

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

IBM PROJECT REPORT

Team ID -

PNT2022TMID14966

Project Report by

Manoj kumar M-111519106085

Harikrishnan T-111519106045

Jegan S-111519106057

Kirthik Saran P-111519106070

Bachelor of Engineering In

R.M.D Engineering College
Kavaraipettai

Academic Year:2022-2023

Final Deliverables Report

Date	18.11.2022
Team ID	PNT2022TMID14966
Project Name	Signs with Smart Connectivity for Better Road Safety

CONTENTS

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2. LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition
- 2.4 IDEATION & PROPOSED SOLUTION
- 2.5 Ideation & Brainstorming
- 2.6 Proposed Solution
- 2.7 Problem Solution fit

3. REQUIREMENT ANALYSIS

- 3.1 Functional requirement
- 3.2 Non-Functional requirements

4. PROJECT DESIGN

- 4.1 Data Flow Diagrams
- 4.2 Solution & Technical Architecture
- 4.3 User Stories

5. PROJECT PLANNING & SCHEDULING

- 5.1 Sprint Planning & Estimation
- 5.2 Sprint Delivery Schedule

6. CODING & SOLUTIONING

6.1 Feature 1

6.2 Feature 2

7. TESTING

7.1 Test Cases

7.2 User Acceptance Testing

8. RESULTS

8.1 Performance Metrics

9. ADVANTAGES & DISADVANTAGES

10. CONCLUSION

11. FUTURE SCOPE

12. APPENDIX

Source Code

GitHub & Project Demo Link

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

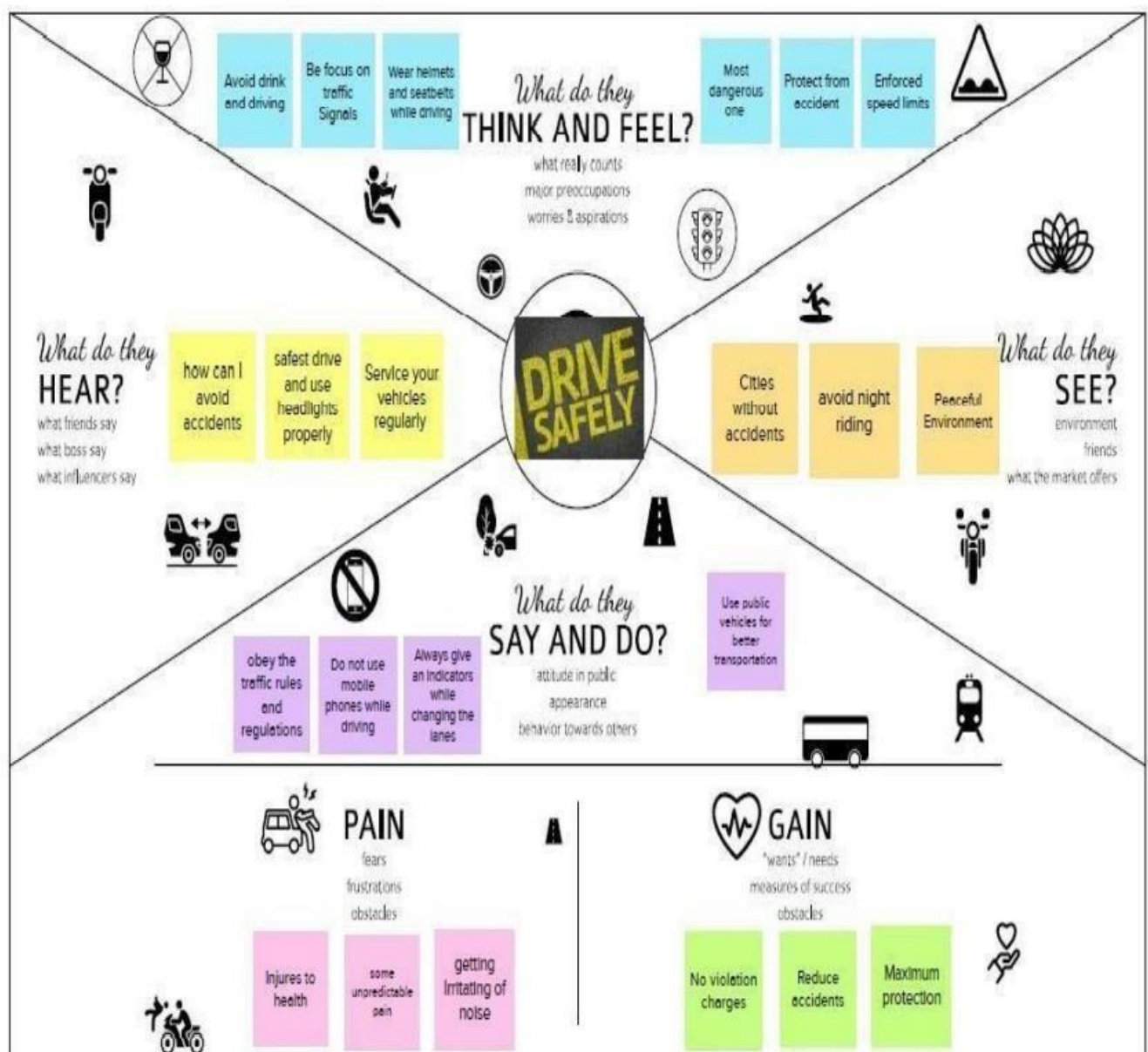
- [Islam, 2015; Sisiopiku et al., 2015](#)
- [Yannis et al., 2013, Staffeld \(1953\) and Ady \(1967\)](#)

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

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP



You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

JEGAN S

Avoid Drunk and Driving	Ensures safety measures	Drive the vehicle in correct speed limit
Manage stress	Roads need to be in good condition	

HARIKRISHNAN T

Rash driving need to be avoided	Understanding the road sign	Optimizing the functions of the vehicle
Use of Helmets properly	Indication of seat belt	

KIRTHIK SARAN P

Use of public transports	Speed need to be reduced near schools	Detection of turning
Avoid using mobile while driving	Service your car regularly	

MANOJ KUMAR M

Automatic detection of traffic light	Minimizing the use of own vehicles	Do not cross the speed limits.
Maintain lane discipline.	Maintain a safe distance.	

Prioritize

🕒 20 minutes



After you collaborate

Quick add-ons

- Keep moving forward

-  [Share template feedback](#)

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> ➤ To prevent road accidents by ensuring suitable measures using IOT.
2.	Idea / Solution description	<ul style="list-style-type: none"> ➤ By Preparing smart signs using IOT instead of regular signs hung on the road. ➤ These smart signs provide a better clarity and are built with IOT and LED.
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> ➤ Since LED is highly energy efficient – Less heat, more light, lower cost. Use less electricity for the same light output. ➤ The smart signs consist of temperature, humidity, wind speed. ➤ This information is received from weather monitoring app. ➤ It also gives information about nearby Places such as hospitals, schools, etc. so that the users can decide their speeding according to that information.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> ➤ These create a noticeable impact on the road safety department. ➤ By deciding a speed limit for the user, there is significant chance in reducing the accidents.
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> ➤ By executing these for commoners by the government, it is great initiative increasing an awareness among the people. ➤ A separate budget can be allotted for this by the government, which paves away for a safer environment.
6.	Scalability of the Solution	<ul style="list-style-type: none"> ➤ It has greater chance in reducing the risk for the people as it is more visible than the normal signs, which saves a lot of lives at stake.

3.4 Problem Solution fit

Project Title: Signs with Smart Connectivity for Better Road Safety		Project Design Phase-I - Solution Fit Template		Team ID: PNT2022TMID14966	
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? I.e. working parents of 0-5 y.o. kids Highway Signals	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? I.e. spending power, budget, no cash, network connection, available devices. When the connection or network breaks it causes the communication between the vehicle and the cloud breaks.	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? I.e. pen and paper is an alternative to digital notetaking In the roadways, signs with correct directions gives clear solution.	Explore AS, differentiate	
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. The boards which are used for signaling should keep track of temperature sensor and should inform the board about the speed of Customer's vehicle.	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? I.e. customers have to do it because of the change in regulations. If the connection between the sensor and the board breaks there will be weather alerts. Unnecessary pressing of indicator buttons by anyone would lead to problems.	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? I.e. Directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) The IOT cloud updates the smart boards so that the customer can get the job done easily.		
Focus on J&P, tap into BE, understand RC	3. TRIGGERS TR What triggers customers to act? I.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. The sensors in the board will alert the customers will by displaying about the weather. As weather causes most of the accidents.	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. With the help of weather API and an application we can use smart boards in alternative to static sign boards which informs about the weather conditions ahead. It provides instant support and safe journey.	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 The officers can receive the messages from from the customers.	Identify strong TR & EM	
	4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? I.e. lost, insecure → confident, in control - use it in your communication strategy & design. Based on the readings of the sensor the customers can change operation mode with the use of smart board and will follow the instructions from the smart board.	8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. By using smartboards to check the state of the weather and roads instead of only direction boards.			

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 signs correctly so that it is visible even to far awaydrivers

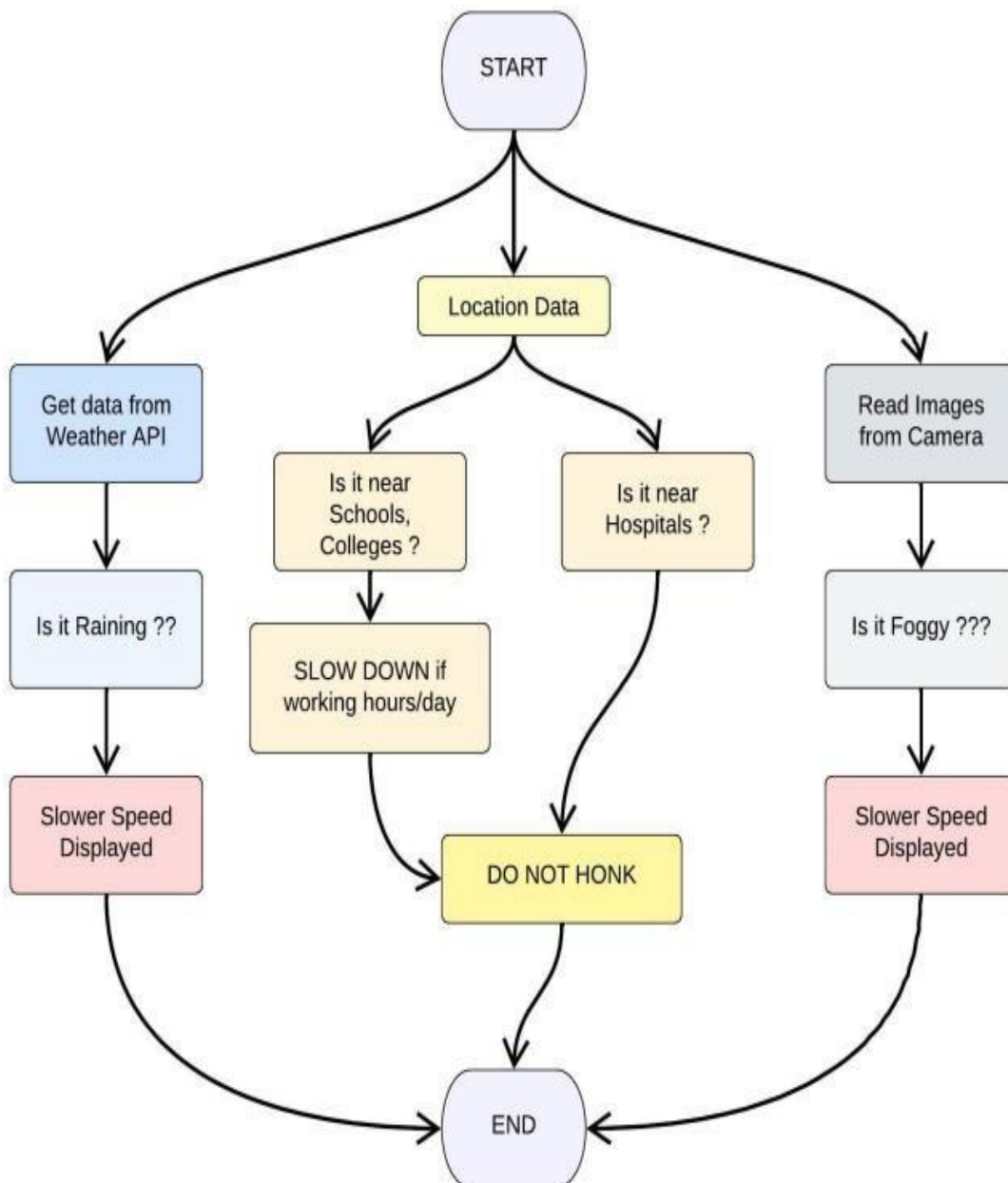
4.2 Non-Functional Requirements

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 respect to time.
NFR-2	Security	Should be secure enough that only the intended messages are displayed in the display.
NFR-3	Reliability	Should convey the traffic information correctly.
NFR-4	Performance	Display should update dynamically whenever the weather 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 on servers horizontally.

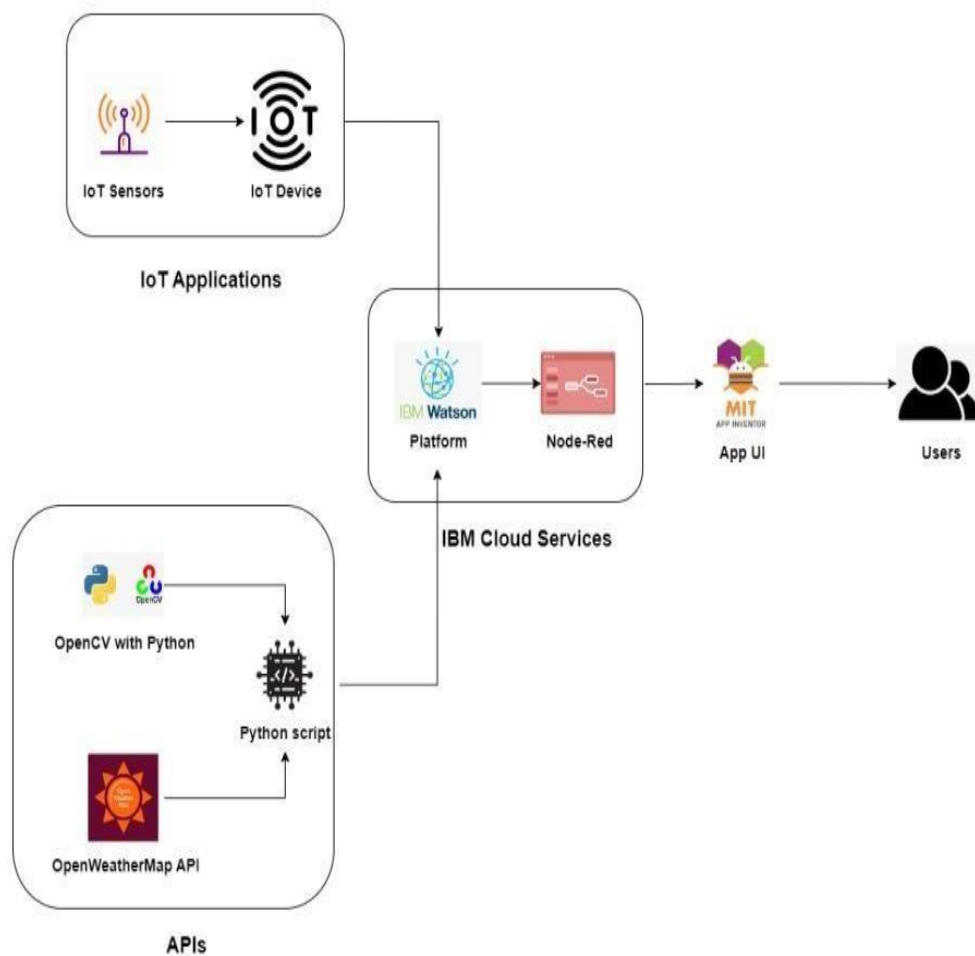
5. PROJECT DESIGN

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

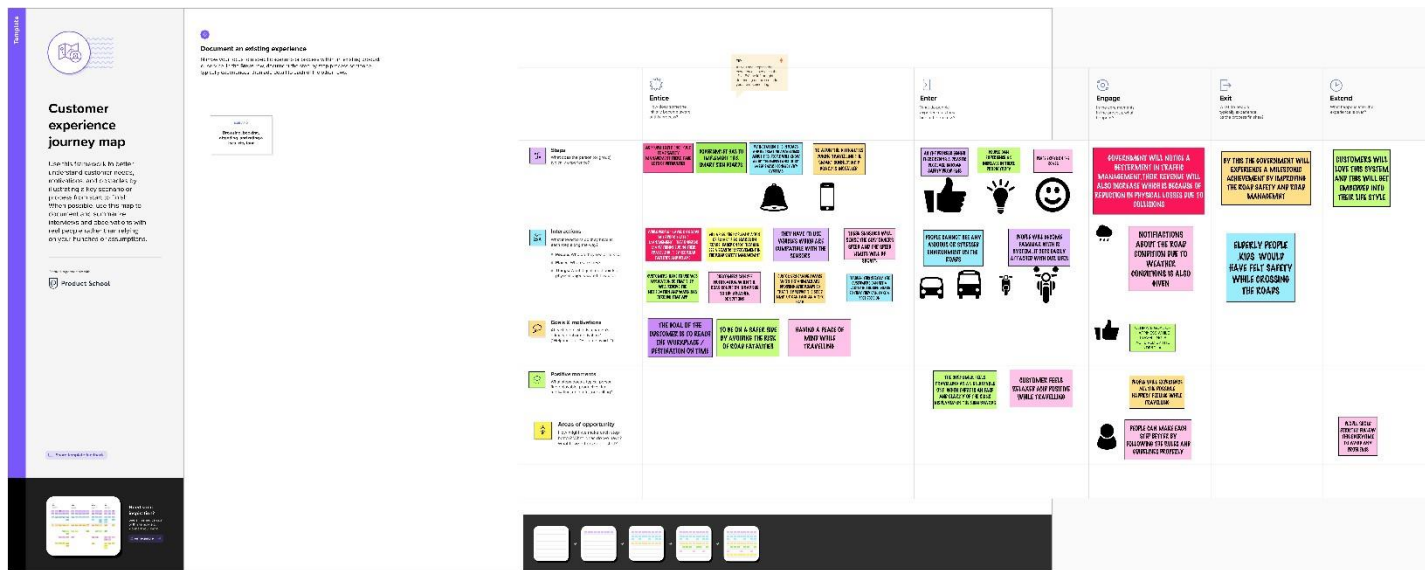
Table-1 : Components & Technologies:

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

5.3 User Stories



6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initialization of All required Services	USN-1	Initialize services like OpenWeatherMap, NodeRED, etc.	2	High	Manoj kumar Harikrishnan Jegan Kirthik Saran
Sprint-2	Implementation of Code	USN-2	Coding to integrate all services as one	2	High	
Sprint-3	Hardware Integration	USN-3	Hardware implementation on IoT Enabled Device	2	Medium	
Sprint-4	Optimization	USN-4	Bug fixes and improvements	2	Low	

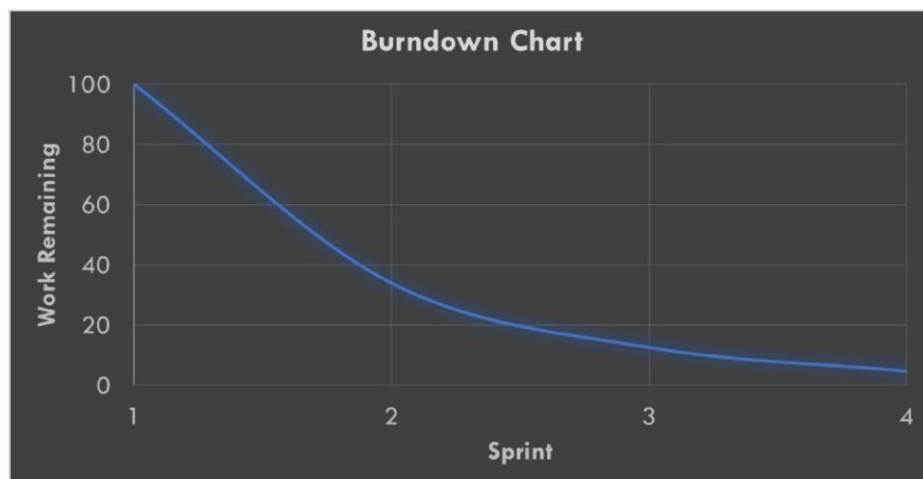
6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart:

Project Tracker, Velocity & Burndown Chart: (4 Marks)

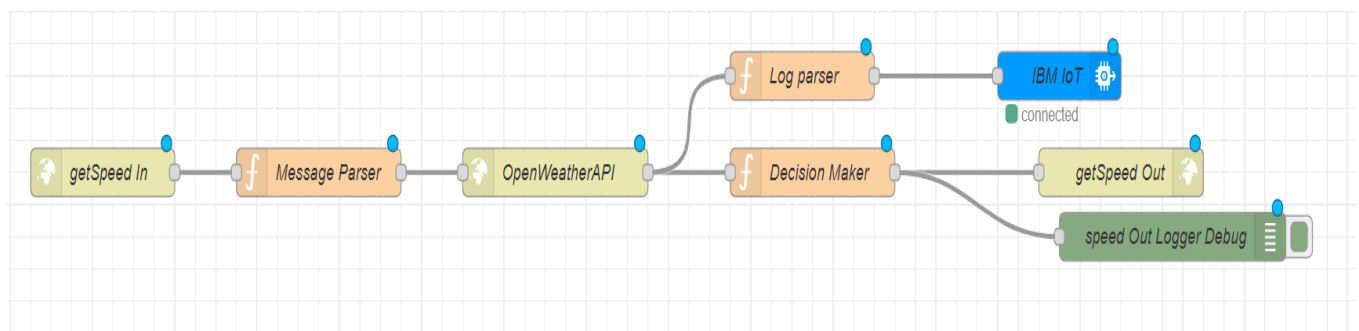
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	18 Nov 2022

Burndown Chart:



7. CODING AND SOLUTIONING

7.1 Feature 1 - GET SPEED FOR GIVEN LOCATION & CLIMATE



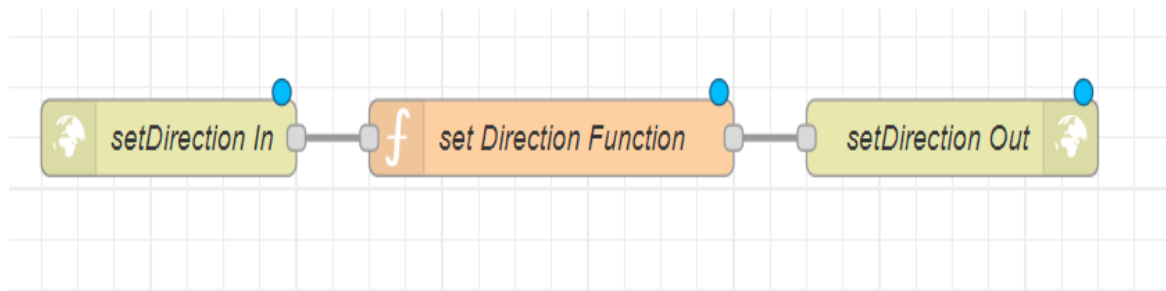
This part of Node RED flow accepts an http GET end point at **"/getSpeed"** from which the location, uid, hospital/school zone info are passed.

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.

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

- **DISADVANTAGES**

- 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 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) could result in smoother traffic flows and, what is more important, in increasing a driver's awareness of the road situation.

13. APPENDIX

- **GITHUB:**

<https://github.com/IBM-EPBL/IBM-Project-34101-1660231714>

- **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 httpStatusCode = http.GET();

    if (httpStatusCode>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);  
}
```