



# Workshoplogistik

Udvikling af system, der holder styr på udstyr til konsulentkørsel og workshops

Benjamin Jon Leonhardt  
Cecillie Skoven Møller  
Jens Tirsvad Nielsen  
Michael Kragh

TEAM 11, 1. semester, Datamatiker Online, UCL

## Indhold

Forord .....	3
Problemstilling.....	3
Teknologier .....	4
Projektstyring .....	4
Versions styring .....	4
Udviklingsmiljø .....	4
Udviklingsprocessen .....	4
Scrum roller .....	4
Backlogs .....	4
Retrospective meeting / sprint review meeting .....	4
Scrum board .....	5
Risikoanalyse .....	5
Udvælgelse af opgave i et sprint .....	5
Iterationen i et sprint .....	5
Scrum .....	5
Sprint 0 .....	5
Use Cases: .....	6
BMC (Business Model Canvas): .....	7
BPMN (Business Process Model Notation): .....	8
Sprint 1 .....	9
Udvælgelse af Use Cases: .....	9
Design .....	9
Overblik over kasser, udstyr og workshops: .....	9
Design Class Diagram .....	12
Systemsekvensdiagram .....	14
Wireframes .....	14
Programmering / Tests .....	14
Retrospektive meeting .....	15
Sprint 2 .....	15
Design .....	15
Wireframes .....	16

Programming / Tests .....	16
Retrospektive meeting.....	17
Sprint 3 .....	17
Design.....	17
Programming / Tests .....	18
Retrospektive meeting.....	18
Sprint 4 .....	18
Design.....	19
Programming / Tests .....	20
Retrospektive meeting.....	21
Sikkerhed .....	21
Konklusion .....	21
Litteratur .....	23
Bilag.....	24
Ordbog: .....	24
Referencemanual til kode: .....	24

## Forord

Dette projekt tager udgangspunkt i en reel virksomhed, som vores team skal udvikle et digitalt produkt til. Projektet startede som et konsolprogram, men undervejs skiftede vi til en WPF-applikation med grafisk brugerflade for bedre at kunne afspejle en moderne, digital løsning til virksomheden. Der er ikke tale om et færdigt, fuldt fungerende produkt, men en prototype, der demonstrerer centrale funktioner og potentialet for digitalisering af virksomhedens arbejdsprocesser.

## Problemstilling

Frederik Hugger, CEO i konsulentvirksomheden 451 Grader, afholder workshops og deltager i diverse arrangementer, hvor han skal medbringe forskelligt udstyr pakket i kasser. Hver kasse indeholder forskellige materialer, og det er vigtigt at holde styr på, hvilke kasser og hvilket udstyr der skal med til de enkelte workshops.

Derfor efterspørger kunden en løsning i form af et program eller en app, der kan:

- Administrere og registrere indholdet i hver kasse via en afkrydsningsliste.
- Sikre, at de korrekte kasser er pakket til arrangementerne via en separat afkrydsningsliste.
- Muliggøre søgning efter specifikt udstyr og angive, hvilken kasse det befinder sig i.
- Opbygge forløb (workshops) baseret på udstyrsbehov og automatisk beregne, hvilke kasser der skal med.

# Teknologier

## Projektstyring

I vores projektstyring bruger vi metoden Scrum for en agil softwareudvikling.

## Versions styring

Vi bruger Git til versionsstyring og GitHub som repository. Repository kan ses på [GitHub](#).

## Udviklingsmiljø

Projektet er udviklet i Visual Studio med C# som programmeringssprog. Vi har valgt at benytte WPF (Windows Presentation Foundation) sammen med MVVM-arkitekturmønsteret for at adskille brugergrænseflade, logik og data.

# Udviklingsprocessen

## Scrum roller

**Product Owner:** Cecilie

**Scrum Master:** Benjamin

**Development Team:** Jens og Michael

## Backlogs

Vi benytter en product backlog til at samle og prioritere alle funktionelle krav og opgaver i projektet. I starten blev 27 use cases identificeret og lagt i backloggen. Ud fra denne vælger vi relevante opgaver til hvert sprint og overfører dem til en sprint backlog.

Under sprint planning-møde prioriterer vi, hvilke opgaver vi vil arbejde med, baseret på værdi for kunden og systemet som helhed. Backloggen justeres løbende gennem projektet.

## Retrospective meeting / sprint review meeting

Vi afholder et retrospektivt møde hver mandag, umiddelbart før næste sprint går i gang. Her evaluerer vi forløbet i det netop afsluttede sprint – hvad der fungerede godt, hvilke udfordringer vi mødte, og hvad vi kan gøre bedre i næste sprint.

Til mandagsmødet hører en nedskrevet dagsorden, samt et efterfølgende referat, som begge vedlægges som dokumentation i vores repository.

Derudover gennemgår vi i sprint review-delen, hvad der er blevet udviklet og demonstrerer eventuelt for hinanden. På den måde sikrer vi løbende læring og tilpasning af vores arbejdsproces.

## Scrum board

Vi bruger et Scrum board via GitHub Projects til at visualisere vores opgaver. Opgaverne er inddelt i kolonnerne: To Do, In Progress, og Done, hvilket giver os overblik over status på opgaver og ansvar. Tavlen opdateres løbende i sprintet og danner grundlag for vores Scrum-møder.

## Risikoanalyse

I starten af projektet identificerede vi en række risici – bl.a. manglende erfaring med WPF/MVVM, begrænset tid til test, udfordringer med fremmøde og generel tidsstyring i teamet.

Vi håndterer disse risici løbende ved at fordele opgaver efter kompetencer, afholde regelmæssige møder og justere arbejdsmetoden, hvis noget ikke fungerer.

## Udvælgelse af opgave i et sprint

Til hvert sprint planning meeting prioriterer vi opgaver fra backloggen, som er relevante og realistiske at løse i sprintets tidsramme. Vi sørger for, at alle teammedlemmer får en opgave, som passer til deres rolle og kompetencer.

## Iterationen i et sprint

Vi arbejder iterativt i hvert sprint og holder scrum meetings tirsdag og torsdag, hvor vi kort fortæller, hvad vi har lavet, hvad vi skal i gang med, og om vi har nogle forhindringer.

Mellem disse møder koordinerer vi os via Teams og mødes fysisk eller online efter behov. På den måde sikrer vi løbende fremdrift og samarbejde.

# Scrum

## Sprint 0

I dette sprint udvalgte teamet de opgaver, vi kunne varetage på nuværende tidspunkt.

Disse blev tildelt på de enkelte teammedlemmer:

- Use Cases – **Benjamin**
- BMC – **Jens**
- BPMN (Business Process Model Notation) – Michael
- Skelet til kode – Jens
- Rapportskabelon – **Jens har oprettet rapportskabelonen og Cecilie har ansvaret for at opdatere og renskrive rapporten hele vejen igennem projektet.**

## Use Cases:

Benjamin havde udarbejdet 27 use cases, som vi skrev ind i vores backlog:

### Create – Read

#1 Brugeren opretter en kasse med nr. og beskrivelse. Kassen bliver gemt i systemet.

#2 Brugeren opretter et materiale med navn og antal komponenter/dele som gemmes i systemet.

#3 Brugeren placerer et materiale i en given kasse. Lokationen gemmes i systemet.

#4 Brugeren udskriver en kasses indhold til skærmen.

#5 Brugeren printer en liste over alle materialer og hvilken kasse de er placeret i.

#6 Brugeren opretter en aktivitet med et navn. Aktiviteten gemmes i systemet.

#7 Brugeren tilknytter et materiale til en given aktivitet. Tilknytningen gemmes i systemet.

#8 Brugeren udskriver en liste over hvilke kasser, der skal medbringes hvis en given aktivitet skal udføres, baseret på de tilknyttede materials placering.

#9 Brugeren opretter en workshop, som gemmes i systemet.

#10 Brugeren tilknytter en aktivitet til en given workshop. Tilknytningen gemmes i systemet.

#11 Brugeren printer en liste over kasser der skal benyttes til en workshop, til skærm.

#12 Brugeren printer en indholdsliste over kasser tilknyttet en workshop, til skærmen.

#13 Brugeren eksporterer en liste over kasser og deres indhold for en workshop til PDF eller papirprint.

#14 Systemet gemmer afholdte workshops i systemet.

#15 Brugeren tilknytter en evaluerende kommentar til en afholdt workshop, som gemmes i systemet.

#16 Brugeren gemmer ofte afholdte/standardiserede workshops i systemet.

#17 Brugeren sletter en given kasse fra systemet.

### Delete

#18 Brugeren sletter et materiale fra systemet.

#19 Brugeren sletter aktivitet fra systemet.

#20 Brugeren sletter workshop fra systemet.

#21 Brugeren fjerner materiale fra aktivitet.

#22 Brugeren fjerner aktiviteter fra workshop

## Update

#23 Brugeren ændrer en given kasses beskrivelse.

#24 Brugeren ændrer et materiales navn og/eller antal dele.

#25 Brugeren ændrer et materiales placering til anden kasse, eller “ikke i kasse”.

#26 Brugeren ændrer aktivitetsnavn.

#27 Brugeren ændrer workshopnavn.

## BMC (Business Model Canvas):

Business Model Canvas		Designed for: 451 Grader	Designed by: Team 11	Date: 27-04-2025	Version: 1
<b>Key Partners</b> Skoler Foreninger	<b>Key Activities</b> Undervisning Events	<b>Value Propositions</b> Læring	<b>Customer Relationships</b> Personlig betjening Rådgivning	<b>Customer Segments</b> Børn Lærer	
	<b>Key Resources</b> Undervisningsmateriale Ejeren		<b>Channels</b> Telefonisk E-mail		
<b>Cost Structure</b> Bil Tlf IT Hjemmeside Bærbar			<b>Revenue Streams</b> Events		

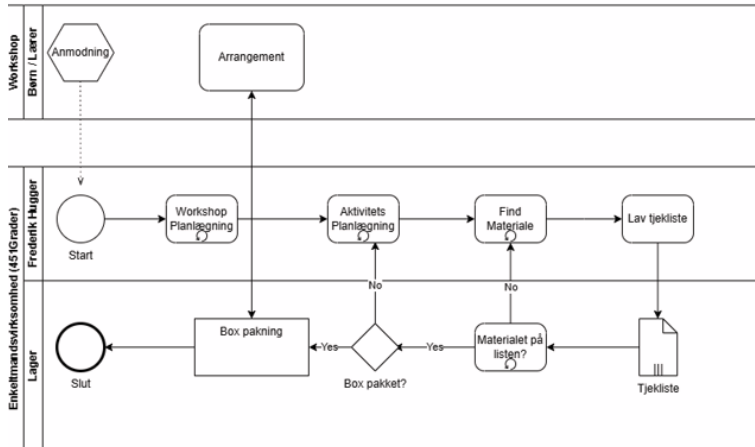


## BPMN (Business Process Model Notation):

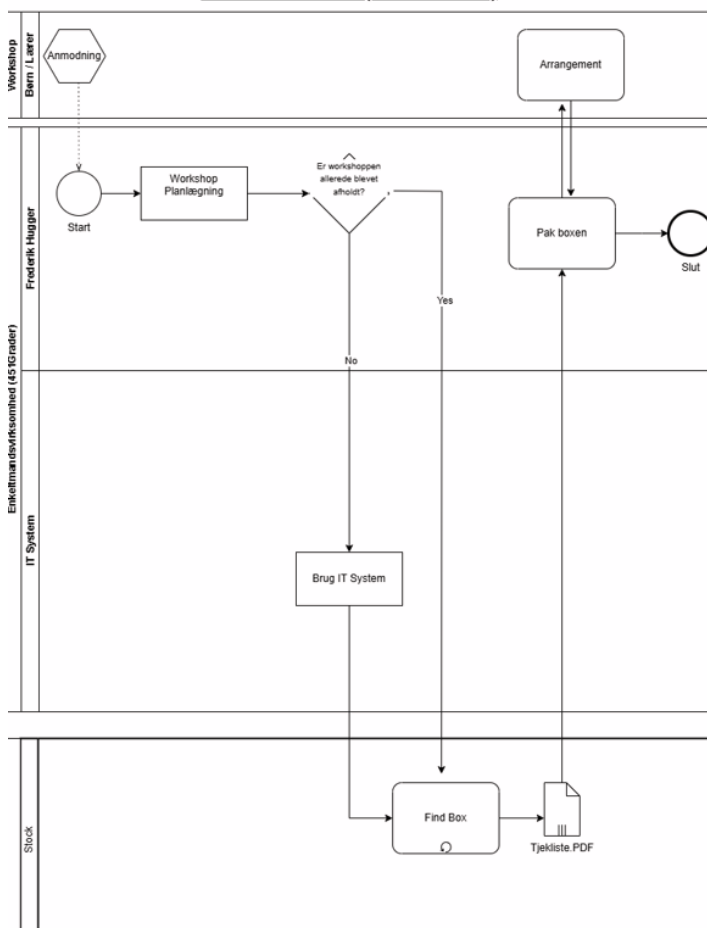
Projekt Getting Real BPMN  
Proces for planlægning og pakning af workshop i 451Grader

### Nuværende Model (as-is Model)

Symbol ref: Draw.IO



### Fremtids Model (to-be Model)



## Sprint 1

### Tildelte opgaver:

**Jens:** Design Class Diagram

**Cecilie:** Objektmodel, domænemodel og wireframes

**Benjamin:** Systemsekvensdiagrammer

**Michael:** Model Classes og Model Repositories

### Udvælgelse af Use Cases:

Vi udvalgte de 5 mest relevante, som vi ville kunne udvikle et brugbart produkt ud fra.

### De 5 use cases:

- **UC#1:** Brugeren opretter en kasse med nr. og beskrivelse. Kassen bliver gemt i systemet.
- **UC#2:** Brugeren opretter et materiale med navn og antal komponenter/dele som gemmes i systemet.
- **UC#3:** Brugeren placerer et materiale i en given kasse. Lokationen gemmes i systemet.
- **UC#6:** Brugeren opretter en aktivitet med et navn. Aktiviteten gemmes i systemet.
- **UC#7:** Brugeren tilknytter et materiale til en given aktivitet. Tilknytningen gemmes i systemet.

## Design

I Sprint 1 arbejdede vi med det overordnede systemdesign. Vi begyndte med at kortlægge relationerne mellem de centrale elementer: kasser, materialer, aktiviteter og workshops.

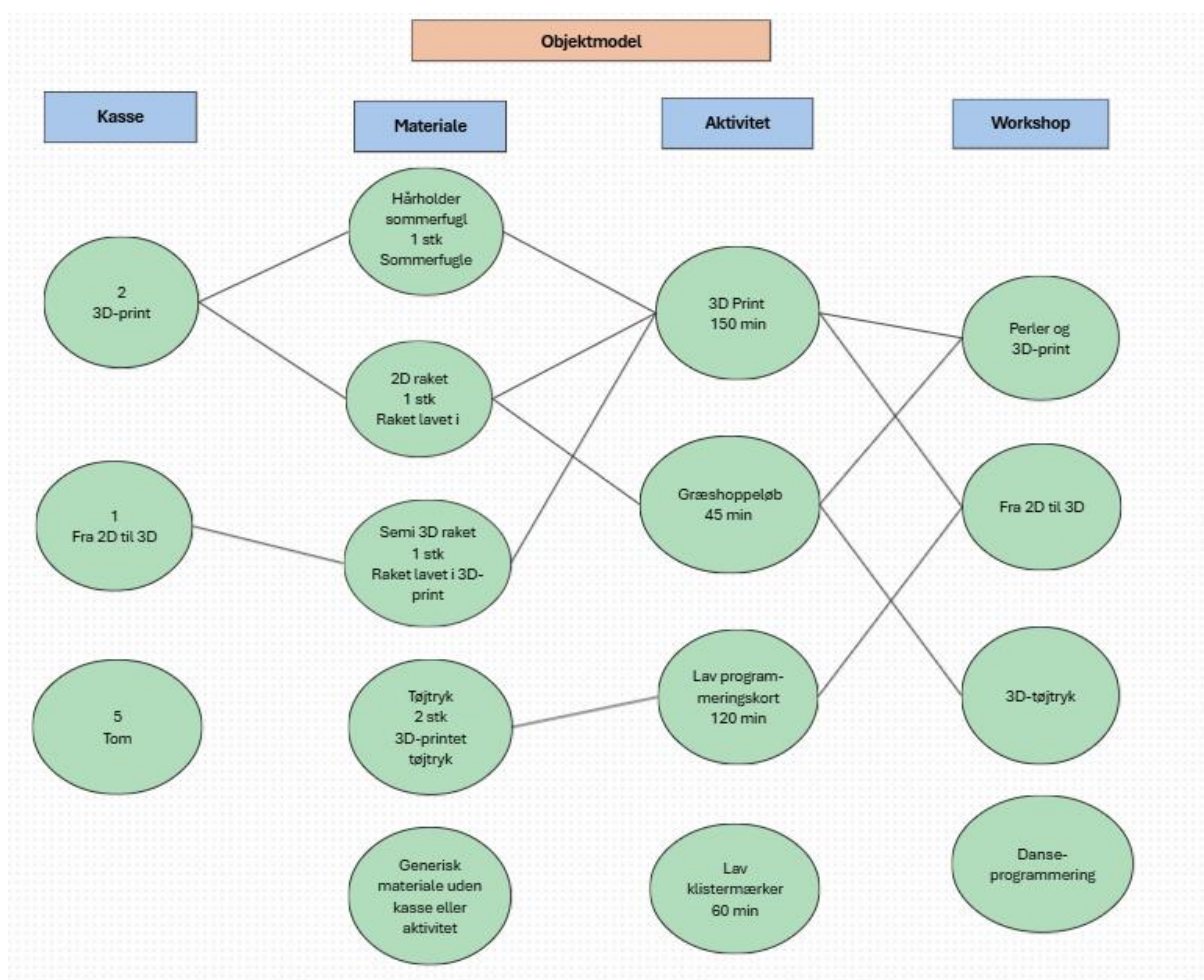
### Overblik over kasser, udstyr og workshops:

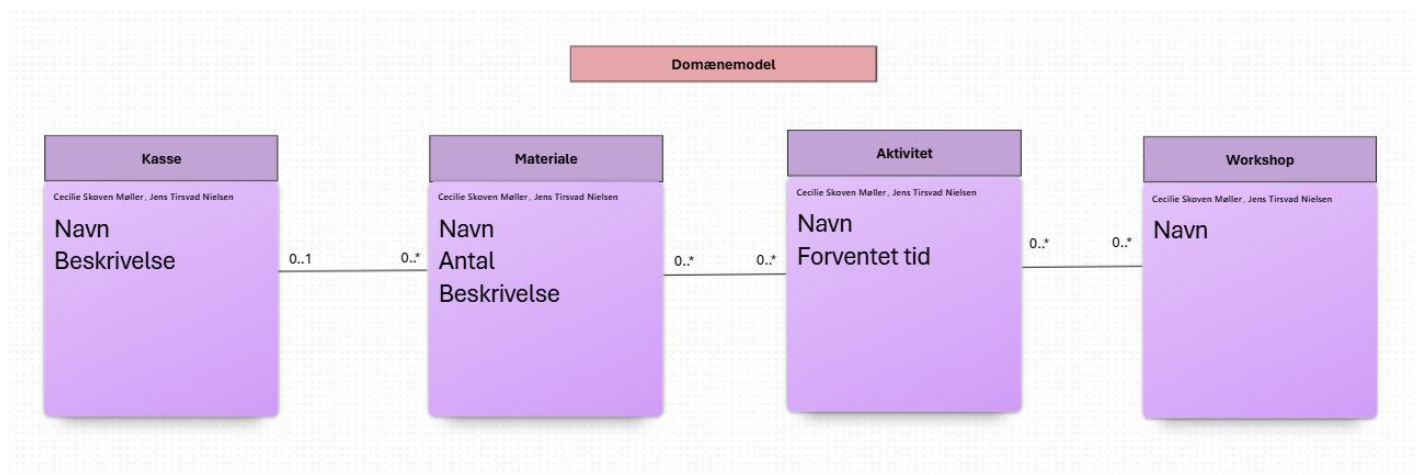
Vi udarbejdede en objekt- og domænemodel, som viser, hvordan de forskellige elementer hænger sammen. Modellen viser fx at:

- En kasse kan indeholde flere materialer.

- Et materiale kan placeres i én kasse ad gangen.
- En aktivitet kan have tilknyttet flere materialer.
- Workshops består af en eller flere aktiviteter.

Domæne- og objektmodellen blev udarbejdet af Cecilie og tilrettet af resten af teamet. Vi måtte tilbage og justere flere gange i løbet af hele sprint 1, for at sikre sporbarhed igennem alle vores artefakter.





## Design Class Diagram

DCD



### Begrundelse for designet af Design Class Diagram (DCD):

DCD'en er opbygget for at modellere de centrale entiteter og relationer i systemet med fokus på klarhed, genbrug og nem vedligeholdelse.

### Adskillelse af ansvar:

- **Model-namespace:** Indeholder domæne-entiteter (**Material**, **Box**, **Activity**, **Workshop**) og deres repositories. Dette adskiller forretningslogik og datarepræsentation fra datahåndtering og persistens.

- Handler-namespace: Indeholder hjælpeklasser som XmlFileHandler til persistens, så filoperationer holdes adskilt fra forretningslogik.

### **Repository-mønster:**

- Hver hovedentitet (Material, Box, Activity, Workshop) har en tilhørende repository-klasse, der håndterer samlinger af disse entiteter.
- Alle repositories arver fra en generisk abstrakt Repository<T>, hvilket fremmer genbrug og ensartethed i CRUD-operationer.

### **Interface og abstraktion:**

- Interfacet IUniqueIdGenerateAble sikrer, at alle repositories kan generere unikke id'er, hvilket understøtter dataintegritet.
- Den abstrakte Repository<T>-klasse sikrer et ensartet API for alle repositories.

### **Persistenslag:**

- Klassen XmlFileHandler bruges af alle repositories til at gemme og indlæse data, hvilket centraliserer fil-I/O og gøre det nemt at ændre persistensmekanisme senere.

### **Designprincipper:**

- Single Responsibility: Hver klasse har et klart, afgrænset ansvar.
- Open/Closed Principle: Nye entitetstyper eller repositories kan tilføjes med minimale ændringer i eksisterende kode.
- Løs kobling: Entiteter refererer til hinanden via GUID og ikke direkte objektreferencer, hvilket reducerer afhængigheder og forenkler serialisering.

### **Konklusion:**

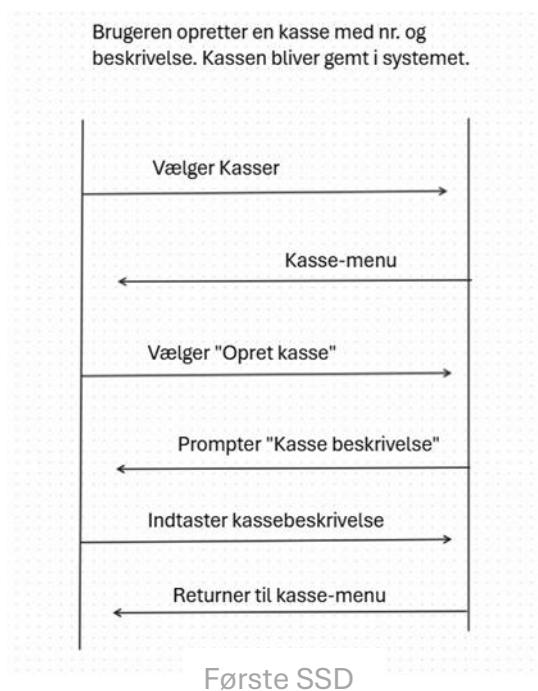
DCD'en er designet med henblik på udvidelsesmuligheder, vedligeholdelse og klar ansvarsfordeling, ved at bruge etablerede mønstre som repositories og interfaces til at håndtere kompleksitet. Den understøtter også fremtidig udvikling og man kan nemt opdele programmet som backend og frontend, hvor backend laves som library. ReportGenerator kan forbedres ved, at metoderne for at lave rapport returnerer et Xaml-objekt og print modtager Xaml-objekt som parameter.

## Systemsekvensdiagram

Til hver af de fem valgte use cases lavede Benjamin et systemsekvensdiagram (SSD). Diagrammerne viser interaktionen mellem bruger og system og danner grundlag for både brugergrænseflade og programmets flow.

Det viste sig dog, at når vi begyndte at lave wireframes, måtte vi tilføje ændringer til sekvensdiagrammerne.

Derudover beslutter vi os for at skifte vores arkitektur til at være WPF og MVVM, hvilket betød, at vi måtte ændre sekvensdiagrammerne igen, hvilket vi gik videre med i sprint 2.



## Wireframes

Ud fra sekvensdiagrammerne lavede vi wireframes, som visualiserer brugeroplevelsen.

De første wireframes blev lavet af Cecilie, men dog ikke i overensstemmelse med WPF og MVVM-strukturen og måtte derfor revideres.

## Programmering / Tests

Vi besluttede os for at skifte vores arkitektur til at være WPF og MVVM.

Vi tilføjede Model Classes og Model Repositories.

Tests nåede vi ikke i første Sprint.

## Retrospektive meeting

I slutningen af Sprint 1 afholdte vi et retrospektivt møde, hvor vi evaluerede Sprintets forløb. Vi havde en god dialog og snakkede om tingene.

Skiftet til WPF og MVVM gav ny læring, som vi tog med videre i næste sprint.

Vores sekvensdiagrammer og wireframes skulle revideres i andet sprint, så det passede med den nye struktur.

Det blev konstateret, at Scrum-metoden ikke blev fulgt korrekt i projektets indledende faser. I stedet fulgte processen en "vandfalds"-struktur, hvilket ikke er i overensstemmelse med Scrum-principperne.

Ifølge en korrekt anvendelse af metoden burde projektet være startet med udvælgelsen af én use case, baseret på én enkelt konceptuel klasse. Denne klasse skulle danne grundlag for implementeringen af alle fire CRUD-operationer (Create, Read, Update, Delete) som et basis API. Dette ville sikre en stabil og genanvendelig arkitektur. Først i de efterfølgende sprints burde der være tilføjet flere use cases og konceptuelle klasser, hvor samme CRUD-struktur kunne genbruges og videreudvikles.

Vi blev ikke helt færdige med tests.

Tilføjelser på GitHub skulle, indtil nu, altid godkendes et andet teammedlem. Det fungerede dog ikke i forhold til den måde vi arbejdede på, da det simpelthen tager for lang tid, så det slog vi fra igen.

## Sprint 2

Vi startede sprint 2 med at holde et sprintplanning meeting.

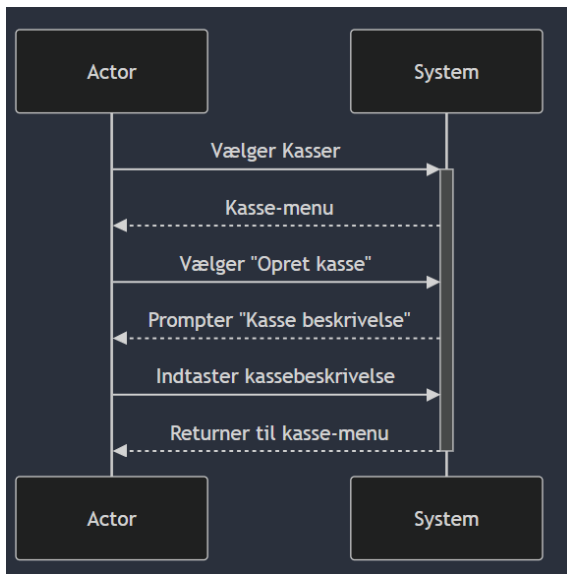
Opgaver i dette sprint:

- Nye SSD til GUI-applikation – **Jens**
- Wireframes – **Cecilie**
- ViewModel – **Jens**
- Repository - **Benjamin**
- MS-tests – **Michael**

## Design

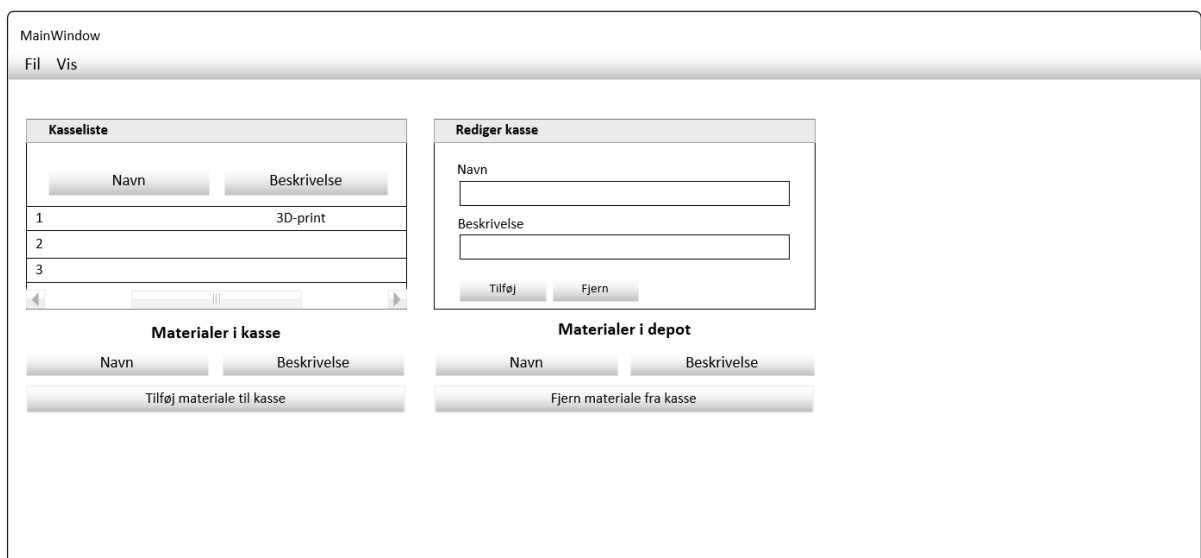
Jens lavede nye SSD til GUI-applikation.





## Wireframes

Cecilie lavede nye wireframes efter vores nye WPF-MVVM-struktur.



## Programmering / Tests

Benjamin og Jens forbedrede vores håndtering af filer og implementerede et repository-pattern til håndtering af XML-data.

Jens lavede ViewModel og fik lagt alle materialerne ind.

Michael lavede MS-tests.

## Retrospektive meeting

I sprint 2 ændrede vi projektets struktur til at benytte WPF med MVVM-arkitektur. Overgangen fra et konsolprogram til en grafisk brugerflade var en markant ændring, da vi nu skulle arbejde med elementer som drop-down-menuer og knapper.

Vi erfarede, at vores wireframes blev udarbejdet for sent i processen og derfor ikke nåede at fungere som et forberedende grundlag for WPF-opsætningen, hvilket påvirkede arbejdsgangen negativt.

Desuden blev det tydeligt, at vi kunne have været mere strukturerede i vores brug af Scrum. Vi manglede blandt andet hyppigere Daily Scrums, hvilket gjorde det svært at bevare overblikket over sprintets fremdrift.

## Sprint 3

Vi startede sprint 3 med at holde et sprint planning meeting. Vi nåede ikke at lave en dagsorden til mødet.

Opgaver i dette sprint:

- ObservableCollection – **Jens**
- MS-test – **Michael**
- Videreudvikling af workshopfunktionen – **Benjamin**
- Rapportgenerator – **Cecilie**

## Design

Der blev videreudviklet på menuerne i programmet for at forbedre brugeroplevelsen og navigationen.

Navn	Beskrivelse
1	2D til 3D
2	3D

**Kasse**

Navn:

Beskrivelse:

## Programmering / Tests

Jens oprettede en ObservableCollection, som dynamisk tilpasser indholdet, så kun det materiale, der er tilgængeligt, vises.

Michael fortsatte med at lave MS test, for at sikre korrekt funktionalitet i programmet.

Benjamin videreudviklede workshop-funktionen, så det nu er muligt at tilknytte aktiviteter til en workshop, gemme denne tilknytning i systemet, samt fjerne dem igen efter behov.

Cecilie implementerede en rapportgenerator, der gør det muligt at generere rapporter baseret på data fra workshops.

I sprint 2 tilføjede Jens en funktion til at håndtere materialer i workshops. Gruppen diskuterede, om dette var en hensigtsmæssig tilføjelse, og vi besluttede at fjerne koden igen. Funktionen blev vurderet som unødvendig i forhold til projektets formål, og fjernelsen gjorde det lettere at fokusere og færdiggøre de vigtigste dele af løsningen.

Vi planlagde at tilføje en boolsk egenskab til materialerne, der angiver, om der kræves strømforsyning. Dette skyldes, at Frederik nogle gange afholder workshops på steder med begrænset adgang til strøm, hvor batteridrevet udstyr derfor kan være nødvendigt. Implementeringen af denne funktion ville hjælpe med bedre planlægning og logistik. Dette betød, at vi tilføjede en task: "Tilføj strøm og batteriafhængighed til Materialemodel". Dette prioriteres sidst, når alt andet virker.

## Retrospektive meeting

I sprint 3 fik vi løst de fleste opgaver, som planlagt. Vores teamstruktur kunne stadig godt være bedre. Der blev ikke lavet dagsorden til mødet, ej heller referat.

En udfordring ift. programmeringen var behovet for bedre prioritering af funktionalitet. Et konkret eksempel var tilføjelsen af materialer til workshops, som vi efterfølgende valgte at fjerne, da det ikke bidrog væsentligt til projektets mål. Dette viste os vigtigheden af løbende at vurdere relevansen af nye funktioner i forhold til det overordnede scope.

Jens opdagede derudover, at vi ikke havde lavet sekvensdiagrammer og operationskontrakter forud for vores programmering. Vi aftalte at tale om det på vores sprint planning meeting.

Cecilie ville kontakte Product Owner Frederik Hugger og aftale et møde i sprint 4, så vi kunne vise ham, hvordan programmet var kommet til at se ud.

## Sprint 4

Sprint 4 blev indledt med et sprint planning-møde, hvor vi fordelte opgaverne for sprintet:

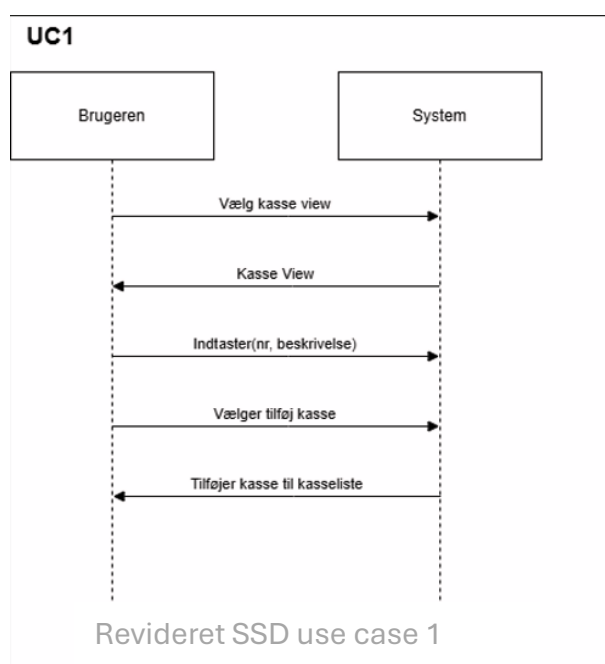
- Operationskontrakter, samt revidering af SSD – **Benjamin**
- Tilføjelse af funktionalitet til at udskrive rapporter som PDF og fysisk print samt generering af rapporter over kasseindhold – **Cecilie**
- Udvikling af view til visning af rapporter over kasseindhold – **Michael**
- Sekvensdiagrammer, samt visning af, hvilke kasser der skal med til workshop – **Jens**

## Design

I slutningen af sprint 3 blev det opdaget, at vi ikke havde udarbejdet sekvensdiagrammer inden programmeringen gik i gang. Derfor blev vi enige om at udarbejde disse i sprint 4 og samtidig dokumentere vigtigheden af at lave sekvensdiagrammer før implementering.

Benjamin opdaterede vores SSD, så det stemte overens med SD'et, samt udarbejdede operationskontrakter til syv use cases.

### SSD FOR USE CASE 1:



### OPERATIONSKONTRAKT FOR USE CASE 1:

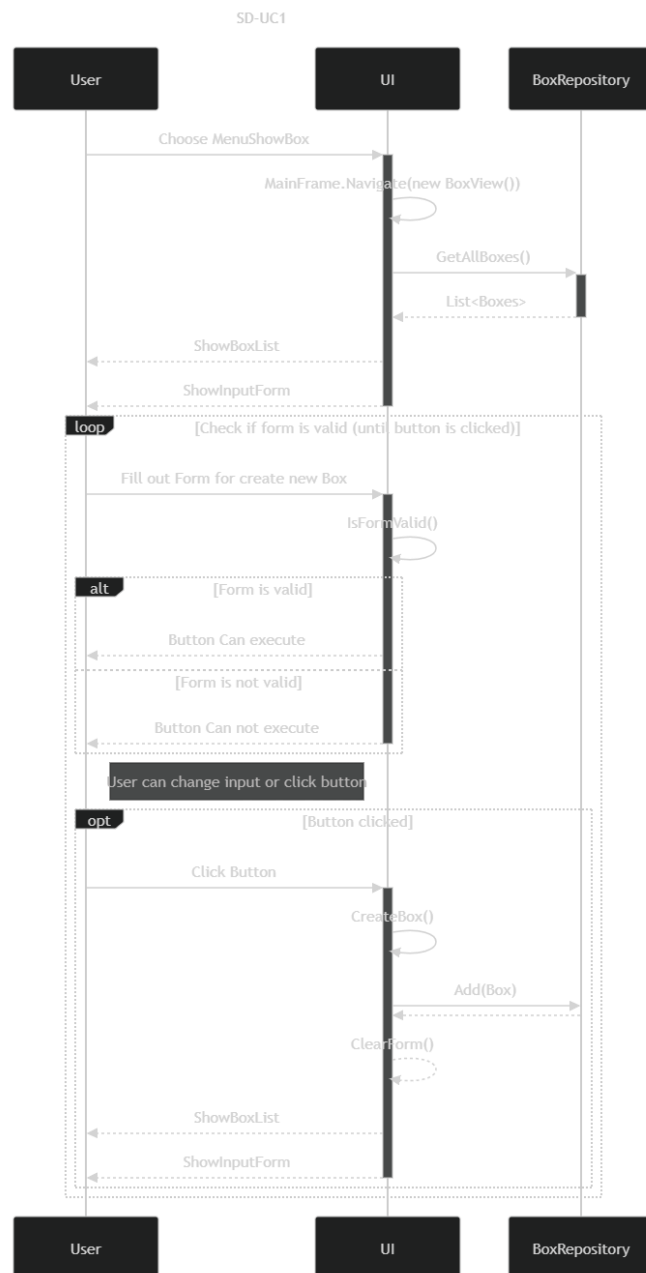
**Operation:** Vælg Kasse view

**Cross References:** UC1

**Preconditions:** n/a

**Postconditions:** GUI displayer kasse-view

## SD TIL USE CASE 1:



## Programmering / Tests

Vi færdiggjorde programmet i dette sprint, og det endelige resultat blev en funktionel løsning, der opfylder kravene.

Vi tilføjede muligheden for at få et overblik over workshops, se indholdet i de enkelte kasser samt udskrive rapporter over disse. I rapportgeneratoren blev der desuden

tilføjet en afkrydsningsliste, som giver brugeren mulighed for at vælge, hvilke kasser der skal inkluderes i rapporten.

## Retrospektive meeting

Vi måtte erkende, at det ville have været en fordel at udarbejde sekvensdiagrammer og operationskontrakter løbende gennem projektet. Det ville ikke blot have gjort det nemmere at komme i gang med implementeringen, men det kunne også have bidraget til at forbedre og justere vores use cases tidligere i forløbet.

Vi når derfor også kun at lave SD til use case 1.

Ved færdiggørelse af vores rapport, opdagede vi en fejl i BPMN. IT-systemet er ikke en afdeling i sig selv, så den skal ikke være en del af modellen.

## Sikkerhed

Da systemet kun anvendes af én bruger – virksomhedens ejer – har vi valgt en simpel tilgang til sikkerhed. Der er ikke implementeret brugerstyring eller adgangskontrol, da behovet for adskilte brugerroller eller beskyttelse mod uautoriseret adgang er begrænset.

Sikkerheden er derfor primært fokuseret på dataintegritet og sikring af, at data ikke går tabt. Systemet gemmer data i XML-filer, som let kan eksporteres og sikkerhedskopieres. For at undgå fejl og datakorruption ved læsning og skrivning til filer er der indbygget kontrol i filhåndteringen.

Da applikationen kører lokalt og ikke er forbundet til internettet, er risikoen for eksterne angreb minimal. Skulle systemet i fremtiden udvides til flere brugere eller netværksadgang, bør der tilføjes adgangskontrol, kryptering og eventuelt brugerautentificering.

## Konklusion

Projektet har givet os en praktisk og realistisk indsigt i, hvordan man udvikler en softwareløsning fra idé til fungerende prototype. Gennem brug af Scrum og iterative sprints har vi oplevet vigtigheden af løbende planlægning, samarbejde og tilpasning – både i forhold til tekniske valg og teamorganisering.

Vi startede med et simpelt konsolprogram, men besluttede hurtigt at skifte til en WPF-applikation for at opnå en mere brugervenlig og moderne løsning. Denne ændring krævede, at vi satte os ind i MVVM-arkitekturen og arbejdede med nye begreber som ViewModels, data binding og UI-design. Det har styrket vores forståelse af adskillelse af logik og præsentation samt gjort programmet lettere at vedligeholde og udvide.

Systemet understøtter nu de centrale funktioner, som kunden efterspurgte – herunder oprettelse og styring af kasser, materialer, aktiviteter og workshops samt generering og udskrivning af rapporter. Gennem arbejdet med domænemodeller, sekvensdiagrammer, tests og rapportgenerator har vi fået erfaring med flere aspekter af softwareudvikling, herunder analyse, design, implementering og dokumentation.

Undervejs har vi også identificeret områder med plads til forbedring – eksempelvis i vores brug af Scrum, dokumentation af design før implementering, samt prioritering af funktionalitet. Disse refleksioner har givet os en værdifuld forståelse for, hvordan man balancerer teori og praksis i et virkelighedsnært projekt.

Vi opdagede også nogle andre fejl undervejs i form af sekvensdiagrammer, der blev lavet efter programmeringen og BPMN, hvor IT-systemet stod som en afdeling for sig selv.

Alt i alt er vi tilfredse med det resultat, vi har opnået, og vi vurderer, at prototypen viser et klart potentiale for videreudvikling og anvendelse i en rigtig virksomhedssammenhæng.

## Litteratur

- *Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and Iterative Development*  
**Af Craig Larman**
- [ucl.itslearning.com](http://ucl.itslearning.com)
- *Undervisere på 1. semester, Datamatiker Online, UCL*  
*Anja Birklund, Lene Vestergaard Andersen, Diaa Zobair Shollar,*  
*Thomas Tjellesen, Tue Hellstern*



## Bilag

Ordbog:

### Dictionary

Da	Eng	Description
Kasse	Box	The boxes used for storage and transportation of materials
Materiale	Material	The variety of things used for conducting workshops.
Aktivitet	Activity	Individual thematic segments of a workshop
Workshop	Workshop	Collection of activities

Referencemanual til kode:

[refman.docx](#)