Here is an overview of the expected architecture:
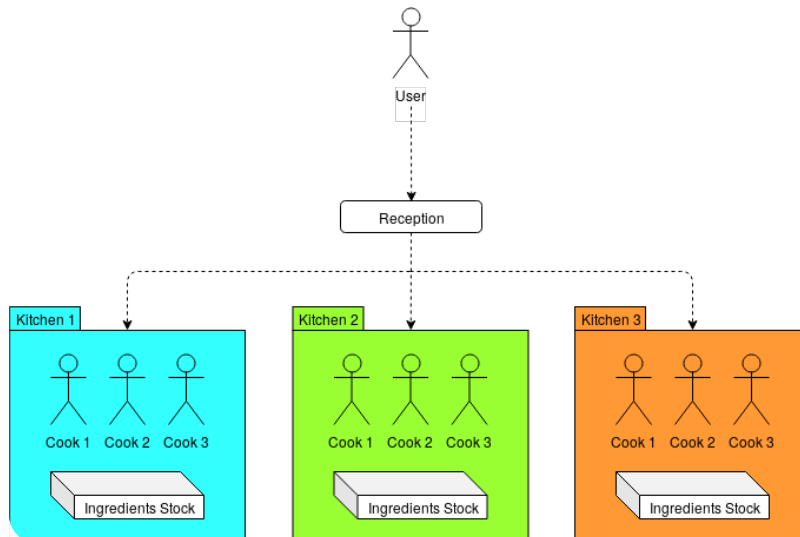

The reception

The reception must be started using the command line the following way:

```
▽                              Terminal                          –  +  ×
B-CCP-400> ./plazza 2 5 2000
```

✓ The first parameter is a multiplier for the cooking time of the pizzas. It is used to examine your program more easily, so it must **INEVITABLY** be implemented. Otherwise it will not be possible to grade you. Moreover this parameter **MUST** be able to accept numbers with value in between 0 to 1 to obtain a divisor of the pizzas cooking time… *Cooking time is detailed later*.
✓ The second parameter is the number of cooks per kitchen. **Cook definition is detailed later.**
✓ The third parameter is the time in milliseconds, used by the kitchen stock to replace ingredients. **Ingredient definition is detailed later.**

The reception **MUST** be an interactive shell with at least the following actions:

✓ Commands of a pizza by the user though command line, for example "`regina XXL x7`". **This will be detailed later.**
✓ Displays the kitchens status, including the current occupancy of the cooks, as well as theirs stocks of ingredients. using the `status` command.

{EPITECH}

> In the `bonus` directory you may add graphical version of the reception, that'll be a great bonus to make!

Pizza ordering MUST respect the following grammar:

```
S      := TYPE SIZE NUMBER [; TYPE SIZE NUMBER]*
TYPE   := [a..zA..Z]+
SIZE   := S|M|L|XL|XXL
NUMBER := x[1..9][0..9]*
```

Ordering example which is grammatically valid:

```
regina XXL x2; fantasia M x3; margarita S x1
```

> It is not because the grammar is very simple that your parser may be too basic! spits, as wall as other hacks, must definitively be avoided...

The reception **MUST**:

- ✓ be able to place more orders when the program is running. The program MUST be able to adapt.
- ✓ allocate pizza by pizza to kitchens when receiving an order.
  When all the kitchens are saturated, it MUST create a new one (do a fork as explained later.)
- ✓ always allocate pizza to kitchens so that the occupancy is as balanced as possible. You must not have one kitchen with all the pizzas and the others not doing anything!

When an order is ready, the reception MUST display the information to the user and keep a record. (A log file on top of other displays should be a good idea...)

{EPITECH}