**Applying AJAX**

1. When a waiter is checking for chef’s email address. On "btnShowEmail\_Click" Event instead of reloading the whole page the label lblEmail is AJAXified by putting it inside the Content Template tags of the Update Panel and setting the trigger to be clicking the btnShowEmail.

**(UC008B) Show Chef Email Address**

Main Success Scenario:

1. Waiter selects chef from ListBox and clicks show email Button.
2. System retrieves the email address from the selected item in the ListBox.
3. System outputs the email address in a Label.

Extensions:

1a: No item selected in ListBox.

.1: Output appropriate error message.

1. On the UpdatePassword page the system shows a message regarding the strength of the new password. lblPassword is AJAXified by putting it inside the Content Template tags of the Update Panel and setting the trigger to be txtNewPassword\_TextChanged.

AutoPostBack property of txtNewPassword is set to True - a post back will occur when it loses focus.

1. On the WaiterRegistration page the system shows a message regarding the strength of the password. lblPasswordStren is AJAXified by putting it inside the Content Template tags of the Update Panel and setting the trigger to be txtPassword\_TextChanged.

AutoPostBack property of txtPassword is set to True - a post back will occur when it loses focus.

**Secure password storage**

References:

<https://www.janaks.com.np/password-hashing-csharp/>

<https://msdn.microsoft.com/en-us/library/3ffysz34(v=vs.110).aspx>

<http://stackoverflow.com/questions/212510/what-is-the-easiest-way-to-encrypt-a-password-when-i-save-it-to-the-registry>

**2.1.1. Use Case(s)**

**(UC001) Login User**

Main Success Scenario:

1. System loads the page.
2. User enters their username and password and then clicks login Button.
3. System validates input fields.
4. System queries database to authenticate the username and password.
5. System queries database to authenticate the provided password match the recorded hashed password for that user.
6. System creates Session objects for UserID, UserName, RestaurantID and RoleID.
7. User is redirected to UserAccount.aspx.

Extensions:

3a: Entered username length is less than 5 or greater than 20 characters.

.1: Output appropriate error message.

3b: Entered password length is less than 6 characters .

.1: Output appropriate error message.

4a: Database query returns no rows - details must be incorrect.

.1: Output appropriate error message.

4b: Catch database error.

.1: Output appropriate error message.

**(UC002B) Register Waiter**

Main Success Scenario:

1. Waiter enters a username, password, password confirmation, full name, email address, selects a restaurant and then clicks register Button.
2. System validates input fields.
3. System queries database to ensure username does not already exist.
4. System generates salt and is hashing and salting the password for that user.
5. System inserts new waiter user into the database.
6. Waiter is redirected to UserLogin.aspx.

Extensions:

2a: Entered username length is less than 5 or greater than 20 characters.

.1: Output appropriate error message.

2b: Entered password length is less than 6 characters.

.1: Output appropriate error message.

2c: Entered password confirmation not equal to password input.

.1: Output appropriate error message.

2d: Entered real name field is empty.

.1: Output appropriate error message.

2e: Entered email field is empty or the email is not in a legitimate format.

.1: Output appropriate error message.

3a: Database query returns a record - username must already exist.

.1: Output appropriate error message.

4a: Catch database error.

.1: Output appropriate error message.

***2.5. UpdatePassword.aspx***

**2.5.1. Use Case(s)**

**(UC005) Update User Password**

Main Success Scenario:

1. System ensures a user is logged in (i.e. a Session exists).
2. System loads the page.
3. User enters current password, new password and new password confirmation then clicks update password Button.
4. System validates input fields.
5. System queries database to get the salt saved for that user. Adds it to the provided password and hashes it and checks if its equal to the saved password in the database.
6. If the provided password is the same system generates new salt for the user, creates hash for the new password and saves the hashed and salted password and the salt string in the database.
7. System updates user's password in database.
8. User is redirected to UserAccount.aspx?UpdateSuccess=Password.
9. System processes query string and shows a success label message.

Extensions:

1a: User not logged in.

.1: Redirect user to UserLogin.aspx.

4a: Entered current password length is less than 6 characters.

.1: Output appropriate error message.

4b: Entered new password length is less than 6 characters.

.1: Output appropriate error message.

4c: Entered new password confirmation not equal to new password input.

.1: Output appropriate error message.

5a: Inputted current password does not match that in database for this user.

.1: Output appropriate error message.

7a: Catch database error.

.1: Output appropriate error message.

**Additional use cases**

Adding two radio buttons in and changing WaiterRegistration to UserRegistration, in order to be able to register Chefs as well. The user must select their role when registering.

**(UC002B) Register Waiter**

Main Success Scenario:

1. Waiter enters a username, password, password confirmation, full name, email address, selects a restaurant and then clicks register Button.
2. System validates input fields.
3. System queries database to ensure username does not already exist.
4. System inserts new waiter user into the database.
5. Waiter is redirected to UserLogin.aspx.

Extensions:

2a: Entered username length is less than 5 or greater than 20 characters.

.1: Output appropriate error message.

2b: Entered password length is less than 6 characters.

.1: Output appropriate error message.

2c: Entered password confirmation not equal to password input.

.1: Output appropriate error message.

2d: Entered real name field is empty.

.1: Output appropriate error message.

2e: Entered email field is empty or the email is not in a legitimate format.

.1: Output appropriate error message.

3a: Database query returns a record - username must already exist.

.1: Output appropriate error message.

4a: Catch database error.

.1: Output appropriate error message.