Railway Ticket Booking System

Problem statement:

1) The Railway Reservation System is basically developed in order to perform Online ticket reservation and inquiry. Every time a ticket is booked there must be a proper record to be maintained.

2) The reservation option enables a person to reserve for a ticket in an easier and efficient manner.

3) Once the train is available the person will be asked to enter his personal details and his requirements regarding the coach and berth.

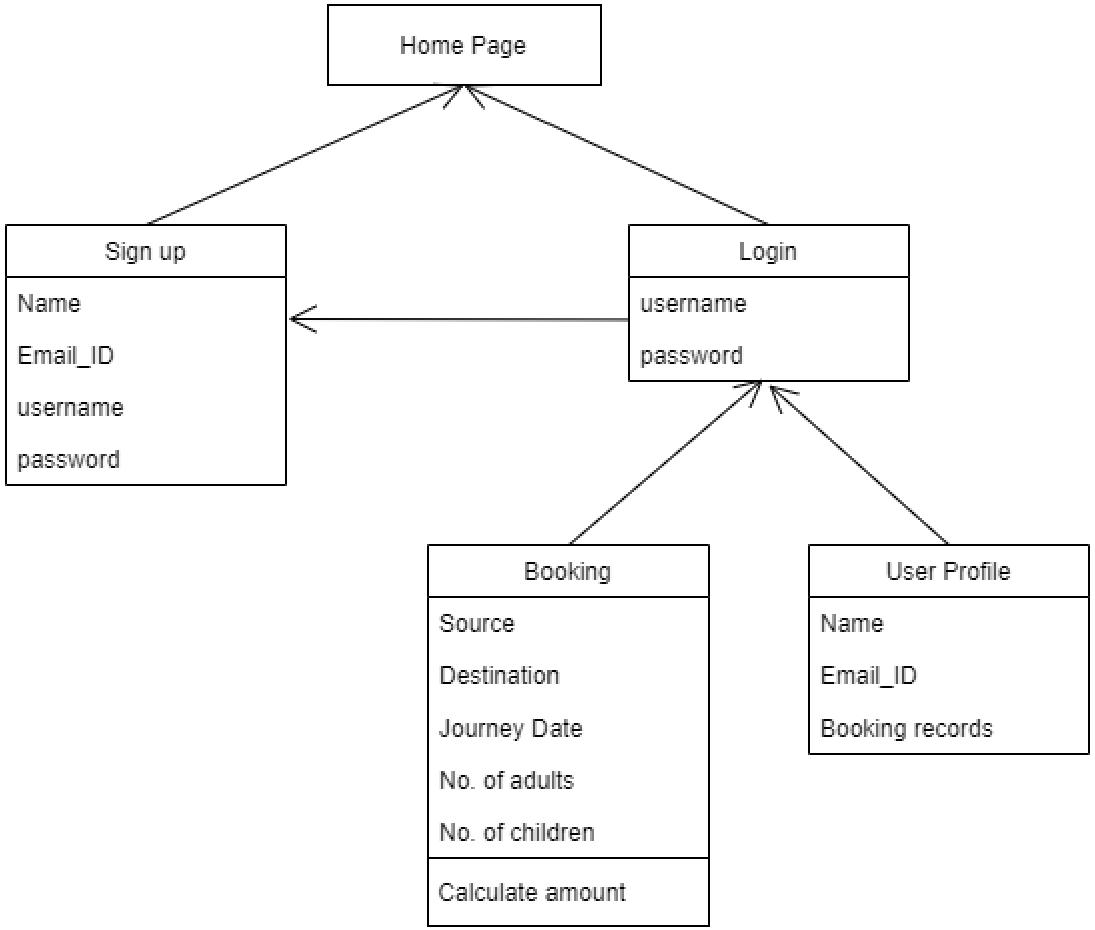
Objective:

1) The main objective of this System is to manage the details of users, trains, bookings.

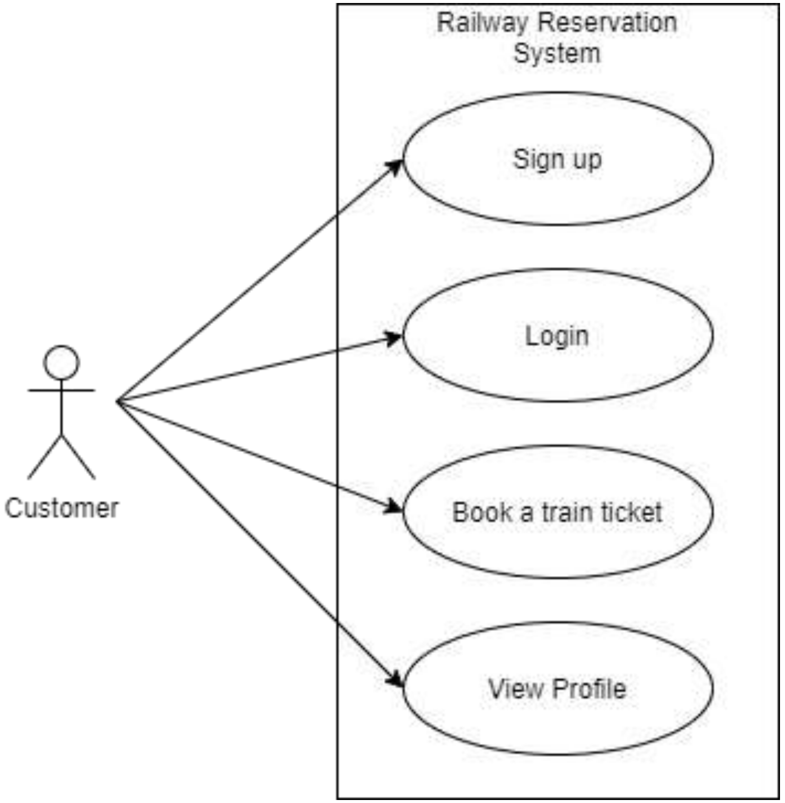
2) The aim is to automate the existing manual system by the help of computerized equipments so that valuable data/information can be stored for a longer period with easy accessing.

3)The project provides good performance and better service to the clients.

Class diagram:



Use Case Diagram:



Database tables used:

1)Train details – Train_id, source, destination.

2)Customer details- Cust_id, name, username, password, email-id

3)Booking record- Booking_id, cust_id, train_id, source, destination, amount

OOPS concepts implemented:

1) Data abstraction:

Data abstraction is the art of representing only the essential features and hiding the background details or explanations.

For example: While customer creates new account, we ask the user for name, email, password, phone number, DOB, but we only use the important/distinguishable features.

OOPS concepts implemented:

2) Encapsulation:

Encapsulation is the wrapping up of data and functions into a single unit called class.

Each application window has its own purpose and can be used by other classes without knowing it’s details.

For example: Homepage will call login or signup page, but it won’t know the working.

OOPS concepts implemented:

3) Modularity:

Modularity is the act of partitioning a program into individual components.

In our project we will be having around 5 individual components namely; homepage, login, signup, booking and user profile. The decomposition of the problem statement into small modules helps us to debug, reduce complexity and increase reusability of code.

OOPS concepts implemented:

4) Polymorphism:

Where an object takes more than one form.

For example: For each button clicked, we have ActionListener() function which will can either open a dialogue box or another application window, etc.

Thank You