

Informatics 2C – Introduction to Software Engineering Coursework

3

Team Member:

Teh Min Suan (s1817967)

Michitatsu Sato (s1807428)

CW3 Discussion and Summarize

In the process of redesigning, it becomes clear that we need a wrapper class for bike named Quote to include deposit. It is better to return a collection of Quote rather than a collection of Bike when searching for bikes as we need to calculate deposit of bikes. Quote class is designed to calculate deposit based on the deposit policy set by BikeProvider. If none, default policy is used to calculate deposit.

There are 4 classes that work as data holder, namely Invoice, Location, DateRange and Condition. Location is a class that record address and postcode while DateRange is a class that record two LocalDate object indicating start and end of booking. Even though we don't have to include a Condition object to search for bikes, it is more convenient to have an object that holds all the requirements of customer and passing this information

around is easier. Also, a new method called toCondition() is added to Condition class to convert user input to Condition class. Invoice is a summary of all booking information. It is designed as planned.

There are 4 core classes in this system which are Bike, BikeProvider, Booking and Customer. Bike, Booking and Customer remains as planned in our current design. Booking now automatically prints out invoice to customer and notify both customer and bike provider after doing so.

Finally, there are 3 lists in our system. Bikelist which records all bikes remains the same in the current design. ProviderList and BookingList are newly added to our system. BookingList is needed as we need to find a Booking object using orderNumber. Having a providerList makes filtering bikes easier when searching for bikes.

Self-assessment

1. Extension submodules 10%

- Implementation of extension submodule
 - i. LinearDepreciation and DoubleDecliningBalanceDepreciation have both been implemented. We did create Junit tests for both class and they both run fine. 10%

- Peer review of other group's submodule
 - i. We did complete peer review of group 63 and feedback was provided 10%

2. Tests 35%

- System tests covering key use cases 20%
 - i. All tests are documented with comment to describe the method we used to check our implementation.
 - ii. Tests we run cover all use cases. Our tests will make sure that all methods run fine.

iii. A lot of test data has been setup before running the tests.

iv. MockDeliveryService has been used.

20%

- Unit tests for Location and DateRange 5%

- i. Tests for both class have been created

5%

- Systems test including implemented extension to pricing/valuation 5%

- i. The last test, called Integration test is the unit test for deposit valuation 5%

- Mock and test pricing/valuation behavior given other extension 5%

- i. We did try to do it but not sure if we did it perfectly 3%

3. Code 45%

- Integration with pricing and valuation policies 10%

- i. LinearDepreciation and DoubleDecliningBalanceDepreciation implement valuation policy while Quote use the implemented method to calculate deposit. 10%
- Functionality and correctness 25%
 - i. Each use case should be correctly implemented since we were able to run system test successfully. 25%
- Quality of design and implementation 5%
 - i. While we tried to follow rules of making a good design, there might still be something that we left out. Assertions are used in our code. 4%
- Readability 5%
 - i. We did follow coding standards therefore our code should be readable. Javadoc is also added for Location and DateRange

classes	5%
4. Report	10%
<ul style="list-style-type: none"> • Revisions to design <ul style="list-style-type: none"> i. We corrected our design document when implementing the system. Reason of the change is documented in report document. 	5%
<ul style="list-style-type: none"> • Self-assessment <ul style="list-style-type: none"> i. After implementing the system, we did a revise on our system according to the assessment criteria. 	5%