Restaurant Supply Express! Drone Delivery

CS 2340: Objects and Design
Course Project: Summer 2022 Semester

Version History

Version	Date	Notes
0	May 17, 2021	Initial release

Problem Domain Description

The following is a text description of the system you are being tasked to develop. The system requirements – explicit and implicit – are included this document and need to be reflected in your various design artifacts. You have been tasked to design an object-oriented system (to be implemented in Java) that meets these requirements and provided these functional and non-functional capabilities.

You've just arrived with a group of friends to your favorite restaurant. You're excited to enjoy your favorite dish – however, as you begin to place your order, you're informed that they're missing a key ingredient. Well, that's disappointing. Wouldn't it be nice if there was a way to get that key ingredient delivered quickly so that your craving could be fulfilled?...

…Well, wait no longer. You are being asked to design and develop a system to help manage the quick and efficient delivery of key ingredients to restaurants to help them keep their operations – and their customer's appetites – running as smoothly as possible. This system will represent one or more ingredient delivery services. Each delivery service will manage a fleet of drones. Those drones will be used to deliver the ingredients to the restaurants as quickly as possible. Restaurants will purchase ingredients from the drones. The money paid for the ingredients purchased will flow from the restaurant owners back to the drone owner's delivery service, helping each stay in business – a proverbial "win-win" situation.

The system that you have been tasked to design and implement will monitor the activities of various drone delivery services that are delivering ingredients to restaurants. Your system will support a simulation of the various types of entities – associated people, places, and things – relevant to our scenario. It will also need to be capable of executing functions that will maintain the state of the simulation accurately: updating the state as required when valid requests are entered and displaying appropriate error messages when invalid requests are made by the system's users.

Our scenario includes people acting in various capacities. For example, there are numerous employees in our scenario. Your system must maintain the first name, last name, address, and birthdate of each person in the system. We will use the birthdate to help authenticate them for tax verification purposes, and occasionally to reward them with special offers (e.g., customer rewards). And to avoid confusion, all persons must have a distinct username in this system.

Some people in the system manage and/or work for delivery services. Each delivery service has a distinct identifier, and a (possibly duplicated) longer name. Each service also keeps track of the revenue it has earned from delivering orders successfully. Services are supported by various employees. Employees can support the company in different roles – for example, by stocking and monitoring the warehouse or moving packages to the proper drones (i.e., warehouse workers); or, controlling swarms of drones (i.e., pilots). One of the more experienced employees is normally also selected to be the sole manager for the service.

We must track the unique tax identifier (e.g., Social Security Number for some people) for each employee for legal purposes. We will also keep track of the date each employee was hired, along with the number of months (as an integer) that they have worked (i.e., experience) for the company. We keep these separately because employees sometimes take a leave of absence to train, attend conferences, or support the family (e.g., maternity leave). We must also keep track of the salary for each employee.

Pilots control the swarms of drones as they carry ingredients back and forth between the different restaurants. Each pilot must have a valid license to signify that they have received the proper training to operate the drone safely. Each license will have a unique ID for tracking purposes. Flight skills also tend to improve with experience, so we must also keep track of the number of successful trips (e.g., moving drone from the current location to a new location) for each pilot.

The warehouse workers are like the people you often encounter in grocery stores stocking shelves; serving in the deli, bakery, or seafood section; or acting as a cashier. The only difference is that they work in the warehouse for the delivery service as opposed to being directly visible to the customers. They are responsible for making sure that the drones are repaired, restocked, and refueled, and otherwise ready to deliver ingredients as needed.

Employees might be temporarily unemployed, though they normally work for at least one delivery service. Workers can also work at multiple services as part of a "time-flex" plan. Pilots, on the other hand, might be unemployed, but (to avoid conflicts of interest) can’t be employed by more than one delivery service at a time. Each delivery service can employ multiple warehouse workers and pilots and must have a manager.

Each delivery service will have an identifier and a name. The service identifiers will be unique, though names might be shared. Managing a service implies that you must also be employed by that service. And an employee may be the manager for at most one service at a time – it’s simply too much work to manage multiple services simultaneously. The job of manager is also too busy for piloting or warehouse worker duties.

Delivery services can purchase many drones to deliver orders to restaurants in a timely manner. Each drone has been purchased/sponsored by a single service, and is used to make deliveries (i.e., carry ingredients for) that service alone. Also, each drone will have an identifier that is unique for that service, but not necessarily unique across the other drones in our system; therefore, each drone must be identified relative to the specific delivery service that it supports.

And given new flight control technologies, a pilot can control multiple drones at the same time. Some drones take their flight directions directly from a pilot. Other drones get their flight directions from another drone that is being controlled directly. These drones act as a “swarm” that work together to move safely to the same location. The drones in a swarm must move to the same location or stay where they are – always together.

Drones need to be repaired, restocked, and refueled periodically. Each drone has a limited fuel supply where moving from restaurant to restaurant consumes fuel based on the distance between the different locations. We must keep track of the drone’s fuel supply, as well as its current location. Each location is a combination of (X, Y) coordinates, where X and Y are positive or negative numerical values. And the system must be able to track the money that has been earned from ingredient sales for each drone.

We will track the drone using a “discrete state simulation” approach where we only keep track of the drone’s current location – we don’t track the points during transit. Therefore, the only valid locations for a drone are at a restaurant, or at its owning delivery service’s “home base.” The home base is the only location where the drone can be refueled, repaired, and restocked with new ingredients. And the new flight control technologies also ensure that a drone will move to a location when, and only when, it has enough fuel to get from that location back to home base. Also, each location also has a space limit on the number of drones that can hover and land safely in that location at the same time, so a drone can move to a certain location only if there's enough space for the drone to maneuver safely.

A drone is configured so that it can only carry a certain number of fixed-sized packages of varying weight. The fixed-size packages facilitate easier loading of the drones. Each drone has a limited capacity of packages – the total weight (measured in pounds) of the ingredients is important but is not a limiting factor. A drone may be controlled by at most one pilot at a time. Having multiple pilots attempting to control a single drone would eventually lead to conflicts and crashes.

Delivery services offer lots of different ingredients, where each ingredient has a 'universally' identifying barcode, a name, and a unit weight measured in an even (i.e., integer) number of pounds. The unit weight is equivalent to the amount of the ingredient that will fit in one of the drone's fixed-size packages. Restaurants that wish to purchase ingredients can purchase them from a drone, but only when that drone is located at the restaurant. When a drone is at the location of a restaurant, the system should be able to provide the quantity and price per package for each of the ingredients being carried by the drone.

Also, a specific ingredient might be offered by many different services, and the inventory of possible ingredients is so large that we are not interested in tracking which specific services sell which specific ingredients. Our goal is simply to ensure that each ingredient identified by a specific barcode has a consistent name and unit weight for tracking purposes.

The managers of the delivery services will use their collected data and experience to send the drones to the restaurants that they feel are most likely to buy those ingredients. Part of that collected data includes a rating for each restaurant. The rating is an integer from one (1) to five (5), where a higher number indicates that the restaurant has been more reliable in purchasing ingredients over time. We must also track the restaurant's money as the number of dollars that they have to purchase ingredients in the future. Each restaurant has a distinctive name. Also, some restaurants are funded by an owner, while others are "independently managed" (e.g., no owner).

Whenever a restaurant purchases an ingredient from a drone, the money from the restaurant must be transferred to the revenue of the appropriate delivery service. Our system must be able to track the total amount of money that has been spent for all the ingredients that a restaurant has purchased so far. It also must be able to calculate the total amount of money earned by each delivery service through all these sales. And there are no "IOUs" or "promises to pay later" – a restaurant must always have enough money to purchase ingredients before the transaction can be completed. Note that each delivery service can sell a fixed-size package of an ingredient at a price of its choosing, and these prices can change over time (e.g., supply and demand effects).

Finally, the system must be able to calculate and display the total weight and cost of the ingredients being carried by each drone (i.e., the drone's payload). The system must also be able to calculate and display the total cost of the payloads for all the drones that it owns. The system must also be able to calculate and display the debt incurred for each owner as the sum of all the money spent buying ingredients from the restaurants for which they provide funds.