Abstract:

Since COVIDE-19 the world is not comfortable as it was before, everyone apprehensive from most places specially the closed places like the closed restaurant, classrooms, waiting rooms. The most able pleases could be controlled is the waiting room, technical help the world to be easier than before, that something could be used to control anything in the public. This idea is helpful to reduce people fears about the closed places.

Introduction:

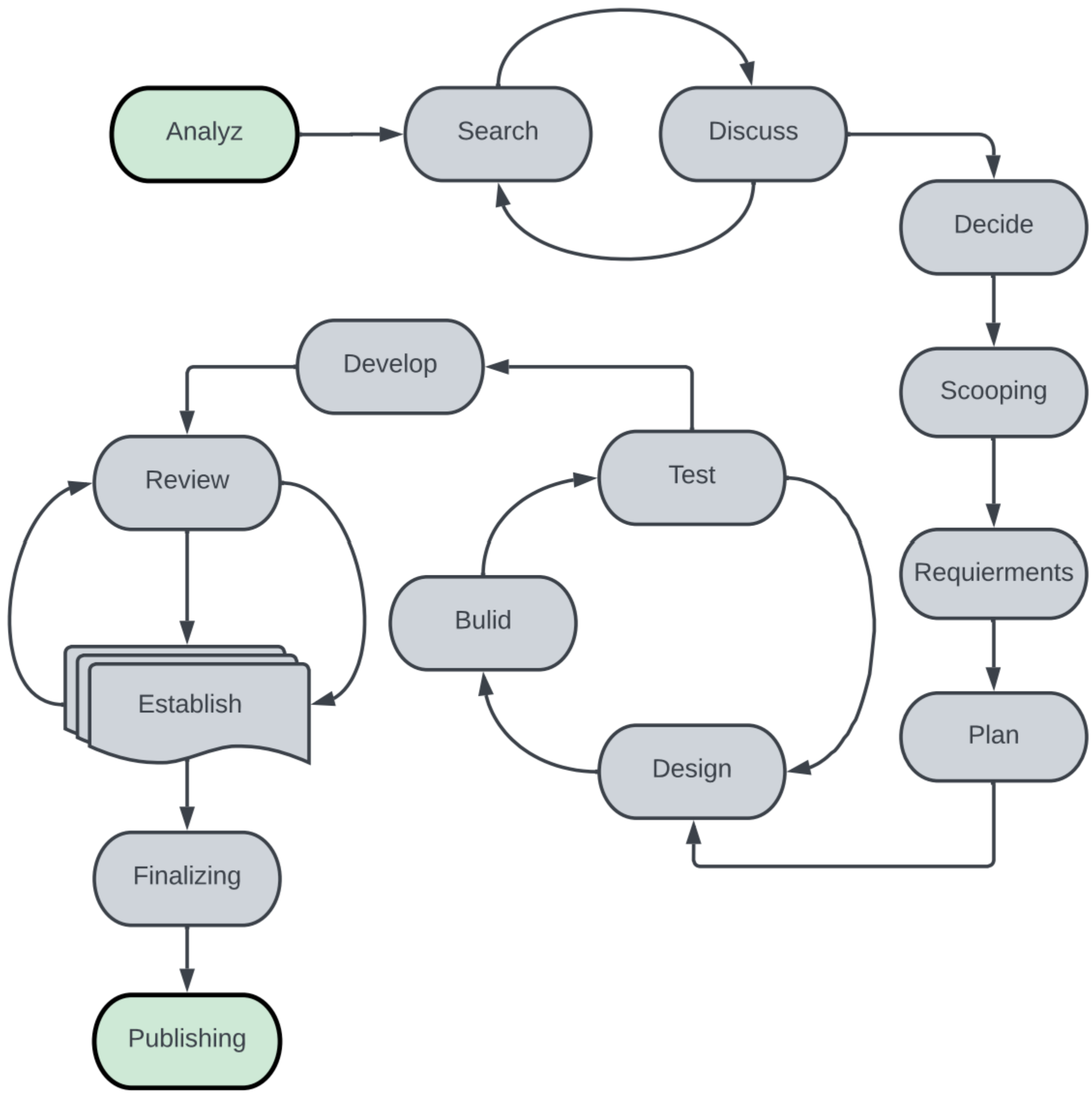
Crowded, this term is used for any place, open or closed, that is dealing with a huge number of people more than it can stand. Crowded places were always a problem but no one understood that clearly. Crowded rooms, as waiting rooms in any organization like hospitals, shopping malls, banks, colleges and universities, etc. Congestion in enclosed spaces has a bad effect on a person's health for many reasons, as most consequences of overcrowding are; suffocation, infection transmission and sometimes may lead to death. After COVID-19, the whole world realized how throngs affect social health. It is one of the biggest proofs for the effects of crowded rooms. Whenever there are overcrowding places, it spreads faster. On the other hand, the number of injured people was increasing in countries without safety precautions and measures for citizens. Corona costs the world more than 6 million people. One of the most important safety recommendations was to stay 2 meters away from the person in front of you and that was really hard for everyone, society, and offices, that is why a quarantine has been imposed in most countries in the world. Moreover, there are many studies dealings with crowded areas, especially in the Emergency Department and how crowding affects negatively on patient’s safety, and how it is evidence for lack of flexibility in hospital capacity, poor organization and slow workforce. Even after the pandemic ended the world did not return to what it was before, at least 25% of people are still wearing masks, 35% keep their distance by 6.5 feet apart.

Problem statement:

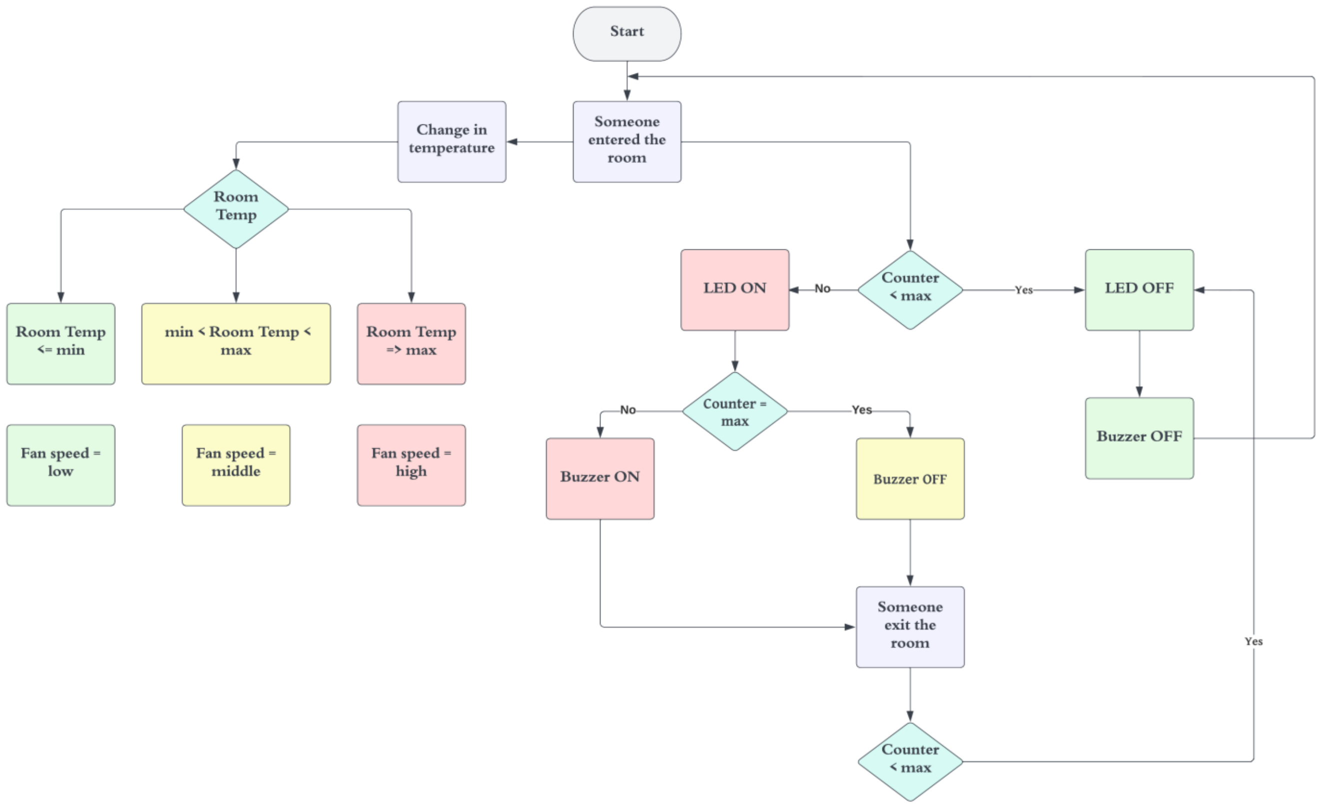
After the danger phase of the pandemic has passed, how to deal with crowds so that we do not go back to the stage of danger again or indirectly incite a new pandemic, especially in enclosed places without imposed quarantine?

The purpose is to reduce the effect of crowds by reducing the number of people in certain places in a technical way starting from waiting rooms.

Design Methodology:



Block Diagram:



Project Scope:

The scope of this project is to reduce the crowd in waiting rooms in any organization in an easy technical way to make the organization able to control the maximum number of people in a room, reduce infection and try to keep the air in an enclosed room as pure as possible. Also, this project can be improved by using the AI to control the room functionality by using specific commands.

Proposed Budget:

Components	price
2x Arduino UNO	48 SAR => 96 SAR
1x RIP Sensor	38 SAR
1x Ultrasonic (HC – SR04)	20 SAR
1x TM1637 4 digit	35 SAR
1x Electrolytic Capacitors	1 SAR
1x Jumper Wires	20 SAR
1x Fan 4*4*2cm 5 volt	20 SAR
1x Fan 4*4*2cm 12 volts	13 SAR
2x Temperature sensor (LM35)	20 SAR
2x Half-size Breadboard	20 SAR
1x 2N2222 Transistor	1 SAR
1x Buzzer	2 SAR
2x Diode 1N4007	1 SAR => 2 SAR

Conclusion:

As a conclusion, smart rooms will make controlling the number of people easy by some techniques, technically the project will count each person entering the room and take an action if the room reaches the maximum number. Moreover, it will be able to refresh the air in the room whenever there is a person inside. All of this will be implemented without human intervention. After adoption of the project by its outcomes that were shown previously. The project could be improved depending on the organization needs, for example, if we want to improve the smart room to be used in hospital, we could add a barcode helping the patient to choose its clinic, a report will send to the reception section in each clinic, shows the patients number that waiting for their turn in this room, this will help the hospital reception to be more efficient. the project will be under the modification and improvement to suit the requirements for each organization, each organization could have its own design useful waiting room depend on its need and scope.

Appendix – Code:

Counter Code:

```
#include <Arduino.h>
#include <TM1637Display.h>
int calibrationTime = 30;   //the time we give the sensor to calibrate (10-60 secs according to the datasheet)
long unsigned int lowIn;    //the time when the sensor outputs a low impulse
long unsigned int pause = 5000;  //the amount of milliseconds the sensor has to be low  //before we assume all motion has stopped
boolean lockLow = true;
boolean takeLowTime;
int pirPin = 6;    //the digital pin connected to the PIR sensor's output
int ledPin = 13;
#define CLK 2
#define DIO 3
#define wechoPin 4
#define wtrigPin 5
#define Bzr 8
int maxppl = 4; // maximmum number of people are allowed to enter the room
int crppl = 0; // current people "counter count var"
long durts1; // variable for the duration of sound wave travel
int distn1; // variable for the distance measurement
uint8_t allDigits[] = {0xff, 0xff, 0xff, 0xff}; // declear all the digits
TM1637Display CN(CLK, DIO); // assign the pins to CN


void setup() {
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(Bzr, OUTPUT);
  digitalWrite(pirPin, LOW);
  CN.setBrightness(0x0f);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      delay(100);
      }
    Serial.println(" done");
    Serial.println("SENSOR ACTIVE");
    delay(1000);
}

void measurDist()
```

```
{
  //w[0] => echo, w[1] => trig
  pinMode(wechoPin, INPUT); //set echo as input
  pinMode(wtrigPin, OUTPUT); // set trig as ouput
  digitalWrite(wtrigPin, LOW); //clear the trig condition
  delayMicroseconds(2);
  digitalWrite(wtrigPin, HIGH); //set the trig condition active at 10microSec
  delayMicroseconds(10);
  digitalWrite(wtrigPin, LOW);
  durts1 = pulseIn(wechoPin, HIGH); //(pin, mode, timeout in microsec)
  distn1 = (durts1*0.034)/2;
 // Serial.print("Distance Sens1: ");
 // Serial.print(distn1);
 // Serial.println(" cm");
}

void pir()
{
      if(digitalRead(pirPin) == HIGH){
       crppl = crppl +1;
       CN.showNumberDec(crppl);
       if(lockLow && crppl != maxppl || crppl == maxppl){
         //makes sure we wait for a transition to LOW before any further output
is made:
         switch(crppl){
           case 1:
           crppl != maxppl;
           digitalWrite(ledPin, LOW);
           break;

           case 2:
           crppl == maxppl;
           digitalWrite(ledPin, HIGH);
           break;

           case 3:
           crppl > maxppl;
            digitalWrite(ledPin, HIGH);   //the led visualizes the sensors output
pin state
            digitalWrite(Bzr, HIGH);
           break;
         }
         lockLow = false;
         Serial.println("---");
         Serial.print("motion detected at ");
         Serial.print(millis()/1000);
         Serial.println(" sec");
```

```
         delay(3000);
         //if(crppl > maxppl){digitalWrite(Bzr, HIGH);}
         }
         takeLowTime = true;
       }

     if(digitalRead(pirPin) == LOW){
       digitalWrite(ledPin, LOW);  //the led visualizes the sensors output pin
state

       if(takeLowTime){
        lowIn = millis();          //save the time of the transition from high to
LOW
        takeLowTime = false;       //make sure this is only done at the start of
a LOW phase
        }
       //if the sensor is low for more than the given pause,
       //we assume that no more motion is going to happen
       if(!lockLow && millis() - lowIn > pause){
           //makes sure this block of code is only executed again after
           //a new motion sequence has been detected
           lockLow = true;
           Serial.print("motion ended at ");      //output
           Serial.print((millis() - pause)/1000);
           Serial.println(" sec");
           delay(4000);
           }
       }
  }


void US()
{
  if(distn1<10  && crppl> maxppl)
  {
    measurDist();
    digitalWrite(Bzr, HIGH);
    crppl = crppl -1;
    CN.showNumberDec(crppl);
    delay(3000);
    } else if (distn1<10 && crppl<=maxppl){
    measurDist();
    digitalWrite(Bzr, LOW);
    crppl = crppl-1;
    CN.showNumberDec(crppl);
    delay(3000);
    }
```

```
}

void loop() {
  CN.showNumberDec(crppl);
 pir();
 measurDist();
 US();
}
```

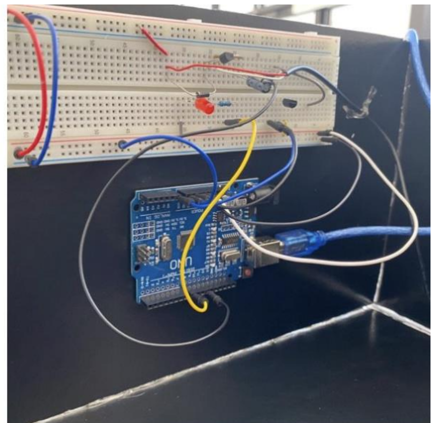
Temperature Code:

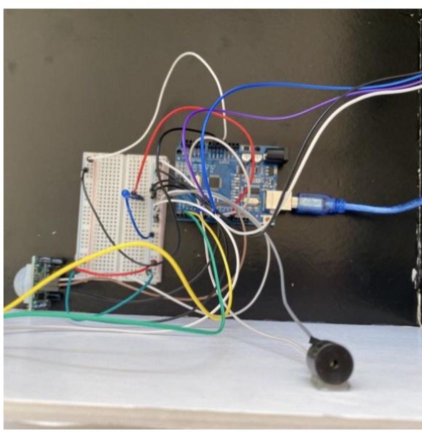
```
#include <Arduino.h>
#define tempPin A0 // the output pin of LM35
#define fan 11 // the pin where fan should be
#define led 8 // led pin
int tempr;
int temp;
int tempMin = 15; // the temperature to start the fan 0%
int tempMax = 27; // the maximum temperature when fan is at 100%
int fanSpeed;
int TMS;

 //pwm impulse with modulation
void setup() {
  pinMode(fan, OUTPUT);
  pinMode(led, OUTPUT);
  Serial.begin(4800);
}

void readTemp() { // get the temperature and convert it to celsius
  pinMode(tempPin, INPUT);
  tempr = analogRead(tempPin);
  temp = tempr * 0.48828125;
  Serial.print("temp: ");
  Serial.print(temp);
  Serial.println("C");
}

void CTMP()
{
  if(temp < tempMin) // if temp is lower than minimum temp
  {
    fanSpeed = 0; // fan is not spinning
    analogWrite(fan, fanSpeed);
    Serial.print("Fan speed: ");
    Serial.print(fanSpeed);
    Serial.println("%");
    digitalWrite(led, LOW);
    digitalWrite(fan, LOW);
  } else if(temp>=tempMin && temp<=tempMax) // if temperature is higher than
minimum temp
  {
    TMS = temp;
    fanSpeed = 1.5*TMS;
    analogWrite(fan, fanSpeed); // spin the fan at the fanSpeed speed
    Serial.println("Fan speed: ");
```

```
    Serial.print(fanSpeed);
    digitalWrite(led, LOW);
    digitalWrite(fan, HIGH);
  } else{
    digitalWrite(led, HIGH);
    digitalWrite(fan, HIGH);}
}

void loop()
{
 readTemp();
 CTMP();

}
```

Appendix – Hardware:

Temperature





Counter

References:

1- (2020). *Room Occupancy Counter*. Hackster.io. https://www.hackster.io/ryanchan/room-occupancy-counter-3b3ffa

2- (2017). *How to use HC-SR04 Ultrasonic range finder with TM1637 display*. Youtube. https://www.youtube.com/watch?v=SlgLj_T5GaY

3- (2019). *How to use motion sensor as a counter with Arduino*. RoboticaDIY. https://roboticadiy.com/how-to-use-motion-sensor-as-a-counter-with-arduino/

4- (2022). *Temperature based fan speed controller using Arduino and LM35*. Circuit Schools. https://www.circuitschools.com/temperature-based-fan-speed-controller-using-arduino-and-lm35/

5- (2022). *Language Reference - Functions, Variables, Structure*. Arduino.cc. https://www.arduino.cc/reference/en/

6- .cc, A. (Director). (2022). *Arduino Datasheet* [Film].