

Serviceprogramma	
Titel: UML – Klassendiagram	
Studentversie	
Afdeling	ICT
Opleiding	Applicatieontwikkelaar
Crebonummer	95311
Niveau + BOL/BBL	4 BOL
Startdatum in curriculum	
Leerjaar	2
Periode	5
Week	3
Niveau serviceprogramma	Verkenning

ALGEMENE INFORMATIE

Soort serviceprogramma	Vaktechnische competentie	
Competenties	I (Presenteren), J (Formuleren en rapporteren) en M (Analyseren)	
Werkprocessen	1.4 (Maakt een technisch ontwerp)	
Tijd	160 minuten	
Veronderstelde voorkennis	Geen	
Opbouw serviceprogramma	20 minuten	Opstart (welkom, onderwerp bekend maken, doelen doornemen, VOORDOEN)
	120 minuten	Oefenen, Oefenen, Oefenen (bijsturen waar nodig, rondlopen)
	20 minuten	Afsluiting (evalueren doelen, aantekeningen laten maken en in portfoliomap laten stoppen)
Didactische werkvormen	Instructie Oefening – UML-klassendiagram 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: <ol style="list-style-type: none"> 1. Een klassendiagram ontwerpen van een aangeleverde context. 2. De attributen en operaties toevoegen aan de klassen. 3. De relaties en multipliciteiten tussen de klassen vaststellen. 	

	KENNISDOELEN, de student WEET: <ol style="list-style-type: none"> 1. De methode om tot een klassendiagram te komen. 2. De betekenis van de verschillende symbolen die gebruikt worden in klassendiagrammen. 3. De manier waarop deze symbolen gebruikt worden om een klassendiagram logisch te ontwerpen, zodat dit de basis voor het technisch 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	

UITVOERING DOOR DOCENT

Opstart (20 minuten)	<ul style="list-style-type: none"> - Studenten laten plaatsnemen aan U-opstelling - Welkom - Onderwerp bekend maken : waarom dit nodig voor je beroep: Dit onderwerp heeft te maken met Werkproces 1.4 “Maakt een technisch ontwerp”. - Doelen laten voorlezen door studenten / doelen op bord noteren : DIT GAAN WE VANDAAG TRAINEN - VOORDOEN
Oefenen (120 minuten)	<p>OPSTELLING GROEPJES / SUBGROEPJES:</p> <ol style="list-style-type: none"> 1. Studenten werken in groepjes van 2 aan de oefeningen <p>SPECIFIEKE OPDRACHT / THEMA / OEFENING:</p> <ol style="list-style-type: none"> 1. Zie bijlage 2
Afsluiting (20 min)	<ul style="list-style-type: none"> - Studenten laten plaatsnemen aan U-opstelling - Oefening 1 en 2 worden door een groepje gepresenteerd (eventueel wordt dit 2 maal gedaan, afhankelijk van de tijd). Niet iedere groep hoeft aan bod te komen. - Serviceprogramma evalueren - Studenten aantekeningen laten maken - Student aantekeningen en uitgewerkte sequentiediagrammen laten stoppen in portfoliomap

Bijlage 2

Serviceprogramma UML_ Klassendiagram

Dit is het derde 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

Klassendiagrammen

Klassendiagrammen vormen een onderdeel van het functioneel ontwerp. In het klassendiagram gaan we aangeven welke onderdelen wezenlijk zijn voor de applicatie en welke contacten er bestaan tussen de onderdelen. Hierbij moeten we de onderdelen benoemen met al hun kenmerken en handelingen. Om dubbel werk te voorkomen proberen we onderdelen te ontwerpen die op verschillende plaatsen bruikbaar zijn.

Het centrale onderdeel in ons ontwerp is de **klasse**. Dit is de basis van het gehele objectgeoriënteerde ontwerp. Een klasse (Eng. class) is een verzameling van objecten met overeenkomstige eigenschappen. De klassebeschrijving geeft de naam van de klasse en de beschrijving van de eigenschappen van de instanties (een instantie is een concreet voorkomen van de klasse. Bijvoorbeeld: Klasse = Leraar, Instantie = Dhr. Pietersen; klasse = les, Instantie = Aardrijkskunde). Deze eigenschappen worden verdeeld in attributen en operaties.

Klassendiagram: hierin projecteren we de klassen en hun relatie met andere klassen. Een goed klassendiagram kenmerkt zich door zoveel mogelijk onafhankelijke klassen en zo min mogelijke relaties met andere klassen.



Dit is een eenvoudig klassendiagram.

Een klasse heeft:

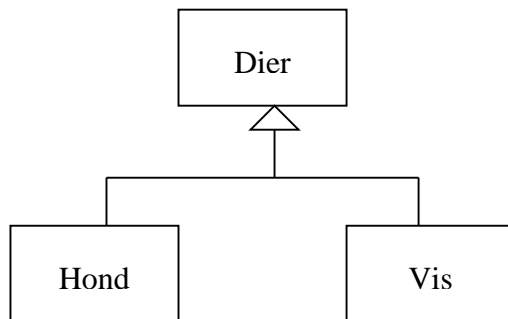
- attributen (kenmerken)
- operaties (handelingen).

Hond
<i>Attributen:</i> -Ogen -Poten
<i>Operaties:</i> +Eten() +Slapen()

Als we bijvoorbeeld de klasse Hond definiëren, kunnen we bepalen dat elke hond ogen en poten heeft en dat elke hond eet en slaapt. De ogen en poten zijn attributen, eten en slapen zijn handelingen.

Overerving

Op soortgelijke wijze kunnen we de klasse Vis gaan bekijken. Net als een hond, heeft de vis ogen, eet hij en slaapt hij. In plaats van poten heeft de vis vinnen. Daar is dus een verschil. Allebei (Hond en Vis) zijn dieren. Ze hebben een aantal gemeenschappelijke kenmerken en kunnen dus samen tot de klasse Dier gerekend worden. De afwijkende kenmerken (poten voor de hond en vinnen voor de vis) zijn dan geen kenmerken van de klasse Dier.



We kunnen de klasse Dier bouwen en daarin alle gemeenschappelijke kenmerken voor Hond en Vis plaatsen. Vervolgens bouwen we de klassen Hond en Vis en zetten daarin alleen de afwijkingen (dus poten voor de hond en vinnen voor de vis). We verbinden de klassen Hond en Vis met de klasse Dier door een lijn met aan de kant van de klasse Dier een driehoek. Dit betekent dat de klassen Hond en Vis alle kenmerken van de klasse Dier erven. We noemen dit **overerving** of in het Engels **inheritance**. De klasse waarvan geërfd wordt, noemen we **superklasse**, de ervende klassen **subklassen**.

Attributen en operaties

Zoals al eerder aangegeven, kunnen bij elke klasse attributen en operaties vermeld worden. Zo kan het verstandig zijn om bij de klasse Klant de volgende attributen te vermelden:

- Bedrijfsnaam
- Vestiging (filiaal)
- Contactpersoon
- Adres van het filiaal
- Datum van de laatste bestelling

Als we de attributen en operaties in een applicatie willen gebruiken, zullen we ze daar moeten declareren: voordat we ze gaan gebruiken, melden we aan het programma aan welk gegevenstype ze voldoen. Zo kunnen we registreren of het gaat om gehele getallen, getallen met een komma erin, tekst, logische waarden (juist/ onjuist), enzovoort. Elke programmeertaal heeft zijn eigen datatypes. In onze klassen vermelden we achter het attribuut of de operatie aan welk datatype de invoer, bewerking en uitvoer moet voldoen. Dat ziet er dan als volgt uit:

Bedrijf
Contactpersoon: String Laatste contact: date
Bestellen() Factureren()

Een operatie beschrijft een service die een object levert. De verzameling operaties bij een object representeert het gedrag van het object. Omdat alle instanties van een klasse dezelfde operaties hebben, worden de operaties beschreven bij de klasse.

Klasse en object

Een klasse is altijd een abstract gegeven. Er wordt bijvoorbeeld een hond gedefinieerd. Als ik de kenmerken van mijn hond wil aangeven, dan kan ik die in de definitie van hond wel kwijt. Op het moment dat ik de voor mij van belang zijnde gegevens ga invullen, spreken we niet meer van een klassendiagram, maar van een **objectdiagram**. Een object bevat dus de beschrijving van een specifieke set gegevens, passend binnen de definitie van de klasse.

Regiocollege:Klant
contactpersoon: string = R. Kinneging adres: string = Cypressehout 97 plaats: string = Zierikzee

Het object Regiocollege van de klasse Klant

Van kandidaat-klassen tot klassen

Hoe bepalen we uit een probleem welke klassen we moeten gaan gebruiken?:

1. Onderstreep de zelfstandige naamwoorden in een tekst (kandidaat-klassen). Neem het bijvoeglijk naamwoord mee als dit een verduidelijking van de kandidaat-klasse oplevert.
2. Kijk of alle zelfstandige naamwoorden bruikbaar zijn:
 - a. Sommige kandidaat-klassen zijn synoniemen. Deze worden dan **redundante** klassen genoemd. We kunnen deze klassen wegstrepen.
 - b. Sommige kandidaat-klassen zijn **vaag** en daardoor onbruikbaar. De klasse moet een duidelijke betekenis hebben.
 - c. Sommige zelfstandige naamwoorden geven eigenschappen weer van andere zelfstandige naamwoorden. Dit zijn dan de **attributen** (of soms zelfs de **operaties**) van een klasse.

Voorbeeld:

De keuze voor een computer hangt vooral af van het soort gebruik en de maximaal te betalen prijs. Een spelfanaat zal graag wat meer willen betalen voor een computer met uitgebreide multimediamogelijkheden terwijl een zakelijke computergebruiker vooral de stabiliteit van het systeem als belangrijkste aandachtspunt heeft. Toch kan men zich gemakkelijk vergissen. Tegenwoordig is een ingebouwde cd-brander zowel door de spelfanaat als de zakelijke gebruiker gewenst.

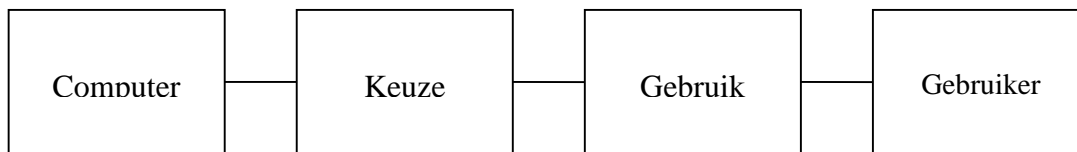
We kunnen deze zelfstandige naamwoorden in een tabel opnemen en daarnaast aangeven of we deze ook daadwerkelijk als klasse gaan opnemen of afkeuren met opgave van reden

Kandidaatklasse	Beslissing
Keuze	Klasse
Computer	Klasse
Gebruik	Klasse
Prijs	Attribuut van computer
Spelfanaat	Wordt klasse gebruiker
Multimediamogelijkheden	Attribuut van computer
Zakelijke computergebruiker	Wordt klasse gebruiker
Stabiliteit	Attribuut van computer
Systeem	Redundant met computer
Aandachtspunt	Te vaag
cd-brander	Attribuut van computer

Zo weten we het aantal klassen te beperken tot vier:

- Keuze
- Computer
- Gebruik
- Gebruiker

Als we ervan uitgaan, dat we nu het optimale resultaat bereikt hebben, is de volgende stap om te bepalen welke klassen relaties met elkaar hebben. De gebruiker maakt zijn keuze en bepaalt hoe hij zijn computer gaat gebruiken. Een klassendiagram kan er dan zo uitzien:



Multipliciteit

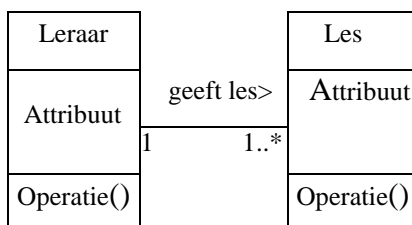
Bij de associatie (de relatie) tussen twee klassen kunnen we aangeven hoeveel objecten van de ene klasse geassocieerd zijn (een relatie hebben) met een object van een andere klasse.

Bijvoorbeeld: Een leraar zal één of meer lessen per week geven. Een les wordt altijd maar door één leraar gegeven

Het is nu eenvoudig om ook de aantallen die in de tekst vermeld zijn toe te voegen. We zetten naast de associatie getallen. Er zijn veel combinaties mogelijk:

- 1: het object aan deze kant komt 1 keer voor, dus niet minder (0) of meer dan 1.
- 1..8: het object aan deze kant komt in deze associatie 1 tot 8 keer voor.
- 2, 4, 6, 8: het object aan deze kant kan uitsluitend 2 of 4 of 6 of 8 keer voorkomen
- *: het object kan oneindig veel keren voorkomen.

Als we deze getallen naast de associatie zetten, spreken we van multipliciteiten.



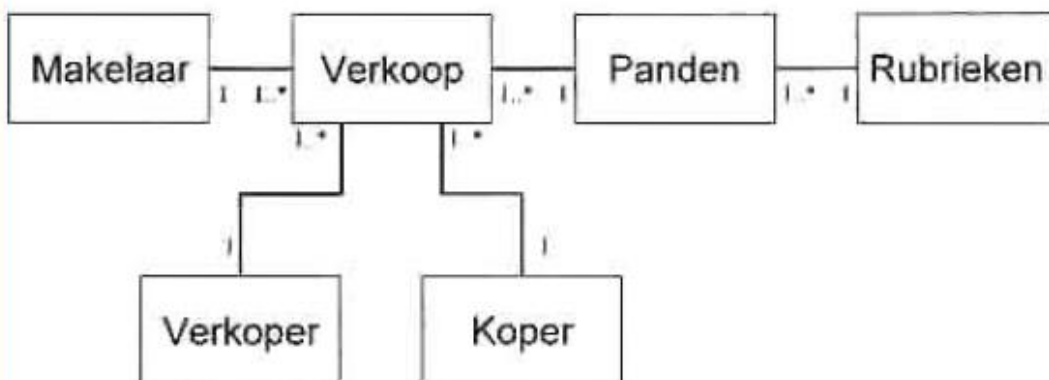
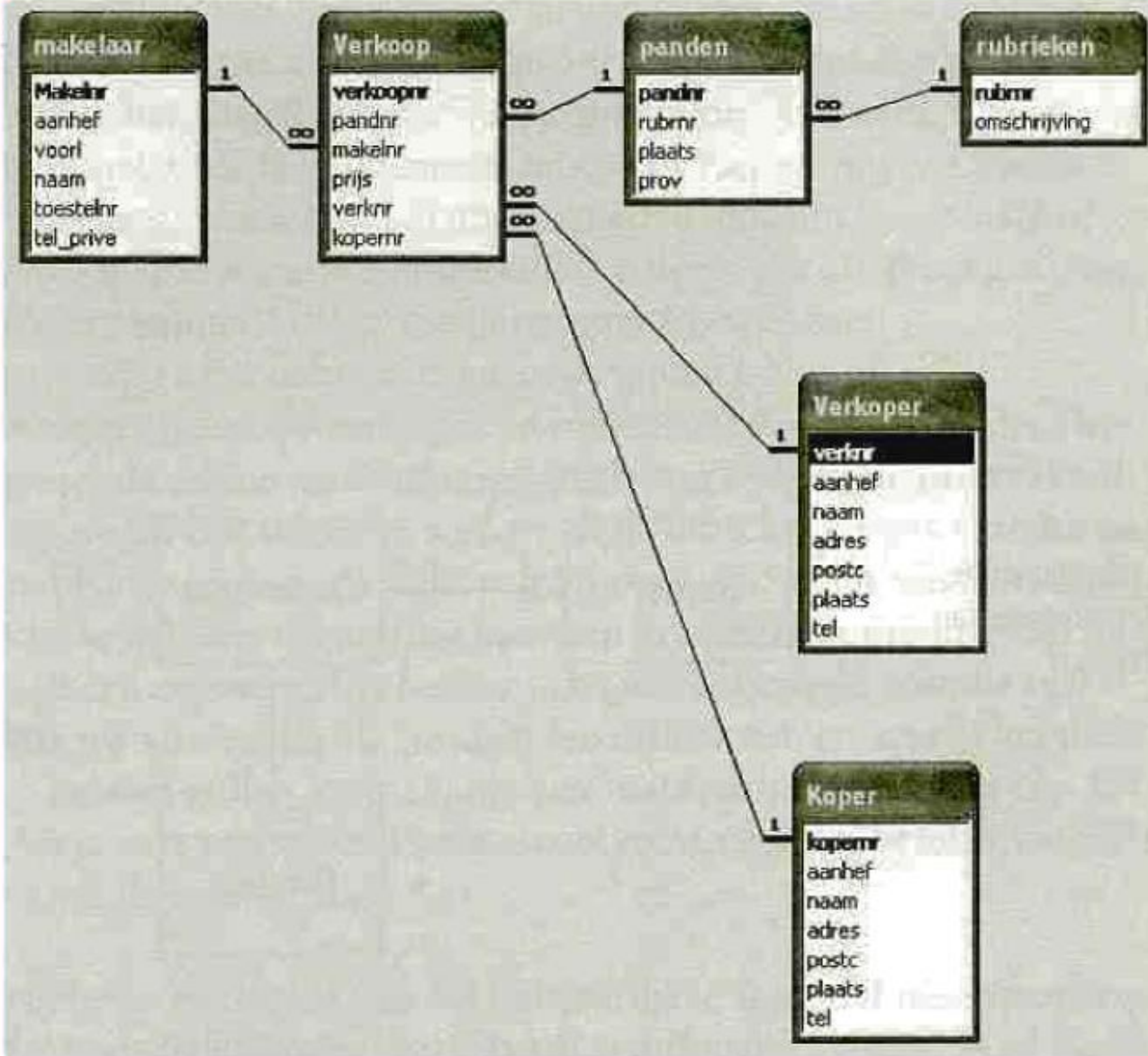
De leraren zullen niet blij zijn met dit resultaat, want er staat hier dat een leraar een onbeperkt aantal lessen per week kan geven.

Toepassing op databases

Ook de inhoud en de relaties van een database zijn zo in een klassendiagram onder te brengen. Velen zullen voor het weergeven van de opbouw van een database nog het klassieke Entity Relation Diagram (ERD) gebruiken, maar een klassendiagram kan dezelfde functionaliteit weergeven. We gaan hier dus niet in op het ERD, maar houden ons bij het klassendiagram. De kracht van het relationele model voor databases zit in de één-op-één en één-op-meer relaties van de tabellen in de database. In ieder geval levert een meer-op-meer relatie altijd grote problemen op. Elke tabel is een klasse en bij de associaties geven we de multipliciteiten aan.

Opmerking

Bij de ICT-opleidingen van het Radius, gebruiken we de klassendiagrammen vooral om het model van een relationele database op papier te zetten.



Oefening 1

Gegeven is de volgende context:

- Maak een tabel van kandidaatsklassen en de beslissing die je daarover neemt.
- Maak van je uiteindelijke klassen een klassendiagram met attributen en operaties.
- Geef de relaties en multipliciteiten aan tussen de klassen.
- Bij de afsluiting zal je gevraagd worden om je klassendiagram te presenteren, waarbij je duidelijk moet aangeven, waarom je het op jouw manier hebt getekend/beschreven.

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 betaalt. 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 verwijderd.

Oefening 2

Gegeven is de volgende context:

- Maak een tabel van kandidaatsklassen en de beslissing die je daarover neemt.
- Maak van je uiteindelijke klassen een klassendiagram met attributen en operaties.
- Geef de relaties en multipliciteiten aan tussen de klassen.
- Bij de afsluiting zal je gevraagd worden om je klassendiagram te presenteren, waarbij je duidelijk moet aangeven, waarom je het op jouw manier hebt getekend/beschreven.

Internetbedrijf SMALL

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

- Inkopen van kleding bij leveranciers: SMALL heeft verschillende leveranciers waarbij ze kleding inkoopt. 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 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 Voorraadbeheer opgeborgen en de gegevens van de bestelling worden geregistreerd in het systeem. De factuur gaat naar de afdeling Financiën, die deze factuur betaalt. 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 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 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 verwijderd.