Grading criteria: Readability

Grade: 9, 1 – 10

Comment text: <This UML class diagram is well done and highly understandable. One small aspect I notice, that negatively affects the readability, is the lack of cardinality between the DronePilot and Employee class. As outlined in the assignment requirements, “Drone pilots are employees hired by stores to control the drones as they carry groceries back and forth between the stores and the customer's homes.”. As such, I think showing a 1:1 cardinality between the two is necessary. Without it, it could be inferred that an employee is not a drone pilot.>

Grading criteria: Validity

Grade: 10, 1 – 10

Comment text: <In the UML textbook and UML Guidelines found on Canvas, it is stated that valid access specifiers are public (+), private (-), and protected (#). In the UML class diagram being reviewed, there are a few errors as it relates to access. In the Store class, the revenue attribute has access specified as “-/”. This is also true with the totalCost and totalWeight attributes in the Order class. This is an invalid specifier and looks like an error/oversight in the UML diagram. Having another set of eyes review the final document before submitting may help mitigate this in the future.>

Grading criteria: Fluidity

Grade: 10, 1 – 10

Comment text: <The organization of the design components, including the choice of data

structures and class models, of this UML diagram is very well done. The only area where I would question your choice of data structures is the relationship between the Employee and Store class. My belief is that aggregation should have been used because these two classes are strongly associated with each other, but one can exist without the other.>

Comment #: 1

Course Topic: <Class methods>

Course Content Reference: Page 298 of Applying UML and Patterns – Second Edition. Larman

Comment text: <I note in your system diagram the use of several get() methods. As an example, the Item class has a getCost() and getWeight() method. As it relates to get and set methods, Larman states that “These methods are usually excluded from depiction in the class diagram because of the high noise-to-value ratio they generate; for n attributes, there are 2n uninteresting methods.” I can see why these were used, and there may be some validity to them being there, however, are they 100 percent necessary or do they create unnecessary noise?>

Comment #: 2

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.” An excellent job has been done showing a high-level representation of the requirements defined in assignment 1. >

Comment #: 3

Course Topic: <UML Guidelines, Class Relationships>

Course Content Reference: UML Guidelines found in the Pages section of Canvas and Page 158 of Applying UML and Patterns – Second Edition. Larman

Comment text: <Expanding on the note I made in the Readability section of this peer review, this courses UML Guidelines mention that “Cardinalities should appear on both ends of all relationship(s).” As mentioned, there appears to be a case where this is not done. As a result, the statement Larmam makes, that “The multiplicity value communicates how many instances can be validly associated with another, at a particular moment, rather than over a span of time”, is missed and creates confusion. This lack of clarity creates difficulty when interpreting the relationship between the DronePilot and Employee classes.