Pass Task 3.2 Business logic layer

Design justification:

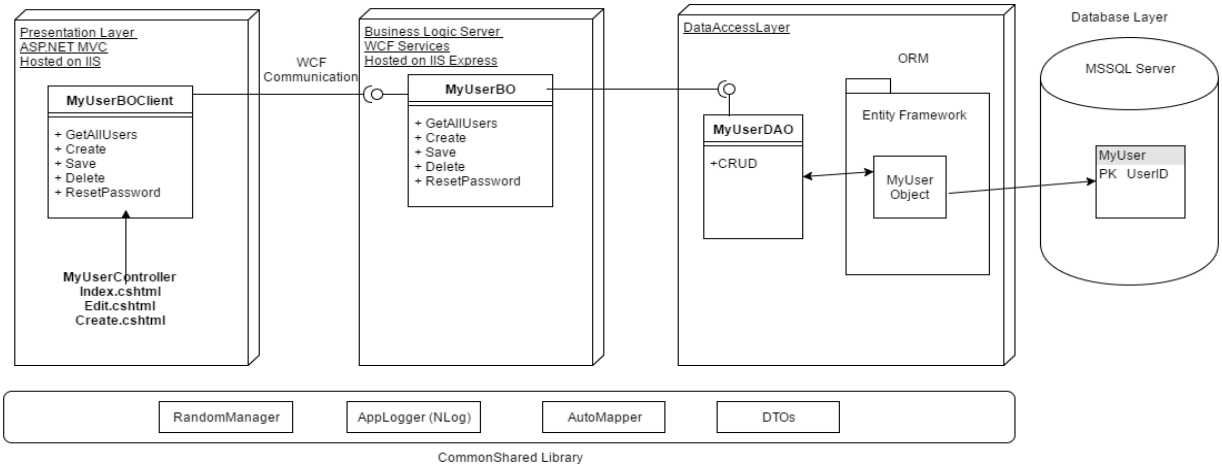


Figure 1. Architecture diagram.

Database Layer, DAL: These layers are similar to the design on PassTask 2.2: A MSSQL Server database, effectively SQL Express server which contain the data table MyUser. DAL contains one DAO object: MyUserDao that implemented using Entity Framework to communicate with the MyUser data table.

Business Logic Server: This layer is implemented using WCF Services. It contains 1 service MyUserBO that expose an interface or more service classes which maintain business logics and provide business data to the upper level in the format of data transfer object (DTO). This layer will handle tasks related to the business rules such as: business rule data validation, data processing and manipulating, data mapping from entity object into DTO and vise-versa. This layer use DAO interfaces provided by the Data Access Layer to interact with the persistence storage. The Business Logic Service is deployed to an IIS hosting service with a specific port number and URL address.

Presentation Layer: The design of this layer is similar to the design on PassTask 2.2. However, instead of using the Business Logic directly as an object reference, it use a proxy class to handle the communication with the equivalent business object in the Business Logic Server. This class is generated using the svcutil.exe tool provided by the .Net Framework.

Common Shared Library: This is a shared library between all layers in this architecture which contains DTO, helper classes, logging tools that can be used by all layers. Data mapping could be considered to put into this layer. However, this library doesn't have any knowledge about Entity Object, the mapping of Entity Object to DTO cannot be put here and have to put into Business Logic Layer. The AutoMapper is a class library that facilitate the object mapping process that is referenced by the Business Logic Server.

Design justification:
This design makes it possible to deploy each component of a system onto a separate or multiple machines, thus making the overall system much more scalable. In this case, the Business Logic Server can be deployed on to 1 server so that the logic can be reused for different application. For example, we can easily build a mobile application that use the same user authentication function provided from this Business Logic Server. In term of development, this architecture allows each part of the system be developed or managed separately by different teams. For instance, web developer can build the Web front end while a separate team maintain the Business Logic Server.

However, this design also comes with some disadvantages. First of all, 2 servers have to be utilize to run the Web application and the Business Logic Server. In an enterprise environment, this mean that more administration and security issue have to considered. Secondly, an extra layer of communication between the Presentation Layer and the Business Logic Layer have to be developed. In our example, we can see that each method request may require a DTO object to be defined for the request and response.

In my opinion, although this design may introduce more cost and complexity in development, it allows the logic layer to be reused by multiple applications and improve the scalability of the whole system.

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

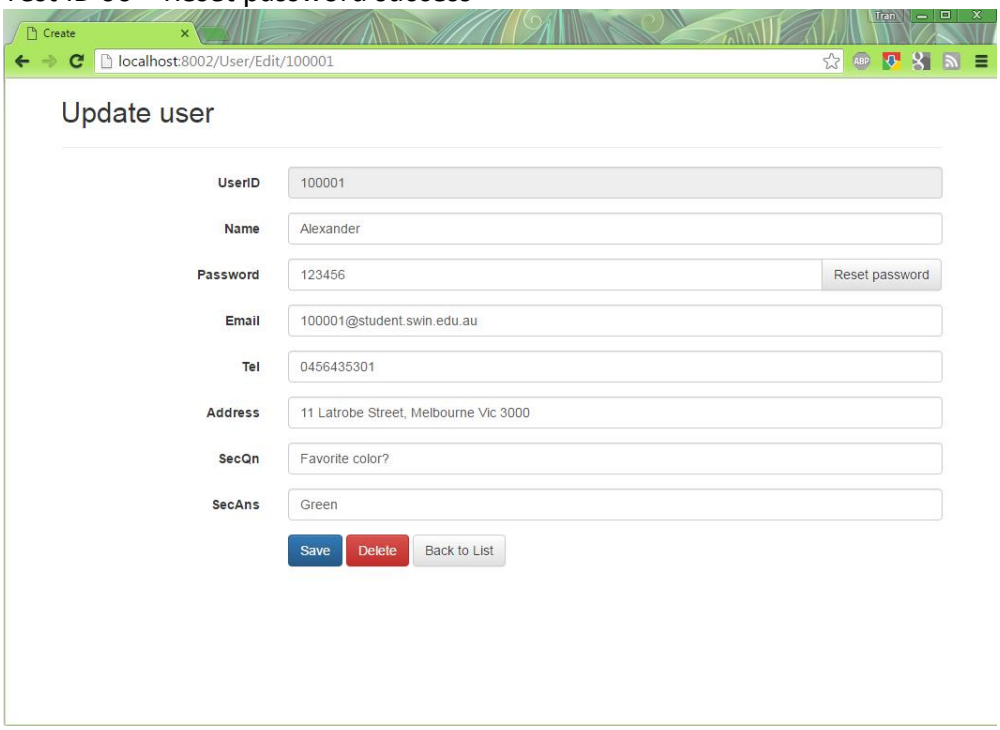
ID	Test Case Name	Step	Expected Output	Result
01	Get all users	View the page /User	A list of 10 users is displayed on the screen	Pass
02	Get one user	Click to the Edit link of one user in the list	Display the detailed information of that user in a form	Pass
03	Create an user	- Click to the Create New link. - Enter new user information - Click Create	The new user is added to the list. The list is now showing 11 users	Pass
04	Update existing user	- Click to the Edit link of one user in the list - Change one or more detail of the user using the form - Click Save	The updated information of that user should be displayed on the list	Pass
05	Delete existing user	- Click to the Edit link of one user in the list - Click Delete	The user is removed from the list	Pass
06	Reset password success	- Click to the Edit link of one user in the list	A success message will display	Pass

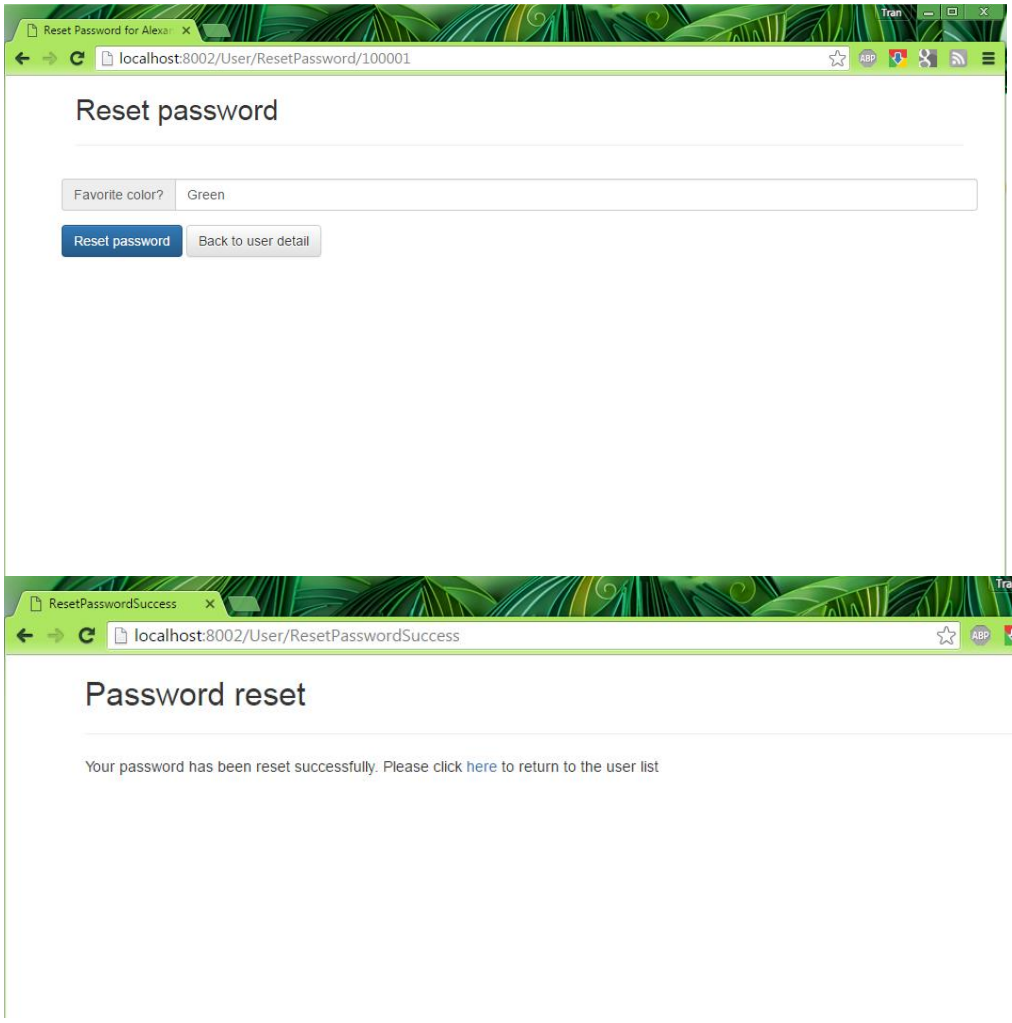
		- Click on the “Reset Password” button - Enter the correct answer for security question	The password is randomly modified A log file is record on the Business Logic Server	
07	Reset password fail	- Click to the Edit link of one user in the list - Click on the “Reset Password” button - Enter the wrong answer for security question	An error message is display.	Pass

Table 1. Test cases and test results

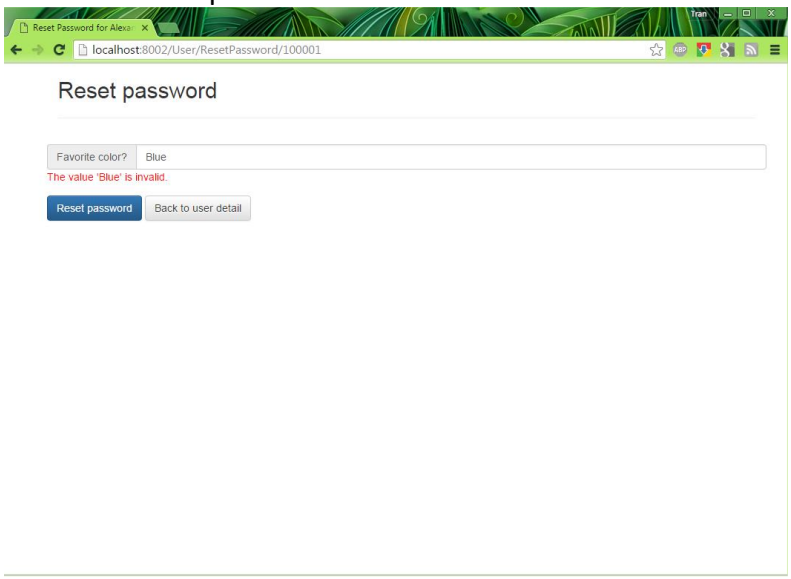
Test case screen shots and success screen shots:

Test ID 06 – Reset password success





Test 07 – Reset password fail



References:

(1) 2015, *How to: Host a WCF Service in IIS*, Microsoft MSDN, viewed 31th Mar 2015, <https://msdn.microsoft.com/en-us/library/ms733766(v=vs.110).aspx>

(2) BitBucket source code:
https://bitbucket.org/werynguyen/swinschool/src/55e559ffe9792cc33ce35ff41bf34771a85789da/?at=PT32