Pass Task 6.2 Messaging

Design justification:

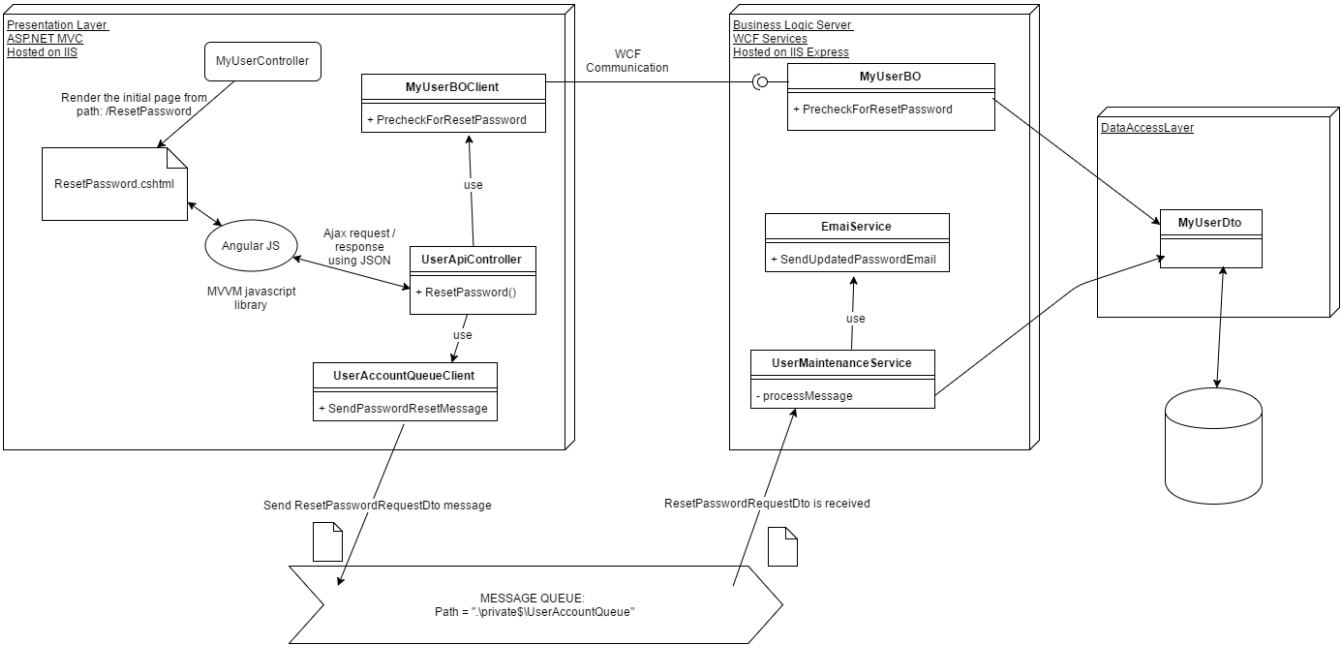


Figure 1. Architecture diagram.

Database Layer, DAL: This layer is very simple with one Dao object to interact with the MyUser table in Sql Express Database using Entity Framework.

Business Logic Server: In this implementation, the business logic server contains 2 services: MyUserBO and UserMaintenanceService. The MyUserBO is effectively a WCF service which is exposed via IMyUserBO interface. The main service it provides in this scenario is PreCheckForResetPassword that is to check the SecAnswer from the user if it is valid or not. The second service is UserMaintenanceService that actively listen on a MessageQueue on MSMQ server. Once there is a message available, it will receive the message out of the queue and process it. This service uses a helper EmailService to send email to user.

Presentation Layer: This layer is similar to the design on Credit Task 4.4 where it use Ajax model to communicate between client & server front end. The UserApiController acts as a web service role that will call the MyUserBO to check the SecAnswer, and if the result is valid, it will use UserAccountQueueClient to send a message to the MessageQueue on the MSMQ server.

Design justification:
Compare to the implementation of Business Logic Layer in Pass Task 3.2, the using of messaging object in this design has a little delay before a new password is generated. However, a quicker response is achieved as the client does not need to care when the password update process is completed. Once a message is sent to the queue, the client response straight away to the user even when the message is not yet processed in the

MSMQ. In term of software architecture, the MSMQ further promote a loosely coupled of components or layers in the system. The client program can run without any knowledge on the process of the backend service. The backend logic of UserMaintenanceService is also running independently from the client.

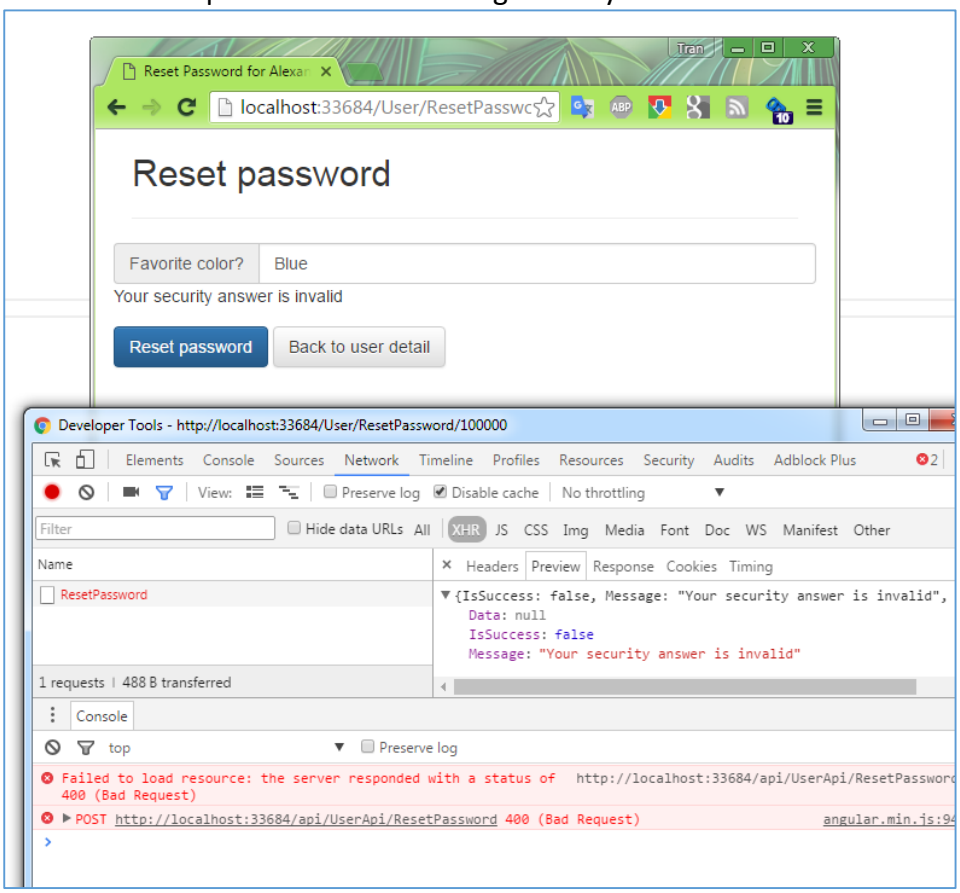
Test cases:
The following test cases have been taken out to make sure that the whole application works as expected:

ID	Test Case Name	Step	Expected Output	Result
01	Reset password with a wrong security answer	Request to reset password for an account Enter a wrong security answer to the input.	An error message should be displayed and the form stays the same.	Pass
02	Reset password with correct security answer	Request to reset password for an account Enter the correct security answer to the input	A success page is displayed for the user A log file is recorded with the new password update An email is sent to the user email address with the above password	Pass

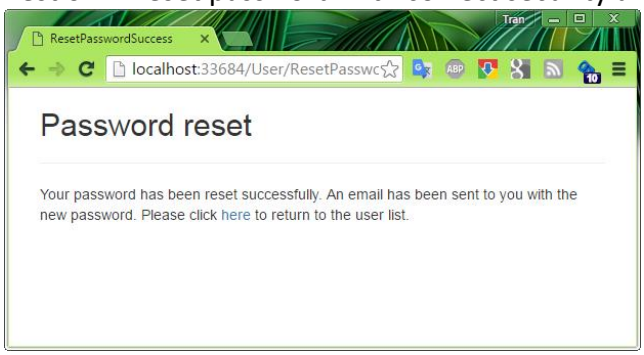
Table 1. Test cases and test results

Test case screen shots:

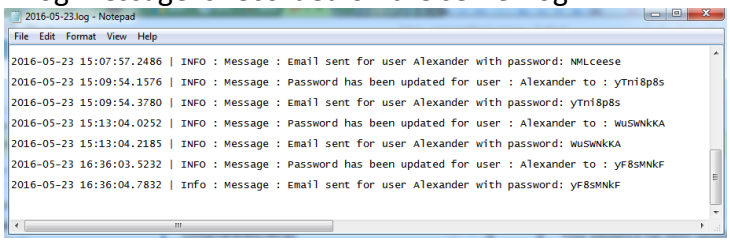
Test 01 – Reset password with a wrong security answer



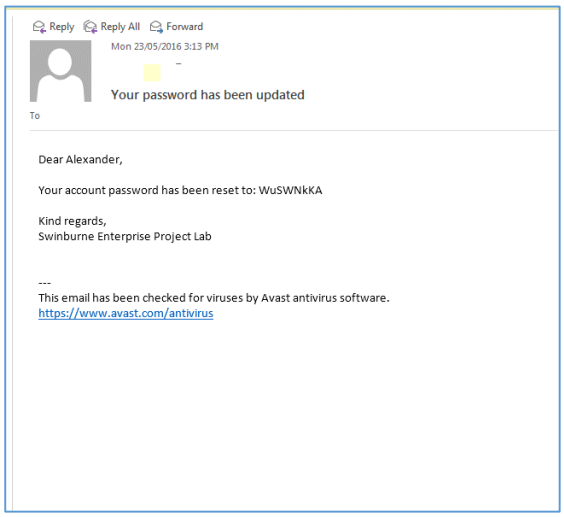
Test 02 – Reset password with correct security answer



A log message is recorded on the server log:



An emai is sent to user email address



References:

(1) 2016, *SmtpClient Class*, Microsoft MSDN, viewed 23th May 2016, <https://msdn.microsoft.com/en-au/library/system.net.mail.smtpclient(v=vs.110).aspx>

(2) BitBucket source code: https://bitbucket.org/werynguyen/swinschool/src/5c746123ab319e06ba450191c9dd63b834a62ce6/?at=PT62