Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	15 October 2022
Team ID	PNT2022TMID35329
Project Name	Project - Smart Solutions for Railways
Maximum Marks	4 Marks

Technical Architecture:

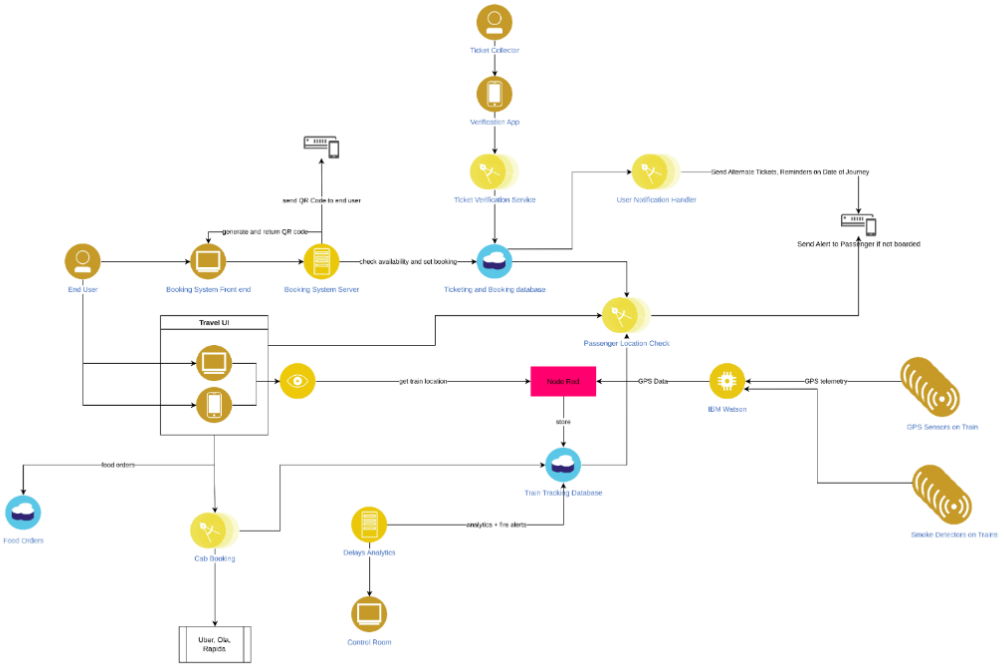

Figure 1: Architecture and data flow of the Smart Solutions for Railways Application

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface - Travel UI	Users interact with the application through web and mobile. This provides interfaces for tracking, food orders, alerts, and cab booking.	React JS and SCSS for the web ui. React Native for the mobile UI.
2.	User Interface - Booking System Front end	Web Interface for ticket booking.	React
3.	Application Logic - Booking System Server	Logic for booking processing.	Nodejs on IBM Serverless
4.	Application Logic - Ticket Verification Service	Logic for verifying tickets	Nodejs on IBM Serverless
5.	Application Logic - Verification Application	UI for scanning QR Code and verifying tickets	Mobile App built using React Native.
6.	Application Logic - User Notification Handler	Logic for sending alerts, notification and reminders	Nodejs on IBM Serverless
7.	Application Logic - Passenger Location Check	REST logic for reading and updating passenger location through mobile GPS.	IBM Serverless + GPS on mobile
8.	Application Logic - Train Tracking	Logic and infrastructure for tracking train location	IBM Watson + Node RED
9.	Application Logic - Cab Booking	Book Cabs based on train location and destination	IBM Serverless + External API
10.	Application Logic - Food Ordering	Collect food orders and send to pantry	IBM Serverless
11.	Database - Ticketing Storage	Tickets Database	MySQL Database with available tickets and bookings.
12.	Database - Train Tracking	Store train GPS telemetry data	IBM Cloudant - NoSQL Time Series
13.	Database - Smoke Alerts	Store smoke detector state telemetry data	IBM Cloudant - NoSQL Time Series
14.	Database - Food Orders	Store food order information and status	NoSQL
15.	External API - Cab Booking	Invoke APIs for ride providers for Cab Booking service	Nodejs + IBM Serverless

16.	Machine Learning Model	Perform analytics on train GPS data to find patterns in delay	Python, Scikit Learn
17.	Infrastructure (Local) - GPS Sensor	GPS Sensors on trains for tracking	ESP32 + GPS Sensor
18.	Infrastructure (Local) - Smoke Detector	Smoke detectors on trains for safety	ESP32 + Smoke Detector
19.	Infrastructure (Cloud) - REST Services	IBM Serverless deployments for REST handlers	IBM Serverless
20.	Infrastructure (Cloud) - Databases	Storage for time series data	IBM Cloudant
21.	Infrastructure (Cloud) - Iot	Fleet provisioning, authentication and telemetry	IBM Watson

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Node.js	Javascript, v8 engine
2.	Security Implementations	VPC for cloud databases, user authentication, ticket checker authentication	IAM Controls, OAuth for third party authentication, Json Web Tokens, Password salting
3.	Scalable Architecture	Serverless microservices - scalable through replication and parallelism. Databases, NoSQL for time series data - scalable through replication and partitioning. MySQL Database - scalable through partitioning	IBM Serverless, NoSQL + Database Partition
4.	Availability	Serverless ensures replication - hence availability. NoSQL and MySQL instances can also be replicated hence available	Database replication, across multiple availability zones
5.	Performance	Use Caches for tracking services. Serverless ensures fast spin up times, partitioned databases execute parallel queries faster	Microservices, Partitioning, Redis caches

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d