A logo with white text and blue dots

AI-generated content may be incorrect.A logo with white text and blue dots

AI-generated content may be incorrect.

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, Datamatiker Online UCL

Indhold

[Forord 3](#_Toc196464520)

[Problemstilling 3](#_Toc196464521)

[Teknologier 4](#_Toc196464522)

[Projektstyring 4](#_Toc196464523)

[Versions styring 4](#_Toc196464524)

[Udviklingsmiljø 5](#_Toc196464525)

[Udviklingsprocessen 5](#_Toc196464526)

[Scrum roller 5](#_Toc196464527)

[Backlogs 5](#_Toc196464528)

[Retrospective meeting / sprint review meeting 5](#_Toc196464529)

[Scrum board 5](#_Toc196464530)

[Risikoanalyse 5](#_Toc196464532)

[Udvælgelse af opgave I et sprint 5](#_Toc196464533)

[Iterationen i et sprint 5](#_Toc196464534)

[Scrum 5](#_Toc196464535)

[Sprint 0 5](#_Toc196464536)

[Sprint 1 5](#_Toc196464537)

[Udvælgelse af use cases 5](#_Toc196464538)

[Design 5](#_Toc196464539)

[Wireframes 5](#_Toc196464540)

[Programmering / Tests 5](#_Toc196464541)

[Retrospektive meeting 5](#_Toc196464542)

[Sprint 2 5](#_Toc196464543)

[Design 5](#_Toc196464545)

[Wireframes 5](#_Toc196464546)

[Programmering / Tests 6](#_Toc196464547)

[Retrospektive meeting 6](#_Toc196464548)

[Sprint 3 6](#_Toc196464549)

[Design 6](#_Toc196464551)

[Wireframes 6](#_Toc196464552)

[Programmering / Tests 6](#_Toc196464553)

[Retrospektive meeting 6](#_Toc196464554)

[Sprint 4 6](#_Toc196464555)

[Design 6](#_Toc196464557)

[Wireframes 6](#_Toc196464558)

[Programmering / Tests 6](#_Toc196464559)

[Retrospektive meeting 6](#_Toc196464560)

[Sikkerhed 6](#_Toc196464561)

[Konklusion 6](#_Toc196464562)

[Litteratur 7](#_Toc196464563)

[Bilag 8](#_Toc196464564)

# Forord

Dette projekt tager udgangspunkt i en rigtig virksomhed, som vores team skal udvikle et produkt til.

Der er ikke tale om et færdigt, fuldt fungerende program, men et konsolprogram, der skal afspejle et program, som ville kunne digitalisere virksomheden.

# 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](https://github.com/DMOoF25-Team-11-2/GettingReal).

## 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 retrospective meeting 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.

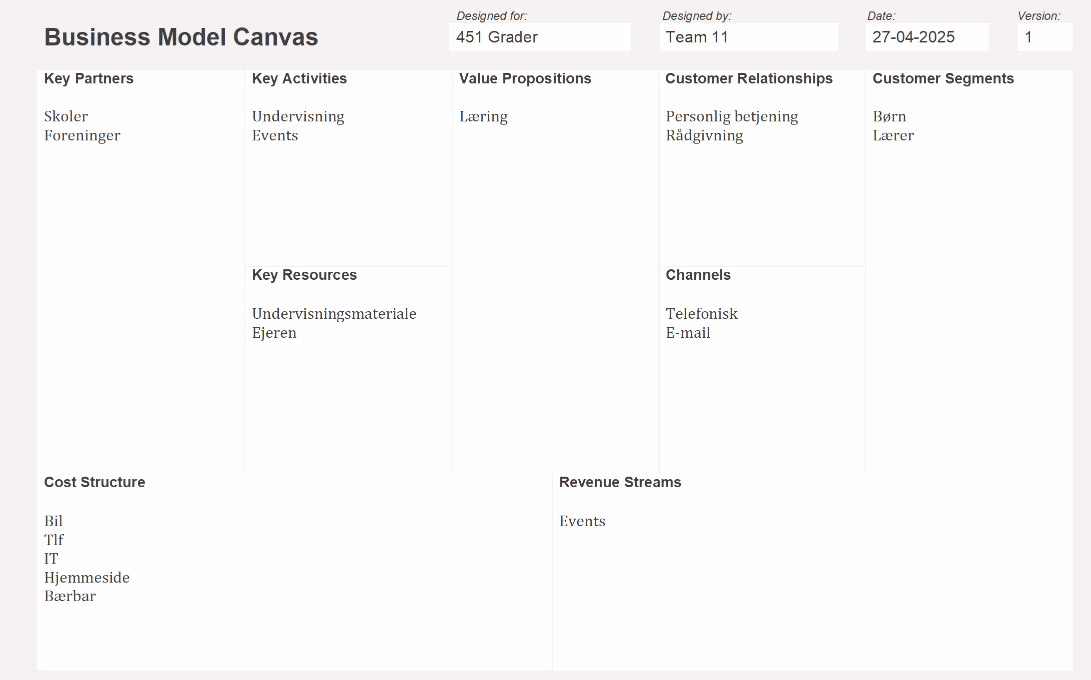
### **Use Cases:**

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

Use Cases

* Brugeren opretter en kasse med nr. og beskrivelse. Kassen bliver gemt i systemet.
* Brugeren opretter et materiale med navn og antal komponenter/dele som gemmes i systemet.
* Brugeren placerer et materiale i en given kasse. Lokationen gemmes i systemet.
* Brugeren udskriver kasses indehold til skærmen.
* Brugeren printer en liste over alle materialer og hvilken kasse de er placeret i.
* Brugeren opretter en aktivitet med et navn. Aktiviteten gemmes i systemet.
* Brugeren tilknytter et materiale til en given aktivitet. Tilknytningen gemmes i systemet.
* Brugeren udskriver en liste over hvilke kasser, der skal medbringes hvis en given aktivitet skal udføres, baseret på de tilknyttede materialers placering.
* Brugeren opretter en workshop som gemmes i systemet.
* Brugeren tilknytter en aktivitet til en given workshop. Tilknytningen gemmes i systemet.
* Brugeren printer liste over kasser, der skal benyttestil en workshop, til skærm.
* Brugeren printer en indholdsliste over kasser tilknyttet en workshop til skærmen.
* Brugeren eksporterer en liste over kasser og deres indhold for en workshop til PDF eller papirprint.
* Systemet gemmer afholdte workshops i systemet.
* Brugeren tilknytter en evaluerende kommentar til en afholdt workshop, som gemmes i systemet.
* Brugeren gemmer ofte afholdte / standardiserede workshops i systemet.
* Brugeren sletter en given kasse fra systemet.
* Brugeren sletter et materiale fra systemet.
* Brugeren sletter aktivitet fra systemet.
* Brugeren sletter workshop fra systemet.
* Brugeren fjerner materiale fra aktivitet.
* Brugeren fjerner aktiviteter fra workshop
* Brugeren ændrer en given kasses beskrivelse.
* Brugeren ændrer et materiales navn og/eller antal dele.
* Brugeren ændrer et materiales placering til anden kasse, eller “ikke i kasse”.
* Brugeren ændrer aktivitet navn.
* Brugeren ændrer workshops navn.

BMC (Business Model Canvas):



### **BPMN (Business Process Model Notation):**

**Et billede, der indeholder tekst, diagram, Plan, linje/række

Indhold genereret af kunstig intelligens kan være forkert.**

## Sprint 1

**Tildelte opgaver:**

**Jens: DCD**

**Cecilie: Objektmodel, domænemodel og wireframes**

**Benjamin: Sekvensdiagrammer**

**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:**

* Brugeren opretter en kasse med nr. og beskrivelse. Kassen bliver gemt i systemet.
* Brugeren opretter et materiale med navn og antal komponenter/dele, som gemmes i systemet.
* Brugeren placerer et materiale i en given kasse. Lokationen gemmes i systemet.
* Brugeren opretter en aktivitet med et navn. Aktiviteten gemmes i systemet.
* 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 danner ramme om aktiviteterne

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.

Et billede, der indeholder tekst, skærmbillede, diagram, cirkel

Indhold genereret af kunstig intelligens kan være forkert.

Et billede, der indeholder tekst, Post-it-note, skærmbillede, pink

Indhold genereret af kunstig intelligens kan være forkert.

### Sekvensdiagram

### Til hver af de fem valgte use cases lavede Benjamin et sekvensdiagram (SD). 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.

### Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, diagram Indhold genereret af kunstig intelligens kan være forkert.Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal Indhold genereret af kunstig intelligens kan være forkert.

Første SD

SD i Mermaid

### 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.

Vi fandt ud af, at vi ikke havde brugt Scrum korrekt. Vi skulle have udvalgt en enkelt use case til at starte med og så have valgt en ny for hvert sprint.

Vi var egentlig ved at køre ”Vandfalds”-strukturen, hvilket ikke er meningen med Scrum.

I stedet for at lave alle klasser, burde vi have nøjedes med én fælles klasse for at afprøve arkitekturen først.

Derudover gav skiftet til WPF og MVVM 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.

Vi blev ikke helt færdige med test.

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 SD til GUI-applikation – Cecilie (I samarbejde med Jens)
* Wireframes – Cecilie (I samarbejde med Jens)
* ViewModel – Jens
* Repository - Benjamin
* Test – Michael

### Design

Benjamin lavede nye SD til GUI-applikation.

### Wireframes

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

Et billede, der indeholder tekst, skærmbillede, nummer/tal, software

Indhold genereret af kunstig intelligens kan være forkert.

### 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 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

### Design

I sprint 3 blev der videreudviklet på menuerne i programmet.

### Programmering / Tests

Der blev oprettet en ObservableCollection, som dynamisk tilpasser indholdet baseret på en bool-værdi, så kun det materiale, der er markeret som tilgængeligt, vises.

Michael fortsatte med at lave ms-tests.

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

### Retrospektive meeting

## Sprint 4

### Design

### Wireframes

### Programmering / Tests

### Retrospektive meeting

# Sikkerhed

# Konklusion

# Litteratur

# Bilag

## Ordbog:

**Use Case:** Brugssituation. Forklarer hvem der gør hvad i systemet, og hvordan systemet skal reagere.