**Delloite ToDo Application - Stephen Carey**

**Discussion on work done**

* **Approach taken**

Took the approach of attempting to make it as extensible and maintainable as possible.

I tried to decouple the code as much as I could in pursuit of the above.

I didn't use any specific design patterns.

I tried more to hit all the requirements, AKA the CRUD operations and just ensuring everything worked. I didn’t spend very much time scoping it as it seemed simple but that came back to bite me when i was setting up classes and designing how they interacted with each other and the web forms.

* If i had more time

If i had more time I would put in some better error handling, comments, spruce up the UI and if I had loads of time, go back and rethink the layout of classes and the flow of the application.

* **Notes**

1. Time spent - Approx 8-9 Hours
2. The **error handling** is poor. If i had more time i would create an error handling class to handle them. If i was using a desktop application i would save these to a log file. Not actually certain how to deal with that in a web application. Regardless, the errors could be a lot better. It would be good to display the actual error message in the status textbox.
3. The **UI** is not very good. Front end design isn't my strong point. Even when I know all the tools available to me, I couldn't build a nice looking application to save my life. I decided it would take too long and not be worth the benefit to spruce up the look and feel. I focused more on functionality.
4. **Security** is not the best. Had difficulty finding a way to hide the password characters. Also the SQL queries do not account for SQL injection. Beyond that, I don’t have huge knowledge in the realm of web security, something I need to work on.
5. **Performance** isn't bad but i question what happens when multiple users are accessing the site simultaneously. I never actually close the connection to the SQL DB. This helps while the site is active as we don't have to open and close multiple connections but i don't have a way of knowing when the site is closed and as such don't ever close the connection which i think is an issue.
6. **Extensibility** could be a lot better. While I tried to make the website extensible, I didn't have time to add a sign up screen which is a pity as it wouldn't have been too hard to do. A lot of the code is hard coded and making changes or extending the applications functionality would break most of the classes code so as much as I say I tried to decouple the code, I didn't really. Perhaps making another SQL class which was an abstraction above the one currently in use would be a good idea.
7. I didn't put in any **comments** at all which is something I need to work on. Nothing really to say here. I should have put them in as I go but I planned on doing it at the end. Of course I ran out of time and didn't put them in.
8. In terms of multiple users on the website simultaneously, I don't have any real safety checks. I use sessions to differentiate between them but I have nothing protecting any data. Not sure if putting locks around them would be enough or not.

* **Other notes**

There are 2 users in the Login Table DB who have access to the website.

Username = stephen, password = MyPassword

Username = user1, Password = pass1

* Had to set up SQL Server on my PC for this, so that ate into time a bit.
* I think looking at the variables, time I spent on the project, and to what extent I fulfilled the requirements, I think it's a fair reflection on my abilities.
* I enjoyed the project. If I move further through the application process, I would love some feedback on how I could have done better with structuring the code throughout the project.