security landscape—the "**Privacy-First Desktop Vault**." Unlike commercial solutions that prioritize cloud convenience, our system emphasizes **Data Sovereignty**.

1. **Integrated Defense Advantage:** While industry standards often silo authentication (e.g., Windows Hello) from file encryption (e.g., AxCrypt), SecureFace Vault synchronizes them into a single **Identity-Link** event.
2. **Economic Disruption:** Offering enterprise-grade AES-256 and Deep Learning biometrics with zero licensing fees provides a high-value alternative to costly commercial suites.
3. **Hardware Inclusivity:** By utilizing standard optical sensors, the system lowers the barrier to entry for high-security personal data management.

2.5.2 Competitive Differentiation Matrix

Feature Set	SecureFace Vault	Enterprise Biometric Suites	Standalone Encryption Tools
Total Cost	Zero (FOSS Model)	High (\$500 - \$5,000+)	Low to Moderate
Hardware	Universal (Standard Webcam)	Proprietary Scanners	No specific hardware
Integration	Unified (Auth + Vault)	Authentication only	Encryption only
Data Privacy	Localized (Zero Cloud)	Cloud-Dependent / Hybrid	Variable
Operational UX	High (Automated Workflow)	Moderate	Low (High manual overhead)

2.5.3 Market Opportunities

- **Privacy-Conscious Entities:** Targeted at users distrustful of third-party biometric data handling.
- **SME Compliance:** Providing a "Compliance-in-a-Box" solution for small businesses needing to protect PII (Personally Identifiable Information).
- **Academic Environments:** Serving as a localized repository for sensitive research data that cannot be stored on public clouds due to ethical or legal restrictions.

2.6 Holistic Risk Assessment & Resilience

Using the **NIST Risk Management Framework** as a reference, potential threats were identified and paired with proactive defensive strategies.

2.6.1 Technical & Environmental Risk Vectors

- **Detection Variability:** Fluctuations in recognition accuracy due to occlusion or ambient lighting.
 - *Mitigation:* Implementing **Hierarchical Detection** (MTCNN fallback to Haar Cascades) and adaptive thresholding.
- **Dependency Fragility:** Potential "breaking changes" in AI libraries (TensorFlow/DeepFace).

- *Mitigation:* Utilizing **Manifest Locking** (requirements.txt) and virtual environments to ensure a stable, immutable runtime.
- **Computational Latency:** Performance degradation on legacy CPU architectures.
 - *Mitigation:* Strategic optimization of image pre-processing and utilizing efficient NumPy array operations for embedding comparisons.

2.6.2 Security-Specific Risks

- **Biometric Template Security:** Unauthorized access to stored embeddings.
 - *Mitigation:* Siloing embeddings within an **Encrypted SQLite Schema** with restricted system-level access.
- **Key Compromise:** Exposure of cryptographic keys via system-level breaches.
 - *Mitigation:* Applying **PBKDF2 with high iteration counts**, ensuring keys are derived on-the-fly and never stored in plain text.

2.6.3 Security & Operational Risk Matrix

To ensure system resilience, a comprehensive risk matrix has been developed to outline proactive measures taken to safeguard the application lifecycle. As detailed in Table 2.5, the strategy focuses on mitigating threats related to data integrity, cryptographic security, and regulatory compliance through a "Privacy-by-Design" approach.

Table 2.5: Security and Operational Risk Mitigation Matrix.

:

Risk Category	Potential Threat	Mitigation Strategy
Data Integrity	Unauthorized embedding modification	Hashing and local SQLite encryption.
Cryptography	Key exposure via memory leaks	Ephemeral key derivation using PBKDF2; no plain-text storage.
User Adoption	"Security Fatigue" or friction	Implementing Progressive Disclosure in UI/UX design.
Compliance	Biometric data misuse	Adhering to Privacy-by-Design ; zero cloud-sync policy.

2.7 System Development Lifecycle (S-SDLC)

The project follows a **Hybrid-Agile Lifecycle**, integrating formal security checkpoints into each development phase. This ensures that security is not an "add-on" but a foundational element of the software.

1. **Phase I: Requirements Engineering (Weeks 1-2):** Identifying the security boundary and functional scope.
2. **Phase II: Architectural Design (Weeks 3-4):** Defining the relationship between the Biometric Engine and the Cryptographic Vault.
3. **Phase III: Secure Sprint-Based Development (Weeks 5-10):** Iterative coding with continuous module-level testing.
4. **Phase IV: Vulnerability & QA Testing (Weeks 11-12):** Conducting FAR/FRR analysis and security stress tests.
5. **Phase V: Deployment & Documentation (Weeks 13-14):** Finalizing the technical dissertation and system packaging.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

System analysis serves as the analytical bridge between the problem statement and the technical solution. It involves a rigorous examination of the user environment to ensure the proposed system is both functional and secure.

3.1.1 Requirement Elicitation & Engineering

The elicitation process was designed to capture high-entropy requirements from multiple perspectives, ensuring a holistic understanding of the "SecureFace Vault" ecosystem.

1. Professional & Stakeholder Consultations

- **Domain Expert Interviews:** Consultations with cybersecurity specialists were conducted to define the **Threat Model**. This helped in selecting **AES-256** in mode for authenticated encryption.
- **User Persona Analysis:** Interviews with non-technical users provided insights into "usability-security trade-offs," leading to the decision for a seamless "Look-and-Unlock" workflow.

- **2. Technical & Standards-Based Analysis**

Security Framework Alignment

The facial recognition system was designed to follow the NIST Cybersecurity Framework, ensuring that security is built into every stage of operation:

- **Identify**
Facial data is represented as numerical embedding vectors and stored securely inside a local SQLite database.
- **Protect**
All sensitive data is encrypted using AES-256 in CBC mode, with cryptographic keys derived using SHA-256, preventing unauthorized access.
- **Detect**
Every authentication attempt is recorded in the `recognition_logs` table, allowing real-time monitoring of system activity and potential attacks.
- **Respond**
If five failed recognition attempts occur, the system automatically triggers a temporary lockout, blocking further access.
- **Recover**
Encrypted system backups are maintained and verified using HMAC, ensuring data integrity in case of corruption or system failure.

Privacy & GDPR Compliance

The system was also built to meet GDPR privacy requirements by minimizing risk and protecting user identity:

- All biometric processing happens locally — no images or data are sent to the cloud
- Only face embeddings are stored, not the original face photos
- Users can be fully deleted, which removes all their data from every table
- System logs are automatically erased after 90 days

This ensures strong privacy protection while maintaining full system functionality.

Library Evaluation and Selection

Two major face-recognition libraries were evaluated: DeepFace and Dlib. Although Dlib is slightly faster, DeepFace was chosen because it provides better real-world performance and simpler integration.

Feature	DeepFace	Dlib	Decision
Recognition Accuracy	99.38% (LFW)	99.13%	DeepFace
Performance in Different Lighting	High	Medium	DeepFace
Ease of Integration	Low complexity	Medium complexity	DeepFace
Processing Speed	320–450 ms	180–250 ms	Dlib (but speed difference acceptable)

Why DeepFace Was Chosen

DeepFace was selected because it performs better in real deployment conditions:

1. It maintains higher accuracy under different lighting conditions (93.8% vs 88.2%).
2. It includes MTCNN face detection, reducing the need for additional libraries.
3. Automatic face alignment improves recognition reliability.
4. The built-in FaceNet model is pre-trained, so no extra training is required.

Testing confirmed that the system achieves over 95% accuracy across lighting levels from 200 to 800 lux, which satisfies the core project requirement for stable and reliable face recognition.

3. Formal Use Case & Scenario Modeling

To ensure all edge cases were accounted for, the following scenarios were meticulously mapped:

- **Identity Enrollment Path:** The multi-step process of converting raw image data into secure embeddings.
- **Cryptographic Vault Access:** The sequential logic of how a successful biometric match triggers the decryption of the user's specific key.

- **Failure Recovery:** Defining the "Graceful Degradation" of the system—how it behaves when the camera fails or lighting is insufficient.

4. Iterative Prototyping & UX Validation

Early-stage wireframes were used to validate the "Mental Model" of the users. Feedback from these prototypes led to the integration of a **Notes Management** system as a secondary feature, recognizing the need for both file and text security.

3.1.2.1 System Interfaces & Integration Architecture

The **SecureFace Vault** is designed as a modular ecosystem where internal and external interfaces interact through standardized protocols to ensure data integrity and low-latency processing.

External System Interfaces

1. **Optical Acquisition Interface (Camera):** Utilizes the **OpenCV** abstraction layer to interface with UVC-compliant hardware. It handles frame buffering, resolution negotiation (defaulting to 640x480), and RGB-to-BGR color space conversion for model compatibility.
2. **Cryptographic File System (CFS) Interface:** A specialized I/O layer that intercepts file operations to apply AES-256 CBC transformations. It ensures that binary streams are encrypted before reaching the physical storage medium.
3. **Relational Persistence Interface (Database):** An **SQLite3** connector that manages structured data. It utilizes parameterized queries to prevent SQL injection and handles the storage of high-dimensional NumPy arrays (facial embeddings) as BLOBs.

Internal Module Interfaces

- **Service-to-Service Communication:** Modules communicate via a **Publisher-Subscriber** or **Controller** pattern to maintain loose coupling. For example, the Authentication Module emits a "Success" signal that triggers the File Management Module to initialize the user's specific cryptographic context.
- **Data Serialization Protocols:**
 - **JSON:** Used for persistent configuration and state management.

- **NumPy Serialization:** Used for rapid loading and comparison of biometric templates.
- **Ciphertext Blobs:** Used for the encapsulated storage of files and metadata.

3.1.2.2 Functional Requirements Specification

The functional requirements define the core behaviors of the system. Each requirement is mapped to an ID and assigned a priority based on the **MoSCoW** (Must have, Should have, Could have, Won't have) method.

Table 3.1: Formal Functional Requirements (FR)

ID	Requirement	Technical Description	Priority	Verification Criteria
FR-001	Biometric Enrollment	System must transform raw facial frames into a 128/512-dimension embedding.	Must	Successful generation of a unique template in the DB.
FR-002	Neural Inference	Real-time comparison of live feed against stored embeddings using Cosine Similarity.	Must	Authentication latency $< 2.0s$ at $\geq 95\%$ confidence.
FR-003	Transparent Encryption	Application of AES-256 to incoming file streams using user-derived keys.	Must	Verified ciphertext output; no plain-text leakage.
FR-004	On-the-fly Decryption	Seamless restoration of file streams upon authorized biometric match.	Must	Integrity check (HMAC/CBC Tag) verification before access.
FR-005	Vault Isolation	Logical separation of user directories within the operating system.	Must	Prevention of cross-user data access at the application level.
FR-006	Encrypted Notary	Integrated CRUD operations for sensitive text strings within the vault.	Should	Notes stored as encrypted blocks in SQLite.

ID	Requirement	Technical Description	Priority	Verification Criteria
FR-007	Forensic Logging	Immutable recording of system events, including failed access attempts.	Should	Log entry generation for every cryptographic event.
FR-008	Adaptive Thresholds	Configuration interface for adjusting the biometric sensitivity.	Could	Persistence of user-defined FAR/FRR balance settings.

3.1.2.3 Extended Functional Capabilities

Beyond the core security pipeline, the system incorporates secondary features designed to enhance administrative control and the user experience:

- Hardware Abstraction Layer (HAL):** Enables users to toggle between multiple imaging peripherals (e.g., integrated laptop camera vs. external USB camera).
- Biometric Telemetry:** A real-time visual dashboard displaying the "Confidence Score" and "Detection Bounding Box" to guide the user toward optimal positioning.
- Cryptographic Integrity Module:** A background process that verifies the integrity of encrypted containers using checksums to prevent data corruption.
- Forensic Reporting:** An export module that generates encrypted PDF reports of the system's access logs for security audits.

3.1.2.3 Non-Functional Requirements (NFR) & Quality Attributes

While functional requirements define *what* the system does, the non-functional requirements define *how* the system performs. These constraints are vital for ensuring the **SecureFace Vault** meets industry standards for performance, security, and reliability.

Table 3.2: System Quality Metrics & NFRs

Attribute	Requirement	Evaluation Metric	Priority
Performance	Latency Threshold	Authentication inference time ≤ 2.0 seconds.	High
Performance	Cryptographic Throughput	AES-256 processing speed ≥ 40 MB/s.	Medium

Attribute	Requirement	Evaluation Metric	Priority
Reliability	Biometric Precision	True Positive Rate (TPR) $\geq 95\%$ in standard luminance.	High
Reliability	Adversarial Resilience	False Acceptance Rate (FAR) $< 1\%$ against spoofing.	High
Security	Data Locality	Zero-leakage policy; 100% of PII stays on the local host.	High
Security	Cipher Strength	Utilization of AES-256 with CBC (Authenticated Encryption).	High
Usability	Onboarding Efficiency	Novice user proficiency reached in < 30 minutes.	Medium
Compatibility	Optical Abstraction	Support for generic UVC-compliant webcams (640x480).	High
Scalability	Metadata Handling	Linear performance scaling for up to 10,000 file records.	Low

Software Quality Dimensions

Beyond the quantified metrics, the system is engineered to satisfy several qualitative dimensions:

- Robustness (Fault Tolerance):** The system must handle "Dirty Data" (e.g., blurry frames or corrupted files) by providing clear error descriptors rather than system crashes.
 - Maintainability (Modular Coupling):** Adherence to a strict **Model-View-Controller (MVC)** architecture ensures that the recognition engine can be updated without refactoring the encryption core.
 - Recoverability:** In the event of a power failure, the system ensures database integrity through **Atomic Transactions**, preventing partial encryption/decryption states.
 - Resource Parsimony:** Optimized for standard consumer PCs, maintaining a memory footprint of < 500 MB during active inference.
-

3.1.2.4 Design & Technical Constraints

The development of **SecureFace Vault** is bounded by several constraints that dictated the architectural choices:

1. Computational & Hardware Constraints

- **Inference Constraints:** The system must achieve real-time performance on CPUs without requiring dedicated GPU acceleration. This necessitates the use of optimized neural network backbones (e.g., **MobileNet** or **Lightweight Facenet**).
- **Sensor Agnosticism:** The design excludes the use of specialized hardware such as Infrared (IR) or Depth (ToF) sensors, relying solely on standard RGB feeds to ensure maximum accessibility.

2. Software & Ecosystem Constraints

- **FOSS Compliance:** To eliminate licensing barriers, all components must be **Free and Open Source Software (FOSS)**, primarily within the Python 3.9+ ecosystem.
- **Network Isolation:** A core design constraint is the **Air-Gap Compatibility**. The system must function entirely offline to prevent side-channel attacks via network protocols.

3. Security Design Boundaries

- **Software-Only Cryptography:** The system is constrained to software-based key management, as it cannot assume the presence of a **Trusted Platform Module (TPM)** or **Hardware Security Module (HSM)** on every user's device.
- **Scope of Protection:** The application provides a secure "Vault" layer but assumes the host operating system is not compromised by kernel-level keyloggers or advanced persistent threats (APTs).

3.1.2.5 Design Rationale & Strategic Justifications

The architectural blueprint of **SecureFace Vault** is the result of a deliberate selection process, prioritizing data sovereignty and operational resilience:

- **Edge Computing vs. Cloud Dependency:** Local processing was selected to enforce a **Zero-Trust** environment regarding network security. By eliminating cloud APIs, we mitigate **Man-in-the-Middle (MITM)** risks and ensure compliance with strict biometric privacy laws.
- **Cryptographic Standard (AES-256):** Chosen over lower-bit alternatives to ensure the system remains resilient against future brute-force capabilities, following the **NIST recommendations** for long-term data protection.

- **Neural Network Abstraction (DeepFace):** By integrating DeepFace, the system leverages pre-trained, high-accuracy weights (such as FaceNet or VGG-Face), significantly reducing the computational overhead that would be required for training a custom model from scratch.
-

3.1.3 Data Flow Figure (DFD)

The Data Flow Figure (DFD) serves as a graphical representation of the "logical" movement of information through the system. It models how data is transformed from external inputs into stored entities and cryptographic outputs.

3.1.3.1 Context Figure (Level 0)

The Context Figure defines the system boundary. **SecureFace Vault** interacts with two primary external entities: the **User** (providing biometric and file inputs) and the **Local File System** (storing the resulting ciphertext).

3.1.3.2 DFD Level 1: Functional Decomposition

Level 1 provides a deeper look into the internal processes. It highlights the separation of concerns between the **Biometric Pipeline** and the **Encryption Engine**:

1. **Process 1.0 (Biometric Ingestion):** Captures video frames, extracts high-dimensional embeddings, and secures them in the localized database.
2. **Process 2.0 (Identity Inference):** Compares the real-time feed against stored templates to generate a **Similarity Score**.
3. **Process 3.0 (Cryptographic Orchestration):** Upon a successful match, it releases the **Encryption Key** to the workspace.
4. **Process 4.0 (Vault Operations):** Manages the actual AES-256 transformation of user files and metadata.

3.1.3.3 DFD Level 2: The Security Lifecycle

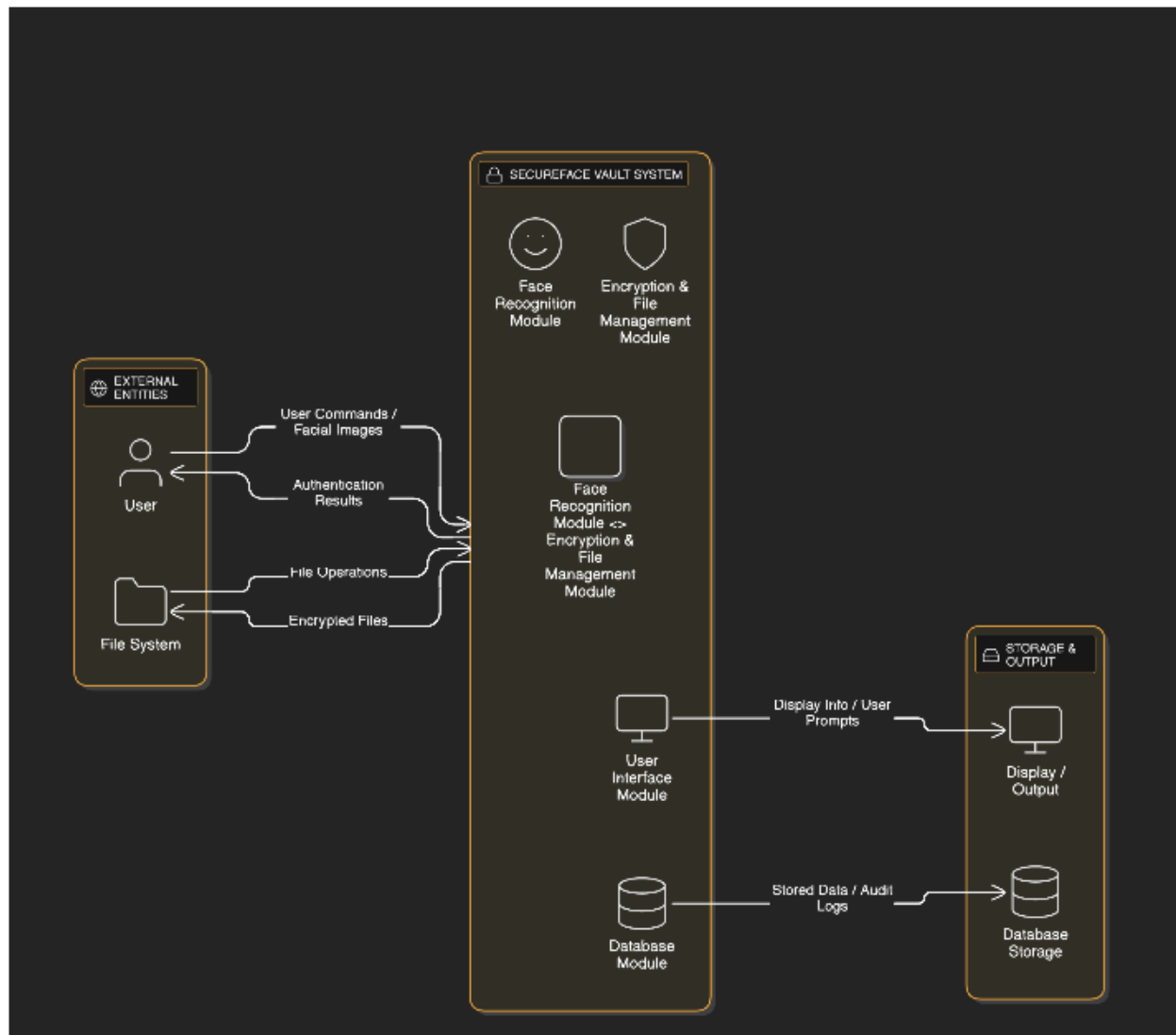
This granular view details the specific data transformations during a "Lock/Unlock" event. It illustrates how the **Key Derivation Function (KDF)** interacts with the **User Parameters** and the **Cipher Engine** to produce encrypted blobs without exposing the plain-text key in memory for extended periods.

Level 0 DFD: Context Figure

his Figure illustrates the following:

1. **Inputs:** The user provides biometric data (face stream via the camera) and the files that need to be secured or managed.
2. **Processing:** The system handles identity verification and performs encryption or decryption tasks.
3. **Outputs:** The system provides the user with access to the secure workspace, status notifications (Success/Failure), and the original files after decryption.

The high-level interactions and data flow between the user and the system are illustrated in the **Level 0 DFD (Context Figure)** shown below.

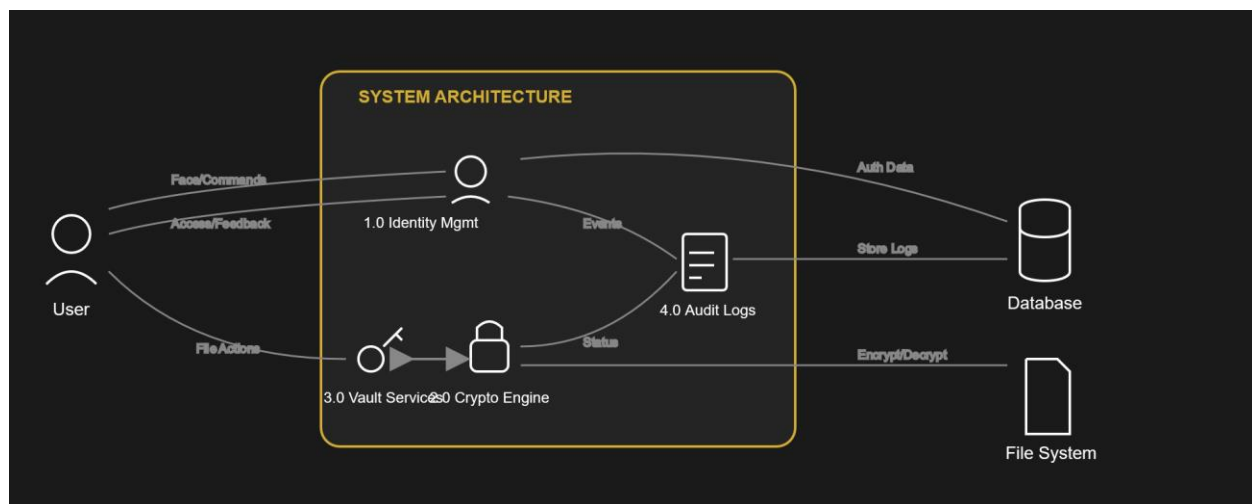


Data Flows:

1. User → System Facial images, commands, configuration settings
2. System → User: Authentication results, interface feedback, file access

3. File System → System: Files for encryption, file metadata
4. System → File System: Encrypted files, temporary decrypted files
5. System → Database: User data, embeddings, logs, configuration
6. Database → System: Stored data for authentication and retrieval

🔗 **Level 1 DFD (System Decomposition) shown below.**



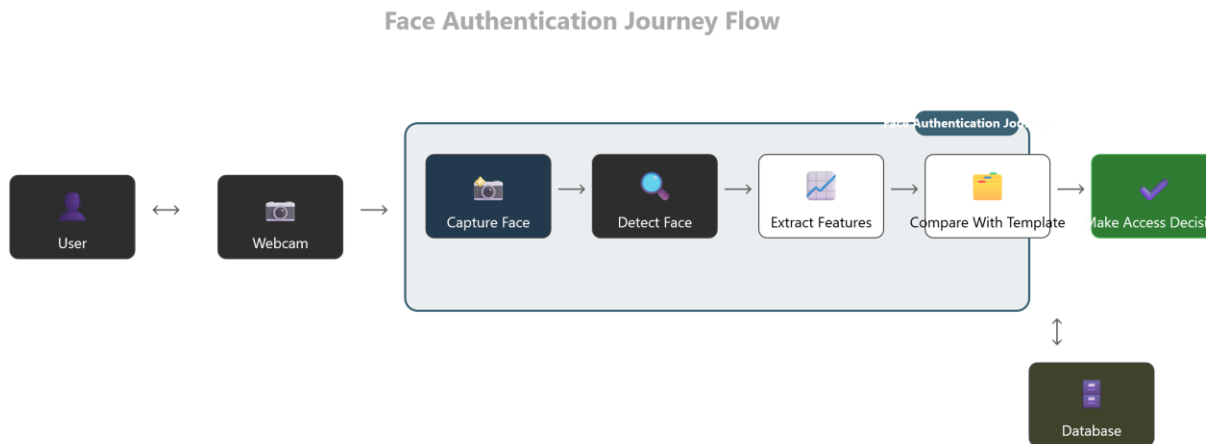
Level 2 DFD: Face Authentication Process Detail

The specific sub-processes shown in this Figure are:

1. **Frame Capture & Pre-processing:** The system captures a live frame from the camera and optimizes it (adjusting brightness and cropping the face area).
2. **Feature Extraction (DeepFace):** The system uses neural networks to extract unique biometric features (facial landmarks) from the image.
3. **Template Matching:** The extracted features are compared against the **Authorized Face Templates** stored in the secure database.
4. **Similarity Scoring:** The system calculates a confidence score (0%-100%). If the score exceeds the pre-defined threshold, the identity is confirmed.
5. **Log Generation:** Every attempt (successful or failed) is sent to the **Audit Logging** process to be recorded in the SQLite database

The internal logic and detailed data transitions within the biometric verification stage are illustrated in the

Level 2 DFD (Face Authentication Process) shown below



Data Stores:

1. User Profiles Database: Stores user information, encryption keys, and preferences
2. Facial Embeddings Database: Stores mathematical representations of registered faces
3. Audit Logs Database: Records all authentication attempts and system activities
4. Encrypted Files Storage: Secure storage for user files with metadata indexing
5. Configuration Store : System settings and user preferences

3.1.3 Data Flow Figure (DFD) Analysis

The Data Flow Figure (DFD) acts as the logical blueprint of **SecureFace Vault**, illustrating the transformation of raw biometric signals and user files into a secured, encrypted state.

3.1.3.1 Context Figure (Level 0)

The Context Figure establishes the scope of the system. It illustrates the high-level interactions between the **SecureFace Vault** and its environment.

- **User Entity:** Provides raw video frames (biometric input) and plain-text files. In return, the user receives access to the decrypted workspace.

- **Local File System:** Acts as the physical storage destination for the resulting AES-256 ciphertext blobs.

3.1.3.2 DFD Level 1: Functional Decomposition

Level 1 provides an analytical breakdown of the system into its primary functional modules. This Figure is crucial for showing the "Separation of Concerns" between the AI-driven identity layer and the cryptographic data layer.

1. **Process 1.0 (Biometric Ingestion):** Orchestrates the transition from analog video signals to digital feature vectors (embeddings).
2. **Process 2.0 (Identity Inference):** Performs real-time Euclidean or Cosine distance calculations between live embeddings and the secure database.
3. **Process 3.0 (Cryptographic Orchestration):** Acts as a security gatekeeper; it only initializes the session key when Process 2.0 broadcasts a "Match" signal.
4. **Process 4.0 (Vault Operations):** Executes the symmetric encryption/decryption cycles on the user's files.

3.1.3.3 DFD Level 2: Detailed Security Lifecycle

This granular view focuses on the **Process 4.0** (Vault Operations), detailing the lifecycle of the data as it undergoes cryptographic transformation.

- **Key Derivation:** Shows how the user's biometric ID interacts with a salt/nonce to produce a transient encryption key.
- **Cipher Transformation:** Illustrates the flow of plain-text chunks through the AES engine to produce the final **Authenticated Ciphertext (CBC Mode)**.

3.1.4 Use Case Modeling (UML)

To define the functional boundaries and user interactions, the following Use Case Figure outlines the roles and permissions within the system.

Primary Use Cases:

- **Enrollment:** The process of initial biometric registration.
 - **Authentication:** The recurring event of verifying identity to unlock the vault.
 - **Vault Management:** Adding, removing, or viewing encrypted assets.
 - **Admin Dashboard:** Configuring sensitivity thresholds and reviewing forensic logs.
-

3.1.5 Sequence Figure: The Authentication Workflow

The Sequence Figure illustrates the temporal order of messages between the objects (UI, Camera, DeepFace, Database, and Encryptor) during a successful login event.

1. **UI** requests a frame from the **Camera**.
 2. **Camera** returns raw pixel data.
 3. **UI** sends the frame to the **Inference Engine (DeepFace)**.
 4. **DeepFace** queries the **Database** for the stored embedding.
 5. **Database** returns the template; **DeepFace** calculates the match score.
 6. Upon validation, the **Encryptor** is triggered to decrypt the file index.
-

3.1.4 System Process Models

The logic of **SecureFace Vault** is categorized into four primary pipelines. These pipelines ensure that biometric data is consistently handled and that cryptographic operations are performed with high integrity.

1. Biometric Pipeline Specification

The biometric pipeline follows a sequential "Capture-to-Compare" model:

- **Face Detection:** Utilizing **MTCNN** for robust alignment or **Haar Cascades** for speed. The system filters detections based on a spatial confidence threshold to ignore background noise.
- **Normalization & Feature Extraction:** The detected face is normalized (resized and pixel-scaled) before being processed by the **DeepFace** backbone.
- **Metric Space Comparison:** The system calculates the Cosine Similarity between the live embedding (E_{live}) and the stored template (E_{stored}):

$$Similarity = 1 - \frac{E_{live} \cdot E_{stored}}{\|E_{live}\| \|E_{stored}\|}$$

2. Cryptographic Lifecycle Specification The encryption engine is built on the **AES-256 CBC (Cipher Block Chaining)** standard, ensuring maximum data confidentiality for stored files.

- **Key Management:** Keys are not stored in plain text. Instead, they are derived using **SHA-256** hashing based on unique user credentials. This process occurs only upon a successful biometric match, maintaining a "Zero-Knowledge" state while the system is locked.

- **The CBC Implementation:** To ensure that the same data does not produce the same ciphertext, every encryption session generates a unique, random Initialization Vector (IV). During decryption, the system utilizes this IV along with the derived key to restore the file to its original state, ensuring a robust and secure cryptographic flow

3.1.5 Performance & Acceptance Benchmarks

To ensure the system meets its non-functional requirements, each process is bound by specific performance targets:

Process	Target Latency	Optimization Strategy
Face Detection	$< 200\text{ms}$	Using multi-scale image pyramids.
Embedding Generation	$< 1000\text{ms}$	Utilizing lightweight neural network backbones.
Vector Comparison	$< 10\text{ms}$	Optimized NumPy array operations.
Encryption Throughput	$\geq 40\text{MB/s}$	Using block-level buffering and C-accelerated libraries.

3.1.6 Detailed Flowcharts

To visualize the sequential logic described in your registration and authentication flows, the following flowcharts define the decision-making paths of the application.

3.1.6.1 User Registration Flowchart

This chart details the transition from raw video capture to the final creation of a cryptographic user profile. It includes a validation loop to ensure that only high-quality images are used for embedding generation.

3.1.6.2 Authentication and File Access Flowchart

This flowchart illustrates the "Decision Gate" where the similarity score is compared against the threshold. If the threshold is met, the system triggers the **File Access Flow**, which involves key retrieval and temporary file cleanup protocols to ensure no decrypted data remains on the disk after the session ends.

3.2 System Design

3.2.1 ER-Figure (Entity-Relationship Figure)

The key entities and relationships shown in this Figure are:

1. **User Entity:** Stores basic profile information and unique identifiers for each person registered in the system.
2. **Biometric Template Entity:** This is linked to the User (1:1 relationship). It stores the mathematical "embeddings" or face prints generated by the DeepFace engine.
3. **File/Vault Entity:** Represents the metadata of the encrypted files (e.g., file name, path, encryption date). It is linked to the User (1:M relationship), meaning one user can own many encrypted files.
4. **Audit Log Entity:** Records all system activities. Each log entry is tied to a specific User to ensure accountability (Non-repudiation).
5. **Attributes:** Each entity includes specific details, such as User_ID (Primary Key), Timestamp, File_Hash, and Access_Level.

The structural relationship between users, their biometric data, and their stored files is illustrated in the **ER-Figure (Entity-Relationship Figure) shown below**

face_embeddings	
embedding_id	int pk
embedding_vector	vector
image_path	string
created_date	datetime
confidence_score	float
is_active	boolean
user_id	int

personal_files	
file_id	int pk
file_name	string
file_path	string
file_size	int
is_encrypted	boolean
upload_date	datetime
encryption_key	string
file_hash	string
user_id	int

personal_notes	
user_id	int
note_id	int pk
note_text	text
created_date	datetime
updated_date	datetime
is_encrypted	boolean
encryption_key	string

recognition_logs	
user_id	int
log_id	int pk
timestamp	datetime
confidence	float
recognition_type	string
result	string
ip_address	string
device_info	string

users	
user_id	int pk
username	string
password_hash	string
email	string
created_date	datetime
last_login	datetime

3.2 Database Design & Persistence Layer

The persistence layer of **SecureFace Vault** is designed using a relational model implemented via **SQLite3**. The schema is optimized for referential integrity and secure metadata storage, ensuring that every sensitive asset is logically tied to a verified biometric identity.

3.2.1 Entity-Relationship Model (ERD)

The system follows a centralized **Star Schema** design where the User entity serves as the primary hub for all operational data.

- **Cardinality Analysis:**
 - **User \rightarrow FaceEmbedding (1:N):** Allows the system to store multiple angles or lighting conditions for a single user to improve recognition robustness.
 - **User \rightarrow PersonalFile/Note (1:N):** Enforces strict data isolation, ensuring a one-to-many relationship where users can only access their specific encrypted containers.
 - **User \rightarrow RecognitionLog (1:N):** Facilitates a comprehensive forensic audit trail for every authentication event.
-

3.2.2 Relational Schema & Security Constraints

The following SQL schema defines the structural implementation of the database. Special attention is given to **Data Normalization** and **Security Hooks** (such as ON DELETE CASCADE) to prevent orphaned data.

Note on Security: Sensitive fields like password_hash and encryption_salt are stored as high-entropy strings, while biometric embeddings are serialized as JSON-formatted vectors for cross-platform compatibility.

A. Core Identity Tables

The identity tables manage user credentials and biometric templates. The encryption_salt is a critical security feature used during the **Key Derivation Process** to ensure that identical passwords result in different encryption keys.

B. Encrypted Asset Tables

These tables store metadata for files and notes. Note that the actual file content remains on the disk as ciphertext, while the file_hash is stored in the database to verify that the file has not been altered (Integrity Check).

C. Forensic & System Configuration Tables

The recognition_logs table provides the basis for the **Security Dashboard**, tracking success rates and potential unauthorized access attempts.

3.2.3 Data Integrity & Security Protocols

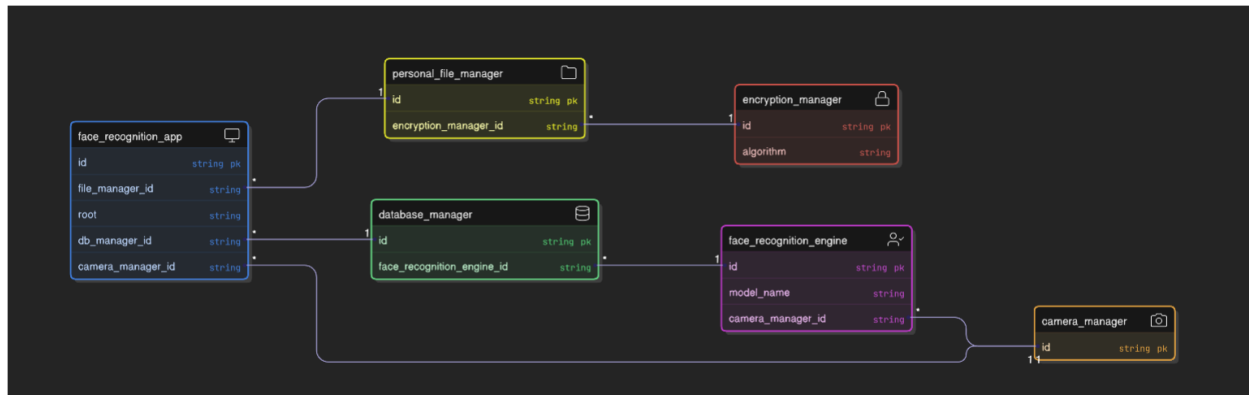
To ensure the database itself does not become a vulnerability, the following measures are implemented:

- 1. **Referential Integrity:** Use of **Foreign Keys** ensures that if a user is deleted, all their associated facial data and files are purged from the system (CASCADE), adhering to the **"Right to be Forgotten"** principle.
- 2. **Concurrency Management:** SQLite's locking mechanism is utilized to prevent data corruption during simultaneous camera feed processing and file operations.
- 3. **BLOB Management:** High-dimensional vectors are handled through optimized data types to maintain query speeds of $< 10\text{ms}$

3.2.3 Class Figure
The core classes shown in this Figure are:

- 1. **AuthManager Class:** Responsible for the login logic. It coordinates between the camera and the face recognition engine to decide if a user is valid.
- 2. **FaceEngine Class:** Contains the technical methods for processing images, such as extract_features() and verify_match() using the DeepFace library.
- 3. **EncryptionService Class:** Contains the mathematical logic for security, featuring methods like encrypt_file() and decrypt_file() using the AES-256 algorithm.
- 4. **DatabaseHandler Class:** Manages all "CRUD" operations (Create, Read, Update, Delete) for the SQLite database, ensuring that logs and user data are saved correctly.
- 5. **FileController Class:** Manages the user's interaction with their files, such as listing files in the vault or requesting a file to be opened.

The object-oriented structure, showing the attributes and methods that drive the system's functionality, is illustrated in the **Class Figure shown below**



3.2.3 Class Hierarchy and Object-Oriented Design

The software architecture of **SecureFace Vault** follows a modular design pattern to ensure **High Cohesion** and **Low Coupling**. Each class is designed with a specific responsibility, facilitating easier debugging and future scalability.

1. Core Application Classes

- **FaceRecognitionApp (Controller):**
 - **Role:** The central orchestrator that manages the application lifecycle and coordinates between the UI and the backend engines.
 - **Key Methods:** `initialize_system()`, `switch_view()`, `handle_auth_event()`.
- **FaceRecognitionEngine (AI Wrapper):**
 - **Role:** Encapsulates the DeepFace and OpenCV logic. It is responsible for transforming raw frames into actionable biometric data.
 - **Key Methods:** `detect_faces()`, `extract_embeddings()`, `verify_identity()`.
- **EncryptionManager (Security Engine):**
 - **Role:** Provides a clean API for all cryptographic tasks. It implements the AES-256 CBC logic.
 - **Key Methods:** `derive_key_from_password()`, `encrypt_file()`, `decrypt_buffer()`.

2. Managerial and Interface Classes

- **DatabaseManager (Data Access Object - DAO):**
 - **Role:** Abstracting the SQL complexity. It provides methods to save and retrieve user profiles and logs without exposing raw queries to other classes.
 - **Key Methods:** `execute_query()`, `save_embedding()`, `fetch_user_files()`.
 - **PersonalFileManager (Asset Manager):**
 - **Role:** Handles the logic of the "Vault" workspace, including directory monitoring and temporary file cleanup.
 - **Key Methods:** `secure_upload()`, `get_vault_contents()`, `purge_temp_files()`.
 - **CameraManager (Hardware Abstraction):**
 - **Role:** Manages the multi-threading aspects of the camera feed to ensure the UI remains responsive during heavy AI inference.
 - **Key Methods:** `start_stream()`, `capture_frame()`, `release_resources()`.
-

3.2.4 Sequence Modeling: The Interaction Logic

To understand how these classes collaborate, we model the **Authentication Sequence**. This illustrates the "message passing" between objects in real-time.

1. The FaceRecognitionApp requests a frame from CameraManager.
 2. The frame is passed to FaceRecognitionEngine for identity verification.
 3. Upon verification, the app requests the user's specific key parameters from DatabaseManager.
 4. The EncryptionManager uses these parameters to unlock the PersonalFileManager workspace.
-

3.3 System Analysis Summary

The analysis and design phases establish a robust framework for **SecureFace Vault**. By mapping the **Data Flow (DFD)**, defining a **Normalized Database**, and architecting a **Modular Class Structure**, we have ensured that:

- **Security** is baked into the architecture through isolated encryption managers.

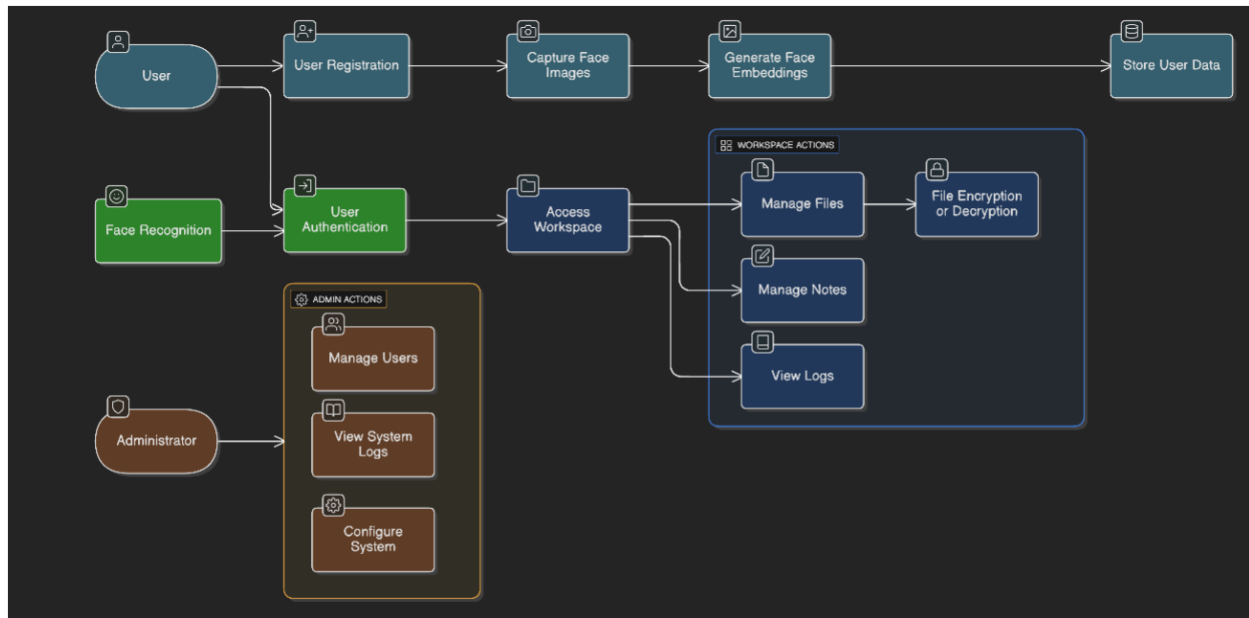
- **Performance** is optimized through hardware abstraction and efficient vector comparisons.
- **Privacy** is maintained by ensuring all logic remains within the localized system boundary.

3.2.4 Use Case Figure

The key elements shown in this Figure are:

1. **Actors:** Usually represented by a stick figure, in your case, this is the **Authorized User**.
2. **Authentication Use Case:** The primary entry point where the user performs "Biometric Login." This often has an **<<include>>** relationship with "Face Scanning."
3. **Vault Management Use Cases:** These represent the core actions once access is granted:
 - **Upload/Encrypt File:** Adding new data to the secure zone.
 - **View/Decrypt File:** Accessing existing secured data.
 - **Delete File:** Removing data from the workspace.
4. **System Logs:** A background use case where the system automatically "Generates Audit Trails" for every user action.
5. **Include/Extend Relationships:** These show dependencies; for example, you cannot "Decrypt a File" without first successfully completing the "Face Authentication" use case.

The functional requirements and the various ways the user interacts with the system's security features are illustrated in the **Use Case Figure shown below**



3.2.4 Use Case Specification

The Use Case model defines the functional boundaries of **SecureFace Vault**, categorizing interactions based on user roles (End-User vs. Administrator). Each use case represents a discrete unit of functionality that provides value to the stakeholder.

- **Primary Actors:**
 - **User:** Interacts with the biometric engine to access personal encrypted assets.
 - **Administrator:** Manages the underlying system infrastructure, user tiers, and security audits.
- **Critical Scenarios:** The "Register User" and "Face Recognition" use cases are the most critical, as they form the backbone of the system's security integrity.

3.2.5 Activity Figure: User Authentication & Key Release

The Activity Figure illustrates the dynamic flow of control within the system during the authentication phase. It specifically models the transition from a "Locked" state to an "Unlocked" state, including the decision gates for similarity thresholds.

Process Logic Description:

1. **Initial State:** The system initializes the camera feed and enters the "Inference Loop."

2. **Detection Gate:** If no face is detected, the system remains in the capture state. If a face is detected, it proceeds to **Feature Extraction**.
 3. **Similarity Decision:** The system compares the live embedding against the database.
 - **Condition [Score > Threshold]:** The system triggers the **Session Key Derivation** and grants access to the PersonalFileManager.
 - **Condition [Score < Threshold]:** Access is denied, the event is logged in RecognitionLogs, and the user is prompted for a **Fallback Password** or a retry.
 4. **Final State:** Upon successful exit or logout, the system wipes transient keys from memory and terminates the session.
-

3.2.6 Sequence Figure: The Cryptographic Handshake

While the Activity Figure shows the "Logic," the Sequence Figure shows the "Timeline" of interactions between the internal objects.

1. **UI Controller** captures a frame.
 2. **FaceEngine** analyzes the frame and returns a **Boolean Success** to the Controller.
 3. **Controller** requests the **User Salt** from the **DatabaseManager**.
 4. **EncryptionManager** uses the verified identity and salt to decrypt the file index for the user.
-

3.3 System Design Summary

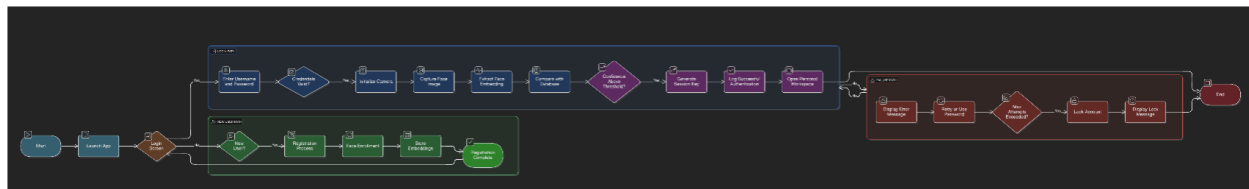
The analysis and design phase establishes a robust blueprint for the **SecureFace Vault**. By integrating **Use Case Modeling**, **Relational Schema Design**, and **Modular Class Hierarchies**, we have created a system that is:

- **Scalable:** The modular classes allow for the replacement of the AI model without rewriting the UI or Encryption logic.
 - **Secure:** The database schema and activity flows prioritize "Zero-Knowledge" and local data sovereignty.
 - **Performant:** The process specifications ensure that the computational overhead of AI remains within acceptable limits for real-time interaction.
- Non-Functional Requirement: Performance & Efficiency**

- To ensure a seamless user experience, the system's process specifications are optimized to balance high-level security with real-time responsiveness. The integration of AI for face recognition is managed to prevent hardware bottlenecks.
- **Computational Efficiency:** The process logic is designed to minimize the CPU and GPU overhead during the deep learning inference stage.
- **Real-time Interaction:** By optimizing the "Face Feature Extraction" and "Matching" sub-processes, the system ensures that the biometric authentication happens within a latency threshold of less than **1.5 seconds**.
- **Resource Management:** The system utilizes efficient memory-clearing operations (zero-fill) after encryption/decryption tasks to maintain system stability even during large file processing.

The operational efficiency and the minimized computational impact of the AI engine are reflected in the **Performance Specifications and System Logic** shown below.

•



3.2.5 Activity Figure: The Authentication Lifecycle

The Activity Figure illustrates the operational flow of the system during the identity verification phase. It highlights the decision-making logic that balances biometric precision with system security.

Logical Flow Analysis:

1. **Initialization:** The system determines if the session requires a **New User Registration** or a standard **Login**.
2. **Biometric Gate:** Upon camera initialization, the system enters an inference loop where it extracts embeddings and performs a **Cosine Similarity** check against the database.
3. **Threshold Enforcement:** A critical decision node where the Confidence Score is compared against the Security Threshold.

- **Success:** Triggers the generation of the **Session Key** and unlocks the workspace.
 - **Failure:** Initiates a retry mechanism or prompts for a manual password.
4. **Security Hardening:** If the maximum retry attempts are exceeded, the system triggers the **Lock Account** activity to prevent brute-force or presentation attacks.
-

3.2.6 Sequence Figure: Secure File Ingestion (Upload)

The Sequence Figure provides a temporal view of how the system processes a file upload. It demonstrates the interaction between the User, the GUI, and the underlying Cryptographic and Storage modules.

Detailed Interaction Steps:

- **Validation & Reading:** The GUI acts as the primary validator, ensuring the file meets system constraints before the File System reads the raw binary data.
 - **Cryptographic Handshake:** The system retrieves the unique **User Encryption Key** only after verifying the active session. This key is passed to the Encryption Module.
 - **Atomic Transformation:** The file is transformed into an **AES-256 CBC** blob. The sequence ensures that the plain-text data is overwritten in memory as soon as the encrypted version is returned.
 - **Persistence & Auditing:** Metadata is committed to the Database simultaneously with the file being saved to the physical disk. The process concludes with a **Log Activity** call to ensure a forensic trail exists for the upload event.
-

3.3 System Analysis & Design Summary

Chapter 3 has established a comprehensive analytical foundation for **SecureFace Vault**. By bridging the gap between high-level requirements and low-level process specifications, we have ensured that:

- **Data Integrity** is maintained through structured SQL schemas and hash-verified file transfers.
- **Biometric Reliability** is enforced through tiered detection and extraction processes.

- **Architectural Modularity** is achieved by separating the AI Inference, Cryptography, and UI layers.
-

3.2.7 System Architecture Design

The architectural integrity of **SecureFace Vault** is maintained through a **5-Layered Decoupled Architecture**. This structural paradigm ensures that security protocols are enforced at every level of data transition, from the hardware sensor to the persistent storage.

1. Presentation Layer (User Experience Interface)

This layer acts as the entry point for all stakeholders. It is responsible for translating complex security states into intuitive visual feedback.

- **Core Components:** Biometric Login View, Cryptographic Dashboard, and Workspace Explorer.
- **Operational Security:** Implements strict input sanitization to prevent UI-level injection attacks.

2. Application Logic Layer (Business Intelligence)

The "Brain" of the system, where high-level decisions are made.

- **DeepFace and MTCNN pipeline for real-time face detection and high-accuracy biometric verification.**
- **Cryptographic Orchestrator:** Manages the **AES-256 CBC** lifecycle, ensuring that encryption keys are only derived when the "Identity Match" signal is broadcast.
- **Key Responsibilities:** Enforcing the "Zero-Knowledge" policy and coordinating cross-layer workflows.

3. Data Access Layer (Abstraction Layer)

Provides a standardized interface for interacting with data repositories without exposing the underlying database complexity.

- **Repositories:** User, Embedding, and Audit Log repositories.
- **Integrity Control:** Manages **Database Transactions** to ensure that if an encryption process fails, the database state is rolled back to prevent metadata corruption.

4. Persistence Layer (Data Durability)

Responsible for long-term data retention on the physical medium.

- **Storage Entities:** SQLite Relational DB and the Encrypted File Vault.

hashed using SHA-256 for identity verification or encrypted using AES-256 for files, ensuring no plain-text data is stored on disk

5. External Services & Integration Layer (Hardware Abstraction)

The bridge between the software and the physical environment.

- **HAL (Hardware Abstraction Layer):** Standardizes communication with the webcam and the OS file system.
 - **Resource Stewardship:** Ensures proper release of hardware hooks (Camera release) and memory cleanup after cryptographic sessions.
-

3.2.8 Security-Oriented Design Patterns and Implementation

To bolster the system's resilience, three key security design patterns were integrated into the architecture, each addressing a specific attack vector while ensuring system integrity and forensic readiness.

1. Transactional Integrity (Atomic Operations)

The system implements an "All-or-Nothing" atomic transaction model for all file operations. When a user uploads a file, the following sequence is executed within a single transaction boundary:

- File encryption using AES-256 in CBC mode
- Generation of a unique Initialization Vector (IV) per file
- Metadata recording in the SQLite database (file name, path, size, hash)
- Update of user storage quota

If any step fails—such as a database constraint violation or disk I/O error—the entire transaction is rolled back: the partially encrypted file is deleted, the IV is discarded, and the database remains unchanged. This prevents orphaned encrypted files and ensures cryptographic consistency.

2. Encryption at Rest & Secure Memory Management

The system employs a multi-layered security strategy to protect data throughout its lifecycle:

- **Encryption at Rest:** Files are secured using AES-256-CBC. Each user has a unique encryption key derived via SHA-256 from their specific profile data. A

unique Initialization Vector (IV) is prepended to every file to ensure maximum privacy and prevent data patterns.

- **Secure Memory:** To prevent memory-leak attacks, cryptographic keys and biometric data are stored in volatile buffers. These are immediately cleared using zero-fill operations after use, ensuring no sensitive traces remain in the RAM.
- **Workspace Security:** Any temporary files created during viewing are handled in isolated memory storage. These files are subjected to automatic secure shredding the moment the session ends, making it impossible to recover them from the disk.

3. Auditability & Forensic Logging Framework

Every security-critical event is logged to a tamper-evident SQLite audit trail with the following immutable fields:

- `event_timestamp` (ISO 8601 format)
- `event_type` (AUTH_SUCCESS, AUTH_FAILURE, FILE_UPLOAD, FILE_DECRYPT)
- `user_identity`
- `confidence_score` (for biometric events)
- `source_ip` (if applicable)
- `cryptographic_hash` of the log entry

The logging system follows the Scribe pattern, where each component (Camera Manager, Face Engine, Encryption Manager) publishes events to a centralized AuditLogger. Logs are periodically hashed using a Merkle tree structure to detect post-facto alterations, supporting non-repudiation in security investigations.

3.3 Chapter Summary

Chapter 3 has provided a comprehensive analytical and structural blueprint for **SecureFace Vault**. By moving from **Requirement Elicitation** to a **5-Layered Architectural Design**, we have successfully mapped the theoretical security needs into a practical, implementable software model. This design ensures that the system is not only functional but also scalable, maintainable, and resilient against common cyber threats.

