Inf2C - Software Engineering Coursework

Implementation Report

Team members:
Martin Lewis - s1824863
Alasdair MacGillivray - s1837803

Changes to Design during Implementation

In the initial design we suggested using unique IDs for each major class (Bike, Booking, Customer and provider). But we removed these on the advice of the feedback as it didn't make sense as we used them as an index to access the objects.
While we didn't use them for indexing the objects for access we did end up bringing them back as we needed a way to easily do equality on objects during testing. So we added a static counter for the class with each object using the current value of the counter as an ID. Then iterating the counter.

We also had to make some changes based on the skeleton code that we were given. On the simpler end was the change of class names like our DateRange class was called Duration. On the further end we modelled the type/generic type idea in a different way and then had to deal with the type being modelled as a separate class in the skeleton code provided. We also adjusted how the location worked as we assumed a more complex class members.
Also in our version of Location we had not designed a way to tell if two locations/addresses were close to each other, but the coursework 3 instructions told us how to implement isNear() in the skeleton.
There was also the idea that prices and monetary values were to be represented with big decimal rather than our preferred pence in integer we put in the design. This meant some of our methods had to be tweaked to work better with the big decimal class.

We also had to deal with the skeletons idea of a deliverable object where we had to add onPickUp() and onDropOff() function to booking. Which could then adjust the state/status of the Booking/Bike. This also then lead to another issue where we were using strings for the states but we wanted to easily control what values there could be. So we had to implement some enums to hold the various valid options for the state/status and then adjust the code that used the state/status to use these enums.

Self Assessment

1 Extension Submodules (9%)

Implement extension submodule (9%)

Added a functional multi day price submodule and created unit tests to ensure it worked correctly.

Peer Review (10% Bonus)

Completed peer assessment on 27/11/2019 (5-6pm) with group 12 and recived feedback as well as providing feedback

2. Tests (30%)

Systems tests covering key cases (17%)

Wrote system tests covering the use cases with comments documentation how they covered them. Also wrote system tests covering any multiclass substeps. These tests included MockDeliveryService and some variety of test data

Unit Tests for Location and DateRange(4%)
We wrote unit tests for these two classes covering normal usage of the classes
System tests including implemented extensions to pricing/valuation (5%)
We wrote our system tests including our pricing extension.

Mock and test pricing/valuation behaviour given other extension (4%)
We created two mock classes that carried out this behavior and we wrote some tests to ensure these worked correctly.

3. Code (39%)
Integration with pricing and valuation policies (9%)
Our system interfaces with the two policies correctly and interfaces with theses policies. And default policies were written that are used if no policy is specifically given.

Functionality and correctness (22%)
We implemented all the methods that we created in the design excluding the outside entity (UI) that calls these methods/classes. We also have system tests that ensure these methods work as expected

Quality of design and implementation (4%)
We have followed the design practices and made use of assertions for testing purposes

Readability (5%)
Our code followed the coding standards provided and we included full JavaDoc for the required Location and DateRange classes as well as some sporadic JavaDoc elsewhere in the code. We also made use of commenting frequently to explain code that was not immediately obvious.

4. Report (9%)
Revision to design (4%)
We made note of the major changes we made to the design during implementation and discussed the reasoning for these changes

Self-Assessment (5%)
We made a reasonable attempt at self assessment against the criteria given and our assessment of our own code and conduct in the implementation.

Total (87%)