PROJECT REPORT

Smart Solutions For Railways

Team ID: PNT2022TMID28657

Batch: B8-2A4E

TEAM LEADER

Name: HARISH K

Register Number: 410719104030

TEAM MEMBERS:

Name: SRIHARISH S

Register Number: 410719104100

Name: VETRIVEL R

Register Number: 410719104118

Name: BALU YOGESHWAR

Register Number: 410719104010

TABLE OF CONTENTS

S.NO	TOPICS	PG.NO
1	INTRODUCTION	3
2	LITERATURE SURVEY	3
3	IDEATION & PROPOSED SOLUTION	5
4	REQUIREMENT ANALYSIS	8
5	PROJECT DESIGN	9
6	PROJECT PLANNING & SCHEDULING	11
7	CODING & SOLUTIONING	15
8	TESTING	16
9	RESULTS	18
10	ADVANTAGES & DISADVANTAGES	18
11	CONCLUSION	19
12	FUTURE SCOPE	19
13	APPENDIX	19

1. INTRODUCTION

1.1 Project Overview

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 present 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.

1.2 Purpose

The Purpose of our Project is to reduce the work load of the user and also the use of paper, to enable online Ticket Booking, to track the live location of train, to enable Automatic Ticket Verification system, to reduce the work load of Travelling Ticket Examiner (TTE).

2. LITERATURE SURVEY

2.1 Existing Problem

Passengers who need to book train tickets have to follow a tedious procedure to get a ticket reserved. They have to wait in the Ticket booking counter to do the reservation. Also the TTE has a heavy workload in processing all the paper documents in verifying a user's ticket. Public users wish to track their train journey to have a sophisticated travel.

2.2 References

1. INTERNET OF THINGS FOR SMART RAILWAYS

Authors: Ohyun Jo, Yong-Kyu Kim, Juyeop Kim

Date of Publication: 06 September 2017

Project Description:

The explosively growing demand of Internet of Things (IoT) has rendered broad scale advancements in the fields across sensors, radio access, network, and hardware/software platforms for mass market applications. In spite of the recent advancements, limited coverage and battery for persistent connections of IoT devices still remains a critical impediment to practical service applications. In this paper, we introduces a cost-effective IoT solution consisting of device platform, gateway, IoT network, and platform server for smart railway infrastructure. Then, we evaluate and demonstrate the applicability through an in-depth case study related to IoT-based maintenance by implementing a proof of concept and performing experimental works. The IoT solution applied for the smart railway application makes it easy to grasp the condition information distributed over a wide railway area. To deduce the potential and feasibility, we propose the network architecture of IoT solution and evaluate the performance of the candidate radio access technologies for delivering IoT data in the aspects of power consumption and coverage by performing an intensive field test with system level implementations. Based on the observation of use cases in interdisciplinary approaches, we figure out the benefits that the IoT can bring.

2. SMART TRAIN DETECTOR USING IoT APPROACH

Authors: Payal Srivastava, Rana Majumdar, Bonny Paulose, Sunil Kumar Chowdhary, Abhishek Srivastava

Date of Publication: January 2019 **Project Description:**

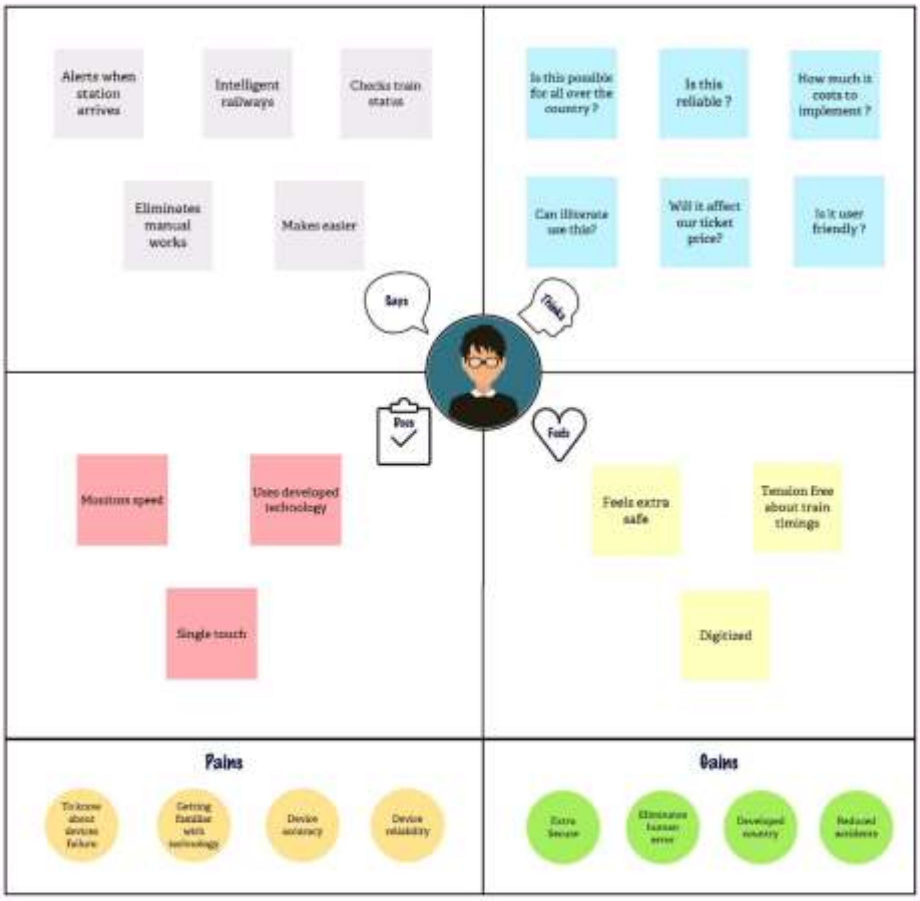
Only metal detection in railway tracks to indicate the movement of train is not sufficient. The sensors present in the railway tracks can detect any metal object, be it a train or mere a coin. Thus, in order to make the working more foolproof, introduction of another parameter, i.e., weight on the railway track is necessary. This paper describes an approach to collaborate metal detection with weight detection in railway tracks to detect the train movement using the principle of IoT, using the load cells along with the metal sensors.

2.3 Problem Statement definition

Railway Passengers need a way to book their tickets easily without delay and tedious process so that the passengers get benefits by saving their time and energy in travelling. Ticket checkers are in need of a Digital verification system of passenger’s tickets to avoid huge use of pen and paper. Passengers need a platform to track the current location and status of trains in order to avoid anxiety of fear of missing the train.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



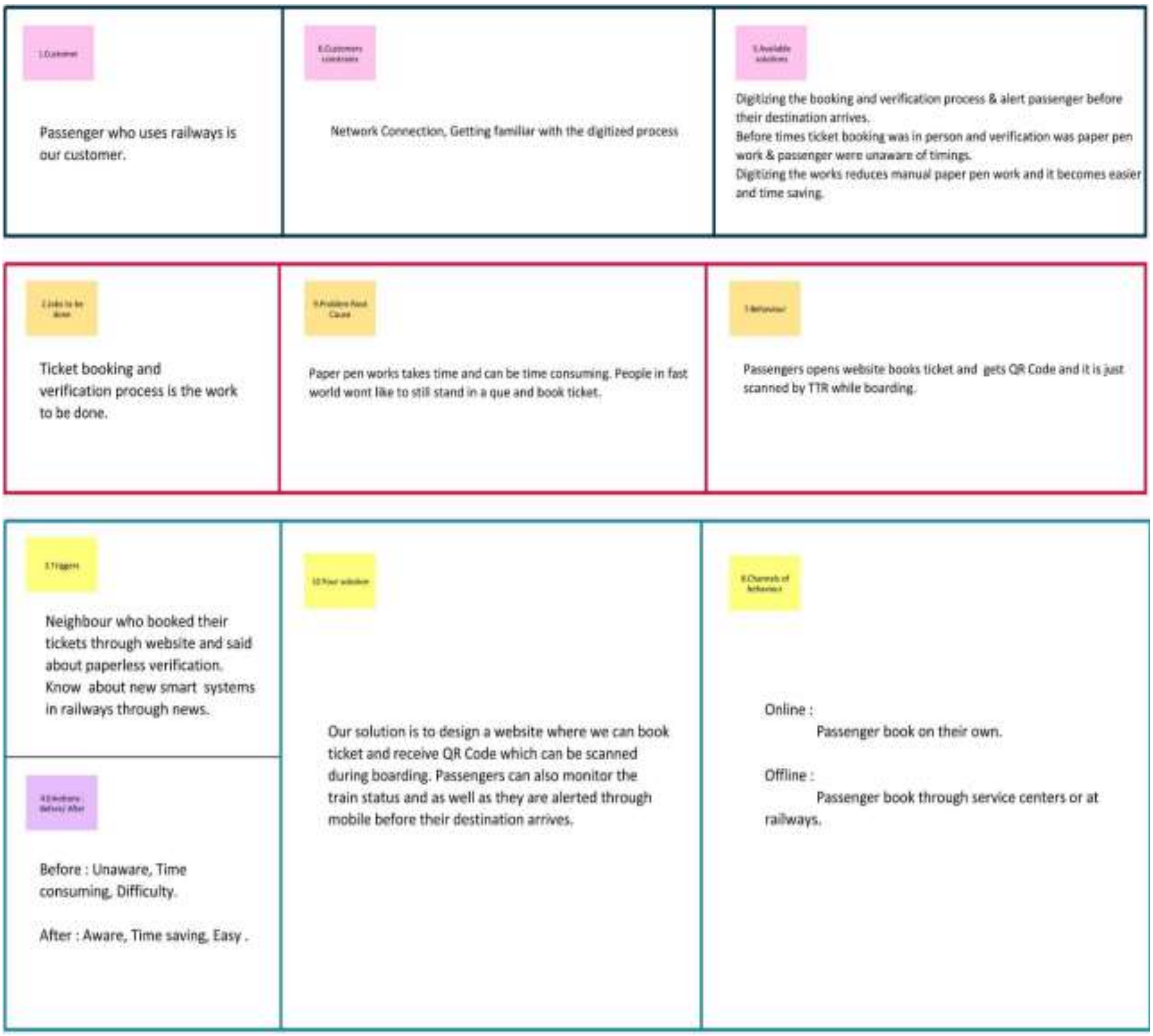
3.2 Ideation & Brainstorming

3.3 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	To make ticket booking and verification process easier and alert, make aware about passenger’s destination before it arrives.
2	Idea / Solution description	We have come up with the idea of designing a website to book tickets and to see other information about train. After booking passenger receive QR code and it is scanned by ticket collector during boarding to verify identity. Unique ID database is stored in the cloud. Through GPS tracking live status of train can be monitored and alerts passenger before their destination arrives.
3	Novelty / Uniqueness	Passenger is alerted though mobile phone before destination arrives.

4	Social Impact / Customer satisfaction	People feel relaxed and tension free about tickets and arrival of destination.
5	Business Model (Revenue Model)	Digitizing the paper works minimizes the cost, and people more likely prefer this and it is cost efficient and secure.
6	Scalability of the Solution	It supports adding new features and it is user friendly with high security.

3.4 Problem Solution Fit



1.Customer

Passenger who uses railways is our customer.

6.Customers constraints

Network Connection, Getting familiar with the digitized process

5.Available solutions

Digitizing the booking and verification process & alert passenger before their destination arrives.
Before times ticket booking was in person and verification was paper pen work & passenger were unaware of timings.
Digitizing the works reduces manual paper pen work and it becomes easier and time saving.

2.Jobs to be done

Ticket booking and verification process is the work to be done.

9.Problem Root Cause

Paper pen works takes time and can be time consuming. People in fast world wont like to still stand in a que and book ticket.

7.Behaviour

Passengers opens website books ticket and gets QR Code and it is just scanned by TTR while boarding.

3.Triggers

Neighbour who booked their tickets through website and said about paperless verification.
Know about new smart systems in railways through news.

4.Emotions Before/ After

Before : Unaware, Time consuming, Difficulty.

After : Aware, Time saving, Easy .

10.Your solution

Our solution is to design a website where we can book ticket and receive QR Code which can be scanned during boarding. Passengers can also monitor the train status and as well as they are alerted through mobile before their destination arrives.

8.Channels of behaviour

Online :
Passenger book on their own.

Offline :
Passenger book through service centers or at railways.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

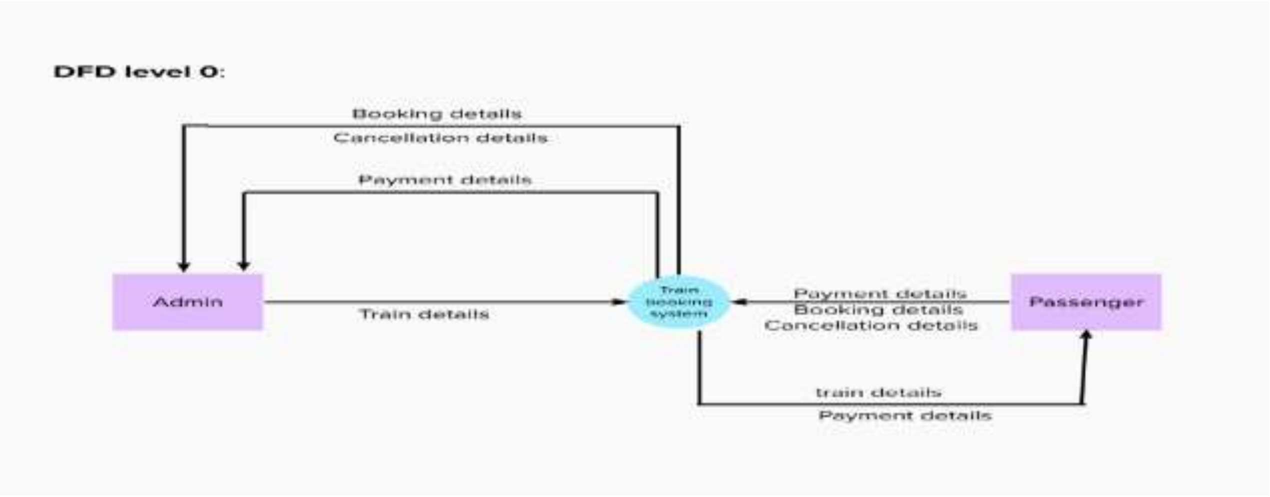
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Booking	Book tickets through website/App.
FR-2	User Confirmation	Confirmation via OTP.
FR-3	User’s Ticket	Receiving QR-code through mail.
FR-4	User’s Remainder	One day before their journey via SMS.
FR-5	User’s Destination	SMS Alert before 15 mins of their departure.

4.2 Non-Functional Requirements

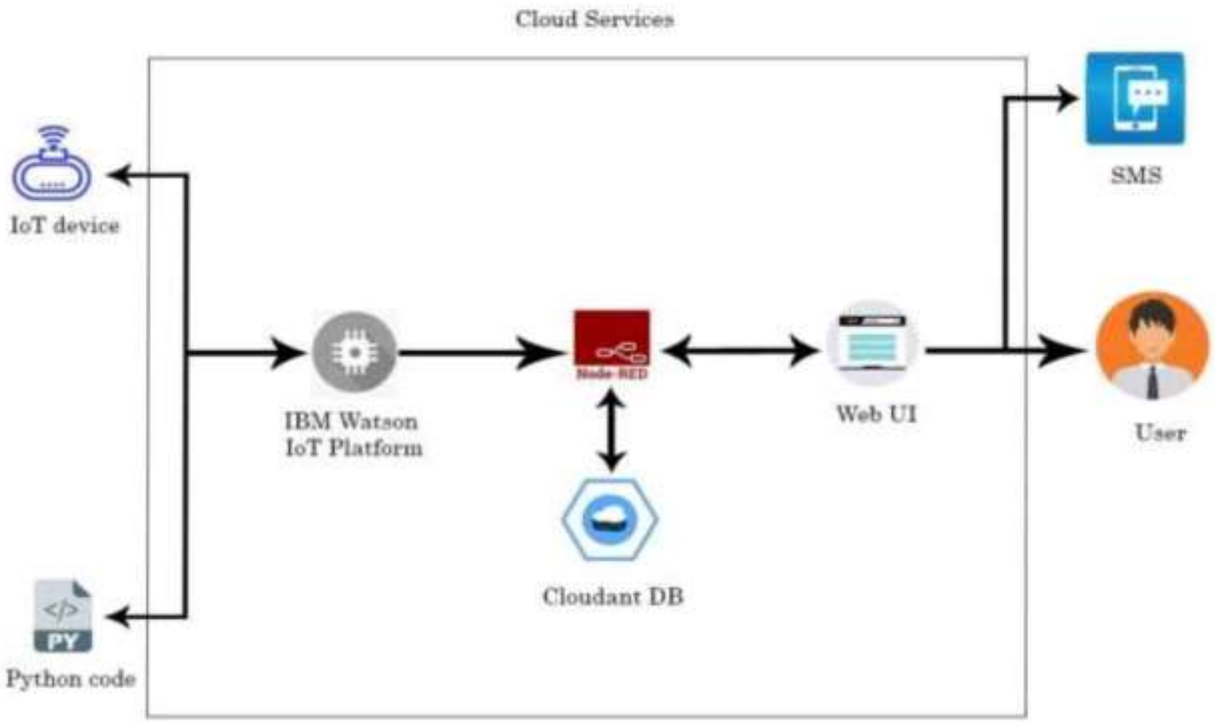
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Simple booking procedure and user can use website easily.
NFR-2	**Security**	Personal information of the user is secured in database.
NFR-3	**Reliability**	Multiple number of users can access without any failure.
NFR-4	**Performance**	Uder friendly
NFR-5	**Availability**	24/7 availability of website.
NFR-6	**Scalability**	Can have the feature of selecting the required favourite seat.

5. PROJECT DESIGN

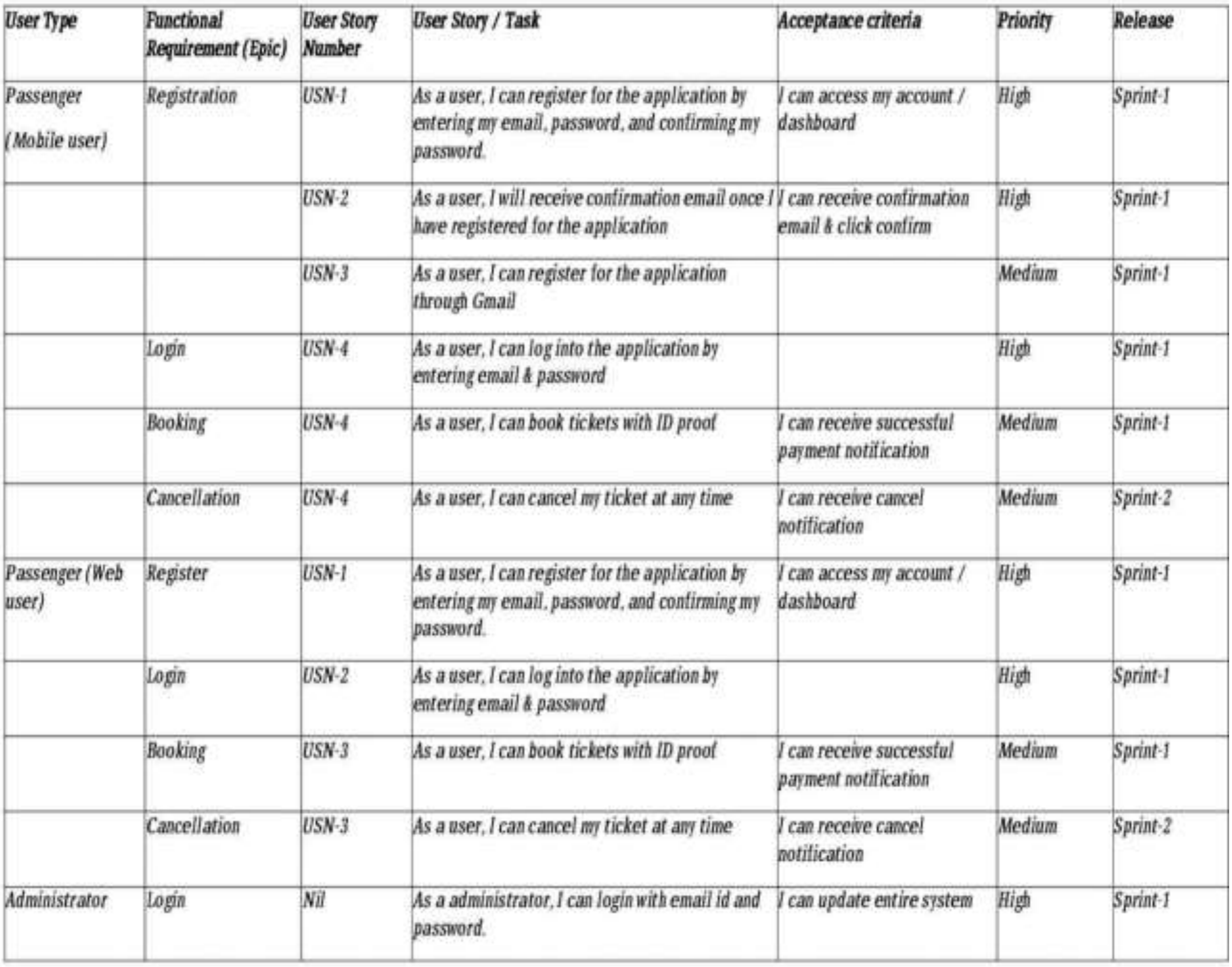
5.1 Data Flow Diagram



5.2 Solution and Technical Architecture



5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Passenger (Mobile user)	Registration	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
		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
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Booking	USN-4	As a user, I can book tickets with ID proof	I can receive successful payment notification	Medium	Sprint-1
	Cancellation	USN-4	As a user, I can cancel my ticket at any time	I can receive cancel notification	Medium	Sprint-2
Passenger (Web user)	Register	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
	Login	USN-2	As a user, I can log into the application by entering email & password		High	Sprint-1
	Booking	USN-3	As a user, I can book tickets with ID proof	I can receive successful payment notification	Medium	Sprint-1
	Cancellation	USN-3	As a user, I can cancel my ticket at any time	I can receive cancel notification	Medium	Sprint-2
Administrator	Login	Nil	As a administrator, I can login with email id and password.	I can update entire system	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
SPRINT 1					
Sprint-1	Login	USN-1	As a Developer, I can create the login page with email & password using Node-Red App.	2	Medium
Sprint-1	Dashboard	USN-2	As a Developer, I can create a search box for the trains and seat availability.	6	High
Sprint-1	Booking the Tickets	USN-3	As a Developer, I can provide a facility to book the train ticket according to user preference.	6	High
Sprint-1	QR code Generation	USN-4	As a Developer, I can write a code to view and download the QR code for the Booked Ticket.	6	High
SPRINT 2					
Sprint-2	Login for TTE	USN-1	As a Developer,I can create the login page for TTE with email and password using the Node-red app.	6	High

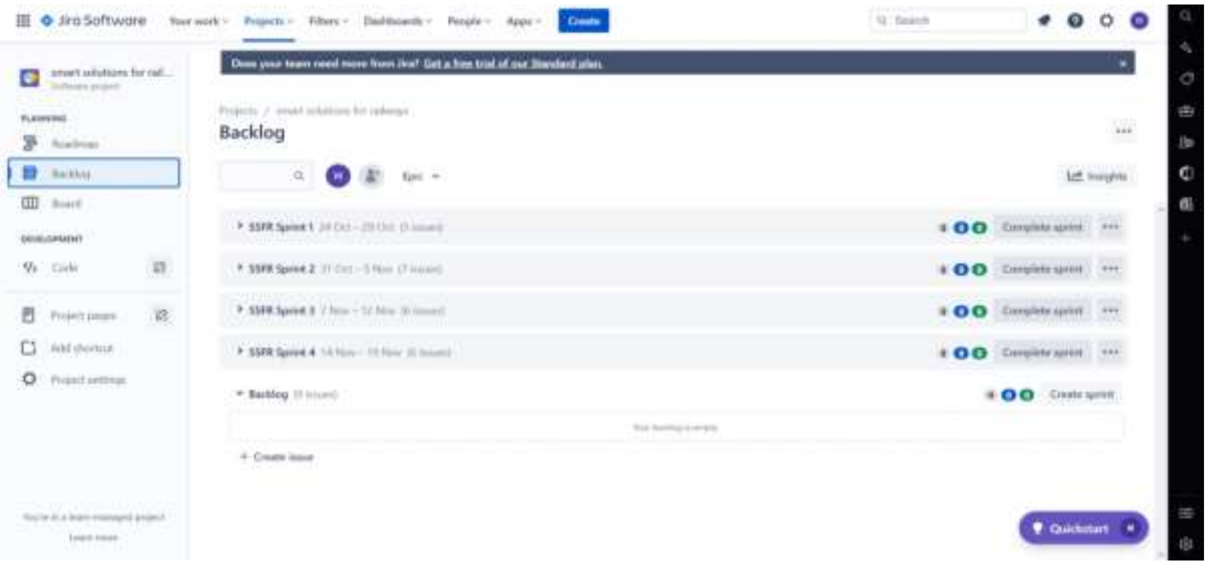
Sprint-2	Dashboard	USN-2	As a Developer,I can create a drop down box for the number of passengers on board and remaining number of passengers yet to	6	High
Sprint-2	Passenger Details	USN-3	As a developer,I can provide a feature to view the passenger details which are stored in the Cloud and are retrieved and displayed in the webpage.	2	Medium
Sprint-2	Ticket verification	USN-4	As a developer,I can provide a verification mechanism to check the integrity of QR code and the passenger details.	6	High
SPRINT 3					
Sprint-3	Collaboration	USN-1	As a developer, I will provide a smooth way to book the ticket through a website and also to refund if the passenger cancels the ticket.	6	High
Sprint-3	Verifying the passenger details	USN-2	As a developer, I will check whether the passenger is taking his/her journey with a	6	High

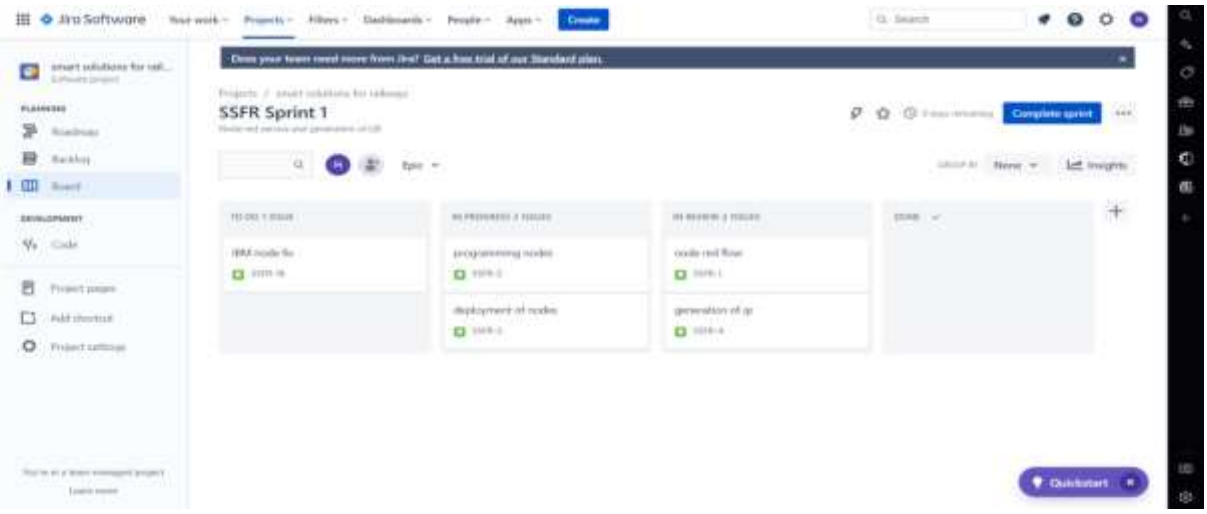
			proper confirmed ticket.		
Sprint-3	Identification of the passenger	USN-3	As a developer, I will make sure that the on- boarded passenger is not involved in any travel fraudulently.	2	Medium
Sprint-3	Verifying the tickets	USN-4	As a developer, I will scan the QR code generated by python code to extract and verify the passenger details.	6	High
			SPRINT 4		
Sprint-4	Tracking webpage	USN-1	As a developer , I can create web page to view train status using Node RED	6	High
Sprint-4	Live location details	USN-2	As a developer I can extract details from IoT device using python code and IBM watson.	6	High
Sprint-4	Retrieving from cloud	USN-3	As a developer, I can upload the details to cloud and display it to user by connecting it with Node RED application.	6	High
Sprint-4	Sending updates	USN-4	As a passenger ,I would like to receive updates over my train status during my journey via fast SMS or App notification.	2	Medium

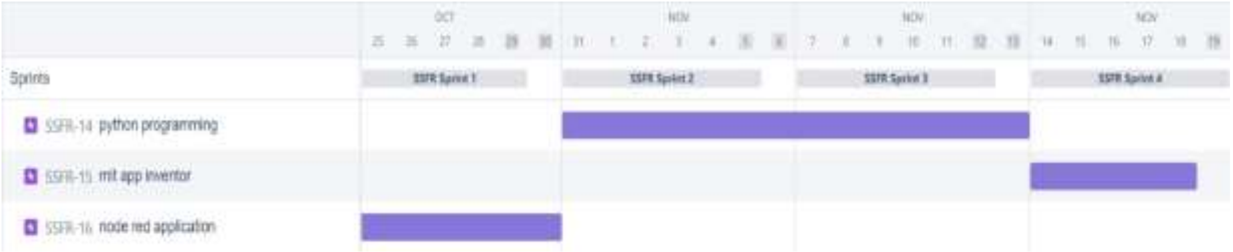
6.2 Sprint Delivery Schedule

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	07 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	10 Oct 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	15 Oct 2022

6.3 Reports from JIRA







7.CODING AND SOLUTIONING

7.1. Feature 1

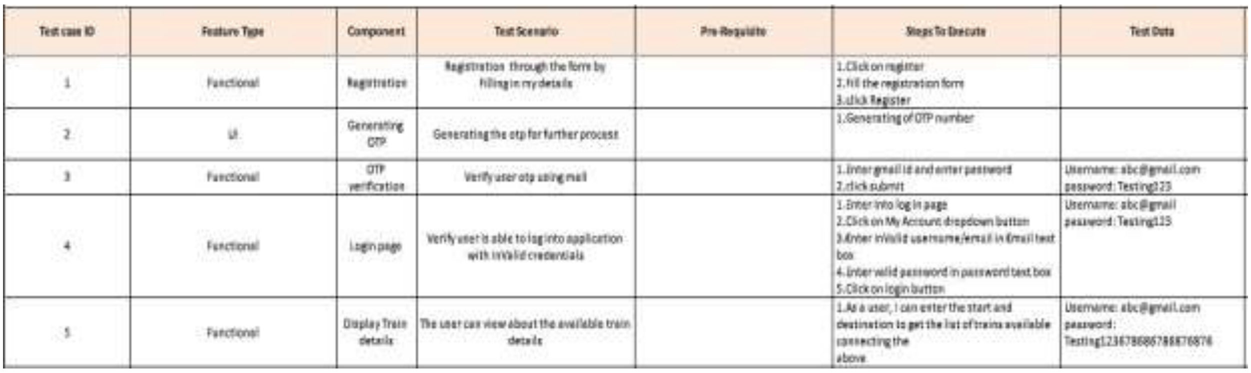
- IOT device
- IBM Watson platform
- Node red
- Cloudant DB
- Web UI
- Geofence MIT App
- Python code

7.2. Feature 2

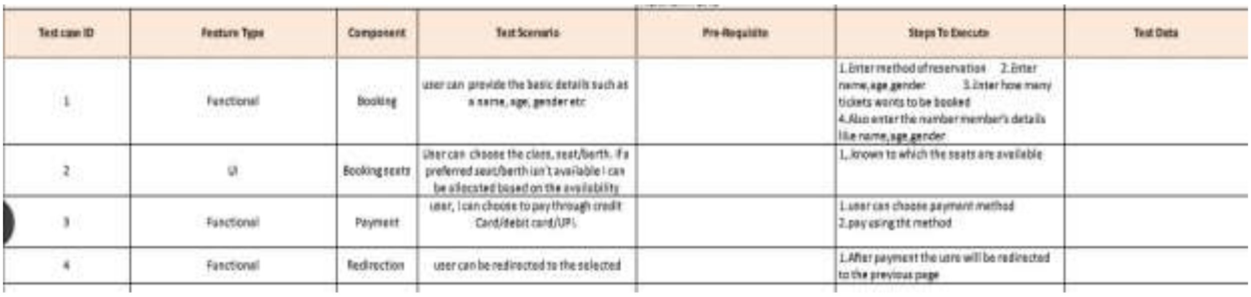
- Registration
- Login
- Verification
- Ticket Booking
- Payment
- Ticket Cancellation
- Adding Queries

8. TESTING

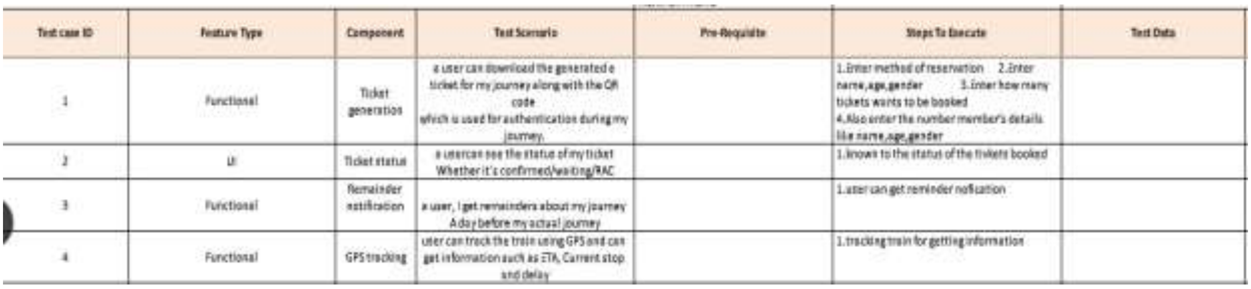
8.1 Test Cases



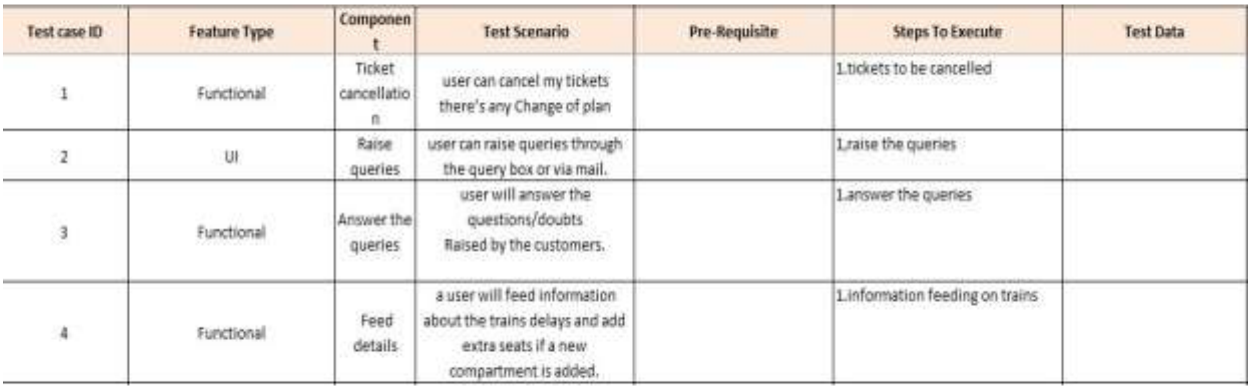
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data
1	Functional	Registration	Registration through the form by filling in my details		1.Click on register 2.fill the registration form 3.click Register	
2	UI	Generating OTP	Generating the otp for further process		1.Generating of OTP number	
3	Functional	OTP verification	Verify user otp using mail		1.Enter gmail id and enter password 2.click submit	Username: abc@gmail.com password: Testing123
4	Functional	Login page	Verify user is able to log into application with inValid credentials		1.Enter into log in 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: abc@gmail password: Testing123
5	Functional	Display Train details	The user can view about the available train details		1.As a user, I can enter the start and destination to get the list of trains available connecting the above	Username: abc@gmail.com password: Testing123678686788876876



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data
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 details like name,age,gender	
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	
3	Functional	Payment	user, I can choose to pay through credit Card/debit card/UPI.		1.user can choose payment method 2.pay using tht method	
4	Functional	Redirection	user can be redirected to the selected		1.After payment the usre will be redirected to the previous page	



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data
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	
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	
3	Functional	Remainder notification	a user, I get remainders about my journey A day before my actual journey		1.user can get reminder nofication	
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 getting information	



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data
1	Functional	Ticket cancellation	user can cancel my tickets there's any Change of plan		1.tickets to be cancelled	
2	UI	Raise queries	user can raise queries through the query box or via mail.		1.raise the queries	
3	Functional	Answer the queries	user will answer the questions/doubts Raised by the customers.		1.answer the queries	
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	

8.2 User Acceptance Testing

Test case #2	Select arrival railway station
Time/Date	22:30 / 16th December 2022
Action	Click the drop down list next to “To” Select the station “Kumbakonam Railway central”
Expected Result	Application displays “Kumbakonam railway station” as selected in the drop down list next to the text “To”
Result	(Enter actual result here)
Pass/Fail	(Enter Test pass or fail here)

9.RESULTS

9.1PERFORMANCE METRICS

Encouraged by the results of deployment of OMRS, including some critical detection which could have potentially been cause of an accident, not otherwise detectable by normal maintenance procedure, Indian Railways is now going ahead with greater adoption of track side based maintenance systems with an aim towards predictive maintenance.

Further, moving towards predictive maintenance practices in yards, Indian Railways is envisaging to convert its ‘freight examination yards’ into technology driven ‘Smart Yards’ for automatic detection of faults/defects/deficiencies in freight wagons.

These Smart Yards will predict anomalies like Hot Wheel Hot Axle, defective bearings, defective wheels, hanging/loose/missing parts etc.

Long before any failure actually happens. Smart Yards will be equipped with various automated technology driven systems including OMRS, Hot Box Detector, Wheel Profile Recorder and Machine Vision Equipments etc.

10. ADVANTAGES

- Increased efficiency
- Reduced downtime
- Enhanced safety
- Increased passenger satisfaction

DISADVANTAGES

- To establish the entire network it is quite a costly task. Since these are the issues of the government cost doesn't matter a lot.
- The Arduino board is a delicate device so it has to be handled carefully.

11. Conclusion

The railway industry is on its way to integrate predictive maintenance and Big Data. Recent advancements in sensors and condition monitoring technologies have led to continuous data collection and evaluation, significantly minimising the number and cost of unscheduled maintenance. Most significant improvements have been evidenced by more informative and user- friendly websites, mobile applications for real-time information about vehicles in motion, and e-ticket purchases and timetable information implemented at stations and stops. With the rise of Industry 4.0, railway companies can now ensure that they are prepared to avoid the surprise of equipment downtime.

12. Future Scope

- One of the most significant benefits of an online reservation system is that they are always open for business and can accept bookings 24/7, so the customers don't have to wait until the next day to make a reservation or worse.
- By having an online reservation system, we're one step ahead of the game, tipping the scales in the favor when prospects research and compare the capabilities against the competitors.
- We will be able to create mobile smartphone applications in the future.The framework already serves desktop computers and we will be able to make web browsers compliant in the future.
- India is projected to account for 40% of the total global share of rail activity by 2050. This allows great potential for the online ticket booking platform industry**.**

13.APPENDIX

Source Code

Scanner Code

```
from http import client
import cv2
import pyzbar
from pyzbar.pyzbar import decode
import time
```

```
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-
1oj043bu90m78ng4h2j27w5nob2nvcma6xanc6bk0a7m',
'daf3c00c2cc182af425a5691a07f7b93')
service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-
1oj043bu90m78ng4h2j27w5nob2nvcma6xanc6bk0a7m:daf3c00c2cc182af425a5691
a07f7b93@932393aa-9f82-4144-9251-2c519fb30962-
bluemix.cloudantnosqldb.appdomain.cloud')

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

while True:
        _, frame = cap.read()
        decodedObjects = 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(a)
                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()
```

Python Code

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "xfxj98",
        "typeId": "railway23",
        "deviceId":"Device1"
    },
    "auth": {
        "token": "987456321"
    }
 }
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,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()
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-50656-1660920613

Project Demo Link

https://drive.google.com/drive/folders/1-iwQHqUm9AwWVOrsPKwk5yYjCcNgCgaE