INDEX

Project Report

Name: SMART SOLUTIONS FOR RAILWAYS

Team ID: PNT2022TMID44878

College: CARE College of Engineering

Team: Nishanth.K

Vijei.R

Citybabu.M

Dhinakaran.S

1. INTRODUCTION

1.1 Project Overview

As trains are one of the most preferred modes of transportation among middle class and impoverished people as it attracts for its amenities. Simultaneously there is an increase at risk from thefts and accidents like chain snatching, derailment, fire accident. In order to avoid or in better words to stop all such brutality we came up with a solution by providing an application which can be accessed by the user after booking their tickets. With a single click this app addresses issues by sending a text message to TC and RPF as an alert. In our project we use Node-Red service, IBM cloud platform to store passenger data.

1.2 Purpose

The purpose of this project is to report and get relived from the issues related to trains.

2. LITERATURE SURVEY

2.1 Existing problem

- A Web page is designed for the public where they can book tickets by seeing the available seats.
- After booking the train, the person will get a QR code which has to be shown to the Ticket Collector while boarding the train.
- The ticket collectors can scan the QR code to identify the personal details.
- A GPS module is in the train to track it. The live status of the journey is updated in the Web app continuously

• All the booking details of the customers will be stored in the database with a unique ID and they can be retrieved back when the Ticket Collector scans the QR Code

2.2 References

S.NO	TITLE	AUTHOR	YEAR	KEY TECHNOLOGY
1.	Problems of Indian Railways	Benjamin	2021	Common problems in Indian railways
2.	Construction and Building Materials	Sañudo, Roberto, Marina Miranda, Carlos García, and David GarcíaSanchez	2019	Drainage in railways
3.	Main geotechnical problems of railways and roads in kriolitozone and their solutions	Kondratiev, ValentinG	2017	Main problems in railways
4.	A comparative study of Indian and worldwiderailways	Sharma, Sunil Kumar, and AnilKumar	2014	Study of Indian railways
5.	Ticketing solutions for Indian railways using RFID technology	Prasanth, Venugopal, and K.P. Soman	2009	Solution for ticketing using RFID

2.3 Problem Statement Definition

Smart Solutions for railways are designed to reduce the work load of the user and the use of paper.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

Figure-1

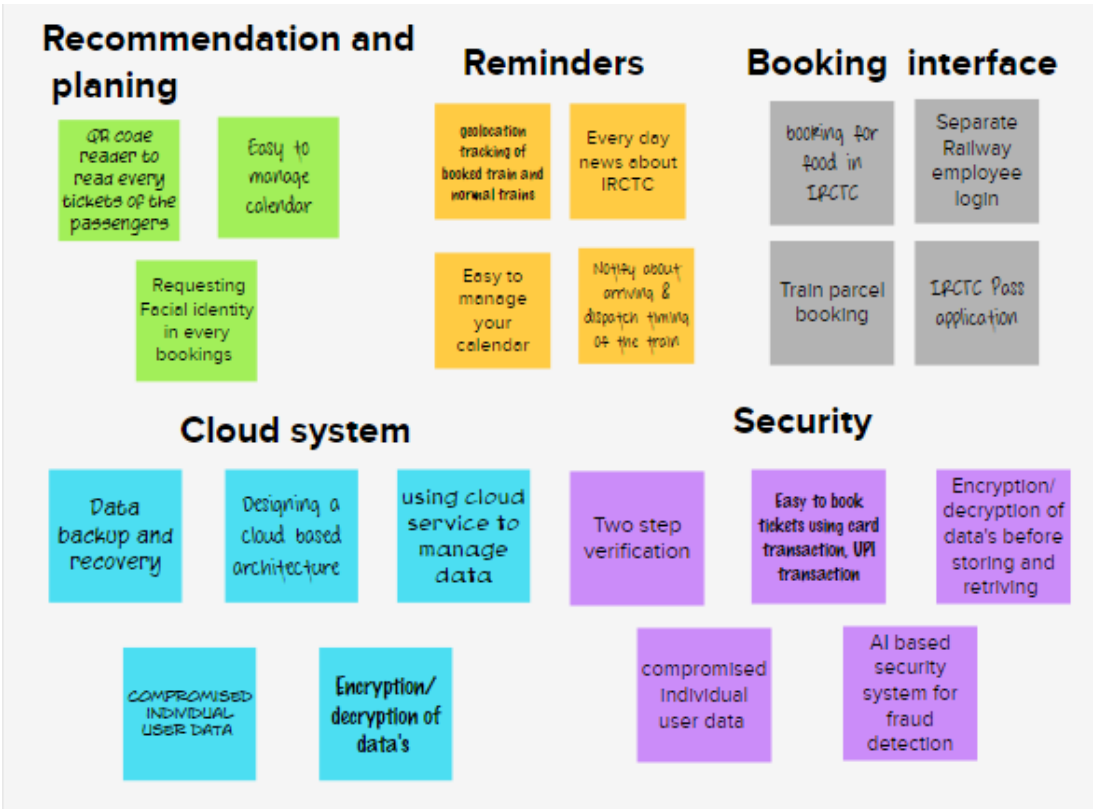


Figure-2

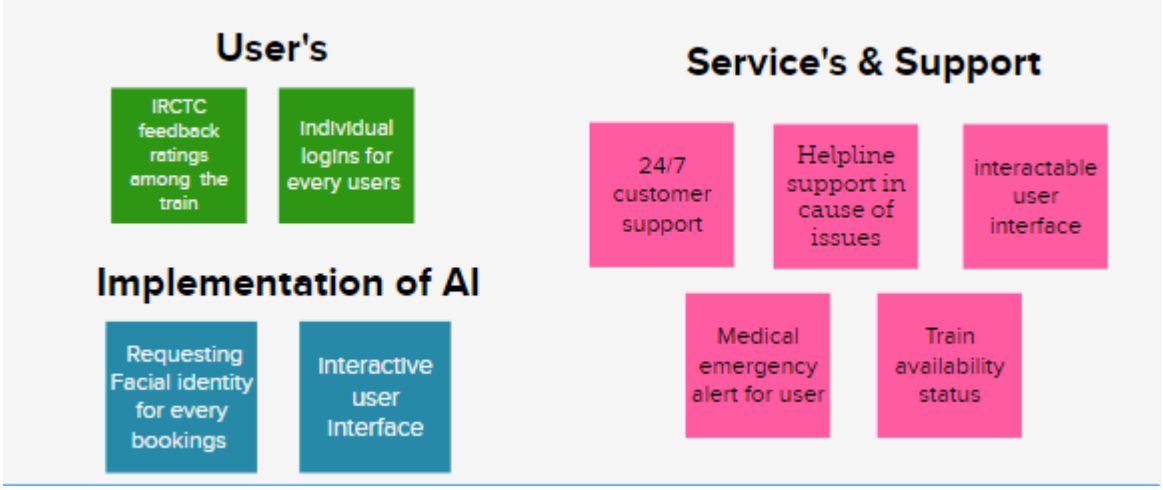


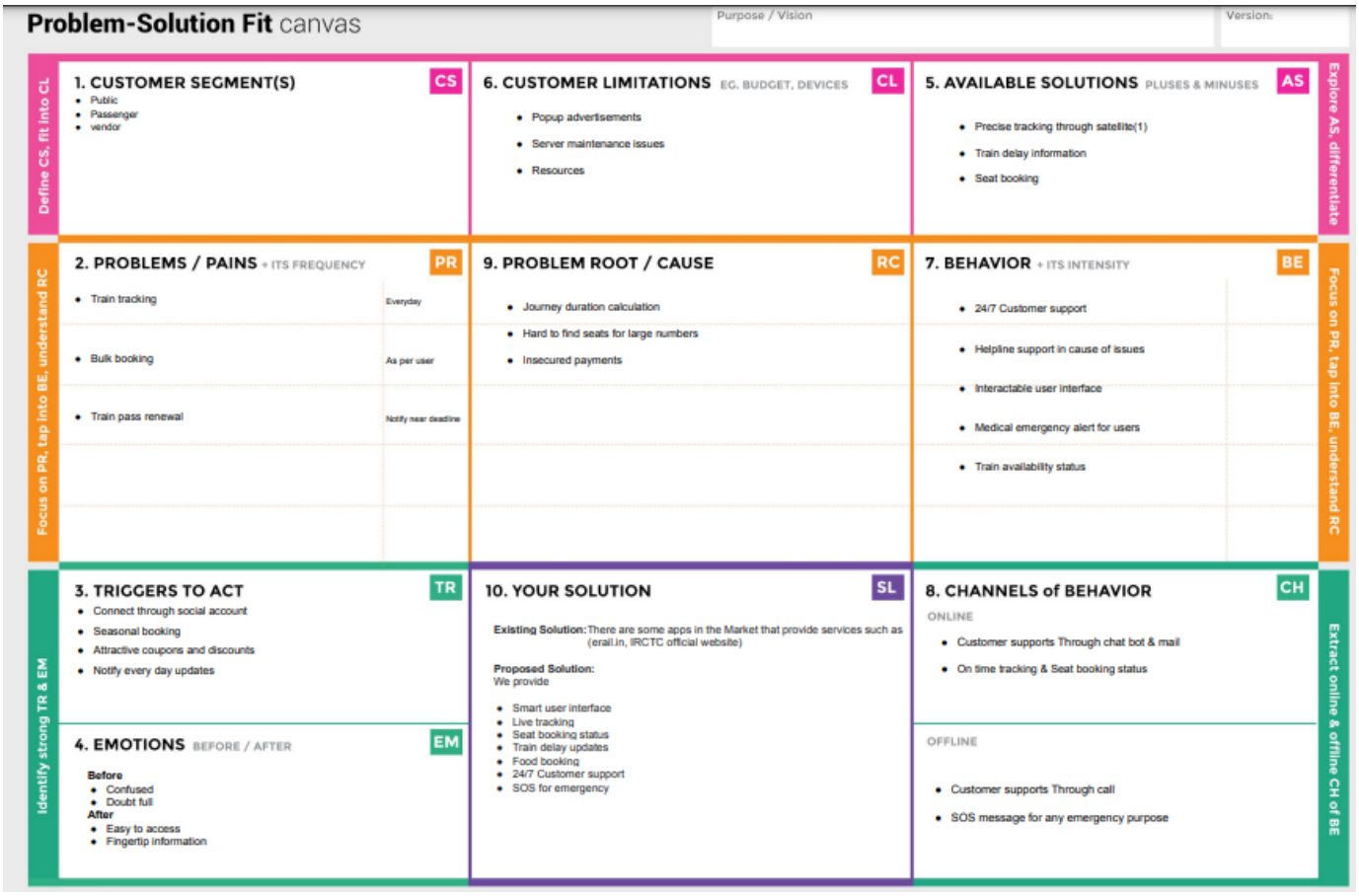
Figure-3

Figure-4

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Web application based on IOT for efficient railway system based on user requirements and security
2.	Idea / Solution description	1)QR code reader to read every tickets of the passengers. 2)Requesting Facial identity in every bookings
3.	Novelty / Uniqueness	1)Interactable user interface 2) QR code, Virtual assistance
4.	Social Impact / Customer Satisfaction	1)Easy to access 2)More confidential 3)Better customer experience
5.	Business Model (Revenue Model)	1)Online payments 2)connect through social account
6.	Scalability of the Solution	1)Train tickets 2)Train availability

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

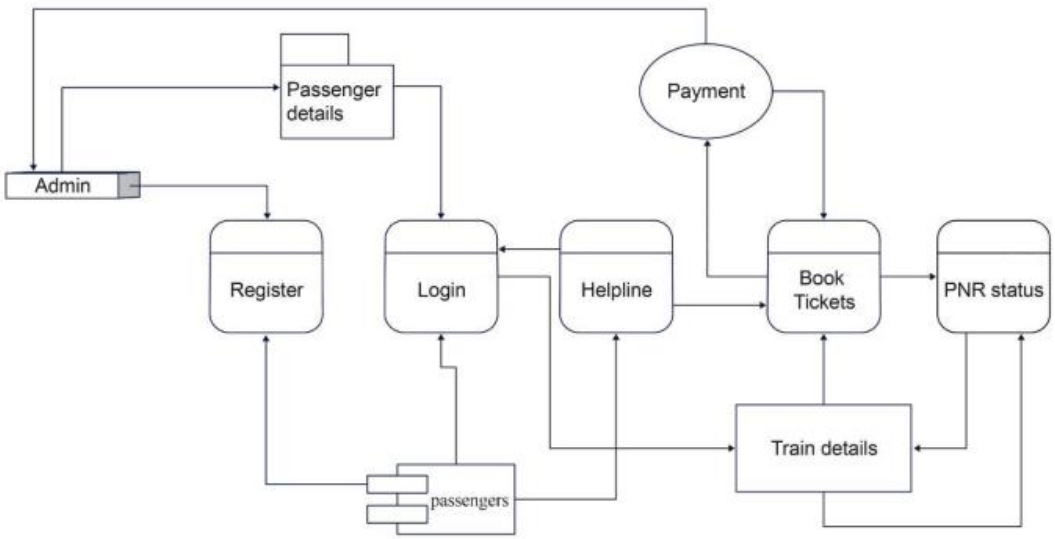
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Web page
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User QR code generation	QR code generated
FR-4	GPS tracking	Live location

4.2 Non-Functional requirements

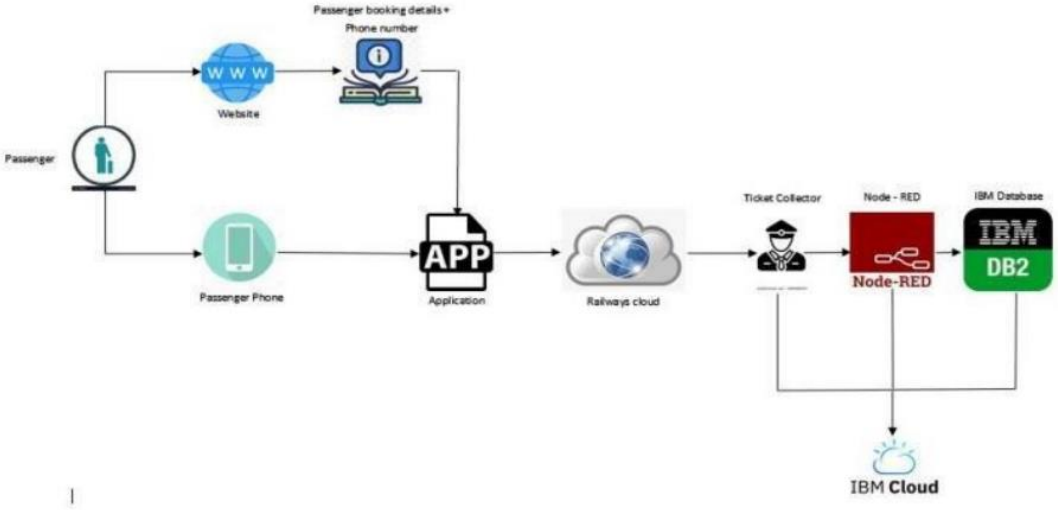
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User can navigate easily
NFR-2	Security	The user detail are secured in cloud
NFR-3	Reliability	Reliable to the user without any failure as it is not fixed to limited number of users
NFR-4	Performance	User friendly
NFR-5	Availability	At any time
NFR-6	Scalability	Support the user with their need in ticket booking & tracking the train live location

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

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, password, and confirming my password	We can access my account / dashboard	High	Sprint-1
Customer (Mobile user)	Reserving tickets	USN-2	As a user, I will receive confirmation email once I have registered for the application	We can receive confirmation email & click confirm	High	Sprint-1
Customer (Mobile user)	Reserving tickets	USN-3	As a user, I can register for the application through Facebook	We can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Mobile user)	Dashboard	USERS	As a user, I can register for the application through Facebook	We can access it using database	Medium	Sprint-3
Customer (Mobile user)	Reserving tickets	USER	Enter the details & book the tickets easily	We can use the QR code which is been generated	High	Sprint-1

Customer (Mobile user)	Connecting the service provider	Customer	Connects with the service by logging in	Can get connected with the server	Medium	Sprint-3
Administrator	Provides the services	Admin	The data is given by the user	Can add or upload the data provided by the user	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	USN-1	Create the IBM Cloud services which are being used in this project.	6	High	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-1	USN-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-1	USN-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	5	Medium	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-1	USN-4	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.	5	High	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-2	USN-1	Configure the connection security and create API keys that are used in the Node -RED service for accessing the IBM IoT Platform.	10	High	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-2	USN-2	Create a Node -RED service	10	High	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-3	USN-1	Develop a python script for publishing the location (latitude and longitude) data to the IBM IoT Platform and the other python code to read the QR Code and fetch the data from Cloudant DB.	20	High	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S

Sprint-4	USN-1	Develop the web application using Node-RED	10	Medium	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S
Sprint-4	USN-2	Testing the Web UI by giving the required inputs	10	High	Nishanth.K Vijei.R Citybabu.M Dhinakaran.S

6.2 Sprint Delivery Schedule

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

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

7.1 Feature 1

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- Geofence
- Python code

7.2 Feature 2

- Registration
- Login
- OTP Verification
- Ticket Booking

•Payment

•Pantry

•QR code generation

8. TESTING

Test case-1

				Date	03-Nov-22				
				Team ID	PNT2022TMID44878				
				Project Name	Smart solutions for Railways				
				Maximum Marks	4 marks				
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Executed By**
1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or		Login/Signup popup should display	Working as expected	Pass	Vijei.R
2	UI	Generatimg OTP	Generatin thee OTP for further process	Generate the OTP to the Login email ID		User can register through email ID and to get OTP	Working as expected	Pass	Vijei.R
3	Functional	OTP verification	Verify user OTP usin email	1.Enter Valid email in Email text box 2.Enter valid password in password text box 3.Click on login button	Username: trainindian44878@gmail.com password: qwertyuiop12@	OTP verified is to be displayed	Working as expected	Fail	Dhinakaran.S
4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter into login page 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: trainindian44878@gmail.com password: qwertyuiop12@	Application should show 'Incorrect email or password ' validation message.	Working as expected	Fail	Citybabu.M
5	Functional	Display train details	the user can view adout theavailable train details	As a user, I can enter the start and destination to get the list of train available connecting the above	Username: trainindian44878@gmail.com password: qwertyuiop12@	A user can view about the available train to enter start and destination details	Working as expected	Pass	Nishanth,K

Test case-2

				Date	11-Nov-22				
				Team ID	PNT2022TMID44878				
				Project Name	Smart solutions for Railways				
				Maximum Marks	4 marks				
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**	**Executed By**	
1	Functional	Booking	User can provide the basic details such as name,number, etc....	Enter the member's detail like name, number	Ticket booked to be displayed	Working as expected	Pass	Vijei.R	
2	UI	Booking seat	User can choose the train start and destination date of travel.	Know to which the train is available	Know to which the seat are available	Working as expected	Fail	Vijei.R	
3	Functional	Payment	User, I can choose to pay through credit card/debit card/UPI	1.User can choose payment method 2.payment method	Payment for the booked tickets to be done using payment method through either the following methods credit card / debit	Working as expected	Fail	Dhinakaran.S	
4	Functional	Redirect	User can be redirected to the selected	After payment the user will be redirected to the previous page	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Citybabu.M	

Test case-3

				Date	14-Nov-22			
				Team ID	PNT2022TMID44878			
				Project Name	Smart solutions for Railways			
				Maximum Marks	4 marks			
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**	**Executed By**
1	Functional	Ticket generation	A user can download the generated e-ticket for my journey along with the QR code which is used for authenticationduring my journey	1.Enter method of reservation 2.Enter name, age, gender 3Enter how many ckets wants to be booked. 4.Also enter the number of member's details	Ticket booked to be displayed	Working as expected	Pass	Nishanth.K
2	UI	Ticket status	A user can see the staus of my ticket weather itsconfirmed / waiting / RAC	Generate the OTP to the Login email ID	Known to the status of the ticket booked	Working as expected	Fail	Vijei.R
3	Functional	Reporting issues	User can access the reporting portal once the journey begins	1.Enter Valid email in Email text box 2.Enter valid password in password text box 3.Click on login button	Issues have been reported	Working as expected	Pass	Nishanth.K

Shopenzer Testcases | Testscearnios

Test case-4

				Date	18-Nov-22			
				Team ID	PNT2022TMID44878			
				Project Name	Smart solutions for Railways			
				Maximum Marks	4 marks			
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**	**Executed By**
1	Functional	Ticket cancellation	User can cancel their ticket any change of plans	Ticket to be cancel	Tickets booked to be cancelled	Working as expected	Fail	Dhinakararn.S
2	Functional	Rate	A user will feed rating about the train journey	Information feeding on trains	Information feeding on trains	Working as expected	Fail	Citybabu.M

9. ADVANTAGES

- The passengers can use this application, while they are travelling alone to ensure their safety.
- It is easy to use.
- It has minimized error rate.

10. DISADVANTAGES

- Network issues may arise.

11. CONCLUSION

Accidents occurring in Railway transportation system cost a large number of lives. So, this system helps us to prevent accidents and giving information about faults or cracks in advance to railway authorities. So that they can fix them and accidents cases becomes less. This project is cost effective. By using more techniques, they can be modified and developed according to their applications. By this system many lives can be saved by avoiding accidents. The idea can be implemented in large scale in the long run to facilitate better safety standards for rail tracks and provide effective testing infrastructure for achieving better results in the future.

12. FUTURE SCOPE

This application is ensured for safety for the passengers while they are travelling alone as well as they travel with their family or friends. In future, this application may also be used by passengers who travel through bus. By further enhancement of the application the passengers can explore more features regarding their safety.

13. APPENDIX

13.1 Source Code

python code for publishing the location (latitude and longitude).

```
import wiotp.sdk.device
import time
from tkinter import *
root=Tk()
root.geometry('400x200')
# Provide your IBM Watson Device Credentials
myConfig = {    "identity": {                "orgId": "ie9ki3",
                        "typeId": "mydevice",
                        "deviceId":"mydeviceid"},
          "auth": {                    "token": "bW(_2O({aRG8E6fij6"}}
myData1 = {'name': 'EXPRESS', 'lat': 13.913128, 'lon': 79.360651}
myData2 = {'name': 'EXPRESS', 'lat': 13.729034,'lon': 79.472997}
myData3 = {'name': 'EXPRESS', 'lat': 13.478878, 'lon': 79.541901}
myData4 = {'name': 'EXPRESS', 'lat': 13.216907, 'lon': 79.592364}
myData5 = {'name': 'EXPRESS', 'lat': 13.093835, 'lon': 79.683645}
```

```
myData6 = {'name': 'EXPRESS', 'lat': 13.128028, 'lon': 79.932913}
array1=[myData1,myData2,myData3,myData4,myData5,myData6]
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=data, qos=0, onPublish=None)
   print("Published data Successfully: %s", data)
def button():
   text=data.get()
   print(text)
   while True:
      if text == "IND001": ### mumbai to chennai
         for i in array1:
            pub(i)
            time.sleep(3)
      elif text=="IND002": ### chennai to mumbai
         for j in reversed(array1):
            pub(j)
            time.sleep(3)
         client.commandCallback = myCommandCallback
      client.disconnect()
data = Entry(root ,width=20)
data.pack()

Button(root, text="SUBMIT", command=button).pack()




root.mainloop()
```

QR Code and fetch the data from CloudantDB.

```
import cv2
import time
import pyzbar.pyzbar as pyzbar
from ibmcloudant.cloudant_v1 import
CloudantV1
from ibm_cloud_sdk_core.authenticators
import BasicAuthenticator
import tkinter as tk
authenticator =
BasicAuthenticator('apikey-
5fa841dab9544e31b6a1b6f9ba432422',
'591f8ef66aac6f9bc779c8e7bf4a670f4058c
f13')
service =
CloudantV1(authenticator=authenticator)
service.set_service_url('https://c14dc572-
82d6-4b8b-9e6f-01fc8aedecb0-
bluemix.cloudant.com')
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)
        try:
            response =
service.get_document(db='crendentials',do
c_id=a).get_result()
            print(response)
            NAME1 =response['Name']
            NAME = "NAME:" + " " +
NAME1
            #time.sleep(0.5)
            AGE1 = response['Age']
            AGE = "AGE:" + " " + AGE1
            #time.sleep(0.5)
            GENDER1 = response['GENDER']
            GENDER = "GENDER:" + " " +
GENDER1
            #time.sleep(0.5)
            MOBILE1 = response['Mobile']
```

```
        MOBILE_NO = "MOBILE:" + " "
+ MOBILE1
        #time.sleep(0.5)
        DATE1 = response['DATE']
        DATE = "DATE:" + " " + DATE1
        #time.sleep(0.5)
        TIME1 = response['TIME']
        TIME = "TIME:" + " " + TIME1
        #time.sleep(0.5)
        BOARDING1 =
response['Boarding']
        BOARDING = "BOARDING:" + "
" + BOARDING1
        #time.sleep(0.5)
        DESTINATION1 =
response['Destination']
        DESTINATION =
"DESTINATION:" + " " +
DESTINATION1
        SEAT1 = str(response['Seat'])
        SEAT = "SEAT:" + " " + SEAT1



        root = tk.Tk()
        root.geometry('900x900')
        l = tk.Label(root, text="
PASSENGER DETAILS", font='Helvetica
14 bold')
        l.pack()
        l.place(width=900, height=40)
        k = tk.Label(root,
                text=NAME+"\n"+AGE+"\
n"+GENDER+"\n"+MOBILE_NO+"\n"+S
EAT+"\n"+DATE+"\n"+TIME+"\n"+BOA
RDING+"\n"+DESTINATION,
                font='Helvetica 11 bold')
        k.pack()
        k.place(width=300, height=350)

        root.mainloop()
```

```
            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()
```

13.2 GitHub

https://github.com/IBM-EPBL/IBM-Project-45038-1660727979

13.3 Project Demo Link

https://drive.google.com/file/d/1YV2rS4ppiqk7_E_BPClGuHCXYiKyFxrk/view?usp=share_link