IBM – NALAIYA THIRAN PROJECT 2022

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

Batch :B2-2M4E

Team ID :PNT2022TMID26631

College :ST.JOSEPH COLLEGE OF ENGINEERING

Team Leader :HEMACHANDRAN R (212919106029)

Team Members:ANTONY AROCKIA LASAR B (212919106010)

BHUVANESHWAR K (212919106019)

HEMARJUN M (21291906031)

THEEPAN SHANKAR S (212919106089)

CONTENTS

1.INTRODUCTION

1.1 Project Overview

Connected vehicle technology aim to solve some of the biggest challenges in the transportation in the areas of safety, mobility and environment. The safety application for Intelligent Transport System (ITS) is one of the main objectives in this project. Safety application is research and industrial initiative which aim to contribute to the global advancement of automobile industry. In this project wefocus on V2V communication, once cars are connected which is able to share data with other cars on the road and which help to reduce Highway accidents. Ultimately, vehicles are connect via multiple complementary technologies of vehicle to-vehicle (V2V) and vehicle-to-infrastructure (V2I) connectivity based on Wi-Fi, GPS, Dedicated Short Range Communication (DSRC). VANETS are also considered as one of the most important Simulator for safety of intelligent transportation systems. The use of the DSRC technologies support low latency vehicle-to-vehicle (V2V) communication. Inpresent 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. 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 accordingly. Clearly, intelligent roadway placards can be a vital part of our driving experience. They enable a better way for drivers to access the information they need in real time on the roads. These signs can increase awareness of upcoming issues, which people might otherwise discover too late. They may also augment the functionality of driverless vehicles.

1.2.Purpose

A large amount of research is being carried out in the domain of accident avoidance and accident alarms by a large number of researchers and practitioners. To avoid accidents, many approaches are utilized to enhance safety. For ease of reference, the literature on accident detection and avoidance is separated into three approaches: stand-alone, cooperative, and hybrid. Stand-alone approaches use sensors, such as radar and light detection and ranging (LiDAR), for accident avoidance and detection, whereas cooperative approaches rely on V2X technology and hybrid approaches.

2.LITERATURE SURVEY

2.1.Existing Problem

The Existing road system and connectivity, emphasis on the traffic and route reckoning features which cordially provisions the user acceptability to have better connectivity management. But, this often results in nonparallel road conditions and high noise ratios through the calibrations. It reiterates various subjections in its compilation and leading to segmentation error throughout. It penetrates the various unit cases in order to subsequently manifest the output. This alternatively symbolizes the ineffectively programmed web user interface. The IOT based model of our project complies of the verdict to specify the soft zone in the path. It manually ask the user to turn off the horn, which in variably decreases the decibel level of the power output. Illustratively, it confides the work schematics of the precedent evaluation under the system and allows the user to access the terminals of the app nodes variably. IBM Cloud indefinitely helps in reviving the data sets required in web application. MIT app inventor segments the creation of the user interface.

2.2.References

1. Pooja Pawar, Suvarna Langade, Mohini Bandgar: **IOT Based digital Notice Board using Arduino ATMega 328**.
International Research Journal of Engineering and Technology(IRJET).- 2019.
•Circulates notice regularly & reduce physical efforts.
• Send message at any distant location within a second.

2.. Sandeep Chaware, Trushitha Chaware: **Proposed Algorithm for Smart Traffic Control using Ultrasonic Sensor**
.International Journal of Engineering and Advanced Technology (IJEAT).- 2019.

•The outcome of the project is to learn insights of the traffic controlling and management at the signal with the dynamically changing in timing of timer as per need.

3. KamnaSingh, Deepa Bura: **IOT distinct algorithms for the Sensor Connectivity with Comparative Study between node MCU and Arduino MCU.**
NVEO Journal– 2021

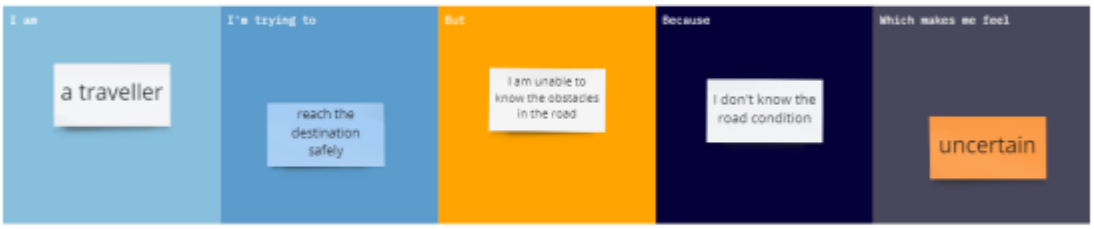
• Presents different algorithms for the connection between different types of sensors. •Brief description of node MCU & Arduino MCU. • Step by step solution to provide connectivity with IOT technology.

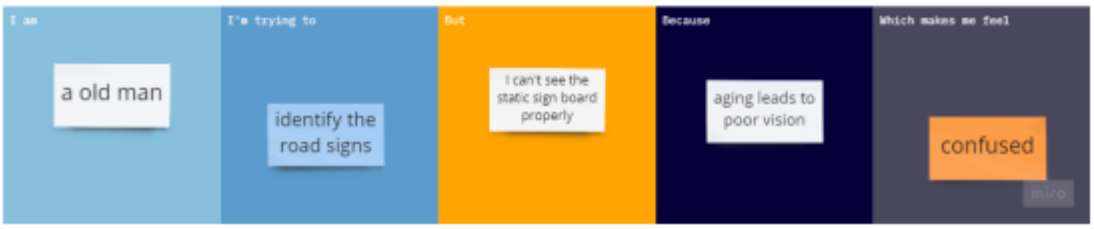
4. Jack Greenhaigh: **Recognizing Text Based Traffic Signs**. IEEE – 2015

• Detect all possible Road sign candidates. •Reduce total regions based on contextual constraints.

• A Novel System for the automatic detection and recognition of text in traffic sign based on MSER & MSV.

2.3.Problem Statement Definition

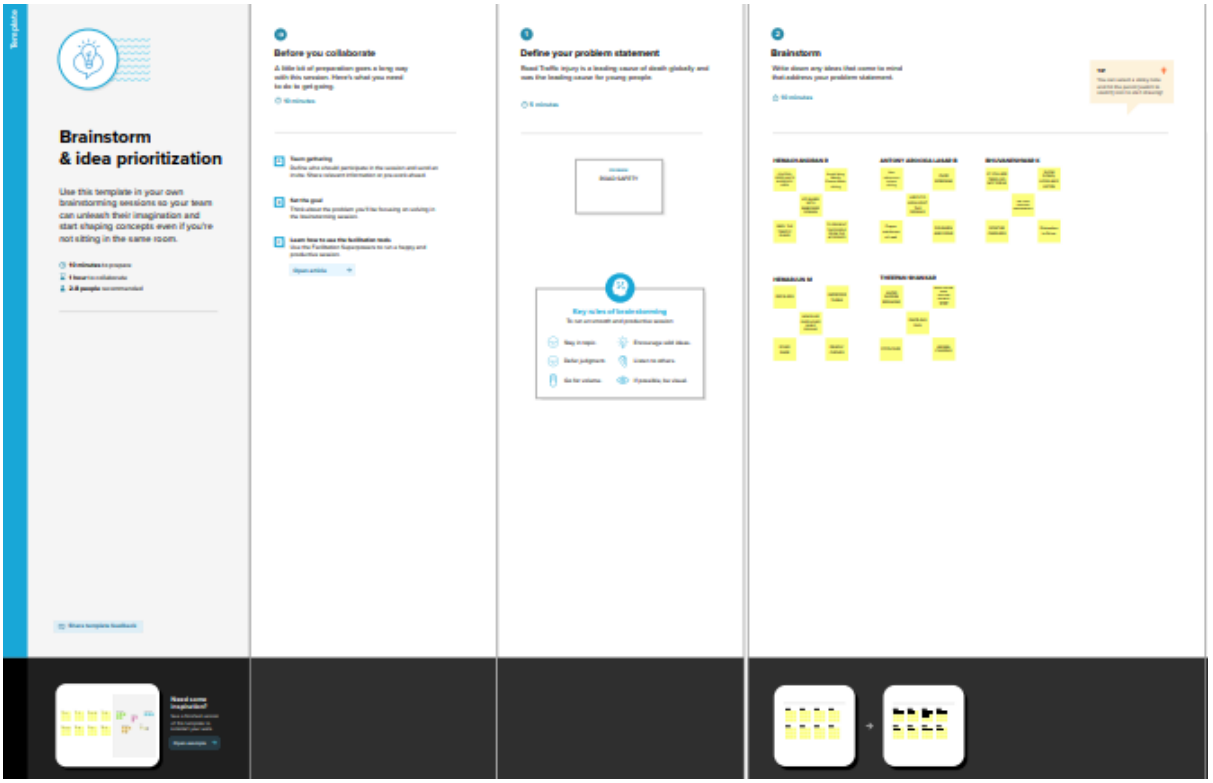


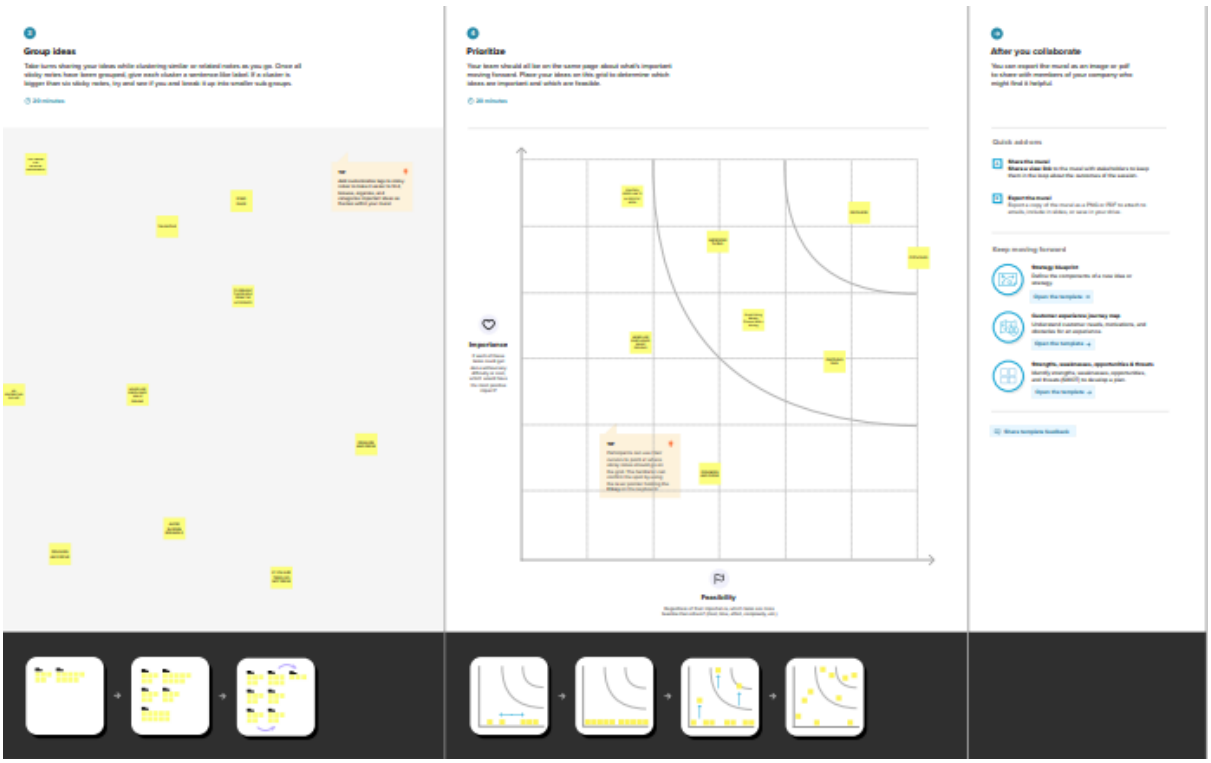


3.IDEATION & PROPOSED SOLUTION

3.1.Empathy Map Canvas

3.2.Ideation & Brainstroming



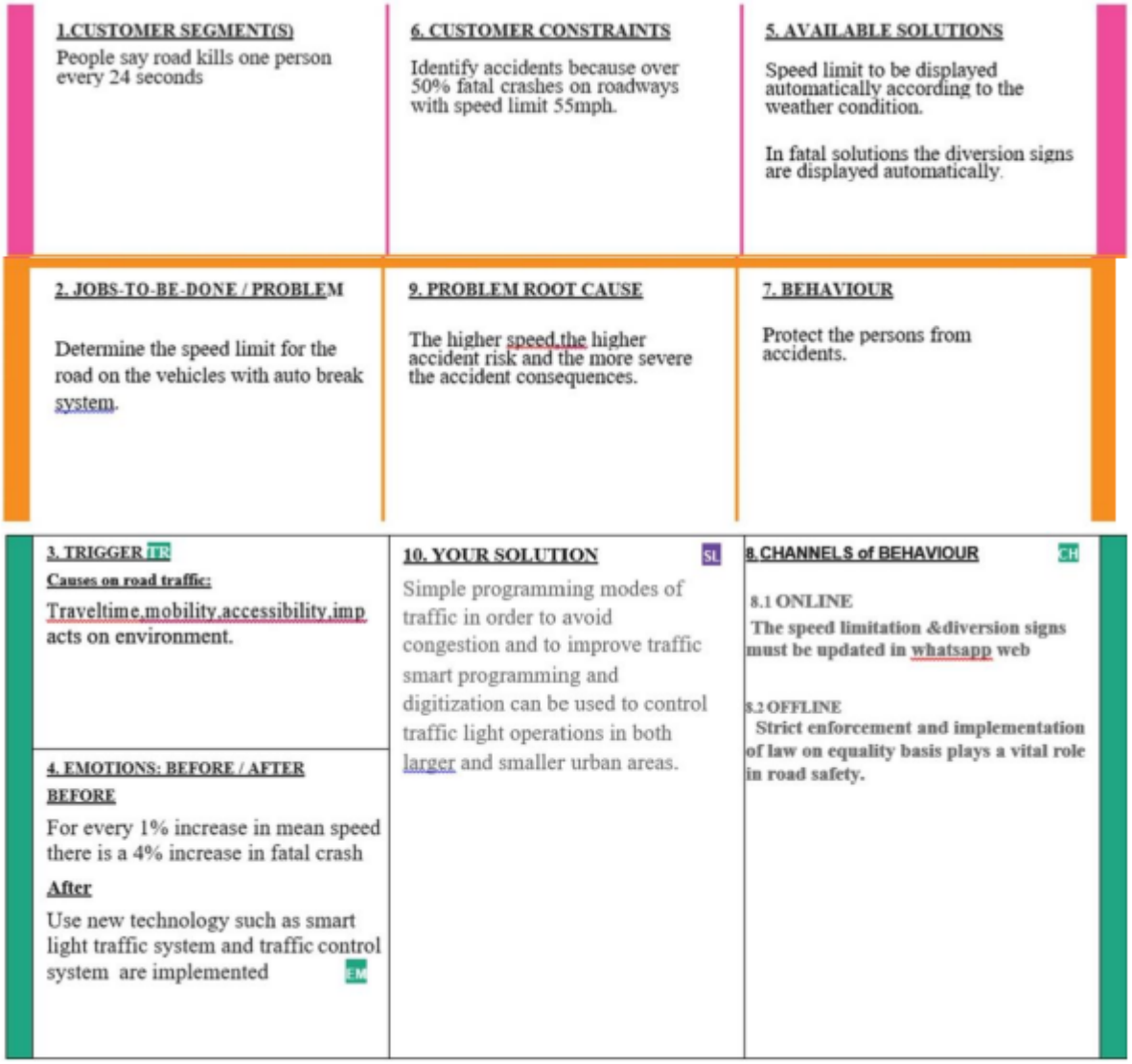


3.3.Proposed Solution

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..
2.	Idea / Solution 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
3.	Novelty / Uniqueness	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.	Social Impact / 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 (Revenue 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.
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.

3.4.Problem Solution Fit



4.REQUIREMENT ANALYSIS

4.1.Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Manual Registration Through a Website or Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Requirements	Static signboards will be replaced with smart linked sign boards that meet all criteria.
FR-4	Payments options	Bank Transfer
FR-5	Product Delivery and installation	The installation fee will be determined by the length of the road.
FR-6	Product Feedback	Through a website via Gmail

4.2.Non functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Dashboard must be simple, clean, and customizable.
NFR-2	**Security**	Using two-factor authentication and maintaining session period for some actions.
NFR-3	**Reliability**	Ensure the code is well and good before making it to production.
NFR-4	**Performance**	Writing an efficient code to give better performance to the low-end devices too.
NFR-5	**Availability**	Deploying the application in two or more Availability Zones to ensure the availability SLA of 99.999%
NFR-6	**Scalability**	Using auto-scalable services in the cloud for database, compute, etc….

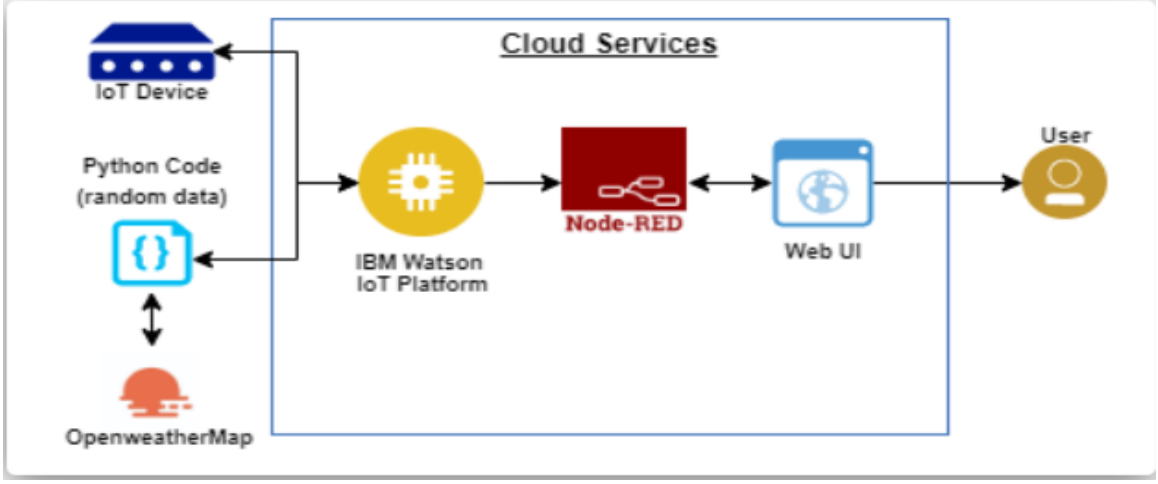
5.PROJECT DESIGN

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

5.1.Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

5.2. Solution & Technical Architecture:



5.3.User Stories

Use the below template to list all the user stories for the product

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have regToll-freeor the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by	I can access my account/	High	Spirnt-1

			entering my email, password, and confirming my password	dashboard		
Customer Care Executive	Toll free number	USN-1	As an executive I can solve people queried and complaints	I can acces account information of users	Medium	Spirnt-1
Administrator	Login	USN-1	As a administration, I can log into application and web by entering my email, password and confirming my password	I can access all data in the application, I can change Or alter	High	Spirnt-1
	Update	USN-2	As an administrator I can update the information given by the user .	I can change the data based on user given	High	Spirnt-1
	Monitor	USN-3	As an administrator I have to monitor the details given and make use of that.	I will make the monitoring needs to check the information.	Medium	Spirnt-1
	Testing	USN-3	As an administrator, testing is needed to check how reliable the application is.	I will ensure the testing process correctly and make it for user usage.	High	Spirnt-1

6. PROJECT PLANNING & SCHEDULING

The definition of a sprint is a dedicated period in which a set amount of work will be completed on a project. It's part of the agile methodology, and an Agile project will be broken down into a number of sprints, each sprint taking the project closer to completion.

6.1.Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Members.
Sprint-1	Resources Initialisation	Create and initialize accounts in various public APIs like Open Weather API.	1	Low	Hemachandran, Antony Arockia Lasar, Bhuvaneshwar, Hemarjun, Theepan Shankar
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location.	1	Medium	Hemachandran, Antony Arockia Lasar, Bhuvaneshwar, Hemarjun, Theepan Shankar
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	Hemachandran Antony Arockia Lasar, Bhuvaneshwar Hemarjun, Theepan Shankar
Sprint-3	Hardware initialisation	Integrate the hardware to be able to access the cloud functions and provide inputs to the same.	2	High	Hemachandran, Antony Arockia Lasar, Bhuvaneshwar, Hemarjun, Theepan Shankar
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience.	2	Low	Hemachandran, Antony Arockia Lasar, Bhuvaneshwar, Hemarjun, Theepan Shankar

6.2.Sprint Delivery Schedule

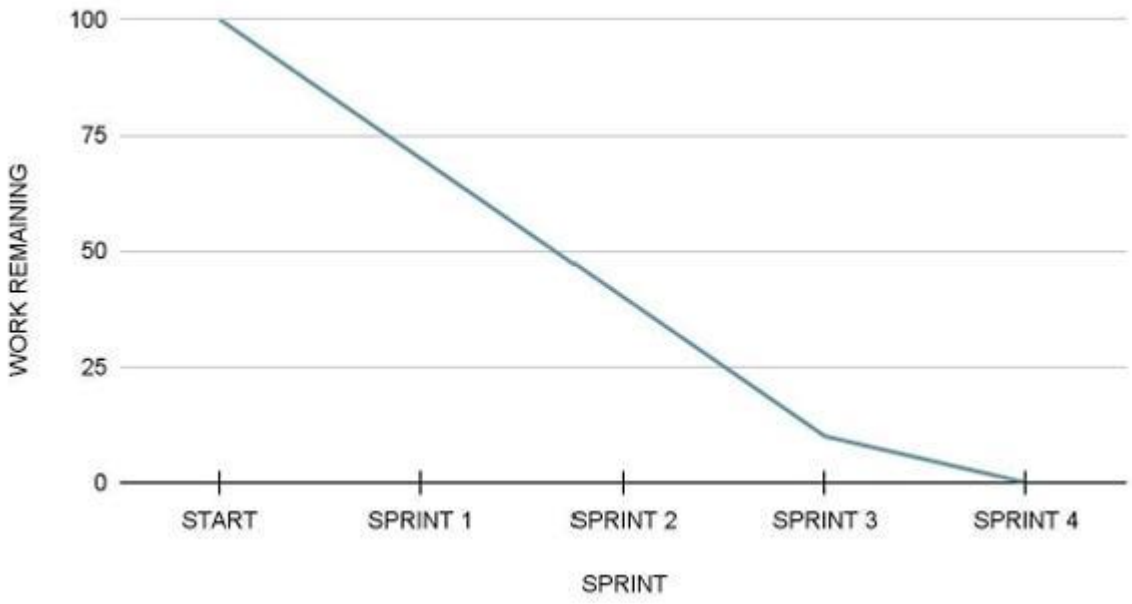
Project Tracker, Velocity & Burndown Chart

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	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

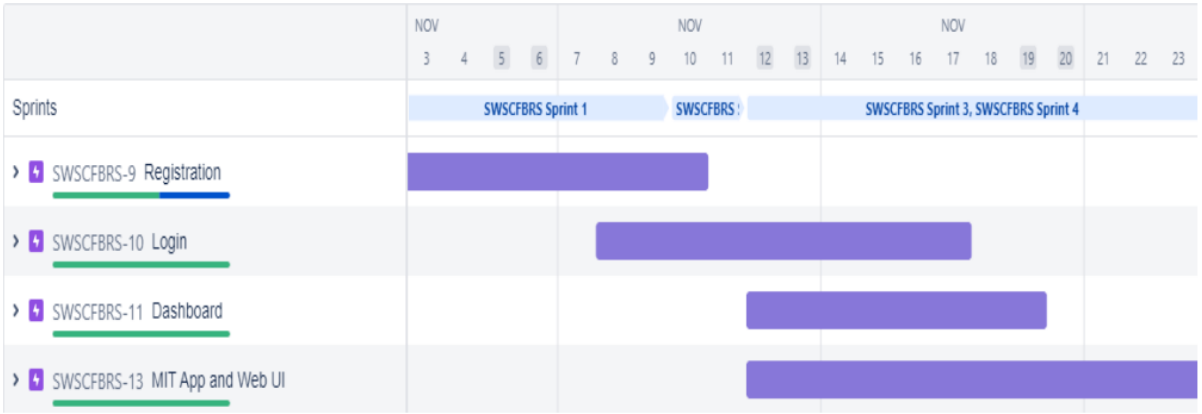
Velocity:

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

Burndown Chart:



6.3.Reports from JIRA



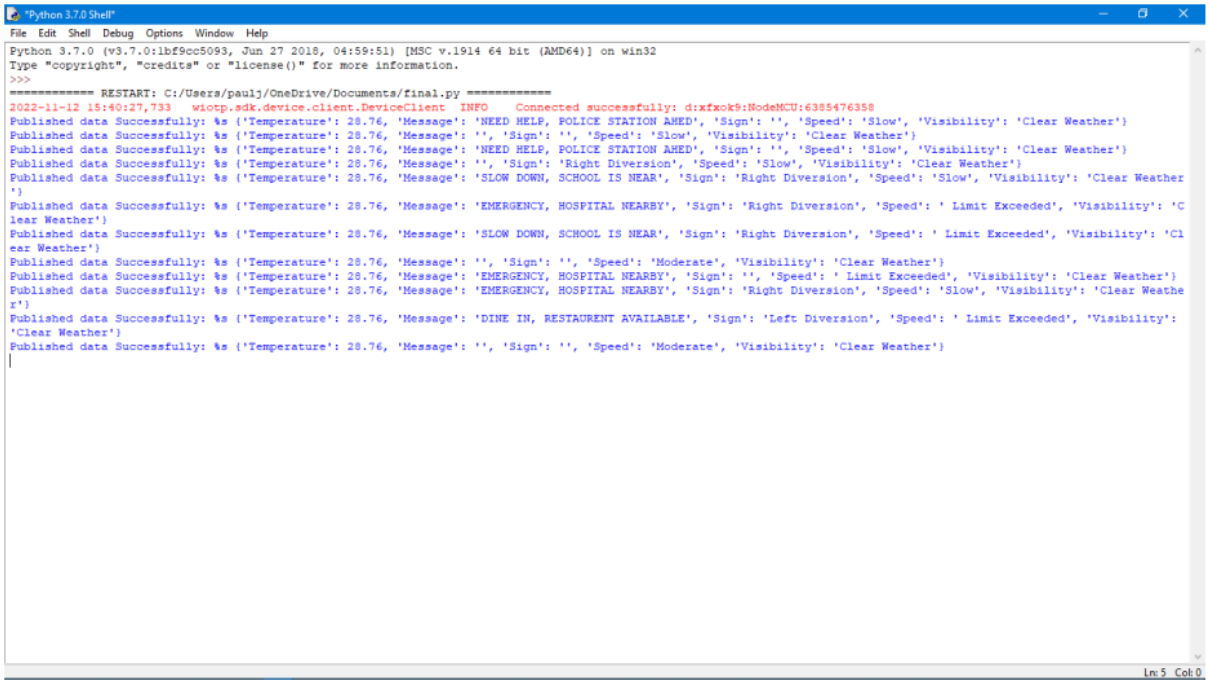
7.CODING & SOLUTIONING

7.1.Feature 1

```
import wiotp.sdk.device import time
import random
import ibmiotf.application import ibmiotf.device import requests, json

myConfig = { #Configuration "identity": {
    "orgId": "xfxok9",
    "typeId": "NodeMCU", "deviceId":"6385476358"
    },
#API Key
"Auth":{
"token": "9384731286"
}
}
#Receiving callbacks from IBM IOT platform def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']

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

#OpenWeatherMap Credentials
BASE_URL ="https://api.openweathermap.org/data/2.5/weather?" CITY =
 "Chennai"
URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" +
"01df65417ab3968e3fc2a38c4aee27bb"

while True:
response = requests.get(URL) if response.status_code == 200:
data = response.json() main = data['main'] temperature = main['temp']
humidity = main['humidity'] pressure = main['pressure'] report = data['visibility']

#messge part msg=random.randint(0,5) if msg==1:
message="SLOW DOWN, SCHOOL IS NEAR"
elif msg==2:
message="NEED HELP, POLICE STATION AHED"
elif msg==3:
```

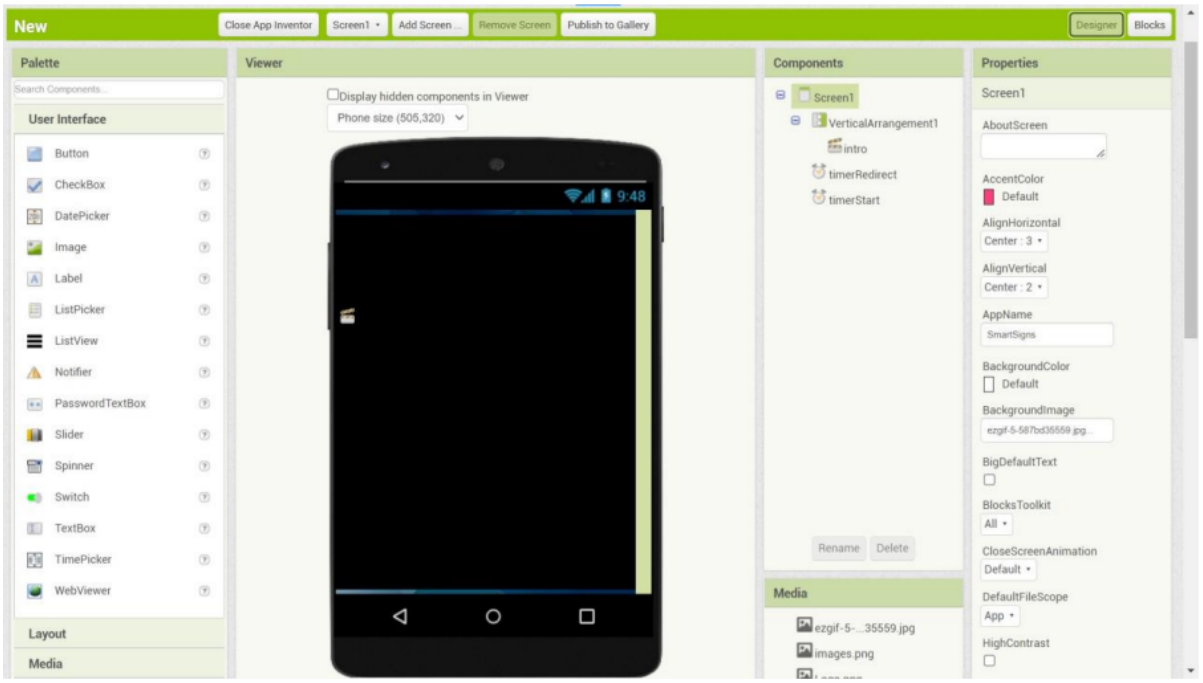
```
message="EMERGENCY, HOSPITAL NEARBY"
elif msg==4:
message="DINE IN, RESTAURENT AVAILABLE"
else:
    message="" #Speed Limit part

speed=random.randint(0,150) if speed>=100:
    speedMsg=" Limit Exceeded" elif speed>=60 and speed<100:
    speedMsg="Moderate" else:
speedMsg="Slow"

#Diversion part sign=random.randint(0,5) if sign==1:
    signMsg="Right Diversion" elif sign==3:
    signMsg="Left Diversion" elif sign==5:
    signmsg="U Turn" else:
signMsg=""
#Visibility
if temperature < 24:
    visibility="Fog Ahead, Drive Slow" elif temperature < 20:
    visibility="Bad Weather" else:
    visibility="Clear Weather"

else:
print("Error in the HTTP request")
    myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg,
'Speed':speedMsg, 'Visibility':visibility}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None) #PUBLISHING TO IOT WATSON
print("Published data Successfully: %s", myData) client.commandCallback =
myCommandCallback time.sleep(5)
client.disconnect()
```

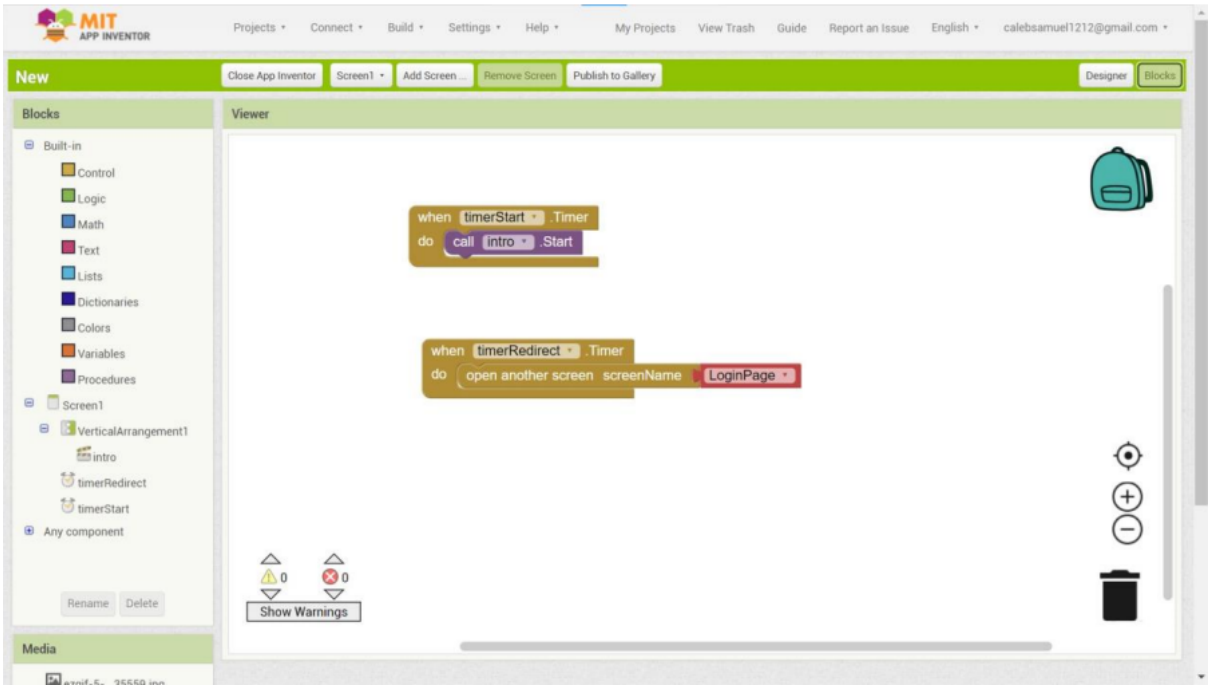
OUTPUT



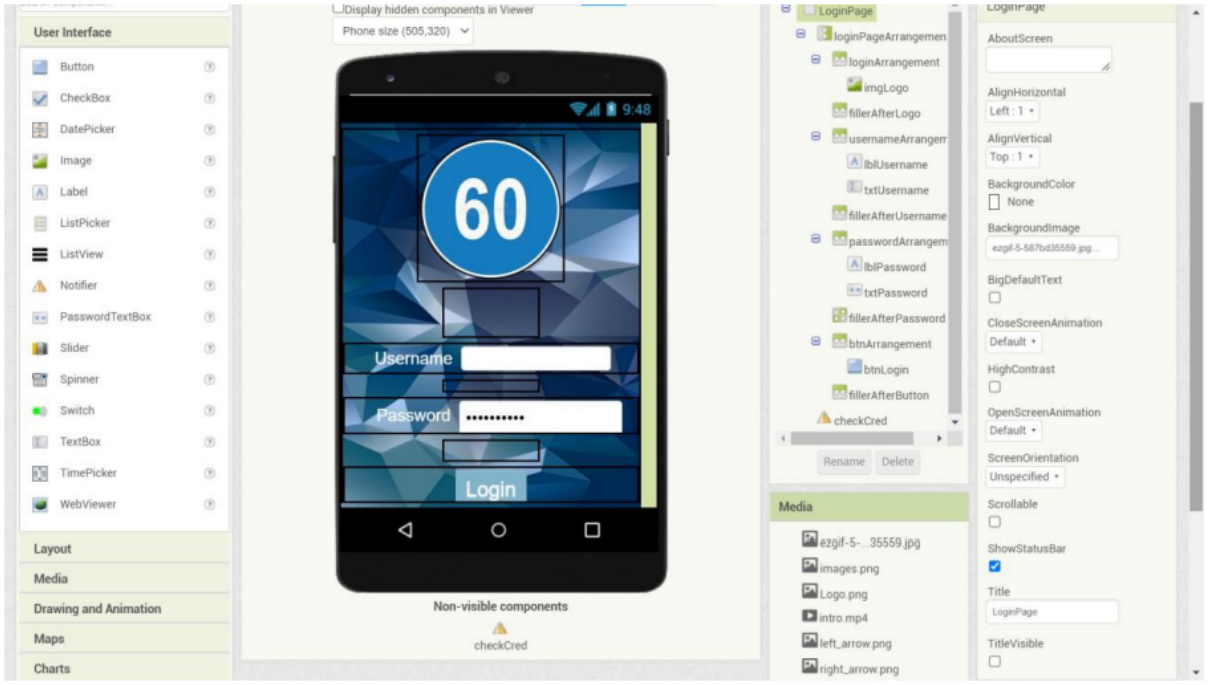
7.2.Feature 2(MIT APP INVENTER)

MIT APP INVENTOR: ICON PAGE FOR SCREEN 1



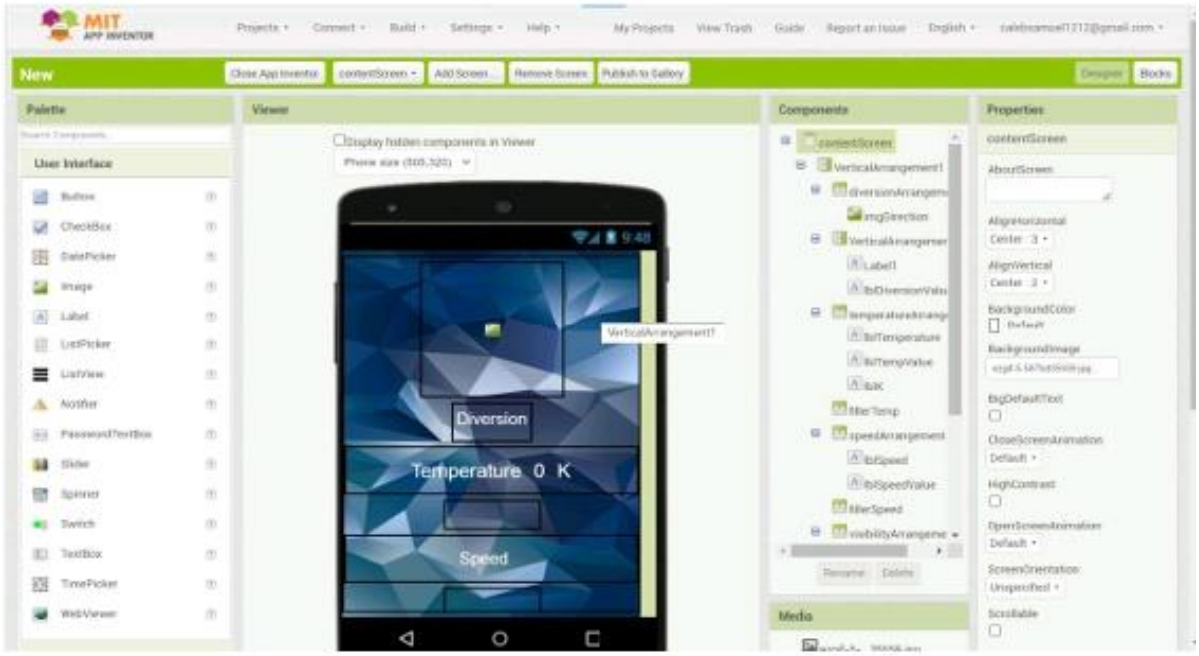

MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
calebsamuel1212@gmail.com
New
Close App Inventor
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
VerticalArrangement1
intro
timerRedirect
timerStart
Any component
Rename
Delete
Media
Viewer
when timerStart .Timer
do call intro .Start
when timerRedirect .Timer
do open another screen screenName LoginPage
Show Warnings

FOR SCREEN 2


User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Charts
Display hidden components in Viewer
Phone size (505,320)
9:48
60
Username
Password
Login
Non-visible components
checkCred
LoginPage
loginPageArrangemen
loginArrangement
imgLogo
fillerAfterLogo
usernameArrangem
lblUsername
txtUsername
fillerAfterUsername
passwordArrangem
lblPassword
txtPassword
fillerAfterPassword
btnArrangement
btnLogin
fillerAfterButton
checkCred
Rename
Delete
Media
ezgif-5-...35559.jpg
images.png
Logo.png
intro.mp4
left_arrow.png
right_arrow.png
AboutScreen
AlignHorizontal
Left : 1
AlignVertical
Top : 1
BackgroundColor
None
BackgroundImage
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar
Title
LoginPage
TitleVisible


MIT
APP INVENTOR
New
Blocks
Viewer
Show Warnings
Media

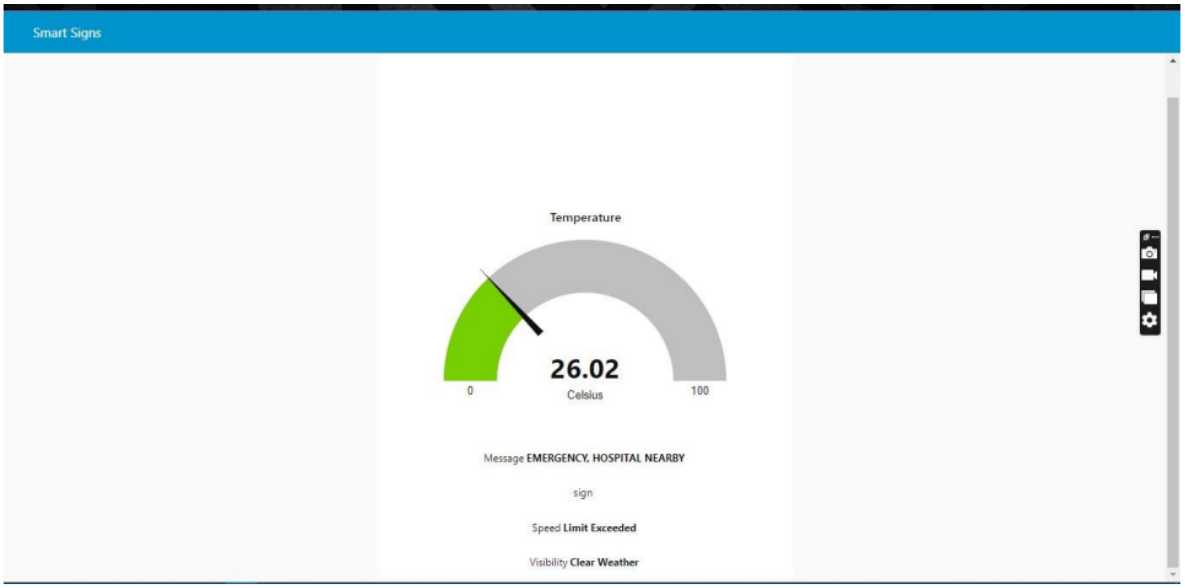
FOR SCREEN 3


MIT
APP INVENTOR
New
Palette
User Interface
Viewer
Components
Properties
Diversion
Temperature 0 K
Speed
Media

8.TESTING

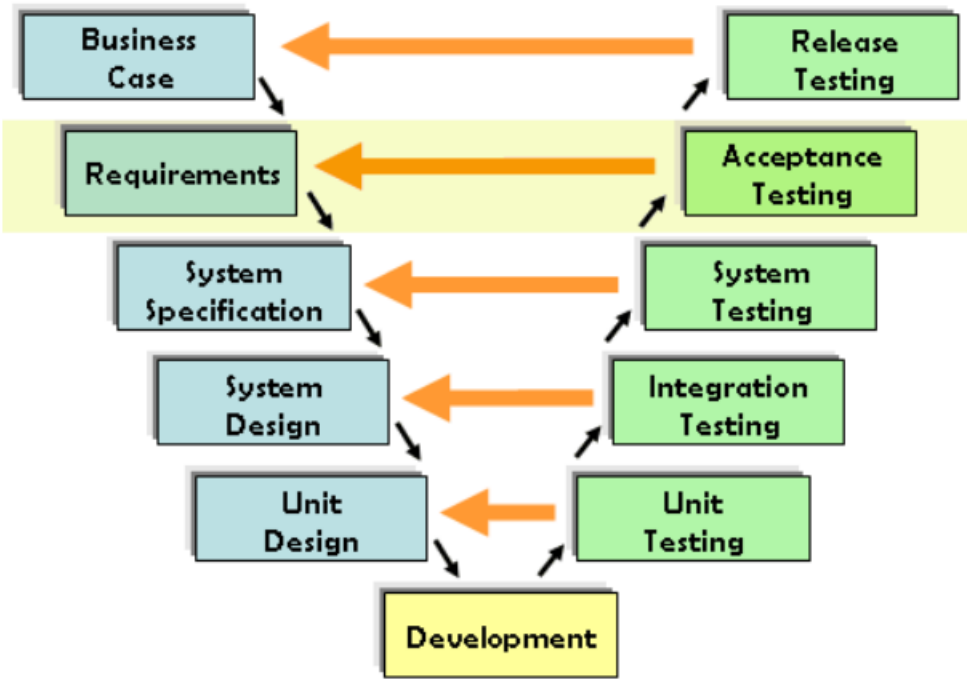
Test cases help guide the tester through a sequence of steps to validate whether a software application is free of bugs, and working as required by the end-user. Learning how to write test cases for software requires basic writing skills, attention to detail, and a good understanding of the application under test (AUT).

8.1.Test Cases



8.2.User Acceptance Testing

UAT consists, in practice, of people from the target audience using the application. The defects they find are then reported and fixed. This scenario is what most closely resembles "the real world." The process allows users to "get their hands dirty" with the application. They can see if things work as intended.



The main purpose of UAT is to validate end-to-end business flow. It does not focus on cosmetic errors, spelling mistakes, or system testing. User Acceptance Testing is carried out in a separate testing environment with a production-like data setup. It is a kind of black box testing where two or more end-users will be involved.

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, NodeRED is capable of handling upto 10,000 requests per second. Moreover, since the system is horizontally scalable, a 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.
*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
* It will manage road conditions, creating a more sustainable environment within cities.

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.
*Security and privacy. Keeping the data gathered and transmitted by IoT devices safe is challenging, as they evolve and expand in use
*Technical complexity.

11.CONCLUSION

The world doesn’t change on its own but we humans can change the world to be safe, better, and harmless. Since the road isn’t said to be safe let’s make it safer with the technologies present and available to us. The Internet of Things is one of the technologies that can lead us to travel on enhanced safe roads. So let's come together to create a better world with no accidents and a smart road for the future generation.

12.FUTURE SCOPE

IoT obtains the majority of its data with the help of connected cars. These incorporate a large number of sensors that establish communication with the cloud, other vehicles, and devices. Thanks to this it provides data and information of great utility for the improvement of road safety. The safe system approach to road safety emphasizes safety by desigh ensuring safe vehicles, road networks, and road users. Evolving towards the future, the road needs to boil with advanced sensors and antenna systems to have peace with the new era.

13.APPENDIX

Source Code

```
import wiotp.sdk.device import time
import random
import ibmiotf.application import ibmiotf.device import requests, json

myConfig = { #Configuration "identity": {
    "orgId": "xfxok9",
    "typeId": "NodeMCU", "deviceId":"6385476358"
},
 #API Key
"auth": {
"token": "9384731286"
}
}
#Receiving callbacks from IBM IOT platform def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']

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

#OpenWeatherMap Credentials
BASE_URL ="https://api.openweathermap.org/data/2.5/weather?" CITY = "Chennai"
URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" +
"01df65417ab3968e3fc2a38c4aee27bb"

while True:
response = requests.get(URL) if response.status_code == 200:
data = response.json() main = data['main'] temperature = main['temp']
humidity = main['humidity'] pressure = main['pressure'] report = data['visibility']
```

```
#messge part msg=random.randint(0,5) if msg==1:
message="SLOW DOWN, SCHOOL IS NEAR"
elif msg==2:
message="NEED HELP, POLICE STATION AHED"
elif msg==3:
message="EMERGENCY, HOSPITAL NEARBY"
elif msg==4:
message="DINE IN, RESTAURENT AVAILABLE"
else: message=""

#Speed Limit part speed=random.randint(0,150) if speed>=100:
  speedMsg=" Limit Exceeded" elif speed>=60 and speed<100:
speedMsg="Moderate" else:
speedMsg="Slow"

#Diversion part sign=random.randint(0,5) if sign==1:
    signMsg="Right Diversion" elif sign==3:
    signMsg="Left Diversion" elif sign==5:
    signmsg="U Turn" else:
signMsg=""

#Visibility
if temperature < 24:
visibility="Fog Ahead, Drive Slow" elif temperature < 20:
visibility="Bad Weather" else:
visibility="Clear Weather"
else:
print("Error in the HTTP request")
    myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg,
'Speed':speedMsg, 'Visibility':visibility}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None) #PUBLISHING TO IOT WATSON
print("Published data Successfully: %s", myData) client.commandCallback =
myCommandCallback time.sleep(5)
client.disconnect()
```

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-28234-166

Demo link

https://youtube.com/watch?v=I19tESbx-E8&feature=share