Grading criteria: **Readability**

Grade: 08, 1 – 10

Comment text: < What immediately stands out is where readability is affected by the multiplicity defined in your diagram. In some instances ‘0..\*’ is used to show a cardinality of zero or more, and in some cases no multiplicity is given. My assumption is, that where no multiplicity is defined, the intent is that zero or more is meant to be used. As an example, there is no multiplicity given between the Employee 🡪 Store class, but 0..\* is used between the LineItem and Item class. While this isn’t necessarily incorrect, I believe in negatively affects the readability of this UML diagram.>

Grading criteria: **Validity**

Grade: 08, 1 – 10

Comment text: <According the UML Guidelines page on Canvas, access specifiers should be limited to “public: +, private: -, and protected: # “. It looks like you used ‘/’ in several classes. Additionally, I observed that the Drone class is missing the remaining number of trips remaining before maintenance is required which is a documented requirement.>

Grading criteria: **Fluidity**

Grade: 7.5, 1 – 10

Comment text: <The fluidity of this UML diagram is well done. I think aggregation between employee and store should have been used because these two classes are associated with each other, but one can exist without the other. Moreover, the use of composition seems incorrect between the User and Order class. I think a more accurate relationship would be the use of composition between the Customer and Order class because, without the Customer, an order cannot exist. This relationship appears to be missing. As such, it seems there is a gap in this systems design and the choice of data structures and the data the model.>

Comment #: 1

Course Topic: <UML Goals>

Course Content Reference: UML Summary, Section 1.4 Goals of the UML

Comment text: <According to the above-mentioned material, “The UML must and can support all reasonable programming languages. It also must and can support various methods and processes of building models. The UML can support multiple programming languages and development methods without excessive difficulty.” This UML class diagram does a great job showing a high-level representation of the requirements defined in assignment 1. >

Comment #: 2

Course Topic: <UML Guidelines, Class Relationships>

Course Content Reference: UML Guidelines found in the Pages section of Canvas

Comment text: <According to the above-mentioned material, a classes cardinality should appear on both ends of all relationships. In the UML class diagram submitted, there appears to be a couple cases where this is not done. As an example, the Item class is missing the relationships cardinality to several relationships (LineItem, Store, User). >

Comment #: 3

Course Topic: <Class diagram methods>

Course Content Reference: Page 288 of Applying UML and Patterns – Second Edition. Larman

Comment text: <In this section of Larman’s text, Larman goes into detail regarding the “do’s and don’ts” when creating methods. You do a good job following the recommendation of minimizing/eliminating the getters and setters, thus eliminating the high noise-to-value ratio they generate. However, it is my opinion that some methods are missing. As an example, the Employee class should include a method to update the salary attribute and the Customer class should include a method which updates the customer rating.>