LOGBOOK

01/23/25; Start of Project

- Initially I was figuring out how to have a proper logbook that works with git.
- I had started with word doc but I wasn't too sure how to have my comments. The issue I faced was no edits were uploaded to the git. The only thing that was updated on the document was the document title.

01/28/25: Working on two project definitions & Hardware Selection

- Talked about my own project that I may do.
- Real time BP Ring
- Smart Light System
- Group agreed on using the **ESP32** as the main component: built in **WI-FI & BLE**
- **RFID RC522 reader**

01/28/25 to 02/03/25: Design

- I worked on a quick KIcad schematic design
- Included potential power and basic writing
- Played/figured out what components worked and did not work

02/11/25 to 02/18/25 : Parts delivered and Software

- **Esp32** was delivered so we are going to meet and start working on the prototypo
- Suggested which **LCD** to use
- Team worked on **debugging code**
- Determined **power consumption** using **ESP32** sleep mode during idle stage

02/25/25 to 03/10/25 Prototype testing and Refinements

- Working on midterm presentation and preparation for Midterm on Tuesday
- I am focusing on the device/parts slide
- Pictures and description
- Unfortunately after running the **RFID** reader multiple time due to hardware or maybe software issue the reader tested to be defective/unknown
- New RFID has been purchased and was delivered "today" March 10.
- Need to revise **PCB design** to optimize with proper features

Future/Next Step (Copied as determined by Daniel)

- Design and manufacture **PCB for a more compact system**.
- Further optimize **power consumption** for long-term use.
- Adding additional features such as app integration and/or Bluetooth

03/10/25 to 03/14/25 Retesting

- Received new RFID

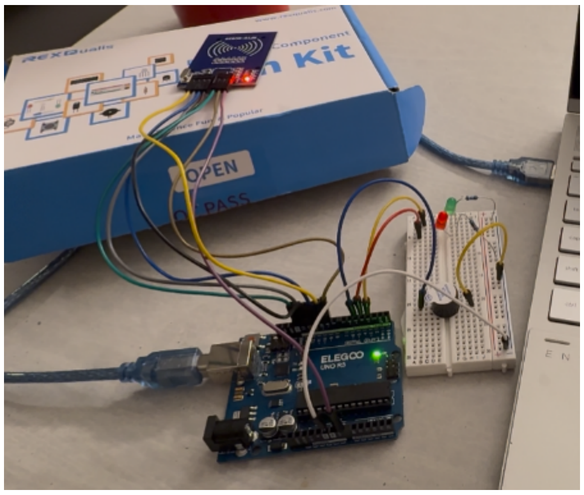
- Upon doing quick searches online, I determined that when using the Arduino we must provide the RFID sensor direct 3v because of the way the Arduino is set up.
- We haven't used the ESP32 yet as we do not want to damage the chip

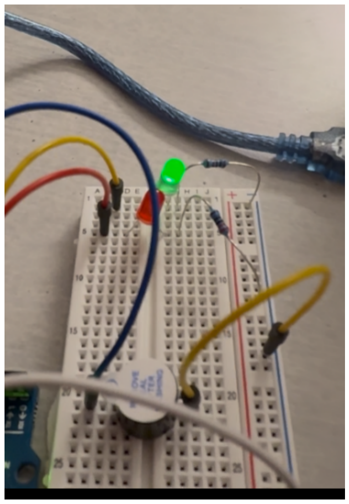
03/14/25-03/22/25 Spring Break

No work was done except team worked on designing the PCB board

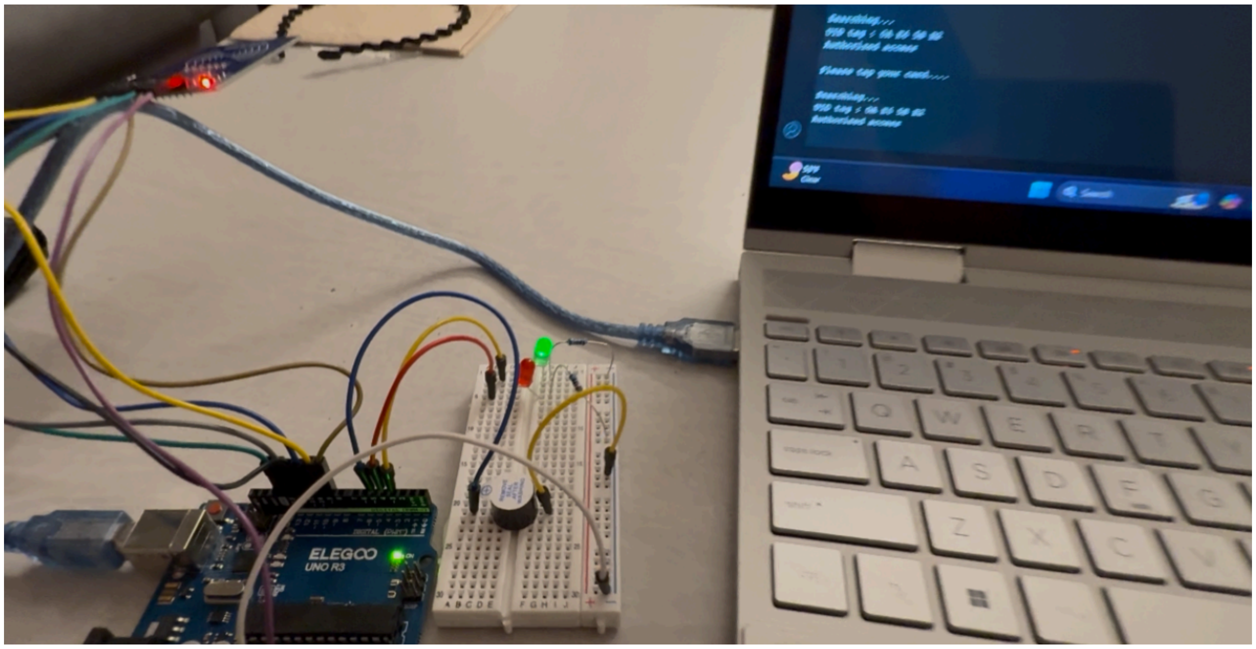
03/24/25-04/04/25 Return to board design

- Submitted design to professor and received feedback or awaiting feedback
- Must redo design with proper connections and make sure there are no errors
- New design was worked on while I attend class lectures and took notes on future development for PCB board
- Team is debating on not printing the PCB as we weren't able to spend time on it


Development using UNO board with RFID


Red and Green LED indication along with buzzer

Currently showing green light or access granted



```
Searching...
UID tag : [illegible]
Authorized access

Please tap your card.....

Searching...
UID tag : [illegible]
Authorized access
```

On screen it indicated the term Authorized access

04/07/25-04/16/25 Team didn't have the time to meet or work on project

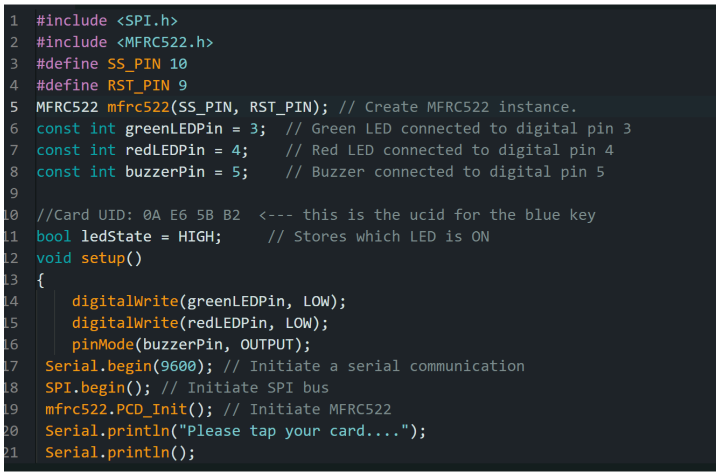
- Managed to edit previous powerpoint
- Update logbook

04/21/25-04/25/25

- Clean up code
- Daniel made last minute changes to the design
- Team started focusing on classwork
- I looked into ESP32 datasheet
- Download the required libraries for the ESP32
- Prepare for presentations
- Took medical time off due to surgery

04/30/25-05/09/25 Cut off date for project

- As of right now we are focusing on finishing up the project
- Revising the codes and making it optimal for the ESP32
- troubleshooting the ESP32 connections
- New wiring for the ESP32 to ensure secure connection
- Determine how to connect the ESP32 on the breadboard
- Final Edits for the Presentation



```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
const int greenLEDPin = 3;  // Green LED connected to digital pin 3
const int redLEDPin = 4;    // Red LED connected to digital pin 4
const int buzzerPin = 5;    // Buzzer connected to digital pin 5

//Card UID: 0A E6 5B B2  <--- this is the ucid for the blue key
bool ledState = HIGH;     // Stores which LED is ON
void setup()
{
    digitalWrite(greenLEDPin, LOW);
    digitalWrite(redLEDPin, LOW);
    pinMode(buzzerPin, OUTPUT);
 Serial.begin(9600); // Initiate a serial communication
 SPI.begin(); // Initiate SPI bus
 mfrc522.PCD_Init(); // Initiate MFRC522
 Serial.println("Please tap your card....");
 Serial.println();
```

Working Code for the UNO must optimize with new library for the ESP32

Additional information from other documents

Reason for Choosing ESP32

The **ESP32** was selected due to its **combination of processing power, built-in connectivity (WiFi + BLE), and GPIO support**. These features are critical for our project, which involves:
✔ **RFID authentication** (SPI interface)
✔ **BLE-based child location tracking**
✔ **Display for status updates**
✔ **Low power consumption for portability**

ESP32 allows **all required functionalities to be implemented without additional communication modules**, simplifying the design and reducing costs.

Items Required to Start Development

Hardware Requirements:

- **ESP-WROOM-32 (ESP32 ESP-32S) Development Board**
- **RFID Module (MFRC522) for child-parent authentication**
- **BLE module (if external testing needed)**
- **Display (TBD)**
- **LEDs and buzzer for feedback**
- **Power Supply (Li-ion battery or 5V DC input)**

Software Requirements:

- **IDE & Compiler:**
    - **Arduino IDE** (with ESP32 Board Manager)
- **Toolchain & Libraries:**
    - **ESP32 core for Arduino**
    - **MFRC522 RFID Library**
    - **BLE Scanner App** (for mobile testing)
- **Hardware/Software Debugging Tools:**
    - **Serial Monitor (Arduino IDE)**
    - **Logic Analyzer (for SPI, I2C debugging)**