SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

IBM PROJECT REPORT

Team ID - PNT2022TMID17040

SUBMITTED BY

N.Tamilarasi (92172019102151)

S.Varsha (92172019102160)

P.Swathika (92172019102150)

H.Vijaya Preya Darshini (92172019102169)

Final Deliverables Report

Date	19.11.2022
Team ID	PNT2022TMID17040
Project Name	Signs with Smart Connectivity for Better Road Safety

Team members and their Contributions:

Name	Roll no	Contribution
N.Tamilarasi	92172019102151	CREATED SOURCE CODE FOR THE WOKWI SIMULATOR AND MIT APP CODE.
S.Varsha	92172019102160	CREATED NODE RED AND IOT WATSON PLATFORM.
P.Swathika	92172019102150	PROJECT REPORT MAKING PROCESS AND GATHERING IDEAS FOR CREATING PROJECT.
H. Vijaya Preya Darshini	92172019102169	WORKINGS IN NODE RED FLOW AND IBM CLOUD DEPLOYMENT.

Introduction:

1. Sprint 1 – Create and initialize accounts in various public APIs like OpenWeatherMap API, and write a Python program that outputs results given the inputs like weather and location.
2. Sprint 2 – Push data from local code to cloud
3. Sprint 3 – Hardware & Cloud integration
4. Sprint 4 – UI/UX Optimization & Debugging

CONTENTS

1. INTRODUCTION

1.1 Project Overview

- To replace the static signboards, smart connected signboards are used.
- These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.
- Based on the weather changes the speed may increase or decrease.
- Based on the traffic and fatal situations the diversion signs are displayed.
- Guide (Schools), Warning and Service (Hospitals, Restaurants) signs are also displayed accordingly.
- Different modes of operations can be selected with the help of buttons.

1.2 Purpose

- Smart Traffic Management is a system to monitor and control traffic signals using sensors to regulate the flow of traffic and to avoid congestion for a smooth flow of traffic.

- Prioritizing traffic like ambulances, police etc. is also one application comes under smart traffic management.

2. LITERATURE SURVEY

2.1 Existing problem

- Analysis of crash data has suggested a link between roadside advertising signs and safety.
- Research suggests that crash risk increases by approximately 25–29% in the presence of digital roadside advertising signs compared to control areas.
- On the other hand, static roadside advertising signs have not been linked with differences in the crash count.
- However, this finding is contrary to previous research that suggests differences in crash counts exist in the presence of static roadside advertising.
- The quantity and quality of available evidence limit our conclusion.
- Fixed object, side swipe and rear end crashes are the most common types of crashes in the presence of roadside advertising signs.
- In addition, drivers showed increased eye fixations and increased drifting between lanes on the road.

2.2 References

- Cairney and Gunatillake, 2000; Sisiopiku et al., 2015
- Islam, 2015; Sisiopiku et al., 2015
- Yannis et al., 2013, Staffeld (1953) and Ady (1967)

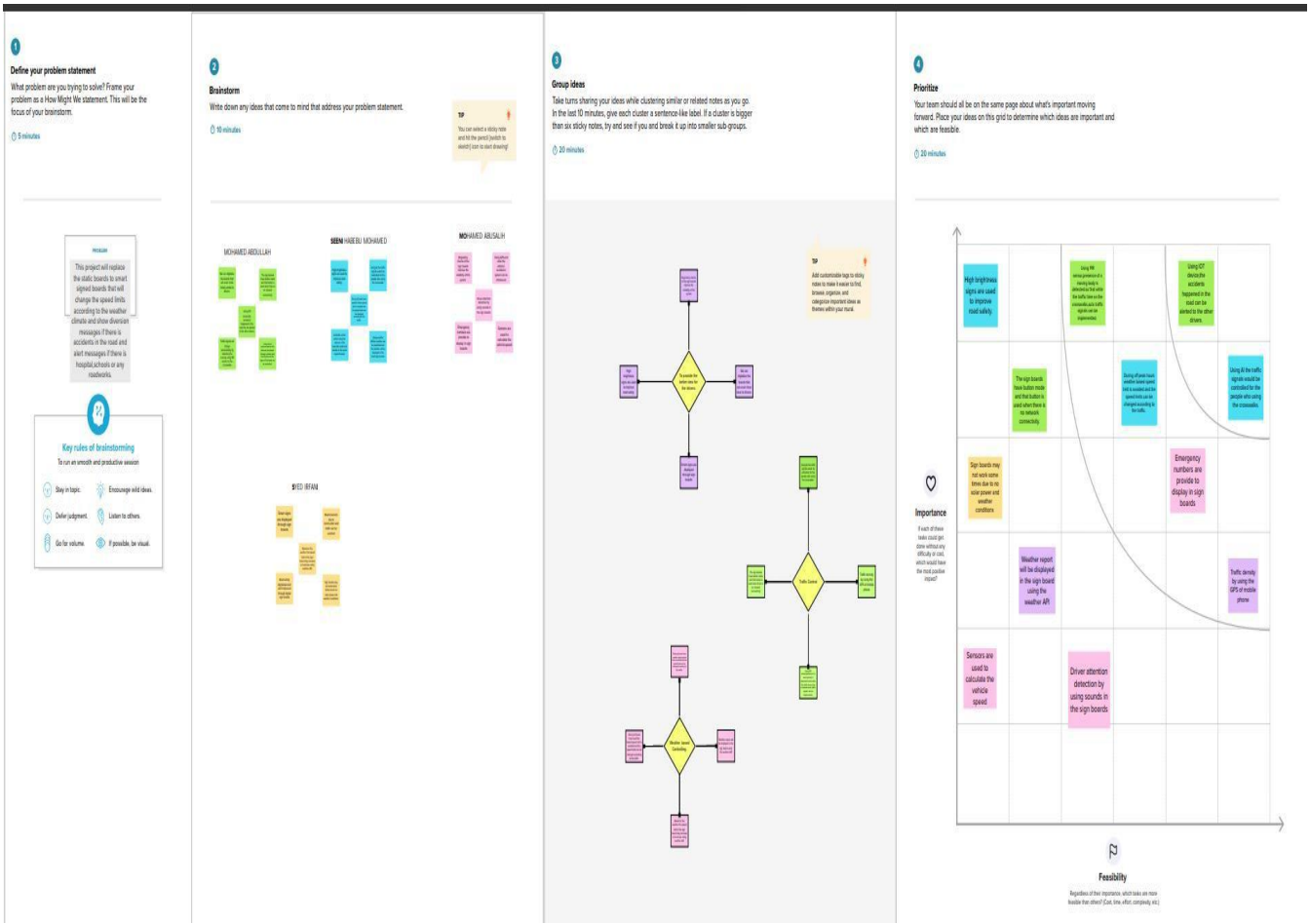
2.3 Problem Statement Definition

This project will replace the static boards to smart signed boards that will change the speed limits according to the weather climate and show diversion messages if there are accidents in the road and alert messages if there is hospital, schools or any roadworks.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming Map



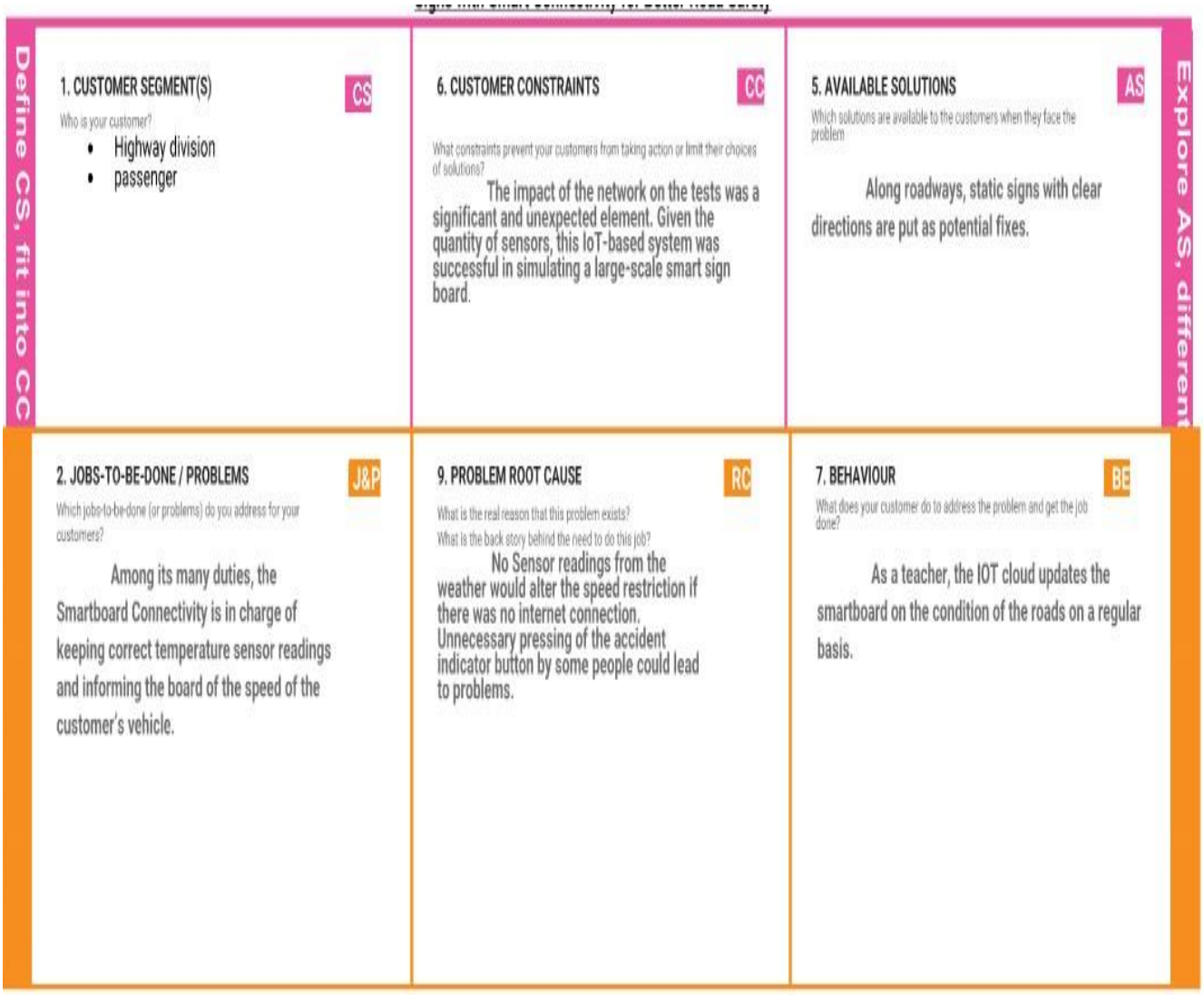
3.3 Proposed Solution

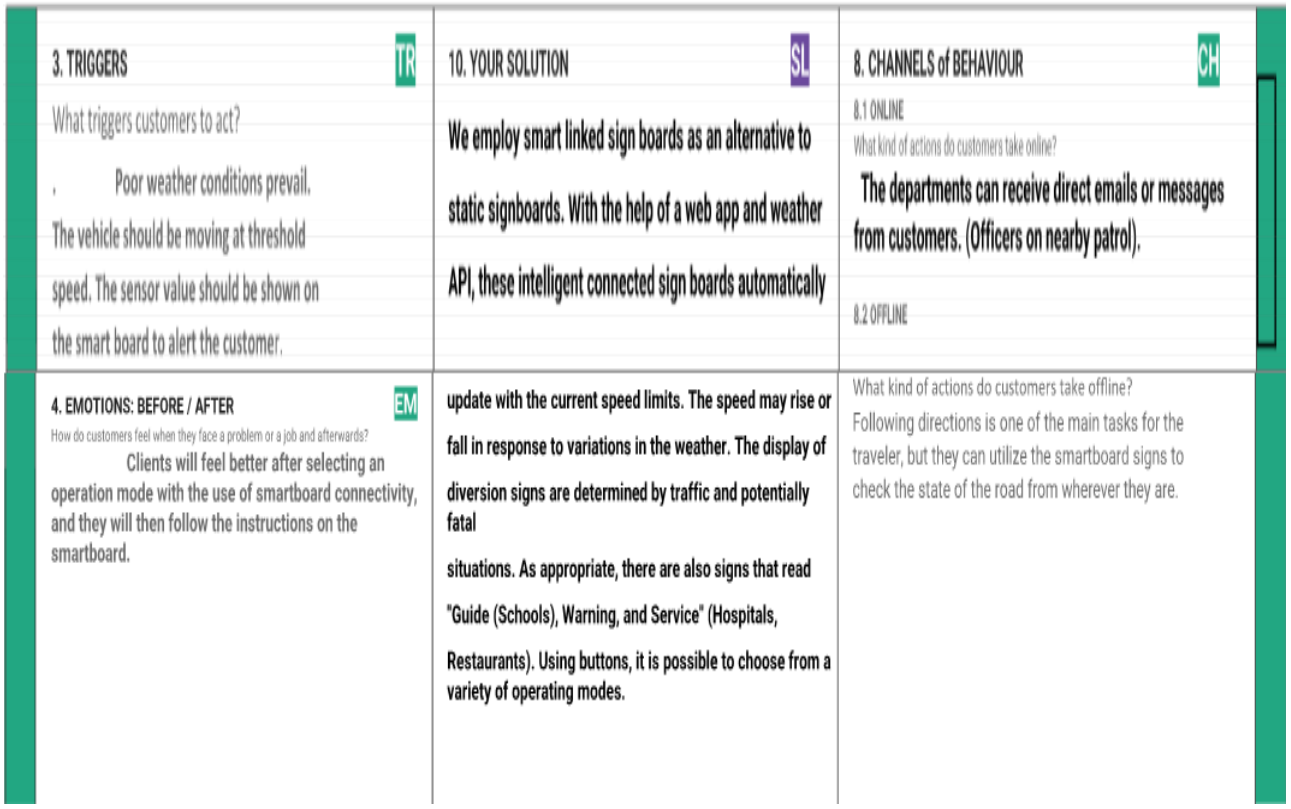
S.No.	Parameter	Description
1.	Problem Statement	To replace the static signboards, smart connected sign boards are used. These smart connected sign boards get the speed limitations from a web app using weather API and update automatically. Based on the weather changes the speed may increase or decrease Based on the traffic and fatal situations the diversion signs are displayed. Guide (Schools), Warning and Service (Hospitals, Restaurant) signs arealso displayed accordingly. Different modes of operations can be selected with the help of buttons.
2.	Idea description	The weather and temperature details are obtained from the OpenWeatherMap API. Using these details, the speed limit will be updated automatically in accordance with the weather conditions. Also, the details regarding any accidents and traffic congestion faced on the particular road are obtained. Based on this, the traffic is diverted followed by a change in map path and the traffic is cleared. So, in the traffic sign board, some buttons will be placed which willbe used to make it generic; where each button will be given a functionality such as changing the warning signs, which are predefined and separate signs will be present for both school and

S.No.	Parameter	Description
		hospital zones. By activating this button, either through the web application or the physical buttons, sign of the board can be changed accordingly, and the speed limit will also be set depending upon the zones. Also, the pedestrians are given an option to change the traffic signs if they want to cross the road. If the pedestrian presses the button that is present on the post at the end of the road, then the trafficwill be analyzed immediately. Accordingly, the sign of the traffic signal will be changed. This in turn reduces the frequent changing of the traffic signs even if the pedestrians are not present.
3.	Novelty	Generic Sign board for all applications that uses both buttons and web service for updation. Pedestrians are given the access to request the sign change of the signal to cross the road
4.	Customer Satisfaction	Diversion reasons will be displayed If there is no traffic, pedestrians can cross the street without waiting. Customer can reach the destination before the expected time

5.	Business Model	Since APIs are used to actively monitor the customer's environment, this project employs a business strategy in which revenue will be generated on the basis of the length of time in which the customers actively interact with the product. This product is aimed to be free of cost to the public, but the revenue will be generated by selling this product to the government at a low cost, so there will be less accidents and the public will be aware of the discrepancies or accidents in the particular road. The public will also gain all the information about the road, even if they are checking for an alternate path because of some mishaps that happen on the roads and these functionalities will increase the value of the product in the global market.
6.	Scalability of the Solution	In the future, if any update is required either on the hardware or software side, it can be easily implemented. The hardware components can be directly interfaced with the microcontroller and small modifications can be made in the programming of the existing product. In case of the software, the website application has to be updated with the additional functionality by creating a new section for the updated hardware. So, this will not affect the existing functionality of the product and new functionality can be easily integrated. In addition, a separate circuit will be kept along with the hardware to detect any problem which informs the web application. Also, a notification will be sent to the product service department.

3.4 Problem Solution fit




3. TRIGGERS
TR
What triggers customers to act?
, Poor weather conditions prevail.
The vehicle should be moving at threshold speed. The sensor value should be shown on the smart board to alert the customer.
10. YOUR SOLUTION
SL
We employ smart linked sign boards as an alternative to static signboards. With the help of a web app and weather API, these intelligent connected sign boards automatically update with the current speed limits. The speed may rise or fall in response to variations in the weather. The display of diversion signs are determined by traffic and potentially fatal situations. As appropriate, there are also signs that read "Guide (Schools), Warning, and Service" (Hospitals, Restaurants). Using buttons, it is possible to choose from a variety of operating modes.
8. CHANNELS of BEHAVIOUR
CH
8.1 ONLINE
What kind of actions do customers take online?
The departments can receive direct emails or messages from customers. (Officers on nearby patrol).
8.2 OFFLINE
What kind of actions do customers take offline?
Following directions is one of the main tasks for the traveler, but they can utilize the smartboard signs to check the state of the road from wherever they are.
4. EMOTIONS: BEFORE / AFTER
EM
How do customers feel when they face a problem or a job and afterwards?
Clients will feel better after selecting an operation mode with the use of smartboard connectivity, and they will then follow the instructions on the smartboard.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-I	User Visibility	Sign Boards should be made of bright colored LEDs capable of attracting driver's attention Not too distracting to cause accidents
FR-2	User Understanding	Should display information through means like images/illustrations with text so that the user can understand the signs correctly
FR-3	User Convenience	Display should be big enough to display all the signscorrectly so that it is visible even to far away drivers

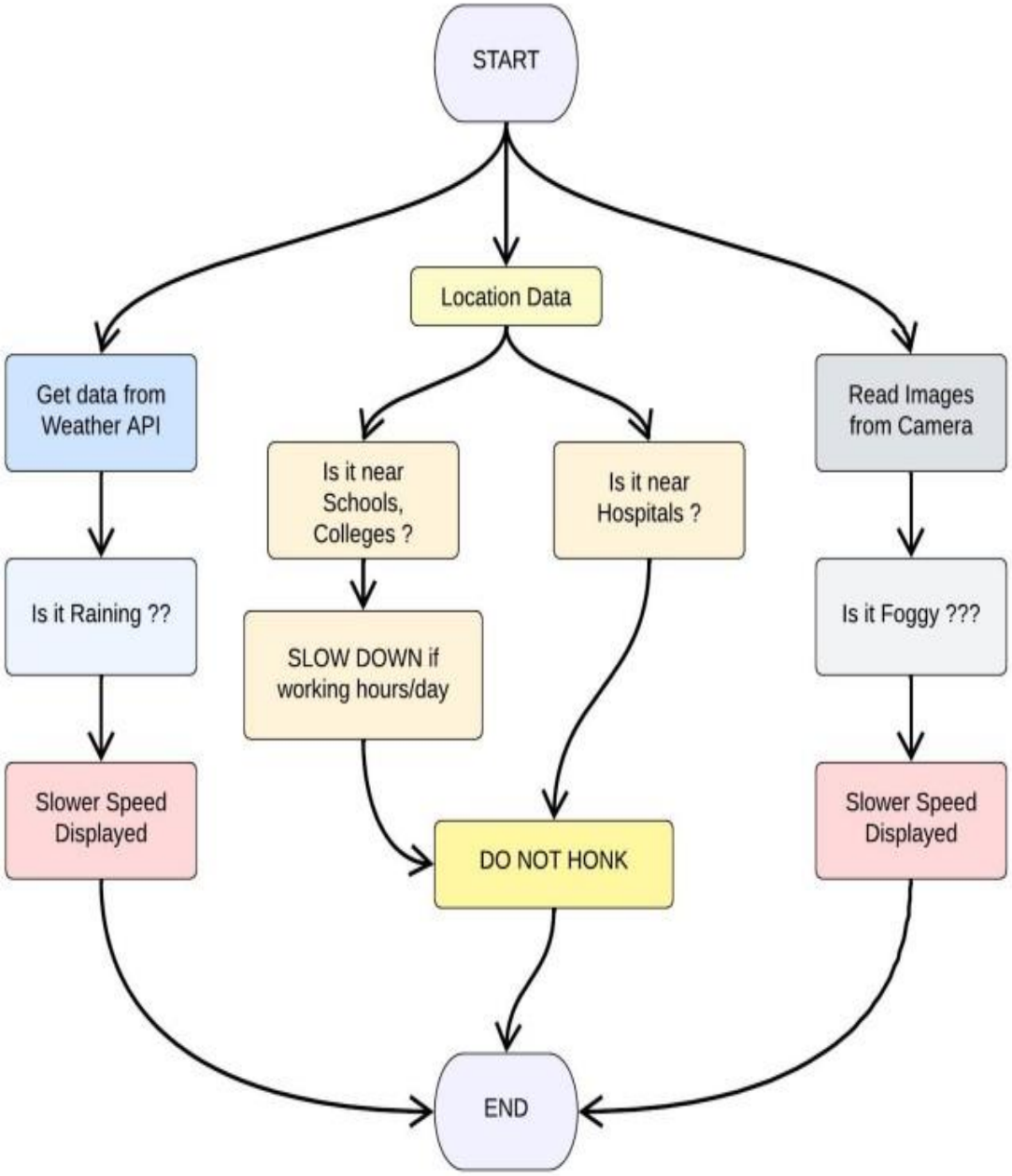
4.2 Non-Functional Requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-I	Usability	Should be able to dynamically update with respectto time.
NFR-2	Security	Should be secure enough that only the intendedmessages are displayed in the display.
NFR-3	Reliability	Should convey the traffic information correctly.
NFR-4	Performance	Display should update dynamically whenever theweather or traffic values are updated
NFR-5	Availability	Should be on service 24/7
NFR-6	Scalability	Should be modular and hence able to scale onservers horizontally.

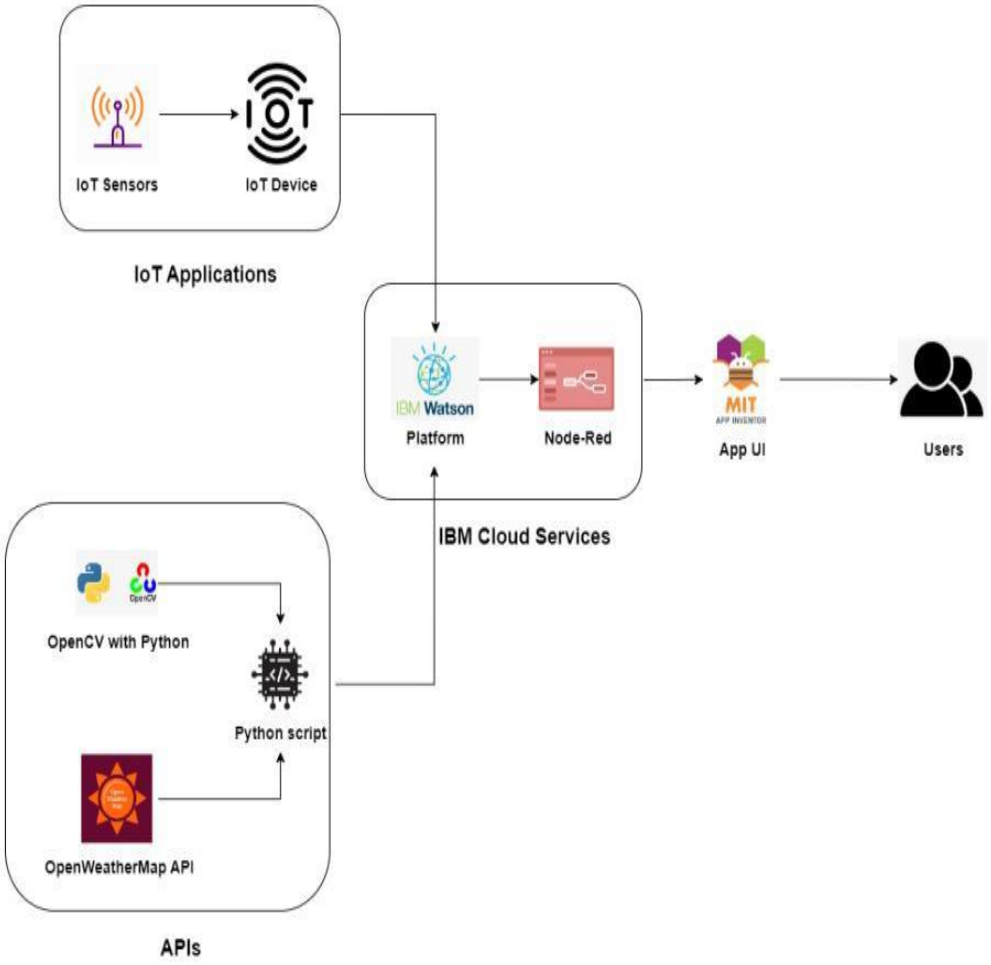
5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution & Technical Architecture

Following is the Technical Architecture with slight change and is **without the implementation of OpenCV API**.



Following is the Solution Built

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User can interact with the app using MIT App	HTML, CSS, JavaScript / Angular Js /React Js
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	IBM Cloud
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other StorageService or Local Filesystem
8.	External API-1	Purpose of External API used in the application	Open Weather Map API
9.	External API-2	Purpose of External API used in the application	IBM Watson Platform, Node - Red
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	*OpenWeatherMap, NODE-RED, IBM WATSON,MIT App Inventor*	IoT, internet
2.	Security Implementations	*Powerful security system for everyone'speace of mind No access data Hackers cannot access network*	Firewall, Firebase, cyber resiliency, strategy
3.	Scalable Architecture	*EASY TO EXTEND THE NETWORK WITH THEAID OF THE BANDWIDTH OF THE NETWORK*	IBM Cloud
4.	Availability	*Available every time and everywhere 24/7so long as the consumer is signed into thenetwork.*	IBM Cloud
5.	Performance	*AIDS MASSIVE RANGE OF USERS TO USE TECHNOLOGY*	IBM Cloud

5.3 User Stories

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

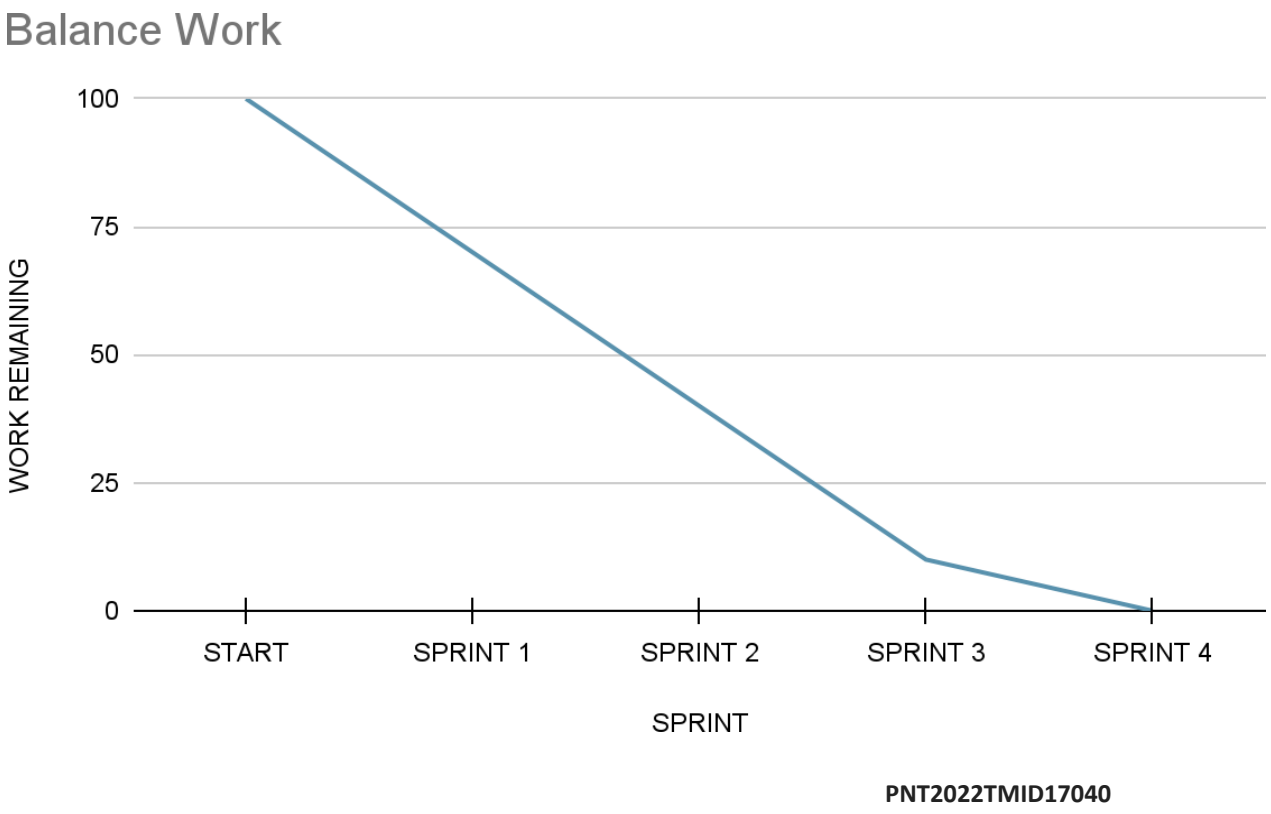
Sprint	Functional Requirement (Epic)	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like OpenWeatherMap API.	1	LOW	Tamilarasi Varsha Swathika vijayapreyadarsini
Sprint-1	Local Server/Software Run	Write a Python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

6.2 Sprint Delivery Schedule

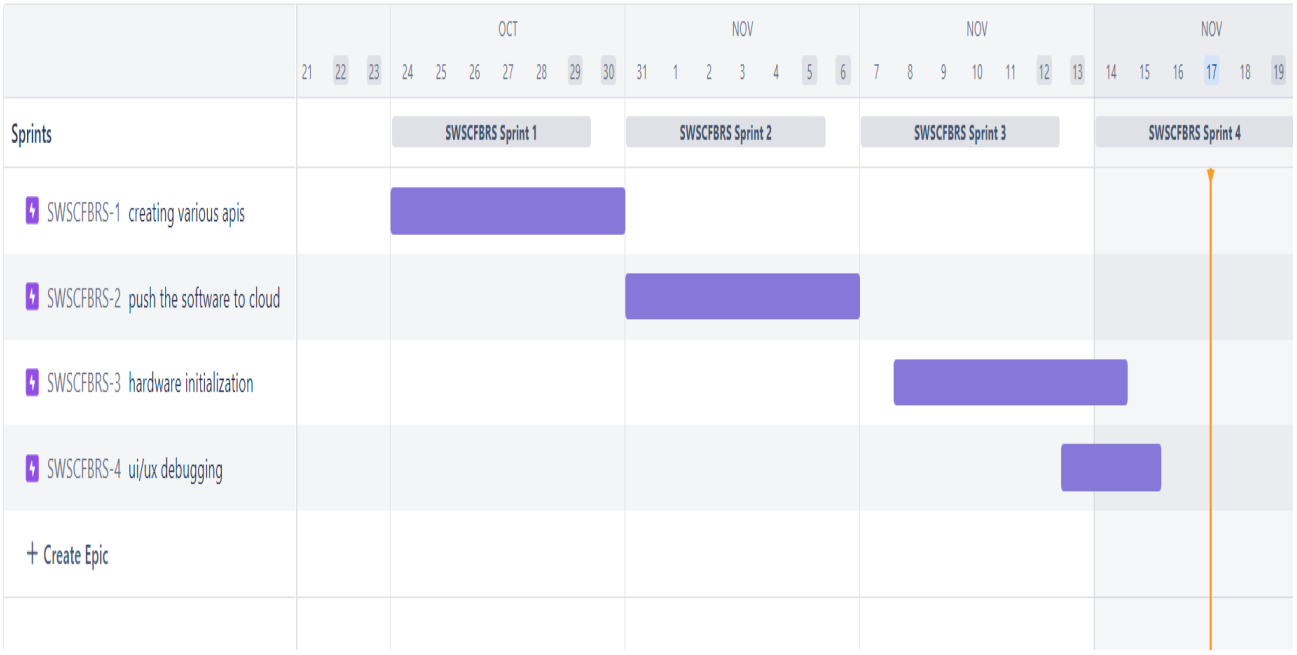
Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

Burndown Chart:

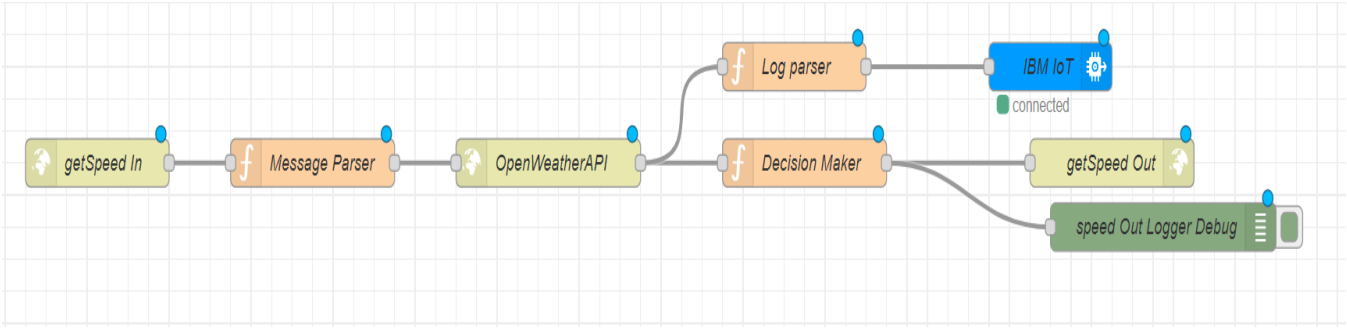


6.3 Reports from JIRA Software



7. CODING AND SOLUTIONING

7.1 Feature 1 - GET SPEED FOR GIVEN LOCATION & CLIMATE



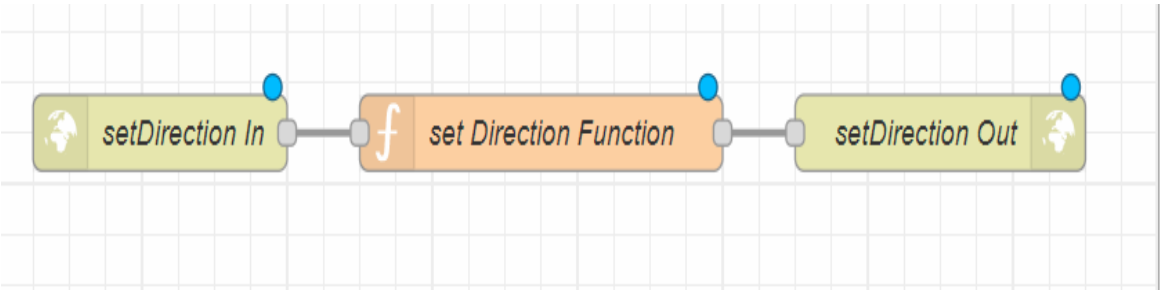
This part of Node RED flow accepts an http GET end point at **"/getSpeed"** from which the location, uid, hospital/school zone info are passed.

Message parser sets the required APIKEY for **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 microcontroller. Thus, a lot of battery is saved due to lesser processing time.

7.2 Feature 2 - SET DIRECTION REMOTELY FOR A GIVEN SIGN BOARD



This part of Node RED flow accepts an **http GET** end point at **"/setDirection"** from which 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 workflows is available at the **following link**:

8. TESTING

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

8.2 User Acceptance Testing

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

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, Node RED is capable of handling up to 10,000 requests per second. Moreover, since the system is horizontally scalable, an even higher demand of customers can be served.

10. ADVANTAGES & DISADVANTAGES

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

- **DISADAVNTAGES**
  - The size of the display determines the requirement of the micro controller
  - Dependent on OpenWeatherMap API and hence the speed reduction is same for a large area in the scale of cities.

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

12. 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.

13. APPENDIX

- **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";
String usualSpeedLimit = "70"; // kmph

int schoolZone = 32;
int hospitalZone = 26;

int uid = 2504;

String getString(char x)
{
    String s(1, x);
    return s;
}

String stringSplitter1(String fullString,char delimiter='$')
{
    String returnString = "";
    for(int i = 0; i<fullString.length();i++) {
        char c = fullString[i];
        if(delimiter==c)
            break;
        returnString+=String(c);
    }
    return(returnString);
}

String stringSplitter2(String fullString,char delimiter='$')
{
    String returnString = "";
    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;
    }
    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);
}

void upArrow()
{
  int refX = 125;
  int refY = tft.getCursorY() + 30;

  tft.fillTriangle(refX-
40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED);
  tft.fillRect(refX-15,refY+40,30,20,ILI9341_RED);
}

String APICall() {
  HTTPClient http;

  String url = "https://node-red-nwmrt-2022-11-04.eu-
gb.mybluemix.net/getSpeed?";
  url += "location="+myLocation+"&";
  url += "schoolZone="+(String)digitalRead(schoolZone)+(String)"&";
  url +=
"hospitalZone="+(String)digitalRead(hospitalZone)+(String)"&";
  url += "usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";
  url += "uid="+(String)uid;
  http.begin(url.c_str());
  int httpResponseCode = http.GET();

  if (httpResponseCode>0) {
    String payload = http.getString();
    http.end();
```

```
    return(payload);
  }
  else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
  http.end();
}

void myPrint(String contents) {
  tft.fillScreen(ILI9341_BLACK);
  tft.setCursor(0, 20);
  tft.setTextSize(4);
  tft.setTextColor(ILI9341_RED);
  //tft.println(contents);

  tft.println(stringSplitter1(contents));
  String c2 = stringSplitter2(contents);
  if(c2=="s") // represents Straight
  {
    upArrow();
  }
  if(c2=="l") // represents left
  {
    leftArrow();
  }
  if(c2=="r") // represents right
  {
    rightArrow();
  }
}

void setup() {
  WiFi.begin(ssid, password, 6);

  tft.begin();
  tft.setRotation(1);

  tft.setTextColor(ILI9341_WHITE);
  tft.setTextSize(2);
  tft.print("Connecting to WiFi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    tft.print(".");
  }
```

```
  tft.print("\nOK! IP=");
  tft.println(WiFi.localIP());
}

void loop() {

  myPrint(APICall());

  delay(100);
}
```