Pass Task 5.1. Business logics 2 – Stateful behaviour

In object oriented programming, we know that objects have state and behaviour. State is represented by instance variables while behaviour is represented by methods. In enterprise system, business objects with instance variables can be considered as they have state and that state is maintained throughout the whole transactions. Take ecommerce application as an example, business object maintaining customer's cart can be an example of a business object with instance variables. Each time the customer adding or remove item to and from the cart, then the state of the cart has to be maintained in each action by the cart business object. In contrast, business objects without instance variables do not maintain its state. The lifetime of those objects is spanned in a single method call only. For example, the google map application will send a map tile graphic to the client's browser per request. Normally, there are many requests that are sent at a time when a client view a map region. Those requests are best handled by a stateless business object because there is no need to maintain the state of client after each request.

Stateful business objects come with many advantages as well as disadvantages. The upside of using stateful business object is that it uses dedicated computing resources and reserve them for the whole client transaction. This behaviour is preferable for enterprise application where the client is consistently sending request to the server and those requests has a high computation cost on the server end. Maintaining state of the objects would reduce the time that server need to handle the request.

In addition, maintaining state of a business object means that the information of that object is stored in memory and ready to be use at any time. This can be considered as a caching method and thus it will help saving round trip to the lower speed persistence storage such as DBMS or file system. As a result, it takes the less time to process the request by saving the computation cost.

However, this behaviour is potentially wasting computing resources. In client server design model of the web server, it is very often that the client after a few request to the server, the user navigate to other browser tab or leaves the desk for a break. Then if the server is maintaining the state for the client, the resource spending for that cause will be wasted. It would be more efficient for the server to release that resources holding the state, pointing that resource to serve other customers while waiting for this current user to come back. In the event that the current user comes back, the server can recreate the state by using a persistence storage such as Database or Memcache system.

Moreover, it is more complex to develop and has a high risk that runtime error will occurred due to the reliability of the physical network connection. Stateful business object is made possible by maintaining the connection and the state of both client's and server's objects. If for a reason that a request sending from client is not reaching to server, then it would be possible that the client state and server state are mismatch that would pose a risk of runtime error occurred for subsequence method calls.