Inf2C Software Engineering 2019-20 Coursework 3

Creating a software design for a federated bike rental system

S1979471 LIU Shouheng

S1980526 KLAUSS Fredrik

1. Extension submodules **10%/**10%

Implemented extension submodule, added multiple unit tests to show that our system is working

- Implementation of extension submodule **10%**/10%
  - Should implement extension submodule
  - Should include unit tests for extension submodule
- Peer review of other group's submodule (up to 10% bonus marks)

2. Tests **35%**/35%

Comments for every method and test, all use cases covered with a variety of test data.
MockDeliveryService is used

- System tests covering key use cases **20%**/20%
  - Should have comments documenting how tests check the use cases are correctly implemented
  - Should cover all key use cases and check they are carrying out the necessary steps
  - Should have some variety of test data
  - Should use MockDeliveryService

Added unit tests

- Unit tests for Location and DateRange **5%**/5%

Implemented thorough tests

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

Attempted, worked out

- Mock and test pricing/valuation behaviour given other extension (challenging) **5%**/5%

3. Code **42%**/45%

Integrated pricing and valuation policies, had to use provider for the pricing policy

- Integration with pricing and valuation policies **7%**/10%
  - System should correctly interface with pricing and valuation policies
  - System should correctly implement default pricing/valuation behaviour

Fully functionality for use cases, as evidenced by system tests

- Functionality and correctness **25%**/25%
  - Code should attempt to implement the full functionality of each use case
  - Implementation should be correct, as evidenced by system tests

Included assertions, good design practices are sometimes followed. Implementation of a useful design pattern (observer pattern) for bike updates is missing.

- Quality of design and implementation **3%**/5%
  - Your implementation should follow a good design and be of high quality
  - Should include some assertions where appropriate

Javadoc supplied, code is readable

- Readability **5%**/5%
  - Code should be readable and follow coding standards
  - Should supply javadoc comments for Location and DateRange classes

4. Report **10%**/10%

Updated class diagram. Included another controller, the *booking controller*, to handle bookings. Separated controller into two for higher modularization. Included a bike type class to capture the idea of a “bike type”.

- Revisions to design **5%**/5%

  Design document class diagram matches implemented system

  Discuss revisions made to design during implementation stage

- Self–assessment **5%**/5%

  Attempt a reflective self-assessment linked to the assessment criteria