IBM NALAIYA THIRAN 2022-23 PROJECT REPORT

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD

SAFETY TEAM ID - PNT2022TMID29476

1. INTRODUCTION

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.

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.

2. LITERATURE SURVEY

EXISTING PROBLEM

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

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 smart road 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.

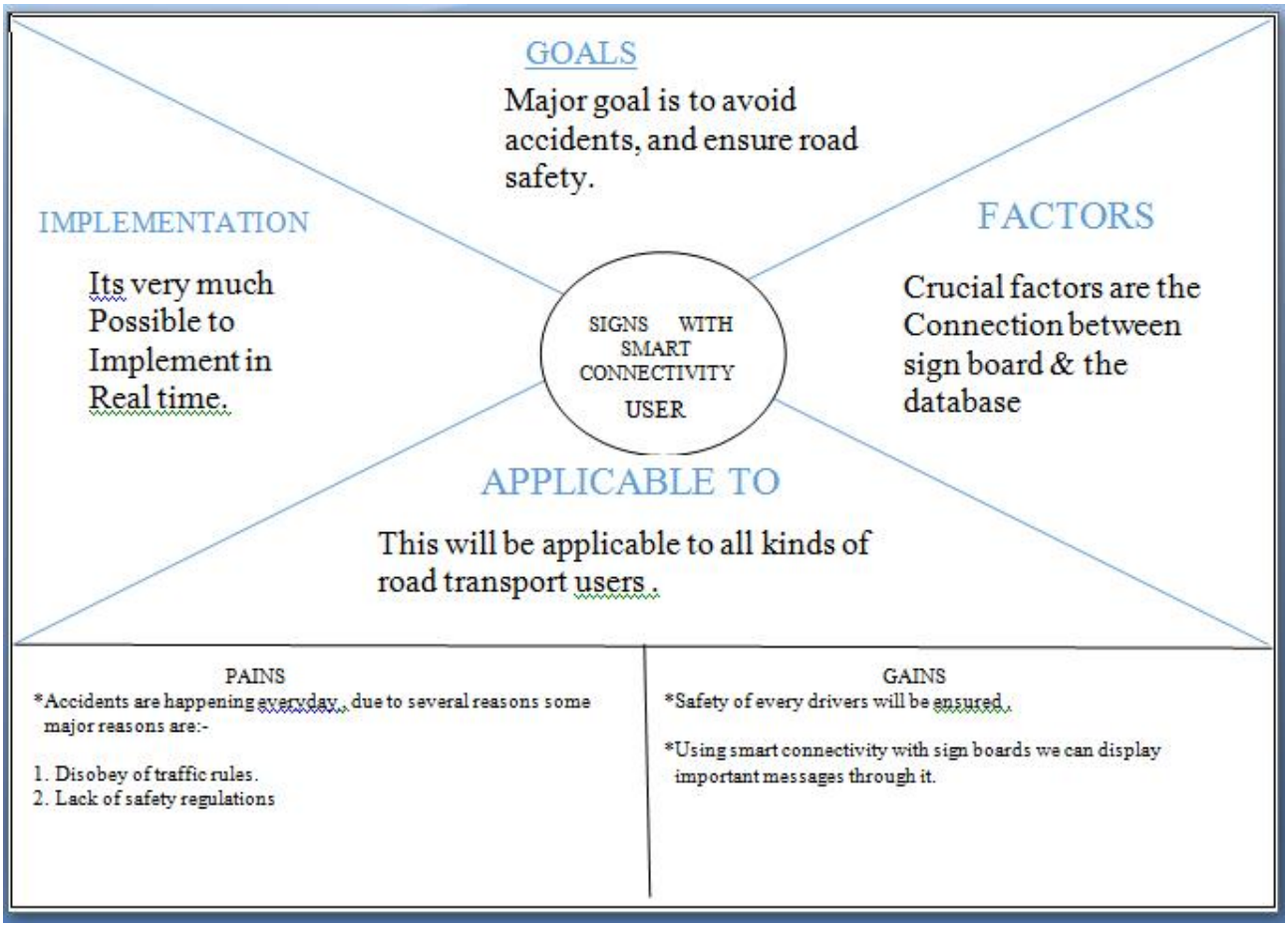
PROBLEM STATEMENT DEFINITION

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.

3. IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/tree/main/IDEATION%20PHASE/EMPATHY%20MAP/VADIVEL%20S



SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD

IDEATION & BRAINSTORMING

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/tree/main/IDEATION%20PHASE/IDEATION%20(ideas)/SURYA%20N

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 thesign 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.

PROBLEM SOLUTION FIT

- The display replaces the static signs
- Processing requirement of microcontroller is reduced since all the processing isdone in the cloud servers.
- Direction can be remotely set by the concerned authorities without needing to personally attend the site.

4. REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/blob/main/Project%20Design%20%20Phase%202/N%20SATHYAMOORTHY/Functional_requirement.pdf

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

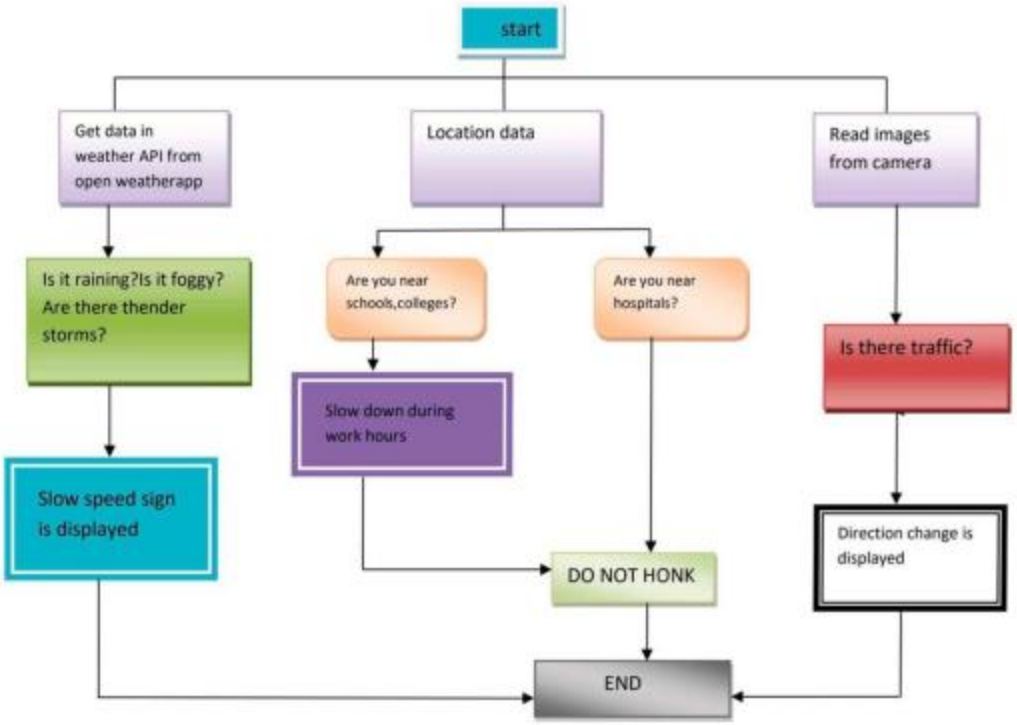
NON-FUNCTIONAL REQUIREMENTS

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/blob/main/Project%20Design%20%20Phase%202/N%20SATHYAMOORTHY/Functional_requirement.pdf

NFR No.	Non Functional Requirement	Desciption
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.

5. PROJECT DESIGN

DATA FLOW DIAGRAMS



SOLUTION & TECHNICAL ARCHITECTURE

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/blob/main/Project%20Design%20%20Phase%202/E%20VETRI%20SELVAN/Technology_Architecture.pdf

USER STORIES

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/blob/main/Project%20Design%20%20Phase%202/N%20SURYA/Customer_Journey.pdf

6. PROJECT PLANNING AND SCHEDULING PHASE

6.1 SPRINT PLANNING & ESTIMATION

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/tree/main/Project%20Development%20Phase

6.2 SPRINT DELIVERY SCHEDULE

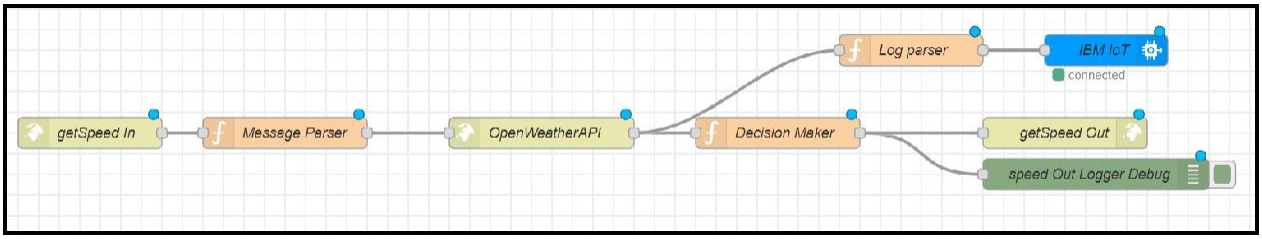
https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/tree/main/Project%20Development%20Phase

6.3 REPORTS FROM JIRA

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/tree/main/JIRA%20Software

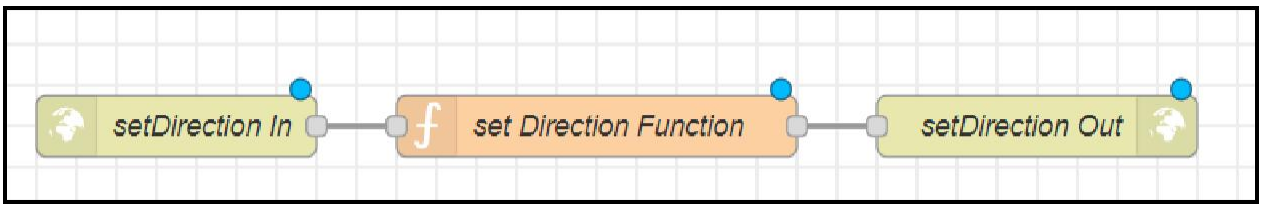
7.CODING & SOLUTIONING

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 OpenWeatherAPI 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.

FEATURE 2 - SET DIRECTION REMOTELY FOR A GIVEN SIGN BOARD



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 the same 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 workflow is available at the following link : https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/blob/main/Project%20Development%20Phase/Sprint%201/Sprint%201.pdf

TESTING

TEST CASES

- **TEST CASE 1**

  Clear weather - Usual Speed Limit.

- **TEST CASE 2**

  Foggy Weather - Reduced Speed Limit.

- **TEST CASE 3**

  Rainy Weather - Further Reduced Speed Limit.

- **TEST CASE 4**

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

USER ACCEPTANCE TESTING

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

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD

7. RESULTS

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.

8. 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 sinceprocessing requirements are reduced.
    - Longer lasting systems.
    - Dynamic Sign upgradation.
    - School/Hospital Zone alerts
- **DISADVANTAGES**
    - The size of the display determines the requirement of the micro controller
    - Dependent on Open Weather API and hence the speed reduction is samefor a large area in the scale of cities.

9. 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 reduce a lot of accidents and maintain a more peaceful traffic atmosphere in the country.

10. FUTURE SCOPE

Introduction of intelligent road sign groups in real life scenarios could have great impact 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.

11. APPENDIX

- **GITHUB AND PROJECT DEMO LINK**
- **https://github.com/IBM-EPBL/IBM-Project-50988-1660962599**
- **DEMO VIDEO DOWNLOAD LINK**

https://github.com/IBM-EPBL/IBM-Project-50988-1660962599/tree/main/Final%20Deliverables/Demo%20video

- **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";
```

```
15 String usualSpeedLimit = "70"; // kmph
16
17 int schoolZone = 32;
18 int hospitalZone = 26;
19
20 int uid = 2504; // ID Unique to this Micro Contoller
21
22 String getString(char x)
23 {
24     String s(1, x);
25     return s;
26 }
27
28 String stringSplitter1(String fullString,char delimiter='$')
29 {
30     String returnString = "";
31     for(int i = 0; i<fullString.length();i++) {
32         char c = fullString[i];
33         if(delimiter==c)
34             break;
35     returnString+=String(c);36  }
37     return(returnString);
38 }
39
40 String stringSplitter2(String fullString,char delimiter='$')
41 {
42     String returnString = "";
43     bool flag = false;
```

```
    for(int i = 0; i<fullString.length();i++) {
        char c = fullString[i];
        if(flag)
            returnString+=String(c);
        if(delimiter==c)
    flag = true;50    }
    return(returnString);
}

void rightArrow()
{
  int refX = 50;
  int refY = tft.getCursorY() + 40;

  tft.fillRect(refX,refY,100,20,ILI9341_RED);
                                tft.fillTriangle(refX+100,refY-
  30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
}

void leftArrow()
{
  int refX = 50;
  int refY = tft.getCursorY() + 40;

  tft.fillRect(refX+40,refY,100,20,ILI9341_RED);
                                 tft.fillTriangle(refX+40,refY-
  30,refX+40,refY+50,refX,refY+10,ILI9341_RED);
}
```

```
71
72 void 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) {
```

```
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();
121     }
122     if(c2=="r") // represents right
```

```
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   }
```