D

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

IBM NAALAIYA THIRAN
(TEAM ID - PNT2022TMID37463)

A PROJECT REPORT

Submitted by

SALMAN.S (311819106303)

JAHANGEER NADIR KHAN.P (311819106009)

MOHSIN KHAN.M (311819106014)

NAVINASH.S (311819106301)

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

MOHAMED SATHAK A J COLLEGE OF ENGINEERING

SIRUSERI, OMR, CHENNAI-603 103

NOVEMBER- 2022

D

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY"** is the bonafide work of **SALMAN S(311819106303), JAHANGEER NADIR KHAN P (311819106009), MOHSIN KHAN M (311819106014), NAVINASH S (311819106301)** who carried out the **IBM NAALAIYA THIRAN** project work under our supervision.

Industry Mentor	Faculty Mentor	Faculty Evaluator
(Mr. Santoshi)	(Mrs.Jayanthi.E/AP)	(Mr.M.Kamarajan/Asso Proff)

Head of the Department/ECE

(Dr.M.Sivakumar)

D

ACKNOWLEDGEMENT

First and foremost we thank the almighty for helping us in all situations for bringing out this project successfully.

We express our sincere heartfelt gratitude to **ALHAJ JANAB S.M.MOHAMMED YUSUF**, **CHAIRMAN, Mrs. S.M.H. SHARMILA**, Secretary, **JANAB P.R.L HAMID IBRAHIM**, Executive director and **Mr**. **MOHAMED SATHAK,** Director**,** Mohamed Sathak A J College of Engineering, Chennai.

We would like to express our thanks to our principal **Dr.K.S.SRINIVASAN, M.E, Ph D,**, For his kind consent, inspiration and constant encouragement towards this project work.

We profusely thank **Dr.M.SIVAKUMAR, PhD**. Head of the department, Electronics and Communication Engineering for the help and support, without which our project would have been sculpted successfully.

We express our heart full thanks to our Industry Mentor **Mr. Santoshi, IBM** Faculty Mentor **Mrs.E.JAYANTHI,Assistant Professor/Sr.G,** Faculty Evaluator **Mr.M.KAMARAJAN, Associate Professor** and **Mr.RAJASEKAR, AP, SPOC** For invaluable support, guidance, utmost patience, inspirational co-ordination and constant encouragements in completing this project successfully.

Also we would like to thank all the faculty members and non teaching staff of Electronics and Communication Engineering department for the kind advice and encouragement.

D

TABLE OF CONTENT

D

D

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The goal of this project is to replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones.

1.2 Purpose

- To replace the static signboards, smart connected sign boards are used.
- These smart connected sign boards get the speed limitations from a web appusing weather API and update automatically.
- Based on the weather changes the speed may increase or decrease.
- Traffic diversion signs are remotely controlled using APIs.
- **"DO NOT HONK"** message displayed at School and Hospital Zones which can weset using buttons.

D

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

- Rain makes brakes inefficient and leads to accidents
- Fog reduces visibility and increases the probability of accidents
- Traffic diversion requires human intervention

2.2 References

- Andrzej Czyżewski in his paper titled **"Development of Intelligent Road Signs withV2X Interface for Adaptive Traffic Controlling"**, IEEE 2019, developed IOT based intelligent road signs capable of interacting with both the vehicles and other neighbouring sign boards using LORA. These sign boards were capable of communicating with one another and changing the speed limit based on traffic and weather.
- Muhammed O. Sayin, Chung-Wei Lin, Eunsuk Kang, Shinichi Shiraishi & Tamer Basar in their paper titled **"Reliable Smart Road Signs"**, IEEE 2019, proposed a game theoretical adversarial intervention detection mechanism for reliable smartroad signs. A future trend in intelligent transportation systems is “smart road signs” that incorporate smart codes (e.g., visible at infrared) on their surface to provide more detailed information to smart vehicles.
- L.F.P. Oliveira, L.T. Manera, P.D.G. Luz in their paper titled **"Smart Traffic Light Controller System"**, IEEE 2019, developed smart traffic lights capable of traffic accident detection enabling the enhancement of traffic light management systems, blocking and creating alternative routes to not only avoid the traffic jams, but also avoid new accidents.
- Dariusz Grabowski & Andrzej Czyzewski in their paper titled **"System for**

monitoring road slippery based on CCTV cameras and convolutional neural networks", Springer Publications 2020, made use of Convolutional Neural Networks to identify slippery roads using CCTV cameras.

2.3 Problem Statement Definition

To replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zonesand hospital zones.

D

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

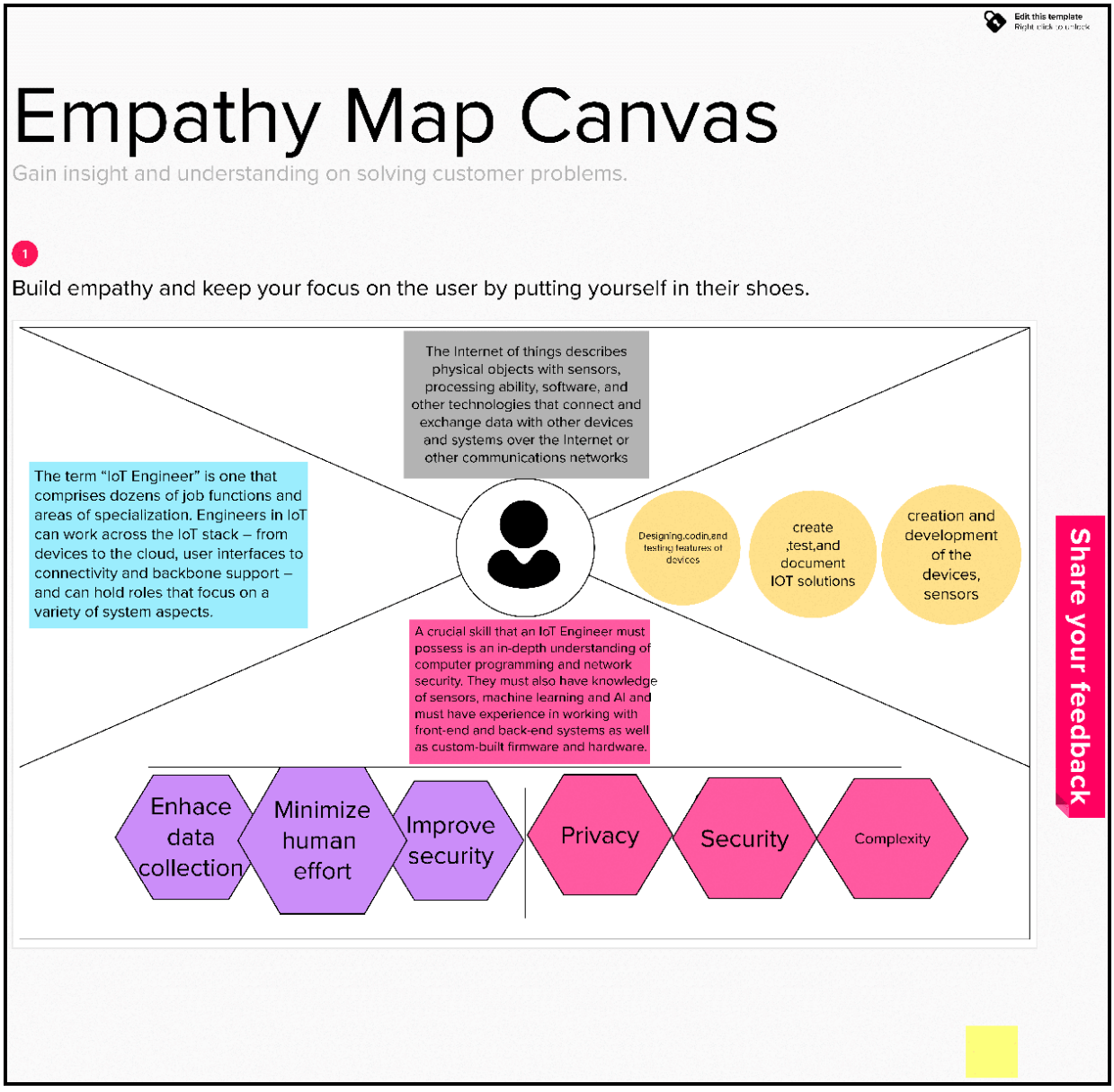


Fig:3.1 Empathy Map

D

3.2 Ideation & Brainstorming

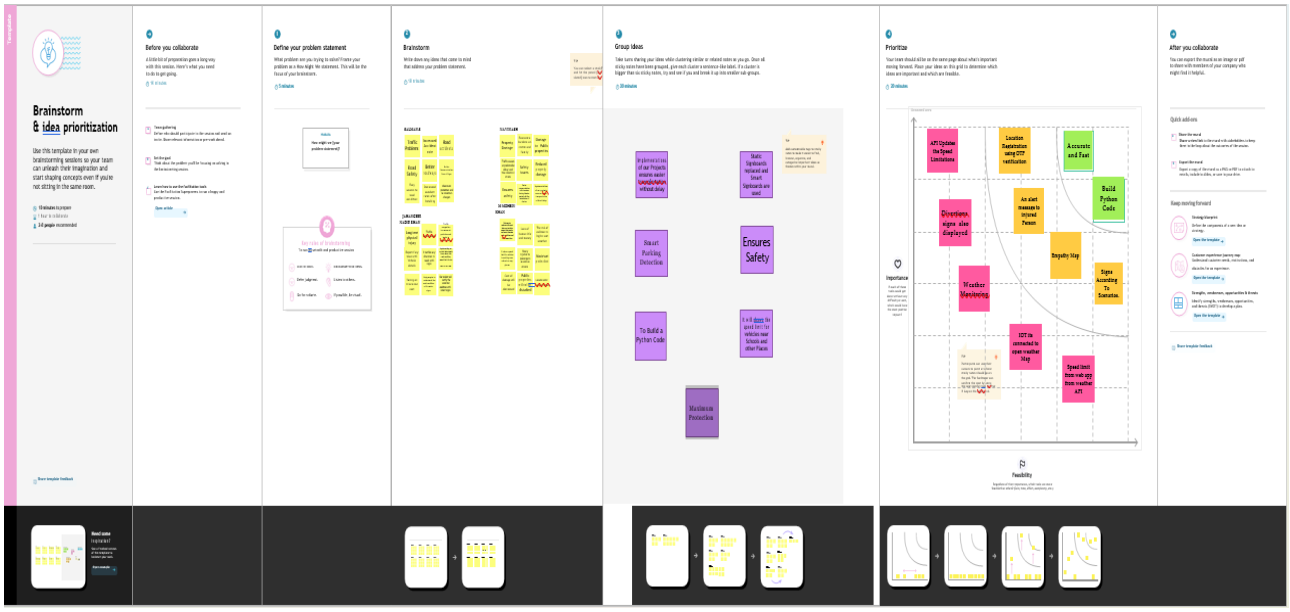


Fig:3.2 Brainstorming

3.3 Proposed Solution

- Use a ESP32 to drive a display as a replacement for static sign boards.
- Configure IBM cloud server such that upon making a single http request with location, unique id, usual speed limit & hospital/school zone info, it returns processes the data at cloud and returns only the message to be displayed at the sign board display.
- Another http end point is configured to set the direction to be displayed. Upon accessing this http end point, the direction is set remotely for a display using it'sunique id.

3.4 Problem Solution Fit

The display replaces the static signs

Processing requirement of microcontroller is reduced since all the processing isdone in the cloud servers.

D

Direction can be remotely set by the concerned authorities without needing topersonally attend the site.

D

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement	Sub Requirement
FR-1	User Visibility	Displays must be made with bright colored LEDs just enough for User attention
FR-2	User Understanding	Display images/text for easier user understanding.
FR-3	User Convenience	Text must be big enough for users to grasp message with ease

4.2 Non-Functional Requirements

NFR No.	Non Functional Requirement	Description
NFR-1	Usability	Should be able to dynamically update with time and weather.
NFR-2	Security	Should be secure enough that only intended messages are displayed.
NFR-3	Reliability	Should convey the traffic sign correctly.
NFR-4	Performance	Display should be updated as soon as traffic signs are updated.
NFR-5	Availability	Should be working 24/7
NFR-6	Scalability	Should be modular and hence be able to scale horizontally on servers.

D

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams

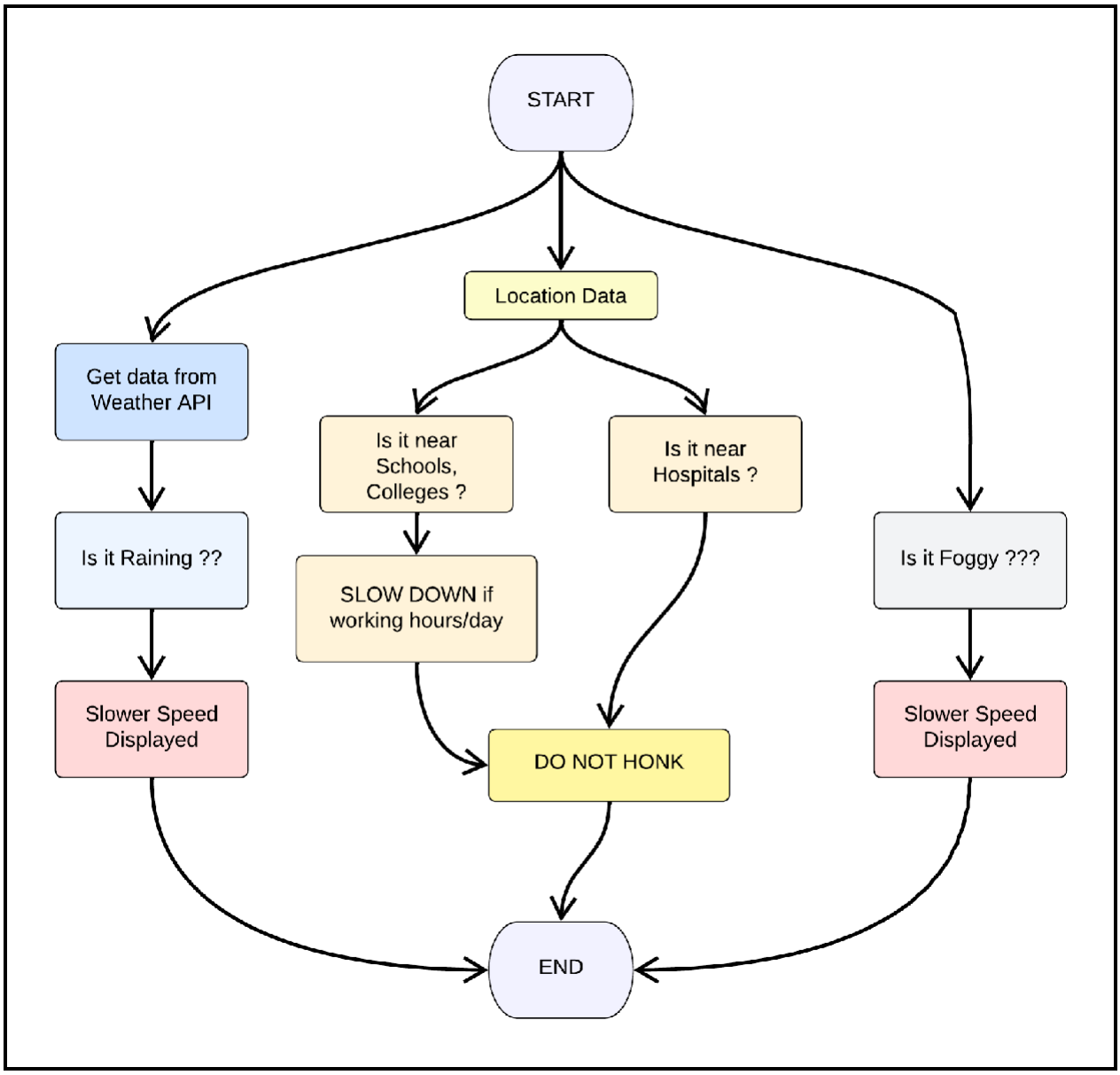


Fig:5.1 Data Flow Diagram

D

5.2 Solution & Technical Architecture

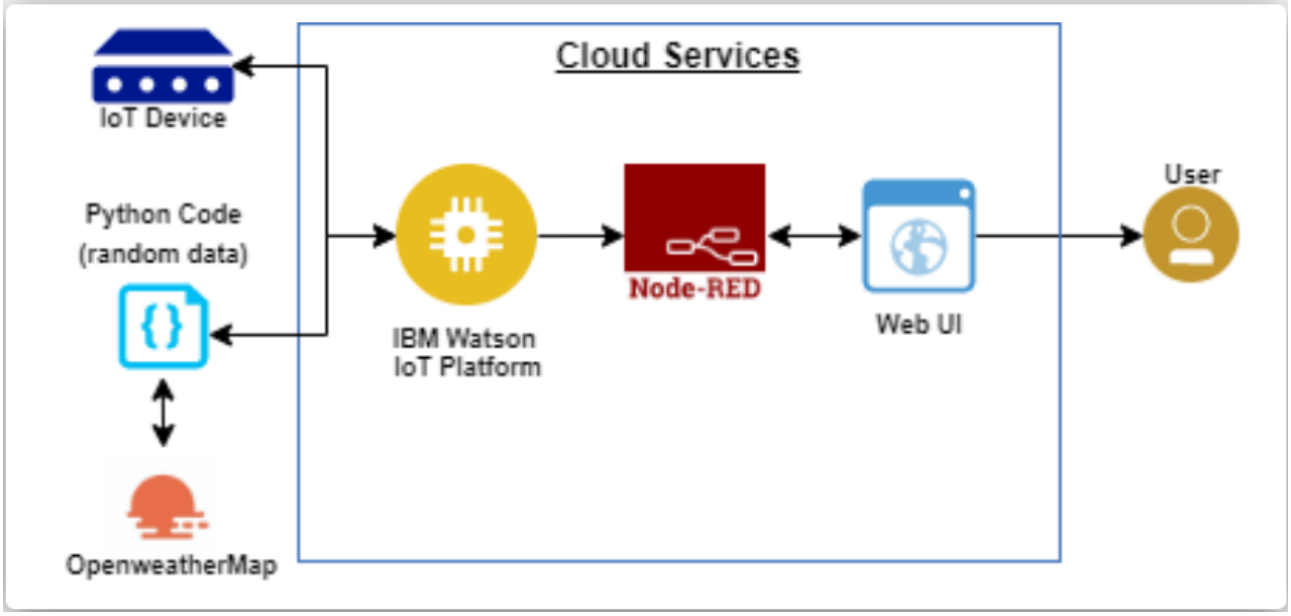


Fig:5.2 Solution & Technical Architecture

5.3 User Stories

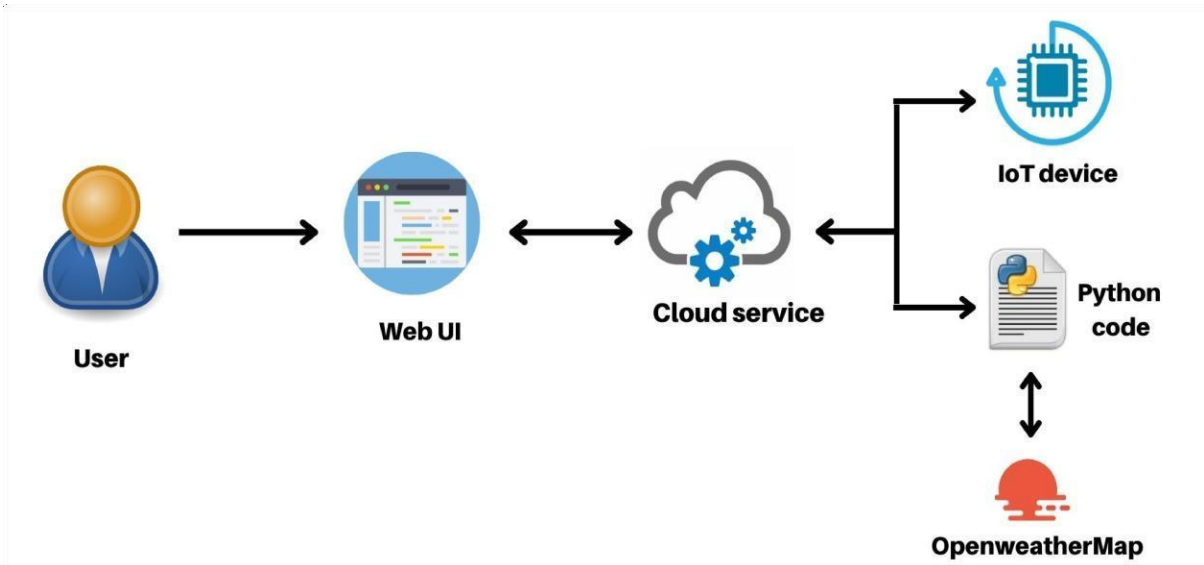


Fig:5.3 User stories

D

CHAPTER 6

PROJECT PLANNING AND SCHEDULING PHASE

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like OpenWeatherMap API.	1	LOW	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location	1	MEDIUM	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-2	Push the server/software to cloud	Push the code from Sprint1 to cloud so it can be accessed from anywhere	2	MEDIUM	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same	2	HIGH	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-4	UI/UX Optimization & Debugging	Optimize all the short comings and provide better user experience	2	LOW	Salman Jahaheer nadir khan Mohsin khan navinash

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	IDE	USN-1	Installing all the softwares which is required like python IDE	2	High	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-1	Checking the simulation with conditions	USN-1	Simulating the circuits and experimenting	2	High	Salman Jahaheer nadir khan Mohsin khan navinash

D

Sprint-2	Software	USN-2	- IBM Watson Iot - NodeRed integration	2	High	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-2	Software	USN-2	Test the device and workflow.	2	High	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-3	Application Development	USN-3	Using MIT App Inventor create an App	2	High	Salman Jahaheer nadir khan Mohsin khan navinash
Sprint-3	Testing	USN-3	Testing the Application.	2	High	Salman Jahaheer nadir khan Mohsin khan navinash

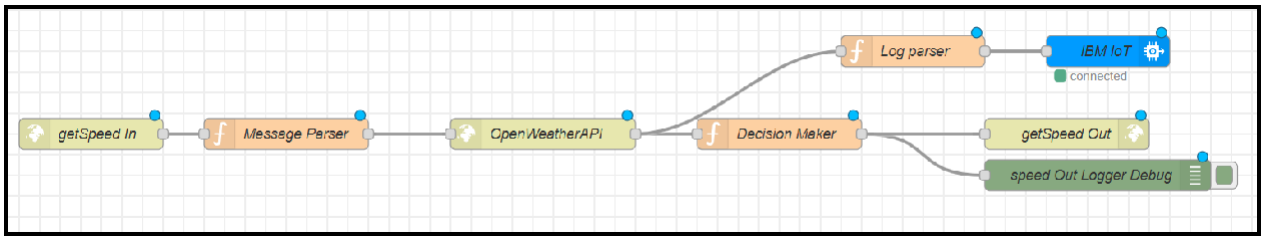
Sprint-4	WEB UI	USN-4	User interface with the Software	2	High	Salman Jahaheer nadir khan Mohsin khan navinash

D

CHAPTER 7

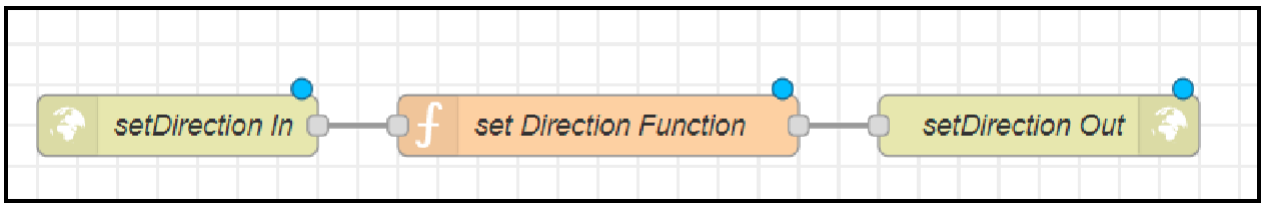
CODING & SOLUTIONING

7.1 FEATURE 1 - Get Speed For Given Location & Climate



This part of Node RED flow accepts an http GET end point at **"/getSpeed"** from which the location, uid, hospital/school zone info are passed. Message parser sets the required APIKEY for Open Weather API for the next block. This data is then passed onto Decision Maker which makes all the decisions regarding the message to be output at the display and sends it as a http response. This data is displayed at the micro- controller. Thus a lot of battery is saved due to lesser processing time.

7.2 FEATURE 2 - Set Direction Remotely For A Given Sign Boar



This part of Node RED flow accepts an http GET end point at **"/setDirection"** fromwhich the uid and direction information are passed by the respective authorities. Set Direction Function block adds the direction information to the database and returns thesame as an http response. This data is sent to the microcontroller along

with the **"/getSpeed"** path and the microcontroller displays it.

A detailed documentation of all the workflows is available at the following

CHAPTER 8

TESTING

8.1 TEST CASES

- **TEST CASE 1**

  Clear weather - Usual Speed Limit.

  Check the weather condition using DHT11 sensor

  Read the data using processor and deployed in to cloud

  Display the result

- **TEST CASE 2**

  Foggy Weather - Reduced Speed Limit.

  Check the Foggy condition using sensor

  Read the data using processor and deployed in to cloud

  Display the result and direction is given for user

- **TEST CASE 3**

  Rainy Weather - Further Reduced Speed Limit.

  Check the rainy condition using sensor

  Read the data using processor and deployed in to cloud

  Display the result and direction is given for user

- **TEST CASE 4**

School/Hospital Zone - Do not Honk sign is displayed.

Based on data received direct the user for easy travel

8.2 User Acceptance Testing

Dynamic speed & diversion variations based on the weather and traffic helps userto avoid traffic and have a safe journey home. The users would welcome this idea to beimplemented everywhere.

- Business Requirements must be available
- Need to to fully develop Application Code
- Ensure you complete Unit Testing, Integration Testing & System Testing
- No Show stoppers, or High or Medium defects in the System Integration Test Phase
- Only Cosmetic errors are acceptable before UAT
- Complete Regression Testing with no major defects
- Ensure you fix and test all the defects reports
- Complete traceability matrix for all testing
- UAT Environment must be ready
- Sign off mail or communication from System Testing Team that the system is ready for UAT execution

D

As we were saying, User Acceptance Testing, also known as UAT (or UAT testing), in a nutshell, is:

A process of verifying that a solution works for the user.

D

CHAPTER 9

RESULTS

9.1 Performance Metrics

Based on the IBM pack we chose, the performance of the website varies. Built upon NodeJS, a light and high performance engine, NodeRED is capable of handling upto 10,000 requests per second. Moreover, since the system is horizontally scalable, aeven higher demand of customers can be served.

			NFT - Risk Assessment					
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Load/Volume Changes	Risk Score	Justification
1	Led ON/OFF	Existing	Low	No Changes	Moderate	>10 to 30%	ORANGE	There are some changes that occur
2	Fast2SMS	New	No changes	No Changes	Low	>5 to 10%	GREEN	Hardly any changes occur
3	Sensor values	Existing	Moderate	No Changes	Moderate	>10 to 30%	ORANGE	There are some changes that occur

	NFT - Detailed Test Plan				
	S.No	Project Overview	NFT Test approach	Approvals/SignOff	Assumptions/Dependencies/Risk
	1	Python script	Python coding		Depends on the delivered code
	2	Node Red	Sensor & command values		Sensor values
	3	MIT App Inventor	Led control\Sensor control notifications		Notifications on Phone

			End Of Test Report					
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Identified Defects (Detected/Closed/Open)	Recommendations	Approvals/SignOff
1	Python Code	Python coding	Met	Pass	GO	Closed	Efficient and easy code	
2	Node Red	Sensors&command values	Met	Pass	GO	Closed	It senses the values perfectly	
3	MIT App Inventor	Led control/Sensors notification	Met	Pass	GO	Closed	Notifies the users at the right time	

D

CHAPTER 10

ADVANTAGES & DISADVANTAGES

- **ADVANTAGES**
  - Lower battery consumption since processing is done mostly by Node REDservers in the cloud.
  - Cheaper and low requirement micro controllers can be used since processing requirements are reduced.
  - Longer lasting systems.
  - Dynamic Sign updation.
  - School/Hospital Zone alerts

- **DISADVANTAGES**
  - The size of the display determines the requirement of the micro controller
  - Dependent on OpenWeatherAPI and hence the speed reduction is samefor a large area in the scale of cities.

D

CHAPTER 11

CONCLUSION

Our project is capable of serving as a replacement for static signs for a comparatively lower cost and can be implemented in the very near future. This will help to reduce a lot of accidents and maintain a more peaceful traffic atmosphere in the country.

D

CHAPTER 12

FUTURE SCOPE

Introduction of intelligent road sign groups in real life scenarios could have greatimpact on increasing the driving safety by providing the end-user (car driver) with the most accurate information regarding the current road and traffic conditions. Even displaying the information of a suggested driving speed and road surface condition (temperature, icy, wet or dry surface) could result in smoother traffic flows and, what is more important, in increasing a driver's awareness of the road situation. In future this project can be implemented using AI

D

CHAPTER 13

Appendix

Github And Project Demo Link: https://drive.google.com/file/d/1nQ4Bs3THdvdZPk63sklj-jWKR0ineYKR/view?usp=sharing

Demo Video Download Link: https://youtu.be/uzrf8jAeHzA

- **SOURCE CODE - ESP 32**

D

- **SOURCE CODE - ESP 32**

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <string.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";

#define TFT_DC 2
#define TFT_CS 15
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

String myLocation = "Chennai,IN";
```

D

```
15String usualSpeedLimit = "70"; // kmph16
17 int schoolZone = 32;
18 int hospitalZone = 26;
19
20int uid = 2504; // ID Unique to this Micro Contoller21
22String getString(char x)23{
24        String s(1, x);
25        return s;
26}
27
28String stringSplitter1(String fullString,char delimiter='$')29{
30        String returnString = "";
31        for(int i = 0; i<fullString.length();i++) {
32                char c = fullString[i];
33                if(delimiter==c)
34                        break;
35        returnString+=String(c);36}
37        return(returnString);38}
39
40String stringSplitter2(String fullString,char delimiter='$')41{
42        String returnString = "";
43        bool flag = false;
```

D

```
44        for(int i = 0; i<fullString.length();i++) {
45              char c = fullString[i];
46              if(flag)
47                    returnString+=String(c);
48              if(delimiter==c)
49        flag = true;50    }
51        return(returnString);52}
53
54void rightArrow()
55{
56    int refX = 50;
57    int refY = tft.getCursorY() + 40;58
59    tft.fillRect(refX,refY,100,20,ILI9341_RED);
60                                          tft.fillTriangle(refX+100,refY-
    30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
61}
62
63void leftArrow()
64{
65    int refX = 50;
66    int refY = tft.getCursorY() + 40;67
68    tft.fillRect(refX+40,refY,100,20,ILI9341_RED);
69                                          tft.fillTriangle(refX+40,refY-
    30,refX+40,refY+50,refX,refY+10,ILI9341_RED);
70}
```

D

```
71
72void upArrow()
73{
74    int refX = 125;
75    int refY = tft.getCursorY() + 30;76
77                                                    tft.fillTriangle(refX-
    40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED);
78    tft.fillRect(refX-15,refY+40,30,20,ILI9341_RED);79}
80
81 String APICall() {
82    HTTPClient http;
83
84      String  url  =  "https://node-red-grseb-2022-11-05-test.eu-
    gb.mybluemix.net/getSpeed?";
85    url += "location="+myLocation+"&";
86                                                    url                    +=
    "schoolZone="+(String)digitalRead(schoolZone)+(String)"&";
87                                                    url                    +=
    "hospitalZone="+(String)digitalRead(hospitalZone)+(String)"&";
88                                                    url                    +=
    "usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";
89    url += "uid="+(String)uid;
90    http.begin(url.c_str());
91    int httpResponseCode = http.GET();92
93    if (httpResponseCode>0) {
```

D

```
94        String payload = http.getString();
95        http.end();
96        return(payload);97
}
98    else {
99        Serial.print("Error code: ");
100           Serial.println(httpResponseCode);
101       }
102       http.end();
103   }
104
105    void myPrint(String contents) {
106       tft.fillScreen(ILI9341_BLACK);
107       tft.setCursor(0, 20);
108       tft.setTextSize(4);
109       tft.setTextColor(ILI9341_RED);
110       //tft.println(contents);111
112       tft.println(stringSplitter1(contents));
113       String c2 = stringSplitter2(contents);
114       if(c2=="s") // represents Straight
115       {
116          upArrow();
117       }
118       if(c2=="l") // represents left
119       {
120          leftArrow();
```

D

```
if      "r"  // represents right
```

D

```
123       {
124           rightArrow();
125       }
126   }
127
128    void setup() {
129       WiFi.begin(ssid, password, 6);130
131       tft.begin();
132       tft.setRotation(1);133
134       tft.setTextColor(ILI9341_WHITE);
135       tft.setTextSize(2);
136       tft.print("Connecting to WiFi");137
138       while (WiFi.status() != WL_CONNECTED) {
139           delay(100);
140           tft.print(".");
141       }
142
143       tft.print("\nOK! IP=");
144       tft.println(WiFi.localIP());
145   }
146
147    void loop() {
148       myPrint(APICall());
149       delay(100);
150   }
```