

# **Seminar 2**

**Object-Oriented Design, IV1350**

**Adrian Kastrati, [kastrati@kth.se](mailto:kastrati@kth.se)**

**2022-04-12**

# Contents

<b>1 Introduction</b>	<b>X</b>
<b>2 Method</b>	<b>X</b>
<b>3 Result</b>	<b>X</b>
<b>4 Discussion</b>	<b>X</b>

# 1 Introduction

The seminar task is to design a program in an object-oriented programming language that can handle a purchase of items at a store. The design should have high cohesion, low coupling and good encapsulation divided into layers according to the MVC and layer patterns. Discussions of the task with Deni Persson, Jesper Munkeby and William Eriksson.

## 2 Method

The program design progress started with adding the interactions from the SSD in order. When an action in the SSD resembled a potential method between a class in the domain model it was taken into consideration for how it could be handled to minimize the connection to other classes. All while following the design process from lectures where the principles of “high coherency” and “low coupling”. After this, the design was revised to see where possible changes for the better could be made.

# 3 Result

The finished design had 10 different classes with one Data Transfer Object where needed. The classes are used in six different layers where it handles different aspects of the intended program.

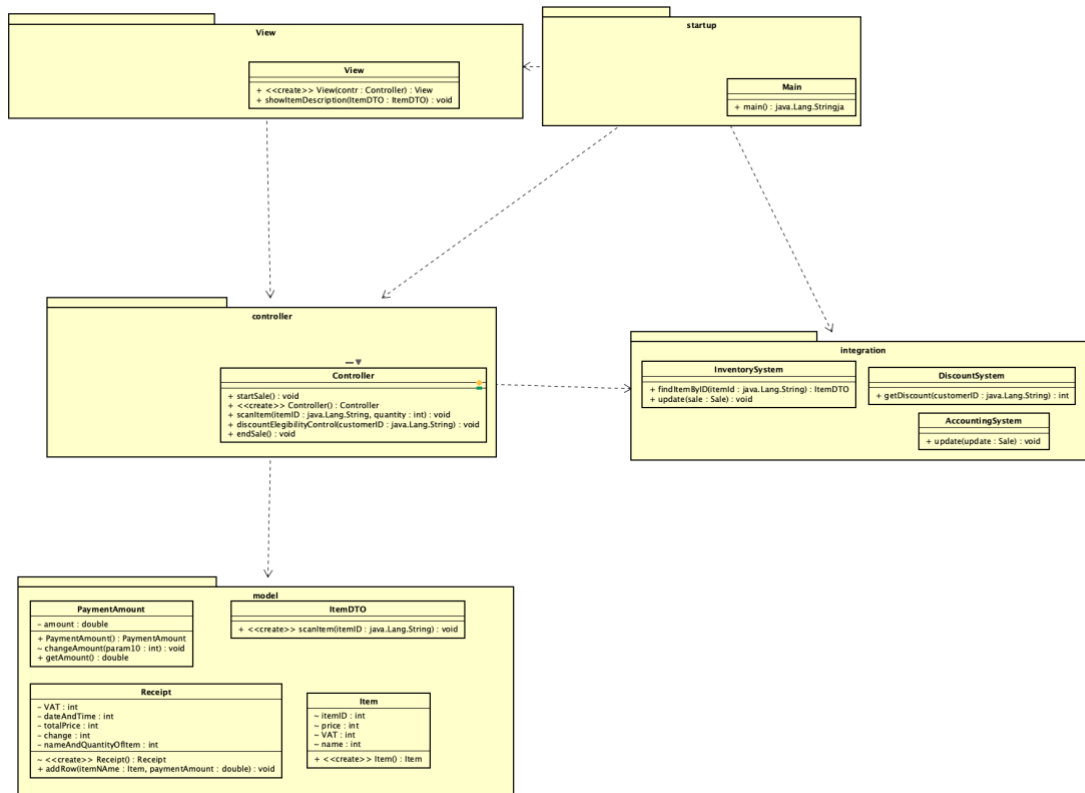


Figure 3.1 complete class diagram

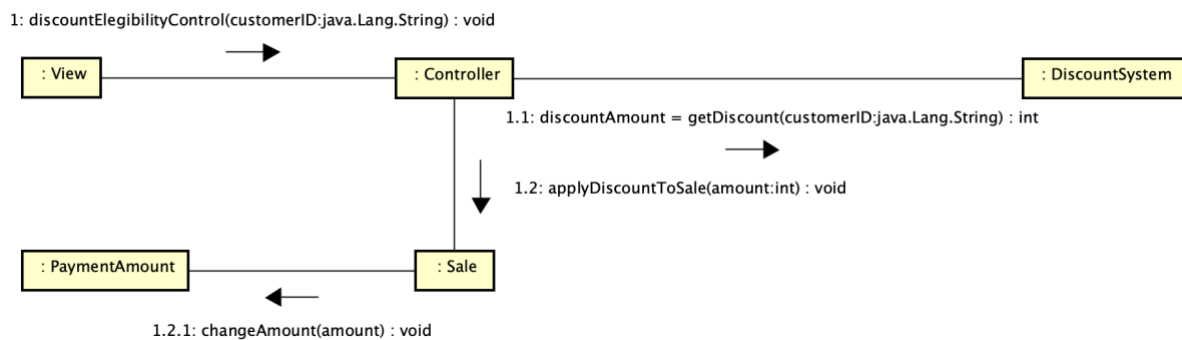


Figure 3.2 discount layer

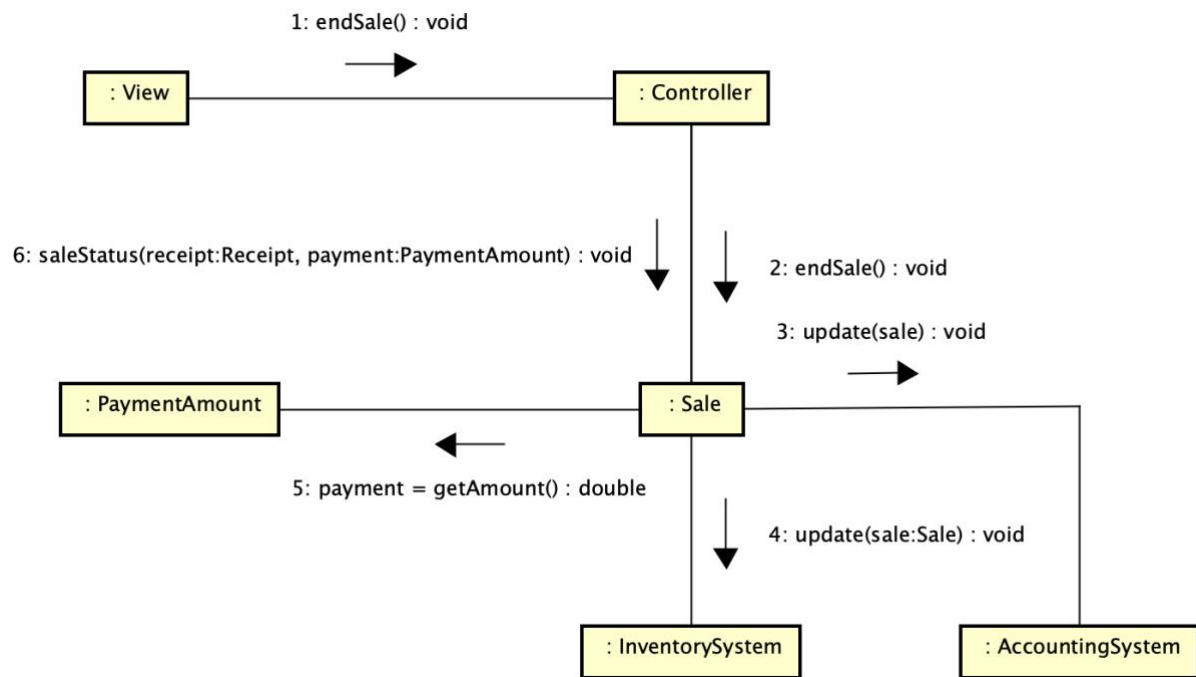


Figure 3.4 paymentLayer

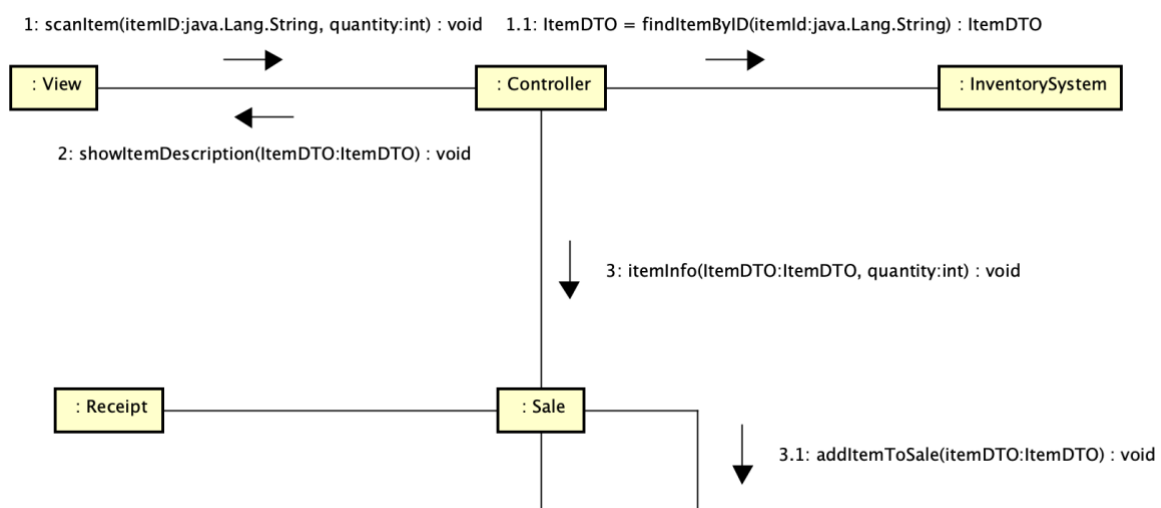
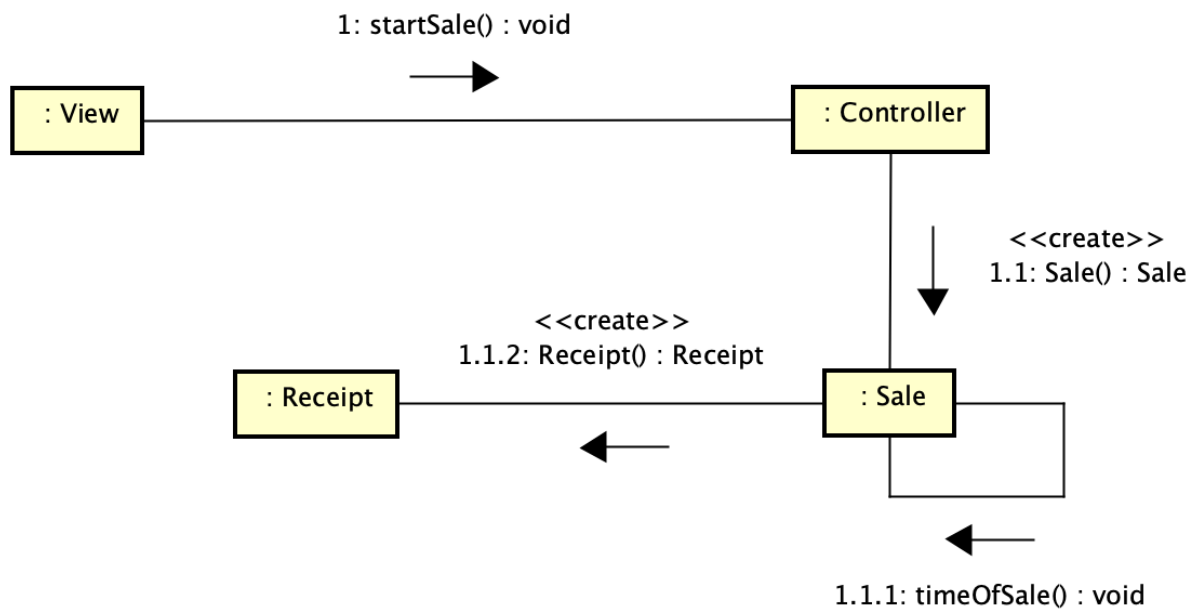
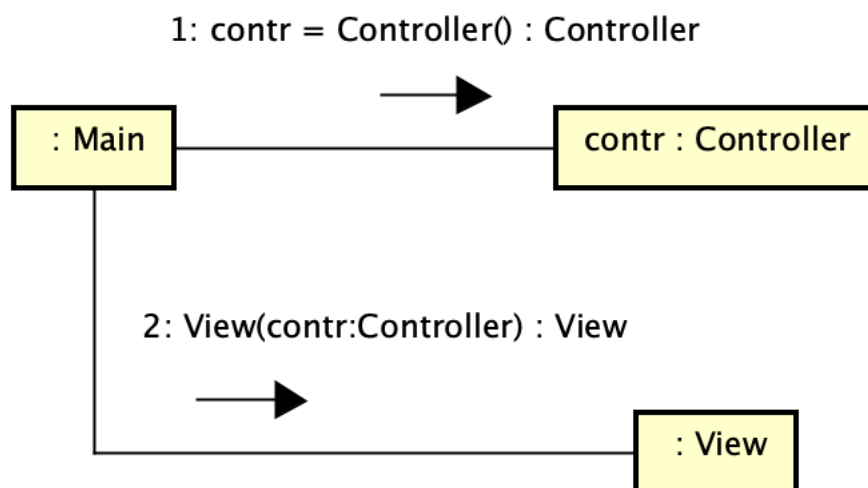


Figure 3.5 scanItem layer

**Figure 3.6** startSale layer**Figure 3.7** main layer

## 4 Discussion

The payment layer could have had more classes or methods in it but I came to the conclusion of just using one layer to alter the paymentAmount after each item and using a different layer for discount since it takes the discount information from a database which alters the paymentAmount at the end and not during the sale. The scanItem layer is used for what changes during the entering itemID and how they are connected, which is why it only updates the receipt and places items in the sale object. The saleStatus object could be made into a DTO since its use is just to transfer data to the View class for presenting.