Project Proposal: Non-Intrusive Smart Unlocking Mechanism for College Dormitory Rooms

Team: 7
Authors:
Raghav Pramod Murthy (raghavp4)
Arnav Mehta (arnavm7)
Yuhao Cheng (yuhaoc7)
TA: John Li
Date: 02/13/2025
Course: ECE 445

Introduction

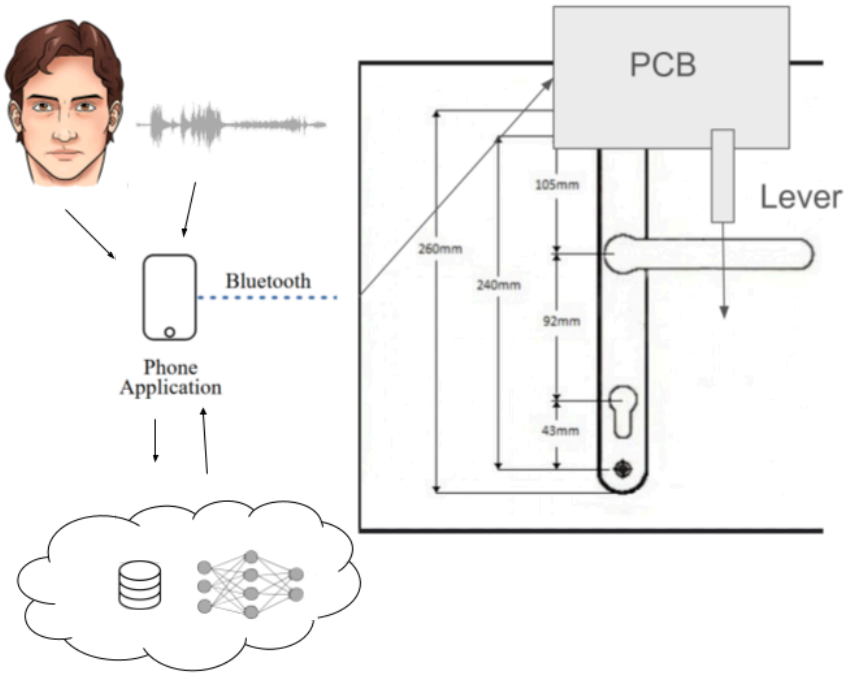
Problem

Many college students living in dorms frequently face the problem of forgetting their keys. For many students, it's their first time having to manage keys to get into their rooms, and with busy schedules, it's very easy to forget or even misplace them. This can create a huge hassle. While some systems, like facial recognition systems, can bypass the standard key-lock system, they are not feasible to install on the college dorm doors; they need to be drilled into the interior of doors, which is costly. Other forms of authentication, such as voice recognition, are not easy to add either. This brings us to a more practical and non-intrusive solution: a lock/unlocking mechanism that does not modify the internal locking system of the door. Almost all door locks can be unlocked through the rotation of some exterior component of the door like the lock or the handle. This naturally leads us to explore a solution geared towards a flexible rotation system that can more easily integrate with existing door locks.

Solution

We propose a portable system that turns the lock on the door (similar to how a person on the inside of the door would manually turn it to let someone in). This non-intrusive unlocking mechanism will be portable and transferable – it can be easily removed from one door and put onto another. The user attempting to access a room would scan their face on an app and make a sound for 5 seconds (picked up by a microphone on the cell phone) to initiate voice authentication. If the face and the voice match a face and voice that has been previously registered on the app, the web app will send a signal back to the microcontroller to initiate the unlocking process. The user will also be able to register other faces and voices (for example, for their roommate) to allow multiple people to use this unlocking system. An important note is that this entire unlocking system will not interfere with manual unlocking with a key.

Visual Aid



High-Level Requirements

- The facial recognition model must achieve at least **90% accuracy**, and the voice recognition system must achieve at least **90% accuracy** in correctly identifying registered users. This accuracy is both on test datasets and actual full-system trials.
- The system needs to be portable. It should be able to be removed from one door and mounted on another within **10 minutes**.
- The total time from when a user submits their face and voice for authentication to when the unlocking mechanism fully turns the lock must be **under 5 seconds**.

Design

Solution Components

Authentication System: Facial and Voice Recognition

The authentication system verifies the user's identity before triggering the unlocking mechanism. It consists of an Android app, a Flask backend hosted on Google Cloud Platform (GCP), and cloud-based processing tools including Google Cloud Speech-to-Text and DeepFace for facial recognition. Users initiate the unlocking process by scanning their faces

and speaking a short phrase into their phone's microphone. These inputs are processed in the cloud, where the system compares them against previously registered data stored in a Faiss instance. If the authentication is successful, a signal is sent to the ESP32-S3 microcontroller, allowing the door to unlock. The system is designed to support multiple users, enabling roommates or family members to register their faces and voices for access.

For the authentication system to work effectively, the facial recognition model must achieve at least 90% accuracy to minimize false positives and false negatives. The Google Cloud Speech-to-Text and Recognition API must achieve at least 90% accuracy in transcribing and recognizing voices. The authentication process must be completed within five seconds from submission to unlock signal transmission. The Android app must securely transmit user credentials and biometric data to prevent unauthorized access. The system must allow multiple users to register by securely storing face and voice data in Faiss. Unauthorized users must not be able to trigger the unlocking process, ensuring security against spoofing attempts.

Physical Unlocking Mechanism

The physical unlocking mechanism is responsible for actuating the lock after the user has been authenticated. This subsystem consists of an ESP32-S3 microcontroller, a stepper motor driver, a stepper motor, and a flexible steel cable that physically turns the lock handle. When the authentication system confirms the user’s identity, the backend sends a signal to the ESP32-S3, which then activates the stepper motor via the driver by sending a PWM STEP signal to the driver. The stepper motor rotates the lock handle using the steel cable, mimicking how a person would manually unlock the door from the inside. The other end of the steel cable is connected to the lock. This mechanism is designed to be non-intrusive, meaning it does not interfere with manual key-based unlocking and can be transferred between different doors with ease.

For this subsystem to function effectively, the stepper motor must provide about 50 N·cm of torque to turn standard lock handles. The ESP32-S3 must receive and process unlock signals within 1 second of receiving authentication approval. The motor driver must regulate the stepper motor current between 0.5A and 2A to prevent overheating while ensuring sufficient force. The flexible steel cable must be securely attached to the stepper motor to ensure reliable operation. It must be heavy enough and be placed far enough from the door handle to generate enough torque to be able to rotate the door handle. The system must not interfere with manual unlocking using a key, ensuring that users can still operate the lock conventionally if needed. The unlocking process must be completed within five seconds of receiving the signal to provide a seamless experience.

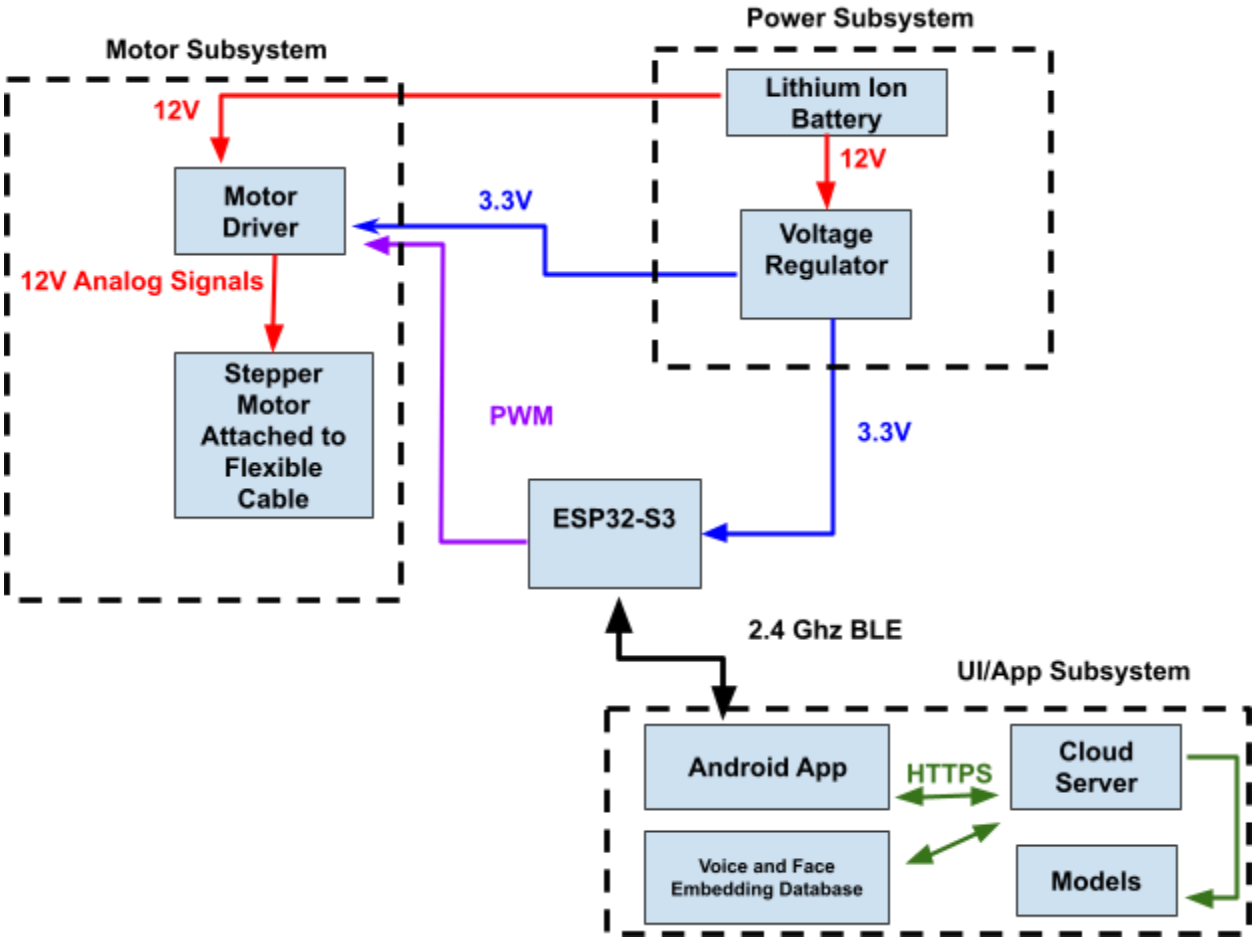
Power System

The power system ensures that all components receive a stable and reliable power supply. The system is powered by a 12V battery, which provides sufficient energy for the stepper motor, ESP32-S3 microcontroller, and other electronic components. Since different components operate at different voltages, a voltage regulator is used to step down the voltage where necessary. The motor driver requires a stable power source to ensure smooth stepper motor

operation without overheating or excessive current draw. This subsystem is designed to maximize battery efficiency while maintaining continuous operation.

For the power system to function properly, it must provide at least 500mA at 3.3V to support the microcontroller's operation. The 12V battery must provide sufficient capacity to support at least 100 unlock cycles per charge. The voltage regulator must provide a stable 3.3V output with a tolerance of ±0.1V to prevent damage to sensitive components. The motor driver must use 3.3 volts for logic and 12 volts to send to the motor, and regulate motor current appropriately. The system must include power-saving features to ensure the stepper motor is only active when unlocking. The power system must prevent over-discharge of the battery, ensuring longevity and safety.

Block Diagram



Tolerance Analysis

Torque Analysis

To ensure the unlocking mechanism reliably rotates the lock handle, we need at least **50 N·cm** of torque. We calculate the torque (T) using:$T = F \times r$ Assuming:

- **Radius (r) of lock handle:** 3 cm
- **Force (F) provided by the cable:** 16.7 N

$T = 16.7\,N \times 3\,cm = 50.1\,N \cdot cm$

This meets our requirement, with a small margin to account for friction and mechanical losses.

Latency Tolerance

The total unlocking process must complete within 5 seconds. Breakdown:

- **Authentication (facial + voice):** ~2.5–3.5 s
- **Bluetooth signal transmission:** ~0.5 s
- **Motor actuation:** ~1 s

Estimated total latency is approximately 4 s, which stays within our 5-second limit, even accounting for potential network delays.

Power Tolerance

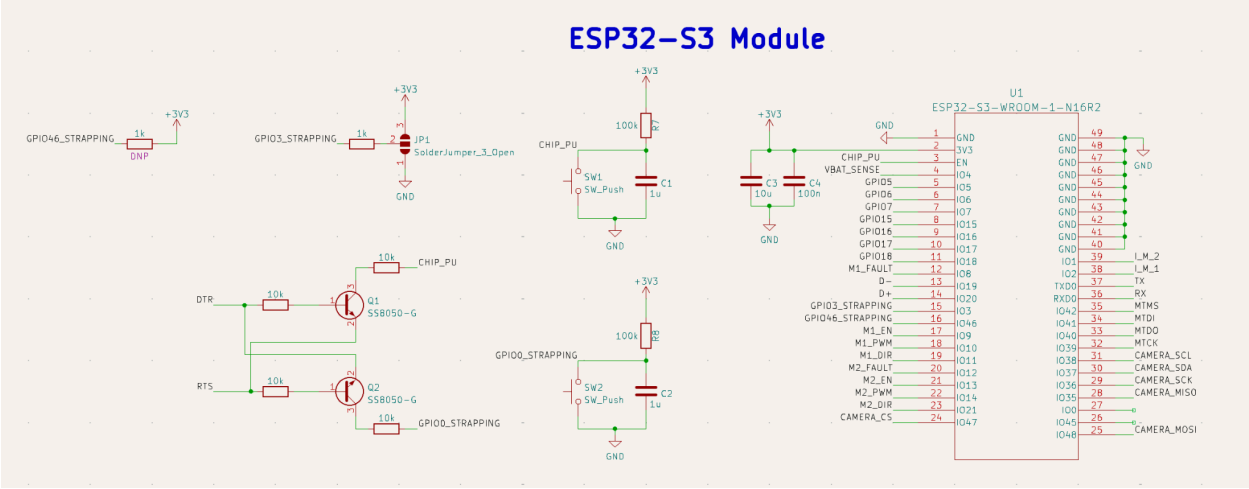
The system uses a 12V battery and must support 100 unlock cycles per charge. With the stepper motor drawing about 1.2A during operation:

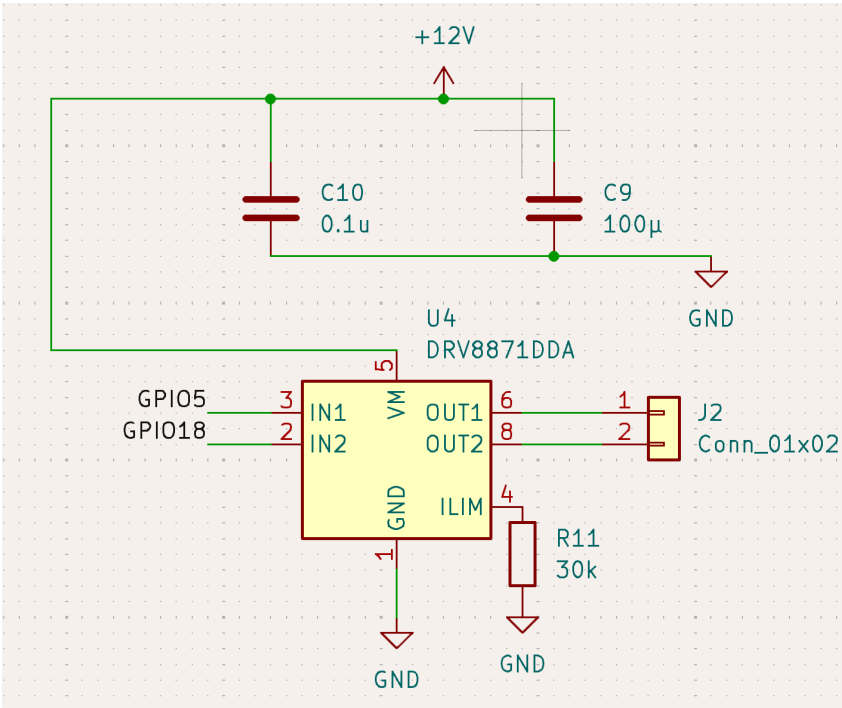
- **Energy per unlock:** $1.2\,A \times 12\,v \times 1s = 14.4\,J$

These calculations guide our design to ensure reliable performance under expected conditions.

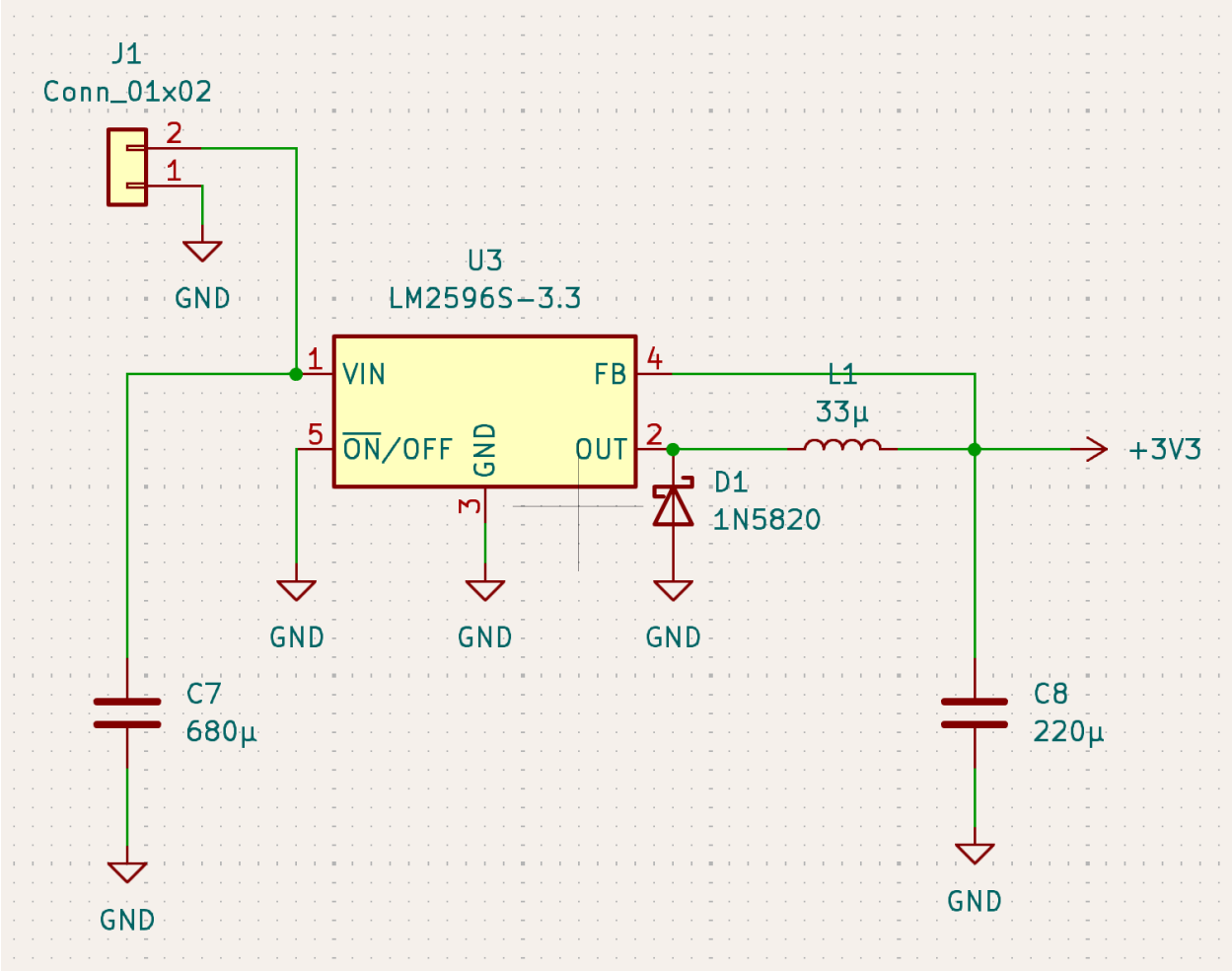
Schematic

ESP-32 Bear Minimum





Motor Driver



Voltage regulator

Requirements and Verification

Subsystem	Requirement	Verification
Authentication System: Facial and Voice Recognition	The facial recognition model must achieve at least 90% accuracy in recognizing registered users.	1. Train the facial recognition model on a dataset with at least 1000 images of different users. 2. Conduct a test using 200 images of registered users and 200 images of unregistered users. 3. Measure the True Positive Rate (TPR) and False Positive Rate (FPR) to verify 90%+ accuracy.
	The voice recognition system must achieve at least 90% accuracy in identifying registered users.	1. Train the voice recognition model using at least 100 different voices. 2. Conduct a test where each registered user provides five voice samples, and each unregistered user provides five samples. 3. Measure the accuracy based on successful and failed identifications.

Subsystem	Requirement	Verification
	The system must process authentication requests within 5 seconds.	1. Conduct 20 authentication trials. 2. Record the time from when the user submits their face and voice input to when the authentication decision is made. 3. Verify that the time taken is within the 5-second limit.
Physical Unlocking Mechanism	The stepper motor must generate at least 50 N·cm of torque to rotate standard lock handles.	1. Attach the stepper motor to a torque measurement device. 2. Apply a resistance equivalent to a standard door lock mechanism. 3. Measure and confirm that the torque output is at least 50 N·cm.

Subsystem	Requirement	Verification
	The unlocking mechanism must complete the unlocking process within 5 seconds of authentication.	1. Perform 20 test trials where the authentication system triggers the unlocking process. 2. Measure the time taken from the authentication success signal to the full rotation of the lock. 3. Verify that the unlocking process is completed within 5 seconds.
	The system must be portable and mountable on different doors within 10 minutes.	1. Conduct five trials where the system is mounted on different door types. 2. Measure the time taken to complete each installation and removal. 3. Verify that the installation time does not exceed 10 minutes.

Subsystem	Requirement	Verification
Power System	The voltage regulator must provide a stable 3.3V output with a tolerance of ±0.1V.	1. Connect the voltage regulator to an oscilloscope. 2. Apply a varying load from 100mA to 500mA. 3. Measure and verify that the output voltage remains within 3.2V to 3.4V.
	The 12V battery must provide sufficient power for at least 100 unlock cycles per charge.	1. Fully charge the battery. 2. Conduct 100 unlocking cycles while monitoring battery voltage and current draw. 3. Verify that the system remains operational for all 100 cycles before the battery is depleted.
	The stepper motor driver must regulate current between 0.5A and 2A to prevent overheating.	1. Connect a current probe to the motor driver. 2. Run the stepper motor continuously for 30 minutes. 3. Verify that the current remains within the 0.5A to 2A range and that the driver does not overheat.

Subsystem	Requirement	Verification
Communication System	The ESP32-S3 must receive and process authentication signals within 1 second.	1. Use a logic analyzer to capture the Bluetooth signal transmission time. 2. Conduct 20 trials measuring the time taken from authentication success to the ESP32-S3 receiving the signal. 3. Verify that the processing time does not exceed 1 second.
	The Android app must securely transmit authentication data using encryption.	1. Inspect the network traffic using Wireshark. 2. Verify that all authentication data is transmitted using TLS encryption. 3. Ensure that no plaintext data is visible in network captures.
Safety and Security	The system must prevent unauthorized access through biometric spoofing.	1. Conduct 50 spoofing attempts using printed facial images and recorded voice samples. 2. Measure the False Acceptance Rate (FAR). 3. Verify that unauthorized attempts remain below 5% acceptance rate.

Subsystem	Requirement	Verification
	The system must allow users to delete stored biometric data upon request.	1. Implement a ‘Delete Data’ option in the app. 2. Conduct 10 trials where a user deletes their biometric data. 3. Verify that the database no longer contains the deleted user’s face and voice embeddings.

Cost

Parts

Description	Manufacturer	Part #	Quantity	Cost	Link
DC Motor	Pololu	Pololu 12V 150:1 Gearmotor	1	$23.45	Link
Schottky Diode	STMicroelectronics	1N5824	1	$1.00	Link
Step down voltage regulator	Texas Instruments	LM2596S-3.3	1	$8.76	Link
Motor Driver	Texas Instruments	DRV8871DDA	1	$3.21	Link
10k Resistor	Koa Speer Electronics	CFP1/4CT52 R103J	2	$0.22	Link
100k Resistor	Koa Speer Electronics	CF1/4C104J	3	$0.39	Link

1 uF Capacitor	Kemet	C322C105K5 R5TA	2	$2.04	Link
10 uF Capacitor	Kemet	C322C106K3 R5TA	2	$1.66	Link
0.1 uF Capacitor	Kemet	C322C104K5 R5TA	2	$0.78	Link

Each group member might make around $25 an hour. For the number of hours the project will take, which we estimate to be around 60, the cost of the project is $25/hour * 2.5 hours * 60 = $3750. Multiplying this by 3 for each of the 3 partners yields $11250 in labor costs.

The parts total is 23.45 + 1 + 8.76 + 3.21 + 0.22 + 0.39 + 2.04 + 1.66 + 0.78 = 41.51

The total cost is 11250 + 41.51 = $11291.51

Schedule

Week	Task	Member
March 10 – March 17	Finalize PCB design and submit design Create Flask server to deploy image and voice models for voice and face recogition Write code for ESP32 to receive bluetooth signal Work on Android app, make UI and connect app to server Connect all subsystems and create mini-demonstration with motor for breadboard demo, get ready for breadboard demo and Design Review	Yuhao and Arnav Raghav Arnav Yuhao All
March 17 – March 24	Make sure that chosen models are robust and have high accuracy	Raghav

	Connect app to vector database to store voice and image embeddings to store user data Improve UI for app	Yuhao and Arnav Raghav
March 24 – March 31	Soldering PCB and test to make sure the PCB works as intended Make sure that the app works with the PCB (app can connect esp32-3 and esp32 can drive motor to open door)	All All
March 31 – April 7	Create mounting mechanism for PCB onto door, make sure that our contraption can successfully turn lock on door. Make adjustments to the PCB (change motor, etc.., to make sure that lock will change)	All
April 7 – April 14	Prepare for final demo, Continue to integrate subsystems together and testing the project	All
April 14 – April 21	Prepare for final demo, Continue to integrate subsystems together and testing the project	All
April 21 – April 28	Final Demo, and prepare for final presentation	All
April 28 – May 5	Final Presentation, write final report	All

Ethics and Safety

One of the most critical ethical issues in this project is privacy and data security. Our system collects and stores facial recognition and voice authentication data – both are sensitive biometric identifiers. According to IEEE Code of Ethics Principle 1, we must “protect the privacy of individuals” and ensure that collected data is stored securely.

Unauthorized users getting access to the biometric data or being able to use biometric data to get into rooms would be a large breach of privacy. To mitigate this, our app uses secure authentication protocols, such as OAuth 2.0, to prevent unauthorized access. Furthermore, we will allow users to delete their stored biometric data upon request, ensuring compliance with privacy laws such as the Biometric Information Privacy Act**.**

Citations

IEEE - IEEE Code of Ethics. Accessed February 13, 2025. https://www.ieee.org/about/corporate/governance/p7-8.html