Cases - VI Company



Made by: Oscar Veldman & Jeffrey Hoogland

Table of Contents

[Case 1- Meertaligheid 3](#_Toc461713463)

[Scoop 3](#_Toc461713464)

[Puntsgewijs 3](#_Toc461713465)

[Wat kan het niet 3](#_Toc461713466)

[Software 3](#_Toc461713467)

[Login rollen 3](#_Toc461713468)

[User stories 4](#_Toc461713469)

[Case 2 – Identity Provider 5](#_Toc461713470)

[Case 3 – Kleinere projecten 6](#_Toc461713471)

# 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.

## Scoop

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. Deze worden veranderd in productie en op de lokale machines. Dit geld ook voor het verwijderen van labels. Deze labels moeten ook steeds opnieuw gesorteerd worden. Deze worden ingelezen en weggeschrijven in json. Alleen de bevoegde personen mogen erbij kunnen overal. De bevoegde personen kunnen klanten en werknemers van VI-Company.

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 of op route.

### Puntsgewijs

* Toevoegen van labels
* Vertaling veranderen van labels
* Verwijderen van labels
* Sorteren van labels
* Inlezen van json
* Wegschrijven van json
* Mark down
* Routes makkelijk vertalen
* Zoekbalk
* Duidelijk hoe het werkt.
  + Na 1 of 2 keer gebruiken
* Runtime inlezen

## Wat kan het niet

## 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?

## User stories

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 kan worden
2. Een gebruiker kan een label kunnen verwijderen, omdat het label niet meer nodig is
3. Een gebruiker kan een label vertalen, omdat je meerdere vertalingen hebt
4. Een gebruiker kan een route vertalen, omdat je meerdere vertalingen hebt
5. Het systeem geeft automatisch aan of een nieuwe vertaling is voor de andere talen, omdat dat overzichtelijk blijft.
6. Een gebruiker kan een label veranderen on run time, omdat dat veel minder tijd kost voor het opnieuw deployen
7. Een gebruiker kan inloggen, zodat hij kan laten zien wie hij is
8. Een gebruiker kan uitloggen, zodat niet iemand anders zomaar er in kan
9. Een gebruiker moet labels kunnen opzoeken, zodat hij snel bij het label kan zijn.

# Case 2 – Identity Provider

# 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