School of Information Technology and Engineering (SITE)

M.Tech (Software Engineering)

Project Report

TEMPERATURE DETECTION AND PREDICTION SYSTEM (SMART UMBRELLA)

Submitted for the Course

SWE4005 Internet of Things

Offered during winter 2017-2018

(Dr.R.K.Nadesh)

by

G.SAI KRISHNA **15MIS0194**
P.BHANU SUDHEER **16MIS0351**

TEAM Number :2

Project Title : **TEMPERATURE DETECTION AND PREDICTION SYSTEM (SMART UMBRELLA)**

1. Introduction

1.1 Background (System Study Details in brief)

1.2 Problem Statement

Now a day, the temperature and humidity changes very rapidly because different factors influencing the climatic changes in the present world. For our own prediction, we need some device, which can smartly sense the temperature and humidity there by providing a solution of taking precautions before there is a change in the temperature. Hence, our smart umbrella plays a major role in the temperature analysis and prediction in a smarter way, which makes the people to take the precautions before there is a rain around the surroundings.

1.3 Importance

The main importance of the system is to predict the temperature and humidity around the surroundings and provide to the user, which make the user to have the precautions over the temperature change, and helps in making the decision of taking the umbrella along with him/her during that time which also helps him in maintaining him dry.

2. Overview and Planning

2.1 Proposed System Overview

A Smart Umbrella that notifies you when it rains and even before the rain. If you are leaving somewhere without your umbrella and you have a need for it then you will be in trouble. Therefore, our new smart umbrella senses the temperature/ humidity and alerts the user, whether to carry an umbrella or not. This helps you in times of urgent needs where you are not affected

by the rain.

The smart umbrella is just a same classic umbrella with attached sensor to detect the humidity/temperature. The sensor detects the humidity for am hours and notifies the user before itself so that the user can easily know they need an umbrella or not. They may also suggest to their friends and neighbors. A humidity/temperature sensor, ESP8266, can easily achieve this.

An umbrella is the last thing you want to be stuck without in the middle of a downpour, and the cost of all those "emergency" umbrellas you buy in a pinch really adds up. If you have been guilty of forgetting your umbrella at home, work, or in the umbrella stands of numerous coffee shops around town, this high-tech version may be just what you need.

The umbrella's smart tech is reflected in its futuristic appearance. The opalescent canopy is constructed from an exclusive material that, according to one of the designers, "makes you feel you are holding an aurora borealis in your hands." This is not the first umbrella to combine new technology with classic design.

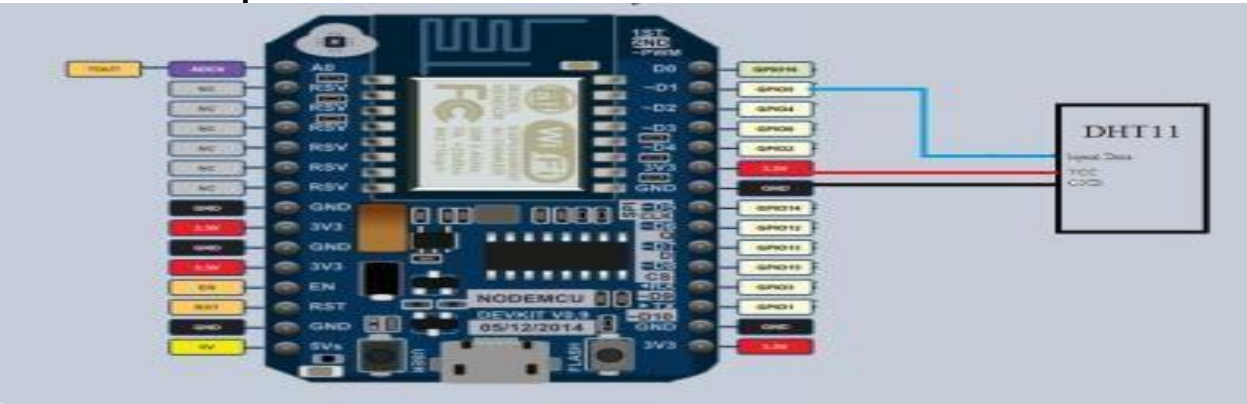
2.2 Challenges:

Every time the predictions that are made may not be true but the accuracy can be about 90%. The Temperature may change very rapidly, which may fail the systems predictions. This system can be reliable at a large extend for the predictions of the during the rainy season and summer season. The main challenge is to provide with the continuous electric supply to the smart umbrella for the better prediction and analysis of the temperature and humidity.

2.3 Assumptions:

All the predictions are made considering the climate is ideal.

2.4 Architecture Specifications:



2.5 Realistic Constraints and Standards

- Prediction of temperature of is very difficult task because no matter how much time you spend running a computer simulation it will still not be 100% accurate.
- In addition, for IOT devices internet connection is very important when the connection is speed is low the processed data take time to send data to mobile or pc.
- The main challenge is to provide with the continuous electric supply to the smart umbrella for the better prediction and analysis of the temperature and humidity.

- In addition, integration of IOT devices with normal umbrella is difficult. Lack of awareness of IOT devices and system among consumers.
- Security constraints for IOT devices in smart umbrella.
- Investment is light bite high because the consumer has to bare umbrella cost and IOT devices

3. IOT DESIGN METHODOLOGIES:

STEP-1: PURPOSE AND REQUIREMENT SPECIFICATION

PURPOSE: Smart umbrella is a system that allows in determining the temperature and provide the user to carry an umbrella with him or not to keep himself dry.

BEHAVIOUR: The smart umbrella system should have auto and manual modes. In auto mode, the system measures the temperature and humidity around the surroundings, which is used to predict that weather the rainfalls or not.

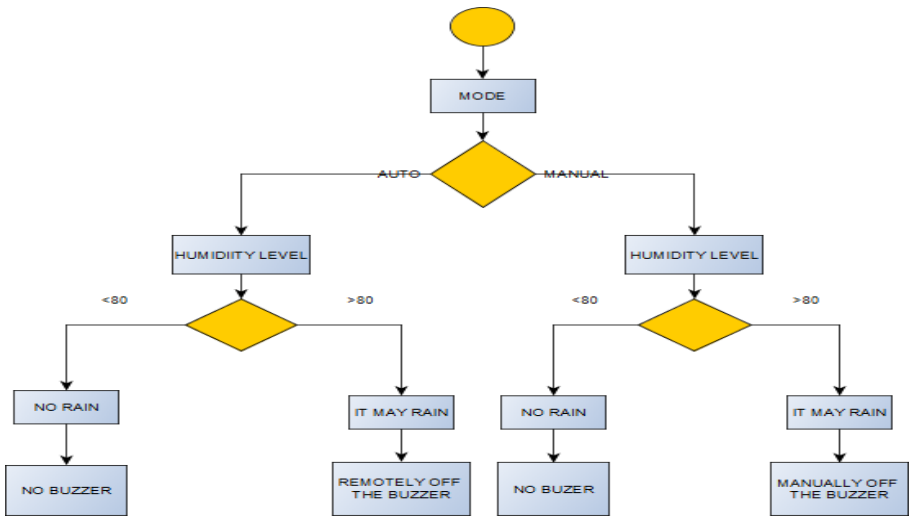
SYSTEM MANAGEMENT REQUIREMENT: The system should provide remote monitoring and control functions.

DATA ANALYSIS REQUIREMENTS: The system should perform local analysis of the data and cloud analysis.

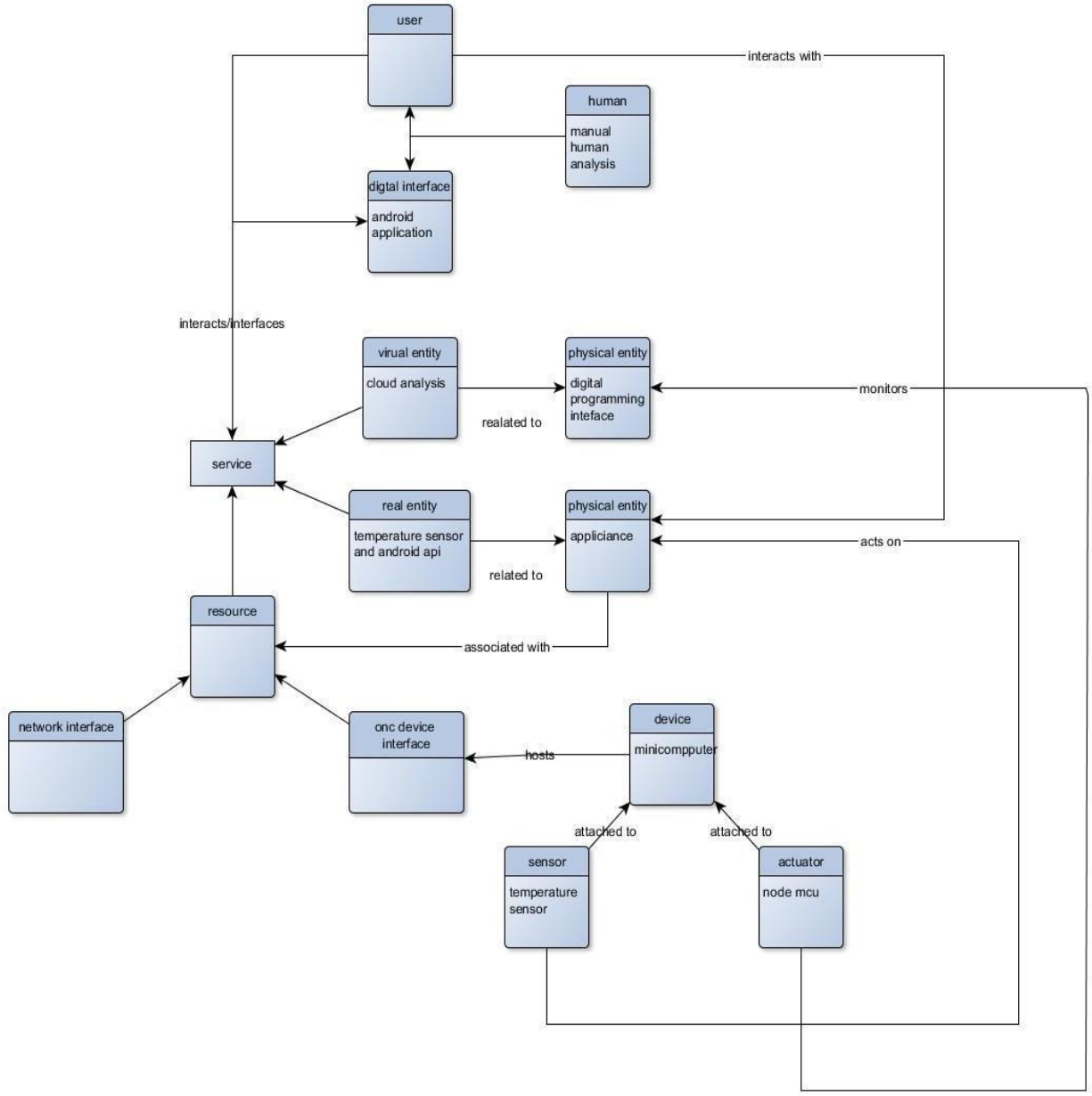
APPLICATION DEPLOYMENT REQUIREMENT: The application should be deployed locally on the device, but should be accessible remotely.

SECURITY REQUIREMENT: The system should have basic user authentication capability.

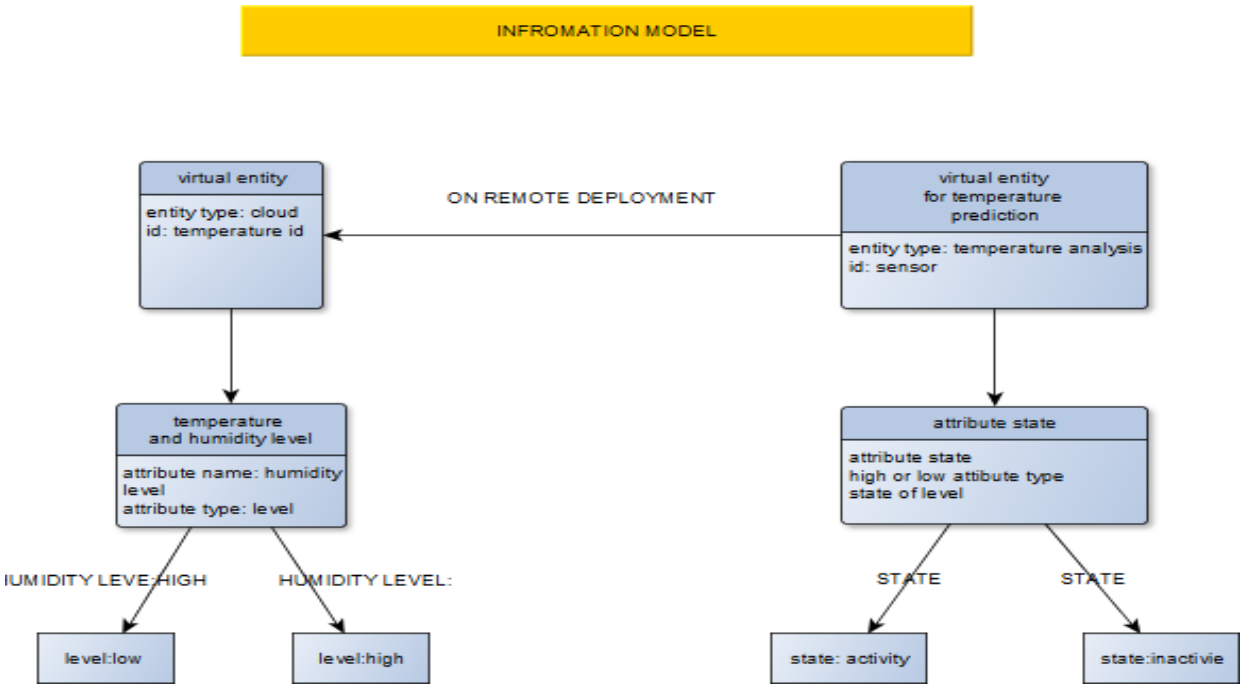
STEP 2: PROCESS MODLE SPECIFICATION:



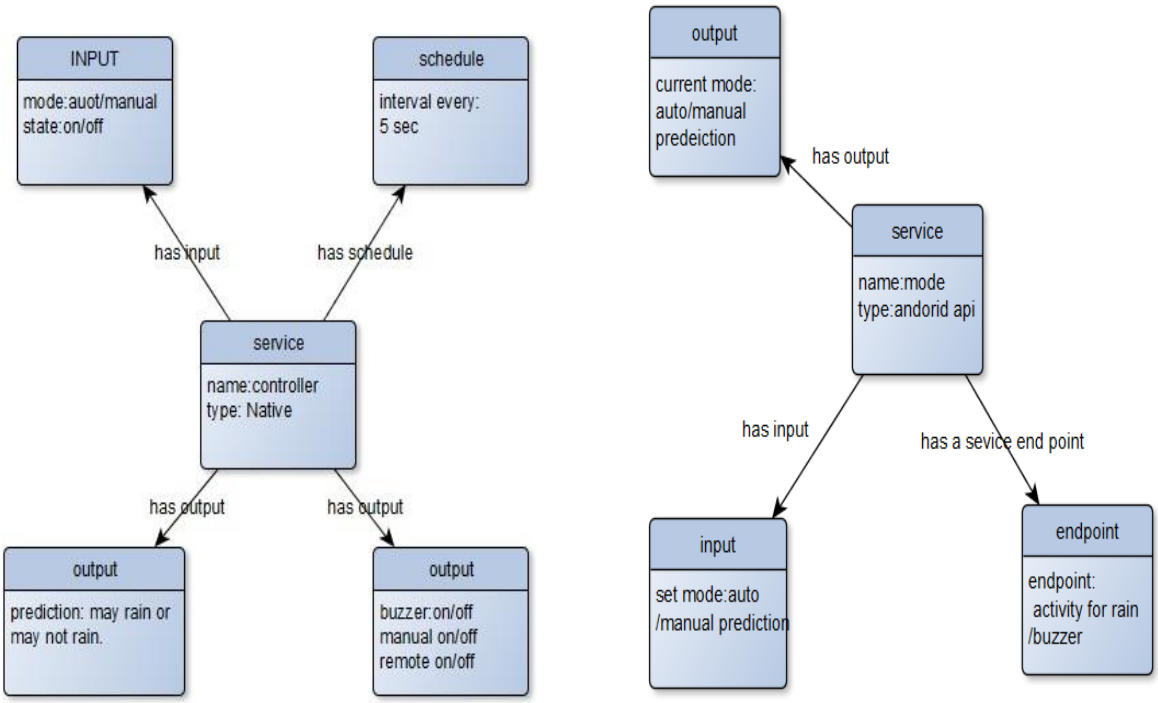
STEP 3: DOMAIN MODEL SPECIFICATION:



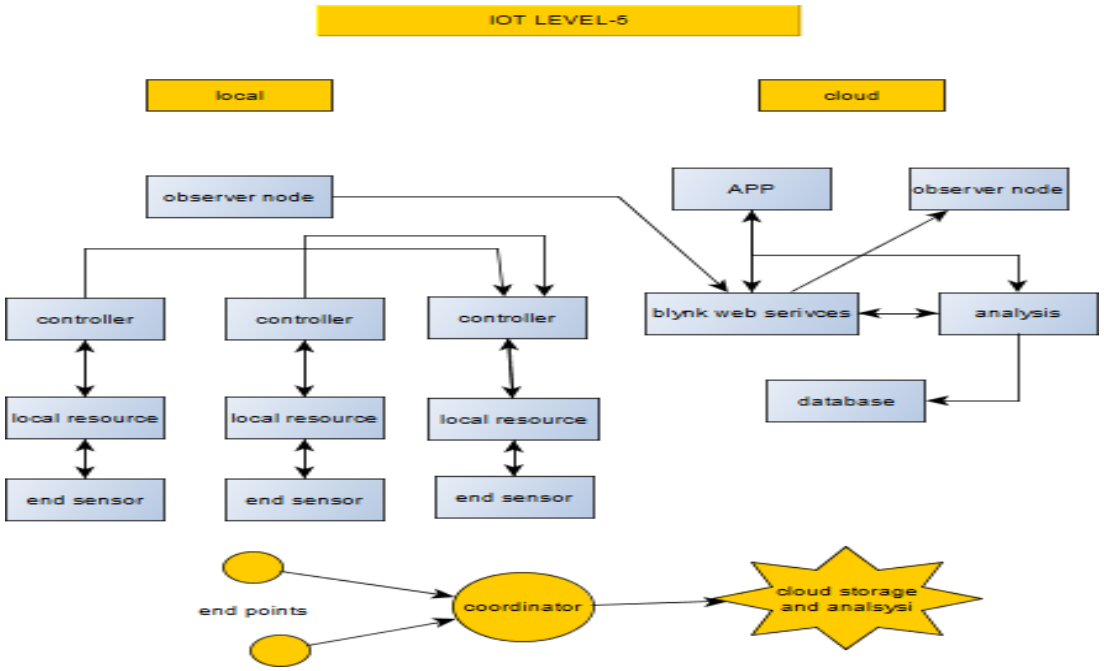
STEP 4: INFORMATION MODEL SPECIFICATION:



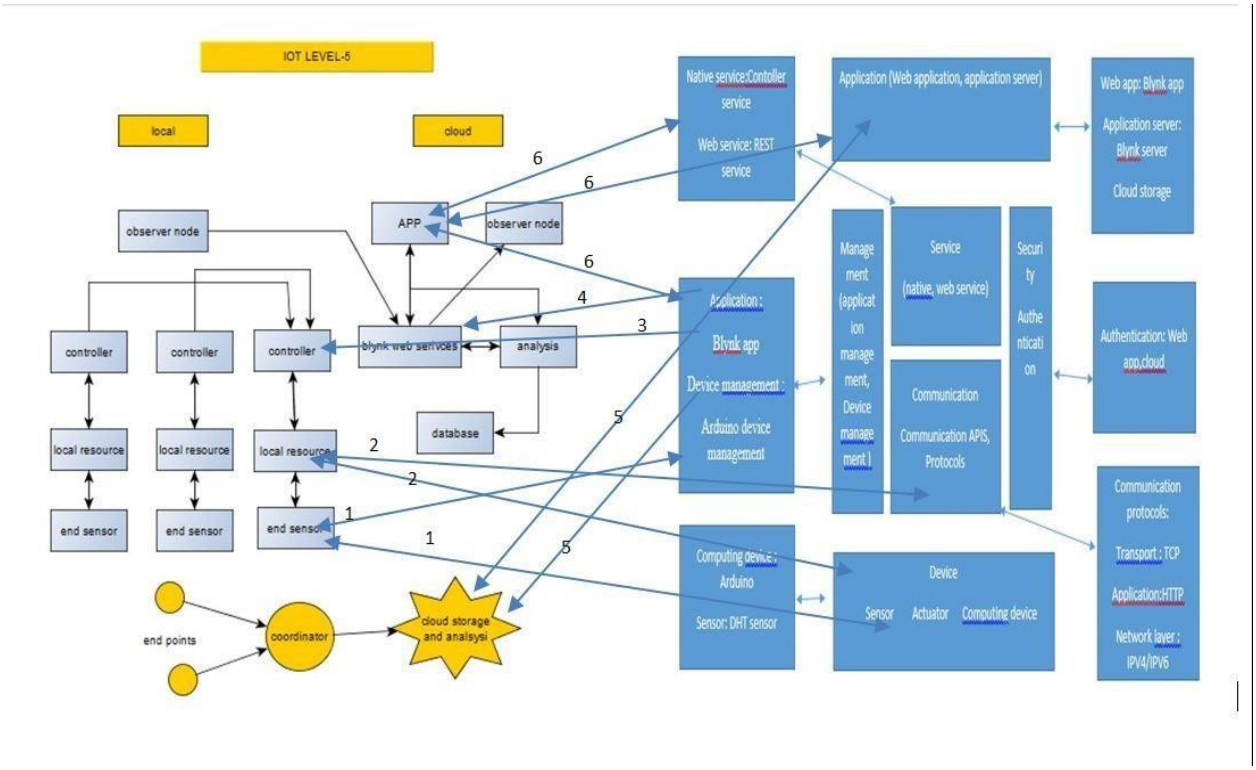
STEP 5: SERVICE SPECIFICATIONS:



STEP 6: IOT LEVEL SPECIFICATION:

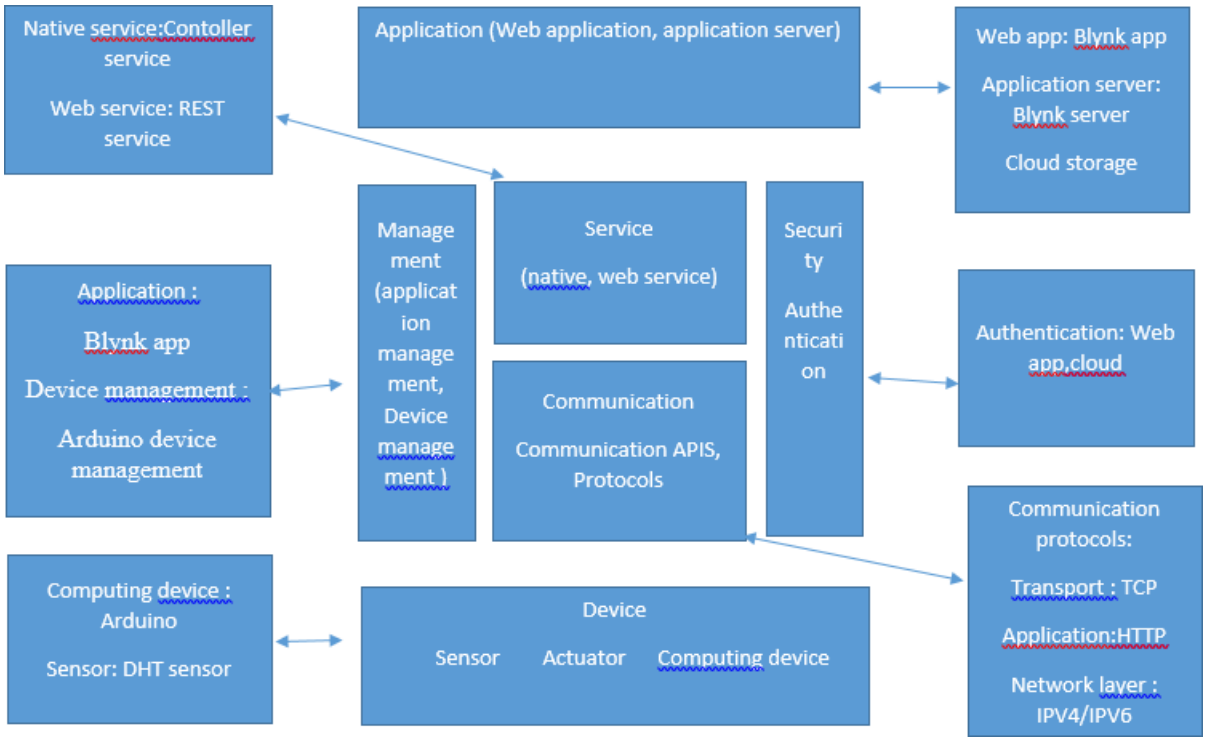


STEP 7: FUNCTIONAL VIEW SPECIFICATION:

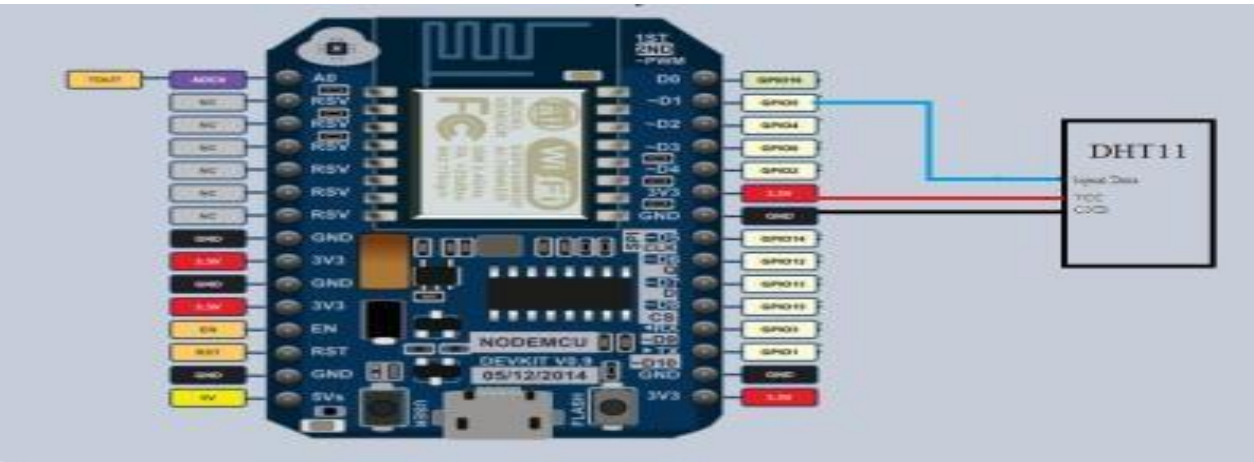


1. IOT devices (DHT Sensor has three pins Vcc, Data, GND) are map with Device (Sensor, Actuator, Computing device) and Management.
2. Resources are map to the device and communication API'S.
3. Controller services map to services.
4. Rest services map to service (web service).
5. Cloud is map to security and services.
6. Application maps to web application and application server.

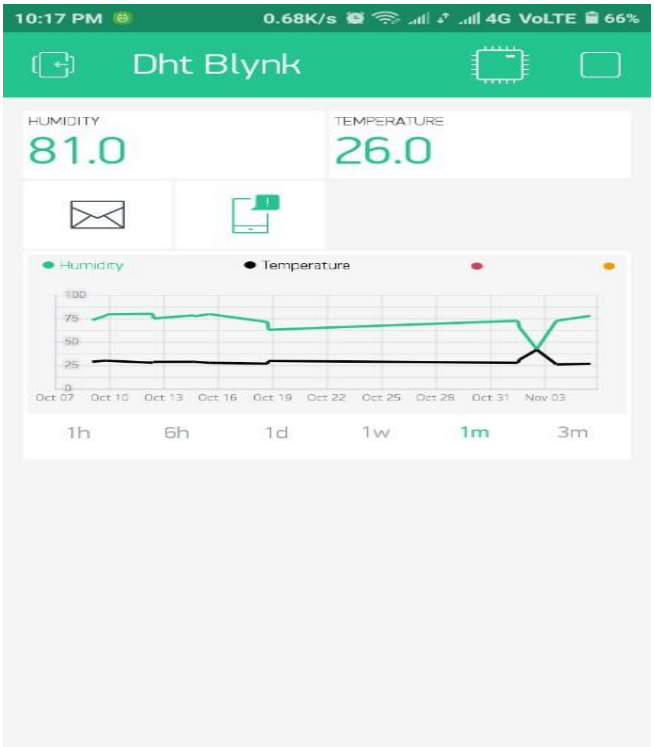
STEP 8: OPERATIONAL VIEW SPECIFICATION:



STEP 9: DEVICE & COMPONENT INTEGRATION:



STEP 10: APPLICATION DEVELOPMENT:



4. System Implementation

4.1 Module Development –Code

```
#include <SPI.h>
#include<ESP8266WiFi.h>
#include <DHT.h>
#include <BlynkSimpleEsp8266.h>
#define DHTPIN 5
#define DHTTYPE DHT11
```

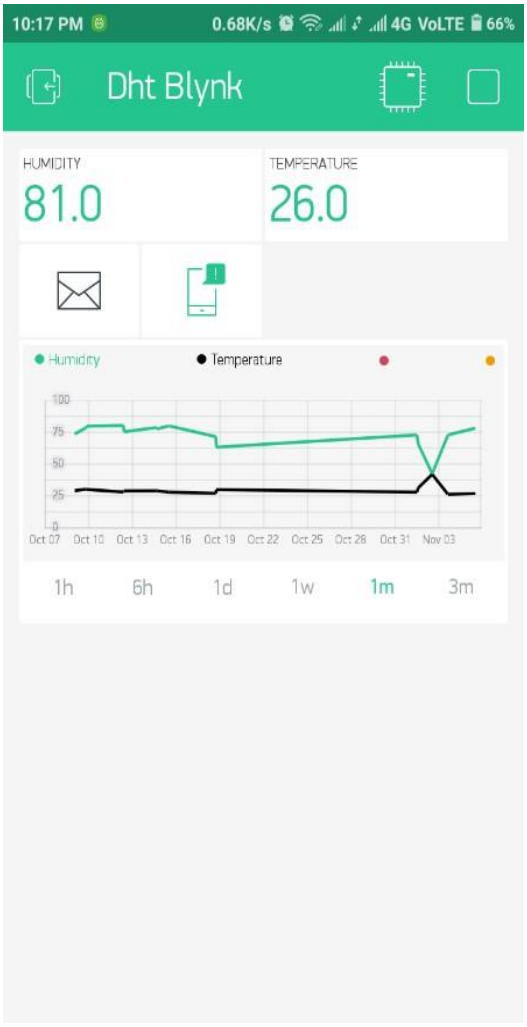
```
#define BLYNK_MAX_SENDBYTES 256

char auth[] = "5cf88c42e90249c5a1aba66448787a0d";
char ssid[]="use";
char pass[]="12345678";

int pin = 4;
int touch=12;
int count=0;
int a=0,b=0,c=0,d=0,hum=0;
int a1=0,b1=0,c1=0,d1=0,tem=0;

DHT dht(DHTPIN, DHTTYPE);

void sendSensor()
{
  int h = dht.readHumidity();
  int t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
  }
  else
  {
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print("Temperature: ");
    Serial.println(t);
  }

  count++;
  if(count==1)
  {
    a=h;
    a1=t;
  }
  else if(count==2)
  {
    b=h;
    b1=t;
  }
  else if(count==3)
  {
    c=h;
    c1=t;
  }
```

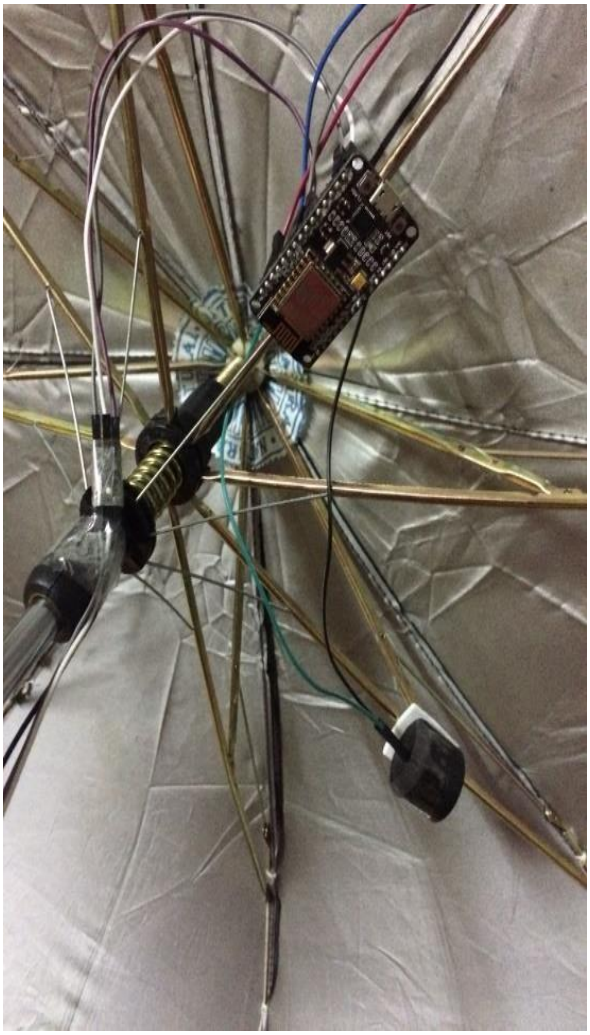
```
  else if(count==4)
  {
    d=h;
    d1=t;
  }
  else
  {
    count=0;
  }
  hum=(a+b+c+d)/4;
  tem=(a1+b1+c1+d1)/4;
  float accuracy = 0;
  int accuracyCount = 0;
  c=pd.read_csv(io.StringIO(s.decode('utf-8')));
  Serial.print(c.apply(lambda b: sum(b.isnull())));
  data=b.dropna();
  for(int i = 0; i < 20; i++){
    randNum = randNumGenerator(testingSet.size());
    while(numsUsedAlready[randNum]){
      randNum = randNumGenerator(testingSet.size());
  }
  float probOver80 = (float)numOver80 / trainingSet.size();
  float probUnder80= (float)numUnder80 / trainingSet.size();
  for(int i = 0; i < sampleSet.size(); i++){
        Humidity sample = sampleSet[i];

        for(int j = 0; j < trainingSet.size(); j++){
           Humidity trained = trainingSet[j];
           if(sample.hum == trained.hum){
              if(trained.value == ">80"){
                 hum++;
              }
              else{
                 hum++;
              }
           }
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6,  t);
  if(h>80)
  {
    Blynk.email("ramjihind12@gmail.com","Temperature Alert","Humidity over 80. Take ur
Umbrella.");
    String text="Humidity over 80. Take ur Umbrella.\n Predicted Humidity: ";
    text+=String(hum);
    text+="\n Predicted Temperature: ";
    text+=String(tem);
```

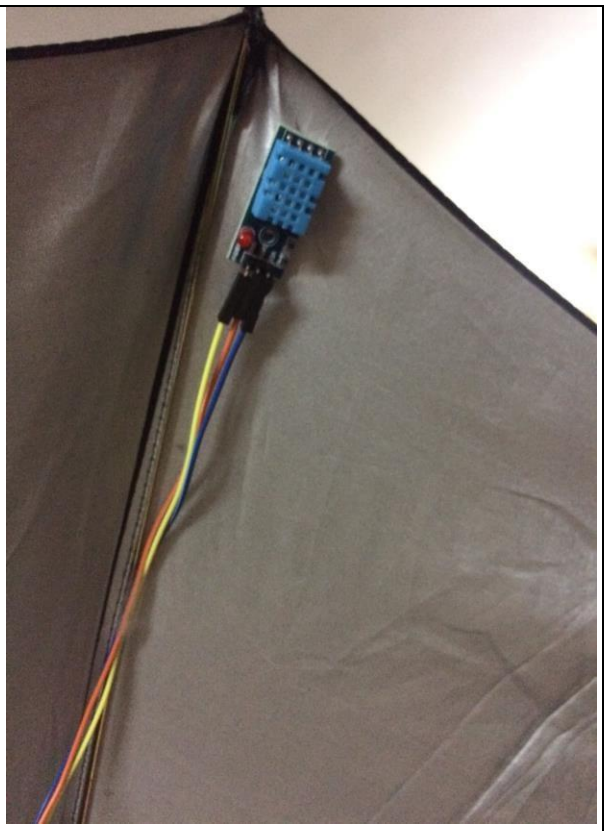
```
    Blynk.notify(text);
    tone(pin,60000);
  }
  else
  {
    Blynk.email("ramjihind12@gmail.com","Temperature Alert","No need for an Umbrella.");
    String text="No need for an Umbrella.\n Predicted Humidity: ";
    text+=String(hum);
    text+="\n Predicted Temperature: ";
    text+=String(tem);
    Blynk.notify(text);
  }
  if(digitalRead(touch)==HIGH)
  {
    noTone(pin);
  }
}
BLYNK_CONNECTED() {
  Blynk.syncVirtual(V1);
}

BLYNK_WRITE(V1)
  {
    int pindata=param.asInt();
    if(pindata==1)
    {
      Serial.println("Buzzer On");
      tone(pin,10000);
    }
    else if(pindata==0)
    {
      Serial.println("Buzzer Off");
      noTone(pin);
    }
  }

void setup()
{
  Serial.begin(9600);
  pinMode(pin,OUTPUT);
  pinMode(touch,INPUT);
  Blynk.begin(auth,ssid,pass);
  dht.begin();

}
```

```
void loop()
{
  Blynk.run();
  delay(1000);
  sendSensor();
}
```

4.2 Output/Results







5. Conclusion and Future Developments

We can see that this reduces human effort and is automated. Still the current position of the project is not efficient. To make it efficient there are many updates. Some are as the temperature can be sensed only at a single place at once. When installed in many places all the sensors can form a network and a single person can know the weather situations at many places accurately.

Another adjustment that can be done is to locate the umbrella using GPS. When a person loses their umbrella they cannot find it, in that situation the GPS can help them. It help-s them to find the location of the umbrella with ease. Similarly, many changes can be brought down to the current application and be published as a daily used product, which can be used by the people for many years to come.

A product should not be so ease that one will become lazy. A product should be automated only at a particular level, or else it can become poison to them. Make sure that everything should be in normal and safe range and does not affect the user physically and mentally. That is called as a perfect automated product.

6. References

[1] Miorandi, D., Sicari, S., De Pellegrini, F., &Chlamtac, I. (2012). Internet of things: Vision,applications and research challenges. Ad Hoc Networks, 10(7), 1497-1516.

[2] Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues. Computer Communications, 54, 1-31.

[3] Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431-440.

[4] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., &Ayyash, M. (2015). Internet ofthings: A survey on enabling technologies, protocols, and applications. IEEE CommunicationsSurveys & Tutorials, 17(4), 2347-2376.

[5] Khajenasiri, I., Estebsari, A., Verhelst, M., &Gielen, G. (2017). A Review on Internet of Things Solutions for Intelligent Energy Control in Buildings for Smart City Applications. Energy Procedia, 111, 770-779.

[6] Jin, D., Hannon, C., Li, Z., Cortes, P., Ramaraju, S., Burgess, P., ...&Shahidehpour, M. (2016). Smart street lighting system: A platform for innovative smart city applications and a new frontier for cyber-security. The Electricity Journal, 29(10), 28-35.

[7] Gaur, A., Scotney, B., Parr, G., & McClean, S. (2015). Smart city architecture and its applications based on IoT. Procedia Computer Science, 52, 1089-1094.

[8] Su, K., Li, J., & Fu, H. (2011, September). Smart city and the applications. In Electronics, Communications and Control (ICECC), 2011 International Conference on (pp. 1028-1031). IEEE.

[9] Hui, T. K., Sherratt, R. S., & Sánchez, D. D. (2016). Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies. Future Generation Computer Systems.

[10] Patchava, V., Kandala, H. B., &Babu, P. R. (2015, December). A Smart Home Automation technique with Raspberry Pi using IoT. In Smart Sensors and Systems (IC-SSS), International Conference on (pp. 1-4). IEEE.

OTHER INFORMATION:

All the data that is collected from the sensor is given to the blynkcloud and the analysis is done in the cloud based on the algorithm used. The output of the present temperature and humidity can be viewed as graph and the future predictions can be viewed in the graph. All the data that is collected from the sensor can be converted in the CSV file such that we can make any other predictions also by using some algorithms to predict the temperature.