Must Haves

* The system must be a food ordering system
* The use must be able to create an account
* The user must be able to specify a pickup time for their order
* The user must be able to indicate allergies
* The system must warn the user of allergens in a pizza
* The system must have the default pizza selection (e.g. pepperoni, margherita…)
* There must be a set of configurable toppings for pizzas
* The user must be able to filter out pizzas with their allergens
* The user must be able to edit their order/basket
* The respective store in the chain must be notified of a completed order
* The users must pass an authentication process before using the system
* The user must have a member ID (must be unique string)
* The ID must be associated with the user account and type of subscription
* The accounts must have credentials (ID and password)
* The passwords must be stored safely

Should have

* The system should store coupon codes
* The codes should have an activation key (4 alphabetical, 2 numeric)
* The users should be able to submit an activation key
* The restaurant should be able to configure the coupons to be available for a limited amount of time
* We should have 2 coupon types: buy one get one free, x% discount
* The system should allow for adding more coupon types in the future (working with certain combinations of products)
* The system should apply the cheapest coupon redeemed
* The system should only apply one coupon per order
* The user should be able to cancel their order up to 30 minutes before selected time
* The user should not be able to edit the order from this point on
* The store should be notified on a cancellation

Could have

* The system could offer an admin role (regional manager) to see all the currently placed orders
* The admin could have the option to cancel a given set of orders from the currently placed ones
* More coupon types

Won’t have

* GUI

Non-functional:

* The system must be built such that it can be extended with extra functionalities (modular)
* The system must allow easy integration with other systems (APIs)
* Individual components must be scalable (microservices)
* The system must be written in Java 11
* The system interactions must happen through APIs