ASSIGNMENT 2 FRONT SHEET

Qualification	**TEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 43: Internet of Things**		
Submission date	17/8/2023	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Dao Xuan Cat	**Student ID**	Bh00339
Class	IT0501	**Assessor name**	Nguyen Thai Cuong
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Cat

Grading grid

P5	P6	P7	M5	M6	D2	D3

❒ Summative Feedback:	❒ Resubmission Feedback:

Grade:	Assessor Signature:	Date:

Internal Verifier's Comments:

Signature & Date:

Table of Contents

I. Introduction

I'm presently work as a product inventor for a new incipiency where you design IoT products for the consumer, commercial, government and defence guests. As part of my part, my director has assigned me to plan and develop a new IoT product, service or operation for a implicit customer. I'm needed to identify a target stoner and conduct tests with this stoner and include this feedback into multiple iterative performances of my product.

II. Develop an IoT application using any combination of hardware, software, data, platforms and services (LO4)

1. Problem

Moment, smart homes are decreasingly developed with temperature operation bias similar as air conditioners and a many other bias. These bias fleetly increase humidity in the air and don't have the function of negativing them. so I developed an operation that can tone- regulate the moisture grounded on the current moisture, lowering the moisture to a position that protects the stoner's health. can also be used for shops because this species is veritably sensitive to humidity in the air.

In this project I will develop a fan humidifier that can be integrated into air conditioners or offices.

2. Solution

I'll have to design and make an operation model that can balance the moisture and regulate the temperature in the air. Because of the variety of guests, I'll choose a addict to balance the temperature, it's cheap and can be integrated into the air conditioner in the future. With large-scale models similar as smart metropolises, the model can also be widely covered by its convenience and health benefits.

III. Hardwares

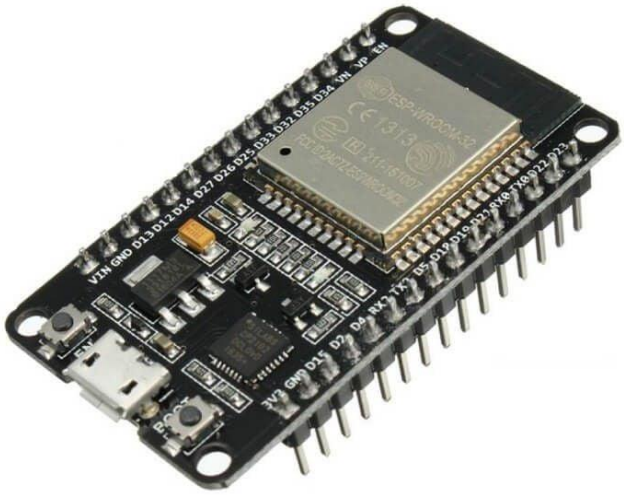

figure 1 esp32

ESP32 is power packed with hardware features. The high speed dual core processors along with the numerous built in peripherals it is set to replace micro-controllers in connected products. The WiFi, Bluetooth Classic and BLE make it great choice to build anything connected. Even if a project does not require a particular feature initially, it could be utilized as required. The built-in hardware accelerator enables secure code storage and securely connecting to the Internet with TLS (SSL)

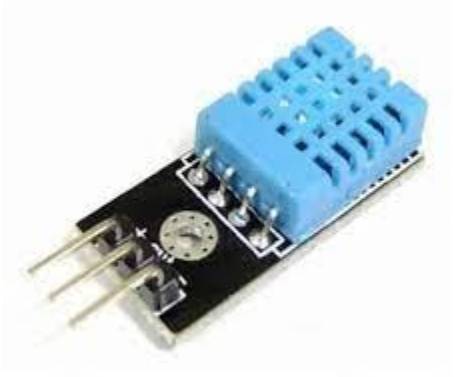

figure 2 dht11

The DHT11 sensor is a basic sensor that lets you measure the current temperature and humidity. This chip combines a thermistor and a humidity sensor and allows you to read both at through its digital output.

figure 3 relay 5V

Useful to control a motor, a led strip, or any other module. How to use it: Just connect a digital output of your board to your relay module, and you can control a power-demanding appliance with the digital signal

figure 4 DC motor

Useful to control a motor, a led strip, or any other module. How to use it: Just connect a digital output of your board to your relay module, and you can control a power-demanding appliance with the digital signal

figure 5 wire

to connect all electric device

1. Software

figure 6 arduino software

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

figure 7 blynk software

Blynk is a low-code IoT software platform for connecting devices to the cloud, building mobile apps to remotely control and monitor them, and managing thousands of users and deployed products. It's a PaaS (Platform-as-a-Service) that helps businesses and individuals seamlessly progress from a prototype of a connected product to its commercial launch and further growth. All Blynk plans include native mobile apps, in addition to all of the other typical IoT infrastructure. With over 400 hardware models support, customers can connect any device to the Internet and use a suite of software products to run commercial projects.

2. Services

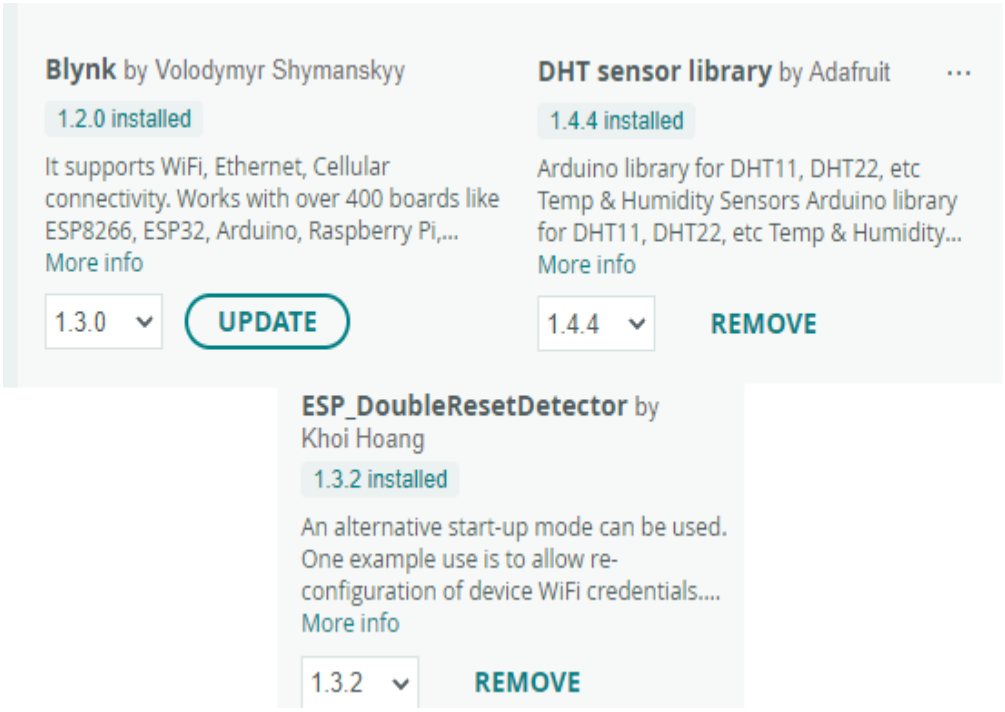

figure 8 library for arduino

some libraries need to control device and connect to blynk

DHT is for control some device like DHT11 and DHT22, the DHT11 will work in this assignment

ESP library to allow the arduino to recognize the ESP 32 device.

IV. Employ an appropriate set of tools to develop your plan into an IoT application (P5)

1. Diagram

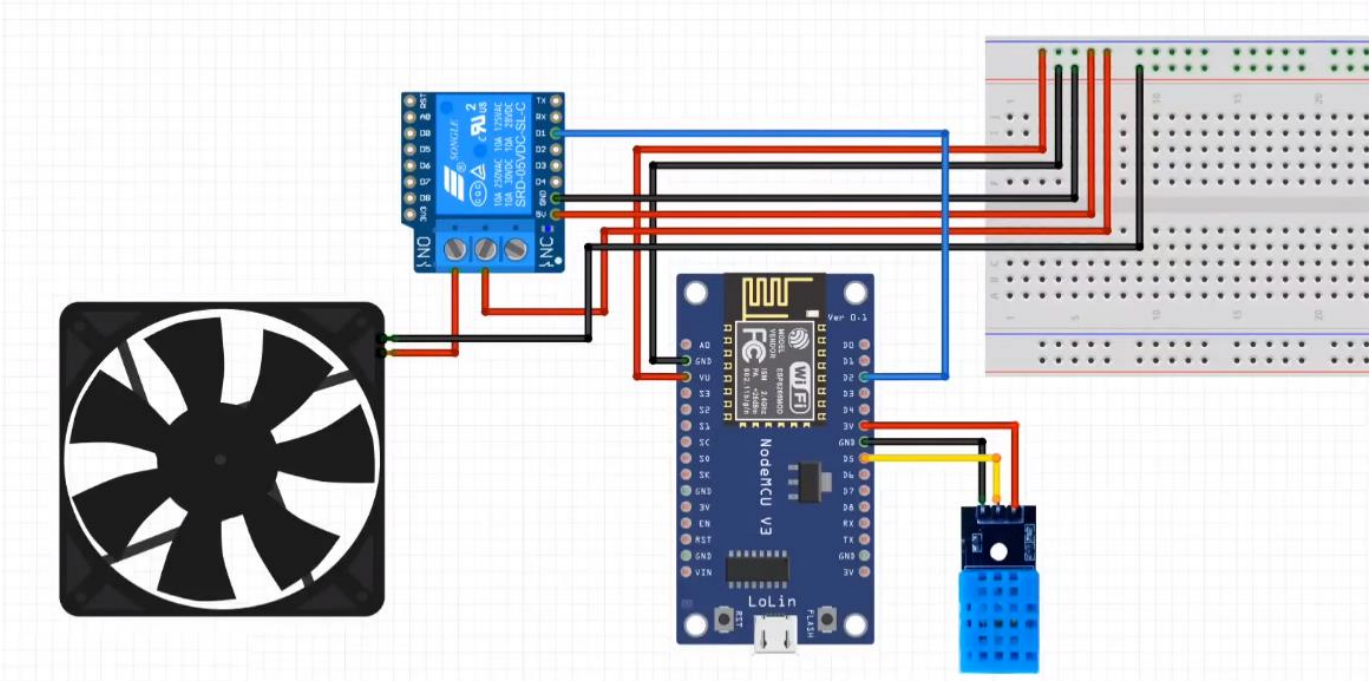

figure 9 diagram setup

you can draw your project in draw.io or star uml or etc... to create idea design then apply for physics setup

2. Coding

After prepare some library , now we can import or coding in arduino, here my coding project:

dcmotorcontrol.ino

```
#define BLYNK_TEMPLATE_ID "TMPL6nPXxwOgT"
#define BLYNK_TEMPLATE_NAME "Speed Controlby temp"
#define BLYNK_AUTH_TOKEN "N7lM1-cYozJLt9fG57lbwF2XHrQ3iZQE"

#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#define DHTPIN 5     // Digital pin connected to the DHT sensor

// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
#define FAN_PIN 2   // FAN RELAY

// Initialize DHT sensor.
DHT dht(DHTPIN, DHTTYPE);

char auth[] = "N7lM1-cYozJLt9fG57lbwF2XHrQ3iZQE";
char ssid[] = "Mạng như chó nhai";
char pass[] = "15122001";

float humDHT = 0;
float tempDHT = 0;


```

figure 10 source code

Part 1 is for connect to blynk and call some device like DHT11 to recognize them.

dcmotorcontrol.ino

```

void setup() {
  Serial.begin(115200);
  pinMode(FAN_PIN, OUTPUT);
  digitalWrite(FAN_PIN, LOW);
  Serial.println(F("DHTxx test!"));
  dht.begin();
  Blynk.begin( auth, ssid , pass );
}


void loop() {

  Blynk.run();

  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  humDHT = dht.readHumidity();
  // Read temperature as Celsius (the default)
  tempDHT = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(humDHT) || isnan(tempDHT))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
```

figure 11 source code

Part 2 set the application's information refresh delay, read the returned DHT11 temperature and humidity data

dcmotorcontrol.ino

```
58
59    Serial.print(F("Temperature: "));
60    Serial.print(tempDHT);
61    Serial.print(F("°C "));
62    Serial.println();
63    Serial.print(F("Humidity: "));
64    Serial.print(humDHT);
65    Serial.print(F("%"));
66    Serial.println();
67
68    Serial.println("************************");
69    Serial.println();
70
71    // Compare Threshold value from Blynk and DHT Temperature value.
72    if (humDHT<90)
73    {
74      digitalWrite(FAN_PIN, HIGH);
75      Serial.println("Bat");
76
77  }
78    else {
79      digitalWrite(FAN_PIN, LOW);
80      Serial.println("Tat");
81
82    }
83
84    Blynk.virtualWrite(V1, tempDHT);
85    Blynk.virtualWrite(V2, humDHT);
86  }
87
```

Output Serial Monitor ×

figure 12 source code

compare threshold value with Blynk and DHT value, the If else loop is for project automatic running it self and print value back so we can check that the code running normally or not.

3. Blynk setup

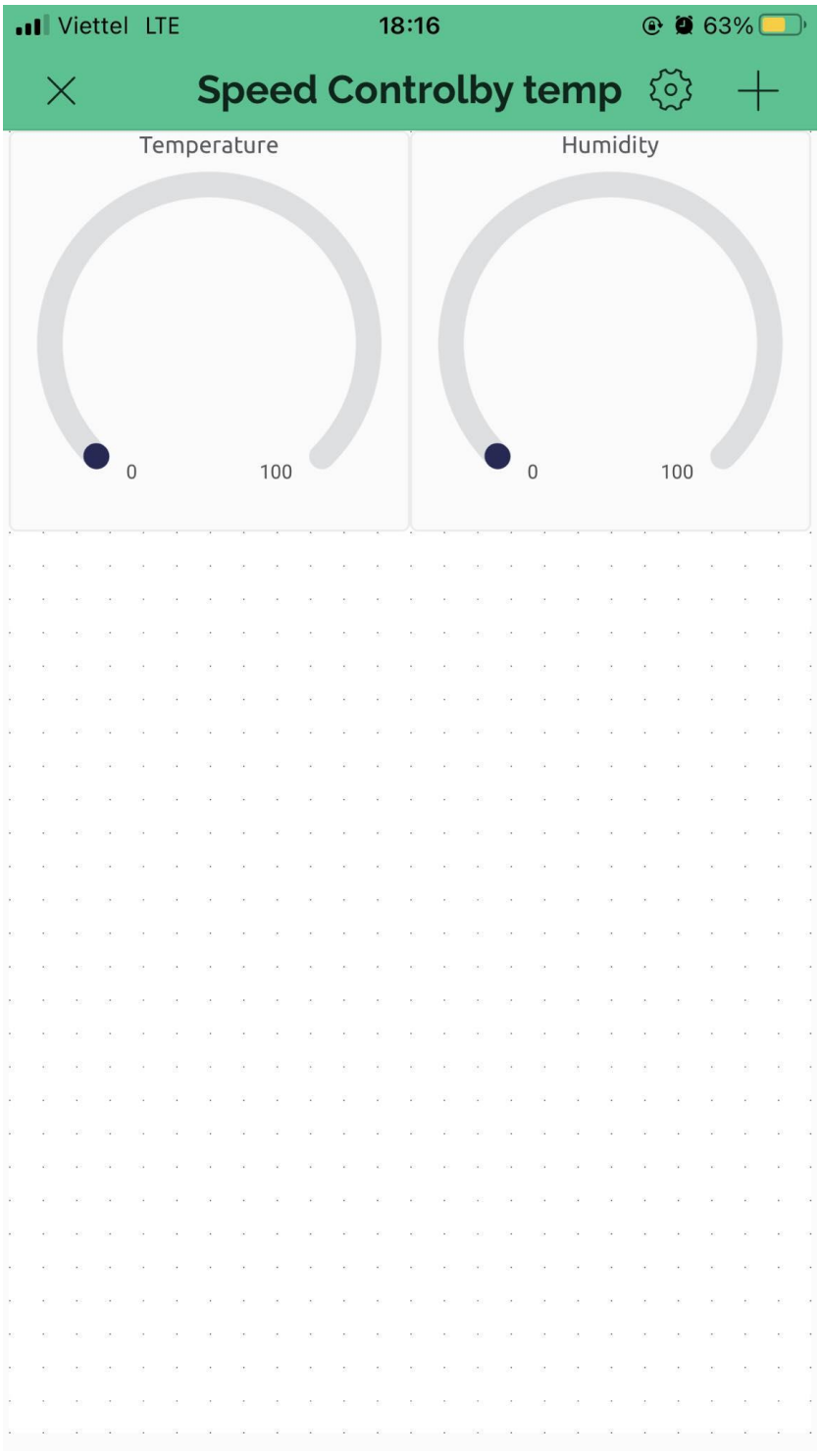

figure 13 Blynk setup

setup some gauge on blynk to report value of tempurature and humidity in your phone.

4. Video demo

My video demo how the project work: https://youtu.be/T8cqHOjLLJk

Source code on github: https://github.com/azizgaming2001/DCmotorcontrolbyhumidity.git

V. Run end user experiments and examines feedback (P6)

1. Survey

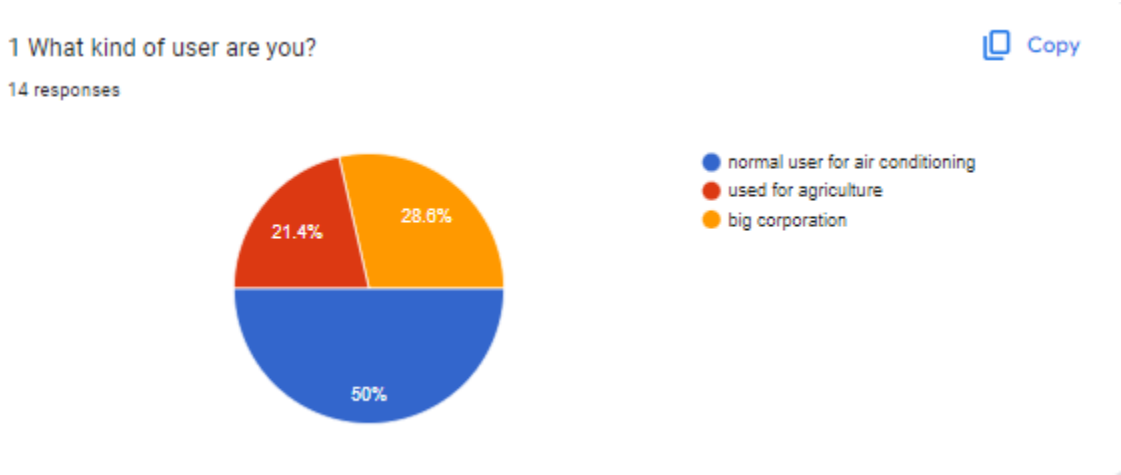

figure 14 question 1

Based on the answers, we see that the majority are individual users, accounting for 50% and for businesses 28.6%, the rest is used in agricultural support.

For this question I want to evaluate users from that to filter out different user groups and possibly develop other versions to suit different types of users.

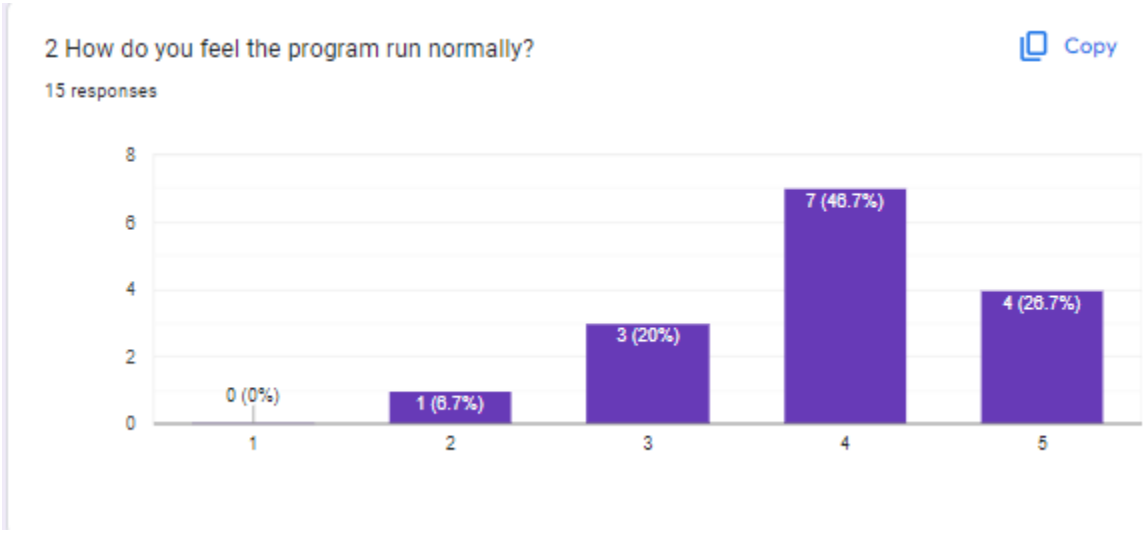

figure 15 question 2

Regarding the second question I would like to receive ratings based on how smooth the application is, so that the delay parameters of the application can be reviewed. with more than 72% of users feeling that the project works smoothly, this result can be temporarily accepted

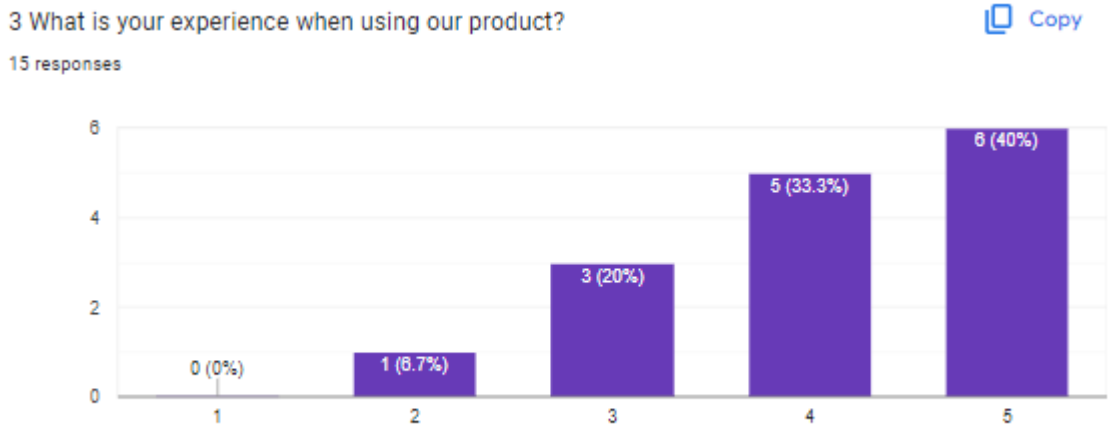

figure 16 question 3

on the question of experience when using. majority of users feel really good to use, only one user has an uncomfortable experience

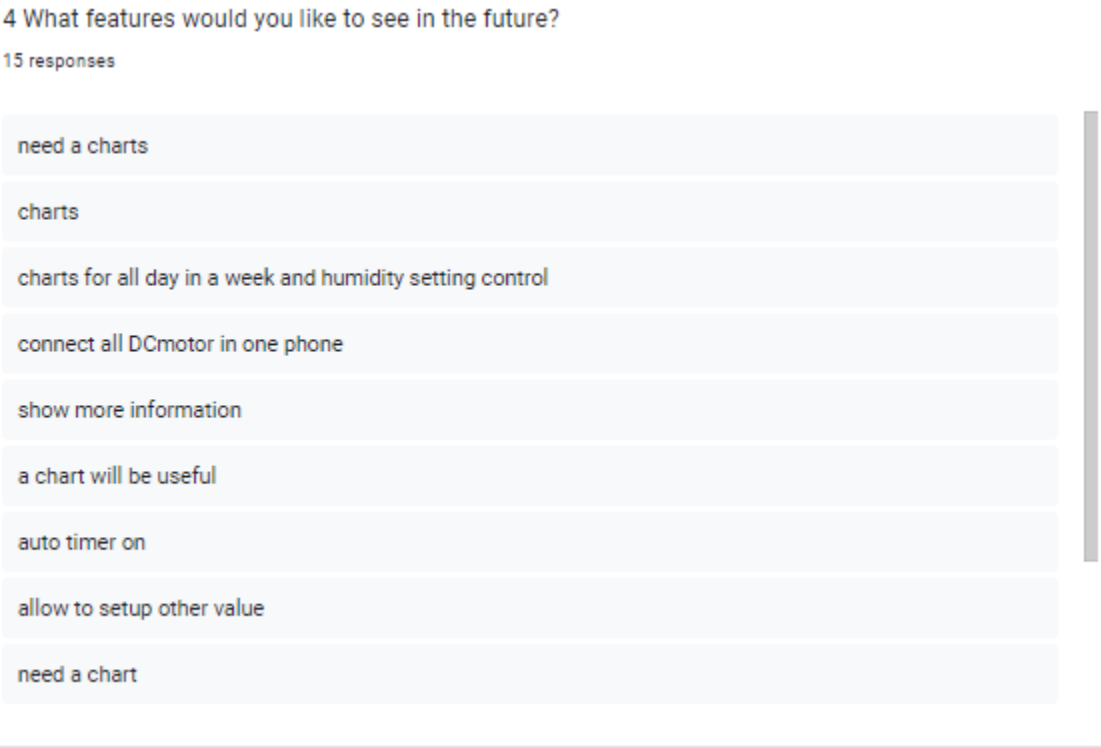

figure 17 question 4

on the last question about what features customers want to be added in the future, most users want to add one or more multi-day report graphs, so I'll come up with an idea to meet this need.

My survey link: https://forms.gle/Fzmp99R751QpwEJ39

2. Product improvement

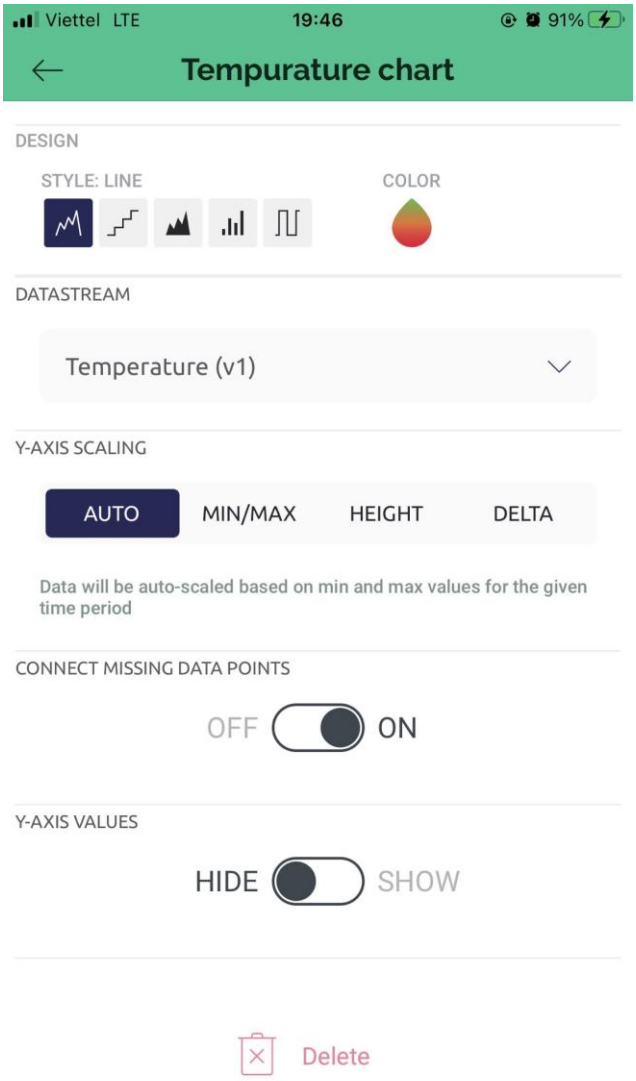

figure 18 add chart tempuratre

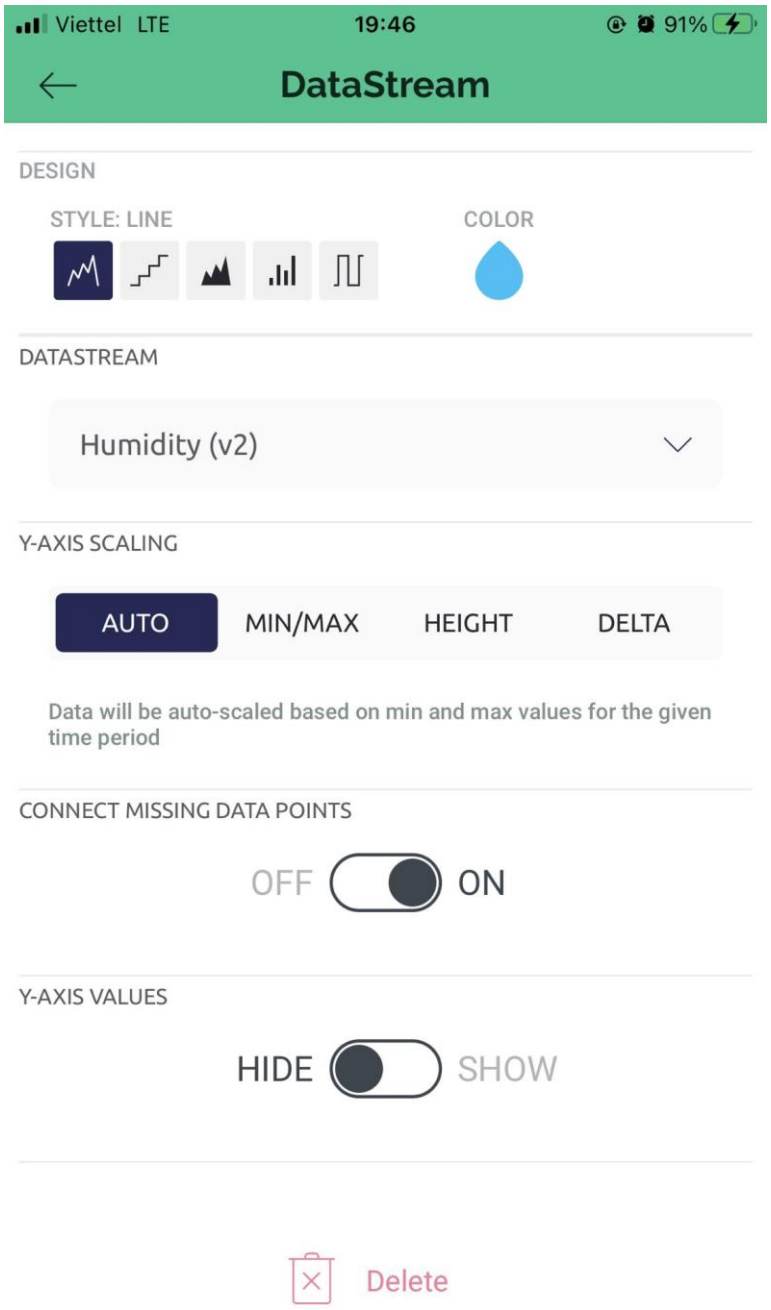

figure 19 add humidity chart

I have add two super charts in blynk cloud to report multi-day for user.

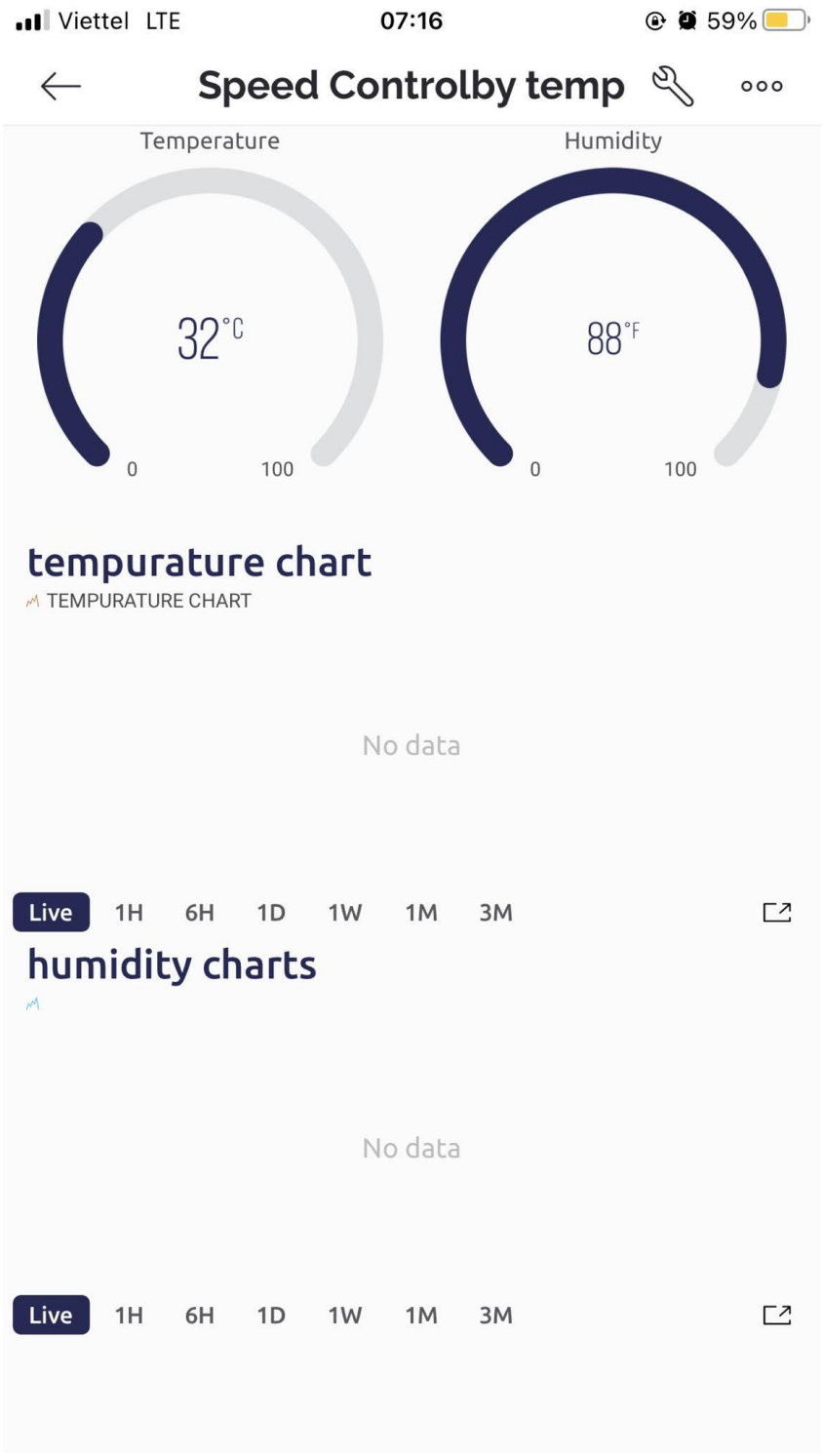

figure 20 multi-day charts

For the convenience of covering the temperature and moisture range of the weekdays, I've added a custom shadowing section. Our operation, in addition to connecting via electronic bias

similar as laptops, can use 5v dishes similar as phone dishes, so druggies can connect bias at home without demanding to go electronic bias. attached.

VI. Evaluate your IoT application and detail the problem (P6)

while developing the application and testing the test cases I find my application works quite smoothly, meeting the needs of many types of customers. Our problem is that software development is based on average humidity, which is not suitable for many kinds of situations. In the future we may develop the ability to select the desired humidity according to the user's discretion.

test case

Test Case ID	Test Description	Test Steps	Expected Results	Actual Results	Pass/Fail
TC-OM-001	Automatic Activation - High Humidity	1. Simulate high humidity conditions above 90%. 2. Observe the operation model's response.	Operation model detects high humidity, activates the moisture balancing mechanism, and starts regulating temperature.	The operation model successfully detected high humidity, activated the mechanism, and effectively regulated temperature.	Pass
TC-OM-002	Temperature Regulation during High Humidity	1. Simulate high humidity conditions. 2. Monitor the temperature regulation process.	Operation model balances moisture and regulates temperature effectively during high humidity, maintaining a comfortable environment.	The temperature regulation process was smooth and effective during high humidity, ensuring a comfortable environment for occupants.	Pass
TC-OM-003	Automatic Activation - Low Humidity	1. Simulate low humidity conditions below 90%. 2. Observe the operation	Operation model detects low humidity, activates the moisture balancing	The operation model promptly responded to low humidity, initiated the	Pass

		model's response.	mechanism, and starts regulating temperature.	mechanism, and achieved temperature regulation as expected.	
TC-OM-004	Temperature Regulation during Low Humidity	1. Simulate low humidity conditions. 2. Monitor the temperature regulation process.	Operation model balances moisture and regulates temperature effectively during low humidity, ensuring a comfortable environment.	The operation model effectively adjusted temperature during low humidity, maintaining comfort levels and preventing over-drying.	Pass
TC-OM-005	Normal Operation - Moderate Humidity	1. Set humidity within the moderate range. 2. Observe the operation model's behavior.	Operation model remains inactive and maintains a comfortable temperature as humidity is within the acceptable range.	The operation model remained inactive during normal humidity levels, successfully maintaining a comfortable environment.	Pass
TC-OM-006	Activation Threshold Verification - Above 90%	1. Gradually increase humidity levels above 90%. 2. Observe the point at which the operation model activates.	The operation model activates automatically as humidity crosses the threshold of 90%.	The operation model initiated at the expected humidity threshold, ensuring efficient humidity and temperature regulation.	Pass
TC-OM-007	Activation Threshold Verification - Below 90%	1. Gradually decrease humidity levels below 90%.	The operation model deactivates automatically as humidity	The operation model deactivated as humidity dropped	Pass

		2. Observe the point at which the operation model deactivates.	falls below the threshold of 90%.	below the expected threshold, preventing unnecessary operation.	
TC-OM-008	Integration with Air Conditioning System	1. Simulate high humidity. 2. Monitor the interaction between the operation model and the air conditioning system.	The operation model successfully integrates with the air conditioning system, providing enhanced moisture balancing and temperature control.	The operation model seamlessly integrated with the air conditioning system, optimizing moisture and temperature management.	Pass
TC-OM-009	Scalability and Large-Scale Implementation	1. Deploy the operation model in a larger area or smart metropolis-like environment. 2. Observe its effectiveness and coverage.	The operation model demonstrates wide coverage and effectively maintains moisture balance and temperature regulation in the larger environment.	In a larger-scale deployment, the operation model showcased consistent performance, regulating conditions across a wide area.	Pass
TC-OM-010	Health and Convenience Benefits Evaluation	1. Monitor the overall comfort and health of occupants using the operation model. 2. Gather feedback on user experience.	Occupants experience improved comfort, balanced moisture, and regulated temperature, contributing to overall well-being.	Users reported enhanced comfort, especially during humidity variations, highlighting the positive impact on well-being.	Pass

figure 21 test case table

VII. Evaluate end user feedback from your IoT application (P7)

Grounded on stoner reviews, we have added a map showingmulti-weekly reports and showing further required information.

- development point
- fresh support for connecting to wifi in the room
- Allows selection of temperature and moisture values and issemi-automatic
- No warnings on air quality and moisture yet

We will develop further apps in the future grounded on the below sins

Through stoner reviews, we find that our products support a veritably smart life. we can develop apps in different directions like medical and agrarian. We'll try our stylish to ameliorate our products in the future to meet consumer demand.

VIII. Conclusion

Above I showed you my IoT model, real- world operations. In addition I've also given you installation plates of my design. The videotape running my real- time operation also met prospects, didn't induce crimes and was fully applicable in practice. Our operation products will be further developed in the future, in the near future we will hear to guests' opinions to directly fix the product, furnishing a better stoner experience.

The conducted test cases validate the functionality and effectiveness of the operation model in balancing moisture and regulating temperature based on humidity levels. Actual results align with expected outcomes, showcasing successful activation, seamless integration, and optimized performance. The operation model demonstrates its potential to offer enhanced comfort, convenience, and health benefits, making it a promising solution for maintaining optimal indoor environments for diverse scenarios and larger-scale applications.

IX. References

My video : https://youtu.be/T8cqHOjLLJk

My source code: https://github.com/azizgaming2001/DCmotorcontrolbyhumidity.git

My survey Link: https://forms.gle/Fzmp99R751QpwEJ39