Railway Management System (RMS) Requirements Specification

Version 1.0

March 8th, 2025

**

**Table of Contents**

[**1. Executive Summary 2**](#_2et92p0)

**1.1 Project Overview 3**

**2. Product/Service Description 3**

**2.1 Product Context 3**

**2.2 User Characteristics 3**

**2.3 Assumptions 3**

**2.4 Constraints Dependencies 3**

**3. Requirements 3**

**3.1 Functional Requirements 4**

[3.2 Non-Functional Requirements 5](#_35nkun2)

**3.2.1 Product Requirements 5**

[3.2.1.1 Usability Requirements 5](#_q8yo01f084zc)

[3.2.1.2 Performance Requirements 5](#_ddqqf5pzexcw)

[3.2.1.3 Availability 6](#_vshy0pmxh14t)

[3.2.1.4 Security 6](#_8i9hvv5zz8n4)

[3.2.2 Organizational Requirements 6](#_sas9qa6j3xo9)

[3.2.3 External Requirements 6](#_3as4poj)

**4. User Scenarios/Use Cases 6**

[**5. Diagrams 7**](#_41mghml)

[**6. Design Patterns 7**](#_xr1wgl8t7616)

[**7. Appendix. 7**](#_uucczfw9bw0v)

[**Organizing the Requirements 7**](#_5l1q87u4mqs7)

# Executive Summary

## Project Overview

This project seeks to model and design a management system that is capable of automating and streamlining the operations of a given railway network, ensuring quicker operations, precise commands, higher security, and general improvements in user experience.

# ***Product/Service Description***

The Railway Management System is designed to modernize and optimize the operations of railway networks by addressing the core challenges of managing trains, passengers, schedules, and maintenance, ultimately improving efficiency, safety, and customer satisfaction across the entire transportation ecosystem.

## Product Context

RMS does not have to necessarily be a standalone product. During development, we keep in mind functionalities that the system may gain from the use of other parent or child systems. For example, the system may need to serve as a smaller cog in a local public transportation management system in regards to train traffic control (parent system), or it might interact with systems such as embedded systems in the trains and stations to communicate availability (child system). The system should also need to be aware of other neighbouring railway systems that fall outside of its jurisdiction.

## User Characteristics

* Station Managers
* Train and Station Personnel
* Passengers
* Traffic Manager
* Railway and Train Maintenance (External Maintenance)
* Admin

## Assumptions

An assumption is made that the system "owns" and directly controls a fleet of trains. This simplifies the design by reducing the complexity involved in interacting with other external systems or organizations. Under this assumption, the system has full access to operational data, maintenance schedules, and scheduling control for the entire fleet.

## Constraints and Dependencies

In the design and development of the Railway Management System (RMS), we must consider several **constraints** and **dependencies** to ensure that our system operates effectively and meets the required standards. Some examples are:

**1.Access, management and security**

* **Access Control**: The system must implement strict access control mechanisms to ensure that only authorized users (e.g., administrators, maintenance personnel, and passengers) can access sensitive information. Role-based access control will be used to enforce security policies.
* **Data Security**: The system must adhere to industry-standard encryption protocols (e.g., AES, SSL/TLS) to protect passenger data, payment information, and operational data both in transit and at rest.

**2.Non-stop operation**

* **Availability:** The system must be highly available and capable of operating 24/7, ensuring no downtime during peak travel times or maintenance windows. High availability solutions like load balancing, failover systems, and backup servers will be implemented.

#### 3. Record Keeping

* **Data Retention**: The system must maintain historical records of train schedules, passenger transactions, maintenance logs, and other operational data for a specified retention period, as mandated by regulatory requirements and internal policies.

#### 4. Scalability

* **User Load**: The system must be scalable to handle increased user traffic, especially during peak times (e.g., holiday travel, rush hours). This includes passenger reservations, ticket sales, and administrative access.
* **Data Volume**: As the railway network expands, the system must scale to accommodate increasing amounts of operational data, such as train schedules, maintenance records, and passenger transactions.

#### 5. Budget and Resource Constraints

* **Budget Limitations**: The design, development, and deployment of the system must adhere to budget constraints, which may limit the choice of technologies, features, and scale of implementation.

#### 6. Time Sensitivity and Real-Time Data

* **Real-Time Updates**: The system must provide real-time updates on train schedules, delays, cancellations, and seat availability to passengers, as well as to the administrative team. This requires low-latency data processing and communication protocols.
* **Synchronization**: Time synchronization across multiple components (trains, stations, servers) is critical to ensure consistent and accurate data for scheduling and reporting purposes.

# **Requirements**

## Functional Requirements Naming Key

#### Actors

## S = System

## TM = Traffic Manager

## SM = Station Manager

## P = Passenger

## E = Employee

## M = Manager

## U = User (All Users)

## A = Admin

## Requirement Types

## MNG = Management (Tracks, Scheduling, etc.)

## INT = Interoperability (External Systems)

## TRK = Tracking (Stations, Railways, Trains, etc.)

## SCH = Scheduling & Timetables

## PER = Personnel & Passenger Management

## TKT = Ticketing (Booking, Cancellation, Loyalty)

## PAY = Payment Processing

## NOT = Notifications & Updates

## WRK = Employee Work & Tasks

## VIW = Information Access & Viewing

## ALR = Alerts & Emergency Responses

## SVC = Passenger Services (Accessibility, Special Requests)

## CAP = Capacity Management (Prevent Overbooking)

## AUTH = Authentication & Logging (Login/Logout)

## FIN = Financial Tracking & Reporting

## LOG = System Logging & Operations

## ADM = Admin-Specific Controls

| **Req#** | **Requirement** | **Comments** | **Priority** | **Date Rvwd** | **SME Reviewed / Approved** |
| --- | --- | --- | --- | --- | --- |
| **TM\_MNG\_01** | System must be able to keep track of constructed tracks under management | General geographic mapping | 1 |  |  |
| **S\_INT\_02** | System needs to be able to interact with other external railway systems outside of its jurisdiction | Interaction in this case involves keeping track of trains going from and to, closed railways and stations. | 1 |  |  |
| **S\_TRK\_03** | The system needs to be able to keep track of its own stations availability and condition | Conditions: onAlert, fullCapacity, underMaintenance.  More conditions TBD when other edge cases present themselves. | 1 |  |  |
| **S\_TRK\_04** | The system needs to be able to keep track of its railways availability and condition | Conditions: onAlert, underMaintenance, occupied.  More conditions TBD when other edge cases present themselves. | 1 |  |  |
| **S\_TRK\_05** | The system needs to be able to keep track of its train cars availability and condition | Conditions: onAlert, underMaintenance, inUse, outsideBounds  More conditions TBD when other edge cases present themselves | 1 |  |  |
| **S\_SCH\_06** | The system needs to keep track of a current schedule of trains and their specific itineraries. | Keep track of both time and locations. | 1 |  |  |
| **S\_PER\_07** | The system needs to keep track of human personnel availability and condition | Conditions: onLeave, onStandby, busy, onAlert  More conditions TBD when other edge cases present themselves | 1 |  |  |
| **S\_PER\_08** | The system needs to be able to keep track of passenger information. |  | 1 |  |  |
| **TM\_SCH\_09** | The Traffic Manager must be able to create, update, and delete schedules for trains under their jurisdiction. |  | 1 |  |  |
| **SM\_PER\_10** | The Station Manager must be able to allocate human personnel to complete tasks on stations under their jurisdiction. |  | 1 |  |  |
| **TM\_VIW\_11** | The Traffic Manager must be able to view train information, station information, and track information. |  | 1 |  |  |
| **S\_ERR\_12** | The system needs to prevent the Traffic Manager from causing errors | Main errors include:  Routing through stations and tracks under repair or under alert  Causing train collisions on tracks | 1 |  |  |
| **SM\_PER\_13** | The Station Manager needs to be able to access personnel information. |  | 1 |  |  |
| **SM\_SCH\_14** | The Station Manager needs to be able to view train schedules under his jurisdiction. |  | 1 |  |  |
| **TM\_VIW\_15** | The Traffic Manager needs to be able to view the location of every train in real time |  | 1 |  |  |
| **SM\_PER\_16** | The Station Manager is allowed to access passenger information. |  | 1 |  |  |
| **P\_TKT\_17** | Passengers are able to book tickets for trains at an available time, with specifications such as class or sleep accommodation |  | 1 |  |  |
| **P\_VIW\_18** | Passengers are able to view train information |  | 1 |  |  |
| **P\_TKT\_19** | Passengers are able to cancel their reservations within a reasonable timeframe | Timeframe TBD | 1 |  |  |
| **S\_PAY\_20** | System must process payment for the booking of a ticket either using card or cash on the ticket booths |  | 1 |  |  |
| **S\_NOT\_21** | System must notify all users about changes in the time table according to their jurisdiction. |  | 1 |  |  |
| **S\_NOT\_22** | System must notify External Maintenance on scheduled maintenance checks |  | 1 |  |  |
| **E\_WRK\_23** | Employees should be able to log their hours into the system |  | 1 |  |  |
| **E\_WRK\_24** | Employees should be able to check specific tasks issued to them from the Station Manager. |  | 2 |  |  |
| **SM\_WRK\_25** | Station Managers should be able to send specific tasks to Employees |  | 2 |  |  |
| **U\_VIW\_26** | All users must be able to see their personal information |  | 2 |  |  |
| **S\_ALR\_27** | System must be able to detect Alert sensors on railways, trains, maintenance centers, train depots and stations |  | 2 |  |  |
| **S\_ALR\_28** | System must notify the correct user for the specific alerts | Railway and Train alerts -> Traffic Manager  Station Alert -> Station Manager  Train Depots, maintenance Centers -> External Maintenance | 2 |  |  |
| **M\_ALR\_29** | Managers are allowed to call alerts on their own |  | 2 |  |  |
| **P\_SVC\_30** | Passengers can choose to call for accessibility services | “wheelchair accommodation, blindness accommodation, etc” | 3 |  |  |
| **P\_TKT\_31** | Passengers can opt into loyalty programs or alternate payment methods |  | 3 |  |  |
| **S\_CAP\_32** | System must observe train capacity to eliminate overbooking |  | 2 |  |  |
| **SM\_ALR\_33** | Station Managers are allowed to alert passengers to any of their alerts |  | 2 |  |  |
| **U\_AUTH\_34** | All users must have login logout functionality |  | 1 |  |  |
| **S\_FIN\_35** | The system must keep track of expenses and profit. |  | 1 |  |  |
| **SM\_FIN\_36** | Station Managers are able to check station revenue and expenses |  | 1 |  |  |
| **S\_FIN\_37** | The system must be able to generate detailed reports on revenue and expenses. |  | 1 |  |  |
| **S\_LOG\_38** | The system must keep track of all operations done in a specified timeframe |  | 1 |  |  |
| **S\_SCH\_39** | The system must keep track of the schedule of everything within a specific timeframe |  | 1 |  |  |
| **E\_WRK\_40** | Employees may request a shift change or time off |  | 3 |  |  |
| **A\_VIW\_41** | An admin user is allowed to view all info on the system |  | 1 |  |  |
| **A\_ADM\_42** | An admin user is allowed to access all functionalities of the system, including other users |  | 1 |  |  |

## Non-Functional Requirements

**Performance requirements:**

**1.System response time :**

* The system must respond to user requests, including ticket bookings, schedule lookups, and other operations ,within 4 seconds under typical operational conditions.

**2. Scalability:**

* The system must support a minimum of 100 concurrent users without performance degradation, ensuring it can handle seasonal or event-based traffic spikes.

3.**Data Processing Efficiency:**

* The system must be capable of processing large data sets (e.g., real-time train schedules, passenger data) within 3 minutes to facilitate timely decision-making.

**Security requirements :**

4.**Data Encryption:**

* All sensitive data, such as passenger details, payment information, and personal records, must be encrypted at rest and in transit using AES-256(Advanced Encryption Standard with a 256-bit key) encryption.

5. **Authentication and Authorization**

* The system must implement multi-factor authentication (MFA) for all users and enforce role-based access control (RBAC) to restrict sensitive operations based on user roles.

6. **Data Integrity**

* The system must ensure that data is accurate, complete, and unaltered by unauthorized users. All data modifications should be logged with timestamps and user identifiers.

**7.Activity Log**

* The system must maintain an audit trail of all critical operations, including user logins, data modifications, and system configurations, with logs retained securely for at least 12 months.

### 

### Usability Requirements:

**8.User Interface (UI) Design**

* The system must feature an intuitive, user-friendly interface for all user types (e.g., passengers, station staff, administrators), minimizing the need for extensive training.

9. **Cross-Platform Compatibility**

* The system must be fully responsive, providing an optimal user experience on both desktop and mobile platforms, supporting the major browsers and mobile operating systems.

10. **Localization and Internationalization**

* The system must support multiple languages (at least English, Spanish, and French) and be capable of adapting to different regional settings (e.g., date formats, currency).

11. **User Feedback and Support**

* The system must incorporate a mechanism for users to provide feedback or report issues. This feedback should be easily accessible to administrators for further analysis.

**Scalability and Maintainability Requirements:**

**12. Modular Architecture**

* The system must be designed using a modular architecture, enabling easy addition or removal of features and integration with third-party services without disrupting core functionality.

**13.Maintainability**

* The system must support regular maintenance activities (e.g., database backups, software updates)

14.**Data Backup and Restore**

* The system must implement daily backups of critical data, with the ability to restore data within 2 hours in the event of a system failure, ensuring our business continuity.

15. **Fault Tolerance and Recovery**

* The system must be fault-tolerant, ensuring continuity of service in case of hardware failures. It must provide automatic failover mechanisms.

### Product Requirements

#### 3.2.1.1 Usability Requirements

* The system must be detailed to ensure safety in operation
* A future mobile app for passengers should have simple and understandable menus

#### 3.2.1.2 Performance Requirements

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

* All traffic control management operations should be instantaneous (under 1 second)
* The system must handle at least 1000 users in every station on the railway grid
* All alert operations must be instantaneous (under 1 second)
* Menu loads must be done in under 3 seconds
* To ensure optimal performance, all stations must have a local server on which the operations and logs are kept.

#### 3.2.1.3 Availability

* System must be able to cover all the geographical area traced by the stations, railways, depots, and maintenance centers within jurisdiction.
* In case of system failure, the traffic control management system must restart itself in 10 minutes or less
* System maintenance should not interfere with traffic control management
* Maximum downtime must be 1 hour every month
* The system must be operational at all times

#### 3.2.1.4 Security

* All accounts and their respective passwords must be encrypted
* System must keep track of all operations in the form of logs.
* All data must be consistent and avoid duplication or loss of critical information
* All data must be ACID compliant to avoid data corruption
* Hourly backups of traffic control must be performed, and retrieval systems must be in place.
* Logs must be accessed by only the relevant parties, either the management in jurisdiction, or the admin.

### Organizational Requirements

* System must support different time zones, units, currencies, and language translations.
* System should be integrated with existing systems for the railway management system.

### External Requirements

* System must adhere to the relevant local authorities laws.

# 4.User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

* Describes a significant business need
* Identifies, documents, and ranks the problem that is driving the scenario
* Describes the business and technical environment that will resolve the problem
* States the desired objectives
* Shows the “Actors” and where they fit in the business model
* Is specific, and measurable, and uses clear metrics for success

Use cases are associated with a particular Functional Requirement. Assuming you have the first functional requirement named BR\_01, you will map it into the Use Case called UC\_01 and user scenario US\_01. Please keep this naming convention throughout all your use cases and diagrams.

# 5. Diagrams

# 

In this section you are going to place all of the diagrams that you build throughout the course, in following with the slides presented throughout the weeks.

5.1 ER Diagram

Standard ERD for your project. Not much but the skills gained in the DBMS course are required.  
  
5.2 Use Case Diagram (general)

Use Case Diagram (only one, with all the use cases).

5.3 Activity Diagram

Each Activity Diagram should be associated with an use case, associated with a particular requirement which is further associated with a particular use-case. E.g BR\_01 which becomes UC\_01 which becomes AC\_01.

5.4. Class diagram.

One class diagram (general) for all the classes. Edit it afterwards with the design pattern implemented in it.

5.5 State diagram

Place all the relevant state diagrams here.

5.6 Sequence diagram.

All sequence diagrams are associated with an Activity Diagram. A Sequence Diagram is built based on an activity diagram. If the activity diagram is named AC\_07, the Sequence Diagram will be named SC\_07.

5.7. Collaboration diagram

All collaboration diagrams directly relate to a sequence diagram. If a sequence diagram is named SC\_07, then the collaboration diagram is named CC\_07

# 6. Design Patterns

Choose the relevant design patterns for your project. For each, give a reasoning and the associated class and sequence diagram. These are NOT part of the above diagrams, and need not carry the following naming scheme.

# 7. Appendix.

# **Organizing the Requirements**

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

**By System Mode**

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

**By User Class**

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

**By Objects**

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

**By Feature**

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

**By Stimulus**

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

**By Response**

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

**By Functional Hierarchy**

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

**Additional Comments**

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.