# Scriptie

# Business Case

Voor het ontwikkelen van concentrator V3 is er een nieuwe front-end nodig. De vorige was met “ext.js” ontwikkeld en was destijds

# Orientatie hoofdvraag

De wensen van diract-IT zijn het opzetten van een API

# Onderzoeksmethoden

Randvoorwaarden en wensen voor worden gedaan met kwalitatieve interviews aan ontwikkelaars en gebruikers.

Om in te spelen op de wensen van de gebruikers en ontwikkelaars wordt er gebruik gemaakt van deskresearch en kleinere proof-of-concepts.

# Glossary

# Hoofdvraag

Hoe ontwerp ik een responsive en herbruikbare front-end voor de business case van Diract-IT?

# Wat is de business case voor C.Next?

Om randvoorwaarden vast te stellen zijn er meerdere interviews gedaan. Uit het eerste interview met de Teamlead (Pieter de Wit) bijlage kwam naar voren dat er organisatie-breed een wens is over te stappen naar een “service oriented architectuur” (meer in het volgende hoofdstuk).

De tweede wens vanuit klanten is het beschikbaar maken van de applicatie op mobile devices. De huidige front-end is enkel in desktopformaat beschikbaar op tablets en smartphones. “Responsive design” is ook niet makkelijk toe te passen op de huidige opzet.

Onuitgesproken wens is dat er gebruik wordt gemaakt

# Wat zijn de grootste valkuilen in javascript applicaties?

# Wat verschilt er tussen MVC en SOA?

Om een goed beeld te krijgen van

# Welk SPA framework past het best?

Er zijn momenteel veel gratis javascript frameworks beschikbaar om applicaties in te maken. Deze worden ondersteund vanuit grote organisaties als google en facebook.

Om een goede keuze te maken zijn er criteria opgesteld. Deze komen deels uit ervaring en deels uit de business case.

Het framework moet actief in ontwikkeling zijn zodat het met de tijd mee kan gaan.

Het framework moet volwassen zijn, of in ieder geval geen grote veranderingen doormaken.

Het framework moet op de client-side draaien, niet serverside.

Het framework moet een community hebben. Een grote community maakt het makkelijk zonder (dure) ondersteuning aan de slag te kunnen.

Het framework moet features hebben die het makkelijk maakt om functionaliteit te implementeren en hergebruiken, zoals dependency injectie en dependency management.

Het heeft de sterke voorkeur dat gebruikte frameworks al in-house worden gebruikt om de kennisdrempel tussen teams te verlagen.

## Ext.JS

Is het huidige framework waarop de front-end van concentrator is gebouwd. De ontiwkkelaar van Ext is Sencha, een bedrijf dat zich specialiseert in het ontwikkelen van front-end frameworks.

In tegenstelling tot andere frameworks komt ext.js inclusief een layout, die bij de basisversie niet responsive is. Een ander, neiuwer en duurder product is Sencha Touch, dat met functionaliteit voor swipe en etc komt.

## AngularJS

Veruit het meest populaire framework. In 2007 opgekocht door google, en sindsdien in actieve ontwikkeling. Angular 2 staat op het moment van schrijven op de valreep.

Angular is breed opgezet, en bevat onder andere functionaliteit voor dependency injection, het managen van object levenscyclussen tot het routeren van URL’s.

Door de populariteit van Angular bestaat er ook een grote community; er zijn veel onofficiële codevoorbeelden aanwezig.

Voornaamste nadeel van Angular is dat Angular2 volledig herschreven t.o.v. de vorige versie. Dit houdt in dat enig proof-of-concept gebaseerd op Angular1 zal moet worden herschreven naar Angular2.

## Knockout.js

## React

React is een facebook framework, en onderdeel van de Flux suite. Het is gericht op het snel en makkelijk maken van webapplicatie.

De kern van de applicatie bestaat uit het gebruik van de mixin, een javascript patroon dat

## Meteor

MEteor.js

## Deelconclusie matrix

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Actief | Mature | In gebruik | Client render | Server | Community | DI/DM | gratis | Totaal |
| AngularJS2 | V | X | V | V | V | V | V | V | 7 |
| Ext.JS | X | V | V | V | X | X | ? | X | 3,5 |
| Knockout.js | X | X | X | V | V | X | X | V | 3 |
| Meteor | V | X | X | V | V | V | X | V | 5 |
| Backbone.js | V | X | X | V | V | O | X | V | 4,5 |
| Vue.js | V | X | X | V | X | O | X | V | 3,5 |
| React | V | V | X | V | V | V | V | V | 7 |
| Ember | V | V | X | V | V | O | V | V | 6,5 |

De matrix geeft duidelijk aan dat AngularJS en React goed aan de eisen voldoen. Er is gekozen voor AngularJS, voornamelijk omdat er al operationele kennis aanwezig is, en omdat migreren van Angular1 naar Angular2 met ngUpgrade makkelijk kan.

# Hoe maak ik de front-end bruikbaar op mobiele apparaten?

Een van de randvoorwaarden voor acceptatie van het PoC is het toegankelijk maken van de applicatie op mobiele systemen. Primaire doelsystemen zijn de desktop / laptop, secundair zijn tablets en forse smartphones.

## Visueel

Een van de problemen is dat een mobiel beeldscherm over het algemeen een stuk kleiner is dan een computermonitor. Een klein beeldscherm met een slechte CSS zal over het algemeen proportioneel schalen ‘en/’of horizontale scrollbars weergeven.

Schalen verkleint de tekst dusdanig dat de tekst onleesbaar wordt. De gebruiker word dan geforceerd om in te zoomen of in twee richtingen te scrollen, wat frustratie en teleurstelling veroorzaakt

Daarnaast kunnen mobiele telefoons dankzij hun gyroscoop makkelijk draaien. Dit maakt het voor mobiele gebruikers gemakkelijk het apparatuur in de lengte of breedte te gebruiken.

De oplossing is het gebruikmaken van een ontwerp dat als het ware kan “inklappen”. Door alle elementen een relatieve eenheden breedte te geven is, kan enig overvloeisel makkelijk onder elkaar worden gezet zonder te veel discrepantie in onderdelen te veroorzaken. Deze ontwerpmethode heet responsive design, en maakt het gemakkelijk dezelfde inhoud op desktop en mobile te tonen.

## Invoer

Mobiele platformen beschikken over het algemeen over een touchscreen met multitouch, en that’s it. Tekstuele invoer is over het algemeen tijdrovend t.o.v. het werken met een toetsenbord.

### Toetsenbord

Mobiele platformen beschikken over het algemeen over een softwarematig toetsenbordje. Deze is geschikt voor kleine invoer, niet om mee te werken.

### 

## Gebruik

Sommige functionaliteit is simpelweg niet te tonen zonder

# Hoe kan ik complexe data transparant maken?

De data onderliggend aan de concentrator is relatief complex. Om deze data en onderlinge relaties inzichtelijk te maken moet er worden gekeken naar specifieke oplossingen voor relationele data.

## Layout

Er zijn v

## Grid

## Zoekfunctionaliteit

# Hoe maak ik de applicatie dynamisch meertalig?

Omdat de klanten van de opdrachtgever in verschillende landen actief zijn, is het noodzaak de applicatie makkelijk meertalig te maken, en makkelijk talen toe te kunnen voegen.

## Angular niet afdoende

De officiele documentatie laat alleen statisch geladen internationalisatie zien. Dit is niet genoeg om de requirements te vervullen, en daarom zelf wat geschreven. Het gaat hier om een localisatie service voor Angular1.

## Symbooltekst

Talen worden in JSON formaat van de webserver geladen. Deze JSON files bestaan uit Key/Value pairs met het symbool als key en de uitwerking als value.

In de code zelf is een lijstje van ondersteunde talen aanwezig met relevante data (ISO 639-1) en kunnen dynamisch worden geladen. Door de i10n service in de Scope te laden wordt het heel makkelijk om aan de hand van symbolische sleutels de vertaling te vinden.

## Internationalisatie

Internationalisatie of i18n (welke angular1 wel ondersteund) is ook niet afdoende voor de applicatie, omdat deze meerdere valuta-eenheden simultaan moet ondersteunen en dus dynamisch moet kunnen inladen.

# Hoe integreer ik het project in bestaande workflows en bouwsystemen?

De huidige ontwikkelings workflow is sterk geïntegreerd in Visual Studio 2013. Om enige transitie zo makkelijk mogelijk te maken is het de noodzaak verschillenden processen tegen visual studio te integreren.

# Hoe maak ik de applicatie herbruikbaar voor andere teams?

Er is binnen diract-it een drang naar standaardisatie van onderdelen. Huidige producten zijn afzonderlijk van elkaar ontwikkeld en delen weinig code. Elke afdeling gebruikt een eigen codebasis voor de front-end.

## Documentatie

Ontwikkelaars zijn sterk afhankelijk van kwali

## Ontkoppeling van functionaliteit en gebruik

# Onderzoek gebruikers en ontwikkelaars

## Ontwikkelaars

Ontwikkelaars zijn primair C# gericht, maar javascript kunde is in andere teams aanwezig.

## Gebruikers