American University of Sharjah

College of Engineering

Department of Computer Engineering

Embedded Systems (COE 410L)

Midterm 2

Smart COVID 19 Detector System

Group 3

Amir Mohideen Basheer Khan - b00074559

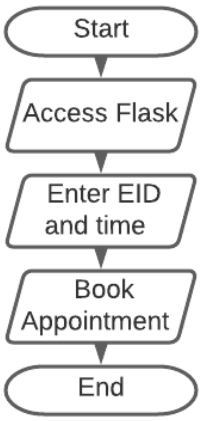
Syed Raza - b00073832

1. Software and Hardware System Requirements:

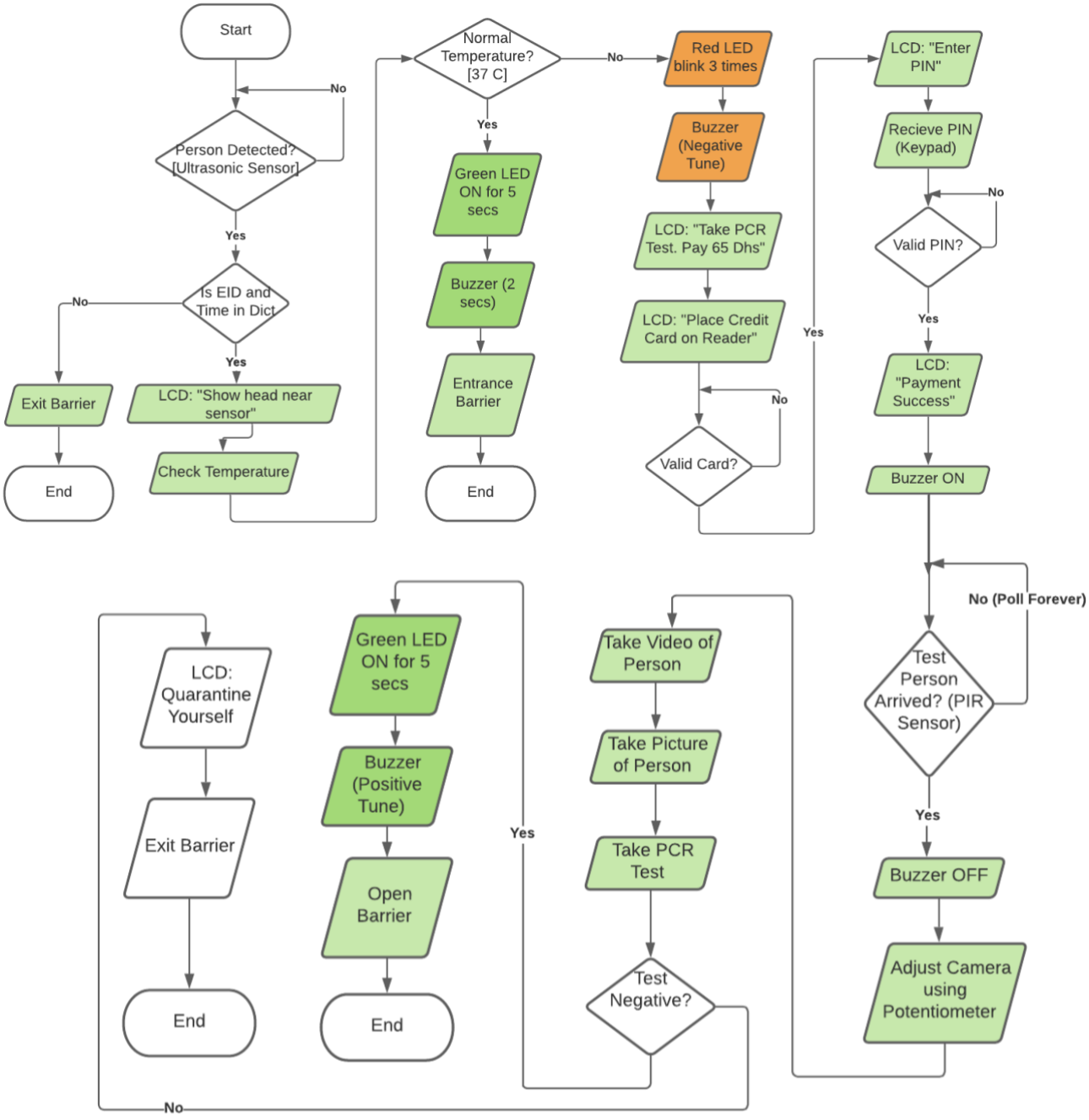
- Sensors:
    - Ultrasonic Sensor (Interrupt) – to detect person
    - Temperature Sensor (ADC) – to check fever
- Digital Inputs:
    - Keypad – To enter credit card PIN
    - RFID – to scan credit card
    - Camera Knob (potentiometer) (ADC) – to adjust camera angle
    - Camera Switch - take picture
- Digital Outputs:
    - LCD – prints instructional and status statements
    - Buzzer (with PWM) – notifies if there is fever or no fever [Digital-to-Analog output, PWM output signal]
    - Red & Green LED – if test result is: positive (RED) or negative (GREEN)
    - Entrance and Exit Barrier (GPIO output) – to open barrier (1) and to close barrier (0)
    - Camera with adjustable angle (with DAC) - capture patient photo and testing procedure video
- Flask Server – to book appointments and view test
    - Index page (welcome and menu)
    - Static address 1 to display available times to book
    - Static address 2 to display safety guidelines
    - Dynamic address 1 to book an appointment results (takes Emirates ID and Time as input)
    - Dynamic address 2 to view test result results (takes Emirates ID and Time as input)

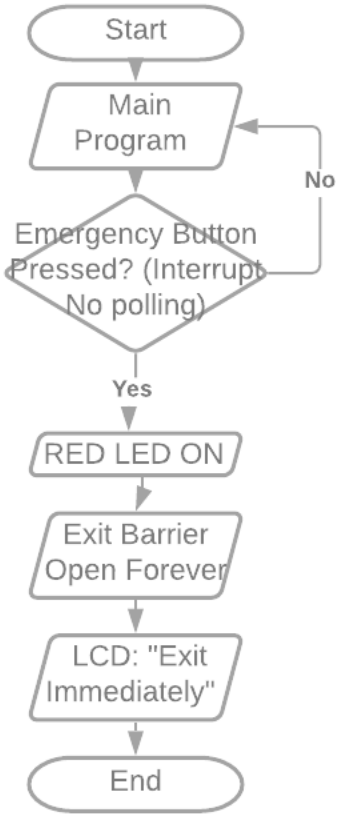
2. Block diagram:

   Software block diagram (Flask Server)

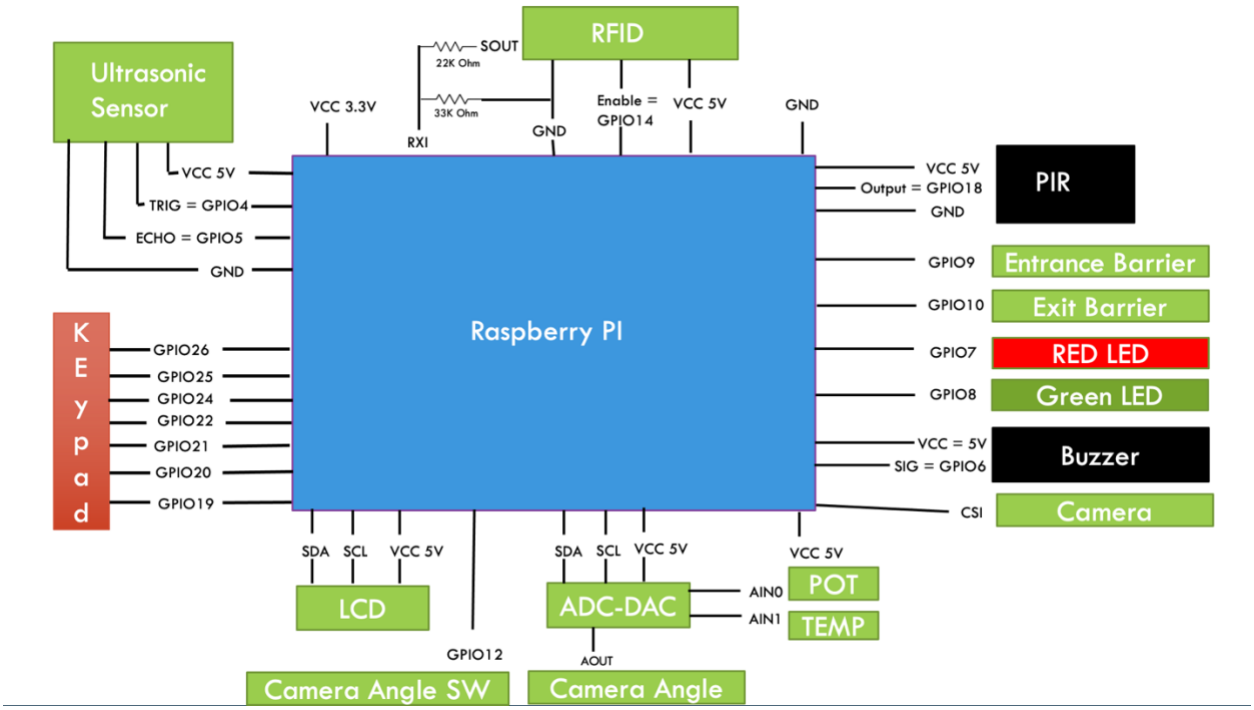


Software block diagram (Program)





Hardware block diagram:



3. Requirements:
    a. Functional Requirements:
        i. Detecting People - Ultrasonic sensor and PIR Motion Sensor detects people
        ii. Reliable Payment Method - RFID verifies credit card
        iii. Fever Detection - Temperature sensor
        iv. Customer Data- RPi camera captures persons photo, video and test result
        v. Flask server to display test result
    b. Non-functional Requirements:
        i. Accuracy: Two layers of testing: Temperature testing + PCR testing
        ii. Promptness: Change in the buzzer tune to call the testing staff whenever a person has to be tested
        iii. Performance: Using the latest RPi for good performance
        iv. User Friendly: Clear Instructions for the person to follow the process seamlessly

4. Code with comments:

```
import RPi.GPIO as GPIO
import LCD1602 as LCD
import PCF8591 as ADC

from picamera import PiCamera
from flask import Flask
from flask import jsonify

import serial
import time
import random

GPIO.setwarnings (False)
GPIO.setmode(GPIO.BCM)

#declaring pins
Trigger = 4
ECHO = 5
Buzzer = 6
LED_Red = 7
LED_Green = 8
EntranceBarrier = 9
ExitBarrier = 10
CameraAngleSwitch = 12
ENABLE = 14
PIR = 18
emergencySwitch = 19


#declaring constants
distance = 100      # 1m distance for ultrasonic sensor
validcreditcard = '5400678912345432'  #sample credit card to check
password = '6196'  #sample credit card PIN
fever = 37
CoronaResult = "Negative"
```

```
booking_availability_dict = {  #dictionary with available booking times for person to attend system
  "1pm": "Yes",
  "2pm": "Yes",
  "3pm": "Yes",
    "4pm": "Yes",
    "5pm": "Yes",
}

booked_eid_dict = { #dictionary that stores eid of person that books at specific time
  "1pm": 0,
  "2pm": 0,
  "3pm": 0,
    "4pm": 0,
    "5pm": 0,
}

# setting up pins as input or output
GPIO.setup (Trigger, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.setup(Buzzer, GPIO.OUT)
GPIO.setup(LED_Red, GPIO.OUT)
GPIO.setup(LED_Green, GPIO.OUT)

GPIO.setup(EntranceBarrier, GPIO.OUT)
GPIO.setup(ExitBarrier, GPIO.OUT)

GPIO.setup(CameraAngleSwitch, GPIO.IN)
GPIO.setup(ENABLE, GPIO.OUT)
GPIO.setup(PIR, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(emergencySwitch,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)

#keypad 4x3
GPIO.setup(19, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(20, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(21, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(22, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```

```
GPIO.setup(24, GPIO.OUT)
GPIO.setup(25, GPIO.OUT)
GPIO.setup(26, GPIO.OUT)

# setting up serial port
SERIAL_PORT = '/dev/ttys0'
ser = serial.Serial(baudrate = 2500,
                    bytesize = serial.EIGHTBITS,
                    parity = SERIAL_PORT,
                    stopbits = serial.STOPBITS_ONE,
                    timeout = 1)

#turn OFF outputs, closing all barriers
GPIO.output(LED_Red, GPIO.LOW)
GPIO.output(LED_Green, GPIO.LOW)
GPIO.output(EntranceBarrier, GPIO.LOW)
GPIO.output(ExitBarrier, GPIO.LOW)
GPIO.output(Trigger, GPIO.LOW)

#initialize PWM
Buzz = GPIO.PWM(Buzzer, 10)
Buzz.start(50)

#initialize LCD
LCD.init(0x27, 1)

#setting up ADC
ADC.setup(0x48)

#creating camera instance
myCamera = PiCamera()


#--------------------------------FLASK---------------------------#

#default index that displays welcome message and menu of choices
app= Flask(__name__)
```

```
@app.route("/")
def hello():
    return "Welcome to Smart COVID 19 Detector System!"
    + "\n Go to 'View_Available_Booking' to view available appointments" +
    "\n Go to 'safety_policies' to view COVID 19 safety policies"
    + "\n Go to 'Book' to book an appointment." +
    + "\n Go to 'view_result' to view your test result"

#displays all the time when system is free for person to attend
@myapp.route('/view_booking')
def view_booking():

    for i in booking_availability_dict:
        if booking_availability_dict[i] == "Yes":
            avail.append(i)

    return jsonify(avail)

@myapp.route('/safety_policies)
def safety_policies():
    return "Maintain at least 1 metre distance between you and people coughing or sneezing."
    + "\n Avoid touching your face."
    + "\n Cover your mouth and nose when coughing or sneezing."


@myapp.route('/book/<time>/<userEID>')
def book(time, userEID):
    if booking_availability_dict[time] == "Yes":
        booking_availability_dict[time] = "No"
        intuserEID = int(userEID)
        booked_eid_dict[time] = intuserEID
        return "Booking Appointment Success!"

    else:
        return "Wrong Input or Booking Time Unavailable!"
```

```
@myapp.route('/view_result/<time>/<userEID>')
def view_result(userEID):
    intuserEID = int(userEID)
    if booked_eid_dict[time] == intuserEID:
        return CoronaResult
    else:
        return "EID does not match"


if __name__="__main__":
    app.run(host='0.0.0.0', debug=True, port = 5000)


# ~~~~~~~~~~~~~~ FUNCTIONS ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ #

#function to open barrier in case of emergency. This function is called when the emergency switch is pressed and a falling edge is detected
def action(self):

    GPIO.OUTPUT(LED_Red,GPIO.HIGH) #turn on red led
    LCD.clear()
    LCD.write(0, 0, 'Exit')
    LCD.write(0, 1, 'Immediately')
    EntranceBarrier(1)     # open Entrance barrier


def isPersonThere():

    step = 1
    # generating pulse of 10us
    GPIO.output(Trigger, 0)
    time.sleep(0.000002)
    GPIO.output(Trigger, 1)
    time.sleep(0.00001)
    GPIO.output(Trigger, 0)

    # read the ECHO pin signal and calculate the distance in cm
    while GPIO.input(ECHO) == 0:
```

```
        x = 0
    time1 = time.time()

    while GPIO.input(ECHO) == 1:
        x = 0
    time2 = time.time()

    duration = time2 - time1
    dist= duration * 1000000 / 58

    # checking if person is in front of the smart system
    if dist <= distance:
        LCD.write(0, 0, 'Welcome')
        time.sleep(1)
        LCD.clear()
        LCD.write(0, 0, 'Show forehead')
        LCD.write(0, 1, 'near Sensor')
        return True
    else:
        return False



def scanCreditCard():
    # get code from serial.read()

    # enabling RFID
    GPIO.output(ENABLE, GPIO.LOW)

    while 1:

        # prompt to scan card
        LCD.clear()
        LCD.write(0, 0, 'Place Credit ')
        LCD.write(0, 1, 'Card on Reader')

        # read 12 bytes from the serial port
```

```
        data = ser.read(12)

        # attempt to validate the data we just read
        s = data.decode("ascii")
        if (len(s) == 12) and (s[0] == "\n") and (s[11] == "/r"):
            code = s[1:-1]
        else:
            code = False
            LCD.clear()
            LCD.write(0, 0, 'Card Invalid')

        # checking if correct card
        if code:
            if (code == validcreditcard):
                LCD.clear()
                LCD.write(0, 0, 'Card Valid')
                return True
            else:
                LCD.clear()
                LCD.write(0, 0, 'Try again...')
                return False

def checkPIN():

    # check if PIN matches
    LCD.clear()
    LCD.write(0, 0, 'Enter PIN:')

    # reading 4 digits pin
    key= Keypad()
    LCD.clear()
    LCD.write(0, 0, 'Enter PIN: *')
    time.sleep(0.5)
    key2= Keypad()
    LCD.clear()
    LCD.write(0, 0, 'Enter PIN: **')
    time.sleep(0.5)
```

```
    key3= Keypad()
    LCD.clear()
    LCD.write(0, 0, 'Enter PIN: ***')
    time.sleep(0.5)
    key4= Keypad()
    LCD.clear()
    LCD.write(0, 0, 'Enter PIN: ****')
    time.sleep(0.5)

    # converting digits to a single string
    Key = str(key) + str(key2) + str(key3) + str(key4)

    # checking PIN
    if (Key == password):
        LCD.clear()
        LCD.write(0, 0, 'Payment Success')
        return True
    else:
        LCD.clear()
        LCD.write(0, 0, 'Incorrect PIN!')
        return False

def OperateCamera(EID):

    #preview to check how camera looks
    start_preview()

    #time to adjust
    time.sleep(2)

    #while the camera adjustment(GPIO 12) button is off, you can adjust the camera angle
    while(GPIO.input(CameraAngleSwitch) == 0):
        POT = ADC.read(0) # read pot ADC value from AIN0
        ADC.write(POT) # write POT value to DAC

    stop_preview()
```

```
    # start video recording
    myCamera.start_recording('/home/pi/Desktop/Vids/%d.h264' % EID)

    # code to take picture
    myCamera.capture('/home/pi/Desktop/Pics/%d.jpg' % EID)

    #stop recording
    myCamera.stop_recording()
    myCamera.close()

def checkTemperature():

    #check if temperature is normal
    tempADC = ADC.read(1) #read temperature ADC value from AIN1
    tempVolts = (tempADC*3.3)/256 #convert to volts
    tempCelsius = tempVolts/0.01 #temperature in degrees Celsius

    #if persons temperature is higher than fever, buzzer is on, red LED fashes twice, PCR test and price is printed on
LCD
    if tempCelsius > fever:

        #buzzer is on
        Buzz.ChangeFrequency(700)
        #red led flash twice
        GPIO.output(LED_Red, GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(LED_Red, GPIO.LOW)
        time.sleep(0.5)

        GPIO.output(LED_Red, GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(LED_Red, GPIO.LOW)
        time.sleep(0.5)
        #buzzer is off
        Buzz.ChangeFrequency(0)

        LCD.clear()
```

```
        LCD.write(0, 0, "Fever Detected!")
        LCD.clear()
        LCD.write(0, 0, 'Take PCR Test')
        LCD.write(0, 1, 'Price: 65 AED')

        return True

    # if persons temperature is less than or equal to fever, Green LED is ON, buzzer is on, print safe message on LCD,
opens barrier

    #Buzzer On for 2 secs
    Buzz.ChangeFrequency(200)
    # make LED Green
    GPIO.output(LED_Green, GPIO.HIGH)
    time.sleep(2)
    GPIO.output(LED_Green, GPIO.LOW)
    #Buzzer Off
    Buzz.ChangeFrequency(0)

    LCD.clear()
    LCD.write(0, 0, "Safe to Proceed!")

    EntranceBarrier(1)     # open Entrance barrier
    time.sleep(6)         # give time for person to exit
    EntranceBarrier(0)     # close Entrance barrier

    return False

def PCRTest(EID):

    LCD.clear()
    LCD.write(0, 0, 'Wait for Staff..')

    Buzz.ChangeFrequency(555) #Buzzer On until staff arrives
    GPIO.wait_for_edge(PIR, GPIO.FALLING)  # wait for PCR Testing Staff to arrive (PIR Sensor)

    LCD.clear()
```

```
    Buzz.ChangeFrequency(0) #buzzer OFF once staff arrives

    OperateCamera(EID)

    # code that randomly determines if person has corona
    hasCorona = bool(random.getrandbits(1))

    # if positive, red LED and print LCD
    if hasCorona:
        CoronaResult = "Positive"
        LCD.write(0, 0, "Test: Positive")
        LCD.clear()
        LCD.write(0, 0, "Quarantine yourself")
        LCD.write(0, 1, "for 10 days")

        ExitBarrier(1) #Open exit barrier
        time.sleep(6) #wait 6 sec for person to exit
        ExitBarrier(0) #close exit barrier

    else:
        CoronaResult = "Negative"

def EntranceBarrier(mode):

    # if 0, then close
    if mode==0:
        GPIO.output(Barrier, GPIO.LOW)
        LCD.clear()
        LCD.write(0, 0, "Closing Entrance")
        LCD.write(0, 1, "Barrier..")
    # if 1, then open
    elif mode==1:
        GPIO.output(Barrier, GPIO.HIGH)
        LCD.clear()
        LCD.write(0, 0, "Opening Entrance")
        LCD.write(0, 1, "Barrier..")
def ExitBarrier(mode):
```

```
    # if 0, then close
    if mode==0:
        GPIO.output(Barrier, GPIO.LOW)
        LCD.clear()
        LCD.write(0, 0, "Closing Exit")
        LCD.write(0, 1, "Barrier..")
    # if 1, then open
    elif mode==1:
        GPIO.output(Barrier, GPIO.HIGH)
        LCD.clear()
        LCD.write(0, 0, "Opening Exit")
        LCD.write(0, 1, "Barrier..")


#keypad function from lab
def Keypad():
    while(True):

        GPIO.output(26, GPIO.LOW)
        GPIO.output(25, GPIO.HIGH)
        GPIO.output(24, GPIO.HIGH)

        if (GPIO.input(22)==0):
            return(1)
            break

        if (GPIO.input(21)==0):
            return(4)
            break

        if (GPIO.input(20)==0):
            return(7)
            break

        if (GPIO.input(19)==0):
            return(0xE)
```

```
        break

    GPIO.output(26, GPIO.HIGH)
    GPIO.output(25, GPIO.LOW)
    GPIO.output(24, GPIO.HIGH)

    if (GPIO.input(22)==0):
        return(2)
        break

    if (GPIO.input(21)==0):
        return(5)
        break

    if (GPIO.input(20)==0):
        return(8)
        break

    if (GPIO.input(19)==0):
        return(0)
        break


    GPIO.output(26, GPIO.HIGH)
    GPIO.output(25, GPIO.HIGH)
    GPIO.output(24, GPIO.LOW)

    if (GPIO.input(22)==0):
        return(3)
        break

    if (GPIO.input(21)==0):
        return(6)
        break
    #Scan row 2
    if (GPIO.input(20)==0):
        return(9)
```

```
            break

        if (GPIO.input(19)==0):
            return(0XF)
            break



# waits for a falling edge to occur and calls the action function to open barries in emergency
GPIO.add_event_detect(19,GPIO.FALLING,callback=action,bouncetime=2000)

# ~~~~~~~~~~~~~~~~~~ MAIN ~~~~~~~~~~~~~~~~~~~~ #


if __name__ == "__main__":

    while True:

        print('detecting people...')

        while(not(isPersonThere())):  # keep looping until person arrives (ultrasonic)
            continue

        #Ask user to input EID and time
        usertime = input('Please enter your Appointment Time [5PM]: ')

        EID = input('Please enter your Emirates ID number [12 digits]: ')
        EID = int(EID)
        if not(EID == booked_eid_dict[usertime] ): # assuming person comes at the right time
            LCD.clear()
            LCD.write(0, 0, "EID does not")
            LCD.write(0, 1, "match Appointment")
            ExitBarrier();
            break;


        #check temperature
```

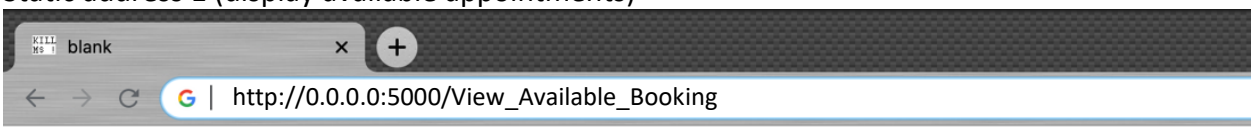
```
    if not(checkTemperature()):  # returns true if person has fever

        break

    #check card
    while(not(scanCreditCard())):
        # indicate error in LCD
        continue

    #check pin
    while(not(checkPIN())):
        continue

    #Take PCR test of person
    PCRTest(EID)

#Person that doesnt have covid leaves
EntranceBarrier(1)              # open barrier
time.sleep(6)               # give time for person to exit
EntranceBarrier(0)              # close barrier
```

5. Sample Output:

Index Page (Welcome and Menu):

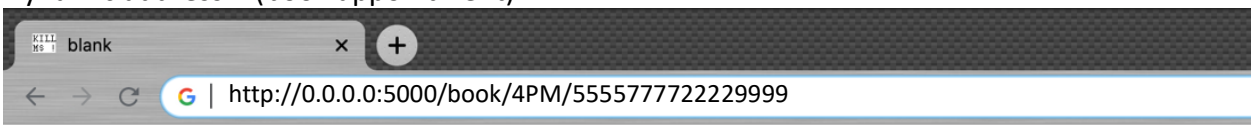
Welcome to Smart COVID 19 Detector System!
Go to 'View_Available_Booking' to view available appointments
Go to 'safety_policies' to view COVID 19 safety policies
Go to 'Book' to book an appointment.
Go to 'view_result' to view your test result

Static address 1 (display available appointments)



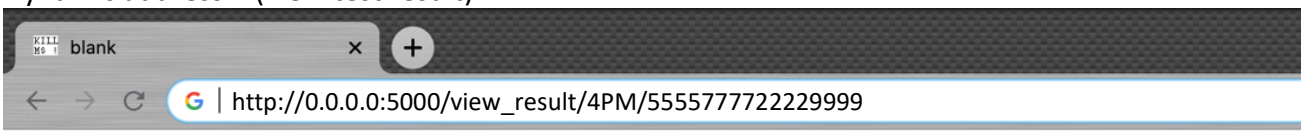
1PM
2PM
4PM
5PM

Dynamic address 1 (book appointment)



Booking Appointment Success!

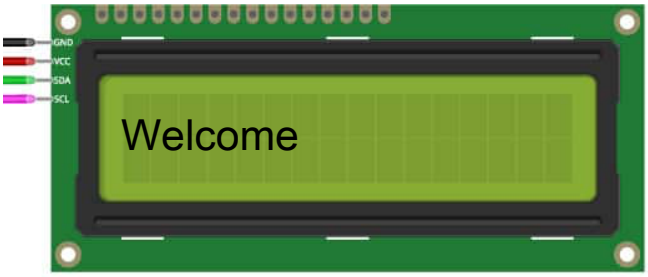
Dynamic address 2 (view test result)

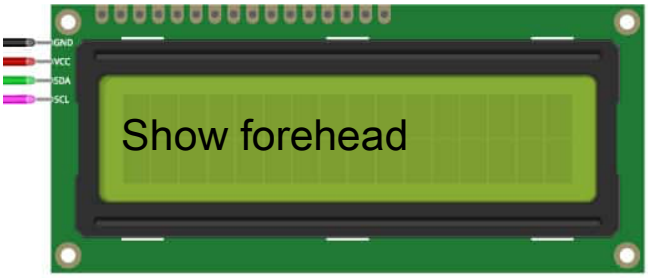


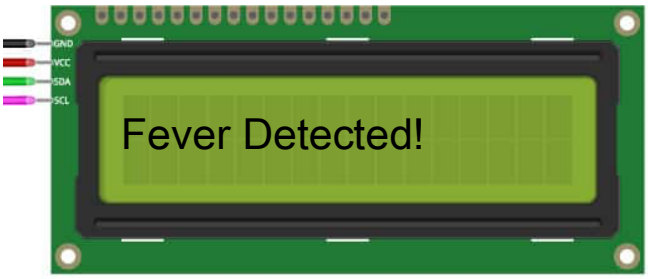
Positive

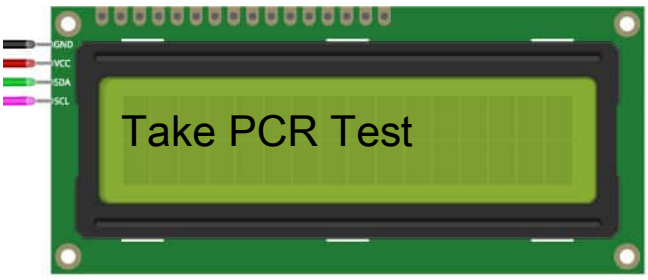
Screenshots shown as per the flow of program in the flow diagram:

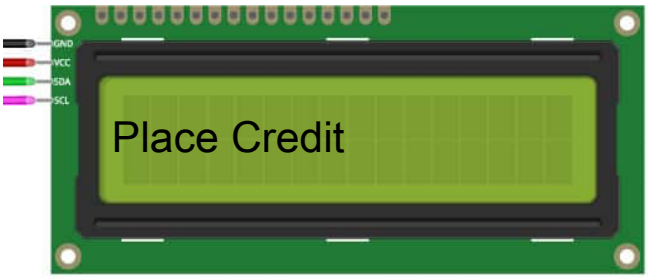
```
Please enter your Appointment Time [E.g., 5PM]:  4PM
Please enter your Emirates ID number [12 digits]:
555577772222999 9
```

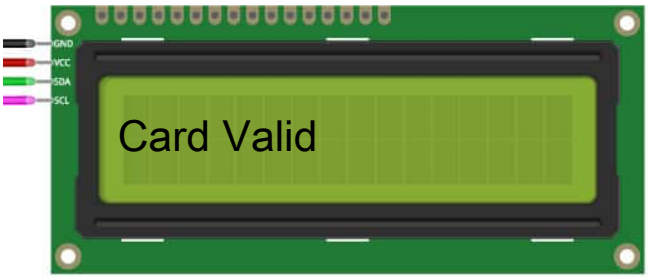


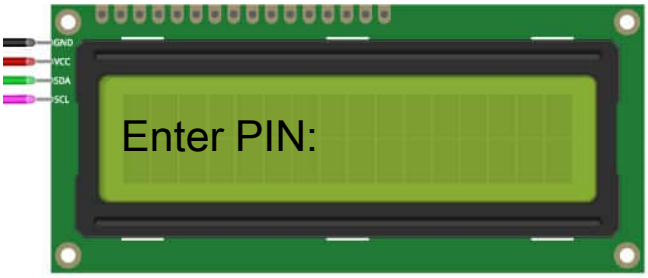


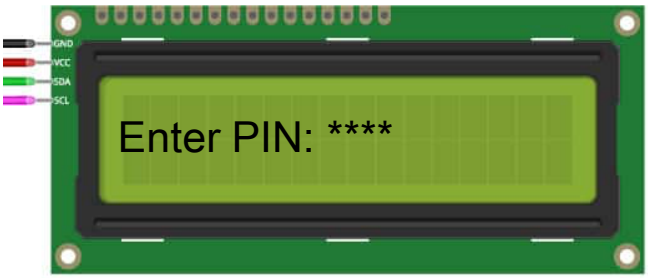




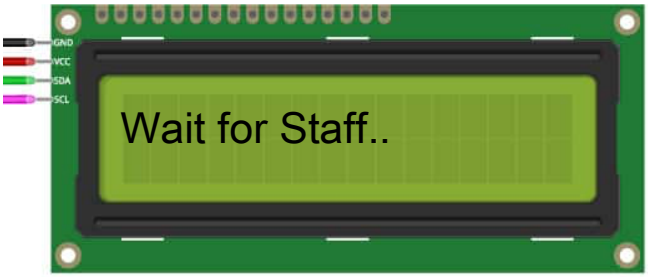



GND
VCC
SDA
SCL
Card Valid

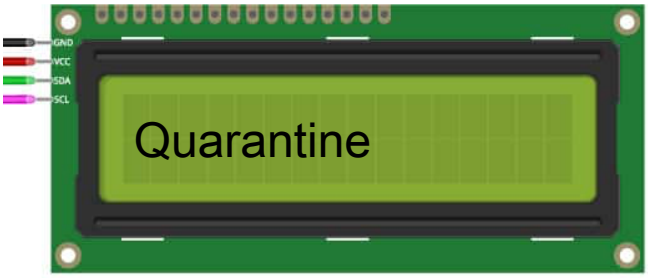

GND
VCC
SDA
SCL
Enter PIN:

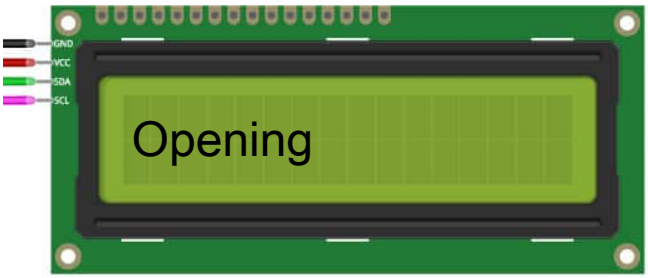

GND
VCC
SDA
SCL
Enter PIN: ****

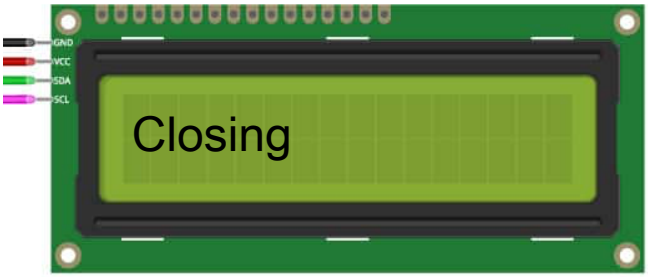

GND
VCC
SDA
SCL
Payment Success


GND
VCC
SDA
SCL
Wait for Staff..


GND
VCC
SDA
SCL
Test: Positive


GND
VCC
SDA
SCL
Quarantine


GND
VCC
SDA
SCL
Opening


GND
VCC
SDA
SCL
Closing