Project Report

INTRODUCTION

1. INTRODUCTION

1.1 PROJECT OVERVIEW

SMART SOLUTIONS FOR RAILWAYS is to manage Indian Railways is the largest railway network in Asia and additionally world's second largest network operated underneath a single management. Due to its large size it is difficult to monitor the cracks in tracks manually. This paper deals with this problem and detects cracks in tracks with the help of ultrasonic sensor attached to moving assembly with help of stepper motor. Ultrasonic sensor allows the device to moves back and forth across the track and if there is any fault, it gives information to the cloud server through which railway department is informed on time about cracks and many lives can be saved. This is the application of IoT, due to this it is cost effective system. This effective methodology of continuous observation and assessment of rail tracks might facilitate to stop accidents. This methodology endlessly monitors the rail stress, evaluate the results and provide the rail break alerts such as potential buckling conditions, bending of rails and wheel impact load detection to the concerned authorities.

1.2. PURPOSE

Internet is basically system of interconnected computers through network. But now its use is changing with changing world and it is not just confined to emails or web browsing. Today's internet also deals with embedded sensors and has led to development of smart homes, smart rural area, e-health care's etc. and this introduced the concept of IoT . Internet of Things refers to interconnection or communication between two or more devices without human-to-human and human-to-computer interaction. Connected devices are equipped with sensors or actuators perceive their surroundings. IOT has four major components which include sensing the device, accessing the device, processing the information of the device, and provides application and services. In addition to this it also provides security and privacy of data . Automation has affected every aspect of our daily lives. More improvements are being introduced in almost all fields to reduce human effort and save time. Thinking of the same is trying to introduce automation in the field of track testing. Railroad track is an integral part of any company's asset base, since it provides them with the necessary business functionality. Problems that occur due to problems in railroads need to be overcome. The latest method used by the Indian railroad is the tracking of the train track which requires a lot of manpower and is time-consuming

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

In the Existing train tracks are manually researched. LED (Light Emitting Diode) and LDR (Light Dependent Resister) sensors cannot be implemented on the block of the tracks]. The input image processing is a clamorous system with high cost and does not give the exact result. The Automated Visual Test Method is a complicated method as the video color inspection is implemented to examine the cracks in rail track which does not give accurate result in bad weather. This traditional system delays transfer of information. Srivastava et al., (2017) proposed a moving gadget to detect the cracks with the help of an array of IR sensors to identify the actual position of the cracks as well as notify to nearest railway station . Mishra et al., (2019) developed a system to track the cracks with the help of Arduino mega power using solar energy and laser. A GSM along with a GPS module was implemented to get the actual location of the faulty tracks to inform the authorities using SMS via a link to find actual location on Google Maps. Rizvi Aliza Raza presented a prototype in that is capable of capturing photos of the track and compare it with the old database and sends a message to the authorities regarding the crack detected. The detailed analysis of traditional railway track fault detection techniques is explained in table

2.2 REFERENCES

1. D. Hesse, “Rail Inspection Using Ultrasonic Surface Waves” Thesis, Imperial College of London, 2007.

2. Md. Reya Shad Azim1 , Khizir Mahmud2 and C. K. Das. Automatic railway

track switching system, International Journal of Advanced Technology, Volume 54, 2014.

3. S. Somalraju, V. Murali, G. saha and V. Vaidehi, “Title-robust railway crack detection scheme using LED (Light Emitting Diode) - LDR (Light Dependent Resistor) assembly IEEE 2012.

4. S. Srivastava, R. P. Chourasia, P. Sharma, S. I. Abbas, N. K. Singh, “Railway Track Crack detection vehicle”, IARJSET, Vol. 4, pp. 145-148, Issued in 2, Feb 2017.

5. U. Mishra, V. Gupta, S. M. Ahzam and S. M. Tripathi, “Google Map Based Railway Track Fault Detection Over the Internet”, International Journal of Applied Engineering Research, Vol. 14, pp. 20-23, Number 2, 2019.

6. R. A. Raza, K. P. Rauf, A. Shafeeq, “Crack detection in Railway track using Image processing”, IJARIIT, Vol. 3, pp. 489-496, Issue 4, 2017.

7. N. Bhargav, A. Gupta, M. Khirwar, S. Yadav, and V. Sahu, “Automatic Fault Detection of Railway Track System Based on PLC (ADOR TAST)”, International Journal of Recent Research Aspects, Vol. 3, pp. 91-94, 2016

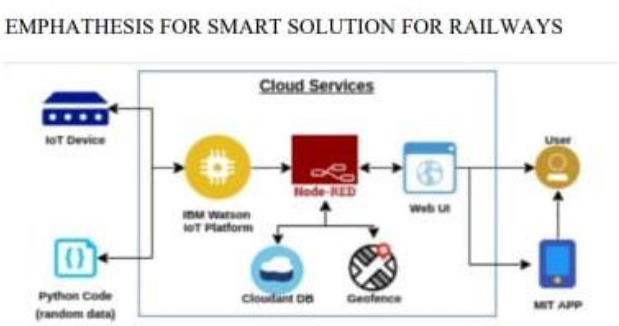
2.3 PROBLEM STATEMENT DEFINITION

Among the various modes of transport, railways is one of the biggest modes of transport in the world. Though there are competitive threats from airlines, luxury buses, public transports, and personalized transports the problem statement is to answer the question “What are the problems faced by the passengers while travelling by train at station and on board”

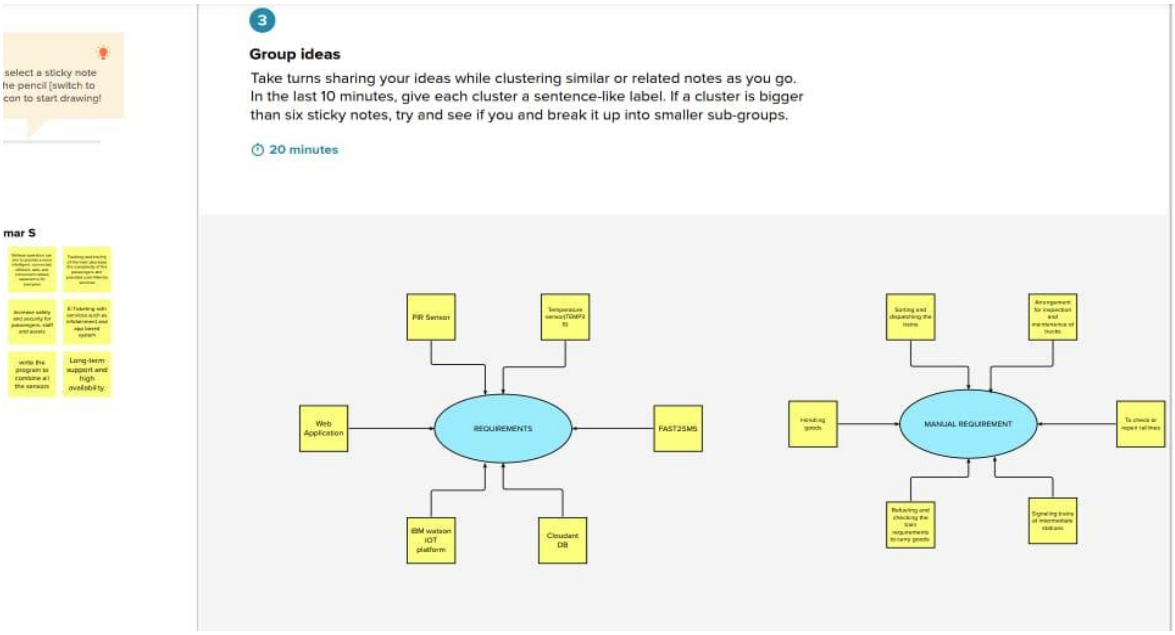
IDEATION AND PROPOSED SOLUTION

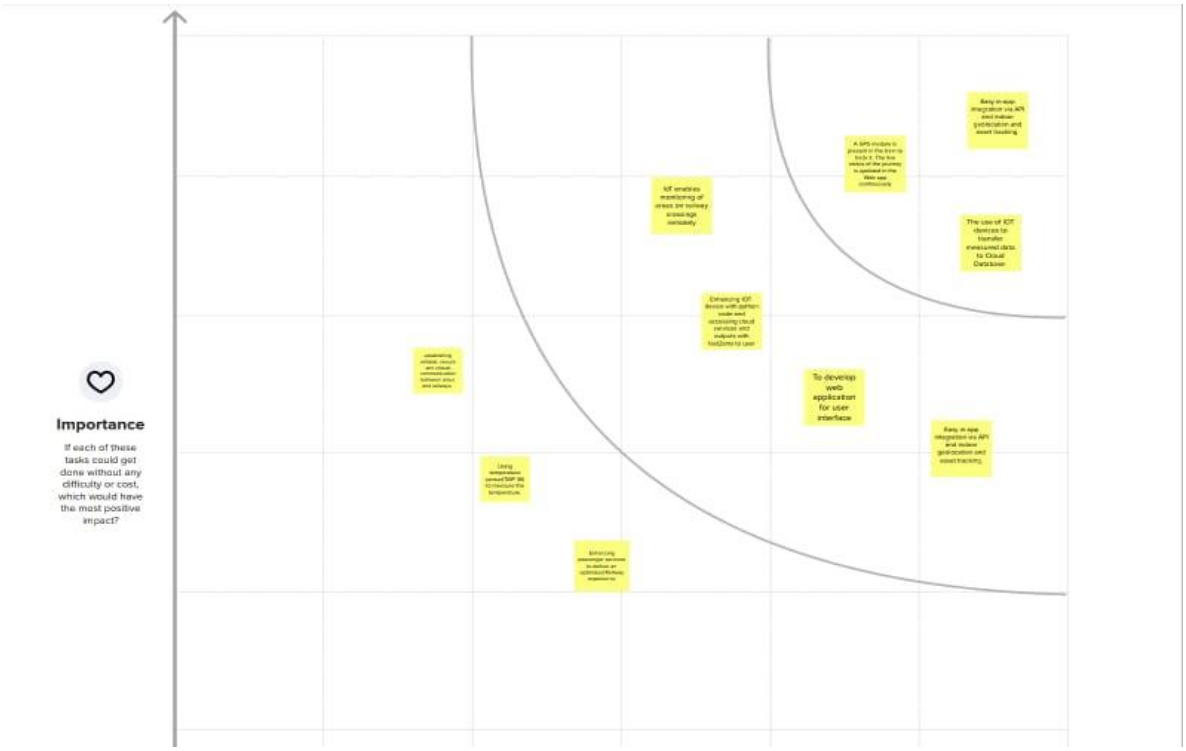
3. IDEATION AND PROPOSED SOLUTON

3.1 EMPATHY MAP CANVAS



3.2 BRAINSTORMING


select a sticky note
he pencil (switch to
con to start drawing!
mar S
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go.
In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
PIR Sensor
Temperature sensor(TEMP2 11)
Web Application
REQUIREMENTS
FAST2SMS
IBM watson IOT platform
Cloudant DB
Sorting and dispatching the trains
Arrangement for inspection and maintenance of trains
Handing goods
MANUAL REQUIREMENT
To check at repair railways
Refueling and checking the train requirements to carry goods
Signaling trains at intermediate stations


Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
To develop web application for user interface

PROJECT DESIGN PHASE

4.1 Problem Solution Fit

4.2 Proposed Solution

Project Design Phase-I Proposed Solution Template

Date	29 September 2022
Team ID	PNT2022TMID08784
Project Name	Project-Smart Solution for Railways
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Creating an application for developing app for GPS tracking and ticket booking. In this mode of application we have planned to accommodate with IBM Watson platform, node red, WEB UI, cloudant DB these are the platform used for storing the database. Finalizing with QR code for making it more easier
2.	Idea / Solution description	Even with the greatest ideas to overcome solutions for railway ticket generating this is a time consuming process for checking the tickets generated. Even much resources available fraudulent may not be rectified. This project provides the solutions generating the QR code for tickets and verified easily
3.	Novelty / Uniqueness	Creating a website and develop into a QR code for easier way for tracking train and updating a location
4.	Social Impact / Customer Satisfaction	Customer can easily track the train location and prepare accordingly.By this way human's can save their time and increasing of machine work.
5.	Business Model (Revenue Model)	Business model makes a major impact on economic level by this project man can improves in technology as well as reduce their expense
6.	Scalability of the Solution	Tracking of train location `app acquires minimum of memory which makes installation easier and eco-friendly with the user

4.3 Solution architecture

SOLUTION ARCHITECTURE

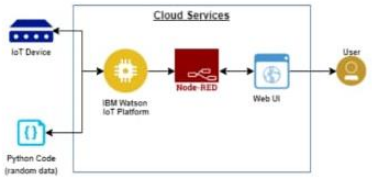
Design:

- Even with the greatest ideas to overcome solutions for railway ticket gathering this time consuming process for checking the tickets generated.
- Even much resource available fraudulent may not be rectified
- This project provides the solution generating the QR code for tickets and verifies easily.
- In this project IOT device is connected and a python random code is generated and cloud services include(IBM Waston Platform, Node RED, WEB UI, Cloudant DB which stores the database of the applied tickets)
- All booking details of customer is stored in the cloud database with an unique ID and they can be retrieved back when the ticket collector scans the QR code.
- Finally architecture connects with user and Fast2SMS application to send message to user.
- A GPS module is present in the train to track.
- The live status of the train is uploaded in the web app continuously

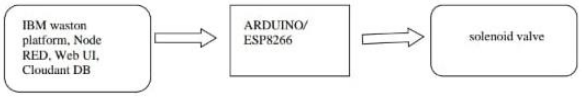
Software and system required:

- Arduino IDE
- Embedded C
- 4GB processor and OS-Windows/Linux/MAC

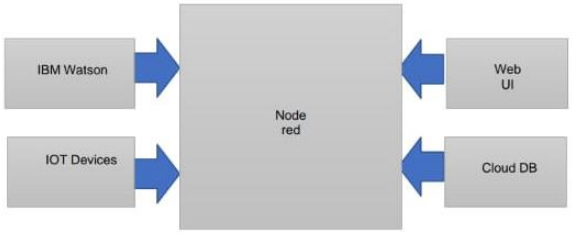
Block diagram:



IOT Device



Cloud Service



PROJECT DESIGN PHASE II

5.1 Customer Journey

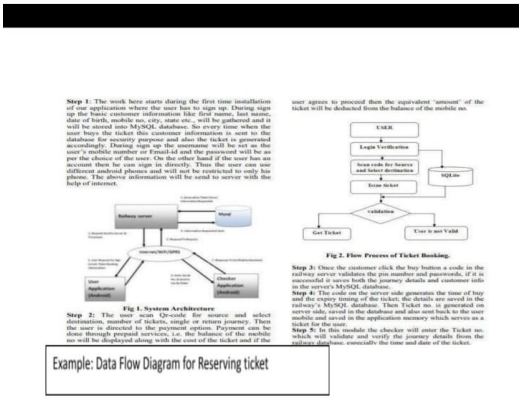
PHASES	Motivation	Information gathering	Analyzes various products	Choose the efficient product	Payment
Actions	Electronic display of tickets is used and encouraged in order to reduce the colossal wastage of paper.	Search for the train and Seat availability	Customer chooses the preferred seat	The chosen seat is reserved and asked for confirmation.	Makes the Payment for the prefered Ticket.
Touch Point	The passengers and most importantly environmentalists would be excited and welcome this new system whole heartedly.	If dedicated crime teams is being set up, then travelers would don't have to worry about their confidentiality.	The user is entertained by a variety of new possibilities offered.	As every bank transaction is also digitalized nowadays, the passengers won't have to worry about the safety of their wallets and tickets.	Since QR code of ticket is directly sent to the travelers and that's enough for the travel, travelers would welcome this system.
Customer Feeling					
Customer Thoughts	Easy handling and support all operating system	Less complexity for searching the seat availability	Show the available seats closest to the preference.	A web application with simple interface	Availability of numerous payment options and simple process
Opportunities	Travel experience for the passengers and travels can be improved if ticketing system is digitalized.	After installation, customers could have a complete track on their travel history and there are some other unclear benefits with security risks	Because of this system, travelers would be aware of the various booking sites available instead of relying on websites issuing paper tickets that's existing now.	One of the advantages of using QR codes is that it facilitates instant payments. Applications installed with QR would have a interesting user experience.	Ticket booking and verifying process would be fastened by implementing this application.

5.2 Data flow diagram

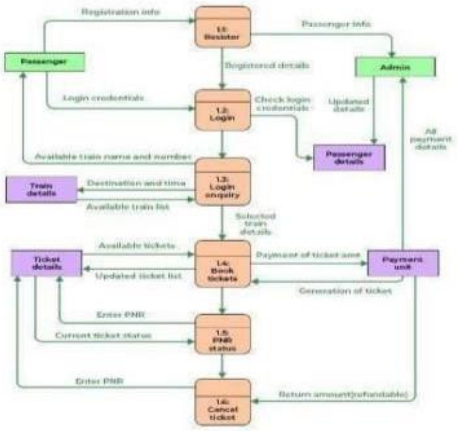
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a s
right amount of the system requirement graphically. It shows how data enters and leaves the syst
data is stored.

2 / 4



Example: Data Flow Diagram for Reserving ticket



User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Reserving ticket	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Mobile user)	Reserving ticket	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Mobile user)	Reserving ticket	USN-3	As a user, I can register for the application and enter the details for reserving the ticket.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Mobile user)	Dashboard	Users	The details will be stored safely	I can access it using database	Medium	Sprint-3
Customer (Web user)	Reserving ticket	User	Enter the details and click submit button to book ticket	I can use the QR code which is been generated	High	Sprint-1
Customer Care Executive	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 update the data provided by the user	High	Sprint-1

5.3 Solution requirement

The Following are the functional requirements of the proposed solution.

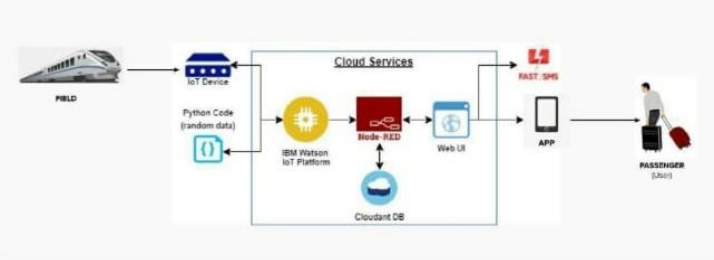
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Requirements**	1.Mobile Phone
		2.Internet
		3.QR Code Scanner
FR-2	**User Registration**	1.Manual Registration
		2.Registration through web page
		3.Registration through Application
FR-3	**User Confirmation**	1.Confirmation via Phone.
		2.Confirmation via Email.
		3.Confirmation via OTP.
		4.Confirmation via SMS.
FR-4	**Payment Options**	1.Net Banking/UPI.
		2.Credit/Debit/ATM Card.
		3.Digital Wallet.
FR-5	**Application**	1.Free Installation via Play Store and App store.
	Installation	2.Website is available for free and will function always.
FR-6	**Application Feedback**	1.Through Web page
		2.Through Phone calls

Non-Functional Requirement

The Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	1.Have a Simple and Efficient application demo
		Video.
		2.Easier to use.
		3.If a Traveller has a Mobile Phone,they may
		easily Understand the procedure and make
		Reservations.
NFR-2	**Security**	1.Two-step authorization is required to secure the application.
		2.Username and password will be assigned in accordance with
		user requirements.
NFR-3	**Reliability**	1.Periodic updates should be made to websites and
		applications.
		2.If the booking process is interrupted by an internet outage,
		we offer an offline mode to complete the detail process.
NFR-4	**Performance**	1.The user interface of the web application must be
		user-friendly.
		2. Payment methods should be quick and easy.
NFR-5	**Availability**	1.Provided with the proper train location.
		2.Databases are maintained for passenger history.
		3.Anytime and Anywhere for online ticket booking

5.4 Technology stark



PROJECT PLANNING PHASE

6.1 Project Milestone

6.2 Sprint Delivery

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	15 Oct 2022
Team ID	PNT2022TMID08784
Project Name	Project – smart solutions for railways
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register through the form by Filling in my details	2	High	Swetha
Sprint-1		USN-2	As a user, I can register through phone numbers, Gmail, Facebook or other social sites	1	High	Vimal
Sprint-1	Conformation	USN-3	As a user, I will receive confirmation through email or OTP once registration is successful	2	Low	Suganthan
Sprint-1	login	USN-4	As a user, I can login via login id and password or through OTP received on register phone number	2	Medium	Manoj
Sprint-1	Display Train details	USN-5	As a user, I can enter the start and destination to get the list of trains available connecting the above	1	High	Swetha
Sprint-2	Booking	USN-6	As a use, I can provide the basic details such as a name, age, gender etc....	2	High	Vimal
Sprint-2		USN-7	As a user, I can choose the class, seat/berth. If a preferred seat/berth isn't available I can be allocated based on the availability	1	Low	Suganthan
Sprint-2	Payment	USN-8	As a user, I can choose to pay through credit Card/debit card/UPI.	1	High	Manoj

[redacted]

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-9	As a user, I will be redirected to the selected	2	High	Swetha
Sprint-3	Ticket generation	USN-10	As a user, I can download the generated e-ticket for my journey along with the QR code which is used for authentication during my journey.	1	High	Vimal
Sprint-3	Ticket status	USN-11	As a user, I can see the status of my ticket Whether it's confirmed/waiting/RAC.	2	High	Suganthan
Sprint-3	Remainders notification	USN-12	As a user, I get remainders about my journey A day before my actual journey.	1	High	Manoj
Sprint-3	Ticket cancellation	USN-13	As a user, I can track the train using GPS and can get information such as ETA, Current stop and delay	2	High	Swetha
Sprint-4		USN-14	As a user, I can cancel my tickets if there's any Change of plan	1	High	Vimal
Sprint-4	Raise queries	USN-15	As a user, I can raise queries through the query box or via mail.	2	Medium	Suganthan
Sprint-4	Answer the queries	USN-16	As a user, I will answer the questions/doubts Raised by the customers.	2	High	Manoj
Sprint-4	Feed details	USN-17	As a user, I will feed information about the trains delays and add extra seats if a new compartment is added.	1	High	Swetha

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	5 Nov 2022

[redacted]

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-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 Nov2022

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)

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

CODING

7.1 Location Data

```
        import wiotp.sdk.device
import time
import random
myConfig={
"identity": {
      "orgId": "2oab6l",
      "typeId": "GPS_Tracking",
      "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)
   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': 'Trainl', 'lat': 17.6340889, 'lon': 78.4745052 }
  pub (myData)
  time.sleep(3)
  myData={'name': 'Trainl', 'lat': 17.6248626, 'lon': 78.4720259}
```

```
    pub (myData)
    time.sleep (3)
    myData={'name': 'Trainl', '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()
```

7.2 QR

```
 Import cv2
import numpy as np
import time
Import pyzbar.pyzbar as pyzbar
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_ sdk_core.authenticators import BasicAuthenticator
authenticator= BasicAuthenticator ('apikey-v2-16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz',
'b0ab119f45d3e6255eabb978
service Cloudant V1 (authenticator-authenticator)
service.set_service_url('https://apikey-v2-16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119
f45d3e6255eabb978e7e2f0
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()
```

7.3 Feed Information

```
from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

subject = "An email with attachment from Python"
body = "This is an email with attachment sent from Python"
sender_email = "my@gmail.com"
receiver_email = "your@gmail.com"
password = input("Type your password and press enter:")

# Create a multipart message and set headers
message = MIMEMultipart()
message["From"] = sender_email
message["To"] = receiver_email
message["Subject"] = subject
message["Bcc"] = receiver_email  # Recommended for mass emails

# Add body to email
message.attach(MIMEText(body, "plain"))

filename = "document.pdf"  # In same directory as script
```

7.4 Raise Queries

```
import smtplib, ssl
from email.mime.text import MIMEText
```

```
from email.mime.multipart import MIMEMultipart

sender_email = "my@gmail.com"
receiver_email = "your@gmail.com"
password = input("Type your password and press enter:")

message = MIMEMultipart("alternative")
message["Subject"] = "multipart test"
message["From"] = sender_email
message["To"] = receiver_email

# Create the plain-text and HTML version of your message
text = """\
Hi,
How are you?
Real Python has many great tutorials:
www.realpython.com"""
html = """\
<html>
  <body>
    <p>Hi,<br>
       How are you?<br>
       <a href="http://www.realpython.com">Real Python</a>
       has many great tutorials.
    </p>
  </body>
</html>
"""

# Turn these into plain/html MIMEText objects
part1 = MIMEText(text, "plain")
part2 = MIMEText(html, "html")

# Add HTML/plain-text parts to MIMEMultipart message
# The email client will try to render the last part first
message.attach(part1)
message.attach(part2)

# Create secure connection with server and send email
context = ssl.create_default_context()
```

```
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(
        sender_email, receiver_email, message.as_string()
```

7.5 Answer Queries

```
import email, smtplib, ssl

from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

subject = "An email with attachment from Python"
body = "This is an email with attachment sent from Python"
sender_email = "my@gmail.com"
receiver_email = "your@gmail.com"
password = input("Type your password and press enter:")

# Create a multipart message and set headers
message = MIMEMultipart()
message["From"] = sender_email
message["To"] = receiver_email
message["Subject"] = subject
message["Bcc"] = receiver_email  # Recommended for mass emails

# Add body to email
message.attach(MIMEText(body, "plain"))

filename = "document.pdf"  # In same directory as script

# Open PDF file in binary mode
with open(filename, "rb") as attachment:
    # Add file as application/octet-stream
    # Email client can usually download this automatically as attachment
    part = MIMEBase("application", "octet-stream")
    part.set_payload(attachment.read())

# Encode file in ASCII characters to send by email
encoders.encode_base64(part)

# Add header as key/value pair to attachment part
part.add_header(
    "Content-Disposition",
    f"attachment; filename= {filename}",
)

# Add attachment to message and convert message to string
message.attach(part)
text = message.as_string()
```

```
# Log in to server using secure context and send email
context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(sender_email, receiver_email, text)
```

7.6 Ticket cancellation

```
IBM-EPBL
from pickle import load,dump
import time
import random
import os
class tickets:
    def _init_(self):
        self.no_ofac1stclass=0
        self.totaf=0
        self.no_ofac2ndclass=0
        self.no_ofac3rdclass=0
        self.no_ofsleeper=0
        self.no_oftickets=0
        self.name=''
        self.age=''
        self.resno=0
        self.status=''
    def ret(self):
        return(self.resno)
    def retname(self):
        return(self.name)
    def display(self):
        f=0
        fin1=open("tickets.dat","rb")
        if not fin1:
            print "ERROR"
        else:
            print
            n=int(raw_input("ENTER PNR NUMBER : "))
            print "\n\n"
            print ("FETCHING DATA . . .".center(80))
            time.sleep(1)
            print
            print('PLEASE WAIT...!!'.center(80))
            time.sleep(1)
            os.system('cls')
            try:
```

```
            while True:
                tick=load(fin1)
                if(n==tick.ret()):
                    f=1
                    print "="*80
                    print("PNR STATUS".center(80))
                    print"="*80
                    print
                    print "PASSENGER'S NAME :",tick.name
                    print
                    print "PASSENGER'S AGE :",tick.age
                    print
                    print "PNR NO :",tick.resno
                    print
                    print "STATUS :",tick.status
                    print
                    print "NO OF SEATS BOOKED : ",tick.no_oftickets
                    print
            except:
                pass
            fin1.close()
            if(f==0):
                print
                print "WRONG PNR NUMBER..!!"
                print
    def pending(self):
        self.status="WAITING LIST"
        print "PNR NUMBER :",self.resno
        print
        time.sleep(1.2)
        print "STATUS = ",self.status
        print
        print "NO OF SEATS BOOKED : ",self.no_oftickets
        print
    def confirmation (self):
        self.status="CONFIRMED"
        print "PNR NUMBER : ",self.resno
        print
        time.sleep(1.5)
        print "STATUS = ",self.status
        print
    def cancellation(self):
        z=0
        f=0
        fin=open("tickets.dat","rb")
        fout=open("temp.dat","ab")
        print
```

```
    r= int(raw_input("ENTER PNR NUMBER : "))
    try:
       while(True):
          tick=load(fin)
          z=tick.ret()
          if(z!=r):
             dump(tick,fout)
          elif(z==r):
              f=1
    except:
       pass
    fin.close()
    fout.close()
    os.remove("tickets.dat")
    os.rename("temp.dat","tickets.dat")
    if (f==0):
       print
       print "NO SUCH RESERVATION NUMBER FOUND"
       print
       time.sleep(2)
       os.system('cls')
    else:
       print
       print "TICKET CANCELLED"
       print"RS.600 REFUNDED...."
def reservation(self):
    trainno=int(raw_input("ENTER THE TRAIN NO:"))
    z=0
    f=0
    fin2=open("tr1details.dat")
    fin2.seek(0)
    if not fin2:
       print "ERROR"
    else:
       try:
          while True:
             tr=load(fin2)
             z=tr.gettrainno()
             n=tr.gettrainname()
             if (trainno==z):
                print
                print "TRAIN NAME IS : ",n
                f=1
                print
                print "-"*80
                no_ofac1st=tr.getno_ofac1stclass()
                no_ofac2nd=tr.getno_ofac2ndclass()
```

```
            no_ofac3rd=tr.getno_ofac3rdclass()
            no_ofsleeper=tr.getno_ofsleeper()
        if(f==1):
            fout1=open("tickets.dat","ab")
            print
            self.name=raw_input("ENTER THE PASSENGER'S NAME ")
            print
            self.age=int(raw_input("PASSENGER'S AGE : "))
            print
            print"\t\t SELECT A CLASS YOU WOULD LIKE TO TRAVEL IN :- "
            print "1.AC FIRST CLASS"
            print
            print "2.AC SECOND CLASS"
            print
            print "3.AC THIRD CLASS"
            print
            print "4.SLEEPER CLASS"
            print
            c=int(raw_input("\t\t\tENTER YOUR CHOICE = "))
            os.system('cls')
            amt1=0
            if(c==1):
                self.no_oftickets=int(raw_input("ENTER NO_OF FIRST CLASS AC SEATS TO BE BOOKED :
"))
                i=1
                while(i<=self.no_oftickets):
                    self.totaf=self.totaf+1
                    amt1=1000*self.no_oftickets
                    i=i+1
                print
                print "PROCESSING. .",
                time.sleep(0.5)
                print ".",
                time.sleep(0.3)
                print'.'
                time.sleep(2)
                os.system('cls')
                print "TOTAL AMOUNT TO BE PAID = ",amt1
                self.resno=int(random.randint(1000,2546))
                x=no_ofac1st-self.totaf
                print
                if(x>0):
                    self.confirmation()
                    dump(self,fout1)
                    break
                else:
                    self.pending()
```

```
                    dump(tick,fout1)
                    break
                elif(c==2):
                    self.no_oftickets=int(raw_input("ENTER NO_OF SECOND CLASS AC SEATS TO BE
BOOKED :  "))
                    i=1



def menu():
    tr=train()
    tick=tickets()
    print
    print "WELCOME TO PRAHIT AGENCY".center(80)
    while True:
        print
        print "="*80
        print " \t\t\t\t  RAILWAY"
        print
        print "="*80
        print
        print "\t\t\t1. **UPDATE TRAIN DETAILS."
        print
        print "\t\t\t2. TRAIN DETAILS. "
        print
        print "\t\t\t3. RESERVATION OF TICKETS."
        print
        print "\t\t\t4. CANCELLATION OF TICKETS. "
        print
        print "\t\t\t5. DISPLAY PNR STATUS."
        print
        print "\t\t\t6. QUIT."
        print"** - office use....."
        ch=int(raw_input("\t\t\tENTER YOUR CHOICE : "))
        os.system('cls')
        print "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\tLOADING. .",
        time.sleep(1)
        print (".",
        time.sleep(0.5)
        print (".")
        time.sleep(2)
        os.system('cls')
        if ch==1:
            j="***"
            r=raw_input("\n\n\n\n\n\n\n\n\n\n\n\t\t\t\tENTER THE PASSWORD: ")
            os.system('cls')
            if (j==r):
```

```
        x='y'
        while (x.lower()=='y'):
            fout=open("tr1details.dat","ab")
            tr.getinput()
            dump(tr,fout)
            fout.close()
            print"\n\n\n\n\n\n\n\n\n\n\n\t\t\tUPDATING TRAIN LIST PLEASE WAIT . .",
            time.sleep(1)
            print ("."),
            time.sleep(0.5)
            print ("."),
            time.sleep(2)
            os.system('cls')
            print "\n\n\n\n\n\n\n\n\n\n\n"
            x=raw_input("\t\tDO YOU WANT TO ADD ANY MORE TRAINS DETAILS ? ")
            os.system('cls')
        continue
    elif(j<>r):
        print"\n\n\n\n\n"
        print "WRONG PASSWORD".center(80)
elif ch==2:
    fin=open("tr1details.dat",'rb')
    if not fin:
        print "ERROR"
    else:
        try:
            while True:
                print"*"*80
                print"\t\t\t\tTRAIN DETAILS"
                print"*"*80
                print
                tr=load(fin)
                tr.output()



                raw_input("PRESS ENTER TO VIEW NEXT TRAIN DETAILS")
                os.system('cls')
        except EOFError:
            pass
elif ch==3:
    print'='*80
    print "\t\t\t\tRESERVATION OF TICKETS"
    print'='*80
    print
    tick.reservation()
elif ch==4:
```

```
            print"="*80
            print"\t\t\t\tCANCELLATION OF TICKETS"
            print
            print"="*80
            print
            tick.cancellation()
        elif ch==5:
            print "="*80
            print("PNR STATUS".center(80))
            print"="*80
            print
            tick.display()
        elif ch==6:
            quit()

        raw_input("PRESS ENTER TO GO TO BACK MENU".center(80))
        os.system('cls')

menu()
```

TESTING

8.1 Sprint 1

Date	[illegible]
Team ID	PNT2022TMID08784
Project Name	smart solutions for railways
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	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 authentication during my journey		1.Enter method of reservation 2.Enter name,age,gender 3.Enter how many tickets wants to be booked 4.Also enter the number member's		Tickets booked to be displayed	Working as expected	Pass				swetha
2	UI	Ticket status	a usercan see the status of my ticket Whether it's confirmed/ waiting/RAC		1 known to the status of the tvkeets booked		known to the status of the tvkeets booked	Working as expected	pass				vimal
3	Functional	Remainder notification	a user, I get remainders about my journey A day before my actual journey		1 user can get reminder nofication		user can get reminder nofication	Working as expected	pass				suganthan
4	Functional	GPS tracking	user can track the train using GPS and can get information such as ETA, Current stop and delay		1 tracking train for geting information		tracking process through GPS	Working as expected	pass				manoj

8.2 Sprint 2

Date	26 Oct 22
Team ID	PNT2022TMID08784
Project Name	smart solutions for railways
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
1	Functional	Booking	user can provide the basic details such as a name, age, gender etc		1 Enter method of reservation 2 Enter name,age,gender 3 Enter how many tickets wants to be booked 4 Also enter the number member's		Tickets booked to be displayed	Working as expected	Pass				Swetha
2	UI	Booking seats	User can choose the class, seat/ berth. If a preferred seat/berth isn't available I can be allocated based on the availability		1. known to which the seats are available		known to which the seats are available	Working as expected	pass				Vimal
3	Functional	Payment	user, I can choose to pay through credit Card/debit card/UPI		1 user can choose payment method 2 pay using fit method		payment for the booked tickets to be done using payment method through either the following methods credit	Working as expected	pass				Suganthan
4	Functional	Redirection	user can be redirected to the selected		1 After payment the user will be		After payment the user will be	Working as expected	pass				Manoj

8.3 Sprint 3

				Date	11Nov22							
				Team ID	PNT2022TMID08284							
				Project Name	smart solutions for railways							
				Maximum Marks	4 marks							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	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 authentication during my journey.		1.Enter method of reservation 2.Enter name,age,gender 3.Enter how many tickets wants to be booked 4.Also enter the number member's details like name,age,gender		Tickets booked to be displayed	Working as expected	Pass				swetha
2	UI	Ticket status	a usercan see the status of my ticket Whether it's confirmed/ waiting/RAC		1.known to the status of the tickets booked		known to the status of the tickets booked	Working not expected	FFail				vimal
3	Functional	Remainder notification	a user, I get remainders about my journey A day before my actual journey.		1.user can get remainder nofication.		user can get remainder nofication.	working as expected	pass				suganthan
4	Functional	GPS tracking	user can track the train using GPS and can get information such as ETA, Current stats and delay.		1.tracking train for getting information		tracking process through GPS	Working as expected	pass				Manoj

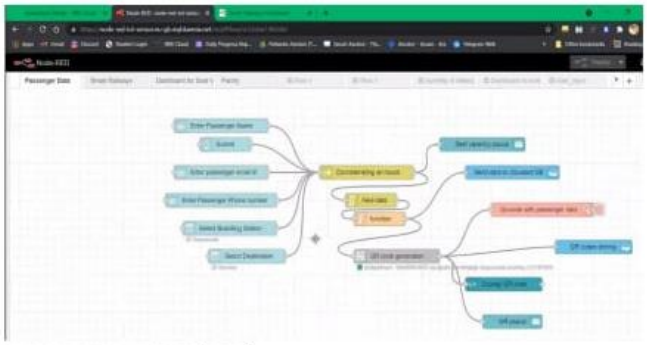
8.4 Sprint 4

				Date	03Nov22							
				Team ID	PNT2022TMID48423							
				Project Name	smart solutions for railways							
				Maximum Marks	4 marks							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
1	Functional	Ticket cancellatio	user can cancel my tickets there's any Change of plan		1.tickets to be cancelled		Tickets booked to be cancelled	Working not as expected	fail				swetha
2	UI	Raise queries	user can raise queries through the query box or via mail		1.raise the queries		raise the queries	Working as expected	pass				vimal
3	Functional	Answer the queries	user will answer the questions/ doubts Raised by the customers.		1.answer the queries		answer the queries	Working as expected	pass				suganthan
4	Functional	Feed details	a user will feed information about the trains delays and add extra seats if a new compartment is added		1.information feeding on trains		information feeding on trains	Partially	pass				Manoj

RESULTS

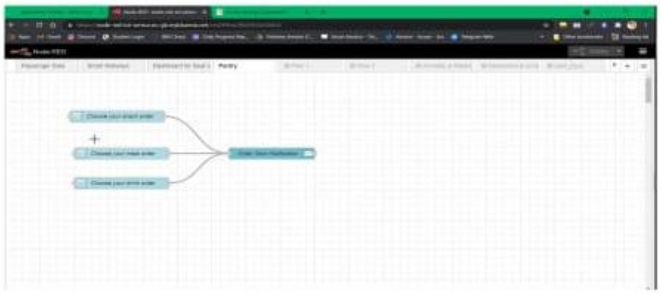
RESULTS

9.1 PERFORMANCE METRICS

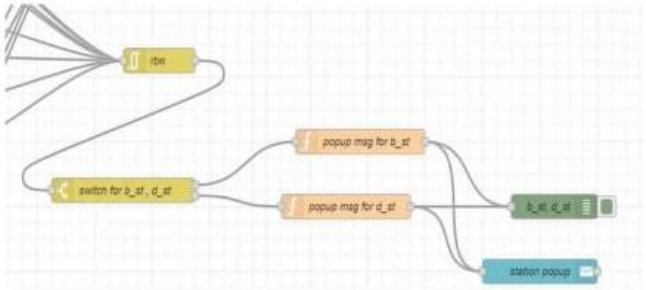


SMART RAILWAYS:

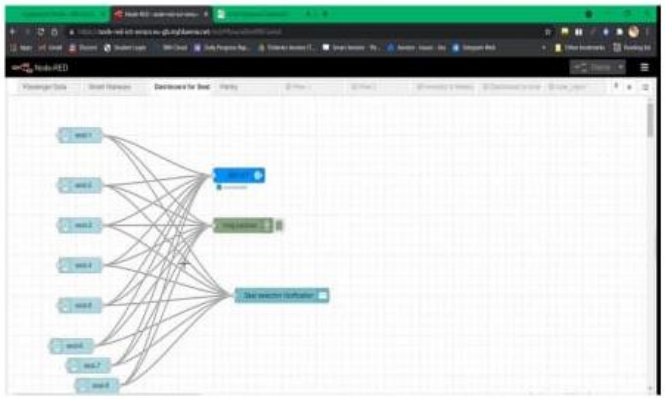
PANTRY:



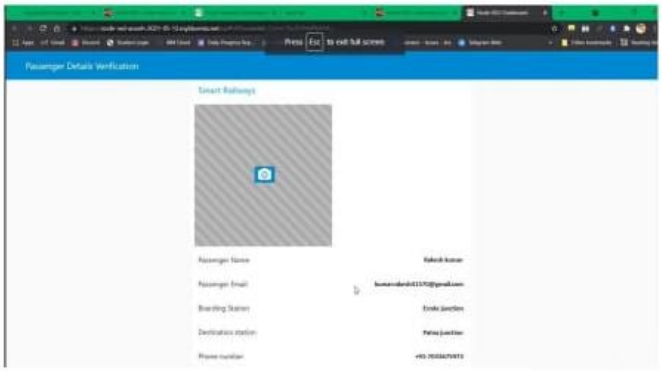
TC:



DASHBOARD FOR SEAT VACANCY:



SCANNED OUTPUT AT TC:



DAIGRAM

ADVANTAGES &DISADVANTAGES

ADVANTAGES &DISADVANTAGES

11.1 ADVANTAGES

- Openness – compatibility between different system modules, potentially from different vendors;
- Orchestration – ability to manage large numbers of devices, with full visibility over them;
- Dynamic scaling – ability to scale the system according to the application needs, through resource virtualization and cloud operation;
- Automation – ability to automate parts of the system monitoring application, leading to better performance and lower operation costs.

11.2 DISADVANTAGES

- Approaches to flexible, effective, efficient, and low-cost data collection for both railway vehicles and infrastructure monitoring, using regular trains;
- Data processing, reduction, and analysis in local controllers, and subsequent sending of that data to the cloud, for further processing;
- Online data processing systems, for real-time monitoring, using emerging communication technologies;
- Integrated, interoperable, and scalable solutions for railway systems preventive maintenance.

CONCLUSION

4. 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.

FUTURE SCOPE

5. FUTURE SCOPE

In future CCTV systems with IP based camera can be used for monitoring the visual videos captured from the track. It will also increase security for both passengers and railways. GPS can also be used to detect exact location of track fault area, IP cameras can also be used to show fault with the help of video. Locations on Google maps with the help of sensors can be used to detect in which area track is broken