

# SSB Sager – Projektrapport

[illegible]**Projektperiode:** 05-02-2019 / 29-05-2019

**Vejleder:** Jeppe Schmidt ([jpe@mmmi.sdu.dk](mailto:jpe@mmmi.sdu.dk))

**Uddannelse:** SDU, Tekniske fakultet, MMMI, Software Engineering & Teknologi,  
2. Semester. SI2-PRO-U1

**Projektgruppe:** Gruppe 25

Morten Krogh Jensen [mortj18@student.sdu.dk](mailto:mortj18@student.sdu.dk)

Nicolai Ankjær Kruuse *nikru18@student.sdu.dk*

Thomas Steinfeldt Laursen [tlaur18@student.sdu.dk](mailto:tlaur18@student.sdu.dk)Michael Haugaard Pedersen [micpe18@student.sdu.dk](mailto:micpe18@student.sdu.dk)Oliver Marco van Komen [olvan18@student.sdu.dk](mailto:olvan18@student.sdu.dk)

**Afleveringsdato:** 29-05-2019

## 1 Resume

Dette projekt omhandler et problem fra en case om EG Team Online. EG Team Online specialiserer sig i sundhedssektoren og bygger systemer til danske bosteder. De ønsker at lave ét samlet system til alle landets bosteder. Dette system skal kombinere al funktionalitet fra deres tidligere systemer, som f.eks. kalender/planlægning, dagbogshåndtering og sagsbehandling. I dette projekt er det gruppernes opgave at give et bud på hvordan sådan et system kan se ud. Hver projektgruppe skulle kun fokusere på ét (evt. flere) af de tre fokus-punkter; planlægning, dagbog eller sagsbehandling.

I denne rapport er der valgt fokuspunktet sagsbehandling. Beboere på bosteder gennemgår et forløb, hvilket dokumenteres ved hjælp af såkaldte VUM-dokumenter. Disse dokumenter indeholder særdeles følsom information om den pågældende beboer, så man skal være varsom med, hvem der har adgang til det. Derfor er problemet om datasikkerhed sat i fokus i dette projekt. Dog kan det hurtigt blive komplekst at udvikle et sådant system, og der er risiko for, at den øgede mængde af datasikkerhed går på kompromis med brugervenligheden. Dette ledte til følgende hovedspørgsmål som blev behandlet igennem projektforsøget:

Kan alle VUM-dokumenter fra alle bosteder i hele Danmark samles i én dataløsning, hvor data-lovgivning bliver overholdt og brugervenlighed af systemet er i fokus?

Det var et krav at projektarbejdet skulle følge procesmodellen Unified Process (UP). På grund af projektets begrænsede størrelse, gav det kun mening at gennemgå de to første faser af UP; inceptionsfasen og elaborationsfasen. I inceptionsfasen blev problemet undersøgt, og der blev udarbejdet en brugsmønstermodel for at finde de mest essentielle krav. Dette dannede grundlag for elaborationsfasen som skulle kombineres med Scrum. Elaborationsfasen var delt op i to iterationer som hver dækkede over et scrum-sprint. I starten af fasen blev der udarbejdet en product backlog baseret på inceptionsfasens brugsmønstre og supplerende krav. Ud fra denne kunne projektgruppen samle en sprint-backlog i starten af hvert sprint. Denne kombination af agile arbejdsmetoder igennem projektet sikrede et fleksibelt arbejdsmiljø.

Igennem udviklingsforløbet lykkedes det at få skabt et program der var i stand til at begrænse hvilken data der var synlig for bestemte brugere. Systemet er rollebaseret, hvor hver rolle har adgang til forskellig funktionalitet i programmet. Derudover kan brugerne udelukkende se data fra de bosteder, som de er tilknyttet. Disse bosteder kan bestemmes af en bruger med admin-rettigheder som er i stand til at logge ind i systemet og bestemme hvem der har ret til systemets funktioner og beboeres data. Systemet indeholder derudover forskellige hjælpefunktioner til medarbejdere der behandler beboeres sager. De kan f.eks. oprette, se og udfylde VUM-dokumenter, med gode støttefunktioner såsom validering af tekst og autoudfyldning af felter. Generelt endte systemet med at have nogle gode og relevante funktioner. Dog kunne datasikkerhed og brugervenlighed have været bedre. F.eks. kunne databasens indhold krypteres for at få øget sikkerhed og brugervenligheden burde have været mere i fokus igennem udviklingsprocessen. Gruppen fik ikke foretaget de nødvendige tests og fik derfor ikke feedback på dette aspekt af løsningen.

## 2 Forord

Denne rapport er resultatet af et semesterprojekt, udarbejdet i perioden 5. februar til den 29. maj, ved 2. semester i software Engineering og software teknologi på Syddansk universitet Odense, under vejledning af Jeppe Schmidt. Rapporten henvender sig til lærer og vejledere inden for Software Engineering, samt andre med ekspertise eller interesse inden for området.

Hensigten med rapporten, er at rapportere den arbejdsproces som blev anvendt til udvikling af det software som skulle udvikles på baggrund af EG team Online's oplæg. Oplægget har fokus på bosteder og sagsudredning i danske kommuner. For at løse denne problemstilling blev der udviklet et software ud fra sagsbehandling aspektet, som skal forsøge at forsimple platformen til sagsudredninger og behandlinger.

Aine M.v.K.  
Thomas Steinfeldt  
Michael H. Pedersen  
Niclas Krause  
Morten K. Jensen

### 3 Indhold

1	Resume .....	2
2	Forord .....	3
3	Indhold.....	4
4	Læsevejledning.....	6
5	Redaktionelt.....	7
6	Ordliste .....	8
7	Indledning.....	9
8	Metode og planlægning .....	10
8.1	Metode.....	10
8.2	Planlægning .....	10
8.2.1	Sprint-planlægning og backlogs .....	10
8.2.2	Rollefordelingen i Scrum.....	11
8.2.3	Ceremonier i Scrum.....	11
8.2.4	ScrumButs i dette projekt .....	11
9	Hovedtekst .....	12
9.1	Teori .....	12
9.1.1	Voksen udredningsmetoden .....	12
9.1.2	General Data Protection Regulation .....	13
9.2	Overordnet kravspecifikation .....	15
9.2.1	Brugsmønsterdiagrammer .....	15
9.3	Detaljeret kravspecifikation.....	17
9.3.1	Detaljerede brugsmønsterbeskrivelser.....	17
9.3.2	Supplerende krav .....	23
9.3.3	Sprint backlog.....	23
9.4	Analyse .....	26
9.4.1	Statiske side af analysemodellen .....	26
9.4.2	Dynamiske side af analysemodellen.....	28
9.5	Design.....	31
9.5.1	Softwarearkitektur .....	31
9.5.2	Subsystemdesign .....	32
9.5.3	Statiske side af designmodellen.....	32
9.5.4	Dynamiske side af designmodellen .....	32
9.5.5	Design af persistenslag: .....	33
9.6	Implementering.....	35
9.6.1	Kontrol af brugers rettigheder .....	35

9.6.2	Administrator muligheder.....	36
9.6.3	Logning af systemet.....	37
9.6.4	Afsnit om databaseimplementering .....	39
9.6.5	Threads .....	39
9.6.6	Delkonklusion.....	40
9.7	Test .....	40
10	Diskussion .....	42
11	Konklusion .....	45
12	Perspektivering .....	46
13	Bibliografi.....	48
14	Procesrapport.....	49
14.1	Læring og refleksion.....	49
14.2	Projektstyring .....	51
14.3	Identifikation, analyse og bearbejdning af problemer .....	51
14.4	Udviklingsprocessen.....	51
14.5	Formidling og kommunikation.....	52
14.6	Samarbejde i gruppen .....	52
14.7	Samarbejde med vejleder.....	53
15	Kildekode .....	54
16	Brugervejledning .....	55
16.1	Opstart af programmet.....	55
16.2	Programmets login-system .....	55
16.3	SQL-script fil.....	55
17	Projektlog .....	56
18	Interne bilag.....	56
18.1	Rapportkontrolskema.....	56
18.2	Inceptionsdokument .....	61
18.3	Testresultater .....	90
18.4	Projektforslag .....	93

## 4 Læsevejledning

For at få det største udbytte af denne rapport skrevet af Gruppe 25, på 2.Semester af Software Engineering og Software Teknologi, anbefales det at læse rapporten kronologisk. I starten af rapporten findes en indholdsfortegnelse, hvis læseren ønsker et overblik over rapportens indhold, eller ønsker at gå direkte til en given overskrift. Der vil igennem rapporten linkes til figurer. Ligeledes henvises der i rapporten til bilaget som er placeret sidst i rapporten. Hvis der igennem rapporten skulle forekomme ukendte ord, er der til dette formål lavet en ordliste som giver en kort forklaring af udvalgte fagbegrebers betydning, samt en kort beskrivelse af redskaberne brugt i dette semesterprojekt. Rapporten henvender sig til lærer og vejledere inden for Software Engineering og Software Teknologi, samt andre med ekspertise eller interesse inden for området.

## 5 Redaktionelt

Afsnit	Ansvarlig	Bidrag fra	Kontrolleret af
Titelblad	Nicolai		Alle
Resume	Thomas	Morten	Alle
Forord	Michael		Alle
Læsevejledning	Morten		Alle
Indledning	Thomas		Alle
Metode og planlægning:	Nicolai	Thomas	Alle
Hovedafsnit			Alle
Indledning	Michael		Alle
Teori	Michael		Alle
Overordnet kravspecifikation	Morten	Thomas	Alle
Detaljeret kravspecifikation	Thomas		Alle
Analyse	Oliver		Morten
Design	Oliver		Morten
Implementering	Morten		Oliver
Test	Oliver	Thomas	Morten
Diskussion	Oliver		Alle
Konklusion	Oliver		Alle
Perspektivering:	Thomas, Morten	Oliver	Alle
Litteraturliste	Michael		Alle
Læring og refleksion	Alle		
Projektarbejdet	Alle		
Identifikation af problemet	Alle		
Analyse af problemet	Alle		
Bearbejdning af problemet	Alle		
Projektforløb	Alle		
Formidling og kommunikation	Alle		
Samarbejde i gruppen	Alle		
Samarbejde med vejleder	Alle		
Kildekode	Nicolai		Alle
Brugervejledning	Nicolai	Thomas, Oliver	Morten

## 6 Ordliste

**Scrum:** Scrum er en agil udviklingsmetode som kan bruges i forbindelse med softwareudvikling.

**CRUD-operationer:** CRUD-operationer står for Create, Read, Update og Delete. Det er de essentielle handlinger en database kan udføre på data. Lave data, læse data, opdatere data og slette data.

**JDBC:** Java database connectivity, er en API for programmeringssproget Java som bliver benyttet i dette projekt. Den består af en lang række handlinger som kommunikerer og interagerer med en database.

**Query:** En forespørgsel om hentning af data i databasen.

**E/R diagram:** Entity relationship diagram også kaldet ERD, viser forbindelserne mellem relationelle tabeller i en database.

**SQLite:** Et relationel database administrations sprog som implementerer alle SQL-operationer og som gemmer alt data i én fil.

**PostgreSQL:** Et relationel database administrations-sprog som implementerer SQL-operationer, er open-source og som tilbyder en lang række funktioner udover grundlæggende SQL.

**Java.util.logging:** En indbygget pakke i Java, som gør det muligt at logge systemet og gemme loggen som fil eller i database.

**VUM:** Voksen udredningsmetoden er et sagsbehandlingsværktøj, som bliver brugt til behandling af udfordrede personer og handicappede i de danske kommuner.

**GDPR:** General Data Protection Regulation er et en regulerings pakke som blev vedtaget af EU parlamentet. Pakken har til formål at regulere behandlingen og indsamlingen af persondata i virksomheder.

**DHUV:** Digitalisering af handicap og udsatte voksne. DHUV er en administrativ projekt og implementerings værktøj af VUM.

### NetBeans

Netbeans er en IDE (Integrated Development Environment) der er blevet brugt til udvikling i dette projekt

### Scenebuilder

Scenebuilder er et visuelt layout værktøj til JavaFX, der gør det mere overskueligt at arbejde med JavaFX og få et "live" view af ens arbejde.



## 7 Indledning

Overskriften for dette semesters projekt var ”Organisationsorienteret softwareudvikling”. I dette projekt skulle grupperne få indsigt i softwareudvikling i en organisatorisk kontekst. Dette opnåedes ved et tæt samarbejde med organisationen EG Team Online igennem forløbet. De udleverede en case med et problem der skulle løses, og det var op til projektgrupperne at finde ud af hvordan.

EG Team Online leverede i 2005 deres første store system, som nu hedder Sensum Bosted, til lokale bosteder og institutioner. De har siden da opnået dyb forståelse for de udfordringer, som bosteder og sagsbehandlere står overfor. Siden da, er der kommet to store udfordringer. En økonomiaftale fra 2013 bestemte at IT-løsninger for en række offentlige områder, heriblandt sagsoverblik, skulle udvikles som landsdækkende løsninger, hvor alle kommuner har indflydelse på løsningerne. Derudover blev der i 2016 vedtaget en ny EU-databeskyttelsesforordning, som har betydet ændringer i, hvordan personfølsomme data skal håndteres. For EG Team Online betyder det, at deres produkter skal samles til en landsdækkende løsning for alle bosteder i Danmark, og at socialfaglige medarbejdere kun ser nødvendige data på patienter, som har afgivet samtykke til det. I dette projekt arbejdes der derfor med sagsudredning.

Når en borger er under udredning på et bosted, bliver der gemt data ved brug af VUM-redskabs dokumenter. Borgerens data er følsomme, og samlingen af alle IT-løsninger for bosteder under ét system kræver yderligere sikkerhedsforbehold mht. dataopbevaring. Dataene må ikke kunne tilgås af andre end borgeren selv og af relevant sundhedsfagligt personale, hvortil borgeren har givet samtykke. Dette betyder, at når sundhedsfaglige medarbejdere der arbejder med patienten, skal have adgang til nødvendig information, skal der kun vises de informationer, der er behov for, og som borgeren har givet samtykke til. Dette kan være svært, da dataene både skal gemmes væk, så de ikke kan tilgås, men samtidigt være tilgængelig for de rette personer. Det kræver en høj mængde autentificering af personen, som vil tilgå patientens data. Det kan hurtigt blive komplekst at udvikle et sådant system, og der er risiko for, at den øgede mængde af datasikkerhed går på kompromis med brugervenligheden i den daglige brug af den samlede IT-løsning for alle bosteder. Denne problematik ledte derfor til hovedspørgsmålet:

Kan alle VUM-dokumenter fra alle bosteder i hele Danmark samles i én dataløsning, hvor data-lovgivning bliver overholdt og brugervenlighed af systemet er i fokus?

For at kunne besvare hovedspørgsmålet skulle følgende spørgsmål besvares igennem projektforløbet:

- Er det muligt at begrænse brugeres adgangsrettigheder baseret på deres jobtitler?
- Er det muligt at begrænse brugeres adgangsrettigheder baseret på deres tilknyttede bosteder?
- Skal beboeren selv have mulighed for at tilgå systemet?
- Vil en trelagsarkitektur kunne øge mængden af datasikkerhed i dette system?
- Vil GDPR-loven være relevant at tage højde for i dette system?
- Kan der designes en database, som er hurtig nok til at hente data til en travl hverdag på bosteder?
- Kan en skybaseret PostgreSQL-database tilkobles et Java-program?
- Kan en PostgreSQL-database gemme dynamisk og komplekst data som VUM-dokumenter?
- Kan kritiske handlinger foretaget af medarbejdere blive gemt i tilfælde af eventuelle fejl?

Igennem projektforløbet har det været meningen, at grupperne skulle anvende brugsmønsterstyret iterative processer som UP og Scrum. Dette gav indsigt i brugen af ingeniørens arbejdstilgang, da projektet krævede planlægning, selvstændigt arbejde og anvendelse af udvalgte redskaber. Samtidig ville projektet også komme omkring emner fra semesterets resterende fag. I forbindelse med videregående objektorienteret programmering, er der blevet brugt Java, threads, J-Unit, og generelt den objektorienterede tankegang samt læringen om UP fra Organisation og Software Engineering, til at skabe vores produkt. I forbindelse med faget

databasesystemer, skulle der fremstilles en database som gemte informationer omkring brugere, beboere, dokumenter samt brugernes roller.

## 8 Metode og planlægning

I det følgende afsnit vil der blive gennemgået de metoder som er blevet brugt i projektet til at svare på hovedspørgsmålet. Afsnittet beskriver bl.a. brugen af UP og Scrum i projektet samt fordele og ulemper. Projektgruppens specifikke brug af metoderne vil blive beskrevet, og hvordan brugen i projektet afviger fra de officielle definitioner af metoderne.

### 8.1 Metode

For at besvare hovedspørgsmålet og underspørgsmålene blev Unified Process (UP) -procesmodellen brugt. Denne softwareprocesmodel er bygget op af fire faser; inceptionsfasen, elaborationsfasen, konstruktionsfasen og transitionsfasen. På grund af projektets begrænsede størrelse, er det kun de to første faser af UP; inceptionsfasen og elaborationsfasen, som er brugt. Målet i inceptionsfasen er at få skudt projektet i gang og sikre, at projektet er muligt at gennemføre. I projektet er det blevet gjort ved at foretage markedsundersøgelser, dokumentere essentielle krav og identificere de mest kritiske risici. En stor del af arbejdet lå i kravspecifikationen, da der her skulle udarbejdes brugsmønstre og domænemodeller som danner fundamentet for designet i den næste fase, elaborationsfasen. Krav- og analyse var derfor en væsentlig del af denne fase.

I elaborationsfasen begyndte design- og implementering at gå i gang. Her begyndte gruppen at udvikle løsningen. Denne udvikling foregik over to iterationer, som var nogenlunde lige lange i elaborationsfasen. Derudover var elaborationsfasen kombineret med Scrum. Hver iteration dækkede over ét Scrum-sprint, hvor de kendte værktøjer fra Scrum blev anvendt. UP og Scrum kombineres for at få en agil arbejdsproces, hvori der er klare retningslinjer ift. hvilket arbejde der skal udføres og hvilken prioritering samt hvilket tidsrum at det udføres.

Kombinationen af UP og Scrum giver projektet et struktureret miljø. Resultaterne fra UP's inceptionsfase hjælper med at udarbejde product backlog og sprint backlogs til Scrum. Ved brug af disse backlogs er fremtidige arbejdsopgaver beskrevet så man efter afslutning af en del-opgave er klar over næste opgave. Inceptionsfasen fra UP giver en god idé om hvad produktet skal ende ud med.

En af ulemperne ved brug af en agil proces som UP, er at teams helst skal bestå af projektmedlemmer som er eksperter inde for det område de arbejder med, en metode der umiddelbart ikke passer på studerende der ikke kan anses som eksperter i et fagområde de stadig er under læring i. Derudover har det i dette projekt ikke været muligt at følge Scrums regelsæt fuldt ud. Scrum er lavet til at blive brugt på arbejdsmarkedets rigtige projekter, og da dette var et skoleprojekt som er tidsbegrænset og af væsentlig mindre størrelse end et typisk projekt, var det besværligt at opfylde alle roller og holde alle møder.

### 8.2 Planlægning

#### 8.2.1 Sprint-planlægning og backlogs

Inceptionsdokumentets resultater blev i dette projekt brugt til planlægning af første iteration i elaborationsfasen. Med udgangspunkt i brugsmønsterdiagrammer, brugsmønsterbeskrivelser, aktørlistes og domænemodeller, blev der prioriteret krav ved hjælp af MoSCoW-metoden. Dette dannede grundlag for product backloggen. Product backlog er en liste af alle krav og al funktionalitet som systemet gerne skulle opfylde i slutningen af projektførløbet. Heri er alle krav prioriteret, så udviklingsholdet ved hvad der skal udarbejdes først. I starten af hver iteration kiggede udviklingsholdet på product backlog og udvalgte de vigtigste krav som skulle implementeres i løbet af det pågældende sprint. Disse krav blev tilføjet til sprintets sprint-backlog, som indeholdt alle disse krav samt prioritering og en estimering af tiden det ville tage at implementere hvert krav. Denne backlog var hvad udviklingsholdet skulle holde sig til, hvis de skulle starte en ny opgave. Efter første sprint blev der udviklet en ny sprint-backlog, som indeholdt de krav der skulle udarbejdes i andet sprint/iteration af

elaborationsfasen. Disse krav blev taget fra product backloggen samt den mængde krav der ikke nåede at blive implementeret i første sprint-backlog.

### 8.2.2 Rollefordelingen i Scrum

Med Scrum følger tre forskellige roller, som deltagerne kan påtage sig. De tre roller er; product owner, development team (udviklingsholdet) og Scrum master. Product owner skal sikre at arbejdet skaber så meget værdi som muligt ved at opretholde product backloggen. Product owner repræsenterer kunden i projektet og ved derfor præcist hvad der skal laves, men ikke hvordan det skal laves. I dette projekt var det ikke muligt at få en person fra EG Team Online med ind over gruppens projekt. Derfor var gruppen selv nødt til at tage rollen som product owner. Gruppen blev enige om, at alle medlemmer tog rollen som product owner i fællesskab. Dermed var alle fælles om at opretholde product backlog og vurdere udviklingen.

Udviklingsholdet består typisk af ca. fire til otte personer, og har til ansvar for at udvikle løsningen. Udviklingsholdet er helt og aldeles selvstyrende og bestemmer selv hvordan der skal tages hånd om udviklingen. De står sågar for at samle hver sprint-backlog ud fra product backloggen i starten af hvert sprint. I dette projekt blev gruppen enige om, at alle gruppemedlemmer var en del af udviklingsholdet samtidig med, at de var product owners. Udviklingsholdet står for at levere program der opfylder deres definition af "done". Dette er en fælles definition, der fortæller hvornår et stykke arbejde er færdig på programmet. Definitionen af "done" blev i dette projekt bestemt ved at sætte bestemte mål for kravene. Disse konkrete og målbare mål skulle programmet opfylde før man kunne gå videre til næste opgave.

Den sidste rolle, Scrum master, står for at hele scrum-holdet overholder reglerne for Scrum-metoden. Det er scrum master der indkalder til daily scrum. Scrum master har også til opgave at holde udviklingsholdets miljø behageligt for optimal arbejds morale. Rollen som Scrum master påsatte gruppen én person, som skulle have ekstra fokus på daily scrum og hvornår pauser skulle holdes.

### 8.2.3 Ceremonier i Scrum

Scrum-metoden indeholder nogle ceremonier der afholdes på forskellige tidspunkter igennem sprintsne. Gruppen startede hvert sprint planlægning. Baseret på product backlog, diskuterede udviklingsholdet hvilken funktionalitet der kunne udvikles i det kommende sprint. Planlægningen resulterede i en sprint backlog.

Daily scrum meetings, er et værktøj der er ment til at holde en projektgruppe opdateret på daglig basis. Disse møder bliver indkaldt til og styret af den givne Scrum Master. Ifølge Scrum, skal der afholdes daily scrum hver eneste dag. Her bruger holdet maks. 15 minutter på at snakke om hvad de har lavet siden sidst, hvad de skal lave nu, og om de har haft nogen problemer med det. Hvis det er tilfældet, er det op til scrum master at sørge for, at den pågældende udvikler får hjælp. Mødet bliver ofte afholdt stående for at sikre, at mødet ikke trækker ud. Projektgruppen gjorde meget ud af at afholde daily scrum i starten af hvert gruppemøde. På denne måde var alle deltagere opdateret omkring arbejdet.

I slutningen af hvert sprint afholdes sprint review. Her mødes scrum-holdet med andre interessenter, og der tages der et kig på hvad der opnået igennem sprintet. Baseret på dette, foretages der eventuelt ændringer i product backlog. Derudover ser man på hvad der skal foretages videre for at maksimere den værdi der skabes.

Kort efter sprint review afholdes sprint retrospektiv. Dette er et internt møde mellem scrum-holdet. Her kigger man tilbage på det afsluttede sprint og diskuterer hvad der er gået godt og skidt i forhold til processer, metoder, værktøjer og mennesker. Holdet bliver derefter enige om hvordan næste sprint kan forbedres i disse afdelinger.

### 8.2.4 ScrumButs i dette projekt

Det er ikke alle projekter hvor det er muligt eller giver mening at anvende alle værktøjer fra Scrum. Her må projektgruppen lave undtagelser i form af ScrumButs. I dette projekt er der gjort brug af flere ScrumButs. For det første, har det ikke været muligt for gruppemedlemmerne at afholde daily scrum på daglig basis, da der

kun var sat tid af til projektet ca. to gange om ugen. Denne ScrumBut var ikke til for stort besvær, da møderne to gange om ugen viste sig at være tilstrækkelige ift. Mængden af arbejde projektet har krævet.

En anden scrumbut er Product owner rollen. Under rollefordelingsafsnittet blev det nævnt, at alle gruppemedlemmer tog rollen som product owner. Som tidligere beskrevet, var dette nødvendigt, da gruppen ikke havde mulighed for at assimilere en person der repræsenterede virksomheden. Dette gav friere tøjler i forhold til hvad der skulle udvikles, da gruppemedlemmerne nu selv bestemte over product backlog. Gruppen valgte denne løsning, da de mente, at alle skulle have lige meget autoritet over hvad produktet skulle indeholde. Men på grund af mangel på en rigtig product owner, var det ikke muligt at afholde ordentlige sprint reviews. Til gengæld afholdtes der et projektseminar med gruppens vejleder, der fungerede som en erstatning.

## 9 Hovedtekst

I det kommende afsnit vil der blive redegjort for teori og UP's inceptions- og elaborationsfaser som er blevet brugt gennem projektet. UP indeholder bestemte workflows som er vigtige for projektet. Disse workflows er som følgende; overordnet og detaljeret kravspecifikation, analyse, design, implementering og test. Der vil i nogle af afsnittene komme eksempler på både første og anden iteration, for at give bedre forståelse til de overvejelser og valg som er blevet taget gennem projektet.

### 9.1 Teori

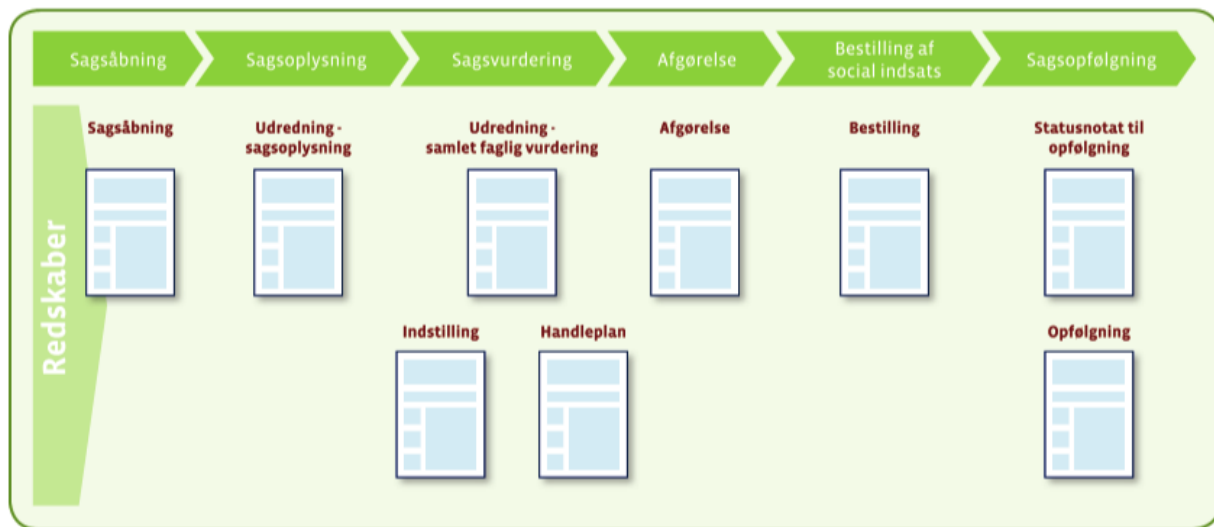
I teoriafsnittet bliver der beskrevet ekstra viden, som er blevet anvendt gennem projektet. I dette omfang vil der blive gennemgået, hvad voksen udredningsmetoden (VUM) er og hvordan de bliver brugt. Samt hvad General Data Protection (GDPR) er og hvilken betydning der har for organisationer.

#### 9.1.1 Voksen udredningsmetoden

Voksen udredningsmetoden er en metode, som bliver brugt i de danske kommunale bosteder. Bostederne bruger denne metode til at behandle personer med handicap, misbrug eller andre udsatte. VUM blev udarbejdet af KL og staten, og har resulteret i at kommunernes sagsbehandlinger er blevet mere systematiske samt borger orienteret. Det gavner derfor sagsbehandlerens forløb, fra en beboer møder op og vurderet til at være berettiget kommunal støtte og til beboeren er behandlet, og beboernes oplevelse af at modtage hjælp.

For at implementere VUM, bliver DHUV brugt til at introducere og træner personalet til at bruge de værktøjer som er til rådighed i VUM-metoden. Til dette formål, er der udarbejdet en implementeringsvejledning (Socialstyrelsen, 2019). Når DHUV bliver brugt, er det optimalt for arbejdspladsen at indsætte nye ledere, oftest ledere som har erfaring på VUM og optræningen af personalet. Dette skaber også et, forhåbentlig, nyt og forbedret arbejdsmiljø. Dette sker oftest, ved at lave en gradvis indførelse af VUM. Dette sker ved at træne superbrugere som har ekstra meget indsigt i VUM, og derved også hjælpe lederne med at fremme implementeringen i resten af afdelingen.

VUM består af en række redskaber som, gennem en sagsbehandling, bliver brugt til at strukturere forløbet og sikre at det bliver lovmæssigt gennemgået det som den udsatte borger har behov for. Dette bliver gjort gennem de forskellige dokumenter som VUM består af. Dette handlingsforløb starter med at sagsbehandleren, via et sagsåbings dokumentet indsamler oplysninger som er nødvendige for det videre forløb, samt oplysning om borgerens rettigheder. Forløbet kan blandt andet ses i billedet nedenfor.



Figur 1- Sagsbehandlingsforløb

Forløbet fortsætter, efter sagsåbning med udredningssagsoplysnings dokumentet, hvor borgerens behov bliver vurderet og klargjort til det videre forløb i sagsvurderings forløbet. Dette forløb beskæftiger sig hovedsageligt med at udarbejde en vurdering og ud fra vurderingen en handleplan, og hvis der er behov for det yderligere dokumenter, som skaber basis for hvad beboeren kan forvente at komme igennem. Resten af forløbene er udførelsen af handleplan, og de potentielle ekstra behov der måtte være, samt en samlet vurdering for at afslutte forløbet eller revurdere forløbet og potentielt skabe en ny handlingsplan.

### 9.1.2 General Data Protection Regulation

GDPR er en persondatalovgivning som blev vedtaget tilbage i 2016 af det europæiske parlament. Lovgivningen trådte officielt til i 2018 og betød at virksomheder skulle håndtere persondata på en mere sikker måde. Dette bliver opnået gennem mange reguleringer, hvoraf fem af dem er ekstra vigtige for virksomheder (GDPR.DK ApS, 2019).

- Føre en fortegnelse.
- Dokumentere at lovgivningens principper for god databehandling efterleves.
- Dokumentere at virksomheden har indført passende tekniske og organisatoriske foranstaltninger.
- Oplyse kunder og ansatte om hvordan deres data behandles.
- Bevise, at virksomheden efterlever lovgivningen fx hvis der anvendes samtykke, databehandlere, mv.

#### Fortegnelse:

Ud fra artikel 30, betegnes en fortegnelse som en kortlægning af hvordan en virksomhed behandler personoplysninger. En virksomhed skal, i henhold til artikel 30, lave en fortegnelse for at kunne dokumentere bearbejdningen af personoplysninger. (GDPR.DK ApS, 2019)

#### Principper for god databehandling:

Der er 7 principper som fortæller, hvordan virksomheder bør bearbejde personoplysninger. Disse 7 principper er som følgende:

- Ansvarlighed
- Formålsbegrænsning
- Dataminimering
- Lovlighed, rimelighed og gennemsigtighed

- Rigtighed
- Integritet og fortrolighed
- Opbevaringsbegrænsning

For yderligere information kan det findes via denne kilde (GDPR.DK APS, 2019). Principperne gennemgår blandt andet, god praksis til behandling af persondata.

**Passende tekniske og organisatoriske foranstaltninger:**

Her er der fokus på det mere tekniske, så som kryptering. Dette gælder både for den data som virksomheden har indsamlet, men også indsamler.

**Oplyse kunder og ansatte om hvordan deres data behandles:**

Betyder at en bruger til en hver tid har ret til en udprintet samling, af den data som virksomheden har indsamlet. Blandt andet også at virksomheden skal oplyse om, hvilken data det er de indsamler.

**Bevise, at virksomheden efterlever lovgivningen fx hvis der anvendes samtykke, databehandlere, mv:**

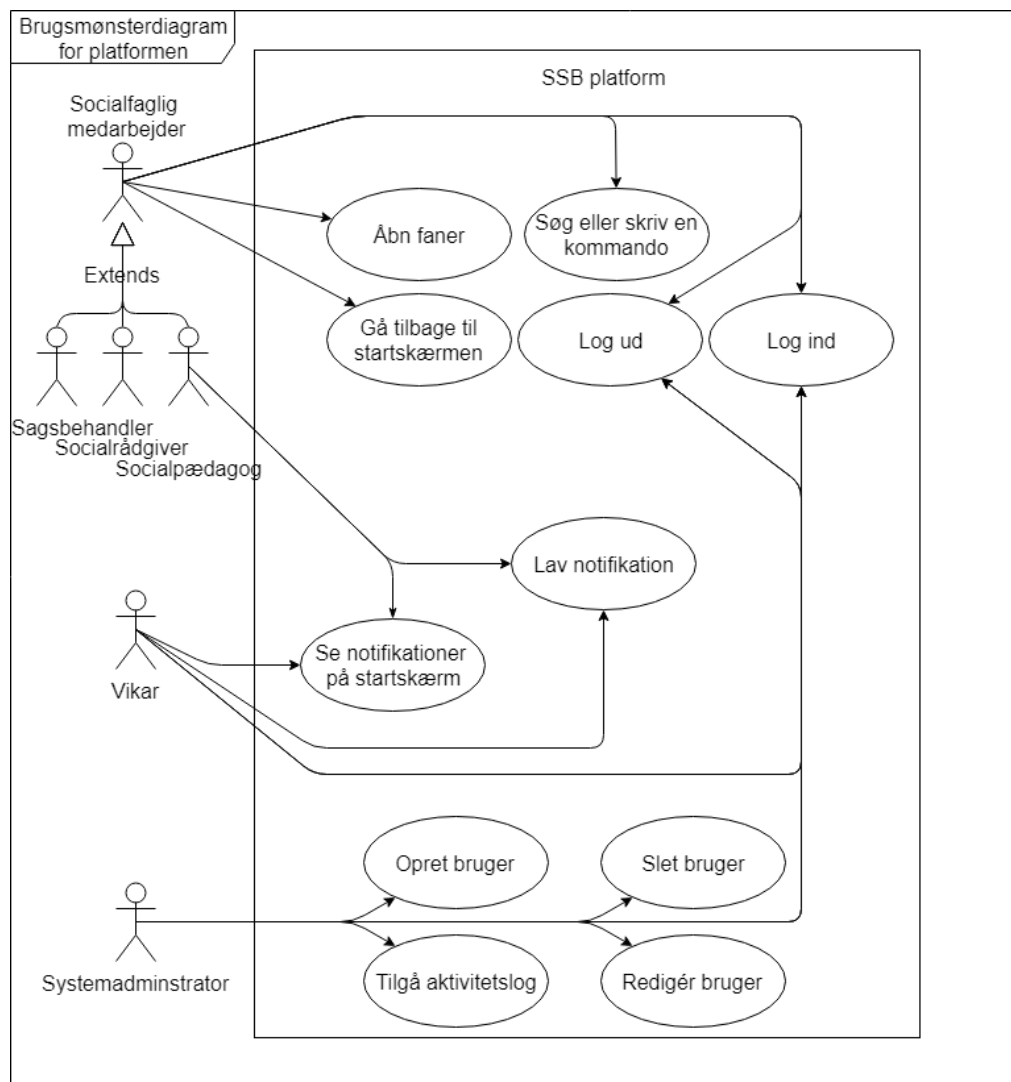
En virksomhed skal kunne vise dokumentation for data- behandling og opsamling.

## 9.2 Overordnet kravspecifikation

SSB-systemet skal være en samling af alle Sensum og Sensum bosteds moduler, hvor medarbejdere skal kunne tilgå notifikationer og opdateringer på hjem-skærmen. Derudover skal medarbejderne kunne skifte imellem relevante sider, såsom sager siden, som skal håndtere sagsbehandling for socialrådgiver og sagsbehandler. Sager siden skal kunne vise VUM-dokumenter og give medarbejdere mulighed for at oprette nye sager.

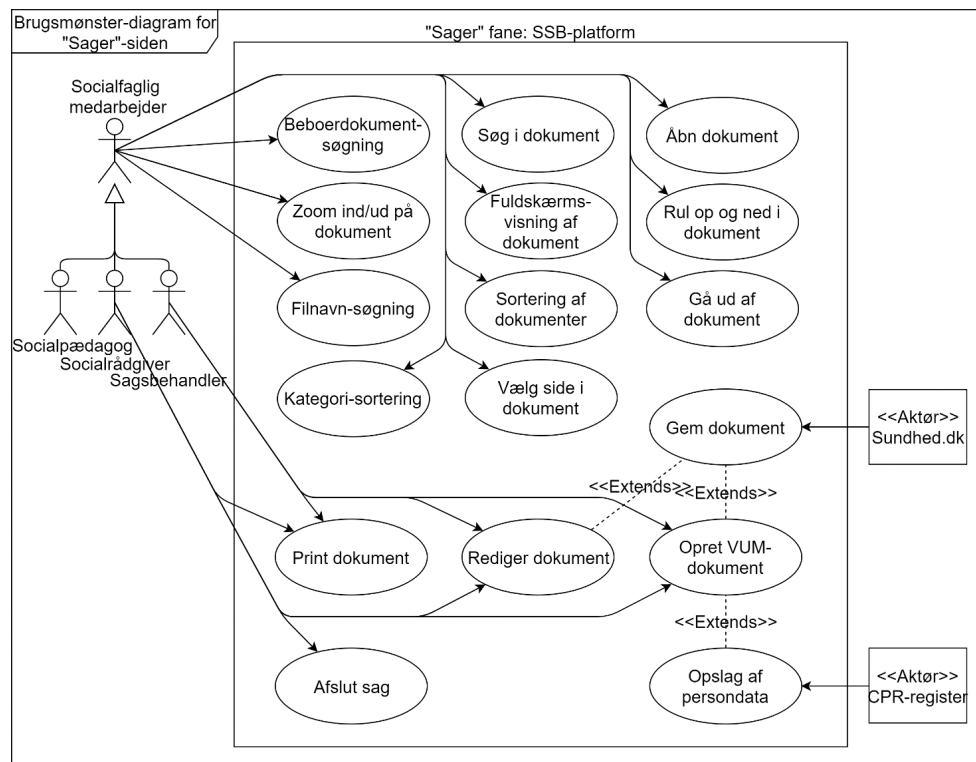
### 9.2.1 Brugsmønsterdiagrammer

Herunder ses brugsmønsterdiagrammerne for systemet SSB. Der er to forskellige. Det første diagram har afgrænsningen "SSB-platform". Det andet diagram har afgrænsningen "Sager-siden". Denne opdeling af brugsmønstre har givet et overblik i modsætning til hvis alle brugsmønstrene var samlet på ét diagram. I projektet skulle gruppen komme med et bud på, hvordan EG Team Online kunne samle sine tjenester under én hat samtidig med udvælge specifikke aspekter af programmet, man ville arbejde på. Det første brugsmønsterdiagram som ses på figur 2, illustrerer hvilke brugsmønstre programmets overordnede platform har. Funktionaliteten fra de individuelle tjenester er derfor ikke illustreret her. I projektet blev der fokuseret på sagsbehandling-delen af Sensum Bosted. Derfor har gruppen lavet et brugsmønsterdiagram for den afdeling af platformen der har at gøre med sagsbehandling. Brugsmønsterdiagrammet kan ses på figur 3.



Figur 2 - brugsmønsterdiagram for platform





Figur 3 - Brugsmønster-diagram for "Sager"-siden.

Som det fremgår af de to brugsmønsterdiagrammer, er der en mængde forskellige aktører. Nogle besidder mange fællestræk som f.eks. socialpædagog, socialrådgiver og sagsbehandler. Derfor er der lavet en aktørgeneralisering, hvor de tre aktører nedarver fra den mere generelle aktør Socialfaglig medarbejder. I tabellen herunder findes en beskrivelse af systemets aktører:

Aktørliste	
Sagsbehandler	Sagsbehandleren sidder ved kommunen og foretager sagsudredninger på beboere. De har blandt andet til opgave at finde ud af, hvordan beboerens behandlingsforløb skal foregå. De dokumenterer udredningsforløbet igennem dagbogsnotater. De skal have adgang til relevant information om sager.
Socialpædagog	En socialpædagog er en person ude på bostederne som hjælper med det praktiske omkring beboerne. En socialpædagog skal have adgang til de relevante dokumenter for de borgere de har at gøre med. De skal ikke have adgang til information, fra andre bosteder. En socialpædagog kan fungere som socialpædagog på ét bosted, og vikar på et andet, hvor de har forskellige rettigheder hvert sted.
Socialrådgiver	En socialrådgiver er en person som hjælper med sagsudredningen for beboere. De befinder sig både ude på bosteder og hos kommunen. De har adgang til information fra deres bosted og arbejdsområde og arbejder med VUM.
Vikar	En vikar skal støtte socialpædagogen, men skal ikke have samme mængde rettigheder.
Systemadministrator	Bruger med eksklusiv adgang til at oprette brugere, give privilegeret adgang til andre brugere.
CPR-registeret	Dette er et eksternt system, som systemet anvender til at få information om beboere, når en ny sag oprettes.



**Sundhed.dk**

Dette eksterne system gemmer data og dokumenter til hver gang nye VUM-dokumenter oprettes og redigeres. Beboere skal have mulighed for at tilgå personlige dokumenter omkring sine sagsudredninger på denne hjemmeside.

### 9.3 Detaljeret kravspecifikation

Hvis man ønsker at fremstille en løsning til et problem der tilfredsstiller kunden, er det essentielt, at kundens ønsker bliver defineret præcist og detaljeret. Denne detaljering foretages i størst grad i inceptionsfasen og kan blandt andet gøres i form af brugsmønstre og krav. I dette afsnit præsenteres kravene til projektgruppens produkt. Detaljerede brugsmønsterbeskrivelser vil fremstilles for at fastsætte de funktionelle krav, og supplerende krav vil beskrives efter FURPS+.

#### 9.3.1 Detaljerede brugsmønsterbeskrivelser

De mest essentielle af de opstillede brugsmønstre er:

- Opret VUM-dokument
- Gem dokument
- Opslag af persondata
- Log ind
- Lav notifikation
- Rediger dokument
- Åbn dokument
- Log ud

Disse brugsmønstre var mest i fokus igennem projektet og vil derfor være præsenteret i detaljer i denne del af rapporten. I første iteration i elaborationsfasen fokuserede gruppen dog kun på den delmængde af brugsmønstrene som var mest essentielle i forhold til sagsbehandlingsdelen af programmet. Det var også et ønske at implementere muligheden for at logge ind og ud af programmet. De essentielle brugsmønstre i første iteration var:

- Opret VUM-dokument
- Gem dokument
- Åbn dokument
- Rediger dokument
- Log ind
- Log ud

De detaljerede beskrivelser af disse brugsmønstre kan ses herunder:

Opret VUM-dokument
<b>ID:</b> 3
<b>Primære aktører:</b> Sagsbehandler, Socialrådgiver
<b>Sekundære aktører:</b> Sundhed.dk, CPR-registeret
<b>Kort beskrivelse:</b> Oprettelse af et bestemt VUM-dokument knyttet til en ny eller eksisterende borger. Brugeren vælger selv hvilket VUM-dokument der skal oprettes og bliver præsenteret for en passende skabelon som skal udfyldes med information, hvor CPR-registeret bl.a. automatisk indhenter information og udfylder den information. Sagsbehandlere må kun oprette sagsåbnings dokumenter, hvorimod socialrådgiver må oprette alle andre VUM-dokumenter end sagsåbnings dokumenter.
<b>Prækonditioner:</b>

1. Brugeren der er logget på, er inde på "sager"-fanen i SSB-plattformen
<b>Hovedhændelsesforløb:</b> <ol style="list-style-type: none"> <li>Brugsmønstret begynder når brugeren klikker på "opret dokument" knappen</li> <li><b>Hvis</b> brugeren har rollen sagsbehandler <ol style="list-style-type: none"> <li>Brugeren kan oprette et nyt sagsåbningsdokument</li> <li>Det første felt i dokumentet skal udfyldes med beboernes CPR-nr.</li> <li>Udvidelsespunkt: Opslag af persondata</li> <li>Specifikke oplysninger om sagen udfyldes af brugeren</li> <li>Brugeren vælger bosted som borgeren skal sendes til <ol style="list-style-type: none"> <li>Brugeren bliver præsenteret med mængden af beboer på det valgte bosted</li> </ol> </li> </ol> </li> <li><b>Hvis</b> brugeren har rollen socialrådgiver <ol style="list-style-type: none"> <li>Brugeren får mulighed for at oprette et af de 9 VUM redskabs-dokumenter med undtagelse af sagsåbnings dokument</li> <li>Brugeren vælger dokument der skal oprettes</li> <li>Brugeren ser en oversigt over beboere de er associeret med som skal tilknyttes det nye dokument</li> <li>Allerede eksisterende data bliver automatisk udfyldt i det nye dokument. (Dette indbefatter navn, adresse, CPR-nummer osv.)</li> </ol> </li> <li>Det for-udfyldte dokument åbner op hvor brugeren skal udfylde relevant information om beboeren</li> <li>Udvidelsespunkt: Gem dokument</li> <li>Systemet giver en bekræftelse på at dokumentet er blevet gemt succesfuldt</li> </ol>
<b>Post-konditioner:</b> <ol style="list-style-type: none"> <li>Der er oprettet et VUM-dokument af den type som brugeren har valgt</li> <li>VUM-dokumentet kan ses i oversigten og alle dokumenter</li> <li>Hvis der blev oprettet en ny beboer, skal denne beboer være tilknyttet til de rette socialpædagoger og socialrådgivere</li> </ol>
<b>Alternative håndelsesforløb:</b> <ul style="list-style-type: none"> <li>Overalt: Annullering</li> </ul>

<b>Udvidelsesbrugsmønster: Gem dokument</b>
<b>ID:</b> 8
<b>Primære aktører:</b> Sagsbehandler, socialrådgiver
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Efter udfyldelse af VUM-dokument skal det gemmes i en database og vises i dokumentoversigten for alle brugere af systemet
<b>Prækonditioner:</b> <ol style="list-style-type: none"> <li>Brugeren er i gang med at oprette eller redigere et VUM-dokument</li> </ol>
<b>Hovedhændelsesforløb:</b> <ol style="list-style-type: none"> <li>Brugsmønsteret starter når brugeren vil gemme et dokument</li> <li>Systemet tjekker for om alle de nødvendige felter i dokumentet er blevet udfyldt korrekt <ol style="list-style-type: none"> <li><b>Hvis</b> felterne ikke er udfyldt korrekt, får brugeren en notits om at de skal udfylde dem korrekt</li> </ol> </li> <li>Systemet sender alt information ind til en database</li> <li>Systemet lukker dokumentet som brugeren havde åbent</li> <li>Systemet giver brugeren besked på at det er gemt korrekt</li> </ol>
<b>Post-konditioner:</b> <ol style="list-style-type: none"> <li>Dokumentet kan ses i oversigten</li> <li>Dokumentet kan hentes fra databasen</li> </ol>

<b>Alternative hændelsesforløb:</b> Ingen
--

<b>Alternative flow: Annullering (opret/rediger dokument)</b>
<b>ID:</b> 3.1
<b>Primære aktører:</b> Sagsbehandler, socialrådgiver
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Igennem hele brugsmønsteret "Opret VUM-dokument" er det muligt at annullere oprettelsen af en ny sag.
<b>Prækonditioner:</b> 1. Et VUM-Dokument er blevet oprettet
<b>Hovedhændelsesforløb:</b> 1. Brugsmønsteret starter når brugeren trykker på knappen "Annuller" ved enten oprettelse eller redigering af et dokument. 2. Systemet spørger brugeren om de vil bekræfte annulleringen 3. <b>Hvis</b> brugeren bekræfter annulleringen 3.1. Alle ændringer og nyoprettede dokumenter kasseres 3.2. Brugeren sendes tilbage til "sager"-siden 4. <b>Hvis</b> brugeren annullerer annulleringen 4.1. Brugeren sendes tilbage til dokumentet og kan redigere videre
<b>Alternativt flow:</b> Ingen
<b>Post-konditioner:</b> 1. Indtastede oplysninger fra VUM-dokumentet er blevet slettet

<b>Åbn Dokument</b>
<b>ID:</b> 6
<b>Primære aktører:</b> Sagsbehandler, Socialpædagog, Socialrådgiver
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Brugere med adgang til et dokument er i stand til at åbne det. Efter åbning vil dokumentets indhold vises for brugerne.
<b>Prækonditioner:</b> 1. Et dokument er oprettet 2. Brugeren er tilknyttet samme bosted som dokumentet er oprettet på.
<b>Hovedhændelsesforløb:</b> 1. Brugsmønsteret starter når brugeren klikker på et dokument fra deres liste af tilgængelige dokumenter. 2. Dokumentets indhold hentes fra databasen 3. Indholdet af dokumentet vises for brugeren
<b>Alternativt flow:</b> Ingen
<b>Post-konditioner:</b> 1. Dokumentet er åbent

Rediger dokument
<b>ID:</b> 7
<b>Primære aktører:</b> Sagsbehandler, Socialrådgiver
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Sagsbehandlere eller socialrådgivere med adgang til et dokument har lov til at redigere dokumentets indhold efter det er åbnet.
<b>Prækonditioner:</b> <ol style="list-style-type: none"> <li>1. Et dokument er åbnet</li> <li>2. Brugeren er af rollen sagsbehandler eller socialrådgiver</li> </ol>
<b>Hovedhændelsesforløb:</b> <ol style="list-style-type: none"> <li>1. Brugsmønstret starter når brugeren har åbnet et dokument (se brugsmønsterbeskrivelse "Åbn Dokument")</li> <li>2. Brugeren foretager eventuelt ændringer til dokumentets indhold.</li> <li>3. Udvidelsespunkt: Gem dokument</li> <li>4. Systemet giver en bekræftelse på at dokumentet er blevet gemt succesfuldt</li> </ol>
<b>Alternativt flow:</b> <ul style="list-style-type: none"> <li>• Overalt: Annullering</li> </ul>
<b>Post-konditioner:</b> <ol style="list-style-type: none"> <li>1. Det redigerede dokument er gemt i databasen med det nye indhold.</li> <li>2. Brugeren befinder sig på dokumentoversigtssiden.</li> </ol>

Log ind
<b>ID:</b> 1
<b>Primære aktører:</b> Alle
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Når en bruger skal tilgå systemet, skal man anvende sit brugernavn og adgangskode. Ved at logge ind får brugeren adgang til systemets funktioner, som f.eks. sagsbehandling. De kan oprette, se og redigere dokumenter. Mængden af data brugeren har adgang til at se, afhænger af brugerens tildelte rolle.
<b>Prækonditioner:</b> <ol style="list-style-type: none"> <li>1. Systemet er på log ind-skærmen</li> <li>2. Brugerens login skal eksistere i systemets database</li> </ol>
<b>Hovedhændelsesforløb:</b> <ol style="list-style-type: none"> <li>1. Brugsmønsteret starter når brugeren åbner systemet</li> <li>2. Systemet spørger brugeren om brugernavn og adgangskode, som brugeren indtaster</li> <li>3. Brugeren trykker på log ind-knappen</li> <li>4. Brugeren bliver logget ind i systemet, og har adgang til relevante data og funktioner</li> </ol>
<b>Alternativt flow:</b> <ol style="list-style-type: none"> <li>1. Forkert Login</li> </ol>
<b>Post-konditioner:</b> <ol style="list-style-type: none"> <li>1. Brugeren kan navigere videre i systemet og har adgang til relevant data.</li> </ol>

Alternativt flow: Forkert Login
<b>ID:</b> 1.1
<b>Primære aktører:</b>

Alle
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Ved indtastning af forkert information bliver brugerens adgang nægtet til systemet.
<b>Prækonditioner:</b> 1. Brugeren har fejlet login ved indtastning af forkert information
<b>Hovedhændelsesforløb:</b> 1. Brugeren indtaster forkert login-information 2. Brugeren får besked på at indtastet information ikke stemmer overens 3. Brugeren får chancen for at prøve igen
<b>Alternativt flow:</b> Ingen
<b>Post-konditioner:</b> Ingen

I slutningen af første iteration efter at have fået skabt kernefunktionerne af systemet, kunne de resterende brugsmønstre komme i fokus. De mest essentielle af disse er følgende:

- Opslag af persondata
- Lav notifikation

Disse brugsmønstre er beskrevet i detaljer herunder:

<b>Udvidelsesbrugsmønster: Opslag af persondata</b>
<b>ID:</b> 9
<b>Primære aktører:</b> Sagsbehandler
<b>Sekundære aktører:</b> CPR-register
<b>Kort beskrivelse:</b> Under oprettelse af en ny sag til en ny beboer, skal sagsbehandleren kunne hente information om den pågældende beboer fra CPR-registeret ud fra beboerens CPR-nummer.
<b>Prækonditioner:</b> 1. Brugeren er en sagsbehandler 2. Brugeren er i gang med at udfylde information om den nye beboer 3. Et CPR-nummer er indtastet i CPR-register-søgefeltet
<b>Hovedhændelsesforløb:</b> 1. Brugsmønsteret starter når sagsbehandleren trykker ”søg” med det indtastede CPR-nummer ved oprettelse af en ny beboer i forbindelse med en sagsåbning. 2. Det indtastede CPR-nummer sendes til CPR-registeret 3. CPR-registeret sender den relevante information om beboeren tilbage til systemet 4. De hentede informationer fra CPR-registeret bliver udfyldt i dokumentet.
<b>Post-konditioner:</b> 1. Felter i dokumentet er udfyldt med alt relevant information fra CPR-registeret
<b>Alternative hændelsesforløb:</b> 1. CPR-nummer ikke fundet i register

Alternativt flow: CPR-nummer ikke fundet i register
<b>ID:</b> 9.1
<b>Primære aktører:</b> Sagsbehandler
<b>Sekundære aktører:</b> CPR-register
<b>Kort beskrivelse:</b> Ved indtastning af et CPR-nummer der ikke findes, melder systemet fejl.
<b>Prækonditioner:</b> 1. Et ugyldigt CPR-nummer er indtastet i CPR-register-søgefeltet
<b>Hovedhændelsesforløb:</b> 1. Brugsmønsteret starter når systemet ikke modtager noget information fra CPR-registeret. 2. Systemet viser brugeren en fejlmeddelelse og tillader brugeren at prøve igen.
<b>Alternativt flow:</b> Ingen
<b>Post-konditioner:</b> Ingen

Lav notifikation
<b>ID:</b> 10
<b>Primære aktører:</b> Socialfaglig medarbejder, vikar
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> De ansatte på bostederne kan anvende systemet til at skrive notifikationer til hinanden. Det skal fungere som en nem og hurtig måde at dele information på mellem de ansatte på et bosted.
<b>Prækonditioner:</b> 1. Brugeren er af typen socialpædagog eller vikar. 2. Brugerne befinder sig på notifikationssiden.
<b>Hovedhændelsesforløb:</b> 1. Brugsmønsteret starter når brugeren indtaster den tekst som ønskes indeholdt i notifikationen i et tekstfelt på notifikationssiden. 2. Brugeren trykker på "send"-knappen. 3. Systemet gemmer notifikationens indhold, forfatter og oprettelsestidspunkt, og viser den på skærmen.
<b>Alternativt flow:</b> Ingen
<b>Post-konditioner:</b> 1. Notifikationen er gemt i systemet og kan ses tilgås af alle på det pågældende bosted.

### 9.3.2 Supplerende krav

Igennem FURPS+ blev følgende supplerende krav fundet:

ID	Kravbeskrivelse
F00	SSB skal logføre alle aktiviteter
F01	SSB skal være kompatibelt med Windows 7, Windows 8 & Windows 10
F02	SSB skal kunne udveksle data med CPR-registeret
F03	SSB skal kunne sende information til Sundhed.dk
F04	SSB skal fordele brugere på specifikke roller og ID for at afgrænse adgang til data
U00	SSB skal have en tilfredsstillende vurdering fra testbrugere
U01	SSB skal give brugeren mulighed for at gå tilbage til systemets forrige grafiske tilstand.
U02	SSB skal have en responstid på op til 1 sekund
U03	SSB skal synliggøre hvilke brugere, som har oprettet, redigeret og slettet dokumenter.
U04	SSB skal have ensartet placering af gentagende knapper på alle sider
U05	SSB skal give brugeren adgang til navigationsbaren til venstre uanset hvor brugeren befinder sig
U06	SSB skal vise en dialogboks i tilfælde af at brugeren går væk fra arbejde der ikke er gemt
U07	SSB skal give brugeren besked på manglende udfyldte felter i tilfælde af de prøver at gemme et dokument der ikke er udfyldt i alle nødvendige felter
U08	SSB skal kunne være brugbart i forskellige størrelser (skalerbar)
R00	SSB skal stå til rådighed 97% af tiden
R01	SSB skal have en opdateringstid på max 2 timer
P00	SSB skal kunne håndtere 1750 samtidige logins
P01	SSB skal kunne håndtere 20GB data fra VUM-dokumenter
P02	SSB skal muliggøre at hente og sende af data på mindst 10 MB/sekundet
P03	SSB skal max bruge op til 2GB ram
S00	SSB skal indeles i en plug and play samlekasse med moduler

De krav der var mest essentielle blandt ovenstående, var blandt andet dem der havde med datasikkerhed at gøre. Det supplerende krav **F04** siger ”SSB skal fordele brugere på specifikke roller og ID for at afgrænse adgang til data”. Alle brugere skulle ikke have samme rettigheder. F.eks. skal en socialpædagog være i stand til at læse sagsbehandlingsdokumenter, men de skal ikke kunne oprette dem. Derfor blev systemet nødt til at have et rollesystem som bestemmer hvad de forskellige brugere skal være i stand til. Derudover skal alle brugere ikke have adgang til samme data. F.eks. må en ansat på ét bosted ikke kunne læse sagsbehandlingsdokumenter for en person på et andet bosted. På grund af vigtigheden af dette, blev kravet prioriteret højt.

Et krav som **U03** var også meget vigtigt i forhold til systemets sikkerhed/tryghed. Hvis det opnås, kan man altid holde folk til ansvar for deres handlinger i systemet, hvilket forhindrer ondsindede gerninger. **U06** som siger, at brugeren skal advares, hvis man lukker et dokument med ikke-gemte ændringer, vil være med til at forhindre utilsigtet tab af data.

Et andet supplerende krav til dette projekt var, at systemet skulle kobles op på en database i anden iteration. Kravet var defineret i projektbeskrivelsen, som grupperne fik udleveret i starten af projektet. Databasen skulle stå for at opbevare al data i systemet. Dette inkluderede f.eks. ansatte, beboere, sagsbehandlingsdokumenter og notifikationer.

### 9.3.3 Sprint backlog

De detaljerede brugsmønstre og de supplerende krav blev udarbejdet i løbet af inceptionsfasen. Denne detaljering af hvad systemet skulle være i stand til, gjorde det muligt at starte på elaborationsfasen. Fasen startede med at udvikle en product backlog som indeholdt alle krav og al funktionalitet som systemet gerne skulle opnå i slutningen af projektet. I starten af første sprint udvalgte udviklingsholdet en delmængde af kravene fra

product backlog til første sprint backlog. Denne indeholdt alle krav og al funktionalitet som udviklingsholdet mente skulle udvikles i løbet af første sprint. Tabellen herunder viser første sprint backlog. Her ses kravene samt deres prioritering. Løbende kunne udviklingsholdet også opdatere status for kravenes færdiggørelsesgrad.

Række-følge	Krav	Status	MoSCow
0	Adgang til navigationsbar uanset hvor brugeren befinder sig	Ikke startet	M
0	Sortering af dokumenter i dokumentoversigt	Ikke startet	S
0	Se dokumentoversigt	Ikke startet	M
0	Skalerbart interface	Ikke startet	M
1	Fordele brugere på specifikke roller og id	Ikke startet	M
2	Opret VUM-dokument	Ikke startet	M
3	Åben dokument	Ikke startet	M
4	Rediger dokument	Ikke startet	M
5	Gemme dokument	Ikke startet	M
6	Gå ud af dokument	Ikke startet	M
7	Rul op og ned i dokument	Ikke startet	M
8	Notits af manglende udfyldte felter ved gemning af dokument	Ikke startet	M
9	Log ind	Ikke startet	M
10	Log ud	Ikke startet	M
11	Afslut sag	Ikke startet	M
12	Beboer dokument søgning	Ikke startet	M
13	Filnavn søgning	Ikke startet	M
14	Åben faner	Ikke startet	M
15	Mulighed for at gå tilbage til systemets grafiske tilstand	Ikke startet	S
16	Opslag af persondata via CPR-nr.	Ikke startet	S
17	Ensartet placering af UI-elementer	Ikke startet	S
18	Søg eller skriv en kommando	Ikke startet	C
19	Synliggør brugere der har oprettet eller redigeret dokument	Ikke startet	C
20	Printe VUM-dokument	Ikke startet	C
21	Tilkobl database til opbevaring af data	Ikke startet	M

Ved slutningen af første iteration, ca. halvvejs igennem elaborationsfasen, var en mængde af sprint backloggen opnået. I starten af 2. iteration blev den nye sprint backlogs indhold sammensat fra dels nye krav og dels krav fra første sprint backlog. Backloggen for anden iteration kan ses på tabellen herunder:



Række-følge	Krav	Status	MoSCow
1	Gå tilbage til startskærm	Ikke startet	M
2	Lav notifikation	Ikke startet	S
3	Se notifikation på startskærm	Ikke startet	S
4	Opret bruger	Ikke startet	M
5	Slet bruger	Ikke startet	M
6	Redigér Bruger	Ikke startet	S
7	Ændre VUM-dokument-implementering	Ikke startet	M
8	Ændre database til at køre på bosteder i stedet	Ikke startet	M
9	Printe Dokument pdf	Ikke startet	W
9	Åbn Dokument Word	Ikke startet	W
10	Slette beboere	Ikke startet	M
10	Få database til at køre på threading	Ikke startet	S
11	Fikse dokumentåbning	Ikke startet	S
12	Afslut sag	Ikke startet	W
13	Simulation af SSB skal kunne udveksle data med CPR-registeret	Ikke startet	S
14	Simulation af SSB skal kunne sende information til Sundhed.dk	Ikke startet	C
15	Beboer dokument søgning	Ikke startet	C
16	Filnavn-søgning	Ikke startet	C
17	Kategori-filtrering	Ikke startet	C
18	Automatisk log ud ved inaktivitet	Ikke startet	S
19	Søg, eller skriv en kommando	Ikke startet	C
21	Søg i dokument	Ikke startet	C
22	Fikse trelagsarkitektur	Ikke startet	S

Med de detaljerede beskrivelser i slutningen af inceptionsfasen var udviklingsholdet nu i stand til at fremstille en backlog til det kommende sprint. At kravene var beskrevet detaljeret sikrede, at udviklingsholdet vidste helt præcist hvad det endelige produkt skulle være i stand til, før løsningen var tilfredsstillende. Dette førte videre til analysedelen, hvor gruppen kiggede på hvordan systemet skulle hænge sammen på et højt niveau.

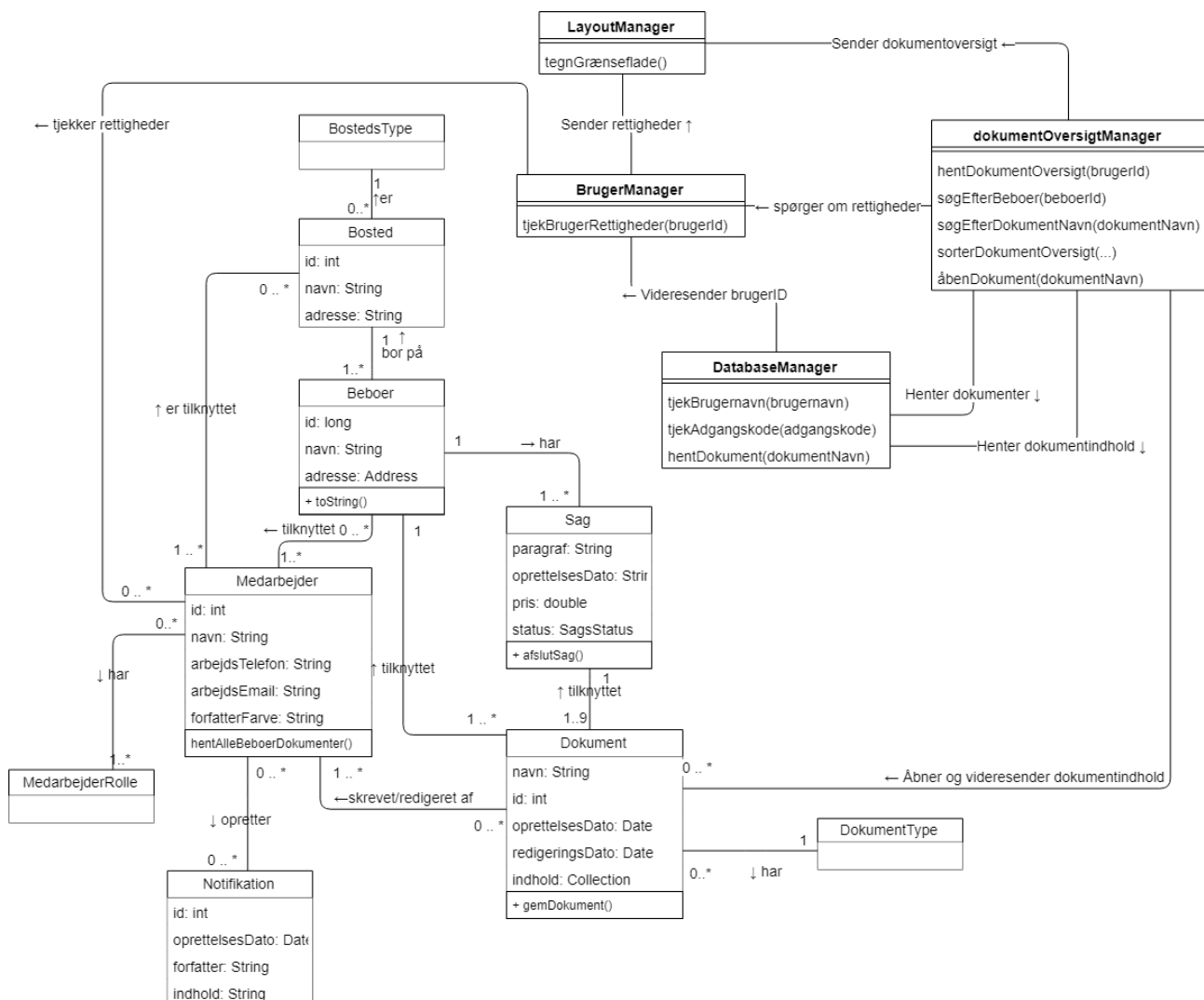
## 9.4 Analyse

I inceptionsfasen i Unified Process bliver analysen opstartet med UML-modellering af krav og forretningsbegreber. Når elaborationsfasen derefter går i gang bliver analysemodellerne, både statiske og dynamiske modeller, ordentligt modelleret. Når Unified Process kombineres med en agil metode, er det vigtigt at udføre opgaver i parallel. Derfor er den dynamiske side af analysemodellen, specielt i dette projekt, blevet fremstillet i takt med at features fra Sprint backloggen skulle implementeres. Denne arbejdsmetode skyldes at det krævede en stor mængde arbejde at udarbejde hele analysen før implementering og det var en meget statisk arbejds metode som mindede meget om vandfaldsmodellen. Gruppen vurderede derfor at tiden ville blive bedre brugt på at lave analyse og design til hver opgave i Sprint backloggen i takt med at de skulle implementeres. Hvert delafsnit vil indeholde resultater både fra første og anden iteration.

### 9.4.1 Statiske side af analysemodellen

#### 9.4.1.1 Første iteration

På nedenstående figur ses analysemodellen fra første iteration, som blev modelleret før implementering.



Figur 4 - Analysemodel 1. iteration

**Dokumenter:**

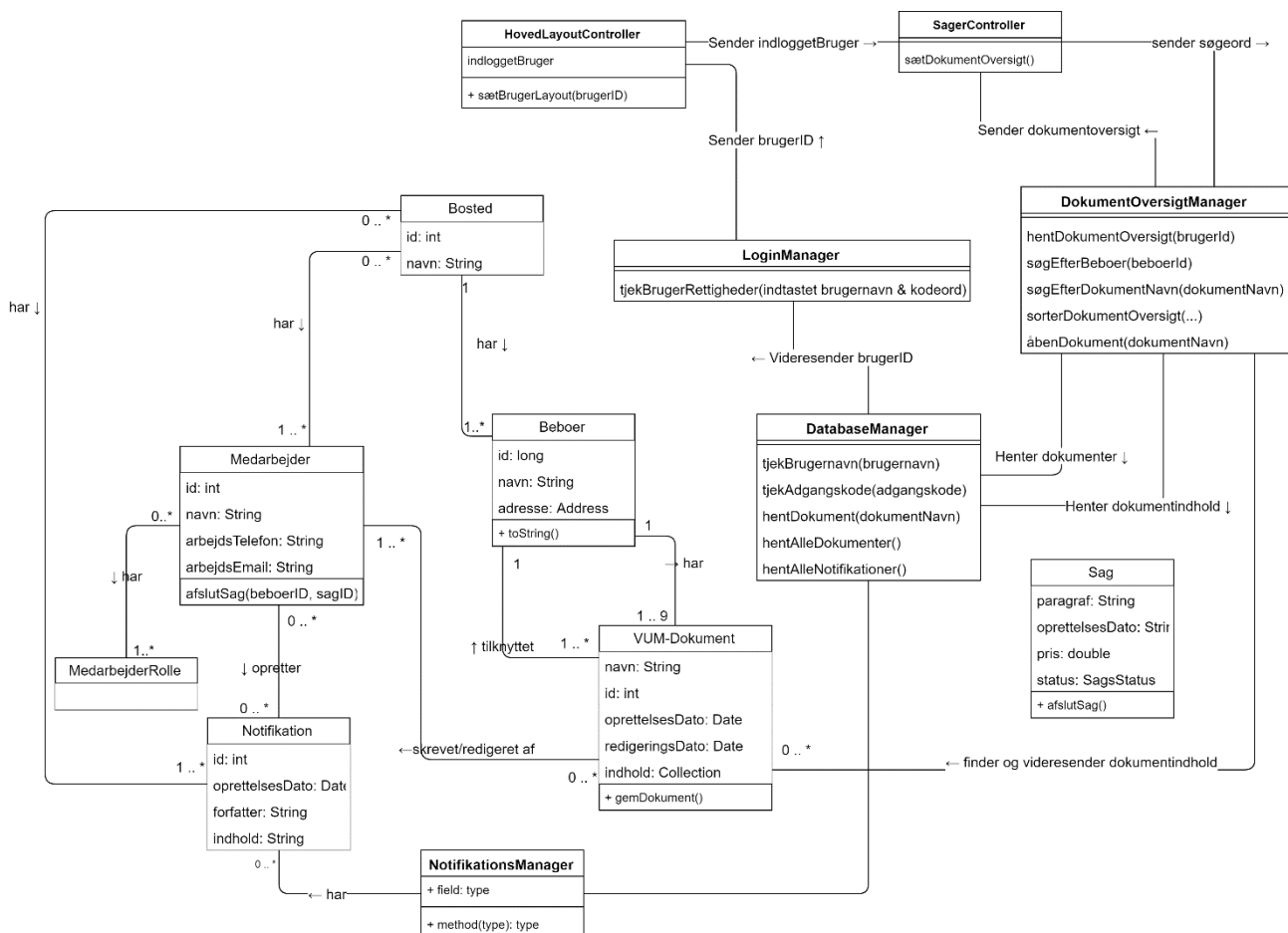
I første iteration blev en medarbejder tilknyttet en specifik beboer. Hvor hver beboer havde en sag, og hver sag havde VUM-Dokumenter, som kunne have forskellige dokumenttyper. Disse dokumenter blev fremvist for medarbejderen i en oversigt, som DokumentOversigtManager i samarbejde med DatabaseManager henter og fremviser fra en database. Medarbejderen kan kun se dokumenter som er tilknyttet de beboere som medarbejderen er tilknyttet.

**Medarbejderrettigheder:**

I første iteration var det tiltænkt at en medarbejder havde en rolle som bestemte generelle ting såsom om medarbejderen havde rettigheder til at åbne et VUM-dokument, opret et nyt VUM-dokument, skrive en notifikation og lignende. Derudover havde en medarbejder også specifikke tilknyttede ting såsom dokumenter, dataafgrænsningen gjorde at de kun kan se dokumenter pga. deres tilknytning til beboeren.

**9.4.1.2 Anden iteration**

I anden iteration blev håndteringen af dokumenter og implementeringen af medarbejderens rettigheder simplificeret. På nedenstående figur ses et opdateret UML-analyseklassediagram.



Figur 5- Analysemodel 2. iteration

**Beboere har dokumenter og et bosted har beboere**

Dokumentimplementeringen er skiftet fra at være mellem en medarbejder og en beboer, til at være mellem et bosted og en beboer. Bosteder har derfor tilknyttede beboere og medarbejdere, hvor medarbejdere kan logge ind og automatisk få relevante information for præcis det bosted som de er tilknyttet. Den relevante information som medarbejderen får vist, inkluderer notifikationer, dokumenter og beboere som kun tilhører et specifikt

bosted. data-afgrænsningen er derfor ændret, til at afgrænse pr. bosted i stedet for pr. medarbejder, da det giver bedre workflow for medarbejderen.

Denne ændring skyldes at det blev besværligt at tilknytte en beboer til en medarbejder, i tilfælde af at en medarbejder bliver syg, siger op eller lignende. Det var derfor nemmere hvis alle medarbejdere tilknyttet et bosted kan se alle dokumenter for alle beboere på det pågældende bosted. Det støtter også at medarbejdere kan hjælpe hinanden med forskellige beboere på bostedet.

### **Beboer og dokumentsøgning**

Eftersom alle medarbejder på et bosted kan se alle dokumenter for alle beboere, var det vigtigt at få indført en søgefunktion, så medarbejderen hurtigt kan finde det dokument de leder efter. Derfor indførtes en forbindelse mellem `sagerController` og `DokumentOversigtManager` hvor `sagerController` kan søge via beboerens navn eller dokumentets navn baseret på brugers input som derefter justerer dokumentoversigten som medarbejderen kigger på.

### **Ændringer i analysemodellen**

Udover tilføjslen af de ovenstående punkter, blev analysemodellen også rettet til.

- Layoutmanager blev fjernet og opdelt til flere controllere i stedet for da det var mere realistisk i forhold til hvad vi kom frem til i første iteration
- BrugerManager blev refaktoreret til LoginManager
- Bostedstype blev fjernet da det ikke havde nogen relevans
- Forfatterfarve for medarbejder fjernet
- Hent alle dokumenter metode på medarbejder fjernet og erstattet med `"sætDokumentOversigt()"` i `SagerController`
- Afslutsag tilføjet til medarbejder
- Sag blev frakoblet andre forbindelser, da det var usikkert hvordan vi ville håndtere det på nuværende stadie
- Notifikationsmanager tilføjet som har alle notifikationer. Databasemanager har mulighed for at hente notifikationer

## **9.4.2 Dynamiske side af analysemodellen**

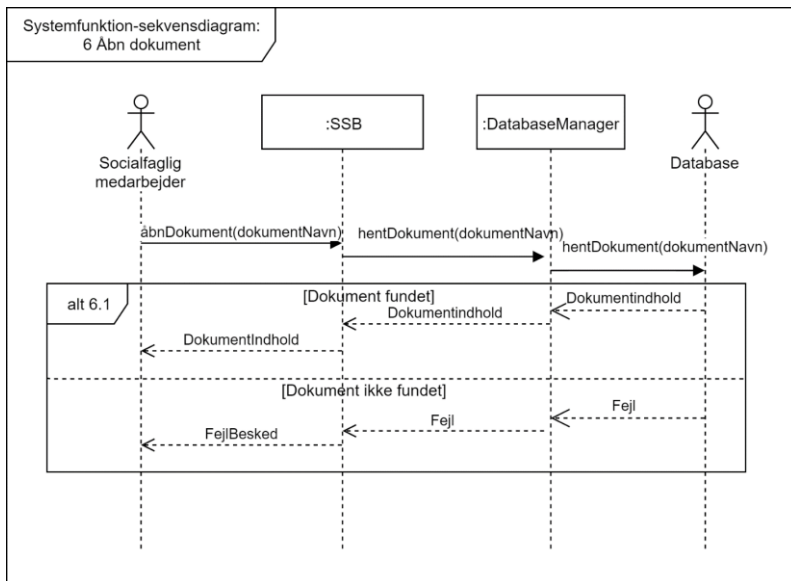
I takt med at features skulle implementeres undervejs i elaborationsfasen i UP blev den dynamiske side af analysemodellen udarbejdet i form af sekvensdiagrammer, som havde til formål at modellere programmets sekvens flow og separationen af ansvar mellem klasser.

### **9.4.2.1 Første iteration**

Fra første iteration blev disse sekvensdiagrammer udarbejdet:

#### **Åbn Dokument**

Dette sekvensdiagram kan ses på figur 6. Dette dokument blev udarbejdet før påbegyndelse af implementering, og er derfor ikke detaljeret i forhold til senere sekvensdiagrammer, specielt i forhold til anden iteration sekvensdiagrammer.

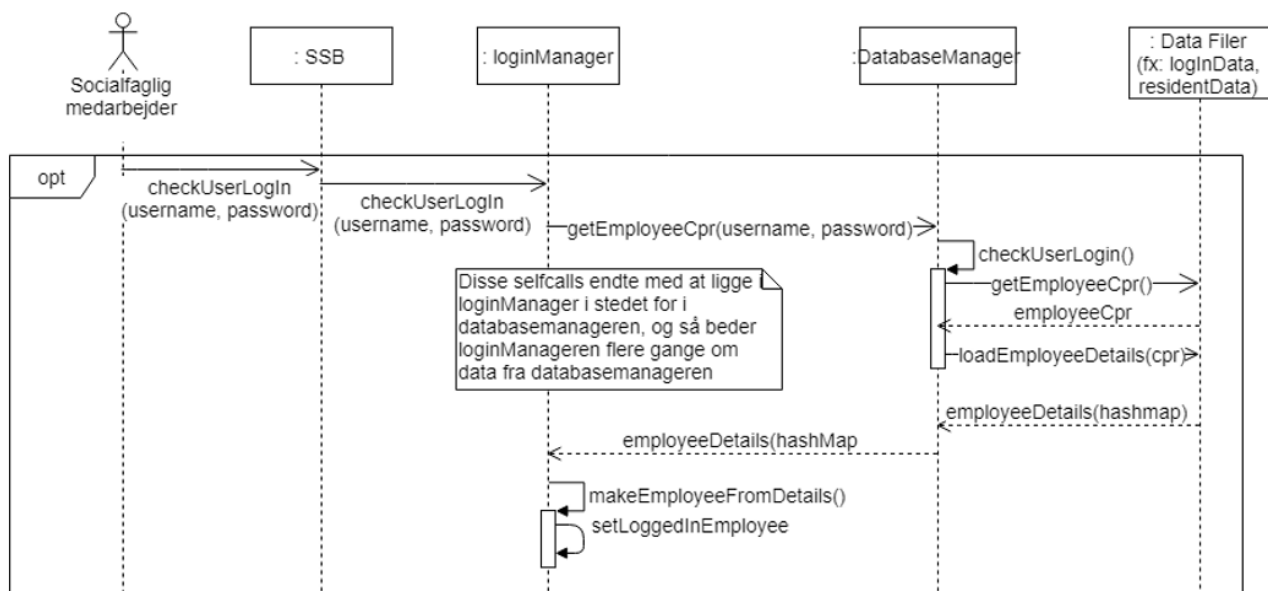


Figur 6 - sekvensdiagram over åben dokument

Når en medarbejder skal åbne et dokument, vil det blive hentet fra databasen og hvis det er fundet, bliver det returneret op igennem lagene tilbage til medarbejderen og ellers vil en fejlbesked blive sendt tilbage. Dette flow viste sig at være et dårligt flow, da der skal undgås kontakt med databasen så meget som muligt, da det tager væsentligt længere tid end at have dokumenterne gemt i en datastruktur i det kørende program.

### Login-sekvens

Til at starte med blev der designet et login sekvens som ikke var helt korrekt, men var fint til at komme i gang med at implementere login. Senere blev et nyt sekvensdiagram for login designet, som kan ses på figur 7.

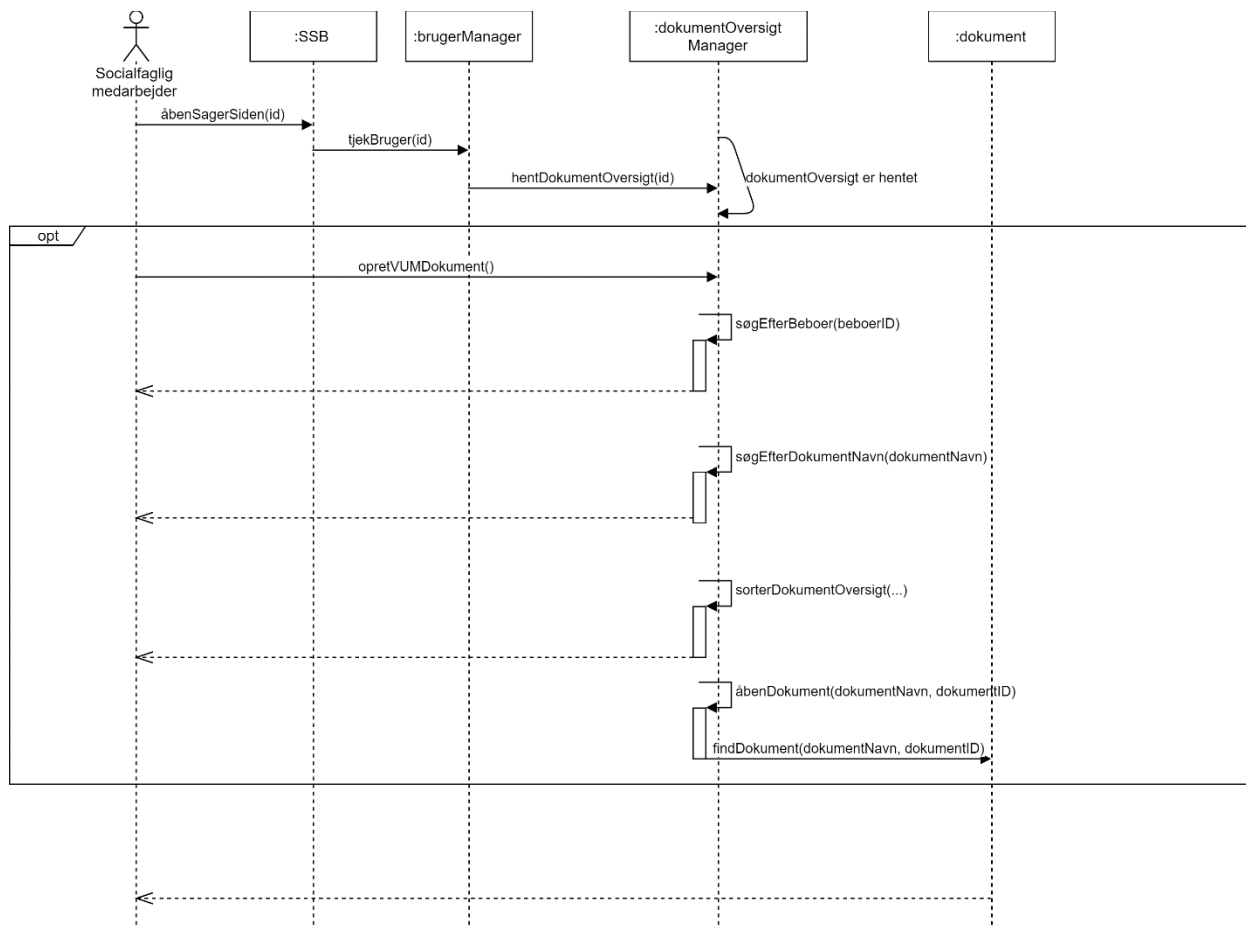


Figur 7- sekvensdiagram over login

Dette sekvensdiagram viste sig at være mere korrekt i forhold til tre lags arkitektur, da medarbejdernes interaktioner skal registreres af systemet selv (SSB) igennem forskellige controllere, hvorefter controlleren skal snakke sammen med en loginManager, der skal befinde sig i domain laget. Herefter samarbejder loginManageren med DatabaseManageren som befinder sig i data laget, og er samtidig den eneste klasse i datalaget som

domænelaget har kontakt med. I sekvensdiagrammet findes også den sidste livslinje, som hedder Data filer (fx: loginData, residentData) osv. Denne livslinje skal repræsentere en samling af klasser som står for at kontakte databasen, f.eks for at query i forhold til username og passwords.

### Dokument-oversigt

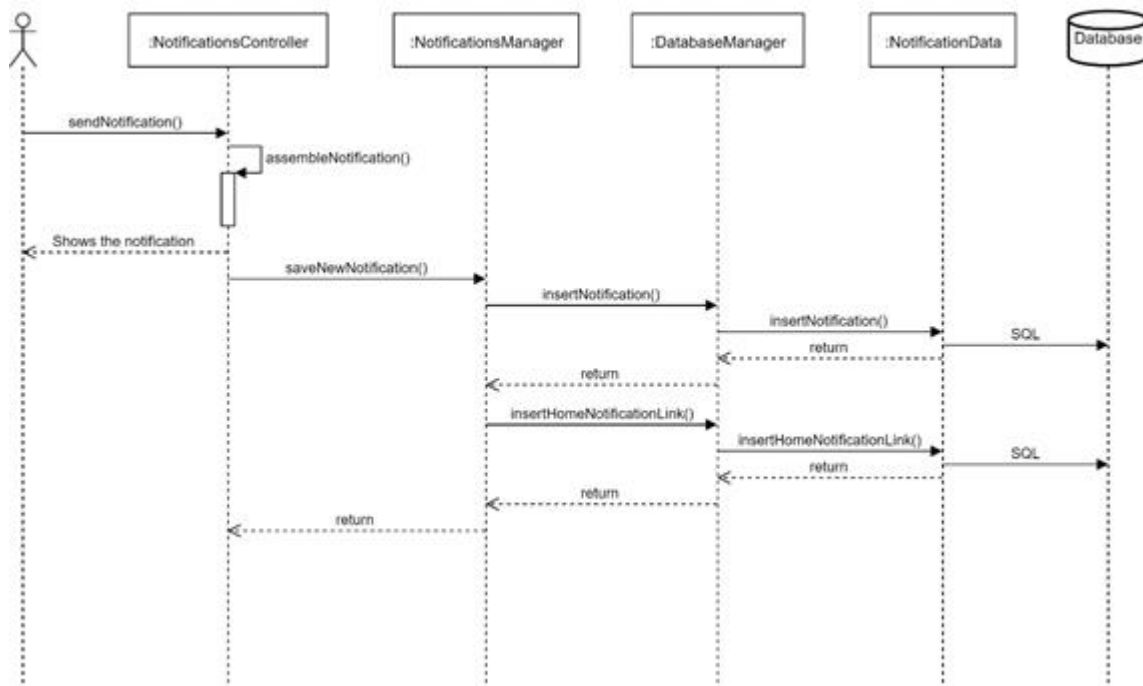


Figur 8 - sekvensdiagram over åben sager siden

På figur 8 ses sekvensdiagrammet for åbne sager, hvor alle interaktioner som en medarbejder har med systemet når de befinder sig i dokumentoversigten. Først bliver medarbejderen verificeret i forhold til deres rettigheder, hvor det bliver vurderet om de har rettigheder til at oprette et nyt VUM-dokument

#### 9.4.2.2 Dynamiske side af analysemodellen

For at lave brugsmønsterrealisering var gruppen nødsaget til at fokusere på operationssekvensdiagrammer, og derfor ikke lave systemsekvensdiagrammer, kontrakter eller opdatere vores analysemodel efterfølgende. Gruppen vurderede at det ikke gav nogle fordele, og der var i forvejen tidspres for at nå tingene. Derfor består brugsmønsterrealisering kun af operations sekvensdiagrammer. Derudover blev sekvensdiagrammerne ikke alle lavet på samme tid, men rettere efterhånden som opgaverne blev startet igennem iterationen.

**Brugsmønster realisering af notifikationer**

Figur 9 - Sekvensdiagram over notifikation

På figur 9 ses sekvensdiagrammet for notifikationer. Når et bosted er blevet sat for den pågældende medarbejder, vil notifikationer begynde at blive loadet, hvor notifikationscontroller og manager arbejder sammen for at hente notifikationer i databasen. Databasen sender derfor notifikationerne tilbage i en datastruktur til notificationsManageren. Så snart notifikationerne er klar vil notifikationernes layout blive vist og sat ind af notificationsControlleren i samarbejde med notificationsManageren.

## 9.5 Design

Den færdige og komplette designmodel indeholder ideelt implementeringsdetaljer, og kan omdannes direkte til kode uden at programmøren selv skal udtænke detaljerne. Designmodellen indeholder blandt andet designklassediagrammer. Disse indeholder information om klassernes attributter, operationer og associationer.

Hver klasse i designklassediagrammet får givet de relevante operationer der hører til. Her er information om navn, returtype og parametre.

### 9.5.1 Softwarearkitektur

I anden iteration begyndte gruppen at benytte sig af managere: notifikationsmanager, databasemanager, CPR-RegisterManager, dokumentmanager. Disse managere blev brugt for at kontrollere flowet i systemet og for at reducere kobling mellem klasser og lag.

Derudover er det forsøgt at undgå oprettelsen af instanser i andre lag end det tilhørende lag. For eksempel sender data laget ikke domæne klasser tilbage og sender i stedet datastrukturer som indeholder den data som der ligger i databasen. Domæne laget sørger så for at omdanne dataene til domæne klasser. Derudover indeholder domæne laget ingen forbindelser fra view laget og datalaget indeholder ingen forbindelser til domæne-laget.

### 9.5.2 Subsystemdesign

Systemet er designet med en trelagsarkitektur hvor hvert lag er uafhængig af lag der er ovenpå. Derudover er GUI laget designet med et subsystem i tankerne. Menuer for oven og til venstre er statiske og kan ikke blive ændret. Layoutet i midten er designet til at kunne blive skiftet ud afhængig af brugerens ønsker. Hele layoutet bliver derfor aldrig genindlæst. Kun det centrale indhold på siden bliver genindlæst når brugeren for eksempel går fra fane til fane ude i den venstre menu, eller når de går hjem og indlæser notifikationer siden.

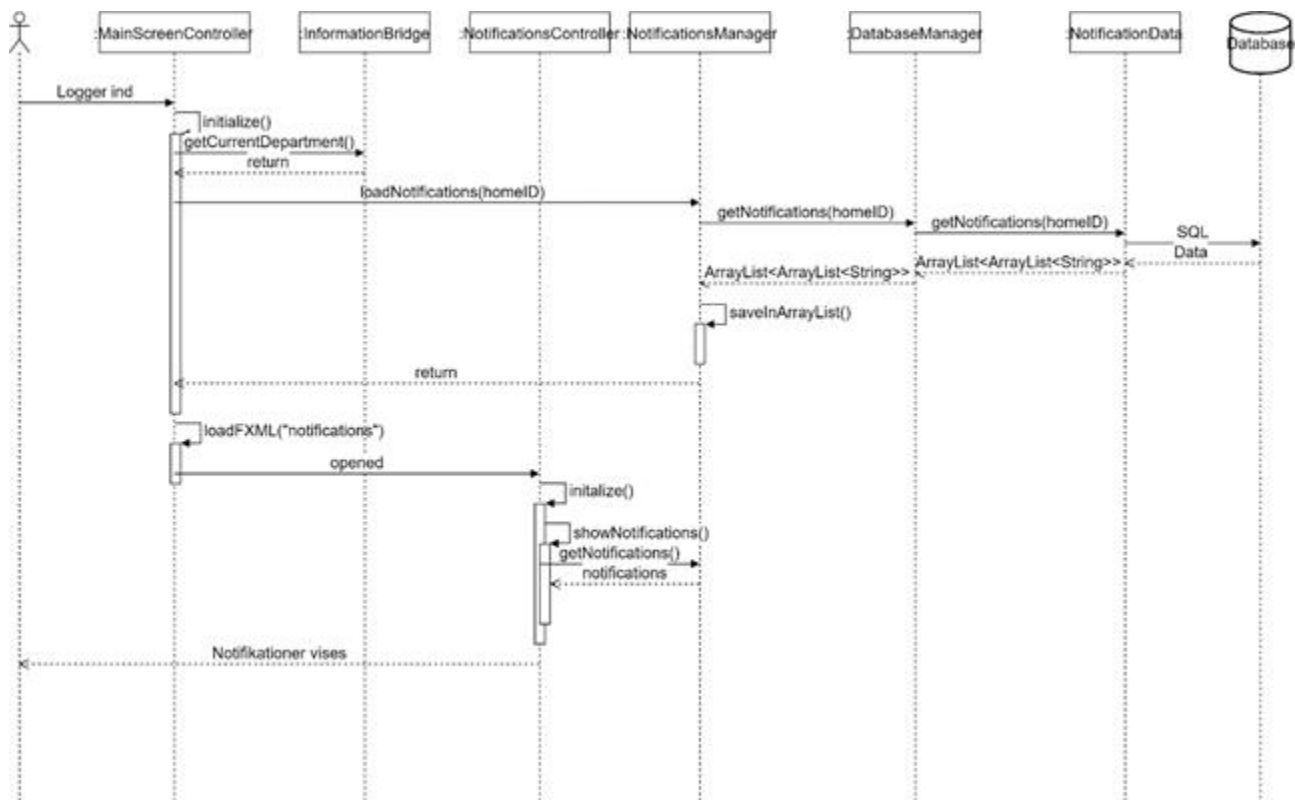
### 9.5.3 Statiske side af designmodellen

I dette projekt er der ikke blev udarbejdet et design klasse diagram, da det ikke gav nogen værdi med det analyseklassediagram der i forvejen var blevet udarbejdet. Det gav derimod væsentligt mere værdi at lave sekvensdiagrammer med rimelig høj grad af detaljegrad, som beskrev flowet i programmet og definerede klassers ansvarsområder. Disse sekvensdiagrammer følger ikke UML syntaks til punkt og prikke, da det tager for lang tid i forhold til at lave UML sketches.

### 9.5.4 Dynamiske side af designmodellen

#### Notifikationer

For at specificere yderligere hvordan notifikationer skal blive samlet, loadet, fremvist og programmeret blev dette diagram udarbejdet:



Figur 10 - Sekvensdiagram over notifikation interaktion med database

Som det kan ses på ovenstående figur, blev sekvensdiagrammet udført med en rimelig høj detaljegrad og er en udvidelse af analyse notifikationer sekvensdiagram. Når en bruger logger ind, bliver mainScreenController initialiseret hvor, afhængig af bostedet, bliver notifikationer hentet fra managere, som finder dem i databasen, ved at sende en forespørgsel til databasemanagere. Når notifikationerne er hentet og returneret bliver indholdet i centeret af skærmen sat til at være notifikationer, og notificationsController bliver initialiseret.



### 9.5.5 Design af persistenslag:

Persistenslaget består af kontrakt klasser, data klasser, connection klasser til databasen og en manager til at styre udefrakommende forespørgsler for data.

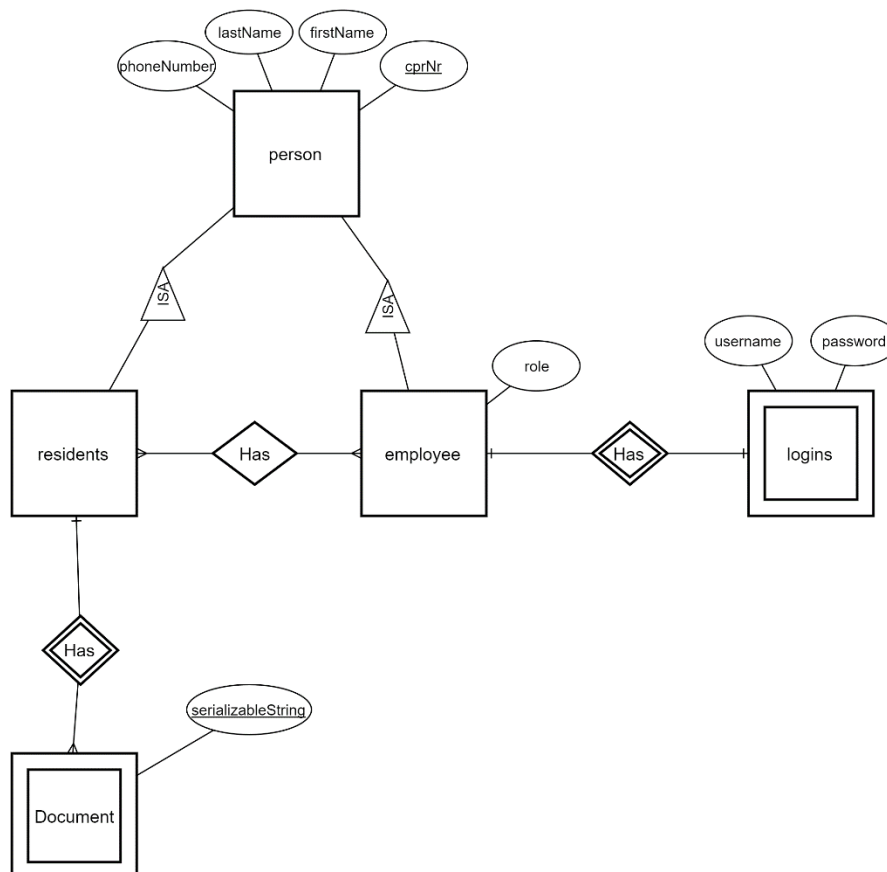
Kontraktklasser er blevet udarbejdet for at undgå fejl. I tilfælde af at ens database bliver opdateret, med nye navne eller nye tabel navne, så er det kun kontrakterne der skal ændres for at alle steder i koden benytter korrekte tabel og kolonne navne. Dette er en stor fordel i forhold til at manuelt omstrukturerer steder i koden hvor et kolonnenavn fra databasen bliver refereret.

Dataklasserne er opdelt i forhold til ansvar, så en DocumentData-klasse kun indeholder CRUD operationer for dokumenter. Derudover, hver gang en dataklasse henter data fra databasen skal der først oprettes forbindelse til databasen, dette gøres ved at få et Connection objekt fra en af Connection klasserne som indeholder en Connect metode. Når data klassen er færdig med kommunikationen med databasen, lukkes connection objektet.

#### 9.5.5.1 Databasedesign

##### 9.5.5.1.1 Første iteration

I første iteration startede blev systemet udviklet med et fil-baseret data system, men det fungerede ikke optimalt, derfor blev SQLite benyttet i stedet, til at gemme lokale data som senere i anden iteration ville blive lagt op i skyen ved brug af PostgreSQL, derfor blev et database E/R-design udarbejdet som kan ses på nedenstående figur:



Figur 11 - Entity diagram første iteration

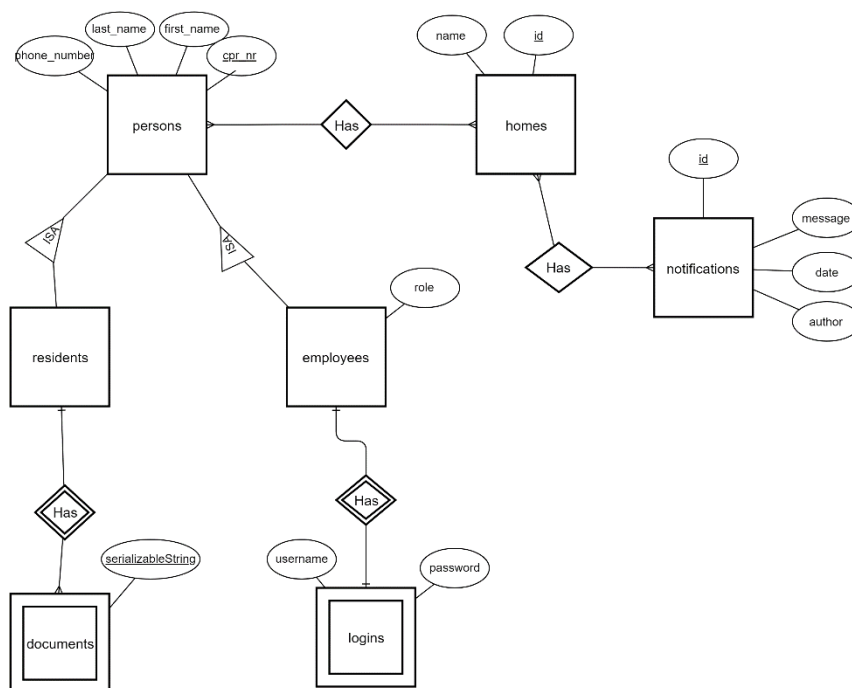
Det ses på figur 11 at databasen bestod i første iteration af 5 entities. Person har 4 attributter med cpr\_nr som primary key. Person nedarver til Residents og employee. Disse to har en forbindelse, hvor en employee har en

resident og omvendt. Derudover har residents tilknyttede dokumenter, som er gemt i en serializableString. Employee har attributen rolle og en relation til logins.

Alle forbindelser mellem entities som ikke er weak blev lavet om til en tabel som indeholdt primary keys fra de to tilhørende tabeller. For eksempel blev forbindelsen mellem residents og employee lavet om til en tabel med to kolonner, employee\_cpr\_nr og resident\_cpr\_nr for hurtigt at kunne finde alle tilknyttede beboere baseret på et employee ID. Dette gøres for at opnå en god datastruktur hvor databasen ønskes i BCNF eller 3NF, som er database strukturer der sikrer højest effektivitet.

#### 9.5.5.1.2 Anden iteration

Som det blev beskrevet i analyse-afsnittet, skete der et skift i dataafgrænsningen i anden iteration for hvordan dokumenter skulle opbevares og hvilke personer der har rettigheder til at se på dokumenterne. Det blev ændret fra, at en medarbejder havde tilknyttede beboere med tilknyttede dokumenter, til at et bosted skal have tilknyttede medarbejdere og beboere, hvor alle medarbejdere på et bosted kan se alle beboers dokumenter fra bostedet. Derudover skulle notifikationer bliver implementeret og derfor skulle disse også opbevares i databasen. Den nye implementering kan ses på figur 12.

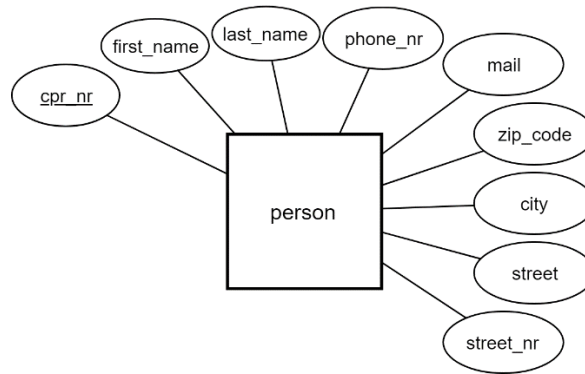


Figur 12 - Entity diagram for anden iteration

Hvor der nu er 7 entities, med tilføjelsen af homes og notifikationer. Hvor de hver har et id som primary key til at kunne identificere dem. Ligesom i første iteration blev alle forbindelser som ikke er weak forbindelser lavet om til en tabel for at forbinde dem.

Det var også tanken at her i anden iteration skulle få indført Kryptering og få fjernet cpr\_nr fra persons så der ikke var lige så mange følsomme data i vores database.

Dette endte med ikke at blive implementeret og skubbet til en senere iteration. Dog blev det indført at man kunne slå personer op i et simuleret cpr-register for at simulere funktionen, derfor blev der lavet et database-design som kun indeholder person data som ses på figur 13.



Figur 13 - cpr-register database design

## 9.6 Implementering

De primære funktioner af programkoden vil blive gennemgået i dette afsnit med små kode eksempler. Disse funktioner er, kontrol af brugers rettigheder, administrator muligheder, logning af systemet, databaseimplementering og threads.

Gruppen har brugt associationerne i analyseklassediagrammet til at finde referencevariabler, f.eks har klassen *Beboer* en aggregering til klassen *Dokument*, som betyder, at *Beboer* i koden har en variabel der referer til et objekt af typen *Dokument*.

Gruppen definerede metodernes navne ud fra designklassediagrammet, men for at finde kroppen til metoderne brugte de her operationernes sekvensdiagrammer. Her kunne gruppen se hvilke metoder som kaldes inden for en given metode i sekvensdiagrammet og hvilke objekter metoderne kaldes på. Dette var et vigtigt element i at sikre at arkitekturen i koden blev overholdt, og hjalp samtidig gruppen til at tage implementeringen af metoderne skridt for skridt.

Selvom alle brugsmønsterrealiseringerne aldrig blev fuldendt, har sekvensdiagrammer hjulpet med at definere hvad metoderne skulle indeholde og hvordan klasserne skulle snakke sammen. Disse sekvensdiagrammer var ikke fremstillet som en del af brugsmønsterrealisering, som man normalt ville gøre, men enkeltstående for at visualisere hvordan specifik funktionalitet i systemet skulle fungere og arbejde sammen. Figur 7 viser et sekvensdiagram over et par metoder der skal hente information fra databasen. I dette sekvensdiagram er der ikke fokuseret på strengt at overholde reglerne indenfor sekvensdiagrammernes syntaks. Det er udelukkende lavet for programmøren selv for at gøre det nemmere at forstå hvad der skulle snakke sammen under implementeringen. Dette mente vi var nemmere og meget mindre tidskrævende at følge analyse-workflowet fuldstændigt. Denne metode brugte vi flere gange igennem implementeringen for at gøre processen lidt nemmere.

### 9.6.1 Kontrol af brugers rettigheder

Ud fra klassediagrammet, figur 5, i analysefasen, besluttede gruppen at bruge hver enkelt employee til at styre rollefordelingen i systemet, og derved styre de forskellige rettigheder. Brugernes rettigheder i systemet, styrede gruppen med en række attributter som lå i Employee klassen. Disse attributter bliver tildelt brugere når der bliver oprettet en instans af underklassen.

Figur 14 viser klassen Socialrådgiver, og hvordan en bruger som bliver oprettet som socialrådgiver får sat værdierne på de relevante attributter til "true".

```
public Socialrådgiver(String firstName, String lastName, String phoneNr, String cprNr) {
    super(firstName, lastName, phoneNr, cprNr);
    setCanCloseCase(true);
    setCanPrintDoc(true);
    setCanCreateReportDocs(true);
    setCanEditDoc(true);
}
```

Figur 14 - Socialrådgiverklassen i koden. Rollens rettigheder sættes i constructoren

I systemet ville attributterne blandt andet blive kontrolleret i Initialize() metoden for SagerTabControlleren, som sørgede for at vise de elementer som var relevant for bruger med et givent sæt af rettigheder. Figur 15 viser et eksempel på hvordan SagerTabControlleren viser eller skjuler dele af systemet, hvis brugers attributter ikke opfylder de værdier systemet ønsker.

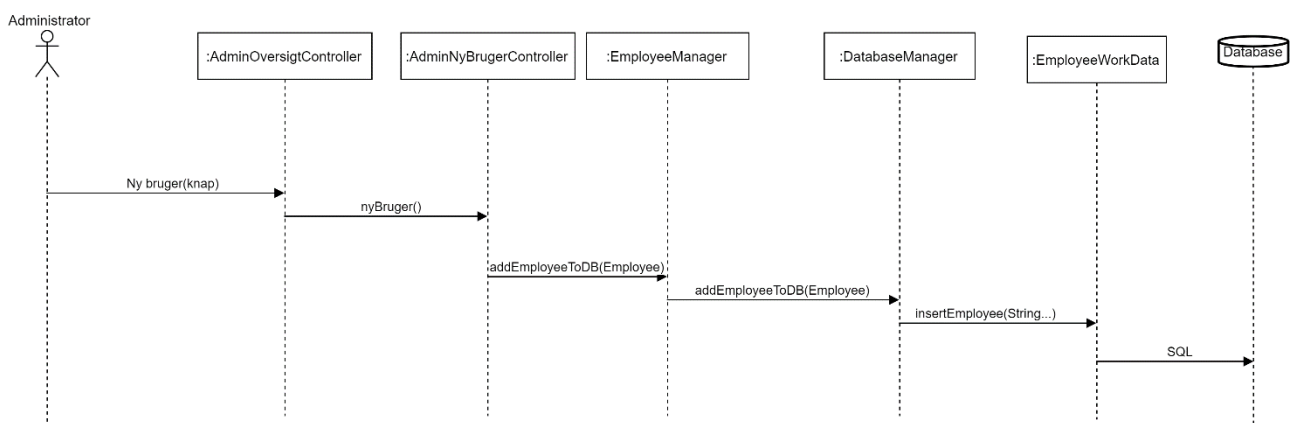
```
// Changes the create new document button if the user is Sagsbahandler
if (loggedInEmployee.canCreateNewProcessDoc()) {
    createVUMDocBtn.setText("Opret ny sag");
}

if (!loggedInEmployee.canCreateReportDocs() && !loggedInEmployee.canCreateNewProcessDoc()) {
    createVUMDocBtn.setVisible(false);
}
```

Figur 15 - Eksempel på rettighedstjek

### 9.6.2 Administrator muligheder

Systemet har et adminpanel og en administratorbruger. Administratorens muligheder i systemet er at oprette en ny bruger, samt redigere og slette brugere. Til implementeringen af oprettelsen af ny bruger, bruge gruppen sekvensdiagrammet vist i figur 16.



Figur 16 - Admin lav ny bruger

Brugen af sekvensdiagrammet gjorde det muligt for gruppen at følge metoderne og dermed implementere de korrekte metodekald i de relevante klasser. Figur 16 viser metoden `addEmployeeToDB` fra `EmployeeManager` klassen. Denne metode blev oprettet som resultat af sekvensdiagrammet, og derved blev metodekroppen implementeret til det videre kald til `Databasemanageren`. Denne metode også for at oprette en sammenkobling mellem den nye bruger og de hjem vedkommende var tilknyttet, samt at oprette den nye bruger login i databasen.

```
public void addEmployeeToDB(Employee employee, String username, String password, List<Home> homes) {  
    database.insertEmployee(employee.getCprNr(), employee.getFirstName(), employee.getLastName(), employee.getPhoneNr(),  
        employee.getEmployeeRole());  
    for (Home home : homes) {  
        database.insertPersonHomeLink(employee.getCprNr(), home.getId());  
    }  
    database.insertEmployeeLogin(employee.getCprNr(), username, password);  
    EMPLOYEE_LOGGER.log(Level.INFO, "{0}Has added: {1} To the Database", new Object[]  
        {informationBridge.getLoggedInEmployee().getFirstName(), employee.getFirstName()});  
}
```

Figur 17 - Tilføj bruger til database

### 9.6.3 Logning af systemet

I systemet blev der implementeret en log således at man hele tiden kunne gå tilbage og kontrollere brugeres aktivitet. Dette er en vigtig del af datasikkerheden i sådanne systemer, da man dermed kan kontrollere om medarbejdere har haft adgang til dele af systemet de ikke måtte, eller om administratoren havde lavet en bruger som dermed kunne misbruges til at finde oplysninger om beboers sager.

For at lave loggen i systemet blev Javas egen `Java.util.logging` brugt. Det er en pakke der indeholder muligheden for at systemet kan logge forskellige aktiviteter og samtidig kategorisere dem efter alvorlighedsgrad. I systemet findes to forskellige loggere, den ene sørger for at logge administratoren, mens den anden sørger for at logge de resterende brugere. Opdelingen skyldtes at loggen genererer én fil pr. log og derved kan vi holde loggen overskuelige og relevante. Figur 18 viser hvordan admin loggeren er sat up. Hver af loggernes filer bliver tildelt et datostempel med dag, måned og år. Dette betyder at der bliver oprettet en ny logfil hver dag, og dermed kan man gå tilbage og sortere i loggen efter dato.

```
public static void setupLogger() {
    LogManager.getLogManager().reset();
    ADMIN_LOGGER.setLevel(Level.ALL);

    Date date = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd") ;
    try {
        FileHandler fh = new FileHandler("AdminLogger " + dateFormat.format(date), true);
        SimpleFormatter simpleFormatter = new SimpleFormatter();
        ADMIN_LOGGER.addHandler(fh);
        fh.setFormatter(simpleFormatter);
        fh.setLevel(Level.FINE);
    } catch (java.io.IOException e) {
        // don't stop my program but log out to console.
        ADMIN_LOGGER.log(Level.SEVERE, "File logger not working.", e);
    }
}
```

*Figur 18 - start logging*

Loggen bruger en SimpleFormatter, da denne giver logbeskeden uden irrelevante informationer. Det er også muligt at lave sin egen formatter, men det nåede gruppen ikke i dette projekt. Figur 19 viser et eksempel på hvordan loggen ser ud når en given bruger har lavet en aktivitet i systemet.

```
maj 24, 2019 3:46:19 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
WARNING: Employee logger Startet

maj 24, 2019 3:46:34 PM ssb.domain_layer.person.EmployeeManager addResidentToHome
INFO: thomas has added resident: Iben Ingeborg to home 1

maj 24, 2019 3:46:38 PM ssb.domain_layer.document.DocumentManager addDocument
INFO: thomas has added a new document for resident: Iben Ingeborg

maj 24, 2019 3:46:47 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
INFO: nicolai Has logged in!

maj 24, 2019 3:46:47 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
WARNING: Employee logger Startet

maj 24, 2019 3:47:20 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
INFO: nicolai Has logged in!

maj 24, 2019 3:47:20 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
WARNING: Employee logger Startet

maj 24, 2019 3:47:27 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
INFO: jeppe Has logged in!

maj 24, 2019 3:47:27 PM ssb.domain_layer.person.EmployeeManager$1 succeeded
WARNING: Employee logger Startet
```

*Figur 19 - Eksempel på log*

Inden i loggen kan det ses hvilken klasse loggen er skrevet fra. Samt hvilket niveau hver enkelt log besked har. I loggen ses niveauet "INFO" som er niveauet for alle de log beskeder som man ville forvente en bruger ville generere. Niveauet "WARNING" er et udtryk for en vigtig hændelse, og i denne logbesked er den brugt til at vise at loggen er startet op og hvornår denne er startet. I administrator loggen vil hver aktivitet en administrator laver blive tildelt niveauet "SEVERE".

#### 9.6.4 Afsnit om databaseimplementering

Under projektets første iteration brugte gruppen SQLite, som laver en lokal database på computeren. Dette var gruppens alternativ til at gemme dokumenter og brugere i filer. Igennem SQL havde gruppen forberedt brugen af database i skyen, og da databasen lå lokalt på den enkeltes computer, kunne gruppens medlemmer lave test og rettelser i databasen uden påvirkning fra andre gruppemedlemmer som brugte databasen.

I anden iteration kobledes gruppen databasen på i skyen via ElephantSQL. ElephantSQL er dog væsentligt langsommere da databasen er en gratisversion som delte server med mange andre databaser.

Designfasen gjorde det muligt for gruppen at danne sig et overblik over hvordan databasen skulle opsættes, samt hvilke informationer databasen skulle have fra systemet, for at blive udfyldt. Dette var en essentiel del af at kunne implementere en ensartet fungerende database.

Databaseopkoblingen blev lavet igennem forskellige klasser som hver har ansvaret for at arbejde med hver sin del af databasen. PersonData er et eksempel på en klasse som blev oprettet til at håndtere alle personer i systemet. PersonData's implementering indbefatter indsættelse af person, opdatering, samt fjernelse af en person fra databasen. PersonData står også for at koble personen sammen med det hjem de skal tilknyttes i databasen. Figur 20 viser et eksempel på metoden InsertPerson og hvordan gruppen brugte informationerne fra en person til at indsætte en ny person i databasen ved brug af et PreparedStatement, som udskiftede alle spørgsmålstejn, i SQL Queries med de værdier som skulle indsættes i databasen. De resterende førnævnte implementeringer bruger samme opbygning som InsertPerson, men med **UPDATE** og **DELETE**.

```
void insertPerson(String personCpr, String firstName, String lastName, String phoneNumber) {  
    String sqlPersonInsertion = "INSERT INTO " + PersonsContract.TABLE_NAME + " VALUES "  
        + "(?, ?, ?, ?)";  
    try (Connection connection = db.connect();  
        PreparedStatement insertStatement = connection.prepareStatement(sqlPersonInsertion)) {  
        insertStatement.setString(1, personCpr);  
        insertStatement.setString(2, firstName);  
        insertStatement.setString(3, lastName);  
        insertStatement.setString(4, phoneNumber);  
        insertStatement.execute();  
    } catch (SQLException ex) {  
        System.out.println(ex.getMessage());  
    }  
}
```

Figur 20 - Tilføj person til database

#### 9.6.5 Threads

Brugen af Threads var med til væsentligt at øge hastigheden på systemet da dette tillader kald til databasen at ske simultant med at andre dele loades. Derved skulle systemet ikke hele tiden vente på svar fra databasen før systemet kunne fortsætte. Dette betød en markant kortere tid fra brugeren var logget ind til at systemet var opstartet.

Gruppen kunne bruge analyse og designdiagrammerne til at danne sig et overblik over hvor det ville give mest mening af implementere brugen af Threads. Ved hjælp af sekvensdiagrammerne i designfasen, kunne gruppen se hvilke metoder der kaldte databasen og implementere en Thread løsning.

Figur 21 viser implementeringen af Thread i hentningen af notifikationer fra databasen.

```
new Thread(new Task() {  
    @Override  
    protected Void call() throws Exception {  
        notificationLoadingIndicator.setVisible(true);  
        notificationManager.checkForNewNotifications(currentHome.getId());  
        return null;  
    }  
  
    @Override  
    protected void succeeded() {  
        notificationLoadingIndicator.setVisible(false);  
        Platform.runLater(() -> {  
            showNotifications();  
        });  
    }  
}).start();
```

Figur 21 - Eksempel på threading

Et af læringspunkterne ved brugen af Threads var at få dem synkroniseret således at visse Threads ikke blev kørt før logikken fra tidligere Threads var færdig. Hvis dette skete, ville der opstå problemer i at metoder fra den næste Thread ikke kunne køre eller gav fejl fordi den manglede de oplysninger som metoderne fra den endnu ikke færdige Thread skulle hente. Til løsning af dette problem kunne gruppen igen bruge sekvensdiagrammerne til at finde de metoder som krævede oplysninger fra andre Threads, før de kunne køre, og derved implementere deres løsning, så Threads ikke blev færdige for tidligt.

### 9.6.6 Delkonklusion

Gruppen fik løbende under implementeringen brugt de redskaber der fulgte med 2.semesteres viden. Dette giver en bedre optimering af systemets hastighed samt at viderebygge den overordnede funktionalitet i systemet. Opkobling med database gjorde det muligt at gemme objekterne fra systemet, og derved gøre systemet brugbart på flere forskellige enheder samtidig. Da det var gruppens første gang med opsætning og brug af database var dette et stort læringsmæssigt punkt. Brugen af UP i opbygningen af systemet gjorde der var tænkt over den generelle opbygning, dog blev størstedelen af metoderne oprettet løbende, hvor gruppen kunne have draget større fordel af at bruge flere sekvensdiagrammer. UP hjalp gruppen med at styre de vigtige detaljer af dataafgrænsningen, da der i designfasen kunne drages konklusioner fra analysefasen om hvilke rettigheder hver enkelt bruger skulle have og dermed simplificere implementeringen.

## 9.7 Test

Både i løbet af og efter udviklingsprocessen, er programmet blevet testet for at se, om det lever op til de krav der er stillet. I dette afsnit vil der præsenteres to forskellige tests der er foretaget på programmet i løbet af projektet.

Imens man er i gang med at udvikle sit program, kan man med fordel på samme tid foretage tests. De såkaldte udviklingstests foregår ved, at man skriver sine tests før funktionaliteten implementeres. På denne måde er man sikker på hvornår programmet opfylder en særlig funktionalitet. I dette projekt blev der anvendt udviklingstest med J-Unit, men i et begrænset omfang. Gruppen lærte først om denne testmetode langt inde i projektforløbet, og har derfor ikke kunne gøre nytte af den siden starten af projektet. Derfor var det svært at overgå til en helt ny form for udviklingsmetode, så der er mange steder, hvor dette er ikke er anvendt



Efter programmet var færdigudviklet, har det været nødvendigt at teste dets kvalitet. For lige netop dette produkt, var det vigtigt, at det reagerede hurtigt nok for at opretholde en god brugeroplevelse. For at teste programmets responstid blev der udført i alt 8 tests:

1. Hentning af 1000 notifikationer som de eneste i databasen
2. Hentning af 5000 notifikationer som de eneste i databasen
3. Finde 1000 notifikationer specifikt til et bosted blandt 6000 notifikationer
4. Hentning af 500 dokumenter som de eneste i databasen
5. Hentning af 2000 dokumenter som de eneste i databasen
6. Hentning af 500 dokumenter specifikt til et bosted blandt 2000 dokumenter tilhørende andre bosteder
7. Oprettelse af nyt dokument
8. Opdatering af dokument blandt 10.000 dokumenter

Tallene er baseret på disse overvejelser:

1. Et bosted som skriver 5 notifikationer om dagen, vil der gå 200 dage før databasen vil være fyldt med 1000 notifikationer.
2. Stress test. Et bosted der skriver 25 notifikationer om dagen skal databasen kunne hente 5000 notifikationer indenfor 200 dage.
3. Når der er mange bosteder, skal notifikationer kunne findes og sorteres blandt notifikationer som ikke tilhører bostedet som vil tage ekstra tid, her bliver der stress testet med 6000 dokumenter.
4. 500 dokumenter svarer til et mindre bosted som har 50 beboere som hver især har 9 VUM-dokumenter.
5. Stress test. 2000 dokumenter svarer til et større bosted som kan have omkring 200 beboere som hver især har 9 VUM-dokumenter
6. Når der er mange bosteder, skal dokumenter kunne findes og sorteres blandt dokumenter som ikke tilhører et specifikt bosted.
7. Essentiel del af programmet. At indsætte nyt dokument og gemme i databasen.
8. Finde specifikt dokument blandt 10.000 dokumenter hvilket svarer til 60 bosteder som hver har plads til 25 beboere som hver især har 9 VUM-dokumenter gemt i databasen.

For at teste disse scenarier bliver databasen fyldt op med den mængde som skal testes (for eksempel 2000 dokumenter) og derefter bliver programmet kørt start og slut tider vil blive noteret i milisekunder. Differencen mellem start og slut bliver målt 5 gange hvorefter gennemsnittet af de 5 forsøgsresultater bliver det endelige resultat.

Resultaterne af disse tests kan ses i tabellen nedenfor:

<i>Test nr.</i>	<i>Resultat (milisekunder)</i>
<i>1: Hentning af 1000 notifikationer</i>	902,4
<i>2: Hentning af 5000 notifikationer</i>	6379,2
<i>3: Finde 1000 specifikke notifikationer blandt mange</i>	852,8
<i>4: Hentning af 500 dokumenter</i>	1984,8
<i>5: Hentning af 2000 dokumenter</i>	2529,2

6: Hentning af 500 specifikke dokumenter blandt mange	2819,6
7: Oprettelse af nyt dokument	676,8
8: Opdatering af et specifikt dokument blandt 10.000 dokumenter	345,8

Alle 5 forsøg for hver test er vedlagt som bilag.

Som det kan ses, er der stor forskel på hentning af 1000 notifikationer og 5000 notifikationer, hvilket vil sige databasen ikke er særlig skalerbar. Dog tager det lige så lang tid at hente 1000 notifikationer hvor alle notifikationer bare skal hentes i forhold til at hente 1000 notifikationer hvor der skal sorteres blandt 6000 notifikationer hvilket er et underligt resultat, da det burde tage længere tid, hvilket man også kan se er tilfældet i en lignende test med dokumenter.

Hentning af dokumenter er ligesom notifikationer heller ikke skalerbare, da det tager et halvt sekund længere at hente 2000 dokumenter fremfor 500 dokumenter. Derudover tager det et helt sekund længere at finde 500 dokumenter blandt mange dokumenter fra flere forskellige bosteder. Det realistiske resultat er derfor at det vil tage 3 sekunder at hente 500 dokumenter, da der vil være flere bosteder end et.

Oprettelsen af et nyt dokument tager 700ms at få skrevet til databasen, hvilket er acceptabelt.

Opdatering af et specifikt dokument blandt 10.000 dokumenter viser sig ikke at være noget problem.

## 10 Diskussion

Projektforløbets resultater vil blive opsummeret ved at beskrive om delspørgsmålene fra indledningen er blevet besvaret og hvordan de er blevet besvaret. Herefter vil styrker og svaghederne af resultaterne blive diskuteret og om resultatet af projektet er tilfredsstillende.

- **Er det muligt at begrænse brugeres adgangsrettigheder baseret på deres jobtitler?**

Brugerens adgangsrettigheder er blevet begrænset i forhold til deres rolle. Systemet endte med at have fem forskellige roller; admin, socialpædagog, socialrådgiver, sagsbehandler og vikar. Rollernes rettigheder bliver unikt defineret baseret på handlinger som systemet kan. Nogle af de handlinger er for eksempel: "rolle må oprette nye VUM-dokumenter". Som udgangspunkt er der ingen af rollerne der må udføre handlingerne, men socialrådgiverrollen har specificeret at de skal have adgang til at oprette nye VUM-dokumenter. Admin har som den eneste rolle for eksempel specificeret handlingen "har adgang til admin panelet".

- **Er det muligt at begrænse brugerens adgangsrettigheder baseret på deres tilknyttede bosteder?**

Udover at brugerens roller bestemmer hvilke handlinger de kan udføre, så er der også indført dataafgrænsning baseret på tilknyttede bosteder.

En medarbejder kan have en til flere tilknyttede bosteder. Hvis en medarbejder har mere end et bosted vil de under login, vælge et bosted som de vil logge ind på. Dette betyder at en medarbejder kun kan se information for ét bosted ad gangen. Den information som er begrænset pr. bosted, inkluderer ting specifikt for det bosted. For systemets nuværende tilstand inkluderer det følgende ting; tilknyttede beboere, deres VUM-dokumenter og notifikationer skrevet på bostedet. I takt med at systemet vil blive videreudviklet vil denne liste inkludere ting såsom dagbøger, journaler, kalender mv.

- **Skal beboeren selv have mulighed for at tilgå systemet?**

Det er blevet besluttet baseret på møder med EG Team Online og refleksion i gruppen, at beboeren ikke skal have mulighed for at tilgå systemet ude på afdelingerne. Beboeren skal derimod have adgang til at se alt information omkring dem via services såsom Sundhed.dk. Derfor skal systemet kunne udveksle data med Sundhed.dk, hvis det skulle udgives.

- **Vil en trelagsarkitektur kunne øge mængden af datasikkerhed i dette system?**

En trelagsarkitektur øger ikke umiddelbart datasikkerheden i et system. Den gør systemet mere robust og simple opbygget, hvilket betyder at det er lettere at indføre forsvar mod udefrakommende angreb, de rette steder i programmet, da alt funktionalitet der arbejder med data, befinder sig i datalaget. Dette har ikke været et fokuspunkt i dette projektforsløb, men ville være noget der er relevant at undersøge efterhånden som funktionaliteten begynder at være på plads.

- **Vil GDPR-loven være relevant at tage højde for i dette system?**

Der er ikke blevet gået i dybden med selve GDPR-loven i dette projekt. Der er dog gået i dybden med dataafgrænsning. Regler såsom at føre en fortegnelse, dokumentation for databehandlingen, samtykke og at oplyse beboere og brugere hvordan deres data bliver behandlet, er ikke regler som er blevet fulgt i projektet.

Derfor ville dette system heller ikke kunne blive udgivet pga. GDPR. Det var utrolig relevant at tage højde for GDPR-loven i dette projekt under udviklingen i iterationerne. Derudover skulle man også undersøge om der er danske love som tilsidesætter GDPR-loven i nogle henseende.

- **Kan der designes en database, som er hurtig nok til at hente data til en travl hverdag på bosteder?**

I test-afsnittet viste det sig at databasen overordnet set er langsom, hvor det eneste testresultat under 500 ms er opdatering af et specifikt dokument. Dog bliver brugeren ikke påvirket af disse resultater, da der bliver brugt baggrundstråde til at undgå at UI fryser. Derudover når et dokument for eksempel bliver indsat eller opdateret skal brugeren ikke vente på databasen da den først opdaterer dokumentet i programmets hukommelse og derefter opdatere dokumentet i databasen på en baggrundstråd.

Dog skal brugeren vente på databasen, når notifikationer eller dokumenter skal hentes for første gang når de logger ind. Programmet forsøger at gøre dette imens brugeren laver noget andet, men i tilfælde af at brugeren går direkte ind på dokumentoversigten, vil de være afhængige af databasens hastighed. Det er derfor uacceptabelt, at det tager 3 sekunder for databasen at finde de rette dokumenter for det bosted man logger ind på.

Dette kan derfor godt tyde på, at der skulle have været indført indeksering i databasen som skulle være baseret på bosteders ID. Det ville betyde at databasen skal tjekke færre gange for at finde dokumenter tilhørende et bosted, hvis alle bosteders dokumenter er sorteret på forhånd.

- **Kan en skybaseret PostgreSQL-database tilkobles et Java-program?**

Der er blevet brugt ElephantSQL og PostgreSQL JDBC til at koble systemet skrevet i Java, sammen med en SQL-database.

- **Kan en PostgreSQL-database gemme dynamisk og komplekst data som VUM-dokumenter?**

En PostgreSQL database kan opbevare store filer i et format kaldet bytea. Typen bytea er binære data som kan gemme alle slags filer såsom .jpg, .png, .docx, .pdf m.m.

Under udviklingen af programmet blev det testet om programmet kunne åbne VUM-dokumenter i Word-format i stedet for i en FXML-skabelon. I denne forbindelse var der behov for at kunne gemme .docx-filerne og deres indhold et sted i databasen. Til dette var bytea typen oplagt da et Word dokument ikke fylder meget ift.

et billede. Dette kunne også have været gjort ved brug af en server hvorpå filerne bliver gemt i stedet for i en database. Løsningen med at åbne VUM-dokumenter i Word viste sig dog at være problematisk da SSB-programmet ikke længere har kontrol over dokumentet så snart det er åbnet i et eksternt program. Derfor blev handlinger såsom at gemme, autoudfyld, validering af felter m.m. vanskelige. Løsningen endte så derfor med at være tekstbaseret. Dokumenter vil blive gemt ved at dokumenter implementerer serializable som betyder at alt dokumentets indhold bliver krypteret til en lang streng, som kan dekrypteret igen når dokumenterne åbnes.

- **Kan kritiske handlinger foretaget af medarbejdere blive gemt i tilfælde af eventuelle fejl?**

Ved alle kritiske handlinger såsom at oprette et dokument, logge ind, oprettelse af ny bruger, opdatere dokument m.m. bliver aktiviteten gemt i en fil, som logger hvilken medarbejder der har foretaget en handling, hvornår det er sket, dato og tidspunkt og hvilken handling der er sket.

### Styrker

- Brugers rettigheder er baseret på deres rolle
- Fleksibel VUM-Dokument håndtering for medarbejdere ude på bosteder
- Nye VUM-dokumenter vil blive autoudfyldt med information der befinder sig i systemet på forhånd.
- VUM-dokumenter valideres på basale felter, med mulighed for senere udvidelse
- Solidt opbygget og skalerbar database struktur
- Fleksibel tilknytning til bosteder
- På trods af langsom database, bliver der hentet informationer i baggrunden så programmet ikke føles ligeså langsomt
- Skabeloner for VUM-dokumenter følger voksen udredningsmetodens standarder for redskaber.
- Logning af brugers aktivitet og handlinger som giver øget sikkerhed.

### Svagheder

Problemformuleringen er:

Kan alle VUM-dokumenter fra alle bosteder i hele Danmark samles i én dataløsning, hvor data-lovgivning bliver overholdt og brugervenlighed af systemet er i fokus?

Den første del af problemformuleringen spørger om alle VUM-dokumenter kan samles i én dataløsning. Ud fra testresultaterne kan det ses at ja, det er muligt at samle det i én dataløsning, men dataløsningen som er blevet brugt i dette projekt, er for langsom. Det tager væsentligt længere tid at hente data jo større databasen bliver, og da der er en del bosteder i Danmark, vil der skulle være ret mange dokumenter i databasen. Derfor skulle der have været indført indeksering for databasen og en hurtigere og bedre database skulle have været brugt. Databasen der bliver brugt, er en gratis plan som derfor ikke er specielt hurtig.

Den næste del af problemformuleringen omhandler data-lovgivning og brugervenlighed. Der bliver opbevaret mange personlige data i systemet, som ikke burde blive opbevaret, dette inkluderer cpr nr., login-brugernavn, login-adgangskode, adresse og telefon for alle mennesker i systemet. Der er ikke nogen grund til at ting såsom adresse og telefon bliver opbevaret i systemets hukommelse, og som minimum skal de blive krypteret i databasen. Derudover bliver cpr-numre brugt som primary key i databasen. Det betyder at alle handlinger som kræver identificering af en medarbejder, baseres på et cpr-nummer. Dette er ikke i orden, og burde ikke være tilfældet ift. datasikkerhed. Derudover burde cpr nummeret som minimum blive krypteret.

Brugervenligheden er ikke blevet testet ved brugervenligheds-tests og det vides derfor ikke om systemet er brugervenligt. Baseret på egen kritik, kunne oprettelsen af VUM-dokumenter godt forbedres da en serie af dialogbokse åbner op, ventende på brugerinput. Denne serie af dialogbokse tager et stykke tid.

Derudover bliver dokumenter opbevaret i databasen som en serializable string. Det betyder at hvis programkodens implementering af dokumenter ændrer sig, kan tidligere oprettet dokumenter ikke længere dekrypteres. Derfor i tilfælde af at implementering af dokumenter på et tidspunkt ændrer sig, skal der medtænkes bagudkompatibilitet, hvilket kan kræve meget ekstra arbejde og nogle dårlige løsninger.

### Bedre resultater

Brugervenlighed og data-lovgivning er essentielle dele af problemformuleringen, men undervejs i programudviklingen fik andre ting prioritet. Der kunne for eksempel været brugt tid på at implementere og bruge hashing algoritmer såsom SHA512, som ville kunne blive brugt til følsomme data, brugernavn og adgangskode. Derudover skulle medarbejdere blive identificeret med et andet ID end deres cpr-nummer. Deres ID kunne blive autogeneret baseret på en række faktorer, som også blive brugt som medarbejderens brugernavn. Desuden kunne der have været brugt mere tid på design klassediagrammer for at få bedre kvalitets programkode og bedre struktur.

Ift. Brugervenlighed ville det have været en fordel at have kontaktet en håndfuld mennesker som kunne teste brugervenligheden af nyligt indførte features undervejs i programudviklingen.

Derudover er brugen af serializable strings til at gemme dokumenter ikke optimal, da det giver store problemer i forhold til bagud-kompatibilitet. Dokumenter indeholder en række data felter, id, dokument navn, tilknyttet beboer, dokumenttype, redigeringsdato og oprettelsesdato. Dette data kan gemmes som en attribut i databasen. Problemet opstår når alle tekstbokse og checkbokse fra selve dokumentindholdet skal gemmes i databasen. Det ville derfor være bedre at gemme dokument-indholdet i XML eller JSON-format med kryptering. På denne måde ville selve dokument-indholdet aldrig gå tabt i tilfælde af at dokument implementering eller hvis VUM-skabelonerne ændrer sig. Der vil stadig være overvejelser ift. bagud-kompatibilitet, men det ville være væsentligt lettere at ændre implementeringsdetaljer uden risiko for tab af kritisk data.

## 11 Konklusion

Problemformulering er således:

Kan alle VUM-dokumenter fra alle bosteder i hele Danmark samles i én dataløsning, hvor data-lovgivning bliver overholdt og brugervenlighed af systemet er i fokus?

Ud fra denne problemformulering er der blevet udviklet et program som kan oprette, redigere og gemme redskaber fra Voksenudredningsmetodens, også kaldet VUM-dokumenter. Dokumenterne styrer og støtter socialfaglige medarbejdere i et udredningsforløb. Systemet der er blevet udviklet, hjælper medarbejderen med at håndtere disse redskaber.

### Opsummering af resultater

Programmet er blevet udviklet med rollebaseret adgang til funktionalitet i programmet. Systemet indeholder en række handlinger som ikke alle roller skal have adgang til såsom at oprette nye VUM-dokumenter. Derfor er der for hver rolle blevet defineret hvilke handlinger rollen skal have adgang til ved brug af sande eller falske attributter. Det har den fordel hvis en rolles adgangsrettigheder skal ændres, behøves det kun at ændres inde i rollens klasse.

VUM-dokumenterne bliver gemt i en sky-baseret database, som er i stand til at samle alle dokumenter fra alle bosteder i Danmark. Databasen er blevet opbygget i forhold til database-designprincipper.

Når systemet henter dokumenterne fra databasen, benytter systemet Threads, for at undgå at den grafiske brugerflade fryser.

Systemet indeholder også støttefunktioner når medarbejderen skal arbejde med VUM-dokumenter. Det inkluderer funktioner såsom; hentning af persondata fra CPR-registeret under oprettelse af en ny sag, automatisk udfyldelse af felter i VUM-dokumenter hvis systemet i forvejen kender informationen og validering af felter og notits om forkert udfyldte felter.

Derudover har systemet et logningssystem som sørger for at gemme brugers aktivitet, hvilket betyder at enten menneskefejl eller systemfejl kan spores tilbage til årsagen.

### Opsummering af diskussion

Systemet endte med at tilbyde relevante og gode funktioner til en socialfaglig medarbejder der arbejder på et bosted. Dog er der mangler ift. Data lovgivning og brugervenlighed. Databasen mangler kryptering, hashing og indeksering for at lave en sikker og skalerbar database. Derudover er den valgte sky-baseret database ikke hurtig nok og der ville derfor skulle bruges ressourcer på en bedre database plan. VUM-dokumenter bliver gemt ved brug af en serializable string i databasen, som betyder at dataen er afhængig af implementeringsdetaljer, hvilket betyder tab af data hvis implementeringen af dokumenterne ændres, som giver store komplikationer med bagud-kompatibilitet.

Der har også været mangel af fokus på brugervenlighed. Der skulle løbende under programudviklingen været blevet foretaget brugervenlighedstest på en håndfuld mennesker for at få feedback.

## 12 Perspektivering

I starten af projektet havde gruppen valgt at beskæftige sig med planlægningsmodulet for projektet, dog blev gruppen senere enige om at skifte til sagsbehandlingsmodulet. Grunden til ændringen var at gruppen på virksomhedsmødet fik indtrykket af at planlægningsmodulet var nedprioriteret, samtidig med at gruppen blev enige om at håndteringen af VUM-dokumenter sammen med databegrænsningen, kunne være et mere interessant område at udforske. På trods af det, hvis projektet skulle startes forfra, havde gruppen valgt at fastholde det først valgte modul, og fokusere på dette moduls krav, analyse, design og implementering og forsøge at være innovative og udforske muligheder som EG Team Online ikke selv har overvejet.

En frisk start på projektet, ville samtidig bidrage til et hurtigere arbejdsflow, da det var første gang gruppen arbejdede med UP. Gruppen kunne bruge den nyhvervede erfaring, i anden omgang af projektet. Gruppen ville have prioriteret design fremfor analyse og implementering. Da det ville gavne implementeringen og derved give bedre programkode.

Gruppen ville også have brugt database fra starten i projektet. Ifølge projektoplægget, skulle database-delen af programmet først implementeres i anden iteration af projektet. Projektgruppen indvilligede sig denne opfordring, men var derfor nødsaget til at anvende tekstfiler til opbevaring af data. Dette viste sig at skabe mere skade end gavn, da filsystemet voldte utallige problemer. Det endte med at gruppen overgik til en database efter disse problemer. Hvis projektgruppen startede med at implementere en database, kunne meget tid have været sparet.

Gruppen valgte at have høje forventninger til arbejdet i projektet, og til det afleverede produkt. Dette gjorde at gruppen havde rigeligt med arbejdsopgaver, men betød samtidig at gruppen ikke nåede alle ønskede implementeringer som de havde skrevet i deres scrum backlog. Blandt de mangler som det afleverede produkt har, ville gruppen, hvis der var tid, have implementeret en mulighed for at afslutte sager, samt fjerne beboere fra systemet. Dette er vigtige egenskaber for systemet og havde derfor været gruppens næste implementering. Gruppen havde diskuteret vigtigheden af de to førnævnte implementeringer, og var derfor tæt på udarbejdelsen af relevante diagrammer, for derefter at kunne implementere dem. Gruppen kom frem til vigtigheden af at kunne fjerne beboere fra systemet, både i forbindelse med GDPR, men også for at opnå kravet sat i projektcasen, som omhandlede at personer ikke skulle kunne ses, hvis de havde afsluttet et behandlingsforløb eller tidligere havde været beboer på samme eller et andet bosted. Gruppen ville have implementeret systemet

anderledes således at brugere blev gemt på baggrund af et id-nummer, i stedet for at bruge deres CPR-nummer. Dette skyldes at CPR-numre er følsomme personoplysninger og derfor bør undgås at skulle indtastes og bruges hvis muligt. Ligeledes ville gruppen have implementeret databasen sådan at denne havde krypterede strenge til at gemme brugeres kodeord, samt beboers CPR-numre. Gruppen ønskede også at implementere loggen sådan at exceptions og crash blev gemt i en separat log til systemudviklere for at simplificere debugging og optimering af systemet.

I løbet af projektet opdagede gruppen, at en del af de opstillede underspørgsmål til problemformuleringen var for irrelevante og svære at besvare. Derfor er de siden starten af projektet blevet ændret. Et af de tidligere underspørgsmål lød: *"Kan det færdige system opnå et højt tilfredsheds-niveau (over 75% af test-brugere kan lide systemet) ved brugervenligheds testning?"*. Gruppen fandt ud af, at dette spørgsmål skulle skiftes ud på grund af dets vaghed. Spørgsmålet er svært at forholde sig til, da det ikke forklarer noget om, hvordan brugervenlighedstesten skal foretages, og giver derfor ikke anledning til konkret handling. Derudover er det heller ikke meget sigende, at 75% af brugere skal kunne *"lide systemet"*. Andre spørgsmål som *"Kan man opbygge systemet af moduler, som er uafhængige af hinanden til en sådan grad, at hvert modul kan fjernes uden resten af systemet, bliver ubrugeligt?"* er også fjernet, da det ikke havde nok med hovedspørgsmålet at gøre. Hvis gruppen startede forfra med projektet, ville der blive lagt mere fokus på at finde relevante underspørgsmål til hovedspørgsmålet som gav anledning til konkret handling.

Angående scrum-rollefordelingen, ville gruppen også have foretaget nogle ændringer. Som tidligere beskrevet, fik alle gruppemedlemmer fælles tildelt rollen som product owner. Denne rolle er normalt skabt til én enkelt person, og dette ville eventuelt have været bedre i dette projekt. Igennem forløbet oplevede gruppen ofte, at programmets funktioner ikke virkede som oprindeligt planlagt. F.eks. blev det først sent opdaget, at vikarrollen havde rettigheder den ikke burde. En dedikeret product owner ville have bedre styr på disse krav og derfor opdaget fejlen tidligere. Et andet eksempel på at en dedikeret product owner kunne være brugbar, var dengang gruppen implementerede notifikationssystemet. Dette system endte med at tage forfærdelig lang tid, men havde ikke meget relevans for projektet. En dedikeret product owner ville eventuelt have forhindret dette.

## 13 Bibliografi

GDPR.DK ApS. (27. Maj 2019). *gdpr.dk*. Hentet fra gdpr.dk: <https://gdpr.dk/>

GDPR.DK ApS. (27. Maj 2019). *gdpr.dk*. Hentet 27. Maj 2019 fra Artikel 30 Fortegnelser over behandlingsaktiviteter: <https://gdpr.dk/databeskyttelsesforordningen/kapitel-4-dataansvarlig-og-databehandler/artikel-30-fortegnelser-over-behandlingsaktiviteter/>

GDPR.DK APS. (27. Maj 2019). *gdpr.dk*. Hentet fra De 7 principper: <https://gdpr.dk/persondataforordningen/de-7-principper/>

Socialstyrelsen. (27. Maj 2019). *Implementering af DHUV*. Hentet fra Socialstyrelsen.dk: <https://socialstyrelsen.dk/tvaergaende-omrader/sagsbehandling/voksenundredningsmetoden/viden-og-metoder/implementering-af-dhuv>



## 14 Procesrapport

### 14.1 Læring og refleksion

**Morten:**

Mit mål med anden semester var at opnå en dybere forståelse for emnerne, specielt emner som UP og Scrum, som begge var nye for mig, og sætte mit dybt nok ind i dem til at jeg ville kunne arbejde med dem i projektarbejdet. Min rolle som gruppemedlem involverer typisk at arbejde konkret med produktet, og samtidig hjælpe der hvor jeg kan. Jeg får ofte tanker som jeg deler ud af i gruppen, for dermed forhåbentlig at sætte fokus på et punkt vi kunne have misset, eller for at øge min egen viden inden for et bestemt område. Jeg er stor tilhænger af den agile proces, da jeg fandt at denne arbejdsmetode, og friheden der følger med, passer godt ind i mine måde at lære og arbejde med projektopgaver.

Jeg fandt det interessant at bruge en kombination af forståelsen fra Organisation og ledelse, sammen med Organisation og Software Engineering, for at få et dybere perspektiv på hvordan man fra et professionelt synspunkt kunne bruge værktøjer til at formidle sit arbejde ud til kunderne. Arbejdet med en gruppe af andre dygtige studerende har hjulpet mig meget, ikke kun til at komme fremad i projektet, men også for min forståelse for de emner vi har været igennem i andre fag. Dette har været specielt vigtigt når jeg til tider har følt der har været mange nye ting at holde styr på samtidig, i mange forskellige fag.

En kombination af flere holdninger og en konkret tilgang til projektet, mener jeg har været vigtig for at projektet har holdt sig på sporet, og at vi som gruppe har været engagerede for at opnå det bedste resultat. Forventningerne til projektet var vi, som gruppe, enige om blev sat højt. Dette gjorde også at var forbered på, der ville være elementer som vi ønskede at få med, men som ikke kunne nås på den korte tid vi havde til projektet. Derfor føler jeg der var mere vi kunne have lavet, men er samtidig tilfreds med det produkt vi endte ud med, da systemet implementerer en stor del af de ønskede funktionaliteter, samt de elementer vi mente var relevante i forbindelse med faget Videregående Objektorienteret Programmering.

Jeg syntes det er ærgerligt at vi midt i projektet desværre mistede et medlem af gruppen. Specielt denne person kunne have været værdigivende at have med under implementering og rapportskrivningen. Selvom jeg føler vi i projektet har lært en del nyt, mener jeg stadig det er ærgerligt at vi ikke har haft mere kontakt med organisationen, og har derfor måttet antage en masse selv. Vi havde virksomhedsmøderne, men på baggrund af de tidligere møder og hvor langt vi var kommet i projektet, havde vi i forvejen lavet antagelserne og derved mente jeg ikke vi fik nogen værdi i at tage til det sidste møde med virksomheden.

**Thomas:**

Som udgangspunkt vidste jeg ikke så meget om de agile processer som UP og Scrum. Igennem dette projekt regnede jeg med at sætte mig ind i disse metoder for projekthåndtering, da det var et krav at anvende dem. I gruppen forsøgte vi så godt som muligt at følge UP. Vi gennemgik alle workflowsne og prøvede at lave alle de opgaver der hørte til hvert workflow. Det viste sig at være ret svært, da man ender med at bruge meget tid på alting. Det endte med at blive umuligt at nå det hele. Derfor var der mange ting, vi var nødt til at springe over eller udarbejde til mindre færdiggørelsesgrad. Dette gjorde, at jeg ikke nåede at lære hvordan alle dele af UP faktisk skal bruges i et projekt som dette. F.eks. nåede vi sjældent at bruge designmodellen til at fremstille kode med, hvilket er et formål med den. Derfor har jeg ikke fået så meget ud af dette projekt som jeg ønskede fra starten. Generelt følte jeg, at det var mærkeligt struktureret i forhold til hvordan UP fungerer. Ifølge UP skal man arbejde på alle workflows samtidig (men til forskellige grad), men i dette projekt var der slet ikke tid til at lave det hele. I starten skulle vi lave et inceptionsdokument som indeholdt mange brugsmønstre og krav. Derefter begyndte iterationerne, hvor man blev nødt til gå direkte til at skrive koden til programmet. Hvis man først brugte tid på at færdiggøre analysen og lave designet, ville der slet ikke være tid til at have et ordentligt produkt i slutningen af projektet. Jeg føler, at hvis projektet havde spændet over længere tid, ville man måske kunne have nået det, men med kun projektarbejde to gange om ugen i et halvt år virkede det nærmest umuligt.

Scrum har været godt at bruge. Her har de mange forskellige værktøjer og ceremonier været hjælpsomme og interessante. Scrum er noget der anvendes i den virkelige verden, og brugen i dette projekt har givet en idé om hvordan det kommer til at foregå.

Det var også min forestilling i starten af dette projekt, at vi ville lære om hvordan man udfører et projekt i samarbejde med en organisation. Nu var der jo en rigtig case fra en virkelig organisation inde over projektet. Dette gav et mere realistisk syn på hvordan arbejdet kommer til at fungere ude i virkeligheden. Dog kunne det have været rart, hvis virksomheden var lidt mere inde over projektet. Vi har kun haft to virksomhedsmøder med dem, og man fik ikke så meget ud af det. Det endte med at virke næsten ligegyldigt at virksomheden var med inde over projektet.

**Oliver:**

Før starten af projektet havde jeg ikke arbejdet med hverken en agil proces eller Unified Proces. Derfor var mine forventninger og mål fra starten at jeg ville få noget erfaring med disse arbejdsprocesser. Derudover regnede jeg også med at få erfaring i at samarbejde med en virksomhed og forbedre mine egenskaber til at strukturere et projekt fra starten så vi ikke ville føle os bagud på noget tidspunkt. Jeg synes projektet har været fint nok, men den agile arbejdsproces er svær at udføre når man ikke arbejder intensivt med projektet. Vi har haft andre fag at se til samtidig med dette projekt, hvilket har betydet at projektet kun har fået omkring 33% af min tid. Dette føler jeg ikke er godt nok til ordentligt at arbejde med en agil proces såsom Scrum. Jeg synes dog Unified Proces har fungeret rimelig fint. I starten kørte processen alt for meget som en vandfaldsmodel. Det var først da det gik op for mig senere at man helst skal køre arbejdsopgaver i parallel, at det begyndte at fungere ordentligt. Jeg synes dog ikke at man skal benytte sig særlig meget af brugsmønsterrealiseringer, detaljerede sekvensdiagrammer, operations kontrakter m.m. da det overkomplicere tingene uden at give rent faktisk værdi og det føltes slet ikke agilt. Specielt med den mængde tid som vi havde i dette projekt på omkring 3 måneder. Det har føltes som om alt for mange ting har skulle blive udarbejdet ift. analyse og design modeller. Hvor vi så var nødsaget til at begynde på implementeringen da vi ellers ikke ville have nået at lave et færdigt produkt. Jeg glæder mig derfor ekstremt meget til næste semester at udforske andre metoder såsom user stories som er meget mere agilt og derefter lave design diagrammer baseret på user stories. Samarbejdet med virksomheden var heller ikke det jeg havde regnet med. Jeg havde troet det ville have været mere design og implementerings specifikt, men det viste sig at være meget overfladisk og til tider irrelevante svar på spørgsmål vi havde. Jeg havde også forventet at vi ville nå noget mere, det tog lang tid at udarbejde inceptionsdokumentet og forskellige analyse sekvensdiagrammer, hvor jeg næste semester vil fokusere mere på at lave design diagrammer så alle er enige om hvordan koden skal struktureres og implementeres. Der er brugt lidt for lang tid på implementering fremfor design hvertfald. Jeg føler derfor at jeg stadig mangler erfaring med en agil proces, på trods af dette projekt og at arbejde med krav fremstillet af en virksomhed i "den rigtige verden", jeg synes kun dette projekt har givet en introduktion til det, som godt kunne have været mere fyldestgørende.

**Michael:**

Jeg forventede at lære en mere omfattende måde at arbejdede med software projekt udvikling. Dette inkluderer at jeg forventede at følge med i undervisningen, for at kunne bruge relevant undervisning til fremmelsen af projektet. Dette viste sig at blive en udfordring, fra et personligt perspektiv og resulterede i at jeg kom bagefter i undervisningen. Jeg derved skulle forsøge at følge op på dette gennem resten af semesteret. Jeg nåde dog at for en mindre forståelse, ved hjælp fra projektgruppens medlemmer, og derved stadig være med projekt essigt. Jeg føler blandt andet at jeg her en, ok forståelse for UP, og meget god forståelse for Scrum som gennem projektet har været det primære fokus punkt. Jeg vil personligt sige at jeg ikke selv syntes at have gået nok op i semesteret og har også kunne mærke det meget gennem, implementeringsfasen. Denne personlige udfordring kommer i forlængelse af 1. semester hvor jeg gennem semesteret havde optakt til manglende personlig løst, og udfordringer. Dette er noget jeg har forsøgt at forbedre gennem semesteret, desværre uden den store forbedring før til slut i forløbet. Mine generelle mangler er primært, ved UP og programmering. Dette resulterede i at jeg derfor ikke har kunne implementere meget kode og arbejdet på de nødvendige analyser. Med forbehold

for dette er det blevet klart for mig, at jeg bør stramme gevaldigt op for at kunne lære det nødvendige gennem projektet.

**Nicolai:**

Inden starten på projektarbejdet, havde jeg ikke særlig godt styr på datastrukturering og dataafgrænsning som koncepter. Mit mål i projektet var at få mere viden om disse emner, da det blev givet til kende i projektcasen at det var emner der ville blive sat fokus på. Derudover ville jeg have fokus på hvordan man arbejder med softwareudvikling ift. en mere striks projektcase, hvori der ikke var tilladt fri leg ift. hvilket produkt eller problemer man kom ud af arbejdet med.

Jeg synes projektet umiddelbart har forløbet fint, og at vi som gruppe har haft høje ambitioner i vores arbejde. Jeg føler også selv at jeg kunne have haft en større indflydelse på arbejdet, men at min mangel på viden indenfor nogle af processerne har gjort at jeg ikke har kunne præstere lige så højt. Jeg er umiddelbart tilfreds med mit teoretiske og designarbejde, men utilfreds med mit praktiske arbejde, f.eks. i koden.

Jeg føler jeg lærte mest i inceptionfasen, da det var her at alt grundarbejdet for udviklingen af projektet blev lavet, det var her vi opstillede alle vores krav til projektet og gik de forskellige teorier igennem ift. at strukturere vores kommende arbejde i den næste iteration. Logbogen har også været rar personligt for mig, da jeg let kunne tilgå hvad der blevet lavet sidst vi arbejdede, og dermed let kunne gendanne min hukommelse fra der. Brugen af UP og Scrum i projektet føltes dog til tider rimelig irrelevant, da det at vi skulle behandle det som et projektarbejde, ikke er nær det samme som en arbejdsplads, hvor UP og Scrum er mere relevante værktøjer.

## 14.2 Projektstyring

I starten af projektet fungerede projektplanlægning og styringen meget godt. Tidsplanen blev opdateret, vi fulgte undervisningsmaterialet i samarbejde med projektet, der var mange overvejelser som blev dokumenteret i inceptionsdokumentet. Vi nåede så et punkt i projektet hvor der begyndte at opstå tidspres. Dette punkt opstod da vi opdagede hvor meget tid det tog at udarbejde UML-diagrammer, både analyse- og designdiagrammer, hvor der nærmest opstod panik i gruppen, da det også samtidig virkede som om vi skulle nå at implementere meget kode før vi ville kunne opnå et godt resultat til eksamenen. Det betød at nogle ting ikke blev lavet og dermed sprunget over. Vi underprioriterede ting såsom statiske side af analysemodellen og design modellen, tidsplan m.m. Vi begyndte derfor at prioritere implementering så vi ville have kørende kode til midtvejs seminaret. Nu når vi er nået slutpunktet indser vi så at vi burde have holdt fast i den oprindelige plan og derfor prioriteret design væsentligt mere. Det kunne så godt være at vi måske skulle have sat nogle personer til at designe og nogle til at lave kode koncepter undervejs.

## 14.3 Identifikation, analyse og bearbejdning af problemer

Casen har lagt op til en rimelig klar arbejdsopgave, det har dog været meget svært for os at identificere konkrete problemer og finde information som snakker om mulige problemer. Gruppen har for eksempel ikke noget kendskab til frustrationer eller irritationer for en socialfaglig medarbejder ude på et bosted. Vi ved ikke om de har nogle klager i forhold til VUM-dokumenter eller deres nuværende system. Hvis vi havde denne viden, ville vi kunne have udviklet et system som forsøgte at løse et konkret problem. Vi er dog endt op med et system som løser nogle problemer som vi har antaget socialfaglige medarbejdere står over for. Vi forsøgte igennem virksomhedsmøder at få mere kendskab til social faglige medarbejders daglige problemer og frustrationer, men virksomhedsrepræsentanten kunne ikke svare på noget af det. Det har derfor været meget svært at relatere til virkeligheden, da vi ikke har haft noget samarbejde med dem som rent faktisk skal bruge produktet.

## 14.4 Udviklingsprocessen

**Inceptionsfasen:**

Denne fase gik godt, der var rimelig meget arbejde men det var godt struktureret og godt planlagt det meste af tiden. Work flowsene i fasen blev også brugt som de skulle, det var primært fokus på forretningen, krav og

analyse. Hvor alle 3 ting blev dækket rimelig godt. Forretningsbegreber blev defineret i form af domænemodeller, supplerende krav og brugsmønstre blev defineret godt og der kom et godt overordnet billede af hvad produktet skulle ende med at kunne.

### **Elaborationsfasen:**

Her begyndte det at gå en smule ned ad bakke. Der var lige pludselig meget at lave, hvilket gjorde, at vi glemte at fokusere på hvordan workflowsne samt problemstillingen skulle arbejdes med. I starten af denne fase prøvede vi at arbejde med analyse og design, men vi vandt hurtigt ud af, at dette ville tage meget lang tid. Derfor gik det hurtigt op for os, at vi skulle i gang med at lave programmet, hvis vi skulle have et produkt i slutningen af projektet. Dette endte med, at arbejdet blev lidt noget rod, hvor flere ting burde være lavet, men manglede.

Derudover var der ikke nok fokus på problemformuleringen i denne fase. Vi glemte at følge de underspørgsmål, som vi havde defineret i starten, og nogle af de ting vi arbejdede med, var indre relevante i forhold til problemformuleringen.

Dette projekts elaborationsfase var delt op i to iterationer, ligesom sidste projekts udviklingsfase. Til gengæld var der ikke lige så stor skeln mellem de to iterationer i dette projekt. I sidste projekt var der direkte mål som skulle opfyldes i slutningen af hvert projekt. Dette gjorde det sværere at se formålet med de to iterationer i dette projekt. Dette tillod dog at vi selv kunne sætte målene for iterationerne, hvilket var hvad vi gjorde igennem iterationernes sprint backlogs.

## **14.5 Formidling og kommunikation**

Der er forsøgt at bruge et akademisk sprog igennem rapporten. Derudover blev der igen dette semester holdt et midtvejsseminar hvor vi skulle fremlægge vores projekt hvor vi fik raffineret vores formidlingsteknikker.

## **14.6 Samarbejde i gruppen**

Samarbejdet i gruppen har generelt været velfungerende. Gruppen havde løbende gode dialoger og debatter, som hjalp med at sikre retning i projektarbejdet.

Konstruktiv kritik i gruppensamarbejdet var en vigtig del af at vi som gruppe var enige i hvordan projektet skulle udformes. Kritikken internt i gruppen var også med til allerede udført arbejde blev optimeret, samtidig med at de medlemmer som havde udført arbejdet, lærte af kritikken til fremtidig brug. Arbejdseffektiviteten i gruppe har svinget lidt. Gruppen følte sig lidt demotiveret grundet de manglende svar fra virksomhedsmøderne, og derved ofte stod uden konklusion. Desuden dalede arbejds effektiviteten også lidt da vi mistede et gruppe-medlem, da det opbyggede samarbejde pludselig var noget andet.

Der var i gruppen et stærkt socialt fællesskab, blandt andet var der en fast kageordning, som sørgede for at der hver tirsdag når vi mødtes var kage til gruppens medlemmer. Det lykkedes os rent faktisk at overholde den hver eneste tirsdag. Derudover blev der ofte spillet om det sidste stykke kage, hvilket var et sjovt afbræk fra projektarbejdet.

Hvert gruppemedlem har taget en Belbin-test, som har defineret deres rolle. Thomas og Oliver blev vurderet til at være afsluttere, Nicolai blev kontaktskaber, Michael og Morten blev koordinører. Vi manglede altså en stor del af rollerne i projektgruppen. Gruppemedlemmerne blev påtvunget til at tage rollerne som ikke allerede var der, som f.eks. opstarter. Dette viste sig dog ikke at være et stort problem og gruppearbejdet endte med at fungere meget godt på trods af de manglende roller, fordi gruppemedlemmerne var villige til at udfylde de manglende roller når nødvendigt.

I starten af projektarbejdet lavede vi en samarbejdsaftale, hvor vi forinden drøftede forventningerne til projektet. Forventningerne var generelt høje, både i forbindelse med projektet, men også i forhold til engagementet i undervisningen. Efter start af projektet oplevede gruppen dog meget svingende ambitionsniveauer blandt medlemmerne både i forhold til projektet men også den øvrige undervisning, som gav en skæv fordeling af

arbejdsbyrden. Til næste semesterprojekt er det derfor meget vigtigt med realistiske forventningsafstemning for alle gruppens medlemmer.

Undervejs i projektet mistede gruppen et medlem, Mathias, som gjorde at gruppen mistede en stærk rapport skriver, han var god at have med i gruppen da vi skulle skrive inceptionsdokumentet. Det betød så ikke så meget i forhold til implementeringen, da Mathias stoppede den første uge af første iteration, derfor har det primært haft en betydning i forhold til skriftlig formulering.

I starten af projektarbejde lavede vi fælles en samarbejdskontrakt. Denne indeholdt aftaler og retningslinjer som alle var enige om. I dette projekt føler vi ikke, at den har været nok i brug. Den har haft mest betydning i forhold til godkendelse af hinandens arbejde ved f.eks. godkendelse af pull requests. Vi burde have henvendt os til den så straks vi blev uenige eller i tvivl om noget. Det har én gang været nødvendigt at ændre den da nogle gruppemedlemmer udeblev fra undervisningen. Der blev det nødvendigt at tilføje, at hvert gruppemedlem skulle have tilstrækkelig viden om de øvrige fag.

Gruppen kunne godt have været bedre til at hjælpe hinanden til tider, det har fungeret rimelig fint, men der er plads til forbedringer.

Samarbejdet i gruppen har ikke levet op til forventningerne, men der har været mange gode sociale aspekter.

### 14.7 Samarbejde med vejleder

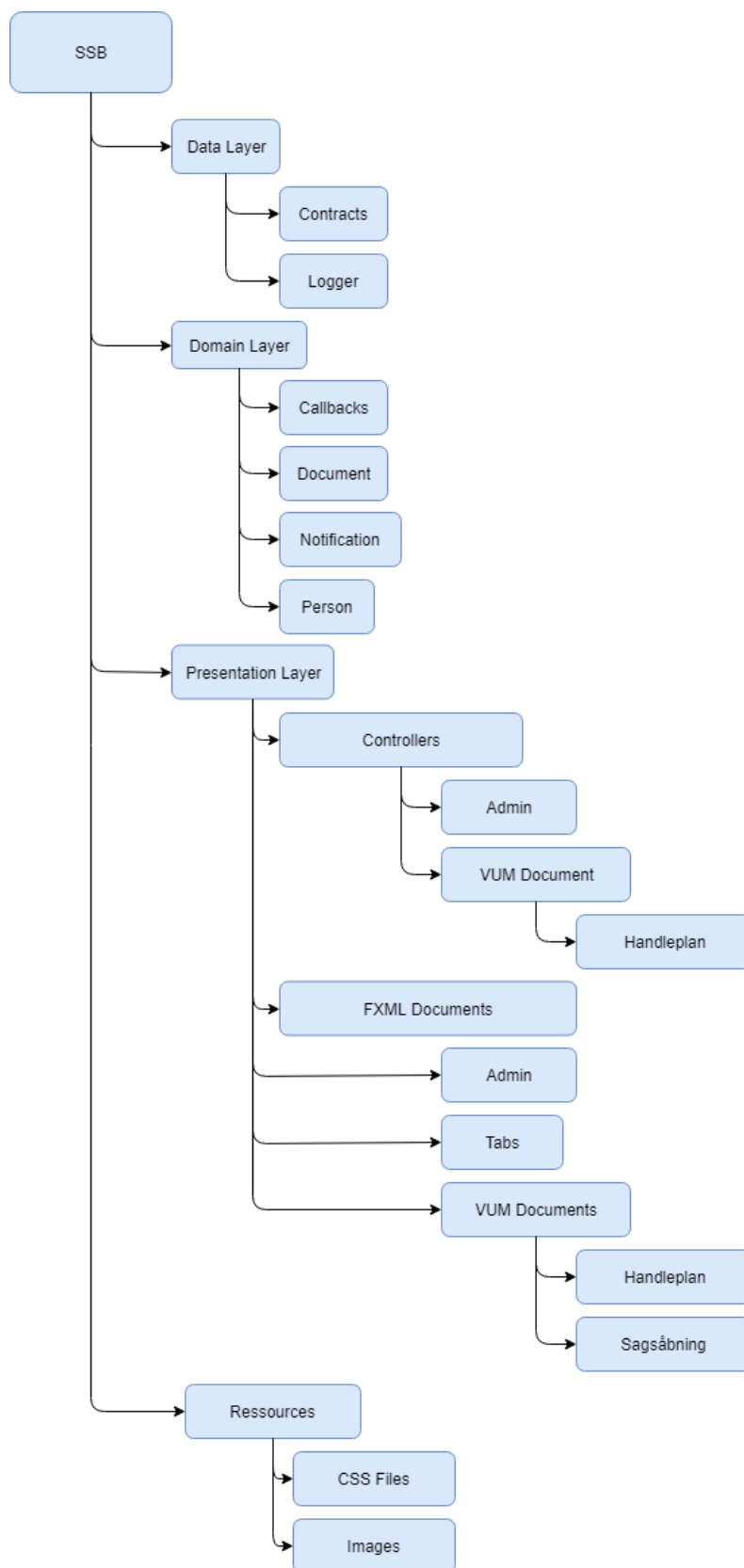
Der har været ugentlige møder med vejleder Jeppe Schmidt. Nogle få af møderne i starten og i midten af projektet blev ikke brugt til særlig meget, men helt generelt har der været stort udbytte i at snakke med vejleder. Vi har generelt fået et indblik i at tage vores egne beslutninger i projektet, da der har været en holdning til at vi som gruppe skal gøre det som vi føler giver os mest værdi. Udover det har vejleder givet god feedback på de spørgsmål vi har haft under vejledermøder, bl.a. har vi kunne diskutere dataafgrænsning ift. dokumenternes synlighed for medarbejdere på et bosted. Han har også været relativt nem at komme i kontakt med, hvis man havde spørgsmål uden for vejledermøderne. Dog har gruppen manglet lidt mere i dybde gående skriftlig feedback på afleveringsmaterialer.

Samarbejdet har haft en stor indflydelse på styringen af projektet, da vejleder har været god til at få os til at arbejde rigtig agilt, og arbejde med opgaver i parallel med andre. Vi har også kunne få inspiration gennem vejleder, da Jeppe har meget arbejds erfaring. Derudover har vejleder også åbnet vores perspektiv, hvor gruppen har låst sig selv fast til en tankegang.

Til næste projekt vil vi lægge flere kræfter i spørgsmål til vejledermøder, så der er mere at diskutere. Nogle gange følte vi, at vi ikke fik afsat nok tid til at finde gode og relevante spørgsmål til vejlederen. Derudover også gør spørgsmålene mere konkrete til vejleder så vi er sikre på at vi får det svar vi leder efter. Generelt bliver vi nødt til at huske på, at vejlederen er en uvurderlig ressource, som man bare skal udnytte så meget man overhovedet kan.

## 15 Kildekode

Figuren herunder viser hvordan systemets mappestruktur er bygget op.



## 16 Brugervejledning

### 16.1 Opstart af programmet

Gruppen har afleveret de nødvendige filer i en samlet .zip-fil ved navn "225 Java". Følgende er guide til hvordan enten første eller anden iteration af programmet startes op ved hjælp af NetBeans.

- Åbn NetBeans IDE
- Oppe under "File", vælg "Import project" → "From ZIP"
- Tryk på den første "Browse"-knap og find den downloadede .zip-fil og vælg den.
- Vælg en projektdestination ved hjælp af den anden "Browse"-knap
- Tryk "Import"
- Projektet "SSB" burde nu dukke op i venstre kolonne af NetBeans under "Projects"-fanen.
- Højreklik på projektet og vælg "Run"
- Programmet burde nu starte

### 16.2 Programmets login-system

Der er forskellige brugerlogins og adgangskoder baseret på hvilken medarbejderrolle man ønsker at logge ind som. (Hertil skal det nævnes at programmet er case-sensitive ved login.)

Rolle	Brugernavn	Adgangskode
<b>Sagsbehandler</b>	sagsbehandlerU	sagsbehandlerP
<b>Socialrådgiver</b>	rådgiverU	rådgiverP
<b>Socialpædagog</b>	pædagogU	pædagogP
<b>Vikar</b>	vikarU	vikarP
<b>Admin</b>	adminU	adminP

Som sagsbehandler har brugeren mulighed for at se alle dokumenter og oprette sagsåbningsdokumenter, så længe at de er tilknyttet den afdeling beboeren er tilknyttet til.

Socialrådgivere har adgang til at se alle dokumenter og oprette alle dokumenter undtagen sagsåbningsdokumenter (kun handleplan indtil videre) på eksisterende beboere.

Adminbrugeren har adgang til at oprette, redigere og slette brugere fra systemet og allokere brugerne til de forskellige bosteder.

Vikaren har ikke adgang til sagsbehandling.

Alle kan bruge notifikationssystemet.

Ved login kan brugeren vælge, alt afhængig af hvilken rolle man er logget ind som, hvilken handling man nu vil påtage sig, enten fra startsiden eller ved at navigere ved hjælp af overskrifterne i sidebaren.

### 16.3 SQL-script fil

Den downloadede scriptfil kan åbnes med et valgfrit tekst redigerings værktøj som kan læse UNICODE UTF-8. I filen findes alle relationel tabeller og alt test data.

For at få en database op at køre med tabellerne og dataene, skal følgende fremgangsmåde følges:

- Der oprettes en ny database, for eksempel ved brug af [elephantsql.com](http://elephantsql.com)
- Et databaseadministrationsprogram som kan køre PostgreSQL SQL benyttes, for eksempel pgAdmin, eller elephantsql egen "browser" ude i venstre side.



- Alt indholdet fra script filen bliver kopieret og indsat i databaseadministrationsprogrammets SQL-værktøj.
- Det køres
- Databasen indeholder nu alle tabeller og alt data som er blevet brugt som test data i dette projekt

## 17 Projektlog

[GitHub Repository](#)

## 18 Interne bilag

Er projektforslaget medtaget?

Er Inceptionsdokument medtaget? Inceptionsdokumentet skal afleveres i den udgave det blev bedømt.

Er der et udfyldt rapportkontrolskema

Er der andre relevante interne bilag, dvs. materialer produceret af gruppen selv?

### 18.1 Rapportkontrolskema

A. Produktrapport		
Kapitel	Krav	Opfyldt +/-
<b>Omslag</b>	Indeholder omslaget projekttitel, uddannelsesinstitution, fakultet, institut, uddannelse, semester, kursuskode, projektperiode, vejleder, projektgruppe og projektdeltagere (fornavn, efternavn, sdu-email)?	+
<b>Titelblad</b>	(Som omslag ekskl. evt. illustration + evt. kildehenvisning til evt. omslagsillustration. Omslaget kan udgøre både omslag og titelblad. Hvis der medtages selvstændigt titelblad, så er titelbladet rapportens første højre side)	-
<b>Resumé</b>	Omfatter resuméet: <ul style="list-style-type: none"> <li>• Den behandlede problemstilling - hvad blev der arbejdet med og hvorfor?</li> <li>• Fremgangsmåden - anvendte metoder - hvordan blev der arbejdet med det? (hvordan angreb I problemet og hvordan realiserede I løsningen (hvem, hvad, hvornår og hvorfor)</li> <li>• Hovedresultater og konklusioner – hvad kom der ud af arbejdet?</li> </ul> (max 1 side)	+
<b>Forord</b>	Indeholder forord hensigten med rapporten, målgruppe, forhistorie, anerkendelser, afleveringsdato samt underskrifter af alle projektdeltagere? Bemærk: Projektdeltagernes aktive deltagelse i projektførløbet anerkendes gensidigt ved projektdeltagernes underskrifter i rapporten.	+
<b>Indholdsfortegnelse</b>	Er der en samlet indholdsfortegnelse for hele projektrapporten <b>med to eller tre niveauer?</b> <b>Er afsnittene nummererede?</b>	+
<b>Læsevejledning</b>	Er der en vejledning i, hvordan rapporten kan læses, eksempelvis i form af hvilken rækkefølge afsnittene kan læses i og hvordan sammenhængen er mellem de forskellige dele af rapporten, fx mellem produktrapport og bilag?	+
	Er rapportens målgruppe beskrevet?	+



Redaktionelt	Beskriver redaktionelt skriveprocessen og ansvarsområder i skriveprocessen?	+		
	Ansvarsområder kan fx beskrives på fx følgende form:			
	Afsnit	Ansvarlig	Bidrag fra	Kontrolleret af
	Afsnit a	Person a	Person b	Person a, b, c
	Afsnit b	Person b	Person a	Person a, b, c
Afsnit c	Person c	Person b	Person a, b, c	
Ordliste	Er der en kort beskrivelse af de fagtermer der bruges gennem rapporten?	+		
Indledning	Giver indledningen et overblik over projektet og baggrunden for det?	+		
	Giver indledningen et resume af den udleverede case?	+		
	Indeholder indledningen problemformuleringen, jfr. inceptiondokumentet?	+		
	Beskriver indledningen formålet med projektet? Er formålet i overensstemmelse med hensigten med 2. semester?	+		
	Beskriver indledningen målene med projektet? Udtrykker målene specifikke, målbare resultater, jfr. inceptionsdokumentet? Er målene i overensstemmelse med de overordnede mål for 2. semester som udtrykt i studieordningens kap. 9 og de mere specifikke mål for projektet, som udtrykt i fagbeskrivelsen for SI2-PRO?	+		
Metode		+		
1. Indledning	Giver indledningen en introduktion til afsnittet?	+		
2. Metode	Er metoden i det samlede projekt beskrevet? Er det beskrevet hvordan UP og Scrum kombineres i projektet, samt hvilke fordele og ulemper der er ved det?	+		
3. Planlægning	Er planlægningen af elaborationsfasen og de enkelte iterationer beskrevet. Er backlogs beskrevet? Er rollefordelingen i projektgruppen beskrevet? Er ceremonierne beskrevet? Er scrum-buts beskrevet? Bygger planen på prioriteringen af kravene efter inceptionsfasen.	+		
Hovedtekst	Hvis der indgår et teoretisk problem: Omfatter hovedteksten et teori-afsnit?	+		
1. Indledning	Indeholder indledningen en overordnet introduktion til afsnittet?	+		
2. Overordnet kravspecifikation (resume, opdateret)	Indeholder afsnittet et opdateret resumé af systemafgrænsningen fra inceptionsdokumentet, herunder det samlede overordnede brugsmønstediagram.	+		

<b>3. Detaljeret kravspecifikation</b>	Omfatter den detaljerede kravspecifikation <ul style="list-style-type: none"> <li>• Detaljeret brugsmønsterdiagram (hvis relevant)</li> <li>• Detaljerede brugsmønsterbeskrivelser</li> <li>• Detaljerede beskrivelser af supplerende krav fx organiseret efter FURPS+</li> </ul>	+
<b>4. Analyse</b>	Omfatter afsnittet overvejelser, beslutninger og resultater vedr. <ul style="list-style-type: none"> <li>• Den statiske side af analysemodel</li> <li>• Den dynamiske side af analysemodel</li> </ul>	+
<b>5. Design</b>	Omfatter afsnittet overvejelser, beslutninger og resultater vedr. <ul style="list-style-type: none"> <li>• Softwarearkitektur</li> <li>• Subsystemdesign</li> <li>• Den statiske side af designmodel</li> <li>• Den dynamiske side af designmodel</li> <li>• Design af persistens</li> <li>• Databasedesign</li> </ul>	+
<b>6. Implementering</b>	Omfatter afsnittet overvejelser, beslutninger og resultater vedr. konvertering fra design til kode illustreret gennem udvalgte centrale eksempler, samt andre vigtige implementeringsbeslutninger Omfatter afsnittet implementering af database.	+
<b>7. Test</b>	Omfatter afsnittet en beskrivelse af de udførte test samt resultatet af dem.	+
	Er der medtaget resultater både fra iteration #1 og fra iteration #2?	+
<b>Diskussion</b>	Omfatter diskussionen hvad der er opnået, og hvad der ikke er opnået i projektet i forhold til det forventede som beskrevet i indledningen. Hvad er styrkerne og svaghederne ved jeres resultater? Kunne I have opnået bedre resultater?	+
<b>Konklusion</b>	Opsummerer konklusionen resultaterne og diskussionen af dem og giver det på problemformuleringen?	+
<b>Perspektivering</b>	Fremtidigt arbejde: Hvad ville de næste skridt i projektet være, hvis der var mere tid? Refleksion: Hvordan ville I gribe projektet an, hvis I skulle starte forfra?	+
<b>Litteraturliste</b>	Er litteratur angivet på en anerkendt form? Er alle former for litteratur som bøger, artikler og hjemmesider medtaget? Er der kildehenvisninger i teksten? Materiale som gruppen ikke selv har fremstillet i dette projekt skal være angivet med kilde! Er alle kildehenvisninger i teksten anført på samme måde? Er der kildeangivelser på figurer, grafer etc. som projektgruppen ikke selv har frembragt?	+
<b>B. Procesrapport</b>		
<b>Kapitel</b>	<b>Krav</b>	<b>Opfyldt +/-</b>
<b>Læring og refleksion</b>	Er der en redegørelse for læring og refleksion?	+

<b>Projektstyring</b>	Er der en redegørelse for <b>projektplanlægning og styring</b> ? Er der en beskrivelse af den faktiske <b>projektplanlægning og styring</b> , med inddragelse af væsentlige artefakter, som fx sprint backlogs, væsentlige ceremonier, som fx daily scrum, rollevaretagelse mm?	+
<b>Identifikation, analyse og bearbejdning af problemer</b>	Er der en redegørelse for identifikation, analyse og bearbejdning af problemer	+
<b>Udviklingsprocessen</b>	Er der en redegørelse for det faktiske udviklingsarbejde. Er der en beskrivelse af faserne, iterationerne og det faktiske arbejde i dem?	+
<b>Formidling og kommunikation</b>	Er der en redegørelse for formidling og kommunikation	-
<b>Samarbejde i gruppen</b>	Er der en redegørelse for samarbejde i gruppen	+
<b>Samarbejde med vejleder</b>	Er der en redegørelse for samarbejde med vejleder	+
<b>C. Kildekode</b>		
<b>Kapitel</b>	<b>Krav</b>	<b>Opfyldt +/-</b>
<b>Oversigt over projektets kildekode</b>	Er der en oversigt over projektets kildekode, fx filstruktur eller javadoc?	+
<b>D. Brugervejledning</b>		
<b>Kapitel</b>	<b>Krav</b>	
<b>Brugervejledning</b>	Er der en <u>kortfattet</u> brugervejledning?	+
<b>E. Projektlog</b>		
<b>Kapitel</b>	<b>Krav</b>	
<b>Projektlog</b>	Er der en udskrift af eller en adresse på og et link til projektloggen i Github	+
<b>F. Interne bilag</b>		
<b>Kapitel</b>	<b>Krav</b>	<b>Opfyldt +/-</b>
<b>Projektforslag</b>	Er projektforslaget medtaget?	+
<b>Inceptionsdokument</b>	Er Inceptionsdokumentet medtaget? <b>Inceptionsdokumentet skal afleveres i den udgave det blev bedømt.</b>	
<b>Rapportkontrolskema</b>	Er der et udfyldt rapportkontrolskema	+
<b>Andet</b>	Er der andre relevante interne bilag, dvs. materialer produceret af gruppen selv?	+
<b>G. Eksterne bilag</b>		
<b>Kapitel</b>	<b>Krav</b>	<b>Opfyldt +/-</b>

<b>Eksterne bilag</b>	Er der medtaget relevante eksterne bilag, dvs. materialer som gruppen ikke selv har produceret med som er nødvendige for at kunne læse rapporten?	-
-----------------------	---	---

<b>Rapporttekniske elementer</b>		<b>Opfyldt +/-</b>
<b>Layout</b>	Er der anvendt samme layout i alle kapitler? Er layout overskueligt/harmonisk?	+
<b>Sprog</b>	Er rapporten skrevet i en neutral sprog tone? Er sproget let læseligt og flydende? Er der udført stavekontrol og kontrol af tegnsætning?	+
<b>Sidenummerering</b>	Er der korrekt og konsistent sidenummerering i rapporten?	+
<b>Figurer/diagrammer</b>	Er alle figurer konsekvent nummererede? Er der figurtitel og figurtekst til alle figurer? Er figurtitler og figurtekster dækkende og afklarende? Er figurerne tydelige og læsbare? Er figurerne informationsgivende og i den rette sammenhæng?	(+)
<b>Tabeller</b>	Er alle tabeller konsekvent nummererede? Er der en forklarende tabeltekst til alle tabeller? Er alle søjler og rækker forsynet med parametre? Er der enheder på alle relevante rækker og søjler?	-
<b>Sporbarhed af begreber</b>	Er der en konsekvent brug af samme betegnelse for et givet begreb igennem rapporten?	(+)

## 18.2 Inceptionsdokument

## SSB Sager – Inceptionsdokument

The screenshot shows a web application interface for 'SSB Sager'. On the left is a blue sidebar with navigation links: Sager, Dagbog, Handleplan, Opgaver, Annoncer, Kalender, and Statistik. At the top of the sidebar is a home icon and a search bar. The main content area has a search bar at the top with the text 'Søg, eller skriv en kommando'. Below the search bar are three buttons: 'Vælg beboer' (dropdown), 'Søg efter dokument' (search), and '+ Opret VUM-dokument'. The main area displays a table with the following columns: Navn, Beboer, Type, Ændringsdato, and Oprettelsesdato. The table contains 15 rows of data, all with 'Lorem ipsum lorem' as the name and 'Lars larsen' as the resident. The type is 'Sagsåbning' and the dates are '23-03-2018' and '01-03-1998'. At the bottom of the table are navigation arrows and page numbers 1 through 5.

Navn	Beboer	Type	Ændringsdato	Oprettelsesdato
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998

<b>Projektfase:</b>	<i>Inceptionsfasen</i>
<b>Faseperiode:</b>	25-02-2019 / 22-03-2019
<b>Vejleder:</b>	Jeppe Schmidt ( <a href="mailto:jpe@mmmi.sdu.dk">jpe@mmmi.sdu.dk</a> )
<b>Projektgruppe:</b>	Gruppe 25
<i>Morten Krogh Jensen</i>	<a href="mailto:mortj18@student.sdu.dk">mortj18@student.sdu.dk</a>
<i>Nicolai Ankjær Kruuse</i>	<a href="mailto:nikru18@student.sdu.dk">nikru18@student.sdu.dk</a>
<i>Thomas Steinfeldt Laursen</i>	<a href="mailto:tlaur18@student.sdu.dk">tlaur18@student.sdu.dk</a>
<i>Mathias Haurum Løyché</i>	<a href="mailto:maloe18@student.sdu.dk">maloe18@student.sdu.dk</a>
<i>Michael Haugaard Pedersen</i>	<a href="mailto:micpe18@student.sdu.dk">micpe18@student.sdu.dk</a>
<i>Oliver Marco van Komen</i>	<a href="mailto:olvan18@student.sdu.dk">olvan18@student.sdu.dk</a>
<b>Afleveringsdato:</b>	22-03-2019

**Indhold**

2	Indledning.....	63
3	Fremgangsmåde.....	65
4	Business case.....	66
5	Forretningsområdet og eksisterende løsninger .....	67
5.1.1	Forretningsområde:.....	67
5.1.2	Eksisterende løsninger:.....	67
6	Overordnet kravspecifikation .....	68
6.1	Beskrivelse af afgrænsning.....	68
6.2	Brugsmønsterdiagrammer .....	69
6.3	Aktørbeskrivelser .....	71
6.4	Korte brugsmønstre beskrivelser .....	71
6.5	Detaljeret beskrivelse af essentielle brugsmønstre .....	73
6.6	Supplerende krav .....	75
6.7	Domænemodel.....	77
7	Kritiske risici .....	78
8	Tekniske gennemførlighed .....	80
9	Prioritering.....	81
9.1	Prioritering med MoSCoW.....	85
10	Metoder i elaborationsfasen.....	86
10.1	Undersøgelsesmetoden .....	86
10.2	Konstruktionsmetoden.....	86
10.3	RUP/UP .....	86
10.4	Scrum.....	86
11	Ressourcer .....	88
12	Konklusion .....	89
13	Bilag .....	<b>Fejl! Bogmærke er ikke defineret.</b>
13.1	Bilag 1 - Link til logbog .....	<b>Fejl! Bogmærke er ikke defineret.</b>
13.2	Bilag 2 - Business case .....	<b>Fejl! Bogmærke er ikke defineret.</b>
13.3	Bilag 3 - Yderligere korte brugsmønsterbeskrivelser .....	<b>Fejl! Bogmærke er ikke defineret.</b>

## Indledning

EG Team Online leverede i 2005 deres første store system, som nu hedder Sensum Bosted, til lokale bosteder og institutioner og har siden da opnået dyb forståelse for de udfordringer, som bosteder og sagsbehandlere står overfor. Siden da, er der kommet to store udfordringer. En økonomiaftale fra 2013 bestemte at it-løsninger for en række offentlige områder, heriblandt sagsoverblik, skulle udvikles som landsdækkende løsninger, hvor alle kommuner har indflydelse på løsningerne. Derudover blev der i 2016 vedtaget en ny EU-databeskyttelsesforordning, som har betydet ændringer i, hvordan personfølsomme data skal håndteres.

For EG Team Online betyder det, at deres produkter skal samles til en landsdækkende løsning for alle bosteder i Danmark, og at socialfaglige medarbejdere kun ser nødvendige data på patienter, som har afgivet samtykke til det. I dette projekt vil der derfor blive arbejdet med sagsudredning.

Interessenter for denne løsning vil være:

	Stor indflydelse	Lille indflydelse
Bliver påvirket af projektet	Kommunen, medarbejdere	Beboere, pårørende
Bliver ikke påvirket af projektet	Investorer i EG, ledelse i EG	Politikere (lovgivning)

Når en borger er under udredning på et bosted, bliver der gemt data ved brug af VUM-redskabs dokumenter. Borgerens data er følsomt og samlingen af alle IT-løsninger for bosteder under et system kræver yderligere sikkerhedsforbehold mht. dataopbevaring. Dataene må ikke kunne tilgås af andre end borgeren selv og af relevant sundhedsfagligt personale, hvortil borgeren har givet samtykke. Dette betyder, at når sundhedsfaglige medarbejdere der arbejder med patienten, skal have adgang til nødvendig information, skal der kun vises de informationer, der er behov for, og som borgeren har givet samtykke til. Dette kan være svært, da dataene både skal gemmes væk, så de ikke kan tilgås, men samtidigt være tilgængelig for de rette personer. Det kræver en høj mængde autentificering af personen, som vil tilgå patientens data. Det kan hurtigt blive komplekst at udvikle et sådant system, og der er risiko for, at den øgede mængde af datasikkerhed går på kompromis med brugervenligheden i den daglige brug af den samlede IT-løsning for alle bosteder. Denne problematik leder derfor til hovedspørgsmålet:

Kan alle VUM-dokumenter fra alle bosteder i hele Danmark samles i én data-løsning, hvor data-lovgivning bliver overholdt og brugervenlighed af systemet er i fokus?

For at kunne besvare hovedspørgsmålet skal følgende spørgsmål besvares igennem projektforløbet:

- Kan der designes en database, som er hurtig nok til at hente data til en travl hverdag på bosteder? (Indexing, BCNF, datasikkerhed osv.)
- Er der anerkendte databasearkitekturer, som kan bruges til opbevaringen af patientdata? (allerede eksisterende løsninger)
- Kan det færdige system opnå et højt tilfredsheds-niveau (over 75% af test-brugere kan lide systemet) ved brugervenligheds testning?
- Vil systemet blive langsommere igennem tiden ved ophobning af store mængder data? (Simulering af ophobning af data igennem årene)
- Er det muligt at gøre brugeres rettigheder afhængig af deres arbejde?
- Er det muligt at gøre de ansattes rettigheder afhængig af, hvilke beboere de er tilknyttet?
- Kan man opbygge systemet af moduler, som er uafhængige af hinanden til en sådan grad, at hvert modul kan fjernes uden resten af systemet, bliver ubrugeligt?

- Kan vi indenfor projektets tidsramme opnå tilstrækkelig viden om datasikkerhed?
- Kan vi gøre en digitaliseret brug af VUM-Dokumenter mere effektiv end alternativerne?
- Har den danske lovgivning love som tilsidesætter GDPR-love som påvirker vores system?

For at besvare hovedspørgsmålet og underspørgsmålene bliver Rational Unified Process procesmodel brugt, hvor den første fase er inceptionsfasen. Målet i denne fase er at få skudt projektet i gang med et godt grundlag og sikre, at projektet er muligt at gennemføre. Dette undersøges ved at lave prototyper, markedsundersøgelser, dokumentere essentielle krav i en kravspecifikation og identificere risiko. En stor del af arbejdet ligger i kravspecifikationen, da der her skal udarbejdes brugsmønstre og domænemodeller, som skal danne fundament for det detaljerede design og arkitektur modellering i den næste fase, elaborationsfasen.



**Fremgangsmåde**

Inceptionsfasens varighed var på 4 uger i dette projektforsløb og fremgangsmåde var således:

**Uge 9**

- Brainstorm af forskellige udkast til systemet i form af layout-design
  - o Sketches
  - o Vigtige brugsmønstre
- Diskussion omkring nuværende og manglende viden
  - o Informationssøgning på nettet og i leveret case materiale
- Udarbejdelse af overordnet brugsmønstermodel
  - o Aktører og tilhørende brugsmønstre
- Forberedelse til virksomhedsmøde med EG Team Online efterfølgende uge
  - o Diskussion af vigtige spørgsmål, som vil bidrage gruppen værdifuld information
  - o Brug manglende viden som en faktor for hvilke spørgsmål, der skulle stilles

**Uge 10**

- Forberedelse til virksomhedsmøde med EG Team Online
  - o Forbedring og potentielle ændringer i spørgsmål
- Deltagelse i mødet
  - o Referent nedskriver mødet
- Begyndelse på krav til systemet
  - o Fokus på hvordan brugeren vil anvende systemet

**Uge 11**

- Gennemgang af virksomhedsmødet
- Arbejde med nye sketches og brugsmønstre efter ny information fra EG Team Online
- Analyseret miljøet ved brug af Porters Five Forces og PESTEL (Business case)
- Uddybelse af problemstillingen til et beskrivende problemafsnit
- Lavet brugsmønsterdiagrammer
- Lavet et mock-up af systemet

**Uge 12**

- Færdiggørelse af korte- og detaljerede brugsmønster beskrivelser
- Udarbejdet domænemodel
  - o Brugt navneord/udsagnsord fra detaljeret brugsmønster beskrivelserne
- Lavet resumé af business cases fra uge 11
- Udarbejdet supplerende krav
- Færdiggørelse og rettelse af inceptionsdokumentet

**Business case**

I det sideløbende fag, Organisation og Ledelse, har gruppen foretaget en analyse af det virksomhedsmiljø, der omgiver organisationer som EG Team Online. Både det generelle og specifikke miljø blev analyseret og vurderet, hvorefter det optimale design af organisationer i denne branche kunne bestemmes ud fra The Contingency Approach.

EG Team Online opererer i Danmark, hvor det generelle miljø er forholdsvis stabilt. Her vil en IT-virksomhed trives på grund af fordele som lav selskabsskat, højt uddannelsesniveau og fair konkurrence.

Igennem en analyse af Porters Five Forces, har vi dannet os et indblik i branchens specifikke miljø. I denne branche er truslen for nye indtrængere høj, da det er billigt at starte op, brand ikke har en stor betydning. Miljøet kan derfor ses som en smule ustabil på dette punkt. Leverandører har lav betydning for et firma som EG Team Online, og bidrager derfor ikke til et ustabil miljø, grundet at deres eneste leverandører er f.eks. til deres servere. Kundernes forhandlingskraft har kun en smule indflydelse på EG Team Online, da kunden er underlagt en række strenge love og at kunden altid vil være kommuner og ikke almindelige borgere. Til gengæld vurderes truslen for erstattende produkter til at være meget lav. Konkurrencen mellem virksomheder i branchen er ikke lige så ustabil, som den kunne være. Noget der øger konkurrencen, er EG Team Onlines konkurrenter som Systematic og KMD. Disse er henholdsvis lige så store eller større end EG Team Online, og kan derfor være magtfulde i denne branche. Til gengæld har de ikke lige så stor direkte erfaring med sundhed og socialområdet.

Analyserne viser, at det generelle miljø der omgiver EG Team Online er meget stabilt. I et sådant miljø, kan man uden problemer designe sin virksomhed mekanistisk. Dog skal man huske at overveje de få faktorer der bidrager til et ustabil miljø. Ifølge analysen af branchens specifikke miljø, er det hverken stabilt eller ustabil og ligger et sted i midten. Dette taler for, at EG Team Online burde, ifølge the Contingency Approach, designe sin virksomhed med en overvejende mekanistisk struktur med elementer af en organisk struktur.

**Forretningsområdet og eksisterende løsninger****Forretningsområde:**

Sensum Bosted er et system produceret til den sociale sektor, som skal have fokus på at effektivisere og gøre det nemmere at gennemføre borgernes sagsbehandling. Det vil sige, at EG Team Online befinder sig i den sociale sektor, hvilket er i den kommunale sektor, med deres produkt. At produktet skal leveres til den kommunale sektor, betyder at EG Team Online skal konkurrere med organisationer på samme marked og forholde sig til det kommunale miljø. Den nuværende løsning produceret af EG Team Online er Sensum Bosted og Sensum, som sammen forbinder arbejdet og data fra sagsbehandlere, borgere, socialrådgivere og socialpædagoger. Hensigten med projektførelsen er her at samle disse systemer, for at effektivisere denne proces yderligere for at få en stabil og forsat plads i det danske forretningsområde i den kommunale sektor.

**Eksisterende løsninger:**

Der findes i forvejen forskellige eksisterende løsningsmuligheder inde for den sektor EG Team Online arbejder i. Disse løsningsmuligheder stammer fra virksomheder som f.eks. KMD, NetCompany og Systematic. En difference mellem disse virksomheder og EG Team Online er fokusområdet, som EG Team Online specialiserer sig i, hvilket er at servicere kommunerne med helhedsløsninger for et enkelt område, frem for specificerede produkter til enkelte arbejdsområder eller helhedsløsninger for mange arbejdsområder. Der kan her bl.a. tages udgangspunkt i KMD Nexus. Nexus er en helhedsløsning, som dækker mange forskellige kommunale behov i et enkelt system. EG Team Online formår dog, ved at fokusere deres system på det specifikke område, at opnå en mere fyldestgørende løsning, som er specielt udformet til arbejdsmetoden i det arbejdsmiljø.

**Overordnet kravspecifikation****Beskrivelse af afgrænsning**

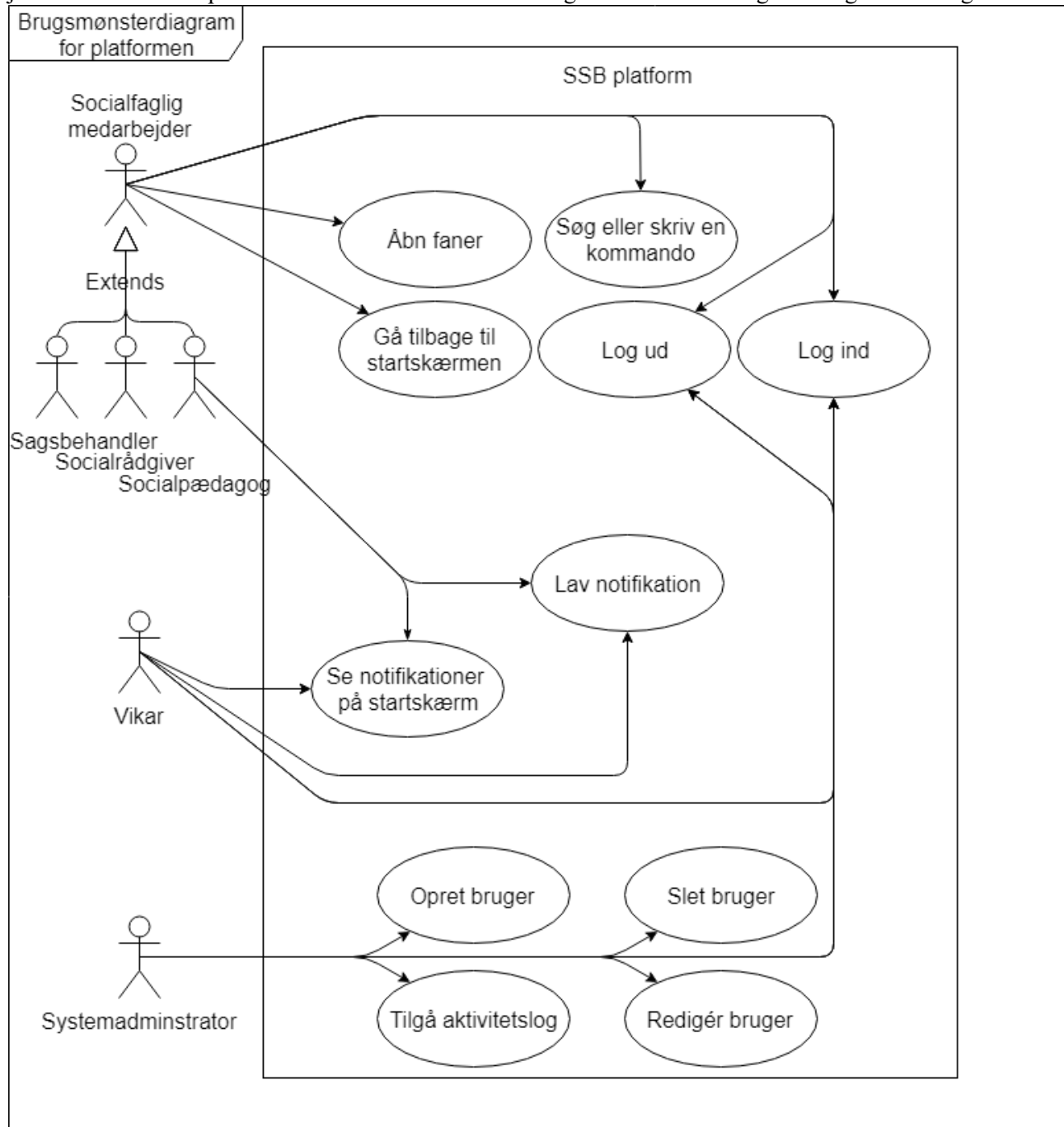
Brugsmønstrene er blevet delt op i to:

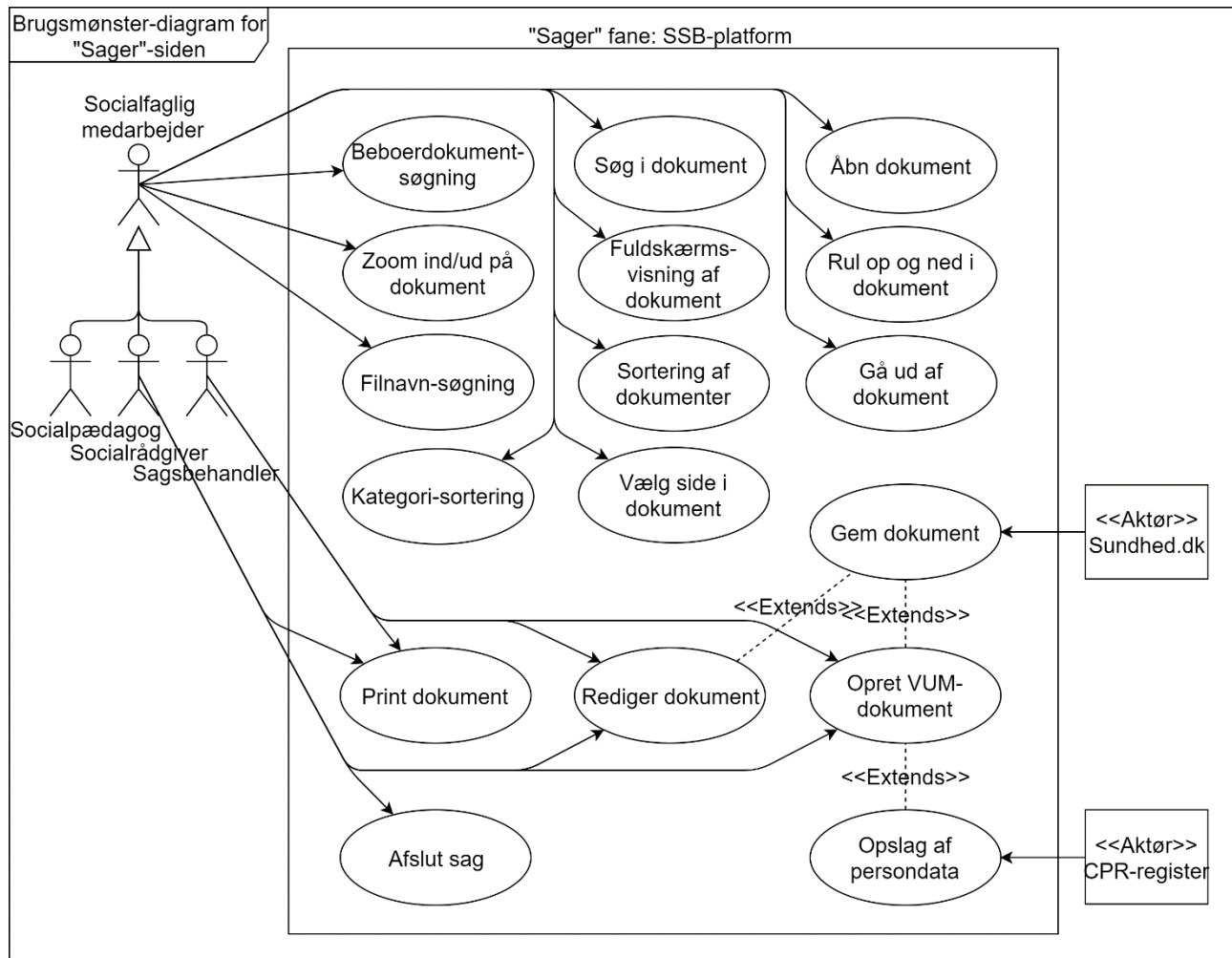
1. Platformen for SSB (Sensum Sensum Bosted)
2. Sagsudredningsfane som eksisterer indenfor SSB

Den overordnede platform repræsenterer hele det samlede system. Herfra er man i stand til at tilgå alle systemets moduler fra både Sensum og Sensum Bosted. Ét af disse moduler er sagsudredningsmodulet som er fokusområdet i dette projekt. Hertil har vi lavet endnu et brugsmønsterdiagram hvor subjektet er "sager-siden". Fra denne afdeling af systemet skal brugere være i stand til at håndtere alt der har med sagsudredning og VUM-dokumenter at gøre. Denne opdeling af brugsmønsterdiagrammerne er foretaget for at gøre modellen så overskuelig som mulig.

**Brugsmønsterdiagrammer**

Herunder ses brugsmønsterdiagrammerne for systemet SSB. Der er to forskellige. Det første diagram har subjektet "SSB platform". Det andet diagram her afgrænsningen "Sager-siden".





**Aktørbeskrivelser**

<b>Aktørliste</b>	
Sagsbehandler	Sagsbehandleren sidder ved kommunen og foretager sagsudredninger på beboere. De har blandt andet til opgave at finde ud af, hvordan beboerens behandlingsforløb skal foregå. De dokumenterer udredningsforløbet igennem dagbogsnotater. De skal have adgang, til relevant information om sager.
Socialpædagog	En socialpædagog er en person ude på bostederne som hjælper med det praktiske omkring beboerne. En socialpædagog skal have adgang til de relevante dokumenter for de borgere de har at gøre med. De skal ikke have adgang til information, fra andre bosteder. En socialpædagog kan fungere som socialpædagog på ét bosted, og vikar på et andet, hvor de har forskellige rettigheder hvert sted.
Socialrådgiver	En socialrådgiver er en person som hjælper med sagsudredningen for beboere. De befinder sig både ude på bosteder og hos kommunen. De har adgang til information fra deres bosted og arbejdsområde og arbejder med VUM.
Vikar	En vikar skal støtte socialpædagogen, men skal ikke have samme mængde rettigheder.
Kommunen	Alle afdelinger/bosteder skal videregive information til kommunen i form af overblik over processen til borgeren møder op til borgeren er færdig.
Systemadministrator	Bruger med eksklusiv adgang til at oprette brugere, give privilegeret adgang til andre brugere.
CPR-registeret	Dette er et eksternt system, som systemet anvender til at få information om beboere, når en ny sag oprettes.
Sundhed.dk	Dette eksterne system gemmer data og dokumenter til hver gang nye VUM-dokumenter oprettes og redigeres. Beboere skal have mulighed for at tilgå personlige dokumenter omkring sine sagsudredninger på denne hjemmeside.

**Korte brugsmønstre beskrivelser**

Da der er mange brugsmønstre, bliver kun de vigtigste udvalgt som også vil blive prioriteret i et senere afsnit.

<b>Log ind</b>
<b>ID:</b> 1
<b>Formål:</b> Bruger skal have adgang til systemet
<b>Detaljeret beskrivelse:</b> Brugeren møder en login-skærm. Brugeren skal her indtaste deres generelle adgangskode til at få adgang til ikke kritiske funktioner, såsom notifikationer og at bevæge sig rundt i systemet. Hvis brugeren senere vil se personlig information såsom at åbne VUM-dokumenter skal de indtaste en mere specifik adgangskode. Ved indtastning af korrekt login-information videresendes brugeren til startskærmen.
<b>Aktører:</b> Sagsbehandler, socialpædagog, socialrådgiver, vikar, systemadministrator (Alle)

<b>Lav notifikation</b>
<b>ID:</b> 2
<b>Formål:</b> Brugeren kan oprette en ny notifikation til andre medarbejder om hverdags relaterede
<b>Detaljeret beskrivelse:</b> Brugeren kan oprette en ny notifikation for at kommunikere med andre medarbejdere. Det kunne for eksempel være en notifikation om, at andre medarbejdere skal være ekstra opmærksomme på en beboer den dag eller lignende. Andre brugere vil ude på startskærmen kunne se disse notifikationer og se hvem der har

skrevet dem og hvornår den er skrevet. Efter at brugeren har stået på startskærmen i et stykke tid, vil alle de nuværende notifikationer på skærmen blive markeret automatisk til "læst". Medarbejdere kan derfor nemt få sig et overblik over hvilke notifikationer der er gamle og nye. Derudover har medarbejdere også mulighed for at slette deres egne notifikationer.
<b>Aktører:</b> Sagsbehandler, socialrådgiver, socialpædagog, vikar.

<b>Opret VUM-dokument</b>
<b>ID:</b> 3
<b>Formål:</b> Opret et af ni VUM-dokumenter.
<b>Detaljeret beskrivelse:</b> Oprettelse af et bestemt VUM-dokument knyttet til en ny eller eksisterende borger. Brugeren vælger selv hvilket VUM-dokument der skal oprettes og bliver præsenteret en passende skabelon som skal udfyldes med information, hvor CPR-registeret bl.a. automatisk indhenter information og udfylder relevante felter i skabelonen. Sagsbehandlere må kun oprette sagsåbnings dokumenter, hvorimod socialrådgiver må oprette alle andre VUM-dokumenter end sagsåbnings dokumenter. Når brugeren skal gemme dokumenter, vil et tjek køre igennem for at tjekke om alle vigtige felter er udfyldt.
<b>Aktører:</b> Sagsbehandler, Socialrådgiver.

<b>Redigere dokument</b>
<b>ID:</b> 4
<b>Formål:</b> Brugeren kan redigere information fra VUM-dokument i tilfælde af opdatering af en sag eller ny relevant information.
<b>Detaljeret beskrivelse:</b> Når brugeren befinder sig i dokument-oversigten og har fremhævet et dokument, vil det være muligt for dem at klikke på en redigerings knap, hvorefter dokumentet vil åbne op og alle felter som kan redigeres, vil være tilgængelig for brugeren til at redigere. Når dokumentet skal gemmes, vil der igen tjekkes for om de redigere felter er udfyldt korrekt.
<b>Aktører:</b> Sagsbehandler, socialrådgiver.

<b>Afslut sag</b>
<b>ID:</b> 5
<b>Formål:</b> En sag kan afsluttes når borgeren er færdig med udredningen.
<b>Detaljeret beskrivelse:</b> Når en borger er færdigbehandlet, skal medarbejderen der har været tilknyttet sagen ikke længere have mulighed for at oprette nye dokumenter på beboeren eller se og redigere beboernes dokumenter.
<b>Aktører:</b> Socialrådgiver

<b>Åbn dokument</b>
<b>ID:</b> 6
<b>Formål:</b> At kunne få adgang til alle informationer for et VUM-dokument
<b>Detaljeret beskrivelse:</b>



Ved at åbne et dokument, åbnes dokumentet i et vindue og giver adgang til dokumentets information.
--

**Aktører:**

Socialpædagog, Socialrådgiver, Sagsbehandler
--

**Log ud****ID:** 7**Formål:**

Brugeren kan manuelt logge ud, men systemet vil også automatisk logge brugeren ud ved inaktivitet.
--

**Detaljeret beskrivelse:**

Når en bruger ikke skal bruge systemet mere, skal de logge ud for at undgå at brugerens rolle og ID kan udnyttes af uvedkommende, når brugeren ikke er tilstede ved computeren. Hvis en bruger glemmer at logge ud, skal systemet automatisk kunne logge ud for brugeren.
---

**Aktører:**

Sagsbehandler, socialrådgiver, socialpædagog, vikar.
--

**Detaljeret beskrivelse af essentielle brugsmønstre****Opret VUM-dokument****ID:** 3**Primære aktører:**

Sagsbehandler, Socialrådgiver
-------------------------------

**Sekundære aktører:**

Sundhed.dk, CPR-registeret
----------------------------

**Kort beskrivelse:**

Oprettelse af et bestemt VUM-dokument knyttet til en ny eller eksisterende borger. Brugeren vælger selv hvilket VUM-dokument der skal oprettes og bliver præsenteret for en passende skabelon som skal udfyldes med information, hvor CPR-registeret bl.a. automatisk indhenter information og udfylder den information. Sagsbehandlere må kun oprette sagsåbnings dokumenter, hvorimod socialrådgiver må oprette alle andre VUM-dokumenter end sagsåbnings dokumenter.
---

**Prækonditioner:**

- |  |
|--|
| 2. Brugeren der er logget på, er inde på "sager"-fanen i SSB-plattformen |
|--|

**Hovedhændelsesforløb:**

- |   |
|---|
| <ol style="list-style-type: none"> <li>7. Brugsmønstret begynder når brugeren klikker på "opret dokument" knappen</li> <li>8. <b>Hvis</b> brugeren har rollen sagsbehandler <ol style="list-style-type: none"> <li>8.1. Brugeren kan oprette et nyt sagsåbningsdokument</li> <li>8.2. Det første felt i dokumentet skal udfyldes med beboernes CPR-nr.</li> <li>8.3. Udvidelsespunkt: Opslag af persondata</li> <li>8.4. Specifikke oplysninger om sagen udfyldes af brugeren</li> <li>8.5. Brugeren vælger bosted som borgeren skal sendes til <ol style="list-style-type: none"> <li>8.5.1. Brugeren bliver præsenteret med mængden af beboer på det valgte bosted</li> </ol> </li> </ol> </li> <li>9. <b>Hvis</b> brugeren har rollen socialrådgiver <ol style="list-style-type: none"> <li>9.1. Brugeren får mulighed for at oprette et af de 9 VUM redskabs-dokumenter med undtagelse af sagsåbnings dokument</li> <li>9.2. Brugeren vælger dokument der skal oprettes</li> <li>9.3. Brugeren ser en oversigt over beboere de er associeret med som skal tilknyttes det nye dokument</li> <li>9.4. Allerede eksisterende data bliver automatisk udfyldt i det nye dokument. (Dette indbefatter navn, adresse, CPR-nummer osv.)</li> </ol> </li> <li>10. Det for-udfyldte dokument åbner op hvor brugeren skal udfylde relevant information om beboeren</li> <li>11. Udvidelsespunkt: Gem dokument</li> </ol> |
|---|

12. Systemet giver en bekræftelse på at dokumentet er blevet gemt succesfuldt
<b>Post-konditioner:</b> <ol style="list-style-type: none"> <li>Der er oprettet et VUM-dokument af den type som brugeren har valgt</li> <li>VUM-dokumentet kan ses i oversigten og alle dokumenter</li> <li>Hvis der blev oprettet en ny beboer, skal denne beboer være tilknyttet til de rette socialpædagoger og socialrådgivere</li> </ol>
<b>Alternative hændelsesforløb:</b> <ul style="list-style-type: none"> <li>Overalt: Annullering</li> </ul>

Udvidelsesbrugsmønster: Gem dokument
<b>ID:</b> 8
<b>Primære aktører:</b> Sagsbehandler, social rådgiver
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Efter udfyldelse af VUM-dokument skal det gemmes i en database og vises i dokumentoversigten for alle brugere af systemet
<b>Prækonditioner:</b> <ol style="list-style-type: none"> <li>Brugeren er i gang med at oprette eller redigere et VUM-dokument</li> </ol>
<b>Hovedhændelsesforløb:</b> <ol style="list-style-type: none"> <li>Brugsmønsteret starter når brugeren vil gemme et dokument</li> <li>Systemet tjekker for om alle de nødvendige felter i dokumentet er blevet udfyldt korrekt <ol style="list-style-type: none"> <li><b>Hvis</b> felterne ikke er udfyldt korrekt, får brugeren en notits om at de skal udfylde dem korrekt</li> </ol> </li> <li>Systemet sender alt information ind til en database</li> <li>Systemet lukker dokumentet som brugeren havde åbent</li> <li>Systemet giver brugeren besked på at det er gemt korrekt</li> </ol>
<b>Post-konditioner:</b> <ol style="list-style-type: none"> <li>Dokumentet kan ses i oversigten</li> <li>Dokumentet kan hentes fra databasen</li> </ol>
<b>Alternative hændelsesforløb:</b> Ingen

Udvidelsesbrugsmønster: Opslag af persondata
<b>ID:</b> 9
<b>Primære aktører:</b> Sagsbehandler
<b>Sekundære aktører:</b> CPR-register
<b>Kort beskrivelse:</b> Under oprettelse af en ny sag skal sagen udfyldes med en række informationer som skal hentes fra CPR-registeret.
<b>Prækonditioner:</b> <ol style="list-style-type: none"> <li>Brugeren er i gang med oprettelsen af en sagsåbning</li> <li>Der er indtastet et gyldigt CPR-nummer i sagsåbningsdokumentet</li> </ol>

<b>Hovedhændelsesforløb:</b>
5. Brugsmønsteret starter efter sagsbehandleren har udfyldt CPR-nummeret ved oprettelse af en sags-åbning
6. Det indtastede CPR-nummer sendes til CPR-registeret
7. CPR-registeret sender den relevante information om beboeren tilbage til systemet
8. De hentede informationer fra CPR-registeret bliver udfyldt i dokumentet.
<b>Post-konditioner:</b>
2. Felter i dokumentet er udfyldt med alt relevant information fra CPR-registeret
<b>Alternative håndelsesforløb:</b>
Ingen

<b>Alternative flow: Annullering</b>
<b>ID:</b> 3.1
<b>Primære aktører:</b> Sagsbehandler, socialrådgiver
<b>Sekundære aktører:</b> Ingen
<b>Kort beskrivelse:</b> Igennem hele brugsmønsteret "Opret VUM-dokument" er det muligt at annullere oprettelsen af en ny sag.
<b>Prækonditioner:</b> 2. Et VUM-Dokument er blevet oprettet
<b>Hovedhændelsesforløb:</b>
5. Brugsmønsteret starter når brugeren trykker på knappen "Annuller" ved enten oprettelse eller redigering af et dokument.
6. Systemet spørger brugeren om de vil bekræfte annulleringen
7. <b>Hvis</b> brugeren bekræfter annulleringen
7.1. Alle ændringer og nyoprettede dokumenter kasseres
7.2. Brugeren sendes tilbage til "sager"-siden
8. <b>Hvis</b> brugeren annullerer annulleringen
8.1. Brugeren sendes tilbage til dokumentet og kan redigere videre
<b>Alternativt flow:</b> Ingen
<b>Post-konditioner:</b> 2. Indtastede oplysninger fra VUM-dokumentet er blevet slettet

## Supplerende krav

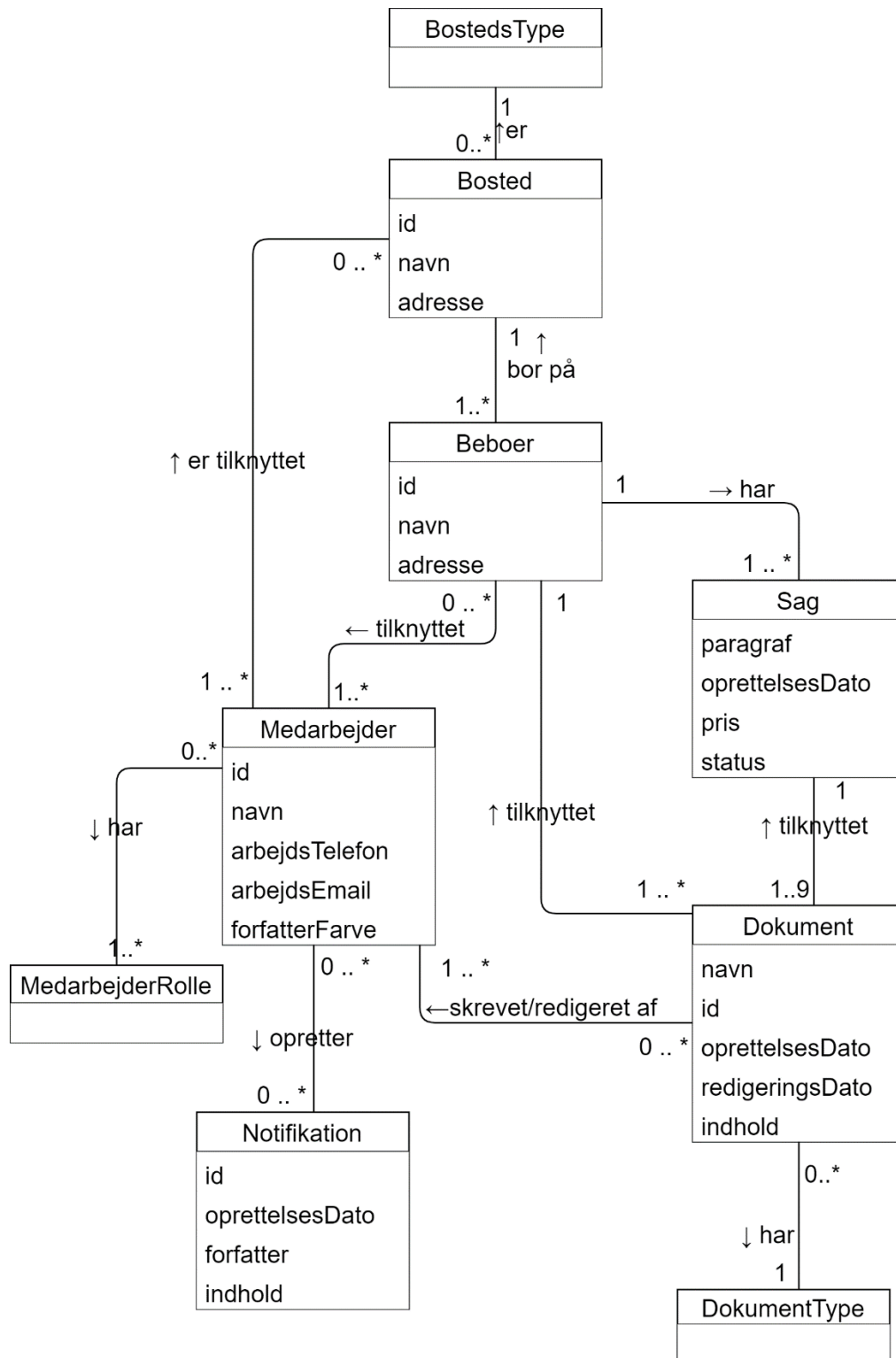
Ved at bruge FURPS+ opdeles de supplerende krav:

ID	Krav beskrivelse
F00	SSB skal logføre alle aktiviteter
F01	SSB skal være kompatibelt med Windows 7, Windows 8 & Windows 10
F02	SSB skal kunne udveksle data med CPR-registeret
F03	SSB skal kunne sende information til Sundhed.dk
F04	SSB skal fordele brugere på specifikke roller og ID for at afgrænse adgang til data
U00	SSB skal have en tilfredsstillende vurdering fra testbrugere
U01	SSB skal give brugeren mulighed for at gå tilbage til systemets forrige grafiske tilstand.
U02	SSB skal have en responstid på op til 1 sekund
U03	SSB skal synliggøre hvilke brugere, som har oprettet, redigere og slettet dokumenter.
U04	SSB skal have ensartet placering af gentagende knapper på alle sider

U05	SSB skal give brugeren adgang til navigationsbaren til venstre uanset hvor brugeren befinder sig
U06	SSB skal vise en dialogboks i tilfælde af at brugeren går væk fra arbejde der ikke er gemt
U07	SSB skal give brugeren besked på manglende udfyldte felter i tilfælde af de prøver at gemme et dokument der ikke er udfyldt i alle nødvendige felter
U08	SSB skal kunne være brugbart i forskellige størrelser (skalerbar)
R00	SSB skal stå til rådighed 97% af tiden
R01	SSB skal have en opdateringstid på max 2 timer
P00	SSB skal kunne håndtere 1750 samtidige logins
P01	SSB skal kunne håndtere 20GB data fra VUM-dokumenter
P02	SSB skal muliggøre at hente og sende af data på mindst 10 MB/sekundet
P03	SSB skal max bruge op til 2GB ram
S00	SSB skal inddeles i en plug and play samlekasse med moduler

**Domænemodel**

Herunder ses domæneklassediagrammet for vores system. Domænet er her en kombination af SSB-platformen og "Sager"-siden. Alle andre moduler i SSB er udeladt.



**Kritiske risici**

Risiko bliver analyseret på baggrund af at dette er et studieprojekt, der vil derfor ikke blive inddraget business risici, men kun risici som er relevante for et studieprojekt. Skemaet herunder giver en oversigt over hvilke kritiske risici projektet indeholder.

Risiko	Beskrivelse	Sandsynlighed	Konsekvens
Uretmæssigt adgang til system	Uvedkommende personer kan tilgå følsomme data på en computer der står logget ind og er autentificeret som en anden person	Moderat	Høj
Dårligt resultat af brugervenlighedstest	Gennem brugertest af brugervenligheden opnås der et dårligt resultat	Moderat	Lav
Tidsplanen skrider	Funktioner tager længere tid at implementere end forventet, og tidsplanen skrider derfor.	Moderat	Moderat
For lang tid at udarbejde arkitekturen for platformen	Arkitekturen tager for lang tid at implementere og resulterer derfor i for lidt tid til at lave de basale funktioner i platformen	Lav	Katastrofalt
Manglende muligheder i Java	Funktioner fra mock-up er ikke mulige at lave i Java eller JavaFX	Høj	Lav
Uventet information fra virksomhedsmøde	Information fra virksomhedsmøde slår løsning på problemet ud af kurs	Høj	Moderat
Gruppen mister et medlem	Et gruppemedlem enten dropper ud eller smidt ud	Moderat	Moderat

Skemaet herunder viser de strategier gruppen vil bruge imod de fundne risici.

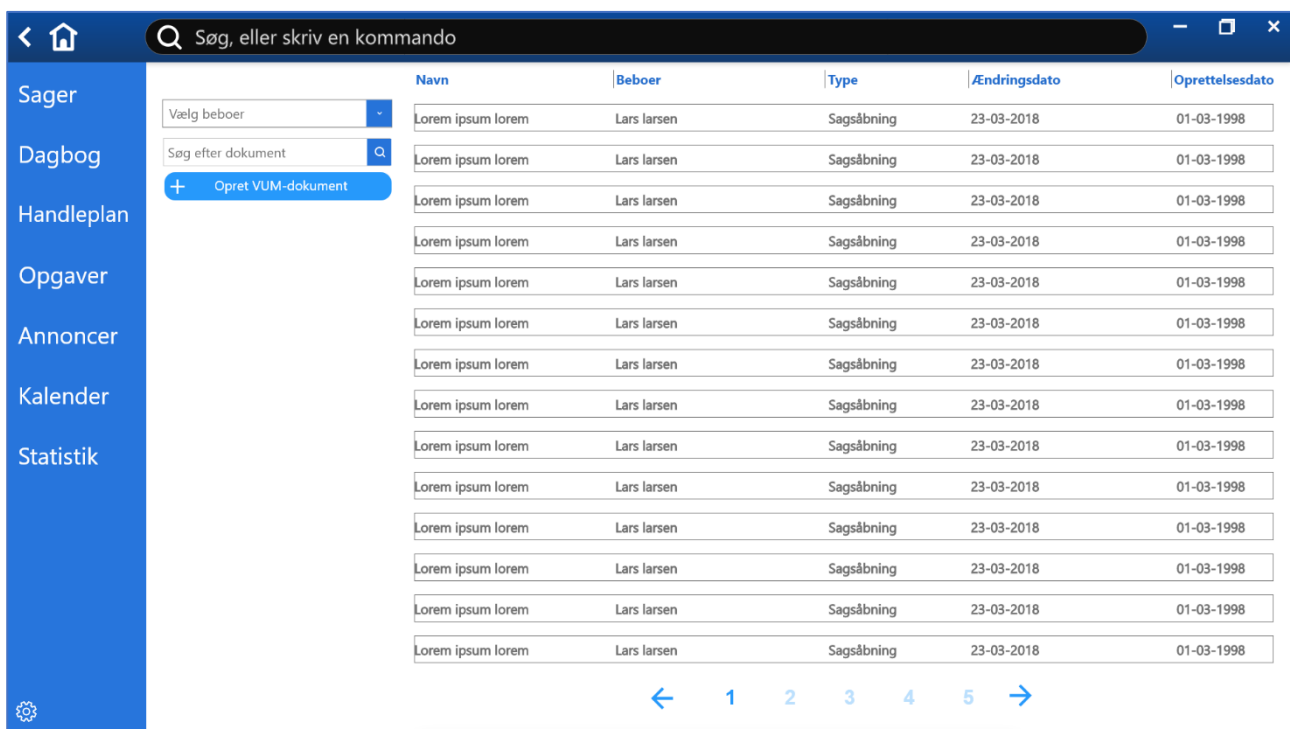
Risikonavn	Strategi	Type
Uretmæssigt adgang til system	To punkts validering for at logge ind og for adgang til dokumentet. Derudover auto log-ud ved inaktivitet	Undgåelse
Dårligt resultat af brugervenlighedstest	Lave en meget konkret og specifik brugervenlighedstest; Lave brugervenlighedstest på forskellige dele af platformen; Have minimum 15 brugere der tester programmet; Teste tidligt i elaborationsfasen	Undgåelse
Dårligt resultat af brugervenlighedstest	Lave ny brugervenlighedstest hvis tiden tillader det; Argumentere for hvorfor testen ikke virkede i rapporten	Beredskabsplan
Tidsplanen skrider	Omprioriterer krav; Dynamisk tidsplan; Ekstra overarbejde;	Beredskabsplan
For lang tid at udarbejde arkitekturen for platformen	Følge tidsplan; Prioriterer krav; Ekstra fokus på Scrum Master;	Undgåelse
Manglende muligheder i Java	Undersøger eksterne libraries; Ændre designet efter mulighederne der er i Java;	Beredskabsplan
Uventet information fra virksomhedsmøde	Prioriterer information fra virksomhedsmøde; Udforsker information fra virksomhedsmødet og diskuterer for retningsvalg i rapport	Beredskabsplan
Gruppen mister et medlem	Genovervej Scrum-roller; Prioriterer arbejdsbyrde og implementering af krav;	Beredskabsplan
Gruppen mister et medlem	Bruger tid på at diskutere konflikter i gruppen	Undgåelse

**Tekniske gennemførlighed**

Det vil ikke være alle dele af SSB-systemet der vil være mulige at gennemføre i dette projekt. Flere krav, som ville være udmærkede krav til et rigtigt projekt, kan ikke opfyldes i projektet her. For det første har vi planer om at udarbejde den overordnede platform hvorfra man kan tilgå alle systemets moduler. Det vil ikke være muligt at implementere funktionalitet for alle disse moduler på grund af projektets tidsrammer. Derfor fokuseres der udelukkende på sagsudredningsmodulet. I systemet kan vi eventuelt lave skitser til de andre moduler, så man kan forestille sig hvordan det endelige produkt ville ende ud med at kunne. Hvis det viser sig, at projektets ambitioner er opnåede efter kort tid, kan vi gå i gang med at udvikle funktionalitet til de yderligere moduler.

Projektet er også begrænset på andre måder end bare tidsmangel. F.eks. ønskes det, at systemet skal være i stand til at hente information om beboere igennem CPR-registeret. Dette vil vi højest sandsynligt ikke få lov til i dette studieprojekt. Det samme gælder for kravet der handler om at sende beboerinformation til Sundhed.dk.

Andre tekniske udfordringer kan også opstå. Igennem mockup-programmet Adobe XD har gruppen udarbejdet nogle skitser for, hvordan det ønskes, at systemet skal ende med at se ud (se billedet herunder). Det er ikke sikkert, at vi med de tilgængelige værktøjer (NetBeans, SceneBuilder) kan realisere skitserne fuldstændigt.



The screenshot shows a web application interface. On the left is a blue sidebar with navigation links: Sager, Dagbog, Handleplan, Opgaver, Annoncer, Kalender, and Statistik. At the bottom of the sidebar is a gear icon. The main content area has a search bar at the top with the text 'Søg, eller skriv en kommando'. Below the search bar are two input fields: 'Vælg beboer' with a dropdown arrow and 'Søg efter dokument' with a search icon. Below these is a blue button with a plus icon and the text 'Opret VUM-dokument'. The main area contains a table with the following columns: Navn, Beboer, Type, Ændringsdato, and Oprettelsesdato. The table has 15 rows of data, all with 'Lorem ipsum lorem' in the 'Navn' column and 'Lars larsen' in the 'Beboer' column. The 'Type' column contains 'Sagsåbning' for all rows. The 'Ændringsdato' column contains '23-03-2018' for all rows. The 'Oprettelsesdato' column contains '01-03-1998' for all rows. At the bottom of the table is a pagination bar with a left arrow, the numbers 1, 2, 3, 4, 5, and a right arrow.

Navn	Beboer	Type	Ændringsdato	Oprettelsesdato
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998
Lorem ipsum lorem	Lars larsen	Sagsåbning	23-03-2018	01-03-1998



## Prioritering

Der bliver kun skrevet på de udvalgte brugsmønstre fra kravspecifikation og altså ikke alle korte brugsmønstre fra bilaget. Derudover er der også udvalgt nogle supplerende krav som har indflydelse på projektet.

### Brugsmønster 1 – Log ind

Det skal være muligt at logge ind sådan så bosteders medarbejdere kan adskilles fra hinanden og at medarbejderes rettigheder kan skelnes for at undgå brud på datasikkerhed.

Log ind vil have en stor indflydelse på arkitekturen da rettighederne for brugeren vil påvirke mængden af funktioner og adgangsrettigheder til information i systemet, for eksempel kan en vikar ikke navigere rundt i systemet, det eneste vikaren kan er at se notifikationer på startskærmen. Derfor skal informationen fra log ind skærmen være tilgængelig for resten af systemet.

Der er en risiko i at implementere brugsmønsteret, da adgangen til personlig information afhænger af dette log in. I tilfælde af at der er en fejl i implementeringen af dette krav, har det store konsekvenser. På den anden side, er dette brugsmønster også løsningen på en risiko ved at begrænse adgangen til personlig information.

Implementeringen af brugsmønsteret vil være første gang vi skal fremstille et log in system, og det involverer database og sikkerhedstjek hvilket vil give stor læringsmæssigt udbytte.

### Brugsmønster 2 - Lav notifikation

I et system som SSB er det vigtigt at kunne kommunikere med sine medansatte. Dette øger koordinationen og dermed sænker transaktionsomkostningerne for virksomheden. Derfor kan opfyldelsen af dette krav ende med at spare bostederne penge.

Arkitekturen vil umiddelbart ikke blive påvirket af dette krav.

Der medfølger en lille risiko ved dette opfyldelsen af dette krav. De ansatte kan igennem disse notifikationer skrive information om beboere til andre ansatte. Information som eventuelt er fortroligt. Her ligger ansvaret hos de ansatte til ikke at bryde GDPR loven ved at dele information om beboere som er fortrolige.

Det læringsmæssige udbytte for dette krav er rimelig stort da der skal laves en form for besked system, som vil kræve kommunikation mellem forskellige computere, dog på samme netværk.

### Brugsmønster 3 - Opret VUM-dokument

Dette brugsmønster er en central funktion i sagsudredningen, som bostederne skal benytte sig af. Derfor har kravet stor betydning for det forretningsmæssige benefit, da bostederne omhandler sagsudredning og det er bostedernes funktioner er.

Opret VUM-Dokument er en vigtig del af arkitekturen da arkitekturen er centeret omkring sagsudredning, og det indebærer oprettelse og udfyldelse af VUM-dokumenter. Derudover skal der også være fokus på data-opbevaring da de skal opbevares i en database.

Det er vigtigt at implementere brugsmønsteret korrekt, da det indebærer opslag af persondata op, og derfor kan man risikere at indhente information som ikke var hensigten at indhente.

Dette brugsmønster vil give os indsigt i, opbevaring af data ifølge GDPR-love. Samtidig giver det os indsigt i at lave skabeloner som skal kunne udfyldes senere.

### Brugsmønster 4 - Redigere dokument

Redigere dokument har en stor betydning for brugervenligheden for medarbejderne, da det vil være frustrerende ikke at kunne redigere fejl og mangler.

Brugsmønstret har ikke en stor betydning for arkitekturen af programmet.

Der er en risiko i at brugeren kan komme til at slette vigtig information, som ikke var hensigten, som både kan påvirke brugervenligheden, men også datasikkerhed. Denne risiko kunne eventuelt håndteres ved at lave en fortryd funktion (CTRL + Z), eller at gemme versioner af dokumenter i et antal dage, så brugeren kan gendanne dokumenter.

Hvis risikoen bliver håndteret ved brug af en fortryd funktion for eksempel, kunne det godt give et læringsmæssigt udbytte, dog ikke i så høj grad i forhold til andre krav.

### **Brugsmønster 5 - Afslut sag**

Det er vigtigt at afslutte sagen, da det er her, at dokumenterne fra en beboer skal fjernes fra medarbejdernes liste af dokumenter de kan tilgå og rykkes til et arkiv. Det er vigtigt for virksomheden, da det potentielt kunne skade dem, at der lå sensitiv og privat information om deres tidligere beboere frit tilgængeligt.

Brugsmønstret er en vigtig funktion, men har ikke stor indflydelse på arkitekturen.

Brugsmønstret giver ikke et stort læringsmæssigt udbytte, da det er simpelt at implementere.

### **Brugsmønster 6 - Åbn dokument**

Det er meget vigtigt for bostederne at dette krav er opfyldt. Hvis de ikke er i stand til at åbne dokumenterne, kan der ikke foretages sagsudredning, hvilket er et stort fokuspunkt i dette system.

Ved implementeringen af dette krav, bliver arkitekturen nødt til at laves således, at systemet er i stand til at hente data fra en database.

Der følger en risiko med dette krav. Hvis personalet er i stand til at åbne dokumenter med fortrolig information, kan det slippe ud til uautoriserede personer. Man kan forestille sig, at personale åbner et dokument og derefter forlader sin computer. Dokumentet er nu frit tilgængeligt for alle forbipasserende mennesker. Dette bliver man nødt til at forhindre ved f.eks. at få systemet til selv at lukke dokumentet og logge ud efter et lille tidsrum med inaktivitet. Dog vil selv denne forbeholdsregel ikke løse problemet fuldstændigt.

Der er et læringsudbytte fra dette krav, da dokumentet skal hentes fra en database. Derfor bliver vi nødt til at få systemet til at kommunikere med en database og hente data derfra.

### **Brugsmønster 7 – Log ud**

Brugsmønstret har stor betydning for medarbejdere og for datasikkerheden. Hvis det står logget ind på brugere som betyder at uvedkommende kan benytte systemet.

Det har lille betydning for arkitekturen.

Dette krav fjerner ikke nogen form for risiko. Til gengæld er det med til at fjerne en risiko, da muligheden for at logge ud mindsker chancen for at en uautoriseret bruger får adgang til en andens konto.

Dette krav bidrager til at lære os hvordan man fremstiller et login-system.

### **F00 SSB skal logføre alle aktiviteter:**

Med det høje krav til Datasikkerhed der sættes i dag, er det vigtigt at der i et offentligt system som SSB bliver logført alle aktiviteter der bliver foretaget af enten systemet eller en medarbejder. Dette sikrer høj

datasikkerhed både for vedkommende hvis personlige oplysninger der ligger i systemet, men også for medarbejdere og ejere af systemet.

Logføring af systemet vil hjælpe med altid at kunne dokumentere hvis der skulle opstå fejl i systemet, men samtidig også kunne fortælle hvis en persons oplysninger er blevet tilgået eller redigeret af uvedkommende.

Fra et arkitektonisk perspektiv vil logføring betyde at systemets log skal været forbundet med alle systemets elementer og derved vil der opstå høj kobling mellem logføringsklassen og de resterende klasser. Da logføringen skal være et sikkerhedselement i systemet, må et eventuelt nedbrud, ikke resultere i at elementer af systemet ikke bliver logført og at man derved kan bevidst eller ubevidst få udnytte en fejl i systemet til at undgå sin aktivitet bliver logført.

En risiko ved at logføre aktiviteter vil være at loggen vil indeholde en masse information samlet på ét sted, og derved opstår der en risiko hvis uvedkommende får adgang til loggen. Udover førnævnte kan der også opstå problemer med brugerne af systemet der føler sig overvåget i det alt hvad de laver er logført, samt der kan opstå nye love omkring logføring (Overvågning af medarbejdere), som kan skabe komplikationer med det logføringssystem som SSB får.

Som en del af vores research i logføring vil vi opnå dybere viden inden for love og regler for datasikkerhed, samt opbevaring af data omkring medarbejdere. Udarbejdelsen af logføringssystemet vil kræve at er kreative i vores programmering og vil derfor bidrage til vores evner som programmører.

#### **F02 SSB skal kunne udveksle data med CPR-registeret:**

Dette krav vurderes til at være meget vigtigt i et system som SSB. Projektet fokuserer på at gøre systemet så sikkert som muligt imens brugervenligheden opretholdes. Hvis systemet er i stand til automatisk at finde information om beboeren igennem CPR-registeret, kan brugervenligheden øges gevaldigt ved at lade programmet udfylde specifikke felter selv. Dog i et studieprojekt som dette, vil det ikke være nemt at implementere denne funktionalitet. Vi sigter dog stadig efter at kunne lave en form for simulering af CPR-registeropslag.

Kravet vurderes til ikke at have en betydelig betydning på systemets overordnede arkitektur.

Hvis systemet skal opfylde kravet, vil risikoen øges en smule. Informationen man kan hente fra CPR-registeret, skal ikke være frit tilgængeligt for alle. Det er derfor vigtigt at sørge for, at kun autoriserede personer kan foretage denne form for opslag.

Ved opfyldelsen af dette krav, lærer det, hvordan man får sit program til at kommunikere med udefrakommende systemer. Dette kommer sandsynligvis ikke til at være tilfældet i dette projekt, da vi bliver nødt til at simulere CPR-registeropslagene. Dog giver det en idé om hvordan emner som disse håndteres.

Derudover vil dette krav også tvinge os til at undersøge hvilket data der må og ikke må vises og datasikkerhed generelt.

#### **F03 SSB skal kunne sende information til Sundhed.dk:**

F03 gør det muligt for systemet at overføre specifik information til et eksternt system, som kan opbevare og gøre det muligt for brugeren af sundhed.dk, at se personlig data omkring dem selv. Dette er en vigtig funktion for systemet, da brugeren har mulighed for at tilgå deres sundhedsjournaler på sundhed.dk, hvis de giver samtykke til dette. Funktionen giver i sig selv ikke meget nytte til SSB-systemet og er ikke relevant for at systemet kan fungere optimalt.

Kravet har ikke nogen påvirkning på systemets arkitektur, da det udelukkende er en overførsel af data.

Kravet udgør ikke en stor risiko for SSB-systemet, da dataen allerede er opbevaret i systemet. Risikoen vil derimod være pålagt Sundhed.dk, som modtager denne kritiske information. F03 har en stor lighed med krav

F02 og vil derfor have mange af de samme læringsmæssige udbytter, da der læres om kommunikation mellem systemer. Der vil dog i dette krav være fokus på en ensidig kommunikation, hvilket kun vil være afsendelsen af data. At udarbejde og implementere denne funktion, vil ikke være en del af projektet og vil derimod enten være en antagelse for, at det vil virke eller bliver udarbejdet som en simulation.

#### **F04 SSB skal fordele brugere på specifikke roller og ID for at afgrænse adgang til data:**

F04 spiller en stor rolle for systemets grundlæggende datasikkerhed. Uden at implementere denne funktion, ville systemet været frit tilgængelig for alle brugere og derfor være i strid med lovgivning om dataafgrænsning. Rolle-inddeling er også funktionen, som separerer de forskellige medarbejdere ud på deres givne arbejdspositioner, såsom socialrådgiver og sagsbehandler. Ved uddelingen af disse positioner, er systemet også i stand til at sammenkoble specifikke beboere til de ansatte.

Kravet spiller en stor rolle for systemets arkitektur, da det konstant skal validere din rolle ved forskellige aktioner i systemet. Dette medfører en høj kobling for systemet, da der er mange klasser, som skal snakke sammen for at kunne validere brugerens rolle.

Der ligger en stor risiko bag dette krav, da det er en af hovedpunkterne for dataafgrænsningen i systemet. Hvis en forkert rolle uddeles af systemets administrator, vil denne bruger få adgang til information, som skulle være utilgængeligt for den givende bruger.

Kravet har en stor indflydelse for arkitekturen af systemet, og det udgør derfor et stort læringsmæssigt udbytte. Der vil f.eks. her blive gået i dybden med det arkitektoniske i forhold til dataafgrænsning.

#### **U00 SSB skal have en tilfredsstillende vurdering fra testbrugere:**

Kravet skal opfylde en, hvis form for brugervenlighed og dette måles ved en brugervenlighedstest. Testen måler på om systemet er overskueligt, da dette vil bidrage til at systemet vil kunne fungere effektivt efter udgivelsen.

Da dette indbefatter resultatet af en test, har det ingen indflydelse på arkitekturen af systemet, da alt redigering efter svaret af denne test, vil blive udført på GUI delen af systemet. Kravet har heller ikke nogen betydning når det angår risiko.

Kravet vil medføre en forståelse for hvordan et system skal designes på præsentationslaget for at give et godt overblik og være effektivt at arbejde med. Kravet vil også give en erfaring med at udvikle teste til brugere, som kan give brugbar feedback, der kan arbejdes med.

#### **U01 SSB skal give brugeren mulighed for at gå tilbage til systemets forrige grafiske tilstand:**

Dette krav er brugervenligheds fokuseret og der vil derfor være overvejelser om at hvis brugeren er inde på VUM-dokumenters overblik for eksempel og har valgt en specifik beboer at se dokumenter over, og brugeren herefter klikker sig ind på en af dokumenterne og herefter går tilbage, så ville det være en fordel hvis brugeren ikke skal vælge den specifikke beboer igen, men at systemet har gemt hvilken beboer der var valgt.

U01 har en stor betydning for arkitekturen af programmet, da der konstant skal gemmes systemets forrige tilstand (grafiske layout) så brugeren kan gå tilbage til den tilstand. Dette kan give høj kobling i arkitekturen da der er mange forskellige muligheder i at implementere denne funktion. Der skal derfor fokuseres på at fremstille en løsning som giver lav kobling i systemet.

Der vil være en risiko for at brugere af systemet har arbejdet på et dokument og uden at gemme forsøger at gå tilbage til systemets tidligere tilstand, og data derved går tabt. Der vil ikke være risiko for brud på datasikkerheden eller andre kritiske risici.

#### **U03 SSB skal synliggøre hvilke brugere, som har oprettet, redigeret og slettet dokumenter:**

U03 gør det muligt for systemet at holde et overblik over, hvem som tilgår de forskellige dokumenter og opretter dem. Man kan ved brug af dette lokalisere, hvem der har tilgået dokumentet sidst, hvis der er blevet lavet en fejl eller man har brug for yderligere forklaring på en beskrivelse eller andet.

Kravet udgør ikke en stor betydning for arkitekturen af systemet eller for risiko.

#### **U06 SSB skal vise en dialogboks i tilfælde af at brugeren går væk fra arbejde der ikke er gemt:**

Kravet er en foranstaltning imod at miste arbejde ved en uhensigtsmæssig aktion, som vil medføre, at man forlader ens dokument.

Dette krav har ikke en stor betydning for arkitektur.

Kravet har derimod en risiko for systemet, da det minimerer chancen for at arbejde, bliver slettet og skal produceres forfra.

#### **S00 SSB skal inddeles i en Plug and Play samlekasse med moduler:**

Inddeling af systemet i moduler hjælper med at modificere systemet. Det er derved muligt for et bosted at få en modificeret løsning. Det giver også mulighed for at opdatere enkelte moduler separat, uden større indflydelse på resten af systemet.

Fra et arkitektonisk perspektiv kræver modulopbygningen at hvert enkelt modul er selvstændigt. Dette begrænser mulighederne for et modul så længe modulet er frakoblet systemet.

Ved at gøre systemet Plug n Play, kan man hjælpe med at hindre systemfejl, da et modul ikke længere vil have samme indflydelse på andre moduler. Dog kan man ved Plug n Play, risikere at miste noget funktionalitet, da modulerne ikke kan sammenkobles på samme niveau som ét sammenbygget system.

Som læringsudbytte får gruppen mulighed for at designe de forskellige moduler, men med tanke på senere sammen bygning, dette er en evne som er relevant til senere brug i erhvervslivet.

#### **Prioritering med MoSCoW**

Der er gjort brug af MoSCoW-metoden til at prioritere de forskellige krav der er sat til programmet. Det gør, at man får et overblik over hvilke krav der er et *Must*, *Should*, *Could* eller *Won't have*-krav. *Must have*-krav er alle de essentielle krav som skal til for at programmet giver mening. *Should have*-krav er alle de krav som programmet burde have for at det er mest optimalt. *Could have*-krav er alle de krav man kunne implementere hvis der opstod mere tid eller flere penge til at kunne implementere i programmet. *Won't have*-have krav som man vil fokusere på i fremtiden som ofte er urealistiske krav i den pågældende situation.

B står for brugsmønstre

MoSCoW Chart			
Must have	Should have	Could have	Won't have
F00	U00	U02	F03
F02	U04	F01	
F04	U05	P00	
U01	R00	P02	
U06	P01	P03	
U07	U03	S00	
B01	U08	R01	
B03	B02		
B04			
B05			

B06

B07

**Metoder i elaborationsfasen****Undersøgelsesmetoden**

Undersøgelsesmetode er en løsningsmetode til et teoretisk problem. Dette foregår ved brug af teoretiske litteraturstudier, informationssøgninger og egen dataindsamling. I dette forløb vil litteraturstudier fra internettet blive brugt til at understøtte et teoretisk problem, samt information leveret af den vedrørende organisation EG Team Online ApS. Denne metode bliver brugt i den indledende del af forløbet og i inceptionsfasen (Se afsnit RUP/UP for uddybelse af inceptionsfasen). Senere i projektforsløbet bliver spørgeskemaer og evt. målinger undersøgt ved en praktisk undersøgelse af produktet til at indsamle yderligere data.

**Konstruktionsmetoden**

Konstruktionsmetoden er en løsningsform til praktiske problemer, hvilket i dette projekt er samlingen af forskellige systemer under en "hat". Dette skal løses ved at konstruere et produkt eller system, som overholder kravene leveret af EG Team Online. Konstruktionsmetoden er ofte opdelt i faser, som minder meget om projektets faser, og vil derfor omfatte flere metoder, som f.eks. undersøgelsesmetoden til vidensindsamling.

**RUP/UP**

Rational Unified Process er en softwareudviklingsproces der forsyner developers med en model for effektiv udvikling og implementering af softwareudviklingscyklussen. RUP er en kommerciel ekstension af UP (Unified Process) og indeholder flere standarder, værktøjer og andre "features". Modellen er inddelt i 4 faser; *Inception, Elaboration, Construction, Transition*.

**Inceptionsfasen** er byggeklodsen for ideudvikling og strukturen af projektet. Det er i denne fase at vurdere om projektet er værd at arbejde videre med og om det har potentiale for en succesfuld løsning. Dette vurderes ved brug af værktøjer som brugsmønsterdiagrammer, aktørlistes, kravspecifikationer osv.

**Elaborationsfasen** er til for at analysere krav, udvikle nødvendig arkitektur og brugsmønstermodeller for systemet. Denne fase er yderst vigtigt, da det processen går fra lav risiko til høj risiko efter dette forløb i konstruktionsfasen. I elaborationsfasen udvikles der også prototyper af produktet i iterationer, før det viderearbejdes og færdiggøres i konstruktionsfasen.

**Konstruktionsfasen** er udviklingsfasen. Det er i dette forløb, at alt kodning og implementering foregår. Når alle milestene og krav fra forrige faser er opfyldt, er konstruktionsfasen overstået.

**Transitionsfasen** er udgivelsesfasen. Det er forløbet, hvor det færdige produkt er udgivet og leveret til kunderne. Dog er det også fasen som supplerer produktet efter udgivelsen med bug fixes, patches osv.

I dette forløb vil vi gennemgå og bruge de to første faser; *Inception* og *Elaboration*. Inceptionsfasen i dette forløb er bestået af uddybelsen af den forarbejdet problemstilling, hvor der herefter er blevet arbejdet med brugsmønster, aktørlistes, brugsmønsterdiagrammer, domænemodel og kravspecifikation for at bedømme omfanget af arbejdet i projektet før elaborationsfasen. Elaborationsfasen er den sidste fase i dette projektforsløb og det vil derfor også være i denne fase at holdet udvikler det "færdige" produkt, som et løsningsforslag på den stille problemformulering. Der vil derfor i denne fase blive inddraget Scrum og agil UP for at hurtigt og fleksibelt udvikle et produkt.

**Scrum**

Scrum hører under den agile udviklingsmetode. Den agile arbejdsmetode tager udgangspunkt i, at software er en flydende proces med et kompliceret forløb og uforudsigelige kravændringer. Det er derfor vigtigt for denne udviklingsmodel at tage højde for, at der skal være mulighed for ændringer i forløbet og produktet på en effektiv måde. Scrum er en optimal udviklingsmetode for et forløb, hvor man ikke kender til alle krav i begyndelsen af processen og hvor man muligvis skal bruge nye værktøjer eller teknologier. Den agile arbejdsmetode er oftest brugt i små udviklingshold, da dette gør det nemmere at organisere fleksible tidsplaner og deadlines, i forhold til at skulle informere store teams med separate lokationer omkring ændringer i forløbet.

En vigtig arbejdsmetode i Scrum udviklingsforløbet hedder et *sprint*. Arbejdet i et Scrum-forløb inddeles ofte i *sprints* som varer i maksimalt op til 30 dage. Et *sprint* indledes med et møde, som også hedder en *Sprint Planning* og herefter går holdet i gang med at producere en ny iteration af det kørende system, hvor de krav og ændringer, som blev vedlagt ved startmødet, er blevet implementeret. Scrum-metoden indeholder oftest tre roller, som er essentielle for et projekt. Disse roller inkluderer: *Product Owner*, *Scrum Master* og *Development Team*.

Scrum og den agile arbejdsmetode er modsætninger til de kendte planstyret arbejdsmodeller som vandfald- og spiralmetoderne. Scrum er derfor heller ikke en optimal projektmodel og er ikke brugbar i forløb, hvor man skal have højt fokus på ressourcestyring og omkostninger. Modellen er derimod bygget til at have et struktureret forløb over leverancer i hurtige udviklingsprocesser.

Da gruppen i elaborationsfasen skal udvikle to iterationer af løsningsforslaget over en kort periode på 7 uger, vil *Scrum* være en optimal arbejdsmetode at arbejde med, da der udover dette ikke skal tænkes over ressourcestyring og omkostninger i projektforslaget.

**Ressourcer**

Skemaerne herunder viser gruppens arbejdsindsats (pr. person) uge for uge i de to faser.

Inceptionsfasen					
Uge:	9	10	11	12	I alt
Sammen	8	8	12	21	49
Hjemmearbejde	0	0	2	2	4

Elaborationsfasen									
Uge:	13	14	15	16	17	18	19	20	I alt
Sammen	8	14	14	0	16	14	14	20	100
Hjemmearbejde	4	4	4	8	6	4	4	17	51



## Konklusion

Ud fra business casen udarbejdet i Organisation og ledelse, kunne det ses at produktet vi skal fremstille, befinder sig i et stabilt miljø, men med konkurrenter såsom KMD Nexus. Det er derfor vigtigt at udarbejde systemet med fokus på negative sider af konkurrenters produkter. Det har vi forsøgt at gøre ved at lave en overordnet kravspecifikation.

Domæneelementerne som den overordnede kravspecifikation fremhæver og som skal implementeres i systemet senere, medbringer dog en række risici i forhold til dataafgræsning og datasikkerhed som er vigtigt at overveje i det færdige system, da systemet skal opbevare meget følsomme informationer. For at overkomme risiciene fokuserer vores løsning på et rolle- og identifikationssystem med flere loginpunkter som derved bl.a. giver adgang til de rette VUM-dokumenter.

Udover domænemodellen som viser elementer der skal indgå i løsningen, er der også blevet udarbejdet mock-ups, hvor designet er blevet fastlagt og en række funktioner undersøgt, såsom opslag af persondata og videre-sending af sager til Sundhed.dk.

For at få udarbejdet den rent faktiske løsning, skal der i elaborationsfasen laves grund-arkitekturen og gå i dybden med en række krav og brugsmønstre. Der er derfor blevet lavet en MOSCOW prioritering af krav og brugsmønstre, hvor must-have kravene er det første, der skal arbejdes med i elaborationsfasen, dette inkluderer krav, såsom at medarbejdere skal have roller og at man skal oprette VUM-dokumenter i systemet.

For at nå frem til disse resultater er det blevet fastlagt at gruppen skal bruge Scrum, undersøgelsesmetoden og konstruktionsmetoden.

Milepælsplan, samarbejdsaftale er blevet revurderet og vejlederaftale har der ikke været behov for at opdatere.

### 18.3 Testresultater

Hentning af 1000 notifikationer:

Difference:

1,5587E+12	1,5587E+12 934
1,5587E+12	1,5587E+12 847
1,5587E+12	1,5587E+12 891
1,5587E+12	1,5587E+12 917
1,5587E+12	1,5587E+12 923

Gennemsnit:	902,4
-------------	-------

Hentning af 5000 notifikationer:

1,5587E+12	1,5587E+12 6292
1,5587E+12	1,5587E+12 6545
1,5587E+12	1,5587E+12 6386
1,5587E+12	1,5587E+12 6247
1,5587E+12	1,5587E+12 6426

Gennemsnit:	6379,2
-------------	--------

Hentning af 1000 specifikke notifikationer blandt 6000 notifikationer

1,5587E+12	1,5587E+12 957
1,5587E+12	1,5587E+12 819
1,5587E+12	1,5587E+12 866
1,5587E+12	1,5587E+12 801
1,5587E+12	1,5587E+12 821

Gennemsnit:	852,8
-------------	-------

Hentning af 500 dokumenter

1,5587E+12	1,5587E+12 2010
1,5587E+12	1,5587E+12 1961
1,5587E+12	1,5587E+12 2049

1,5587E+12	1,5587E+12	1983
1,5587E+12	1,5587E+12	1921

Gennemsnit:	1984,8
-------------	--------

## Hentning af 2000 dokumenter

1,5587E+12	1,5587E+12	2819
1,5587E+12	1,5587E+12	2480
1,5587E+12	1,5587E+12	2438
1,5587E+12	1,5587E+12	2442
1,5587E+12	1,5587E+12	2467

Gennemsnit:	2529,2
-------------	--------

## Hentning af 500 dokumenter blandt 2000 dokumenter

1,5587E+12	1,5587E+12	2799
1,5587E+12	1,5587E+12	2968
1,5587E+12	1,5587E+12	2859
1,5587E+12	1,5587E+12	2728
1,5587E+12	1,5587E+12	2744

Gennemsnit:	2819,6
-------------	--------

## Oprettelse af nyt dokument

1,5587E+12	1,5587E+12	696
1,5587E+12	1,5587E+12	669
1,5587E+12	1,5587E+12	669
1,5587E+12	1,5587E+12	679
1,5587E+12	1,5587E+12	671

Gennemsnit:	676,8
-------------	-------

## Opdatering af et dokument blandt 10.000 dokumenter

1,5587E+12	1,5587E+12	364
1,5587E+12	1,5587E+12	355
1,5587E+12	1,5587E+12	333
1,5587E+12	1,5587E+12	347
1,5587E+12	1,5587E+12	330
Gennemsnit:		345,8

## 18.4 Projektforslag

SSB Kalender - Projektforslag

**Afleveringsdato:** 22-02-2019

**Gruppemedlemmer: E-mail:**

Morten Krogh Jensen [mortj18@student.sdu.dk](mailto:mortj18@student.sdu.dk)

Nicolai Ankjær Kruuse [nikru18@student.sdu.dk](mailto:nikru18@student.sdu.dk)

Thomas Steinfeldt Laursen [tlaur18@student.sdu.dk](mailto:tlaur18@student.sdu.dk)

Mathias Haurum Løycher [maloe18@student.sdu.dk](mailto:maloe18@student.sdu.dk)

Michael Haugaard Pedersen [micpe18@student.sdu.dk](mailto:micpe18@student.sdu.dk)

Oliver Marco van Komen [olvan18@student.sdu.dk](mailto:olvan18@student.sdu.dk)

**Vejleder: E-mail:**

Jeppe Schmidt [jep@mmmi.sdu.dk](mailto:jep@mmmi.sdu.dk)

**Gruppenummer:** 25

## Problemstilling

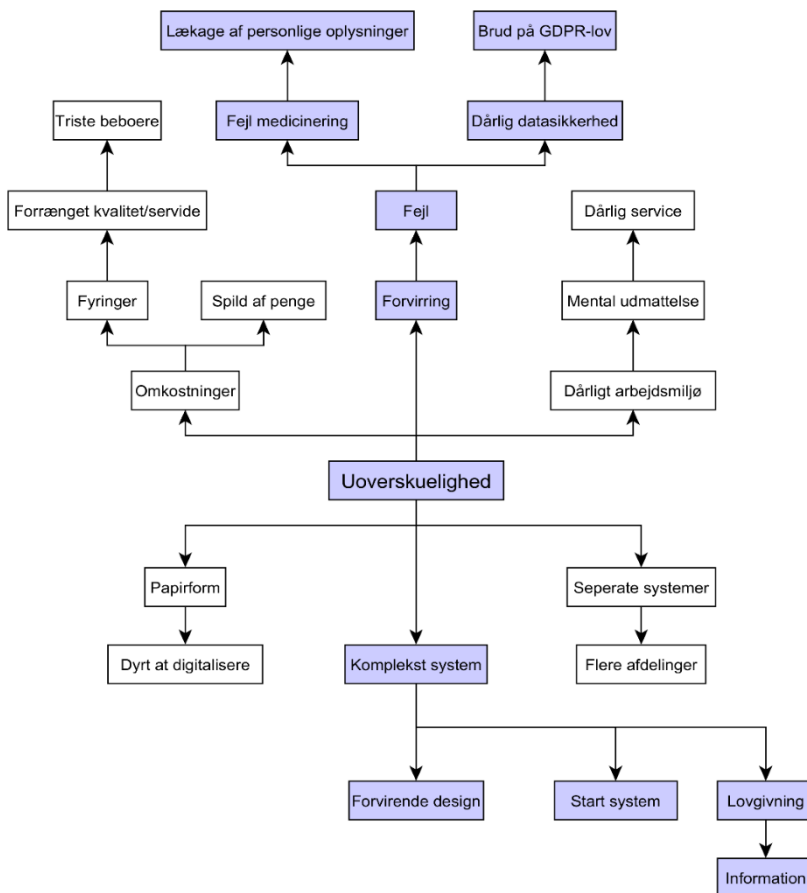
I et arbejdsmiljø hvor information og persondata bliver samlet og delt på separate systemer, kan der opstå et uoverskueligt og fejlskabende workflow. Ved brug af et problemtræ - fig. 1 - kan der ses forskellige årsager og konsekvenser til dette. Her er et specifikt område blevet isoleret og sat som fokus for projektideen. De udvalgte årsager og konsekvenser lægger op til løsningsmuligheder, som omhandler reducere af kompleksiteten ved god datasikkerhed, den medførende forvirring og menneskefejl ved et kompliceret system.

**Undren**

Efter skærpelse af datasikkerhed og de ny-indførte GDPR-love, risikeres der brud eller lækage af person-data, hvis systemet er uoverskueligt eller alt for komplekst. Vil det derfor være muligt at samle Sensum Bosted og Sensum til ét sikkert system og samtidig bevare brugervenligheden?

### Projekttideen

En samling af hverdagens aktiviteter for borgere på bosteder (Sensum Bosted) og sagsudredning for medarbejdere (Sensum) i form af en kalender. Kalenderen skal kunne åbne borgers journaler, oprette nyt dagbogsindlæg via aktiviteter i kalenderen, udskrivning af kalenderens aktiviteter med piktogrammer m.m. Disse funktioner skal være mulige uden at gå på kompromis med brugervenligheden for medarbejdere.



Figur 1 Problemtræ med afgrænsning markeret med blå

### Motivation

Personlige data bliver i større og større grad lagret online, og der har været mange sikkerhedsbrud ved lækage af informationer. Ved indførslen af nye GDPR-regler, skulle data beskyttes bedre af virksomheder. Dog er der stadig mange systemer, som ikke lever op til det. Personligt data kræver høj datasikkerhed, da det indbefatter alt fra sagsbehandlinger omkring misbrug, etnisk baggrund, navn, adresse og meget mere.

I vores dagligdag bliver vi ofte udsat for at skulle dele information omkring vores eget liv. Dette sker, når vi tager på hospitalet, til lægen, laver kontoer på diverse hjemmesider, skal bestille mad osv. Vi er derfor alle

sammen i risiko, når det kommer til invadering af vores private informationer på internettet. Datasikkerhed er et stort og kompliceret emne, som softwareingeniør-studerende skal kunne takle og kende nært. Derfor vil en motivation for gruppen være at få et grundlæggende indblik i, hvordan dette håndteres, da der er stor sandsynlighed for, at vi kommer til at skulle arbejde med lignende projekter i fremtiden.

UP kommer også til at spille en stor rolle i projektets forløb. Gruppen er motiveret til at følge denne model, da den inkorporerer mange dele af professionel softwareudvikling. Igennem denne model opnår gruppens medlemmer viden om brugsmønstermodellen, kravspecifikationen m.m.

Gruppen er også motiveret ved introduktionen af database, som skal tilføjes senere i projektforsløbet. Ved brug af database får gruppens medlemmer et indblik i, hvordan lagring af information foregår, for eksempel af opbevaring af oprettede aktiviteter såsom kalenderen og personlige data.

## Nuværende viden

Den nuværende viden gruppens medlemmer har stammer fra familie, bekendte, personlige erfaringer med forskellige systemer på arbejdsmarkedet og igennem nyhederne. Denne viden inkluderer frustration omkring systemernes brugervenlighed, såsom ineffektiv funktionsudnyttelse og forvirrende brugergrænseflade.

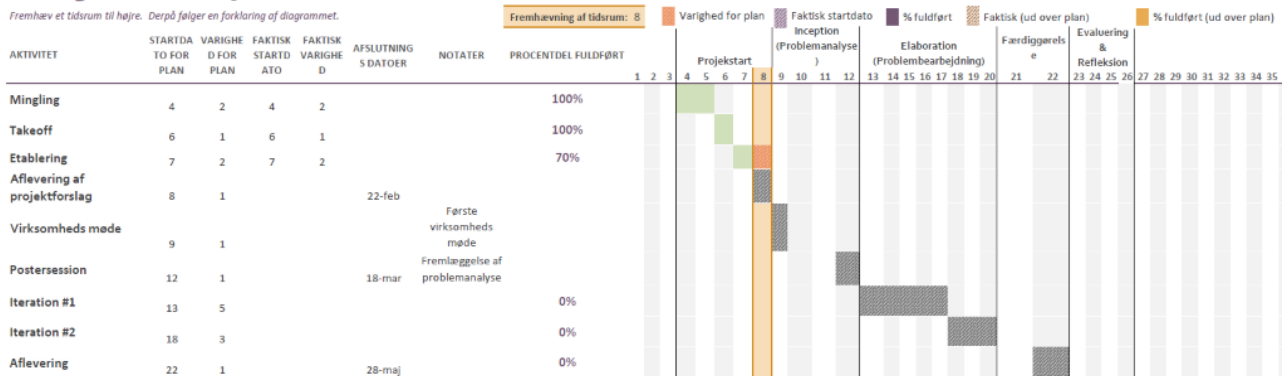
Gruppen har en mangel af viden, når det kommer til datasikkerhed og de tilhørende GDPR-love. Gruppen besidder også minimal viden omkring brugen af databaser i sammenhæng med projektforsløbet. Viden omkring projektets nuværende produkt er også begrænset indtil mødet med EG Team Online i uge 10. Der er her for eksempel tale om manglerne på det nuværende system og systemets alder.

## Tidsplan

Tidsplanen er udarbejdet ud fra projektrammeplanen med enkelte tilføjelser i form af milepæle. Planen er opsat sådan, at hver periode har en forventet arbejdsfordeling med en tilhørende afslutningsfase til hver. Tidsplanen vil ændre sig med tiden, efter at der bliver udarbejdet milepæle og fokuspunkter. Designet er nu opbygget sådan, at den forventede plan er markeret som grå, allerede udførte fremhævet med grøn, mens manglende udførelse af den forventede plan markeres med orange.

## ProjektTidsplan

Fremhæv et tidsrum til højre. Derpå følger en forklaring af diagrammet.



## Værktøjer

I det kommende projektarbejde vil gruppen anvende forskellige værktøjer som skal hjælpe med projektets organisering og samarbejde.

### GitHub

Versionsstyringsværktøjet GitHub vil blive brugt i mange sammenhænge. Herinde kommer gruppen til at holde logbog over projektet. Efter hvert gruppe- og vejledermøde bliver et referat skrevet i projektets GitHub-wiki. Wikien kommer til at indeholde en samling af referater til alle gruppemøder igennem projektforløbet, så alle gruppemedlemmer nemt kan tilgå tidligere logbøger igennem hele projektet.

Andre vigtige dokumenter såsom samarbejds- og vejlederaftale opbevares også inde i projektgruppens GitHub-side, så det er nemt at tilgå for både gruppemedlemmer og vejleder.

Udover at samle al dokumentation af gruppemøderne, vil projektgruppens GitHub også stå for versionsstyringen af projektets produkt. Med GitHub er man i stand til nemt at holde styr på hvem, hvad og hvornår funktionalitet tilføjes til det udviklede program. GitHubs branch-funktion gør det muligt for alle gruppe-medlemmer at arbejde på hver sin del af produktet uden at forstyrre de andre. Af samme årsag bliver det også nemt at bevæge sig tilbage til tidligere versioner af produktet hvis noget eventuelt skulle gå galt.

Link til projektgruppens GitHub-side: <http://www.GitHub.com/Omvk97/SensumBosted>

### Microsoft Teams

Microsoft Teams er et andet værktøj som gruppen vil anvende til organisering af projektet og forenkling af samarbejdet mellem gruppemedlemmerne. Denne digitale samarbejdsplatform anvender gruppen til at samle vigtige dokumenter (tekstfiler, billeder osv.), så de nemt kan tilgås af alle. Herfra er alle gruppens medlemmer yderligere i stand til at arbejde i samme Word-dokument på samme tid, hvilket sikrer maksimal effektivitet. Microsoft Teams tilbyder en række af ekstra funktioner og muligheder, som f.eks. at optage



møder, video-chatte, tekst-chat, kanban-board og meget mere som projektgruppen kan få gavn af igennem projektet.

### Samarbejds- og vejlederaftale

Linket herunder fører til gruppens samarbejdsaftale:

<https://GitHub.com/Omvk97/SensumBosted/wiki/Samarbejdsaftale>

Linket herunder fører til gruppens vejlederaftale:

<https://GitHub.com/Omvk97/SensumBosted/wiki/Vejlederaftale>