

An underwater photograph showing a large number of dead fish lying on a sandy seabed. The fish are mostly silver and appear to be of the same species. The water is dark blue, and sunlight filters down from the surface, creating a bright, hazy area at the top. The overall mood is somber and concerning.

# **Produktrapport**

## **Iltsvind**

Mathias Wriedt Kamp

Marius Martin Møller

## Titelblad

### Iltsvind produktrapport

Uddannelse: Datateknikker med speciale i programmering  
Hovedforløb: 6. Hovedforløb  
Titel på projektet: Iltsvind  
Projektperiode: Fra mandag d. 13/11/2023 til fredag d. 15/12/2023  
Vejledere: Kris Kristensen - faglærer ZBC Ringsted  
Udarbejdet af: Mathias Wriedt Kamp, Marius Martin Møller

---

Mathias Wriedt Kamp

---

Marius Martin Møller

---

Kris Kristensen

Antal normalsider: xx

Afleveringsdato: 07/12/2023

Mathias Wriedt Kamp

Marius Martin Møller

## Indholdsfortegnelse

Titelblad .....	2
Kravspecifikation .....	4
Introduktion .....	4
Formål med kravspecifikationen .....	4
Definitioner, akronymer og forkortelser .....	4
System .....	5
Funktionalitet .....	6
Usability .....	13
Performance .....	13
Svartid .....	13
Opstarts- og nedlukningstid .....	13
Design Constraints .....	13
Interfaces .....	13
Software interfaces .....	13
Use-case diagram .....	14
Produktblade .....	15
Oxygenmåler .....	15
Fugleskræmsel .....	16
Muslingebur .....	17
Teknisk produkt dokumentation .....	18
Deployment diagram .....	18
Arduino – Circuit diagrams .....	19
Arduino muslingebur – circuit diagram .....	19
Iltmåler circuit diagram .....	20
Fugleskræmsel circuit diagram .....	21
Arduino system – circuit diagram .....	22
System – flowdiagram .....	23
Iltmåler flowdiagram .....	24
Fugleskræmsel flowdiagram .....	25
SD-diagrammer .....	26
Klasse diagram .....	26
Er diagram .....	28
Topologi .....	29

Mathias Wriedt Kamp

Marius Martin Møller

Testrapport .....	30
Udførte unittests .....	30
Udførte integrationstests .....	33
Test Ikke-Funktionelle Krav .....	35
Testperson 1 resultat use-case id 1 .....	41
Referencer .....	42
Bilag.....	42

## Kravspecifikation

### Introduktion

Den aktuelle tilstand af iltmængden i de danske farvande er nu på et bekymringsvækkende niveau, markeret som det værste set i de seneste 20 år. Aarhus Universitet har gennemført adskillige undersøgelser for at identificere løsninger på denne udfordring. En af deres banebrydende undersøgelser fokuserer på anvendelsen af blåmuslinger som et potentielt redskab til at filtrere vandet og forbedre dets renhed som resulterer i at solens stråler har nemmere ved at nå havbunden og algerne kan benytte fotosyntese til at generere ilt.

### Formål med kravspecifikationen

Denne kravspecifikation har til formål at udforske og definere de nødvendige skridt og krav for implementeringen af en "Proof of Concept" løsning. Specifikationen er struktureret i henhold til FURPS-modellen (Functionality, Usability, Reliability, Performance, Security). Da fokus er på krav og ikke-funktionelle krav, er der foretaget en nøje udvælgelse, og visse punkter er blevet udeladt.

### Definitioner, akronymer og forkortelser

**Fully-dressed:** en use-case med veldefineret forløbsbeskrivelse.

**Muslinger:** referer specifikt til blåmuslinger

**Muslingebur:** Automatisk bur til opbevaring af blå muslinger, buret har en automatisk tippe og åbne funktion.

**Fugleskræmsel:** Et automatiseret fugleskræmsel bestående af 4 servo motorer.

Mathias Wriedt Kamp

Marius Martin Møller

## System

Dette system består af flere delelementer som er forbundet sammen.

### Iltmåler:

- Ansvarlig for at måle iltindholdet i vandet.
- Præsenterer målte iltværdier på et display, der giver brugere og operatører mulighed for løbende at overvåge iltindholdet.

### Automatisk muslingebur:

- Ansvarlig for at opbevare muslinger.
- Modtager besked fra iltmålerkomponentet om at åbne eller lukke buret afhængigt af den målte iltværdi.

### kommunikationsmodul:

- Ansvarlig for at sende http-request til Web Api.
- Modtager målinger fra iltmåleren.

### Web Api:

- Grænseflade imellem kommunikationsmodulet og databasen.

### Fugleskræmsel:

- Introducerer et diffust fugleskræmsel, der aktiveres, når der tilføjes flere muslinger til vandet, hvilket effektivt afskrækker edderfugle og beskytter muslingerne.

### Database:

- Er ansvarlig for at gemme data.
- Er tilgængelig igennem Web Api grænsefladen.

Dette integrerede system muliggør en automatiseret og effektiv tilgang til at kunne overvåge og forbedre iltindholdet.

Mathias Wriedt Kamp

Marius Martin Møller

## Funktionalitet

<b>Use Case navn</b>	<b>Aflæsning af iltmængde</b>
<b>Id</b>	1
<b>Version</b>	1.0
<b>Beskrivelse</b>	Denne use case beskriver, hvordan en bruger kan aflæse iltmængden i vand ved hjælp af oxygenmåleren. Oxygenmåleren består af en Arduino Uno, med en tilsluttet iltmåler og et display, der viser iltmængden i mg/L. Brugeren ønsker at aflæse disse oplysninger for at vurdere iltindholdet i vandet.
<b>Problemstillingen</b>	En bruger skal have mulighed for at kunne aflæse iltindholdet i noget vand.
<b>Scope</b>	Iltmåleren bestående af Arduino Uno, iltmåler og display.
<b>Aktør(er)</b>	Bruger: Autoriseret person, der ønsker at aflæse iltmængden i vandet.
<b>Stakeholder og Interesser</b>	Bruger: Ønsker at få nøjagtige og pålidelige oplysninger om iltmængden i vandet for at vurdere miljøets tilstand.
<b>Prækonditioner</b>	<ul style="list-style-type: none"><li>• Iltmåleren er korrekt monteret og nedsænket i vandet.</li><li>• iltmåleren er korrekt kalibreret i henhold til producentens specifikationer.</li><li>• Strømforsyningen er tilsluttet.</li></ul>
<b>Postkonditioner</b>	Den målte iltmængde kan aflæses fra displayet.
<b>Success forløb</b>	<ol style="list-style-type: none"><li>1. Brugeren tænder iltmåleren.</li><li>2. Brugeren placerer iltmåleren i vandet og sikre at iltmålerens probet er dækket af vand.</li><li>3. Displayet viser iltmængden i mg/L</li><li>4. Brugeren aflæser iltmængden og vurderer vandkvaliteten.</li></ol>
<b>Alternativt forløb</b>	1.1 (alt 1) Iltmåleren er ikke tændt korrekt.

Mathias Wriedt Kamp

Marius Martin Møller

	<ul style="list-style-type: none"><li>• Brugeren forsøger igen at tænde iltmåleren.</li></ul> <p>3.1 (alt 2) Displayet viser ikke korrekte værdier.</p> <ul style="list-style-type: none"><li>• Brugeren undersøger forbindelsen mellem iltmåler og Arduino Uno og forsøger igen.</li></ul>
<b>Udvidelsesmuligheder</b>	<ol style="list-style-type: none"><li>1. Iltmåler enheden kan opgraderes til at kunne forbinde til internettet og overføre den målte værdi til en cloud.</li><li>2. Iltmåleren kan udvides så den målte værdi bliver læst op.</li><li>3. Iltmåleren kan udvides til at i stedet for at vise den målte værdi, så viser den delta værdien (den forrige målte værdi trukket fra den nyeste målte værdi) dette gør at man kender iltmængdens udvikling, er den stigende eller faldende.</li></ol>
<b>Åbne problemer</b>	Ingen.

Mathias Wriedt Kamp

Marius Martin Møller

<b>Use Case navn</b>	<b>Automatisk start / stop muslinge- bur</b>
<b>Id</b>	2
<b>Version</b>	1.0
<b>Beskrivelse</b>	Denne use case beskriver, hvordan det automatiserede muslinge- bur åbner og lukker muslinger ud, når iltniveauet er lavt.
<b>Problemstillingen</b>	Iltniveauet er lavt, og systemet ønsker at lukke muslinger ud, for at hæve iltniveauet.
<b>Scope</b>	Muslinge- bur.
<b>Aktør(er)</b>	Systemet.
<b>Stakeholder og Inte- resser</b>	Systemet – ønsker at åbne for muslingeburet, da iltniveauet er målt til at være lavt.
<b>Prækonditioner</b>	<ol style="list-style-type: none"><li>1. Systemet er tændt.</li><li>2. Systemet har målt en lav iltmængde.</li><li>3. Muslingeburets mekaniske system er i funktion og klar til at lukke muslinger ud.</li><li>4. Der er muslinger i muslingeburet klar til at blive sat ud.</li></ol>
<b>Postkonditioner</b>	<ol style="list-style-type: none"><li>1. Muslingeburet har været tippet.</li><li>2. Muslingerne har forladt buret.</li><li>3. Muslingeburet er returneret til stående position.</li></ol>
<b>Success forløb</b>	<ol style="list-style-type: none"><li>1. Systemet måler iltniveauet til at være lavt.</li><li>2. Systemet sender en kommando til muslingeburet om at lukke muslinger ud.</li><li>3. Muslingeburet aktiverer sin mekanisme til at tippe.</li><li>4. Muslingeburet åbner lågen.</li><li>5. Muslingerne udsættes i vandet.</li><li>6. Muslingeburet lukker lågen.</li><li>7. Muslingeburet vender tilbage til stående position.</li></ol>



Mathias Wriedt Kamp

Marius Martin Møller

<b>Alternativt forløb</b>	4.1 (alt 1) Der er ingen muslinger i buret <ul style="list-style-type: none"> <li>• Systemet fortsætter uden at lukke nogen ud.</li> </ul>
<b>Udvidelsesmuligheder</b>	1. Det kunne være hensigtsmæssigt at integrere buret med internettet, hvilket ville give mulighed for, at buret kan sende relevante notifikationer. Såsom, buret er tomt, der er nu lukket muslingerne ud.
<b>Åbne problemer</b>	Ingen.

<b>Use Case navn</b>	<b>Start diffust fugleskræmsel</b>
<b>Id</b>	3
<b>Version</b>	1.0
<b>Beskrivelse</b>	Denne use-case illustrerer hvordan det automatiserede fugleskræmsel går fra at være stoppet til at starte.
<b>Problemstillingen</b>	Edderfugle spiser muslinger. Ved udsættelse af muslinger skal edderfuglene blive skræmt væk. Systemet ønsker at starte fugleskræmslet.
<b>Scope</b>	Fugleskræmsel.
<b>Aktør(er)</b>	Systemet.
<b>Stakeholder og Interesser</b>	Systemet – ønsker at starte fugleskræmslet for at skræmme edderfugle væk.
<b>Prækonditioner</b>	<ol style="list-style-type: none"> <li>1. systemet er tændt.</li> <li>2. fugleskræmslet er forbundet til iltsensoren.</li> <li>3. Der er forbindelse mellem iltsensoren og fugleskræmslet.</li> </ol>
<b>Postkonditioner</b>	<ol style="list-style-type: none"> <li>1. fugleskræmslet er startet.</li> <li>2. fugleskræmslet bevæger sig diffust (uforudsigeligt).</li> </ol>
<b>Success forløb</b>	<ol style="list-style-type: none"> <li>1. fugleskræmslet modtager en besked fra iltmåleren og afkoder den til at det skal starte.</li> </ol>

Mathias Wriedt Kamp

Marius Martin Møller

	2. fugleskræmslet starter.
<b>Alternativt forløb</b>	1.1. (alt 1) Beskeden som fugleskræmslet har modtaget, kan ikke afkodes. Fugleskræmslet fortsætter uden ændring.
<b>Udvidelsesmuligheder</b>	1. Hvis fugleskræmslet var tilsluttet til internettet, ville det muliggøre fjernstyring af fugleskræmslet.
<b>Åbne problemer</b>	Ingen.

<b>Use Case navn</b>	<b>Stop diffust fugleskræmsel</b>
<b>Id</b>	4
<b>Version</b>	1.0
<b>Beskrivelse</b>	Denne use-case illustrerer hvordan det automatiserede fugleskræmsel går fra at være startet til stoppet.
<b>Problemstillingen</b>	Iltniveauet er nået et tilpas niveau, og det er blevet tid til at muslingerne skal høstes. Systemet ønsker at stoppe fugleskræmslet, for at spare strøm.
<b>Scope</b>	Fugleskræmsel.
<b>Aktør(er)</b>	System.
<b>Stakeholder og Interesser</b>	System – ønsker at stoppe fugleskræmslet.
<b>Prækonditioner</b>	<ol style="list-style-type: none"> <li>1. systemet er tændt.</li> <li>2. fugleskræmslet er forbundet til iltsensoren.</li> <li>3. Der er forbindelse mellem iltsensoren og fugleskræmslet.</li> <li>4. fugleskræmslets lemmer kan bevæge sig frit.</li> </ol>
<b>Postkonditioner</b>	<ol style="list-style-type: none"> <li>1. Fugleskræmslet har stoppet med at bevæge sig.</li> </ol>

Mathias Wriedt Kamp

Marius Martin Møller

<b>Success forløb</b>	<ol style="list-style-type: none"> <li>1. fugleskræmslet modtager en kommando fra iltsensoren om at den skal stoppe med at bevæge sig.</li> <li>2. fugleskræmslet afkoder beskeden, til at den skal stoppe med at bevæge sig.</li> <li>3. Fugleskræmslet afbryder bevægelsesrutinen.</li> <li>4. Fugleskræmslet er stoppet.</li> </ol>
<b>Alternativt forløb</b>	<ol style="list-style-type: none"> <li>1.1. (alt 1) Fugleskræmslet har modtaget en ukendt kommando. - Fugleskræmslet fortsætter uden ændring.</li> <li>3.1. (alt 1) Fugleskræmslet har fået en fejl ved afbrydningen af bevægelsesrutinen. - Fugleskræmslet sender en besked tilbage til iltmåleren om at den ikke kan stoppe.</li> </ol>
<b>Udvidelsesmuligheder</b>	<ol style="list-style-type: none"> <li>1. Hvis fugleskræmslet var tilsluttet internettet, ville det muliggøre fjernstyring af fugleskræmslet.</li> </ol>
<b>Åbne problemer</b>	Ingen.

<b>Use Case navn</b>	<b>Gem iltmåling</b>
<b>Id</b>	5
<b>Version</b>	1.0
<b>Beskrivelse</b>	Denne use-case viser hvordan iltmåleren sender iltmålingen videre til systemet og bliver gemt i databasen.
<b>Problemstillingen</b>	Hver gang iltmåleren tager en måling ønsker systemet at gemme den, for at holde styr på hvad iltindholdet har været på de givende tidspunkter.
<b>Scope</b>	Iltmåler.
<b>Aktør(er)</b>	System.

Mathias Wriedt Kamp

Marius Martin Møller

<b>Stakeholder og Interesser</b>	System – ønsker at gemme iltmåling.
<b>Prækonditioner</b>	<ol style="list-style-type: none"><li>1. systemet er tændt.</li><li>2. Iltmåleren er klar til at måle.</li><li>3. Iltmålerens kommunikationsmodul har forbindelse til internettet.</li><li>4. web Api og database er tændt og klar til at modtage iltmålinger.</li></ol>
<b>Postkonditioner</b>	<ol style="list-style-type: none"><li>1. Den målte værdi er gemt i databasen.</li></ol>
<b>Success forløb</b>	<ol style="list-style-type: none"><li>1. Iltmåleren måler en iltmåling.</li><li>2. Iltmåleren sender værdien til Api 'et.</li><li>3. Api 'et har modtager værdien, sender den videre til databasen.</li><li>4. værdien gemmes i databasen.</li><li>5. iltmåleren modtager et svar fra api'et om at værdien er oprettet.</li></ol>
<b>Alternativt forløb</b>	5.1 (alt 1) Iltmålingen kunne ikke gemmes, systemet giver en fejlbesked tilbage.
<b>Udvidelsesmuligheder</b>	<ol style="list-style-type: none"><li>1. Administratorerne af systemet kunne modtage én fejl notifikation i form af mail eller notifikation på telefonen når der sker fejl i systemet.</li></ol>
<b>Åbne problemer</b>	Ingen.

Mathias Wriedt Kamp

Marius Martin Møller

## Usability

- På systemets display skal hvert af tekstens bogstaver fylde 1 pixel.

## Performance

### Svartid

- Systemet skal sikre, at aktiveringen af fugleskræmslet sker inden for en maksimal tidsramme på 10 sekunder efter, at muslingerne er blevet udsat.
- Buret skal kunne frigive flere muslinger inden for en maksimal tidsramme på 10 sekunder efter modtagelse af beskeden.

### Opstarts- og nedlukningstid

- 1 minut om at lukke ned.
- 2 minutter om at starte op.

## Design Constraints

- Alt kode og dokumentation er skrevet på Engelsk.

## Interfaces

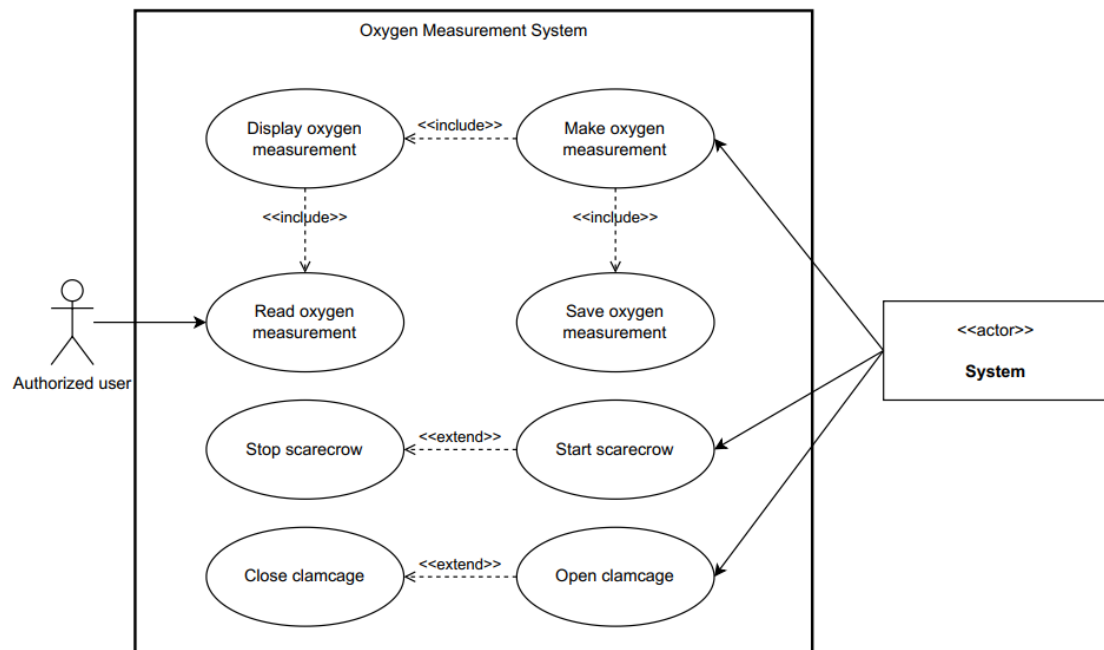
### Software interfaces

- Web API 'et skal være platformuafhængig og skal kunne afvikles på Linux, macOS og Windows.

Mathias Wriedt Kamp

Marius Martin Møller

## Use-case diagram

[Bilag\System documentation\Usecase diagrams\usecase diagram.pdf](#)

Figur 1 Use-case diagram.

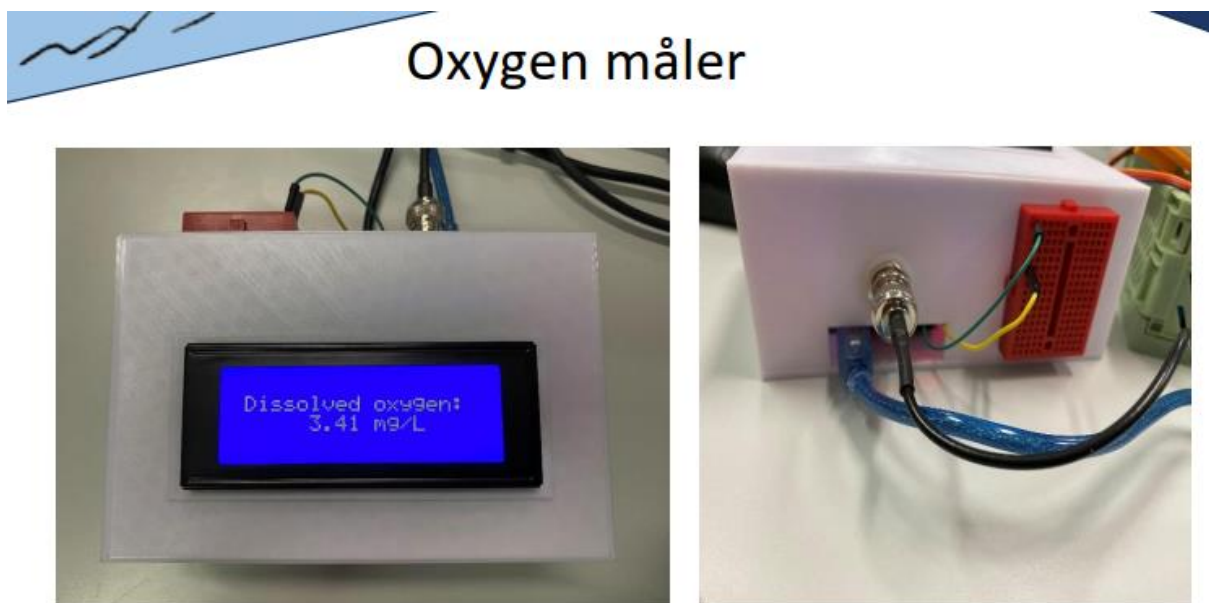
Mathias Wriedt Kamp

Marius Martin Møller

## Produktblade

### Oxygenmåler

For at samle oxygensensoren og de tilhørende komponenter på en nem og overskuelig måde har vi udarbejdet en 3D-printet kasse, der specifikt er designet til at rumme oxygensensoren, Arduino Uno, Arduino MKR1010 samt et breadboard, som kan anvendes til I2C-kommunikation. Det fulde produktblad kan findes her [Bilag\product datasheet\Oxygen sensor\Oxygen sensor - datasheet.pdf](#).



Figur 2 udklip fra produktbladet over oxygensensoren

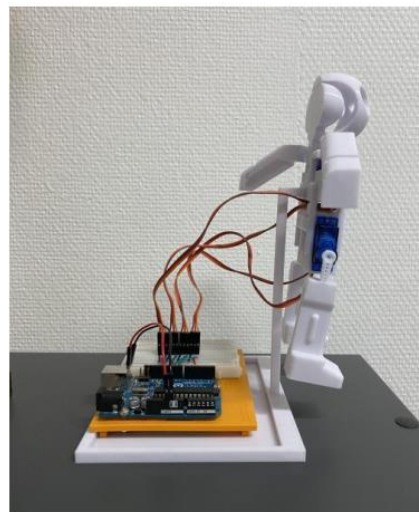
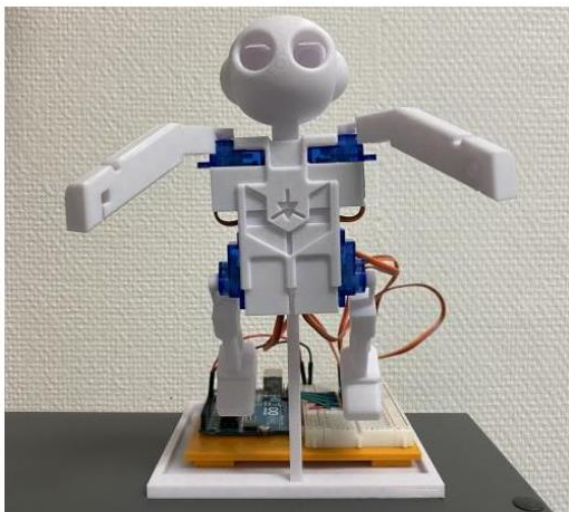
Mathias Wriedt Kamp

Marius Martin Møller

## Fugleskræmsel

for at lave et automatiseret fugleskræmsel som kan bevæge sig med et uforudsigeligt mønster, har vi special designet et fugleskræmsel som har bevægelige arme og ben. Det fulde produktblad kan ses her [Bilag\product datasheet\Scarecrow\Scarecrow - datasheet.pdf](#)

### Diffust fugleskræmsel



Figur 3 udklip af fugleskræmslets produktblad

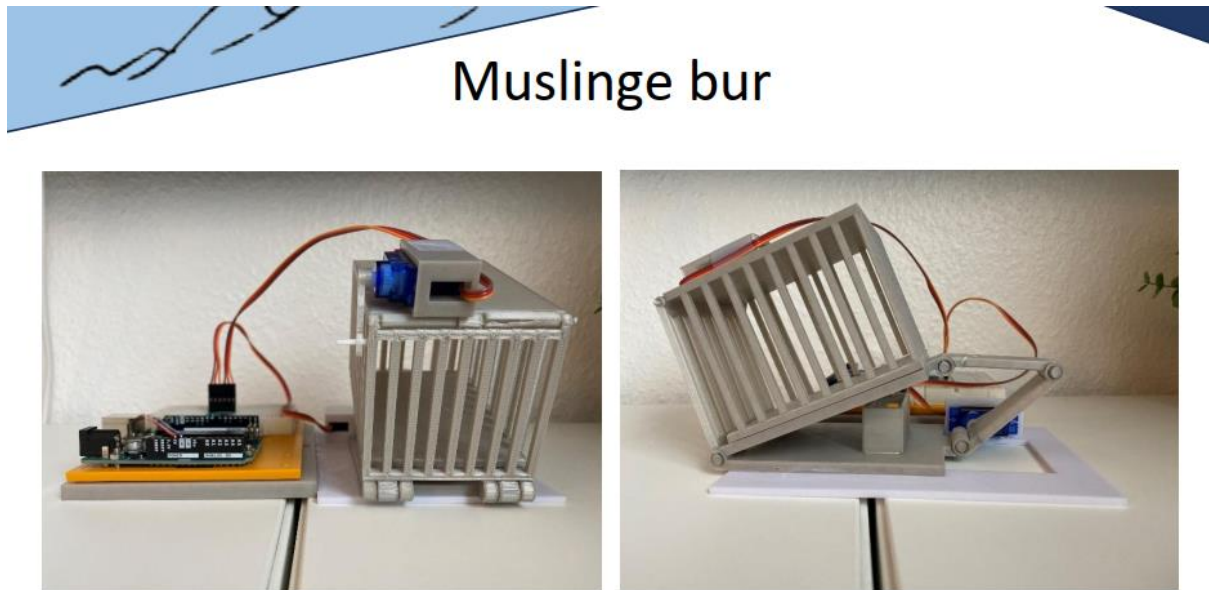


Mathias Wriedt Kamp

Marius Martin Møller

## Muslingebur

For at kunne håndtere automatisk tilførsel af muslinger har vi udarbejdet et 3D-printet muslingebur. Muslinge buret er udstyret med en vippe mekanisme. Det fulde produktblad kan ses her [Bilag\product datasheet\Clamcage\Clamcage - datasheet.pdf](#).



Figur 4 Viser et udklip af muslinge burets produktblad

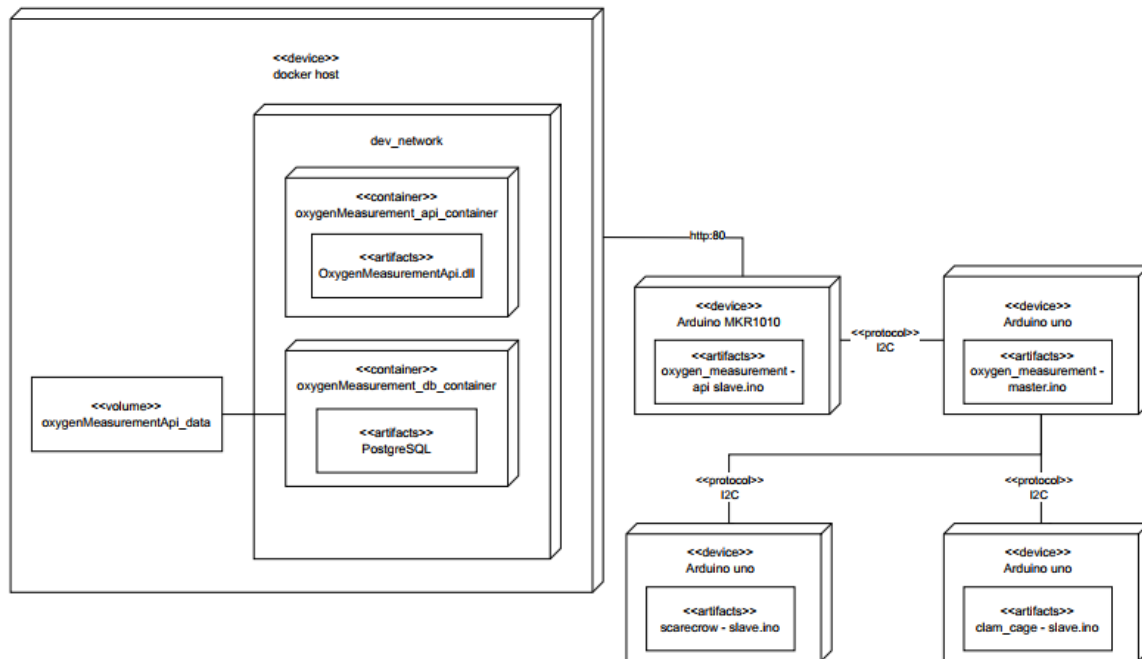
Mathias Wriedt Kamp

Marius Martin Møller

## Teknisk produkt dokumentation

### Deployment diagram

[Bilag\System documentation\Deployment diagram\deployment diagram.pdf](#)



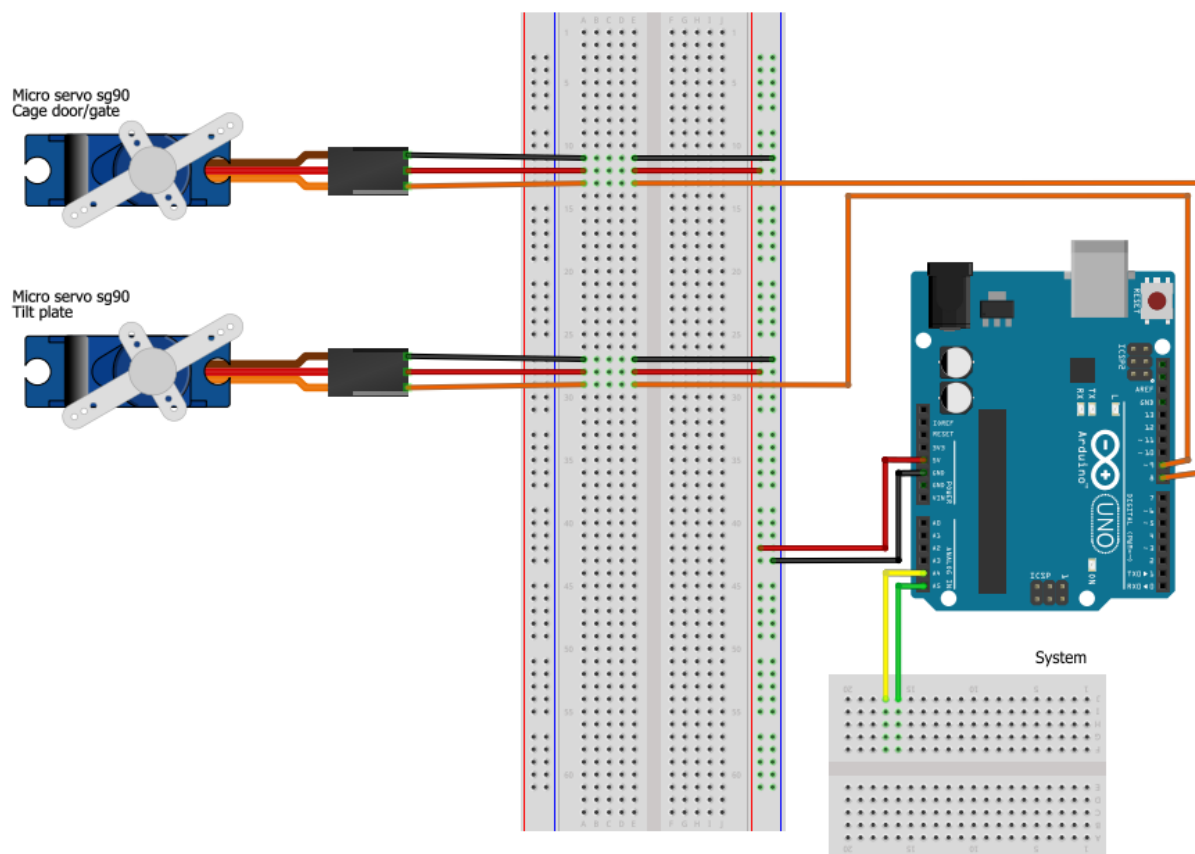
Figur 5 Deployment diagram over systemet.

Mathias Wriedt Kamp

Marius Martin Møller

## Arduino – Circuit diagrams

Arduino muslingebur – circuit diagram

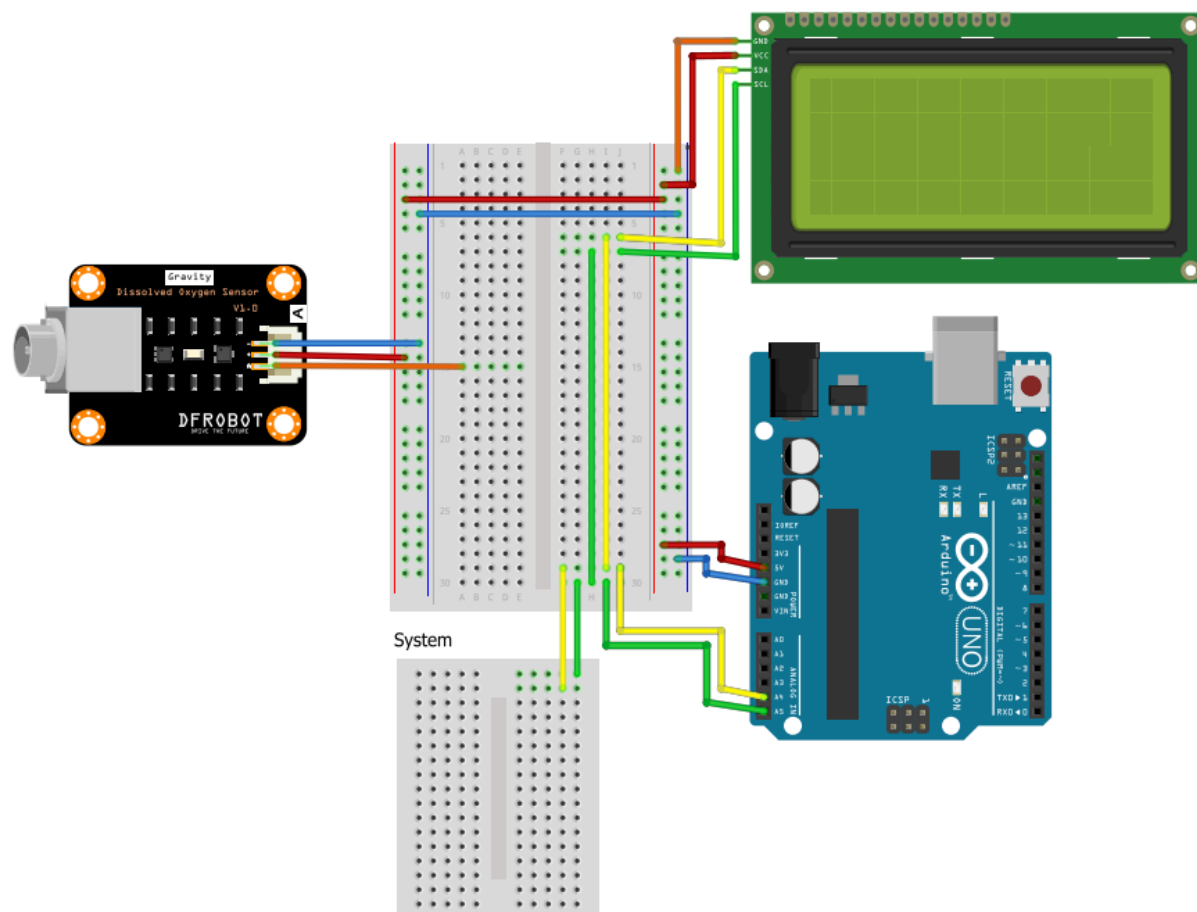
[Bilag\System documentation\Circuit diagrams\Clamcage circuit diagram.pdf](#)

Figur 6 Muslingebur circuit diagram

Mathias Wriedt Kamp

Marius Martin Møller

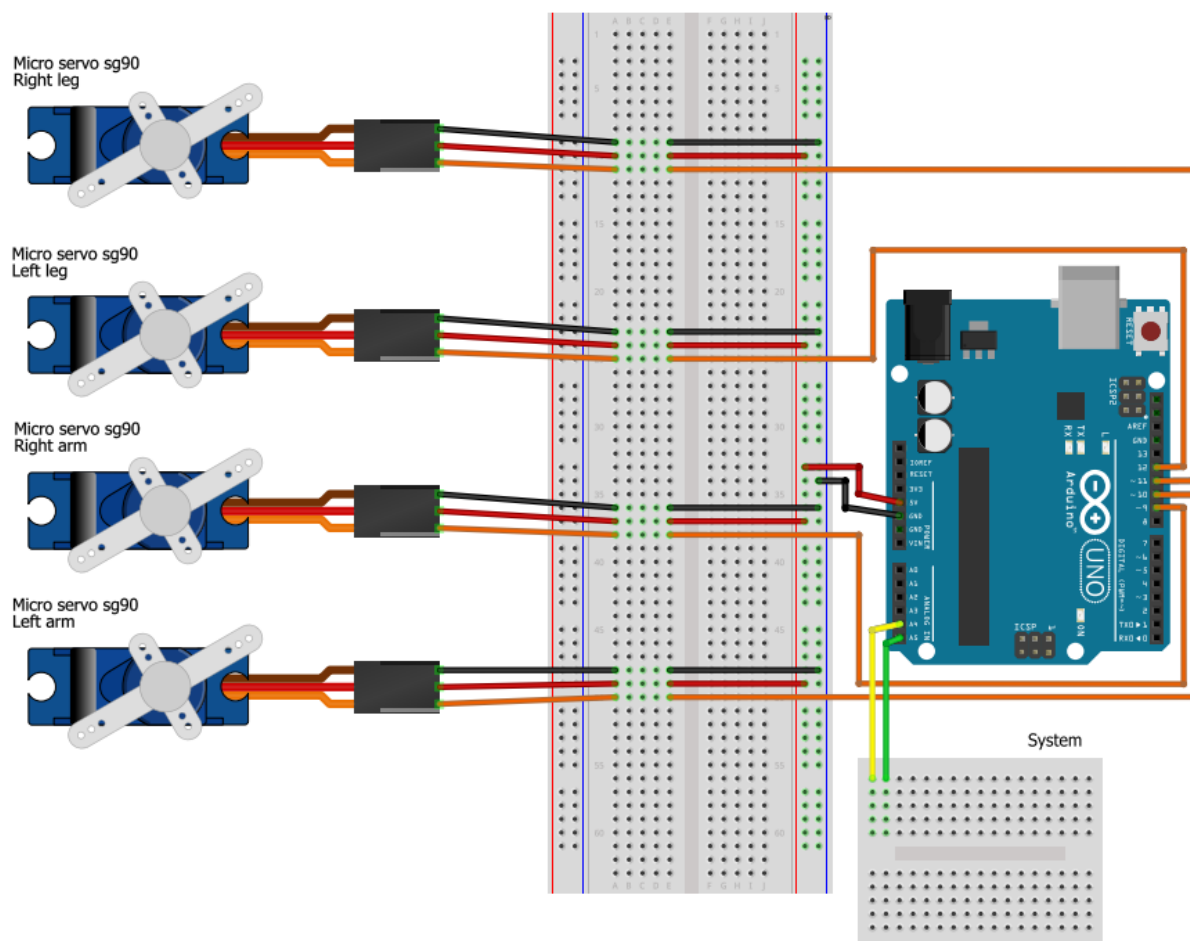
Iltmåler circuit diagram

[Bilag\System documentation\Circuit diagrams\Oxygen measurement circuit diagram.pdf](#)*Figur 7 Circuit diagram over iltmåleren*

Mathias Wriedt Kamp

Marius Martin Møller

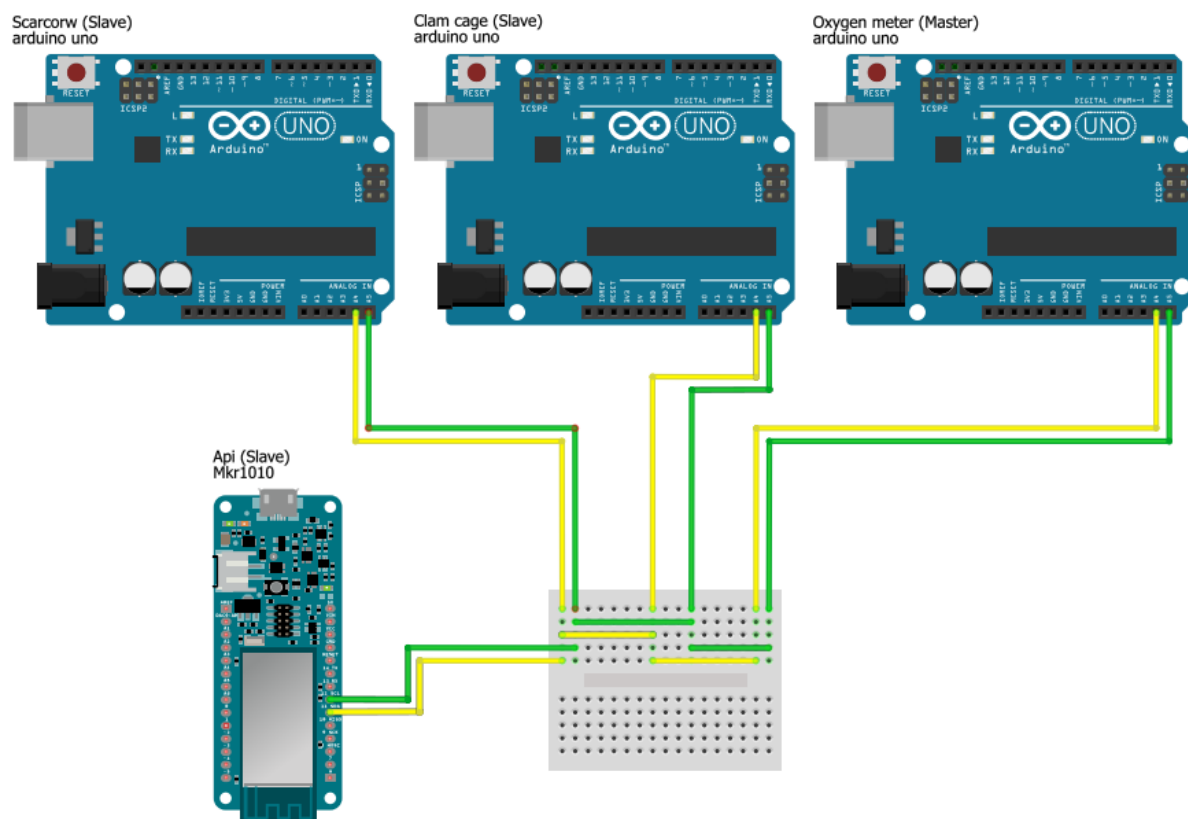
Fugleskræmsel circuit diagram

[Bilag\System documentation\Circuit diagrams\Scarecrow circuit diagram.pdf](#)*Figur 8 Circuit diagram over fugleskræmslet*

Mathias Wriedt Kamp

Marius Martin Møller

## Arduino system – circuit diagram

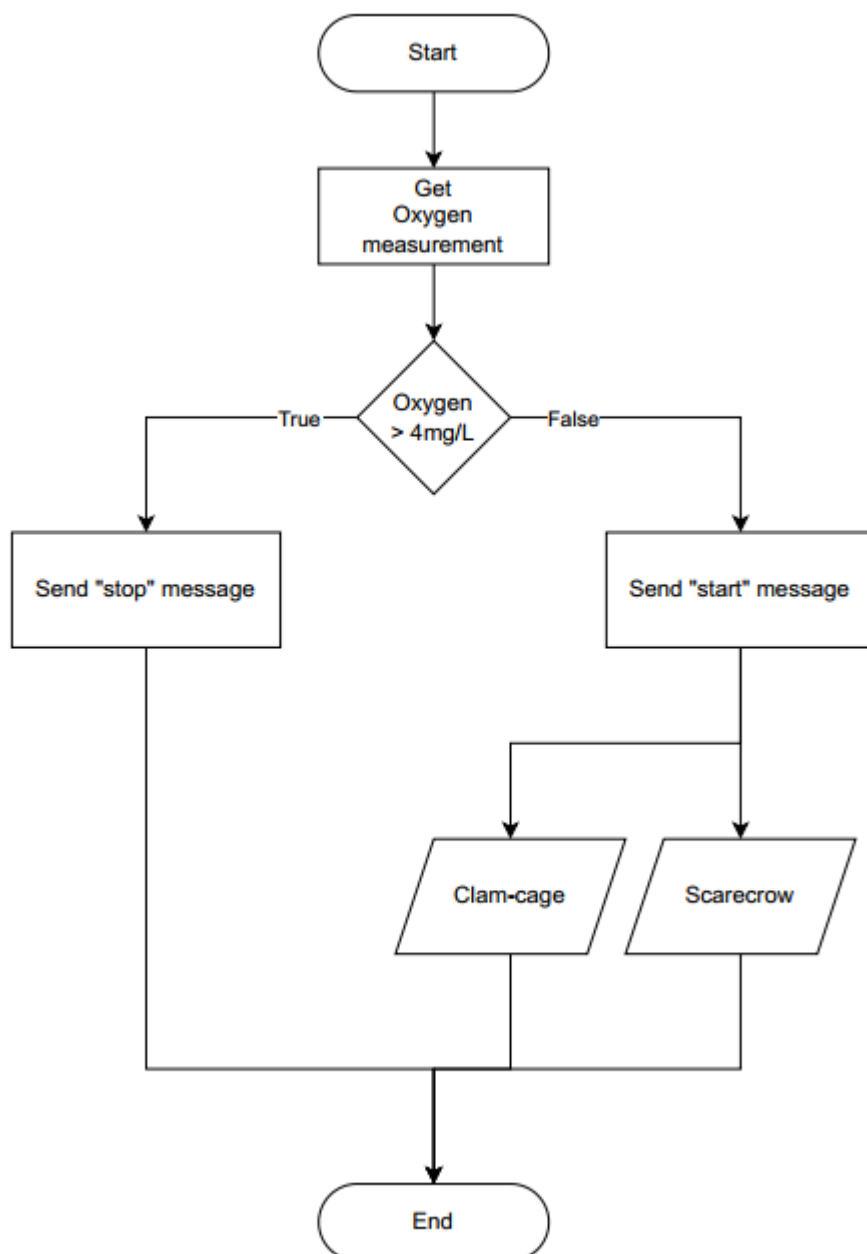
[Bilag\System documentation\Circuit diagrams\system circuit diagram.pdf](#)*Figur 9 Viser Arduino system Circuit diagram*

Mathias Wriedt Kamp

Marius Martin Møller

Arduino – Flow diagram

System – flowdiagram

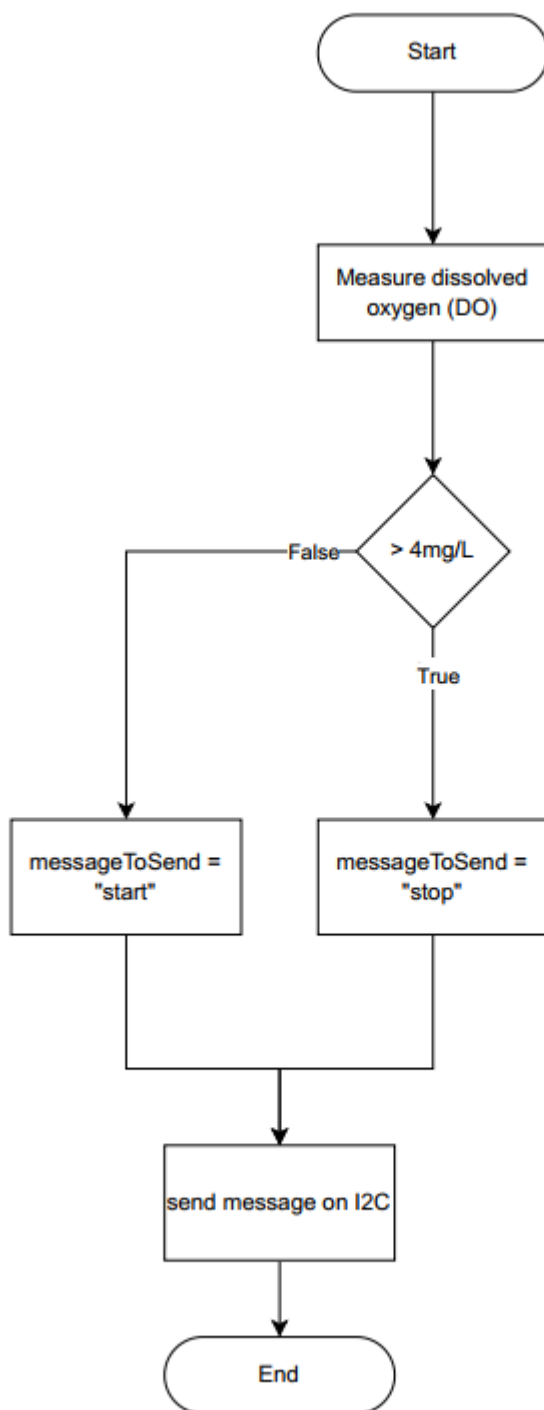
[Bilag\System documentation\Flow diagrams\System flow diagram.pdf](#)

Figur 10 System flowdiagram

Mathias Wriedt Kamp

Marius Martin Møller

Iltmåler flowdiagram

[Bilag\System documentation\Flow diagrams\Oxygen Sensor flow diagram.pdf](#)

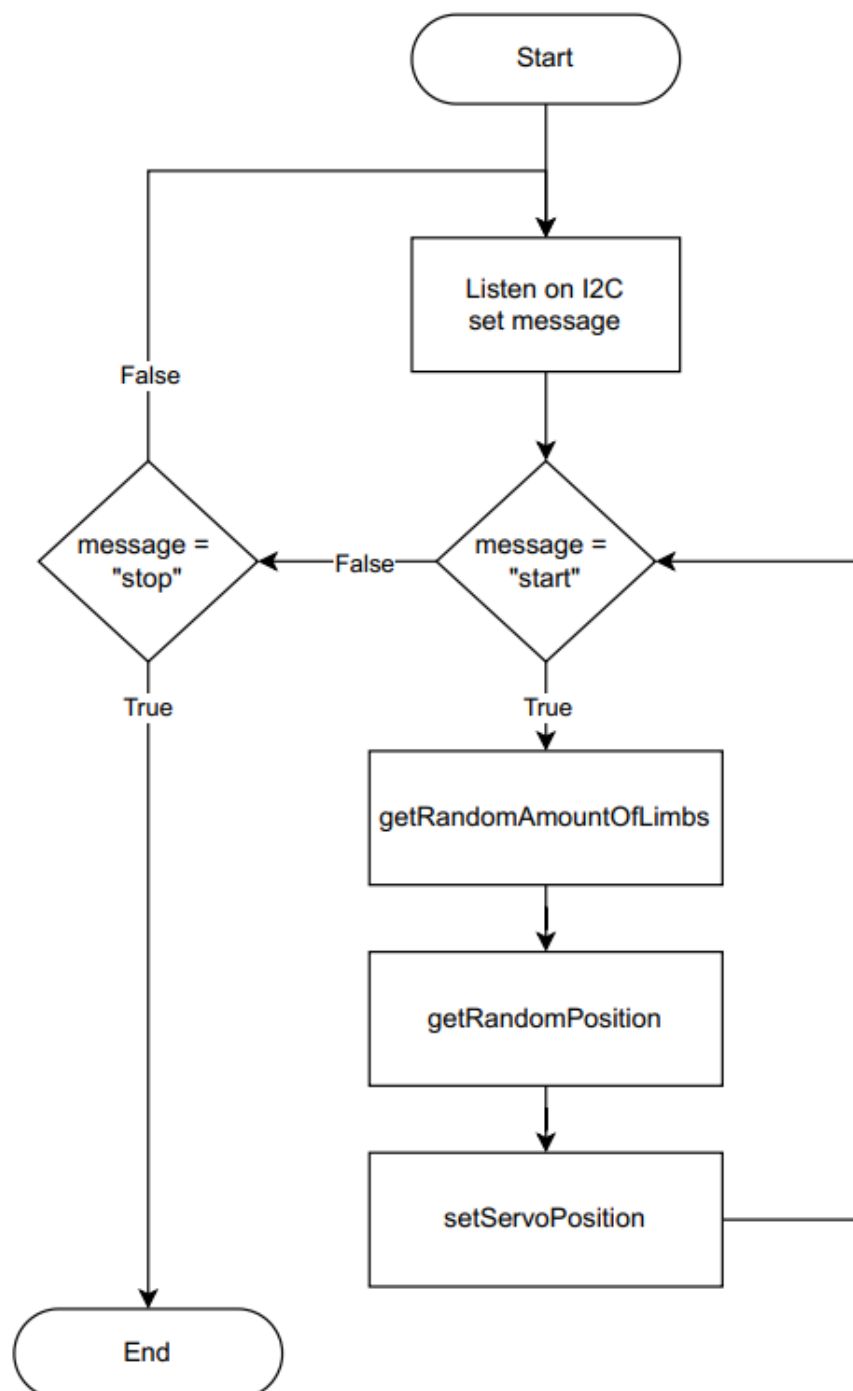
Figur 11 Iltmåler flowdiagram



Mathias Wriedt Kamp

Marius Martin Møller

Fugleskræmsel flowdiagram

[Bilag\System documentation\Flow diagrams\Scarecrow flow diagram.pdf](#)

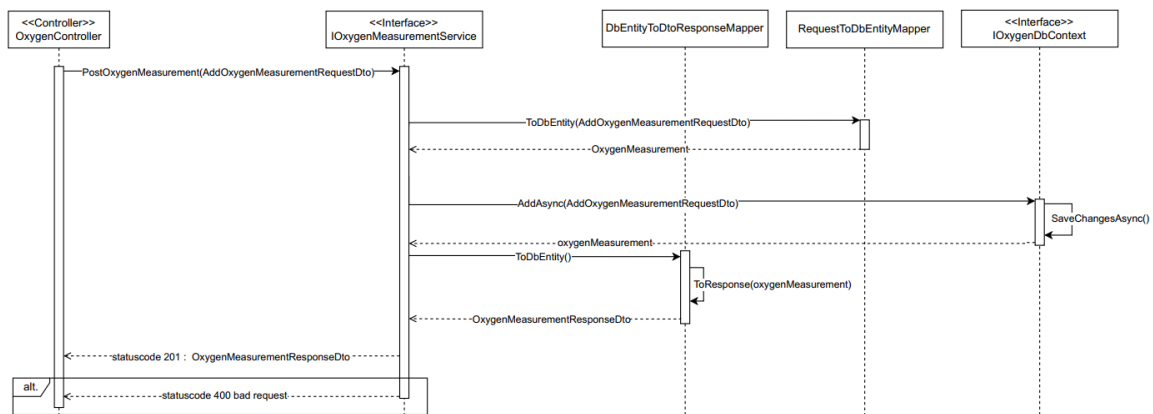
Figur 12 fugleskræmsel flowdiagram

Mathias Wriedt Kamp

Marius Martin Møller

## SD-diagrammer

[Bilag\System documentation\SSD-diagrams\SSD - create\\_new\\_oxygen\\_measurement.pdf](#)



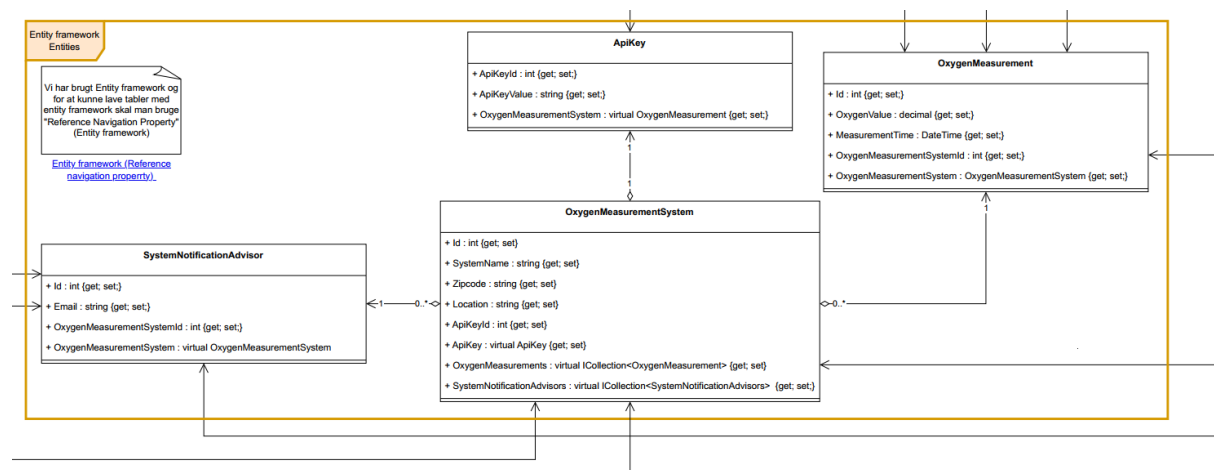
Figur 13 Viser SSD diagrammet over create new oxygen measurement.

## Klasse diagrammer

### Api klasse diagram

Det fulde klassediagram kan ses på nedenstående link.

[Bilag\System documentation\Class diagrams\Api\Class diagram.png](#)

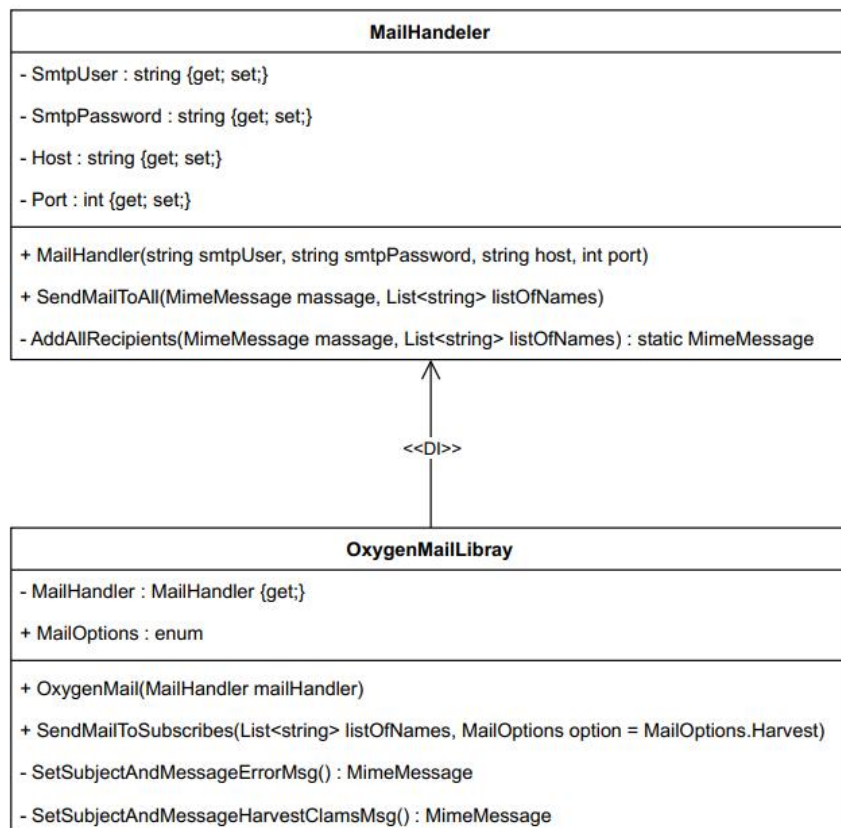


Figur 14 Viser klassediagrammet over Web api'et

Mathias Wriedt Kamp

Marius Martin Møller

## MailLibrary klasse diagram

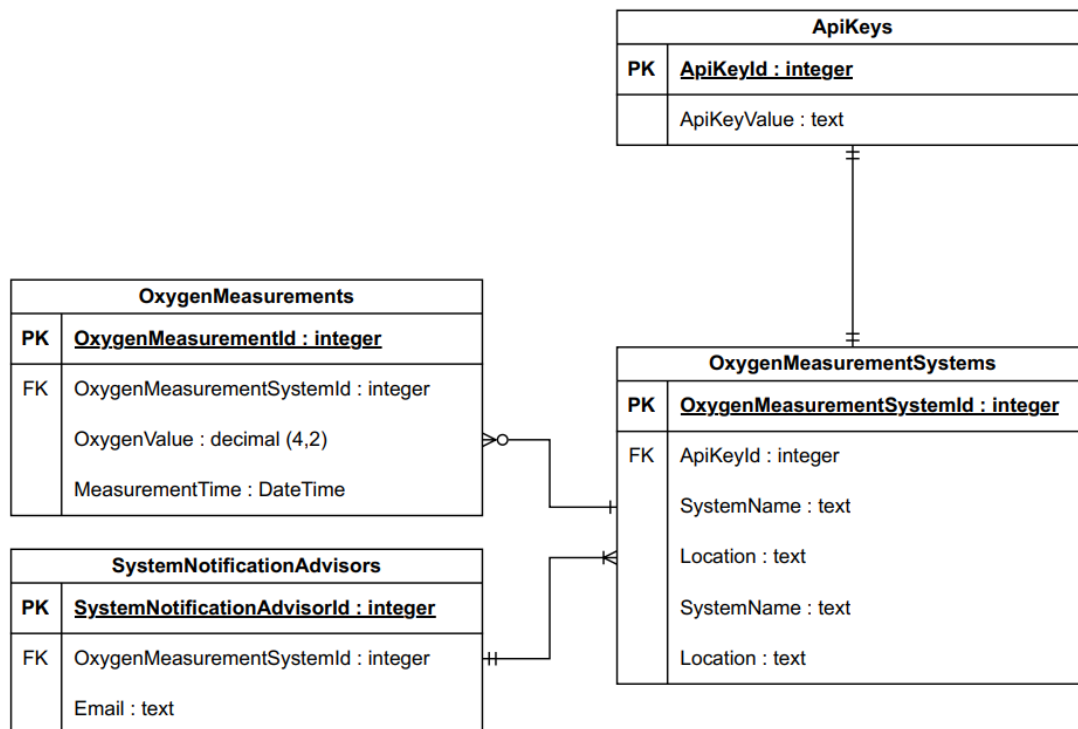
[Bilag\System documentation\Class diagrams\MailLibrary\Mail library class diagram.pdf](#)

Figur 15 Viser klassediagrammet over mail Library

Mathias Wriedt Kamp

Marius Martin Møller

## Er diagram

[Bilag\System documentation\Er diagram\ER diagram.pdf](#)

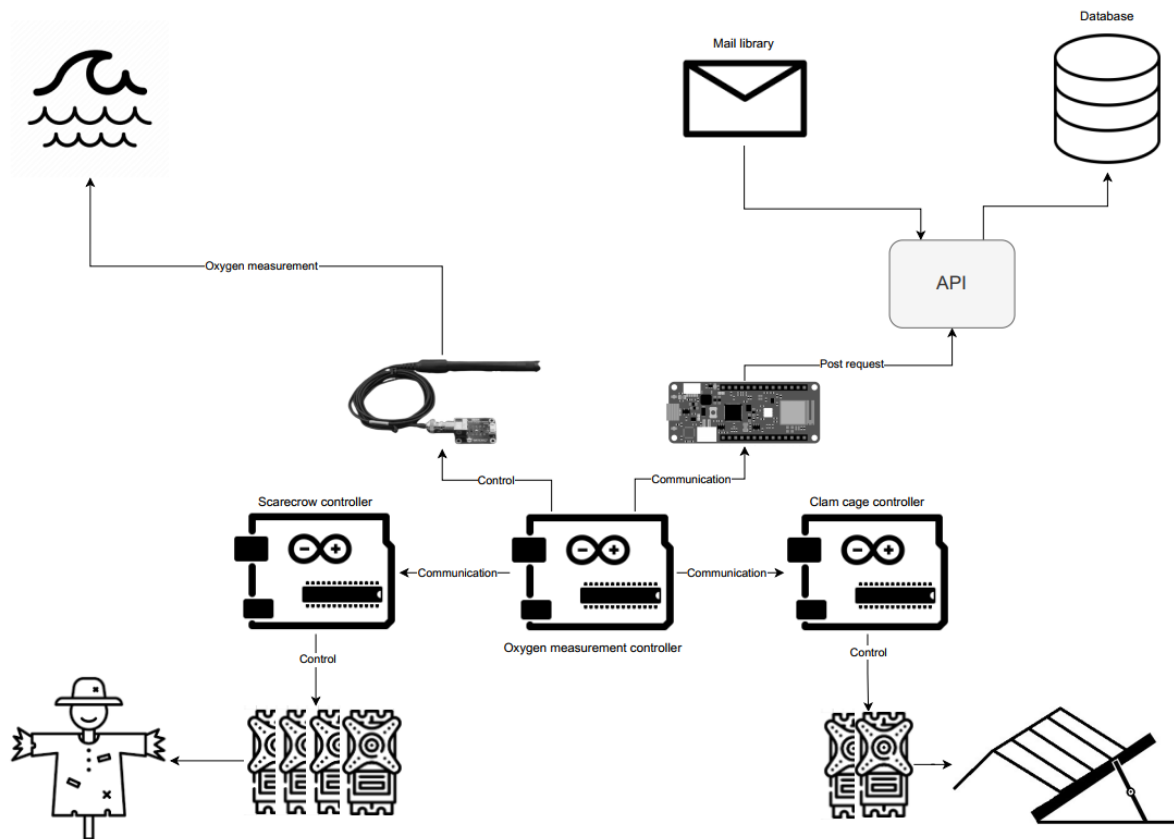
Figur 16 viser ER diagrammet over databasen.

Mathias Wriedt Kamp

Marius Martin Møller

## Topologi

[Bilag\System documentation\Topology\Topology.pdf](#)



Figur 17 viser systemets Topologier

Mathias Wriedt Kamp

Marius Martin Møller

## Testrapport

Til vores unittests og integrationstests har vi valgt at implementere TestContainers, en tredjeparts NuGet-pakke (TestContainers, n.d.), der har vist sig værdifuld i vores testmiljø. TestContainers automatiserer processen med at oprette og håndtere containere specifikt til hver test. Dette giver os en mere præcis test, da vi starter med en ren container til hver test uden behov for manuel oprydning af testdata efterfølgende.

Denne tilgang eliminerer kompleksiteten ved at skulle håndtere en separat testdatabase og bidrager til at gøre vores tests mere pålidelige og uafhængige. Ved at bruge TestContainers kan vi fokusere på selve testscenarierne og undgå bekymringer omkring testdataintegritet mellem forskellige tests.

### Udførte unittests

#### 1. Should Add New Oxygen Measurement If Oxygen Measurement System Exists:

##### Formål:

- At verificere, at systemet korrekt kan tilføje en ny iltmåling, når der eksisterer et eksisterende iltmålingssystem.

##### Metode:

- En asynkron testmetode, der indsætter et testsystem, opretter en ny iltmåling og forsøger at tilføje den til databasen.

Mathias Wriedt Kamp

Marius Martin Møller

**Kildekode:**

```
public async Task Should_Add_New_OxygenMeasurement_If_OxygenMeasurementSystem_Exists()
{
    // Arrange
    // Insert a test oxygen measurement system
    var testSystem = TestDataProvider.GetTestOxygenMeasurementSystemRequestDto();
    var testSystemInserted = await OxygenMeasurementSystemService.AddOxygenMeasurementSystemAsync(testSystem);

    // Check if the test system was successfully inserted
    if (testSystemInserted != null)
    {
        // Create a new test oxygen measurement associated with the inserted system
        var newOxygenMeasurement = TestDataProvider.GetTestOxygenMeasurementRequestDto(testSystemInserted.Id);

        // Act
        // Add the new oxygen measurement
        var actual = await OxygenMeasurementService.AddOxygenMeasurementAsync(newOxygenMeasurement);

        // Assert
        // Verify that the returned object is not null
        Assert.NotNull(actual);

        // Verify that the returned object is of the correct type
        Assert.IsType<OxygenMeasurementResponseDto>(actual);

        // Verify that the returned object has a valid ID
        Assert.True(actual != null && actual.Id > 0);


        // Verify that the returned object has the correct oxygen value
        Assert.True(actual != null && newOxygenMeasurement.OxygenValue == actual.OxygenValue);
    }
    else
    {
        // assert a failure if the test system insertion fails
        Assert.True(false, "Failed to insert the test oxygen measurement system.");
    }
}
```

Figur 18 kildekode for testmetoden should add new Oxygen Measurement if Oxygen Measurement System Exists

**Resultater:**

- Testen er vellykket, hvis et responsobjekt modtages med en gyldig ID og korrekt iltværdi.
- Testen fejler, hvis indsættelsen af testsystemet mislykkes, eller hvis responsobjektet ikke opfylder forventningerne.

**Faktisk resultat:**

-  Should\_Add\_New\_OxygenMeasurement\_If\_OxygenMeasurementSystem\_Exists Success

Mathias Wriedt Kamp

Marius Martin Møller

**Test navn: When Add Oxygen Measurement Should Throw NotFoundException If Oxygen Measurement System Does Not Exist****Formål:**

- Verificere, at forsøg på at tilføje en iltmåling kaster en NotFoundException, hvis det angivne iltmålingssystem ikke eksisterer.

**Metode:**

- En Asynkron testmetode, der forsøger at oprette en iltmåling med forkert systemId.

**Kildekode:**

```
public async Task When_Add_OxygenMeasurement_Should_Throw_NotNotFoundException_If_OxygenMeasurementSystem_Does_Not_Exist ()
{
    // Arrange
    const int notValidSystemId = 999;
    var oxygenMeasurementRequestDto = TestDataProvider.GetTestOxygenMeasurementRequestDto(notValidSystemId);

    // Actual
    var actual = await Assert.ThrowsAsync<NotFoundException>(async () =>
        await OxygenMeasurementService.AddOxygenMeasurementAsync(oxygenMeasurementRequestDto));

    // Assert
    Assert.Contains("not found", actual.Message);
}
```

Figur 19 Viser kildekoden for testmetoden *When Add Oxygen Measurement Should Throw NotFoundException If Oxygen Measurement System Does Not Exist*

**Resultater:**

- Testen er vellykket, hvis forsøget på at tilføje en iltmåling kaster en NotFoundException som forventet
- Testen fejler, hvis der ikke kastes en NotFoundException eller hvis fejlmeddelelsen ikke indeholder "not found"

**Faktisk resultat:**

✓ when\_add\_oxygenMeasurement\_should\_throw\_notNotFoundException\_if\_oxygenMeasurementSystem\_does\_notExists Success



Mathias Wriedt Kamp

Marius Martin Møller

## Udførte integrationstests

**Testnavn: Add Oxygen Measurement should return StatusCode 201 If Oxygen Measurement System exists.**

### Formål:

- verificere, at forsøg på at tilføje en iltmåling returnerer statuskoden 201 (created), hvis det angivne iltmålingssystem eksisterer.

### Metode:

- En Asynkron testmetode, der kalder endpointet til at oprette et testsystem, og derefter kalder endpointet til at oprette en iltmåling.

### Kildekode:

```
public async Task Add_OxygenMeasurement_shouldReturn_statusCode_201_if_OxygenMeasurementSystem_exists()
{
    // Arrange
    // Insert test system
    var testSystem = TestDataProvider.GetTestOxygenMeasurementSystemRequestDto();

    // Act
    // Post the test system to the endpoint for creating oxygen measurement systems
    var testSystemResponse = await HttpClient.PostAsJsonAsync("api/OxygenSystem/PostOxygenMeasurementSystem", testSystem);
    // Parse the response into an object for further usage
    var parsedTestSystemResponse = await testSystemResponse.Content.ReadFromJsonAsync<OxygenMeasurementSystemResponseDto>();

    // Check if the parsing was successful
    if (parsedTestSystemResponse != null)
    {
        // Set request headers with the API key and system ID for subsequent requests
        HttpClient.SetRequestHeaders(parsedTestSystemResponse.ApiKeyValue, parsedTestSystemResponse.Id);
        var newOxygenMeasurement = TestDataProvider.GetTestOxygenMeasurementRequestDto(parsedTestSystemResponse.Id);

        // Act
        // Post the new oxygen measurement to the endpoint
        var response = await HttpClient.PostAsJsonAsync("api/Oxygen/PostOxygenMeasurement", newOxygenMeasurement);

        // Get the status code from the response
        var createdOxygenMeasurementResponse = await response.Content.ReadFromJsonAsync<OxygenMeasurementResponseDto>();

        // Assert
        // Check that the status code is 201 (Created)
        Assert.True(response.StatusCode == HttpStatusCode.Created);

        // Check that the response content is not null
        Assert.NotNull(createdOxygenMeasurementResponse);

        Assert.True(newOxygenMeasurement.OxygenValue == createdOxygenMeasurementResponse.OxygenValue);
    }
}
```

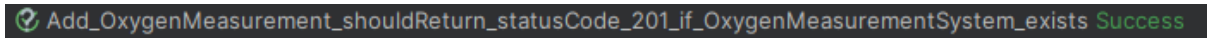
Figur 20 Viser kildekoden for testmetoden Add Oxygen Measurement should return StatusCode 201 If Oxygen Measurement System exists.

### Resultater:

- Testen er succesfuld, hvis forsøget på at tilføje en oxygenmåling returnerer statuskode 201 (Oprettet).
- Testen mislykkes, hvis statuskoden ikke er 201.

Mathias Wriedt Kamp

Marius Martin Møller

**Faktisk resultat:****Testnavn: add oxygen measurement system should return statuscode 201****Formål:**

- Verificer, at forsøget på at tilføje et oxygenmålesystem returnerer statuskode 201 (Oprettet).

**Metode:**

- En Asynkron testmetode, der kalder endpointet til at oprette et testsystem.

**Kildekode:**

```
public async Task Add_OxygenMeasurementSystem_shouldReturn_statusCode_201()
{
    // Arrange
    // Insert test system
    var testSystem = TestDataProvider.GetTestOxygenMeasurementSystemRequestDto();

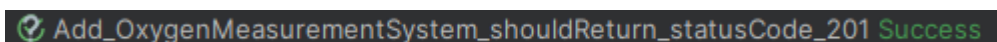
    // Act
    var testSystemResponse = await HttpClient.PostAsJsonAsync("api/OxygenSystem/PostOxygenMeasurementSystem", testSystem);
    var parsedTestSystemResponse = await testSystemResponse.Content.ReadFromJsonAsync<OxygenMeasurementSystemResponseDto>();

    // Assert
    Assert.True(testSystemResponse.StatusCode == HttpStatusCode.Created);
    Assert.True(parsedTestSystemResponse != null && parsedTestSystemResponse.Id > 0);
}
```

Figur 21 Viser kildekoden for testmetoden add oxygen measurement system should return statuscode 201

**Resultater:**

- Testen er succesfuld, hvis forsøget på at tilføje et oxygenmålesystem returnerer statuskode 201 (Oprettet).
- Testen mislykkes, hvis statuskoden ikke er 201, eller hvis det parsede svarobjekt ikke opfylder forventningerne.

**Faktisk resultat:**

Mathias Wriedt Kamp

Marius Martin Møller

## Test Ikke-Funktionelle Krav

### Tekststørrelse på Display:

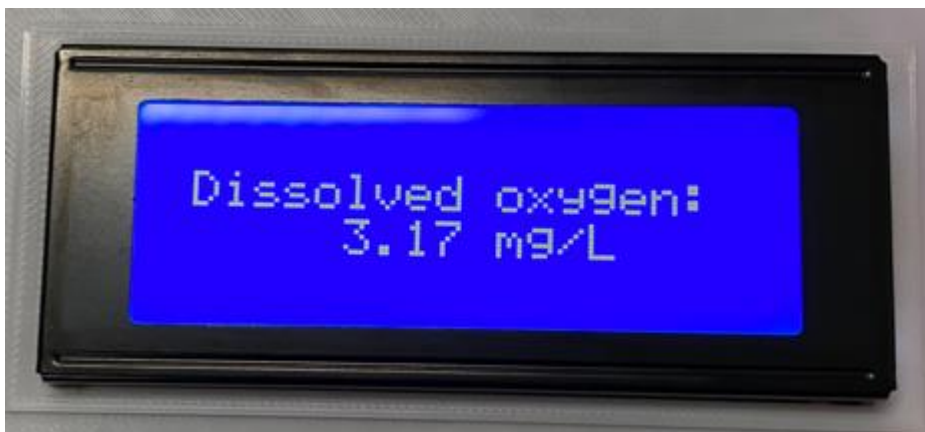
#### Krav:

- Hvert bogstav på systemets display skal fylde 1 pixel.

#### Test:

- Verificér, at hvert bogstav på displayet opfylder kravet om at fylde 1 pixel ved visuel inspektion.

#### Resultat:



Figur 22 Viser displayet har 1 bogstav pr pixel.

Mathias Wriedt Kamp

Marius Martin Møller

### **Aktivering af Fugleskræmsel:**

#### **Krav:**

- Systemet skal sikre, at aktiveringen af fugleskræmslet sker inden for en maksimal tidsramme på 10 sekunder efter, at muslingerne er blevet udsat.

#### **Test:**

- Mål tiden mellem udsættelse af muslinger og aktivering af fugleskræmslet for at verificere, at det sker inden for den specificerede tidsramme.

#### **Resultat:**

Fugleskræmslet starter inden for 5 sekunder efter at muslingeburet har sat muslinger ud.

Mathias Wriedt Kamp

Marius Martin Møller

**Frigivelse af Muslinger fra Buret:****Krav:**

- Buret skal kunne frigive flere muslinger inden for en maksimal tidsramme på 10 sekunder efter modtagelse af beskeden.

**Test:**

- Mål tiden mellem modtagelse af beskeden og frigivelse af muslinger for at sikre, at det sker inden for den specificerede tidsramme.

**Resultat:**

Kigger man i videoen vi har, så sker det typisk inden for et par sekunder efter iltmåleren har målt en lav værdi.

**Opstartstid:****Krav:**

- Systemet skal kunne starte op inden for 2 minutter.

**Test:**

- Mål tiden fra startkommandoen gives, indtil systemet er fuldt operationelt, for at verificere, at det sker inden for den specificerede tidsramme.

**Resultat:**

- Det tager cirka 30 sekunder for hele systemet at starte op, og være forbundet til hinanden via I2C samt til internettet.

Mathias Wriedt Kamp

Marius Martin Møller

**Nedlukningstid:****Krav:**

- Systemet skal kunne lukke ned inden for 1 minut.

**Test:**

- Mål tiden fra nedlukningskommandoen gives, indtil systemet er fuldt lukket ned, for at verificere, at det sker inden for den specificerede tidsramme.

**Resultat:**

Det tager cirka 20 sekunder for systemet at lukke ned.

**Platformuafhængighed af Web API:****Krav:**

- Web API'et skal være platformuafhængigt og kunne afvikles på Linux, macOS og Windows.

**Test:**

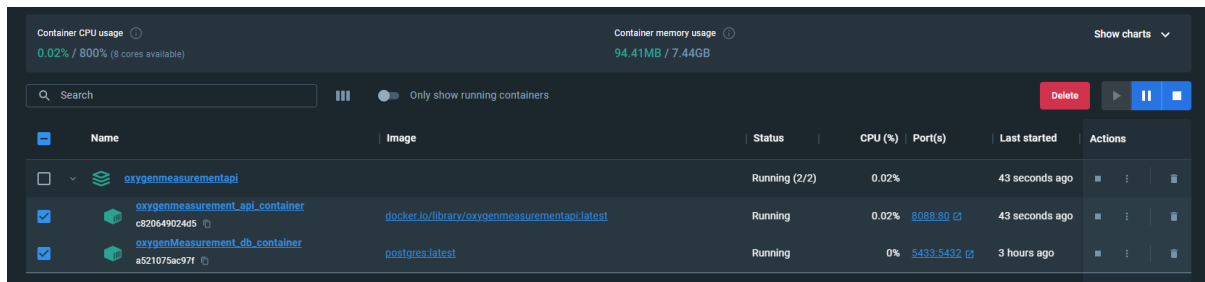
- Udfør test på hver platform (Linux, macOS, Windows) for at bekræfte, at Web API'et fungerer som forventet på hver platform.

**Resultat:**

Da Api'et kører i isoleret Docker container. Så kan det hostes på alle platforme da Docker / Docker Desktop er tilgængelig på Windows, Linux og MacOS.

Mathias Wriedt Kamp

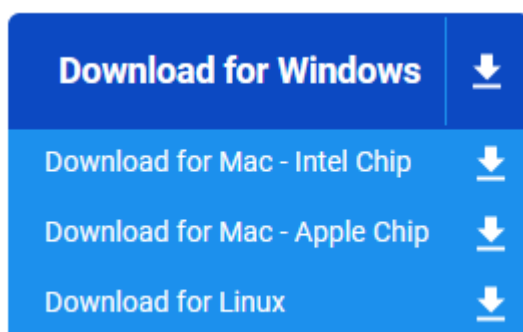
Marius Martin Møller



The screenshot shows the Docker Desktop interface. At the top, it displays 'Container CPU usage' at 0.02% / 800% (8 cores available) and 'Container memory usage' at 94.41MB / 7.44GB. Below this is a search bar and a toggle for 'Only show running containers'. A table lists the following containers:

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
oxygenmeasurementapi		Running (2/2)	0.02%		43 seconds ago	[Stop] [Refresh] [Delete]
oxygenmeasurement_api_container c820649024d5	docker.io/library/oxygenmeasurementapi:latest	Running	0.02%	8088:80	43 seconds ago	[Stop] [Refresh] [Delete]
oxygenMeasurement_db_container a521075ac97f	postgres:latest	Running	0%	5433:5432	3 hours ago	[Stop] [Refresh] [Delete]

Figur 23 Viser Windows versionen af Docker desktop, med Web Api og databasecontainerne



Figur 24 Viser at Docker Desktop er tilgængelig for Windows, Linux og MacOS.

Mathias Wriedt Kamp

Marius Martin Møller

*User Acceptance Test (UAT) - Aflæsning af Iltmængde use-case id 1*

Vi har haft 1 testperson til at udføre denne UAT. Testpersonen har en baggrund inden for IT.

**Formål:** Verificere, at systemet korrekt muliggør aflæsning af iltmængden i vand ved hjælp af oxygenmåleren og opfylder brugerens krav og forventninger.

**Forberedelse:**

1. Sikre, at iltmåleren er korrekt monteret og nedsænket i vandet.
2. Bekræft, at iltmåleren er korrekt kalibreret i henhold til producentens specifikationer.
3. Sikre, at strømforsyningen er tilsluttet.

**Testscenarier:****1. Tænding af Iltmåleren:**

- **Handling:**
  - Brugeren tænder iltmåleren.
- **Forventet Resultat:**
  - Iltmåleren starter korrekt uden problemer.

**2. Korrekt Aflæsning af Iltmængde:**

- **Handling:**
  - Brugeren placerer iltmåleren i vandet og sikrer, at iltmålerens probehoved er dækket af vand.
  - Brugeren aflæser iltmængden på displayet.
- **Forventet Resultat:**
  - Displayet viser iltmængden i mg/L korrekt.

**3. Håndtering af Tændingsproblemer:**

- **Alternativt Forløb (Hvis relevant):**
  - Iltmåleren er ikke tændt korrekt.
  - Brugeren forsøger igen at tænde iltmåleren.
- **Forventet Resultat:**
  - Iltmåleren tændes korrekt efter gentagne forsøg.

**4. Håndtering af Displayproblemer:**

- **Alternativt Forløb (Hvis relevant):**
  - Displayet viser ikke korrekte værdier.



Mathias Wriedt Kamp

Marius Martin Møller

- Brugeren undersøger forbindelsen mellem iltmåler og Arduino Uno og forsøger igen.
- **Forventet Resultat:**
  - Displayproblemer løses, og korrekte værdier vises efter undersøgelsen.

**Resultat og Godkendelse:**

- **Godkendt:** Alle testscenarier er gennemført med succes, og brugerens feedback er positiv.
- **Ikke Godkendt:** Et eller flere testscenarier mislykkedes, eller brugerens feedback indikerer uacceptable problemer.

## Testperson 1 resultat use-case id 1

UAT: Aflæsning af iltmængde (use-case id 1)			
Accept kriterie	Resultat		Kommentar
	Ja	Nej	
Kig på displayet.			
Det forventes at systemet har foretaget en iltmåling og viser den på displayet i mg/L. Står der en værdi?	X		Værdien er 4.33 mg/L

Mathias Wriedt Kamp

Marius Martin Møller

## Referencer

TestContainers. (n.d). Hentet fra <https://testcontainers.com/>

## Bilag

Alle dokumenter, billeder og diagrammer har en reference til det oprindelige dokument og hvor det kan findes.