

Seminarium 2

Niloofar Rahmani, [nrahmani@kth.se](mailto:nrahmani@kth.se)  
June 9, 2021

## Innehåll

<b>1 Introduktion .....</b>	<b>3</b>
<b>2 Metod .....</b>	<b>4</b>
<b>3 Resultat .....</b>	<b>5</b>
<b>4 Diskussion .....</b>	<b>9</b>

## 1 Introduktion

Seminarium 2 handlar om grunden i objekt orienterad design och att skapa ett program baserad på scenariot i seminarium 1. Fokus i detta seminarium är att programmet ska vara tydligt, med Low Coupling och High Cohesion. Jag samarbetade med Alva Ols och vi använde oss av föreläsningar och kurslitteraturen.

För komplettering har vi använt oss av feedback som vi fick av läraren och peer review.

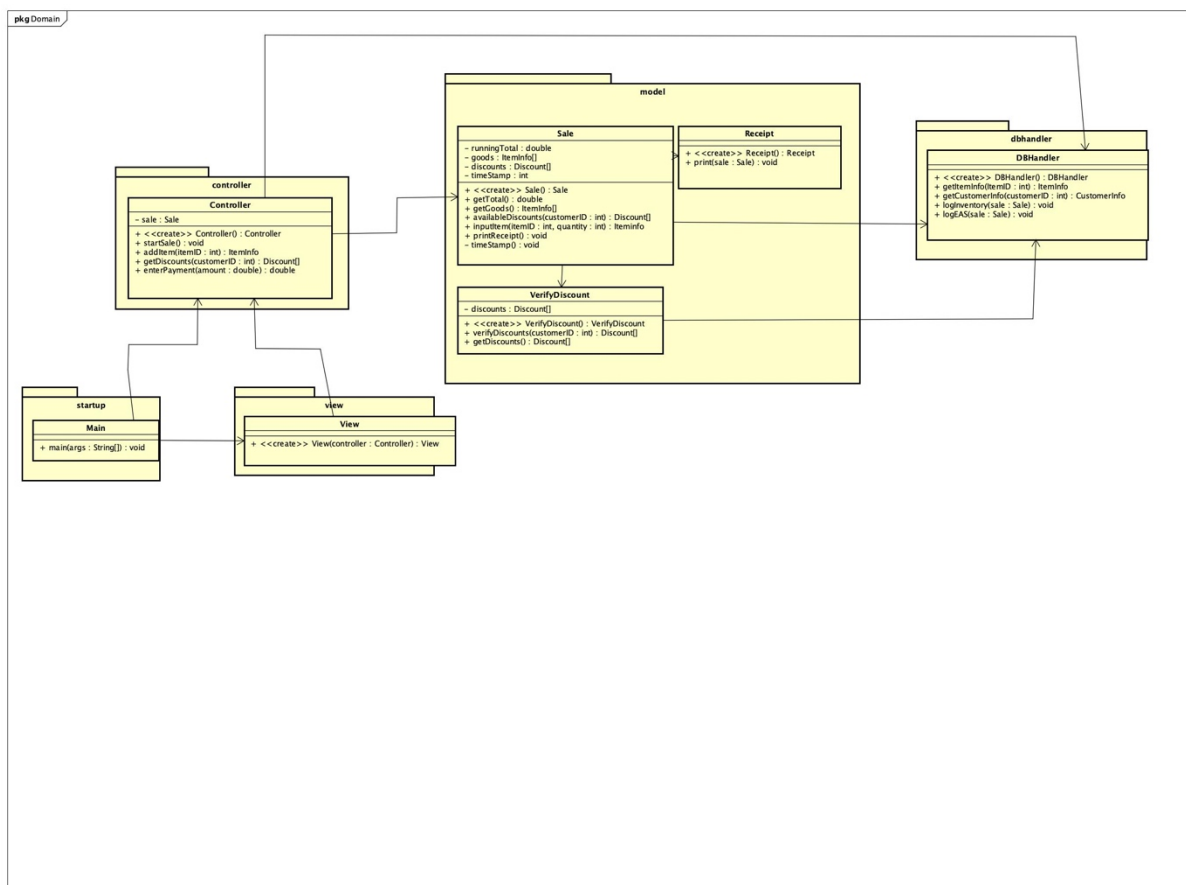
## 2 Metod

Vi började med att diskutera och skapa olika paket som vi kunde börja med och de var view, controller, startup och model. Sedan la vi till även dbHandler. Efter detta började vi med att skapa kommunikation diagram för de metoder som kallades i SSD. Detta blev till slut startsale, startup, scanItem, scanItemInvalid, discount och payment. För att lösa detta använde vi oss av exemplet som visades på föreläsning 7 och i kurslitteraturen. Vi skapade olika klasser och metoder först och sedan fixade vi deras namn så att det blev så tydligt som möjligt. Till slut fixade vi paketens olika klasser, fixade även några attribute i de klasserna, associationer och dependency mellan klasser och mellan paket.

För att komplettera vår gamla design läste vi genom vår feedback och använde kurslitteratur och internet för att få bättre förståelse. Vi ändrade det mesta till den nya versionen vi har nu och för att skapa dem gick vi genom ganska samma metod som beskrivs ovan. Men denna gång visste vi om våra brister. Parametrar skapades och de alternativa flöden som saknades skapades. Vår numrering var helt fel föra gången och det fixades.

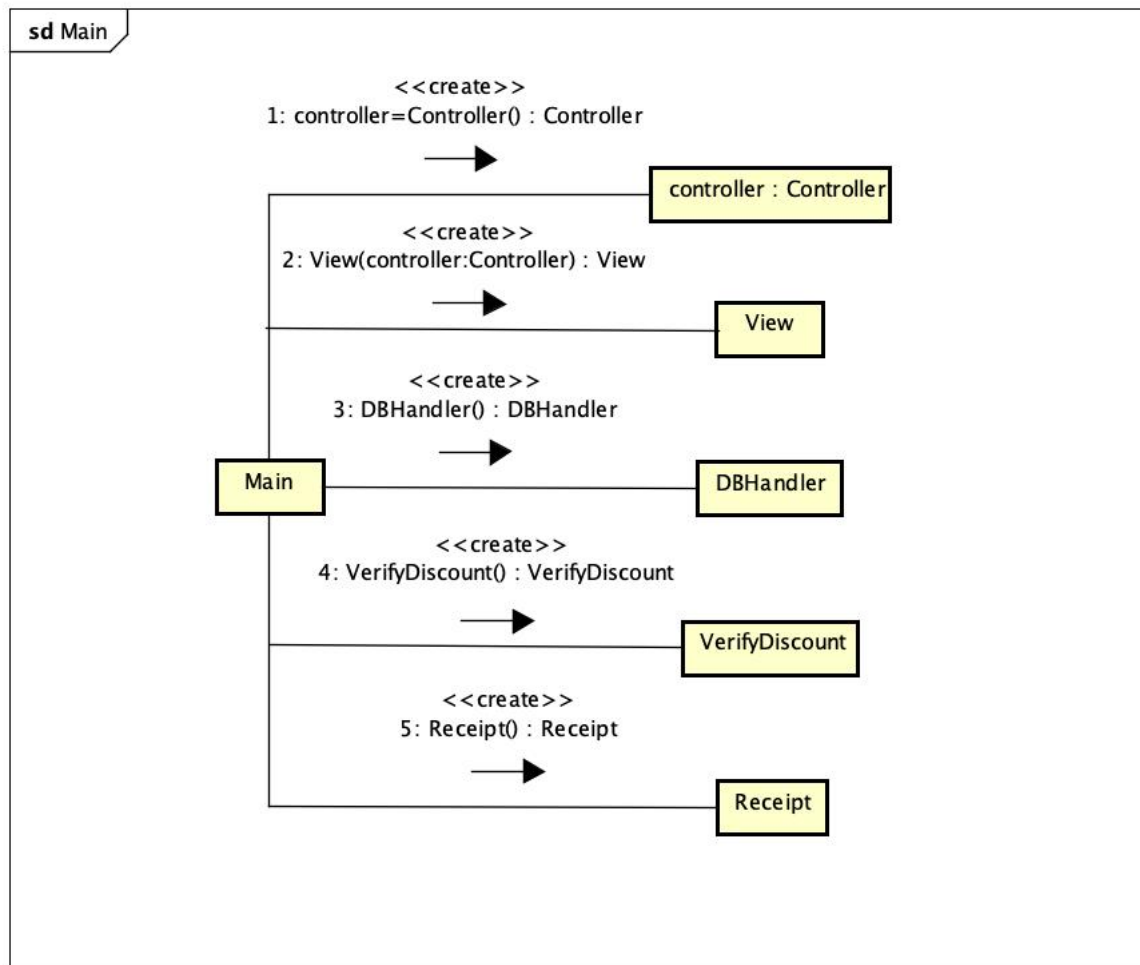
### 3 Resultat

Det slutliga program samt alla kommunikationsdiagram visas nedan. Paketen har placerats på ett sätt så att det blir lättast att se och förstå associationer och dependency. Startup paketet innehåller main och därför ska ha tillgång till de andra paketen som man kan se dras det mest dependency från startup. Däremot under vår komplettering har vi minskat antal associationer och dependency och har fått en lättare beskrivning av samarbete och koppling mellan olika klasser.



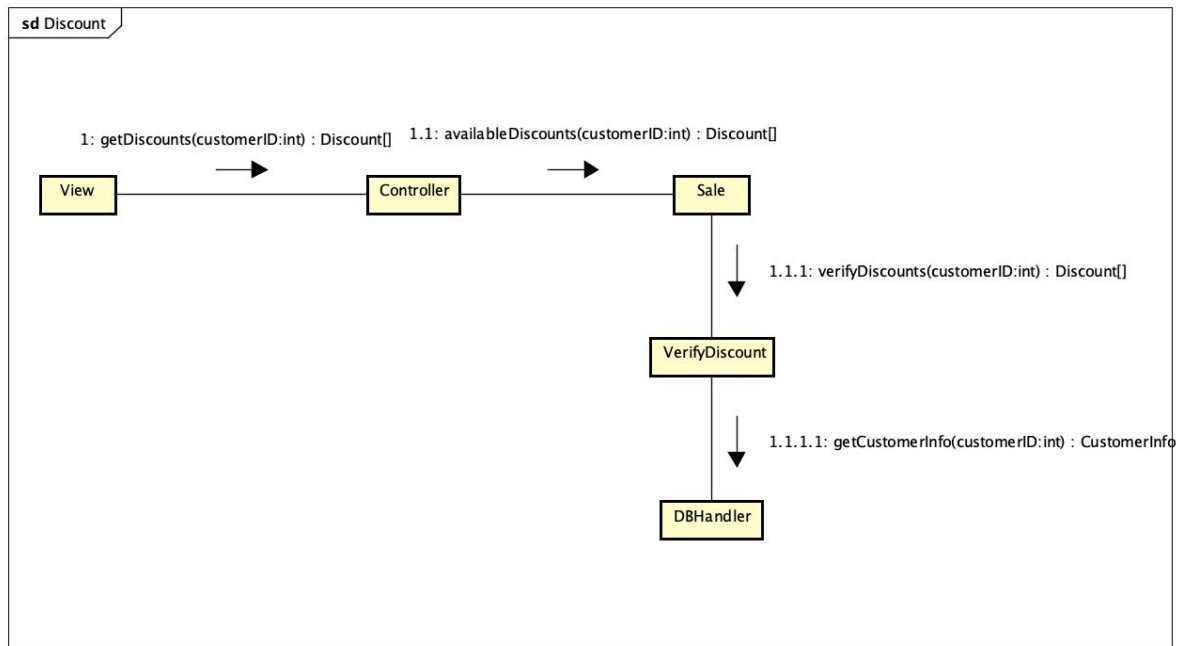
Figur 3.1 visar själva klassdiagrammet med alla paket.

Antal associationer i vår nya version av diagrammet har minskat så det finns bättre förståelse av hur allt är kopplade till varandra.

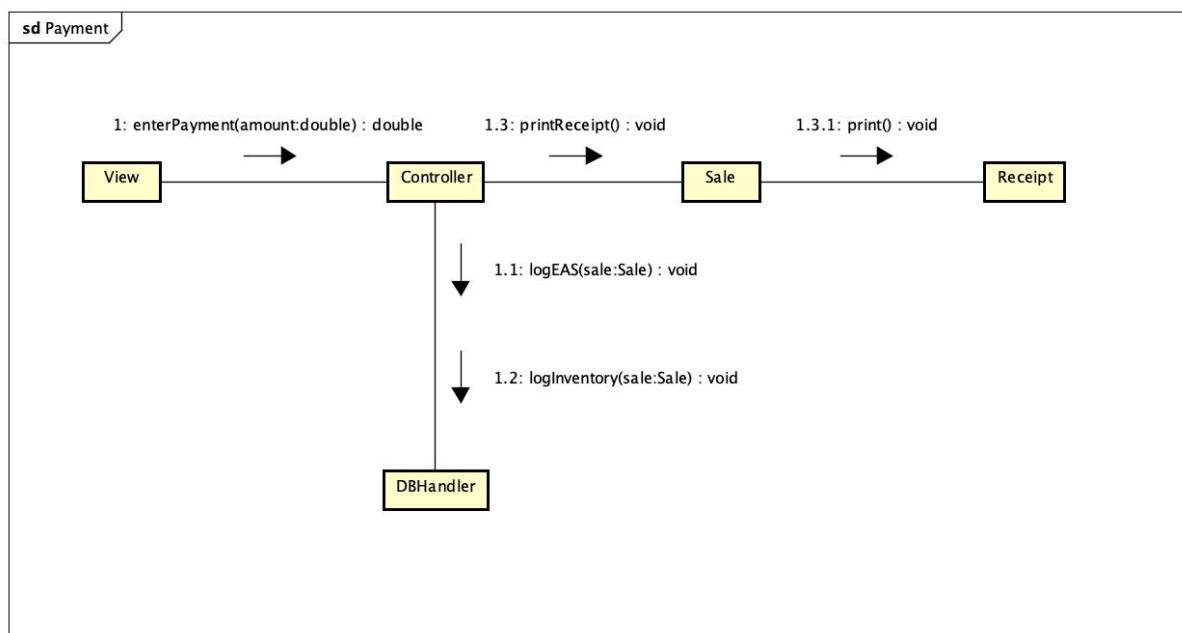


Figur 3.2 visar Main eller startup

Detta visar hur viktigt och centralt main är i ett program och har tillgång till andra klasser. Detta koppling krävs för att allt i systemet ska hängas ihop.

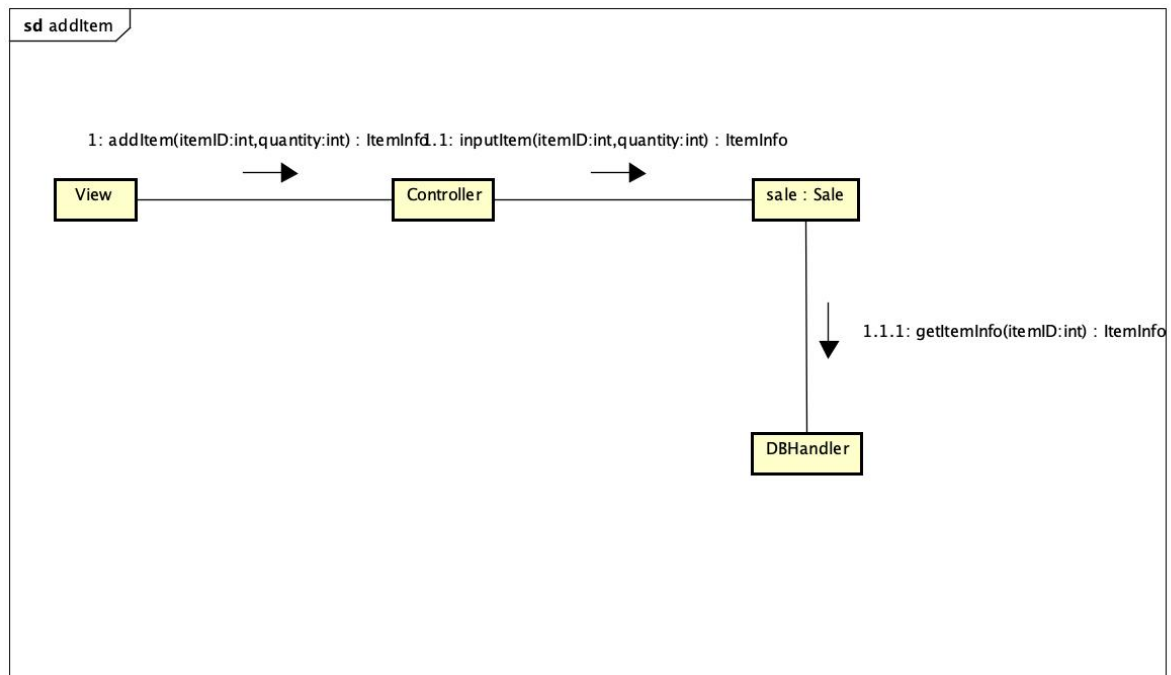


Figur 3.3 visar discount klassen



Figur 3.4 visar Payment klassen som tar hand om process av en betalning.

Betalningen har mycket med att skriva ut kvitto att göra. Detta ska bland annat controller och sale se till att händer och ett kvitto skrivs ut. Alla logning angående external system dvs accounting och inventory system ska vara under dbHandler.



Figur 3.5 visar process och klassen addItem som helt enkelt lägger till varor.

För att lägga till varor behövs det item id eller description och även antal. dbHandler innehåller info om vilka varor det existerar osv då det är inventory system som ligger där.



## 4 Diskussion

Jämfört med seminarium 1 känner jag att jag har förstått detta seminarium bättre och är mer nöjd med mina diagram. Allt verkar att vara tydligt och lätt att förstå. Däremot är jag lite osäker om mina klasser vilket gör att jag blir även osäker om high cohesion stämmer i mitt program. Till exempel under klass Sale, jag vet inte om Diff och logSale borde kanske vara under en annan klass eller om det stämmer som det är. Men jag kunde inte komma på någon annan klass som de skulle passa i och därför lämnade det så. Jag känner mig mer säker om low coupling då jag tycker att dependency i programmet är så få som möjligt.

Jag känner mig även lite osäker angående anropningar och metoder då jag inte är helt inne i kodningstanke än men jag känner att det jag har gjort känns rimligt och jag kunde inte göra något förbättring.

Vi hade mest problem med parametrar och numrering på vår föra version av seminarium 2. Detta kunde ha varit på grund av vår brist i förståelse av programmering och hur saker och ting är kopplat till varandra, vad skickar signal till vem osv. Denna förståelse har utvecklats nu och vi tycker att vi har undviktt alla misstag med numrering och parametrar som vi har haft. Föra gången hade vi även problem med att förstå hur controller samarbetar med andra och detta förstod vi nu med hjälp av andra elever.