

Projekt Reolmarked

Om projektet:

Dette projekt har til formål at udvikle et system til et loppemarked, hvor man kan leje stande i form af reoler.

Projektstyringsværktøjer:

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

Vi bruger Lucidchart til versionsstyring og GitHub som repository. Repository kan ses på GitHub.

Vi følger ikke konsekvent Scrum, da vi ikke har givet hinanden roller og heller ikke holder Daily Scrum møder, men vi har alle styring på projektet i fællesskab.

Discovery:

I supplement til casen og på baggrund af den åbent-formulerede problemstilling har vi indledningsvist gennemgået en discoveryfase for at afdække potentiel værdiskabende muligheder i forretningsmodellen.

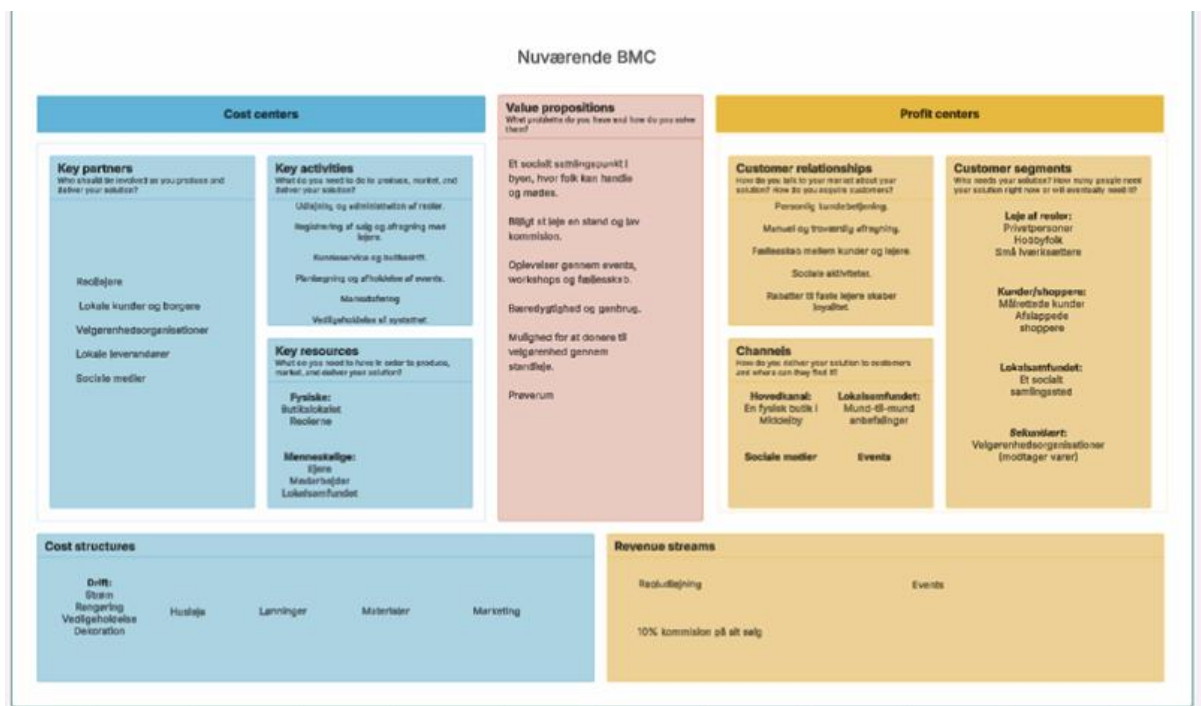
Vi oprettede Scrumboard på Lucidchart, udformede Use Cases og påbegyndte HLD.

High Level Design:

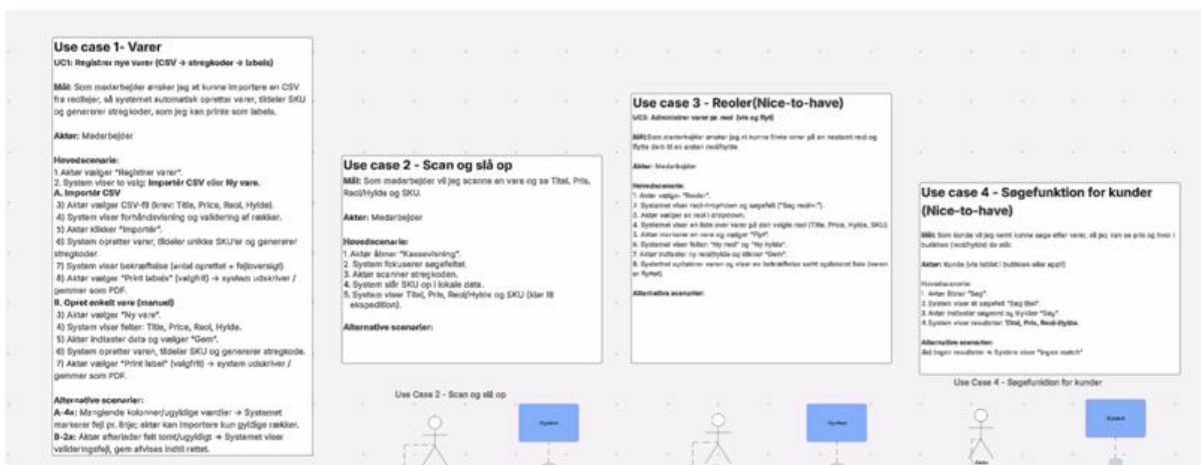
Vi udformede artefakter til HLD:

BMC, use cases, objektmodel, domænemodel, operationskontrakter
systemsekvensdiagrammer.

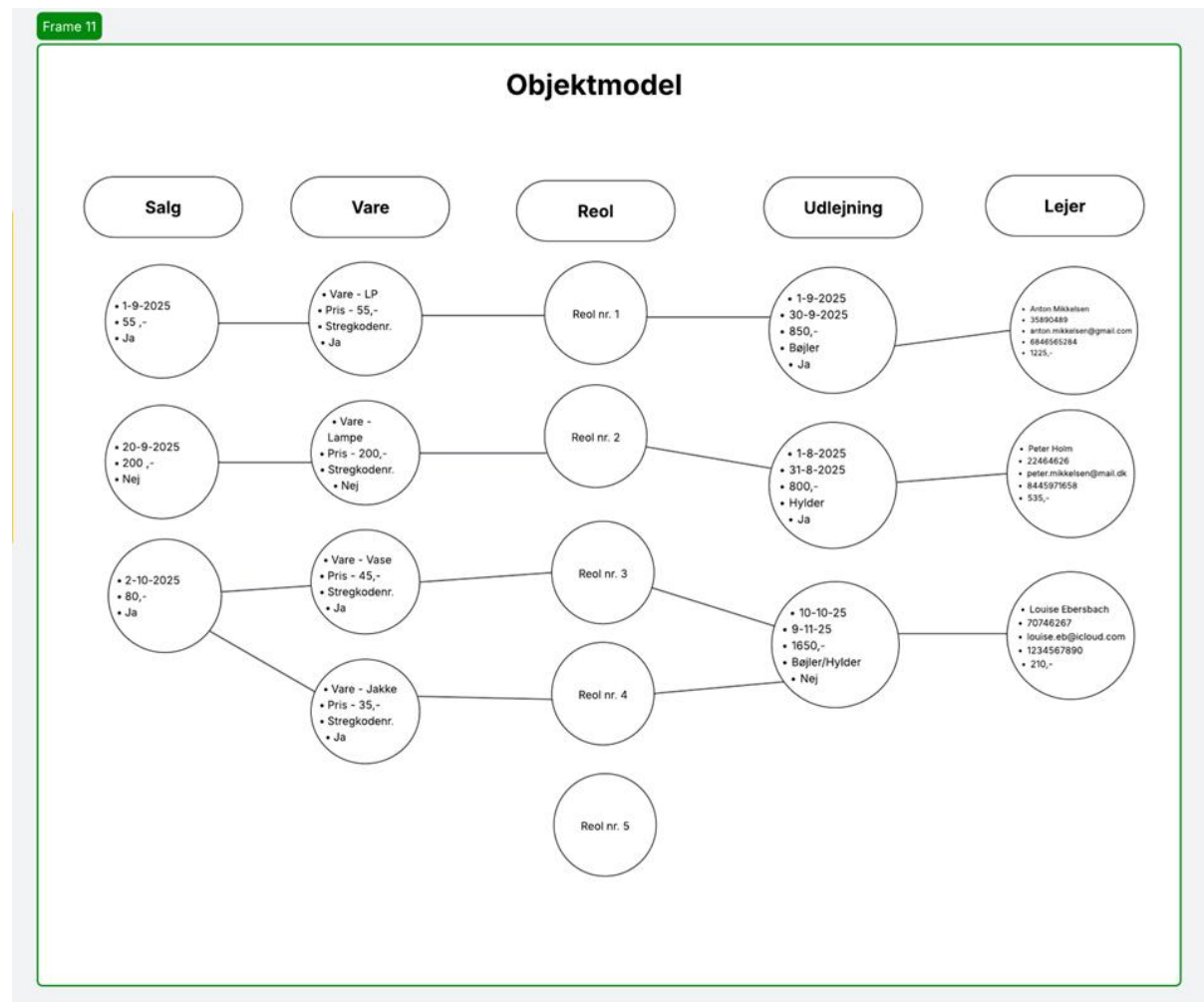
Business Model Canvas



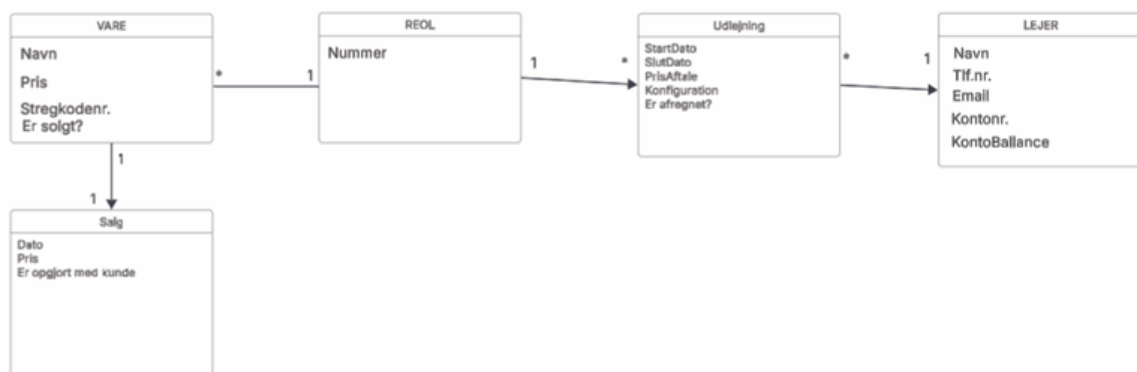
Use Cases



Objektmodel

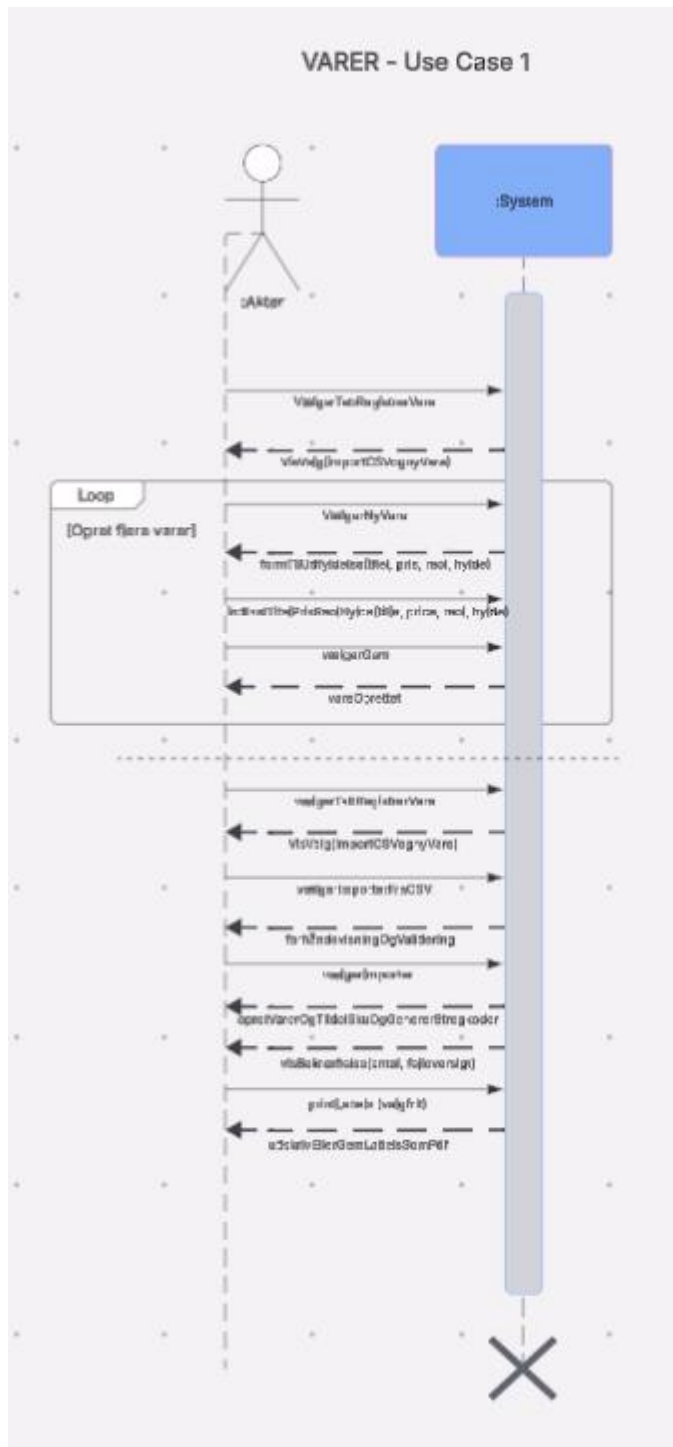


Domænemodel



Sprint 1:

Systemsekvensdiagram



Operationskontrakt - UC1 A (Import)

Operation:	ImporterVarer (csvFil: File)
Cross References:	Registrer nye varer (A Importer CSV)
Preconditions:	Medarbejderen er logget ind i systemet. csvFil er læsbar og indeholder kolonnerne: Title, Price, Reol, Hylde. Systemet er sat op til lave SKU-nummerserie og stregkode.
Postconditiong:	For hver gyldig række oprettes et Item med unikt SKU, Title, Price, Reol, Hylde og der generes en stregkode. Rækker med fejl afvises og det fremgår af en fejlrapport (linjenr. + fejlfårsag). Systemet viser en bekræftelse på oprettede varer.

Vi påbegyndte også LLD.

Low Level Design:

Artefakter blev udarbejdet.

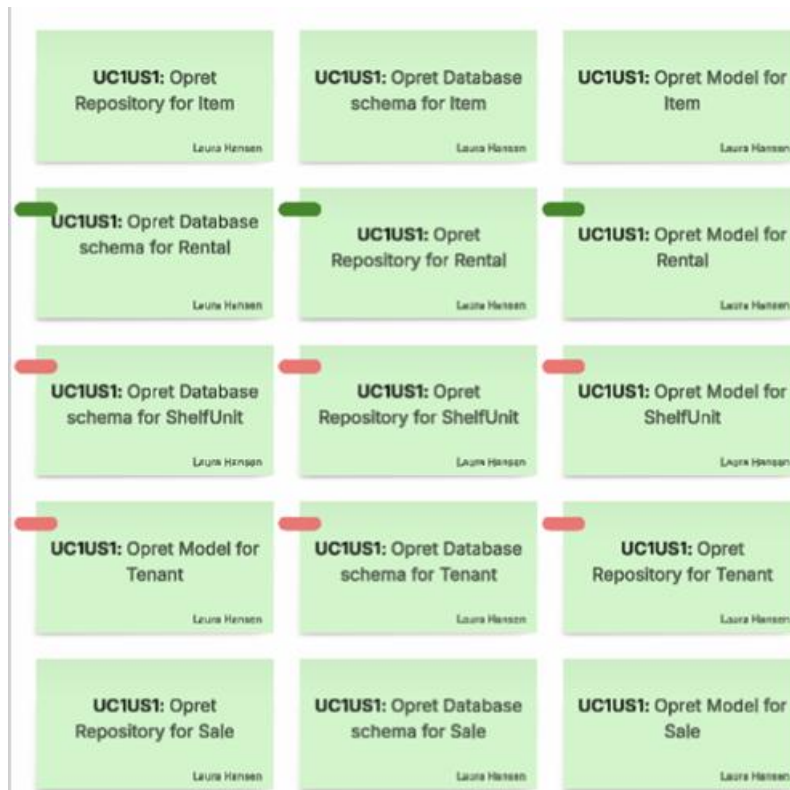
Der er oprettet kodeskelet med 3 projekter:

- Database_Connection til databasen med "Schema" folder.
- Model til alle modellerne.
- Et WPF -projekt til brugergrænsefladen.

Sprint 2:

Vi kunne nu gå i gang med at programmere.

Vi fik hver især tildelt nogle opgaver:

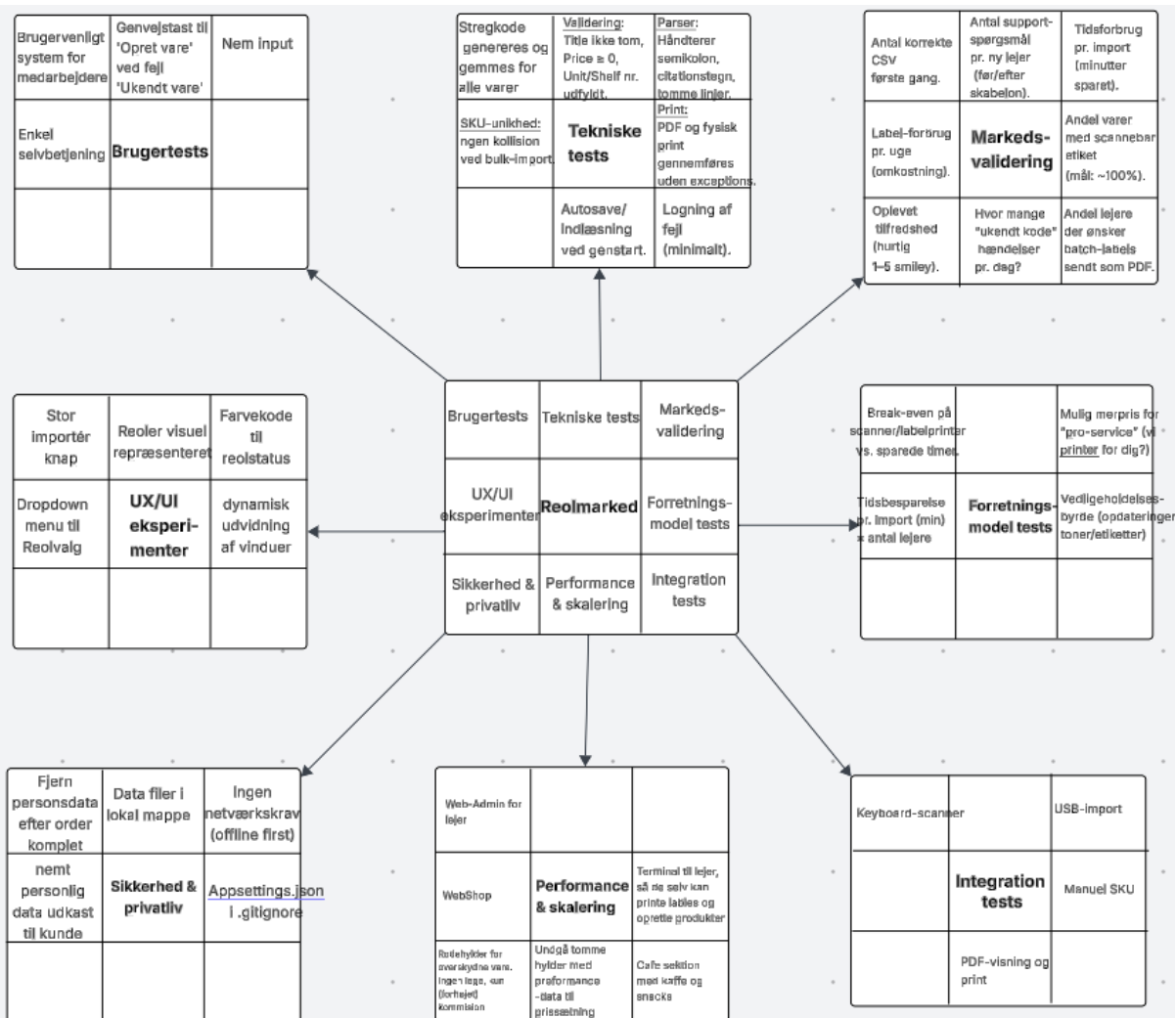


Vi udarbejdede derudover Design Class Diagram, et sekvensdiagram, et udkast low fidelity wireframes, Relationsskema og en ordbog til forklaring af de navngivelser vi bruger i systemet.

Sprint 3:

Vi blev introduceret for en ny idégenereringsmetode *Lotusblomsten*, som vi medtog som et artefakt til vores projekt:

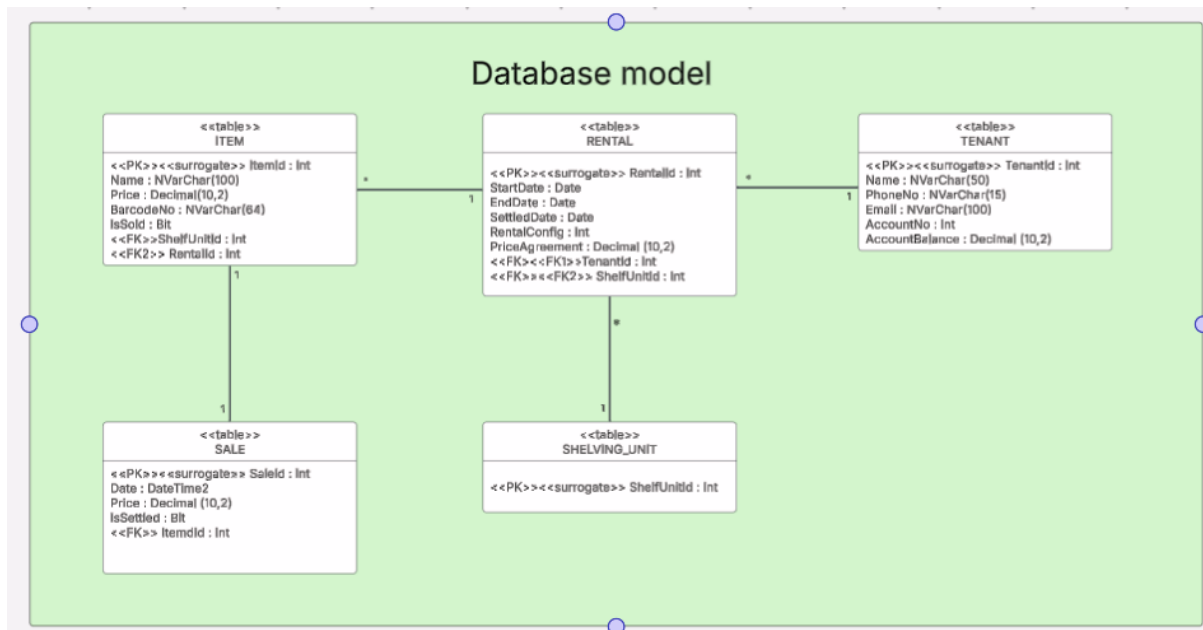
Lotusblomst



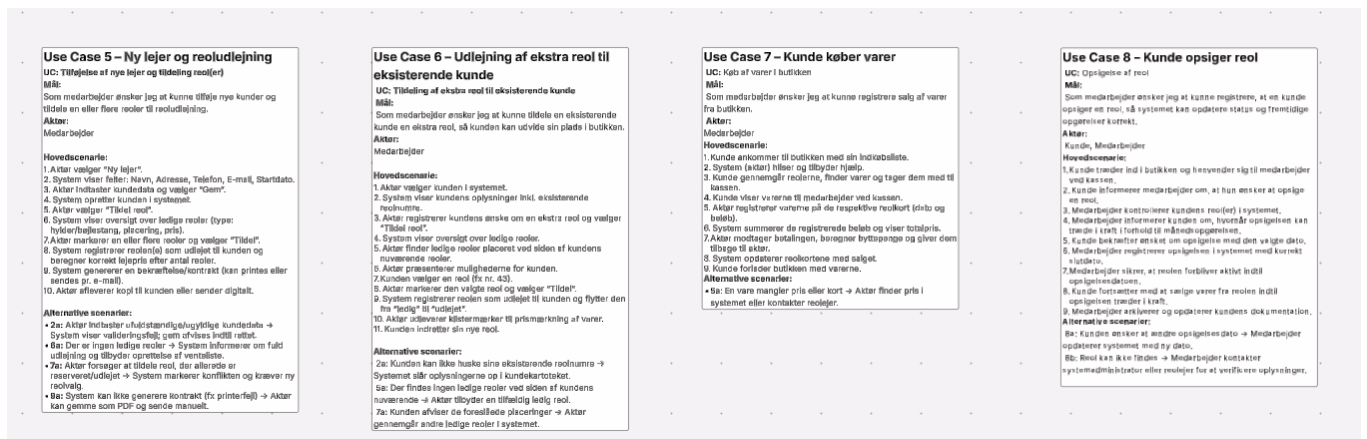
Sprint 4:

ItemView i View-laget:

Database model i LLD.



Der blev udarbejdet use cases 5-8.



Sprint 5:

Vi udarbejdede Mid-fidelity wireframes:

Reolmarkedet - medarbejder

Varer Rentaler Lejere Salg

Varer

Søg

Søg efter titel/nr

Id	Titel	Pris	Reol	Status
12	Vase	50,00	R28	På lager
...

Tilføj vare

Titel
Pris
Reol
Opret ny vare

Gem Slet

Oprettede et nyt **GUI**-projekt inde i projektet, så vi kunne starte med et helt rent projekt og så overføre fra det gamle, da det var blevet lidt rodet.

GUI-projektet indeholder 4 main views:

- Sales
- Items
- Tenants
- Rentals

Sprint 6:

Vi kunne nu konstatere, at vi ikke ville nå at lave hele programmet færdigt.

Vi valgte derfor at færdiggøre de vigtigste komponenter, så programmet kunne det mest essentielle.

Vi var tilbage og tilrette vores artefakter og opdatere Scrum-boardet.

Vi fik samlet op på koden så man nu kan:

- Tildele og vise reoler i TenantView.
- RentalView opdaterer automatisk, når man tilføjer en reol i Tenantview.
- I RentalView kan man opdatere alle parametre af en udlejning.

Retrospektive overvejelser

Der er mange ting fra projektet at reflektere over og tage med videre.

Bl.a. kunne vi med fordel have udarbejdet nogle fully dressed use cases, som kunne have givet et tydeligere overblik over systemets funktionalitet og interaktioner.

Vi nåede ikke at gennemføre kvalitetssikring af vores artefakter, herunder dokumentation, diagrammer og tests. Det betyder, at nogle artefakter kan afvige en smule fra den endelige implementering.

Vores Scrum-proces har heller ikke fungeret optimalt. Vi oplevede, at overblikket blev udfordret undervejs, og vi mistede delvist struktur i sprintforløbet. En del af årsagen var, at vi valgte at køre vores Scrum-board i *Lucidchart*, hvilket viste sig at være upraktisk. Det gjorde det svært at bevare et tydeligt overblik over opgaver, prioriteringer og status på sprintmål. Derudover har vi ikke afholdt Daily Scrums, hvilket kan have haft betydning for koordinering og fremdrift.

I selve programmet nåede vi ikke at implementere alle de funktioner, vi oprindeligt havde planlagt. Mod slutningen valgte vi at fokusere på at samle og stabilisere koden, så de vigtigste dele af systemet fungerede pålideligt.

Vi nåede heller ikke at implementere unit tests, hvilket betyder, at dele af programmet endnu ikke er testet systematisk. Det gør, at der stadig kan forekomme fejl, som først opdages under kørsel.

Efter oprydningen fungerer sammenhængen mellem *TenantView* og *RentalView* markant bedre. Det er nu muligt at tildele og vise reoler direkte i *TenantView*, og ændringer her opdateres automatisk i *RentalView*. Dette har gjort applikationen mere dynamisk og brugervenlig.

Derudover kan man i *RentalView* nu redigere og opdatere alle parametre for en udlejning, hvilket giver en mere komplet og fleksibel håndtering af data.

Dog mangler programmet stadig en del funktionalitet. Vi er blandt andet ikke helt sikre på, hvor og hvordan man bedst skal oprette en lejeperiode, og nogle knapper fungerer stadig ikke som tilsigtet. Dette gør, at brugeroplevelsen endnu ikke er fuldt gennemført.

Der har desuden været forvirring i programmet omkring brugen af ShelfUnitID og ShelvingUnitID. Vi har ikke været konsekvente i vores navngivning, og de to betegnelser bruges flere steder til forskellige formål. Det har gjort koden sværere at overskue og fejlsøge. En fremtidig forbedring vil være at gennemgå og standardisere navngivningen i hele projektet for at øge læsbarhed og vedligeholdelse.

Selvom vi ikke nåede alt det planlagte, har vi opnået en stabil og sammenhængende grundstruktur i projektet. Det giver et solidt udgangspunkt for videre udvikling, hvor vi fremadrettet kan prioritere kvalitetssikring, bedre værktøjer til Scrum og mere konsekvent kodekonvention.