**Documentation: Group 7**

Project Name *: MakeMyTrip*

**Milestones:**

1. Offering Data as a Service

Description

The milestone starts with the server layer designing where we will be designing and implementing the DAO layers which are PassengerDao, FlightsDao and BookingsDao. All the Dao Classes will be deliverable and at the end we will be able to push records into the NOSQL database where we will be using Mongo Db. In the PassengerDao class we will first design a model for it where we will include all the necessary fields having one primary key emailId which will be generated by the passenger once he signs up. This model will gather all the necessary information from the passenger who intends to book a flight ticket. The PassengerDao class will have all the essential methods in it required to perform any CRUD operation in the business layer. We will have various necessary methods to extract data and push the data to the database. This will give the opportunity to the passenger to create his account by giving his details, delete his account, update his account and view his details. Secondly In FlightDao we will design it in a way that when the passenger hits the search button after entering all the basic requirements to book a flight, he/she gets all the flights details on that route along with their cost and time. This will enable the passenger to view better options in terms of cost and time in order to book the most suitable flight. Information like flight name, date of arrival, date of departure, source, destination, flight capacity, travel charges etc will be displayed to the passenger in order to compare the most convenient flight for the journey. We will also generate an auto-generated unique sequence for this flightDao class which will be the flightId to distinguish between the flights. Lastly in the BookingsDao, we will have all the booking details of the passengers who have successfully made the payment and booked a ticket. This collection will have a primary key which will be PNR of the ticket. This will contain all the details about the journey including the date of booking, date of arrival, date of departure, passenger name, passenger’s email-id, source, destination and cost of the travel.

This DAO class will have all the necessary methods to interact with the database and fetch all the details related to the booking whenever the passenger wants to view it. It will also enable the passenger to delete his/her already booked ticket.

The above mentioned Dao classes will form the data service layer of our application and will be deliverable solely in the vertical architecture.

Stories

1. Creating model classes for Passenger, Flights and Bookings.
2. Learning and establishing a connection with Mongo Db.
3. Creating PassengerDao class:-

i)Create addPassenger() method which will take passenger object as an argument and return type is String, which will add the passenger details in collection.

ii)Create getAllPassenger() method of the return type List<Passenger> which will return the entire list of all the Passenger.

Iii)Create getPassenger() method which will take emailid as an argument and it will return Passenger details based on that id.

Iv)Create updatePassenger() method which will take emailid as an arguments and it will update details of user based on user requirement.

v) Create deletePassenger() method which will take emailid as an argumet and return type is Boolean through which we can eliminate the record of that passenger from the database.

1. Creating a getAllPassengers() method of the return type List<Passenger> which will return the entire list of all the Passengers and addPassenger() method of the return type Boolean which will return true value if the passenger is successfully added.

4. Creating a getPassenger() method of return type Passenger through which we can extract the data of a particular Passenger and Creating a deletePassenger() method of return type Boolean through which we can eliminate the record of that passenger from the database.

5. Creating a updatePassenger() method of return type Passenger through which we will update the information of the already existing passenger.

6.Creating FlighsDao having method getAllFlights() and getFlights() which will return the list of all functional flights available in the database and on the route and date mentioned by the passenger.

7.Creating a method checkSource() and checkDestination() in order to check if the source city and destination city exist in the airport list.

1. Creating BookingsDAO having method getBookingbyPNR() which will return flight details which has been booked by the particular passenger having PNR as a particular field