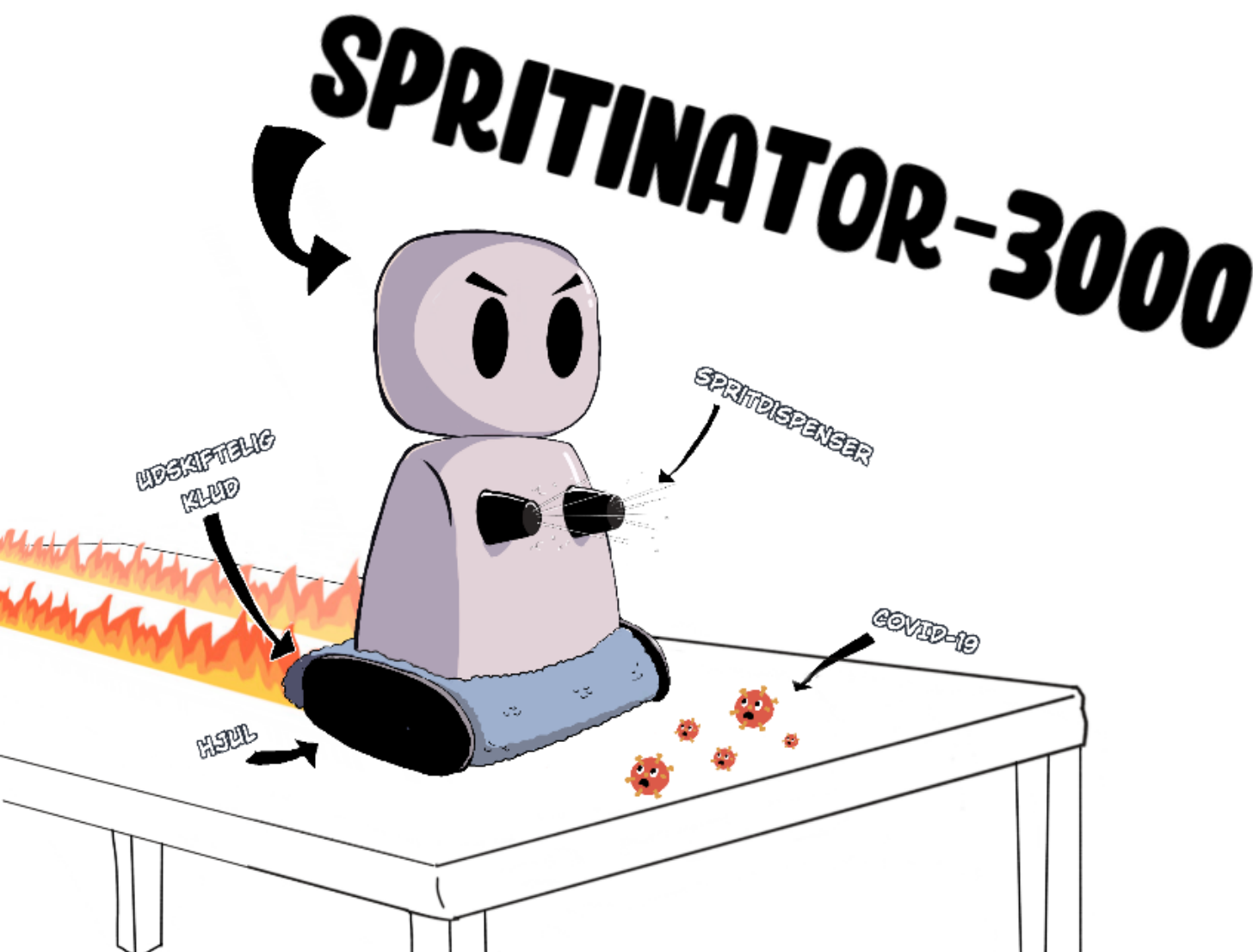


3. semester projekt

Projektformulering




Medlemmer

Murat Eroglu
Asger Busk Breinholm
Simon Shahrour
Yuhu Su Stougaard Nielsen
Erda Naz Cecillia
Magnus Gjøttrup Kristensen
Mathias Old Jensen

Studienummer

201906338
201807859
201810704
201904972
201900001
201908511
201610790

Vejleder

Mads 
Dato
01/01/2020
Gruppe
Nr. 2

Indledning:

Hvis vi tænker på de største problemer der er i verden på nuværende tidspunkt, hvad er så det første man kommer i tanker om? Covid-19. Vi har desværre accepteret at vi ikke kan undgå at tænke på Corona og det eneste vi kan gøre, er at passe på med at sprede den. Derfor er alles hverdage påvirket af situationen og det går udover undervisningen på de fleste uddannelser, da man bliver nødt til at stoppe undervisningen tidligere, for at have tid til at spritte bordene af i lokalet. Men spritter de studerende virkelig af efter sig? Og gør de det rigtigt? Er det en chance vi som et lært sted som et universitet er villig til at tage? Derfor vil vi prøve på at finde på en løsning til dette problem, hvor afspritningen er sikkert udført og de studerende ikke skal udsætte sig til overfladesmitte.

Målet for projekt:

Målet for dette projekt er at løse et problem med vores læreringskrav som en ramme for vores proces. Vi skal bruge viden fra de forrige semestre blandt andet brug af ASE-modellen, use cases og kombineret af viden fra andre fag. Som noget nyt skal vi implementere et embedded system hvor Linux og vores PSoC indgår. Vi skal også bruge DC- eller stepper motorer samt sensorer.

Vores mål er at lave en robot der effektivt kan afspritte et bord og overholde kravene om afspritning fra SSI (Statens Serum Institut).

Beskrivelse af projekt:

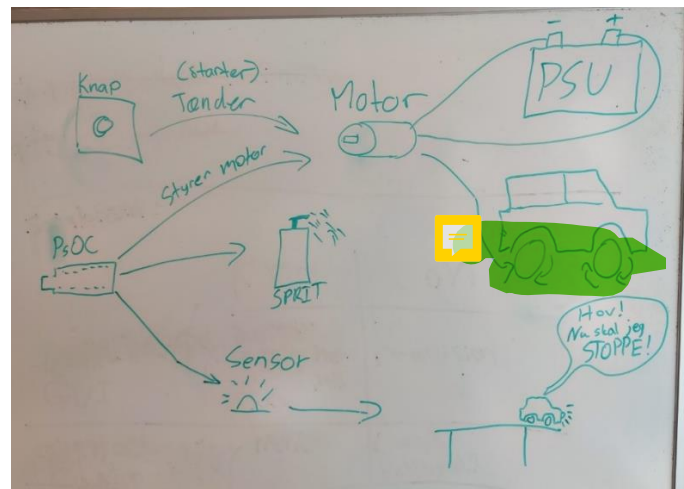
Spritinator-3000 er en selvkørende og selvspitende robot, der kan afspritte bordet systematisk efter folk har siddet og arbejdet eller spist. Pga. Covid-19 er det nødvendigt at afspritte alle overflader man har rørt ved rigtig. Dette kan tage lang tid, og det er ikke altid man husker at gøre dette hver gang man rejser sig, hvis man har travlt eller bare glemmer det. Med et tryk på en knap evt. fra en underviser kan dette problem løses med vores helt egen afspritte-robot. Den tager overfladerne på bordet og afspritter dem. Vi mener at dette er det rigtige valg, fordi at vi har talt med medstuderende som til tider har glemt at spritte af eller gør det forkert. Det er i vores fælles interesse at vi ikke overbelaster sundhedssystemet ved mangelfulde profylaktiske antag.

Spritinator-3000 kan desuden også fungerer som en stationær spritdispenser der gør det sjovt at have god håndhygiejne. På den måde kan den ramme flere målgrupper.

Systemskitse:

Vores systemskitse viser at vi gør brug af disse komponenter til vores udførsel af projekt:

- En PSoC
- En sensor
- En interface
- En motor
- Power supply Unit
- En aktuator til spritdispenseringen



God tegning!

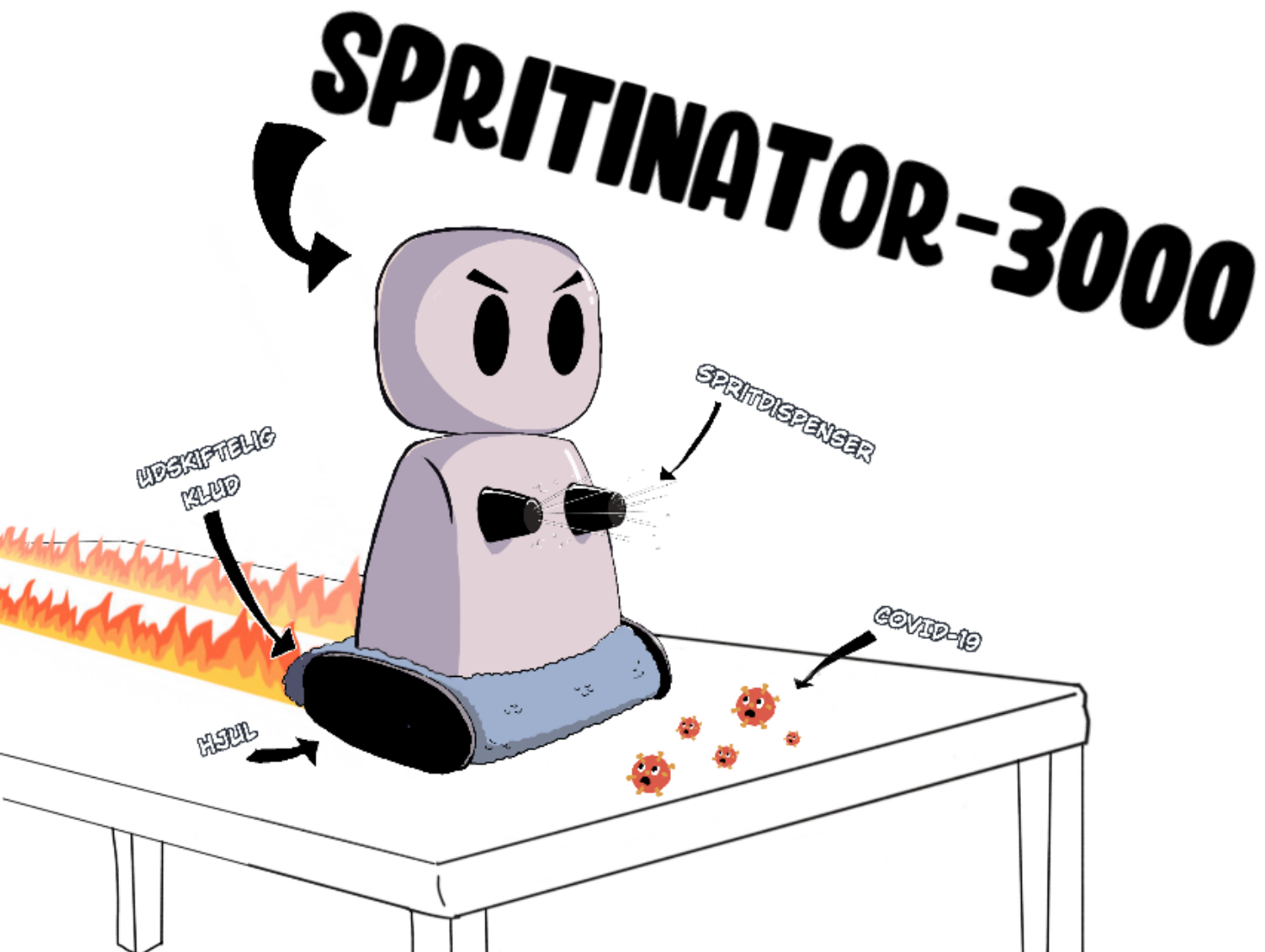
Figur 1: Rigt billede af projektet

Vi ser at vi igennem en PSoC styrer motoren, og motoren får hjulene til at køre, og den styrer intervallet af sprit sprayen. Igennem en knap sættes maskinen i gang. Ved hjælp af sensorer ved robotten hvor langt den skal køre for at undgå at falde af overfladen den afspritter.

Hvad der ikke er vist er en homestation, eller en dock, som robotten kan gå tilbage til når dens arbejde er afsluttet.

3. semester projekt

Analyse



Medlemmer

Murat Eroglu
Asger Busk Breinholm
Simon Shahrour
Yuhu Su Stougaard Nielsen
Erda Naz Cecillia Kocaman Bal
Magnus Gjøttrup Kristensen
Mathias Old Jensen

Studienummer

201906338
201807859
201810704
201904972
201900001
201908511
201610790

Vejleder

Mads Dyrmann

Dato

11/09/2020

Gruppe

Nr. 2

Indhold

Aktuatorer.....	3
DC-motor.....	3
Stepper motor.....	3
Sprit dispenser	3
IC'er	4
PSoC	4
Raspberry Pi W Zero	4
Vandmåler.....	5
LCD-display.....	6

Aktuatorer

DC-motor

IGARASHI-motor bliver der arbejdet i 3. semesterkursus GFV. Det er godt at vælge det, da vi har mere viden i den slags DC-Motor og den er relativt små som kun vejer 66g. Den har nominel spænding på 12V/DC, hvilket er ret standard for elektronik. Den kan man bruges til at køre på Spritinator.

Mere informationer kan findes via: https://blackboard.au.dk/bbcswebdav/pid-2697054-dt-content-rid-8821774_1/courses/BB-Cou-UUVA-91816/BB-Cou-UUVA-69270_ImportedContent_20170609013941/DC%20Motor%20IGARASHI%2024475.pdf

Stepper motor

28BYJ-48 bliver der også arbejdet i 3. semesterkursus GFV. Den har mindre nominel spænding end IGARASHI-motor som er 5VDC. Den kan man bruges til at styre køreretning på Spritinator.

Mere informationer kan findes via: https://blackboard.au.dk/bbcswebdav/pid-2697054-dt-content-rid-8821775_1/courses/BB-Cou-UUVA-91816/BB-Cou-UUVA-69270_ImportedContent_20170609013941/28BYJ-48-5V.pdf

Sprit dispenser

Her bruges DRV8825 (36JXS60K100G/42STH38-1684) Steppemotor som sprøjte dosering aktuator, der er lånt fra embedded stock, med et 12V controlboard, sprit dispenser bliver styret af Psoc'en.



IC'er

PSoC

~~Vi har fået oplyst en PSoC af skolen, CY8CKIT-059 PSoC 5LP (Cypress), som vi skulle købe. Vi bruger denne i andre fag, og er den PSoC vi har valgt at bruge til dette projekt, da vi har gjort bekendtskab med denne, og det er et krav at skulle bruge. Vi bruger denne PSoC i vores robot til at styre vores aktuatorer og vores UART. Den sidder i robotten og er koblet til RPi så vi kan styre den fra PC.~~

Raspberry Pi W Zero

Igennem de sidste semestre har vi gjort os bekendte med Raspberry Pi W Zero (RPi Zero).

RPi Zero har wireless LAN og bluetooth 4.1, og da Sprit-inator skal fungere som en ledningsfri enhed er dette et oplagt valg til at sende og modtage data, som videregives til PSoCen. Dvs. at vi fra computeren sender data fra en RPi Zero til den RPi Zero som sidder i Sprit-inator, og dette fungerer som vores start og stop mekanisme.

<https://www.raspberrypi.org/products/raspberry-pi-zero-w/>

Vandmåler

Når det kommer til vandmåleren, har vi forskellige muligheder. Hvis vi bare simpelt skal måle om der er væske i beholderen, eller om den er løbet tør, kan man købe en komponent til få penge på nettet, som fungerer meget simpelt og sender et analogt signal, alt efter den kan mærke væske eller ej. Dette sørger for at gøre det til en billig løsning, samt også lille og billig.

Her kan man se en lille illustration af sensoren:



Figur 1 - Vand sensor komponent¹

En anden mulighed vi også har er ved at gøre brug af en vægt, som vi kommer til at arbejde med i GFV faget, og putte beholderen på vægten. Så snart vægten af beholderen, falder under en sat grænse, vil vi få besked på dette. Ulempen ved dette er pladsbehovet og vægten.

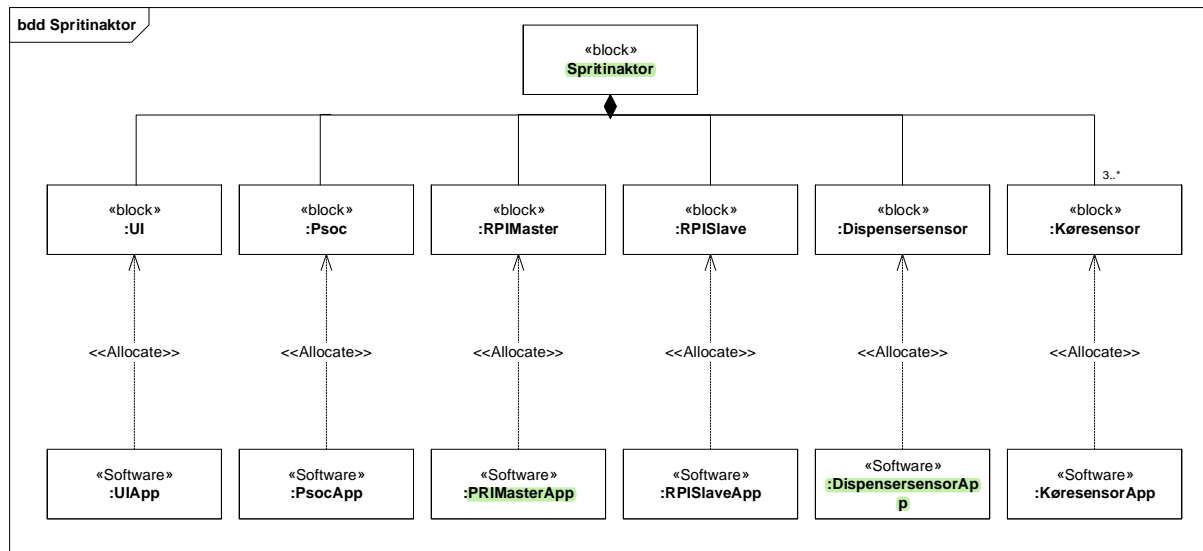
¹ <https://bit.ly/350RNmB>

LCD-display

For at kunne vide hvilke beskeder Sprit-inator sender til os vil vi bruge en LCD-skærm. Den kan bruges som et lille display som regel til at vise tekst. Her kan der udsendes besked til brugeren hvis der ikke er nok sprit eller strøm, hvis den er stødt på en forhindring osv. Skærmen skal selvfølgelig have strøm feks. fra MCU eller power supply. Det vil højst sandsynligt være 3.3V eller 5V. En MCU skal sende beskeder ud der skal **printes** på skærmen når dette er nødvendigt. Dermed kan den være del af vores UI. Et andet alternativ ville være at kommunikere med en PC og sende beskederne derover for at **printe** dem. Dette vil dog ikke være helt hensigtsmæssigt ved det endelige produkt.

Systemarkitektur

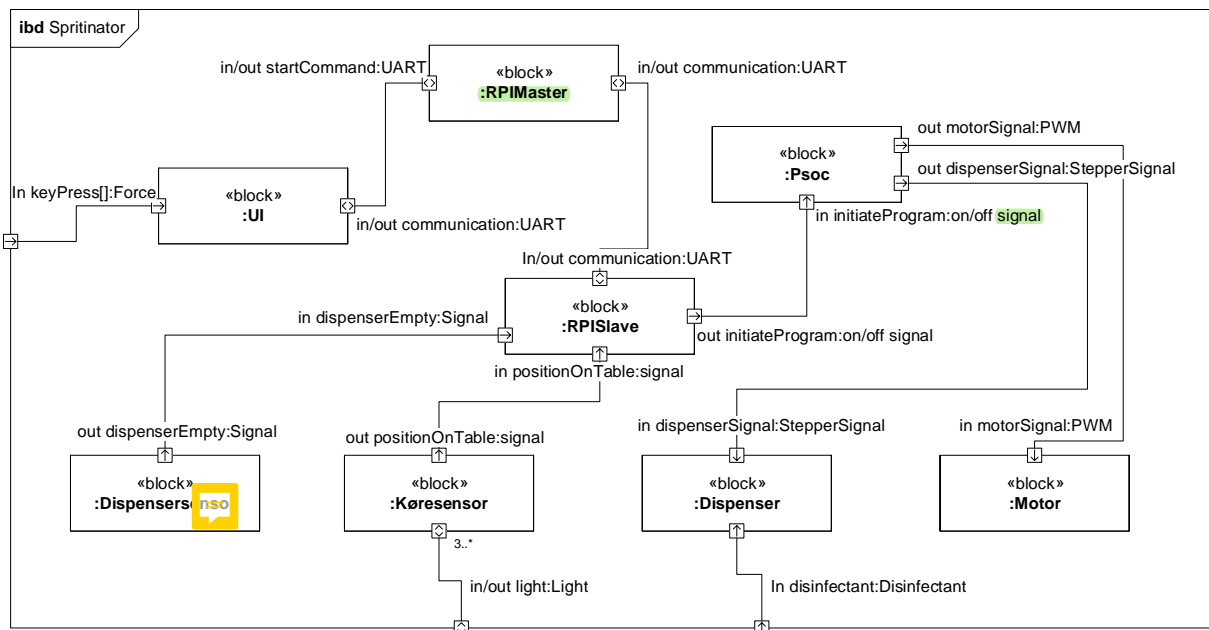
Softwareallokering (BDD)



Block beskrivelse(kort):

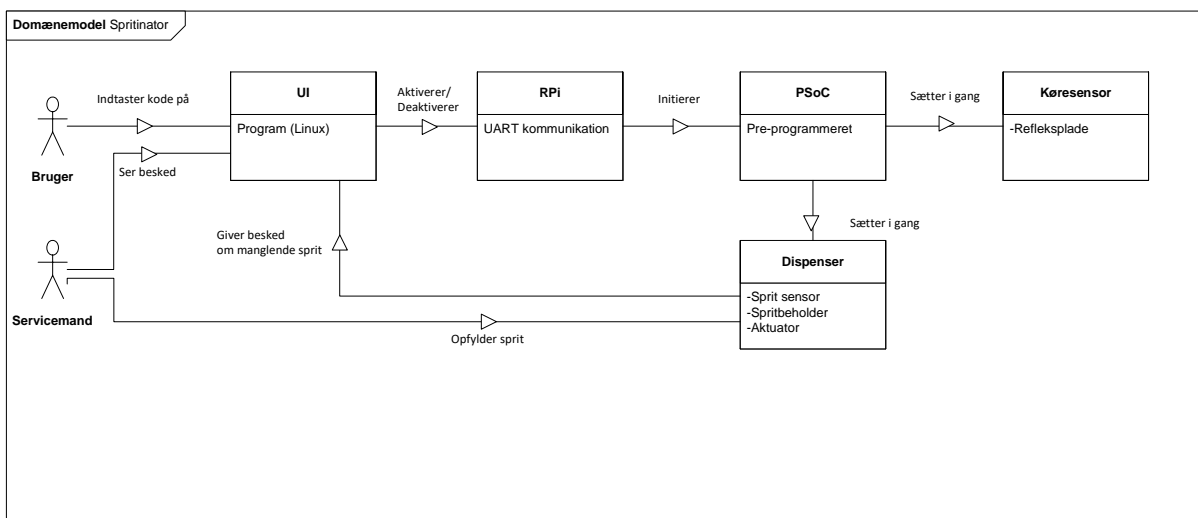
Block	Beskrivelse
UI	Giver brugeren mulighed for at kunne tænde og slukke for Spritinator og kan give forskellige beskeder til brugeren om vedrørende kørsel, så som advarsel hvis der ikke er mere sprit i beholderen eller hvis robotten er faldet af bordet.
PSoC	PSoC'ens funktion er at være robottens styreenhed. Den skal være pre-programmeret, og skal kunne sørge for at robotten kører sit program. PSoC modtager sin start funktion fra RPISlave.
PRIMaster	RPIMaster's funktion er at videregive start funktionen fra PC'en til RPISlave. PRIMaster skal også kunne modtage retur fra RPISlave, hvis der kommer en fejlmeddelelse.
RPISlave	RPISlave's funktion er at være modtager af den funktion der kommer fra RPIMaster og videregive den til PSoC'en, samt sende fejlmeddelelser retur til RPIMaster.
Dispensersensor	Dispensersensorens funktion er at returnere til RPISlave om der er nok sprit i vores dispenser beholder.
Køresensor	Kørersensorernes funktion er at holde robotten inden på bordet, så den ikke kører af bordet. Der skal være tre styk, én foran der skal registrere når robotten er nået til bordenden og derefter rotere 90°, og to sensorer i højre side der skal sørge for at robotten ikke kører ud over kanten, når den kører langs siden af bordet.

IBD



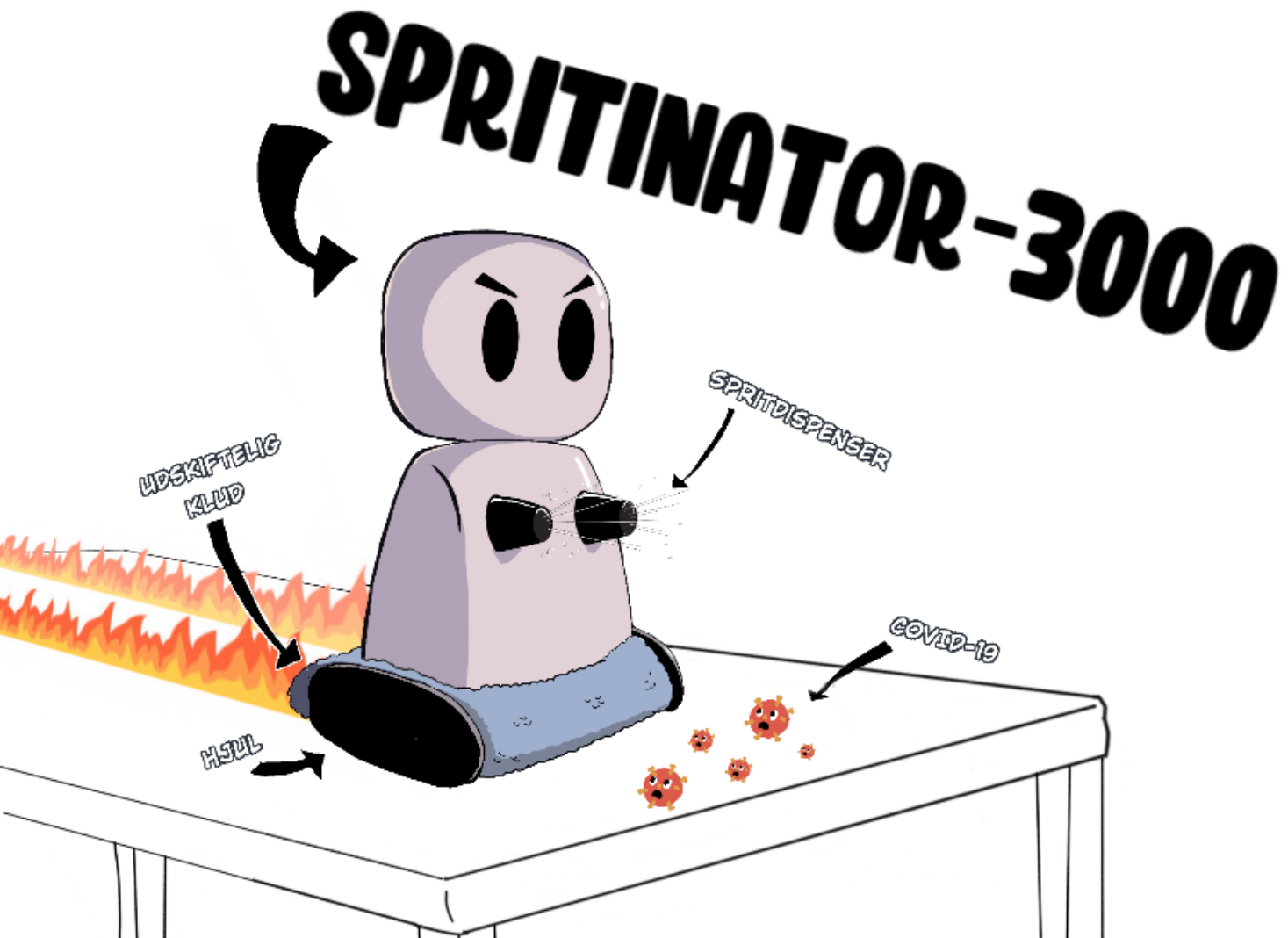
Domænemodel

(Den er vist ikke helt rigtig lige nu)



3. semester projekt

Kravspecifikation og acceptbeskrivelse



Medlemmer

Murat Eroglu
Asger Busk Breinholm
Simon Shahrour
Yuhu Su Stougaard Nielsen
Erda Naz Cecillia Kocaman Bal
Magnus Gjøttrup Kristensen
Mathias Old Jensen

Studienummer

201906338
201807859
201810704
201904972
201900001
201908511
201610790

Vejleder

Mads Dyrmann

Dato

02/10/2020

Gruppe

Nr. 2

Indhold

Kravspecifikation	3
Systembeskrivelse	3
MoSCoW	4
Aktørbeskrivelse	5
Aktør-kontekst-diagram	5
Use Case beskrivelse:	6
Use Case diagram	6
Fully Dressed use case:	6
Use Case 1: Afsprit bord:	6
Use Case 2: Opfyld dispenser:	7
Use case 3: Oplad	8
Ikke-funktionelle krav, FURPS.....	9
Accepttestbeskrivelse.....	11
Komponenter der bruges til at teste	11
Ordforklaring	11
Beskrivelse af funktionelle krav (Use Cases)	11
Use Case 1: Afsprit bord:	11
Use Case 1: EXT1:.....	11
Use Case 1: EXT2:.....	12
Use Case 1: EXT3:.....	12
Use Case 1: EXT4:.....	13
Use Case 1: EXT5:.....	13
Use Case 2:	13
Use Case 3:	14
Beskrivelse af de ikke-funktionelle krav	15
Functionality:	15
Usability:	15
Reliability:	16
Performance:	16
Supportability:	17

Kravspecifikation

Vi vil her give et systemoverblik over Sprit-inator. Vi går ind på vores systembeskrivelse, vi laver en MoSCoW, som er en prioritering af de egenskaber vores system skal besidde, og igennem Use Cases vil vi vise hvad der er af funktionelle- samt ikkefunktionelle krav.

Systembeskrivelse

"We are not ready for the next pandemic" -Bill gates

Menneskeheden glemmer hurtigt sin egen dødelighed, vi glemmer hurtigt at verdenens problemer ikke går væk selvom vi er klar til det næste store event. Vi har et stort ansvar over for hinanden der ikke går væk, men vi nedprioriterer det for andre ting så som at hente en hotdog fra føtex med vennerne der er på vej ud af døren, for at komme hjem, for at spare på tid. Og når det kommer til tid, så er der intet der tager mindre tid end at trykke på én knap. Denne knap er på **Sprit-inator**, en robot der tager sig af afspritningen! Selvom vennerne er på vej ud ad døren, kan DU hurtigt vise dit samfundssind med ét tryk! Et tryk kan gøre en forskel. Og den forskel kan være DIG.

Vores mål ved vores prototype er at lave en enkelt fungerende robot der kan afspritte et bord ordentligt og effektivt. Den skal kunne overholde kravene fra SSI om hvordan man afspritter korrekt.

Vores system består af:

- En PSoC som styrer motoren
- En sensor som sørger for at robotten ikke falder af bordet
- Et interface for at starte/stoppe robotten
- En motor som får hjulene til at køre
- En Power supply Unit for at holde robotten kørende
- En aktuator til spritdispenseringen som med et interval sprayer sprit

Brugeren tænder for Sprit-inator igennem interfacet, Sprit-inator fortæller hvor meget sprit der er i dispenseren. Og derefter sætter gang i aktuatoren og så motoren. Så kører robotten hen over bordet og når den har dispenseret en passende mængde sprit, så kører den tilbage til sin Dick. Alt hvad brugeren skal gøre, er at tænde for Sprit-inator.

MoSCoW

MUST:

- Embedded Linux interface (UI)
- Styring af DC-motor via PSoC, så robot kan køre forlæns, baglæns og stoppe
- Kontrolleret sprit-dispensering via en aktuator
- Detektere og stoppe når robotten når en bordkant via en sensor der reflekterer ned i **bordet**

SHOULD:

- Mulighed for at dreje på forreste hjul.
- Registrere forhindringer ved hjælp af en berøringssensor
- Registrere når spritbeholderen er ved at være tom
- Registrere hvis den kører ud over bordet via en sensor der reflekterer ned i bordet
- Trådløs forbindelse mellem UI og robot

COULD:

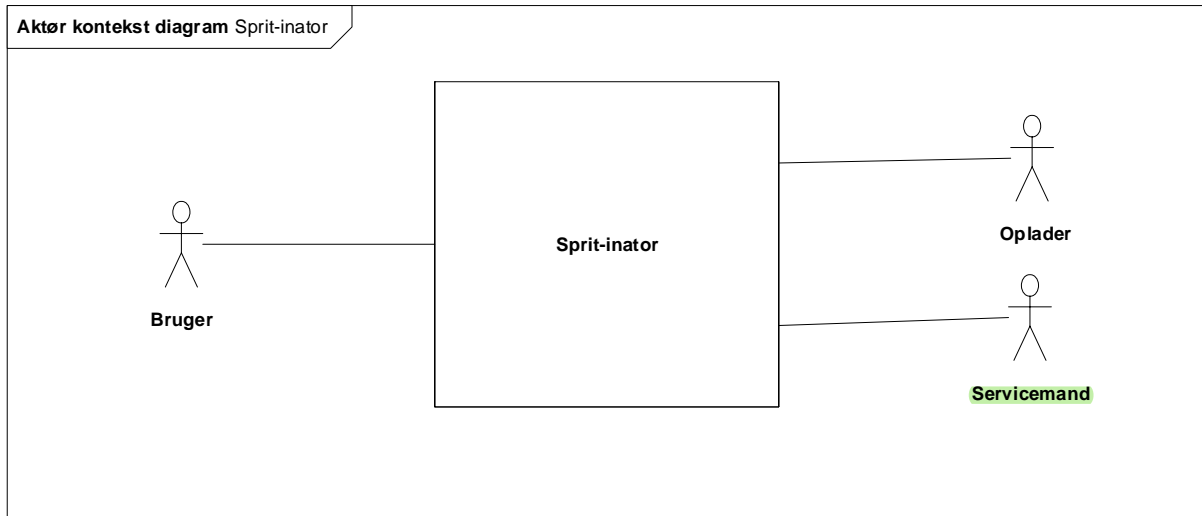
- Dock – en “home station” den selv kan finde og hvor den kan oplade og evt. blive påfyldt mere sprit
- APP/smartphone-tilgængeligt UI til styring af robotten.

WON'T:

- Mulighed for selv at skifte bord (hoppe/flyve fra bord til bord)
- Mulighed for selv at påfylde sprit gennem home stationen

Aktørbeskrivelse

Aktør-kontekst-diagram



Figur 1, Aktør diagram

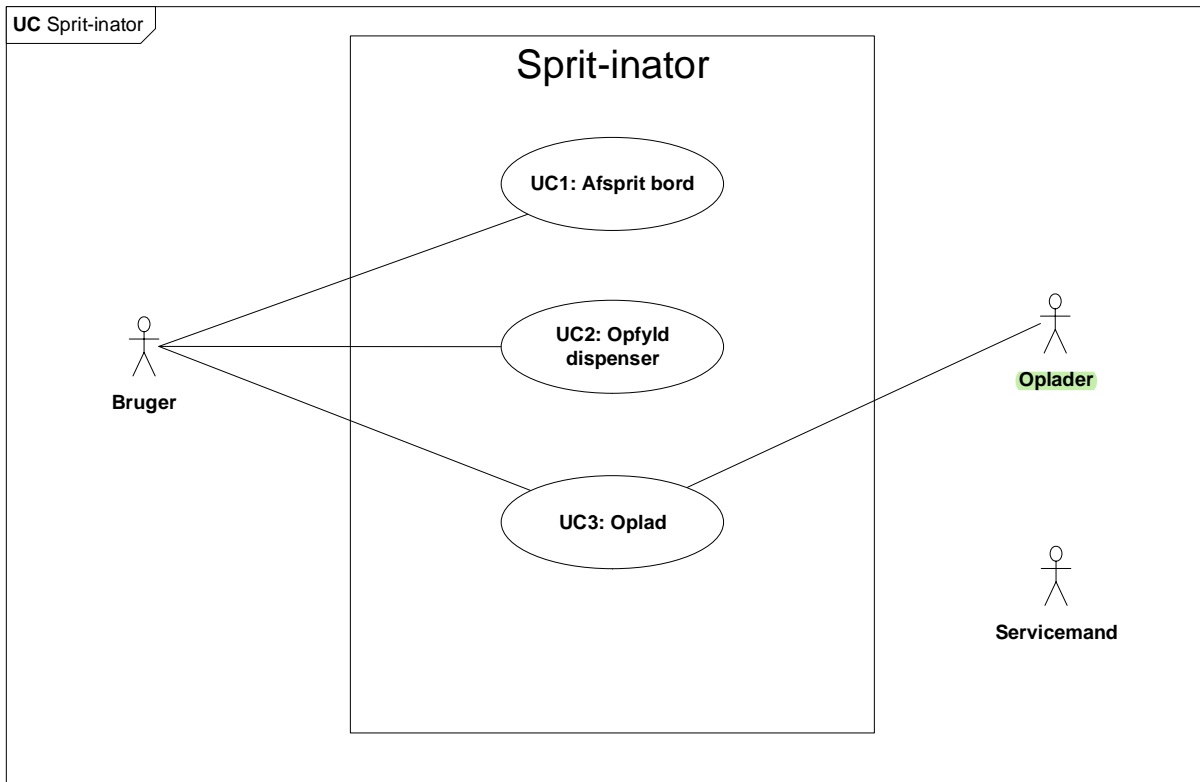
Aktørnavn:	Bruger
Type:	Primær
Beskrivelse:	Tænder for Sprit-inator så den kan spritte borde af og samler den op hvis den er faldet på gulvet. Brugeren kan også påfylde sprit og sætte den til opladning

Aktørnavn:	Servicemand
Type:	Primær
Beskrivelse:	Sørger for at reparere Sprit-inator hvis den er i stykker. Servicemanden kan også hjælpe med påfyldning af sprit og opladning hvis der er behov for dette.

Aktørnavn:	Oplader
Type:	Sekundær
Beskrivelse:	Kan lade Sprit-inator op, når den har kørt sit program.

Use Case beskrivelse:

Use Case diagram



Figur 2, Use case diagram

Fully Dressed use case:

Use Case 1: Afsprit bord:

Navn:	Afsprit bord automatisk
Mål	Bord bliver desinficeret for at mindske COVID-19.
Initiering	Bruger initierer Sprit-inator igennem UI.
Aktører	Primær: Bruger.
Antal samtidige forekomster	Ingen.
Prækondition	Sprit-inator står på et bord som er ryddet og frit for forhindringer.
Postkondition	Bordet er blevet sprittet af, og Sprit-iniator er ikke faldet af bordet.
Hovedscenarie	<ol style="list-style-type: none"> Sprit-inator starter op og returnerer mængden af sprit og strøm. [EXT. 1: Ikke opladt] [EXT. 2: Lav mængde sprit] Sprit-inator begynder at køre ligeud langs med bordkanten, mens den afspritter bordet. [EXT. 3: Der bliver stillet noget på bordet] [EXT. 4: Den falder ned fra bordet] [EXT. 5: Bruger stopper programmet på UI] Sprit-inator kører indtil den registrerer en bordkant og stopper. [EXT. 3: Der bliver stillet noget på bordet] [EXT. 4: Den falder ned fra bordet] [EXT. 5: Bruger stopper programmet på UI] Sprit-inator drejer 90° grader mod venstre. [EXT. 3: Der bliver stillet noget på bordet]

	<p>[EXT. 4: <i>Den falder ned fra bordet</i>] [EXT. 5: <i>Bruger stopper programmet på UI</i>]</p> <p>5. Loop fra punkt 2. Dette gøres 4 gange i alt for at nå rundt om hele bordet. [EXT. 3: <i>Der bliver stillet noget på bordet</i>] [EXT. 4: <i>Den falder ned fra bordet</i>] [EXT. 5: <i>Bruger stopper programmet på UI</i>]</p> <p>6. Sprit-inator har afsprittet hele bordet, stopper med at afspritte og bliver opladet ved lade stationen.</p>
<p>Udvidelser/undtagelser</p> <div style="border: 2px solid red; padding: 10px; margin-top: 10px;"> <p>Super gode extensions!</p> </div>	<p>EXT. 1: <i>Ikke opladt</i></p> <ol style="list-style-type: none"> Sprit-inator kan ikke starte, da batteriet ikke er opladt. UI returnerer "Ingen forbindelse til Sprit-inator. Kontroller venligst og prøv igen." Use case afsluttes. <p>EXT. 2: <i>Lav mængde sprit</i></p> <ol style="list-style-type: none"> Sprit-inator giver besked til brugeren om at påfylde sprit. Use case afsluttes. <p>EXT. 3: <i>Der bliver stillet noget på bordet</i></p> <ol style="list-style-type: none"> Sprit-inator opdager forhindringen og stopper. Sprit-inator giver en besked til brugeren at den har mødt en forhindring og behøver fysisk hjælp for at kunne fortsætte. Brugeren fjerner forhindringen. Use case fortsætter. <p>EXT. 4: <i>Den falder ned fra bordet</i></p> <ol style="list-style-type: none"> Sprit-inator registrerer ikke bordkant og kører ud over bordet. Sprit-inator registrerer via sensor på undersiden, at den ikke har fast underlag i kort tid, og returnerer til UI "Kørt af bord. Send help, plz!" Brugeren hjælper Sprit-inator og sætter den på plads. Use case fortsætter. <p>EXT. 5: <i>Bruger stopper programmet på UI.</i></p> <ol style="list-style-type: none"> Sprit-inator stopper med at køre. Use case afsluttes.

Use Case 2: Opfyld dispenser:

Navn:	Opfyld dispenser
Mål	Sprit-inator's dispenser bliver fyldt op med sprit.
Initiering	Sprit-inator giver besked til brugeren om at påfylde sprit.
Aktører	Primær: Bruger.
Antal samtidige forekomster	Ingen.
Prækondition	Sprit-inator mangler sprit i dispenseren.
Postkondition	Sprit-inator's dispenser er fyldt op.
Hovedscenarie	<ol style="list-style-type: none"> Sprit-inator giver besked til brugeren om at påfylde sprit. Bruger opdager at Sprit-inator er løbet tør for sprit. Bruger fylder dispenseren op med sprit. Sprit-inator er klar til at køre.
Udvidelser/undtagelser	Ingen.

Use case 3: Oplad

Navn:	Oplad
Mål	Sprit-inator bliver opladt.
Initiering	Sprit-inator står ovenpå sin dock.
Aktører	Sekundær: Oplader.
Antal samtidige forekomster	Ingen.
Prækondition	Sprit-inator mangler strøm.
Postkondition	Sprit-inator er opladt.
Hovedscenarie	<ol style="list-style-type: none"> 1. Sprit-inator står på sit startpunkt/dock. 2. En kontaktflade fra dock'en rammer en kontaktflade underneden på Sprit-inator. 3. Sprit-inator's batteri bliver opladt.
Udvidelser/undtagelser	Ingen.

Ikke-funktionelle krav, FURPS

Til FURPS har vi valgt at beskrive, Functionality, Usability, Reliability, Performance, og Supportability derefter inddele dem efter vigtighed ved at skrive det ind ved hjælp af MoSCoW som kan ses efter FURPS.

- **Functionality**
 - Skal køre ved hjælp af DC-motor.
 - Skal sprøjte sprit ved hjælp af en forstøver.
 - Skal detektere bordkant med sensor, så den ikke falder udover.
 - Brug af sensor til at finde bordkant så den kan dreje for at holde sig inde på bordet.
 - Brug af tryksensor så den kan mærke forhindringer og stoppe.
- **Usability**
 - Den skal være sjov og sej at se på.
 - Den skal kunne startes nemt ved et enkelt tryk af en kontakt.
 - Alle skal kunne finde ud af at tænde den uden besvær.
 - Skal ikke fylde mere end den halve bredde af bordet den kører på.
- **Reliability**
 - Skal kunne være fejlfrit ~99% af gangene.
 - Skal kunne afspritte bordet helt, så der ikke er usprittede områder.
 - Skal ikke falde ned fra bordet, nogensinde.
- **Performance**
 - Den skal kunne afspritte et bord, inden næste time starter (under 10min). Limited af, sprit skal sidde mindst 30sek. På overfladen for at være effektiv.
 - Den skal ikke bruge en overflødig mængde af sprit (bordet skal være dækket, men ikke oversvømmet, så det ikke kan nå at fordampe inden næste time).
- **Supportability**
 - Skal kunne køre mindst 1600 gange (2 år), før den går i stykker/skal til reparation (20 uger pr semester mandag-fredag 4 gange dagligt = 1600 gange for 2 år).
 - Nem at sætte op ved et nyt bord.
 - Nem at genopfylde sprit på, så de fleste kan finde ud af det.

Baseret på ovenstående, ikke-funktionelle krav, har vi opdelt dem yderligere ved hjælp af "MoSCoW". Disse er opdelt i MUST, minimumskravene for vores robot. SHOULD, noget vores robot burde kunne gøre. COULD, noget vores robot kunne komme til at være i stand til at kunne gøre. WON'T, hvad vores robot ikke vil blive i stand til at overholde.

MUST

- Skal køre ved hjælp af DC-motor.
- Skal detektere bordkant med sensor, så den ikke falder udover.
- Brug sensor til at finde bordkant så den kan dreje.
- Den skal kunne startes nemt ved et enkelt tryk af en kontakt.
- Skal kunne være fejlfrit ~99% af gangene.
- Skal kunne afspritte bordet helt, så der ikke er usprittede områder.
- Den skal kunne afspritte et bord, inden næste time starter (under 10min). Limmited af, sprit skal sidde mindst 30 sekunder på overfladen, for at være effektiv.

SHOULD

- Brug af tryksensor så den kan mærke forhindringer og stoppe.
- Nem at genopfylde sprit på, så de fleste kan finde ud af det.
- Skal ikke fylde mere end den halve bredde af bordet den kører på.
- Skal kunne kører mindst 1600 gange (2 år), før den går i stykker/skal til reparation (20 uger pr semester mandag-fredag 4 gange dagligt = 1600 gange for 2 år).
- Nem at sætte op ved et nyt bord.

COULD

- Skal sprøjte sprit ved hjælp af en forstøver.
- Den skal være sjov og sej at se på.
- Alle skal kunne finde ud af at tænde den uden besvær.
- Den skal ikke bruge en overflødig mængde af sprit (bordet skal være dækket, men ikke oversvømmet, så det ikke kan nå at fordampe inden næste time).

WON'T

- Skal ikke falde ned fra bordet, nogensinde.

Accepttestbeskrivelse

- Beskrivelse af funktionelle krav (Use Cases)
- Beskrivelse af ikke-funktionelle krav

Komponenter der bruges til at teste

- Stopur
- Analog Discovery
- Lineal
- En mængde afsprittingsmiddel til borde

Ordforklaring

UI	User Interface - En computer eller LCD-skærm

Beskrivelse af funktionelle krav (Use Cases)

Use Case 1: Afsprit bord:

Use Case under test:	Use Case 1: Afsprit bord
Scenarie:	Hovedscenarie
Prækondition:	Spritinator-3000 står på et bord som er ryddet og frit for forhindringer.

No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren tænder for Sprit-inator på UI.	Sprit-inator tænder og returnerer mængden af sprit og strøm på UI.		
2	Brugeren venter på at Sprit-inator udfører sit arbejde.	Sprit-inator har afsprittet bordet.		

Use Case 1: EXT1:

Use Case under test:	Use Case 1: Afsprit bord
Scenarie:	EXT. 1: Ikke opladt
Prækondition:	Sprit-inator er ikke ladet nok op til at spritte et bord af og har givet besked til bruger via UI

No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
-----	----------	----------------------------------	--------------------------------	---------------------

1	Brugeren sætter Sprit-inator til opladning	Sprit-inator begynder at lade op		
2	Brugeren venter på at Sprit-inator bliver ladet op	Sprit-inator er blevet ladet nok op		

Use Case 1: EXT2:

Use Case under test:	Use Case 1: Afsprit bord
Scenarie:	EXT.2: <i>Lav mængde sprit</i>
Prækondition:	Sprit-inator har ikke nok sprit til at kan udføre sit arbejde og har givet besked til bruger via UI

No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren bruger Sprit-inator indtil der ikke er sprit til en enkelt tur	Sprit-inator giver besked til brugeren om at påfylde sprit.		
2	Brugeren fylder spritbeholderen op med sprit.	Sprit-inator kan spritte af igen.		

Use Case 1: EXT3:

Use Case under test:	Use Case 1: Afsprit bord
Scenarie:	EXT.3: <i>Der bliver stillet noget på bordet</i>
Prækondition:	Sprit-inator mødes forhindring på bordet

No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren sætter Sprit-inator i gang	Sprit-inator begynder at spritte af		
2	Brugeren stiller en eller flere forhindringer på bordet	Sprit-inator giver besked til brugeren om at der er en eller flere forhindringer på bordet.		
3	Brugeren fjerner den/de forhindring(er) der er i vejen.	Sprit-inator fortsætter med at spritte af		

Use Case 1: EXT4:

Use Case under test:	Use Case 1: Afsprit bord			
Scenarie:	EXT.4: <i>Den falder ned fra bordet</i>			
Prækondition:	Sprit-inator er faldet ned fra bordet			
No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren sætter Sprit-inator i gang	Sprit-inator begynder at spritte af		
2	Brugeren skubber eller flytter Sprit-inator	Sprit-inator falder ned fra bordet og giver besked til brugeren om dette.		
3	Brugeren stiller Sprit-inator op på bordet igen.	Sprit-inator fortsætter med at spritte af.		

Use Case 1: EXT5:

Use Case under test:	Use Case 1: Afsprit bord			
Scenarie:	EXT.5: <i>Bruger stopper programmet på UI</i>			
Prækondition:	Sprit-inator er blevet stoppet			
No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren stopper sprit-inator på UI	Sprit-inator stopper med at køre og slukkes.		

Use Case 2:

Use Case under test:	Use Case 2: Opfyld dispenser			
Scenarie:	Hovedscenarie			
Prækondition:	Sprit-inator mangler sprit i dispenser			
No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren prøver at starte Sprit-inator	Sprit-inator giver besked til brugeren igennem UI om at der ikke er nok sprit	God use case - Minder om Opfyldning Ext. - blot bedre.	
2	Brugeren fylder dispenser med mere sprit.	Sprit-inator giver besked til ønsket mængde er påfyldt		
3	Bruger har fyldt nok sprit på.	Sprit-inator er klar til at køre igen.		

Use Case 3:

Use Case under test:	Use case 3: Oplad
Scenarie:	Sprit-inator skal genoplades
Prækondition:	Sprit-inator mangler strøm.

No.	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
1	Brugeren stiller Sprit-inator på startpunktet/docken.	Sprit-inator står på sit startpunkt/dock.		
2	Brugeren sætter strøm til Sprit-inator	Sprit-inator begynder at lade op.		
3	Brugeren venter på at Sprit-inator bliver ladet op.	Sprit-inator er blevet ladet helt op.		

Beskrivelse af de ikke-funktionelle krav

Functionality:

Ikke-funktionelle krav			Functionality		
No.	Krav	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
F1	Skal køre ved hjælp af DC-motor.	Der sættes strøm til DC-motor og signal til at køre.	Sprit-inator begynder at kører.		
F2	Skal sprøjte sprit ved hjælp af en forstøver.	Der sendes signal til sprøjtemekanismen.	Sprit-inator sprøjter sprøjt på bordet.		
F3	Skal detektere bordkant med sensor, så den ikke falder udover.	Sensoren registrer at den er ved at kører ud over kanten. Signal sendes til PSoC.	Sprit-inator stopper med at kører og falder ikke ned fra bordet.		
F4	Brug af sensor til at finde bordkant så den kan dreje for at holde sig inde på bordet.	Sensoren registrer at den har noget bordkanten. Der sendes signal til PSoC at den skal dreje.	Sprit-inator drejer og kører ikke ud over bordkanten. Den Fortsætter med at kører.		
F5	Brug af tryksensor så den kan mærke forhindringer og stoppe.	Brugeren stiller en forhindring hvor Sprit-inator skal køre	Sprit-inator mærker forhindringen og stopper inden den kører helt ind i den.		

Usability:

Ikke-funktionelle krav			Usability		
No.	Krav	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
U1	Den skal være sjov og sej at se på.	Den skal være cool beanz, mening fra 10 forskellige tilfældige personer.	Den ser OK cool ud 😊		
U2	Den skal kunne startes nemt ved et enkelt tryk af en kontakt.	Der bliver trykket start en enkelt gang.	Spritinator starter.		

U3	Alle skal kunne finde ud af at tænde den uden besvær.	10 tilfældig vaglte mennesker skal starte den uden hjælp.	Alle kan starte spritinator uden besvær.		
U4	Skal ikke fylde mere end den halve bredde af bordet den kører på.	Vi laver den ikke særlig stor.	Den bliver ikke særlig stor.		

Reliability:

Ikke-funktionelle krav			Reliability		
No.	Krav	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
R1	Skal kunne være fejlfrit ~99% af gangene.	Brugeren tester Sprit-inator 100 gange	Sprit-inator kører fejlfrit 99 ud af de 100 gange +-1 gang.		
R2	Skal kunne afspritte bordet helt, så der ikke er usprittede områder.	Brugeren tester Sprit-inator 100 gange?	Sprit-inator spritter hele border af hver gang.		
R3	Skal ikke falde ned fra bordet, nogensinde.	Sensor skal virke til fulde hver gang den kører de 100 gange fra fejlfri testen.	Spritinator falder aldrig ned.		

Performance:

Ikke-funktionelle krav			Performance		
No.	Krav	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
P1	Den skal kunne afspritte et bord, inden næste time starter (under 10min). Limiteret af, sprit skal sidde mindst 30sek. På overfladen for at være effektiv.	Stopur der tager tid for afviklingen.	Spritinator er færdig med at spritte et helt bord af på under 10 min.		

P2	Den skal ikke bruge en overflødig mængde af sprit (bordet skal være dækket, men ikke oversvømmet, så det ikke kan nå at fordampe inden næste time).	Måling af sprit i flasken før og efter en kørsel.	Der bliver ikke brugt mere sprit end nødvendigt.		
-----------	---	---	--	--	--

Supportability:

Ikke-funktionelle krav			Supportability		
No.	Krav	Handling	Forventet observation / resultat	Faktisk observation / resultat	Vurdering (OK/FAIL)
S1	Skal kunne kører mindst 1600 gange (2 år), før den går i stykker/skal til reparation (20 uger pr semester mandag-fredag 4 gange dagligt = 1600 gange for 2 år).	Producenten giver en garanti på funktionel kørsel mindst 1600 gange (2 år).	Kunden skal have Spritinator repareret efter 1600 gange eller betaling fra producenten hvis Spritinator går i stykker før 1600 gange.		
S2	Nem at sætte op ved et nyt bord.	Skal gennemfører forskellige test fejlfrit på et nyt bord.	Den virker som den skal på et nyt bord.		
S3	Nem at genopfylde sprit på, så de fleste kan finde ud af det.	10 tilfældige valgte mennesker opfylder sprit uden hjælp.	Den bliver påfyldt sprit uden problemer alle 10 gange.		