BHARATH NIKETAN ENGINEERING COLLEGE, AUNDIPATTI

ELECTRONICS AND COMMUNICATION ENGINEERING

IBM NALAIYA THIRAN

TITLE : Signs with Smart Connectivity for Better Road Safety

DOMAIN : IOT

TEAM LEADER : BALAMITHRA.T

TEAM MEMBERS : 1.POOBESH.V

2.YOGESHWARI.S

INDUSTRY MENTOR NAME : Santhoshi

FACULTY MENTOR NAME : SANGEETHA.J

Project Report Format

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature

7.2 Feature

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

1.INTRODUCTION

1.1 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 when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. Intelligent transportation systems (ITS) offer significant opportunities to save live

A Road safety International task force, comprising leading international experts in road safety and connected mobility, has focused on the relation between interconnected mobility and road safety

This project proposes a system which has digital sign boards on which the signs can be changed dynamically. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards according.

1.2 Purpose

Based on current research and development efforts, we can all be certain that smart road signs will be broadly utilized in the years to come. They serve as one of the major components of an emerging system designed to enhance the current infrastructure. These indicators are useful tools, and they can have a positive impact on all who share the roads. Most importantly, this type of signage has the potential to improve our way of life.

2.LITERATURE SURVEY

2.1 Existing Problem

Problems Regarding to Traffic Signal

•Accidents:

Smart Traffic Signal

•Accidents:

In India near about 4.40.123 accidents are happened per year and most of the accidents are happened at Traffic Signal.

•Deaths:

In India near about 1, 34, 834 people are dying per year only because of road accidents and most of the accidents are causes at Traffic Signal

•At Every 3 minutes an Indian loses life just only because of road accidents

•Bribe taking one another big issue found at Traffic Signal by Traffic Police then how can we recognized the irresponsible people.

Why it is happened

• No proper standing arrangement is there for vehicles

• Carelessness of traffic police and bribery of police which is shameful thing to our Nation.

Now a day there is important responsibility on traffic controller section of government. Traffic signal system in used is control the traffic but if any vehicle is passes during red signal, then this system is not useful and as result there is chances of accidents and to avoid this it requires a traffic police at the every road of signal in these way human efforts is increase and it needs more manpower for controlling traffic in better way. Smart Traffic Signal device is essential to make the traffic signal robust, secure and safe, to reduce human efforts and to increase

the efficiency of traffic signal for detecting the vehicles which are breaking the signal. The proposed system involves detection of vehicles which are breaking traffic signal. It has central controller that is Arduino at junction which is connected to sensors are placed at every road of junction which detects the vehicles which might be inefficient when more than one peoples simultaneously go through that way but single person is allow because of particular arrangement of sensors[7],[8]. Consider if red signal is on and vehicle break signal then it will detect by sensors and this signal is sent to Arduino. The Arduino capture image of that vehicle using camera and save it as time and date format [7]. The cameras are set at every junction point is equal to the number of road meet at that junction.

CURRENT SOLUTION ON TRAFFIC PROBLEMS

• Traffic Lights are there to control the Traffic in Particular way but still there causes accidents

• Traffic police is there but there are very big chances of taking bribe from vehicle driver

2.2 REFERENCES:

Singh, R., Sharma, R., Akram, S. V., Gehlot, A., Buddhi, D., Malik, P. K., & Arya, R. (2021). Highway 4.0: Digitalization of highways for vulnerable road safety development with intelligent IoT sensors and machine learning. *Safety science*, *143*, 105407.

2.3 Problem statement definition

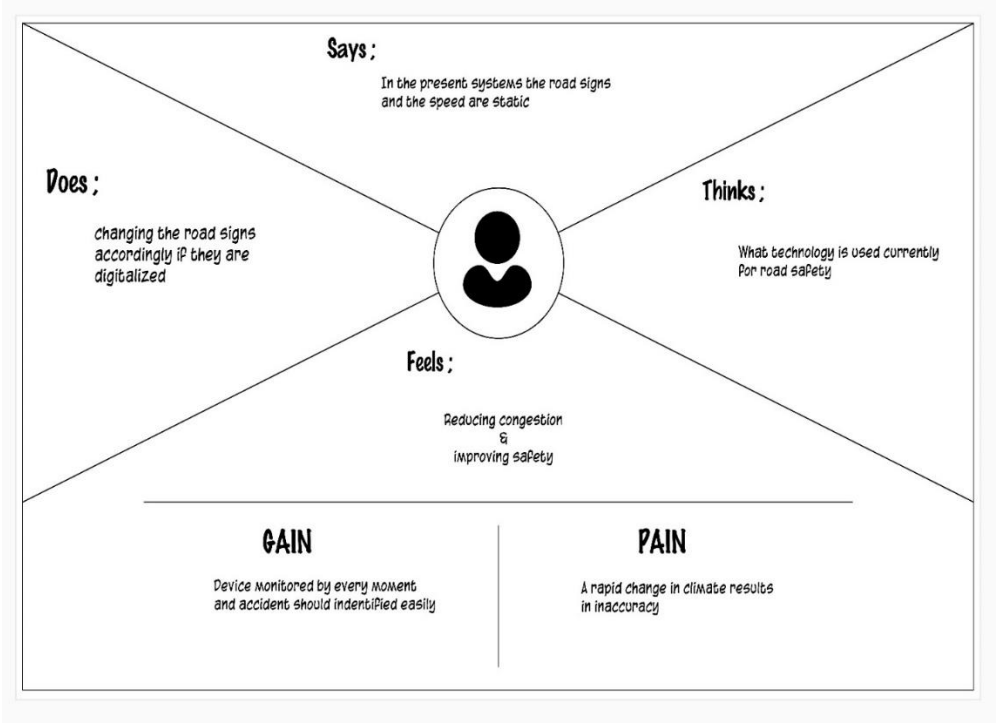
In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

At the heart of user experience (UX) design is empathy. As a UX designer, you are the biggest advocate the user has in ensuring an enjoyable, stress-free product. In order to be an effective advocate, a designer must be able to understand their user, how they think, what they need, and how they feel.

Empathy maps are an excellent starting point for developing this understanding of your users. They're also a useful tool when it comes to cultivating empathy for users with your colleagues and key stakeholders.



(Empathy Map Canvas)

3.2 Ideation & Brainstorming

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

3.3 Proposed Solution

Your proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

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 are also 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 will be 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 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 traffic will be analyzed immediately. Accordingly, the sign of the traffic signal will be changed. This inturn 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

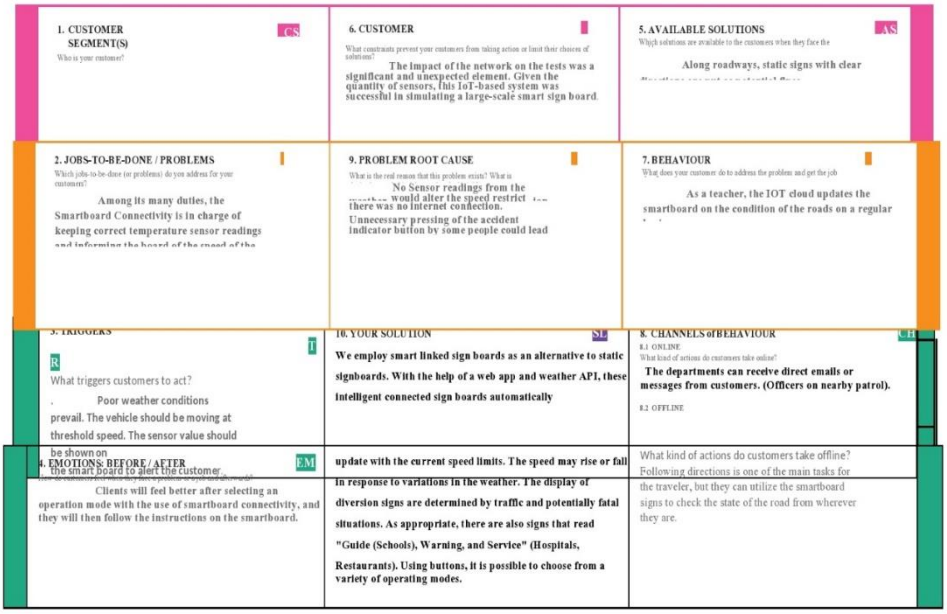
		length of time in which the customers actively interact with the product. This product is aimed to be free of cost to the public, but the revenue will be generated by selling this product to the government at a low cost, so there will be less accidents and the public will be aware of the discrepancies 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

What is a Problem-Solution Fit? The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

1. A Minimum Viable Product (MVP).
2. Satisfied early adopters (earlyvangelists) who use your MVP.
3. A validated problem that you solve for the earlyvangelists.

It varies tremendously by product, service or business, but according to Steve Blank, you need at least 3-5 satisfied paying customers for a Problem-Solution Fit.



4.REQUIREMENT ANALYZIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	Sign Boards should be made of bright coloured 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 away drivers

4.2 Non Functional Requirements

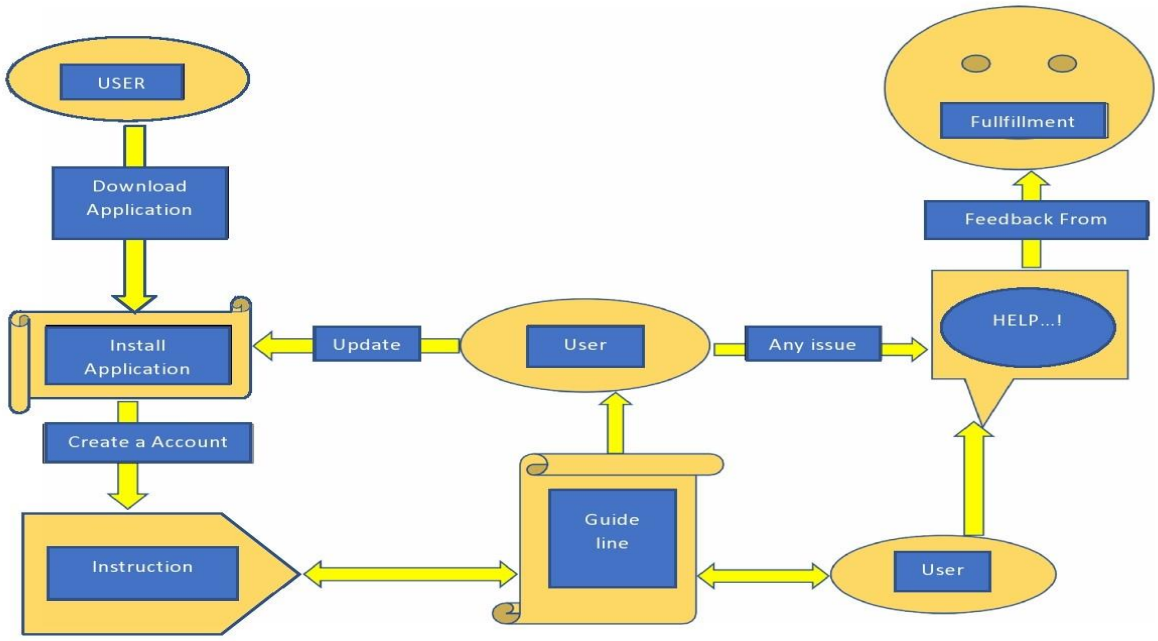
Following are the non-functional requirements of the proposed solution Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs

FR No	Non-Functional Requirement	Description
NFR-1	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 Diagrams

It's easy to understand the flow of data through systems with the right data flow diagram software. This guide provides everything you need to know about data flow diagrams, including definitions, history, and symbols and notations. You'll learn the different levels of a DFD, the difference between a logical and a physical DFD and tips for making a DFD.

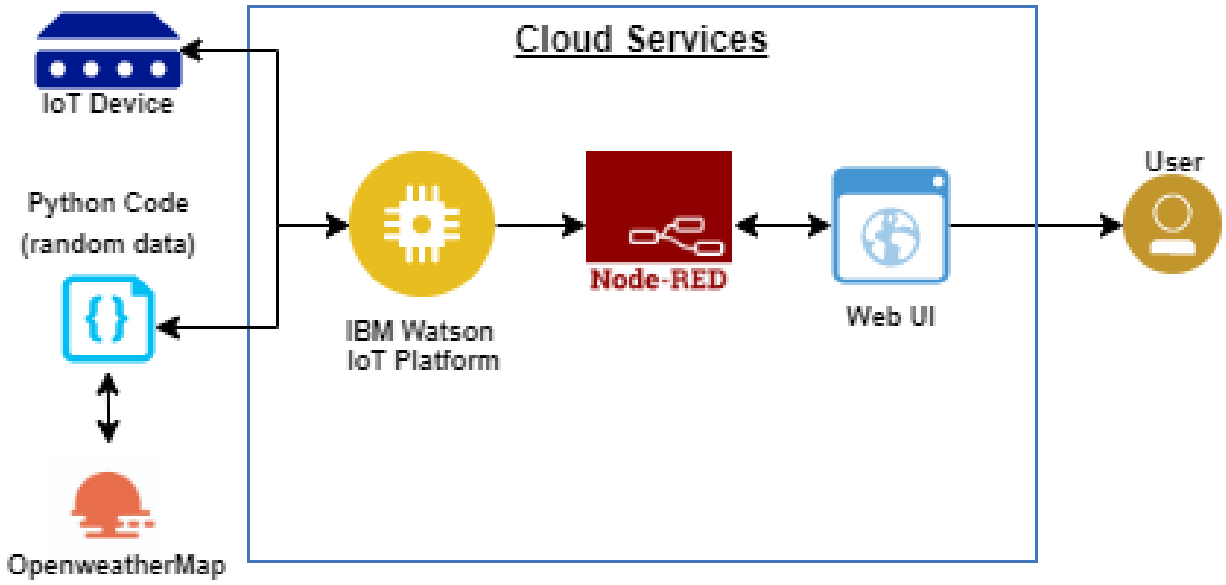


5.2 Solution & Technical Architecture

In a fast-paced digital world where new technical innovations flood the market every year, businesses need to act quickly to keep up with the competition. However, adopting random technologies without knowing how they will create more value, in the long run, can have detrimental effects on a business. In addition, cloud computing, microservices, and distributed systems are adding another level of complexity to the IT landscape. All these factors combined have created a growing need for skilled IT architects.

IT architecture aims to align the business strategy with new technological solutions and consists of various disciplines ranging from strategic to highly technical. One that fits into the latter category is technology architecture as it mainly focuses on the design and documentation of software applications. Thus, technical architects create blueprint schematics of technical solutions making sure that new products or systems meet specified requirements.

Through this focus on detail and the clear rules that are defined in technology architecture, new software solutions can be successfully delivered by developing teams and easily connect with the existing systems of a company once they're implemented. Working with technical architects ensures that developers create products that fit into the computer architecture and create the desired value for a business. It also saves companies from major headaches that are caused by system noncompliance.



Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

Throughout the past decade, architecture has become a broadly used term in the context of information technology. This doesn't come as a surprise considering how most companies had to redesign their IT landscape to adopt digital trends like cloud computing and software as service (SaaS). This digital transition required not only skilled developing teams but first and foremost IT architects.

But IT architecture encompasses a variety of different roles and disciplines that are sometimes difficult to tell apart. This is largely due to the highly dynamic nature of IT, and its widespread adoption throughout all industries and businesses that have developed their own practices. In the world of technology architecture design, the focus lies on technology limitations, meaning that a technology architect makes sure that a new application is compatible with the existing technology at a company by specifying things like the communications network or hardware that it uses.

5.3 User Stories

In software development and product management, a user story is an informal, natural language description of features of a software system. They are written from the perspective of an end user or user of a system, and may be recorded on index cards, Post-it notes, or digitally in project management software. Depending on the project, user stories may be written by different stakeholders like client, user, manager, or development team.

User stories are a type of boundary object. They facilitate sensemaking and communication; and may help software teams document their understanding of the system and its context

User Type	***Functional Requirement (Epic)***	***User Name***	***User Feedback***	***Acceptance criteria***	***Priority***	***Release***
Customer (Mobile user)	Registration	User_01	As a user, I can register for the application by entering my email, password, and confirming my password	I can access my account / dashboard	High	Sprint-1
	Registration	User_02	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email &click confirm	High	Sprint-1
Customer (Web user)	Registration	User_03	As a user, I can register for the application through Facebook	I can register & accessthe dashboard with Facebook Login	Low	Sprint-2
	Registration	User_04	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Registration	User_05	As a user, I can log into the application by entering email & password		Low	Sprint-1

6.PROJECT PLANNING & 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	BALAMITHRA T, POOBESH V, YOGESHWARI S
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location.	1	MEDIUM	BALAMITHRA T, POOBESH V, YOGESHWARI S
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	BALAMITHRA T, POOBESH V, YOGESHWARI S
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same.	2	HIGH	BALAMITHRA T, POOBESH V, YOGESHWARI S
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience.	2	LOW	BALAMITHRA T, POOBESH V, YOGESHWARI S

6.2 Sprint Delivery Scheduling

The main event during agile methodology is the sprint, the stage where ideas turn into innovation and valuable products come to life. On one hand, agile sprints can be highly effective and collaborative. At the same time, they can be chaotic and inefficient if they lack proper planning and guidance. And for this reason, making a sprint schedule is one of the most important things you can do to ensure that your efforts are successful. Keep reading to learn everything you need to know about sprint scheduling, including some tips to drive the best results.

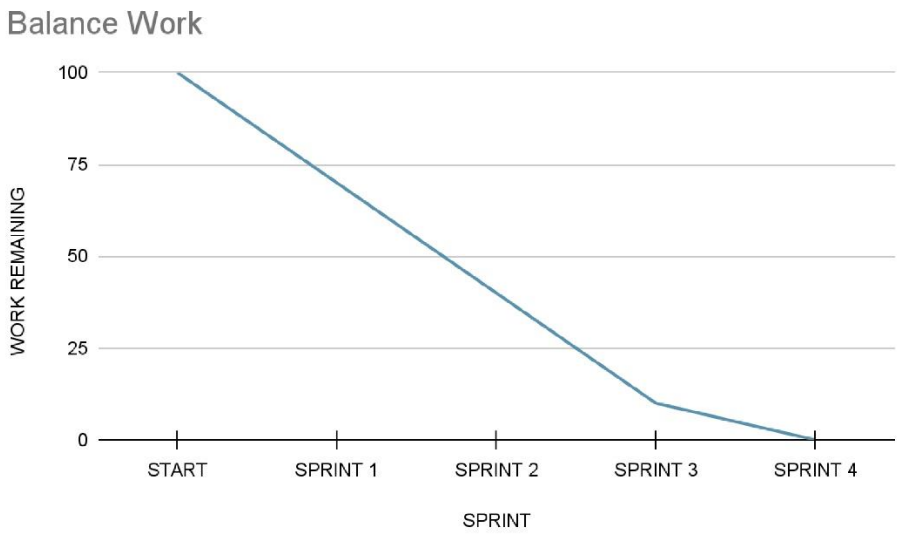
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022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

6.1 Sprint Planning & Estimation

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

In scrum, the sprint is a set period of time where all the work is done. However, before you can leap into action you have to set up the sprint. You need to decide on how long the time box is going to be, the sprint goal, and where you're going to start. The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful. Bad sprint plans can derail the team by setting unrealistic expectations.

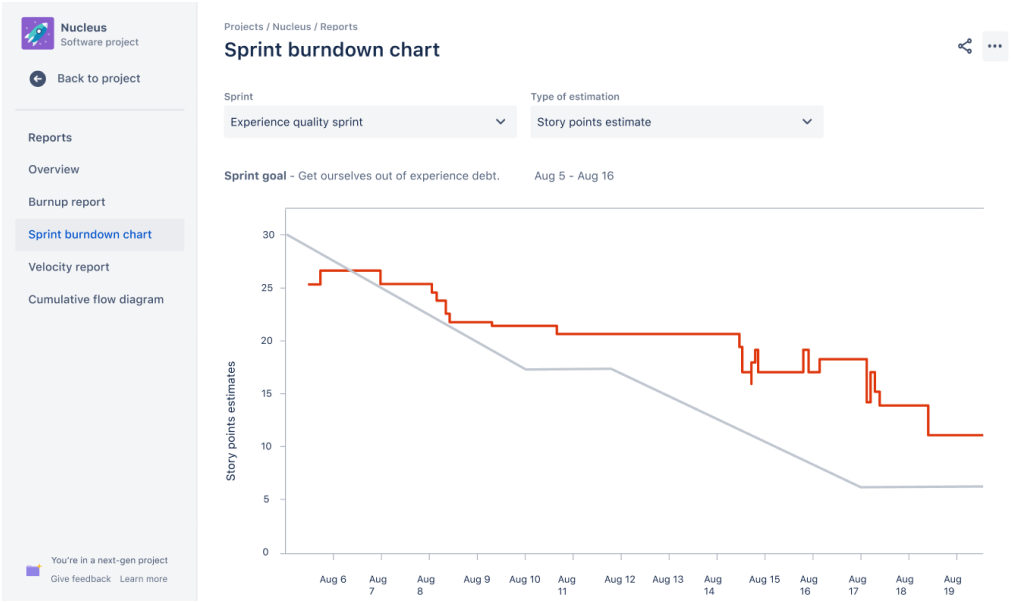
Burndown Chart:



6.3 Reporting from JIRA

Deliver value to customers faster with real-time insights at your fingertips. Jira Software enables teams to make data-driven decisions with agile reports,

One part of ensuring the success and smooth operations of your projects in JIRA is reporting. It involves gaining the knowledge about the health, progress and overall status of your JIRA projects through Gadgets, report pages or even third party applications. The goal of this guide is to provide an overview of the tools available to JIRA users today and how they can be used to fulfill the different types of reporting needs that users face today.



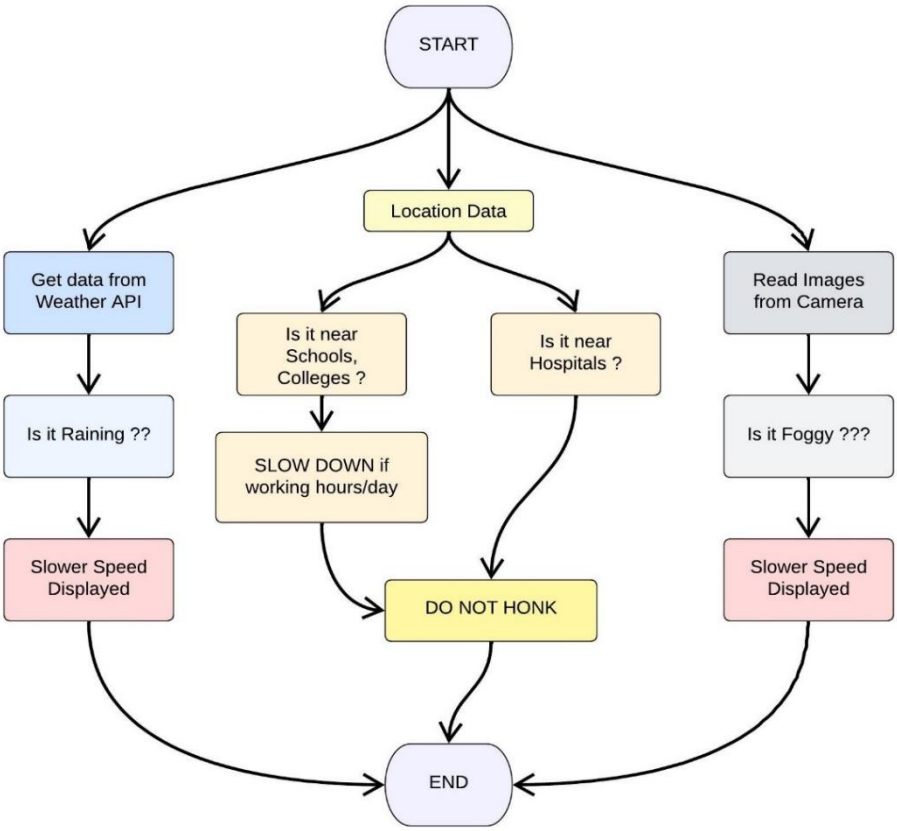
7. CODING & SOLUTIONING

7.1 FEATURE 01 :

Crash avoidance features are ones that help the driver or order avoid having a crash, such Electronic Stability Control (ESC), Autonomous Emergency Braking (AEB) and lane keep assist. Some technologies, such as AEB, can detect imminent crashes and automatically apply the brakes if the driver does not react in time to avoid or mitigate a crash.

Crash protection features are ones that help reduce the consequences of a crash such as airbags and body engineering that directs the force of a crash away from vehicle occupants**.**

Code Flow :



Python Code

```
# IMPORT SECTION STARTS
import weather
from datetime import datetime as dt
# IMPORT SECTION ENDS
# -----------------------------------------------
# UTILITY LOGIC SECTION STARTS
def processConditions(myLocation,APIKEY,localityInfo):
    weatherData = weather.get(myLocation,APIKEY)

    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else localityInfo["usualSpeedLimit"]/2
    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2

    if(localityInfo["hospitalsNearby"]):
        # hospital zone
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            # neither school nor hospital zone
            doNotHonk = False
        else:
            # school zone
            now = [dt.now().hour,dt.now().minute]
            activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["schools"]["activeTime"]]
            doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and activeTime[0][1]<=now[1]<=activeTime[1][1]

    return({
        "speed" : finalSpeed,
        "doNotHonk" : doNotHonk
    })
```

UTILITY LOGIC SECTION ENDS

> main.py

The code that runs in a forever loop in the micro-controller. This calls all the util functions from other python files and based on the return value transduces changes in the output hardware display.

Python code

```
# IMPORT SECTION STARTS


import brain


# IMPORT SECTION ENDS

# ----------------------------------------------

# USER INPUT SECTION STARTS


myLocation = "Chennai,IN"

APIKEY = "9cd610e5fd400c74212074c7ace0d62c"


localityInfo = {

   "schools" : {

      "schoolZone" : True,

      "activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:30 PM

      },

   "hospitalsNearby" : False,

   "usualSpeedLimit" : 40 # in km/hr

}


# USER INPUT SECTION ENDS

# ----------------------------------------------

# MICRO-CONTROLLER CODE STARTS
```

```
print(brain.processConditions(myLocation,APIKEY,localityInfo))


'''
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 2 AS PER OUR PLANNED SPRINT SCHEDULE
'''


# MICRO-CONTROLLER CODE ENDS
```

Output :

```
# Code Output
{'speed': 40, 'doNotHonk': False}
```

7.2 FEATURE 02:

Crash avoidance features are ones that help the driver or order avoid having a crash, such Electronic Stability Control (ESC), Autonomous Emergency Braking (AEB) and lane keep assist. Some technologies, such as AEB, can detect imminent crashes and automatically apply the brakes if the driver does not react in time to avoid or mitigate a crash.

Crash protection features are ones that help reduce the consequences of a crash such as airbags and body engineering that directs the force of a crash away from vehicle occupants**.**

Program Code :

This file is a utility function that fetches the weather from OpenWeatherAPI. It returns only certain required parameters of the API response.

```
# Python code


import requests as reqs


def get(myLocation,APIKEY):
    apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"
    responseJSON = (reqs.get(apiURL)).json()
    returnObject = {
```

```
        "temperature" : responseJSON['main']['temp'] - 273.15,
        "weather" : [responseJSON['weather'][_]['main'].lower() for _ in range(len(responseJSON['weather']))],
        "visibility" : responseJSON['visibility']/100, # visibility in percentage where 10km is 100% and 0km is 0%
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]
    return(returnObject)
```

> publishData.py

This code pushes data to the cloud and logs data. IBM Cloud is configured such that the data is displayed in the following website: CLICK TO OPEN NODE RED DASHBOARD

Python code

```
# IMPORT SECTION STARTS

import wiotp.sdk.device # python -m pip install wiotp
import time

# IMPORT SECTION ENDS
# ----------------------------------------------
# API CONFIG SECTION STARTS

myConfig = {
    "identity" : {
        "orgId" : "f59trs",
        "typeId" : "testdevice",
        "deviceId" : "device1"
    },
    "auth" : {
        "token" : "Jrwa7c8Os2Zpq)WW18"
    }
```

```
}


# API CONFIG SECTION ENDS

# ----------------------------------------------

# FUNCTIONS SECTION STARTS


def myCommandCallback(cmd):

    print("recieved cmd : ",cmd)




def logData2Cloud(location,temperature,visibility):

    client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)

    client.connect()

    client.publishEvent(eventId="status",msgFormat="json",data={

        "temperature" : temperature,

        "visibility" : visibility,

        "location" : location

    },qos=0,onPublish=None)

    client.commandCallback = myCommandCallback

    client.disconnect()

    time.sleep(1)
```

 # FUNCTIONS SECTION ENDS

This file is a utility function that returns only essential information to be displayed at the hardware side and abstracts all the unnecessary details. This is where the code flow logic is implemented.

```
# Python code


# IMPORT SECTION STARTS


import weather
```

```
from datetime import datetime as dt
from publishData import logData2Cloud as log2cloud


# IMPORT SECTION ENDS
# ----------------------------------------------
# UTILITY LOGIC SECTION STARTS
def processConditions(myLocation,APIKEY,localityInfo):
    weatherData = weather.get(myLocation,APIKEY)

    log2cloud(myLocation,weatherData["temperature"],weatherData["visibility"])

    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else localityInfo["usualSpeedLimit"]/2
    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2

    if(localityInfo["hospitalsNearby"]):
        # hospital zone
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            # neither school nor hospital zone
            doNotHonk = False
        else:
            # school zone
            now = [dt.now().hour,dt.now().minute]
            activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["schools"]["activeTime"]]
            doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and activeTime[0][1]<=now[1]<=activeTime[1][1]

    return({
        "speed" : finalSpeed,
        "doNotHonk" : doNotHonk
```

```
    })


# UTILITY LOGIC SECTION ENDS
```

The code that runs in a forever loop in the micro-controller. This calls all the util functions from other python files and based on the return value transduces changes in the output hardware display.

```
# Python code


# IMPORT SECTION STARTS


import brain


# IMPORT SECTION ENDS

# ---------------------------------------------

# USER INPUT SECTION STARTS


myLocation = "Chennai,IN"

APIKEY = "9cd610e5fd400c74212074c7ace0d62c"


localityInfo = {

    "schools" : {

        "schoolZone" : True,

        "activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:30 PM

        },

    "hospitalsNearby" : False,

    "usualSpeedLimit" : 40 # in km/hr

}


# USER INPUT SECTION ENDS

# ---------------------------------------------

# MICRO-CONTROLLER CODE STARTS
```

```
while True :
    print(brain.processConditions(myLocation,APIKEY,localityInfo))


'''
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 3 AS PER OUR PLANNED SPRINT SCHEDULE
'''


# MICRO-CONTROLLER CODE ENDS
```

Output :

LINK TO NODE RED DASHBOARD

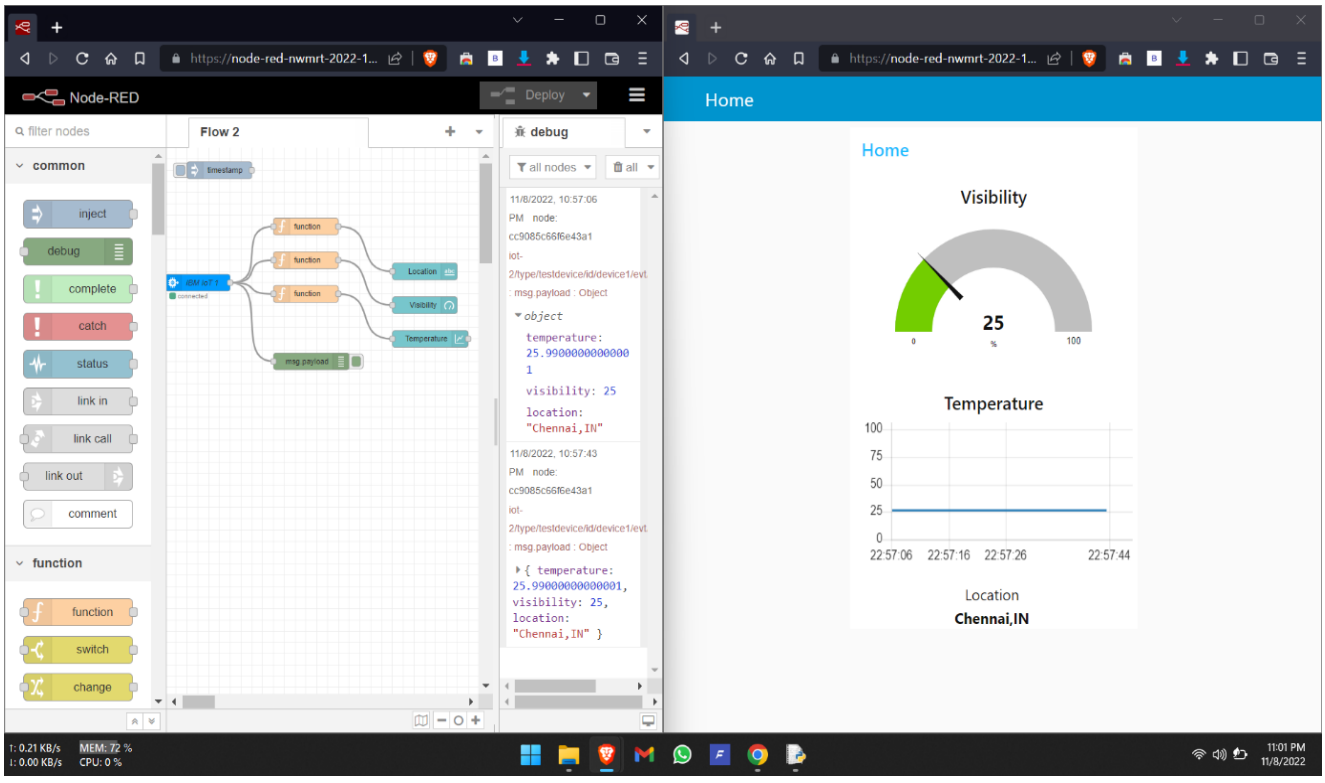
Code Output

```
22022-11-08 22:57:43,506   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:f59trs:testdevice:device1

2022-11-08 22:57:43,574   wiotp.sdk.device.client.DeviceClient  INFO    Disconnected from the IBM Watson IoT Platform

2022-11-08 22:57:43,580   wiotp.sdk.device.client.DeviceClient  INFO    Closed connection to the IBM Watson IoT Platform

{'speed': 20.0, 'doNotHonk': False}
```



. **7.3 Database Schema :**

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
WiFiClient wifiClient;
String data3;
#define DHTTYPE DHT11
#define DHTPIN 9
DHT dht(DHTPIN, DHTTYPE);
#define ORG "v6wg8x"
#define DEVICE_TYPE "nodeMcu"
#define DEVICE_ID "NodeMCU"
#define TOKEN "123456789"
#define speed 0.034
void callback(char* topic, byte* playload, unsigned int payloadLength);
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, callback , wifiClient);
void publishData();
String command;
String data = "";
long duration;
float dist;
void setup()
{
Serial.begin(115200);
dht.begin();
wifiConnect();
mqttConnect();
}
void loop() {
publishData();
delay(500);
if (!client.loop()) {
```

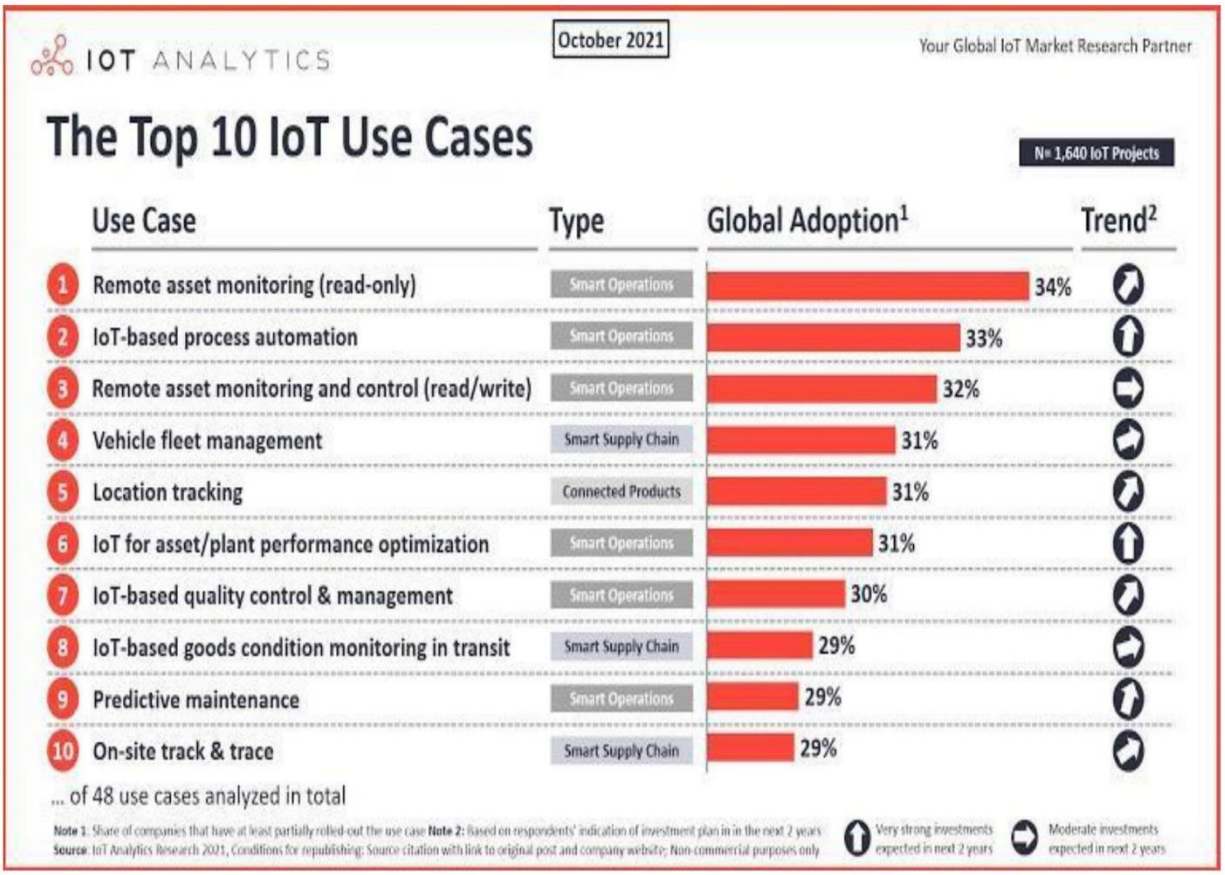
```
mqttConnect();
}
}
void wifiConnect() {
Serial.print("Connecting to "); Serial.print("Wifi");
WiFi.begin("SSID","Passord");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}
void mqttConnect() {
if (!client.connected()) {
Serial.print("Reconnecting MQTT client to "); Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice() {
if (client.subscribe(topic)) {
Serial.println("IBM subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
int sensorValue = analogRead(34); //MQT 135 connected to GPIO 34 (Analog
ADC1_CH6)
Serial.print("AirQua=");
Serial.print(sensorValue, DEC);
Serial.println(" PPM");
float humid = dht.readHumidity();
```

```
float temp = dht.readTemperature(true);
float airQty = sensorValue/4095;
String payload = "{\"Temperature\":";
payload += temp;
payload += "}";
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
payload = "{\"Air Quality\":";
payload += airQty;
payload += "%}";
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}
}
void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength) {
Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);
for (int i = 0; i < payloadLength; i++) {
dist += (char)payload[i];
}
Serial.println("data:" + data3);
if (data3 == "lighton") {
Serial.println(data3);
}
data3 = "";
}
```

8.TESTING

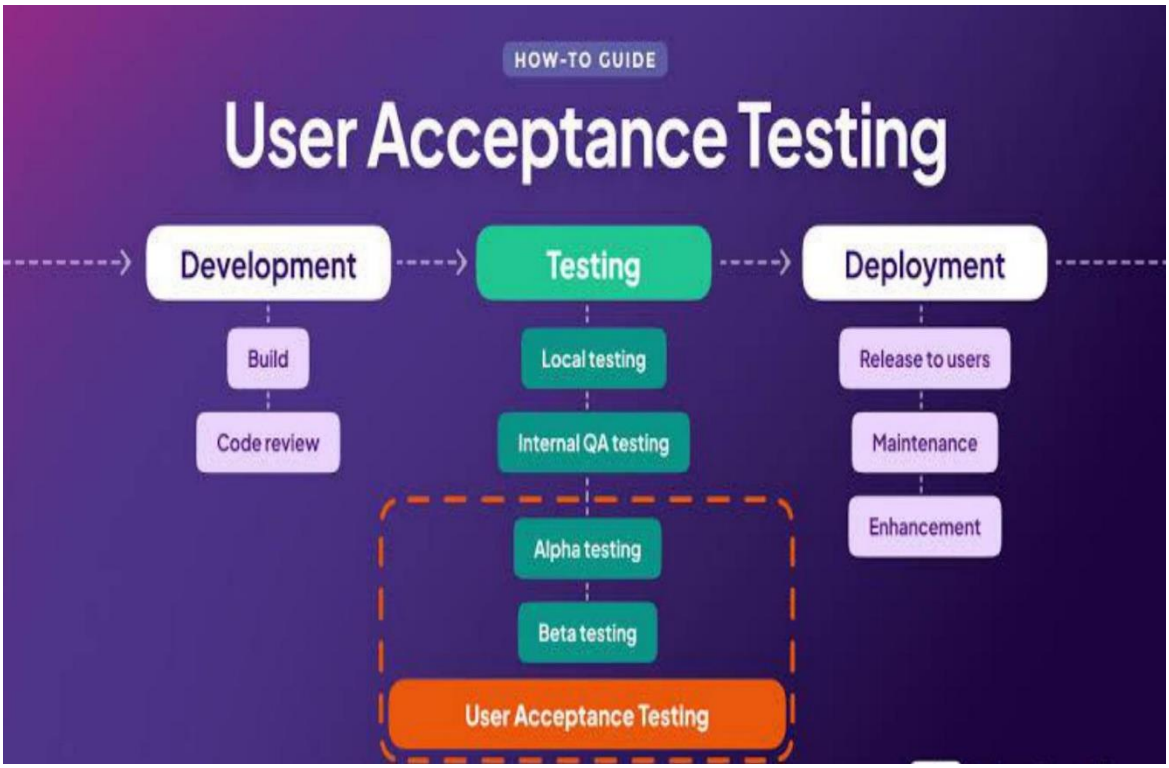
8.1 Test Cases

A **Test Case** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.



8.2 User Acceptance

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.



9.RESULTS

9.1 Performance Metrics

The analysis of road safety performance and the success of policy measures are usually focused on the number of crashes, injuries, and fatalities. Although these figures are important to monitor the level of road safety, they do not provide adequate information to understand the causes underlying road traffic crashes and do not indicate the interventions a country should focus on . For this reason, data of crashes and victims must be supplemented by other indicators that give a more complete picture of the level of traffic safety and a better understanding of the process that leads to crashes

10.ADVANTAGES & DISADVANTAGES

Advantages:

Multimodal sensors and edge computing help speed up the flow of traffic with real-time processing, reducing congestion and emissions. Smart road technology can assist in optimizing traffic flow and managing road conditions, creating a more sustainable environment within cities.

Disadvantages:

Increased traffic can increase carbon emissions and other pollution. Land use for roads can damage built and natural environment, impose mortality on wildlife if habitats are severed, and construction has associated environmental costs.

11.CONCLUTION

These conclutions and guidelines are addressed to policy makers and private companies that are willing to use innovative solutions to decrease road-related fatalities and injuries amidst populations. Both chapters take into account the potential users of connected technologies: individual drivers, commercial drivers, pedestrains, cyclists and motorcyclists. The task force decided to study first the potential of connected technologies in high-income and middle-income countries. Indeed middle-income countries represent 72% of the World population, 80% of road traffic deaths and 47% of connected vehicles.

12.FUTURE SCOPE

With new pressures for cities to develop more effective roadways and highways, smart infrastructure is essential for modernization. Smart roads built on IoT and information and communications technology (ICT) can make it possible for cities and transportation authorities to collect and analyze data to improve day-to-day traffic management. Smart road infrastructure can also help cities adapt for long-term sustainable transportation needs. With IoT sensors, cameras, radar, and 5G-equipped technologies, data can be analyzed in near-real time and used to improve congested roadways, streamlining traffic flow. Data can also be sent to the cloud for long-term analysis, providing critical insight for efforts such as reducing CO2 emissions. Edge computing opens myriad possibilities for smart and connected roads.

13.APPENDIX

Source code

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt
import cv2
import tensorflow as tf
from PIL import Image
import os
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from keras.models import Sequential, load_model
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout
data = []
labels = []
classes = 43
cur_path = os.getcwd()
#Retrieving the images and their labels
for i in range(classes):
path = os.path.join(cur_path,'train',str(i))
images = os.listdir(path)
for a in images:
try:
image = Image.open(path + '\\'+ a)
```

```
image = image.resize((30,30))
image = np.array(image)
#sim = Image.fromarray(image)
data.append(image)
labels.append(i)
except:
print("Error loading image")
#Converting lists into numpy arrays
data = np.array(data)
labels = np.array(labels)
print(data.shape, labels.shape)
#Splitting training and testing dataset
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
#Converting the labels into one hot encoding
y_train = to_categorical(y_train, 43)
y_test = to_categorical(y_test, 43)
#Building the model
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.shape[1:]))
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(43, activation='softmax'))
#Compilation of the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
epochs = 15
history = model.fit(X_train, y_train, batch_size=32, epochs=epochs, validation_data=(X_test, y_test))
model.save("my_model.h5")
#plotting graphs for accuracy
plt.figure(0)
plt.plot(history.history['accuracy'], label='training accuracy')
plt.plot(history.history['val_accuracy'], label='val accuracy')
plt.title('Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
plt.figure(1)
plt.plot(history.history['loss'], label='training loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.title('Loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
#testing accuracy on test dataset
```

```
from sklearn.metrics import accuracy_score
y_test = pd.read_csv('Test.csv')
labels = y_test["ClassId"].values
imgs = y_test["Path"].values
data=[]
for img in imgs:
image = Image.open(img)
image = image.resize((30,30))
data.append(np.array(image))
X_test=np.array(data)
pred = model.predict_classes(X_test)
#Accuracy with the test data
from sklearn.metrics import accuracy_score
print(accuracy_score(labels, pred))
model.save('traffic_classifier.h5')
```

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-29482-1660126076010