

Seminarie 1

Object-Oriented Design, IV1350

Fredrik Pettersson fpe4@kth.se

2020-03-24

Innehåll

1. Introduktion

2. Metod

2.1 Task 1

2.2 Task 2

3. Resultat

3.1 Task 1

3.2 Task 2

4. Diskussion

4.1 Task 1

4.2 Task 2

1. Introduktion

Denna rapport beskriver och analyserar flödet vid kassadisen på en snabbköpsaffär. Det finns både ett grundflöde men även ett alternativt flöde som måste tas i åtanke. Jag har fått feedback och diskuterat lösningar på distans med Oscar Eklund och Viktor Jäger.

2. Metod

2.1 Metod Task 1

Det första jag gjorde när jag skapade domän modellen var att skriva upp alla tänkbara klasser. Jag hade hela tiden i bakhuvudet att det bästa är att brainstorma fram så många klasser som det går. Det är bättre att komma på för många och sedan vara tvungen att radera dem i ett senare tillfälle än att inte skriva upp dem alls då detta kan leda till att viktiga klassen missas. Först gick jag igenom punkt för punkt i uppgiften och skrev upp alla substantiv. Därefter gick jag efter kategori listan som finns i boken på sidan 26, då gick jag igenom uppgiften igen och skrev upp fler tänkbara klass kandidater.

När alla tänkbara klasser hade tagits fram så tog jag fram de attribut som klasserna har, dvs egenskaper hos klassen. I detta steg så raderades ganska många klasser då dessa inte var nödvändiga.

Därefter så tog jag fram associationer, dessa är till för att förtydliga och förklara flödet i vår domän modell. Detta förklarar hur klasserna är förknippade med varandra.

Det viktigaste i processen att ta fram domän modellen var att den ska spegla verkligheten samt att den ska vara enkel att förstå.

2.1 Metod Task 2

Precis som vid arbetet med domän modellen så började jag med att ta fram de klasser som finns i kravspecifikationen. Det vi är intresserade av här är in och utdata till systemet så med det i åtanke tar vi fram de klasser som finns.

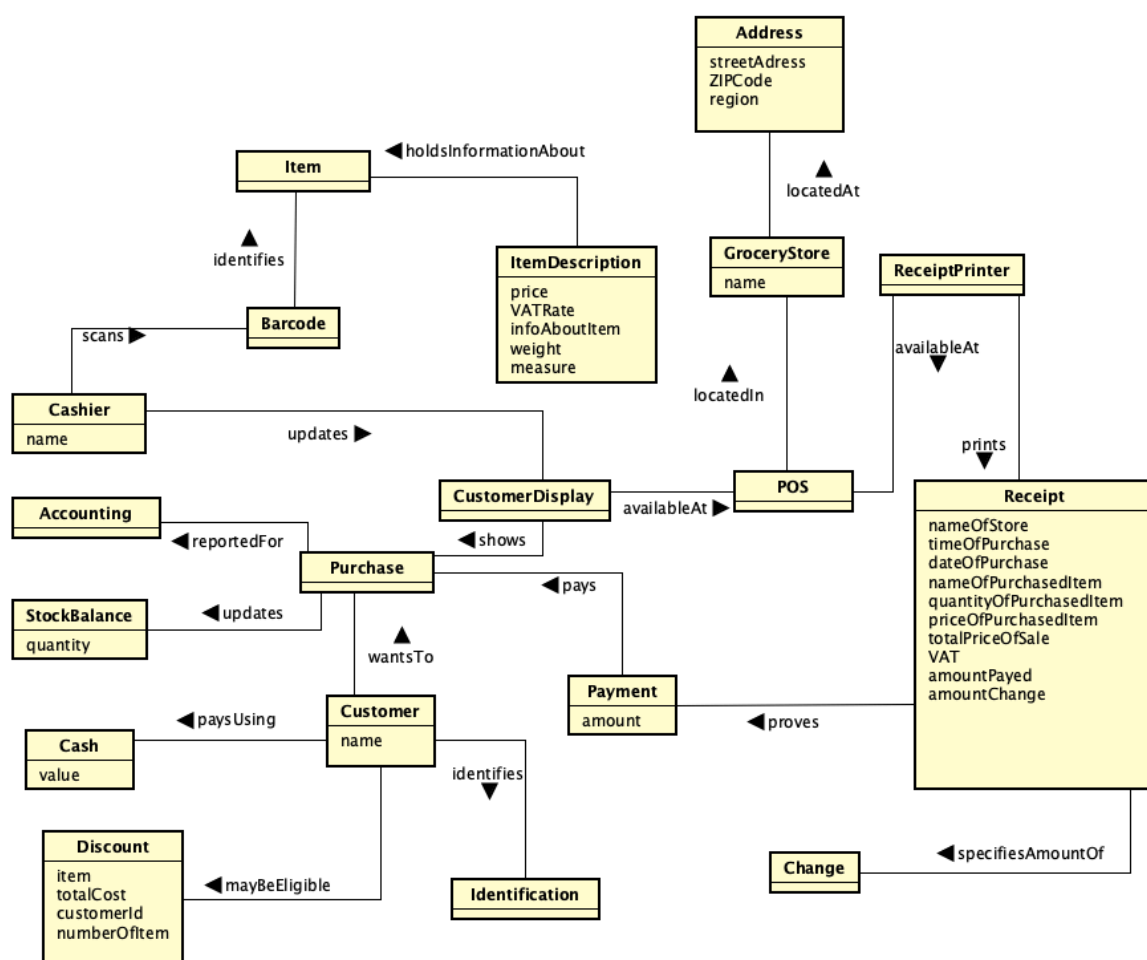
Därefter gick jag igenom punkt för punkt och analyserade om de gör något med systemet. Om så är fallet så ritades en pil samt att namn och eventuella argument samt returvärde skapades.

Det som jag gjorde allra sist var att skapa de loopar och if-satser som måste finnas med för att system sekvensdiagrammet ska vara rätt.

3. Resultat

3.1 Resultat Task 1

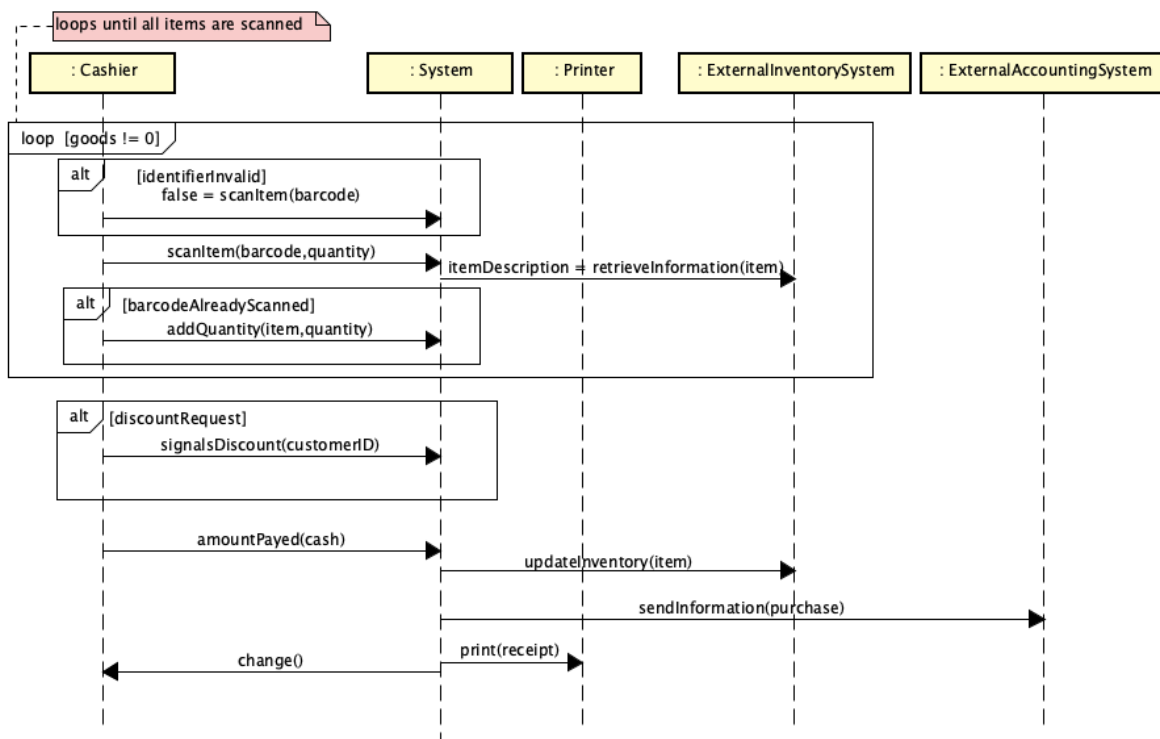
Den designade domän modellen (figur 3.1) skapades för att enkelt spegla verkligheten samt att på ett tydligt sätt visa det flöde som beskrevs i kravspecifikationen. De mest essentiella delarna av modellen ligger så centralt det går och sedan är den designat utåt. Klasser, attribut samt associationer är skapade efter de namnkonventioner som finns i språket Java. Både grundflödet samt alternativt flöde är medräknat när domän modellen skapades.



Figur 3.1 domän modell som visar flödet i en snabbköpsaffär.

3.2 Resultat Task 2

System sekvensdiagrammet (figur 3.2) visar hur externa aktörer samspelar med programmet. Dels aktörens (i vårt fall Cashier) anrop till systemet, men även vad systemet sedan anropar vidare till externa system (i vårt fall Printer, Externa inventory System och External Accounting System). Operationernas namn är skapade på ett sådant sätt att man enkelt kan förstå deras syfte. Den första loopen som finns med är en while loop som körs tills alla varor har lagts till. Första if-satsen körs om streckkoden är ogiltig, då returneras "false". If-sats nummer två körs om streckkoden redan har registrerats i ett tidigare fall, så adderas kvantiteten av det föremålet. Den tredje och sista if-satsen körs om kunden hävdar att denne är berättigad till rabatt.



Figur 3.2 system sekvensdiagram som visar interaktionen mellan aktören och systemet.

4. Diskussion

4.1 Diskussion Task 1

Jag tycker att min domän modell speglar verkligheten. Med det menar jag att den inte är programmatisk utan att den är uppbyggd på ett sådant sätt så att en person som inte kan programmera enkelt ska förstå denna. Genom de val av klasser, attribut samt associationer samt på det sätt jag har namngivit dem så bör min modell vara enkel att förstå.

Domän modellen har ingen klass som är "spindeln i nätet" och med det menas att den inte innehåller någon klass som har överdrivet många associationer. En modell som är uppbyggd på det sättet utgår överdrivet mycket från en klass och skulle vara av större värde om associationerna fördelades ut bättre. Om en domän modell innehåller för många associationer kan den även blir svår att förstå då den gärna blir väldigt rörig.

Ett vanligt problem som kan uppstå när man gör en domän modell är att göra den för naiv. Med det menas att den speglar verkligheten men ger ingen mer information än den som står i kravspecifikationen eller innehåller saker som är för uppenbara. Första utkastet av min domän modell blev alldeles för naiv och jag hade med flera associationer som var för uppenbara för att behöva vara med. Ett exempel på detta är kund → går fram till → kassan. Detta är underförstått och behöver inte vara med. Men på det sätt som domän modellen är uppbyggd nu så innehåller den inte några associationer som är onödiga utan de som är med fyller en funktion.

Utöver detta så har de namnkonventioner som gäller följts för de klasser, attribut samt associationer som är med i domän modellen.

4.2 Diskussion Task 2

System sekvensdiagrammet visar samspelet mellan användare och systemet, användaren skickar data till systemet och får antingen tillbaka något eller så skickas data vidare till externa system. Det som är viktigt vid designen av detta diagram är att få till alla operationer rätt samt att om det ska returneras något värde så ska detta vara med. Även gäller att de loopar och if-satser som måste vara med för att den ska stämma ska vara rätt inkluderade. De externa system som finns med är "External Inventory System" som lagrar information angående de föremål som säljs. Det systemet måste även uppdateras vid färdigt köp för att rätt lagersaldo ska hållas. "External Accounting System" är det bokföringssystem som måste anropas efter ett köp har gjorts för att logga köpet med information till redovisningen. Utöver detta är "Printer" med som skriver ut kvittot som bevisar köpet.