



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Classroom Clarity Portable Hub

Team 21: Maddie Donku, Kaitlin Gowens, Jesse Gruber

ECE 445

TA: Aishee Mondal
Professor: Michael Oelze

Team Introduction



Maddie Donku

Electrical Engineering



Kaitlin Gowens

Electrical Engineering



Jesse Gruber

Electrical Engineering





Table of Contents

Introduction

Power Management Subsystem

Control Subsystem

Feedback Subsystem

Mobile App Subsystem

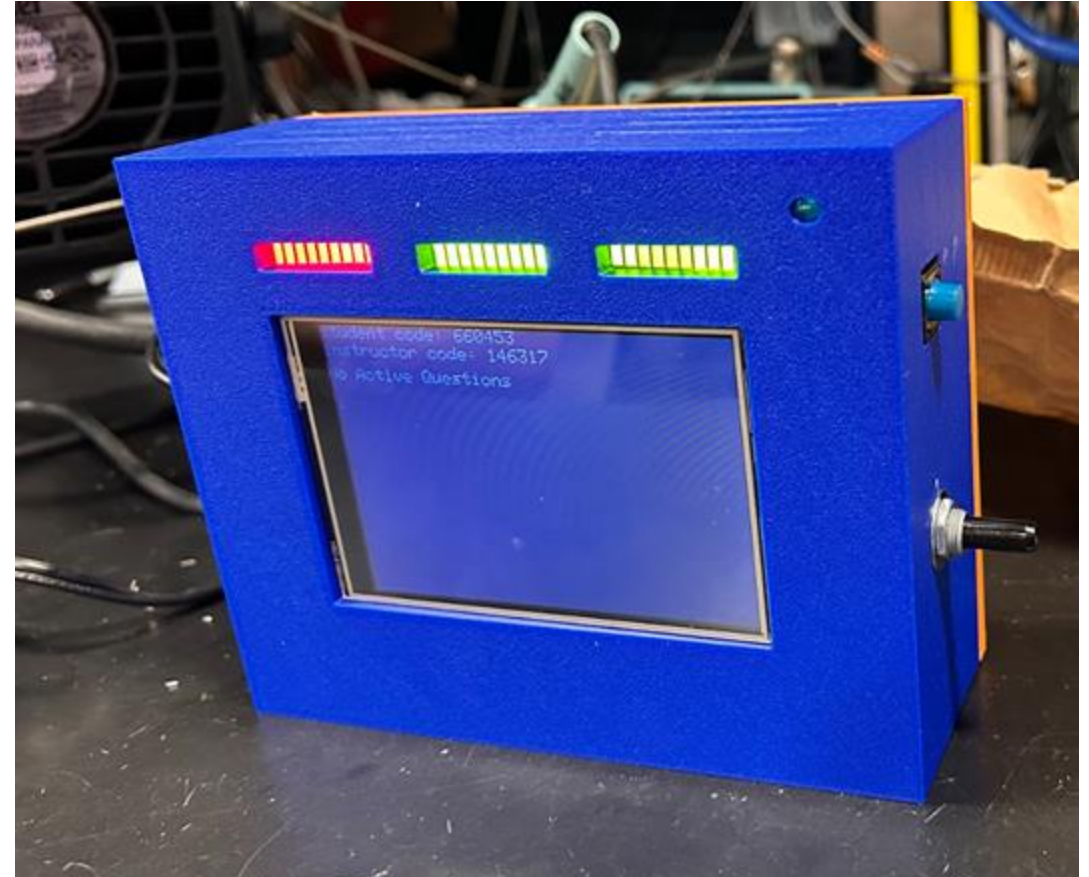
Conclusion

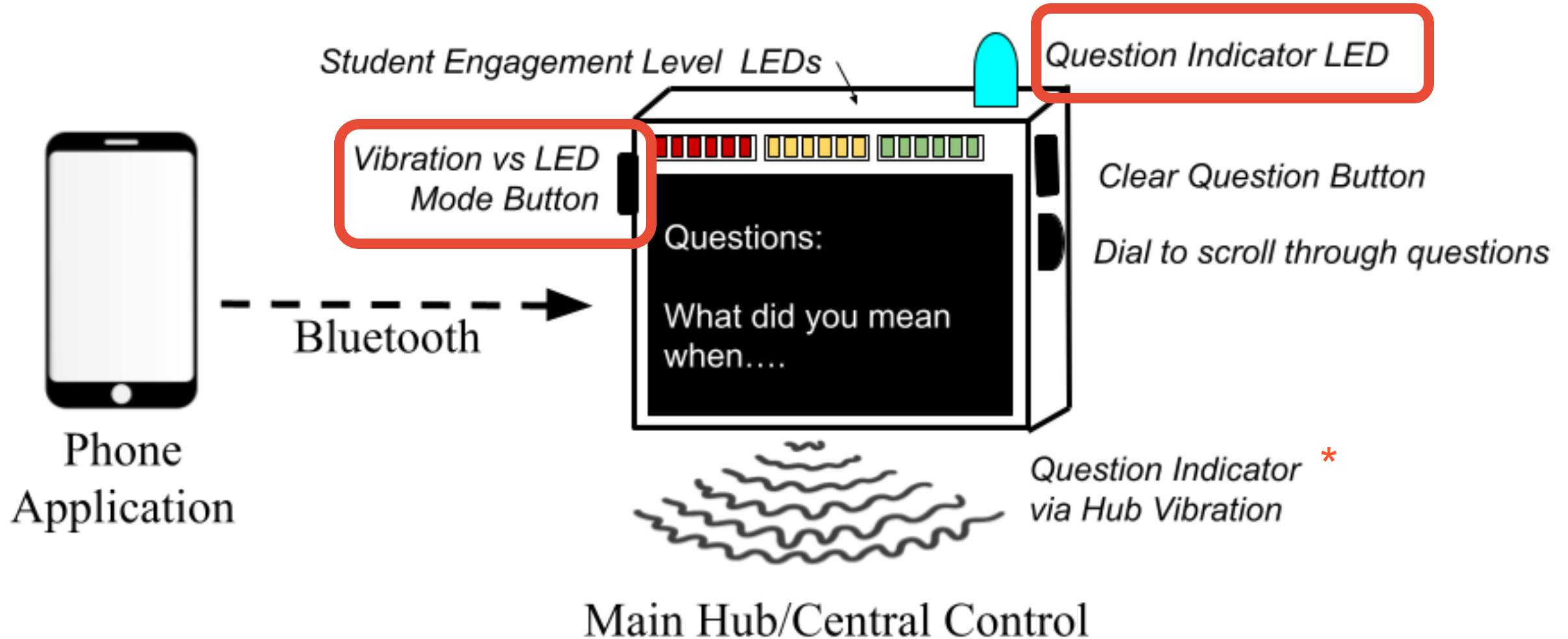
What is Classroom Clarity?

A portable teacher support hub that facilitates communication between students and instructors during a live lecture through LEDs and a text display to share understanding levels and questions.

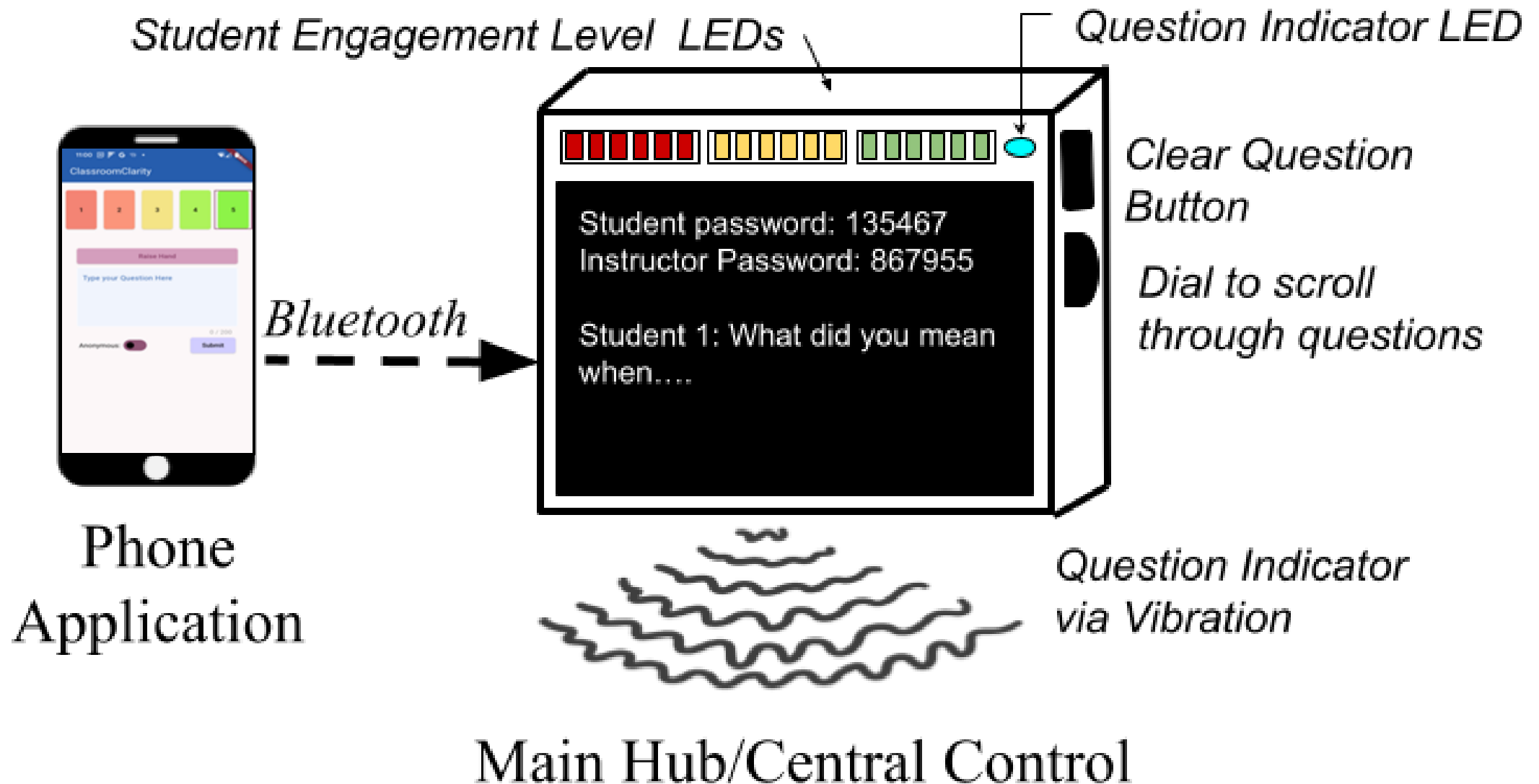
Problems Addressed?

1. Instructors going too fast through difficult material
2. Instructors don't notice questions waiting to be answered
3. Students are too shy to ask a question

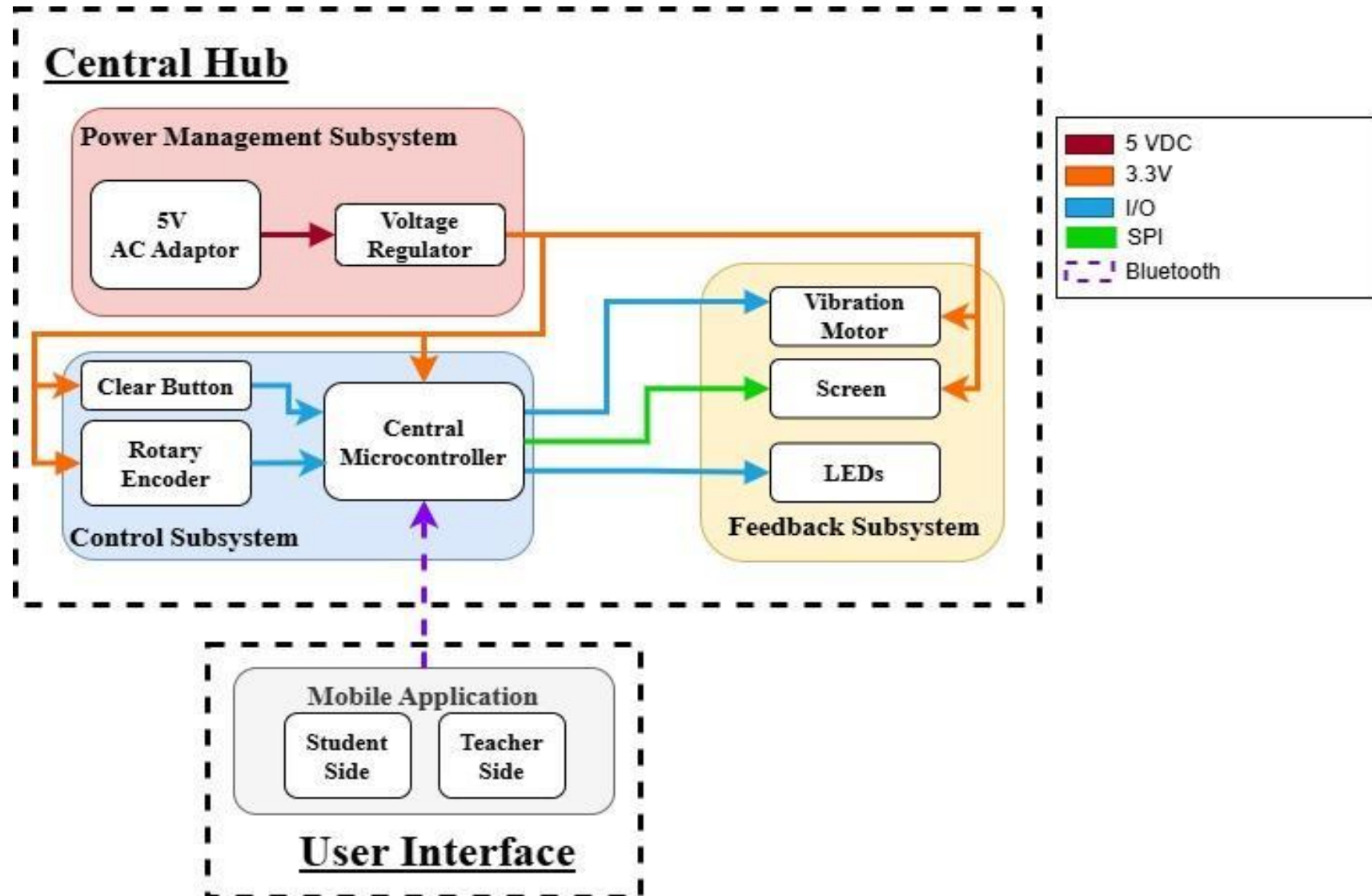




**Note: In RFA, vibration motor was originally in a wearable*

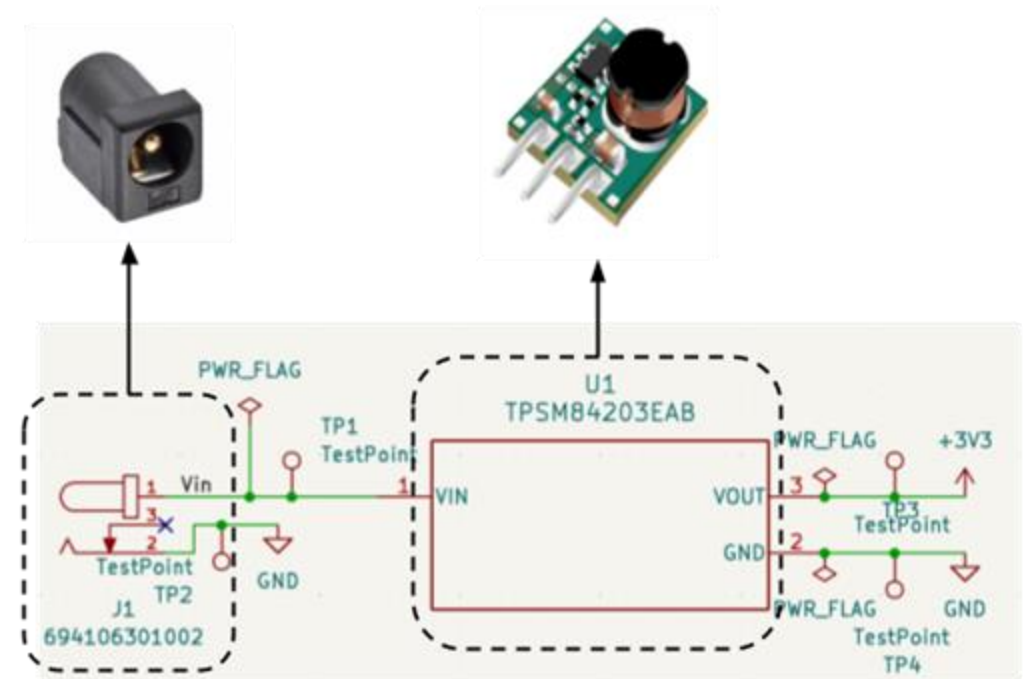
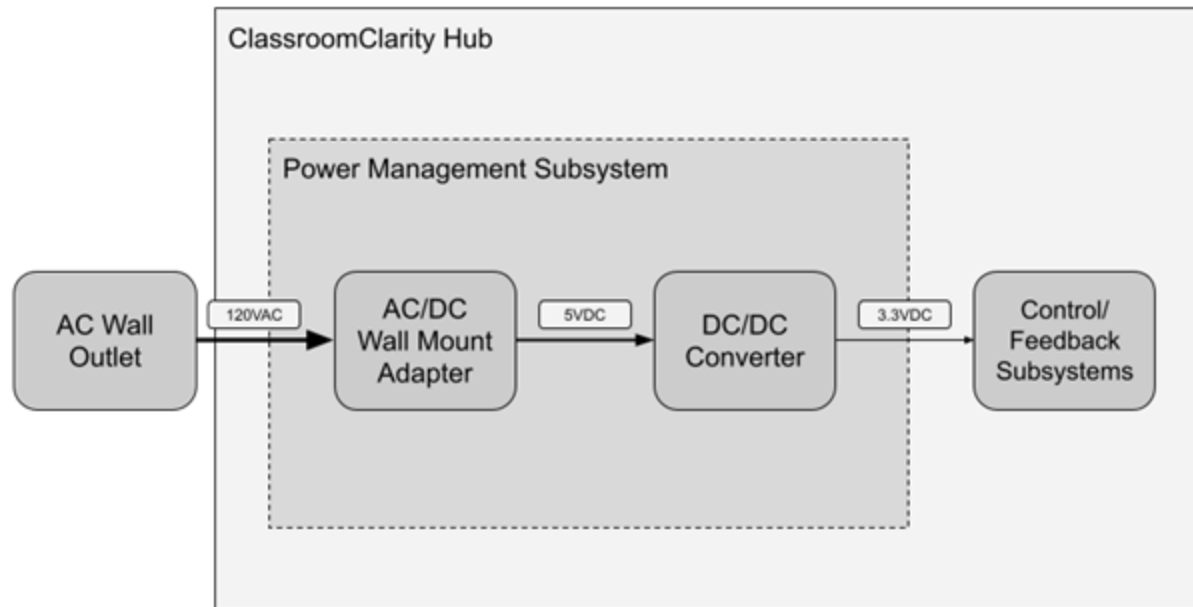


Block Diagram



Power Management Subsystem

How it Works



Requirements and Verification

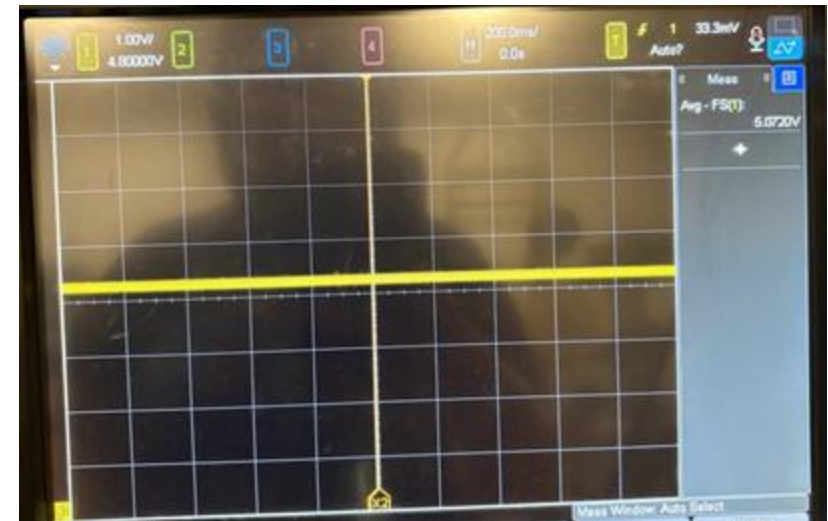
- The AC DC wall adapter must supply at least 1.5A to the hub and **5 V +/- 0.5 V**
- The DC DC converter module must output **3.3 V +/- 0.3 V**

	Voltage [V]
5V Test point (Vin)	5.056
3V3 Test point (Vout)	3.291

Requirements and Verification

- The adapter must be able to **safely provide a stable supply of power** at the very least the length of one lecture (**50 minutes**)
- 43°C $\rightarrow$ limit for safe human touch

	Temperature [$^{\circ}\text{C}$]
Block	24.9
Barrel Jack	33.5
Casing	30.3





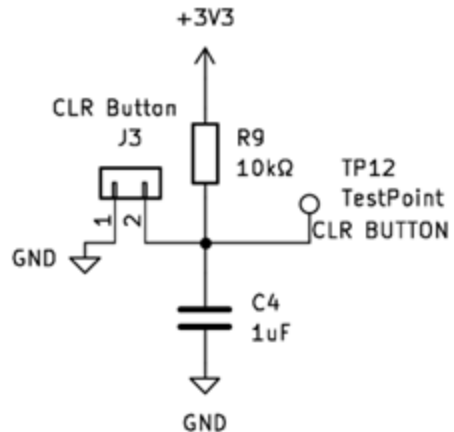
Control Subsystem

Design

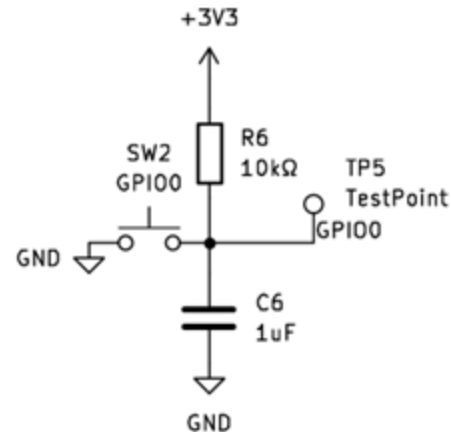
- Debouncing and Pull-up Circuits

$$\tau = R \cdot C$$

Question Clear
Button

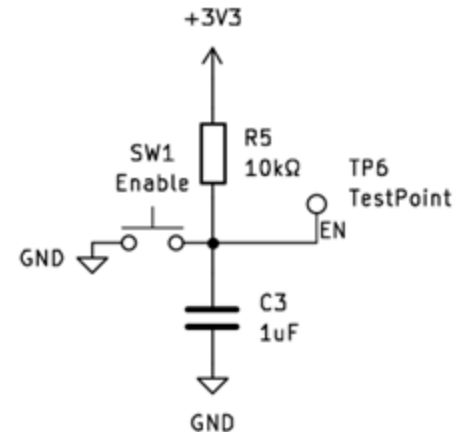


ESP32 Boot
Mode Button

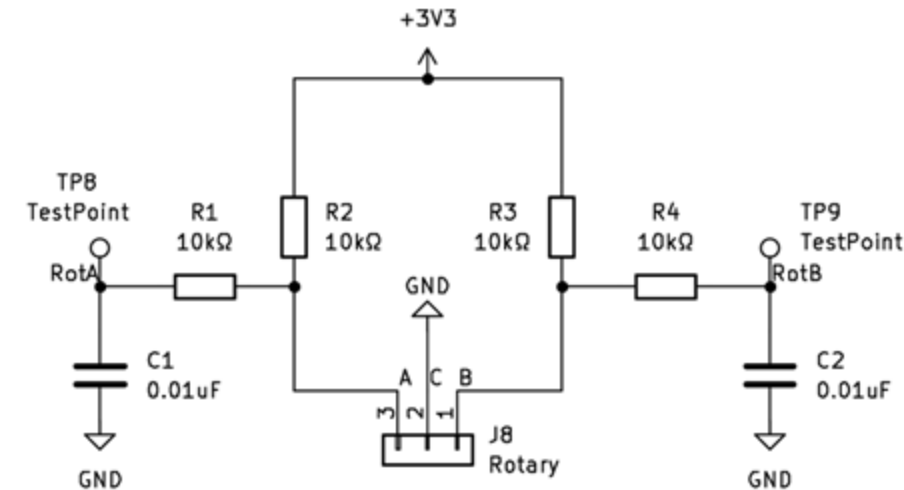


$$\tau = 10 \text{ ms}$$

ESP32 Chip
Enable Button



Rotary Encoder



$$\tau = 0.1 \text{ ms}$$

Requirements and Verification

- The **unpressed input voltage** of each **button GPIO** must fall between **2.5 - 3.6 V**, and the **pressed input voltage** must be between **-0.3 - 0.8 V** within one second of pressing the button.
- The **rotary encoder processed signals** must have a **maximum of 2.5 - 3.6 V** and a **minimum of -0.3 - 0.8 V**.
- The **high output voltage** of the **vibration motor GPIO** pin must be **greater than 2 V** and the **low output voltage** must be **less than 1 V**.

Verifications		
Component	High GPIO [V]	Low GPIO [mV]
Clear Button	3.285	28
Rotary Encoder A Signal	3.264	-80
Rotary Encoder B Signal	3.263	-110
Vibration Motor	2.5	0.185

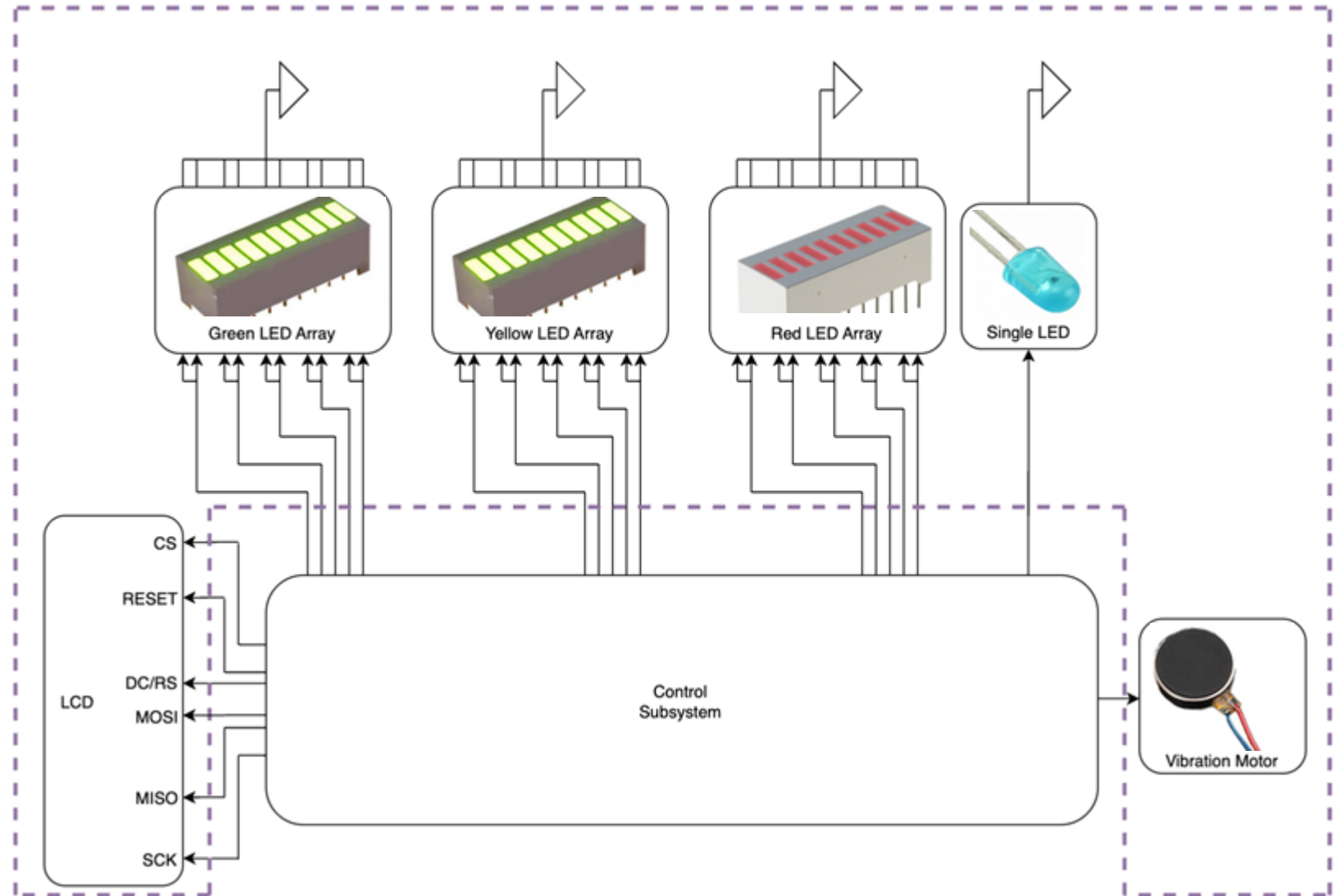
Requirements and Verification

- The **high output voltage** of the green, yellow, and red **LED array GPIO** pins must be **greater than 1.9 V** and the **low output voltage** must be **less than 1.5 V**.

Verifications		
Component	High GPIO [V]	Low GPIO [mV]
Red LED 1	1.95	0
Red LED 2	1.93	0.1
Red LED 3	1.89	0.11
Red LED 4	1.896	0.13
Red LED 5	1.898	0.14
Yellow LED 1	2.113	0.14
Yellow LED 2	2.112	0.14
Yellow LED 3	2.113	0.14
Yellow LED 4	2.092	0.0
Yellow LED 5	2.086	0.13
Green LED 1	2.118	0.09
Green LED 2	2.114	0.09
Green LED 3	2.008	0.09
Green LED 4	2.009	0.09
Green LED 5	2.102	0.1

Feedback Subsystem

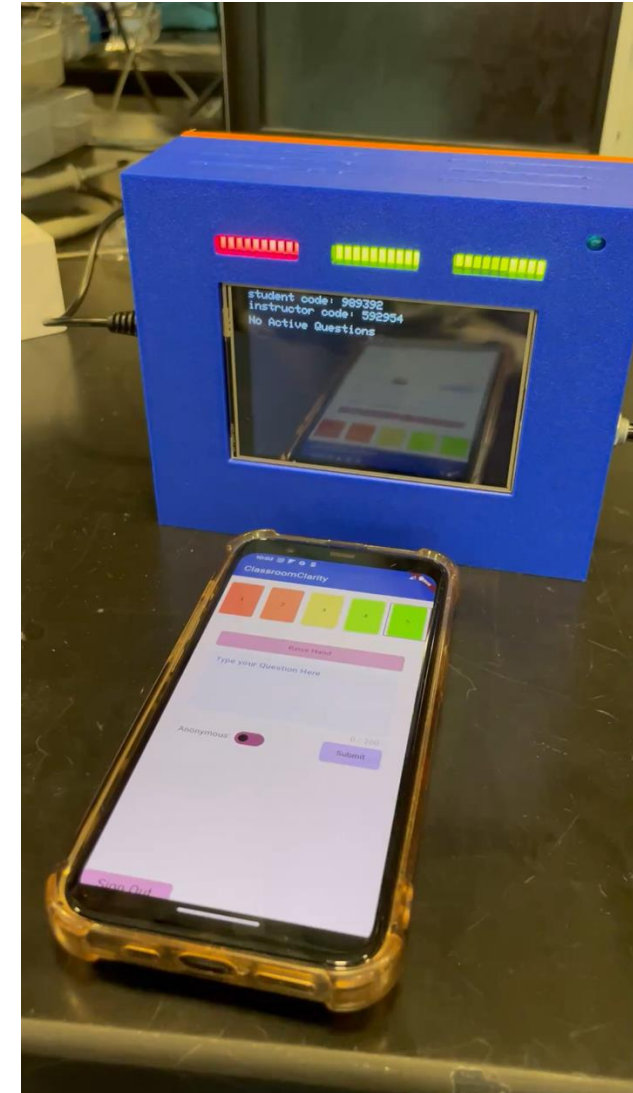
How it Works



Requirements and Verification

- **LED arrays should react to any changes in “Understanding Rating” data sent by the app within 30 seconds.**

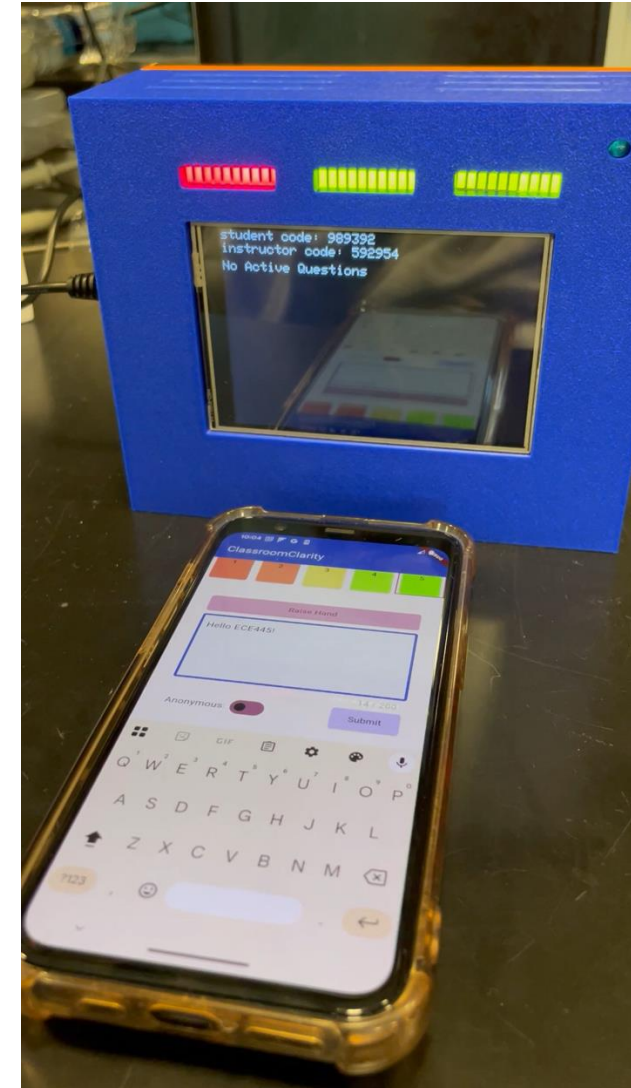
*High Level Requirement



Requirements and Verification

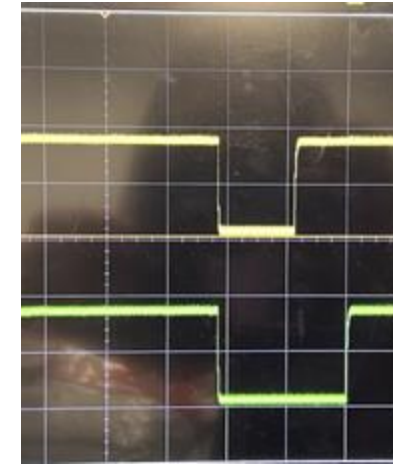
- The **notification LED** should **light up** and **vibration motor** should **vibrate** **within 30 seconds** of a question being sent from the app.

*High Level Requirement

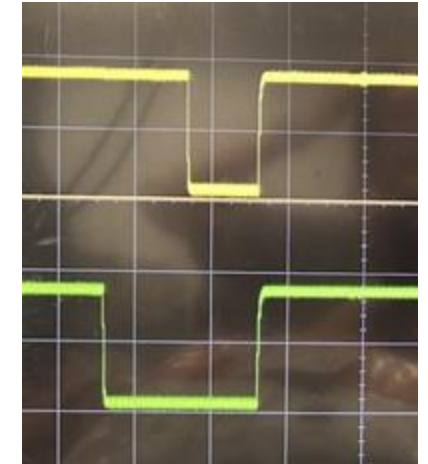


Requirements and Verification

- **Signal A** of the rotary encoder must **lead** **signal B** when turning the encoder **clockwise**. **Signal B** must **lead** **signal A** when turning **counterclockwise**.
- The **LCD** should **display a maximum of 200 characters** with a black background and white font.



a)



b)

Signal A (yellow) and Signal B (green) for
a) clockwise and b) counterclockwise rotation



Requirements and Verification

- Hub should **hold at least 5 questions** and **1 question should be displayed** by the central hub if the queue is not empty.

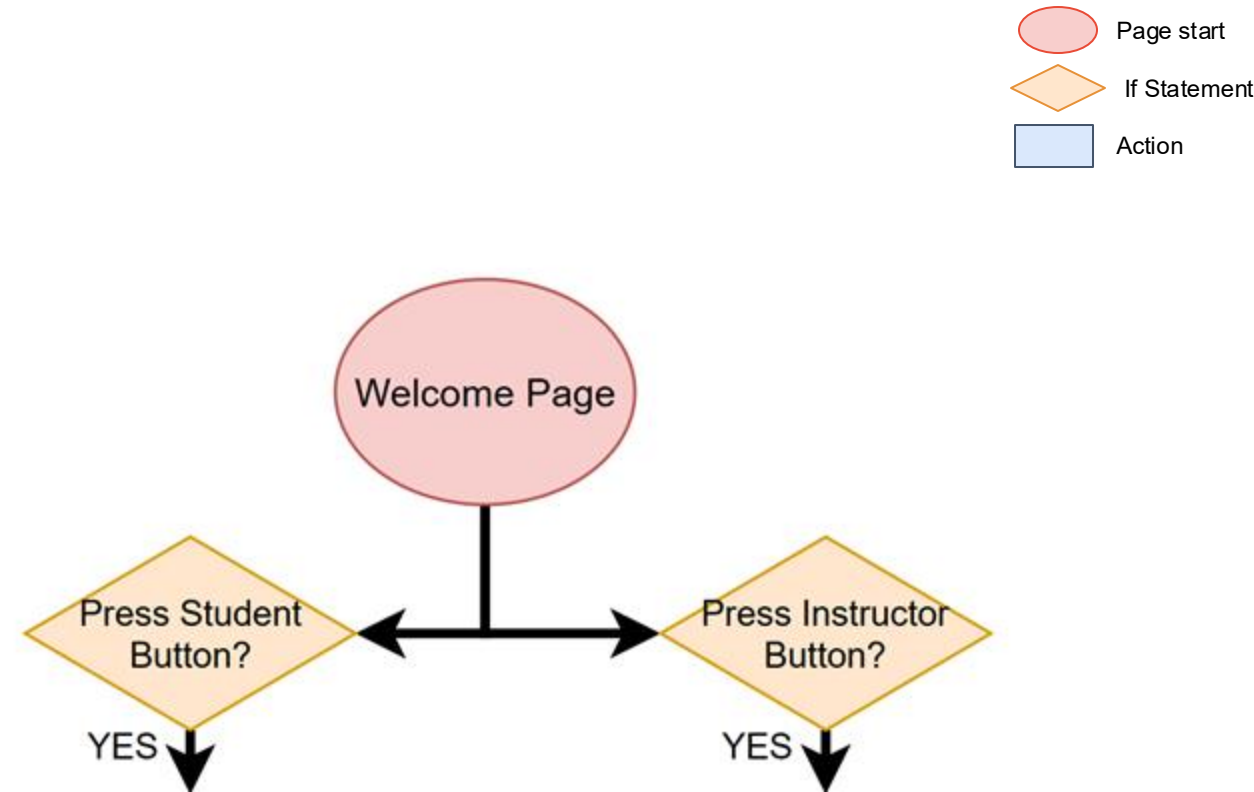
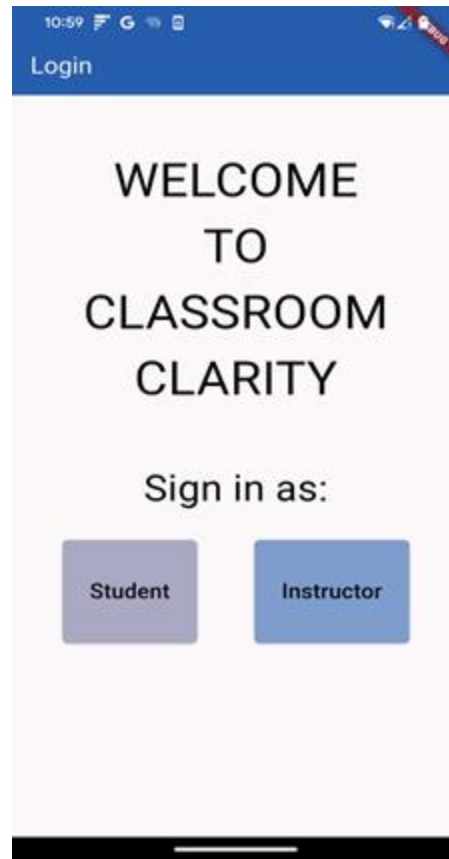
*High Level Requirement



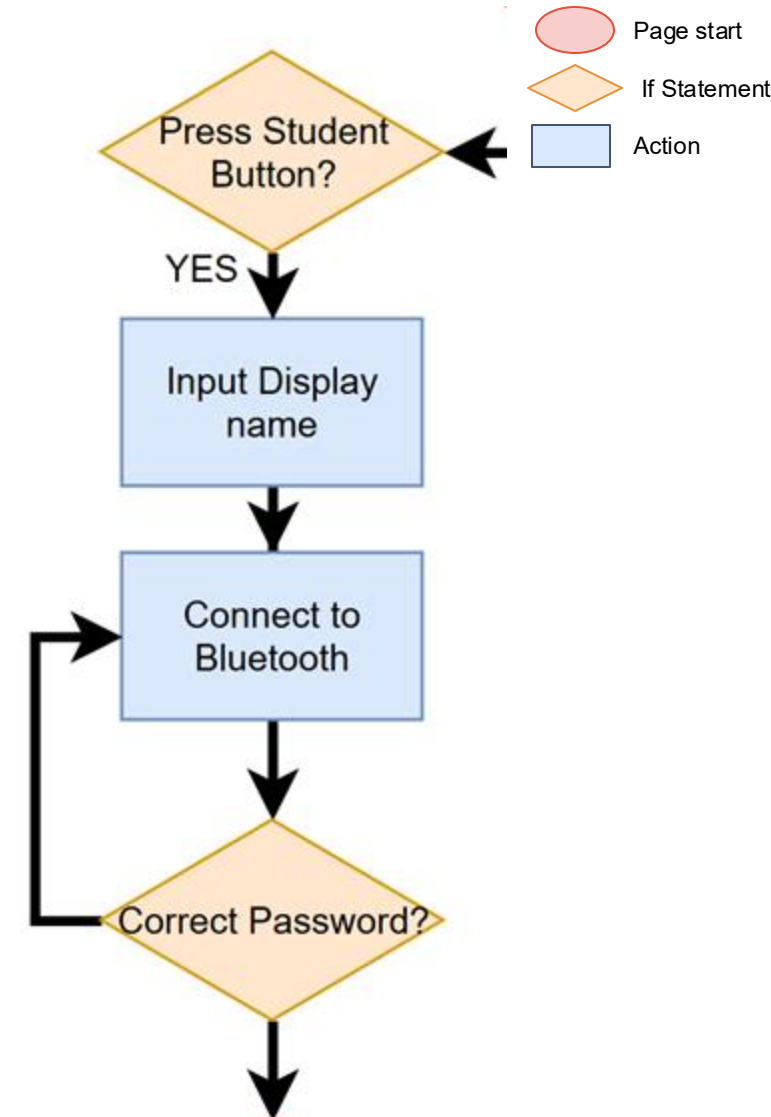
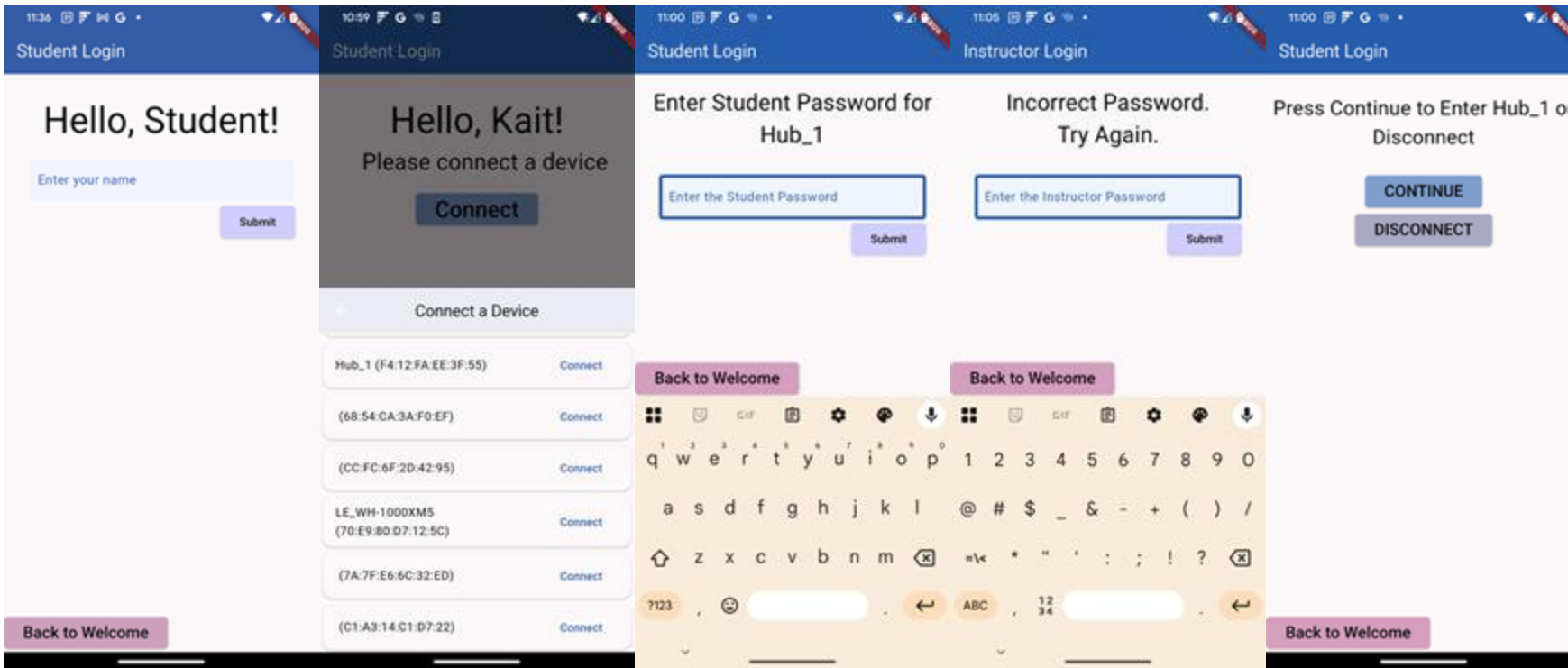


Mobile Application

How it Works

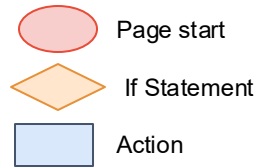
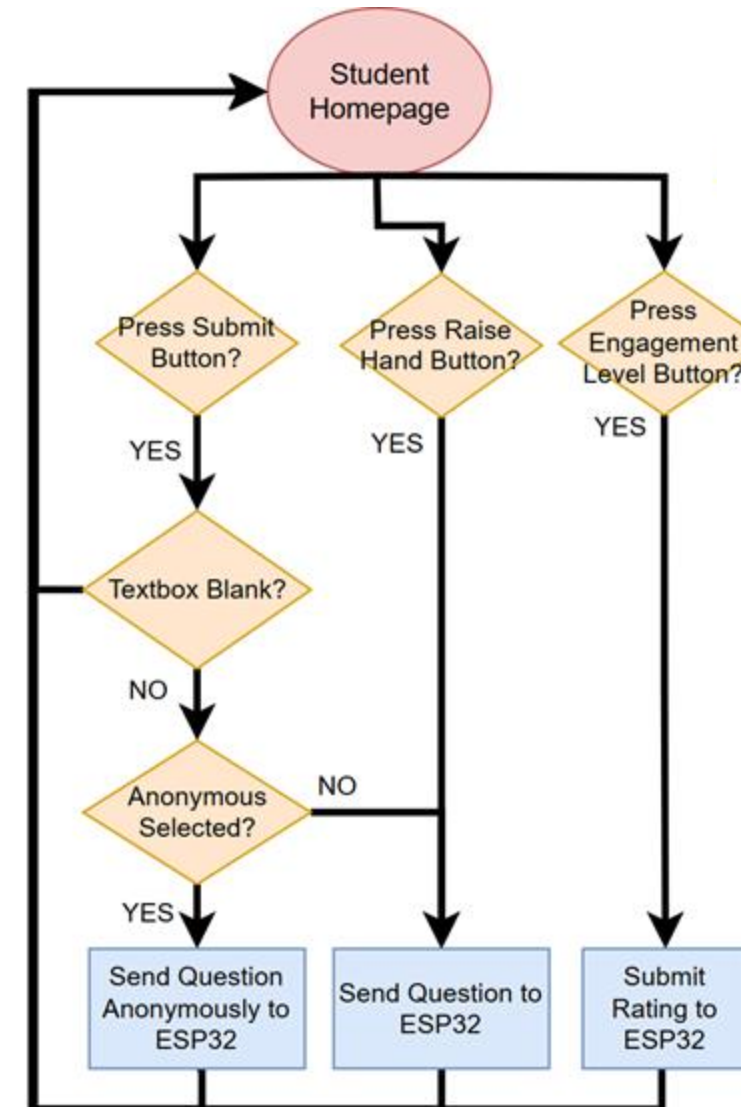


Student Side

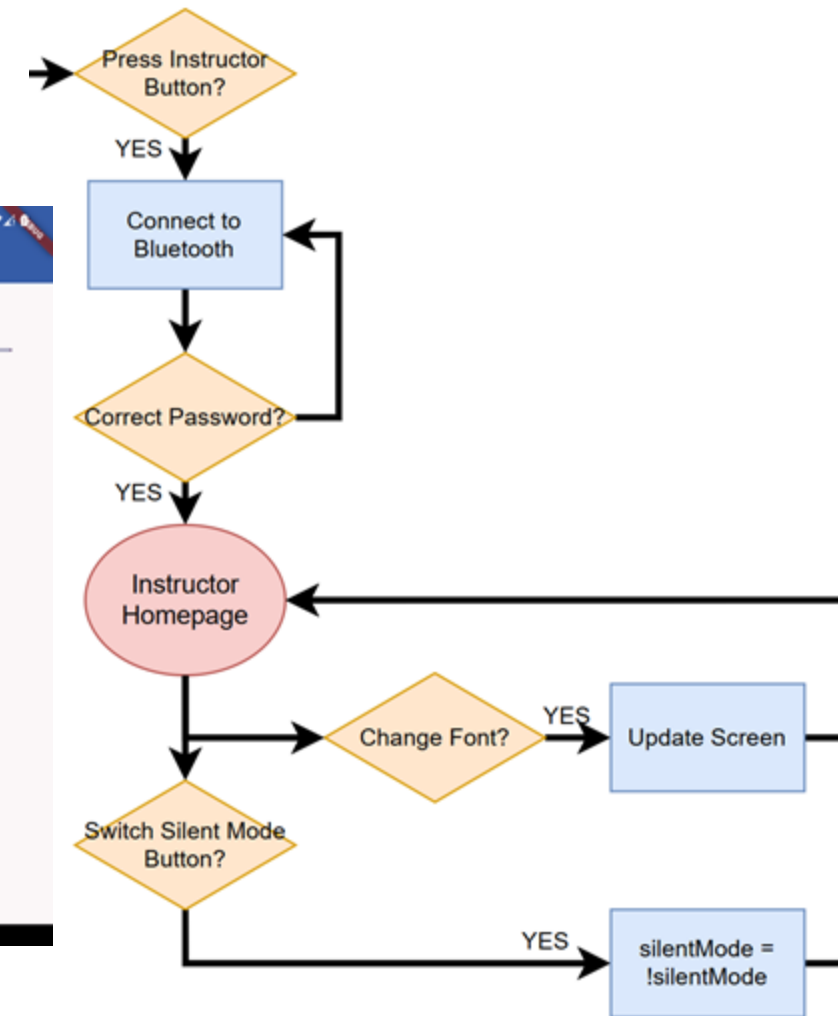
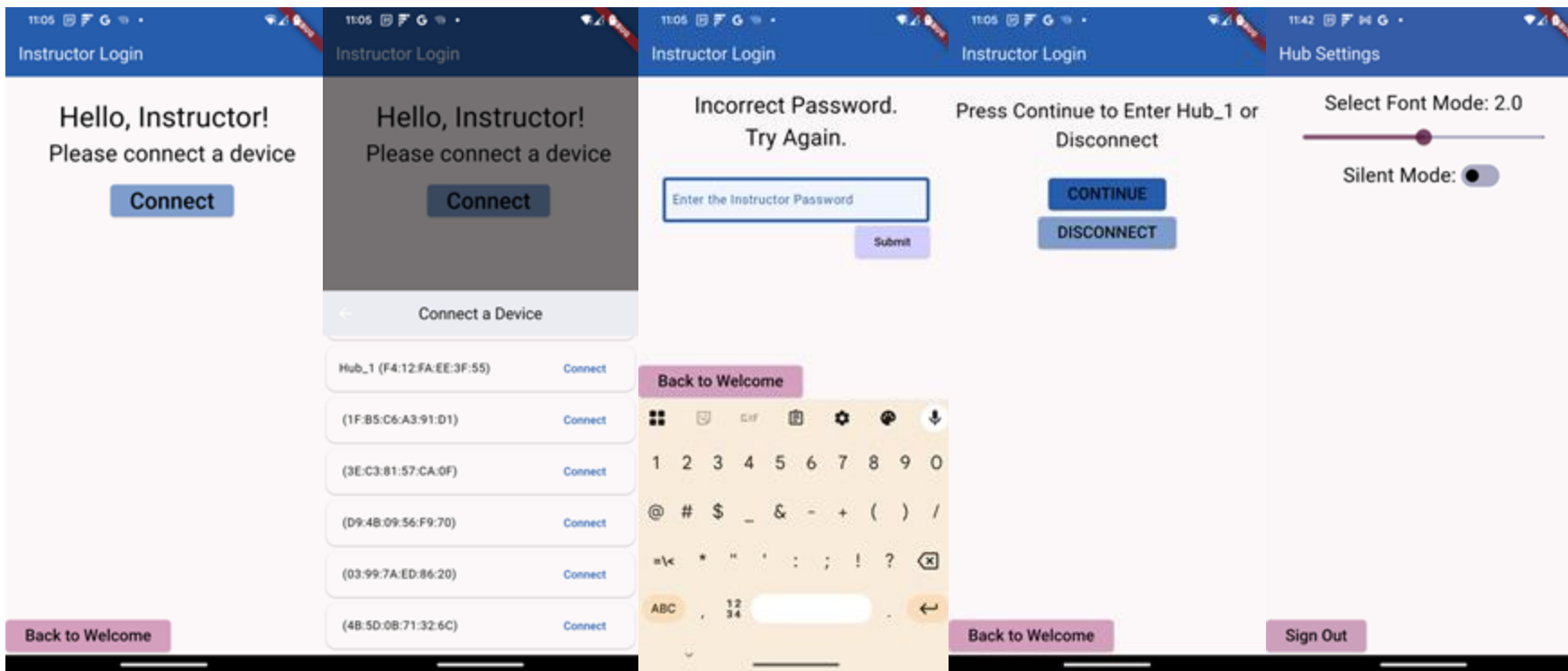


Student Side

The screenshot shows the ClassroomClarity mobile application interface. At the top, there is a status bar with the time 11:00 and various icons. Below the status bar is a blue header with the text "ClassroomClarity". Under the header, there are five colored buttons labeled 1, 2, 3, 4, and 5. Below these buttons is a pink button labeled "Raise Hand". Under the "Raise Hand" button is a light blue text input field with the placeholder text "Type your Question Here". Below the text input field is a character count "0 / 200". At the bottom, there is an "Anonymous:" toggle switch and a purple "Submit" button.



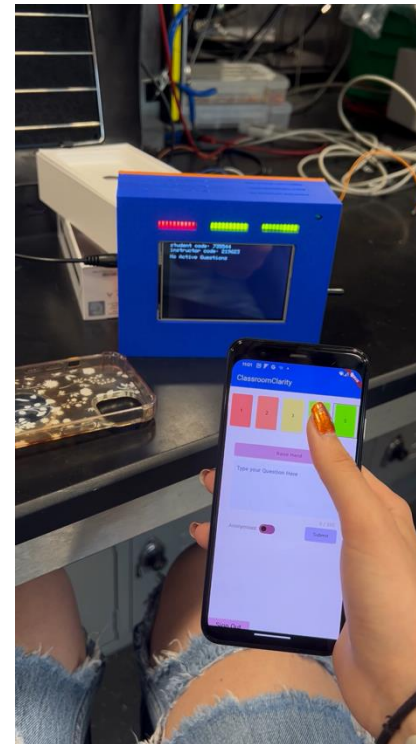
Instructor Side



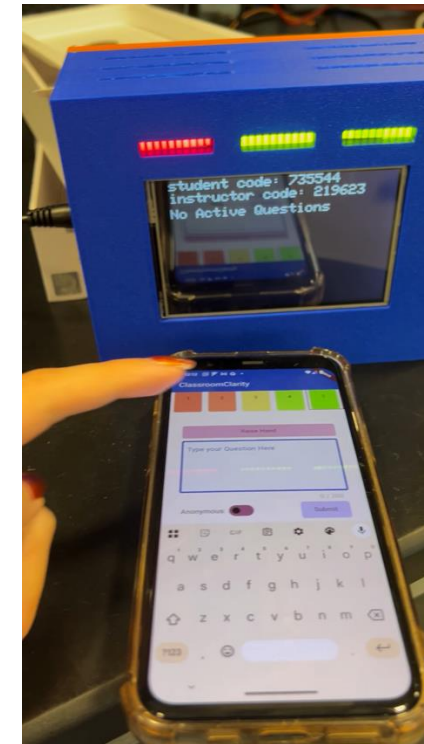
Requirements and Verification

- The **app should respond** to any user interactions **within 1 second**
- The app should be **able to send and receive data** with a less than **1% error** rate per transmission.*

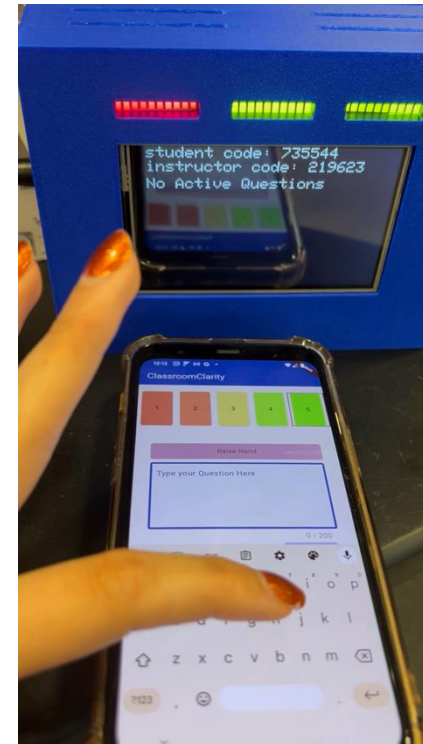
LED



Raise
Hand



Type
Question



*High Level Requirement

Requirements and Verification

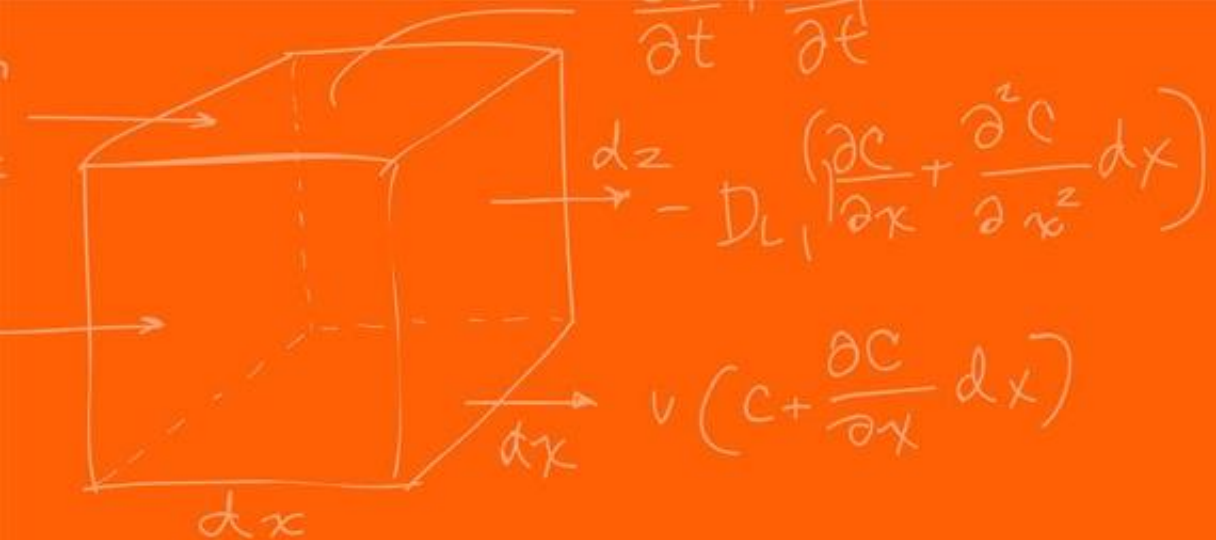
- The app must **bluetooth connect to ESP32 within 10 seconds** of initiating a connection request

Start Time HH:MM:SS	End Time HH:MM:SS	Connect Time
11:27:55.106995	11:27:56.639120	1.532125 s

Conclusion

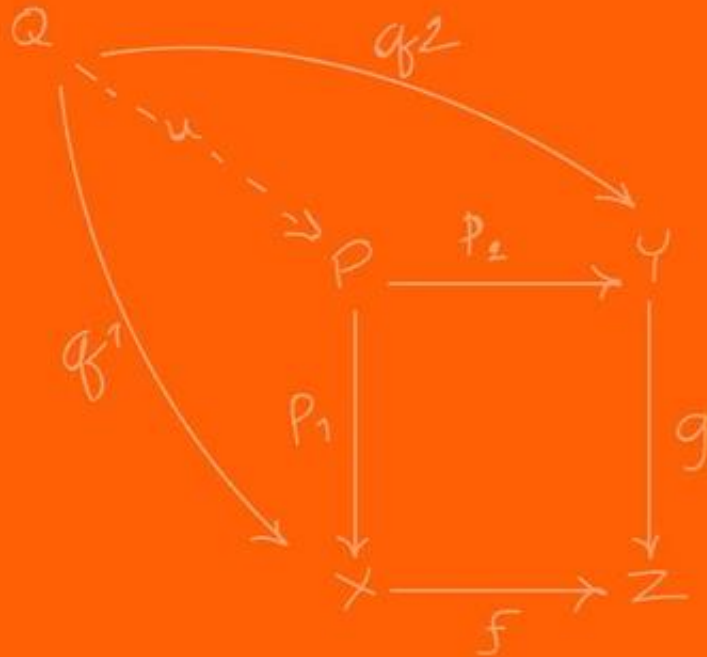
Future Work

- **Testing**
 - Test multiple phones connected at once
- **Additional Software Features**
 - Students can view queued questions in the app
 - A question counter on the Hub
- **Additional Physical Features**
 - Use a rotary encoder with smaller pulse count for easier use
 - Design a stand for the hub



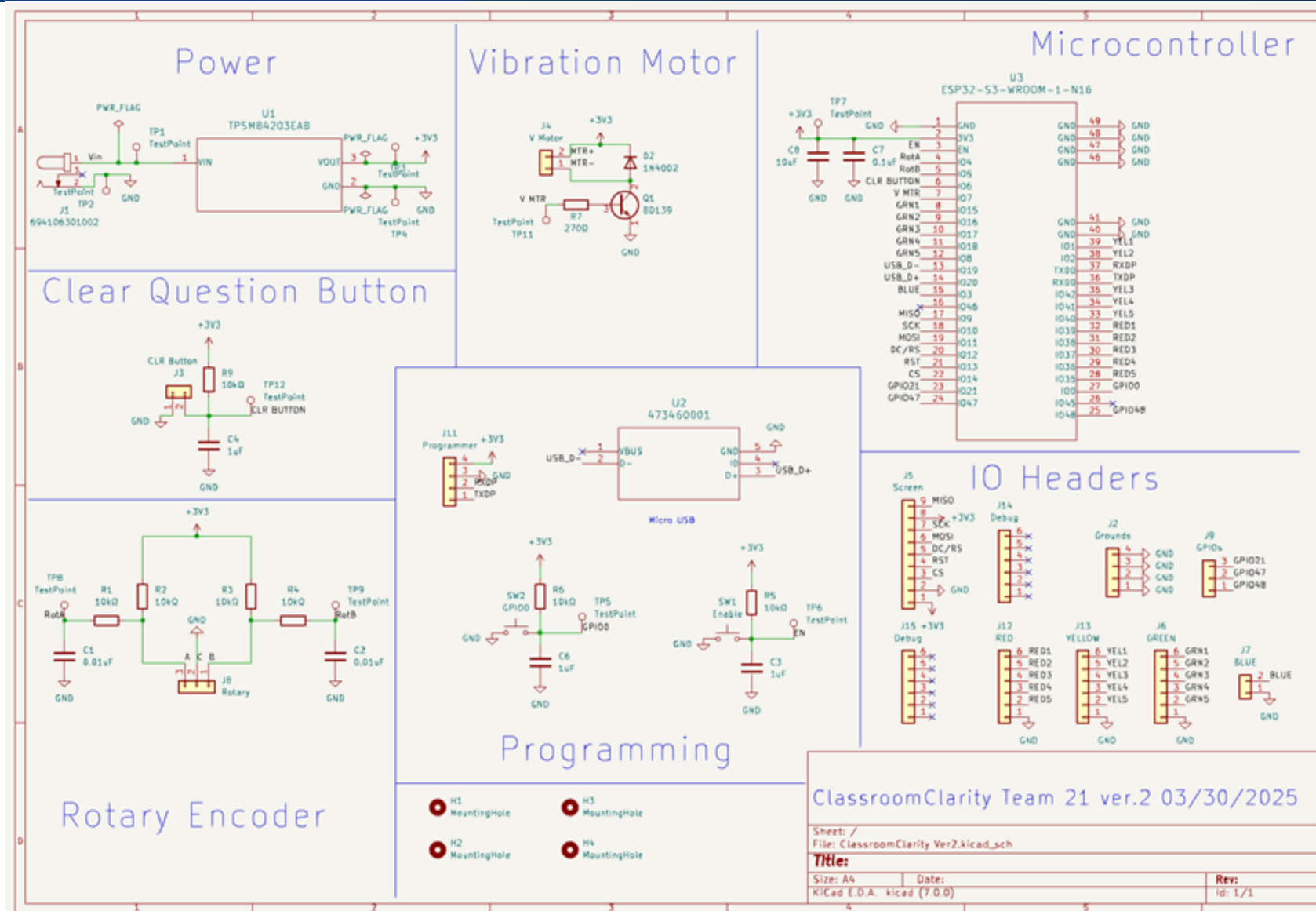
The Grainger College of Engineering

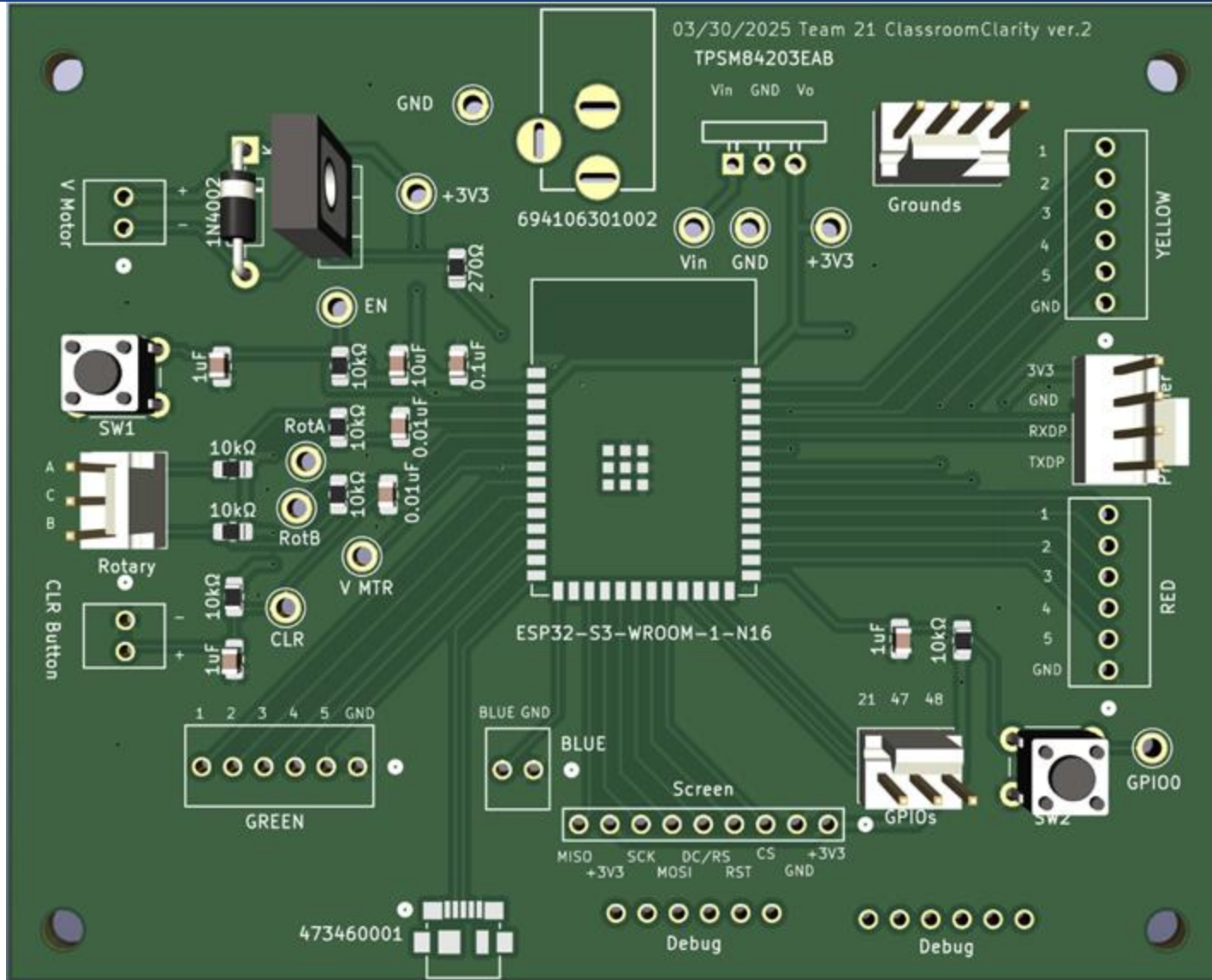
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN



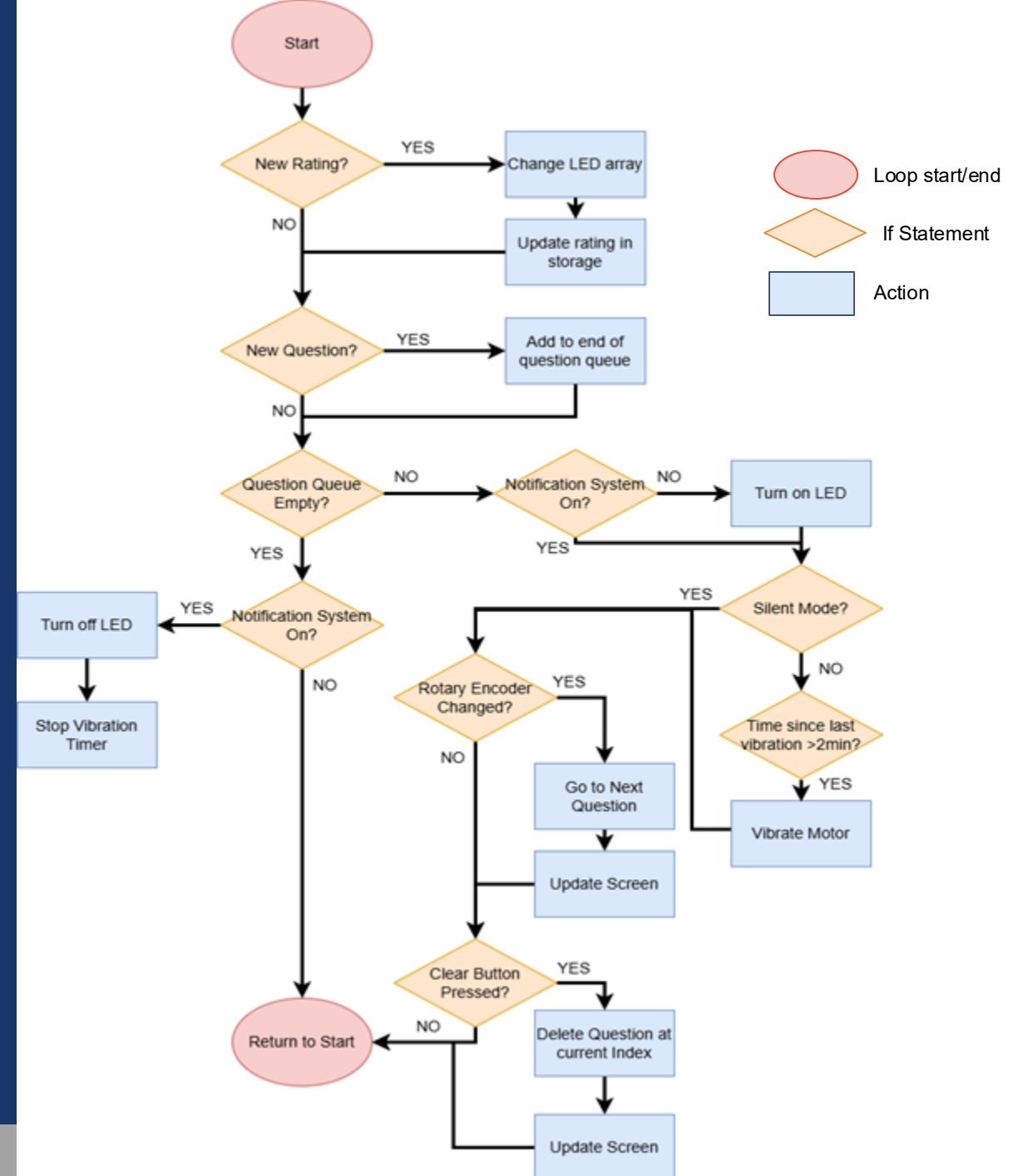
Appendix

Full Circuit Schematic

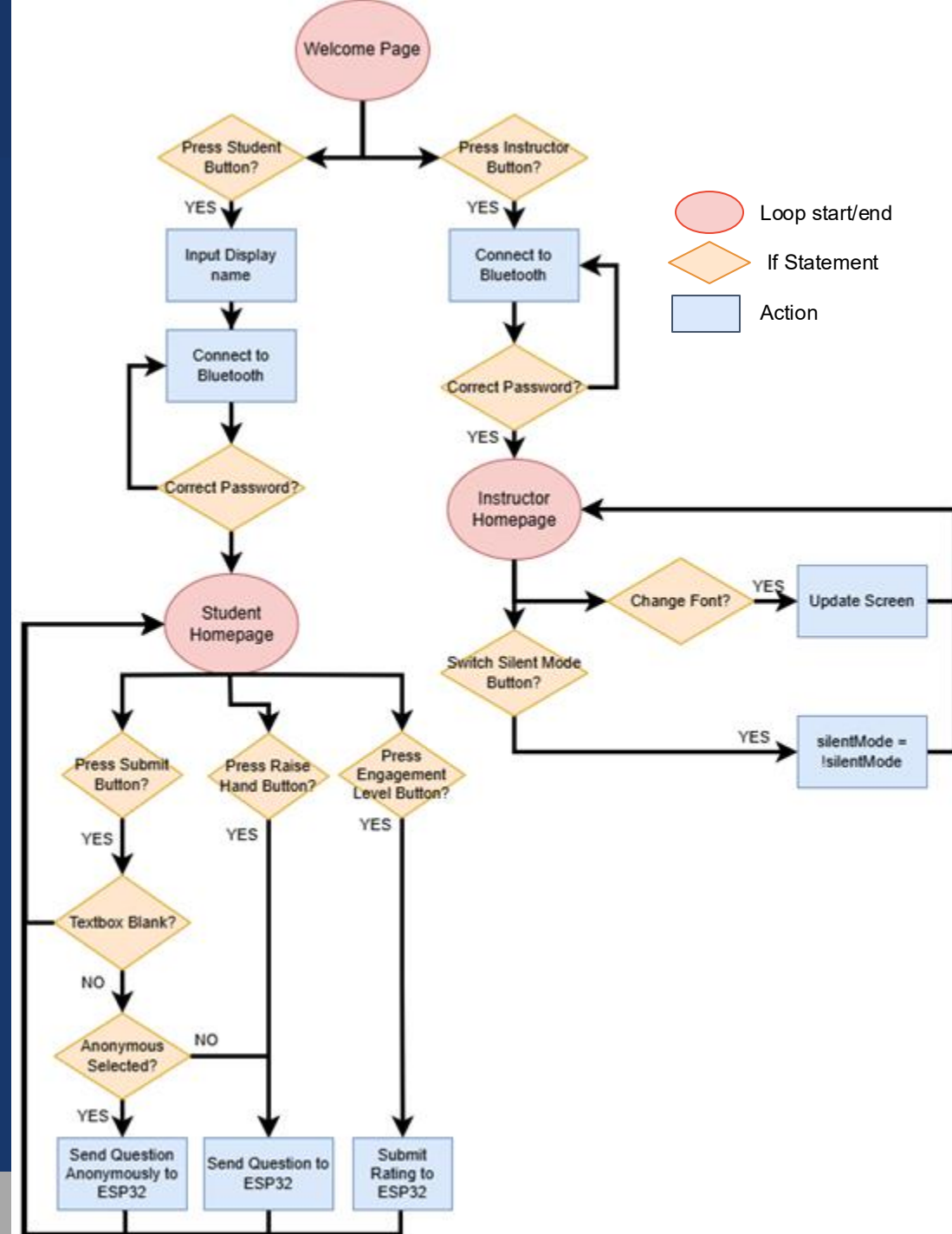




Control Procedure Flowchart



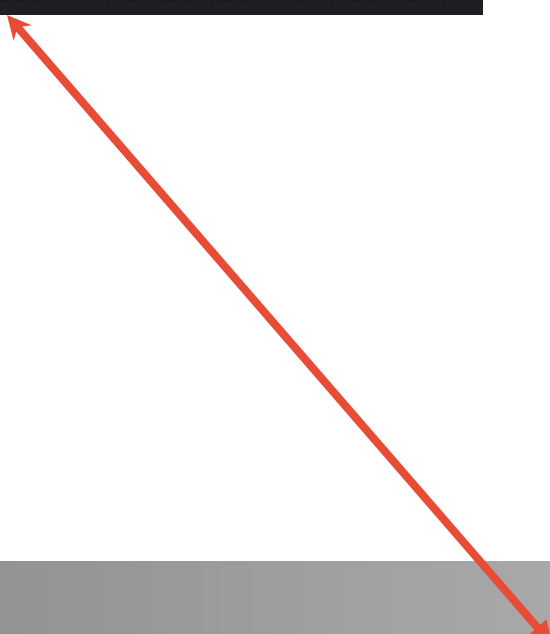
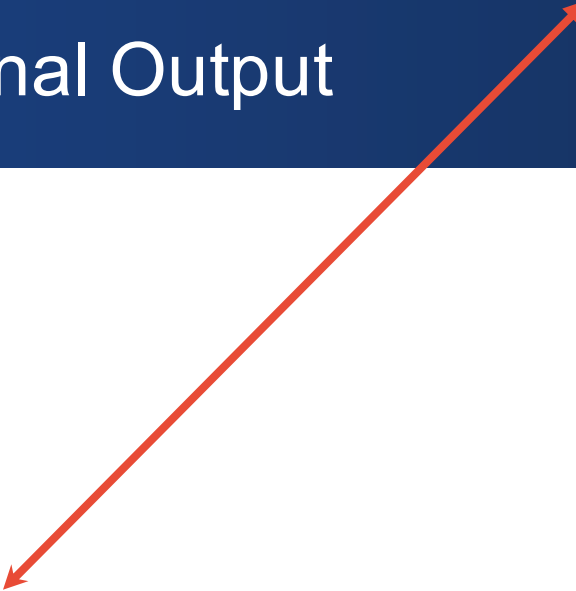
Mobile App Full Flowchart



Connection Time Terminal Output

I/flutter (19673): '2025-05-02T11:27:55.106995 - START CONNECT

I/flutter (19673): '2025-05-02T11:27:56.649120 - FINISHED CONNECTION



```
I/flutter (19673): '2025-05-02T11:27:55.106995 - START CONNECT
D/[FBP-Android](19673): [FBP] onMethodCall: connect
D/BluetoothGatt(19673): connect() - device: F4:12:FA:EE:3F:55, auto: false
D/BluetoothGatt(19673): registerApp()
D/BluetoothGatt(19673): registerApp() - UUID=0dac1b99-d26f-4f27-9f78-c87e5bb0e43d
D/BluetoothGatt(19673): onClientRegistered() - status=0 clientIf=6
D/[FBP-Android](19673): [FBP] onMethodCall: getAdapterState
D/BluetoothGatt(19673): onClientConnectionState() - status=0 clientIf=6 device=F4:12:FA:EE:3F:55
D/[FBP-Android](19673): [FBP] onConnectionStateChange:connected
D/[FBP-Android](19673): [FBP] status: SUCCESS
D/[FBP-Android](19673): [FBP] onMethodCall: requestMtu
D/BluetoothGatt(19673): configureMTU() - device: F4:12:FA:EE:3F:55 mtu: 512
D/BluetoothGatt(19673): onConnectionUpdated() - Device=F4:12:FA:EE:3F:55 interval=6 latency=0 timeout=500 status=0
D/BluetoothGatt(19673): onConfigureMTU() - Device=F4:12:FA:EE:3F:55 mtu=512 status=0
D/[FBP-Android](19673): [FBP] onMtuChanged:
D/[FBP-Android](19673): [FBP] mtu: 512
D/[FBP-Android](19673): [FBP] status: GATT_SUCCESS (0)
I/flutter (19673): Connected Device: BluetoothDevice{remoteId: F4:12:FA:EE:3F:55, platformName: Hub_1, services: null}
D/[FBP-Android](19673): [FBP] onMethodCall: discoverServices
D/BluetoothGatt(19673): discoverServices() - device: F4:12:FA:EE:3F:55
D/BluetoothGatt(19673): onSearchComplete() = Device=F4:12:FA:EE:3F:55 Status=0
D/[FBP-Android](19673): [FBP] onServicesDiscovered:
D/[FBP-Android](19673): [FBP] count: 3
D/[FBP-Android](19673): [FBP] status: 0GATT_SUCCESS
D/[FBP-Android](19673): [FBP] onMethodCall: setNotifyValue
D/BluetoothGatt(19673): setCharacteristicNotification() - uuid: 00002a05-0000-1000-8000-00005f9b34fb enable: true
D/[FBP-Android](19673): [FBP] status: 0GATT_SUCCESS
D/[FBP-Android](19673): [FBP] onMethodCall: setNotifyValue
D/BluetoothGatt(19673): setCharacteristicNotification() - uuid: 00002a05-0000-1000-8000-00005f9b34fb enable: true
D/BluetoothGatt(19673): onConnectionUpdated() - Device=F4:12:FA:EE:3F:55 interval=36 latency=0 timeout=500 status=0
D/[FBP-Android](19673): [FBP] onDescriptorWrite:
D/[FBP-Android](19673): [FBP] chr: 2a05
D/[FBP-Android](19673): [FBP] desc: 2902
D/[FBP-Android](19673): [FBP] status: GATT_SUCCESS (0)
I/flutter (19673): Discovered Services: [BluetoothService{remoteId: F4:12:FA:EE:3F:55, serviceUuid: 1801, primaryServiceUuid: null, characteristics:
[BluetoothCharacteristic{remoteId: F4:12:FA:EE:3F:55, serviceUuid: 1801, characteristicUuid: 2a05, primaryServiceUuid: null, descriptors:
[BluetoothDescriptor{remoteId: F4:12:FA:EE:3F:55, serviceUuid: 1801, characteristicUuid: 2a05, descriptorUuid: 2902, primaryServiceUuid: null, lastValue: [2,
0]}], properties: CharacteristicProperties{broadcast: false, read: false, writeWithoutResponse: false, write: false, notify: false, indicate: true,
authenticatedSignedWrites: false, extendedProperties: false, notifyEncryptionRequired: false, indicateEncryptionRequired: false, value: []}], },
BluetoothService{remoteId: F4:12:FA:EE:3F:55, serviceUuid: 1800, primaryServiceUuid: null, characteristics: [BluetoothCharacteristic{remoteId:
F4:12:FA:EE:3F:55, serviceUuid: 1800, characteristicUuid: 2a06, primaryServiceUuid: null, descriptors: [], properties: CharacteristicProperties{broadcast:
false,
I/flutter (19673): Checking characteristic: 2a05
I/flutter (19673): Checking characteristic: 2a00
I/flutter (19673): Checking characteristic: 2a01
I/flutter (19673): Checking characteristic: 2a0d
I/flutter (19673): Checking characteristic: 1afd884a-d939-4805-9875-4cde8b10d07a
I/flutter (19673): Checking characteristic: 0b9e8c81-39d7-4b86-8d34-4c192b6e3926
I/flutter (19673): Checking characteristic: d909a0e1-c07e-44bf-9e71-53826482688d
I/flutter (19673): Checking characteristic: c583e5c9-2d3c-4beb-97ea-8996d7a97493
I/flutter (19673): Checking characteristic: 979d3386-99d7-4684-9e50-79e0ee765f7
I/flutter (19673): '2025-05-02T11:27:56.649120 - FINISHED CONNECTION
```