Pass Task 3.2 Business logic layer

Design justification:

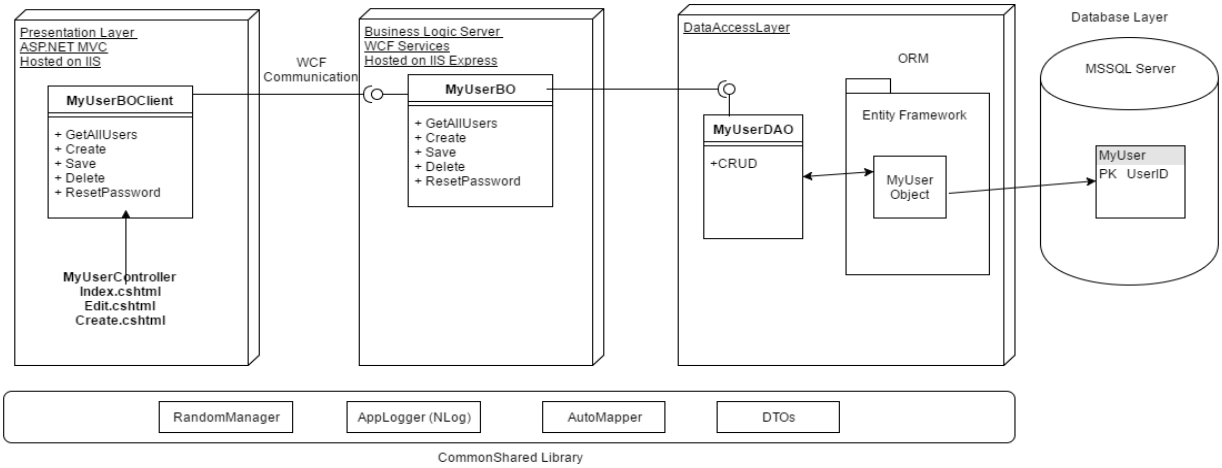


Figure 1. Architecture diagram.

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:
In distributed computing, this design is the most effective solution that make it possible to deploy each component of a system onto a separate or multiple machines. When a component can be deployed to multiple machines, it is clear that the design ultimately offer a scalable system where it is possible to deploy a grid computing systems of a web farm or business logic farm servers. As a result, such system is capable of servicing a massive requests without delays or system overload. With the support of latest technology trends such as cloud computing, using virtual machines, virtual configurable platforms using Docker, it is possible to configure the system to automatically scale depend on the number of requests this system has to handle. In term of development, each component can be developed or managed separately by different teams, which potentially improve the efficiency of system development. Such system models are essential for large scale environment such as government IT network, banking system or stock market system, global enterprise IT systems and so on…

However, those benefits come with a high cost that have to be considered carefully before utilize this development strategy. First of all, the more servers to be deployed, the more administration and security that system administrator have to managed. Each server on each different geographical locations may have different security protocols that have to be managed. The task of preventing system hacking from multiple servers is more complex and difficult than managing on one server. Secondly, there is more restriction and cost applied when running such distributed system. Distributed computing always involved with network connection and that connections have to be maintained though the whole transactions. Therefore, each workstation have to be connected to the network and maintain connected while accessing to those system servers. Thirdly, the cost of maintaining the system can be a deal-breaker in the long run. Maintaining server hardware, renting connection and bandwidth, hiring administrator staff are major cost involved to make that system online that usually result in a high bills to pay. Furthermore, upgrading a distributed computing system is a complex process that may negatively affect system uptime. An important attribute of software system is that it is continuously evolving with new features, updates to support the rapid changing of real world. In distributed system, each update need to be carefully tested and deployed on all machines so that all the components are compatible with each other. In the event that multiple teams involved in developing the system, there must be a clear communication between all teams when a deployment taking place so that the system as a whole is not crash.

In my opinion, the disadvantages of this design outweigh the advantages in middle and small systems. Distributed computing has a high cost in implementing and utilizing that make it only feasible for massive enterprise. Distributed computing should only be implemented when there is a real needs of its offering. Nevertheless, if distributed computing is correctly implemented, the power of such system is limitless.

Test cases:

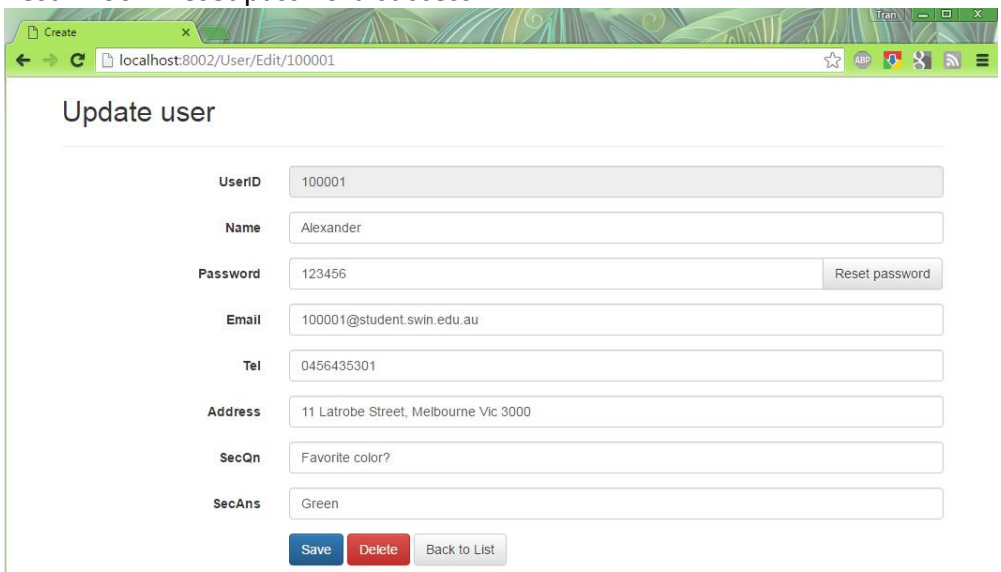
The following test cases have been taken out to make sure that the whole application works as expected:

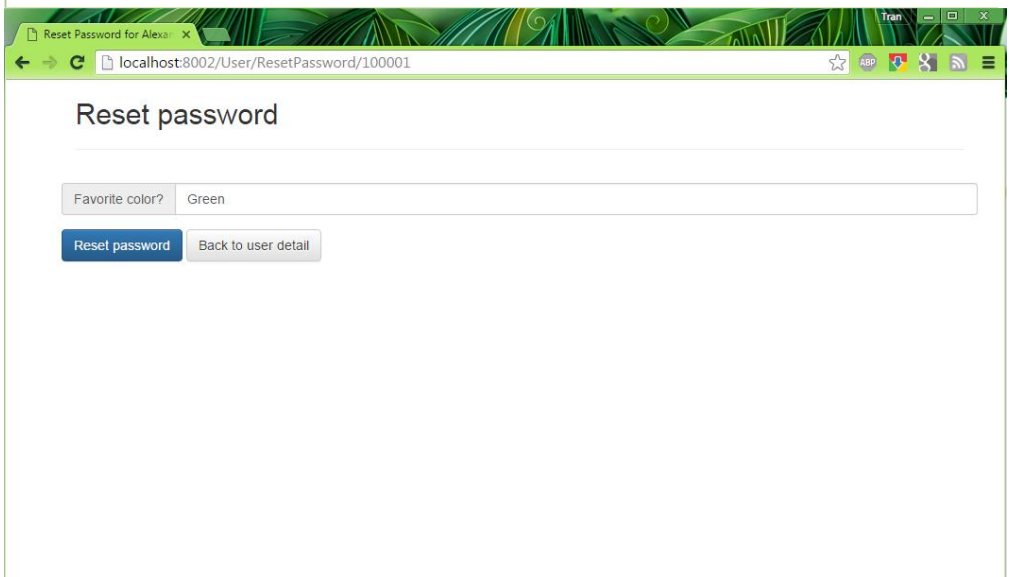
ID	Test Case Name	Step	Expected Output	Result
01	Get all users	View the page /User	A list of 10 users is displayed on the screen	Pass
02	Get one user	Click to the Edit link of one user in the list	Display the detailed information of that user in a form	Pass
03	Create an user	- Click to the Create New 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 - Click on the “Reset Password” button - Enter the correct answer for security question	A success message will display The password is randomly modified A log file is record on the Business Logic Server	Pass
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

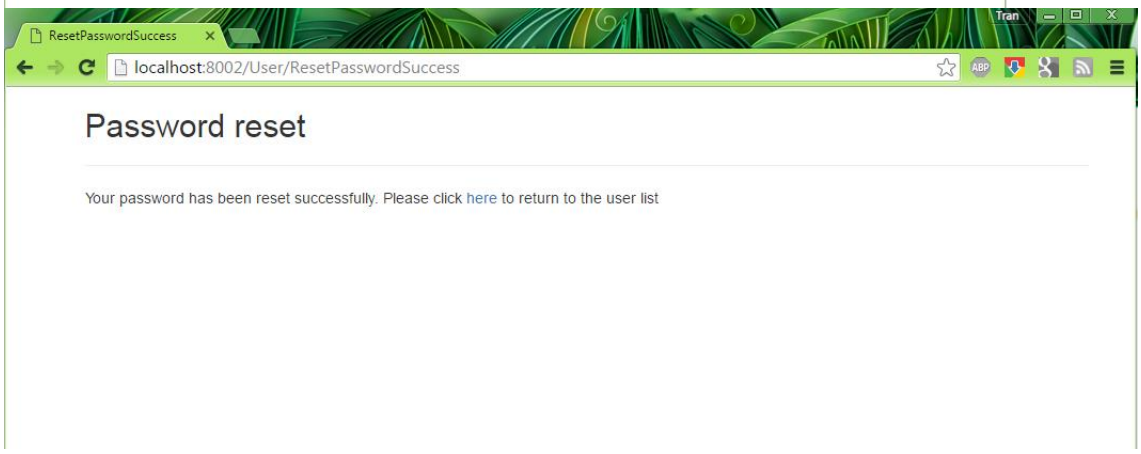
Table 1. Test cases and test results

Test case screen shots and success screen shots:

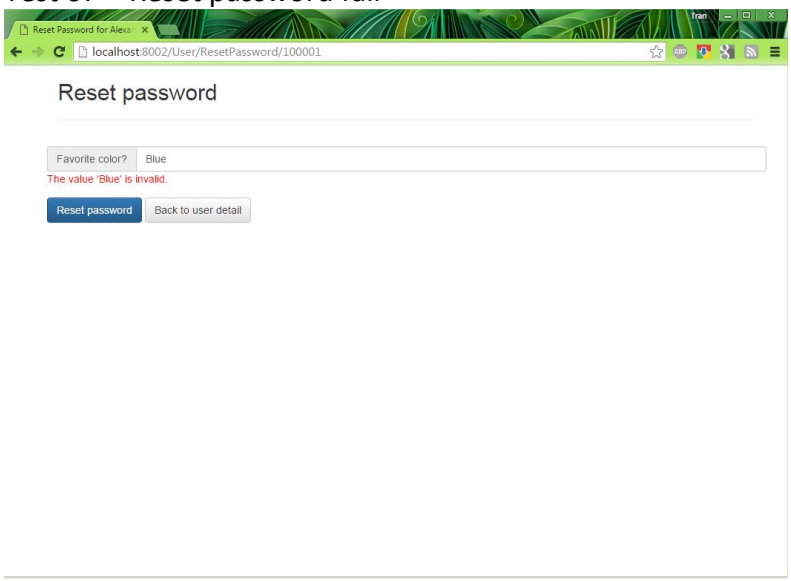
Test ID 06 – Reset password success







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