

Project approach tool

EEN APPLICATIE OM HBO-I STUDENTEN TE HELPEN MET HUN
PROJECTAANPAK.

STUDENTNUMMER

432889

PERIODE

MEI-NOVEMBER 2020

BEGELEIDENDE DOCENT

ANTHONY VAN DEN BERG

BEDRIJFBEGELEIDER

DANNY PLASS

Versiebeheer

Versie	Beschrijving	Datum
0.1.1	Begin maken afstudeerverslag	29-05-2020
0.1.2	Review stakeholder analyse, DOT-framework onderzoek	03-06-2020
0.2.1	Visie voor online collaboratie toegevoegd	08-06-2020
0.2.2	Keuzeverantwoording toegevoegd voor online collaboratie	09-06-2020
0.2.3	Benchmarktest toegevoegd	10-06-2020
0.2.4	Betrokkenheid stakeholder analyse toegevoegd	11-06-2020
0.3.1	Benchmarktest data, server aanpassing toegevoegd	16-06-2020
0.3.2	Implementatie sockets, context toepassing toegevoegd en eindconclusie benchmark uitgebreid	03-07-2020
0.4.1	Loadtesting toegevoegd	24-08-2020
0.4.2	Visuele componenten toegevoegd & tekst scan uitgevoerd	07-09-2020
0.5.1	Visuele componenten aangepast	14-09-2020
0.5.2	Authenticatie toegevoegd	18-09-2020
0.5.3	Authenticatie uitbreiden met library vergelijking	19-09-2020
0.6.1	Onboarding toevoegen	28-09-2020
0.6.2	Gebruikerstesten voor het onboarding toevoegen + extra aanbevelingen toegevoegd	30-09-2020
0.6.3	Beginsituatie analyse, onboarding afgemaakt	01-10-2020
0.6.4	Gebruikerstesten toegevoegd voor online collaboratie, hoofdstuk aanpak is ook toegevoegd.	02-10-2020
0.6.5	Samenvatting en inleiding toevoegen	03-10-2020
0.7.1	Beginsituatie uitbreiden	12-10-2020
0.7.2	Eindsituatie toegevoegd en begonnen met GDPR	13-10-2020
0.7.3	GDPR uitbreiden en feedback verwerken	14-10-2020
0.7.4	Verder gegaan met het verwerken van feedback	15-10-2020
0.7.5	Realtime connectie opties toegevoegd	16-10-2020
0.7.6	Verder gegaan met het verwerken van feedback	17-10-2020
0.8.1	Feedback verwerken	19-10-2020
0.8.2	Test resultaat in bijlages gestopt en lege plekken weggehaald om meer ruimte te creëren	20-10-2020
0.8.3	feedback van danny verwerkt	22-10-2020
0.8.4	Veel naar de bijlages verplaatst en implementatie toegevoegd	23-10-2020
0.8.5	Veel ik vorm weggehaald en tekst scan gedaan	25-10-2020

Samenvatting

De project approach tool is een applicatie waarmee gebruikers een project aanpak kunnen maken, door de concepten en kaarten te gebruiken van het DOT-framework, ICT-research methods kaarten en de stepping stones van miriam losse.

Het doel van de opdracht was om de applicatie uit te breiden zodat gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent. Om dit te realiseren is het project begonnen met een requirement analyse, dit is een proces waarbij de wensen en eisen van de stakeholders worden gedefinieerd. Toen kwam ik er al snel achter dat de opdracht feitelijk uit twee onderdelen bestaat, die samen de oplossing vormen.

Het eerste onderdeel zal ervoor zorgen dat de gebruikers samen kunnen werken, hiervoor is er een concept uitgewerkt waarbij sockets gebruikt worden, hiervoor is er onderzoek gedaan om te kijken welke socket libraries het meest geschikt zijn voor deze applicatie. Daarna zijn er mock-ups gemaakt, tijdens het maken van de mock-ups is er nog onderzoek gedaan, naar welke visuele effecten andere applicaties gebruiken om het online collaboratie proces te ondersteunen, om inspiratie op te doen voor de visuele effecten die geïmplementeerd kunnen worden in de project approach tool. Na het implementeren van het online collaboratie proces in deze applicatie, zijn er meerdere testen uitgevoerd om aan te geven dat het gewenste doel behaald is.

Het tweede onderdeel zorgt ervoor dat gebruikers zelfstandig met de applicatie om kunnen gaan, dit wordt gedaan door een onboarding proces te realiseren waarbij alle functionaliteiten in de applicatie uitgelegd worden. Er is een gebruikerstest uitgevoerd voordat het onboarding proces geïmplementeerd werd, om te kijken hoe gebruikers omgaan met de applicatie zonder een onboarding proces en waar de meeste problemen zitten en een gebruikerstest nadat het onboarding proces geïmplementeerd werd om te kijken hoeveel effect het heeft gehad en wat er eventueel nog aan verbeterd kon worden.

Uiteindelijk is er een product opgeleverd waarmee gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent.

Inhoudsopgave

1	Inleiding.....	5
2	Projectachtergrond	6
2.1	Context.....	6
2.2	DOT-framework	6
2.3	Beginsituatie	9
2.4	Gewenste situatie	14
2.5	Onderzoeksvragen	15
2.6	Interne stakeholders	15
3	Aanpak	17
3.1	Communicatie	17
3.2	Scrum	17
3.3	Kwaliteitscontrole	18
3.4	Tools.....	18
4	Requirements.....	19
5	Authenticatie	19
5.1	Visie.....	20
5.2	Authenticatie service	20
6	Online collaboratie.....	22
6.1	Realtime connectie opties	22
6.2	Concept.....	23
6.3	Socket library	27
6.4	Load testen.....	31
6.5	Visuele componenten	33
6.6	Implementatie.....	36
6.7	Gebruikerstest.....	38
7	Applicatie onboarding.....	40
7.1	Onboarding informatie	40
7.2	Types	41
7.3	Mock-ups.....	42
7.4	Onboarding library.....	43
7.5	Gebruikerstest.....	44
8	Eindsituatie	46

8.1	GDPR	50
9	Eindconclusie	51
9.1	Was het succesvol?	51
9.2	Aanbevelingen	51
9.3	Reflectie	55
10	Bibliografie	55
11	Bijlages	58
11.1	Requirements.....	58
11.2	Error log	60
11.3	GDPR-onderzoek.....	61
11.4	Benchmark resultaat.....	63
11.5	Loadtest resultaat	64
11.6	Gebruikerstest online collaboratie resultaat.....	65
11.7	Gebruikerstest onboarding resultaat.....	66
11.8	Scrumboard.....	67
11.9	mock-ups onboarding	68

Begrippenlijst

Begrip	Beschrijving
MEAN-stack	MEAN staat voor MongoDB, Express, Angular en Node.JS. Dit zijn javascript technologieën die samen een MEAN stack vormen.
Workspace	Dit is de term die gebruikt wordt voor het bord waar een project aanpak op gepland zal worden.
DOT-Framework	DOT-Framework staat voor Development Oriented Triangulation Framework. Dit is een onderzoeksmethode.
Kennisdomein	Een duidelijk begreemd kennisgebied dat gaat over een specifiek onderwerp
Deliverable	Een deliverable is een tastbaar of immaterieel goed of een service die is geproduceerd als resultaat van een project of fase dat bedoeld is om aan een klant te worden geleverd. Een deliverable kan een rapport, een document, een softwareproduct, een serverupgrade of een andere bouwsteen van een algemeen project zijn.
MoSCoW	Dit is een prioritering techniek voor requirements. MoSCoW staat voor Must, Should, Could, Wont.
F/NF	Dit wordt gebruikt tijdens een requirement analyse om aan te geven of een requirement functioneel of niet functioneel is.

Applicatie onboarding	Applicatie onboarding is een proces waarbij de gebruiker begeleid wordt door de gehele applicatie als het de eerste keer is dat de gebruiker de applicatie gebruikt
PAT	Project Approach Tool, dit is de naam voor de applicatie.
Realtime	De werkelijke tijd waarin een proces of gebeurtenis plaatsvindt.
Benchmark	een onderzoek waarbij de prestaties van organisaties, producten of programma's op identieke wijze worden onderzocht en vergeleken.
Library	een verzameling code (functies/routines) die door programma's kunnen worden gebruikt.
Observable	Een observable is een object die bij het aanmaken een component registreerd en observeerd. De observable wordt geobserveerd voor data en als er dan data binnen komt dan kan daar wat mee gedaan worden.
Room	Een room is in dit geval een andere term voor een socket cluster.
Group	Een group is een andere term dat gebruikt wordt voor phase. Dit zal meer voorkomen in code.
Load test	dat zijn testen waarbij de gebruikersbelasting van een systeem getest wordt.
Typescript-linter	Een linter is een statische code analyse tool dat gebruikt wordt om fouten, bugs te vinden en een code style te forceren.
CamelCase	CamelCase is samengestelde woorden of zinnen, waar de woorden met een spatie gescheiden worden, als één woord te schrijven door de spatie en de daarop volgende letter te vervangen door de corresponderende hoofdletter.

1 Inleiding

Dit verslag is geschreven als afstudeeropdracht als afsluiting voor de HBO-ICT software engineering opleiding aan de hogeschool Saxion ter Enschede. Het afstudeertraject heeft plaatsgevonden van 5 mei tot en met 13 november 2020 bij het lectoraat Ambient intelligence op het Saxion.

De project approach tool is een applicatie waarmee gebruikers een project aanpak kunnen maken, door de concepten en kaarten te gebruiken van het DOT-framework, ICT-research methods kaarten en de stepping stones van Miriam losse, over deze concepten en kaarten is meer informatie te vinden in hoofdstuk DOT-framework.

Het doel van de opdracht was om de applicatie uit te breiden zodat gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent. Tijdens het eerste gesprek met de product owner kwam al snel de hoofdvraag:

“Hoe kan de applicatie worden uitgebreid, zodat gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent?”

In dit document wordt die vraag behandeld, door de opzet van het project, de processen tijdens het project en de resultaten van het project te beschrijven.

2 Projectachtergrond

2.1 Context

Elke project heeft een goede projectaanpak nodig om een goede oplossing op te leveren. De applicatie die uitgebreid zal worden, zal studenten en professionals helpen met het nadenken over wat er opgeleverd moet worden tijdens een project, welke informatie je nodig hebt om dit doel te bereiken en hoe je aan deze informatie kunt komen. Hiervoor past de tool de concepten en kaarten uit het DOT-framework, ICT-research methods kaarten en Miriam Losse's Stepping stones toe.

Deze kaartensets zijn merendeels nog plastic of papieren kaarten en met deze applicatie worden de kaarten gedigitaliseerd en kunnen ze gebruikt worden om een projectaanpak in te plannen en hoewel deze sets een goede basis geven, kan de online tool toch extra mogelijkheden bieden.

2.2 DOT-framework

Het DOT-framework (Plass, 2020) helpt mensen bij het creëren van een structuur van de onderzoeksactiviteiten. Het framework bestaat uit 3 lagen: wat, hoe en waarom. Deze 3 lagen hebben samen als doel om van een vraag naar een oplossing te komen.

Wat – De kennisdomeinen waarover je informatie naar boven kunt halen.

Er zijn 3 verschillende domeinen die je gebruikt om informatie op te halen:

Toepassingscontext

1. Het eerste domein is toepassingscontext, dit is de context waar het project zich in afspeelt. Denk bijvoorbeeld aan een werkproces dat geanalyseerd kan worden om meer informatie te verkrijgen over hoe de gebruikers de applicatie moeten kunnen gebruiken.

Beschikbaar werk

2. Het tweede domein is beschikbaar werk, dit zijn de producten en informatie wat al beschikbaar is of experts die beschikbaar zijn.

Innovatie

3. Het derde domein is innovatie, dit is waar alle innovaties plaats vinden.

Hoe - De onderzoeksmethodes die je helpen om de informatie te verkrijgen.

Tijdens een project probeer je zoveel mogelijk informatie te verkrijgen over de 3 kennisdomeinen. Het framework groepeerde de methodes om informatie te verkrijgen in 5 onderzoeksstrategieën. Meer informatie over de beschikbare onderzoeksmethodes zal te vinden zijn in het volgende hoofdstuk.

1. Bibliotheek

Binnen het bibliotheekonderzoek kun je erachter komen wat er al bestaat, welke guidelines er zijn en welke theorieën al bestaan.

2. Veld

Binnen het veldonderzoek kun je erachter te komen hoe de toepassingscontext in elkaar zit, wie zijn de gebruikers, wat hebben ze nodig, wat verlangen ze en wat zijn de limitaties.

3. Lab

Binnen het labonderzoek kun je onderdelen, oplossingen, concepten testen. Je kunt er dan achter komen of het werkt zoals het bedoeld was.

4. Showroom

Binnen het showroomonderzoek kun je de oplossing vergelijken met bestaand werk of kun je experts raadplegen over de oplossing.

5. Werkplaats

Binnen het werkplaatsonderzoek worden mogelijkheden verkent door middel van verschillende methodes zoals brainstormen en prototyping.

Waarom - De afwegingen die duidelijk maakt waarom een methode gekozen is.

Uitleggen wat je wilt bereiken kan al veel helpen met structureren van je aanpak. Daarom zijn er een aantal afwegingen die gemaakt moeten worden als je een onderzoeksmethode kiest.

1. Fit of expertise

Er zijn namelijk weinig methodes die voor een betere fit met de toepassingscontext zorgen en die zorgen voor een betere aansluiting op de expertise, daarom is het ook vaak een aanrader om methodes te gebruiken uit beide domeinen binnen een project.

2. Overzicht of zekerheid

Voor overzicht en zekerheid, geldt dat je in het begin vaak zoveel mogelijk overzicht wilt creëren over wat nodig is en wat al beschikbaar is en in een later stadium wil je graag meer zekerheid creëren door bijvoorbeeld specifieke onderdelen te testen etc.

3. Data of inspiratie

In de meeste gevallen zul je data-gericht de keuzes onderbouwen. Maar je zult ook gebruik kunnen maken van inspiratie-gerichte methodes, waarin jijzelf persoonlijk de antwoorden bepaald zoals brainstormen.

2.2.1 Kaartensets

Er zijn 2 verschillende kaartensets die momenteel gebruikt worden binnen de applicatie:

- **ICT-research methodes**

Dit een kaartenset die bestaat uit onderzoeksmethodes, die aansluiten op de 5 onderzoeksmethodes genoemd in de uitleg over het DOT-framework. Bijvoorbeeld: brainstormen,

- **Stepping stones**

Dit is een kaartenset die bestaat uit deliverables, methodes, beroepsprocessen en reflectiekaarten. Deze kaarten hebben geen relatie met de ICT-research methodes, maar passen in de praktijk heel goed bij de ICT-research methodes.

2.2.2 Toepassing

Hieronder zal informatie te vinden zijn over hoe het DOT-framework en de verschillende kaartensetten gebruikt worden om een projectaanpak op te zetten.

De studenten beginnen met het verzinnen van een titel en een doel voor het project, dit is belangrijk om het project herkenbaar te maken binnen de applicatie. Daarna wordt een beroepsproces gekozen, dit kan een standaard beroepsproces zijn zoals het V-model of het advies proces, deze zullen als template beschikbaar zijn tijdens het aanmaken van een nieuwe workspace. Het is ook mogelijk om lege workspace te gebruiken als het gewenste beroepsproces niet beschikbaar is als template.

Nu dat de workspace aangemaakt is, kunnen de stepping stones kaarten gebruikt worden om in kaart te brengen wat er opgeleverd moet worden en welke informatie hiervoor nodig is. Nu dat in kaart is gebracht welke informatie nodig is, kan er nagedacht worden over hoe de benodigde informatie verzameld wordt. Hiervoor kunnen de ICT-research methodes kaarten gebruikt worden. Door beide kaarten te gebruiken is er een complete project aanpak opgebouwd.

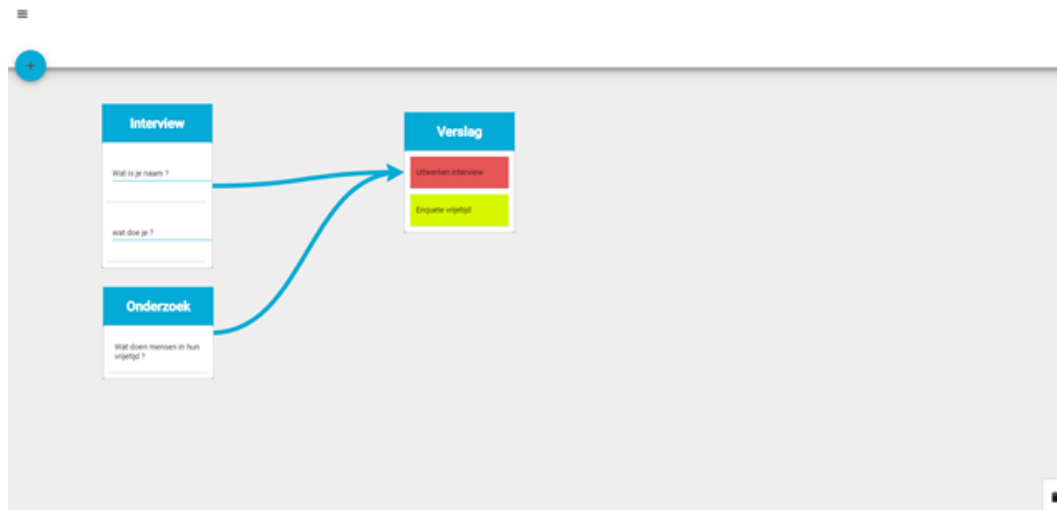
2.3 Beginsituatie

In een eerder project is er een eerste MEAN-stack (MongoDB, Express, Angular, NodeJS stack) prototype gemaakt die gebruikt zal worden door Saxion HBO-IT studenten in 2020-2021. Maar de functionaliteiten van dit prototype zijn erg beperkt.

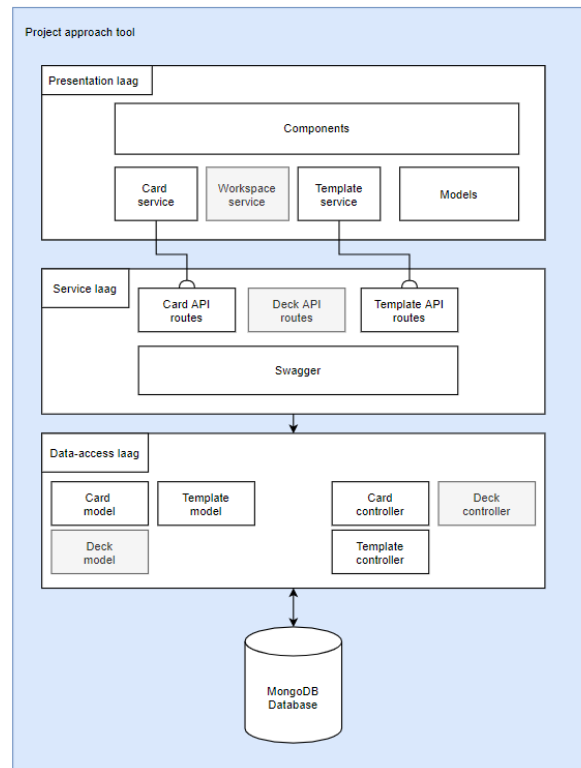
Nu kan een gebruiker dus één workspace aanmaken, dit is het bord waar de projectaanpak opgemaakt zal worden. Er zijn verschillende optionele templates beschikbaar om het makkelijker te maken voor de gebruikers, zodat ze kunnen beginnen met een goede basis.

Omdat de templates vaak niet voldoende zijn om de gehele projectaanpak in te plannen. Is het ook mogelijk voor gebruikers om aparte objecten toe te voegen zoals fases, pijlen, vragen en kaarten.

Hieronder is een afbeelding te zien van een voorbeeld projectaanpak. Er zijn in dit geval 3 fases genaamd interview, onderzoek en verslag. Dit zijn de lijsten met een blauwe header waar de titel in staat. In de verslag fase zie je 2 verschillende kaarten met verschillende kleuren waar een onderzoeksmethode (bijvoorbeeld brainstormen, requirements, prototype) opstaat en in de andere 2 objecten staan een andere vragen. Deze objecten zijn verbonden door middel van 2 pijlen, zodat een volgorde aangegeven kan worden.



Afbeelding 1 Voorbeeld workspace

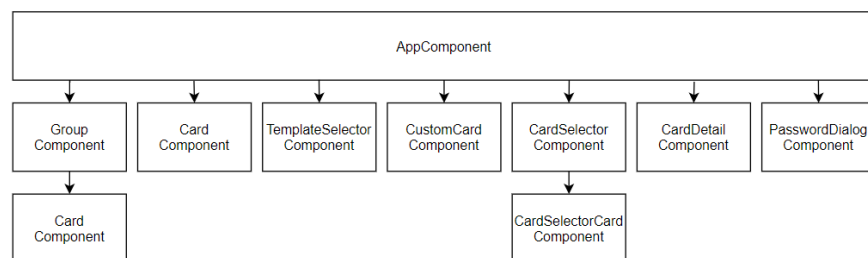


Afbeelding 2 Software architectuur beginsituatie

In afbeelding 2 wordt de software architectuur van de beginsituatie getoond. De onderdelen met een grijze kleur, zijn onderdelen waar uiteindelijk geen gebruik van is gemaakt. Momenteel worden de kaarten en templates opgeslagen in de database. De workspace wordt niet opgeslagen in de database, maar wordt opgeslagen in de LocalStorage. Dit houdt in dat het opgeslagen wordt op de computer van de gebruiker.

Presentation laag

Om aan te tonen welke componenten er zijn in de angular applicatie, is de volgende componenten hierachy overview aangemaakt:



Afbeelding 3 component hierachy overview beginsituatie

De AppComponent is de root component van de applicatie en omdat de applicatie maar een pagina bevat, wordt de root component gebruikt. De andere componenten worden gebruikt binnen de AppComponent.

Verder bevat de presentation laag ook nog services, met deze services kunnen er requesten naar de routes in de service laag gestuurd worden. Ook zijn er nog models aanwezig, deze models worden gebruikt om de typing te hanteren in typescript en ze staan gelijk aan de models in de data-access laag.

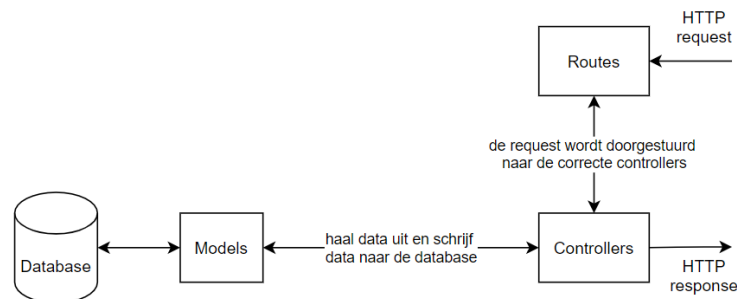
Service laag

De service laag bestaat merendeels uit nodeJS routes, Routes worden gebruikt om requesten door te sturen naar de bijbehorende controllers, controllers worden beter uitgelegd in de volgende laag.

Verder bevindt zich nog swagger functionaliteit in de service layer, hiermee wordt er OpenAPI 2.0 specificatie gegenereerd. Deze specificaties worden gebruikt om API's te beschrijven.

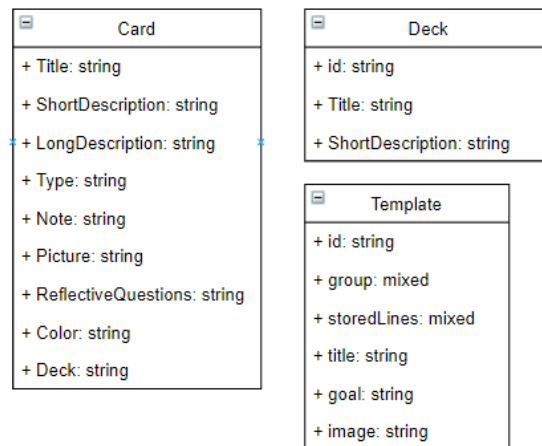
Data-access laag

De data-access laag, is de verbinding tussen de mongoDB database en service laag. De routes uit de service laag zullen requesten doorsturen naar de controllers in de data-access laag, zodat deze requesten behandeld kunnen worden en zodat er een response teruggestuurd kan worden.



Afbeelding 4 data flow backend

In Afbeelding 4 wordt uitgelegd hoe de controller te werk gaat, ook is in dezelfde afbeelding te zien dat de models gebruikt worden om data uit de database te halen. Hieronder zal meer informatie over de models te vinden zijn, in de vorm van een database diagram.



Afbeelding 5 Database diagram beginsituatie

Er zijn 3 models aanwezig, deze models hebben geen relatie met elkaar. De card model wordt gebruikt om alle research methodes cards en stepping stones op te slaan in de database. Het deck model wordt nog niet gebruikt. Het template model wordt momenteel gebruikt om templates te maken van bestaan workspaces.

2.3.1 Analyse

Voordat er begonnen is met programmeren, is er een software analyse uitgevoerd. Deze analyse is uitgevoerd zodat ik erachter kom waar nog problemen zitten en hoe de huidige situatie in elkaar zit.

Tijdens het analyseren werden er in totaal 28 problemen gevonden, deze heb ik opgesplitst in 4 categorieën, de grootte van een bug heb ik bepaald door te kijken naar hoe makkelijk het op te lossen lijkt:

- De grote bugs zullen veel tijd kosten omdat er onderzoek gedaan moet worden en/of veel functionaliteit aangepast moet worden.
- De middelmatige bugs, zijn bugs die vaak makkelijk te googlen zijn maar ook wel tijd gaan kosten om te repareren.
- De kleine bugs zijn vaak veroorzaakt door de grote/middelmatige bugs of makkelijk te repareren.
- De styling fouten zijn fouten die op te lossen zijn door de CSS of HTML aan te passen.

Categorie	Aantal gevonden
Grote bugs	1
Middelmatige bugs	2
Kleine bugs	15
Styling fouten	9

Tabel 1 Gevonden problemen

De grootste bug is eigenlijk de keuze voor een library. Er is namelijk gekozen om de library Leaderline te gebruiken voor het maken van pijlen. Deze library is een javascript library en geen typescript library. Vaak gaat het wel goed als je javascript en typescript combineert. Maar dit geldt niet voor de front-end.

Het is namelijk zo dat Angular een HTML-object aanmaakt in de body genaamd app-root waar in alle typescript pagina's geladen worden en dat de leaderline library de pijlen aanmaakt in de body. Dit zorgt ervoor dat de pijlen over de app-root komen en niet in de app-root en dit zorgt voor veel problemen, In Bijlage Requirements.

Verder zijn er nog een aantal zaken die opvielen tijdens het analyseren van de applicatie:

- Er is een typescript-linter geïntialiseerd maar deze is niet gebruikt. Dit houdt in dat de code niet aan de geconfigureerde conventies voldeed en ook niet consistent was.
- Er was javascript geschreven in plaats van typescript. Een voorbeeld is het te veel gebruiken van JSON-objecten in plaats van de typing die typescript biedt.
- Er was geen commentaar geschreven. Dit zorgde ervoor dat de code moeilijke te begrijpen was en dat het analyseren erg veel tijd kostte.
- Er was geen camel case of andere case style gebruikt. Dit houdt in dat de naamgeving van object, functies, attributen, etc. niet consistent was. Dit is ook te zien in Afbeelding 5, de attributen van de card model beginnen met een hoofdletter en die van template niet.
- Momenteel kan er maar een workspace aangemaakt worden en deze wordt opgeslagen in de localStorage.
- Er waren een aantal onderdelen van de applicatie die niet gebruikt werden zoals de workspace service en de deck routes, controller en model.
- Er was geen data loading aanwezig voor de kaarten of templates, dus toen ik een nieuwe instantie aanmaakte van de project approach tool met een andere database. Was de tool niet bruikbaar omdat er geen kaarten of templates waren.
- Er is geen rekening gehouden met toekomstige in-app navigatie, er is namelijk geen routing file aangemaakt.

Conclusie:

Het is mijn mening dat de developer van de beginsituatie geen rekening gehouden heeft met toekomstige developers die het project van hem overnemen, door te weinig commentaar te schrijven, geen linters te gebruiken en geen gebruik te maken van case styles was het erg lastig de applicatie te analyseren. Ook valt op dat er te veel fouten in het product zitten voor de omvang van het opgeleverde werk en dat er weinig gebruik is gemaakt van typing, daarom kom ik tot de conclusie dat de kwaliteit van de beginsituatie erg matig is.

Ook zal dit veel effect hebben op het project, omdat het tijd zal kosten om de code te refactoren. Voor dit project zijn er ook extra kwaliteitseisen toegevoegd, hierover zal meer informatie te vinden zijn in het hoofdstuk Kwaliteitscontrole.

2.4 Gewenste situatie

Samenwerking

De gewenste situatie is dat studenten door middel van de applicatie online samen een goed project kunnen opzetten en uitvoeren. Momenteel zijn de functionaliteiten er al, zodat één student een goed project op kan zetten maar studenten kunnen nog niet samenwerken.

Het is namelijk belangrijk dat studenten samen aan een project kunnen werken, omdat ze dan makkelijk samen aan een projectaanpak kunnen werken, het werk kunnen verdelen en elkaar ook goed kunnen helpen met het maken van de projectaanpak, ook is het belangrijk dat iedereen terug kan kijken naar de projectaanpak tijdens het project, zodat ze kunnen zien wat er nog moet gebeuren en eventueel gedurende het project, de projectaanpak kunnen uitbreiden of aanpassen.

Om deze situatie te bereiken zal het platform uitgebreid moeten worden, zodat een of meerdere workspaces gedeeld kunnen worden met meerdere studenten. Hiervoor zal de applicatie deels aangepast en uitgebreid moeten worden.

Maar tijdens het veranderen van de applicatie zal er ook nagedacht moeten worden over de beveiliging van de applicatie en de toegang tot de applicatie. Want momenteel is de applicatie toegankelijk voor iedereen en zit er nog geen login optie in de applicatie. Daarom zal er een kleine requirements analyse plaatsvinden en zal er mock-ups ontwikkeld worden om dit in kaart te brengen.

Onboarding

Om ervoor te zorgen dat gebruikers zelfstandig kunnen werken. Is er gekozen om applicatie onboarding te implementeren in het portaal, zodat studenten goed doorhebben hoe de applicatie werkt. Applicatie onboarding is een proces waarbij de gebruiker begeleid wordt door de gehele applicatie als het de eerste keer is dat de gebruiker de applicatie gebruikt, dit verhoogd namelijk de kans dat gebruikers de applicatie beter begrijpen en de kans dat ze de applicatie daarom ook eerder zullen gebruiken.

Dit zal bovendien wel eerst onderzocht en met gebruikerstesten getest moeten worden, zodat we weten hoeveel invloed het proces heeft en wat er nog aangepast kan worden, zodat de gebruikers het platform begrijpen.

2.5 Onderzoeksvragen

Door de gewenste situatie is duidelijk geworden wat de hoofdvraag is, namelijk “Hoe kan de applicatie worden uitgebreid, zodat gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent?” Deze hoofdvraag kan opgesplitst worden in de volgende deelvragen:

1. Hoe werken de DOT-framework, ICT-research methods kaarten en Miriam Losse's Stepping stones methodes?
2. Wat zijn standaarden en best practices voor real time collaboration, zodat wijzigingen onmiddellijk worden bijgewerkt en inconsistenties op een voor de gebruiker intuïtieve manier worden afgehandeld?
3. Welke libraries kunnen helpen met het consistent houden van de gedeelde workspaces en wanneer er een aanpassingen gedaan worden?
4. Welke visuele effecten kunnen we gebruiken om aan te tonen welke gebruikers verschillende aanpassingen aan het doen zijn? (Typen, slepen, zien wie aanwezig is)
5. Welke vorm van app onboarding is het meest geschikt voor het platform? (Benefit, Features, Action-oriented, Account, Interactive, Combination)
6. Welke informatie heeft de gebruiker nodig in de onboarding, zodat hij zelfstandig gebruik kan maken van de applicatie?

2.6 Interne stakeholders

Stakeholder	Beschrijving
Lars Hendriks	Dit ben ik zelf, ik ben de ontwikkelaar van de applicatie.
Danny Plass	Dit is de eigenaar van het product en ze is ook bedrijfsbegeleider
Saxion	Saxion als een instelling is geïnteresseerd in de applicatie en ze verwachten een applicatie die aan studenten aanbevolen kan worden. Saxion is ook de schoolinstelling waarbij ik een opleiding volg.
Studenten	Dit zijn de hoofdgebruikers van de applicatie.
Etto Salomons	Dit is de vertegenwoordiger van HBO-ICT 3S.
Miriam Losse	Dit is de ontwikkelaar van de stepping stone kaartenset en ze definieert de visie op onderzoek bij het Saxion en sterk betrokken bij 3S.
Koen van Turnhout	Dit is de co-ontwikkelaar van het DOT-framework en ICT research cards set.
Sonja Rouwhorst	Dit is een docent van hogeschool van Amsterdam en ze maken daar een soortgelijke product.
Kasper Kamperman	Dit is de vertegenwoordiger van CMGT (Saxion).
HBO-I	Dit is een koepelvereniging voor HBO ICT opleidingen.
Anthony van den berg	Dit is de begeleidende docent voor mijn afstudeerperiode.

Tabel 2 stakeholders beschrijving

2.6.1 Analyse

		Belang			
		Laag	Matig	Hoog	Zeer hoog
Invloed	Zeer hoog	Miriam Losse Koen van Turnhout Anthony van den berg		Lars Hendriks Danny Plass	
	Hoog				
	Matig	Kasper Kamperman Sonja Rouwhorst HBO-I		Saxion Etto Salomons	Studenten
	Laag				

Tabel 3 Stakeholder analyse

Kleur	Beschrijving
	Beïnvloeder (veel invloed, weinig belang)
	Belangrijkste spelers (veel invloed, veel belang)
	Toeschouwers (Weinig invloed, Weinig belang)
	Belanghebbende (Weinig invloed, Veel belang)

Tabel 4 Legenda stakeholder analyse

2.6.2 Betrokkenheid

Stakeholder	Hoe worden ze bij het project betrokken ?
Danny Plass	<p>Elke week worden er 2 meetings gepland om deze stakeholder op de hoogte te houden van de stand van zaken.</p> <p>Deze stakeholder neemt ook deelname aan de 3 stakeholder meetings die gedurende de afstudeeropdracht gepland zijn.</p>
Saxion	Saxion wordt indirect op de hoogte gehouden door meerdere stakeholders van het Saxion bij het project te betrekken.
Studenten	De studenten worden op de hoogte te houden van de stand van zaken door verschillende gebruikerstesten uit te voeren om feedback te krijgen op het platform.
Etto Salomons	Er zijn 3 stakeholders meetings gepland gedurende de afstudeeropdracht om de stakeholders op de hoogte te houden van de stand van zaken.
Miriam Losse	Er zijn 3 stakeholders meetings gepland gedurende de afstudeeropdracht om de stakeholders op de hoogte te houden van de stand van zaken.
Koen van Turnhout	Er zijn 3 stakeholders meetings gepland gedurende de afstudeeropdracht om de stakeholders op de hoogte te houden van de stand van zaken.
Sonja Rouwhorst	Er zijn 3 stakeholders meetings gepland gedurende de afstudeeropdracht om de stakeholders op de hoogte te houden van de stand van zaken.
Kasper Kamperman	Er zijn 3 stakeholders meetings gepland gedurende de afstudeeropdracht om de stakeholders op de hoogte te houden van de stand van zaken.

HBO-I	HBO-I wordt indirect op de hoogte gehouden door meerdere stakeholders van het HBO-I bij het project te betrekken.
Anthony van den Berg	Elke 2 weken wordt er een meeting gepland om deze stakeholder op de hoogte te houden van de stand van zaken.

Tabel 5 Beschrijving betrokkenheid stakeholders

3 Aanpak

Hieronder zal informatie te vinden zijn, hoe er te werk ben gegaan en welke processen er gebruikt zijn tijdens het werken.

3.1 Communicatie

Door coronamaatregelen zijn fysieke meetings niet mogelijk. Daarom zijn er alleen maar online meetings gehouden. Er was afgesproken om elke week een meeting te hebben met de bedrijf begeleider/product owner (in dit geval dezelfde persoon) en als het nodig blijkt te zijn, kunnen er altijd extra meetings ingepland worden.

In deze gesprekken wordt vooral besproken wat er is gemaakt, wat er verbeterd moet worden en wat er de volgende week gedaan zal worden.

3.2 Scrum

3.2.1 Daily standup

Omdat het projectgroep maar uit 1 lid bestaat, was het in begin toch moeilijk om geen daily standup te kunnen doen. Daarom dat er na een paar weken ervoor gekozen is om dit met mezelf te doen. Elke dag voordat ik begon met programmeren, heb ik mezelf afgevraagd wat wil ik vandaag bereiken? En dat doel is op papier gezet, zodat ik mezelf er ook zoveel mogelijk aan probeer te houden.

In het begin was het even inschatting wat ik aan kan in een dag, maar naar een aantal weken ging het veel beter en het zetten van een doel heeft me goed geholpen, omdat gefocust te blijven.

3.2.2 Workflow

Om een issue af te kunnen handelen, zijn er een aantal stappen in het scrumbord aangemaakt:

1. Backlog

Hier staan alle issues die bekenden zijn, geprioriteerd via de MoSCoW manier. De prioritering is in overleg met de product owner afgesproken.

2. To Do

Als er in een meeting wordt besloten om issues op te pakken voor de aankomende sprint dan worden ze in deze lijst geplaatst. Mochten issues overblijven dan zullen deze meegenomen worden naar de volgende sprint. Het is ook voorgekomen dat alles issues af waren voor de nieuwe meeting, toen is er begonnen aan de eerstvolgende issue op de geprioriteerde lijst.

3. In Progress

In deze stappen, staan de issues waar momenteel aan gewerkt wordt.

4. To Test

Hier worden alle issues geplaatst, die getest moeten worden.

5. To Review

In de review stap worden de issues besproken met de product owner en wordt feedback verwerkt, zo werd de definition of done bereikt.

6. Done

Alle issues die afgemaakt zijn, komen hier te staan.

In het hoofdstuk Tools zal meer informatie te vinden zijn over welke tool zijn gebruikt om het scrumbord te maken.

3.3 Kwaliteitscontrole

Er is afgesproken dat de kwaliteit van het product zoveel mogelijk zelf bewaakt wordt. De kwaliteit van het project werd op de volgende manier gecontroleerd en gehanteerd:

Was het gebruik van linting tools, dit is een tool die waarschuwing laat zien als code niet voldoet aan de geconfigureerde conventies. Voor Angular wordt er gebruik gemaakt van TSLint, dit is een linter voor typescript. Voor de node back is er gebruik gemaakt van ESLint, dit is een linter voor javascript/nodeJS.

Ook wordt camel casing gebruikt voor alle case style voor dit project en zal er commentaar geschreven worden om code duidelijker te maken.

Verder zijn er ook meerdere componenten en onderdelen getest van de applicatie. Hierover zal meer informatie te vinden zijn in hoofdstuk Load testen en Gebruikerstest.

3.4 Tools

Tijdens het ontwikkelen van de applicatie zijn er verschillende tools gebruikt, die hieronder beschreven zullen worden:

- **GitHub**
Git is gebruikt voor het versiebeheer van de code. De code zelf wordt opgeslagen in GitHub. Er werd ook gebruikt gemaakt van het projectbord. Dit werd ingericht als scrumbord.
- **Microsoft Teams**
Microsoft Teams werd gebruikt als primaire communicatiemiddel. Hiermee werden alle online meetings gehouden.
- **Visual studio code**
Visual studio code werd gebruikt als IDE.

4 Requirements

De requirements zijn een resultaat uit een gesprek met de product owner danny plass en een gesprek met de stakeholders.

Voor het gesprek met de product owner, is er een brainstormsessie gehouden en hieruit kwam een lijst van alle requirements die bedacht werden. Deze requirements zijn toen besproken met de product owner en toen is de lijst aangepast en uitgebreid zodat de lijst compleet was.

Na het uitbreiden van de lijst hebben we de lijst geprioriteerd op wat het meest belangrijk was voor de product owner.

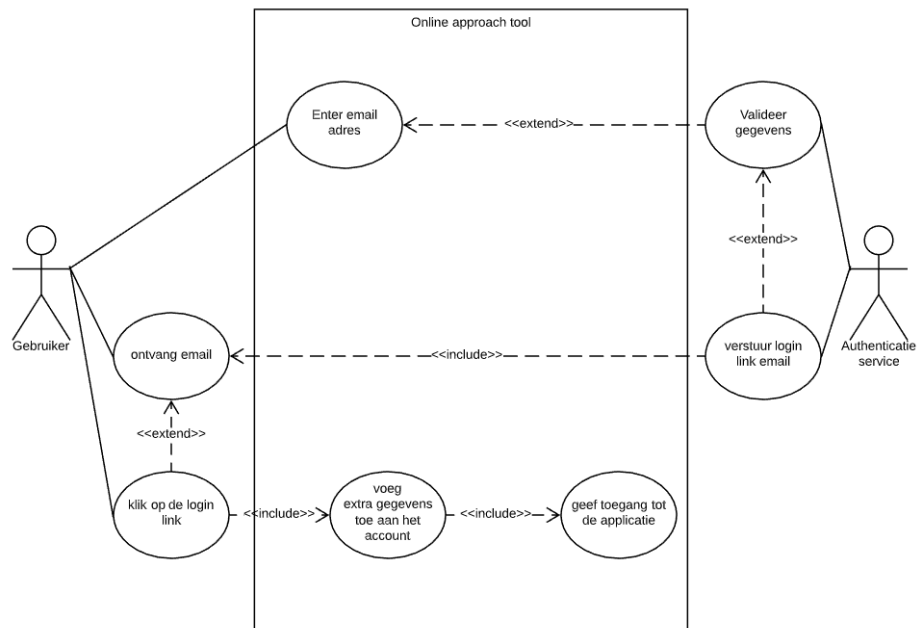
Vervolgens hebben we een afspraak gemaakt met de stakeholders. Voor deze meeting is er een kleine presentatie voorbereid zodat er feedback verzameld kon worden op de huidige requirements en op de prioritering van de high level requirements (de onderwerpen die bij de nummering beschreven staan).

Hiermee kan er namelijk een schatting mee gemaakt worden over wat welke stakeholder het meest belangrijk vond. De requirements zijn geprioriteerd volgens de MoSCoW manier. De requirements zijn te vinden in de bijlage Requirements.

5 Authenticatie

Dit hoofdstuk zal vooral ingaan op online collaboratie binnen de applicatie en de keuzes die gemaakt zijn. Voordat er begonnen is met het onderzoeken van de authenticatie, gaf de product owner aan dat ze graag passwordless-login wou, dit betekent dat er ingelogd kan worden zonder wachtwoord en dat je alleen een email account nodig hebt. De volgende hoofdstukken zullen gebaseerd zijn op de wens van de product owner.

5.1 Visie



Afbeelding 6 Use case diagram authenticatie

In afbeelding 6 wordt het authenticatie proces beschreven. Omdat de wens van de product owner is dat er passwordless login geïmplementeerd wordt. Is het login/registratie proces gecombineerd tot één proces.

Een gebruiker geeft een email adres op, dit email adres wordt gevalideerd bij de authenticatie service, mocht het account nog niet bestaan dan wordt er een aangemaakt en er wordt een login email verstuurd naar het opgegeven email adres.

De gebruiker ontvangt de email en klikt op de login link die zich bevindt in de login email, de gebruiker wordt dan ingelogd en gestuurd naar applicatie. Als het een nieuwe account is, zal er een pop up komen om te vragen voor extra gegevens die benodigd zijn voor de applicatie zoals voornaam, achternaam, etc. Als de gebruiker klaar is met het opgeven van de extra gegevens dan zal de gebruiker toegang krijgen tot de applicatie. Gebruikers die bekend zijn bij het systeem zullen gelijk toegang krijgen tot de applicatie, er worden namelijk gegevens opgeslagen door de firebase authenticatie service in de indexedDB waarmee gekeken wordt of een gebruiker ingelogd is.

5.2 Authenticatie service

Hieronder zal meer informatie te vinden zijn over de keuze die gemaakt is voor een authenticatie service. De keuzeverantwoording bestaat uit een tabel waarin gewenste functionaliteiten worden en een eindconclusie.

5.2.1 Vergelijking

Hieronder zal een vergelijking te vinden zijn van verschillende authenticatie services. Er zijn veel authenticatie services en hieruit zijn er een paar gekozen om te vergelijken, ze zijn gekozen omdat ze

een gratis instap prijs hebben en omdat ze een hoge score hebben op de G2, hieronder zijn deze onderdelen met een asterisk aangegeven.

	AWS cognito	Auth0	Okta	Firebase authentication
Instap prijs	Gratis	Gratis	Gratis	Gratis
Aantal gratis actieve gebruikers	50.000	7.000	1.000	onbeperkt
Mogelijkheid tot passwordless login	Onofficieel ondersteund	Officieel ondersteund (gratis)	Officieel ondersteund (maar alleen voor betalende klanten)	Officieel ondersteund (gratis)
Makkelijk te gebruiken *	7.1	8.5	9.2	9.0
Makkelijk op te zetten *	8.3	8.2	8.7	9.0
Makkelijk te onderhouden *	8.3	8.6	9.1	9.0
Kwaliteit van de ondersteuning *	6.9	8.6	8.6	8.1
Productrichting *	6.9	8.0	8.2	8.4

Tabel 6 Vergelijking authenticatie services

Alle onderdelen met een asterisk zijn cijfers die verzameld zijn van een website genaamd g2. Dit is een website waar IT-medewerkers, services zoals de authenticatie services die hierboven in de tabel te vinden zijn kunnen beoordelen. Deze waardes zijn verzameld op 02-06-2020 en kunnen afwijken van de huidige waardes die op de website te zien zijn.

Hieronder zal een beschrijving van de onderdelen te vinden zijn.

- **Instap prijs**
Dit is een tarief die betaald moeten worden als je de dienst af wilt nemen.
- **Aantal gratis actieve gebruikers**
Dit is een aantal actieve gebruikers per maand in mogen loggen.
- **Mogelijkheid tot passwordless login**
Hier wordt er gekeken of de authenticatie service passwordless login mogelijk maakt en ondersteunt.
- **Makkelijk te gebruiken**
Dit is een cijfer dat weergeeft hoe makkelijk het te gebruiken is.
- **Makkelijk op te zetten**
Dit is een cijfer dat weergeeft hoe makkelijk het op te zetten is.
- **Makkelijk te onderhouden**
Dit is een cijfer dat weergeeft hoe makkelijk het te onderhouden is.
- **Kwaliteit van de ondersteuning**
Dit is een cijfer dat de kwaliteit van de ondersteuning weergeeft.

- **Productrichting**

Dit is een cijfer dat de productrichting weergeeft.

Conclusie:

Bij het kiezen van de authenticatie service, is er vooral gekeken naar de actieve gebruikers en mogelijkheid tot passwordless login als grote onderdelen die goed moeten zijn. Want zeventuizend gebruikers of minder is veel te weinig voor het Saxion als het gaat om actieve gebruikers. Het is nog niet zeker hoeveel scholen gebruik zullen maken van de applicatie, vandaar dat een groot aantal gebruikers wenselijk is, daarom vallen Auth0 en Okta af.

Nu gaat het nog tussen AWS cognito en firebase authentication, er is gekeken naar de onderdelen met een asterisk en firebase heeft in alle onderdelen een hoger score, dit houdt in dat het makkelijkere werken is met firebase authentication.

Ook is er gekeken naar een wens van de product owner en dat is passwordless login, dit is wel mogelijk met AWS cognito maar wordt niet officieel ondersteunt, firebase authentication officieel ondersteunt passwordless login wel.

Omdat firebase authentication hoger scoort op de onderdelen met een asterisk en omdat het passwordless login officieel ondersteunt, is ervoor gekozen om firebase authentication te gebruiken.

6 Online collaboratie

Het is belangrijk voor gebruikers dat ze samen aan een project approach kunnen werken, online collaboratie maakt dit voor de gebruikers mogelijk. Dit hoofdstuk zal ingaan op online collaboratie binnen de applicatie en alle zaken die hierbij komen kijken.

6.1 Realtime connectie opties

In dit hoofdstuk wordt er gekeken naar welke realtime connectie opties er zijn, uit dit onderzoek kwam de volgende lijst met opties (Basu, 2019):

- **Polling**

Is een techniek waarbij de cliënt, de server regelmatig vraagt of er nieuwe data is. Ook al is er geen nieuwe data bij de server, dan zal er nog steeds gevraagd worden.

- **Long-polling**

Is een verbeterde polling techniek, waarbij de cliënt vraagt of er nieuwe data is en de server net zo lang wacht met een response sturen, totdat er nieuwe data is. Als de data gestuurd wordt, wordt er automatisch een nieuwe request gestuurd.

- **Server Sent Events(SSE)**

Dit is een grotendeels een eenrichting http-connectie van de server naar de cliënt. Dit wordt ook wel een simplex connectie genoemd.

- **Forever frame**

Is een techniek waarbij een iframe, dat een request stuurt naar een server die niet afgemaakt wordt, de server blijft dan scripts sturen naar de cliënt, waardoor er een realtime connectie ontstaat. Dit houdt in dat een request een eindeloos response heeft.

- **Websockets**

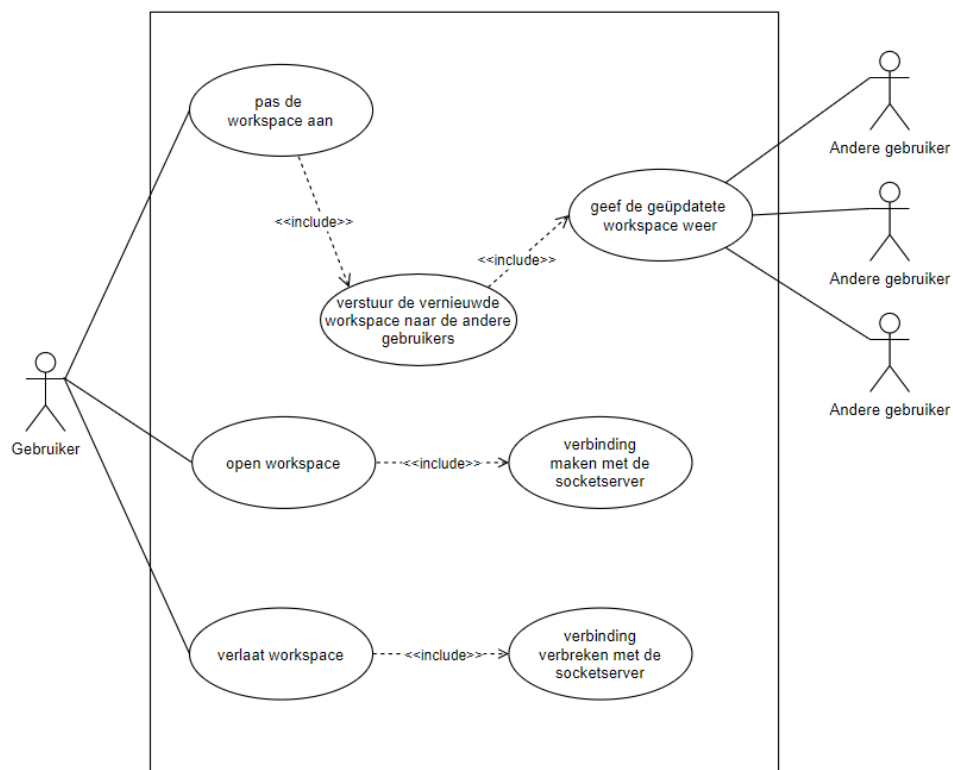
Dit is een techniek waarbij een persistente connectie tot stand wordt gebracht, waarbij de cliënt naar server kan communiceren en tegelijkertijd de server ook naar de cliënt kan communiceren, dit heet een full-duplex connectie.

Conclusie:

Als er in de workspace aanpassingen worden gedaan dan zullen deze gestuurd moeten worden van de gebruiker via de server naar andere gebruikers in dezelfde workspace, dit is een full/half-duplex connectie. Daarom lijkt Websockets mij de beste keuze, dit maakt namelijk gebruik van een full-duplex connectie en is daarom het meest effectief.

6.2 Concept

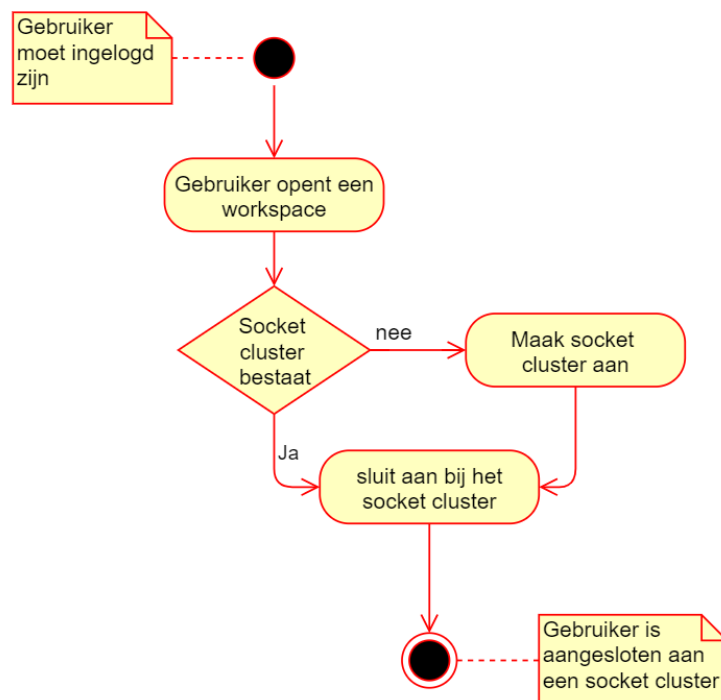
In dit hoofdstuk zal het concept voor online collaboratie uitgelegd worden.



Afbeelding 7 Use case diagram online collaboratie

Om meer overzicht te creëren is er een use-case diagram gemaakt die uitlegt wat de stappen zijn. Want het is namelijk de bedoeling dat alle aanpassingen die gemaakt worden door een gebruiker gestuurd zal worden naar alle andere gebruikers die actief zijn in de workspace.

Om dit te wezelijken zal er een realtime connectie gemaakt moeten worden tussen de gebruikers en de server, zodat als een gebruiker een aanpassing maakt dit via de server naar de andere gebruikers gestuurd kan worden. Hieronder zal informatie te vinden zijn over hoe de realtime connectie tot stand gebracht zal worden.



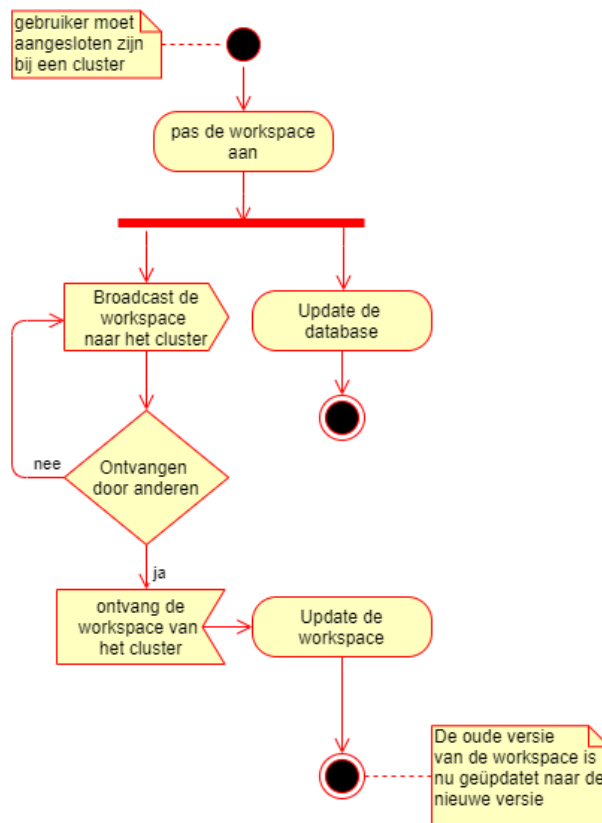
Afbeelding 8 Activity diagram workspace joinen

Als een gebruiker een workspace opent zal er gekeken worden als een socket cluster bestaat, sockets worden gebruikt om de realtime connectie tot stand te brengen. Een socket cluster is in dit geval een groepering van realtime connecties.

Mocht het cluster nog niet bestaan dan zal die aangemaakt worden en zal er een connectie toegevoegd worden aan het cluster. Mocht het cluster wel bestaan dan zal de connectie toegevoegd worden aan het bestaande cluster.

Zodra de gebruiker is aangesloten aan de socket cluster, dan kan de gebruiker aanpassingen maken aan de workspace. Hieronder zal meer informatie te vinden zijn over hoe dit is werking zal gaan. Mocht het niet lukken voor een gebruiker om zich aan te sluiten bij een cluster dan wordt dit meerdere keren

geprobeerd. Mocht dan nog geen connectie tot stand zijn gekomen, dan worden de aanpassingen alleen in de database opgeslagen.

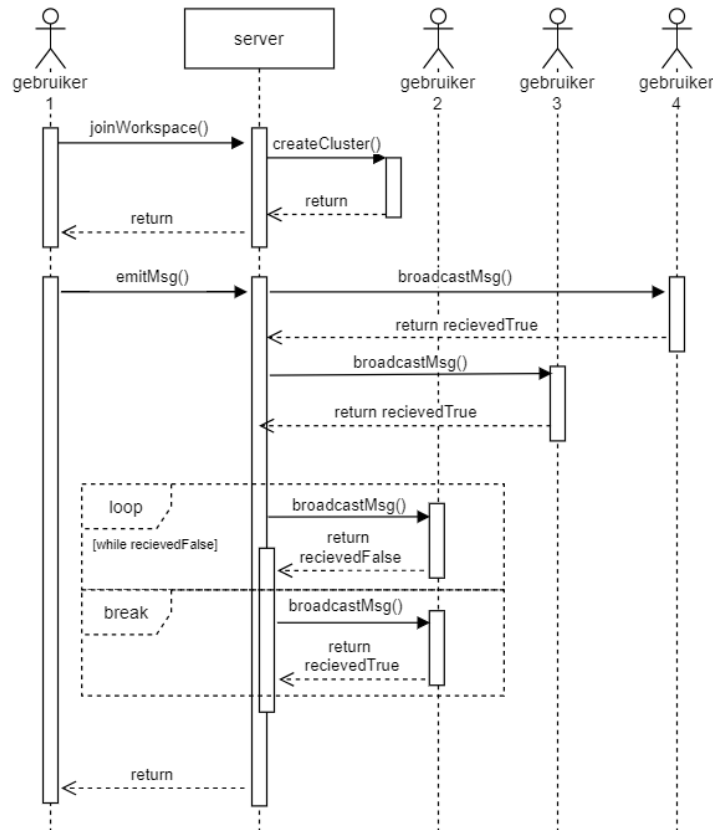


Afbeelding 9 Activity diagram workspace aanpassen

Nu dat de gebruiker is aangesloten bij een cluster, kan de gebruiker de workspace aanpassen of uitbreiden. Nadat de gebruiker dit gedaan heeft zullen er 2 synchrone taken uitgevoerd worden:

- De database zal geüpdatet worden, zodat de data synchroon is in de backend met de data in de front-end.
- Ook zal de nieuwe versie van de workspace gebroadcast worden via de socketcluster naar de andere gebruikers. Mocht de data door de andere gebruikers niet goed verzonden worden, dan zal de data verzonden worden zodra de connectie weer goed is. Na het ontvangen van de nieuwe versie zal de workspace dat op het scherm te zien is geüpdatet worden.

Hieronder zal meer informatie te vinden zijn over de broadcasting.

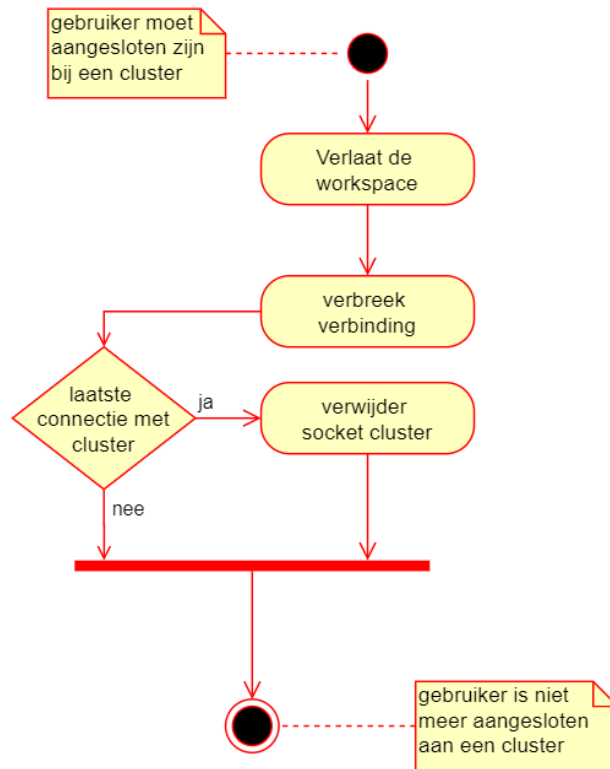


Afbeelding 10 Sequence diagram workspace aanpassen

Als scenario is er gekozen voor een projectgroep van 4 personen, om goed uit te kunnen beelden hoe het broadcasten zal werken. Gebruiker opent een workspace en een socketcluster wordt aangemaakt. Zodra het socketcluster is aangemaakt wordt er ook een realtime connectie opgezet met de backend. Nu kan de gebruiker dus aanpassing maken aan de workspace. Zodra de gebruiker dit doet zal de workspace geüpdatet worden en zal er een emit uitgevoerd worden. Dit houdt in dat de workspace naar alle mensen in het cluster gestuurd zal worden.

Zoals in Afbeelding 10 te zien is gaat de broadcast goed bij gebruiker 2 & 3, bij deze gebruikers wordt de workspace nu geüpdatet. Helaas gaat het bij gebruiker 4 fout en nu wordt er een seintje gegeven aan de server dat de broadcast niet goed gegaan is en nu wordt de broadcast voor die gebruiker herhaalt totdat het gelukt is, nu wordt bij deze gebruiker ook de workspace geüpdatet. Dit proces wordt gebruikt om de workspace voor alle gebruikers in het cluster consistent te houden.

Hieronder zal meer informatie te vinden zijn over het verlaten van een workspace.



Afbeelding 11 Activity diagram workspace verlaten

En als laatste het verlaten van een workspace, dit kan zijn in verschillende vormen zoals uitloggen, of van pagina wisselen of het verlaten van de applicatie. Maar als een gebruiker een van deze vormen gebruikt om de workspace te verlaten dan zal de realtime connectie verbroken worden, mocht dit de laatste connectie zijn met het cluster dan zal het cluster ook verwijderd worden.

6.3 Socket library

Hieronder zal meer informatie te vinden zijn over de keuze die gemaakt is voor een socket library. De keuzeverantwoording bestaat uit een tabel waarin gewenste functionaliteiten worden vergeleken en een benchmark van de library die uit de tabel het beste naar boven komt. Hieronder zal eerst uitgelegd worden welke gewenste functionaliteiten er vergeleken worden en waarom deze nodig zijn.

- **Websockets**
Het gebruik van websockets is verplicht zodat er een full-duplex connectie opgezet kan worden.
- **Clustering**
Dit is een techniek waarbij sockets gegroepeerd worden in een cluster, dit is nodig om meerdere sockets op een makkelijk manier met elkaar te kunnen verbinden.
- **Broadcasting to cluster**
Hiermee kan een bericht naar alle sockets in een cluster gestuurd worden en dit is nodig om de workspaces voor meerdere gebruikers synchroon te houden.

- **Lifecycle callbacks**
Dit zijn callback functies die vast zitten aan de lifecycle van een verbinding en hiermee kan er gedetecteerd worden wanneer een socket zich verbindt of sluit. Dit is nodig om te detecteren wanneer een gebruiker de workspace verlaat of opent.
- **Fallback opties**
Dit is een systeem dat geïmplementeerd is om automatische terug te vallen naar een andere transport type zoals long-polling, als een websocket connectie niet mogelijk blijkt te zijn.
- **Load balancing**
Is een techniek waarbij connecties verdeeld worden over meerdere servers. Dit is meegenomen in de vergelijking voor een toekomstige uitbreiding.
- **Debugging**
Hiermee kunnen problemen makkelijk gevonden worden, dit is toegevoegd als gewenste functionaliteit om het development proces te versnellen.

Hieronder zullen ook nog 4 best practices (Vallet, 2020) beschreven staan, die mee worden genomen in de vergelijking:

- **Reconnecting**
Hiermee wordt opnieuw verbinding gemaakt wanneer een verbinding verbroken is. Dit is nodig om een socket verbonden te houden.
- **Clean disconnect**
Mocht het echt niet mogelijk zijn om verbinding te maken met de socket server, dan moet hier op een nette manier mee omgegaan worden.
- **Refreshing**
Als een verbinding tussen de cliënt en de server al een lange tijd bestaat, dan is het handig om de verbinding te verversen.
- **Message queue**
De message queue zorgt ervoor dat berichten op een nette manier achter elkaar verstuurd worden, dit is een van de technieken die gebruikt worden om synchronisatie problemen te voorkomen.

6.3.1 Vergelijking

Gewenste functionaliteit	WS	Socket.io	SockJS	Socket Cluster
Websockets	Ja	Ja	Ja	Ja
Clustering	Nee	Ja	Nee	Ja
Broadcasting to Cluster	Nee	Ja	Nee	Ja
Lifecycle callbacks	Ja	Ja	Ja	Ja
Fallback opties*	Nee	Ja	Ja	Nee
Load balancing*	Nee	Ja	Ja	Ja
Debugging*	Ja (Dev tools)	Ja	Ja (Dev tools)	Ja (Dev tools)
Reconnecting	Nee	Ja	Ja	Ja
Clean disconnect	Ja	Ja	Ja	Ja

Refreshing	Nee	Ja	Ja	Ja
Message queue	Nee	Ja	Ja	Ja

Tabel 7 Vergelijking socket library

*Notitie: alle gewenste functionaliteiten met een *, zijn optioneel.*

Conclusie vergelijking:

Socket.io beschikt over alle gewenste functionaliteiten en daarom zal er een benchmark uitgevoerd worden om te kijken of deze library, de gewenste performance behaald.

6.3.2 Benchmark

Om erachter te komen of socket.io de verwachte performance heeft, wordt de volgende scenario gebruikt om testen uit te voeren:

- Een projectgroep heeft gemiddeld 5 projectleden, dit is gebaseerd op hbo-ICT en 3S projecten.
- Een student heeft ongeveer 5 seconde nodig om een aanpassing te doen aan de werkplek, dit is een gemiddelde voor een actie (typen, slepen, objecten toevoegen, etc.). Dit is een schatting dat in overleg met de productowner voor is gekozen.
- Disclaimer: dit is een schatting en kan uiteindelijk afwijken van hoe het gebruikt zal worden.

Het scenario die hierboven beschreven wordt, zal gebruikt worden om de volgende testen uit te voeren:

- De testen worden uitgevoerd om te testen hoeveel actieve sockets de server aan kan en welke performance dit oplevert.
 - 100 sockets en alle clusters versturen een bericht
 - 1000 sockets en alle clusters versturen een bericht
 - 5000 sockets en alle clusters versturen een bericht
 - 10000 sockets en alle clusters versturen een bericht
- Deze testen worden uitgevoerd om te testen hoeveel inactieve sockets effect hebben op de performance van de server.
 - 10000 sockets en 10% van de clusters versturen een bericht
 - 10000 sockets en 25% van de clusters versturen een bericht
 - 10000 sockets en 50% van de clusters versturen een bericht
 - 10000 sockets en 75% van de clusters versturen een bericht

Tijdens deze testen wordt de volgende informatie verzameld elke seconde dat de benchmark test loopt:

- Aantal clusters
- Bericht ontvangen per seconde per cluster
- Bericht verstuurd per seconde per cluster
- Bericht ontvangen per seconde per socket
- Bericht verstuurd per seconde per socket
- CPU-gebruik (in %)
- Memory-gebruik (in %)

Het bericht wat verstuurd zal worden tijdens de test, is een 6KB JSON-object van een voorbeeld workspace, die gevuld is met 4 fases, 3 pijlen en 10 kaarten.

6.3.3 Server aanpassingen

Voordat er begonnen is met het uitvoeren van een benchmark test. Er is eerst eens gekeken naar de specificaties van de server en die zijn als volgt:

CPU	Intel Xeon Silver 4114s
GPU	VMware SVGA II Adapter
Cores	1 Single Core
RAM	1 GB

Tabel 8 Server voor de aanpassingen

Dit was op zich genoeg voor de oude versie waar op verder gewerkt wordt, maar in overleg met Etto Salamons (de beheerder van server) hebben we gekozen om de server als volgt uit te breiden, om de sockets beter te kunnen ondersteunen:

CPU	Intel Xeon Silver 4114s
GPU	VMware SVGA II Adapter
Cores	4 Single cores
RAM	8 GB

Tabel 9 Server na de aanpassingen

NodeJS is eigenlijk single-cored, maar omdat de mongodb database ook aanwezig is op deze server, is ervoor gekozen om meerdere cores toe te voegen. Het benchmark testen worden uitgevoerd op de productieserver (de hierboven gespecificeerde server), omdat hier de applicatie uiteindelijk op komt te staan.

6.3.4 Eindconclusie

Voordat het benchmark uitgevoerd werd, vond er een gesprek plaats met de productowner om een goal te zetten voor een minimaal viable product. Uiteindelijk kwam 10.000 actieve sockets naar boven als een aantal die wij aan zullen houden voor piekmomenten.

Een piekmoment is vaak de eerste en tweede week van een kwartiel, omdat in deze periode vaak aan de projectaanpak gewerkt wordt. Dit zal namelijk nog een mogelijkheid geven tot uitbreiden van de doelgroep, bijvoorbeeld door andere opleidingen die aansluiten bij het platform.

Uit het benchmark kwam naar boven dat 10.000 sockets wel zeker haalbaar is met de geteste library en in overleg met de product owner is de library ook goedgekeurd voor gebruik. Het resultaat van de benchmark is te vinden in de bijlage Benchmark resultaat.

Maar er zal zeker goed opgelet moeten worden naar het aantal sockets die actief zijn. Mocht het aantal actieve sockets groter worden dan 10000 dan zal de server zeker uitgebreid moeten worden. Want momenteel is de server verticaal geschaald om de test uit te voeren, dit houdt in dat de test plaats heeft gevonden op 1 server, maar socket.io ondersteunt ook load-balancing dus horizontaal schalen is mogelijk.

In het hoofdstuk aanbevelingen zal een aanbeveling te vinden zijn over het uitbreiden van de server en wat de beschikbare opties zijn. Het resultaat van de benchmark is te vinden in bijlage Benchmark resultaat.

6.4 Load testen

Nu dat het systeem gebouwd is, zal er hieronder aangetoond worden dat het goed geïmplementeerd is en dat het systeem een goede performance heeft. Door middel van load testen, dat zijn testen waarbij de gebruikersbelasting van een systeem getest wordt.

Voor de load testen is er gekeken naar welke test frameworks het beste waren om te gebruiken. Er zijn 2 verschillende types test tool waarvan gebruik gemaakt kan worden.

- Een test tool dat het web socket protocol ondersteunt, want socket.io maakt onder water gebruik van web sockets.
- Een test tool dat socket.io direct ondersteunt.

Maar omdat er gebruik gemaakt wordt van socket clusters, is het beter om gebruik te maken van een test tool dat socket.io direct ondersteunt, zodat de clusters makkelijk getest kunnen worden.

Momenteel is Artillery.io de enige loadtest tool, die socket.io direct ondersteunt, daarom is er gekozen om hier gebruik van te maken.

Er zijn van tevoren 2 requirements opgesteld voor de testen:

- De response-time van een request mag niet langer zijn dan 500ms
- Het faalpercentage van alle request mag niet meer zijn dan 1%

Deze requirements werden mij als advies mee gegeven, door een aantal leden van de artillery.io spectrum community, Spectrum is een applicatie waar vragen gesteld kunnen worden over een bepaald onderwerp, in dit geval artillery.io.

6.4.1 Test scenario's

De hieronder beschreven test scenario's zullen getest worden.

Test naam	Beschrijving
Join-1	1 gebruiker sluit zich aan bij het cluster en verlaat daarna het cluster.
Join-10	10 gebruikers sluiten zich aan bij het cluster en verlaten daarna het cluster.

Join-100	100 gebruikers sluiten zich aan bij het cluster en verlaten daarna het cluster.
Join-1000	1000 gebruikers sluiten zich aan bij het cluster en verlaten daarna het cluster.
move-1	1 gebruiker sluit zich aan bij het cluster ,verplaatst een fase object en verlaat daarna het cluster.
move-10	10 gebruikers sluiten zich aan bij het cluster, verplaatsen een fase object en verlaten daarna het cluster.
move-100	100 gebruikers sluiten zich aan bij het cluster, verplaatsen een fase object en verlaten daarna het cluster.
move-1000	1000 gebruikers sluiten zich aan bij het cluster en verplaatsen een fase object en verlaten daarna het cluster.
setEffect-2	2 gebruikers sluiten zich tegelijkertijd aan bij het cluster, zetten een kleur op een fase en verlaten daarna het cluster.
Wait-60	1 gebruiker sluit zich aan bij het cluster, wacht 1 minuut en verlaat daarna het cluster.
Wait-600	1 gebruiker sluit zich aan bij het cluster, wacht 10 minuten en verlaat daarna het cluster.
Wait-1800	1 gebruiker sluit zich aan bij het cluster, wacht 30 minuten en verlaat daarna het cluster.
Wait-3600	1 gebruiker sluit zich aan bij het cluster, wacht 1 uur en verlaat daarna het cluster.

Tabel 10 Beschrijving loadtest scenarios

- **Join tests**
Deze testen worden uitgevoerd om te testen of het systeem goed om gaat met de hoeveelheid mensen die zich aansluiten bij een cluster.
- **Move tests**
Deze testen worden uitgevoerd om te testen of het systeem goed om gaat met vele van dezelfde events en om de event wachtrij te testen, hierin komen alle events als er meerdere tegelijkertijd aangeroepen worden.
- **SetEffect test**
Deze test wordt uitgevoerd om te testen wat er gebeurt als 2 dezelfde events op exact hetzelfde moment aangeroepen worden.
- **Wait test**
Deze test worden uitgevoerd om te testen wat er gebeurt als er in een cluster een lange tijd gewacht wordt.

6.4.2 Eindconclusie

De testen zijn goed verlopen, het resultaat van de testen is te vinden in de bijlage Loadtest resultaat. Elke test is 10 keer uitgevoerd en alle 10 de keren hebben een 100% count behaald, dit betekend dat alle testen geslaagd zijn. Ook valt op dat elke bericht minder dan 2msec duurt, als we dit vergelijken met de 500 msec dat als requirement gezet was, dan lijkt het systeem veel beter dan verwacht. Door dit resultaat, is het systeem goedgekeurd.

6.5 Visuele componenten

Om de online collaboratie te ondersteunen, zijn er een aantal visuele componenten gemaakt die dit proces verbeteren. Het is namelijk belangrijk voor gebruikers om te zien wie er nog meer aanwezig is in de workspace en waar diegene mee bezig is. Omdat er geen best practices hierover te vinden zijn, is ervoor gekozen om te kijken naar applicaties die vergelijkbaar zijn, om inspiratie op te doen. Hieronder zullen deze inspiratiebronnen beschreven worden:

- **Google Docs**

Is een gratis tekstverwerker die vergelijkt kan worden met Word. Er zijn een aantal redenen waarom deze tool gekozen is als inspiratiebron. Het wordt veel door studenten gebruikt en het houdt zich heel goed aan de material design standaarden. Dit is erg belangrijk omdat dit de applicatie herkenbaar en intuïtief maakt. Vandaar dat google docs een goed uitgangspunt is voor de project approach tool.

- **Miro**

Is een online collaboratie whiteboard applicatie. Dit wordt als inspiratiebron gebruikt omdat het veel lijkt op de project approach tool. Er wordt bij Miro vooral gekeken naar hoe ze de online collaboratie hebben gemaakt en welke functies dit ondersteunen.

- **Mural**

Is een online visualisatie collaboratie workspace applicatie. Dit wordt vooral als inspiratiebron gebruikt omdat het veel lijkt op de project approach tool. Ook bij Mural wordt vooral gekeken naar hoe ze de online collaboratie hebben gemaakt en welke functies dit ondersteunen.

In alle 3 inspiratiebronnen, waren de volgende 2 visuele componenten aanwezig:

1. Een visueel component om aan te tonen wie er actief is in de workspace.
2. Een visueel component om aan te tonen wat die actieve gebruikers aan het doen zijn.

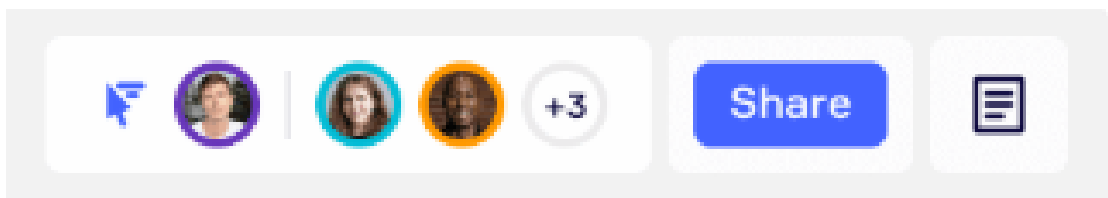
In overleg met de product owner, is ervoor gekozen om deze 2 componenten ook te implementeren in de project approach tool.

6.5.1 Actieve gebruikers

Eerst zal er getoond worden hoe de inspiratiebronnen, aantonen wie er online is en daarna zal er getoond worden wat wij uiteindelijk besloten hebben om te doen. Het aantonen van de actieve gebruikers is belangrijk omdat dit de samenwerking tussen gebruikers verbeterd.



Afbeelding 12 Google's manier om actieve gebruikers aan te tonen



Afbeelding 13 Miro's manier om actieve gebruikers aan te tonen



Afbeelding 14 Mural's manier om actieve gebruikers aan te tonen

Conclusie:

Het valt op dat elke applicatie een cirkel gebruikt voor de avatar en een kleur gebruikt om de gebruikers uniek te maken. Ze gebruiken allemaal ook een foto als avatar, maar google docs heeft ook een standaardoplossing voor als gebruikers geen afbeelding willen gebruiken en dat is de eerste letter van de naam te gebruiken.

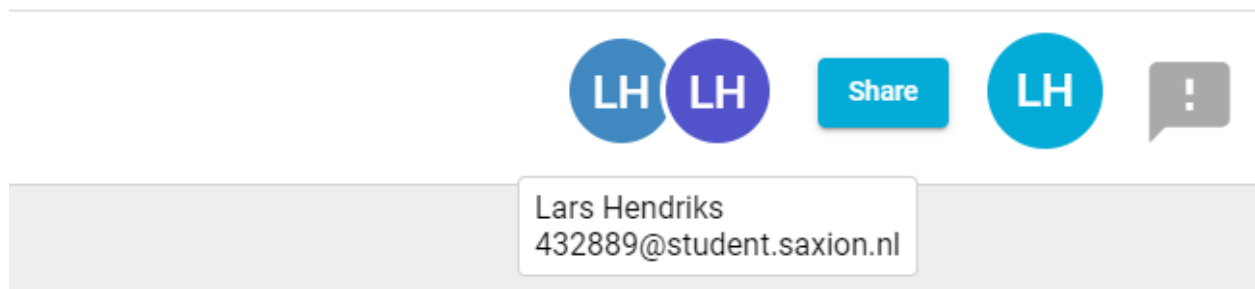
Er is uiteindelijk besloten in overleg met de product owner om het meer te laten lijken op google docs, omdat dit tijd zal schelen en omdat dit goed bij de applicatie past. Hieronder zal het resultaat te zien zijn, er is geen library gebruikt om dit te realiseren.

Om aan te tonen wie er aanwezig zijn, is er een lijst aan de navbar toegevoegd. Elke cirkel staat gelijk aan een gebruiker en de 1^e letter van de voornaam en de achternaam worden gebruikt, ook krijgt elke cirkel een unieke kleur zodat gebruikers makkelijk te herkennen zijn.



Afbeelding 15 Voorbeeld profiel icons

Als er over een cirkel wordt gehoverd zal er meer informatie te zien zijn over de gebruiker:



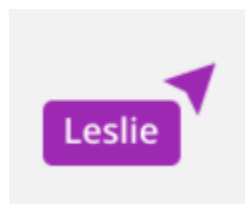
Afbeelding 16 Voorbeeld hover over profiel icon

6.5.2 Herkenning

Eerst zal er getoond worden hoe de inspiratiebronnen, aantonen wat degene die online is aan het doen is en daarna zal er getoond worden wat wij uiteindelijk besloten hebben om te doen.

A pink rectangular notification box with a small icon of a document with a red 'X' on the left. The text '100 active sockets' is written in white bold font inside the box.

Afbeelding 17 Voorbeeld google docs



Afbeelding 18 voorbeeld miro

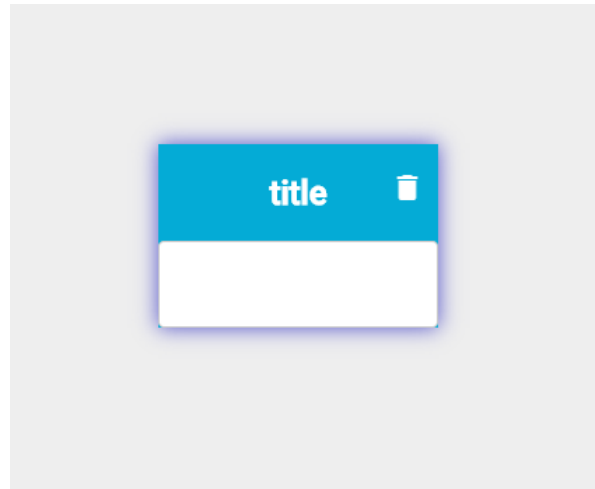


Afbeelding 19 Voorbeeld Mural

Conclusie:

Alle 3 de applicaties gebruiken kleuren om verschil te maken tussen alle gebruikers. Miro en Mural doen het op een vergelijkbare manier. Waarbij ze de muis van andere gebruikers weergeven met daarbij de naam van de gebruiker en een kleur. Deze kleur slaat vaak aan bij de avatar van de gebruikers die getoond worden in de navbar. Google docs gebruikt ook een kleur maar doet het op een simpelere manier bij ze de tekst een kleur geven zodat de gebruiker kan zien dat de andere gebruiker daarmee bezig is.

Miro en Mural hebben voor ons de mooiste oplossing omdat er duidelijk aangegeven wordt welke gebruiker de pijl is door een naam eraan te verbinden, maar dit zal door een te kort aan tijd niet mogelijk zijn. Daarom hebben we gekozen om net als google docs een gloed rondom objecten te plaatsen. Dit effect wordt getoond als een gebruiker, een object verplaatst of tekst veranderd of de titel/doel van een workspace aanpast. De kleur van de gloed staat gelijk aan de kleur van de avatar van een gebruiker.



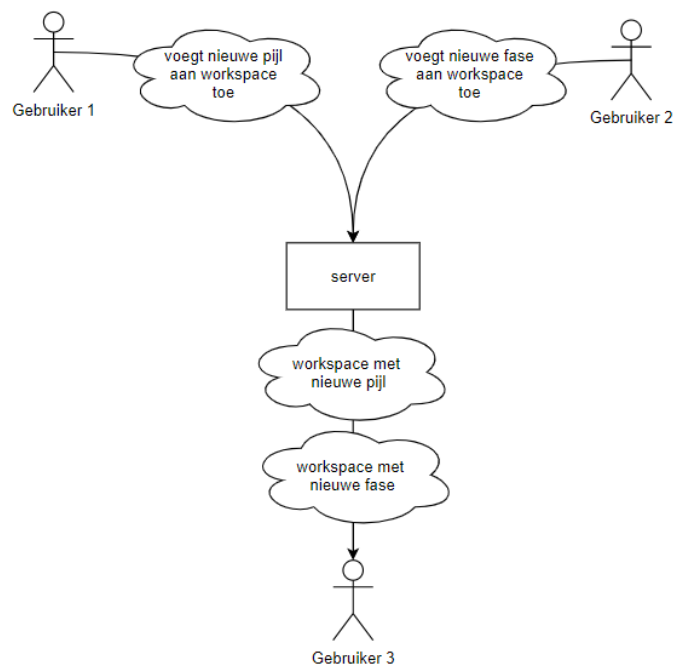
Afbeelding 20 Voorbeeld herkenning

6.6 Implementatie

In dit hoofdstuk wordt er beschreven, welke aanpassingen er nodig waren aan het concept om de applicatie te verwezenlijken en tegen welke problemen ik aan gelopen ben.

6.6.1 Synchronisatie probleem

De grootste aanpassing die gemaakt is, zit in het updaten van een workspace. Het was namelijk de bedoeling om het workspace object te sturen elke keer als er een aanpassing gedaan wordt. Maar dit brengt een synchronisatie probleem met zich mee, die hieronder te zien is:



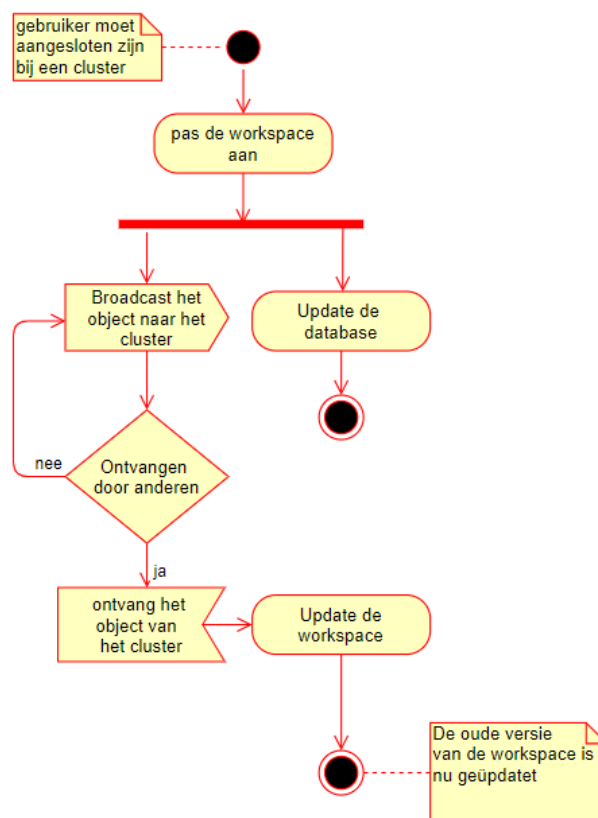
Afbeelding 21 Synchronisatie probleem

Gebruiker 1 voegt een nieuwe pijl toe aan de bestaande workspace, Gebruiker 2 voegt een nieuwe fase toe aan de bestaande workspace. Het object dat als eerste aankomt bij de server zal als eerst verstuurd worden naar de andere gebruikers. Dus gebruiker 3 ontvangt nu 2 workspace objecten met allebei een ander object dat is toegevoegd aan de workspace. Dus de applicatie overschrijft de workspace met het eerste workspace object dat ontvangen wordt, en daarna overschrijft de workspace met het tweede workspace object dat ontvangen is. Het resultaat is dan dat gebruiker 3 uiteindelijk een workspace heeft zonder nieuwe fase maar met een nieuwe pijl.

Er zijn 2 opties om dit probleem op te lossen, hieronder zullen ze kort beschreven worden:

- De verandering uit het verzonden object halen en die in de workspace van de gebruiker plaatsen.
- De payload veranderen zodat alleen de verandering (bijvoorbeeld alleen een pijl of alleen een fase) opgestuurd wordt in plaats van de gehele workspace.

Er is uiteindelijk gekozen om de payload te veranderen omdat dit ook een andere positief effect met zich mee brengt, namelijk de payload wordt kleiner. Dit is gunstig omdat hierdoor de berichten sneller verstuurd kunnen worden.

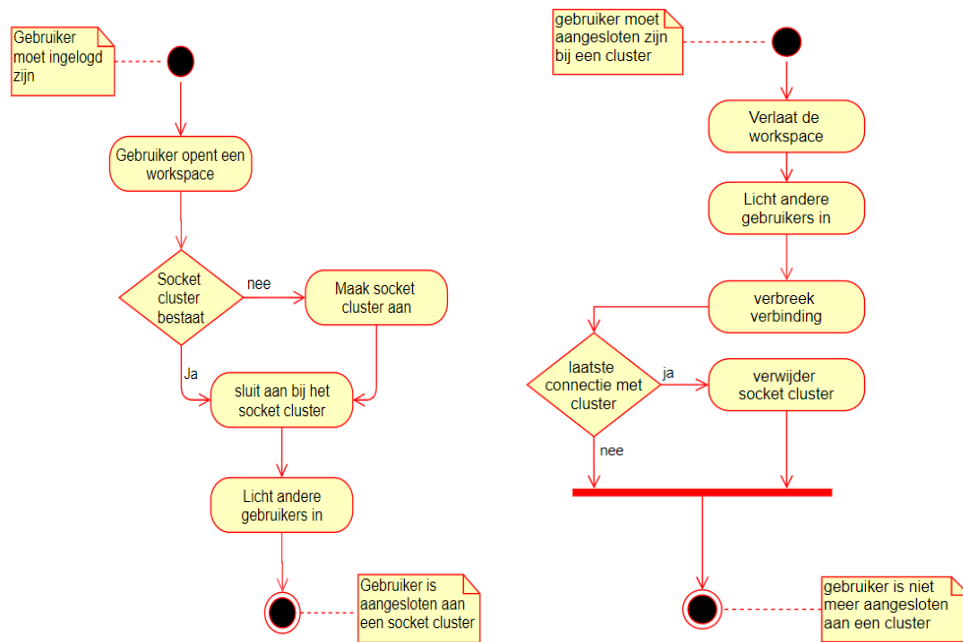


Afbeelding 22 Activity diagram vernieuwd updaten workspace

In Afbeelding 22 wordt getoond hoe dit is opgelost, nu worden objecten zoals pijlen, kaarten, fases etc. apart gebroadcast. Deze objecten worden dan aangepast of toegevoegd aan de huidige versie van de workspace van de gebruiker en daardoor blijven de workspaces gelijk.

6.6.2 Openen of verlaten workspace

Iets waar niet aan gedacht was, was dat gebruikers die aanwezig zijn in de workspace ingelicht zullen moeten worden als andere gebruikers de workspace openen of sluiten. Dit is benodigd om de component dat aangeeft wie er actief is in de workspace te updaten.



Afbeelding 23 Activity diagram openen workspace Afbeelding 24 Activity diagram verlaten workspace

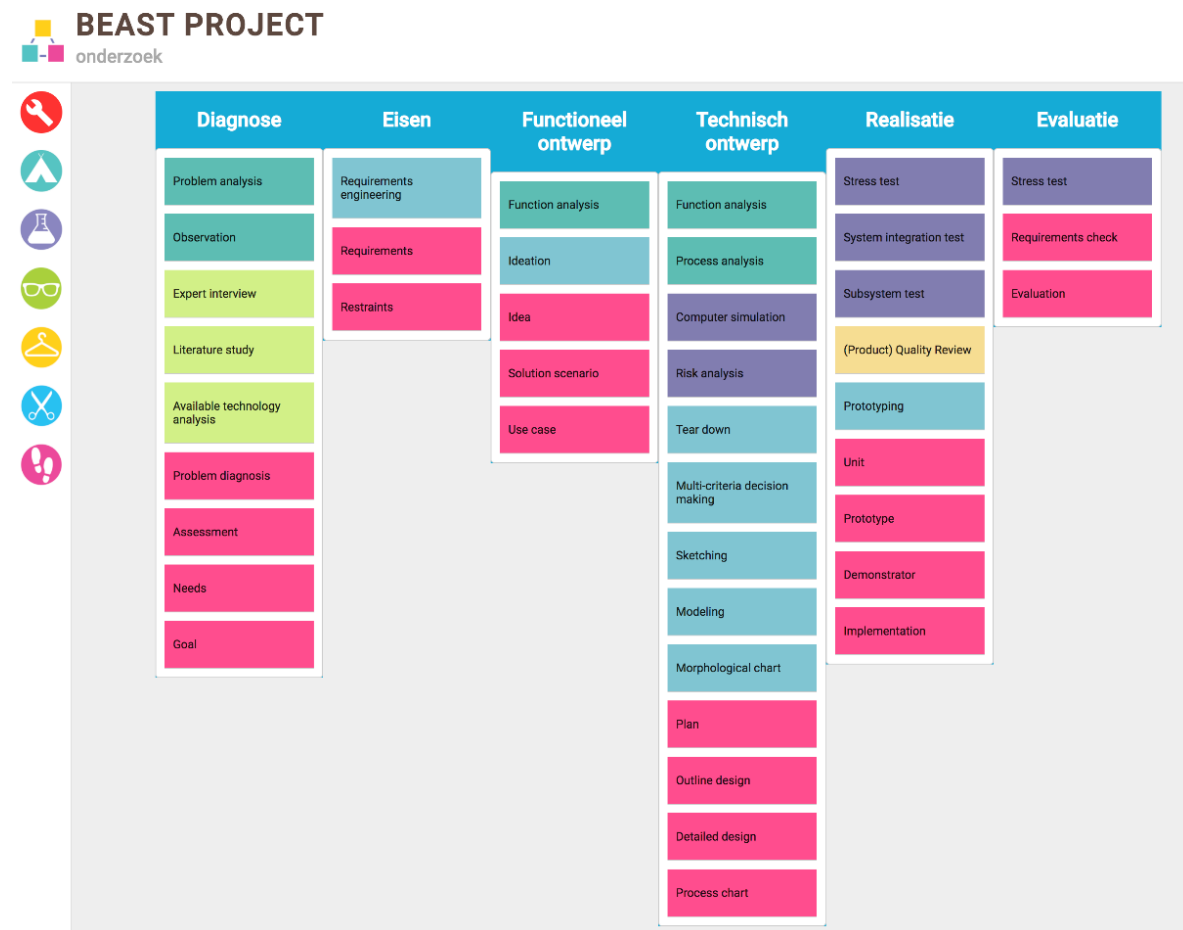
In de activity diagrams die hierboven getoond worden, is te zien dat nu andere de gebruikers worden ingelicht als een gebruiker het cluster verlaat of zich erbij aan sluit. Bij het openen van een workspace wordt het inlichten gedaan nadat hij is aangesloten bij het cluster, zodat hij een bericht naar alle gebruikers kan sturen. Bij het verlaten wordt het voor dat de verbinding verbroken wordt gedaan.

6.7 Gebruikerstest

Om online collaboratie goed te testen is er gekozen om te testen met een groep, deze test wordt tevens ook gebruikt om erachter te komen waar de gebruikers het meeste moeite mee hebben zodat daarop gefocust kan worden in de applicatie onboarding. In overleg met Etto Salomons is ervoor gekozen om de applicatie te gebruiken bij de minor 3S, dit zijn meerdere groepen studenten van verschillende richtingen die aan een project werken. Deze groepen zullen de applicatie gebruiken voor het maken van

hun projectaanpak en nadat ze het gemaakt hebben, zullen ze ongestructureerd bevraagd worden over wat ze van de applicatie vinden en of ze het nog eens zouden gebruiken.

Een van de 3S groepen was het beast project, zij hebben de applicatie gebruikt om een project approach in te plannen. Hieronder zal het resultaat te vinden zijn:



Afbeelding 25 Project aanpak gemaakt door leden van het beast project

6.7.1 Eindconclusie

Na het uitvoeren van de testen, is het resultaat geanalyseerd. Het resultaat zal te vinden zijn in de bijlage Gebruikerstest online collaboratie resultaat. Hieronder zal kort beschreven staan wat er is aangepast aan de applicatie.

- Het toevoegen van kaarten en bouwelementen zal uitgelegd worden tijdens het onboarding proces, waardoor het duidelijk zal worden voor gebruikers.
- Het is helaas niet mogelijk om de pijlen op een makkelijke manier te verwijderen, dit komt merendeel omdat er een verkeerde library gekozen is voor de pijlen, vandaar dat er ook aanbevolen is om de pijl library te vervangen.
- Er is in deze projectscope niet genoeg tijd om voor alle kaarten een beschrijving erbij te zetten, vandaar dat is aanbevolen om een beschrijving bij een kaart te zetten.
- Er is gekeken waarom sommige gebruikers geen invite krijgen en dit wordt veroorzaakt door het feit dat de invite email terecht komt in een spamfilter. Vandaar dat is aanbevolen om het huidige email adres te vervangen naar een andere email adres.
- Het verwijderproces is veranderd. Eerst moest een gebruiker de kaarten slepen naar een prullenbak, om de kaart te verwijderen. Nu kan een gebruiker de kaarten verwijderen door met de rechtermuisknop op de kaart te klikken en in het contextmenu op de “remove card” knop te klikken.

Alle aanbevelingen kunnen gevonden worden in het hoofdstuk Aanbevelingen.

7 Applicatie onboarding

Dit hoofdstuk zal vooral ingaan op applicatie onboarding en alle zaken die hierbij komen kijken.

7.1 Onboarding informatie

Om de flow van het onboarding proces te bepalen, heb ik me voorgedaan als een nieuwe gebruiker om erachter te komen wat er uitgelegd moet worden en welke informatie benodigd is.

- **Introductie**
Hierbij wordt de gebruiker welkom geheten en wordt uitgelegd wat het doel is van de applicatie.
- **Functies onder de profiel knop**
Hier wordt uitgelegd dat gebruikers, accountgegevens kunnen veranderen en uit kan loggen.
- **Functies onder de help knop**
Hier wordt uitgelegd dat gebruikers, een probleem kunnen melden, het onboarding proces kunnen herstarten, de privacy statement kunnen lezen en de about pagina kunnen bekijken.
- **Workspace**
Hier wordt uitgelegd wat een workspace is en hoe de workspace geopend kan worden.

- **Workspace aanmaken**
Hier wordt uitgelegd hoe een workspace aangemaakt kan worden.
- **Het veranderen van de titel en het doel van een workspace**
Hier wordt uitgelegd hoe de titel en het doel van de workspace aangepast kan worden.
- **Het plaatsen van bouwelementen zoals pijlen, fases en vragen**
Hier wordt uitgelegd wat het doel van deze elementen is en hoe ze geplaatst kunnen worden in de workspace.
- **Het plaatsen van onderzoeksmethodes in een fase**
Hier wordt uitgelegd wat het doel is van de verschillende types onderzoeksmethodes en hoe deze in een fase geplaatst kunnen worden.
- **Het selecteren van een deck**
Hier wordt uitgelegd hoe de gebruikers een ander deck kunnen selecteren.
- **Uitnodigen van andere gebruikers**
Hier wordt uitgelegd hoe de gebruikers een andere gebruiker kunnen uitnodigen.
- **Afsluiting**
Dit zal aangeven dat dit de einde van het onboarding proces is.

7.2 Types

Er zijn verschillende types van applicatie onboarding. Voordat er geschetst zal worden moet er eerst een keuze gemaakt worden over welke type onboarding toegepast moet worden op de applicatie. Hieronder zal een beschrijving te vinden zijn over de verschillende types (Psheycho, n.d.) (Mullin, 2019).

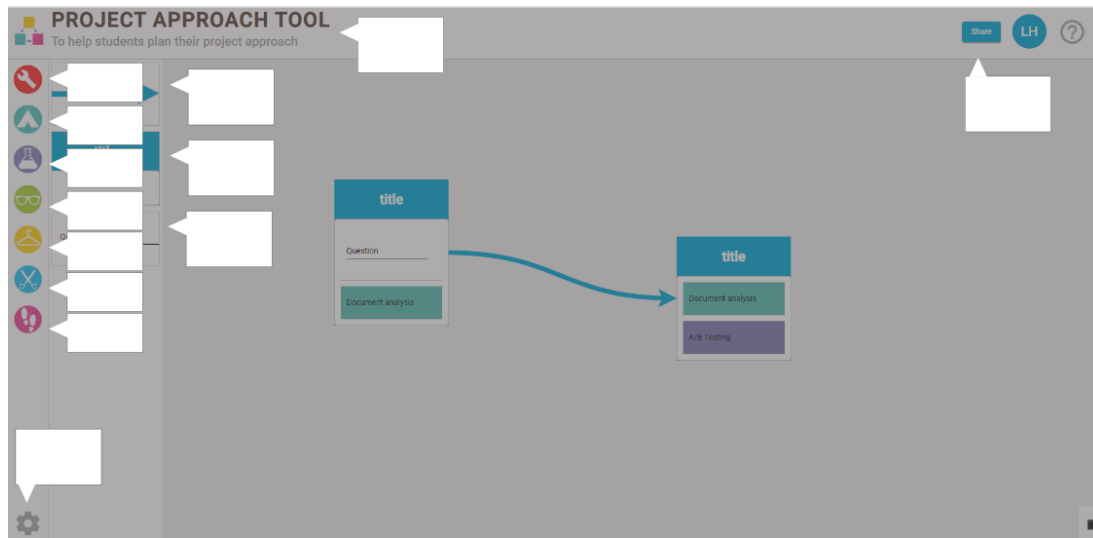
- **Benefit focused**
Focust zich vooral op de hoofdvoordelen (max 2-3) die je behaalt uit de applicatie.
- **Function focused**
Focust zich vooral op de hoofdfuncties (max 2-3) die zich bevinden in de applicatie.
- **Doing focused**
Hierbij wordt de gebruiker begeleid door de gehele applicatie.
- **Account focused**
Hierbij wordt de gebruiker begeleid door het account aanmaak proces.
- **Combined**
Voor complexe applicatie kun je ook verschillende types combineren.

Benefit & Function focused zijn niet geschikt omdat we in de applicatie meer dan 3 functies uit moeten leggen en omdat we het aanmaken van een account niet gaan ondersteunen met onboarding valt account focused ook af.

Nu blijven combined en doing focused nog over en omdat de doing focused geschikt is om te gebruiken om de gehele applicatie uit te leggen, is het overbodig om combined te gebruiken.

7.3 Mock-ups

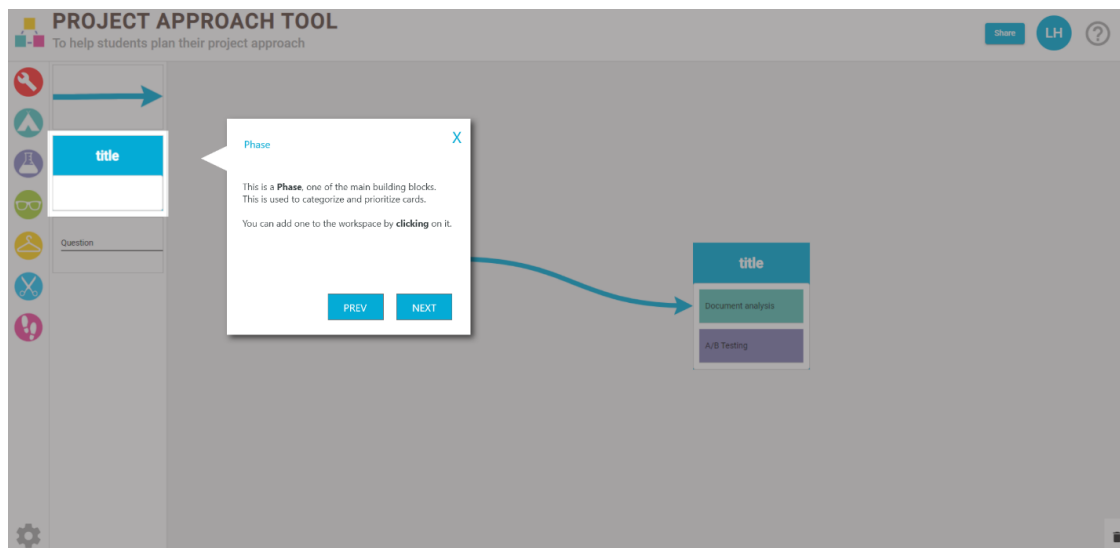
Voor de interface zijn er 2 concepten uitgewerkt. Het eerste idee was om een overlay te tonen waar alle informatie op staat. Het doel hiervan is om te kijken of alle informatie op een pagina past.



Afbeelding 26 Voorbeeld alle informatie op een pagina

Het werd al gauw duidelijk dat de informatie niet op één pagina ging passen en dat er te veel informatie in een keer aan een gebruiker getoond wordt, wat tegen best practices in gaat (Kosolapov, 2016).

Vandaar dat er is overgestapt naar het tweede concept, het tonen van de informatie stap voor stap. Deze zijn eerst geschetst op papier en dan vertaald naar mock-ups. Zie als voorbeeld Afbeelding 27 en zie de bijlage voor alle mock-ups.



Afbeelding 27 Voorbeeld onboarding stap voor stap

7.4 Onboarding library

Hieronder zal meer informatie te vinden zijn over de keuze die gemaakt is voor de onboarding library. De keuzeverantwoording bestaat uit een tabel waarin gewenste functionaliteiten worden vergeleken. Deze lijst met libraries is samengesteld door te kijken of ze geschikt zijn voor Angular 9+ en of ze ook nog onderhouden worden.

Gewenste functionaliteit	ngx-tour	ngx-joyride	Bdc-walkthrough
Mogelijkheid om terug te gaan in de tour	Ja	Ja	Nee
Mogelijkheid om de tour halverwege te sluiten	Ja	Ja	Nee
Mogelijkheid om de tour te herstarten	Ja	Ja	Ja
Lifecycle callbacks	Ja(maar beperkt)	Ja	Ja
Maakt gebruik van een overlay mogelijk	Nee	Ja	Nee
Maakt het mogelijk om naar meerdere pagina's te gaan *	Nee	Ja	Nee
Maakt het mogelijk om de styling makkelijk aan te passen *	Ja	Ja(maar beperkt)	Ja
Maakt het mogelijk om de plaatsing van een stap te veranderen *	Nee	Ja	Ja

Tabel 11 Vergelijking onboarding library

*Notitie: alle gewenste functionaliteiten met een *, zijn optioneel.*

Op de volgende pagina zullen de gewenste functionaliteiten uitgelegd worden.

Hieronder zullen de gewenste functionaliteit beschreven worden:

Gewenste functionaliteit	Beschrijving
Mogelijkheid om terug te gaan in de tour	De mogelijkheid om terug te gaan in een tour is belangrijk als de gebruiker toch iets niet begrijpt, kan deze altijd teruggaan om het nog eens door te lezen.
Mogelijkheid om de tour halverwege te sluiten	De mogelijkheid om de tour halverwege te stoppen is belangrijk. Omdat je niet wilt dat gebruikers een slechte ervaring hebben en zich geforceerd voelen om de tour te volgen.
Mogelijkheid om de tour te herstarten	De mogelijkheid om de tour te herstarten is belangrijk. Zodat gebruikers zichzelf kunnen helpen aan informatie als ze iets niet begrijpen.
Lifecycle callbacks	De lifecycle callbacks zijn belangrijk om een vloeiend proces te creëren. Want halverwege de tour wordt de sidebar geopend om de bouwelementen uit te leggen.
Maakt gebruik van een overlay mogelijk	Het is belangrijk dat er een overlay gebruikt omdat het geen interactieve onboarding tour is. Daarom dat het belangrijk is dat gebruikers niet ergens op kunnen klikken en de tour onderbreken.
Maakt het mogelijk om naar meerdere pagina's te gaan	Het is belangrijk dat er meerdere pagina's uitgelegd kunnen worden, zodat alles behandeld kan worden.
Maakt het mogelijk om de styling makkelijk aan te passen *	Het is belangrijk om de styling aan te kunnen passen, zodat het overeenkomt met de styling van de applicatie.
Maakt het mogelijk om de plaatsing van een stap te veranderen *	Het is kan handig zijn om de plaatsing van de stappen te veranderen. Zodat het niet buiten het scherm terecht kan komen.

Tabel 12 Beschrijving gewenste functionaliteit

Conclusie:

De ngx-joyride library komt als beste uit de vergelijking en heeft de meeste gewenste functionaliteit. Vandaar dat deze library gekozen is om te implementeren als onboarding library in de applicatie.

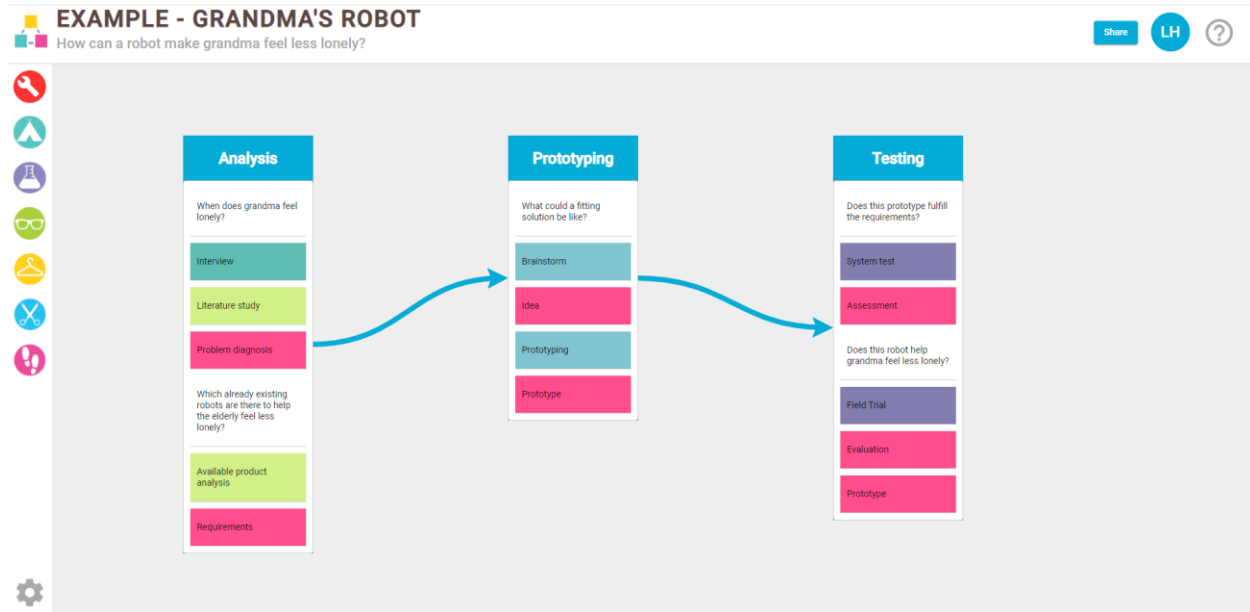
7.5 Gebruikerstest

Om te testen of de implementatie van de onboarding ook goed werkt en duidelijk is, is er gekozen om gebruikerstesten uit te voeren. Hieronder zal er uitgelegd worden hoe de testen eruitzien en wat de uitkomst hiervan is. De gebruikerstest is verdeeld in drie test scenarios.

Als eerste test scenario zijn de gebruikers de onboarding tour laten volgen. De gebruikers zijn geobserveerd tijdens dit proces en er is genoteerd wat er op viel, aan het einde van de onboarding zijn er nog een aantal vragen gevraagd die gebaseerd waren op wat er op viel tijdens de observaties. Uit dit scenario kwam vooral naar boven welke teksten onduidelijk waren en waar nog informatie nodig is.

Het tweede scenario bestaat uit het tonen van een voorbeeld workspace en de gebruikers vragen of ze uit kunnen leggen hoe het voorbeeld in elkaar zit en op deze uitleg vragen stellen. Zo wordt het duidelijk of het voorbeeld dat gebruikt wordt tijdens de onboarding begrijpelijk is en wordt het duidelijk of ze de uitleg van de onboarding ook kunnen toepassen op het voorbeeld.

Het voorbeeld zal hieronder te zien zijn. Dit voorbeeld is in samenwerking met Danny Plass gemaakt, zodat het geschikt is voor studenten.



Afbeelding 28 Voorbeeld workspace

Tijdens het derde scenario worden de gebruikers gevraagd om een fase toe te voegen, de fase te verbinden met een pijl aan het huidige voorbeeld, 2 vragen en 4 kaarten toe te voegen aan de fase. De gebruikers worden geobserveerd tijdens dit scenario, daarna worden er vragen gesteld die gebaseerd zijn op de observaties. Hiermee wordt gekeken of de gebruikers met de uitleg van de onboarding zelf kunnen werken aan een project approach.

Er is gekozen in overleg met de productowner Danny Plass, om met 2 gebruikers te testen. Want met 2 gebruikers worden de meeste problemen al gevonden (Nielsen, 2012) (Nguyen, 2013).

7.5.1 Eindconclusie

Na het uitvoeren van de testen, is het resultaat geanalyseerd. Het resultaat van deze testen is te vinden de bijlage Gebruikerstest onboarding resultaat. Hieronder zal kort beschreven staan wat er is aangepast aan de applicatie.

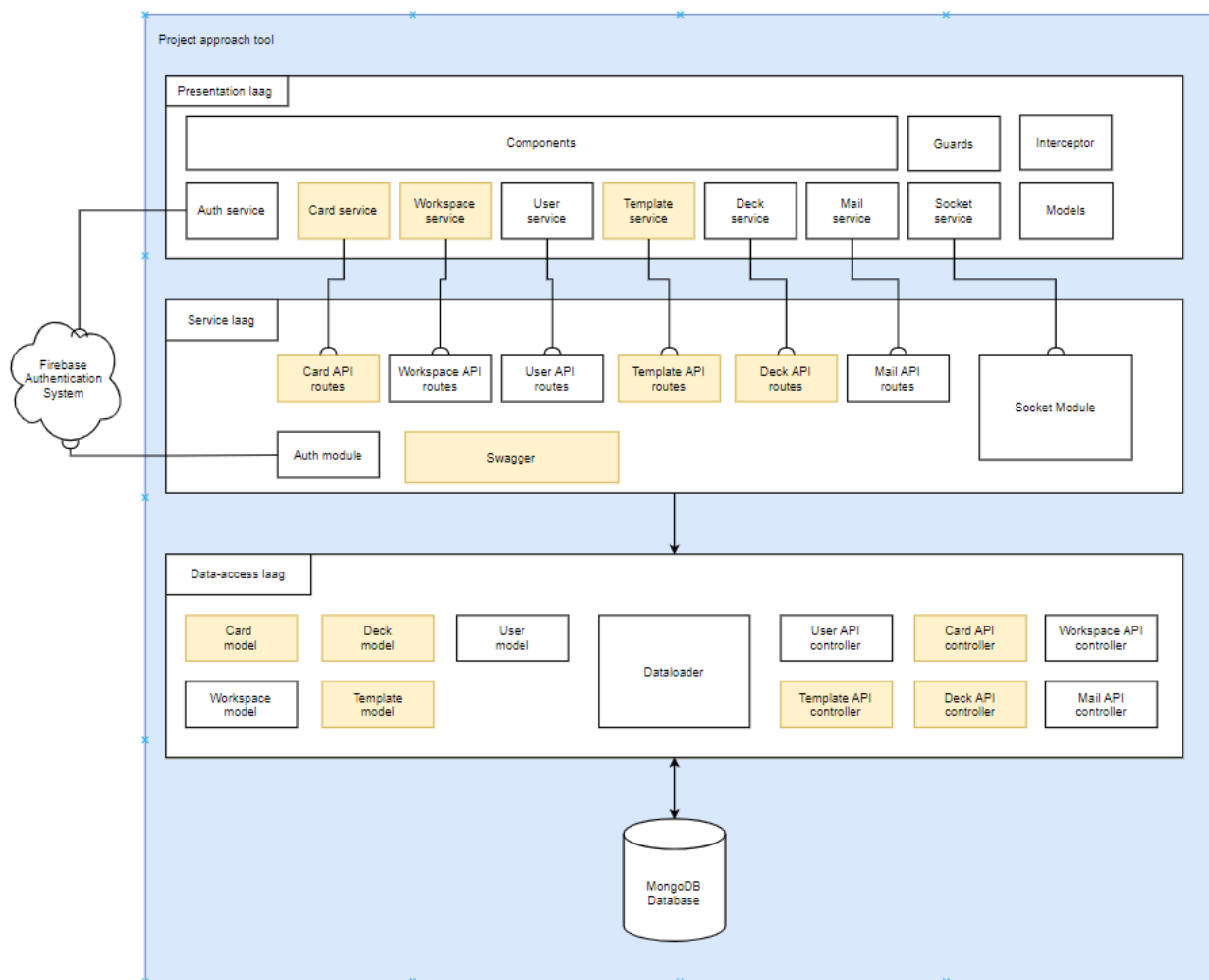
- De introductie is aangepast zodat gebruikers weten dat ze het onboarding proces kunnen stoppen en dat ze het doel van de applicatie weten.
- De workspace uitleg is aangepast, zodat het duidelijk is voor gebruikers dat ze met de next knop naar de workspace gaan.
- Bij de uitleg van de research methodes is erbij gezet dat het gaat om kaarten, die naar fases gespeeld kunnen worden. Zodat hier geen verwarring over kan ontstaan.

- De uitleg van de pijlen is zo uitgebreid, dat het nu hopelijk duidelijker is voor de gebruikers hoe een pijl geplaatst moet worden en ook is er bijgezet dat de pijlen vast zitten aan de fases en niet aan de pijlen.
- Ook is de wachttijd van het openen van de sidebar verminderd, zodat het niet meer lijkt op een crash.
- Ook is de padding tussen de kaarten en de zijkant van de fase vergroot zodat het nu duidelijk is dat een pijl verbonden is aan een fase in plaats van aan een kaart.

Verder kwamen er ook nog 3 aanbevelingen uit het resultaat. Deze zullen te vinden zijn in het hoofdstuk Aanbevelingen.

8 Eindsituatie

Om een beter beeld te geven van hoe de eindsituatie eruitziet en om aan te geven wat er aan de applicatie heb toegevoegd, is er een software architectuur gemaakt.

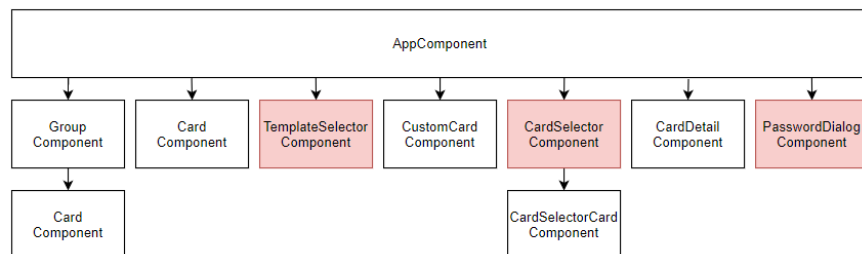


Afbeelding 29 Software architectuur eindsituatie

In Afbeelding 29 wordt de software architectuur van de eindsituatie getoond, de onderdelen met een gele kleur, zijn de onderdelen die in de beginsituatie aanwezig waren, deze onderdelen zijn tevens wel uitgebreid of aangepast.

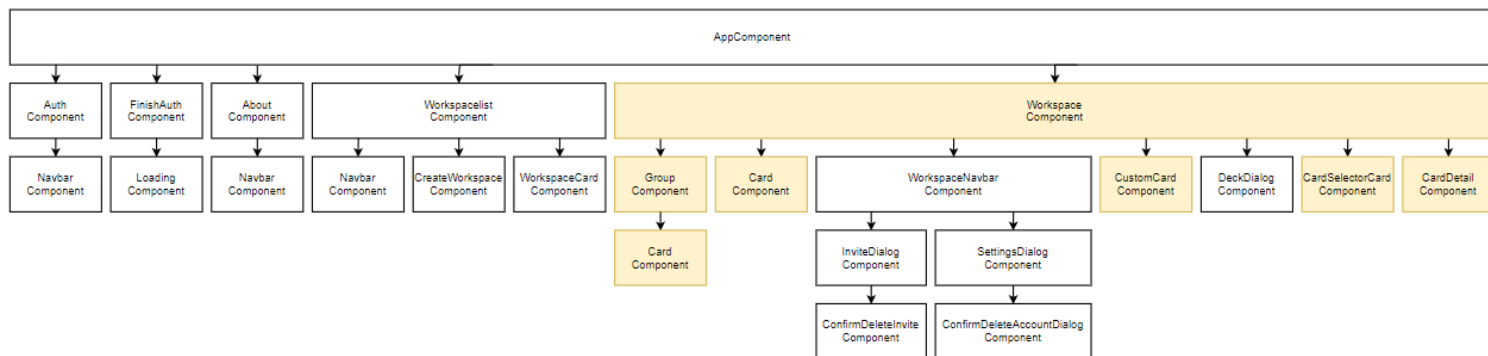
Presentation laag

Om beter aan te kunnen tonen welke componenten er aanwezig zijn in de angular applicatie en welke er aan toegevoegd zijn. Zullen er 2 component hierarchy overviews getoond worden, een van de beginsituatie en een van de eindsituatie.



Afbeelding 30 component hierarchy overview beginsituatie

In Afbeelding 26, wordt de component hierarchy overview van de beginsituatie getoond, de rode onderdelen zijn onderdelen die in de eindsituatie zijn verwijderd.



Afbeelding 31 component hierarchy overview eindsituatie

In Afbeelding 31, wordt de component hierarchy overview van de eindsituatie getoond, de gele onderdelen zijn de onderdelen in beginsituatie ook aanwezig waren. De AppComponent van de beginsituatie is nu omgezet naar de WorkspaceComponent.

Ook zijn er nog andere onderdelen aanwezig in de presentation laag zoals guards, interceptors en models. De guards zorgen voor de bewaking van een route en worden gebruikt om toegang naar een specifieke route toe te staan of te weigeren. Denk bijvoorbeeld aan het inloggen, als een gebruiker niet

ingelogd is, hoort hij geen toegang te krijgen tot de pagina's die bedoeld zijn voor gebruikers die ingelogd zijn.

De interceptors worden gebruikt om de uitgaande requesten aan te passen, in dit geval door een authenticatie code toe te voegen aan de header van het request, als de gebruiker is ingelogd. Hierdoor krijgt de applicatie toegang tot data van de server.

Verder bevat de presentation laag ook nog services, met deze services kunnen er requesten naar de routes in de service laag gestuurd worden en er zijn ook nog models aanwezig, deze models worden gebruikt om de typing te hanteren in typescript en ze staan gelijk aan de models in de data-access laag.

Service laag

De service laag bestaat merendeels uit nodeJS routes, Routes worden gebruikt om requesten door te sturen naar de bijbehorende controllers, controllers worden beter uitgelegd in de volgende laag.

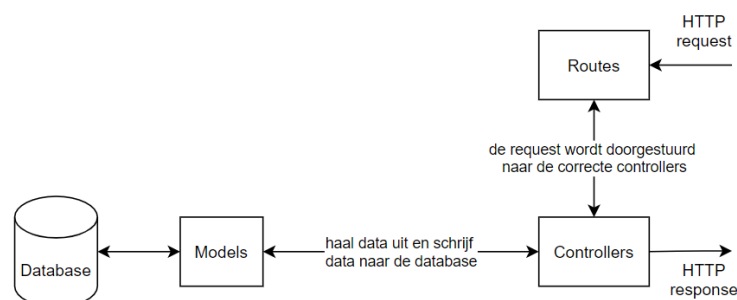
Verder bevindt zich nog swagger functionaliteit in de service layer, hiermee wordt er OpenAPI 2.0 specificatie gegenereerd. Deze specificaties worden gebruikt om API's te beschrijven.

Ook bevat de service laag een auth module, dit is een module waarmee de routes beveiligd worden. Er wordt namelijk een authenticatie code in de header meegestuurd door de interceptor in de presentation laag. De auth module controleert de authenticatie code en dan wordt het request doorgestuurd naar de route, als de code goedgekeurd is en als het niet goedgekeurd is, dan wordt het request geweigerd.

De service laag bevat ook een socket module, deze module bevat alle functionaliteit met betrekking tot de sockets, over de werking van deze sockets kan er meer informatie gevonden worden in hoofdstuk Online collaboratie.

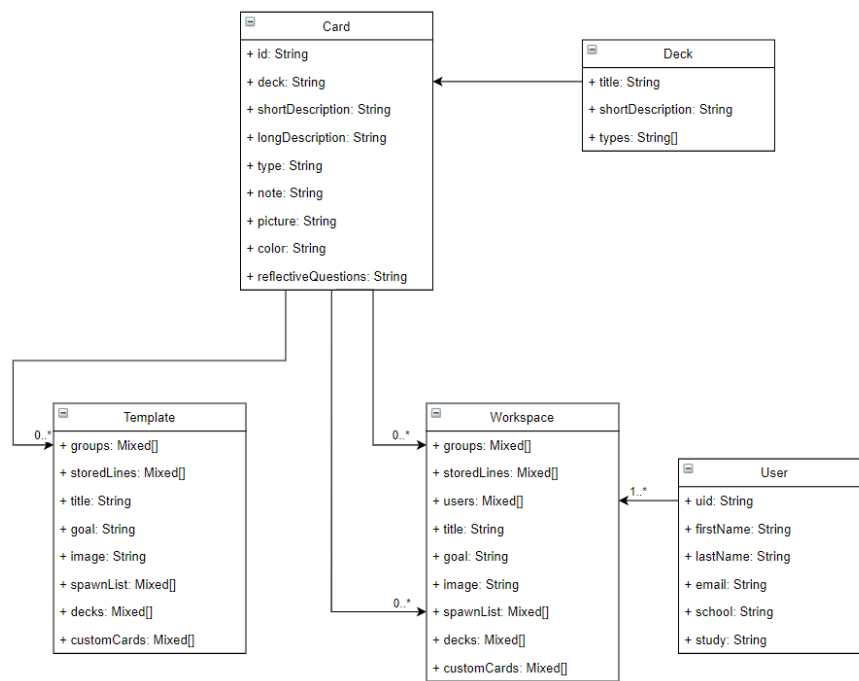
Data-access laag

De data-access laag, is de verbinding tussen de mongoDB database en service laag. De routes uit de service laag zullen requesten doorsturen naar de controllers in de data-access laag, zodat deze requesten behandeld kunnen worden en zodat er een response teruggestuurd kan worden.



Afbeelding 32 data flow backend

In Afbeelding 32 wordt uitgelegd hoe de controllers te werk gaan, ook is in dezelfde afbeelding te zien dat de models gebruikt worden om data uit de database te halen. Hieronder zal meer informatie over de models te vinden zijn, in de vorm van een database diagram.



Afbeelding 33 Database diagram eindsituatie

In Afbeelding 4 data flow backend Afbeelding 33 worden 5 models getoond, er zijn een aantal relaties die beter uitgelegd zullen worden. Card model hebben een relatie met template en workspace omdat groups kaarten bevatten, groups zijn in dit geval de fases die in een workspace te zien zijn. Ook kan de spawnlist kaarten bevatten, de spawnlist is een lijst waarin kaarten terecht komen als ze in eerste instantie toegevoegd worden aan een workspace.

Ook heeft een Card een relatie met een Deck, omdat er meerdere kaarten in een deck zitten. Verder is er nog een relatie dus de User en de Workspace, dit wordt gebruikt om te kijken welke gebruikers toegang hebben tot de workspace.

Verder zit er nog een dataloader in de data-access laag, de dataloader haalt research methods kaarten, stepping stones kaarten, templates en decks uit JSON-bestanden. De dataloader controleert of deze data al aanwezig en of de data gelijk is aan wat er in de database staat. Als de data nog niet aanwezig is of als de data niet gelijk is dan wordt de data toegevoegd/overschreven.

8.1 GDPR

Om de applicatie klaar te maken voor gebruik, moet de applicatie voldoen aan de GDPR. Dit is een wet dat gaat over privacy and IT-beveiliging. Er is kort onderzocht, welke punten van de GDPR relevant zijn. In de bijlage GDPR-onderzoek, zal het gehele onderzoek te vinden zijn. Hieronder zal een korte beschrijving te vinden zijn, dat gaat over wat er aan de applicatie is veranderd om te voldoen aan de GDPR (Saltis, 2020).

- In de accountinstellingen is het nu mogelijk om accountgegevens te veranderen, dit is geïmplementeerd om te voldoen aan artikel 16 van de GDPR.
- In de accountinstellingen is het nu mogelijk om het account en alle bijbehorende gegevens te verwijderen om te voldoen aan artikel 17 van de GDPR.
- In de accountinstelling is het nu mogelijk om de accountgegevens te exporteren naar JSON, dit is geïmplementeerd om te voldoen aan artikel 20 van de GDPR.
- Er wordt een privacy statement geschreven door de product owner, om te voldoen aan artikel 12,13,14 en 15 van de GDPR.
- Persoonlijke gegevens van de gebruikers worden geëncrypt met AES-256, dit is een van de betrouwbaarste en veiligste versleutelingstechnieken (Bradford, 2020) (Location, 2018) (Parsons, 2020). Dit is toegevoegd om te voldoen aan artikel 32 van de GDPR.

Er is ook nog een gesprek geweest met Erik van den beld, hij is de functionaris gegevensbescherming & docent bij het Saxion. Hij gaf aan dat door de omvang van het project, de applicatie zal voldoen aan de GDPR als dit geïmplementeerd is. Hij gaf wel aan dat hij het belangrijk vindt dat een DPIA uitgevoerd wordt, DPIA staat voor data protection impact assessment en wordt gebruikt om te kijken of er nog privacy risico's aanwezig zijn in de applicatie. Omdat het uitvoeren van een DPIA tijdens dit projectduur niet meer lukt, zal hierover ook een aanbeveling te vinden zijn, in hoofdstuk Aanbevelingen.

9 Eindconclusie

In dit hoofdstuk wordt antwoord gegeven op de hoofdvraag **“Hoe kan de applicatie worden uitgebreid, zodat gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent?”**

Ook worden er aanbevelingen gedaan zodat de applicatie nog verbeterd kan worden en zal ik reflecteren over hoe het project gegaan is en hoe ik mezelf kan verbeteren.

9.1 Was het succesvol?

Het doel van het afstudeerproject was om de project approach tool uit te breiden zodat gebruikers online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent. Er zijn 2 onderdelen geïmplementeerd die samen dit probleem oplossen. Voor beide oplossingen zijn requirements opgesteld, zijn er onderzoeksvragen onderzocht, zijn er schetsen en mock-ups gemaakt, is het geïmplementeerd, is er documentatie geschreven en zijn er testen gemaakt en uitgevoerd.

Het eerste onderdeel was het implementeren van een socket service, die het mogelijk maakt dat data realtime gedeeld kan worden met de andere gebruikers die toegang hebben tot de workspace waarin de projectaanpak gemaakt is, in samenwerking met de elementen die aangeven wie er aanwezig is in de applicatie en wat deze gebruikers aan het doen zijn. Zal dit ervoor zorgen dat gebruiker met elkaar kunnen werken aan een projectaanpak.

Het tweede onderdeel was het implementeren van het onboarding proces. Dit is een tour waarmee gebruikers geïnformeerd worden over de werking van de applicatie. Dit zorgt ervoor dat gebruikers zonder begeleiding van een docent om kunnen gaan met de applicatie.

Als ik het eindproduct vergelijk met de requirements en de mock-ups die aan het begin van het project opgesteld waren, is het duidelijk dat het minimal viable product behaald is en dat gebruikers hiermee online kunnen samenwerken aan een projectaanpak, zonder de begeleiding van een docent.

9.2 Aanbevelingen

Op basis van mijn ervaring gedurende het project, komen hier de volgende aanbevelingen uit:

Socketserver uitbreiding

Mocht het voorkomen dat 10.000 actieve sockets behaald wordt, dan zou ik voorstellen om de backend uit te breiden. De server kan op 2 manier uitgebreid worden:

- **Verticaal schalen**
Door de server uit te breiden met meer verwerkingscapaciteit (CPU, RAM).
- **Horizontaal schalen**
Door de server uit te breiden door meer servers toe te voegen. De socket.io library is deels uitgekozen omdat het ook load balancing ondersteunt, dit is namelijk nodig om het horizontaal schalen van de socketserver mogelijk te maken.

Sockets monitoring

Het kan handig zijn om te monitoren hoeveel sockets verbonden zijn aan de backend. Hiermee kan namelijk bekeken worden wanneer de server uitgebreid dient te worden.

Ook zal de monitoringtool, de productowner van data kunnen voorzien. Zoals hoeveel gebruikers online zijn en hoeveel berichten verstuurd worden etc.

Email aanpassen

Voor het verifiëren van logingegevens en het uitnodigen van andere gebruikers voor een workspace, worden 2 verschillende email adressen gebruikt. Momenteel wordt namelijk een Gmail account gebruikt voor het uitnodigen en een no-reply firebase email voor het verifiëren. Als er een domein gekocht wordt voor deze applicatie zou het handig zijn dat deze email adressen vervangen worden tot 1 no-reply email, dit zal namelijk ook deels voorkomen dat het automatische als spam wordt gemarkeerd.

Gebruik door studenten

Het is naar mijn mening belangrijk dat studenten les krijgen over het DOT-framework, de research methods en de stepping stones. De applicatie is namelijk getest met tweedejaars studenten die nog geen kennis hadden van deze onderzoeksmethodes en het werd al gauw duidelijk dat ze niet begrepen hoe ze de kaarten in de applicatie kunnen gebruiken om een project aanpak kunnen maken.

Kaarten uitbreiden

Uit de testen kwam vooral ook naar boven dat de kaarten niet duidelijk zijn voor gebruikers die nog niet met research methods of stepping stones hebben gewerkt. Daarom is het belangrijk dat deze informatie te vinden is binnen de applicatie. Dit zal namelijk het maken van een project aanpak versnellen en de aanpak ook duidelijker maken. Nu moeten de gebruikers van de applicatie namelijk externe bronnen gebruiken om informatie over de kaarten te vinden.

Pijlen library vervangen

Tijdens het analyseren van de beginsituatie kwam al snel naar boven dat de meeste bugs veroorzaakt worden door de library leaderline dat gebruikt wordt voor het maken van de pijlen. Daarom stel ik voor om de library te vervangen door een library die geschikt is voor Angular 9+. Dit zal ervoor zorgen dat 9 bugs opgelost worden en dat er ook geen andere bugs kunnen ontstaan door deze library als er verder wordt gewerkt aan de applicatie.

Groepering in fases

Dit kwam tijdens het testen van de onboarding met gebruikers naar boven. Een van de gebruikers die de applicatie testte, had niet helemaal door welke methodes en deliverables bij een onderzoeksvraag hoorde. Vandaar dat een groepering binnen in de fases een goede optie kan zijn. Zodat meerdere groepen gemaakt kunnen worden binnen fases en het duidelijk is, wat bij elkaar hoort.

Zoom functie in de workspace

Toen ik het resultaat bekeek wat er gemaakt was tijdens de gebruiker test van de online collaboratie. Zag ik dat de groep geen pijlen en onderzoeksvragen had gebruikt omdat anders de projectaanpak te groot werd voor het scherm. Daarom denk ik dat de applicatie uitbreiden met een zoom functie, veel kan helpen bij het maken van grote projectaanpakken.

Ook zal dit het gebruik van de stepping stones templates mogelijk maken. Deze templates zijn nu nog te groot, met de zoom zou er meer plek zijn en zal het mogelijk zijn om deze templates te gebruiken.

Notitie overzicht

Nu worden alle notities toegevoegd aan de kaarten door op de kaart te klikken, dit is tevens ook de enige manier om de notities te bekijken. Vandaar dat het handig kan zijn een notitie overzicht toe te voegen. Zodat alle notities in een keer te zien zijn, in plaats van dat je elke kaart langs moet klikken om alle notities te lezen.

Controleer functie

Danny en ik hebben het er vaak over gehad hoe wij de hints en tips zouden kunnen toevoegen, want je wilt eigenlijk niet dat gebruikers er mee overladen worden. Vandaar dat een controleer functie misschien een handig optie is om toe te voegen. Die bijvoorbeeld controleert welke methodes zijn gebruikt en daarna controleert of de bijbehorende deliverables ook aanwezig zijn en als deze niet aanwezig zijn, hierover een hint zal geven.

Testen met eerstejaars

Momenteel worden de eerstejaars geleerd om gebruik te maken van de concepten en kaarten, die gebruikt worden in de applicatie. Helaas kon ik geen eerstejaars vinden om de applicatie mee te testen en ik denk dat dit toch belangrijke feedback terug kan geven. Vandaar dat ik aanraad om toch met een groep eerstejaars de applicatie te testen.

SSL/TSL

Het is belangrijk dat de applicatie een SSL/TSL certificaat gebruikt om de applicatie beter te kunnen beveiligen. Het beschermt namelijk alle communicatie en de data die meegestuurd wordt met de requesten. Er kan dan namelijk gebruik gemaakt worden van HTTPS en WSS. Dit zal alle communicatie naar de server beveiligen van onder andere man-in-the-middle-attacks. HTTPS zal gebruikt kunnen worden om de requesten te beveiligen en WSS zal gebruikt kunnen worden om alle socket communicatie te kunnen beveiligen. (WebSocket Security, 2020) (Penney, 2017)

Backups

Het is na mijn mening belangrijk dat er dagelijks back-ups gemaakt worden van de database. Aangezien de applicatie veel gebruikt zal worden door studenten, is het belangrijk dat hun werk niet verloren gaat. Vandaar dat een back-up systeem heel handig is, om dit te voorkomen.

DPIA

Tijdens het gesprek met Erik van den beld , gaf hij aan dat er een DPIA uitgevoerd moet worden. Voor de applicatie is het niet verplicht volgens de GDPR maar hij gaf wel aan dat ermee privacy risico's ontdekt kunnen worden en dat is wel heel belangrijk.

9.2.1 Top 5

Omdat er zoveel aanbevelingen zijn, is ervoor gekozen om een top 5 te maken van welke aanbevelingen de hoogste prioriteit hebben, de top 5 geldt als volgt:

1. Socketserver uitbreiden

Dit is naar mening de belangrijkste aanbeveling, omdat andere scholen ook gebruiken kunnen maken zal het aantal sockets snel omhooggaan. Vandaar dat het belangrijk is dat de socket server uitgebreid kan worden.

2. SSL/TSL

Het is erg belangrijk dat de communicatie met de server beveiligd wordt, om aanvallen en privacy schending te voorkomen.

3. Gebruik door studenten

Om ervoor te zorgen dat studenten het DOT-framework ook goed begrijpen, is het naar mijn mening belangrijk dat ze hier les over krijgen.

4. Pijlen library vervangen

Omdat de pijlen library voor veel problemen zorgt, is het belangrijk dat deze uiteindelijk vervangen wordt. Het kan momenteel gebruikt worden maar tijdens het scrollen en zoomen van de pagina, zijn er veel problemen.

5. Kaarten uitbreiden

Het is erg belangrijk dat de kaarten duidelijk zijn, vandaar dat het handig is dat er informatie over de kaart beschikbaar is, vooral het doel van de kaarten was vaak niet duidelijk.

9.3 Reflectie

Een van de grootste dingen waar ik last van had waren de maatregelen voor het coronavirus in combinatie met de zomervakantie, dit zorgde voor meerdere problemen.

In het eerste kwartiel was mijn motivatie heel hoog en was ik naar mijn mening goed aan het project bezig. Maar in het tweede kwartiel dat na de zomervakantie plaatst vindt, merkte ik dat de maatregelen steeds meer effect op mij begonnen te hebben met name op mijn motivatie en moet ik eerlijk toegeven dat ik een paar keer eerder gestopt ben met het werken aan het project omdat ik gewoon niet aan de gang kwam. Uiteindelijk heb ik dit gemeld bij beide begeleiders en kwamen ze wel allebei met oplossingen om mij te helpen en sommige tips helpt mij om het project verder goed door te komen.

Door de coronamaatregelen moest ik ook vanuit huis werken aan mijn project. In het begin van het project had er geen last van en vond ik het zelfs fijn om thuis te werken maar hoe langer dit doorging hoe meer ik merkte dat ik vaker afgeleid werd door andere dingen. Toen heb ik dit met mijn begeleiders besproken en die gaven aan dat het misschien mogelijk is dat ik een paar dagen op het lectoraat en een paar dagen thuis werk, maar helaas kwam een week daarna strenge maatregelen die dit niet meer mogelijk maakte. Uiteindelijk heb ik zoveel mogelijk mijn best gedaan om mijn concentratie te bewaren, maar ik merk toch dat ik er over algemeen veel last van heb gehad.

Ik denk ook dat ik misschien verder gekomen zou zijn als ik de beginsituatie opnieuw gemaakt zou hebben. Ik merkte namelijk veel problemen tijdens de development die oorspronkelijke uit de beginsituatie kwamen en dit heeft uiteindelijk veel tijd gekost.

Wat ik wel erg fijn vond was de tijd die aan mij besteed werd, we hadden elke week een meeting ingepland waardoor ik elke week feedback kon krijgen en met Danny goed kon bespreken wat er de week daarop gedaan zou worden. Hierdoor vond ik dat het projectmanagement erg makkelijk verliep.

Ook vind ik het erg jammer dat ik niet heb kunnen testen met eerstejaars, de studenten waarmee ik heb kunnen testen hadden geen kennis van de concepten en kaarten waarvan de tool gebruikt maakt, vandaar dat het resultaat uit de gebruiker testen van het onboarding proces andere problemen opleverde. Eerstejaars studenten wordt momenteel wel geleerd om deze concepten en kaarten te gebruiken. Mocht ik extra tijd gehad hebben en mochten de omstandigheden rondom de coronamaatregelen anders zijn dan had ik zeker met de eerstejaars nog eens getest.

Al met al vond ik het een erg leuk project om aan te werken en heb ik er veel van kunnen leren. Ook vond ik het fijn dat er geluisterd werd naar mijn ideeën, vandaar dat ik ook aardig veel aanbevelingen heb kunnen doen.

10 Bibliografie

Au-Yeung, J. (2020, April 28). *Angular Security Best Practices*. Opgeroepen op Mei 29, 2020, van Medium: <https://medium.com/javascript-in-plain-english/angular-security-best-practices-54fadf8974a1>

Basu, S. (2019, December 10). *Real-Time Communication Techniques*. Opgeroepen op Juni 4, 2020, van Telerik: <https://www.telerik.com/blogs/real-time-communication-techniques>

- Bradford, C. (2020, Juli 31). *5 Common Encryption Algorithms and the Unbreakables of the Future*. Opgeroepen op Oktober 1, 2020, van storagecraft: <https://blog.storagecraft.com/5-common-encryption-algorithms/>
- ictresearchmethods. (2018, September 7). *The DOT Framework*. Opgeroepen op September 7, 2020, van ictresearchmethods: http://ictresearchmethods.nl/The_DOT_Framework
- Indrukwekkende documenten maken*. (sd). Opgeroepen op September 7, 2020, van google docs: <https://www.google.nl/intl/nl/docs/about/>
- Kosolapov, A. (2016, Juni 20). *A Roadmap To Building A Delightful Onboarding Experience For Mobile App Users*. Opgeroepen op September 30, 2020, van Smashing magazine: <https://www.smashingmagazine.com/2016/06/complete-roadmap-building-delightful-onboarding-experience-mobile-app-users/>
- Location, I. (2018, November 18). *What are the most secure encryption algorithms*. Opgeroepen op Oktober 1, 2020, van iplocation: <https://www.iplocation.net/encryption>
- Methods*. (2018, September 7). Opgeroepen op Mei 29, 2020, van ictresearchmethods: <http://ictresearchmethods.nl/Methods>
- Mullin, S. (2019, Juli 16). *6 User Onboarding Flow Examples (with Critiques)*. Opgeroepen op September 7, 2020, van CXL: <https://cxl.com/blog/6-user-onboarding-flows/>
- MURAL is a digital workspace for visual collaboration*. (sd). Opgeroepen op September 7, 2020, van mural: <https://www.mural.co/>
- Nguyen, Q. (2013, Maart 19). *The RITE Way to Prototype*. Opgeroepen op September 20, 2020, van UXMag: <https://uxmag.com/articles/the-rite-way-to-prototype>
- Nielsen, J. (2012, Juni 3). *How Many Test Users in a Usability Study?* Opgeroepen op September 20, 2020, van NNGroup: <https://www.nngroup.com/articles/how-many-test-users/>
- Parsons, J. (2020, Januari 14). *Which are some most secure encryption algorithms in the world?* Opgeroepen op Oktober 1, 2020, van Quora: <https://www.quora.com/Which-are-some-most-secure-encryption-algorithms-in-the-world>
- Penney, M. (2017, Juni 16). *What is HTTPS and Why Is It Important?* Opgeroepen op Juni 9, 2020, van sessions: <https://www.sessions.edu/notes-on-design/what-is-https-and-why-is-it-important/#:~:text=HTTPS%20is%20valuable%20because%20it,will%20feel%20safer%20shopping%20there.>
- Plass, D. (2020, Mei 29). *Het DOT-framework*. Opgehaald van wikiwijs: https://maken.wikiwijs.nl/127721/DOT_framework
- Psheychko, V. (sd). *Getting Acquainted – A Guide to User Onboarding Flows*. Opgeroepen op September 7, 2020, van toptal: <https://www.toptal.com/designers/ux/user-onboarding-flow>

Saltis, S. (2020, Mei 19). *GDPR Explained In 5 Minutes: Everything You Need to Know*. Opgeroepen op September 17, 2020, van CoreDNA: <https://www.coredna.com/blogs/general-data-protection-regulation>

Socket.io documentation. (sd). Opgeroepen op Juni 9, 2020, van Socket.io: <https://socket.io/docs/>

Sticky load balancing. (sd). Opgeroepen op Juni 9, 2020, van Socket.io: <https://socket.io/docs/using-multiple-nodes/>

The online collaborative whiteboard platform to. (2020, September 7). Opgehaald van miro: <https://miro.com/>

Vallet, V. (2020, Maart 11). *Node.js and Websockets best practices checklist*. Opgeroepen op Juni 7, 2020, van Medium: <https://medium.com/voodoo-engineering/websockets-on-production-with-node-js-bdc82d07bb9f>

Variance. (2020, Januari 21). *How to secure your MEAN stack app for the future?* Opgeroepen op Mei 29, 2020, van Variance: <https://blog.varianceinfotech.com/2020/01/21/how-to-secure-your-mean-stack-app-for-the-future/>

WebSocket Security. (2020, April 9). Opgeroepen op Juni 9, 2020, van heroku: <https://devcenter.heroku.com/articles/websocket-security>

Welcome to Artillery Docs! (2020, Juni 6). Opgeroepen op Augustus 24, 2020, van Artillery: <https://artillery.io/docs/guides/overview/welcome.html#Stay-in-touch>

11 Bijlages

11.1 Requirements

Nummering	Onderwerp
1.0	Beveiliging
2.0 - 2.6	Online collaboratie
3.0 - 3.2	Onboarding
4.0 - 4.1	Account management
5.0 - 5.7	Workspaces
6.0 - 6.4	Kaarten
7.0 - 7.2	Hints en vragen
8.0	Styling
9.0	Kwaliteit

Tabel 13 Nummering requirements

11.1.1 Must

Deze requirements zijn een must requirement omdat deze requirements nodig zijn om tot een minimal viable product te komen.

#	Requirements	F/NF
1.0	De beveiliging van de applicatie moet in orde zijn volgens de best practices (Au-Yeung, 2020) (Variance, 2020).	NF
2.0	De eigenaar van de workspace moet andere gebruikers kunnen uitnodigen, zodat ze online samen kunnen werken aan een workspace.	F
2.1	De eigenaar van de workspace moet andere gebruikers kunnen kicken uit een workspace, zodat hij eventueel toegang kan weigeren waar nodig is.	F
2.2	De gebruikers moeten kunnen zien welke gebruikers actief zijn in de workspace, zodat ze weten wie er aan de projectaanpak werken.	F
2.3	De workspace moet realtime geupdate worden als er veranderingen gemaakt worden, zodat de gebruikers realtime kunnen samenwerken.	F
4.0	De gebruiker moet in kunnen loggen, zodat ze toegang krijgen tot de workspaces.	F
4.1	De gebruiker moet een account kunnen registreren en verifiëren, zodat ze kunnen inloggen om toegang te krijgen tot de workspaces.	F
5.0	De gebruiker moet een lijst met workspaces kunnen zien, zodat diegene weet tot welke workspaces zij toegang hebben.	F
5.1	De gebruiker moet een workspace kunnen kiezen uit een lijst, zodat ze de workspace kunnen openen.	F
5.2	De gebruiker moet een nieuwe workspace kunnen aanmaken, zodat ze een nieuwe projectaanpak uit kunnen werken.	F
5.3	De gebruiker moet een workspace kunnen verwijderen, zodat ze overzicht kunnen bewaren in de lijst.	F
8.0	De applicatie interface moet voldoen aan de guidelines van material design, met google drive als inspiratie, zodat het intuïtief is, omdat het een van de meest gebruikte style guides is en dat gebruikers er makkelijk mee om kunnen gaan.	NF

Tabel 14 de must requirements

11.1.2 Should

Deze requirements zijn een should, omdat ze wel belangrijk zijn maar niet benodigd zijn voor een minimal viable product.

#	Requirements	F/NF
2.4	Het platform moet visueel onderscheid maken tussen de gebruikers, wanneer de gebruiker een actie uitvoert, zodat andere gebruikers kunnen zien wie er met een bezig is met een bepaald object.	F
3.0	De gebruiker moet het platform zelfstandig kunnen gebruiken.	NF
3.1	Het platform moet een onboarding process bevatten die getoond wordt aan nieuwe gebruikers, zodat de gebruiker begrijpt hoe het platform werkt.	F
3.2	Als gebruiker wil ik het onboarding proces opnieuw starten, zodat de gebruiker het beter kan begrijpen als dat nodig is.	F
5.4	Als gebruiker wil ik dat als ik kaarten slepen dat ze makkelijk te alignen zijn, zodat het er goed uit ziet.(onderliggend grid)	F
5.6	Als gebruiker wil een kopie kunnen maken van een workspace, zodat ik een versie kan bewaren.	F
5.7	Als gebruiker wil ik een inspiratietoolbar kunnen zien op het scherm, zodat ik makkelijk inspiratie op kan doen en makkelijk kaarten kan slepen van de inspiratietoolbar naar het bord.	F

Tabel 15 de should requirements

11.1.3 Could

Deze requirements zijn een could, omdat ze extra functionaliteit zullen bieden aan de minimal viable product.

#	Requirements	F/NF
5.5	Als gebruiker wil ik toegang tot een specifieke workspace(waar ik als gebruiker aan ben toegevoegd) door middel van een link, zodat ik snel toegang kan krijgen en makkelijk kan delen met andere deelnemers.	F
6.0	De gebruiker moet een kaart als favoriet kunnen kiezen, zodat het proces van het projectaanpak maken sneller zal gaan.	F
6.1	De gebruiker moet alle favorieten kaarten uit een lijst kunnen filteren, zodat ze hun favoriete kaarten makkelijk kunnen gebruiken tijdens het maken van een projectaanpak.	F
6.2	Als gebruiker wil ik als een methode of stepping stone nog niet bestaat, die kunnen aanmaken. Zodat mijn projectaanpak compleet online staat als er speciale gevallen zijn. (binnen het overzicht van methode of stepping stone)	F
6.3	Als gebruiker wil ik mijn custom methodes zien in de methode lijst, zodat ik de custom methodes in hun bijbehorende lijst kan terugvinden.	F
6.4	Als gebruiker wil ik mijn custom stepping stones zien in de stepping stone lijst, zodat ik de custom stepping stones in de bijbehorende lijst kan terugvinden	F
7.0	Als gebruiker wil ik hints kunnen zien voor kaarten, zodat die mij verder kunnen helpen met het uitwerken van de projectaanpak.	F

Tabel 16 de could requirements

11.1.4 Wont

Deze requirements zijn een wont, om aan een afbakening van het project aan te tonen.

#	Requirements	F/NF
2.5	Als gebruiker wil ik de historie van de workspace in kunnen kijken, zodat diegene eventueel naar een oud punt terug kan.	F
2.6	Als gebruiker wil ik de historie van de workspace terug kunnen zetten, zodat diegene eventueel naar een oud punt terug kan.	F
7.1	Als gebruiker wil ik vragen kunnen krijgen over bepaalde kaarten, zodat die mij verder kunnen helpen met het aanvullen van mijn projectaanpak, zodat de gebruiker informatie krijgt over onderdelen waar ze misschien nog niet aan dachten (wizard?)	F
7.2	als gebruiker wil ik aan deels geautomatiseerd en aan de hand van een aantal vragen de kwaliteit kunnen meten van mijn projectaanpak, zodat dit mijn projectaanpak helpt verbeteren.	F

Tabel 17 de wont requirements

11.2 Error log

1. Fases kunnen onder de toolbar geplaatst worden, daarna zijn ze niet meer terug te halen.
2. Fases kunnen uit het beeld aan de zijkant geplaatst worden, dit creëert 2 scrollbars.
3. De templateselector wordt automatisch getoond als er geen template gekozen is voor een workspace, maar als er geen templates in de database zit, blijft de dialog zichzelf tonen.
4. Titel wordt onzichtbaar en is onaanpasbaar als er geen tekst ingezet wordt, hetzelfde geldt voor het doel van een project.
5. Als een gebruiker de applicatie uitzoomt en dan een fase aan de andere kant van het bord plaatst, en dan terug zoomt naar 100%. En dan begint te scrollen naar de fase, wordt de toolbar afgekapt.
6. De kaarten hebben een slechte styling en passen zich slecht aan, als een gebruiker in/uitzoomt.
7. De titel wordt afgekapt als je een lange titel intypt, opslaat en dan de pagina ververst, zelfde geldt voor het doel van een project.
8. De onderzoeksvragen worden afgekapt als er een grote vraag ingetypt wordt.
9. Een fase zonder titel, is niet sleepbaar vanaf de blauwe tegel, alleen van de witte tegel.
10. Als er een pijl getrokken wordt van fase naar een andere fase, dan is dit proces herhaalbaar, maar visueel veranderd er niets.
11. Vragen en kaarten die worden toegevoegd aan de spawnlist, worden niet opgeslagen in een workspace en daarvoor verdwijnen alle kaarten uit de spawnlist als je de pagina ververst.
12. Als er een pijl getrokken wordt van een fase naar dezelfde fase, dan crasht het plaatsen van de pijlen, en moet de pagina ververst worden om weer pijlen te kunnen zetten.
13. De huidige templateselector geeft ook een ExpressionChangedAfterItHasBeenCheckedError, als de dialog geopent wordt.
14. De styling van de tekst en de input van de project titel is niet consistent.
15. De styling van de tekst en de input van het project doel is niet consistent.
16. De styling van de tekst en de input van de fase titel is niet consistent.

17. De styling van de tekst en de input van de kaart titel in de detailModal is niet consistent.
18. De vragen maken in de code gebruik van `_id` en de kaarten maken gebruik van `id`.
19. De aanpasbare titel van een project, geeft nu de volgende `nativeElement` of `undefined error`.
20. Als een pijl getrokken wordt van fase naar een andere fase, dan zullen ze over kaarten heen gaan.
21. Als een pijl getrokken wordt van fase naar een andere fase, dan zullen ze over fases heen gaan.
22. Als een pijl getrokken wordt van fase naar een andere fase, dan zullen ze over vragen heen gaan.
23. Als een pijl getrokken wordt van fase naar een andere fase, dan zullen ze over de toolbar heen gaan.
24. Als een pijl getrokken wordt van fase naar een andere fase, dan zullen ze over de scrollbar heen gaan.
25. Als je een fase de naam "title" geeft, wordt de gehele fase blauw en kunnen er geen kaarten in geplaatst worden.
26. Het trekken van een pijl is niet vlekkeloos, als je in het debug menu zit. Loopt erg veel vast
27. Het trekken van een pijl is niet vlekkeloos, als je op firefox aan het werk bent.

11.3 GDPR-onderzoek

Hieronder zullen relevante artikelen van de GDPR wet, te vinden zijn.

- Wees transparant over wat je doet met de data (Artikel 12)
 - Beschrijf in een duidelijke taal wat je met de data doet op het moment van verzamelen.
 - Als iemand hier iets over vraagt, reageer binnen een maand.
 - Het is mogelijk om iemand te weigeren of om geld te vragen, als iemand gekke eisen heeft.
 - Het is mogelijk om iemand te vragen om hun identiteit te bevestigen, als je hier twijfels over hebt.
- Als je iemands persoonlijke of niet persoonlijke data verzameld, maakt dit aan hem duidelijk (Artikel 13/14)
 - Geef de volgende informatie weer als je informatie van een gebruiker verzameld:
 - Contact informatie van het bedrijf en de dataprivacy officer.
 - Beschrijf waarvoor de informatie die je verzameld gebruikt wordt
 - Maak een lijst van de data die je verzameld en in welke categorie deze data valt.
 - Hoe lang de data bewaart zal worden.
 - Hoe contact opgenomen kan worden als er problemen zijn of als er een aanvraag gedaan moet worden
 - Of de data gebruikt wordt voor profilering.
- De gebruiker moet geïnformeerd worden over welke rechten ze hebben met betrekking tot hun eigen data (Artikel 15)
 - De gebruiker heeft het recht om te vragen of er data over hem verzameld wordt, een antwoord is hierop verplicht.
 - Als je data verzameld over de gebruiker, dan moet hem het volgende worden verteld:

- Waarom je het verzameld
- Welke categorieën data het zijn? (Gevoelig, genetische, gezondheid, biometrische, etc.).
- Wie toegang heeft tot deze data
- Hoe lang het bewaard wordt
- Dat ze het recht hebben om de data te veranderen
- Dat ze het recht hebben om de data te verwijderen
- Waar de data van verzameld is
- Dat ze het recht hebben om een klacht in te dienen bij de EU-commissie als ze niet tevreden zijn met het antwoord.
- Toon de data als het niet gevoelig is
- De gebruiker heeft het recht om te vragen of data veranderd kan worden (artikel 16)
 - De applicatie moet het veranderen van de data ondersteunen.
- De gebruiker heeft het recht om te vragen of data verwijderd kan worden (artikel 17)
 - De applicatie moet het verwijderen van de data ondersteunen.
- De gebruiker heeft het recht om te vragen of het data verwerk proces gepauzeerd kan worden (artikel 18)
 - De gebruiker moet een aanvraag in kunnen dienen door middel van een mail als er een legale reden is om de data van de gebruiker niet te verwerken.
- Als er massa correcties aan de data van de gebruikers worden gedaan, dan moeten ze hierover geïnformeerd worden (artikel 19)
 - Er moet een mail gestuurd worden naar de gebruiker als hun data aangepast wordt.
- De gebruiker heeft het recht om de data te exporteren in een goed formaat (artikel 20)
 - De applicatie moet het exporteren van data ondersteunen.
 - De data moet leesbaar zijn voor computers (CSV, XLS, XLM of JSON)
- De gebruiker heeft het recht om zich af te melden voor profilering en filtering van informatie (artikel 21)
 - Heb een afmeldsysteem zodat gebruikers zich kunnen afmelden voor marketing, profileren etc.
 - Hoeft alleen als er gebruik van profileren of filtering gemaakt wordt.
- De gebruiker heeft het recht om te vragen of een mens een beslissing kan maken over de gebruiker (artikel 22)
 - Het een afmeldsysteem zodat gebruikers zich kunnen afmelden voor beslissingen die gemaakt zijn door computers.
 - Hoeft alleen als er gebruik van beslissingen door computers gemaakt wordt.
- Het volgende is minimaal nodig om je data veilig te houden (artikel 32)
 - Persoonlijke data geencrypt opslaan.
 - Data back-uppen
 - Regelmatig de veiligheid testen
 - Let goed op datalekken en de gevolgen
- Een gebruiker moet geïnformeerd worden wanneer een datalek plaatsvindt (artikel 34)
 - Moet binnen 72 uur gebeuren

11.4 Benchmark resultaat

Hieronder zullen de resultaten van de benchmark test te vinden zijn.

Testscenario	Totaalaantal berichten ontvangen en verstuurd per seconde
100 actieve sockets	3600
1000 actieve sockets	3301
5000 actieve sockets	3226
10000 actieve sockets	1019
1000 actieve en 9000 inactieve sockets	2882
2500 actieve en 7500 inactieve sockets	3049
5000 actieve en 5000 inactieve sockets	3228
7500 actieve en 2500 inactieve sockets	2620

Testscenario	Totaalaantal berichten ontvangen en verstuurd per seconde per socket
100 actieve sockets	177
1000 actieve sockets	16
5000 actieve sockets	2.2
10000 actieve sockets	0.5
1000 actieve en 9000 inactieve sockets	15
2500 actieve en 7500 inactieve sockets	6.5
5000 actieve en 5000 inactieve sockets	3
7500 actieve en 2500 inactieve sockets	1.3

Testscenario	Gemiddeld CPU gebruik	Gemiddeld Memory gebruik
100 actieve sockets	6.9%	1%
1000 actieve sockets	12.8%	4.4%
5000 actieve sockets	16.1%	13%
10000 actieve sockets	25.2%	20.1%
1000 actieve en 9000 inactieve sockets	16.2%	9.9%
2500 actieve en 7500 inactieve sockets	17.2%	13.4%
5000 actieve en 5000 inactieve sockets	18.3%	16.9%
7500 actieve en 2500 inactieve sockets	20.6%	20.8%

11.5 Loadtest resultaat

Hieronder zullen de resultaten te vinden zijn voor de loadtest.

Voor elke test wordt de volgende informatie bijgehouden:

- Min
 - De kortste responseduur in milliseconde
- Mediaan
 - De gemiddelde responseduur in milliseconde
- Max
 - De hoogste responseduur in milliseconde
- 95%
 - De wachttijd van 95% van alle berichten in milliseconde
- 99%
 - De wachttijd van 99% van alle berichten in milliseconde
- Count
 - Een percentage van geslaagde berichten

Test naam	min	mediaan	max	95%	99%	count
Join-1	0.1 msec	0.6 msec	1.1 msec	n.v.t.	n.v.t.	100%
Join-10	0.1 msec	0.1 msec	1.1 msec	1.1 msec	1.1 msec	100%
Join-100	0.1 msec	0.1 msec	0.7 msec	0.1 msec	0.7 msec	100%
Join-1000	0.1 msec	0.1 msec	1.8 msec	0.1 msec	0.2 msec	100%
move-1	0.1 msec	0.6 msec	1.1 msec	n.v.t.	n.v.t.	100%
move-10	0.1 msec	0.2 msec	1.5 msec	0.8 msec	1.5 msec	100%
move-100	0.1 msec	0.1 msec	1.3 msec	0.3 msec	1.2 msec	100%
move-1000	0.1 msec	0.1 msec	1.6 msec	0.2 msec	0.4 msec	100%
setEffect-2	0.1 msec	0.4 msec	1.4 msec	1.4 msec	1.4 msec	100%
Wait-60	0.1 msec	1.2 msec	1.3 msec	1.3 msec	1.3 msec	100%
Wait-600	0.1 msec	1.1 msec	1.7 msec	1.7 msec	1.7 msec	100%
Wait-1800	0.1 msec	1.3 msec	1.9 msec	1.9 msec	1.9 msec	100%
Wait-3600	0.1 msec	1.1 msec	1.1 msec	1.1 msec	1.1 msec	100%

11.6 Gebruikerstest online collaboratie resultaat

De testresultaten zullen opgesplitst worden in positieve punten en verbeterpunten:

Positief:

- Veel kaarten beschikbaar die ze kunnen gebruiken om een project samen te stellen.
- Projecten kunnen gedeeld worden met meerdere gebruikers, hierdoor kunnen meerdere gebruikers tegelijkertijd aan het projectaanpak werken.
- Het was een voordeel dat eerst de deelproducten geselecteerd kunnen worden en dat daarna de fases ingericht kunnen worden.
- Er werd ook aangegeven dat de applicatie zeker vaker gebruikt zal worden voor het maken van een project aanpak.

Verbeterpunten:

- Het toevoegen van kaarten en bouwelement (fases, pijlen, vragen) vergt nog enige uitleg.
- Jammer dat de pijlen niet verwijderd kunnen worden.
- Er mist een beschrijving bij de kaarten, waardoor het soms nogal onduidelijk was, ze moesten nu namelijk externe bronnen gebruiken om de kaarten te begrijpen.
- Verder werkt het uitnodigen nog niet helemaal correct, ze gaven aan soms geen mail te ontvangen.
- Ook vonden ze het erg onhandig dat ze kaarten naar de prullenbak moeten slepen.

11.7 Gebruikerstest onboarding resultaat

De eerste gebruiker die getest werd heet Nick Lohuis en is een tweedejaars software engineering student.

Scenario #1: volgen onboarding:

Hij gaf aan dat de teksten over het algemeen goede en duidelijke informatie geven. Hij miste alleen informatie op de beginpagina van de onboarding over wat het doel is van de applicatie.

Ook gaf hij aan een fout gevonden te hebben op de workspace onboarding pagina.

Hij vond ook dat het verwarrend was dat er een wachttijd zit tijdens het openen van de sidebar, hij dacht namelijk dat de applicatie gecrasht was.

Scenario #2: voorbeeld uitleggen:

Hij kon mij goed uitleggen wat de fases, pijlen en vragen waren en waarvoor ze dienen. Hij begreep niet dat de kleur van de kaarten bij een kaartengroep (methode/stepping stones) hoorde.

Ook gaf hij aan dat hij dacht dat de research methodes andere type fases waren en niet door had dat dit de kaarten waren die in fases horen.

Hij kon mij ook niet de individuele kaarten uitleggen en hij gaf aan dat hij in de applicatie wel informatie over de kaarten kan gebruiken.

Scenario #3: voorbeeld uitbreiden:

Het uitbreiden aan de hand van de uitleg ging aardig goed, op de pijlen na. Hij vond het systeem van hoe de pijlen geplaatst worden heel lastig en maar uiteindelijke ging dit ook wel goed.

De tweede gebruiker die getest werd heet Niels ter haar, hij is ook een tweedejaars software engineering student.

Scenario #1: volgen onboarding:

Hij vond dat de teksten over het algemeen erg duidelijk waren, maar hij gaf ook aan dat er een fout zat in de workspace onboarding pagina.

Hij gaf ook aan dat hij het erg fijn vond dat er bij de research methodes bij stond welke doel de verschillende type kaarten heeft. Omdat dit het beeld voor hem erg duidelijk maakte.

Ook gaf hij nog aan, dat er wel een stop de onboarding knop is. Maar dat er nog nergens uitleg overstaat dat je het onboarding proces kunt stoppen.

Scenario #2: voorbeeld uitleggen:

Hij gaf aan het net leek dat de pijlen verbonden waren tussen de kaarten in plaats van tussen de fases.

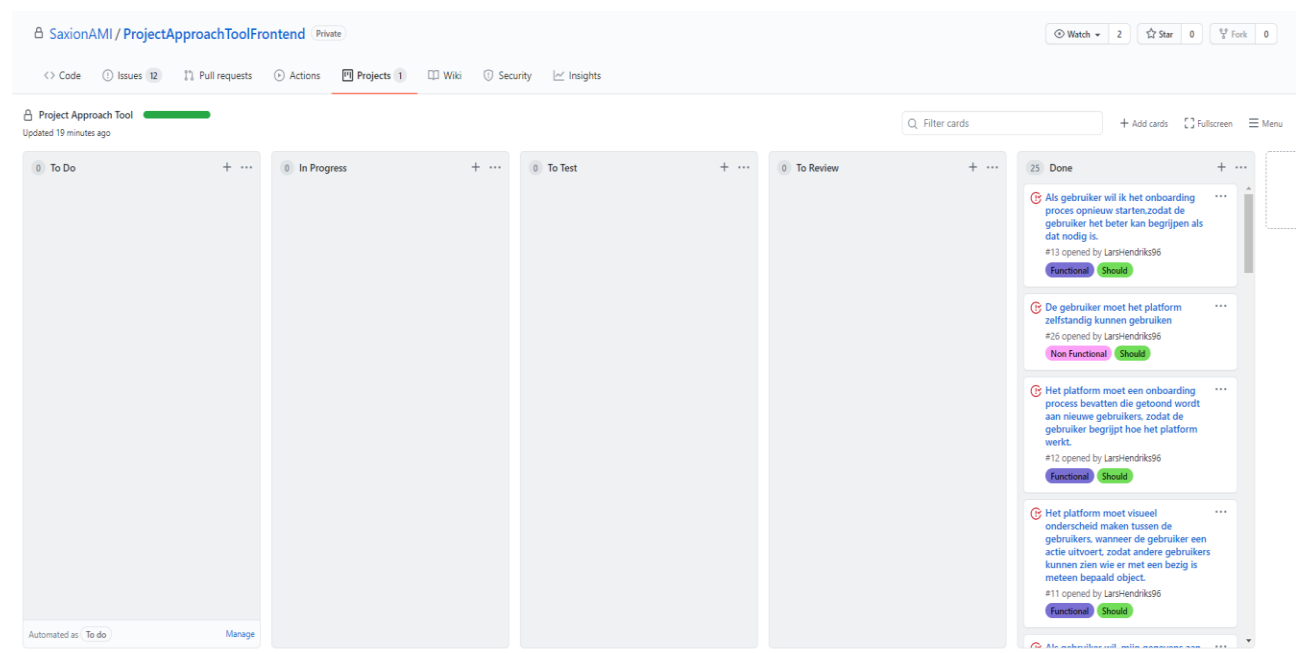
Ook kon hij mij aardig goed vertellen hoe het voorbeeld in elkaar zat, hij kon mij zelfs sommige kaarten uitleggen die in het voorbeeld waren gebruikt. Hij had geen problemen met het plaatsen van objecten, ook niet met het plaatsen van pijlen.

Hij gaf wel aan dat hij informatie mist bij de kaarten, omdat velen gewoon nog niet duidelijk zijn.

Scenario #3: voorbeeld uitbreiden:

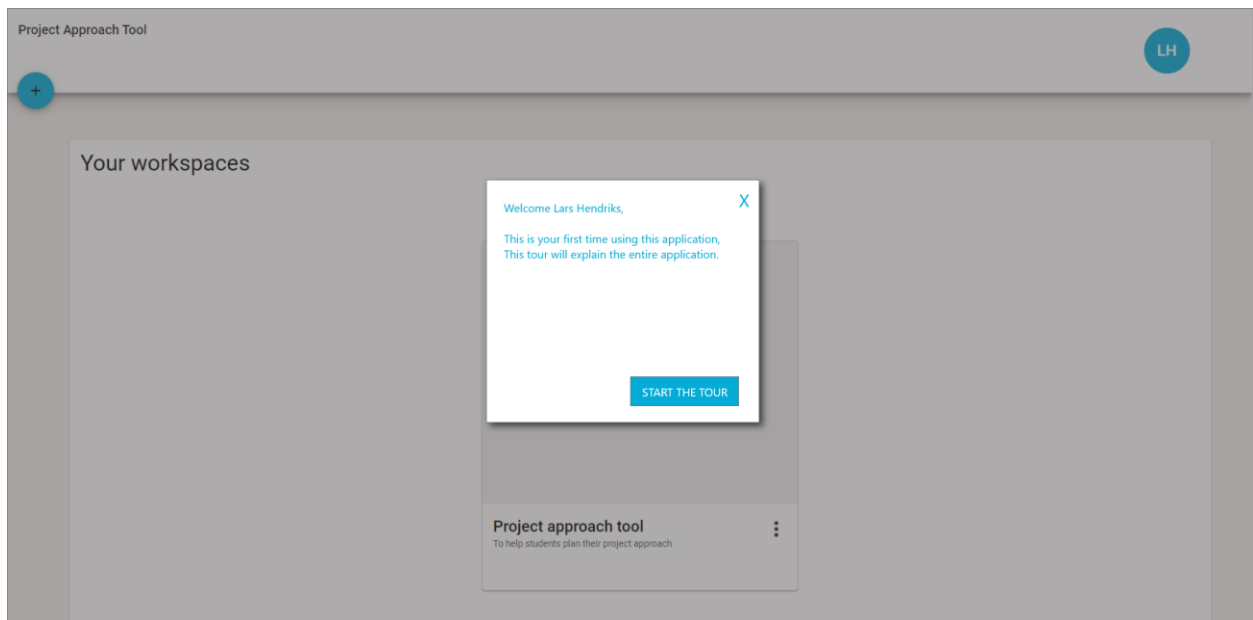
Het ging heel goed, hij had totaal geen problemen met het uitbreiden van het voorbeeld. Ook niet met het plaatsen van de pijlen.

11.8 Scrumboard

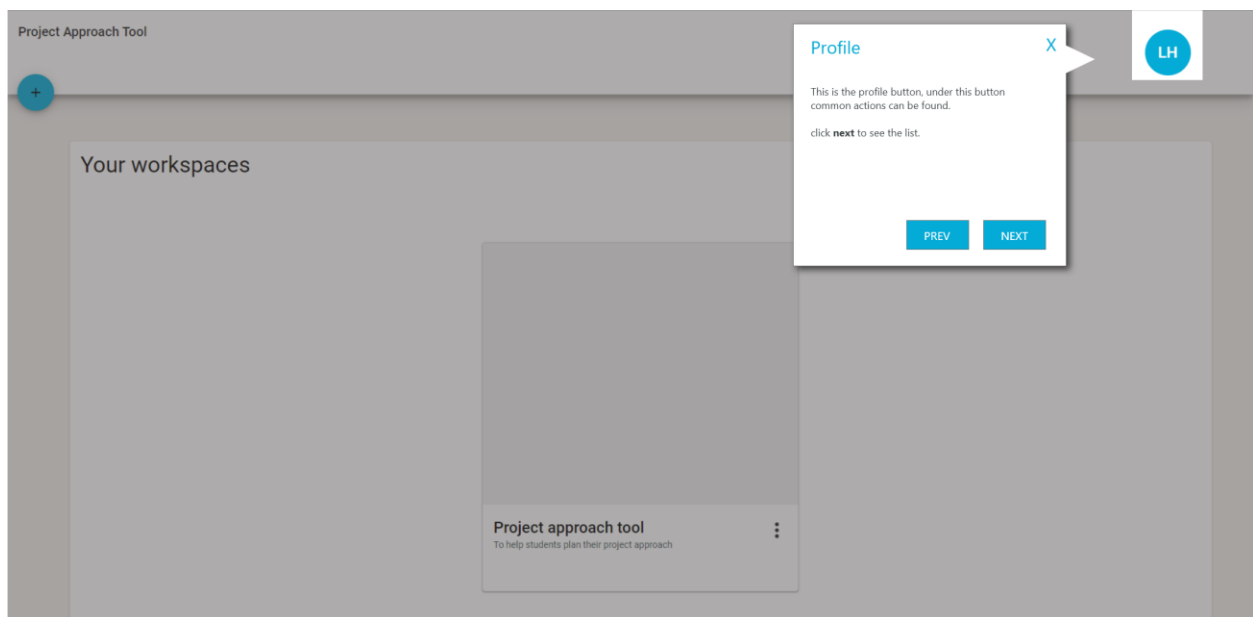


11.9 mock-ups onboarding

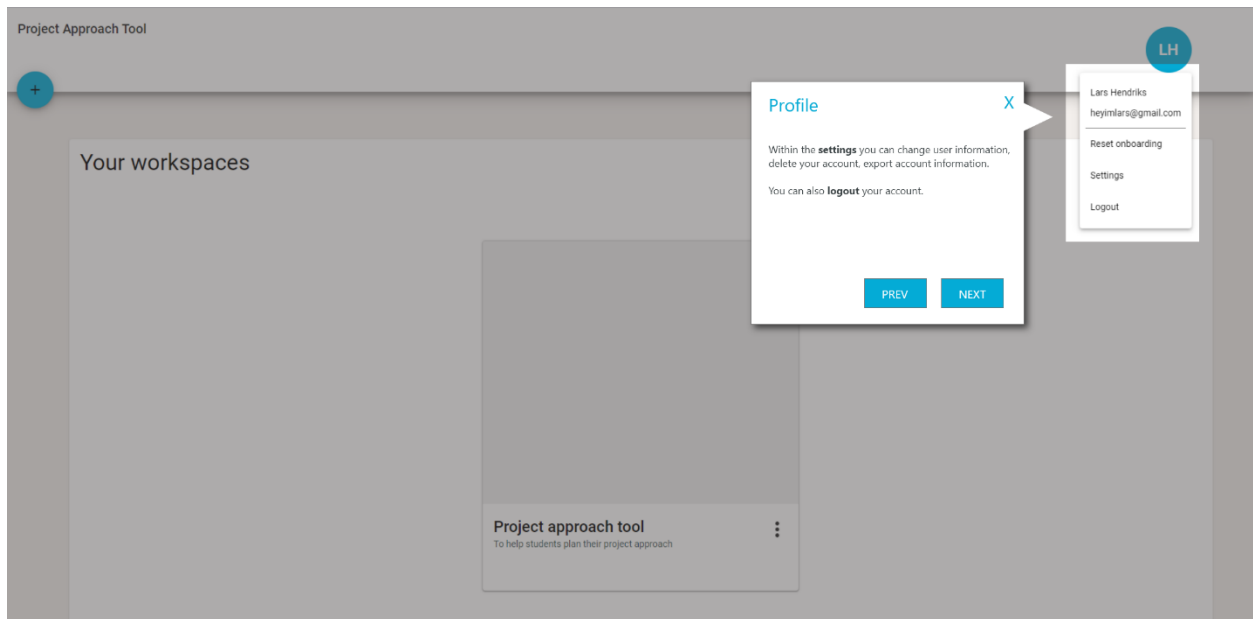
Begin onboarding



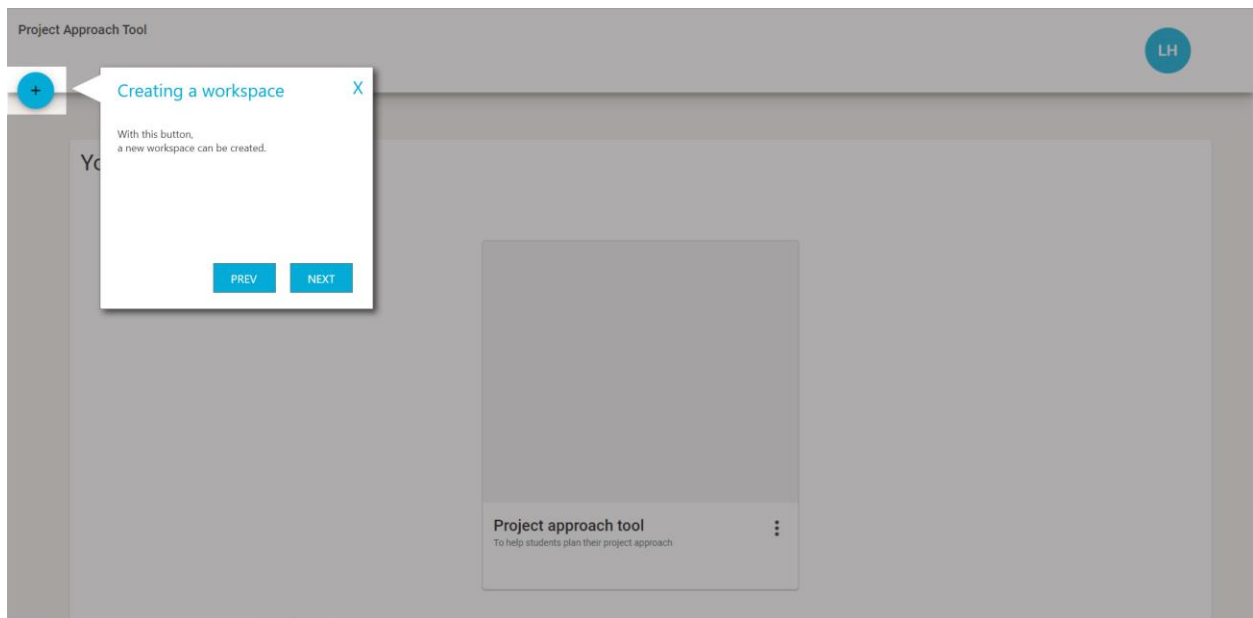
Profiel



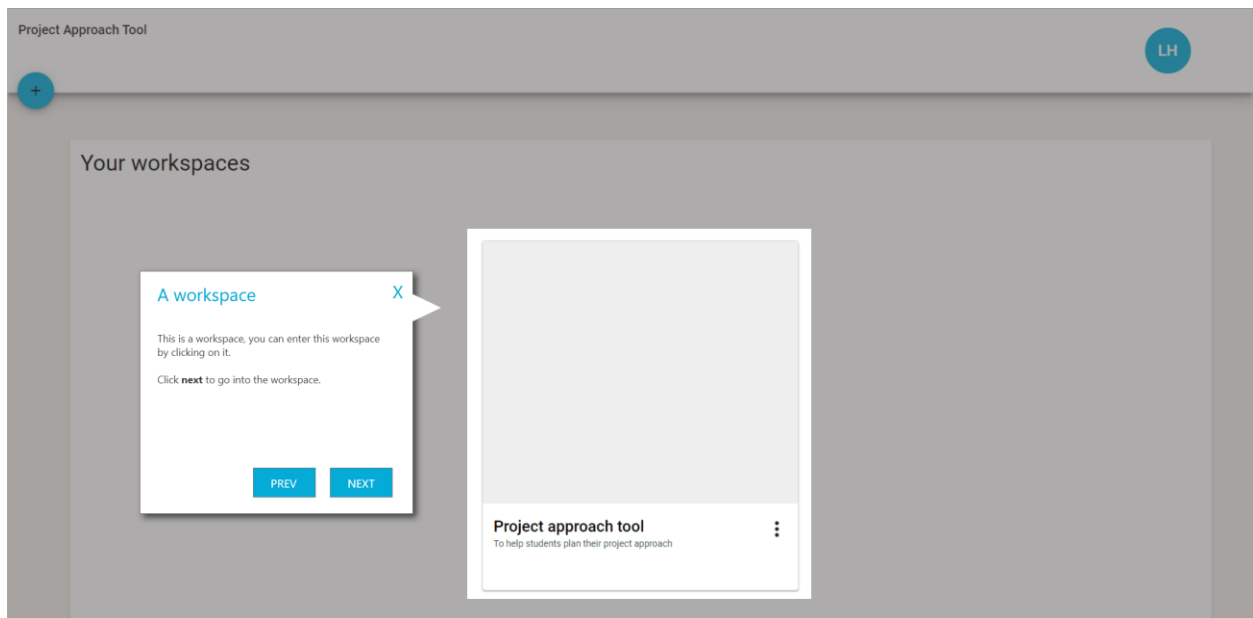
Profiel menu



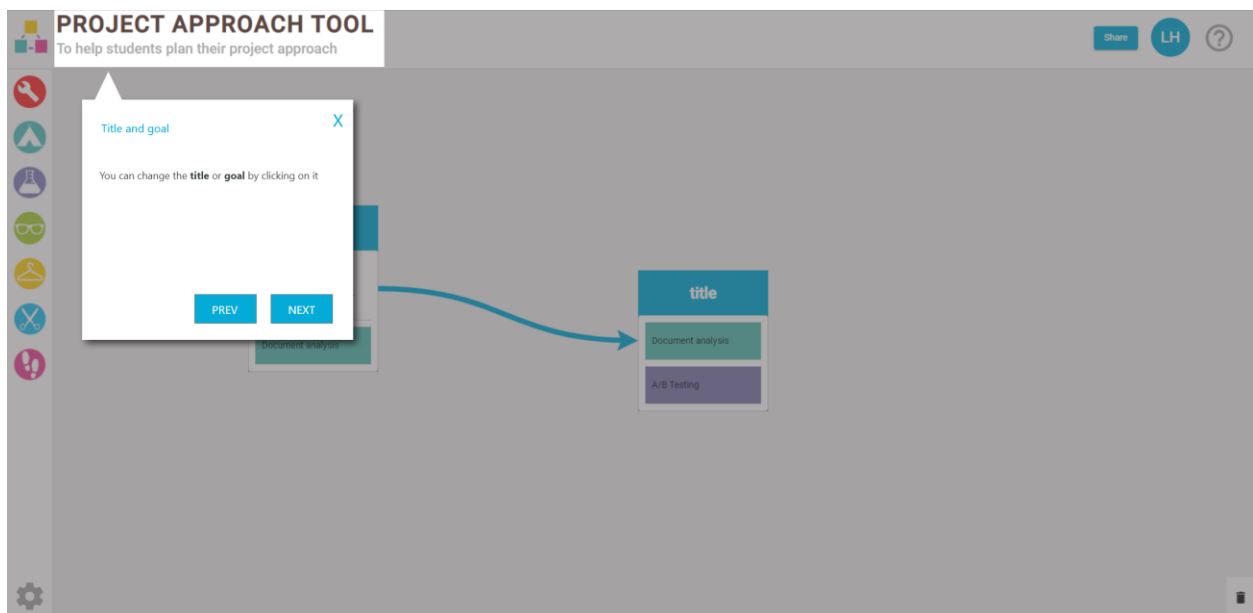
Workspace aanmaken



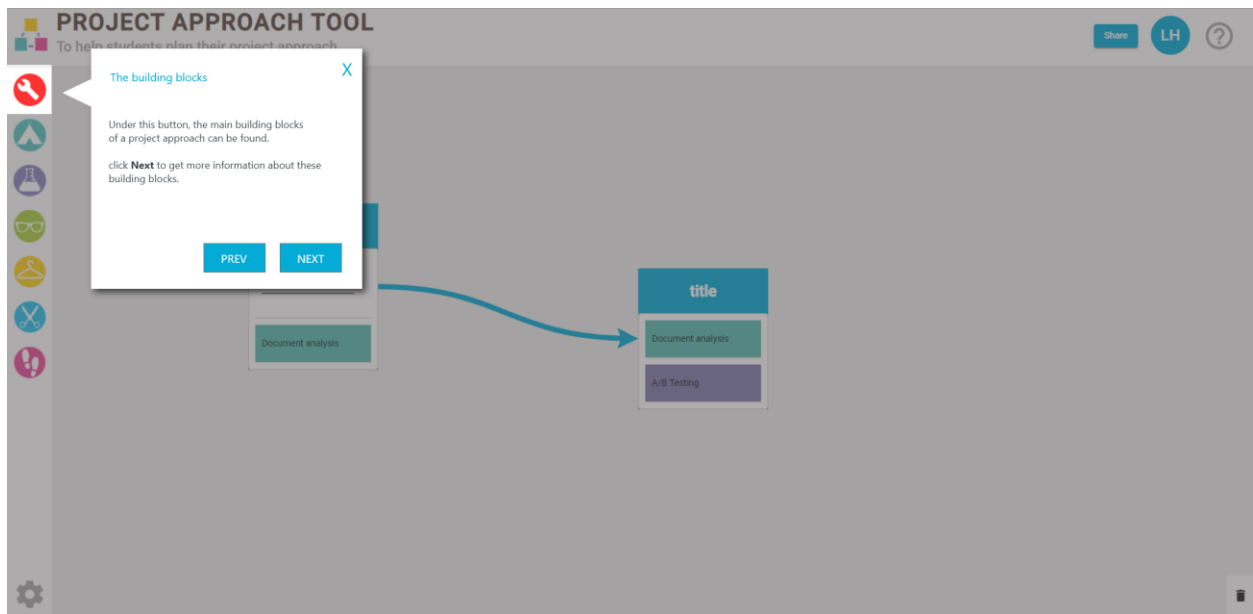
Workspace



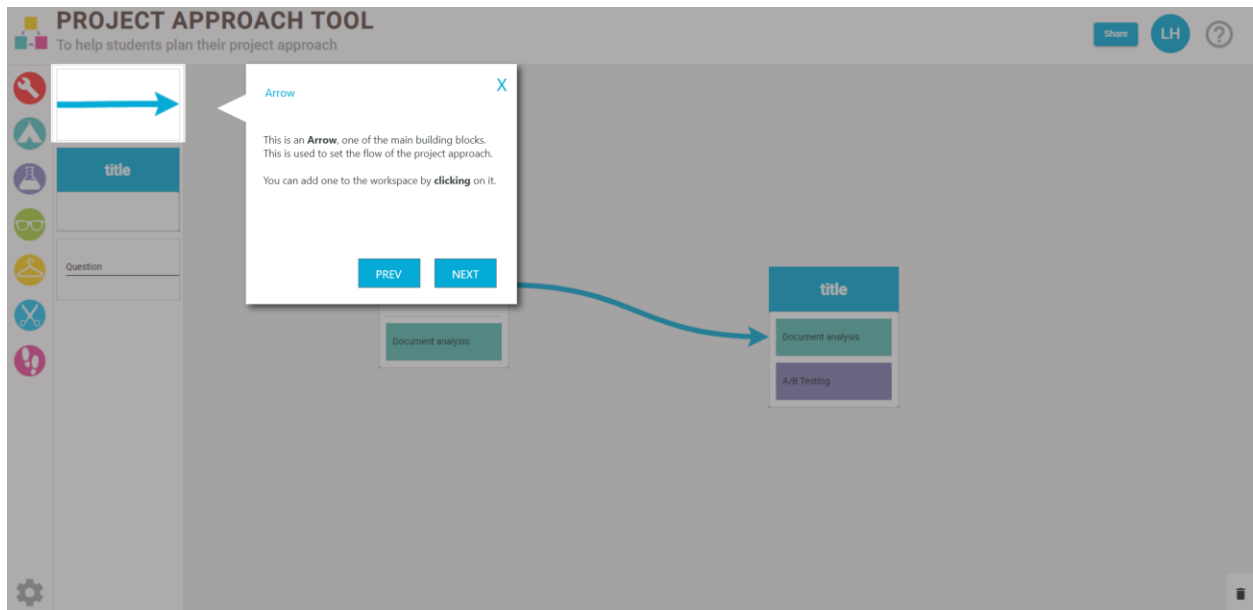
Titel en doel



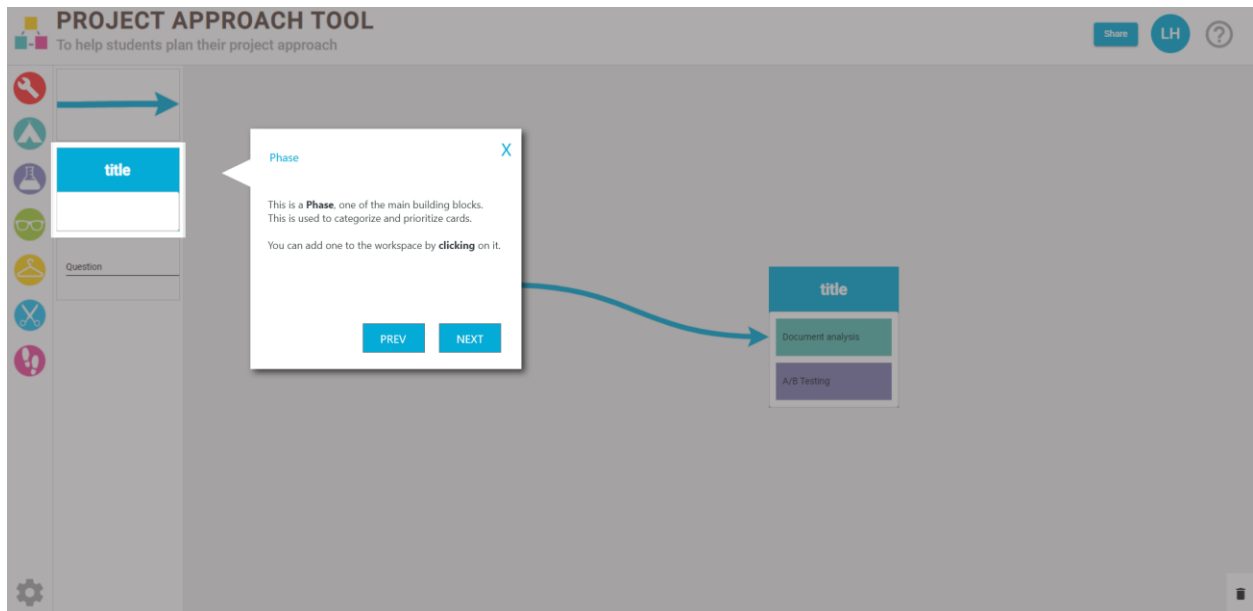
Bouwstenen



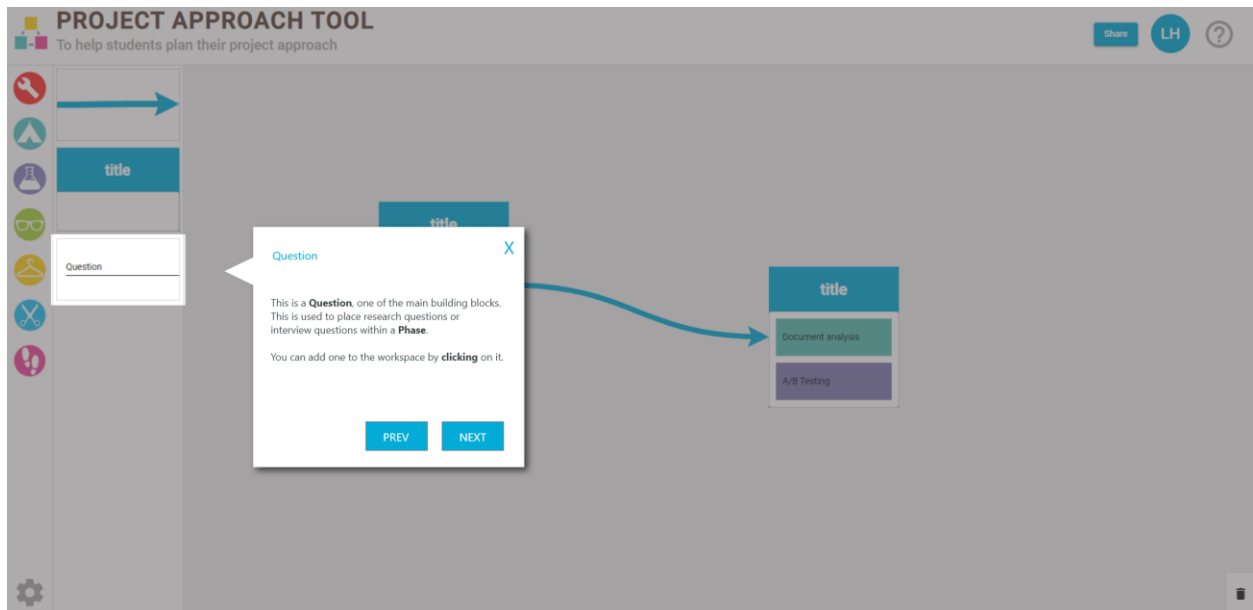
Pijlen



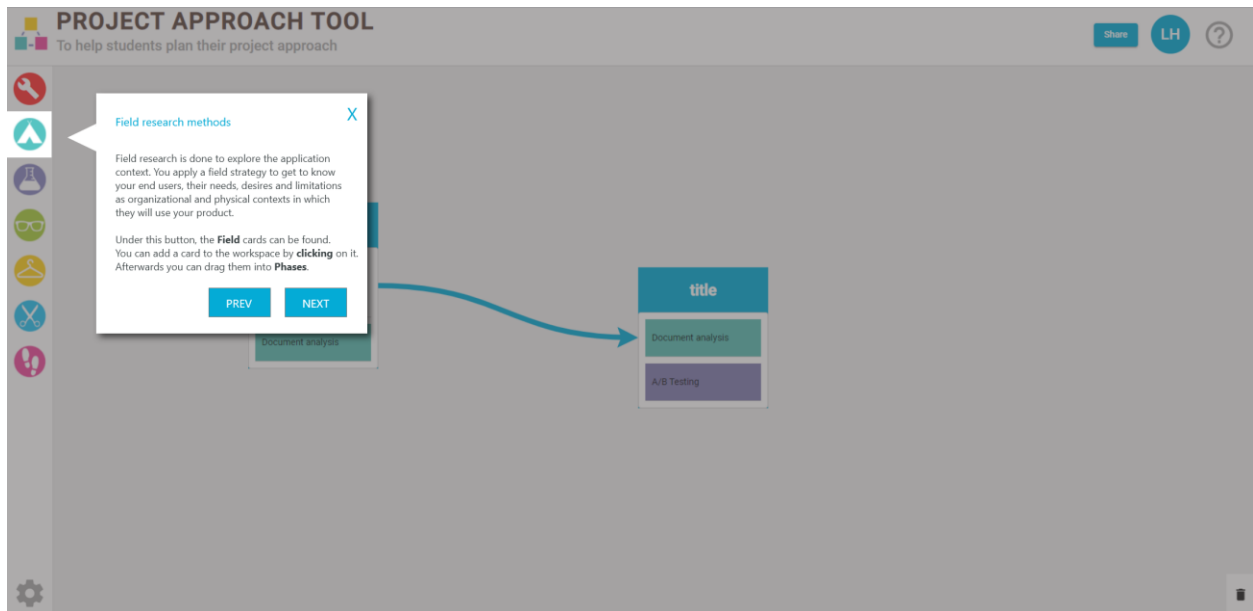
Fases



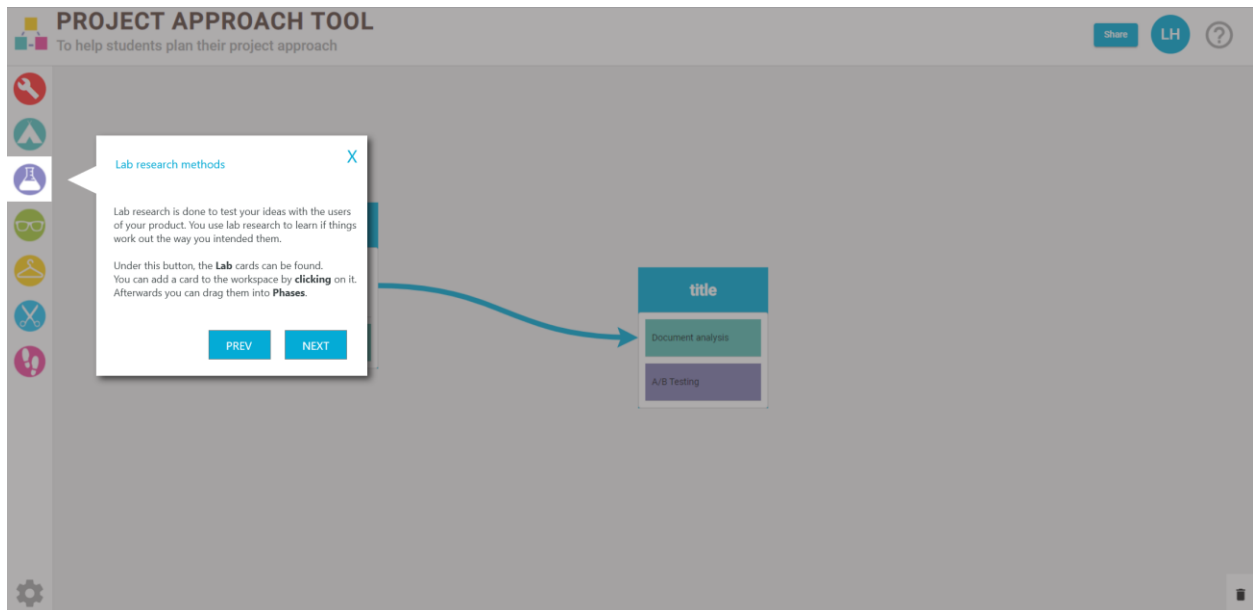
Vragen



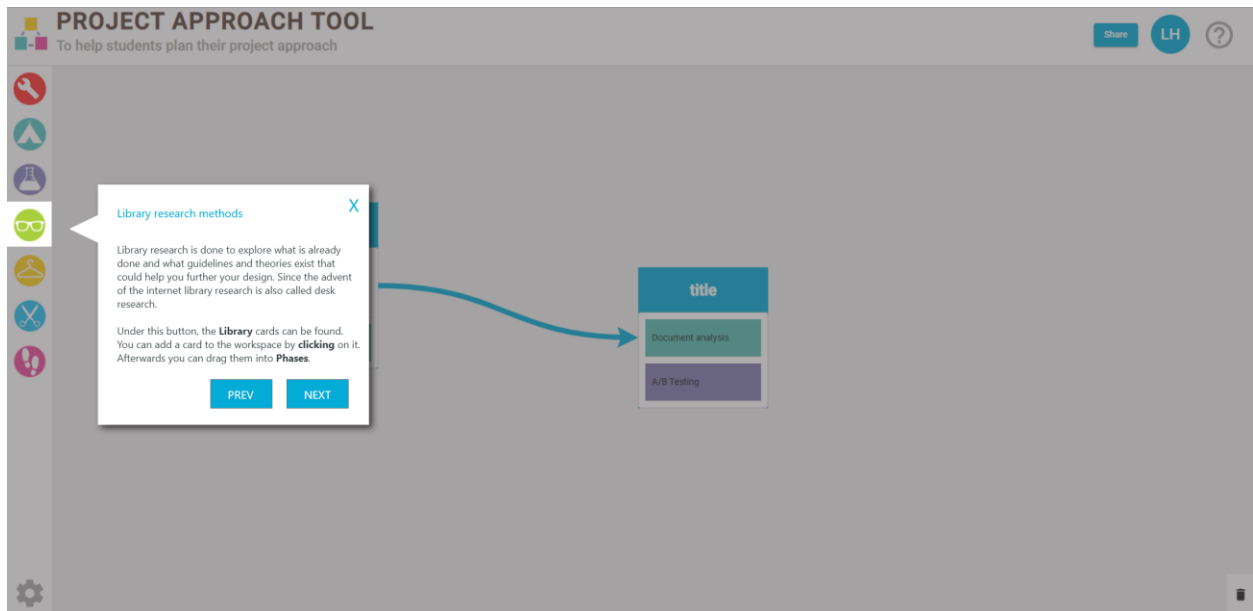
Veld methodes



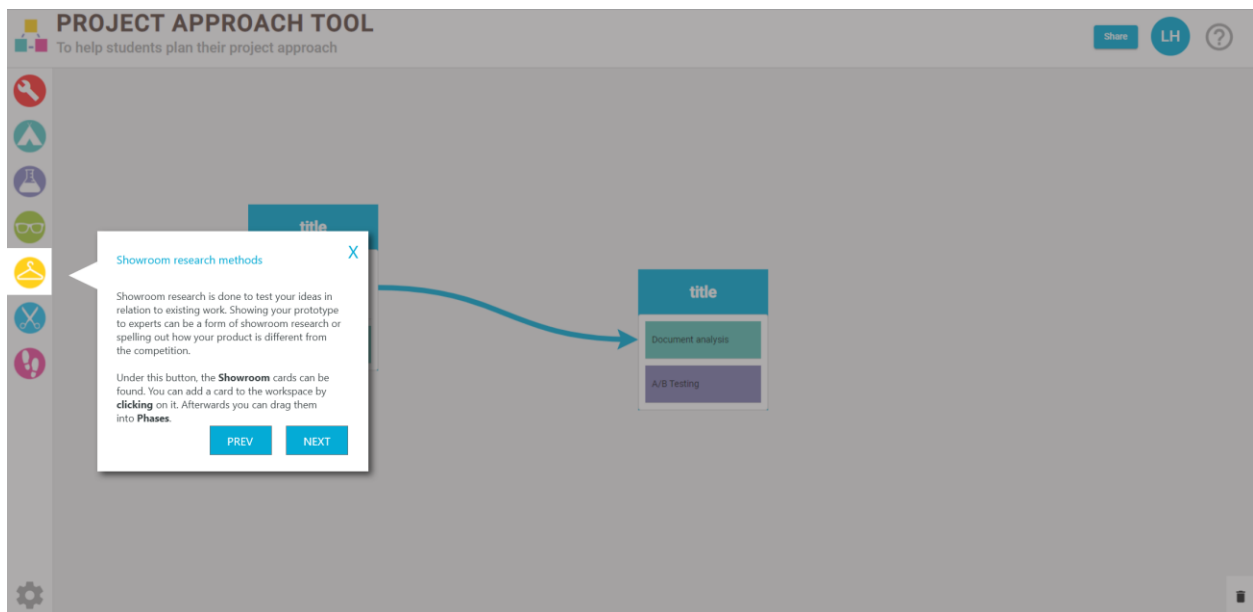
Lab methodes



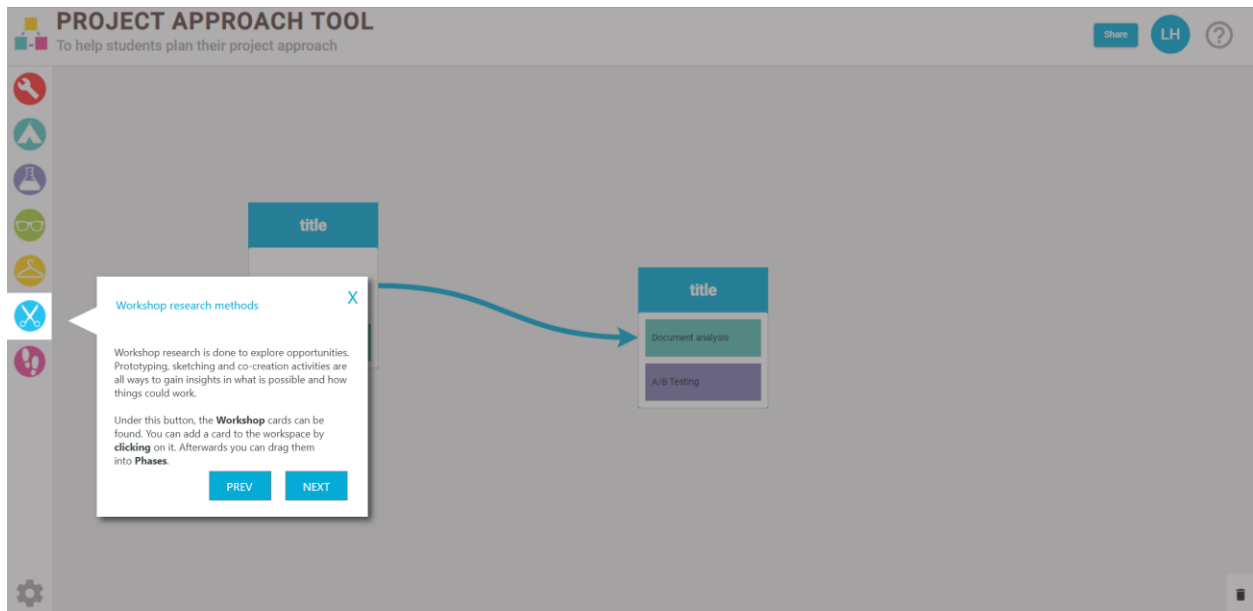
Bibliotheek methodes



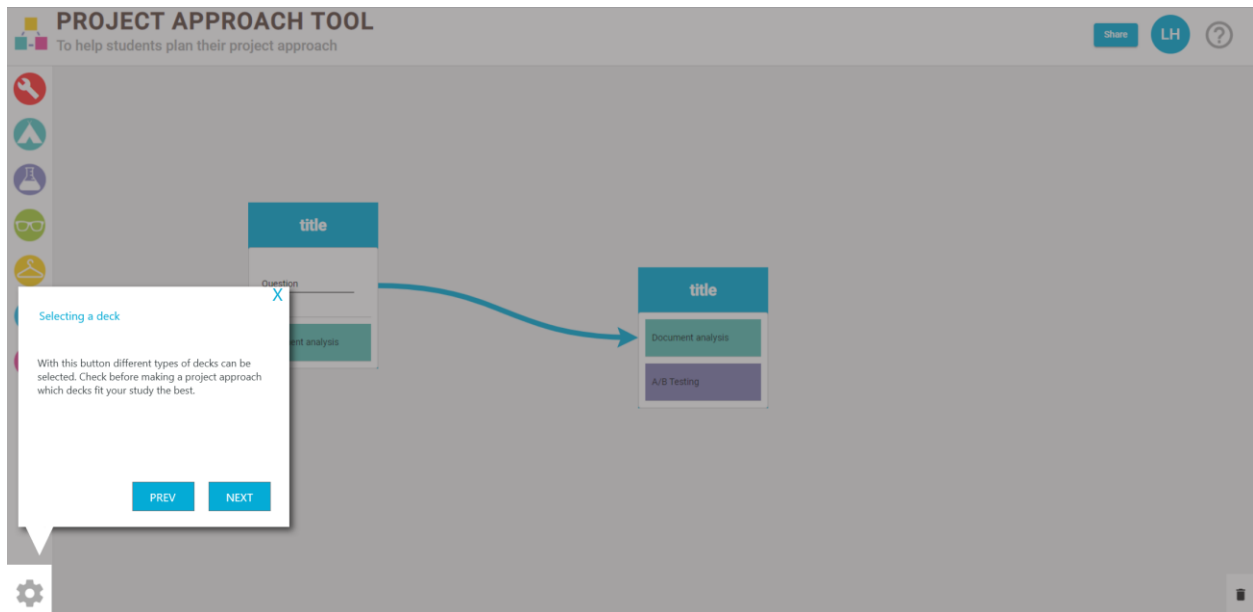
Showroom methodes



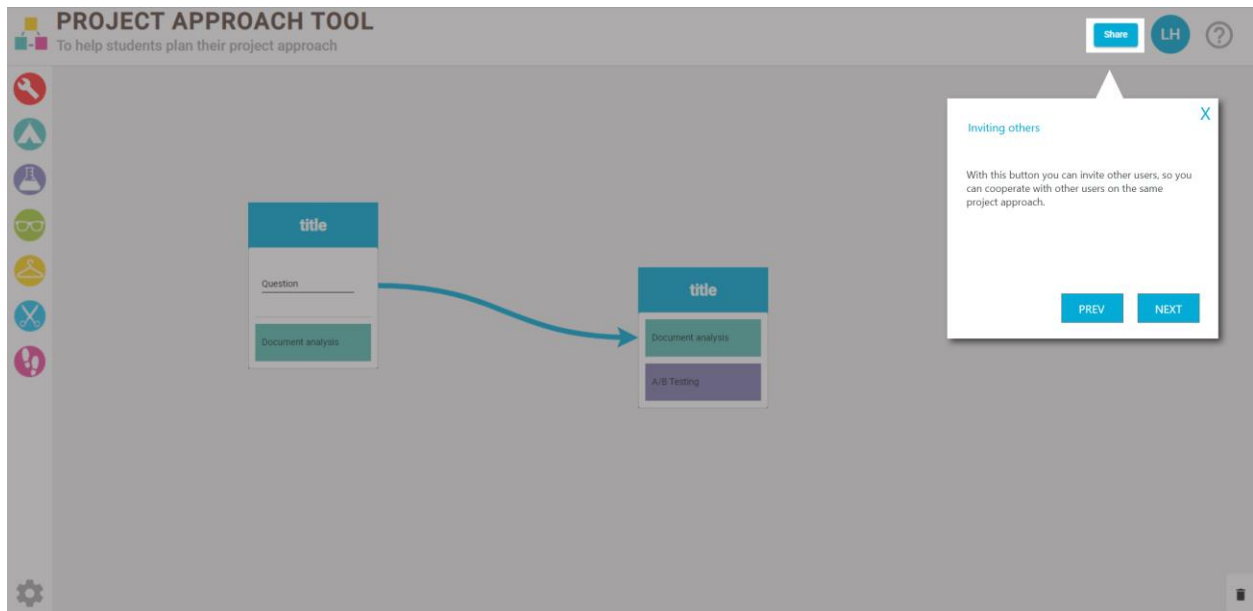
Workshop methodes



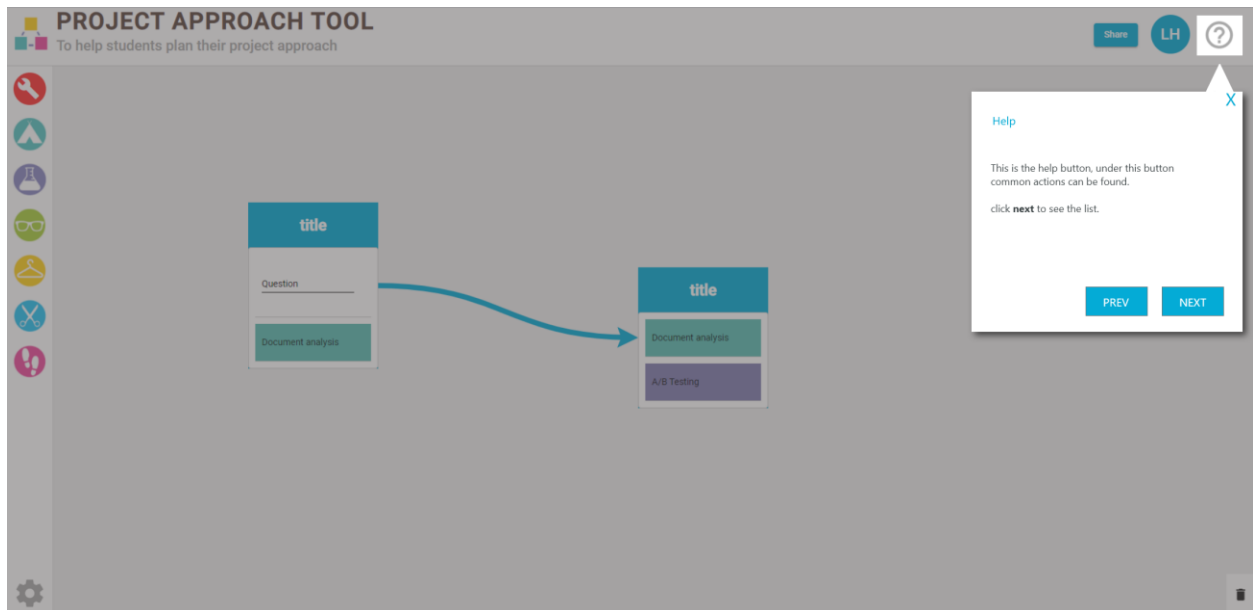
Decks



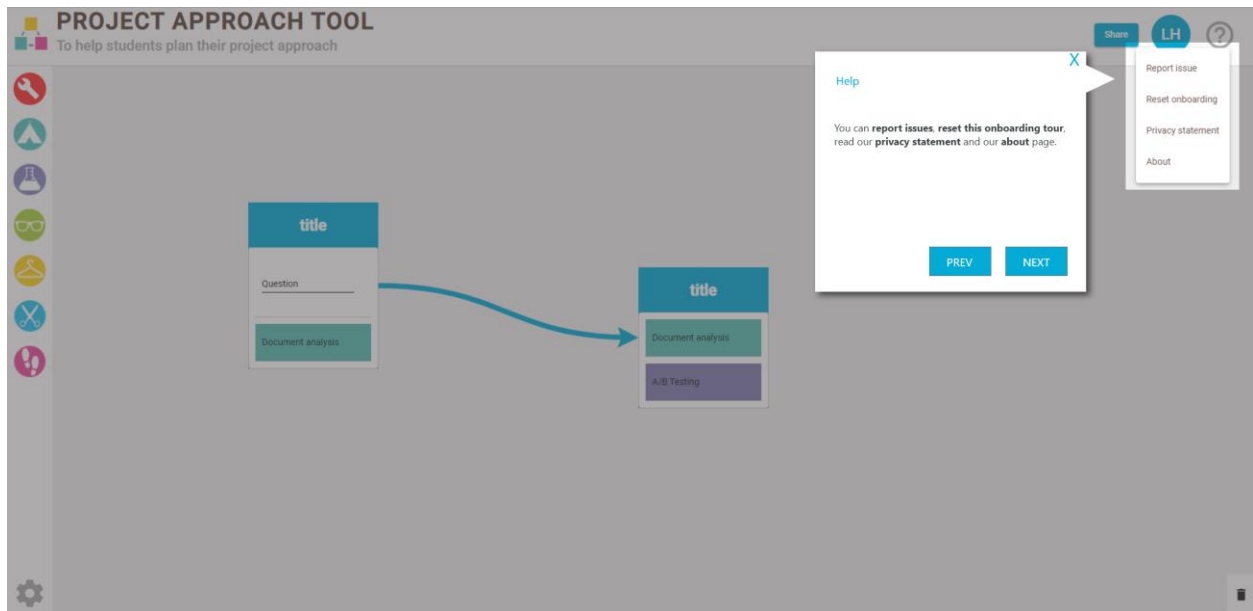
Mensen uitnodigen



Help



Help menu



Einde onboarding

