Software Requirements Specification(SRS) Document

1. Introduction

1.1 Purpose

Traveling by train can be stressful, especially when passengers don't have real-time updates on train locations, delays, or platform changes. Many people, especially in remote areas, struggle to get accurate and timely information, leading to missed connections and last-minute confusion.

This project aims to solve these issues by creating a Railway Tracking & Arrival Time Prediction System that provides live updates on train locations, expected arrival times, and platform details. Using GPS technology and smart prediction algorithms, the system will ensure that passengers always have the latest information at their fingertips. This will help make train travel more smooth, reliable, and stress-free, allowing people to plan their journeys with confidence.

1.2 Scope

Railways Tracking and Arrival Time Prediction System" is an attempt to simulate the basic concepts of an online tracking system. The system enables to perform the following functions:

- Tracking the Train
- Timing Details of the Train
- Expected Arrival and Departure Timings of the Train
- Improved and Optimised Service
- Remote Deployment to Travellers
- Time Saving

Intended Audience:

- Passengers travelling in train (any age group)
- Customers who are in the Station for pickup
- Passengers who have a linked train to board

1.3 Definitions, Acronyms and Abbreviations

- RTATP System: Railway Tracking & Arrival Time Prediction System – A system designed to provide real-time train tracking and estimated arrival times.
- GPS: Global Positioning System – A satellite-based system used to track train locations accurately.
- ETA: Estimated Time of Arrival – The predicted time at which a train is expected to reach a specific station.
- Admin: The system administrator responsible for maintaining the RTATP system and ensuring data accuracy.
- Passenger: Any individual using the railway services, including those traveling or waiting for arrivals.
- Platform Information: Details regarding which platform a train will arrive at, helping passengers plan their movement.
- Predictive Algorithm: A technology used to analyze historical data and real-time inputs to estimate train arrival times.

1.4 References

- Indian Railways Guidelines – Official regulations and protocols for train tracking and passenger information systems.
- GPS Technology Standards – Specifications and best practices for real-time location tracking using satellite systems.
- Machine Learning in Transportation – Research papers and case studies on predictive algorithms for arrival time estimation.
- Data Privacy and Security Laws – Compliance requirements for handling passenger data securely.
- Existing Railway Tracking Systems – Analysis of current solutions and their limitations in providing real-time updates.

2. Overall Description

2.1 Product Perspective

The Railway Tracking & Arrival Time Prediction (RTATP) System is a real-time train tracking solution designed to enhance passenger convenience by providing accurate train locations,

estimated arrival times, and platform details. It is a stand-alone web and mobile-based application that integrates with existing railway infrastructure, including GPS tracking systems, railway databases, and scheduling platforms.

The system aims to address the challenges of train delays and lack of real-time updates, particularly in low-connectivity regions, ensuring seamless travel planning for passengers. By leveraging GPS technology and predictive algorithms, the RTATP system will improve efficiency, reduce uncertainty, and enhance the overall railway travel experience.

2.2 Product Functions

The major functions of this system that make it simple and user-friendly are:

1. LOGIN/REGISTER:

- Administrator
- Passenger/Customer

2. SEARCH

3. SELECTION

4. TRACKING

2.3 User Characteristics

This section ensures **passenger/customer comfort** while using the system.

- Educational Level
  - Users should be comfortable with **English or Hindi**.
- Technical Expertise
  - Users should be comfortable with **general-purpose applications** on **computers or mobile phones**.

2.4 Constraints

1. Real-Time Accuracy & Internet Dependency
    - The system must provide **accurate real-time tracking** with minimal delays but depends on a **stable internet connection**. Functionality may be limited in **low-connectivity** areas.

2. Scalability & Security
    - The system should handle **a large number of users and multiple trains** efficiently while ensuring **data privacy and security** by restricting modifications to **authorized personnel only**.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Performance Requirements:

- **User Satisfaction** – The system is such that it stands up to the user's expectations.
- **Response Time** – The response of all the operations is good. This has been made possible by careful programming.
- **Error Handling** – Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- **Safety and Robustness** – The system is able to avoid or tackle disastrous action. In other words, it should be foolproof. The system safeguards against undesired events, without human intervention.
- **Portable** – The software should not be architecture-specific. It should be easily transferable to other platforms if needed.
- **User Friendliness** – The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

3.1.2 Design Constraints:

There are a number of factors in the client's environment that may restrict the choices of a designer. Such factors include standards that must be followed, resource limits, operating environment, reliability and security requirements, and policies that may have an impact on the design of the system.

- **Standard Compliance** – This specifies the requirements for the standards the system must follow. The standards may include the report format and accounting properties.
- **Hardware Limitations** – The software may have to operate on some existing or predetermined hardware, thus imposing restrictions on the design. Hardware limitations can include the types of machines to be used, the operating system available on the system, languages supported, and limits on primary and secondary storage.
- **Reliability and Fault Tolerance** – Fault tolerance requirements can place a major constraint on how the system is to be designed. Fault tolerance requirements often make the system more complex and expensive. Requirements about system behavior in the face of certain kinds of faults are specified. Recovery requirements are often an integral part here, detailing what the system should do if some failure occurs to ensure certain properties. Reliability requirements are very important for critical applications.
- **Security** – Security requirements are particularly significant in defense systems and database systems. They place restrictions on the use of certain commands, control access to data, provide different kinds of access requirements for different people, require the use of passwords and cryptography techniques, and maintain a log of activities in the system.

3.2 Non-Functional Requirements:

3.2.1 Performance Requirements:

- **Response Time:** The system should respond to user actions (e.g., page loads, button clicks) within 2 seconds. Critical operations, such as processing transactions or retrieving data, should complete within 5 seconds.
- **Scalability:** The system should be able to handle a growing number of users and requests without performance degradation. It should support at least 1,000 concurrent users.

- **Throughput:** The system should process at least 100 transactions per minute during peak hours.
- **Availability:** The system should be available 99.9% of the time, ensuring minimal downtime.
- **Resource Utilization:** The system should efficiently use server resources, keeping CPU and memory usage under 80% under normal load conditions.
- **Capacity:** The system should be capable of storing up to 1 million records in the database without performance issues.

3.2.2 Security Requirements:

The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.

3.2.3 Usability Requirements:

- **Ease of Use:** The system should be easy to learn and use, even for non-technical users. It should follow intuitive navigation patterns and use familiar icons and buttons.
- **Consistency:** The user interface should maintain a consistent layout, color scheme, and design across all pages to provide a seamless experience.
- **Accessibility:** The system should be accessible to users with disabilities, complying with accessibility standards (e.g., WCAG 2.1).
- **Help and Documentation:** The system should provide on-screen help, tooltips, and comprehensive user documentation to assist users in navigating and using the application effectively.
- **Error Messages:** Error messages should be clear, concise, and provide guidance on how to resolve issues.
- **Feedback and Confirmation:** The system should provide immediate feedback for user actions (e.g., loading indicators, success messages) to confirm that actions have been processed.
- **Customization:** Users should be able to personalize their experience by adjusting settings such as themes, language preferences, and notification options.

3.2.4 Reliability Requirements:

The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also, the system will be functioning inside a container. Thus the overall stability of the system is ensured.

3.3 Interface Requirements:

3.3.1 User Interface:

- Layout and Design: The system should have a clean and intuitive user interface with a consistent layout across all pages. It should follow modern design principles for ease of navigation.
- Responsiveness: The UI should be responsive and adapt to different screen sizes, including desktops, tablets, and mobile devices.
- Navigation: The system should have a clear and consistent navigation menu, allowing users to easily move between different sections.
- Accessibility: The interface should support accessibility features, such as keyboard navigation and screen reader compatibility, ensuring usability for all users.
- Input Controls: The system should use standard input controls such as text fields, dropdowns, checkboxes, and radio buttons for user inputs.
- Visual Feedback: Actions like button clicks or form submissions should provide immediate feedback, such as loading indicators or success messages.
- Error Handling: The UI should display clear and helpful error messages with guidance on resolving issues.
- Branding and Aesthetics: The interface should align with the brand's color scheme, typography, and style guidelines.

3.3.2 Hardware Interface:

- Input Devices: The system should support standard input devices, including a keyboard, mouse, and touch input for touchscreen devices.
- Output Devices: The system should be compatible with various display resolutions and support both standard monitors and mobile screens.
- Network Requirements: A stable internet connection is required for accessing and interacting with the system.
- Compatibility: The system should work on standard hardware configurations, including desktops, laptops, tablets, and smartphones.

3.3.3 Software Interfaces:

- Operating Systems: The system should be compatible with major operating systems, including Windows, macOS, Android, and iOS.
- Web Browsers: The application should support modern web browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- Third-Party Integrations: The system should integrate with third-party services such as payment gateways, social media platforms, and email services through APIs.
- Database Management: The system should interact with a relational database (e.g., MySQL, PostgreSQL) for data storage and retrieval.
- Security and Authentication: The system should support secure authentication methods, including OAuth and JWT (JSON Web Tokens) for user sessions.
- APIs and Web Services: The system should provide RESTful APIs for interaction with client applications and third-party services.
- Development Frameworks: The front-end should be built using React.js or a similar modern JavaScript framework, while the back-end should use Node.js with Express for server-side logic

4. System Architecture

4.1 System Components

- **Frontend:** The user interface developed using HTML, CSS, JavaScript, and a frontend framework (e.g., React, Angular).
- **Backend:** Server-side logic developed using a programming language like Python, Java, or Node.js, connected to a database (e.g., MySQL, PostgreSQL).
- **Database:** Centralized database for storing course materials, student information, grades, and other data.
- **APIs:** REST APIs for communication between the frontend and backend.

5. Glossary:

This should define all technical terms and abbreviations used in the document:

- **NTES** - National Train Enquiry System
- **SRS** - Software Requirements Specifications
- **ART** - Accident Relief Trains
- **UML** - Unified Modelling Language
- **ETA** - Estimated Time of Arrival

6. REFERENCES

- www.google.com
- www.irctc.co.in
- www.nationaltrainenquirysystem.com
- www.indianrail.gov.in
- www.academia.edu
- www.railyatri.com