

Software Requirements Specification (SRS)

E-Ticket Booking System for Trains

Technology Stack: ASP.NET Web Forms/MVC (C#), .NET Web Services (REST), SQL Server

Student Name: Akash M [2023BCSE07AED134]

1. Introduction

The rapid advancement of information technology has transformed traditional service systems into efficient online platforms. One such area is railway ticket booking, which has shifted from manual counter-based systems to automated web-based solutions. An online train e-ticket booking system allows passengers to conveniently search trains, check seat availability, book tickets, and manage cancellations through an internet-enabled application. This improves user convenience, reduces manual effort, and ensures better data accuracy and transaction security.

The E-Ticket Booking System for Train is a web-based application developed using ASP.NET with C#, web services, and SQL Server. The system is designed to automate the train ticket booking process by providing functionalities such as user registration, train search, seat availability checking, ticket reservation, online

1.1 Purpose

This SRS specifies the complete functional and non-functional requirements for the E-Ticket Booking System for Trains (ETBST). It serves stakeholders, architects, developers, testers, and operations teams as the authoritative reference for scope, verification, and acceptance criteria.

1.2 Document Conventions

- IEEE-style numbering for sections and requirements (FR-#, NFR-#).
- Priority tags: [MUST], [SHOULD], [MAY].
- All dates in ISO-8601 (YYYY-MM-DD).

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audience:

- Project Stakeholders and Supervisors: To understand the scope, objectives, and functionality of the system.
- Developers: To use the requirements as a baseline for system design and implementation.

- Testers: To derive test cases and perform verification based on defined requirements.
- Evaluators or Reviewers: To assess the completeness, correctness, and quality of the system requirements.

1.4 Project Scope

ETBST is a web-based platform enabling passengers to search trains, check availability and fares, book seats, pay online, receive e-tickets, and manage bookings (cancellations, refunds). The system provides admin tools for schedule, fare, and quota management, and exposes secure web services for integration with external partners and mobile apps.

1.5 References

[1] IEEE Std 830-1998 — Software Requirements Specifications

[2] IEEE Std 1016-2009 — Software Design Descriptions

[3] Organization Security & Compliance Policies

2. Overall Description

2.1 Product Perspective

ETBST is a multi-tier web application with ASP.NET presentation, C# business services, and SQL Server data storage. It integrates identity/SSO, payment gateways, notification services (email/SMS), and external rail operations systems for schedules and PNR lifecycle.

2.2 Product Functions

The E-Ticket Booking System for Train provides the following basic and essential functionalities required for online train ticket reservation:

- **User Registration and Authentication**
The system allows users to register, log in, and reset their passwords to securely access the application.
- **Train Search**
Users can search for available trains by selecting source station, destination station, journey date, and travel class.
- **Seat Availability and Fare Enquiry**
The system enables users to check seat availability and view ticket fare details for the selected train and class.

- **Ticket Booking and PNR Generation**
The system allows users to book train tickets, automatically allocate seats, and generate a unique Passenger Name Record (PNR) for each confirmed booking.
- **Online Payment Processing**
Users can complete ticket booking by making online payments using available payment options supported by the system.
- **E-Ticket Generation**
After successful booking, the system generates an electronic ticket that can be viewed or downloaded by the user.
- **Ticket Cancellation**
The system allows users to cancel booked tickets according to defined cancellation rules.
- **Administrative Management**
The system provides an administrator module to manage train details, routes, schedules, and booking information.

2.3 User Classes and Characteristics

- **Passenger**
 - Primary user of the system.
 - Can register, log in, search trains, book tickets, make payments, view e-tickets, and cancel bookings.
 - Requires basic knowledge of computer and internet usage.
 - Has limited access to personal booking information only.
- **Administrator**
 - Authorized user responsible for managing system data.
 - Manages train details, routes, schedules, and booking records.
 - Has full access to administrative features.
 - Ensures smooth operation of the system.

2.4 Operating Environment

- Web: Latest Chrome, Edge, Firefox, Safari.
- Server: Windows Server with IIS, .NET 8 (or LTS), ASP.NET Core/ASP.NET MVC.
- Database: Microsoft SQL Server 2019 or later.
- APIs: REST over HTTPS; JSON payloads.

2.5 Design and Implementation Constraints

- The system shall be developed using ASP.NET with C#, Web Services, and Microsoft SQL Server.
- The application shall be implemented as a web-based system accessible through standard web browsers.

- All system data shall be stored and managed using a SQL Server database with proper data integrity constraints.
- User authentication and access control shall be implemented to ensure system security.
- Online payment processing shall follow secure transaction practices, and sensitive payment details shall not be stored.
- The system shall support multiple concurrent users with acceptable performance.
- The user interface shall be simple and user-friendly for easy navigation.
- The system requires a stable internet connection for operation.
- The application should be designed to allow easy maintenance and future enhancements.

2.6 User Documentation

- Passenger user guide and FAQ.
- Admin/Operations manual.
- API documentation (OpenAPI/Swagger).

2.7 Assumptions and Dependencies

The development and operation of the E-Ticket Booking System for Train are based on the following assumptions and dependencies:

Assumptions

- Users have basic knowledge of using web applications and internet browsers.
- Users have access to a stable internet connection while using the system.
- The system will be accessed through commonly used web browsers.
- Administrator users are trained to manage train and booking data correctly.
- Data entered by users and administrators is assumed to be valid and accurate.

Dependencies

- The system depends on the availability of the web server hosting the application.
- Online payment functionality depends on the availability of external payment services.
- Email or notification delivery depends on configured communication services.
- System performance and reliability depend on proper database availability and maintenance.
- Future enhancements depend on available system resources and technical support.

3. System Features (Functional Requirements)

FR-9 Booking Management [MUST]

- Users shall view booking history and ticket details.
- Users shall cancel tickets as per policy with automated refund computation and processing.

- Rescheduling/Change of Journey (COJ) shall be supported (SHOULD) subject to availability and fees.

FR-1 User Registration and Authentication [MUST]

- The system shall allow users to register by providing basic personal details.
- The system shall allow registered users to log in using valid credentials.
- The system shall provide a password reset facility for registered users.
- The system shall ensure secure access to user accounts.

FR-2 Train Search [MUST]

- The system allow users to search for trains by selecting source station and destination station.
- The system shall allow users to select the journey date.
- The system shall display a list of available trains based on the search criteria.
- The system shall display train details such as train number and train name.

FR-3 Seat Availability and Fare Enquiry [MUST]

- The system shall allow users to check seat availability for the selected train and class.
- The system shall display fare details for the selected journey.
- The system shall show availability and fare information before ticket booking.

FR-4 Ticket Booking and PNR Generation [MUST]

- The system shall allow users to enter passenger details for booking.
- The system shall allocate seats automatically during booking.
- The system shall generate a unique Passenger Name Record (PNR) for each confirmed booking.
- The system shall store booking details in the database.

FR-5 Online Payment Processing [MUST]

- The system shall provide online payment options to complete ticket booking.
- The system shall confirm booking only after successful payment.
- The system shall notify the user of payment success or failure.

FR-6 E-Ticket Generation [MUST]

- The system shall generate an electronic ticket after successful booking.
- The system allows users to view or download the e-ticket.
- The ticket shall contain journey and passenger details along with the PNR.

FR-7 Ticket Cancellation [MUST]

- The system allows users to cancel confirmed tickets.
- The system shall update booking status after cancellation.
- The system shall display cancellation confirmation to the user.

FR-8 Administrative Management [MUST]

- The system shall provide an administrator login.
- The administrator shall be able to manage train details, routes, and schedules.
- The administrator shall be able to view booking records.
- The administrator shall maintain overall system data.

FR-10 Check-In & Verification [Optional]

- Support digital verification by TTE using QR/PNR lookup (MAY).
- Offline verification fallback for limited connectivity zones (MAY).

FR-11 Notifications [MUST]

- System shall send transactional notifications for registration, OTP, payment, booking, cancellation, refund.
- Users shall manage notification preferences where applicable.

FR-12 Reporting & Reconciliation [SHOULD]

- Finance users shall download daily settlement and refund reports (CSV/Excel).
- Admins shall access operational dashboards (bookings, load factor, revenue).

FR-13 Audit & Compliance [MUST]

- All critical operations shall be logged with user, timestamp, IP, and context.
- System shall retain audit logs per policy and expose them for compliance reviews.

FR-14 Administration [MUST]

- CRUD for stations, routes, trains, classes, coaches, quotas, timetable, fares, blackout dates.
- Configuration of payment providers, taxation, service fees, refund rules.
- User, role, and permission management.

FR-15 Public APIs [SHOULD]

- Provide secure REST endpoints for mobile/partner channels for search, availability, booking, and ticket retrieval.
- Rate limiting and API keys/OAuth2 required.

4. External Interface Requirements

4.1 User Interfaces

- Responsive web UI (ASP.NET MVC/Razor Pages) with WCAG 2.1 AA compliance.
- Forms with client/server-side validation, progress indicators, error messaging.
- Localization/i18n for content and date/time formats.

4.2 Hardware Interfaces

- Client: standard desktops/laptops/mobile devices.
- Server: Windows Server with IIS;

4.3 Software Interfaces

- External Rail OPS: schedule/inventory/PNR APIs (REST/Message queue).
- Database: Microsoft SQL Server; stored procedures for critical transactions.

4.4 Communication Interfaces

- HTTPS/TLS 1.2+ for all endpoints.
- Webhooks for payment status and refund updates.
- Retry with idempotency keys for at-least-once delivery.

5. Non-Functional Requirements

NFR-1 Performance & Capacity

- Search results shall render in ≤ 2 seconds for p95 under normal load.
- Booking confirmation workflow (payment callback to PNR issued) shall complete in ≤ 5 seconds p95, excluding gateway latency.
- Initial capacity target: 200 concurrent bookings/min and 2,000 searches/min; scalable to 5 \times with horizontal scaling.

NFR-2 Security

- TLS 1.2+ in transit; Transparent Data Encryption (TDE) at rest for SQL Server.
- MFA for privileged roles; least-privilege service accounts; secrets in vault.
- OWASP ASVS L2 controls; periodic penetration testing; secure coding practices.
- PCI DSS scope isolation for payment flows; tokenization of PAN; no card data storage.

NFR-3 Reliability & Availability

- 99.9% monthly uptime target with multi-instance deployment behind load balancer.
- Graceful degradation when external providers are unavailable (read-through caches, circuit breakers).
- Automated backups (RPO ≤ 15 min, RTO ≤ 60 min).

NFR-4 Usability & Accessibility

- WCAG 2.1 AA; keyboard navigation; screen reader semantics.
- Consistent information architecture and visual design system.

NFR-5 Observability & Maintainability

- Centralized logging with correlation IDs for each request/PNR.
- APM metrics (latency, error rate, throughput), dashboards, alerting (SLOs).
- 12-factor configuration; infrastructure as code; CI/CD with automated tests.

NFR-6 Scalability

- Stateless web tier enabling horizontal scaling.
- Database scaling via read replicas and partitioning where needed.

NFR-7 Compliance & Privacy

- Data minimization; consent management; retention and deletion policies.
- Audit trails immutable and time-synchronized.

6. System Architecture Overview

Layered architecture:

- Presentation Layer: ASP.NET MVC/Razor Pages with client-side enhancements.
- Business Layer: C# services implementing domain logic, validation, pricing, allocation.
- Integration Layer: REST clients/webhooks for payments, messaging, rail OPS; retry/idempotency.
- Data Layer: SQL Server schema with stored procedures for booking/PNR transactions.

7. System and Data Models (Overview)

Core entities (indicative): Station, Route, Train, ServiceDay, Coach, Class, Quota, Seat, Schedule, FareRule, User, Role, Passenger, Cart, Booking, Payment, Ticket, Refund, Notification, AuditEvent.

8. Validation and Acceptance Criteria

- All functional requirements FR-1..FR-15 have corresponding test cases passed in SIT and UAT.
- Performance tests meet targets in NFR-1 with test reports attached.
- Security assessment with no Critical/High findings open; PCI attestation for payment flows.
- Business sign-off on sample tickets, refund scenarios, and reconciliation reports.

9. Appendices

Appendix A — Glossary

- PNR: Passenger Name Record (booking identifier).
- RAC: Reservation Against Cancellation.
- WL: Waitlist status.
- MFA: Multi-Factor Authentication.

Appendix B — Sample APIs

- GET /api/trains/search?from=...&to=...&date=...&class=..."a=...
- GET /api/trains/{trainId}/availability?date=...&class=..."a=...
- POST /api/bookings (cart->PNR)
- POST /api/payments/{pnr}/capture
- GET /api/tickets/{pnr}