Software Requirements Specification

Version 1.0

October 4, 2020

Railway Reservation System

Ankit Kumar Yadav (2K18/SE/024)
Ashish Kumar (2K18/SE/041)
B.Tech.

Object Oriented Software Engineering
5th Sem

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope of Project

1.3 Definitions, acronyms and abbreviations

1.4 References

1.5 Overview

2. Overall description

2.1 Product perspective

2.1.1 System Interfaces
2.1.2 User Interfaces

2.1.3 Hardware Interfaces
2.1.4 Software Interfaces
2.1.5 Communication Interfaces
2.1.6 Memory Constraints
2.1.7 Operations
2.1.8 Site adaption Requirements

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

2.6 Apportioning of requirements

3. Specific Requirements

3.1 External interfaces

3.1.1 User Interfaces

3.1.2 Hardware Interfaces
3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional requirements

3.2.1 Login
3.2.2 Register
3.2.3 Update Profile
3.2.4 View Train Details
3.2.5 Maintain Train Details
3.2.6 Ticket Booking

3.2.7 Booking History

3.2.8 Cancellation of tickets

3.2.9 Logout

3.3 Performance requirements
3.4 Safety requirements

3.5 Security requirements

3.5 Software system attributes

3.6 Logical Database Requirements

1. Introduction

This document namely software requirement specification (SRS) gives a detailed description of the Railway Reservation System, brief description about purpose of the proposed system, scope, acronyms and abbreviations used for the system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. The document provides clear and brief description about the requirements and stakeholders involved in the system. It also concentrates on the capabilities required by the stakeholders while defining the system.

1.0. Purpose

The purpose of this project is to develop software for railway reservation system with various functional and non- Functional part of design namely,

- Book a ticket,
- Cancel ticket,
- Viewing train information,
- Viewing booking details,
- Updating profiles.
- Maintaining Train details

1.1. Scope of Project

This software system will be a Railway Reservation System for Administrator, Customers and staff. This system will be designed to provide an interface where the customers, admin, staff can view train details. Customer and admin can book tickets and also cancels the ticket. Moreover, the customers can view their booking history and administrator shall be able to view all customers' booking histories. Also the customers and staff have access to update their profiles.

More specifically, this system is designed to allow the staff and admin to maintain the details of trains and the admin can manage the details of customers as well as of staff.

1.2. Definitions, acronyms and abbreviations

The definitions and acronyms used in RRS are given as:

- SRS : software requirement specification
- RRS : Railway Reservation system
- RAM : Random Access memory
- Admin/Administrator: User having all the privileges to operate the RRS.
- Staff: User having privilege to maintain train details.
- Customer: Any user who books a ticket in the system.

1.3. References

a) Object-Oriented Software Engineering by Yogesh Singh & Ruchika Malhotra, PHI Learning Pvt. Ltd., 2012

b) IEEE Recommended Practice for Software Requirements Specifications – IEEE Std. 830-1998.

1.5 ***Overview of the Document***

The rest of the SRS document describes various system requirements, interfaces, features and functionalities.

2.0. Overall Description

This section contains the details of product functionalities, hardware details which are discussed below. It also contains the detailed product perspective from different Stakeholders, it provides detailed description of product characteristics with permitted constraints, assumptions and dependencies and requirement subsets.

The RRS help customers, admin, and staff to login in to the system. The RRS help customers, admin to book a ticket, cancel a ticket, searching for available trains on different routes, their booking history and brings/save the same data to/from the system. The RRS calculates total payment to pay for ticket booking and give various payment options to pay for the same. RRS also calculate return amount in case of cancellation of ticket.

The admin will have to maintain details of the following:

- Customers
- Customer's booking history
- Staff
- Train

The staff will perform the following functions:

- Login to the system
- Maintain train details
- View Train details
- Updating their personal profiles
- Logout from the system

The Customer will perform the following functions:

- Login to the system
- Book a ticket
- Cancel a ticket
- View a train on a specific route
- View booking history
- Updating their personal profiles
- Logout from the system

The Admin will perform the following functions:

- Login to the system
- Book a ticket
- Cancel a ticket
- Maintain train details
- View a train on a specific route
- View booking histories of customer
- Add new customer and staff
- Updating existing customer and staff profiles
- Logout from the system

2.1 Product Perspective

The Railway reservation System shall be developed using client/server architecture and will be compatible with any Operating System. The front end of the system will be developed using HTML and CSS, JavaScript, Bootstrap and the back-end will be developed using PHP and MySQL. It provides simple database rather than complex ones for high requirements and it provides good and easy graphical user interface to both new as well as experienced user of the computer.

2.1.1 System Interfaces

None

2.1.2 User Interfaces

The RRS will have the following user-friendly and menu-driven interfaces:

a) Login: to allow the entry of only authorized users through valid username and password.

b) Register: to allow new customers to register themselves into the system to access RRS.

c) View Train: to allow the admin, customers and staff to view the train details.

d) Book ticket: to allow the admin and customers to book tickets.

e) Booking history: to allow the admin and customer to view booking histories.

f) Cancel ticket: to allow the admin and customers to cancel tickets.

g) Profile details: to allow the admin, customers and staff to maintain their profiles.

h) Maintain Train: to allow the admin and staff to add, update, delete the details of train.

i) Logout: to allow the admin, customers and staff to logout from the system.

2.1.3 Hardware Interfaces

a) Screen Resolution of at least 640 * 480 or above.
b) Support for printer.
c) Computer systems will be in the networked environment as it is a multi-user system.

2.1.4 Software Interfaces

a) It can run on any Operating system.
b) HTML, CSS, JavaScript, Bootstrap for designing front end.
c) PHP and MySQL for designing the back end.

2.1.5 Communication Interfaces

In the RRS, communication is via local area network (LAN).

2.1.6 Memory Constraints

At least 512MB RAM and 500MB space of hard disk will be required to run the software.

2.1.7 Operations

None

2.1.8 Site Adaptation Requirements

The terminal at the client side will have to support the hardware and software interfaces specified in sections 2.1.3 and 2.1.4, respectively.

2.2 Product functions

- The system shall allow the admin, customers and staff to login using valid username and password.
- The system shall allow the Admin and staff to maintain the details of trains.
- The system shall allow the admin, customers and staff to view the details of train.
- The system shall allow the admin and customers to book tickets, view their booking histories and also cancels the tickets.
- The system shall allow the customers and staff to update personal profiles.
- The system shall allow the admin to add new customers and staffs and also allow admin to maintain existing customers and staffs details.
- All the actors (Admin, staff, and customer) involved will be able to log out from the system.

2.3 User Characteristics

Qualifications: At least matriculation and comfortable with English.

Experience: The Administrator and the staff should be well versed with the details of train as well as the users.

Technical Experience: Elementary knowledge of computers.

2.4 Constraints

- There will be only one administrator.
- Staffs are not allowed to register through RRS system. Only admin can add new staff.
- Customer will not be allowed to update the primary key.
- The delete, add and update train details is only available to the administrator and the staff. To reduce the complexity of the system, there is no check on delete operation. Hence, the admin and staff should be very careful before deletion of any record and he/she will be responsible for data inconsistency.

2.5 Assumptions and Dependencies

- The List of the trains as well as no. of seats in train must be available already.
- Staffs are already registered into the system and have their username and password for login.
- Customers should have basic knowledge of computing.
- Admin, customers and staff should be connected to internet to access the system.

2.6 Apportioning of Requirements

Not required

3.0 Specific Requirements

This section contains the software requirements in detail along with the various forms to be developed.

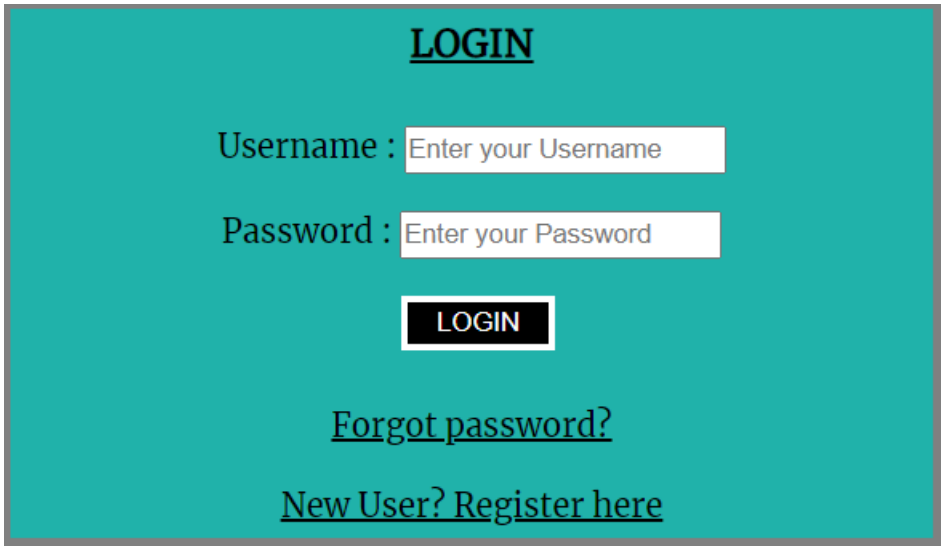
3.1 External Interface Requirements

3.1.1 User Interfaces

The following user interfaces will be provided by the system.

(i) Login

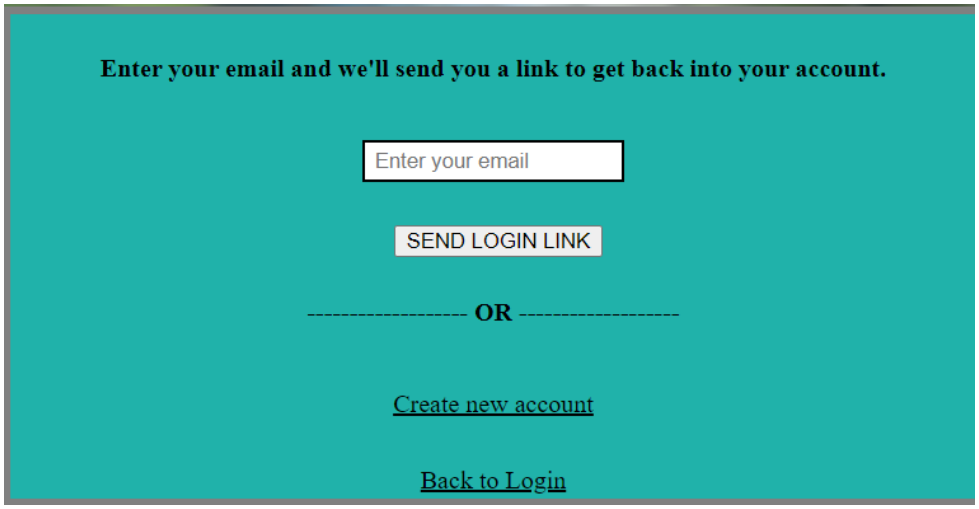
Login form will allow the admin, customer and staff to access system after logging in.



Various fields available on this form will be:

- **Username:** Username should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet and blank spaces are not allowed.
- **Password:** Alphanumeric of length in range of 8 to 50 characters. Blank spaces are not allowed. However, special symbols are allowed.

Forgot password

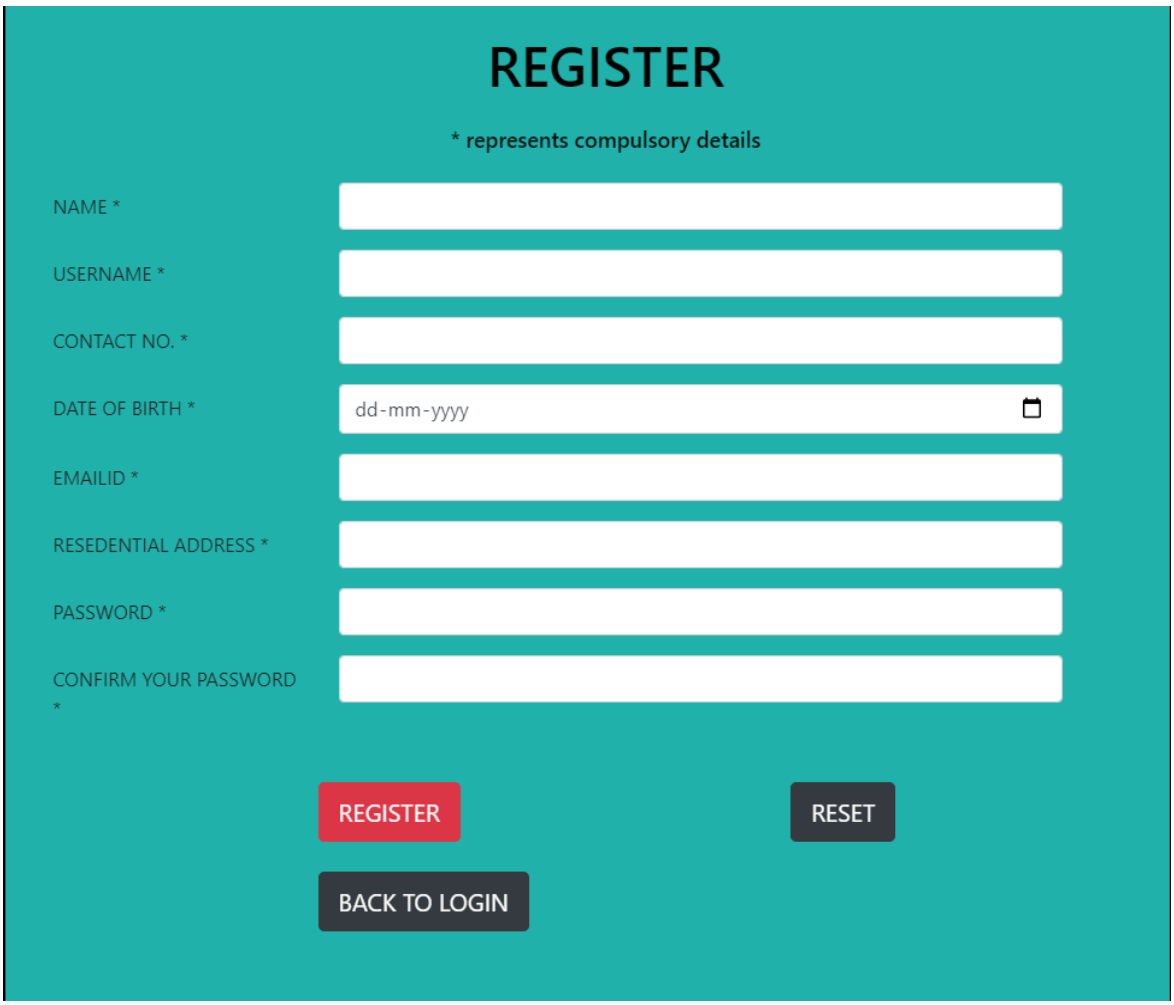


Various fields available on this form will be:

- **Email ID:** Email id should not be blank and should be alphanumeric of length 5 to 50. Must contain '@' and '.' Symbols.
- **Back to login:** Option to go to login page.
- **Create new account:** Option to register new users.

(ii) Register

Register interface allow the customers to register themselves to access the system. The customer will be prompted with the following interface if he chooses to Register.

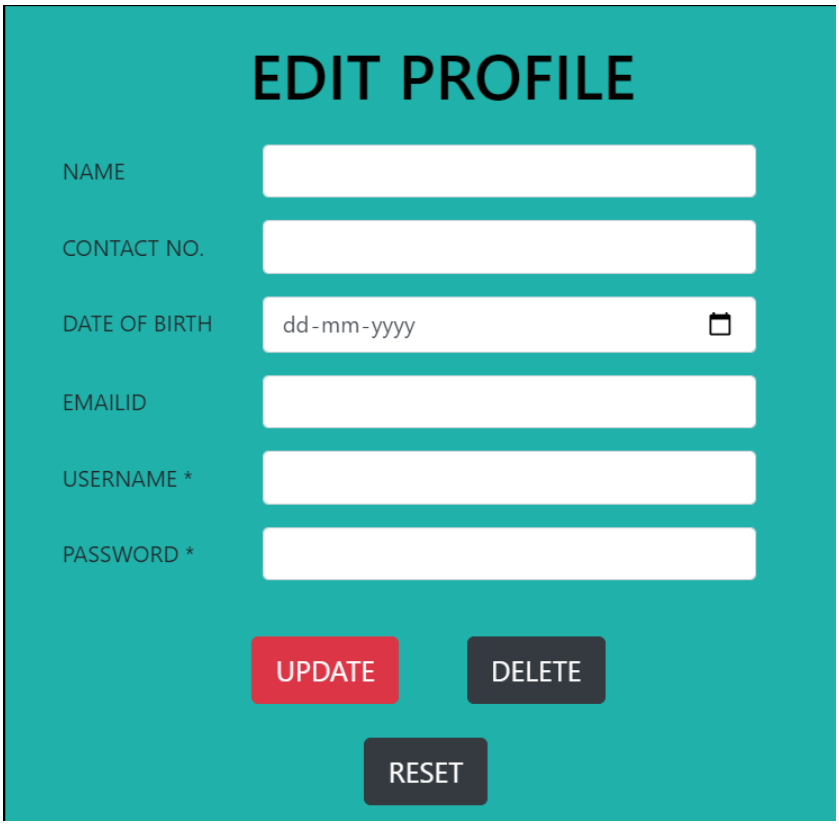


Various fields available on this form will be:

- **Name:** Must be of string type with 5 to 20 characters long. Special characters and blank spaces are not allowed.
- **Username:** It should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet. Special characters and blank spaces are not allowed.
- **Contact Number:** Must be a numeric of length 10. Blank spaces are not allowed.
- **Date of Birth:** Should not be blank and must be in dd/mm/yyyy format.
- **Email ID:** Should be alphanumeric of length 6 to 32. Blank spaces are not allowed. Email must have one '@' and '.' Symbol.
- **Address:** Must be alphanumeric of length 10 to 500. Blank spaces are allowed.
- **Password:** Alphanumeric of length in range of 8 to 15 characters. Blank spaces are not allowed. However, special symbols are allowed.
- **Confirm Password:** Alphanumeric of length in range of 8 to 15 characters. Blank spaces are not allowed. However, special symbols are allowed. It should be match with password.
- **Back to login:** Option to go to login page after registering.

(iii) Update Profile

Update Profile interface allows the customer as well as staff to update their profiles. Admin will be able to add, update, delete customer and staff details. The customer, admin, staff will be prompted with the following interface if he chooses to Update Profile.

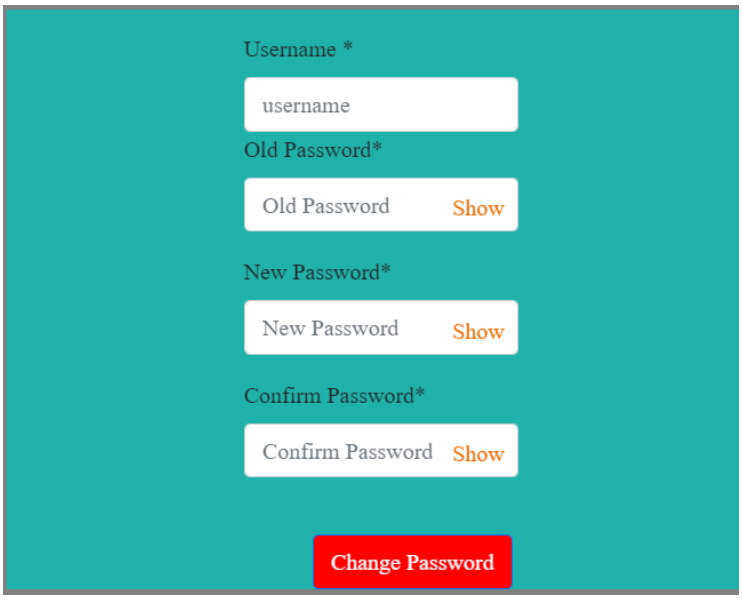


Various fields available on this form will be:

- **Name:** Must be of string type with 5 to 20 characters long. Special characters and blank spaces are not allowed.
- **Username:** It should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet. Special characters and blank spaces are not allowed.
- **Contact Number:** Must be a numeric of length 10. Blank spaces are not allowed.
- **Date of Birth:** Should not be blank and must be in dd/mm/yyyy format.
- **Email ID:** Should be alphanumeric of length 6 to 32. Blank spaces are not allowed. Email must have one '@' and '.' Symbol.
- **Password:** Alphanumeric of length in range of 8 to 15 characters. Blank spaces are not allowed. However, special symbols are allowed.

After choosing CHANGE PASSWORD option, the following interface will be prompted to the admin, customer as well as staff where he/she can change profile's password.

Change password

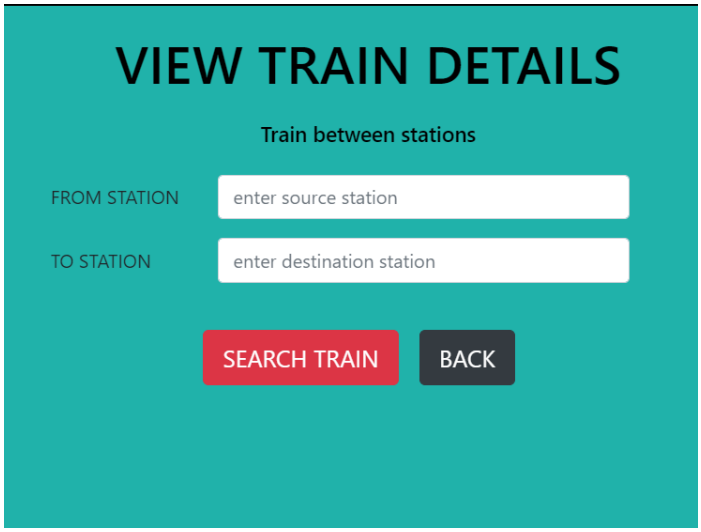


Various fields available on this form will be:

- **Username:** It should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet. Special characters and blank spaces are not allowed.
- **Old Password:** Alphanumeric of length in range of 8 to 15 characters. Blank spaces are not allowed. However, special symbols are allowed.
- **New password:** Alphanumeric of length in range of 8 to 15 characters. Blank spaces are not allowed. However, special symbols are allowed.
- **Confirm Password:** Alphanumeric of length in range of 8 to 15 characters. Blank spaces are not allowed. However, special symbols are allowed. It should be match with new password.

(iv) View train details

View train details interface allows the admin, customer as well as staff to view train details. The customer, admin, staff will be prompted with the following interface if he chooses View train details.



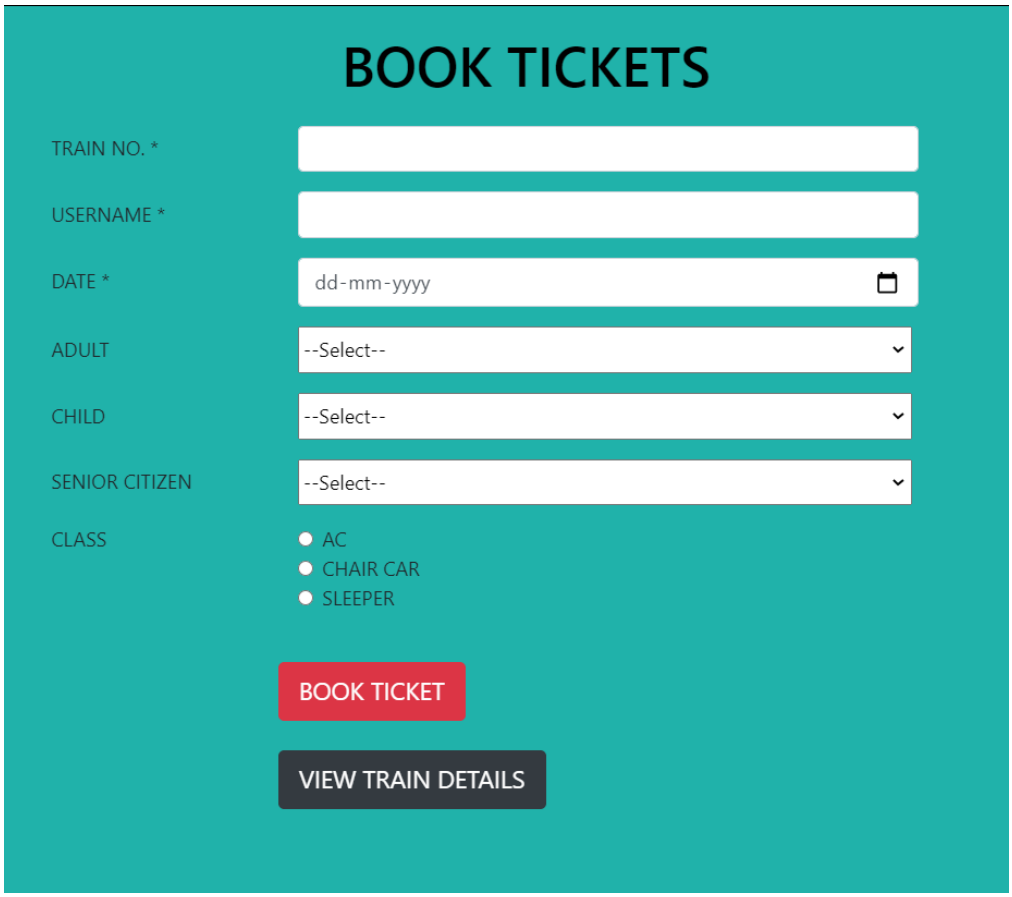
After clicking on SEARCH TRAIN button, the following interface will be prompted to the admin, customer as well as staff where he/she can see details of train.

Various fields available on this form will be:

- **Search Train:** option to search train by station name.
- **From** to which Source station customer is travelling, customer has to enter station name and it should not be blank and must be of string of length 2 to 30.
- **To** which Destination station, customer has to enter station name and it should not be blank and must be of String of length 2 to 30.
- **Train name:** should not be blank which is of string of length 5 to 30. Blank spaces are allowed.
- **Train No.**: should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.
- **Source and destination:** should not be blank and must be of string of length 2 to 30. Blank spaces are allowed.
- **Departure and arrival Timing**: Should not be blank and must be of time format.
- **Any train's day of running:** would not left empty for whole week. Train will run for at least one single day. The days on which trains would run will appear as "Y", otherwise "N" that means train is not running on that day.

(v) Ticket Booking

Ticket booking interface allows the admin and customer to book train tickets. The Admin and customer will be prompted with the following interface if he chooses Book Tickets

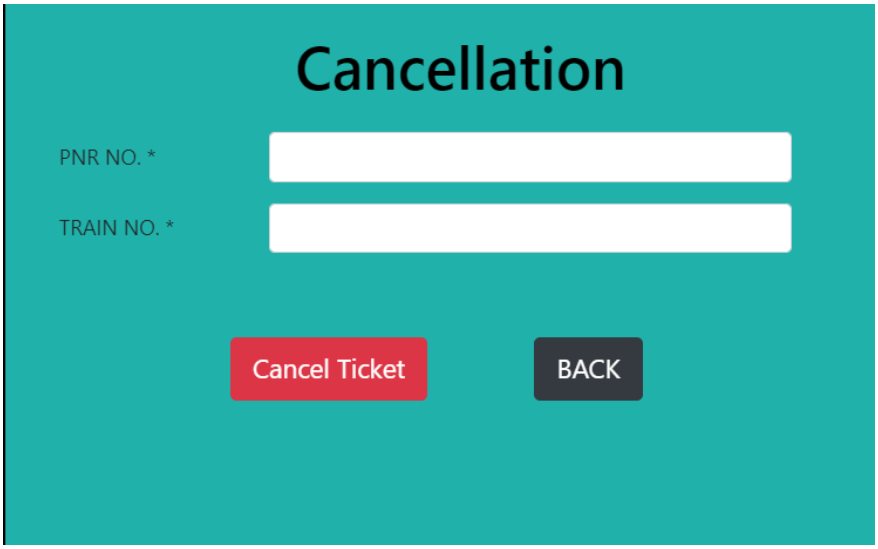


Various fields available on this form will be:

- **Book Ticket:** option to book train ticket after specifying reqd. details.
- **View Train Details:** option to view train details.
- **Train No.**: should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.
- **Date:** Admin / Customer can choose any date as per his/her choice from the button in dd/mm/yyyy format.
- **Username**: should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet. It is compulsory at the time of booking.
- **Class**: should not be blank and customer can choose options for AC, sleeper, Chair car from radio buttons.
- **No. of customers:** should not be blank and must be of numeric of length upto 1 digits and can fill numbers with the help of buttons.

(vi) Cancellation of tickets

Cancellation of tickets interface allows the admin and customer to cancel train tickets. The customer will be prompted with the following interface if he chooses Cancellation.

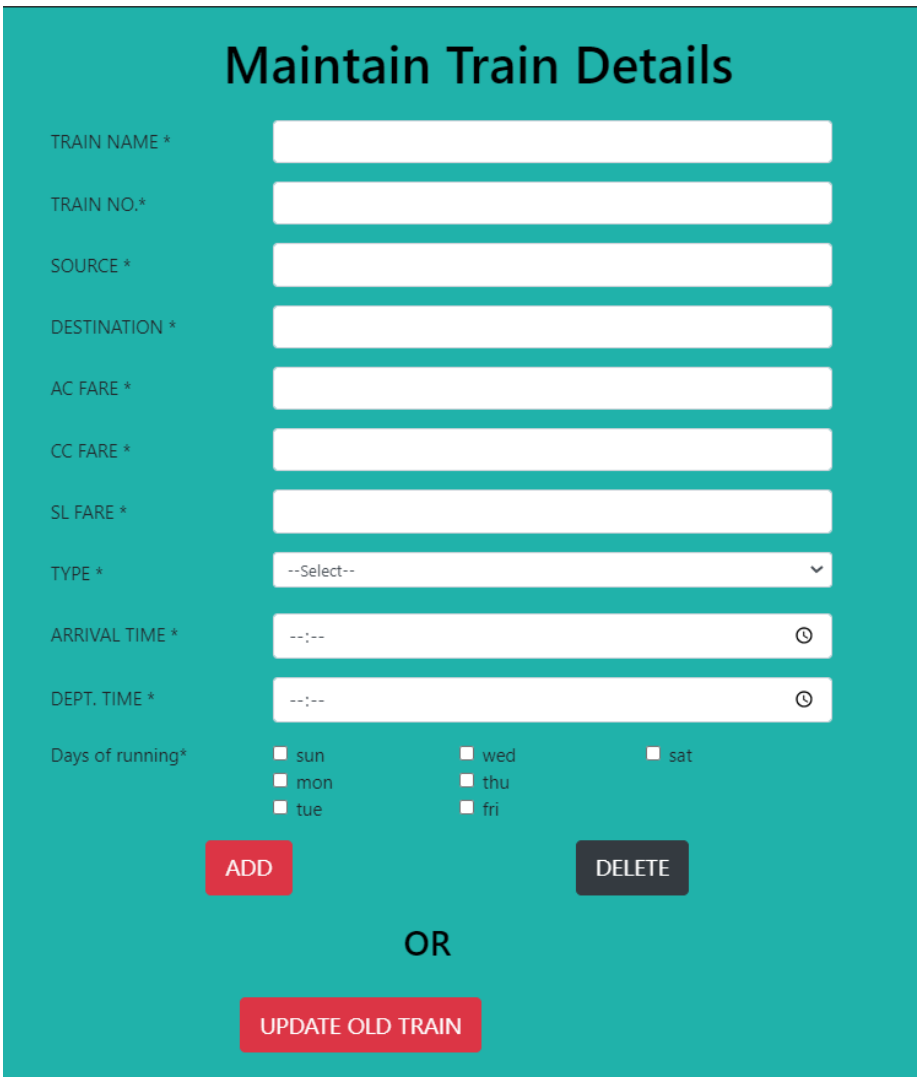


Various fields available on this form will be:

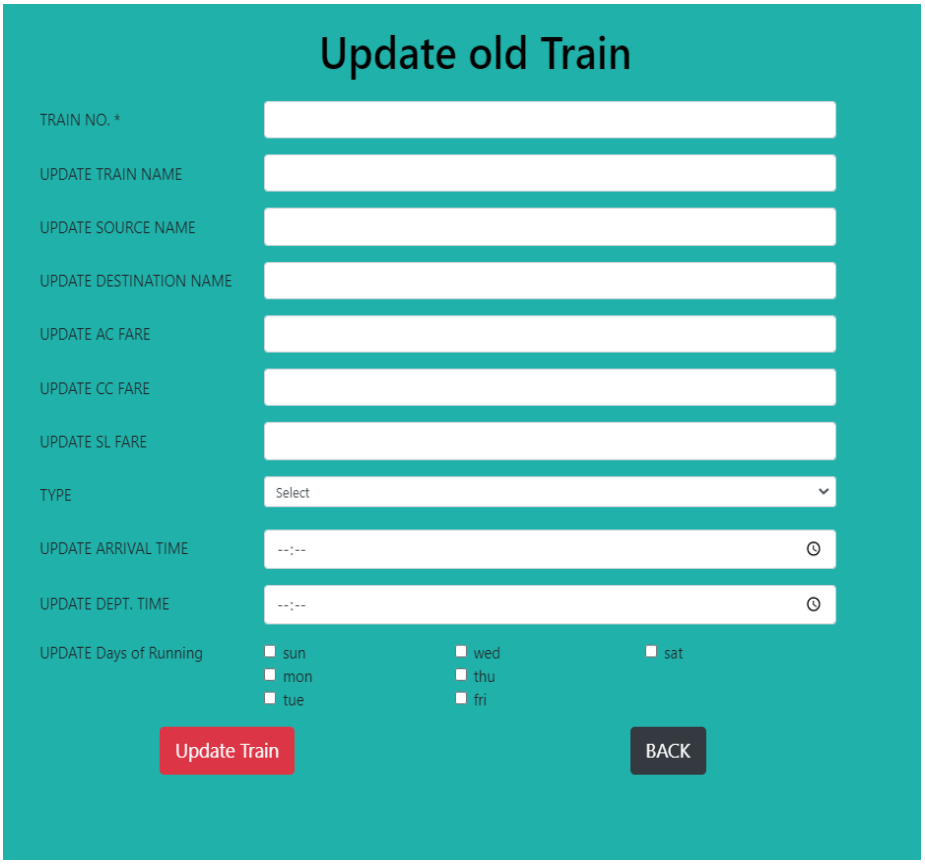
- **Cancel Ticket:** option to cancel booked train ticket after specifying reqd. details.
- **PNR No.**: Should not be blank and should be numeric of length 5. Special characters & blank spaces are not allowed.
- **Train No.**: Should not be blank and must be unique for each train and must be numeric of length 5. Spaces are not allowed.

(vii) Maintain train details

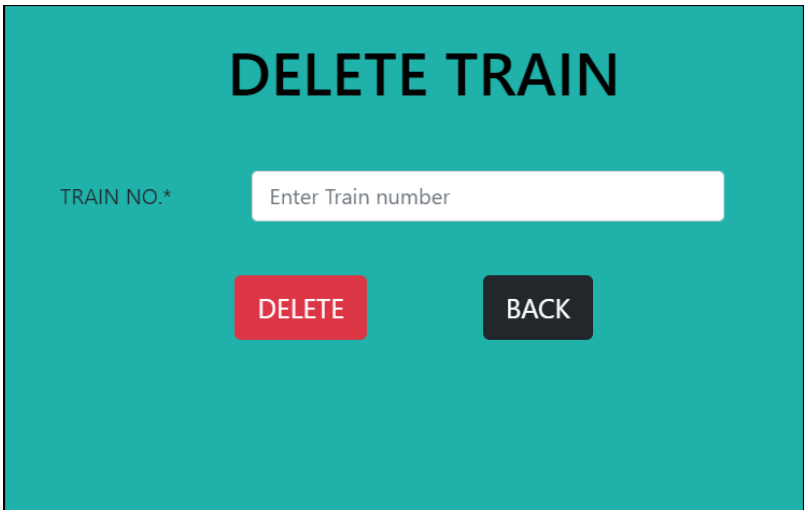
Maintain train details interface allows the admin, as well as staff to add, update or delete train details. The admin, staff will be prompted with the following interface if he chooses Maintain train details.



After choosing whether the admin/staff wants to maintain, i.e., add, delete or update the details of a train, the following interface will be prompted to the admin/staff when he/she chooses Update operation.



The following interface will be prompted to the admin and staff when he/she chooses DELETE.



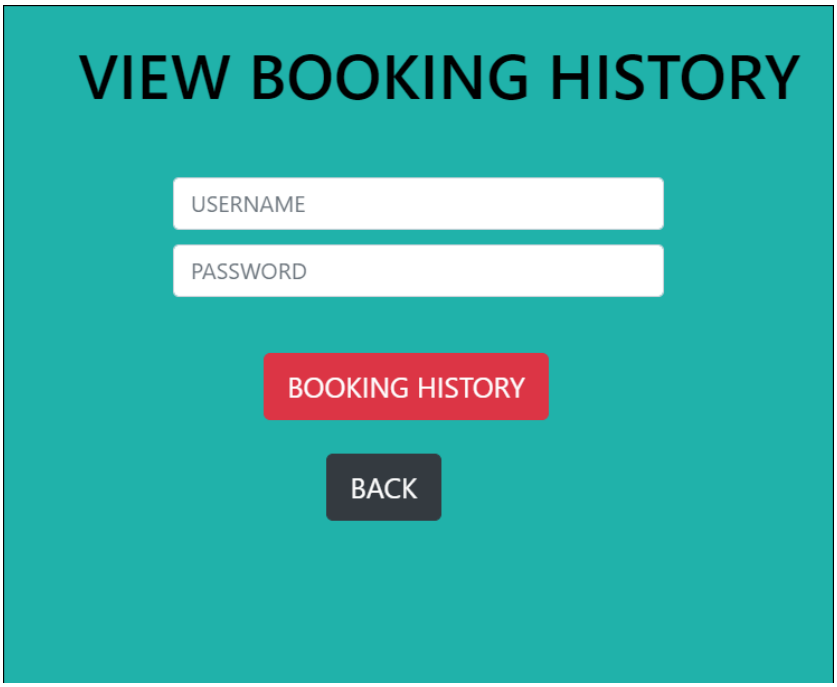
Various fields available on this form will be:

- **Train name:** Should not be blank and must be of string of length 5 to 30.Spaces are allowed.
- **Train No.**: Should not be blank and must be unique for each train and must be numeric of length 5. Spaces are not allowed.
- **Source and destination: S**hould not be blank and must be of string of length 2 to 20 characters. Spaces are allowed.
- **Departure and arrival Timing**: Should not be blank and must be of time format.

- **Type of train:** Should not be blank and must be of string of length 5 to 30.
- **Fares** of different classes should not be blank and must be of numeric of length 2 to 5.
- **Day of running of trains**: Should not be blank and can be chosen from the checkboxes, each day must be of string of length 6 to 9 characters.

(viii) Booking History

Booking history interface allows the admin, customer to view their booking histories. The customer and admin will be prompted with the following interface if he chooses Booking History.



After clicking on BOOKING HISTORY button, the following interface will be prompted to the admin, customer where he/she can see their booking history.

Various fields available on this form will be:

- **Username:** should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet.
- **Password:** Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters.
- **Train No.**: should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.
- **PNR No.**: Should not be blank and should be numeric of length 5. Special characters & blank spaces are not allowed.

- **Source and destination:** should not be blank and must be of string of length 2 to 30. Blank spaces are allowed.
- **Departure and arrival Timing**: Should not be blank and must be of time format.
- **Fare:** Should not be blank and must be of numeric of length 2 to 5.
- **Date of travelling:** Should not be blank and must be in dd/mm/yyyy format.
- **Class type:** Should not be blank and must be of string of length 5 to 20.
- **No. of customers:** Should not be blank and must be of numeric of length upto 1 digit.

3.1.2 Interfaces

(a) Screen Resolution of at least 640 * 480 or above.
(b) Support for printer.
(c) Computer systems will be in the networked environment as it is a multi-user system.

3.1.3 Software Interfaces

(a) It can run on any Operating system.
(b) HTML and CSS for designing front end.
(c) PHP and MySQL for designing the back end.

3.1.4 Communication Interfaces

In the RRS, communication is via local area network (LAN).

3.2 Functional Requirements

Functions are described below with the help of use case description.

3.2.1 Login

A. Use case description

1.	**Introduction-** This use case document the steps that must be followed for login into the system.
2.	**Actors-** Administrator, Staff, Customer
3.	**Pre Condition-** None
4.	**Post Condition-** After successful execution of this use case, the actors shall be able to login into the system.
5.	**Flow of events-** **Basic Flow:** **Basic Flow 1: Login** 1. The system shall prompt the login screen to the actor. 2. Actor enters his/her username and password. 3. The credentials are authenticated and access to the system is granted. **Alternative Flow:** **Alternative Flow 1: invalid password or username** 1. If password or username is not valid, or left empty then an appropriate error message is flagged and system will redirect the actor to beginning of the basic flow. **Alternate Flow 2: Forgot password** 1. If actor forgot his/her account's password, then he/she can choose this option to reset account's password. 2. Actor has to enter registered email id, then system validates with the database and if it matched, then a password reset link shall be sent to the corresponding email ID, from where he/she can reset password. **Alternative Flow 3: actor exits** 1. If the actor exits in the middle of the use case, the use case terminates.
6.	**Special Requirements-**None
7.	**Associated Use Cases-** None

B. Validity Checks

- All actors shall be able to access this usecase.
- Username is userID of the customer which will be used during login and should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet.
- Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters.
- Email ID should not be blank and should be alphanumeric of length 5 to 50. Must contain '@' and '.' Symbols.

C. Sequencing Information

None

D. Error Handling

If any of validation flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.2 Register Use Case

A. Use case description

1.	**Introduction-** This use case documents the steps that must be followed in order to create a new account and access the system.
2.	**Actors-** Administrator, Customer
3.	**Pre Condition-** None.
4.	**Post Condition-** After successful execution of this use case, a new user account will be created for the customer.
5.	**Flow of events-** **Basic Flow :** 1. The customer enters the following details: • Name • Username • Contact Number • Date of Birth • Emailid • Residential Address • Password • Confirm password 2. The customer clicks the "Register" button. 3. A new user account is created for the customer. **Alternative Flow :** **Alternate Flow 1 – Customer already exists** 1. If username entered by the customer is already present in the system database, then an error message is generated that the customer with this username already exists. 2. Control goes back to the beginning of the basic flow. **Alternative Flow 2 – invalid details** 1. If any of the above-mentioned attributes are invalid or left empty, like name contains numeric digits, contact number is not of 10 digits, email id doesn't contain '@' and '.' and password is not of atleast 8 characters, then an error message is generated, telling the customer "invalid details" or "empty fields" respectively and system will redirect the actor to beginning of the basic flow. **Alternative Flow 3 : Actor exits** 1. If the Actor exits in the middle of the use case, the use case terminates.
6.	**Special Requirements**-None
7.	**Associated Use Cases-** None

B. Validity Checks

- Only customers are allowed to access this use case.
- For registering into the system, customer has to fill above mentioned fields; otherwise appropriate message will be displayed like “empty fields”.
- Username is userID of the customer which will be used during login and should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet.
- Name cannot be blank and must be of string type with 3 to 50 characters long. Blank spaces are allowed. Special characters are not allowed.
- Email ID should not be blank and should be alphanumeric of length 5 to 50. Must contain ‘@’ and ‘.’ Symbols.
- Contact Number should not be blank and must be an integer of length 10.
- Date of Birth should be choose from the button and must be in dd/mm/yyyy format. It should not be blank
- Address should not be blank and must be alphanumeric of length 2 to 500. Blank spaces are allowed.
- Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters.
- Confirm Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters. It should be match with password; otherwise appropriate message will be displayed.

C. Sequencing Information

None

D. Error Handling

If any of validation flows does not hold true, appropriate message will be prompted to the customer for doing the needful.

3.2.3 Update Profile

A. Use case description

1.	**Introduction-** This use case document the steps that must be followed for managing or updating customer and staff details.
2.	**Actors-** Administrator, Staff, Customer
3.	**Pre Condition-** Actors must be logged-in to the system.
4.	**Post Condition-** After successful execution of this use case, Actors shall be able to update or delete profile details.
5.	**Flow of events-** **Basic Flow:** **Basic Flow 1: update profile** 1. The customers and staff shall be able to update their profile by entering valid new name, emailid, DOB, contact no., after providing their registered username and password and then by clicking on "UPDATE" button, profile would be updated in the database concurrently. Admin shall be able to update profiles of customer as well as of staff. **Basic Flow 2: Change password** 1. The actors shall be able to change his/her profile password by clicking on "Change password" option, wherein actors have to enter following details a) enter username b) enter old password c) enter new password **Basic Flow 3 : delete profile** 1. The admin shall be able to delete pre-existing customers and staff details. 2. The customer and staff shall be able to delete his/her own profiles by clicking "DELETE" button. Customer and staff have to provide their username and password for security purpose. **Alternative Flow:** **Alternative Flow 1** : **invalid details** 1. If any of the above-mentioned attributes are invalid or left empty, like name contains numeric digits, contact number is not of 10 digits, email id doesn't contain '@' and '.' or password is incorrect, username is not correct then an error message is generated, telling the actor "invalid detail" or "empty fields" respectively and system will redirect the actor to beginning of the basic flow. **Alternative Flow 2 : actor exits** 1. If the actor exits in the middle of the use case, the use case terminates.
6.	**Special Requirements**-None

7.	**Associated Use Cases**-login

B. Validity Checks

- All actors are allowed to access this use case.
- For updating profiles, actor have to fill above mentioned fields; otherwise appropriate message will be displayed like "empty fields".
- Username is userID of the actor and should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet. It is compulsory at the time of updation and deletion.
- Name cannot be blank and must be of string type with 3 to 50 characters long. Blank spaces are allowed. Special characters are not allowed.
- Email ID should not be blank and should be alphanumeric of length 5 to 50. Must contain '@' and '.' Symbols.
- Contact Number should not be blank and must be an integer of length 10.
- Date of Birth should be choose from the button and must be in dd/mm/yyyy format. It should not be blank
- Old Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters.
- New Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters. It should be match with password; otherwise appropriate message will be displayed.

C. Sequencing Information

None

E. Error Handling

If any of validation flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.4 View Train Details

A. Use case description

1.	**Introduction-** This use case document the steps that must be followed for viewing Train details.
2.	**Actors-** Administrator, Staff, customer
3.	**Pre Condition-** Actors must be logged-in to the system.
4.	**Post Condition-** After successful execution of this use case, Actors shall be able to view train details.
5.	**Flow of events-** **Basic Flow:** **Basic Flow 1: View train details** 1. First of all, actors have to enter source and destination name. 2. After entering details, Actor clicks on "SEARCH TRAIN" button , then actor is able to view train details like train name, train number, source, destination, train type, arrival and departure time, fares of different classes type and days of running. **Alternative Flow:** **Alternative Flow 1** : **invalid details** 1. If details provided by the actor are invalid or left empty, then a message is generated, telling the actor "Invalid details" or "empty fields" respectively and system will redirect the actor to beginning of the basic flow. **Alternative Flow 2** : **actor exits** 1. If the actor exits use case in the middle, the use case terminates.
6.	**Special Requirements**-None
7.	**Associated Use Cases-**Login

B. Validity Checks

- Search Train is an option to search train by station name.
- From to which Source station customer is travelling, customer has to enter station name and it should not be blank and must be of string of length 2 to 30.
- To which Destination station, customer has to enter station name and it should not be blank and must be of String of length 2 to 30.

- After customer clicks on “search button”
  - Train name should not be blank which is of string of length 5 to 30. Blank spaces are allowed.
  - Train No. should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.
  - Source and destination should not be blank and must be of string of length 2 to 30. Blank spaces are allowed.
  - Type of train should not be blank and must be of string of length 5 to 30.
  - Departure and arrival Timing should not be blank and is of time format.
  - Fares of different classes should not be blank and must be of numeric of length 2 to 5.
  - Any train’s day of running would not left empty for whole week. Train will run for at least one single day. The days on which trains would run will appear as “Y”, otherwise “N” that means train is not running on that day.

C. Sequencing Information

None

D. Error Handling

If any of validation/sequencing flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.5 Maintain Train Details

A. Use case description

1.	**Introduction-** This use case document the steps that must be followed for maintaining Train details.
2.	**Actors-** Administrator, Staff
3.	**Pre Condition-** Actors must be logged-in to the system.
4.	**Post Condition-** After successful execution of this use case, Actors shall be able to maintain the train details.
5.	**Flow of events-** **Basic Flow:** **Basic Flow 1: Add new train** 1. Actor enters the following details: • Train Number • Train Name • Source • Destination • AC fare, CC fare, SL fare • Train Type(can be selected from dropdown's choices) • Arrival time • Departure Time • Days of Running 2. Actor clicks the "ADD" button. 3. A new train is added in the database. **Basic Flow 2:Update train details** 1. Actor has to enter train number of train (it is mandatory) and one field (or more) that he/she wanted to update in train, rest of the fields can be left empty. Actor clicks the "UPDATE" button, and then train details is updated in database. **Basic Flow 3:Delete pre-existing train** 1. Actor has to enter train number of that train that he/she wanted to delete. 2. Actor clicks on "DELETE" button. 3. The system checks the records from the train database if it is matched with the actor entered details, then train is deleted from the database. **Alternative Flow:** **Alternative Flow 1** : **invalid details** 1. If any of the above-mentioned attributes are invalid or left empty, like train name contains numeric digits, train number is invalid and source, destination contains numeric digits, then actor will not be able to add, delete, update train and an error message is generated, telling the actor "invalid details" or "empty field" respectively and system will redirect the actor to beginning of the basic

/

	flow. **Alternative Flow 2** : **Train already exist (in adding new train)** 1. While adding a new train, if new train details entered by the actor are matched with the pre-existed train details, then actor will not be able to add new train and a message is generated, telling the actor "Train already exist" and system will redirect actor to beginning of the basic flow. **Alternative Flow 3** : **actor exits** 1. If the actor exits use case in the middle, the use case terminates.
6.	**Special Requirements**-None
7.	**Associated Use Cases-** Login

B. Validity Checks

- Only admin/staff is allowed to access maintain train details.
- Train name should not be blank which is of string of length 5 to 30. Blank spaces are allowed.
- Train No. should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed. It is compulsory for adding, updating or deleting train
- Source and destination station name should not be blank and must be of string of length 2 to 20 characters. Blank spaces are allowed.
- Type of train should not be blank and must be of string of length 5 to 30.
- Fares of different classes should not be blank and must be of numeric of length 2 to 5.
- Departure and arrival Timing should not be blank and is of time format.
- Day of running of trains should not be blank and can be chosen from the checkboxes, each day must be of string of length 6 to 9 characters.

C. Sequencing Information

None

D. Error Handling

If any of validation/sequencing flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.6 Ticket Booking

A. Use case description

1.	**Introduction**- This use case document the steps that must be followed for booking of tickets.
2.	**Actors**- Administrator, Customer
3.	**Precondition** - Actors must be logged-in to the system.
4.	**Post Condition**- After successful execution of this use case, Admin and customer shall be able to book tickets from the system.
5.	**Flow of events-** **Basic Flow:** 1. Actor has to enter username, Train number, Date of travelling, no. of customers and class Type. 2. Actor clicks on "CHECK SEAT AVAILABIITY" button and actor can view Train details also from "VIEW TRAIN DETAILS" button provided in booking page, if required. 3. Actor will be able to see no. of customers, Class type, seats available in particular class that he/she entered previously and fare to pay. 4. For booking ticket, Actor has to make payment; Actor will click on BOOK TICKET button. 5. Then Actor will choose mode of payment and this will take actor to the payment gateway. 6. After successful payment, Actor will get a notification of confirmation of payment and confirmation message of booking. **Alternative Flow:** **Alternative Flow 1: Invalid details** 1. If details entered by Actor do not match with system's database like he/she entered incorrect username / Train number or left empty or not in specified format (like train no. contains characters), then Actor is not be able to book tickets and a message is generated, telling the Actor "Invalid details" or "empty fields" respectively, then system will redirect Actor to beginning of the basic flow. **Alternative Flow 2: seats are not available** 1. If seats are not available in particular class that Actor entered, then Actor is not be able to book tickets and a message is generated, telling the Actor "Seats are not available" then system will redirect Actor to beginning of the basic flow. **Alternative Flow 3: Payment failed** 1. If payment got failed due to some issues, then Actor will not be able to book tickets and a message is generated, telling the actor "Payment failed" then system will redirect Actor to beginning of the basic flow.

	Alternative Flow 4: Actor exits 1. If the Actor exits in the middle of the use case, the use case terminates.
6.	**Special Requirements**- None
7.	**Associated Use Cases** – login, View train details

B. Validity Checks

- Admin and customer are allowed to book tickets.
- Username is userID of the actor and should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet. It is compulsory at the time of booking.
- Train No. should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.
- No. of customers should not be blank and must be of numeric of length upto 1 digits and can fill numbers with the help of buttons.
- Customer can choose any date as per his/her choice from the button, must be in dd/mm/yyyy format.
- Class should not be blank and customer can choose options for AC, sleeper, Chair car from radio buttons.
- Fare & seats available should not be blank and it retrieves from the database, so customers don't have to write in it.
- Payment details like card number, card No., expiry date etc. should not be blank and must be in specified format.

C. Sequencing Information

None

D. Error Handling

If any of validation/sequencing flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.7 Booking History Use Case

A. Use case description

1.	**Introduction**- This use case document the steps that must be followed for viewing booking history of customer.
2.	**Actors**- Administrator, customer.
3.	**Precondition** - Actors must be logged-in to the system.
4.	**Post Condition**- After successful execution of this use case, customer shall be able to view his/her booking history and admin shall be able to view all customers' booking histories.
5.	**Flow of events-** **Basic Flow:** **Basic Flow 1: View booking history** 1. First of all, actors have to enter username and password. 2. After entering details, Actor clicks on "BOOKING HISTORY" button , then actor is able to view booking history like train number, PNR number, source, destination, train type, arrival and departure time, no. of customers, fare and date of travelling. **Alternative Flow:** **Alternative Flow 1:Invalid details** 1. If username entered by the Actor is incorrect or left empty, then Actor is not be able to view booking history and a message is generated, telling the Actor "Invalid username" or "empty fields" respectively and system will redirect the customer to beginning of the basic flow. **Alternative Flow 2: Actor exits** 1. If the Actor exits in the middle of the use case, the use case terminates
6.	**Special Requirements**- None
7.	**Associated Use Cases** – login

B. Validity Checks

- Username is userID of the actor and should be alphanumeric of length 5 to 50 characters, should starts always with an alphabet.
- Password should not be blank and should be alphanumeric of length 8 to 30, can contain special characters.
- Train No. should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.
- PNR No. Should not be blank and should be numeric of length 5. Special characters & blank spaces are not allowed.
- Source and destination should not be blank and must be of string of length 2 to 30. Blank spaces are allowed.
- Departure and arrival Timing Should not be blank and must be of time format.
- Fare should not be blank and must be of numeric of length 2 to 5.
- Date of travelling should not be blank and must be in dd/mm/yyyy format.
- Class type **s**hould not be blank and must be of string of length 5 to 20.
- No. of customers **s**hould not be blank and must be of numeric of length upto 1 digit.

C. Sequencing Information

None

D. Error Handling

If any of validation/sequencing flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.8 Cancellation of tickets Use Case

A. Use case description

1.	**Introduction**- This use case document the steps that must be followed for cancellation of tickets.
2.	**Actors**- Administrator, customer.
3.	**Precondition** - Actors must be logged-in to the system.
4.	**Post Condition**- After successful execution of this use case, actors shall be able to cancel tickets.
5.	**Flow of events-** **Basic Flow:** **Basic Flow 1: cancel the ticket** 1. Actor has to enter PNR number and Train Number. 2. The system checks the records from the database if it is matched with the Actor entered details then Actor cancels the tickets by clicking "CANCEL TICKET" button. 3. After cancellation of ticket, actor will get a notification of confirmation of cancellation and message of refund amount. **Alternative Flow:** **Alternative Flow 1:invalid details** 2. If PNR number or train number entered by the Actor is incorrect or not in specified format (like train no. or PNR no. contains characters) or left empty, then no cancellation of ticket will proceed and a message is generated, telling the Actor "Invalid PNR number or train number" or "empty fields" respectively and system will redirect the Actor to beginning of the basic flow. **Alternative Flow 2: Actor exits** 1. If the Actor exits in the middle of the use case, the use case terminates.
6.	**Special Requirements**- None
7.	**Associated Use Cases** – login

B. Validity Checks

- PNR No. Should not be blank and should be numeric of length 5. Special characters & blank spaces are not allowed
- Train No. should not be blank and must be unique for each train and must be numeric of length 5. Special characters are not allowed.

C. Sequencing Information

None

D. Error Handling

If any of validation/sequencing flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

3.2.9 Logout Use Case

A. Use case description

1.	**Introduction**- This use case document the steps that must be followed for logout from the system.
2.	**Actors**- Administrator, Staff, customer.
3.	**Precondition** - Actors must be logged-in to the system.
4.	**Post Condition**- After successful execution of this use case, actor shall be able to logout from the system.
5.	**Flow of events-** **Basic Flow:** 1. The use case starts when the actors want to exit from the system. 2. The actor clicks on the "Logout" button. 3. The actors are logged out of the system and the login screen is displayed. **Alternative Flow:** None
6.	**Special Requirements**- None
7.	**Associated Use Cases** - login

B. Validity Checks

- Actor must be logged-in to the system.

C. Sequencing Information

None

D. Error Handling

If any of validation/sequencing flows does not hold true, appropriate message will be prompted to the actor for doing the needful.

1.3 Performance Requirements

RRS requires a system with at least a 500 megahertz CPU and 512 megabytes of RAM and a web browser to access it. However, internet connectivity might be required in case the system is hosted remote in a WAN.

1.4 Safety Requirements

If there is extensive damage to a wide portion of the database due to failure, such as crash, the recovery method restores a past copy of the database that was backed up to archival storage.
To ensure that no one of RRS's customers loses any data while using RRS (due to a crash or a bug of some kind) the developer team updates RRS regularly.

1.5 Security Requirements

All the admin, staff, customers have their unique username and password, so that no outsider will able to login to the system. Customers and staff must have their password to update their profile details or to change password. Admin, customers and staff must have their username at the time of accessing any system's feature.

1.6 Software System Attributes

Usability: The system will be user-friendly and easy to operate and the functions will be easy understandable.

Reliability: The system will be available to the admin and customers throughout the booking ticket period or any updation period and have a high degree of fault tolerance.

Security: The system will be password protected. Admin, customers and staff will have to enter correct username and password to access the system. No one can access functionalities of RRS without login to the system. Customers and staff must enter their password to update their profile details or to change password.

Maintainability: The system will be designed in a maintainable manner. It will be easy to incorporate new requirements in individual modules.

1.7 Logical Database Requirements

Login	Stores login details of admin, customers and staff
Customer	Stores details of Customers.
Train	Stores details of train.
Ticket	Stores details of booked ticket of customer.
Staff	Stores details of staff.
Admin	Stores details of admin.