A group of fish under water

Description automatically generated

**Produktrapport**

**Iltsvind**

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

Indholdsfortegnelse

[Titelblad 2](#_Toc152845517)

[Kravspecifikation 4](#_Toc152845518)

[Introduktion 4](#_Toc152845519)

[Formål med kravspecifikationen 4](#_Toc152845520)

[Definitioner, akronymer og forkortelser 4](#_Toc152845521)

[System 5](#_Toc152845522)

[Funktionalitet 6](#_Toc152845523)

[Usability 13](#_Toc152845524)

[Performance 13](#_Toc152845525)

[Svartid 13](#_Toc152845526)

[Opstarts- og nedlukningstid 13](#_Toc152845527)

[Design Constraints 13](#_Toc152845528)

[Interfaces 13](#_Toc152845529)

[Software interfaces 13](#_Toc152845530)

[Use-case diagram 14](#_Toc152845531)

[Produktblade 15](#_Toc152845532)

[Oxygenmåler 15](#_Toc152845533)

[Fugleskræmsel 16](#_Toc152845534)

[Muslingebur 17](#_Toc152845535)

[Teknisk produkt dokumentation 18](#_Toc152845536)

[Deployment diagram 18](#_Toc152845537)

[Arduino – Circuit diagrams 19](#_Toc152845538)

[Arduino muslingebur – circuit diagram 19](#_Toc152845539)

[Iltmåler circuit diagram 20](#_Toc152845540)

[Fugleskræmsel circuit diagram 21](#_Toc152845541)

[Arduino system – circuit diagram 22](#_Toc152845542)

[System – flowdiagram 23](#_Toc152845543)

[Iltmåler flowdiagram 24](#_Toc152845544)

[Fugleskræmsel flowdiagram 25](#_Toc152845545)

[SD-diagrammer 26](#_Toc152845546)

[Klasse diagram 26](#_Toc152845547)

[Er diagram 28](#_Toc152845548)

[Topologi 29](#_Toc152845549)

[Testrapport 30](#_Toc152845550)

[Udførte unittests 30](#_Toc152845551)

[Udførte integrationstests 33](#_Toc152845552)

[Test Ikke-Funktionelle Krav 35](#_Toc152845553)

[Testperson 1 resultat use-case id 1 41](#_Toc152845554)

[Referencer 42](#_Toc152845555)

[Bilag 42](#_Toc152845556)

# 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.

# 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.

# Funktionalitet

|  |  |
| --- | --- |
| **Use Case navn** | **Aflæsning af iltmængde** |
| **Id** | 1 |
| **Version** | 1.0 |
| **Beskrivelse** | 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. |
| **Problemstillingen** | En bruger skal have mulighed for at kunne aflæse iltindholdet i noget vand. |
| **Scope** | Iltmåleren bestående af Arduino Uno, iltmåler og display. |
| **Aktør(er)** | Bruger: Autoriseret person, der ønsker at aflæse iltmængden i vandet. |
| **Stakeholder og Interesser** | Bruger: Ønsker at få nøjagtige og pålidelige oplysninger om iltmængden i vandet for at vurdere miljøets tilstand. |
| **Prækonditioner** | * Iltmåleren er korrekt monteret og nedsænket i vandet. * iltmåleren er korrekt kalibreret i henhold til producentens specifikationer. * Strømforsyningen er tilsluttet. |
| **Postkonditioner** | Den målte iltmængde kan aflæses fra displayet. |
| **Success forløb** | 1. Brugeren tænder iltmåleren. 2. Brugeren placerer iltmåleren i vandet og sikre at iltmålerens probehovedet er dækket af vand. 3. Displayet viser iltmængden i mg/L 4. Brugeren aflæser iltmængden og vurderer vandkvaliteten. |
| **Alternativt forløb** | 1.1 (alt 1) Iltmåleren er ikke tændt korrekt.   * Brugeren forsøger igen at tænde iltmåleren.   3.1 (alt 2) Displayet viser ikke korrekte værdier.   * Brugeren undersøger forbindelsen mellem iltmåler og Arduino Uno og forsøger igen. |
| **Udvidelsesmuligheder** | 1. Iltmåler enheden kan opgraderes til at kunne forbinde til internettet og overføre den målte værdi til en cloud. 2. Iltmåleren kan udvides så den målte værdi bliver læst op. 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. |
| **Åbne problemer** | Ingen. |

|  |  |
| --- | --- |
| **Use Case navn** | **Automatisk start / stop muslingebur** |
| **Id** | 2 |
| **Version** | 1.0 |
| **Beskrivelse** | Denne use case beskriver, hvordan det automatiserede muslingebur åbner og lukker muslinger ud, når iltniveauet er lavt. |
| **Problemstillingen** | Iltniveauet er lavt, og systemet ønsker at lukke muslinger ud, for at hæve iltniveauet. |
| **Scope** | Muslingebur. |
| **Aktør(er)** | Systemet. |
| **Stakeholder og Interesser** | Systemet – ønsker at åbne for muslingeburet, da iltniveauet er målt til at være lavt. |
| **Prækonditioner** | 1. Systemet er tændt. 2. Systemet har målt en lav iltmængde. 3. Muslingeburets mekaniske system er i funktion og klar til at lukke muslinger ud. 4. Der er muslinger i muslingeburet klar til at blive sat ud. |
| **Postkonditioner** | 1. Muslingeburet har været tippet. 2. Muslingerne har forladt buret. 3. Muslingeburet er returneret til stående position. |
| **Success forløb** | 1. Systemet måler iltniveauet til at være lavt. 2. Systemet sender en kommando til muslingeburet om at lukke muslinger ud. 3. Muslingeburet aktiverer sin mekanisme til at tippe. 4. Muslingeburet åbner lågen. 5. Muslingerne udsættes i vandet. 6. Muslingeburet lukker lågen. 7. Muslingeburet vender tilbage til stående position. |
| **Alternativt forløb** | * 1. (alt 1) Der er ingen muslinger i buret * Systemet fortsætter uden at lukke nogen ud. |
| **Udvidelsesmuligheder** | 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. |
| **Åbne problemer** | Ingen. |

|  |  |
| --- | --- |
| **Use Case navn** | **Start diffust fugleskræmsel** |
| **Id** | 3 |
| **Version** | 1.0 |
| **Beskrivelse** | Denne use-case illustrerer hvordan det automatiserede fugleskræmsel går fra at være stoppet til at starte. |
| **Problemstillingen** | Edderfugle spiser muslinger. Ved udsættelse af muslinger skal edderfuglene blive skræmt væk. Systemet ønsker at starte fugleskræmslet. |
| **Scope** | Fugleskræmsel. |
| **Aktør(er)** | Systemet. |
| **Stakeholder og Interesser** | Systemet – ønsker at starte fugleskræmslet for at skræmme edderfugle væk. |
| **Prækonditioner** | 1. systemet er tændt. 2. fugleskræmslet er forbundet til iltsensoren. 3. Der er forbindelse mellem iltsensoren og fugleskræmslet. |
| **Postkonditioner** | 1. fugleskræmslet er startet. 2. fugleskræmslet bevæger sig diffust (uforudsigeligt). |
| **Success forløb** | 1. fugleskræmslet modtager en besked fra iltmåleren og afkoder den til at det skal starte. 2. fugleskræmslet starter. |
| **Alternativt forløb** | * 1. (alt 1) Beskeden som fugleskræmslet har modtaget, kan ikke afkodes. Fugleskræmslet fortsætter uden ændring. |
| **Udvidelsesmuligheder** | 1. Hvis fugleskræmslet var tilsluttet til internettet, ville det muliggøre fjernstyring af fugleskræmslet. |
| **Åbne problemer** | Ingen. |

|  |  |
| --- | --- |
| **Use Case navn** | **Stop diffust fugleskræmsel** |
| **Id** | 4 |
| **Version** | 1.0 |
| **Beskrivelse** | Denne use-case illustrerer hvordan det automatiserede fugleskræmsel går fra at være startet til stoppet. |
| **Problemstillingen** | 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. |
| **Scope** | Fugleskræmsel. |
| **Aktør(er)** | System. |
| **Stakeholder og Interesser** | System – ønsker at stoppe fugleskræmslet. |
| **Prækonditioner** | 1. systemet er tændt. 2. fugleskræmslet er forbundet til iltsensoren. 3. Der er forbindelse mellem iltsensoren og fugleskræmslet. 4. fugleskræmslets lemmer kan bevæge sig frit. |
| **Postkonditioner** | 1. Fugleskræmslet har stoppet med at bevæge sig. |
| **Success forløb** | 1. fugleskræmslet modtager en kommando fra iltsensoren om at den skal stoppe med at bevæge sig. 2. fugleskræmslet afkoder beskeden, til at den skal stoppe med at bevæge sig. 3. Fugleskræmslet afbryder bevægelsesrutinen. 4. Fugleskræmslet er stoppet. |
| **Alternativt forløb** | * 1. (alt 1) Fugleskræmslet har modtaget en ukendt kommando. - Fugleskræmslet fortsætter uden ændring.   2. (alt 1) Fugleskræmslet har fået en fejl ved afbrydningen af bevægelsesrutinen.  - Fugleskræmslet sender en besked tilbage til iltmåleren om at den ikke kan stoppe. |
| **Udvidelsesmuligheder** | 1. Hvis fugleskræmslet var tilsluttet internettet, ville det muliggøre fjernstyring af fugleskræmslet. |
| **Åbne problemer** | Ingen. |

|  |  |
| --- | --- |
| **Use Case navn** | **Gem iltmåling** |
| **Id** | 5 |
| **Version** | 1.0 |
| **Beskrivelse** | Denne use-case viser hvordan iltmåleren sender iltmålingen videre til systemet og bliver gemt i databasen. |
| **Problemstillingen** | Hver gang iltmåleren tager en måling ønsker systemet at gemme den, for at holde styr på hvad iltindholdet har været på de givende tidspunkter. |
| **Scope** | Iltmåler. |
| **Aktør(er)** | System. |
| **Stakeholder og Interesser** | System – ønsker at gemme iltmåling. |
| **Prækonditioner** | 1. systemet er tændt. 2. Iltmåleren er klar til at måle. 3. Iltmålerens kommunikationsmodul har forbindelse til internettet. 4. web Api og database er tændt og klar til at modtage iltmålinger. |
| **Postkonditioner** | 1. Den målte værdi er gemt i databasen. |
| **Success forløb** | 1. Iltmåleren måler en iltmåling. 2. Iltmåleren sender værdien til Api ‘et. 3. Api ‘et har modtager værdien, sender den videre til databasen. 4. værdien gemmes i databasen. 5. iltmåleren modtager et svar fra api’et om at vædien er oprettet. |
| **Alternativt forløb** | 5.1 (alt 1) Iltmålingen kunne ikke gemmes, systemet giver en fejlbesked tilbage. |
| **Udvidelsesmuligheder** | 1. Administratorerne af systemet kunne modtage én fejl notifikation i form af mail eller notifikation på telefonen når der sker fejl i systemet. |
| **Åbne problemer** | Ingen. |

## 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.

## Use-case diagram

[Bilag\System documentation\Usecase diagrams\usecase diagram.pdf](Bilag/System%20documentation/Usecase%20diagrams/usecase%20diagram.pdf)

A diagram of a diagram

Description automatically generated

Figur 1 Use-case diagram.

# 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](Bilag/product%20datasheet/Oxygen%20sensor/oxygen%20sensor%20-%20datasheet.pdf).

A close-up of a device

Description automatically generated

Figur 2 udklip fra produktbladet over oxygensensoren

Fugleskræmsel  
for at lave et automatiseret fugleskræmsel som kan bevæge sig med et uforudsigeligt mønster, har vi speciel designet et fugleskræmsel som har bevægelige arme og ben. Det fulde produktblad kan ses her [Bilag\product datasheet\Scarecrow\Scarecrow - datasheet.pdf  
  
A collage of a robot

Description automatically generated](Bilag/product%20datasheet/Scarecrow/Scarecrow%20-%20datasheet.pdf)

[Figur 3 udklip af fugleskræmslets produktblad](Bilag/product%20datasheet/Scarecrow/Scarecrow%20-%20datasheet.pdf)

Muslingebur  
For at kunne håndtere automatisk tilførsel af muslinger har vi udarbejdet et 3D-printet muslingebur. Muslingeburet er udstyret med en vippe mekanisme. Det fulde produktblad kan ses her [Bilag\product datasheet\Clamcage\Clamcage - datasheet.pdf](file:///C:\Users\mk\Documents\dokumenter\skole\ZBC\ZBC%20moduler\H6\git_repos\Documentation\Rapporter\Bilag\product%20datasheet\Clamcage\clamcage%20-%20datasheet.pdf).

A collage of a muslin bur

Description automatically generated

Figur 4 Viser et udklip af muslingeburets produktblad

# Teknisk produkt dokumentation

## Deployment diagram

[Bilag\System documentation\Deployment diagram\deployment diagram.pdf](Billag/System%20documentation/Deployment%20diagram/deployment%20diagram.pdf)

A diagram of a computer

Description automatically generated

Figur 5 Deployment diagram over systemet.

# Arduino – Circuit diagrams

### Arduino muslingebur – circuit diagram

[Bilag\System documentation\Circuit diagrams\Clamcage circuit diagram.pdf](Bilag/System%20documentation/Circuit%20diagrams/Clamcage%20circuit%20diagram.pdf)

A circuit board with wires connected to it

Description automatically generated

Figur 6 Muslingebur circuit diagram

### Iltmåler circuit diagram

[Bilag\System documentation\Circuit diagrams\Oxygen measurement circuit diagram.pdf](Bilag/System%20documentation/Circuit%20diagrams/Oxygen%20measurement%20circuit%20diagram.pdf)

A diagram of a circuit board

Description automatically generated

Figur 7 Circuit diagram over iltmåleren

### Fugleskræmsel circuit diagram

[Bilag\System documentation\Circuit diagrams\Scarecrow circuit diagram.pdf](Bilag/System%20documentation/Circuit%20diagrams/Scarecrow%20circuit%20diagram.pdf)

A circuit board with wires and wires

Description automatically generated

Figur 8 Circuit diagram over fugleskræmslet

### Arduino system – circuit diagram

[Bilag\System documentation\Circuit diagrams\system circuit diagram.pdf](Bilag/System%20documentation/Circuit%20diagrams/system%20circuit%20diagram.pdf)

A diagram of a circuit board

Description automatically generated

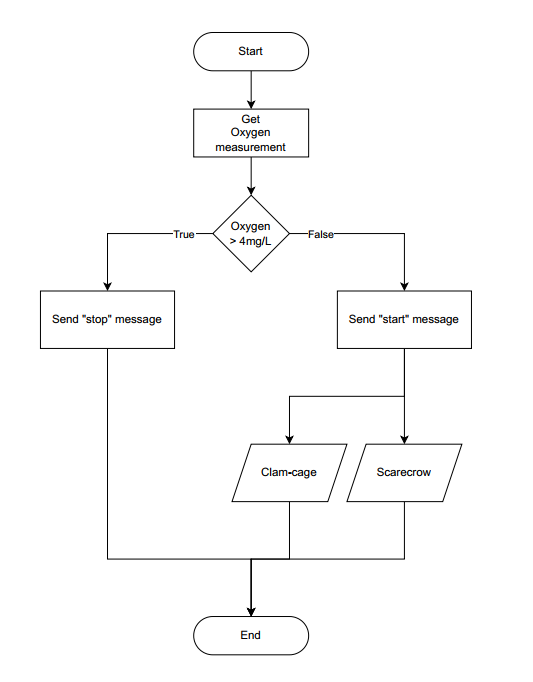
Figur 9 Viser Arduino system Circuit diagram

### 

Arduino – Flow diagram

### System – flowdiagram

[Bilag\System documentation\Flow diagrams\System flow diagram.pdf](Bilag/System%20documentation/Flow%20diagrams/System%20flow%20diagram.pdf)



Figur 10 System flowdiagram

### Iltmåler flowdiagram

[Bilag\System documentation\Flow diagrams\Oxygen Sensor flow diagram.pdf](Bilag/System%20documentation/Flow%20diagrams/Oxygen%20Sensor%20flow%20diagram.pdf)

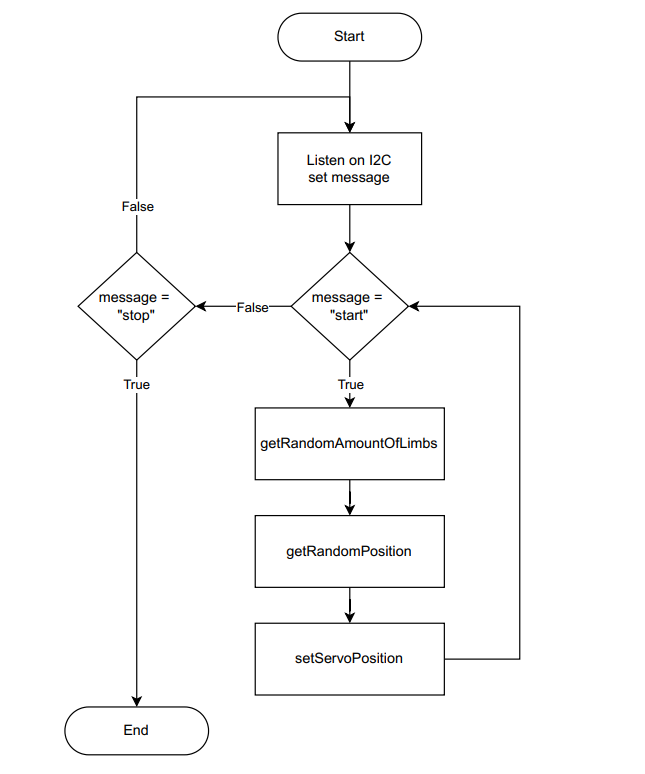
A diagram of a flowchart

Description automatically generated

Figur 11 Iltmåler flowdiagram

### Fugleskræmsel flowdiagram

[Bilag\System documentation\Flow diagrams\Scarecrow flow diagram.pdf](Bilag/System%20documentation/Flow%20diagrams/Scarecrow%20flow%20diagram.pdf)



Figur 12 fugleskræmsel flowdiagram

## SD-diagrammer

[Bilag\System documentation\SSD-diagrams\SSD - create\_new\_oxygen\_measurement.pdf](Bilag/System%20documentation/SSD-diagrams/SSD%20-%20create_new_oxygen_measurement.pdf)

A diagram of a project

Description automatically generated with medium confidence

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](Bilag/System%20documentation/Class%20diagrams/Api/Class%20diagram.png)

A diagram of a computer program

Description automatically generated with medium confidence

Figur 14 Viser klassediagrammet over Web api'et

## MailLibrary klasse diagram

[Bilag\System documentation\Class diagrams\MailLibrary\Mail library class diagram.pdf](Bilag/System%20documentation/Class%20diagrams/MailLibrary/Mail%20library%20class%20diagram.pdf)

A screenshot of a computer screen

Description automatically generated

Figur 15 Viser klassediagrammet over mail Library

# Er diagram

[Bilag\System documentation\Er diagram\ER diagram.pdf](file:///C:\Users\mk\Documents\dokumenter\skole\ZBC\ZBC%20moduler\H6\git_repos\Documentation\Rapporter\Bilag\System%20documentation\Er%20diagram\ERD%20V2.pdf)

A diagram of a function

Description automatically generated

Figur 16 viser ER diagrammet over databasen.

# Topologi

[Bilag\System documentation\Topology\Topology.pdf](Bilag/System%20documentation/Topology/Topology.pdf)

A diagram of a computer system

Description automatically generated

Figur 17 viser systemets Topologier

# 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.

**Kildekode:**

A screenshot of a computer program

Description automatically generated

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:**

* 

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

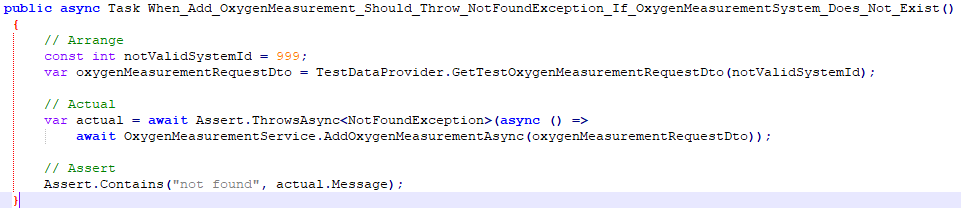
**Formål:**

* 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:**



Figur 19 Viser kildekoden for testmetoden When Add Oxygen Measurement Should Throw NotFoundException If Oxygen Meas-urement 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 fejlmeddelsen ikke indeholder ”not found”

**Faktisk resultat:**

****

## 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:**

A screenshot of a computer error

Description automatically generated

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.

**Faktisk resultat:**



**Testnavn: add oxygen measurement system should return statuscode 201**

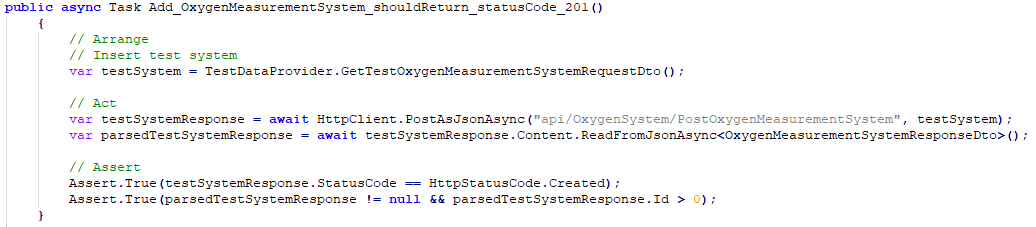
**Formål:**

* 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:**

****

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:**

****

## 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:**

**A blue screen with white text

Description automatically generated**

Figur 22 Viser displayet har 1 bogstav pr pixel.

**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.**

**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.

**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.

A screenshot of a computer

Description automatically generated

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

A blue rectangular sign with white text

Description automatically generated

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

## 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.