Interviews VI-Company



Naam: Oscar Veldman  
Datum: 21-09-2016  
Versie: 1.0

Contents

[Voorbereiding interview 3](#_Toc462212668)

[Vragen/Antwoorden 4](#_Toc462212669)

[Standaardvragen 4](#_Toc462212670)

[FRT – Extra vragen 5](#_Toc462212671)

[Keuze case 6](#_Toc462212672)

[Case 1 – Meertaligheid 6](#_Toc462212673)

[Case 2 - Identity Provider 6](#_Toc462212674)

[Overige cases 6](#_Toc462212675)

[Advies case 6](#_Toc462212676)

[Reflectie 7](#_Toc462212677)

[Bijlage 8](#_Toc462212678)

[Case 1 – Meertaligheid 8](#_Toc462212679)

## Voorbereiding interview

We moesten in de eerste weken een interview gaan houden. Deze moesten we ook gaan voorbereiden en na elke interview verbeteren. In de volgende alinea’s staan alle informatie wat we allemaal gedaan hebben voor de voorbereiding. Deze voorbereiding deed ik samen met Jeffrey in overleg, zodat we ook elkaars meningen konden horen.

We hadden van tevoren een lijstje met vragen bedacht. De volgende vragen zie je in het volgende hoofdstuk met de antwoorden. Om meer het idee te krijgen waar de projecten over gingen, hadden we ook veel door de code en op het internet zitten kijken. Sommige van die informatie hadden we ook gebruikt om zelf uit te proberen in projecten. Bijvoorbeeld hoe controllers en routes werken in het MVC-model. Natuurlijk hebben we nog meer onderwerpen gekeken. Ook wilde we op sommige onderwerpen meer vragen gaan stellen. Deze interviews werden ook opgenomen om achteraf te luisteren, wat er allemaal gezegd werd. Deze antwoorden werden ook gedigitaliseerd in een docx bestand. Ook hield een van de twee interviewers een notulen bij en de ander deed de vragen. Zodat het makkelijker werd om de audio terug te luisteren. Dan weet je wat eraan kwam. Deze notulen gebruikten we dan voor het beantwoorden van de vragen. De notulen is opgebouwd in welke ideeën ze hebben, de ideeën uitgelegd en alle antwoorden op de vragen die in het interview waren voorgekomen.

Het idee was dat Jeffrey en ik elk interview met twee developers gingen voeren. We wilden graag één front-end en één back-end developers. Hierdoor konden de twee developers met elkaar in discussie aan gaan om meer informatie te krijgen. We hadden ook het idee om met een hele grote groep te gaan interviewen. Dit zou tot veel discussie kunnen leiden. Maar we hebben toch gekozen voor twee personen per keer, omdat we telkens het interview konden verbeteren en de informatie die we kregen konden laten zinken. We deden de interviews ook goed verspreid over de andere halve week, zodat we de interviews konden verwerken in een mooi resultaat.

We bedachten om het FRT eerst interviewen, omdat FRT een overzicht heeft van alle projecten. Met deze informatie kunnen we volgende keer betere en/of gerichtere vragen te kunnen stellen aan de andere programmeurs. Ook wilden we weten van het FRT waar veel problemen voorkomen. We wilde graag horen wat de meningen zijn van de developers. Als de developers niet veel input gaven. Dan hadden we ook een lijstje bedacht om wat ideeën te geven. Hier konden ze dan ook hun eigen mening op geven.

Als we alle interviews gehad hadden en alle informatie verwerkt hebben. Daarna gaan we een groot overleg houden over welke optie het beste is om een opdracht van uit te werken. Van tevoren van deze overleg gaan we een paar cases uitwerken, zodat we tijdens het overleg de case die we gekozen hebben verder kunnen uitbreiden en verbeteren.

## Vragen/Antwoorden

Hieronder staan alle antwoorden op de vragen die we hebben bedacht. Dit zijn antwoorden van alle interviews bij mekaar.

### Standaardvragen

1. **Wat voor system onderdeel moeten jullie vaak ontwikkelen in jullie projecten?**

* Caching
* Charts
* Contact formulier
* E-mail versturen
* Formatting
* Genereren van bestanden
* Identity provider
  + Inclusief inloggen/registeren
* Prijs informatie
* Quotes
* Security
* SignalR
* Tabellen en sorteren daarvan
* Transactie maken
* Unittesten
* Validatie
* Vertaalbaarheid
* Watchlist

1. **Waar hebben jullie het meeste baat bij?**

* Vertaalbaarheid
  + Er wordt soms aangegeven dat het lastig was alle talen bij te houden, welke al vertaald waren
* Identity provider
  + Bijna elke website heeft een login, registeren en email verzenden. Dit zonde om telkens opnieuw te maken.
* Validatie
  + Er werd aangegeven dat dit ook veel terug komt

1. **Zijn er nog onderdelen die hier niet bij staan maar wel een goed onderdeel is dat gestandaardiseerd kunnen worden?**

*We hadden al een lijstje gemaakt. Hiermee vroegen we of ons af welke ideeën aangevuld konden worden. Deze staan ook in het lijstje bij vraag 1.*

De nieuwe ideeën waren: Formatting, Identity provider, prijs informatie, SignalR, unittest   
 automatiseren en vertaalbaarheid(Labels).

1. **Welke onderdelen hebben jullie al gestandaardiseerd in een Nuget package?**

* Logging
* Code genereren C#
* Authenticatie voor het product van VI. Waar alleen klanten bij kunnen
* Caching
* Excel
  + Checkt of het Excel bestand wel goed is
* Performance
  + Kijken hoelang een pagina erover doet om te laden. Dit wordt weergegeven in een grafiek
* Validatie

1. **Welke methodieken hebben jullie gebruikt voor de ontwikkeling en waarom?**

* Database server in datacenter
* Fileserver
* Loadbalancer (verdeeld het werk voor meerdere webservers)

1. **Missen jullie iets in de koppeling tussen back-enders en front-enders?**

* Soms wordt er in de front-end iets gevalideerd en in de back-end het precies hetzelfde gevalideerd. Dit is een beetje zinloos.
* Er wordt in de back-end data verstuurd uit de database die de front-end nodig heeft. Later heeft de front-end niet meer nodig. Daar wordt het label verwijdert. Maar vaak wordt er niks in de back-end verandert. Waardoor het wel steeds weer uit de database wordt gehaald. Dit kan effect hebben op performance.

1. **Zijn er nog dingen die wij moeten weten en nog niet beantwoord zijn of uitgelegd zijn?**

*Dit is al verwerkt in de overige vragen. Tijdens het interview hadden we ook gevraagd hoe sommige programmeer of software werkten.*

### FRT – Extra vragen

1. **Welke handelingen verrichten jullie vaak?**

* Kleine nieuwe features
* Bugs oplossen
* Plannen

1. **Welke bugs komen jullie vaak tegen?**

* Langzaam verwerken
* Software bugs
* Excel sheets
* Performance

1. **Hoe pakken jullie deze bugs aan?**

Er wordt eerstaangewezen waar de fout gebeurt. Vervolgens wordt er naar de bugs gezocht.   
Deze wordt vervolgens opgelost.

## Keuze case

Uit de interviews kwamen eigenlijk twee cases veel naar voren, namelijk meertaligheid en de identity provider.

### Case 1 – Meertaligheid

De programmeurs van VI vinden soms lastig om alle labels in verschillende talen te vertalen. Als je bij de ene taal toevoegt, dan moet je dat bij de andere ook doen. Soms wordt heel onoverzichtelijk. Daarom is er een programma nodig die het allemaal overzichtelijk houdt en alles makkelijk kan aanpassen.

Hierdoor willen we een programma maken dat je heel makkelijk kan zien welke labels nog vertaald moeten worden en als je bijvoorbeeld een label niet meer nodig hebt. Dat het gelijk voor de andere talen wordt verwijderd. We denken dan dat het dan overzichtelijker wordt om de labels te vertalen. Verder willen ook dat je met het programma de route kan vertalen. Je moet het label ook heel makkelijk kunnen vinden. Zie verder ook bij de bijlage Case 1.

### Case 2 - Identity Provider

De developers bij VI Company moeten bij elk project die zij maken een login systeem maken en daarbij moeten zij ook vaak rollen aanmaken. Omdat dit vaak hetzelfde is bij elk project is dit zonde om dit telkens weer opnieuw te ontwikkelen. Als het een standaard package zou worden zou dit veel tijd schelen voor de developers.

We willen een systeem maken waar je alleen een pakket hoeft te downloaden. Waardoor je gelijk een inlog systeem hebt met een registratiesysteem. Dit systeem moet je met verschillende rollen met verschillende rechten kunnen inloggen. Er wordt een automatisch e-mail verstuurd. Dit wordt de Identity Provider.

### Overige cases

De andere onderwerpen zie je ook bij vraag 1 bij het vorige hoofdstuk. De meeste van de onderwerpen die hierin staan zijn of te klein of te groot qua opdracht. Natuurlijk kan je de te grote opdrachten ook splitsen, maar uit de interviews bleek dat de vorige cases meer prioriteit ligt. Ook lijkt het ons interessanter en uitdagender om met de vorige cases bezig te zijn, omdat we dan meer met het framework ASP.NET CORE MVC bezig zijn.

### Advies case

We gaan voor Case 1 omdat Case 2 ons ook wat te groot lijkt. Hier komt veel bij kijken, bijvoorbeeld database en meerdere rollen. Bij het inloggen komt ook meer kijken dan alleen inloggen. Het e-mail verzenden, beveiliging, registeren. Dit worden heel veel templates voor een packets.

Ons lijkt Case 1 het interessants, omdat we dan een volledig project kunnen maken met meerdere aspecten. Ook schatten we in dat we dit project binnen het stage tijdperk kunnen halen. Hebben hier ook een helder idee over, wat het resultaat moet worden. Hierbij denken we dat het ook een handige tool kan worden voor de programmeurs. Een tool dat je een hoop tijd bespaard door niet meer zo lang te zoeken door de labels.

## Reflectie

Voor mijn gevoel ging de meeste interviews goed. Ik denk dat Jeffrey bij de interviews een beetje in de voorgrond bleef. Daardoor had de andere interviews meer structuur in. Daardoor viel ik bij de laatste interview een beetje in de mand. Hierdoor was ik iets te zenuwachtig dan de bedoeling was. Ik had de details van de interviews beter moeten voorbereiden. Maar we hebben genoeg goede informatie binnen gekregen om een goed project op te zetten.

Misschien hadden we achteraf op sommige onderwerpen wat dieper op in moeten gaan, maar dat kwamen we achter toen we de case aan het maken waren. De volgende keer is het beter voor mij om de goede ideeën al halve wegen een beetje uit te werken. Voor de rest is de informatie die we konden kregen, wel de goed informatie om een besluit te maken. Hieruit bleek ook welke veel prioriteit heeft. Na overleg met Jeffrey, Niels en Sebastian hebben gekozen voor Case 1. Behalve wat hierboven omschreven staan, is het gegaan hoe we voorbereid hebben. Mijn conclusie is dat ik meer moet durven vragen.

## Bijlage

### Case 1 – Meertaligheid

Hier staat de uitwerking van Case 1 die we tot nu toe hebben De scope en acceptance criteria moeten nog verder uitgewerkt worden. De rest kunnen we nog over nadenken of het beter kan.

#### Waarom?

De programmeurs van VI vinden soms lastig om alle labels in verschillende talen te vertalen. Als je bij de ene taal toevoegt, dan moet je dat bij de andere ook doen. Soms wordt heel onoverzichtelijk. Daarom is er een programma nodig die het allemaal overzichtelijk houdt en alles makkelijk kan aanpassen.

#### Doelstelling

De software moet makkelijk labels kunnen veranderen voor meerdere talen. Als je bij de ene taal een label toevoegt, dan herkent het programma gelijk dat je bij de andere talen ook moet vertalen. Stel dat er een label niet vertaald is, dan komt daar in de plaats een waarde van de default taal. Dit geld ook voor het verwijderen van labels. Alleen de bevoegde personen mogen erbij kunnen overal. De bevoegde personen kunnen klanten en werknemers van VI-Company zijn.

Het programma moet in 1 opslag duidelijk zijn voor de gebruiker. Alle labels die kunnen gevonden door een goed werkende zoekbalk bijvoorbeeld. De labels moeten op meerdere manieren gevonden worden, bijvoorbeeld op naam. De labels kan je ook groeperen, bijvoorbeeld met getallen. Je hebt soms bij een winkelmandje dat je 1 product hebt, maar ook 2 producten hebt. Het meervoud moet met 1 label aangegeven kunnen worden.

Ook kan je in het programma zien hoe vaak een label wordt gebruikt en aangeroepen. Als een label niet vaak wordt aangeroepen, dan geeft het programma automatisch aan dat het label niet of nauwelijks wordt gebruikt. Stel een label heeft meerdere dezelfde vertalingen. Dan wordt er gevraagd of je niet wil samenvoegen.

In de labels wordt soms ook opmaak gebruikt. Je wilt niet als je een bijvoorbeeld link wilt plaatsen, dat je meerdere labels moet aanmaken. Dus er komt een opmaak systeem in de labels. Dit kan je aan de hand van een preview zien, hoe het eruit komt te zien. Ook kan je hier variables maken zodat je gegevens die niet vast staan kunt gebruiken. De variabele worden wel met een duidelijke naam aangegeven, zodat je weet wat je verwachten staat.

De applicatie runt eerst op alleen de produceer omgeving. Daarna kijken we of alles tegelijk aangepast kan worden op alle omgevingen. Ook misschien voor de toekomst dat er een front-end in line editing komt. Als op de pagina van de website een label aanklikt, dat je dan precies kan zien wat voor naam het label heet.

#### Puntsgewijs

* Toevoegen van labels
* Vertaling veranderen van labels
* Verwijderen van labels
* Sorteren van labels
* Preview van labels
* Statistieken van labels
* Placeholder variabele
* Labels groeperen
* Opmaak labels
  + Links & Dik gedrukt bijvoorbeeld
* Inlezen van json
* Wegschrijven van json
* Mark down
* Zoekbalk
* Duidelijk hoe het werkt.
  + Na 1 of 2 keer gebruiken
* Runtime inlezen

#### Scope

* De software is bedoeld voor de programmeurs van VI Company.
  + De gui hoeft niet fantastisch eruit te zien
* De routes worden buiten beschouwing laten
* Local project zonder login

#### Software

Hieronder staan meerdere mogelijkheden voor het maken van software

* Een webapplicatie (ASP.NET Core)
  + Json
* Een webapplicatie (ASP.NET Core)
  + Database
* \*.Exe voor Windows. (Programma local)

#### Login rollen

* Iemand die alles mag aanpassen
* Iemand die alleen mag kijken?

#### The acceptance criteria

Snelheid:

Uiterlijk (GUI):

Tests:

Code leesbaarheid:

#### User stories/Requirements

Hieronder komen al een paar voorbeelden van user stories. Dit worden er later meer.

1. Een gebruiker kan een label kunnen toevoegen, omdat het label onthouden moet worden
2. De gebruiker kan switchen van taal, omdat hij bij de talen kan om ze te vertalen
3. Een gebruiker kan een label vertalen, omdat je meerdere vertalingen hebt
4. Een gebruiker kan een nieuwe taal invoegen, omdat er meer talen bij kunnen komen.
5. Een gebruiker kan een module aanmaken, omdat het netjes gesorteerd blijft.
6. Een gebruiker kan meerdere groeperingen voor een label maken voor 1 label, omdat er een meervoud en enkelvoud van woorden gebruikt kan worden
7. Een gebruiker kan makkelijk zien welke labels nog niet vertaald zijn, omdat dat dan overzichtelijk blijft.
8. Een gebruiker kan labels opzoeken, omdat hij snel bij het label kan zijn.
9. Een gebruiker kan een label een opmaak meegeven, omdat hij dan geen 3 labels hoeft aan te maken voor 1 zin
10. Een gebruiker kan een variabele naam in een label zetten, omdat er soms een tekst niet vast staat van tevoren
11. Een gebruiker kan een preview van het label zien, omdat hij dan gelijk kan checken of de label goed uitziet
12. Een gebruiker kan een label kunnen verwijderen, omdat het label niet meer nodig is
13. De gebruiker kan makkelijk zien welke labels niet meer in de site worden gebruikt, omdat de gebruiker makkelijk kan beslissen of hij het niet meer nodig heeft
14. Een gebruiker kan zien hoe vaak een label is bekeken, omdat hij kan zien of het label nog relevant is.
15. Een gebruiker kan inloggen, omdat hij kan laten zien wie hij is
16. Het systeem kan voor niet vertaalde labels een default taal invullen, omdat de labels niet leeg blijven staan
17. Een gebruiker kan zien hoeveel labels dezelfde omschrijving hebben, omdat hij kan beslissen of ze samen gevoegd moeten worden
18. Een gebruiker kan een label naam veranderen, omdat hij met een betere naam komt.
19. De gebruiker kan de labels sorteren op verschillende tabelkolommen, omdat de gebruiker het makkelijker kan opzoeken
20. Een gebruiker kan uitloggen, omdat niet iemand anders zomaar er in kan.
21. De administrator kan nieuwe accounts aanmaken, omdat er meerdere mensen kunnen inloggen
22. Een gebruiker kan een label veranderen on run time, omdat dat veel minder tijd kost voor het opnieuw deployen
23. Een gebruiker kan een nieuwe taal verwijderen, omdat er soms talen niet meer nodig zijn.
24. Een gebruiker kan labels exporteren, omdat hij het dan kan met de labels aantekeningen kan maken.