

Requirements and Design Specification Document

<Smart Track>

Project Code:

Advisor:

Miss Aneeka Shayan

Project Team:

Moazam Attiq (221400003)

Ammar Qaiser (221400040)

Muhammad Ahmad Butt (221400006)

Muhammad Ibrahim Iyaz (221400027)

Submission Date:

Sep 10, 2023

Document Information

Category	Information
Customer	None (Project for SE-304)
Project	<Smart Track>
Document	Requirement Specifications
Document Version	Final Documentation
Identifier	
Status	Draft
Author(s)	<Moazam Attiq (221400003), Ammar Qaiser (221400040), Muhammad Ahmad Butt (221400006), Muhammad Ibrahim Iyaz (221400027)>
Approver(s)	
Issue Date	Aug 11, 2023
Document Location	
Distribution	

Definition of Terms, Acronyms and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Term	Description

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	5
2.	Overall Description	5
3.	Specific Requirements	6
3.1	Functionality	6
	<Functional Requirement One>	7
3.2	Usability	7
	<Usability Requirement One>	7
3.3	Reliability	7
	<Reliability Requirement One>	Error! Bookmark not defined.
3.4	Performance	7
	<Performance Requirement One>	7
3.5	Supportability	7
	<Supportability Requirement One>	Error! Bookmark not defined.
3.6	Design Constraints	Error! Bookmark not defined.
	<Design Constraint One..>	Error! Bookmark not defined.
3.7	On-line User Documentation and Help System Requirements	Error! Bookmark not defined.
3.8	Purchased Components	Error! Bookmark not defined.
3.9	Interfaces	Error! Bookmark not defined.
	User Interfaces	Error! Bookmark not defined.
	Hardware Interfaces	Error! Bookmark not defined.
	Software Interfaces	Error! Bookmark not defined.
	Communications Interfaces	Error! Bookmark not defined.
3.10	Licensing Requirements	Error! Bookmark not defined.
3.11	Legal, Copyright, and Other Notices	Error! Bookmark not defined.
3.12	Applicable Standards	Error! Bookmark not defined.
4.	System Architecture	Error! Bookmark not defined.
5.	Use Cases	Error! Bookmark not defined.
5.1	Use Case Diagram	Error! Bookmark not defined.
5.2	Use Case Description	Error! Bookmark not defined.
6.	Detailed System Design	Error! Bookmark not defined.
7.	Research and Development	Error! Bookmark not defined.
8.	Learning Outcomes	Error! Bookmark not defined.
9.	Practical Applications	Error! Bookmark not defined.
10.	Supporting Information	Error! Bookmark not defined.

1. Introduction

The beginning of the Software and Design Requirements Specification (SRS/DS) helps us understand the entire SRS and DS document better. It gives us a broad view of what's inside, including the purpose, scope, definitions, acronyms, abbreviations, references, and a summary of the SRS/DS content.

Our Railway Application Software is a solution to the challenges faced by the railway industry. It aims to make railway systems work better. This SRS/DS document contains all the details about what this software needs to do and how it should work. It's like a recipe book for building the software.

This project follows what we learned from a previous project and takes it further. We want to make the railway system even better by combining train control, managing traffic, helping passengers, doing maintenance, and making sure everyone can communicate smoothly.

This SRS/DS document is organized in a way that you won't need to read any extra papers. It's designed to be simple and understandable. We explain everything in regular words. As you read on, you'll discover all the details about how this software will work. It's like discovering the pieces of a puzzle that come together to create a modern and efficient railway system. The Railway Application Software is our step toward a better way of running trains – safer, more efficient, and making the future of travel brighter.

1.1 Purpose

The reason we're creating this Railway Application Software, as explained in the Software and Design Requirements Specification (SRS/DS), is to give a clear picture of how the software will work in the railway world. This document describes everything the software should do and how it should act. It covers both what the software will do, functional capabilities, like controlling trains and managing passengers, and how well it should do it, non-functional capabilities, like being safe and working fast. It also considers things like how the software should look and what technology it can use. To sum up, the purpose of this Railway Application Software is to guide its creation, make sure everyone understands what it needs to do, and help create a better and more efficient railway system.

1.2 Scope

The scope of this project is to create a railway application software that combines various features such as train control, traffic management, ticketing, maintenance, and communication. The goal is to improve the efficiency of railway operations, enhance safety measures, provide a better experience for passengers, and enable real-time coordination for effective control. By addressing the challenges faced by the railway industry and offering a single solution, this software will help modernize and make the railway system more reliable.

1.3 Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **SRS/DS**.]

1.4 References

Smith, J., & Johnson, A. (2021). "Improving Railway Operations through Integrated Software Solutions." *International Journal of Railway Engineering*, 15(2), 123-145.

Brown, R., & Davis, M. (2020). "Enhancing Passenger Experience in the Railway Industry: A Case Study of Integrated Ticketing Systems."

Journal of Transportation Technology, 8(3), 210-225.

Anderson, L., & Wilson, C. (2019). "Real-time Coordination for Improved Control in Railway Systems." Proceedings of the International Conference on Railway Engineering, 45-58.

Garcia, M., & Patel, S. (2018). "Streamlining Railway Operations through Integrated Maintenance Systems." Journal of Infrastructure Management, 12(4), 315-330.

Thompson, E., & Roberts, K. (2017). "Enhancing Infrastructure Reliability in the Railway Industry: A Comparative Study of Signaling Systems." Transportation Research Part C: Emerging Technologies, 79, 123-138.

1.5 Overview

Booking unreserved train seats using the Pakistan Railway Station app is a straightforward process that involves utilizing QR codes for convenience. After downloading and installing the app, passengers can access the QR code booking option in the app's navigation bar. At the train station, passengers can locate the QR code and use it to book a seat for their desired destination. Making the payment through the app confirms the booking instantly. Once the booking is complete, passengers receive a unique QR code via a message, which they can scan within the app to access their booked train seat. This innovative approach simplifies the ticketing process and enhances the travel experience for passengers.

2. Overall Description

2. Product Overview

This section provides an overview of the factors that influence the product and its associated requirements. It sets the context for the detailed requirements defined in Section 3, enhancing their clarity and comprehension.

2.1 Product Perspective

The product perspective encompasses the relationships between the proposed software and its surroundings. This includes interactions with other software systems, hardware components, or external interfaces. It helps establish how the software fits into the larger technological landscape.

2.2 Product Functions

The intended functions of the product are outlined in this section. It highlights the core capabilities and operations that the software is designed to perform. Understanding these functions sets the groundwork for developing specific requirements that align with the desired outcomes.

2.3 User Characteristics

The characteristics of the users who will interact with the software are discussed here. This includes their expertise level, technical familiarity, and potential roles. Recognizing the diversity of user characteristics informs the design of user interfaces and overall user experience.

2.4 Constraints

Constraints refer to the limitations or restrictions that impact the development and operation of the software. These could be related to technical, regulatory, or environmental factors. Addressing constraints at this stage aids in crafting requirements that accommodate these limitations effectively.

This section serves as an introductory framework for the detailed requirements outlined in Section 3. By providing insights into the product's perspective, functions, user characteristics, and constraints, it enhances the comprehension and relevance of subsequent requirement specifications.

- assumptions and dependencies
- requirements subsets]

3. Specific Requirements

[This section of the **SRS/DS** contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the Use Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

3.1 Functionality

Step1: Download and install the Pakistan railway station app , which can be used to book unreserved train seats.

Step2: A QR book option will be place on the navigation bar of the railway station Pakistan online mobile app

Step3: Go to the train station app where QR code is located.

Step4: Choose the destination where you want to travel.

Step5: Make the payment to immediately book the train ticket.

Step6: Once a ticket is booked passenger receive a message with QR code Scan URL.

Step7: Passenger simply scan the QR code through QR code scanner.

<Functional Requirement One>

[The requirement description.]

3.2 Usability

- It provides usability warning messages based on the present usability study, and can be used for both generating and reading QR codes in railways online system after booking the ticket online. Passenger receives a QR Code and reads it to receive payment from the passenger's account if they pay amount through QR code Scanner.

<Usability Requirement One>

[The requirement description goes here.]

3.3 Reliability

Reliability Consideration

Ensuring the accuracy and dependability of QR code scanning is crucial for the proper functioning of the system. The scanning process must consistently decode the information contained within QR codes without errors. Any deviations or inaccuracies during decoding could result in incorrect passenger information or erroneous ticket details being processed.

Reliability Requirement: QR Code Decoding

<Reliability Requirement One>

Requirement Description:

The QR code scanning mechanism must have a minimum accuracy rate of 99%. This means that out of 100 QR codes scanned, no more than one QR code can be inaccurately decoded. This level of accuracy guarantees that passenger information and ticket details are correctly processed and avoids potential discrepancies that could adversely affect the travel experience.

This reliability requirement sets a high standard for the accuracy of QR code scanning, enhancing the overall dependability of the system and minimizing the chances of errors in passenger data interpretation.

3.4 Performance

The QR code scanning process should be quick and responsive. Passengers should not experience significant delays while waiting for the system to process their QR codes.

3.5 Supportability

Ensure that the QR code scanning functionality is designed in a way that allows for easy version updates. New versions should be compatible with existing systems and not disrupt ongoing operations.

Design the QR code scanning functionality in a modular way, with clear separation of components. This makes it easier to identify and fix issues without disrupting the entire system. Use coding practices that promote maintainability, such as clear commenting and adherence to coding standards.

3.8 Purchased Components

This section outlines the various purchased components that will be integrated into the system, along with relevant licensing, compatibility considerations, and interface standards.

1. PayPal Integration:

The system will incorporate PayPal as a purchased component for online payment processing. This integration allows users to make secure transactions using their PayPal accounts, enhancing the convenience and security of the payment process.

Licensing: The PayPal integration will adhere to PayPal's terms of use and licensing agreements.

Compatibility: The system will ensure seamless compatibility with PayPal's APIs and software development kits (SDKs).

Interface Standards: The integration will follow PayPal's recommended interface standards for consistent and user-friendly payment experiences.

2. Bank ATM Cards:

The system will support integration with various bank ATM cards, enabling users to make payments directly from their bank accounts. This component enhances payment options for users who prefer using their ATM cards.

Licensing: Integration with bank ATM cards will comply with respective bank agreements and regulations.

Compatibility: The system will be compatible with standard banking protocols and interfaces to securely process ATM card transactions.

Interface Standards: The integration will adhere to industry standards for secure communication between the system and banks' payment gateways.

3. Credit and Debit Cards:

The system will include support for credit and debit card payments, allowing users to conveniently make transactions using their cards. This component broadens the range of payment options available to users.

Licensing: Integration with credit and debit cards will follow payment network regulations and agreements.

Compatibility: The system will be designed to work with major credit card networks and processors.

Interface Standards: The integration will meet industry standards for secure data transmission during credit and debit card transactions.

4. Third-Party APIs:

Incorporating third-party APIs for various services, such as location-based services or additional payment gateways, will enrich the system's functionality.

Licensing: Integration with third-party APIs will be guided by the terms and conditions set by the respective service providers.

Compatibility: The system will be compatible with the APIs' endpoints and data formats.

Interface Standards: Integration will adhere to API documentation and standards provided by the third-party service providers.

Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

3.9 *User Interfaces*

The software will feature user interfaces that provide an intuitive and user-friendly experience for interacting with the application. These interfaces will encompass navigation menus, input forms, buttons, and displays that allow users to access and utilize the software's features effectively.

Hardware Interfaces

The software will interface with various hardware components to ensure smooth operation and functionality. This includes compatibility with physical devices such as printers, scanners, and sensors. The software will interact with these hardware components based on their logical structures, physical addresses, and expected behaviors.

Software Interfaces

[The software will establish interfaces with external software components, whether they are purchased, reused from other applications, or developed for subsystems outside the scope of this Software Requirements Specification (SRS). These interfaces will enable seamless interaction between the software and other software components, enhancing the overall system functionality.]

Communications Interfaces

The software will incorporate communications interfaces to enable interaction with external systems and devices. This encompasses connections to local area networks (LANs), remote serial devices, and similar communication channels. These interfaces will facilitate data exchange and communication between the software and other systems or devices.