Interview

## Voorbereiding

We hadden van tevoren een lijstje met vragen bedacht. De volgende vragen zie je in het volgende hoofdstuk. Om meer het idee te krijgen waar de projecten over gaan, hebben we ook veel door de code en op internet zitten kijken. Sommige van die informatie hebben 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 worden ook opgenomen om achteraf te luisteren, wat er allemaal gezegd werd. Deze antwoorden worden ook gedigitaliseerd in een docx bestand. Ook houdt een van de twee interviewers een notulen bij en de ander doet de vragen. Zodat het makkelijker is om de audio terug te luisteren. Dan weet je wat eraan komt.

Het idee was dat Jeffrey en ik elk interview met twee developers gingen voeren. 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.

## Vragen/Antwoorden

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

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
* Performence
  + Kijken hoelang een pagina erover doet om te laden. Dit wordt weergegeven in een grafiek
* Validation

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

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

1. **Welke veel voorkomende problemen lopen jullie tegen aan?**
2. **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

### 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.

### 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.

### Advies case