

# KT1

- [Planning](#)
- [1 Inleiding en overzicht](#)
- [2 Programma van Eisen](#)
- [3 Projectplan](#)
- [4 Functioneel Ontwerp](#)
- [5 Technisch Ontwerp](#)
- [5.0 ERD in het kort](#)
- [5.1 ERD - oefeningen](#)
- [6 Configuratieoverzicht](#)
- [Tip Top 10](#)

# Planning

## Planning

Datum	Onderwerp 1	Onderwerp 2	Huiswerk
27 aug (korte les)	Introductie KT1	Introductie Yii	Intro Yii en intro KT1
3 sep	PvE	Yii Crud	PvE - Yii Crud (les 1)
10 sep	PP	ERD 1	PP en ERD (opgaven 1 en 2)
17 sep	FO	ERD 2 (uitleg)	FO en ERD (opgave 5)
24 sep	FO	ERD 3 toets	FO en ERD (opgave 5)
1 okt	TO	Yii Crud	TO incl ERD
9 okt	Yii	ERD toets II	Yii les 1 t/m 4
15 okt	Vakantie	-	Proefexamen - start
22 okt	Proefexamen PvE en PP		PvE en PP
29 okt	Proefexamen FO		FO
5 nov	Proefexamen TO		TO
12 nov	Laatste voorbereidingen en check		

## 9 oktober

### Toets (30 m.)

Socrative multiple choice, toets vragen over DB ontwerp/ ERD tweede keer. Doel is om de ontwikkeling te meten en om weer even met ERD's bezig te zijn. Duurt 30 minuten.

### Instructie

- Ga naar socrative.com,
- kies login,
- selecteer Student Login
- room name, <to be determined> (is pas bekend en actief tijdens de les)
- dan voor- en achternaam invullen
- 18 multiple choice vragen, vrije navigatie (voor- en achteruit) en aan het eind krijg je je score.
- Als je klaar bent dan ga je verder met Yii (waar je bent gebelven).

## Uitleg TO (15 min)

Use Case schema's zijn al uitgelegd, wil ik niet te lang bij stil staan. Ik wil vooral naar het ERD kijken. Ik heb een voorbeeld ERD gemaakt van de case en daarbij heb ik de maximale versie gemaakt.

## Configuratieoverzicht (15 min)

Bespreken laatste werkproces van KT 1: <https://www.roc.ovh/books/kt1/page/6-configuratieoverzicht>

Maken is vrijwillig en handig want dan hebben ze dat vast, het overzicht is namelijk generiek.

## Yii (90 min)

Zelfstandig werken aan Yii <https://www.roc.ovh/books/yii>

Les 1 moet echt af want daarmee bewijzen de studenten dat ze de ontwikkelomgeving hebben draaien.

Als er behoefte aan is kan ik een intro geven over routing (les 2) in Yii.

In principe kan les 1 t/m 7 helemaal zelfstandig worden doorgewerkt en biedt dat de basis van waaruit je verder kan bouwen.

## Online les 5 nov

klaarzetten: presentie, beoordeling FO, voorbeeld en tool URL

### **presentie**

FO - serieus gemaakt.

Meest gemaakte fout missen van formulieren

FO-beoordeling: **functionaliteiten** (duidelijk, prio, , **schematechnieken (relaties** tussenschermen is duidelijk, welke gegevens nodig- **formulieren!**) in- en uitvoerschermen!

Voorbeeld laten zien <http://www.phpform.org/>

1:1 wie?

## Laatste les

- 30 minuten ERD maken - indien af ERD-oefening fietsenmaker.
- Bespreken ERD, check of hoofdfuncties terug te vinden zijn.
- FK's !
- Oefenen:
  - opdracht inleveren in Teams
  - examen.student.ovh allemaal een gesprek aanvragen - aanvraag, wordt uit de klas gehaald, naar kantine, ik zie je zitten en de rolspeler belt jou via Teams.
- Gesprekken, wil iemand oefenen?
- Tip Top 10
- Geen uitgewerkte opdrachten (sorry)

# 1 Inleiding en overzicht

*In deze les krijg je een overzicht van wat kerntaak 1 van het examen applicatieontwikkeling inhoud.*

Kerntaak-1 examen AO omvat alle taken die je kunt uitvoeren **voordat** je aan coderen begint.

Je leert hoe je de samen met de opdrachtgever de afspraken maakt en hoe je samen met de klant en jouw leidinggevende komt tot een ontwerp op papier.

Het examen bestaat uit vier werkprocessen

1. B1-K1-W1, Stelt de opdracht vast (vraaggesprek, PvE)
2. B1-K1-W2, Levert een bijdrage aan het projectplan (Projectplan)
3. B1-K1-W3, Levert een bijdrage aan het ontwerp (FO/TO)
4. B1-K1-W4, Bereidt de realisatie voor (Inrichten testen en documenteren dev. omgeving)

Om beter te begrijpen waarom en hoe we dit allemaal moeten doen gaan we eerst een paar korte oefeningen doen.

## Opdracht 1

(schooljaar 2021/22)

Begin deze les met het maken van de mini-module::

[https://xerte.talnet.nl/play.php?template\\_id=842](https://xerte.talnet.nl/play.php?template_id=842)

Download aan het eind de PDF met jouw resultaten en zet die in Teams.

## Opdracht 2, Keuken

Stel je wilt een keuken kopen, en je gaat naar een keuken-winkel, hoe gaat dat dan? Stel dat jij in de winkel staat en een klant komt binnen en vraagt om een nieuwe keuken, wat zou jij de klant dan allemaal willen vragen?

Uiteindelijk vraagt de klant: "Hoeveel kost dat?". Als je antwoorden op al je vragen hebt gekregen kun je de klant dan vertellen wat de nieuwe keuken gaat kosten?

Schrijf al je vragen op en bespreek die in een groep.

Als alle vragen beantwoord zijn heb je dan genoeg informatie om:  
een **prijs** en een **planning** af te spreken?

## Prijs

Dit is wat de keuken kost, kosten bestaan uit: keuken-onderdelen en alle uren die moeten worden besteed om de keuken te bestellen, af te leveren en te installeren.

## Planning

De planning moet duidelijk maken wanneer de keuken kan worden geplaatst en wanneer alles klaar is.

# Software Development

Als software developer leveren we natuurlijk geen keukens, maar een software product, of een "IT Solution".

Toch zijn er veel overeenkomsten tussen het bestellen van een keuken en het laten ontwikkelen van een software product.

Wat moet je allemaal weten als je een software product gaat bouwen?

## Opdracht 3, Software bouwen

Een hip koffie-tentje in Amsterdam wil proberen meer klanten te krijgen. Er worden hele speciale koffiebonen gebruikt voor de koffie en die bonen worden direct van de koffieboeren gekocht. De bonen worden met zeilschepen op een milieu vriendelijke manier naar Amsterdam vervoerd. Het probleem is dat de klanten van dit hippe koffie-tentje dit niet weten en jouw opdrachtgever wil iets met een web-site doen om zijn verhaal te vertellen en zijn koffie-tentje te promoten. De opdrachtgever wil ook graag iets doen op zijn website, zodat bezoekers terugkomen. Hij wil via de nieuwe website zijn klanten aan zich binden. Zo wil hij wekelijks een bepaalde koffiesoort onder de aandacht brengen en daar via de website korting op geven.

Wat moet jij allemaal weten om een prijs en een planning te bepalen.

Schrijf minimaal vijf vragen op die jij aan deze klant zou stellen. Zet deze vragen in een text document (notepad) en zet dit documentje ook in Teams.

## Structuur

Je hebt nu heel veel informatie van de klant en om daar structuur in te brengen maken we een onderverdeling; we maken alle afspraken stapsgewijs.

## Stap 1, Programma van Eisen (werkproces 1, 115 m.)

Door een vraaggesprek te houden, stellen we vast wat de opdracht op hoofdlijnen inhoud. We proberen te achterhalen welk probleem de klant wil oplossen en hoe hij die oplossing voor zich ziet.

## Stap 2, Projectplan (werkproces 2, 120 m.)

Je stelt een gedetailleerd projectplan op. Dit bespreek je met de klant.

## Stap 3, FO/TO (werkproces 3, 2x 210 minuten)

Je maakt een functioneel ontwerp. Deze bespreek je met je klant. Het functioneel ontwerp beschrijft hoe de software er uit gaat zien en hoe het werkt.

Je maakt een technisch ontwerp. Dit bespreek je met je projectleider. Het technisch ontwerp beschrijft hoe de software technisch in elkaar zit.

## Stap 4, Maak ontwikkelomgeving en beschrijf deze (werkproces 4, 60 minuten)

Deze stap behoort niet echt meer bij het ontwerp, maar is meer een voorbereiding op het ontwikkelproces. Het ontwikkelproces vindt plaats bij kerntaak 2.

In de volgende lessen gaan we alle werkprocessen en opdrachten oefenen.

--

# 2 Programma van Eisen

## Wat gaan we leren?

*De eerste opdracht van kerntaak 1 is het maken van een programma van eisen. Je voert hiervoor twee gesprekken en maakt een document. In deze les leren we hoe je je voorbereidt op de gesprekken, hoe je de gesprekken voert en wat er in het programma van eisen moet staan. We gaan ook goed kijken hoe je precies wordt beoordeeld.*

*Ten slotte gaan we oefenen. We gaan ons voorbereiden op een gesprek en we gaan dit in de klas oefenen. Daarna gaan we allemaal (thuis) werken aan ons eigen PvE (Programma van Eisen).*

## Examen-template

We beginnen gelijk met de deur-in-huis en gaan kijken naar de examen-template; wat moet er in het PvE komen te staan?

De onderstaande indeling is voor een groot deel gekopieerd uit de template die je krijgt op het examen.





## Inleiding

In de examen-template is deze vaak al ingevuld; besteed hier dus niet te veel tijd aan).

## Bedrijf

In de examen-template is deze vaak al ingevuld; besteed hier dus niet te veel tijd aan.

In deze paragraaf staat een korte beschrijving van het bedrijf en de aanleiding wordt kort beschreven.

## Probleemstelling

*Beschrijf kort waarom er een nieuwe applicatie wordt gebouwd. Geef een algemene beschrijving van de applicatie en van de belangrijkste doelen van de applicatie. Betrek hierbij ook de consequenties voor de organisatie en haar medewerkers.*

Beschrijf het probleem dat moet worden opgelost (dus niet de oplossing).

## Doelgroepen

*Beschrijf voor wie de applicatie bestemd is.*

Wie gaat het systeem, de website gebruiken. Beschrijf de rollen.

TIP: gebruik je (klad) ERD en denk in CRUD

## De vormgeving

*Beschrijf de vormgeving van de website (kleuren, lettertypes). Gebruik eventueel een vlekkenplan om de grove indeling van de vensters weer te geven (Een vlekkenplan is een ruwe voorlopige indeling.)*

Geef een algemene indruk van hoe de site er uit gaat zien. Kleur en stijl zijn belangrijk. Je kunt eventueel naar een voorbeeld website verwijzen. Maak ook een schets. Of dit erg belangrijk is, hangt ook van de opdrachtgever. Soms worden er hele duidelijke eisen gesteld aan de vormgeving en soms ligt dat nog open.

## Informatie

*Beschrijf globaal welke informatie door de applicatie gegeven wordt en welke overzichten en rapportages door de applicatie gegenereerd worden.*

Beschrijf alle eisen, zeg maar alles wat de website moet kunnen. Je kunt dit in use-case formaat doen.

TIP: beter een eis te veel dan te weinig!

TIP: denk in CRUD, je hebt je (klad) ERD al gemaakt. Ga uit van CRUD op de tabellen.

## Overig

*Geef hier andere relevante informatie, indien nodig.*

Beschrijf hier de niet-fuctionele eisen (security, performance, backup etc.) en geef eventueel ook wat je *niet* gaat doen binnen dit project.

# Beoordeling Examen

In totaal moet je 9 punten (van de 17) halen en je moet minimaal 1 punt voor je cruciale criterium halen.

Het officiële [beoordelingsformulier](#) is bij deze pagina toegevoegd.





## Vorbereiding gesprek

(0 of 1 punt)

Je hebt je wel of niet voorbereid.

## Gespreksvorm

(0 tot 3 punten)

1. je leidt het gesprek in
2. je stel vragen en als het nodig is vraag je door
3. je gaat in op wat de opdrachtgever zegt
4. je vat het gesprek samen en legt uit hoe het proces verder gaat.

## Gespreksinhoud

(0 tot 3 punten)

1. de vragen zijn altijd goed (hebben betrekking op de opdracht).
2. je gebruikt weinig vaktaal en legt alles begrijpelijk uit.

## Inhoud van het programma van eisen

(0 tot 3 punten **cruciaal**)

Het gaat hier om de eisen die je hebt opgeschreven die moeten minimaal in de meeste gevallen goed aansluiten bij de opdracht en je mag maar een paar eisen vergeten zijn. Je mag ook niet te veel vaktaal gebruiken.

## Vorm van het programma van eisen

(0 tot 3 punten)

Alle onderdelen zijn goed en duidelijk beschreven.

Tips: Zorg dat alles er **verzorgd** uit ziet. Haal ook de template-aanwijzingen weg en vul je naam netjes in. geef tenslotte het document ook een duidelijke naam.

## Onderbouwing programma van eisen

(0 tot 3 punten)

Er is goed gebruik gemaakt van aanvullende bronnen en/of onderzoek. Je moet dus laten zien dat je actief op zoek bent gegaan naar oplossingen. Je beschrijft duidelijk wat mogelijk is maar ook als er iets niet mogelijk is.

## Communiceren programma van eisen

(0 of 1 punten)

# Aanvullende bronnen

Boek: Applicatie- en mediaontwikkelaar door Hans van Rheenen, hoofdstukken 3.1, 3.2, 3.4, 3.5

## Opdracht

Lees hieronder project Windmee of Kinderopvang (docent geeft aan welke).

In de klas zullen we het gesprek oefenen.

Na het gesprek stel je een PvE op. Gebruik hiervoor deze template.

Lever het PvE in via Teams.

### Project Zeilschool WindMee

Zeilschool WindMee is een bedrijf met 2 directeuren, 2 administratief medewerkers en een 10-tal instructeurs. Bij de zeilschool kan je boten huren of leren zeilen. Tijdens een cursus van een week leren cursisten de fijne kneepjes van het vak of worden hun capaciteiten verbeterd. Voor dit onderdeel moet een nieuwe website komen.

Naast een contactpagina en een home pagina met algemene informatie moet een cursist zich kunnen inloggen (na eventuele registratie). Een cursist kan zich daarna inschrijven voor één of meerdere cursussen. Er zijn 3 typen cursussen: beginners, gevorderd of expert. Elke cursus duurt een week en begint op zaterdag.

Als een cursus vol zit kan een cursist zich daar niet meer voor inschrijven. Bij de inschrijving ziet een cursist ook hoe duur een cursus is en hoeveel hij totaal kwijt is. Er wordt een nota gegenereerd in PDF.

Medewerkers van zeilschool kunnen, als zij ingelogd zijn, cursussen toevoegen, wijzigen en eventueel (bij geen deelnemers) verwijderen. Medewerkers kunnen een deelnemersoverzicht per cursus opvragen. Voor elke cursus zijn schepen nodig. Voor de beginners cursus is dat per 4 deelnemers 1 schip, voor de gevorderden en experts 2 per schip. Medewerkers kunnen zien welke schepen er beschikbaar zijn (een schip kan in onderhoud zijn). Bij het begin van de cursus is bekend welke schepen er gebruikt gaan worden. Elk schip heeft een naam en een unieke code (WM- gevolgd door 2 cijfers).

De administratie kan medewerkers toevoegen, wijzigen en verwijderen en op verzoek de gegevens van een cursist aanpassen. Ook heeft de administratie inzicht in alle cursisten.

## Project Kinderopvang

Je bent als applicatieontwikkelaar werkzaam bij internetbedrijf FastDevelopement. Dit bedrijf maakt in opdracht maatwerk (web)applicaties voor haar opdrachtgevers. Aan jou is gevraagd een webapplicatie te ontwikkelen voor de administratie van Kinderopvang Klavertje Vier.

Kinderopvang Klavertje Vier is een kinderdagverblijf (KDV) gevestigd in Diemen Noord, die de opvang verzorgt voor 0 tot 4 jarigen. Ouders brengen hun kinderen wekelijks 1 of meer dagen naar het KDV. Er zijn momenteel drie groepen, Loor, Kanga en Poeh. Per groep is ruimte voor maximaal 10 kinderen en per groep zijn 2 leidsters. Ouders hebben per kind een contract met het KDV waarop staat op welke weekdagen zij hun kind brengen. Indien een ouder een contract wil voor meer dagen per week en er is geen plek vrij, dan komt dat kind voor die dag(en) op de wachtlijst. Indien er een plek vrij komt, dan hebben deze kinderen voorrang boven nieuwe kinderen.

Onlangs is er veel nieuwbouw in de buurt gerealiseerd. Hier zijn veel jonge gezinnen komen wonen. Hierdoor is de vraag naar kinderopvang dermate toegenomen dat het bestuur van Klavertje Vier heeft besloten om een extra groep te starten. Deze uitbreiding heeft ook geleid tot een toename van het administratieve werk. Om de administratie efficiënt te laten verlopen heeft het bestuur besloten om hiervoor een webapplicatie te laten ontwikkelen.

Directeur Boris Brand van Klavertje Vier heeft de volgende aanvraag bij FastDevelopement ingediend: Wij hebben behoefte aan een webapplicatie waarmee we op eenvoudige wijze de administratie van het kinderdagverblijf kunnen bijhouden.

Deze applicatie moet ons de volgende functionaliteit bieden:

- Een alfabetische lijst van alle kinderen van het KDV met voornaam, achternaam, geboortedatum, naam van de ouder en telefoonnummer en bijzonderheden.
- Een weekoverzicht met per dag welke kinderen verwacht worden.
- Een weekoverzicht met per dag welke kinderen naar het KDV komen: Kinderen worden in principe op de afgesproken dagen naar het KDV gebracht, maar ze komen niet als ze ziek zijn of op vakantie zijn. In dat geval kan een ouder hun kind een extra dag brengen. Een extra dag brengen is bestemd voor de kinderen die in de afgelopen tijd een of meer dagen gemist hebben omdat zij ziek waren of op vakantie.
- Eens per kwartaal een uitdraai van de nota's volgens de bijlage. Hierop worden de contractueel vastgelegde aantal dagen in rekening gebracht.
- Een uitdraai van de wachtlijst. Op de wachtlijst staan de reeds geplaatste kinderen (die meer dagen willen) en nieuwe kinderen die nog niet op het KDV zitten.

# 3 Projectplan

## Wat gaan we leren?

*De tweede opdracht van kerntaak 1 is het maken van een projectplan. Je voert hiervoor één of twee gesprekken en maakt één document. Je begint met het maken van het projectplan.*

*Uitgangspunt hierbij is het PvE.*

*Je bespreekt het projectplan met de opdrachtgever en als er opmerkingen zijn dan verwerk je die. Daarna lever je de definitieve versie van het projectplan in.*

*In deze les bespreken we wat er in het projectplan moet komen te staan en hoe je wordt beoordeeld. Ten slotte gaan we oefenen met de cases die we al kennen vanuit de vorige les.*

## Examen-template

We beginnen gelijk met de deur-in-huis en gaan kijken naar de examen-template; wat moet er in het PvE komen te staan?

De onderstaande indeling is voor een groot deel gekopieerd uit de template die je krijgt op het examen.





## Inleiding

In de inleiding van het projectplan staat voor welk bedrijf een applicatie wordt gemaakt, wat het bedrijf globaal doet en wat de opdracht is. Uiteraard kunnen hier nog andere onderwerpen aan toegevoegd worden.

Voorbeeld

Dhr Muntjes van bedrijf ABC.com heeft aSoft de opdracht gegeven om een webapplicatie te bouwen waarmee zij reserveringen kunnen maken voor ..... In dit projectplan wordt een planning gemaakt voor dit project.

Tip: 'be smart' en gebruik de inleiding vanuit het PvE als basis. Houd het kort en krachtig.

## Doelstelling

Waarom doen we dit? Wat willen we bereiken?

Voorbeeld

ABC.com gaat uitbreiden waardoor de administratie van de reserveringen van xxx zoveel werk wordt dat er een extra medewerker moet worden aangenomen. De handmatige administratie leidt verder tot fouten en is lastig toegankelijk voor alle medewerkers. Een nieuw te ontwikkelen applicatie zou ervoor moeten zorgen dat de uitbreiding met de huidige capaciteit kan worden uitgevoerd en dat er minder fouten zouden worden gemaakt. Daarbij is het dan ook eenvoudiger voor alle medewerkers

## Benodigheden

Wat heb je nodig om het project uit te voeren? Maak een opsomming van zaken die je nodig hebt, te denken valt aan: technische IT-middelen (computer e.d.), software, netwerk en werkplek.

## Takenlijst

Dit is een opsomming van taken. Dit zijn de taken op de hoofdpunten die later in de planning, in het volgende hoofdstuk, verder worden gespecificeerd.

- Ontwerpen van de applicatie
  - Projectplan, opstellen, toelichting geven, eventueel verwerken feedback en overdracht naar opdrachtgever
  - Functioneel ontwerp, opstellen en overdracht aan opdrachtgever
  - Technisch ontwerp opstellen en overdracht aan opdrachtgever
  - Inventariseren benodigde hard- en software voor ontwikkelomgeving,

# Beoordeling Examen

In totaal moet minimaal 5 (van de 11) punten behalen en er zijn geen cruciale criteria.

Het officiële [beoordelingsformulier](#) is bij deze pagina toegevoegd.

## Project Doelstelling

(0 tot 3 punten)

De projectdoelstellingen zijn in lijn met de opdracht, ze zijn duidelijk beschreven en ze zijn reëel (=haalbaar).

## Projectactiviteiten

(0 tot 3 punten)

1. alle werkzaamheden zijn volledig, correct en overzichtelijk weergegeven.
2. benodigdheden zijn beschreven
3. de ontwikkelmethode is benoemd; dat zijn alle stappen die je gaat uitvoeren:  
Projectplan schrijven, projectplan toelichten en aanpassen, functioneel ontwerp schrijven, functioneel ontwerp toelichten, technisch ontwerp opstellen, technisch ontwerp uitleggen en aanpassen, ontwikkelomgeving inrichten en documenteren.  
(Deze methode is wat ouderwets en wordt de *watervalmethode* genoemd; weet je waarom deze naam zo is gekozen?)

## Planning

(0 tot 3 punten)

De planning is inclusief voortgangsgesprekken volledig, realistisch en overzichtelijk weergegeven. Denk aan de verzorgde uitstraling!

## Afstemmen / goedkeuring

(0 of 1 punt)

Het projectplan wordt afgestemd en er wordt om **goedkeuring** gevraagd.

Je moet minimaal 5 (van de 11) punten behalen en er zijn geen cruciale criteria.

# Opdracht

Maak een projectplan en gebruik het gemaakte PvE uit de vorige les als uitgangspunt.

--

# 4 Functioneel Ontwerp

## Wat gaan we leren?

*De derde opdracht van kerntaak 1 is het maken van functioneel en technisch ontwerp. Deze les gaat over het functioneel ontwerp (FO). Als uitgangspunt neem je het PvE en als jouw functioneel ontwerp klaar is dan licht je die toe aan de opdrachtgever en vraag je om akkoord.*

*In deze les bespreken we wat er in het functioneel ontwerp moet komen te staan en hoe je wordt beoordeeld. Ten slotte gaan we oefenen met de cases die we al kennen vanuit de vorige les.*

## Examen-template

We beginnen gelijk met de deur-in-huis en gaan kijken naar de examen-template; wat moet er in het functioneel ontwerp komen te staan?

De onderstaande indeling is voor een groot deel gekopieerd uit de template die je krijgt op het examen.

Let op het FO heeft een voorblad waarop de titel en jouw naam staat. Op de tweede pagina staat een inhoudopgave en de pagina's hebben paginanummers. Het FO bestaat uit ongeveer 10 pagina's, maar bevat vaak ook veel plaatjes/schema's.

Verder geldt in zijn algemeen, zorg dat alles er netjes en verzorgd uitziet en dat je de zaken zo eenduidig mogelijk beschrijft, zorg ervoor dat het zo duidelijk mogelijk is en dat jouw woorden niet verkeerd begrepen kunnen worden.

Het FO is heel veel tekenen en schetsen. Bepaal voor jezelf welke tools jij wilt gaan gebruiken en dat je alles klaar hebt staan voor het examen.





## Inleiding

In dit functioneel ontwerp wordt een overzicht gegeven van de functionaliteiten waaraan de applicatie XYZ moet voldoen.

## Functionaliteiten

*(dit is een cruciaal criterium)*

Hier vermeld je alle functionaliteiten. Dit mag in 'use case formaat' (kan met Lucichart en/of Creatly) en als je een heel duidelijk PvE hebt gemaakt dan kan je daar veel uit kopiëren. Beschrijf de eisen duidelijk en geeft prioriteiten aan. Zet alles duidelijk en netjes op papier gebruik bijvoorbeeld tabellen of schema's.

Voorbeelden:

### **Fout** (waarom?)

- de applicatie moet er mooi uit zien en mag geen fouten bevatten
- de applicatie moet veilig zijn
- als admin kan ik de applicatie beheren
- buttons zullen met Bootstrap 4.0 worden vormgegeven

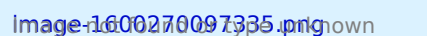
### **Goed** (waarom?)

- als ingelogde gebruiker kan ik zien van welke cursus nog plaatsen beschikbaar zijn.
- als ingelogd gebruiker kan ik mij inschrijven voor een cursus als er ten miste nog voldoende plaatsen zijn
- als ik als admin ben aangemeld dan kan ik alle cursus gegevens beheren, ik kan alleen geen cursussen weggoien waar inschrijvingen aan gekoppeld zijn.
- als ik als admin ben aangemeld kan ik de inschrijvingen op een cursus beheren. Ik kan dus inschrijvingen toevoegen en verwijderen.

## Basisschermlay-out

Hoe ziet de website eruit; kleurgebruik, lettertype, en standaardcomponenten. Is er een voorbeeld website, zijn er logo's of banners die moeten worden gebruikt?

Gebruik een voorbeeld van hoe de website er uit ziet. Maar een proto-type, dat kan met een wireframe of met een schermvoorbeeld. Je krijgt hierdoor een algemeen beeld van hoe de website er uit komt te zien:



Als tool wordt <https://balsamiq.cloud> aangeraden. Op verzoek kan je via teams een licentie key aanvragen.

## Tool

Een ander voorbeeld van een wireframe vind je op:

<https://www.roc.ovh/books/examentraining-b1->

# Beoordeling Examen

In totaal moet ja samen met het TO minimaal 10 (van de 19) punten behalen en er zijn geen cruciale criteria.

Het officiële [beoordelingsformulier](#) is bij deze pagina toegevoegd. Let op deze bevat de beoordeling voor het FO en het TO samen.

	<h2>Functionaliteiten</h2> <p>(0 tot 3 punten)</p> <p>Cruciaal criterium! Minimale eis: "bijna alle eisen en wensen uit het PvE zijn opgenomen en deze zijn begrijpelijk en eenduidig beschreven".</p> <h2>Toegepaste schematechnieken</h2> <p>(0 tot 3 punten)</p> <p>De juiste schematechnieken zijn gekozen en verduidelijken de functionaliteit.</p> <h2>User Interface</h2> <p>(0 tot 3 punten)</p> <p>Het is duidelijk hoe het systeem eruit gaat zien en de relaties tussen de schermen is ook duidelijk. Hiervoor is een sitemap of navigatiestructuur-schema opgenomen.</p> <h2>Communiceren FO</h2> <p><b>Cruciaal</b></p> <p>(0 of 1 punt)</p> <p>Het FO wordt duidelijk toegelicht en er wordt om <b>goedkeuring</b> gevraagd.</p>
--	--

Je moet minimaal 5 (van de 11) punten behalen en er zijn geen cruciale criteria.

Over het algemeen moet je hier aan voldoen (richtlijn):

1. Functionaliteiten *duidelijke* beschrijven.
2. use-case diagram.
3. navigatiesstructuur schema
4. Eén of twee uitvoerontwerpen (schermen)
5. Een aantal invoerschermen (forms)

# Aanvullende bronnen

Boek: Applicatie- en mediaontwikkelaar door Hans van Rheenen, hoofdstuk 5.

## Opdracht

Maak een FO van de case waarvan we ook een PvE en Pojectplan hebben gemaakt.

--

# 5 Technisch Ontwerp

## Wat gaan we leren?

*De derde opdracht van kerntaak 1 is het maken van functioneel en technisch ontwerp. Deze les gaat over het functioneel ontwerp (FO). Als uitgangspunt neem je het PvE en als jouw functioneel ontwerp klaar is dan licht je die toe aan de opdrachtgever en vraag je om akkoord.*

*In deze les bespreken we wat er in het functioneel ontwerp moet komen te staan en hoe je wordt beoordeeld. Tenslotte gaan we oefenen met de cases die we al kennen vanuit de vorige les.*

Voor deze les gebruiken we nog steeds de case over de [Zeilschool](#).

## Examen-template

We beginnen gelijk met de deur-in-huis en gaan kijken naar de examen-template; wat moet er in het technisch ontwerp komen te staan?

De onderstaande indeling is voor een groot deel gekopieerd uit de template die je krijgt op het examen.

Let op het TO heeft een voorblad waarop de titel en jouw naam staat. Op de tweede pagina staat een inhoudopgave en de pagina's hebben paginanummers. Het TO bestaat uit ongeveer 10 pagina's, maar bevat vaak ook veel plaatjes/schema's.

Verder geldt in zijn algemeen, zorg dat alles er netjes en verzorgd uit ziet en dat je de zaken zo eenduidig mogelijk beschrijft, zorg ervoor dat het zo duidelijk mogelijk is en dat jouw woorden niet verkeerd begrepen kunnen worden.

## Tools

Als teken tools, kun je dezelfde tools gebruiken als die ook bij het maken van het FO zijn gebruikt.



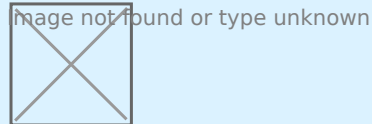


# Technische Specificaties

## Functionaliteiten

Laat met een schema zien hoe de flow van de (belangrijkste) functies is.

Bijvoorbeeld (de meeste TO's bevatten een hele serie van deze diagrammen, elk diagram komt vaak overeen met een Use Case):




## Gebruikersschermen

Laat een aantal invoerschermen zien; ook hier geldt dat de belangrijkste schermen moet tekenen en moet laten zien.

Voorbeelden:





Vergeet niet de schermen te beschrijven. Maak een **technische** beschrijving, dus benoem de data types en de constraints. Dus bijvoorbeeld, bij het hierboven getoonde scherm *gespreksaanvraag*:

- De voor en achternaam is een string van minimaal 5 - en maximaal 40 karakters.
- Gesprekstype is een drop down en je kiest een gesprekstype dat is gekoppeld aan het huidige examen.
- Rolspeler laat alle namen van alle rolspelers zien die de status actief hebben.  
De status is een drop down en heeft drie waarden 1=wachten, 2=loopt en 3=klaar.

Net als bij het FO kan je <https://app.lucidchart.com> of <https://balsamiq.cloud> gebruiken.

## Uitvoerontwerp

Hier geldt hetzelfde, maar dan voor de uitvoer.



Laat minimaal de belangrijkste uitvoerontwerpen zien. Uitvoer is het resultaat van het programma. Bijvoorbeeld een overzicht van alle cursusreserveringen of een overzicht van de klassenindeling. Geef dit in een prototype weer.

Ook hier geldt dat je het scherm technisch dient te beschrijven. In het scherm hierboven beschrijf wat de knopjes (rechtsbovenin) doen, beschrijf wat de eerste (lege) regel doet, etc.

# Beoordeling Examen

In totaal moet minimaal 5 (van de 11) punten behalen en er zijn geen cruciale criteria.

Het officiële [beoordelingsformulier](#) is bij deze pagina toegevoegd. Let op deze bevat de beoordeling voor het FO en het TO samen.

	<h2>Technische specificateis</h2> <p>(0 tot 3 punten)</p> <p>De technische specificaties zijn duidelijk omschreven en/of met de juiste schematechnieken verduidelijkt.</p> <p>De specificaties zijn goed als de proces flows (use cases) zijn beschreven, er de belangrijkste in- en uitvoer schermen zijn geschetst en of zijn beschreven.</p> <h2>Relationeel datamodel</h2> <p><b>Cruciaal</b></p> <p>(0 tot 3 punten)</p> <p>Alle entiteiten moeten zijn beschreven. Daarnaast moet je de meeste attributen en relaties goed hebben weergegeven.</p> <p>Zorg dus voor een goed ERD en benoem alle entiteiten (desnoods één te veel).</p> <h2>Communiceren TO</h2> <p>(0 of 1 punt)</p> <p>Het FO wordt duidelijk toegelicht en er wordt om <b>goedkeuring</b> gevraagd.</p>
--	---

Samen met het FO moet je minimaal 10 (van de 19) punten behalen en je moet een ERD met alle entiteiten hebben opgenomen in je TO.

## Aanvullende bronnen

Applicatie- en Mediaontwikkelaar, Hans van Rheenen. Hoofdstuk 6

## Opdracht

Maak een TO van de case waarvan we ook een PvE en Pojectplan hebben gemaakt.

--



# 5.0 ERD in het kort

In deze les gaan we normaliseren volgens de ERFDK methode. In deze methode wordt stapje voor stapje uitgelegd hoe je uit een verhaal een ERD, een datamodel moet maken,

De **ERFDK** - methode -> **Entiteiten** - **Relaties** - **Filter** - **Diagram** - **Attributen** - **Keys**

## Stap 1 - Entiteiten

Bepaal alle entiteiten.

Een entiteit is een groep van dingen, personen, gebeurtenissen. Bijvoorbeeld: leerlingen, klaslokalen, adres, belastingaanslag, verkeersboete, rekening, kassabon, docenten, developers, medewerkers, etc.

Van entiteiten leg je gegevens vast, dit heten attributen. Zo kan je van een medewerker, zijn naam, zijn adres, zijn bankrekeningnummer, etc vastleggen.

Voorbeeld:

Een bedrijf wil een planning maken voor de telefonische hulpdienst die zij bieden aan hun klanten. Zij willen vastleggen welke medewerker wanneer moet meedraaien met de telefonisch hulpdienst.

De entiteiten zijn: bedrijf, hulpdienst, klanten en medewerker. **MAAR** let op! Je wilt alleen gegevens vastleggen van medewerkers en de planning (van de hulpdienst). Je legt namelijk vast *welke* medewerker *wanneer* moet meedraaien. *Welke medewerker* bepaal je door de entiteit *medewerker* te maken en *wanneer* leg je vast in de entiteit *planning*.

## Stap 2 - Relaties

Leg de directe relaties.

Gebruik daarvoor een tabel. Stel je hebt vier entiteiten, entiteit 1, entiteit 2, entiteit 3 en entiteit 4.

Bij 4 entiteiten heb je 6 mogelijke relaties (3+2+1), bij 5 10 (4+3+2+1) en bij 6 15 (5+4+3+2+1), etc.

Stel van alle combinaties de relatie vast en kies daarbij uit:

- 1:1

- 1:N
- N:1
- N:M
- geen directe relatie

Entiteit 1	Entiteit 2	Relatie (voorbeelden)
entiteit 1	entiteit 2	N:M
entiteit 1	entiteit 3	1:1
entiteit 1	entiteit 4	1:N
entiteit 2	entiteit 3	geen (directe) relatie
entiteit 2	entiteit 4	1:N
entiteit 3	entiteit 4	geen (directe) relatie

## Stap 3- Filter

Haal alles eruit wat niet van belang is.

Entiteit 1	Entiteit 2	Relatie (voorbeelden)
entiteit 1	entiteit 2	N:M
entiteit 1	entiteit 3	1:1
entiteit 1	entiteit 4	1:N
entiteit 2	entiteit 3	geen (directe) relatie
entiteit 2	entiteit 4	1:N
entiteit 3	entiteit 4	geen (directe) relatie

Een 1:1 relatie kan soms worden samengevoegd (dit kan later ook nog).

## Koppeltabel

Een N:M relatie moet in een RDBMS altijd met een koppeltabel worden vormgegeven. We hebben dus een extra entiteit, de koppeltabel. We vinden daarvoor nog wel een toepasselijke naam.

## Stap 4 - Diagram

We maken een diagram (ERD) - eerst nog zonder keys en attributen, maar wel met de relaties.

Vervang N:M relaties met 1:N relaties via de koppeltabel. Uitleg zie: [https://youtu.be/4bwu4s\\_wjUs](https://youtu.be/4bwu4s_wjUs)

[https://www.youtube.com/embed/4bwu4s\\_wjUs](https://www.youtube.com/embed/4bwu4s_wjUs)

## Stap 5 - Attributen

Zet alle items die je wilt vastleggen als attributen in de juiste entiteit.

Attributen zijn de eigenschappen van een entiteit. Als iets een nummer is, bijvoorbeeld, eindcijfer, prijs, temperatuur, datum, huisnummer, .... dan is het een attribuut (en dus geen entiteit).

## Stap 6 - Keys

### Primary Key

Bepaal de primary keys (PK's) dat is een uniek gegeven per entiteit en mag nooit wijzigen.

Tip: voeg een aparte key (ID) toe aan elke entiteit, dus TeamID en TainingID, SpelerID, etc.

### Foreign Key

Bepaal de foreign keys (FK's) dat zijn de verbindingen naar de ander tabellen. Zodra er sprake is van een 1:N relatie dan heeft de N-kant de pprimary key van de 1-kant en dat heet dan de foreign key. Zo heeft speler een TeadID als foreign key.

[image1600268897589.png](#)

Zie jij waar Greg woont? Kijk eerst goed naar het plaatsje en bepaal waar Greg woont. Lees dan pas verder.

Gefeliciteerd om te bepalen waar Greg woont heb je gebruik gemaakt van de foreign key en de primary key.

Je zoekt namelijk eerst de FK op in de kolom stad in de tabel personen. Deze FK verwijst dan naar de kolom ID in de tabel plaatsnamen. In de tabel plaatsnamen is deze kolom de PK.

## Extra - Stap 7 check

- Leg je attributen meer dan 1x vast (los van de foreign key's) dan is er iets niet goed, dan moet je meer entiteiten maken?  
In zijn algemeenheid; als je een gegeven/attribuut wilt aanpassen, hoef je dan maar op één plaats te doen?

- Heb je 1:1 relaties, dan kun je die waarschijnlijk samenvoegen?
- Is alles een 1:N relatie?
- Heeft elke entiteit tenminste een relatie?  
Uitzondering kan zo kan login/user soms los van alles staan en helemaal geen relatie hebben met de andere entiteiten.

--

# 5.1 ERD - oefeningen

Hieronder staan 4 opgaven om te oefenen met het opstellen van een databaseontwerp. Voor KT-1 is het nodig dat je een ERD kunt opstellen. Zonder goed ERD kun je het examen niet halen!

## Bronnen

Voor de theorie wordt verwezen naar de volgende bronnen:

- Theorie: hoofdstuk 6.5 Relationeel Datamodel, boek Applicatie en Mediaontwikkelaar
- Goed voorbeeld van hoe je komt tot een genormaliseerd database-ontwerp (hoe bepaal je de entiteiten): [https://www.youtube.com/watch?v=mFU\\_s\\_9kY1U](https://www.youtube.com/watch?v=mFU_s_9kY1U)
- Nog een voorbeeld van normaliseren: <https://www.youtube.com/watch?v=fS-8Hw44wd0>
- Voorbeeld gebruik Lucichart: <https://www.youtube.com/watch?v=eghffa6ykF0>

---

## Opgave 1

Afdeling personeelsadministratie van een ziekenhuis registreert in een dossier per medewerker welke training hij/zij heeft gevolgd. Tijdens het jaarlijkse functioneringsgesprek wordt deze informatie gebruikt om te bespreken of de kennis van de medewerker op peil is en welke trainingen deze medewerker het komende jaar moet volgen. Welke trainingen verplicht zijn, is afhankelijk van de functie van de medewerker. Er worden verschillende soorten trainingen aangeboden voor verschillende functies. De volgende tabellen worden hiervoor gebruikt:

MEDEWERKER (MedNummer, MedNaam, MedGeboortedatum, FunctieCode, DatumInDienst)

TRAININGSOORT (TrCode, TrainingOmschrijving, Doelgroep, StudieBelastingUren)

TRAINING (TrainingID, TrCode, docent, startdatum, AantalWeken, Aanvang, Eindtijd, Locatie)

DEELNAME (MedNummer, TrainingID, startdatum, resultaat)

FUNCTIE (FunctieCode, FunctieOmschrijving, MaxSalaris, StartSalaris)

1. Onderstreep de Primary Keys
2. Zet een sterretje bij de Foreign Keys\*

3. Teken een ERD met tabelnamen (geen velden) en de relaties tussen de tabellen. Toon in de relaties de één en véél kant met kraaienpootjes.
- 

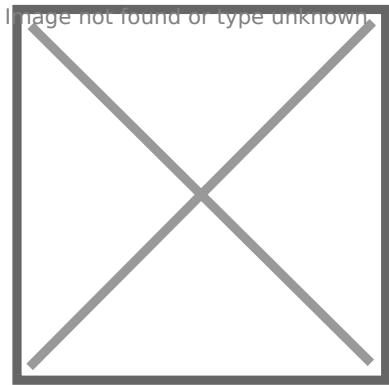
## Opgave 2

De secretaris van een voetbalclub wil aan het begin van het seizoen de teamindeling in het clubblad publiceren volgens het voorbeeld hieronder.

Ook wil hij een overzicht publiceren met per team de trainingsavond en -tijden, de trainer en de coach.

Voor de verzending van het clubblad heeft hij adresstickers van alle spelers nodig.

Een speler kan slechts in één team meespelen.



Ontwerp het datamodel. Je kunt gebruik maken van [lucidchart](https://lucidchart.com) voor het opstellen van jouw ERD.

1. Welke **entiteiten** (personen, dingen gebeurtenissen) kun je onderscheiden? Benoem alle entiteiten.
  2. Noteer per entiteit (=tabel) de **attributen** (=velden) die nodig zijn om de informatie zoals hierboven beschreven staat, te kunnen leveren.
  3. Bedenk per tabel wat je als **primary key** (pk, sleutelveld) kiest en in welke tabellen je een **foreign key** (FK) opneemt.
  4. Teken het uiteindelijke ERD.
- 

## Opgave 3

Een database bestaat uit tabellen die aan een aantal voorwaarden moeten voldoen. Hieronder staan een aantal tabelontwerpen. (Deze tabellen hebben niets met elkaar te maken.)

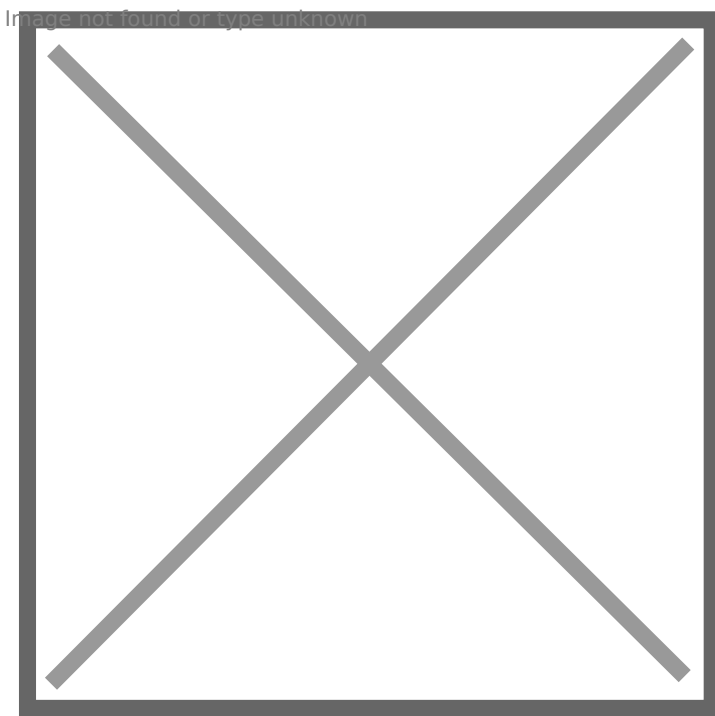
1. Wat is er fout aan de volgende tabellen?
2. Onderstreep in iedere tabel het sleutelveld.

- LEERLING (voornaam, tussenv, achternaam, leerlingnummer, leeftijd, adres, postcode, woonplaats)
  - BESTELLING(bestelnummer, besteldatum, klantnr, product1, aantal1, product2, aantal2, product3, aantal3)
  - CONTRIBUTIE\_BETALING(lidno, achternaam, bankrekening, datum\_betaling, bedrag)
  - PRODUCT(productnummer, prijs, voorraad, leveranciersnaam, leverancierstelefoonnummer)
- 

## Opgave 4

Bij het bedrijf AIKIA worden meubels verkocht. Een meubel bestaat uit een aantal standaard componenten die artikelen worden genoemd. Als je alle artikelen van een product optelt dan kom je op de prijs van een product.

Zo maakt de afdeling calculatie van AIKIA het volgende periodieke overzicht:



Verder worden offertes gemaakt. Een offerte is gericht aan een klant. Van deze klant worden de Naw-gegevens vastgelegd. Een offerte bestaat uit één of meerdere producten en een klant kan meerdere offertes krijgen. Een offerte heeft een bepaalde geldigheidsduur en op elke offerte kan een kortingspercentage zijn toegekend.

Maak een ERD van de database waarmee je de hierboven beschreven gegevens kan vastleggen.

1. Bepaal **alle** entiteiten (voor het examen geldt; beter een entiteit te veel dan te weinig).

2. Bepaal per entiteit de attributen
  3. Bepaal de primary- en foreign keys van elke entiteit
  4. Bepaal de relaties tussen de entiteiten
  5. maak een diagram in bijvoorbeeld [app.lucidchart.com](https://app.lucidchart.com) van jouw ERD.
- 

## Opgave 5

De planner van het schoolrooster wil een planning maken. In de planning moet komen te staan welke klas wanneer welk vak heeft. De volgende functionaliteiten moet het datamodel ondersteunen.

- als planner wil je een klas inplannen op een bepaalde tijd van de week voor een bepaald vak.
- als planner wil je de mogelijkheid hebben om elke week het rooster te kunnen aanpassen
- als leerling kan ik zien bij wie ik in de klas zit
- als leerling kan de planning zien, wanneer heb ik welk vak
- de planner wil van elk vak de naam, de afkorting en naam van de vakgroepcoördinator vastleggen
- elk vak heeft precies één vakgroepcoördinator
- van de leerling wil ik de naam (voor- en achternaam) en geboortedatum kunnen vastleggen
- van een klas wil ik de naam, afkorting en startjaar kunnen vastleggen

Maak een ERD dat bovenstaande functionaliteiten ondersteund. Dus bepaal de entiteiten, de relaties, alle attributen en de PK's en FK's. En maak hiervan een ERD diagram.

---

## Opgave 6

Jouw collega was bezig met een datamodel, maar hij is plotseling ziek geworden. Er ontbreken nog een paar keys en relaties.

Maak het ERD af, zet overal de juiste key bij (PK of FK) en maak de juiste relaties.

[image1600976911135.png](#)

---

## Opgave 7

Bekijk het onderstaande datamodel. Wat is onjuist? Tip: stel je bent een klant en je wilt vier artikelen bestellen; hoe leg je dat vast in de database?

Verbeter het datamodel.

[image-1600976695560.png](#)

---

# 6 Configuratieoverzicht

## Wat gaan we leren?

*Voor de laatste opdracht van Kerntaak-1 moeten we de ontwikkelomgeving installeren. Dit is aan de ene kant een technisch verhaal waarbij je moet zorgen dat je laptop helemaal goed is ingericht om te kunnen ontwikkelen. Is de IDE goed geïnstalleerd, draait je framework, is alles ge-update (ook Windows!).*

*Naast de technisch kant waarbij jouw laptop moet inrichten, moet je ook een beschrijving opleveren waarin je opsomt wat er allemaal in de ontwikkelomgeving moet worden geïnstalleerd.*

## Examen-template

	<h3>Te gebruiken hardware</h3> <p>Geef een overzicht van welke hardware je nodig hebt, specificeer de minimale eisen ten aanzien van scherm, harddisk, geheugen en CPU.</p> <h3>Software</h3> <p>Maak een lijst met alle software componenten met de versienummers die je nodig hebt voor de ontwikkeling. Denk aan:</p> <ul style="list-style-type: none"><li>- Editor</li><li>- PHP engine (XAMMP)</li><li>- PHP versie</li><li>- SQL versie</li><li>- Framework en versie</li><li>- etc.</li></ul>
--	---

## Beoordeling Examen

In totaal moet minimaal 5 (van de 11) punten behalen en er zijn geen cruciale criteria.

Het officiële [beoordelingsformulier](#) is bij deze pagina toegevoegd. Let op deze bevat de beoordeling voor het FO en het TO samen.

## Benodigde onderdelen

(0 tot 3 punten)

De benodigde onderdelen voor de ontwikkelomgeving is volledig beschreven.

## Installeren en configureren

(0 of 1 punten)

De ontwikkelomgeving is juist geïnstalleerd en geconfigureerd.

## Testen

(cruciaal onderdeel)

(0 of 1 punten)

Alles is getest en werkt.

## Documenteren

(0 of 1 punt)

Documentatie is duidelijk en volledig.

Je moet minimaal 3 (van de 6) punten behalen en de omgeving moet zijn getest. Met andere woorden alles moet werken!

# Opdracht

Maak het hard- en software overzicht aan de hand van de template.

--

# Tip Top 10

1. Eerste gesprek, **bereid je voor**, neem je tijd. Bedenk je vragen van tevoren goed. Je hebt maar één kans.
2. Gesprekken. Eindig alle gesprekken waarbij je document presenteert met de vraag of de klant of projectleider **akkoord** gaat en of je verder kan.
3. Project plan. Vergeet niet de **ontwikkelmethode** te benoemen.
4. ERD. Beter een **entiteit** te veel dan te weinig, bij twijfel dus beter één extra.
5. ERD, keys: **PK's (id)** en **FK 'horen' bij de harkjes** en verwijzen naar de andere kant van de relatie.
6. **Check je Laptop** een week van tevoren; windows update, ontwikkel omgeving. No games for one week. En zondag voor het examen nog een laatste check.
7. Klopt de inhoudsopgave nog in je document (right mouse update table)?
8. Formuliertje voor FO, zo gemaakt: <http://www.phpform.org/>

9. Niet vergeten, **ID, laptop en lader, eten/drinken, pen, oortjes**, laptop opgeladen meenemen!
10. Zondag op tijd naar bed!