

# Sprint Retrospective, Iteration # 3

User Story #	Task #	Task Assigned To	Estimated Effortper Task (in hours)	Actual Effortper Task (in hours)	Done (yes / no)	Notes
Report	Write Chain of Responsibility Section	Anna Lantink	45 min	45 min	Yes	For the report the chain of responsibilities design pattern has to be explained and reasoned about, which I wrote.
	Create UML diagram for Chain of Responsibility	Matyas Kollert	30 min	30 min	Yes	
	Improving the previous UML diagram for strategy design patter	Kyongsu Kim	20 min	20 min	Yes	The previous uml diagram did not contains a correct way of naming.
Testing	Authentication Test	Anna Lantink	4 hours	7 hours	Yes	The authentication microservice needed to be more fully tested, so that was added
	Menu Test	Matyas Kollert	3h	3h	Yes	
	Order Testing	Javier Paez Franco	5 hours	11 hours	Yes	I tested the whole order microservice. I also refactored some of the code to make testing easier, and fix a few bugs I found while testing.
	Task 3					
Strategy Design Pattern	Implementing strategy design pattern	Anna Lantink, Kyongsu Kim	3 hours	c	Yes	We finalized implementing the strategy design pattern in the coupon microservice.

Editing/Updating Issues in Gitlab	Rewriting Issues in Gitlab	Anna Lantink, Matyas Kollert	2 hours	2 hours	Yes	We were advised to add more detail, such as done criteria, to the Gitlab issues. We ended up rewriting many of the issues, and adding more detail to them.
Allergens	Add validation for allergens	Matyas Kollert	2h	2h	Yes	Allergens must be numbers between 1 and 14
Chain of responsibility pattern application	Implemented the chain of responsibility design pattern in coupon application	Bogdan Iordahe	3h	7h	Yes	Took a bit longer then expected because I had to do it over a couple of times, it is however implemnetd well as of right now.
Order	Fixed HTTP bug where the returned HTTP status of the response was sometimes incorrect	Javier Paez Franco	2h	2h	Yes	
	Finished updating Get Order(s) methods	Javier Paez Franco	30 mins	30 mins	Yes	
	Started refactoring the Order microservice	Javier Paez Franco	5 hours	5 hours	No	I started refactoring the Order microservice, because the ModifyOrderService is full of duplicate code, weird naming, etc., and bad style conventions in general.
	Modify Order Functionality	Kristof Levay	3 hours	6 hours	Yes	
	Canceling Order functionality	Kristof Levay	3 hours	5 hours	Yes	
	Notifications	Kristof Levay	3h		No	

--	--	--	--	--	--	--

Project: SEM\_\_\_\_  
Group: 04a\_\_\_\_

# Main Problems Encountered

## Problem 1: Debugging

Description: A lot of us were producing tests and updating parts of the code in parallel, which caused some problems because sometimes tests would pass in one pipeline, and not in another's pipeline. This could get a little annoying because we'd have to continuously check and update MRs.

Reaction: This is a relatively minor error and can be solved with some better coordination on our part. Next time, we should make sure that we don't edit code and write tests for that code in parallel, or if we do that, we pull from the development branch frequently.

## Problem 2: Following the Schedule

Description: For the most part, everyone followed the schedule we laid out in the beginning of the sprint accordingly. However, as can be reflected in the issues that weren't closed, there was an instance where one person couldn't finish the notifications MR in time. We didn't want to rush the coding or review process, because we didn't want to risk introducing bugs in our code, so we don't have notifications in our system for canceling/creating an order.

Reaction: The main problem here was that we could not contact the person responsible for this MR. Attempts at sending reminders went unheard. We will try to bring this up in future meetings and work out a resolution.

## Adjustments for the next Sprint Plan

*Motivate any adjustments that will be made for the next Sprint Plan.*

I think in the future we will have another meeting about time management, to ensure that every member can follow the schedule we have set out, and maybe this will prevent delays from happening in the future. As for the debugging problem, we will have more meetings in the future to coordinate this even better, and we can lay out a specific schedule to help us with who does what when.