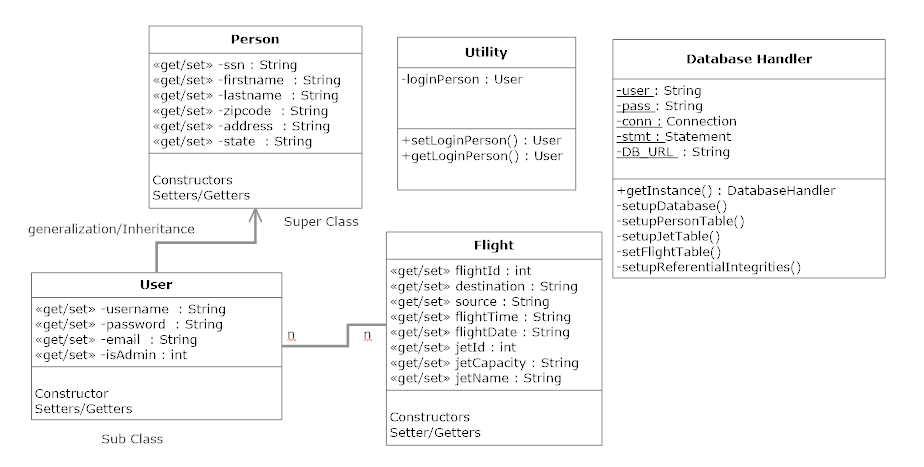
AIRLINE MANAGEMENT SYSTEM  
 (Documentation)

Requirements:

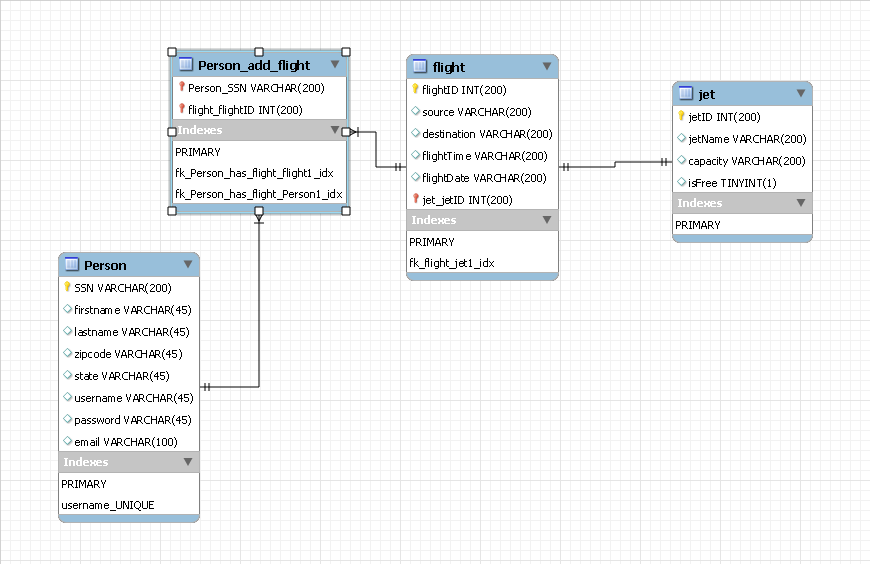
The code is written in Java language and at least in four different classes. At least 3 concepts of encapsulation, inheritance, polymorphism or composition is utilized. The code has proper comments where needed. Application does not crash and exception handling is implemented where needed. Database programming is done using SQL server.

# Class UML Diagram:

* <<get/set>> shows that attributes have setters/getters
* n to n shows that both user and flight have many to many relationship i.e One User can add many flighst and one flight can be added by many users.
* Underline shows the attributes are static.
* - shows private attributes
* + shows public attributes



# ERD Diagram:



* **Table Person** 
  + **PK : SSN**
  + **UNIQUE : username**
* **Table Flight**
  + **PK : flightID**
  + **FK : jet\_jetID**
* **Table Jet**
  + **PK : jetID**

# Relationships:

* **Person (many – to – many ) Flight**
  + A person can add many flights and one flight can be added by many persons
* **Flight (many – to – one ) Jet**
  + A jet can have only many numbers of flights at a time and one flight can have only one jet.

Functional Requirements:

1. When the application starts, the user sees a splash screen with login
2. Admin login username and password is “admin” and “admin”
3. The Main menu gives the option to user to register or login
4. Login requires username and password
5. Registration requires personal information such as first name, last name, username, password, email, SSn
6. User can login using username and password
7. The admin has the privilege to add flight, delete, search, dispay flights.
8. After logging in, customer is able to search, book, delete flights
9. Customers are not able to book same flight more than once
10. Admin is able to do all customer activities
11. User is able to logout and login again