



SRI VENKATESWARAA COLLEGE OF TECHNOLOGY

PROJECT REPORT

Project Name: Smart Solution for Railways

Team ID : PNT2022TMID38325

Team Leader : Subathra. S

Team Member : Sasikumar. M

Priyanka. N

Muniyappan. R

CONTENTS

1. INTRODUCTION

1.1 Project Overview

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

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning, Schedule & Estimation

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

7.1 Feature

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

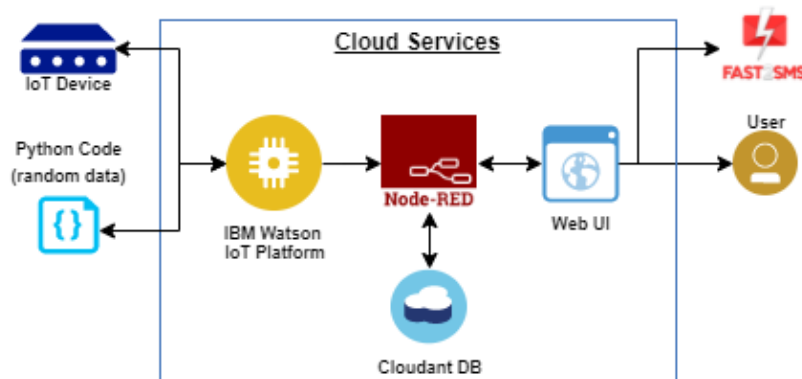
14. Source Code

GitHub Link

1.INTRODUCTION

1.1 Project Overview

- Using the Web application, a user books a ticket based on the availability of the seats by giving the general required information.
- Once a user clicks on the submit button, a QR code is generated with a Unique ID and the data is stored in the Cloudant DB with that Unique ID.
- Users can save the QR code for further process.
- In python code, a Ticket collector can scan the QR code and extract the information from the QR Code i.e., Unique ID. With that Unique ID, data is fetched from the Cloudant DB, if it is not found, then it displays Not a Valid Ticket.
- Also, the live location of the train will be published to IBM IoT platform using python code
- The train location can be tracked from a Web Application.



2.LITERATURE SURVEY

2.1 Existing problem

1.TITLE : Smart Solutions For Railways

Authors: N. C. Brintha; Chakka Venkata Pavan Tarun; Lingisetty Abhishikth; Bitra Purna Koteswara Rao; Maddika Taruneshwar Reddy

Description:

Most of us waste hours of our lives waiting near railway gates due to the traditional human management system imbedded near railway gates, and many individuals have lost their lives by mistakenly crossing the gate at the moment of train arrival near railway gate. In our proposed solution, we have addressed these issues by displaying whether the railway gate is opened or closed based on the train passing, the most recent gate closure timestamp, and each time when the gate is closed, a centralized platform for knowing the condition of the gate, traffic density, and intrusion detection near the railway gate. Based on the experimental results, this automated railway crossing assistance outperformed human-assisted railway crossings, resulting in an efficient and cost-effective solution for traffic regulation and avoidance of accidents at railway gates.

2. TITLE : Controlling railway gates using Smart phones by tracking with GPS

Authors: R. Velayutham;T. Sangeethavani;K. Sundaralakshmi

Description:

Railway is the most commonly used transportation vehicle. Most of the people choose this transportation mainly for low cost and it gives comfort ability. To increase this comfort zone and to reduce the number of accidents, our system gives a complete solution. Nowadays, we met a lot of railway accidents. Most of these accidents occur at railway gate level crossings. Up to date there are several proposals to avoid the number of accidents and to reduce the manpower at railway gates. Most of the proposals used sensors as a key device to detect the arrival of the train but which requires maintenance cost and it is not efficient method. It also requires monitoring process to detect the fault in sensors. This paper track and detect the arrival of the train by using GPS and not by the sensors. This way of train tracking using GPS is embedded with our mobile application. Using this application the engine driver controls the railway gate.

3.TITLE : The IDex Safety Measures of An IoT-based railway Infrastructures

Authors: C. K. Wu, Y. He, K. F. Tsang and S. Mozar

Description:

Railway safety is a tremendously essential aspect of urban management. The railway network is a highly sophisticated infrastructure comprising numerous devices, machines, networks, services, etc. Therefore, the Internet-of-things (IoT) and Artificial Intelligence (AI) can be conceived as the crucial technologies for enabling ubiquitous data collection and unmanned decision-making, respectively. The IoT maturity index, namely the IDex, will be standardized to provide a comprehensive evaluation for all IoT objects pertinent to their performance and application requirements.

4. TITLE: 5G Key Technologies for Smart Railways

Authors: B. Ai, A. F. Molisch, M. Rupp and Z. -D. Zhong

Description:

Railway communications has attracted significant attention from both academia and industries due to the booming development of railways, especially high-speed railways (HSRs). To be in line with the vision of future smart rail communications, the rail transport industry needs to develop innovative communication network architectures and key technologies that ensure high-quality transmissions for both passengers and railway operations and control systems. Specifically, we have identified significant 5G-based key technologies for HSRs, such as spatial modulation, fast channel estimation, cell-free massive multiple-input-multiple-output (MIMO), mmWave, efficient beamforming, wireless backhaul, ultra-reliable low latency communications, and enhanced handover strategies

2.2 REFERENCE

1) N. C. Brintha;Chakka Venkata Pavan Tarun;Lingisetty Abhishikth;Bitra Purna Koteswara Rao;Maddika Taruneshwar Reddy

2) R. Velayutham;T. Sangeethavani;K. Sundaralakshmi

3) C. K. Wu, Y. He, K. F. Tsang and S. Mozar

4) B. Ai,A. F. Molisch, M. Rupp and Z. -D. Zhon

IEEE Papers:

IOT Based smart Railways

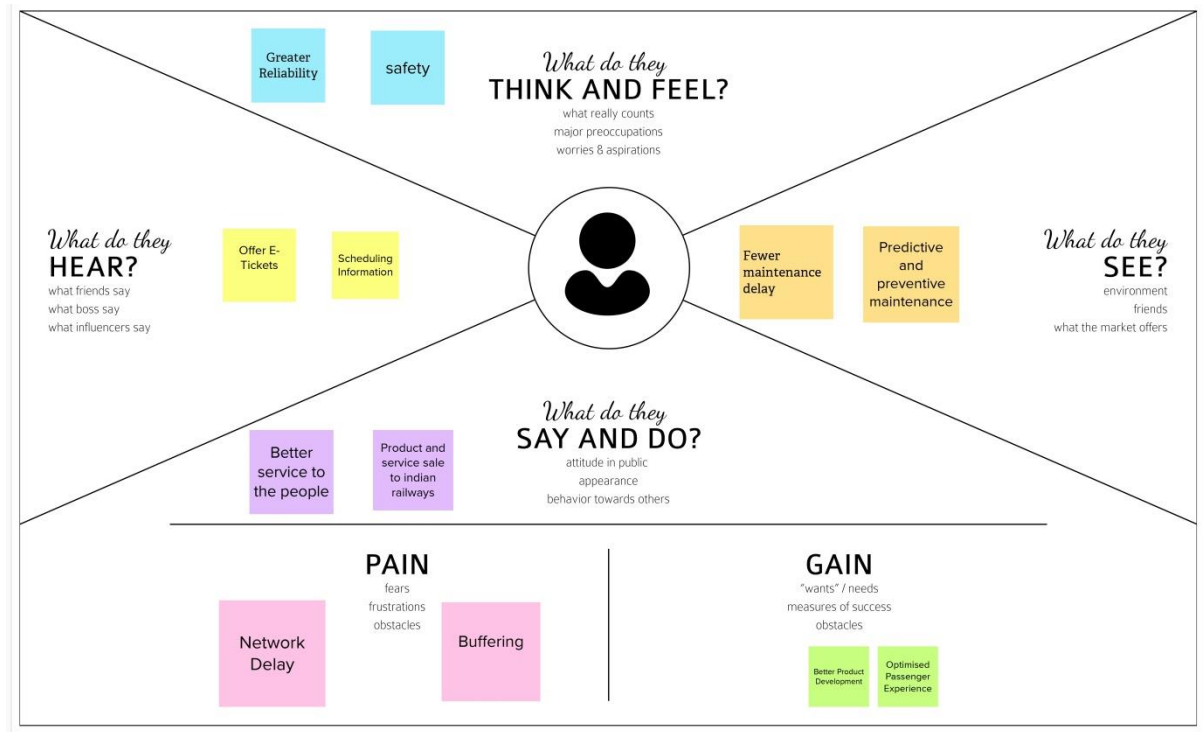
Publisher: IEEE

2.2 Problem Statement Definition

- To develop and implement advance technology in ticket booking and railways for new generation and
- To save the booking time of passengers through virtual mode

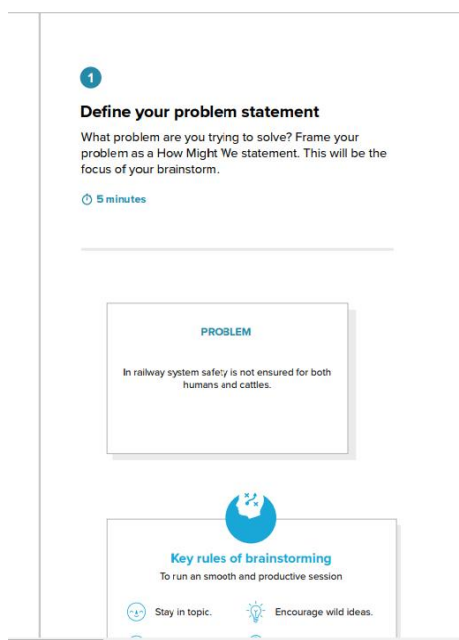
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map

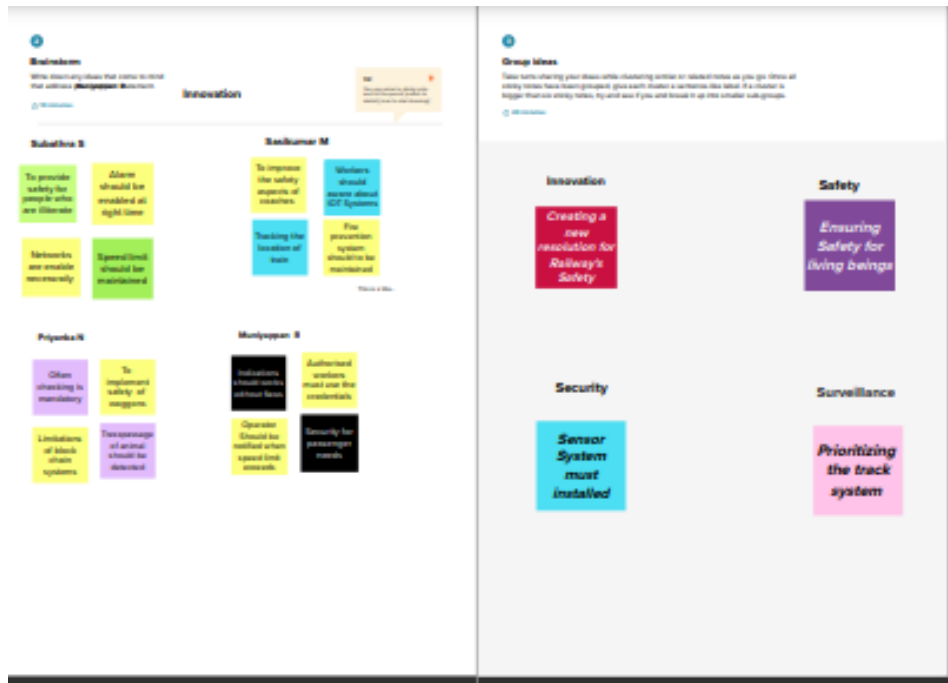


3.2 Ideation & Brainstorming

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



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

Proposed Solution Template:

Parameter

Description

1.

Problem to be solved

To reduce the work loads of the user and also use the paper

2.

Idea and solution description

1. To ensure the safety of passenger towards machinery
2. Using IoT system

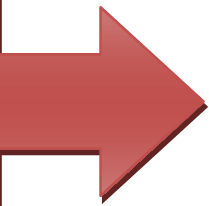
3.

Novelty/
Uniqueness

Our project is based on safety precaution and preventing them before incident using new technology

4.

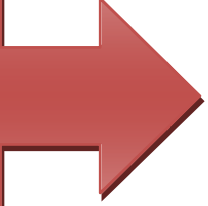
Social Impact /
Customer
Satisfaction



```
graph LR; A[Social Impact / Customer Satisfaction] --> B[So many peoples and animals lives can be saved. Easily handles by common people];
```

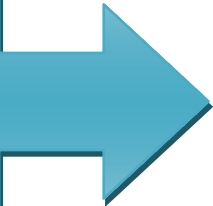
So many peoples
and animals lives
can be saved.

Easily handles by
common people



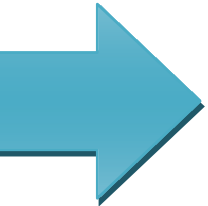
5.

Scalability of the
Solution

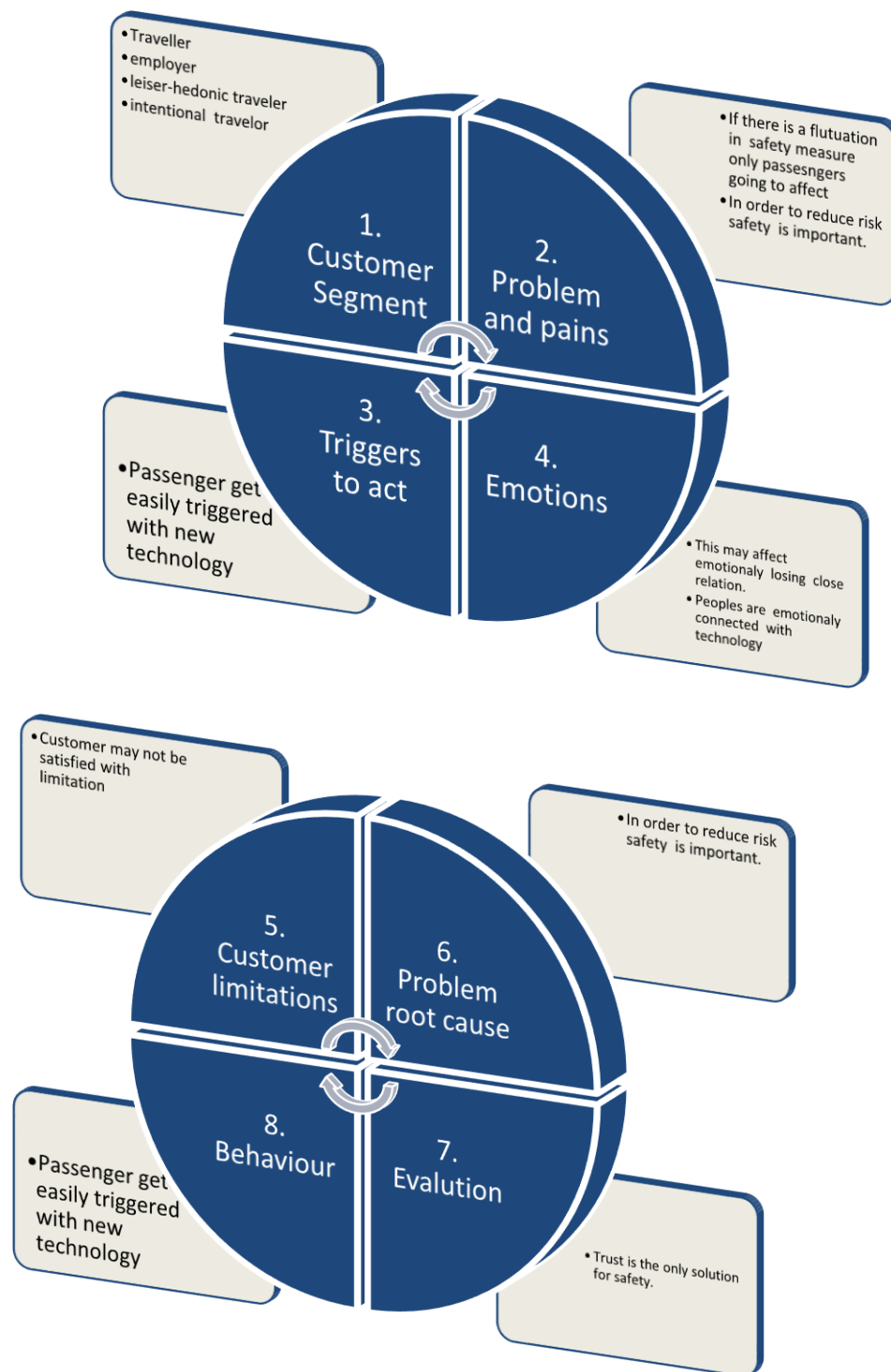


```
graph LR; A[Scalability of the Solution] --> B[This will take the technology level to the next stage and will reduce the hypertension of passengers during traveling];
```

This will take the
technology level to the
next stage and will
reduce the
hypertension of
passengers during
traveling



3.4 Problem solution fit



4. Requirement Analysis

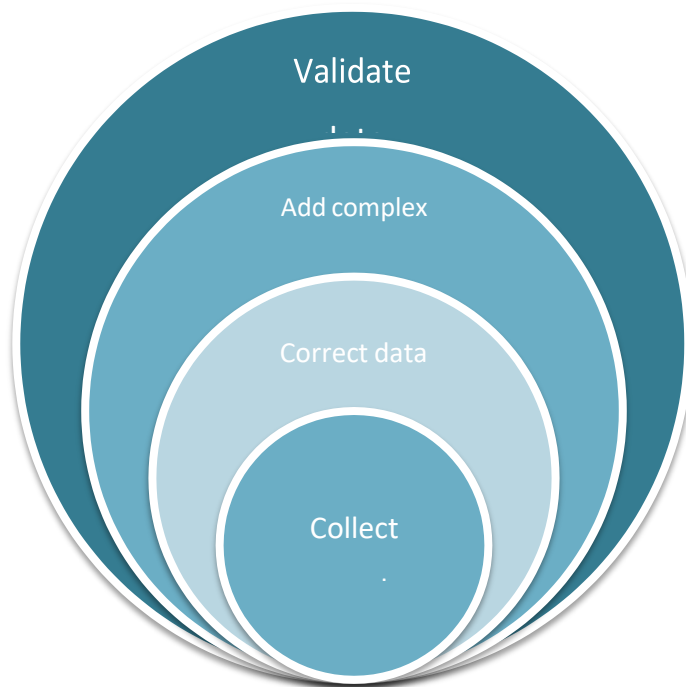
4.1 Functional Requirement

Functional Requirements	Sub Requirements
Safety Gap Analysis	a form of a safety audit that measures your current safety procedures against your safety requirements
Safety Confirmation Reviews	assess how well the work product adheres to the standard whereas verification
Safety System Audit	a structured process that is used to collect information related to a company's efficiency, effectiveness, reliability, and safety.
Safety (Project) Audit	an activity where a facility gathers information about one or more aspects of the workplace in order to evaluate the risk levels for health or safety issues
Safety Assessment	the systematic collection of information on threatening family conditions and current, significant, and clearly observable threats to the safety of the child or youth.

5. PROJECT DESIGN

5.1 Data flow diagrams

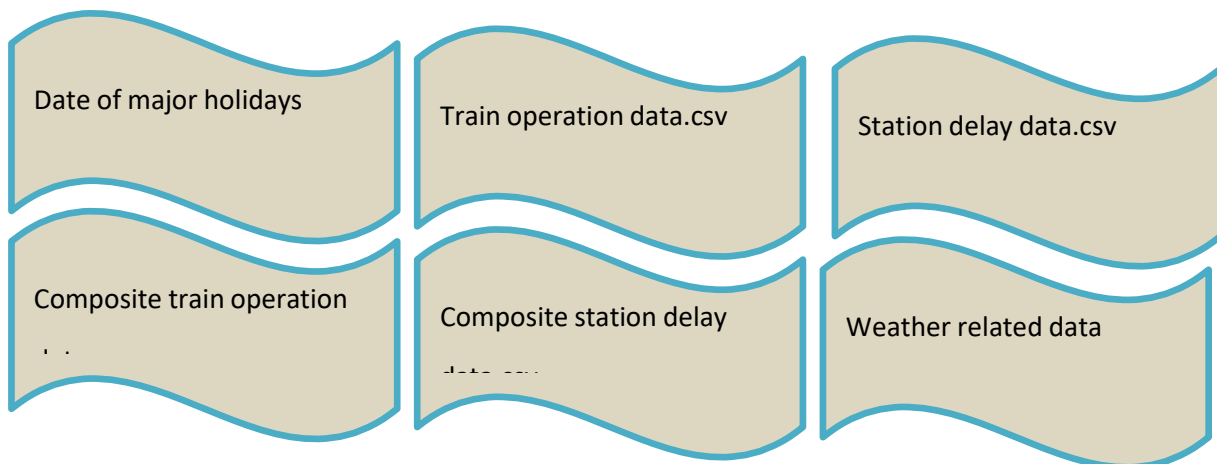
[A high speed railway network dataset from operation records and weather data](#)



Collect source data:-



Correct data:-



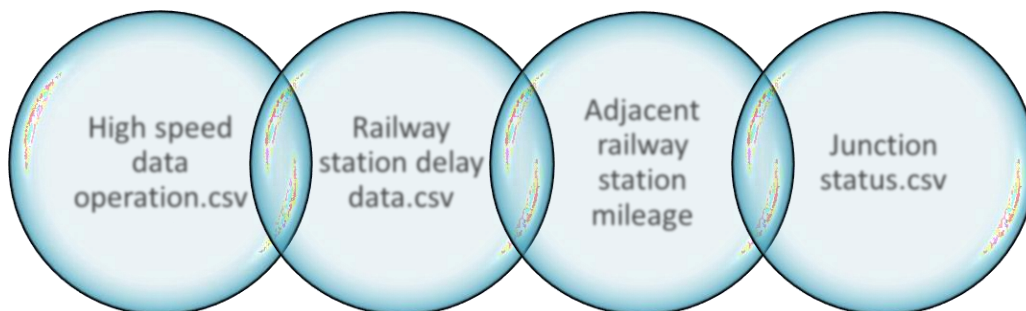
Add complex data factors:

Railway station location

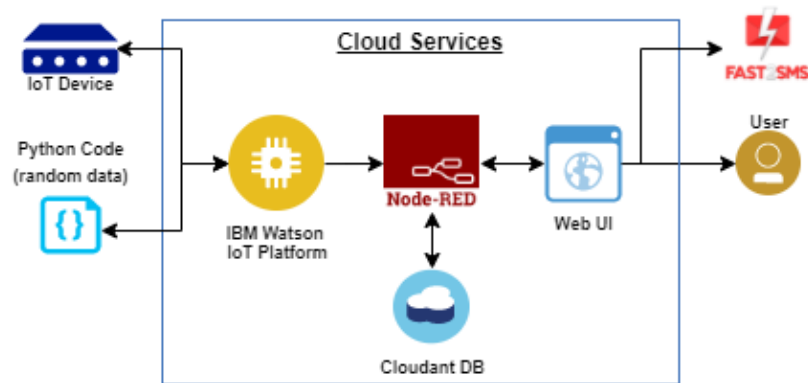
Junction status

Composite train opposite data

Validate data:-



5.2 Solution and Technical Architecture



6.PROJECT PLANNING AND SCHEDULING

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table 1 & table 2

Example: Order processing during pandemics for offline mode

Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>

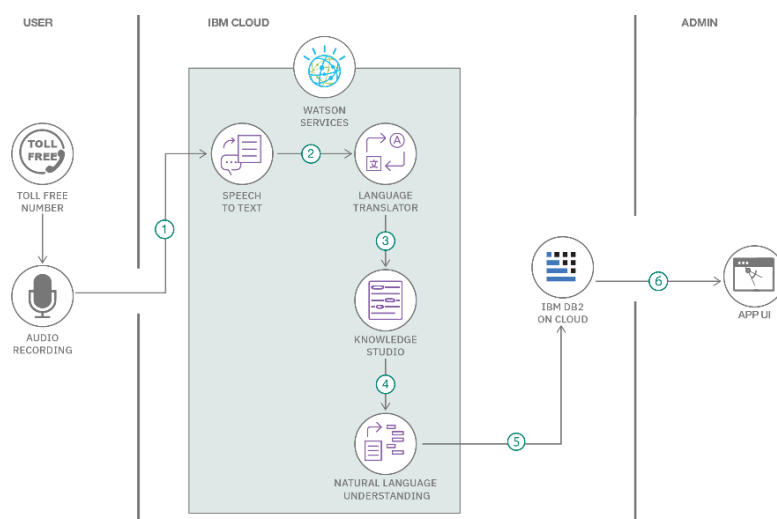


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

References:

<https://c4model.com/>

<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/> <https://www.ibm.com/cloud/architecture>

<https://aws.amazon.com/architecture>

<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>

7.CODING AND SOLUTIONING

7.1 Feature

Coding for determine the location of the Train

PROGRAM :

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": " 64j80w",
        "typeId": "GPS",
        "deviceId":"12345"
    },
    "auth": {
        "token": "12345678"
    }
}
```

```
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()

def pub (data):
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print ("Published data Successfully: %s", myData)

while True:
    myData={'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
    pub (myData)
    time.sleep (3)
    #myData={'name': 'Train2', 'lat': 17.6387448, 'lon': 78.4754336}
    #pub (myData)
    #time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
    pub(myData)
    time.sleep(3)
    myData={'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
    pub (myData)
    time.sleep (3)
    client.commandCallback = myCommandCallback
    client.disconnect ()

```

coding for determine the QR code for Checking ticket availability and seating

PROGRAM :

```

from ibmcloudant import CouchDbSessionAuthenticator

```

```

from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-16u3crmdpkgghxefdi kvpssoh5fwezrmuup5fv5g3ubz',
'b0ab119f45d3e6255eabb978')
service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-
16u3crmdpkgghxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119145d3e6255eabb978e7e2f0')

cap= cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_PLAIN

while True:
    __, frame = cap.read()
    decodedObjects = pyzbar.decode (frame)
    for obj in decodedObjects:
        #print ("Data", obj.data)
        a=obj.data.decode('UTF-8')
        cv2.putText(frame, "Ticket", (50, 50), font, 2,          (255, 0, 0), 3)

        #print (a)
        try:
            response = service.get_document(
                db='booking',
                doc_id = a
            ).get_result()
            print (response)
            time.sleep(5)
        except Exception as e:
            print ("Not a Valid Ticket")
            time.sleep(5)

    cv2.imshow("Frame",frame)
    if cv2.waitKey(1) & 0xFF ==ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
client.disconnect()

```

8.TESTING

8.1 Test case

Step 1: Adding nodes in Node-Red

The screenshot shows the IBM Cloud Developer console interface. The main header displays the app name "Smart Solution for Railways" and an "Actions..." dropdown. The left sidebar contains a "Details" section with the following information:

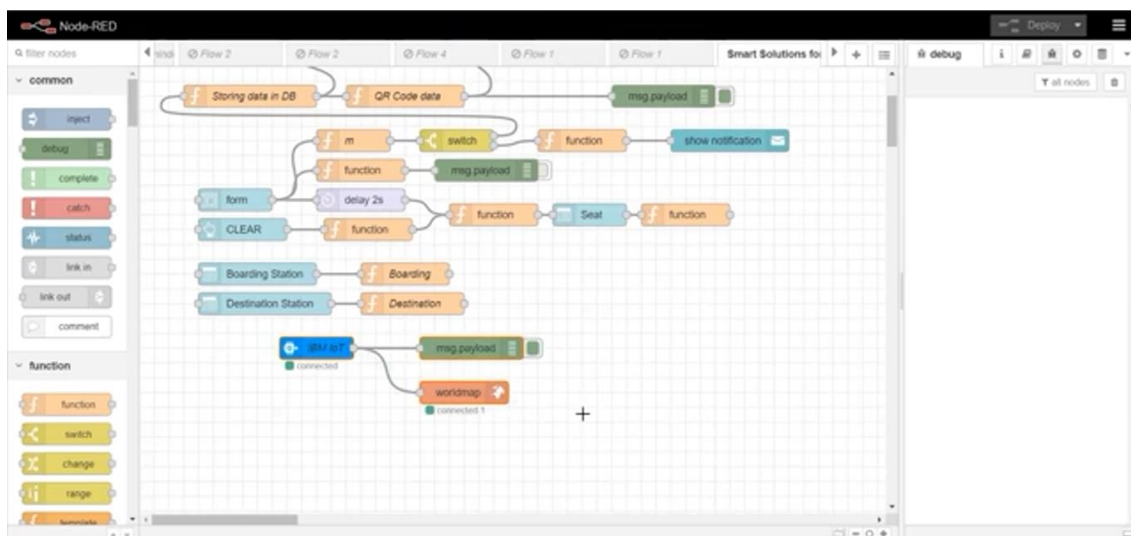
- App URL: <http://169.51.200.90:32499>
- Source: <https://us-south.git.cloud.ibm.com/412619106017/SmartSolu...>
- Resource group: Default
- Deployment target: mycluster-free
- Created: 11/19/2022

Below the details is a "Services" section showing a "Cloudant" service with links to "Open dashboard", "Documentation", and "API reference". There are also buttons for "Connect existing services" and "Create service".

The right sidebar contains a "Deployment Automation" section with the following details:

- Name: SmartSolutionforRailways
- Location: Dallas
- Tool integrations: (Icons for various tools)
- Delivery Pipelines: (List of pipelines)

Below this is a "Getting started quickly" section with a list of steps for configuring the app. A vertical "ASK A QUESTION" button is visible on the far right.

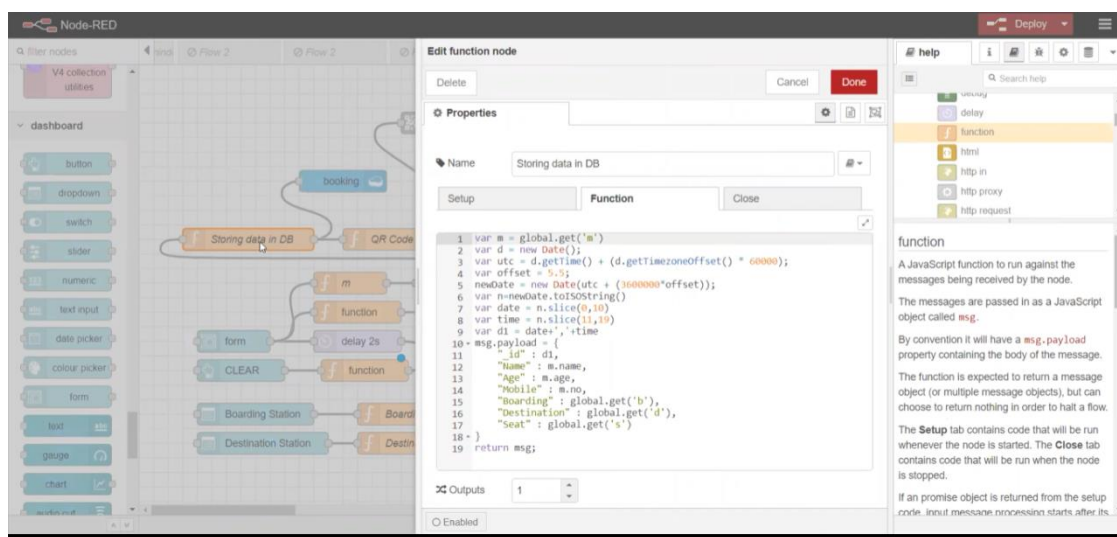
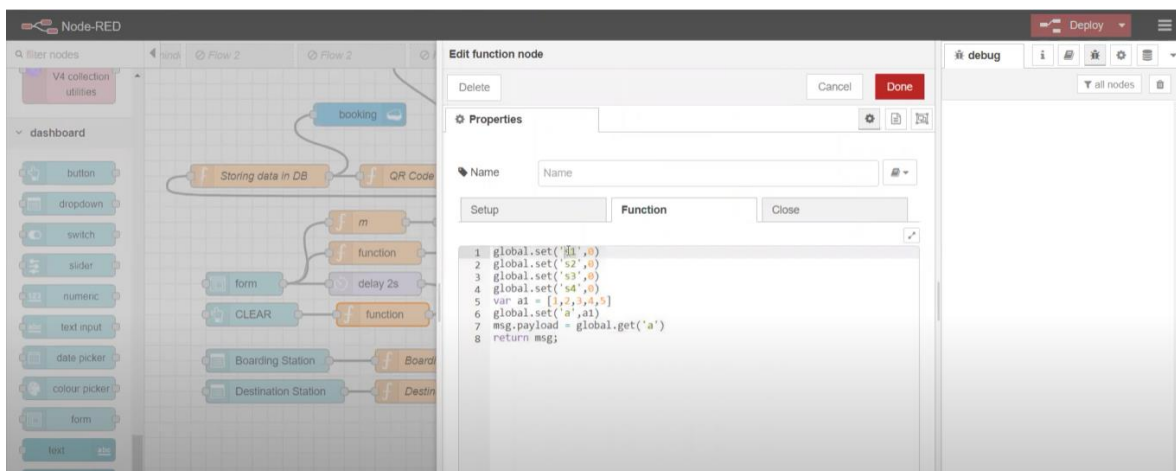
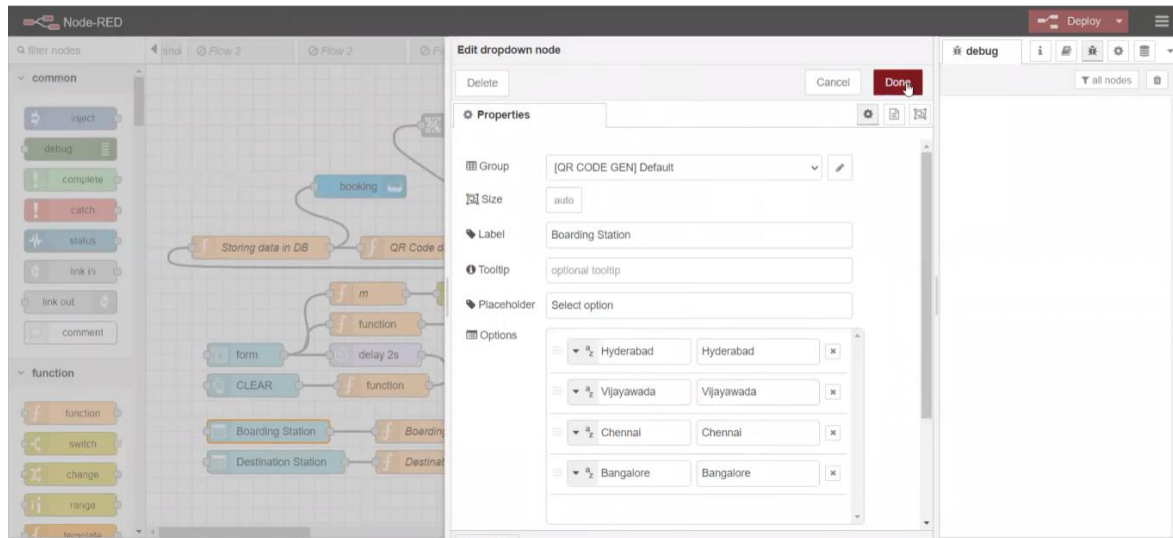


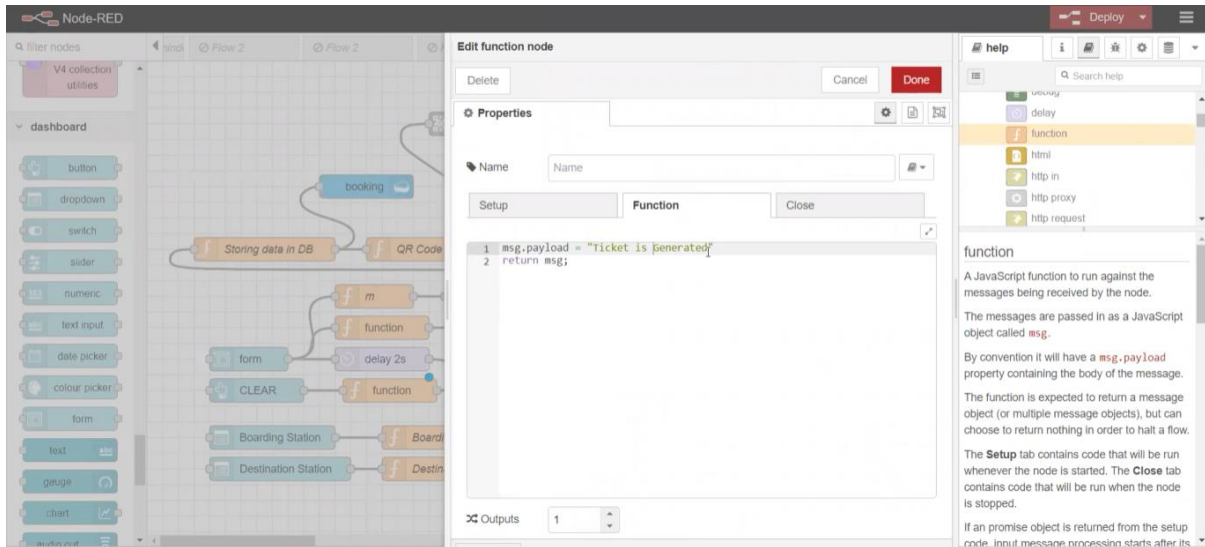
The screenshot shows the "QR CODE GEN" form in the application. The form has a blue header with the title "QR CODE GEN". Below the header, there are four input fields:

- Boarding Station: Hyderabad
- Destination: Vijayawada
- Seat: Select option
- Name: (empty)
- Age: (empty)
- Mobile No: (empty)

At the bottom of the form, there are two buttons: "SUBMIT" and "CANCEL".

Step 2: After adding nodes add functions for nodes in node-red editor





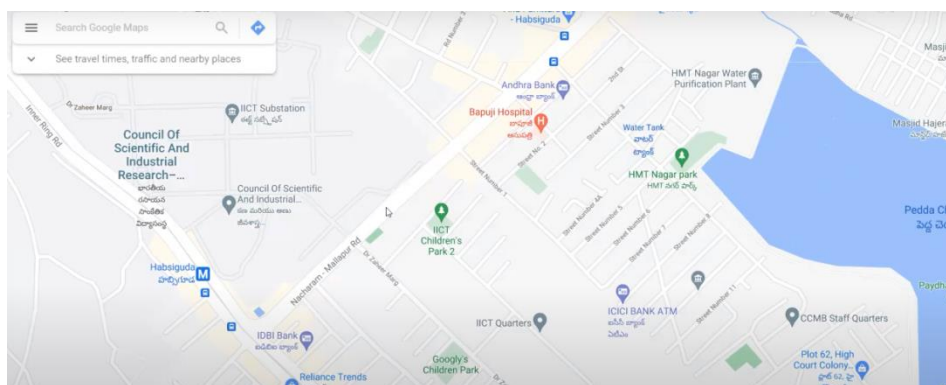
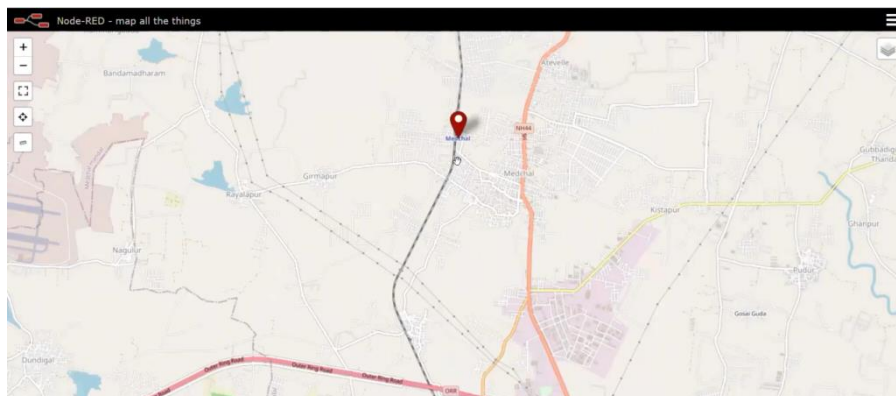
9.RESULTS

9.1 Performance Metrics

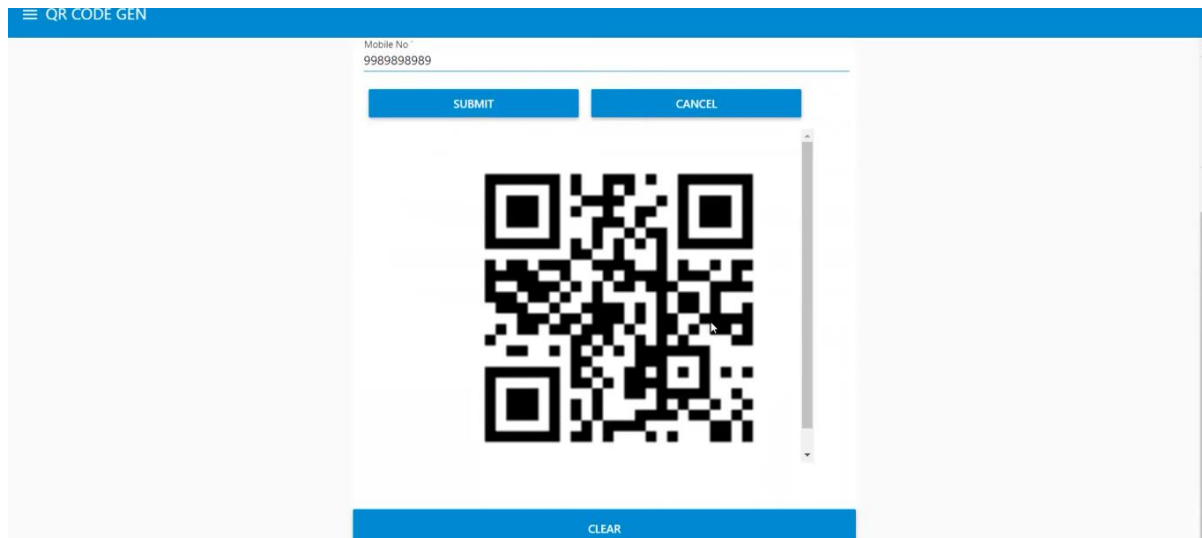
Develop A Web Application Using Node-RED Service

Testing The Web UI By Giving The Required Inputs

Viewing world Map for Train location



After Booking ticket QR Code generation is Completed



10. Advantages and disadvantages

Advantages:

- Rail transport can be cost effective. ...
- Shipping via train is more environmentally friendly. ...
- Trains are capable of hauling large loads. ...
- Railways are reliable. ...
- Rail freight can be efficient. ...
- Rail options provide you with access to capacity.

Disadvantage:

- Huge Capital Outlay
- Lack of Flexibility
- Lack of Door to Door Service
- Monopoly
- Unsuitable for Short Distance and Small Loads
- Booking Formalities
- No Rural Service
- Under-utilised Capacity
- Centralised Administration

11. CONCLUSION

Passengers can benefit greatly from an IoT-based smart Solution for Railways. As a result of the lack of irrigation, railway suffers. Tracking factors such as Latitude, longitude, and seats Availability can be adjusted dependent on the local Train Availability variables. This technology also detects also passenger , which are a major cause of seat selecting. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source. It helps in deciding the traveller to whether to do irrigation or not to do. Continuous internet connectivity is required for continuous monitoring of data from sensors. This also can be overcome by using GSM unit as an alternative of mobile app. By GSM, SMS can be sent to passenger phone.

12.Future scope

In the current project we have implemented the project that can protect and maintain the the train. In this project the train tracking and control the field remotely. In future we can add or update few more things to this project

- . • We can create few more models of the same project ,so that the passenger can have information of a entire.
- We can update the this project by using advanced cloud DB . So that the Dttas of passanger also include and we the passenger's relatives also can their location of passengers. We can add cloud DB technology to this project.
- We can use GPM technology to this project so that the passengers can get the information directly to his home through SMS. This helps the passengers to get information if there is a internet issues.
- We can add security feature so that the passengers can monitor his Train in real time.

This helps in avoiding thefts.

13.Appendix

Source Code :

#Track Location

```
import wiotp.sdk.device
import time
```

```

import random
myConfig = {
    "identity": {
        "orgId": " 64j80w",
        "typeId": "GPS",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

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()

def pub (data):
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    onPublish=None)
    print ("Published data Successfully: %s", myData)

while True:
    myData={'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
    pub (myData)
    time.sleep (3)
    #myData={'name': 'Train2', 'lat': 17.6387448, 'lon': 78.4754336}
    #pub (myData)
    #time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
    pub(myData)
    time.sleep(3)
    myData={'name': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
    pub (myData)
    time.sleep (3)
    client.commandCallback = myCommandCallback

```



```
client.disconnect ()
```

#get QR code

```
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator
```

```
authenticator = BasicAuthenticator('apikey-v2-16u3crmdpkghxefdi
kvpssoh5fwezrmuup5fv5g3ubz', 'b0ab119f45d3e6255eabb978')
service = CloudantV1(authenticator=authenticator)
```

```
service.set_service_url('https://apikey-v2-
16u3ermdpkghxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119145d3e6255eabb978e7e2f0')
```

```
cap= cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_PLAIN
```

```
while True:
```

```
    _, frame = cap.read()
```

```
    decodedObjects = pyzbar.decode (frame)
```

```
    for obj in decodedObjects:
```

```
        #print ("Data", obj.data)
```

```
        a=obj.data.decode('UTF-8')
```

```
        cv2.putText(frame, "Ticket", (50, 50), font, 2,          (255, 0, 0), 3)
```

```
        #print (a)
```

```
        try:
```

```
            response = service.get_document(
```

```
                db='booking',
```

```
                doc_id = a
```

```
            ).get_result()
```

```
            print (response)
```

```
            time.sleep(5)
```

```
        except Exception as e:
```

```
            print ("Not a Valid Ticket")
```

```
            time.sleep(5)
```

```
cv2.imshow("Frame",frame)
```

```
if cv2.waitKey(1) & 0xFF ==ord('q'):
```

```
    break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

```
client.disconnect()
```

Github link : <https://github.com/IBM-EPBL/IBM-Project-3430-1658561082>