## horizontale lijn



Snooty Snakes

25-10-2020

**─**

Analyse project veeteelt management

Dave Saenen, Wouter Pardon, Niels De Cat, Fabio Puissant

Uhasselt Martelarenlaan 42, 3500 Hasselt

# Probleemstelling

**Productieproces**

Voor een varkenshouder is het moeilijk om data over zijn bedrijf bij te houden op een georganiseerde manier. Zo zijn er 5 cruciale parameters om de evolutie van een varkensbedrijf op te volgen. (Voedingspatroon, Groei (kg/dag), Kosten van het voer, Opname voedsel/dag, Kwaliteit vlees) Deze spelen een belangrijke rol voor het opvolgen van het productieproces.

Deze parameters worden per groep opgevolgd, zodat veranderingen in groepen zichtbaar worden.

Om dit te verduidelijken gaan we uit van het volgend voorbeeld: een varkenshouder wil een nieuw voedsel type uittesten, dit type kan gevolgen hebben voor de groei van de varkens maar ook voor de kosten van het voer en de opname van voedsel per varken of per dag.

Aan de hand van dit voorbeeld wordt er duidelijk gemaakt dat er nood is aan een systeem om structuur in het productieproces te hebben, aangezien op het moment alles via spreadsheets bijgehouden wordt.

Doordat dit systeem deze data op een gestructureerde manier bijhoud kan dit dat gevisualiseerd worden, om zo de varkenshouder een inzicht te geven in het productieproces van zijn bedrijf.

**Rollen van het systeem**

Het systeem moet zo eenvoudig te gebruiken dat de gebruiker een minimum aan tijd nodig heeft om acties te voltooien. Het systeem is immers een oplossing om het veehouder proces te vergemakkelijken. Een boer die werknemers in dienst heeft, moet in staat zijn deze werknemers gespecificeerde taken te kunnen laten volbrengen en dit te kunnen opvolgen in het systeem. Er zijn mogelijk managers, administratief bedienden en andere type werknemers. Een boerderij is immers ook een bedrijf!

Niet elke werknemer van de boerderij heeft dezelfde verantwoordelijkheden en moet dus begrensd zijn in de acties die hij/zij kan nemen binnen het systeem. Een werknemer die de wegingen doet mag bijvoorbeeld niet in staat zijn inkoopfacturen te bekijken.

Om de eenvoud en flexibiliteit te bewaren bij het toevoegen van werknemers van bepaalde afdelingen, moet de klant in staat zijn elke werknemer rechten te kunnen toekennen, aanpassen en verwijderen. Omwille van de vele mogelijkheden zal het systeem moeten opgebouwd worden uit claims die rechten geven aan werknemers tot het systeemgebruik in plaats van statisch gedefinieerde type gebruikers (zoals administrator, gebruiker, manager, …).

**Agenda**

Boeren kunnen soms ook een slecht overzicht hebben van hun werknemers als ze een groot bedrijf hebben. Ze steken te veel tijd in het zoeken van hun personeel en het uitdelen van taken wat elke dag opnieuw gebeurt. Ook als er wordt gewerkt met dagelijkse of wekelijkse werkroosters neemt dit ook tijd in beslag voor een boer (of administratieve medewerker) om zo’n rooster te maken en uit te delen evenals de tijd die het personeel gebruikt om al zijn/haar werkuren in dit rooster in te vullen en af te geven. Deze tijd kan beter worden benut! Ons systeem maakt gebruik van een agenda waar een gebruiker met de rechten “agendabeheerder” taken kan aanmaken en verwijderen/bevestigen - in ons systeem is een bevestiging geassocieerd met het verwijderen van een taak - in functie van groepen (dieren). Elke andere gebruiker met de rechten “werknemer” kan taken voltooien. Een taak kan aangemaakt worden op twee manieren:

* door een agendabeheerder die een gewone taak aanmaakt. Dit betekent dat de taak direct actief is (zichtbaar is in de kalender).
* door het systeem dat automatisch taken aanmaakt op basis van acties van gebruikers met een beheerdersrol (vb: als een manager een nieuwe groep dieren aanmaakt dan maakt het systeem een taak “nieuwe weging voor groep X”).

# Use cases

Hieronder vindt u een kleine legende van alle rollen in ons systeem.

|  |
| --- |
| Gebruiker |
| Werknemer |
| Administratieve medewerker |
| Manager |
| Agendabeheerder |
| Voedings-beheerder |
| Weger |
| Baas |
| Tijd |

## Inloggen

**Actoren:** Gebruiker

**Beschrijving:**

een gebruiker geeft zijn logingegevens aan het systeem en het systeem controleert of deze gegevens correct zijn. Als de gegevens correct zijn wordt de gebruiker doorverwezen naar zijn systeem.

## Grafieken bekijken

**Actoren:** Baas

**Beschrijving:**

De baas navigeert naar het dashboard om visualisaties te krijgen over de gegevens. Het systeem gaat aan de hand van allerlei berekeningen verschillende grafieken tonen. De baas kan ook filteren welke grafieken voor hem het belangrijkste zijn. Het systeem zal dan alleen de gefilterde grafieken tonen. De belangrijkste parameters die gevisualiseerd worden zijn degenen die in de probleemstelling aangehaald worden.

## Gebruikers beheren

**Actoren:** Baas

**Beschrijving:**

De baas navigeert naar de pagina waar hij gebruikers kan beheren. Op deze pagina kan hij de volgende operaties uitvoeren:

1. gebruikers bekijken
2. gebruikers toevoegen
3. gebruikers wijzigen
4. gebruikers verwijderen

**let op:** als een baas operaties 1 tot 3 uitvoert, worden ook rechten toegekend aan een gebruiker. Meerdere rechten kunnen worden toegekend aan een gebruiker dus een gebruiker kan zo bijvoorbeeld administratieve medewerker en manager zijn.

## Beheren van een groep

**Actoren:** Manager

**Beschrijving:**

De manager navigeert naar groep management en kan zijn groepen beheren en bekijken. De manager heeft dan de mogelijkheid om volgende operaties uit te voeren:

1. groepen lezen
2. groepen toevoegen
3. groepen wijzigen
4. groepen verwijderen

## Een groep importeren

**Actoren:** Administratieve medewerker

**Beschrijving:**

De administratieve medewerker voorziet een .csv bestand dat het systeem kan importeren in een juiste formaat en importeert deze. Het systeem gaat alle data uit het bestand halen en slaat dit op in een databank.

## Taken beheren

**Actoren:** Agendabeheerder

**Beschrijving:**

De agendabeheerder navigeert naar taakmanagement en kan taken beheren en lezen. Hij kan de volgende operaties uitvoeren:

1. een taak lezen
2. een taak aanmaken
3. een taak wijzigen
4. een taak verwijderen

## Taken aanmaken

**Actoren:** Het Systeem

**Beschrijving:**

Wanneer een nieuwe groep wordt aangemaakt, worden automatisch taken aangemaakt door het systeem die bij iedere groep van toepassing zijn.

## Voer beheren

**Actoren:** Voedings-beheerder

**Beschrijving:**

De voedings-beheerder navigeert naar voer management. Hij kan de volgende operaties uitvoeren:

1. voeding bestellingen bekijken
2. voeding bestellingen toevoegen
3. voeding bestellingen wijzigen
4. voeding bestellingen verwijderen

## Een weging doen

**Actoren:** Weger

**Beschrijving:**

De weger navigeert naar groep management. Het systeem toont een lijst van groepen. De weger kan op elke groep een weging updaten.

## Groepen bekijken

**Actoren:** Werknemer

**Beschrijving:**

De werknemer gaat naar het overzicht van groepen. Het systeem toont een overzicht van alle groepen in het systeem. De werknemer kan deze bekijken, maar geen operaties op uitvoeren.

## Taken voltooien

**Actoren:** Werknemer

**Beschrijving:**

De werknemer gaat naar de agenda. Het systeem toont een agenda. De werknemer selecteer een bestaande taak. Het systeem toont een een overzicht van de taak in detail. De werknemer kan deze taak voltooien.

# URPS

## Usability:

Het systeem moet een een visuele indicatie geven wanneer er een fout optreed.

De gebruiker moet 90% van de functionaliteit kunnen uitvoeren zonder documentatie te moeten lezen.

## Reliability:

Het systeem moet foute invoer van data kunnen verwerken, zodat het systeem niet crashed.

Het systeem gebruikt enkel onafgeronde waardes om berekeningen te doen. Enkel om waarden weer te geven mogen deze afgerond worden tot 3 decimalen

## Performance:

Bij ieder interactie met het systeem moet er een reactie zijn binnen een seconde.

## Supportability:

Het systeem moet zonder bestaande features aan te passen toch uitbreidbaar zijn, zodat dit eventueel kan evolueren naar een volledig administratief veeteelt platform.

Wanneer het systeem wordt aangepast moeten alle bestaande testen steeds blijven slagen eer het aangepaste systeem kan gereleased worden.

# Technologie & effort

## Technologie

Het systeem is een REST-webapplicatie die gehost wordt in azure waarbij we makkelijk CI/CD pipelines kunnen integreren samen met de testen van de code. De database taal zal bestaan uit PostgreSQL. De backend zal geschreven worden in ‘.net core’ waar we een ORM gebruiken om de database aan te spreken. De front-end wordt geschreven in React/typescript.

We kiezen voor een webapplicatie omdat veel van de team members al wat ervaring hebben met deze technologie en het is ook bruikbaar in iedere browser dus ook op mobiele toestellen.

We gebruiken SQL omdat de data perfect in een relationele manier kan gestructureerd worden.

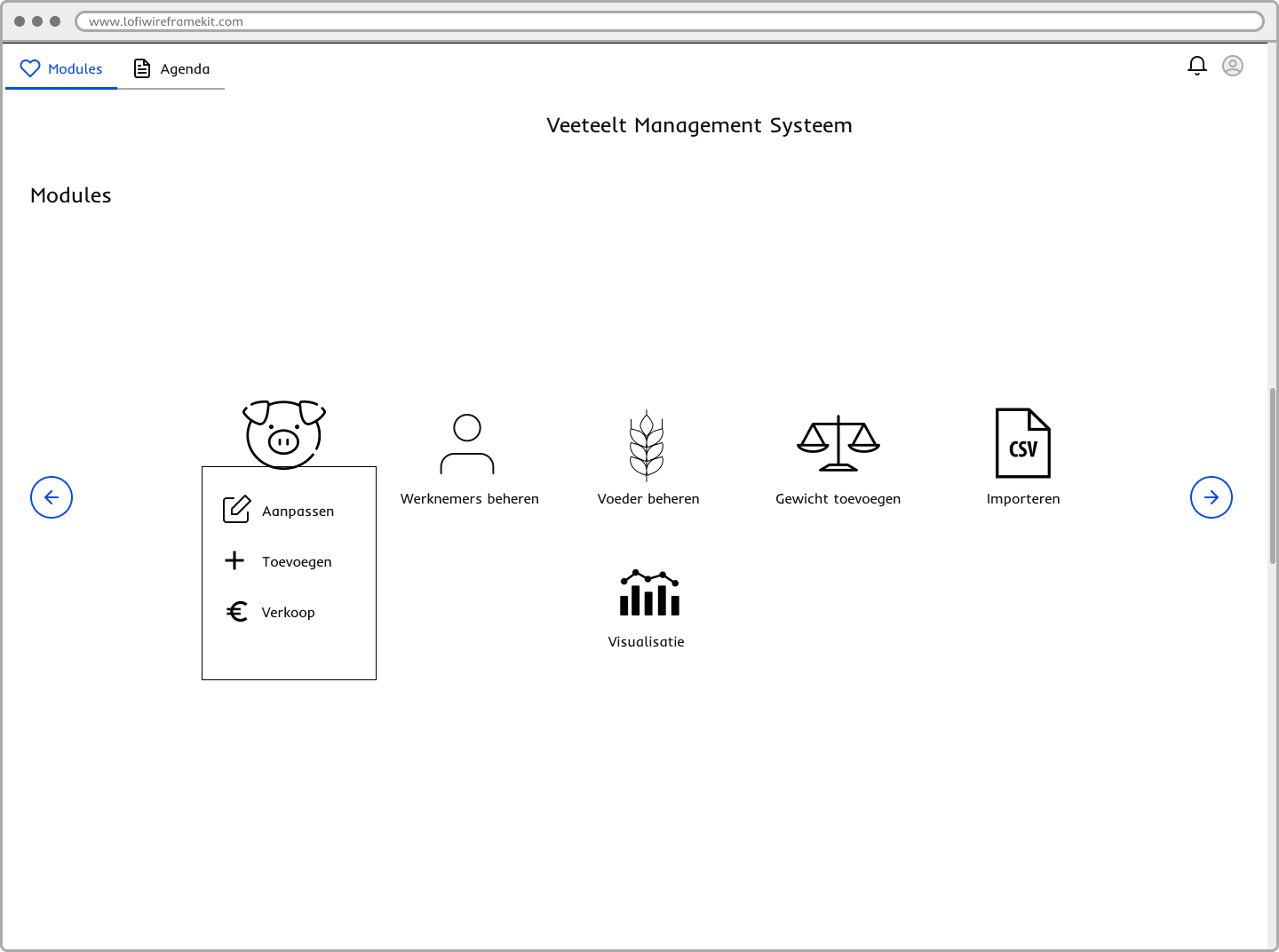
We gebruiken ‘.net core’ omdat ook hier wat team members al ervaring mee hebben en het heeft een goede integratie met ORM’s.

## Effort

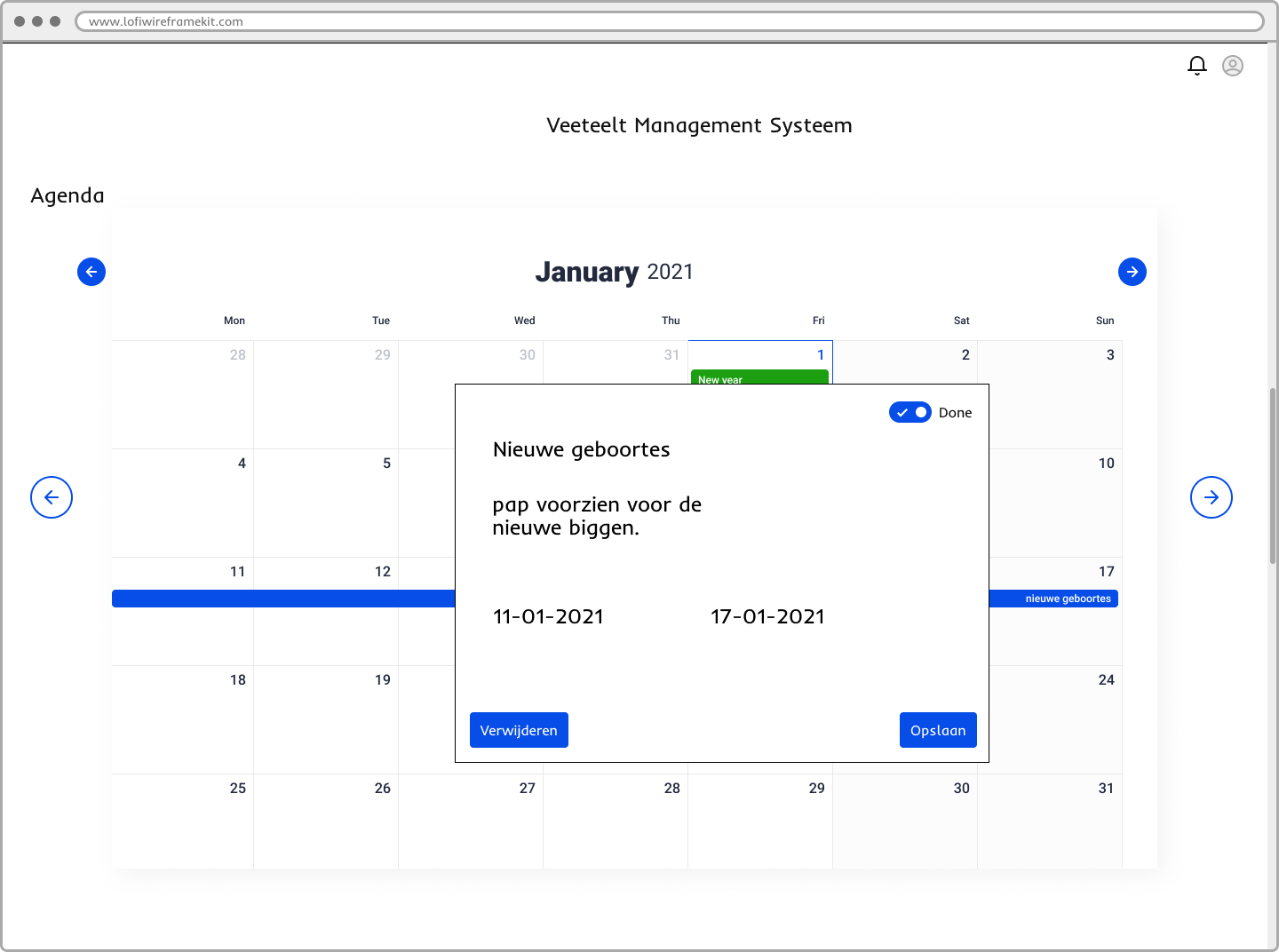
|  |  |
| --- | --- |
| **Omschrijving** | **Inschatting in dagen** |
| Analyse | **3** |
| inloggen/uitloggen | **1** |
| Grafieken bekijken | **10** |
| Gebruikers beheren (+rechten) | **6** |
| Beheren van een groep | **4** |
| Een groep importeren | **5** |
| Taken beheren | **4** |
| Taken aanmaken | **3** |
| Taken voltooien | **2** |
| Een weging doen | **2** |
| Groepen bekijken | **1** |
| Voer beheren | **6** |
| **Totaal:** | **47** |

# 

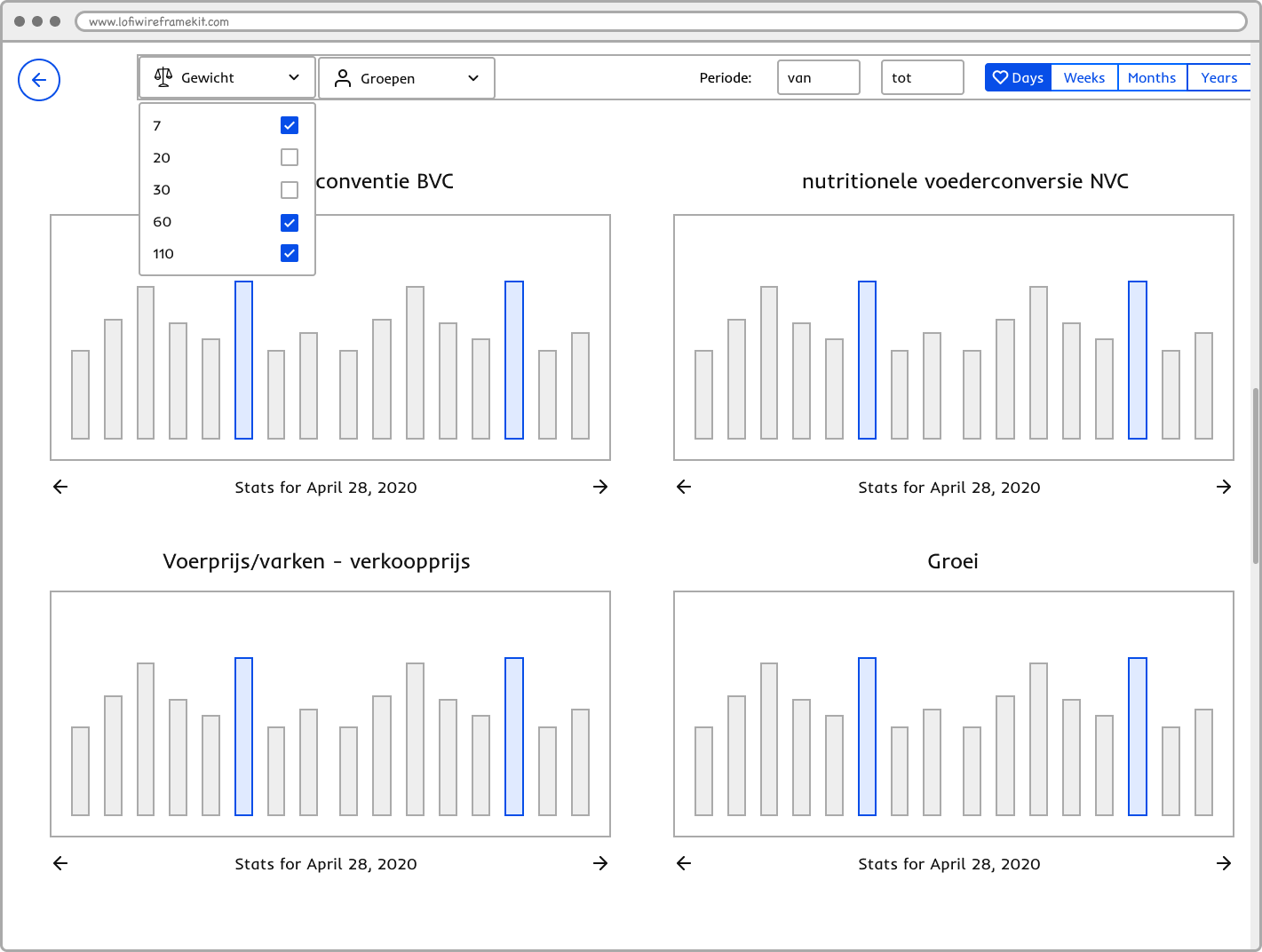
# Mockups

Startscherm met alle functies waartoe de gebruiker rechten heeft.

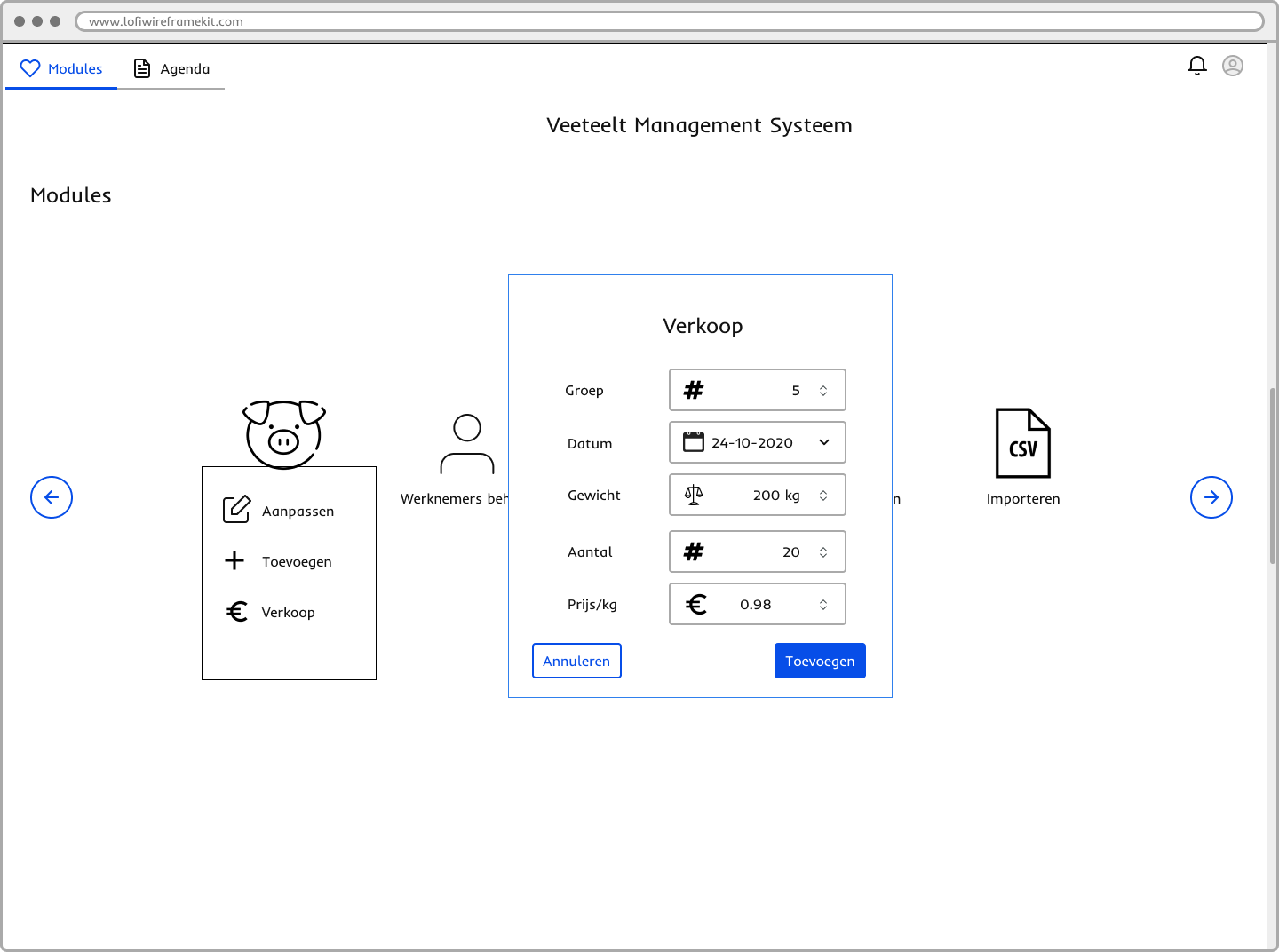
Kalender



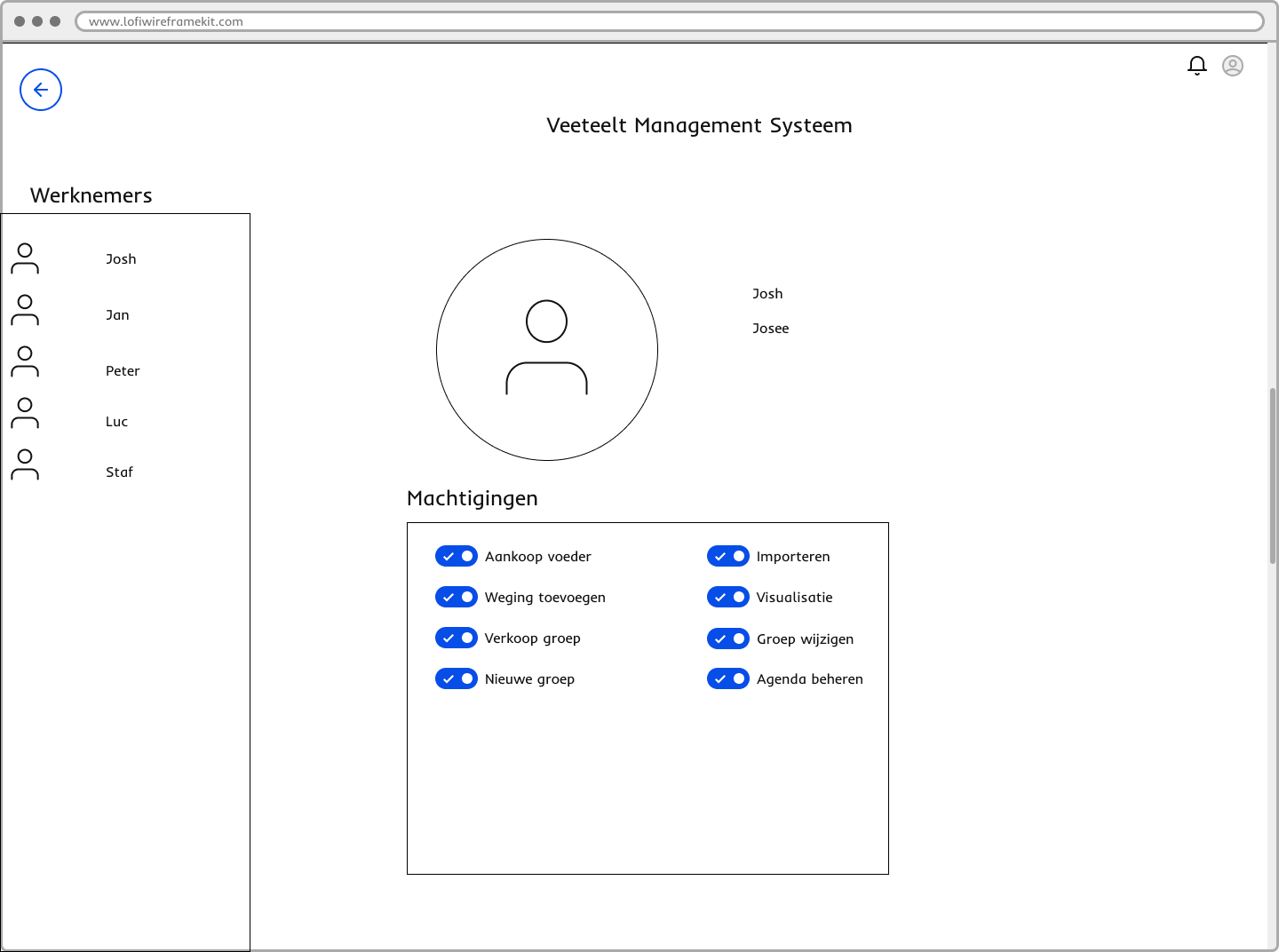
Grafieken



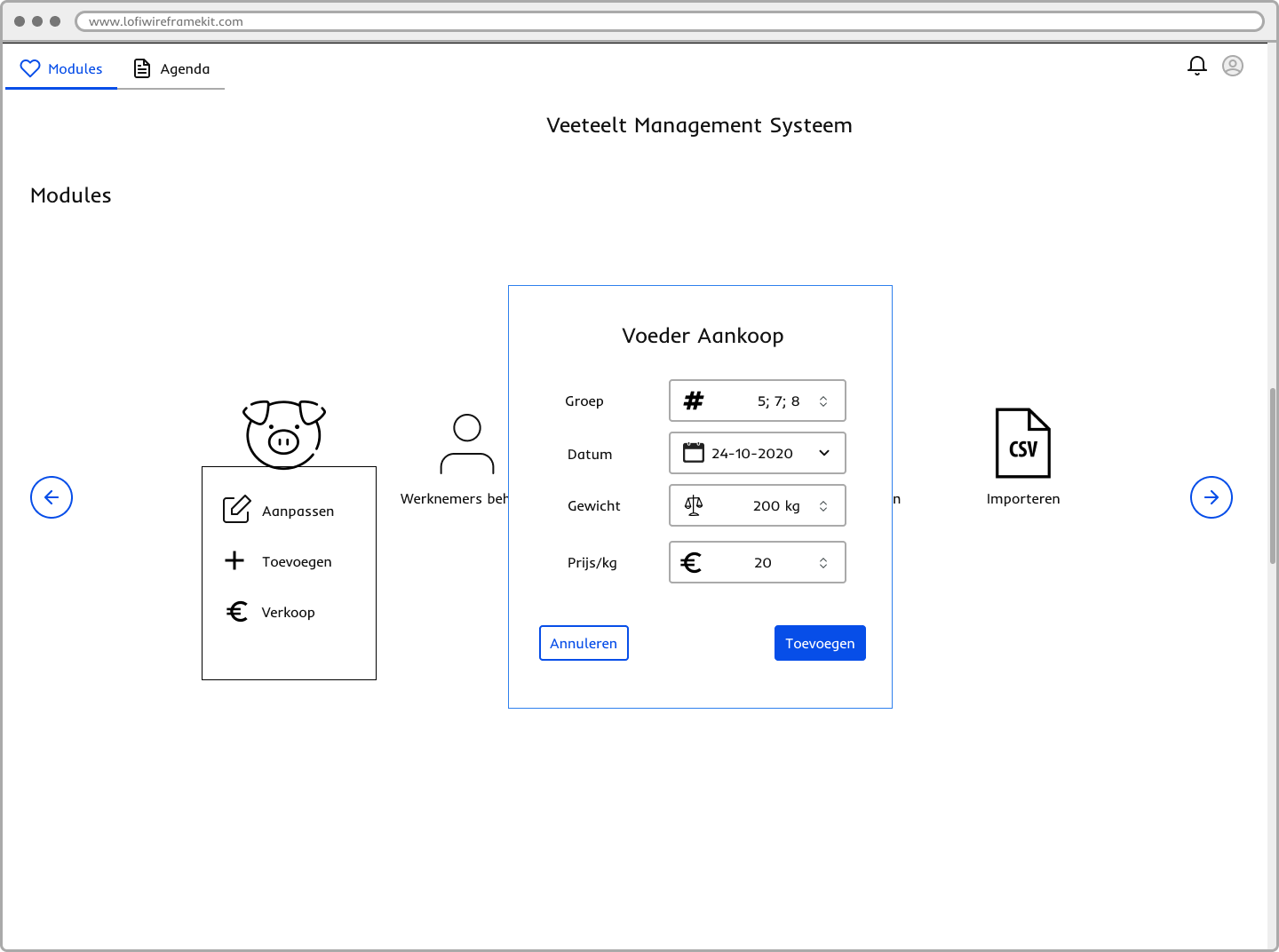
Verkoop



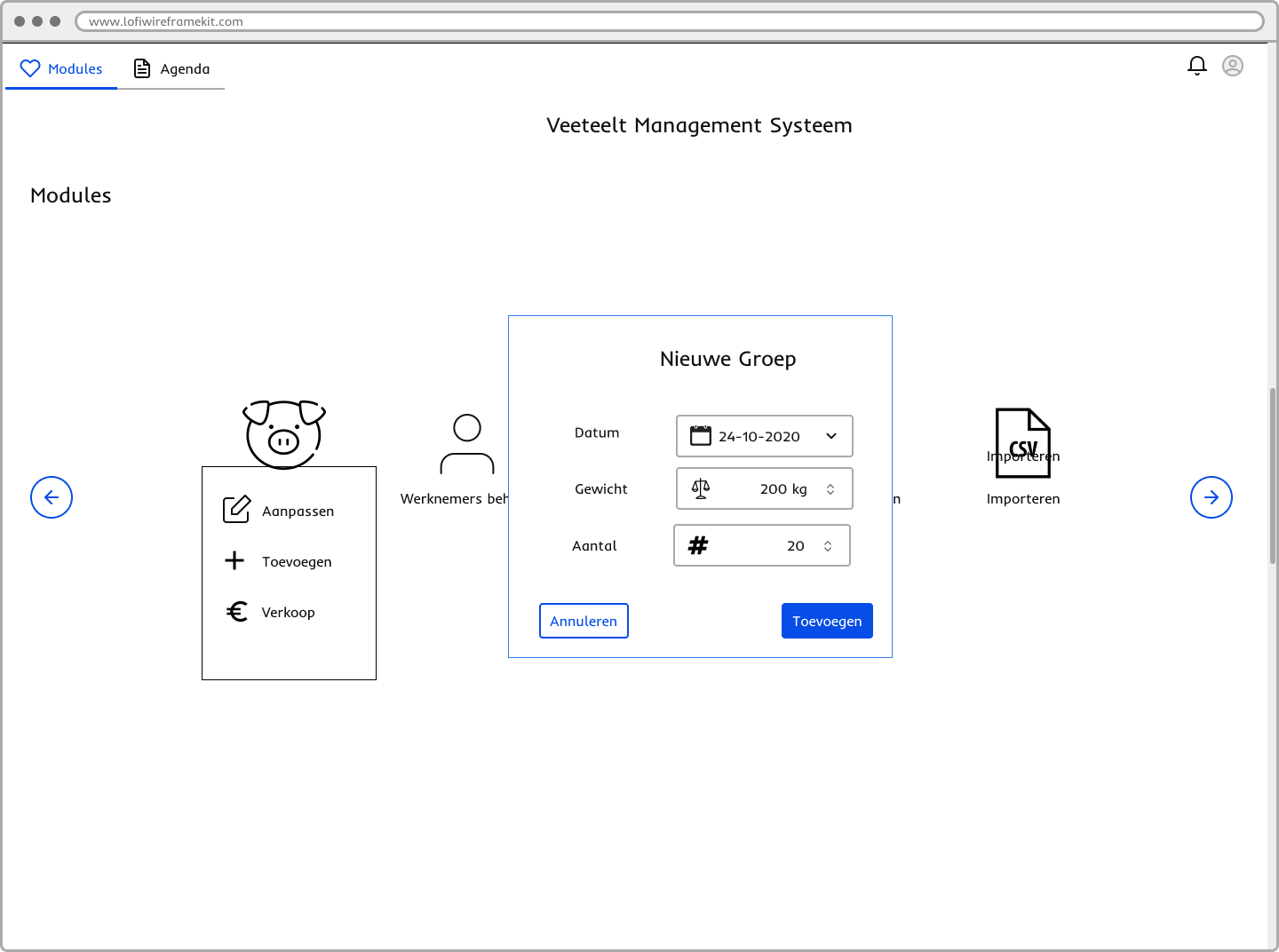
Gebruikers beheren



Voeder aankoop



Nieuwe groep



Groep wijzigen

