

# Seminar 1

Object-Oriented Design, IV1350

Amanda Koinberg, amakoi@kth.se

2021-03-31

# Contents

1 Introduction	3
2 Method	4
3 Result	5
4 Discussion	7

# 1 Introduction

This seminar concerns the course objective “*interpret and clarify a given specification by applying established guidelines for analysis*”.

To achieve this objective, the students are asked to, through analysis, create a domain model and a system sequence diagram that displays a real-life scenario.

The teacher’s concept of Analysis is referred to in the course literature (ch. four) “A First Course in Object Oriented Development” by Leif Lindbäck, and the students must follow the modelling steps and guidelines when creating their diagrams.

## **Assignment 1**

Make a domain model for a retail store. In the current iteration we are implementing both the basic flow and the alternative flows described in the Process Sale. The specific requirement specifications are described in the assignment.

## **Assignment 2**

Draw a system sequence diagram (SSD) illustrating basic flow and alternative flow of Process Sale, who's requirement specifications are described in the assignment.

## 2 Method

To do a domain model, step one is to do a noun identification as described in the literature. Every noun in the task specifications is treated as potential class candidates. Once that is done, step two is to try to come up with more class candidates by going through a noun category list.

After spending a good set of time to freely and creatively coming up with all potential classes, it is time for step three and four - to critically remove classes that we think are useless and to convert classes that better serve as attributes to other classes.

When task three and four seems to be completed, it is time for the final step - to draw associations between classes and potentially numerical values if it makes the diagram clearer.

I, personally, have no problem with task one and two, however I find it more useful to jump back and forth between three and five. Sometimes I might even realize that I need to add even more classes during this process.

### 3 Result

The solution of my domain model is shown in figure 3.1 which shows the relationship between classes. I noticed that place of sale (POS) and receipt became central classes but did my best to avoid a naïve or spider-in-the-web model. I believe that several of the associations I initially drew became attributes, Reciept and StoreInfo is an example.

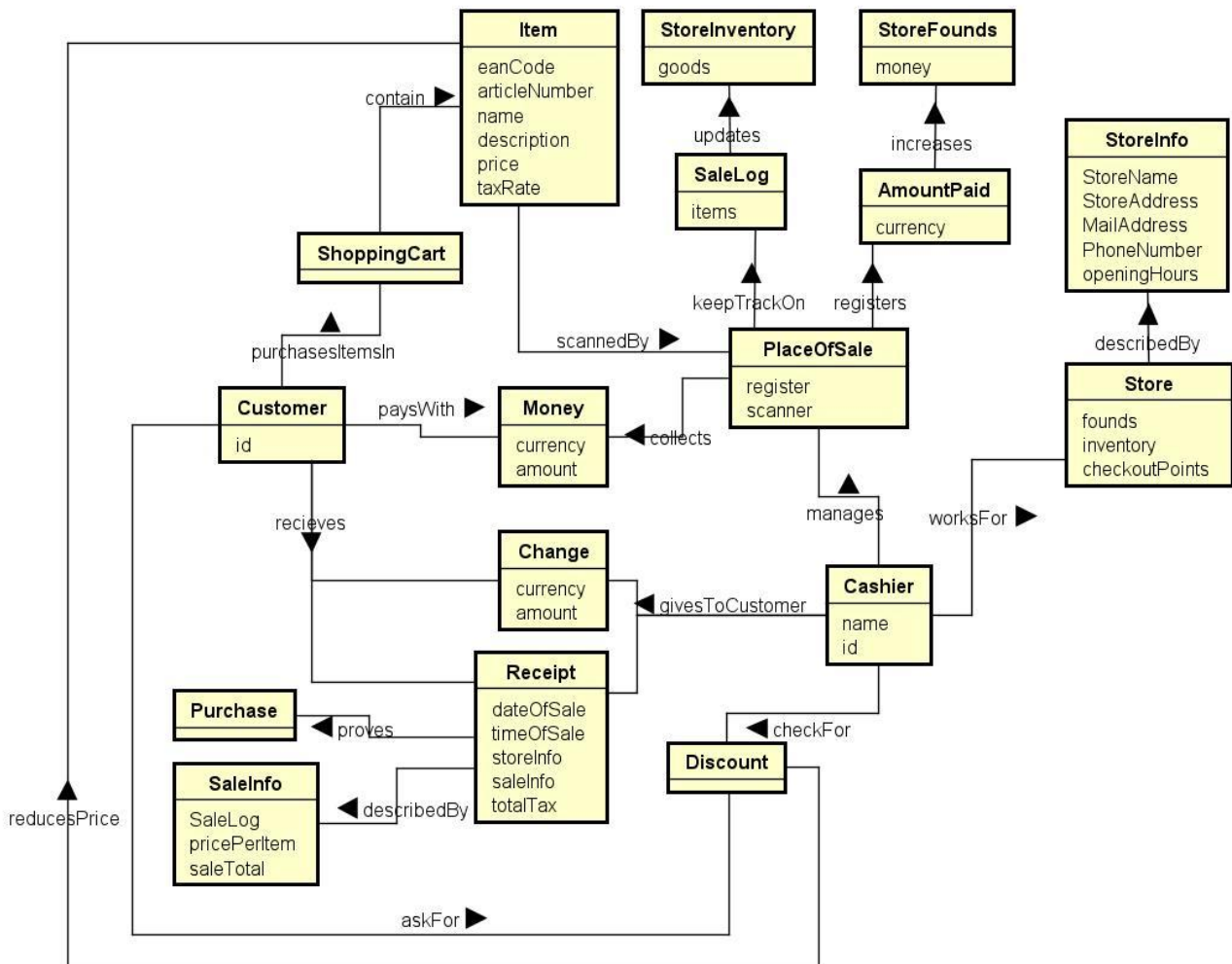


Figure 3.1: A domain model for a purchase taking place in a local store.

As for the system sequence diagram, I visualized the cashier, the place of sale (POS: where individual cashiers operate the store's general system) and the store's general system. The general system would in real life keep track on several things, but in our model, it will only describe inventory and funds. The solutions to this assignment can be found in figure 3.2. It is the cashier that operates the register and communicates through it to reach the general system where things such as item details, prices and tax rates, and inventory details can be read from and written to. The POS will ask the general system for data, but it will also keep track of all items scanned locally. It will not update the general system until the cashier has received payment and stores it into the register.

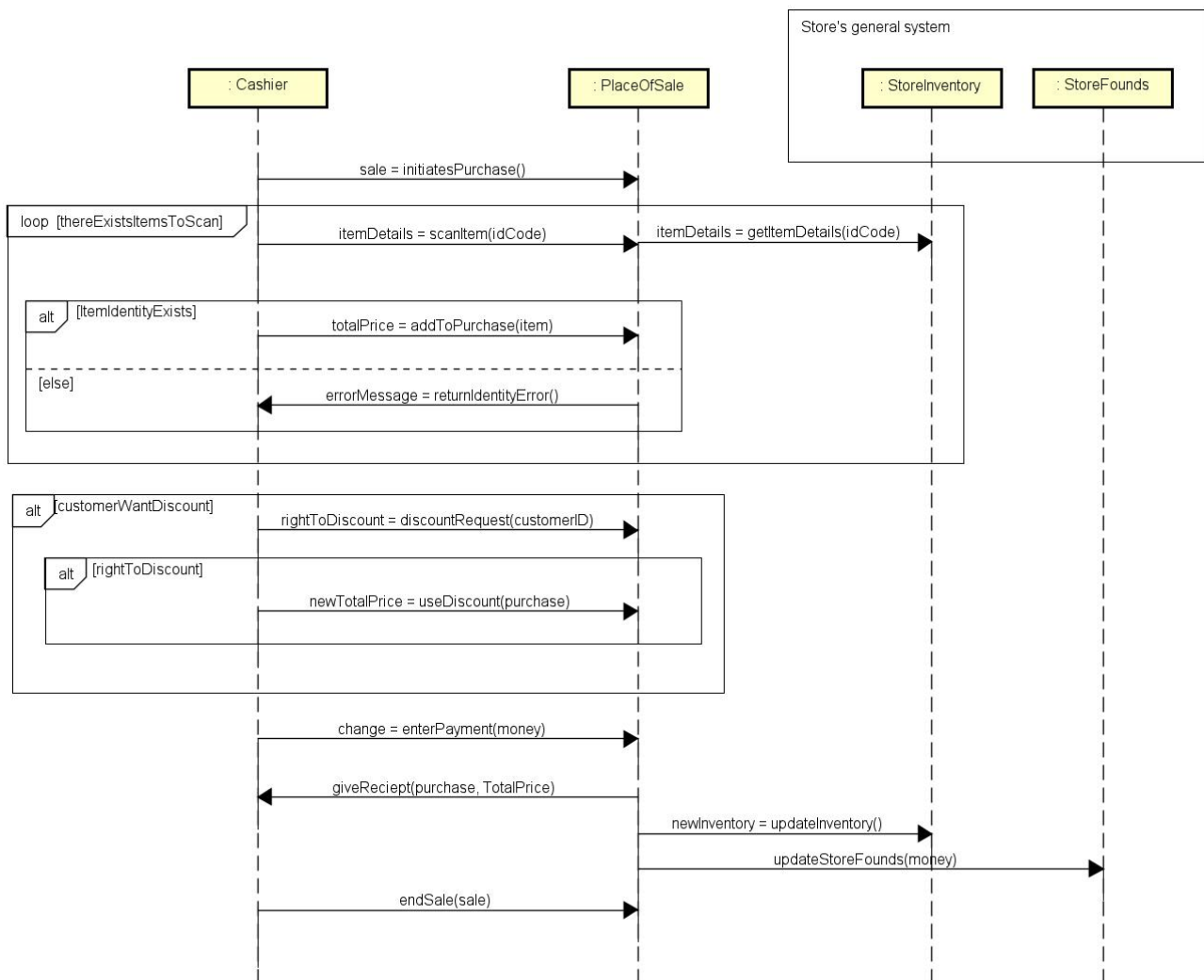


Figure 3.2: A system sequence diagram describing the interaction between cashier and store systems

## 4 Discussion

The point of this seminar is to get familiar with UML and learn how to model a real-life scenario. My opinion about chapter 4 in the course literature is, except for learning how to model is also what to avoid, specifically naïve, programmatic, and spider-in-the-web models. I personally do not think that my DM is either if those.

I tried not to make too many associations, and only keep the ones that gave the diagram meaning. I am not sure I succeeded entirely. As for class choices, I think that these are the most relevant ones for the model.

I am very happy with how the SSD turned out. I think it gives a real description to how registers work in real life, even though it only covers the requests described in the assignment.