



Leeswijzer van mijn stageportfolio

Datum: 01-01-2021

Gelezen en goedgekeurd door René Dohmen

Gegevens student:	Gegevens bedrijf:	Gegevens docent begeleider:
Luuk van Ettinger - 369175 ICT & Media Design 31-08-2020 t/m 22-01-2021 (90 werkdagen)	Formatics Dynamic Research & Development Stramproy René Dohmen - Directeur	Maikel Putman

De opdracht

Improviser Education

Improviser Education is een muziek educatie platform voor het leren improviseren van het spelen van muziek.

Opdrachtomschrijving

Welk UI framework is het beste-en meest geschikte UI framework om de Improviser Education website mee op te bouwen?

Na een afgewogen antwoord op deze vraag gebaseerd op de onderzoeksmethodes, moet de code van de oude UI framework componenten omgebouwd worden naar code die de nieuwe UI componenten gebruikt. Hierbij zullen nog verschillende nieuwe functies moeten worden ontworpen om zodanig tot een high fidelity prototype te komen.

Hierbij moet gekeken worden naar de nieuwe UI elementen, hier moet gekeken worden naar nieuwe mogelijkheden om het product zover te optimaliseren voor een zo efficiënt mogelijke user experience.

In het onderzoek moet vermeld worden welke verbeteringen het nieuwe UI framework met zich mee zal brengen tegenover het al bestaande UI framework. Daarnaast moet ook nagedacht worden over het onderhoudsprobleem; hoe zorg je ervoor dat het systeem niet jaarlijks vernieuwd moet worden?

Aanpak & uitvoering

In de eerste week van mijn stage heb ik een globale planning verwerkt in mijn [project document](#). Deze planning heb ik gedurende het stagetraject gebruikt als richtlijn om te kijken of ik op schema lag met het onderzoeken en beantwoorden van mijn onderzoeksvragen.

Week 1 - 3	Vorbereidingsfase
Week 3 - 12	Analysefase
Week 13 - 18	Ontwerpfase & ontwikkelingsfase
Week 19 - 20	Afrondings fase

*Uiteindelijke globale planning

Onderzoeksvragen

- Welk UI framework is het beste-en meest geschikte framework om de Improviser Education website mee op te bouwen?
- Is er een manier om te weten of een UI framework nog lang gebruikt zal worden?
- Kunnen tools van een nieuw UI framework ingezet worden om de user experience van de applicatie te verbeteren?

Het proces

De eerste 3 weken was mijn voornamelijke doel het compleet maken van het [project document](#) zodat het voor mij duidelijk was wat de opdracht precies is en hoe ik hier mee aan de slag ga de komende maanden. Daarnaast heb ik mijn eerste stappen gemaakt met het bekend worden en leren van React.js en de verschillende applicaties waaraan ik mee zou gaan werken vanuit Formatics.

Met verschillende feedback sessies van mijn stagebegeleider was het mij gelukt om in week 3 een compleet [project document](#) te hebben zodat we dit ook in het eerste bedrijfsbezoek konden bespreken. Wat voor mij het meest lastig was hierbij, was de globale planning. Het plannen van het ontwikkelen van software op zo'n lang termijn is erg lastig aangezien er altijd dingen op technisch gebied gebeuren waar je niet over na had gedacht. Vandaar dat ik in Formatics met de SCRUM methode heb gewerkt waarin we in sprints werken en de planning op basis van de resultaten van de sprint aanpassen. Hierdoor is mijn originele globale planning niet precies gelopen als in de praktijk.

Mijn eerste stap was het maken van een [storybook](#) voor Improviser Education. Op deze manier leerde ik kennis te maken met de applicatie en de bijbehorende componenten. Dit heeft mij ondersteuning gegeven bij het opstellen van een [Moscow analyse](#), aangezien ervaring en kennis hebben van de applicatie waarvoor je onderzoek uitvoert cruciaal is.

Er zijn een groot aantal UI framework om uit te kiezen en mijn voornamelijke doel in het begin van het onderzoek was het kaderen van deze opties naar een kleinere selectieve hoeveelheid. Dit heb ik gedaan door het opstellen van een [Moscow analyse](#) waarin ik de 10 meest gebruikte UI frameworks heb beoordeeld op een ontworpen criteria waaraan het uiteindelijke UI framework aan moet voldoen.

1. De moscow analyse

In de analyse komen eisen voor als het hebben van een minimaal aantal ontwikkelaars wegens veiligheid voor het UI framework, dat dit ook in de toekomst onderhouden zal blijven worden. Componenten moeten responsive zijn, het moet typescript ondersteuning hebben, en nog een aantal van dit soort technische eisen die mogelijke valkuilen moeten voorkomen.

2. Vergelijkingen en resultaten

Na het bekijken van de vergelijkingen is als resultaat te zien dat de UI frameworks: Elastic UI, Material UI en Semantic UI, alle 3 goede kandidaten zijn voor de applicatie. Deze UI frameworks voldoen alle 3 aan de opgestelde must haves van de [Moscow analyse](#) en deze kandidaten zal ik dus verder meenemen in het onderzoek.

Semantic UI zal minder de focus op komen te liggen aangezien dit het al bestaande UI framework is voor Improviser Education en hier dus al meer over bekend is.

Vervolgens ben ik met Material UI en Elastic UI gaan [prototypen](#) om mezelf meer te verdiepen in de voor-en nadelen van de UI frameworks en wat ze uniek maken. Material UI heeft code documentatie beschikbaar in typescript en de styling is eenvoudig maar behoudt de Material look. Elastic UI heeft een eenvoudig aanpasbare styling, maar ook hier behoudt het zonder te veel moeite de Elastic look. Verder heeft het erg fijne componenten voor data focused websites.

Naast de verschillende UI frameworks die ik tot nu toe heb onderzocht staat er ook nog een optie hier los van. [Styled Components](#) geeft de mogelijkheid om je persoonlijke UI framework in elkaar te zetten. Ik ben gaan onderzoeken of dit een beter alternatief is voor de Improviser in plaats van de 3 UI framework kandidaten. Ik heb [Styled Components](#) onderzocht door er mee te prototypen in een React applicatie van mij waarin nog geen gebruik werd gemaakt van een UI framework. Hierin heb ik gekeken naar de criteria van de [Moscow analyse](#) en de eisen getest. [Styled Components](#) voldoet voornamelijk dankzij zijn flexibiliteit aan alle opgestelde eisen. Nog een extra voordeel is het feit dat het ook in React Native gebruikt kan worden in tegenstelling tot de andere UI frameworks.

Mijn conclusie hierop is dat [Styled Components](#) de gebruiker meer vrijheid biedt en controle over de styling van hun applicatie, maar dit levert ook een hoop extra werk. Bij al bestaande UI frameworks is er al een hoop werk voor je gedaan, denk aan al gebouwde componenten, hier kun je dus vrij snel aan de slag. Bij styled components begin je from scratch en kan het maken van simpele componenten al een hoop werk kosten. Wat [Styled Components](#) uniek maakt tegenover de rest, is dat het gebruikt kan worden in [React Native](#), vandaar dat ik het aanraad te gebruiken wanneer de product owner een app wilt ontwikkelen voor Improviser Education, maar niet voor de web applicatie, aangezien het extra werk dat het met zich mee neemt te zwaar weegt voor het maken van deze keuze tegenover de andere kandidaten.

Voor de Improviser Education heb ik een app prototype gebouwd met [React Native](#) om te onderzoeken of [Styled Components](#) een geschikte optie is voor de styling in een [React Native](#) app. De UI framework kandidaten van het onderzoek zijn namelijk niet geschikt voor gebruik in een [React Native](#) app. Dit komt omdat deze UI frameworks gebouwd zijn voor React.JS (web apps) waarbij de output HTML/CSS is. De output van [React Native](#) is dit niet waardoor de UI frameworks ook niet hierin gebruikt kunnen worden.

Heeft de Improviser Education applicatie profijt van [React Query](#)? Deze vraag ben ik gaan onderzoeken nadat ik zag dat veel applicaties hier gebruik van maken en in deze voorbeelden bleek de data fetching library veel voordelen met zich mee te nemen. Ik heb dit onderzocht door middel van prototyping. Ik heb in een simpele blog applicaties het ophalen van data met en zonder [React Query](#) uitgetest om te kijken naar de verschillen. Omdat de blog applicatie zo klein en simpel is, ligt de focus erg op hetgeen dat ik onderzoek en zijn de verschillen ook erg duidelijk. Hieruit heb ik uiteindelijk kunnen concluderen dat het gebruik maken van [React Query](#) in de Improviser Education applicatie een positief effect heeft op zowel technisch gebied als op UX gebied door het opslaan van opgehaalde data



in de cache. Hierdoor hoeft de gebruiker een pagina maar 1 keer te laden en kan die door de applicatie blijven navigeren zonder telkens al geopende pagina's opnieuw te fetchen. Ik geef dan ook het advies om [React Query](#) op zoveel mogelijk plekken in de Improviser te implementeren.














Als voorbereiding op het onderzoeksdocument heb ik voor de verschillende UI kandidaten [proof of concepts](#) gebouwd voor de Improviser Education. Het doel hiervan was om verschillende criteria en mogelijkheden te onderzoeken om tot een uiteindelijk advies te komen voor de onderzoeksvraag.

In het [onderzoeksdocument](#) wordt er onderzocht welk UI framework het beste-en meest geschikte UI framework is om de Improviser Education website mee om te bouwen. Het document bevat alle relevante voorgaande onderzoeksdocumenten en prototypes om zodanig tot een definitieve conclusie te komen op de onderzoeksvraag.

Met Elastic UI als definitieve keuze op de onderzoeksvraag, ben ik aan de slag gegaan met het omzetten van de Improviser Education naar Elastic UI. Naast het omzetten van het UI framework heb ik mij ook bezig gehouden met het verbeteren en implementeren van een aantal nieuwe functies om de user experience te verbeteren. Deze nieuwe functies zijn onderaan de pagina van [high fidelity prototype](#) te zien met een beschrijving en wat ze precies verbeteren.

Met de definitieve keuze van Elastic UI heb ik een opzet ontworpen met behulp van de Elastic UI design guidelines. Hierbij heb ik het kleurgebruik van de Improviser behouden, maar heb ik meer gebruik gemaakt van wit, en de blauwe Improviser kleur op een subtielere wijze gebruikt voor het markeren van interactieve buttons en links om de gebruiker duidelijk te maken waar hij/zij interactie kan verwachten. Daarnaast ziet het data georiënteerde overzicht er overzichtelijker uit door het gebruik van meer witruimte. Ook is het logo van de Improviser verwerkt in de navigatiebalk waardoor deze op elke pagina is te zien. Op deze manier is de huisstijl van de Improviser overal in de applicatie terug te vinden.


[Scales](#)
[Riffs](#)
[Exercises](#)


Scales 								←	<	>	→
VIEW	BARS	CHORD	NAME	TAGS	CHORD INFO	CREATED	IMAGE				
	1	C	Triads whole step down 3 of 3			a year ago					
	1	C	Triads whole step down 2 of 3			a year ago					
	1	C	Triads whole step down 1 of 4			a year ago					
	1	Cmaj	Thirds up over Major scale with eights in 1 bar		c1:maj	2 years ago					
	2	Cmaj	Thirds up over Major scale with eights		c1:maj c1:maj	2 years ago					
	1	Cmaj	Thirds down over Major scale with eights in 1 bar		c1:maj	2 years ago					

Resultaten

Op dit moment staat er een definitieve keuze vast voor de onderzoeksvraag: welk UI framework het meest geschikt is voor de Improviser Education. Dit is Elastic UI en de Improviser applicatie is al voor een groot deel omgezet naar Elastic UI. Daarnaast staat er een nieuw ontwerp gebaseerd op de Elastic UI guidelines die de applicatie overzichtelijker en gebruiksvriendelijker maakt. Mijn doel voor de laatste weken van mijn stage is om de applicatie volledig omgezet te hebben. Elastic UI brengt verschillende tools mee die gebruikt kunnen worden om de user experience van de Improviser te verbeteren. Ik heb hier gebruik van gemaakt door het maken van een aantal nieuwe functies waarbij ik deze tools inzet. Door de hoeveelheid ontwikkelaars van het UI framework zal de kans minimaal zijn dat de applicatie binnen een korte tijd weer vervangen moet worden.

Reflectie

Ik heb een erg waardevolle en leerzame tijd gehad bij Formatics. Ik heb enorm veel kunnen leren op technisch gebied aangezien mijn bedrijfsbegeleider niet alleen mijn baas was maar ook een mentor, aangezien ik elk moment van dag altijd kon laten weten als ik ergens tegen aan liep en dan kreeg ik binnen no time ondersteuning. Dit heeft mij geholpen om van niks van programmeren af te weten naar een niveau te komen waar ik taken op mezelf kan nemen en iets kan betekenen voor het bedrijf. Aan het begin van mijn stage heb ik de volgende persoonlijke leerdoelen opgesteld:

- Kennis in React.js verbeteren
- Communicatief sterker worden
- Kennis over UX opdoen

Ik heb het afgelopen half jaar 10 keer zo veel geprogrammeerd dan al de jaren dat ik op Fontys Hogeschool studeer. Door de hoeveelheid uren die ik in het programmeren heb gestoken, ben ik op dit vlak ook erg gegroeid. Net als alle andere skills in de wereld, word je beter in programmeren door het veel te doen. Het React.JS ecosysteem is enorm en wat ik vooral heb geleerd dit stagetraject is dat er nog zoveel meer te leren is. Ik wil graag blijven leren en doorgroeien in React.JS en mogelijk andere programmeertalen en om dit te doen is het belangrijk dat ik aan de gang blijf. Ik zal naast mijn studie daarom ook voor Formatics blijven werken om in dit grote ecosysteem te blijven doorleren. Ik geloof dat deze kans erg waardevol is voor mijn ontwikkeling als ICT'er en hierom ben ik erg dankbaar voor Formatics dat ze mij deze kans geven.

In het project document heb ik verschillende afspraken gemaakt over de communicatie tussen mij en mijn bedrijfsbegeleider en stagebegeleider. Ik heb mij aan deze afspraken gehouden door elke ochtend met een standup te beginnen met mijn bedrijfsbegeleider en elke week een reflectie te sturen naar mijn stagebegeleider. Ik heb erg proberen te letten op dat ik communiceer naar mijn baas wanneer ik iets niet begrijp en ondersteuning nodig heb. Wat ik hier voornamelijk heb geleerd is dat het laten weten dat iets niet lukt zonder hier informatie bij toe te lichten niet productief is voor zowel mij als de persoon waar ik hulp aan vraag. Het is belangrijk om na te denken over hoe je de vraag formuleert en welke informatie je hierbij kan geven zodat de persoon waar je hulp aan vraagt ook daadwerkelijk kan helpen. Op deze manier kunnen we beide sneller weer aan de slag en leer ik er ook nog eens meer van.

Ik heb dit stage traject het minste tijd besteed aan mijn leerdoel om kennis over UX op te doen. Ik vond het interessant om te zien hoe UX en het maken van software met elkaar omgaan en wat voor conflicten hieruit komen. Een UI/UX designer kan een ontwerp doorgeven terwijl dit technisch gezien helemaal niet logisch/mogelijk kan zijn. Dit zag ik bijvoorbeeld voorbij komen in het ontwikkelen van de app voor de coffeeshop. Ik geloof dat het daarom belangrijk is om van beide kanten kennis te hebben. Om een UI te ontwerpen terwijl je geen verstand hebt

van hoe het gebouwd wordt en het bouwen van een applicatie zonder iets af te weten van UX gaat voor problemen zorgen. Ik heb in mijn high fidelity prototype nog een aantal wins proberen te maken op UX gebied door gebruik te maken van de tools van het gekozen UI framework.

Bewijs

	Beoordelings-dimensie	Bewijs
1	Kennis en Inzicht	Storybook Moscow Onderzoeksdocument
2	Toepassen Kennis en Inzicht	
2 a	Beheer	Storybook React Native Proof of concepts High fidelity prototype
2 b	Analyse	Storybook Moscow Prototyping Styled Components React Query Proof of concepts Onderzoeksdocument
2 c	Advies	Moscow Styled Components React Query Onderzoeksdocument

2 d	Ontwerp	React Native High fidelity prototype
2 e	Realisatie	React Native High fidelity prototype
3	Oordeelsvorming	Moscow Prototyping Styled Components React Query Proof of concept Onderzoeksdocument
4	Communicatie	Project document Onderzoeksdocument Leeswijzer
5	Leervermogen	Onderzoeksdocument Portfolio Leeswijzer