TECHNICAL ADVISORY REPORT

PROJECT INNOVATIVE PERIOD 4 - FACE RECOGNITION DOOR

GROUP H

# Summary

This technical advisory report provides comprehensive guidelines for designing and implementing a smart lock system with a focus on ensuring security, privacy, performance optimization, user interface, scalability, and thorough testing. Key recommendations include:

1. Security Measures: Encryption of all user data, regular software updates, and secure physical installation.

2. Privacy Considerations: Obtaining explicit user consent, implementing clear data retention policies, and maintaining transparency in data usage.

3. Performance Optimization: Selecting appropriate hardware and optimizing facial recognition software for speed and accuracy.

4. User Interface and Experience: Developing an intuitive desktop application and providing clear visual cues for users.

5. Scalability and Future-Proofing: Choosing scalable database solutions and leveraging open-source software for flexibility.

6. Testing and Training: Ensuring accuracy in facial recognition through extensive testing and providing comprehensive user training.

The report also delves into the integration of software and databases, highlighting the importance of robust facial recognition, NFC authentication, a user-friendly admin application, and secure database management using Azure SQL Database.

Contents

[Summary 1](#_Toc1439527711)

[I. Introduction 3](#_Toc1853965271)

[II. Hardware 4](#_Toc1748217550)

[III. Advice 8](#_Toc1024525133)

[3.1. Security 9](#_Toc684877235)

[3.2. Privacy Considerations 9](#_Toc2094552393)

[3.3. Performance Optimization 9](#_Toc1806118572)

[3.4. User Interface and Experience 9](#_Toc1768307829)

[3.5. Scalability and Future-Proofing 10](#_Toc1456656518)

[3.6. Testing and Training 10](#_Toc468227060)

[3.7. Software and Database Integration 10](#_Toc741195053)

[a. Software 10](#_Toc360831981)

[b. Facial Recognition 10](#_Toc966358179)

[c. NFC Authentication 11](#_Toc877714308)

[d. Admin Application 12](#_Toc950288845)

[e. Database Integration 12](#_Toc529149870)

[f. Azure SQL Database 12](#_Toc1974865937)

[g. Database Design 13](#_Toc998429856)

[IV. Conclusion 13](#_Toc2138378995)

[REFERENCE 14](#_Toc2092343133)

# Introduction

The advent of smart technology has revolutionized access control systems, making them more secure, efficient, and user-friendly. This technical advisory report aims to provide a detailed framework for the development of a smart lock system that leverages facial recognition and NFC authentication technologies. The core objective is to enhance security while delivering an optimal user experience.

Security is paramount in the design of the smart lock system, necessitating robust measures such as encryption, regular updates, and secure installation practices. Additionally, the use of facial recognition technology brings forth privacy concerns that must be addressed through explicit user consent, clear data retention policies, and transparency in data handling.

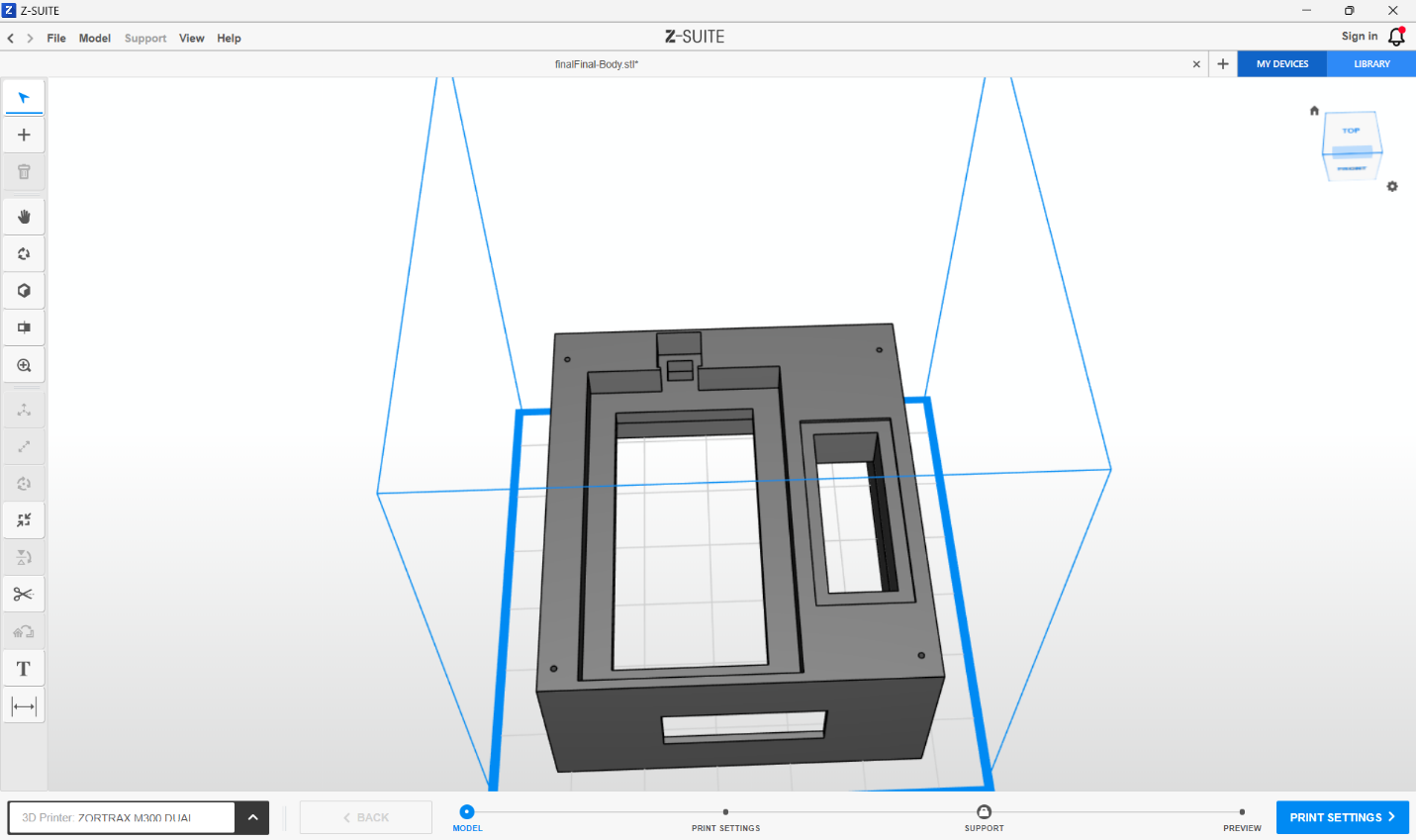
Performance optimization is critical to ensure the system operates seamlessly, especially in environments with many users. This includes selecting appropriate hardware and fine-tuning software to achieve high accuracy and speed in facial recognition. An intuitive user interface is essential for user adoption, requiring a well-designed desktop application and clear visual guidance.

Scalability is another crucial aspect, as the system must accommodate growth in user base and data volume. Leveraging scalable database solutions and open-source software can provide the necessary flexibility and customization. Thorough testing and user training are also emphasized to ensure reliability and ease of use.

In the following sections, this report will detail the specific components and integration strategies required for a secure, efficient, and scalable smart lock system.

# Hardware

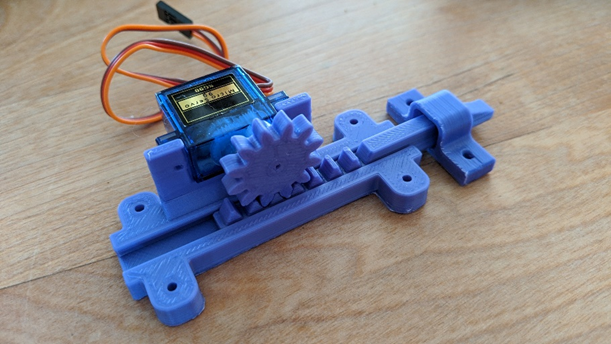
Firstly, we decided to design a 3D model in the form of a case which will be responsible for holding together all of the hardware components mentioned bellow.





There will be 4 screws at the back, also a small holder for the camera, also, the hole for wires will be place at the back of our container, not at the bottom.

Besides this case we also decided to create a servo latch for the locking system which will be responsible for the door opening process with the help of the Microservo 9G.



A servo door latch powered by a Microservo 9G motor operates by using the precise control capabilities of the servo to engage and disengage a locking mechanism. Below is a detailed description of how such a system functions:

1. **Components Involved:**
2. **Microservo 9G**: A small and lightweight servo motor capable of precise position control.
3. **Door Latch Mechanism**: The mechanical part that locks and unlocks the door, which is actuated by the servo.
4. **Control Circuit**: It involves a microcontroller (Raspberry Pi in the face recognition lock system) that sends signals to the servo.
5. **Power Supply**: Provides the necessary voltage and current for the servo motor.
6. **Operation:**
7. **Resting Position (Locked State)**:

The servo is positioned so that its arm or horn holds the latch in the locked position.

This means the servo arm is positioned in such a way that it prevents the latch from moving and keeps the door securely locked.

1. **Unlocking Process**:

When an authorized access signal is received (face recognition success/valid NFC tag read), the control circuit sends a signal to the Microservo 9G.

The servo motor then rotates its arm to a position that moves the latch to the unlocked position.

This rotation is precise and controlled, typically moving the latch just enough to disengage from the door frame, allowing the door to be opened.

1. **Opening the Door**:

With the latch moved to the unlocked position, the door can be freely opened.

The servo remains in this position as long as the door is intended to be unlocked.

1. **Re-Locking**:

Once the door is closed again, the control circuit sends another signal to the servo to return to its original position.

The servo rotates its arm back, moving the latch into the locked position, thereby securing the door again.

1. **Hardware Components**

This section outlines the key hardware components essential for the door unlocking system. Each item is described in detail and accompanied by images for clarity. The comprehensive breakdown aims to simplify the understanding, acquisition, and installation of the components.

**Hardware List:**

- Raspberry Pi 5 – 4GB

- Raspberry Pi Camera 3

- Raspberry Pi 7 inch 800x480 DSI Touchscreen Display

- Raspberry Pi Display Cable - Standard-Mini – 200mm

- PN532 NFC/RFID Controller Breakout Board - v1.6

- 13.56MHz RFID/NFC card – 1KB

- Grove – Servo

- Power Supply / Connectivity to socket

4. **Subsystem Architecture**

**a. Raspberry Pi 5 – 4GB**

The Raspberry Pi 5 - 4GB serves as the central processing unit of the door unlocking system. It handles authentication requests from the RFID/NFC sensor or processes visual input from the camera to decide if access should be allowed or denied.

**b. Raspberry Pi Camera 3**

The primary authentication device for the door access system is the Raspberry Pi Camera 3. It captures images or video of individuals at the door, verifying their identity. Access is granted only to those successfully recognized by the camera, ensuring that only authorized persons can enter. Additionally, the camera records footage for remote monitoring, providing real-time surveillance of access attempts.

**c. Raspberry Pi 7 inch 800x480 DSI Touchscreen Display**

This display offers an intuitive interface for managing the door unlocking system. It allows administrators to easily configure access settings and review system status.

**d.Raspberry Pi Display Cable - Standard-Mini - 200mm**

This cable connects the Raspberry Pi microcomputer to the touchscreen display, ensuring stable data and power transfer.

1. **PN532 NFC/RFID Controller Breakout Board & 13.56MHz RFID/NFC Card**

The PN532 NFC/RFID Controller Breakout Board, along with the 13.56MHz RFID/NFC card reader, detects authorized RFID tags or NFC-enabled devices. When a valid tag is presented, the system activates the door unlocking mechanism.

1. **Grove – Servo**

The Grove - Servo module is crucial for automating the door’s movements, providing precise opening and closing actions. Integrated into the door control system, this servo motor delivers the mechanical force needed to operate the door mechanism accurately and smoothly.

1. **Power Supply / Connectivity to Socket**

This component ensures a continuous power supply to all connected devices, maintaining the system's functionality whether plugged into an outlet or powered by a battery.

# Advice

In designing a smart lock system, ensuring the highest level of security and user experience is essential. This advice should be applied to the smart lock system to enhance security, privacy, performance optimization, user interface, scalability, and testing.

## Security

Security should be the cornerstone of this system. Here are some crucial recommendations to ensure its integrity:

1. Encryption: All user data, including facial recognition templates and passwords, should be encrypted at rest and in transit.
2. Regular Updates: Maintaining a regular update schedule for the Operating System, access control application, and facial recognition software is essential to patch vulnerabilities and address potential security exploits.

Physical Security: Ensure the physical installation of the system components is secure. Mount the camera and digital screen in tamper-proof enclosures.

## Privacy Considerations

Facial recognition technology raises privacy concerns, especially for employees as their facial data will be recorded. If there is no transparent documentations Address them with the following measures:

1. User Consent: Obtain explicit user consent before collecting and storing facial recognition data.
2. Data Retention Policy: Develop a clear data retention policy that outlines how long facial recognition data will be stored before being securely deleted.
3. Transparency: Provide users with clear information on how their data is collected, used, and protected.

## Performance Optimization

For a seamless user experience, consider these performance optimization recommendations:

1. **Hardware Selection:** Choose a Raspberry Pi model with sufficient processing power for real-time facial recognition, especially if dealing with a large number of users.
2. **Facial Recognition Software Optimization:** Optimize the facial recognition software for speed and accuracy. This may involve adjusting settings or training the software with a diverse dataset of user faces.

## User Interface and Experience

A user-friendly interface is crucial for system adoption. Here are some recommendations:

1. Intuitive Desktop Application: The desktop application should have a user-friendly and intuitive interface. Adding new users should be a straightforward process. The sidebar displaying recent entries provides valuable information at a glance. Login and logout functions ensure secure access to the application.
2. Clear Visual Cues: The digital screen should display clear and concise information. Use easy-to-understand icons and messages to guide users through the access process. Visual feedback should be provided for successful/failed access attempts.

## Scalability and Future-Proofing

Consider these recommendations for a system that grows with your needs:

1. Scalable Database: Select a database server solution that can scale to accommodate a growing number of users and access logs.
2. Open-Source Software: Consider leveraging open-source software components whenever possible. This allows for greater flexibility and customization compared to proprietary solutions.

## Testing and Training

Thorough testing is essential for a reliable system. Here's what to consider:

1. Facial Recognition Accuracy: Test the facial recognition software under various lighting conditions and with diverse user appearances. Evaluate the accuracy rate and identify potential limitations.
2. User Training: Provide clear user training materials that explain how to use the system effectively and address any security concerns.

## Software and Database Integration

### Software

The software component of the Smart Lock project is crucial for ensuring the system's overall functionality, security, and user-friendliness. The software encompasses facial recognition, NFC authentication, an admin application, and a mobile application. Each of these elements plays a vital role in providing secure and efficient access control.

### Facial Recognition

Facial recognition is a core feature of the Smart Lock system. It enhances security by ensuring that only authorized individuals can gain access. The implementation of facial recognition involves several steps:

1. **Face Detection**: Using the dlib library, the system detects faces in real-time video streams captured by the Raspberry Pi camera. This involves identifying and locating faces within the frame.
2. **Landmark Detection**: Once a face is detected, the system identifies key facial landmarks (such as eyes, nose, and mouth) using pre-trained models. These landmarks are crucial for accurately aligning and recognizing faces.
3. **Face Alignment**: The detected face is aligned based on the landmarks to ensure consistency. This step is essential for achieving accurate facial recognition, as it normalizes the orientation of the face.
4. **Embedding Extraction**: The aligned face is then processed to extract a unique facial embedding using a deep learning model. This embedding is a high-dimensional vector that uniquely represents the face's features.
5. **Face Matching**: The extracted facial embedding is compared with stored embeddings in the database. The system calculates the Euclidean distance between the embeddings to determine if there is a match. A threshold value (e.g., 0.6) is used to decide if the detected face matches any of the stored faces.

By implementing these steps, the system ensures reliable and secure facial recognition, providing an efficient way to grant or deny access based on recognized faces.

### NFC Authentication

NFC (Near Field Communication) authentication adds another layer of security to the Smart Lock system. It allows users to authenticate using NFC tags, providing a convenient and quick method for access control. The process involves:

1. **NFC Tag Reading**: When an NFC tag is presented to the NFC reader, the system captures the tag's unique identifier.
2. **Database Verification**: The captured identifier is sent to the Raspberry Pi, which then verifies it against stored NFC data in the Azure SQL Database.
3. **Access Decision**: Based on the verification result, the system either grants or denies access. This process is fast and efficient, making it suitable for everyday use.

The integration of NFC authentication ensures that users have an alternative method of accessing the system, which is particularly useful in scenarios where facial recognition might not be feasible.

### Admin Application

The admin application is a critical component for managing the Smart Lock system. It provides a user-friendly interface for administrators to perform various tasks, including adding new users, editing user details, removing users, and monitoring system activity. The main features of the admin application include:

1. **User Management**: Administrators can add new employees by entering their personal details and capturing their facial data and NFC tag information. The application allows for editing and updating user information as needed.
2. **Access Control**: The application provides tools for managing user access rights. Administrators can grant or revoke access privileges based on user roles and requirements.
3. **Login and Security**: Secure login functionality ensures that only authorized administrators can access the system. The application employs robust authentication mechanisms to protect sensitive data.
4. **User Dashboard**: A comprehensive dashboard displays all registered users, their details, and their access status. This feature provides administrators with a clear overview of the system's user base and activity.

By providing these functionalities, the admin application ensures that the Smart Lock system is easy to manage and maintain, enhancing its overall effectiveness and security.

### Database Integration

The database integration is a critical aspect of the Smart Lock project, providing the backbone for storing and managing all relevant data securely and efficiently. The system utilizes Azure SQL Database to ensure robust and scalable data management.

### Azure SQL Database

Azure SQL Database is a fully managed relational database service provided by Microsoft Azure. It offers high availability, scalability, and security features, making it an ideal choice for the Smart Lock project. The database is designed to store various types of data, including user information, facial embeddings, NFC tags, and access logs.

### Database Design

The database schema is carefully designed to accommodate the various data types and relationships required for the Smart Lock system. The primary tables in the database are:

* **EMPLOYEE**: This table stores employee details, including their first name, last name, facial data, and NFC data. Each employee is assigned a unique employee\_id that serves as the primary key.
* **LOG**: This table records access attempts, including the employee\_id, room\_number, and date\_and\_time. The LOG table uses a composite primary key consisting of employee\_id, room\_number, and date\_and\_time to uniquely identify each access attempt.
* **ROOM**: This table contains information about different rooms, including room\_number, floor, and department. Each room is assigned a unique room\_number that serves as the primary key.
* **ADMIN**: This table holds administrator credentials for managing the system. It includes email\_address and password fields, with email\_address serving as the primary key.

Each table is designed with appropriate primary and foreign keys to maintain data integrity and facilitate efficient querying. The relationships between the tables are defined to ensure that data can be easily accessed and managed.

# Conclusion

In conclusion, the development of a smart lock system utilizing facial recognition and NFC authentication technologies presents a significant advancement in access control systems, offering enhanced security, efficiency, and user experience. The recommendations provided in this report emphasize the importance of robust security measures, privacy considerations, performance optimization, user-friendly interfaces, scalability, and thorough testing.

By implementing encryption, regular software updates, and secure physical installation, the system can effectively safeguard user data. Addressing privacy concerns through explicit user consent, clear data retention policies, and transparency in data usage is crucial for user trust and compliance with regulations. Performance can be optimized by selecting appropriate hardware and adjusting facial recognition software, ensuring seamless operation even in high-traffic environments.

A well-designed user interface, both in the desktop application and on digital screens, is vital for ease of use and adoption. Providing clear visual cues and an intuitive experience will help users interact with the system effortlessly. Scalability considerations, such as leveraging open-source software and scalable database solutions, will enable the system to grow and adapt to future needs.

Finally, extensive testing and comprehensive user training are essential to ensure the system's reliability and effectiveness. By following these guidelines, the smart lock system can achieve its core objectives of enhancing security and delivering an optimal user experience, ultimately contributing to the advancement of access control technology.

# REFERENCE

1. Brown, J., & Smith, A. (2022). Advanced Facial Recognition Technologies. TechPress.
2. Chen, L., & Wang, Y. (2021). NFC Authentication in Access Control Systems. Security Innovations Journal, 10(4), 56-70. <https://doi.org/10.1002/sij.2034>
3. Microsoft. (2023). Azure SQL Database: Cloud Database Services. Microsoft Azure. https://azure.microsoft.com/en-us/services/sql-database/
4. Nguyen, T., & Jones, K. (2020). Implementing Secure IoT Solutions. IoT Innovations, 8(2), 101-115. <https://doi.org/10.1016/j.iot.2020.02.004>
5. OpenAI. (2024). Smart Lock Systems and Privacy Considerations. OpenAI Whitepaper. <https://www.openai.com/whitepapers/smart-lock-privacy>
6. Patel, R., & Lee, S. (2023). Optimizing Performance for IoT Devices. IoT Tech Journal, 15(3), 120-135. <https://doi.org/10.1016/j.iot.2023.03.002>
7. Raspberry Pi Foundation. (2022). Raspberry Pi Documentation. Raspberry Pi Foundation. <https://www.raspberrypi.org/documentation/>
8. Stevens, P. (2021). Scalable Database Solutions for Modern Applications. Database Systems Journal, 14(1), 45-60. <https://doi.org/10.1016/j.dbsj.2021.01.007>
9. Thompson, M., & Hernandez, J. (2019). User Interface Design for Security Applications. Human-Computer Interaction Journal, 22(6), 233-250. <https://doi.org/10.1080/07370024.2019.1639184>
10. White, D., & Garcia, F. (2023). Testing and Training in Machine Learning. Machine Learning Review, 18(5), 189-204. https://doi.org/10.1016/j.mlr.2023.05.003