

Projekt: Sorting Industrial Robot

Dato: 01-06-2012

Titel:

**Kravspekifikation
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 use cases fra branches
1.2	23-04-12	CTD, SLT	Merge af alle use cases fra alle branches og ændret alle use cases, så de har samme format.
1.3	14-05-12	SLT	Revideret kravspecifikation efter review møde
1.4	23-05-12	SLT	Revidering til endelige bestemte use cases.
2.0	01-06-12	SLT, RHT	Endelig version.

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 Røvsing
Projektnummer:	1
Dokument-id:	Kravspecifikation.odt
(filnavn)	Kravspecifikation.pdf
Antal sider:	35
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 Røvsing

Indholdsfortegnelse

1. Indledning.....	4
1.1 Formål.....	4
1.2 Referencer.....	4
1.3 Læsevejledning.....	4
2. Generel beskrivelse.....	5
2.1 Systembeskrivelse.....	5
2.1.1 Systemoversigt.....	6
2.1.2 Aktør-kontekst diagram.....	7
2.1.3 Aktør beskrivelser.....	7
2.2 Systemets funktioner.....	9
2.2.1 Use Case diagram.....	9
2.3 Systemets begrænsninger.....	10
2.4 Systemets fremtid.....	11
2.5 Brugerprofil.....	11
2.6 Krav til udviklingsforløbet.....	11
2.7 Omfang af kundeleverance.....	12
2.8 Forudsætninger.....	12
3. Funktionelle krav – Use Cases.....	13
3.1 Styresystem:.....	14
3.1.1 Use Case 1: Styre klodsplacering.....	14
3.1.2 Use Case 2: Tjekke loggen.....	15
3.1.3 Use Case 3: Foretage manuel styring.....	16
3.1.4 Use Case 4: Skifte mellem robot og simulator.	17
3.1.5 Use Case 5: Se klodser lagret.....	18
3.1.6 Use Case 6: Logge ind.....	19
3.2 Simulator:.....	20
3.2.1 Use Case 7: Eksekvere koden i simulatoren.....	20
3.3 IDE:.....	21
3.3.1 Use Case 8: Se DSL manual.....	21
3.3.2 Use Case 9: Have flere filer åbne	22
3.3.3 Use Case 10: Gemme filer.....	23
3.3.4 Use Case 11: Eksekvere DSL kode.	24
3.4 Vægt.....	25
3.4.1 Use Case 12: Måle vægten af objekter.....	25
4. Eksterne grænseflader.....	26
4.1 Bruger-grænseflade.....	26
4.2 Hardware-grænseflade.....	29
4.3 Kommunikations-grænseflade.....	29
4.4 Software-grænseflade.....	29
5. Krav til systemets ydelse.....	31
6. Kvalitetsfaktorer.....	31
8. Andre krav.....	32

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

Udleverede materialer er vedlagt bilag i form af links:

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

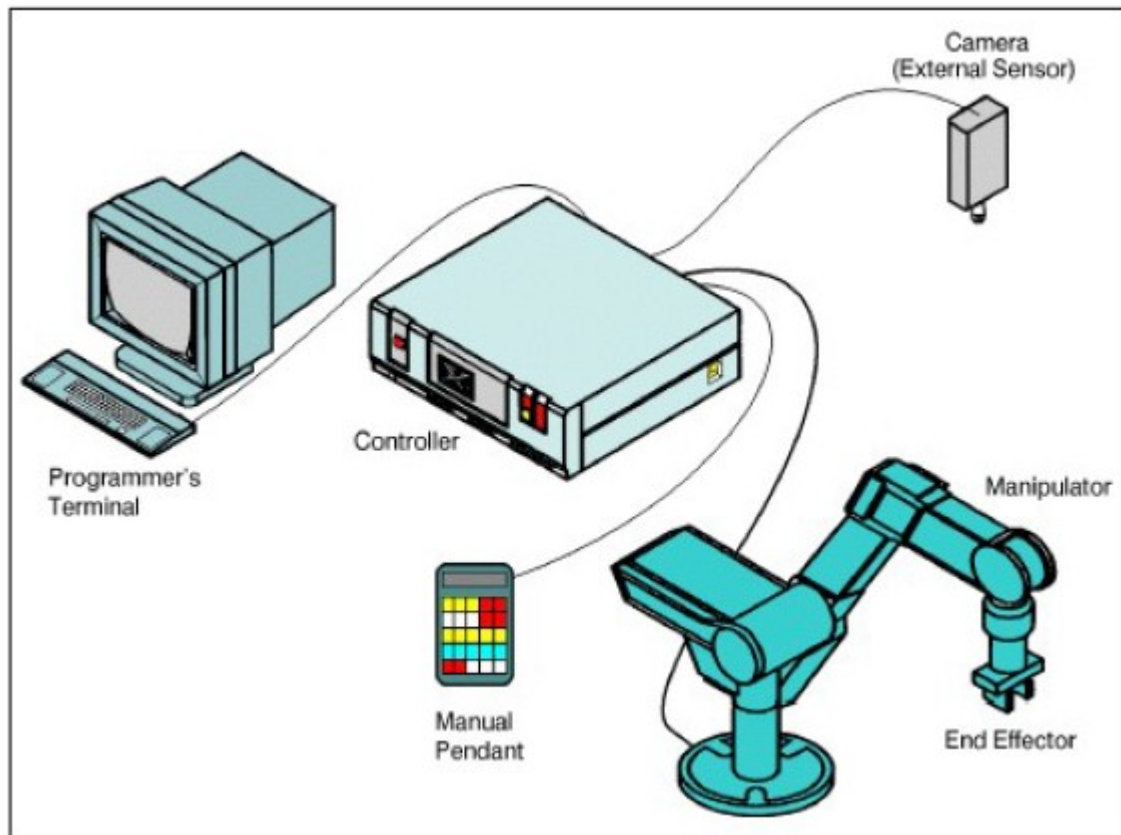
Vores projektopgave omhandler, at vi skal automatisere en sorteringsproces, der indtil nu er blevet udført manuelt. Problemet løses ved at udvikle robotsoftware til en robot, der automatisk kan sortere emner efter massefylde og dimensioner. Vores system har vi valgt at kalde Sorting Industrial Robot (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.

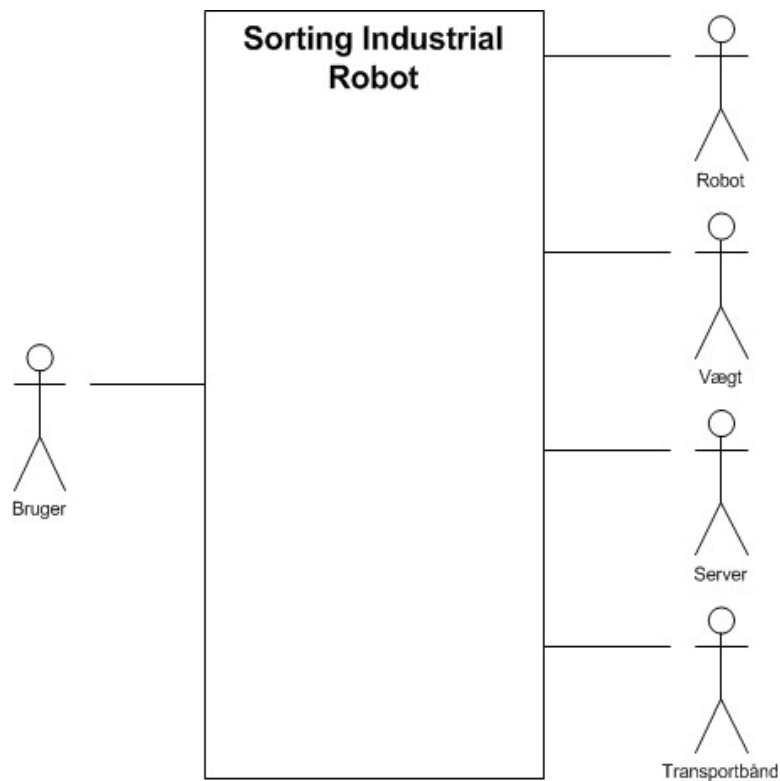
Controlleren kommunikerer med Robotten, en Terminal og en Vejecelle.

2.1.1 Systemoversigt



Figur 1: Visuel beskrivelse af vores system.

2.1.2 Aktør-kontekst diagram



Figur 2: Viser 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:	Transportbånd
Type:	Sekundær
Beskrivelse:	En aktør, der hjælper robotsystemet med at registrere klodsen samt flytte denne. I denne er indbygget en sensor til registrering.
Antal samtidige aktører:	1

Aktør navn:	Vægt
Type:	Sekundær
Beskrivelse:	En aktør, som er med til at indlæse vægten af klodsen.
Antal samtidige aktører:	1

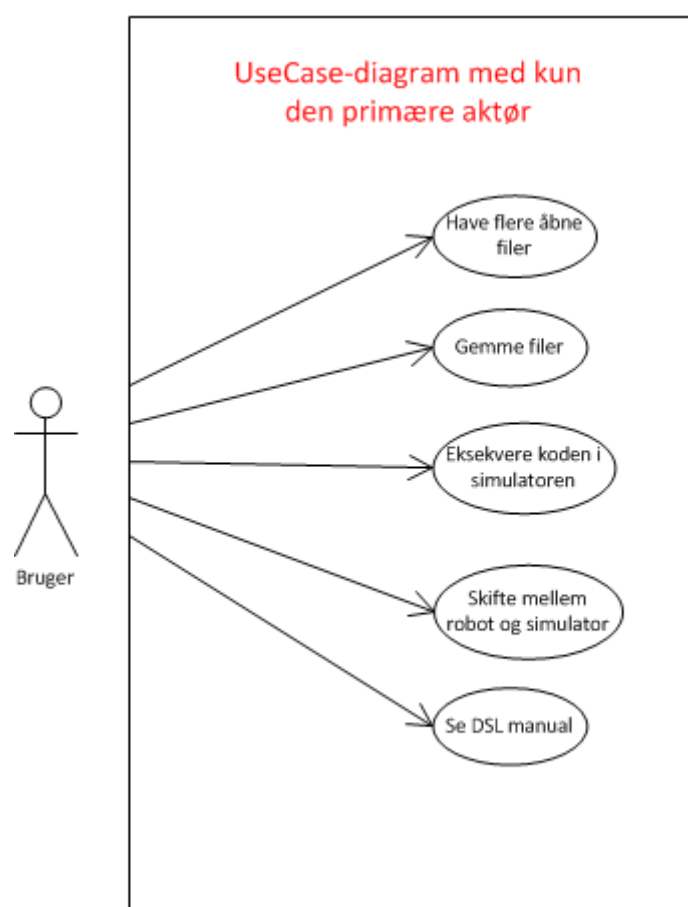
Aktør navn:	Robot
Type:	Sekundær
Beskrivelse:	En aktør, som modtager inputs til diverse funktionaliteter
Antal samtidige aktører:	1

Aktør navn:	Server
Type:	Sekundær
Beskrivelse:	En aktør, der hjælper robotsystemet til at holde styr på hvilke klodser og komponenter der er til rådighed.
Antal samtidige aktører:	1

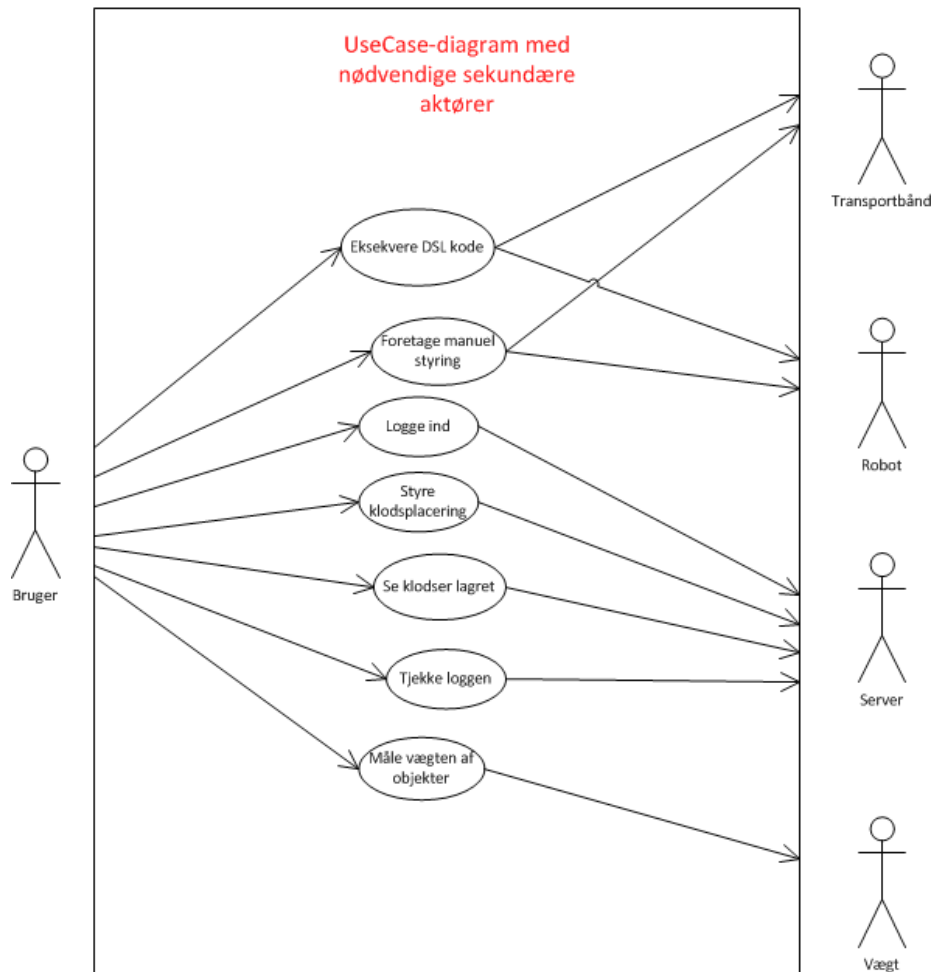
2.2 Systemets funktioner

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 for kun primær aktør.



Figur 4: Viser use-case diagram for med den primær aktør og de nødvendige sekundær aktører.

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 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 én brugerprofil:

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 for at kunne styre robotten.

2.6 Krav til udviklingsforløbet

Projektet udvikles efter en iterativ udviklingsproces, UP, hvor vi indtil videre har fire 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.
- Systemet er oppe at køre, samt at det er initieret.
- Databasen er tilgængelig og man er forbundet til den.

3. Funktionelle krav – Use Cases

Use casene er delt op tre 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: Styre klodsplacering.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Server.

Stakeholders og interessenter:

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

Preconditions:

Ingen

Postconditions:

Brugeren får ændret værdierne i databasen og får dem gemt.

Main success scenarie:

1. Brugeren går ind i databasen og ændrer værdier.
2. Brugeren gemmer værdierne.
3. Værdierne gemmes.

Undtagelser:

Ingen.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

3.1.2 Use Case 2: Tjekke loggen.

Scope: Styresystem

Niveau: Bruger

Frekvens: Brugerbestemt

Primær Aktør: Bruger

Sekundære aktører: Server.

Stakeholders og interessenter:

RoboGO: de har interesse i, at en proces logges, så der kan ses på, hvad der er lavet.

Preconditions:

Ingen

Postconditions:

Der laves en log, som logger en liste af events, der er forekommet.

Main success scenarie:

1. Brugeren tjekker loggen.
2. Brugeren kan se loggen over events.

Undtagelser:

Ingen.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

3.1.3 Use Case 3: Foretage manuel styring.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Robot, Transportbånd.

Stakeholders og interessenter:

Ingen.

Preconditions:

Ingen

Postconditions:

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

Main success scenarie:

1. Brugeren vælger manuel styring.
2. Brugeren interagerer med brugergrænsefladen med musen.
3. Robotten bevæger sig efter brugerens ønske.

Undtagelser:

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

1. Robotten bevæger sig ikke.

Extensions:

Ingen

Ikke funktionelle krav:

Styringen foregår ikke via 'toggle' , dvs. brugeren skal holde musen nede i manuel styring eller tasten nede på tastaturet, da robotten vil stoppe, når den bliver sluppet.

3.1.4 Use Case 4: Skifte mellem robot og simulator.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og interessenter:

Ingen.

Preconditions:

Ingen

Postconditions:

Der skiftes til robotten fra simulatoren.

Main success scenarie:

1. Brugeren tjekker forbindelsen mellem GUI og simulator.
2. Brugeren skifter til robot.
3. Brugeren tjekker forbindelsen mellem GUI og robot.
4. Der er forbindelse til robotten.

Undtagelser:

4.1 Der er ikke forbindelse til robotten.

1. Der indikeres, at der ikke er forbindelse.

Extensions:

Ingen.

Ikke funktionelle krav:

Systemet skal starte med simulatoren som førstevalg.

3.1.5 Use Case 5: Se klodser lagret.

Scope: Styresystem

Niveau: Bruger

Frekvens: Brugerbestemt

Primær Aktør: Bruger.

Sekundære aktører: Server.

Stakeholders og interessenter:

Ingen.

Preconditions:

Ingen

Postconditions:

Brugeren får vist en liste af klodserne, der er blevet placeret.

Main success scenarie:

1. Brugeren beder om klodsernes informationer.
2. Brugeren får en liste af klodserne, der er placeret.

Undtagelser:

Ingen.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

3.1.6 Use Case 6: Logge ind.

Scope: Styresystem

Niveau: Bruger

Frekvens: Brugerbestemt

Primær Aktør: Bruger

Sekundære aktører: Server.

Stakeholders og interessenter:

RoboGO: de er interesseret i at få lavet sikkerhed, så der er en nødvendighed for login.

Preconditions:

Ingen

Postconditions:

Brugeren logger ind.

Main success scenarie:

1. Brugeren skriver brugernavn og kodeord.
2. Brugeren logger ind.

Undtagelser:

2.1 Brugeren skriver ugyldigt brugernavn og kodeord.

1. Der fremkommer besked om ugyldigt kodeord og brugernavn.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

3.2 Simulator:

3.2.1 Use Case 7: Eksekvere koden i simulatoren.

Scope: Simulator.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og interessenter:

RoboGO: de har interesse i at se en fysisk proces blive simuleret.

Preconditions:

Editoren indeholder kørbare kode, og syntaksen for koden er overholdt.

Postconditions:

DSL-koden bliver kørt via simulatoren, hvor der vil blive vist/skrevet, hvad robotten(virtuelt) gør ud fra koden.

Main success scenarie:

1. Brugeren skriver sin kode.
2. Koden bliver eksekveret i simulatoren.

Undtagelser:

Ingen.

Extensions:

1.1 Brugeren åbner en eksisterende fil:

1. Koden bliver indlæst til editoren.

Ikke funktionelle krav:

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

3.3 IDE:

3.3.1 Use Case 8: Se DSL manual.

Scope: IDE.

Niveau: Bruger

Frekvens: Brugerbestemt.

Primær Aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og interessenter:

RoboGO: de har interesse i at kunne hente hjælp fra softwaren mht. kommandoer.

Preconditions:

Ingen.

Postconditions:

En liste med kommandoer åbnes til rådighed.

Main success scenarie:

1. Brugeren vælger "Help".
2. Brugeren vælger "View Commands".
3. En liste med kommandoer kommer frem.

Undtagelser:

Ingen.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

3.3.2 Use Case 9: Have flere filer åbne

Scope: IDE

Niveau: Bruger

Frekvens: Brugerbestemt.

Primær Aktør: Bruger.

Sekundære aktører: Ingen.

Stakeholders og interessenter:

Ingen.

Preconditions:

Der er filer tilgængelige til at blive åbnet og indlæst.

Postconditions:

Flere filer vil blive åbnet.

Main success scenarie:

1. Brugeren vælger at åbne flere filer.
2. Filerne bliver åbnet.

Undtagelser:

Ingen.

Extensions:

1.1 Bruger anvender en genvejstast for at åbne flere filer.

1. Brugeren markerer valgte filer.
2. Filerne bliver åbnet.

Ikke funktionelle krav:

Filerne, der åbnes, skal være af typen .py.

3.3.3 Use Case 10: 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:

Ingen.

Postconditions:

Brugeren får gemt koden.

Main success scenarie:

1. Brugeren skriver kode i editoren.
2. Brugeren gemmer filen med bestemt filnavn.

Undtagelser:

Ingen.

Extensions:

2.1 Brugeren anvender genvejstast.

1. Filen gemmes via genvejstast.
2. Brugeren angiver filnavn.
3. Filen gemmes.

Ikke funktionelle krav:

Filen skal gemmes med filtypen .py. Der kan endvidere vælges filsti.

3.3.4 Use Case 11: Eksekvere DSL kode.

Scope: IDE.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Robot, Transportbånd.

Stakeholders og interessenter:

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

Preconditions:

Det forudsættes, at koden har overholdt syntaksen.

Postconditions:

Brugeren vil kunne eksekvere kode.

Main success scenarie:

1. Brugeren skriver kode i editoren.
2. Brugeren eksekverer koden.

Undtagelser:

Ingen.

Extensions:

- 1.1 Brugeren åbner en fil med kørbart DSL kode.
1. Brugeren eksekverer koden.

Ikke funktionelle krav:

Ingen.

3.4 Vægt

3.4.1 Use Case 12: Måle vægten af objekter.

Scope: Styresystem.

Niveau: Bruger.

Frekvens: Brugerbestemt.

Primær aktør: Bruger.

Sekundære aktører: Vægt, Robot.

Stakeholders og interessenter:

RoboGO; de har interesse i at måle vægten af et objekt.

Preconditions:

Der er tilgængelige objekter til stede.

Postconditions:

Vægten af objektet måles.

Main success scenarie:

1. Objektet lægges på vægten.
2. Brugeren beder om målingen.
3. Brugeren får værdien.

Undtagelser:

Ingen.

Extensions:

Ingen.

Ikke funktionelle krav:

Ingen.

4. Eksterne grænseflader

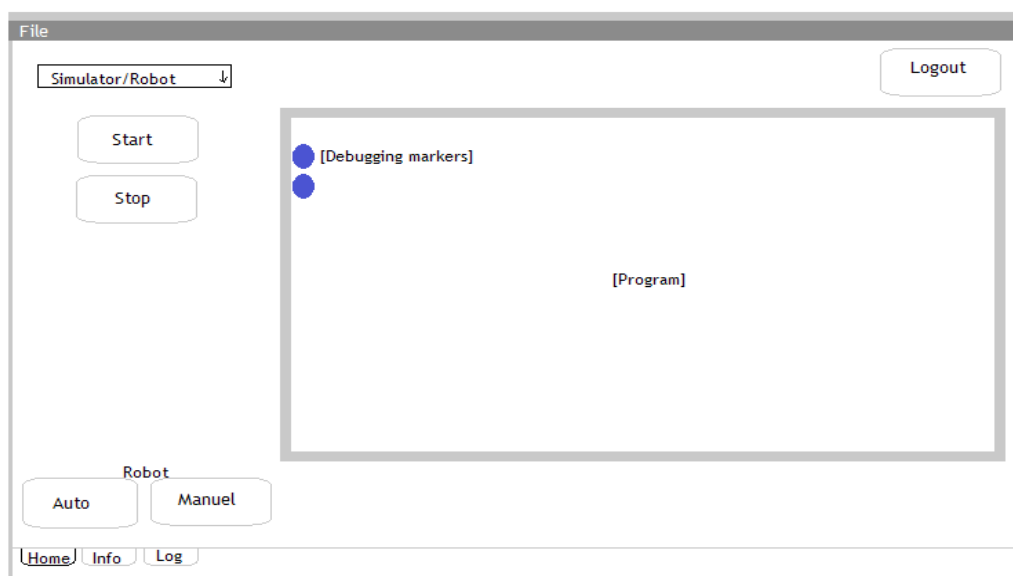
4.1 Bruger-grænseflade

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 er skitseret se således ud for styresystemet, IDE og simulator:

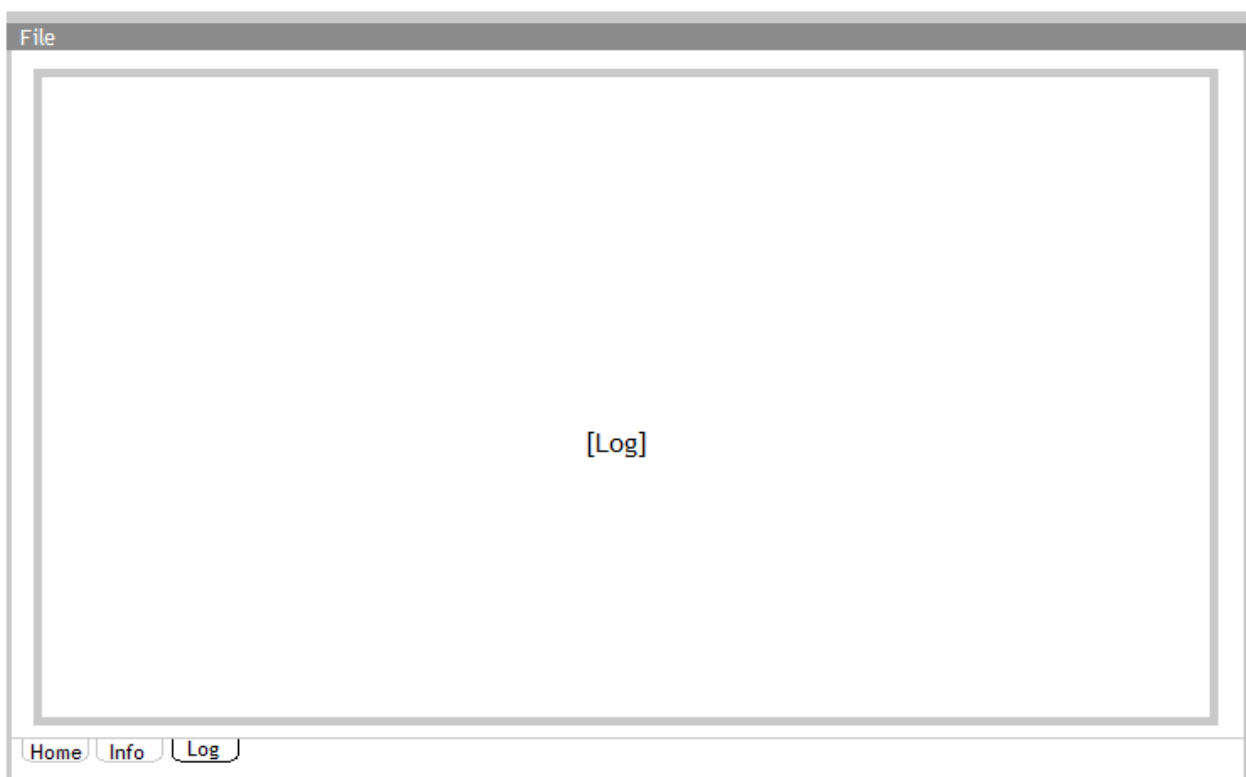
Styresystemet (home tab):



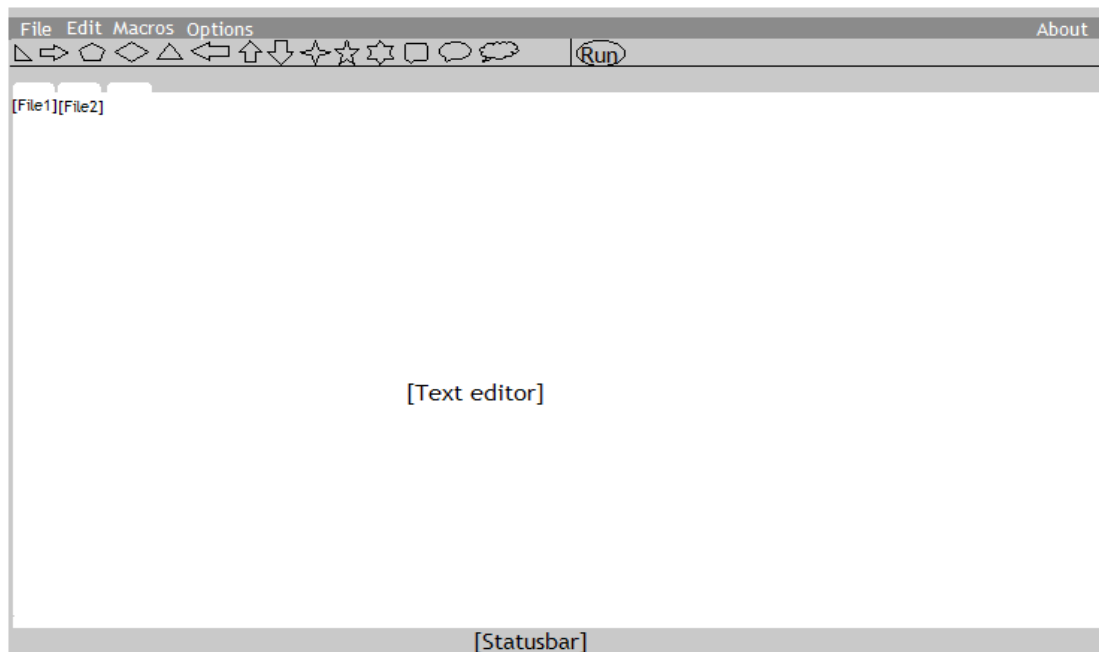
Figur 5: Viser eksempel for GUI af styresystem – Home tab.

Styresystemet (info tab):

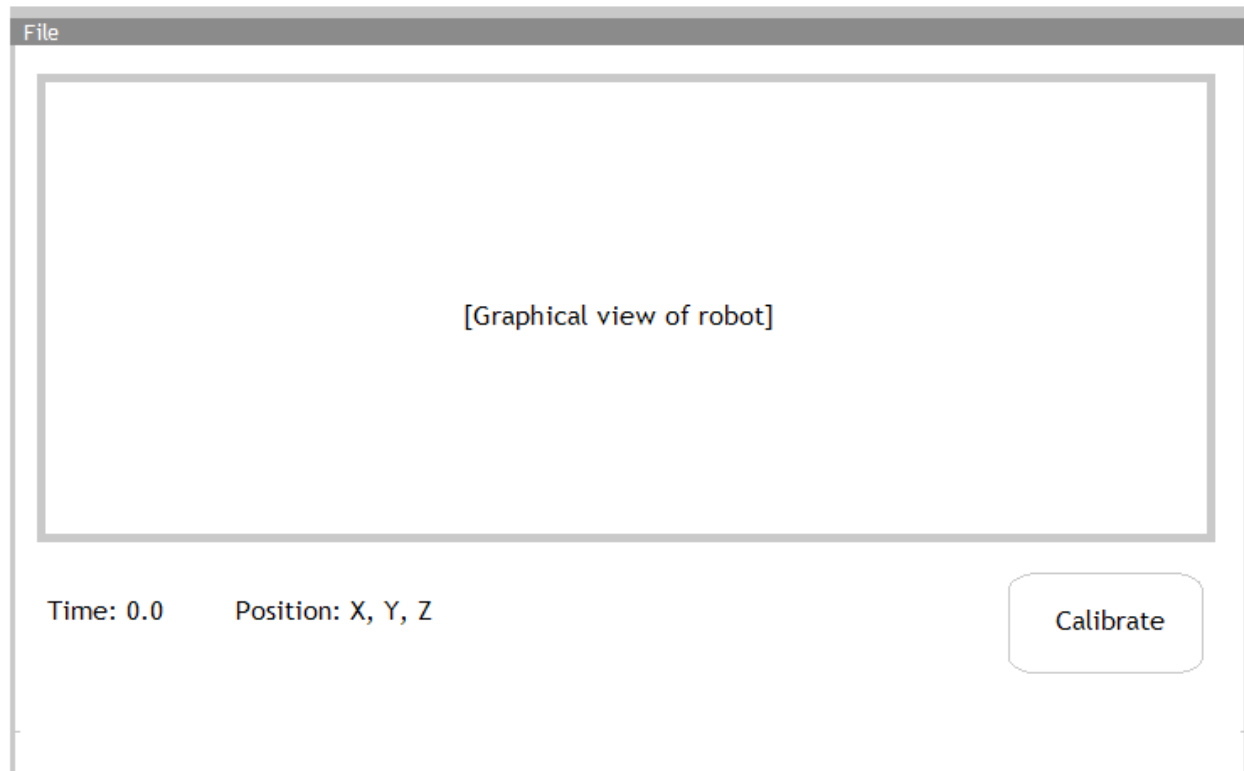
Figur 6: Viser eksempel for GUI af styresystem – Info tab.

Styresystemet (log tab):

Figur 7: Viser eksempel for GUI af styresystem – Log tab.

IDE:

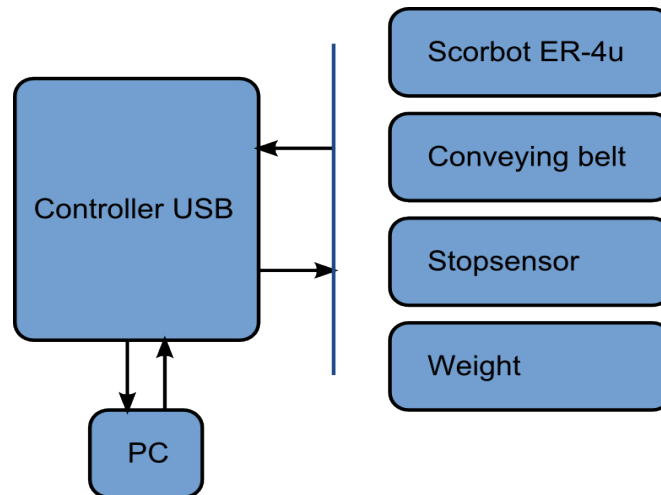
Figur 8: Viser eksempel for GUI af IDE.

Simulator:

Figur 9: Viser eksempel for GUI af simulator.

4.2 Hardware-grænseflade

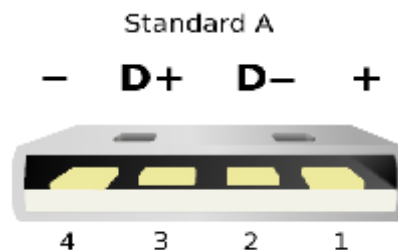
Robotten, transportbåndet, stopsensor og vejecelle, er alle forbundet til henholdsvis USB-controller's analoge og digitale indgange m.m. USB-controlleren er derefter forbundet til PC'en gennem en USB-forbindelse.



Figur 10: 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 11: Viser USB-forbindelse

Den anden protokol er fra PC ↔ Database – der bliver der brugt en netværksforbindelse, som kan være enten 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 sig af relationelle databaser.

5. Krav til systemets ydelse

Responstiden mellem brugeren og brugergrænsefladen skal være mindre end 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

Brugervenligheden har højeste prioritet, dernæst pålidelighed. Endvidere prioriteres effektiviteten og genbrugbarheden højt. Muligheden for udvidelse af systemet er ikke prioriteret højt

Prioriteringsskala fra 1 til 5, hvor 5 er højest:

- | | |
|----------------------|---|
| ○ Brugervenlighed | 5 |
| ○ Pålidelighed | 4 |
| ○ Genbrugbarhed | 4 |
| ○ Effektivitet | 4 |
| ○ Udvidelsesmulighed | 2 |

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 - 02. april 2012

Kravspecifikation:

I dette sprint har der været fokus på at få styresystemet op, og derfor valgte vi use cases, som var centrale for styresystemet jf. product owner. Følgende use case ID blev valgt:

1, 9, 11, 13 og 37.

Accepttesten:

Efterfølgende blev der skrevet en accepttest til denne kravspecifikation, således at systemet kan blive testet gennem nogle testscenarier, der stemmer overens med netop kravspecifikation.

- Sprint 2 - 23. april 2012

Kravspecifikation:

I dette sprint fokuseres der på, at GUI'en skal kunne benyttes. Vi har derfor valgte følgende use cases, som product owner bestemte ville være centrale for sprintet: 4, 12, 17, 19, 25 og 26.

Accepttesten:

Efterfølgende blev der skrevet en accepttest til denne kravspecifikation, således at systemet kan blive testet gennem nogle testscenarier, der stemmer overens med netop kravspecifikation. Endvidere blev der tilføjet yderligere dokumentation og opdateret versionshistorik.

- Sprint 3 - 14. maj 2012

Kravspecifikation:

I dette sprint fokuseres der på 'at have noget, der virker og få databasen oppe og køre samt lave vægten'. Endvidere skulle resterende manglende dokumentation i kravspecifikation skrives. Følgende ID'er blev valgt fra product backlog:

2, 3, 6, 7 og 39.

Accepttesten:

Der blev rettet og skrevet testscenarier for de ovennævnte use cases.

- Projekt aflevering - 01. juni 2012

Endelig aflevering af både kravspecifikation og accepttest.

10. Bilag

10.1 Ordliste

Ingen.

10.2 Litteratur

Her er linksene, der refererer til de udleverede dokumenter samt anvendt boglitteratur:

- **Controller-USB, Users Manual**
[http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100341_D_Controller_USB\(0305\).PDF](http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100341_D_Controller_USB(0305).PDF)
- **Scorbase, Users Manual**
[http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100342-g%20Scorbase-usb-v53\(0602\).pdf](http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100342-g%20Scorbase-usb-v53(0602).pdf)
- **Scorbot-ER 4u, Users Manual**
http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100343-b%20ER_4u.pdf
- **Robotics and Materials Handling 1, Student Activities Book**
[http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100344-1-d-RMH1_AC%20\(4u\)%200204.pdf](http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100344-1-d-RMH1_AC%20(4u)%200204.pdf)
- **RoboCell, Users Manual**
[http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100346-F%20RoboCell-USB-v501%20\(0510\).pdf](http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100346-F%20RoboCell-USB-v501%20(0510).pdf)
- **Datablad for vejecelle**
<http://kurser.iha.dk/eit/i4prj4/>
- **USBC-documentation**
<http://kurser.iha.dk/eit/i4prj4/USBC-documentation.pdf>
- **Projektoplæg**
<http://kurser.iha.dk/eit/i4prj4/Projektoplaeg.doc>

- **"Applying UML and Patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development"**

Blev benyttet til at se på forskellige arkitekturer og designmønstre

- **The Art of Unit Testing with Examples in .NET.**

Blevet benyttet til at sætte standard for enhedstest og få brugbar teori til enhedstest.