Software Design Document

-by Saumya Prabhakar

**Introduction:**  
Online Railway ticket reservation is very useful nowadays. This is very important to design a good-working system software for ticket booking and related transactions. To design it, full-track documentation of models(ER, DFD, Class, Use-case, Activity, Sequence) is required as per as software development is concerned.

**Features of the System:**

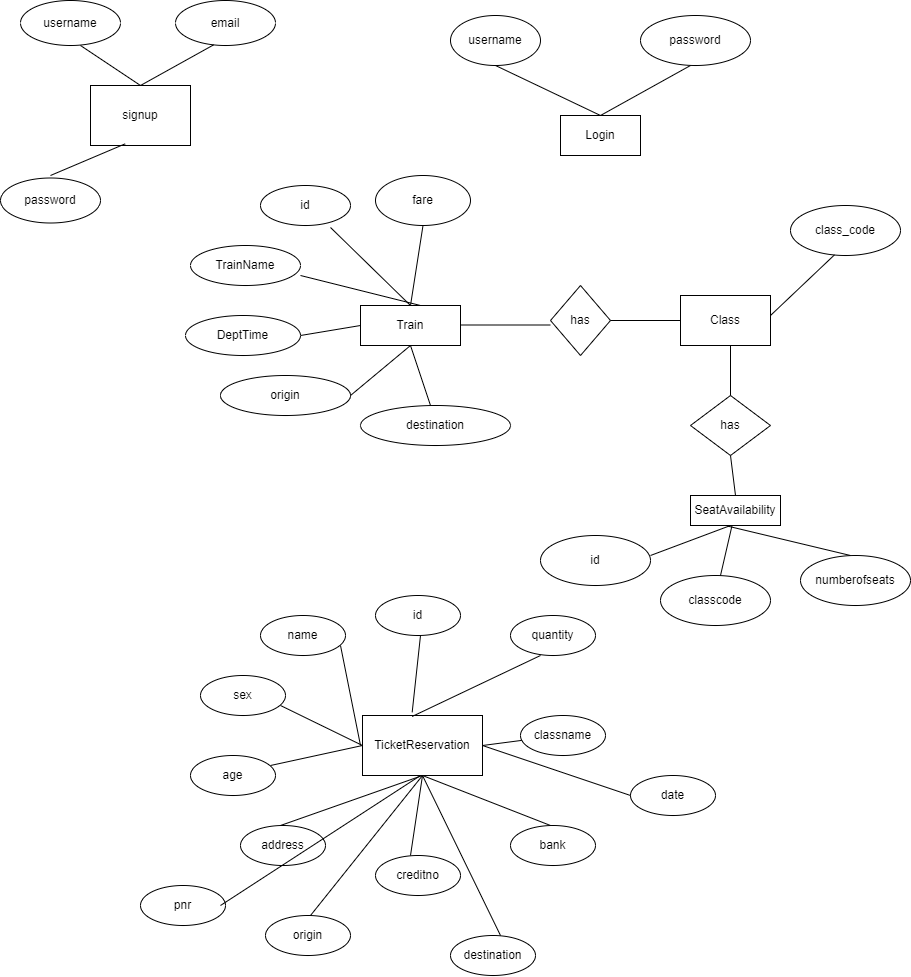
The Reservation system should contain the following features:

* If a passenger wants to reserve ticket(s), firstly, he/she has to log in to the Railway system with valid credentials. Then, the passenger has to provide his/her details with the date of the journey, names of the passengers and their details, origin station details, destination station details, and the class type of the required ticket(s).
* The Railway Reservation System will provide the available Train-list, and Seat-availability, via-details.
* The Reservation system should store all train details. This maintenance should be controlled by the Admin.

**Required Diagrams:**

Entity Relationship(ER) Diagram:

ER diagram displays the relations between the various entities(classes and their attributes) stored in the database. ER diagrams are very important for any database project. This diagram shows the communication between entities and their attributes:



The above ER diagram illustrates the key information about the railway reservation system, including entities like signup\_info, Login\_credentials, Ticket\_reservation, Train, Seat\_availability, Class, pay\_info. This diagram also shows the relationships between entities.

Entities and their attributes are:

* SignUp: Attributes of Signup entity are username, email and password.
* Login\_credentials: Attributes of Login\_credentials entity are username(PK) , password.
* Ticket\_reservation: Attributes of Ticket\_reservation entity are id, name , sex , age, address, creditno, bank ,classname, quantity, origin, destination, date, pnr.
* Train: Attributes of Train entity are Train\_code(PK), Destination, Train\_name, Start\_time, End\_time, fare.
* Seat\_availability: Attributes of Seat\_availability entity are Train\_code, Class\_code, and Number of seats.
* Class: Attributes of Class entity are Class\_id(PK).
* Pay\_info: Attributes of Pay\_info entity are payment\_id(PK), pay\_mode, amount, pay\_date, srl-no, PNR\_no, inst\_type, inst\_amt.

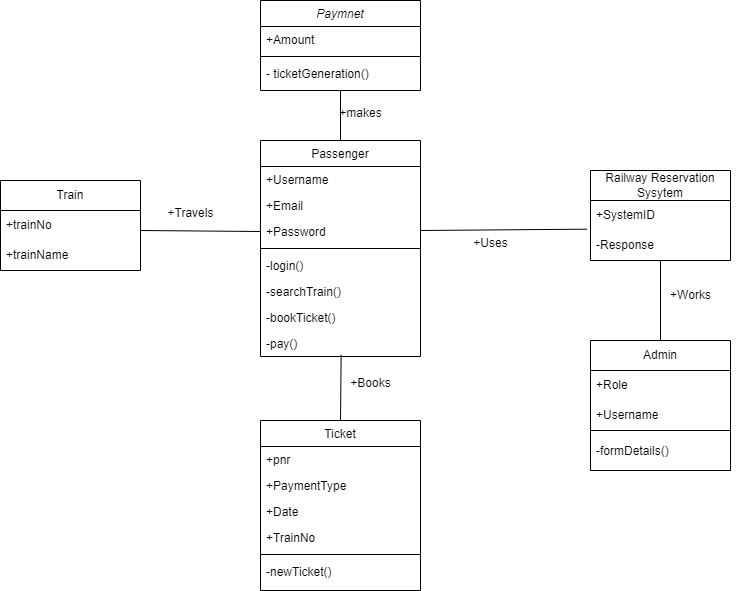
Relationships between Entities:

* One-to-one relation: signup to login\_credentials , Train to Via\_details, Class to train\_fare.
* Many-to-one relation: signup to ticket\_reservation, Class to seat\_availability.
* Many-to-many: Train to the station, Train to class.

**Class Diagram:**

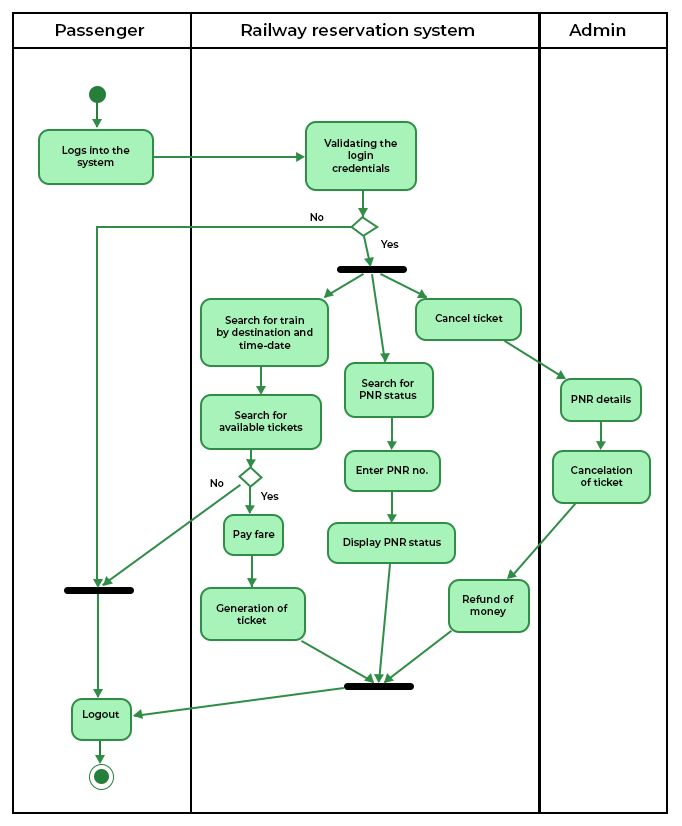
These diagrams describe the operation and attributes of a class with imposed constraints in the system. In this article the classes to be considered are ‘payment’, ‘train’, ‘passenger’, ‘ticket’, ‘railway reservation system’, ‘admin’. The description of the classes is given below.

|  |  |  |
| --- | --- | --- |
| **Class** | **Attributes** | **Processes** |
| Payment | amount | ticketGeneration |
| Train | train code, train name, frequency | — |
| passenger | Username, password | login, search Train,  pay charges, book tickets now ticket |
| ticket | PNR\_no, status, payment type, train code search train,  date of journey | new Ticket |
| railway reservation system | system | response |
| Admin | ID, name | formDetails |



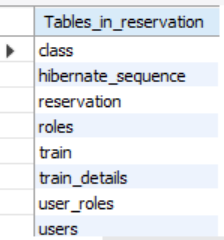
**Activity Diagram:**

This diagram shows the flow of processes from one to another activity.

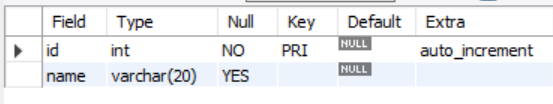


**Database Information:**

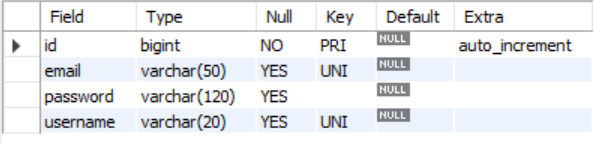
Tables in Used System as in Railway Reservation System.



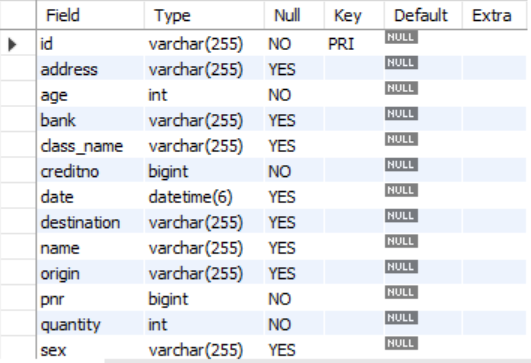
1.Roles Table



2.Users Table



3.Reservation Table



4. Train Details

