Pass Task 8.4 Securing your enterprise application

Task 1: Research and study of role-based authentication & authorization

a) ASP.NET comes with a number of built-in classes that support user authentication & authorization in its core. There are 2 general abstract classes that provide this features:
   - MembershipProvider class define a structure of ASP.NET implementation to support membership account management feature. They have 4 following built-in derived classes:
     - ClientWindowsAuthenticationMembershipProvider: This support Windows OS authentication with client programs.
     - ClientFormsAuthenticationMembershipProvider: Allow forms authentication with client programs.
     - ActiveDirectoryMembershipProvider: manages storage of membership information in Active Directory & Active Directory Application Mode servers.
     - SqlMembershipProvider: this provider manages storage of membership information in a SQL Server Database.
   - RoleProvider class define a structure of ASP.NET implementation to support role management features. They have 4 following built-in derived classes:
     - ClientRoleProvider: Get role information for Windows-based applications from a Microsoft Ajax roles service
     - AuthorizationStoreRoleProvider: Manages storage of role-membership information for an ASP.NET application in an authorization-manager policy store, either in an XML file, in an Active Directory, or on an Active Directory Application Mode server
     - SqlRoleProvider: Manages storage of role membership information for an ASP.NET application in a SQL Server database
     - WindowsTokenRoleProvider: Gets role information for an ASP.NET application from Windows group membership

b) In our case, we can utilize the SqlRoleProvider to implement the role-based authorization service in our ASP.NET MVC website while still keeping the original User data at the enterprise server.

Design justification:

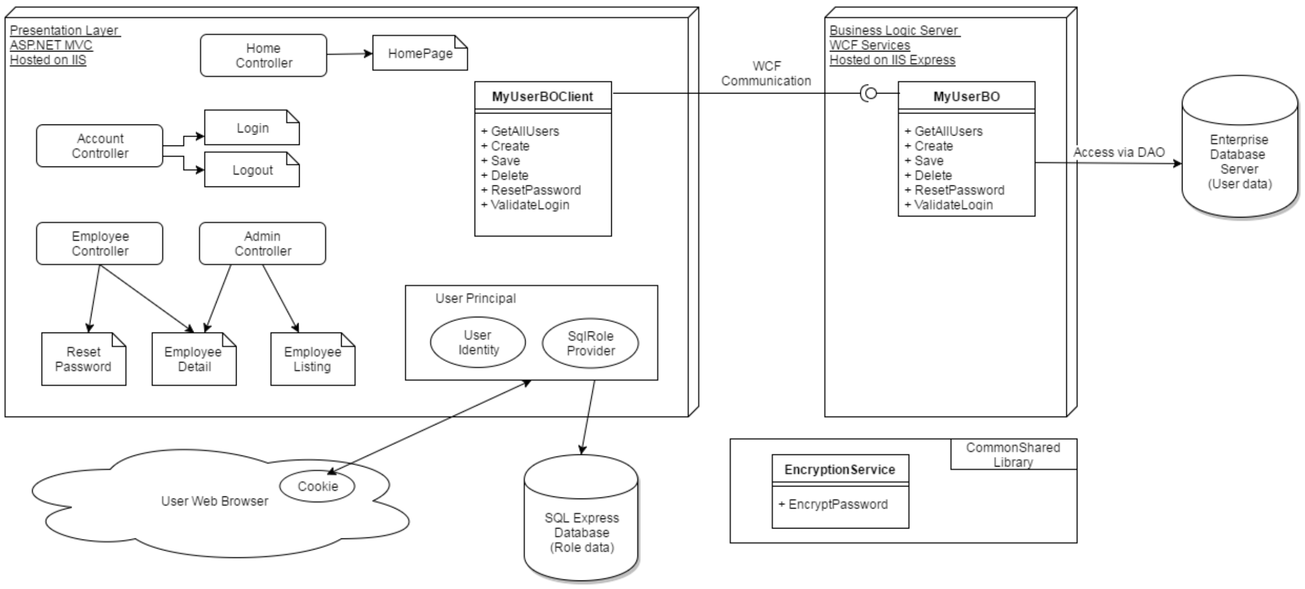


Figure 1. Architecture diagram.

Business Logic Server: This layer is no different with the implementation in Pass Task 8.2. The business logic object MyUserBO is using in the layer as part of the authentication process to validate user login by using user name and password. However, the role information doesn't need to be checked and stored in the enterprise database. Role information is managed and stored at the application layer by using SqlRoleProvider service.

Presentation Layer: This layer is similar with the implementation in Pass Task 8.2. However, instead of using a custom code and member attributes in UserPrincipal, this implementation uses the Roles class to check if user is in a role or not. This Roles class is provided by .Net Framework to manage user membership in roles for authorization checking in the ASP.NET MVC application:

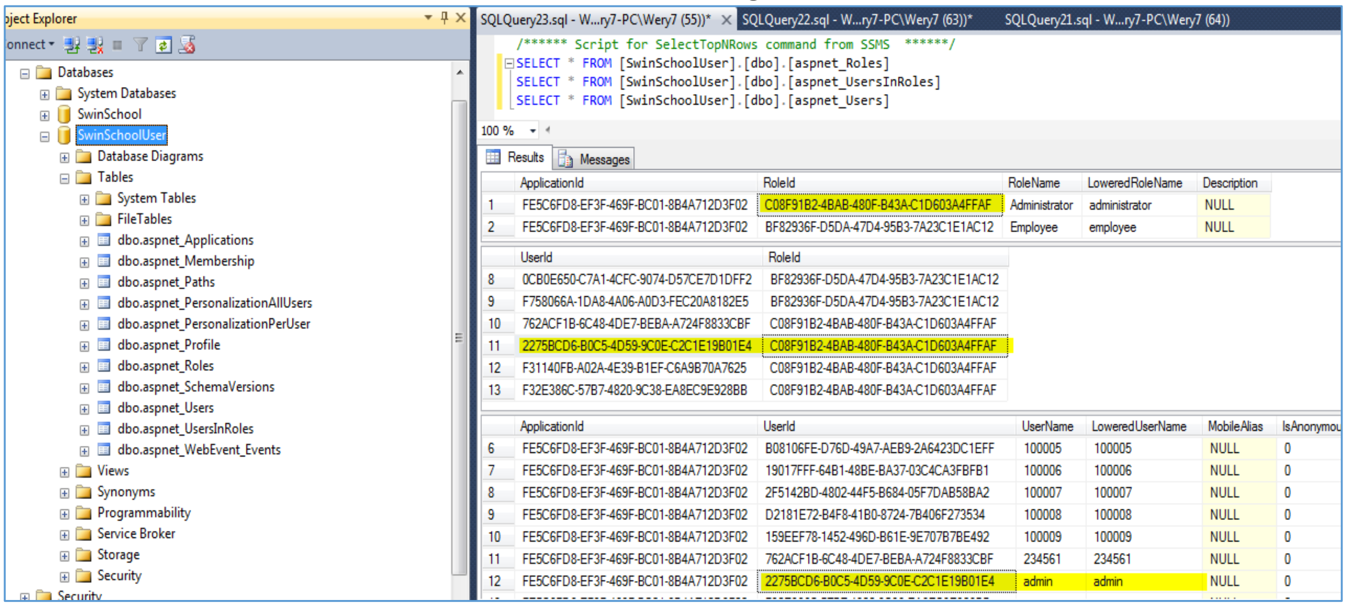
```
public class UserPrincipal : IPrincipal
    {
        private UserIdentity _identity;
        …………
        public bool IsInRole(string role)
        {
            return Roles.IsUserInRole(_identity.UserID, role);
        }

        …………
    }
```

This Roles class use the behavior of a RoleProvider that configured in the web configuration file of the project. We specifies explicitly that SqlRoleProvider is enabled on this project and uses a connection to a client database where the role information is stored to work. In this example, it stored in the SwinSchoolUser database at the localhost SQLExpress Server:

```
<connectionStrings>
    <add name="SchoolContext" providerName="System.Data.SqlClient"
connectionString="Data Source=localhost\SQLEXPRESS;Initial
Catalog=SwinSchoolUser;Integrated Security=SSPI" />
  </connectionStrings>
……
……
<roleManager defaultProvider="MyRoleProvider" enabled="true">
      <providers>
        <add name="MyRoleProvider"
             connectionStringName="SchoolContext"
             applicationName="SwinSchool"
             type="System.Web.Security.SqlRoleProvider" />
      </providers>
    </roleManager>
```

This database has to be pre-setup by using the aspnet_regsql tool found under installation folder of .Net framework. The schema of this database is generated as below:



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT * FROM [SwinSchoolUser].[dbo].[aspnet_Roles]
SELECT * FROM [SwinSchoolUser].[dbo].[aspnet_UsersInRoles]
SELECT * FROM [SwinSchoolUser].[dbo].[aspnet_Users]
```

	ApplicationId	RoleId	RoleName	LoweredRoleName	Description
1	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	C08F91B2-4BAB-480F-B43A-C1D603A4FFAF	Administrator	administrator	NULL
2	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	BF82936F-D5DA-47D4-95B3-7A23C1E1AC12	Employee	employee	NULL

	UserId	RoleId
8	0CB0E650-C7A1-4CFC-9074-D57CE7D1DFF2	BF82936F-D5DA-47D4-95B3-7A23C1E1AC12
9	F758066A-1DA8-4A06-A0D3-FEC20A8182E5	BF82936F-D5DA-47D4-95B3-7A23C1E1AC12
10	762ACF1B-6C48-4DE7-BEBA-A724F8833CBF	C08F91B2-4BAB-480F-B43A-C1D603A4FFAF
11	2275BCD6-B0C5-4D59-9C0E-C2C1E19B01E4	C08F91B2-4BAB-480F-B43A-C1D603A4FFAF
12	F31140FB-A02A-4E39-B1EF-C6A9B70A7625	C08F91B2-4BAB-480F-B43A-C1D603A4FFAF
13	F32E386C-57B7-4820-9C38-EA8EC9E928BB	C08F91B2-4BAB-480F-B43A-C1D603A4FFAF

	ApplicationId	UserId	UserName	LoweredUserName	MobileAlias	IsAnonymou
6	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	B08106FE-D76D-49A7-AEB9-2A6423DC1EFF	100005	100005	NULL	0
7	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	19017FFF-64B1-48BE-BA37-03C4CA3FBFB1	100006	100006	NULL	0
8	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	2F5142BD-4802-44F5-B684-05F7DAB58BA2	100007	100007	NULL	0
9	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	D2181E72-B4F8-41B0-8724-7B406F273534	100008	100008	NULL	0
10	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	159EEF78-1452-496D-B61E-9E707B7BE492	100009	100009	NULL	0
11	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	762ACF1B-6C48-4DE7-BEBA-A724F8833CBF	234561	234561	NULL	0
12	FE5C6FD8-EF3F-469F-BC01-8B4A712D3F02	2275BCD6-B0C5-4D59-9C0E-C2C1E19B01E4	admin	admin	NULL	0

A benefit of using SqlRoleProvider is that it speed up the development process when implementing a role-based security model in an application. It provides a quick and comprehensive tool for checking and manipulating user roles and store that information in a SQL Database. In ASP.NET MVC, each method in a controller is a resource URL that can be access via a web browser. In order to protect those resources from unauthorized access, we can decorate the controllers with Authorize attribute at the class level or method level:

```
[Authorize(Roles = "Administrator,Employee")]
    public class EmployeeController : Controller
    {

        [Authorize(Roles = "Administrator,Employee")]
        public ActionResult Index()
        {
                ………
        }
    }
```

To add an user to a role, we can simply call the following method AddUserToRole: e.g

```
Roles.AddUserToRole(userData.UserID, Constants.RoleValue.Employee);
```

Test cases:

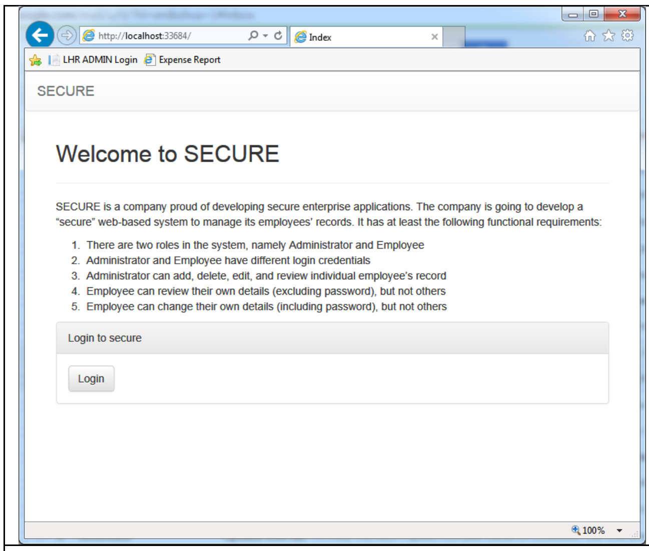
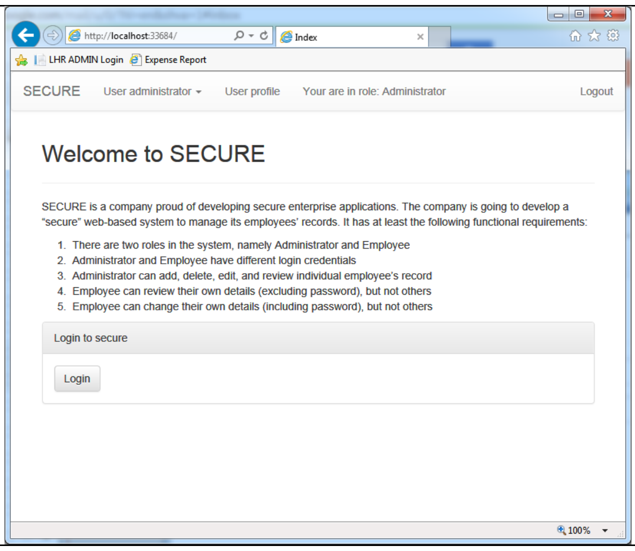
The following test cases have been taken out to make sure that the whole application works as expected:

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:

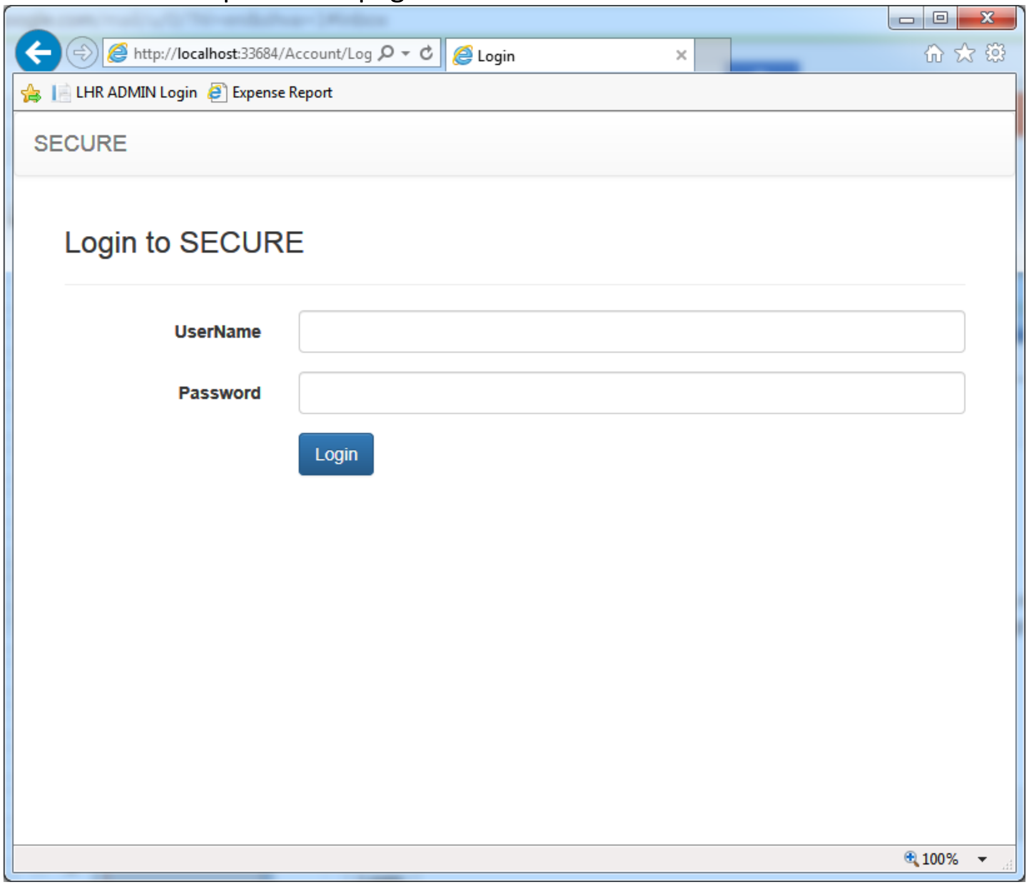
1. There are two roles in the system, namely Administrator and Employee
2. Administrator and Employee have different login credentials
3. Administrator can add, delete, edit, and review individual employee's record
4. Employee can review their own details (excluding password), but not others
5. Employee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

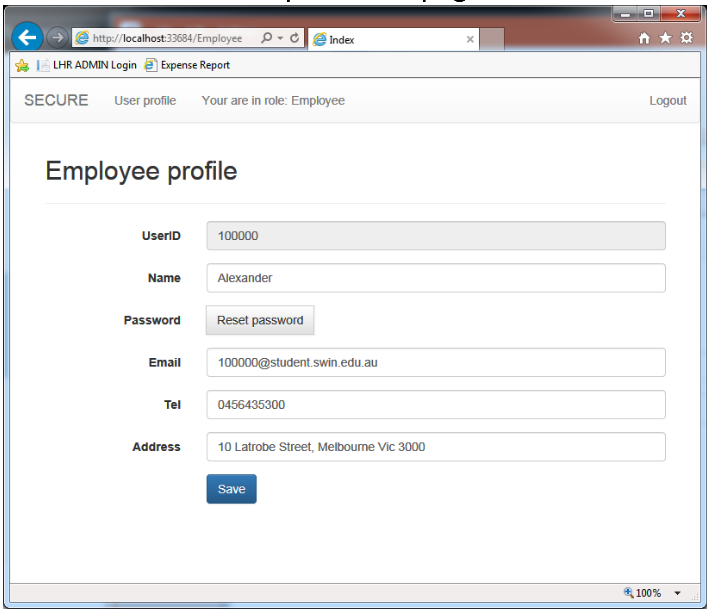
Login to SECURE

UserName
Password
Login

100%

The login page is displayed

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

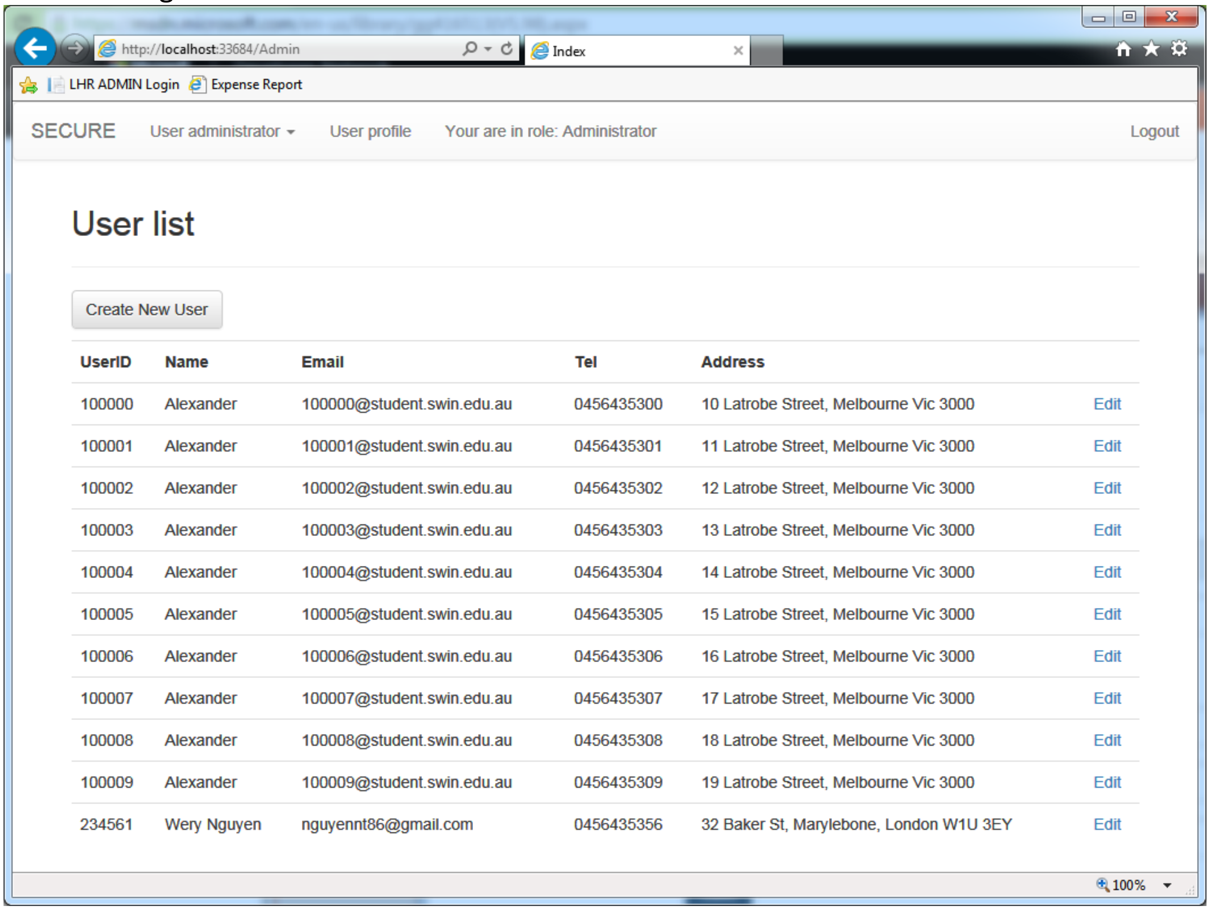


Employee profile

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

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

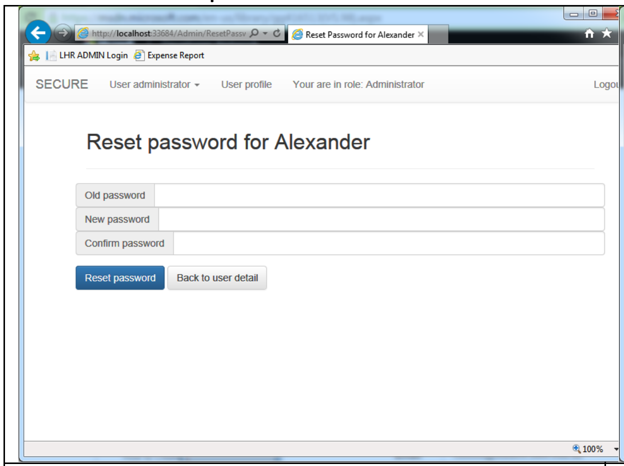
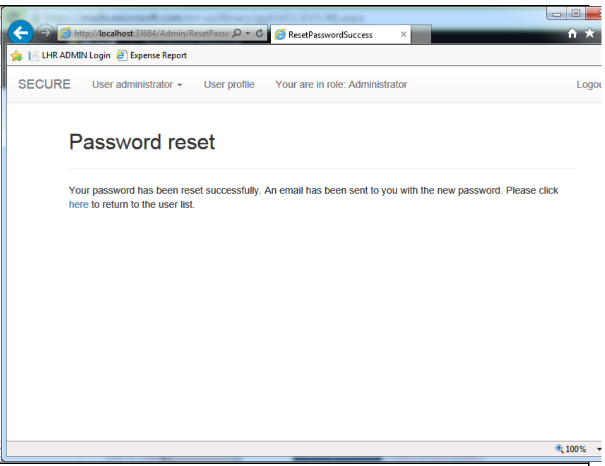
Test 05 – Login as admin



User list

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

Reset password for Alexander

Old password

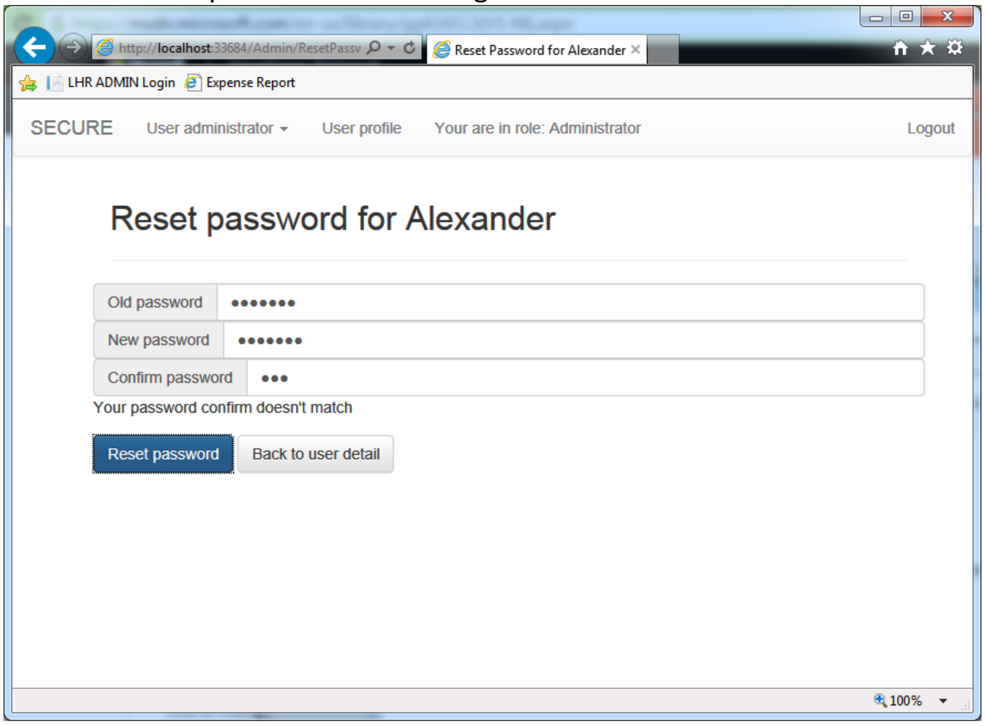
New password

Confirm password

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



Reset password for Alexander

Old password •••••••

New password •••••••

Confirm password •••

Your password confirm doesn't match

References:

(1) 2016, *RoleProvider Class*, Microsoft MSDN, viewed 03rd Jun 2016, <https://msdn.microsoft.com/en-us/library/system.web.security.roleprovider(v=vs.110).aspx>

(2) 2016, *MembershipProvider Class*, Microsoft MSDN, viewed 03rd Jun 2016, <https://msdn.microsoft.com/en-us/library/system.web.security.membershipprovider(v=vs.110).aspx>

(3) BitBucket source code: https://bitbucket.org/werynguyen/swinschool/src/fa1bf4f02fc80e1d1d445caf256d526e04692a30/?at=CT84