

Practical 4

Software Requirement Specification (SRS) Development for the Selected System

Aim: Perform a requirement analysis and develop a Software Requirement Specification (SRS) sheet for the selected system. The SRS should include the following sections:

1. **Functionality:** Describe what the software is supposed to do.
2. **External Interfaces:** Explain how the software interacts with people, the system's hardware, other hardware, and other software.
3. **Performance:** Outline the expected speed, availability, response time, recovery time, and other performance-related characteristics of the software functions.
4. **Attributes:** Define considerations related to portability, correctness, maintainability, security, and other relevant attributes.
5. **Design Constraints:** Specify any design constraints imposed on the implementation, such as required standards, implementation languages, database integrity policies.

4. Software Requirement Specification (SRS)

A Software Requirement Specification (SRS) is a structured document that defines the functionalities, constraints, and requirements of a software system. It serves as a foundation for development, testing, and stakeholder communication.

4.1 Overview

An E-Train Booking System is a digital solution designed to streamline and automate train ticket booking operations. It replaces traditional manual booking with an efficient, organized system that enables seamless management of train schedules, user data, booking, and cancellations. The E-Train Booking System aims to enhance the user experience by providing faster access to train schedules, improving accountability, and reducing human errors. This Software Requirement Specification (SRS) document follows the IEEE SRS standard and outlines the functional and non-functional requirements of the system, as well as external interfaces, constraints, and dependencies.

4.2 Purpose

The primary purpose of this E-Train Booking System is to provide an automated platform for managing train ticket bookings, cancellations, and user transactions. The system enables users to search for, book, and cancel train tickets efficiently while allowing administrators to monitor booking activities, track train schedules, and generate reports. It also supports notifications for booking confirmations, cancellations, and train schedule updates.

4.3 Document Conventions

This document follows structured formatting using numbered sections, bullet points, and industry-standard terminology for clarity and ease of reference. This document is intended mainly for users like Booking Administrators who manage user accounts, train schedules, and booking policies and others related to the project such as Developers who understand and implement system requirements, Testers who validate system functionality and performance, Stakeholders who review system capabilities and constraints etc. This document follows IEEE 830-1998: Recommended Practice for Software Requirements Specifications.

4.4 Overall Description

This section provides an overview of the system's purpose, functions, and constraints.

Product Perspective

The E-Train Booking System is a standalone software system designed to digitize train ticket booking operations. It integrates with existing railway infrastructure, replacing traditional booking methods with a centralized database.

Product Functions

The system will provide:

- **User Account Management:** Role-based access control for passengers, booking agents, and system administrators.
- **Train Management:** Add, update, delete, and search train schedules and routes.
- **Ticket Booking Management:** Book, cancel, and modify tickets with real-time availability tracking.
- **Notifications & Alerts:** Alerts via email/SMS for booking confirmations, cancellations, and schedule changes.
- **Report Generation:** Statistical insights on booking trends, revenue, and train occupancy.

Users of the E-Train Booking System include:

- **Passengers:** Registered users who search for trains, book, modify, and cancel tickets.
- **Booking Agents:** Authorized agents who assist passengers with ticket booking and

cancellations.

- **System Administrators:** IT personnel responsible for system maintenance, user management, and ensuring data security.

4.5 Specific Requirements

The Specific Requirements section of the SRS outlines detailed functional and non-functional needs of the E-Train Booking System. Functional requirements define what the system should do, describing specific actions and behaviors. These requirements directly contribute to the core functionalities of the software. Non-functional requirements define how the system performs its functions, focusing on system attributes like security, performance, usability, and scalability. The External Interfaces section of an SRS describes how the E-Train Booking System interacts with users, hardware, and other software. These interfaces ensure seamless operation and integration within the existing environment.

4.5.1 Functional Requirements

Functional requirements describe what the system should do:

1. User Authentication & Authorization: Secure login and role-based access.
2. Train Schedule Management: CRUD (Create, Read, Update, Delete) operations for train schedules.
3. Booking & Cancellation: Tracking ticket bookings, cancellations, and payments.
4. Payment Management: Automatic calculation and processing of ticket fares.
5. Search & Filter: Advanced search options for train schedules.
6. Reports & Analytics: Generate reports on system usage.
7. Notifications: Email/SMS reminders for booking confirmations and train schedule updates.

4.5.2 External Interfaces

External interfaces define how the system interacts with users, hardware, and other software components:

1. User Interface (UI): A web-based interface that provides an intuitive dashboard for users to interact with the system.
2. Hardware Interface: The system must support barcode scanners for ticket verification, printers for ticket receipts, and mobile devices for remote access.
3. Software Interfaces: The system integrates with SQL databases for storage.
4. Communication Interfaces: The system supports email, SMS, and push notifications to alert users about booking confirmations and train schedule updates.

4.5.3 Non-Functional Requirements

Non-functional requirements describe system performance, security, and usability expectations:

1. **Performance:** The system must handle maximum concurrent users, with response times under 2 seconds.
2. **Security:** Implements encryption, multi-level authentication, and role-based access control to prevent unauthorized access.
3. **Usability:** The UI should be user-friendly, accessible, and follow WCAG guidelines.
4. **Scalability:** The system must support expansion without performance degradation.
5. **Availability:** Ensures maximum uptime, with automated backups for data recovery.

4.5.4 Design Constraints

Design Constraints Design constraints are limitations that affect the implementation and architecture of the E-Train Booking System. These constraints ensure that the system is developed within predefined technical, security, and operational guidelines. Design constraints define system limitations:

1. **Database Management:** The system should support both SQL and NoSQL databases depending on scalability and performance needs.
2. **Technology Stack:** The system should be built using Node.js for the backend, React.js for the frontend, and MongoDB as the primary database.
3. **Security Compliance:** The system must encrypt user data, follow GDPR compliance, and implement multi-factor authentication (MFA).
4. **Performance Optimization:** The system should implement load balancing and caching mechanisms for high availability.

4.5.5 System Attributes

1. **Portability:** The system must be accessible on Windows, macOS, Linux, and mobile devices.
2. **Maintainability:** Modular design to allow easy updates and fixes.
3. **Reliability:** Ensures consistent operation with failover mechanisms.
4. **Scalability:** The system must handle increasing users and booking records efficiently.
5. **Interoperability:** Ability to integrate with third-party systems for extended functionality.
6. **Extensibility:** The system should allow future enhancements and new feature additions with minimal modifications.

4.4.6 Implementation details and Hardware Requirements

- **Server:** 8-core CPU, 32GB RAM, 1TB SSD, stable high-speed internet.
- **Client Machines:** Dual-core CPU, 4GB RAM, 250GB storage, internet access.
- **Network:** Reliable broadband connection for real-time operations.
- **Barcode Scanners:** For e-ticket validation at railway stations.
- **Printers:** For printing tickets and transaction reports.
- **Mobile Devices:** Smartphones for mobile bookings and access.
- **Backend:** Node.js with Express.js for handling server-side logic.
- **Frontend:** React.js for an intuitive user interface and seamless experience.
- **Database:** MongoDB for efficient data storage and retrieval.
- **Security:** JWT for secure user sessions, SSL for data encryption.
- **APIs:** REST APIs for communication between components.
- **Deployment:** Cloud-based deployment using AWS for reliability.
- **Maintenance:** Regular system updates, security patches, and automated backups.