Login Page Analysis

**Use Case1:**

* The use case will start when the user requests a login
* The user should be able to hit a login button that will prompt him to enter his information and successfully login
* The system should provide the ability to the user to enter his information in only two fields, Email and password.
* The system should take the input and make sure it matches the data in the database for this specific email.
* The system should return a successful log in if it exists and the user case will end here.

**Display Responsibilities:**

The display team should provide a button that the user will hit to log in, then the fields should show up. Only two fields(Email Address and Password).Please provide a naming convention for these fields and the button to make the job easier. Such as emailAddressField, PasswordField…

**DataBase Responsibilities:**

The database should create the following

* 1. Create a Table in SQL that has ID, EmailAddress, Password, RestaurantName, PhoneNumber columns. Please make sure that the EmailAddress and ID are unique data.
  2. Should create a main class Named ReserveDTB. This is the class the the database team will be maintained during the whole project. Create a method Called **Boolean CompareInfo(string emailAddress, string Password)**. This method should go through all the rows in the table and check if the email address and password match any existed restaurants. If it does, return yes, else return no.

**Middleware Responsibilities:**

the middleware team should create a main class Called ReserveMLW. This class will be maintained from the team during the whole project.

The class fetch the email address and password from display and save them into variables

The class should contain a method RestaurantLogIn(). This method will call  **CompareInfo(string emailAddress, string Password)** from the database class. If the value return is true. The login is successful, else prompt the user to reenter his information.

**Use Case2:**

* The second use case is when a new restaurant wants to register to the site.
* The use case will start when the restaurant request a sign up action.
* The system should give the restaurant the ability to enter its data. Then the system should take these data and compare them with the data in the database. If the email doesn’t exists in the data, the registration is successful and the use case will end here

1. The restaurant should see a signup button or a Hyperlink that will prompt them to enter information. These information include. Email address, the name of the restaurant, location, phone number, and Password.
2. The restaurant should see a button “Register” that’s only available when the information are filled correctly

**Display Responsibilities:**

The display team should use HTML code to show the signup button. When the button is clicked, the fields and a register button should show up.

Please provide a naming convention for these fields and the button to make the job easier. Such as emailAddressField, registerBtn….

I think only the phone number field need to have a specific data type which is integers. It doesn’t have to have a format to keep it simple

**DataBase Responsibilities:**

1. The DataBase team should create the following
   1. The main class should contain the following functions:
      1. **Boolean CheckData(string emailAddress)**. This function will take the email address that was received from the user input and will check if the email address already exist in the database. It will return true if exists, false if not.
      2. **SaveData(string emailAddress, string password, int phoneNumber, string location, string restaurantName)**. This function will save the data in the database and it will generate a unique ID to the new restaurant.

**Middleware Responsibilities:**

1. The main class should contain the following:
   1. The class should initialize variables and fetch these variables from the fields in the display.
   2. **IsValid(string EmailAddress).** This function should call the **CheckData** function in database to make sure the email address doesn’t exist. If it does show an error message to the user to reenter the information again
   3. **RegisterRestaurant()**. This function should check the validity of the information using  **IsValid**  method. If it doesn’t exist the function should call SaveData method and pass the right parameters to save the information.