

# **SMART SOLUTIONS FOR RAILWAYS**

## **A PROJECT REPORT**

*Submitted by*

<b>1. KIRTHIKA. J</b>	<b>310619106061</b>
<b>2. KAVIYA. P</b>	<b>310619106060</b>
<b>3. MOBICA. R</b>	<b>310619106077</b>
<b>4. PREETHA. A</b>	<b>310619106103</b>
<b>5. RATNA TEJASWINI PUDOTA</b>	<b>310619106111</b>

**TEAM ID: PNT2022TMID54359**

*In partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF ENGINEERING**

*In*

**ELECTRONICS AND COMMUNICATION ENGINEERING**



**EASWARI ENGINEERING COLLEGE, RAMAPURAM,  
CHENNAI – 600089**



**ANNA UNIVERSITY, CHENNAI - 600025**

## TABLE OF CONTENTS:

S.NO	TITLE
<b>1</b>	<b>INTRODUCTION</b>
1.1	Project Overview
1.2	Purpose
<b>2</b>	<b>LITERATURE SURVEY</b>
2.1	Existing problem
2.2	References
2.3	Problem Statement Definition
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>
3.1	Empathy Map Canvas
3.2	Ideation & Brainstorming
3.3	Proposed Solution
3.4	Problem Solution Fit
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>
4.1	Functional requirements
4.2	Non-Functional requirements
<b>5</b>	<b>PROJECT DESIGN</b>
5.1	Data Flow Diagrams
5.2	Solution & Technical Architecture
5.3	User Stories

<b>S.NO</b>	<b>TITLE</b>
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>
6.1	Sprint Planning & Estimation
6.2	Sprint Delivery Schedule
6.3	Reports from JIRA
<b>7</b>	<b>CODING &amp; SOLUTIONING</b>
7.1	Python code
7.2	Database Schema
<b>8</b>	<b>TESTING</b>
8.1	Test Cases
8.2	User Acceptance Testing
<b>9</b>	<b>RESULTS</b>
9.1	Performance Metrics
<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>
<b>11</b>	<b>CONCLUSION</b>
<b>12</b>	<b>FUTURE SCOPE</b>
<b>13</b>	<b>APPENDIX</b>

# 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 :**

- S. Karthick and A. Velmurugan, "Android suburban railway ticketing with GPS as ticket checker," 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 63-66, 2012.
- B. Mallikarjuna, A. K. R. Doddi and G. Sailaja, "Enhanced Railway Reservation System using Internet of Things," 2018 IADS International Conference on Computing, Communications & Data Engineering (CCODE), 2018.
- G. Shelar, V. Rathod and S. Patil, "Railway Ticket Booking System with Restricted Wi-Fi Zone," International Journal of Trend in Scientific Research and Development (ijtsrd), vol. 2, no. 4, pp. 611-615, 2018.
- Swarup, M. Mohan, A. Dwivedi, C. Sonkar, R. Prasad, M. Bag and V. Singh, "A QR code based processing for dynamic and transparent seat allocation in Indian railway," International Journal of Computer Science Issues (IJCSI) 9, no. 3 (2012), p. 338, 2012.
- R. I. Rajkumar, P. E. Sankaranarayanan and G. Sundari, "GPS and Ethernet based real time train tracking system," 2013 International Conference on Advanced Electronic Systems (ICAES), pp. 282-286, 2013.
- C. Ulianov, P. Hyde and R. Shaltout, "Railway Applications for Monitoring and Tracking Systems," Marinov, M. (eds) Sustainable Rail Transport. Lecture Notes in Mobility. Springer, Cham., 20

## 2.3 Problem Statement definition:

QUESTION	DESCRIPTION
Who does the problem affect?	Passengers, Voyagers.
What are the boundaries of the problem?	Railway ticket booking system that generates a unique QR code for each ticket. Tracking the live location of trains using the GPS.
What is the issue?	Passengers cannot choose the seats they prefer. They have to carry a physical ticket which may get lost. Passengers who got to go quick may not have enough time to wait for the train indefinitely. The live location of the train can help them make the decision to wait or take some other mode of transport.
When does the issue occur?	All the time.
Where is the issue occurring?	Existing Train booking systems, Train and in Railway Stations.
Why is it important that we fix the problem?	Railway is one of the most common modes of transport. Improving the user experience is very important. An efficient way to check the tickets is of top priority.

1. Arun is a trekker and likes long distance travelling. His preferred mode of transport is Railways as it is cost effective. He wants to enjoy his journey along the way. So, he wants the window seat to adore the nature while travelling.
2. Bharath is a very busy guy and goes from one place to another frequently. He needs a way to minimize the time he waits between the travels so that he finishes his work quickly.
3. KG is a forgetful guy and the only thing he carries all the time is his iPhone. As a result, he likes to have the train ticket in the digital form in his mobile so that he has one less thing to worry about forgetting.
4. HP is a TTE and he wants to verify if the tickets are legitimate reliably and quickly so that he can check a greater number of passengers, reducing the possibilities of travelling without the ticket and escaping the fine.

## 3. IDEATION & PROPOSED SOLUTION

### 3.1 EMPATHY MAP CANVAS

[Click this link to view EMPATHY MAP](#)



#### Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Team ID: PNT2022TMID37863

#### Says

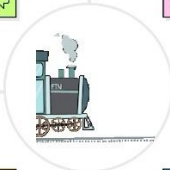
What have we heard them say?  
What can we imagine them saying?

Will I get a Smooth Journey??

About the cleanliness of seats and Toilets

Praising the Train travel that it is budget friendly, cheap and best

Will this QR code Works Properly



#### Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

Is it trust worthy to give my details?

Will the payment process be an end-to-end encrypted one?

Is it possible to know the current status of the Train?

Will I reach the Destination on time?



#### Does

What behavior have we observed?  
What can we imagine them doing?

Think of Alternate solutions

Travel to different places

checking their Luggages

wait in queue for long time to book tickets

having conversations with other passengers for timepass



Does I booked my ticket correctly?

Is the QR code is enough?

Is the website is authorised

Will the QR code work properly

#### Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

### 3.2 Ideation & Brainstorming

[Click this link to view BRAINSTORM](#)

[illegible]

### Group ideas

Start by showing your ideas within the starting point or selected notes as you go. Once all ideas are added, then move to group ideas into related boxes as a visual aid. Budget: 10-15 min for 10-15 ideas, 15-20 min for 20-25 ideas, 25-30 min for 30-35 ideas.

20 minutes

### Prioritize

Now focus on what is the most important and what is the least important. Prioritize your ideas by the importance of the idea and the importance of the idea. Budget: 10-15 min for 10-15 ideas, 15-20 min for 20-25 ideas, 25-30 min for 30-35 ideas.

20 minutes

### After you collaborate

Now you have a list of ideas. You can use this list to create a list of ideas. Budget: 10-15 min for 10-15 ideas, 15-20 min for 20-25 ideas, 25-30 min for 30-35 ideas.

20 minutes



### 3.3 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	How might we reduce the workload of the user and the use of paper while booking tickets?  How to ease the work of verifying the Tickets which is usually done by TTE thereby reducing the paperwork?
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	<ul style="list-style-type: none"> <li>• User friendly interface (Web Page) to book the ticket.</li> <li>• Unique QR code for each user to verify their tickets.</li> <li>• GPS Module to track the location of Train and live status is updated in the Web app.</li> </ul>
4	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> <li>❖ This model reduces the time consuming ticket booking by automating the process.</li> <li>❖ TTE can easily verify the details of the passenger using the QR code generated while booking the ticket which may reduce the use of paper work.</li> <li>❖ 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.</li> </ul>
5	Business Model (Revenue Model)	<p>→ This solution gives a feasibility to reduce the conventional ticket booking mechanism at counters which is tedious and time consuming, also the verification of the tickets is simplified. Since it also gives an additional train tracking method it will be more welcomed by the users.</p> <p>→ This model helps businessmen and travellers to book the tickets easily and it provides flexibility so customers will prefer this model which may increase the revenue of this model.</p> <p>→ 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.</p>
6	Scalability of the Solution	★ We propose a solution which works on the SaaS (Software as a Service) cloud model wherein all

### 3.4 Problem Solution fit

Define CS, fit into CC	<div>1. CUSTOMER SEGMENT(S)</div> <div>CS</div> <div>Passengers, Voyagers who wish to travel to different locations.</div>	<div>6. CUSTOMER CONSTRAINTS</div> <div>CC</div> <div>Passengers cannot choose the seats they prefer. They have to carry a physical ticket which may get lost. Passengers who got to go quick may not have enough time to wait for the train indefinitely.</div>	<div>5. AVAILABLE SOLUTIONS</div> <div>AS</div> <div>Passengers can take multiple physical copies of a ticket to prevent losing them.  TTE will have to manually verify the <del>validity</del> of each <del>passenger</del>.</div>	Explore AS, differentiate
	<div>2. JOBS-TO-BE-DONE / PROBLEMS</div> <div>J&amp;P</div> <div>The live location of the train must be easily accessible by the users. Ticket verification must be streamlined. Unnecessary documents should not be carried by passengers.</div>	<div>9. PROBLEM ROOT CAUSE</div> <div>RC</div> <div>Train booking infrastructure is outdated. Popularity of train travel has exploded. Trains are rarely on schedule.</div>	<div>7. BEHAVIOUR</div> <div>BE</div> <div>Bring original documents on train rides.  Take multiple copies of train tickets.  Arrive at station early to ensure they don't miss the train.</div>	
Focus on J&P, tap into BE, understand RC	<div>3. TRIGGERS</div> <div>TR</div> <div>Holidays Neighbours going on vacation Week-ended travel</div>	<div>10. YOUR SOLUTION</div> <div>SL</div> <div>Using GPS module to provide users with the train's location and estimated time of arrival. A web UI will be used as a portal for users, which also generates unique QR codes on successful ticket booking.  QR codes can be used to streamline the ticket verification process.</div>	<div>8. CHANNELS of BEHAVIOUR</div> <div>CH</div> <div>8.1 ONLINE Ticket booking through IRCTC website  8.2 OFFLINE Arriving at station early to check train status  Verifying passenger's ID proof</div>	Focus on J&P, tap into BE, understand RC
	<div>4. EMOTIONS: BEFORE / AFTER</div> <div>EM</div> <div>Confident -&gt; Confused: No way to know about the validity of the ticket Excited -&gt; Impatient: Not sure when train will arrive Energetic -&gt; Tired: TTE ticket verification takes too long per person</div>			
Identify strong T & EM				

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

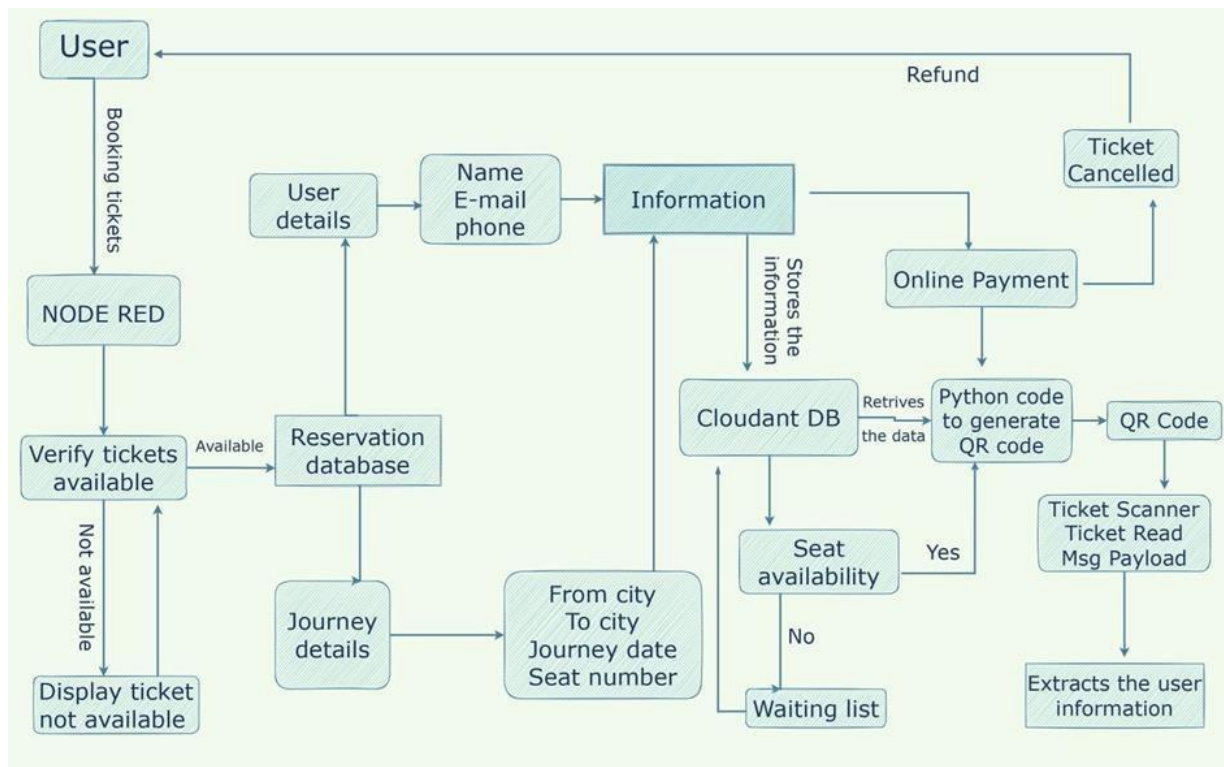
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Official website Registration through Form Registration through App
FR-2	User Confirmation	Confirmation via QR code Confirmation via message
FR-3	Ticket Verification	Ticket collector check the tickets using the QR code which is generated during ticket booking.
FR-4	TTE work load reduction	TTE can simply scan the QR code to identify the personal details and also to reduce the use of paper.
FR-5	Database Storage	All the booking details of the customer is stored in the database with unique ID which can retrieved back when the ticket collector scans the QR code.
FR-6	Train Tracking	The live status of the journey is updated in the web continuously

### 4.2 Non-Functional requirement

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	The user can easily book the tickets at their own pace based on the availability of seats without waiting in the ticket counter.
NFR-2	<b>Security</b>	User's personal details are safely stored in the Cloud. QR code reduces the duplicate train tickets.
NFR-3	<b>Reliability</b>	As IBM cloud is used to store the information about the customers this product is highly reliable.
NFR-4	<b>Performance</b>	The Web UI provides smooth user experience and improves the performance of this solution.
NFR-5	<b>Availability</b>	App is readily available on play store for mobile phone users or user can access the website using web browser.
NFR-6	<b>Scalability</b>	This solution can be easily scaled up based on the customer demand.

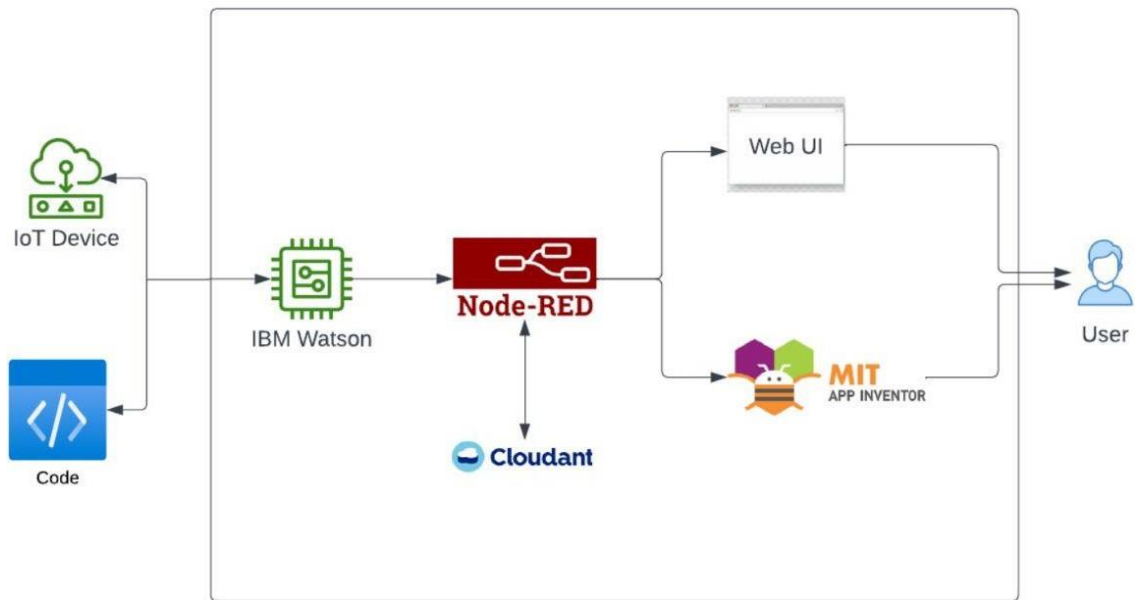
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagram

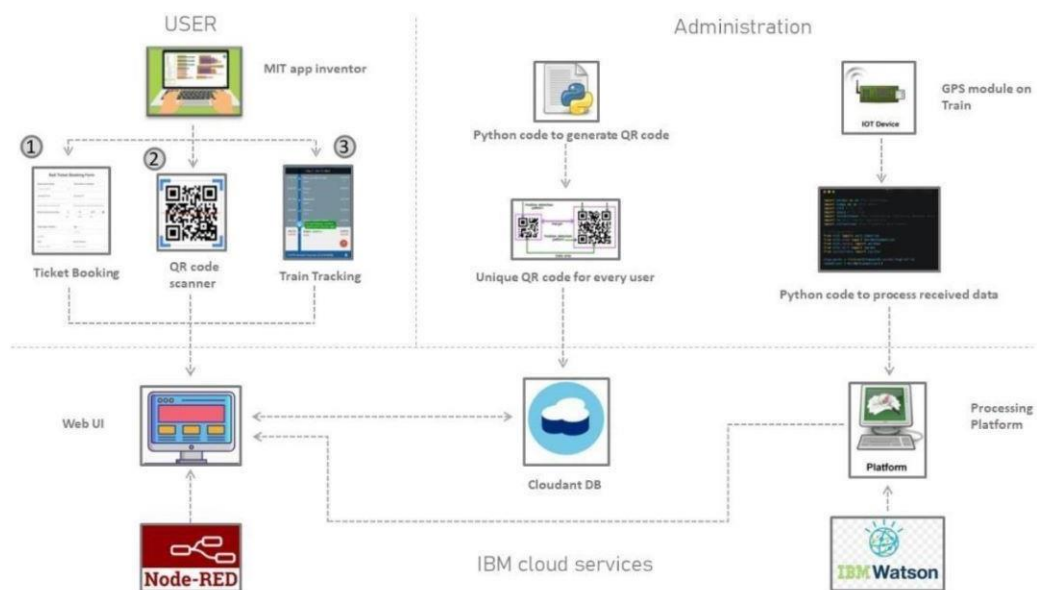


## 5.2 Solution and Technical Architecture

### Solution Architecture:



### Technical Architecture





## 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web user)	Registration	USN-1	As a user, I can register for the website 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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can access my account / dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can search the trains and seat availability.	I can see the booked tickets	High	Sprint-2
	Booking the Tickets	USN-7	As a user, I can book the train ticket according to my preference.	I can upload my details to fill the form for ticket booking	High	Sprint-2
	Tracking the Train	USN-8	As a user, I can track the location of Train.	I can access the arrival time to my boarding	High	Sprint-3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application by entering my mobile number and verifying it via OTP.	I can register & access the dashboard.	High	Sprint-1
		USN-5	As a user, I can register for the application through Gmail.	I can access my account / dashboard	Medium	Sprint-1
	Login	USN-6	As a user, I can log into the application by entering email & password.	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-7	As a user, I can search the trains and seat availability.	I can see the booked tickets	High	Sprint-2
	Booking the Tickets	USN-8	As a user, I can book the train ticket according to my preference.	I can upload my details to fill the form for ticket booking and get the QR code for it.	High	Sprint-2
	Tracking the Trains	USN-9	As a user, I can track the location of Train.	I can access the arrival time to my boarding	High	Sprint-3
Customer Care Executive	Customer service	USN-1	As a Customer Care Executive, I can help the customers by solving their queries via chat.	I can access the Help forum.	Low	Sprint-4
Train Ticket Examiner	Registration	USN-1	As a TTE, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
(TTE- Mobile user)						
		USN-2	As a TTE, I will receive a confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a TTE, I can register for the application through Facebook..	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a TTE, I can register for the application by entering my mobile number and verifying it via OTP.	I can register & access the dashboard.	High	Sprint-1
		USN-5	As a TTE, I can register for the application through Gmail.	I can access my account / dashboard	Medium	Sprint-1
	Login	USN-6	As a TTE, I can log into the application by entering email & password.	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-7	As a TTE, I can access my dashboard showing me verified tickets.	I can see the verified and non-verified tickets.	High	Sprint-2
	Ticket Verification	USN-8	As a TTE, I can verify the tickets by scanning the QR code shown by the passenger.	I can verify the Passenger's ticket.	High	Sprint-2

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
<b>Sprint-1</b>					
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
<b>Sprint-2</b>					
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 boarded and remaining number of passengers yet to be boarded	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
<b>Sprint-3</b>					
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 proper confirmed ticket.	6	High



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
<b>Sprint-4</b>					
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	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

## 6.3 Reports from JIRA:

This screenshot shows the Jira Backlog view for the project 'smart solution for railways'. The left sidebar contains navigation options: Roadmap, Backlog (selected), Board, Code, Project pages, Add shortcut, and Project settings. The main area displays two sprints: 'SSFR Sprint 1' (18 Nov – 29 Nov, 4 issues) and 'SSFR Sprint 2' (31 Oct – 5 Nov, 4 issues). Each sprint contains a list of issues with their descriptions and a 'TO DO' status. A right-hand sidebar shows the user's account information (ECE IV 6007 JAYESH M S) and options to manage the account, upgrade to the Standard plan, open the Quickstart, and access Jira settings.

Projects / smart solution for railways

### Backlog

SSFR Sprint 1 18 Nov – 29 Nov (4 issues)

As a developer I can provide a verification mechanism to check the integrity of QR code and the passenger details.

- SSFR-2 As a developer I can create the login page with email & password using Node-Red App
- SSFR-3 As a Developer, I can create a search box for the trains and seat availability.
- SSFR-4 As a Developer, I can provide a facility to book the train ticket according to user preference.
- SSFR-5 As a developer, I can write a code to view and download the QR code for the Booked Ticket

+ Create issue

SSFR Sprint 2 31 Oct – 5 Nov (4 issues)

- SSFR-6 As a Developer, I can create the login page for TTE with email and password using the Node-red app. TO DO
- SSFR-7 As a Developer, I can create a drop down box for the number of passengers on boarded and remaining number of passengers yet to be boarded. TO DO
- SSFR-8 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 TO DO
- SSFR-9 As a developer I can provide a verification mechanism to check the integrity of QR code and the passenger details. TO DO

+ Create issue

ACCOUNT: ECE IV 6007 JAYESH M S  
Manage account

UPGRADE: Try the Standard plan FREE 14-DAY TRIAL

Open Quickstart

JIRA: Profile, Personal settings, Log out

This screenshot shows the Jira 'All sprints' view for the project 'smart solution for railways'. The left sidebar is identical to the previous view. The main area displays a Kanban-style board with three columns: 'TO DO 13 ISSUES', 'IN PROGRESS 2 ISSUES', and 'DONE 1 ISSUE'. Issues are represented as cards with their descriptions and IDs. A right-hand sidebar shows the user's account information and options to manage the account, upgrade to the Standard plan, open the Quickstart, and access Jira settings.

Projects / smart solution for railways

### All sprints

TO DO 13 ISSUES | IN PROGRESS 2 ISSUES | DONE 1 ISSUE

Everything else 16 issues

- As a developer, I can write a code to view and download the QR code for the Booked Ticket (SSFR-5)
- As a Developer, I can create the login page for TTE with email and password using the Node-red app. (SSFR-6)
- As a Developer, I can create a drop down box for the number of passengers on boarded and remaining number of passengers yet to be boarded. (SSFR-7)
- As a Developer, I can create a search box for the trains and seat availability. (SSFR-3)
- As a Developer, I can provide a facility to book the train ticket according to user preference. (SSFR-4)
- As a developer I can create the login page with email & password using Node-Red App (SSFR-2)

ACCOUNT: ECE IV 6007 JAYESH M S  
Manage account

UPGRADE: Try the Standard plan FREE 14-DAY TRIAL

Open Quickstart

JIRA: Profile, Personal settings, Log out

smart solution for railways - Agil

https://itzsmartrailways.atlassian.net/jira/software/projects/SSFR/boards/1/backlog

Jira Software Your work Projects Filters Dashboards People Apps Create Search

smart solution for rail... Software project

PLANNING Roadmap Backlog Board

DEVELOPMENT Code

Project pages Add shortcut Project settings

You're in a team-managed project Learn more

Projects / smart solution for railways

### Backlog

SSFR Sprint 1 18 Nov – 29 Nov (4 issues) 0 0 0 Complete sprint

As a developer I can provide a verification mechanism to check the integrity of QR code and the passenger details.

- SSFR-2 As a developer I can create the login page with email & password using Node-Red App DONE
- SSFR-5 As a developer, I can write a code to view and download the QR code for the Booked Ticket DONE
- SSFR-4 As a Developer, I can provide a facility to book the train ticket according to user preference. DONE
- SSFR-3 As a Developer, I can create a search box for the trains and seat availability. DONE

+ Create issue

SSFR Sprint 2 31 Oct – 5 Nov (4 issues) 0 0 0 Complete sprint

- SSFR-6 As a Developer, I can create the login page for TTE with email and password using the Node-red app. DONE
- SSFR-7 As a Developer, I can create a drop down box for the number of passengers on boarded and remaining number of passengers yet to be boarded.. DONE
- SSFR-9 As a developer I can provide a verification mechanism to check the integrity of QR code and the passenger details. DONE
- SSFR-8 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 DONE

smart solution for railways - Agil

https://itzsmartrailways.atlassian.net/jira/software/projects/SSFR/boards/1/backlog

Jira Software Your work Projects Filters Dashboards People Apps Create Search

smart solution for rail... Software project

PLANNING Roadmap Backlog Board

DEVELOPMENT Code

Project pages Add shortcut Project settings

You're in a team-managed project Learn more

Projects / smart solution for railways

### Backlog

SSFR Sprint 3 7 Nov – 12 Nov (4 issues) 0 0 0 Complete sprint

- SSFR-13 As a developer, I will scan the QR code generated by python code to extract and verify the passenger details. DONE
- SSFR-12 As a developer, I will make sure that the on-boarded passenger is not involved in any travel fraudulently. DONE
- SSFR-14 As a developer, I will check whether the passenger is taking his/her journey with a proper confirmed ticket. DONE
- SSFR-10 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. DONE

+ Create issue

SSFR Sprint 4 14 Nov – 19 Nov (4 issues) 0 0 0 Complete sprint

- SSFR-14 As a developer, I can create web page to view train status using Node-red. IN PROGRESS
- SSFR-15 As a developer, I can extract details from IOT device using python code and IBM watson. IN PROGRESS
- SSFR-16 As a developer, I can upload the details to cloud and display it to use by connecting it with Node-red application. TO DO
- SSFR-17 As a passenger, I would like to receive updates over my train status during my journey via fast SMS or App notification. TO DO

+ Create issue

## 7 CODING & SOLUTIONING

### Tracking the Live location of train

```
import wiotp.sdk.device
import time
import random
myConfig =
{
    "identity": {
        "orgId": "cqfdow",
        "typeId": "NodeMCU", "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

def myCommandCallback (cmd):
    print ("Message received from IBM IoT Platform: %s" %
cmd.data['command'])
    m=cmd.data['command']
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig,  
logHandlers=None)
```

```
client.connect()
```

```
def pub (data):
```

```
    client.publishEvent(eventId="status", msgFormat="json", data=myData,  
qos=0, 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()
```

## QR Code Generation:

```
import cv2 as
cv import
numpy as np
import time
import pyzbar
from ibmcloudant import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import
BasicAuthenticator import wiotp.sdk.device

authenticator = BasicAuthenticator('apikey-v2-
xjj4vtu0o6zki1nv5xufzjrznfdbkbpap58pmn8qai9','c38318d6cb4dd64515f50d
5822fca43f')
service =
CloudantV1(authenticator=authen
ticator)
service.set_service_url('https://api
key-v2-
xjj4vtu0o6zki1nv5xufzjrznfdbkbpap58pmn8qai9:c38318d6cb4dd64515f5
0d5822fca43f@d30acc5e-636b- 4561-bfe7-5ffa4397a057-
bluemix.cloudantnosqldb.appdomain.cloud')
```

```
cap = cv.VideoCapture(0)
```

```
font =
```

```
cv.FONT_HERSHEY_PL
```

```
AIN if not
```

```
cap.isOpened():
```

```
    print("Cannot open  
    camera") exit()
```

```
myConfig = {
```

```
    "identity": {
```

```
        "orgId": "cqfdow",
```

```
        "typeId":
```

```
        "NodeMCU",
```

```
        "deviceId": "123
```

```
45
```



```
    },  
    "auth": {  
        "token": "12345678"  
    }  
}
```

```
def myCommandCallback(cmd):  
    print("Message received from IBM IoT Platform: %s" %  
          cmd.data['command']) m = cmd.data['command']
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig,  
logHandlers=None) client.connect()
```

```
def pub(data):  
    client.publishEvent(eventId="status", msgFormat="json", data=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 ...")
```

```
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_re  
    sult() 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'):
```

```
    client.commandCallback =
```

```
    myCommandCallback
```

```
    cap.release()
```

```
    cv.destroyAllWindows()
```

```
client.disconnect()
```

## QR Validation Code



**WEBSITE LINK:**

[http://127.0.0.1:1880/ui/#!/0?socketid=KCv\\_qH50nodXdeZIAAAD](http://127.0.0.1:1880/ui/#!/0?socketid=KCv_qH50nodXdeZIAAAD)

**GITHUB LINK :**

[IBM-EPBL/IBM-Project-41811-1660645178: Smart Solutions For Railways](https://github.com/IBM-EPBL/IBM-Project-41811-1660645178)  
[\(github.com\)](https://github.com)