**Train Ticket Booking**

# Problem Statement: -

Build a Train Ticket Booking System that allows customers to book train tickets online , search for availability of trains and seats and also allows for viewing and cancelling tickets online.

# Technologies used:

1. Java Spring
2. AngularJS
3. MongoDB
4. REST APIs
5. Junit for testing
6. Log4J Library
7. Apache Tomcat Server

# Features: -

1. The system should be able to add Train Routes consisting of all the middle stations.
2. System should allow customers and Train officials to login and perform various operations.
3. Searching for Trains from source to destination on particular date.
4. Checking for availability of seats in the train.
5. View/ Print booked Train Tickets. Also view past bookings.
6. Facility for cancellation of tickets.

## Users are privileged with following services: -

1. Register / Login as a customer by providing certain personal details
2. Search for Availability for trains for particular route and get the fare details.
3. Make fresh reservations for One or more tickets.
4. View Past bookings and allow cancellation of tickets.

## Administrator is privileged with following services: -

1. Add Train, Station and Route.
2. See users and passengers’ details.

# Business Scenarios: -

|  |  |
| --- | --- |
| BS1 | User A wants to book a ticket from Pune to Mumbai on 1-08-2018. |
| Prerequisites: - | * User A needs to visit “Ticket Booking” portal. |
| Input: - | Journey Details(Mandatory):   1. Source Station: Pune 2. Destination Station: Mumbai 3. Date: 1-08-2018 |
| Steps: - | [User A carries out the following steps]  BS1.1 Enter details and Search Trains  BS1.2 Choose from Available Trains  BS1.3 Accept Passenger Details  BS1.4 Confirm ticket by clicking “Book Now” option. |
| Result: - | User will be directed to Booked Tickets page for printing/viewing tickets. |
| BS1.1 | User A enters journey details and clicks “View Available Trains”. |
| BS1.1 Conclusion | List of all available trains will be displayed on the page. |
| BS1.2 | User chooses Train from displayed result. |
| BS1.2 Conclusions | Train details such as Train number, Fare, Train Type will be displayed. |
| BS1.3 | User enters passenger details such as Name, Age, Gender, etc. |
| BS1.3 Conclusion | Summary of entire ticket booking will be displayed. |

|  |  |
| --- | --- |
| BS2 | User A wants to view past bookings. |
| Prerequisites: - | * User A is logged in to the portal. * User A has done reservations in past. |
| Input: - | Reservation Details(Optional):   1. Booking ID 2. Date: 1-08-2018 3. Source or Destination Station |
| Steps: - | [User A carries out the following steps]   1. Enter details and Search for past bookings 2. Enter Filtering Criteria for result 3. Select booked ticket order. 4. View/Print details |
| Result: - | Train Booking details will be displayed to user. |

|  |  |
| --- | --- |
| BS3 | Adding Train Route to database. |
| Prerequisites: - | * Administrator is logged in to portal. |
| Input: - | Train & Route Details(Mandatory):   1. Add Train Details 2. Add intermediate stations details (Unique Sequence Number in Route) 3. Enter Arrival and Departure Time 4. Train Availability details |
| Steps: - | [Admin carries out the following steps]   1. Enter train details 2. Enter separate record for Each Intermediate Station between source and destination station in Train’s route. 3. Accept Distance in KMs from source for calculation of fare. 4. Enter availability details. |
| Result: - | Train route will be added to databse. |

|  |  |
| --- | --- |
| BS4 | Cancellation of reservation. |
| Prerequisites: - | * User is logged in to the portal. * Tickets booked in past. |
| Input: - | Booking Details:   1. Booking ID: PN1012MUM |
| Steps: - | [User carries out the following steps]   1. Click on “Ticket Cancellation” 2. Enter unique Booking ID or select booking from the list. 3. Click on “Cancel Reservation” |
| Result: - | Reservation cancelled message will be displayed. |

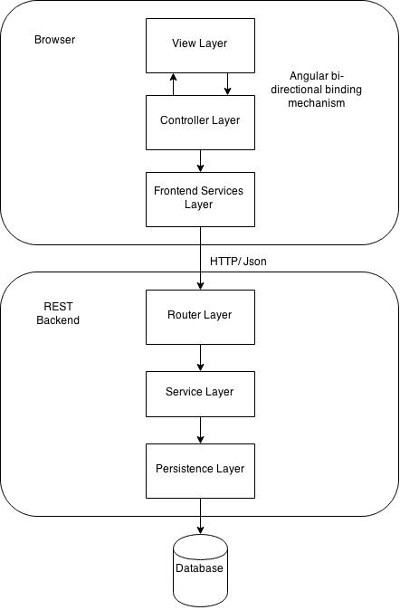
# Architecture Diagram: -[Draft]

The Frontend is built using following layers:

* View Layer: Views are used to represent the presentation layer which is provided to the end users
* Controller Layer: The Controller represents the layer that has the business logic. User events trigger the functions which are stored inside your controller. The user events are part of the controller.
* Services Layer: Models are used to represent your data. The data in your model can be as simple as just having primitive declarations

The backend is built using the usual backend layers:

* Router Layer: defines which service entry points correspond to a given HTTP url, and how parameters are to be read from the HTTP request
* Service Layer: contains any business logic such as validations, defines the scope of business transactions
* Persistence Layer: maps the database to/from in-memory domain objects



# ER Diagram:-

