

Final Deliverables Project Report

Signs with Smart Connectivity for Better Road Safety

Date	19 November 2022
Team ID	PNT2022TMID41539
Project Name	Signs with Smart Connectivity for Better Road Safety
Team Lead	Mugila.R
Team Members	1. Ishwariya.P 2. Kalpana.T 3. Shreein Fathima.S
College	Idhaya Engineering College for Women – Chinnasalem Affiliated to Anna University
Degree	B.E. – Electronics and Communication Engineering
Year/Semester	Final year/7th Semester

Contents

S.No.	Title	Page No.
1	INTRODUCTION 1.1 Project Overview 1.2 Purpose	3
2	LITERATURE SURVEY 2.1 Existing problem 2.2 References 2.3 Problem Statement Definition	5
3	IDEATION & PROPOSED SOLUTION 3.1 Empathy Map Canvas 3.2 Ideation & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution fit	11
4	REQUIREMENT ANALYSIS 4.1 Functional requirement 4.2 Non-Functional requirements	16
5	PROJECT DESIGN 5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories	17
6	PROJECT PLANNING & SCHEDULING 6.1 Sprint Planning & Estimation 6.2 Sprint Delivery Schedule 6.3 Reports from JIRA	20
7	CODING & SOLUTIONING (Explain the features added in the project along with code) 7.1 Feature 1 7.2 Feature 2 7.3 Database Schema (if Applicable)	28
8	TESTING 8.1 Test Cases 8.2 User Acceptance Testing	34
9	RESULTS Performance Metrics	36
10	ADVANTAGES & DISADVANTAGES	37
11	CONCLUSION	38
12	FUTURE SCOPE	38
13	APPENDIX Source Code GitHub Project Demo Link	38

1. Introduction:

Roads are the foremost source of linking between cities and villages. Due to the ease of traveling by road, vehicles have become the main way people travel. The chances of vehicular accidents (Vas) have increased with the growing number of vehicles on the roads. During a journey, one does not know what will happen on the next road, particularly during bad weather conditions (BWC). In such a situation, driving can be difficult due to bad visibility, which can lead to an accident. It was also noticed that in BWC, multiple vehicle collisions (MVCs) can occur owing to delays in receiving information about an incident. According to one study by the Islamabad police, there were 9582 accidents from 2016 to 2017 all over Pakistan, involving 11,317 vehicles, leading to 5047 fatalities and 12,696 persons injured

Digital technologies like the Internet of Things (IoT) are reshaping road safety measures. Many technology initiatives are undertaken the world over to make smarter and safer roads, the ones that can interact with traffic and pedestrians. Assuming that by giving in vehicle technology information to the driver, accidents can be averted, several technology-based products have been developed. The latest technology researchers are working on is based on the Internet of Things (IoT). IoT is all about data. Data is becoming a valuable resource for our world.

Many sectors and industries have adopted IoT to reduce errors and improve performance in manufacturing, energy, health care, and communication. The WHO describes different measures that can be implemented with minimal economic impacts in its “Save LIVES: Road Safety Technical Package”. A cornerstone of these steps is realizing economic systems for “monitoring road safety by strengthening data systems”. Meanwhile, a key theme in the package is motivating the adoption of a Safe System approach, which is a holistic approach to road safety that parts from traditional management solutions by emphasizing safety by design.

Mobile-phone-based applications use built-in sensor data to detect the speed limit based on environmental situations.

The main contributions of this research are

1. A brief survey on the state of the art related to pre-accident as well as post-accident models, frameworks, and techniques;
2. Identification and reporting of limitations in previous studies related to accident detection;
3. The concept of a smart road with an event-sensing capability, plus implementation and testing through various experiments;
4. Demonstration of a new and modern way to quickly detect accident and communicate with nearby vehicles and EOCs.

The risks for loss of life, injuries, and other damage may increase if an incident is not reported to an EOC in a timely fashion. Lives can be saved by sending timely information about an accident through an automated mechanism. Moreover, quick automobile accident detection and an alert system are required to protect approaching vehicles against an MVC. Several methods have been implemented in advanced vehicles (AVs) for avoiding an accident. An accident threat is detected through sensors installed in vehicles or by using smartphone sensors. Previous researchers have used accelerometers, smoke detectors, infrared (IR) obstacle sensors, proximity sensors, and biosensors to detect an accident.

1.1. Project Overview:

The main aim of this project is to help people automate the roads by providing them with a Web App through which they can monitor the parameters of the road like temperature, speed limit, and visibility of the road. They also show guides for schools and provide services of displaying hospitals, and restaurant signs accordingly.

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:

Abstract:

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 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 that has digital signboards 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 on road diversions, accident-prone areas, and information sign boards can be entered through the web app. This data is retrieved and displayed on the signboards accordingly.

Introduction:

An automated deep learning (DL)-based system was developed for detecting accidents from video data. The system uses visual components in temporal order to represent traffic collisions. As a result, the model architecture is composed of a visual-features-extraction phase followed by transient pattern identification. Convolution and recurrent layers are used in the training phase to learn visual and temporal features. In public traffic accident datasets, an accuracy of 98% was attained in the detection of accidents, demonstrating a strong capacity for detection independent of the road structure. The solution is limited to automobile crashes, not motorbikes, bicycles, and pedestrians. Furthermore, the model makes mistakes when determining accident segments under poor illumination (e.g., at night), at low resolutions, and when there are occlusions.

An accident management system was proposed in that makes use of cellular technology in public transportation. This method enables communication across various components, including those in ambulances, RSUs, and servers. Furthermore, in this system, an optimal route-planning algorithm (ORPA) is proposed to optimize aggregate spatial utilization of road networks while lowering the travel cost to operate a vehicle. The ORPA was evaluated through simulations, and findings were compared with other current algorithms. In congested areas, the proposed method can also be used to offer fast routes for ambulances. All vehicles, including ambulances, are required to have a route indicator installed, as well as the ability to use remote correspondence. The ORPA outperformed in terms of average speed and travel duration, according to the evaluation data. The proposed system only works for predicted patterns and can fail due to the unpredicted behavior of traffic.

Existing problem:

The Safe System Approach

The Safe System (SS) approach to transport networks originated with the “Safe Road Transport System” model developed by the Swedish Transport Agency. In its essence, the approach migrates from the view that accidents are largely and

automatically the driver's fault to a view that identifies and evaluates the true causes of accidents. Through the categorization of safety into the safety of three elements (vehicle, road, and road user), SS minimizes fatalities and injuries by controlling speeds and facilitating prompt emergency response. The model has been widely adopted since its introduction and is currently motivated by the WHO as a basis for road safety planning, policy-making, and enforcement

REFERENCE:

1. World Health Organization, "Global status report on road safety 2015," https://www.who.int/violence_injury_prevention/road_safety_status/2015/en/. View at: [Google Scholar](#)
2. World Health Organization, "Decade of Action for Road Safety 2011-2020 seeks to save millions of lives," http://www.who.int/roadsafety/decade_of_action/en/. View at: [Publisher Site](#) | [Google Scholar](#)
3. F. Wegman, "The future of road safety: A worldwide perspective," *IATSS Research*, vol. 40, no. 2, pp. 66–71, 2017. View at: [Publisher Site](#) | [Google Scholar](#)
4. World Health Organization, "Save LIVES-A road safety technical package," 2017. View at: [Google Scholar](#)
5. W. E. Marshall, "Understanding international road safety disparities: Why is Australia so much safer than the United States?" *Accident Analysis & Prevention*, vol. 111, pp. 251–265, 2018. View at: [Publisher Site](#) | [Google Scholar](#)
6. "Open Street Maps, with New York County highlighted," <https://www.openstreetmap.org/relation/2552485>. View at: [Google Scholar](#)

7. United States Census Bureau, “TIGER/Line® Shapefiles: Roads,”<https://www.census.gov/cgi-bin/geo/shapefiles/index.php?year=2018&layergroup=Roads>.
View at: [Google Scholar](#)
8. X. Wang, X. Wu, M. Abdel-Aty, and P. J. Tremont, “Investigation of road network features and safety performance,” *Accident Analysis & Prevention*, vol. 56, pp. 22–31, 2013.
View at: [Publisher Site](#) | [Google Scholar](#)
9. European Road Assessment Program (EuroRAP), “European Road Safety Atlas,”<http://atlas.eurorap.org/>.
View at: [Google Scholar](#)
10. European Road Assessment Programme (EuroRAP), “Star Ratings,”
<http://www.eurorap.org/protocols/star-ratings/>.
View at: [Google Scholar](#)
11. International Road Assessment Programme (iRAP), “iRAP,”<https://www.irap.org/>.
View at: [Google Scholar](#)
12. H. M. Hassan and H. Al-Faleh, “Exploring the risk factors associated with the size and severity of roadway crash in Riyadh,” *Journal of Safety Research*, vol. 47, pp. 67–74, 2013.
View at: [Publisher Site](#) | [Google Scholar](#)
13. E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, “Internet-of-things-based smart environments: State of the art, taxonomy, and open research challenges,” *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 10–16, 2016.
View at: [Publisher Site](#) | [Google Scholar](#)
14. Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani,

- “Internet-of-Things-Based Smart Cities: Recent Advances and Challenges,”
IEEE Communications Magazine, vol. 55, no. 9, pp. 16–24, 2017. View at:
[Publisher Site](#) | [Google Scholar](#)
15. AARON, “GPS Logger including Gyro/Tilt/Compass & Accelerometer,”
<https://www.aaronia.com/products/spectrum-analyzers/gps-logger/>.
 View at: [Google Scholar](#)
 16. M. Farsi, K. Ratcliff, and M. Barbosa, “Overview of controller area network,”
Computing and Control Engineering Journal, vol. 10, no. 3, pp. 113–120, 1999.
 View at: [Publisher Site](#) | [Google Scholar](#)
 17. MUNIC (company website), <https://www.munic.io/>.
 18. Torque Pro (OBD 2 & Car), “Torque Pro (OBD2 & Car) Google Play Page,”
<http://goo.gl/CWD5VT>.
 View at: [Google Scholar](#)
 19. J.-S. Zhou, S.-H. Chen, W.-D. Tsay, and M.-C. Lai, “The implementation of
 OBD-II vehicle diagnosis system integrated with cloud computation
 technology,” in *Proceedings of the 2013 2nd International Conference on
 Robot, Vision and Signal Processing, RVSP 2013*, pp. 9–12, Japan, December
 2013.
 View at: [Google Scholar](#)
 20. A. Mednis, A. Elsts, and L. Selavo, “Embedded solution for road condition
 monitoring using vehicular sensor networks,” in *Proceedings of the 2012 6th
 International Conference on Application of Information and Communication
 Technologies, AICT 2012*, Georgia, October 2012.
 View at: [Google Scholar](#)
 21. M. Yamada, K. Ueda, I. Horiba, and N. Sugie, “Discrimination of the
 Road Condition Toward Understanding of Vehicle Driving
 Environments,” *IEEE Transactions on Intelligent Transportation
 Systems*, vol. 2, no. 1, pp. 26–31, 2001.

View at: [Publisher Site](#) | [Google Scholar](#)

22. J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, “The pothole patrol: using a mobile sensor network for road surface monitoring,” in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 29–39, Breckenridge, Colo, USA, June 2008.

View at: [Publisher Site](#) | [Google Scholar](#)

23. M. Valentino, C. Quiligotti, and L. Carone, “Branchial cleft cyst,” *Journal of Ultrasound*, vol. 16, no. 1, pp. 17–20, 2013.

View at: [Publisher Site](#) | [Google Scholar](#)

Problem Statement Definition

A problem statement is a concise description of an issue to be addressed or a condition to be improved upon. It identifies the gap between the current (problem) state and desired (goal) state of a process or product. Focusing on the facts, the problem statement should be designed to address the **Five Ws**. The first condition of solving a problem is understanding the problem, which can be done by way of a problem statement.

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 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 that has digital signboards 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 on road diversions, accident-prone areas, and information sign boards can be entered through the web app. This data is retrieved and displayed on the signboards accordingly.

3. IDEATION & PROPOSED SOLUTION:

Empathy MapCanvas:



Ideation & Brainstorming:

Brainstorm & idea prioritization

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.

- 45 minutes to prepare
- 1 hour to collaborate
- 24 people recommended

Before you collaborate

A list of preparation gives a long way with this session. Hence, what you need to do to get going.

- 1. **Get going**
Define who should participate in the session and send an invite. Share relevant information to provide context.
- 2. **Get the goal**
Think about the problem you're looking to solve during the brainstorming session.
- 3. **Learn how to use the facilitator tools**
Use the Facilitator's Guide to learn how to use the tools and make the session successful.

Open action

Define your problem statement

What problem are you trying to solve? Frame your problem as a clear, specific statement. This will be the focus of your brainstorm.

1. **Key insight and design**
A smart, witty sign board improves the look, up board with smart connected sign board with speed indicators.

2. **Key values of brainstorming**
To use a smart and productive session

- 1. Day to topic
- 2. Grouping will drive
- 3. Make judgment
- 4. Create ideas
- 5. Set a volume
- 6. Possible to build

Brainstorm

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

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

2. **Group ideas**
Two teams sharing your ideas with the clustering similar or related ideas as you go. Once all sticky notes have been proposed, give each team a category like idea. If a cluster is bigger than six sticky notes, try and use it you and break it up into smaller sub-groups.

3. **Testing and accuracy**
Make sure that the sign boards are reliable. Ensure that the efficiency is high. Test the model against various data. Check the algorithm with rigorous tests.

4. **Designing algorithms**
Design the algorithm for the given problem. Deploy classification algorithm when required. Make pseudo codes for the given algorithm.

5. **Collection of data**
Research about all the land scapes available. Discard the false data. Analyse the data and verify the info. Decide upon the user inputs and provide solution.

6. **User's end**
Make sure that smart boards are not clumsy. Ensure that the sign boards are user friendly. Display the video tutorial on how the boards work. Ease of reliability and accessibility to be maintained throughout.

Prioritize

You have shared all the ideas on the same page about which important moving forward. Then you share the ideas to determine which ideas are important and which are feasible.

7. **After you collaborate**
You can expect the result as an image or pdf. To share with members of your company who might find it helpful.

8. **Quick actions**
1. **Share the result**
Share the result as an image or pdf. To share with members of your company who might find it helpful.
2. **Share the result**
Share the result as an image or pdf. To share with members of your company who might find it helpful.

9. **Keep moving forward**
1. **Bring the result**
Bring the result as an image or pdf. To share with members of your company who might find it helpful.
2. **Customer experience journey map**
Understand customer needs, motivations, and obstacles for experiences.
3. **Strengthen, weaknesses, opportunities & threats**
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

Brainstorm & idea prioritization

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.

- 45 minutes to prepare
- 1 hour to collaborate
- 24 people recommended

Before you collaborate

A list of preparation gives a long way with this session. Hence, what you need to do to get going.

- 1. **Get going**
Define who should participate in the session and send an invite. Share relevant information to provide context.
- 2. **Get the goal**
Think about the problem you're looking to solve during the brainstorming session.
- 3. **Learn how to use the facilitator tools**
Use the Facilitator's Guide to learn how to use the tools and make the session successful.

Open action

Define your problem statement

What problem are you trying to solve? Frame your problem as a clear, specific statement. This will be the focus of your brainstorm.

1. **Key insight and design**
A smart, witty sign board improves the look, up board with smart connected sign board with speed indicators.

2. **Key values of brainstorming**
To use a smart and productive session

- 1. Day to topic
- 2. Grouping will drive
- 3. Make judgment
- 4. Create ideas
- 5. Set a volume
- 6. Possible to build

Brainstorm

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

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

2. **Group ideas**
Two teams sharing your ideas with the clustering similar or related ideas as you go. Once all sticky notes have been proposed, give each team a category like idea. If a cluster is bigger than six sticky notes, try and use it you and break it up into smaller sub-groups.

3. **Testing and accuracy**
Make sure that the sign boards are reliable. Ensure that the efficiency is high. Test the model against various data. Check the algorithm with rigorous tests.

4. **Designing algorithms**
Design the algorithm for the given problem. Deploy classification algorithm when required. Make pseudo codes for the given algorithm.

5. **Collection of data**
Research about all the land scapes available. Discard the false data. Analyse the data and verify the info. Decide upon the user inputs and provide solution.

6. **User's end**
Make sure that smart boards are not clumsy. Ensure that the sign boards are user friendly. Display the video tutorial on how the boards work. Ease of reliability and accessibility to be maintained throughout.

Prioritize

You have shared all the ideas on the same page about which important moving forward. Then you share the ideas to determine which ideas are important and which are feasible.

7. **After you collaborate**
You can expect the result as an image or pdf. To share with members of your company who might find it helpful.

8. **Quick actions**
1. **Share the result**
Share the result as an image or pdf. To share with members of your company who might find it helpful.
2. **Share the result**
Share the result as an image or pdf. To share with members of your company who might find it helpful.

9. **Keep moving forward**
1. **Bring the result**
Bring the result as an image or pdf. To share with members of your company who might find it helpful.
2. **Customer experience journey map**
Understand customer needs, motivations, and obstacles for experiences.
3. **Strengthen, weaknesses, opportunities & threats**
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

Proposed Solution:

The project team shall fill in the following information in the proposed solution template.

S.No.	<u>Parameter</u>	<u>Description</u>
1.	Problem Statement (Problem to be solved)	<p>To replace the static signboards, smart connected sign boards are used.</p> <p>These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.</p> <p>Based on the weather changes the speed may increase or decrease</p> <p>Based on the traffic and fatal situations the diversion signs are displayed.</p> <p>Guide(Schools), Warning and Service(Hospitals, Restaurant) signs are also displayed accordingly.</p> <p>Different modes of operations can be selected with the help of buttons.</p>
2.	Idea / Solution description	<p>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.</p>
3.	Novelty / Uniqueness	<p>Generic Sign board for all applications that uses both buttons and web service for updation</p> <p>Pedestrians are given the access to request the sign change of the signal to cross the road</p>
4.	Social Impact / Customer Satisfaction	<p>Diversion reasons will be displayed</p> <p>If there is no traffic, pedestrians can cross the street without waiting.</p> <p>Customer can reach the destination before the expected time</p>
5.	Business Model (Revenue Model)	<p>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</p>

		<p>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</p> <p>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.</p>
6.	Scalability of the Solution	<p>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</p> <p>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.</p>

Problem Solutionfit:

Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids <ul style="list-style-type: none"> Public who uses transport Officers who maintain and regulate road safety 	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. <ul style="list-style-type: none"> Public who have/use automobiles (any type) Government Transport The vehicle should have digitally supported sensors which can be compatible with the smart sign boards 	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking <ul style="list-style-type: none"> Already available analog road safety signs which are ineffective Signs painted on walls and roads by the corporation which disappeared or perished in a period of time Non reliable and rigid road safety signs which gets damaged during natural disaster or calamity 	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <ul style="list-style-type: none"> Hard to maintain data accuracy Choosing the position of placing the smart sign board Possibility of malfunction of sensors placed in the smart sign boards Damage of the sign boards due to external/internal factors 	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations. <ul style="list-style-type: none"> Position of static sign boards is not visible and is inappropriate There is no way to predict the weather in the desired destination through the static boards 	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) <ul style="list-style-type: none"> As public use various routes for travelling, prediction of the desired routes should be on hand with the weather there Static boards are not reliable to find weather in the destination 	
3. TRIGGERS TR What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. <ul style="list-style-type: none"> People want to make their travel easier and comfortable Public are aware of the traffic situations around them 4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design. <ul style="list-style-type: none"> Public are not aware of the static sign boards, so the smart sign boards are introduced which is more attractive People get more info about the needful resources in the route 	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. <ul style="list-style-type: none"> Connect the smart sign boards to access the applications provided by them such as speed limitations and weather predictions 	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 <ul style="list-style-type: none"> Video tutorial is made to educate the public about the smart sign board Online influencers can advertise the smart sign boards through their influencing medium 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <ul style="list-style-type: none"> Traffic law maker should give awareness programs to the public 	Extract online & offline CH of BE	

Define CS, fit into CC

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	<u>User Visibility</u>	Sign Boards should be made with LED's which are bright 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.
FR-2	<u>User Need</u>	The smart sign boards should be placed frequently in places it is needed and less in places where it is not needed much to avoid confusion for the user during travel.
FR-3	<u>User Understanding</u>	For better understanding of the driver, the signs should be big, clear and legible and it can also include illustrations which will make it easily understandable to the driver.
FR-4	<u>User Convenience</u>	The display should be big enough that it should even be visible from far distance clearly.

Non-Functional requirements:

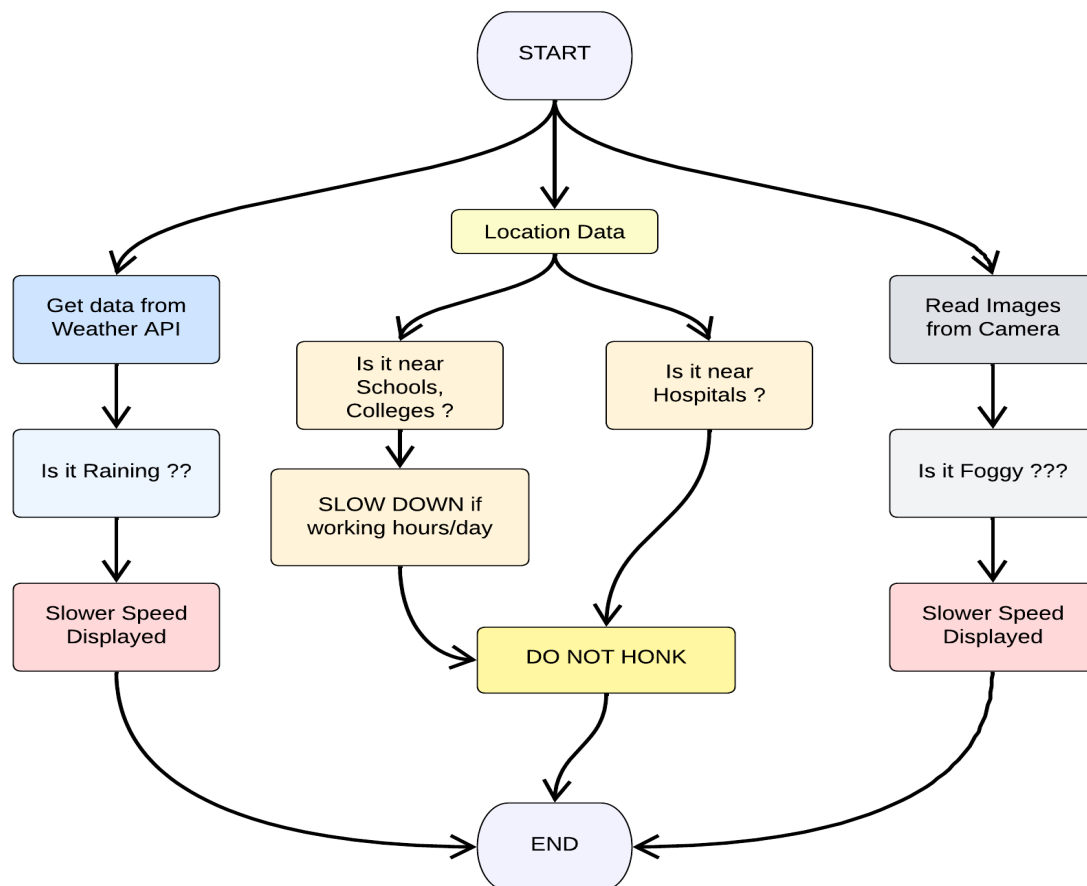
FR No.	Non-Functional Requirement	Description
NFR-1	<u>Usability</u>	It should be able to Upgrade and Update when there is a need for it.
NFR-2	<u>Security</u>	It should have good security system so that no other person is able to hack and display their own directions.
NFR-3	<u>Reliability</u>	It should be able to display to information correctly and error-free.
NFR-4	<u>Performance</u>	It should be able to automatically update itself when certain weather or traffic problem occurs.
NFR-5	<u>Availability</u>	It should be available 24/7 so that it can be beneficial to the customer i.e the driver.
NFR-6	<u>Scalability</u>	It should be able to easily change and upgrade according to change and need in requirement.

5. PROJECT DESIGN:

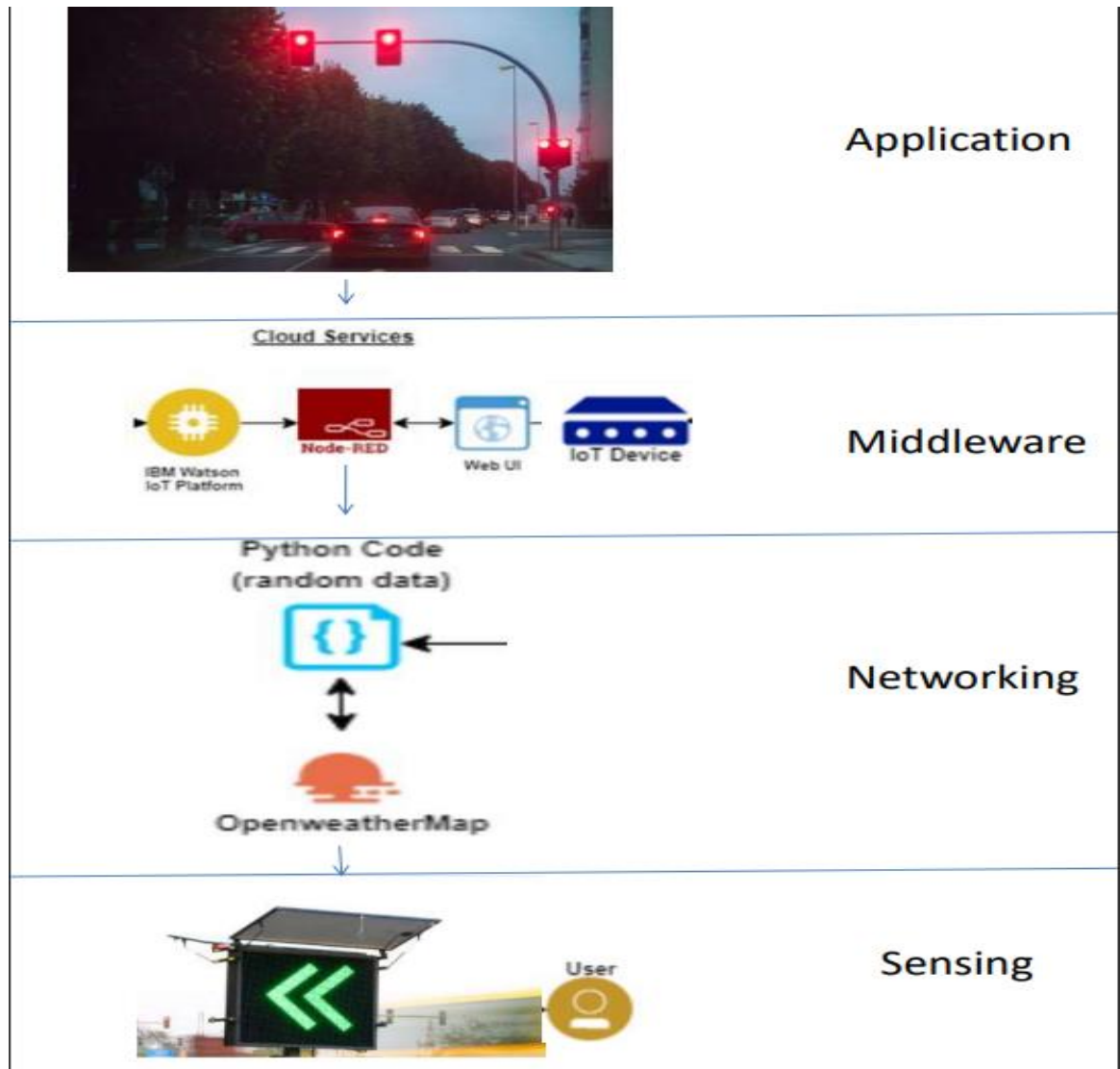
Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

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.



Solution & Technical Architecture:



Userstories:

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

User Type	Functional Requirement (Epic)	User Story Number	UserStory/Task	Acceptancecriteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	I can get my speed limitation using weather application .	I can receive speed limitations	High	Sprint-1
		USN-2	As a user, I can register for the application by entering my email, password, and confirming my password. As a user,	I can access my account / dashboard	Medium	Sprint-2
		USN-3	As a user, I can increase or decrease my speed according to the weather change	I can increase or decrease my speed	High	Sprint-1
		USN-4	As a user, I can I get my traffic diversion signs depending on the traffic and the fatal situations.	I can access my traffic status ahead in my travel	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the open weather map by entering email & password	I can access the application through my Gmail login	High	Sprint-2
	Interface	USN-6	As a user the interface should be simple and easily accessible	I can access the interface easily	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
Administrator (Officials)	Problem solving/ Fault clearance	USN-8	As an official who is in charge for the proper functioning of the sign boards have to maintain it through periodic monitoring.	Officials can monitor the sign boards for proper functioning.	Medium	Sprint-2
			entering my email, password, and confirming my password	dashboard		

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.

Sprint Planning & Estimation:

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
Sprint-1	Preparation	USN-1	<p>Install the Python IDE.</p> <p>Install the required python libraries:</p> <ul style="list-style-type: none"> Install Watson IoT Python SDK to connect to IBM Watson IoT Platform using python code: <p>give the following command in command prompt: pip install wiotp-sdk</p> <p>Download the required files from the link</p> <p>Create a fast SMS service for sending the messages and getting the API</p> <p>Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Prerequisites/Software.pdf</p>	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-1	Preparation	USN-2	<p>Create An Account In Openweathermap Website</p> <p>Using Openweathermap we can get current weather details of a location and integrate this with our project</p> <p>Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Prerequisites/Create%20An%20Account%20In%20Openweathermap%20Website.pdf</p> <p>Create An Account In MIT App inventor Website</p> <p>Create an account in MIT app inventor website and download MIT AI2 companion app in mobile.</p>	5	Low	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-1	Preparation	USN-3	<p>IBM Cloud Services</p> <p>Need to have basic knowledge of the following cloud services:</p> <ul style="list-style-type: none"> IBM Watson IoT Platform 	5	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
			<ul style="list-style-type: none"> Node-RED Service <p>Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Prerequisites/IBM%20Cloud%20Services.pdf</p> <p>Sprint-1 Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-1/Sprint-1.pdf</p>			
Sprint-2	Create And Configure IBM Cloud Services	USN-4	<p>Create IBM Watson IoT Platform And Device</p> <ul style="list-style-type: none"> IBM Watson IoT platform acts as the mediator to connect the web application to IoT device, so create the IBM Watson IoT platform. In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials. Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform. <p>Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Create%20And%20Configure%20IBM%20Cloud%20Services/Create%20IBM%20Watson%20IoT%20Platform%20And%20Device.pdf</p>	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-2	Create And Configure IBM Cloud Services	USN-5	<p>Create Node-RED Service</p> <p>To create a web application create a Node-RED service.</p> <p>Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Create%20And%20Configure%20IBM%20Cloud%20Services/Create%20Node-RED%20Service.pdf</p> <p>Sprint-2 Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-2/Sprint-2.pdf</p>	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
Sprint-3	Develop The Python Script	USN-6	Develop A Python Script Create a code snippet using python to <ul style="list-style-type: none"> Extract weather data from OpenWeatherMap using APIs Send the extracted data to the cloud Receive data from the cloud and view it in the python compiler Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Develop%20The%20Python%20Script/Develop%20A%20Python%20Script.pdf	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-3	Develop The Python Script	USN-7	Publish Data To The IBM Cloud Python code is used to send random sensor data to the cloud and also to receive commands from the cloud. When the commands are received just print the statements which represent the control of the devices. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/tree/main/Develop%20The%20Python%20Script/Publish%20Data%20To%20The%20IBM%20Cloud Sprint-3 Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-3/Sprint-3.pdf	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web Application Using Node-RED Service.	USN-8	Develop The Web Application Using Node-RED Configure the Node-RED flow to send data to the IBM IoT platform. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Develop%20A%20Web%20Application%20Using%20Node-RED%20Service/Develop%20The%20Web%20Application%20Using%20Node-RED/Create%20Node%20Red%20Flow%20To%20Send%20Data%20To%20Device.pdf	5	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web	USN-9	Use Dashboard Nodes For Creating UI(Web App)	5	High	1.Mugila R 2.Ishwariya P 3.Kalpana T

Milestone number	Functional Requirement (Epic)	User Story Number	Activity	Story Points	Priority	Team Members
	Application Using Node-RED Service.		Create use dashboard nodes to visualize the data in graphical format. Git Repo Submission Link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Development%20Web%20Application%20Using%20Node-RED%20Service/Use%20Dashboard%20Nodes%20For%20Creating%20UI/Web%20App/UI%20Nodes%20Installation.pdf			4.Shreein Fathima S
Sprint-4	Develop A Mobile Application Using MIT App inventor.	USN-10	Use MIT App For Creating Mobile application Build the app. Using MIT AI2 companion(in mobile) by connecting it with the app build(click connect in app inventor then click AI companion and scan the QR code in mobile companion- it will be connected) we can see the Road Safety informations in mobile. Sprint-4 link: https://github.com/IBM-EPBL/IBM-Project-18272-1659682357/blob/main/Project%20Development%20Phase/Sprint-4/Sprint-4.pdf	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Preparation	USN-1	Install the Python IDE. Install the required python libraries: <ul style="list-style-type: none"> Install Watson IoT Python SDK to connect to IBM Watson IoT Platform using python code: give the following command in command prompt: pip install	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			wiotp-sdk Download the required files from the link Create a fast SMS service for sending the messages and getting the API			
Sprint-1	Preparation	USN-2	Create An Account In OpenweatherMap Website Using Openweathermap we can get current weather details of a location and integrate this with our project Create An Account In MIT App inventor Website Create an account in MIT app inventor website and download MIT AI2 companion app in mobile.	5	Low	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-1	Preparation	USN-3	IBM Cloud Services Need to have basic knowledge of the following cloud services: <ul style="list-style-type: none"> • IBM Watson IoT Platform • Node-RED Service 	5	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-2	Create And Configure IBM Cloud Services	USN-4	Create IBM Watson IoT Platform And Device <ul style="list-style-type: none"> • IBM Watson IoT platform acts as the mediator to connect the web application to IoT device, so create the IBM Watson IoT platform. • In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials. 	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			<ul style="list-style-type: none"> Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform. 			
Sprint-2	Create And Configure IBM Cloud Services	USN-5	Create Node-RED Service To create a web application create a Node-RED service.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-3	Develop The Python Script	USN-6	Develop A Python Script Create a code snippet using python to <ul style="list-style-type: none"> Extract weather data from OpenWeatherMap using APIs Send the extracted data to the cloud Receive data from the cloud and view it in the python compiler 	10	Medium	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-3	Develop The Python Script	USN-7	Publish Data To The IBM Cloud Python code is used to send random sensor data to the cloud and also to receive commands from the cloud. When the commands are received just print the statements which represent the control of the devices.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web Application Using Node-RED Service.	USN-8	Develop The Web Application Using Node-RED Configure the Node-RED flow to send data to the IBM IoT platform.	5	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Web	USN-9	Use Dashboard Nodes For Creating UI(Web App)	5	High	1.Mugila R 2.Ishwariya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	Application Using Node-RED Service.		Create use dashboard nodes to visualize the data in graphical format.			P 3.Kalpana T 4.Shreein Fathima S
Sprint-4	Develop A Mobile Application Using MIT App inventor.	USN-10	Use MIT App For Creating Mobile application Build the app. Using MIT AI2 companion(in mobile) by connecting it with the app builded(click connect in app inventor then click AI companion and scan the QR code in mobile companion- it will be connected) we can see the Road Safety informations in mobile.	10	High	1.Mugila R 2.Ishwariya P 3.Kalpana T 4.Shreein Fathima S

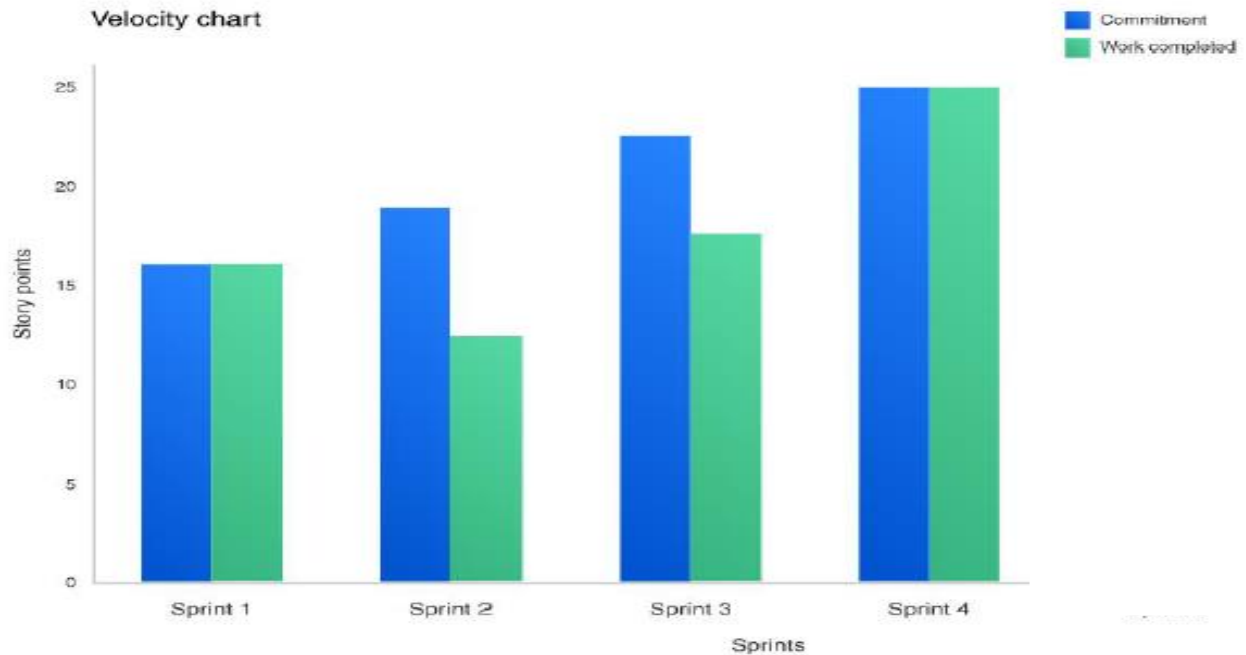
Velocity:

Imagine we have a 10-day sprint 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)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = 20/6 = 3.33$$

Burndown Chart:

A burn-down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as scrum. However, burn-down charts can be applied to any project containing measurable progress over time.



7.CODING & SOLUTIONING:

(Explain the features added in the project along with code):

Feature 1 (coding and result):

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

```
myConfig = {
    #Configuration
    "identity": {
        "orgId": "3dpjnk",
        "typeId": "Sign_Board",
        "deviceId": "Board_1"
    },
    #API Key
    "auth": {
        "token": "1234567890"
    }
}
```

```
#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 = "Nagercoil"
URL = BASE_URL + "q=" + "Chennai" + "&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==3:
            message="SLOW DOWN, HOSPITAL NEARBY"
        elif msg==5:
            message="NEED HELP, POLICE STATION NEARBY"
        else:
            message=""

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

        #Sign 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 < 50:

```

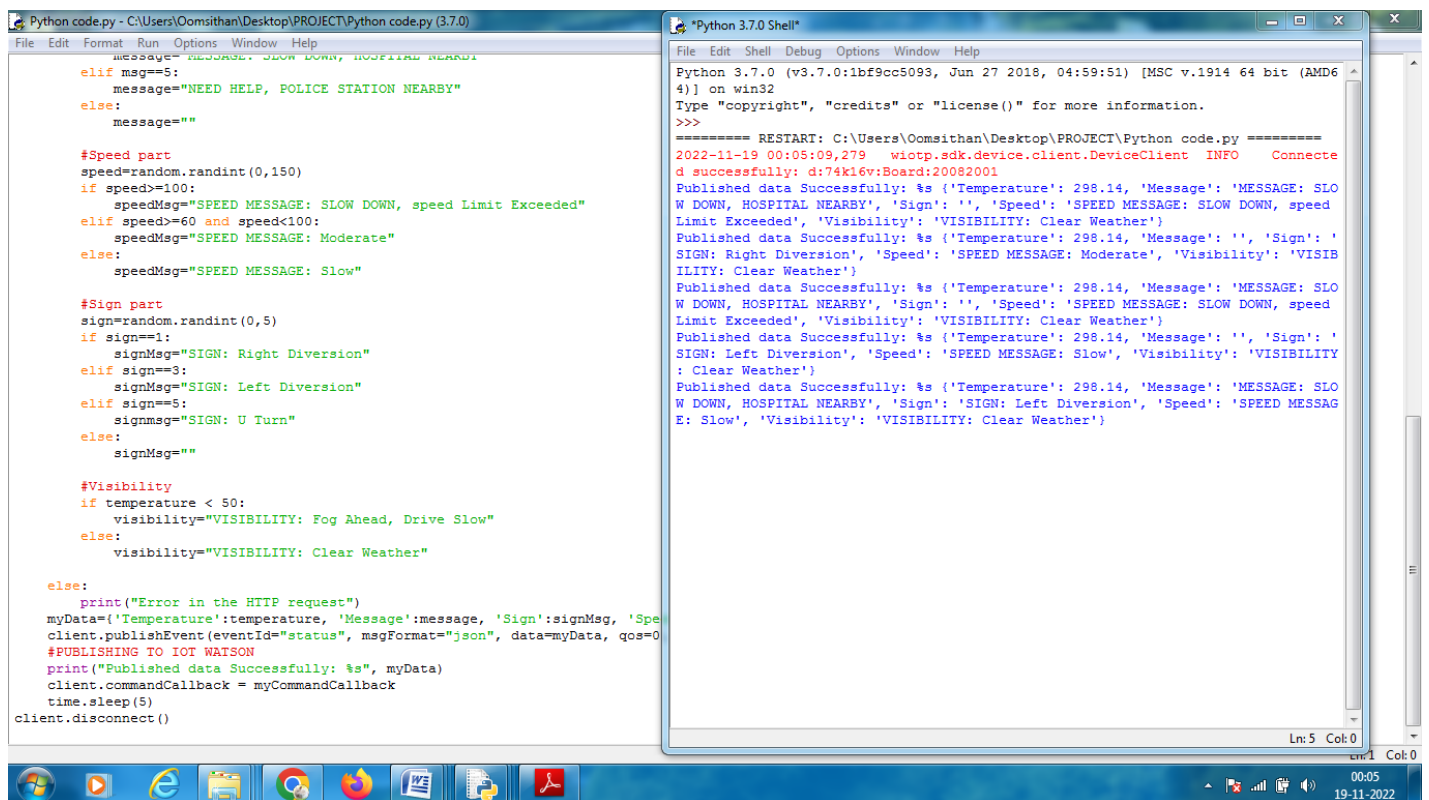
```

        visibility="Fog Ahead, Drive Slow"
    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



The screenshot shows a Python IDE with two windows. The left window displays a Python script for an IoT device simulation. The right window shows the output of the script in a Python 3.7.0 Shell.

Python code.py (Left Window):

```

message="MESSAGE: SLOW DOWN, HOSPITAL NEARBY"
elif msg==5:
    message="NEED HELP, POLICE STATION NEARBY"
else:
    message=""

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

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

#Visibility
if temperature < 50:
    visibility="VISIBILITY: Fog Ahead, Drive Slow"
else:
    visibility="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()

```

Python 3.7.0 Shell (Right Window):

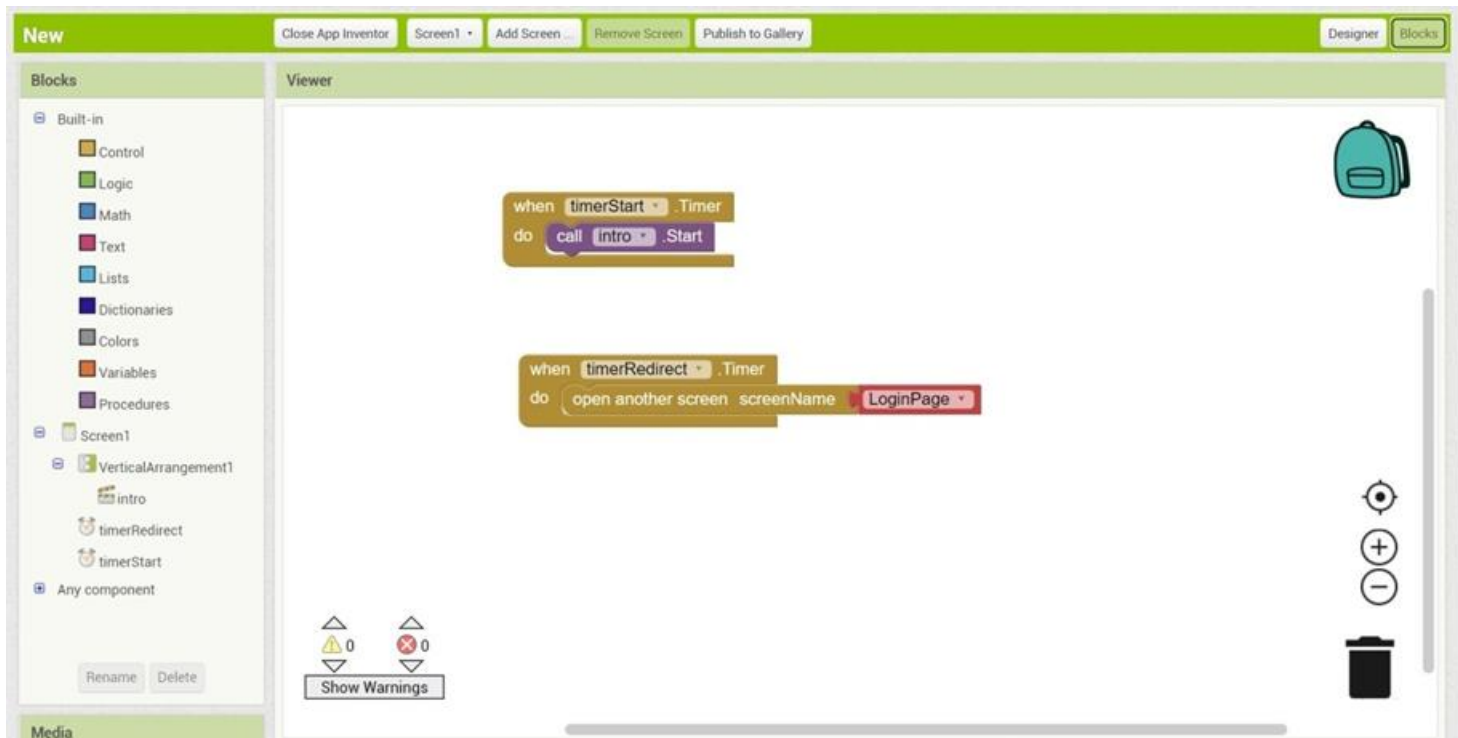
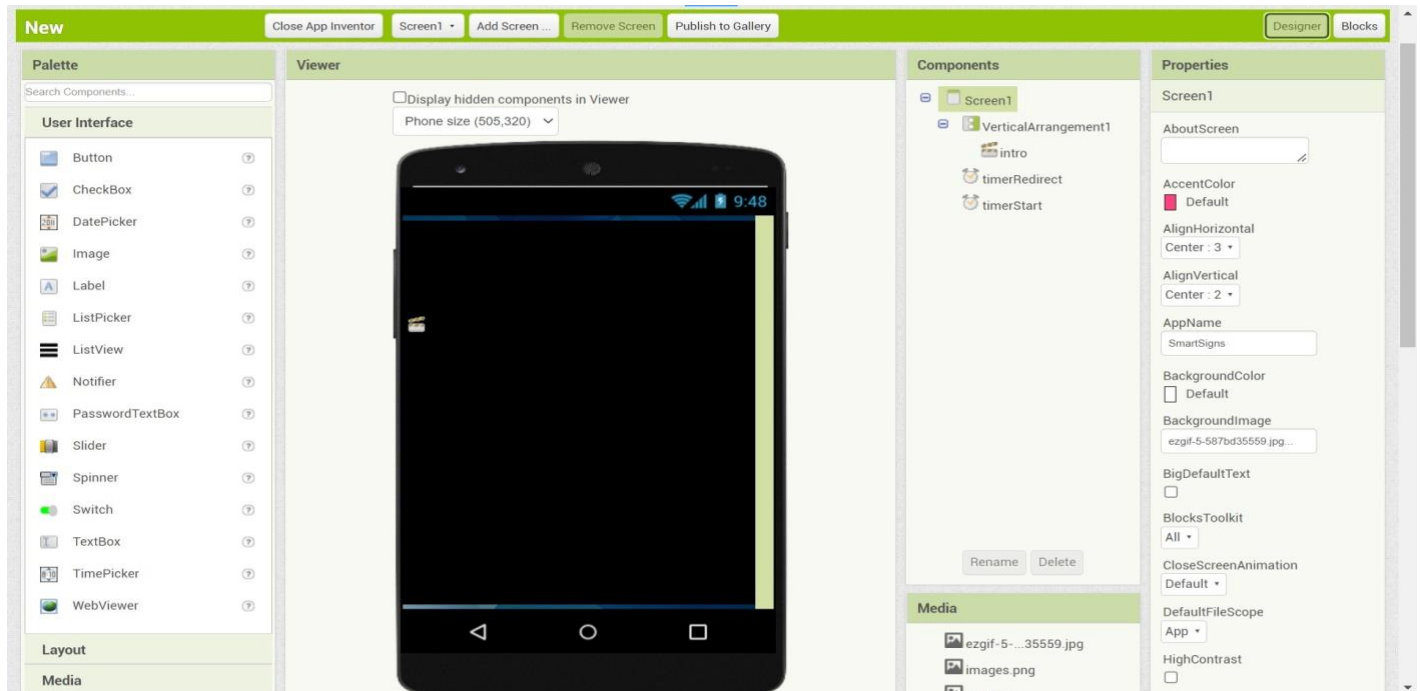
```

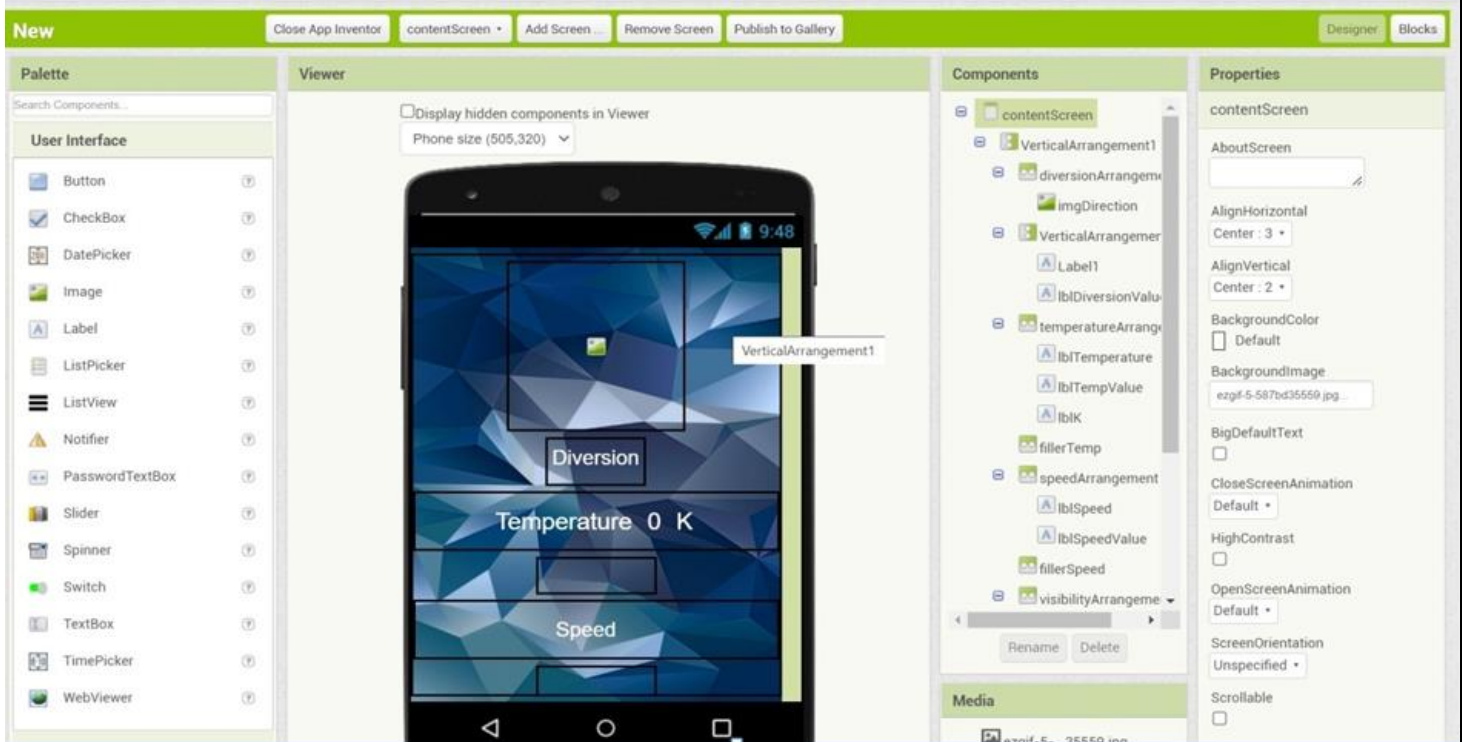
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Oomsithan\Desktop\PROJECT\Python code.py =====
2022-11-19 00:05:09.279 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:74kl6v:Board:20082001
Published data Successfully: %s {'Temperature': 298.14, 'Message': 'MESSAGE: SLO
W DOWN, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'SPEED MESSAGE: SLOW DOWN, speed
Limit Exceeded', 'Visibility': 'VISIBILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': '', 'Sign': '
SIGN: Right Diversion', 'Speed': 'SPEED MESSAGE: Moderate', 'Visibility': 'VISIB
ILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': 'MESSAGE: SLO
W DOWN, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'SPEED MESSAGE: SLOW DOWN, speed
Limit Exceeded', 'Visibility': 'VISIBILITY: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': '', 'Sign': '
SIGN: Left Diversion', 'Speed': 'SPEED MESSAGE: Slow', 'Visibility': 'VISIBILITY
: Clear Weather'}
Published data Successfully: %s {'Temperature': 298.14, 'Message': 'MESSAGE: SLO
W DOWN, HOSPITAL NEARBY', 'Sign': 'SIGN: Left Diversion', 'Speed': 'SPEED MESSAG
E: Slow', 'Visibility': 'VISIBILITY: Clear Weather'}

```

Feature 2: (MITAPPINVENTER):

MIT APP INVENTOR: ICON PAGE:





New Close App Inventor contentScreen Add Screen Remove Screen Publish to Gallery Designer Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Dictionaries
 - Colors
 - Variables
 - Procedures
- contentScreen
 - VerticalArrangement1
 - diversionArrangement1
 - imgDirection
 - VerticalArrangement2
 - Label1
 - lblDiversionValue

Rename Delete

Media

Viewer

0 0

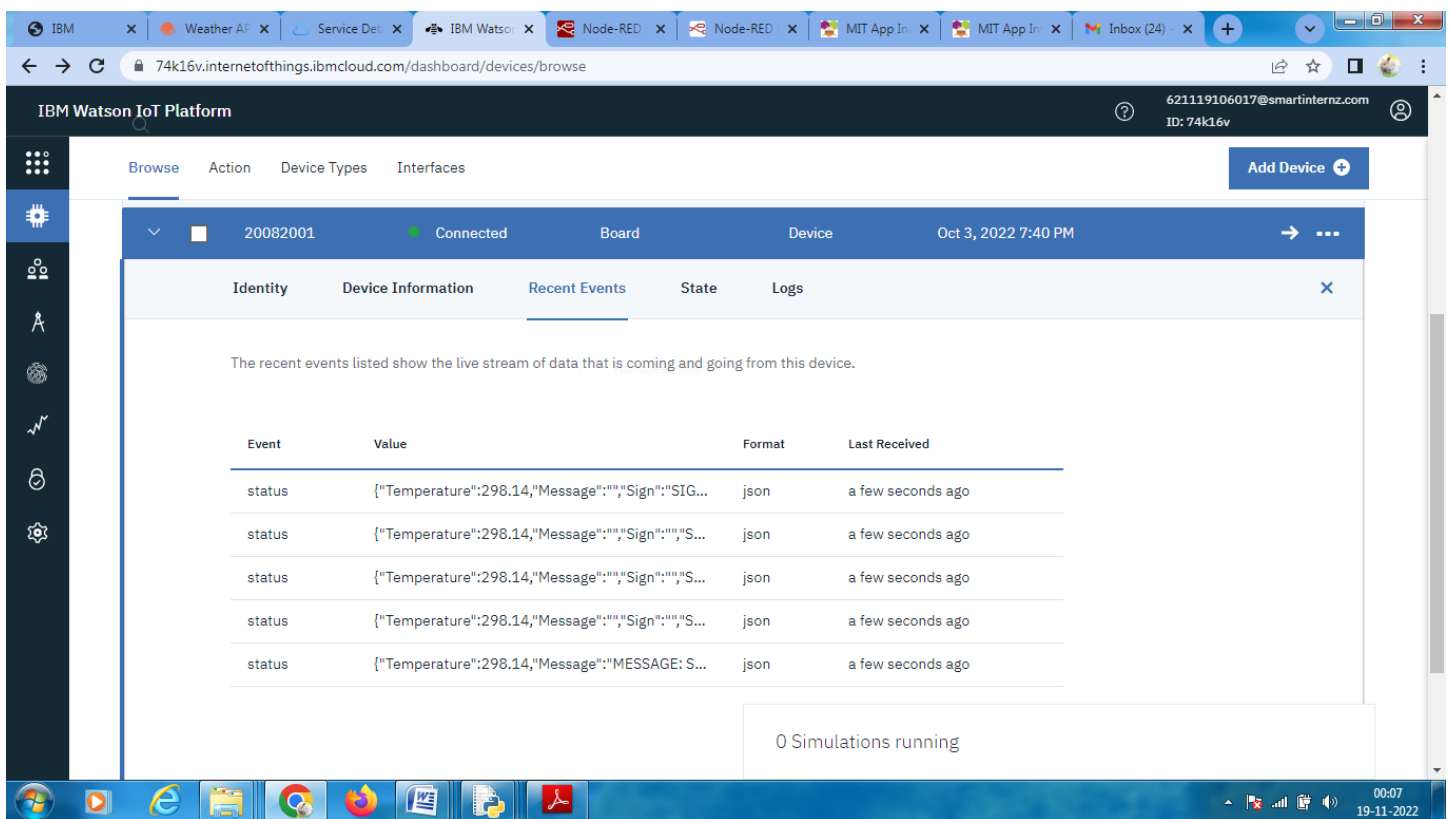
Show Warnings



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).

TestCases:



IBM Watson IoT Platform

621119106017@smartintenz.com
ID: 74k16v

Browse Action Device Types Interfaces

20082001 Connected Board Device Oct 3, 2022 7:40 PM

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"Temperature":298.14,"Message":"","Sign":"SIG...	json	a few seconds ago
status	{"Temperature":298.14,"Message":"","Sign":"","S...	json	a few seconds ago
status	{"Temperature":298.14,"Message":"","Sign":"","S...	json	a few seconds ago
status	{"Temperature":298.14,"Message":"","Sign":"","S...	json	a few seconds ago
status	{"Temperature":298.14,"Message":"MESSAGE: S...	json	a few seconds ago

0 Simulations running

UI : <https://74k16v.internetofthings.ibmcloud.com/dashboard/devices/browse>
<https://node-red-xqkno-2022-10-03.mybluemix.net/red/#flow/895a58d22de00831>

The screenshot displays the Node-RED web interface. On the left, the 'common' and 'function' node palettes are visible. The main workspace shows 'Flow 1' with an 'IBM IoT' node connected to a 'msg.payload' node. This is followed by five parallel function nodes: 'temperature', 'MESSAGE', 'sign', 'Speed', and 'Visibility'. Each function node is connected to a corresponding output node. The right sidebar shows the debug console with logs for each message received.

```

11/19/2022, 6:00:19 AM node: f2f2649a.0d0d98
iot-2/type/Board/id/20082001/evnt/status/fmt/json :
msg.payload : Object
{
  Temperature: 296.14, Message: "",
  Sign: "", Speed: "SPEED MESSAGE: SLOW
DOWN, spee-", Visibility:
"VISIBILITY: Clear Weather"
}

11/19/2022, 6:00:24 AM node: f2f2649a.0d0d98
iot-2/type/Board/id/20082001/evnt/status/fmt/json :
msg.payload : Object
{
  Temperature: 296.14, Message: "",
  Sign: "", Speed: "SPEED MESSAGE: SLOW
DOWN, spee-", Visibility:
"VISIBILITY: Clear Weather"
}

11/19/2022, 6:00:30 AM node: f2f2649a.0d0d98
iot-2/type/Board/id/20082001/evnt/status/fmt/json :
msg.payload : Object
{
  Temperature: 296.14, Message:
"MESSAGE: SLOW DOWN, SCHOOL IS ...",
  Sign: "", Speed: "SPEED MESSAGE: SLOW
DOWN, spee-", Visibility: "VISIBILITY: Clear
Weather"
}

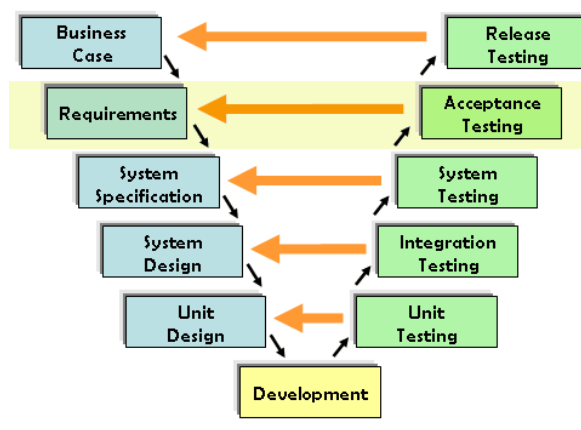
11/19/2022, 6:00:35 AM node: f2f2649a.0d0d98
iot-2/type/Board/id/20082001/evnt/status/fmt/json :
msg.payload : Object

```

The screenshot displays the IBM IoT dashboard. The top section has a blue header with the text 'Home'. Below this, there is a section titled 'Signs with smart connectivity for better road safety'. This section contains three status indicators: 'SIGN: Right Diversion', 'SPEED MESSAGE: Moderate', and 'VISIBILITY: Clear Weather'. Below these indicators is a 'temperature' gauge showing a value of 296.14 degrees Celsius.

User Acceptance Testing:

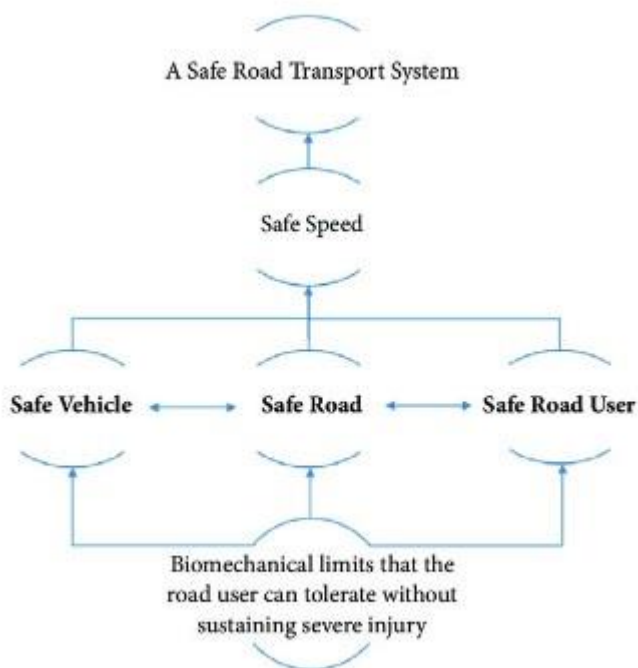
UAT consists, in practice, of people from the target audience using the application. The defects they find are then reported and fixed. This scenario is what most closely resembles “the real world.” The process allows users to “get their hands dirty” with the application. They can see if things work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

Performance Metrics:



10. ADVANTAGES & DISADVANTAGES

Advantages:

Connected vehicles have various benefits such as

- Multimodal sensors and edge computing helps 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.
- Improved control and safety can be achieved through IoT-enabled cars. In case of over-speeding, the notification gets displayed.
- Ensuring a safe driving experience with real-time assistance, navigation, and even monitoring driving patterns and any emergency. Additionally, along with the state of the traffic, IoT drivers can receive updated information on the state of the roads, i.e., potholes, ice, grade changes, black spots, etc.

DISADVANTAGES:

- Security and privacy. Keeping the data gathered and transmitted by IoT devices safe is challenging, as they evolve and expand in use....
- Technical complexity....
- Connectivity and power dependence....
- Integration....
- Higher costs (time and money)

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. FUTURESCOPE:

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 design ensuring safe vehicles, road networks, and road users. Evolving towards the future, the road needs to be 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": "3dpjnk",
        "typeId": "Sign_Board",
        "deviceId": "Board_1"
    },
    #API Key
    "auth": {
        "token": "1234567890"
    }
}

#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 = "Nagercoil"
URL = BASE_URL + "q=" + "Chennai" + "&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==3:
            message="SLOW DOWN, HOSPITAL NEARBY"
        elif msg==5:
            message="NEED HELP, POLICE STATION NEARBY"
        else:
            message=""

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

        #Sign 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 < 50:
            visibility="Fog Ahead, Drive Slow"

```

```
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 Link :

<https://github.com/IBM-EPBL/IBM-Project-18272-1659682357>

Project demo link :

<https://youtu.be/iJvhOudqiV4>

THANKYOU