Voice Biometrics Lock

ECE 445 Design Document - Spring 2021

Team 34 - Bella Chen(aotingc2) Lixin Guo(lixing2) Zaki Khan(zakik2)

TA: Anand Sunderrajan

Table of Contents

1. Introduction

1.1 Objective

Losing keys is not a pleasant experience. In addition, key locks are not good at protecting homes because all of them can be picked, unless you have a keyless lock [1]. Therefore, electronic locks like the Nest x Yale [2] are becoming more prevalent for modern homes. They even work with voice recognition like Google Home, and users can lock the door from the inside with one command. Yet when entering from the outside, a passcode is still needed because home devices would not recognize who should not be allowed into the door, and passcodes can be compromised. Our project will solve this security issue and improve home safety by using biometric traits, such as voice and fingerprints, that are unique and are usable as a more secure and convenient means of owner security.

In order to solve this issue, we propose to design a biometrics voice lock. This lock would work with voice recognition and a fingerprint sensor along with an application that users could download on their phones. This lock would use real-time voice recognition to distinguish the voice of the authorized user(s). It would do so by generating a random command for the user to read aloud. The app would verify that the command said was correct and also that the voice was that of any authorized user. This would prevent an intruder from getting a recording of the user and attempting to pose as the authorized user. In addition, in the case that the user is unable to use his/her voice, we would also implement a fingerprint scanner on the lock to prevent him/her from being locked out. In other words, the fingerprint scanner would help prevent false negatives and the randomized command would help prevent false positives. The mobile application would be connected to a receiver circuit through Bluetooth. Overall, our solution would target the everyday individual that is working to improve the security of his/her home.

1.2 Background

According to the US Department of Justice, there are 2.5 million burglaries annually in the United States, with 66% of these being home invasions [3]. In addition, 34% of burglars use the front door when breaking into a home, according to the Bureau of Justice [3]. Our solution would add an additional level of security to the user's front door. We would remove the keyhole to prevent the burglar from picking the lock. This would reduce the method of entry for any intruder. Reducing a method of entry would discourage some home burglars who look for easy and not obvious methods of entrance and would in result lower the number of burglaries that occur.

Recently, there has been an increase in mobile keys. For example, Hyundai and Mercedes have both implemented applications that enable owners to unlock their cars from their phones [4]. In addition, some hotels such as Marriott now allow guests to unlock their rooms with their phones using a mobile room key [5]. However, there are not any commercially available door locks with voice recognition and fingerprint scanners. As time is passing, users are beginning to prefer

mobile solutions to everyday problems, home security included among them. Our solution would satisfy this by allowing them to control their front door with their phone and would reduce the need for people to carry bulky house keys.

During the COVID pandemic, many businesses are considering incorporating voice-based features into their product, as there is an increase in demand for contactless solutions to contain the spread of the virus. By primarily relying on speech recognition and using a fingerprint sensor only as a backup, we create a door lock that minimizes physical contact with its surface.

1.3 Physical Design

Our solution would require the user to install the physical lock on his/her door and download a smartphone app that communicates with the Bluetooth module inside the lock. An example of the physical lock is shown in Figure 1 and Figure 2. The door status will be shown at the top-center of the screen, and the below smartphone screen shows a locked status. A gray button will prompt the user to tap and speak a one-time, random passcode. If the app's voice recognition system deems that it is the rightful owner, a wireless signal will be sent from the smartphone to trigger the lock. The fingerprint sensor would be activated after three failed attempts of voice recognition. Mockups of the smartphone user interface are shown in Figure 3 through Figure 6.

In Figure 3, we have two potential views of the application home screen. These views correspond to two potential lock statuses: unlocked and locked. If the lock is locked, a "locked lock" icon would appear on the center of the screen along with the "Locked" status message written in red on the bottom of the screen. IF the lock is unlocked, a "unlocked lock" icon would appear in the middle of the screen with a "Unlocked" message on the bottom of the screen. In both cases, there would be a navigation bar menu which would allow users to toggle between different parts of the application.

In Figure 4, we have the view of the application setup screen This screen would allow users to setup their lock by registering their voices. It would greet them with a welcome message on the top screen.The instructions for registering the voice would be in the center of the screen There would be a microphone button on the screen which would allow the user to speak a phrase to register themselves with the system. There would also be a menu on the bottom of the screen allowing users to move between different parts of the application.

In Figure 5, the view of the lock screen is shown. THis would allow users to lock the lock with the push of a button. There would be a button in the center of the screen which when pressed would send a signal to the lock to lock itself. The navigation menu bar would also be present.

In Figure 6, we have the view of the unlock screen. This allows the user to unlock his door on a successful verification of his voice. There would be a button on the center of the screen. When the button is pressed a random alpha-numeric one time password would be generated and

displayed on the bottom of the screen. The user would be able to speak that phrase to authenticate himself. The application navigation bar menu would also be shown.

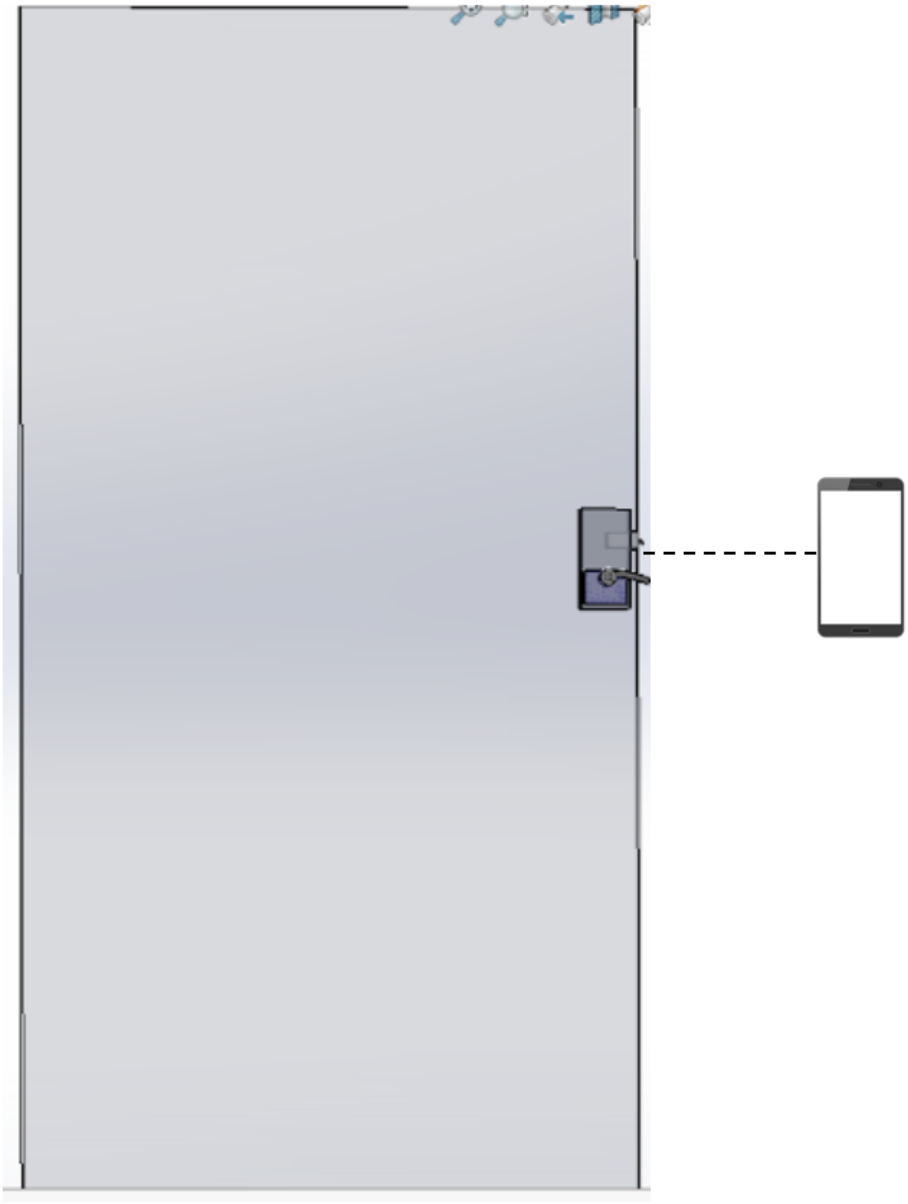

Figure 1. Electronic Lock Installed on a Door

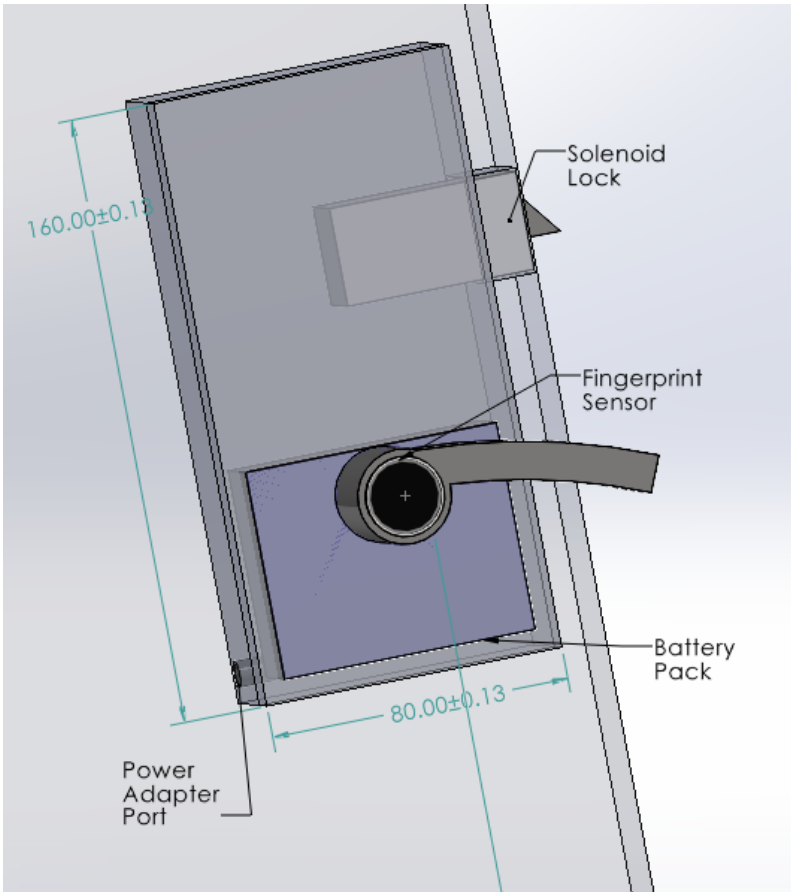

Figure 2. Annotated, Zoomed-in View of the Electronic Lock

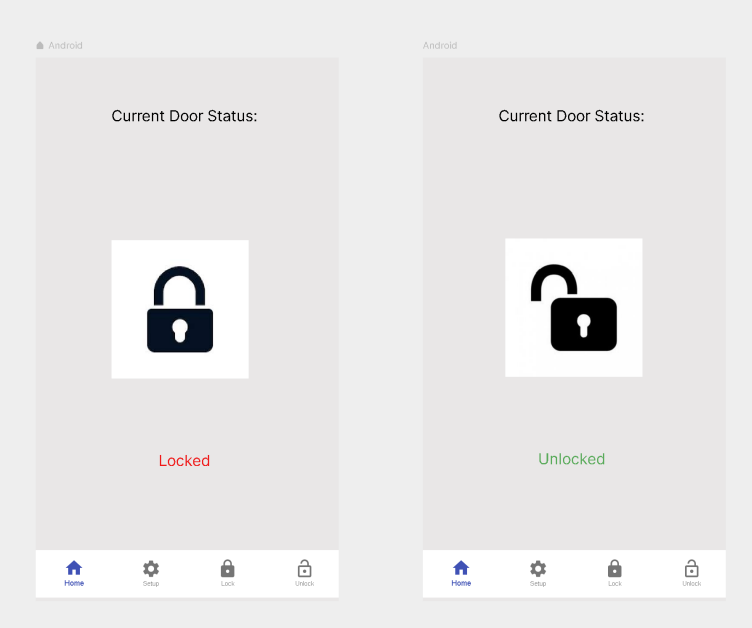

Figure 3. Application Home Screen Mockup

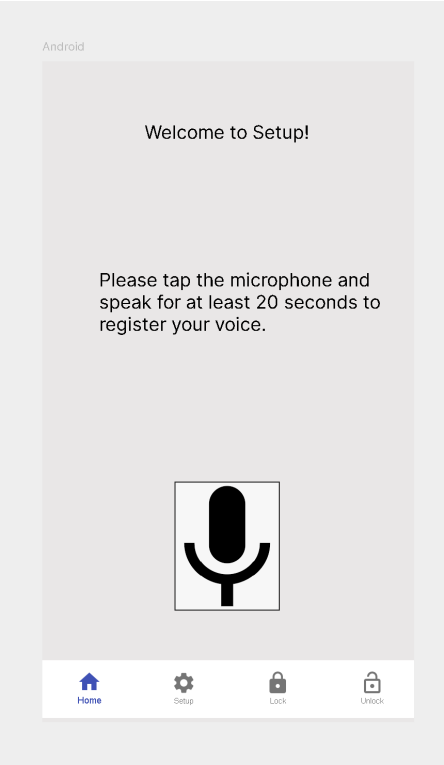

Figure 4. Application Setup Screen Mockup

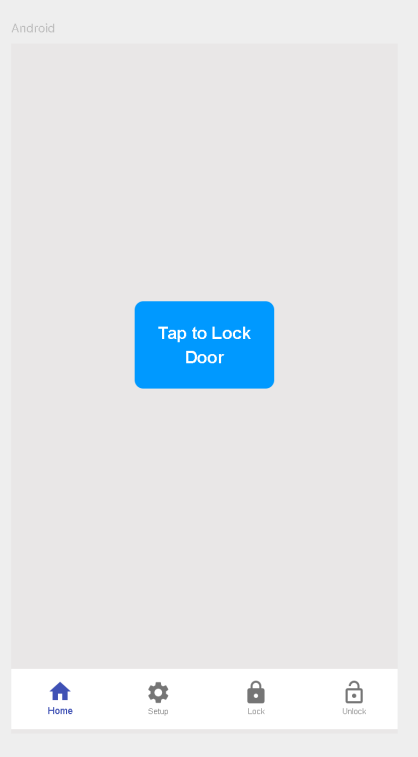

Figure 5. Lock Screen Mockup

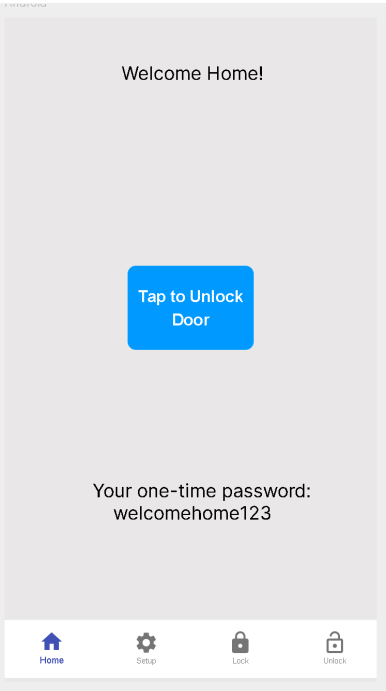

Figure 6. Unlock Screen Mockup

1.4 High-level Requirements

1. The smartphone app should be able to verify a spoken string of at least six characters long with a mix of letters and numbers [6] with a minimum of 90±5% percent accuracy, compared to 95%, the highest rate that is achieved by Google in 2017 [7].
2. The processing time of speech recognition should be real-time, taking no longer than 3 seconds for a 5-second audio clip, for example [8].
3. Verification with the fingerprint sensor should serve as a backup method and have an extremely low error rate of lower than 1% [9].

2. Design/Block Diagram

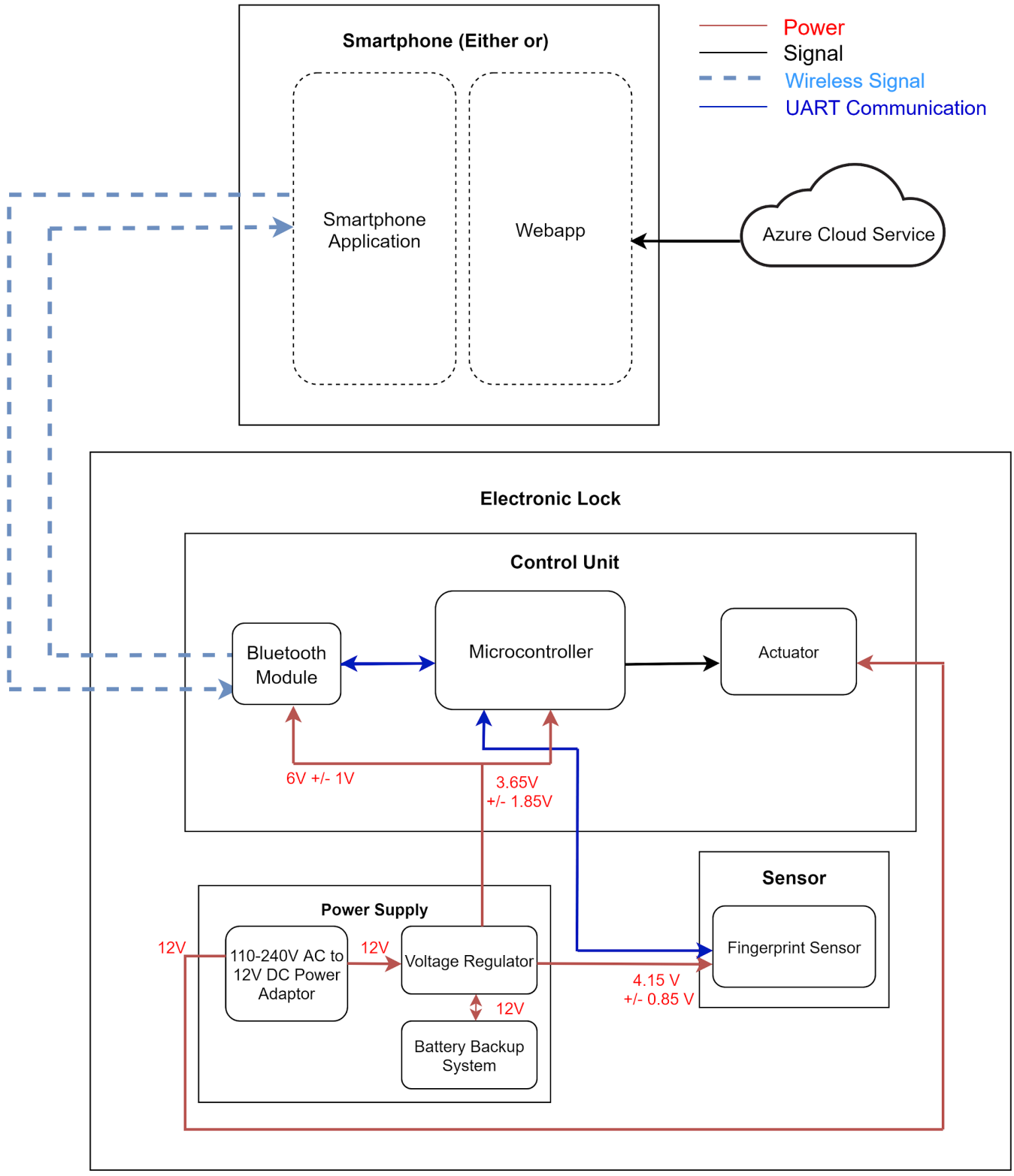

Figure 7. Design Block Diagram

Figure 7 shows the block diagram that will meet our high-level requirements. We will use Azure cloud service to carry out the voice-recognition feature and store any voice or account data to achieve a high success rate and to minimize computational delay. The smartphone application is a user interface between the cloud and the control unit. The lock itself will require a control unit to lock or unlock the door by a series of connections that can receive external signals that trigger the actuator. The microcontroller is the central processor of all information and will be able to communicate back with the phone to report back the lock status. The sensor data is selected based on a high success rate. The microcontroller only processes the sensor's input when fingerprint verification is activated by the app. The power supply unit must be able to supply at least 12 V. A voltage regulator will provide the required voltage to all subcomponents inside the lock. The battery backup system will deliver power to the voltage regulator during a power outage. Detailed requirements will be explained in the functional overview section.

2.1 Smartphone

The smartphone would serve as one way for the user to interact with the lock. Our project would use three main components on the phone: an application that we would develop, a microphone, and a Bluetooth receiver/transmitter. The user would use the microphone to communicate with the application and say the command needed to unlock the lock. The user's voice data collected by the microphone would help the application learn the voice of the user. Whenever a successful unlock is made, the application would use the Bluetooth sensor to send the lock a signal to unlock. This signal would be sent to the Bluetooth module on the control unit of the electronic lock module. In addition, it also receives a signal whenever the fingerprint scanner was used to unlock the lock. This enables lock status updating on the application. In addition, the application would let the user lock the lock by tapping a button. Overall, this block would add the features of allowing the user to unlock the lock using a passphrase, allowing the user to lock the door with one tap in the app, and having the correct status of the lock displayed.

Requirement	Verification
1. The application should be able to handle the user speaking for 20 seconds which is necessary for user registration.[10] 2. The user should be able to unlock the door within 200 ms from successful verification. [11] 3. The smartphone app should be able to verify a spoken string of at least six characters long with a mix of letters and numbers [6] with a minimum of 90±5% percent accuracy,	1. Open the application, go to setup screen, press button to speak and use another timer to time for 20 seconds 2. Perform an unlock and time the latency of unlock and make sure under 200 ms. 3. Try performing an unlock using a password with length of six with a mix of letters and numbers and check how many times out of 100 it can successfully unlock.

2.2 Control Unit

The control unit works as the central processor unit for the electronic lock, it controls the actions on the lock based on verification result signals. There are four parts connected here including the Bluetooth module for wireless signals, the microcontroller for data operation, and the motor with a relay to physically lock or unlock the door. This unit will be implemented with the lock and communicate with the smartphone and fingerprint sensor to simultaneously update lock status and take actions by instructions.

2.2.1 Bluetooth Module

The Bluetooth receiver unit uses a Bluetooth audio receiver gadget to build a wireless connection with the smartphone to transmit the signals forward or backward accordingly. We plan to use Adafruit Bluefruit LE UART Friend [12] for Bluetooth Low Energy and this serves as the interconnection between the phone and the microcontroller. This module operates at the standard Bluetooth frequency of 2.4 GHz. Once the phone finishes operation on voice data, it will send the signal to the receiver which is then passed to the microcontroller to determine the instruction. The information of the lock identified in the microcontroller is also sent back to the phone through this block for the user to check.

Requirement	Verification
1. The BLE module must communicate reliably with a phone for a distance of 5 ± 1m in open space [12].	1. Place the BLE module 4 m away and send or receive a message from a phone. After 5 successful attempts, move 1 m further and repeat to get the maximum working distance.
2. The module's voltage regulator must step down 5V±0.5 V to 1.8 - 3.3 V for up to 15.2 mA in data mode [12].	2. Supply a voltage of 5V±0.5V at the Vin pin and monitor with an oscilloscope. Set the mode selection to UART. Probe the 3Vo pin using the multimeter to measure the voltage and the current outputs of the regulator.
3. The BLE module must be able to transfer data from the microcontroller to the Android phone in 400 ± 50mS [13].	3. In the Adafruit Bluefruit LE Connect app, send a "Hello, world!" message to the Serial Monitor, and enter the message in the Serial Monitor to receive it from the phone. Measure the time elapsed.

2.2.2 Microcontroller

The microcontroller processes all signals to control the actuator. It communicates with the phone to read the Bluetooth data and send commands to turn the actuator on and off. The condition of the actuator will be simultaneously sent back to the Bluetooth module and then delivered to the phone displayed for users. If the voice recognition fails by accident, the processor will receive data from the fingerprint system and operate for the commend passed to the actuator that if the data matches, the actuator will be triggered and this status update will be sent back to the smartphone at the same time.

We plan to use the ATmega328, since it is compatible with UART communication with both the sensor and the Bluetooth module and all other subcomponents inside the control unit. On an Arduino Uno, serial communication occurs with 1 start bit, 8 data bits, and 1 stop bit. A 16MHz crystal oscillator provides the clock input. We have considered the ESP microcontrollers because of the built-in WiFi, but the selected fingerprint sensor has never been successfully tested on it and we may encounter difficulties during the implementation.

Requirement	Verification
1. The microcontroller can communicate with the smartphone app through the BLE module at 9600 bps [14].	1. Wire up the BLE module with the microcontroller. Download the Adafruit Bluefruit LE Connect app on the phone and verify that it can send and receive messages through UART.
2. It can verify the input fingerprint data with the images stored in the sensor's SD card at 115.2k bps [9].	2. Wire up the fingerprint sensor with the microcontroller and load the program. Use the Serial Monitor on a PC to test each function, including add, verify, and delete fingerprints.
3. It can activate the solenoid lock whenever a verification is complete and deactivate it when a "close door" command is sent from the app.	3. Wire up the solenoid lock with the microcontroller. Use the Serial Monitor to activate or deactivate the lock. Verify the reported lock status.

2.2.3 Actuator

We plan to use a solenoid lock as an actuator for the door lock. The solenoid lock receives a control signal from the microcontroller to pull or release the latch, realizing the locking mechanism. A power transistor with a protection diode is required to drive the solenoid. When it is inactive, the latch is sticking out so that the door is locked. When a pulse activates the lock, the solenoid draws power from the 12 V supply and pulls in the latch to unlock the door.

Requirement	Verification
1. The lock needs to operate when a 9-12 V DC supply is provided, with a current draw of 500 - 650 mA [15].	1. Power the solenoid lock with an adjustable power supply to provide 9-12 V in 1V increments and observe whether the lock operates. Verify the current with an ammeter.
2. The time it takes to activate must be within 1-10 s [15].	2. Connect the solenoid lock with an Arduino Uno. Send a pulse through the digital output pin to activate the lock. Measure the time between when the pulse is sent and when the lock is activated.

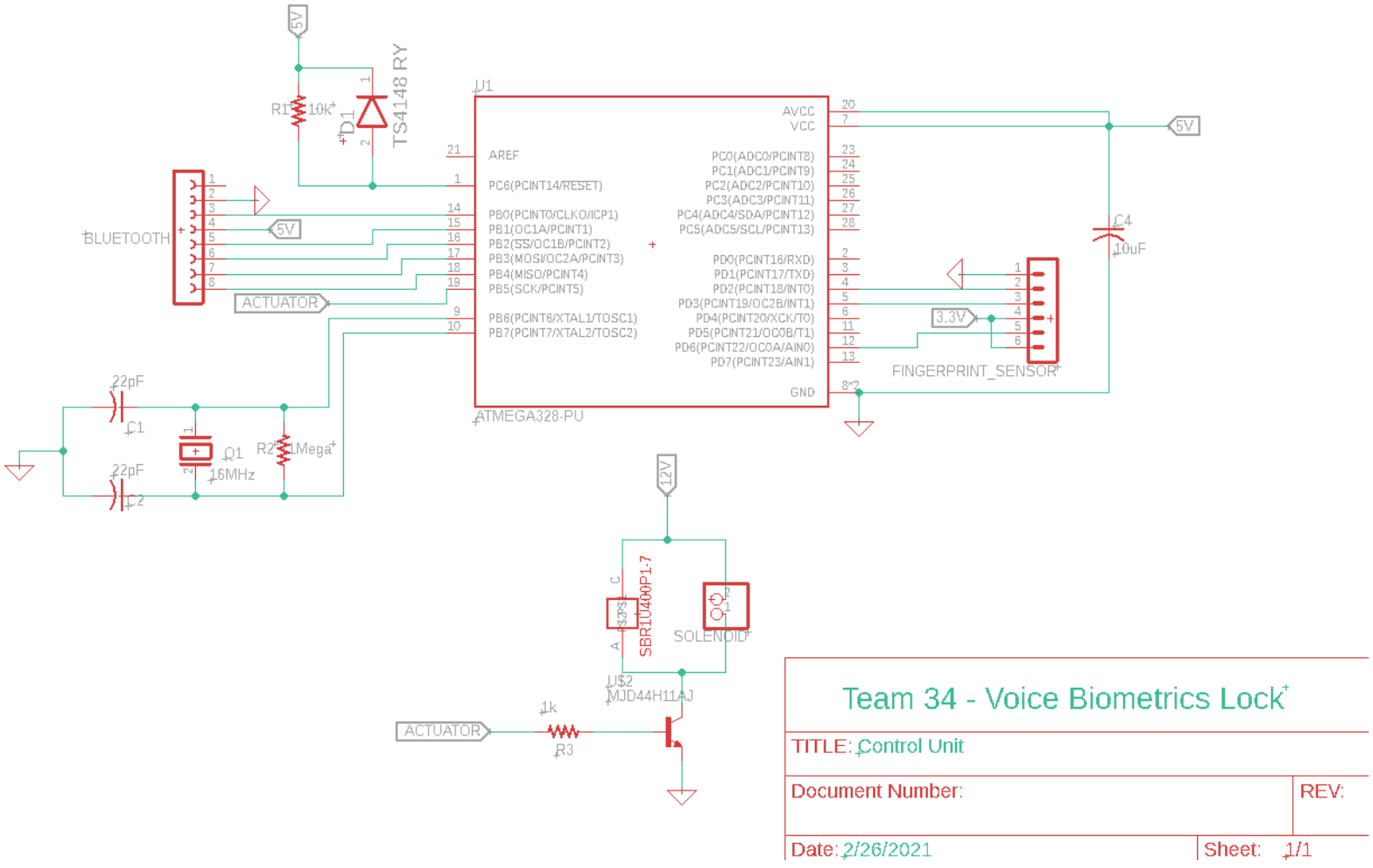

Figure 3. Schematic of the Control Unit

2.3 Power Supply

All sub-components inside the electronic lock need a power supply for operation. Due to different needs in the voltage source, we would need a central power supply that can provide sufficient power to all subsystems and a voltage regulator that allocates the correct amount of voltage according to the specifications of each unit.

2.3.1 12 V Supply

We will use a wall adaptor as the power supply. A DC power supply needs to provide a constant 12 V because the required voltage for the solenoid lock is 10.5±1.5 V. Supplying it at the maximum voltage allows less current to be drawn from the solenoid lock, and more current to drive the rest of the components. It is ideal to supply the solenoid lock at 12 V because it draws a large current of 500 mA and thus might not be powered by voltages near 9 V[15].

Requirement	Verification
1. The DC power supply preferably to provide a constant 12 V ± 0.5 V. 2. The current supported by the DC supply should be in range 800 - 2000 mA.	1. Connect the DC power adaptor to the oscilloscope and record the output voltage to check it is in the range of 12.5 V to 11.5 V for at least 5min length. 2. Connect the DC power adaptor to the oscilloscope and observe the current is in the range 800 mA to 2000 mA.

2.3.2 Voltage Regulator

Because the lock subcomponents require different voltages that are lower than the microcontroller's minimum voltage requirement, we need a voltage regulator to deliver different amounts of voltages to the subcomponents without losing too much power.

Requirement	Verification
1. The voltage regulator must have a minimum input voltage of 7 V and must handle at least 350 mA peak current draw.	1. Verify voltage and current: - Use a separate test circuit with load and variable resistor as shown in Figure 4.

2. It must supply 6±1 V to the bluetooth module[12], 4.15± 0.85V to the fingerprint sensor[9], and 3.65±1.85 V to the microcontroller[14].	- Connect the regulator and adjust the load to deliver 350 mA current measured by a multimeter. - Test the voltage between terminals within the maximum output range. 2. Turn on the DC power supply and then connect the regulator to the oscilloscope and record the output voltage. Repeat this process for three times separately for each subsystem by adjusting the regulator to the desired output, check the output voltage in range 5 V to 7 V, 3.3 V to 5 V, 1.8 V to 5.5 V separately.

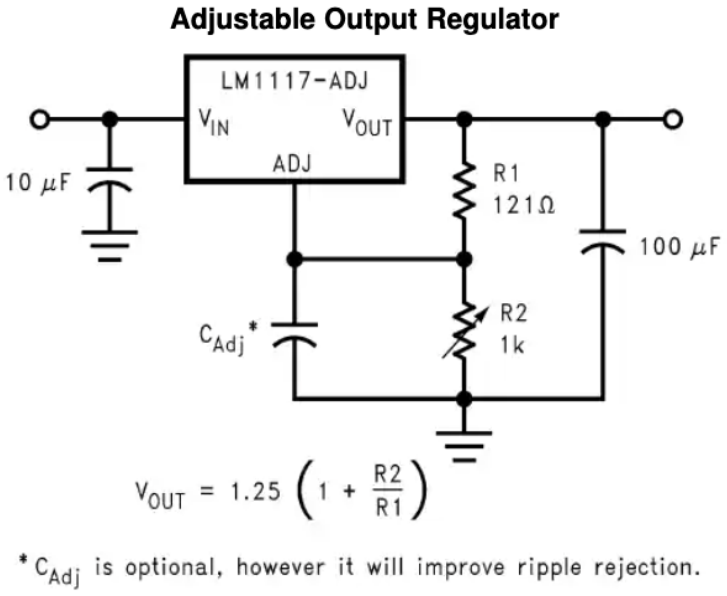

Figure 4. Schematic for LM1117 Low-Dropout Linear Regulator [16]

2.3.3 Battery Backup System

The battery backup system is connected to the voltage regulator and DC power supply to protect the board from operating out of the safety region and provides a backup power source for

the whole system with little delay in case the main supply fails. The battery must be rechargeable and have protection against overcharging. The adaptor will deliver power to the circuit and recharge the battery while in operation. During a power outage, the circuit will draw power from the battery.

Requirement	Verification
1. The backup system needs to operate normally when the mains are working and backup the circuit whenever the DC power stops to provide the regulator required voltage supply in range 12 V ± 0.5 V.	1. Connect the circuit with a power supply and charge it for a while using constant 12V voltage then turn off the supply, check the voltage output using a multimeter to be in range 12 V ± 0.5 V.

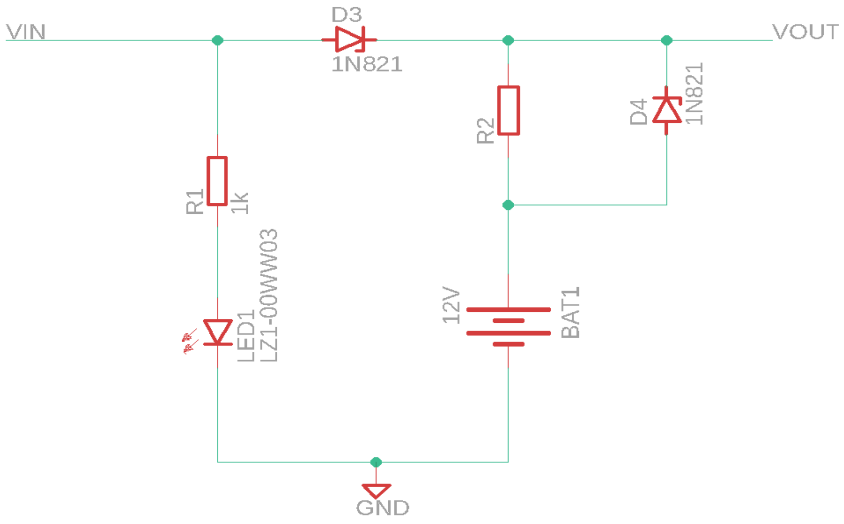

Figure 5. Schematic for Battery Backup Circuit

2.4 Sensor

The sensor module would contain the fingerprint sensor. The fingerprint sensor would be used to prevent false negatives. The power supply module would be used to provide power to the fingerprint sensor and to regulate the voltage that it receives. It would communicate with the microcontroller in the control unit through UART. When a person places his fingerprint on the sensor, the fingerprint sensor would communicate with the microcontroller and if the verification of print is successful, will send a signal to the application to unlock the lock. This module would add the feature of allowing the user to unlock the lock using a fingerprint scanner to our overall project.

Requirement	Verification
1. The sensor should identify living things only, which eliminates artificial replicas of the user's fingerprint. 2. It should have an extremely low false acceptance rate of less than 0.005% and a false rejection rate of less than 1% [9].	1. Connect the sensor to be powered by the voltage regulator and the microcontroller, then try the sensor through real fingers and printed pictures while reading the collected information stored on the monitor to ensure only proper test data is accepted. 2. Wire up the sensor with the control unit and test the sensor through a large amount of attempts to record the number of false recognition and adjust until the false rate is within the acceptable range.

2.5 Tolerance Analysis

Fast and reliable communications amongst the sub-components in our hardware system are critical to the project's success. UART is one of the most common protocols for asynchronous serial communication between two devices, and it is the one we will use for the microcontroller to talk to the BLE module and the processor on the fingerprint sensor. Baud rate is a merit of how fast data is transported in serial communication, defined as "the number of distinct symbol changes (signaling events) made to the transmission medium per second in a digitally modulated signal" [17]. In digital communication, it is defined as "the transfer rate in bit per second (bps)" [12]. To ensure that the ATmega328 MCU can produce the desired baud rate and to avoid the potential issues in hardware communication, we need to ensure that the maximum baud-rate mismatch is $\pm$ 2.5% on each side so that the overall tolerance does not exceed 5%[18].

As stated in the ATmega328P datasheet, the error rate is given by:

$$Error[\%] = \left(\frac{BaudRate_{Cloest\ Match}}{BaudRate} - 1\right) \times 100\%[14].$$

During UART initialization, the desired baud rate is set by setting the value of the baud rate register (UBRRn). Asynchronous normal mode or double speed mode is available by setting the U2Xn bit. The data transmission rate is twice as fast in double speed mode by reducing the clock divider, but we would need a more accurate baud rate setting and system clock [14]. Equations

for calculating the baud rate and the UBRRn register value are listed below. The UBRRn value will be rounded down to the nearest integer.

Operation Mode	Baud Rate	UBRRn
Normal mode (U2Xn = 0)	$Baud\ Rate = \frac{f_{OSC}}{16(UBRRn+1)}$	$UBRRn = \frac{f_{OSC}}{16\ Baud\ Rate} - 1$
Double speed (U2Xn = 1)	$Baud\ Rate = \frac{f_{OSC}}{8(UBRRn+1)}$	$UBRRn = \frac{f_{OSC}}{8\ Baud\ Rate} - 1$

Table 1. Equations for ATmega328 baud rate calculations [14]

The choice of baud rate is also dependent on the mismatch between the transmission and the receiving baud rates. If the transmitter is sending the data at a rate that is either too fast or too slow, the receiver unit would not be able to synchronize to the start bit. Therefore, we need to ensure that the BLE module and the fingerprint sensor will both operate at the select baud rate.

	f_{osc} = 16.0000MHz			
	U2Xn = 0		U2Xn = 1	
Baud Rate (bps)	UBRRn	Error	UBRRn	Error
2400	416	–0.1%	832	0.0%
4800	207	0.2%	416	–0.1%
9600	103	0.2%	207	0.2%
14.4k	68	0.6%	138	–0.1%
19.2k	51	0.2%	103	0.2%
28.8k	34	–0.8%	68	0.6%
38.4k	25	0.2%	51	0.2%
57.6k	16	2.1%	34	–0.8%
76.8k	12	0.2%	25	0.2%
115.2k	8	–3.5%	16	2.1%
230.4k	3	8.5%	8	–3.5%
250k	3	0.0%	7	0.0%
0.5M	1	0.0%	3	0.0%
1M	0	0.0%	1	0.0%
Max.(1)	1Mbps		2Mbps	

Table 2. List of the baud rates and error rates based on a 16 MHz system clock [14]

The fingerprint sensor has a default baud rate of 115.2k bps [9]. The Adafruit Bluefruit LE Friend has a maximum recommended baud rate at 9600 bps, but a 115.2k bps baud rate is acceptable [12]. From Table 2, generating a baud rate of 115.2k bps would yield a -3.5% error in normal mode and a 2.1% error in double speed mode. The error for 9600 bps is 0.2 % in both

cases. For best performance, we can use either the hardware or software serial to control the BLE module and software serial to control the fingerprint sensor, as specified by its wiki page [9].

3. Costs

The hourly salary of each partner is $36/hr, obtained by an estimation based on the statistics of the 50th percentile engineering annual salary provided by the most recent UIUC Grad Success Report [19]. We expect each partner to work 10 hours per week. The labor cost of each partner will be:

$$36\ \$/\text{hr} \times 2.5 \times 10\text{hrs/week} \times 9\text{weeks} = \$8,100 \text{ per partner}$$

The total labor cost will be:

$$\$8,100/\text{partner} \times 3 = \$24,300$$

Component Costs:

Description	**Manufacturer**	**Part #**	**Quantity**	**Cost**
ATmega328P Microcontroller	Atmel (Sparkfun)	DEV-10524	1	$5.95
Adafruit Bluefruit LE UART Friend	Adafruit Industries (Amazon)	ADA2479	1	$17.87
Capacitive Fingerprint Sensor / Scanner	DFRobot	SEN0348	1	$16.50
Lock-style Solenoid - 12VDC	Adafruit Industries (Amazon)	ADA1512	1	$22.75
100-240V AC to 12V DC Power Adapter	VeeDoo (Amazon)		1	$8.99
Voltage Regulator	Texas Instruments (Digi-Key)	LM1117	1	$1.17
12V 4000mAh Lithium Ion Rechargeable Battery Pack	Folk Battery		1	$12.50
Assorted RLC, diodes, crystal, pin headers and connectors	Digi-Key		Assorted	$10
Total Cost:				$95.73

Grand total: $\$24,300 + \$95.73 = \$24,395.73$

4. Schedule

Week	Zaki	Bella	Lixin
3/1/21	Research on the smartphone block of the Design Document.	Work on the HL&RV table of the power supply block.	Work on the HL&RV table of the MCU sub-block and the cost section.
3/8/21	Start on initial development of the application. Do research on voice recognition frameworks.	Design circuit for power supply and check on bluetooth module.	Order components. Finalize the schematic and work on PCB.
3/15/21	Work on UI of application.	Test and set up the fingerprint sensor subsystem.	Test functionality of each control unit and make adjustments to the PCB design if needed.
3/22/21	Finish UI of application. Start work on the voice recognition part of the application.	Solder and check the PCB and verify subsystems.	Solder parts onto the PCB and verify board functionalities at test points.
3/29/21	Continue work on the voice recognition part.	Bench test the power and sensor systems.	Bench test the control unit.
4/5/21	Finish work on the voice recognition . Start working on integration with hardware	Bench test the voice recognition system after integration and debug the hardware.	Integrate the app with the hardware and perform bench tests.
4/12/21	Finish hardware integration.	Final edge tests on the hardware subsystems.	Finalize current design and prepare for Mock Demo.
4/19/21	Complete end-to-end testing of the application with the hardware.	Refine the systems after Mock Demo and record successful demo videos for backup.	Make adjustments based on Mock Demo and test hardware & software integration.
4/26/21	Edit final report and prepare for the final presentation	Edit final report and prepare for final presentation.	Prepare for the presentation and start on the final report.

5. Ethics and Safety

During the development of our project, we will follow the instruction from ACM Code 2.9 to make design and implement “robustly and usably secure”[20]. We as a team will use our own information during the testing process of the voice and fingerprint systems. We will ensure agreements reached with conditions on no third-party usage of all testing data to avoid the potential issues of data breaches in this phase. We will erase the data permanently after the development phase or any closed beta tests. For the accidental misuses which may arise in this case, we will keep track of the recording of data accesses to protect every team member.

Typically, fingerprint sensors are not as secure as our proposed voice recognition scheme because fingerprint data can be easily stolen, and sensors can be cheated with a master fingerprint. A Forbes article discussed that researchers from the X-Lab have demonstrated that any fingerprint lock from Android or iPhones can be hacked in 20 minutes [21]. However, the total cost of hacking fingerprints costs more than $142, and the research methodologies on how to replicate fingerprints are not publicly accessible [21]. For the everyday person, we can increase the security of fingerprint encryption by choosing a capacitive fingerprint sensor that requires detection of electrical properties, and one that can achieve high accuracy, such as one with a false positive rate of $< 0.005\%$ [9]. This eliminates the possibility of hacking the sensor with a printed photograph of the fingerprint pattern and increases the cost of breaking in. Fabricating a fingerprint on conductive material would require access to a clear image of the fingerprint and expensive equipment, such as 3D printers with precision on the micrometer scale, that are not commonly accessible.

For training the voice recognition model, our project would require the collection of the user’s voice, which is a form of biometric information and is under severe regulation by local and international laws. When accessing the user’s voice, we must proceed with caution and follow all legal compliances to ensure the user’s privacy and security are protected. In May 2018, the EU passed the General Data Protection Regulation (GDPR) that tightened the regulation of a class of data called “personal information” [22], any form of data that can be used to identify an individual. It requires businesses to own personal information to give their subjects the “right to access” and “delete” those data [22]. In addition, state laws require businesses that possess personal information to destroy the data when they are no longer needed and to send a notice to the subjects in a data breach, such as when their voice data is exposed to hackers trying to access their accounts [22]. Regarding IEEE and ACM Codes of Ethics, the development and employment of voice recognition and fingerprint systems are restricted by ACM Code 1.6 of “respect privacy” and 1.7 of “honor confidentiality” [20]. This part reflects also on the aim of protecting others’ privacy and informs potential dangers stated in IEEE [23].

To adhere to those laws, we must provide the users of our product full transparency regarding the use of voice data. During the product development, we would use public databases and mostly our own voice data to test the product’s functionality. If this were made into a product, we would

need to explain what personal information would be collected, the purpose of collecting those data, and how they will be used. The user will only proceed with registration by checking a box indicating they have reviewed and agreed to those terms. The user's voice will be the only personal data that we would access for training the voice recognition model, and it will be stored online or on the device. Once the training phase is completed, the data will be automatically removed. We will use subsequent recordings of the user's voice to update the model, but they will be permanently erased from the device afterward. The fingerprints of the user will need to be stored for recognition. We will never distribute any biometric information to any third party. Users will always hold the right to end the program and amend or delete their information and if they decide to close the account, we will destroy all pertinent information of the user. If we ever notice a data breach, we will be responsible for notifying the affected users in a timely manner. With our promises on the respect of data collection and declaration of responsibility, the potential safety concern over unexpected hacker intrusion to steal user's data still exists. For this type of issue, we will follow the regulation process to help customers track the criminal's legal liability, and we will inform them of this possibility at the beginning.

Our product utilizes Bluetooth technology for wireless communication between the smartphone and the microcontroller. The FCC poses regulations on all RF devices, including the Bluetooth modules. For devices operating below 6 GHz, the allowed energy exposure is 1.6 W/kg [24]. Since we plan to use a Bluetooth LE module that has low power requirements and maximum transmission power of 0.7 mW [25], exposure to the radiation for an extended amount of time would have no adverse effect on health.

We agree to comply with the ethical codes and safety regulations to protect and respect user privacy and develop prevention rules to follow throughout the development and employment to minimize client concerns. We guarantee to take our professional responsibilities to pursue high standards on our project and we expect the wide usage of this system to contribute to the society with advanced security.

References

[1] “Pick Proof Locks – Are They Worth It?,” *NGCL*, 13-Aug-2019. [Online]. Available: https://www.thengcl.co.uk/pick-proof-locks/. [Accessed: 19-Feb-2021].

[2] “Learn about the Nest × Yale Lock before you buy,” *Google Nest Help*. [Online]. Available: https://support.google.com/googlenest/answer/9251009?hl=en#zippy=,does-it-support-voice-commands. [Accessed: 19-Feb-2021].

[3] “Burglary Statistics & Research from the BSJ and FBI: The Zebra,” Burglary Statistics & Research from the BSJ and FBI | The Zebra. [Online]. Available: https://www.thezebra.com/resources/research/burglary-statistics/. [Accessed: 18-Feb-2021].

[4] B. C. Corum, “NFC vehicle keys let Mercedes drivers go keyless,” SecureIDNews, 16-May-1970. [Online]. Available: https://www.secureidnews.com/news-item/nfc-vehicle-keys-let-mercedes-drivers-go-keyless/. [Accessed: 18-Feb-2021].

[5] “Marriott expands mobile requests as keyless room entry takes flight,” *Latest News*. [Online].Available:https://www.marketingdive.com/ex/mobilemarketer/cms/news/strategy/21718.html. [Accessed: 18-Feb-2021].

[6] “Password strength,” *VoiceThread*. [Online]. Available: https://voicethread.com/howto/password-strength/. [Accessed: 18-Feb-2021].

[7] “The Complete Guide to Speech Recognition Technology,” *Globalme*, 05-Jan-2021. [Online]. Available: https://www.globalme.net/blog/the-present-future-of-speech-recognition/. [Accessed: 18-Feb-2021].

[8] “Speech-to-Text basics | Cloud Speech-to-Text Documentation,” *Google*. [Online]. Available: https://cloud.google.com/speech-to-text/docs/basics. [Accessed: 18-Feb-2021].

[9] B. Zuo, “Grove - Capacitive Fingerprint Scanner/Sensor,” seeedstudio. [Online]. Available: https://wiki.seeedstudio.com/Grove-Capacitive-Fingerprint-Sensor/. [Accessed: 17-Feb-2021].

[10] Trevorbye, “Speaker Recognition quickstart - Speech service - Azure Cognitive Services,” *Speaker Recognition quickstart - Speech service - Azure Cognitive Services | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/get-started-speaker-recognition?tabs=script&pivots=programming-language-cpp.[Accessed:02-Mar-2021].

[11] “Android's Bluetooth latency needs a serious overhaul,” *SoundGuys*, 12-Apr-2019. [Online]. Available: https://www.soundguys.com/android-bluetooth-latency-22732/. [Accessed: 18-Feb-2021].

[12] K. Townsend, “Introducing the Adafruit Bluefruit LE UART Friend,” Adafruit Learning System. [Online]. Available: https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-uart-friend/current-measurements. [Accessed: 01-Mar-2021].

[13] “Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications,” *International Journal of Wireless Information Networks*. [Online]. Available: https://link.springer.com/article/10.1007/s10776-017-0357-0

[14] “ATmega328P Automotive - Complete Datasheet,” *Microchip*. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf [Accessed: 17-Feb-2021].

[15] A. Industries, “Lock-style Solenoid - 12VDC,” adafruit industries blog RSS. [Online]. Available: https://www.adafruit.com/product/1512. [Accessed: 17-Feb-2021].

[16] “LM1117 Datasheet by Texas Instruments,” Texas Instruments. [Online]. Available: https://www.digikey.com/htmldatasheets/production/2065787/0/0/1/lm1117.html?utm_adgroup=xGeneral&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=&utm_content=xGeneral&gclid=Cj0KCQiA7NKBBhDBARIsAHbXCB5nCk_Vd_1WIXL5eEdEjNCL38qQri6WXfEIvoLMX6WtuK7oaiMSFzUaAnmEEALw_wcB [Accessed: 1-Mar-2021]

[17] “Baud,” Wikipedia, 27-Dec-2020. [Online]. Available: https://en.wikipedia.org/wiki/Baud. [Accessed: 01-Mar-2021].

[18] “How high of a baud rate can I go (without errors)?,” Arduino Stack Exchange, 01-Jan-1963. [Online].Available: https://arduino.stackexchange.com/questions/296/how-high-of-a-baud-rate-can-i-go-without-errors. [Accessed: 01-Mar-2021].

[19] “Annual Reports,” *Illini Success*. [Online]. Available: https://illinisuccess.illinois.edu/annual-reports/. [Accessed: 03-Mar-2021].

[20] ACM (Association for computing machinery) Code of Ethics and Professional Conduct, 2016. Available: https://www.acm.org/code-of-ethics

[21] D. Winder, “Hackers Claim 'Any' Smartphone Fingerprint Lock Can Be Broken In 20 Minutes,” Forbes, 02-Nov-2019. [Online]. Available: https://www.forbes.com/sites/daveywinder/2019/11/02/smartphone-security-alert-as-hack

ers-claim-any-fingerprint-lock-broken-in-20-minutes/?sh=4d37a44d6853. [Accessed: 17-Feb-2021].

[22] J. J. Lazzarotti and M. Atrakchi, "As Voice Recognition Technology Market Surges, Organizations Face Privacy and Cybersecurity Concerns," *Voice Recognition Tech Privacy and Cybersecurity Concerns*, 10-Dec-2020. [Online]. Available: https://www.natlawreview.com/article/voice-recognition-technology-market-surges-organizations-face-privacy-and. [Accessed: 12-Feb-2021].

[23] IEEE (Institute of Electrical and Electronics Engineers) Code of Ethics, 2015. Section 7 - Professional Activities, Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[24] "Wireless Devices and Health Concerns," *Federal Communications Commission*, 04-Nov-2020. [Online].Available:https://www.fcc.gov/consumers/guides/wireless-devices-and-health-concerns. [Accessed: 19-Feb-2021].

[25] "Adafruit Industries Bluetooth Module BLUEFRUIT FCC ID S6OBLUEFRUIT," FCC ID, 08-Apr-2013. [Online]. Available: https://fccid.io/S6OBLUEFRUIT.