Pass Task 8.2 Securing your enterprise application

Task 1: Case study analysis

a) The actual change of information should occurred at the data access layer. As this layer is responsible for storing & retrieving data, it should contain all logics related to data manipulation such as creating, changing & delete data record. This practice will allow business layer to focus on business logic, validations and rules.

b) It is always a good practice to handle sensitive data carefully in a secure manner. Therefore, in this case of excluding the password from the DTO to send to client, it protect the password from being exposed and prevent potential risk of miss-use this sensitive data. For the purpose of displaying employee’s details, instead of using the real password for displaying, the company can use a fake text input with a number of dots (or asterisks) to illustrate that the password field is holding a password value.

c) In the context of changing employee’s record, with the discussion in (b), it will be the best choice not to send the existing password for displaying before update. Alternatively, it will be acceptable to not display this password input field at all and instead using a “Change Password” button for the event that user need to change their password. Once user click to this button, a dedicated form for changing password will display. This form can contains 3 input fields: old password or security question & answer, new password & confirm new password. Once this information is sending to business logic server, the first field will be used along with userID to identify if the user is who he claims for, the next 2 fields are compared together and if they are match, then the password will be changed.

d) In the context of deleting employee’s record, it is a best practice to use an “active” flag to indicate if the user record is valid to use across the application. This is to prevent deletion of any existing data that has relation to this user and allow the database to be auditable. It is also allow a mistake tolerant where accidental deletion may occurred. However, this behaviors should be align with the actual context of the company’s business and how it manages employee’s data as the best practice is not always be the correct fit for any application.

Design justification:

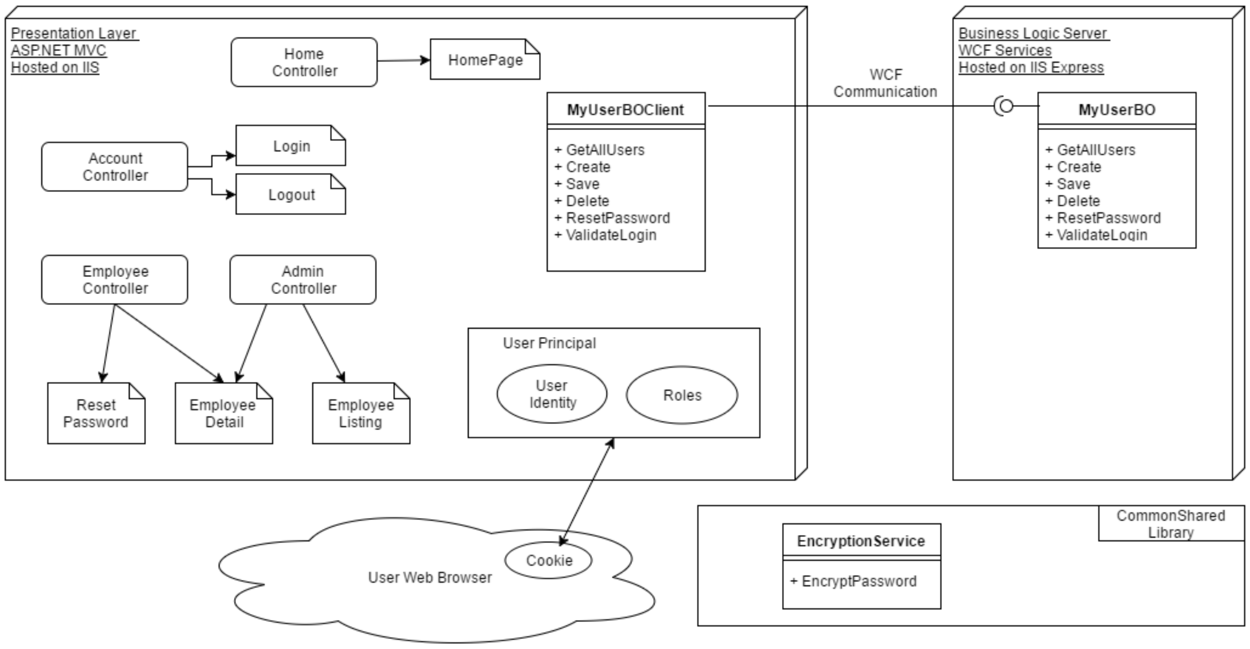


Figure 1. Architecture diagram.

Business Logic Server: In this implementation, the business logic object MyUserBO contains a method to validate user login that accepts 2 parameters: UserName and Password. The Password parameter is accepted in encrypted format of an array of byte. The encryption is done by using the EncryptionService on the CommonShared project. The logic is validate user is done on this layer: once this business object receives ValidateLogin request from a client, it will use the UserName to find a matching user in the database, then compute a hash from the Password field of the found user. If the 2 hashes are matching, then the business object will response to the client with a UserDto object that contains user information. It is important to note that, this dto object doesn't contain any sensitive data such as Password or Security Question or Answer.

Presentation Layer: This layer has an enhancement on the Security classes: It uses FormsAuthenticationProvider for the Web Security layer that is to identify the user by a login form input with a pair of username & password. Due to a fact that our data model architecture is using Business Logic Layer to validate users and control the access to database, the presentation layer doesn't have any knowledge about a database access layer, it is required to build a custom authentication strategy to fit to the 3-tier design. This layer contains a number of Controllers as follow:

- HomeController : present a landing page when user reach the main homepage.
- AccountController: present a Login form and perform LogIn & LogOut process
- EmployeeController: present all functions that are available to Employee role such as: View Profile, Reset Password
- AdminController: present all functions that are available to Administrator role such as: User Listing, Create User, Delete User, Edit User.

Security filter is applied on EmployeeController and AdminController to allow only appropriate users with valid access:

```
[Authorize(Roles = "Administrator|Employee")]
    public class EmployeeController : Controller {……}
[Authorize(Roles ="Administrator")]
    public class AdminController : Controller {……}
```

Authorize attribute will only allow Authenticated user to access. The Roles property specified on this attribute will allow only Authorized user with valid role to access. Hence, from the above configuration, an Employee will not be able to access to AdminController but an Administrator can access to all Employee functions. This makes sense when Administrator can be considered as a normal user and can update his/her own information. The web configuration setting is setup so that when unauthenticated users trying to access any of the above 2 protected controllers, they will be redirect to the Login page:

```
    <authentication mode="Forms">
      <forms loginUrl="~/Account/Login" timeout="2880" />
    </authentication>
```

After the user enter a valid username and password to the Login form, the password will be encrypted using the EncryptionService and the information will be passed on to the BusinessObject (BO) to validate. If the BO returns with a valid user information then a Form Authentication ticket is created and the user information will be serialized to attach to this ticket and stored in client's Browser cookie. Any subsequence requests sending from this client will contain this ticket so that the server can open it and check if the user has been authenticated or not. By doing so, it will save a round trip back to the BusinessLogicServer to retrieve user information for every request, hence greatly improve the performance of the Web Application. In the implement of this design, this is done by using 2 classes: UserPrincial which extends IPrincial and UserIdentity which extends IIdentity interfaces so that they can be used to Integrate with the build-in User Context management of ASP.NET MVC Framework:

```
protected void Application_PostAuthenticateRequest(Object sender, EventArgs e)
        {
            ……
                UserPrincipal newUser = new UserPrincipal(serializeModel);
                HttpContext.Current.User = newUser;
            ……
        }
```

Compare to the using of messaging object in Pass Task 6.2, using a messaging object for resetting password would be inappropriate in this case as the user needs to know the outcome of the password reset process as soon as possible after he triggers the reset button. In this design, the user is required to enter the old password, a new password and a confirmed password so that these information can be sent to the business object to process. If any issue occurred with the data such as wrong old password or the new password is too short, then user must be notified straight away to fix the error. Using a messaging communication strategy in this process will introduce delay and inconvenience. Therefore, it is more appropriate to use business logic service done via WCF in this case.

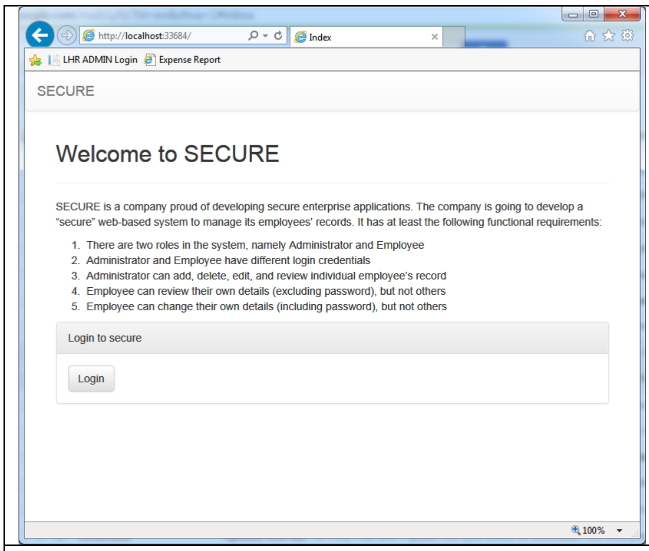
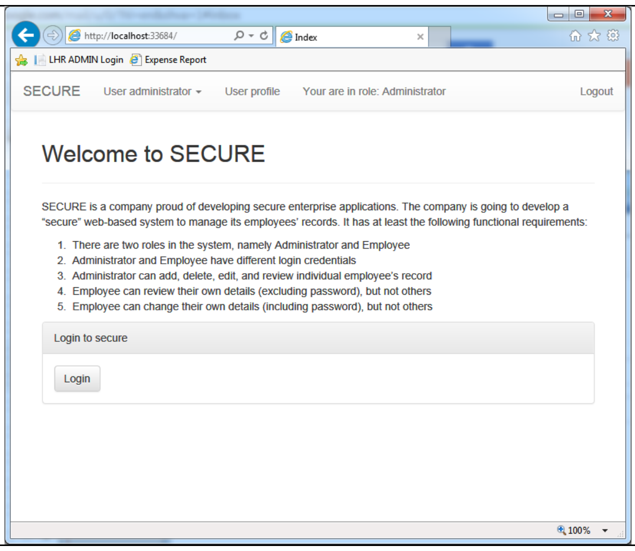
Test cases:
The following test cases have been taken out to make sure that the whole application works as expected:

ID	Test Case Name	Step	Expected Output	Result
01	Access public pages with and without Authentication information	1. View content of Home Page 2. View Login Page 3. Login to the site using any user account 4. View content of Home Page	There should be no restriction on those pages and User can see and navigate freely between these public pages	Pass
02	Access protected pages without Authentication information	1. Try to access to Path: /Admin/Index 2. Try to access to Path: /Employee/Index	The page should be redirected to /Account/Login page	Pass
03	Access protected pages with valid Authentication ticket but invalid role	1. Login to the site using an Employee account 2. Try to access to Path: /Admin/Index	The path cannot be access due to invalid role.	Pass
04	Access protected pages with valid authentication ticket and role	1. Login to the site using an Employee account 2. Try to access to Path: /User/Index	The user profile form should be displayed	Pass
05	Login as admin account	1. Login to the site using an Admin account	The admin page should be able to access with user listing page	Pass
06	Try to reset password	1. Login to the site using an Employee account 2. Navigate to path: /User/Index 3. Click Reset Password button 4. Enter old password and new password information 5. Click Reset password	The password of the existing user should be updated. A success message will be displayed	Pass
07	Try to reset password with wrong detail	1. Repeat from step 1 to step 3 of test ID# 5 above 2. Enter a wrong old password 3. Click reset password	An error message will display along with the reset password form	Pass

Table 1. Test cases and test results

Test case screen shots:

Test 01 – Access public pages with and without Authentication information

	
Access to home page before login	Access to Homepage after login

http://localhost:33684/ Index

LHR ADMIN Login Expense Report

SECURE

Welcome to SECURE

SECURE is a company proud of developing secure enterprise applications. The company is going to develop a "secure" web-based system to manage its employees' records. It has at least the following functional requirements:

1. There are two roles in the system, namely Administrator and Employee
2. Administrator and Employee have different login credentials
3. Administrator can add, delete, edit, and review individual employee's record
4. Employee can review their own details (excluding password), but not others
5. Employee can change their own details (including password), but not others

Login to secure

Login

100%

http://localhost:33684/ Index

LHR ADMIN Login Expense Report

SECURE User administrator User profile Your are in role: Administrator Logout

Welcome to SECURE

SECURE is a company proud of developing secure enterprise applications. The company is going to develop a "secure" web-based system to manage its employees' records. It has at least the following functional requirements:

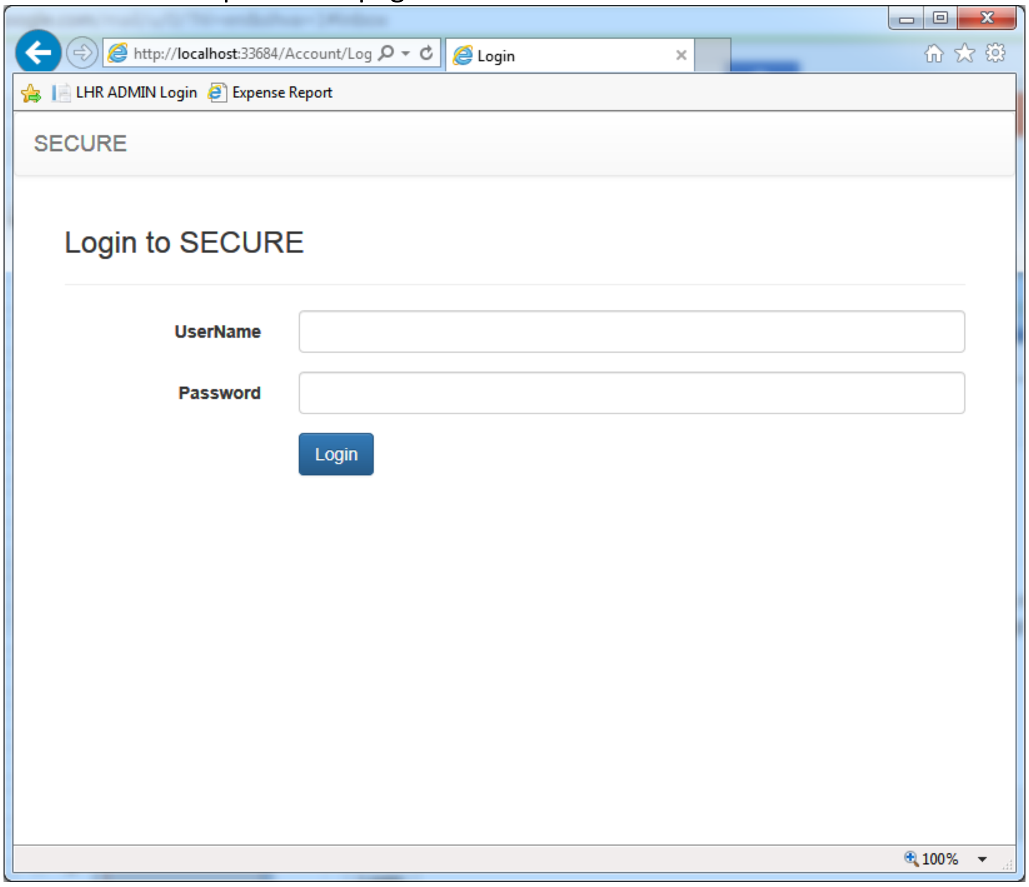
1. There are two roles in the system, namely Administrator and Employee
2. Administrator and Employee have different login credentials
3. Administrator can add, delete, edit, and review individual employee's record
4. Employee can review their own details (excluding password), but not others
5. Employee can change their own details (including password), but not others

Login to secure

Login

100%

Test 02 – Access protected pages without Authentication information



http://localhost:33684/Account/Log Login

LHR ADMIN Login Expense Report

SECURE

Login to SECURE

UserName

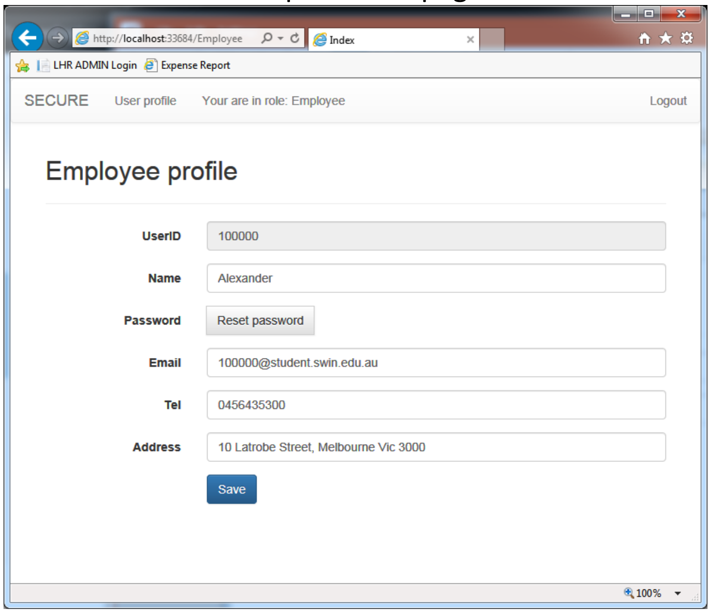
Password

Login

100%

The login page is displayed

Test 03 & 04 - Access protected pages with valid Authentication ticket but invalid role



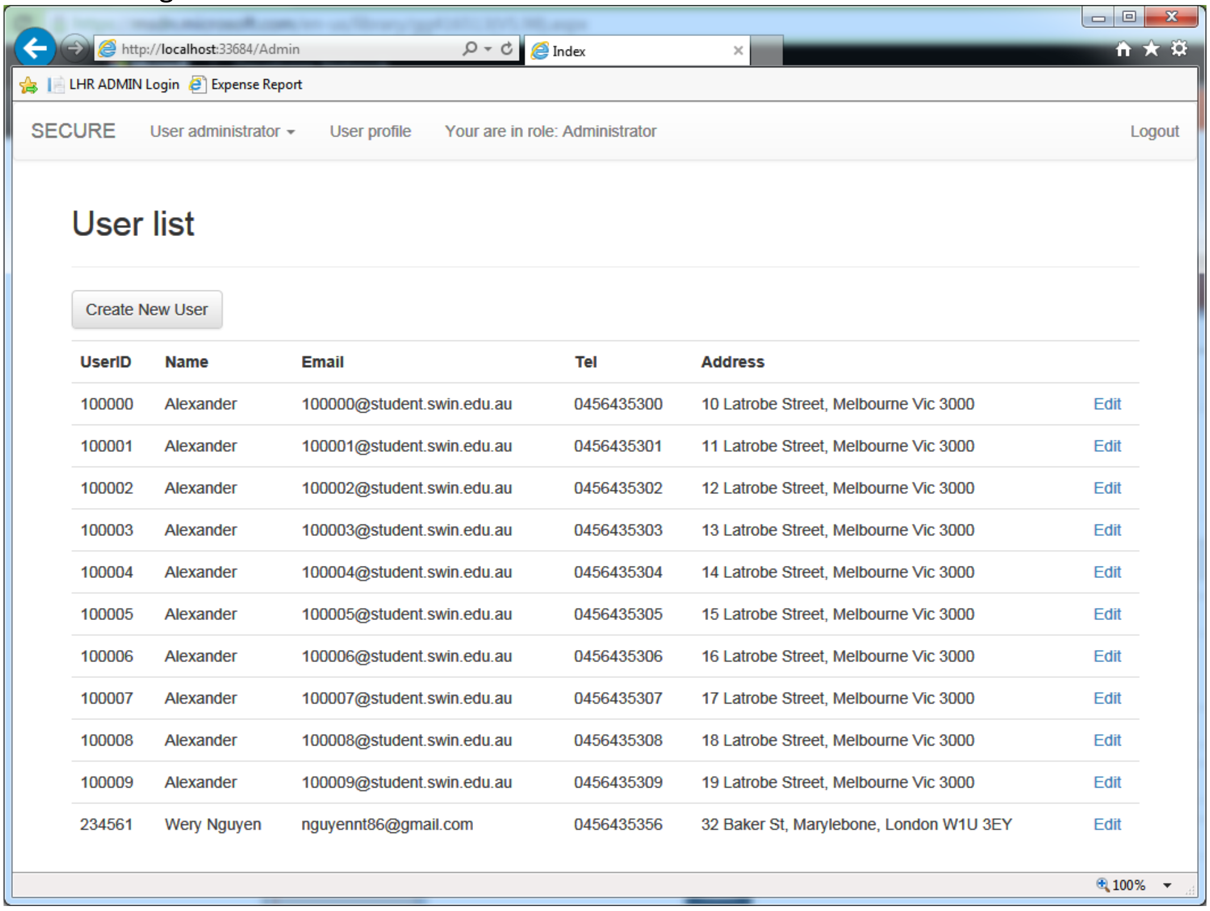
Employee profile

UserID	100000
Name	Alexander
Password	Reset password
Email	100000@student.swin.edu.au
Tel	0456435300
Address	10 Latrobe Street, Melbourne Vic 3000

Save

The page is redirected to LoginPage but with user has already authenticated, it redirect to the landing page of the Employee role.

Test 05 – Login as admin

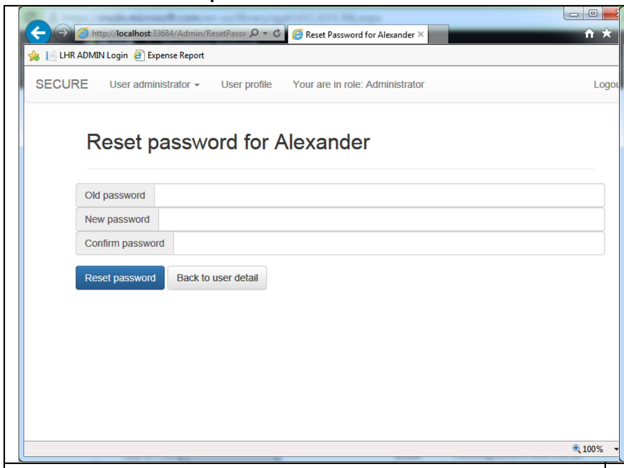
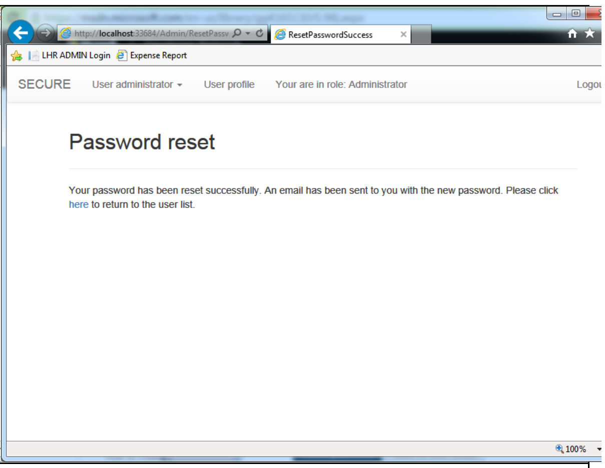


User list

Create New User

UserID	Name	Email	Tel	Address	
100000	Alexander	100000@student.swin.edu.au	0456435300	10 Latrobe Street, Melbourne Vic 3000	Edit
100001	Alexander	100001@student.swin.edu.au	0456435301	11 Latrobe Street, Melbourne Vic 3000	Edit
100002	Alexander	100002@student.swin.edu.au	0456435302	12 Latrobe Street, Melbourne Vic 3000	Edit
100003	Alexander	100003@student.swin.edu.au	0456435303	13 Latrobe Street, Melbourne Vic 3000	Edit
100004	Alexander	100004@student.swin.edu.au	0456435304	14 Latrobe Street, Melbourne Vic 3000	Edit
100005	Alexander	100005@student.swin.edu.au	0456435305	15 Latrobe Street, Melbourne Vic 3000	Edit
100006	Alexander	100006@student.swin.edu.au	0456435306	16 Latrobe Street, Melbourne Vic 3000	Edit
100007	Alexander	100007@student.swin.edu.au	0456435307	17 Latrobe Street, Melbourne Vic 3000	Edit
100008	Alexander	100008@student.swin.edu.au	0456435308	18 Latrobe Street, Melbourne Vic 3000	Edit
100009	Alexander	100009@student.swin.edu.au	0456435309	19 Latrobe Street, Melbourne Vic 3000	Edit
234561	Wery Nguyen	nguyennt86@gmail.com	0456435356	32 Baker St, Marylebone, London W1U 3EY	Edit

Test 06 – Reset password

	
Password reset form	Password reset success

http://localhost:33684/Admin/ResetPassw
Reset Password for Alexander
LHR ADMIN Login Expense Report
SECURE User administrator User profile Your are in role: Administrator Logout

Reset password for Alexander

Old password
New password
Confirm password

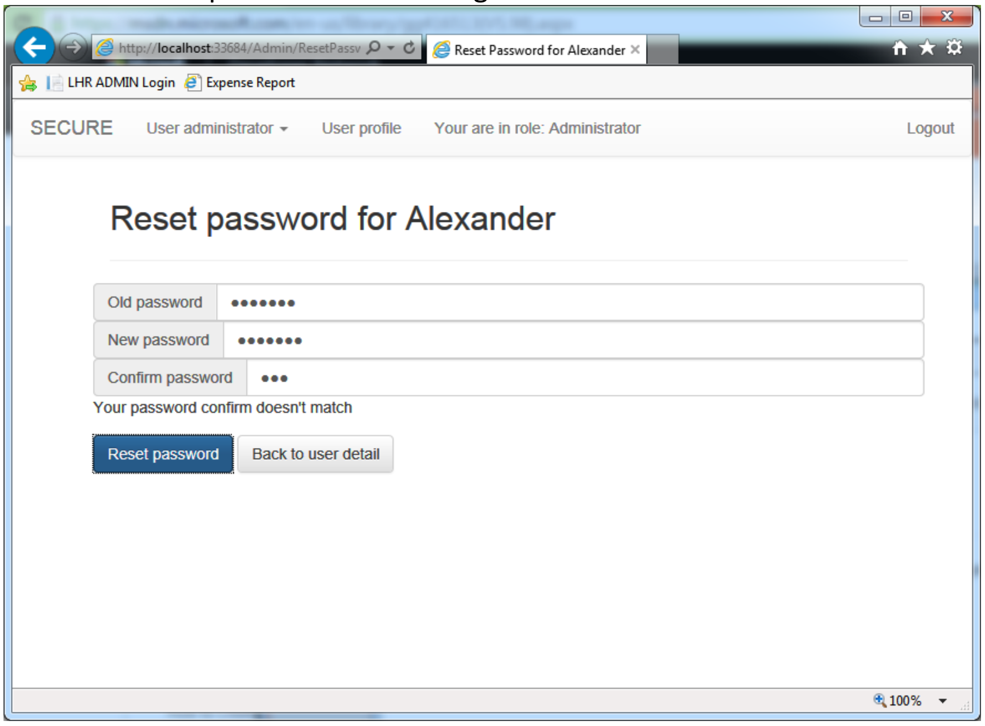
Reset password Back to user detail

http://localhost:33684/Admin/ResetPassw
ResetPasswordSuccess
LHR ADMIN Login Expense Report
SECURE User administrator User profile Your are in role: Administrator Logout

Password reset

Your password has been reset successfully. An email has been sent to you with the new password. Please click here to return to the user list.

Test 07 – Reset password with wrong detail



http://localhost:33684/Admin/ResetPassw
Reset Password for Alexander
LHR ADMIN Login Expense Report
SECURE User administrator User profile Your are in role: Administrator Logout

Reset password for Alexander

Old password •••••••
New password •••••••
Confirm password •••
Your password confirm doesn't match

Reset password Back to user detail

100%

References:

(1) 2016, *Forms Authentication Provider*, Microsoft MSDN, viewed 27th May 2016, < https://msdn.microsoft.com/en-us/library/9wff0kyh.aspx>

(2) BitBucket source code: https://bitbucket.org/werynguyen/swinschool/src/4c090c6cef0ddb9b49a1e5f23c0886d9975ed88b/?at=PT82