

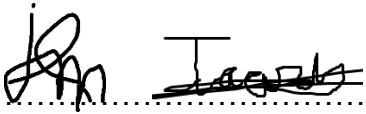
Projectverslag

PEUTERSPEELZAAL APP
JORRIT MEEUWISSEN, TEUN AARTS

Inhoudsopgave

Titelvervolgblad	2
Inleiding	3
Opdracht	3
Competenties KT1	4
KT 1-1 Stelt de vraag en/of de informatiebehoefte vast (J,K,M,R)	4
Uitgewerkt interview met opdrachtgever: interviewvragen en gespreksverslag	4
Hernieuwde opdracht	6
Offerte	7
Prototype (wireframes) gebaseerd op interview	8
KT 1-2 Maakt een plan van aanpak (E, J, Q)	8
Plan van aanpak	8
Globale planning	10
KT 1-3 Levert een bijdrage aan een functioneel ontwerp of game design document (E, I, J, K, U) ..	11
Ontwerpdocument – Functioneel ontwerp	11
Use Case diagram en Use Case Templates	12
Prototype toelichting volgorde van de ontwikkeling	14
KT 1-4 Maakt een technisch ontwerp (E, I, J, K, P)	14
Ontwerpdocument – Technisch ontwerp	14
Modeldictionary	15
Klassendiagram	16
Datadictionary	17
Sequentiediagram	17
KT 1-5 Richt de ontwikkelomgeving in (J, L, Q, S)	18
Taakverdeling ontwikkelomgeving	18
Materialenlijst: Hard- en software van de ontwikkelomgeving	19
Back-up procedure	20
Verslag over de installatie, configuratie en testen van de testomgeving	21

Titelvervolgblad

Titel verslag	: Projectverslag
Project naam	: Peuterspeelzaal app
Naam leerlingen	: Jorrit Meeuwissen, Teun Aarts
E-mail adressen	: D206218@edu.rocwb.nl , D195250@edu.rocwb.nl
Mobiele nummer leerlingen	: 06-39 11 06 62, +31641774751
Groep	: RIO4-MED3A
OV-nummers	: D206218, D195250
Schooljaar	: 2016-2017
Datum	: 3-5-2017
Handtekeningen groepsleden	: 
Naam opdrachtgever-bedrijf	: Humpie Dumpie
Adres opdrachtgever-bedrijf	: Pastoor Doenstraat 5
Postcode opdrachtgever-bedrijf	: 4854 CP Bavel
Naam opdrachtgever	: Marlies Aarts

Inleiding

Ik zit nu in het examenjaar van mijn opleiding. Om mijn opleiding goed af te ronden moet ik alle 4 de kerntaken voldoende afronden (Proeve van Bekwaamheid). Ik zit nu op school om een project uit te voeren om alle kerntaken te voltooien en mijn diploma te halen. Ik zit nu bij kerntaak 1 ontwerpen van een applicatie, (cross)media uiting of game. Hiervoor heb ik een heel verslag gemaakt waarop te zien is dat ik alle puntjes van de kerntaken heb gemaakt.

Bij het kopje van de opdracht kun je lezen wat de opdracht is die ik moet gaan doen voor het project.

Opdracht

De peuterspeelzaal wilt graag een app hebben die als digitale lijst werkt. In de app moeten ze kunnen zien welke kinderen er op die dag moeten zijn en welke zijn afgemeld voor die dag. Hier kunnen ze dan controleren of alle kinderen aanwezig zijn. Als er een kind niet is afgemeld en er ook niet is moet de leidster in staat kunnen zijn om naar de ouders van het kind een bericht te sturen.

In een week zijn er 4 dagdelen. Per groep zijn er 2 dagdelen en elk dagdeel heeft een aparte lijst nodig. In de presentielijst moet de leidster kunnen aantekenen hoe laat een kind binnenkomt en wanneer het kind weer vertrekt. In de lijst moet het aantal kinderen worden bijgehouden. Het maximale aantal kinderen per lijst is 16. Per kind moeten er ook verschillende gegevens beschikbaar zijn. Voorbeelden hiervan zijn allergie of dat het kind niet gefotografeerd mag worden. Bij de kinderen moet ook een foto beschikbaar zijn zodat de leidsters de kinderen makkelijker kunnen herkennen. De leidsters moeten berichten kunnen sturen naar de ouders en de ouders moeten dan ook berichten kunnen sturen naar de peuterspeelzaal. Hiermee kunnen de leidsters de ouders iets over het kind ontvangen of in geval van calamiteiten.

De app en de gegevensverzameling moeten zelf gemaakt worden want hiervoor is nog niks eerder gemaakt in de peuterspeelzaal. Voor elk kind moet er een account worden aangemaakt voor de ouders van het kind waarmee ze dan kunnen inloggen in de app om berichten te kunnen sturen en om op hoogte gehouden worden van ontwikkelingen. De leidsters willen 1 gedeeld account hebben waar ze alles mee kunnen doen.

Competenties KT1

KT 1-1 Stelt de vraag en/of de informatiebehoefte vast (J,K,M,R)

Uitgewerkt interview met opdrachtgever: interviewvragen en gespreksverslag

Om meer informatie te krijgen over de opdracht die ik heb gekregen heb ik een interview gepland.

Ik had een mailtje naar de opdrachtgever gestuurd met een uitnodiging om een interview te houden.



Meeuwissen, Jorrit (student) <D206218@edu.rocw.nl>



Allen beantwoorden | v

di 4-4, 11:17

Aarts, Teun (student); antoonaarts@hetnet.nl ↘

Vereist: Meeuwissen, Jorrit (student) <D206218@edu.rocw.nl>; Aarts, Teun (student); +1 meer ↘



Wanneer: ma 10-4-2017 14:30 - 15:30

Waar: Kloosterstraat 48A, 4854 CN Bavel

[Deze uitnodiging is bijgewerkt nadat dit bericht is verzonden. Open de update of open het item in de agenda.](#)

Beste mevrouw Aarts,

Wij willen graag een interview houden op **10-04-2017 om 14:30**.

In dit interview gaan wij vragen stellen over de peuterspeelzaal app die u wilt hebben.

Kunt u zo spoedig mogelijk doorgeven of deze datum u schikt?

Met vriendelijke groet,

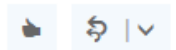
Teun Aarts

Jorrit Meeuwissen

Uitnodiging voor interview



Toon Aarts <antoonaarts@hetnet.nl>
di 4-4, 16:54



Beste Teun en Jorrit,
Maandag 10-04-2017 om 14.30 uur is goed.
Tot dan,
Marlies Aarts.
...



Aarts, Teun (student)

di 4-4, 11:18

Meeuwissen, Jorrit (student) <D206218@edu.rocw.nl> ✉



Allen beantwoorden | v



Wanneer: ma 10-4-2017 14:30 - 15:30

Waar: Kloosterstraat 48A, 4854 CN Bavel

✓ Aarts, Teun (student) heeft deze gebeurtenis geaccepteerd

Goedkeuring voor het interview

Nadat ik goedkeuring had gekregen voor het houden van het interview ging t ik het interview voorbereiden. Dit heb ik gedaan door te bedenken wat ik allemaal nog moet weten om de applicatie te maken.

Hierdoor heb ik verschillende vragen verzonnen die mij helpen bij het beter begrijpen van de opdracht.

Ik had de vragen getypt in een document die ik dan kon meenemen naar het interview.

De vragen zijn te vinden in de map *1.1 Stelt de vraag enof de informatiebehoefte vast* bij het document *Interview vragen.docx*.

Daarna ben ik het interview gaan houden op 10-04-2017 om 14:30.

In het interview heb ik alle vragen gesteld die ik had verzonnen en ben ik op elke vraag gaan doorvragen zodat ik het beste antwoord kon krijgen op de vraag die ik gesteld had.

De antwoorden die de opdrachtgever gaf heb ik allemaal opgeschreven zodat ik duidelijk inzicht heb wat ze had geantwoord.

De antwoorden op de vragen staan in de map *1.1 Stelt de vraag enof de informatiebehoefte vast* bij het document *Interview antwoorden.docx*.

Nadat ik het interview heb gehouden heb ik er een verslag van gemaakt en heb het laten onderteken door de opdrachtgever zodat die kon controleren of alle antwoorden die ik heb opgeschreven klopte. Het verslag is te vinden in de map *1.1 Stelt de vraag enof de informatiebehoefte vast* bij het document *gesprekverslag.docx*

Door het interview te houden is de opdracht duidelijker geworden en weet ik hoe ik alles moet gaan doen om aan de eisen van de opdrachtgever te kunnen voldoen.

Hernieuwde opdracht

Na het houden van het interview is de opdracht een stuk duidelijker geworden. Omdat de opdracht nou duidelijker is maak ik een hernieuwde opdracht waar de opdracht duidelijker is uitgelegd. Er is bijgezet wat de huidige situatie is en wat de gewenste situatie is. Daarna staat de opdracht hoe we dit moeten veranderen dat de huidige situatie in de gewenste situatie veranderd. Ook staan alle eisen van de opdracht in de hernieuwde opdracht.

De eisen zijn onderverdeeld in 3 verschillende categorieën namelijk:

- Grafische eisen
- Functionele eisen
- Technische eisen

De eisen heb ik ook nog in de MoSCoW methode gezet. Door dit overzicht is het duidelijk wat erin moet of wat minder hoge prioriteit heeft.

1	Naam functionaliteit	Must Have	Should Have	Could Have	Would Have
2	Grafische eisen				
3	De huisstijl moet rood, geel en blauw worden				
4	Het logo moet een kabouter worden				
5	Er mogen geen foto's bij de kinderen in de lijst komen				
6	In de lijst met alle kinderen van de groep moet bij een kind zijn allergieën				
7	komen te staan met een rood teken				
8					
9	Functionele eisen				
10	Leidsters en ouders moeten kunnen inloggen/uitloggen				
11	Leidsters moeten kinderen kunnen aanmaken, bekijken, wijzigen en verwijderen (CRUD)				
12	Leidsters moeten kinderen aan dagen kunnen koppelen				
13	Leidsters moeten de afmeldingen kunnen zien bij de kinderen				
14	Leidsters moeten kinderen aanwezig of afwezig kunnen zetten				
15	Leidsters moeten kinderen kunnen afmelden wanneer ze vertrekken				
16	Leidsters moeten kunnen chatten met ouders				
17	Leidsters moeten alle gegevens van een kind kunnen zien				
18	Ouders moeten een absentie kunnen invoeren				
19	Ouders moeten de leidsters een bericht kunnen sturen				
20	Accounts moeten 1 voor 1 worden toegevoegd				
21	De app moet in het Nederlands worden				
22	Leidsters en ouders moeten foto's via de chat kunnen sturen				
23	De foto's moeten uit de galerij geselecteerd kunnen worden				
24	Elke leidster moet een apart account hebben				
25	Er kunnen meerdere accounts per kind zijn				
26	Als er een kind verwijderd wordt moet het gearhiveerd worden				
27	De absentie moet onderverdeeld worden in de categorieën ziek, afwezig en vakantie				
28	De leidsters moeten de verstuurd berichten van de andere leidsters kunnen lezen				
29					
30	Technische eisen				
31	De app moet voor Android en IOS worden gemaakt				
32	De applicatie wordt in Ionic gemaakt				
33	De tijd moet in 24 uur notatie				

De eisen in de MoSCoW methode

De hernieuwde opdracht is te vinden in de map *1.2 Hernieuwde opdracht* bij het document *Hernieuwde opdracht.docx* en de uitgewerkte MoSCoW methode staat in dezelfde map bij het document *moscow.xls*.

De hernieuwde opdracht heb ik laten accorderen door de opdrachtgever zodat ik weet dat alle eisen zo kloppen en ik niks vergeten ben.

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 haar 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 zien 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.

Voor : Humple Dumpie Geldigheidsduur: tot 5 mei
Offertenummer : 0001 Leverdatum : 11-4-2017
Offertedatum : 11-4-2017 Uw referentie : 0001

omschrijving	Prijs per week	Bedrag
Honorarium	€ 1.500,-	€ 12.000,00

omschrijving	Prijs per jaar	Bedrag
Play store kosten	€ 25,-	€ 25,00

omschrijving	Prijs per maand	Bedrag
<u>firebase</u>	€ 25,-	€ 25,00

omschrijving	Prijs per implementatie	Bedrag
Implementatie	€ 25	€ 25,00

omschrijving	Overige kosten(eenmalig)	Bedrag
Vervoerskosten	€ 50,-	€ 50,00

Voor akkoord,

Bedrijf	: JT Development	Uw bedrijfsnaam	: <u>Humple Dumpie</u>
Naam	: <u>Jorrit Meeuwissen</u>	Naam	: Marlies Aarts
Plaats	: Breda	Functie	: Eigenaar
		Datum	: 11-4-2017
		Plaats	: Bavel

Handtekening:



Uw handtekening :

Prototype (wireframes) gebaseerd op interview

Nu heel de opdracht duidelijk is en alle eisen goed staan kon ik de prototypes gaan maken. Bij de prototypes heb ik alle schermen getekend die nodig waren om de applicatie een logische volgorde te geven en ook alle eisen erin te verwerken. Alle schermen die in de applicatie moeten komen zijn getekend zodat er een goed overzicht ontstaat van hoe de applicatie eruit komt te zien en hoe de applicatie gaat werken.

De prototypes zijn besproken met de opdrachtgever en uitgelegd hoe de volgorde werkt en wat alle knoppen voor functies hebben.

De opdrachtgever heeft toen goedkeuring gegeven voor de prototypes.

Hierdoor weet ik ook beter hoe mijn applicatie moet gaan werken en eruit moet komen te zien.

Alle prototypes zijn te vinden in de map *1.4 Prototypes* bij het document *prototypes.docx*. In dit document zie je alle prototypes en met uitleg hoe de volgorde van de prototypes in elkaar zit.

KT 1-2 Maakt een plan van aanpak (E, J, Q)

Plan van aanpak

Het plan van aanpak is gemaakt om te omschrijven hoe we precies het project gaan uitvoeren en welke stappen wij allemaal gaan uitvoeren om het project succesvol te laten verlopen. In het plan van aanpak staan verschillende onderwerpen over hoe we het succesvol gaan laten verlopen.

Het plan van aanpak is te vinden in de map *1.5 plan van aanpak* onder de naam *Plan van aanpak.docx*. Hieronder staan korte beschrijvingen van wat je allemaal in het plan van aanpak kan vinden.

Het eerste stuk gaat over achtergronden. Daarin is beschreven wie de opdrachtgever is en wat het bedrijf allemaal doet. En wat wij voor hun moeten gaan doen.

Dit betekent dat de huidige situatie is beschreven en wat ze willen bereiken. Ook staat erin hoe de communicatie is geregeld.

Na achtergronden wordt verteld wat de doelstellingen zijn van dit project. Hierin staat waarom het project wordt uitgevoerd en wat de doelstellingen zijn. Dus wat het bedrijf wil bereiken met de app.

Daarna heb je het onderdeel projectopdrachten. Hierin staat wat er allemaal in het eindproduct moet komen en aan welke eisen wij allemaal moeten houden. Dus wat je allemaal met de app moet kunnen en hoe het allemaal moet werken.

Om het eindproduct te maken heb je ook verschillende activiteiten die je moet doen om het succesvol te laten verlopen. Dit staat onder het kopje

projectactiviteiten. Hierin staat welke dingen ik allemaal moet doen om meer informatie te krijgen over het project. Hierbij horen verschillende documenten die allemaal gemaakt moeten worden om het project volledig en goed uitwerkt te krijgen.

Bij de projectgrenzen staat beschreven hoelang het project gaat duren en wat de werktijden zijn. De opdracht begint op 10-4-2017 en het de opdracht moet helemaal klaar zijn op 16-6-2017. Er staat ook in wat ik allemaal in de opdracht moet zetten en wat mijn werkwijze ervoor is. Ook staat erin aan welke voorwaarden we moeten voldoen.

Voor het project maak ik verschillende producten die ik oplever. De producten die ik allemaal moeten opleveren heb ik in de planning gezet van mijn project. In de planning staat dus wanneer alles af moet zijn.

Ik wil ook een goede kwaliteit leveren. Dit doe ik door alles te laten controleren.

In het kopje kwaliteit staat hoe de waarborging van de kwaliteit van het eindproduct en de tussenproducten is geregeld.

Ook staat erin welke controles ik doe om de kwaliteit te behouden en welke terugkoppelingen ik doe met de opdrachtgever. Er staat ook in welke afspraken zijn opgenomen over de te gebruiken hulpmiddelen.

Bij de organisatie staat in welke functies er verdeeld zijn in mijn projectgroep. Hierbij staan de verschillende namen van de groepsleden met de gegevens van de persoon. Er staat wanneer de leden beschikbaar moet zijn.

Er staat ook hoe de interne communicatie tussen de projectgroep is.

Daarnaast ook hoe de communicatie met de opdrachtgever is geregeld.

Bij de communicatie staat ook hoe ik de tijdsregistratie regel en hoe vaak er vergaderd wordt.

Elke week wordt er ook een weekverslag gemaakt van de activiteiten die elk groepslid heeft gedaan.

Bij de planning staan alle activiteiten die uitgevoerd moeten worden. De planning is gemaakt in MS-Project. In de planning is aangegeven wie welke taak op zich moet nemen. De planning is realistisch gemaakt zodat het haalbaar is om het eindproduct goed af te ronden.

Bij de kosten en baten zijn alle kosten in een overzicht gezet met een korte toelichting erbij. De kosten en baten zijn geschatte prijzen zodat ze niet de precieze kosten zijn, want die kan ik nog niet precies weten. Er kan namelijk altijd verandering in prijzen komen.

Bij de risico's staan alle risico's die mogelijk zijn bij het project.

De risico's zijn onderverdeeld in 2 categorieën:

- Interne risico's
- Externe risico's

Hierbij is ook nog een risicoanalyse gemaakt waarin staat wat de kans precies is dat dit een risico gaat worden in het project en hoe belangrijk risico het is.

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 kostte 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 gepland 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.

	Taaknr	Taaknaam	Duur	Beginndatum	Einddatum	Voorafgaande taken	Resourcenamer	zuwe kolom to
		▲ Kerntaak 1.2	1,75 dagen	don 20-4-17	vri 21-4-17			
		Plan van aanpak maken	3 uur	don 20-4-17	don 20-4-17	9	Jorrit;Teun	
		planning maken	3 uur	vri 21-4-17	vri 21-4-17		Teun	
		▲ Kerntaak 1.3	1,75 dagen	vri 21-4-17	din 2-5-17			
		Use case diagram	2 uur	maa 1-5-17	maa 1-5-17	11;12	Teun	
		use case template	2 uur	maa 1-5-17	maa 1-5-17	14	Teun	
		Prototype toelichting volgorde	1 uur	vri 21-4-17	vri 21-4-17	11;12	Jorrit	
		Ontwerpdocus - functioneel ontwerp	2 uur	din 2-5-17	din 2-5-17	14;15;16	Jorrit	
		▲ Kerntaak 1.4	1,5 dagen	din 2-5-17	woe 3-5-17			
		Modeldictonar	1 uur	din 2-5-17	din 2-5-17	17	Jorrit	
		Klassendiagram	1 uur	din 2-5-17	din 2-5-17	17	Teun	
		Datadictionary	2 uur	woe 3-5-17	woe 3-5-17	19	Jorrit	
		Sequentiediagr	1 uur	din 2-5-17	din 2-5-17	20	Teun	
		Ontwerpdocus - technisch ontwerp	2 uur	woe 3-5-17	woe 3-5-17	19;20;21;22	Teun	

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.

✓	📁	▸ Kerntaak 1.2	0,75 dagen	din 18-4-17	din 18-4-17			
✓	📁	Plan van aanpak maken	3 uur	din 18-4-17	din 18-4-17	9	Jorrit;Teun	
✓	📁	planning maken	3 uur	din 18-4-17	din 18-4-17		Teun	
✓	📁	▸ Kerntaak 1.3	1 dag	woe 19-4-17	woe 19-4-17			
✓	📁	Use case diagram	2 uur	woe 19-4-17	woe 19-4-17	11;12	Teun	
✓	📁	use case template	2 uur	woe 19-4-17	woe 19-4-17	14	Teun	
✓	📁	Prototype toelichting volgorde	2 uur	woe 19-4-17	woe 19-4-17	11;12	Jorrit	
✓	📁	Ontwerpdocus - functioneel ontwerp	2 uur	woe 19-4-17	woe 19-4-17	14;15;16	Jorrit	
	📁	▸ Kerntaak 1.4	5 dagen	don 20-4-17	woe 3-5-17			
✓	📁	Modeldictonar	1 uur	don 20-4-17	don 20-4-17	17	Jorrit	
✓	📁	Klassendiagram	1 uur	don 20-4-17	don 20-4-17	17	Teun	
📅	📁	Datadictionary	2 uur	woe 3-5-17	woe 3-5-17	19	Jorrit	
✓	📁	Sequentiediagr	1 uur	don 20-4-17	don 20-4-17	20	Teun	
	📁	Ontwerpdocus - technisch ontwerp	2 uur	woe 3-5-17	woe 3-5-17	19;20;21;22	Teun	

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.

KT 1-3 Levert een bijdrage aan een functioneel ontwerp of game design document (E, I, J, K, U)

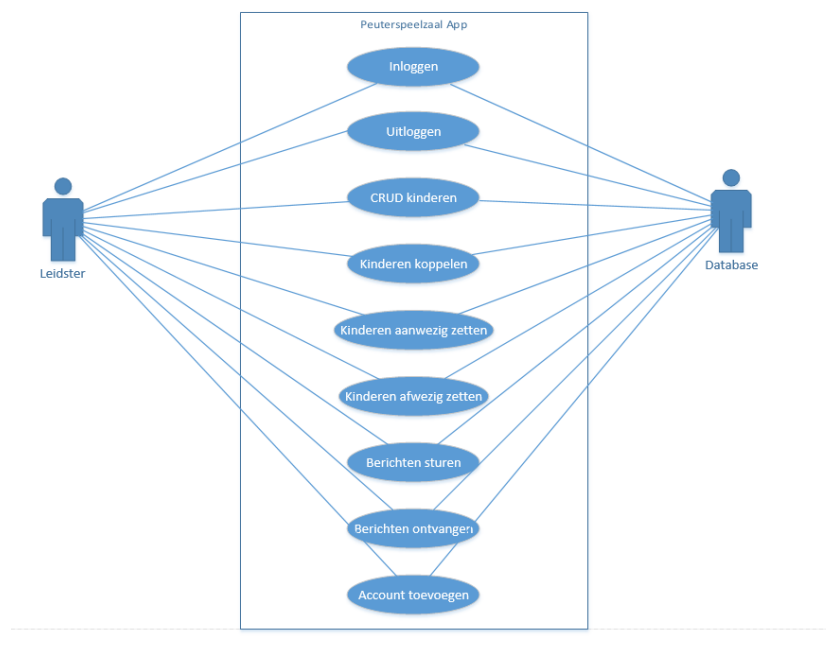
Ontwerpdocus – Functioneel ontwerp

Het functioneel ontwerp bestaat uit use-case diagrammen, use-case templates en een toelichting van de volgorde van de prototypes. In het functioneel ontwerp staat beschreven wat er met die drie onderwerpen bedoeld wordt 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 Templates

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.

Kinderen toevoegen

Naam	Kinderen toevoegen
Versie	1.0
Actor	Database, Leidster
Preconditie	Leidster moet ingelogd zijn in de app.
Beschrijving	Leidster klikt op het menu knopje. Dan komt er een menu met een knopje beheer. Leidster gaat dan naar de beheer pagina waar een knop met kinderen toevoegen. Leidster moet dan alle gegevens van het kind invullen. Leidster klik op opslaan.
Uitzonderingen	Als niet alle vakken zijn ingevuld dat kan het kind niet worden toegevoegd. Als het kind er al instaat dan zal het kind niet nog een keer worden toegevoegd. De Leidster is niet ingelogd.
Niet-functionele eisen	De Leidster moet ingelogd zijn.
Postconditie	Kind is opgeslagen in het systeem.

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.

Prototype toelichting volgordeijkheid

Bij de prototypes is uitgelegd hoe alles werkt.

Dus bij elke knop dat je kan indrukken is verteld wat je ermee kan doen en als je iets wilt uitvoeren net als een kind toevoegen, staat er uitgelegd hoe je dat precies doet dus welke stappen je allemaal moet uitvoeren voordat je dat resultaat hebt behaald.

Leidsters

<u>Inloggen:</u>	1→gegevens invullen en op de knop inloggen drukken
<u>Uitloggen:</u>	1→gegevens invullen en op de knop inloggen drukken→2→Op de menu knop drukken→4→op uitloggen drukken
<u>Lijst met kinderen voor huidige dag:</u>	1→gegevens invullen en op de knop inloggen drukken→2
<u>Kind detail bekijken:</u>	1→gegevens invullen en op de knop inloggen drukken →2→Op een kind drukken→3
<u>Afwezigheid bekijken van een kind:</u>	1→ gegevens invullen en op de knop inloggen drukken→2→Op een kind drukken→3→Op de knop aanwezigheid drukken→5
<u>Leidster menu:</u>	1→gegevens invullen en op de knop inloggen drukken→2→Op de menu knop drukken→4
<u>Beheer pagina bekijken:</u>	1→gegevens invullen en op de knop

Prototypes volgordeijkheid document

Alle volgordes voor de prototypes zijn te vinden in de map *1.9 Prototype toelichting volgordeijkheid* bij het document *prototypes volgorde.docx*.

KT 1-4 Maakt een technisch ontwerp (E, I, J, K, P)

Ontwerpdokument – 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 wordt 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.

Modeldictionary

Om de modeldictionary te maken moet ik verschillende stappen uitvoeren.

De eerste stap is om de kandidaatklassen te identificeren.

Dit heb ik gedaan door bij de hernieuwde opdracht alle zelfstandige naamwoorden te selecteren.

Humpie Dumpie is een **peuterspeelzaal**. Per week zijn er 4 **dagdelen** die worden verdeeld in 2 **groepen**. Als er een **kind** niet aanwezig is kunnen ze het alleen opschrijven en dan de **ouders** bellen hoe het komt. De ouders kunnen de **absentie** van hun kinderen opgeven doormiddel van te bellen naar de peuterspeelzaal. Dit is niet overzichtelijk omdat het lastig is om alles goed bij te houden.

Het contact met de ouders werkt zo ook niet goed want dan moeten ze via het kind het **telefoonnummer** van de ouder vinden en dan kunnen ze de ouders pas contacteren.

Gewenste situatie

De peuterspeelzaal wil graag een **app** hebben waar ze een **digitale lijst** hebben met de kinderen van die **dag** en waarbij ze al kunnen zien of er een absentie voor een kind is toegevoegd. Als er dan een kind afwezig is en geen absentie heeft kunnen de leidsters een bericht naar de ouders sturen over het kind dat niet aanwezig is. Dan kunnen de ouders een bericht terugsturen met de reden. Hierdoor wordt het bijhouden van de kinderen erg overzichtelijk en werkt alles een stuk sneller.

Kandidaatklassen identificeren

Daarna heb ik gekeken of de zelfstandige naamwoorden geschikt zijn. Toen ben ik gaan kijken of het een klasse is of iets anders zoals een irrelevant, vaag, niet kwantificeerbaar, attribuut. Dit heb ik in een schema gezet zodat ik er een goed overzicht van heb.

Kandidaatklasse	Beslissing m.b.t. geschiktheid
Peuterspeelzaal	Irrelevant
Dagdelen	Te vaag
Groepen	Klasse
Kind	Redudant met Kinderen
Ouders	Klasse
Absentie	Klasse
Telefoonnummer	Attribuut
App	Implementatie
Digitale lijst	Te vaag
Dag	Attribuut
Kinderen	Klasse
Leidsters	Klasse
Afmeldingen	Te vaag
Aanwezig	Te vaag
Afwezig	Te vaag
Gegevens	Attribuut
Bericht	Redudant met Berichten
Accounts	Redudant met Kinderen, Ouders en Leidsters
Foto's	Attribuut
Chat	Klasse

Resultaten van klasseselectie

Nu ik alle klassen heb gevonden kan ik ze gaan uitwerken in een tabel met een omschrijving van elke klasse.

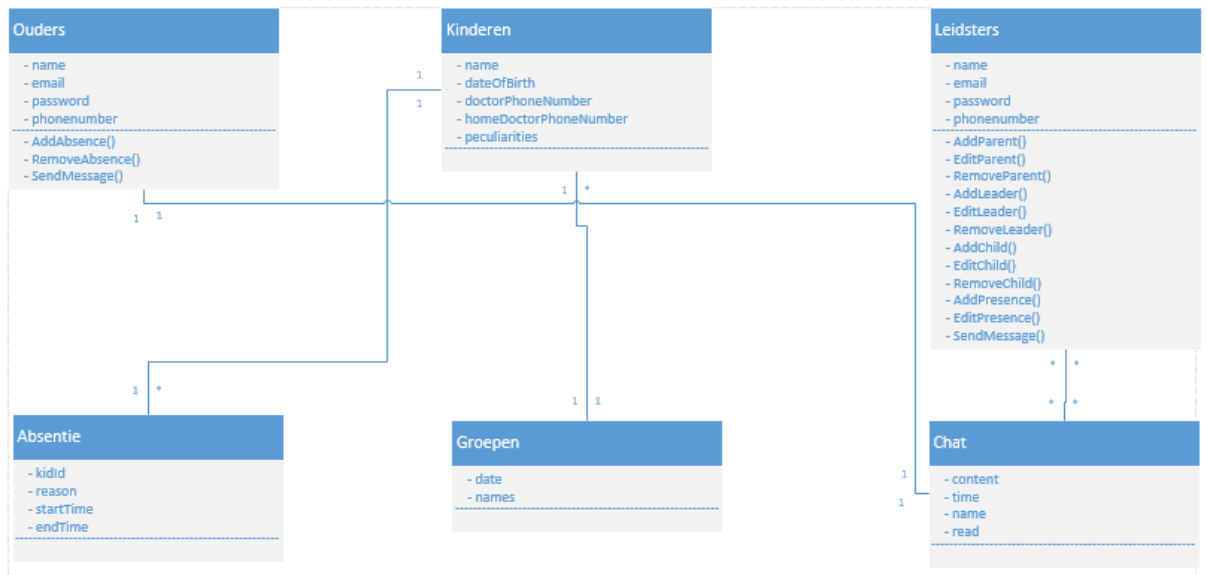
Klasse	Omschrijving
Groepen	Een groep wordt aangemaakt door <i>Leidsters</i> en in een groep zitten <i>Kinderen</i>
Ouders	Een gebruiker die gekoppeld is aan een <i>kind</i> en absenties kan invoeren voor een <i>kind</i>
Absentie	Een Absentie kan toegevoegd en verwijderd worden door <i>Ouders</i> en is in te zien door <i>Leidsters</i>
Kinderen	Een account dat gekoppeld kan worden aan een <i>Groepen</i> en waar een <i>Absentie</i> aan toegevoegd kunnen worden.
Leidsters	Een account dat kinderen aan en afwezig kunnen zetten en <i>Groepen</i> kan aanmaken, wijzigen of verwijderen
Chat	In een chat zitten berichten die aangemaakt worden door <i>Leidsters</i> en <i>Ouders</i>

Modeldictionary

Klassendiagram

Nadat ik de modeldictionary heb gemaakt zijn alle klassen duidelijk geworden. Dus kon ik een klassendiagram van de klassen gaan maken. De klassendiagram heb ik gemaakt met het programma Visio.

Bij alle klassen in de klassendiagram staan de attributen die een klas bevat en welke acties de klas allemaal heeft. De klassen zijn met elkaar verbonden. Er zijn namelijk multipliciteiten aangegeven. Neem als voorbeeld Kinderen en groepen. 1 kind heeft namelijk 1 groep maar 1 groep heeft meerdere kinderen.



Klassendiagram met multipliciteiten

Datadictionary

In de datadictionary zijn alle attributen uitgewerkt die in een klas moeten komen. Voordat ik ben begonnen met de datadictionary heb ik eerst een overzicht van de database gemaakt, dus hoe de database eruit moet komen te zien. Dit heb ik gedaan doormiddel van normaliseren.

De genormaliseerde database is te vinden in de map *1.13 Datadictionary* bij het bestand *database normaliseren.xlsx*.

Toen ik de database had genormaliseerd ging ik de datadictionary maken.

In de datadictionary staat elke attribuutnaam van een tabel met daarbij de volgende gegevens:

- Naam : De naam van de attribuut
- Attribuut : Of het een attribuut bevat net als AUTO-INCREMENT
- Type : Wat soort attribuut het is zoals INT, VARCHAR etc.
- Lengte : De lengte van het attribuut
- Verplicht : Of het attribuut gevuld moet zijn
- Omschrijving : De omschrijving van het attribuut
- Keys : Wat voor key het is zoals Primary key of Foreign Key

tbl_users						
Naam	Attribuut	Type	Length	Verplicht	Omschrijving	Keys
<u>id</u>	AUTO-INCREMENT	INT	*	Yes	<u>Id</u> voor de gebruiker	Primary key
<u>kidid</u>		INT	*	No	<u>Id</u> voor het kind van de gebruiker	Foreign key
<u>role</u>		VARCHAR	50	Yes	De rol van de gebruiker	
<u>name</u>		VARCHAR	50	Yes	De naam van de gebruiker	
<u>email</u>		VARCHAR	50	Yes	De email van de gebruiker	
<u>password</u>		VARCHAR	50	Yes	Het wachtwoord van de gebruiker	
<u>phonenummer</u>		VARCHAR	50	No	Het telefoonnummer van de gebruiker	

Datadictionary

De volledige datadictionary is te vinden in de map *1.13 Datadictionary* bij het document *Datadictionary.docx*

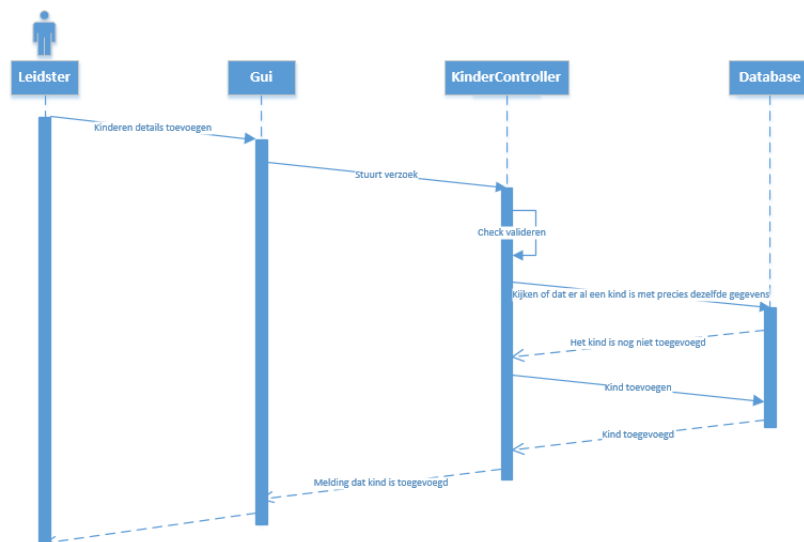
Sequentiediagram

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.

Kinderen toevoegen

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



KT 1-5 Richt de ontwikkelomgeving in (J, L, Q, S)

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.

kerntaak 1 Inrichten	0,08 dagen	maa 10-4-17	maa 10-4-17		
Microsoft office 2013	20 min	maa 10-4-17	maa 10-4-17		Jorrit;Teun
Sourcetree	3 min	maa 10-4-17	maa 10-4-17		Jorrit;Teun
Skype	3 min	maa 10-4-17	maa 10-4-17		Jorrit;Teun
Adobe photoshop cs6	5 min	maa 10-4-17	maa 10-4-17		Jorrit;Teun

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.

✦ Kerntaak 2 Inrichten ontwikkelomgeving	0,02 dagen	maa 8-5-17	maa 8-5-17		
Firebase	3 min	maa 8-5-17	maa 8-5-17		Jorrit;Teun
phpstorm	3 min	maa 8-5-17	maa 8-5-17		Jorrit;Teun
Ionic	5 min	maa 8-5-17	maa 8-5-17		Jorrit;Teun
Cordova	5 min	maa 8-5-17	maa 8-5-17		Jorrit;Teun
NPM	5 min	maa 8-5-17	maa 8-5-17		Jorrit;Teun

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.

Materialenlijst: Hard- en software van de ontwikkelomgeving

Voor het project heb ik verschillende hard- en software gebruikt om alle producten te maken voor dit project. Hier heb ik een lijst van gemaakt.

In het document staat ook hoe de ontwikkelomgeving ingericht moet worden.

Er staat namelijk welke github repositorie gebruikt moet worden en welke database gegevens.

Daarnaast staan alle hard- en software in een lijst met alle versie nummers erbij en hoelang het duurt om het te installeren en configureren.

Per groepslid is er een lijst gemaakt aangezien soms de versienummers kunnen verschillen.

Software:		
Programma Naam	Versie	Installatietijd
Visio 2013	15.0.4753.1000	+/- 5 min
Microsoft Office 2013 (word, excel, powerpoint)	15.0.4753.1003	+/- 15 min
MS Project 2013	15.0.4753.1000	+/- 5 min
PHPStorm	9.0.2	+/- 3 min
Sourcetree	1.10.20.1	+/- 3 min
Skype	7.32.0.104	+/- 3 min
Windows laptop	Windows 10	+/- 30 min
Adobe photoshop CS6	13.0x64	+/- 5 min
Ionic	5.5.12	+/- 2 min
NPM	HTML5	n.v.t
Cordova	5.6.17	n.v.t

Hardware:	
Hardware naam	Omschrijving
Lenovo Y500	Besturingssysteem: Windows 10
Sony xperia z5	Android versie: 6.0.1

hard- en software van teun

Het document met alle hard- en software en de andere gegevens staat in de map *1.16 Materialenlijst Hard - en software van de ontwikkelomgeving* bij het document *Materialenlijst.docx*.

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.

Als iemand van het team een fout maakt in git dan is er de optie om dat terug te kunnen draaien. Hieronder vind je de uitleg om dit te kunnen terug draaien. Dit is van de opties die je kan gebruiken.

- Kies een commit in de log view die je wilt terugzetten
- Selecteer de file die je wilt gaan terugzetten
- Als je dan op de Reverse knop boven aan klikt dan zet sourcetree heel die file terug. Maar als je maar een stukje van de file wilt terugzetten dan kan je drukken op Reverse hunk of Reverse selected lines.
- Deze veranderingen worden dan in je working copy gezet en zijn dan klaar om getest en gecommit te worden.

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.

Verslag over de installatie, configuratie en testen van de 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 er 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.

Taken	Gelukt	Niet gelukt
Node.js installeren		
NPM in terminal kunnen gebruiken		
Ervoor zorgen dat je <u>cordova</u> en <u>ionic</u> tools geïnstalleerd zijn		
Ervoor zorgen dat je <u>ionic</u> project gemaakt hebt.		
<u>Ionic</u> project in de browser laten draaien		
<u>Ionic</u> project op je mobiel laten draaien		

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.