Michael Lukacsko: CS6310 Fall 2022 Assignment 1

Class Diagram Analysis Questions

Question 1: How does your design address the need to ensure that drones can lift/carry a new order line?

- The first step in ensuring that each drone can lift/carry a new order line is defining the maximum amount of weight a drone can carry and/or lift. This is done using the "maxCapacity" attribute in the Drone class. Next, when a customer implements the placeOrder() method in the Customer class, a new order is created and will be immediately assigned to a drone for eventual delivery. Once an order has been placed by the customer and assigned to a drone, the order is added to the "orderQueue" attribute within the Drone class and the current weight of the drone is updated and stored using the currentWeight attribute. As such, when the order weight being passed to the Drone class is less than the drones' maximum capacity, it can lift/carry that order. If the new order is accepted, the orderUpdated() method in the Drone class will return True, else it will return False.
- Alternatively, when a customer is going to modify an order by adding items, they can call the addItemToOrder() method. By passing the OrderLines attributes to this method, the customer can update the items and subsequently update the List <> of items. Again, as long as the updated orders orderWeight attribute does not exceed the maximum capacity of the drone, the order will be updated accordingly. To determine this, the value in the currentWeight attribute and updated order weight will be added together. As long as this total is not greater than the value of the maxCapacity attribute, the modification to the order can move forward and the orderUpdated() method in the Drone class will return True, else it will return False.

Question 2: How does your design address the need to ensure that a customer can afford a new order line?

- Like question 1, the first step to ensure that a customer can afford a new order line is defining what the customers maximum credit limit is. This is done in the Customer class using the customerCreditLimit attribute. To determine whether the customer can afford a new order line, the total cost of the order is determined in the Order class using the totalCost() method and is then compared to the value being stored for customers credit limit referenced above. If the orders total is less than the customers credit limit, the process of placing the order and assigning it to a drone proceeds.
- If a customer is modifying an order, the addItemToOrder() method is used to modify an order. If the order does not exceed the assigned drones maximum carry limit, and is under the customers maximum credit limit, it will be added to the drones' order. If not, the order would need to be assigned to a new drone for delivery.

Question 3: How does your design address the need to display the incoming revenue for each store?

- The need to display the incoming revenue for each store is addressed using the Customer, Order, and Store classes. When an order is placed by a customer, the Order class stores the list of items in the orderItems attribute. Finally, the Store class calls the updatePendingRevenue() method and passes the Order class as a parameter. By passing the Order parameter, the value of the order in the orderCost attribute of the Order class is used. Hence, this will calculate the cost of the order by multiplying the cost of each item as listed on that order multiplied by the quantity purchased (cost = quantity purchased * item price).