

# Kravspecifikation

Silver Bullet Sort

Gruppe 5

## Medlemmer

Studienummer	Navn	Underskrift
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	
10893	Rasmus Bækgaard	

## Versionshistorie

Ver.	Dato	Initialer	Beskrivelse
1.00	06.03.2012	Grp5	Første version af kravspecifikationen, færdiggjort med alle kravene udspecificeret

## Godkendelsesformular

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

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

Sted og dato:

Kundeunderskrift

Leverandrunderskrift

# Indhold

<b>1</b>	<b>Indledning</b>	<b>4</b>
1.1	Formål . . . . .	4
1.2	Referencer . . . . .	4
1.3	Læsevejledning . . . . .	4
<b>2</b>	<b>General beskrivelse</b>	<b>6</b>
2.1	Systembeskrivelse . . . . .	6
2.1.1	Systemoversigt . . . . .	8
2.1.2	Aktør kontekst-diagram . . . . .	8
2.1.3	Aktør beskrivelse . . . . .	9
2.2	Systemets funktioner . . . . .	10
2.2.1	Use Case diagram . . . . .	10
2.3	Systemets begrænsninger . . . . .	11
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 . . . . .	11
<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) . . . . .	15
3.3	Use Case 1.2: Bestem materialetype (Include) . . . . .	17
3.4	Use Case 2: Programmer robot . . . . .	19
3.5	Use Case 2.1 - Manipuler data (Extend) . . . . .	22
3.6	Use Case 3: Test program . . . . .	25
<b>4</b>	<b>Eksterne grænseflader</b>	<b>27</b>
4.1	Brugergrænseflade . . . . .	27
4.2	Hardware grænseflader . . . . .	28
<b>5</b>	<b>Kvalitetsfaktorer</b>	<b>29</b>
<b>6</b>	<b>Designkrav</b>	<b>30</b>
<b>7</b>	<b>Delleveringer</b>	<b>31</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\_HardwareSpec.
- Scrobot-ER 4u, Users Manual – 100343b ER 4u

Udover disse dokumenter, er der lavet en accepttest specifikation, der stå for at verificerer at kravene er opfyldt.

## 1.3 Læsevejledning

Først skal det nævnes at vi har valgt at kalde vores system 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

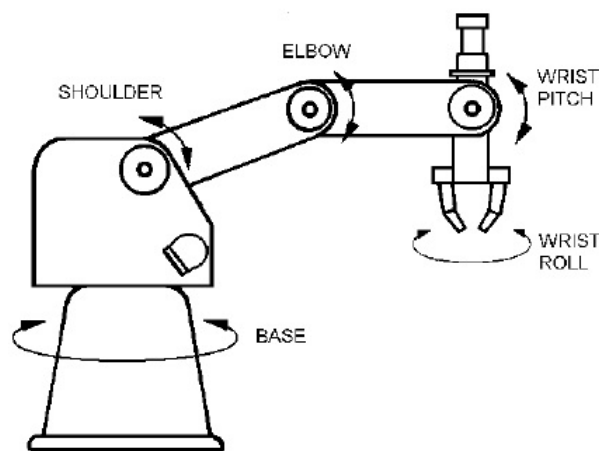
## 2 General beskrivelse

### 2.1 Systembeskrivelse

Systemet er sammensat af seks dele, som er beskrevet nedenfor. Information omkring delene er fundet i dokumentet: *Scorbot-ER 4u, Users Manual – 100343-b ER\_4u*

- 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

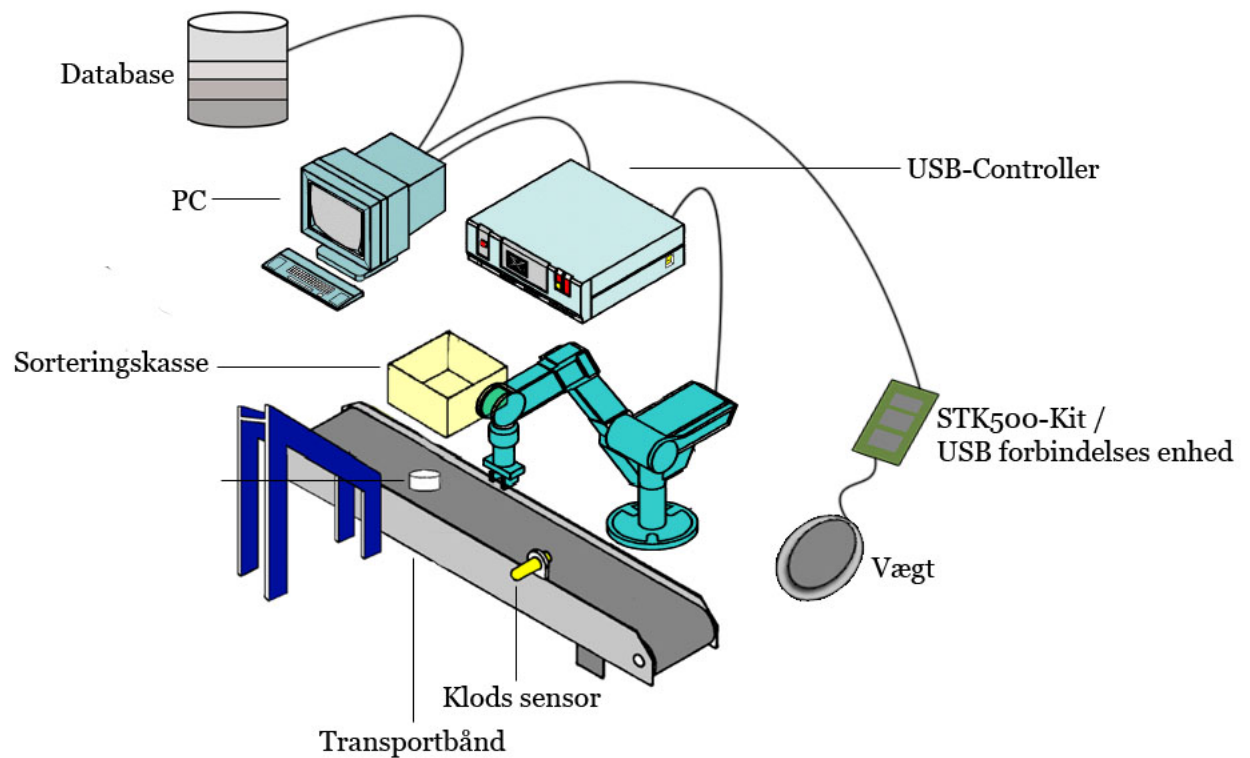
- USB-Controller og PC

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.

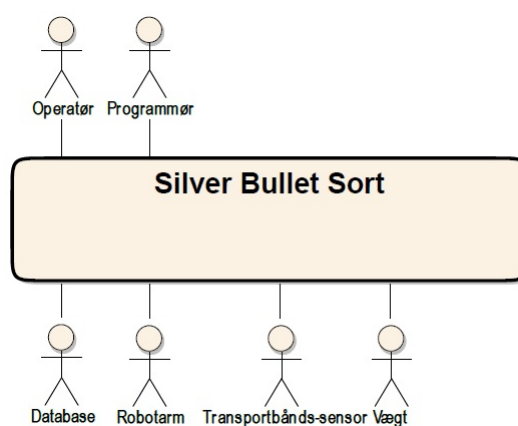
### 2.1.1 Systemoversigt



Figur 2: 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*

### 2.1.2 Aktør kontekst-diagram



Figur 3: Aktør kontekst-diagram

Ovenstående figur viser hvilke aktører, der interagerer med sorteringssystemet. Videre beskrivelse af disse aktører, findes i følgende afsnit *2.1.3 Aktørbeskrivelser*



### 2.1.3 Aktør beskrivelse

#### Primære aktører:

<b>Aktør navn</b>	Operatør
<b>Beskrivelse</b>	Denne aktør starter og stopper systemet. Målet for aktøren er at få sorteret klodser, alt efter deres materiale-type.

<b>Antal samtidige aktører</b>	1
--------------------------------	---

<b>Aktør navn</b>	Programmør.
<b>Beskrivelse</b>	Denne aktørs opgave er at lave brugerdefinerede programmer til systemet.

<b>Antal samtidige aktører</b>	1
--------------------------------	---

#### Sekundære aktører

<b>Aktør navn</b>	Vægt
<b>Beskrivelse</b>	Denne aktør vejer et objekt, så systemet kan bestemme dets materialetype.

<b>Antal samtidige aktører</b>	1
--------------------------------	---

<b>Aktør navn</b>	Transportbåndssensor
<b>Beskrivelse</b>	Denne aktør registrerer, hvornår et objekt er klar til at blive samlet op af robotten.

<b>Antal samtidige aktører</b>	1
--------------------------------	---

<b>Aktør navn</b>	Robotarm.
<b>Beskrivelse</b>	Denne aktørs opgave er at flytte og måle forskellige typer klodser.

<b>Antal samtidige aktører</b>	1
--------------------------------	---

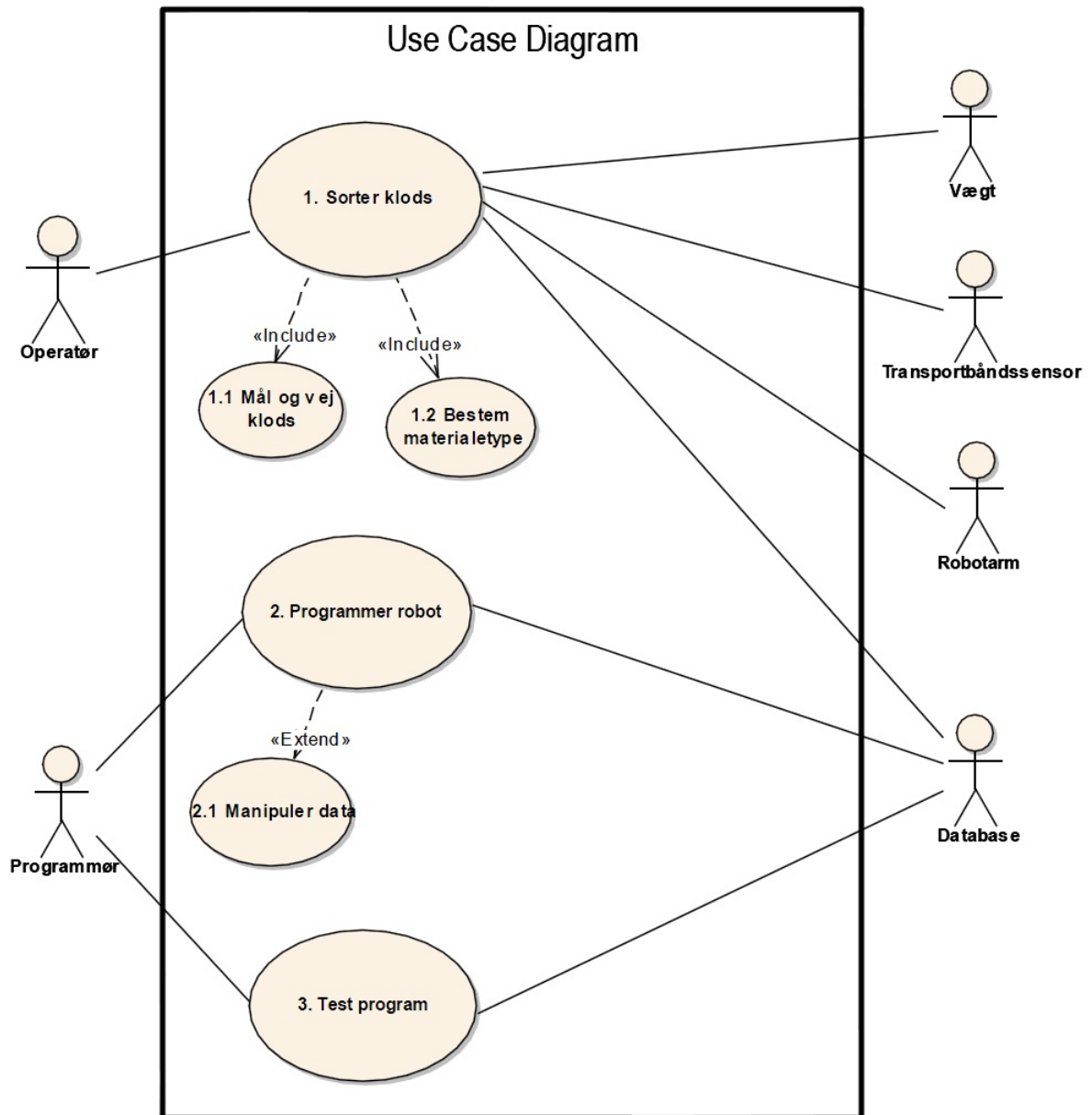
<b>Aktør navn</b>	Database.
<b>Beskrivelse</b>	Databasen indeholder logfiler for systemets data for de enkelte klodser, samt informationer om robotten og systemets status. Ligeledes indeholder databasen programmerne, som programmøren har lavet.

<b>Antal samtidige aktører</b>	1
--------------------------------	---

## 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 robotens 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 beskrevet et kamera. 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:

- **Operator**  
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 inden for programmering og kendskab til systemet. Programmøren kan lave brugerdefinerede programmer til robotten.

## 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 kilderne. 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.

Softwaremæssigt kræves det, at kundens computer kører en nyere version af Windows operativsystem (XP, Vista eller Windows 7) da der udvikles i WPF/.NET.

## 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.

**Aktører og interessenter:**

Primære aktører:

- Operatør.

Sekundære aktører:

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

**Antal samtidige forekomster:**

En samtidig forekomst.

**Frekvens:**

En gang pr. tredje minut.

**Ikke funktionelle krav:**

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

**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, at den ikke må have fat i en klods.

**Slutresultat ved succes:**

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

**Slutresultat ved undtagelser:**

Klodsen 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:**

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

**Undtagelser:**<sup>1</sup>

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

1. Strømmen afbrydes fysisk.

Undtagelse \* b: Operatøren vælger at stoppe systemet.

1. Systemet stoppes.

Undtagelse \* c: Operatøren vælger at genstarte systemet.

1. Hvis systemet er stoppet, sættes robotarmen i startposition.

Undtagelse 1-3 a: Klodsen kan ikke registreres.

1. Transportbåndet kører videre.
2. Kloden ryger ud over kanten ned i en kasse.

Undtagelse 6 a: Klodsen tabes.

1. Operatøren bliver præsenteret for en visuel alarm der indikerer, at klodsen er blevet tabt.
2. Robotarmen føres tilbage til startposition.
3. Systemet stoppes.

---

<sup>1</sup>Stjernen indikerer, at undtagelsen gør sig gældende for alle normalforløb

## 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:

- Systemet.

Sekundære aktører:

- Robotarm.
- Vægt.
- Transportbånd.

### Antal samtidige forekomster:

En samtidige forekomst.

### Frekvens:

En gang pr. tredje minut.

### Ikke funktionelle krav:

- Klodsen skal måles med en præcision på  $\pm 0.5$  cm.
- Klodsen skal vejes med en præcision på  $\pm 5$  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. Ligeledes skal der være en klods på transportbåndet.

### Slutresultat ved succes:

Klodsens mål og masse er fundet.

### Slutresultat ved undtagelser:

Klodsens mål og masse er ikke blevet fundet.

### Normalforløb:

1. Robotarmen tager fat i klodsen.
2. Klodsens første side måles.
3. Klodsen slippes.
4. Robotarmen løftes og roteres 90 grader.

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.

**Undtagelser:**

Undtagelse \* a: Klodsen tabes på hvilket som helst tidspunkt

1. Gennem en visuel alarm bliver operatøren informeret om, at klodsen er blevet tabt.
2. Robotarmen føres tilbage til startposition.
3. Systemet stoppes.

Undtagelse 1 og 5 a: Klodsen kan ikke opfanges af robotarmen.

1. Transportbåndet startes.
2. Kloden ryger ud over kanten ned i en kasse.
3. Gennem en visuel alarm bliver operatøren informeret om, at klodsen fjernes automatisk.

Undtagelse 10 a: Klodsen kan ikke opfanges (på vægten) af robotarmen.

1. Systemet stopper.
2. Gennem en visuel alarm bliver operatøren informeret om, at klodsen skal fjernes manuelt.



### 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:

- Systemet.

Sekundære aktører:

- Database.

**Antal samtidige forekomster:**

En samtidig forekomst.

**Frekvens:**

En gang pr. tredje minut.

**Ikke funktionelle krav:**

- Klodsens densitet gemmes i databasen i med enheden  $\frac{g}{cm^3}$ .
- Densiteten gemmes med to decimaler. Ex: Bly  $11,34 \frac{g}{cm^3}$ .
- Rumfanget gemmes med to decimaler. Ex:  $33,78 cm^3$ .
- Klodsens målte værdi gemmes i databasen i cm.
- Klodsens masse gemmes i databasen i gram.
- Klodsen dimensioner gemmes med en decimal. Ex: 15.2 cm.
- Massen gemmes uden decimaler. Ex: 176 g.
- !! GRÆNSER EFTER SORTERING SPECIFICERES SENERE !!

**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:**

1. Rumfanget af klodsen udregnes.
2. Ud fra klodsens rumfang og masse udregnes densiteten.
3. Der oprettes forbindelse til databasen.
4. Den udregnede densitet sammenlignes med liste af eksisterende materialetyper, og deraf bestemmes klodens type.
5. Mål, masse, rumfang, densitet og materialetype gemmes i databasen.

**Undtagelser:**

Undtagelse 3 a: System kan ikke oprette forbindelse til databasen.

1. Gennem en visuel alarm bliver operatøren informeret om, at forbindelsen ikke kunne oprettes.
2. Systemet prøver at genetablere forbindelse til databasen indtil det lykkedes.
3. Operatøren bliver informeret om, at forbindelsen til databasen er blevet genoptaget.

Undtagelse 4 a: Densiteten kan ikke findes i databasen

- Inden for  $\pm$  **NOGET** gram antages materiale.
- Uden for  $\pm$  **NOGET** gram udskrives fejlmeddelelse om ikke kendt materiale.

Undtagelse 4-5 a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver operatøren informeret om, at forbindelsen til databasen er blevet afbrudt
2. Systemet prøver at genetablere forbindelse til databasen indtil det lykkedes.
3. Operatøren bliver informeret om, at forbindelsen til databasen er blevet genoprettet.

### 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.

**Initiering:**

Programmøren tilkendegiver over for systemet, at han ønsker at omprogrammere det.

**Aktører og interessenter:**

Primære aktører:

- Programmør

Sekundære aktører:

- Database

**Antal samtidige forekomster:**

En samtidig forekomst.

**Frekvens:**

Ikke defineret.

**Ikke funktionelle krav:**

- Brugeren skal være logget ind som 'programmør'.

**Referencer:**

Til denne Use Case kan der anvendes en extend Use Case:

- *Use Case 2.1: Manipuler data*

**Startbetingelser:**

Databasen skal være tilsluttet.

**Slutresultat ved succes:**

Programmøren har færdiggjort et program, som gemmes på databasen.

**Slutresultat ved undtagelser:**

Et endnu ikke færdiggjort program gemmes på databasen, som kan tilgås senere.

**Normalforløb 1 - Opret nyt program:**

1. Programmøren tilkendegiver over for systemet, at denne ønsker at lave en ny styring til robotten.
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 uploades.
5. Systemet gemmer automatisk programmet.

**Undtagelser 1 - Opret nyt program:**

Undtagelse 3.a: Programmøren afslutter program uden at gemme.

- (a) Systemet tilkendegiver over for programmøren, om denne ønsker at gemme programmet.

Undtagelse 4.a: Programmøren har ikke gemt programmet før.

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

**Normalforløb 2 - Anvend gammelt program:**

1. Programmøren tilkendegiver over for systemet, at denne ønsker at bruge et gammelt program fra databasen.
2. Systemet lister gamle programmer, der er tilgængelige.
3. Programmøren tilkendegiver over for systemet, hvilket program fra listen denne ønsker at køre.
4. Systemet indlæser dette program til robotten.

**Undtagelser 2 - Anvend gammelt program:**

Undtagelse 2.a: Der er ingen gamle programmer tilgængelige.

- (a) Systemet giver besked om, at der ikke er gamle programmer at indlæse.

**Normalforløb 3 - Anvend standard program:**

1. Programmøren tilkendegiver over for systemet, at denne ønsker at køre standard programmet.<sup>2</sup>
2. Systemet indlæser programmet til robotten.

**Normalforløb 4 - Rediger gammelt program:**

1. Programmøren tilkendegiver over for systemet, at denne ønsker at redigere et gammelt program fra databasen.
2. Systemet lister gamle programmer, der er tilgængelige.
3. Programmøren tilkendegiver over for systemet, 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.

---

<sup>2</sup>Program, der fulgte med fra udviklerne til almen sortering

6. Programmøren tilkendegiver over for systemet, at denne skal gemme programmet.
7. Systemet gemmer programmet.

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

Undtagelse 2.a: Der er ingen gamle programmer tilgængelige.

- (a) Systemet giver besked om, at der ikke er gamle programmer at indlæse.

Undtagelse 6.a: Programmøren afslutter program uden at gemme.

- (a) Systemet tilkendegiver over for programmøren, om denne ønsker at gemme programmet.

### 3.5 Use Case 2.1 - Manipuler data (Extend)

**Mål:**

Målet med denne Use Case er, at ændre data i databasen.

**Initiering:**

Use Casen initieres gennem *Use Case 2: Programmer robot*.

**Aktører og interessenter:**

Primære aktører:

- Programmør.

Sekundære aktører:

- Databasen.

**Antal samtidige forekomster:**

En samtidige forekomst.

**Frekvens:**

Ikke defineret.

**Ikke funktionelle krav:**

- Brugeren skal være logget ind som 'programmør'.

**Referencer:**

Denne Use Case er en udvidelse af *Use Case 2: Programmer robot*.

**Startbetingelser:**

Startbetingelserne gør sig gældende i, at *Use Case 2: Programmer robot* skal være igangsat.

Ligeledes har brugeren tilkendegjort overfor systemet, at denne ønsker at manipulere data.

**Slutresultat ved succes:**

Ønsket data er blevet manipuleret.

**Slutresultat ved undtagelser:**

Ved fejlindtastning, bliver data ikke manipuleret.

**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
  - Placering af materiale
  - Densitet
  - Følsomhed<sup>3</sup>

---

<sup>3</sup>Den fejlmargen, der godtages for det pågældende materiale genkendes

3. Programmøren tilkendegiver overfor systemet, at denne ønsker de indtastede data gemt.

**Undtagelser 1 - Tilføj ny materialetype:**

Undt. 2 a. Placering har ugyldige koordinater.

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

Undt. 2 b. Densiteten er ikke gyldig.

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

Undt. 2 c. Følsomheden er ikke gyldig.

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

Undt. 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.

**Normalforløb 2 - Fjern materialetype:**

1. Programmøren tilkendegiver overfor systemet, at denne ønsker at fjerne en materialetype.
2. Systemet fjerner materialetypen.

**Normalforløb 3 - Ændre placering i boks for materialetype:**

1. Programmøren tilkendegiver overfor systemet, at denne ønsker at ændre placeringen for en materialetype.
2. Systemet gør det muligt at ændre placering af materialetype.
3. Programmøren tilkendegiver overfor systemet, hvor materialetypen skal placeres.
4. Systemet gemmer placering for den pågældende materialetype.

**Undtagelser 3 - Ændre placering i boks for materialetype:**

Undt. 3 a. Programmøren indtaster ugyldig placering.

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

**Normalforløb 4 - Slet program:**

1. Programmøren tilkendegiver overfor systemet, at denne ønsker at slette et program.

2. Systemet lister alle tilgængelige programmer der kan slettes.
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 4 - Slet program:**

Undt. 5 a. Programmøren tilkendegiver overfor systemet, at dette ikke ønskes.

- (a) Systemet sletter ikke programmet.

**Normalforløb 5 - Hent logfil:**

1. Programmøren tilkendegiver overfor systemet, at denne ønsker at hente en logfil.
2. Systemet lister alle tilgængelige logfiler.
3. Programmøren tilkendegiver overfor systemet, hvilken logfil der skal hentes.
4. Systemet åbner et vindue, hvor programmøren vælger hvor logfilen skal gemmes.
5. Programmøren tilkendegiver overfor systemet, hvor logfilen skal gemmes, samt hvilket navn den skal gemmes med.
6. Systemet uploader filen til programmørens computer.

**Undtagelser 5 - Hent logfil:**

Undt. 3 a. Der er ikke nogle logfiler i listen.

Undt. 4 a. Programmøren tilkendegiver overfor systemet, at han ikke ønsker at gemme filen alligevel.

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

Undt. 5 a. Filnavnet er allerede optaget.

- (a) Systemet spørger programmøren om denne ønsker at overskrive den eksisterende fil.
  - i. Programmøren tilkendegiver overfor systemet at dette ønskes.
    - A. Filen overskrives med den nye.
  - ii. Programmøren tilkendegiver overfor systemet, at dette ikke ønskes.
    - A. Systemet viser punkt 4 igen.



### 3.6 Use Case 3: 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:

- Database.

**Antal samtidige forekomster:**

En samtidig forekomst.

**Frekvens:**

Ikke defineret.

**Ikke funktionelle krav:**

- Loggen udskrives med dato/tid [YYYY.MM.DD-hh.mm.ss] Ex: "[2012.04.15-06.28.42]"

**Referencer:**

Ingen.

**Startbetingelser:**

Det forudsættes at programmet er i en tilstand, hvor den ikke er optaget af andre processer.

**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 fra databasen.
3. Programmøren vælger en af programmerne.
4. Programmøren starter simuleringen af programmet.
5. Alle systemets informationer fra programmet vises i loggen.

6. Simulationen afsluttes.

**Undtagelser:**

Undtagelse \* a: Programmøren trykker på stop.

1. Systemet stoppes.

Undtagelse 2 a: Der mistes forbindelse til databasen.

1. Gennem en visuel alarm bliver operatøren informeret om, at forbindelsen til databasen er blevet afbrudt
2. Systemet prøver at genetablere forbindelse til databasen indtil det lykkedes.
3. Operatøren bliver informeret om, at forbindelsen til databasen er blevet genoprettet.

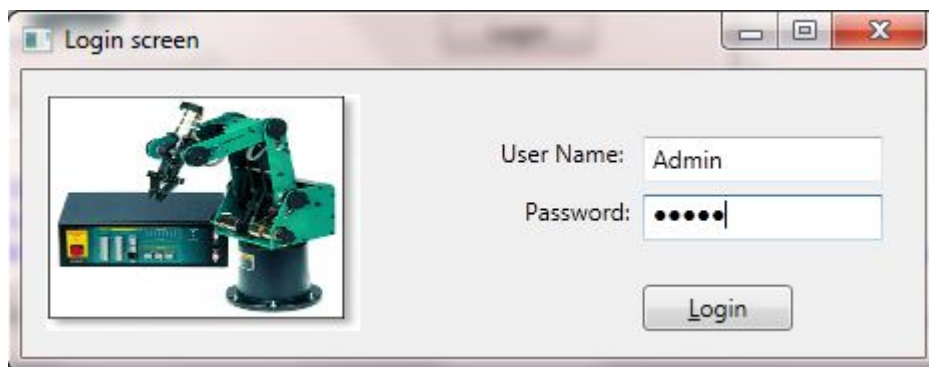
Undtagelse 2 b: Der er ingen tilgængelige programmer i databasen.

1. Programmøren får en fejlmeddelelse der informerer herom.
2. Systemet returnerer til hovedmenuen.

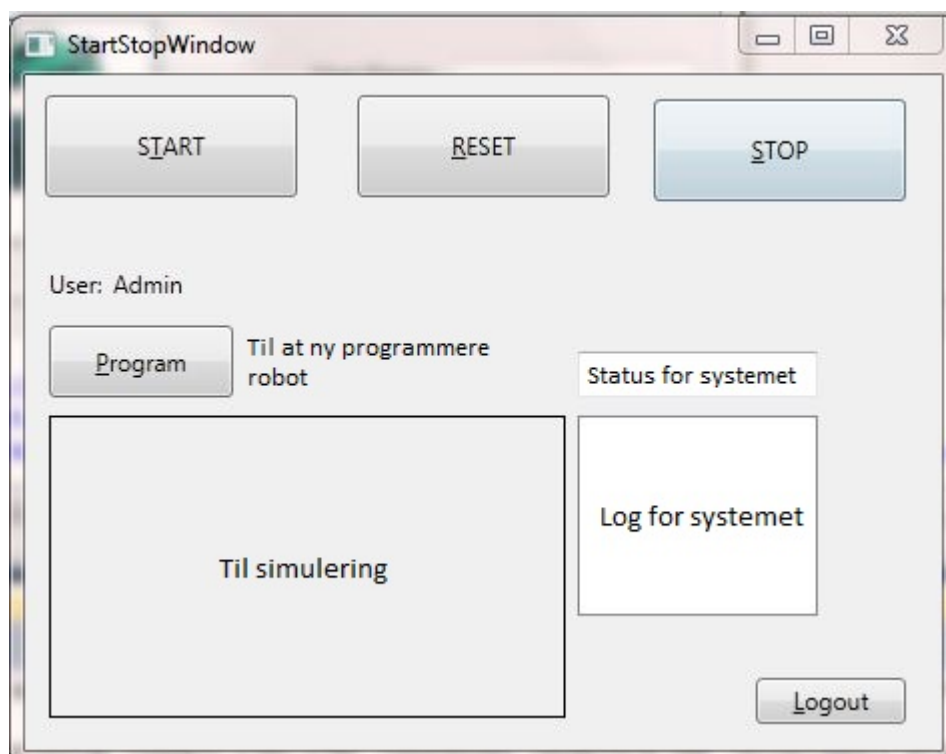
## 4 Eksterne grænseflader

### 4.1 Brugergrænseflade

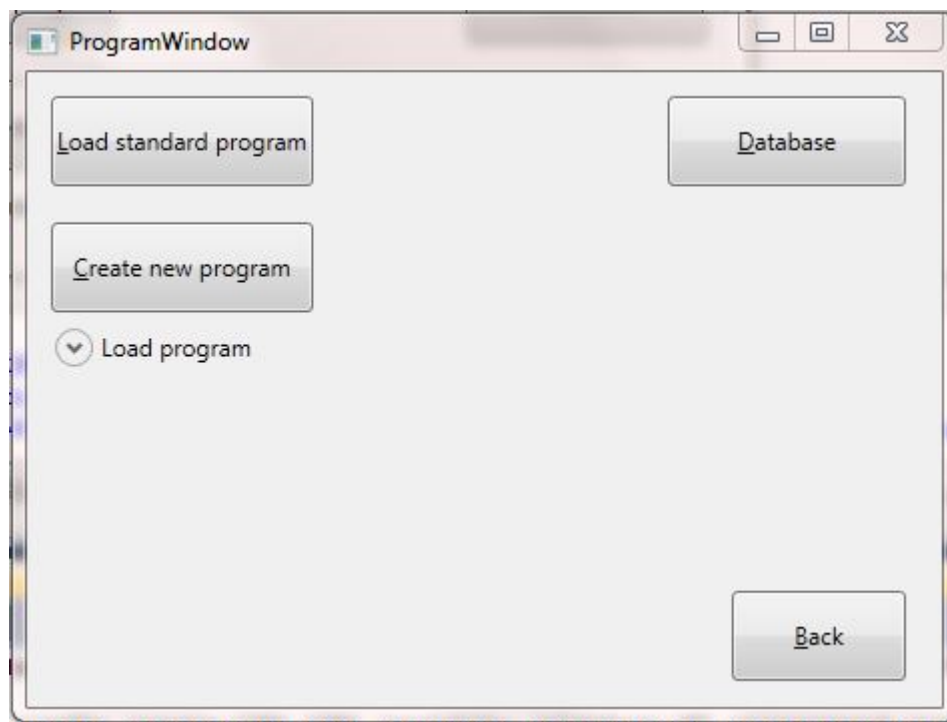
Der er blevet udformet en skitse over brugergrænsefladen, så kunden kan få en idé om, hvordan det grafiske vil komme til at se ud. Det er vigtigt at understrege, at det er en skitse, og at den endelige brugergrænseflade ikke nødvendigvis vil ligne denne fuldstændigt:



Figur 5: Logon skærm



Figur 6: Hovedvindue



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: Særdeles 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 Scorbaser-program.
- Robotten skal programmeres via det udleverede bibliotek USBC.dll.

## 7 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.