Programming – “Pretty Hair” #03

*The following exercise will focus on improving your coding skills. In order to do so, we will put TDD on hold and focus solely on the coding itself. If, however, you really like TDD or you feel confident in your coding, then keep using TDD throughout this exercise as well. If, on the hand, you are getting nowhere with TDD then start coding the old fashioned way – BUT – you still have to make unit tests for all your code, only difference is that you can do them afterwards. Oh! One more thing: Forget about BDD for now, we can always come back to that later!*

We are going to continue with implementing part of the system for “Pretty Hair” which you have worked on previously, and as usual:

* Continue the already existing Pretty Hair solution
* Break down the Use Cases into tasks
* Implement at least 8 unit tests for each Use Case (either before or after).
* Implement the source code.

## Exercise 1: 180 minutes (est.)

Solve the following exercise using Boss-Secretary/Chef-Sekretær:

At this point, you have already implemented the Use Case: Process Order. However, since we are building an entire system anyway; why not make it an automated system?

We will go back and change Process Order to work event based so the warehouse manager won’t have to check which orders are ready to be processed; instead he will be advised in a FIFO manner.

Let’s revisit the use case: (changes in **bold**)

**Process order**

(brief)

Every day the *warehouse manager* ~~checks the system for orders that need to be processed~~ **receives an email for each order that needs to be processed**. The system shows an overview of the day’s orders, that need to be processed, ~~and then the~~ *~~warehouse manager~~* ~~chooses an order from the overview~~. **The email includes** ~~system shows~~ the order with the product type id, description, and quantity for each product type. The *warehouse manager* then packs the order and registers it in the system, which automatically adjusts the remaining amount of the effected product types.

(casual)

When (**the warehouse manager gets his email**) and the system presents the overview of orders, to be processed that day, the system also checks whether there is an adequate amount of the product type in stock to fulfill the orders that should be packed. If there is an order where this is not the case, the order will not be shown in the overview, but the system will send an email with the order id, product type id, description, quantity on the order line, and the registered amount in stock, to the purchaser

**Events will occur in the following places:**

* When an order is ready to be packed, an event should be fired, and an email should be sent to the warehouse manager.
* If the amount of the product type is inadequate, this should fire an event and an email should be sent to the warehouse manager.
* When the warehouse manager registers the order as packed, this should fire an event to adjust the remaining amount of the effected product types.

**Extra info:**

In the cases where emails are sent, Eventargs should include the content for the email

The email system should be mocked.