SRS Documentation

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the Train Management System (TMS). The system is designed to manage trains, passengers, and stations, providing functionalities for adding, removing, and listing these entities.

1.2 Scope

The Train Management System will:

Allow administrators to add, remove, and list trains, passengers, and stations.

Provide a user-friendly menu-driven interface for interaction.

Ensure data integrity through input validation.

1.3 Definitions

Train: A vehicle that transports passengers along a fixed route.

Passenger: A person traveling on a train.

Station: A location where trains stop to pick up or drop off passengers.

1.4 References

Google

2. Overall Description

2.1 System Overview

The Train Management System is a console-based application that allows administrators to manage trains, passengers, and stations. It provides functionalities for adding, removing, and listing these entities.

2.2 User Characteristics

Administrators: Users who manage trains, passengers, and stations.

End Users: Passengers who use the system to view train and station information.

2.3 Assumptions and Dependencies

The system will be used by a single administrator at a time.

The system will run on any platform with a Java Runtime Environment (JRE).

3. Functional Requirements

3.1 Add Train

Description: The system shall allow the administrator to add a new train.

Inputs: Train number, destination, capacity.

Outputs: Confirmation message.

Validation:

Train number must be unique.

Capacity must be a positive integer.

3.2 Add Passenger

Description: The system shall allow the administrator to add a new passenger to a specific train.

Inputs: Passenger name, train number.

Outputs: Confirmation message.

Validation:

Train must exist.

Train must have available capacity.

3.3 Add Station

Description: The system shall allow the administrator to add a new station.

Inputs: Station code, station name, location.

Outputs: Confirmation message.

Validation:

Station code must be unique.

3.4 List Trains

Description: The system shall display a list of all trains.

Inputs: None.

Outputs: List of trains with details (train number, destination, capacity, number of passengers).

3.5 List Passengers

Description: The system shall display a list of all passengers.

Inputs: None.

Outputs: List of passengers with details (name, train number).

3.6 List Stations

Description: The system shall display a list of all stations.

Inputs: None.

Outputs: List of stations with details (station code, station name, location).

3.7 Remove Train

Description: The system shall allow the administrator to remove a train and all associated passengers.

Inputs: Train number.

Outputs: Confirmation message.

Validation:

Train must exist.

4. Non-Functional Requirements

4.1 Performance

The system shall respond to user inputs within 2 seconds.

4.2 Usability

The system shall provide a simple, menu-driven interface for ease of use.

4.3 Reliability

The system shall handle invalid inputs gracefully and provide meaningful error messages.

4.4 Security

The system shall ensure data integrity by validating all inputs.

4.5 Maintainability

The system shall be modular and well-documented for easy maintenance.

5. System Features

5.1 Menu-Driven Interface

The system shall provide a menu with the following options:

1. Add New Train

2. Add Passenger

3. Add Station

4. List All Trains

5. List All Passengers

6. List All Stations

7. Remove Train

8. Exit

5.2 Input Validation

The system shall validate all inputs to ensure data integrity.

5.3 Error Handling

The system shall display meaningful error messages for invalid inputs or operations.

6. System Models

6.1 Use Case Diagram

|  |
| --- |
| Administrator |

1. Add Train

2. Add Passenger

3. Add Station

4. List Trains

5. List Passengers

6. List Stations

7. Remove Train

|  |
| --- |
| Train Management  System |

6.2 Class Diagram

|  |  |  |
| --- | --- | --- |
| Train | Passenger | Station |
| - trainNumber  - destination  - capacity  - passengers | - name  - trainNumber | - stationCode  - stationName  - location |
| + getTrainNumber()  + getDestination()  + getCapacity()  + setDestination()  + increaseCapacity()  + decreaseCapacity()  + addPassenger()  + removePassenger() | + getName()  + getTrainNumber() | + getStationCode()  + getStationName()  + getLocation() |

7. Other Requirements

7.1 Documentation

The system shall include a README.md file with instructions for setup and usage.

7.2 Testing

The system shall include unit tests for all critical functionalities.

7.3 Deployment

The system shall be deployed as a standalone Java application.

8. Appendices

8.1 Glossary

TMS: Train Management System.

JRE: Java Runtime Environment.

8.2 Acronyms

SRS: Software Requirements Specification.