IT-SDJ3, Course Assignment, Autumn 2022

Case

Your task is to simulate the operation of a nearby slaughterhouse. In one end of the slaughterhouse living animals arrive and in the other end various products, consisting of one or more animal parts, depart. The slaughterhouse has three stations where smaller parts of the whole process take place.

1. At the first station, the animals arrive, are weighed and registered.
2. At the second station, the animals are cut into smaller parts. Each part is weighed and registered, including a reference to which animal it comes from.
   The parts are put into trays with each tray containing only one type of parts. Each tray has a maximum weight capacity.
3. At the third station, products are packed for distribution. One product might be a package with a number of the same kind of parts intended for repackaging in supermarkets. Another kind of product might be "half an animal" where all the expected parts are included, but not necessarily coming from the same animal. All products are registered including references back to the trays the parts came from.

If it is later discovered, that there is some kind of trouble with a slaughtered animal, it should be possible to recall all products, which might contain parts of the animal. This function should be accessible outside the slaughterhouse.

Project part 1

This part is about making the above description more precise and designing the architecture of system.

Hand in:

- A domain model of the slaughterhouse
- An architectural overview of the system. Focus on the types of communication (direct/indirect etc.), not technology. This does not need to be UML it can be a rich picture or similar.

Project part 2

Implement a registration system for the animals as a RESTful web service. The service should store date, weight, registration number and origin (i.e. farm) of the animal. It should be possible to read the details of a specific animal, all animals arriving at a particular date, and all animals of a particular origin.

The service doesn't need a graphical user interface, but it must be tested.