**Observables and Observable Streams**

An Observable is basically a function that can return a stream of values to an observer over time, this can either be synchronously or asynchronously (Lotanna, 2019). There are 3 different values that can be delivered to the Observable by an Observer which are:

1. Next value,
2. Error value, and
3. Complete value.

The Next Value is when the observer sends the value. Values can be objects, numbers or strings. The Error Value sends JavaScript exceptions and if there is an error then nothing else may be delivered to the Observable. If there is a Complete Value then this usually tells the subscription that the Observable is complete and nothing more may be delivered.

Observables are data source wrappers. In order for Observables to work an Observer needs to execute some instructions to the observable through a subscription. The subscription lets the observer connect to the observable through a block of code. Observables have a lifecycle that contains 4 different stages.

1. Creation
2. Subscription
3. Execution
4. Destruction (Lotanna, 2019).

Observables asynchronously send out values, however if there isn’t a subscription then the values will not be sent out. Errors can also be handled within Observables.

Streams are data values that are in a sequence which have been collected over time. When Observables produce multiple values it is then called an Observable Stream. It then pushes the values out to a customer or client. Observables’ communication system is basically a push or a pull communication. Observables are so vital and can be resolved multiple times as well in comparison to functions, promises, etc.