CW3 Discussion and Summarize

Through the process of re-designing, it become clear that, we need Quote class as the Wrapper class of the Bike class when get Quote is required. It is better idea to return collection of Quote rather than collection of Bike which is searched form BikeList directory. The benefit of implementing Quote is not only this. Quote class is designed to calculate deposit by using Valuation policy BikeProvider chose. Therefore, it is the solution to “How to integrate extension Valuation Policy”.

The required class BikeType, Location, DateRange is implemented. In addition, the LinerDepriciation and DoubleDecliningBalanceDepreciation which implements ValuationPolicy is implemented too.

The core class of the system, Bike, BikeProvider, Customer remains almost same with some changing in their attribution. New attribute booked which is List<DateRange> contains unavailable DateRange. This enable easier search for Bike satisfied condition by using overlap() in DateRange .

The Invoice class remains same, and it will be used in book quote stage. There is some change in Condition class. It attributes for searching appropriate bike, in addition it has static method toCondition() which get user input(String[]) and return Condition which will be used in getQuote().

Booking class is still existing, however, its behavior is different from previous plan. The most of the process required for book quote use case is done when Booking constructor is called in bookQuote() in Customer class. It will construct an invoice print it out, at the same time, notify to both customer and bike provider, if delivery is required, it will schedule delivery too. (notify method and display method is not implemented because these are outside the system).

Finally, ProviderList and ListofBooking is added to system for the convenience of storing and searching data.

About test, there is three unit-test for DateRange, Location, and ValuationPolicy. They check the functionality of their methods. In the system test, there is several test for checking and simulating each required use case, and in the integration test extension and all the required use case is integrated as one scenario and checked.