DDD Case Study: Restaurant Management System

In this case study, you will work on a restaurant management system for the franchise chain "Delicious". The brand plans for a large number of restaurants, all working with the same concept and workflows. Your software will be provided to all franchises.

Disclaimer

This description is just intended as a starting point for your design and implementation process. It is not a living document – i.e. after the initial Event Storming workshop, it will not be updated. It is most probably incomplete, and may contain inconsistency. In the workshop and afterwards, we can jointly decide to deviate from this written version.

The description deliberately tries to avoid making assumption about a (backend) software architecture. It is clear which clients exist, but otherwise we will only talk about "the system". One of your tasks will be to determine appropriate subsystems, or (in DDD terms) sub-domains and bounded contexts.

How to visit the restaurant

Guest can use the **Delicious App** to search for a nearby restaurant, select it by clicking, and then book a table. They need to specify the day and time, and the number of persons. The app then either confirms the reservation and displays the table number, or asks the customer to select another time or day, or change the number of persons. Guests can only reserve timeslots that are more than 2h in the future.

Alternatively, the guests can just walk in, without a reservation. Each table displays its number and the reservation state on a tablet attached to the table. This tablet can display advertisements and other information, see below. A reservation will block the table for 90 minutes from other reservations. If there is a no-show (i.e. no food is ordered from this table), it is put back on the "free table" list 30 min after the original reserved time.

If guests sit down at a free table without having a reservation, the table is marked as "taken" once the guests have ordered food (see below).

How to order food

There are no waiters in the restaurant. The only way to order food at a table is via the Delicious App. The guest can each order (and pay) their own meal, or one guest orders and pays for the whole group.

A guest first needs to scan the QR code shown on the table's tablet (denoting the table number). Then he or she selects the food items from a "menu catalog". He or she then pays via credit card in the app. The app then displays an estimated waiting time. This waiting time is updated every 1 min. The backend system responsible for taking the food orders tries to level out the waiting time on each table, so that the whole group gets their food at (roughly) the same time.

When the food is ready, the guest's Delicious App displays a notification and a QR code. The table's tablet shows a blinking sign. The guest needs to fetch the food him/herself from counter at the kitchen. The cook scans the QR code and hands out the food to the guest.

How to order drinks

Drinks are ordered without any digital processes at the bar, from a human bartender. Drinks are paid cash or by credit card at the bar's cash register.

The bars are outsourced to a different company, which operates independently. Therefore the whole aspect of "drinks" can be treated as out-of-scope for this restaurant management system.

How the food is prepared

For cost reasons, the cooks at Delicious restaurants don't have (or need) much training. The kitchen is organized into a number of "cooking stations". Each station contains a stove, the necessary cooking utensils, and a space for used pots / spoons etc.

(The used utensils are cleared regularly by a kitchen help, who puts them into the dishwasher and brings clean pots etc. to the cooking stations. This process doesn't need any digital support, and is not modelled in the restaurant system.)

Above each cooking station, there is a large monitor. When the cooking station receives a food order, the corresponding recipe is fetched and displayed in easy steps on the monitor. The cook working at this the station fetches the ingredients from the adjacent storage / cooling room, and starts cooking.

The cook has a dedicated "**Delicious Cooking App**" on his/her smartphone. At the beginning of the cooking process, the cook presses "start". After each recipe step, the cook acknowledges the step in the app. (These information will go into the waiting time estimation for the guest's app.). When handing over the food, the cook scans the guest's QR code with his/her cooking app. This is also the sign for the system that the cooking station is free to receive the next preparation order.

How cooks check in and out at a cooking station

The cooks use their cooking app to check in or out of a cooking station (the stations are numbered). They check out for breaks, and at the end of their shifts.

How the supply is managed

The ingredients in the storage / cooling room are managed outside this restaurant management system. The manager will estimate the requirement amount of ingredients per day, and makes sure the storage is adequately stocked.

The manager also monitors the stock manually during the day. If, despite his planning, stock for an ingredient gets low, the manager will manually deactivate the affected food items in the restaurant's menu catalog. He uses an **Admin Client** to do so. This action marks the food item as "temporarily unavailable" in the guests' Delicious App.

How personnel is managed

The Admin Client allows to enter new cooks into the system, so that they can log into the Cooking App. The duty rota for the cooks are managed manually, outside the restaurant management system.

How food items and recipes are managed

The Admin Client allows to add new food items with their recipes using a dedicated editor. Food items can also be disabled or deleted.