

Verskil functioneel- en technisch ontwerp, zie ook internet waar veel documentatie staat als het gaat om de verschillen tussen een functioneel ontwerp en een technisch ontwerp.

*Vaak wordt het functionele ontwerp verward met een technisch ontwerp. **Waar een functioneel ontwerp de details geeft van wat een systeem moet doen, geeft een technisch ontwerp (TD) de details van hoe de software de acties moet uitvoeren.***

3.1 De opdracht vaststellen

Je wordt opgeleid om een programma te maken in opdracht van een klant. Daarom goed om duidelijk te communiceren, zodat je betere beslissingen kunt nemen, je kunt aansluiten bij de ander en je een oplossing kan zoeken. Mensen denken vaak dat iemand anders het toch wel begrijpt. Goed luisteren, samenvatten en doorvragen.

3.2 Luisteren samenvatten en doorvragen

Luisteren: Goeie vragen stel je door te luisteren, lichaamshouding aantekeningen

Voordeel van luisteren is dat andere ook terug zullen luisteren en openstaan voor jouw mening.

Samenvatten: Duidelijk maken wat er zojuist is besproken. Alles duidelijk? Dan maak je de belangrijkste aantekeningen.

Doorvragen: Let op wat de ander zegt en wat hij niet zegt, let op moeten of kunnen.

3.5 Programma van eisen

MoSCoW-methode de o's hebben niks te zeggen

M: Must have

S: Should have

C: could have

W: Wont have

Hiermee wordt duidelijk wat er is afgesproken. Alle must have's moeten verwerkt zijn.

3.7 Rapport programma van eisen

Wanneer alles duidelijk is is het verstandig om een rapport te maken. Dit geeft een goede indruk op de klant, een rapport moet bestaan uit

Voorblad

Inhoudsopgave

Samenvatting

inhoud van het rapport

samenvatting.

Het is belangrijk dat alles hierin staat om te voorkomen dat er wat mis gaat.

5.0 Functioneel ontwerp

Functioneel ontwerp is blauwprint van app/website

Technisch ontwerp is hoe de software gemaakt gaat worden.

5.1 Requirements

Dit omschrijft wat er elementen en functies nodig zijn per project.

Business requirements:

groter marktaandeel

hogere efficiëntie

meer omzet

meer tevreden klanten

lagere kosten

kortere inwerktijd

minder risico's

user requirements: Wat gaan de gebruikers met het systeem doen

functionele requirements: Wie gaat van welke functie gebruik maken, en welke functies zijn nodig

niet functionele requirements: Hier gaat het over betrouwbaarheid, snelheid, onderhoud en veiligheid

5.2 Unified modeling language (UML)

UML helpt het systeem te visualiseren en er over te communiceren.

UML

Statische diagrammen:

- Klassendiagram *

- objectendiagram

- componentendiagram

- gebruikendiagram

Dynamische diagrammen:

- use-case-diagram *
- collaboratiediagram
- volgordediagram *
- activiteitendiagram *
- correlatiediagram
- toestanddiagram

5.3 Use-case-diagram

Voorbeeld: Lijn met include -----> Betekend vereist

Als de klant keuze heeft ----> Extend

use-case bestaat uit: Systeemgrens, actor, use case en verbinding (include exclude)

5.4 De use-case-tabel

actor: wie is betrokken, vaak is er 1 actor maar meerdere is ook mogelijk

precondities: voorwaarden om de handeling te doen

scenario: hier omschrijf je zo volledig en beeldend mogelijk wat er gebeurt in volgorde

uitzondering: hier beschrijf je wat er gebeurt als de voorwaarden niet goed zijn of andere fouten

niet functionele eisen: betreft snelheid, veiligheid, betrouwbaarheid en onderhoud

postconditie: De situatie omschrijven nadat de handeling is voltooid

voorbeeld:

Scenario:

A Candy Store sells candies. Based on the information in Table 1, draw a **use case diagram** and a **class diagram**. If you discover while drawing the diagram that the scenario is incomplete, make up reasonable explanations to complete the story. Supply your explanations along with the diagram.

Table 1. Information for Candy Store

Name:	Purchase Candies
Actor:	Customer/Employee
Description:	This describes the process used to purchase candies at Morehead Candy Store
Successful Completion:	<ol style="list-style-type: none">1. Customer requests candies2. Employee checks on availability of candies3. Candies are available and customer pays4. Customer receives candies and stock is updated
Alternative:	<ol style="list-style-type: none">1. Customer requests candies2. Employee checks on availability of candies3. Candies are not available and customer selects alternate candies or no candies4. Customer receives candies and stock is updated (if alternate candies selected) or customer leaves with no candies
Pre-Condition:	Customer wants to purchase candies
Post-Condition:	None
Assumptions:	None

5.6 Klassendiagram

Klassendiagram wordt gemaakt in groepen bijvoorbeeld -> Bestek: Lepels, vorken en messen.

Classname: Specificeert over wat voor object het gaat. Moet een unieke naam en altijd enkelvoud zijn.

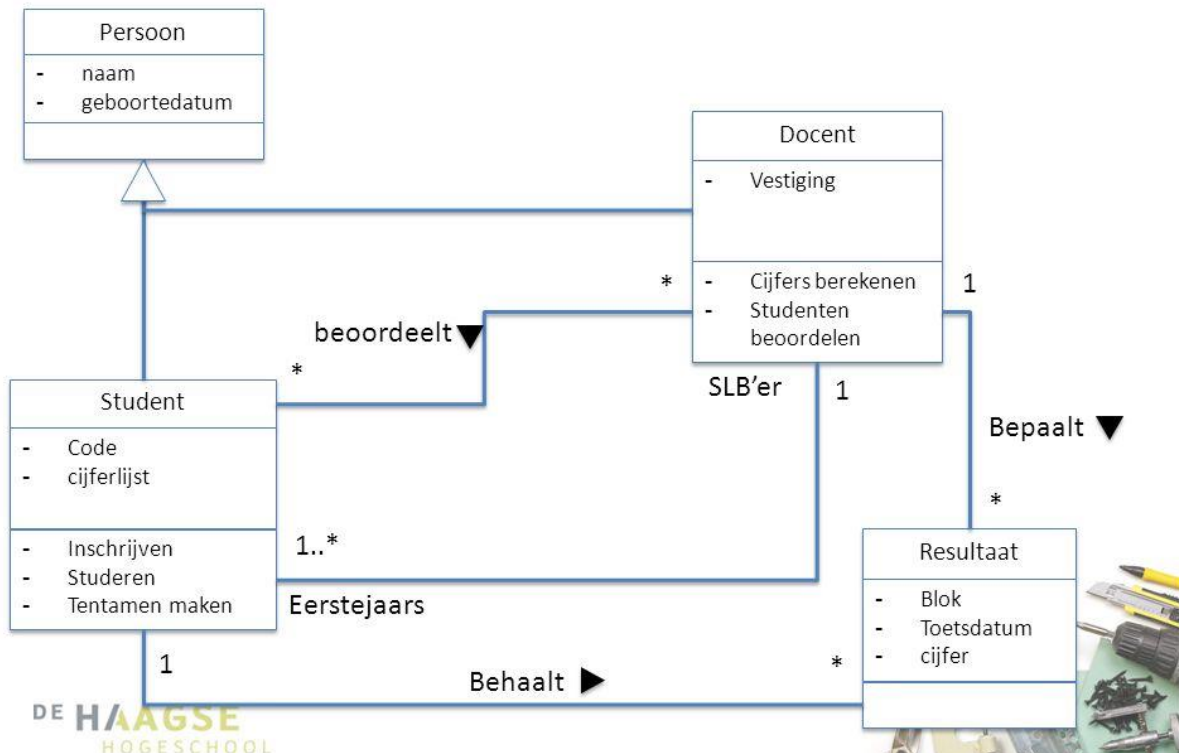
Attributen (fields): Omschrijft het object dus bijvoorbeeld van een auto een wiel.

Methods: functies die mensen kunnen uitvoeren op het object (classname)

5.7 Relaties in een klassendiagram

Klassendiagrammen kunnen relaties met elkaar hebben bijvoorbeeld een student en een adres. Zo leeft de student bijvoorbeeld op dit adres.

Case 2: Studeren



Compositie aggregatie: Voorbeeld: Je wilt een boterham, hiervoor heb je nodig -> brood nu KAN je er boter of beleg op doen. Dus dit is niet noodzakelijk

Klassendiagram wordt ontworpen op basis van de beschrijving van een systeem. Je begint met alle zelfstandige naam woorden aan te duiden. Vervolgens ga je ze ordenen en stop je ze in groepen. Bijvoorbeeld Woensdag -> AGENDA en iemand -> LID, positie -> RANGLIJST

5.9 Functioneel-ontwerp-rapport

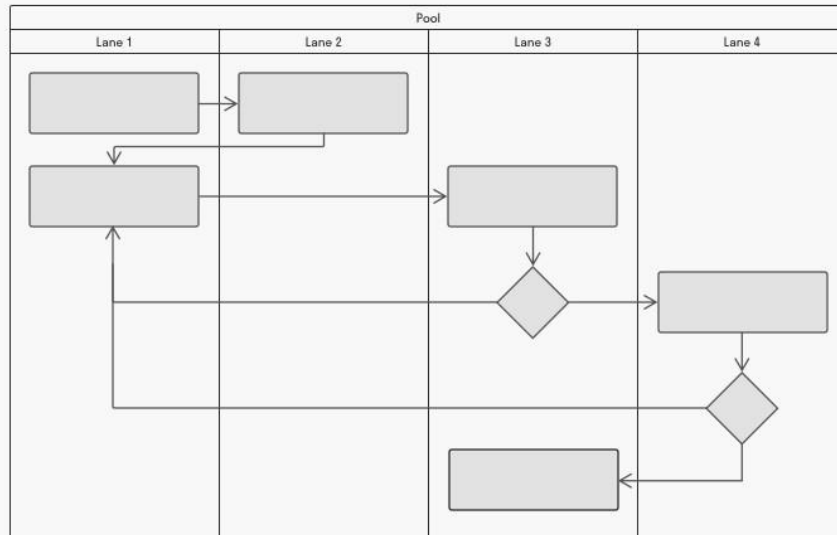
Is een document die bedoeld is voor de klant/opdrachtgever. Voorblad
 Voorwoord Hier schrijf je waarom je dit document gemaakt hebt, je benoemt de klant/opdrachtgever en het bedrijf waar jij voor werkt
 Inhoudsopgave Samenvatting Analyse huidige situatie
 Informatieverwerking: Hoe zit alles in elkaar Applicaties: Wat wordt er gebruikt? Maak een use-case Infrastructuur: Is er een netwerkinfrastructuur moet deze grafisch worden weergegeven Analyse gewenste situatie de MoSCoW methode. En opnieuw de huidige situatie elementen
 Consequenties: Wat veranderd er? wat moet er komen/vervangen worden
 Kosten Planning

6.0 Technisch ontwerp

Wanneer de functionaliteiten klaar zijn kunnen de technische randvoorwaarden gemaakt worden. Dit doen we met een technisch ontwerp. Het technisch-ontwerp-rapport is het overdrachtsdocument. Bij technisch ontwerp wordt bepaald hoe de software gemaakt gaat worden. Welke technieken kiezen we? Waar wordt de data opgeslagen? Hoe worden ingevoerde gegevens verwerkt? Er moet rekening gehouden worden met Hoeveelheid gegevens Toegankelijkheid van gegevens Toegangsnelheid verwerkingssnelheid onderhoud randvoorwaarden: beveiliging van website ontwikkelomgeving/progammeertalen structuur van database technische infrastructuur Om alles goed uit te voeren gebruik je UML-diagrammen, activiteitendiagram en sequentiediagram.

6.1 activiteitendiagram

Je kunt een user-story omzetten in activiteitendiagram Symbolen in een activiteitendiagram: start activity decision merge (balk waarin de stromen na een beslissing samen komen) fork (voor parallele samenwerking) join (samenkomst van parallele samenwerking) end maken van activiteitendiagram: kies een use-case teken swimlane



www.toolshero.nl

teken voor elke interactie tussen de actoren en systeem de activiteit. plaats decision merge fork en joins teken de transitie pijlen tussen alle elementen

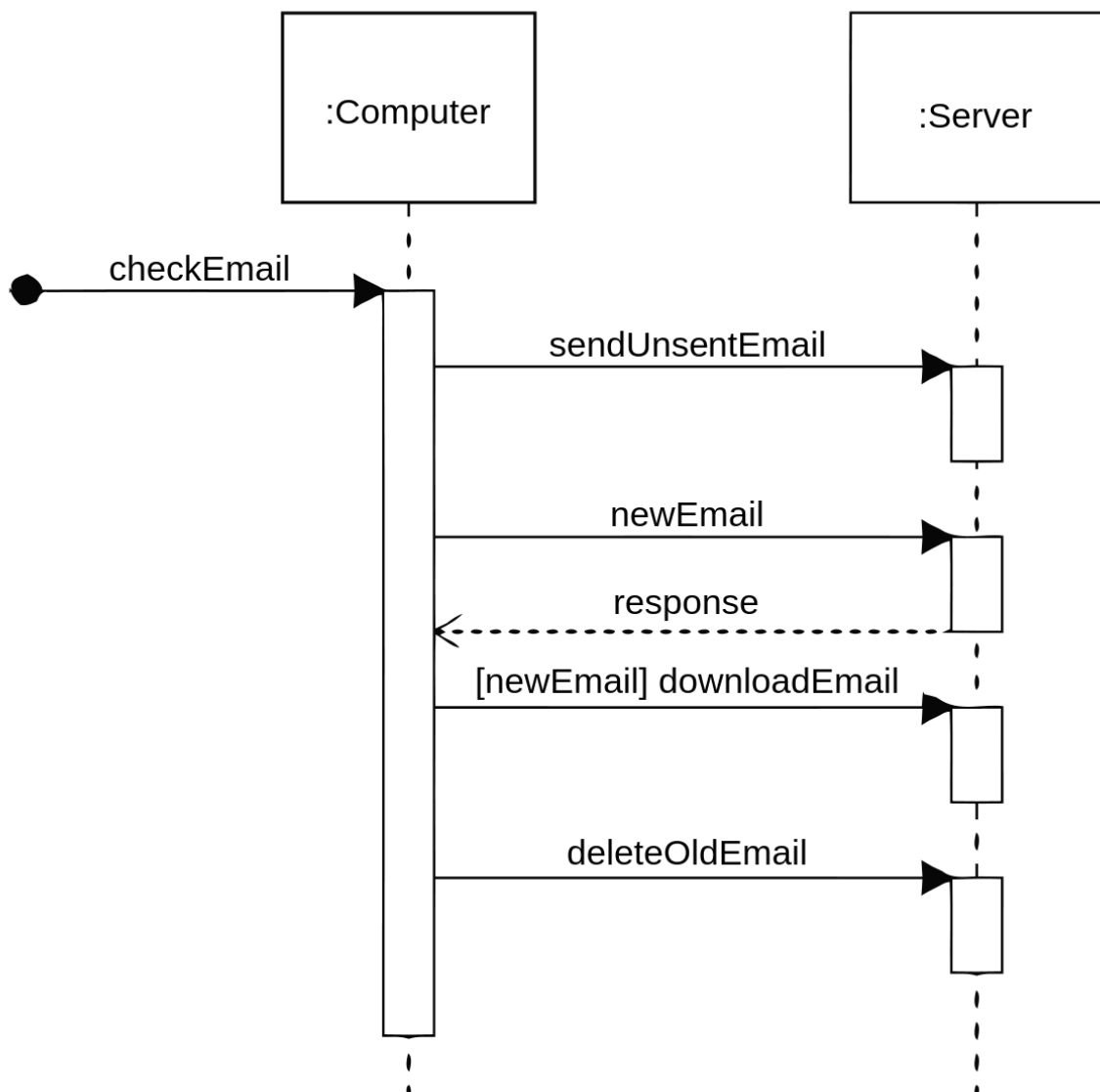
Een artikel aan Wikipedia toevoegen:



voorbeeld

6.3 Sequence diagram

sequencediagram is een stroomdiagram voorbeeld:



11.1 Voorbereiden van de presentatie

Bepaal voor welke doelgroep je gaat presenteren zodat je aan de hand daarvan de informatie kan opstellen

Bepaal de boodschap van je presentatie

Goeie eerste indruk leveren

Vertel het in een vorm van een verhaal

Sluit goed af, maak hierbij keuze uit een samenvatting, een vraag, een stelling of een belofte

11.2 De presentatie maken

niet te veel informatie in een diastelling zetten. Maak een presentatie met powerpoint
Gebruik leuke animaties en effecten