Cases - VI Company



Made by: Oscar Veldman & Jeffrey Hoogland  
Datum: September 2016  
Versie: 1.0

Table of Contents

[Case 1- Meertaligheid 3](#_Toc462124229)

[Waarom? 3](#_Toc462124230)

[Scoop 3](#_Toc462124231)

[Puntsgewijs 3](#_Toc462124232)

[Wat kan het niet 3](#_Toc462124233)

[Software 4](#_Toc462124234)

[Login rollen 4](#_Toc462124235)

[User stories/Requirements 4](#_Toc462124236)

[Case 2 – Identity Provider 5](#_Toc462124237)

[Waarom? 5](#_Toc462124238)

[Scoop 5](#_Toc462124239)

[Software 5](#_Toc462124240)

[Requirements 5](#_Toc462124241)

[Case 3 – Kleinere projecten 6](#_Toc462124242)

# Inleiding

We hebben in september 1,5 week programmeurs van VI Company geïnterviewd. Hier moesten een aantal cases uitkomen. We hebben twee cases die daadwerkelijk uitgevoerd kan worden. Of we moesten een project maken met allemaal kleine projectjes. Maar we hebben hieronder 2 cases uitgewerkt. De eerste case die gaan we uitvoeren. Dus die is het belangrijkst om te begrijpen. Bij de eerste case hebben we ook nog een aantal diagrammen bijeen toegevoegd. Zodat je het misschien het beter begrijpt.

# Case 1- Meertaligheid

## 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. Vroeger was er een backoffice met de mankementen, maar die is niet goed. We moeten een hele makkelijke begrijpbare software nodig.

## 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. De labels kunnen verdeeld worden in groepen. Deze groepen kunnen per pagina’s 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. De programmeur moet nadenken of het in de toekomst niet meer veranderd. Als het toch gebeurt, dan moet hij een nieuw label aanmaken.

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. Misschien later dat alle omgevingen tegelijk wordt up-ge-date met dit stukje software. Alleen een nadeel van dat is, is dat je dan meer labels moet gaan aanmaken voor een nieuwe naam. Maar het kan ook een voordeel zijn, omdat ons programma ook gaat gebruiken van of een label vaak wordt gebruikt of niet. 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

*Snelheid van het runnen van alle code*

* For example (2 seconde)

### Uiterlijk (GUI)

* Het moet duidelijk

### Tests

* Unittest
* Clean Code Test

### Code leesbaarheid

*De regels of code schrijven*

* Alles in het Engels
* De variabele begint met een kleine letters
* De class naam begint altijd met een hoofdnaam

## 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 het 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.
25. Een gebruiker kan de file locatie van labels veranderen, omdat de file locatie niet vast staat

## Research

### NuGet

**Maken van NuGet**  
<https://docs.nuget.org/create/creating-and-publishing-a-package>

### Multi language

**Met \*.resx file**  
<http://www.ezzylearning.com/tutorial/developing-multi-language-web-sites-with-asp-net>

**Met \*.Json**

* **Performance**<http://stackify.com/top-11-json-performance-usage-tips/>
* **Zie project ING.SME**
* **Variable**<https://www.packer.io/docs/templates/user-variables.html>

### ASP.NET Core

**De basisinformatie over APS.NET Core**  
<https://docs.asp.net/en/latest/>

**Maak een map of bestand aan in C#**<https://msdn.microsoft.com/en-us/library/as2f1fez.aspx>

**Schrijf naar een \*. JSON-file**<http://stackoverflow.com/questions/22536818/how-to-write-to-a-json-file-using-c>

### Charts

**Make charts in C#/ASP.NET**  
<http://www.asp.net/web-pages/overview/data/7-displaying-data-in-a-chart>

### Markdown

**Uitleg**  
<https://en.wikipedia.org/wiki/Markdown>

**Cheatsheet**<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

### JSON

**Nestelen**<http://stackoverflow.com/questions/26992587/how-to-parse-nested-json-data-structure>

### Mouseover

**Title**  
<http://sites.cognitivescience.co/knowledgebase/resources/using-google-sites/creating-mouseover-text-with-html>

## Implementatie

* Nadenken over deploy momenten
* Web of lokaal programma
  + In een package of andere manier
* Opslaan van labels
* Eventueel libaries
* Database?
* Grafieken of getallen voor statistieken
* Preview
* Hoe we gaan nestelen
* Misschien alternatieve van Markdown
* Of we een login nodig hebben
* Planning
* Welke tools

# Case 2 – Identity Provider

## Waarom?

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.

## Scoop

De identity provider houdt in dat er een gebruiker aangemaakt kan worden en rollen aan deze gebruiker toegevoegd kan worden. Omdat dit nu elk project handmatig gebeurd moet er een package gemaakt worden die ingeladen kan worden in elk project en hierdoor automatisch een identity provider in het project zit. Deze identity provider moet dan een registratie en inlogsysteem bevatten. Ook moet het mogelijk zijn om rollen toe te voegen en deze aan gebruikers toe te wijzen.

## Software

Er moet een package gemaakt worden van de identity provider. Het moet een package worden die door de projecten ingeladen kan worden.

## Requirements

1. De gebruiker kan zich registreren
2. Het systeem stuurt een activatiemail naar de gebruiker.
3. De gebruiker kan zich inloggen
4. Het systeem stuurt een link naar de gebruiker, als de gebruiker op de knop wachtwoord vergeten drukt.
5. De administrator kan de inloggegevens veranderen (bijvoorbeeld i.p.v. gebruikersnaam een email)
6. De administrator kan het de registratievelden veranderen (bijvoorbeeld datum dd-mm-jjjj 🡪 mm-dd-jjjj)
7. De administrator kan gebruikers aanmaken
8. De administrator kan gebruikers wijzigen
9. De administrator kan gebruikers verwijderen
10. De administrator kan rollen aanmaken
11. De administrator kan rollen verwijderen
12. De administrator kan gebruikers rollen toewijzen
13. De administrator kan gebruikers rollen afnemen
14. Het systeem moet real-time valideren als de gebruiker zich wilt registreren.
15. De administrator moet een bepaald type wachtwoord kunnen toevoegen. Zodat gebruikers ook met bijv. een pin kunnen inloggen.
16. Het systeem blokkeert de gebruiker, als de gebruiker x aantal keer zijn inloggegevens verkeerd heeft ingevoerd.

# Case 3 – Kleinere projecten

* Charts
* Prijs informatie
  + Binnenhalen en weergave quotes
* Validatie
* E-mail versturen
* Bestanden importeren
* Genereren van bestanden
* Formatting
* Security
* Caching
* SignalR
* Unittesten in een packets
* Tabellen en sorteren daarvan
* Contact formulieren
* ~~Nui-Company~~