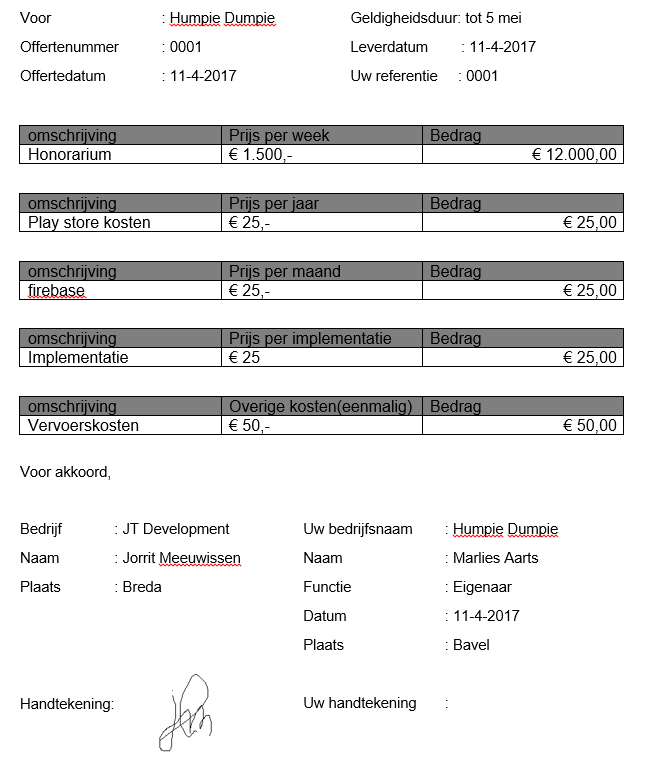
# Offerte

Om ervoor te zorgen dat de klant een goed beeld kreeg van wat ik deed en wat het zou gaan kosten deed ik een offerte maken. Dit zorgde ervoor dat de klant goed wist wat haat situatie op dat moment was en naar wat voor situatie de klant naartoe wilde. Dit zorgde er ook goed voor dat de klant een goed beeld kreeg van wat het ging kosten voor wat ze precies wilde. Ook deed ik erbij schrijven wat de klant nou precies wilde en wat mijn visie op het project was. Want zo kon de klant zie of dat ik met het project eens was of juist helemaal niet. En dat geeft de klant een beetje zekerheid. Want in de offerte heb ik een paar kopjes gebruikt om alles juist goed uit te leggen aan de klant. En die kopjes zijn:

* Inleiding;
* Actuele klantsituatie;
* Klantvraag;
* Mijn visie op uw situatie;
* De gewenste klantsituatie;
* Het aanbod;
* Uitvoerders;
* Planning;
* Prijs;
* Offerte;
* Afsluiting



Bij elk puntje hierboven staat goed omschreven wat ik bedoelde met het onderwerp om het zo duidelijk mogelijk te maken voor de klant.

Zoals bij de offerte zelf liet ik zien wat de prijs van alles is en of het 1 keer betaald hoeft te worden of dat het per maand betaald moet worden. En dat maakt het overzichtelijk en duidelijk voor de klant.

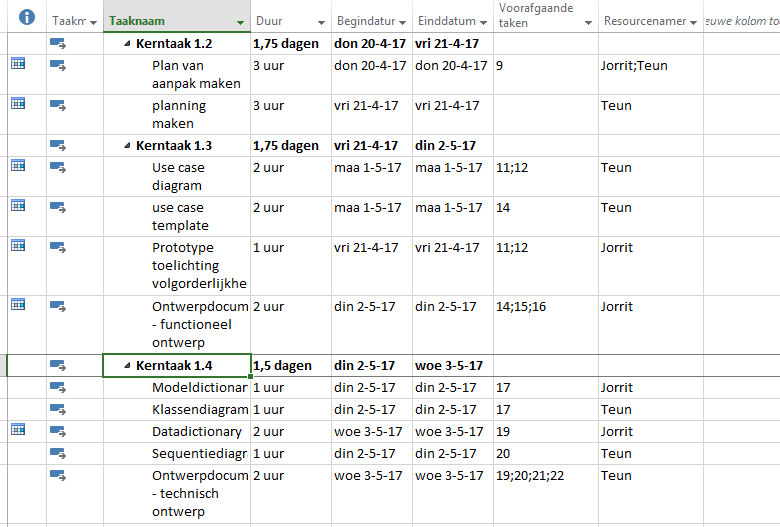
Wat ik allemaal bij de rest van de puntjes heb geschreven en uitgelegd staat in het mapje 1.3 offerte.

# Globale Planning

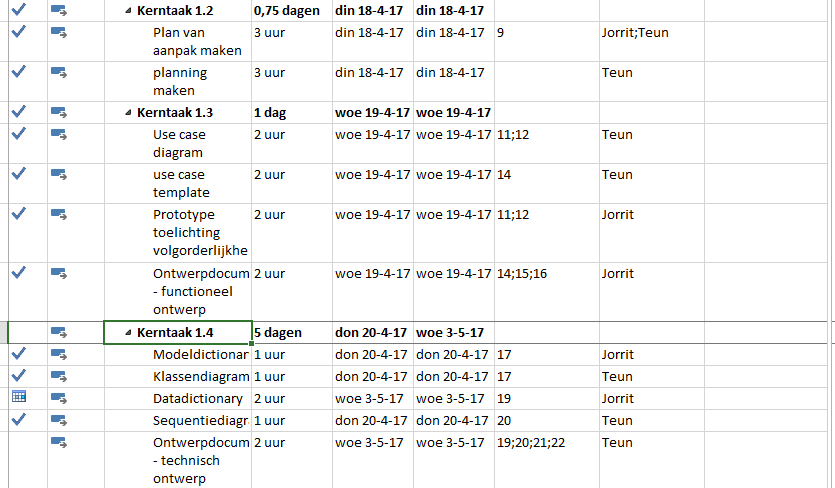
Ik heb een planning gemaakt over het hele project. Dit is omdat ik veel moest doen voor het project en een paar deadlines had bij het project. En door de planning kon ik een goed beeld krijgen van wat ik allemaal moest gaan doen. Dit zorgde er ook voor dat ik goed kon zien hoelang iets duurde en of ik het project wel zou halen. Want ik werkte 4 uur op een dag en 5 dagen in de week. Dit deed ik ook in mijn planning opnemen zodat ik goed kon zien wat ik op een dag zou doen en of ik het afkreeg.

Maar een planning is iets wat je van te voren doet dus ik moest goed schatten hoeveel tijd iets zou gaan kosten. Dit was best moeilijk omdat sommige dingen meer tijd koste dan dat ze werkelijk kosten. Dit komt omdat ik over sommige dingen gewoonweg wat langer over deed. Maar er was af en toe ook een meevaller omdat ik eerder klaar was met een opdracht dan dat ik had geplant en die tijd kon ik dan weer gebruiken voor iets wat, wat langer duurde.

Naarmate de eerste week vorderde deed ik de planning aanpassen omdat het niet meer goed was op wat ik werkelijk deed. Dus deed ik mijn planning aanpassen zodat het wel weer goed en kon ik weer goed zien wat ik allemaal moest gaan doen en hoelang dat zou gaan duren.



Dit is een voorbeeld van de planning die ik in het begin had gemaakt. Maar na een week zag hij er zog een beetje anders uit omdat sommige dingen toch al eerder klaar waren dan gepland.



En dit is een voorbeeld van de planning nadat de eerste week voorbij was. Dat is dan toch wel een beetje anders dan de eerste planning. Maar dat kan gebeuren omdat het toch van tevoren inschatten is en dat ik altijd moeilijk.

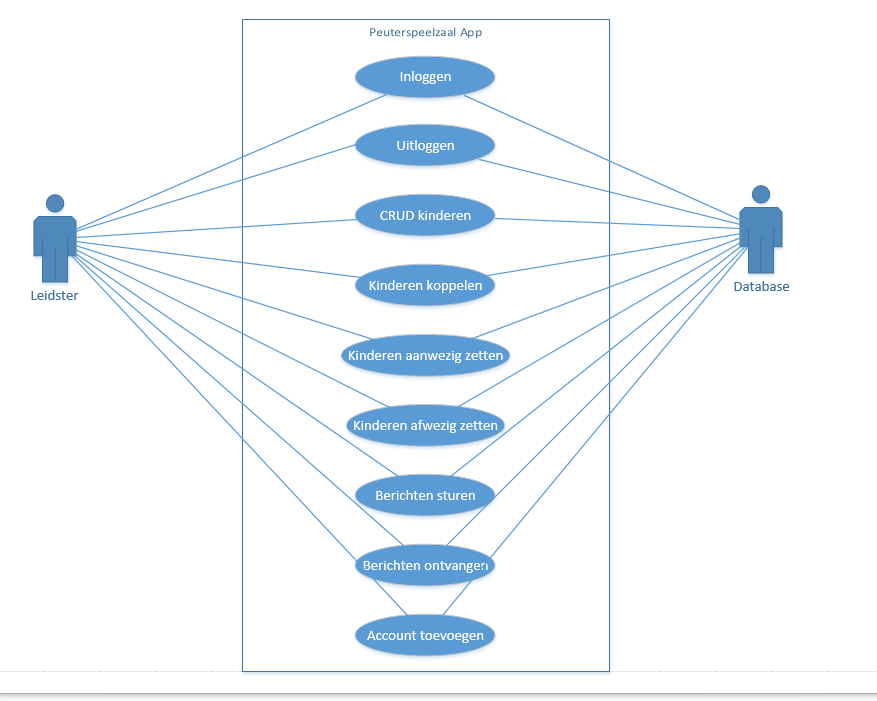
De gehele planning zijn te zien in het mapje 1.6 Planning.

# Ontwerpdocument – Functioneel ontwerp

Het functioneel ontwerp bestaat uit use-case diagrammen, use-case templates en een toelichting van de volgordelijkheid van de prototypes. In het functioneel ontwerp staat beschreven wat er met die drie onderwerpen bedoeld word en waar ze precies staan. Want het functioneel ontwerp is meer een kleine uitleg over de onderwerpen en dan waar je de documenten echt kan vinden. Dit komt doordat het functioneel document meer een verzamel document is en niet de opdrachten zelf. Het hele functionele ontwerp kun je vinden in het mapje 1.7 functioneel ontwerp.

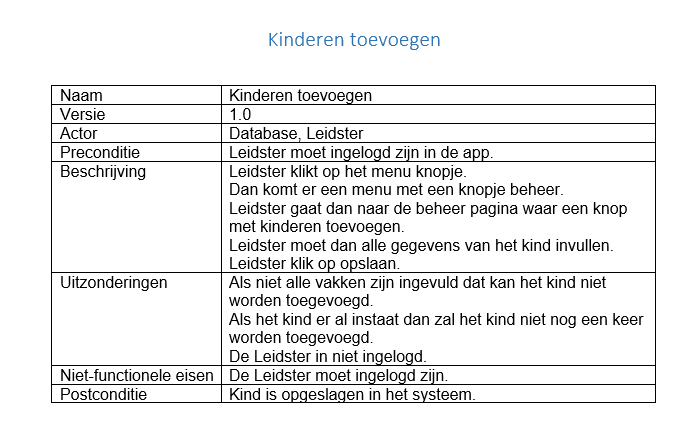
# Use case diagram en Use Case template

Ik heb ervoor gezorgd met een use case diagram dat iedereen kan zien wat een gebruiker voor functies heeft. Dit heb ik gedaan omdat het dan overzichtelijker word voor iedereen van elk van de gebruikers kan doen en wat een gebruiker niet kan of mag doen. Ik heb het voor elke gebruiker apart gedaan omdat het dan overzichtelijker word voor iedereen. Dus ik heb een aparte use case diagram voor de leidster en voor de ouder omdat dat de enige gebruikers zijn die aan het systeem deel nemen. Dus bijvoorbeeld bij de leidster staat de leidster aan de linkerkant omdat het daar begint en dan alle functies die de leidster kan doen in het midden en dan aan de rechterkant staat de database omdat het systeem daarmee in contact komt met de functie.

Hieronder staat een voorbeeld van een use case diagram die ik heb gemaakt. En daarin kun je zien wat de leidster allemaal kan doen. De gehele use case diagram is te lezen in het mapje 1.8 use-case diagrammen en use-case templates. Hier is tevens ook de use case van de ouder te zien.

Bij een use case diagram hoort ook een use case template omdat je ook moet uitloggen wat elke functie doet in het systeem en hoe de gebruiker door het systeem moet gaan om van de functie gebruik te maken. Want de use case is alleen om te laten zien wat de gebruiker kan en de use case template is echt om uit te leggen wat de functie nou precies inhoud.

Dus in de use case template heb ik omschreven welke actoren er allemaal bij de functie in meewerken daarna moest ik zeggen wat de gebruiker eerst moest doen of waar de gebruiker eerst moest zijn. Daarna heb ik een uitleg gegeven van hoe de gebruiker met de functie tewerk gaat. Daarna heb ik uitgelegd van waarom als de gebruiker de functie doorloopt en hij doet het niet wat de gebruiker dan fout heeft gedaan. Daarna staat er nog wat de gebruiker moet hebben voordat hij aan de functie begint. En ten slotte waar de gebruiker komt of wat de gebruiker heeft gedaan voor als alles goed is gegaan.



Dit is een voorbeeld van de leidster use case template. Hier zie je dus alles van wat ik hierboven heb uitgelegd. Dit is heel erg handig omdat dan iedereen goed weet wat er aan de gang gaat met de functie en hoe de functie precies werkt. De gehele use case template is te vinden in het mapje 1.8 use-case diagrammen en use-case templates en hier is ook de use case template voor de ouder te vinden.

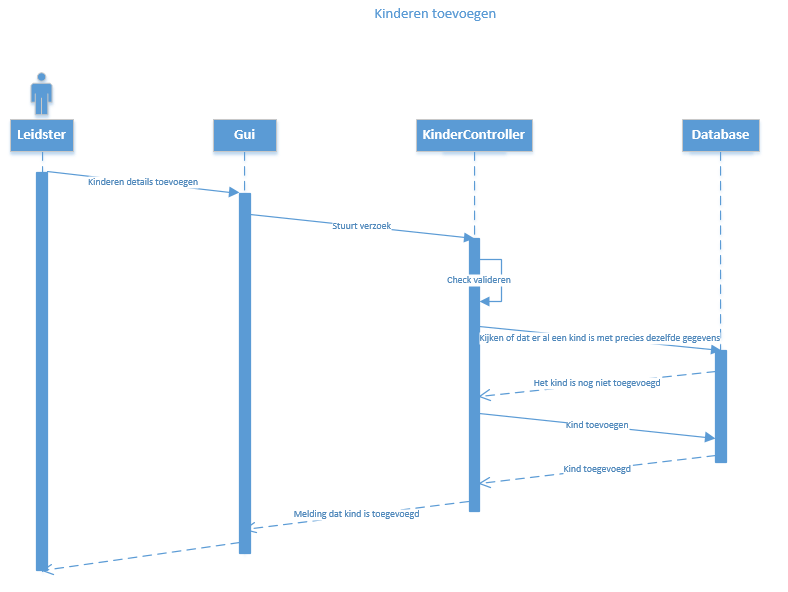
# Ontwerpdocument – technisch ontwerp

Het technisch ontwerp bestaat uit de modeldictionary, klassendiagram, datadictionary en een sequentiediagram. In het technisch ontwerp staat beschreven wat er met die vier onderwerpen bedoeld word en waar ze precies staan. Want het technisch ontwerp is meer een kleine uitleg over de onderwerpen en dan waar je de documenten echt kan vinden. Dit komt doordat het technisch document meer een verzamel document is en niet de opdrachten zelf. Het hele technisch ontwerp kun je vinden in het mapje 1.10 technisch ontwerp.

# Sequentiediagram

De sequentiediagram heb ik gemaakt om te laten zien hoe een use case in elkaar zit. Dit is omdat je bij de use case diagram ziet wie wat kan doen, dan bij de use case template leg je uit hoe het voor de gebruiker werkt en hoe het moet. En dan bij de sequentiediagram laat ik zien hoe het er in het systeem aan toegaat bij 1 functie. Dus 1 bolletje van de use case diagram is ook 1 vakje bij de use case template en dat is ook 1 sequentiediagram. En de sequentiediagram is echt voor de programmeur om te laten zien hoe het systeem moet lopen als die functie moet worden uitgevoerd.

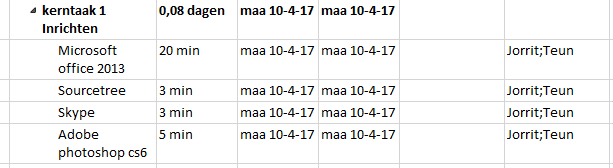
Dit is de sequentiediagram die ik heb gemaakt. Dit gaat over het toevoegen van kinderen. Hierin kun je zien welke dingen in het systeem allemaal te maken hebben met de functie. Zo is het makkelijker om te programmeren omdat je al goed kan zien hoe het er allemaal aan toe gaat.



De gehele sequentiediagram is te vinden in het mapje 1.14 sequentiediagram.

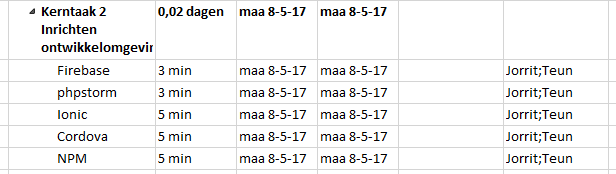
# Taakverdeling ontwikkelomgeving

Voordat ik iets aan het project kon doen had ik een aantal programma’s nodig om mee te werken. Dit komt omdat ik bijvoorbeeld een editor nodig had om een interview in te schrijven en die uit te printen om het aan de opdrachtgever te laten zien. Ik moest daarvoor eerst weten wat ik allemaal nodig had. En ook wanneer ik ze kon gaan installeren.



Bijvoorbeeld voordat ik ook maar iets kon gaan doen aan het project had ik deze programma’s nodig. Dus deze programma’s moesten op mijn computer staan zodat ik aan de slag kon met mijn project. Dit was tevens voor kerntaak 1. Voor kerntaak 2 had ik ook een aantal programma’s nodig. En hier heb ik ook alvast een taakverdeling over gemaakt omdat dat wel zo slim was.

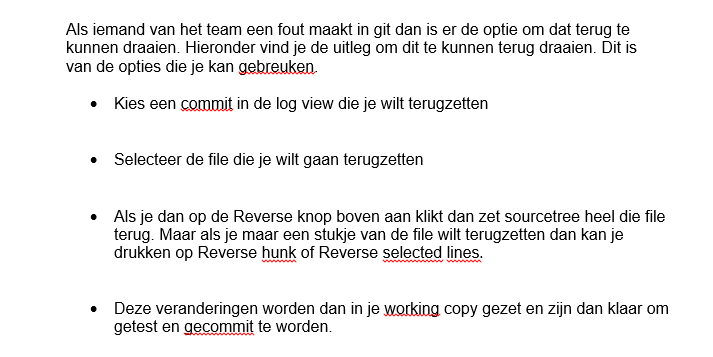
Dus voordat ik aan kerntaak 2 kan gaan werken moest ik eerst natuurlijk kerntaak 1 afhebben. Maar daarna moest ik nog de nodige programma’s installeren omdat ik die gewoon nodig had om te programmeren.



Hier kan je zien wat ik allemaal moest installeren voordat ik aan kerntaak 2 kan beginnen. Dit komt omdat ik echt al deze producten nodig had om dan verder met het project te gaan. Voor het gehele taakverdeling staat in het mapje 1.15 taakverdeling ontwikkelomgeving.

# Back-up procedure

De back-up procedure is om ervoor te zorgen dat iedereen weet van wanneer iets fout gaat met het project dat iedereen het project dan kan terughalen. Dit zorgt ervoor dat je niet afhankelijk bent van andere mensen als er iets fout gaat met het project. Want als je de procedure erbij pakt dan kan je gelijk goed lezen hoe je iets moet verhelpen en dat is wel zo fijn dat iedereen dat gelijk goed kan lezen en het dan ook uitvoeren.



Dit is een voorbeeld van een stappenplan om je project correct terug te halen als er iets fout gaat met het project. Want dit is vooral met github want daar gaan de meeste dingen mis als iemand iets wilt gaan comitten. En daarom is het belangrijk dat iedereen weet hoe hij iets kan terughalen.

Ook staat er in wat ik elke dag doe in het project. Want ik heb een paar vaste afspraken die ik doe aan het eind van elke dag. Dit heb ik ook beschreven in de back-up procedure. De gehele back-up procedure staat in het mapje 1.17 back-up procedure.

# Testomgeving

Ik heb een verslag gemaakt over hoe je de testomgeving moet inrichten zodat als iemand het project moet testen gelijk weet wat hij allemaal nodig heeft op zijn computer. Ook is het voor mij heel erg handig want zo weet ik goed hoe ik de testomgeving goed moet zetten. Ik heb eer ook een checklist in gemaakt zodat ik goed kan kijken of dat ik alles op mijn computer heb en ook of dat alles het goed doet. Dat is wel zo fijn van een checklist en een verslag over de testomgeving zodat iedereen kan zien hoe de testomgeving in elkaar zit en hoe ze het op hun eigen computer krijgen.



Hier is de checklist die ik heb gemaakt. Want als hier alles van werkt dan heb je de testomgeving goed werkend en dan kan je het project testen. Want deze checklist is laat gelijk zien wat je moet installeren en omdat in deze checklist staat wat het moet doen ben je gelijk ook de testomgeving aan het testen. En als alles niet gelukt is dan weet je ook gelijk dat je testomgeving niet goed is gemaakt. En moet je iets doen om ervoor te zorgen dat het wel lukt want anders kan je het hele project niet testen. Het gehele verslag over de testomgeving staat in het mapje 1.18 testomgeving. Hier staat ook in hoe je alles moet installeren.