Find Your Cafe

Finding Study Space Made Easier Throughout the City

In this exercise, I'm proposing a cloud connected interactive computing system that predicts the density of people in a cafe and sends a message to the user's phone. My target audience are students who enjoy studying in public cafes in cental London, but struggle to find available seatings without having to be physically pre

I use the level of CO2 in a given indoor space to predict the density of people in the cafe, which informs the user whether there may be available spaces to sit at the cafe without having to physically go to the place, therefore helping them save time.

I use Arduino ESP32 and CCS811 sensors to capture the CO2 level in the room, and connect the data to the cloud. When the user inputs a command to check the availability of a given cafe, the system pulls the backend data, analyzes the density through an embdedded AI program, and provides a feedback to the user.

1/ Persona

Carrie

Age	24
Location	Central London
Occupation	Student

Motivations

Interactive Design

Digital Ecology

Architecture and Research

Frustrations

Less Exposure to New Tech

Little Non-academic Time

No Traveling Opportunities

Goals

Find free places to study in public cafes without spending time on finding seats

Save time by using a tracking App that notifies cafe availabilities

Personality

Curious

Creative

Social

Bio

Carrie is an architecture student at the AA and she is highly interested in learning digital technology and interactive design.

Carrie enjoys studying in public spaces and especially in cafes, but she finds it difficult to easily find places in central London that provide power sockets, is not too loud, and not too crowded. She would like to create a system that will track in real time where seats have become available at such cafes in central London and what set-ups they come with, so she can decide where to go based on instant availability.

Preferred Channels

Discord

Instagram

Whatsapp

Logos

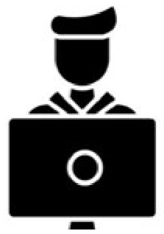
1/ User Stories

As a student who enjoys studying in public spaces in central London

I would like to design an App that tracks the availabilities of cafes suitable for studying

So that I can save time without arriving somewhere without seats

1/ Storyboard



Step1

Wants to go to a public café to study

Step2

Walks to the café

Step3

Discovers that there are no seats available

Step4

Must leave and find new place

Step5

Uses App to find a place with availability in advance

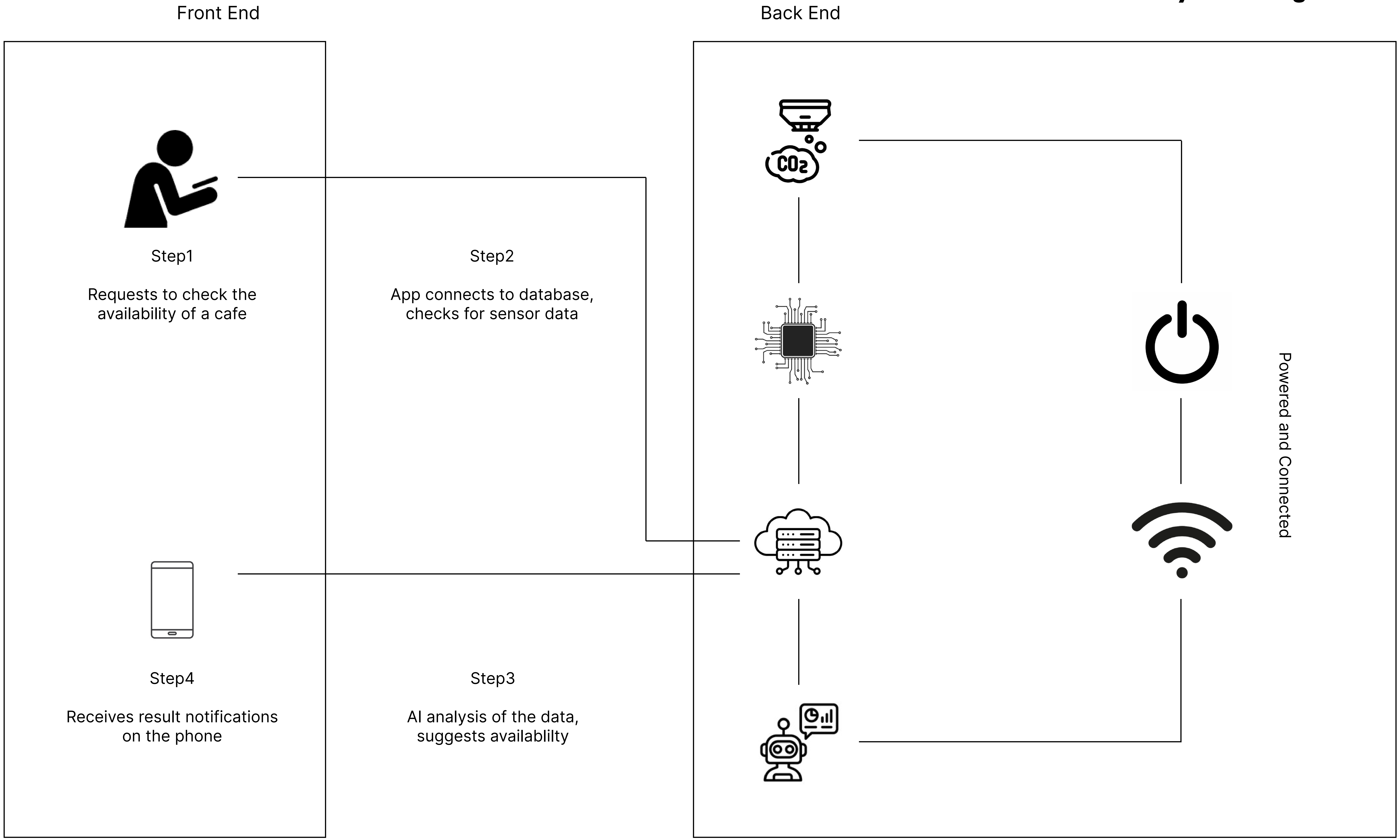
Step6

Walks there to study without wasting any time

2/ Components

Power	Powering the Sensors
WiFi	Connecting Components
ESP 8266	Microcontroller
CCS811 Air Quality Sensor	Detects CO2 Level in Room
Cloud Server	Data Transit and Storage
AI Analysis	AI Computation of Sensor Data
Mobile Phone	User Interface and Notifications

2/ System Diagram and Protocols



Front End

Back End

Step1

Requests to check the availability of a cafe

Step2

App connects to database, checks for sensor data

Step3

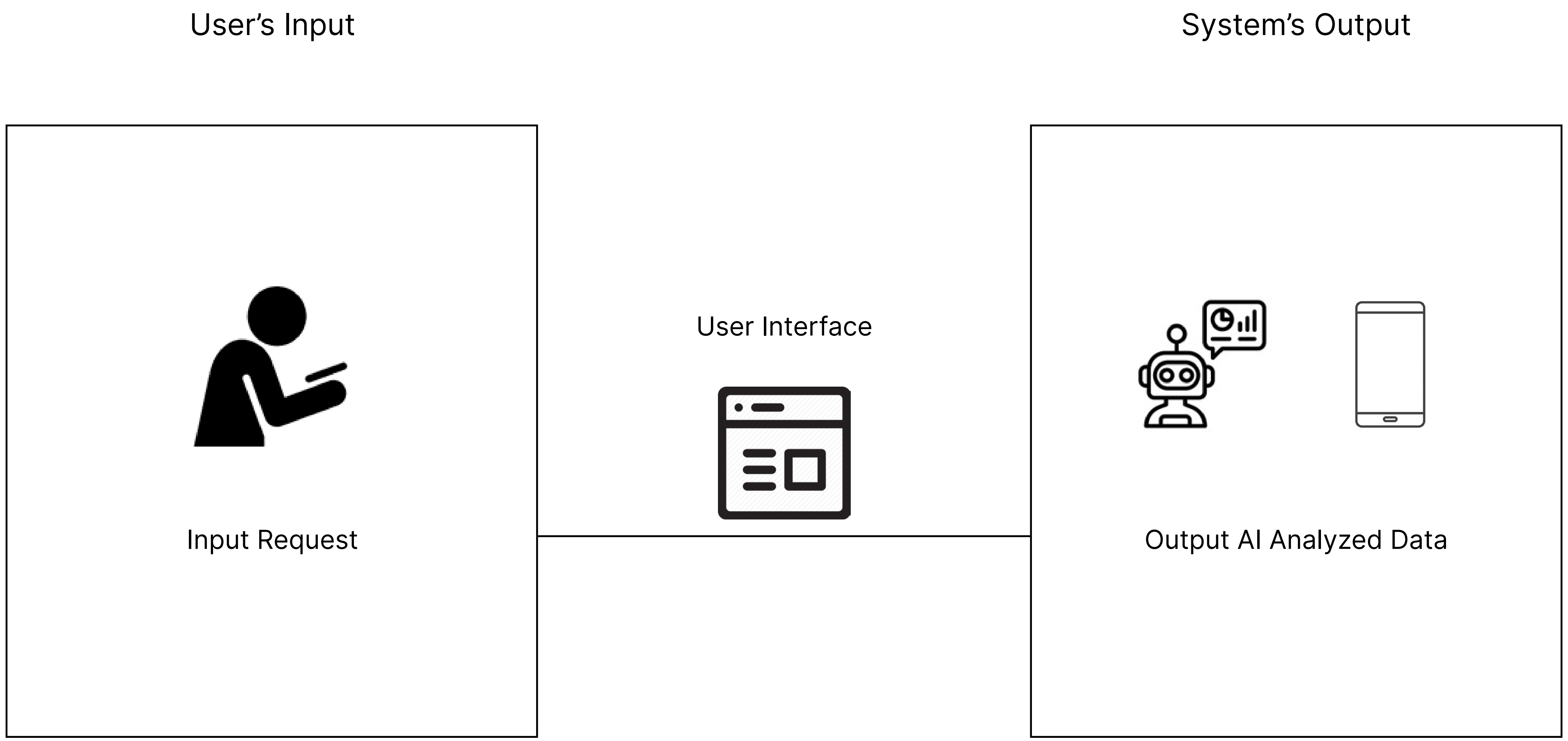
AI analysis of the data, suggests availablilty

Step4

Receives result notifications on the phone

Powered and Connected

2/ Touch Points and Modes of Interaction



User’s Input

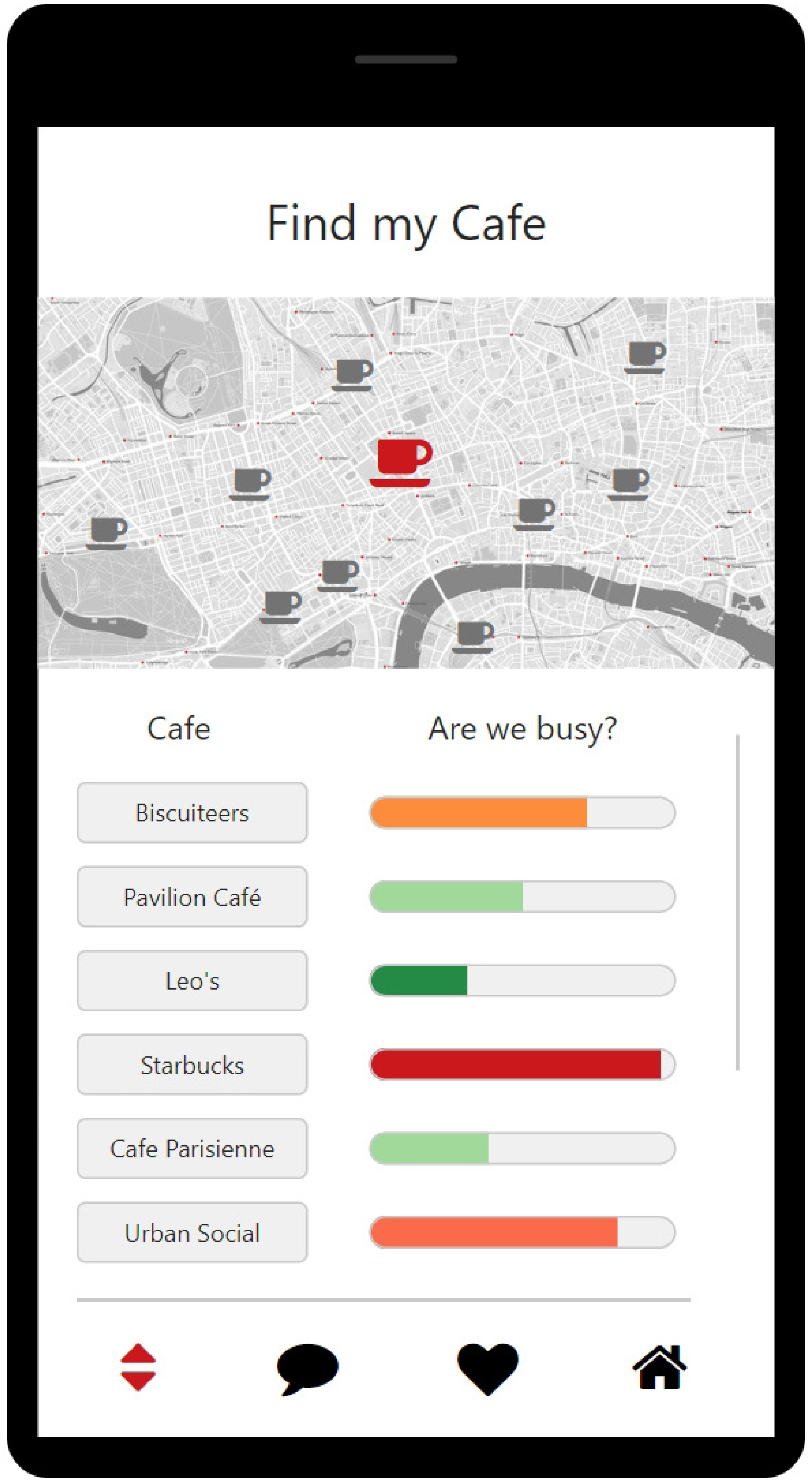
System’s Output

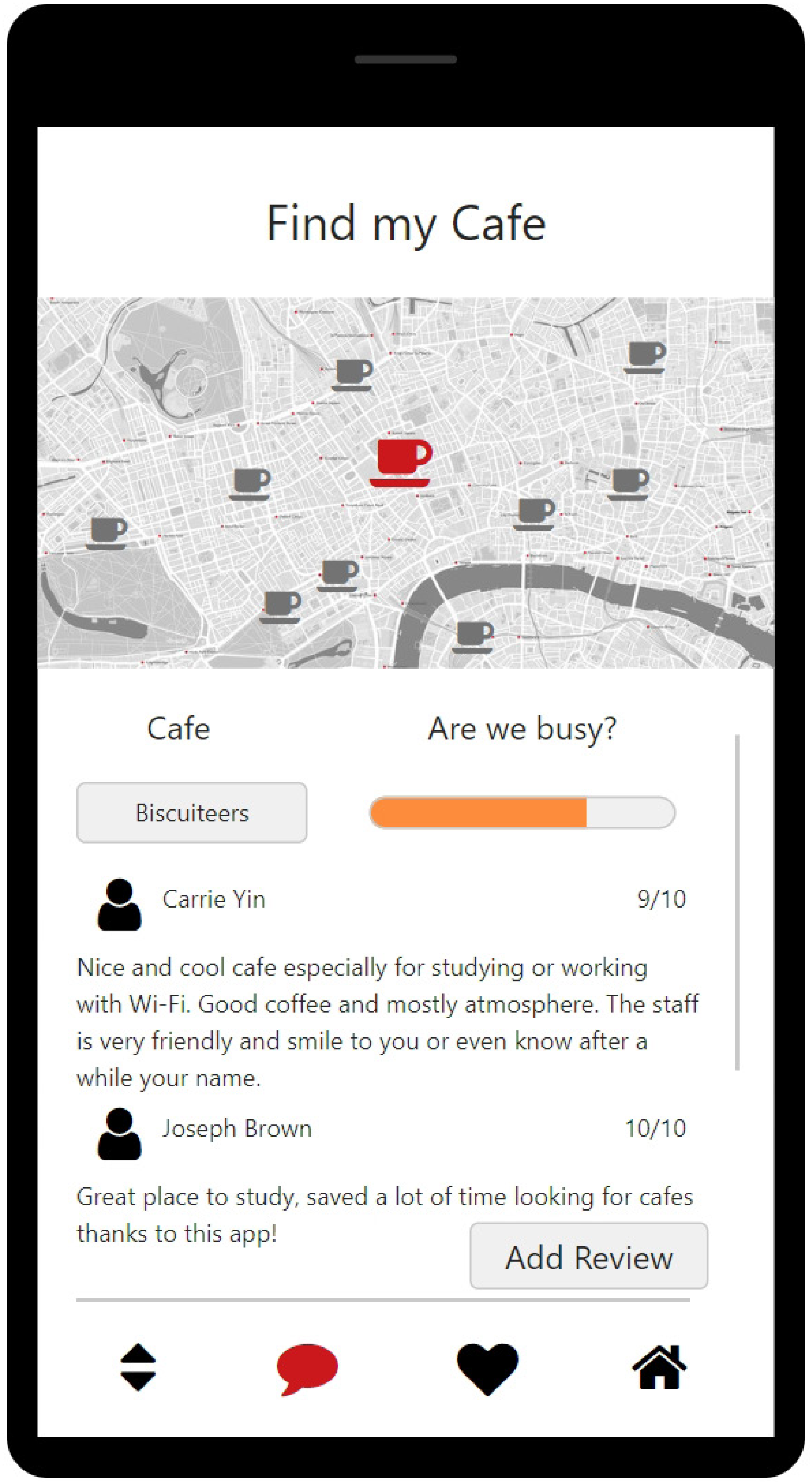
User Interface

Input Request

Output AI Analyzed Data

3/ UI Wireframe





Carrie Yin 9/10

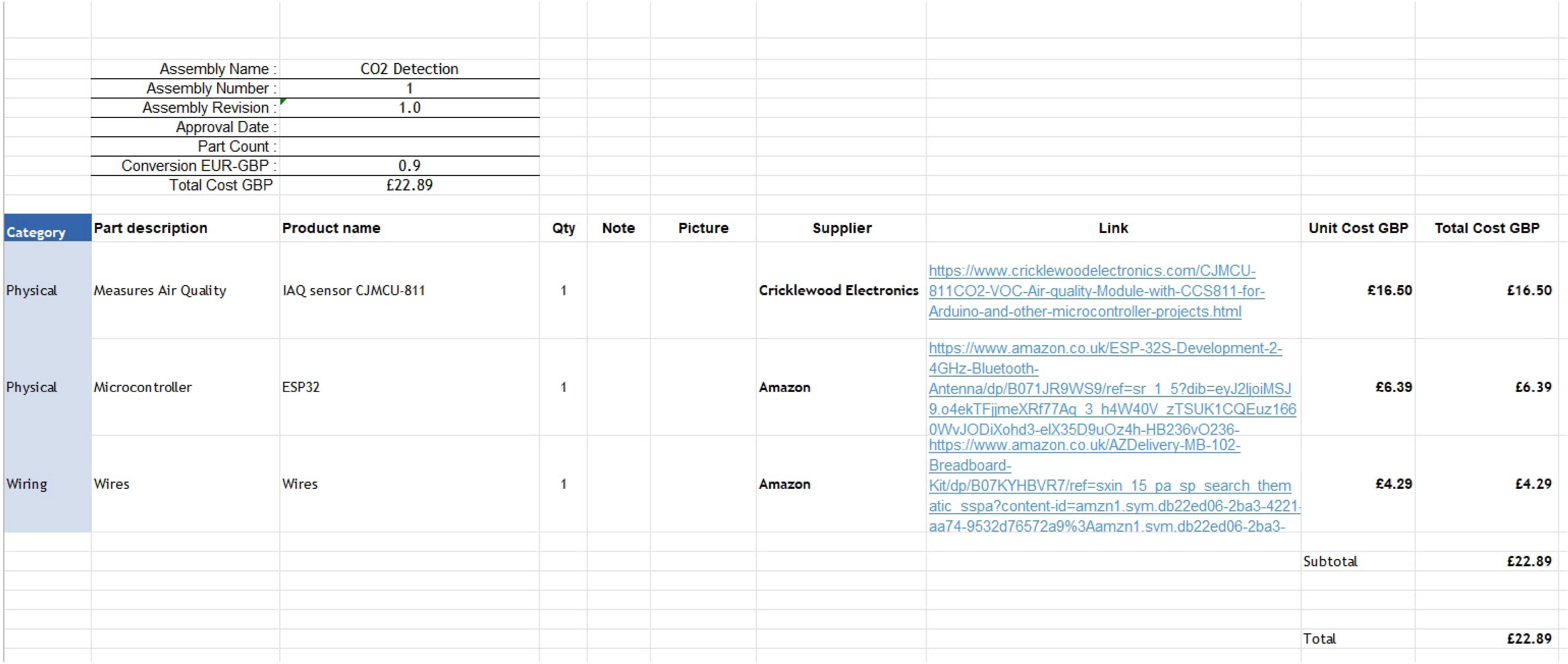
Nice and cool cafe especially for studying or working with Wi-Fi. Good coffee and mostly atmosphere. The staff is very friendly and smile to you or even know after a while your name.

Joseph Brown 10/10

Great place to study, saved a lot of time looking for cafes thanks to this app!

User Name: carrieyin1

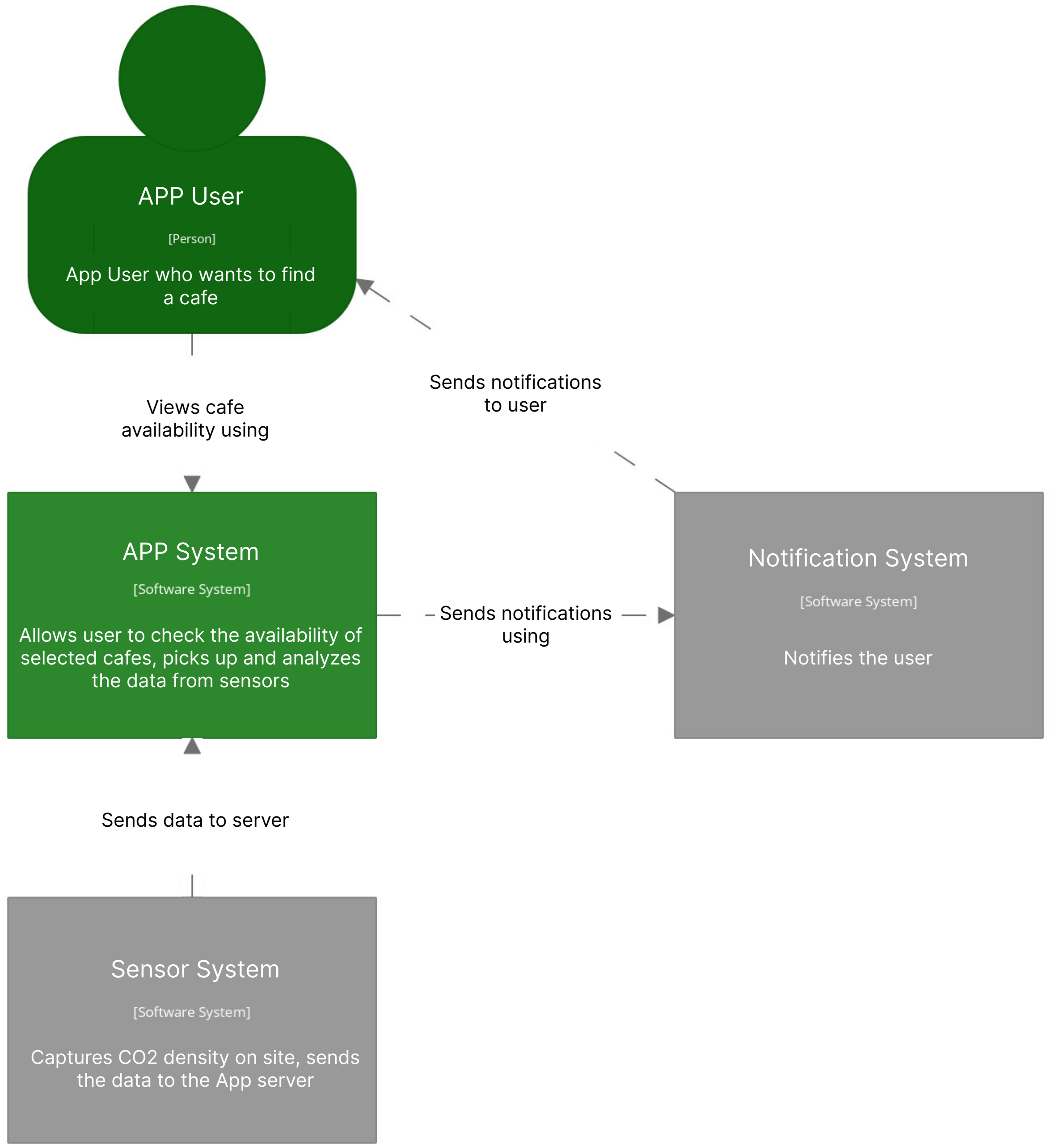
4/ Bill of Materials



Assembly Name :	CO2 Detection
Assembly Number :	1
Assembly Revision :	1.0
Approval Date :	
Part Count :	
Conversion EUR-GBP :	0.9
Total Cost GBP	£22.89

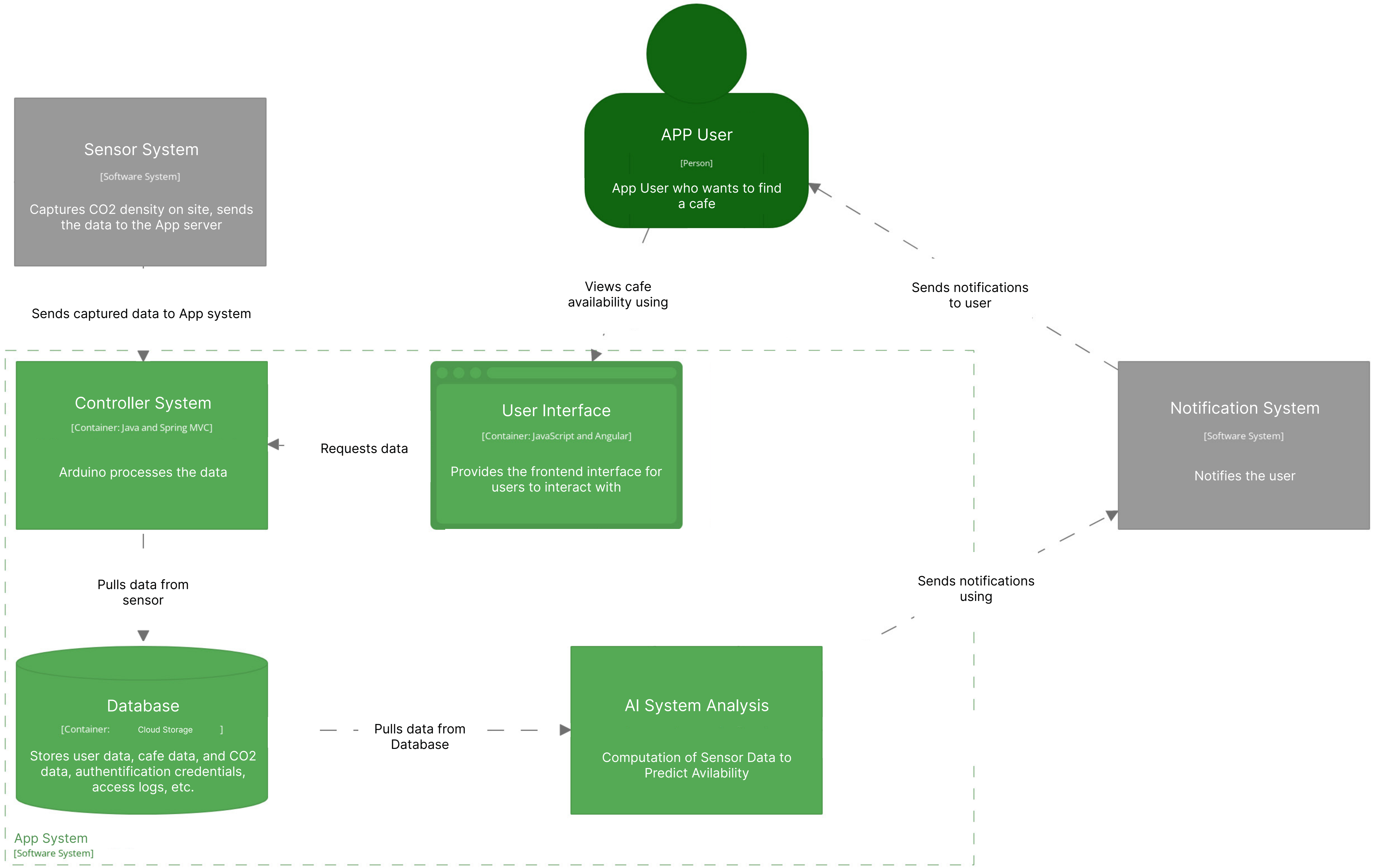
Category	Part description	Product name	Qty	Note	Picture	Supplier	Link	Unit Cost GBP	Total Cost GBP
Physical	Measures Air Quality	IAQ sensor CJMCU-811	1			**Cricklewood Electronics**	https://www.cricklewoodelectronics.com/CJMCU-811CO2-VOC-Air-quality-Module-with-CCS811-for-Arduino-and-other-microcontroller-projects.html	**£16.50**	**£16.50**
Physical	Microcontroller	ESP32	1			**Amazon**	https://www.amazon.co.uk/ESP-32S-Development-2-4GHz-Bluetooth-Antenna/dp/B071JR9WS9/ref=sr_1_5?dib=eyJ2IjoiMSJ9.o4ekTFjjmeXRf77Aq_3_h4W40V_zTSUK1CQEuz1660WvJODiXohd3-elX35D9uOz4h-HB236vO236-	**£6.39**	**£6.39**
Wiring	Wires	Wires	1			**Amazon**	https://www.amazon.co.uk/AZDelivery-MB-102-Breadboard-Kit/dp/B07KYHBVR7/ref=sxin_15_pa_sp_search_thematic_sspa?content-id=amzn1.sym.db22ed06-2ba3-4221-aa74-9532d76572a9%3Aamzn1.sym.db22ed06-2ba3-	**£4.29**	**£4.29**
								Subtotal	**£22.89**
								Total	**£22.89**

5/ C4 Model


APP User
[Person]
App User who wants to find a cafe
Views cafe availability using
Sends notifications to user
APP System
[Software System]
Allows user to check the availability of selected cafes, picks up and analyzes the data from sensors
Sends notifications using
Notification System
[Software System]
Notifies the user
Sends data to server
Sensor System
[Software System]
Captures CO2 density on site, sends the data to the App server

C4 Level 1

5/ C4 Model



APP User

[Person]

App User who wants to find a cafe

Sensor System

[Software System]

Captures CO2 density on site, sends the data to the App server

Sends captured data to App system

Views cafe availability using

Sends notifications to user

Controller System

[Container: Java and Spring MVC]

Arduino processes the data

Requests data

User Interface

[Container: JavaScript and Angular]

Provides the frontend interface for users to interact with

Notification System

[Software System]

Notifies the user

Pulls data from sensor

Sends notifications using

Database

[Container: Cloud Storage]

Stores user data, cafe data, and CO2 data, authentification credentials, access logs, etc.

Pulls data from Database

AI System Analysis

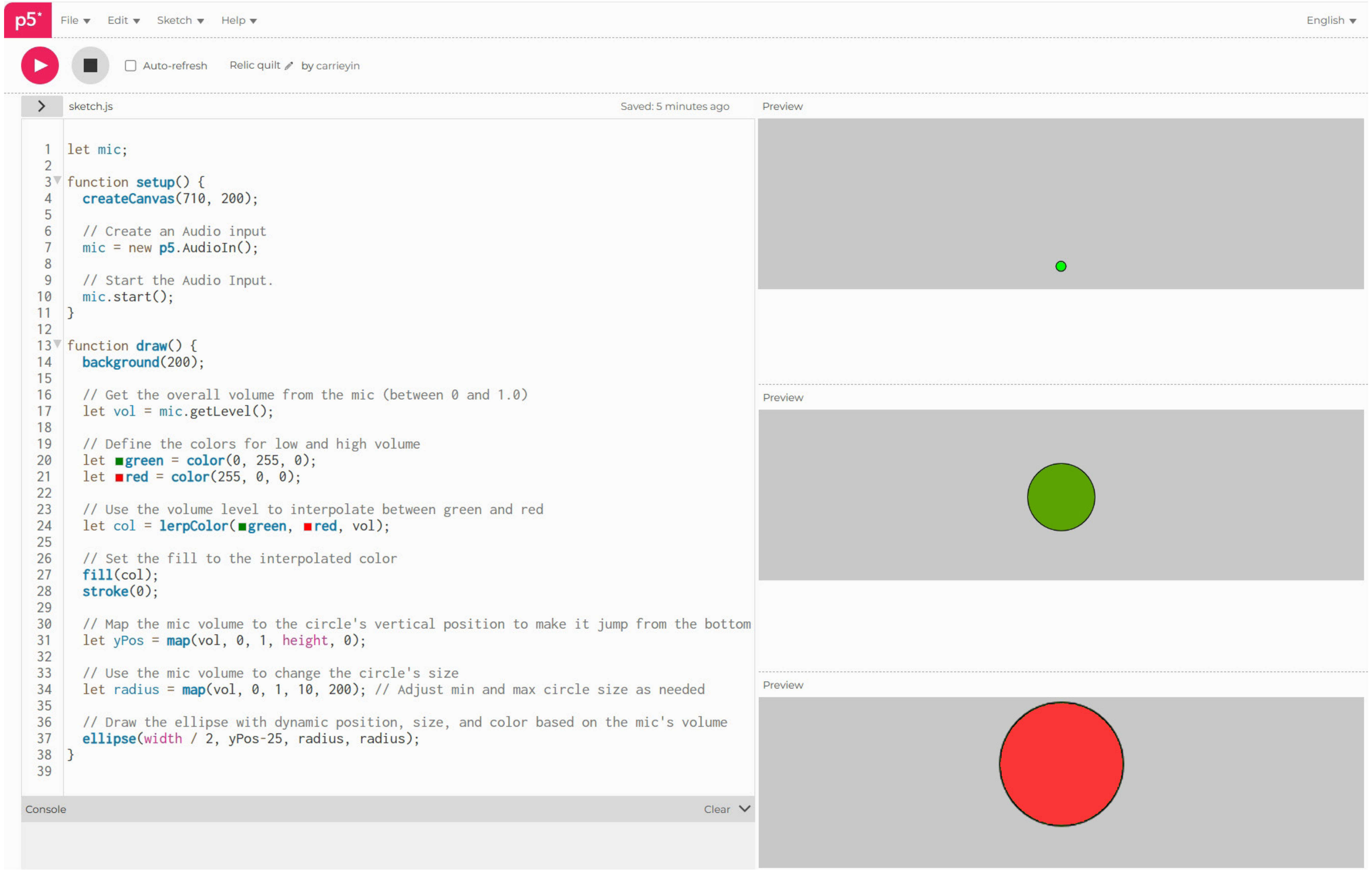
Computation of Sensor Data to Predict Avilability

App System

[Software System]

C4 Level 2

5/ P5 Modification



```
let mic;

function setup() {
  createCanvas(710, 200);

  // Create an Audio input
  mic = new p5.AudioIn();

  // Start the Audio Input.
  mic.start();
}

function draw() {
  background(200);

  // Get the overall volume from the mic (between 0 and 1.0)
  let vol = mic.getLevel();

  // Define the colors for low and high volume
  let 🟩green = color(0, 255, 0);
  let 🟥red = color(255, 0, 0);

  // Use the volume level to interpolate between green and red
  let col = lerpColor(🟩green, 🟥red, vol);

  // Set the fill to the interpolated color
  fill(col);
  stroke(0);

  // Map the mic volume to the circle's vertical position to make it jump from the bottom
  let yPos = map(vol, 0, 1, height, 0);

  // Use the mic volume to change the circle's size
  let radius = map(vol, 0, 1, 10, 200); // Adjust min and max circle size as needed

  // Draw the ellipse with dynamic position, size, and color based on the mic's volume
  ellipse(width / 2, yPos-25, radius, radius);
}
```

Library:

p5.sound library

Modified based on:

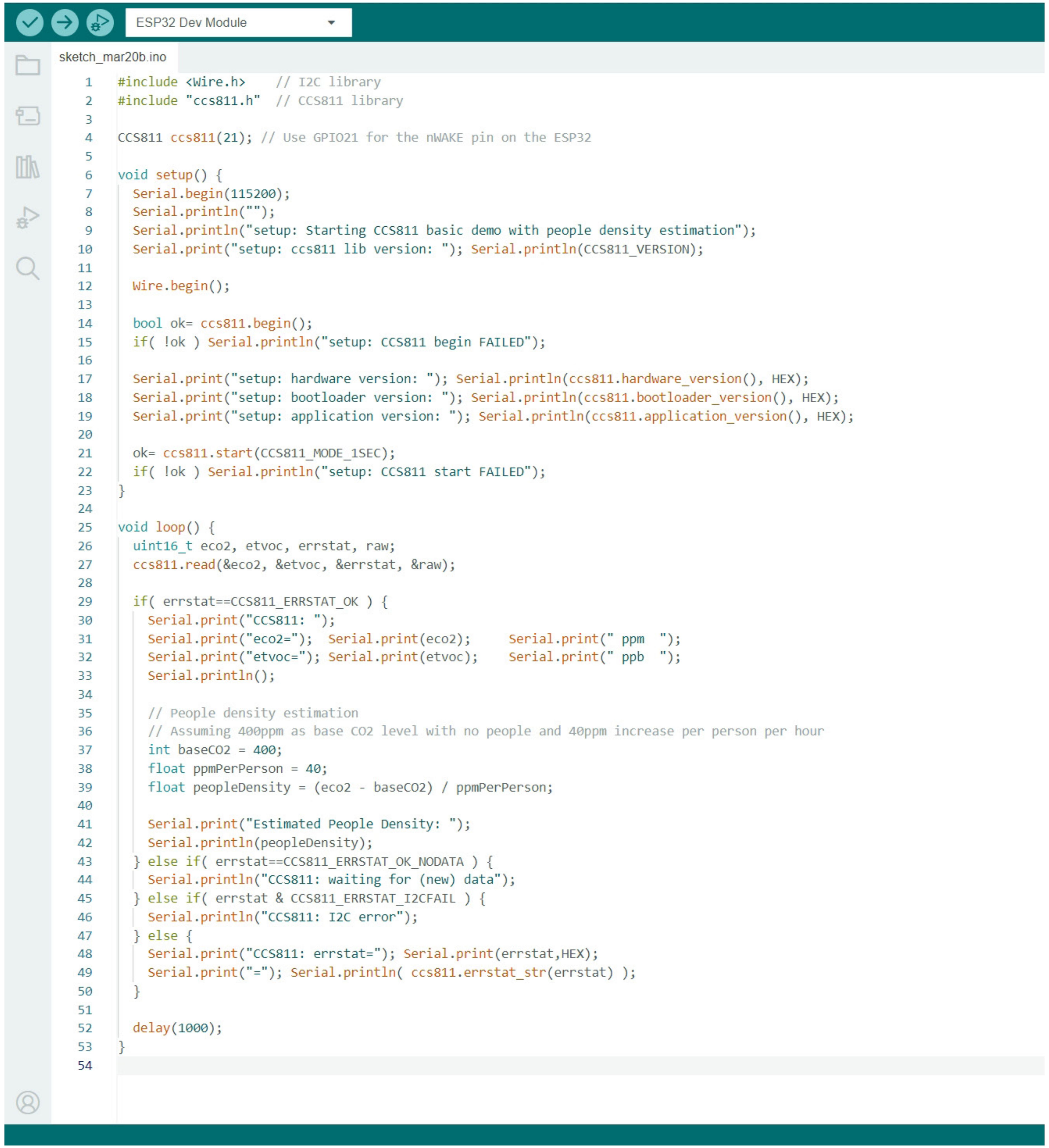
Mic Input

Measure Amplitude

I started my exercise by using sound to measure the desnity of people, and using the location/size/color of this circle to indicate the loudness picked up from the mic. The louder it is, the higher/larger/redder.

However, I later decided to switch to measuring the CO2 level in a room, because I believed that sound sensors can be influenced by more external factors, and therefore result in a less accurate result.

6/ Code



```cpp
#include <Wire.h>    // I2C library
#include "ccs811.h"  // CCS811 library

CCS811 ccs811(21); // Use GPIO21 for the nWAKE pin on the ESP32

void setup() {
  Serial.begin(115200);
  Serial.println("");
  Serial.println("setup: Starting CCS811 basic demo with people density estimation");
  Serial.print("setup: ccs811 lib version: "); Serial.println(CCS811_VERSION);

  Wire.begin();

  bool ok= ccs811.begin();
  if( !ok ) Serial.println("setup: CCS811 begin FAILED");

  Serial.print("setup: hardware version: "); Serial.println(ccs811.hardware_version(), HEX);
  Serial.print("setup: bootloader version: "); Serial.println(ccs811.bootloader_version(), HEX);
  Serial.print("setup: application version: "); Serial.println(ccs811.application_version(), HEX);

  ok= ccs811.start(CCS811_MODE_1SEC);
  if( !ok ) Serial.println("setup: CCS811 start FAILED");
}

void loop() {
  uint16_t eco2, etvoc, errstat, raw;
  ccs811.read(&eco2, &etvoc, &errstat, &raw);

  if( errstat==CCS811_ERRSTAT_OK ) {
    Serial.print("CCS811: ");
    Serial.print("eco2=");  Serial.print(eco2);     Serial.print(" ppm  ");
    Serial.print("etvoc="); Serial.print(etvoc);    Serial.print(" ppb  ");
    Serial.println();

    // People density estimation
    // Assuming 400ppm as base CO2 level with no people and 40ppm increase per person per hour
    int baseCO2 = 400;
    float ppmPerPerson = 40;
    float peopleDensity = (eco2 - baseCO2) / ppmPerPerson;

    Serial.print("Estimated People Density: ");
    Serial.println(peopleDensity);
  } else if( errstat==CCS811_ERRSTAT_OK_NODATA ) {
    Serial.println("CCS811: waiting for (new) data");
  } else if( errstat & CCS811_ERRSTAT_I2CFAIL ) {
    Serial.println("CCS811: I2C error");
  } else {
    Serial.print("CCS811: errstat="); Serial.print(errstat,HEX);
    Serial.print("="); Serial.println( ccs811.errstat_str(errstat) );
  }

  delay(1000);
}
```

Library:

Webserver Library

I2C library

CCS811 library

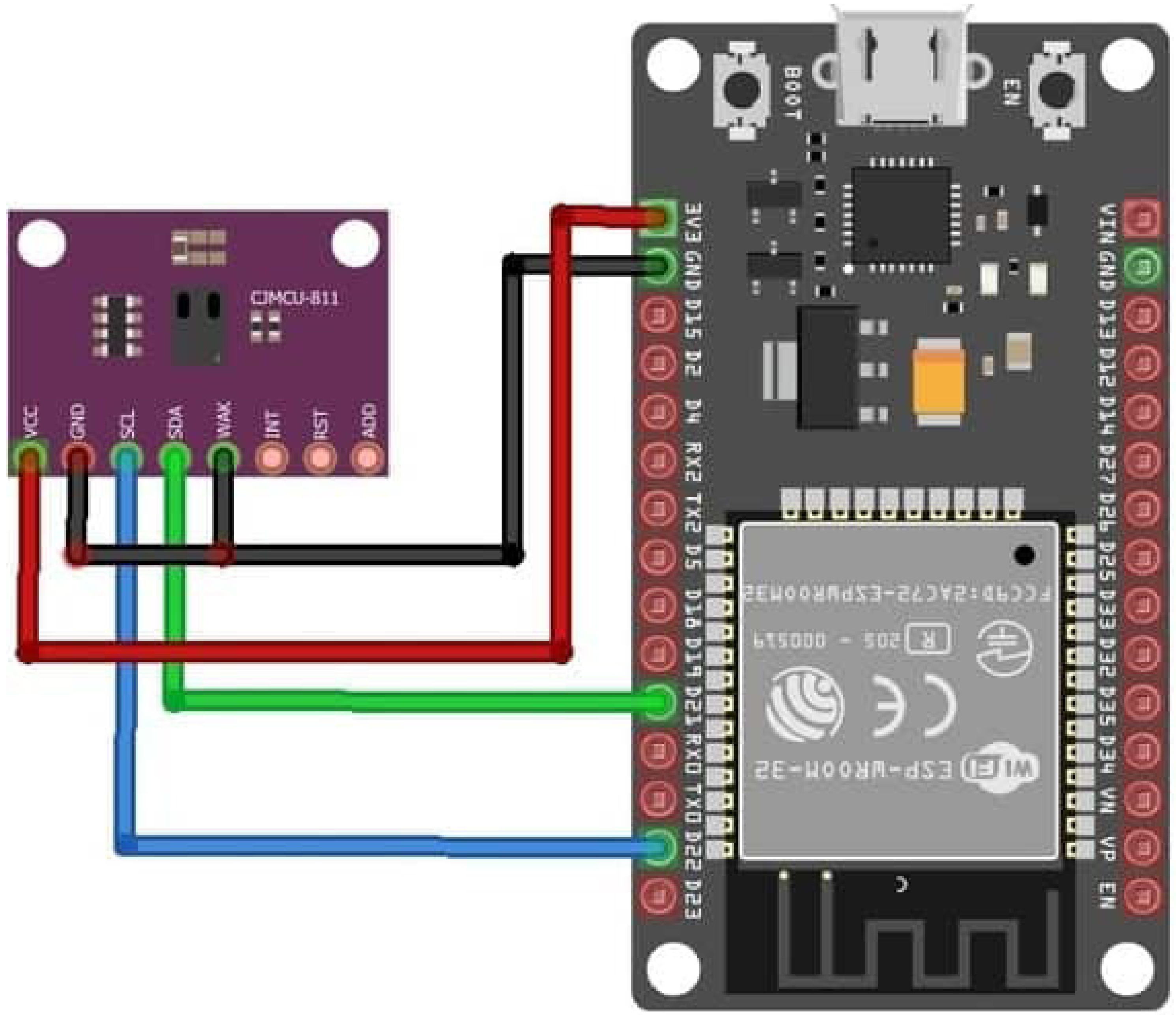
Language:

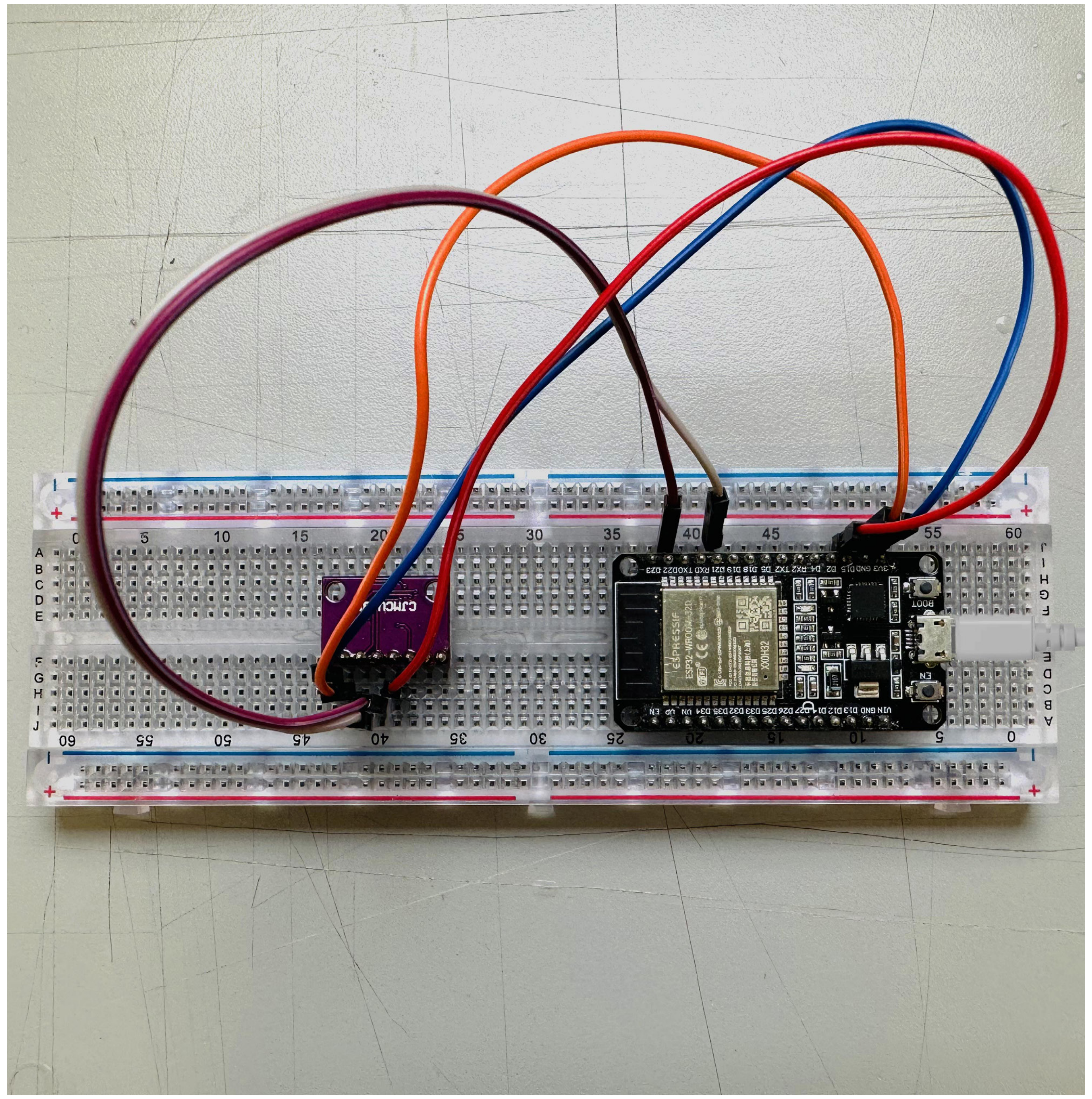
C++

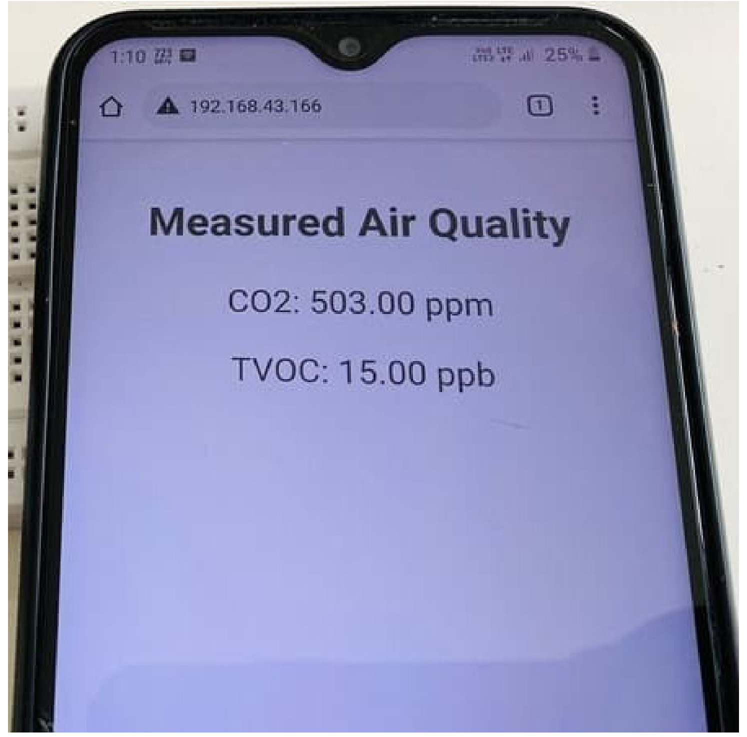
The conversion between CO2 density and human density can be difficult, as it depends on various factors such as the size of the room, ventilation rate, and activity level of the people.

For the purposes of this model, I assume that the density of CO2 and the density of people in the room follow a positive and linear relationship. Based on online research, I am assuming 400ppm as base CO2 level with no people and 40ppm increase per person per hour.

6/ Arduino







Measured Air Quality

CO2: 503.00 ppm

TVOC: 15.00 ppb