# Front-end

## Software

De applicatie maakt gebruik van het Angular framework. Dit framework is enorm krachtig en met behulp van Ionic zeer geschikt om mobiele applicaties te schrijven. Ionic zorgt ervoor dat de applicatie volledig in html en typscript geschreven kan worden om daarna gecompileerd te worden naar een native IOS- of Android-app. Niet alleen kan dezelfde broncode gebruikt worden voor het compileren voor beide besturingssystemen, Ionic heeft ook voldoende 'mobile-friendly’ componenten die gebruikers-klaar zijn en de develop-tijd sterk verkorten.

## Structuur van de applicatie

### Filosofie achter de structuur

Steal-my-meal is een single-page applicatie. Alles wordt opgebouwd boven op de startpagina. Dit heeft verschillende voordelen. Het navigeren van de app wordt heel natuurlijk voor de gebruiker op deze manier. Het sluit ook eventuele wachttijden uit voor het laden van pagina’s. De meerderheid van de applicatie wordt geladen wanneer hij voor het eerst gestart wordt. Dit is het moment dat de user het meeste geduld heeft. Moest de applicatie tussen pagina’s laden, kan de aandacht spanne van de user onderbroken worden en wordt de hele user-ervaring beïnvloed.

### Mappenstructuur

Een logische mappenstructuur is zeer belangrijk in een project dat meerde pages en componenten door elkaar gebruikt. Gezien dat pages en componenten vaak van verschillende plaatsen aangesproken worden, wordt er gebruik gemaakt van een contextuele mappenstructuur. Een hiërarchische structuur zou in dit geval niet de correcte keuze zijn.

Pages en componenten zijn verdeeld op basis van hun inhoud, zodat het duidelijk is waar welk component zich bevindt.

### Modals

Bijna elke pagina in de app wordt geopend via een modal. Hiervoor wordt gebruik gemaakt van een modalservice. Deze service is in staat om modals te openen, parameters mee te geven naar de gevraagde pagina’s en modals te sluiten. Als er in een modal contact gelegd wordt met de databank, krijgt de user een laadscherm te zien, tot de data volledig verwerkt is.

### Styling

Voor de styling maakt elke component gebruik van zijn eigen sass bestand om pagina-unieke elementen te stijlen. Algemene elementen, of elementen die vaak voorkomen, worden gestijld in een globaal sass bestand. Doorheen de applicatie wordt er ook gebruik gemaakt van kleur variabelen, die geïmporteerd kunnen worden door de pagina’s.

### Models

Doorheen de app wordt er vaak gebruik gemaakt van dezelfde objecten. Om dit zo efficiënt mogelijk te doen zijn er van deze objecten models gemaakt. Er is op deze manier bijvoorbeeld een adres klasse, waar de straat, postcode, coördinaten, … gedeclareerd en geinitialiseerd zijn.

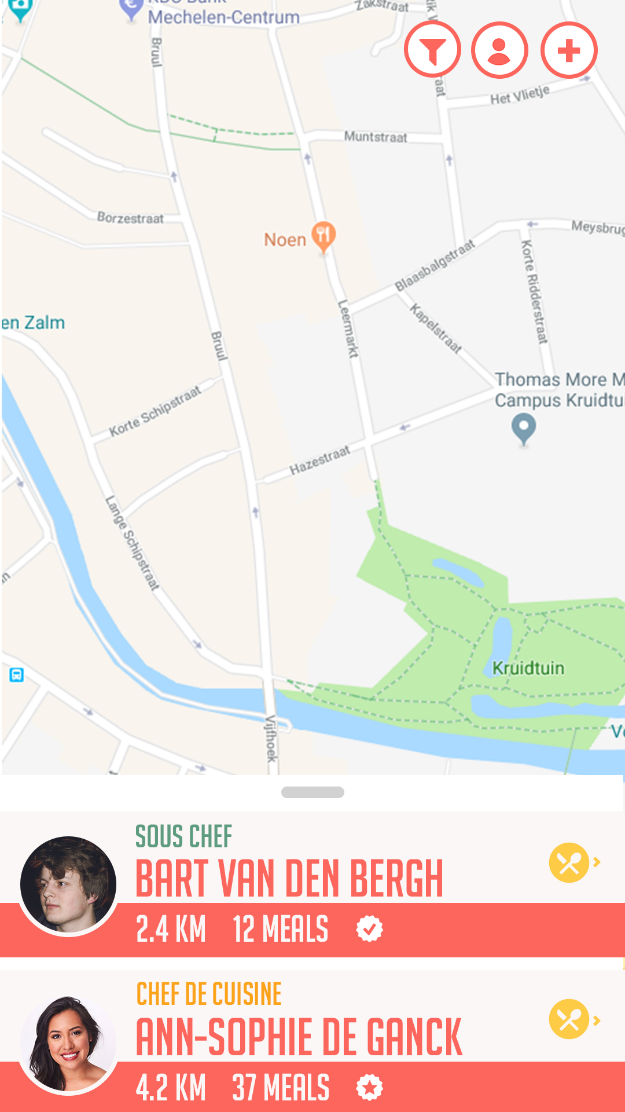
## UI

Voor de definitieve lay-out beslist was, waren enkele opties om de applicatie op te bouwen. Een meer traditionele lay-out werkte met een navigatie aan de onderkant van de app. Er is uiteindelijk gekozen om deze navigatie niet te gebruiken, maar voor een lossere meer dynamische optie te gaan.



Oude navigatie

Voor er geprogrammeerd was, is er een mockup gemaakt van de uiteindelijk gekozen layout. Dit helpt een beeld te scheppen van de nodige functionaliteiten en structuur van de app. Samen met het ontwerpen van de layout is er ook een kleurenstudie gemaakt om de sfeer van de app te bepalen.



1Uiteindelijke applicatie

2Origineel ontwerp