
# SMART SOLUTION FOR RAILWAYS

## A PROJECT REPORT

***Submitted by***

**ASAN MYDEEN A (95071912012)**

**ESAKKI MUTHU M (95071912025)**

**FAYAZ AHAMED R(95071912028)**

**GUROO SUNTAR (95071912030)**

# Project Report

1. INTRODUCTION
    a. Project Overview
    b. Purpose
2. LITERATURE SURVEY
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. IDEATION & PROPOSED SOLUTION
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. REQUIREMENT ANALYSIS
    a. Functional requirement
    b. Non-Functional requirements
5. PROJECT DESIGN
    a. Data Flow Diagrams
    b. Solution & Technical Architecture
    c. User Stories
6. PROJECT PLANNING & SCHEDULING
    a. Sprint Planning & Estimation
    b. Sprint Delivery Schedule
    c. Reports from JIRA
7. CODING & SOLUTIONING (Explain the features added in the project along with code)
    a. Feature 1
    b. Feature 2
    c. Database Schema (if Applicable)
8. TESTING
    a. Test Cases
    b. User Acceptance Testing
9. RESULTS
    a. Performance Metrics
10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX Source Code

# INTRODUCTION

## A.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.

## B. 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 humanto-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

## A.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.

## B. 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.

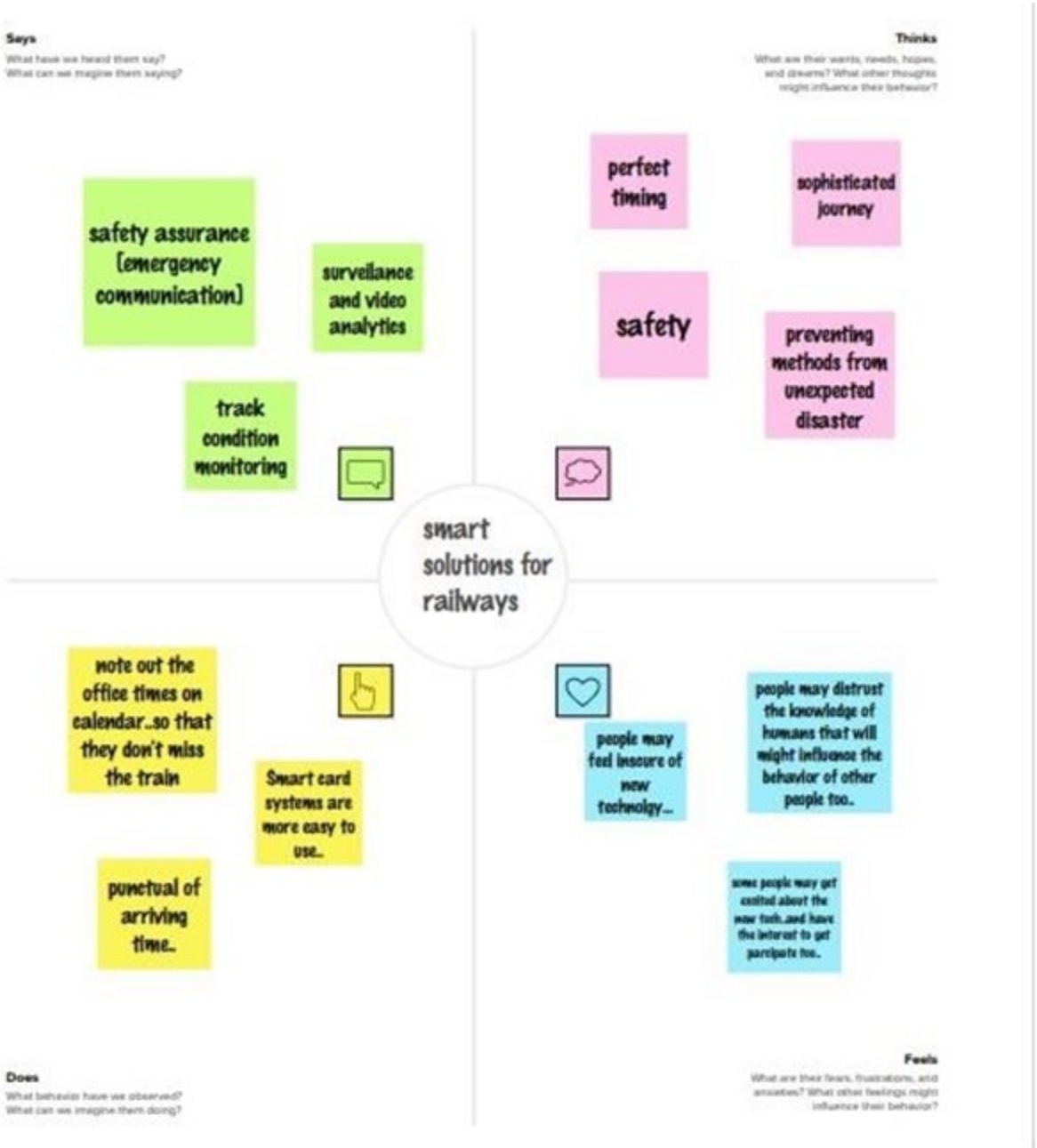
6. R. A. Raza, K. P. Rauf, A. Shafeeq, “Crack detection in Railway track using Image processing”, IJARIIT, Vol. 3, pp. 489-496, Issue 4, 2017.

## C.PROBLEM STATEMENT DEFINITION

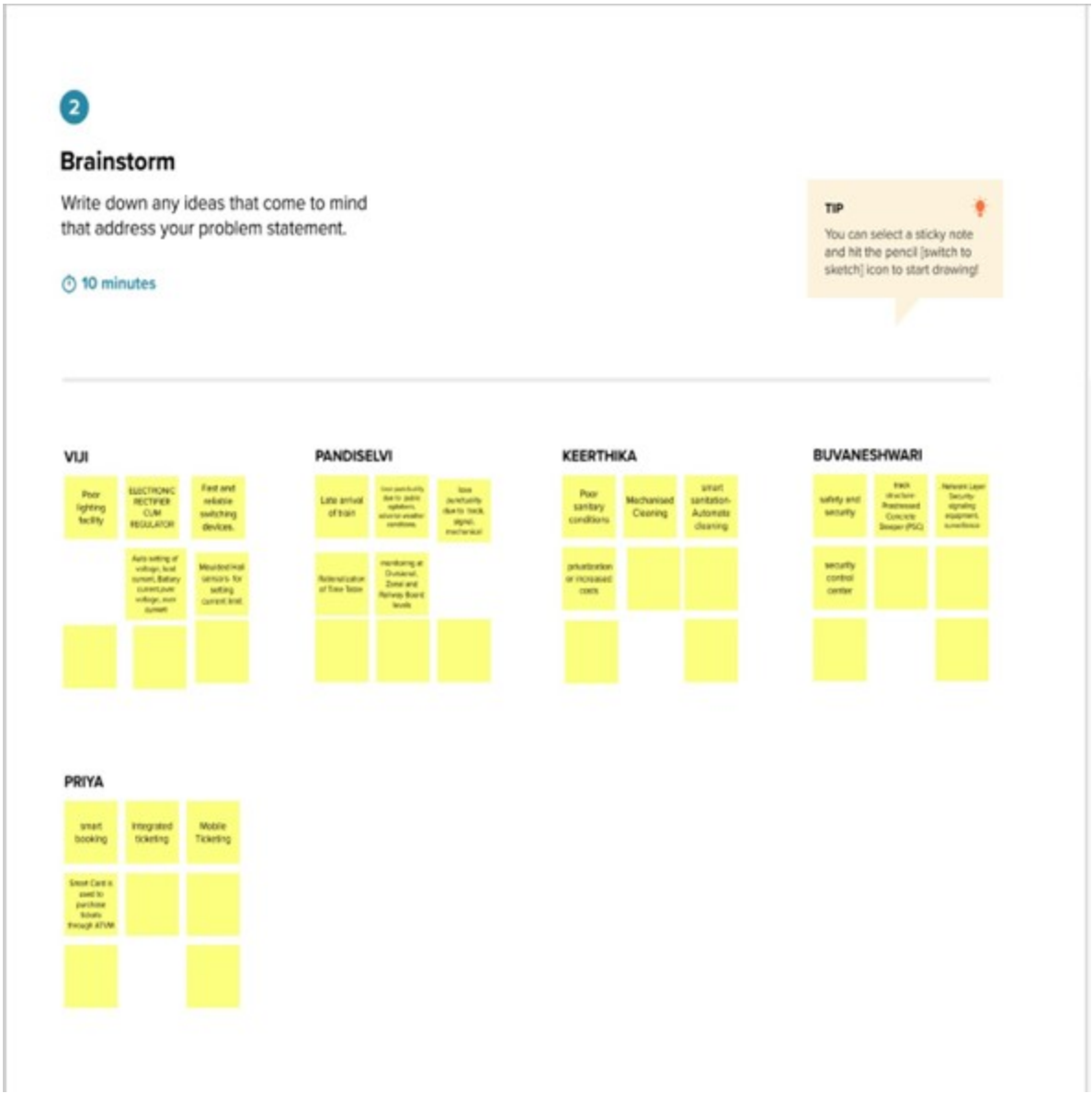
Among the various modes of transport, railways is one of the biggest modes of transport in the world. Though there are competitive threats from airlines, luxury buses, public transports, and personalized transports the problem statement is to answer the question “What are the problems faced by the passengers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

# IDEATION AND PROPOSED SOLUTION

# A.EMPATHY MAP CANVAS



# B.IDEATION & BRAINSTORMING

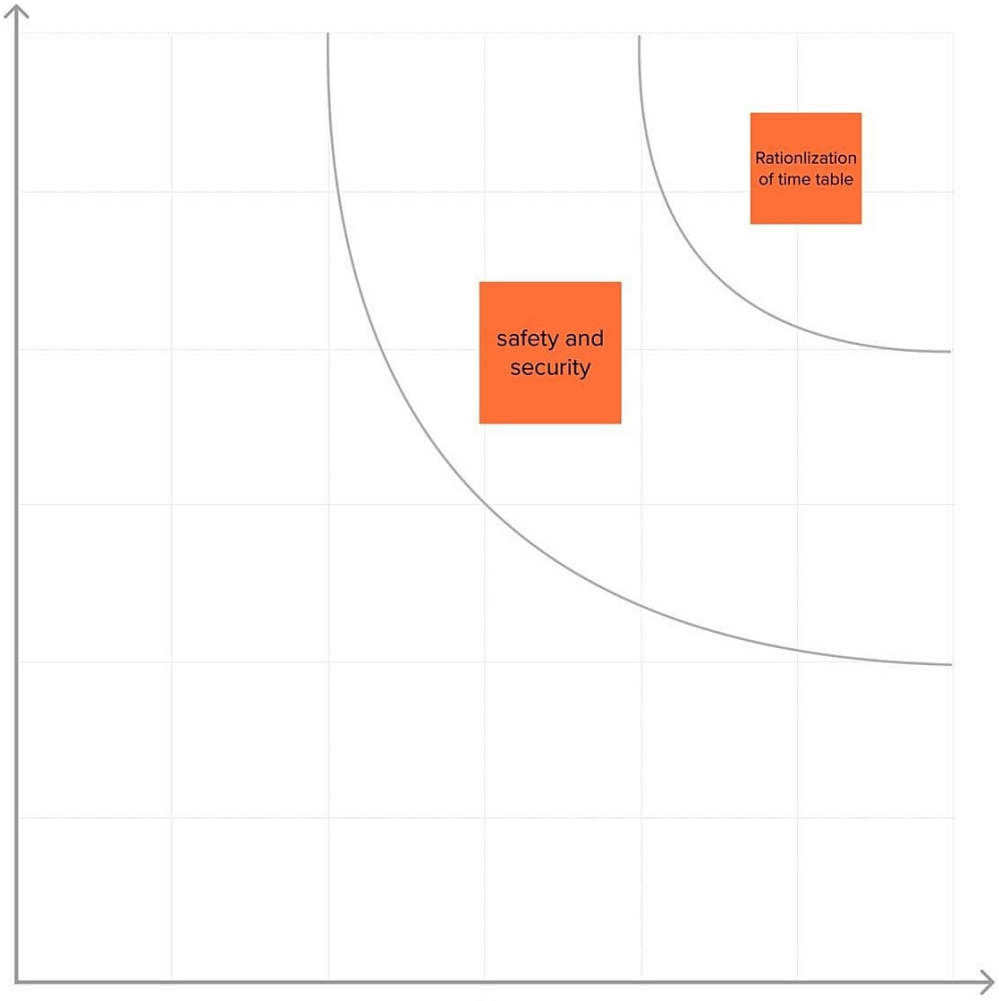


4

## Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes




**Importance**

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

**Feasibility**

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

## C.PROPOSED SOLUTION

| S.NO | PARAMETERS | DESCRIPTIONS |
|---|---|---|
| 1 | Problem Statement (Problem to be solved) | In order to satisfy the passengers, the Railways provides various services to its passengers But, the passengers can face some problems. |
| 2 | Idea / Solution description | The idea is to minimize the ticket booking problems among the passengers by providing Online mode of booking rather than papers. . In queues in front of the ticket counters in railway stations have been drastically increased over the time. |
| 3 | Novelty / Uniqueness | Online mode of booking is most common and so ease of access to everyone that makes more efficient uniqueness of utilizing the technique. People can book their ticket through online and they get a QR code through SMS |
| 4 | Social Impact / Customer Satisfaction | Customers for sure they get satisfied as they are in the fast roaming world this technique makes more easier for travelling passengers. A web page designed in which the user can book tickets and will be provided with the QR code, which will be shown to the ticket collector and by scanning the QR codticketcollector will get the passenger details |

| | | |
|---|---|---|
| 5 | Business Model (Revenue Model) | A web page is designed in which<br>the user can book tickets and will be provided with the QR code, which will be shown to the ticket collector and by scanning<br>the QR code the ticket collector will get the passenger details. The booking details of the user will be stored in the database, which can be retrieved any time. |
| 6 | Scalability of the Solution | The scalability of this solution is most feasible among the passengers who are willing to travel. No need of taking printout Counter ticket has to be handled with care, but SMS on mobile is enough. No need to taking out wallet and showing your ticket to TTR just tell your name to TTR that you are a passenger with valid proof. |

# D.PROBLEM SOLUTION FIT

# REQUIREMENT ANALYSIS

## A.FUNCTIONAL REQUIREMENTS

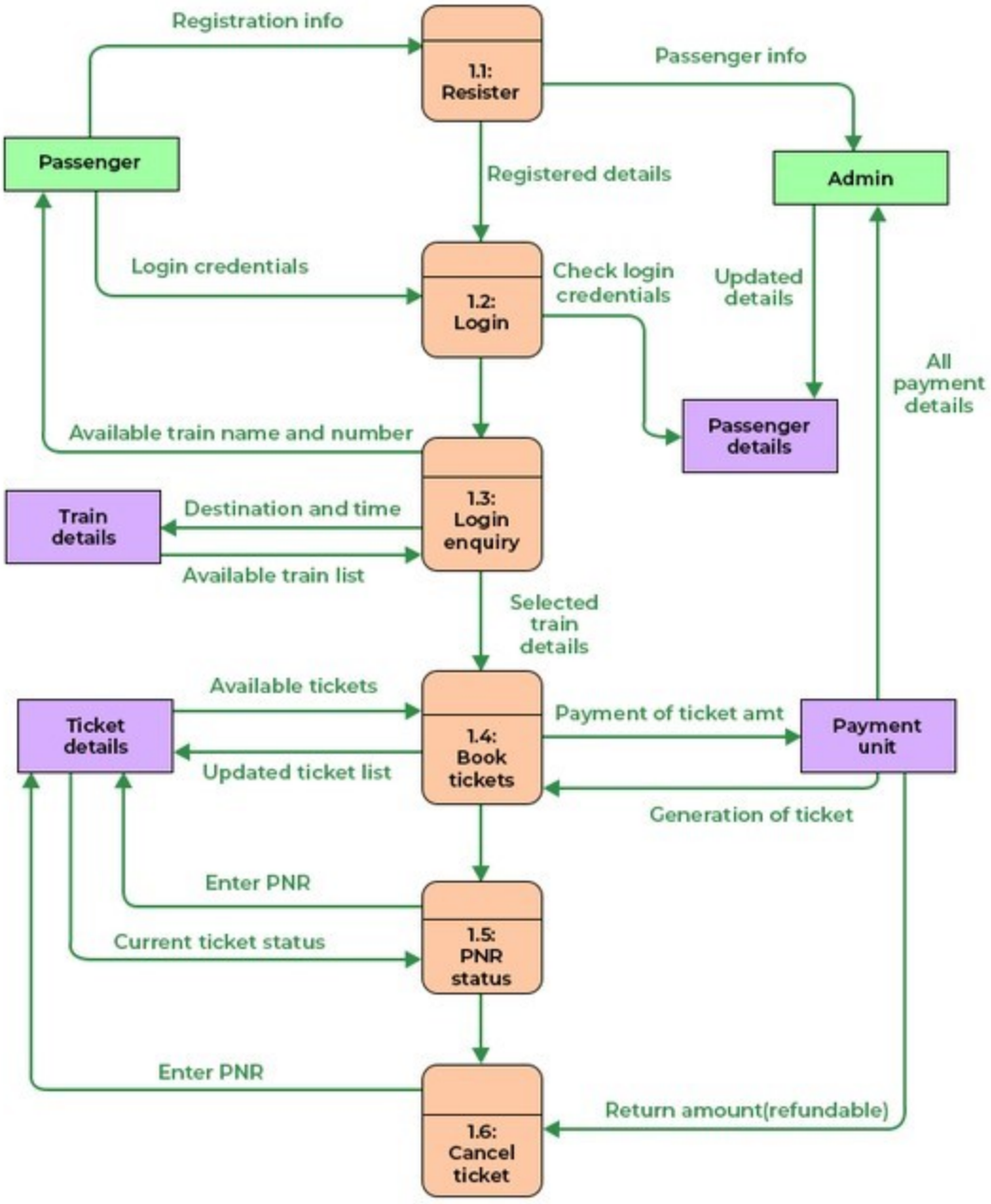
| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | Unique accounts | 1. Every online booking needs to be associated with an account<br>2. One account cannot be associated with multiple users |
| FR-2 | Booking options | □ Search results should enable users to find the most recent and relevant booking options |
| FR-3 | Mandatory fields | □ System should only allow users to move to payment only when mandatory fields such as date, time, location has been mentioned |
| FR-4 | Synchronization | □ System should consider timezone synchronisation when accepting bookings from different timezones |
| FR-5 | Authentication | □ Booking confirmation should be sent to user to the specified contact details |

## B.NON-FUNCTIONAL REQUIREMENTS

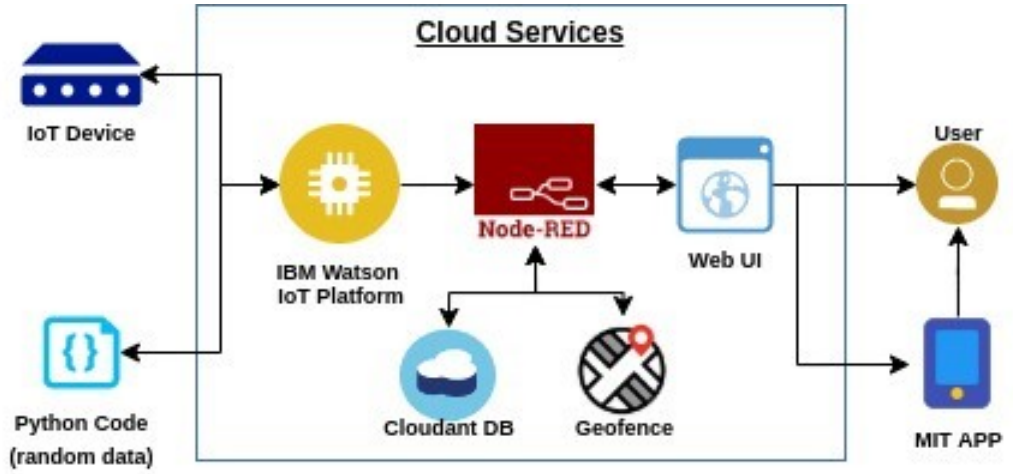
| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Search results should populate within acceptable time limits |
| NFR-2 | **Security** | System should visually confirm as well as send booking confirmation to the user's contact |
| NFR-3 | **Reliability** | System should accept payments via different payment methods, like PayPal, wallets, cards, vouchers, etc |
| NFR-4 | **Performance** | Search results should populate within acceptable time limits |
| NFR-5 | **Availability** | User should be helped appropriately to fill in the mandatory fields, incase of invalid input |
| NFR-6 | **Scalability** | Use of captcha and encryption to avoid bots from booking tickets |

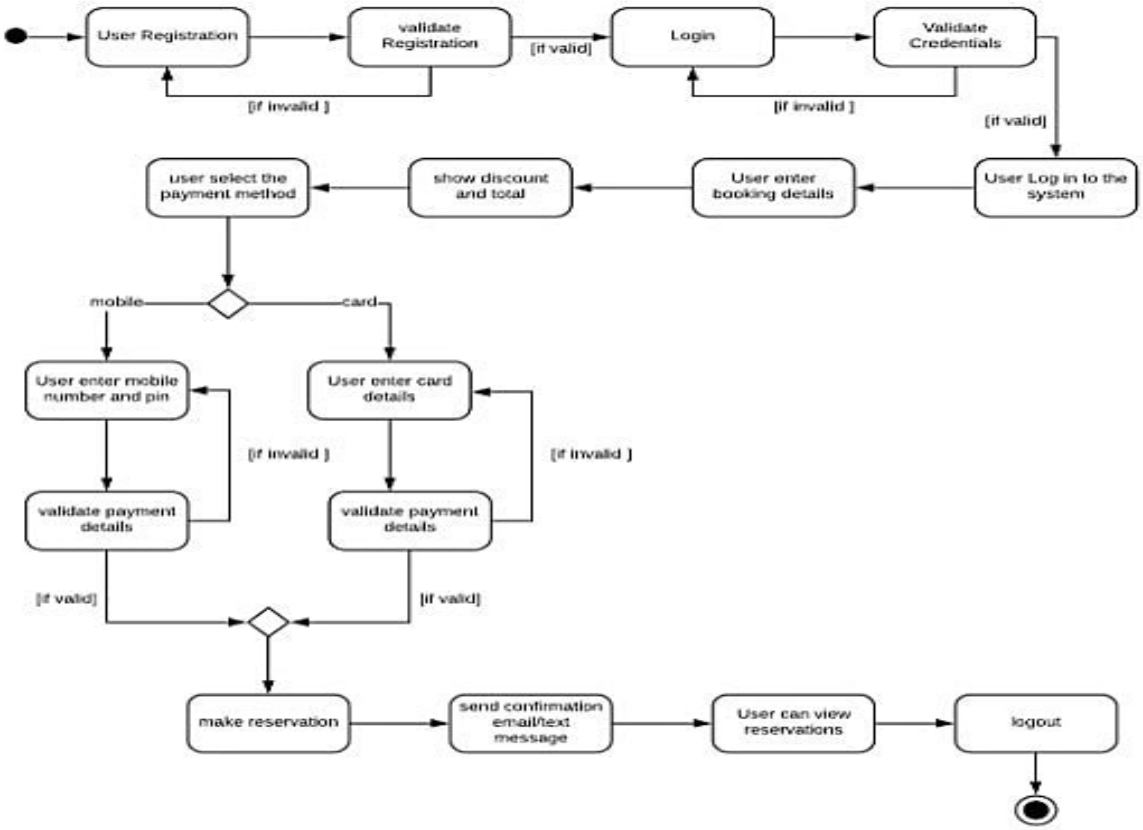
# PROJECT DESIGN

# A.DATA FLOW DIAGRAMS



# B.SOLUTION & TECHNICAL ARCHITECTURE



# C.USER STORIES

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer<br><br>(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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Payment | USN-8 | As a user, I can choose to pay through credit Card/debit card/UPI. | I can view the payment Options available and select my desirable choice To proceed with the payment | 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 |
| | Raise queries | USN-15 | As a user, I can raise queries through the query box or via mail. | I can view my pervious queries. | Low | Sprint-2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Customer care Executive | Answer the queries | USN-16 | As a user, I will answer the questions/doubts Raised by the customers. | I can view the queries and make it once resolved | Medium | Sprint-2 |
| Administrator | 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. | I can view and ensure the corrections of the information fed. | High | Sprint-1 |

# PROJECT PLANNING AND SCHEDULING

# A.SPRINT PLANNING& ESTIMATION

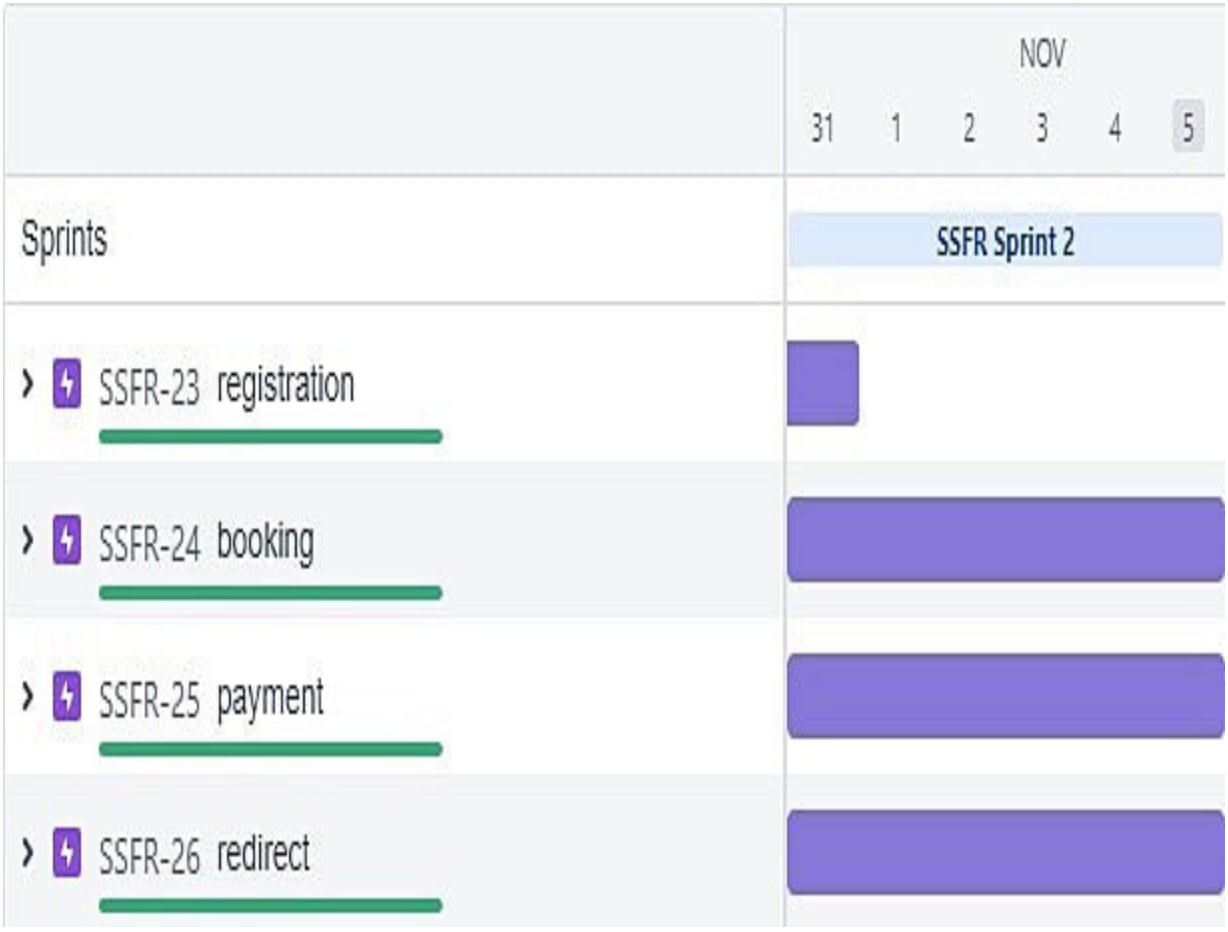
| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority |
|---|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | As a user, I can register through the form by Filling in my details | 2 | High |
| Sprint-1 | | USN-2 | As a user, I can register through phone numbers, Gmail, Facebook or other social sites | 1 | High |
| Sprint-1 | Conformation | USN-3 | As a user, I will receive confirmation through email or OTP once registration is successful | 2 | Low |
| 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 |
| 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 |
| Sprint-2 | Booking | USN-6 | As a use, I can provide the basic details such as a name, age, gender etc… | 2 | High |
| 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 |
| Sprint-2 | Payment | USN-8 | As a user, I can choose to pay through credit Card/debit card/UPI. | 1 | High |
| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority |
| Sprint-2 | | USN-9 | As a user, I will be redirected to the selected | 2 | High |

| | | | | | |
|---|---|---|---|---|---|
| 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 |
| 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 |
| Sprint-3 | Remainders notification | USN-12 | As a user, I get remainders about my journey A day before my actual journey. | 1 | High |
| 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 |
| Sprint-4 | | USN-14 | As a user, I can cancel my tickets if there's any Change of plan | 1 | High |
| Sprint-4 | Raise queries | USN-15 | As a user, I can raise queries through the query box or via mail. | 2 | Medium |
| Sprint-4 | Answer the queries | USN-16 | As a user, I will answer the questions/doubts Raised by the customers. | 2 | High |
| 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 |

## B.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 | 5 Nov 2022 |
| 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 |

## C.REPORTS FROM JIRA




NOV
13
14
15
16
17
18
19
Sprints
SSFR Sprint 4
SSFR-23 registration
SSFR-24 booking
SSFR-25 payment
SSFR-26 redirect
SSFR-27 ticket generation\
SSFR-28 status
SSFR-29 notification
SSFR-30 tracking location
SSFR-31 cancellation
SSFR-32 raise queries
SSFR-33 ans queries
SSFR-34 feed details

# CODING AND SOLUTIONING

## A. FEATURE 2

1. IOT device
2. IBM Watson platform
3. Node red
4. Cloudant DB
5. Web UI
6. Geofence  MIT App
7. Python code

## B. FEATURE 2

1. Registration
2. Login
3. Verification
4. Ticket Booking
5. Payment
6. Ticket Cancellation
7. Adding Queries

```
labl_0 = Label(base, text="Registration
form",width=20,font=("bold",
20))
labl_0.place(x=90,y=53)


lb1= Label(base, text="Enter Name", width=10,
font=("arial",12))   lb1.place(x=20, y=120)   en1=
Entry(base)
en1.place(x=200, y=120)

lb3= Label(base, text="Enter Email", width=10,
font=("arial",12))   lb3.place(x=19, y=160)   en3=
Entry(base)
en3.place(x=200, y=160)

lb4= Label(base, text="Contact Number",
width=13,font=("arial",12))   lb4.place(x=19, y=200)
en4= Entry(base)
en4.place(x=200, y=200)

lb5= Label(base, text="Select Gender", width=15,
font=("arial",12))   lb5.place(x=5, y=240)
var = IntVar()

Radiobutton(base, text="Male", padx=5,variable=var,
value=1).place(x=180, y=240)
```

```
Radiobutton(base, text="Female", padx
=10,variable=var, value=2).place(x=240,y=240)

Radiobutton(base, text="others", padx=15,
variable=var, value=3).place(x=310,y=240)

list_of_cntry = ("United States", "India", "Nepal",
"Germany")   cv = StringVar()   drplist=
OptionMenu(base, cv, *list_of_cntry)
drplist.config(width=15)   cv.set("United States")
lb2= Label(base, text="Select Country",
width=13,font=("arial",12))
lb2.place(x=14,y=280)
drplist.place(x=200, y=275)

lb6=  Label(base,  text="Enter  Password",
width=13,font=("arial",12))   lb6.place(x=19, y=320)
en6= Entry(base, show='*')
en6.place(x=200, y=320)

lb7= Label(base, text="Re-Enter Password",
width=15,
font=("ar
ial",12))
lb7.place(
x=21,
y=360)
en7
=Entry(b
```

```
ase,
show='*')
en7.place(x=200, y=360)

Button(base, text="Register",
width=10).place(x=200,y=400)   base.mainloop()
 def generateOTP() :

   # Declare
a digits
variable     #
which stores
all digits
digits =
"012345678
9"
   OTP = ""

  # length of
password can be
changed    # by
changing value in
range     for i in
range(4) :
       OTP += digits[math.floor(random.random() * 10)]

   return OTP

# Driver
code if
```

```
__name__
==
"__main_
_" :

   print("OTP of 4 digits:", generateOTP())

digits="0123
456789"
OTP=""
for i in range(6):

OTP+=digits[math.floor(rando
m.random()*10)] otp = OTP + "
is your OTP" msg= otp s =
smtplib.SMTP('smtp.gmail.com
', 587)
s.starttls()
s.login("Your Gmail Account", "You app
password") emailid = input("Enter your
email: ")
s.sendmail('&&&&&&&&&&
&&',emailid,msg) a =
input("Enter Your OTP >>:
") if a == OTP:

print(
"verifi
ed");
else:
print("Please Check your OTP again")
```

# TESTING

# A.TEST CASES

| Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Commnets | TC for Automati | BUG |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Functional | Registration | Registration through the form by Filling in my details | | 1.Click on register<br>2.Fill the registration form<br>3.click Register | | Registration form to be filled is to be displayed | Working as expected | Pass | | | |
| UI | GeneratingOTP | 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 | | | |
| Functional | OTP verification | Verify user otp using mail | | 1.Enter gmail id and enter password<br>2.click submit | Username: abc@gmail.com password: Testing123 | OTP verified is to be displayed | Working as expected | pass | | | |
| Functional | Login page | Verify user is able to log into application with InValid credentials | | 1.Enter into log in page<br>2.Click on My Account dropdown button<br>3.Enter InValid username/email in Email text box<br>4.Enter valid password in password text box<br>5.Click on login button | Username: abc@gmail password: Testing123 | Application should show "Incorrect email or password" validation message. | Working as expected | pass | | | |
| 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: Testing12367868678687 6878 | 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<br>2.Enter name,age,gender<br>3.Enter how many tickets wants to be booked<br>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<br>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 | | | |

| Feature Type | Component | Test Scenario | Pre-Requisit | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Commnets | TC for Autom | BUG ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | | | |
| 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 | | | |
| Functional | Remainder notification | a user, I get remainders about my journey A day before my actual journey | | 1.user can get reminder nolication | | user can get reminder nolication | Working as expected | pass | | | |
| 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 | | | |

| Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | Commnets | TC for Automation(Y | BUG ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Functional | Ticket cancellation | user can cancel my tickets there's any Change of plan | | 1.tickets to be cancelled | | Tickets booked to be cancelled | Working as expected | Pass | | | |
| UI | Raise queries | user can raise queries through the query box or via | | 1,raise the queries | | raise the queries | Working as expected | pass | | | |
| 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 | | | |
| 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 | Working as expected | pass | | | |

# RESULTS

# A.PERFORMANCE METRICS

# ADVANTAGES &DISADVANTAGES

## A. 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.

## B. 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

## A.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

## B. 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

# APPENDIX

## C.SOURCE PROGRAM

```
        import math, random

import os
import
smtplib
import
sqlite3
import
requests
        from bs4 import BeautifulSoup
        from django.contrib.auth.base_user import
AbstractBaseUser         from django.db import models

import logging
import pandas
as pd
import pyttsx3
        from plyer import notification
        import time
import numpy as np
import matplotlib.pyplot as
plt         from PIL
import Image, ImageDraw
from pickle import
load,dump
        import smtplib, ssl
```

```
            from email.mime.text import
MIMEText            from
email.mime.multipart import
MIMEMultipart            import email

          from email import encoders
          from email.mime.base import MIMEBase


            import attr
            from flask import Blueprint, flash,
redirect, request, url_for            from flask.views
import MethodView            from flask_babelplus
import gettext as _
            from flask_login import current_user, login_required
            from pluggy import HookimplMarker

           from tkinter import*
           base = Tk()
           base.geometry("500x
           500")
           base.title("registration form")

           labl_0 = Label(base, text="Registration
           form",width=20,font=("bold",
           20))
           labl_0.place(x=90,y=53)

           lb1= Label(base, text="Enter Name", width=10,
           font=("arial",12))   lb1.place(x=20, y=120)   en1=
           Entry(base)
```

```
en1.place(x=200, y=120)

lb3= Label(base, text="Enter Email", width=10,
font=("arial",12))   lb3.place(x=19, y=160)   en3=
Entry(base)
en3.place(x=200, y=160)

lb4= Label(base, text="Contact Number",
width=13,font=("arial",12))   lb4.place(x=19, y=200)
en4= Entry(base)
en4.place(x=200, y=200)

lb5= Label(base, text="Select Gender", width=15,
font=("arial",12))   lb5.place(x=5, y=240)   var =
IntVar()
Radiobutton(base, text="Male", padx=5,variable=var,
value=1).place(x=180, y=240)
Radiobutton(base, text="Female", padx
=10,variable=var, value=2).place(x=240,y=240)
Radiobutton(base, text="others", padx=15,
variable=var, value=3).place(x=310,y=240)

list_of_cntry = ("United States", "India", "Nepal",
"Germany")   cv = StringVar()   drplist=
OptionMenu(base, cv, *list_of_cntry)
drplist.config(width=15)   cv.set("United States")
lb2= Label(base, text="Select Country",
width=13,font=("arial",12))
lb2.place(x=14,y=280)
drplist.place(x=200, y=275)
```

```
lb6= Label(base, text="Enter Password", width=13,font=("arial",12))  lb6.place(x=19, y=320)
en6= Entry(base, show='*')
en6.place(x=200, y=320)

lb7= Label(base, text="Re-Enter Password", width=15,
font=("ar
ial",12))
lb7.place(
x=21,
y=360)
en7
=Entry(b
ase,
show='*')
en7.place(x=200, y=360)

Button(base, text="Register", width=10).place(x=200,y=400)  base.mainloop()

def generateOTP() :

   # Declare
a digits
variable     #
which stores
all digits
```

```
digits =
"012345678
9"
   OTP = ""

  # length of
password can be
changed    # by
changing value in
range     for i in
range(4) :
       OTP += digits[math.floor(random.random() * 10)]

   return OTP

# Driver
code if
__name__
==
"__main_
_" :

   print("OTP of 4 digits:", generateOTP())

digits="0123
456789"
OTP=""
for i in range(6):

OTP+=digits[math.floor(rando
m.random()*10)] otp = OTP + "
```

```
is your OTP" msg= otp s =
smtplib.SMTP('smtp.gmail.com
', 587)
s.starttls()
s.login("Your Gmail Account", "You app
password") emailid = input("Enter your
email: ")
s.sendmail('&&&&&&&&&&
&',emailid,msg) a =
input("Enter Your OTP >>: ")
if a == OTP:

print(
"Verifi
ed")
else:
   print("Please Check your
OTP again") root = Tk()
root.title("Python: Simple
Login Application") width =
400 height = 280
screen_width =
root.winfo_screenwidth()
screen_height =
root.winfo_screenheight() x
= (screen_width/2) -
(width/2)
y = (screen_height/2) - (height/2)
root.geometry("%dx%d+%d+%d" % (width,
height, x, y)) root.resizable(0, 0)
USERNAME = StringVar()
```

```
PASSWORD = StringVar()
Top = Frame(root, bd=2,  relief=RIDGE)
Top.pack(side=TOP, fill=X)
Form = Frame(root, height=200)
Form.pack(side=TOP, pady=20)
lbl_title = Label(Top, text = "Python: Simple
Login Application", font=('arial', 15))
lbl_title.pack(fill=X) lbl_username =
Label(Form, text = "Username:", font=('arial',
14), bd=15)
lbl_username.grid(row=0, sticky="e")
lbl_password = Label(Form, text =
"Password:", font=('arial', 14), bd=15)
lbl_password.grid(row=1, sticky="e") lbl_text =
Label(Form)
lbl_text.grid(row=2, columnspan=2)
username = Entry(Form,
textvariable=USERNAME, font=(14))
username.grid(row=0, column=1) password =
Entry(Form, textvariable=PASSWORD,
show="*", font=(14)) password.grid(row=1,
column=1) def Database():
    global conn, cursor     conn =
sqlite3.connect("pythontut.db")     cursor =
conn.cursor()     cursor.execute("CREATE
TABLE IF NOT EXISTS `member` (mem_id
INTEGER NOT NULL PRIMARY KEY
AUTOINCREMENT, username TEXT, password
TEXT)")          cursor.execute("SELECT * FROM
`member` WHERE `username` =
'admin' AND
`password` =
```

```
'admin'")     if
cursor.fetchone() is
None:
        cursor.execute("INSERT INTO `member`
(username, password)
VALUES('admin', 'admin')")
conn.commit() def Login(event=None):
Database()     if USERNAME.get() ==
"" or PASSWORD.get() == "":
        lbl_text.config(text="Please complete the required
field!", fg="red")     else:
        cursor.execute("SELECT * FROM `member`
WHERE `username`
= ? AND `password` = ?", (USERNAME.get(),
PASSWORD.get()))        if cursor.fetchone() is not
None:
            HomeWindow()
            USERNAME.set("")
PASSWORD.set("")
lbl_text.config(text="")
else:
            lbl_text.config(text="Invalid username or
password", fg="red")
            USERNAME.set("")
PASSWORD.set("")

cursor.clo
se()
conn.close
()
btn_login = Button(Form, text="Login", width=45,
```

```
command=Login) btn_login.grid(pady=25, row=3,
columnspan=2) btn_login.bind('<Return>', Login)


def
HomeWi
ndow():
global
Home
root.with
draw()
   Home = Toplevel()
   Home.title("Python: Simple
Login Application")    width =
600    height = 500
screen_width =
root.winfo_screenwidth()
screen_height =
root.winfo_screenheight()    x =
(screen_width/2) - (width/2)    y
= (screen_height/2) - (height/2)
   root.resizable(0, 0)
   Home.geometry("%dx%d+%d+%d" % (width,
height, x, y))    lbl_home = Label(Home,
text="Successfully Login!", font=('times new roman',
20)).pack()    btn_back = Button(Home, text='Back',
command=Back).pack(pady=20, fill=X)


def Back():

root.deiconify()
```

```
def
getdata(url):
r =
requests.get(ur
l)     return
r.text


                # input by geek
                from_Station_code = "GAYA"
                from_Station_name = "GAYA"

                To_station_code = "PNBE"
                To_station_name = "PATNA"
                # url
                url = "https://www.railyatri.in/booking/trains-between-
                stations?from_code="+from_Station_code+"&from_na
                me="+from_Stat
                ion_name+"+JN+&journey_date=+Wed&src=tbs&to_
                code=" + \
                   To_station_code+"&to_name="+To_station_name +
                \
                   "+JN+&user_id=-
                1603228437&user_token=355740&utm_source=dwebse
                arch_tbs_search_ trains"

                # pass the url
                # into getdata function
                htmldata = getdata(url)
                soup =
```

```
BeautifulSoup(htmldata,
'html.parser')

# find the Html tag
# with find()
# and convert into string
data_str = "" for item in soup.find_all("div",
class_="col-xs-12 TrainSearchSection"):
   data_str = data_str +
item.get_text() result =
data_str.split("\n")

print("Train between "+from_Station_name+" and
"+To_station_name) print("")



print("\n\nTicket Booking System\n")
restart = ('Y')
 while restart !=
('N','NO','n','n):
      print("1.Check PNR status")
print("2.Ticket Reservation")
       option = int(input("\nEnter your option : "))

                        if option == 1:
                        print("Your PNR status is t3")
                         exit(0)

      elif option == 2:        people = int(input("\nEnter
```

```
no. of Ticket you want :
"))
        name_l = []
age_l = []
sex_l = []
for p in
range(people):
name =
str(input("\nNa
me : "))
name_l.append(n
ame)
                age  = int(input("\nAge  : "))
age_l.append(age)
                                    sex  = str(input("\nMale or Female
: "))
                                       sex_l.append(sex)

    restart = str(input("\nDid you forgot someone? y/n:
"))    if restart in
('y','YES','yes','Yes'):
                restart =
('Y')         else :
x = 0
                print("\nTotal Ticket : ",people)
for p in range(1,people+1):
print("Ticket : ",p)
print("Name : ", name_l[x])
print("Age  : ", age_l[x])
                        print("Sex : ",sex_l[x])
x += 1
```

### 7.2. FEATURE 2

```
class User(AbstractBaseUser):
    """
    User model.
    """

    USERNAME_FIELD = "email"

    REQUIRED_FIELDS = ["first_name",
"last_name"]

    email = models.EmailField(
verbose_name="E-mail",
        unique=True
    )

    first_name = models.CharField(
```

```
verbose_name="First name",
        max_length=30
    )

    last_name = models.CharField(
verbose_name="Last name",
        max_length=40
    )

    city =
models.CharField(
verbose_name="City",
max_length=40
    )

    stripe_id = models.CharField(
verbose_name="Stripe ID",

unique=True,
max_length=50,
blank=True,
        null=True
    )

    objects = UserManager()


@prope
rty
def
```

```
            get_full
            _name(
            self):
                return f"{self.first_name} {self.last_name}"

            class Meta:
                verbose_name = "User"
                verbose_name_plural = "Users"


            class Profile(models.Model):
                """
                User's profile.
                """

    phone_number = models.CharField(        verbose_name="Phone
number"):
```