2.3 User Classes and Characteristics

Class Drone:

Class Members:

- **Drone_id** = Contain the unique id assigned to the drone
- **Status** = Binary variable which tell the status of the drone . If status = 1 then the drone is flying else status=0 drone is at rest.
- **Battery** = The amount of battery in the drone. This will be updated accordingly by drone system.
- **Capacity** = The max amount of capacity the drone can carry.
- **Remaining_capacity** = If the drone is flying caring some amount of load. If it is near to the pickup point then how much more amount of load it will be able to carry efficiently.
- **Current_location** = Current location of drone. Updated by the GPS in drone.
- **Trav_dist** = The amount of distance the drone can travel with the amount of battery it is containing now.
- **Point_dist** = The distance between the pickup point and the drone.
- **Work_done** = W = F*d
    - Where
        - W = Work Done
        - F = force
        - D = distance

    F = mgh + 1/2mg^2
    - Where
        - m = mass
        - g = gravitational force
        - h = height

    Here the Work_Done attribute will contain the value of work_done by the drone to fly or work efficiently without any additional efforts or battery

requirement. Hence this attribute will contain value of work_done at which the battery consumption will be ideal.

- **Weather** = ****

Member Funtions:

- **getDistance()**: This function will calculate the distance between the pick up point the drone current location. The value will be stored in Point_dist attribute. This will done with the help of Location class.

- **getOrder()**: This function will calculate the amount of distance the drone can travel with the current amount of battery. The value will stored in Trav_dist attribute.

- **Capacity()**: Will calculate the amount of load the drone can carry . The value will be stored in Remain_capacity attribute.

Hub/Station:

Class Members:

- **Name** = Name of the Hub or the station.

- **No_of_drone_present** = Number of drone present at that hub or station.

- **Source_Name** = The name of the hub or station from where the supplies is to be pick up.

- **Dest_Name** = The name of the hub or station from where the supplies is to be delivered.

- **Supplies** = The list of supplies along with its amount present at the station or hub. The name of supply as key and its quantity as value in the dictionary.

- **Supplies_status** = The attribute will contain the binary value of weather the hub or the station satisfies the supplies need of order. If the value is 1 then yes else 0 then no.

- **Order_dist** = Dist between Source location and destination location.

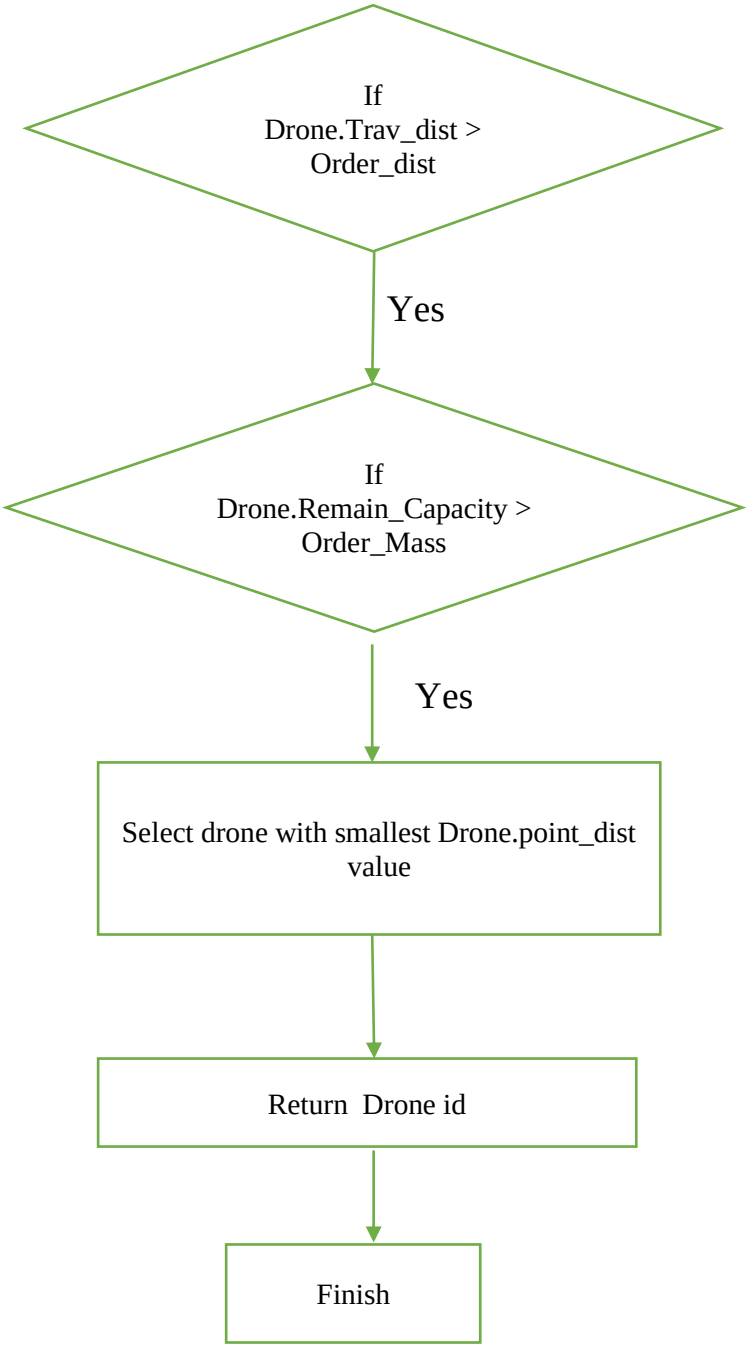
- **Order_Mass** = The mass of the ordered supplies.

Member Funtions

- **getDistance():** Distance between the pick up point and the drop point . This will be done with the help of the Location class .The value is stored in Order_dist attribute

- **getOrder():** This function will get the information about the order like pick up point , drop point , list of supplies , estimated time, etc.

- **Efficient_drone():** This drone will contain an algorithm to get the drone_id of the most efficient drone.



- **setDelivery():** This function will be calling the work done function of the cost_Calculation class. This function will call the work done function by passing all the parameters of the order and the object of the selected drone . The function will then return the cost in return.

- **getMass()**: Calculate the mass of the supplies and store it in Order_mass.

- **suppliesStatus():** Check the availability of supplies as per order requirement.

Cost_Calculation

Class members:

- **Work_done_cost** = the additional cost that we required to due additional work done.

- **Cost** = The overall cost of the order given.

Member Funtions:

- **WorkDone():** The will calculate the additional amount of work done by the drone by considering the weather conditions and other geographical condition and store the value in attribute work_done_cost . It will call the cost_calculation() function within it by passing the additional work_done_cost whose return value will be stored in cost attribute . This function will return the value of cost attribute .

- **Cost_calculation(work_done_cost):** Return the value of Cost
  Where
  Cost = battery_cost + maintenance_cost +work_done_cost

(*Location: This will be done using python libraries* .)