USE CASE DESCRIPTION

1. Login Use Case

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

2. Register Use Case

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

3. Update Profile Use Case

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

4. View Train Details Use Case

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

5. Maintain Train Details Use Case

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

6. Ticket Booking Use Case

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

7. Booking History Use Case

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

8. Cancellation of tickets Use Case

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

9. Logout Use Case

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