



## SILVER BULLET SORT

4. SEMESTERPROJEKT

---

# Kravspekifikation for Silver Bullet Sort

---

*Author:*

Gruppe 5

*Supervisor:*

Poul Ejnar ROVSING

13. september 2013

## Versionshistorie

Ver.	Dato	Initialer	Beskrivelse
0.3	06.03.2012	Gruppe 5	Skabelon til kravspecifikationen udfærdiget.
1.0	20.03.2012	Gruppe 5	Krav udfærdiget, mange stadig som brief og casual Use Cases.
2.0	15.04.2012	Gruppe 5	Alle Use Cases udfærdiget som fully dressed.
3.0	17.05.2012	Gruppe 5	Dokumentet udfærdiget til endelig aflevering.

## Godkendelsesformular

Forfattere	Projekt gruppe 5
Godkendes af	Kunde: Poul Ejnar Røvsing Leverandør Repræsentant: Lars Anker Christensen
Projektnummer	4. Semester projekt
Dokument-id	SBS_Kravspecifikation
Antal sider	31
kunde	Poul Ejnar Røvsing

Ved underskrivelse af dette dokument accepteres det af begge parter, som værende kravene til udviklingen af det ønskede system.

Sted og dato: \_\_\_\_\_

---

09421 Lasse Lindsted Sørensen

---

10063 Lasse Hansen

---

10648 Lars Anker Christensen

---

10719 Michael Bojsen-Hansen

---

10750 Kasper Vinther Andersen

---

10770 Christian Smidt-Jensen

---

10832 Christoffer Lousdahl Werge

---

PER Poul Ejnar Røvsing

---

10893 Rasmus Bækgaard

# Indhold

<b>1</b>	<b>Indledning</b>	<b>5</b>
1.1	Formål . . . . .	5
1.2	Referencer . . . . .	5
1.3	Læsevejledning . . . . .	5
<b>2</b>	<b>General beskrivelse</b>	<b>7</b>
2.1	Systembeskrivelse . . . . .	7
2.1.1	Systemoversigt . . . . .	9
2.2	Systemets funktioner . . . . .	10
2.2.1	Use Case diagram . . . . .	10
2.3	Systemets begrænsninger . . . . .	10
2.4	Brugerprofil . . . . .	11
2.5	Krav til udviklingsforløbet . . . . .	11
2.5.1	Obligatoriske udviklingsværktøjer . . . . .	11
2.5.2	Gruppedefinerede udviklingsværktøjer . . . . .	11
2.6	Forudsætninger . . . . .	12
<b>3</b>	<b>Funktionelle krav - Use Cases</b>	<b>13</b>
3.1	Use Case 1: Sorter klods . . . . .	13
3.2	Use Case 1.1: Mål og vej klods (Include) . . . . .	16
3.3	Use Case 1.2: Bestem materialetype (Include) . . . . .	18
3.4	Use Case 2: Programmer robot . . . . .	20
3.5	Use Case 3: Rediger materialetype . . . . .	24
3.6	Use Case 4: Tilgå log . . . . .	29
3.7	Use Case 5: Test program . . . . .	32
<b>4</b>	<b>Eksterne grænseflader</b>	<b>34</b>
4.1	Brugergrænseflade . . . . .	34
4.2	Hardware grænseflader . . . . .	35
<b>5</b>	<b>Kvalitetsfaktorer</b>	<b>36</b>

<b>6</b>	<b>Designkrav</b>	<b>37</b>
<b>7</b>	<b>Øvrige krav</b>	<b>38</b>
<b>8</b>	<b>Delleveringer</b>	<b>39</b>

# 1 Indledning

## 1.1 Formål

Dette dokument har til formål at opstille de krav, der skal være opfyldt, når projektet er færdiggjort. Dokumentet er blevet udformet af projektgruppe 5, der skal udfærdige produktet, og sikre at kravene i dette dokument er opfyldt.

De forskellige krav til produktet, er blevet udformet i samarbejde med kunden, som i dette tilfælde er projektgruppens eget firma. Derved er det blevet sikret, at misforståelser bliver undgået. Både kunden og projektgruppen har underskrevet dokumentet, som tegn på deres indforståelse med kravene.

Der er taget forbehold for, at ny viden vil blive indsamlet under produktudviklingen, og derfor ser projektgruppen sig berettiget til at gå tilbage og tilpasse Kravspecifikationen, så dokumentet forbliver opdateret. Der vil naturligvis KUN ske ændringer i dokumentet, hvis kunden kan acceptere ændringerne og er indforstået med følgerne af disse.

## 1.2 Referencer

Følgende dokumenter refereres der til i kravspecifikationen

- SBS\_Ide-funktioner.
- SBS\_HardwareSpec.
- Scrobot-ER 4u, Users Manual – 100343b ER 4u.

Udover disse dokumenter, er der lavet en accepttest specifikation, der står for at verificere at kravene er opfyldt.

## 1.3 Læsevejledning

Først skal det nævnes at det er valgt at kalde systemet for Silver Bullet Sort (SBS), så læser ikke er i tvivl når dette navn fremgår at kravspecifikationen. Derudover ses herunder en kort beskrivelse af de forskellige afsnit:

- General Beskrivelse

I starten af dokumentet beskrives og illustreres systemet i sin helhed, samt et overblik over systemet funktioner gives.

- Funktionelle krav

Her er alle Use Cases beskrevet fully dressed.

- Eksterne Grænseflader

Her ses en skitse over hvordan brugergrænsefladen vil komme til at se ud. Derudover er Hardwarens grænseflader beskrevet. Hardwarens grænseflader beskrives dog kun overfladisk

- Kvalitetsfaktorer og Designkrav

Her er beskrevet den kvalitet og den ydelse kunden kan forvente. Derudover kan her også læses, hvilke krav der stilles til designer, i forbindelse med implementering

- Dellevering

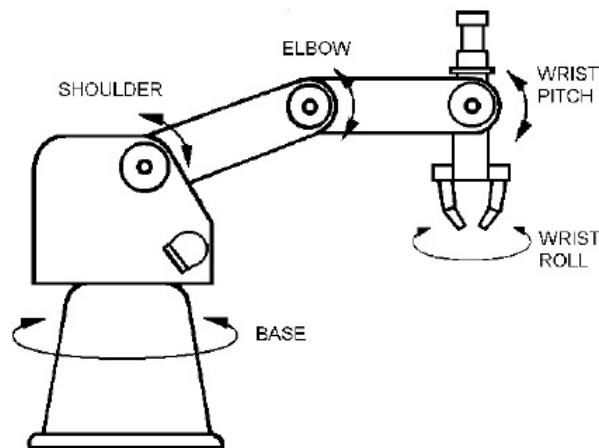
Her er beskrevet, hvilke leverancer kunden kan forvente

## 2 General beskrivelse

### 2.1 Systembeskrivelse

- Robot

Robotten har fem omdrejningsakser, så den kan manøvrere rundt i systemets omgivelser. Hver del har sin egen motor til at roterer denne del. Ydermere har robotten en klo, der kan gribe fat i elementer og samtidig måle dem. Figuren nedenfor giver et overblik over robotten.



Figur 1: Robotoversigt

- Transportbånd

Transportbåndets opgave er at transportere objekter bestående af klodser fra en feeder (se billede Figur 2: Systemoversigt) til en position, hvor robotarmen kan få fat på klodsen. Dertil er påmonteret en sensor, som stopper båndet, når klodsen er i den rette position.

- Vægt

Der er udleveret en vægtcelle, hvorudfra der skal konstrueres en fungerende vægt. Vægten bruges til at veje klodserne, så materialetypen kan bestemmes.



- STK500-kit

STK500-kittet bliver benyttet til AD-konvertering af spændingssignalet der kommer fra vægten

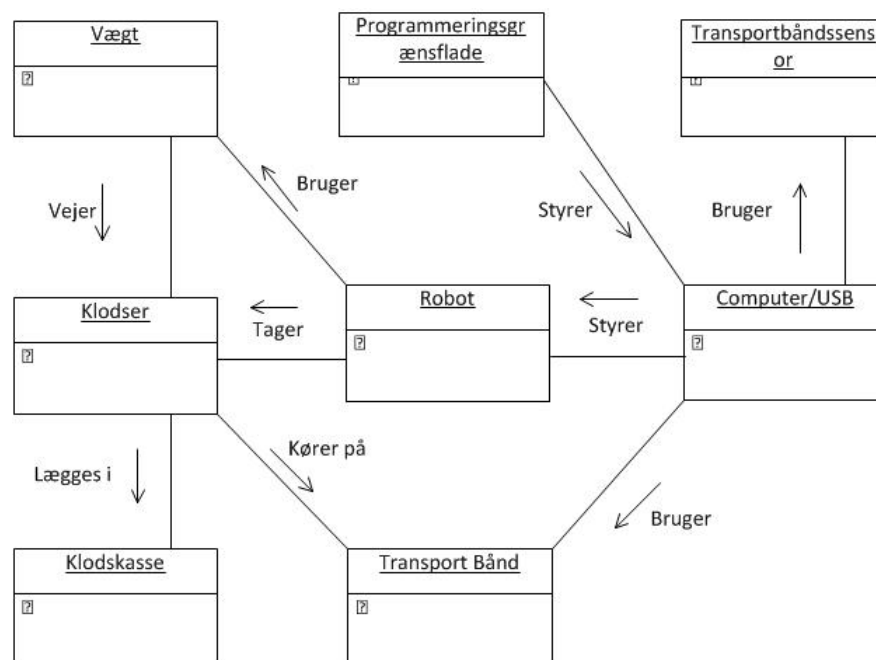
- USB-Controller og PC

For at bruge robotten, skal der gøres brug af en USB-Controlleren, som forbinder disse vha. et 60-pins interface. USB-Controller er koblet til PC'en via et USB-kabel.

- Database

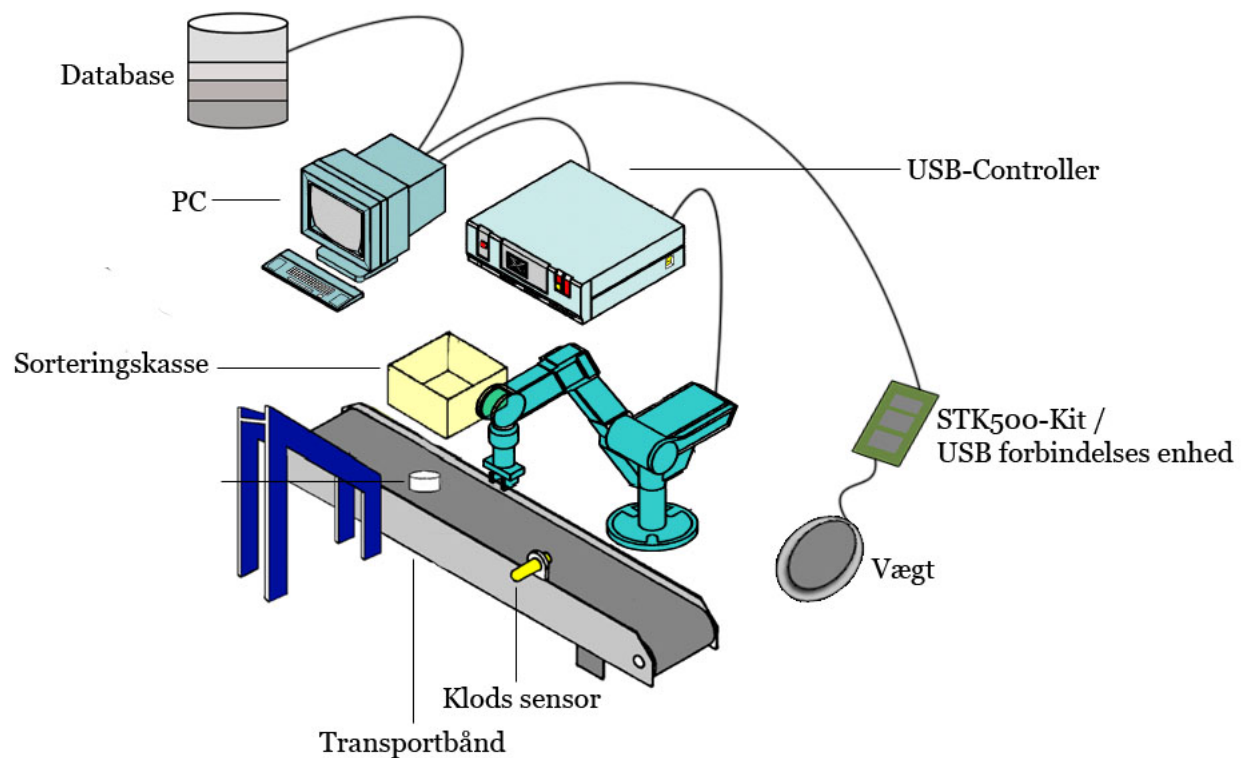
Databasen indeholder data om systemets elementer og logfiler. Det skal være muligt at tilgå databasen og ændre i bestemte elementer.

Nedenfor er vist en domænemodel, som giver et overblik over systemet.



Figur 2: Domænemodel

### 2.1.1 Systemoversigt



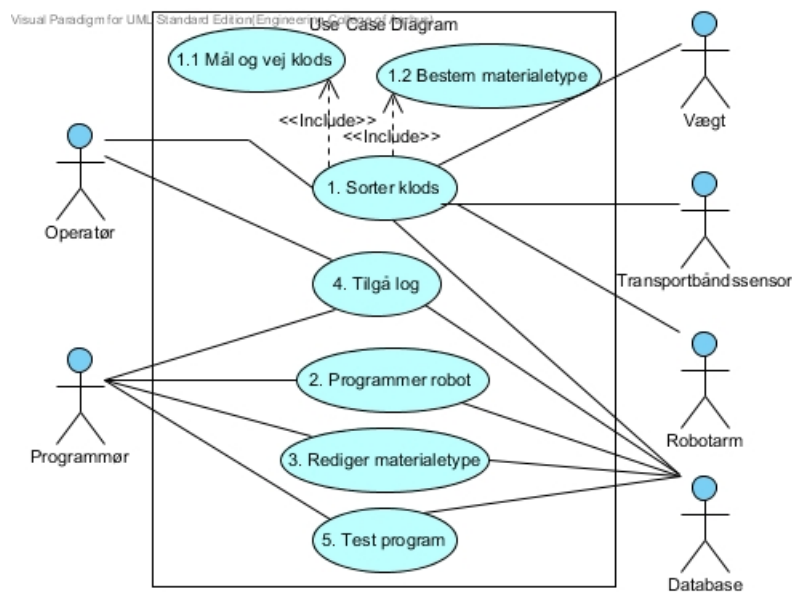
Figur 3: Systemoversigt

Ovenstående figur giver et overblik over systemet, hvor enhederne ses i sammenhæng. De forskellige enheder er beskrevet kort i ovenstående afsnit, *2.1 Systembeskrivelse*

## 2.2 Systemets funktioner

I det følgende punkt er systemets funktioner beskrevet ved Use Case-teknikken. Diagrammet nedenfor giver overblik over systemets funktioner.

### 2.2.1 Use Case diagram



Figur 4: Use Case diagram

Systemets hovedfunktion er at sortere klodser efter materialetype. Klodserne transporteres på transportbåndet, hvor en robotarm samler dem op og placerer dem i en kasse svarende til deres materialetype.

Det er muligt for programmøren at ændre i robottens program via en PC.

De forskellige Use Cases er detaljeret beskrevet senere i dokumentet, i afsnittet *3 Funktionelle krav - Use Cases*

## 2.3 Systemets begrænsninger

I produktoplægget er et kamera beskrevet. I det endelige produkt er dette erstattet af en optisk sensor, som indgår i styringen af transportbåndet. Dette er foretaget efter aftale med kunde.

Den manuelle pendant er desuden fjernet, og i stedet kan robotten programmeres via et user interface på en PC.

## 2.4 Brugerprofil

Det forventes af systemet udelukkende betjenes af fagpersonale. Der vil forekomme to brugertyper:

- Operatør

Operatøren er en primær aktør af systemet. Han skal have grundlæggende kendskab til selve systemet, men behøver ingen erfaring inden for programmering. Det er dennes opgave at starte og overvåge systemet.

- Programmør

Programmøren er en primær aktør af systemet. Han skal have god viden indenfor programmering og kendskab til systemet. Programmøren kan lave brugerdefinerede programmer til robotten. Desuden har han mulighed for at tilføje materialetyper til systemet, redigere i de eksisterende og slette klodser der er registrerede i systemet.

## 2.5 Krav til udviklingsforløbet

Selvom der ikke blev vedlagt mange obligatoriske krav til dette projekt, er der i gruppen blevet fastlagt nogle udviklingsmæssige rammer, som vil blive fulgt:

### 2.5.1 Obligatoriske udviklingsværktøjer

Programmeringssproget skal i dette projekt være C#, hvor der skal gøres brug af WPF til det grafiske interface samt .NET 4.0 frameworket. Til persistente data skal der gøres brug af SQL databaser, herunder logning og lagring af data fra klodserne. Undervejs i projektet udarbejdes to store dokumenter, en processrapport og et designdokument.

### 2.5.2 Gruppedefinerede udviklingsværktøjer

Til selve udviklingen er der blevet valgt at følge væsentlige principper fra Scrum frameworket. Der blev gjort brug af dette i tredje semesterprojektet, hvor det blev set som

ganske brugbart. Herunder blev der også gjort brug af nogle af Extreme Programming-principperne.

## 2.6 Forudsætninger

Det forudsættes, at der gennem hele projektudviklingsforløbet vil være en testrobot til rådighed, således at SW kan blive testet. Dog er dette primært en forudsætning i starten af projektet, da et testprogram, som kan simulere robotten, vil blive udviklet. Endvidere forudsættes det, at det i hele projektet vil være muligt at få løbende feedback fra kunden, så projektet udmunder i et produkt, der lever op til kundens behov og forventninger.

Softwaremæssigt kræves det, at kundens computer kører en nyere version af Windows XP, da der udvikles i WPF og .NET samtidig med, at robotten kun understøttes i Windows XP.

## 3 Funktionelle krav - Use Cases

### 3.1 Use Case 1: Sorter klods

**Mål:**

Målet med denne Use Case er at sortere en klods efter materiale, der bestemmes vha. densiteten. Densiteten findes ved at måle klodsens tre sider samt at finde vægten.

**Initiering:**

Operatøren tilkendegiver overfor systemet, at han vil starte Use Casen. Use Casen stopper igen, når han tilkendegiver at den skal stoppes eller at der ikke er plads til flere klodser af den givne materialetype.

**Aktører og interessenter:**

Primære aktører:

- Operatør.<sup>1</sup>

Sekundære aktører:

- Vægt.
- Transportbåndssensor.
- Robotarm.
- Database.

**Antal samtidige forekomster:**

En samtidig forekomst.

**Ikke funktionelle krav:**

1. Det må maksimum tage tre minutter at sortere en klods.
2. Klodsen skal placeres med en maksimum afvigelse på 2 centimeter fra det ønskede punkt.

---

<sup>1</sup>Programmøren kan også tilgå Use Casen, men når det er tilfældet, opfattes han som operatør.

3. Der skal minimum være 4 cm, imellem klodserne på transportbåndet.

#### Referencer:

I denne Use Case anvendes følgende include Use Cases:

- *Use Case 1.1: Mål og vej klods.*
- *Use Case 1.2: Bestem materialetype.*

#### Startbetingelser:

Robotarmen skal være i startposition samt, desuden må den ikke have fat i en klods.

#### Slutresultat ved succes:

Klodsens er placeret i det rum, hvori klodsens materialetype tilhører.

#### Slutresultat ved undtagelser:

Klodsens er ikke blevet lagt i et rum, hvori materialetypen ikke tilhører.

Operatøren har modtaget en fejlmeddelelse, der informerer om at sorteringen ikke er fuldført.

#### Normalforløb:<sup>2</sup>

1. Transportbåndet startes.
2. En klods registreres. Transportbåndet stoppes.
3. Klodsens måles: *Include 1.1 Mål og vej klods.*
4. Klodsens densitet udregnes: *Include 1.2 Bestem materialetype.*
5. Klodsens flyttes over i kassen, i det respektive rum for materialetypen.
6. Robotarmen går tilbage til startposition, ved transportbåndet.

#### Undtagelser:<sup>3</sup>

Undtagelse \* a: Der trykkes på den fysiske nødstop-knap.

---

<sup>2</sup>Efter hvert punkt skabes der et loggingindlæg til databasen, der specificerer hvad der er sket.

<sup>3</sup>Stjernen indikerer, at undtagelsen gør sig gældende for alle normalforløb (også include Use Cases).

1. Sorteringsmekanismen stoppes.

Undtagelse \* b: Operatøren vælger at stoppe systemet igennem brugergrænsefladen.

1. Sorteringsmekanismen stoppes.

Undtagelse \* c: Operatøren vælger at pause systemet igennem brugergrænsefladen.

1. Sorteringsmekanismen stoppes midlertidigt, indtil systemet igen startes igennem brugergrænsefladen.

Undtagelse \* d: Der mistes forbindelse til databasen.

1. Systemet sørger for at gemme data/loggingindlæg og skrive dem til databasen, når forbindelsen er blevet genoprettet.

Undtagelse 5 a: Der er ikke plads til klodsen i boksen for den til klodsen tilhørende materialetype.

1. Operatøren modtager en besked om, at der ikke er mere plads i kassen.
2. Robotarmen placere klodsen til højre for sensoren, så den kasseres, næste gang sorteringen startes.



## 3.2 Use Case 1.1: Mål og vej klods (Include)

### Mål:

Målet med denne Use Case er at måle og veje en given klods.

### Initiering:

Use Casen initieres gennem *Use Case 1: Sorter Klods*.

### Aktører og interessenter:

Primære aktører:

- *Use Case 1: Sorter klods*.

Sekundære aktører:

- Robotarm.
- Vægt.
- Transportbånd.
- Database.

### Antal samtidige forekomster:

En samtidig forekomst.

### Ikke funktionelle krav:

- Klodsen skal måles med en præcision på  $\pm 2$  mm.
- Klodsen skal vejes med en præcision på  $\pm 3$  gram.

### Referencer:

Denne Use Case er i forlængelse af *Use Case 1: Sorter klods*.

### Startbetingelser:

Startbetingelserne gør sig gældende i at *Use Case 1: Sorter klods* skal være igangsat.

**Slutresultat ved succes:**

Klodsens mål og masse er fundet.

**Slutresultat ved undtagelser:**

Klodsens mål og masse er ikke blevet fundet.

**Normalforløb:<sup>4</sup>**

1. Robotarmen tager fat i klodsen
2. Klodsens første side måles.
3. Klodsen slippes.
4. Robotarmen løftes og roteres 90 graders.
5. Robotarmen sænkes og tager fat i klodsen på den anden led.
6. Klodsens anden side måles.
7. Robotarmen vender klodsen.
8. Klodsen slippes på vægten.
9. Klodsen vejes på vægten.
10. Robotarmen tager fat i klodsen på sidste led.
11. Klodsens sidste side måles.

---

<sup>4</sup>Efter hvert punkt skabes der et loggingindlæg til databasen, der specificerer hvad der er sket.

### 3.3 Use Case 1.2: Bestem materialetype (Include)

**Mål:**

Målet med denne Use Case er at bestemme klodsens materialetype vha. densiteten, som beregnes ud fra de målte sider samt vægten.

**Initiering:**

Use Casen initieres gennem *Use Case 1: Sorter Klods*.

**Aktører og interessenter:**

Primære aktører:

- *Use Case 1: Sorter klods*.

Sekundære aktører:

- Database.

**Antal samtidige forekomster:**

En samtidig forekomst.

**Ikke funktionelle krav:**

- Klodsens densitet gemmes i databasen i med enheden  $\frac{g}{cm^3}$ .
- Klodsens målte værdi gemmes i databasen i cm.
- Klodsens masse gemmes i databasen i gram.

**Referencer:**

Denne Use Case er udvidelse af *Use Case 1: Sorter klods*.

**Startbetingelser:**

Startbetingelserne gør sig gældende i at *Use Case 1: Sorter klods* skal være igangsat. Det forudsættes også at *Include Use Case 1.1 Mål og vej klods* er eksekveret forinden.

**Slutresultat ved succes:**

Klodsens materialetype er bestemt og gemt.

**Slutresultat ved undtagelser:**

Klodsens materialetype blev ikke bestemt eller gemt.

**Normalforløb:<sup>5</sup>**

1. Klodsens rumfang udregnes.
2. Ud fra klodsens rumfang og masse udregnes densiteten.
3. Den udregnede densitet sammenlignes med liste af eksisterende materialetyper på databasen, og deraf bestemmes klodsens type.
4. Klodsens data persisteres på databasen.

Undtagelse 3 a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver operatøren informeret om, at forbindelsen til databasen er blevet afbrudt.
2. Operatøren bliver spurgt, om han vil vente på, at forbindelsen bliver genetableret.
  - i. Operatøren tilkendegiver overfor systemet, at dette ønskes.
    - A. Systemet forsøger at genetablere forbindelsen til databasen.
    - B. Hvis forbindelsen ikke kan genoprettes, præsenteres operatøren for punkt 2.
  - ii. Operatøren tilkendegiver overfor systemet, at dette ikke ønskes.
    - A. Programmet lukker ned.

Undtagelse 3 b: Densitet kan ikke pares med en materialetype på databasen

1. Operatøren modtager en besked om, at materialetypen ikke kunne findes på databasen.
2. Robotarmen placere klodsens til højre for sensoren, så den kasseres, næste gang sorteringen startes.

---

<sup>5</sup>Efter hvert punkt skabes der et loggingindlæg til databasen, der specificerer hvad der er sket.

## 3.4 Use Case 2: Programmer robot

### Mål:

Målet med denne Use Case er at omprogrammere robotten til at tjene et nyt formål eller udføre samme opgave på en ny måde samt at anvende selve programmet.

### Initiering:

Programmøren tilkendegiver over for systemet, at han ønsker at ændre i programmer eller anvende et program.

### Aktører og interessenter:

Primære aktører:

- Programmør.

Sekundære aktører:

- Ingen.

### Antal samtidige forekomster:

En samtidig forekomst.

### Ikke funktionelle krav:

- Ingen.

### Referencer:

SBS\_Ide-funktioner.pdf.<sup>6</sup>

### Startbetingelser:

Brugeren skal være logget ind som 'programmør'.<sup>7</sup>

### Slutresultat ved succes:

Programmøren har færdiggjort eller ændret i et program, og programmet/ændringerne er

---

<sup>6</sup>Dokumentet beskriver de funktioner programmøren kan bruge til at programmere robotten

<sup>7</sup>Gælder ikke for normalforløb 4, da operatøren også kan anvende et program.

blevet gemt. Hvis et program ønskes anvendt, er dette blevet anvendt.

#### **Slutresultat ved undtagelser:**

Programmet eller ændringerne er ikke blevet gemt og programmet er ikke blevet anvendt.

#### **Normalforløb 1 - Opret nyt program:**

1. Programmøren tilkender overfor systemet, at han vil redigere i programmer.
2. Systemet åbner faciliteter for at kunne ændre styringen til robotten.
3. Programmøren laver den nye styring.
4. Programmøren tilkendegiver over for systemet, at programmet skal gemmes.
5. Systemet gemmer programmet.

#### **Undtagelser 1 - Opret nyt program:**

Undtagelse 3 a: Programmøren afslutter program uden at gemme.

- (a) Systemet spørg programmøren, om han er sikker på, at programmet skal lukkes uden at gemmes.

Undtagelse 4 a: Programmøren har ikke navngivet programmet.

- (a) Systemet tilkendegiver over for programmøren at programmet skal gemmes med et navn.
- (b) Programmøren giver navn for program.

#### **Normalforløb 2 - Rediger gammelt program:**

1. Programmøren tilkender overfor systemet, at han vil redigere i programmer.
2. Systemet lister gamle programmer, der er tilgængelige.
3. Programmøren tilkendegiver, hvilket program fra listen denne ønsker at redigere.
4. Systemet åbner faciliteter for at kunne ændre styringen til robotten.

5. Programmøren redigerer den gamle styring.
6. Programmøren tilkendegiver over for systemet, at denne skal gemme programmet.
7. Systemet gemmer programmet.

#### **Undtagelser 2 - Rediger gammelt program:**

Undtagelse 2 a: Der er ingen gamle programmer tilgængelige.

- (a) Der fremkommer ingen gamle programmer.

Undtagelse 3 a: Programmøren vælger standardprogrammet.

- (a) Programmøren modtager en besked om, at det ikke er muligt at redigere i standardprogrammet.

Undtagelse 7 a: Programmøren afslutter program uden at gemme.

- (a) Systemet giver programmøren en meddelelse, om han ønsker at afslutte uden at gemme.

#### **Normalforløb 3 - Slet program:**

1. Programmøren tilkender overfor systemet, at han vil redigere i programmer.
2. Systemet lister gamle programmer, der er tilgængelige.
3. Programmøren tilkendegiver overfor systemet, hvilket program der ønskes slettet.
4. Systemet spørger programmøren, om handlingen ønskes udført.
5. Programmøren tilkendegiver overfor systemet, at dette ønskes.
6. Systemet sletter programmet.

#### **Undtagelser 3 - Slet program:**

Undtagelse 2 a: Der er ingen gamle programmer tilgængelige.

- (a) Der fremkommer ingen gamle programmer.

Undtagelse 3 a: Programmøren har valgt standardprogrammet.

- (a) Systemet giver en besked om, at det ikke er muligt at slette standardprogrammet.

Undtagelse 5 a: Programmøren tilkendegiver overfor systemet, at programmet ikke ønskes slettet alligevel.

- (a) Systemet sletter ikke programmet.

#### **Normalforløb 4 - Anvend program:**

1. Programmøren tilkendegiver over for systemet, at han vil anvende et program.
2. Systemet lister programmer, der er tilgængelige.
3. Programmøren tilkendegiver over for systemet, hvilket program fra listen denne ønsker at anvende.
4. Det valgte program anvendes.

#### **Undtagelser 4 - Indlæs gammelt program:**

Undtagelse 2 a: Der er ingen gamle programmer tilgængelige.

- (a) Programmøren kan kun vælge standardprogrammet.



### 3.5 Use Case 3: Rediger materialetype

**Mål:**

Målet med denne Use Case er, at tilføje, ændre eller fjerne en materialetype.

**Initiering:**

Use Casen initieres, når programmøren tilkendegiver overfor systemet, at han vil redigere materialetyper.

**Aktører og interessenter:**

Primære aktører:

- Programmør.

Sekundære aktører:

- Databasen.

**Antal samtidige forekomster:**

En samtidig forekomst.

**Ikke funktionelle krav:**

- Densiteten skal være større end  $0 \frac{g}{cm^3}$  og mindre end  $23 \frac{g}{cm^3}$ .

**Referencer:**

Ingen.

**Startbetingelser:**

Programmøren skal være logget ind som 'programmør' for at have rettigheder til at redigere materialetyper. Ligeledes skal der være forbindelse til databasen.

**Slutresultat ved succes:**

Hvis det ønskes at oprette en ny materialetype, er denne blevet oprettet og gemt.

Hvis det ønskes at fjerne en materialetype, er denne blevet fjernet.

Hvis det ønskes at redigere i en materialetype, er ændringerne blevet gemt.

**Slutresultat ved undtagelser:**

Hvis det ønskes at oprette en ny materialetype, er ikke blevet oprettet eller gemt.

Hvis det ønskes at fjerne en materialetype, er denne ikke blevet fjernet.

Hvis det ønskes at redigere i en materialetype, er ændringerne ikke blevet gemt.

**Normalforløb 1 - Tilføj ny materialetype:**

1. Programmøren tilkendegiver overfor systemet, at denne ønsker at tilføje en ny materialetype.
2. Systemet gør det muligt at indtaste data:
  - Navn
  - Densitet
3. Programmøren indtaster data.
4. Programmøren tilkendegiver overfor systemet, at denne ønsker at gemme de indtastede data.
5. Systemet gemmer de indtastede data.

**Undtagelser 1 - Tilføj ny materialetype:**

Undtagelse \* a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver programmøren informeret om, at forbindelsen til databasen er blevet afbrudt.
2. Programmøren bliver spurgt, om han vil vente på, at forbindelsen bliver genetableret.
  - i. Programmøren tilkendegiver overfor systemet, at dette ønskes.
    - A. Systemet forsøger at genetablere forbindelsen til databasen.
    - B. Hvis forbindelsen ikke kan genoprettes, præsenteres programmøren for punkt 2.

- ii. Programmøren tilkendegiver overfor systemet, at dette ikke ønskes.

A. Programmet lukker ned.

Undtagelse 2 a: Densiteten er ikke gyldig.

- (a) Systemet giver besked om, at der er indtastet en ugyldig værdi.
- (b) Programmøren indtaster en ny værdi.

Undtagelse 3 a: Alle data er ikke udfyldt.

- (a) Systemet giver besked om, at ikke alle datafelter er udfyldt.
- (b) Programmøren udfylder de datafelter der ikke er udfyldt.

Undtagelse 3-4 a: Programmøren annullerer tilføjelsen af en ny materialetype.

- (a) Den nye materialetype gemmes ikke.

#### **Normalforløb 2 - Fjern materialetype:**

1. Programmøren tilkendegiver overfor systemet, at denne ønsker at fjerne en materialetype.
2. Systemet lister alle mulige materialetyper i systemet.
3. Programmøren tilkendegiver overfor systemet, hvilken type denne ønsker at fjerne.
4. Systemet spørger om bekræftelse på, at den markerede type ønskes slettet.
5. Programmøren bekræfter sit valgt.
6. Systemet fjerner materialetypen.

#### **Undtagelser 2 - Fjern materialetype:**

Undtagelse \* a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver programmøren informeret om, at forbindelsen til databasen er blevet afbrudt.

2. Programmøren bliver spurgt, om han vil vente på, at forbindelsen bliver genetableret.

i. Programmøren tilkendegiver overfor systemet, at dette ønskes.

A. Systemet forsøger at genetablere forbindelsen til databasen.

B. Hvis forbindelsen ikke kan genoprettes, præsenteres programmøren for punkt 2.

ii. Programmøren tilkendegiver overfor systemet, at dette ikke ønskes.

A. Programmet lukker ned.

Undtagelse 3 a: Programmøren tilkendegiver vælger en af standardmaterialerne.<sup>8</sup>

(a) Programmøren modtager en besked om, at materialetypen ikke kan slettes.

Undtagelse 5 a: Programmøren tilkendegiver overfor systemet, at det ikke var denne materialetype, der skulle slettes,

(a) Systemet ændrer intet og viser listen igen.

#### **Normalforløb 3 - Ændre data for materialetype:**

1. Programmøren tilkendegiver overfor systemet, at han vil ændre data for en given materialetype.

2. Systemet lister alle mulige materialetyper i systemet.

3. Programmøren vælger en materialetype på listen.

4. Systemet gør det muligt at ændre data for den givne materialetype.

5. Programmøren indtaster data, han ønsker ændret.

6. Systemet gemmer de indtastede data.

#### **Undtagelser 3 - Ændre data for materialetype:**

---

<sup>8</sup>Materialetyper tilhørende de fem klodser, der er i systemet

Undtagelse \* a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver programmøren informeret om, at forbindelsen til databasen er blevet afbrudt.
2. Programmøren bliver spurgt, om han vil vente på, at forbindelsen bliver genetableret.
  - i. Programmøren tilkendegiver overfor systemet, at dette ønskes.
    - A. Systemet forsøger at genetablere forbindelsen til databasen.
    - B. Hvis forbindelsen ikke kan genoprettes, præsenteres programmøren for punkt 2.
  - ii. Programmøren tilkendegiver overfor systemet, at dette ikke ønskes.
    - A. Programmet lukker ned.

Undtagelse 5 a: Programmøren indtaster ugyldig data.

- (a) Systemet giver besked om, at der er indtastet en ugyldig værdi.
- (b) Programmøren bliver bedt om at indtaste en ny værdi.

Undtagelse 5 b: Programmøren annullerer redigeringen i det eksisterende materiale.

- (a) De nye data for materialetypen gemmes ikke.

## 3.6 Use Case 4: Tilgå log

### Mål:

Målet med denne Use Case er at tilgå en logfil og få denne vist.

### Initiering:

Use Casen initieres ved at operatøren/programmøren<sup>9</sup> tilkendegiver overfor systemet, at han vil tilgå logfiler.

### Aktører og interessenter:

Primære aktører:

- Operatør.
- Programmør.

Sekundære aktører:

- Databasen.

### Antal samtidige forekomster:

En samtidig forekomst.

### Ikke funktionelle krav:

- Loggen udskrives med tid [hh.mm.ss] eks. [15:02:42].

### Referencer:

Ingen.

### Startbetingelser:

Der skal være forbindelse til databasen, så logfiler kan hentes.

### Slutresultat ved succes:

Logfiler er blevet hentet og gemt.

---

<sup>9</sup>Da det både er programmøren og operatøren der kan anvende Use Casen betegnes de samlet som bruger.

#### **Slutresultat ved undtagelser:**

Logfiler er ikke blevet hentet eller gemt.

#### **Normalforløb**

1. Brugeren tilkendegiver overfor systemet, at denne ønsker at hente en logfil.
2. Systemet lister alle tilgængelige logfiler.
3. Brugeren tilkendegiver overfor systemet, hvilken logfil der skal vises.
4. Systemet viser data for den givne logfil.
5. Brugeren vælger at gemme logfilen.
6. Systemet åbner et vindue, hvor brugeren kan vælge, hvor logfilen skal gemmes.
7. Brugeren tilkendegiver overfor systemet, hvor logfilen skal gemmes, samt med hvilket navn den skal gemmes med.
8. Systemet gemmer logfilen på computeren.

#### **Undtagelser**

Undtagelse \* a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver brugeren informeret om, at forbindelsen til databasen er blevet afbrudt.
2. Brugeren bliver spurgt, om han vil vente på, at forbindelsen bliver genetableret.
  - i. Brugeren tilkendegiver overfor systemet, at dette ønskes.
    - A. Systemet forsøger at genetablere forbindelsen til databasen.
    - B. Hvis forbindelsen ikke kan genoprettes, præsenteres brugeren for punkt 2.
  - ii. Brugeren tilkendegiver overfor systemet, at dette ikke ønskes.
    - A. Programmet lukker ned.

Undtagelse 2 a: Der er ikke nogle logfiler i listen.

- (a) Systemet viser ingen data fra logfilerne.

Undtagelse 5 a: Brugeren vil ikke gemme logfilen, men blot have den vist.

- (a) Brugeren tilkendegiver ikke overfor systemet, at han vil gemme logfilen.

Undtagelse 6 a: Brugeren tilkendegiver overfor systemet, at han ikke ønsker at gemme filen alligevel.

- (a) Systemet lukker vinduet til, at gemme placering for filen.

Undtagelse 7 a: Filnavnet er allerede optaget.

- (a) Systemet spørger brugeren om denne ønsker at overskrive den eksisterende fil.
  - i. Brugeren tilkendegiver overfor systemet at dette ønskes.
    - A. Filen overskrives med den nye.
  - ii. Brugeren tilkendegiver overfor systemet, at dette ikke ønskes.
    - A. Brugeren får lov til at indtaste et nyt navn på filen.



### 3.7 Use Case 5: Test program

**Mål:**

Målet med denne Use Case er at teste et færdigt program eller en programsekvens via en simulering af robotten. Det giver programmøren mulighed for at teste sit program uden brug af robotten.

**Initiering:**

Programmøren starter tilkendegiver overfor systemet, at han vil starte en simulering.

**Aktører og interessenter:**

Primære aktører:

- Programmøren.

Sekundære aktører:

- Ingen.

**Antal samtidige forekomster:**

En samtidig forekomst.

**Ikke funktionelle krav:**

- Simuleringsbeskeder udskrives med et tidspunkt for begivenheden sammen med informationer om selve handlingen.

**Referencer:**

Ingen.

**Startbetingelser:**

Det forudsættes, at brugeren er logget ind som programmør.

**Slutresultat ved succes:**

Programmøren får fuldendt en testsimulering af programmet.

**Slutresultat ved undtagelser:**

Programmøren får ikke fuldendt en testsimulering af programmet

**Normalforløb:**

1. Programmøren tilkendegiver over for systemet at vedkommende ønsker en simulering.
2. Systemet indlæser tilgængelige programmer.
3. Programmøren vælger en af programmerne.
4. Programmøren starter simuleringen af programmet.
5. Alle systemets informationer fra programmet gemmes i en simuleringslogfil.
6. Simulationen afsluttes.

**Undtagelser:**

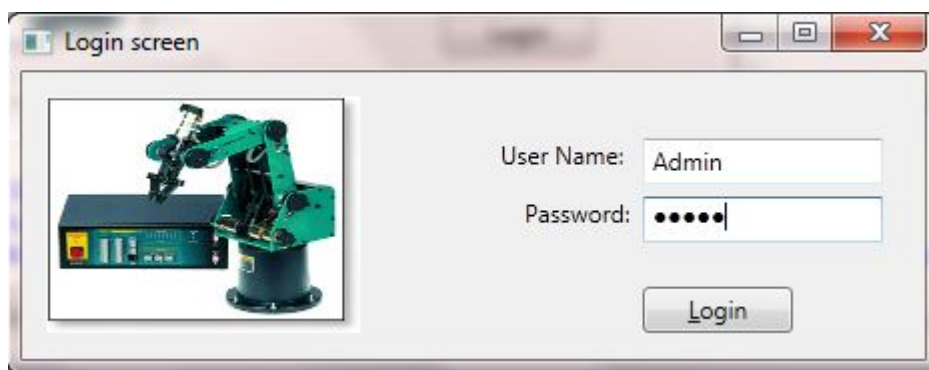
Undtagelse \* a: Programmøren trykker på stop.

1. Simuleringen stoppes.

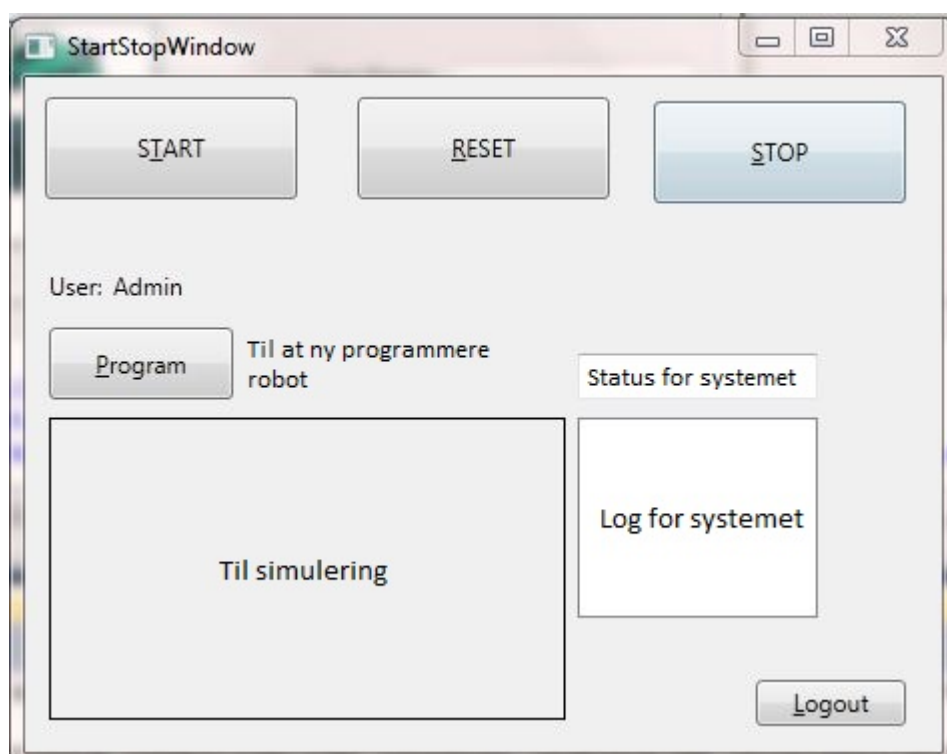
## 4 Eksterne grænseflader

### 4.1 Brugergrænseflade

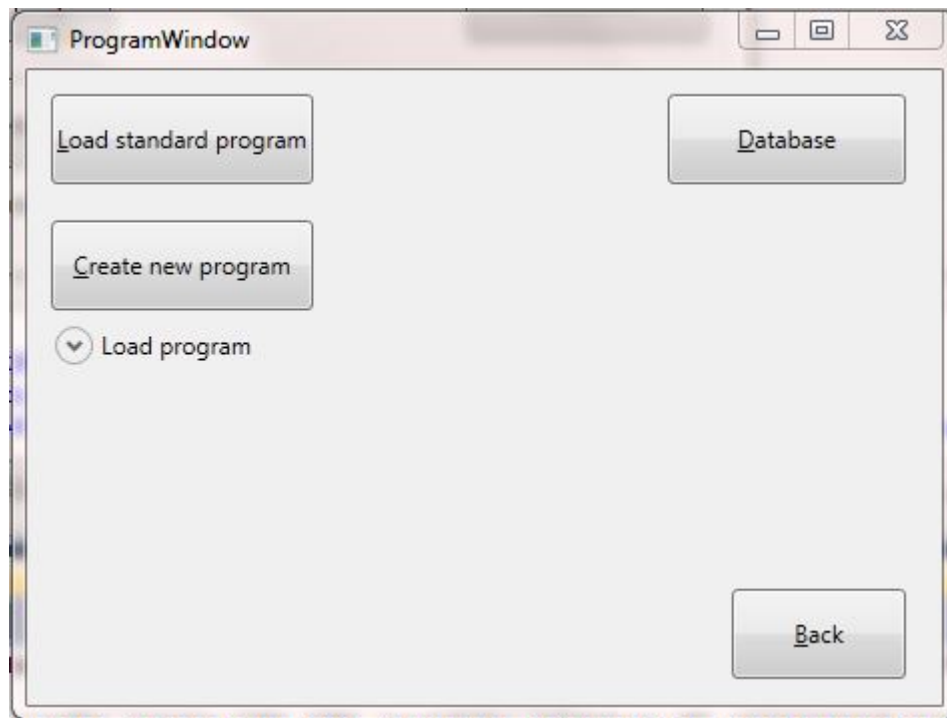
Der er blevet udformet en skitse over brugergrænsefladen, så kunden kan få en idé om, hvordan det grafiske vil komme til at se ud. Det er vigtigt at understrege, at det er en skitse, og at den endelige brugergrænseflade ikke nødvendigvis vil ligne denne fuldstændigt:



Figur 5: Loginvindue



Figur 6: Kørselsvindue



Figur 7: Programmeringsvindue

## 4.2 Hardware grænseflader

Hardware-specifikationerne kan ses i dokumentet "Hardwarespec". I dette dokument er der beskrevet hvordan de forskellige hardware-enheder er forbundet og grænsefladen mellem dem.

## 5 Kvalitetsfaktorer

Herunder er opstillet nogle kvalitetsfaktorer. Hver kvalitetsfaktorer har fået en værdi ud fra følgende skala:

**1: Ubetydelig   2: Ikke særlig vigtig   3: Vigtig   4: Meget vigtig   5: Særlig vigtig**

- **Pålidelighed: 5**

Det vigtigste er at systemet, virker og udfører det den er sat til. Det skal ske på en pålidelig måde, hvor man kan regne med at robotten udfører det den er sat til, og giver besked, hvis fejl skulle være opstået.

- **Effektivitet: 2**

Pålideligheden er sat over effektiviteten, og hurtighed og lignende ses derfor ikke som særlig vigtig.

- **Udvidelsesvenlighed: 4**

Udvidelsesvenlighed ses også som meget vigtigt, da robotten skal kunne omprogrameres til at virke i forskellige omgivelser.

- **Brugervenlighed: 3**

Dette er en vigtig faktorer, men ikke altafgørende for om systemet vil fungerer, da det også kommer an på, hvor godt operatøren og programøren bliver sat ind i systemet

- **Vedligeholdelse og genbrugbarhed: 5**

Dette er helt klart noget der vil bliver bestræbt, da det disse bliver opnået via en god objektorienteret programmeringsstil.

## 6 Designkrav

- Systemet implementeres i det objektorienteret programmeringssprog, C#.
- Funktionsstrukturen i IDE'en skal minde om den, der er i det udleverede Scorbaseret program.
- Robotten skal programmeres via det udleverede bibliotek USBC.dll.

## 7 Øvrige krav

Det kræves, at vægten er tilsluttet COM-port 20 på PC'en

## 8 Delleveringer

Der har ikke være nogen krav om delleveringer fra kundens side. Derfor har det været op til leverandøren selv at bestemme antal delleveringer. Eftersom der arbejdes efter scrumprincipperne, har gruppen valgt at præsentere udbyttet af hvert sprint som en deliverance. Sprintene har en varighed af 14 dage, så dette vil resultere i 10 små delleveringer, som så udmunder sig i det samlede produkt. Grundet at udviklingen ikke foregår over en 37-timers arbejdsuge, som normalt er tilfældet, vil den øgede funktionalitet i hver deliverance være begrænset.

Den øgede funktionalitet vil primært bestå af tilføjelse af funktionalitet til systemet, men det kan ligeledes være stabilisering og optimering af systemet. Systemets funktionalitet vil være meget begrænsede i de første delleveringer, da der skal udarbejdes nogle retningslinjer for produktet i samarbejde med kunden, før produktet kan udvikles til kundens behov.