Internet Of Things

Signs with Smart Connectivity For Better Road Safety

completion of project

Team members

SUNDAR RAJAN M
TARUN G
NIBIN P N
SUDHARSON P

Team ID: PNT2022TMID22491

Date
18 November 2022

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

VEL TECH MULTI TECH
Dr.RANGARAJAN Dr.SAKUNTHALA ENGINEERING COLLEGE
(Approved by AICTE New Delhi, Affiliated to Anna University, Chennai & ISO 9001:2008 Certified Institution & Accredited by NBA New Delhi)

S.NO	TITLE	PAGE NO
1	Introduction	3
2	Literature survey	4
3	Ideation & Proposed Solution	5
4	Requirement Analysis	10
5	Project Design	11
6	Project Planning & Scheduling	20
7	Requirements	21
8	Coding & Solutioning	23
9	Testing	31
10	Results	31
11	Advantages & Disadvantages	32
12	Conclusion	32
13	Github link	32

INTRODUCTION

Traffic has recently become a big issue for the people of India. As a result, it wastes valuable time, fuel, and electricity. The Internet of Things (IOT) is a network of electrical appliances, cars, physical devices, and other items thatare integrated with electronics, actuators, sensors, software, and connectivity, allowing these objects to connect and share data. Each object is uniquely identifiedby its embedded computing system, but it may interact with the existing Internetinfrastructure.

Project Overview

In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases whenthereare some road diversions due to heavy traffic or due to accidents then we canchange the road signs accordingly if they are digitalized. This project proposes asystem which has digital sign boards on which the signs can be changed dynamically. By using the Weather API we can get the weather reports based on which we can set the speed limit to particular area. If there is rainfall thentheroads will be slippery and the speed limit would be decreased. There is a webappthrough which you can enter the data of the road diversions, accident prone areasand the information sign boards can be entered through web app. This datais retrieved and displayed on the sign boards accordingly. There are three switches through which you can switch the display to different modes.

Purpose

Due to this heavy traffic, the number of road accidents are increased which is a major issue. Our project helps to decrease the number of road accidents using smart connected sign boards using Internet Of things (IOT).

LITERATURE SURVEY
Existing System

The individual traffic signals are connected with traffic control system to perform network wide traffic operation .These control systems contain a central computer, a communication network, and intersection traffic signals. Coordination of control system can be implemented through different techniques like time-base, hardwired interconnection method. Coordination between traffic signals and agencies requires the development of data sharing and traffic signal control agreements. A traffic-signal system has only one purpose i.e. to deliver signal timings to the driver. The system provides features that improve the traffic engineer's ability to achieve this goal. These are primarily access features. They provide access to the intersection signal controller for maintenance and operations. The more complete and convenient the access, the more efficient the operator will be and the more effective the system. In addition to control the traffic signals, modern technology also provide surveillance capabilities, including different kinds of video surveillance and traffic detection.

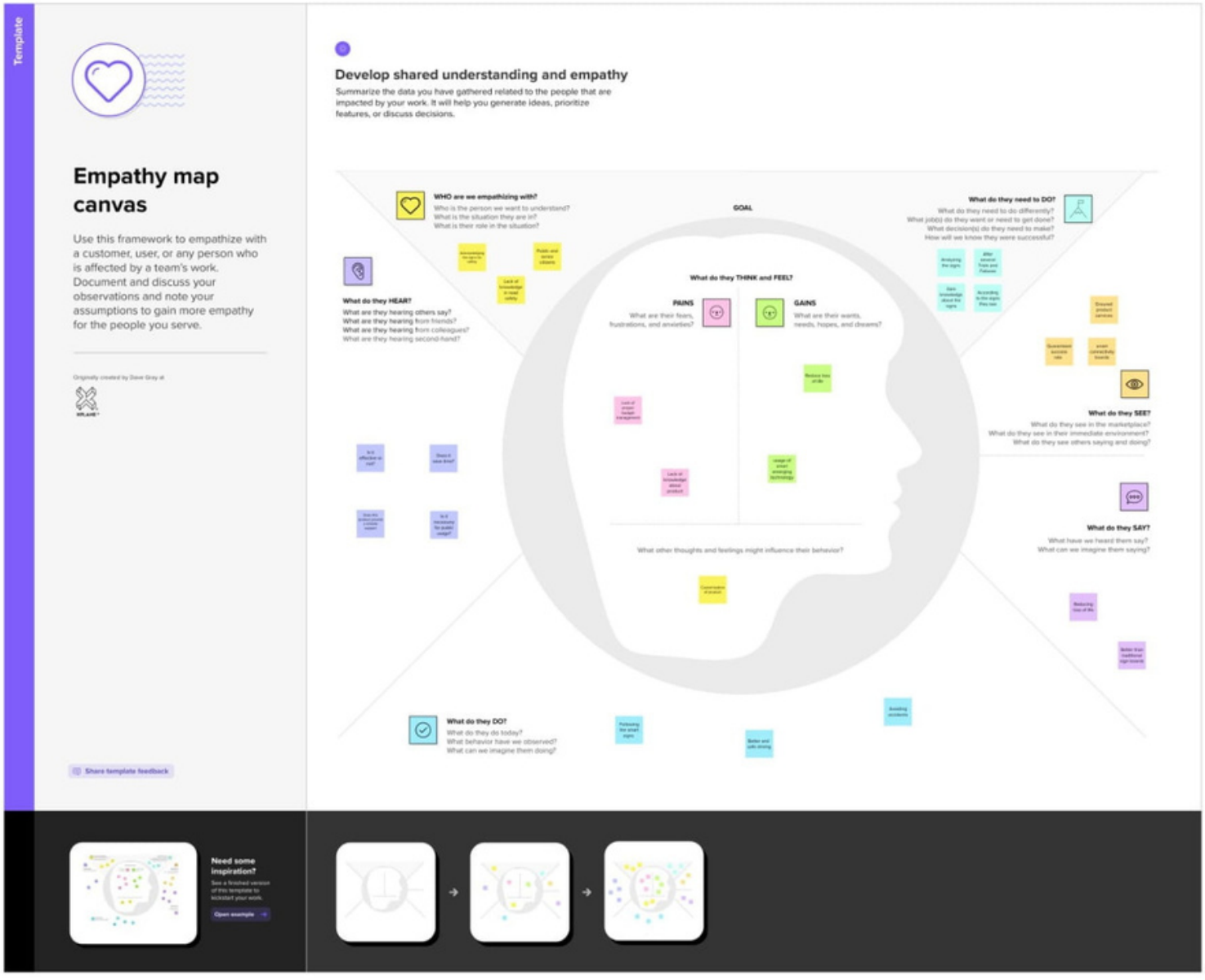
Problem Statement Definition

This project will replace static signs with smart signs that can adjust speed restrictions based on the weather and climate, display detour instructions in the event of an accident, and display alert messages in the event of hospitals, schools, or roadworks.

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

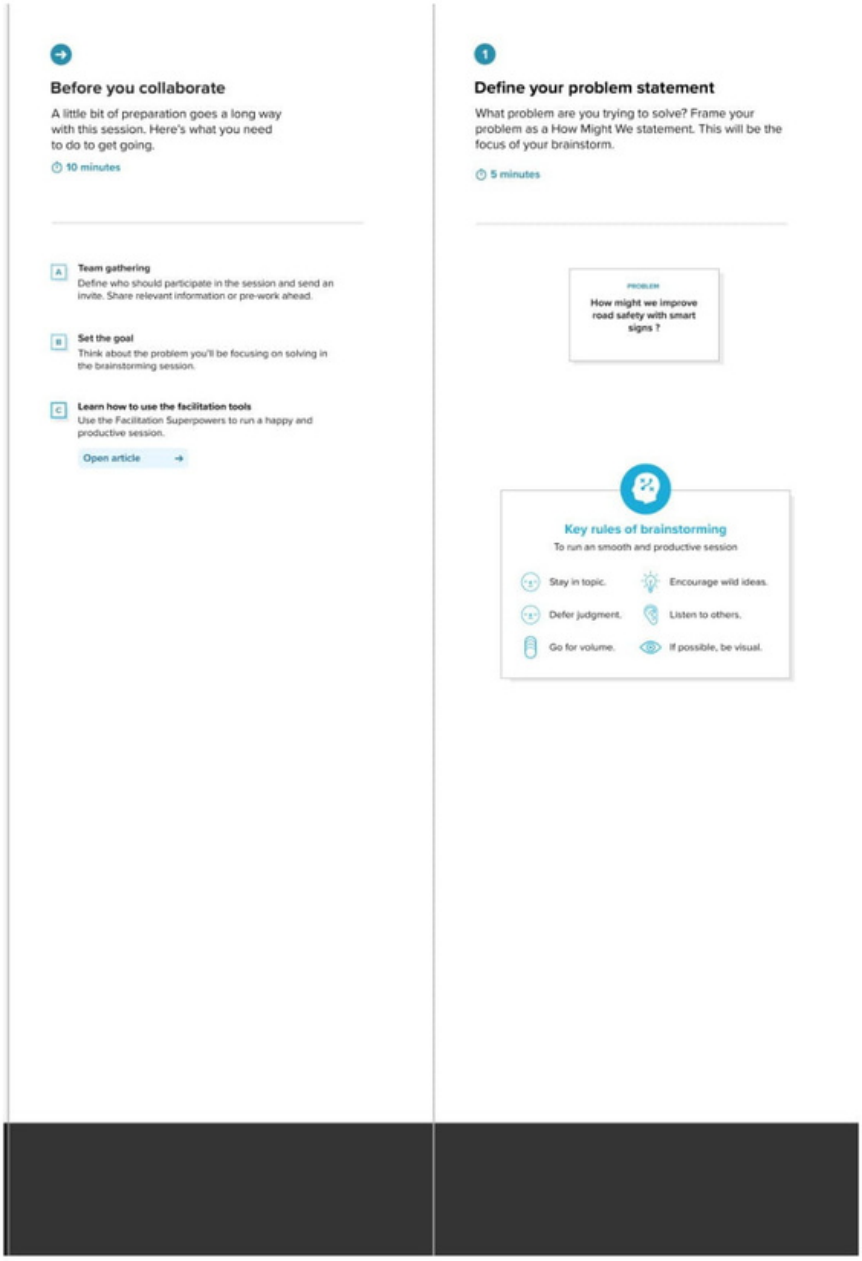


Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/empathy-map-canvas

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

Project Design Phase-I

Proposed Solution

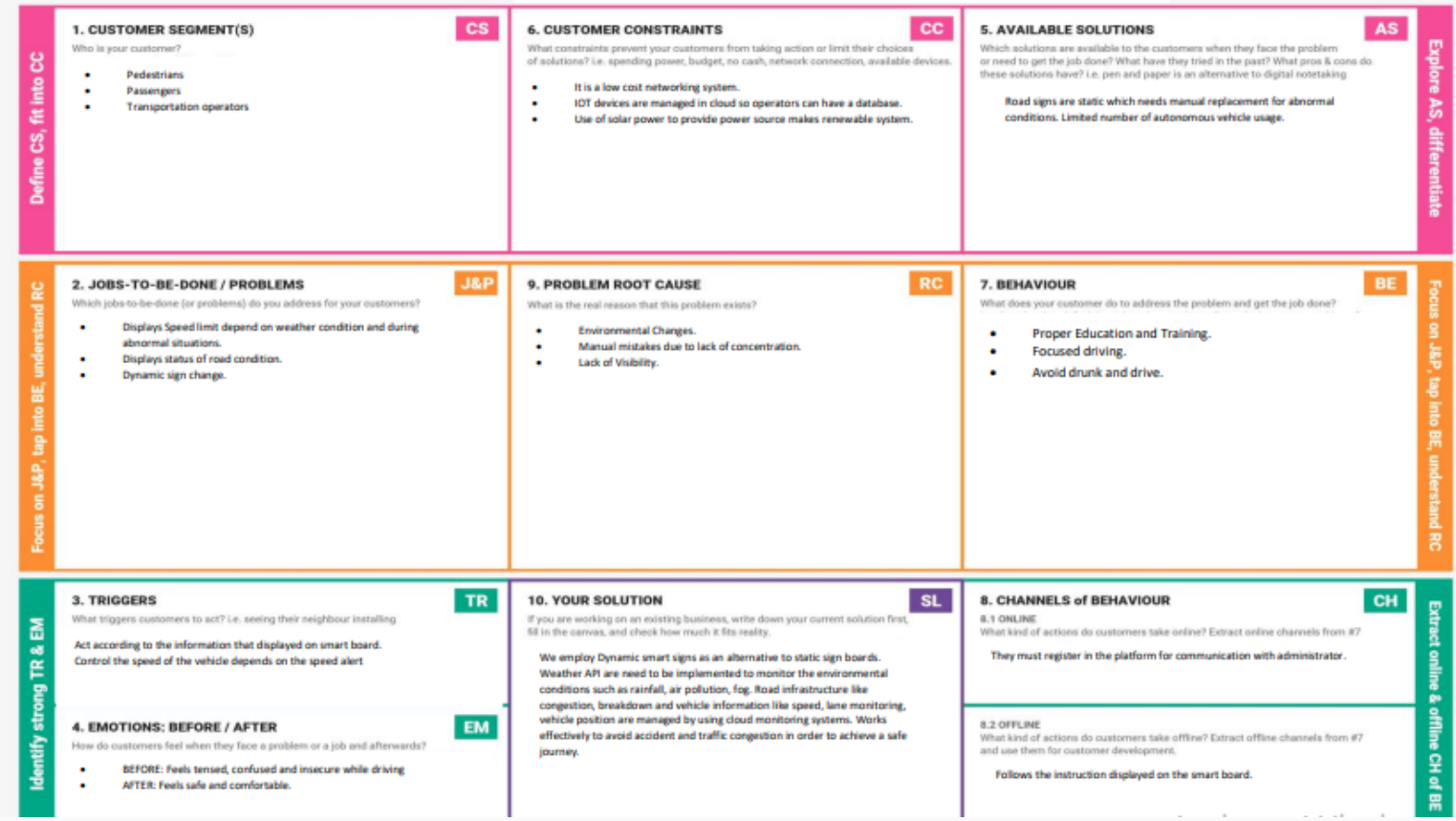
1	**Problem Statement**	To replace the static signboards, smart connected sign boards areused. Because These smart connected sign boards get the speed limitations from aweb application using weather API and update it automatically. Based on the weather changes the speed may increase or decrease gradually Based on the traffic situations the diversion signs are displayed. Guide (Schools), Warning and Service (Hospitals) signs are also displayed accordingly.
2	**Idea description**	The weather and temperature details are obtained from the Open Weather Map API. Using these details, the speed limit will be updated automatically with the weather conditions. Also,details regarding any accidents and traffic faced on the particular road are obtained. Based on this the traffic is divertedfollowed by a change in map path and the traffic is cleared. Changing the warning signs, which are predefined and separate signs will be present for both school and hospital zones

Project Design Phase-I

Proposed Solution

3	**Novelty**	Generic Sign board for all the applications that uses both buttons and webservice.
4	**Idea description**	If there is no traffic, can cross the street without waiting. Customer can reach the destination before the expected time
5	**Business Model**	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 accidents in the particular road. The public will also gain all the information about the road, even if they are checking for analternate path

Problem Solution Fit



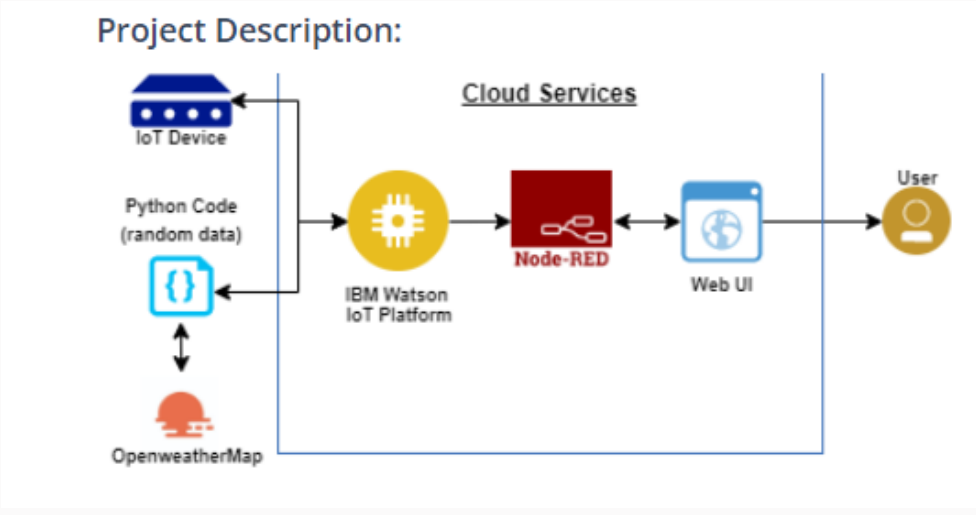
Problem Solution Fit

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• To replace the static signboards with digital smart connected signboards which updates the realtime signs from the weather API automatically.

• This project contains several process before the final smart sign boards this includes WEB UI, node RED & cloud Watson IOT platform for cloud update to the boards.

• To Assist the end users such as NHAI ,Pedestrians ,Traffic police & travellers Some of salient features was added adaptable by current environment and mode can be selected by provided buttons .
• This solution requirements are python code, cloud services & weather API .



Project Design Phase-II

Technology Stack (Architecture & Stack)

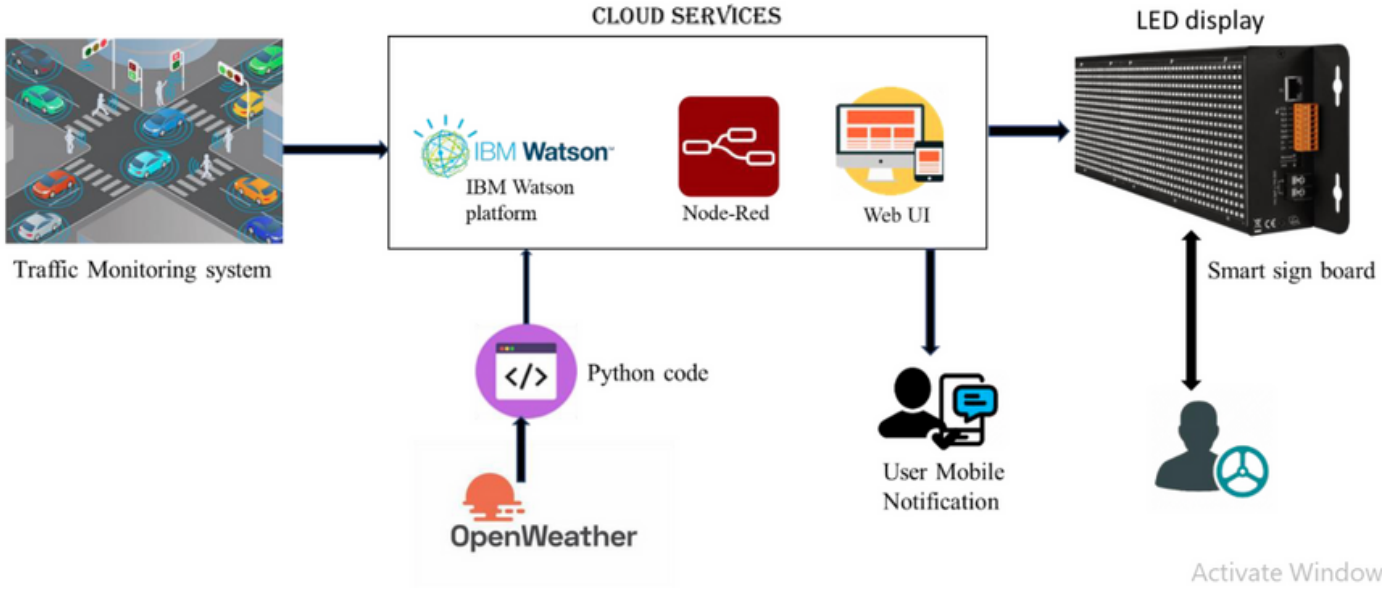
S.no	Component	Description	Technology
1	User Interface feature	How user interacts with application e.g.Web UI	HTML, CSS, JavaScript (Web application)
2	Application Logic-1	Logic for a process in the application	Python
3	Application Logic-2	Logic for a process in the application	IBM Watson STT service (Cloud)
4	Application Logic-3	Logic for a process in the application	IBM Watson Assistant (Cloud)
5	Database	Data Type, Configurations etc	MySQL
6	Cloud Database	Database Service on Cloud	IBM DB2
7	File Storage	File storage requirements	Local Filesystem
8	Infrastructure	Server Application Deployment on Local	Local Server
9	Cloud	System / Cloud	Configuration: Local System Cloud Server Configuration: IBMWatson (Cloud)

Table-2:

Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used	IoT devices,OpenweatherMap,IBM Watson,Node-RED,Web UI
2	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	Encryptions,Decryptions
3	scalability	Justify the scalability of architecture	Python
4	Availability	Justify the availability of application	IBM Watson – Can easily beaccessed

Technical Architecture:



Solution Requirements
(Functional &Non-functional)

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	Sign Boards should be made with LED’s which arebright colored and are capable of attracting the drivers attention but it should also not be too distracting or blinding cause it may lead to accidents. Red light can be bright than others due to higher wavelength
FR-2	User Need	The smart sign boards should be placed frequently inplaces it is needed and less in places where it is not needed much to avoid confusion for the user during travel. It is vital that drivers are aware of their own limitations
FR-3	User Understanding	For better understanding of the driver, the signsshould be big, clear and legible and it can also include illustrations which will make it easily understandable to the driver.
FR-4	User Convenience	The display should be big enough that it should even be visible from far distance clearly.

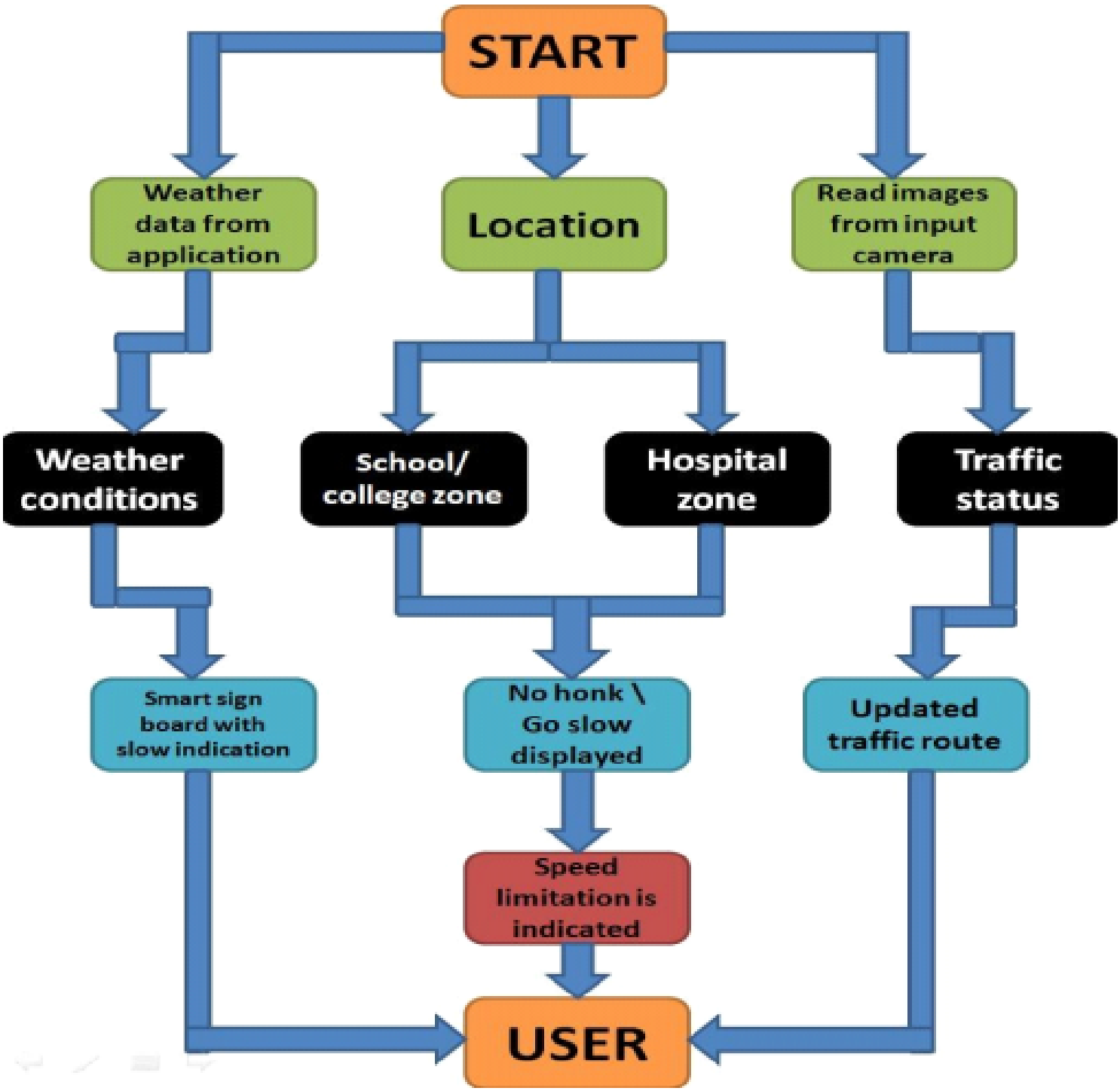
Non-functional Requirements:

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

FR No	Non - Functional Requirement	Description
NFR-1	Usability	It should be able to Upgrade and Update when there is a need for it.
NFR-2	Security	It should have good security system so that no other person is able to hack and display their owndirections. The technology enables you to control traffic, catch the lawbreakers,and provide road safety
NFR-3	Reliability	It should be able to display to information correctly and error-free.
NFR-4	Performance	It should be able to automatically update itself when certain weather or traffic problem occurs.
NFR-5	Availability	It should be available 24/7 so that it can bebeneficial to the customer i.e the driver

Data Flow Diagram User Stories

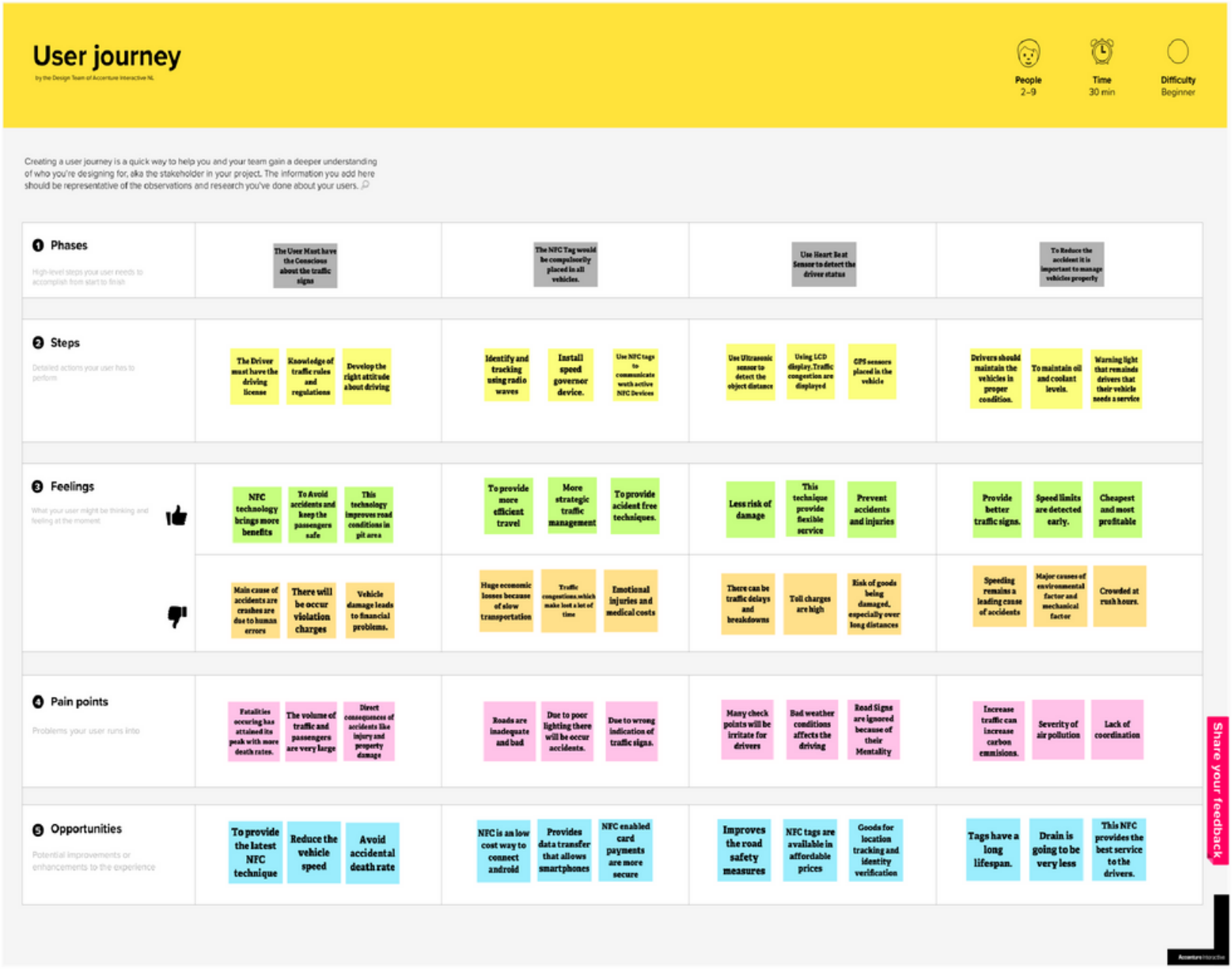
Data Flow Diagram:



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer(Mobile user)	Registration	USN-1	I can get my speedlimitation using weather application.	I can receive speed limitations	High	Sprint-1
		USN-2	As a user, I can registerfor the application byentering my email, password, and confirmingmy password. As a user,	I can access my account /dashboard	Medium	Sprint-2
		USN-3	As a user, I can increaseor decrease myspeed according to the weatherchange	I can increase or decrease my speed	High	Sprint-1
		USN-4	As a user, I can I get mytraffic diversion signs depending on the trafficand the fatalsituations.	I can access my traffic status ahead in mytravel	Medium	Sprint-1
	Login	USN-5	As a user, I can log intothe open weathermap byentering email & password	I can accessthe application through myGmail login	High	Sprint-2
	Interface	USN-6	As a user the interface shouldbe simple andeasily accessible	I can access the interfaceeasily	High	Sprint-1
Customer(Web user)	Data generation	USN-7	As a user I use open weather application to access the data regarding the weather changes.	I can access the data regarding the weather through the application	High	Sprint-1

CUSTOMER JOURNEY



MILESTONES AND ACTIVITY LISTS

1. Preparation Phase (22.08.22 to 27.08.22)

- Pre-requisites
  - **Pre-requisites**

    IBM Cloud
    Software

  - **Project Objectives**

    Abstract
    Brainstorming

  - **Create and Configure IBM Cloud Services**

    Create IBM Watson IoT Platform and Device
    Create Node Red Service
    Create a Database in Cloudant DB

  - **Develop the Python Script**

    Develop a Python Script

  - **Develop a web application using Node RED Service**

    Develop a web application using Node RED

    Registrations
    Environment Set-up

2. Ideation Phase (29.08.22 to 17.09.22)

- Literature Survey
- Empathize
- Defining Problem Statement
- Ideation

3. Project Design Phase 1 (19.09.22 to 09.10.22)

- Proposed Solution
- Problem Solution Fit
- Solution Architecture

4. Project Design Phase 2 (09.10.22 to 25.10.22)

- Requirement Analysis
- Customer Journey
- Data flow Diagram
- Technology Architecture

5.Project Planning Phase (25.10.22 TO 30.10.22)

- Milestone and Tasks
- Sprint Schedules

6. Project Development Phase (30.10.22 to 19.11.22)

- Coding and Solutioning
- Acceptance Testing
- Performance Testing

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Product Backlog, Sprint Schedule and Estimation(4Marks)

Use the below template to create product backlog and sprint scheme

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	Sundarrajan Tarun sudharson Nibin
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location	1	MEDIUM	Sundarrajan Tarun sudharson Nibin
Sprint-2	Push the server/software to cloud	Push the code from Sprint1 to cloud so it can be accessed from anywhere	2	MEDIUM	Sundarrajan Tarun Sudharson Nibin
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same	2	HIGH	Sundarrajan Tarun sudharson Nibin

Sprint-4	UI/UX Optimization & Debugging	Optimize all the short comings and provide better user experience	2	LOW	Sundarrajan Tarun sudharson Nibin

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

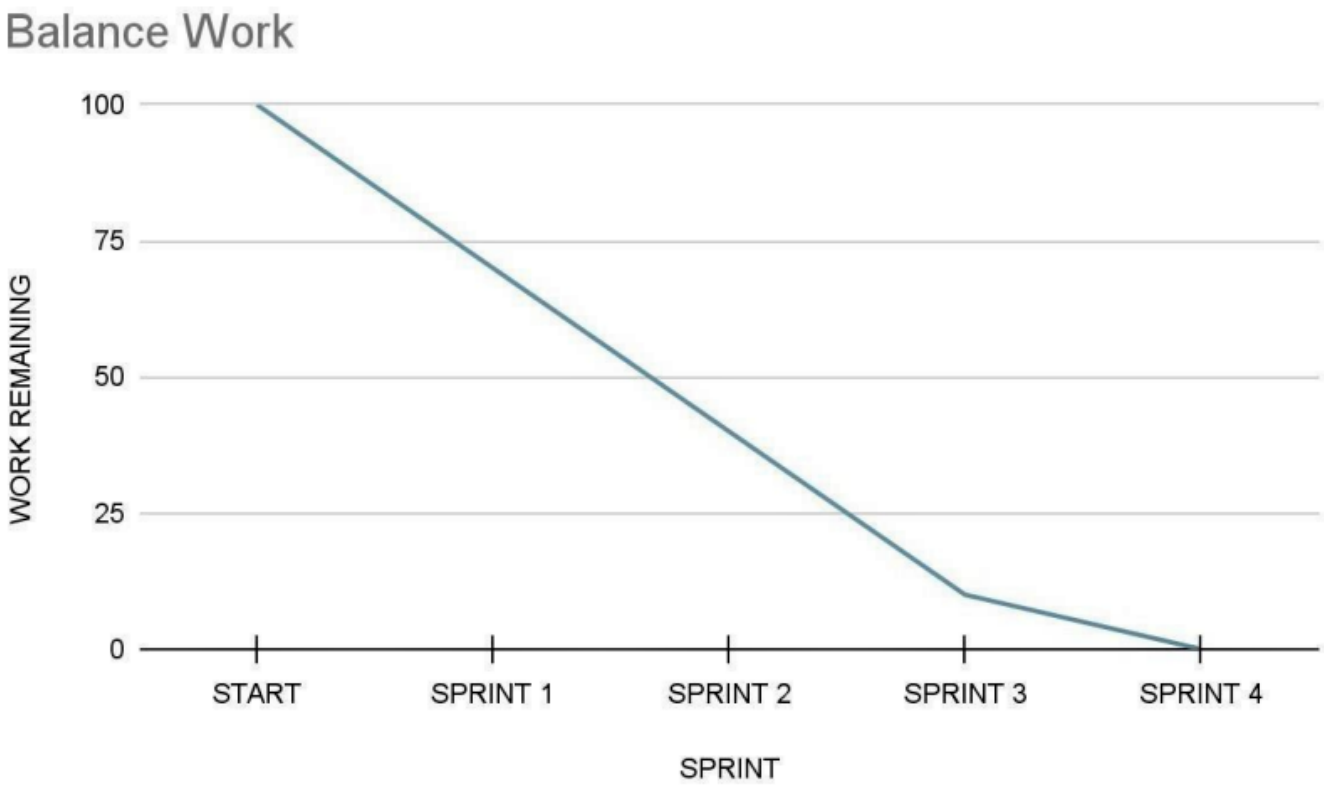
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6Days	24Oct2022	29Oct2022	20	27Oct2022
Sprint-2	20	6Days	31Oct2022	05Nov2022	20	02Nov2022
Sprint-3	20	6Days	07Nov2022	12Nov2022	20	09Nov2022
Sprint-4	20	6Days	14Nov2022	19Nov2022	20	15Nov2022

Velocity:

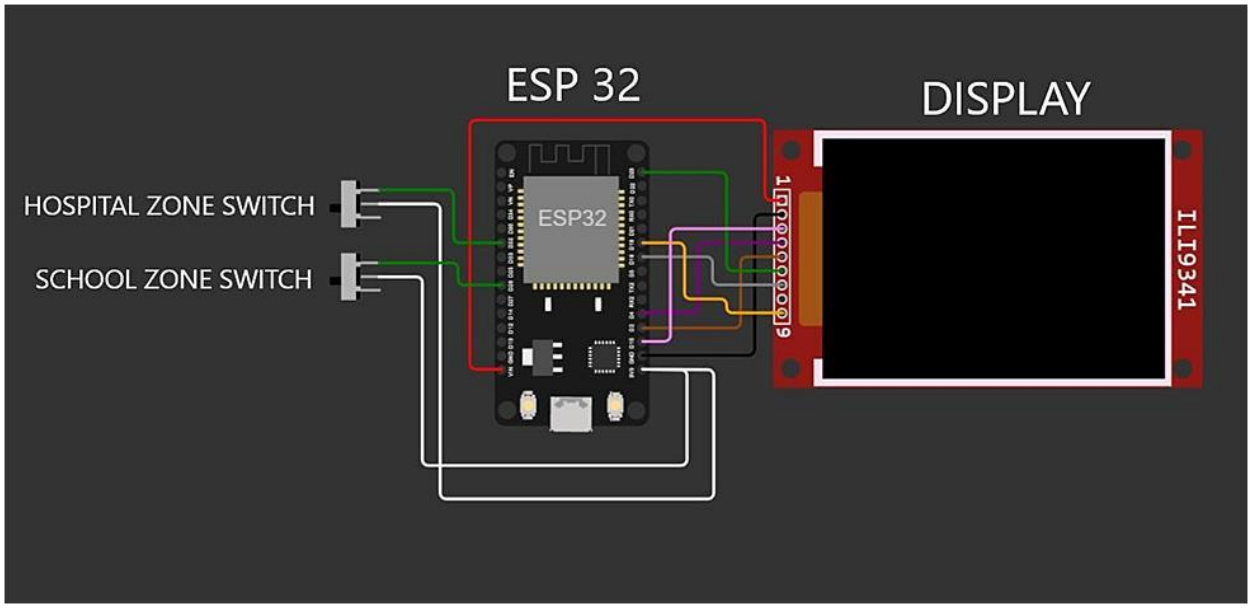
Imagine we have a 10-days print duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity(AV) per iteration unit(story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:



Circuit Diagram:



ESP 32 CODE :

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

const char* ssid = "WokwiGUEST";
const char* password =
"";

#defineTFT_DC 2
#defineTFT_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; // ID Unique to this MicroContoller


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

StringstringSplitter1(String fullString,char delimiter='$')
{
   StringreturnString = "";
   for(int i = 0; i<fullString.length();i++)
      { char c = fullString[i];
      if(delimiter==c)
         break;
      returnString+=String(
      c);
    }
   return(returnString);
}

StringstringSplitter2(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
```

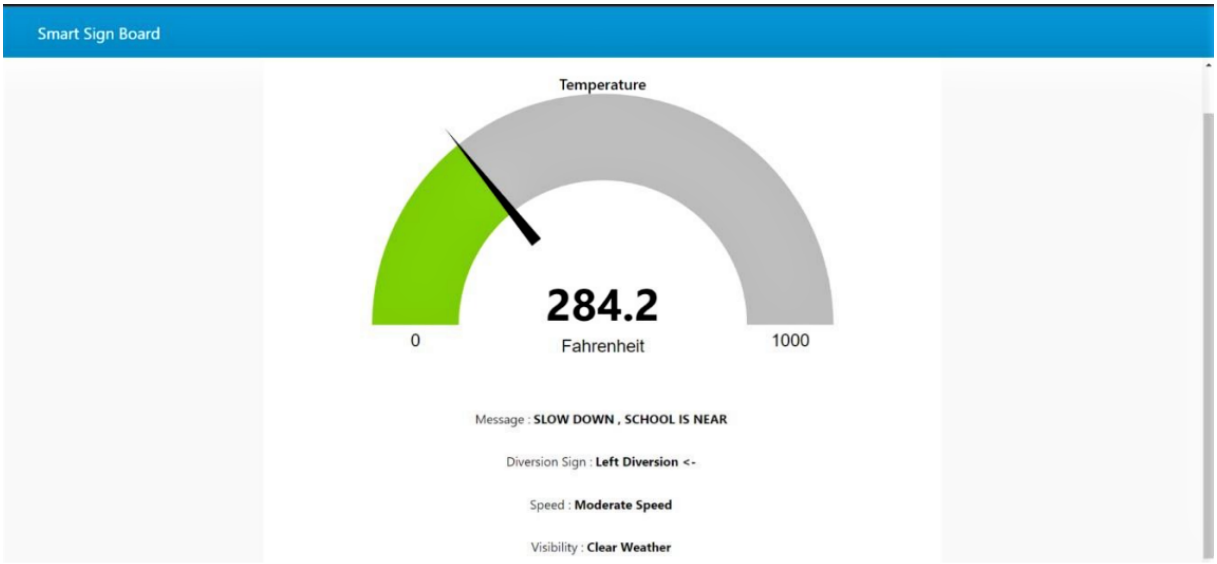
```
        );if(delimiter==c)
            flag = true;
    }
    return(returnString);
}

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-grseb-2022-11-
05test.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_RE
  D);
  //tft.println(contents);
```

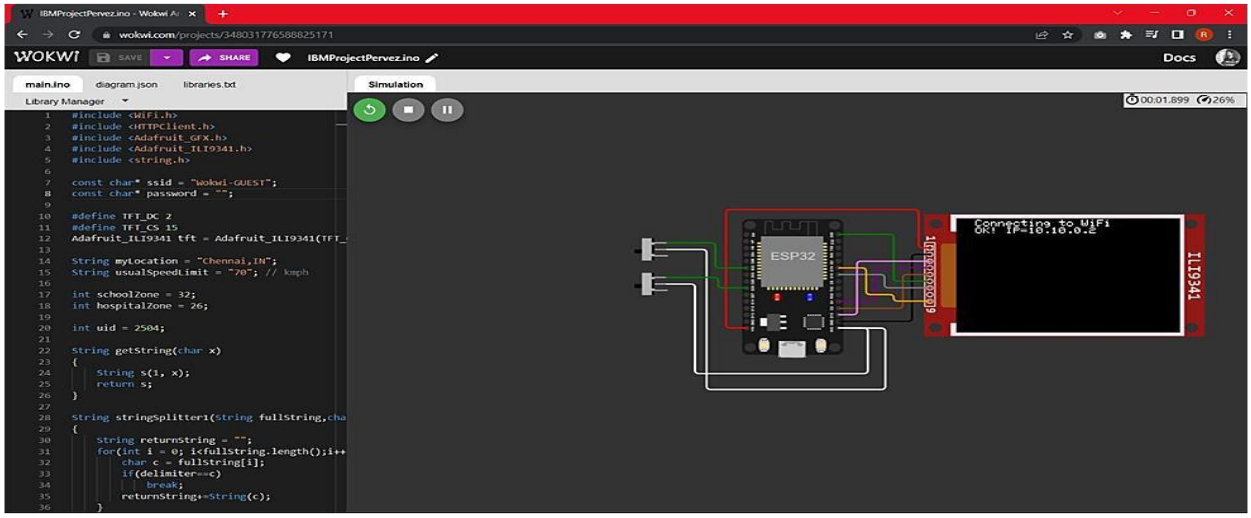
```
  tft.println(stringSplitter1(contents)
  ); String c2 =
  stringSplitter2(contents);
  if(c2=="s") // represents Straight
  { upArrow()
    ;
  }
  if(c2=="l") // represents left
  { leftArrow()
    ;
  }
  if(c2=="r") // represents right
  { rightArrow()
    ;
  }
}

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=");
```

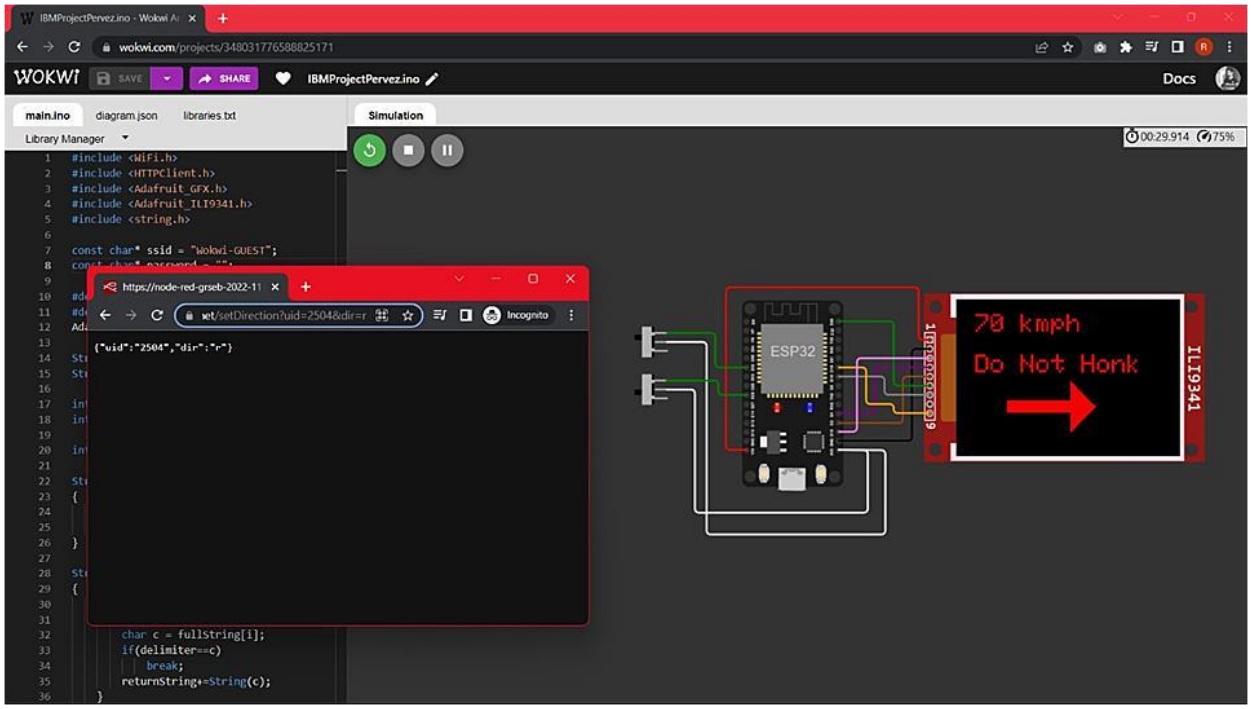
```
  tft.println(WiFi.localIP(
  ));
}

void loop()
  { myPrint(APICall(
  )); delay(100);
}
```

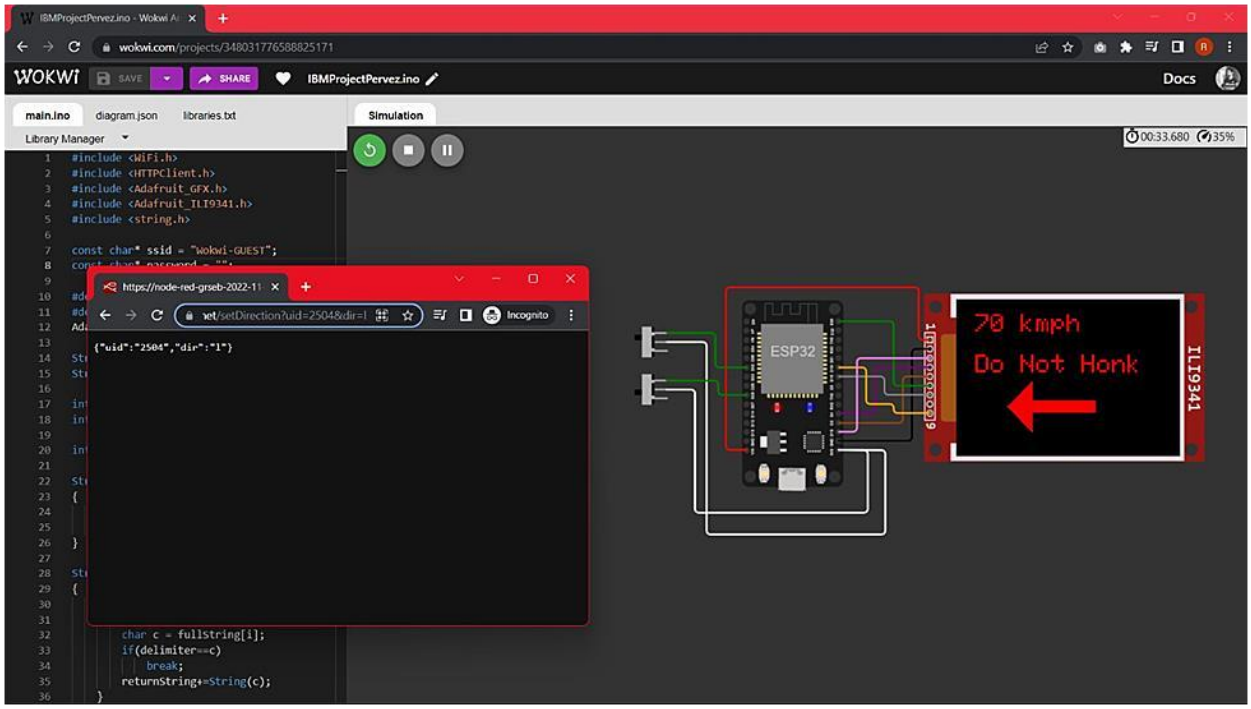
Output :



Wokwi Output :


IBMProjectPervez.ino - Wokwi Ar
wokwi.com/projects/348031776588825171
WOKWI
SAVE
SHARE
IBMProjectPervez.ino
Docs
main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:01.899
26%
#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_
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,cha
{
String returnString = "";
for(int i = 0; i<fullstring.length();i++
char c = fullstring[i];
if(delimiter==c)
break;
returnString+=String(c);
}
ESP32
Connecting to WiFi
OK! IP=10.10.0.2
ILI9341


IBMProjectPervez.ino - Wokwi Ar
wokwi.com/projects/348031776588825171
WOKWI
SAVE
SHARE
IBMProjectPervez.ino
Docs
main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:29.914
75%
https://node-red-grseb-2022-11
wet/setDirection?uid=2504&dir=r
Incognito
{"uid":"2504","dir":"r"}
ESP32
70 kmph
Do Not Honk
ILI9341


IBMProjectPervez.ino - Wokwi
wokwi.com/projects/348031776588825171
WOKWI
SAVE
SHARE
IBMProjectPervez.ino
Docs
main.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:33.680
35%
#include <WiFi.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_TLI9341.h>
#include <string.h>
const char* ssid = "Wokwi-GUEST";
https://node-red-grseb-2022-11
yet/setDirection?uid=2504&dir=1
Incognito
{"uid":"2504","dir":"1"}
char c = fullString[i];
if(delimiter==c)
break;
returnString+=String(c);
ESP32
70 kmph
Do Not Honk
ILI9341

TESTING

TEST CASE 1

Clear weather - Usual Speed

TEST CASE 2

Foggy Weather - Reduced Speed

TEST CASE 3

Rainy Weather - Further Reduced Speed Limit

TEST CASE 4

School/Hosipital Zone - Do not Honk sign is display

USER ACCEPTANCE TEST

Dynamic speed & divertion 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 everywh

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 s

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 red**
- **Longer lasting system**
- **Dynamic Sign updation**
- **School/Hospital Zone a**

DISADVANTAGES

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

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.

Git hub link :

https://github.com/IBM-EPBL/IBM-Project-20566-1659755214