Domotica: Automatische dierenvoeder



**Project:** Automatische dierenvoeder

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

**Teamleden:**

* Dirk Grent ([dirk.grent@student.nhlstenden.com](mailto:dirk.grent@student.nhlstenden.com))
* Rick Andreae ([rick.andreae@student.nhlstenden.com](mailto:rick.andreae@student.nhlstenden.com))
* Albert Berends ([albert.berends@student.nhlstenden.com](mailto:albert.berends@student.nhlstenden.com))
* Damiaen Toussaint ([damiaen.toussaint@student.nhlstenden.com](mailto: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 1](#_Toc12390135)

[2. Requirements en User stories 2](#_Toc12390136)

[2.1. User stories 3](#_Toc12390137)

[2.2. Requirements 4](#_Toc12390138)

[3. Beschrijving van de applicatie 4](#_Toc12390140)

[3.1. Wireframes 4](#_Toc12390141)

[3.2. Flowchart 5](#_Toc12390142)

[3.3. Klassendiagram / database ontwerp 6](#_Toc12390143)

# Inleiding

Dit document betreft het technische document 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.

# 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).

## 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 eigenaar wil ik een simpele voederbak, zodat ik niet plotseling een Ikea kast in elkaar moet zetten.
6. 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.

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

**Should have:**

**Could have**

**Wont have:**

## 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 3‑1 User stories met requirements

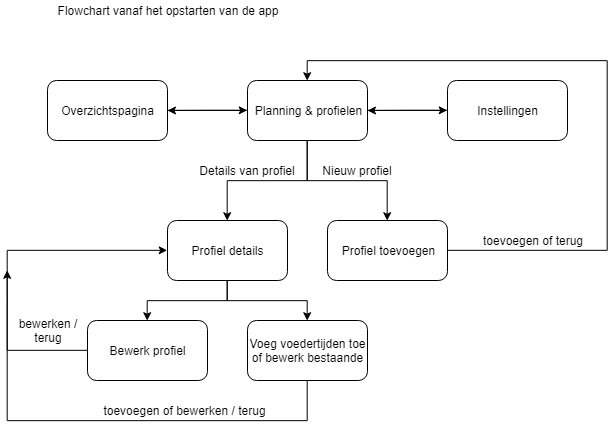
# Beschrijving van de applicatie

Het beschrijven van de applicatie (App) zal gaan via wireframes(3.1). De algemene flow van de applicatie is te vinden in de flowchart (3.2). Het database ontwerp van de sql-lite database is te vinden in (3.3).

## Wireframes

## 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.

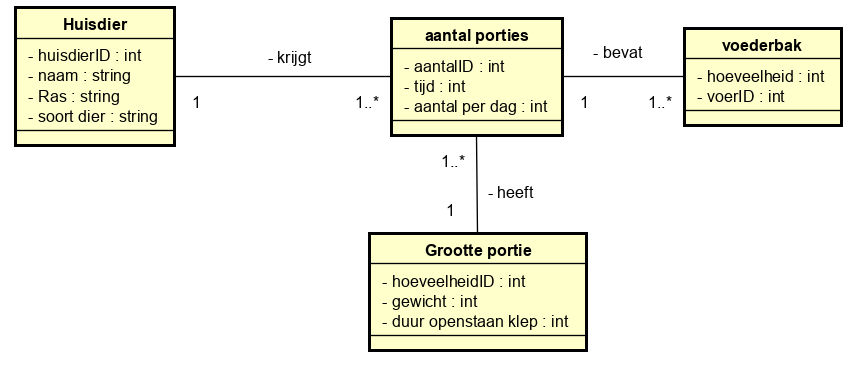


Figuur 4‑9 Flowchart inlogproces website

**Flowchart login proces (Figuur 4-9)**

Figuur 4-9 laat zien hoe een gebruiker inlogt op de webapplicatie. Eerst gaat de gebruiker naar de login pagina, hier worden de gegevens ingevuld in de velden. Dan checkt het systeem of de gegevens overeenkomen met de gegevens in de database. Komen deze overeen dan is de gebruiker ingelogd, komen deze niet overeen dan komt er een notificatie op het scherm.

## Klassendiagram / database ontwerp



Figuur 4‑12 Database ontwerp

Het bovenstaande database ontwerp (Figuur 4-12) is de basis van de database die voor dit project ontwikkeld gaat worden. De database is in principe simpel van structuur.

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/tijdschemas 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 lostaande table voor de instellingen, zo kan de gebruiker de standaard informatie van de app opslaan. Dit zijn instellingen zoals het ip adress van de Arduino of dat de app automatisch op de achtergrond taken mag uitvoeren.