**CHARABANC- BUS RESERVATION SYSTEM**

**Submitted By**: Ashutosh Bang - 160001011

Sandarbh Sahu – 160001053

**Project Guide** : Mrs. Aruna Tiwari

**Under the assistance of** : Mr. Nihar Ranjan Panda

**Dated**: 27-09-2017

**INTRODUCTION**

Bus in India is one of the most important modes of transportation. It is one of the world's largest Bus networks. More than sixty percent of the tickets are booked online through various bus reservation websites. In this project we present the Bus reservation model on a small scale. The main purpose behind the implementation of the Bus reservation is to enhance the query processing and the user experience of Bus reservation.

**OBJECTIVE**

In this project, we aim to make a user-friendly website covering the following parts : -

* Registration of new user
* Logging in of existing user
* Query processing for existence of required buses and their seat availability
* Fare estimation
* Ticket generation
* Query processing to find out the schedule of any Bus
* Re-direction to payment site

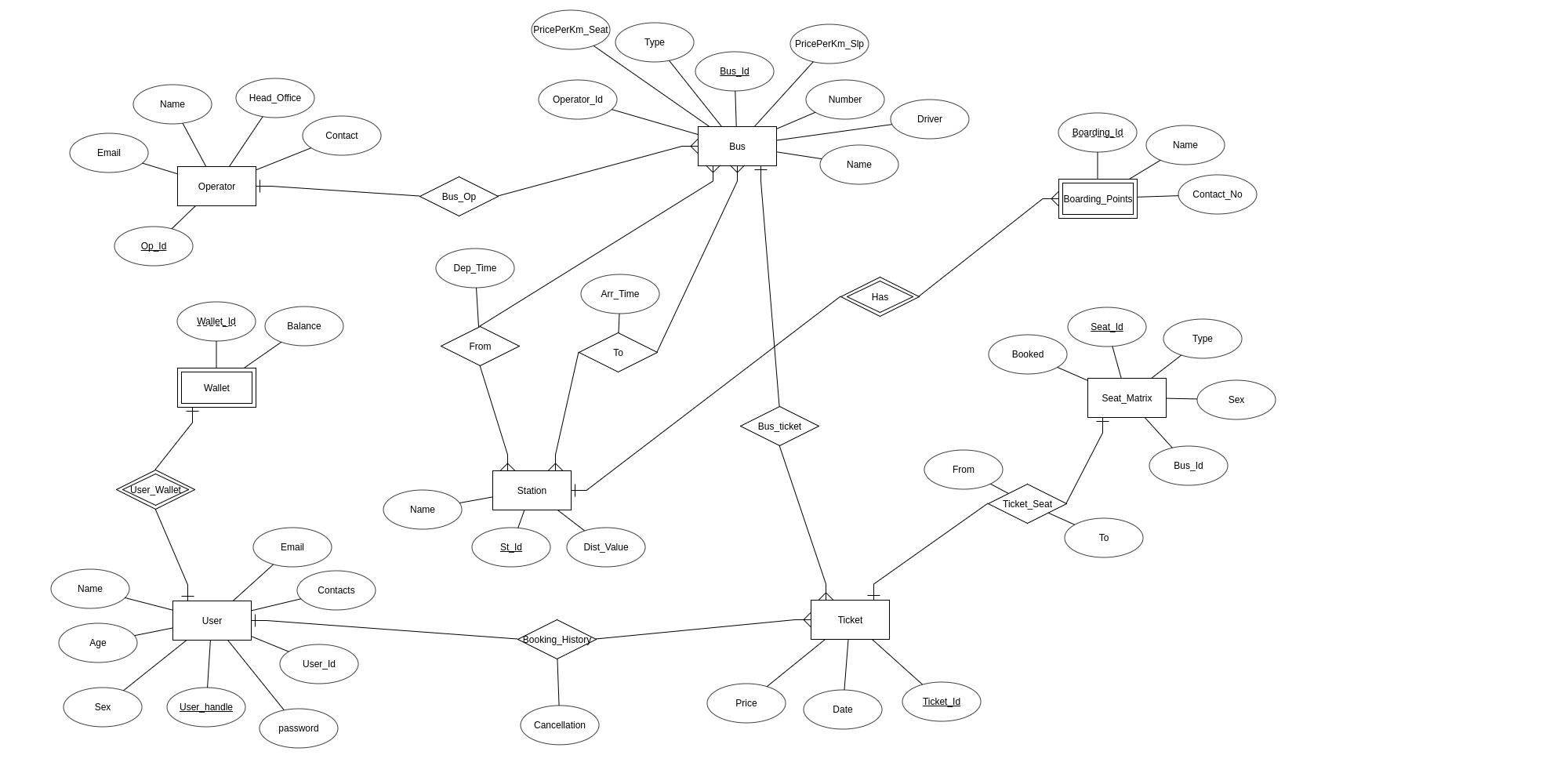
**ER DIAGRAM FOR THE PROBLEM**

**ENTITY SET**

* User
* Operator
* Station
* Bus
* Ticket
* Seat\_Matrix
* Wallet : Weak-Entity set dependent on User
* Boarding\_Points : Weak-Entity set dependent on Station

**ENTITY-RELATIONSHIPS**

* Bus\_Op (Bus-Operator)
* User\_Wallet (Wallet-User)
* Booking\_History (User-Ticket)
* Ticket\_Seat (Ticket-Seat\_Matrix)
* Bus\_Ticket (Bus-Ticket)
* Has (Station-Boarding\_Points)
* From (Bus-Station)
* To (Bus-Station)



**IDENTIFYING CONSTRAINTS AND AUTHENTICITY ISSUES**

**USER**

1.UNIQUE USER\_ID - PRIMARY KEY.

2.Email id ending in @gmail.com.

3.Password must be of 8 characters.

4.Age should be above 18 years.

**TICKET**

1.Auto generated ticket id must be unique.

2.No. of seats booked per ticket can be atmost one.

3.Payment must be complete.

**BUS**

1.Bus no. must be unique.

2.Reserved seats must be at most 20 for any type of seats.

**FUNCTIONAL DEPENDENCIES:**

* User(user\_handle) -> (Password, Name, Gender, Age, Email\_id, Contact Number)
* Operator (Op\_Id) -> (Name, Email, Contact, Head\_Office)
* Station (Stn\_Id) -> (Station\_name, Distance\_Value)
* Bus (Bus\_Id) -> ( Name, Type, Priceperkm\_Seat, Priceperkm\_Slp, Operator\_Id, Bus\_Number, Driver)
* Ticket (Ticket\_Id) -> (Date, Price)
* Seat\_Matrix (Seat\_Id) -> (Type, Sex, Booked)
* Wallet (Wallet\_Id,user\_handle) -> (Balance)
* Boarding\_Points (Boarding\_Id, Stn\_Id) -> (Name, Contact\_No)

**TABLES**

**User** (user\_handle, Password, Name, User\_Id, Gender, Age, Email\_id, Contact Number)

**Station** (Stn\_Id, Station\_name, Distance\_Value)

**Bus** (Bus\_Id, Name, Type, Priceperkm\_Seat, Priceperkm\_Slp, Operator\_Id, Bus\_Number, Driver)

**Seat\_Matrix** (Seat\_Id, Bus\_Id, Type, Sex, Booked)

*Foreign key(Bus\_Id) references Bus.Bus\_Id*

**Ticket** (Ticket\_Id, Date, Price)

**Operator** (Op\_Id, Name, Email, Contact, Head\_Office)

**Wallet** (Wallet\_Id, Balance)

**Boarding\_Points** (Boarding\_Id, Name, Contact\_No)

**Ticket\_Seat** (Ticket\_Id, From, To,Seat\_Id)

*Foreign key(Ticket\_Id) references Ticket.Ticket\_Id*

**User\_Wallet** (user, Wallet\_Id)

*Foreign key(user) references User.user\_handle*

**Bus\_Op** (Id, Op\_Id)

*Foreign key(Id) references Bus.Bus\_Id*

*Foreign key(Op\_Id) references Operator.Op\_Id*

**From** (Stn\_Id, Bus\_Id, Dep\_Time)

*Foreign key(Stn\_Id) references Station.Stn\_Id*

*Foreign key(Bus\_Id) references Bus.Bus\_Id*

**To** (Stn\_Id, Bus\_Id, Arr\_Time)

*Foreign key(Stn\_Id) references Station.Stn\_Id*

*Foreign key(Bus\_Id) references Bus.Bus\_Id*

**Booking\_History** (Ticket\_Id, user, Cancellation)

*Foreign key(Ticket\_Id) references Ticket.Ticket\_Id*

*Foreign key(user) references User.user\_handle*

**Has**  (Stn\_Id, Boarding\_Id)

*Foreign key(Stn\_Id) references Station.Stn\_Id*

*Foreign key(Boarding\_Id) references Boarding\_Points.Boarding\_Id*

**Bus\_Ticket** (Ticket\_Id, Bus\_Id)

*Foreign key(Ticket\_Id) references Ticket.Ticket\_Id*