Domotica: Automatische dierenvoeder

Project: Automatische dierenvoeder

Klas & TeamNr: HBO-ICT 1G - Team 3

Teamleden:

- Dirk Grent (dirk.grent@student.nhlstenden.com)
- Rick Andreae (rick.andreae@student.nhlstenden.com)
- Albert Berends (albert.berends@student.nhlstenden.com)
- Damiaen Toussaint (damiaen.toussaint@student.nhlstenden.com) (Coordinator C opdracht)

Datum laatste wijziging: 27-06-2018

Datum	Beschrijving van wijziging	Versie Document
27-06-2018	Afronding versie 1.0 document	1.0

Inhoudsopgave

1. Inleiding

Dit document betreft de technische documentatie van het domotica project. Dit document is eigendom van team 3 van de klas HBO-ICT 1G. Dit team bevat Dirk, Rick, Albert en Damiaen als teamleden. Het domotica project is een opdracht voor periode 4 van de opleiding HBO-ICT.

In dit document zijn de technische aspecten van het project te vinden, zoals schema's/diagrammen van de applicatie en de Arduino. Het document is opgesplitst in 2 delen, de voorbereiding met user stories en de techniek achter het project.

2. Requirements en User stories

Om de requirements van dit project duidelijk in beeld te krijgen wordt er eerst een lijst met user stories opgezet. Deze user stories zijn gesorteerd met behulp van de MoSCoW methode. Vanuit deze user stories worden de requirements van het project opgesteld. Elk onderdeel is op prioriteit gesorteerd (nummering).

2.1. User stories

De hieronder te vinden user stories zijn gesorteerd met behulp van de **MoSCoW**-methode. Hiermee wordt er een duidelijk beeld gemaakt over welke functionaliteit er aanwezig moet zijn, wat er wel en niet mogelijk is om te implementeren en wat de prioriteit van de functionaliteit is.

Must have:

1. Als baasje wil ik dat mijn huisdier op afstand gevoerd kan worden, zodat mijn huisdier niet thuis uitgehongerd op mij hoeft te wachten op eten.

2. Als baasje wil ik dat mijn huisdier op vaste tijden gevoerd kan worden, zodat mijn huisdier over de gehele dag goed gevoed blijft en geen belangrijke maaltijden mist.

3. Als baasje wil ik de hoeveelheid van de portie kunnen instellen, zodat mijn huisdier niet overvoed of ondervoed wordt.

4. Als baasje wil ik weten hoe vol de bak met voer is, zodat ik op tijd de voederbak kan bijvullen wanneer de bak bijna leeg is.

5. Als gebruiker wil gemakkelijk instellingen op de app kunnen veranderen, zodat ik gemakkelijk het dieet van mijn huisdier kan aanpassen mocht hij/zij bijvoorbeeld ziek worden.

6. Als baasje wil ik dat het juiste portie aan he toebehorende huisdier wordt gegeven, zodat de huisdieren niet de verkeerde portie krijgen.

Should have:

7. Als eigenaar wil ik een simpele voederbak, zodat ik niet plotseling een Ikea kast in elkaar moet zetten.

8. Als baasje, wil ik dat de voeder zeker maakt dat al het voer van het portie geserveerd word, zodat de porties niet teveel verschillen.

9. Als baasje wil ik een grote opening, zodat het gemakkelijk zal zijn om de bak bij te vullen.

Could have

10. Als baasje, wil dat ik dat de voeder licht gewicht is, zodat ik het gemakkelijk kan verplaatsen.

11. Als baasje, wil ik dat de voeder robuust gebouwd is, zodat het tegen een stootje van de hond kan.

12. Als baasje, wil ik dat de voeder van goedkope materialen is gemaakt, zodat de prijs betaalbaar blijft.

Wont have:

2.2. Requirements

Om een duidelijk beeld te scheppen over de requirements die bij de user stories horen is hieronder een tabel te vinden (Tabel 3-1). Het bijbehorende nummer is terug te koppelen aan de user stories in "3.1 user stories". "Prio" staat gelijk aan de prioriteit, "US-Nr" is de bijbehorende userstory.

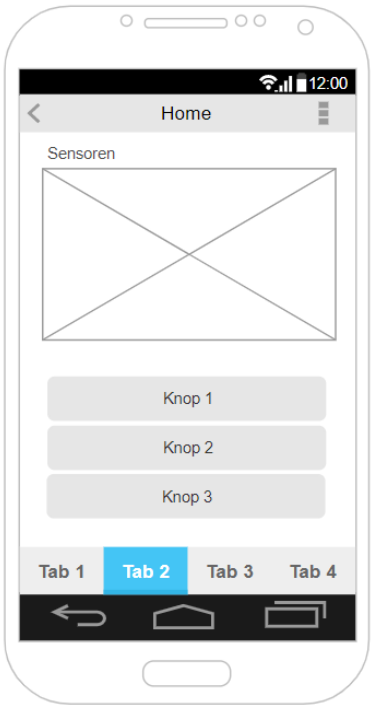
Prio	US-Nr	Bijbehorende requirement
1	1	De arduino moet een component/motortje hebben om het klepje van de voederbak te openen.
2	3	Het motortje moet ingesteld kunnen worden om langer open te staan voor grote porties en korter voor kleiner porties.
3	3	Hoe lang het motortje moet draaien, moet gekoppeld worden aan de grootte van de portie ingesteld in de app.
4	4	De arduino moet een sensor (ultrasonor/gewichts-sensor) hebben waarbij gemeten kan worden hoeveel voer er nog in de bak aanwezig is.
5	5	De voederbak moet makkelijk te gebruiken zijn.
6	2	De app moet een systeem hebben waarin de voedertijden ingesteld kunnen worden.
7	6	De app heeft een duidelijk UI.
8	7	De arduino moet een identificatiesysteem hebben om de juiste portie aan het juiste huisdier te geven.

Tabel 2-1 User stories met requirements

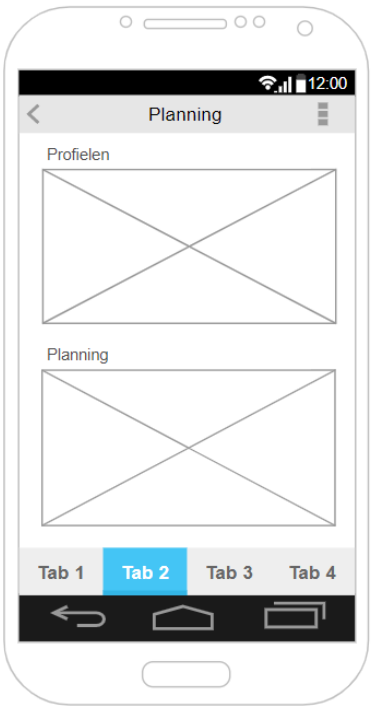
3. Beschrijving van de applicatie

In dit onderdeel van het document zal de gehele applicatie beschreven worden, het gaat hier dus om de Xamarin App. De technische beschrijving van de Arduino is te vinden in hoofdstuk 4 van dit document. Het beschrijven van de applicatie met behulp van wireframes zal in 3.1 zijn. De algemene flow van de applicatie is te vinden in onderdeel 3.2 van dit hoofdstuk. Het databaseontwerp van de onderliggende SQL database is te vinden in 3.3

3.1. Wireframes



Figuur 3-1 Homepagina van de app



Figuur 3-2 Planning en voeder pagina

Figuur 3-3 Instellingen pagina

In de bovenstaande afbeelding zijn de drie standaardschermen van de applicatie te vinden. Deze pagina's zijn als volgt: De homepagina waar een overzicht van de statistieken is te vinden, de planning waar profielen van dieren en voedertijden te vinden zijn en tot slot een tabje voor instellingen.

Home pagina (Figuur 3-1)

Op de homepagina van de applicatie is in één duidelijk overzicht alle informatie over de voeder te vinden. In het bovenste gedeelte de pagina staat huidige informatie van de voeder, deze informatie word automatisch geüpdatet met een timer. Onder deze informatie zitten knoppen om snel handmatig voer te kunnen geven of een sensor update te forceren zonder te wachten op de interval.

Planning pagina (Figuur 3-2)

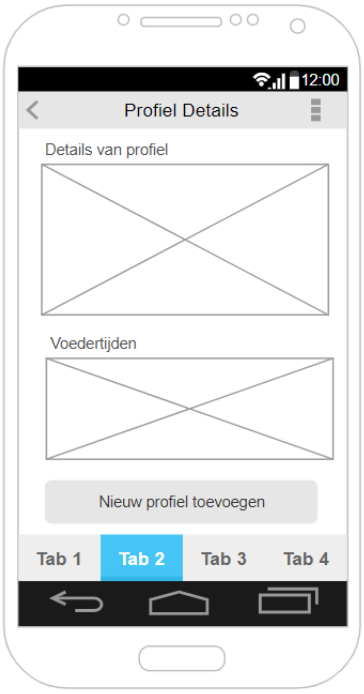
Op de overzichtspagina van de profielen en planning is alle informatie over toegevoegde profielen en huidige voedertijden te vinden. Op deze pagina kun je ook een nieuw profiel toevoegen.

Profielen zijn een manier om snel en makkelijk bij de gegevens van je huisdieren te komen. Door op te slaan wat een standaardportie voer is kun je gemakkelijk de juiste hoeveelheid voer geven met een druk op de knop. Per profiel kun je als gebruiker weer een planning/tijdschema aanmaken.

Met Tijdschema's of de planning kun je snel zien wat de komende voedertijden zijn. Op deze pagina krijg je alle voedertijden te zien, als je verder navigeert door op een profiel te drukken krijg je per profiel de planning te zien.

Instellingen pagina (Figuur 3-3)

De instellingen pagina is de pagina waar een gebruiker gemakkelijk de standaard configuratie van de Arduino kan opslaan. Dit is overigens ook erg handig voor het debuggen van de applicatie en Arduino. Hier kan een gebruiker dus het IP adres van de Arduino opslaan, maar ook of de app op de achtergrond taken mag uitvoeren of dat de Arduino data moet loggen op een SD kaart (RFID). Deze gegevens worden opgeslagen in een lokale database.



Figuur 3-4 Profiel details pagina

Figuur 3-5 Profiel toevoegen/bewerken

Figuur 3-6 Tijd toevoegen/bewerken

Profiel details pagina (Figuur 3-3)

Op de profiel details pagina kan de gebruiker alle gegevens van een profiel terugvinden. Deze gegevens zijn verder aan te passen in deze pagina door via een knop naar de bewerken pagina te gaan. Bij profielgegevens moet men denken aan Naam, diersoort, standaard portie enz.

Bij voedertijden zie je alle huidige voedertijden van het huidige profiel. Hier zijn alleen de huidige actieve voedertijden te vinden. Bij het uitvoeren van de voedertijd zal deze op uitgevoerd komen te staan, waarna deze niet meer terug te vinden is. Behalve als de gebruiker er voor heeft gekozen om de voedertijd als dagelijks in te stellen, waarna deze zich dagelijks herhaalt.

Profiel toevoegen / bewerken pagina (Figuur 3-3)

Bij de profiel toevoegen/bewerken pagina kan een gebruiker een nieuw profiel toevoegen of een bestaande beweren. De pagina is in principe hetzelfde voor beide functies, alleen zijn de velden bij het bewerken al ingevuld. Als het de bewerken variant van deze pagina is heeft de gebruiker ook nog de mogelijkheid om het profiel te verwijderen. De flow van deze functionaliteit is terug te vinden in de flowchart.

Bij het aanmaken van een profiel geeft de gebruiker een standaardportie grootte op, deze zal als standaard worden gehanteerd. Deze standaard kan omzeilt worden door in de planning een andere grootte door te geven.

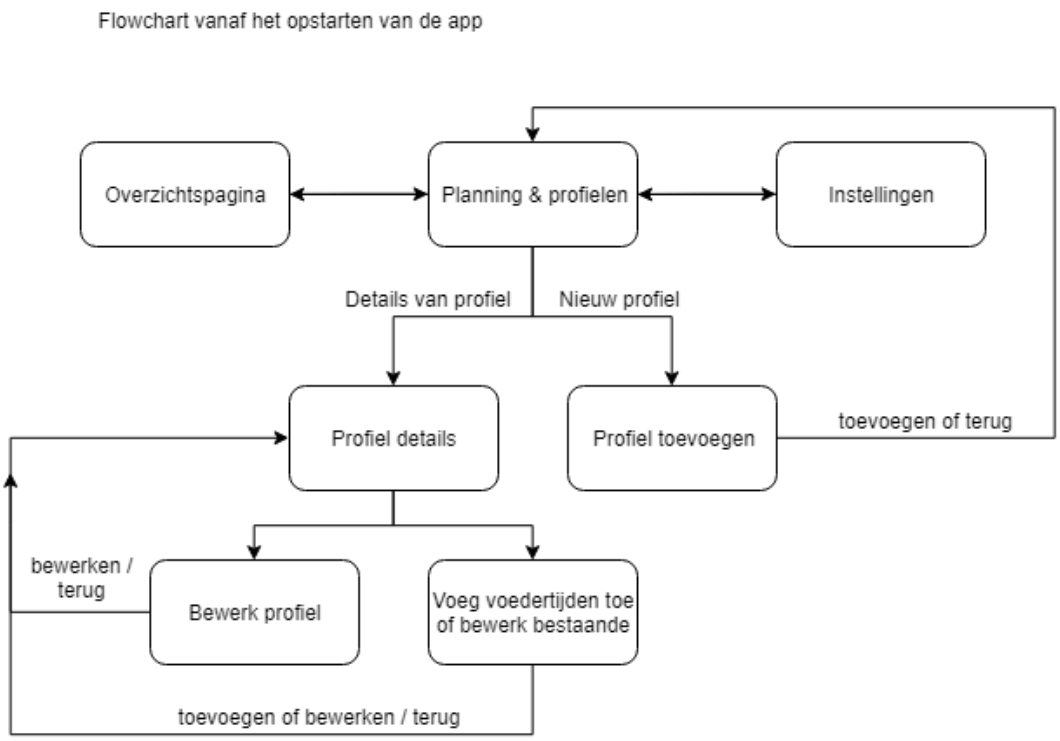
Tijd toevoegen / bewerken pagina (Figuur 3-3)

Bij de tijd toevoegen/bewerken pagina kan een gebruiker een nieuw profiel toevoegen of een bestaande bewerken. Deze pagina is in principe hetzelfde voor beide functies met als enig verschil dat de velden van de bewerken pagina al zijn ingevuld.

Bij het invullen van deze pagina kan de gebruiker optioneel een portiegrootte opgeven, om af te wijken van de standaard portie opgegeven bij het profiel. Het kiezen van de tijd zal gaan via een datepicker. Als het de bewerken variant van deze pagina is heeft de gebruiker ook nog de mogelijkheid om de tijd te verwijderen. De flow van deze functionaliteit is terug te vinden in de flowchart.

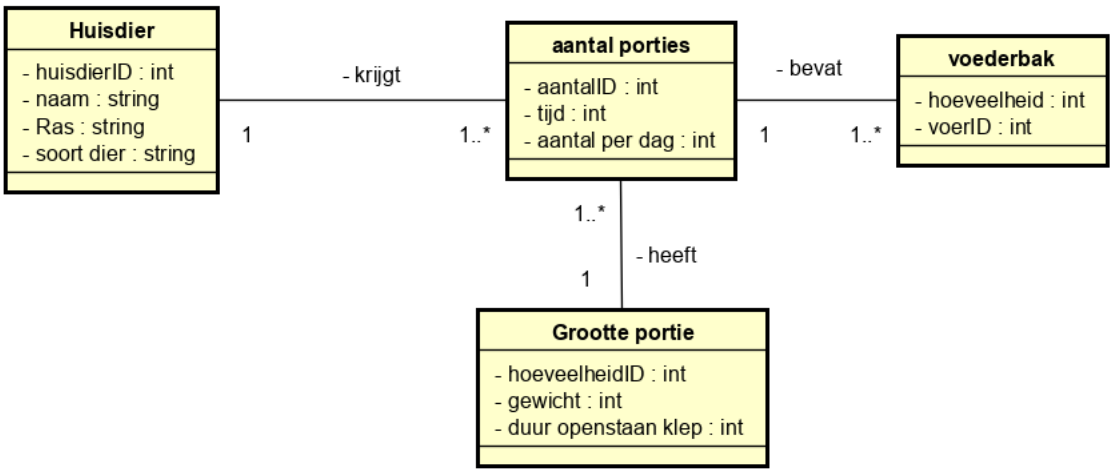
3.2. Flowchart

Om de algemene flow van de app goed op kaart te zetten is er een flowchart gemaakt die hieronder te vinden is. Voor het design van de app is er met opzet gekozen om de UI zo simpel mogelijk te houden.



Figuur 3-7 Flowchart navigatie app

3.3. Klassendiagram / database ontwerp



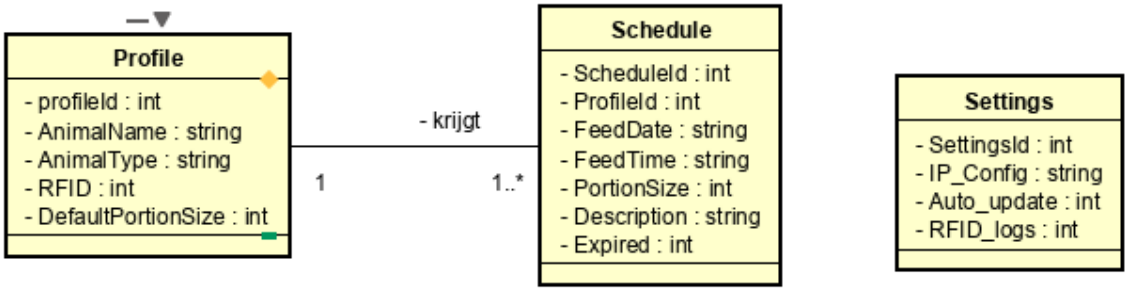
Figuur 3-8 KlassenDiagram App

Het bovenstaande database ontwerp (Figuur 3-8) is de basis van de database die voor dit project ontwikkeld gaat worden. De database is in principe gelijk aan het klassendiagram van de app, omdat deze via sqllite worden gebruikt zijn de benamingen hetzelfde.

Een gebruiker van de app kan verschillende profielen aanmaken, deze profielen zijn bedoelt als “dieren”. In een profiel kan een gebruiker standaard informatie opslaan over een huisdier, zoals naam en standaard portie grootte.

Verder kan een profiel meerdere schedules hebben, in deze planningen/tijdschema’s staan de voedertijden van de dieren. Deze voedertijden zorgen ervoor dat op het juiste moment van de dag de voeder het eten in het bakje laat vallen.

Tot slot is er een losstaande table voor de instellingen, zo kan de gebruiker de standaard informatie van de app opslaan. Dit zijn instellingen zoals het IP adres van de Arduino of dat de app automatisch op de achtergrond taken mag uitvoeren.

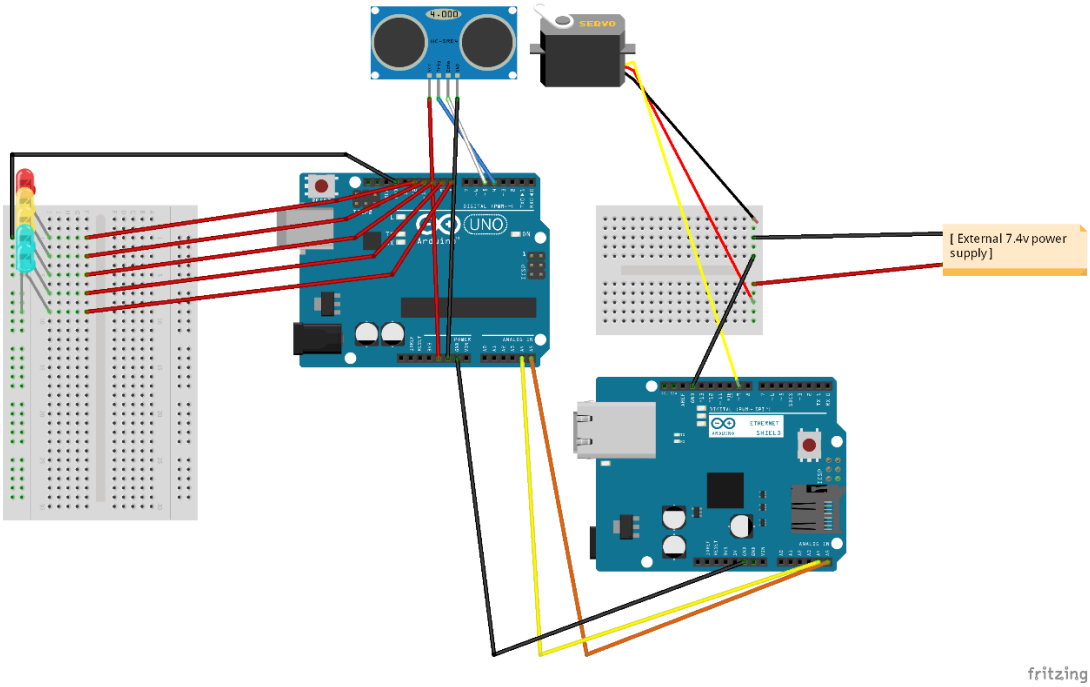


Figuur 3-9 DatabaseOntwerp SQL-Lite

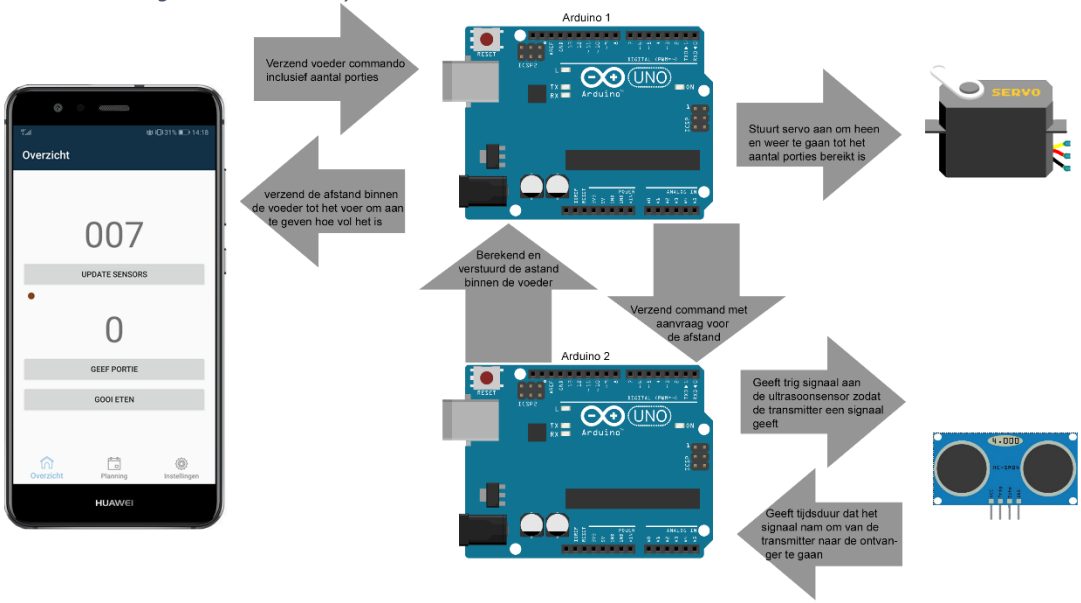
4. Beschrijving van de Arduino

In dit onderdeel van het document zal de gehele applicatie beschreven worden, het gaat hier dus om de voeder zelf. Een gedeelte van de informatie te vinden in dit hoofdstuk is ook terug te vinden in ons projectvoorstel.

4.1. Arduino schema's



Figuur 4 - 1 Fritzing van de arduino layout



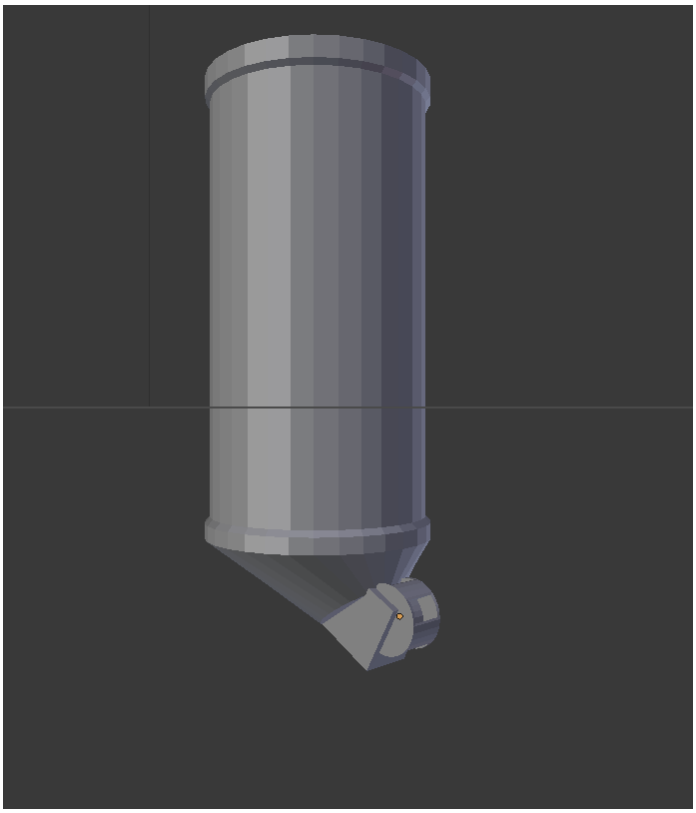
Figuur 4 - 2 Architectuurschema van de communicatie

5. Beschrijving van de voederconstructie

Een van de uitdagingen van dit project is het maken van een robuuste constructie voor de voeder. In het projectvoorstel is de keuze gevallen op het zelf bouwen van een constructie van hout en plastic. In dit hoofdstuk zal het gehele proces in het kort samengevat worden.

5.1. Schema's constructie

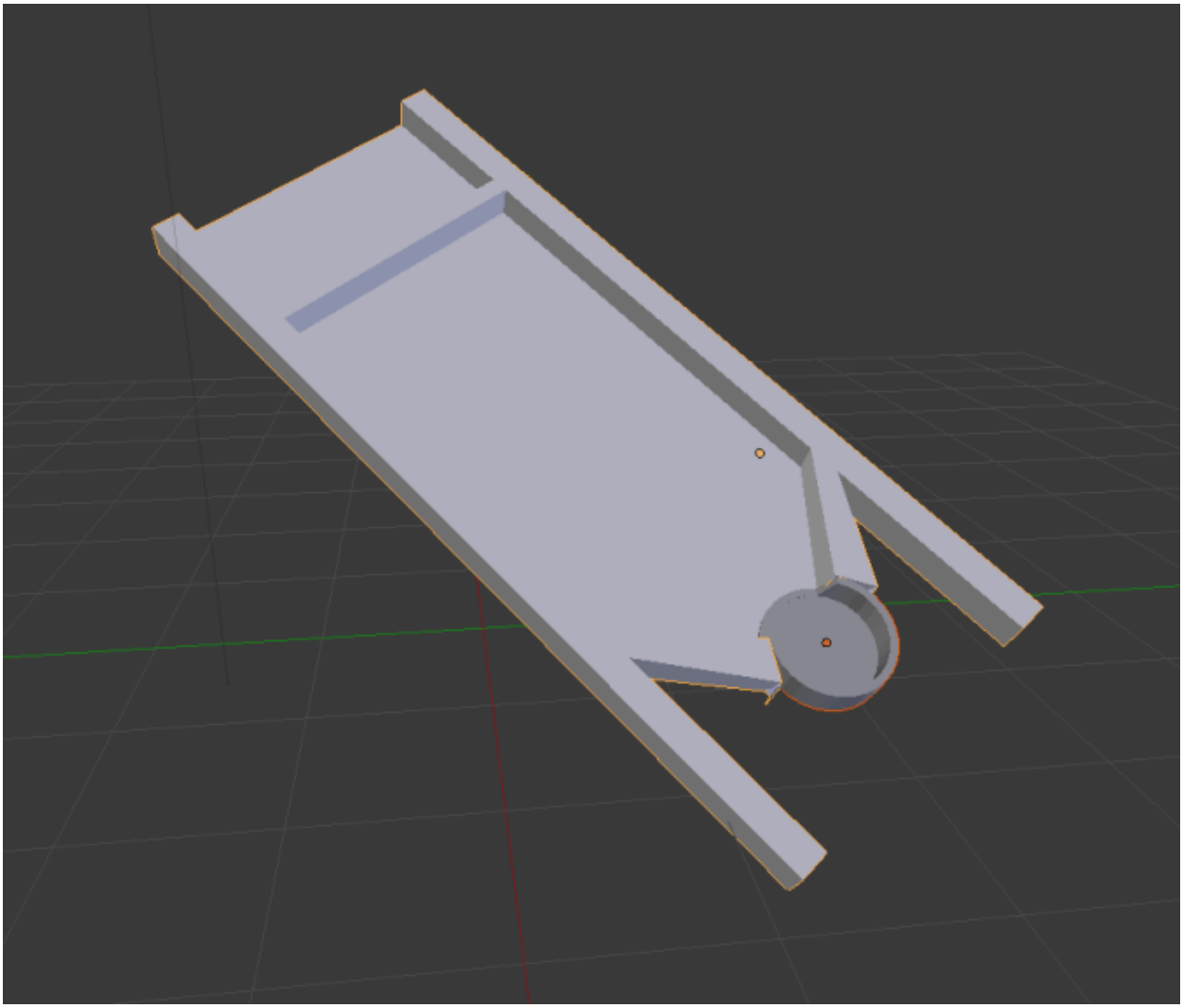
In een van de eerdere versies van het plan was het idee om een plastic koker te gebruiken om de brokjes te houden. Om de brokjes in de voederbak te laten vallen zat beneden een draaiding wat draait. Hieronder is een 3D model te vinden (Figuur 5-1) van het eerste concept van de voeder (gemaakt door Dirk).



Figuur 5 - 1 Concept 1 van de voeder

Later hebben we als team besloten om dit design niet te gebruiken. Het realiseren van de koker zou geen probleem zijn, het probleem lag bij de constructie die het voer laat vallen. Ook waren we bang dat een Servomotor niet de kracht zou hebben om de brokjes te verplaatsen.

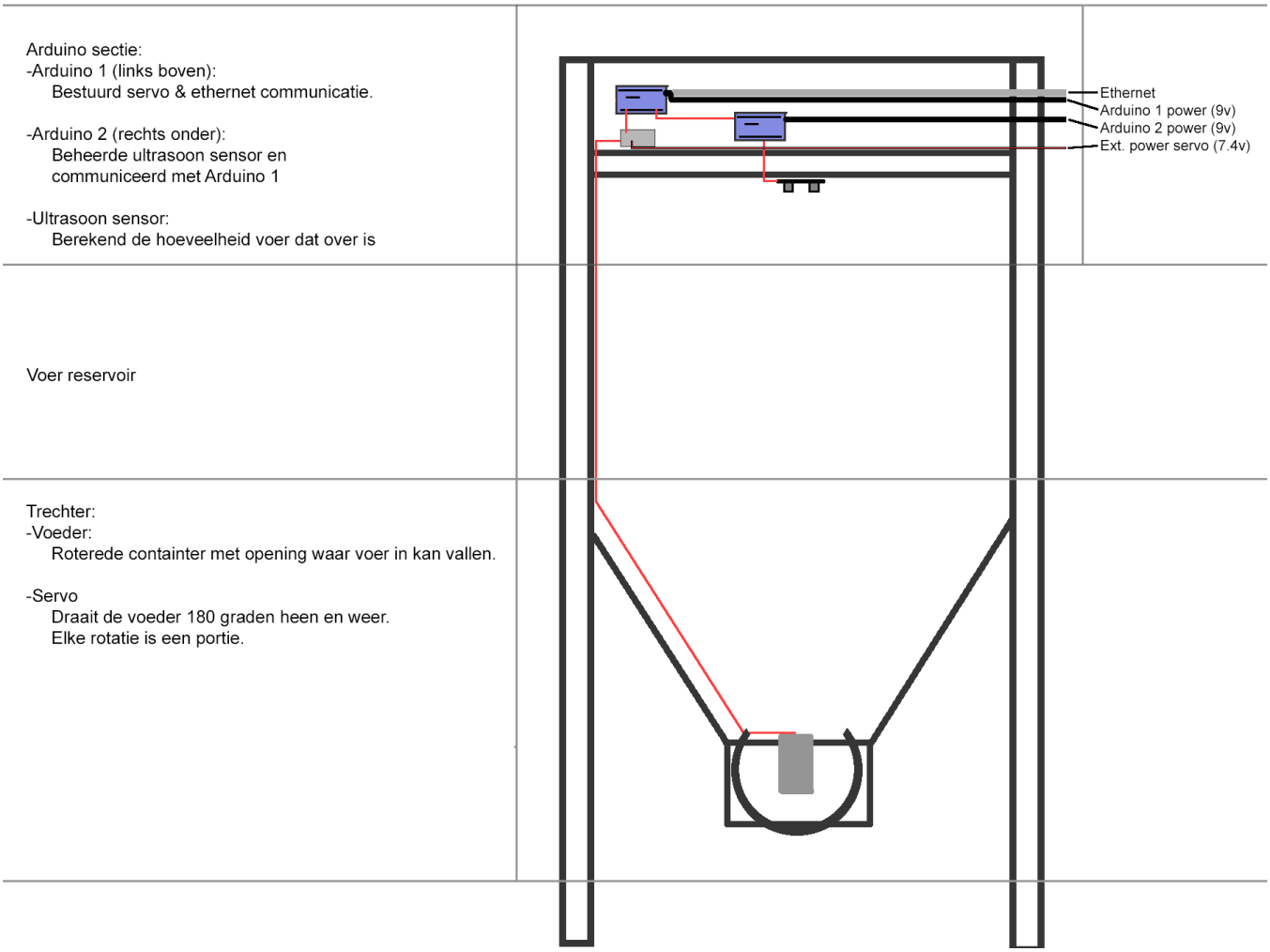
Na het niet gebruiken van het vorige design besloten we als team om gewoon naar de bouwmarkt te gaan en daar te gaan kijken naar spullen die wij konden gebruiken. Na een bezoek aan de bouwmarkt zijn er wat plankjes, balkjes en plastic buizen gekocht waarmee we zijn gaan testen. Het belangrijkste is een systeem wat niet vast loopt als er brokjes vast gaan zitten. Uiteindelijk hebben we de “gewoon doen en daarna denken” aanpak genomen, dat goed uitpakte voor ons. Door te prutsen en te testen met motoren zijn we tot een robuuste constructie gekomen, te zien in de onderstaande afbeelding van de 3D schets.



Figuur 5 - 2 3D schets van de uiteindelijke constructie

Deze contstuctie gaf ons een simpelere methode om het voer te door te laten lopen naar de uitgang. Verder gaf de constructie ook de mogelijkheid om een compartiment aan de top te plaatsen waar de electronica en kabel geplaats konden worden. Dit maakte het makkelijker om de ultrasoon sensor te plaatsen en de container (grootste compartiment) te vullen voer voer.

De contstructie en electronica is volgens de onderstaande afbeelding geplaats, op te delen in 3 delen/compartimenten.



Figuur 5 - 3 Design van voeder en uitleg van componenten