

# GROEIDOCUMENT ICT-B1.2



## Inhoud

WEEK 1 .....	3
Opdracht 1: Project: Casus keuze .....	3
Opdracht 2: UX: doelgroep opdracht .....	4
Opdracht 3: UX: scenario opdracht .....	6
Opdracht 4: Basis vaardigheden: SDG & systeem denken .....	7
Opdracht 5: Technologie: robot laten rijden .....	8
Opdracht 6: Basisvaardigheden: Feedback geven tijdens sprint review .....	8
Week 1 Reflectie.....	8
WEEK 2 .....	9
Opdracht 1: Requirements - user story's.....	9
Opdracht 2: Requirements - use case diagram .....	10
Opdracht 3: Usability onderzoek (week 2-4).....	11
Opdracht 4: SW-ontwikkeling: single page app.....	12
Opdracht 5: Technologie: Sensoren .....	14
Opdracht 6: Basis vaardigheden: Feedback ontvangen tijdens de sprintreview .....	15
Week 2 Reflectie.....	15
Week 3 .....	16
Opdracht 1: Software development: Modelleren met UML (les van week 2).....	16
Opdracht 2: Software development: OOP first steps (les van week 2).....	19
Opdracht 3: MQTT in practice - Proof of Concept .....	21
Opdracht 4: Software ontwikkelen - Databases 1: Modelleren .....	22
Opdracht 5: UX Opdracht: Card sorting .....	23
Opdracht 6: Basisvaardigheden Presenteren: Tussenopdracht 0 en 1, stemgebruik en zelfverzekerdheid.....	24
Opdracht 7: Basisvaardigheden Systeemdenken: Opdracht 2 Systeemijsberg en Solutionary Scale .....	25
Loopbaan opdracht: Reflecteren .....	27
Week 3 reflectie .....	28
Week 4 .....	29
Opdracht 1: UX: Functioneel ontwerp (Wireframes) .....	29
Opdracht 2: Software ontwikkelen: Databases 2: Relatieel Database Ontwerp .....	31
Week 4 reflectie .....	33
Week 5 .....	33
Opdracht 1: UX Opdracht: Testing 1 .....	33
Week 5 reflectie .....	34
Week 6+ .....	34
Week 6+ reflectie.....	34
Conclusie: .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>

# WEEK 1

## Opdracht 1: Project: Casus keuze

In deze opdracht is het de bedoeling om de gekozen casus verder uit te werken en aan te tonen waarom ik hiervoor gekozen heb. Gekozen casus: "FZH - Langer zelfstandig thuiswonend met dementie".

### **Wat zijn de belangrijkste activiteiten die gebeuren?**

- *Dagelijkse ondersteuning:*

De robot herinnert mensen met dementie eraan om op tijd te eten en drinken.

Het verzamelt ook gegevens, zo krijg je een gezondheidsbeeld voor de zorgverleners.

- *Interactie met de gebruiker:*

De robot moet de gebruiker op een vriendelijke en interactieve manier benaderen. Het biedt eenvoudige en begrijpelijke aanwijzingen zodat de gebruiker geen stress ervaart.

- *Data-analyse en rapportage:*

Verzamelde gegevens worden verwerkt en weergegeven in een dashboard voor mantelzorgers en zorgverleners.

De data kan worden gebruikt om de robot meer gepersonaliseerd en nuttig te maken en voor de zorgverleners om te zien wat de stand van zaken is.

### **Wat is een interessant vraagstuk (probleem) dat je wil aanpakken?**

Hoe kan een zorgrobot effectief bijdragen aan het ondersteunen van dagelijkse routines bij mensen met dementie zonder hun stress te geven?

### **Welk mogelijke oplossing of toepassing kun je bedenken of heb je bedacht en hou zou robotisering hierbij een rol kunnen spelen?**

Een oplossing kan zijn dat de zorgrobot op basis van vaste tijden en verzamelde gegevens actief herinneringen geeft, sociale interactie of eventueel verbinding met slimme keukens zoals een automatische waterkraan. Ook een dashboard waar de robot zijn data-analyses naar toe stuurt en mantelzorgers informeert over routines en mogelijke risico's.

### **Wat heb jij nodig aan kennis en vaardigheden om dit te kunnen aanpakken?**

- Basis programmeer en robot kennis.
- Begrip van sensortechnologie en data-analyse.
- Het vertalen van zorgbehoeften naar technologie.
- Kennis van de specifieke behoeften van mensen met dementie.

### **Welke vragen heb jij voor de casus eigenaar?**

- Wat zijn de meest voorkomende problemen van mensen met dementie rondom de thuissituatie?
- Zijn er specifieke triggers of vormen van communicatie die goed werken bij deze doelgroep?

## Opdracht 2: UX: doelgroep opdracht

Tijdens de les hebben we 2 oefeningen gedaan om meer grip te krijgen op de doelgroep. Het is nu de bedoeling om dit te herhalen voor de uiteindelijk gekozen casus. De casus die ik in dit geval gekozen heb is: "FZH - Langer zelfstandig thuiswonen met dementie".

### DEEL 1:

#### **Wat heb je er uit kunnen halen?**

Om het probleem te omschrijven, De toenemende vergrijzing leidt tot een grotere druk op de zorg, wat samen met een tekort aan zorgverleners problematisch is.

Mensen met dementie hebben bepaalde uitdagingen, zoals het vergeten om op tijd en gezond te eten en drinken, medicijnen in te nemen of andere dingen die negatieve gevolgen heeft voor hun gezondheid.

Om dit probleem op te lossen zou je bijvoorbeeld technologie kunnen gebruiken om mensen met dementie langer te helpen met ondersteunen bij hun zelfstandigheid. Deze technologie moet wel aan bepaalde eisen voldoen, zo moet het gebruiksvriendelijk en interactief zijn. De data die de robot verzameld over bijvoorbeeld medicijn innname moet zichtbaar zijn voor zorgverleners via een dashboard.

#### **Wat weet je over je doelgroep?**

De doelgroep voor dit project betreft: Mensen met opkomend/vroege dementie, zo heeft deze bijvoorbeeld moeite met het onthouden van dagelijkse routines zoals eten en drinken. De ondersteuning moet vriendelijk zijn en eenvoudig te gebruiken zijn zodat deze persoon langer gezond en zelfstandig kan functioneren.

Ook de groep Zorgverleners en eventuele mantelzorgers zijn een betrokken doelgroep, doormiddel van een inzicht in de gezondheidsdata van de patiënt.

#### **Wat is de belangrijkste uitdaging voor de doelgroep?**

De grootste uitdaging is het behouden van zelfstandigheid thuis. Daarom is het essentieel om:

- Balans te vinden tussen technologische ondersteuning en menselijke betrokkenheid.
- Gezondheidsproblemen door vergeten activiteiten te voorkomen.
- Routines zo goed mogelijk te behouden

## DEEL 2

Hier ga ik aan de slag met zoeken naar ideeën voor de doelgroep binnen deze casus, vervolgens ga ik deze in 2 verschillende mindmaps verwerken.

Ik begin met het bedenken van ideeën die horen bij deze casus:

### ZORGROBOT VOOR EET- EN DRINKHERINNERINGEN

- Robot die gebruikers herinnert aan routines.
- Dashboard met data-analyse voor mantelzorgers.
- Interactieve functies zoals stemherkenning en spelletjes.

### ROBOT VOOR SOCIALE INTERACTIES

- Robot die naast zorgtaken ook sociale interactie kan bieden.
- Gesprekken voeren en/of muziek afspelen.
- Fysieke oefeningen of ademhalingsoefeningen begeleiden.

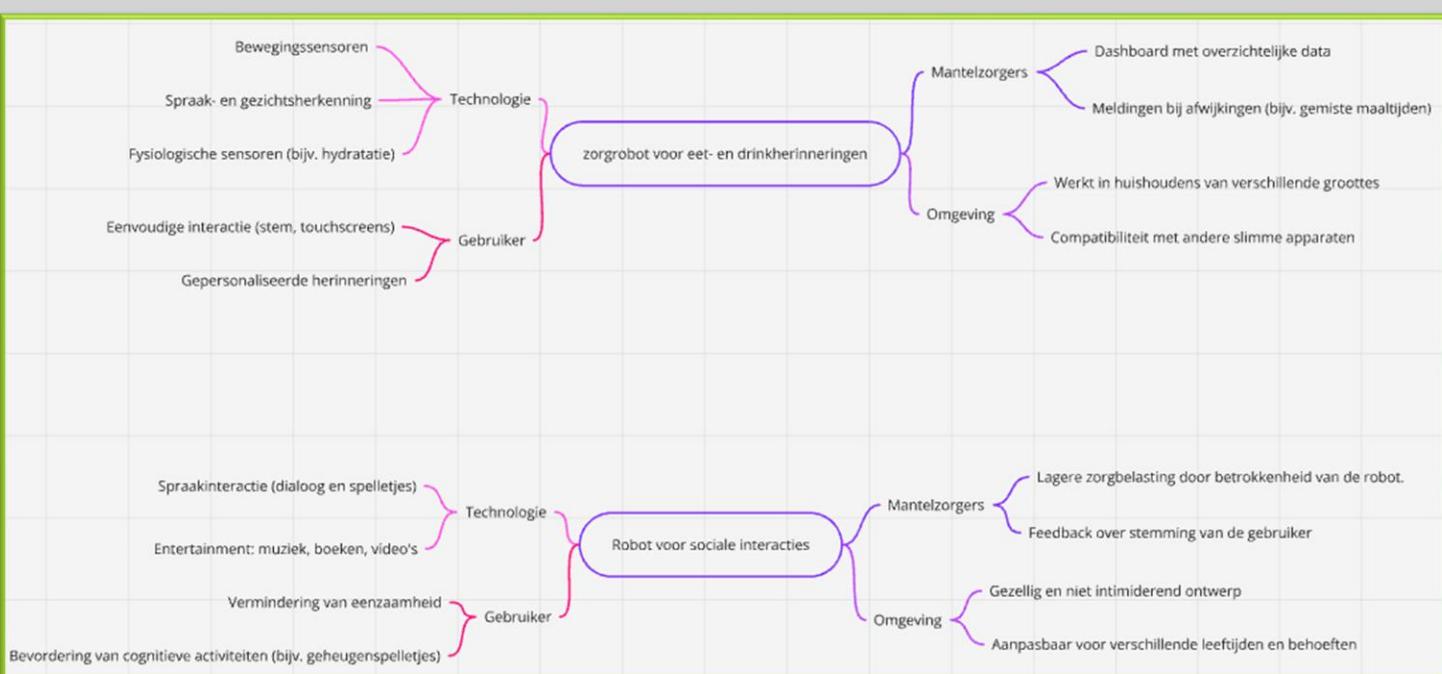
### AUTOMATISCHE MAALTIJDPLANNER

- Robot die maaltijden voordraagt op basis van de persoonlijke gezondheid.
- Verbinding met slimme keukenapparaten.
- Neemt allergieën of dieetvoorschriften op.

### VEILIGHEIDSMONITOR

- Robot die bijvoorbeeld valincidenten detecteert en hulpdiensten waarschuwt.
- Rijdt rond door het huis om te controleren op veiligheidsrisico's.

*Hieronder heb ik deze ideeën uitgewerkt in de vorm van 2 mindmaps:*



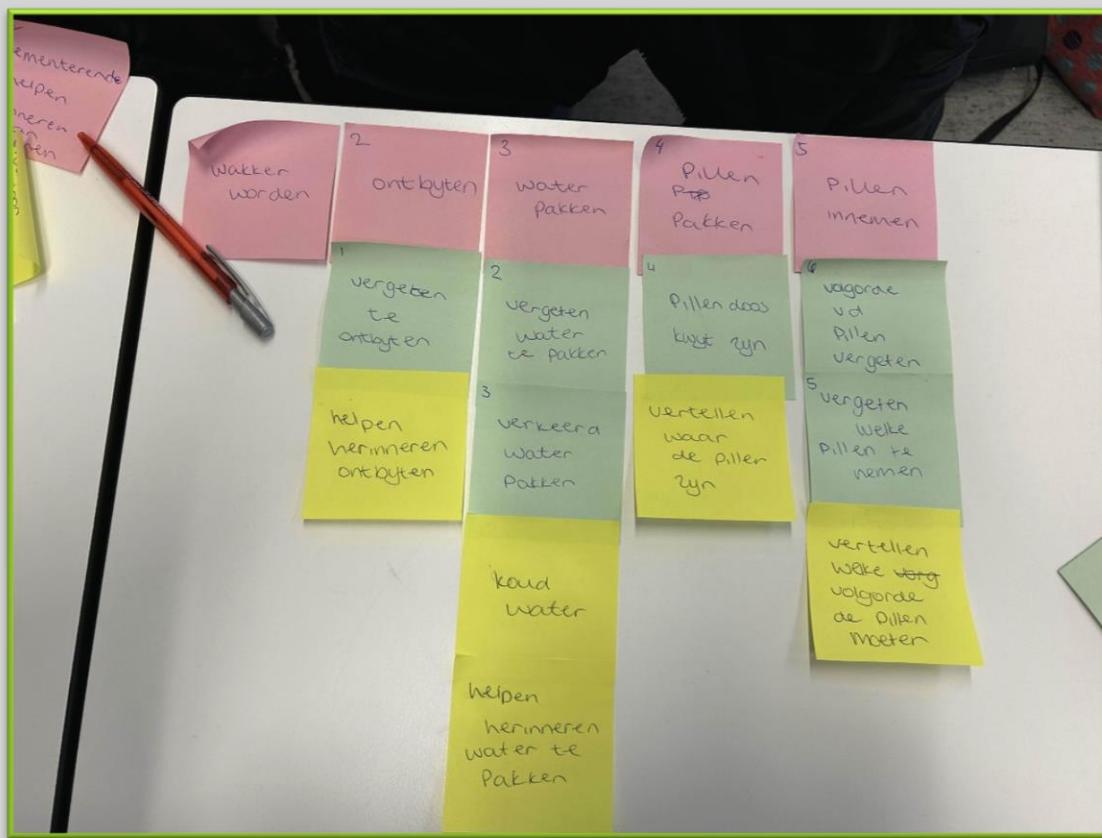
## CONCLUSIE:

De zorgrobot voor eet- en drinkherinneringen heeft de meeste connecties en directe voordelen voor de doelgroep. Het combineert innovatie met een dringende zorgbehoefte.

## Opdracht 3: UX: scenario opdracht

Bij deze les/opdracht was het de bedoeling om scenario's te bedenken, deze geven een rijk beeld van belangrijke factoren, handelingen en behoeften die de gebruikers bij de uitvoering ervaren. Ik heb zoals eerder aangegeven gekozen voor "FZH - Langer zelfstandig thuiswonend met dementie" en daar gaan deze notities dus ook over.

Het scenario-mapping ik gedaan met 2 medestudenten, tijdens de les hebben wij hier ook feedback voor gevraagd en deze meteen verwerkt in het resultaat die hierboven te zien is. Zo kregen we feedback over bepaalde dingen waar je extra op moest letten maar kregen we ook complimenten over de meelevendheid die we toonden door de connecties op deze manier te leggen.



Als ik terugkijk op deze les en oefening vond ik het enorm lastig om in grote details mijzelf in een situatie te zetten, desondanks vind ik het wel een goede en duidelijke uitwerking.

## Opdracht 4: Basis vaardigheden: SDG & systeem denken

In deze opdracht pas ik toe wat we in de les systeem denken geleerd hebben. Hierbij kies ik een casus en dit word “FZH - Langer zelfstandig thuiswonen met dementie.

### SDG's waar deze casus effect op heeft:

#### Positief:

##### SDG 3: Goede gezondheid en welzijn

- Het effect is dat de robot mensen gaan helpen met dementie doormiddel van herinneringen voor eten en drinken en andere zaken die gezondheidsproblemen zoals ondervoeding of uitdroging voorkomen.

**Systeemdenken:** *Een gezondere doelgroep verlaagt de druk op mantelzorgers en andere zorgverleners, wat een positieve impact heeft.*

##### SDG 9 : Industrie, innovatie en infrastructuur

- Het project stimuleert technologische innovatie in de zorg en daardoor komt er een sterker samenwerkingsmodel met publieke en private partners.

**Systeemdenken:** *innovatie in robotica zorgt voor een kettingreactie, waarbij oplossingen in deze casus ook in andere zorgdomeinen toepasbaar kunnen worden.*

#### Negatief:

##### SDG 10: ongelijkheid verminderen

- Niet iedereen heeft natuurlijk toegang tot zorgbots door digitale vaardigheden of financiële problemen.

**Systeemdenken:** *Dit kan ongelijkheden vergroten en de druk op de zorg dus laten verhogen.*

#### Conclusie:

De robot biedt grote voordelen voor gezondheid en innovatie, maar ook vraagt het aandacht voor eerlijke toegang zodat de negatieve effecten beperkt worden.

## Opdracht 5: Technologie: robot laten rijden

In deze opdracht moesten we alle software en benodigdheden installeren en test-code uploaden naar de robot om te testen of de motoren goed functioneren, als dat zo was zou de robot een rondje draaien.

Tijdens deze opdracht heb ik nauwkeurig de installatie-handleiding gevolgd en heb ik alle benodigde drivers, de IDE en toevoegingen aan deze IDE geïnstalleerd. De code was al als “test” variant geleverd door te drukken op “File>Examples>Romi32u4”. Daarna heb ik de robot met het gekregen draadje in mijn laptop gedaan en de geleverde code geüpload naar de robot. Na lang zoeken vond ik eindelijk de aan-knop en uiteindelijk ook de knop die de robot liet draaien. De robot ging draaien dus dat betekent dat de motoren goed functioneren.

(Het is onduidelijk voor mij hoe ik voor deze opdracht feedback moet vragen?)

Om een korte reflectie te geven over deze les, over het algemeen ging het goed, alles verliep soepel. De knopjes vinden was wat lastiger maar ook dat had ik snel onder de knie.

## Opdracht 6: Basisvaardigheden: Feedback geven tijdens sprint review

Bij deze opdracht is het de bedoeling om te noteren hoe je feedback ontvangen en gegeven hebt in de wekelijkse sprintreview lessen. Deze lessen zijn er om presenteren te oefenen vanwege het opleverpunt aan het einde van de periode waar we een videopresentatie moeten opleveren.

Omdat in de eerste week deze les is uitgevallen en dus geen feedback heb kunnen geven of ontvangen kan ik in deze versie van het groeidocument deze opdracht niet goed maken. Aangezien we deze les aankomende week wel hebben ga ik er voor zorgen dat deze alsnog gemaakt word en af is op het 2<sup>de</sup> opleverpunt.

## Week 1 Reflectie

Reflectie op de eerste week is nogal lastig omdat ik niet tegen veel dingen aanliep en daarom reflecteer ik deze week vooral op de positieve dingen die juist goed gingen. Zo vond ik het enorm gaaf om te zien hoe mijn robot goed functioneerde en vond ik de lessen ook enorm interessant.

## WEEK 2

### Opdracht 1: Requirements - user story's

Tijdens de lessen UX-design hebben we onderzoek gedaan naar de doelgroep en hun behoeftes. Op basis van wat hier is uitgekomen maken we een beschrijving van mogelijke requirements in meerdere user story's. Deze ga ik opnemen in deze opdracht.

Als eerst hebben we een college gehad met informatie hierover, hierna heb ik verdere slides bekeken die bij deze opdracht stonden.

Hieronder heb ik wat behoeftes genoteerd met daaronder de bijbehorende userstory

#### 1. Routineherinneringen

Als een gebruiker met beginnende dementie wil ik op vriendelijke wijze herinneringen ontvangen voor mijn dagelijkse routines zoals eten, drinken en medicijngebruik, zodat ik langer zelfstandig thuis kan wonen en gezond blijf.

#### 2. Toegankelijke technologie

Als een gebruiker met beperkte technische vaardigheden wil ik dat de zorgrobot eenvoudig te bedienen is zodat ik zonder frustratie of minimale extra hulp gebruik kan maken van de robot.

#### 3. Inzicht voor mantelzorgers

Als een mantelzorger of zorgprofessional wil ik via een dashboard inzicht krijgen in de routines en gezondheid van de patiënt, zo kan ik controleren of deze veilig en gezond is en indien nodig actie kan ondernemen op een zichtbaar probleem.

#### 4. Sociale interactie

Als een gebruiker die zich soms eenzaam voelt wil ik dat de robot ook sociale functies biedt zoals gesprekken voeren of muziek afspelen, zodat ik mij minder eenzaam voel in mijn dagelijkse leven.

#### 5. Veiligheidsmonitoring

Als een gebruiker die alleen woont wil ik dat de robot valincidenten en andere veiligheidsproblemen kan detecteren en hulpdiensten kan inschakelen, zo kan ik veilig thuis blijven wonen zonder continu afhankelijk te zijn van anderen.

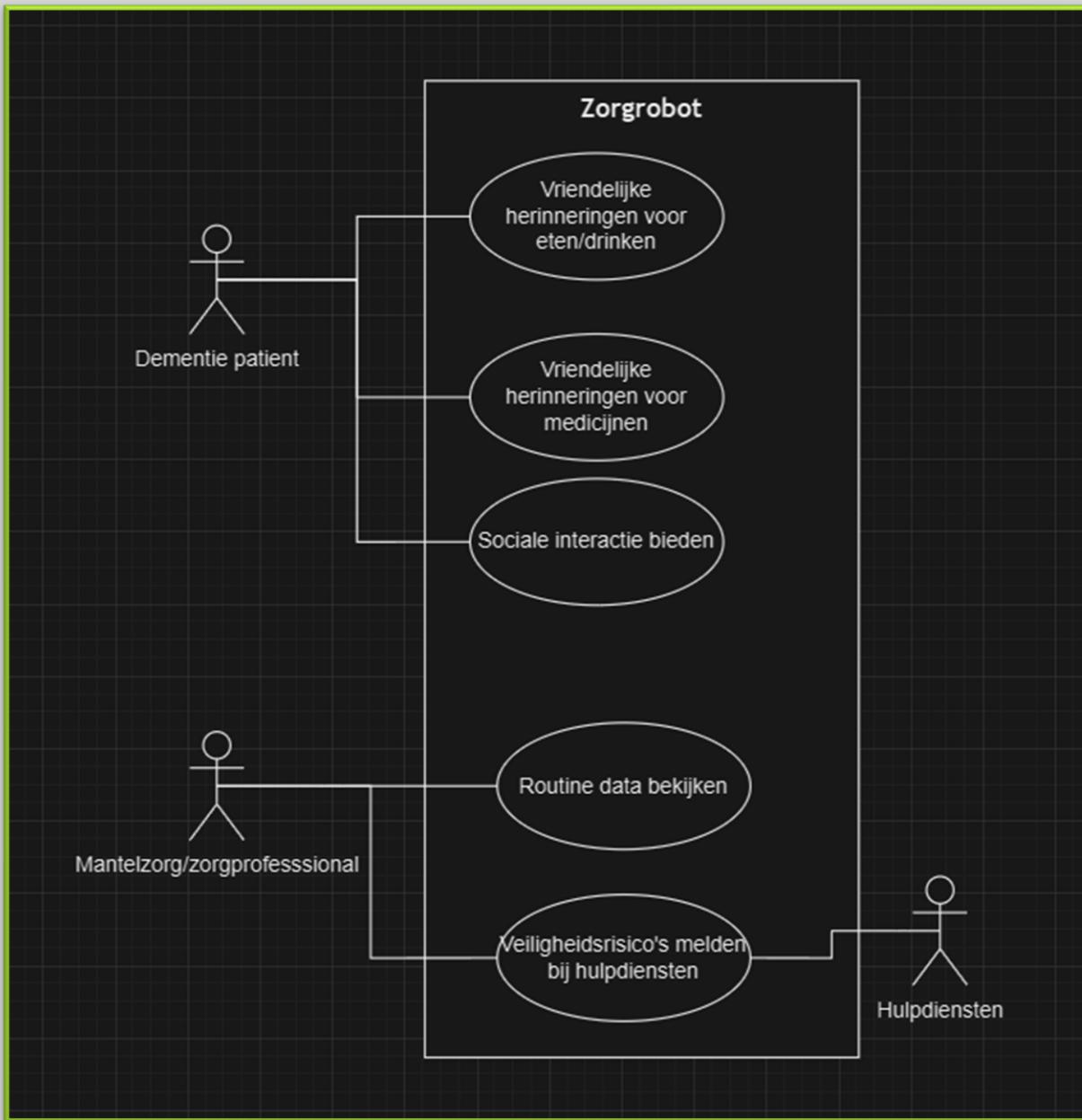
#### Reflectie:

De user story's hebben geholpen om meer inzicht te krijgen in de behoeftes van de verschillende gebruikers. Zo heb ik routineherinneringen en veiligheidsmonitoren als belangrijk punt staan omdat deze direct te maken hebben met de zelfstandigheid en veiligheid van de gebruiker. Daarnaast is sociale interactie een extra optie om patiënten minder eenzaam te laten voelen. Als ik terugkijk is dit een fijne manier van kijken naar eisen die een klant zou hebben.

## Opdracht 2: Requirements - use case diagram

In deze opdracht ga ik de user story's die ik in de vorige opdracht gemaakt had verder uitwerken. Dit doe ik doormiddel van een “use case diagram”. De tool waarmee ik deze use-cases ga maken word: <https://app.diagrams.net/>

De use case diagram:



Ik heb hier als titel “Zorgrobot” omdat de actors allemaal gebruik maken of direct te maken hebben met de robot. Zo heeft de dementerende patiënt hulp bij herinneringen en eenzaamheid en kan de zorg risico's vaststellen doormiddel van data, bij nood gevallen worden eventuele hulpdiensten ingeschakeld.

## Opdracht 3: Usability onderzoek (week 2-4)

In deze opdracht ga ik kijken naar verschillende design principes voor Usability, hierbij gebruik ik de “10-regels” voor hoe gebruiksvriendelijk mijn applicatie word. Deze zijn besproken in de UX-les in week 2 en in deze opdracht ga ik hier verder op in.

Ik moest voor de opdracht 3 heuristieken kiezen om meer in details te gaan:

- Visibility of System Status
- Flexibility and Efficiency of Use
- Aesthetic and Minimalist Design

*Ik heb deze gekozen doormiddel van de bijgevoegde foto in de opdrachtpage(ook hieronder ingevoegd)*

### Visibility of system status:

Houdt in dat het de gebruiker ten alle tijden moet laten zien wat de status van het systeem is, dit moet op tijd en fatsoenlijk gebeuren zodat de gebruiker weet waar hun aan toe zijn. Het nadeel is dat je zelfs snelle problemen dan moet aangeven ook al zou je deze snel kunnen aanpassen, of dat bepaalde problemen de wereld in komen wat eventuele bezoekers/gebruikers weg zou jagen.

### Flexibility and Efficiency of Use:

Hier gaat het vooral om het beschikbaar maken van shortcuts, zo kunnen mensen die al langer het systeem gebruiken sneller navigeren of andere zaken regelen. Het nadeel is dat je wel goed rekening moet houden met nieuwe gebruikers, deze weten niet af van de shortcuts maar zijn net zo goed een bezoeker/gebruiker.

### Aesthetic and Minimalist Design:

Als je een pagina leest moet de inhoud nuttig zijn, niet zinloos. Elk stukje extra zinloze informatie zorgt er voor dat de wel nuttige en belangrijke informatie minder zichtbaar word. Zo hou je dus het design maar ook de inhoud nuttig en minimalistisch. Fijner om te gebruiken/lezen dus. Een nadeel is dat het zou kunnen leiden tot een hele lege pagina met gebrek aan nuttige informatie.

**1 Visibility of System Status**  
Designs should keep users informed about what is going on, through appropriate, timely feedback.  
 Interactive mall maps have to show people where they currently are, to help them understand where to go next.

**2 Match between System and the Real World**  
The design should speak the user's language. Use words, phrases, and concepts familiar to the user, rather than internal jargon.  
 Users can quickly understand what stove-top control maps to each heating element.

**3 User Control and Freedom**  
Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action.  
 Just like physical spaces, digital spaces need quick "emergency" exits too.

**4 Consistency and Standards**  
Users should not have to wonder whether different words, situations, or actions mean the same thing.  
 Follow platform conventions.  
Check-in counters are usually located at the front of hotels, which meets expectations.

**5 Error Prevention**  
Good error messages are important, but the best designs carefully prevent problems from occurring in the first place.  
 Guard rails on curvy mountain roads prevent drivers from falling off cliffs.

**6 Recognition Rather Than Recall**  
Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.  
 People are likely to correctly answer "Is Lisbon the capital of Portugal?".

**7 Flexibility and Efficiency of Use**  
Shortcuts — hidden from novice users — may speed up the interaction for the expert user.  
 Regular routes are listed on maps, but locals with more knowledge of the area can take shortcuts.

**8 Aesthetic and Minimalist Design**  
Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface competes with the relevant units of information.  
 A minimalist three-legged stool is still a place to sit.

**9 Recognize, Diagnose, and Recover from Errors**  
Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.  
 Wrong-way signs on the road remind drivers that they are heading in the wrong direction.

**10 Help and Documentation**  
It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users complete their tasks.  
 Information kiosks at airports are easily recognizable and solve customers' problems in context and immediately.

## Opdracht 4: SW-ontwikkeling: single page app

In deze les hebben we een stukje herhaling van vorige periode gekregen: Blazor. We gingen later in de les dieper in op het refactoren van code door van delen code een “service” te maken, deze component kun je op meerdere punten oproepen zodat je het kan herhalen. Ik heb gekozen om variant B te maken voor wat extra uitdaging.

Als eerste heb ik een account aangemaakt op de weather API:

Key	Name	Status	Actions	Create key
[REDACTED]	Default	Active		<input type="text" value="API key name"/> <button>Generate</button>

(Ik heb natuurlijk de KEY onherkenbaar gemaakt in de screenshots. Vandaar de rode strepen)

Daarna heb ik classes gemaakt om de data van de API die in JSON-format komt goed op te halen:

```
3 references
public class WeatherData
{
    1 reference
    public string Name { get; set; } = string.Empty;
    3 references
    public MainInfo Main { get; set; } = new MainInfo();
    1 reference
    public List<WeatherCondition> Weather { get; set; } = new List<WeatherCondition>();
    1 reference
    public WindInfo Wind { get; set; } = new WindInfo();
}

2 references
public class MainInfo
{
    2 references
    public double Temp { get; set; }
    1 reference
    public int Humidity { get; set; }
}

2 references
public class WeatherCondition
{
    1 reference
    public string Description { get; set; } = string.Empty;
}

2 references
public class WindInfo
{
    1 reference
    public double Speed { get; set; }
}
```

```

C# GetWeatherService.cs
Data > C# GetWeatherService.cs > WeatherData
1  using System.Text.Json;
2  4 references
2  public class GetWeatherService
3  {
4      2 references
4      private readonly HttpClient _httpClient;
5
6      0 references
6      public GetWeatherService(HttpClient httpClient)
7      {
8          _httpClient = httpClient;
9      }
10
11     1 reference
11     public async Task<WeatherData> FetchWeatherDataAsync(string apiLink)
12     {
13         try
14         {
15             var response = await _httpClient.GetAsync(apiLink);
16
17             if (response.IsSuccessStatusCode)
18             {
19                 var jsonResponse = await response.Content.ReadAsStringAsync();
20                 return JsonSerializer.Deserialize<WeatherData>(jsonResponse, new JsonSerializerOptions
21                 {
22                     PropertyNameCaseInsensitive = true
23                 });
24             }
25             else
26             {
27                 Console.WriteLine($"Error fetching weather data: {response.StatusCode}");
28                 return null;
29             }
30         }
31         catch (Exception ex)
32         {
33             Console.WriteLine($"Exception: {ex.Message}");
34             return null;
35         }
36     }
37 }
38

```

Hierboven is de gemaakte service voor de weather API. Later heb ik ook bij de “program.cs” een line toegevoegd die de service werkend maakt:

```

3
4     builder.Services.AddSingleton<GetMagicNumberService>();
5     builder.Services.AddSingleton<GetWeatherService>();
6     builder.Services.AddHttpClient();
7

```

Tot slot de Razor file aangepast om de service op te roepen:

```

<tbody>
    <tr>
        <td>@weatherData.Name</td>
        <td>@weatherData.Main.Temp.ToString("F1")</td>
        <td>@(32 + (int)(weatherData.Main.Temp / 0.5556))</td>
        <td>@weatherData.Main.Humidity%</td>
        <td>@weatherData.Weather.FirstOrDefault()?.Description</td>
        <td>@weatherData.Wind.Speed m/s</td>
    </tr>
</tbody>
</table>
}

```

```

@code {
    private WeatherData? weatherData;

    protected override async Task OnInitializedAsync()
    {
        const string apiLink = "https://api.openweathermap.org/data/2.5/weather?q=London&units=metric&appid=2e0a7a2a2a2a2a2a2a2a2a2a2a2a2a2a";
        weatherData = await WeatherService.FetchWeatherDataAsync(apiLink);
    }
}

```

## Opdracht 5: Technologie: Sensoren

In deze les kregen we uitleg over hoe sensoren werkten en hoe we de raspberry pi moesten aansluiten. Ik heb dit niet in de les gedaan i.v.m. dat je de wifi moest aansluiten en ik dat liever niet op het school netwerk deed, uiteindelijk heb ik het thuis op mijn netwerk aangesloten doormiddel van de handleiding, heb ik de sensor aangesloten en ben ik gaan programmeren.

Na wat onderzoek kwam ik er achter dat je de button ook apart moest aanroepen, zo kun je zien of de knop daadwerkelijk wordt ingedrukt of losgelaten wordt. Als je “CTRL + click” doet op een class kom je er heen en zie je bepaalde eigenschappen hiervan. Zo kwam ik tegen dat er een “GetState()” method was waar je “Pressed” of “Released” kon opmeten.

Hier heb ik gebruik van gemaakt doormiddel van een if statement, deze kijkt of er op gedrukt wordt. Daarbij wordt de status van de led aangepast zodat hij blijft ook al laat je de knop los. En als je er nog een keer op drukt en die status dus “aan” is, gaat hij ook weer uit. Om het af te maken is dit allemaal in een “While-loop” zodat het pas stopt als de robot stopt en er een kleine delay zodat je de knop niet te snel kan indrukken.

```
# Program.cs
1  using System.Device.Gpio;
2  using Avans.StatisticalRobot;
3
4  Led led5 = new ...;
5  led5.SetOff();
6
7  Button button6 = new ...;
8 💡
9  bool isLedOn = false;
10
11 while (true)
12 {
13     if (button6.GetState() == "Pressed")
14     {
15         isLedOn = !isLedOn;
16         if (isLedOn)
17         {
18             led5.SetOn();
19         }
20         else
21         {
22             led5.SetOff();
23         }
24
25         Robot.Wait(200);
26     }
27 }
```

## Opdracht 6: Basis vaardigheden: Feedback ontvangen tijdens de sprintreview

Tijdens de sprintreview moesten we bewust feedback ontvangen van anderen. Tijdens de intervisie op donderdag had ik alleen niet iets waar ik tegenaan liep, nergens had ik een puntje waar ik echt vastliep en hulp bij nodig had, ook was er niemand die mij een algemene tip gaf. Wel luisterde ik aandachtig naar de tips en problemen die de anderen kregen zodat ik daar wel nog van kon leren.

Daarbij kwam ook nog het probleem dat er tijdens de les maar 4 andere leerlingen waren omdat ze naar de eerste les gingen ook al moesten ze eigenlijk bij het 2<sup>de</sup> uur aanwezig zijn, 5 mensen in totaal dus. Hierdoor was de hoeveelheid feedback en gespreksmogelijkheden enorm verminderd. Als ik een betere intervisie in de opkomende weken bijwoon ga ik deze opdracht waarschijnlijk daar op aanpassen zodat het nuttiger overkomt.

*(nu ik de opdracht opnieuw lees denk ik dat dit meer gaat over de feedback die je zou krijgen bij het presenteren, wat we in week 3 pas hebben. Als dit het geval is zal ik deze opdracht ook daarom opnieuw maken maar voor nu heb ik dan toch nog dit.)*

- Liked: Wat ik nuttig vond was het bespreken van algemene problemen, de dingen die we bespraken was niet per se iets waar ik zelf moeite mee had maar wel interessant om te zien hoe andere mensen hier naar kijken.
- Learned: Ik heb geen feedback ontvangen.
- Lacked: Ik had geen moeite met feedback ontvangen, ik sta er altijd voor open en luister graag naar tips die andere te bieden hebben.
- Longed for: Ik vind dat ik beter van te voren kan nadenken of er dingen zijn waar ik moeite mee heb zodat mensen daar passende feedback op kunnen geven, op dit moment heb ik nauwelijks tot geen moeite met opdrachten en andere punten en daarom is dit lastig.

## Week 2 Reflectie

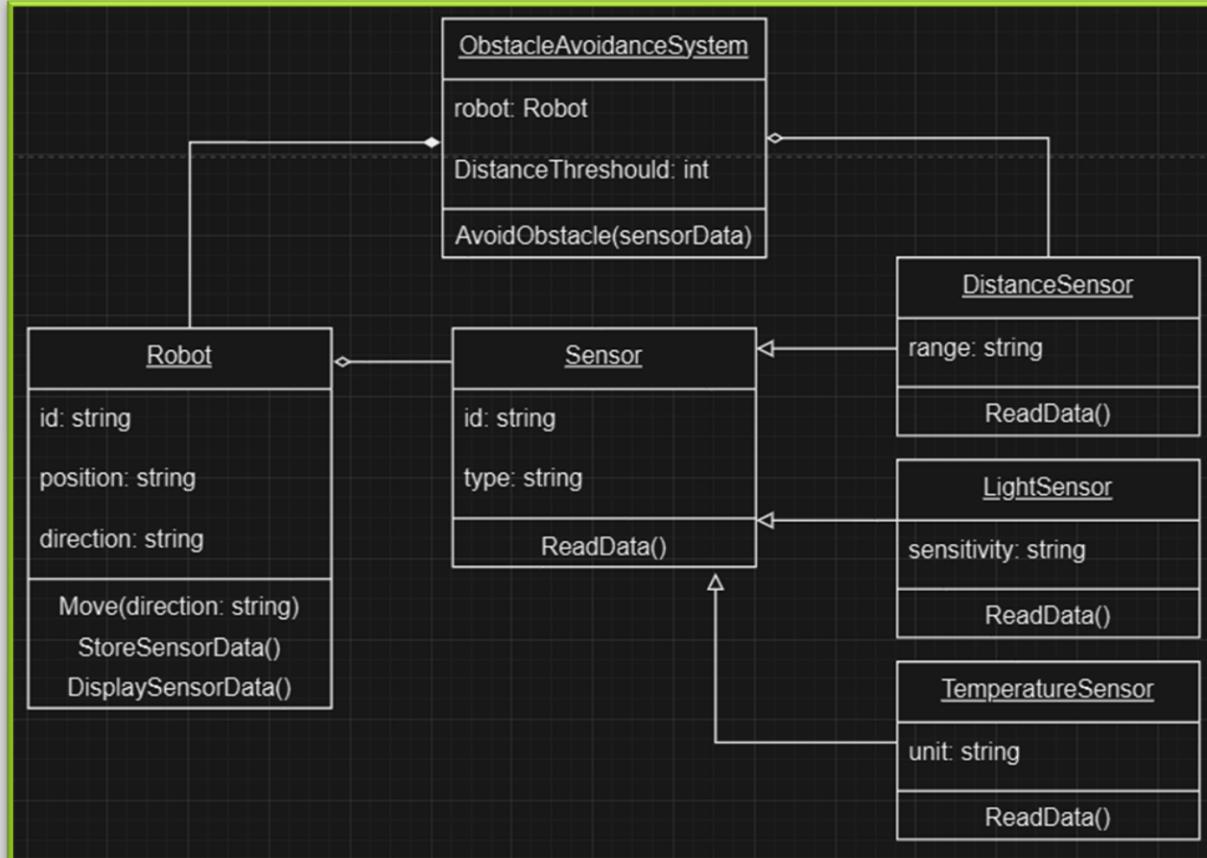
Deze week liep ik niet tegen grote problemen aan, ik heb weinig feedback kunnen vragen en ik wil dit voor volgende week wel een stuk meer gaan doen omdat ik merk dat dit echt nog te weinig is. Ook zijn de reflectie stukken in het groeidocument nog een beetje onduidelijk voor mij en ook hier ga ik duidelijkheid over vragen en eventueel aanpassingen op doen indien nodig.

## Week 3

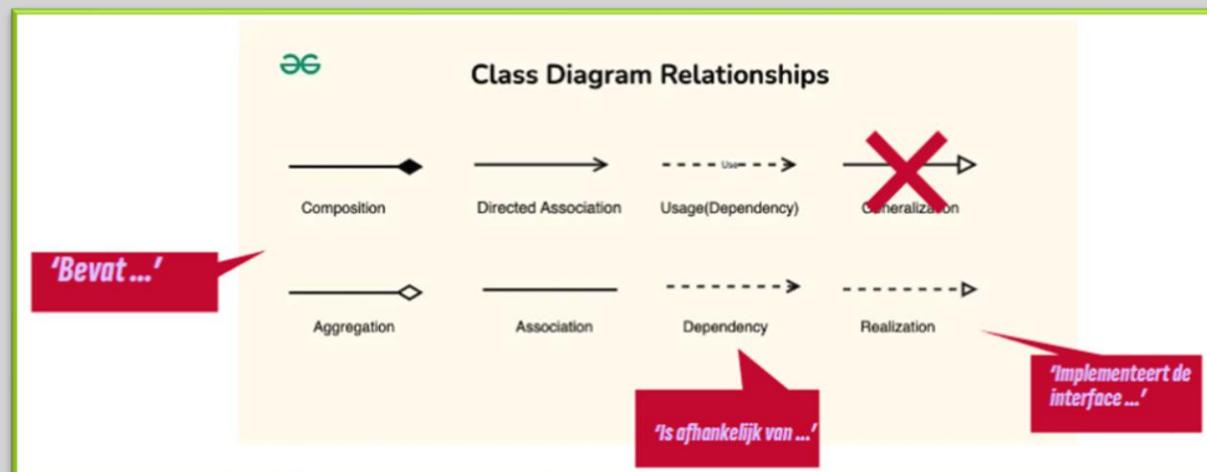
### Opdracht 1: Software development: Modelleren met UML (les van week 2)

In deze opdracht hou ik me bezig met het modelleren van een oefenproject, dit helpt bij het beter ontwerpen van projecten en het geeft meer duidelijkheid. De opdracht bestaat uit 4 deelopdrachten.

#### Opdracht 1/4: UML class diagram



Na feedback te vragen aan een klasgenoot kwam ik er achter dat mijn eerste versie van de class diagram totaal fout was, ik was in de war met andere concepten. Ik heb deze daarna opnieuw gemaakt en die is dus hierboven te zien.



## Opdracht 2/4: UML Use Case diagram:

### Actoren:

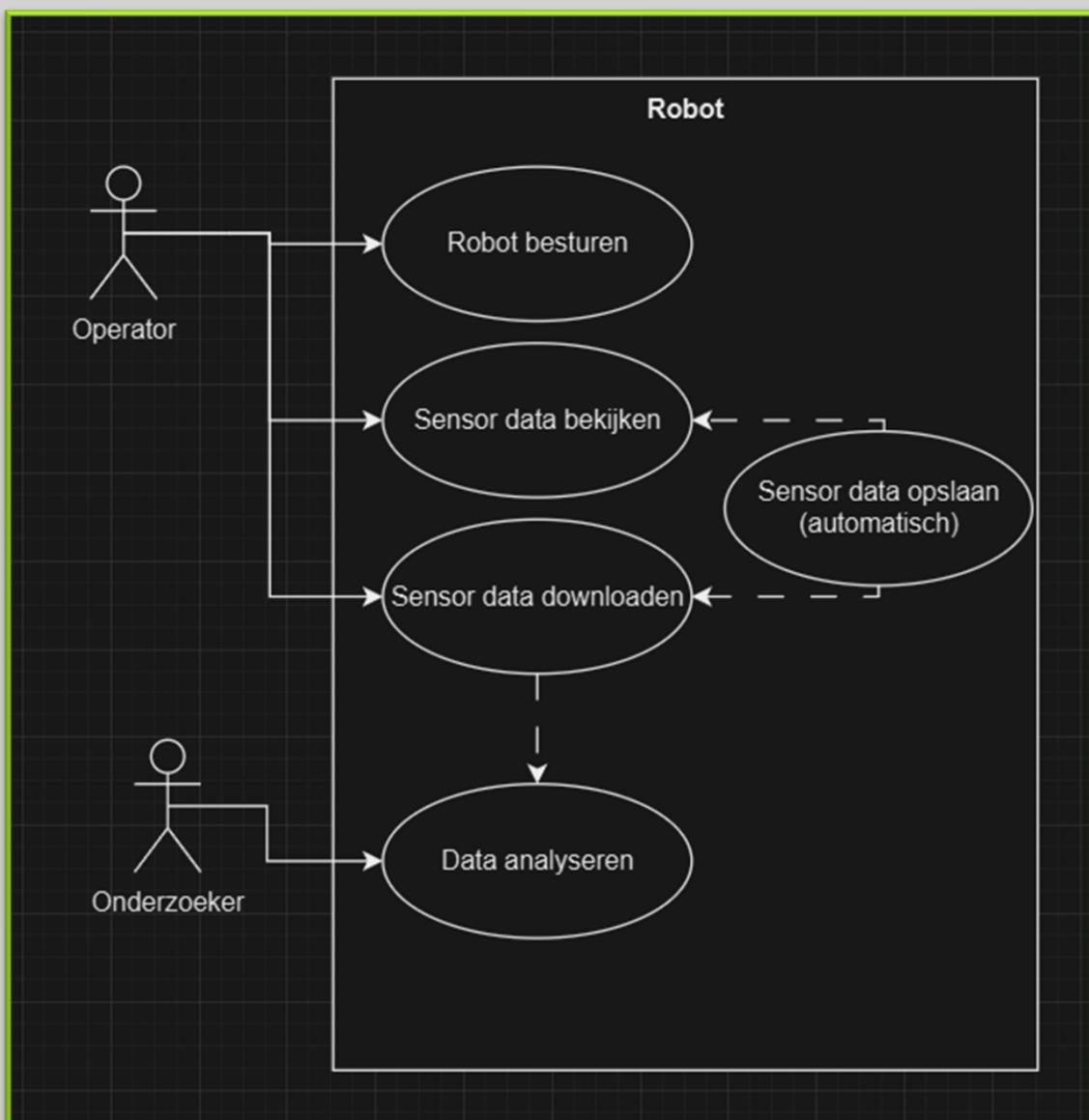
- **Operator:** Bestuurt de robot en beheert sensor data.
- **Onderzoeker (uitbreiding):** Analyseert de gedownloade data.

### Use Cases:

- Robot besturen (voortaan, achteruit, links, rechts).
- Sensor data bekijken.
- Sensor data downloaden.
- Sensor data opslaan (automatisch).

### Relaties:

- Include: De operator moet sensor data opslaan voordat deze bekijken of gedownload kan worden.
- Extend: Sensor data bekijken en downloaden kunnen afhankelijk zijn van een specifieke tijd of trigger.



## Opdracht 3/4 UML Activity diagram

Start: De robot begint met rijden.

Elke 5 seconden verzamelt de robot sensor data.

Als de afstandsmeter een object dichterbij dan 10 cm detecteert:  
*Stop met rijden -> Draai 90 graden naar links -> Rij verder.*

Als de verkenningsperiode is bereikt (bijvoorbeeld 30 seconden), stop de robot.

(Uitbreiding) Als de lichtintensiteit onder een drempel komt, schakel een lamp in.

## Opdracht 4/4 Volgorde diagrammen

De volgorde waarin de diagrammen worden gepresenteerd is logisch maar niet altijd even handig om mee te werken.

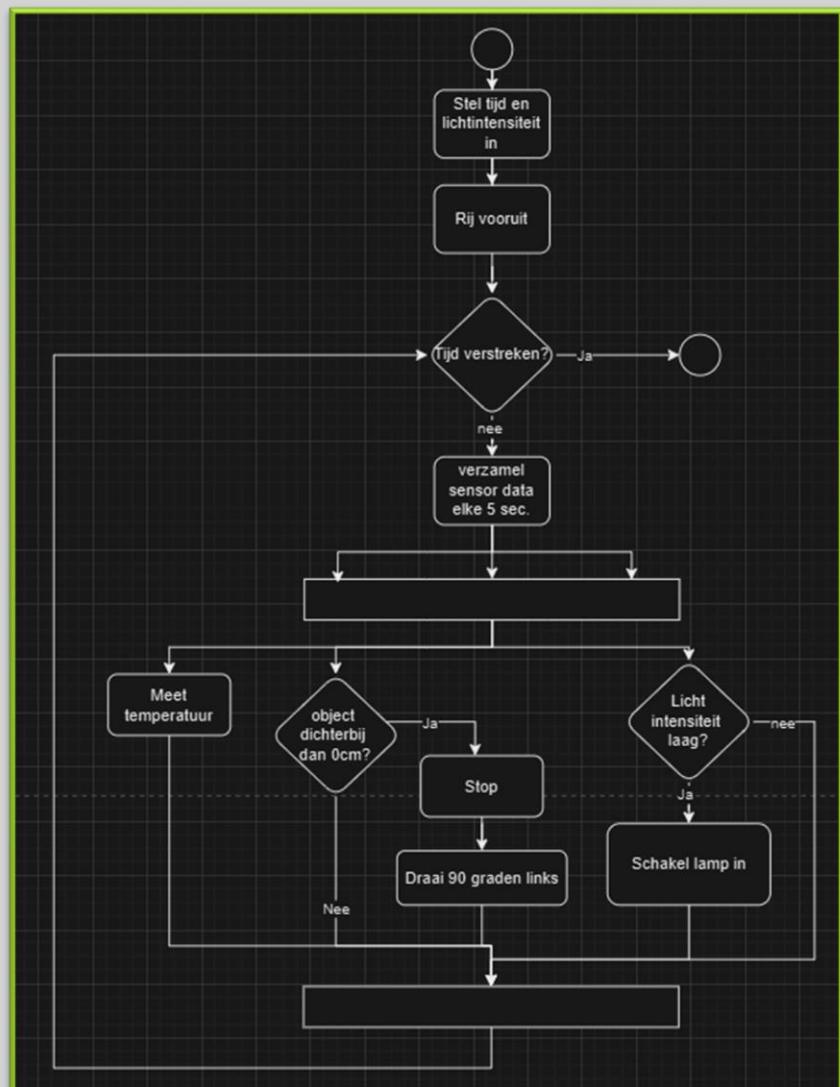
De volgorde:

- Het **UML Class Diagram** helpt om de structuur van het systeem te begrijpen en dient als een hulpmiddel voor de code.
- Het **UML Use Case Diagram** toont de interactie tussen gebruikers en het systeem en helpt om functionaliteiten vast te stellen.
- Het **UML Activity Diagram** legt de workflow uit en kan gebruikt worden om algoritmes te ontwikkelen.

Doelgroep:

- **Class Diagram:** Nuttig voor ontwikkelaars als een soort handleiding voor de code.
- **Use Case Diagram:** Gesprekken met niet-technische stakeholders.
- **Activity Diagram:** Dient als een soort flowchart voor ontwikkelaars.

**Conclusie:** De volgorde is logisch voor documentatie, maar in de praktijk zou je met de Use Cases beginnen om de eisen van de stakeholders te begrijpen, dan het Class Diagram maken voor de basis van het programma, en vervolgens het Activity Diagram om het verder uit te werken.



## Opdracht 2: Software development: OOP first steps (les van week 2)

In deze opdracht was het de bedoeling om de gekregen stof in de OOP les van week 2 toe te passen in een oefen opdracht, de opdracht die we moesten maken was een soort uitbreiding van waar we in de les al mee begonnen waren. Om het kort samen te vatten moesten we een soort “robot-simulatie” maken in een console app, met sensoren en data die random is. Dit heb ik gedaan doormiddel van Classes en Interfaces.

Ik vind zelf mappenstructuur fijn en overzichtelijk, zo kan ik snel en gemakkelijk bestanden terugvinden in het geval dat er steeds meer bijkomen:

The screenshot shows a file explorer window with the following structure:

- Root folder: OOPCONCEPTS
  - bin
  - Models
    - Actuator
      - C# HonkActuator.cs
      - C# IActuator.cs
    - Sensors
      - C# ISensor.cs
      - C# LightSensor.cs
      - C# SpeedSensor.cs
      - C# TemperatureSensor.cs
  - obj
    - Debug
      - OopConcepts.csproj.nuget.dgspec.json
      - OopConcepts.csproj.nuget.g.props
      - OopConcepts.csproj.nuget.g.targets
    - project.assets.json
    - project.nuget.cache
    - OopConcepts.csproj
    - OopConcepts.sln
    - C# Program.cs

Folder voor alle klassen en dergelijke.

Sensoren en actuatoren gesplitst, robot in de gewone Models map

Hieronder is te zien hoe ik te werk ben gegaan in de interface van de sensoren, de naam en eenheid hebben elke sensor en de methodes om deze data terug te geven ook.

```
Models > Sensors > C# ISensor.cs > ...
1  namespace OopConcepts
2  {
3      public interface ISensor
4      {
5          string Name { get; }
6          string Unit { get; }
7          int GetMeasurement();
8          string GetMeasurementAsText();
9      }
10 }
```

```

Models > Sensors > C# TemperatureSensor.cs > ...
1  namespace OopConcepts
2  {
3      3 references
4      public class TemperatureSensor : ISensor
5      {
6          2 references
7          private string _name;
8          1 reference
9          private string _unit = "Celsius";
10
11         2 references
12         public TemperatureSensor(string name)
13         {
14             _name = name;
15         }
16
17         2 references
18         public string Name => _name;
19         2 references
20         public string Unit => _unit;
21
22         2 references
23         public int GetMeasurement()
24         {
25             return new Random().Next(0, 80);
26         }
27
28         2 references
29         public string GetMeasurementAsText()
30         {
31             return $"Sensor: {Name}, value: {GetMeas
32         }
33     }
34 }
```

Om alle sensoren te gebruiken en een robot aan te maken heb ik een robot class gemaakt, hier worden alle actuatoren en sensoren verzameld in lijsten, doorheen gekeken en komt de bijbehorende data in de console. Hier word ook het aanmaken van de robot geregeld. Ik heb gebruik gemaakt van een switch case om het type robot aan te geven en de sensoren die hierbij horen.

Daarna heb ik de sensor Classes aangemaakt, hieronder een voorbeeld. In dit geval de temperatuursensor:

```

Models > C# WheeledRobot.cs
1  namespace OopConcepts
2  {
3      public class WheeledRobot
4      {
5          private string _name;
6          private int _type;
7          private List<ISensor> _sensors;
8          private List<IActuator> _actuators;
9
10         public WheeledRobot(string name, int type)
11         {
12             _name = name;
13             _type = type;
14             _sensors = new List<ISensor>();
15             _actuators = new List<IActuator>();
16             UseSensors();
17         }
18
19         public string Name => _name;
20
21         private void UseSensors()
22         {
23             switch(_type)
24             {
25                 case 1:
26                     _sensors.Add(new SpeedSensor("SpeedSensor"));
27                     _sensors.Add(new TemperatureSensor("TemperatureSensor"));
28                     _actuators.Add(new HonkActuator("HonkActuator"));
29                     break;
30                 case 2:
31                     _sensors.Add(new LightSensor("LightSensor"));
32                     _sensors.Add(new TemperatureSensor("TemperatureSensor"));
33                     _actuators.Add(new HonkActuator("HonkActuator"));
34                     break;
35                 default:
36                     Console.WriteLine("Invalid robot type.");
37                     break;
38             }
39         }
40
41         public void Move()
42         {
43             Console.WriteLine($"{Name} is moving...\n");
44         }
45
46         public void Activate()
47         {
48             Console.WriteLine($"Sensor data for {Name}:");
49             foreach (var sensor in _sensors)
50             {
51                 sensor.Measure();
52             }
53         }
54 }
```

```

C# Program.cs
1  using OopConcepts;
2
3  Console.WriteLine("Enter the name of the first robot:");
4  string name = Console.ReadLine() ?? "RobotOne";
5
6  Console.WriteLine("Enter the type of the first robot (1 = wheeled, 2 = drone):");
7  int type = int.TryParse(Console.ReadLine(), out int result1) ? result1 : 1;
8
9  WheeledRobot robot1 = new WheeledRobot(name, type);
10
11  Console.WriteLine("Enter the name of the second robot:");
12  name = Console.ReadLine() ?? "RobotTwo";
13
14  Console.WriteLine("Enter the type of the second robot (1 = wheeled, 2 = drone):");
15  type = int.TryParse(Console.ReadLine(), out int result2) ? result2 : 2;
16
17  WheeledRobot robot2 = new WheeledRobot(name, type);
18
19  robot1.Move();
20  robot1.Activate();
21
22  robot2.Move();
23  robot2.Activate();
24 }
```

Tot slot heb ik de gebruiker een keuze voor naam en type gegeven en daarna de 2 robots aangemaakt en werkend gebracht door de juiste method aan te roepen.

## Opdracht 3: MQTT in practice - Proof of Concept

Deze opdracht en les waren een introductie tot MQTT. Hier laat ik de PoC(Proof of Concept) zien

Ik begon natuurlijk met het importeren van de “SimpleMqttClient.cs” en het downloaden van de nuget package die hierbij hoorde. Het enige wat ik in deze file aangepast had was de “CreateSimpleMqttClientForHiveMQ”, hier heb ik mijn gegevens gezet voor het publishen en lezen van messages. Dat beantwoord ook meteen de vraag over wat deze methode doet, het stelt een nieuwe Client op en dus een connectie met de site.

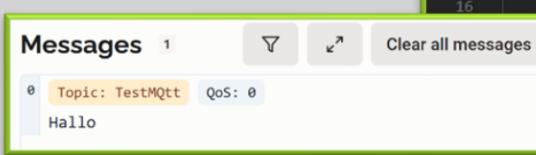
```
C# Program.cs X C# SimpleMqttClient.cs

C# Program.cs
1  using SimpleMqtt;
2  using mqtt.models;
3
4  Console.Clear();
5  // wat doet dit precies? noteer in groeidocument
6  var client = SimpleMqttClient.CreateSimpleMqttClientForHiveMQ("client-id");
7
8  var receive = new MqttReceive(client);
9  receive.Receive();
10
11 string topic = "TestMQTT";
12 await client.SubscribeToTopic(topic);
13 Console.WriteLine($"Subscribed to topic: {topic}\n");
14
15 string text = string.Empty;
16 var publish = new MqttPublish(client)
17 {
18     Topic = topic,
19     Message = text
20 };
21
22 while (true)
23 {
24     Console.WriteLine("Your text:");
25     text = Console.ReadLine() ?? string.Empty;
26     publish.Message = text;
27
28     publish.Publish();
29
30     Console.WriteLine("Press");
31     if (Console.ReadKey().Key == ConsoleKey.Escape)
32     {
33         break;
34     }
35     Console.ReadKey();
36 }
```

Ik heb de code voor het recieven en publishen beide in een class gezet en daarbij heb ik ook alles in een loop gezet zodat het blijft herhalen als een soort chat-bot.

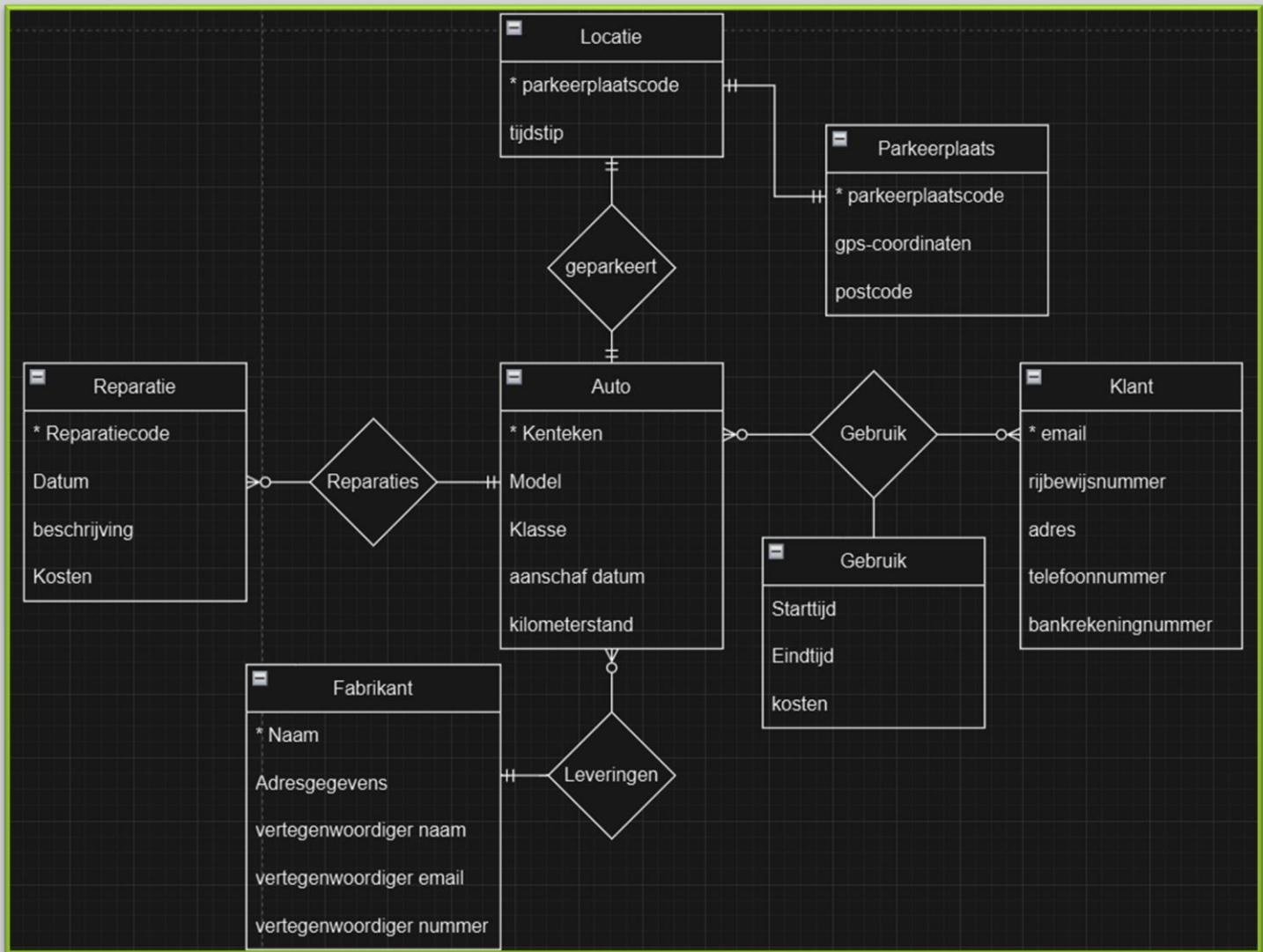
Tijdens de les heb ik ook geprobeerd om berichten naar klasgenoten te sturen en ook dit is gelukt.

```
models > C# receive.cs ...
1  using SimpleMqtt;
2
3  namespace mqtt.models
4  {
4.1      2 references
5      public class MqttReceive
6      {
6.1          1 reference
7          public MqttReceive(SimpleMqttClient client)
8          {
9              Client = client;
10
11             0 references
12             public string? Topic { get; set; }
13             0 references
14             public string? Message { get; set; }
15             2 references
16             public SimpleMqttClient Client { get; set; }
17
18             1 reference
19             public void Receive()
20             {
21                 // Subscribe to the OnMessageReceived event
22                 Client.OnMessageReceived += (sender, args) =>
23                 {
24                     Console.WriteLine($"Bericht ontvangen: topic={args.Topic}, message={args.Message}.\n");
25                 };
26             }
27         }
28     }
29 }
```



## Opdracht 4: Software ontwikkelen - Databases 1: Modelleren

Na de datamodelleren les van deze week hebben we een casus gekregen om met de gekregen stof te oefenen, hieronder is mijn uitwerking te zien:



Tot slot heb ik deze opdracht besproken met een student en die had geen verdere toevoegingen aan deze ERD.

## Opdracht 5: UX Opdracht: Card sorting

Helaas heb ik deze les niet kunnen volgen door een afspraak bij de huisarts. Ik heb de stof doorgenomen en gekeken naar de opdracht, ik heb kaarten bedacht en opgeschreven in draw.io, deze heb ik aan een klasgenoot gegeven en gesorteerd.

Dit is niet de volledige opdracht maar dit is wel het beste wat ik er van kon maken met de mogelijkheden die ik had na de les niet te kunnen volgen.

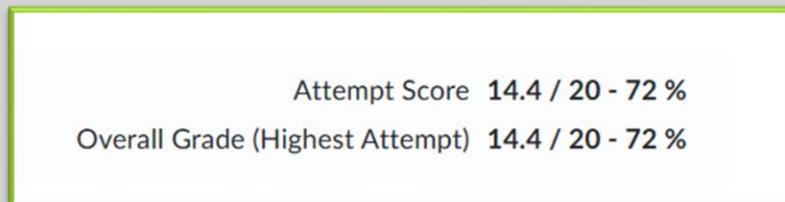
persoons gegevens	prive-gegevens/ info	medicijn informatie	tekst	robot info	maaltijd en drink momenten	contact
persoonlijke informatie	inlog gegevens	voorraad	titel	opladen	ingredienten	contact info
naam	wachtwoord	medicatie niet ingenomen	uiteleg	hulp knop	ontbijt	contact zorg-medewerker
leeftijd	instellingen	zorg achtergrond	meldingen	accu	lunch	
		vaste tijdstippen		ledlamp	avond	
				buzzer	snacks	
				knoppen	water	

## Opdracht 6: Basisvaardigheden Presenteren: Tussenopdracht 0 en 1, stemgebruik en zelfverzekerheid

Bij deze opdracht heb ik voorbereid op presenteren en keek ik hierna op terug wat wel en niet goed ging.

### Tussenopdracht 0:

Voor deze les heb ik de quiz: "Hoe goed kan jij al presenteren?" gemaakt, hieruit kwam deze score:



Ik val daardoor onder:

*"13-17 punten - Je beheerst veel aspecten van presenteren goed, maar er zijn nog enkele verbeterpunten. Overweeg om meer aandacht te besteden aan tempo, oogcontact of voorbereiding."*

Dit klopt inderdaad. Ik vind dat meer tijd besteden aan voorbereiding mij goed zou helpen en ook het tempo heeft meer aandacht nodig, soms praat ik te snel.

### Tussenopdracht 1:

- Liked: Ik heb een vrij lude stem, en ik praat ook altijd duidelijk. Dit was een positief ding omdat zelfs de mensen achterin mij even hard konden horen als de mensen voorin. Ook was het ondanks weinig voorbereiding een goede vorm van improviseren.
- Learned: Beter voorbereiden, en als er alsnog niet voorbereid is zorgen dat je creatief hier mee omgaat.
- Lacked: Halverwege de presentatie zei ik vaak "uhm" of viel ik even stil, ook leek ik meer zenuwachtig.
- Longed for: Volgende keren wil ik beter voorbereiden, ik weet dat met genoeg voorbereiding en eventueel een PowerPoint met keypoint ik veel verder kom en beter informatie kan overbrengen.

*Feedback van medestudenten na mijn presentatie:*

"Tip: Je zei vaak uuhm, Top: duidelijk praten"

"Tip: iets meer je handen gebruiken bij het praten, Top: Duidelijk te verstaan en je bleef kalm praten zonder te haasten"

"Tip: je hebt soms een gesloten houding, Top: je stemgeluid is goed en duidelijk"

*Feedback van docent na mijn presentatie:*

"Top: je spreekt erg rustig en kijkt goed rond. Tip: waar blijven je handen?"

## Opdracht 7: Basisvaardigheden Systeemdenken: Opdracht 2 Systeemijsberg en Solutionary Scale

In de les systeemdenken hebben we het over verschillende onderwerpen gehad, waaronder de systeemijsberg. In deze opdracht ga ik een systeemijsberg maken en deze verder uitbreiden op basis van mijn gekozen casus.

### **Het probleem:**

Mensen met dementie in de vroege stadia vergeten vaak te eten, drinken en medicijnen in te nemen, wat kan leiden tot gezondheidsproblemen zoals ondervoeding en uitdroging.

### **Problem Statement:**

Hoe kunnen robots worden ingezet om mensen met dementie te ondersteunen bij op tijd en gezond eten en drinken en/of medicijnen in te nemen, zodat zij langer zelfstandig thuis kunnen blijven wonen?

### **Systeemijsberg:**

#### *Gebeurtenissen,*

- Mensen met dementie vergeten te eten en drinken.
- Mantelzorgers moeten constant controleren of er voldoende wordt gegeten en gedronken.
- Geen of teveel medicijn innames.
- Gezondheidsproblemen zoals bijvoorbeeld verminderde energie, uitdroging, en verzwakte weerstand.
- Zorgverleners hebben minder tijd voor persoonlijke aandacht door personeelstekort.

#### *Patronen,*

- Dagelijks vergeten ouderen met dementie basiszorgactiviteiten zoals eten, drinken en medicijnen.
- Mantelzorgers en familie ervaren een hoge zorgdruk.
- Zorgverleners hebben moeite om toezicht te houden door een groeiende zorgvraag.
- Mensen met dementie raken gefrustreerd of angstig door het verlies van zelfstandigheid.

#### *Systemen en Structuren,*

- Er is geen automatische ondersteuning in huis die herinnert aan eten en drinken of medicijnen.
- Huizen van ouderen zijn vaak niet aangepast aan bepaalde beperkingen.
- De zorg is overbelast door een tekort aan zorgverleners.
- Beperkte kennis van technologie bij ouderen en mantelzorgers maakt de overstap lastiger.

*Mentale Modellen,*

- De gedachte dat technologie te ingewikkeld is voor ouderen.
- Mantelzorgers voelen zich vaak verantwoordelijk en denken dat zij alles zelf moeten doen.
- De gedachte bij traditionele zorg dat deze beter werkt dan bij technologische ondersteuning.
- Het idee dat ouderen met dementie niet zelfstandig kunnen blijven wonen.

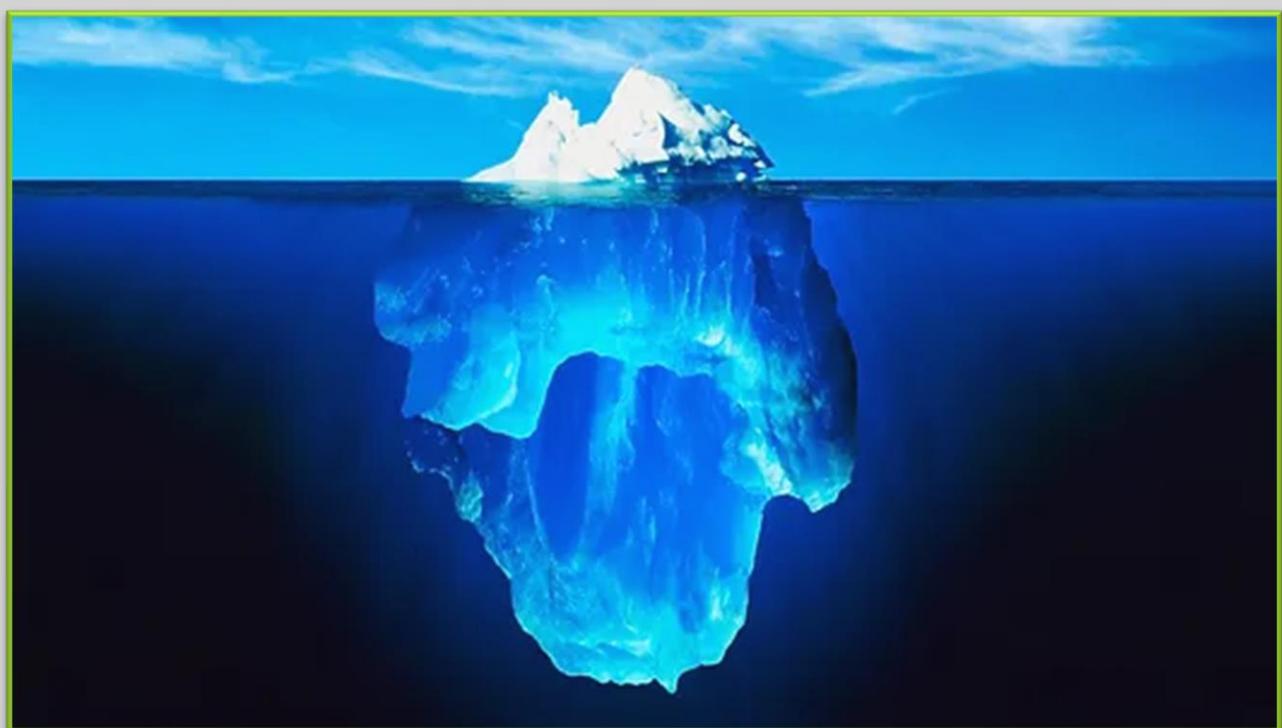
*Oplossing met robottechnologie*

- Een robot die zelfstandig door het huis navigeert en de gebruiker vriendelijk herinnert om te eten en drinken of medicijnen in te nemen.
- Een gekoppeld dashboard voor mantelzorgers dat de data van de robot als gezondheidsgegevens overzichtelijk weergeeft.
- De robot biedt persoonlijke interactie, wat het welzijn van de gebruiker verhoogt.
- De technologie is eenvoudig in gebruik.

Deze oplossing pakt zowel de symptomen als onderliggende patronen aan en kan bijdragen aan het vergroten van zelfstandigheid en het verminderen van zorgdruk.

*Feedback van medestudent:*

Er is goed en uitgebreid nagedacht over alle punten en het staat duidelijk opgeschreven onder elkaar.



## **Loopbaan opdracht: Reflecteren**

In deze opdracht moesten we reflecteren op een situatie, er stond in de opdracht dat je dit ook moest verwerken in je “ontwikkelplan” dus heb het hier alsnog toegevoegd en ook op de juiste plek ingeleverd.

### **Geef een beschrijving van de situatie.**

Ik heb weinig moeite met programmeer gerelateerde opdrachten, dit zorgt er voor dat ik eigenlijk te weinig vragen stel en niet genoeg feedback ontvang op gemaakte opdrachten. Nu was dit in het begin niet een groot probleem maar omdat we nu bezig zijn met een groeidocument waar juist die feedback belangrijk is vind ik het ondanks dat ik het belang er van in zie wel lastig om vragen te stellen/formuleren naar dingen waar ik eigenlijk geen moeite mee heb. Ik weet dat ik fouten maak en ik zie mijzelf zeker niet als iemand die alles weet of alles kan, zo zou ik ook niet willen overkomen alleen dit zorgt er voor dat ik gewoon te weinig terug reflecteer bij opdrachten.

### **Beschrijf gevoelens en gedachten die je bij de situatie had. Weet je waardoor die werden veroorzaakt?**

De situatie is ontstaan door meerdere factoren, vooral de voorkennis speelt een grote rol.

### **Wat ging er goed in deze situatie?**

Alles is snel en op tijd af, vaak al in de les zelf omdat ik tijdens een uitleg vooruit werk.

### **Wat had er beter gekund?**

Het vragen van feedback, kijken naar wat beter kan ook als het al goed was.

### **Wat was precies jouw inbreng in de situatie? Wat was je gedrag?**

Mijn gedrag bestaat uit terughoudendheid, bij alles denk ik “dat lukt mij wel” waardoor ik nooit echt in gesprek ga over de opdrachten die ik oplever.

### **Wat had je zelf (anders) kunnen doen om de situatie beter te laten verlopen?**

k vind het vervelend om telkens aan te geven dat ik dingen “al snap” maar toch moet ik zorgen dat ik die gedachte links laat liggen en echt feedback ga vragen.

### **Wat neem je jezelf voor om in toekomstige situaties te doen?**

Bij bepaalde grotere opdrachten altijd vragen om feedback, vooral als het een opdracht uit het groeidocument betreft, als er niks fout is kan ik bijvoorbeeld ook vragen naar wat er beter kan of wat een andere manier van aanpak zou kunnen zijn.

### **Reflectie niveau:**

Zelf zou ik zeggen dat ik op een niveau van 5 of 6 zit, ik denk veel na over hoe mijn eigen gedrag mij beïnvloed en analyseer mezelf vanaf hoe andere het zouden zien. Als ik op 5 zit zou ik heel graag 6 bereiken door ook daadwerkelijk wat te doen met deze reflectie en mijn eigen belangen opzij zetten.

## Week 3 reflectie

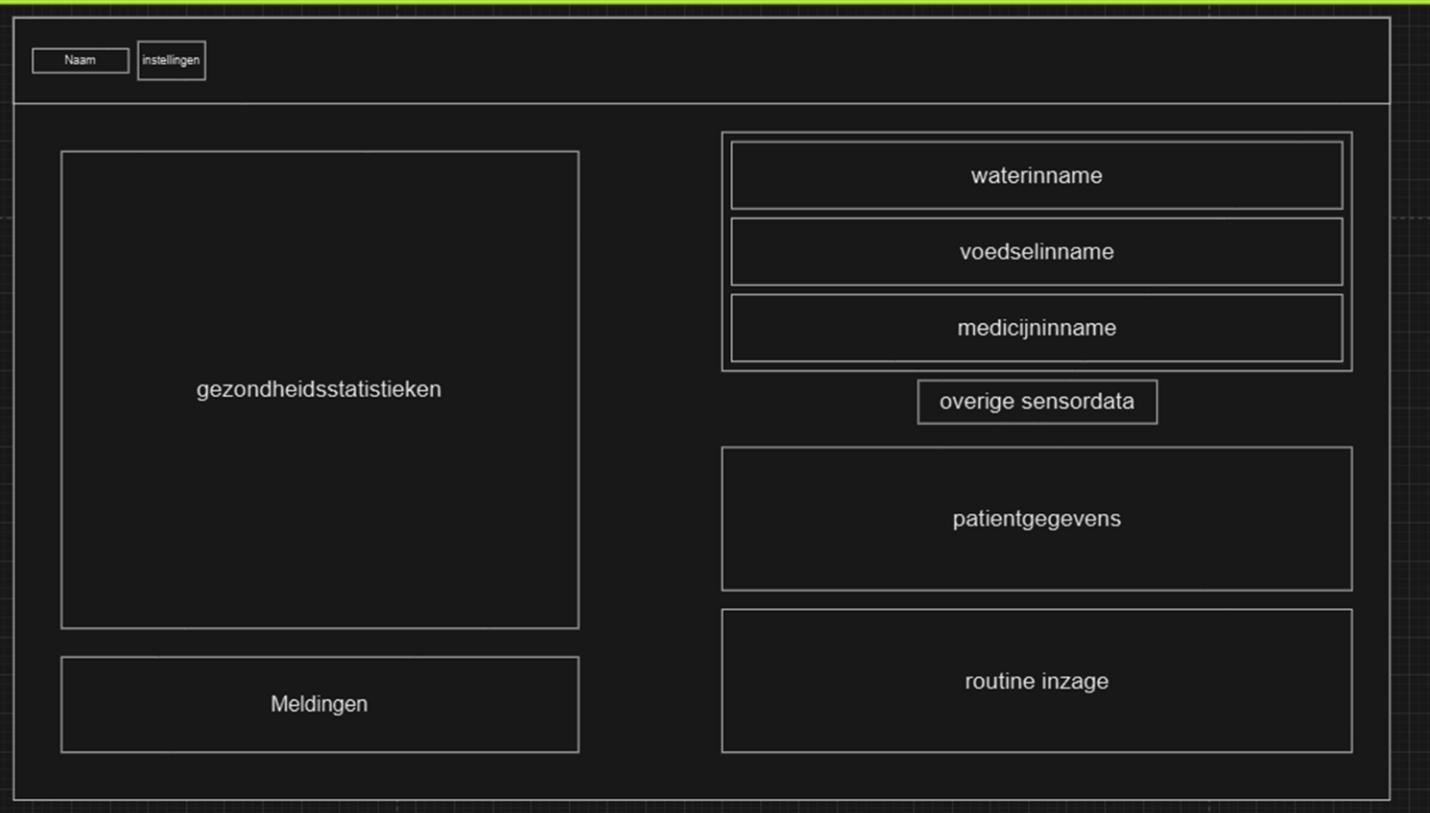
Deze week ging goed. Helaas kon ik niet bij een UX les zijn en heb ik dus deze opdracht zo moeten maken met wat ik kon doen, het missen was in dit geval niet te vermijden maar alsnog is dit jammer. Na het maken van mijn class diagram vroeg ik feedback aan een klasgenoot en toen we er samen naar keken kwam ik er achter dat ik het helemaal verkeerd had gemaakt, ik was in de war met andere concepten. Ik heb het hier dus op aangepast en opnieuw in mijn document gezet.

Naast dat ging de rest goed deze week.

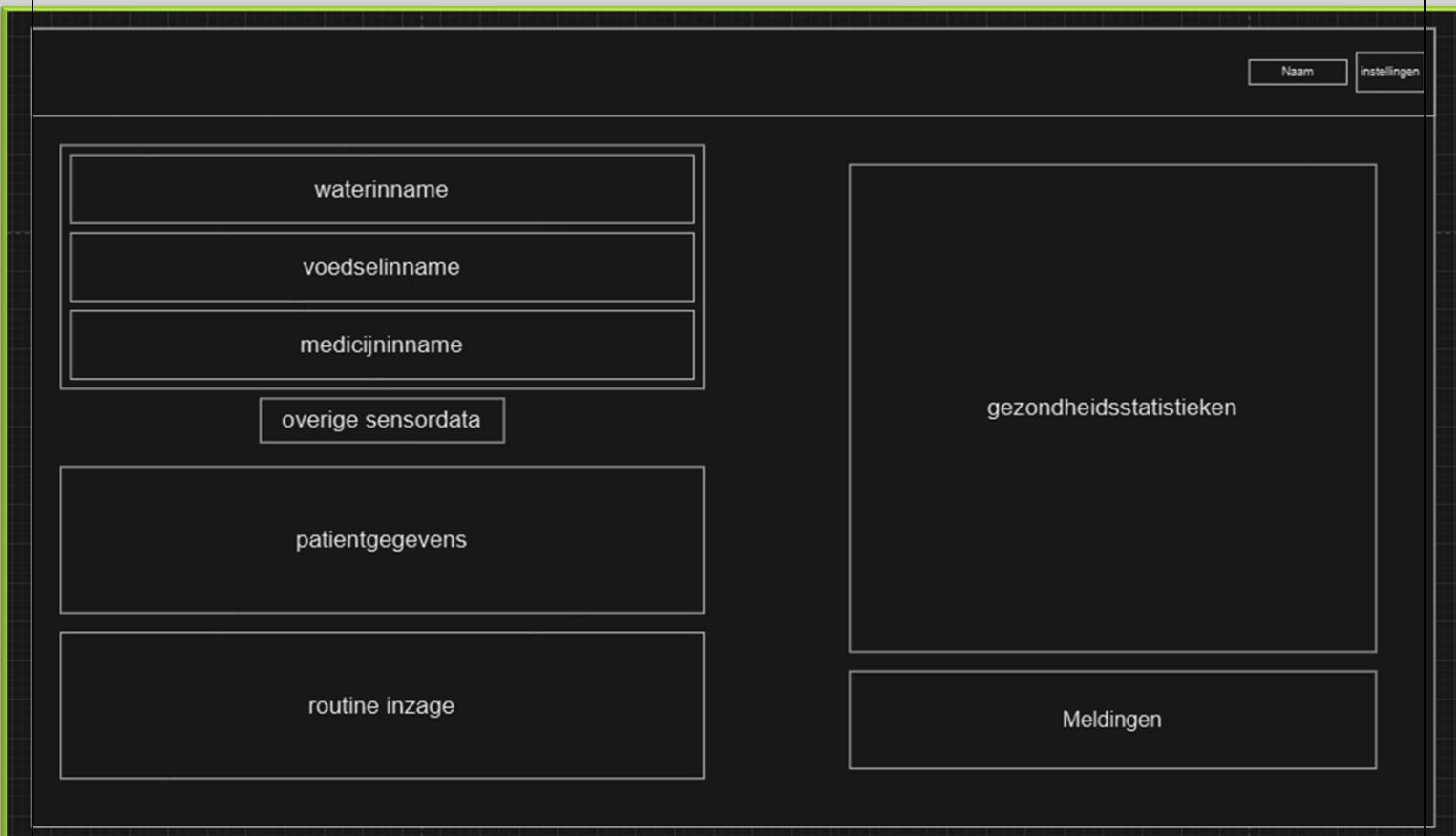
## Week 4

### Opdracht 1: UX: Functioneel ontwerp (Wireframes)

Hier was het de bedoeling om 1 wireframe te maken, en daarbij 2 varianten. In totaal dus 3 verschillende Wireframes van dezelfde pagina.



Bij het maken van deze Wireframes heb ik nagedacht over wat handig is, zo vond ik het belangrijk dat de gezondheidsstatistieken overal groot waren, zodat dat snel te zien was, de rest heb ik verschillende groottes en posities gegeven.

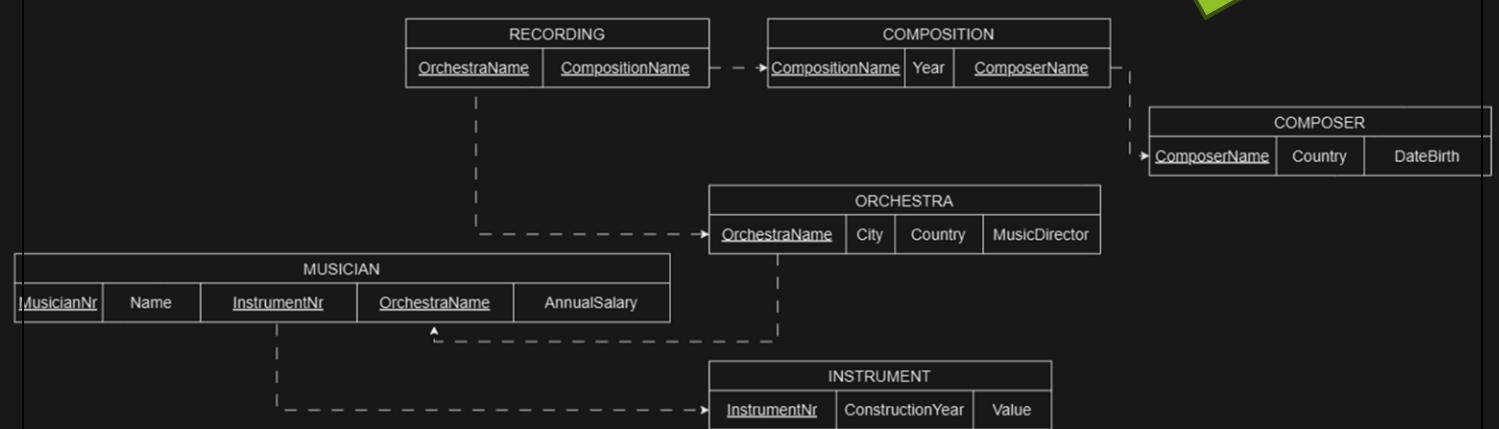
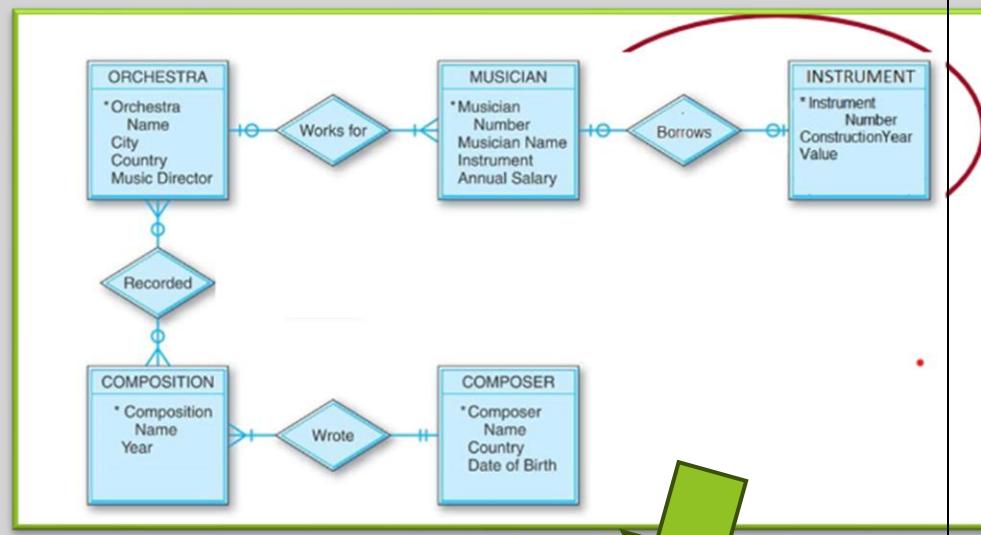


Na het maken van alle 3 de Wireframes heb ik feedback gevraagd aan meerdere studenten, naast dat ze de Wireframes goedvonden hadden ze een voorkeur aan de eerste of de 3<sup>de</sup>, omdat ze het fijner vonden dat de statistieken of het middelpunt was of rechts stond.

## Opdracht 2: Software ontwikkelen: Databases 2: Relationale Database Ontwerp

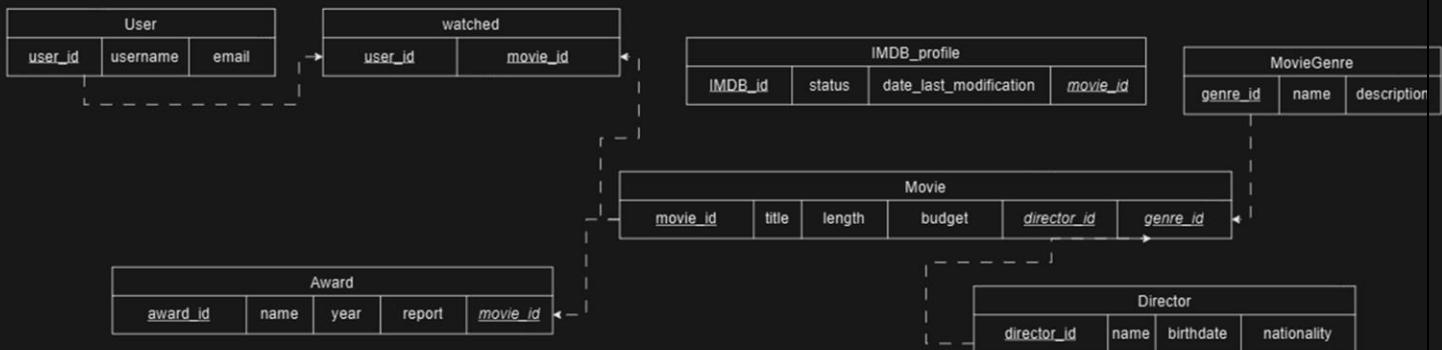
In deze opdracht was het de bedoeling om vanaf een gekregen ERD een relationeel database ontwerp te maken, en daarna hier een SQL-script voor te schrijven die de tabellen dus in de database toevoegt.

Als eerst had ik geoefend met het maken van een relationeel databaseontwerp



Toen ik dit ontwerp gemaakt had in de les vroeg ik meteen aan mijn docent voor eventuele feedback of verbeterpunten. Als feedback kreeg ik dat de “foreign keys” niet onderstreept hoefde te zijn, maar dit wist hij niet zeker. Ik wou dit eerst verwerken in mijn volgende ontwerp maar ik vond met een aanduiding het toch duidelijker dan zonder, ik heb het dus onderstreept en schuingedrukt gemaakt om toch het verschil te zien maar wel aan te geven dat het een foreign key is.

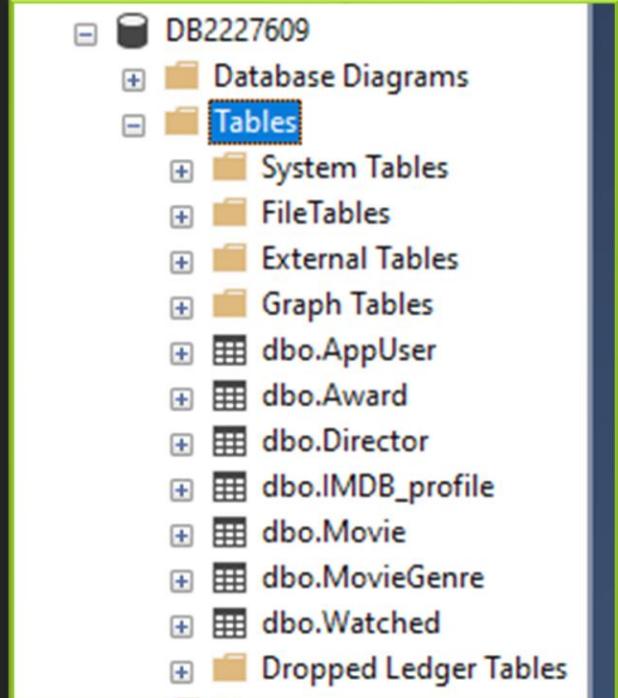
Uiteindelijk had ik dit ontwerp gemaakt voor de gekregen ERD in de opdracht:



Hierna heb ik een script geschreven die de tabellen aanmaakt, ik heb hier gebruik gemaakt van een variatie van een “auto increment” dit zorgt ervoor dat als er nieuwe data bijkomt hij een nieuw ID aanmaakt, zo blijft het uniek. Na het uitvoeren waren de tabellen correct aangemaakt

```

Assets > Sql > Movies.sql
1 -- ('User' vervangen voor 'AppUser' om SQL Server's conflicts te vermijden)
2 CREATE TABLE AppUser (
3     user_id INT IDENTITY(1,1) PRIMARY KEY, -- IDENTITY(1,1) zorgt ervoor dat de user_id automatisch wordt ingevuld, zelfde als AUTO_INCREMENT in MySQL
4     username VARCHAR(50) NOT NULL,
5     email VARCHAR(100) NOT NULL
6 );
7 CREATE TABLE MovieGenre (
8     genre_id INT IDENTITY(1,1) PRIMARY KEY,
9     name VARCHAR(50) NOT NULL,
10    description TEXT
11 );
12 CREATE TABLE Director (
13     director_id INT IDENTITY(1,1) PRIMARY KEY,
14     name VARCHAR(100) NOT NULL,
15     birthdate DATE,
16     nationality VARCHAR(50)
17 );
18 CREATE TABLE Movie (
19     movie_id INT IDENTITY(1,1) PRIMARY KEY,
20     title VARCHAR(100) NOT NULL,
21     length INT,
22     budget DECIMAL(15, 2),
23     director_id INT,
24     genre_id INT,
25     FOREIGN KEY (director_id) REFERENCES Director(director_id),
26     FOREIGN KEY (genre_id) REFERENCES MovieGenre(genre_id)
27 );
28 CREATE TABLE Watched (
29     user_id INT,
30     movie_id INT,
31     PRIMARY KEY (user_id, movie_id),
32     FOREIGN KEY (user_id) REFERENCES AppUser(user_id),
33     FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
34 );
35 CREATE TABLE IMDB_profile (
36     IMDB_id INT IDENTITY(1,1) PRIMARY KEY,
37     status VARCHAR(20),
38     date_last_modification DATE,
39     movie_id INT,
40     FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
41 );
42 CREATE TABLE Award (
43     award_id INT IDENTITY(1,1) PRIMARY KEY,
44     name VARCHAR(100) NOT NULL,
45     year INT,
46     report TEXT,
47     movie_id INT,
48     FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
49 );
  
```



## Week 4 reflectie

Deze week ging snel voorbij, we hadden weinig les en daarom waarschijnlijk ook minder opdrachten dan de week ervoor. Het database ontwerp ging goed, na wat feedback te vragen over hoe mijn “oefen”-ontwerp was heb ik de opdrachten af kunnen ronden.

## Week 5

### Opdracht 1: UX Opdracht: Testing 1

Deze opdracht is een voorbereiding voor de UX-les van week 6! Ik heb voor deze opdracht de stof goed doorgenomen en de Heuristics vragenlijst gedownload en onder eigen naam klaarstaan, zo kan ik die gebruiken tijdens het testen in de les.

Daarnaast moest ik ook een lijst maken met bijvoorbeeld 10 taken die de gebruiker kan uitvoeren tijdens het testen. Het is de bedoeling dat ik genoeg taken heb om alle heuristics mee te controleren.

Usability heuristics vragenlijst			
1	Weten gebruikers altijd wat het systeem aan het doen is en waar zij zich bevinden?		
	geheel oneens	neutraal	geheel eens
	toelichting		
2	Gebruikt het systeem termen die de gebruiker begrijpt?		
	geheel oneens	neutraal	geheel eens
	toelichting		
3	Kunnen gebruikers gemakkelijk fouten herstellen, bijvoorbeeld door een “Ongedaan maken” of “Terug” optie?		
	geheel oneens	neutraal	geheel eens

(Het bestand wat we moesten downloaden)

De taken:

- “Open de pagina en identificeer of duidelijk is waar je je in de applicatie bevindt.”
- “Bekijk de weergegeven data van een sensor. Als iets onduidelijk is, gebruik de knop ‘Meer info’ om meer te weten te komen.”
- “hij klik op verwijderen en krijg een melding of je het zeker weet.”
- “Gebruik de navigatiebalk om naar de pagina ‘sensordata’ te gaan en controleer of het duidelijk is hoe je bij de instellingen-pagina komt.”
- “Open de datavisualisatie pagina en bepaal of je in één oogopslag de status van de sensoren kunt zien.”
- “Gebruik sneltoetsen om een sensoroverzicht te openen en vervolgens (eventueel) een rapport te exporteren”
- “Naveer door de applicatie en bepaal of je alleen de relevante informatie ziet, zonder afgeleid te worden door overbodige details.”

- “Zoek naar een manier om hulp te krijgen en vul het contactformulier in met een vraag over de werking van een sensor.”
- “Probeer een fout te maken, zoals het invoeren van een ongeldig datumformaat. Bekijk de foutmelding en volg de instructies om dit op te lossen.”
- “Controleer of de knoppen en termen op de pagina's van het programma consistent zijn. Bijvoorbeeld, is de knop 'Verwijderen' altijd op dezelfde plek en hetzelfde vormgegeven?”

Daarnaast heb ik de stappen alvast doorgenomen voor de les van volgende week.

Ik heb feedback gevraagd op mijn gemaakte taken en een soort mini-test gegeven aan een mede student, deze gaf aan dat het duidelijk verwoorde taken waren en je makkelijk kon checken of dingen wel of niet voldoen aan bepaalde eisen door deze ook echt uit te voeren op de bijbehorende site.

## Week 5 reflectie

Deze week hadden we heel weinig les, ik heb veel voortgang gemaakt met mijn robot en de bij behorende app en daarnaast heb ik ook veel uitgeprobeerd met MQTT en andere stof die we gekregen hadden. Deze week ging goed en niet veel dingen gingen fout waar ik echt goed op kon reflecteren, wel had ik wat problemen met het connecten van de robot maar dit heb ik met een medestudent opgelost.

## Week 6+

Hier reflecteer ik op de laatste weken van het project, omdat we hier eigenlijk niet echt lessen en nieuwe stof hadden heb ik het zo opgelost.

## Week 6+ reflectie

Deze week voor de vakantie veel thuis gewerkt, verder aan de robot en niet ergens tegen aan gelopen. Ook was ik veel bezig met het tentamen, na deze gemaakt te hebben begon ik echt met het grootste programmeer werk voor de robot en heb ik deze uiteindelijk afgekregen, naast wat moeite met dat ik mijn oudere wireframe niet helemaal passend meer vond heb ik opnieuw een wireframe gemaakt en deze toegepast op mijn site. Ook voor mijn klassendiagram was dit nodig, ik had ook wat database toevoegingen maar deze waren zo klein dat ik een nieuw dataontwerp hier niet bij nodig vond. Uiteindelijk afgekregen met alle eisen en veel extra functionaliteiten.

Voor het afronden van de robot had ik ook nog extra sensoren aangevraagd voor de kerstvakantie nog. Ik was vergeten dit aan te geven in het document dus vandaar dat ik dat op deze manier hier doe. Ik heb uiteindelijk gekozen voor: 2 LED buttons, 1 light sensor, lcd scherm, ultrasonic sensor(afstand) en een speaker.

Op de speaker na heb ik deze allemaal kunnen gebruiken, de speaker had toch niet de dingen die ik dacht dat het zou hebben en heb er daarom toch maar voor gekozen deze niet te gebruiken, ik zorg bij de robot voor data die getoond kan worden op het scherm. Zoals bepaalde taken en herinneringen, dit kan bestuurd worden door 1 led button, de andere led button word gebruikt voor het rij en standby systeem van de robot. Ook stuurt ik de lichtsensor data en de bevestigde taken door via mqtt naar de site en deze verwerkt de data en zet het in een database. Mijn robot kan door de afstand sensor rond blijven

rijden zonder botsing te voorkomen en het voldoet ook aan de andere eisen en nog extra dingen die ik heb toegevoegd om het nog beter te laten werken.

Om het uit te leggen: De robot start, als deze gestart is staat hij stil. Ik heb hier voor gekozen omdat ik niet denk dat elke patient continu een robot wil zien rijden, rust kan ook fijn zijn soms! Dus hij start in standby modus, in deze modus doet hij gewoon nog alles alleen dan zonder te rijden, de gebruiker kan er voor kiezen om op de rode knop te drukken en de robot alsnog te laten rijden, op elk moment kan dit ook ongedaan worden door er nog een keer op te drukken en de robot weer in standby modus te zetten.

Daarnaast haalt hij tasks uit een database op, deze laat hij zien op het scherm. Als de gebruiker op de blauwe knop drukt en deze bevestigd word het via mqtt naar de site gestuurd en opgeslagen. Op de site kan je ook de lux-waardes zien, het batterijpercentage, alle taken met daarbij de tijd en datum van opname en je kan hier taken aanmaken die in de database komen zodat de robot deze op het scherm kan weergeven! Naast wat andere functionaliteiten die de robot heeft zoals een noodstop via de site is dit de grote lijn van wat de robot doet. Perfect dus om bijvoorbeeld op een tafel of kastje te zetten als herinneringsrobot of gewoon om lekker rond te laten rijden!

Natuurlijk kan ik hier veel meer over vertellen maar daar is dit document niet voor, wel vond ik het fijn om ergens gedocumenteerd te hebben hoe de robot is geworden.

Met hulp van de eerste Testing opdracht heb ik veel tests gedaan voor mijn robot, ik heb taken opgesteld die ik kon testen en deze allemaal goed gecheckt en eventuele aanpassingen gemaakt of opgeschreven, mijn applicatie is op het moment van schrijven door alle tests heen en alle fouten zijn er uit gehaald! Voor de zekerheid ga ik nog een paar keer langs alle tests meerdere keren, wie weet komt er een specifiek punt die wel een eventuele error zou geven en dit wil ik voorkomen. Ik heb gebruik gemaakt van de testcases van de opdracht maar ook eigen testcases opgeschreven om deze te testen.