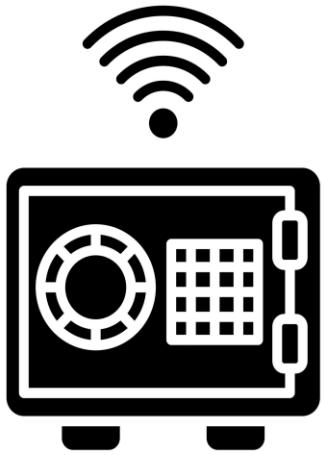
Senior Design 1

Project: Smart Safe



Group 21 Authors:

Nickolas Brandenburg - Computer Engineering
Lakshmi Katravulapalli - Computer Engineering
Alexis Mui - Electrical Engineering
Eric Terry - Electrical Engineering

Reviewer Committee:

Mark Maddox	**ECE**	**Lecturer**
Wei Sun	**ECE**	**Associate Professor**
Saleem Sahawneh	**ECE**	**Lecturer**
Mike Borowczak	**ECE**	**Associate Professor**

University of Central Florida
Department of Electrical and Computer Engineering

Mentor:
Lei Wei

Table Of Contents

Chapter 1 Executive Summary 1
Chapter 2: Project Description 2
2.1 Project Motivation and Background 2
2.2 Project Goals and Objectives 2
2.3 Project Features and Functionalities 3
2.3.1 Basic Goals: 3
2.3.2 Advanced Goals: 3
2.3.3 Stretch Goals: 3
2.4 Project Requirements and Specifications 4
Table 1: Specifications Table 4
2.5 Current/Existing Technologies 4
2.5.1 Ring Doorbell 4
2.5.2 Amazon Hub Locker 5
2.5.3 BILLCONCH Safe 5
2.5.4 Smart Lock Group 6 - Summer 2019 to Fall 2019 5
2.5.5 Smart Cooler Group 23 - Spring 2022 to Summer 2022 5
2.6 Block Diagrams 5
2.6.1 Work Division 6
Figure 1: Hardware Block Diagram [3] 6
2.6.2 Description of Software Diagram 6
Figure 2: Software Block Diagram [2] [7] 7
2.7 House of Quality 7
Figure 3: House of Quality 8
2.8 Prototype Illustration/Sketch 8
Figure 4: Sketch of the Prototype 9
Chapter 3: Research and Investigation 9
3.1 Existing Similar Projects and Products 9
3.1.1 Honeywell 5403 Bluetooth Durable Smart Security Safe 9
3.1.2 FORFEND Fingerprint Smart Home Safe 10

3.1.3 TurboSAFE WiFi Smart Safe 10
3.1.4 Similar Existing Products Summary 10
Table 2: Comparison of Similar Existing Products 11
3.2 Technology 12
3.2.1 MCU 12
3.2.1.2 What are Microcontrollers? 12
3.2.1.3 What are Microprocessors? 12
3.2.1.4 Arduino 12
Table 3: Comparison of Arduino models [20] [21] [22] 12
3.2.1.4.1 Microchip's ATmega2560 12
3.2.1.5 Raspberry Pi 12
Table 4: Comparison of Raspberry Pi models [25] [26] [27] [28] 12
3.2.1.5.1 Broadcom's ARM Cortex-A72(BCM2711) 12
3.2.1.6 ESP32 12
Table 5: Comparison of ESP32 models [32] [33] [34] [35] 12
3.2.1.6.1 Espressif's ESP32-WROOM-32D 12
3.2.1.7 Final Decision 12
3.2.2 Keypad 12
3.2.2.1 Membrane Keypad 12
3.2.2.2 Mechanical Keypads 12
3.2.2.3 Touch-Sensor Keypads 12
3.2.2.4 Keypad Technology Decision 12
Table 6: Comparison of Keypad Technology [123] 12
3.2.2.5 Keypad Part Selection 12
3.2.2.6 NEXTION 16 Keys Matrix Keypad 4x4 Membrane Keyboard Module 12
3.2.2.7 E-outstanding 4x4 Keypad 16 Key Self Adhesive Membrane Matrix Switch Keyboard 12
3.2.2.8 Adafruit 3x4 Phone-style Matrix Keypad 12
3.2.2.9 Part Selection Decision 12
Table 7: Comparison of Existing Membrane Keypads [38] 12
3.2.3 Fingerprint Scanner 12

3.2.3.1 Fingerprint Scanner Technology Comparison 12
3.2.3.2 Capacitive Fingerprint Sensor 12
3.2.3.3 Optical Fingerprint Sensor 12
Figure 5: Optical Fingerprint Sensor Diagram [58] 12
3.2.3.4 Thermal Fingerprint Sensor 12
3.2.3.5 Fingerprint Sensor Technology Decision 12
Table 8: Comparison of Fingerprint Sensor Technology 12
3.2.3.6 Fingerprint Scanner Part Selection 12
3.2.3.7 ADH Tech's GT-521F52 12
3.2.3.8 Hangzhou Grow Tech's GROW R307 12
3.2.3.9 Handson Technology's AS608 Optical Fingerprint Sensor Module 12
3.2.3.10 Part Selection Decision 12
Table 9: Comparison of Existing Fingerprint Scanner Parts 12
3.2.4 Lock Mechanism 12
3.2.4.1 Lock Mechanism Technology Comparison 12
3.2.4.2 In-line Linear Actuator Lock 12
3.2.4.3 Motorized Gear Lock 12
3.2.4.4 Solenoid Lock 12
3.2.4.5 Lock Mechanism Technology Decision 12
Table 10: Comparison of Lock Mechanism Technology 12
3.2.4.6 Lock Mechanism Part Selection 12
3.2.4.7 Elecrow's LY-01 DC12V 12
3.2.4.8 DFRobot's FIT0624 12
3.2.4.9 Uxcell's 1240L-12E07 12
Table 11: Comparison of Existing Lock Mechanism Parts 12
3.2.4.10 Part Selection Decision 12
3.2.5 Motion Sensor 12
3.2.5.2 Ultrasonic Sensor 12
Ultrasonic Sensor Distance Equation [60] (1) 12
Figure 6: Ultrasonic Sensor Pinout [61] 12

3.2.5.3 Infrared Sensors 12
3.2.5.4 Microwave Sensors 12
3.2.5.5 Sensor Selection 12
3.2.5.6 Ultrasonic Sensor Options 12
3.2.5.7 Elegoo HC-SR04 12
3.2.5.8 DigiKey URM37 V5 12
3.2.5.9 SparkFun TCT-40 12
3.2.5.10 Ultrasonic Sensor Selection 12
Table 13: Comparison Table of Ultrasonic Options 12
3.2.6 Display Screen 12
3.2.6.1 LCD (Liquid Crystal Display) 12
3.2.6.2 OLED (Organic Light-Emitting Diode) 12
3.2.6.3 LED (Light-Emitting Diode) 12
3.2.6.4 Display Type Selection 12
Table 14: Display Type Comparisons 12
3.2.6.5 LCD Part Options 12
3.2.6.6 LILYGO ESP32-S3 1.9-inch ST7789 LCD Display Touch Screen 12
3.2.6.7 FREENOVE I2C LCD 1602 Module 12
3.2.6.8 ESP32 Touchscreen 2.8inch TFT LCD 12
3.2.6.9 LCD Part Selection 12
Table 15: LCD Display Comparisons 12
3.2.7 Backup Battery System 12
3.2.7.1 Battery Technology Comparison 12
3.2.7.2 Alkaline Battery 12
3.2.7.3 Lithium Battery 12
3.2.7.4 Battery Technology Decision 12
Table 16: Comparison of Battery Technology 12
3.2.7.5 Battery Part Selection 12
3.2.7.6 Great Power's 18650 Lithium Ion Cell 12
3.2.7.7 Samsung's Lithium Ion Flat Top Battery 12

3.2.7.8 Tenergy Li-Ion Flat Top Rechargeable Battery 12
Table 17: Comparison of Existing Battery Products 12
3.2.7.9 Part Selection Decision 12
3.2.8 Voltage Regulators 12
3.2.8.1 What are Voltage Regulators? 12
3.2.8.2 Linear Regulators 12
Figure 7: Linear Regulator Schematic [83] 12
Figure 8: Series & Switching Regulator Orientations [79] 12
3.2.8.3 Switching Regulators 12
3.2.8.4 Linear vs Switching Regulators 12
3.2.8.5 Voltage Regulator Part Selection 12
3.2.8.6 STMicroelectronics' LD1117V33 12
3.2.8.7 UMW's HT7333 12
3.2.8.8 EVVOSEMI's AMS1117 12
3.2.8.9 Final Decision 12
Table 18: Comparison of Voltage Regulator models [88] [89] [90] 12
3.2.9 Relay 12
3.2.9.1 Relay Technology Comparison 12
3.2.9.2 Electromechanical Relay 12
3.2.9.3 Solid State Relay 12
3.2.9.4 Relay Technology Decision 12
Table 19: Comparison of Relay Technology 12
3.2.9.5 Relay Part Selection 12
3.2.9.6 Omron Electronics' G6E-134P-ST-US DC5 12
3.2.9.7 Songle Relay's JQC-3F-5VDC 12
3.2.9.8 Songle Relay's SRD-05VDC-SL-C 12
Table 20: Comparison of Existing Relay Parts 12
3.2.9.9 Part Selection Decision 12
3.2.10 Buzzer 12
3.2.10.1 What are Buzzers? 12

3.2.10.2 Electromagnetic 12
3.2.10.3 Magnetic 12
3.2.10.4 Piezoelectric 12
3.2.10.5 Pololu Electromagnetic Buzzer 12
3.2.10.6 bnafes Magnetic Buzzer 12
3.2.10.7 Adafruit's Piezo Buzzer 12
Table 21: Comparison of Buzzer models 12
3.2.10.8 Final Decision 12
3.2.11 Status LEDs 12
3.2.11.1 What are LEDs 12
3.2.11.2 Single color LEDs 12
3.2.11.3 Tricolor LEDs 12
Figure 9: RGB LED Pinout [109] 12
3.2.11.4 High-Power LEDs 12
3.2.11.5 Single vs Tricolor vs High-Power 12
3.2.11.6 Shenzhen Fedy Technology's Common Anode LED 12
3.2.11.7 DIYables RGB LED Module 12
3.2.11.8 CHANZON's RGB Common Cathode LED 12
3.2.11.9 Final Decision 12
3.2.12 Website 12
3.2.12.1 Website Development Technologies 12
3.2.12.2 HTML 12
3.2.12.3 CSS 12
3.2.12.4 JavaScript 12
Table 22: Explaining Technology Importances [113] 12
3.2.12.5 Website Technology Use 12
3.2.12.6 Server Hosting Options 12
3.2.12.7 Microcontroller Server 12
3.2.12.8 GitHub Pages 12
3.2.12.9 Firebase Hosting 12

Table 23: Listing Differences in Web Server Hosting Options [114] 12
3.2.12.10 Server Hosting Selection 12
3.2.13 Email Notifications 12
3.2.13.1 Email Protocols 12
3.2.13.2 Simple Mail Transfer Protocol (SMTP) 12
3.2.13.3 Post Office Protocol 3 (POP3) 12
3.2.13.4 Internet Message Access Protocol (IMAP) 12
3.2.13.5 Protocol Decision 12
Table 24: Comparison of Email Protocols [120] 12
3.2.13.6 Email Service Providers 12
3.2.13.7 Postfix 12
3.2.13.8 Postmark 12
3.2.13.9 SendGrid 12
3.2.13.10 Gmail SMTP 12
3.2.13.11 Email Server Decision 12
Table 25: Comparison of Email Service Providers [120][121] 12
3.2.13.12 Email Platforms 12
3.2.13.13 Gmail 12
3.2.13.14 Microsoft Outlook 12
3.2.13.15 Yahoo 12
3.2.13.16 Email Service Platform Decision 12
Table 26: Comparison of Email Service Platforms [120] 12
3.2.13.17 Programming Language for Email Server Testing 12
3.2.13.18 Python 12
3.2.13.19 C++ 12
3.2.13.20 JavaScript (using Node.js) 12
3.2.13.21 Programming Language Decision 12
Table 27: Comparison of Programming Languages for Email Notifications [120] 12
3.2.13.22 Software Libraries 12
Table 28: Comparison of Software Libraries [117] 12

3.2.13.23 Decision 12
3.2.14 Software Development Tools 12
3.2.14.1 Platform IO 12
3.2.14.2 Visual Studio Code 12
3.2.14.3 Putty 12
3.2.14.4 Git 12
3.2.14.5 GitHub 12
3.2.14.6 Python 12
3.2.14.7 C++ 12
3.2.14.8 Java 12
3.2.14.9 HTML 12
3.2.14.10 JavaScript 12
3.2.14.11 C 12
3.2.14.12 PHP 12
Table 29: Comparison of Software Development Tools 12
3.2.15 Hardware Design Software 12
3.2.15.1 Eagle Software 12
3.2.15.2 Code Composer Studio 12
3.2.15.3 Arduino IDE 12
3.2.15.4 KiCad 12
3.2.15.5 Autodesk Fusion 360 12
Table 30: Comparing Relevant Software for Hardware Components 12
Chapter 4: Standards and Design Constraints 12
4.1 Security Standards 12
4.1.1 ISO/IEC 27001: Information Security Management Standard 12
4.1.2 RFC 5246 (TLS 1.2): Transport Security Layer 13
4.1.3 RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email 13
4.2 Biometric Recognition Standards 14
4.2.1 ANSI/NIST-ITL 1-2011 14
4.2.2 ANSI/INCITS 381-2004 14

Figure 10: ANSI/INCITS 381-2004 fingerprint data structure [141] 15
4.2.3 ANSI/INCITS 377-2004 15
Figure 11: ANSIS/INCITS 377-2004 fingerprint cell structure [141] 15
Figure 12: ANSI/INCITS 381-2004 fingerprint data structure [141] 16
4.2.4 ANSI/INCITS 378-2004: 16
4.3 Communication Standards 17
4.3.1 IEEE 802.11 (Wi-Fi Standard) 17
4.3.2 RFC 5321: Simple Mail Transfer Protocol 17
4.4 Website Standards 17
4.5 Ethical Standards 18
4.6 PCB Standards 18
4.6.1 IPC-2221 19
Figure 14: IPC-2221 Structure [144] 19
4.7 Electricity Standards 19
4.8 Time Constraint 19
4.9 Size Constraint 20
4.10 Budget Constraint 20
4.11 Environmental Factors Constraint 21
4.12 Material Constraints 21
4.13 Power Constraints 22
Chapter 5: Comparison of ChatGPT with other Similar Platforms 23
Case Study 1: Choosing the Email Protocol: what are 3 different email protocols I could use to discuss for my paper? 23
Case Study 2: Do you think SMTP is best for this task, or is IMAP or POP better? 24
Comparison of SMTP, IMAP, and POP for Your Task: 24
Summary: 25
Case Study 3: What are all the pros and cons of using a microcontroller versus using a microprocessor for an embedded system? 25
Microcontroller 25
Microprocessor 26
Case Study 4: What are 3 types of relays should I use to discuss for my project report? 28

Chapter 6: Hardware Design 29
6.1 Subsystem Block Diagram 29
Figure 15: Subsystem Block Diagram 29
6.2 Schematic Diagram 29
6.2.1 ESP-WROOM-32 Schematic 29
6.2.2 Keypad Schematic 30
Figure 17: Wiring Diagram for the Keypad 30
6.2.3 Fingerprint Scanner Schematic 31
Figure 18: Wiring Diagram for the Fingerprint Scanner 31
Figure 19: Wiring Diagram for the Locking Mechanism 32
Figure 20: Wiring Diagram for the Ultrasonic Sensor 32
6.2.4 Display Schematic 32
Figure 21: Wiring Diagram for the Display 33
6.2.5 Buzzer Schematic 33
Figure 22: Wiring Diagram for the Buzzer 33
6.2.6 LED Schematic 34
Figure 23: Wiring Diagram for the Buzzer 34
6.2.7 Voltage Regulators 34
Figure 24: Wiring Diagram for the 3.3V & 5V Voltage Regulators 34
6.2.8 DC Power Jack Connector 35
Figure 25: Wiring Diagram for the DC Power Jack 35
6.3 Overall Schematic 35
Figure 26: Complete Schematic Design for the Smart Safe 36
Chapter 7: Software Design 37
7.1 UML Class Diagram 37
Figure 27: UML Class Diagram [148] [149] [150] 38
7.2 Use Case Diagram 38
7.3 Website Application Design and UI 39
Figure 29: UI Website Design/Prototype [161] 39
7.4 User interface for email 40
Figure 30: Email Notification Example 40

Figure 31: Motion Notification Figure 32: Lockdown Mode Notification 41
7.5 Display Prototypes and UI 41
Figure 34: Display Unlock Prototype 41
Figure 35: Display Lockdown Mode Prototype 42
Figure 36: Display Fingerprint Prototype ..... **Error! Bookmark not defined.**
Figure 37: Display Valid Fingerprint Prototype 42
Figure 38: Display Invalid Prototype 42
Figure 39: Display Locked Prototype 42
7.6 Data Transfer 42
Chapter 8: System Fabrication/Prototype Construction 44
8.1 PCB Layer Types 44
8.1.1 Substrate 44
8.1.2 Copper-Conductive layers that form the electrical traces and pads 44
8.1.2.1 Traces 45
8.1.2.2 Pads 45
8.1.2.3 Planes 45
8.1.3 HASL 45
8.1.4 Solder Mask 46
8.1.5 Silkscreen 46
8.2 PCB Material Type 46
8.2.1 FR-4 46
8.2.2 Electrolytic Copper 46
8.3 PCB Design Layout 47
Figure 40: Ground Plane on Top Layer 48
Figure 41: Ground Plane on Bottom Layer 49
Figure 42: Ground Plane from Bottom Layer 49
Chapter 9: System Testing 50
9.1 Hardware Testing 50
9.1.1 ESP-WROOM-32 50
Figure 43: ESP32 powering Figure 44: ESP32 in the terminal 50
9.1.2 Keypad 50

Figure 45: Keypad Component Testing 52
Figure 46: Keypad Connected for Testing 52
9.1.3 Fingerprint Scanner 52
Figure 47: Unregistered fingerprint Figure 48: Unregistered fingerprint results 53
Figure 49: Registered fingerprint Figure 50: Registered fingerprint results 53
9.1.4 Locking Mechanism 53
Figure 52: Solenoid Unlocked 55
9.1.5 Ultrasonic Sensor 55
Figure 54: Ultrasonic Sensor Results (close) 56
Figure 55: Ultrasonic Sensor Testing (far) 56
Figure 56: Ultrasonic Sensor Results (far) 57
9.1.6 Buzzer 57
Figure 57: Buzzer connected to buzzer via breadboard 57
Figure 58: LED glowing red from ESP32 board 58
9.1.7 Display 58
Figure 59: Display Component Testing 59
9.1.8 Web Server 59
Figure 60: Web Server Testing Arduino 60
Figure 61: Web Server Testing Screenshot 60
9.1.9 Email 60
Figure 62: Email Testing Arduino 61
Figure 63: Email Testing Screenshot 61
9.2 Performance Evaluation 61
9.3 Overall Integration 62
Chapter 10 Administrative Content 63
Table 31: Bill of Materials [10] 63
Table 32: Project Timeline 64
Table 33: Senior Design 2 Project Milestones 65
Chapter 11: Conclusion 66
Appendix A – References 67
Appendix E – ChatGPT prompts and outcomes 81

Suggested Structure 81
Explanation of <<include>> in Your Diagram 81
1. Website-to-Microcontroller Communication 82

Chapter 1 Executive Summary

As technology has evolved over time, consumers have increasingly sought to incorporate more tech into their daily lives for greater convenience. Homes have gotten smarter with devices like smart light switches, door locks, and even fridges. While technology keeps advancing, the demand for safety has always been a priority. And as people travel more or lead busier lives, the demand for convenience and security continues to grow.

Safes are a popular choice when people want to secure their personal belongings, whether at home or in a hotel. Typically, these safes tend to be portable due to it being made from a lightweight material and designed to be easy to use. Traditional safes are usually made of metal and feature a key that is used to lock and unlock the door. There are also combination safes, where you turn the dial and align the correct numbers in the right order to open it like a high school locker.

However, traditional safes have their drawbacks. The key can easily be misplaced, the combination lock being jammed, and it's possible to forget the code over time. If that happens, you may need to replace the lock, key, or the whole entire safe which can be costly.

Digital safes offer a modern solution to these drawbacks. These typically have a keypad where you can enter a passcode to unlock the safe and store your valuables. Some digital safes even offer additional alternative ways to access the safe outside of a keypad input. These options include voice recognition, fingerprint scanners, or even a mobile application that allows you to control and monitor the safe remotely from your phone.

The Smart Safe is our attempt at an improved version of digital safes commonly found in the market. It features both a keypad and fingerprint scanner, giving users two distinct options to access the safe to store their valuables. The safe also includes a display showing the user the status of the safe. Additionally, the safe will also include a motion sensor that will detect if someone is close enough to be able to access the safe and would alert a user's email. The owner will receive real-time notifications about their safe, keeping them alerted and updated.

This report outlines the entirety of the Smart Safe's design process from conception to implementation. We begin by discussing the motivations and objectives behind the Smart Safe, along with its requirements. Then, we detail the research and part selection process for each major component of the safe in full explanation. Next, we mention all the relevant standards and constraints that influenced our Smart Safe. We also elaborate on our design, implementation, testing, and integration process from both a hardware and software perspective. Finally, we present our conclusion and appendix containing all our references and copyright permissions.

Chapter 2: Project Description

In this chapter, we will provide an overview of our project, discussing both the motivation behind it and its primary purpose. We'll begin by outlining our project's main goals and objectives, followed by a detailed description of the smart safe's features and functionalities. Each function will be categorized into basic, advanced, and stretch goals. A specifications table will summarize the key requirements and objectives, establishing a benchmark for performance. Next, we'll explore existing technologies to compare our project with similar solutions, highlighting shared and unique attributes. Then, we'll present our software and hardware block diagrams, illustrating the work distribution and individual contributions of each team member. Finally, the chapter will conclude with the House of Quality diagram and a prototype sketch of the smart safe [11].

2.1 Project Motivation and Background

Our project is driven by the need for a highly secure storage solution for personal valuables. In today's world, people require a reliable means to safeguard important items such as passports, personal documents, jewelry, emergency cash, and firearms. Traditional safes offer physical security but may lack advanced protection features. To address this, we aim to develop a digital smart safe that enhances security through modern technology, making users feel more confident about the safety of their belongings.

Our smart safe eliminates the inconvenience of physical keys or cards by offering alternative access methods: a keypad and a fingerprint sensor. Registered users can receive real-time email notifications if any access attempts are made, enhancing their ability to monitor and respond to potential security breaches. Additionally, the integration of a motion sensor and timestamp functionality ensures users are aware of any activity around the safe. By combining these advanced features, our project aims to provide a comprehensive security solution with enhanced protection against theft and unauthorized access.

2.2 Project Goals and Objectives

The goal of this project is to design a smart safe equipped with advanced security features such as a 6-digit keypad code and fingerprint scanner, ensuring secure and convenient access. Additionally, we aim to develop a website where a registered user can input their email to receive notifications whenever the safe is accessed, allowing them to monitor for any unauthorized entry. The safe will also incorporate a motion sensor that detects nearby activity, and a timestamp will be provided to indicate the exact time the safe was accessed.

The keypad will require a 6-digit code, denying access if an incorrect code is entered. To enhance security, if the code is entered incorrectly three times, the safe will enter Lockdown Mode, where only the fingerprint of a registered user can disengage it and access the safe. Multiple users can register their fingerprints for authorized access. To further ensure protection, electromagnetic solenoid locks will secure the safe and prevent

tampering or breakage. Our team aims to create a flexible and user-friendly system, so users can securely store their belongings with ease and confidence.

2.3 Project Features and Functionalities

Our smart safe is designed with a variety of features that provide both security and user convenience. These functionalities ensure that the safe offers multiple layers of protection, making it highly secure and easy to operate. Below are the key features of the safe and our goals:

2.3.1 Basic Goals:

- **Keypad:** A 6-digit security code is required to access the safe. If the incorrect code is entered, access is denied, preventing unauthorized attempts
- **LCD Screen:** Displays clear and readable user prompts during interactions with the safe.
- **Notifications:** Sends timely email alerts to the primary user when access attempts are made, including a timestamp of when the safe was accessed.
- **Solenoid Lock:** Unlocks the safe upon recognizing a registered fingerprint or correct passcode and remains locked if an unregistered fingerprint or incorrect passcode is entered.

2.3.2 Advanced Goals:

- **Fingerprint Scanner:** Accurately reads registered fingerprints and allows additional fingerprints to be added without issues, ensuring that access is limited to authorized users only.
- **Motion Sensor:** Detects individuals who are within proximity to interact with the safe, helping monitor attempts to access it.
- **Website:** Provides a web interface where registered users can input their email to receive notifications and manage safe settings.
- **Lockdown Mode:** Locks the safe after three consecutive incorrect passcode entries, requiring a registered fingerprint to disable the mode and regain access.
- **Buzzer Alert:** Provides audible signals—one buzzer sound indicates the safe is still unlocked, two sounds mean the safe is locked, and repeated buzzing signifies that Lockdown Mode is active after failed access attempts.
- **LED Status Light:** Visually indicates whether the safe is currently locked or unlocked.
- **Self-Locking Mechanism:** Automatically locks the safe after being unlocked for approximately 30 seconds if no further access is requested.
- **Battery Backup:** Keeps the safe operational during power outages or in case of a main power supply failure, ensuring that the system continues to function seamlessly.

2.3.3 Stretch Goals:

- **Camera Module**: Captures images of individuals attempting to access the safe for enhanced monitoring.
- **Motion Sensor Time Limit**: Sets a specific time window for motion detection.
- **Tamper Detection**: Alerts the user in case of any tampering or unauthorized attempts to breach the safe.
- **Temperature sensor**: Detects the temperature of the safe and alerts the user if temperature is above or below predefined temperature range.

2.4 Project Requirements and Specifications

The following table is a list of all of our specifications for the Smart Safe. The ones that are highlighted in blue will serve as our three demonstrable specifications.

Table 1: Specifications Table

Specifications	
*Fingerprint access accuracy	Unlocks safe 90% of the time (unless it's an unregistered fingerprint)
*Passcode access accuracy	Unlocks safe 90% of the time (unless it's the wrong code)
Status LED accuracy	Is correct 90% of the time
*Email notification duration	Received in under 3 minutes
Lockdown Mode activation	Activates after 3 passcode attempts
Self-Lock Time Limit	Locks after 30 seconds
Safe Size	6.7" x 6.7" x 9"

*The specifications highlighted in yellow will be our 3 demonstrable specifications

2.5 Current/Existing Technologies

In this segment, we will take a look at existing products on the market that are similar to our project design. By examining these products, we'll get will a better understanding of the current landscape and identify key features that work well. This will guide our design choices and help us ensure that our Smart Safe stands out in terms of functionality, security, and user experience.

2.5.1 Ring Doorbell

The Ring Doorbell is a smart home device by Amazon that enhances the traditional doorbell with additional security features. It includes a motion detector that alerts the owner via any mobile device using the Ring App or any Amazon Alexa device when motion is detected at the front door [1]. The doorbell also allows users to see who is near it and speak to them through the mobile app [1]. Additionally, the app enables users to monitor the doorbell from anywhere and save camera footage for up to 180 days. It also allows for customization of alert settings to suit user preferences. The Ring Doorbell is part of a

broader range of security products, including security cameras, alarms, and smart lighting [1]. The battery-operated Ring Doorbell features a built-in rechargeable battery or can be hardwired and includes functionalities such as video with night vision, two-way talk, motion detection and alerts, mobile app control, dual-band Wi-Fi, package alerts, and more [1].

2.5.2 Amazon Hub Locker

The Amazon Hub Locker provides a secure place for customers to pick up their Amazon packages. This fully automated locker system offers a safe alternative for those who prefer not to have packages delivered to their home. When a package is ready for pickup, the customer receives an email containing a 6-digit code that can be used to retrieve their package from the locker [8].

2.5.3 BILLCONCH Safe

The BILLCONCH Safe incorporates features like those planned for our smart safe, including a fingerprint sensor, keypad, and notification system for alerting the owner about safe access. However, the BILLCONCH Safe also includes a physical key for manual access and a mobile app for user information and notifications—features that we have not considered for our design [9].

2.5.4 Smart Lock Group 6 - Summer 2019 to Fall 2019

Group 6's Senior Design Project from Summer 2019 to Fall 2019 was a smart lock system with multiple access methods, including a keypad, RFID, fingerprint sensor, and facial recognition. Their project featured a touchscreen with a GUI interface, allowing homeowners to unlock the door using various methods such as a key fob, fingerprint sensor, or security code. Additionally, their system included Bluetooth and mobile app capabilities [5]. We were inspired by this project's approach to integrating advanced access technologies and its focus on providing a secure and convenient solution for home entry.

2.5.5 Smart Cooler Group 23 - Spring 2022 to Summer 2022

The Smart Cooler project from Spring 2022 to Summer 2022 featured a cooler designed to retain cold air and incorporate smart glass to address issues like cold loss mitigation and privacy. The smart glass prevents cooling loss by allowing visibility without opening the cooler. Additionally, the cooler includes USB ports for charging during outdoor activities, aiming to offer a low-cost and efficient solution with features beyond those of traditional coolers [6]. This project inspired us to enhance our safe with features like battery backup and a status light for improved user convenience.

2.6 Block Diagrams

Our group consists of two Computer Engineers, Nickolas Brandenburg and Lakshmi Katravulapalli, and two Electrical Engineers, Alexis Mui and Eric Terry. The Electrical Engineers will primarily focus on hardware applications, while the Computer Engineers will concentrate on software applications. However, all members will collaborate and engage in both areas.

2.6.1 Work Division

Eric & Alexis will handle all PCB work regarding the MCU and the battery while also integrating the status LEDs and the relay. Lakshmi & Nick are tasked with setting up the web server to be able to send email notifications that someone is accessing the safe. Furthermore, Alexis & Lakshmi are tasked with integrating the motion sensor and fingerprint scanner while Nick & Eric are tasked with integrating the Piezo Buzzer, LCD screen & 4x4 Keypad.

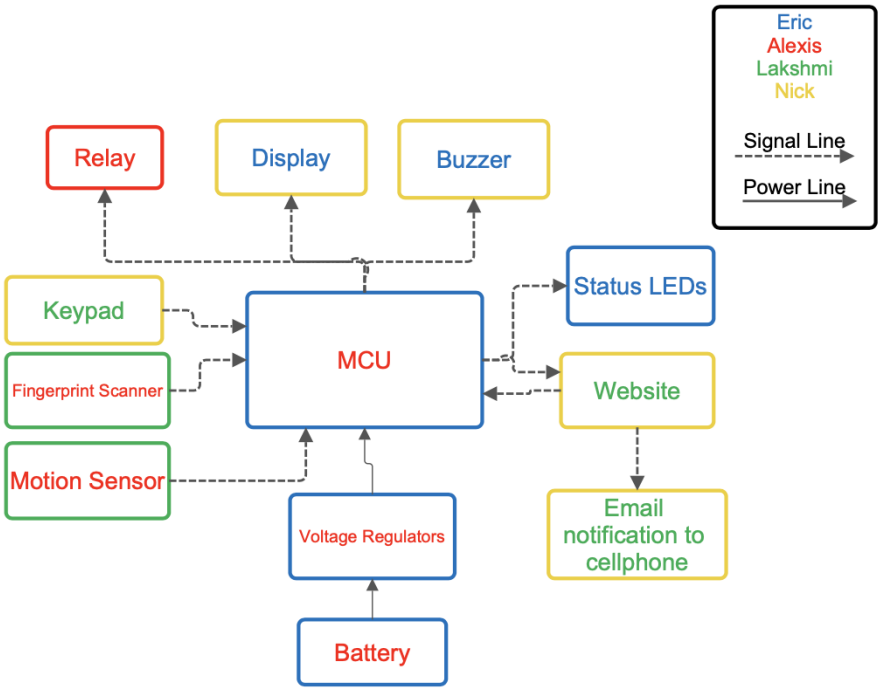


Figure 1: Hardware Block Diagram [3]

2.6.2 Description of Software Diagram

The software diagram above shows how the Smart Safe will work when a user interacts with it. First, the user must be identified as a registered user or an unregistered user. If the user is registered in the system, they may scan their fingerprint to unlock the safe. If the fingerprint is not recognized this is considered an invalid attempt and the safe door will stay locked. If the fingerprint is recognized this will be seen as a valid attempt and the safe door will be unlocked. If the user is unregistered, they will be required to use the keypad to enter a 6-digit security code to unlock the safe. If the code is entered incorrectly then the safe will remain locked. An extra condition is if the user enters the security code on the keypad incorrectly more than 3 times the safe will enter Lockdown Mode. During

Lockdown Mode, the digital safe prevents anyone from trying to access it. The only way to deactivate Lockdown Mode is for a registered user to use the fingerprint scanner. If the safe door is unlocked, the primary owner of the safe will receive an email notification stating their safe has been opened and to verify that this access was authorized. After the safe has been unlocked, the locks will be automatically locked after approximately 30 seconds using the self-locking mechanism. In addition, the safe will have a buzzer sound alert and LED status light. If the user has an invalid attempt the buzzer will make 2 alert noises, and the LED will be red. If the safe is unlocked, the buzzer will make one noise, and the LED will be green. If the safe enters lockdown mode it will make a constant buzzer alert. For the work distribution, Alexis will be working on the parts highlighted in purple which consists of registered users, fingerprint scanner, and making sure the safe door is locked along with the buzzer noise and LED status. Eric will be working on the parts highlighted in blue which consist of unregistered users, keypad, and making sure the safe door is locked along with the buzzer noise and LED status. Lakshmi will oversee when the safe door is unlocked, sending email notifications, and the self-locking mechanism which are highlighted in blue. Nickolas oversees the self-locking mechanism which is highlighted in red. In conclusion, we will all be contributing to the overall tasks.

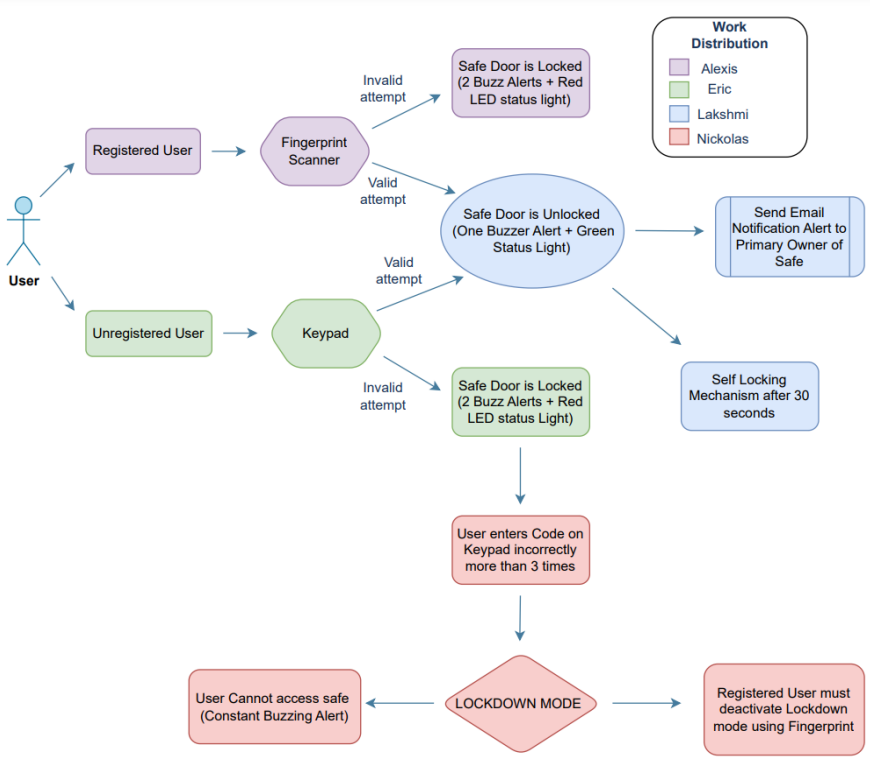


Figure 2: Software Block Diagram [2] [7]

2.7 House of Quality

The house of quality diagram is a chart that labels the relationship between different requirements and engineering fundamentals compared to our target specifications. We can also compare the direction of improvement in terms of how we want to improve each factor as we work on our project. We can identify the correlations between different

features whether they have an inverse or normal relationship in terms of value. This knowledge will help us determine the best possible quality for our project based on our specifications.

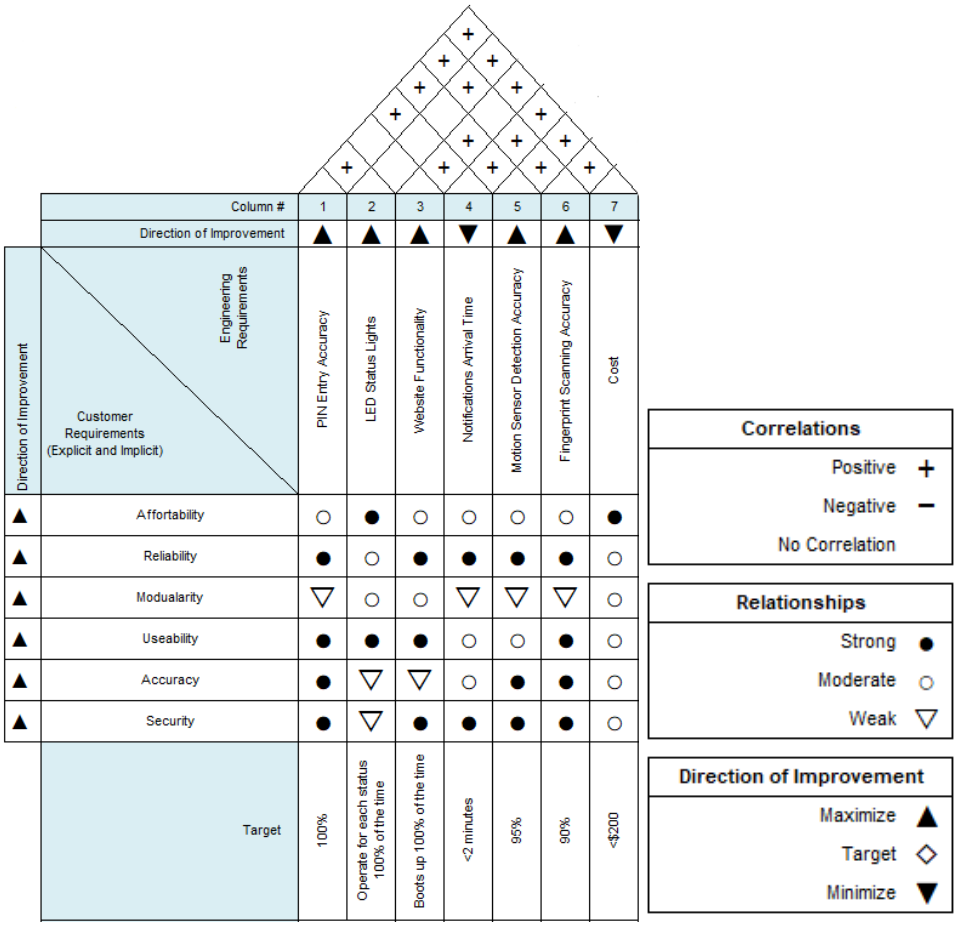


Figure 3: House of Quality

2.8 Prototype Illustration/Sketch

Below is a sketch of our safe design when we were brainstorming and coming up with the component layout and how we were going to build it. The following sketches illustrate the door layout, the direction of the door opening, the type of solenoid we need to use, and a brief example of our door sensor stretch goal.

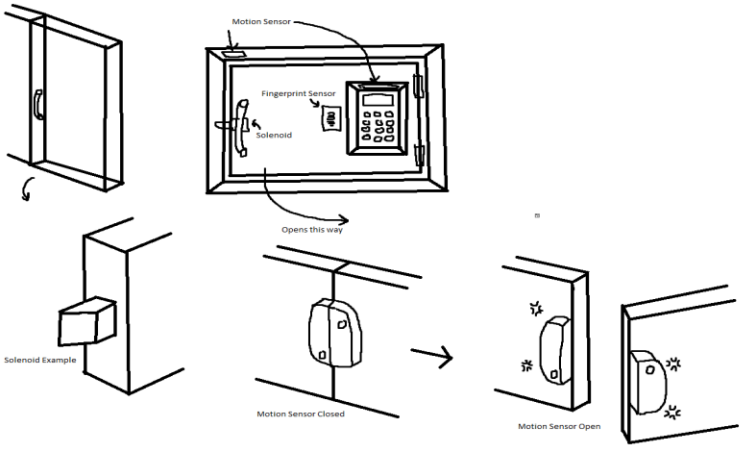


Figure 4: Sketch of the Prototype

Chapter 3: Research and Investigation

In this chapter, we will conduct a comprehensive examination of each component within our project, comparing them against alternative products to ensure we select the most suitable options. For each component, we will provide a technology comparison along with a detailed part selection, supported by tables that outline our reasoning and highlight key features. This approach will ensure that every chosen component meets our project's specifications and contributes to an optimal design for our smart safe system [11].

3.1 Existing Similar Projects and Products

In this section, we will be comparing different existing products in the market that are similar to our Smart Safe. The first product is "Honeywell 5403 Bluetooth Durable Smart Security Safe". This safe has a Bluetooth signal to send alerts which is similar to our safe along with different features. The next product is "FORFEND Fingerprint Smart Home Safe" which has many similar features to our project such as keypad, fingerprint scanner, false attempt alarm, email notifications, etc. The third product is "TurboSAFE WiFi Smart Safe" which has a similar design and lock system to our project. These three products will be compared and evaluated in this section to explore their features and advanced technologies that they offer.

3.1.1 Honeywell 5403 Bluetooth Durable Smart Security Safe

The Honeywell smart safe comes with enhanced Bluetooth enabled security features that can be controlled with the app. In contrast to other safes, this one has a Bluetooth lock that is controlled by its app which can lock and unlock the safe [12]. This safe has a touchscreen keypad with a customized access code and a one-touch lock feature [12]. One feature that is similar to our safe is the Bluetooth signal that sends notifications to users when codes or eKeys were used to open the safe. It can also allow multiple users to access the safe with shared passcodes. Some unique design aspects are LED interior lights and curved door edges to prevent break-ins [12]. It also indicates when the battery is low and has motorized

lock door bolts. This product offers many different unique features and is focused primarily on Bluetooth access and security. The price of this product is currently $539.95 which is reasonably higher than average smart safe products.

3.1.2 FORFEND Fingerprint Smart Home Safe

This product is a smart home safe that comes with a keypad and fingerprint scanner. Some features that this safe provides that are similar to ours is the false-attempt alarm which prevents anyone from using the safe after a certain number of incorrect attempts, frozen mode lockdown, email notifications, and LED light to indicate the status of the safe [13]. Some additional features are kidnap panic alarm, tamper detection alarm, low battery alarm, and gateway alarm [13]. It also comes with voice commands which can be connected with Amazon Alexa or Google Home. Additionally, some items it comes with are batteries, gateway, USB emergency charger, and backup keys [13]. This product can also be controlled by the mobile application that can unlock the safe. The FORFEND Smart Home Safe comes with many advanced capabilities and security features that make it the ideal place to store important belongings. The current price of this product is $159.99 which is a reasonable price for a smart safe.

3.1.3 TurboSAFE WiFi Smart Safe

This product is a smart safe that comes with a keypad, physical key, along with its own mobile app [14]. The size and design space is similar to the prototype illustration of our safe. It comes with pry-resistant hinges and round locking bars which is similar to our plan for the locking system for our safe. This product comes with a TurboSafe monitoring app that updates you on the status of your safe [14]. Unlike other smart safe products, for security concerns the safe cannot be opened directly from the app but can only be unlocked with the passcode using the keypad or the physical keys. The safe comes with motion sensors that detect any movements or if the box is being moved and will immediately alert the user using the app. The physical keys are only used as backup and will sound an alarm as a precaution. The TurboSAFE is an easy and simple to use smart safe with extra heavy duty durability that guarantees to keep your personal items safe [14]. The price of this product is currently $99.99 which is a reasonably affordable price for a Smart Safe.

3.1.4 Similar Existing Products Summary

In this section we summarized similar existing Smart Safes and discussed their features and functionalities. We compared the design qualities and cost of each safe. Table 3.1.5 will compare each of the products and their differences in further detail.

Table 2: Comparison of Similar Existing Products

Product	Access Methods	Features	Cost
Honeywell 5403 Bluetooth Durable Smart Security Safe	Touchscreen Keypad, Mobile App, Bluetooth lock	• Bluetooth Lock Control for App • Touchscreen Keypad • One-Touch Lock on Keypad • Sends notification when Safe is opened • Shared Access • LED interior lights • Low Battery indicator	$539.99
FORFEND Fingerprint Smart Home Safe	Keypad, Fingerprint Scanner, Mobile App, Voice recognition by Alexa/Google Home	• False Attempt Alarm • Kidnap Panic Alarm • Tamper Detection Alarm • Email notifications • LED status lights • Voice Command	$159.99
TurboSAFE WiFi Smart Safe	Keypad, Physical Key	• Monitoring App	$99.99

		• Pry resistant hinges • Password Disguise • Motion Sensor • Alarm if physical keys are used	

3.2 Technology

Chapter 4: Standards and Design Constraints

An important part of our project is making sure that our Smart Safe has met the required standards and policies. In this section, we will discuss the relevant standards we have thoroughly researched and explored. The standards we will be discussing in detail are security standards and biometric recognition standards. The brief standards that we will be discussing are communication standards, website standards, ethical standards, PCB standards and electricity standards. Furthermore, we will also be discussing the design constraints of our project which are certain limitations that will affect the building and designing of our product. The design constraints we will be exploring in detail are time, size, budget, environmental factors, material and power.

4.1 Security Standards

For this section, we will be exploring the different security standards that correspond with the Smart Safe and must be considered when building it during Senior Design 2. Security standards are important because they ensure that the Safe protects user data, and prevents unauthorized access or theft, and improves the credibility of our project. One of the main goals of our project is to prioritize safety and security which is why standards are critical to make sure we meet those requirements.

4.1.1 ISO/IEC 27001: Information Security Management Standard

This standard focuses on protecting the user's sensitive data and handling information security. The goal is to provide a security system that will prevent unauthorized access to any sensitive information regarding the safe. To correctly adhere to this standard, our project will need to implement secure protocols that ensure that every user’s personal information or data is protected.

The ISO/IEC 27001 standards help companies transition into modern security systems and keep up with the enhanced cyber security threats. They make sure that companies have good defenses and security measures to protect sensitive data. It provides a certification

that lasts three years to ensure that organizations are complying with their expectations [125].

4.1.2 RFC 5246 (TLS 1.2): Transport Security Layer

The main goal of the TLS 1.2 is to make sure that email notifications are encrypted which means that unauthorized people cannot access that data. This is critical to guarantee that the communication between the safe and the user's email server is protected. The sensitive data can refer to access notifications or any other sensitive information that might be vulnerable to outside parties. It is important to make sure this data is protected from criminals who might try to intercept or compromise the emails. TLS is usually implemented with email protocols such as SMTP.

TLS 1.2 supports advanced encryption algorithms that confirm that all data sent between user and client is confidential [124]. TLS 1.2 was released in 2008 and added extra flexibility to encryption algorithms as well as making sure that the changes made are stronger against emerging cyberattacks [124]. The TLS protocol will be used when the email notification is triggered from the safe, where the email client and server secure a connection using TLS [124]. After this, a handshake will happen where cryptographic keys are exchanged ensuring that both sides are authorized parties. The data can then be exchanged safely without any problems.

4.1.3 RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email

This security standard verifies that the email sender's domain is valid. This is important to prevent fake users from sending the email which is known as email spoofing. SPF will make sure that the domain of the email sender matches the IP address of the server sending the email [124]. It also helps prevent phishing or email forgery and if the sender of the email is not authorized then it will be marked suspicious. SPF is an important security protocol that will be implemented with our email server for email notifications.

Based on the discussion of the security standards above that will need to be implemented with our project we will discuss some of the important considerations we will need to take into account. The most important consideration is to guarantee that the user's personal information including emails, account information, passcodes, and fingerprint data are safely protected, confidential, and encrypted from unauthorized people. We will need to consider ways to make our safe more protective and prevent any risks that could cause data leaks or breaches. We will also need to implement additional safety measures and regular safety checks. For the TLS standards, we will need to implement this when setting up email notifications as part of our project, and check that there is a secure connection between the smart safe and the client's email server. Similarly, for the SPF protocol we must make sure that the domain of our safe's email server is verified with SPF. In addition, when testing the email, we will check to make sure the emails are sent by the correct sender and not a fake personal. If the SPF protocol is not implemented correctly, it could cause additional

problems when sending the emails. Our group will ensure that security standards are considered for every component in our project and to keep track of future changes.

4.2 Biometric Recognition Standards

Biometric standards are identified as the standards for the collection, storage & sharing of biometric data such as one's fingerprint, eye & face. These standards specify the format of the biometric data, performance metrics, test approaches, application program interfaces and application profiles. There are four types of biometric standards: technical interfaces, data interchange formats, application profile standards and performance testing/reporting [140]. The ones we are focusing on are the data interchange standards for fingerprints, which are ANSI/NIST-ITL 1-2011 [140], ANSI/INCITS 381-2004 [141], ANSI/INCITS 377-2004 [141] and ANSI/INCITS 378-2004 [141].

4.2.1 ANSI/NIST-ITL 1-2011

The ANSI/NIST-ITL 1-2011 standard was produced by both the American National Standards Institute (ANSI) and the National Institute of Standards and Technology (NIST). This standard deals with the content, format and measurement units of the electronic exchange of fingerprint, iris, mugshot and other biometric information. The information regarding this standard is normally used in criminal justice organizations that rely on identification systems for a subject or subjects. The ANSI/NIST-ITL 1-2011 standard has up to 99 unique record identifiers that contain various types of biometric data. However, Type-4 and Type-14 records specifically deal with the scanning and capturing of fingerprint data [142].

4.2.2 ANSI/INCITS 381-2004

The ANSI/INCITS 381-2004 standard was produced by both the American National Standards Institute and the InterNational Committee for Information Technology Standards (INCITS). Just like the ANSI/NIST-ITL 1-2011 standard, the ANSI/INCITS 381-2004 standard deals with how fingerprint data is formatted and measured in an electronic exchange. This standard typically structures its fingerprint data by displaying a header at the top, followed by the fingerprint image & fingerprint data under the header. fingerprint image data will contain information such as the image resolution, pixel depth, pixels per inch, pixels per centimeter, the number of gray levels in the image and the setting level at which all the minimum parameters were satisfied during the capturing of the image [142].

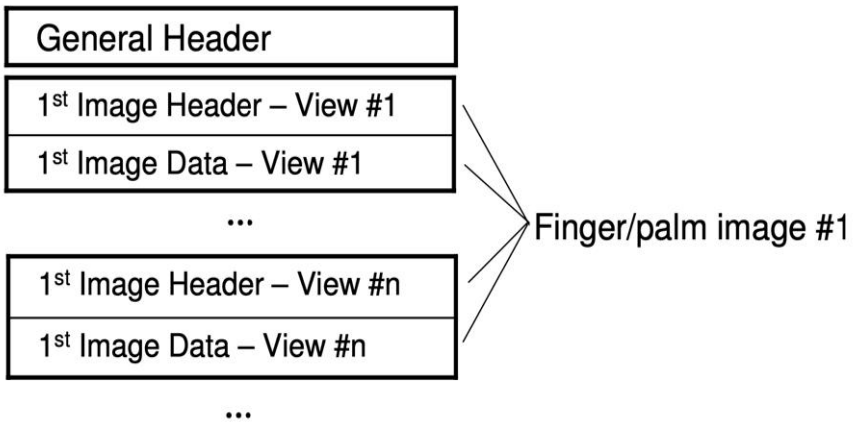


Figure 10: ANSI/INCITS 381-2004 fingerprint data structure [141]

4.2.3 ANSI/INCITS 377-2004

Another ANSI/INCITS fingerprint standard is the ANSI/INCITS 377-2004, also known as The Finger Pattern Based Interchange Format standard. This standard focuses on a specific method of creating biometric templates of fingerprint information using the ridge pattern measurements on the fingerprint. How it works is it takes the image of the fingerprint, reduces it and groups it into cells. Each cell is 5x5 pixels and is measured individually. The parameters for the ridge patterns are its clockwise rotation, the spacing between the ridges and whether the ridge is coincident with the cell's origin. How this format's data is structured is it keeps a log of the data for each unique fingerprint data [2].

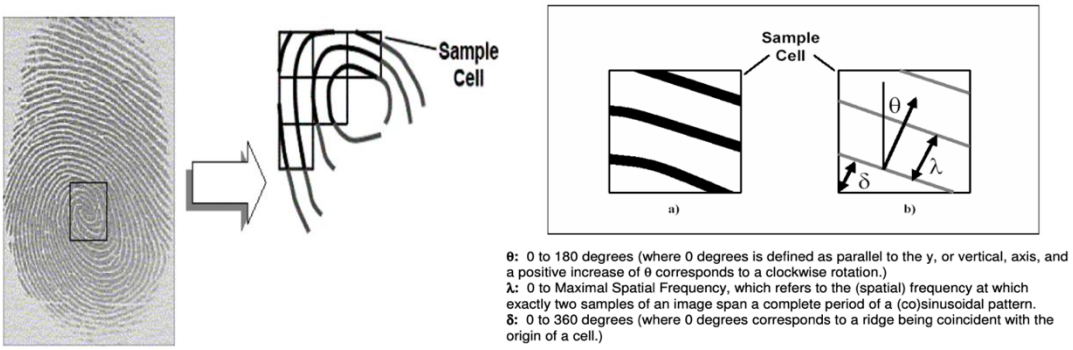


Figure 11: ANSIS/INCITS 377-2004 fingerprint cell structure [141]

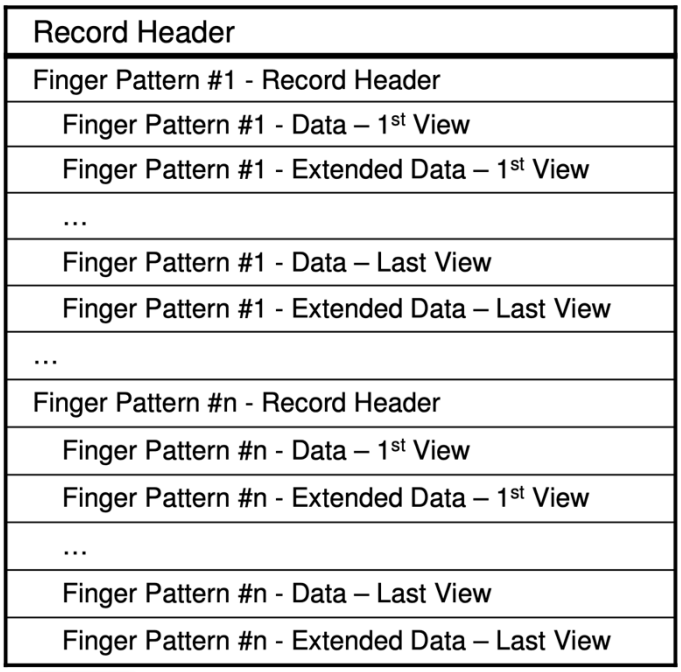


Figure 12: ANSI/INCITS 381-2004 fingerprint data structure [141]

4.2.4 ANSI/INCITS 378-2004:

The ANSI/INCITS 378-2004 standard, also known as the Finger Minutiae Format for Data Interchange standard specifies a method of creating biometric templates of fingerprint minutiae. The parameters for this specification are finger positions, finger impressions, number of ridges, the location of the fingerprint's center and the point of divergence of a ridge. The structure of the data format resembles the one for the ANSI/INCITS 381-2004 standard except it includes the minutia records of each fingerprint [141].

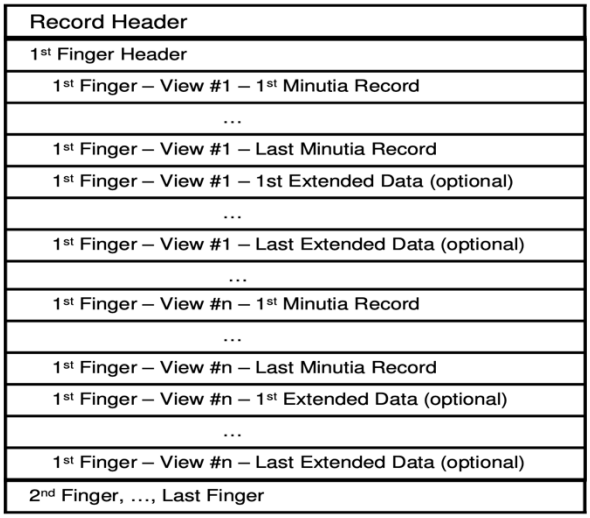


Figure 13: ANSI/INCITS 378-2004 fingerprint data structure [141]

4.3 Communication Standards

Communication standards are crucial to verify that data being sent from the safe to the primary owner is being transmitted safely and efficiently. The first standard we will discuss is IEEE 802.11 also known as the wireless communication standard. The second standard is RFC 5321 which defines the Simple Mail Transfer Protocol.

4.3.1 IEEE 802.11 (Wi-Fi Standard)

The IEEE 802.11 standard which is also known as Wi-Fi is commonly used to control wireless communication across different networks [13]. This standard is important for our project because it defines reliable communication from our safe to external devices. It supports security protocols like WPA2 that ensures data is being transmitted safely over a Wi-Fi network [13]. It will also make sure that unauthorized personnel cannot access that Wi-Fi network and provide a safe connection. Since our safe is using Wi-Fi communication, using this standard is important to ensure that the communication over wireless networks is safe and protected. This will affect our design because it requires us to choose Wi-Fi modules that comply and adhere to this standard.

4.3.2 RFC 5321: Simple Mail Transfer Protocol

This standard defines the procedures and requirements for SMTP. It is responsible for being able to send and receive email notifications between email servers. The protocol has been updated since 1982 to the evolving email standards of today. Its main goal is to ensure that email messages are being sent reliably and consistently between server and client. For example, it checks that a proper connection is established, the content of the email is accurately sent and handles any email failures [124].

SMTP uses basic commands that are used to communicate between the server and client which is why it's known as a text-based protocol. In addition, it uses relays to transfer emails between servers. It includes additional security extensions like TLS and SPF to make sure the data in the emails are encrypted when transferring them. This standard is important for our project because it will help make sure that the email notifications are safely and properly sent to the user.

4.4 Website Standards

The standards for the website are to represent the basic functions and setup of a website. For technical standards, we plan to include a HTML page for the basic structure and a CSS file for styling. These files include the bare minimum for the website to run and function as intended, but other file formats will be included, if necessary, but for basic use, HTML and CSS will suffice. Another standard that we wanted to have been mobile responsiveness, since the ESP32 can work via mobile use as well. We will also include performance

optimizations through good code practices to speed up loading times. The last technical standard would be to have cross-platform compatibility across web browsers, so regardless of what browser the user is on, they will have access to the safe. As for accessibility standards, we plan on using HTML semantics to make the code easily readable and help the user navigate the website effectively. For content standards on the website, we will provide accurate and up-to-date information on the website to avoid misleading users, while also being transparent with the device's purpose and mission. Standards for the users' experience include clear navigation and design, consistency, and readable layouts for fonts and text. With a clear navigation and design, the website will be easily navigable, and all the content will be easy to find. With consistency, the visual layout will be appealing to the user, and it will help them feel comfortable using it. Readable layouts with fonts and line spacing will also make the user feel more comfortable since it makes the website look clean and professional. For legal standards, we will adhere to all copyright and trademark laws when using anything that isn't ours. For social standards, any design used will be to portray the project theme and will have a diverse audience in mind, any offensive or inappropriate images or text will not be used.

4.5 Ethical Standards

The ethical standards of our safe are meant to help clarify the owners' intentions with the safe and privacy concerns. For owners using the safe, owners have an ethical obligation to use them responsibly by not storing illegal goods or goods obtained unlawfully. Owners must comply with legal rules concerning storage of items such as firearms, drugs, or certain sensitive documents. Owners must know the laws to prevent misuse of the safe. It is also the owners' responsibility to place the safe in a location that won't bring potential danger to other people. In terms of privacy ethics, we respect the privacy of the owner and their information. Ethical behavior would be to avoid any unauthorized access to the owners' information and to keep firm confidentiality. Also, for any person owning a safe, we would reveal any data information that could lead to backdoor scenarios or potential vulnerabilities with the owners' data.

4.6 PCB Standards

IPC PCB standards deal with the design, assembly and manufacturing of a PCB to ensure that it is of quality. The benefits of abiding by these standards include reduced costs, improved communication, consistent quality & reliability. The Institute of Printed Circuits (IPC) organizes electronics products into three different classes that correspond to the PCB's quality: Class 1, Class 2 and Class 3. Class 1 products refer to consumer electronics and everyday products that only require the PCB to function as intended. Class 2 products include laptops, PCs and microwaves where high reliability and an extended life are top priority. Finally, Class 3 products include critical systems such as life support systems where consistent performance & reliability are of utmost importance [143]. Based on these classifications, our digital safe would fall under Class 2 as it is not what you would consider a critical system or an everyday electronic device. The specific standard that we will focus on for PCB design is the IPC-2221.

4.6.1 IPC-2221

The IPC-2221 standard is the foundation of general PCB design principles and recommendations that can apply to various kinds of boards. Other IPC standards such as IPC-2222, IPC-2223, IPC-2225 and IPC-2226 all fall under the IPC-2221 umbrella. The standard that is most relevant to our design is the IPC-2222 standard because it covers design requirements for rigid printed circuit boards. The other IPC standards are only relevant to flexible boards, PC cards, multichip modules and HDI boards respectively [144]. IPC-2222 sets standards for important PCB design factors such as material selection, board thickness, routing parameters, component placement and hole/interconnection requirements [145].

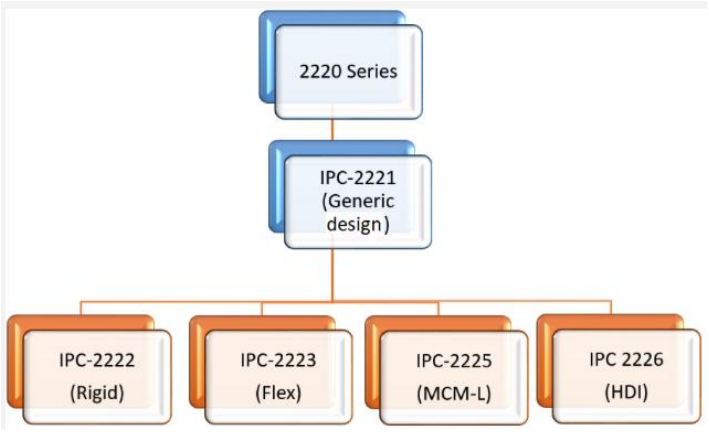


Figure 14: IPC-2221 Structure [144]

4.7 Electricity Standards

There are many different standards for electricity but the most important standard we picked is the IEC 62368-1. IEC 62368-1 is an international safety standard that specifies electrical equipment involving audio, video, information and communication technology, with rated voltage not exceeding 600V. Components like the fingerprint sensor, display, and solenoid lock would need to comply with the safety requirements outlined in IEC 62368-1. This would ensure the safe is designed to protect users from potential hazards related with electrical faults, energy sources, and fire risks. The potential hazard associated with fire risk is extremely important since the housing of the safe will be made from birch plywood, which is relatively flammable. The standard provides guidance on how to effectively apply safeguards in product design and construction. Using a relay and transistor in our circuit provides insulation and protection to help prevent electrical shock and short circuits especially when controlling a solenoid. [146]

4.8 Time Constraint

One of the most important constraints for our Smart Safe is the amount of time we have to complete this project. Since this is a Senior Design project, we only have one academic year to complete this project in its entirety. The first half of the year is mostly dedicated to the completion of the report and setup of our project website. In addition, there's seven quizzes on the lecture material that we must prepare for and complete by a certain date. That is not even mentioning the fact that everyone in our group is taking a full course load along with this class and losing almost two weeks of school due to Hurricanes Helen & Milton. There's also the matter of ordering all our parts and the possible delivery delays we may face when waiting for them to come in. Possible examples of this may include a key component of our project being out of stock or parts coming in later than they were supposed to. The second half of the academic year is fully dedicated to the design, integration, testing and debugging of the various components involved in our Smart Safe project. It is possible that we will be met with various issues whether it be issues with our custom PCB or fixing code issues with our peripherals.

Despite all the existing and potential time constraints that work against the development of our Smart Safe, we have developed ways to counteract or at the very least, mitigate them. First, we did most of our brainstorming and group recruitment in the summer so by the time the fall semester started we had a fully formed team and project idea, which is something most groups didn't have in the beginning. Our decision to be proactive would soon pay off as while other students were still in the process of forming groups and determining their project ideas, our group spent the first two weeks of the semester meeting on Zoom to finalize the scope of our project. Once this was complete, we immediately started working on the rough draft of our Divide and Conquer document and scheduled our advisor meeting with Dr. Wei to be the Tuesday after the due date of the D&C rough draft. We have also had group meetings in person to work on the report and discuss ideas with each other to make sure that everyone is on the same page. Soon after we received our advisor comments on the rough draft, we began working on the final draft while also doing research that would later be included in the rough draft of our 60-page document.

4.9 Size Constraint

Another design constraint of our project is its overall size. As previously stated, this safe is intended for the storage of valuables such as jewelry, cash, passports, personal documents & firearms. The goal is for our safe to be both large enough to contain these items but not unreasonably big. It would be poor design if we make our digital safe too small to fit a variety of personal items. Furthermore, we want our digital safe to be light enough for a user to carry if they want to move it from one location to another. Another aspect of the size constraint is making sure that the components do not take up too much of the safe's storage space.

4.10 Budget Constraint

One of the design constraints of our project is the estimated budget of our project. Since we are not a sponsored project, we will pay for project materials ourselves. Based on our estimated Bill of Materials the total cost is around $191 dollars. We have decided to split

the cost evenly between all team members, but it still means managing the costs of the items we purchase to keep the total price under $200. This means we need to prioritize choosing products that are cost effective and meet our budget requirements. One example is we chose birch plywood instead of metal because it is more expensive and harder to work with. We need to carefully choose the quantity of each product and consider how products sold in bulk or multiples will affect our budget. Our current estimated budget meets our price range, but this may change in the future.

4.11 Environmental Factors Constraint

One important design constraint to consider for our project is environmental factors. This includes temperature sensitivity and outside conditions. We need to consider how the safe should be stored in a safe location and preferably where the temperature is stable and within operating conditions. The average operating temperature is around 40 to 85 degrees Celsius [125]. If the temperature is very high and exceeds this limit it could cause the safe to overheat and potentially affect its performance. Components like the Solenoid Lock may be temperature sensitive and could be affected by extreme temperatures. One thing to consider is choosing components that can withstand certain higher temperature ranges making the safe stronger and less prone to environmental circumstances. One of our stretch goals is to include a temperature sensor that can track the temperature of the safe, informing the user if it is out of range. This could help protect the safe by informing the owner of any dangerous conditions that could cause it to malfunction. Another environmental factor to consider is water or liquid that might harm any components or interfere with its functionality. It would be helpful to choose products that are waterproof. Another critical problem could be a power outage or main power supply failure [125]. For example, if there is a power outage or blackout where the owner lives this could affect the safe. To solve this problem, we plan to implement a battery backup system that will transfer the safe to battery mode in case the power is not working. This will make sure that the safe is still fully functioning in case of these types of situations.

4.12 Material Constraints

Another design constraint of our project is the choice of materials used to build the housing for the safe. As mentioned earlier, this safe is designed to store valuable items like emergency cash, personal documents, or firearms, making security a top priority. We considered three materials: plastic, wood, and metal.

Metal is the most secure option, but it presents challenges such as high cost and the need for specialized tools for cutting and assembly. Plastic is the second option; white it requires access to a 3D printer or a plastic slab along with tools to cut it, it is the least secure material. Lastly, wood offers a balance of security and accessibility all within our budget. Although it also requires similar tools as plastic for cutting, wood is generally more secure, and durable than plastic and easier to obtain and implement than metal. Additionally, wood can offer a more aesthetically pleasing finish, enhancing the user experience. Since we are using wood only for the protype, we chose birth plywood to build our smart safe.

4.13 Power Constraints

An additional design constraint for our project is power consumption, given the variety of components involved. Our power constraints are relatively minimal since the circuit is powered by a wall outlet and a backup battery system in place. However, we must continually monitor the battery status since all components use a lot of continuous power. In the event of a power outage or if the batteries run empty, the solenoid will unlock, disabling the motion sensor, and preventing the ESP32 from sending notifications. This scenario would compromise the safety features of the smart safe.

Chapter 5: Comparison of ChatGPT with other Similar Platforms

We have ChatGPT as an AI platform tool used for aiding us in research and guidance for our project. ChatGPT has been a key tool for all members in our team. Some of the benefits of ChatGPT is that it provides information on topics that might be hard to find, and complex ideas that might be difficult to understand. It is also helpful for brainstorming and providing useful ideas. Our team felt that ChatGPT helped provide guidance when choosing what products or technologies to compare and which ones would be better for our project. It was also beneficial for documentation and proof-reading our writing.

Some of the disadvantages of ChatGPT is that it can easily depend heavily on this platform for information and ideas instead of using other resources equally. In addition, it may not be able to provide real-time information and the latest updates. ChatGPT may not be 100% accurate and can sometimes provide outdated information which is why it is important to cross reference with more reliable sources. Overall, ChatGPT was a key contributor to our paper by assisting us in research, giving insight into choosing the right components, and helping us in drafting our writing.

Below we will be providing 3-4 examples to show how ChatGPT has benefited our learning experience in Senior Design:

Case Study 1: Choosing the Email Protocol: what are 3 different email protocols I could use to discuss for my paper?

Three common email protocols you can discuss for your paper are:

1. Simple Mail Transfer Protocol (SMTP): Primarily used for sending emails. It handles outgoing mail and is responsible for the communication between servers to transfer messages across networks. SMTP is widely used alongside IMAP and POP for complete email functionality.

2. Post Office Protocol 3 (POP3): Used for retrieving emails from a mail server. POP3 downloads emails to the local device and often deletes them from the server afterward, making it useful when you want to access emails offline, but less suitable for accessing mail across multiple devices.

3. Internet Message Access Protocol (IMAP): Allows users to access and manage their email directly on the mail server without downloading it. IMAP is ideal for users who need to access their emails from multiple devices, as messages remain on the server until explicitly deleted.

These protocols serve different functions in the email communication process and offer different advantages, depending on the user's needs and setup [147].

Using ChatGPT helped us understand 3 different types of Email Protocols we could discuss for our project. It summarized the 3 different protocols and the purpose of each of them. This will help us decide which ones are better for our project and a summary that would provide information we could use when comparing them. One limitation is that it doesn't provide specific examples and how it works exactly which is something we would need to prompt or research on our own.

Case Study 2: Do you think SMTP is best for this task, or is IMAP or POP better?

For your **Smart Safe project**, where you want to **send email notifications** to users about events like failed access attempts, the best protocol to use would be **SMTP (Simple Mail Transfer Protocol)**. Here's why:

Comparison of SMTP, IMAP, and POP for Your Task:

1. **SMTP (Simple Mail Transfer Protocol):**
    - **Purpose**: SMTP is specifically designed for **sending emails** from one server to another or from a client to a server.
    - **Best Use Case**: Sending notifications, alerts, and any outbound communication.
    - **Why it's best for your project**: Since the Smart Safe needs to **send email notifications** to users (like access alerts, lock/unlock status), SMTP is the protocol designed for sending emails to recipients. It directly supports the task of transmitting outgoing messages.
    - **Functionality**: Once the event (e.g., failed access) occurs, your system composes an email and uses SMTP to deliver it to the user's email inbox.
    - **Security**: SMTP supports secure communication via **TLS/SSL** to ensure your data is encrypted.

2. **Conclusion**: SMTP is designed to **send** emails, so it's the most appropriate for this task.

3. **IMAP (Internet Message Access Protocol):**
    - **Purpose**: IMAP is for **receiving and managing emails** on a mail server.
    - **Best Use Case**: Reading, organizing, and synchronizing emails across multiple devices.
    - **Why it's not ideal for your project**: IMAP is focused on **retrieving emails** from an inbox, which isn't the primary function your Smart Safe needs. You don't need to retrieve emails from a server; you just need to send them.

4. **Conclusion**: IMAP is not suited for sending notifications, so it's unnecessary for your current requirements.

5. **POP (Post Office Protocol):**
    - **Purpose**: POP is also for **retrieving emails** from a mail server, but it usually downloads the messages and removes them from the server.

    - **Best Use Case**: Retrieving emails in environments where you don't need to access emails from multiple devices or keep emails stored on the server.
    - **Why it's not ideal for your project**: Similar to IMAP, POP is about **receiving** emails, which isn't your goal.

6. **Conclusion**: POP is not relevant to your project because you're focused on **sending notifications**.

Summary:

- **SMTP** is the right protocol for your task, as it's specifically designed for **sending emails** (like notifications and alerts) from your Smart Safe to the user's email inbox.
- **IMAP** and **POP** are for retrieving emails, which is not required in this context, as your system doesn't need to access or manage an inbox—just send outbound emails.

So, for sending out alerts and notifications from your safe, **SMTP** is the best choice [147].

This example of using ChatGPT to determine which Email Protocol can be used for email notifications in our project shows how ChatGPT can provide insightful information including the purpose, Best Use Case, and Features of each protocol. It explains that IMAP and POP are used for retrieving emails and not for sending emails. It also provides a summary that explains why SMTP protocol is best designed for our project. One limitation is that it does not provide any examples or additional information about how it works or how it will be implemented into our project which requires additional research on our part. Even though ChatGPT gives us the best option, we should also consider additional outside factors that could impact this decision.

Case Study 3: What are all the pros and cons of using a microcontroller versus using a microprocessor for an embedded system?

Here's a comprehensive list of the pros and cons of using a microcontroller versus a microprocessor in embedded systems:

Microcontroller

Pros:

1. **Integration:**
    - Combines CPU, memory (RAM and flash), and peripherals (timers, ADCs, I/O ports) in one chip, reducing board space and component count.

2. **Cost-Effective:**
    - Generally lower cost due to reduced components and manufacturing complexity, making them suitable for budget-sensitive projects.

3. **Low Power Consumption:**

- Designed for energy efficiency, ideal for battery-powered and low-power applications.

4. **Ease of Use:**
   - Simpler architecture and often designed for specific tasks, making programming and implementation straightforward.

5. **Real-Time Performance:**
   - Provides predictable timing and interrupts, which is critical for real-time applications.

6. **Development Tools:**
   - Many have user-friendly development environments and extensive community support, facilitating easier prototyping and debugging.

7. **Robustness:**
   - Typically, more resilient in environments with noise or electrical disturbances, which is beneficial for industrial applications.

Cons:

1. **Limited Processing Power:**
   - Generally lower clock speeds and performance capabilities compared to microprocessors, which can limit their use in complex applications.

2. **Memory Constraints:**
   - Limited RAM and flash memory may restrict the size and complexity of applications.

3. **Less Flexibility:**
   - Not as adaptable for varying application needs compared to more modular systems like those using microprocessors.

4. **Peripheral Limitations:**
   - Integrated peripherals may not meet all application requirements, leading to additional external components.

5. **Development Limitations:**
   - Less suited for high-level operating systems or multitasking environments.

Microprocessor

Pros:

1. **High Performance:**
   - Generally, offers superior processing power, suitable for demanding applications and complex calculations.

2. **Scalability:**
    - Can support larger and more complex software systems, making them ideal for advanced applications.

3. **Flexible Memory Options:**
    - Supports extensive external memory configurations, allowing for larger applications and data processing.

4. **Advanced Operating Systems:**
    - Capable of running sophisticated operating systems (like Linux or Windows), facilitating rich user interfaces and multitasking.

5. **Modular Architecture:**
    - More modular design allows for the addition of various components as needed, providing greater flexibility.

6. **High Clock Speeds:**
    - Generally, operates at higher clock frequencies, improving performance for demanding tasks.

Cons:

1. **Higher Cost:**
    - More expensive due to complexity, additional components, and manufacturing costs.

2. **Increased Power Consumption:**
    - Typically consumes more power, which may not be suitable for portable or battery-operated applications.

3. **Larger Physical Size:**
    - Requires more board space and additional components, complicating design and increasing footprint.

4. **Complex Design:**
    - More complex architecture can lead to longer development times and a steeper learning curve for developers.

5. **Peripheral Requirements:**
    - Requires external components for basic functions, increasing the overall system complexity and cost.

Choosing between a microcontroller and a microprocessor depends on the specific requirements of the application. Microcontrollers are ideal for low-power, cost-sensitive,

and simpler applications, while microprocessors are better suited for high-performance, complex tasks requiring flexibility and advanced functionalities [147].

This example of using ChatGPT reaffirmed our decision to choose a microcontroller for our project instead of a microprocessor. Since our Smart Safe will be battery operated, having low power consumption is an important thing to shoot for. In addition, our safe doesn't require the use of an operating system nor does it need to have an incredibly high clock speed to carry out its various functions.

Case Study 4: What are 3 types of relays should I use to discuss for my project report?

For your project report, consider discussing the following three types of relays:

1. **Electromechanical Relays (EMR):** These relays use an electromagnetic coil to mechanically operate a switch. They are commonly used for switching applications in power systems due to their robustness and simplicity.

2. **Solid State Relays (SSR):** SSRs use semiconductor devices to perform the switching action without moving parts. They are known for their fast-switching times, high reliability, and ability to handle high voltages and currents, making them suitable for various applications, including industrial automation.

3. **Reed Relays:** These relays consist of a pair of metal reeds sealed in a glass tube. When an electromagnetic field is applied, the reeds come into contact, allowing current to flow. Reed relays are valued for their compact size, low power consumption, and fast response times, often used in low-voltage applications.

Using ChatGPT helped us explore three types of relays relevant to our project. It provided a summary of their underlying technology, key characteristics, and suitable. The benefits of using ChatGPT is the speed, accessibility, and the amount of detail it can provide. Even if the response is short, you can request for further elaboration, and it's available whenever you need it.

However, the challenges of using ChatGPT is that the information provided is not always accurate, it doesn't cite sources, and there are privacy concerns. The chance of having inaccurate information can be an inconvenience, as it requires additional effort to verify the answers without quick access to resources. Users may also worry about potential sharing of sensitive information since conversations may not be completely private.

Additionally, ChatGPT has limitations, such as a lack of emotional insight, difficulty understanding complex contexts, and sensitivity to how questions are phrased. The lack of emotion and not being able to understand complex contexts can provide short or inaccurate responses. While, how you phrase a question can also impact the answers in many different ways [147].

Chapter 6: Hardware Design

In this chapter, we will elaborate on what our hardware design and how our chosen components communicate with each other. This will include both a high-level description of our design and a schematic level description of how each component is connected.

6.1 Subsystem Block Diagram

The figure below is a block diagram of our overall hardware design. It details how all of our components will be connected to each other on the PCB.

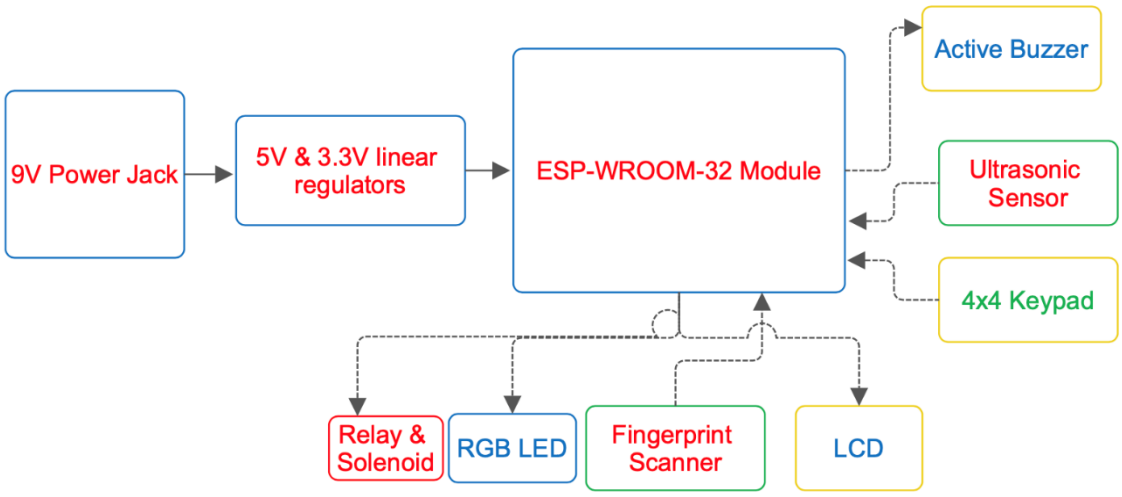


Figure 15: Subsystem Block Diagram

6.2 Schematic Diagram

In the following sections, we will explore how each component in our design is connected using Fusion 360 software. Finally, we will demonstrate how all the components are integrated to work together as a cohesive system.

6.2.1 ESP-WROOM-32 Schematic

To start off, the ESP-WROOM-32 module will handle all communication and power delivery to our peripherals with its many general purpose input/output(GPIO) pins. The module's 3.3V & 5V pins receive power from the 5V regulator & 3.3V regulator respectively. We integrated an enable button and a bootloader button to have the ability to reset the device or flash new firmware to it. In addition, we incorporated a built-in LED connected to GPIO pin 2 so that we can program it to blink on and off. This would serve as confirmation that we were able to incorporate the module into our design successfully.

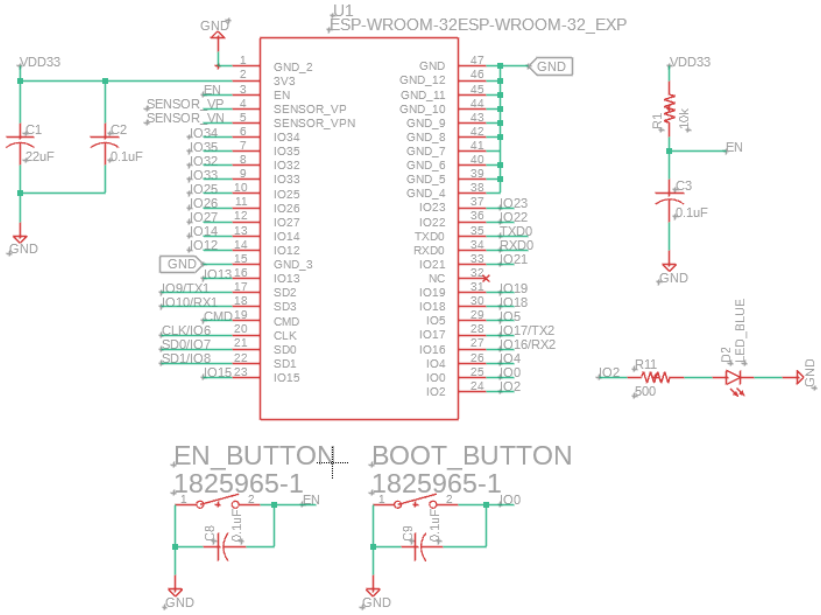


Figure 16: Wiring Diagram for the ESP-WROOM-32

6.2.2 Keypad Schematic

The 4x4 matrix keypad has 16 buttons but only uses 8 total GPIO pins from the microcontroller. The buttons are organized into a matrix with 4 columns and 4 rows. Each column is connected to one GPIO pin and each row. Is connected to another GPIO pin. The keypad is powered by the 3.3V signal from the ESP-WROOM-32 module, as the ESP32 operates at 3.3V logic levels. When the GPIO pins used on the keypad are set to high or 1 in logic level, the pin will output a voltage close to 3.3V. The columns are typically set as inputs with internal pull-ups enabled, and pressing a button in the matrix connects a row and column, allowing the microcontroller to detect the key press. The contact resistance between the button switches and the matrix wiring is typically less than 100 ohms, which is within the range that the ESP32 GPIO pins can easily handle.

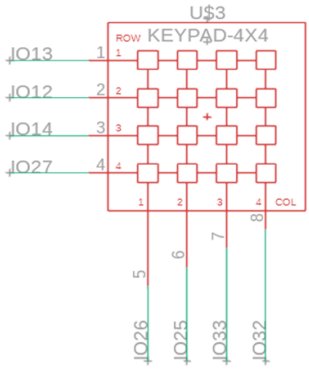


Figure 17: Wiring Diagram for the Keypad

6.2.3 Fingerprint Scanner Schematic

The fingerprint scanner module has a total of 6 pins but we only use 4 total GPIO pins from the microcontroller. Pin 1, labeled 5V, on the fingerprint scanner is connected to Vin pin on the ESP-WROOM-32 to receive 5V. Pin 2, labeled GND, is connected to the ground pin next to Vin on the ESP32. Pin 3, labeled TXD, is connected to GPIO 16, which is configured as the RXD pin. Pin 4, labeled RXD, is connected to GPIO 17 on the ESP32, which is configured as the TXD pin. Pin 1 and pin 2 is used for power connection while pin 3 and pin 4 is for data communication while the other two pins are not necessary. Pin 6 labeled 3.3V, is used when you want to power the fingerprint 3.3V instead of 5V. Pin 5, labeled TOUCH, is not used since we don't need an active low output when during fingerprint scanning [57].

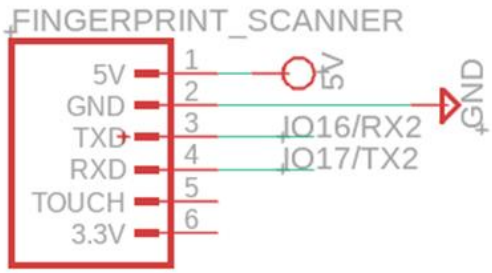


Figure 18: Wiring Diagram for the Fingerprint Scanner

6.2.4 Locking Mechanism Schematic

The locking mechanism in the schematic consists of five total components. A solenoid with 2 terminals, a relay with six pins even though it only shows 5 in the footprint, a transistor with 3 pins, a diode, and resistor both with 2 pins. The push solenoid has its negative terminal wired to ground and the positive terminal connected to the NO (normally open) pin of the relay. The SPDT relay has 9V connected to the COM (common) pin, the solenoid's positive terminal in the NO pin, 5V connected to both the signal and positive pin, and ground to the negative pin. Since the current and voltage required to energize the relay col is typically too high for the ESP32 to control directly, the transistor acts as the relay's driver. The emitter of the transistor is connected to ground, the base is connected to a 330 ohm resistor and GPIO 23 while the collector is connected to the negative pin of the relay. The rectifier diode is used to protect the transistor from voltage spies. The cathode of the diode is connected 5V and the relay's signal pin, while the anode is connected to the negative pin of the relay.

The solenoid is controlled by the relay like a switch and the transistor is the driver. The relay switches from NO position to NC (normally closed) position when the ESP32 signals the transistor to switch the relay, using low-voltage signals. The ESP32 sends 3.3V to energize the base of the transistor through a 330 ohm resistor. This low voltage signal turns

the transistor on, allowing current to flow from the collector to the emitter, energizing the relay coil. Once the relay is activated by the 5V provided by the voltage regulator, the armature moves from the NO position to the NC position, which allows 9V to flow through the solenoid, activating it and unlocking the door.

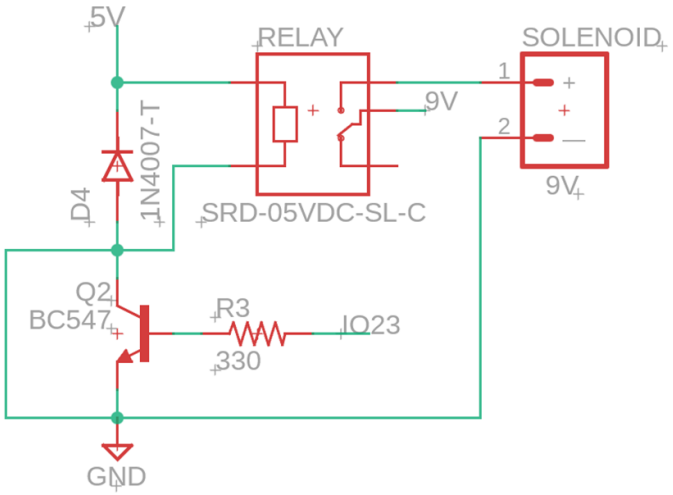


Figure 19: Wiring Diagram for the Locking Mechanism

6.2.5 Ultrasonic Sensor Schematic

The Ultrasonic Sensor uses a total of 3 GPIO pins and one power pin off the ESP-WROOM-32 module. The sensor's Vcc pin is connected to the microcontroller's 5V pin, the trigger pin is connected to GPIO pin 22, the echo pin is connected to GPIO pin 23 with the sensor's final pin being connected to ground.

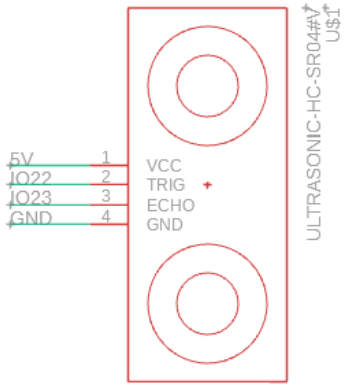


Figure 20: Wiring Diagram for the Ultrasonic Sensor

6.2.4 Display Schematic

The LCD display has a total of 16 pins but we only used 4 GPIO pins from the microcontroller when utilizing the I2C interface. The display typically uses the standard 16 pin configuration for parallel communication. However, the I2C interface integrated into the display, we only need two data pins for communication and two pins for power. The backlight control is built into the display and can be adjusted via code. The contrast of the display can be controlled using a potentiometer located on the back of the module, which adjusts the voltage to the VO pin. This adjustment affects the contrast of the screen by controlling the alignment of the liquid crystals in the LCD panel. For power connection, pin 1 labeled GND is wired to GND on the ESP32 while pin 2 labeled VCC is connected to Vin for 5V. For data connections, pin 3 labeled as SDA is connected to GPIO 2 while pin 4 which is SCL is connected to GPIO 4.

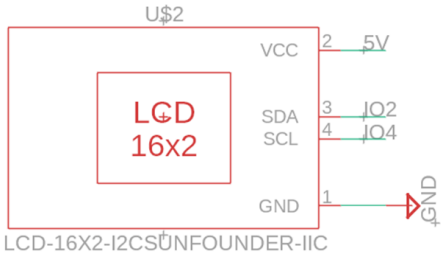


Figure 21: Wiring Diagram for the Display

6.2.5 Buzzer Schematic

The magnetic buzzer only has two pins, the power and ground pin. The power pin can be connected to any GPIO pin on the ESP-WROOM-32 configured to output 3.3V when set to high or 1 in logic level. The ground pin is set to one of the ground pins on the microcontroller. Inside the buzzer housing, there is a built-in coil, transistor, and diode. The NPN transistor acts as a switch to increase current to the buzzer's coil using a 1k resistor connected to the transistor's base. The diode is placed across the buzzer's coil to protect against voltage spikes caused by the inductive properties of the coil when the current is quickly switched off.

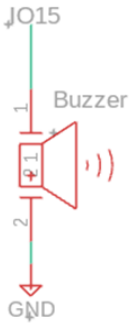


Figure 22: Wiring Diagram for the Buzzer

6.2.6 LED Schematic

The RGB LED module uses a total of three GPIO pins(one for each color channel) and a GND pin for its cathode. The red channel is connected to GPIO pin 2 on the ESP-WROOM-32 module, the green channel is connected to GPIO pin 4 on the ESP-WROOM-32 module and the blue channel is connected to GPIO pin 5 ESP-WROOM-32 module.

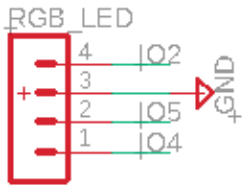


Figure 23: Wiring Diagram for the Buzzer

6.2.7 Voltage Regulators

The 9V from the battery serves as the input for our 5V regulator with bypass capacitors with a value of .22uFat connected to the input and .01uF connected to the 5V output. That 5V output then serves as the input for our 3.3V regulator with bypass capacitors that are both 22uF at the input and output. Furthermore, a red LED is connected to the regulator's input as it will serve as a visual indicator that it is receiving power from the 5V regulator. The goal of these regulators is to supply power to our components that need either 5V or 3.3V to operate.

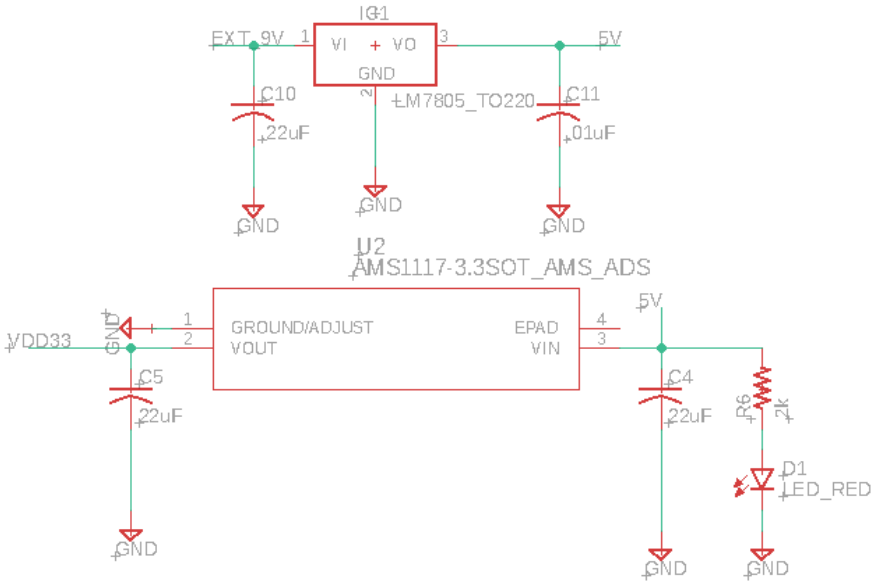


Figure 24: Wiring Diagram for the 3.3V & 5V Voltage Regulators

6.2.8 DC Power Jack Connector

Our power source and backup battery system will use a barrel connector to power the circuit. In the schematic, we have used a female barrel connector that will be attached to our PCB. The female connector has a total of three connections for soldering. Pin 1, known as the power positive pole, tip terminal, or DC voltage, will receive 9V from the male barrel plug. Pin 2, known as the negative moving contact, the sleeve terminal, or ground, will be connected to the ground bus. Pin 3, known as the negative static contact, the shunt terminal, or plug detect, will also be connected to ground. When a plug is inserted, the sleeve terminal acts like a switch and opens the circuit. When no plug is inserted, the sleeve terminal is shorted to ground.

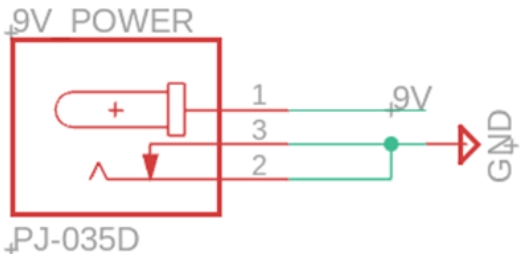


Figure 25: Wiring Diagram for the DC Power Jack

6.3 Overall Schematic

Smart Safe will be powered by using a barrel connector, providing a 9V power supply to the system. The main power source will be from a wall outlet using a power supply adapter that's also attached to an uninterruptible power supply, our battery back system. A 5V voltage regulator will step down the power to supply 5V to the fingerprint sensor, display, ultrasonic sensor, relay, and ESP-Wroom-32 module, with the GPIO Vin pin also outputting 5V to support connected peripherals. A 3.3V voltage regulator will step down the power to supply 3.3V to the ESP32, allowing the GPIO 3V3 pin to supply 3.3V to components like the ESP32 itself and certain components. The keypad, buzzer, and LED will be powered directly through the GPIO pins, which also output 3.3V. The ESP-Wroom-32 module includes an EN button to reset the ESP32 and a BOOT button for firmware uploads. Additionally, the ESP32 features a blue LED to indicate the microcontroller's operational status and a red LED to signal power status. UART will be used for communication between the ESP32 and other connected devices, ensuring reliable data transmission. For added protection and stability, the system is designed to safeguard against potential power surges and maintain efficient energy distribution across all components.

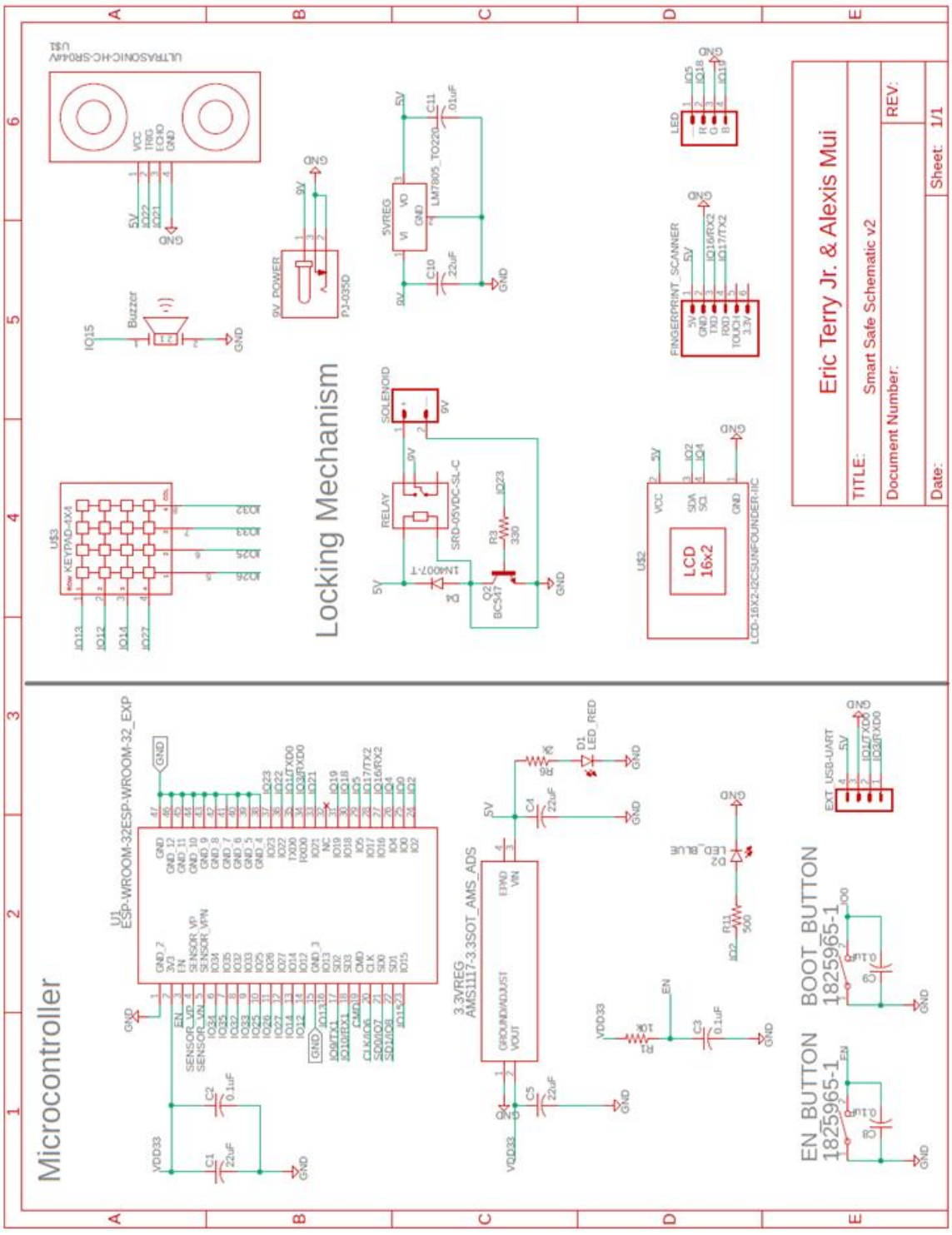


Figure 26: Complete Schematic Design for the Smart Safe

Chapter 7: Software Design

In Chapter 7, we will be discussing software design. We created a UML class diagram that shows how the system is structured on a software level. We created a UML case diagram which shows how the user interacts with the system. We will discuss the user interface for the email, website, and display components. Lastly, we will explain how data is transferred in our project.

7.1 UML Class Diagram

The class diagram is a visual representation of software design which uses UML to show the structure of a system including classes, attributes, methods and the relationship between them. Class diagrams show a high-level overview of the system's structure and play a key role in object-oriented design. First, we define the classes which are the main components of a system, in our project they are User, Safe, Log, and Notification System [148]. Next, we define the attributes which are features of a class. They are given a name like UserID and followed by a data type like String. The third box is methods which are functions that define what actions can be performed on objects in that class. Some examples of methods are lock and check fingerprint followed by () which represents a function that will be implemented in the code. We then need to define the relationships between each class. The solid black lines are class simple association which means that both classes are related to each other but not dependent [149]. In the diagram, we see that User is related to Log and Notification System. The black line with a diamond is aggregation which means that one class is composed of another class [149]. In our diagram we see that the Safe depends on the notification system. The dotted black lines with an arrow represent dependency which means that one class relies on another in some way [149]. In our diagram, we see that the safe relies on the user and vice versa.

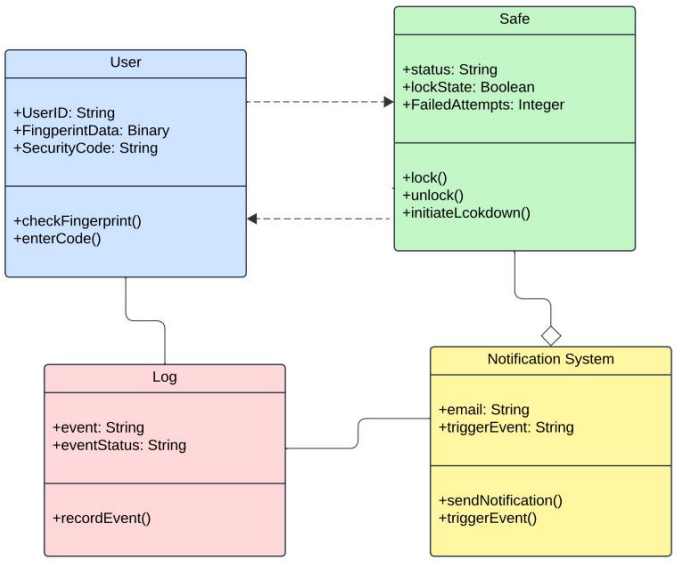


Figure 27: UML Class Diagram [148] [149] [150]

7.2 Use Case Diagram

The use case diagram below shows how the user will interact with the system. This will help show how the user interacts with the safe from their perspective and at a higher level [151]. The diagram has two actors which are the user and the administrator. The administrator represents us and shows how we interact with the safe and control its actions.

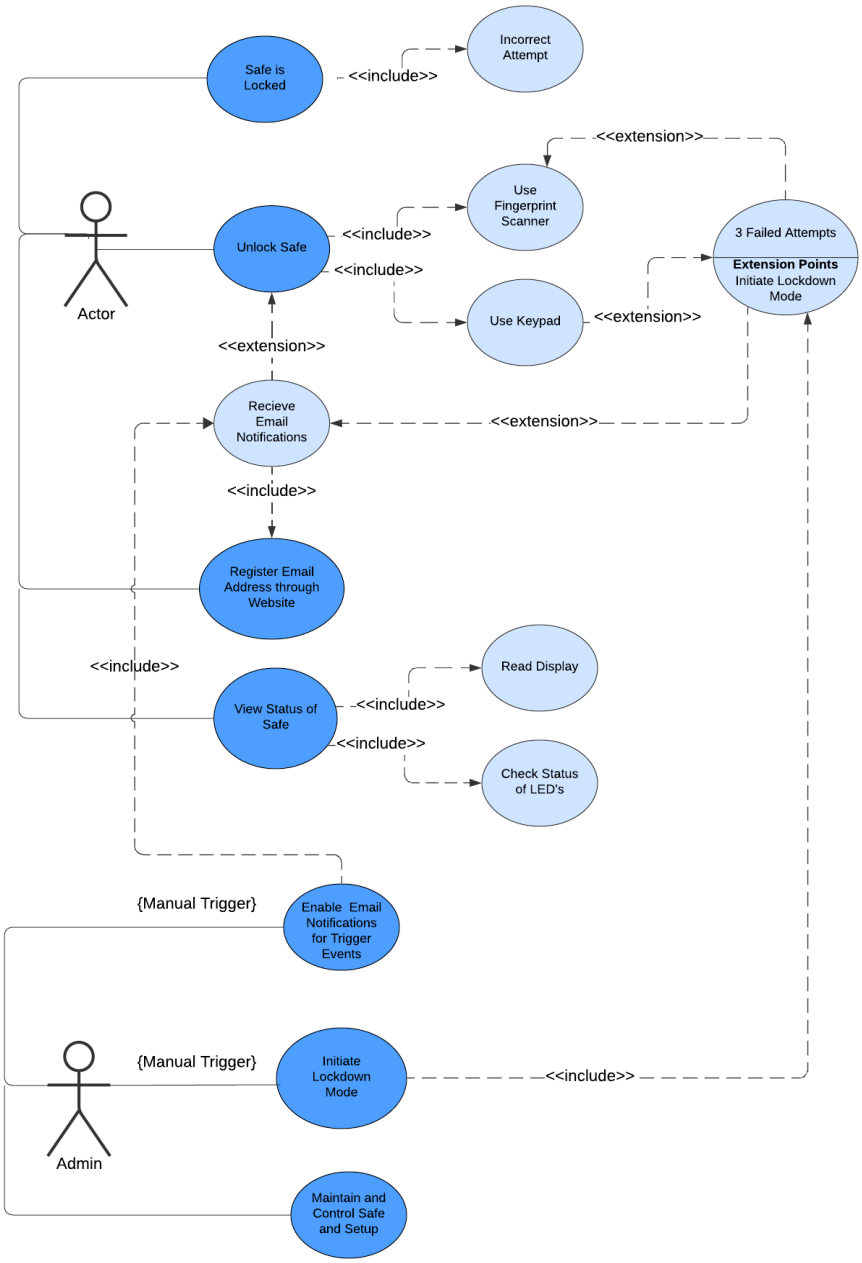


Figure 28: UML Class Diagram [148] [150] [151]

7.3 Website Application Design and UI

Our website application will allow users to enter their email address which will be transferred to the ESP32 to register their email address to send notifications about the safe. We want the user interface for the website to be easy, clear, and user-friendly. We are going for a minimalistic design with a professional style [148]. We plan to also make sure the website is accessible via a mobile device [148]. We also plan to provide a quick overview or user guide that can be viewed on the website and provide our contact information [148]. One of our stretch goals is to create a login for the website and a forgot password feature along with email verification making the website more accessible and secure. In order to get access to the website the user will receive a link to the website when they purchase the safe.

The user will receive the website name for our safe so that they can access this feature. When the User enters the website, they will simply enter their email address in a display entry box and will receive a thank you message after they have submitted it. The user can then confirm that they have registered their email address to receive notifications. Below is a prototype of how we want the website Interface and Design to look like.

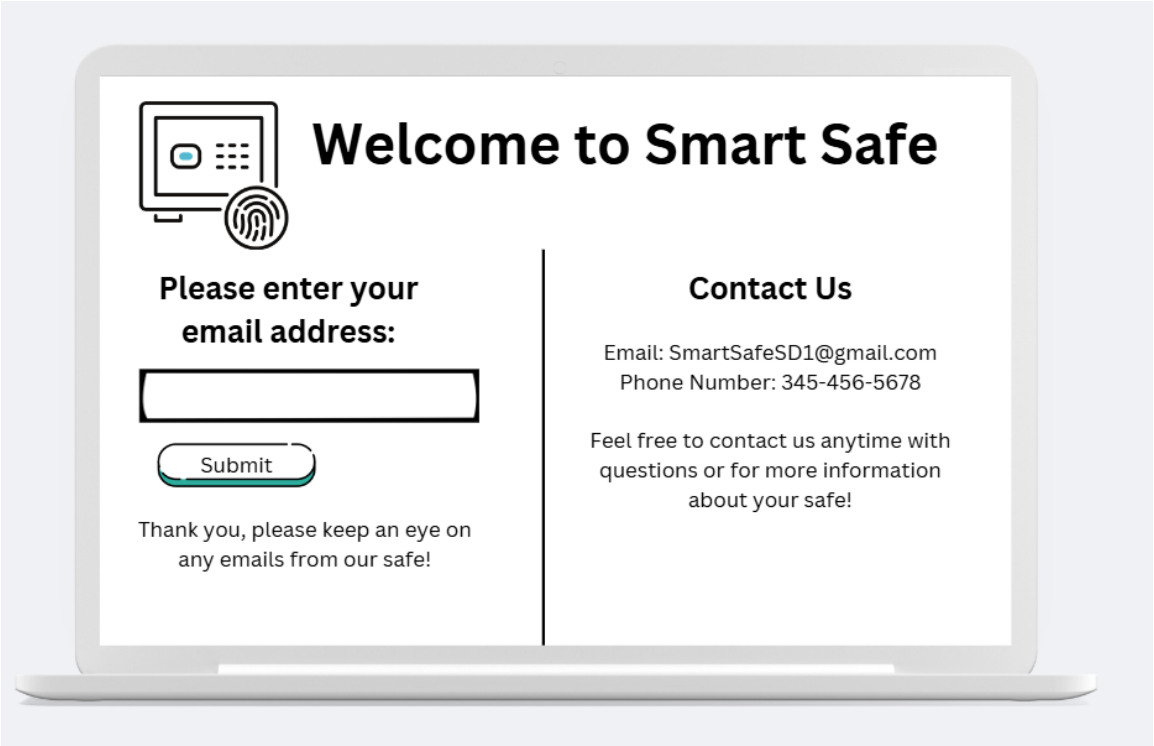


Figure 29: UI Website Design/Prototype [161]

7.4 User interface for email

The user will enter their email address on the website used for the safe which will be transferred to the ESP32 which will automatically send an email regarding the safe to their Gmail account. The user will receive an email notification if the safe has been accessed, motion is detected near the safe, or the safe has entered lockdown mode. The user will receive an email to their Gmail account informing them that someone has accessed or seen near their safe and to confirm that this was an authorized user. The email will also provide a timestamp which will inform the owner of when this action was done. The email will be sent from an email account created for the safe called smartsafe123@gmail.com which will be provided under our contact info on the website. The email should be sent in under 2 minutes but could be affected by unseen outside factors like the internet, service provider, or surroundings.

We will show how the email will look for each Trigger Event:

Case 1: Safe is Accessed [148] is shown in Figure 30.

Case 2: Motion is detected near the safe by the Motion Sensor [148] shown in Figure 31.

Case 3: Lockdown Mode has been Activated [148] is shown in Figure 32.

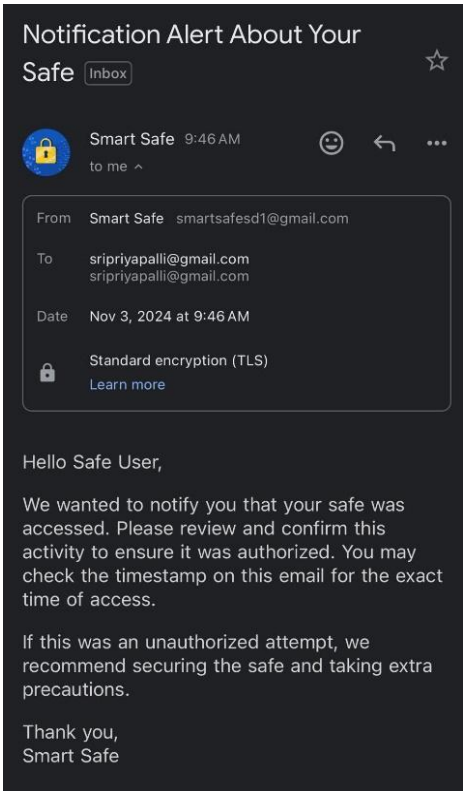


Figure 30: Email Notification Example

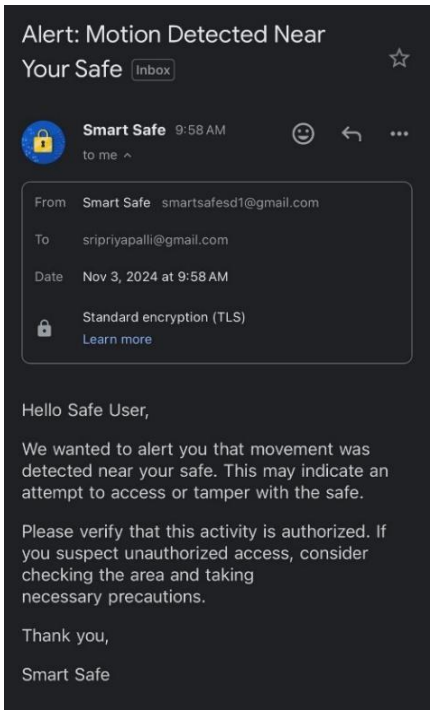


Figure 31: Motion Notification

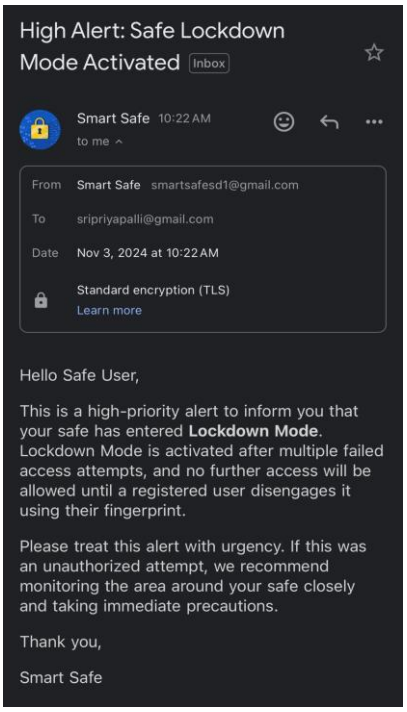


Figure 32: Lockdown Mode Notification

7.5 Display Prototypes and UI

The following images provided are examples of what we thought the display should look like at different states when the safe is being used. We wanted to make it easy to read and understand for the user because complex visuals can cause user satisfaction to not be fulfilled, and it adds difficulty to its use. The software will control and physically print each of the display statements for each required case.

LCD Display when asking for access

Figure 33: Display PIN Prototype

LCD Display when correct PIN is inputted

Figure 34: Display Unlock Prototype

LCD Display when LM Activates

Figure 35: Display LM Prototype

LCD Display after LM activates

Figure 36: Display Fingerprint Prototype

After using a valid fingerprint

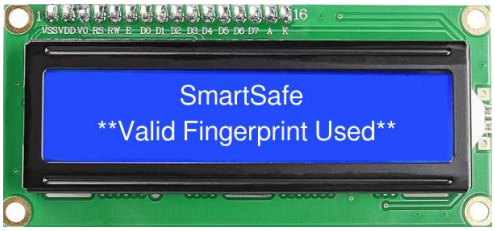


Figure 37: Display Valid Finger Prototype

After using an invalid fingerprint

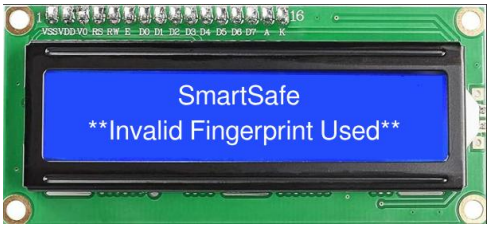


Figure 38: Display Invalid Prototype

After the user manually locks the safe or waits for the self-locking timer

Figure 39: Display Locked Prototype

7.6 Data Transfer

Data Transfer is when we move digital information from one place to another like between devices or networks. Data transfer is important in our project because it is critical for communication between the safe, website, and email system. Since the safe must send real-time notifications when the safe is accessed, data needs to be transferred whenever an event triggers an email and sends it to their inbox [160]. Another important aspect is when the website takes in information from the user where the backend will receive this data and transfer it to the ESP32 so it can send the correct information [160].

First, we will discuss how we want information from the website to be transferred directly to the microcontroller. We will create a form on our website which includes an input box that will allow them to enter their email address. That email address will then be sent to the backend server. We will then use HTTP Post requests from the website that will transfer the email address to the IP address of the microcontroller [160]. We would also need to create an HTTP endpoint on the MCU that will listen for incoming data from the website with assistance from libraries like ESPAsyncWebServer [160]. Once the email is transferred to the ESP32, we will use the ESP32 Mail Client Library explained below to send the email using the SMTP server. The email will then be sent to that User. One aspect we need to take into consideration is error handling in case the user doesn't receive the email, or the email wasn't transferred. In our stretch goals, we plan to include the user's name as an input field on the website which will be transferred as JSON along with the email from the website to the ESP32 [160]. The name field will be parsed as JSON and retrieved which will be used in the email message.

Next, we need to discuss how the microcontroller will trigger the events that will require sending an email to the User. We will need to program the ESP32 to constantly check for these events and use special conditions or sensor input [160]. We will create a loop with if statements that will check each condition and create a function that will send the email with the indicated response if the condition is true [160]. In addition, we will reset this so it can repeat this action every time. The function that will send the email will store the appropriate messages that need to be sent for each case. We will test each event to make sure the verification is functioning properly.

Chapter 8: System Fabrication/Prototype Construction

Printed circuit boards or also known as PCB are essential for electronic projects like Smart Safe. They provide a robust structure that efficiently connects and mount various electrical components. PCBs are designed to ensure reliable, high-performance operation of circuits while optimizing space and minimizing size. In this chapter, we will explore the different types of materials and layers used in PCB design, as well as the process we followed to construct and lay out our PCB design for Smart Safe.

8.1 PCB Layer Types

Our PCB will be a single-sided board, meaning it will consist of a single layer of substrate with a copper layer on one side. In this section, we will be exploring the different layers used in our PCB design and explain their importance. These layers range from those that form the core structure of the board to those that provide protection and labeling. Each layer is critical in ensuring the board's mechanical, electrical, and thermal performance, contributing to its overall functionality and reliability.

8.1.1 Substrate

A substrate is the base layer of a PCB and it plays a crucial role in determining the board's mechanical and electrical properties. It's responsible for holding the components and traces in place while providing mechanical support and durability. The type of substrate material chosen affects important factors like the PCB's dielectric constant and thermal performance. The most commonly used substrate material is FR-4, polyimide, CEM-1, and CEM-3. For high-frequency applications, Teflon can be used while aluminum is preferred for better thermal performance. The choice of substrate depends on factors like the intended application, required performance, and budget. For our project, we decided the FR-4 was the best fit for our base layer, and we will explain further in the following section on materials [152].

8.1.2 Copper-Conductive layers that form the electrical traces and pads

Copper is a primary material used in PCB manufacturing due to its electrical conductivity, which helps electricity flow between components. Copper foils are used to create the conductive layers for traces, planes, and pads on the board, basically acting as the circuit's wiring. The thickness of the copper layer can impact how much current the board can handle and its overall reliability. Choosing the right type and thickness of copper is critical for electrical performance, cost, and weight. The most common types of copper are rolled copper, electrolytic copper, and copper-clad aluminum. Thicker copper is often used for high-power applications to handle high current flow, while tinner copper is fine for low-power circuits. For our project, we decided electrolytic copper was the best fit for the outer layer, which we will cover in more detail in the next section on materials [153].

8.1.2.1 Traces

PBC traces, usually made of copper, are the conductive path that carry electrical signals and power across a printed circuit board. They connect the various components on the board, allowing them to communicate and function as a single unit. The performance of a PCB depends on the designs of its traces, including factors like width, length, thickness, and spacing. If traces are not designed correctly, it can lead to issues like signal loss, voltage drops, excessive heat, and potential malfunction or damage to the board and its components [154].

8.1.2.2 Pads

Pads are small, exposed areas of copper on a PCB that acts as connection points for components. They create an electrical connection between the components and the PCB's layout. Pads provide a designed spot for soldering, ensuring that components are securely attached and integrated into the circuit. Multiple pads form a component footprint, which defines the component's placement and connection points on the board. Pads enables the soldering process possible by providing reliable connections between the component leads and the PCB's traces, allowing electrical signals and power to flow. Without pads, there would be no designated spots for components to connect and the circuit wouldn't be able to function properly [154].

8.1.2.3 Planes

PCB planes, similar to traces but covering a larger continuous area, are conductive layers of copper on a PCB used mainly for power distribution and signal grounding. They help maintain a stable voltage supply and provide a low-resistance path for return currents. Planes reduce noise and minimize voltage fluctuations by covering a large portion of the PCB, especially for high-frequency circuits. This helps improve signal integrity and prevent issues like crosstalk or electromagnetic interference, which can degrade performance and distort signals. Planes also help with thermal management by distributing heat more evenly across the board, helping to prevent overheating of components. By providing a stable foundation, planes help maintain the reliable performance of the circuit, making them a critical element in the design of PCBs [155].

8.1.3 HASL

Hot air solder leveling, also known as HASL, is a popular protective surface finish used on PCBs to improve solderability. A board is coated with molten solder and the excess will be removed by using hot air knives, leaving a thin and even layer on the exposed copper pads. This creates a reliable surface for soldering components, helps protect the copper from oxidation, and prevent accidental short circuits. While HASL isn't ideal for fine components since its thickness can affect soldering accuracy, it has remained widely used. We picked this protective layer because it's affordable, effective, and it's still a good choice for through-hole components and larger surface mount components [156].

8.1.4 Solder Mask

Solder mask is a protective coating layer applied to the surface of a PCB to help prevent short circuits and corrosion. The board is coated with the solder mask then selectively exposed to light or chemical treatment to create openings around the pads where components will be soldered. This process makes sure that solder only sticks to the intended areas, preventing accidental connections between conductive traces that could case malfunctions. Solder masks can be made from materials like epoxy, acrylic or silicone but they're many different types like liquid photoimageable solder mask (LPISM) or dry film solder mask. Other than protecting the circuit, solder mask can also improve the PCB's resistance to environmental factors like moisture and dust, helping to ensure its long-term reliability and performance [153].

8.1.5 Silkscreen

Silkscreen is the printed markings on a PCB that identify components, test points, or other important details. These markings are usually made with epoxy ink, which is applied through stencils and cured to create permanent prints that can withstand the stresses of PCB fabrication and assembly. While silkscreen doesn't impact the performance of the board, it's essential for smooth assembly and troubleshooting. The markings help consumers quickly and accurately identify components, ensuring they're placed and oriented correctly during assembly. Silkscreen also assists in testing and troubleshooting by clearly labeling test points and other key areas on the board [152] [157].

8.2 PCB Material Type

We discussed the layers chosen for our PCB design in the previous section. In this section, we will discuss what materials we picked for the layers of our board. Since our PCB will be a single-sided design, we picked the type of material for our base and copper layer. We selected these materials because they offer the right balance of performance, cost-effectiveness, and reliability, all aligned with our project's goals.

8.2.1 FR-4

FR-4, "Flame Retardant 4," is one of the most commonly used and cost-effective substrate materials for PCBs. It consists of woven fiberglass cloth soaked in a flame-resistant epoxy resin, providing a balance of mechanical strength, electrical insulation, and thermal stability. The fiberglass gives FR-4 its strength, while the epoxy resin holds everything together and acts as an insulating layer, preventing electrical shorts between the conductive layers. It is favored for its versatility, ease of manufacturing, and compatibility with standard PCB processes, making it suitable for various applications. We chose this material for our design because it aligns with our project goals while being affordable and readily available [157].

8.2.2 Electrolytic Copper

Electrolytic copper is a highly pure form of copper that is produced through a process called electrodeposition. This process creates a copper foil with bond strength to dielectric materials, making it an ideal choice for the inner layers of multilayer PCBs. The strong bond between electrolytic copper and the dielectric helps maintain the PCB's durability and prevent delamination, which is essential for the board's reliability. However, since electrolytic copper is generally more expensive than other types of copper, it is typically used for inner layers where it is most needed. For areas where cost is a more significant concern, like the outer layer, rolled copper is generally preferred [153].

8.3 PCB Design Layout

In this section, we elaborate on the routing and layout process of our custom PCB for our project. When it came to component/part organization, we grouped each component by function and importance. First, the parts responsible for power delivery were placed as close as possible to the ESP-WROOM-32 module. We placed our 9V power jack close to the 5V regulator and to our 3.3V regulator. In addition, the decoupling capacitors for each linear regulator were also placed near their appropriate regulators so that they can filter out noise in the PCB effectively. One thing to note about our 3.3V regulator is that it has a LED connected to its input so that when it lights up, that tells us that the regulator is receiving power. All of the components responsible for power delivery have been placed as close as possible to the ESP-WROOM-32 module. The reason for this is because the output of our 5V regulator travels to the ESP-WROOM-32 module's Vin pin and the output of the 3.3V regulator travel's to the module's 3.3V pins. These pins are crucial as they are responsible for powering our various hardware components. Our fingerprint scanner and ultrasonic sensor require 5V to power it and our LCD requires 3.3V to power it. The next set of components that are close to both power delivery and the ESP-WROOM-32 module is our locking mechanism circuit. The relay is placed close to the power delivery components because it needs 5V to function. Furthermore, the solenoid that the relay is connected to needs 9V for it to retract inwards so that it will be placed near the power delivery components and the module since GPIO pin 23 of the ESP-WROOM-32 will be responsible for controlling the locking mechanism. Next, the enable and boot push buttons are placed near the ESP-WROOM-32 module since these buttons are directly connected to the module's EN pin and GPIO pin respectively. This includes their respective capacitors as well. For our components, we placed the header pins reserved for our keypad within the vicinity of the module on its left. Since that component only needs 8 GPIO pins to operate, we felt it didn't need to be as close to the ESP-WROOM-32 module as some of our other components. The header pins reserved for our fingerprint scanner were placed closer on the right of our module due to it requiring 5V to operate. Furthermore, the GPIO pins that it uses are on the right side of the module. It is for that same reason that the header pins for our LED are also on the right side of the ESP-WROOM-32 module. Our buzzer is placed on the leftmost side of our PCB, as it is only connected to GPIO pin 15 and GND. Our LCD and our ultrasonic sensor are both placed a reasonable distance away from the module even though they need 3.3V and 5V respectively to function. Our justification for this is we didn't want our PCB to be so crowded near one section, plus having these components more spread out will make troubleshooting our PCB an easier process. Finally, our header

pins reserved for our external USB to UART converter are placed close to the module but not too close.

This figure shows the routing for our custom PCB design. The red traces represent the connections on the top layer, while the blue traces represent the connections on the bottom layer of the PCB.

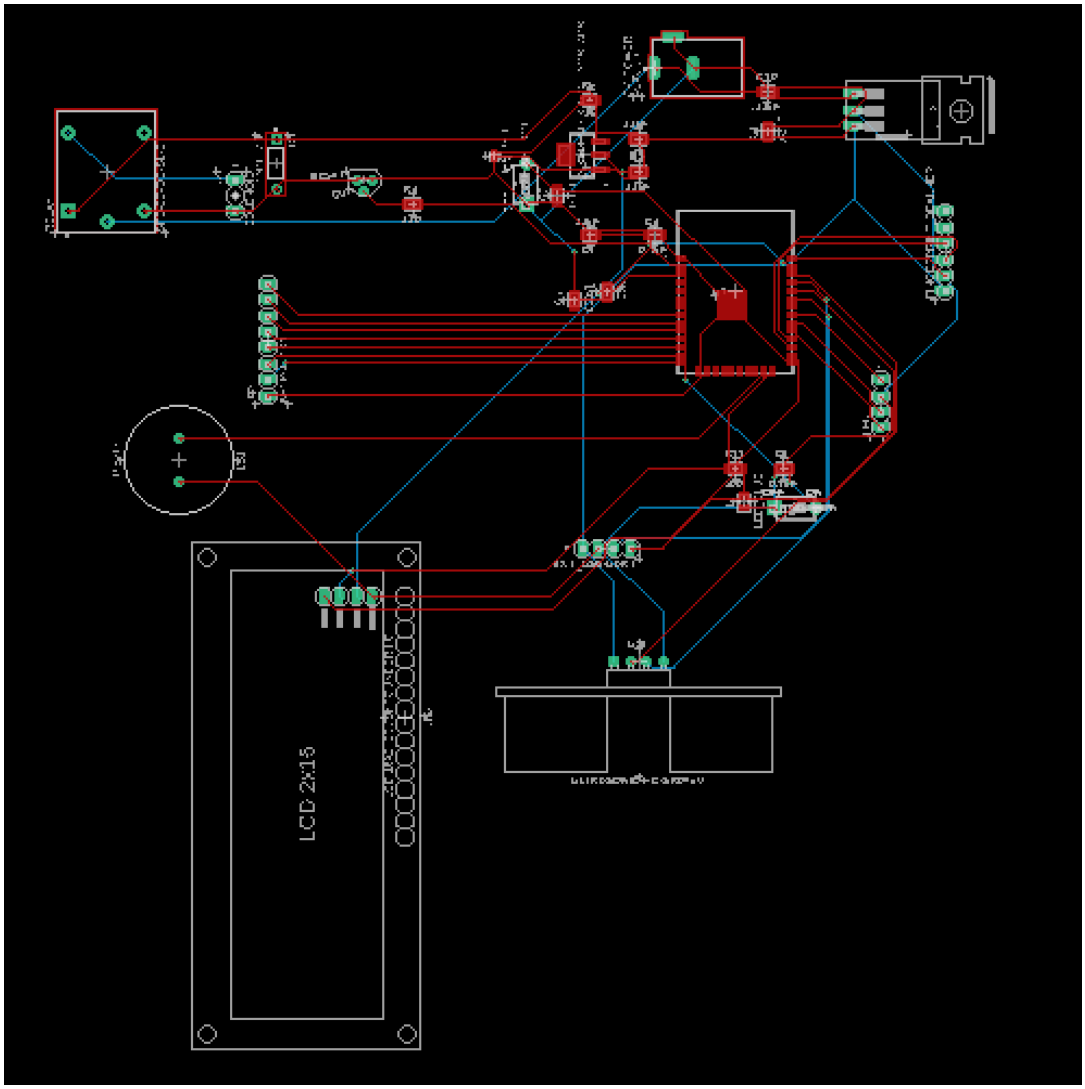


Figure 40: Ground Plane on Top Layer

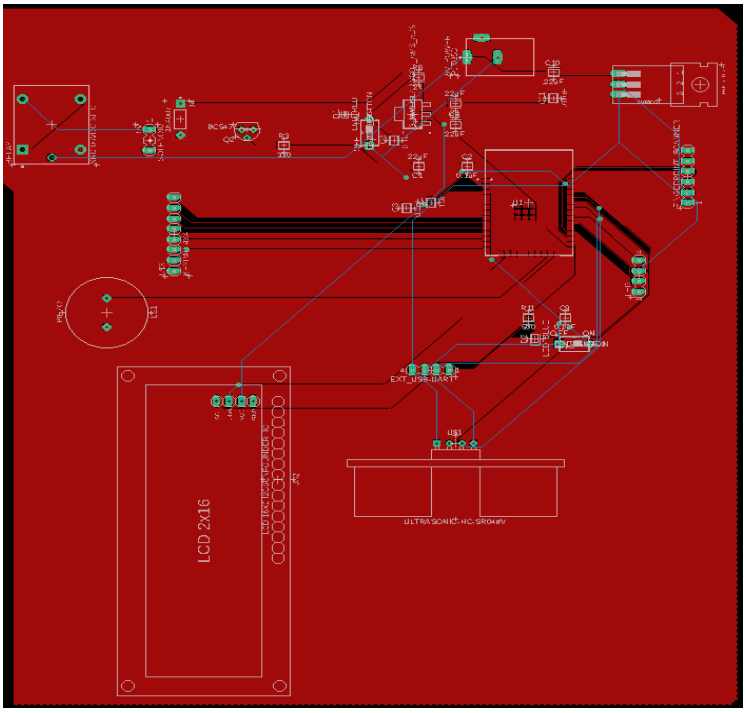


Figure 41: Ground Plane on Bottom Layer

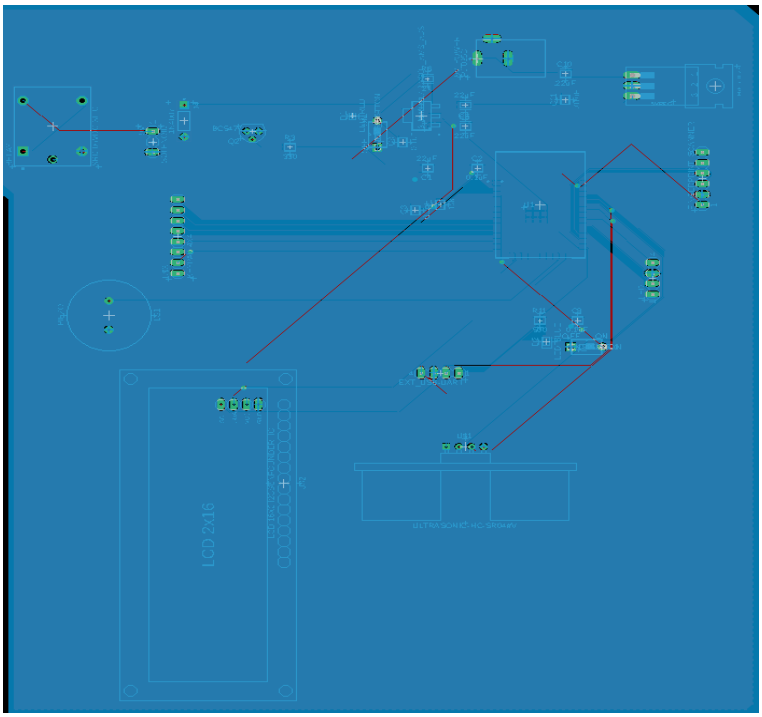


Figure 42: Ground Plane from Bottom Layer

Chapter 9: System Testing

Chapter 9 will briefly cover the hardware and software testing for each component of our project, evaluating the performance of each component. We will also discuss how each component will be integrated for our project. We will present our demo and what we implemented along with our plan for SD2.

9.1 Hardware Testing

This section of the chapter is where we detail our testing process for each of our hardware components to make sure that they are functional with zero issues.

9.1.1 ESP-WROOM-32

To test the functionality of the ESP-WROOM-32 development boards that we purchased, we connected them to our computers and uploaded the Blink example code and the Hello World example code from the Arduino sketchbook. What the Blink example code does is it makes the ESP-WROOM-32's built-in LED blink on and off every second while the Hello World example code prints the message "Hello World" to the Serial Monitor every second. We combined these two example codes and made it so that every second the LED would flash on and off while the phrase "Hello World" was printed to the serial monitor every second. The reason we chose these example codes was due to its simplicity and the fact that getting these codes to work verifies that the GPIO pins and serial communication works properly.

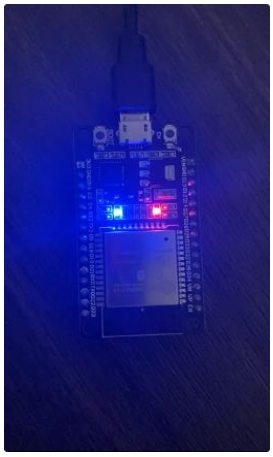


Figure 43: ESP32 powering

Figure 44: ESP32 in the terminal

9.1.2 Keypad

To test the keypad we first figured out how to connect the keypad to the GPIO pins on the ESP32. This was a little bit of a challenge because we had to carefully figure out which pins on the keypad are rows and columns and which GPIO pins we need to connect them to based on our code. After looking at the pinout schematic provided on Amazon we were able to configure the keypad to the ESP32 using double female pin headers. Since the first four pins are connected to the 4 columns of the keypad we connected those to their GPIO pins and did the same with the next four pins which are connected to the 4 rows of the keypad. We could not connect onto a breadboard because the breadboard did not have enough space for both sides and after troubleshooting we decided the breadboard might have a loose connection so we decided to connect the pins directly using jumper wires.

After this we installed the ESP32 Keypad Library from Arduino IDE that is designed to be used for Matrix Style Keypads. The code for the keypad was referenced from deepblueembedded.com and has been cited below [162]. In the code, we need to first include the library header file and define the number of rows and columns in our keypad. Then, we will create a two dimensional array that will include all of the characters on the keypad in their corresponding layout. Following this we will create two arrays that will assign the corresponding GPIO pins to their row and column. After creating an instance of the keypad we will pass the information defined above [162]. Finally, we will call a function that will print any key pressed by the user. We can test the keypad by checking the output on the serial monitor after pressing each key.

Below is a screenshot of the code from Arduino IDE showing the keypad layout and which GPIO pins they are connected to as well as the output on the Serial Monitor showing that each character on the keypad has worked after testing it [162]:

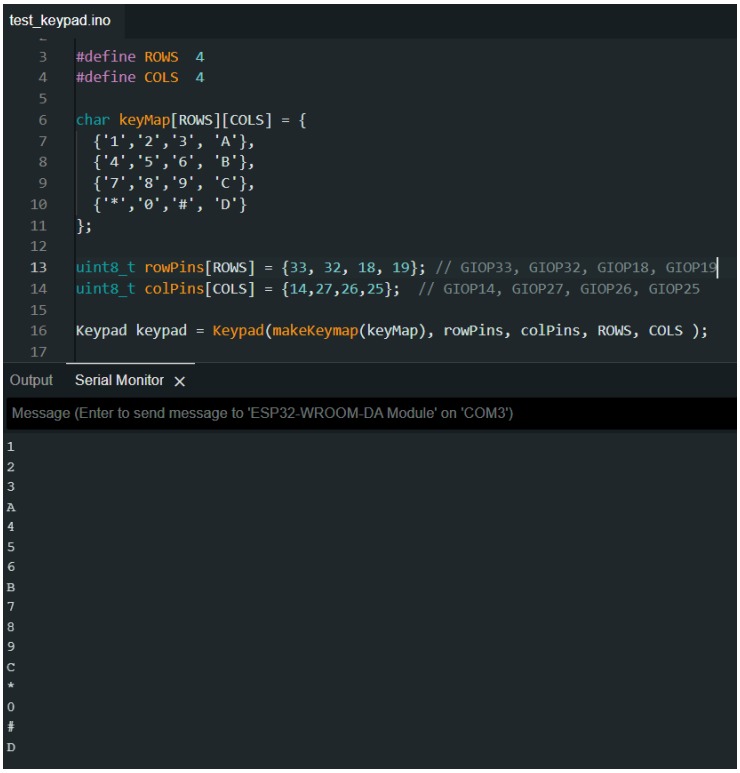


Figure 45: Keypad Component Testing

Below is a screenshot of a picture of the keypad connected to the ESP32 during testing:

Figure 46: Keypad Connected for Testing

9.1.3 Fingerprint Scanner

To test the fingerprint scanner we started by attaching DuPont male connector housings to the wires for a secure connection. For the scanner's setup, we connected GPIO16 to the scanner's TX (transmit) SO it can send data into the ESP32. We used GPIO17 for the scanner's RX (receive), allowing the ESP32 to send data to the scanner. To power the scanner, we connected its 5V and GND pins to the Vin and GND pins on the ESP32, respectively.

To use the fingerprint scanner, we need to write one program in C++ to enroll fingerprints and another to match them. During enrollment, the user must place their finger on the scanner twice to generate a fingerprint template. After the raw image is captured, it is temporarily stored in the image buffer. The system then processes this image to a template to be saved in the character buffer within the fingerprint library in the scanner's flash memory.

Once the fingerprint template is stored, we wrote a code to compare a newly scanned fingerprint in contrast to the templates in the library. When a user places their finger on the scanner, the raw image is temporarily saved in the image buffer. The system then extracts the template from the image and conducts a 1:N matching search through the entire

fingerprint library. After the matching process is completed, the template in the image buffer is deleted to prepare for the next scan.

Figure 47 is a picture of a user placing an unregistered fingerprint on the scanner with the results displayed on Arduino IDE in Figure 48. Figure 49 is a photo that shows a user placing a registered fingerprint on the scanner and the results shown in Figure 50.

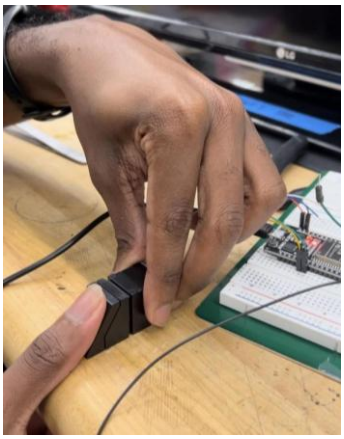


Figure 47: Unregistered fingerprint

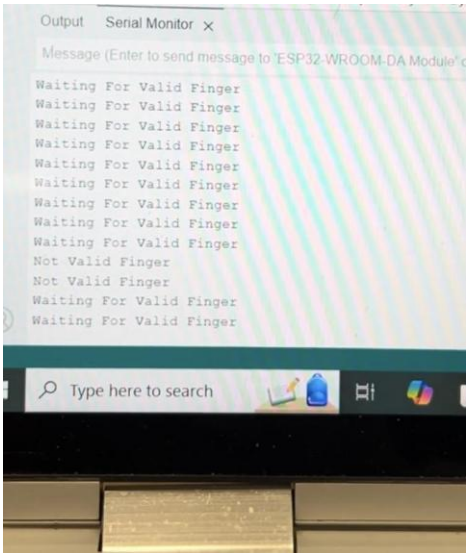


Figure 48: Unregistered fingerprint results

Figure 49: Registered fingerprint

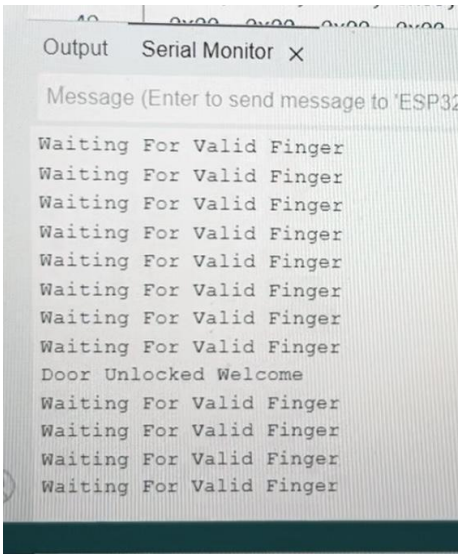


Figure 50: Registered fingerprint results

9.1.4 Locking Mechanism

To test out the locking mechanism, it is made up of five components to operate it. A solenoid, relay, transistor, and voltage regulator. To set it up, the relay has five pins to connect, the transistor has three pins, the voltage regulator has three pins while the solenoid has two terminals. On the relay, the signal and positive pin is wired to 5V from the voltage regulator, the negative pin is wired to ground, the COM (common) pin is wired to 9V, and the NO (normally open) pin is connected to the positive terminal of the solenoid. The anode side of the diode is connected to the negative pin of the relay and a wire to ground while the cathode side of the diode is connected to 5V from the voltage regulator then the positive and signal pin of the relay. The emitter pin of the transistor is wired to ground, the base uses a 330 ohm resistor to connect to GPIO 23 on the ESP32, and the collector is connected to the cathode side of the diode, the positive and signal pin of the relay. The positive terminal of the solenoid is wired to the NC of the relay while the negative terminal is wired to ground.

The solenoid is controlled by the relay like a switch and the transistor is the driver. The relay switches from NO position to NC (normally closed) position when the ESP32 signals the transistor to switch the relay, using low-voltage signals. The ESP32 sends 3.3V to energize the base of the transistor through a 330 ohm resistor. This low voltage signal turns the transistor on, allowing current to flow from the collector to the emitter, energizing the relay coil. Once the relay is activated by the 5V provided by the voltage regulator, the armature moves from the NO position to the NC position, which allows 9V to flow through the solenoid, activating it and unlocking the door.

Figure 51 is a picture that shows the solenoid with the power source on but still locked since the relay is not energized. Figure 52 is a picture that shows the solenoid unlocked, as the relay is energized, allowing current to flow through the solenoid.

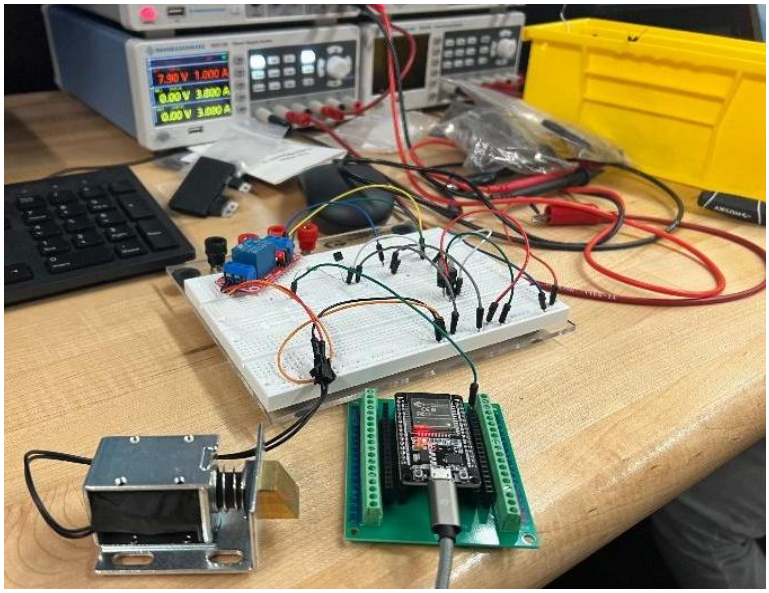


Figure 51: Solenoid Locked

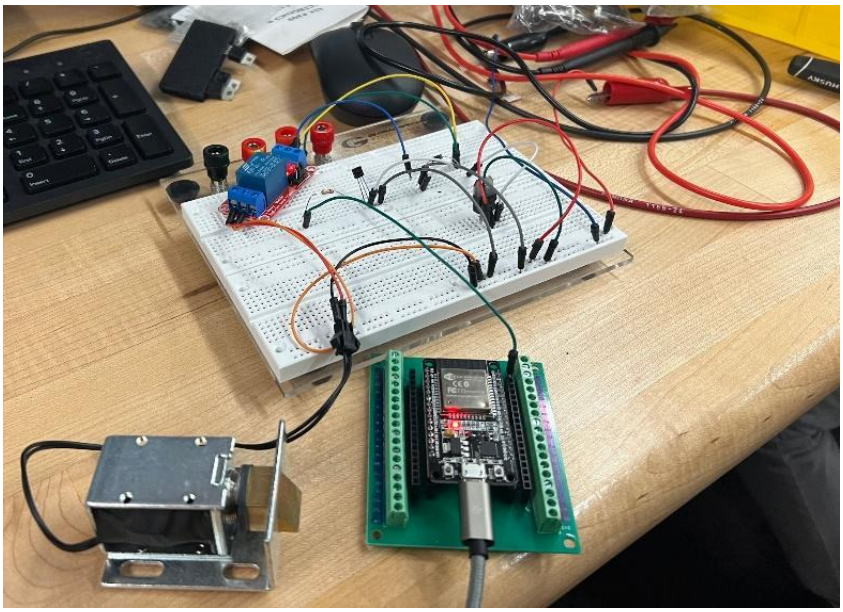


Figure 52: Solenoid Unlocked

9.1.5 Ultrasonic Sensor

To test the HC-SR04 Ultrasonic Sensor, we connected the Trig and Echo pins to GPIO pins 12 & 13 respectively and connected the Vin and GND pins to its appropriate GPIO pins on the ESP-WROOM-32. Next, we programmed it to calculate and print the distance value in inches & centimeters to the serial monitor. Furthermore, we set it up so that the distance would update every second if it detected an object that was closer or farther away from it.

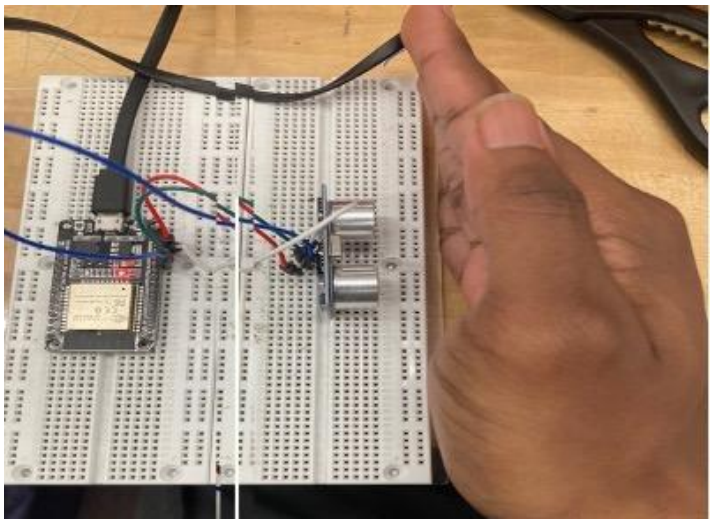


Figure 53: Ultrasonic Sensor Testing (close)

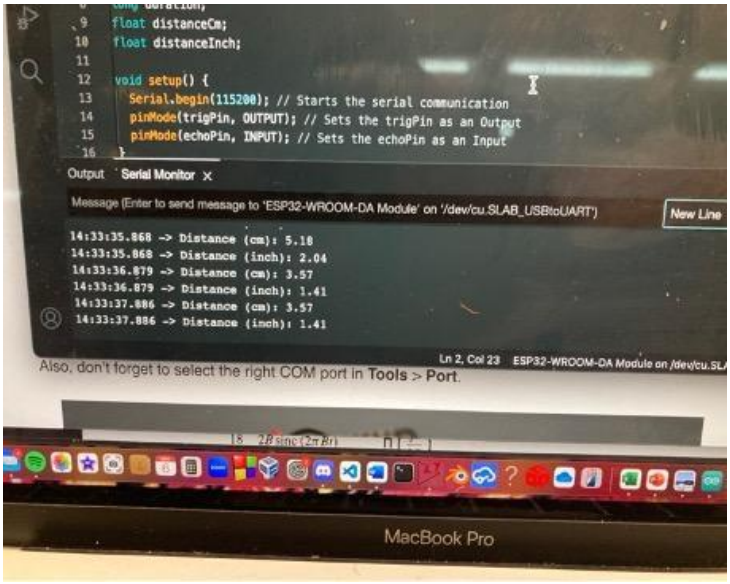


Figure 54: Ultrasonic Sensor Results (close)

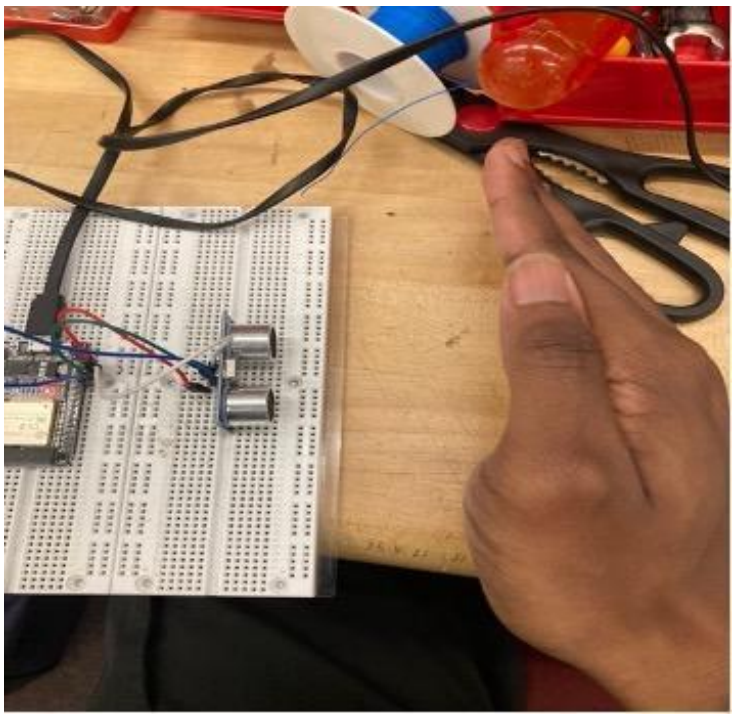


Figure 55: Ultrasonic Sensor Testing (far)

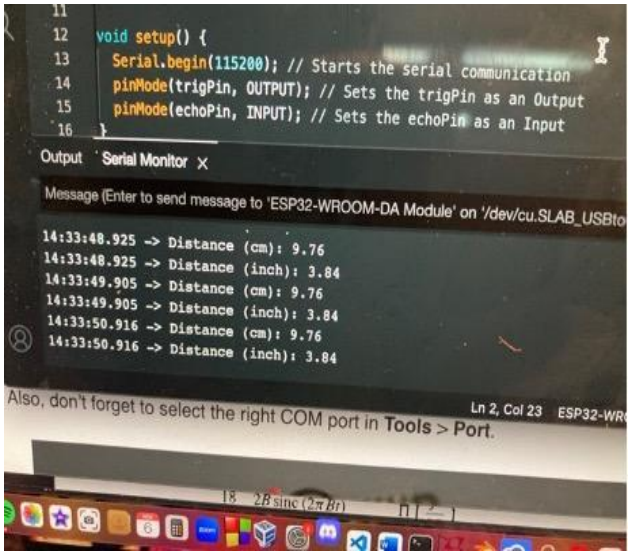


Figure 56: Ultrasonic Sensor Results (far)

9.1.6 Buzzer

To test our buzzer to see if it produces a sound loud enough to be heard, we connected the positive terminal & negative terminal to GPIO pin 25 and the GND pin on the ESP-WROOM-32 respectively on a breadboard. Next, we programmed the buzzer so that it makes a noise for one second and goes silent for four seconds and loops back again.

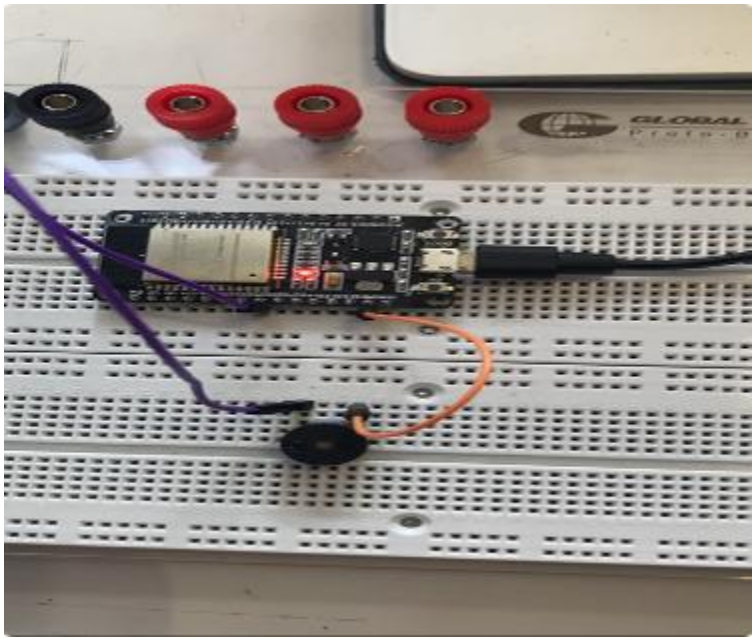


Figure 57: Buzzer connected to buzzer via breadboard

RGB LED Module

To test our RGB LED module, we connected the red channel to GPIO pin 2, the green channel to GPIO pin 4, the blue channel to GPIO pin 5 and the negative terminal to the GND pin. Next, we programmed it to have it output red for one second, output green for one second, blue for one second, purple for two seconds and output white for two seconds. When we first tested it, we ran into two major issues. The first issue was that the LED would output blue when it was supposed to output red, and it would output red when it was supposed to output blue. The second issue was that the LED would not output green at all. After troubleshooting the connections on the breadboard, I realized that the seller of the RGB module mislabeled the channel names to where red and blue were switched, and the green channel and the negative terminal were also switched. After fixing that issue and uploading the code again, the LED worked as intended. What we noticed throughout the testing process is that the green channel is 30-40% brighter than the red & blue channels on the RGB LED.

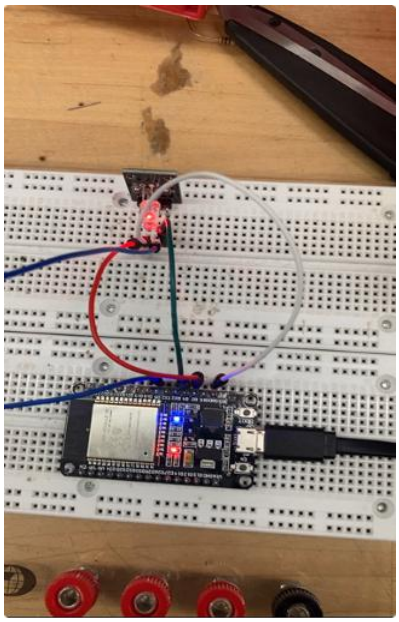


Figure 58: LED glowing red from ESP32 board

9.1.7 Display

We tested the display by connecting the indicated pins from the Display to the ESP32 which are GND, VCC, SDA, SCL. We connected SDA to GPIO 21 and SCL to GPIO 22. After doing this, we used code from randomnerdtutorials.com that provides basic code to print basic static text to the display [163]. After uploading the code to Arduino IDE we checked to make sure the LCD display was displaying a basic print message and connected to the ESP32. Below is a picture of the display printing a basic message:

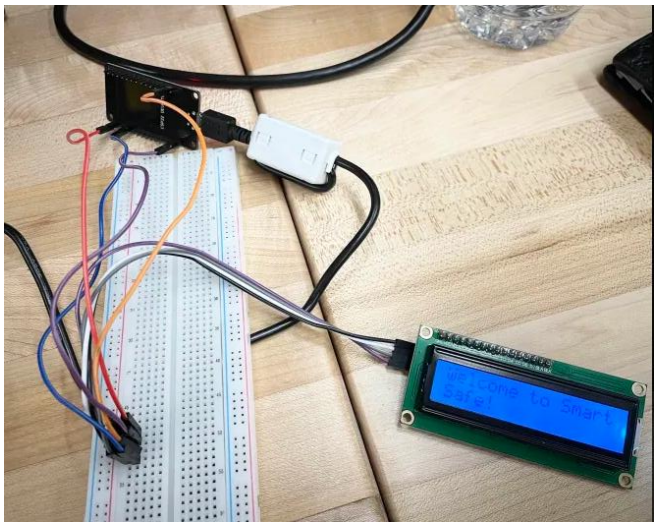


Figure 59: Display Component Testing

9.1.8 Web Server

To test the web server, we will be requesting an IP address through the ESP32 that can be accessed through a web browser and will print a test message showing that it is working. After setting up our development area, we can implement our testing code on Arduino IDE. The code for testing the Web Server was borrowed from instructables.com and has been cited in References [159]. The first thing we need to include in the code is the necessary libraries which are wifi.h and ESPAsyncWebServer.h [159]. The wifi.h library is used to connect the ESP32 to a local wifi-network that will allow it to wirelessly connect to the internet. This library is critical to our project because the email and website components of our project depend heavily on the ESP32 being properly connected to a reliable Wi-Fi network. The ESPAsyncWebServer.h library is specially designed for creating web servers on esp32. It handles HTTP requests which request a server for information to help establish a website. The next thing we need to do in our code is give our Wi-Fi credentials which are our Wi-Fi SSID and Password which will connect the MCU to our local Wi-Fi network. Next we need to create an instance of the AsyncWebServer class and define its route which will be used to send a get request that will provide an IP address [159]. After this we need to use the wifi.begin function which will connect it to the Wi-Fi network and if it was successfully connected we will print a message confirming this. Finally, we will call the wifi.localip function to retrieve an IP address and print this in the serial monitor. We will also do a server.begin function to start the server so it can listen to incoming http requests.

After uploading the code to the ESP32, we ran into a couple of issues with connecting to the Wi-Fi but after we did we were able to check the output in the Serial Monitor in Arduino IDE and received the IP address of 192.168.68.126. We tested this IP address in a web browser and was able to see the expected results with the given test message. This verified that we have created a web server with the esp32 and it has been successfully connected.

Below is a screenshot from Arduino IDE showing the web server request and connection to Wi-Fi:

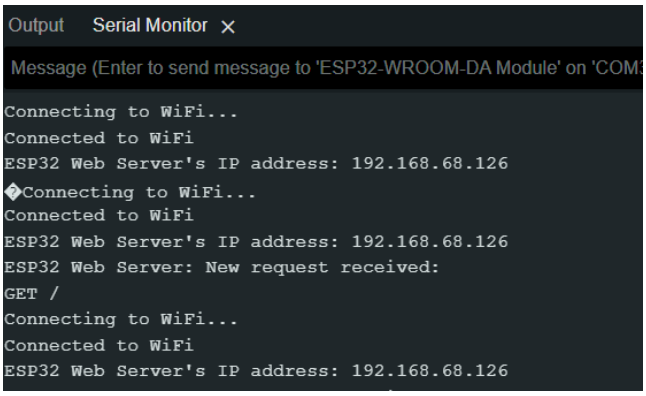


Figure 60: Web Server Testing Arduino

Below is a screenshot of the IP address from the requested web server showing the test message:

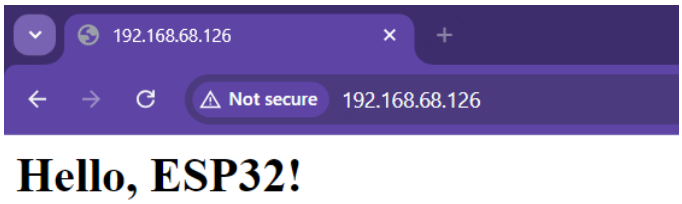


Figure 61: Web Server Testing Screenshot

9.1.9 Email

The email follows a similar format to the website but different integration with the code. We will be connecting to the Wi-Fi in the same way. The code for testing the Email was borrowed from https://randomnerdtutorials.com and has been cited in References [158]. We need to first include the following libraries: Arduino.h, Wifi.h, and ESP_Mail_Client.h [158]. The ESP_Mail_Client.h library is used to send and receive email notifications using ESP32 microcontroller devices. After providing our Wi-Fi credentials, we need to define the host name which is smtp.gmail.com and port number 465 [158]. Next, we need to include the sign in credentials which is our email and its password. We have created a separate Gmail account for this project called smartsafesd1@gmail.com and we need to provide the password we have created for this account. We also need to define the recipient email which will be a teammate's personal email for testing purposes. We then need to call the setup function which will scan the networks and connect to the local Wi-Fi network [158]. After configuring the session, with the given details we need to set the timestamp based on our location. We need to set up the message headers next which include the sender

name, email, subject and Recipient. Next, we need to insert the raw text message we want to send. If the email is not sent properly we will print an error message and the reason. After connecting to the server we need to make sure that email was authenticated and will print a successfully logged in message. We will call a function to get the status of the email which will inform us if it was sent successfully. After this, we checked the email server to confirm that the email was successful.

Below is a screenshot from Arduino showing the email was sent successfully:

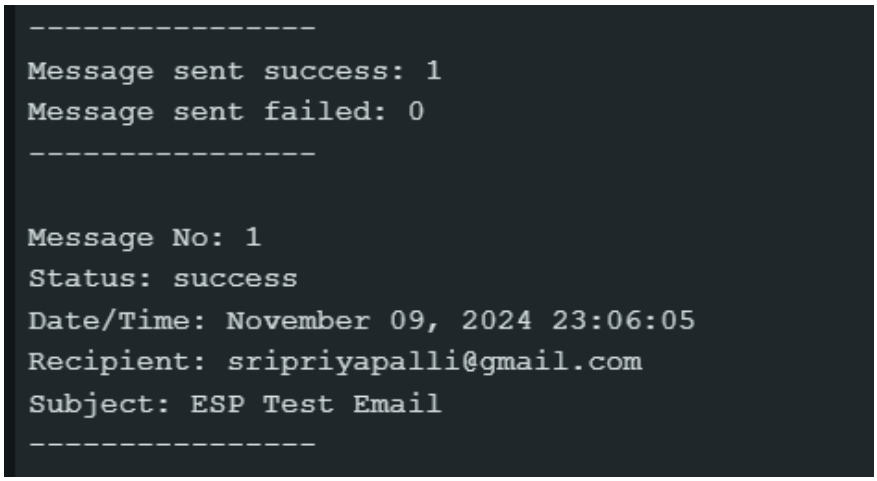


Figure 62: Email Testing Arduino

Below is a screenshot of the test email being shown in the Recipient's gmail inbox:

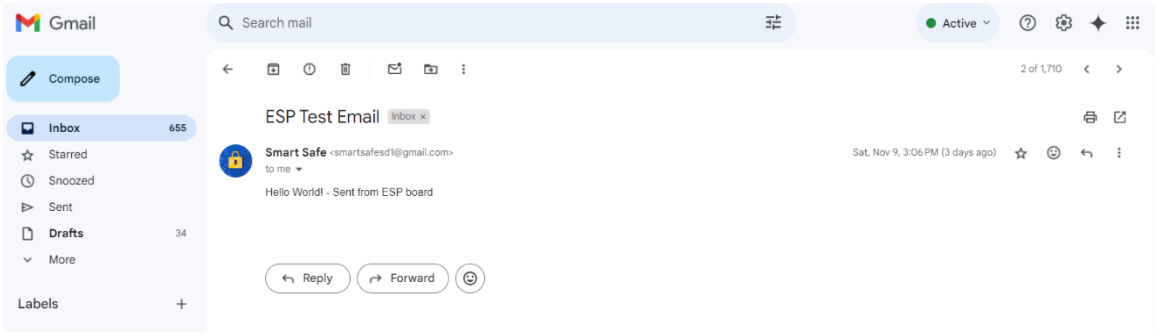


Figure 63: Email Testing Screenshot

9.2 Performance Evaluation

After testing each component required for our project we were able to evaluate its performance and test its individual functionality. We tested each component using Arduino IDE and confirmed that it was working correctly. The performance evaluation for each component is provided in the hardware and software testing section. We will test how some of the components are functioning together during our prototype demonstration.

9.3 Overall Integration

We plan to integrate all our components together and test the overall system in SD2. We plan to connect all our components on the ESP32 microcontroller which will control the main actions and manage communication between devices. We plan to test 3 or 4 components together for our SD1 prototype demonstration video. In our demo, we will be showing the keypad, fingerprint scanner along with the buzzer and LED working together. We will try to enter a code on the keypad if it is correct the buzzer will make a quick noise and the LED will turn on Green then we press the star key to clear the code and enter the wrong code which will make 2 quick buzzer noises. To test the fingerprint scanner, we will use one of the teammates' registered fingerprints and show that it turns the LED green and makes a buzzer noise. This shows that the major functionality of our Safe is working and communicating with the ESP32.

9.4 Plan for Senior Design 2

For Senior Design 2, we have some changes and optimizations that we want to work on for development as well as potential component changes. During integration over the major components, we noticed that some components interfered with each other when connecting them to the ESP32 microcontroller. In terms of interfering, the ESP32 has limited ports for certain components, so we had to prioritize some more major components over others in terms of early application. In the future, we plan on coming up with a solution to either implement both parts so that they can work in tandem such as an extender to provide more GPIOs for the ESP32 or we will compromise with an alternative part that will get the job done without causing any interference. For optimizations of our project, we plan on making our software structure less complex by implementing a style that works well with the keypad.

Chapter 10 Administrative Content

Smart Safe operates with a flexible budget and is self-funded without external sponsorship. One of our primary goals is to keep costs as minimal as possible for each group member, thereby making the project accessible and manageable for everyone involved.

Table 31 outlines the estimated costs for the key components essential to our project. These estimates are based on current prices and anticipated expenses, but it is important to note that they are subject to change. Variations in costs may arise due to several factors, including the size and specifications of the printed circuit board (PCB), which can significantly impact the overall material costs. Additionally, shipping expenses can fluctuate depending on the supplier and chosen shipping method, further influencing the final budget.

The table also provides a comprehensive overview of the materials required for the construction of Smart Safe, detailing both the quantities of items needed and the final price of each item. It also accounts for items sold in bulk, which impacts the overall cost. Ultimately, the total cost exceeds our budget because we opted for higher-quality and more robust materials than initially planned.

The safe will be constructed from birch plywood rather than metal to manage expenses and simplify the installation process. Plywood offers a cost-efficient and user-friendly alternative to metal, which can be more complex and expensive. It is important to note that not all construction items are listed in the table; for instance, the handle is integrated into the plywood measurements for the safe. Additionally, items such as resistors are not included in the material list because they will be provided by the lab at UCF.

Table 31: Bill of Materials [10]

Item	Quantity	Estimated Cost	Cost	Total
MCU	1	$20	$15.99	$15.99
Display	1	$18	$21.90	$21.90
Keypad	1	$3	$8.99	$8.99
Fingerprint Sensor	1	$24	$19.36	$19.36
Ultrasonic Sensor	1	$6	$6.99	$6.99
Solenoid Lock	1	$13	$6.99	$6.99
LED	1	$0.30	$9.99	$9.99
Buzzer	1	$6	$7.99	$7.99
Voltage Regulator	1	$5	$5.99	$5.99
Relay		$3	$3.99	$3.99
DC Power Connector	1	$4	$6.99	$6.99
Power Supply Adapter	1	$8	$9.99	$9.99

Battery	1	$8	$5.99	$5.99
UPS	1	$50	$31.50	$31.50
Birch Plywood	6	$5	$5.48	$32.88
Hinges	2	$4	$5.99	$5.99
PCB Board (with shipping)	5	$3	$2.00	$19.80
Total		$173.30		$207.34

Table 32: Project Timeline

Task	Start Date	Anticipated End Date	Duration
Group Recruitment	07/09/2024	08/18/2024	6 weeks
Project Idea Brainstorming	07/09/2024	08/16/2024	6 weeks
Project Scope Finalized	07/09/2024	08/18/2024	6 weeks
Individual Research	07/09/2024	09/05/2024	9 weeks
Divide and Conquer Report	08/25/2024	09/06/2024	2 weeks
Divide and Conquer Meeting	09/10/2024	09/10/2024	30 minutes
Divide and Conquer Revision	09/10/2024	09/27/2024	2 weeks
Upload DC Report on Website	09/10/2024	9/27/2024	2 weeks
Email DC Report to Committee	09/10/2024	9/27/2024	2 weeks
60 Page Milestone	09/27/2024	10/25/2024	4 weeks
60 Page Report Group Meeting	10/29/2024	10/28-30/2024	30 minutes
Upload Revised 60 Page Report	10/30/2024	11/08/2024	2 weeks
120 Page Milestone	10/25/2024	11/22/2024	5 weeks
Individual System Design	09/06/2024	10/03/2024	4 weeks
Individual System Testing	10/04/2024	10/29/2024	4 weeks
Breadboard Prototyping	10/30/2024	11/17/2024	4weeks
PCB Designing	10/30/2024	12/09/2024	3 weeks
Final Report	11/22/2024	11/26/2024	1 week
Upload Revised Final Report	11/26/2024	12/05/2024	1 week

Table 33: Senior Design 2 Project Milestones

Task	Start Date	Anticipated End Date	Duration
PCB Ordering	12/10/2024	01/06/2025	3 weeks
PCB Assembly	01/07/2025	01/21/2025	2 weeks
PCB Testing	01/22/2025	02/05/2025	2 weeks
System Integration/Testing	02/06/2025	03/06/2025	4 weeks
Practice Project Demo	03/10/2025	03/24/2025	2 weeks
Finalize Documentation	03/25/2025	04/08/205	2 weeks
Practice Final Presentation	04/10/2025	04/18/2025	1 week
Final Presentation	TBD	TBD	TBD

The team for Smart Safe was formed and began brainstorming ideas over the summer in preparation for Senior Design 1. During this period, we explored various concepts and solutions to build a solid foundation for our project.

Table 32 outlines the progress we've made so far and our plans for the remainder of the semester. Certain task deadlines were delayed due to time constraints caused by school closures and power outages from several hurricanes; however, we were still able to complete all tasks on schedule. Table 33 outlines our anticipated milestones and objectives for each month leading up to the end of Senior Design 2. Both tables offer a preliminary timeline and are subject to adjustment based on ongoing developments, although tasks with established deadlines will remain fixed.

Chapter 11: Conclusion

The Smart Safe is a collaborative project developed by engineering students at the University of Central Florida, designed to address the growing demand for secure, user-friendly storage solutions. As outlined in the executive summary, the primary objective of this project is to provide a state-of-the-art safe that incorporates advanced technologies to tackle modern security challenges. In an age where security concerns are ever-increasing, the Smart Safe offers a robust solution to protect valuables from theft or unauthorized access. By integrating features such as a fingerprint scanner, keypad, motion sensors, and a solenoid lock, the safe delivers multiple layers of security while prioritizing user convenience and customization. Additionally, real-time email notifications further enhance the user experience by providing instant updates and heightened security awareness [160].

The Smart Safe includes a battery backup system, ensuring functionality even during power outages or extreme environmental conditions. A stretch goal of incorporating a temperature sensor would provide early detection of high temperatures, alerting the user in case of potential fire hazards. The safe's lockdown mode further strengthens its security by detecting suspicious activity, such as excessive access attempts, and automatically locking the system while notifying the user. Our design also emphasizes accessibility by including a buzzer and LED status indicators, making the safe suitable for users with hearing or visual impairments. Its portable design adds flexibility for users who may need to move or travel with the safe, with the built-in motion sensor detecting unauthorized movements and alerting the owner. A user-friendly website complements the hardware, offering an intuitive platform to manage email notifications and access customer support [160].

The research and development undertaken for this Senior Design 1 report provided a solid framework for the next phase of the project. It allowed us to refine our design, address potential challenges, and establish a clear plan for implementation in Senior Design 2. Chapter 10's timeline and budget analysis equipped us with a detailed understanding of the tasks and materials required for the project's completion. Our goal is to deliver a fully functional, demonstrable Smart Safe by the end of Senior Design 2, presenting it to our peers and the Senior Design Committee [160].

This project has not only addressed pressing security concerns but has also strengthened our problem-solving and teamwork skills. The knowledge and experience gained through this process will undoubtedly enhance our abilities as engineers and contribute to our future endeavors in advancing technology. We are confident that the Smart Safe will serve as a meaningful contribution to modern security solutions while showcasing the technical and collaborative skills we have developed throughout this journey [160].

Appendix A – References

[1] “Video Doorbell Cameras | Wireless and Wired,” *Ring*. https://ring.com/video-doorbell-cameras

Used for Software Diagram:

[2] “Flowchart Maker & Online Diagram Software,” *Draw.io*, 2019. http://draw.io

Used for Hardware diagram:

[3] “SmartDraw - Create Flowcharts, Floor Plans, and Other Diagrams on Any Device,” *www.smartdraw.com*. http://smartdraw.com

Safe logo on Cover page:

[4] V. Icons, *Smart Safebox vector icon*. [Online]. Available: https://www.vecteezy.com/vector-art/21715408-smart-safebox-vector-icon

[5] “SD Group 6 Smart Lock,” *Ucf.edu*, 2024. https://www.ece.ucf.edu/seniordesign/su2019fa2019/g06/#Doc (accessed Sep. 06, 2024).

[6] “Smart Cooler,” *Ucf.edu*, 2024. https://www.ece.ucf.edu/seniordesign/sp2022su2022/g20/ (accessed Sep. 06, 2024).

[7] “Smart Door Security System,” *Ucf.edu*, 2021. https://www.ece.ucf.edu/seniordesign/sp2021su2021/g15/ (accessed Sep. 06, 2024).

[8] *Amazon.com*, 2024. https://www.amazon.com/b?ie=UTF8&node=13853235011#:~:text=Amazon%20Hub%20Locker%20is%20a (accessed Sep. 06, 2024).

[9] “Amazon.com,” *A.co*, 2014. https://a.co/d/b1kJqVp (accessed Sep. 06, 2024).

[10] “ESP32 Fingerprint Sensor Based Door Lock Security System,” How To Electronics, Feb. 06, 2020. https://how2electronics.com/esp32-fingerprint-sensor-security-system/

[11] ChatGPT, "Word Document Formatting Guide" OpenAI. Available: https://chatgpt.com/share/62588d64-90ce-4824-89d5-0ccb2fc2f64b. [Accessed: Sep. 9, 2024].

[12] “Honeywell 5403 Bluetooth Smart Security Safe with Digital Touch Screen (.97 CU ft.),” the official Honeywell brand store, https://www.honeywellstore.com/store/products/bluetooth-smart-security-safe-with-digital-touch-screen-5403.htm (accessed Oct. 24, 2024).

[13] https://www.amazon.com/dp/B0CQY5PS54 (accessed Oct. 24, 2024).

[14] “TurboSAFE WiFi Smart Safe with Monitoring App & Tamper Alerts | 0.75-Ft. Cubed High-Capacity, Aircraft-Grade Durability – Turbolock.com,” Turbolock.com, 2021. https://turbolock.com/product/turbosafe-wifi-smart-safe-with-monitoring-app-tamper-alerts/ (accessed Oct. 24, 2024).

[15] J. Schneider and I. Smalley, “What is a microcontroller? | IBM,” *www.ibm.com*, Jun. 04, 2024. https://www.ibm.com/think/topics/microcontroller

[16] “Everything You Need To Know About Microcontrollers | RS,” *Rs-online.com*, 2024. https://ie.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/microcontrollers-guide&srsltid=AfmBOoq0WOoV1QG3sAj0jRK-5HxIYZ3MSvrQkSNlcxPhYJP1rTzUYGso (accessed Oct. 05, 2024).

[17] “What is a microprocessor? | IBM,” *www.ibm.com*, Jun. 10, 2024. https://www.ibm.com/think/topics/microprocessor

[18] GeeksforGeeks, “Introduction of Microprocessor,” *GeeksforGeeks*, Sep. 27, 2018. https://www.geeksforgeeks.org/introduction-of-microprocessor/

[19] Arduino, “What Is Arduino?,” *Arduino.cc*, Feb. 05, 2018. https://www.arduino.cc/en/guide/introduction

[20] “Arduino Uno Rev3,” *Arduino Online Shop*, 2024. https://store-usa.arduino.cc/products/arduino-uno-rev3?gad_source=1&gbraid=0AAAACbEa86TnXUUJkhFhGR9NoMgYEh0-&gclid=CjwKCAjwxY-3BhAuEiwAu7Y6s84eEW_zFwK82b9dLhQB20JHcDtc3QpUwf9hz5Ac-CfcrfUiZn_ePRoC4S4QAvD_BwE (accessed Oct. 05, 2024).

[21] “Arduino Mega 2560 Rev3,” *Arduino Online Shop*. https://store-usa.arduino.cc/products/arduino-mega-2560-rev3

[22] “Arduino Nano,” *Arduino Online Shop*. https://store-usa.arduino.cc/products/arduino-nano

[23] “• Advanced RISC Architecture -135 Powerful Instructions -Most Single Clock Cycle Execution -32 × 8 General Purpose Working Registers -Fully Static Operation -Up to 16 MIPS Throughput at 16MHz -On-Chip 2-cycle Multiplier • High Endurance Non-volatile Memory Segments -64K/128K/256KBytes of In-System Self-Programmable Flash -4Kbytes EEPROM -8Kbytes Internal SRAM -Write/Erase Cycles:10,000 Flash/100,000 EEPROM -Data retention: 20 years at 85°C/ 100 years at 25°C -Optional Boot Code Section with Independent Lock Bits • In-System Programming by On-chip Boot Program • True Read-While-Write Operation -Programming Lock for Software Security • Endurance: Up to 64Kbytes Optional External Memory Space • QTouch ® library support -Capacitive touch buttons, sliders and wheels -QTouch and QMatrix acquisition -Up to 64 sense channels • JTAG (IEEE ® std. 1149.1 compliant) Interface -Boundary-scan Capabilities According to the JTAG Standard -Extensive On-chip Debug Support -Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface • Peripheral Features -Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode -Four 16-bit Timer/Counter with Separate Prescaler, Compare-and Capture Mode -

Real Time Counter with Separate Oscillator -Four 8-bit PWM Channels -Six/Twelve PWM Channels with Programmable Resolution from 2 to) -Master/Slave SPI Serial Interface -Byte Oriented 2-wire Serial Interface -Programmable Watchdog Timer with Separate On-chip Oscillator -On-chip Analog Comparator -Interrupt and Wake-up on Pin Change • Special Microcontroller Features -Power-on Reset and Programmable Brown-out Detection -Internal Calibrated Oscillator -External and Internal Interrupt Sources -Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby." Available: http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf

[24] "Top 11 Applications Of Raspberry Pi," *Engineers Garage*, Jun. 14, 2020. https://www.engineersgarage.com/top-11-applications-of-raspberry-pi/

[25] A. Industries, "Raspberry Pi 3 - Model B+ - 1.4GHz Cortex-A53 with 1GB RAM," *www.adafruit.com*. https://www.adafruit.com/product/3775?src=raspberrypi

[26] Adafruit Industries, "Raspberry Pi 4 Model B - 2 GB RAM," *Adafruit.com*, 2019. https://www.adafruit.com/product/4292

[27] "Raspberry Pi 5 - 8GB - DEV-23551 - SparkFun Electronics," *www.sparkfun.com*. https://www.sparkfun.com/products/23551?src=raspberrypi

[28] "Processors - Raspberry Pi Documentation," *Raspberrypi.com*, 2024. https://www.raspberrypi.com/documentation/computers/processors.html#bcm2711 (accessed Oct. 07, 2024).

[29] "Processors - Raspberry Pi Documentation," *Raspberrypi.com*, 2024. https://www.raspberrypi.com/documentation/computers/processors.html#bcm2711

[30] Random Nerd Tutorials, "Getting Started with the ESP32 Development Board | Random Nerd Tutorials," *Random Nerd Tutorials*, Aug. 27, 2018. https://randomnerdtutorials.com/getting-started-with-esp32/

[31] I. Hübschmann, "ESP32 for IoT: A Complete Guide," *Nabto*, Aug. 28, 2020. https://www.nabto.com/guide-to-iot-esp-32/

[32] "ESP32-PICO-KIT-1 - - — esp-dev-kits latest documentation," *Espressif.com*, 2016. https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-pico-kit-1/user_guide.html#overview (accessed Oct. 07, 2024).

[33] "ESP32-PICO-KIT-1 | DigiKey Electronics," *DigiKey Electronics*, 2024. https://www.digikey.in/en/products/detail/espressif-systems/ESP32-PICO-KIT-1/21572663?s=N4IgTCBcDaIKIGUAKBmMBaJBJAwgeXQGksAVdARhAF0BfIA (accessed Oct. 07, 2024).

[34] "ESP32-H2-DevKitM-1 - - — esp-dev-kits latest documentation," *Espressif.com*, 2024. https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32h2/esp32-h2-devkitm-1/user_guide.html (accessed Oct. 07, 2024).

[35] "ESP32-H2-DEVKITM-1-N4 | DigiKey Electronics," *DigiKey Electronics*, 2024. https://www.digikey.com/en/products/detail/espressif-systems/ESP32-H2-DEVKITM-1-

N4/18109238?s=N4IgTCBcDaIBZgLQBMCmA3A1gSwC4FtEBGEAXQF8g (accessed Oct. 07, 2024).

[36] "ESP32WROOM32D & ESP32WROOM32U Datasheet." Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

[37] https://www.amazon.com/HiLetgo-ESP-WROOM-32-Development-Microcontroller-Integrated/dp/B0718T232Z/ref=asc_df_B0718T232Z/?tag=hyprod-20&linkCode=df0&hvadid=692875362841&hvpos=&hvnetw=g&hvrand=11844712584640068761&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9053023&hvtargid=pla-2281435178778&psc=1&mcid=3b3ebda12e0838758f86efcad7950f0c&hvocijid=11844712584640068761-B0718T232Z-&hvexpln=73

[38] ChatGPT, "Keypad Technology Comparison," OpenAI. Available: https://chatgpt.com/share/66e70964-324c-8008-967e-d147addbb6e5. [Accessed: Sep. 25, 2024].

Reference for Membrane Keypads:

[39]Butler Technologies Inc, "Membrane Switch Manufacturer | Butler Technologies, Inc.," *Butlertechnologies.com*, 2016. https://butlertechnologies.com/membrane-switches (accessed Oct. 24, 2024).

[40] "ChatGPT," *Chatgpt.com*, 2024. https://chatgpt.com/share/66fddaff-7ab8-8008-93cc-8c47f548ab90 (accessed Oct. 24, 2024).

[41] "Matrix Membrane Keyboard for Arduino Microcontroller," Amazon. https://www.amazon.com/Matrix-Membrane-Keyboard-Arduino-MicrocontrollerWIshioT/dp/B07B4DR5SH (accessed Oct. 24, 2024).

[42] "Adafruit 3x4 Phone-style Matrix Keypad," Amazon. https://www.amazon.com/Adafruit-3x4-Phone-style-Matrix-Keypad/dp/B00QSHPCO8 (accessed Oct. 24, 2024).

[43] "e-outstanding Adhesive Membrane Keyboard for Prototyping," Amazon. https://www.amazon.com/outstanding-Adhesive-Membrane-Keyboard-Prototyping/dp/B08JYNM8D9 (accessed Oct. 24, 2024).

[44]D. Bratu, "How fingerprint sensors work? - Dragoș Bratu - Medium," Medium, Jan. 28, 2023. https://medium.com/@dragos.bratu/how-fingerprint-sensors-works-12b53e6c9f8 (accessed Oct. 25, 2024).

[45]M. Clark, "Fingerprint Sensors Face-off: Capacitive Sensor vs. Optical Sensor," *Bayometric*, Aug. 08, 2018. https://www.bayometric.com/capacitive-vs-optical/ (accessed Oct. 25, 2024).

[46]“The Fingerprint File: 4 Fingerprint Sensor Types,” *www.aratek.co*. https://www.aratek.co/news/the-4-fingerprint-sensor-types

[48]Fingerprints, “Fingerprints FPC1020 Touch Sensor,” *Fingerprints*, 2022. https://www.fingerprints.com/technology/hardware/sensors/fpc1020/ (accessed Oct. 25, 2024).

[49]Fingerprints, “Fingerprints’ Access Sensor Series,” *Fingerprints*, 2022. https://www.fingerprints.com/solutions/access/fpc-access-sensor-series/ (accessed Oct. 25, 2024).

[50]“Blocked Page,” *Techiesms.com*, 2024. https://techiesms.com/product/r307-optical-fingerprint-reader/ (accessed Oct. 25, 2024).

[51]R. Triggs, “How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained,” *Android Authority*, Mar. 29, 2019. https://www.androidauthority.com/how-fingerprint-scanners-work-670934/

[52]“How Fingerprint Scanners Work,” *HowStuffWorks*, Sep. 24, 2002. https://computer.howstuffworks.com/fingerprint-scanner.htm#pt2

[53]“Explainer: Fingerprint Matching,” *Biometric Update*, May 12, 2019. https://www.biometricupdate.com/201512/explainer-fingerprint-matching

[54]Thermal Fingerprint Scanner, “Thermal Fingerprint Scanner,” *Bioenabletech.com*, 2024. https://aadhaar.bioenabletech.com/thermal-fingerprint-scanner (accessed Oct. 25, 2024).

[55]*Sparkfun.com*, 2018. https://www.sparkfun.com/products/14585 (accessed Oct. 25, 2024).

[56]“Optical Fingerprint Module,” *Amazon*, Oct. 24, 2024. https://www.amazon.com/Optical-Fingerprint-Control-Presence-Recognition/dp/B089LXDG94/ref=asc_df_B089LXDG94/?tag=hyprod-20&linkCode=df0&hvadid=693507718463&hvpos=&hvnetw=g&hvrand=1025000540439092044&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9217253&hvtargid=pla-1156803388549&psc=1&mcid=a092d60368ed33c58353e84f14d4be90

[57]“GROW R307S Fingerprint Module_Hangzhou Grow Technology Co., Ltd,” *Hzgrow.com*, 2024. http://en.hzgrow.com/product/188.html (accessed Oct. 25, 2024).

[58]“AS608 Fingerprint Reader Sensor Module Optical Fingerprint Module Locks Serial Communication Interface,” *Elecbee Factory*, 2023. https://www.elecbee.com/en-24787-AS608-Fingerprint-Reader-Sensor-Module-Optical-Fingerprint-Module-Locks-Serial-Communication-Interface (accessed Oct. 25, 2024).

[59]“Handson Technology User Guide AS608 Optical Finger Print Sensor Module.” Accessed: Oct. 25, 2024. [Online]. Available: https://handsontec.com/dataspecs/sensor/AS608%20Finger%20print%20Sensor.pdf

[60] Types of Motion Sensors https://chatgpt.com/share/6709e74f-0610-800d-ab5b-80927c0d6c6f

[61] Ultrasonic Sensor Pin Illustration Ultrasonic HC-SR04 Sensor Interfacing Arduino - Distance Measurement (microcontrollerslab.com)

[62] HC-SR04 Information https://www.elementzonline.com/blog/ultrasonic-sensor-hc-sr04-with-arduino

[63] URM37 V5 Information https://wiki.dfrobot.com/URM37_V5.0_Ultrasonic_Sensor_SKU_SEN0001_

[64] SparkFun TCT-40 Information https://www.sparkfun.com/products/24805

[65] “ChatGPT - Types of Electronic Displays,” *ChatGPT*, 2024. https://chatgpt.com/share/66fbaa3c-0280-800d-baef-aaec8003db4d (accessed Oct. 25, 2024).

[66] https://www.amazon.com/LILYGO%C2%AE-T-Display-RP2040-Raspberry-Development/dp/B09J112YR7/ref=sr_1_4?crid=13TMVEQJZ7ERX&dib=eyJ2IjoiMSJ9.cK8WlhsvRAAwTdqA43fLNULw8XQqzjaEu7WOzPoAP6HxzEzPxkgHkMen09rEhZngBPyeSwkxI3jBdCrOeyK733G5bOM5e4JMf0lZmbBq9Kzjlqh6O7Ecj5WDfxBD_hZ3qOJfzX9kdZAidae_mzjDxc0yuKDsgsMzpRy6OIQVLNEjw8RkoiO-W5fXrQPvG4R59Kcf2o4g4t46tlKRiOcOXPWQ7W46KZNGLsIbhawpW78.FXF2R1Byg-djL_s9m_wFHI_QaXVpDOP5GEoooz3LjN0&dib_tag=se&keywords=LCD%2Bdisplay%2Besp32&qid=1727761095&sprefix=lcd%2Bdisplay%2Besp32%2Caps%2C111&sr=8-4&th=1

[67] https://www.amazon.com/Freenove-Display-Compatible-Arduino-Raspberry/dp/B0B76YGDV4/ref=sr_1_3?crid=1JHC97MKTM0W5&dib=eyJ2IjoiMSJ9.cK8WlhsvRAAwTdqA43fLNULw8XQqzjaEu7WOzPoAP6HxzEzPxkgHkMen09rEhZngBPyeSwkxI3jBdCrOeyK733G5bOM5e4JMf0lZmbBq9Kzjlqh6O7Ecj5WDfxBD_hZ3qOJfzX9kdZAidae_mzjDxc0yuKDsgsMzpRy6OIQVLNEjw8RkoiO-W5fXrQPvG4R59Kcf2o4g4t46tlKRiOcOXPWQ7W46KZNGLsIbhawpW78.FXF2R1Byg-djL_s9m_wFHI_QaXVpDOP5GEoooz3LjN0&dib_tag=se&keywords=LCD%2Bdisplay%2Besp32&qid=1727762555&sprefix=lcd%2Bdisplay%2Besp32%2Caps%2C131&sr=8-3&th=1

[68] https://www.amazon.com/Touchscreen-ESP-WROOM-32-Resistive-ESP32-2432S028R-Development/dp/B0CXHPP7DM/ref=sr_1_4?crid=13TMVEQJZ7ERX&dib=eyJ2IjoiMSJ9.cK8WlhsvRAAwTdqA43fLNULw8XQqzjaEu7WOzPoAP6HxzEzPxkgHkMen09rE

hZngBPyeSwkxI3jBdCrOeyK733G5bOM5e4JMf0lZmbBq9Kzjlqh6O7Ecj5WDfxBD_hZ3qOJfzX9kdZAidae_mzjDxc0yuKDsgsMzpRy6OIQVLNEjw8RkoiO-W5fXrQPvG4R59Kcf2o4g4t46tlKRiOcOXPWQ7W46KZNGLsIbhawpW78.FXF2R1Byg-djL_s9m_wFHI_QaXVpDOP5GEoooz3LjN0&dib_tag=se&keywords=LCD+display+esp32&qid=1727762418&sprefix=lcd+display+esp32%2Caps%2C111&sr=8-4

[69] [1]“Unveiling the power of ups lithium battery - a comprehensive guide,” *Huntkey & GreVault Battery Energy Storage Systems*, Sep. 15, 2023. https://www.huntkeyenergystorage.com/ups-lithium-battery/ (accessed Nov. 26, 2024).

[70] BYJU'S, “How Do Alkaline Batteries Work - Structure of Alkaline Battery, Construction and Working of Alkaline Battery along with FAQs,” *BYJUS*. https://byjus.com/chemistry/how-do-alkaline-batteries-work/

[71] “Lithium-ion Batteries - Advantages, Drawbacks, Alternatives,” *BYJUS*. https://byjus.com/free-ias-prep/lithium-ion-batteries/

[72] *Sparkfun.com*, 2014. https://www.sparkfun.com/products/12895 (accessed Nov. 26, 2024).

[73] “PRODUCT SPECIFICATION,” 2014. Accessed: Nov. 26, 2024. [Online]. Available: https://cdn.sparkfun.com/datasheets/Prototyping/ICR18650%202600mAH%20datasheet.pdf

[74] “Samsung 50S 21700 3.6V 5000mAh 25A Lithium Ion (Li-Ion) Flat Top Battery,” *Batteryjunction.com*, 2016. https://www.batteryjunction.com/samsung-inr-21700-50s?srsltid=AfmBOoqmQrhByI6l2_3fjmcReo-p5ZQcRheIe-if44cm8ia3bRx7LvlxsFw (accessed Nov. 26, 2024).

[75] “Lithium-ion rechargeable cell for power tools Model name : INR21700-50S Energy Business Division,” 2020. Accessed: Nov. 26, 2024. [Online]. Available: https://files.batteryjunction.com/frontend/files/samsung/datasheet/SAMSUNG-INR21700-50S-DataSheet.pdf

[76] [1]“Tenergy Li-ion 18650 Cylindrical 3.7V 2600mAh Flat Top Rechargeable Battery-UL listed,” Tenergy Power, 2024. https://power.tenergy.com/tenergy-li-ion-18650-cylindrical-3-7v-2600mah-flat-top-rechargeable-battery-ul-listed/?srsltid=AfmBOorbErcZY0lne4CFJ-Bc47BJDNiRyoMSxE0moGBu9NP7chvDTyDX7f4 (accessed Nov. 26, 2024).

[77] Prashant digital, “All You Need To Know About Voltage Regulators & Its Application,” *Schneider Electric*, Jun. 03, 2024. https://eshop.se.com/in/blog/post/all-you-need-to-know-about-voltage-regulators-its-application.html

[78] *Monolithicpower.com*, 2024. https://www.monolithicpower.com/en/learning/resources/voltage-regulator-types

[79] "Electronic Circuits - Regulators - Tutorialspoint," *Tutorialspoint.com*, 2019. https://www.tutorialspoint.com/electronic_circuits/electronic_circuits_regulators.htm

[80] "Switching Regulator | Analog Devices," *Analog.com*, 2024. https://www.analog.com/en/resources/glossary/switching_regulator.html

[81] Prashant digital, "All You Need To Know About Voltage Regulators & Its Application," *Schneider Electric*, Jun. 03, 2024. https://eshop.se.com/in/blog/post/all-you-need-to-know-about-voltage-regulators-its-application.html

[82] "Forward Converter," *Sunpower UK*. https://www.sunpower-uk.com/glossary/what-is-a-forward-converter/

[83] "Introduction to Linear Voltage Regulators," *DigiKey*, 2016. https://www.digikey.com/en/maker/tutorials/2016/introduction-to-linear-voltage-regulators

[84] eeeguide, "Switching Regulator Operation | Operation of Regulator," *EEEGUIDE.COM*, Mar. 23, 2019. https://www.eeeguide.com/switching-regulator-operation/(accessed Oct. 02, 2024).

[85] D. Now, "LD1117 Datasheet - Download Now | Arrow.com," *Arrow.com*, 2024. https://www.arrow.com/en/datasheets/9086562009/stmicroelectronics/ld1117v33 (accessed Oct. 02, 2024).

[86] "HT73XX-A series LDO linear regulator." Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/5057/HT73xx-A.pdf

[87] "et 1A LDO Voltage Regulator DESCRIPTION." Accessed: Oct. 02, 2024. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6128/AMS1117-5272.pdf

[88] Linear, "LD1117V33," *Arrow.com*, 2024. https://www.arrow.com/en/products/ld1117v33/stmicroelectronics?gad_source=1&gbraid=0AAAADNkUrqlB2l31hvCDAnFy8ewn1rKJ&gclid=Cj0KCQjwxsm3BhDrARIsAMtVz6OfDxgmLgE3NOfhlG5yr7KYfadHN5vblqyzwbxhf7N-qM_VvlC1QjAaAu6dEALw_wcB&gclsrc=aw.ds (accessed Oct. 02, 2024).

[89] "HT7333-A | DigiKey Electronics," *DigiKey Electronics*, 2024. https://www.digikey.com/en/products/detail/umw/HT7333-A/17635230?s=N4IgTCBcDaIBYBcDsBmNIC6BfIA (accessed Oct. 02, 2024).

[90] "AMS1117-3.3 | DigiKey Electronics," *DigiKey Electronics*, 2024. https://www.digikey.com/en/products/detail/evvo/AMS1117-3-3/22482148 (accessed Oct. 02, 2024).

[91] A. Pieters, "ESP32 - LDO Voltage Regulators -," *Studiopieters.nl*, Aug. 2023. https://www.studiopieters.nl/voltage/ (accessed Oct. 02, 2024).

[92]J. Johnson, "What Are Relays and Why Are They So Important For Applications?," *Amperite*, Apr. 16, 2019. https://amperite.com/blog/relays/

[93]B. G. says, "Different Types Of Relays, Their Construction, Operation & Applications," *ELECTRICAL TECHNOLOGY*, Dec. 25, 2018. https://www.electricaltechnology.org/2018/12/what-is-relay-different-types-of-relays-its-operation-applications.html

[94]"How does an electromechanical relay work? - Sealevel," *Sealevel*, Nov. 08, 2023. https://www.sealevel.com/support/electromechanical-relay/ (accessed Oct. 25, 2024).

[95]"Electromechanical Relays: Working Principle," *www.linkedin.com*. https://www.linkedin.com/pulse/electromechanical-relays-working-principle-world-of-electrical

[96]"Understanding the Basics: What Is a Solid State Relay(SSR)," *BCH Electric Limited*, Sep. 26, 2023. https://bchindia.com/blogs/understanding-the-basics-what-is-a-solid-state-relay-ssr/ (accessed Oct. 25, 2024).

[97]Impulse Insight, "The market for 'Electromechanical Relays (EMRs) and Solid State Relays (SSRs) Market' is examined in this report, along with the factors that are expected to drive and restrain demand over the projected period. Introduction to Electromechanical Relays (EMRs) and Solid State Relays (SSRs) Market Insi," *Linkedin.com*, Oct. 15, 2024. https://www.linkedin.com/pulse/electromechanical-relays-emrs-solid-state-ssrs-market-qwjwe (accessed Oct. 25, 2024).

[98]"G6E-134P-ST-US DC5," *Digikey* , Oct. 25, 2024. https://www.digikey.com/en/products/detail/omron-electronics-inc-emc-div/G6E-134P-ST-US-DC5/127440?utm_adgroup=Relays&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=&utm_content=Relays&utm_id=go_cmp-120565755_adg-7526266395_ad-665604606662_dsa-51002747703_dev-c_ext-_prd-_sig-CjwKCAjw68K4BhAuEiwAylp3kuCghEvcKAbXDt3dYKN_rvyktY0fjei6Fxi3KhNh1mIrCXh9a7XLdRoCn2AQAvD_BwE&gad_source=1&gclid=CjwKCAjw68K4BhAuEiwAylp3kuCghEvcKAbXDt3dYKN_rvyktY0fjei6Fxi3KhNh1mIrCXh9a7XLdRoCn2AQAvD_BwE

[99]"5Pcs QIANJI JQC-3F(T73)-5VDC 12VDC T73 5V 12V 10A 5Pin Relay Module," *Elecbee Factory*, 2024. https://www.elecbee.com/en-28785-5Pcs-QIANJI-JQC-3F-T73-5VDC-12VDC-T73-5V-12V-10A-5Pin-Relay-Module (accessed Oct. 25, 2024).

[100]“12Pcs 5V DC Mini Power Relay Auto PCB Relay SRD-05VDC-SL-C Power Relay 5 Pins DC Power Relay PCB Mount Electric Relay for Universal Household Appliance,” *Amazon* , Oct. 25, 2024. https://www.amazon.com/SRD-05VDC-SL-C-Electric-Universal-Household-Appliance/dp/B098JPM9S7/ref=asc_df_B098JPM9S7/?tag=hyprod-20&linkCode=df0&hvadid=693493810553&hvpos=&hvnetw=g&hvrand=11765140123186628286&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9217253&hvtargid=pla-2060378103502&mcid=b138be8c955037309c2b0cc89a33bc5e&th=1

[101]“SRD-05VDC-SL-C datasheet by | SnapMagic Search,” *Snapeda.com*, 2024. https://www.snapeda.com/parts/SRD-05VDC-SL-C/Songle%20Relay/datasheet/

[102] Elprocus, “Buzzer : Working, Types, Circuit, Advantages & Disadvantages,” *ElProCus - Electronic Projects for Engineering Students*, Jul. 26, 2021. https://www.elprocus.com/buzzer-working-applications/

[103] “Piezo Buzzers vs. Magnetic Buzzers — What’s the Difference?,” *APC International Ltd.*, Nov. 12, 2020. https://www.americanpiezo.com/blog/piezo-buzzers-vs-magnetic-buzzers/

[104] “9mm Electromagnetic Buzzer: 40Ω, 4-6V, Top Opening,” *Pololu.com*, 2024. https://www.pololu.com/product/1485 (accessed Oct. 19, 2024).

[105] https://www.amazon.com/QMseller-Terminals-Raspberry-Electronic-Continous/dp/B07VRK7ZPF/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.OyPyJJ4xtWru7f8xcsfDi7R03NiW0MPkZStq2OCgkR2j9WFm-yJSzAoFloncE9WbvY0gw9FOk5Qcpp7YONZXPlCDr1P9fb3a3QeoP8AavYd4FAU6DZ0yU_NHtiPO1GEUxliQ7F5Kldeetxi_EwOjq1XNeDSH3RHgb7_ViscKwCeQzPP63qvtSQ65dmzo_qdLJx1M3f7Dv9XndURxJepK8AN97lgQUDYCqazs00nZLzU._5wcV1Zig7QJwjjfmCqGKIT5qFI4VCIaSvCSAO8raVc&dib_tag=se&keywords=arduino+piezo+buzzer&qid=1729376657&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1

[106] https://cdn-shop.adafruit.com/product-files/1536/C1888-001_datasheet.pdf

[107] “Learn About LED Lighting,” *ENERGY STAR*, 2024. https://www.energystar.gov/products/light_bulbs/learn-about-led-lighting#basics

[108] D. Upadhyay, “Types of LED Lights: A Comprehensive Guide to LED Lighting - TechTOnions.com,” *TechTOnions.com*, Jul. 29, 2023. https://www.techtonions.com/leds-a-complete-guide-to-types-and-benefits/ (accessed Oct. 20, 2024).

[109] “How do RGB LEDs work? | Random Nerd Tutorials,” *Random Nerd Tutorials*, Feb. 02, 2019. https://randomnerdtutorials.com/electronics-basics-how-do-rgb-leds-work/

[110] “Shenzhen Fedy Technology Co., LTD FD-5WSRGB-A.” Available: https://cdn-shop.adafruit.com/datasheets/FD-5WSRGB-A.PDF

[111] DIYables.io, “DIYables RGB LED Module for Arduino, ESP32, ESP8266, Raspberry Pi,” *DIYables Products*, 2022. https://diyables.io/products/rgb-led-module (accessed Oct. 21, 2024).

[112] https://www.amazon.com/Tricolor-Multicolor-Lighting-Electronics-Components/dp/B01C19ENDM/ref=sr_1_1_sspa?crid=AXOP2Y97NA47&dib=eyJ2IjoiMSJ9.Y8GpEVWyzQO2l20uhW6fixYsJCh2SuB4Iqa20dQnaR9nyGaGulbM-y02uuGOI_kotm7SQ4EDJCLuZhHhgtStpJSrn384CiK4iJ8AEXuKymN08Ec28JOtbjQlQsdTOF1fdkkrTgWfdPIPjgjVsl-3w5k76kpOLa8dnJi6JcAxV7gyLArV5dOxf3svdp6UY-bXy_Qf5fULXv9dd8P6lb6xneS_DY3SNAGCUxqY3Oo4E209JO4pXl8Dcpai8_JXBWJcvVDfRNsoRxkBXv2fWs2Ny9j0dRSXzcG5zXJxB9PKxMY.-a-Tf497mZxKcToFUmFtn7n2zAc8-BskloEUTPRUrqw&dib_tag=se&keywords=rgb%2Bled&qid=1729475819&sprefix=rgb%2Bled%2Caps%2C136&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1

[113] Website Technology Comparisons https://chatgpt.com/share/6716c5b0-e964-800d-b212-614fa218e37a

[114] Server Comparisons https://chatgpt.com/share/6716c728-2810-800d-a790-93b1269970ad

[115] Firebase Hosting Website https://firebase.google.com/docs/hosting

[116] “Email Protocols,” *GeeksforGeeks*, May 16, 2022. https://www.geeksforgeeks.org/email-protocols/

[117] ChatGPT, "Software Senior Design," OpenAI. Available: https://chatgpt.com/share/66f6fcd9-2cf0-8008-915b-5f3dbf8c1e02. [Accessed: Sep. 25, 2024].

[118] [1]“Email Protocols,” *GeeksforGeeks*, May 16, 2022. https://www.geeksforgeeks.org/email-protocols/

[119] GeeksForGeeks, “Simple Mail Transfer Protocol (SMTP) - GeeksforGeeks,” *GeeksforGeeks*, Apr. 23, 2019. https://www.geeksforgeeks.org/simple-mail-transfer-protocol-smtp/

[120] ChatGPT, "Comparison of Email Protocols and Platforms for Smart Safe Project," OpenAI. Available: https://chatgpt.com/c/66e70a19-5c0c-8008-8c17-71d91b11e6c4. [Accessed: Oct. 24, 2024].

[121] "IEEE Citation Request," ChatGPT, https://chatgpt.com/c/671a6a0a-9214-8008-815b-4df0de81d6c5 (accessed Oct. 24, 2024).

[122] ChatGPT, "Keypad Technology Table," OpenAI. Available: https://chatgpt.com/share/66fddaff-7ab8-8008-93cc-8c47f548ab90. [Accessed: Oct. 24, 2024].

[123] ChatGPT, "Keypad Technology Comparison," OpenAI. Available: https://chatgpt.com/share/66e70964-324c-8008-967e-d147addbb6e5. [Accessed: Sep. 25, 2024].

[124] ChatGPT, "Software Senior Design," OpenAI. Available: https://chatgpt.com/share/66f6fcd9-2cf0-8008-915b-5f3dbf8c1e02. [Accessed: Sep. 25, 2024].

[125] "IEEE Citation Generator," MyBib. https://www.mybib.com/tools/ieee-citation-generator (accessed Oct. 24, 2024).

[126] GeeksforGeeks, "What is Git?," *GeeksforGeeks*, May 17, 2024. https://www.geeksforgeeks.org/what-is-git/

[127] "Introduction to Github," *GeeksforGeeks*, Sep. 17, 2015. https://www.geeksforgeeks.org/introduction-to-github/

[128] "EAGLE (program)," *Wikipedia*, Oct. 29, 2021. https://en.wikipedia.org/wiki/EAGLE_(program)

[129] "Code Composer Studio," *Wikipedia*, Apr. 26, 2022. https://en.wikipedia.org/wiki/Code_Composer_Studio

[130] R. Dickson, "Linear actuators 101 - How does an actuator work | FIRGELLI," *Firgelli Automations*, Nov. 16, 2018. https://www.firgelliauto.com/blogs/actuators/linear-actuators-101?srsltid=AfmBOopMTaztfAoVhonkLhrwPNXlHCyWMn5Wceb4CRhQnu2jNXuct9cg (accessed Oct. 25, 2024).

[131] ESP32IO.com, "ESP32 - Actuator | ESP32 Tutorial," *ESP32 Tutorial*, 2024. https://esp32io.com/tutorials/esp32-actuator (accessed Oct. 25, 2024).

[132]P. Hollis, "Guide to Working of Electric Linear Actuators," *Venture Mfg. Co.*, Dec. 05, 2017. https://www.venturemfgco.com/blog/guide-working-electric-linear-actuators/

[133]"Stepper Motors: Types, Uses and Working Principle | Article | MPS," *Monolithicpower.com*, 2024. https://www.monolithicpower.com/en/learning/resources/stepper-motors-basics-types-uses?srsltid=AfmBOorTx5ahRg7xS9_C7OOfws9eKZM9Eiqg1gD08xyJJsDGJFmmz3cb (accessed Oct. 25, 2024).

[134]"Solenoids - Definition, Electromagnets, Types of Solenoids, Applications, Videos, and FAQs," *BYJUS*. https://byjus.com/physics/solenoids/

[135]M. Mancini, "How Solenoids Work," *HowStuffWorks*, Jan. 13, 2021. https://science.howstuffworks.com/solenoid.htm

[136]"DC12V Round Tongue Mini Electric Lock," *elecrow.openhardware*, Sep. 09, 2016. https://www.elecrow.com/ly-01-dc12v-round-tongue-mini-electric-

lock.html?srsltid=AfmBOorwk4QpdsxN89eqxMsgEposaojmCdht7ewGJdoenTrbF_cKqSUK (accessed Oct. 25, 2024).

[137]“dfrobot.com,” *Dfrobot.com*, 2023. https://www.dfrobot.com/product-1891.html?search=FIT0624&description=true (accessed Oct. 25, 2024).

[138]“Buy Cheap 1240L-12E07 DC 12V 20.6W 1kg Holding Solenoid for Electric Door Lock,” *Uxcell.com*, 2019. https://www.uxcell.com/1240l12e07-12v-206w-1kg-holding-solenoid-for-electric-door-lock-p-301304.html (accessed Oct. 25, 2024).

[139]K. Mcdonald, “Group 39 EEL 4915 i Final Documentation SMOCK Lock: A Smart and Secure Lock Dr. Lei Wei Senior Design II Group 39.” Accessed: Oct. 24, 2024. [Online]. Available: https://www.ece.ucf.edu/seniordesign/fa2021sp2022/g39/Final%20Project%20Documentation.pdf

[140] “BIOMETRIC STANDARDS Agenda.” Available: https://egovstandards.gov.in/sites/default/files/2023-05/Biometric%20Standards.pdf

[141] *Google.com*, 2024. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.hsdl.org/c/view%3Fdocid%3D464492&ved=2ahUKEwiZ3NK-voKJAxXQTDABHdp9NccQFnoECBIQAQ&usg=AOvVaw3pjNI4e55JR_r_w8NTD1Cw (accessed Oct. 09, 2024).

[142] K. Mangold, “Data Format for the Interchange of Fingerprint, Facial and Other Biometric Information ANSI/NIST-ITL 1-2011 NIST Special Publication 500-290 Edition 3,” Aug. 2016, doi: https://doi.org/10.6028/nist.sp.500-290e3.

[143] MCL, “IPC Standards for Printed Circuit Boards | PCB Design,” *mcl*, Apr. 2019. https://www.mclpcb.com/blog/ipc-standards-for-pcbs/#ipc (accessed Oct. 22, 2024).

[144] T. S. Team, “IPC-2221 Standards in PCB Design,” *Sierra Circuits*, Jul. 20, 2021. https://www.protoexpress.com/blog/ipc-2221-circuit-board-design/

[145] “IPC-2221A Generic Standard on Printed Board Design,” 2003. Available: https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf

[146]“Standard | IECEE,” *www.iecee.org*. https://www.iecee.org/certification/iec-standards/iec-62368-12018

[147] OpenAI, “ChatGPT,” *ChatGPT*, Oct. 22, 2024. https://chatgpt.com/

[148]“ChatGPT - Smart Safe Diagrams Request,” ChatGPT, 2024. https://chatgpt.com/share/6728f754-d160-8008-b5af-871aa71f8a6a (accessed Nov. 26, 2024).

[149] S. Tsentserensky, “How to Make a UML Class Diagram (and Others) With Examples,” *Slickplan*, Oct. 23, 2023. https://slickplan.com/blog/how-to-make-a-uml-diagram

[150] LucidChart, “Online Diagram Software & Visual Solution | Lucidchart,” *Lucidchart.com*, 2017. https://www.lucidchart.com/

[151] Visual Paradigm, “What is Use Case Diagram?,” *Visual-paradigm.com*, 2019. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

[152] D. Kaufman, “Your Guide to PCB Material Types,” *Cirexx International*, Dec. 15, 2020. https://www.cirexx.com/pcb-material-types/

[153] RayMing, “11 Types of Printed Circuit Board Raw Material Specification - RAYPCB,” *RAYPCB*, Feb. 10, 2018. https://www.raypcb.com/11-types-of-pcb-circuit-board-material-specification/ (accessed Nov. 26, 2024).

[154] “PCB Trace: Managing Copper Pads, Traces, & Planes in High-Load Applications,” *San Francisco Circuits*, 2024. https://www.sfcircuits.com/pcb-school/pcb-trace-widths-high-load-applications (accessed Nov. 26, 2024).

[155] T. S. Team, “What is a Pad in PCB Design and Development,” *Sierra Circuits*, Mar. 16, 2021. https://www.protoexpress.com/blog/what-is-pad-pcb-design-development/

[156] “HASL Surface Finish | Sierra Circuits,” *Sierra Circuits*, Jan. 23, 2024. https://www.protoexpress.com/kb/hasl-surface-finish/ (accessed Nov. 26, 2024).

[157] U. Waseem, “PCB Material: A Comprehensive Guide to Understanding and Choosing the Right Materials,” *www.wevolver.com*. https://www.wevolver.com/article/pcb-material

[158] “ESP32 Send Emails using SMTP Server: HTML, Text, Attachments (Arduino IDE) | Random Nerd Tutorials,” Jun. 08, 2021. https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/

[159] Instructables, “Implementing Web Server on ESP32,” *Instructables*, Dec. 04, 2023. https://www.instructables.com/Implementing-Web-Server-on-ESP32/ (accessed Nov. 26, 2024).

[160]“ESP-Mail client setup,” ChatGPT, 2024. https://chatgpt.com/share/6736329c-0844-8008-b9b7-7d2178720b38 (accessed Nov. 20, 2024).

[161] “Free and customizable desktop prototype templates | Canva,” Canva, 2024. https://www.canva.com/prototypes/templates/desktop/ (accessed Nov. 26, 2024).

[162] K. Magdy, “ESP32 Keypad Matrix (Kaypad-LCD Example) - Arduino IDE,” DeepBlue, Mar. 28, 2023. https://deepbluembedded.com/esp32-keypad-lcd-example-arduino/

[163] “I2C LCD with ESP32 on Arduino IDE - ESP8266 compatible | Random Nerd Tutorials,” Feb. 01, 2019. https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/

Appendix E – ChatGPT prompts and outcomes

ChatGPT was used to check grammar and wording for writing, helping us in research and guiding us with ideas and solutions. It helped us with building initial tables, which we have evaluated and referenced to create our own tables. It helped us proofread and refine our writing, and provide helpful feedback. Anything that has been referenced or copied from ChatGPT has been cited in our writing and the prompts have been cited in references.

Case Study 1: In my use case diagram, should I put unlock safe then connect that to the fingerprint scanner and keypad or the other way around, also can I use the include arrow?

In your use case diagram, it's best to place "Unlock Safe" as the primary use case and connect "Use Fingerprint Scanner" and "Use Keypad" to it, since "Unlock Safe" is the overarching action the user is trying to achieve, and both "Use Fingerprint Scanner" and "Use Keypad" are ways to accomplish it. This approach keeps the diagram focused on "Unlock Safe" as the main action and shows the fingerprint scanner and keypad as alternatives or optional steps to reach that goal.

Suggested Structure

1. Unlock Safe (Main use case)
    - Connect "Use Fingerprint Scanner" and "Use Keypad" as <<include>> relationships under "Unlock Safe".

    - This approach makes it clear that "Use Fingerprint Scanner" and "Use Keypad" are both necessary components (one or the other) of the "Unlock Safe" action.

Explanation of <<include>> in Your Diagram

- Use <<include>> between "Unlock Safe" and both "Use Fingerprint Scanner" and "Use Keypad" because <<include>> indicates that either scanning the fingerprint or entering the keypad code is a required action within "Unlock Safe". The <<include>> arrow shows that these actions are part of the unlock process but aren’t optional steps—they’re integral to achieving the main goal of unlocking.

Using <<include>> will keep the diagram clear and emphasize that "Unlock Safe" depends on completing one of the included actions. Let me know if you'd like a more specific diagram structure, and I can help refine it further!

Case Study 2: How is data from the website (aka user's email's address) going to be sent to the microcontroller and then sent to the email of the user

To send data, like a user's email address, from your website to the microcontroller (ESP32, for example), and then forward that data to the user's email, you can follow these steps:

1. Website-to-Microcontroller Communication

- Create a Form on the Website: On your website, create a form where users can input their email address. When the user submits the form, their email address will be sent to your backend server.

- Send Data from the Website to the Microcontroller via HTTP: Use HTTP POST requests from the website to send the email address to the microcontroller's IP address. You would set up a specific endpoint on the microcontroller to receive this data.

  Here's a basic workflow:

  - User submits the form on the website with their email.

  - The website's backend sends an HTTP POST request containing the email address to the microcontroller's IP address (e.g., http://<ESP32_IP>/set-email).

- Microcontroller HTTP Endpoint: Set up an HTTP endpoint on the ESP32 that listens for incoming data from the website. Using a library like ESPAsyncWebServer (for async communication) or ESP32WebServer, the microcontroller can receive and process the email address.

  In my use case diagram, should I put unlock safe then connect that to the fingerprint scanner and keypad or the other way around, also can I use the include arrow?

Case Study 3: If I want to use C++/C and I'm using an ESP32, which software library should I use?

For C/C++ on the ESP32, you can use the following libraries to handle email notifications:

1. ESP-Mail Client Library:

- This library supports both sending emails via SMTP and receiving emails via IMAP. It is well-suited for microcontrollers like the ESP32.

- Features:
  - Allows sending emails with attachments.
  - Supports secure SMTP with TLS/SSL.
  - Easy configuration for Gmail, Outlook, and other services.

2. Arduino ESP32 Mail Client:

   - Specifically designed for ESP32 using the Arduino IDE, this library simplifies SMTP and IMAP integration.

   - Features:
     - Support for basic SMTP and IMAP functionalities.
     - SSL/TLS encryption for secure connections.
     - Compatibility with multiple email providers.

3. MbedTLS:

   - A lower-level library for handling SSL/TLS encryption if you want to build your own secure communication from scratch with SMTP.

   - This would give you more control over the email-sending process but requires more manual configuration.

Among these, ESP-Mail Client Library is a great choice as it is specifically designed for the ESP32, provides comprehensive support for sending emails, and integrates smoothly with platforms like Gmail and Outlook.

Case Study 4: I need to write an intro paragraph for chapter 3 as well. In this chapter, we will be research each of the components in our project and comparing them with other products to choose the most suitable options for our project. Each component will have a technology comparison and part selections along with tables that explain our selections. This is what I have so far. Can you fix it and make it better?

Here's a refined introduction paragraph for Chapter 3 to enhance clarity and alignment with your project's focus on component research and selection:

In this chapter, we will conduct a comprehensive examination of each component within our project, comparing them against alternative products to ensure we select the most

suitable options. For each component, we will provide a technology comparison along with a detailed part selection, supported by tables that outline our reasoning and highlight key features. This approach will ensure that every chosen component meets our project's specifications and contributes to an optimal design for our smart safe system.

This version maintains your original intent while improving flow and readability. Let me know if this fits, or if there's anything more you'd like to add!