Project 8:
Heart Rate Monitor

Nick Liu & Zach Trainor

Team Members

Zach T.

- Frontend and Backend
- Setting up sensor, sending data
- UI Design

Nick L.

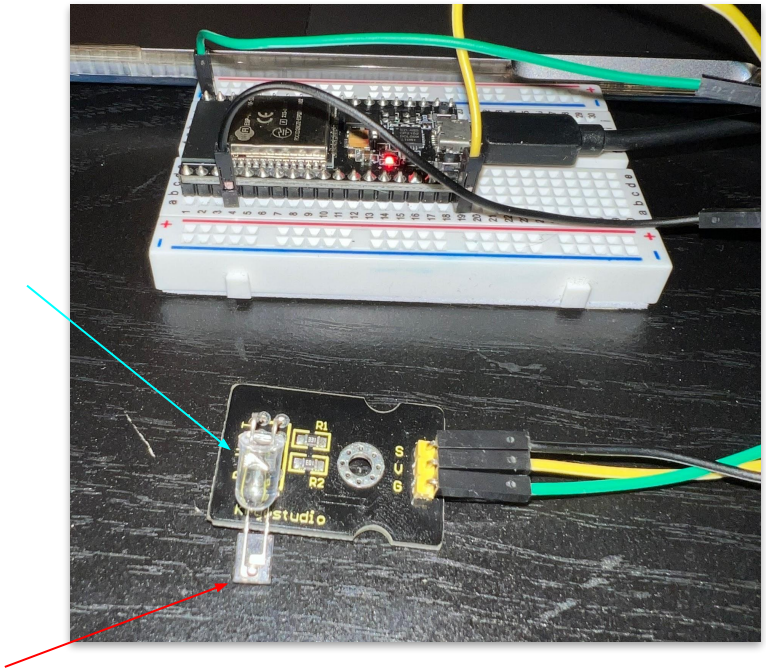
- Frontend
- Implementation of Figma UI into Flutter
- UI design
- Frontend & Backend integration

The Problem

- Quiet but serious issue
- More than half of all Americans age 20 and up suffer from hypertension (high blood pressure) (USC)
- Issues lays in unawareness

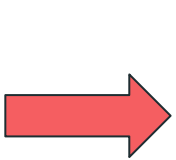
Our App's Solution

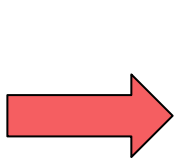
- Monitor user's heart rate on demand via AtSign
- Input age, get reading on current heart rate
- Heart Rate too low/high -> potential remedies



Use Case

- Joe is 22 years old and has been under tremendous stress lately.
- He is worried about his heart health and decides to use our app.
- (demo)





```
42
43      while (true) {
44        int newReading = analogRead(sensorPin);
45          // if the ADC value is less than ~100,
46          // the user does not have their finger on the sensor
47        if (newReading < 100) {
48          std::cout << "ERROR: finger not on monitor" << std::endl;
49          break;
50        }
51
52        if (oldReading > newReading) {
53          pinMode(LED, OUTPUT);
54          digitalWrite(LED, HIGH); // flashes ESP32 LED per beat
55          std::cout << "beat" << std::endl;
56          count += 1;
57          // usleep(100000);
58          delay(100);
59          digitalWrite(LED, LOW);
60        }
61
62        std::cout << " "; // print empty line if no heart beat is read
63
64        // check for heart beat every 0.25 seconds
65        //usleep(250000);
66        delay(250);
67        timeCount += 0.25;
68        oldReading = newReading;
69
70        // measure heart rate for 15 seconds
71        if (timeCount >= 15) {
72          value = count * 4;
73          // calculate heart rate using beat count from 15 seconds (15 * 4 = 60 seconds = BPM)
74          break;
75        }
76      }
```

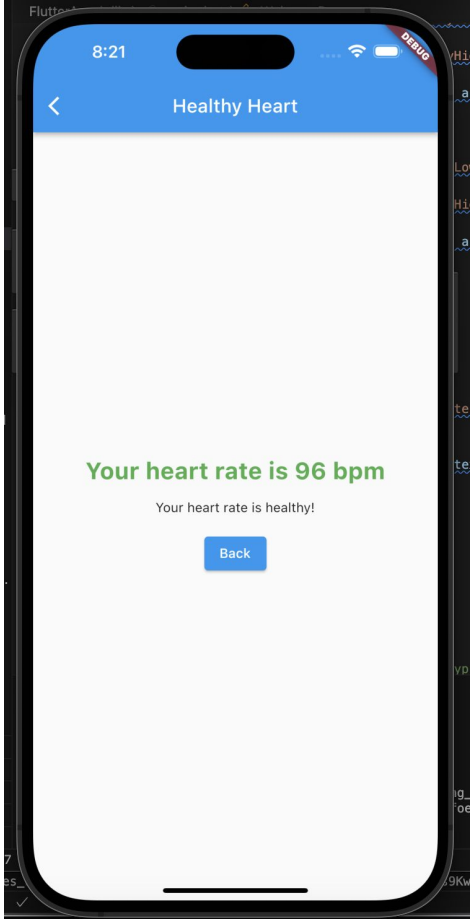
```
aut      beat
    beat
   beat
  beat
   beat
   beat
    beat
 beat
   beat
    beat
 beat
 beat
    beat
 beat
   beat
 beat
 beat
   beat
  beat
 beat
 beat
   beat
 beat
  beat
  COUNT: 24 beats
Heart rate: 96bpm
```

ESP32 Reading Heartbeat

```java

public class App
{
    Run | Debug
    public static void main( String[] args ) throws Exception
    {
        AtSign java = new AtSign(atSign:"@bittersweet8");
        AtSign esp32 = new AtSign(atSign:"@the60melted");

        AtClient atClient = AtClient.withRemoteSecondary(rootUrl:"root.atsign.org:64", java);

        SharedKey sharedKey = new KeyBuilders.SharedKeyBuilder(esp32, java).key(key:"heartrate").build();

        String data = atClient.get(sharedKey).get();
        PrintWriter writer = new PrintWriter("heartrate.txt", "UTF-8");
        writer.println(data);
        writer.close();
        System.out.println("Heart rate: " + data + "bpm");
    }
}
```

```
Zachs-MacBook-Pro:receiving_demo zach$ cd javadata
Zachs-MacBook-Pro:javadata zach$  cd /Users/zach/receiving_demo/javadata ; /usr/bin/env /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -cp /var/folders/k
_/1pw4zj6d0nz67dtk7kmjb84w0000gn/T/cp_ekb13oawsbivae9j36foe8oh5.jar com.example.App
Heart rate: 96bpm
```

App Receives Heart Rate via AtSign



Data Displayed on App

Problems Encountered & Lessons Learned

Sensor Issues:

- Pulse rate sensor was tricky:
  - Wiring
  - Calculating BPM
  - Transferring data
- Sensor data: light readings -> BPM

UI Issues:

- Putting multiple elements in one page was challenging, had to aim for simpler design.
- Getting the data from backend to frontend.

Working Together:

- Finding time
- Single, unified project vs. two separate parts
- Time Management and Communication