SMART SOLUTIONS FOR RALIWAYS

A PROJECT REPORT

Submitted by

ALEN WENISH J	**212219040009**
SATHISHKUMAR K	**212219040142**
KISHORE R	**212219040062**
AJAY A	**212219040006**

TEAM ID: PNT2022TMID03238

SAVEETHA ENGINEERING COLLEGE
CHENNAI

INDEX

S.NO	TITLE
1	**INTRODUCTION**
1.1	Project Overview
1.2	Purpose
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 requirements
4.2	Non-Functional requirements
5	**PROJECT DESIGN**
5.1	Data Flow Diagrams

5.2	Solution &Technical Architecture
5.3	User Stories
6	**PROJECT PLANNING & SCHEDULING**
6.1	Sprint Planning & Estimation
6.2	Sprint Delivery Schedule
6.3	Reports from JIRA
7	**CODING & SOLUTIONING**
7.1	Feature 1
7.2	Feature 2
7.3	Database Schema
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**

CHAPTER 1
INTRODUCTION

1.1 PROJECT SUMMARY

The largest railway system in Asia is operated by Indian Railways. Additionally, it is the second-largest network in the world run by a single organization. The access of the train, the price of the train, the departure and arrival times of the train, and the ability to book a ticket using a debit, credit, or master card or an upi-id are all available to users in this. If a user wants to cancel a ticket after booking it, they can do so with convenience as well. It has a QR code that holds the customer's information and is useful for the ticket inspector to verify. It has the ability to track trains using a gps system. Online ticket booking has many benefits, including cutting down on paper use and electricity use at the train ticket counter. It also saves time by eliminating the need to wait in a lengthy line. Our railroad reservation system is employed in order to prevent manual errors.

1.2. PURPOSE

A train passenger needs to be aware of the availability of their tickets, the status of their reservations for a certain train or location, the arrival and departure times of their train, any special trains, and the location of their train. Such inquiries cannot be answered during peak hours by the customer information centres at train stations. Less reservation counters are available for customers and travellers. On the majority of reservation systems, it takes a while for anyone to make a reservation. The travellers cannot get assistance from call centres. The goal of the online railway ticket reservation system is to provide a web application that offers customers the option to book tickets online as well as information about available trains.

CHAPTER 2
LITERATURE SURVEY

TITLE: TRACKWARN-AN AI-DRIVEN WARNING SYSTEM FOR RAILWAY TRACK WORKERS

AUTHOR: M.I.M.Amjath

YEAR OF PUBLICATION: 2021

he analyzed and looked at the train mishaps, hospitalization keep, and so on. It gets in to extra portrayal of insights. The peril of huge injury, in light of separation cosmopolitan, is multiple times greater for travelers travel via car contrasted and travelers going by rail. The mean length of keep in clinic for a transport mishap including a railroad train was four days that were longer than the mean length of save for all External reasons for injury. A train is a set of vehicles, empty or loaded worked by locomotive, or any other self-propelled unit, including light engine/engines or rail-motor vehicles or a single rail-motor vehicle, empty or conveying passengers, live-stock, parcels or goods, which cannot be readily lifted off the track and running under a particular number or a distinct name from fixed point of departure to a fixed destination. Part of a train shall also be treated as a train for the purpose of these definition, classification and statistics. The train engine or any other vehicle once put on the train continues to be a part of the train until the station is reached beyond which it is not required to go on the same train.

DISADVANTAGE:

Slows down the study process.

TITLE: MONITORING OF THE OPERATING PARAMETERS OF RAILWAY SYSTEMS THROUGH THE USE OF SMARTPHONE DETECTION TECHNOLOGIES

AUTHOR: Francesco Apicella

YEAR OF PUBLICATION: 2021

Our cities are becoming increasingly smart thanks to information and communication technologies, sustainable solutions for human activities and innovative mobility frameworks. In this context, one of the most promising approaches is relying on the Internet of Things (IoT) which allows objects of everyday life to become computing devices exchanging useful data. In particular, the paper proposes to adopt such technologies for monitoring railway systems, thus obtaining information concerning, on one hand, service performance and, on the other, travellers behaviour. A numerical application has been performed in a real rail context, thus pointing out the feasibility of the proposed methodology. Improving safety by early warning of distress in or impending failures in wheels and wheel bearings. Using the vibration signature of the same sensors that are strategically placed on unsprung mass to do track condition monitoring too to indicate deterioration in the health of tracks thereby avoiding sudden failures in service.

DISADVANATGE:

Rail transport cannot provide door to door service as it is tied to a particular track. Intermediate loading or unloading[5] involves greater cost, more wear and tear and wastage of time.

2.1 EXISTING SYSTEM

A GSM and GPS module were used to pinpoint the exact location of the defective tracks so that the authorities could be notified via SMS and sent a link to view the area on Google Maps. A prototype that can take pictures of the track, compare them to an older database, and alert the authorities of a breach in the surface was presented by Rizvi Aliza Raza. Table provides a thorough examination of conventional railway track defect detecting methods.

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 railwaytrack 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.

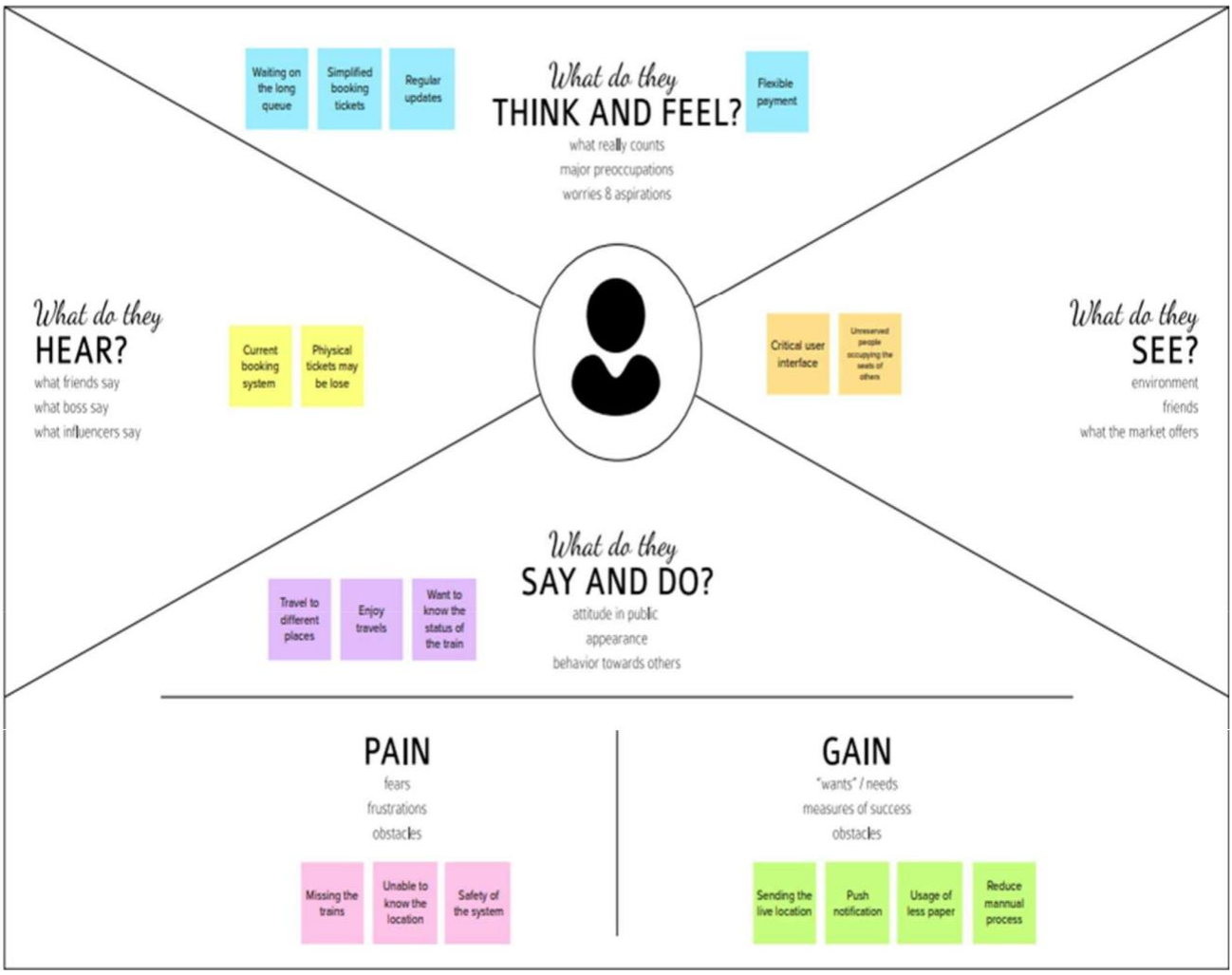
2.3 PROBLEM STATEMENT DEFINITION

In a train a ticket is issued by a railway operator that allows users to travel on the railway . User can use tickets to travel on a specific route at a specific time. The introduction of manual ticketing has always made it easier for authorities to keep track of various events involving financial matters and people's travelling patterns. These activities, however, had to be carried out manually, requiring a large amount of manpower and resources.The use of electronic equipment such as computers, printing machines, paper, and ink are the primary and most important resources required for paper ticketing systems.

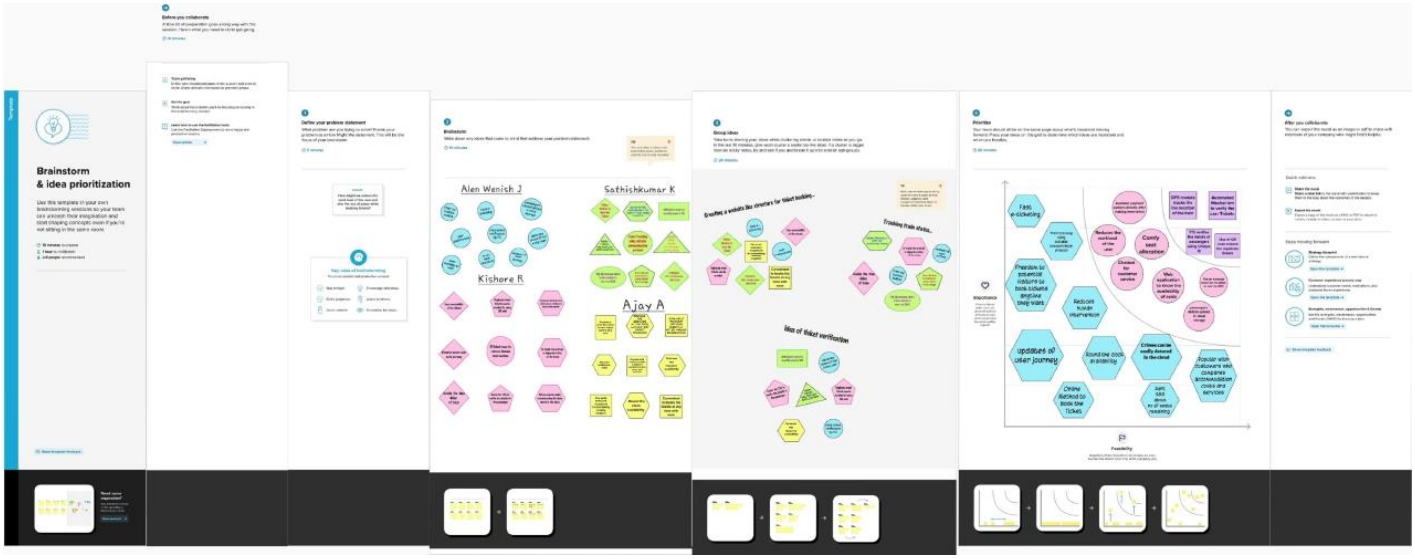
CHAPTER 3

IDEATION AND PROPOSED SOLUTON

3.1 EMPATHY MAP CANVAS



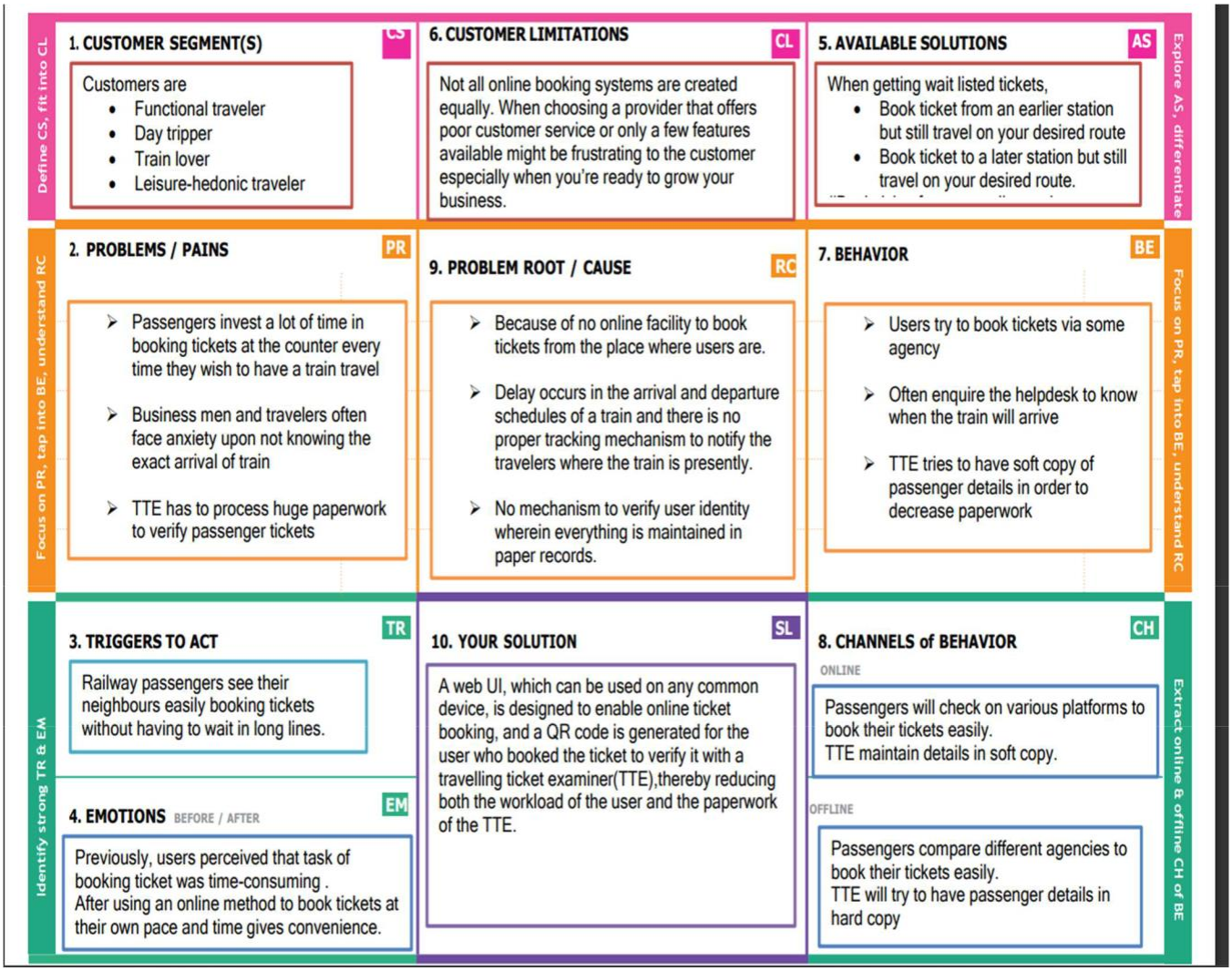
3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Smart Solutions for railways is designed to reduced the work load of the user and also the use of paper
2.	Idea / Solution description	A Web UI is designed to enable online ticket booking and a QR code is generated for the user who has booked the ticket to verify it with Travelling Ticket Examiner (TTE) which is done using Cloud Service .
3.	Novelty / Uniqueness	• Unique QR code for each user to verify their Tickets by Ticket Checker. • GPS Module to track the location of Train and live status is updated in the Web app .
4.	Social Impact / Customer Satisfaction	• GPS Module to track the location of Train and live status is updated in the Web app . • The location of the train is updated periodically in the web app so customers can easily track the status of the train which may help the customer to arrive on time .
5.	Business Model (Revenue Model)	• Selling a Product which enables online booking platform and automatic verification system would fetch more revenues to the Organisation , also the Online Platform has to be maintained continuously and so , the business will sustain and can be improved if required .
6.	Scalability of the Solution	• a QR code is generated to hold the unique data of user in cloud , and a GPS module is attached to lot platform inorder to monitor the train location which gives an additional tracking mechanism to ensure reliability . • We propose a solution which works on the SaaS (Software as a Service) cloud model wherein all necessities are deployed in the web browser which gives an easy access and also the cost is minimal which gives a scaled solution approach .

3.4 PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Before the user registration there will be language selector .All the language is applicable .When user enter in to the website they can see the page which shows enter the email-id ,mobile number and name. After that in screen it shows the verification code which will be sent to the email-id.
FR-2	User verification	The verification code is send to the registered email id.
FR-3	User confirmation	The verification code is entered in to the website. After finishing that home page is opened.
FR-4	Process of booking	When the home page is opened there will be a From and To options. We must enter the details then after that we can able to see the number of trains availability and seats availability. We can select the particular train and particular seats which we need and click the confirm option.
FR-5	Payment process	After entering all the details select the payment option like UPI apps , Net-banking , etc., When we select the comfortable method then it process through selected payment option then payment can be done carefully and securely, then the ticket will be confirmed. After confirmation it will return to the page and we can see the details of booking.
FR-6	Confirmation message	After all the process has been completed the QR code will be send to both mobile number(via SMS) and email id. QR code will be shown to the ticket collector where all the booking details can be viewed by scanning the QR code.

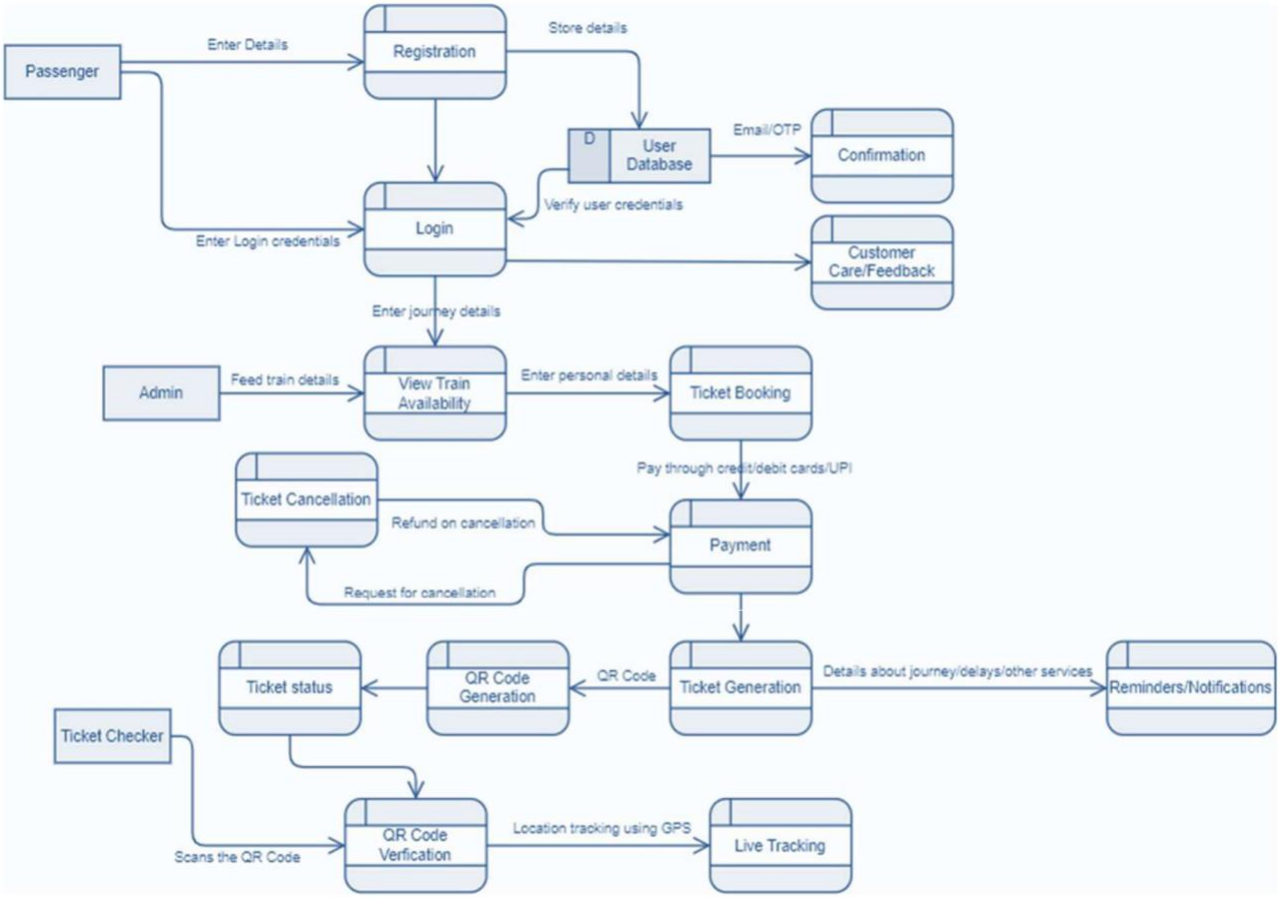
4.2. NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This website is a set of easy methods so there will be no difficulties of operating it. All the languages can be accessed through by user.
NFR-2	**Security**	The permission required is only the access of location and there won't be any other unauthorized permission needed.
NFR-3	**Reliability**	If the network connection is disabled While entering the details of user no worries you no need to enter the details again as all the details will be stored automatically.
NFR-4	**Performance**	The website is more secured and it will obtain through the back end. unauthorized person can't access the website.
NFR-5	**Availability**	only the QR code is sended through the message and email id only no other information is included
NFR-6	**Scalability**	At a time more than 300,000 users can obtain .All the data will be stored carefully without any issues.

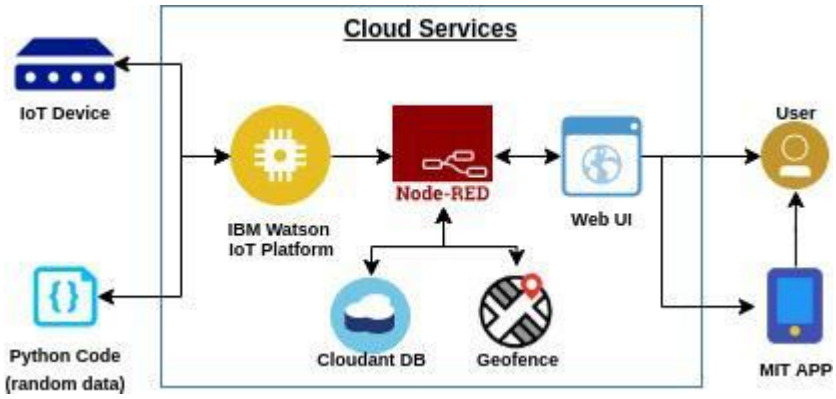
CHAPTER 5

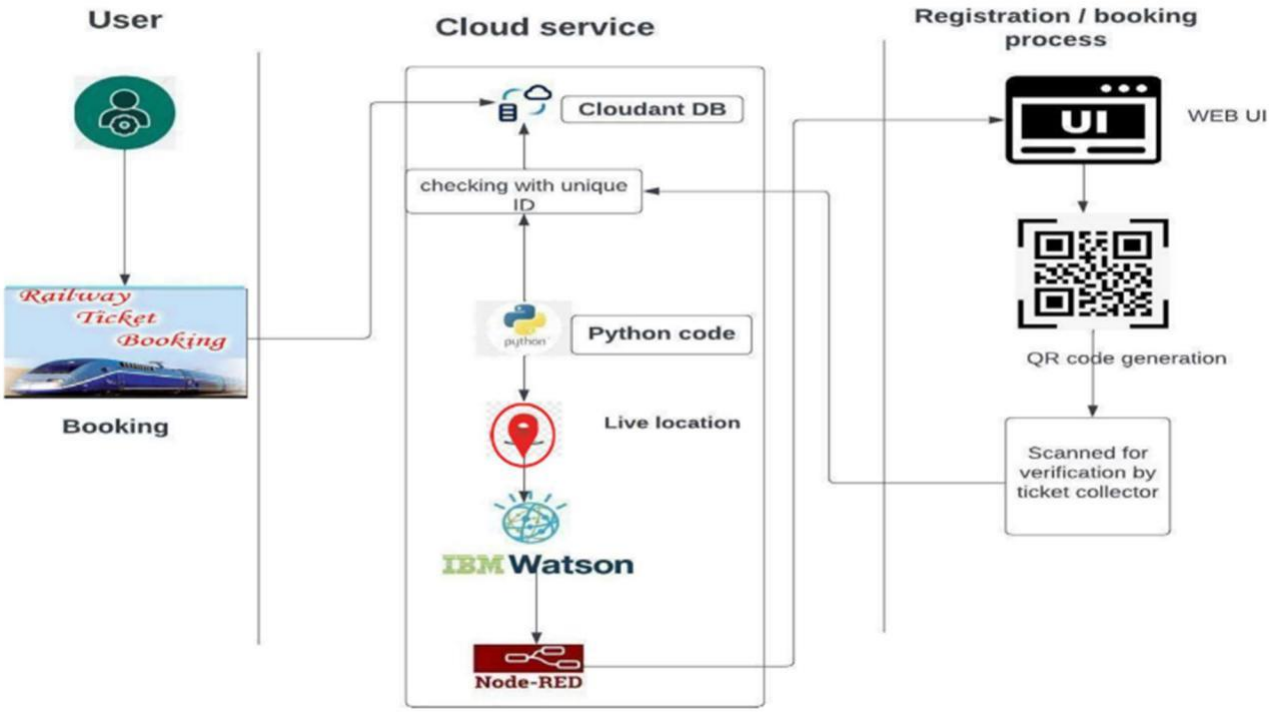
PROJECT DESIGN

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, Web user)	Registration	USN-1	As a user, I can register through the form by Filling in my details	I can register and create my account / dashboard	High	Sprint-1
		USN-2	As a user, I can register through phone numbers, Gmail, Facebook or other social sites	I can register & create my dashboard with Facebook login or other social sites	High	Sprint-2
	Conformation	USN-3	As a user, I will receive confirmation through email or OTP once registration is successful	I can receive confirmation email & click confirm.	High	Sprint-1
	Authentication/ Login	USN-4	As a user, I can login via login id and password or through OTP received on register phone number	I can login and access my account/dashboard	High	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	I can view the train details (name & number), corresponding routes it passes through based on the start and destination entered.	High	Sprint-1
	Booking	USN-6	As a use, I can provide the basic details such as a name, age, gender etc…	I will view, modify or confirm the details enter.	High	Sprint-1
		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.	I will view, modify or confirm the seat/class berth selected	High	Sprint-1
		USN-9	As a user, I will be redirected to the selected Payment gateway and upon successful	I can pay through the payment portal and confirm the booking if any changes need to	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			completion of payment I'll be redirected to the booking website.	be done I can move back to the initial payment page		
	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.	I can show the generated QR code so that authentication can be done quickly.	High	Sprint-1
	Ticket status	USN-11	As a user, I can see the status of my ticket Whether it's confirmed/waiting/RAC.	I can confidentially get the Information and arrange alternate transport if the ticket isn't Confirmed	High	Sprint-1
	Remainders notification	USN-12	As a user, I get remainders about my journey A day before my actual journey.	I can make sure that I don't miss the journey because of the constant notifications.	Medium	Sprint-2
		USN-13	As a user, I can track the train using GPS and can get information such as ETA, Current stop and delay.	I can track the train and get to know about the delays pian accordingly	Medium	Sprint-2
	Ticket cancellation	USN-14	As a user, I can cancel my tickets if there's any Change of plan	I can cancel the ticket and get a refund based on how close the date is to the journey.	High	Sprint-1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

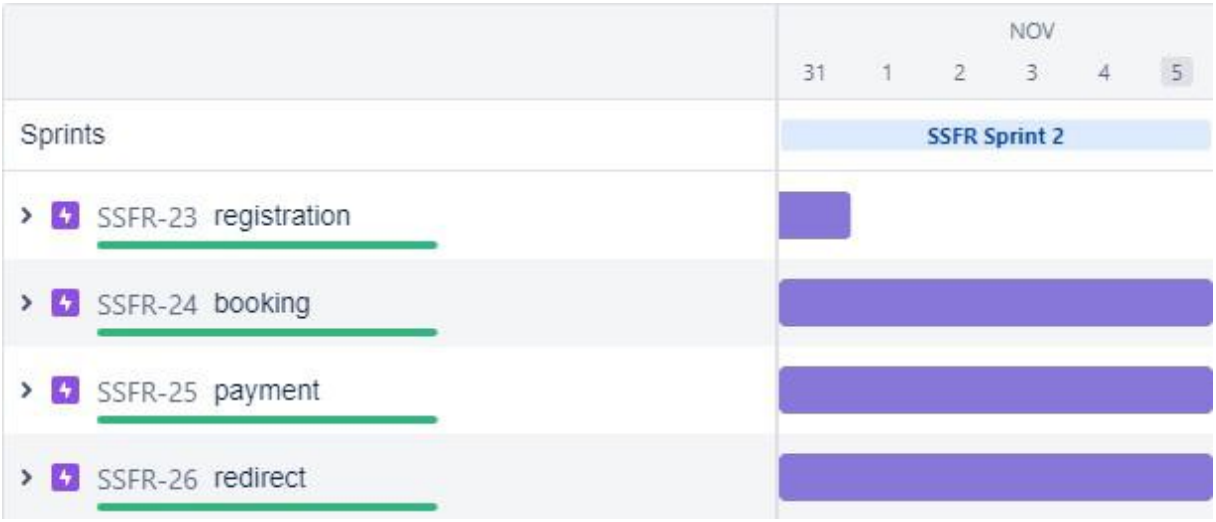
6.1. SPRINT PLANNING& ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dashboard	USN-4	Displaying User the Services we provide in Application	2	Medium	Overall Team
Sprint-2	Overall UI	USN-5	Non functional overall UI developed for User convenience	2	High	Overall Team
Sprint-2	Backend development and QRcode generation	USN-6	Giving User the semi functional UI of the Application	2	High	Kishore R
Sprint-2	Connecting Application with Cloud	USN-7	User can experience the semi functional working of the data(from cloud) with Application	2	Medium	Kishore R
Sprint-3	Connecting Devices/Gateways withCloud	USN-8	User can experience the semi functional working of Iotdevice connected with Application	2	Medium	Overall Team
Sprint-3	Full functional UI	USN-9	Giving User the full functional UI of the Application with supported backended	2	High	Overall Team
Sprint-4	Debug and Testing	USN-10	Testing has been done for finding debugs	2	High	Overall Team
Sprint-4	Testing and Deployment	USN-11	Final Testing for Deployment	2	High	Overall Team

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	6	6 Days	25 Oct 2022	29 Oct 2022	To be Updated(to be 6)	29 Oct 2022
Sprint-2	6	6 Days	30 Oct 2022	07 Nov 2022	To be Updated (to be 6)	07 Nov 2022
Sprint-3	4	6 Days	09 Nov 2022	12 Nov 2022	To be Updated (to be 4)	12 Nov 2022

6.3. REPORTS FROM JIRA



CHAPTER 7

CODING AND SOLUTIONING

7.1. FEATURE 1

IOT device

- IBM Watson
- platform Node red
- Cloudant
- DB Web UI

Geofence

Python code

7.2. FEATURE 2

Verification

Ticket Booking

- Qr code generation
- Qr code verification
- Scanning qr code

Python Code for ticket verification:

```
import cv2 as cv

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

import wiotp.sdk.device

authenticator=BasicAuthenticator('apikey-v2-
1w8tqt2prt3j7qz9d1rgrxhar3w9v43i2359u79ut5jb','86181a38eca19ae487f512b10aca0c80')
service=CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-
1w8tqt2prt3j7qz9d1rgrxhar3w9v43i2359u79ut5jb:86181a38eca19ae487f512b10aca0c80@9163f25
a-

10b8-4374-a8de-cb92e4357567-bluemix.cloudantnosqldb.appdomain.cloud')

cap = cv.VideoCapture(0)

font = cv.FONT_HERSHEY_PLAIN

if not cap.isOpened():
print("Cannot open camera")

exit()


myConfig = {

"identity" :{

"orgId":"ryc4pr",
"typeId":"QR_Reads",

"deviceId":"876543"

},
"auth":{

"token":"GGHvsi!XL-i7x0mC6B"

}

}
def myCommandCallback(cmd):

print("Message received fromIBM 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=response, qos=0,
onPublish=None)
print("Published data Successfully: %s",response)
```

```
print("\n")

while True:

ret, frame=cap.read()

decodedObjects = pyzbar.decode(frame)
if not ret:

print("Can't receive frame (stream end?). Exiting ...")

break

for obj in decodedObjects:
a=obj.data.decode('UTF-8')

cv.putText(frame, "Ticket", (50,50),font,2,

(255 ,0, 0),3)

try:

response=service.get_document(
db='bookingdetails',

doc_id = a

) .get_result()

print(response)
print("\n\n")

pub(response)

time.sleep(5)
except Exception as e:

response={'Error':'Not a Valid Ticket'}

pub(response)

print("Not a Valid Ticket")
print("\n\n")

time.sleep(5)

cv.imshow("Frame" ,frame)

if cv.waitKey(1) & 0xFF == ord('q'):
break

client.commandCallback = myCommandCallback

cap.release()

cv.destroyAllWindows()
client.disconnect()
```

Python code for train tracking:

```
import wiotp.sdk.device
import time

import random

myConfig = {

"identity" :{
"orgId":"ytluse",

"typeId":"2702",
"deviceId":"12345"

},
"auth":{

"token":"O+n)Eh+lNX0y3?rG!8"
}

}
def myCommandCallback(cmd):

print("Message received fromIBM 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':'Delhi Express','lat':13.344279,'lon':80.214367}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':13.515254,'lon':80.093518}
pub(myData)
```

```
time.sleep(3)
myData = {'name':'Delhi Express','lat':13.728799,'lon':80.005627}

pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':13.910160,'lon':79.906750}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':14.102035,'lon':79.851819}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':14.261807,'lon':79.862805}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':14.623537,'lon':79.950695}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':15.111987,'lon':79.994641}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':15.313413,'lon':80.005627}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':15.567568,'lon':80.104504}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':15.747405,'lon':80.269299
}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':15.821409,'lon':80.302258}
pub(myData)
```

```
time.sleep(3)

myData = {'name':'Delhi Express','lat':15.927082,'lon':80.445080}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':16.022141,'lon':80.554943}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':17.033801,'lon':80.295512}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':18.383088,'lon':18.383088}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':19.074762,'lon':79.487698}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':20.179065,'lon':79.001439}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':21.306421,'lon':78.789356}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':22.518024,'lon':77.829404}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':23.264139,'lon':77.429333}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':24.509723,'lon':78.330212}
```

```
pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':25.668840,'lon':78.451062}
pub(myData)

time.sleep(3)

myData = {'name':'Delhi Express','lat':26.177704,'lon':78.170910}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':27.505914,'lon':77.676526}

pub(myData)
time.sleep(3)

myData = {'name':'Delhi Express','lat':28.302041,'lon':77.308484}
pub(myData)

time.sleep(3)
client.commandCallback = myCommandCallback

client.disconnect()
```

Train 2:

```
import wiotp.sdk.device

import time
import random

myConfig = {
"identity" :{

"orgId":"ytluse",
"typeId":"train2",

"deviceId":"mysore"
},

"auth":{
"token":"-BCOX+hMk?*@xc@AV9"

}
}

def myCommandCallback(cmd):
```

```
print("Message received fromIBM 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':'Mysuru SF Express','lat':11.024938,'lon':76.982315} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':11.220325,'lon':77.570083} pub(myData)

time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':11.564960,'lon':77.993057} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':11.780142,'lon':78.037002} pub(myData)

time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.134824,'lon':78.130386} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.226105,'lon':78.091934} pub(myData)

time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.344187,'lon':78.037002} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.489034,'lon':78.009536} pub(myData)

time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.655239,'lon':77.866714} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.735622,'lon':77.756851} pub(myData)

time.sleep(3)
```

```
myData = {'name':'Mysuru SF Express','lat':12.907020,'lon':77.696426} pub(myData)

time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.987323,'lon':77.646988} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.955205,'lon':77.509659} pub(myData)

time.sleep(3)
myData = {'name':'Mysuru SF Express','lat':12.665958,'lon':77.136123

}
pub(myData)

time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.548022,'lon':76.921890} pub(myData)
time.sleep(3)

myData = {'name':'Mysuru SF Express','lat':12.336809,'lon':76.644485} pub(myData)

time.sleep(3)
client.commandCallback = myCommandCallback

client.disconnect()
```

Train 3:

```
import wiotp.sdk.device

import time
import random

myConfig = {
"identity" :{

"orgId":"ytluse",
"typeId":"2702",

"deviceId":"678910"

},
"auth":{

"token":"4skuC*xkhDH&la(CUR"
}
```

```
}
def myCommandCallback(cmd):

print("Message received fromIBM 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':'Goa Express','lat':11.688572, 'lon':78.098877}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':11.711433, 'lon':78.076905}
pub(myData)

time.sleep(3)
myData = {'name':'Goa Express','lat':11.978226,  'lon':78.116730}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':12.085676, 'lon': 78.119477}
pub(myData)

time.sleep(3)
myData = {'name':'Goa Express','lat':12.402400, 'lon':78.023347}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':12.884795, 'lon':77.707490}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':13.018630,'lon':77.614106}
pub(myData)
```

```
time.sleep(3)
myData = {'name':'Goa Express','lat':13.334194, 'lon':77.086762}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':13.299448, 'lon':76.858796}
pub(myData)

time.sleep(3)
myData = {'name':'Goa Express','lat':13.344884,'lon': 76.205109}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':13.619985, 'lon':75.966157}
pub(myData)

time.sleep(3)
myData = {'name':'Goa Express','lat':13.974739, 'lon':76.119965}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':14.423398, 'lon':75.949677}
pub(myData)

time.sleep(3)
myData = {'name':'Goa Express','lat':14.922914, 'lon':75.389374}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':15.119216, 'lon':75.389374}
pub(myData)

time.sleep(3)
myData = {'name':'Goa Express','lat':15.449980, 'lon':74.406230}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':15.352006,'lon':74.307353}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':15.314922, 'lon':74.218089}
pub(myData)
```

```
time.sleep(3)
myData = {'name':'Goa Express','lat':15.283131, 'lon':74.146678}

pub(myData)
time.sleep(3)

myData = {'name':'Goa Express','lat':15.276839, 'lon':74.129855}
pub(myData)

time.sleep(3)
time.sleep(3)

myData = {'name':'Goa Express','lat':15.282800, 'lon':74.125392}
pub(myData)

time.sleep(3)
time.sleep(3)

myData = {'name':'Goa Express','lat':15.296378,'lon':74.135692}
pub(myData)

time.sleep(3)
client.commandCallback = myCommandCallback

client.disconnect()
```

CHAPTER 8

TESTING

8.1.TEST CASES

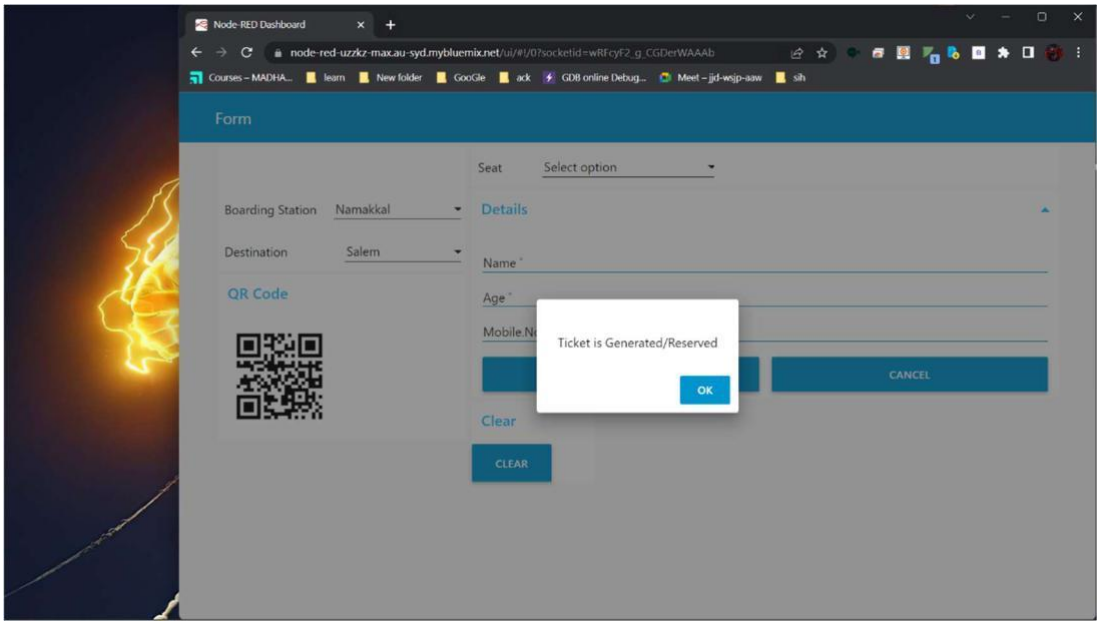
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automati
1	Functional	Registration	Registration through the form by Filling in my details		1.Click on register 2.Fill the registration form 3.click Register		Registration form to be filled is to be displayed	Working as expected	Pass		
2	UI	Generating OTP	Generating the otp for further process		1.Generating of OTP number		user can register through phone numbers, Gmail, Facebook or other social sites and to get oto number	Working as expected	pass		
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	OTP verifed is to be displayed	Working as expected	pass		
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	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass		
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: Testing123678687686876	A user can view about the available trains to enter start and destination details	Working as expected	fail		

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
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		Tickets booked to be displayed	Working as expected	Pass			
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			
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		payment for the booked tickets to be done using payment method through either the following methods credit Card/debit card/UPI.	Working as expected	pass			
Functional	Redirection	user can be redirected to the selected		1.After payment the usre will be redirected to the previous		After payment the usre will be redirected to the previous page	Working as expected	pass			

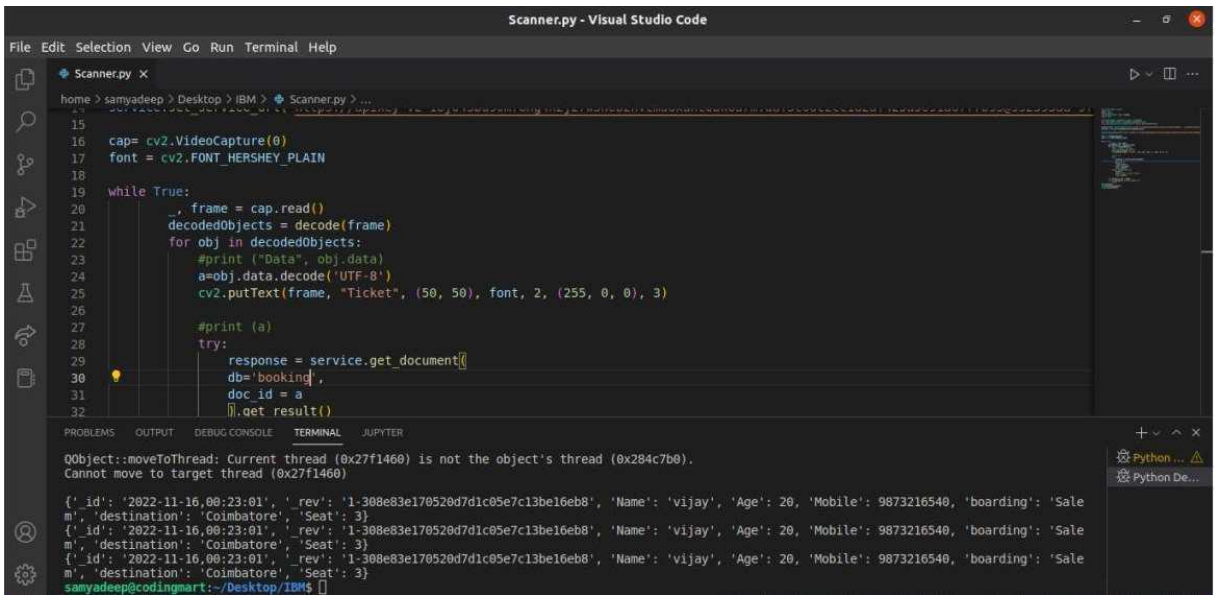
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisit	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Autom	BUG ID
10	Functional	Ticket generation	a user can download the generated e ticket for my journey along with the QR code which is used for authentication 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			
11	UI	Ticket status	a usercan see the status of my ticket Whether it's confirmed/waiting/RAC		1.known to the status of the tivkets booked		known to the status of the tivkets booked	Working as expected	pass			
12	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			
13	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		tracking process through GPS	Working as expected	pass			

CHAPTER 9
RESULTS

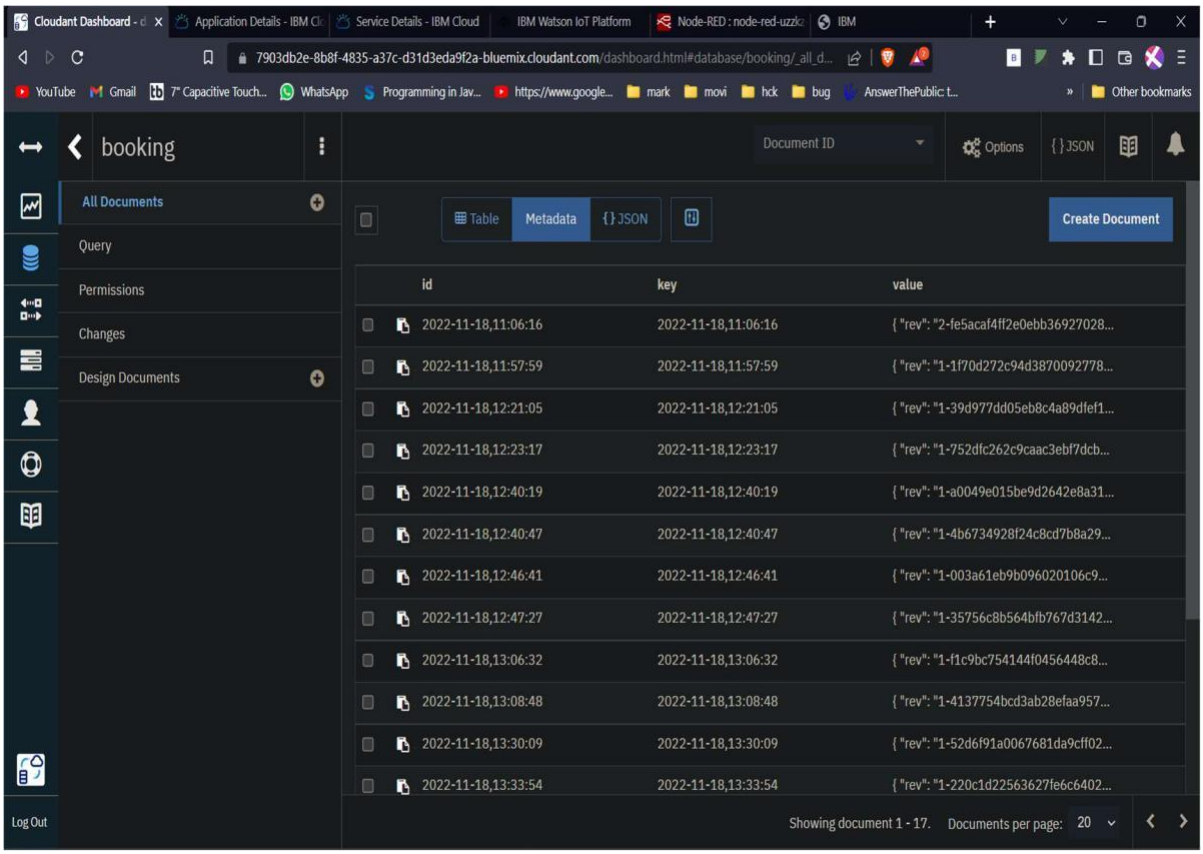
9.1.PERFORMANCE METRICS



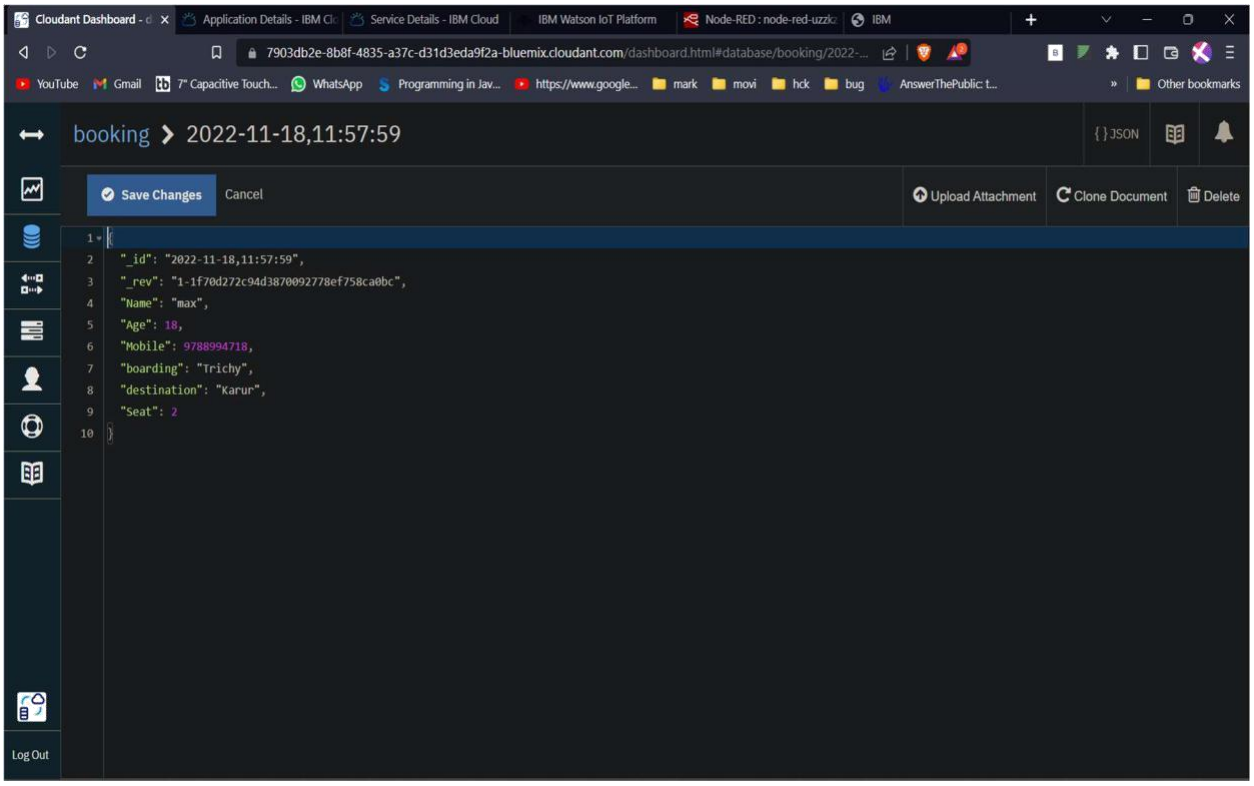
Ticket generated



QR-code verified

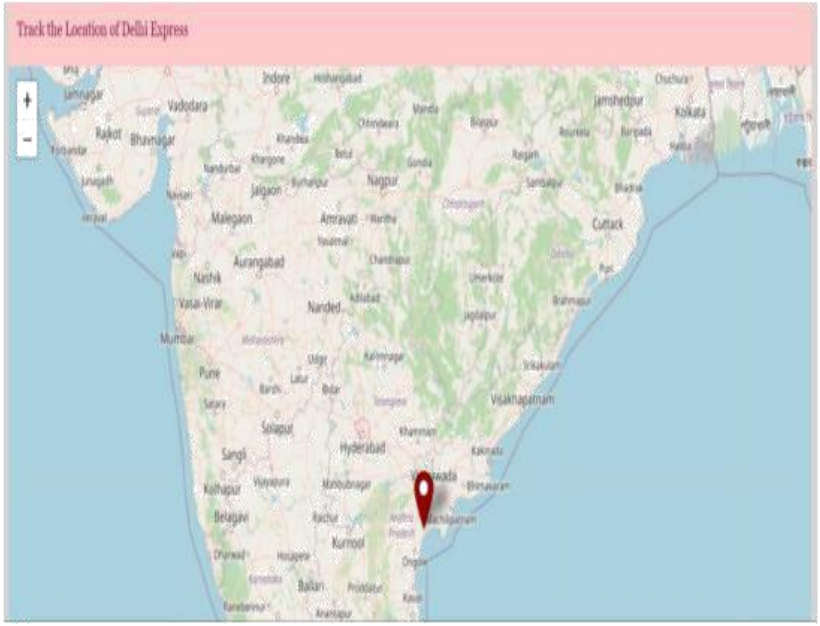


Cloud-database



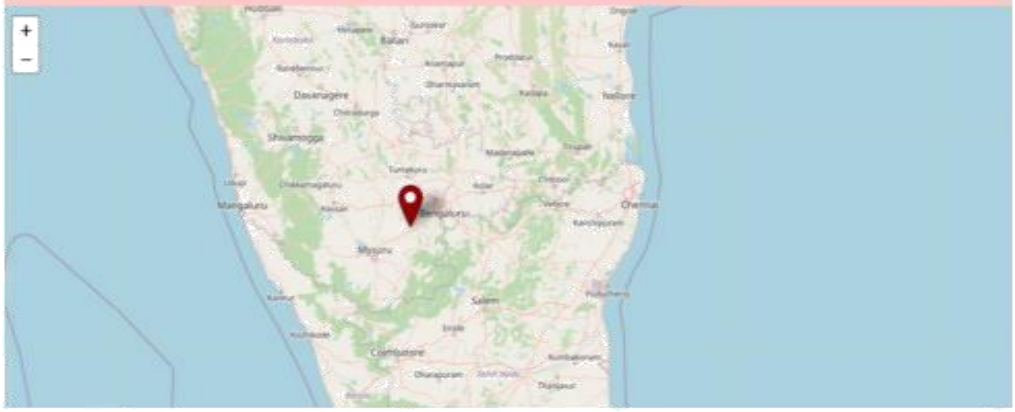
Ticket information

Train Tracking:



Track the Location of Goa Express

Track the Location of Mysuru SF Express



CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1. 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.

10.2. 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.

CHAPTER 11

CONCLUSION

Online railway ticket reservation system was successfully designed and developed as per the specifications. We have created a project to make the user and the worker to reduce the time consumption for the issue and buy a ticket. It is also reduce the fear of lost the ticket .from this project we will generate the qrcode it is more helpful to have all details in the qrcode itself. While ticket checker scan the qrcode,it will show all detail about the passenger and the details of the ticket .

CHAPTER 12

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 over.

CHAPTER 13

APPENDIX

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-6987-1658844789

Project Demo Link:

https://drive.google.com/file/d/1dGw_iXeoXW44Uji6zHdWF7B3rkJMj9V-/view?usp=share_link