Projekt: Sorting Industrial Robot

Dato: 23-04-2012

Titel:

Kravspecifikation
for
Sorting Industrial Robot
(SIR)

RoboGO

Versionshistorik

Ver.	Dato	Initialer	Beskrivelse
0.1	11-02-12	RHT	Første udkast.
0.2	16-02-12	RHT	Sat ind firma og produktnavn.
1.0	08-03-12	Alle	Første version før Construction.
1.1	02-04-12	CTD	Merge af alle usecases fra branches
1.2	23-04-12	CTD	Merge af alle usecases fra alle branches
1.3			
1.4			
1.5			
1.6			

Godkendelsesformular

Forfatter(e):	Søren Howe Gersager(10430) Cong Thanh Dao(10517) Yusuf Tezel(10568) Nicolaj Quottrup(10754) René Høgh Thomsen(10778) Michael Batz Hansen(10791) Sam Luu Tong(10898)
Godkendes af:	Poul Ejnar Rovsing
Projektnummer:	1
Dokument-id: **(filnavn)**	Kravspecifikation.odt Kravspecifikation.pdf
Antal sider:	40
Kunde:	Robotic Global Organization(RoboGO)

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

Sted og dato:

_____ _____

René H. Thomsen Poul Ejnar Rovsing

Indholdsfortegnelse

1. Indledning

1.1 Formål

Robotic Global Organization, RoboGO, har indkøbt en Scorbot ER-4U robotpakke. Pakken indeholder en robotarm, samlebånd og controller. Hele systemet styres manuelt. Virksomheden ønsker derfor, at et internt softwareudviklingsholdet skal udvikle et stykke software, som automatisk kan styre robotpakken samt at lave en vægtforstærker til pakken. Projektet vil dække under navnet SIR, Sorting Industrial Robot.
Dette projekt er internt i virksomheden, og derfor er vi både producent og kunde.

1.2 Referencer

Bøger og udlevering fra følgende fag:

- I4DAB1
- I4INF1 – Measurements & Instrumentations Principles
- I4SWT1 – The Art of Unit Testing with Examples in .NET
- I4WIN1- Pro WPF in C# 2010

Udleverede materialer:

- Controller-USB, Users Manual
- Scorbase, Users Manual
- Scorbot-ER 4u, Users Manual
- Robotics and Materials Handling 1, Student Activities Book
- RoboCell, Users Manual
- Datablad for vejecelle
- USBC-documentation
- Projektoplæg

1.3 Læsevejledning

Udover Indledningen har vi ni punkter i kravspecifikationen.

- Punkt 2 : Omhandler kravene til SIR, som skal udvikles.
- Punkt 3 : Omhandler de funktionelle krav til SIR ved hjælp af use cases.
- Punkt 4 : Omhandler kravene til grænsefladerne.
- Punkt 5 : Omhandler kravene som stilles til systemets ydelse.
- Punkt 6 : Omhandler kvalitetsfaktorer som stilles til systemet.

- Punkt 7 : Omhandler kravene til designet der skal overholdes, når systemet er færdigudviklet.
- Punkt 8 : Omhandler andre krav til systemet såsom myndighedskrav, miljøkrav etc.
- Punkt 9 : Omhandler det aftalte antal delleveringer af SIR.
- Punkt 10: Bilag, herunder skabeloner, datadefinitioner og ordliste.

2. Generel beskrivelse

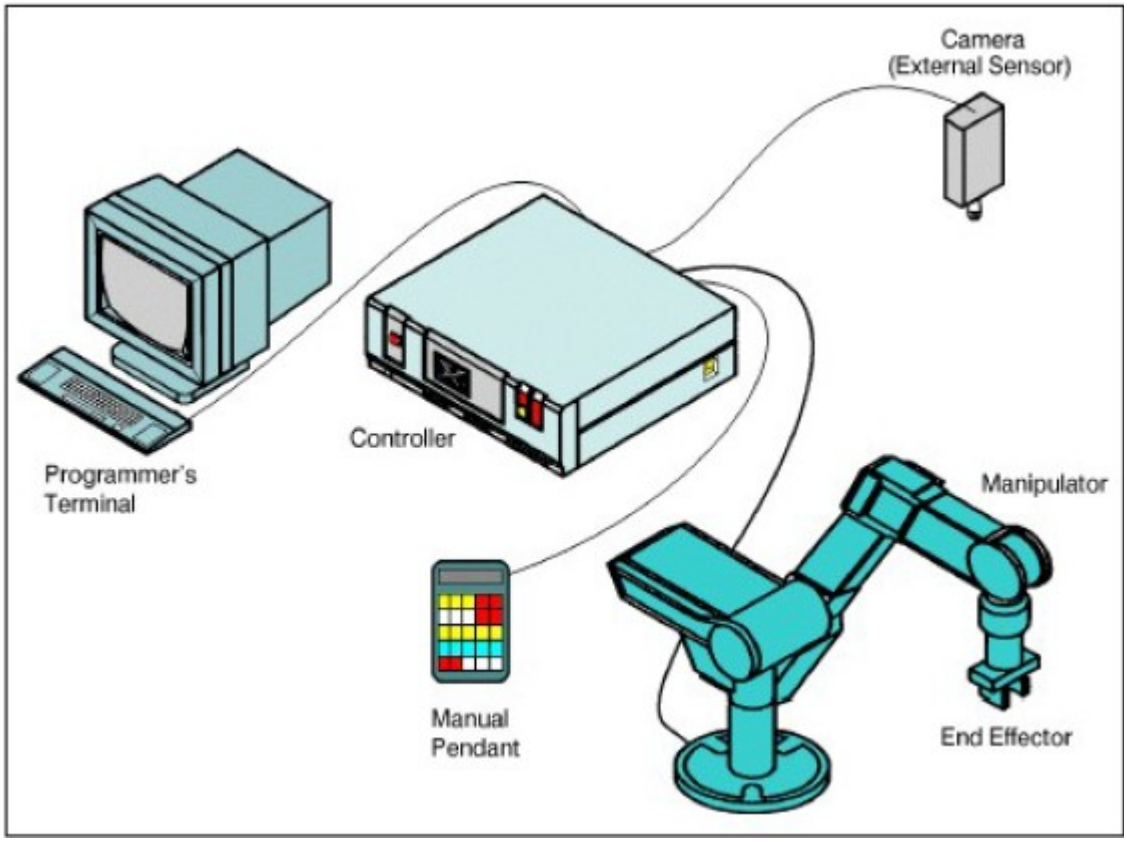
2.1 Systembeskrivelse

Vores projektopgave omhandler, at vi skal automatisere en sorteringsproces, der indtil nu er blevet udført manuelt. Problemet løses ved at udvikle robotsoftware til en robot, der automatisk kan sortere emner efter massefylde og dimensioner. Vores system har vi valgt at kalde Sorting Industrial Robot (SIR).
Emnerne bliver sorteret i forskellige kasser, der er én kasse per emnetype. I systemoversigten, som ses afbilledet i *illustration 1* er det selve Controlleren, vi kommer til at udvikle. Den består i korte træk af:

1) **Database** – Indeholder informationer om hele systemet: emner, aktører, placering af kasser m.m. Brugeren kan via databasen hente og modificere disse informationer.
2) **IDE** – Gør brugeren i stand til at styre robotten via et scripting-language, vi selv designer.
3) **Simulator** – En robot-simulator, der gør det muligt at udføre funktioner og teste systemet, uden der er tilknyttet en robot.
4) **Styresystemet** – tager sig af kommunikationen mellem de andre komponenter og robotten.

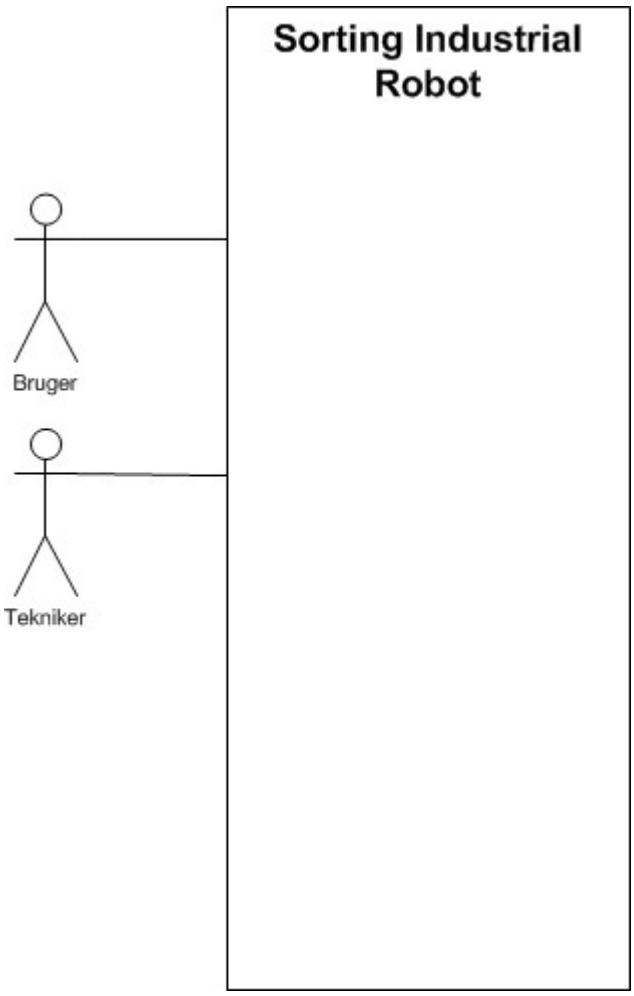
Controlleren kommunikerer med Robotten, en Terminal og en Vejecelle.

2.1.1 Systemoversigt



Figur 1: Visuel beskrivelse af vores system.

2.1.2 Aktør-kontekst diagram



*Figur 2: V*iser et aktør-kontekst diagram for SIR med de aktører, der kommunikerer med systemet. Aktørerne er beskrevet i næste afsnit.

2.1.3 Aktør beskrivelser

En primær aktør beskriver en aktør, der har et eller flere mål, som ønskes opfyldt af systemets Use Cases. En sekundær aktør beskriver derimod en aktør der er en nødvendig deltager i en eller flere Use Cases for at opfylde en primær aktørs mål. I nogle tilfælde kan en aktør både være primær og sekundær.

Aktør navn:	Bruger
Type:	Primær

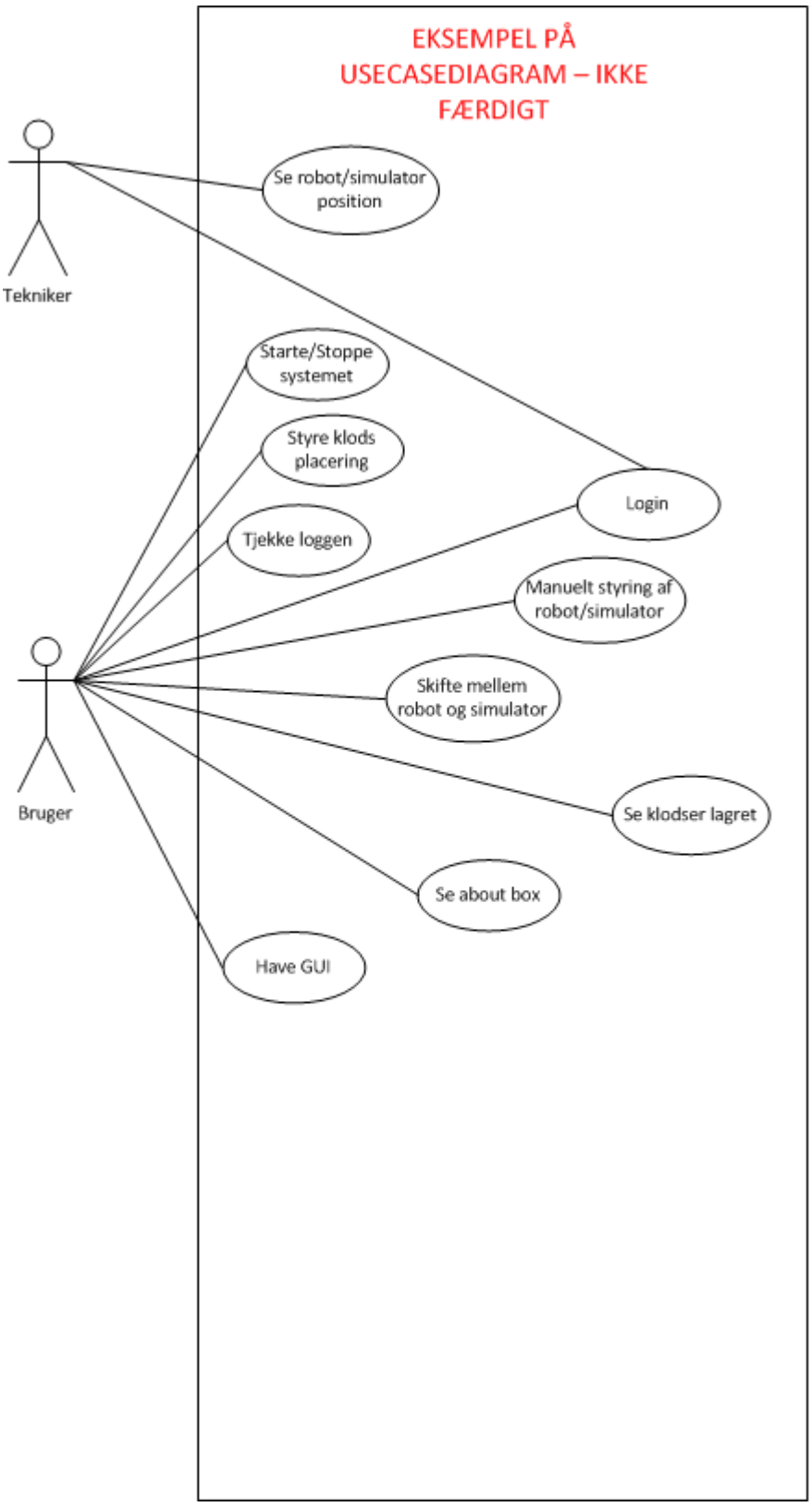
Beskrivelse:	En person, der benytter vores robotsystem. Brugeren er interesseret i at benytte funktioner i systemet.
Antal samtidige aktører:	1

Aktør navn:	Tekniker
Type:	Primær
Beskrivelse:	En person, der benytter de vedligeholdelsesmæssige funktioner i vores system.
Antal samtidige aktører:	1

2.2 Systemets funktioner

Systemets funktioner og de funktionelle krav er fundet og beskrevet vha. Use Case teknikken. Et diagram over Use Cases for vores system kan findes herunder. Formålet med diagrammet er at give overblik over funktionaliteten i vores system. Use Cases i vores diagram kan findes mere detaljeret beskrevet i kapitel 3.

2.2.1 Use Case diagram



Figur 3: Viser use-case diagram.

Use Case diagrammet vil blive opdateret, efterhånden som vi kommer i gang med Use Cases i Construction sprintene.

2.3 Systemets begrænsninger

Vores SIR-system er udviklet udelukkende udviklet til at kunne automatisere en allerede eksisterende proces. Det er derfor uvist, hvor let systemet vil kunne modificeres til brug i andre øjemed, da den udelukkende er udviklet med vores specifikke problem for øje.

2.4 Systemets fremtid

I fremtiden ville der kunne implementeres følgende for at udvide systemets funktionalitet:

- Automatisering af andre problemstillinger.
- Manuel pendant.

Hvis robotten udvides til at kunne løse andre problemstillinger end at sortere klodser, vil vi have en robot der er meget mere generisk. Robotten vil altså kunne bruges i mange forskellige sammenhæng og ikke kun til vores specifikke opgave. Dette vil åbne op for nye muligheder og mange opgaver der tidligere har været manuelt varetaget vil kunne blive automatiseret.

En manuel pendant ville kunne give brugeren en anden tilgang til at kunne kontrollere robotten, brugeren er altså ikke afhængig af computeren.

2.5 Brugerprofil

Til systemet er der 2 brugerprofiler.
Bruger: Behøver ikke at have et dybdegående kendskab til den tekniske del af systemet, da måden hvorpå man interagerer med systemet skal være intuitivt, dog skal man som bruger have et vis kendskab til computerbrug, da man skal være i stand til at kunne kode og compile simpel kode samt debugge det for at kunne styre robotten.
Tekniker: Skal have et mere dybdegående kendskab end brugeren, da teknikerne skal kunne vedligeholde systemet.

2.6 Krav til udviklingsforløbet

Projektet udvikles efter en iterativ udviklingsproces, UP, hvor vi indtil videre har 4 delleveringer, for løbende af få feedback på vores arbejde, udover det benytter vi Scrum som framework.

2.7 Omfang af kundeleverance

Til RoboGo overdrages følgende dokumentation.

- Kravspecifikation.
- Accepttestspecifikation.
- Testresultater.
- Designdokumentation.
- Implementeringsdokument.
- Brugermanual.
- Software i eksekverbar form og kildekoder.
- Dokumentationen afleveres på CD-ROM og i papirform

2.8 Forudsætninger

Der forventes at SCORBASE robotten er til rådighed, samt at den virker.

3. Funktionelle krav – Use Cases

Use casene er delt op 3 dele:

- Styresystem
- Simulator
- IDE

Da de passer til henholdsvis hver af disse komponenter.
De er i første omgang skrevet i en brief format, som så senere kan specificeres mere præcist, når de bliver arbejdet på i sprintene.
De er sat i tilfældig rækkefølge, da det er op til produktejeren/stakeholder om at holde styr på prioritering i hans produkt backlog. Disse use cases er her for reference for udviklerne.

Use case numrene passer med numrene for user stories, så for eksempel use case 1 refererer til product backlog user story 1.

3.1 Styresystem:

3.1.1 Use Case 1: Starte/stoppe systemet.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Systemet.

Stakeholders og Interessanter:

RoboGO, de har interesse i at en tidligere opgave som blev udført manuelt bliver automatiseret.

Preconditions:

Computeren er tændt.

Postconditions:

Systemet er startet med visuel visning.

Systemet er stoppet, og brugergrænsefladen lukkes.

Main success scenarie:

1. Brugeren tænder for systemet.
2. Brugeren stopper systemet.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

3.1.2 Use Case 2: Styre klodsplacering.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og Interessenter:

RoboGO: de har interesse i, at en tidligere manuel, udført opgave bliver automatiseret.

Preconditions:

Computeren er koblet til resten af SCORBOT systemet.

Hermed menes, at USB-Controlleren og SCORBOTTEN er forbundet til med kabel.

Desuden skal der være tilgængeligt, flytbare klodser indenfor robottens område.

Postconditions:

Robotten vil regulere efter den nye position givet fra brugerens input.

Main success scenarie:

1. Brugeren interagerer med brugergrænsefladen.
2. Robotten tager en klods og tager målinger af denne.
3. Brugeren vælger placering af klodsen.
4. Klodsen placeres på det korrekte sted.

Extensions:

4.1 Hvis klodsen ikke placeres på det korrekte sted, skal værdierne ændres i databasen.

Ikke funktionelle krav:

Placering af klodsen skal have en præcision på 1 cm af klodsens placering samt skal den have en nøjagtighed centreret i forhold til andre klodser.

3.1.3 Use Case 3: Tjekke loggen.

Main success scenarie:

"Brugeren interagerer med brugergrænsefladen og får en liste over alle events der er sket siden programmet startede."

3.1.4 Use Case 4: Manuelt styre.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Robot.

Stakeholders og Interessenter:

RoboGO: de har interesse i, at en tidligere manuel, udført opgave bliver automatiseret.

Preconditions:

Computeren er koblet til resten af SCORBOT systemet.

Hermed menes, at USB-Controlleren og SCORBOTTEN er forbundet til med kabel.

(Medmindre simulator benyttes.)

Postconditions:

Robotten/Simulatoren bevæger sig alt efter, hvad brugeren beder den om runtime.

Main success scenarie:

1. Brugeren interagerer med brugergrænsefladen.
2. Manuel brugergrænseflade åbner.
3. Brugeren interagerer med brugergrænsefladen eller bruger en genvejstast.
4. Robotten bevæger sig efter brugerens ønske.

Extensions:

3.1 Der er ikke forbindelse til robotten.

1. En besked om fejlen vil blive vist.
2. Brugeren lukker beskeden.
3. Brugergrænsefladen for manuel styring vil lukke ned.

4.1 Robotten kan ikke bevæge sig i retningen, da den har nået sin rotationsgrænse.

1. Robotten bevæger sig ikke.(Ingen besked vil blive vist.)

Ikke funktionelle krav:

Styringen foregår ikke via 'toggle' – med det menes, at brugeren skal blive ved med at holde musen nede i brugergrænsefladen eller tasten nede på tastaturet, da robotten vil stoppe, når den (musen eller tasten på tastaturet) bliver sluppet.

3.1.5 Use Case 5: Skifte mellem robot og simulator.

Main success scenarie:

”Brugeren skifter mellem 2 forskellige indstillinger og alt efter indstillingen, vil det enten være robotten eller simulatoren, som udfører koden.”

3.1.6 Use Case 6: Se klodser lagret.

Main success scenarie:

”Brugeren interagerer med brugergrænsefladen og får en liste over alle klodser der er blevet placeret af robotten/simulatoren.”

3.1.7 Use Case 7: Se about box.

Main success scenarie:

”Brugeren interagerer med brugergrænsefladen og får listet alle tingene der indgår i systemet.”

3.1.8 Use Case 8: Login.

Main success scenarie:

”Når programmet er startet op vil der være adgang til login, hvor brugeren/teknikeren vil kunne login. Systemer vil så være indstillet for brugeren/teknikeren der er logget ind.”

3.1.9 Use Case 9: Have GUI.

Main success scenarie:

”Brugeren har en GUI, hvor der er adgang til systemets funktionalitet.”

Use Case Navn: Have GUI

Scope: Styresystem

Niveau: User-goal

Frekvens: Brugerbestemt

Primær Aktør: Brugeren, Tekniker

Sekundære aktører: Systemet

Stakeholders og interessenter:

RoboGO, de har interesse i at en tidligere opgave som blev udført manuelt bliver automatiseret.

Preconditions:

PC'en er tændt, logget ind og SIR programmet er installeret

Postconditions:

SIR programmet åbnes og funktionaliteterne er tilgængelige

Main success scenarie:

1. Brugeren kører SIR programmet.
2. Usecase 8 indtræffes
3. Brugeren har nu adgang til GUI funktionaliteterne

Extensions:

Undtagelse:

Ingen.

Ikke-funktionelle krav:

3.1.11 Use Case 10: Følge program udvikling.

Main success scenarie:

”Under kørsel af programmet vil brugeren se hvilken del af koden der kører.”

3.3.12 Use Case 11: Indlæse/køre systemet DSL-filer direkte.

Scrope: Styresystem

Niveau: User-goal

Frekvens: Brugerbestemt

Primær aktør: Bruger

Sekundære aktører: Ingen

Stakeholders og interessenter:

RoboGO, de har interesse i at en tidligere opgave som blev udført manuelt bliver automatiseret.

Preconditions: Styresystemet skal være startet op.

Postconditions: Brugeren får læst sin valgte DSL fil, som robotten/simulatoren kører.

Main success scenarie:

1. Brugeren vælger DSL fil og lader styresystemet indlæse denne.
2. Robotten/simulatoren kører den indlæste fil.

Extensions:

Ingen.

Undtagelser:

1.1 Hvis filen ikke er gyldig, meldes der en fejl i brugergrænsefladen via en beskedboks.

Ikke funktionelle krav:

Ingen.

3.2 Simulator:

3.2.1 Use Case 12: Se robot/simulator position.

Scope: Stimulator.

Niveau: Tekniker.

Frekvens: Brugerbestemt.

Primær aktør: Tekniker.

Sekundære aktører: Ingen?

Stakeholders og Interessenter:

Ingen.

Preconditions:

Computeren er koblet til resten af SCORBOT systemet.

Hermed menes, at USB-Controlleren og SCORBOTTEN er forbundet til med kabel.

Postconditions:

Teknikeren får vist positionen af robotten/simulatoren på brugergrænsefladen.

Main success scenarie:

1. Teknikeren interagerer med brugergrænsefladen.
2. Teknikeren vælger styring og kører robotten/simulatoren.
3. Robottens/simulatorens position vises i brugergrænsefladen.

Extensions:

2.1 Der kunne ikke forbindes til robotten.

1. En besked om fejlen vil blive vist.

Ikke funktionelle krav:

Opdateringen mellem den fysiske robot og brugergrænsefladen skal have en minimal opdatering på ét sekund. Hvis simulatoren vælges, vil dette være det samme.

3.2.2 Use Case 13: Simulere koden.

Scope: Simulator.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og Interessanter:

RoboGO, de har interesse i at en tidligere opgave som blev udført manuelt bliver automatiseret.

Preconditions:

Editoren indeholder koden, der vil køres.

Koden er kørbar idet at DSL syntaksen er overholdt.

Postconditions:

Koden i editoren bliver kørt i simulatoren.

Main success scenarie:

1. Brugeren skriver sin kode.(Måske uden for scope)
2. Brugeren interagerer med brugergrænsefladen.
3. Koden bliver kørt i simulatoren.

Extensions:

1.1 Brugeren åbner en eksisterende fil:

1. Koden bliver indlæst til editoren.

3.1 Koden overholder ikke syntaksten:

1. Kørsel af koden bliver ikke udført.
2. Der vil blive givet besked om at koden ikke kunne køres.

Ikke funktionelle krav:

Koden skal begynde at køre inden for 1 sekund, efter brugeren har bedt om det.

3.2.3 Use Case 14: Vise kørselstid.

Main success scenarie:

”Efter brugeren har startet simuleringen, får han vist en timer, som indikerer hvor lang tid simuleringen ville vare i virkeligheden, hvis koden kørte på robotten.”

3.2.4 Use Case 15: Have grafisk visning.

Scope: Simulator.

Niveau: Bruger

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og Interessenter:

RoboGO: de vil gerne have, at man kan følge med gennem simulatoren i stedet den fysiske robot.

Preconditions:

Brugergrænsefladen og main programmet er oppe og køre, samt befinder brugeren sig i simulator-fanen.

Postconditions:

Brugeren vil få en grafisk visning af robotten i simulatoren.

Main success scenarie:

1. Brugeren interagerer med brugergrænsefladen.
2. Brugeren kører en kommando fra en åbnet fil, således simulationen igangsættes.
3. Simulationen vil vise robottens bevægelser gennem dials i simulatoren, samt opdateres positionen og tiden.

Extensions:

2.1 Filen eksisterer ikke.

1. Brugeren går til IDE-fanen og skriver en kommando i editoren.
2. Brugeren gemmer filen.
3. Brugeren går tilbage til simulator-fanen og åbner filen.
4. Brugeren kører filen, således simulationen igangsættes.

Ikke funktionelle krav:

Opdateringen af brugergrænsefladen skal have en minimal opdatering på et halvt sekund.

3.2.6 Use Case 16: Kalibrere værdier.

Main success scenarie:

”Teknikere aktiverer en speciel menu, hvori han kan finindstille værdier”

3.3 IDE:

3.3.1 Use Case 17: Eksekvere og debugge DSL kode.

Scope: IDE.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og Interessenter:

RoboGO; de har interesse i at se, om DSL kode kan eksekveres.

Preconditions:

Styresystemet er oppe og køre, og brugeren befinder sig i IDE-fanen.

Postconditions:

Brugeren vil kunne eksekvere kode.

Main success scenarie:

1. Brugeren skriver kode i editoren.
2. Brugeren eksekverer koden.
3. Koden kører.

Extensions:

1.1 Brugeren åbner en fil med DSL kode.

1. Følger punkt 2. i main success scenariet.

3.1 Koden kunne ikke køre.

1. GUI´en informerer om fejlen.
2. Brugeren kan tjekke kodens tilstand.
3. Genstarter fra punkt 1.

Ikke funktionelle krav:

Ingen.

3.3.2 Use Case 18: Debugge.

Main success scenarie:

”Brugeren vælger hvor programmet skal starte fra og han kan derfra steppe gennem koden.”

3.3.3 Use Case 19: Se DSL manual.

Scope: IDE.

Niveau: User-goal.

Frekvens: Brugerbestemt.

Primær Aktør: Brugeren, Tekniker.

Sekundære aktører: Systemet.

Stakeholders og interessenter:

RoboGO, de har interesse i at en tidligere opgave som blev udført manuelt bliver automatiseret.

Preconditions:

PC'en er tændt og programmet er startet op.

Inde i programmet er IDE delen valgt.

Postconditions:

En Webbrowser åbnes med hvilke kommandoer der er til rådighed.

Main success scenarie:

1. Brugeren vælger Help
2. Brugeren vælger View Commands.
3. Webbrowser med kommandoer kommer frem.

Extensions:

Ingen.

Ikke-funktionelle krav:

Ingen.

3.3.4 Use Case 20: Indstille editor.

Main success scenarie:

”Brugeren interagerer med brugergrænsefladen og får vist forskellige indstillinger for, hvordan han kan indstille farveskema og font for editorens tekst.”

3.3.5 Use Case 21: Syntaks tjekke.

Main success scenarie:

”Brugeren interagerer med brugergrænsefladen og får markeret i koden, hvor der er syntaksfejl.”

3.3.6 Use Case 22: Bruge makroer.

Main success scenarie:

”Brugeren vælger en kode makro og indholdet bliver sat ind i editoren.”

3.3.7 Use Case 23: Bruge genvejstaster.

Main success scenarie:

”Brugeren trykker på en tastatur genvej, og den tilhørende kommando vil blive udført.”

3.3.8 Use Case 24: Bruge intellisense.

Main success scenarie:

”Mens brugeren skriver noget kode vil der komme forslag til hvad brugeren måske ville skrive. Brugeren kan vælge en af mulighederne og få det indsat.”

3.3.9 Use Case 25: Åbne/lukke flere filer.

Use Case Navn: Som bruger vil jeg kunne åbne flere filer. Lukke en fil.

Scope: IDE

Niveau: User-goal

Frekvens: Brugerbestemt.

Primær Aktør: Brugeren, Tekniker

Sekundære aktører: Systemet

Stakeholders og interessenter:

Ingen.

Preconditions:

IDE kører.

Postconditions:

En eller flere filer er blevet åben. En fil er blevet lukket.

Main success scenarie:

1. Brugeren vælger ”Open file”.
2. Brugeren vælger en eller flere filer.
3. Filerne bliver åbnet.
4. Brugeren vælger den fil/tab han vil lukke.
5. Brugeren vælger ”Close file”.

6. Filen bliver lukket.

Extensions:

2.1 Der er ingen gemte filer:

1. Brugeren vælger i stedet ”New file”.
2. En ny fil vil blive lavet og vist.

2.2 Bruger bruger en genvejstast for at åbne ny tab:

1. En ny blank tab vil blive vist.

Ikke-funktionelle krav:

Ingen.

3.3.10 Use Case 26: Gemme filer.

Scope: IDE.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og Interessenter:

RoboGO; de er interesseret i at se, at det er muligt at gemme filer, som kan bruges senere hen eller arbejdes videre på i fremtiden.

Preconditions:

Styresystemet er oppe og køre, og brugeren befinder sig i IDE-fanen, samt der er skrevet noget tekst/kode i editoren.

Postconditions:

Brugeren får gemt teksten/koden.

Main success scenarie:

1. Brugeren interagerer med brugergrænsefladen og åbner for IDE-fanen.
2. Brugeren skriver kode i editoren.
3. Brugeren gemmer filen med bestemt filnavn ved brug af menuen.

Extensions:

3.1 Brugeren anvender genvejstast til at gemme med.

1. Filen gemmes via genvejstast (CTRL+S).
2. Bruger angiver filnavn, hvorefter filen gemmes.

Ikke funktionelle krav:

Filen skal gemmes med filtypen .txt.

3.3.11 Use Case 27: Søge og erstatte ord.

Main success scenarie:

"Brugeren interagerer med brugergrænsefladen, hvorfra han kan søge efter et bestemt ord eller sætning, og vælger om den skal erstattes med noget andet. Han vil så steppe igennem hver eneste ord/sætning og eventuelt erstatte ordet."

3.3.12 Use Case 28: Udskrive filer.

Main success scenarie:

"Brugeren interagerer med brugergrænsefladen eller bruger en genvejstast til at få en standard print dialog."

3.3.13 Use Case 29: Påmindet om at gemme.

Main success scenarie:

"Brugeren lukker enten editoren, skifter fil eller lukker programmet ned. Hvis filen er blevet ændret siden sidst den er blevet gemt, vil der komme en besked frem, som informerer brugeren om dette. Brugeren vil så kunne vælge at gemme før han fortsætter med sin handling."

3.3.14 Use Case 30: Aktivere tekstombrydning.

Main success scenarie:

"Brugeren interagerer med brugergrænsefladen og kan derefter indstille, om der skal være tekstombrydning eller ej. Når brugeren så skriver kode, vil teksten automatisk ombryde for at passe til en bestemt margin."

3.3.15 Use Case 31: Have auto-indentation.

Main success scenarie:

"Brugeren skriver noget kode og editoren indenter automatisk koden."

3.3.16 Use Case 32: Have flere filer åbne.

Main success scenarie:

”Brugeren åbner en fil som normalt efter allerede at have åbnet en eller han åbner flere på en gang. Alle filerne vil derefter være til rådighed inde fra editoren, uden at skulle finde frem til filerne igen.”

3.3.17 Use Case 33: Have file explorer.

Main success scenarie:

”Brugeren ser en liste over åbne filer i IDE´en og kan derfra vise filerne i editoren.”

3.3.18 Use Case 34: Have autosave.

Main success scenarie:

”Koden i editoren vil automatisk blive gemt med et bestemt interval, hvis brugeren har valgt dette.”

3.3.19 Use Case 35: Have undo/redo.

Main success scenarie:

”Brugeren interagerer med brugergrænsefladen eller bruger en genvejstast, som gør at koden kommer 1 til flere ændringer tilbage eller frem i tiden(Hvis har gået tilbage i tiden for filen).”

3.3.20 Use Case 36: Have syntaks highlighting.

Main success scenarie:

”I editoren vil brugeren få markeret nøgleord/keywords i DSL koden.”

3.3.21 Use Case 37: Have DSL arbejdsområde.

Use Case Navn: Have DSL arbejdsområde

Scope: IDE

Niveau: User-goal

Frekvens: Efter brugerens behov

Primær Aktør: Brugeren, Tekniker

Sekundære aktører: Systemet

Stakeholders og interessenter:

RoboGO, de har interesse i at en tidligere opgave som blev udført manuelt bliver automatiseret.

Preconditions:

SIR programmet kører og IDE starter derfra

Postconditions:

IDE editoren åbnes og funktionaliteterne er tilgængelige

Main success scenarie:

1. Brugeren trykker på ”Filer” i menuen.
2. Derefter vælges ”Ny Script”.
3. Brugeren kan nu skrive på det nye tomt dokument

Extensions:

2. Derefter vælges ”Ny Script”.
    1. Nuværende dokument er ikke gemt.
    2. Use case 29 overtages

Undtagelse:

Ingen.

Ikke-funktionelle krav:

ingen.

3.3.22 Use Case 38: Have fil information vist.

Main success scenarie:

”Mens brugeren arbejder i IDE´en vil han kunne se informationer for filen..”

4. Eksterne grænseflader

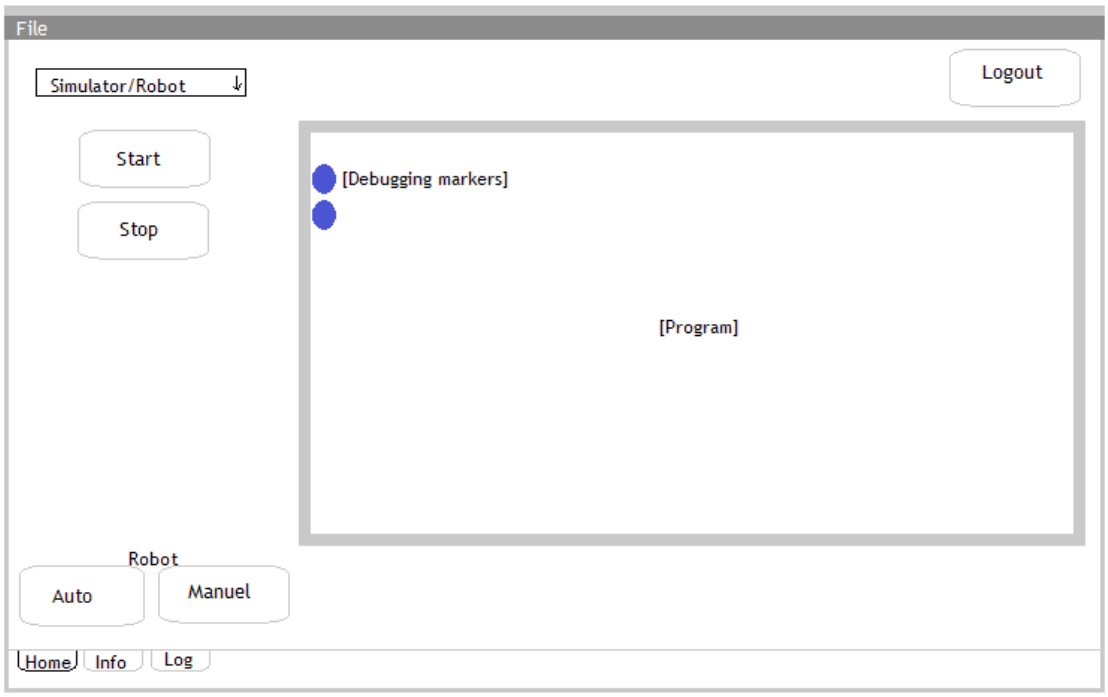
4.1 Bruger-grænseflade

De primære krav til brugergrænsefladen er at det skal være læsbart og brugervenligt.

Brugergrænsefladen laves via WPF (Windows Presentation Foundation), som er et grafisk computer software subsystem, som er en del af .NET Frameworket.

Brugergrænsefladen kunne se således ud for styresystemet, IDE og simulator:

Styresystemet (home tab):

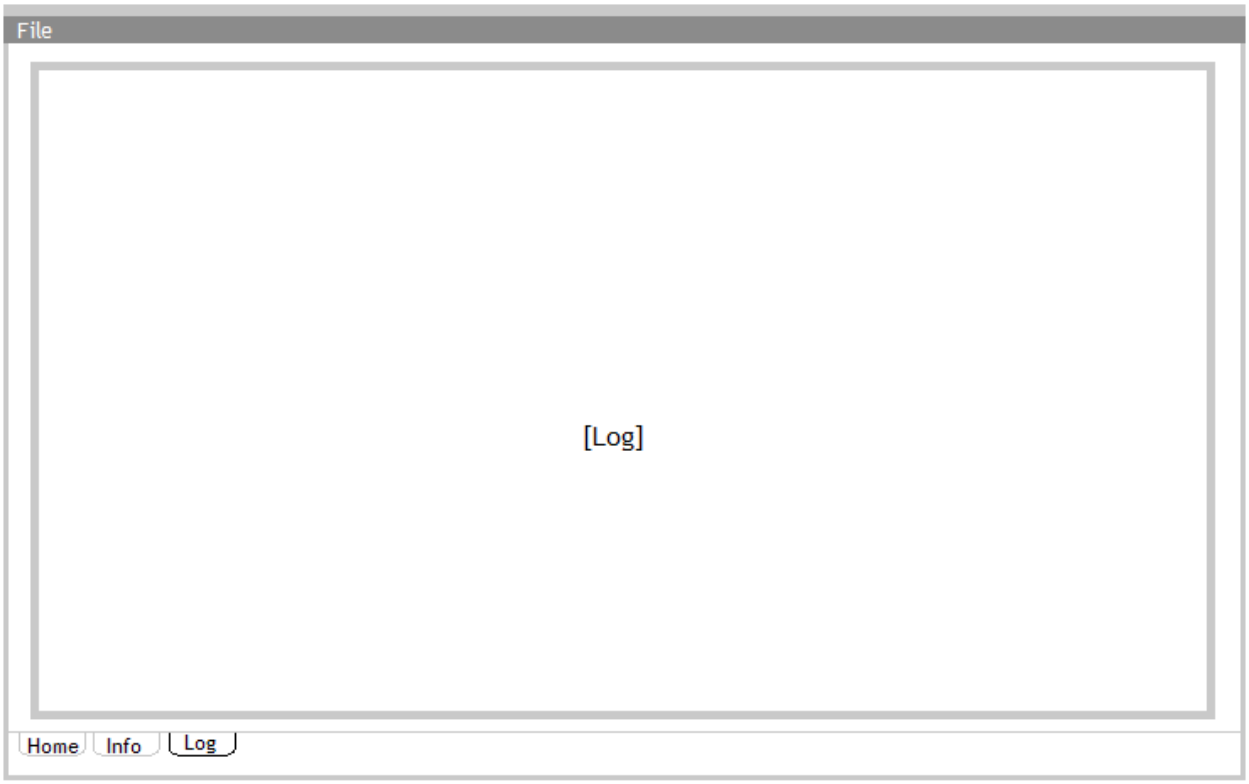


Figur 4: Viser eksempel for GUI af styresystem – Home tab.

Styresystemet (info tab):

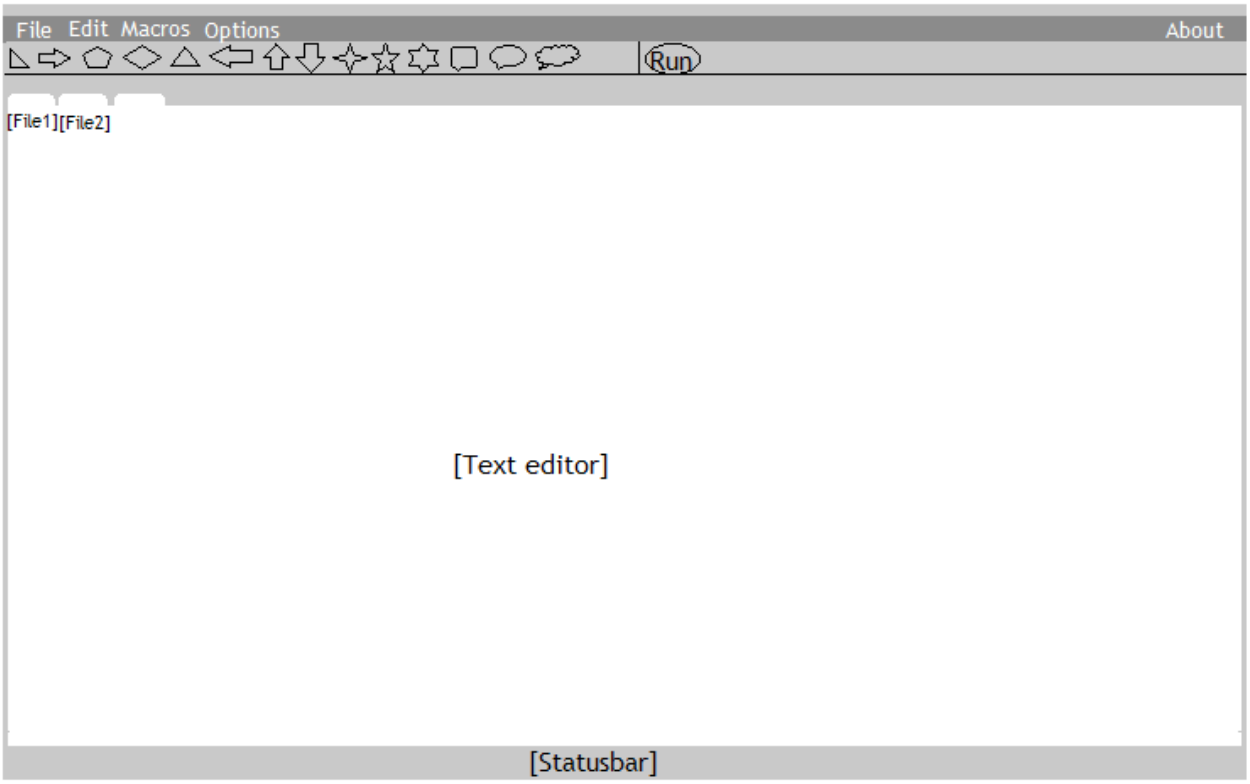
Figur 5: Viser eksempel for GUI af styresystem – Info tab.

Styresystemet (log tab):



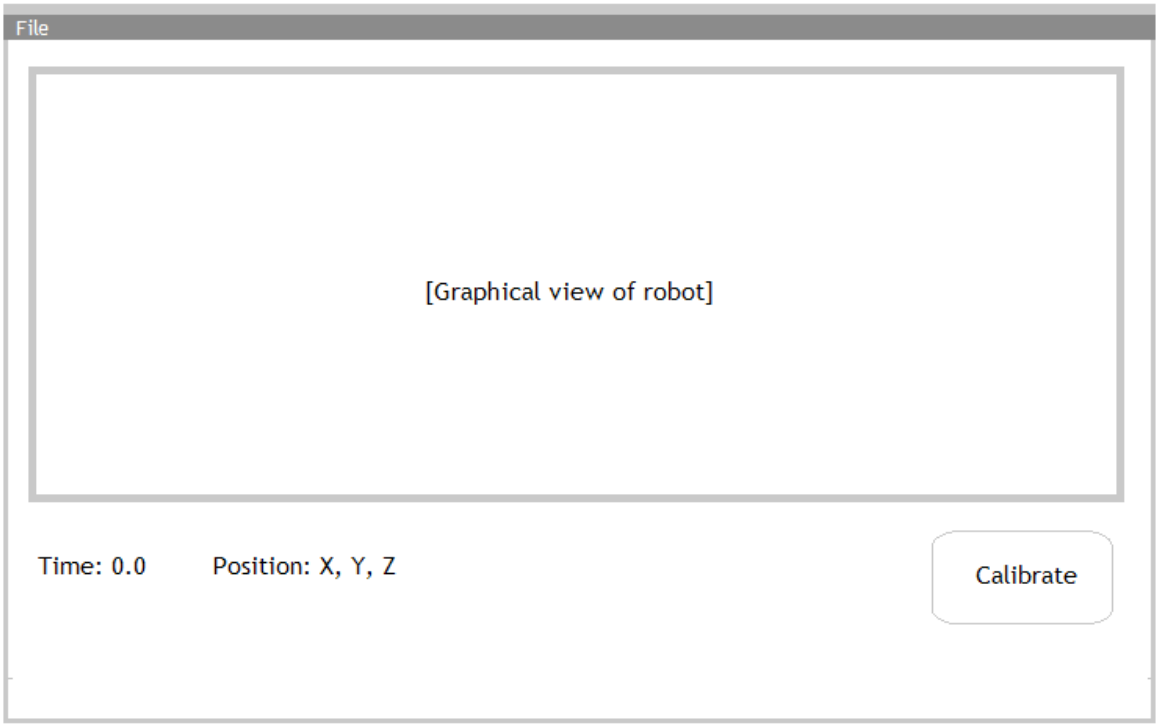
Figur 6: Viser eksempel for GUI af styresystem – Log tab.

IDE:



Figur 7: Viser eksempel for GUI af IDE.

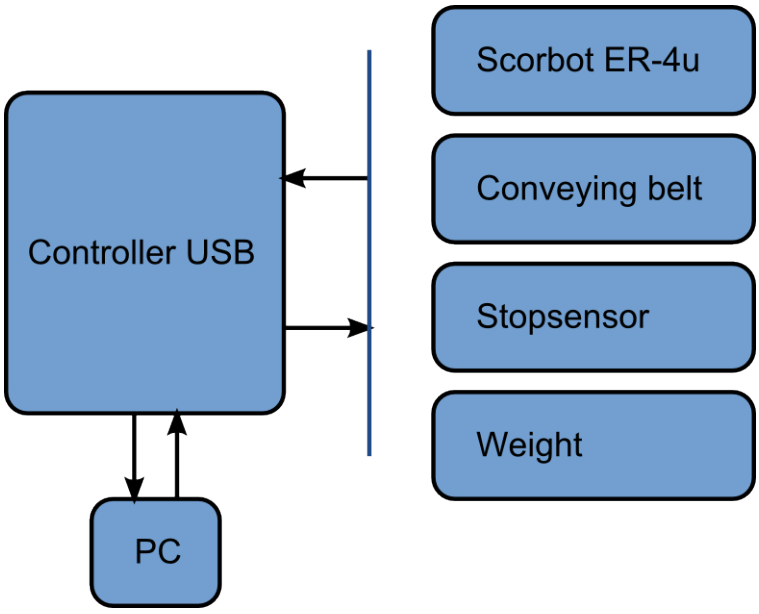
Simulator:



Figur 8: Viser eksempel for GUI af simulator.

4.2 Hardware-grænseflade

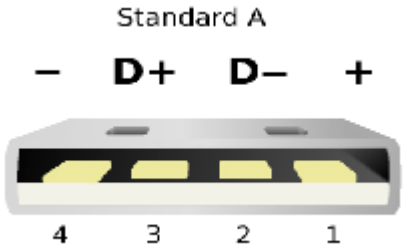
Robotten, transportbåndet, stopsensor og vejecelle, er alle forbundet til henholdsvis USB-controller's analoge og digitale indgange m.m. USB-controlleren er derefter forbundet til PC'en gennem en USB-forbindelse.



Figur 9: Viser hardware-grænseflade.

4.3 Kommunikations-grænseflade

Det bruges to forskellige kommunikations protokoller, den ene er fra PC ↔ USB Controller, hvor der bliver brugt en USB forbindelse.



Figur 10: Viser USB forbindelse.

Den anden protokol er fra PC ↔ Database – der bliver der brugt en ETHERNET forbindelse, som kan være trådløs eller via kabel til et netværk, hvor databasen er forbundet.

4.4 Software-grænseflade

Til styring af robotten bliver systemet udviklet til at bruge biblioteket USBC.dll, som er fra SCORBASE, og det skal kunne køre på en computer med minimum disse specifikationer:

Styresystem:	Windows XP(Service Pack 2)
Processor:	1.00 GHz
RAM:	1 GB
Grafik:	DirectX9
.NET Framework	3.0

Kommunikation til databasen foregår via SQL over en TCP forbindelse, da serveren er sat op til at tage sige af relationelle databaser.

5. Krav til systemets ydelse

Responstiden mellem brugeren og brugergrænsefladen skal være mindre ét sekund, for at det skal være optimalt for brugeren at bruge programmet. I IDE komponenten skal denne responstid være mindre end ét halvt sekund ved redigering af DSL-filer.

6. Kvalitetsfaktorer

Systemet skal være pålideligt forstået på den måde, at klodserne skal placeres med høj præcision og nøjagtighed. Systemet skal samtidig fungere fejlfrit og uden overvågning, dvs. den skal kunne passe sig selv.

Betjening:
Systemet skal være intuitivt at benytte og må ikke være alt for kompliceret. Det skal være nemt for brugeren at kunne kode til robotten og få robotten til at udføre den ønskede funktionalitet.
Integritet:
Systemets data skal være persistent, dvs. brugerens gemte filer, data i databasen m.m. skal være lagret, selvom systemet slukkes eller genstartes.

7. Designkrav

Følgende punkter beskriver følgende designkrav til SIR:

- Robotsystemet skal implementeres med:
    1. En PC med Windows OS (XP[Service pack 2] eller senere OS version).
    2. En processor på mindst 1.0 GHz eller højere.
    3. Harddisk med mindst 20 MB fri plads samt mindst 128 MB RAM
    4. DirectX 9 eller højere.
    5. En mus eller andre former for pegeudstyr samt ledig USB-port.
- Endvidere skal der til oprettes en database, hvori relevante data fra processen kan lagres og hentes.
- En IDE (Integrated Development Enviroment), hvor man kan fremstille programmer til styring af robotten. Det forudsættes, at man designer og implementerer et programmeringssprog og en fortolker til dette.

- En simulator, som kan kobles på systemet i stedet for den fysiske robot. Dette vil til fordel kunne anvendes i forbindelse med test.
- Et styreprogram med en grafisk brugergrænseflade lavet i WPF.
  Ved at bruge brugergrænsefladen skal man kunne starte og stoppe systemet, aflæse status for sorteringsprocessen, få vist relevante data og arbejde med data i databasen.
- Designe brugergrænsefladerne for komponenterne, som skal godkendes af softwareholdets vejleder.

8. Andre krav

8.1 Myndighedskrav

Hardwaredelen af produktet, som består af robotten og tilhørende udstyr som transportbånd og controller, skal opfylde bekendtgørelsen for tekniske hjælpemidler: https://www.retsinformation.dk/Forms/R0710.aspx?id=67407

Ved nogle af punkterne ligger ansvaret ved producenten af SCORBOTTEN, ting som sikkerhedsmanualen og mulighed for nød-stop.
Ved den senere opstilling af systemet for test og ved brug af systemet i arbejdsøjemed er vi ansvarlige for, at den opfylder bekendtgørelsen samt normale arbejdsmiljøkrav.

8.2 Øvrige krav

Brugerne af systemet skal før brug have kursus i brug af systemet samt have læst SCORBOTTEN´s medførende sikkerhedsmanuel.

9. Delleveringer

- ***Dellevering 1. 08/03/2012***
  Kravspecifikation:
  For at få et overordnet overblik over projektet afleveres en kravspecifikation.
  Der afleveres i denne kravspecifikation use cases med beskrivelser i brief format.
  Accepttesten:
  Der afleveres et udkast af denne uden det store indhold af selve testprocedure.
- Sprint 1 - 30. marts 2012

- Sprint 2 - 20. april 2012
- Sprint 3 - 11. maj 2012
- Projekt aflevering - 01. juni 2012

10. Bilag

10.1 Ordliste

Ingen.

10.2 Datadefinitioner

Ingen.