

## **CASE STUDY PROBLEM DEFINITION**

### **GROUP 2: RAJA MUTHU, SOWMYA SHREE NAGARAJU**

#### **I. Introduction:**

AirCare is a small airline company started in 1988 to facilitate domestic travel inside India. The airline was designed to handle small set of customers and today they are thinking to expand their business and facilitate thousands of passengers on a daily basis. AirCare still uses manual ticket booking system which allows only agents to book the flight tickets as in when the customers call to the agency. Customer data and flight details are kept in files and company uses inefficient system which slows down while using. Now the company wishes to transform to modern technology with flexible reservation and inventory management solutions including call Centre, travel agency, internet engine, global distribution systems.

#### **II. Problem Statement:**

The current system is manual, this system is slow, time consuming and it is very difficult for each person to book through office agents. Users inquire about the tickets through phones and it is very difficult for the user to remember all the details that they received through phones. It is very difficult to calculate how many peoples registered and how many seats on a particular plane are vacant. This requires quite a lot of time and wastage of money as it requires quite lot of man power to do.

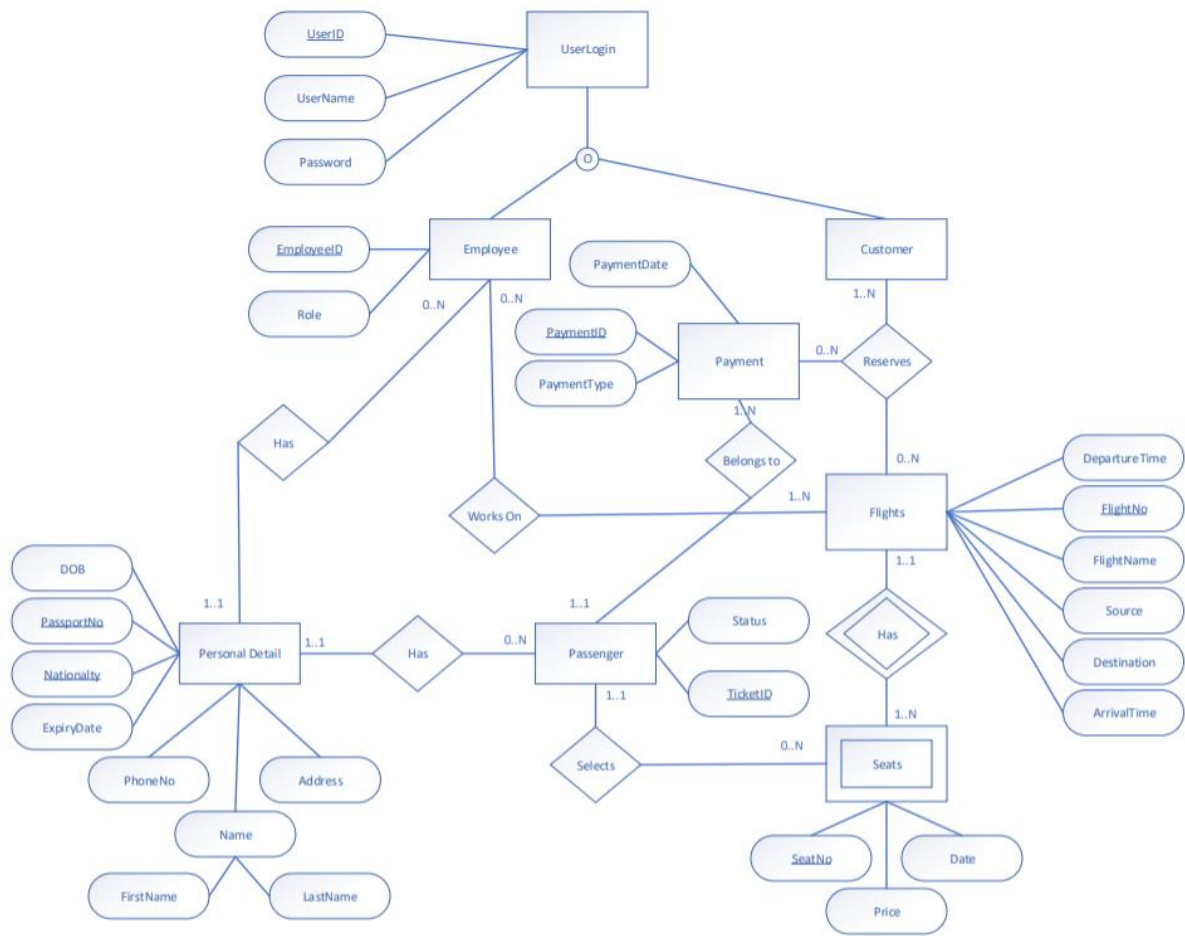
Hence AirCare wants to have a centralized database system for their data which can be accessed easily anytime, anywhere. In this case study we are focusing on building the Relational Database for AirCare.

#### **III. Conceptual Data Modeling:**

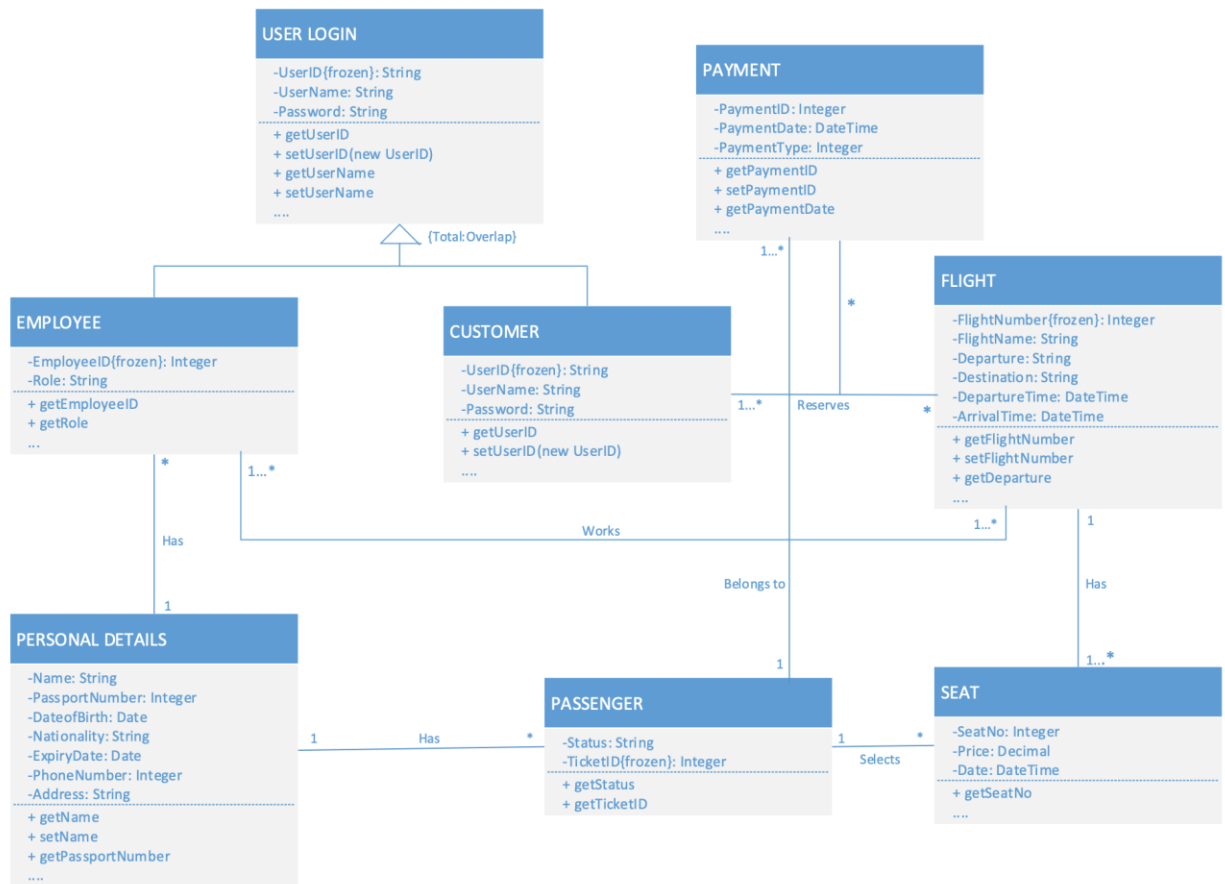
##### **1. EER Diagram**

Some of the Additional conditions which couldn't be mapped in EER model are as follow

- A customer can reserve maximum 6 seats in one booking
- A flight will have minimum of 6 employees
- An employee works in the flight cannot be a customer at the same time



## 2. UML Diagram



## IV. Mapping Conceptual Model to Relational Model

To avoid using Passport no and nationality multiple times we introduce a new unique reference for each record in that relation

**USER**(USERID, USERNAME, USERPASSWORD)

**FLIGHTS**( FLIGHTNO, FLIGHTNAME, SOURCE, DESTINATION, DEPARTURETIME, ARRIVALTIME, *EMPLOYEEID*, SEATS(*DATE*, SEATNO(1.60), PRICE,STATUS), PASSENGER(*TICKETID*, *REFERENCEID*))

**RESERVATION**(*USERID*, PAYMENT(PAYMENTID, PAYMENTDATE,PAYMENTTYPE, AMOUNT, AMOUNT,NO\_OF\_TICKETS,ACCOUNT\_NO, TICKETID1, TICKETID2, TICKETID3, TICKETID4, TICKETID5, TICKETID6))

**PERSONALDETAILS**(REFERENCEID, PASSPORTNO, NATIONALITY, EXPIRYDATE, NAME(FIRST, LAST), DOB, PHONENO, EMAIL, ADDRESS)

**EMPLOYEES**(EMPLOYEEID, ROLE, USERID, *REFERENCEID*)

### **FIRST NORMAL FORM**

By applying first normal form to the above relation we can eliminate the composite and multi-valued attribute types, we get

**USER**(USERID, USERNAME, USERPASSWORD)

**FLIGHTCATALOGUE**( FLIGHTNO, FLIGHTNAME, SOURCE, DESTINATION, DEPARTURETIME, ARRIVALTIME)

**FLIGHTSCHEDULE** ( FLIGHTNO, DATE, SEATNO(1.120), PRICE)

**PASSENGERS** (TICKETID, REFERENCEID, *SEATNO(1.120)*, STATUS)

**RESERVATION**(USERID, PAYMENTID, PAYMENTDATE, PAYMENTTYPE, AMOUNT, AMOUNT, NO\_OF\_TICKETS, ACCOUNT\_NO, TICKETID1, TICKETID2, TICKETID3, TICKETID4, TICKETID5, TICKETID6)

**PERSONALDETAILS**(REFERENCEID, PASSPORTNO, NATIONALITY, EXPIRYDATE, FIRSTNAME, LASTNAME, DOB, PHONENO, EMAIL, ADDRESS)

**EMPLOYEES**(EMPLOYEEID, ROLE, USERID, REFERENCEID)

### **SECOND NORMAL FORM**

By applying second normal form, we can map the attributes with functional relationship.

**USER**(USERID, USERNAME, USERPASSWORD)

**FLIGHTS**( FLIGHTNO, FLIGHTNAME, SOURCE, DESTINATION, DEPARTURETIME, ARRIVALTIME)

**PERSONALDETAILS**( REFERENCEID, PASSPORTNO, NATIONALITY, EXPIRYDATE, FIRSTNAME, LASTNAME, DOB, PHONENO, EMAIL, ADDRESS)

**EMPLOYEES**(EMPLOYEEID, ROLE, *USERID*, *REFERENCEID*)

**PAYMENT**(PAYMENTID, PAYMENTDATE, PAYMENTTYPE,  
AMOUNT, NO\_OF\_TICKETS, ACCOUNT\_NO)

**FLIGHTSCHEDULE** ( FLIGHTNO, DATE, SEATNO(1.120), PRICE)

**PASSENGERS**(*REFERENCEID*, TICKETID, *SEATNO(1.120)*, STATUS)

**RESERVATIONS**(*USERID*, PAYMENTID, TICKETID1, TICKETID2, TICKETID3, TICKETID4, TICKETID5,  
TICKETID6)

The Above relations are in third normal form as well, so further normalization is not required.