Food delivery system

A food delivery system must be designed and implemented. The company that requires this system collects orders during one afternoon and delivers the food during the next morning. This company only delivers food to specific areas. The company would like a client-server system that consists of the following components:

1. **A tool used to create orders (client side). The tool has the following features:**
   - Create a new order:
     - fill in order information (name, location, phone number, comments)
       - the list of possible locations is retrieved from the server
     - add deliverables (food and/or beverages)
       - choices are retrieved from the server
       - filters should be available on various attributes (i.e. vegetarian food only or alcoholic drinks only)
     - set number of portions per deliverable (default to 1)
     - select packaging type – choices are retrieved from the server
   - Preview order
     - contains all the information on all the products, including packaging, and the total price of the order
   - Submit order to server

2. **A tool used to manage orders (server side)**
   - Monitors incoming orders – the user of this tool can see a list with all the orders received so far
   - Provides a feature to generate the shortest possible path to reach all the customers and deliver the goods (at the first launch of the application, the user will be asked to select the location of the headquarters on the map). Once the route is computed, the user can remove the delivered orders from the list (all the orders that were included in the generated route).

A. Deliverable details

There are 3 types of deliverables: Food, Beverage and Packaging. In the client application the user will be allowed to filter by deliverable type (Food or Beverage), but Packaging is mandatory, the user will only select the desired Packaging type (see below for the possible values).
The list of deliverables available on the server side can be read from a file (i.e. text file, XML).
All deliverables have the following common attributes:

- name
- price
- quantity (per portion)
- measurement unit (g/ml/piece)

In addition to these, Food deliverables have:

- vegetarian (y/n) – can filter by
- diet (y/n) – can filter by
- ingredients

Beverages have:

- carbonated (y/n) – can filter by

- alcoholic (y/n) – can filter by

On top of these 2 attributes, Cocktails also have the ingredients attribute.
Packaging can be recyclable or not.

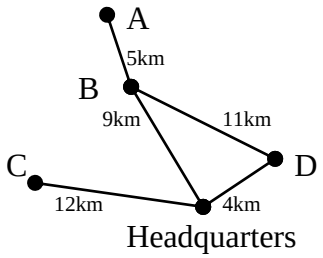
B. Order details

Each Order has the following attributes:

- Deliverables (the items selected in the order creation tool)
- Address (location - place on the map - see below for details)
- Name
- Phone number
- Comments

C. Map details

The system will use a simplified map that contains the locations serviced by the company, together with the distances (measured in km) between the locations. If one location cannot be reached directly from another location, there must be an alternative route through another location on the map (in the picture below, A can't be reached directly from the headquarters, you need to go through B to get to A).
The configuration of this map can be read directly from a file (i.e. text file, XML).
Here is an example of a valid map:



Optional features:

- Validate the map at the start of the server application – i.e. warn if the map contains a location that can be reached (directly or indirectly) from the headquarters.
- When generating the delivery route, orders that arrived earlier should be favored in terms of delivery time
- Considering that the company acquired several delivery vehicles (this number will be configurable in the server side tool), update the delivery route generator to provide the number of vehicles used and the shortest route for each one in order to deliver the food as quick as possible.
- Implement the feature to add deliverables to the server side tool or in another standalone application.
- Create a tool to generate maps.

Note:

For the tools, we would prefer a GUI made with Swing.
The way the communication between the server and the client is implemented is not imposed.
Your architecture must be easily extensible.
When you write the applications take into consideration that:

- a new deliverable type could be added later
- new attributes could be introduced for a given deliverable type