

2017

Bijlage

SCRIPTIE MATCH EXPERIENCE
PATRICK

Table of Contents

No table of contents entries found.

1. Adviesrapport JavaScript framework

Abstract

In dit onderzoek wordt er gekeken naar welk javascript framework en/of library het beste past bij het bedrijf Gamebasics en in het specifiek voor de webafdeling 'Team Best'. De javascript frameworks en/of libraries worden als eerst uitgelegd. Daarna zal er een onderlinge vergelijking plaatsvinden. Waar uiteindelijk een advies uit zal voortkomen over welk framework er het beste gebruikt kan worden.

Introductie

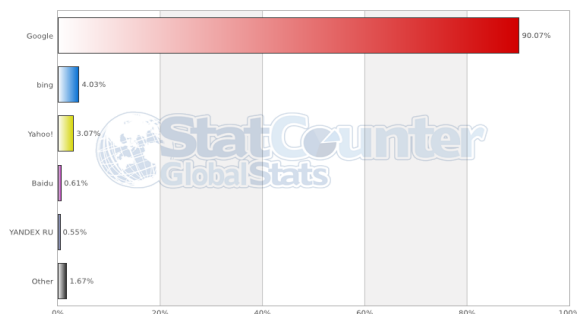
Dit onderzoek wordt gedaan voor het bedrijf Gamebasics en is onderdeel van een afstudeeropdracht. De bedoeling is dat er wordt onderzocht welk javascript framework het beste past bij het bedrijf. Er zal gedurende een werkweek worden onderzocht welk framework het beste past bij het bedrijf. Om te bepalen aan welke eisen het javascript framework moet voldoen zullen er requirements moeten worden opgesteld. Om inzicht te geven in de frameworks zal er over elk framework een uitleg worden gegeven. Daarna zal er een vergelijking plaatsvinden tussen deze frameworks om uiteindelijk een advies te geven aan de hand van de resultaten.

Requirements

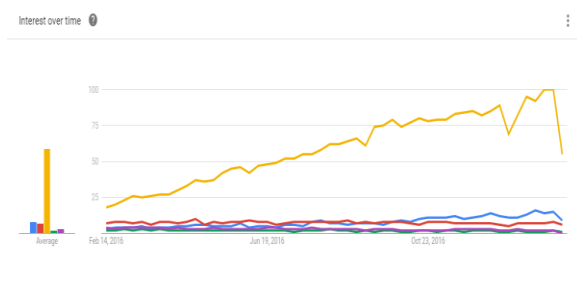
Zoals in elk project is requirements van groot belang. Om de requirements te bepalen zijn er interviews gedaan met de product owner en lead-developer. In de tabel hieronder is te zien welke requirements hieruit zijn voortgekomen.

Requirements:	
1	Het javascript framework moet een single page kunnen weergeven en ondersteunen.
2	Het javascript framework moet performance gericht zijn.
3	Het javascript framework moet typescript kunnen ondersteunen of een andere static typed language
4	Het Javascript framework moet externe libraries kunnen ondersteunen.
5	Het Javascript framework moet goed testbaar van native test libraries of door gebruik van externe libraries.
6	Het Javascript framework moet een actieve community hebben, zodat problemen sneller opgelost kunnen worden
7	Het Javascript framework moet het flux design pattern kunnen ondersteunen door native libraries of externe libraries.
8	Het Javascript framework moet alle browsers ondersteunen vanaf versie IE – 11, Chrome – 56.0.2, Firefox – 51.0 , Safari – 10.0.2 Android – 4.1, iOS – 8.0

Nadat de requirements zijn opgesteld ben ik aan de slag gegaan om te zoeken welke javascript frameworks hieraan voldoen. Aangezien er een groot aantal javascript frameworks bestaan was het van belang om eerst het aantal te verkleinen naar een aantal dat haalbaar was om te onderzoeken in de werkperiode dat hiervoor was uitgerekend. Om het aantal te verkleinen heb ik requirements zes gepakt. Dit zorgt er namelijk voor dat de kleine frameworks die weinig tot geen ondersteuning krijgen van de community gelijk afvallen. Uiteindelijk heb ik na overleg ervoor gekozen om de vijf meest populairste frameworks te pakken. Om te bepalen welke frameworks nu het populairst zijn heb ik twee criteriums opgesteld. Namelijk top zoekopdrachten en in dit geval met de google engine. De reden dat ervoor gekozen is om alleen google te gebruiken is te danken aan dat google een 90% van de markt bezit[11-12].



Als tweede criteria was het aantal backers/supporters op GitHub. Dit heb ik aan de hand van de star rating gedaan die op GitHub bestaat. Als eerste heb ik tien verschillende websites gebruikt om te bepalen om welke javascript frameworks ik zou gaan zoeken [1-10]. Na uit dit onderzoek de vijf meest genoemde frameworks te hebben gepakt heb ik gekeken naar het aantal google searches. In de chart is te zien hoe populair het desbetreffende framework is qua google searches.



Framework	Color
Vue	Blue
React	Red
Angular2	Yellow
Ember	Green
Backbone	Purple

De chart heeft data van de laatste twaalf maanden, hieruit is te halen dat Angular qua searches ver boven de rest uitstijgt met daarbij Vue en React op een tweede en derde plek komen. Van deze top is er verder gekeken naar hoeveel backers/supporters deze frameworks hebben. Als er te weinig supporters waren, in dit geval minder dan 10000, zou het framework niet worden meegenomen. Uiteindelijk zijn alle drie de frameworks ver boven de 10000 geëindigd wat te zien is in de tabel hieronder.

De GitHub star ratings zijn opgehaald op 07-02-2017.

Framework	Stars
React.js	59.154
Angular2	19.832
Vue.js	39.933

Uit de tabel is te halen dat React qua backers/supporters ver boven de andere uitstijgt qua GitHub activiteiten. Vue op een goede tweede plek komt en Angular2 op de laatste plek komt. De reden voor Angular2 zal komen doordat er nog veel gebruik gemaakt wordt van AngularJS de eerdere versie. Omdat alle drie de frameworks voldoen aan requirement zes, namelijk een actieve community zijn ze alle drie meegenomen in de vergelijkingen.

React.js

Het eerste framework dat nader zal worden besproken is React [13]. React is in principe geen framework, maar een library. React zorgt voor het V-gedeelte in een MVC pattern. De React library is ontwikkeld door Facebook en is een open-source library. Er zijn redelijk wat bekende bedrijven die gebruik maken van React, waaronder Facebook, Instagram, Netflix en Walmart [14]. React staat bekend om het als eerste te hebben ontwikkeld van het Flux design pattern [15]. Naast dat React gebruik maakt van het flux design pattern, heeft het als andere features, one way data binding, JSX [16] als javascript syntax, een virtual DOM. JSX is een component based language die gebruik maakt van javascript. Het voordeel van JSX is dat het de javascript code een stuk overzichtelijker maakt. Ook maakt React zoals eerder vermeld gebruik van een virtual DOM. Dit houdt in dat React een in memory data structure cache aanmaakt. Deze virtuele cache bekijkt wat de verschillen zijn tussen het nieuwe model en het model die in de actuele DOM bestaat en zorgt er dan voor dat alleen de componenten die veranderd zijn wordt aangepast. Wat zorgt voor winst op performance gebied. Om te kijken of React de potentie heeft om het nieuwe front-end library te worden van Gamebasics is het van belang dat het voldoet aan alle requirements. React is vooral gemaakt voor single page applicaties, hoewel het ook grote projecten kan ondersteunen. Daarnaast kan React gebruik maken van een flux of redux design pattern [17]. Dit kan gedaan worden door de externe libraries toe te voegen aan het project. Ook is het mogelijk om static typing te ondersteunen dit kan gedaan worden door nogmaals een externe library toe te voegen aan het project. Na alle requirements te hebben nagelopen voldoet React aan alle eisen die zijn gesteld en is om die reden een goede kandidaat om de uiteindelijke library/framework te worden.

Angular2

Als tweede framework is Angular2[18] aan de beurt. Angular2 had bij de google searches een grote meerderheid, dit viel alleen tegen als er wordt gekeken naar de GitHub stars rating. Dit kan zoals eerder vermeld komen doordat er op het moment nog steeds veel gebruik gemaakt wordt van de vorige versie van Angular2, namelijk AngularJS. Angular2 is ontwikkeld door Google. Angular2 maakt gebruik van een MVVM-design pattern [19] en is in dat opzichte anders dan React. Angular2 is een geheel framework dat gebruik maakt van elk aspect van een MVVM-design pattern. Om die reden heeft het dus ook services en controllers. Naast het MVVM pattern maakt Angular2 ook gebruik van een two-way data binding [20]. Ook in dat opzicht is het anders dan React dat een one-way data binding gebruikt. Two-way data binding zorgt ervoor dat de view en model elkaar direct kunnen updaten, indien er een verandering plaatsvindt. Angular2 heeft veel van zijn ontwikkeltijd besteed aan performance. Een van de ontwikkelingen daaraan is het gebruik van Code splitting [21]. Code splitting zorgt ervoor dat alleen de code wordt geladen dat daadwerkelijk wordt gebruikt door de applicatie en in het specifiek de huidige view. Net zoals React kan in Angular2 een single page applicatie worden gemaakt, maar ook grote projecten zijn hierin mogelijk. Ook Angular2 kan gebruik maken van een flux of redux design pattern hoewel dit tegen de native two-way binding ingaat. Om een flux of redux design toe te passen moet in Angular2 een externe library worden toegevoegd aan het project. Daarnaast maakt Angular2 als native al gebruik van static typing en er hoeft daarom geen externe library worden toegevoegd. Na alle requirements nagelopen te hebben, voldoet ook Angular2 aan alle requirements en is om die reden ook een framework dat kan worden gebruikt voor het front-end.

Vue.js

Als laatste komt Vue [22] aan de beurt. Vue is waarschijnlijk de minst bekende van de frameworks die hier worden beschreven. Dit neemt niet weg dat er genoeg grote bedrijven zijn die al gebruik maken van Vue, waaronder Alibaba, Baidu en Gitlab. Net zoals Angular2 maakt Vue gebruik van een MVVM-design pattern. Daarnaast heeft Vue als native ook two-way data binding. Vue maakt net zoals React gebruik van een virtual Dom. Naast bovenstaande design keuzes heeft Vue de focus gelegd op performance en maintainability. Dit wordt gewaarborgd door de eerdergenoemde virtual dom, maar ook door reactive en composable view components en een kleine maar robuuste core [23]. Vue heeft in dat geval genoeg geleerd van React en Angular2 want ook Vue heeft een externe library die het redux design pattern ondersteund. Namelijk de Vuex library. Vue kan hierom ook grote applicaties ondersteunen, maar werkt ook voor single page applicaties. Als laatste is er ook een externe library die kan gebruikt worden op static typing te ondersteunen. Net zoals React en Angular2 voldoet Vue.js aan alle requirements die zijn opgesteld en is daarom ook een kandidaat om het nieuwe front-end framework te worden.

Resultaat

Om een vergelijking te maken tussen de drie frameworks zal ik gebruik maken van de requirements. De requirements hebben een cijfer gekregen met hoe belangrijk een bepaalde requirement is voor het framework. De requirements kunnen een cijfer een tot en met drie krijgen, waarbij het cijfer een heel belangrijk is en drie het minst belangrijk. Om te voorkomen dat er overal het cijfer een wordt gegeven mag er maximaal maar drie requirements een cijfer een krijgen. Dit zorgt ervoor dat alle requirements worden opgedeeld. In de tabel hieronder is het resultaat te zien van deze prioritering.

Requirement	Cijfer
1:	1
2:	2
3:	1
4:	2
5:	2
6:	1
7:	2
8:	3

Nu het resultaat van de requirements bekend is kan er naar de frameworks worden gekeken ten opzichte van deze requirements

	React	Angular2	Vue
Requirement 1: (1)	Single page/Large app	Large app/Single page	Large app/Single page
Requirement 2: (2)	Goed	Goed	Heel Goed
Requirement 3: (1)	Externe Library	Native	Externe Library
Requirement 4: (2)	Goed	Goed	Goed
Requirement 5: (2)	Externe Library	Externe Library	Externe Library
Requirement 6: (1)	Heel Goed	Heel Goed	Goed
Requirement 7: (2)	Native	Externe Library	Externe Library
Requirement 8: (3)	Heel Goed	Goed	Goed

In bovenstaand tabel betekent groen dat het framework, ten opzichte van de andere, het beste resultaat heeft geboekt. React heeft het beste resultaat geboekt op requirement een. De reden hiervoor is, is dat React gemaakt is voor kleine single page applications, dit terwijl Angular2 en Vue zijn gemaakt voor large apps. Dit betekend niet dat Angular2 en Vue geen single page apps ondersteunen. Op requirement twee is Vue als beste naar voren gekomen. Performance blijft ten alle tijden moeilijk om te bepalen wie hier het beste in scoort. Uit extern onderzoek is gebleken dat Vue het beste scoort op dit gebied [24]. Requirement drie is strongly/static typed support. Er is hier het voordeel aan Angular2 gegeven, omdat deze het native al gebruikt en om die reden geen externe libraries nodig heeft. Requirement vier voldoen alle drie de frameworks aan en in alle frameworks is dit weinig tot geen moeite. Requirement vijf was het testen van de applicatie, dit moeten alle drie de frameworks overhouden aan een externe library. Om die reden heeft iedereen goed gescoord op requirement vijf. Requirement zes was de requirement waarmee ik op deze drie frameworks ben gekomen, namelijk active community en hoewel ze alle drie een active community hebben, was het een stuk makkelijker om voor React en /of Angular2 informatie te vinden. React scoort op requirement zeven een beter cijfer dan de overige twee frameworks, dit omdat er native al gewerkt kan worden met het flux pattern in React. Voor de andere twee frameworks is het nodig om een externe library te gebruiken. Als laatste was het nodig om zoveel mogelijk browsers te ondersteunen. Ook hier scoort React iets beter als de rest, doordat deze onder andere vanaf ie 8 ondersteund. Dit terwijl de overige twee van ie 9 pas ondersteunen. Voor de laatste requirement is niet gekeken naar externe libraries en om die reden is het dus mogelijk dat ook React geen ie 8 meer zou kunnen ondersteunen.

Conclusie/Advies

Aan de hand van bovenstaand resultaat zou ik adviseren om gebruik te maken van de React library. Daarnaast zou ik aanraden om, indien er gebruik gemaakt wordt van de React library, gebruik te maken van de redux library. De Redux library zorgt er namelijk voor dat het onderhoud van de applicatie veel beter en gestructureerd is. Ook is er een uitstekende library die door onder andere door Netflix wordt gebruikt voor async calls, namelijk Rx.js. Op deze manier is het mogelijk om een hele mooie gestructureerde applicatie te maken. Indien React uiteindelijk niet wordt gekozen gaat mijn tweede keus naar Angular2. Dit framework kwam als tweede uit de resultaten, maar scoort op alle punten, waar het net niet even goed scoorde als React, beter dan Vue. Naast de bovenstaande frameworks zijn er nog redelijk wat andere frameworks die eventueel de moeite nog waard zijn om te bekijken. Namelijk Ember.js, Backbone.js, Aurelia en Meteor.

Referenties

1. Dunebook – Top javascript frameworks. (n.d.). Retrieved February 07, 2017, from <https://www.dunebook.com/5-best-javascript-frameworks-to-learn-in-2017/>
2. Dzone – Top javascript frameworks. (n.d.). Retrieved February 07, 2017, from <https://dzone.com/articles/top-javascript-frameworks-list-in-2016>
3. Elliott, E. (n.d.). Medium.com – Top javascript frameworks, Retrieved February 07, 2017, from <https://medium.com/javascript-scene/top-javascript-frameworks-topics-to-learn-in-2017-700a397b7111#evgjyzvdg>
4. HaeckDesign - Top javascript frameworks. (n.d.) Retrieved February 07, 2017, from <https://haeckdesign.com/blog/favorites/technology/top-javascript-frameworks>
5. INfoQ.com – Top javascript frameworks. (n.d.). Retrieved February 07, 2017, from <https://www.infoq.com/research/top-javascript-mvc-frameworks>
6. Korotya, E. (n.d.). Hackernoon.com – Top javascript frameworks. Retrieved February 07, 2017, from <https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282#.mdoh30heg>
7. Noeticforce - Top javascript frameworks. (n.d.). Retrieved February 07, 2017, from <http://noeticforce.com/best-Javascript-frameworks-for-single-page-modern-web-applications>
8. Wikipedia - Javascript Frameworks. (n.d.). Retrieved February 07, 2017, from https://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks
9. Tecadmin – Top javascript frameworks. (n.d.). Retrieved February 07, 2017, from <http://tecadmin.net/top-8-best-javascript-frameworks-for-2017/>
10. ValueCoders – Top javascript frameworks. (n.d.). Retrieved February 07, 2017, from <http://www.valuecoders.com/blog/technology-and-apps/top-javascript-frameworks-list-comparison/>
11. Pure-Im – Google marktaandeel (n.d.). Retrieved February 07, 2017, from <http://www.pure-im.nl/blog/marktaandelen-zoekmachines-q4-2016/>
12. Dewebanalist – Google marktaandeel (n.d.). Retrieved February 07, 2017, from <http://www.dewebanalist.nl/marktaandeel-google-zoekmachine-september-2016>
13. React – React javascript library (n.d.). Retrieved February 09, 2017, from <https://facebook.github.io/react/>
14. Wikipedia – React javascript library (n.d.). Retrieved February 09, 2017, from [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

15. React – Flux design pattern (n.d.). Retrieved February 09, 2017, from <https://facebook.github.io/flux/docs/in-depth-overview.html#content>
16. React – JSX javascript syntax (n.d.). Retrieved February 09, 2017, from <https://facebook.github.io/jsx/>
17. Redux – Redux design pattern (n.d.). Retrieved February 10, 2017, from <http://redux.js.org/>
18. Angular2 – Angular2 javascript framework (n.d.). Retrieved February 11, 2017, from <https://angular.io/>
19. Wikipedia – MVVM design pattern (n.d.). Retrieved February 11, 2017, from <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>
20. Angular2 – Angular2 two way data binding (n.d.). Retrieved February 11, 2017, from https://angular-2-training-book.rangle.io/handout/components/app_structure/two_way_data_binding.html
21. Webpacks – Code splitting (n.d.). Retrieved February 11, 2017, from <https://webpack.github.io/docs/code-splitting.html>
22. Vue – Vue javascript framework (n.d.). Retrieved February 12, 2017, from <https://vuejs.org/>
23. Vue – Vue javascript framework (n.d.). Retrieved February 12, 2017, from <https://vuejs.org/v2/guide/comparison.html>
24. Comparison – Performance Comparison (n.d.). Retrieved February 14, 2017, from <http://stefankrause.net/js-frameworks-benchmark4/webdriver-ts/table.html>

2. Plan van aanpak

Inleiding

Gamebasics is de maker van het voetbal spel Online Soccer Manager (OSM). Voor de ontwikkeling van dit voetbalspel worden nieuwe features bedacht en gemaakt.

Match Experience is een van de nieuwste features van OSM. In de feature wordt aan de aan van commentatoren en 2d animaties de wedstrijd die het laatst is gespeeld voorgeschoteld aan de gebruiker. Het doel hiervan is de gebruiker het gevoel geven dat de gekozen tactieken en instellingen daadwerkelijk invloed hebben gehad op het resultaat van de gespeelde match.

Tijdens mijn afstudeerperiode zal ik voor de afdeling web gaan werken en deze feature gaan implementeren. Naast dat de feature zal moeten worden geïmplementeerd, zal er moeten worden gekeken naar onderzoek van javascript frameworks, het ontwerpen van een design dat communiceerd met de huidige applicaties en zal er unit en waar nodig a/b testen gemaakt moeten worden.

Het Bedrijf

- Geschiedenis

Gamebasics is het bedrijf achter het online voetbalspel Online Soccer Manager. Gamebasics is in 2004 opgericht, maar is begonnen in 2001 op de zolderkamer van Jeroen Derwort. Het bedrijf is inmiddels uitgegroeid tot een volwassen bedrijf van ongeveer 40 werknemers. Deze 40 werknemers werken elke dag hard aan het spel OSM dat op het moment meer dan 4 miljoen unieke spelers heeft. Elke avond worden er door de engines alle wedstrijden met de tactieken die de gebruikers hebben ingevoerd afgespeeld. Inmiddels is versie 3.0 van de game OSM uit en is het bedrijf continue bezig om nieuwe features te bedenken en huidige features te verbeteren. Gamebasics werkt in een agile omgeving met multi disciplinaire teams die ieder aan een eigen platform werken.

- Missie

De missie die voordat ik kwam afstuderen, was de onderstaande missie.

"Gamebasics streeft ernaar alle voetballiefhebbers wereldwijd het gevoel te geven dat ze manager zijn van hun favoriete voetbalclub. Elke dag en overal. Dit willen we bereiken door een game ervaring te bieden die authentiek, laagdrempelig en uitdagend is".

Deze missie is veranderd naar het volgende.

"Wij geven alle voetballiefhebbers de mogelijkheid met elkaar de competitie aan te gaan door middel van een laagdrempelig en authentiek voetbalmanagerspel."

Opdrachtgever

Naam: Raymond Manche

Rol: Product Owner / Opdrachtgever

Afdeling: iOS / Web

Naam: Jurgen Welschen

Rol: Software Developer / Begeleider

Afdeling: Web

Probleemanalyse en opdrachtomschrijving

- Probleemstelling

Wanneer een gebruiker aan logt op Online Soccer Manager, vanaf nu OSM genoemd, wordt hij/zij doorgestuurd naar zijn/haar clubslots. Nadat de gebruiker zijn club heeft geselecteerd komt deze terecht op het hoofdmenu, waar direct de uitslag van de match is te zien. Hoewel dit praktisch is, is het niet leuk voor de gebruiker om gelijk zijn uitslag te zien. Dit terwijl er veel gebeurd kan zijn in de wedstrijd. De Match experience moet hierin verandering gaan brengen, door een samenvatting te laten zien van de wedstrijd. De samenvatting zal door gebruik te maken van commentatoren en 2d Animaties de gebruiker het gevoel moeten geven dat ze ook daadwerkelijk de wedstrijd aan het spelen zijn. Dit moet ervoor zorgen dat de gebruiker zich meer verbonden voelt met zijn club en spelers, en uiteindelijk ook met het spel OSM. Doordat de Match Experience al is gemaakt en actief is op de huidige iOS versie van het spel. Zijn de verwachtingen dat door de match experience meer spelers het spel zullen blijven spelen voor een langere tijd.

Om gelijk gebruik te maken van de situatie zal door het maken van de match experience gelijk een tweede probleemstelling worden aangepakt. Namelijk het onderzoeken naar mogelijke nieuwe javascript frameworks. Op het moment is het bedrijf Gamebasics en in het specifiek de afdeling Web bezig met het herinstrueren van de applicatie, hierbij hoort ook in welk framework het front-end zal worden geschreven.

- Opdrachtomschrijving

De opdracht kan in twee delen worden omschreven, in het eerste deel zal onderzoek gedaan worden naar verschillende javascript frameworks en hierbij zal een adviesrapport gegeven worden over welk framework het beste past bij het maken van de feature Match experience en welke eventueel gebruikt zou kunnen worden voor de gehele applicatie. Het tweede gedeelte en ook het belangrijkste is het maken van de Match Experience. De gehele opdracht zal worden gebruikt door het bedrijf en zal worden gezien als vooronderzoek. Ik zal eerst goed moeten kijken naar welk javascript het beste past bij de requirements die de afdeling Web voorlegt. De requirements zullen aan de hand van interviews moeten worden achterhaald en worden gebruikt om het beste javascript framework te kiezen. Ook verwacht het bedrijf een advies rapport over de onderzochte javascript frameworks en waarom er voor een bepaald framework is gekozen om hierin het prototype te maken van de match experience.

De Match experience zelf zal moeten worden gemaakt door gebruik te maken van de al bestaande Api calls die naar de back-end gaan. De afstudeerder zal dus alleen de gehele front-end moeten maken en zal hierbij goed moeten kijken naar hoe hij de applicatie structureert. Naast het maken van de Match experience is het noodzakelijk dat goed wordt nagedacht over welke functies er worden getest en op welke manier. Indien er namelijk gekozen wordt voor een bepaald test framework is het ook hier noodzakelijk om een advies te geven over welk test framework er wordt gebruikt. De match experience, indien volledig zal worden gebruikt in de huidige situatie en zal eventueel het begin zijn van een nieuwe herstructurering van de applicatie.

Doelstelling en eindresultaat

- Doelstelling

Gebruiker:

Het doel voor de gebruiker is om een betere speelervaring te krijgen.

Opdrachtgever:

Het doel van de opdrachtgever is om de match experience te kunnen gebruiken in de huidige situatie.

Begeleider/Web:

Het doel van de afdeling web is om de match experience te kunnen gebruiken in de huidige situatie en het prototype te gebruiken om verder onderzoek op te doen.

- Eindresultaat

De volgende onderdelen zijn een vereiste:

- Een adviesrapport over de desbetreffende javascript frameworks.
- Een werkend prototype van de match experience.

Optional:

- Een adviesrapport met daarin:
 - Gekozen modules
 - Een package manager
 - Test framework
 - Module handlers
 - Websockets

Randvoorwaarden en risicoanalyse

- Randvoorwaarden

Geld: N.V.T.

Huisvesting: Er zal gebruik worden gemaakt van een desktop-pc en bureau. Het bedrijf zal deze moeten leveren aan de student.

Materialen: de keuze in tools, methodes, technieken, frameworks, IDE, etc. is geheel vrije keus aan de student, zolang deze de keuzes goed kan onderbouwen.

Expertise: Aangezien er gebruik zal worden gemaakt van de script taal javascript en typescript is het van belang dat het bedrijf iemand in dienst heeft die hier ervaring in heeft. Ook zou het een toegevoegde waarde zijn als het bedrijf iemand in dienst heeft die ervaring heeft met package manager, module handles, websockets en alle andere randzaken die van belang zijn bij een web-applicatie.

Tijd: Er zal ongeveer 2 werkweken worden gebruikt om het javascript framework te onderzoeken en daarbij een advies rapport te schrijven. Hierna zal alle tijd nodig zijn om een goed gestructureerde front-end te schrijven voor het prototype van de match experience. (design/implementatie/test).

Beschikbaarheid: Aangezien het bedrijf de klant is, en er altijd een software developer aan het werk is op de afdeling web tijdens kantooruren. Zal dit geen probleem zijn.

- Risicoanalyse

Er zijn meerdere Risico's gebonden, hieronder zijn de risico's beschreven en welke maatregelen zijn genomen om deze risico's zo klein mogelijk te maken.

Niet alle requirements aanwezig: Omdat het bedrijf op het moment ook bezig is met een onderzoek om de front-end van de web te herstructureren. Is het mogelijk dat niet alle requirements aanwezig zijn tijdens het maken van het prototype.

- Oplossing:

Als eerste heb ik zowel de begeleider als de opdrachtgever geïnterviewd, zodat van beide de requirements te horen waren die er op het moment bekend waren. Omdat het hier gaat om een prototype en een vooronderzoek is dit een goed moment om de overige requirements boven tafel te krijgen.

Scope te groot: Er is volledige vrijheid in de gehele keuze aan tools, technieken, methodieken, etc. Dit kan zorgen dat het onderzoek gedeelte, samen met het maken van de adviesrapporten zo lang gaan duren dat er te weinig tijd over zal zijn voor een werkend prototype.

- Oplossing:
Om dit te voorkomen heb ik een strict tijdschema gemaakt, waarbij ik het onderzoek gedeelte niet langer wil laten duren dan dat ik daar heb aangegeven. Indien er aan het einde van de afstudeerperiode nog tijd over is, zal er nog terug gekeken kunnen worden en eventueel een nieuw onderzoek kunnen worden gestart.

Te weinig ervaring over het framework dat er wordt gekozen uit het vooronderzoek: Omdat het framework dat eventueel in het adviesrapport komt een nieuw framework kan zijn, waar nog geen werknemer ervaring mee heeft. Is het mogelijk dat ik niet goed genoeg begeleid kan worden.

- Oplossing:
Keuze aanpassen aan de hand van de ervaringen die de webdevelopers hebben
- Oplossing:
Meer tijd naast kantooruren besteden om het desbetreffende framework te leren.
- Oplossing:
Cursussen volgen in eigen tijd, online, school, etc.

Aanpak

1. Plan van aanpak schrijven
2. Requirements opstellen – Framework
3. Frameworks onderzoeken
4. Advies Rapport schrijven Frameworks
5. Requirements opstellen – Match Experience
6. Scrum Indeling maken – Match Experience
7. Design maken – huidige situatie naar Match Experience
8. Requirements + Design implementeren
9. Testen Functionaliteiten

8 en 9 kunnen indien nodig worden omgedraaid. Dit is afhankelijk van wat er wordt gemaakt. Er zal Test Driven Development worden toegepast indien het om business logica gaat en er zal eerst implementatie worden gemaakt en dan getest worden als het om UI-logica gaat. Er is hiervoor gekozen, omdat het ui het best te debuggen is als de implementatie bestaat. Dit voorkomt het maken van stubs, etc. voor het maken van de testen.

Tijdens de afstudeerperiode zal er 6 sprints worden ingedeeld. Op deze manier is er genoeg ruimte gelaten voor eventuele vertragingen.

Hieronder is een globale indeling te zien.

Sprint 1	Onderdeel
	Requirements opstellen match experience
	Klassendiagrammen maken structuur applicatie
	Project opzet
	Testen schrijven
Sprint 2	Onderdeel
	Data ophalen
	Studio (view) opzet (responsive)
	Data weergeven
	Testen schrijven
	Klassendiagrammen maken commentatorenteksten
Sprint 3	Onderdeel
	Studio (view) opzet (responsive)
	Data weergeven
	Testen schrijven

3. Requirements Match Experience

Nr.	Requirement	MOSCOW	Soort
1	Het systeem moet animaties kunnen weergeven.	S	F
2	Het systeem moet animaties fullscreen kunnen weergeven.	S	F
3	Het systeem moet meerdere soorten animaties kunnen weergeven. • Goal/No Goal • Presentatoren • Geel/Rood • Wissel • Geblesseerd	C	F
4	Het systeem moet de animaties voor een schot kunnen simuleren (acceleratie).	S	F
5	Het systeem moet teksten kunnen ophalen.	M	F
6	Het systeem moet opgehaalde teksten kunnen weergeven.	M	F
7	Het systeem moet images kunnen ophalen.	M	F
8	Het systeem moet opgehaalde images kunnen weergeven.	M	F
9	Het systeem moet Json format data kunnen ophalen van de bestaande web-API.	M	NF
10	Het systeem moet time based events kunnen oplossen.	M	NF
11	Het systeem moet een gehele voetbalwedstrijd kunnen weergeven. • Loading -> Studio(voorbeschouwing) -> Timeline (eerste helft) -> Studio (rust) -> reclame -> Timeline (tweede helft) -> Studio (nabeschouwing) -> Timeline (verlenging) -> Studio (nabeschouwing verlenging) -> Penalty's -> Studio (nabeschouwing wedstrijd)	M	F
12	Het systeem moet user input kunnen ontvangen.	W	F
13	Het systeem moet user input kunnen verwerken.	W	F
14	Het systeem moet verschillende knoppen kunnen weergeven. • Skip • Vooruit spoelen • Pauze	C	NF
15	Het systeem moet penalty's kunnen weergeven.	M	F
16	Het systeem moet penalty schieten ondersteunen.	M	F
17	Het systeem moet verlengingen kunnen weergeven.	M	F
18	Het systeem moet verlengingen ondersteunen.	M	F
19	Het systeem moet verschillende views ondersteunen.	S	F
20	Het systeem moet verschillende views kunnen weergeven. • Timeline • Studio • Penalty • Loading	M	F

21	Het systeem moet een timeline kunnen weergeven waarop het wedstrijdverloop is te zien.	M	
22	Het systeem moet een timer kunnen weergeven die van 0 tot 120 kan weergeven.	M	F
23	Het systeem moet events kunnen bijhouden.	M	F
24	Het systeem moet events kunnen weergeven.	M	F
25	Het systeem moet responsive zijn. 0 tot 768 pixels (mobile). 768 pixels tot max (tablet, laptop, desktop).	S	NF
26	Het systeem moet in een single page framework gebouwd worden	M	NF
27	Het systeem moet de thuisspelende ploeg te allen tijde links op het scherm weergeven.	M	F
28	Het systeem moet de uit spelende ploeg te allen tijde rechts op het scherm weergeven.	M	F
29	Het systeem moet de animaties van de ploeg die van de gebruiker is altijd een blauwe tint geven.	M	NF
30	Het systeem moet de animaties van de ploeg waar de gebruiker tegen speelt altijd een rode tint geven.	M	NF
31	Het systeem moet penalty reeksen kunnen weergeven.	M	F
32	Het systeem moet de penalty reeks inkleuren met groen, indien er is gescoord.	S	NF
33	Het systeem moet de penalty reeks inkleuren met rood, indien er niet is gescoord.	S	NF
34	Het systeem moet de matchdata te allen tijde van boven naar beneden op de timeline weergeven.	M	F
35	Het systeem moet wanneer er van de studio naar de timeline view wordt gewisseld inzoomen op de tv die is weergegeven in de studio view.	M	F
36	Het systeem moet te allen tijde de huidige score weergeven.	M	F
37	Het systeem moet te allen tijde de club weergeven met de manager naam en club logo.	M	F
38	Het systeem moet wanneer er een chance of goal event is, de animatie van de presentatoren altijd links weergeven wanneer het een thuis event is.	M	F
39	Het systeem moet wanneer er een chance of goal event is, de animatie van de presentatoren altijd rechts weergeven wanneer het een uit event is.	M	F
40	Het systeem moet tijdens een animatie een donkere overlay geven.	S	NF
41	Het systeem moet data kunnen bijhouden.	M	F