**Documentation Final Term Project CIS\_3270**

**Kristin Erdmann**

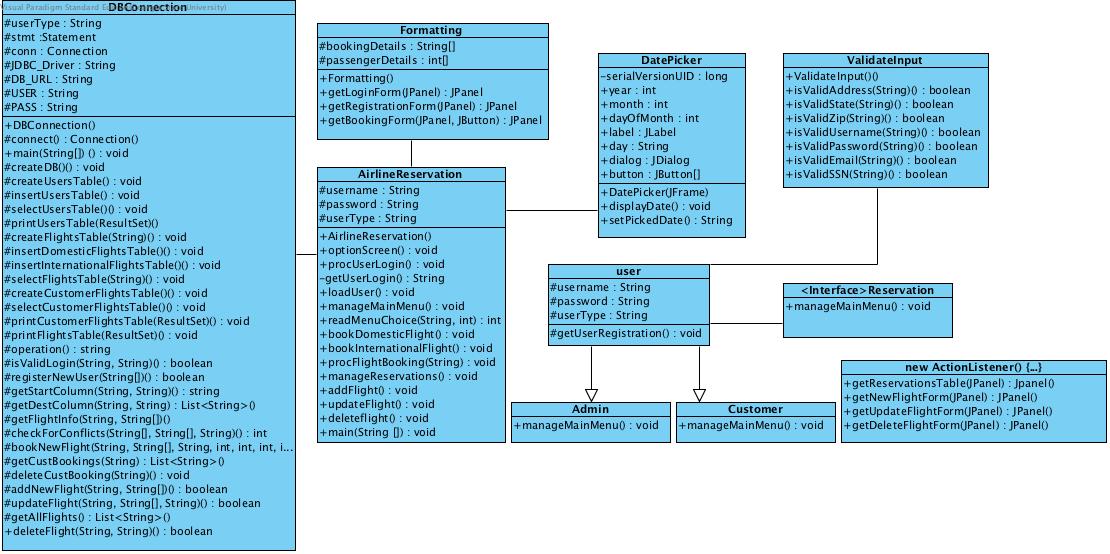
**Ruwel Sarmad**

**Requirements and Specifications:**

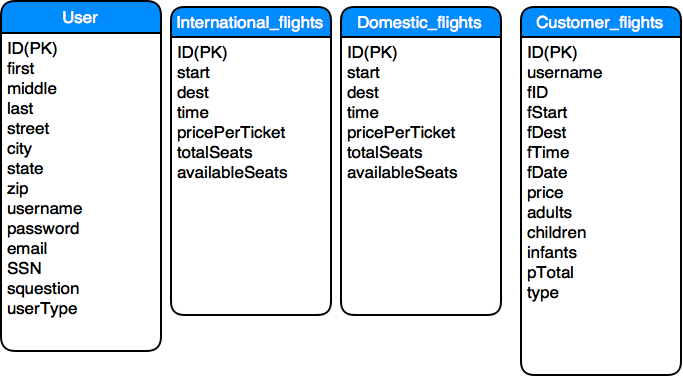
* Create an Airline flight reservation system in which the users are able to register their accounts and start using it for booking flights.
* The System should be fully functional when it comes to booking a flight. There would be two user types for the entire system; the customers who would book the flight and the administrators who would actually manage the reservations and all the flight information. The administrator should have more privileges than the customer.
* To get started the customer should be able to create an account by registering first name, last name, address, zip, state, username, password, email, SSN and a security question for password recovery. After the customer has registered, he/she would be able to search for flights and book those and make any changes by deleting them. Also the customer shouldn’t be able to book a flight more than once (We did that by making sure, that is the customer has one flight on a day, then he should not be able to book a second flight because the departing destination by default would be Atlanta, Georgia). If there is a conflict about date and time of a flight, application should warn the customer about this and should not let user book the conflicting flight. The application should keep track of number of passengers booked at each flight and should not let a user book the flight if it is full. An admin should be able to do all customer activities in addition to be able to add, update or delete a flights. The customer should be able to logout and upon login; previously booked reservation should be there in his/her account. There needs to be a way to navigate to the Main menu.

**Class diagrams UML**

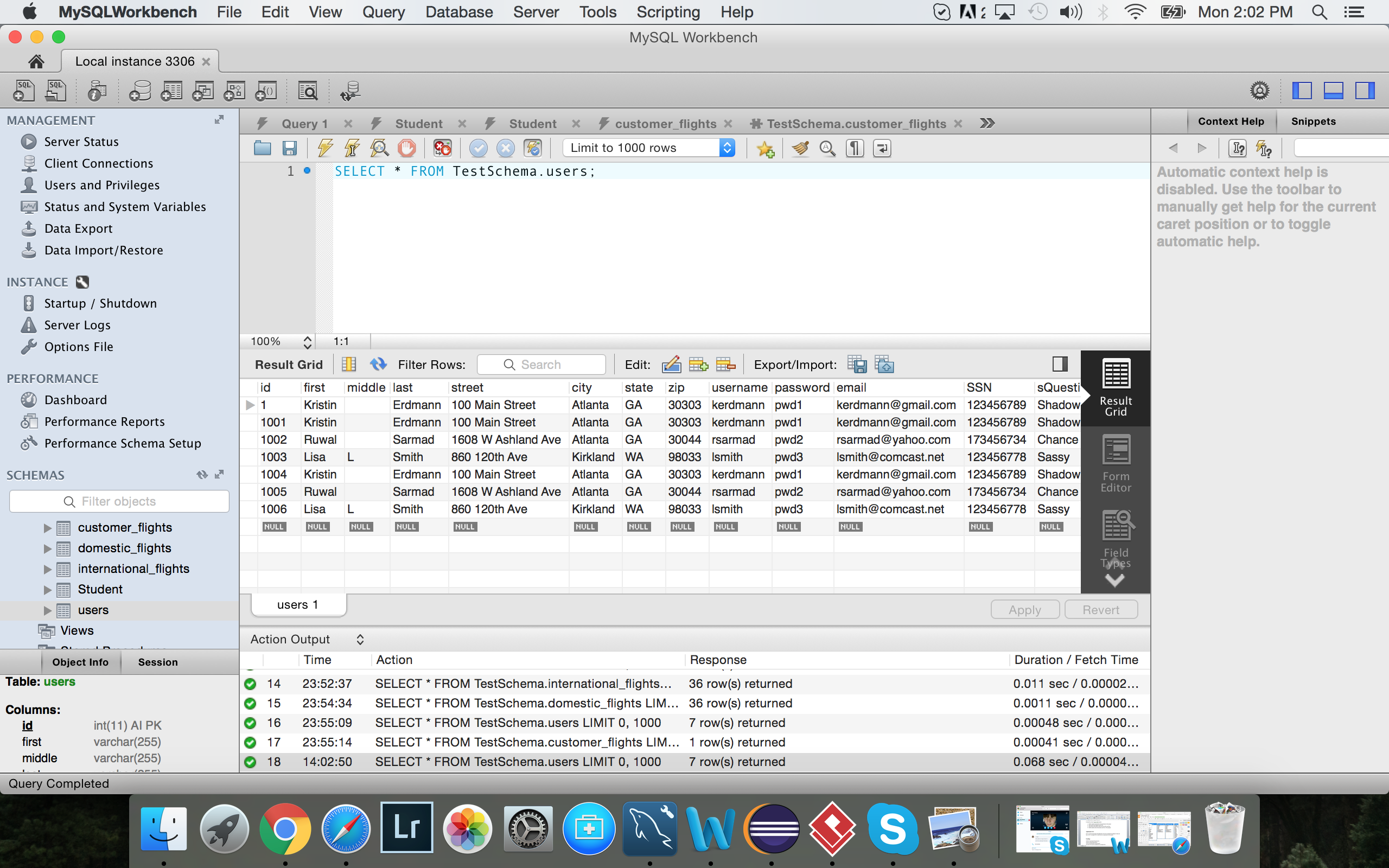
* Class diagrams for the program



**Database tables**

****

**Example of one of the user table:**

****

**Functionality and flow of the program**

* Upon running the program, the user is able to see the splash screen, which is when the application starts to work
* The user sees a screen with two options; login and register. If the user already exists, then he/she can simply login into the account, otherwise they can always register.
* There are two types of users for this application. The customer and the admin. The admin has additional privileges of updating, deleting a flight.
* When the user logs in, he/she would be able to see a path from which they can book a flight. It would ask for destination and time details and would produce the results; tickets and available seats for the flight. The user can book the flight after going through those screens. The DatePicker class makes up the calendar for the program.
* The Airline reservation system class basically does everything in regards to booking a flight for the user.
* The DBConnection class links up to the database in MYSQL which stores all the information in all the four tables that we have; user, International\_flights, Domestic\_flights and customer\_flights(reservations)
* The ‘ValidateInput’ class validates everything as far as the login/registration process is concerned. It would tell a user if the account exists or not and would give an error if somebody tries to log in with and username/password that doesn’t exist. It would also give errors to user if they don’t follow the format of the password to be selected when it comes to registering. This screen would also validate existing user, and let them in the system if they have a valid username and password that is already registered with the system.
* The formatting class handles the GUI
* Once the user books a flight, everything stores into the database and can be viewed there. Also since the admin has additional privileges, they can make changes to the existing data as we mentioned before
* This system could be extended in the future if we want to add options like book a hotel or rent a car