1MAE502 - OBJECT ORIENTED SOFTWARE DEVELOPMENT

Documentation

on

Modeling of the Software for a Drone Delivery System of Parcels

2020-2021

Authors

Neel Chakravarty

Wendi Ding

1. Assumptions and Design Decisions

The drone delivery system is built based on the following assumptions:

- The warehouse takes charge of preparing the parcels, checking the status of drones and send the drones to recharge or maintain if necessary, allocating the parcels to the appropriate drones for delivery, allocating the task way points for drones, and setting the drones delivering parcels. All operations are only made for orders with status NEW, which means that the orders are paid.
- A customer can only place an order on the shopping panel after having been authenticated with a login and password. Otherwise, the system will pop up to login.
- Each customer is linked to exactly one account. Personal information is stored in Customer class, while user's information is stored in Account class.
- The shipping address is represented by an array of integer {x,y}, the horizontal axis position and the vertical axis position, simulating the longitude and latitude respectively.
- Each order is defined by a unique reference number which is randomly generated by the system. An order can contain different kinds of products with different ordered numbers for each product. While ordering a product, the system will check if the quantity ordered is available else, it proposes to reorder the quantity (inventory).
- An order can be delivered in one or more parcels, depending on the total weight of ordered products. Since each delivery must be less than 5kg, if the total weight exceeds 5kg, the order will be separated into several parcels, and more than one drones will be allocated to ship the parcels of the order.
- Drones use GPS waypoints to fly from the warehouse to one shipping address, or from one shipping address to another, traversing easily difficult terrain and choosing shorter routes. Collisions are not considered, and each drone flies with a constant speed around 10km/h.
- Round trip distance to the departure point (warehouse) must be less than 50 Km. Drone's battery have to be charged after each 10 hours of fly, and losts 1% of its charge for each 1km travelled. A drone can carry at most 5kg of payload. If all conditions are satisfied, a drone can deliver several light parcels in one round trip.
- The weight of small things like mouse, headsets, headphone is assumed like 0.1 kg, the parcel weight is assumed as 0.2 kg. For simplicity, the parcels are cubes and the size of parcels is defined by the sum of volumes of all ordered products.
- A 10 % discount for the first client order on the order total purchase amount is offered, and so is a 15€ discount for the tenth client's order. The delivery cost is free if the total amount of the order is greater than 100€.
- Only the list of products given in the table below are listed in the database of the drone delivery system:

Name	Category	Marque	Price	Delivery cost	Quantity
Flash drive 128Go	Computer Accessories	Philips	20€	3€	100
Headphones	Phone Accessories	Philips	12€	5€	55
Headsets	Phone Accessories	Samsung	35€	6€	120
Bluetooth headsets	Phone Accessories	Samsung	55€	6€	59
Computer Mouse	Computer Accessories	HP	30€	3€	15
Wireless keyboard	Computer Accessories	Apple	130€	5€	25
HDMI to VGA cable	Computer Accessoires	Apple	22€	6€	34
USB-C to VGA adapter	Computer Accessories	Apple	54€	0€	14
Hp adapter	Computer Accessories	HP	80€	10€	60
Mouse pad	Desktop Accessories	Boulanger	5€	1€	37
Ethernet Cable	Computer Accessories	Boulanger	9€	3€	46

2. Realization of the System

In order to display the round trip of a delivery while having only one customer, the shipping address is modified by the system automatically every time a new order is placed (which is set in line 340 of class GuiShoppingPanel and can be commented when not using). The realization of the drone delivery system is shown in the pictures below:

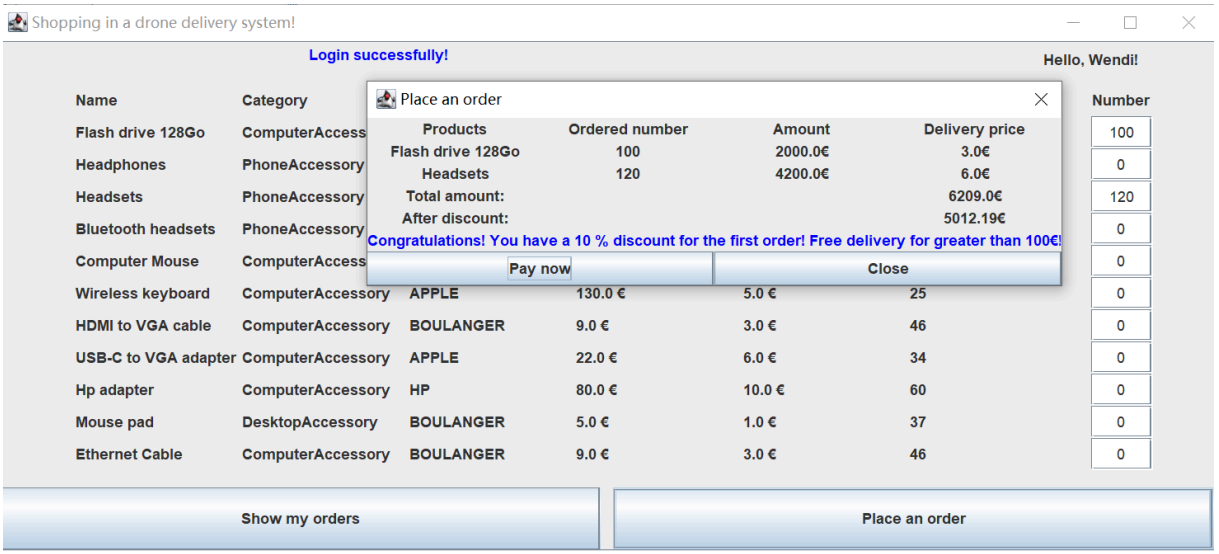

Figure 1: Having a discount for the first order

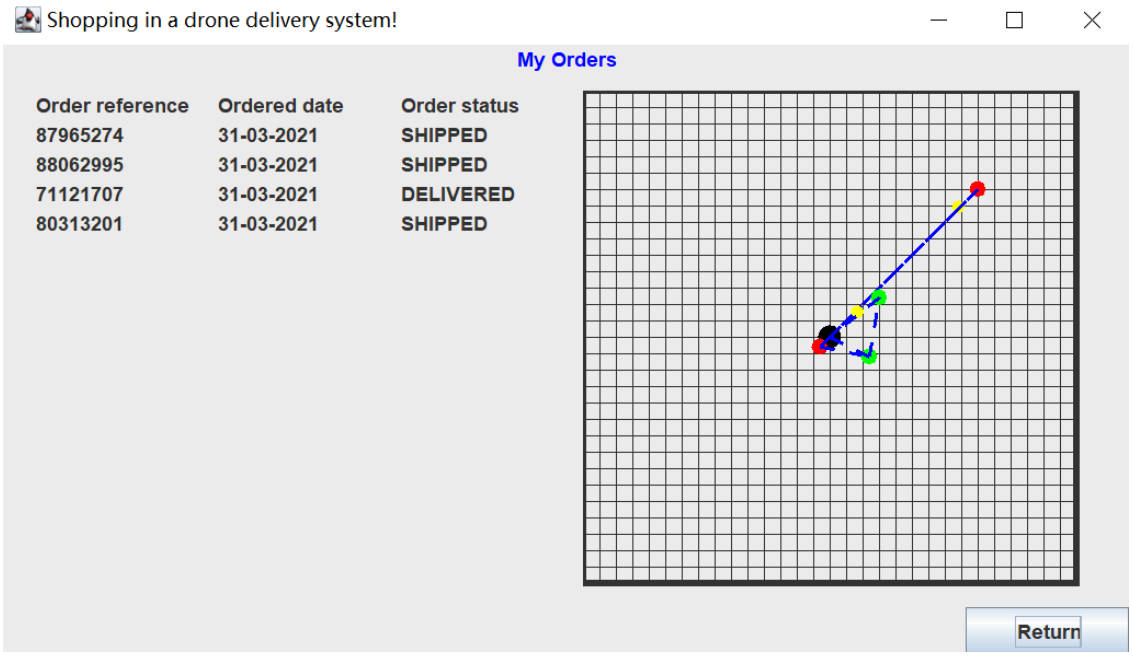

Figure 2: Drones delivering orders

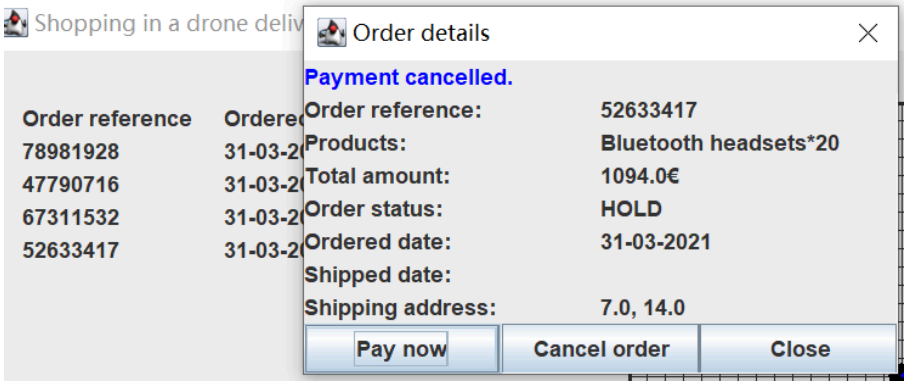

Figure 3:Pay for a hold order

```
I'm a parcel for order 87965274, my weight is 2.2
I'm a parcel for order 88062995, my weight is 0.30000000000000004
I'm a parcel for order 71121707, my weight is 0.5
I'm a parcel for order 80313201, my weight is 0.7
Trip distance:42.42640687119285
```

```
Drone 2101: I'm delivering a parcel from order 87965274
Drone 2102: I'm delivering a parcel from order 71121707
Drone 2101: I'm delivering a parcel from order 88062995
Drone 2102: I'm delivering a parcel from order 80313201
Drone 2102: I have delivered a parcel from order 71121707
```

```
Drone 2102: I have delivered a parcel from order 80313201
Drone 2101: I have delivered a parcel from order 87965274
Drone 2102: I have returned!
Drone 2102: battery charge = 84, remaining flight hours = 7.625999999046327
Drone 2101: I have delivered a parcel from order 88062995
Drone 2101: I have returned!
Drone 2101: battery charge = 40, remaining flight hours = 3.9185999870300297
```

Figure 4: Drones delivering orders gives information

3. Changes made to UML Class Diagram

Classes Customer & Account: The two classes, Customer and Account have been interchanged, as the account has an id and a username, not the customer. It also makes more sense for the account to store details such as the billing and shipping addresses. One attribute has been added to Customer (previously Account) - myOrders, which is an ArrayList of type Order and stores all the previous and current orders made by the customer. Two attributes have been added to Account (previously Customer) - iban and cvc. These attributes have been added as they store the payment information about the customer. Each time a payment is made, the cvc is verified in order to validate the payment.
A function decreaseNumberOfOrders() has been added to the class Account (previously Customer) to decrease the number of orders in case an order is canceled.
Two functions have been added to the class Customer (previously Account) - addAnOrder() and deleteAnOrder(). As their names suggest, their function is to add new orders and to delete canceled orders.

Class Parcel: Added a new attribute - status. The attribute status is of type ParcelStatus, which is a newly defined enumerated type. It has 3 possible states - *NEW, SHIPPED & DELIVERED.*
The class Parcel is now associated with the class Warehouse. There is only one warehouse, and it has multiple parcels, the amount of which is not limited.
The association between Parcel and Drone has also been changed. Now an order need not necessarily have only one parcel. If the order size is large, an order is distributed into multiple parcels. However one parcel can only contain the items of one order.

Class Order: The attribute quantityOfProducts has been removed, and it is replaced with two new attributes, orderedProducts (ArrayList of type Product) and orderedQuantities (ArrayList of type integer). These two attributes allow greater flexibility, as they now let us store a list of the ordered products without predefined length and correspondingly the list of the number of those products which have been ordered, since one order can contain several products. Two more attributes have also been added, namely - orderReference and totalAmount. The orderReference stores the reference id of the order. The totalAmount stores the total amount to be paid for the order, after discount (if any).
A new function calculateDiscount() has also been added, to compute the discount which is given for the first order by a customer (10%), and for the tenth order (15%). It also removes the delivery cost for any order over 100 euros.

Class Drone: Added 2 more attributes - speed and totalFlightHours. The attribute speed stores the speed of the drones, which is assumed to be 10kmph. The second attribute totalFlightHours was added to keep track of the total number of flight hours of each drone. We are making the assumption that after 100 flight hours, a drone must be sent for maintenance. Once maintenance of the drone is completed, its total number of flight hours is reset to zero.

Drone is now associated with Warehouse and with Parcel. There are five drones in the warehouse. Each drone can have several parcels, but one parcel can only be associated with one drone.
The name of the function followTaskWayPoints() was also changed to updateDroneGps() to better reflect the operations being carried out by this function. This function calculates and continuously updates the current GPS coordinates of the drone, thus mapping out its trajectory from the warehouse to delivery points, and back.

Class Warehouse: Added a new attribute guiShoppingPanel. It is of the type GuiShoppingPanel. This attribute is used to obtain the list of new orders from the class GuiShoppingPanel, in order to allocate them to parcels for packing.
Warehouse is now associated with Drone and GuiShoppingPanel. There is one warehouse, and it has five drones, with their id numbers ranging from 2101 to 2105.There is only one GuiShoppingPanel for the warehouse.
The return type of the function getNewOrders() was changed to an ArrayList<Order>, as the type of the parameter being returned (newOrders) was also changed to an ArrayList, so that new orders can be added easily and canceled orders can be removed. This was proving problematic because of the restrictions of the array type. The same change is made to the type of the parameter newOrders in the function prepareParcels() - the type is changed to ArrayList<Order>. Finally, the function deliverOrder() was removed and replaced by two new functions: arrangeParcelsToDrones() and shipParcels(). This was done to better divide the large number of operations taking place in one function into two separate and distinct functions. The function arrangeParcelsToDrones() does exactly what its name implies, it sorts out the new orders into one or several parcels (depending upon the size and weight restrictions). These parcels are then assigned to one or several drones, once again according to the weight restrictions of the drone (5kg). While allocating the parcels, this function first checks whether the drone is free and has enough battery remaining to complete the round trip for the deliveries. If it does not, then it sends the drone for recharging and looks for another drone. Else, if the drone requires maintenance (total flight hours exceeds 100), then it sends the drone for maintenance.
shipParcels() is a simple function with a simple task, it sends the parcels for delivery once they have been assigned to a drone.

Class GuiShoppingPanel: The class Software is changed to the class GuiShoppingPanel, which is composed of several subpanels in order to communicate with the customer and has several methods, public or private, to perform operations to the database and to call actions to the associated warehouse, including deliver(), showFrame(), placeAnOrder(), showOrders(), promptLogin(String), promptLogOut(), addAnAccount(Account), addAProduct(Product) and deleteAProduct(Product). All these new methods ensure the switches between different GUI subpanels as well as necessary information convey. And the subpanels: GuiMainMenu, GuiPlaceAnOrderWindow,GuiShowOrdersPanel, GuiLoginPanel, GuiStartShoppingPanel, GuiDisplayPanel and GuiOrderDetailWindow together with GuiShoppingPanel composes the software part of our drone delivery system. Details of the software part can be seen in the comments in java codes.

Class GuiShowOrdersPanel: This class aims at creating a frame to display all the information concerning orders of the current customer. The method promptOrderDetail() pops up the order detail dialog which allows the customer to see the details and shipping status of an order, as well as to pay for a hold order.

Class GuiLoginPanel: This class makes use of its constructor to pop up a dialog for the customer to login, and wait for the customer's reaction to do further operations. After the customer clicks "confirm" button, the id and password typed in will be read and compared with the database of ArrayList of Accounts to check if they are correct.

Class GuiDisplayPanel: This class extends JPanel and implements ActionListener, in a bid to realize animation of drones delivering orders. Attributes include final values of panel_width and panel_height, the current customer, and a counter and a timer are implemented to flash the drones at a predefined time interval of 1 second. The GPS positions of drones in the panel are refreshed every 0.1 second, and the shipping address turns from red to green once an order is delivered.

4. User Manual

The only executable class with static public void main() is the testDroneDeliverySystem class. All the product information and customer's accounts are set in the testDroneDeliverySystem class and can be modified. Upon running the code, the first menu we are presented with is the main menu.

Main Menu: The main menu displays two options: one to place a new order and the other to exit. Upon selecting the option: "New Order", a new panel appears, asking for your name. Any name can be entered here, this name will be displayed in the shopping panel.

Shopping Panel: Upon selecting the option "Start Shopping" after entering your name, you will be presented with the shopping panel, containing a list of all available products and their current inventory. On the top right, the name you had previously entered will be visible. On the bottom, there are two options available, "Show my orders" and "Place an order". However, both these options only work once you have logged into your account. If you click on either of these options without logging in, you will be prompted to login with your account credentials first.

Login: There are 2 ways to login, the first method is by clicking on your name which is displayed on the top right, and the second is by clicking on either of the two options on the bottom, as mentioned before. We have defined two accounts with which you can login in. Their credentials are:

Account 1 (Wendi) :

username: **dwd**

password: **dwd**

cvc: **000**

Account 2 (Neel) :

username: **Neel**

password: **1234**

cvc: **123**

For logging in, we are only asked to enter the username and password. The cvc is required later during payment.

Once you have logged in, you can place an order, once you have entered the quantity next to each product which you want to order. Upon pressing the “Place an order” button, a payment panel will pop up, displaying the cost of the products and the total amount to be paid. Upon clicking the “Pay now” option, you will be prompted to enter your cvc. If the cvc is correct, the payment is validated and the order will be placed. The payment window can be closed by clicking the cancel button.

If the payment is not validated, then the order is placed in hold status. You can click “Pay Now” to try the payment again, or you can go to the “My orders” screen, where the order status is displayed as on hold. By clicking on the order, you can initiate the payment again.

Drone Trace and My Orders: Once the payment has been validated, you must click on the “My orders” option to start delivery of your order from the warehouse and to see the drone trace. The target is displayed in red, before the package is delivered, and it turns green once delivery is complete. The drone is displayed in yellow as it moves across the grid.

Apart from showing the drone trace, the “My orders” page also shows a list of your previous and current orders, along with their status and reference numbers. By clicking on the order, you can see the details of the order and have options to pay for or cancel a hold order.

**The order status shown on the “My orders” page does not keep refreshing continuously. However every time you move your cursor over the grid area, the status of the order is refreshed immediately. Besides, by clicking on the grid area, you can also initiate a new round of delivery from the warehouse.

You can press the “Return” button to return to the shopping panel and place a new order if you wish, and go back to the “My orders” page to see the drones delivering packages.

Log out: When you have logged in, you can click on the name label to log out of the account.