

Serviceprogramma	
Titel: UML – Sequentiediagram	
Studentversie	
Afdeling	ICT
Opleiding	Applicatieontwikkelaar
Crebonummer	95311
Niveau + BOL/BBL	BOL
Startdatum in curriculum	September 2010
Leerjaar	2
Periode	5
Week	4
Niveau serviceprogramma	Verkenning

ALGEMENE INFORMATIE

Soort serviceprogramma	Vaktechnische competentie	
Competenties	H, I, J, K, M, N, O, R	
Werkprocessen	1.1, 1.2	
Tijd	125 minuten	
Veronderstelde voorkennis	Geen	
Opbouw serviceprogramma	25 minuten	Opstart (welkom, onderwerp bekend maken, doelen doornemen, VOORDOEN)
	75 minuten	Oefenen, Oefenen, Oefenen (bijsturen waar nodig, rondlopen)
	25 minuten	Afsluiting (evalueren doelen, aantekeningen laten maken en in portfoliomap laten stoppen)
Didactische werkvormen	Instructie Oefening – UML-activiteitendiagram maken Presenteren	
Rol trainer	<ul style="list-style-type: none">• Voordoen (expert)• Vragensteller• Feedbackgever	
Doelen	Aan het einde van het serviceprogramma beheerst / weet de student het volgende: VAARDIGHEIDSDOELEN, de student KAN: 1. Een Sequentiediagram ontwerpen van een aangeleverde context. KENNISDOELEN, de student WEET:	

	<ol style="list-style-type: none"> 1. De methode om tot een Sequentiediagram te komen 2. De betekenis van de verschillende symbolen die gebruikt worden in Sequentiediagrammen. 3. De manier waarop deze symbolen gebruikt worden om een Sequentiediagram logisch te ontwerpen, zodat dit de basis voor het functioneel ontwerp is.
Benodigde ruimte	Ruimte waar met laptops gewerkt kan worden
Leermiddelen en materialen	<ul style="list-style-type: none"> • Laptop • Visio
Bijlagen	bijlagen <ol style="list-style-type: none"> 1. Persoonlijke leerdoelen 2. Serviceprogramma
Literatuur en bronnen	<ul style="list-style-type: none"> - Informatieanalyse – Gerlof Donga en Bert Pinkster - Praktisch UML – Jos Warmer en Anneke Kleppe - Handleiding Visio
Opmerkingen	

Bijlage 2

Serviceprogramma UML_ Sequentiediagram

Dit is het vierde Serviceprogramma in de serie Object Oriented Design en UML. De totale Serviceprogramma serie bestaat uit de volgende onderdelen:

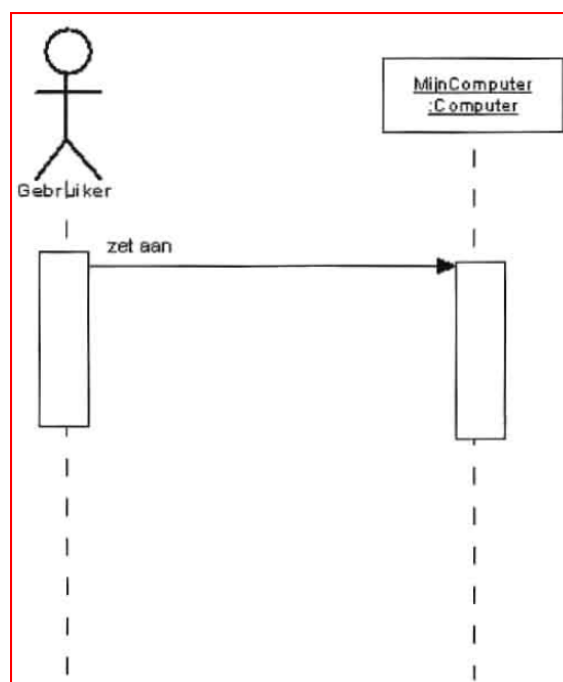
1. Object Oriented Design en UML, Use-cases en Use-case template
2. Object Oriented Design en UML, Activiteitendiagrammen
3. Object Oriented Design en UML, Klassendiagrammen en Specificaties voor een database
4. Object Oriented Design en UML, Sequentiediagrammen

Sequentiediagrammen

In de UML-modellen gaat het steeds over gedrag. Nadat we in vorige Serviceprogramma's bepaald hebben welke objecten met elkaar communiceren, geven we in dit Serviceprogramma aan wanneer en hoe ze dat doen. We geven de volgorde van de berichten tussen klassen aan en ook of er antwoord verwacht wordt. Het hulpmiddel dat UML hiervoor aanbiedt is het *sequentie diagram*. Sequentiediagrammen behoren samen met *collaboratiediagrammen* (die we niet behandelen) tot de *interactiediagrammen* van UML.

Het doel van een sequentiediagram

In sequentiediagrammen geven we aan *hoe een proces verloopt*. Elke handeling wordt ingetekend; elke keer dat een object aangesproken wordt, nemen we dat op in het sequentiediagram. Daarmee lijkt het veel op het activiteitendiagram. Het verschil is dat we ons bij een activiteitendiagram meer richten op het menselijk handelen, terwijl we nu over **programmaonderdelen** praten. We zijn nu dus meer met het programmaontwerp bezig. We hebben al klassen gevormd, we weten op welke wijze de klassen contact met elkaar hebben. Nu wordt het tijd om vast te leggen hoe de berichtenstroom tussen de klassen verloopt. Dat lijkt heel simpel, maar we moeten ook rekening houden met alle mogelijke uitzonderingssituaties.



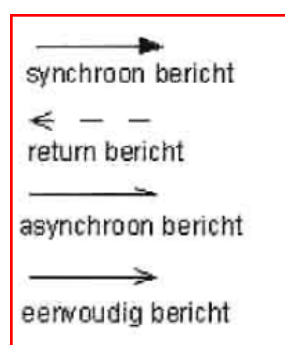
Aan de bovenzijde van de figuur treffen we een paar bekende symbolen. Het draadpoppetje aan de linkerkant is een actor. Deze actor, met de naam Gebruiker, doet hetzelfde als in een use-case diagram. Hij neemt het initiatief, hij zet iets in gang. Daarnaast staat het rechthoekje dat we kennen als klasse of object in een klassendiagram. In dit geval gaat het om het object MijnComputer van de klasse Computer. De schrijfwijze van object en klasse zijn hetzelfde als in het objectdiagram. Onder zowel actor als object bevinden zich stippellijnen. Deze worden **levenslijnen** (lifelines) genoemd. Zolang de actor of het object beschikbaar is, loopt de stippellijn door. Het rechthoekige blokje dat op de levenslijn getekend staat, geeft weer dat de actor of het object het proces bestuurt. De pijl vertelt wat er gedaan wordt. In de figuur zet de actor Gebruiker dus MijnComputer aan.

De berichten

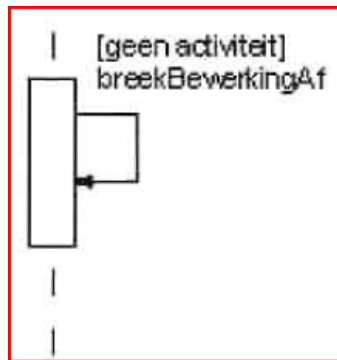
De pijl in het sequentiediagram vertelt ons niet alleen wat er gedaan wordt. Een pijl met een gesloten driehoek als pijlpunt vertelt dat het om een *synchroon bericht* (synchronous message) gaat. Bij een synchroon bericht neemt het bericht de sturing over tot het bericht voltooid is. Daarna volgt er een terugmelding dat het resultaat bereikt is. In dit geval zet de gebruiker MijnComputer aan. Het bericht zorgt ervoor dat alle starthandelingen van de computer uitgevoerd worden en na afloop daarvan, dus bijvoorbeeld na het Windows-melodietje, krijgt gebruiker de besturing weer terug.

Soms is het zinnig om aan te geven dat de besturing vrijgegeven wordt. Meestal is dat vanzelfsprekend. Als we de reactie willen laten zien, kunnen we gebruikmaken van de Return-pijl. Dit is een gestippelde pijl met een normale pijlpunt die terugverwijst naar het object waar het bericht vandaan kwam.

Naast synchrone berichten bestaan er ook *asynchrone berichten*. Hier zendt het actieve object een bericht, maar wacht niet op een reactie. Er kan direct een volgend bericht gestuurd worden. Asynchrone berichten worden aangegeven met een halve pijlpunt. Als we echt niet weten of een bericht synchroon of asynchroon is, kunnen we een gewone open pijlpunt gebruiken. Deze betekent dat niet de ontwerper (wij dus), maar de uitvoerder, de programmeur bepaalt hoe het bericht uitgevoerd zal worden.



In een aantal gevallen kan een bericht terugverwijzen naar de eigen activering. Dat geven we aan met een gebogen pijl.

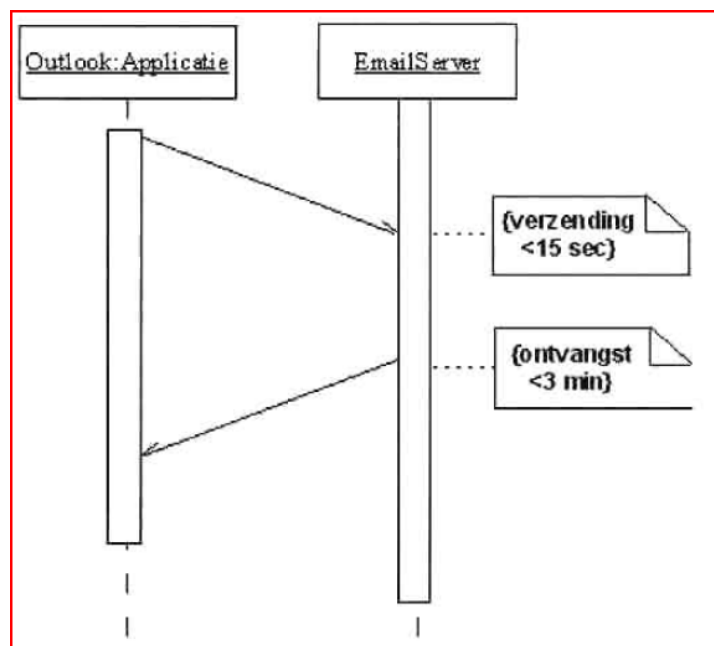


Tijdgebonden overgangen

Normaal gesproken worden alle berichten als horizontale lijnen aangegeven. Eigenlijk is dit vreemd als we bedenken dat we tijd als verticale as in het activiteitendiagram aangeven. We tekenen dus dat de activering door een actor of object tijd kost, maar het sturen van berichten niet. Vaak is het ook zo dat de bewerking door een object veel meer tijd kost dan het zenden of ontvangen van een bericht. Wat we tekenen is dus niet helemaal juist, maar de afwijking van de werkelijkheid is slechts gering.

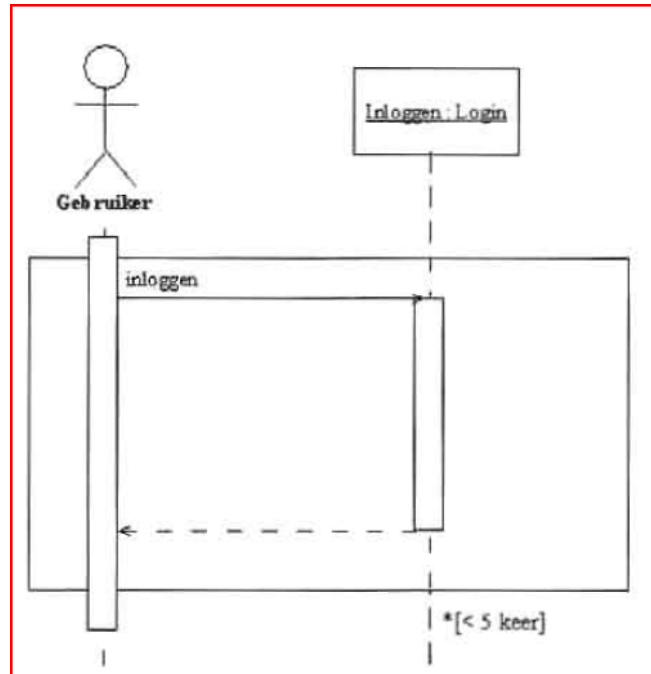
Als we willen aangeven dat het zenden van een bericht wel tijd kost, dan kunnen we dit doen door de pijl niet horizontaal, maar schuin naar beneden gericht te tekenen. Het is nu duidelijk dat dit bericht echt tijd kost. We doen dit vooral in applicaties die echt tijdkritisch zijn. We geven dan niet alleen aan dat het zenden en ontvangen van een bericht tijd kost, maar ook hoeveel tijd het kost.

Als we kunnen benoemen hoe lang de toegestane tijd is, kunnen we dit toevoegen aan ons sequentiediagram. Een dergelijke voorwaarde noemen we een constraint. Deze plaatsen we tussen accolades



Iteraties

Iteraties zijn herhalingen: we doen iets vaker dan één keer. Soms staat van tevoren vast hoe vaak we iets doen, soms gaan we door met herhalen tot een bepaalde toestand bereikt is. Als we herhaling in een sequentiediagram willen aangeven, tekenen we een rechthoek om de te herhalen routine en geven rechts onder de rechthoek de voorwaarde (de guard-conditie).



De volgorde van objecten in een sequentiediagram

In een wat uitgebreider sequentiediagram zien we van links naar rechts altijd dezelfde volgorde:

1. De actor
2. De Graphical User Interface (GUI)
3. De klassen die gebruikt worden in dit proces
4. De database

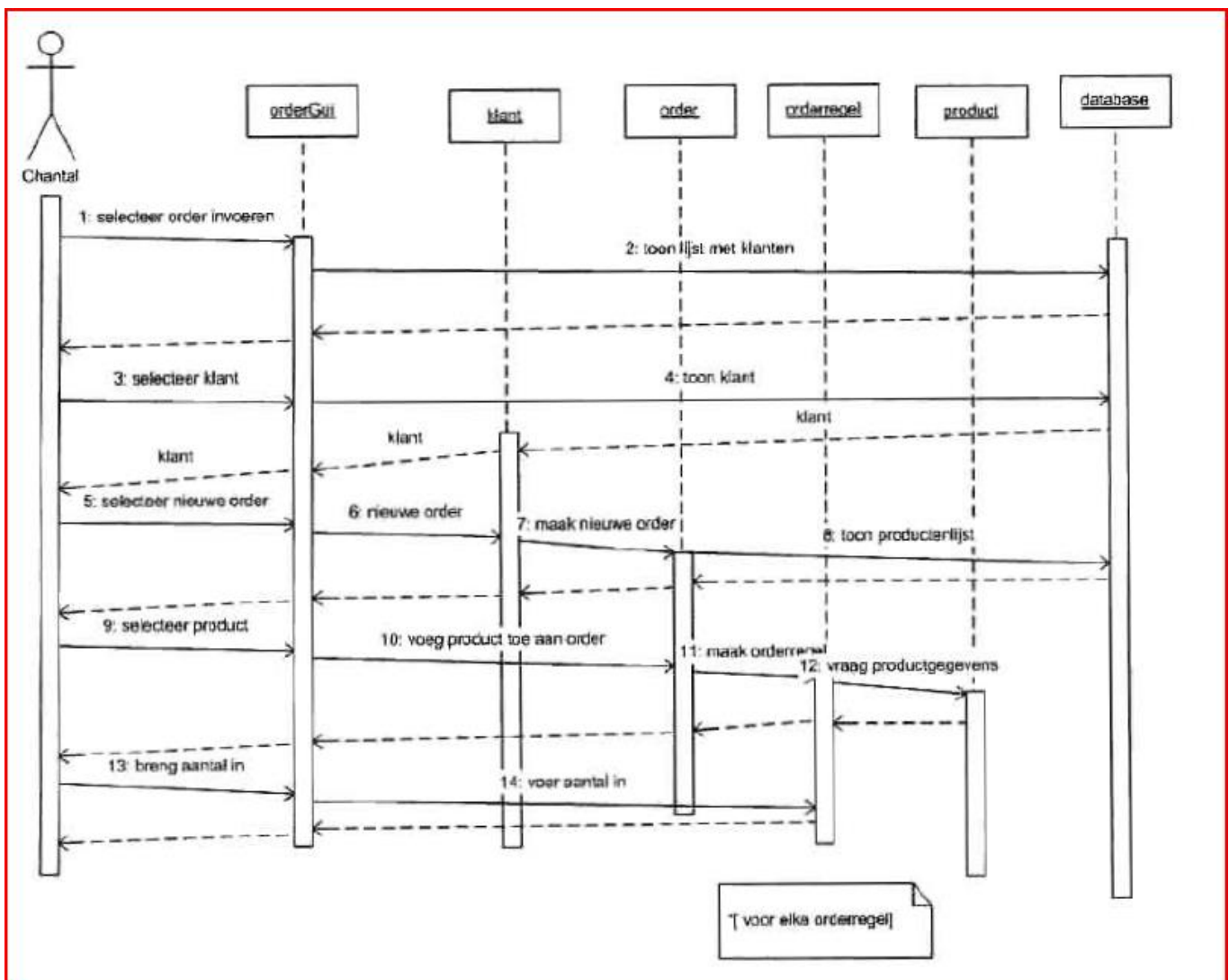
Voorbeeld

Bij het opzetten van een sequentiediagram maken we gebruik van de use-case beschrijving of van enkele door de klant aangegeven scenario's. We willen immers de communicatie met het te ontwerpen systeem eenduidig vastleggen. Het kan gebeuren dat we tijdens het opstellen van een sequentiediagram ontdekken dat de klant onvolledige of soms zelfs onjuiste, tegenstrijdige informatie heeft verstrekt. Dit heeft niets te maken met onwil van de klant, maar vooral met de moeite om precies te formuleren wat er werkelijk gebeurt. Een beschrijving die voor mensen hanteerbaar is, bijvoorbeeld een procedurebeschrijving, zit vaak nog vol met hiaten voor een te ontwerpen systeem.

Bij het maken van de sequentiediagrammen streven we naar een zo goed mogelijke weergave van de werkelijkheid. Elk sequentiediagram moet in ieder geval technisch correct zijn. Ook is het sequentiediagram bedoeld om de communicatie met de klant te ondersteunen. Het vaststellen dat de verstrekte informatie onjuist of onvolledig is, levert voldoende materiaal op

om nieuwe vragen aan de gebruiker te stellen. Ook het maken van sequentiediagrammen gebeurt dus iteratief.

Chantal: “Bij een klantgesprek moet ik de mogelijkheid hebben om een bestelling in te voeren. Ik selecteer de klant en vervolgens de gewenste producten. Bij elk geselecteerd product voer ik het gewenste aantal in. Nadat alle producten zijn ingevoerd, kan ik de order bevestigend afsluiten. Het systeem toont een bevestiging van de opslag van de zojuist geplaatste order.”



Oefening 1

Gegeven is de volgende context:

- a. Maak sequentiediagrammen van de hoofdfuncties in Visio.
- b. Bij de afsluiting zal je gevraagd worden om je Sequentiediagrammen te presenteren, waarbij je duidelijk moet aangeven, waarom je het op jouw manier hebt getekend.

De Fietsfabriek

Een voormalige fietsenmaker is een fietsenfabriek “De Fietsfabriek” begonnen. De fabriek koopt de onderdelen voor de fietsen in en assembleert zelf de fietsen. De ondernemer vraagt om de automatisering van de nieuwe organisatie op te zetten: Hij wil een applicatie om de inkoop, voorraad en verkoop bij te houden. Ook moeten klanten via Internet producten (fietsen en accessoires) kunnen bestellen en betalen. Klanten kunnen zowel privé-personen zijn, als fietswinkels.

Om kort te gaan wil “De Fietsfabriek” de volgende functies:

- Inkopen van onderdelen bij leveranciers: De Fietsfabriek heeft verschillende leveranciers waarbij ze producten (fietsonderdelen) inkoopt. Hierbij gaat het om nieuwe voorraad, maar ook voor het aanvullen van bestaande voorraad. Als er besloten wordt om een nieuw product aan te schaffen door de afdeling Marketing, wordt er contact gelegd met de betreffende leverancier en een bestelling gedaan. De leverancier stuurt een bevestiging van de bestelling en de bestelling wordt verzonden samen met de factuur. Zodra de bestelling is aangekomen, wordt deze door de afdeling Voorraadbeheer opgeborgen en de gegevens van de bestelling worden geregistreerd in het systeem. De factuur gaat naar de afdeling Financiën, die deze factuur betaald. Op vrijdag draait het systeem een lijst uit van alle producten waarvan het aantal onder de vastgestelde limiet komt. Deze worden dan bijbesteld en verwerkt op dezelfde wijze als nieuwe producten.
- Aanmaken klantaccounts: Een klant kan via internet zijn eigen account aanmaken. Deze heeft hij nodig om fietsen te kunnen kopen. Hierbij worden NAW- en rekeninggegevens geregistreerd. Elke klant heeft een unieke combinatie van inlognaam en wachtwoord.
- Verkopen van producten aan klanten: Een klant kan via internet producten (fietsen en accessoires) kopen. Nadat de klant heeft ingelogd, kan hij producten selecteren om in zijn winkelwagentje te doen. Zodra hij klaar is met selecteren, kan hij de bestelling afronden. Hierbij kan hij eventueel het afleveradres veranderen. Betalingen kunnen gedaan worden via creditcard, paypal of ideal. Zodra de betaling binnen is, geeft de afdeling Financiën een bericht door aan de afdeling Voorraadbeheer. Deze verzamelt de bestelde producten, verpakt ze samen met een verpakkingsbon en laat ze ophalen door het vervoersbedrijf. De voorraden worden direct bijgewerkt in het systeem.
- Marketing: Deze afdeling houdt zich bezig met het in en uit de handel nemen van producten. Als het om een nieuw product gaat, registreert deze afdeling de basisgegevens (naam, kleur, maat, minimumvoorraad, etcetera) in het systeem. Wekelijks wordt er een verkooplijst uitgedraaid. Aan de hand van deze lijst, beslist de afdeling Marketing of een product behouden of afgestoten wordt. Dit is afhankelijk van de verhouding verkoopvoorraad en de snelheid waarmee de voorraad verkocht wordt. Wordt een product uit de handel genomen, dan wordt de afdeling Voorraadbeheer hiervan op de hoogte gesteld en wordt het product van de website afgehaald.

Oefening 2

Gegeven is de volgende context:

- a. Maak sequentiediagrammen van de hoofdfuncties in Visio.
- b. Bij de afsluiting zal je gevraagd worden om je Sequentiediagrammen te presenteren, waarbij je duidelijk moet aangeven, waarom je het op jouw manier hebt getekend.

Internetbedrijf SMALL

SMALL is een bedrijf wat kleding wil gaan verkopen voor baby's en peuters (leeftijd 0 tot en met 4) via internet. Hiervoor moet een systeem ontwikkeld worden. Het systeem van SMALL kent de volgende functionaliteit:

- Inkopen van kleding bij leveranciers: SMALL heeft verschillende leveranciers waarbij ze kleding inkopen. Hierbij gaat het om nieuwe voorraad, maar ook voor het aanvullen van bestaande voorraad. Als er besloten wordt om een nieuw product te gaan verkopen door de marketing afdeling, wordt er contact gelegd met de betreffende leverancier en een bestelling gedaan. De leverancier stuurt een bevestiging van de bestelling en de bestelling wordt verzonden samen met de factuur. Zodra de bestelling is aangekomen, wordt deze door voorraadbeheer opgeborgen en de gegevens van de bestelling worden geregistreerd in het systeem. De factuur gaat naar de afdeling financiën, die deze factuur betaald. Op vrijdag draait het systeem een lijst uit van alle producten waarbij het aantal onder de vastgestelde limiet komt. Deze worden dan bijbesteld en verwerkt op dezelfde wijze als nieuwe producten.
- Aanmaken klantenaccounts: Een klant kan via internet zijn eigen account aanmaken. Deze heeft hij nodig om kleding te kunnen kopen. Hierbij worden NAW-gegevens en rekeninggegevens geregistreerd. Elke klant heeft een unieke combinatie van inlognaam en wachtwoord.
- Verkopen van kleding aan klanten: Een klant kan via internet kleding kopen. Nadat de klant heeft ingelogd, kan hij producten selecteren om in zijn winkelwagentje te doen. Zodra hij klaar is met selecteren, kan hij de bestelling afronden. Hierbij kan hij eventueel het afleveradres veranderen. Betalingen kunnen gedaan worden via creditcard, paypal of ideal. Zodra de betaling binnen is, geeft de afdeling Financiën een bericht door aan de afdeling Voorraadbeheer. Deze verzamelt de bestelde producten, verpakt ze samen met een verpakkingsbon en laat ze ophalen door het vervoersbedrijf. De voorraden worden tevens bijgewerkt in het systeem.
- Marketing: Deze afdeling houdt zich bezig met het in en uit de handel nemen van producten. Als het om een nieuw product gaat, registreert deze afdeling de basisgegevens (naam, kleur, maat, minimumvoorraad, etcetera) in het systeem. Dagelijks wordt er een verkooplijst uitgedraaid. Aan de hand van deze lijst, beslist de afdeling marketing of een product behouden of afgestoten wordt. Dit is afhankelijk van de verhouding verkoop-voorraad en de snelheid waarmee de voorraad verkocht wordt. Wordt een product uit de handel genomen, dan wordt de afdeling voorraadbeheer hiervan op de hoogte gesteld en wordt het product van de website afgehaald.