

# MMMI

## Mangfoldigt Manager Management Instrument

Vejleder:

Henrik Lykkegaard Larsen [hlla@mmmi.sdu.dk](mailto:hlla@mmmi.sdu.dk)

Gruppemedlemer for gruppe 02:

Aleksander Grzegorz Duszkiewicz	<a href="mailto:aldus17@student.sdu.dk">aldus17@student.sdu.dk</a>
Aleksander Henriksen	<a href="mailto:alhen18@student.sdu.dk">alhen18@student.sdu.dk</a>
Aslak Mangino Nicolajsen	<a href="mailto:asnic18@student.sdu.dk">asnic18@student.sdu.dk</a>
Mathias Jackobsen	<a href="mailto:mjacok18@student.sdu.dk">mjacok18@student.sdu.dk</a>
Per Terp Larsen	<a href="mailto:pertl18@student.sdu.dk">pertl18@student.sdu.dk</a>
Steffen Vitten Pedersen	<a href="mailto:stped18@student.sdu.dk">stped18@student.sdu.dk</a>

# I Resume

Der blev givet en case hvor der frit kunne prioriteres mellem tre moduler og hvor der skulle arbejdes med mindst et modul. Gruppen valgte at fokusere på udviklingen af modulet, sagsudredning. Her blev der fundet frem til flere potentielle problemstillinger, herunder hvordan et program kan håndtere flere roller uden at de interagerer med hinandens data, samt hvilke sikkerhedsforanstaltninger der skal ligge til grund for at beskytte data. Der blev brugt UP, hvor diagrammer blev lavet og arbejdet i iterationer. Der blev også brugt metoder som kanban og par produktion. Under forløbet blev der benyttet elementer af Scrum. Der blev arbejdet med sagsudredningen, da det er her en sag startes op. Det første der blev gjort, var at udarbejde krav på baggrund af casen, og derudover finde yderligere information omkring den. Dette blev fundet via VUM (Anon., 2019), som havde nogle gode retningslinjer at følge. Resultatet blev en beskrivelse af hvad gruppen ville arbejde hen imod. Sikkerhedsforanstaltninger blev udviklet for at sikre, at brugere ikke sletter noget uden at ville det eller henter information om noget de ikke må have adgang til. Målet var også at en bruger får tildelt en rolle, som så har nogle rettigheder, der bestemmer hvad brugeren har adgang til.

## II Forord

Gruppen fik til opgave at lave et modul til systemet Sensum fra EG Team Online. Der var tre forskellige moduler som virksomheden ønskede belyst. Denne opgave omhandler hvordan gruppen har prøvet at løse opgaven omkring det valgte modul (sagsudredningen). Rapporten er skrevet til folk der har en grundlæggende forståelse for programmering og software engineering (se ordliste). Opgaven skulle ud-føres i perioden: 5-2-19 til 29-5-19. Gruppen vil samtidig benytte lejligheden til at sige tak til vores vejle-der, Henrik Lykkegaard Larsen og EG's repræsentant for hjælpe opgaven på vej.

Navn

Underskrift

Aleksander Grzegorz Duszkiewicz

Aleksander Henriksen

Aslak Mangino Nicolajsen

Mathias Jakobsen

Per Terp Larsen

Steffen Vitten Pedersen

# Indhold

<b>I Resume</b>	<b>1</b>
<b>II Forord</b>	<b>2</b>
<b>III Læsevejledning</b>	<b>5</b>
<b>IV Redaktionelt</b>	<b>5</b>
Ordliste . . . . .	8
<b>1 Indledning</b>	<b>9</b>
1.1 Resume af case . . . . .	9
1.2 Problemformulering . . . . .	10
1.3 Formål med projekt . . . . .	11
1.4 Målene med projektet . . . . .	11
<b>2 Metoder</b>	<b>11</b>
2.1 Samlede metoder . . . . .	12
2.1.1 Anvendelse af UP og par produktion i projektet . . . . .	13
2.1.2 Kombination af Scrum og UP . . . . .	13
2.1.3 Planlægning . . . . .	14
<b>3 Hovedtekst</b>	<b>15</b>
3.1 Overordnet kravspecifikation . . . . .	15
3.2 Detaljeret kravspecifikation . . . . .	16
3.3 Første iteration . . . . .	17
3.3.1 Analyse . . . . .	18
3.3.2 Design . . . . .	20
3.3.3 Implementering . . . . .	24
3.3.4 Test . . . . .	25
3.4 Anden iteration . . . . .	27
3.4.1 Analyse . . . . .	27
3.4.2 Design . . . . .	31
3.4.3 Implementering . . . . .	38
3.4.4 Unit test . . . . .	42
3.4.5 System test . . . . .	44

<b>4 Diskussion</b>	<b>47</b>
<b>5 Konklusion</b>	<b>48</b>
<b>6 Perspektivering</b>	<b>49</b>
<b>7 Litteraturliste</b>	<b>50</b>
<b>B Procesrapport</b>	<b>1</b>
1 Læring og refleksion . . . . .	1
2 Projektstyring . . . . .	2
3 Identifikation, analyse og bearbejdning af problemer . . . . .	4
3.1 Misforståelser . . . . .	4
3.2 Faglighed . . . . .	4
4 Udviklingsprocessen . . . . .	4
5 Formidling og kommunikation . . . . .	5
6 Samarbejde i gruppen . . . . .	5
7 Samarbejde med vejleder . . . . .	6
<b>C Kildekode</b>	<b>7</b>
<b>D Brugervejledning</b>	<b>7</b>
<b>E Projektlog</b>	<b>17</b>
<b>F Interne bilag</b>	<b>I</b>
1 Projektforslag . . . . .	I
2 Inceptionsdokument . . . . .	XIV
3 Rapportkontrolskema . . . . .	LXVII
4 Diverse interne materialer . . . . .	LXXV
<b>G Eksterne bilag</b>	<b>XCVIII</b>
1 Case . . . . .	XCVIII

## III Læsevejledning

Denne rapport er skrevet ud fra antagelsen om at læseren har en grundlæggende forståelse for programmering og software engineering, som svarer til gruppens eget vidensniveau. Det er meningen, at rapporten skal kunne læses igennem fra indledning til perspektivering. Indledning og konklusion kan i øvrigt læses i sammenhæng for at få en klar forståelse for projektets mål og de opnåede resultater. Ved læsning af første og anden iteration er de begge delt op i lignende afsnit. Afsnittene kan læses for en iteration ad gangen eller det ene afsnit fra første iteration og det samme afsnit fra anden iteration. Anden del af rapporten er om, hvad gruppen har lært og hvordan gruppen har arbejdet og hvordan samarbejdet har fungeret. For at se kildekoden anbefales det at se Javadoc i projektet. Der er også en brugervejledning der kort fortæller hvordan programmet skal bruges. Referencer til gruppens eget arbejde, som ikke kan ses på den pågældende side, kan findes i det interne bilag, mens øvrige bilag vil være i eksterne bilag. Dette er gældende for alle afsnit af rapporten.

## IV Redaktionelt

For dette skema vil følgende personer stå under punkter:

- Aleksander D
- Aleksander H
- Aslak
- Mathias
- Steffen
- Per
- Alle - Indholder alle personer

Afsnit	Ansvarlig	Bidrag fra	Kontrolleret af
Omslag	Aleksander H Per		Alle
Titelblad	Aleksander H Per		Alle
I. Resumé	Per		Alle

II. Forord	Per		Alle
Indholdsfortegnelse	Per		Alle
III. Læsevejledning	Per		Alle
IV. Redaktionelt	Per		Alle
Ordliste	Aslak		Alle
1. Indledning	Aleksander H		Alle
1.1 Problemformulering	Aleksander H		Alle
2. Metoder	Aleksander D Steffen		Alle
2.1 Samlede metoder	Aleksander D Steffen	Aslak	Alle
3. Hovedtekst	Per		Alle
3.1 Introduktion	Aslak		Alle
3.2 Overordnet kravspecifikation	Aleksander H Per		Alle
3.3 Detaljeret kravspecifikation	Aleksander H Per		Alle
3.4 Første iteration	Per		Alle
3.4.2 Analyse Statisk	Aslak Mathias		Alle
3.4.2 Analyse Dynamisk	Aleksander D Steffen		Alle
3.4.3 Design Statisk	Aslak Mathias		Alle
3.4.3 Design Dynamisk	Aleksander D Steffen		Alle
3.4.4 Implementering	Aleksander H Per		Alle
3.4.5 Test	Per		Alle
3.5 Anden iteration	Per		Alle
3.5.2 Analyse Statisk	Aslak Mathias		Alle
3.5.2 Analyse Dynamisk	Aleksander D Steffen		Alle
3.5.3 Design Statisk	Aslak Mathias		Alle

3.5.3 Design Dynamisk	Aleksander D Steffen		Alle
3.5.4 Implementering	Aleksander D Steffen		Alle
3.5.5 Test	Aleksander Steffen		Alle
System Test	Aleksander H		Alle
4. Diskussion	Aslak Mathias		Alle
5. Konklusion	Per	Mathias	Alle
6. Perspektivering	Aleksander D Steffen		Alle
7. Litteraturliste	Per	Alle	Alle
B. Procesrapport	Per		Alle
1. Læring og refleksion	Aslak Mathias		Alle
2. Projektarbejde	Aslak Mathias		Alle
3. Projektforløb	Aleksander H Per		Alle
C. Kildekode	Alle		Alle
D. Brugervejledning	Aleksander D Steffen		Alle
E. Projektlog	Alle		Alle

## Ordliste

- VOP  
Videregående Objektorienteret Programmering.
- VUM  
Voksen udredningsmetoden.
- DHUV  
Digitalisering af Handicap og Udsatte Voksne.
- UP  
Unified Process (metoder), forberedelsesfase (inception), etableringsfase (elaboration), konstruktionsfase (construction) og overdragelsesfase (transition).
- Forretningsdomænet  
Omstændighederne som omhandler opgaven.
- Problemdomænet  
Forholder sig til omstændighederne der kan observeres i den virkelige verden, med udgangspunkt i forretningsdomænet.
- Afdelingsautentificering  
godkendelse af korrekt afdeling.
- Software Engineering  
System Udvikling.
- Ydelser  
En given service, som tilbydes baseret på en paragraf.
- Sikkerhedsforanstaltninger  
blev udviklet for at sikre brugere ikke sletter noget uden at mene det eller henter information om noget de ikke må.
- foreign key (fremmed nøgle)  
Der bruges navnet på den primære nøgle, samt navnet på table navnet til at navngive fremmed nøglen  
Table navn = "department"  
primære nøgle = "departmentid"  
Fremmed nøgle = "departmentdepartmentid"

# 1 Indledning

EG Team Online har været med til at udarbejde en case (ekstern bilag 1), hvori de ønsker en belysning af 3 moduler i deres system, Sensum. Gruppen har af de 3 moduler valgt at lægge vægt på modulet sagsudredning. Udviklingen af modulet vil følge arbejdsprocessen UP (Se metode afsnit 2.1). Dette modul er bygget til sagsbehandlere hos kommunerne og er opbygget efter VUM's retningslinjer. (Anon., 2019) Denne metode bliver brugt til at visitere voksne med et handicap eller misbrug til ydelser, der kan hjælpe dem i deres hverdag.

## 1.1 Resume af case

(ref til case) Gruppen har fået udleveret en case som omhandler systemerne Sensum og Sensum Bosted som er udviklet af EG Team Online. Sensum Bosted og Sensum er systemer der understøtter det samlede behov for dokumentationsstyring i den offentlige og private social- og sundhedssektor.

Systemet har hidtil været opdelt i to forskellige systemer det ene er Sensum Bosted og det andet er Sensum, begge dele er modulopbygget plug-and-play, som gør det muligt at tilpasse systemet til kundens behov. Sensum Bosted er det system som har fokus på at digitalisere den daglige arbejdsproces på bostederne, og understøtter alle former for dokumentationsbehov på socialområdet. Sensum er et udredningssystem der skaber et overblik over sagsbehandlingen. Systemet er et DHUV-system ”Digitalisering af Handicap og Udsatte Voksne”, som er en proces beskrevet under VUM (Anon., 2019)

Virksomheden ønsker en belysning af tre moduler: planlægning, sagsudredning og dagbog.

Planlægningsmodulet er et værktøj hvor et botilbud kan tilrettelægge en borgers hverdag igennem en illustration med pictogrammer, som beskriver de aktiviteter borgeren skal nå den pågældende dag. Sagsudredningsmodulet er den del af systemet hvori sagsbehandlere på en kommune udreder borgerne jfr. VUM (Anon., 2019), og tildeler ydelser som f.eks. behandling på et misbrugscenter eller botilbud for handicappet.

Dagbogsmodulet bruges til journalføring på borgerne. Der bliver både skrevet i journal under udredningen og på botilbuddet som daglig notatføring. Dele af dagbogen bliver automatisk genereret af system, det er f.eks. når borgeren er tildelt en ydelse eller når der bliver udleveret medicin.

## 1.2 Problemformulering

Gruppen har diskuteret afgrænsning i forhold til de specifikke moduler. I forhold til persondatalovgivningen har virksomheden beskrevet en dataafgrænsning i casen hvor de beskriver hvem der må kunne interagere med hvad i systemet. På baggrund af de strengere regler om persondata, blev det nødvendigt at opdatere systemet.

**Gruppen kom frem til følgende hovedspørsgsmål.**

Hvordan kan et program håndtere flere roller uden at de interagerer med hinandens data samt hvilke sikkerhedsforanstaltninger ligger til grund for at beskytte data?

### Delspørsgsmål

På baggrund af den centrale problemstilling er der blevet udformet følgende underspørsgsmål:

- Hvad definerer roller?
- Hvilket ansvarsområder ligger til grund for rollerne?
- Hvilken persistent data bearbejdes?
- Hvordan håndterer systemet flere roller?
- Hvordan håndteres roller med flere ansvarsområder?
- Hvilke sikkerhedsforanstaltninger er der i forhold til data afgrænsningen?
- Hvordan sikres det at aktøren kun har adgang til det aktøren har brug for?

Gruppen analyserede den udleverede projektcase. På baggrund af analysen dannede gruppen et overblik over problemstillingen. Analysen gav en forståelse for de forskellige aktørers roller og ansvarsområder. EG Team Online ønsker en belysning af 3 moduler, hvor gruppen valgte dataafgrænsningen i forhold til sikkerhedsforanstaltninger, der adskiller de forskellige roller.

## 1.3 Formål med projekt

Formålet med projektet er at udvikle et udkast til et eller flere af de moduler som er beskrevet i den udleveret case. Her har gruppen så valgt at ligge vægt på modulet sagsudredning hvor fokuspunktet er at lavet et system med en dataafgrænsning.

Gennem processen med udviklingen af modulet skal der blive arbejdet med softwareudviklingsmodellen UP. Der er i løbet af projektet blevet udarbejdet arkitektoniske og analytiske modeller, samt designmodeller til at understøtte tanken med systemet. Disse diagrammer er blevet udarbejdet på baggrund af den kravspecifikation som gruppen er nået frem til.

## 1.4 Målene med projektet

Der ligges primært vægt på teorien fra Organisation og Grundlæggende Software Engineering. Projektet går hovedsageligt ud på at analyser ansvarsområdet, med henblik på at identificere strukturer og organisatoriske processer. For at udvikle det modul som gruppen er kommet frem til er der blevet arbejdet med UP. Der er i forlængelse af UP blevet brugt dele af Scrum. (Se afsnit 2.1)

I udviklingen af modulet, sagsudredning, skal der indrages elementer fra Videregående Objektorienteret Programmering(VOP) og Database, samt arbejdes ud fra en 3-lags arkitektur, hvilket gør at systemet består af et præsentations-, domæne- og persistenslag. For at udarbejde præsentations- og domænelaget skal der bruges dele af teorien fra VOP. For at udarbejde persistens laget skal der bruges dele af teorien fra både VOP og Database, hvor Database ligger vægt på database design.

## 2 Metoder

Metoder er en vigtig del af et projekt, som udformer væsentlige resultater og bruges til at holde styr på projektet. Der vil under metodeafsnittet beskrives de metoder som har været anvendt under hele projektet, som f.eks. UP, kanban og par produktion. (Se afsnit 2.1)

Unified Process (forkortet UP) er en iterativ, inkrementel og brugsmønster-drevet softwareudviklingsmodel. UP bliver brugt med en kombination af den agile metode Scrum der er en trinvis og iterativ tilgang til udarbejdelse af et produkt. Gruppens anvendelse og kombination af disse beskrives i afsnit 2.1.2. Til sidst bliver der uddybet planlægningen af elaborationsfasen, m.m. (Se afsnit 2.1.3)

## 2.1 Samlede metoder

I starten af projektet blev der holdt et møde omkring hvordan og hvilke metoder der kunne og skulle anvendes. I løbet af dette møde blev det forklaret hvilke metoder hvert enkelt gruppemedlem havde anvendt i deres tidligere projekter. Det blev diskuteret hvilke metoder der med fordel kunne anvendes i dette projekt og de mest væsentlige og effektive metoder blev valgt. Ydermere blev der stillet krav til anvendelsen af UP og Scrum.

**Unified process (UP)** har været anvendt gennem hele projektet. Det er en blanding af agile og planstyrede metoder. Den består af de 4 faser: Forberedelsesfase (inception), etableringsfase (elaboration), konstruktionsfase (construction) og overdragelsesfase (transition). Projektrammen begrænser projektet i det omfang at inceptionsfasen og elaborationsfasen er de faser der skal arbejdes med.

*Inceptionsfasen* er en kort fase der har til formål at give et overblik over de indhentede krav. Hovedtrækkene i fasen handler om at:

- Forstå hvad der skal bygges.
- Identificere de væsentligste funktionalteter i systemet og beskrive dem.
- Identificere projektets plan og kritiske risici.
- Udvælge udviklingsværktøjer til selve udviklingsprocessen.

*Elaborationsfasen* handler om at analysere problem- og løsningsdomænet (se ordliste) samt tilegne sig en endelig forståelse af hele systemet. Hovedtrækkene handler om at:

- Brugsmønsterrealisering.
- Producere en arkitektonisk grundlinje for systemet som skal udvikles.

**Kanban** er en mindre formel metode end Scrum der bliver brugt til at holde styr på, hvilke opgaver der skal laves og hvem der arbejder på opgaverne gennem hele projektet. Formålet med at anvende kanban i projektet har været at holde styr på de opgaver der skulle bearbejdes i løbet af hver iteration. Den har fungeret effektivt i forhold til opdeling af opgaverne, hvor hvert par kunne blive tildelt en opgave. Værktøjet Gitkraken Glo blev brugt som et visuel kanban board for at holde styr på de forskellige opgaver.

**Par produktion** er en agil metode, som tager udgangspunkt i principperne fra parprogrammering. Metoden som er blevet anvendt under hele projektet, har haft den fordel at udarbejdelsen af opgaverne blev mere effektiv og produktiv.

Par produktion fungerede på den måde at der blev lavet et par af to gruppemedlemmer. Hvert par fik opgaver der skulle udarbejdes indbyrdes. Det vigtige element i metoden var at man støttede hinanden med ideer, forbedringer og korrektioner. Dette styrkede fagligheden samtidig med at man fik en forståelse for teorien.

**Scrum** er ligeledes en agil metode og beskrivelsen af Scrum kan findes i bilag 2 side 28 under metodeafsnittet. I projektgruppen er det blevet besluttet at bruge dele af Scrum. Der benyttes en ceremoni i form af ugentlige Scrum-møder. Det er ikke hele Scrum frameworkt, der er blevet anvendt.

### 2.1.1 Anvendelse af UP og par produktion i projektet

UP er den metode som har været anvendt med henblik på at lære hvordan metoden fungere både teoretisk og praktisk. Inceptionsfasen var den første fase, hvor gruppen diskuterede projektcasen og skabte overblik over alle moduler, hvor der blev udvalgt og sat fokus på sagsudredningsmodulet. Den væsentlige funktionalitet blev identificeret og der blev skabt en forståelse for forretningsdomænet. Der blev indhentet krav på baggrund af casen (Se ekstern bilag 1) og undersøgelse af VUM, som blev uddybet på virksomhedsmøder med EG Team Online. Disse krav blev modelleret ved at anvende softwareudviklings principper (Se ordliste). Fordelen ved anvendelsen af metoderne i denne fase var at den skabte overblik over projektet samt at gruppen fik udarbejdet et struktureret inceptionsdokument, der beskrev kravspecifikationen. Ydermere blev inceptionsdokumentet udarbejdet med henblik på at få en godkendelse til at fortsætte yderligere arbejde med produktet. Kombination af UP med par produktion har været en væsentlig faktor i realiseringen af projektet.

### 2.1.2 Kombination af Scrum og UP

Hele projektforløbet har været baseret på UP og dens faser, med et forsøg på at benytte Scrum til styring af elaborationsfasens iterationer. Gennem første iteration blev det forsøgt med dele af Scrum og der blev f.eks. lavet en product backlog (Se figur 2.1). Til at starte med mødtes gruppen dagligt, men da der ikke blev arbejdet med projektet hver dag, valgte gruppen at reducere det til to ugentlige møder.

## 2.1.3 Planlægning

Der blev udarbejdet en product backlog baseret på de væsentlige funktionelle og ikke-funktionelle krav og detaljerede brugsmønstre (Se figur 2.1).

Product Backlog			
Krav	Kategori	Status	Prioritet
Sagsforløb - sagsoprettelse	Brugsmønster	Færdig	Must have
Sagsforløb - Sagsbehandling	Brugsmønster	Begyndt	Must have
Sagsforløb - Sagsafslutning	Brugsmønster	Ikke startet	Must have
Sagsforløb - Sagsredigering	Brugsmønster	Ikke startet	Must have
Sagsforløb - sagsafgørelse	Brugsmønster	Ikke startet	Must have
Sagsbehandlers adgang	Sikkerhed	Færdig	Must have
Hånetering af data	Database	Færdig	Must have
Sagsforløb - Find sag	Brugsmønster	Færdig	Should have
Kommunikation mellem moduler	Modularitet	Færdig	Should have
Sagsforløb - Opfølgning af sag	Brugsmønster	Ikke startet	Could have
Borgers sagsbehandler	Sikkerhed	Ikke startet	Could have
Flere sager pr borger	Flexibilitet	Ikke startet	Could have
Flere ydelser pr borger	Flexibilitet	Ikke startet	Could have
Sagsbehandler åbning af flere sager	Flexibilitet	Ikke startet	Could have
Sensum - Login	Brugsmønster	Færdig	Could have
Sensum - log ud	Brugsmønster	Ikke startet	Could have
Sensum - moduler	Brugsmønster	Ikke startet	Could have

Figur 2.1: Gruppens product backlog

Baseret på to iterationer har gruppen valgt at 1. iteration skulle være general funktionalitet og når gruppen nåede 2. iteration, skulle der implementeres GUI og database. Planlægningen skete ud fra kanban board, hvor der blev oprettet opgaver som hvert par blev tildelt og herefter blev opgaven udarbejdet. Generelt har gruppen ikke fundet Scrum særlig anvendelig i projektet. Dette skyldes flere årsager. I forhold til rollefordelingen, så har gruppen ikke haft en bestemt product owner for projektet. Dette skyldes at gruppen ikke var et fuldendt team, hvilket Scrum kræver. Der har heller ikke været benyttet en scrum master fordi gruppen ikke har erfaring nok til at anvende teorien og generelt har der ikke været behov for det. Dermed har gruppen valgt at undlade rollefordeling. Ydermere blev der udarbejdet en product backlog, men uden en product owner kunne den ikke blive opdateret løbende. Til det formål var det bedre og mere effektivt at anvende kanban metoden til at holde styr på hvad der blev udarbejdet og hvad der manglede. Som en del af iterationerne har fokus ligget på de forskellige faser og iterationer, hvor gruppen koncentrerede sig om den generelle funktionalitet i første iteration.

og GUI og database i anden iteration. Det konkluderes, at gruppen har anvendt for meget tid på at forsøge at implementere Scrum i projektet i forhold til udbyttet og udarbejdelsen af en generel oversigt. Som tidligere nævnt blev der udarbejdet en product backlog men uden en product owner til løbende at opdatere den, har gruppen ikke fundet grund til at opdatere det, da det ville være spild af tid og dermed fjerne fokus fra udarbejdelsen af produktet.

### 3 Hovedtekst

I løbet af projektets to faser, inception og elaboration, har der været en løbende udvikling af projektets planlægning og produkt. Denne udvikling er sket på baggrund af ny viden, som gruppen har indhentet fra virksomhedsmøder, relateret undervisning og professionel vejledning.

En projektrammeplan er blevet brugt til at danne grundlaget for projektets planlægning. Den generelle projektrammeplan indeholdte fire afleveringer, samt grundlæggende milepæle, der skulle suppleres med gruppens egne mål. (Se interne bilag figur F.29)

Første aflevering var gruppens projektforslag, som skulle bestå af de første ideer om problemstillingen i forhold til, hvordan den skulle formuleres, afgrænses og takles. (Se interne bilag 1)

Anden aflevering var et Inceptionsdokument, som bestod af en grundlæggende analyse af projektet. Dette dokument skulle bl.a. indeholde systemets omfang, overordnede kravspecifikationer og prioritering af krav. (Se interne bilag 2)

Tredje aflevering bestod af det samlede arbejde fra første iteration af elaborationsfasen. Denne aflevering tog udgangspunkt i inceptionsdokumentet og startede med at opdatere projektets overordnede - og detaljerede kravsspecifikationer. Der blev udarbejdet statiske - og dynamiske analysediagrammer, samt statiske designdiagrammer. Ud fra disse planer, blev der udviklet et stykke software, som skulle være første skridt på vejen, mod projektets endelige kildekode.

Fjerde aflevering bestod af det samlede arbejde fra anden iteration af elaborationsfasen. Denne aflevering var en fortsættelse af første iteration, hvor der var fokus på at runde projektet af. Alle analyse- og designdiagrammer blev opdateret og der blev suppleret med et dynamisk designdiagram. Der var særligt fokus på at implementere den funktionalitet i kildekoden, som blev prioriteret højest, i forhold til fremtidig systemudvikling.

Det kommende afsnit vil gå i dybden med arbejdet og resultatet af det, som er blevet udført i første og anden iteration.

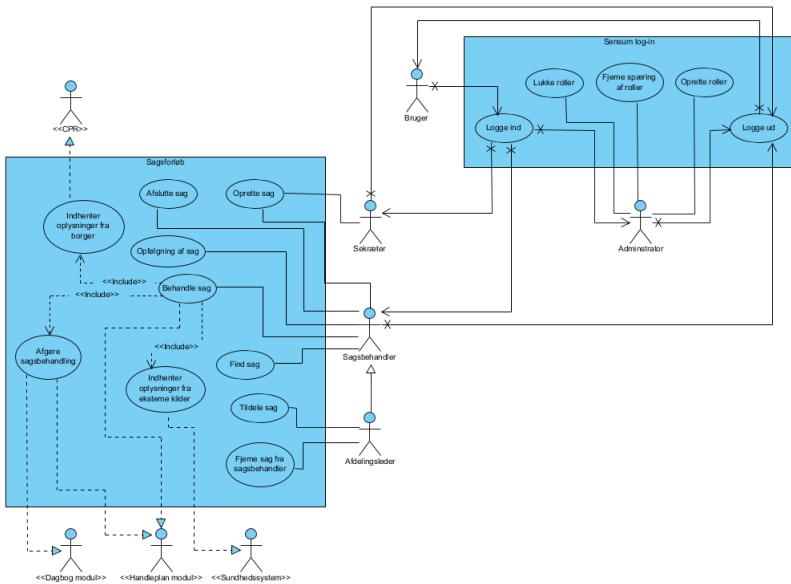
#### 3.1 Overordnet kravspecifikation

Figur 3.1 viser to forskellige systemafgrænsninger. Den ene systemafgrænsning er et login modul, som gruppen har valgt at simulere, da det originale Sensum har et login system tilknyttet.

Den anden systemafgrænsning, som er sagsforløb er det modul som gruppe har lagt fokus i. Her er der lagt størst fokus på opret sag og find sag, som er essentiel for systemet, da en sagsbehandler skal

kunne oprette en sag på en given borger, inden den videre behandling kan forsætte. Dertil er det under sagsudredningen af borgerne også essentiel for en sagsbehandler at kunne gå ind i systemet og nemt finde en borger der er oprettet en sag på.

Den brugsmønstermodel som blev udarbejdet i løbet af inceptionsfasen, har gruppen arbejdet ud fra, gennem elaborationsfasen. Den indeholder de brugsmønstre som gruppen har fundet nødvendige for at få netop modulet sagsudredning til at fungere.



Figur 3.1: Overordnede brugsmønsterdiagram. For fuld størrelse se interne bilag afsnit 4 figur F.9

## 3.2 Detaljeret kravspecifikation

Ud fra den overordnede brugsmønsterdiagram (Se figur 3.1) er der blevet udvalgt forskellige brugsmønster, som er set som mere komplicerede brugsmønster end resten. Der vil blive lagt vægt på to krav i dette afsnit. Disse krav er opret sag og find sag. For at se alle af de detaljeret brugsmønstre se interne bilag figur F.1 til figur F.6

### Opret sag

Som vist på figur 3.2 er der ikke noget specielt svært ved dette brugsmønster. I gennemgang af dette kom der mange forskellige ting op, omkring hvad der ville være fornuftigt at skulle bruges til at oprette sagen. Det endte ud med at der skulle bruges en person, som sagen omhandle samt en grund til at sagen skulle eksistere.

### Find sag

Brugsmønsteret find sag (Se figur 3.3) er til for at finde en vilkårlig sag. Dette skal kunne ske ud fra sagsnummer fra sagen, navnet på personen sagen omhandler og deres CPR-nummer. Det er vigtigt at brugsmønsteret kan åbne en specifik sag, eller stoppes hvis ikke det er den rigtige søgning. Alle krav

<b>Brugsmønster:</b> Opret sag
<b>ID:</b> 1
<b>Primære aktører:</b> Sagsbehandler, afdelingsleder, administrativt personale
<b>Sekundære aktører:</b> CPR
<b>Kort beskrivelse:</b> En aktør kan oprette en sag, som gemmes i systemet.
<b>Prækonditioner:</b> Aktør skal være logget ind.
<b>Hovedhændelsesforløb:</b> Starter når en borger henvender sig til kommunen. 1. Aktør indtaster CPR nummer, borgers navn, begrundelse for henvendelse. 2. Gemmer indtastet data.
<b>Postkonditioner:</b> sag oprettet
<b>Alternative hændelsesforløb:</b>

Figur 3.2: Brugsmønster for opret sag

kan findes i interne bilag afsnit 4 figur F.7 og figur F.8

<b>Brugsmønster:</b> Find sag
<b>ID:</b> 3
<b>Primære aktører:</b> Sagsbeandler
<b>Sekundære aktører:</b> none
<b>Kort beskrivelse:</b> Skal kunne søge i sager og få en sag vist
<b>Prækonditioner:</b> Der skal være oprettet en eller flere sager
<b>Hovedhændelsesforløb:</b> Starter når den primære aktør skal finde en sag. 1. Skal søge en sag på sagsnummer, CPR nummer, eller navn. 2. Vise en liste over de sager som blev fundet. 3. Valgt sag kan åbnes.
<b>Postkonditioner:</b>
<b>Alternative hændelsesforløb:</b>

Figur 3.3: Brugsmønster for find sag

### 3.3 Første iteration

Formålet med elaborationsfasen er at skabe en prototype af produktet MMMI (Se ordliste). Denne udvikling sker på baggrund af inceptionsfasen (Se interne bilag afsnit 2), hvor der blev opstillet en række krav til projektet. For at kunne nå de mål bliver der lavet nogle mindre iterationer herunder analyse, design, kode og test som udarbejdes, basseret på den overordnede kravspecifikation.

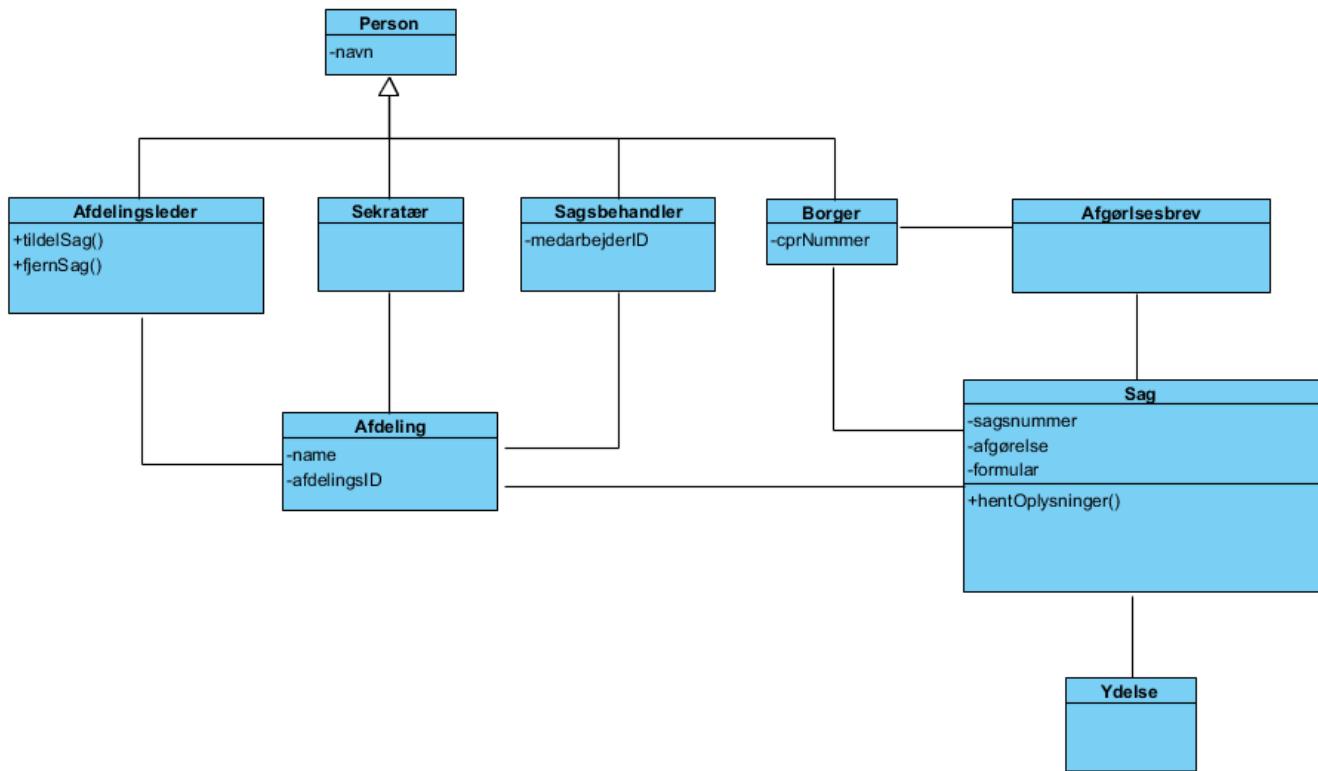
### 3.3.1 Analyse

Der er udarbejdet en række diagrammer over de to detaljerede brugsmønstre.

Der er to forskellige typer af analysediagrammer, der er de dynamiske og de statisk.

#### Statiske side af analysen

Der er udarbejdet et klassediagram (Se figur 3.4) som består af 9 klasser: Afdeling, Sag, Afgørelsесbrev, Ydelse, Person, Sekretær, Sagsbehandler, Afdelingsleder og Borger. De to mest centrale klasser i diagrammet er Afdeling og Sag, som skaber grundlaget for systemet.



Figur 3.4: Analyseklassediagram. For fuld størrelse se interne bilag afsnit 4 figur F.11

Afdeling består af attributterne ”navn” og ”afdelingsID”, som blev valgt for at muliggøre identifikationen af en afdeling. Klassen er central for systemet da den skal bruges til at administrere dem som arbejder i en afdeling og hvilke sager de er ansvarlige for.

Sag er en klasse, som består af attributterne ”sagsnummer”, ”afgørelse” og ”formular”, samt metoden ”hentOplysninger()”. Klassen udgør omdrejningspunktet i arbejdet med sagsbehandling, da Sag klassen skal bruges i forbindelse med behandling af borgere. Attributten ”formular”, repræsentere de dokumenter der, ifølge VUM, (Anon., 2019) skal udfyldes af personalet f.eks. sagsbeandler.

Sag er forbundet til Afdeling for at indikere at en sag behandles i en specifik afdeling.

Ydelse er en klasse som repræsenterer hvilke botilbud og lignende som tildeles en borger ud fra deres behov. Ydelse er forbundet til Sag, da den tildeles baseret på en vurdering af en specifik sag. En borger kan modtage flere ydelser, men disse ydelser er alle baseret på de specifikke sager som berører den enkelte borger.

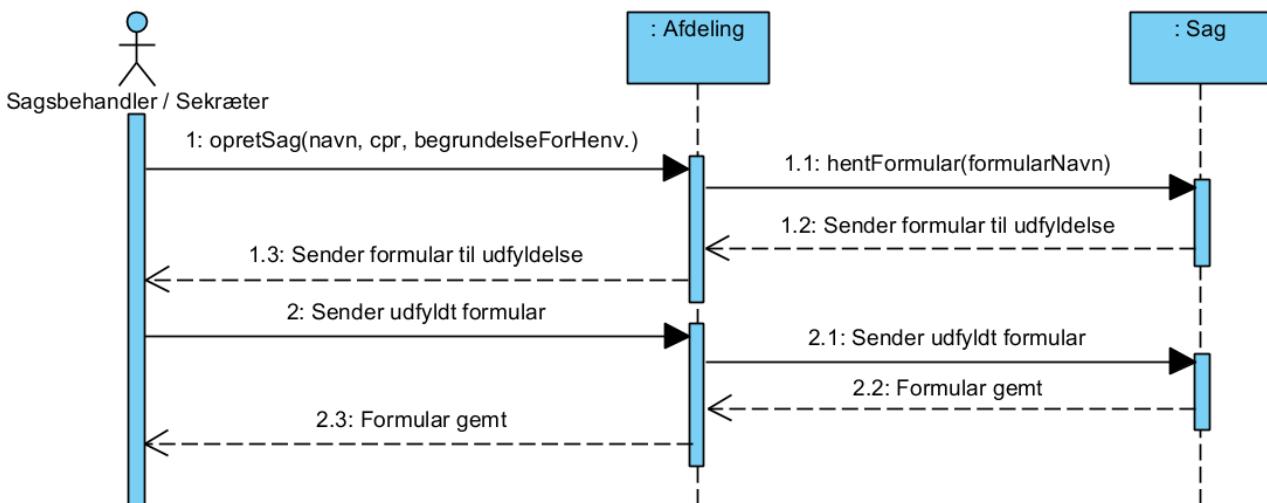
Borger klassen repræsenterer en person som skal registreres i systemet for at modtage et givent tilbud. Borger klassen har attributten ”cpr-nummer”, som bruges til at identificere den borgeren. Yderligere arver Borger klassen fra Person, som giver attributten ”navn”.

## Dynamiske side af analysen

### Opret sag

Beskrivelse:

Som man kan se på figur 3.5 startes opret sag ved at en aktør anmelder systemet om at oprette en sag. Afdelingen håndterer operationen ved at hente en formular ved at sende forespørgsel til Sag. Den sender sagsåbningsformularen tilbage til afdeling. Når afdelingen modtager formularen, videresendes den til den pågældende aktør. Aktøren udfylder formularen med relevante oplysninger, hvorefter den sendes til afdelingen, hvor denne sender det videre til den pågældende sag for opdatering.

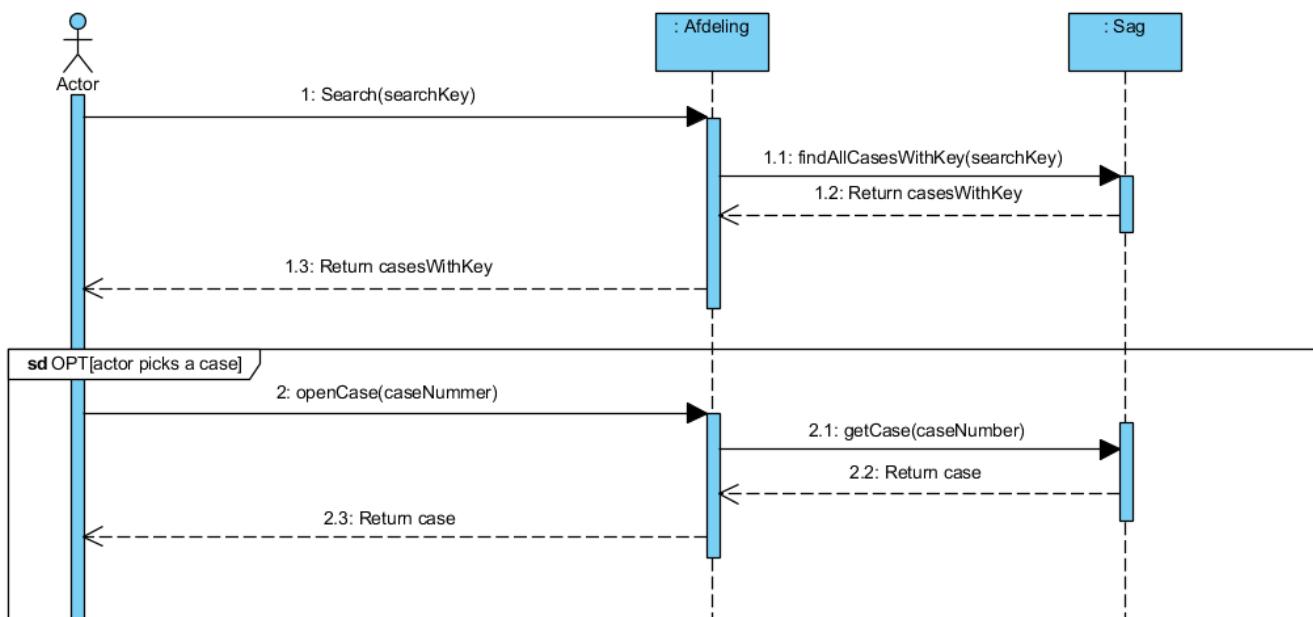


Figur 3.5: Sekvensdiagram for opret sag

### Find sag

## Beskrivelse:

En aktør, f.eks. sagsbehandler eller sekretær, vælger at søge på en sag, som vist i figur 3.6. Search metoden modtager en nøgle værdi. Den sender besked ned til afdelingen og herefter spørger en metode efter at finde alle sager med den specifikke nøgleværdi. Sager med den specifikke nøgle sendes tilbage til afdelingen og afdelingen sender resultatet videre til aktøren. Herefter vælger aktøren at åbne den specifikke sag, hvor der kaldes på openCase metoden som sender besked ned til afdelingen. Afdelingen efterspørger herefter om at få den sag med det specifikke sagsnummer, og der returneres et sagsobjekt med det specifikke sagsnummer. Afdeling returnerer efterfølgende det fundne sagsobjekt til aktøren.



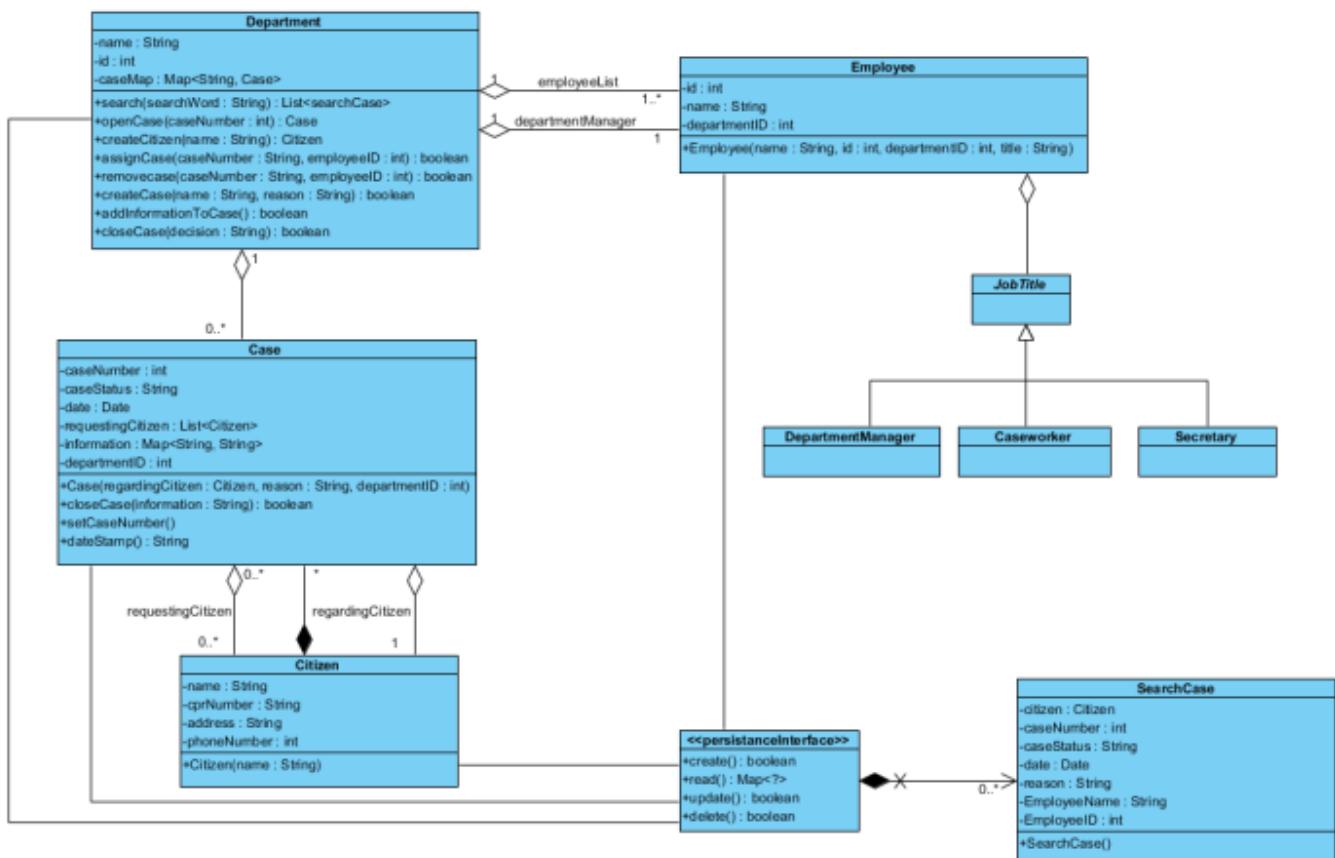
Figur 3.6: Sekvensdiagram for find sag

### 3.3.2 Design

#### Den statiske side af design

Diagrammet i 3.7 beskriver de klasser vi har fundet, der skal implementeres i systemet. Hver klasse har attributter og en del skal have skrevet getter / setter-metoder, men for at diagrammet ikke skal blive for stort og uoverskueligt, valgte gruppen at undlade dem.

Diagrammet er blevet designet med tanker på en tydelig lagdeling og repræsentere systemets domænelag. Det blev besluttet at "Department"-klassen skulle benyttes som facadeklasse til præsentationslaget, mens datahandler interfacet bruges til kommunikation mellem domæne- og persistentslag.



Figur 3.7: Klassediagrammfuld. For fuld størrelse se interne bilag afsnit 4 figur F.11

## Department – Facade klasse

Der var særligt fokus på attributterne: ”id” af typen int og ”caseMap” af typen Map<String, Case>. Det var tænkt at ID skulle bruges til at godkende adgangen til sager i databasen, for at imødekomme projektets ”Dataafgrænsning”. ID er af typen int, for at holde den simpel. Kravet til dataafgrænsningen er, at en ansat der henter information omhandlende en borger, ikke må blive gjort opmærksom på sager, der administreres af andre afdelinger end ansattes egen.

Tanken bag attributten ”caseMap”, var at department skulle have en liste af case-objekter som kunne bruges til at hente de relevante informationer frem for at sende dem til præsentationslaget. Der blev valgt at bruge et Map da Key/Value-strukturen gør det nemmere at finde en sag. Den Key der blev valgt til caseMap er caseNumber og skal bruges til metoden openCase(...).

Klassens metoder, som ”search(...)”, ”openCase(...)”, mm. er designet til at håndtere den logik, som controllerne fra præsentationslaget skal bruge. Metoderne repræsentere de funktioner som er ønsket i det grafiske userinterface (GUI).

Når metoden ”search(...)” kaldes, skal logikken først hente data fra persistenslaget. Dette sker gennem Interfacet PersistanceInterface som opretter et objekt af klassen ”SearchCase”, som er designet til at være en dataklasse i domænelaget, for data der er relevant til en brugers søgeord.

## SearchCase – Dataklasse

Klassen består af en constructor med attributterne: "citizen", "caseNumber", "caseStatus", "date", "reason", "employeeName" og "employeeID". Disse informationer er ikke et udtryk for alt der ligger i en sag, men er nok til at præsentere en borgers sag.

Hvis der er et ønske om at hente den fulde sag, gøres dette ud fra "caseNumber"-attributten gennem "openCase(...)" metoden, fra "Department"-klassen. Den data der er relateret til en sag, hentes fra persistenslaget, gennem interfacet "PersistanceInterface".

## PersistanceInterface – Datahandler interface

For at behandle dataflowet mellem persistenslaget og domænelaget, er der blevet designet et interface med metoderne: "create()", "read()", "update()" og "delete()". Disse metoder blev valgt for at kunne oprette, hente, ændre og slette data, som opbevares i persistenslaget. For at domænelaget kan sende dataforespørgsler til persistenslaget, skal der implementeres klasser med det påkrævede interface i persistenslaget.

## Case : Employee : Citizen – Objektklasser

Objektklasserne blev designet med tanken om at præsentationslaget og persistenslaget, skulle arbejde med konkrete instanser af disse. Persistenslaget skulle tilføje den data, som var forbundet med objektet, som blev kaldt gennem domænelaget. Præsentationslaget skulle registrere indholdet i objekterne, pakke det ud og præsentere det som ønsket.

## JobTitle – Rollebaseret kontrol

Superklassen "JobTitle" blev designet til at kontrollere en given medarbejdernes rettigheder i systemet. Der blev taget udgangspunkt i tre jobstillinger, som blev skrevet op som subklasser - sekretær, sagsbeandler og afdelingsleder. Når en medarbejderklasse blev oprettet, skulle det være et krav, at de fik tildelt en stilling (JobTitle). Denne stilling skulle kunne ændre sig i systemet, f.eks. på grund af en forfremmelse, uden at medarbejderen skulle registreres på ny i systemet.

Stillingen som en medarbejder blev tildelet, skulle begrænse funktionaliterne medarbejder har adgang til. Dette sikrede de kun havde adgang til de områder, som var nødvendig til at udfører deres opgave. En sekretær skulle f.eks. ikke kunne andet end at registrere en person, som søger behandling samt årsagen for at der søges. En sagsbeandler skulle have fuld adgang til relevante sager, dvs. kun sager der er oprettet i den afdeling sagsbeandler er tilknyttet. En afdelingsleder skulle kunne det samme som en sagsbeandler, men også kunne styre hvilke sager de forskellige sagsbeandlerne skulle arbejde på.

## Multiplicitet/Multiplicity – Klassernes forbindelser

Forbindelserne mellem klasserne i designdiagrammet, består primært af aggregeringer. Klassen "Employee" er forbundet med en many-to-one aggregering til klassen "JobTitle". Dette er blevet valgt, da objekter af klassen "JobTitle" ikke skal blive slettet, hvis et "Employee"-objekt fjernes. Tanken der ligger til grund for dette, er at "JobTitle" kan være forbundet til flere forskellige "Employee"-objekter. Klassen "Department" er forbundet til "Employee" med to aggregeringer. Første aggregering er "employeeList", som er en one-to-many relation. Aggregeringen er blevet lavet, for at indikere at der kan være mange der arbejder i samme afdeling, men det skal stadig være muligt at slette en "Department", uden at slette "Employee"-instanserne. Anden aggregering er "departmentManager", som er en one-to-one relation. Denne aggregering er blevet lavet, for at demonstrere at der altid skal være en leder i en afdeling. Igen skal det være muligt at slette en "Department", uden at det sletter "Employee".

Klassen "Department" er forbundet til "Case" med en enkelt aggregering. Dette er en one-to-zero-or-more forbindelse, som betyder at en afdeling ikke altid har nogen sager, men en sag er altid forbundet til en afdeling. Aggregeringen blev valgt, da en sag ikke forsvinder, hvis afdelingen bliver fjernet.

Klassen "Case" er forbundet med to aggregeringer til klassen "Citizen", som også er forbundet med en komposition tilbage igen. Den første aggregering der er lavet, er en zero-or-many-to-zero-or-many relation, som omhandler en sags henvisning. En person kan blive henvist af nul eller flere personer, omhandlende det samme problem og på samme måde er det også muligt for en person at have henvendt sig for at få oprettet nul eller flere sager. Anden aggregering er lavet af samme grund som kompositionen er. En "Citizen" kan have en til flere sager i gang på en gang, men en sag skal altid være angående en enkelt "Citizen". Aggregeringen er anvendt da det skal være muligt at fjerne en "Case" fra systemet, uden at fjerne den "Citizen" sagen omhandler. Kompositionen er derimod blevet brugt, da en "Case" altid vedrører en "Citizen", og hvis denne fjernes fra systemet, skal de relaterede "Case"-objekter fjernes, da de ikke længere vedrører nogen.

Klassen "SearchCase" er forbundet med en komposition fra interfacet "PersistanceInterface". Da et interface ikke kan instantieres, har det ikke nogen multiplicitet. Kompositionen er blevet valgt, da "SearchCase" ikke fungere uden interfacet, og forsvinder derfor, hvis interfacet fjernes.

## Den dynamiske side af design

Der blev ikke arbejdet med den dynasike side af design, da teorien ikke var blevet introduceret i første iteration.

### 3.3.3 Implementering

Målet med første iteration var at skabe et domænelag, der havde de funktionaliteter, som var nødvendige for at systemet kunne fungere. Dette blev gjort så gruppen havde en bedre forståelse for hvordan systemet skulle se ud. Gennem første iteration blev der lavet to klasse (case og department) der implementerede begge brugsmønstre. ”Case” håndterede hvilke data der skulle gemmes i en sag, mens ”department” holdt styr på hvilke specifikke sager der var og hvordan de skulle tilgås. Nedenfor ses de forskellige attributter i ”Case” klassen. (Se figur 3.8)

---

```
public class Case {
    private static int socialCount = 1;
    private static int handicapCount = 1;

    private final String caseNumber;
    private String caseStatus;

    private final String date;
    private final Citizen regardingCitizen;
    private final List<Citizen> requestingCitizen = new ArrayList<>();
    private final Map<String, String> information = new HashMap<>();
    private final int departmentID;
```

---

Figur 3.8: Case klassens attributter

Der bruges et map til at holde den data der ændre og skrives ind.

Klassen indeholder en status på om sagen er lavet, i gang eller færdig.

Der kan kun være én person sagen omhandler, så der er en instans af klassen. Der kan være flere der henvender sig angående den samme person. Dette blev håndteret ved at lave list over disse personer. Den data der kan vises i præsentationslaget kaldes igennem facaden ”Department”. Det var også her at alle de forskellige sager blev gemt i denne iteration. Der bliver genereret et sagsnummer hvor det

---

```
public List<Case> search(String searchWord) {
    List<Case> listCases = new ArrayList<>();

    for (Case searchCase : caseMap.values()) {
        if (searchCase.getRegardingCitizen().getCprNumber().equalsIgnoreCase(searchWord)
            || searchCase.getCaseNumber().equalsIgnoreCase(String.valueOf(searchWord))
            || searchCase.getRegardingCitizen().getName().equalsIgnoreCase(searchWord)) {
            listCases.add(searchCase);
        }
    }
    return listCases;
}
```

---

Figur 3.9: Department search, finder alle sager der har søgeordet i

er muligt at se hvilken afdeling sagen tilhører.

I figur 3.9 kan det ses hvordan klassen ”Department” finder en sag. Den skal bruge en ”string” som kunne være CPR-nummer eller navn for personen den omhandler eller sagsnummeret.

Som det kan ses i koden er bliver der ikke returneret en sag, men en liste af sager.

Dette valgte gruppen at gøre fordi der kan søges på et navn, som ikke er unikt og der kan derfor findes mere end en persons sager. Samtidig er det også muligt at en person har mere end en sag, så selv hvis der kun er én person med et givent navn eller der søges på et CPR-nr kan der stadig returneres mere end én sag. Dette nåede ikke at blive implementeret i denne iteration.

### 3.3.4 Test

---

```

public static void main(String[] args) {
    while (!quit) {
        rollInfor();
        text = sc.nextLine().toLowerCase();

        switch (text) {
            case "secretary":
                Employee trine = detDepartment.createEmployee("trine",
                    252525, "secretary");
                commands(trine);
                break;
            case "caseworker":
                Employee mads = detDepartment.createEmployee("mads",
                    353535, "caseworker");
                commands(mads);
                break;
            case "departmentmanager":
                Employee martin = detDepartment.createEmployee("martin",
                    262626, "departmentmanager");
                commands(martin);
                break;
            case "quit":
                System.out.println("Du quiter");
                quit = true;
                break;
            default:
                System.out.println(text + "Er ikke en kommando\n");
                break;
        }
    }
}

```

---

Figur 3.10: tekst under figuren

Der blev ikke lavet unit test på første iteration, da det ikke blev set som en nødvendighed. Det der blev fokuseret på var systemtest som skulle sikre at forståelsen med projektet og hvordan de forskellige

brugsmønstre ville fungere i et færdigt system. Der blev fundet flere fejl/mangler igennem systemet. Vist på koden (Se figur 3.10) kan det ses at der bliver lavet en ny ”employee” hvilket skyldes det ikke var lavet så den kunne huske mere end en ad gangen. Det kan ses I figur 3.11 er det de forskellige kommandoer som kan bruges returneres det vil derefter testes om personen der er logget ind. Dette blev den eneste rigtige test da dette kom ud på at få ideen til hvordan systemet skulle fungere helt på plads så databasen og præsentations laget kunne blive lagt på. Dette blev den eneste rigtige

---

```
public static String loop() {
    while (true) {
        Scanner s = new Scanner(System.in);
        String t = s.nextLine().toLowerCase();

        switch (t) {
            case "back":
                return t;
            case "create_case":
                return t;
            case "close_case":
                return t;
            case "add_information":
                return t;
            case "assign_case":
                return t;
            case "reassign_case":
                return t;
            default:
                System.out.println(t + " Er ikke en kommando\n");
        }
    }
}
```

---

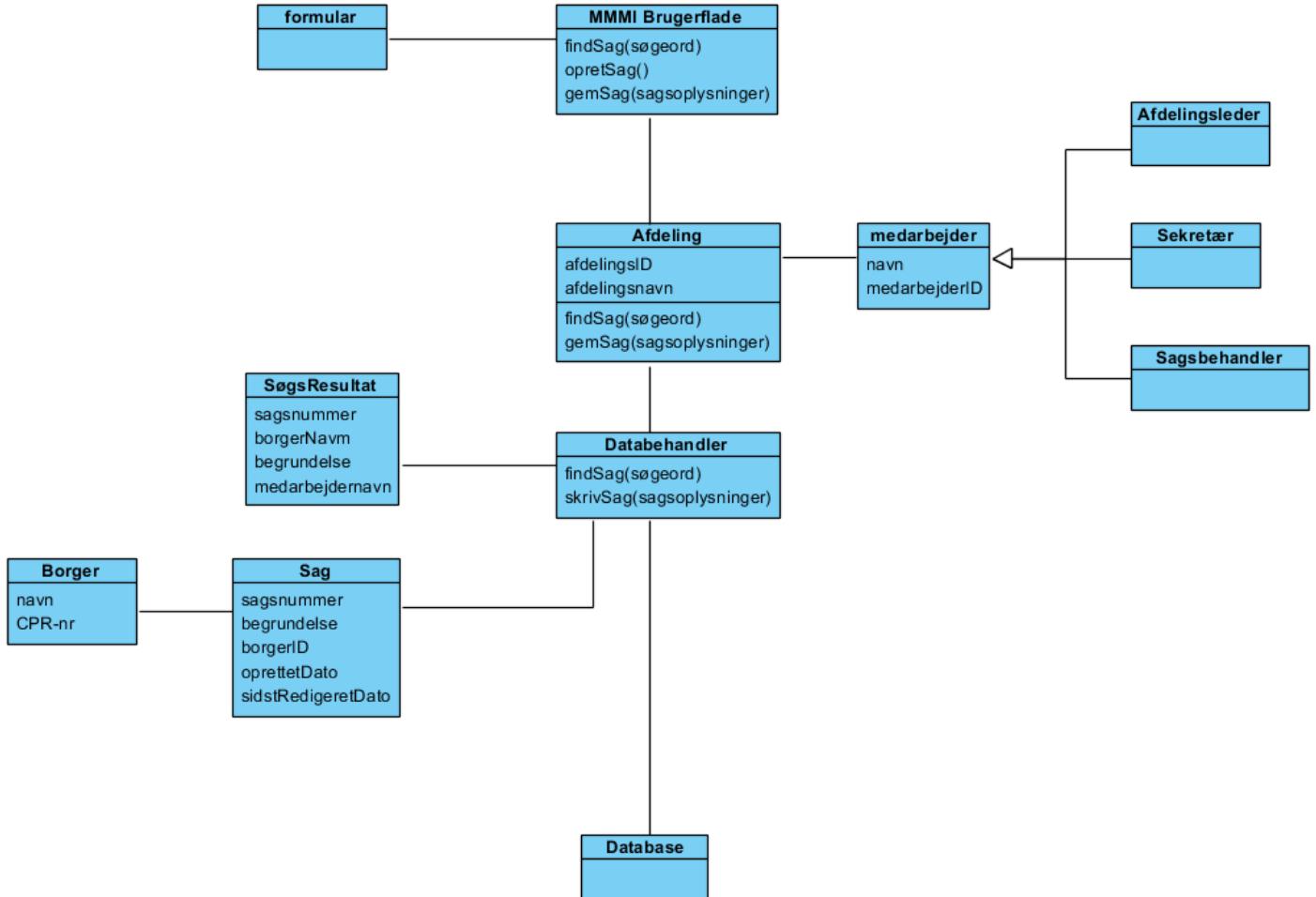
Figur 3.11: loopet der ser om kommandoer kan være rigtig

test da dette kom ud på at få ideen til hvordan systemet skulle fungere helt på plads så databasen og præsentations laget kunne blive lagt på.

## 3.4 Anden iteration

### 3.4.1 Analyse

Den statisk side af analyse



Figur 3.12: Opdateret analyseklassediagram

Da der har været fokus på at få integreret en database og en funktionel grafisk brugergrænseflade i 2. iteration, består de primære ændringer i den statiske analyse, af arbejdet med at revidere klassediagrammet, med henblik på at tilføje persistens og præsentation. Der er bl.a. blevet ændret på hvordan de tre klasser der repræsenterer ansatte, og klassen der repræsenterer borgere, beskrives. Der blev valgt at gå væk fra en overordnet abstrakt klasse (Se figur 3.4) kaldet ”Person”, som ”Borger”, ”Afdelingsleder”, ”Sekretær” og ”Sagsbehandler” klasserne arvede fra. En abstrakt klasse kaldet ”medarbejder” blev lavet, som ”Afdelingsleder”, ”Sekretær” og ”Sagsbehandler” arvede fra i stedet, og holdt ”Borger” som en separat klasse. Dette betød at ”Borger” fik navn og CPR-nr. som attributter,

til forskel fra 1. iteration hvor den kun havde CPR-nr. og arvede navn fra ”Person”. Yderligere blev medarbejderID flyttet til superklassen, da alle arvende klasser skulle bruge den.

Det blev droppet at beholde ”Afgørelsесbrev” som klasse, da det gav mere mening at lægge den i præsentationslagets klasse ”Formular”, som også repræsenterer alle andre formularer der beskrives i VUM.

Klassen ”Sag” er blevet rykket ned til datalaget og kan tilgås gennem Databehandler klassen, som er den klasse der repræsenterer datalaget og skal varetage al kommunikation med databasen.

De vigtigste metoder og attributter, på diverse klasser, er blevet opdateret i forhold til den nye viden, der er blevet indsamlet gennem 2. iteration. Der er blevet lavet nye metoder er på klassen (Se figur 3.12) ”MMMI Brugerflade”, som er ”findSag”, ”opretSag” og ”gemSag”. På klassen ”Afdeling” er metoderne ”findSag” og ”gemSag” blevet lavet og på klassen ”Databehandler” er ”findSag” og ”skrivSag” blevet lavet. De nye attributter som er tilføjet, er ”begrundelse”, ”borgerID”, ”oprettetDato” og ”sidstRedigeretDato” på ”Sag”.

Den nye klasse ”SøgsResultat”, bruges til at holde data samlet i forbindelse med en given søgning af en sag. De attributter som vises i klassen ”SøgsResultat”, repræsenterer en begrænset delmængde af hvad der gemmes når en sag oprettes, men det er nok til at en bruger vil være i stand til at identificere en given sag for at åbne den.

Klassen Database repræsenterer den postgresql database er blevet designet og implementeret i iteration 2 (Se interne bilag figur F.30).

## Den dynamisk side af analyse

I anden iteration er der blevet udarbejdet operationssekvensdiagrammer i forhold til ”opretSag()”, ”opdaterSag()” og ”find()”.

Operationen ”opretSag()” (Se afsnit 3.4.1) og ”opdaterSag()” (Se afsnit 3.4.1) er operationer der stammer fra opretSag() (se figur 3.5) fra første iteration.

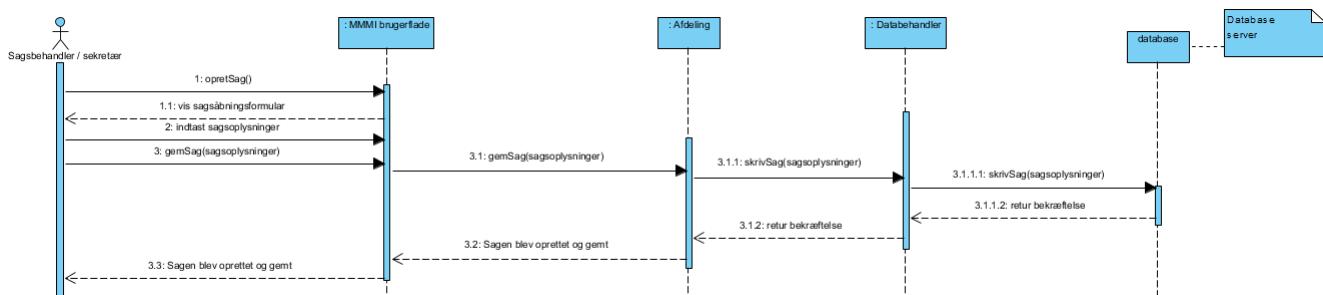
Operationen find() er udviklet ud på baggrund af operationen ”findSag()” (se figur 3.6) fra første iteration.

### opretSag():

Operationen ”opretSag()” er den funktion der opretter en sag i systemet. Der gjort overvejelser om at håndtere opdaterSag() i samme funktion som ”opretSag()”, men det ville ikke give mening. Derfor er de adskilt i to operationer.

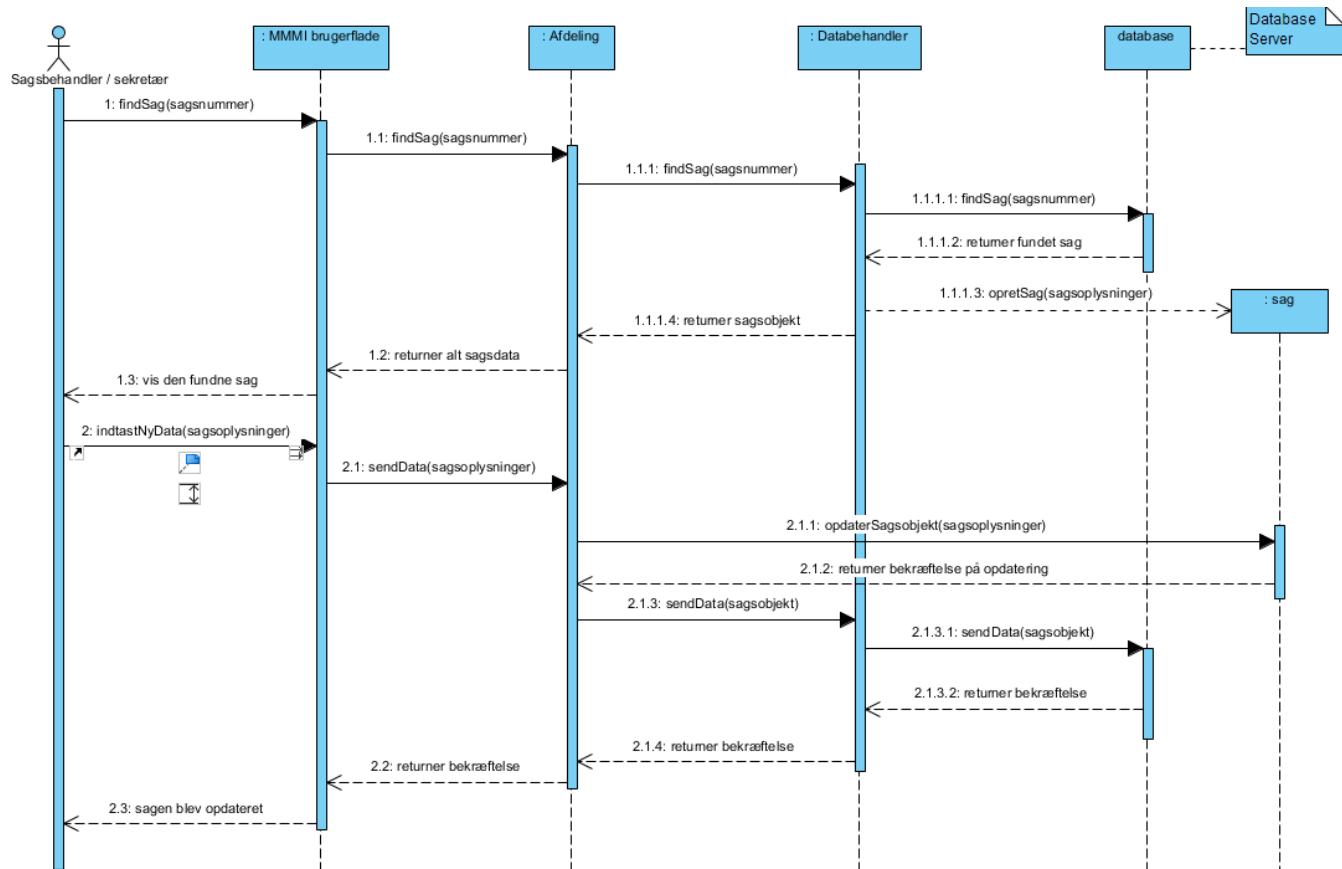
I figur 3.13 begynder funktionen ved at en sagsbehandler eller sekretær vælger ”opretSag()”. Brugergrænsefladen returner en sagsåbningsformular, som skal udfyldes. Når sagsbehandleren har indtastet sagsoplysningerne vælger sagsbehandler at gemme sagen. Operationen ”gemSag()” sender besked til Afdeling som derefter benytter operationen ”skrivSag()” til Databehandleren og til sidst sender Databehandleren informationen videre til databasen. Databasen returner en bekraeftelse til

Databehandleren som returner den gennem Afdeling og brugerfladen som til sidst fortæller aktøren at sagen blev oprettet. Ved at anvende ”opretSag()” operationen har en sagsbehandler eller sekretær



Figur 3.13: Opdateret sekvensdiagram for opret sag. For fuld størrelse se interne bilag 4 figur F.26

mulighed for at vælge en formular og ud fra den gemme og dermed oprette en ny sag. Oprettelsen sker når borgeren henvender sig og dermed kan sekretæren eller sagsbehandleren oprette en sag på den pågældende borger. For at kunne håndtere om en borger allerede har en sag eller om der findes oplysninger om borgen, benyttes ”find()” operation, som kan læses i afsnit 3.4.1



Figur 3.14: Opdateret sekvensdiagram for opdater sag.

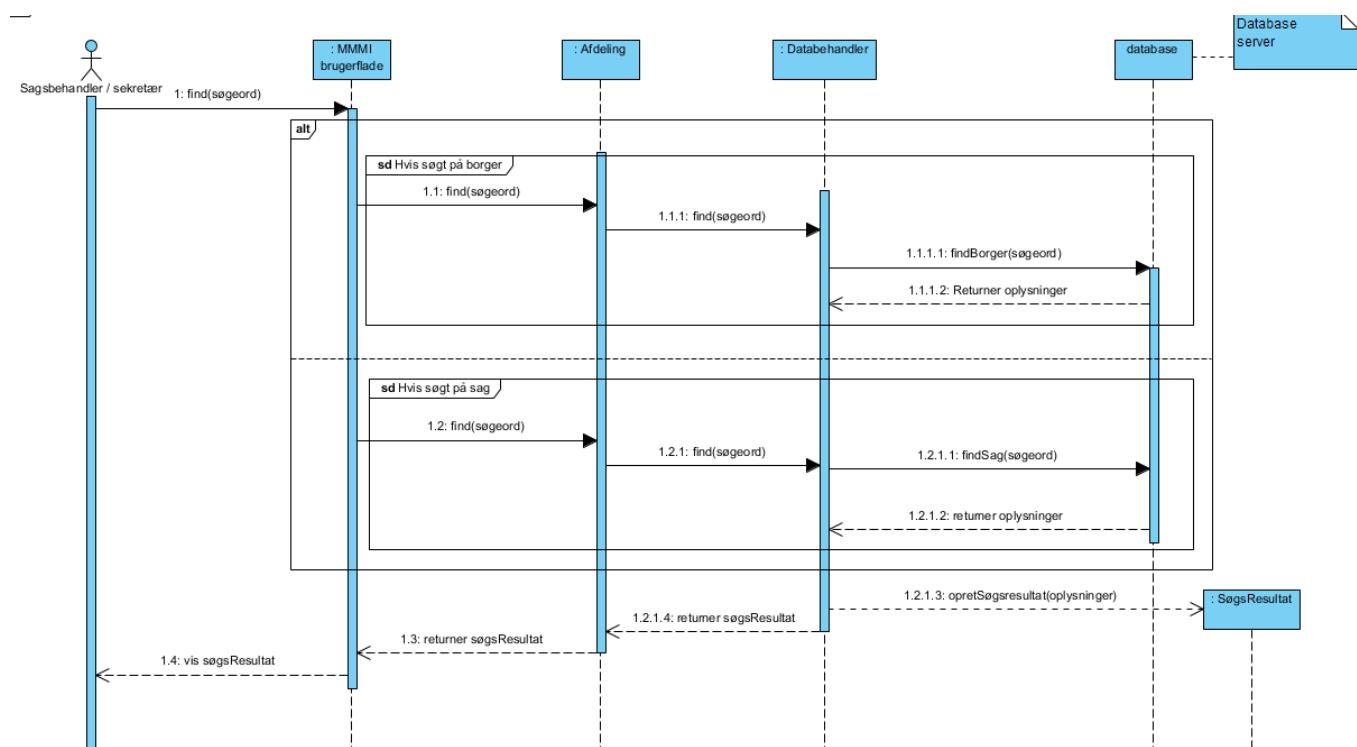
### opdaterSag():

Operationen ”opdaterSag()” i kombination med ”find()” operationen gør det muligt for en aktør at kunne finde en specifik sag og dermed åbne den og opdatere den.

I figur 3.14 ses det at en sagsbehandler eller sekretæren vælger at fremsøge en sag. ”findSag()” beskeden bliver sendt igennem afdelingen og databehandleren. Databehandleren sørger for at sende beskeden til databasen og fremskaffe den søgte sag. Denne returneres til databehandleren og der bliver oprettet et sagsobjekt. Dette sagsobjekt returneres til afdelingen og herefter til brugerfladen som til sidst bliver præsenterer det til brugeren. Når brugeren har indtastet ny data sendes det til afdelingen via operationen ”sendData()”. Afdelingen sørger for at opdatere sagsobjektet gennem beskeden ”opdaterSagsobjekt()” hvor sag returner bekræftelse på opdaterede oplysninger er gemt. Det opdaterede sagsobjekt sendes til Databehandleren gennem beskeden ”sendData()” som derefter sender besked til databasen om at gemme de nye oplysninger. Der returneres en bekræftelse gennem databehandleren, afdelingen og brugerfladen som så bekræfter overfor aktøren at sagen blev opdateret.

### find():

Tanken med ”findSag()” i første iteration afsnit 3.3.1 har været at kunne fremsøge en sag.



Figur 3.15: Opdateret sekvensdiagram for find sag.

I figur 3.15 vælger en sagsbehandler eller sekretær at finde en sag baseret på indtastet søgekriterier. Præsentationslaget sender ”find()” søgeord videre til Afdelingen, som sender det videre til Data-behandleren. Der søges i databasen, hvor det resultat der kommer fra database sendes til Data-behandleren der så, opretter et SøgsResultat objekt, som Databehandleren sender tilbage. Afdelingen sørger for at sende data til brugerfladen som til sidst viser resultatet af søgningen til aktøren.

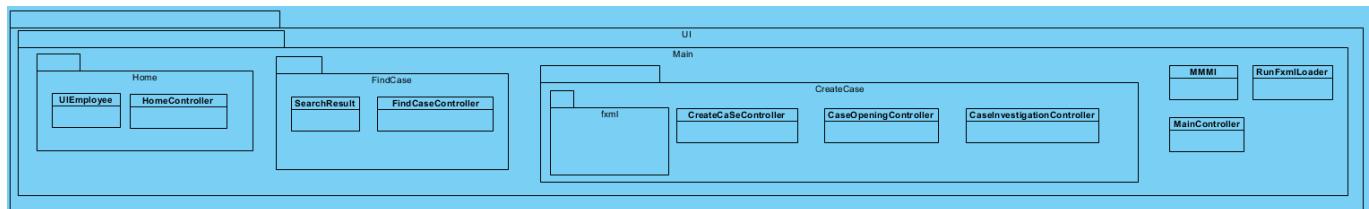
### 3.4.2 Design

#### Den statisk side af dasign

Designdiagrammet i projektets anden iteration, blev opdateret med flere klasser og en tydelig lagdeling. En klar adskillelse af præsentations-, domæne- og persistenslaget er blevet illustreret og indeholder også subsystemernes klasser, med attributter og metoder. Det er valgt at getter og setter metoder ikke vises i diagrammet, da det ville gøre det meget større og sværere at overskue. (Se interne bilag figur F.28)

For at få et samlet overblik se interne bilag 4 figur F.27 og figur F.28

#### Præsentationslaget



Figur 3.16: Der blev valgt at undlade beskrivelsen af kontrollernes indhold og FXML-dokumenterne i subsystemet UI (Præsentationslaget), for at gøre diagrammet mere overskueligt. Præsentationslaget er blevet illustreret med pakker, som indeholder kontrollere og FXML-dokumenter for specifikke funktioner, samt klasserne som styrer systemets GUI.

#### MainController – Præsentations controlleren

Klassen ”MainController”, er systemets primære FXML-handler. Den blev lavet så det var muligt at skifte funktionalitet, på baggrund af et brugerinput, i samme root-node.

#### RunFXMLLoader – Præsentationsmetoden

Klassen ”RunFXMLLoader”, indeholder funktionaliteten som bruges til at skifte FXML-dokumenter. Den bliver arvet af ”MainController”, og implementere, som det eneste, metoden der bruges til at skifte ”pane”, i systemet.

## MMMI – Main Class

Programmet køres gennem MMMI klassen, da det er programmets main class.

### CreateCase – Dokumentstyring og præsentation

Subsystemet ”CreateCase”, blev designet til at begynde oprettelsen af nye sager i systemet og styre præsentationen af sagsdokumenter, samt brugerinteraktionen med disse. I slutningen af anden iteration var ”CreateCase”, pakket med controllerne ”CreateCaseController”, ”CaseOpeningController” og ”CaseInvestigationController”, samt FXML-pakken ”fxml”, som var pakket med de tilsvarende FXML-dokumenter.

Controlleren ”CreateCaseController”, blev designet til at styre bevægelsen mellem sagsdokumenter, samt at gemme brugerindtastet information fra disse dokumenter. Navngivningen kunne virke misvisende, i forhold til at controlleren bruges til at skifte mellem dokumenter, men peger på funktionaliteten bag gem funktionen. Når en given bruger benytter gem funktionen i en sag, sender controlleren de brugerindtastede informationer til domænelaget, for at få det registreret i systemdatabasen.

Hvis en sag gemmes for første gang i systemet, vil informationen som sendes gennem systemlagene, oprette en ny sag i databasen, som derefter modtager informationen. Controlleren er blevet navngivet efter denne proces, men har i sig selv kun at gøre med brugerinput og kommunikationen med domænelaget.

Controlleren ”CaseOpeningController”, er designet til at pakke informationerne indtastet i FXML-dokumentet ”caseOpeningNew”, og derefter sende det videre til ”CreateCaseController”, når en given bruger benytter gem funktionen.

Controlleren ”CaseInvestigationController”, har at gøre med informationen fra FXML-dokumentet ”caseInvestigation”. Controlleren er designet ligesom ”CaseOpeningController”, da de begge er designet til at håndtere information fra FXML-dokumenter, som er designet efter dokumenterne fra VUM. (Anon., 2019)

Designet af ”CreateCase”, var planlagt med fremtidig udvikling i tankerne, for at lave og implementere de dokumenter fra VUM, som mangler.

### FindCase – Sagsrelateret søgning

Subsystemet ”FindCase”, er designet til at håndtere et brugerinput, i form af en søgning, og præsentere information relateret til søgeordet. Subsystemet består af klassen ”SearchResult” og controlleren ”FindCaseController”, samt FXML-dokumentet ”findCase”.

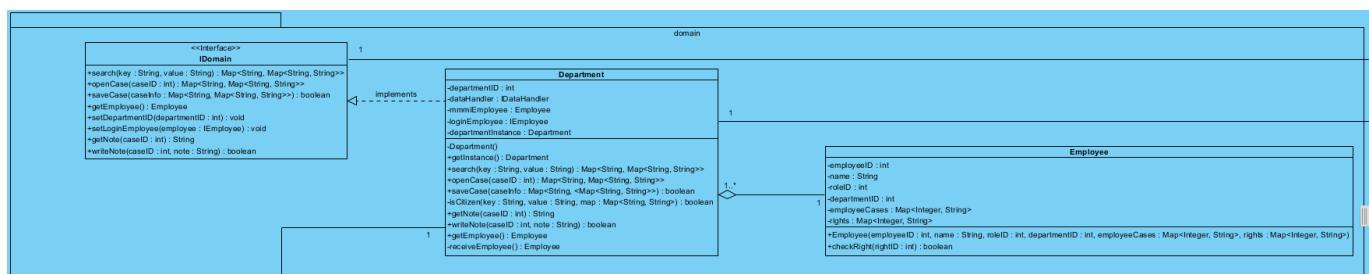
Klassen ”SearchResult”, er en placeholder som fyldes med den data, der hentes på baggrund af en given brugers søgeord. Controlleren ”FindCaseController”, henter informationen som ”SearchResult”, består af og præsentere dem gennem FXML-dokumentet ”findCase”.

## Home – Systemets startside

Subsystemet ”Home”, er designet til at håndtere systemets startside. Subsystemet består af klassen ”UIEmployee”, som er en placeholder klasse for medarbejderen der er logget ind i systemet og controlleren ”HomeController”, samt FXML-dokumentet ”home”.

Klassen ”UIEmployee”, er en placeholder klasse, som fyldes med information relateret til medarbejderen der er logget ind i systemet. Controllerklassen ”HomeController”, bruger informationen fra ”UIEmployee”, til at finde de relevante informationer som vedrør medarbejderen og præsentere dem gennem ”home”-dokumentet.

## Domænelaget



Figur 3.17: Domænelaget er blevet designet med to klasser, med attributter og metoder, og et interface.

## IDomain – Provided interface

IDomain interfacet bruges til at vise præsentationslaget hvilke metoder der er tilgængelige i domænelaget, hvad de returnerer samt hvilke attributter der er nødvendige for at benytte metoderne. Interfacets metoder bliver implementeret i ”Department” klassen.

## Department – Facadeklasse

Klassen ”Department” har en private constructor ”Department()”, attributterne ”departmentID”, ”dataHandler”, ”mmmiEmployee”, ”loginEmployee” og ”departmentInstance”, samt metoderne ”getInstance()”, ”search(...)”, ”openCase(...)”, ”saveCase(...)”, ”isCitizen(...)”, ”getNote(...)”, ”writeNote(...)”, ”getEmployee()” og ”receiveEmployee()”.

For at imødekomme projektets underspørgsmål ”Hvordan sikres det at aktøren kun har adgang til det aktøren har brug for?” og dataafgrænsningen fra sagsudredning i semestercasen, er der blevet lagt stor vægt på attributten ”departmentID”.

Dette ID tildeles via en setter metode, ”setDepartmentID(...)”, når en medarbejder logger ind i systemet. Singleton designpattern blev valgt, da der er nødvendigt at være sikre på at det er det samme ”Department”-objekt der arbejdes med gennem hele systemet.

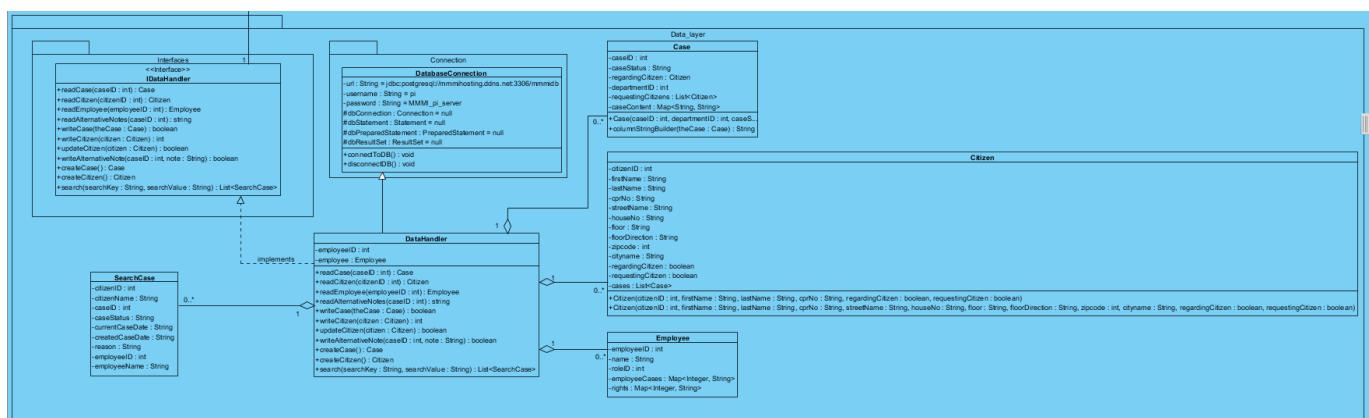
## Employee – Dataklasse, repræsenterer den medarbejder der er logget ind

Klassen ”Employee” er designet til at håndtere tjek af rettigheder i forhold til forskellige medarbejderroller.

Når en bruger logger ind i systemet, sendes der en employeeID til ”Department” som kalder persistenslaget for at få data om medarbejderen for at oprette en instans af Employee.

Dette gøres for at det kan være muligt at der er styr på hvem der er logget ind, samt have mulighed for at tjekke rettighederne som medarbejderen har gennem metoden ”checkRight(... )”.

## Persistenslaget



Figur 3.18: Persistenslaget er blevet designet med seks klasser og et interface.

### IDataHandler – Provided interface

Interfacet ”`IDataHandler`”, bruges til at tillade og etablere kontakten mellem domænelaget og persistenslaget. Metoderne som interfacet består af implementeres i ”`DataHandler`” klassen og kaldes i domænelaget, når databasen skal kontaktes.

### DataHandler – Facadeklasse

Klassen ”`DataHandler`”, er blevet designet til at håndtere dataforespørgsler sendt fra domænelaget og behandle dem som ønsket i systemets database. Klassen implementerer ”`IDataHandler`” interfacet og metoderne der følger med, som specificeres så data bliver hentet og/eller gemt, når de bliver kaldt fra domænet.

Når der skal hentes data fra databasen, er metoderne ”`readCase(...)`”, ”`readCitizen(...)`”, ”`readEmployee(...)`”, ”`readAlternativeNotes(...)`” og ”`search(...)`”, blevet designet for at kunne tilgå specifikke koloner, så den relevante information kan blive hentet.

Designet er lavet så at, når der skal gemmes data i databasen, benyttes metoderne ”`writeCase(...)`”, ”`writeCitizen(...)`”, ”`updateCitizen(...)`” og ”`writeAlternativeNote(...)`”. Metodekaldet der skal bruges, afhænger af hvilken information der ønskes gemt.

Klassens associationer består primært af aggregeringer, med multipliciteten one-to-zero-or-many. ”`DataHandler`” er forbundet med aggregeringer til klasserne ”`SearchCase`”, ”`Case`”, ”`Citizen`” og ”`Employee`”.

### SearchCase – Placeholder til data fra ”`search(...)`”

Klassen "SearchCase" indeholder udelukkende attributterne "citizenID", "citizenName", "caseID", "caseStatus", "currentCaseDate", "createdDateCase", "reason", "employeeID", "employeeName". Klassen bruges når der søges i databasen med metoden "search(...)", til at indeholde de data der er relevant at vise brugeren, i forhold til at kunne vælge den rigtige sag at åbne.

### **Employee – Dataklasse, repræsenterer en medarbejder og alle dennes tilknyttede sager i afdeling**

Klassen "Employee" indeholder udelukkende attributterne "employeeID", "name", "roleID", "employeeCases" og "rights".

### **Case – Dataklasse, repræsenterer en sag**

Klassen "Case" indeholder attributterne "caseID", "caseStatus", "regardingCitizen", "departmentID" som alle skal bruges for at oprette en ny sag i tabellen "case" i databasen. Yderligere er der en attribut "requestingCitizen", som skal bruges for at forbinde en sag med en eventuel henvendende borger, samt en attribut "caseContent", som indeholder alle oplysninger som en sagsbehandler kan indtaste. Der er også en constructor som sætter alle attributterne, samt en metode der bruges til at konstruere den query, som sørger for at "caseContent", bliver gemt i databasen.

### **Citizen – Dataklasse, repræsenterer en borger og alle dennes oprettede sager i afdeling**

Klassen "Citizen" indeholder attributterne "citizenID", "firstName", "lastName", "cprNo", "streetName", "houseNo", "floor", "floorDirection", "regardingCitizen", "requestingCitizen", "Cases". Den har 2 constructore, en som har de primære attributter der skal bruges for at oprette en Citizen i database, og en anden som har alle attributter.

### **DataConnection – Håndtering af forbindelse til database**

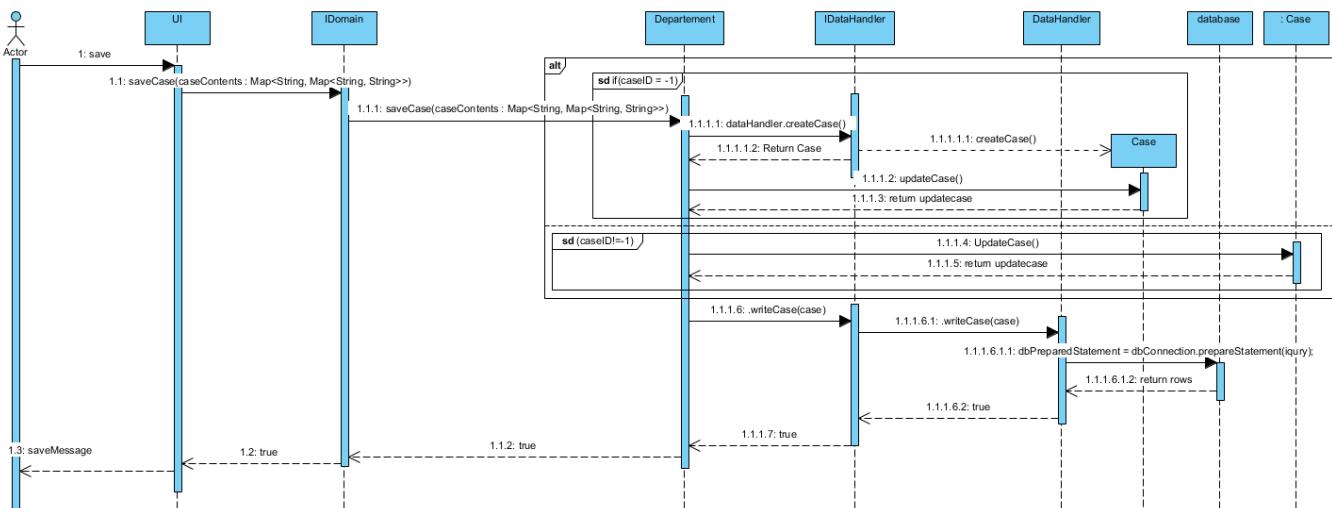
Klassen "DataConnection" blev implementeret for at vi kunne have de nødvendige informationer til at forbinde med databasen placeret et sted som kunne genbruges i andre moduler.

### **Den dynamisk side af design**

**saveCase** Når en aktør har udfyldt vilkårlige felter i sagsdokumenterne, kan denne data gemmes i databasen.

Når aktøren benytter "1.save", gemmes alt data der er indtastet i et contentsMap af typen "Map<String, Map<String, String>>". Der benyttes et Map af Maps for at minimere koblingen.

Når contentsMap gemmes, benyttes metoden "saveCase(...)" der sender den data igennem interfacet "IDomain" ned til "Department" facaden. Denne bliver pakket ud via metoden saveCase(...), i klassen Department, og tjekker på "caseID" om det er lig med -1. Hvis dette er tilfældet, sender den

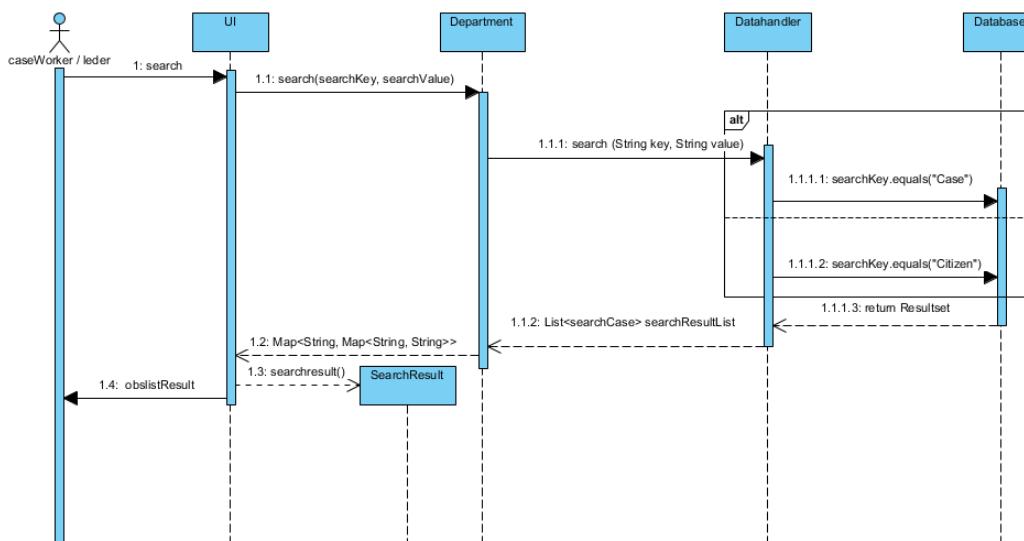


Figur 3.19: Sekvensdiagram for savecase

Map ned videre igennem IDataHandler interfacet og ned til DataHandler. SaveCase(...) metoden i Datahandleren tjekker om der er en ny Case på caseID, hvis denne er -1, benyttes createCase() metoden der opretter et ”Case” objekt.

Hvis en Case allerede findes i Databasen, bliver updateCase() metoden kaldt, som opdaterer ”Case” objektet og returner det opdaterede Case objekt.

Metoden som skriver i tilfældet af et eksisterende Case objekt eller et nyt Case objekt (1.1.1.6), benytter writeCase(...) metoden(1.1.1.6.1), som modtager et Case objekt og herefter skriver det ind i databasen ved at benytte en Prepared Statement. Hvis alle trin udføres succesfuldt, kvitteres der ”true” tilbage til aktøren.



Figur 3.20: Sekvensdiagram for search. Se interne bilag figur F.31

## Search

1. Personalet benytter søgefunktionen i systemet.

Denne bruges for at finde en eller flere sager på en given borger.

1.1. En searchKey er blevet valgt i præsentationslaget og et searchValue er blevet indtastet. Disse informationer sendes via Department metoden ”search(...)” , til domænelaget. Den valgte ”searchKey” bruges til at vælge om der søges efter en specifik sag, eller sager der er relateret til en person. ”searchValue” bruges til at identificere de relevante informationer.

1.1.1. Den valgte ”searchKey” bliver sendt videre som en ”String key” og searchValue bliver sendt videre som ”String value” i datahandler metoden search(...).

”StringKey” bestemmer hvilke SQL statement der skal bruges i persistenselaget. ”String value” er de værdier der skal matches i databasen.

1.1.1.1. Her bliver der undersøgt om den ”String key” der er blevet modtaget, har værdien ”Case”

1.1.1.2. Her bliver der undersøgt om den ”String key” der er blevet modtaget, har værdien ”Citizen”.

Afhængigt af værdien, ”Case” eller ”Citizen”, vil en SQL kode blive kørt og hente informationer relateret til ”String value” fra 1.1.1.

1.1.1.3. Databasen returner et ”ResultSet” med de værdier den har fundet.

Der bliver kun returneret data fra databasen, hvis ”departmentid” er i overensstemmelse med det der står i databasen.

1.1.2. Data bliver pakket ind i ”SearchCase” objekter og bliver gemt i en ”List<SearchCase>searchResultList” liste.

Denne liste sendes til domænelaget for at datapakken kan bearbejdes derfra.

1.2. De modtagede ”SearchCase” objekter bliver læst og skrevet ind i ”searchResultMap”. Disse Maps bliver skrevet ind i HashMap ”searchResultList”.

Der bliver sendt et Map af Maps fordi ”SearchCase” objekterne ikke kan blive sendt videre til præsentationslaget. Vi minimerer kobling mellem lagende ved at skifte en liste af ”SearchCase” objekter og udskifter dem med Map af Maps.

1.3. Det modtagede ”Map<String, Map<String, String>>” pakkes ud og opretter ”SearchResult” objekter af de underste Maps.

1.4. Alle ”SearchResult” objekter bliver tilføjet til en ”ObservableList<SearchResult>”.

Der oprettes ”SearchResult” objekter for at kunne præsentere de modtagede informationer. Disse informationer tilføjes derefter til en ”ObservableList<SearchResult>” liste for at ”TableView” kan præsentere de fundne informationer til brugeren.

### 3.4.3 Implementering

#### Nodefinder

##### Problem

Hvordan kan systemme selv gemme alt data i et map.

##### Eksempel

Håndtering af input fra aktøren i UI. **Løsning**

---

```

public ArrayList<Node> getAllNodesFromRoot(Parent root) {
    ArrayList<Node> nodes = new ArrayList<>();
    addAllDescendentsFromNode(root, nodes);
    return nodes;
}

private void addAllDescendentsFromNode(Parent parent, ArrayList<Node> nodes){
    for (Node node : parent.getChildrenUnmodifiable()) {
        nodes.add(node);
        if (node instanceof Parent) {
            if (node instanceof TextInputControl) {
                if (!((TextInputControl) node).getText().isEmpty()) {
                    String key = node.getId().substring(5);
                    nodeMap.put(key, ((TextInputControl) node).getText());
                }
            }
            if (node instanceof RadioButton) {
                if (((Toggle) node).isSelected()) {
                    String key = node.getId().substring(5, ((RadioButton) node).getId().length() - 1);
                    nodeMap.put(key, ((Labeled) node).getText());
                }
            }
            addAllDescendentsFromNode((Parent) node, nodes);
        }
    }
}

```

---

Figur 3.21: Data input håndtering

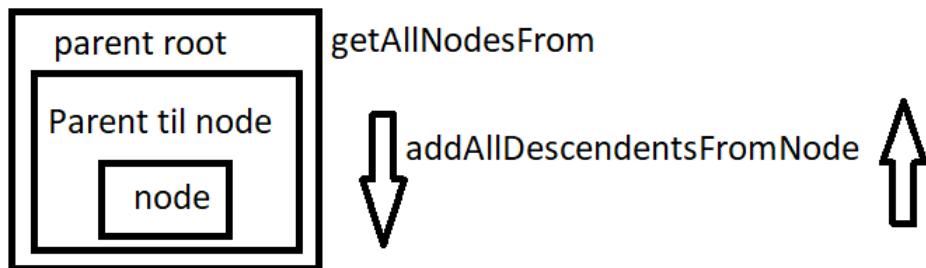
”NodeFinder” der har til formål at håndtering input Aktøren. Klassen har ansvaret igennem metoderne ”addAllDescendentsFromNode” og ”getAllNodesFromRoot” at samle alle nodes og gennem værdierne i et ”map” hvor nøglen er node id og værdien er den værdig der ligger i den node. Den har også formålet at sortere node der er i den forældrenode så vi kun kan få de nods der har en betydning den data vi skal has ind.

”getAllNodesFromRoot” tager alle ”nods” der ligger på den root node og ligger dem i en ArrayList. ”addAllDescendentsFromNode” tager alle node der er i arraylisten og køre dem igennem. Alle de nodes der er en parent node til en node køres igennem igen for at finde alle de nodes der enten er en

”Textinputcontrole” eller en ”radiobutton” og samler dem i et map. Dette map bliver samlet med 2 andre maps ” ”cRegardingMap” og ”cRequestingMap” disse 3 maps udgøre til sammen hele sagen.

### Evaluering

”NodeFinder” kan implementeres i alle fxml og samle den specifik data, hvis node id passer med det



Figur 3.22: Viser oprationen for hvordan den henter alle nodes

kolonnen navn i databasen. Dette gør at det bliver nemmere for fremtidige programøre at implementere fxml dokumenter i systemet og samle den data der bliver sat ind.

### Search case

#### Problem

Hvilke informationer har domænelaget brug for og hvor skal det hentes fra, når der bliver foretaget en søgning?

#### Eksempel

Domænelaget sender to værdier til persistens laget. Værdierne skal behandles så ”search-Key” bestemmer hvordan der skal søges og værdien ”searchValue” hvad der skal søges efter.

#### Løsning

Public List search(String searchKey, String searchValue)

Der er blevet valgt at ”searchKey” argumentet, håndteres som en af to prædefinerede sætninger: ”case” eller ”citizen”.

Argumentet ”searchValue”, består af de attributter som bruges til at finde den ønskede data, samt autorisere personens ret til at se dem.

Ved brug af ”case” sætningen, aktiveres en SQL-Query, som leder efter en enkelt sag. Denne Query benytter attributter fra ”searchValue” til at udfylde metodens wildcards:

”caseid” (Erstatter de første tre wildcards) og ”departmentid” (Erstatter det fjerde wildcard).

Kontrollen af ”departmentid”, bruges til at autorisere et bosteds ansvarsområde i databasen. Hvis brugeren som søger på sagen, arbejder i den afdeling der behandler sagen, vil de kunne se data-outputtet der sendes med ”SearchCase” objektet.

Der bruges et ”caseid” til at identificere den specifikke sag der søges. Dette gøres ved hjælp af en Sub-Query, der sorteres efter ”datestamp” blandt de sager som har samme ”caseid”. Denne Sub-Query

bruges til at hente de relevante data, fra den seneste udgave af en sag.

---

```
if(searchKey.equals("Case")) {  
    try {  
        connectToDB();  
        if(dbConnection != null) {  
            String[] search = searchValue.split("%");  
            selectQuery = "SELECT c.caseid, c.casestatus, to_char(c.createddate, 'DD/MM/YYYY'),  
            ci.citizenid, CONCAT(ci.firstname, ' ', ci.lastname) AS citizenName,  
            to_char(cc.timestamp, 'DD/MM/YYYY') AS timestamp, cc.regardinginquiry,  
            CONCAT(e.firstname, ' ', e.lastname) AS employeeName, e.employeeid  
            ;  
  
            fromQuery = "FROM \"case\"_as_c, _citizen AS_ci, (SELECT casecaseid, timestamp,  
            regardinginquiry FROM case_contents WHERE casecaseid = ? ORDER BY timestamp DESC LIMIT 1)  
            AS_cc, (SELECT casecaseid, employeeemployeeid FROM case_employee  
            WHERE casecaseid = ?) AS_ce, _employee AS_e";  
  
            whereQuery = "WHERE ce.casecaseid = c.caseid AND ce.employeeid = ce.employeeemployeeid  
            AND cc.casecaseid = c.caseid AND ci.citizenid =  
            c.citizenregardingcitizenid AND c.caseid = ? AND  
            c.departmentdepartmentid = ? ORDER BY c.caseid DESC";  
  
            query = selectQuery + fromQuery + whereQuery;  
            searchCaseStmt = dbConnection.prepareStatement(query);  
            searchCaseStmt.setString(1, search[0]);  
            searchCaseStmt.setString(2, search[0]);  
            searchCaseStmt.setString(3, search[0]);  
            searchCaseStmt.setInt(4, Integer.valueOf(search[1]));  
            dbResultSet = searchCaseStmt.executeQuery();  
            while(dbResultSet.next()) {  
                sc.add(new SearchCase(dbResultSet.getInt(4), dbResultSet.getString(5), dbResultSet.  
                getString(1), dbResultSet.getString(2), dbResultSet.getString(6),  
                dbResultSet.getString(3), dbResultSet.getString(7),  
                dbResultSet.getInt(9), dbResultSet.getString(8)));  
            }  
        }  
    }  
}
```

---

Ved brug af ”citizen” sætningen, aktiveres en SQL-Query, som søger data på baggrund af en borgers CPR-nummer navn, adresse, gadenummer og/eller postkode.

---

```
else if(searchKey.equals("Citizen")) {
    try {
        connectToDB();
        if(dbConnection != null) {
            String[] search = searchValue.split("%");
            List<String> searchVal = new ArrayList();
            String searchQuery = "";
            int columns = 0;
            if(!search[0].equals("")) {
                searchQuery += "\"CPR-nr\" LIKE ?";
                columns++;
                searchVal.add(search[0]);
            }
            if(!search[1].equals("")) {
                if(!searchQuery.equals("")) {
                    searchQuery += " AND ";
                }
                searchQuery += "CONCAT(ci.firstname, ' ', ci.lastname) LIKE ?";
                columns++;
                searchVal.add(search[1]);
            }
            if(!search[2].equals("") || !search[3].equals("")) {
                if(!searchQuery.equals("")) {
                    searchQuery += " AND ";
                }
                searchQuery += "(CONCAT(ci.streetname, ' ', cci.houseno) LIKE ?
                AND CONCAT(ci.zipcodezipcode, ' ', cz.cityname) LIKE ?)";
                columns++;
                searchVal.add(search[2]);
                columns++;
                searchVal.add(search[3]);
            }
        }
    }
}
```

---

Denne Query benytter ”departmentid” til at autorisere søgningen, på samme måde som beskrevet for første ”searchKey”-eksempel.

Attributterne der anvendes fra ”searchValue”, placeres i et String array og derefter samles i en liste. Elementerne fra denne liste, kaldes eneklvis mellem to wildcards (%). Ved at tilføje disse wildcards på hver side, underrettes den pågældende SQL-Query om at der søges på dele af dataværdier. Når en String fra arrayet skal matches i databasen, vil alle elementer med samme værdi blive hentet, så længe at værdien er case sensitive. (Se figur 3.23)

Når søgefunktionen implementeres på denne måde, er det muligt at returnere en bred søgning. En List vil blive fyldt med alle elementer som er relevante i forhold til et givent søgeord.

Hvis der bliver søgt efter navnet ”Peter”, vil alle der hedder ”Peter” blive hentet. På grund af brugen af wildcards, vil navne hvor ”Peter” indgår i også blive hentet, f.eks:

”Anders Petersen”, ”Jens Peter”, ”Peter Halfdan”

```
searchCaseStmt = dbConnection.prepareStatement(query);
int index = 1;
for(int i = 1; i <= 3; i++) {
    for (int j = 0; j < columns; j++) {
        searchCaseStmt.setString(index, "%" + searchVal.get(j) + "%");
        index++;
    }
}
searchCaseStmt.setInt(index, Integer.valueOf(search[4]));
dbResultSet = searchCaseStmt.executeQuery();
while(dbResultSet.next()) {
    sc.add(new SearchCase(dbResultSet.getInt(4), dbResultSet.getString(5),
    dbResultSet.getString(1), dbResultSet.getString(2), dbResultSet.getString(6),
    dbResultSet.getString(3), dbResultSet.getString(7), dbResultSet.getInt(9),
    dbResultSet.getString(8)));
}
}
```

Figur 3.23: % search

### 3.4.4 Unit test

#### saveCaseTest()

Der er udført en test på saveCase() metoden i klassen Department. Testens formål er at kunne modtage data fra Præsentationslag og lægge det ned til Persistenslaget som herefter bliver gemt i databasen.

Der er blevet oprettet en metode saveCaseTest() (Se figur 3.24), hvor der oprettes en caseInfo Map som skal kunne modtage en RegardingCitizen af typen Map og en RequestingCitizen af typen Map. Der bliver udover det også oprettet en caseContents af typen Map hvor alt data vedrørende sagsoplysninger lægges i. På figuren ses det at der er sat key og value værdier ind, og da det også skal være muligt at indsætte tomme pladser i f.eks. case\_contents tabellen i databasen, er caseContents ikke fyldt helt ud med alle værdier. Som det næste sættes en RegardingCitizen ind med oplysninger og en RequestingCitizen med oplysninger. Dette for at teste om borgen registreres i databasens Citizen tabel. Til sidst lægges alle 3 Maps ind i caseInfo Map hvor der efterfølgende bliver brugt en assertEquals() metode for at teste om det forventede resultat er sandt.

Efter at have kørt testen igennem returnerede den true hvilket betød at testen var succesfuldt.

@Test

```
public void saveCaseTest() {
    Map<String, Map<String, String>> caseInfo = new HashMap<>();
    department.setDepartmentID(1);
    Map<String, String> conentsMap = new HashMap<>();
    conentsMap.put("regardinginquiry", "Henvendelse_drejer_sig_om_en_Test");
    conentsMap.put("treatment", "paragrafTest2");
    conentsMap.put("carriage", "paragrafTest1");
    conentsMap.put("caseID", "-1");

    Map<String, String> citizenInfoRegarding1 = new HashMap<>();
    citizenInfoRegarding1.put("firstName", "Hans");
    citizenInfoRegarding1.put("lastName", "Frederiksen");
    citizenInfoRegarding1.put("streetName", "Snervej");
    citizenInfoRegarding1.put("cityName", "Taarup");
    citizenInfoRegarding1.put("zipCode", "5560");
    citizenInfoRegarding1.put("houseNo", "75");
    citizenInfoRegarding1.put("regardingCitizen", "true");
    citizenInfoRegarding1.put("requestingCitizen", "false");
    citizenInfoRegarding1.put("cpr", "887799-5544");
    citizenInfoRegarding1.put("floor", "");
    citizenInfoRegarding1.put("floorDirection", "");

    Map<String, String> citizenInfoRequesting1 = new HashMap<>();
    citizenInfoRequesting1.put("firstName", "Mads");
    citizenInfoRequesting1.put("lastName", "Simpson");
    citizenInfoRequesting1.put("streetName", "tillevej");
    citizenInfoRequesting1.put("cityName", "Odense");
    citizenInfoRequesting1.put("zipCode", "5000");
    citizenInfoRequesting1.put("houseNo", "22");
    citizenInfoRequesting1.put("regardingCitizen", "false");
    citizenInfoRequesting1.put("requestingCitizen", "true");
    citizenInfoRequesting1.put("cpr", "120596-7745");
    citizenInfoRequesting1.put("floor", "etage_2");
    citizenInfoRequesting1.put("floorDirection", "venstre");

    caseInfo.put("caseContents", conentsMap);
    caseInfo.put("cRegarding", citizenInfoRegarding1);
    caseInfo.put("cRequesting", citizenInfoRequesting1);

    assertEquals(true, department.saveCase(caseInfo));
}
```

Figur 3.24: Unit test kode for Savecase

### searchTestCase()

Der er udført en test på search() metoden i Department, som skal vise at man får et søges resultat tilbage fra datalaget med det data der er blevet søgt på. I test metoden searchTestCase() er overvejelsen

af at man kan søge på en sag med specifikt sagsnummer og få resultatet af søgning fra datalaget og hele vejen igennem op til brugergrænsefladen.

I figur F.32 fra interne bilag udføres testen searchTestCase() hvor der deklarerer en variable caseNumber med værdien ”123” der skal være formålet med at søge en sag frem på et specifikt sagsnummer. Metoden searchHandler.search(”Case”, caseNumber + ”%1”), er den metode som sender nøglen der ”Case” og værdien ”123%1”. Adskillelsen med % fortæller at den første værdi er sagsnummeret og den anden er departmentID. Der oprettes et expected Map med de værdier der forventes at få. I assertEquals(searchResultMap, Is.is(expectedMap)) metoden tjekker man om searchResultMap er det samme som det forventede.

Ved test af koden blev det forventede map printet ud og var det samme som funktionens map, hvilket gjorde at testen var succesfuldt.

Testen viser at der kan søges på en specifik sag og samtidig se hvilken borger der er tilknyttet sagen. Dette gør det lettere for at kunne finde ud af hvad sag en borger er tilknyttet.

### searchTestCase()

I det forrige afsnit søgte metoden searchTestCase() på en specifik sag. Der har som en beslutning af gruppen også fortaget at implementere search() med at kunne søge på en borger. Testen searchTestCitizen() har til formål at kunne søge på en specifik borger og baseret på borgeren vise de sager den pågældende borger har.

I figur F.33 fra interne bilag anvendes searchTestCitizen() metoden hvor der bliver deklareret en variable citizen af typen String. Den får værdierne ”123123-1231%Poul Johanson%kimvej%”, der bliver separeret med ”%”. Derefter oprettes et Map som kaldes expectedMap der er data som er forventet at få tilbage. Derefter laves assertEquals(searchResultMap, Is.is(expectedMap)) der er tester om searchResultMap er det samme som expectedMap.

Testen viser at det er den rigtige borger der kommer frem når man søger, hvilket gør at testen er succesfuldt

## 3.4.5 System test

### Generelt

- Systemet kan ikke startes ved at trykke på projektmappen og derefter run. Når dette gøres, starter den login systemet om, og man har mulighed for at indtaste sine logininformationer, der er i dette tilfælde blevet brugt ”admin” til både brugernavn og adgangskode, når dette er gjort og man trykker på loginknappen, bliver der i konsollen smidt en ”NullPointerException” men Location is required, og det er ikke muligt at komme videre i systemet.

- Kan ikke ud kommentere Main-metode i mmmi, hvis dette gøres, kan modulet ikke startes op. Det var forventet at samarbejdet mellem loginsystemet og mmmi modulet var sat op så mmmi ikke havde brug for en Main-metode.
- I menuen i venstre side på UI'et er det kun Hjem, Opret sag og Find sag som kan trykkes på, resten af knapperne har ingen funktion.

## Login System

- Når man har skrevet brugernavn eller adgangskode forkert 4 gange, der er i input til brugernavn og adgangskode blevet brugt henholdsvis ”admin” til brugernavn og ”ain” til adgangskode, skal man kontakte admin, men der er muligt at bypass ved at lukke programmet og starte det igen.

## Opstarts side

- Når der er skrevet tekst i eventuelle noter og går videre til sig næste sag, rydder den ikke feltet eller skriver det som er på en anden sag. Der er dog skrevet og gemt i databasen.
- Hvis man skriver i kommentarfeltet og reguardingquery er ”Null” i databasen, der er blevet testet på sagen med sagsnummer ”123” som har noget stående i case\_contents tabellen. bliver der skrevet en SQL-fejl, men ”violates not-null constraint”. Det forventet output var at det lavet en ny række i tabellen med et nyt timestamp og det eksisterne indhold med ændringer i den kolonne som hedder alternativenotes.
- Feltet hvor navnet på brugerne som er logget ind, er ikke stort nok, da der er en begrænsning og ikke kan vise ”Aleksander Henriksen”, men skriver ”Aleksander Henrik...”

## Sagsåbnings dokument

- Henvendelse:
  - Hvis man indtaster et postnummer som ikke eksistere i databasen, der er i testen indtastet ”1”, bliver der smidt en fejl med violates foreign key constraint, der mangler en håndtering.
- Rettigheder og pligter:
  - Er borgerne informeret om ret til bisidder og partsrepræsentant, der mangler mulighed for at svar ja eller nej eller så mangler der at gøres så det er muligt at fjerne markeringen af dette ved fejl udfyldning.
- Vægemål og repæsentant:

- Teksfelt under f.eks. vægemål og repreæsentant hvis der vælges samværgemål aktiveres teksfeltet, men hvis man vælger f.eks. vægemål så er indholdet stadig i feltet hvis der trykkes på gem.

- Ydelser og paragraffer:

- Under de forskellige punkter er det ikke muligt at fjerne sin markering helt i tilfælde af fejltryk eller borgerne ikke skal have denne ydelse, det er dog lavet på nogle få stykker.
- Hvis man marker flere ydelser under de forskellige sektioner forekommer problemet at der ikke gemmes alle værdiger fra de afkrydsede radiobuttons, da der er for mange der er blevet afkrydset.

- Når man gemmer:

- Der skal trykkes på 2 knapper for at gemme et dokument, og når man trykker på den sidste gem knap, bliver dataene som man vil gemme lagt som Null eller tomt i databasen.
- Når man indtaster information omkring en borger, og man trykker på gem 2 gange, bliver der anden gang man trykker gem smidt en ”NullPointerException”, og der bliver kun oprettet en sag selvom det kunne være 2 forskellige sager på samme borger, men der bliver oprettet 2 børger i Citizen tabellen, med de samme værdier.

## Udredning dokument

- Det er ikke muligt at lavet et tab og komme videre til næste tekst felt.

## Find sag

- Når der søges på sagsnummer ”123” som er et case id i databasen, bliver der skrevet en fejl på at integer = character varing, så der er ikke blevet håndteret at caseid i database er blevet serial i stedet for varchar, derved ingen output i UI’et.
- Når der søges efter ”peter” eller ”peter test” under feltet Borgers navn og trykker på knappen Søg, bliver der ikke fundet nogen sager og vist i tableviewet. Det forventet output var to sager på Peter men sagsnumrene 37 og 38.
- Det samme sker når man søger på ”snurvej” under feltet Borgers adresse og trykker på kanppen Søg. Det forventet output var 5 sager med sagsnumrene 46-50.
- Skrives der ”800” eller ”800 Høje Taastrup” i feltet postnummer og by, vises der ingenting i tableviewet, men et komma adskilles af postnummer og by, gives der heller ikke nogen søgning. Det forventet output var en sag på Jesper Jensen men sagsnummer 28 og to sager på Peter Test med sagsnumrene 37 og 38.

- Ved indtastning i CPR-nummer, med CPR-nummeret "120366-1235" eller "1203661235" eller "120366 1235", gives der heller noget resultat, der var ved denne søgning forventet at få vist en sag med sagsnummer 1, en oprettelse dato: 21-05-2019, borgers navn: Jens Hansen og ingen ting i resten af felterne, da der ikke er noget indtastet data dertil.

## 4 Diskussion

Det endelige produkt der er blevet udviklet, er udformet som det var planlagt i forhold til kravspecifikationerne med enkelte undtagelser.

Funktionerne til at afslutte og tildele en sag i systemet, blev aldrig implementeret. Dette betyder, at modulet ikke er færdigt og brugbart. Hvis det blev brugt, ville det ikke være muligt at afslutte en sag eller ændre på hvem der arbejdede på en given sag. Disse funktioner blev nedprioriteret, fordi projektet udelukkende skulle foregå i Inceptions- og Elaborationsfasen.

Nedprioriteringen af disse funktioner blev først besluttet, da gruppen var nået til udvikling af designmodeller til systemet og der findes derfor en del analysediagrammer som danner grundlaget for den videre udvikling af designmodeller og implementering.

Afdelingslederens fulde ansvarsområde, med funktionaliteten som hørte til dette, blev heller ikke realiseret. Området i den udleverede case som der blev arbejdet med, "Sagsudredning", inkluderede ikke krav om administreringen af et overordnet system. Da dette var tilfældet, blev alle planer omhandlende intern systemadministrering, udskudt til fremtidige iterationer.

Funktionerne til at oprette en ny sag og finde en given sag, fik den højeste prioritet, da disse aspekter direkte kunne tolkes ud fra semestercasen. Det lykkedes at implementere begge funktioner samt overholde dataafgrænsningen fra "Sagsudredning". Dataafgrænsningen havde at gøre med at finde en given sag på baggrund af en afdelingsautentifisering. Den afdelingsautentifisering som blev implementeret, skete i systemets database ved brug af et afdelings ID. Dette betød, at når der blev lavet en søgning, skulle afdelingen søgningen kom fra, være registreret i elementet der blev søgt efter.

Det produkt der er udviklet i Elaborationsfasen er baseret på en 3-lagsarkitektur, som gør det nemmere at adskille funktionalitet og afspejler en adskillelse af brugergrænseflade, forretningslogik og datahåndtering. Denne arkitektur blev til dels valgt for at gøre det mere overskueligt at udvikle systemet fremadrettet, men også fordi gruppen ville være i stand til at skifte både præsentationslaget og persistenslaget ud uden større komplikationer.

Det var også en afgrænsning i forhold til de roller en bruger kunne få og der blev gjort klar til

at domænelaget kunne håndtere denne afgrænsning i den kode som er blevet implementeret. Den er dog ikke gjort helt færdig da implementeringen af præsentationslaget tog en del mere tid end først beregnet. Gruppen havde tænkt det skulle være styret af en rolle som en bruger får tildelt, hvor hver rolle så har en mængde rettigheder i forhold til systemet.

For at kunne gemme data blev der udviklet et ER-diagram som blev brugt til at oprette tabellerne i en postgresql database. Der bruges en relationel database, da der var krav om dataafgrænsning, som er besværligt at opnå tilfredsstillende med en nosql database. Valget af postgresql som den understøttende databaseserver, skyldtes at det var den som blev introduceret i faget Database.

Gruppen kunne have opnået bedre resultater, hvis der havde været et bedre kendskab til problemområdet. Hvis dette havde været tilfældet, ville der nemmere kunne opstilles krav og udarbejdes brugsmønstre. Havde der været bedre kendskab til arbejdsmetoderne, som der blev benyttet i starten af projektet, ville det have været nemmere at udpege de relevante områder at modellere i forhold til udarbejdelse af dokumentation.

Der var blevet lagt op til at projektet skulle styres med Scrum, men som helhed blev fravalgt. Scrum blev først gennemgået efter at projektarbejdet var startet, og var derfor besværligt at implementere, da det ville betyde at den allerede etablerede struktur skulle ændres. Tidsinvesteringen som var nødvendig for at implementere den komplette Scrum, blev vurderet til at være ressourcesspild, da der allerede eksisterede en velfungerende arbejdsstruktur.

## 5 Konklusion

For at sikre at programmet kan håndtere flere roller er det nødvendigt at vide, hvad en rolle er. Dette er blevet lavet i gruppens program, så der er tre forskellige roller en bruger kan have mens de benytter systemet. Rollerne er sekretær, sagsbehandler og afdelingsleder.

Sekretæren har mindst ansvar, mens afdelingslederen har mest. Et område som en sekretær ikke har adgang til, men som de andre har, er at behandle en sag og skrive information ind i sagen. Systemet håndterer roller ved at tildele dem, når en person logger på. Et login har en bruger med en specifik rolle, der så fortæller systemet hvad brugeren må og ikke må. De fleste felter i databasen er i et string format og nogle af disse omhandler personfølsomme data. Der benyttes prepared statements til at skrive alt data til databasen, hvilket sikrer at en bruger ikke kan skrive SQL direkte til databasen. Funktioner til tildeling af sager eller afslutningen af sager har gruppen ikke fået at tilføje i systemet og så mangler der en færdig implementering af rettighederne. Dette gør, at det lige nu er ligegyldigt hvad for en rolle brugeren har, men grundlaget er blevet lagt for at de bliver implementeret i fremtidige iterationer. Der er blevet implementeret begrænsninger så brugere ikke kan søge igennem andre afdelingers sager.

Dermed er der fundet frem til en løsning på casens problem samt gruppens problemstilling. Modulet har en række ”TODO”’s i koden, hvilket gør det muligt at se i hvilken retning gruppen havde tænkt sig at arbejde i fremtidige iterationer.

## 6 Perspektivering

Den nuværende fase som projektet befinder sig i er Elaborationsfasen (slutning af elaborationsfasen). Gruppen har arbejdet i 2. iteration med at få implementeret væsentlige brugsmønstre, som f.eks. find en sag eller finde en borger der er tilknyttet en sag, og oprettelse af en sag baseret på forskellige formular.

Det næste skridt i projektet er konstruktionsfasen der har til formål at udarbejde og implementere den manglende funktionalitet f.eks. behandle sag, tildeling af sag og dagbog, samtidig med at lave test, som gøre produktet klar til kunden. Dette kan gøres igennem 2 – 3 iterationer, hvor 1. iteration ville handle om at få optimeret kode i forhold til det der er i forvejen samt færdiggøre funktionaliteten. I 2. iteration af konstruktionsfasen ville der blive fokuseret på at få implementeret den resterende funktionalitet som f.eks. at afdelingslederen har mulighed for at tilde sager til sagsbehandlere. I 3. iteration lave en endelig version af systemet, med en brugergrænseflade der både er brugervenlig og funktionel og som er klar til test. Til sidst når projektet når transationsfasen er det til formål udgiver produktet til kunden, hvor så man får feedback og kan så justere og finpudse de resterende påpegninger til systemet.

Hvis projektet skulle bygges igen, med en realistisk tidsramme og med den viden som gruppen har tilegnet sig, så ville det gøres på følgende måde. Der ville med fordel kunne laves brugsmønstre og til det lave en overordnet kravspecifikation med væsentlige detaljerede brugsmønstre. Efterfølgende ville der blive udarbejdet et analyseklassediagram over alle de klasser og funktioner fundet i de detaljerede brugsmønstre og dermed udarbejde et designklassediagram til brug for implementering. Idet ville et component diagram også blive udarbejdet i forhold til at vise hvordan systemet er opbygget så der er bedre forståelse for forskellige komponenter. Til sidst begynder 1. iteration med at udarbejde funktionaliteten generelt og i 2. iteration udarbejde et databasedesign og database implementering og brugergrænseflade som en klar pakke til at blive sendt ind i en konstruktionsfase. I konstruktionsfasen ville produktet blive mere dybdegående udarbejdet og optimeret og lave en brugervenlig grænseflade igennem 2 – 3 iterationer og samtidig gøre den klar til levering for kunden til test.

## 7 Litteraturliste

Anon., 2019. Socialstyrelsen. [Online]

Available at: <https://socialstyrelsen.dk/tvaergaendeomrader/sagsbehandling/voksenudredningsmetoden>  
(Senest hentet eller vist den 13 Maj 2019)

# B Procesrapport

## 1 Læring og refleksion

Der har været stort fokus på at udarbejde forskellige modeller og diagrammer gennem projektet. De er blevet brugt effektivt til at supplere både koordinering og planlægning af den kildekode som skulle udvikles. Der var en løbende justering af disse modeller, der blev lavet for at undgå problemer og fejl, som først blev opdaget efter first-cut udgaven var lavet. Hvis kildekoden var blevet produceret uden modeller og diagrammer, er det tydeligt at utallige fejl ville være opstået og hvis disse ikke blev løst ordentligt, kunne de have resulteret i et ubrugeligt system.

Brugen af brugsmønster modeller, analyse- og designdiagrammer mm. kan klart anbefales til fremtidige projekter, men med måde.

Der blev erfaret at tiden som blev brugt til at lave dynamiske diagrammer, ikke blev opvejet af værdien der kunne trækkes ud af dem. De blev brugt til at skabe fælles konsensus, om hvilken retning projektet skulle udvikles i, men var ikke præcise nok til at understøtte produktprogrammeringen.

Det er muligt at denne ulempe er et resultat af manglende erfaring med brugen af dynamiske diagrammer. At de blev introduceret sent i projektforløbet og var helt ny viden for gruppemedlemmerne, har sandsynligvis haft en stor indvirkning på hvor effektivt de blev brugt i projektet.

De statiske diagrammer er der kun blevet gjort positive erfaringer med. De blev brugt til at danne et klart grundlag for kildekoden, som gjorde at arbejdstempoet var højt og er løbende blevet brugt til at sikre at klasser blev skrevet korrekt. Den succes som der er blevet gjort med de statiske diagrammer, har gjort det tydeligt, hvor nyttig og essentielt planlægning er for et succesfuldt projekt.

## 2 Projektstyring

Projektplanlægning bruges til at udarbejde estimerater for hvor lang tid en given opgave vil tage at fuldende og derved også give en estimeret tidsplan for et projekt.

Projektstyring bruges til at bestemme hvordan en opgave kan udføres mest effektivt, hvor mange ressourcer der skal bruges på den og for at sikre de estimerater der er blevet stillet i planlægningen ikke overskrides med for stor en margin, hvis den overhovedet overskrides.

Til projektets planlægning blev der udarbejdet en række milepæle, samt der blev lagt en tidsplan for hvornår de forskellige milepæle skulle være nået. Milepælene blev oprettet i github så vi havde dem placeret et centralt sted som alle havde adgang til. Til styring af projektet er der blevet brugt

### Semester projekt gruppe 02



Figur B.1: Tidsplan se interne bilag 4 figur F.29

værktøjet GitKraken Glo, til at håndtere projektets kanban board, som bestod af issues oprettet i githubs issue-tracking system. For at sikre et godt overblik kunne issues være i forskellige stadier ”Issues”, ”To Do”, ”In Progress” og ”Need review”. På denne måde var det muligt at se hvor langt en given opgave var nået.

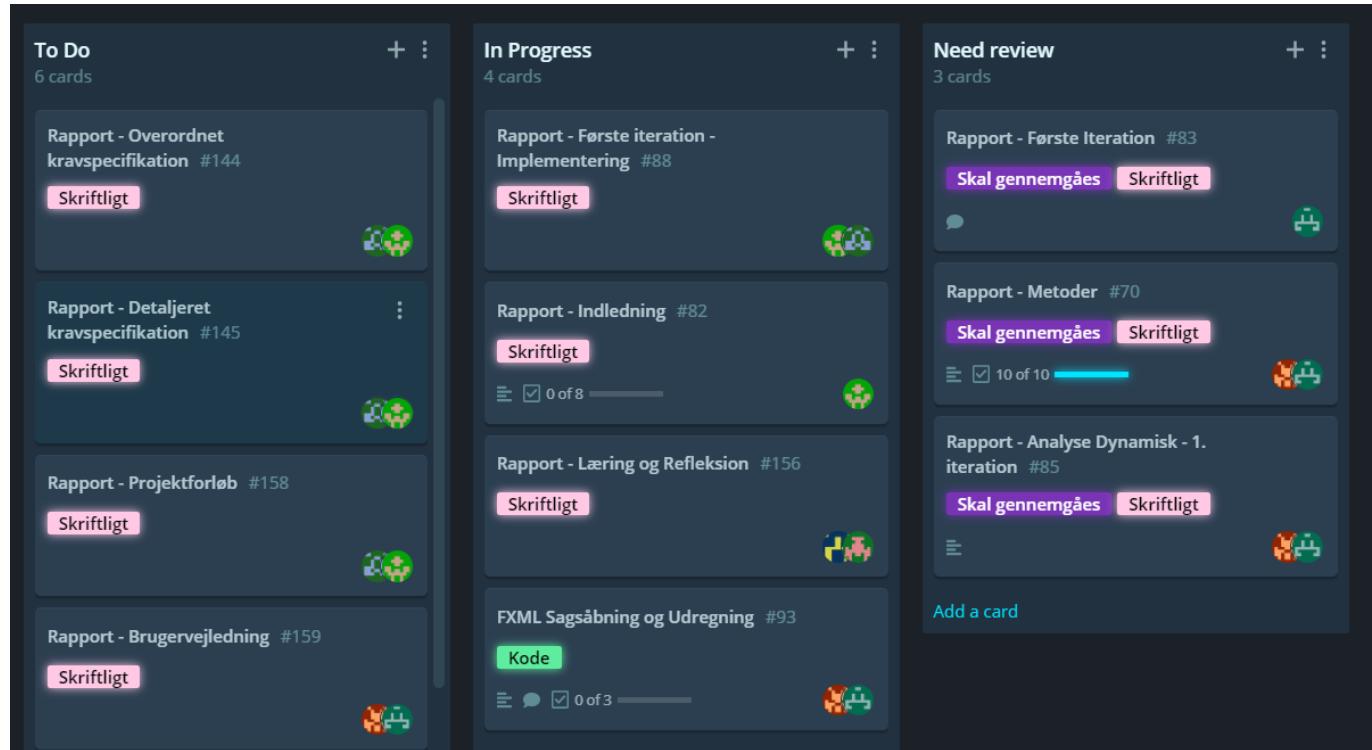
Hvis en opgave var i stadiet ”Issues”, betød det at den manglede at blive tildelt til en i gruppen.

Hvis en opgave var i stadiet ”To Do”, betød det at den var blevet tildelt en person, men ikke var startet.

Hvis en opgave var i stadiet ”In Progress”, betød det at den var ved at blive løst.

Hvis en opgave var i stadiet ”Need review”, betød det at den var løst og skulle gennemgås af gruppen.

Der blev holdt to ugentlige møder, tirsdag og fredag, for at fordele eksisterende opgaver og for at



Figur B.2: Kanban board se interne bilag 4 figur F.29

oprette nye, som opstod i forlængelse af udviklingen. Tirsdagsmødet inkluderede et vejledermøde, som blev brugt til at få feedback på det udførte arbejde og planerne gruppen havde lagt for fremtidigt arbejde. Vejledermødet blev også brugt til at forventningsafstemme, hvordan nyt materiale skulle bruges og vægtes, hvis der var tvivl om dette.

### Ansvarsfordeling

- Planlægger: gitkraken, issue: Aleksander H.
- Ordstyrer, dagsorden i forhold til møder: Aslak
- Referent: Steffen og Aleksander D.
- Rapporten: Per og Mathias.

## 3 Identifikation, analyse og bearbejdning af problemer

### 3.1 Misforståelser

Det største problem som der har været gennem projektarbejdet, har været misforståelser. Disse misforståelser er primært opstået på grund af forkert brug af fagterminer og dårlig formulering, og kan generelt beskrives som dårlig kommunikation.

Når der har været en diskussion mellem to parter og det ikke har været muligt at komme til enighed, afbrydes diskussionen og én af parterne får så ordet. Første part får så chancen for at uddybe deres holdninger, og anden part stiller uddybende spørgsmål til disse. Hvis parterne ikke er kommet til enighed, bliver rollerne byttet, anden part uddyber og første part stiller spørgsmål.

Ved at arbejde med én holdning ad gangen, er det muligt at få en klar forståelse for det perspektiv som der argumenteres ud fra og på denne måde identificeres den eventuelle misforståelse og et kompromis kan indgås hvis nødvendigt.

### 3.2 Faglighed

Den løbende introduktion af nye emner, har givet problemer gennem projektet. Efter at der er blevet introduceret ny teori, har der til tider været problemer med forståelsen af dette, på grund af fejllæsninger eller mistolkning af teori. Nogle gange har der ikke været en fælles forståelse for et givent emne, som har afsportet arbejdet.

Når dette er skete, blev teorien gennemgået i plenum, for at ensrette forståelsen. Når alle teoriens aspekter var gennemgået, blev arbejdet genoptaget og typisk ville problemet så være løst.

## 4 Udviklingsprocessen

De faser som udviklingsprocessen ligger vægt i er Inceptions- og Elaborationsfasen, hvor Elaborationsfasen har været opdelt i to iterationer, disse iterationer er skulle deles op efter brugsmønstre, gruppen valgt at ligge vægt på både opret sag og find sag i begge iterationer, første iteration gik dog ud på at lave et domæne lag som der kunne arbejdes videre ud fra i anden iteration.

I Inceptionsfasen har gruppen haft fokus på at udarbejde aktørliste, brugsmønstre med kortebeskrivelse, samt en beskrivelse af hvad der skal gennemgås for at udføre de givende brugsmønstre. Denne detaljering af brugsmønstre beskrivelsen havde først været forventet at blive udført i løbet af Elaborationsfasen efterhånden som man begynder at arbejde med de enkelte brugsmønstre. Der er dertil også blevet udarbejdet et overordnet brugsmønsterdiagram med systemafgrænsning, som viser hvilken aktører deres kan interagere med hvilken brugsmønstre i systemet. Der blev også i Inceptionsfasen lavet et first-cut domæne klasse diagram, hvilket skulle have været udarbejdet i løbet af Elaborationsfasen efterhånden som der blevet arbejdet med de forskellige brugsmønstre.

I Elaborationsfasen har gruppen arbejdet ud fra brugsmønstrene opret sag og søger sag, her er det blevet udarbejdet sekvensdiagrammer til at forklar hvordan systemet gennemløber brugsmønstre, sekvensdiagrammer har fået tilføjet metode navne med parameter. Der er blevet udarbejdet kontrakter, som giver en forklaring af hvad de forskellige metoder indenfor brugsmønstrene skal stå for og hvad man kan forvente at få af output fra dem, samt hvad der skal gøres før metoden virker. Der er i designfasen af Elaborationsfasen blevet udarbejdet et arkitektonisk sekvensdiagram, som fortæller hvad der sker når bruger arbejder med systemet. Der blev også i designfasen udarbejdet et analyse klasse diagram som blev udarbejdet så metoder og klasser stod klar til implementering, i implementeringsfasen blev der lagt vægt på at få et grafisk udseende for opret sag og søger sag, samt få lavet logikken i domænet og persistens laget, så det giver et indblik i hvad gruppen havde tænkt sig hvis projektet gik videre til den næste fase af Unified Proces.

## 5 Formidling og kommunikation

Gruppen har haft 1-2 ugentlige møder hvor vi har siddet på Syddansk Universitet og snakke om hvad de forskellige parproduktionsgrupper har nået siden vi sidst snakket sammen. Hvis der er nogen af grupperne som ikke har nået det vi havde aftale sad vi sammen som en fælles gruppe og lavet de forskellige ting, inden dagens møde sluttet aftale gruppen hvad der skulle ske til næste gang og aftale en dagsorden for næste dag, og hvis det næste møde lå til en tirsdag blev der også lavet en dagsorden til vejledermødet og sendt til vejleder hvis der var noget punkter til dette møde. Til at kommunikere har gruppen benyttet undervisnings dage, såvel som Facebook og Discord. Facebook er blevet brugt til lette beskeder, så som hvis nogen er syge eller forsinket til et møde. Discord er blevet brugt til beskeder af lidt mere substans, hvor der intet er krævet et telefonisk møde eller video chat, for at hvis man mangler hjælp til eller et evt. svar til det stillede spørgsmål. Discord er også blevet brugt til at holde nogen af de ugentlige møder i tilfælde af at der har enighed om at parproduktionsgrupperne har kunne arbejde hjemme fra, så er der blevet holdt et opsummeringsmøde midt eller sidst på dagen så alle i projektgruppen er uptodate med hvad der er sket.

## 6 Samarbejde i gruppen

Gruppen har arbejdet med parproduktion, hvor grupperne har ændret sig fra iteration til iteration eller når noget andet større begyndte. Dette har været en fordel for gruppen at der var en marker at spørge før et spørgsmål noget ud til gruppen. Dette har været med til at gøre at de spørgsmål der blev stillet i gruppen var mere præcise, da de allerede var kommet et input fra en anden person, som ofte har kunnet svare på dele hvis ikke hele spørgsmålet.

Generalt set har gruppen fungeret godt, med at der ikke var større problemer. Dette mener vi selv skyldes at gruppodynamikken blev sat fast tidligt, men ikke så fast at der ikke var til at udvikle

sig igennem projektet. Der har været nogle misforståelser under vejs, som er håndteret som skrevet tidligere hvilket har været med til at skabe en større forståelse i gruppen.

## 7 Samarbejde med vejleder

Gruppen har haft et godt samarbejde med Henrik, når gruppen har haft nogen spørgsmål omhandlende dele af projektet er han kommet med et svar som inden har givet gruppen en dybere forståelse for problemet ellers har han formået at give en feedback så det har været muligt at se opgaven fra et andet synspunkt og derved også gøre det muligt at løseproblemene uden yderlige hjælp. Henrik har altid mødt velforberedt op til vejleder møderne og vist at han ønsker at gruppen har lykkes med de opgaver som har været et problem for gruppen, hertil har Henrik også altid været forberedt på at gruppen kommer med et problem som ikke er blevet oplyst til ham på forhånd, men stadig indenfor meget kort til kunne komme med en løsning, om det så har været til udarbejdning af diagrammer eller selve implementering.

## C Kildekode

For at se gruppens javadoc gå til:

<https://github.com/Diplom-Software2018/Anden-semester/tree/gh-pages/ProjektMappen/MMMI/dist/javadoc>

For at se javadocs skal branchen hentes og ses der igennem.

## D Brugervejledning

OBS: Som programmets opbygning lige nu gøres der opmærksom på at der mangler nogle funktionaliteter.

Hvis der er problemer med at forbinde til databaseserveren, kan oplysninger omkring forbindelsen findes ved at gå ind under:

NetBeans projektet “MMMI\src\mmmi\Data\_layer\Connection.java”.

Under denne klasse kan variablerne url, username og password findes:

```
String url = "jdbc:postgresql://mmmihosting.ddns.net:3306/mmmidb";
```

```
String username = "pi";
```

```
String password = "MMMI_pi_server";
```

Når man skal til at benytte programmet vil du blive mødt af en login vindue. Dette vindue er et login til selve MMI systemet. Du skal dermed indtaste brugernavn og adgangskode og trykke herefter på knappen ”login”. (se figur D.1)

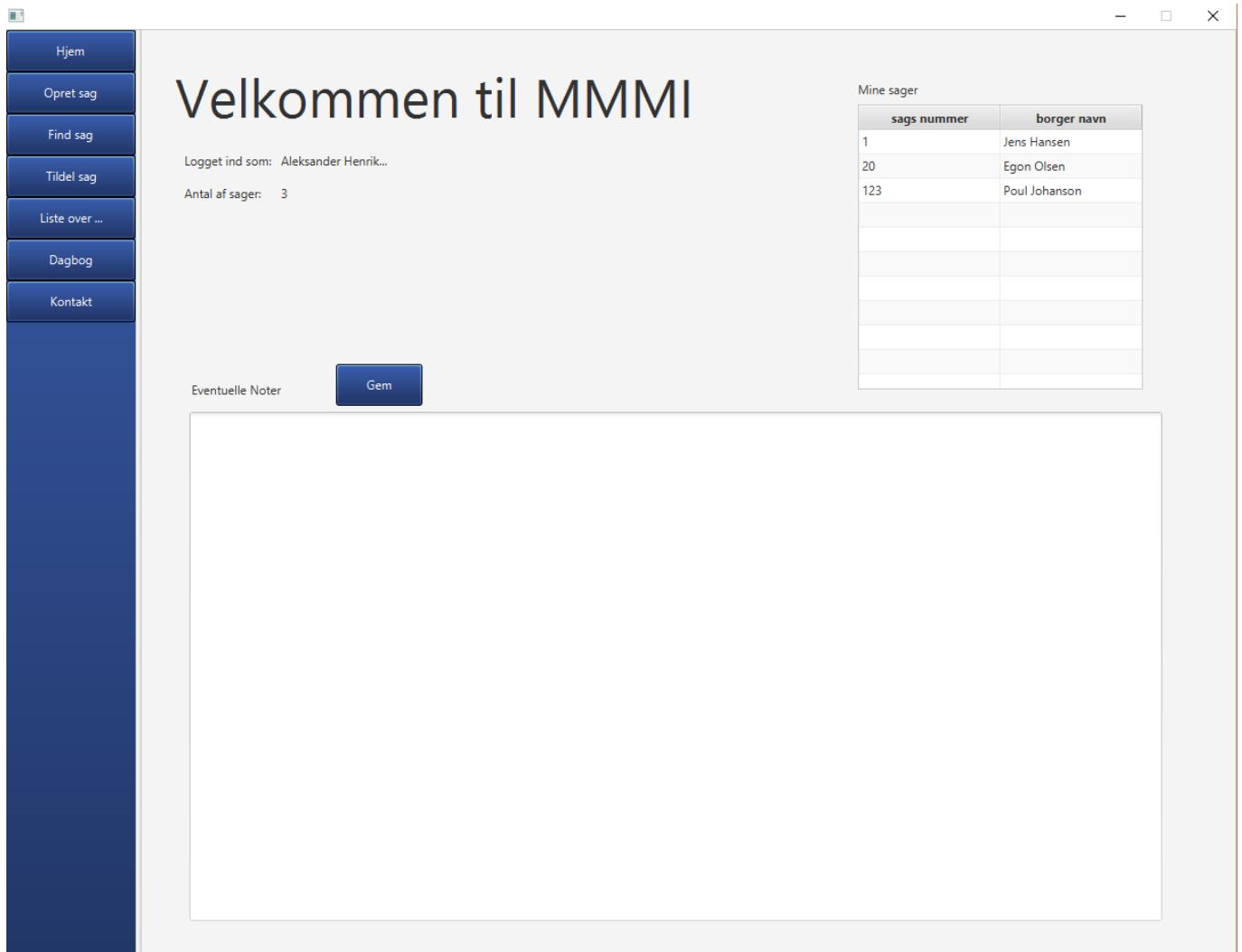


Figur D.1:

username	password
socialsecretary	secretary
socialdepartment	department
socialcase	caseworker
disabilitycase	caseworker
disabilitydepartment	department
disabilitysecretary	secretary

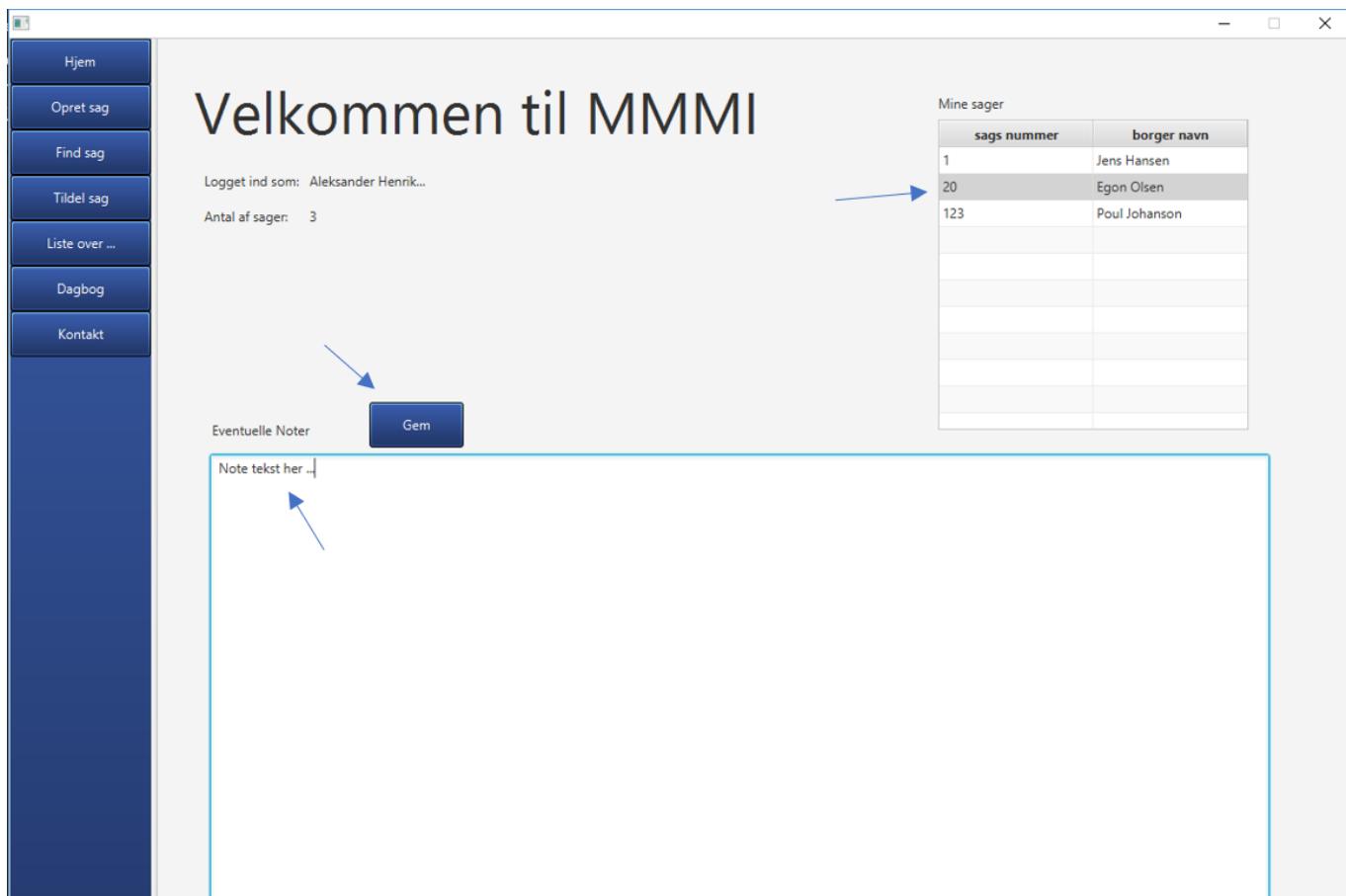
Figur D.2: Viser hvad der kan bruges til at logge ind

**Overordnet konto til test:** Når du har trykket på loginknappen, vil du blive viderestillet til hovedprogrammet MMMI. MMMI er designet til at være et sagsforløb program til at hjælpe sagsbehandlere med at løse sagsopgaver. Når du er logget ind får du vist følgende vindue, se figur D.3.



Figur D.3:

Når du er logget ind får du vist ”hjem” fanen hvor der kan ses hvem du er logget ind som, antal sager og hvilke sager man har i højre side (figur 2). I tekstfeltet under ”gem” knappen kan der skrives en note til en sag ved at markere sagen i sagsdisplayet, se figur D.4.



Figur D.4:

Når der trykkes på opret sag, vil der blive vist en sagsåbningsformular som kan udfyldes. I sagsåbningsformularen skal henvendelsesfeltet udfyldes før en sag kan gemmes. Det er vigtigt at den lille ”gem knappen” i dokumentet skal klikkes på først, før man fortsætter, så de ting der er udfyldt gemmes. (se figur D.5)

Hjem

Opret sag

Find sag

Tildel sag

Liste over ...

Dagbog

Kontakt

Henvendelse

Hvad drejer henvendelsen sig om?

Beskriv henvendelsen

Er det klart hvad borgeren søger om?

Ja Nej

Vedrørende borgers

Har den henvende borgers informationer omkring den borgers det drejer sig om? Eller er det den vedrørende borgers selv der har henvendt sig?

Ja Nej

Henvendelse

Hvor kommer henvendelsen fra?

- Borger
- Pårørende [angiv kontaktoplysninger]
- Lege [angiv kontaktoplysninger]
- Hospital [angiv kontaktoplysninger]
- Anden forvaltning [angiv kontaktoplysninger]
- Igangværende indsats [angiv kontaktoplysninger]
- Anden kommune
- Andre [angiv kontaktoplysninger]

Er borgeren indforstået med henvendelsen?

Ja Nej

Rettigheder og pligter

Aftaler om det videre forløb

Indhentelse af oplysninger

Borgerinddragelse - særlige forhold

Handle- og betalingskommune

Save

Næste

Gem

Figur D.5:

I sagsåbningsformularen kan der skrives oplysninger omkring den henvende borgers samt den vedrørende borgers. (se figur D.6)

**Hvad drejer henvendelsen sig om?**

Beskriv henvendelsen

Er det klart hvad borgeren søger om?

Ja  
 Nej

**Vedrørende borgers**

Har den henvende borger informationer omkring den borger det drejer sig om?  
Eller er det den vedrørende borger selv der har henvendt sig?

Ja  
 Nej

Hvis ja, angiv (mulige) kontaktoplysninger

Navn:  
Efternavn  
Cpr-nummer  
Postnr/By/Adresse  
Husnummer  
Hvis lejlighed, indtast etage og etage retning (f.eks. 2 etage, højre)

Rettigheder og pligter  
Aftaler om det videre forløb  
Indhentelse af oplysninger

Næste      Gem

Figur D.6:

Før en ydelse kan blive valgt, skal borgeren være bekendt med hvad han/hun søger om. Hvis ja er markeret, bliver en ny fane stillet til rådighed og der kan vælges ydelser. (se figur D.7)

**Hvad drejer henvendelsen sig om?**

Beskriv henvendelsen

Er det klart hvad borgeren søger om?

Ja  Nej

**Vedrørende borgers**

Har den henvende borger informationer omkring den borger det drejer sig om?  
Eller er det den vedrørende borger selv der har henvendt sig?

Ja  Nej

► Hvis ja, angiv (mulige) kontaktoplysninger

► Rettigheder og pligter  
► Aftaler om det videre forløb  
► Indhentelse af oplysninger  
► Borgerinddragelse - særlige forhold  
► Handle- og betalingskommune

▼ Ydelser og paragraffer (Hvis ja er afkrydset)

Behandling

▼ Behandling (liste)

- Misbrugsbehandling (§ 101)
- Misbrugsbehandling (SUL § 141)
- Terapi (§ 102)
- Speciallæge behandling (§ 102)
- Psykologisk behandling (§ 102)

Kontrolordning

► Kontrolordning (liste)

Socialpædagogisk støtte

▼ Socialpædagogisk støtte (liste)

- Støtte til administration (§ 85)
- Støtte til indkøb og kost (§ 85)
- Støtte til kontakt og samvær (§ 85)
- Støtte til medicinhåndtering (§ 85)
- Støtte til personlig pleje (§ 85)
- Støtte til praktiske opgaver i hjemmet (§ 85)
- Støtte til uddannelse og beskæftigelse (§ 85)
- Støtte til varetagelse af forældrerollen (§ 85)

Støttepersonordning

► Støttepersonordning (liste)

Næste Gem

Figur D.7:

Der kan også vælges andre elementer såsom ”aftaler om det videre forløb” eller ”rettigheder og pligter”. Det eneste der skal gøres for at få de elementer frem er ved at folde dem ud ved at klikke på den vandrette lille pil ved siden af teksten til venstre. (se figur D.8)

Figur D.8:

Når der er blevet fortaget de nødvendige ændringer, kan sagen gemmes ved at trykke på ”save” knappen. Nu kan der fortsættes med sagsudredningen ved at trykke på ”næste” knappen og der vil blive vist en sagsudredningsformular. I sagsudrednings formularen kan der ligesom i sagsåbningensformularen, udvides faner som har forskellige felter der kan udfyldes. Når man er færdig trykker man på den lille ”gem” knap og herefter trykkes der på den store ”gem” knap”, der gemmer hele dokumentet i systemet. Se figur D.9

Figur D.9:

Man kan også søge på en borger (se figur D.10) hvor der kan søges på fire forskellige kriterier og dermed kan man søge de sager frem som en borger er tilknyttet og man får en oversigt vist i det store oversigtsvindue. Ydermere kan der også vælges at søge på et sagsnummer ved at skrive nummert på sagen og herefter trykke på knappen ”søg”. (se figur D.11)

Figur D.10:

Hjem

Opret sag

Find sag

Tidlig sag

Liste over ...

Dæbog

Kontakt

Find en sag

Borger

Sagsnummer

Sagsnummer:

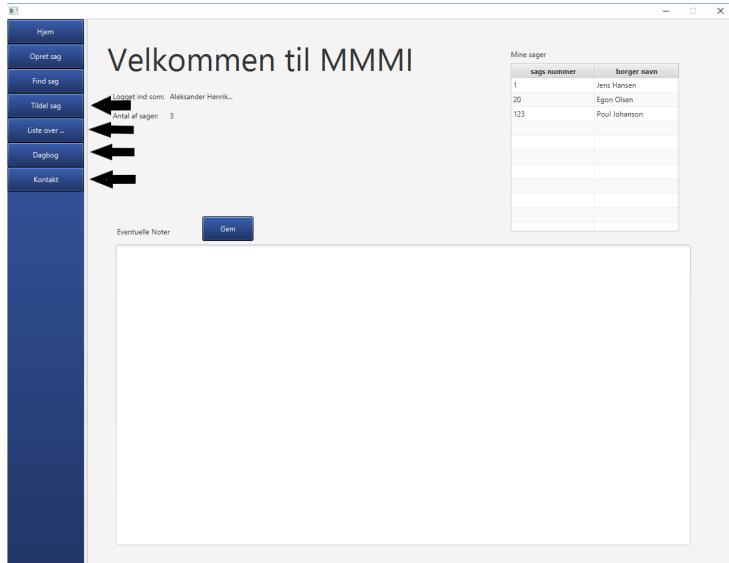
Øgning

Åben sag

Figur D.11:

Udover det er der 4 andre knapper, ”tildel sag”, ”liste over sager”, ”dagbog” og ”kontakt” som ikke er blevet implementeret med funktionalitet og kan derfor ikke benyttes (klikkes på). (se figur D.12)

For at logge ud igen skal vinduet bare lukkes ned ved at trykke på det røde kryds i MMMI vinduet i



Figur D.12:

øverste højre hjørne. Hvis man ønsker at logge ind igen skal man gøre som i starten i brugervejledningen.

## E Projektlog

For at se hvordan gruppen har arbejdet igennem dette projekt.

<https://github.com/Diplom-Software2018/Anden-semester>

## F Interne bilag

### 1 Projektforslag



Det Tekniske fakultet

Gruppe 02

Semesterprojekt for Software Teknologi og Software Engineering Aflevering: MMMI 22/02-19

# Mangfoldigt Manager Management Instrument (MMMI)

**Afleveringsdato:**

22/02-2019

**Semester gruppe 02**

**Projektgruppen:**

Aleksander Grzegorz Duszkiewciz, alodus17@student.sdu.dk

Aleksander Henriksen, alhen18@student.sdu.dk

Aslak Mangino Nicolajsen, asnic18@student.sdu.dk

Mathias Jakobsen, mjako18@student.sdu.dk

Per Terp Larsen, pertl18@student.sdu.dk

Steffen Vitten Pedersen, Stped18@student.sdu.dk

**Vejleder:**

Henrik Lykkegaard Larsen, hlla@mmmi.sdu.dk

Side 1 af 13

## Centrale Problemstilling:

### Baggrunden for Problemstilling

Gruppen har fået udleveret en case som omhandler et system udviklet af EG Team Online ApS, virksomheden ønsker en belysning af 3 moduler planlægning, sagsudredning og dagbogen. Gruppen har diskuteret afgrænsning i forhold til de specifikke moduler, hvor der var en interesse i hvordan flere roller i systemet interagerer med hinanden. I forhold til persondatalovgivningen har virksomheden beskrevet en dataafgrænsning i casen hvor de beskriver hvem der må kunne interagere med hvad i systemet.

Gruppen kom frem til følgende hovedspørgsmål.

Hvordan kan et program håndtere flere roller uden at de interagerer med hinandens data, samt hvilke sikkerhedsforanstaltninger ligger til grund for at beskytte data?

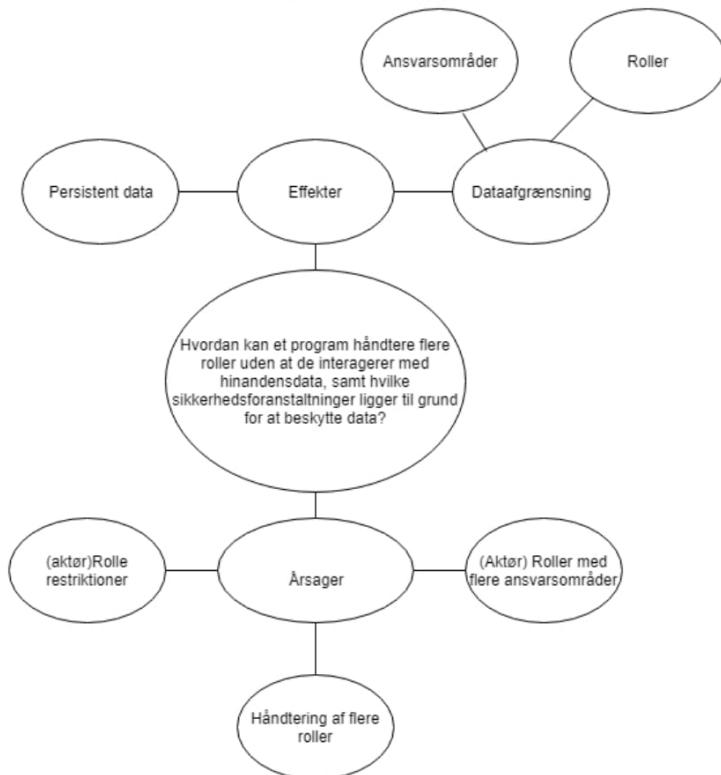
### Problemformulering:

På baggrund af den centrale problemstilling er der blevet udformet følgende underspørgsmål:

- Hvad definerer roller?
- Hvilket ansvarsområder ligger til grund for rollerne?
- Hvilken persistent data bearbejdes?
- Hvordan håndterer systemet flere roller?
- Hvordan håndteres roller med flere ansvarsområder?
- Hvilke sikkerhedsforanstaltninger er der i forhold til data afgrænsningen?
- Hvordan sikrer vi at aktoren kun har adgang til det aktoren har brug for?

Gruppen lavede en analyse af den udleverede case. På baggrund af analysen dannede gruppen et overblik over problemstillingen. Analysen belyser de forskellige aktørers roller og ansvarsområder. EG-teamet ønsker en belysning af data afgrænsningen i forhold til sikkerhedsforanstaltninger, der kan sikre

personfølsomme oplysninger, hvis muligt.



## Faglige og Personlige motivation:

// relevant til studie Håndbogen

//hvor er det relevant i forhold til problemstillingen

// i forhold til semester Håndbogen

Vi lever i en mere digital verden, hvor personlige data er mere blottet end nogensinde før. EG Team Online har udviklet en platform, hvis primære formål er at give sagsbehandlere og socialarbejdere et fælles instrument, hvor de kan håndtere dokumentationen omhandlende borgernes forløb og behandling. EG Team Online ønsker en belysning på 3 moduler indenfor den platform. Gruppen ønsker at arbejde med dataafgrænsningen for at få erfaring indenfor datasikkerhed i den digitale verden. Med vores viden indenfor software engineering og yderligere fag vil gruppen forsøge bedst muligt at bruge denne viden i praksis.

Gruppen har igennem en afstemning af individuelle forventninger nået frem til følgende punkter:

*Aleksander D:*

Ønsker et godt projekt igennem et godt samarbejde, samtidigt med en uformel tone, så projektarbejdet bliver mere flydende og idet får en faglig forståelse samtidigt. Og til sidst, et projekt man kan være stolt af.

*Per:*

Ønsker et underholdende samarbejde i gruppen til at udføre et professionelt opstillet projekt. Forventer et fornuftigt resultat af projektet.

*Mathias:*

Ønsker et godt fællesskab til at udføre et godt produkt, hvilket vil give et godt slutresultat.

*Aslak:*

Ønsker faglig udnyttelse af undervisning i projektet for at kunne skabe en bedre forståelse af projektforløbet, hvilket Aslak kan stå inde for.

*Aleksander H:*

Går op i læringsprocesserne, hvor undervisningen spiller en central rolle, for udviklingen af projektet.

*Steffen V:*

Går op i videreudviklingen af kompetencer gennem hele forløbet hvor muligheden for læring af gruppen, er en essentiel del af projektarbejdet.

### **Gruppens samlede mål:**

Gruppens samlede mål er en samling af de individuelle mål. En opsummering af målene er, fælles kompetenceudvikling i forhold til projektarbejdet, hvor der udvikles et produkt ved hjælp af et godt samarbejde og fælles interesse i fagligheden.

### **Nuværende viden:**

Sensum bosted er et modulbaseret produkt der er specialiseret til at løse hverdagsopgaver indenfor den sociale sektor. Kommunerne har brug for et overblik over de processer der ligger til grund for

sagsbehandling af den enkelte borger. Sensum er en overskuelig og brugervenlig informationsplatform til sags- og udredningsformål. Behovet for at skabe et overblik over sagsbehandlingen samt faciliteterne ved kommunernes arbejdsprocesser, ligger til grund for Sensum's udvikling. Sensum er et DHUV system der varetager journalføring/dokumentation ved behandling af udsatte voksne og handicappede. Systemet dækker arbejdsgangen fra borgens henvendelse til sagsafslutning, som er beskrevet i VUM.

EG-Teamet ønsker et produkt baseret på den udleverede case. Dette produkt til Sensum bosted, er baseret på disse 3 moduler:

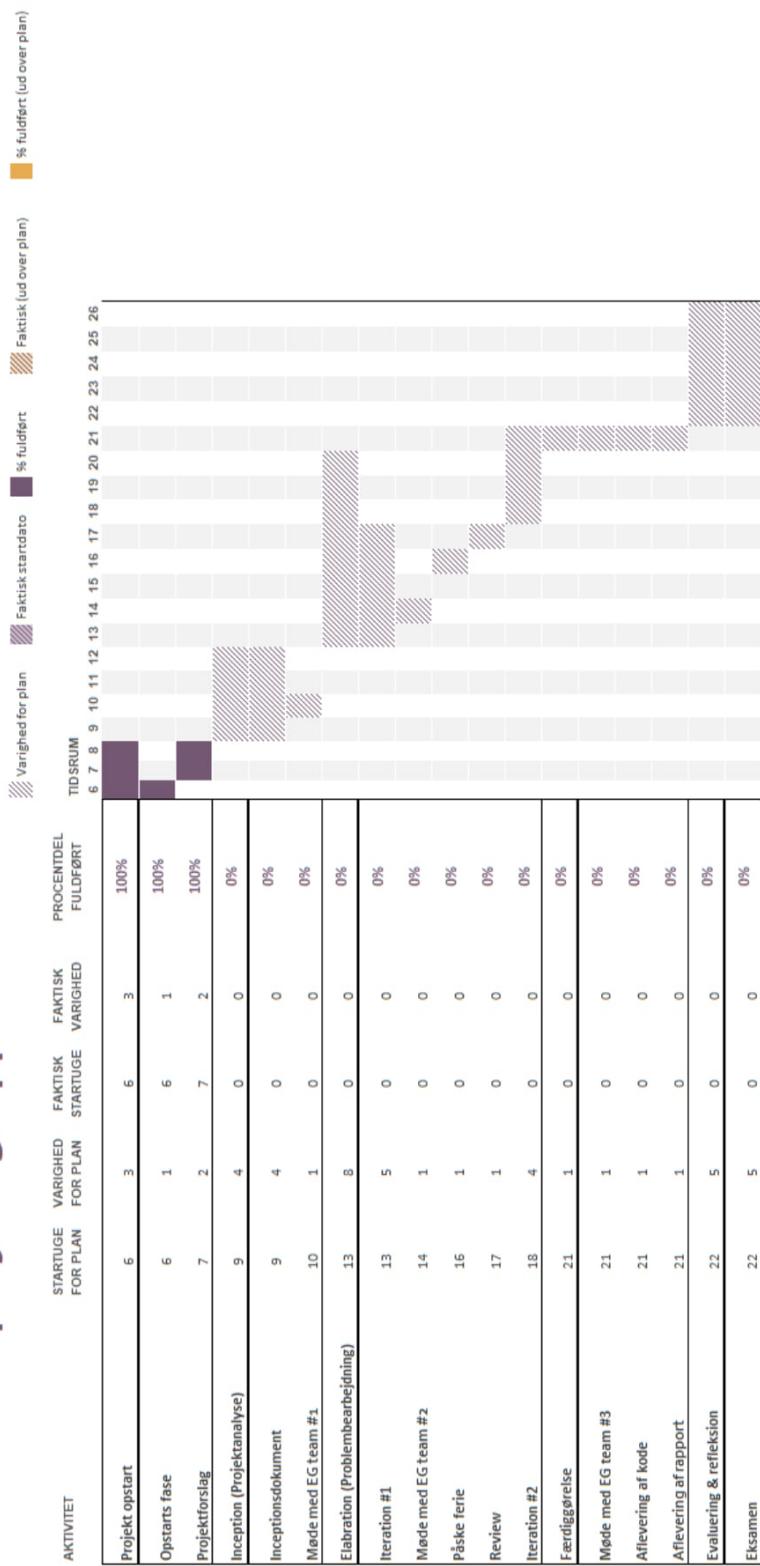
- Sagsudredning
- Dagbogsføring
- Planlægning

### Manglende viden

- Skal vi tage højde for forskellige autisme spektrum forstyrrelser?
  - Hvor mange aktører er der? (kun I for hold til det vi skal lave)
  - Hvordan ser Sensum Bosted ud grafisk?
  - Hvordan er journalerne opbygget, hvilke informationer skal den indeholde?
  - Hvilken type forbindelse er der mellem sensum og plugins?
  - "... Vil EG Team Online gerne have belyst, hvordan man kan samle systemerne under en hat, og samtidigt sikre, at man kun ser data for lige præcis det, man har behov for, samt sikre at man kun ser og opbevarer data, som borgere eller pårørende har afgivet samtykke til."
- Hvad definerer "**behov** for" i denne sammenhæng?
- For at lave et nyt data set til en bruger hvilke oplysninger skal sagsbehandleren bruge?

## Rammetidsplan:

### Semester projekt gruppe 02



## Gruppeværktøjer:

Gruppen har valgt at bruge følgende værktøjer:

- Arbejde i netBeans og IntelliJ, hvor koden afleveres som et netBeans projekt.
- GitKranken
  - Kanban (GitKranken Glo)
  - Versionsstyring
  - Kalender
- Scrum
- UP (Unified Proces)
- Simul (Dokumentstyring)
- GitHub (<https://github.com/Diplom-Software2018/Anden-semester>)
  - Logbog
  - Referat
  - Grupperekumentation (Kontaktinformation, samarbejdsafstale, m.m.)
  - Versionsstyring
- Kommunikation
  - Discord
  - Messenger (Facebook)
  - Mail

## Samarbejdsaftale:

### Projektgruppen:

Aleksander Grzegorz Duszkiewicz, aldus17@student.sdu.dk

Aleksander Henriksen, alhen18@student.sdu.dk

Aslak Mangino Nicolajsen, asnic18@student.sdu.dk

Mathias Jakobsen, mjako18@student.sdu.dk

Per Terp Larsen, pertl18@student.sdu.dk

Steffen Vitten Pedersen, Stped18@student.sdu.dk

[Flere kontakt oplysninger og informationer \(Link\)](#)

### Vejleder:

Henrik Lykkegaard Larsen, hlla@mmmi.sdu.dk

### Kommunikation

1. Konflikter: Få dem løst med det samme, der er hverken tid eller overskud til at høre på drama. Det skal ikke opbygges
2. Obligatorisk besked til alle i gruppen, ved frafaldelse. Giv besked over facebook. Kan du give besked før end den pågældende dag, du ikke vil være her, så gør det. Det hjælper
3. Har du problemer med at nå din deadline, så giv lyd, da vi er flere i gruppen der så kan hjælpe med opgaven. Vi er sammen om det.
4. Uformel tone, dog hold arbejdsmoralen og pjatter du for meget, så løb en tur rundt om hele uni. Vis gensidig respekt

## Generelle aftaler

---

1. Mød hver tirsdag 8-14, og fredag 12-14(Tider kan forlænges). Og til aftalte tidspunkter 7-14 dage frem.
2. Overhold de aftaler der bliver lavet indbyrdes i gruppen fra gang til gang
3. Giv gerne lyd fra dig, hvis der skal holdes et møde udenfor de normale mødetider.
4. Følg godt med til alle fagene og gerne arbejde med opgaverne så man er beredt på det meste
5. Husk at kommunikere gennem facebook og tjek beskeder og svar venligst, med det samme.
6. Der findes ingen dumme spørgsmål. Spørg løs, måske kan vi svaret, men der kan altid hjælpes til at finde et svar.

## Arbejdsformer

---

1. Til kodningen: [Følge gruppens standarter på github \(Link\)](#)
2. Arbejd gerne 2 og 2 sammen, så har man altid en at kontrollere med og kan debattere opgaverne med.
3. Sidst på dagen vil der være en opfølgnings på hvad der er blevet lavet, hvilke overvejelser der har været, løsninger og deslige. Gøres for at få en større samlet indsigt og viden indenfor emnet og opgaver
4. Start af dagen vil der være en lille briefing, gennemgåelse af hvad der lavet hjemme, hvad man har fået af idéer, hvad der kunne bruges hjælp til og hvad vi hver især tænker, dagen skal bruges på.
5. Huske at lave referat af dagsordenen til næste vejleder møde. informationer til vejleder skal være klar senest fredag. [Se vejlederaftale \(Link\)](#)
6. Vi har ca. en times pause på en normal arbejdes dag. Folk vælger selv hvornår de holder pause.

**Belbin**

Eget udsagn:

Rolle	Eget udsagn
Idemand	Steffen Aslak
Kontaktskaber	Aleksander D
Koordinator	Steffen, Aleksander H og D, Aslak, Per
Opstarter	Steffen, Aslak, Per
Analysator	Aslak, Mathias, Per
Formidler	Aslak
Organisator	Aleksander D og H, Steffen, Per
Afslutter	Mathias, Per
Specialist	Mathias, Per

### **Belbin Test:**

Rolle	Foretrukne	Håndterbare
Idemand	Steffen	Aleksander D og H, Aslak, Per
Kontaktskaber	Aleksander H	Steffen, Mathias
Koordinator	Steffen, Aleksander H, Aslak	Aleksander D, Per
Opstarter	Steffen, Aslak, Per	Aleksander H
Analysator	Aleksander D og H, Mathias, Per, Steffen	Aslak
Formidler	Mathias	Aleksander D og H, Aslak
Organisator		Aleksander D, Aslak, Steffen, Per
Afslutter	Aleksander D, Mathias, Per	Aleksander H, Aslak
Specialist	Mathias, Per	Aleksander D

Hvis personen ikke er på rollen så er de "mindst foretrukne".

### **Hvem har ansvaret:**

- Planlægger: gitkraken, issue: Aleksander H.
- Ordstyrer, dagsorden i forhold til planlæggelse: Aslak
- Referent: Steffen og Aleksander D.
- Rapporten: Per og Mathias.

## Vejlederaftale:

### Vejleder:

Henrik Lykkegaard Larsen (hlla@mmmi.sdu.dk)

### Medlemmer:

Aleksander Grzegorz Duszkiewicz, aldus17@student.sdu.dk

Aleksander Henriksen, alhen18@student.sdu.dk

Aslak Mangino Nicolajsen, asnic18@student.sdu.dk

Mathias Jakobsen, mjako18@student.sdu.dk

Per Terp Larsen, pertl18@student.sdu.dk

Steffen Vitten Pedersen, Stped18@student.sdu.dk

[Flere kontakt oplysninger og informationer \(Link\)](#)

### Rettigheder og forpligtelser:

#### Forventninger til “Vejleder”:

- Gruppens forventninger til vejlederen lyder på.
  - Vejleder læser dagsorden igennem, inden mødet. dagsorden findes på github send mail med link.
  - Vejlederen kigger gruppens indvendinger igennem på baggrund af efterspørgsel.
  - Vejleder møder op forberedt til møderne.
  - Vejleder meddeler gruppen omkring forsinkelser eller evt. mødeskift.
  - Vejlederen i samarbejde med gruppen laver en fast mødeplan.
  - Vejlederen har mulighed for at blive kontaktet uden for mødedage. til evt problemer i henhold til kode eller proces. Denne kontakt metode holdes gennem vejlederen valgte metoder.
  - Vejlederen gennemgår materiale vi sender
  - Vejleder holder gruppen inde for rammer som er realistiske i forhold til atstå med et færdigt projekt.
  - Vejlederen kommer med input der er relevant i forhold til gruppens projekt, for at kunne understøtte gruppens tanker om projektet.
  - Vejleder tjekker gruppens individuelle forventninger, hvilket findes i forventning aftale.

**Forventninger til gruppen:**

- Vejlederens forventninger til gruppen lyder på at:
  - Gruppen stiller konstruktive klare spørgsmål under vejleder møderne.
  - Gruppen har lavet en velovervejet dagsorden med klare retningslinjer til møderne senest fredag.
  - Gruppen er velforberedte til møderne.
  - Gruppen gør en indsats for at løse de problemer der kan opstå inden der tages kontakt med vejleder.
  - Gruppen er engageret i samarbejdet samt udførelsen af projektet.
  - Gruppen følger med i undervisningen.

## 2 Inceptionsdokument

Inceptions fasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Vejleder:

Henrik Lykkegaard Larsen [hlla@mogui.sdu.dk](mailto:hlla@mogui.sdu.dk)

Gruppemedlemer:

Aleksander Grzegord duszkiewicz	<a href="mailto:aldus17@student.sdu.dk">aldus17@student.sdu.dk</a>
Aleksander Henriksen	<a href="mailto:alhen18@student.sdu.dk">alhen18@student.sdu.dk</a>
Aslak Mangino Nicolajsen	<a href="mailto:asnic18@student.sdu.dk">asnic18@student.sdu.dk</a>
Mathias Jackobsen	<a href="mailto:mjacok18@student.sdu.dk">mjacok18@student.sdu.dk</a>
Per Terp Larsen	<a href="mailto:pertl18@student.sdu.dk">pertl18@student.sdu.dk</a>
Steffen Vitten Pedersen	<a href="mailto:stped18@student.sdu.dk">stped18@student.sdu.dk</a>

## Indhold

1.	Indledning .....	2
2.	Fremgangsmåde.....	4
3.	Business Case (referat) .....	5
4.	Forretningsområdet og eksisterende løsninger .....	6
5.	Overordnet kravspecifikation .....	7
5.1	Sensum (Log ind simulering) beskrivelse:.....	7
5.2	Sagsforløb.....	7
5.2.1	Aktørliste med beskrivelser .....	8
5.2.2	Brugsmønster med korte beskrivelser.....	9
5.2.3	Overordnet brugsmønstermodel.....	10
5.2.4	Detaljeret brugsmønster .....	11
5.2.5	Overordnet beskrivelse af supplerende krav.....	14
5.2.6	Domænemodel m. klassediagram og beskrivelse af domænet og klasserne. ....	16
6.	Kritiske risici .....	22
7.	Prioritering.....	24
8.	Metode i elaborationsfasen.....	26
8.1.	UP (Unified Process).....	26
8.1.1.	Hvad er UP.....	26
8.1.2.	Elaborationsfasen.....	27
8.2.	Scrum .....	28
8.3.	Kanban .....	29
8.4.	Par produktion .....	30
9.	Resurser .....	31
10.	Konklusion.....	32
11.	Bilag.....	i
11.1.	Logbog.....	i
11.2.	OL Cases.....	ii

## 1. Indledning

Salget af KMD førte til etableringen af det kommunale IT-fællesskab KOMBIT, som arbejder på at sikre de bedste og billigste IT-løsninger set ud fra kommunernes behov. De kommunale behov blev tidligere varetaget af KMD, men som resultat af deres monopolbrud, valgte KOMBIT at udbyde disse behov til forskellige IT-leverandører<sup>1</sup>. Dette skabte muligheden for at EG Team Online kunne træde ind på det kommunale IT-marked.

Allerede i 2005 præsenterede EG Team Online, dengang kendt som Team Online, deres første store system inden for velfærdsområdet, Sensum Bosted, dengang kendt som Bosted System. Dette system var revolutionerende inden for lokale bosteder og institutioner som arbejder med patientbehandling. Ved at digitalisere journaler og andre former for papirarbejde, blev socialarbejdernes arbejdsbyrde mindsket og deres effektivitet øget. I 2007 modtog Team Online ansvaret for it-løsninger hos Fyns Amt bo- og aktivitetsstedet Lindebjerg. Efterfølgende har virksomheden udviklet sig til at være en leverandør inden for innovativ teknologi med speciale i løsninger, der understøtter det samlede behov i den offentlige og private social- og sundhedssektor.

I 2018 blev GDPR indført af EU-kommissionen, hvilket gjorde at den danske regering tilpassede den danske datalovgivning til GDPR. Denne ratificering betød at alle IT-selskaber skulle følge nye regler angående håndtering af persondata. Derfor er EG Team Online ved at tilpasse deres produkt.

Gruppen har fået udleveret en case som omhandler et system udviklet af EG Team Online.

Virksomheden ønsker en belysning af tre moduler: planlægning, sagsudredning og dagbogen. Gruppen har diskuteret afgrænsning i forhold til de specifikke moduler, hvor der var en interesse i hvordan flere roller i systemet interagerer med hinanden. I forhold til persondatalovgivningen har virksomheden beskrevet en dataafgrænsning i casen hvor de beskriver hvem der må kunne interagere med hvad i systemet. På baggrund af de strengere regler om persondata, blev det nødvendigt at opdatere systemet.

---

<sup>1</sup> <https://www.kombit.dk/KOMBITHistorie>

Gruppen kom frem til følgende hovedspørgsmål.

Hvordan kan et program håndtere flere roller uden at de interagerer med hinandens data samt hvilke sikkerhedsforanstaltninger ligger til grund for at beskytte data?

### **Problemformulering:**

På baggrund af den centrale problemstilling er der blevet udformet følgende underspørgsmål:

- Hvad definerer roller?
- Hvilket ansvarsområder ligger til grund for rollerne?
- Hvilken persistent data bearbejdes?
- Hvordan håndterer systemet flere roller?
- Hvordan håndteres roller med flere ansvarsområder?
- Hvilke sikkerhedsforanstaltninger er der i forhold til data afgrænsningen?
- Hvordan sikres det at aktøren kun har adgang til det aktøren har brug for?

Gruppen lavede en analyse af den udleverede case. På baggrund af analysen dannede gruppen et overblik over problemstillingen. Analysen belyser de forskellige aktørers roller og ansvarsområder. EG-teamet ønsker en belysning af 3 moduler, hvor gruppen valgte dataafgrænsningen i forhold til sikkerhedsforanstaltninger, der adskiller de forskellige roller.

Under inceptionsfasen (Life Cycle Objectives) bliver målene sat for, hvordan projektet skal udformes og hvilke forudsætninger skal der tages højde for. Nogle af målene for inceptionsfasen kan være aktørliste, brugsmønster og kravspecifikationer.

Dette skal skabe grundlaget for videre arbejde i elaborationsfasen, samt danne overblik over projektcasen.

## 2. Fremgangsmåde

Vi startede projektet med at læse og analysere den udleverede case. For at få et bedre indblik i systemet blev der læst op omkring VUM-metoden, der har fokus på udredning af borgeren. VUM er et sagsbehandlingsredskab der håndterer systematiske udredningsprocesser til at effektivisere DHUV<sup>2</sup> området. Herfra gik gruppen videre med at lave en problemstilling, for at få en bedre forståelse af casen. Efter en samlet vurdering omkring, hvilket produkt gruppen ville kaste sig ud i, blev der lavet en overordnet problemstilling samt hovedspørgsmål til casen. Som det næste blev der fundet aktører for hele systemet. Dette blev gjort ved at se på hvilke personer eller systemer der interagerer med systemet. Da alle aktører var identificeret, blev der lavet en overordnet brugsmönstermodel hvorefter der blev udarbejdet mere detaljerede beskrivelser for udvalgte brugsmönstre, hvilket skabte en forståelse for kravene i systemet. I samme periode blev der lavet en business case for EG Team Online. Dette kunne laves efter projektcasen var analyseret og sideløbende med resten af projektet. Da business casen var færdig, blev der skrevet et samlet referat.

Da de enkelte, detaljerede brugsmönstre var lavet, blev den overordnede brugsmönstermodel opdateret. Derefter blev aktørlisten opdateret i tilfælde af, at de detaljerede brugsmönstre havde skabt nye aktører eller nogle var blevet overflødige. Opstillingen af krav til systemet skete ud fra de detaljerede brugsmönstre sammen med analysen af projektcasen.

For bedst muligt at kunne effektivisere arbejdet blev parproduktion<sup>3</sup> benyttet. Følgende emner blev uddelegeret: ressourcer, metoder, prioritering, risici og domænemodel.

Konklusionen blev skrevet samlet af gruppen efter uddelegeret inceptionsdokumentet var gennemgået.

---

<sup>2</sup> <https://socialstyrelsen.dk/tvaergaende-omrader/sagsbehandling/voksenudredningsmetoden>

<sup>3</sup> Se metode afsnit parproduktion

### 3. Business Case (referat)

Der er i kurset Organisation og Ledelse blevet udarbejdet to cases, der beskriver det generelle og specifikke miljø omkring EG Team Online. For at beskrive disse miljøer er der blevet udarbejdet en PESTEL- og en Porters Five Forces analyse per case. PESTEL-analysen beskriver det generelle miljø og Porters Five Forces beskriver det specifikke. Disse cases er vedhæftet under afsnit 11.2 OL Cases.

I PESTEL-analysen er der lagt vægt på de politiske, økonomiske, sociale og kulturelle samt lovmæssige forhold. Det er primært disse forhold der påvirker EG Team Online i henhold til det generelle miljø. Afsnittet omhandlende de politiske forhold ligger vægt på den danske digitaliseringstrategi fra 2016, hvis største fokuspunkt er at staten, regionerne og kommunerne skal have et større samarbejde på tværs af de forskellige instanser og derved øge væksten for digitalisering.

De økonomiske forhold beskriver, hvilken økonomisk vinding digitaliseringstrategien giver IT-branchen. Der er fra 2019-2025 blevet afsat 125 millioner kr. årligt, hvor noget af denne fond skal bruges på at styrke det digitale vækstmiljø i Danmark.

I de sociale- og kulturelle forhold bliver der lagt vægt på hvordan tilgængeligheden af den nødvendige specialiserede arbejdskraft påvirker virksomheder i IT-branchen.

Under afsnittet om de lovmæssige forhold bliver der beskrevet hvordan et udbud bliver bearbejdet og vurderet af en offentlige myndighed. Der er to måder hvorpå det er muligt at gennemføre en udbudsproces. Den første af udbudsprocesserne går ud på at lave en prækvalifikation af virksomhederne. I denne proces bliver der stillet nogle minimumskrav til virksomheden, det kan f.eks. være til teknisk udstyr som deres server kraft. Den anden proces er en normal udbudsproces hvor der bliver udarbejdet en kravspecifikation som virksomheden svarer på og derefter bliver vurderet af den myndighed som har udsendt kravspecifikationen.

I Porters Five Forces analysen er der lagt vægt på at vurdere magtbalancen. Truslen fra nye virksomheder, beskriver hvor svært det kan være for en eventuel konkurrent at komme ind på markedet og hvad det kan koste en kunde at skifte til en anden løsning.

Der bliver under afsnittet om deres leverandørers forhandlingskraft beskrevet, at de kun har et hosting selskab som leverandører. Dette kan ikke påvirke dem på en negativ måde, da der er mange alternative hosting selskaber der kan benyttes.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019

Forhandlingskraften fra kunderne er ikke ret stor da udskiftningen af systemudbyder er forbundet med høje omkostninger, hertil skal der også tages højde for udbudsreglerne.

Dem som kan påvirke EG Team Online er eksisterende konkurrenter. EG Team Onlines største konkurrenter i Danmark er KMD og Systematic.

EG Team Online har deres primære domæne i den offentlige sektor. På baggrund af analyserne konkluderes det at miljøet generelt er stabilt og den mest velegnede organisationsstruktur er en mekanistisk.

#### 4. Forretningsområdet og eksisterende løsninger

KL<sup>4</sup> og socialministeriet har udviklet en metode til at lave sagsbehandling inden for udredning af handicappede og utsatte voksne. Denne metoder kaldes VUM og har til formål at forbedre de faglige og lovmæssige kvaliteter i sagsbehandlingen.

MMMI bliver udviklet som et plug and play system til Sensus Bosted, der skal styrke ledelsen og forbedre dokumentationen inden for både små og store socialfaglige tilbud. MMMI's forretningsområde ligger, ved behandling af socialfaglige sager, hos kommunerne.

Systemet tilbyder en professionel IT-løsning inden for sagsbehandling. Det er en platform til sikring af et pålideligt forløb for borgeren under udredningsprocessen. Her samarbejdes der på tværs af områder, hvor kommuner og institutioner kan samarbejde om borgerens forløb, samtidig med at det holdes afgrænset inden for specifikke områder. Eksisterende løsninger til Sensus er KMD'S Nexus og Systematic's Columana Cura.

Nexus er et system der ligger vægt på at arbejde på tværs af forvaltninger og fagligheder, hvilket gør det i stand til at levere det samme som Sensus. Nexus indeholder KMD CARE, hvilket er en løsning der gør den ansatte i stand til at arbejde på tværs af forvaltninger, samt KMD VIVA, hvilket er en løsning der samler digitale serviceydelse på tværs af fagsystemer.

Columana Cura er et system til at hjælpe ansatte inden for sundhedssektoren. Deres vision er at skabe de bedste rammer for offentligt ansatte i forhold til arbejdet med borgeren. Columana Cura er et system der er specialiseret i at håndtere dokumentation af sundhedsdata inden for sundhedsvæsnet.

---

<sup>4</sup> Kommunerne landsforening

## 5. Overordnet kravspecifikation

Sensum vil blive beskrevet i kravspecifikationen, hvor en simulering af et login fra Sensum vil finde sted (læs 5.1). Ydermere indeholder den gruppens valg af modulet sagsudredning, hvor systemet afgrænses til sagsforløb (Læs 5.2).

### 5.1 Sensum (Log ind simulering) beskrivelse:

Sensum-simuleringen bliver anvendt til at simulere selve loginsystemet fra Sensum.

Brugsmønstermodellen giver et overblik over hvordan denne simulering bliver anvendt i gruppens løsning.

### 5.2 Sagsforløb

I denne sektion beskrives sagsforløbets krav, der er analyseret ud fra de udarbejdede brugsmønstre. Sektionen er delt op i to dele.

Første del af sektionen indeholder:

- Aktørliste med beskrivelser
- Brugsmønstre med korte beskrivelser
- Overordnet brugsmønsterdiagram med systemafgrænsning
- Detaljeret brugsmønstre med beskrivelser
- Beskrivelse af de supplerende krav

Den anden del af sektionen består af:

- Domænemodel med klassediagram: herunder beskrivelse af domænet og forretningsbegreberne (klasserne)

### 5.2.1 Aktørliste med beskrivelser

Herunder ligger aktørlisten med beskrivelser for hver aktør:

Aktør	Beskrivelse
Sagsbehandler	Sagsbehandleren er personen der behandler en borgeres sag. Sagsbehandler skal kunne: <ul style="list-style-type: none"><li>• Oprette sag</li><li>• Afslutte sag</li><li>• Opfølgning af sag</li><li>• Behandle sag</li><li>• Find sag</li></ul>
Afdelingsleder	Afdelingsleder kan det samme som en sagsbehandler, samt: <ul style="list-style-type: none"><li>• Tildele sag</li><li>• Fjerne sag fra sagsbeandler</li></ul>
Sekretær	Sekretæren er en del af det administrative personale. Sekretæren skal kunne: <ul style="list-style-type: none"><li>• Oprette sag</li></ul>
Bruger	En bruger er en simulering af et login fra Sensum systemet, hvor en bruger bliver tildelt en rolle, f.eks. sagsbeandler, administrator, sekretær eller afdelingsleder.
Dagbog modul	Dagbog modulet modtager information under sagsafgørelsen.
Handleplan modul	Handleplan modulet modtager informationer under sagsbehandlingen og afgørelsen.
Sundhedssystem	Sundhedssystem tilføjer en lægenote/læge erklæring under sagsbehandlingen
CPR	Det administrative personale, anvender CPR til at indhente personlige oplysninger.

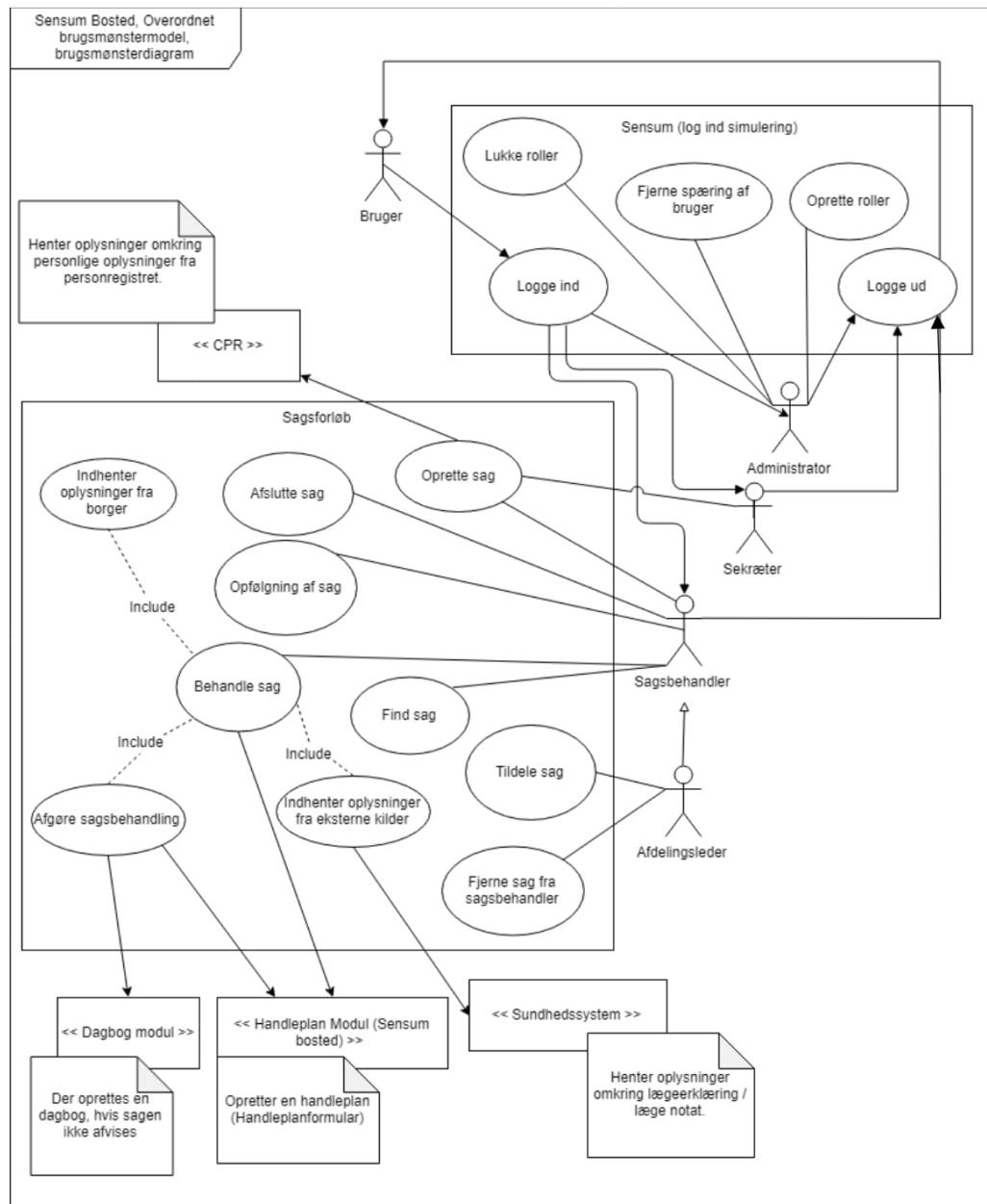
### 5.2.2 Brugsmønster med korte beskrivelser

Herunder er der udarbejdet brugsmønstre ud fra aktørbeskrivelserne:

Navn	Beskrivelse
Oprette sag	Der laves en sagsoprettelse. Personlige oplysninger kan indhentes fra personregisteret.
Indhenter oplysninger fra borgere	Borgerne giver oplysninger om hvorfor de har henvendt sig. Der vælges paragraf baseret på givne oplysninger for videre sagsudredning.
Find sag	Søger på en sag baseret på CPR-nummer, sagsnummer eller fuldt navn. Der vises en liste over sager, hvor der kan vælges en som bliver vist.
Behandle sag	Sagsbehandlerne starter behandling af en sag i forhold til indhentning af oplysninger, hvorefter sagen bliver vurderet. Efterfølgende kommer en sagsafgørelse og til sidst bestilles evt. ydelser. Der oprettes en handleplan, i andet modul indeholdende oplysninger fra pågældende sag.
Opfølgning af sag	En sagsbehandler skal kunne følge op på en pågældende sag for social indsats.
Indhenter oplysninger fra eksterne kilder	Der hentes relevante oplysninger fra sundhedssystem et al. f.eks. lægeerklæring, til videre udfyldelse af sagsudredningsdokument.
Afgøre sagsbehandling	Sagsbeandler registrerer afgørelse af pågældende sag og skriver et brev til borger omkring afgørelsen. Hvis sagen ikke er afvist, udfyldes en bestilling af social indsats.
Afslutte sag	Skal afslutte den valgte sag. Dette skal ikke slette den, men sikre at der ikke kan ske ændringer.
Tildele sag	Afdelingsleder tildeler en sagsbeandler en sag.
Fjerne sag fra sagsbeandler	Afdelingsleder fjerner en sag fra en sagsbeandler.

### 5.2.3 Overordnet brugsmønstermodel

Ud fra aktørbeskrivelserne og brugsmønsterbeskrivelserne er der udarbejdet en overordnet brugsmønstermodel, der viser systemafgrænsningen og alle forbindelser mellem aktør og brugsmønster. Se figur 1.



Figur 1 Brugsmønstermodel over sagsforløbet

Figur 1 viser to systemafgrænsninger. En af dem er sagsforløbet, som består af brugsmønstrene der udgør hele sagsudredningsprocessen. Den anden er Sensus som indeholder selve simuleringsdelen af login og administration af brugerrettigheder. Dette er med til at vise at en bruger skal logge ind og have tildelt en rolle som f.eks. sagsbehandler for at kunne benytte resten af systemet.

#### 5.2.4 Detaljeret brugsmønster

Der er blevet udvalgt nogle essentielle brugsmønstre, som gruppen har valgt at beskrive mere detaljeret. De udvalgte brugsmønstre har stor betydning for hele sagsudredningsforløbet.

Brugsmønster: Oprette sag	
<b>ID:</b> 1	
<b>Primær aktør:</b> Sagsbehandler, afdelingsleder, administrativt personale	
<b>Sekundære aktør:</b> CPR	
<b>Kort beskrivelse:</b> En aktør kan oprette en sag.	
<b>Prækonditioner:</b> Aktør skal være logget ind.	
<b>Hovedhændelsesforløb:</b>  Starter når en borgers henvender sig til kommunen, 1. Aktør indtaster CPR nummer, borgers navn, begrundelse for henvendelse. 2. hvis henvendelsen omhandler anden afdeling. 2.1. henvis borgers til relevant afdeling 3. hvis henvendelsen omhandler aktørs afdeling 3.1. Gemmer indtastet data. 3.2. Sagen oprettes.	
<b>Postkonditioner:</b> Henvendelse behandling.	
<b>Alternative hændelsesforløb:</b>	

Brugsmønster: Behandle sag	
<b>ID:</b> 2	
<b>Primær aktør:</b> Sagsbehandler	
<b>Sekundære aktør:</b> Handleplan modul	
<b>Kort beskrivelse:</b> Sagsbehandlerne starter behandling af en sag i forhold til indhentning af oplysninger, hvorefter sagen bliver vurderet. Efterfølgende kommer en sagsafgørelse og til sidst bestilles evt. ydelser.	
<b>Prækonditioner:</b> Der skal være oprettet en sag.	
<b>Hovedhændelsesforløb:</b>  Starter når sagsbeandler vælger “Behandle sag”. 1. Sagsbeandler udfylder en <a href="#">sagsåbningsformular (Link)</a> . 2. hvis <a href="#">sagsåbningsformular (Link)</a> ikke er udfyldt. 2.1. <i>Alternativt hændelsesforløb</i> Behandle sag: Manglende information 3. hvis <a href="#">sagsåbningsformular (Link)</a> er udfyldt 3.1. <a href="#">Udredningsformularen (Link)</a> udfyldes med data fra <a href="#">sagsåbningsformular (Link)</a> . 3.2. Der bliver sendt data til handleplan modulet ift. oprettelse af en handleplan. 3.3. Der vælges mellem alternativt forløb indhent oplysninger eller sagsafgørelse.	
<b>Postkonditioner:</b> None	
<b>Alternative hændelsesforløb:</b> Indhente oplysninger fra eksterne kilder, Afgøre sagsbehandling, Manglende information.	

Brugsmønster: Behandle sag: Manglende information	
<b>ID:</b> 2.1	
<b>Primær aktør:</b> Sagsbeandler, afdelingsleder, administrativt personale	
<b>Sekundære aktør:</b> None	
<b>Kort beskrivelse:</b> Bruges i tilfælde af at en <a href="#">sagsåbningsformular (Link)</a> ikke er tilstrækkeligt udfyldt.	
<b>Prækonditioner:</b> Der skal være startet en sag, med delvist udfyldt formular.	
<b>Alternativhændelsesforløb:</b>  1. Starter ved mangel på information i trin 2 i behandle sag. 2. Når formularen genoptages. 2.1. Åbnes formularen og manglende information udfyldes. 3. Hvis formularen er udfyldt. 3.1. Gå til trin 3 i behandle sag.	
<b>Postkonditioner:</b> None	

<b>Brugsmønster:</b> Behandle sag: Indhente oplysninger fra eksterne kilder
<b>ID:</b> 2.3
<b>Primær aktør:</b> Sagsbehandler
<b>Sekundære aktør:</b> Sundhedssystem
<b>Kort beskrivelse:</b> Indhenter oplysninger fra eksterne systemer.
<b>Prækonditioner:</b> Der skal være en <a href="#">udredningsformularen (Link)</a> der er påbegyndt.
<b>Alternativhændelsesforløb:</b>
1. Indtastning af oplysninger i relevante felter fra eksterne kilder i <a href="#">udredningsformularen (Link)</a> .
2. Formularen gemmes.
3. Sagen lukkes.
<b>Postkonditioner:</b> Formularen opdateret.

<b>Brugsmønster:</b> Behandle sag: Afgøre sagsbehandling
<b>ID:</b> 2.4
<b>Primær aktør:</b> Sagsbeandler
<b>Sekundære aktør:</b> Handleplan modul, Dagbog modul
<b>Kort beskrivelse:</b> Sagsbeandler registrerer afgørelse af pågældende sag og skriver et brev til borger omkring afgørelsen. Hvis sagen ikke er afvist, udfyldes en bestilling af social indsats.
<b>Prækonditioner:</b> Der er indhentet oplysninger fra eksterne kilder.
<b>Alternativhændelsesforløb:</b>
1. Hvis sagen ikke afvises. <ol style="list-style-type: none"><li>1.1. Udfylder <a href="#">formularen for bestilling af social indsats (Link)</a>.</li><li>1.2. Systemet sender oplysninger til handleplan modulet ift. færdiggørelse af handleplanen.</li><li>1.3. Systemet sender oplysninger til dagbog modulet ift. oprettelse af dagbog.</li></ol>
2. Der udfyldes et afgørelsесbrev.
3. Afgørelsесbrevet gemmes.
4. Sagen lukkes.
<b>Postkonditioner:</b> Sagen er opdateret. Der skal være udfyldt et afgørelsесbrev.

Brugsmønster: Find sag	
<b>ID:</b> 3	
<b>Primær aktør:</b> Sagsbehandler	
<b>Sekundære aktør:</b> None	
<b>Kort beskrivelse:</b> Skal kunne søge i sager og få en sag vist.	
<b>Prækonditioner:</b> Der skal være oprettet en eller flere sager.	
<b>Hovedhændelsesforløb:</b>  Starter når den primære aktør skal finde en sag. 1. Skal søge en sag på sagsnummer, CPR-nummer, eller navn. 2. Vise en liste over de sager som blev fundet. 3. Hvis en sag bliver valgt. 3.1. Vis den valgte sag. 3.1.1. Aktør kan behandle sag.	
<b>Postkonditioner:</b> None.	
<b>Alternative hændelsesforløb:</b> None.	

#### 5.2.5 Overordnet beskrivelse af supplerende krav.

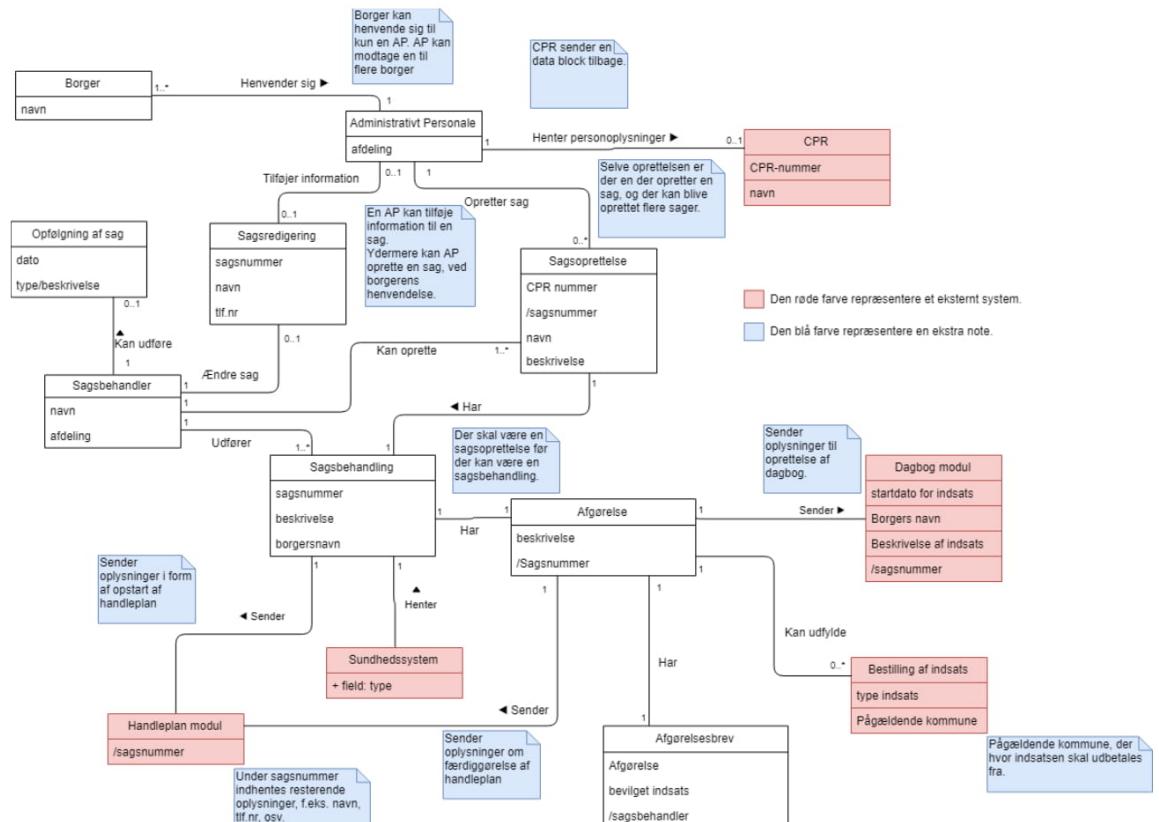
Herunder findes en liste over de supplerende krav (ikke-funktionelle krav) til sagsforløbet, der indeholder en kort beskrivelse, type og prioritet.

ID	Navn	Beskrivelse
01	Modularitet	Det skal være kompatibelt med andre moduler.
02	Sagsbehandlers adgang	En sagsbehandler tilhører en afdeling og kan kun se børger under samme afdeling.
03	Borgers sagsbeandler	En børger kan have flere sagsbehandlere fra forskellige afdelinger.
04	Borger	Der skal kunne være flere sager åbne på en børger.
05	Borger	Borgeren skal kunne få tildelt forskellige ydelser.
06	Sagsbeandler	Sagsbeandleren skal kunne åbne flere sager inden for samme afdeling.

07	Database	Databasen skal kunne håndtere forskellige borgere og deres forskellige sager indenfor samme afdeling og forskellige afdelinger.
08	Sagsbehandler	Skal udelukkende kunne se sager inden for deres afdeling.
09	Database	Databasen skal generere sagsnummer, samt udfylde andre genererede tal og strenge.
10	Sagsbeandler	Sagsbeandleren skal overholde kravene fra VUM
11	Eksternt system (CPR)	Der skal kunne hentes oplysninger fra personregisteret. (Fiktiv data)
12	Eksternt system (Sundhedssystem)	Der skal kunne hentes læge noter/lægeerklæring. (Fiktiv data)
13	Eksternt system (Handleplan modul)	Skal kunne sende data til handleplan. (data sendes til fiktivt modul)
14	Eksternt system (Dagbog modulet)	Skal sende data fra afgørelsen med de nødvendige oplysninger. (data sendes til fiktivt modul)

### 5.2.6 Domænemodel m. klassediagram og beskrivelse af domænet og klasserne.

I dette afsnit bliver der udarbejdet et klassediagram over domænemodellen med en overordnet beskrivelse af forretningsbegreberne (klasserne).



Figur 2 Klassediagram over domænemodellen

Figur 2 viser hele domænet i sagsforløbet. En sag starter ved at en borger henvender sig til den pågældende kommune. I kommunen finder man forskellige afdelinger, hvor hver afdeling har sit administrative personale. En borger oplyser sit fulde navn og giver samtykke til at der må laves CPR-opslag ift. personregisteret og herefter laves en sagsoprettelse. Hvis en sag mangler yderligere information, kan AP (administrativt personale) tilføje informationer, ved at redigere sagen.

Når en sag er oprettet, starter hele sagsudredningsforløbet, hvor en sagsbeandler kan foretage sagsbehandling. Under sagsbehandlingen, sender sagsbeandleren oplysninger til opstart af en handleplan. Som en del af sagsbehandlingen kan der tilføjes informationer fra læger i form af en lægeerklæring, hvis det er relevant for den pågældende sag. Sagsbehandling afsluttes med en

afgørelse, hvor der kan blive sendt oplysninger til oprettelse af en dagbog, samtidig med at der sendes oplysninger til handleplanen for færdiggørelse af denne. Der bestilles evt. indsats i form af ydelser. Til sidst bliver der sendt et afgørelsесbrev til borgeren. Senere kan sagsbehandleren lave en opfølgning af eksisterende sager.

**Definition af klasserne i figur 2:**

Navn (Symbol)	Borger.
Definition (Intention)	En borger repræsenterer en person der kan henvende sig til kommunen.
Beskrivelse/Eksempel (Extension)	Eksempel: En borger med navnet Jens Hansen henvender sig til kommunen for at få afklaret en sag.

Navn (Symbol)	Administrativt personale (AP)
Definition (Intention)	Administrativt personale repræsenterer de mennesker som er første led i afklaring af sag, da de tager i mod en borger.
Beskrivelse/Eksempel (Extension)	Eksempel: Administrativt personale (sekretær, sagsbeandler el.lign.) modtager en henvendelse fra en borger. Henvendelsen kan enten være en ny sag, som så bliver oprettet eller på en igangværende sag, som vil blive opdateret med nye oplysninger som borger har indleveret.

Navn (Symbol)	Sagsoprettelse.
Definition (Intention)	En sagsoprettelse repræsenterer oprettelse af en sag. Den indeholder oplysninger om: borgers CPRnr borgers navn sagsbeskrivelse sagsnummer
Beskrivelse/Eksempel (Extension)	Eksempel: En sag bliver oprettet af administrativt personale, som indtaster borgers navn, CPR-nummer og en beskrivelse af hvad sagen drejer sig om. Der vil blive genereret et sagsnummer af systemet, for at sikre det er unikt.

Navn (Symbol)	Sagsredigering.
Definition (Intention)	En sagsredigering repræsenterer tilføjelse af nye oplysninger på en sag fra borger. Indeholder oplysninger om: sagsnummer navn tlfnr
Beskrivelse/Eksempel (Extension)	Eksempel: Ved henvendelse fra borger kan administrativt personale søge efter en eksisterende sag hvis der findes en, kan nye oplysninger tilføjes.

Navn (Symbol)	Sagsbehandler
Definition (Intention)	En sagsbehandler repræsenterer en person som er ansat til at behandle sager i kommunen. Indeholder oplysninger om: navn afdeling

Navn (Symbol)	Opfølgning af sag.
Definition (Intention)	<p>Opfølgning af sag repræsenterer en opfølgning af en sag efter den er blevet afgjort.</p> <p>Indeholder oplysninger om:</p> <ul style="list-style-type: none"> <li>Dato for opfølgning</li> <li>Beskrivelse af opfølgning</li> <li>Resultat af opfølgning</li> </ul>
Beskrivelse/Eksempel (Extension)	<p>Eksempel:</p> <p>En sagsbehandler henvender sig til bosted for at indhente oplysninger om en borgers forløbet?</p>

Navn (Symbol)	Sagsbehandling
Definition (Intention)	<p>En sagsbehandling repræsenterer tilføjelse af nye oplysninger på en sag til brug i beslutningstagning.</p>
Beskrivelse/Eksempel (Extension)	<p>Eksempel:</p> <p>En sagsbeandler udfylder relevante formularer ud fra indhentede oplysninger fra læger/sundhedssystem.</p> <p>En sagsbeandler opretter en handleplan ud fra de oplysninger der er givet.</p>

Navn (Symbol)	Afgørelse
Definition (Intention)	<p>En afgørelse repræsenterer den afgørelse der er blevet taget i forbindelse med behandlingen af en given sag.</p> <p>Indeholder oplysninger om:</p> <ul style="list-style-type: none"> <li>Sagsnummer</li> <li>Afgørelse af sag</li> </ul>
Beskrivelse/Eksempel (Extension)	<p>Eksempel:</p> <p>En sagsbeandler har afgjort en sag. Der skal bestilles en social indsats og skrives et afgørelsесbrev</p>

Navn (Symbol)	Afgørelsесbrev
Definition (Intention)	<p>Et afgørelsесbrev repræsenterer det brev der skal sendes til borger som fortæller om den afgørelse sagsbehandler er nået frem til.</p> <p>Indeholder oplysninger om:</p> <ul style="list-style-type: none"><li>Afgørelse af sag</li><li>bevilget indsats (valgfrit)</li><li>sagsbehandler</li></ul>
Beskrivelse/Eksempel (Extension)	<p>Eksempel:</p> <p>En sagsbehandler skriver et afgørelsесbrev, som bliver sendt til borger som pågældende sag omhandler. Det indeholder afgørelsen, navn på sagsbeandler, vejledning til klage over afgørelsen samt, hvis tildelt, beskrivelse af indsats.</p>

Navn (Symbol)	CPR
Definition (Intention)	<p>CPR repræsenterer et eksternt system, hvor der hentes personlige oplysninger omkring en borger.</p> <p>Sender oplysninger tilbage ift.</p> <p>CPR-nummer</p> <p>Fulde navn</p>
Beskrivelse/Eksempel (Extension)	<p>Eksempel:</p> <p>Det administrative personale, får et samtykke fra en borger omkring at der må hentes oplysninger fra personregisteret, hvor informationen tilføjes i sagen.</p>

Navn (Symbol)	Handleplan
Definition (Intention)	Handleplanen repræsenterer et eksternt modul i Sensus, hvor der under sagsbehandlingen sendes oplysninger til oprettelse af en handleplan. Hvor der i sagsafgørelsen bliver sendt oplysninger til færdiggørelse af planen.
Beskrivelse/Eksempel (Extension)	Eksempel: Under sagsbehandlingen udfylder sagsbeandleren de rette paragrafer i indhentning af oplysninger fra borgeren, hvor der bliver udfyldt at en handleplan skal oprettes. I afgørelsen bliver handleplanen færdiggjort.

Navn (Symbol)	Dagbog
Definition (Intention)	Dagbog repræsenterer et eksternt modul i Sensus, hvor der under sagsafgørelsen kan blive sendt oplysninger til dagbog modulet om at oprette en dagbog.
Beskrivelse/Eksempel (Extension)	Modul som skal simuleres.

Navn (Symbol)	Sundhedssystem
Definition (Intention)	Sundhedssystemet er et eksternt system, hvor en sagsbeandler kan hente lægeerklæring eller andet relevant information fra fagligt personale.
Beskrivelse/Eksempel (Extension)	Eksternt system, som skal simuleres.

Navn (Symbol)	Bestilling af indsats
Definition (Intention)	Bestilling af indsats er et eksternt system, hvor informationer fra bestilling af indsats formular sendes til det eksterne system omkring oprettelse af evt. ydelser.
Beskrivelse/Eksempel (Extension)	Modul som skal simuleres.

## 6. Kritiske risici

For at skabe et overblik over risici er der lavet et risikoskema. Dette gør det mere synligt for gruppen hvad for nogle risici der er og hvad for en effekt de kan have. De er delt ind efter type, sandsynlighed og effekt.

Der er seks forskellige typer:

- Teknologi: risikoen har med teknologi at gøre.
- Folk: risikoen har med gruppen at gøre.
- Organisation: risikoen har med styringen af projektet eller produktet at gøre.
- Værktøj: risikoen har med de værktøjer der bliver brugt at gøre.
- Krav: risikoen har noget med de krav der er stillet op at gøre.
- Estimater: risikoen har med estimater at gøre.

Sandsynlighed og effekt er opdelt i tre trin (lav, medium og stor). Sandsynligheden repræsenterer risikoen for at det sker, mens effekten er hvor stor en betydning udløsning af risikoen har for gruppen og projektet.

Nedenfor er de risicis vi har fundet og hvilke konsekvenser de har for projektet.

Risici	Betydning	Beskrivelse	Risici type	Sandsynlighed	Effekt
Projektstyring	Projekt	Forkert management kan gøre gruppen mister overblik	Organisation	Medium	Stor
Gruppe frafald	Projekt	Der kan ske noget uventet så et gruppemedlem ikke kan udføre deres arbejde.	Folk	Lav	Stor
Projekt underestimeret	Projekt og produkt	Systemets størrelse er undervurderet så der er mangel på tid	Estimat	Stor	Lav
Systemkrav ændring	Produkt	I tilfælde af at kravene til systemet ændrer	Krav	Stor	Medium
EG Team Online	Projekt og produkt	Ændring af case	Organisation	Lav	Stor

Analyse af VUM	Projekt	Produktet lever ikke op til de reelle krav	Krav	Medium	Lav
Produkt komplikation	Projekt og produkt	Uventede fejl som ikke kan findes en løsning på	Teknologi	Medium	Medium
VUM	Projekt	Ændring af VUM's metoder kan føre til at projektet skal laves om	Krav	Lav	Lav
Effektivitet	Projekt	Hvis folk mangler drivet til at få lavet noget seriøst	Folk	Høj	Medium
Viden	Produkt	Hvis gruppen ikke har nok ud af undervisningen og kan ikke lave et godt produkt	Folk	Lav	Høj
Github	Projekt	I tilfælde af at github går ned kan vi miste vores data	Værktøj	Lav	Høj
Data afgrænsning	Projekt	For ikke løst kravet om data afgrænsning og dermed ikke svaret på vores problemformulering	Krav	Lav	Medium
Forsinkelser	Projekt og produkt	Milepæle bliver ikke overholdt	Estimat	Lav	Medium
Manglende erfaring	Projekt og produkt	Mangel på erfaring I gruppen.	Folk, organisation	Høj	Lav

## 7. Prioritering

I denne sektion bliver der vist foreløbige lister over krav. Kravene er vurderet ud fra nytte, risiko og prioritet, disse vurderinger bliver noteret som Lav, Middel eller Høj.

Nytten er det punkt hvor man beskriver om et krav er kritisk for implementeringen eller ej.

Risiko er det punkt hvor man beskriver risici forbundet med implementering af krav.

Prioriteringen er vigtigheden af kravet for systemet.

Krav #	Kategori	Kravsbeskrivelse	Nytte	Risiko	Prioritet
01	Sensum - Login	Bruger kan logge ind som Sagsbehandler, afdelingsleder, sekretær eller administrator.	Middel	Lav	Lav
02	Sensum - log ud	Som sagsbehandler, afdelingsleder, sekretær eller administrator skal man kunne logge ud.	Middel	Lav	Lav
03	Sensum - Moduler	Ved login, skal man have mulighed for at vælge mellem forskellige moduler (Simulering).	Middel	Lav	Lav
04	Sagsforløb - Sagsoprettelse	Sagsbehandleren og det administrative personale skal kunne oprette en sag	Høj	Middel	Høj
05	Sagsforløb - Find sag	En sagsbehandler og afdelingsleder skal kunne søge efter en sag via et sagsnummer, fulde navn eller CPR-nummer.	Middel	Middel	Middel
06	Sagsforløb - Sagsbehandling	Under sagsbehandlingen skal en sagsbeandler have mulighed for at kunne indhente oplysninger fra den pågældende borger og vælge rette paragraf ud fra en formular ved videre sagsbehandling.	Høj	Høj	Høj
07	Sagsforløb - Sagsafslutning	Sagsbeandleren skal kunne lukke en sag, når den given borger er afsluttet.	Høj	Høj	Høj

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019

08	Sagsforløb - Opfølgning af sag	En sagsbeandler skal kunne følge op på en sag og der skal kunne udfyldes type af opfølgning og datoen opfølgningen er sket.	Middel	Lav	Lav
09	Sagsforløb - Sagsredigering	En sag skal kunne redigeres af enten en sagsbeandler eller en sekretær.	Høj	Middel	Høj
10	Sagsforløb - Sagsafgørelsen	En sag skal kunne afgøres, hvor nødvendige informationer sendes til rette moduler for oprettelse af nødvendige handlinger ifølge med en borgeres sagsstatus.	Høj	Middel	Høj

Ikke funktionelle krav					
#	Kategori	Kravsbeskrivelse	Nytte	Risiko	Prioritet
01	Modularitet	Det skal være nemt at kommunikere med andre moduler.	Middel	Lav	Middel
02	Sagsbehandlers adgang	En sagsbeandler tilhører en afdeling og kan kun se sager under samme afdeling.	Høj	Høj	Høj
03	Borgers sagsbeandler	En borgers kan have flere sagsbeandler fra forskellige afdelinger.	Middel	Middel	Lav
04	Borger	Der skal kunne være flere sager åbne på en borger.	Middel	Middel	Lav
05	Borger	Borgeren skal kunne få tildelt forskellige ydelser.	Middel	Lav	Lav
06	Sagsbeandler	Sagsbeandleren skal kunne åbne flere sager inden for samme afdeling.	Middel	Lav	Lav
07	Data	Data skal kunne håndtere forskellige borgere og deres forskellige sager indenfor samme afdeling og forskellige afdelinger.	Høj	Høj	Høj

08	Data	Der skal generere sagsnummer, samt udfylde andre genererede tal og strenge.	Høj	Høj	Høj
09	Eksternt system (CPR)	Der skal kunne hentes oplysninger fra personregisteret. (Fiktiv data)	Høj	Middel	Middel
10	Eksternt system (Sundhedssystem)	Der skal kunne hentes læge noter/lægeerklæring. (Fiktiv data)	Høj	Middel	Middel
11	Eksternt system (Handleplan modul)	Skal kunne sende data til handleplan. (data sendes til fiktivt modul)	Høj	Middel	Lav
12	Eksternt system (Dagbog modulet)	Skal sende data fra afgørelsen med de nødvendige oplysninger. (data sendes til fiktivt modul)	Høj	Middel	Lav

## 8. Metode i elaborationsfasen

### 8.1. UP (Unified Process)<sup>5</sup>

#### 8.1.1. Hvad er UP

UP er en softwareudviklingsproces, der definerer ”hvem, hvad, hvornår og hvordan”, **hvem** beskriver rollerne i projektet som f.eks. en person eller et team, **hvad** beskriver enhederne som bliver udført af en rolle eller en metode i projektet, **hvornår** er en sekvens af relationer som bringer mening til projektet og **hvorudan** beskriver hvem der udfører hvad og hvornår de gør det. UP er en generisk softwareudviklingsproces der skal integreres i en organisation og derefter i hvert enkelt projekt. Integreringen af UP indebærer at man definerer og inkorporerer interne standarder, dokumentskabeloner, værktøjer til kompilering, management konfiguration osv., databaser, herunder skal der være bug og projekt tracking, samt livscyklusændringer som f.eks. mere sofistikeret kvalitetskontrol af sikkerhedskritiske systemer.

UP bliver drevet af 3 faktorer:

- Krav og risikodreven.
- Arkitektur-centreret.
- Iterativ og trinvis.

---

<sup>5</sup> Arlow, UML 2 and the Unified Process, kap. 2

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019

Som det ses i bilag 1, er UP delt op i 5 kerne workflows som er krav, analyse, design, implementering og test. Hver af disse kerne-flows bliver brugt forskelligt i de fire faser inception, elaboration, construction og transition. Kravene beskriver hvad systemet skal gøre. I analyseworkflowet struktureres kravene og i designworkflowet bliver kravene opdelt efter systemarkitekturen. Derefter udvikles systemet i implementeringsworkflowet, hvorefter der i testworkflowet testes om systemet virker som planlagt.

Hver af de fire faser er delt op i iterationer, hvilket gøre det mere overskueligt. Hovedpointerne for hver iteration er: planlægning, analyse og design, integration og test, intern og ekstern udgivelse.

#### 8.1.2. Elaborationsfasen

Der er 6 overordnede mål for elaborationsfasen:

- At lave en eksekverbar arkitektonisk grundlinje.
- Raffinere risikovurderingen.
- At definerer kvalitetsattributter, såsom fejl opdagelsessatser, acceptable defektdensiteter osv..
- Identificer brugsmønstre til 80% af de funktionelle krav.
- Skabe en detaljeret plan for constructionsfasen.
- Lave en estimering af resurser, tid, udstyr, mandskab og omkostninger.

I denne fase bygger man et konkret og eksekverbart system ud fra den specifikke arkitektur, i stedet for en prototype. Der fokus på kerne workflowsene, men der skal ligges mest vægt på krav samt analyse og design. Implementeringen bliver først vigtig henimod slutningen af fasen, når den eksekverbare arkitektoniske grundlinje bliver udarbejdet. De næste faser bygger videre på det som er blevet lavet i elaborationsfasen, derfor kan dette også være den mest kritiske fase at arbejde i.

## 8.2. Scrum<sup>6</sup>

Scrum består af flere iterationer, hvor en iteration bliver kaldt et sprint. Et sprint forløb varer generelt to til fire uger. Her bliver der arbejdet med funktionalitet, som skal skrives i et forståeligt sprog, som alle interesserter kan forstå. Dette sker i brugerhistorier, som bliver lavet så man kan tale med kunden for at undgå at man laver noget unødvendigt. De forskellige brugerhistorier bliver samlet og deres prioritet vurderes. Den højeste prioritet vil som regel blive valgt som et mål for et sprint.

Når sprintet starter, vil målet blive delt ud i mindre opgaver, som bliver lavet ift. den tid sprintet tager. Under et sprint vil et team som regel mødes daglig. Mødet skulle helst ikke varer mere end 15 minutter og alle deltagere skal svare på tre spørgsmål:

- Hvad har personen nået siden sidste møde?
- Hvad vil personen nå til næste møde?
- Hvilke problemer er der på vejen til personens mål?

Alle de forskellige opgaver bliver skrevet ned og sat op på et board. Når opgaven er færdig, bliver den flyttet hen under udført. Når hele sprintet er slut, bliver der holdt et møde, hvor man viser kunden hvad der er blevet udarbejdet. Efter dette møde holder teamet, der har lavet sprintet et nyt møde og evaluerer hvad de har fået udarbejdet.

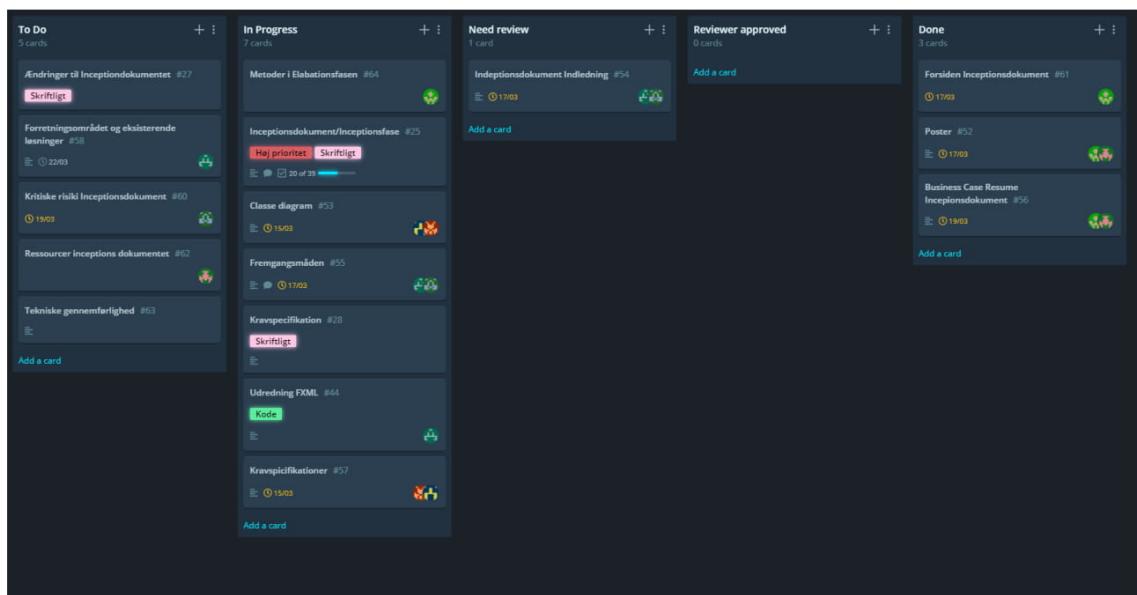
Gruppen vil bruge dele af scrum til iterationerne, samt lave nogle mindre sprints på en uge eller to, hvor der vil blive set på funktionaliteten.

---

<sup>6</sup> [http://cs.au.dk/fileadmin/site\\_files/cs/AA\\_pdf/Evaluering\\_af\\_tilpassede\\_SCRUM\\_metoder\\_-\\_report\\_final.pdf](http://cs.au.dk/fileadmin/site_files/cs/AA_pdf/Evaluering_af_tilpassede_SCRUM_metoder_-_report_final.pdf) s 7-12

### 8.3. Kanban<sup>7</sup>

Kanban betyder visual card på japansk. Det er en metode der kan stå for sig selv, men bliver ofte brugt sammen med Scrum. Kanban er i sig selv en mindre formel metode end Scrum. Det bliver brugt til at holde styr på hvilke opgaver der skal laves og hvem der arbejder på opgaverne gennem hele projektet. Dette sker vha. forskellige kort. Kanban metoden har flere steder hvor et kort kan være placeret. Ofte vil det være noget som skal laves, er i gang eller er færdigt.



Figur 3 Dette er et visuelt board, der viser gruppens Kanban

I figuren vises border der består af flere koloner. Der kan være så mange som det er nødvendigt for projektet. Målet med kanban bordet er at have så få processer så muligt under kolonnen igangværende (in Progress), så de forskellige personer kan fokuser på det de er i gang med.

<sup>7</sup> [http://cs.au.dk/fileadmin/site\\_files/cs/AA\\_pdf/Evaluering\\_af\\_tilpassede\\_SCRUM\\_metoder\\_-\\_report\\_final.pdf](http://cs.au.dk/fileadmin/site_files/cs/AA_pdf/Evaluering_af_tilpassede_SCRUM_metoder_-_report_final.pdf)s 7-12

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019

#### 8.4. Par produktion

Der er gjort gode erfaringer i inceptionsfasen med at arbejde i makkerpar. Dette vil fortsætte i elaborationsfasen og der vil blive arbejdet videre med at optimere processen.

Når en opgave uddelegeres, tilføjes to gruppemedlemmer til denne. Én af opgavens ansvarlige vælges midlertidigt som skribent og noterer det ønskede indhold i form af det umiddelbare svar. Den anden ansvarlige støtter med inputs som skal understøtte svaret, retter skribentens fejl og dokumentere tankerne bag de valgte svar. Skribent-rolen kan gå på tur, hvis makkerparret ønsker dette. Processen foregår som en løbende diskussion mellem de involverede, hvor opgavens emne og materialer bliver analyseret, vurderet og bedst muligt konkluderet.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02  
Aflævering:  
22-03/2019

## 9. Resurser

Gruppen har afsat to dage hver uge, som skal bruges på gruppearbejde. Der skal arbejdes hver tirsdag fra 8-14 og fredag fra 12-14.

Yderligere kan udgående blive suppleret med ekstra gruppearbejde, planlagt op til 14 dage frem, som er obligatorisk. Dette vil ske hvis der er aktiviteter som er blevet forsinket.

Følgende skema indeholder gruppens arbejdsplan for fremtidigt arbejde:

## Semester projekt gruppe 02



## 10. Konklusion

Under afsnit 5.2.1 er der blevet beskrevet hvilke aktører der forventes at benytte vores produkt. Disse aktørers interaktion samt analyse af projektcasen, er grundlaget for udarbejdelsen af brugsmønstre, som den overordnede kravspecifikation bygger på. Den overordnede brugsmønstermodel er udarbejdet ud fra brugsmønstre og aktører.

Ud fra information omkring sagsudredning er der blevet skabt en domænemodel, som viser hvordan projektforløbene kan fortolkes.

Metoderne som fremadrettet vil blive brugt i elaborationsfasen er unified proces, scrum, KanBan og parproduktion.

Ud fra en analyse af projektrammeplanen er der blevet udarbejdet en tidsplan. Projektets resurser er blevet prioriteret ud fra den forventede arbejdsindsats, der skal til for at færdiggøre projektaktiviteterne.

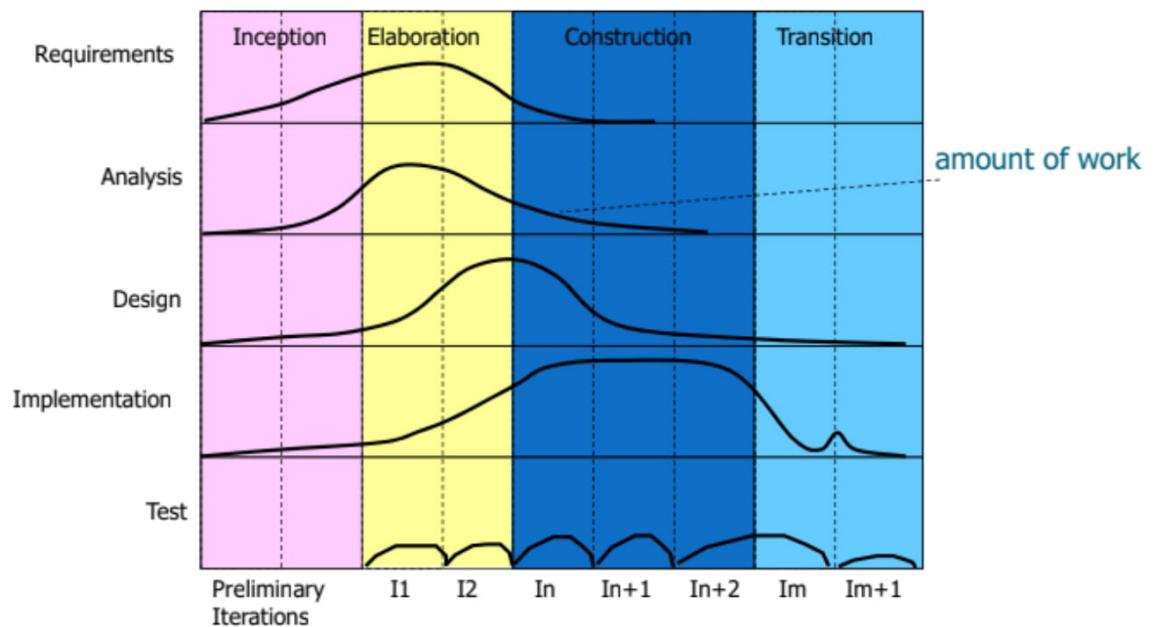
Der er udarbejdet en tabel over projektets kritiske risici. Denne er lavet ud fra en risikovurdering af interne og eksterne faktorer der kan påvirke projektet.

Der er lavet en prioriteringen af krav, som beskriver betydning de kan have for projektet, samt deres nytte. Ud fra problemformuleringen er der blevet lagt vægt på krav der omhandler dataafgrænsningen og sagsudredningen.

Ud fra resultaterne er der blevet dannet et overblik over forretningsområdet, samt en foreløbig kravsspecifikation for det valgte modul.

## 11. Bilag

### Bilag 1:



Figur 4 Dette bilag viser hvordan arbejdflowet er fordelt i UP

### 11.1. Logbog

<https://github.com/Diplom-Software2018/Anden-semester/wiki>

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019

## 11.2. OL Cases



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

# Organisationsanalyse af EG Team Online

ApS

Udarbejdet af:

Aleksander Grzegorz Duszkiewicz

A handwritten signature in black ink, appearing to read "Duszkiewicz".

Aleksander Henriksen

A handwritten signature in black ink, appearing to read "Henriksen".

Per Terp Larsen

A handwritten signature in black ink, appearing to read "Per Terp Larsen".

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

## Indholdsfortegnelse

Indledning .....	3
Hovedtekst .....	3
PESTEL-Analyse .....	3
Politiske faktorer:.....	3
Økonomiske forhold: .....	4
Social- og Kulturforhold:.....	4
Teknologiske faktorer:.....	5
Lovmæssige forhold:.....	5
Poters Five Forces-Analyse .....	6
Trussel af ny indtrængere (Threat of Entry) .....	6
Kunde bytte omkostninger (Customer switching costs) .....	6
Etableret fordele uafhængig af størrelse (Incumbency advantages independent of size) .....	6
Forhandlingskraft fra leverandør (The power of suppliers).....	6
Truslen af erstatnings produkter eller services (The Threat of substitutes) .....	6
Rivalisering blandt eksisterende konkurrenter (The Threat of substitutes) .....	7
Forhandlingskraft fra køber (The power of buyers).....	7
Diskussion .....	7
Konklusion .....	8
Litteraturliste .....	9
Bilag .....	9

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

## Indledning

Gennem denne analyse af EG Team Online ApS, vil der blive set på det generelle og specifikke miljø som påvirker dem indenfor deres branche. Der vil igennem en PESTEL-analyse blive forklaret hvilken faktorer der påvirker det generelle miljø. Der er også udarbejdet en Porters Five Forces-analyse som forklarer om den konkurrence som er indenfor den virksomhedensbranche, og hvilken betydning det kan have på organisationen.

## Hovedtekst

### PESTEL-Analyse

#### Politiske faktorer:

Den danske regering lavet en ny digitaliseringssstrategi i 2016, som gælder frem til 2020, dette er den femte siden 2000. Denne strategi sætter et større fokus på at staten, regionerne og kommunerne skal arbejde sammen på tværs af de forskellige instanser, samt øge væksten inde for digitaliseringen. (Kilde: [Digitaliseringsstrategien 2016-2020 \(Link\)](#))

Selvom der har været svingninger i de ideologiske overbevisninger i den danske regering mellem 2000 og 2019, har der stadig været en handleplan for digitaliseringssområdet som har lagt vægt på en vækst inde for området. Danmark ligger i toppen indenfor digitaliseringen i EU, men det ændrer ikke på at der stadig kan gøres meget indenfor netop dette område.

I forhold til det område hvor EG Team Online ApS arbejder, skal der gøres meget for at det er muligt at højne samarbejdet så digitaliseringssområdet i kommunerne bliver mere flydende og tilpasser sig den politiske strategi.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

#### Økonomiske forhold:

Den danske regering har vedtaget nogle initiativer i digitalisingsstrategien for 2016 - 2020, der nævner at offentlig digitalisering skal give gode vilkår for vækst. (Kilde: [Digitalisingsstrategien 2016 - 2020 \(Link\)](#))

Ydermere byder strategien ind på Danmarks digitale vækst, hvor der bliver afsat ca. 125 mio. kr. årligt i perioden 2019 - 2025, hvor et af ledende af væksten er at styrke det digitale vækstmiljø. (Kilde: [Erhvervs- og Iværksætterinitiativer \(Link\)](#))

Det giver gode muligheder for virksomheder at kunne drive og bidrage til regeringens målsætning af fokusområderne, heriblandt digitalisering af det offentlige.

Det betyder at fremtiden for EG Team Online ApS, har på større sigt muligheder for at kunne etablere sig yderligere i den offentlige sektor, og idet muligheder for at udvikle nye produkter eller forbedre nuværende.

#### Social- og Kulturforhold:

I Odense sættes der rigtig høj fokus på at udvikle og omdanne Odense til en videns by. Det gør at mange studerende og generelt borgere flytter til byen for at enten at udvikle sig eller arbejde. En helhedsplan udarbejdet af Odense Kommune skal sikre udbygning af campusområdet, ift. udbygning af forskerparkens afdelinger og placering af et nyt universitetshospital m.m.

Omkring Vision og planstrategien 2008, vil man gøre Odense til en bæredygtig by. Denne bæredygtig udvikling skal sikre at gøre Odense til et attraktivt sted at leve, ikke kun for nuværende men også kommende generationer. (Kilde: [Byudviklingsplaner, Odense Kommune \(Link\)](#))

Denne udviklingsproces for Odense vil sikre virksomheder arbejdere der både har en fornuftig levestandard, og adgang til at udvikle sig fagligt.

Det giver store fordele for EG Team Online ApS at have en store pulje af glade og uddannede arbejdere der kan bidrage til udviklingen af virksomhedens produkter der samtidig vil styrke den på fremtidigt sigt.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

#### Teknologiske faktorer:

Danmark er et førende digitaliseret land i forhold til andre EU-lande i Europa. Den øgede digitalisering har betydet en større udvikling i forhold til række teknologier som f.eks. kunstig intelligens, 3D-print, dele- og platforms økonomi og m.m. Denne udvikling indbefatter potentiale for teknologiske fremskridt i fremtiden. (Kilde: [Hovedtræk i den teknologiske udvikling, Regeringen \(Link\)](#))

Danmarks førende digitalisering giver fordele for virksomheder i forhold udvikling af flere innovative produkter.

Det giver fordele i fremtiden for EG Team Online ApS i forhold til at udvikle og skabe nye innovative produkter indenfor deres område.

#### Lovmæssige forhold:

Hvis man som en organisation vil udbyde et stykke software til den offentlige sektor er der en række regler som skal overholdes. Den offentlige myndighed har 2 måder de kan starte en udbudsproces på, den ene er ved at lave en prækvalificering af virksomhederne som har til ønske at byde, i prækvalifikationsfasen bliver virksomhederne som har interesse i udbuddet bedt om at sig til kende. Det til bliver der også stillet nogle minimumskrav til hvad virksomheden skal kunne for at blive valgt, disse krav kan f.eks. være stillet til teknisk udstyr.

Når denne proces er slut, begynder de interesserede/udvalgte virksomheder at rekvirere udbudsmateriale og skriver et tilbud som et modsvar til det tilgængelige materiale. (Kilde: [Offentligt udbud \(Link\)](#))

Denne udvælgelses proces er en fordel for EG Team Online ApS, da det er et datterselskab af en større koncern. Dermed har de en mulighed for at få de ressourcer, som er nødvendige for at kunne leve op til de krav som de offentlige instanser sætter som i prækvalifikationsfasen.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

## Poters Five Forces-Analyse

Efter virksomhedsmødet med EG Team Online, har gruppen fået mere information i forhold til de kræfter som påvirker virksomheden. (Kilde: Virksomheds møde med EG Team Online, udviklingschef, Hans Ole Andersen)

### Trussel af ny indtrængere (Threat of Entry)

#### Kunde bytte omkostninger (Customer switching costs)

På selve mødet blev der spurgt om hvor nemt det var for en kunde at kunne skifte til en anden leverandør. Dertil lød svaret at det er ret nemt, men det koster derimod rigtig mange penge i forhold til den data og de oplysninger der skal overføres og opbevares. Så er det dyrt for kunden at skifte.

#### Etableret fordele uafhængig af størrelse (Incumbency advantages independent of size)

Det er rimelig svært for en anden virksomhed at kunne etablere sig i det specifikke område som EG-Team Online opererer i. På mødet blev der forklaret at virksomheder der ønsker at etablere sig i det specifikke marked, skal have nogle gode referencer, skal den have gode referencer fra tidligere branche skift.

#### Forhandlingskraft fra leverandør (The power of suppliers)

EG Team Online har ikke mange leverandører. Den eneste leverandør nævnt til mødet var en hosting virksomhed, der kan hoste deres database server. Sensum tilbyder dog brug af trejdepart udbyder. Dermed betaler kommunerne for leverandørerne i forhold til arkivering af data, indhentning af CPR-numre osv. Derfor er forhandlingskraften ikke høj ift. leverandøren, fordi der er så mange udbyder, derfor har leverandøren ikke stor magt.

#### Truslen af erstatnings produkter eller services (The Threat of substitutes)

Et substituerende produkt kunne være at bruge papir i kommunen. Det betyder at hvis kommunen begynder at udfylde en pågældende sag i papir i stedet for at anvende systemet, vil det være en risiko for virksomheden. Der er dog nævnt i digitaliseringssplanen at det offentlige skal digitaliseres

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

mere ift. at spare på papir. Dermed betyder det at kommunerne har muligheden for at anvende papir, men digitaliseringstrategien forhindrer kommunerne i at anvende papir.

#### Rivalisering blandt eksisterende konkurrenter (The Threat of substitutes)

EG Team Online's største konkurrent er virksomheden KMD, der har mange produkter der kan konkurrere med deres Sensum system, som f.eks. NEXUS systemet. En yderligere konkurrent kunne være Systamatic der også udvikler samme type software. Dermed er rivaliseringen høj i forhold til udbuddene, fordi udbuddet ligger for hele EU, og dermed kan rigtig mange virksomheder byde ind.

#### Forhandlingskraft fra køber (The power of buyers)

EG Team Onlines kunder består primært af regioner, kommuner og private bosteder. Derfor er der få kunder indenfor det pågældende marked. Dermed ligger forhandlingskræften høj hos kunderne, hvis deres system får negative bemærkninger, kan det påvirke deres fremtidige køb fra private bosteder.

## Diskussion

Ud fra PESTEL analysens politiske forhold, kan der ses at de ændres ca. hvert fjerde år, hvis vi ser på hvad der er sket inden for digitaliseringsområdet indtil nu. Det vil sige at EG Team Online skal være klar til at ændre de krav de skal leve op til. Der blev samtidig på mødet forklaret at det er de politiske forhold der er den største faktor i virksomheden, da loven kan ændres hurtigt, og dermed skal systemet ændres for at overholde disse love.

Virksomheden ligger i Odense, som er en by under stor udvikling. Det betyder at de burde prøve at være på forkant med hvad de skal have ifølge loven. Danmark er på forhånd længere fremme end andre EU-lande i forhold til digitalisering. Ved at virksomheden befinder sig i en udviklings by kan det hjælpe dem med nye ideer i forhold til deres produkt. Alt dette gør, at det er bedre for dem med en organisk organisation i forhold til PESTEL analysen.

I porters five forces ses der på truslen fra indtrængere. Denne trussel er ikke stor i dette tilfælde, da det koster en del at skifte systemer og kunden skal betale for overføring af data, hvilket er en stor investering fra kundens side af. Dette lægger op til at EG team online kan køre en forholdsvis mekanisk struktur, da strukturen rundt omkring i miljøet er stabil.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

Leverandørerne har ikke en stor betydning for hvordan EG Team Online arbejder, hvilket giver dem en lav forhandlingskraft, dette gør at en mere mekanisk struktur er en fordel, da der ikke kan komme ændringer ind fra leverandørerne, fordi der er så mange forskellige leverandører som virksomheden kan vælge imellem.

Der findes andre virksomheder med et produkt der går efter de samme udbud som EG Team Online. Disse er allerede på markedet også fra firmaer som har været på markedet i lang tid og har et navn indenfor området. Dette lægger op til en mere organisk struktur.

Forhandlingskraft fra køber er stor, da det primært er kommunerne de sælger produktet til og det er dem der bestemmer hvad for et program de ønsker. Dette lægger op til at EG Team Online har fordel af en organisk struktur.

Der er også udarbejdet et diagram der konkluderer på Porters Five Forces, se bilag 1.

## Konklusion

Ud fra alle disse ovenstående analyser er der nogle områder som ligger op til en mekanisk struktur som at kommunen lægger deres udbud ud i noget tid i forvejen og at truslen for at nye konkurrenter kommer ind på markedet. Der er også at det er let at skifte leverandør.

Med det sagt er det en organisk struktur som resten ligger op til. Både hvilke krav der bliver stillet til deres produkt fra købers side, samt hele PESTEL analysen, hvilket gør at vi vil formode at EG Team Online har en organisk struktur.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

SI2-OL

Aleksander Henriksen

Per Terp Larsen

11. marts 2019

## Litteraturliste

Udbudsportalen.dk – Oversigt over hvordan man udarbejder et tilbud i kommunerne:

<https://tilbud.udbudsportalen.dk/artikler/at-handle-med-det-offentlige/>  
[2019, 25-02-2019]

Regeringen.dk – Faktaark om hovedtræk i den teknologisk udvikling:

<https://www.regeringen.dk/media/3666/faktaark-om-hovedtr%C3%A3k-i-den-teknologiske-udvikling-pdf.pdf>  
[2019, 25-02-2019]

Odense.dk – Politisk vision for byudviklingsplanen:

<https://www.odense.dk/politik/politikker-og-visioner/byudviklingsplaner/planer-for-bydele/helhedsplan-for-omraadet-ved-syddansk-universitet>  
[2019, 25-02-2019]

Skm.dk – Aftale mellem regeringen om erhvers- og iværksætterinitiativer:

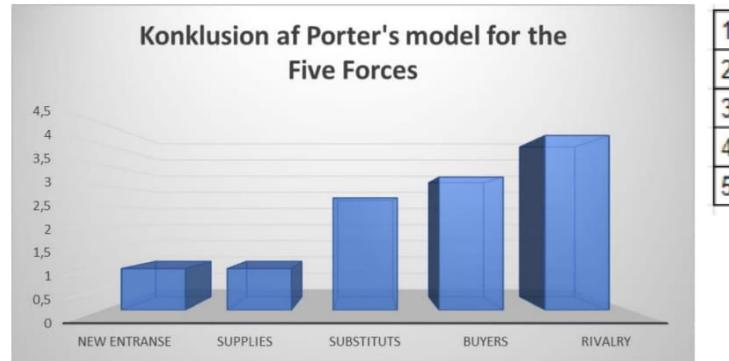
<http://www.skm.dk/media/1542156/aftaletekst.pdf>  
[2019, 07-03-2019]

Digst.dk – Digitaliseringstrategien 2016 – 2020:

<https://digst.dk/strategier/digitaliseringstrategien/om-digitaliseringstrategien-2016-2020/>  
[2019, 03-03-2019]

## Bilag

### Bilag 1:



Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet

Aleksander Grzegorz Duszkiewicz

Aleksander Henriksen

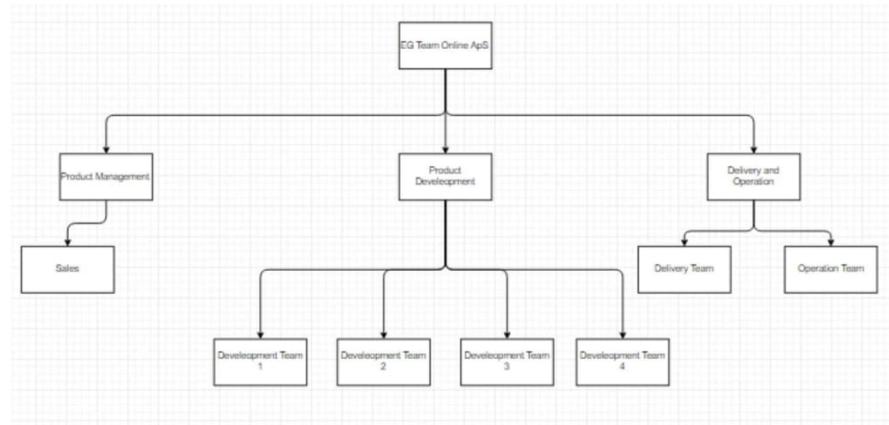
Per Terp Larsen

SI2-OL

11. marts 2019

**Bilag 2:**

Organisations Struktur over EG Team Online ApS



Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019

# Analyse af datterselskab EG Team Online ApS

TÆLLENDE AKTIVITET NR 1 SI2-OL

SKREVET AF:  
ASLAK MANGINO  
MATHIAS JAKOBSEN  
STEFFEN VITTTEN

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



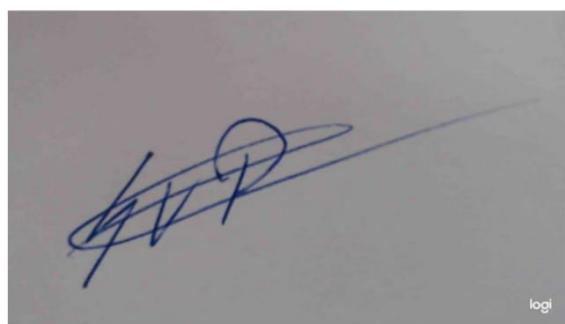
Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

A handwritten signature in black ink, appearing to read 'Aslak Mangino Nicolajsen'.

Aslak Mangino Nicolajsen

A handwritten signature in black ink, appearing to read 'Mathias Jakobsen'.

Mathias Jakobsen



Steffen Vitten Pedersen

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet  
Gruppe 02 A

Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

## INDHOLD

Tællende aktivitet nr 1 sII2-ol.....	0
Indledning .....	3
problemstilling .....	3
Fremgangsmåde .....	<b>Fejl! Bogmærke er ikke defineret.</b>
Resultater .....	3
Pestel analyse .....	3
Politiske forhold.....	3
Økonomiske forhold .....	3
Social og kulturforhold .....	4
Teknologiske faktorer .....	4
Lovmæssige forhold.....	4
Porter's 5 Forces .....	4
Truslen fra potentielle indtrængere .....	4
Substituerende produkter .....	5
Leverandørers forhandlingskraft.....	5
Kunders forhandlingskraft .....	5
Konkurrence situation i branchen .....	5
Diskussion.....	6
The contingency approach .....	6
Konklusion.....	7

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

## INDLEDNING

### PROBLEMSTILLING

Denne rapport vedrører fagområdet organisation og ledelse for virksomheden EG Team Online. Rapporten er bygget på at der skal udarbejdes en virksomhedsanalyse. Rapportens hovedformål er at besvare følgende spørgsmål:

- Hvilke kræfter påvirker organisationen for henholdsvis det generelle og specifikke miljø?
- Hvordan burde EG Team Online organisatoriske design fremstå?

## RESULTATER

### PESTEL ANALYSE

#### POLITISKE FORHOLD

Den danske regering har udarbejdet en digitalisingsstrategi for 2016 -2020, som er den femte digitalisingsstrategi siden 2000. Strategierne har stort fokus på at staten, regionerne og kommunerne skal øge deres samarbejde. (Digitaliseringsstyrelsen, 2016)

Selvom de ideologiske overbevisninger har skiftet med hver ny regering mellem 2000 og 2019, har der altid forelagt en handleplan for digitalisering med fokus på digital vækst. Yderligere er Danmark, ifølge Business Insider, vurderet til at være blandt verdens fem bedste demokratier, som alle har det til fælles at de har et funktionelt og stabilt regeringsapparat. (Ma, 2018)

Den politiske enighed angående digitalisering indikerer et stabilt voksende IT-marked med en stor frihedsgrad i forhold til IT-udvikling. Det stærke demokrati peger på en stabil fremtid uden risiko for et paradigmeskift angående statsapparatet.

#### ØKONOMISKE FORHOLD

Initiativerne i digitaliseringsstrategien for 2016-2020 sigter efter at den offentlige digitalisering skal give gode vilkår for vækst. (Digitaliseringsstyrelsen, 2016)

I *aftalen om erhvervs- og iværksetterinitiativer*, beskrives det hvordan den danske regering har planer om at hjælpe med at forbedre vilkårene for den digitale vækst hos offentlig-private partnerskaber. Dette gøres ved at afsætte 75 mio. kr. i 2018, samt 125 mio. kr. årligt i perioden 2019-2025. (Erhvervs- og vækstministeriet, 2017)

Regeringens økonomiske indsats tyder på et meget stabilt marked, som har en relativ lav risiko for konkurs. Til gengæld bliver der stillet økonomiske krav i anledning af statsprojekterne. For at vinde retten

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Det Tekniske Fakultet  
Gruppe 02 A

Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

til at levere et system til staten, skal det ønskede system leveres til under et bestemt beløb og ellers skal systemet udbydes på det europeiske marked. (IT-Branchen)

---

#### SOCIAL OG KULTURFORHOLD

Danmark er et land, hvor borgene går meget op i arbejdsglæde. Arbejdsglæde kan opnås på forskellig vis, men i Danmark er der et generelt ønske om mere ansvar og indflydelse. (Kristensen 2017, Røndbjerg-Christensen, 2014)

I det danske IT-arbejdsmarked er der et stort behov for kvalificeret arbejdskraft. Dette behov kan dog ikke blive dækket af det nuværende udbud som de danske uddannelsesinstitutioner leverer. Derfor må arbejdspladser kompensere for dette behov ved selvstændigt at uddanne og specialisere deres arbejdere eller udlicitere deres opgaver. (Jasper et al 2018, Stensdal 2018)

---

#### TEKNOLOGISKE FAKTORER

Softwareløsninger inden for den offentlige sektor er i hastig udvikling. Selvom udviklingen kan være svær at forudsige, er den stadig relativt begrænset af staten og EU, da det er deres behov, ønsker og regler der primært dikterer markedet. (Digitaliseringsstyrelsen, 2016)

---

#### LOVMÆSSIGE FORHOLD

Danmark er omfattet af de europæiske regler angående håndtering af personlige data. Disse regler har den danske regering suppleret med egne regler for at sikre den danske retstilstand. Hvis reglerne ikke bliver overholdt, betyder det at en organisation ikke må håndtere persondata. (Datatilsynet) Yderligere skal en organisation opfylde grundlæggende krav for at kunne indgå i konkurrencen om offentlige udbud. Disse regler betyder at der kan være konkurrence fra alle EU-lande, hvis udbuddet omhandler en kontrakt af en vis størrelse. (IT-branthen)

---

#### PORTRÆT'S 5 FORCES

---

#### TRUSLEN FRA POTENTIELLE INDTRÆNGERE

Generelt set er softwarebranchen en billig branche at etablere sig i. Dette skyldes, at det eneste der skal til for at starte en enkeltmandsvirksomhed der udvikler software, er at man køber en computer. Alt software til udviklingen kan findes i en gratis, open-source version. Dog er det ikke helt så simpelt at komme i betragtning ved et offentligt udbud. Årsagen til dette, er at en virksomhed skal påvise en vis erfaring inden for udbuddets område eller kunne give positive referencer til tidligere brancheskift som virksomheden har foretaget.

Ifølge Eg Team Online er det dyrt for kunderne at skifte system. Den primære udgift i den forbindelse er udhentning af data, da der både skal betales for at alt data bliver arkiveret og der skal betales for at data bliver overført til den nye leverandør, som sandsynligvis vil tage sig betalt for at gemme data i det nye system.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

Disse faktorer spiller en markant rolle i forhold til at afholde nye indtrængere fra at komme ind på markedet.

---

#### SUBSTITUERENDE PRODUKTER

Eftersom branchens produkter til dels er blevet udviklet for at mindske forbruget af papir, findes der analoge alternativer i form af diverse skemaer og kalendere. Det er dog ikke særlig sandsynligt at de vil blive taget i brug, da der er stillet krav fra politisk side om øget digitalisering i det offentlige.

---

#### LEVERANDØRERS FORHANDLINGSKRAFT

Ifølge EG Team Online har de en leverandør til hosting af data. EG Hosting udbyder løsninger for hosting af data og har en dybere viden omkring teknologisk infrastruktur.

Da denne virksomhed er ejet af samme koncern som den virksomhed der leverer hosting af data, er der en lav risiko for at blive presset i forhold til ydelser.

---

#### KUNDERS FORHANDLINGSKRAFT

Ifølge EG Team Online er kommunerne deres største kunde. Kommunerne bruger deres system, Sensum, til at håndtere den socialfaglige indsats på specialområdet. En anden kunde er private bosteder, der kan benytte systemet som en hjælp til ledelse, videndeling og dokumentation på de indskrevne borgere.

Det er dog ikke kunderne, som har den store magt over EG Team Online, da en udskiftning af systemudbyder er forbundet med høje omkostninger. Reglerne for EU-udbud skal også tages i betragtning, da de gør sig gældende i forhandlingssituationer.

---

#### KONKURRENCESITUATIONEN I BRANCHEN

Der er ifølge EG Team Online blevet stillet lovråv om at området for ældrepleje skal slås sammen med det sociale område. EG Team Online har ikke nogen erfaring med ældreområdet, hvilket har givet Systematic mulighed for at overtage nogle af EG Team Onlines kunder, da de har haft den ønskede erfaring.

KMD har været i branchen i mange år. Et af deres systemer kan levere det samme som Sensum. Nexus er en platform der indeholder KMD CARE, hvilket er en løsning der kan skabe overblik og sikre en sammenhængende indsats ved at gøre det muligt at arbejde med en samlet plan på tværs af forvaltninger. Platformen indeholder også KMD VIVA, der er en løsning til at samle digitale serviceydelser på tværs af fagsystemer. Kommuner får igennem Nexus en god mulighed for at være opdateret på borgerens forløb. (KMD)

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

## DISKUSSION

### THE CONTINGENCY APPROACH

EG Team Online er baseret i Danmark og benytter primært dansk arbejdskraft. De danske ansatte har generelt et ønske om at have ansvar for og indflydelse på deres arbejdsplads. Hvis virksomheden skulle strukturere deres virksomhed for at imødekomme dette ønske, skulle de benytte en organisk struktur (Organic Structure).

Markedet som EG Team Online har koncentreret deres aktiviteter i, er digitaliseringen af den danske velfærdssektor. Denne velfærdsdigitalisering har den danske stat kontinuerligt haft fokus på at udvikle. For at sikre en stabil udvikling, med et stærkt fundament som understøtter udviklere, har den danske stat valgt at afsætte kapital for at sikre en progressiv fremtid. Denne statslige strategi, danner fundament for et stabilt politisk miljø og giver anledning til at strukturere virksomheden mekanistisk (Mechanistic Structure).

Selvom der er gjort en stor indsats for at skabe de bedste vilkår for udviklingen af velfærdsdigitaliseringen, bliver den stadig styret af regler og restriktioner. Disse regler og restriktioner skaber rammen for EG Team Onlines udviklingsproces. Hvis ikke de overholdes, kan statsfinansieringen blive trukket tilbage. For at have en bæredygtig forretningspraksis skal der laves en klar og tydelig handleplan som medarbejderne kan følge. Skal dette eksekveres ordentligt, kræver det at selskabet udformer rammerne fra toppen af selskabet og ned. Dette skaber et behov for en høj grad af centraliseret styring, der indikerer at selskabet vil nyde godt af at blive struktureret mekanistisk (Mechanistic Structure).

Lovmæssigt er der regler for hvordan en offentlig instans skal forholde sig, når der skal laves et udbud. I opgaver af en vis størrelse må der ikke sættes nogen begrænsninger, der kan gøre det mindre gunstigt for virksomheder fra andre EU-lande at byde på en opgave. Det betyder, at der kan være øget konkurrence i branchen, men også at det er muligt for EG Team Online at udvide deres marked til hele EU ved at byde på en kontrakt i et andet EU-land.

Analyserer man de fire forhold som kan have en indflydelse på konkurrencesituationen i en branche, trusler fra potentielle indtrængende, kunders forhandlingsmagt, leverandørers forhandlingskraft og substituerende produkter, kan man se at de ikke har en særlig stor indflydelse på konkurrencesituationen.

De primære konkurrenter som EG Team Online kæmper mod, er KMD og Systematic. Begge selskaber er allerede etableret på markedet og udgør derfor den største risiko i forhold til fremtidige projekter.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

Aflevering:  
22-03/2019



Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

## KONKLUSION

Ved hjælp af relevante dele af PESTEL har vi kunne drage følgende delkonklusion om det generelle miljø EG Team Online befinder sig i.

De politiske og økonomiske forhold i Danmark er stabile. Markedet som EG Team Online opererer i er ligeledes stabilt. Den offentlige sektor udvider deres digitale domæne og har offentliggjort planer om at benytte private virksomheder til dette. EG Team Online kan derfor forvente at de eksisterende økonomiske muligheder fortsat vil eksistere.

De sociale og kulturelle forhold har betydelige udfordringer i forhold til efterspørgslen på specialister. Efterspørgslen er større end udbuddet og det skaber stor konkurrence om den eksisterende arbejdskraft på markedet og kan tvinge virksomheder til selv at uddanne folk eller at finde den uden for landets grænser.

De lovmæssige forhold som reguleres af den danske regering, samt den Europæiske Union, anviser hvordan personlige data skal håndteres. For at EG Team Online skal være i stand til at vinde et udbud, er de nødt til at overholde persondatalovgivningen. Reglerne for udbud gør at EG Team Online, vil have øget konkurrence fra EU-lande, men samtidig giver de også mulighed for at EG Team Online kan udvide deres marked.

Vi er på baggrund af Porters five forces kommet med en delkonklusion om EG Team Onlines specifikke miljø.

EG Team Onlines leverandørers og kunders lave forhandlingskraft viser at det er virksomheden, der har kontrollen. Samtidig er de store omkostninger forbundet med udskifning af EG Team Onlines systemer hos kunden medvirkende til at formindskede kunders forhandlingskraft. Derudover ejer Koncernen EG den benyttede leverandør og har derfor ingen grund til at presse EG Team Online i forhandlingerne.

Den store adgangsbarriere til markedet, og det at der nærmest ikke findes substituerende produkter, betyder at truslen for at der skulle komme nye konkurrenter, samt at EG Team Onlines kunder finder substituerende produkter er meget lav.

Ifølge Eg Team online er deres største Konkurrent KMD. Branchens konkurrence niveau afhænger af udbuddene. Det betyder at der er en høj konkurrence, når der er et udbud.

Inceptionsfasen  
25-02/2019 – 22-03/2019

Inceptionsdokument  
Semester Gruppe 02

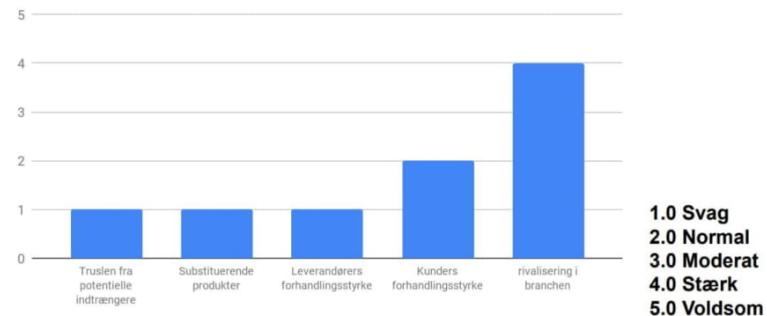
Aflevering:  
22-03/2019



Aslak Mangino Nicolajsen,  
Mathias Jakobsen,  
Steffen Vitten Pedersen

### EG Team Online ApS

Konklusion til Porters 5 forces



Når alle analyser samles, kan der konkluderes at en mekanistisk struktur vil være den mest fordelagtige for EG Team Online.

### 3 Rapportkontrolskema

A. Produktrapport		
Kapitel	krav	opfyldt +/–
Omslag	Indeholder omslaget projekttitel, uddannelsesinstitution, fakultet, institut, uddannelse, semester, kursuskode, projektperiode, vejleder, projektgruppe og projektdeltagere (fornavn, efternavn, sdu-email)?	+
Titelblad	(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)	+
Resumé	<p>Omfatter resuméet:</p> <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> <p>(max 1 side)</p>	+
Forord	<p>Indeholder forord hensigten med rapporten, målgruppe, forhistorie, anerkendelser, afleveringsdato samt underskrifter af alle projektdeltagere?</p> <p>Bemærk: Projektdeltagernes aktive deltagelse i projektforløbet anerkendes gensidigt ved projektdeltagernes underskrifter i rapporten.</p>	+
Indholdsfortegnelse	<p>Er der en samlet indholdsfortegnelse for hele projektrapporten?</p> <p>med to eller tre niveauer?</p> <p>Er afsnittene nummererede?</p>	+

Læsevejledning	<p>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?</p> <p>Er rapportens målgruppe beskrevet?</p>	+																
Redaktionelt	<p>Beskriver redaktionelt skriveprocessen og ansvarsområder i skriveprocessen?</p> <p>Ansvarsområder kan fx beskrives på fx følgende form:</p> <table border="1"> <thead> <tr> <th>Afsnit</th><th>Ansvarlig</th><th>Bidrag fra</th><th>Kontrolleret af</th></tr> </thead> <tbody> <tr> <td>Afsnit a</td><td>Person a</td><td>Person b</td><td>Person a,b,c</td></tr> <tr> <td>Afsnit b</td><td>Person b</td><td>Person a</td><td>Person a,b,c</td></tr> <tr> <td>Afsnit c</td><td>Person c</td><td>Person b</td><td>Person a,b,c</td></tr> </tbody> </table>	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	+
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 fagterminer 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. inceptiondokumentet? 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?	+																
Metoder		+																
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		+
1. Indledning	Indeholder indledningen en overordnet introduktion til afsnittet?	+
2. Overordnet kravspecifikation (resume, opdateret)	Indeholder afsnittet et opdateret resumé af systemafgrænsningen fra inceptionsdokumentet.	+
3. Detaljeret kravspecifikation	Omfatter den detaljerede kravspecifikation  1. Detaljeret brugsmønsterdiagram (hvis relevant) 2. Detaljerede brugsmønsterbeskrivelser 3. Detaljerede beskrivelser af supplerende krav Fx organiseret efter FURPS+	+

	Omfatter afsnittet overvejelser, beslutninger og resultater vedr.	
4. Analyse	<p>1. Den statiske side af analysemodel</p> <p>2. Den dynamiske side af analysemodel</p>	+
5. Design	<p>1. Softwarearkitektur</p> <p>2. Subsystemdesign</p> <p>3. Den statiske side af designmodel</p> <p>4. Den dynamiske side af designmodel</p> <p>5. Design af persistens</p> <p>6. Databasedesign</p>	+
6. Implementering	<p>Omfatter afsnittet overvejelser, beslutninger og resultater vedr. konvertering fra design til kode illustreret gennem udvalgte centrale eksempler, samt andre vigtige implementeringsbeslutninger</p> <p>Omfatter afsnittet implementering af database.</p>	+
7. Test	Omfatter afsnittet en beskrivelse af de udførte test samt resultatet af dem.	+
	Er der medtaget resultater både fra iteration # 1 og fra iteration # 2?	+
Diskussion	<p>Omfatter diskussionen hvad der er opnået, og hvad der ikke er opnået i projektet i forhold til det forventede som beskrevet i indledningen.</p> <p>Hvad er styrkerne og svaghederne ved jeres resultater?</p> <p>Kunne I have opnået bedre resultater?</p>	+

Konklusion	Opsummerer konklusionen resultaterne og diskussionen af dem og giver det på problemformuleringen?	+
Perspektivering	<p>Fremitdigt arbejde: Hvad ville de næste skridt i projektet være, hvis der var mere tid?</p> <p>Refleksion: Hvordan ville I gøre projektet an, hvis I skulle starte forfra?</p>	+
Litteraturliste	<p>Er litteratur angivet på en anerkendt form?</p> <p>Er alle former for litteratur som bøger, artikler og hjemmesider medtaget?</p> <p>Er der kildehenvisninger i teksten? Materiale som gruppen ikke selv har fremstillet i dette projekt skal være angivet med kilde!</p> <p>Er alle kildehenvisninger i teksten anført på samme måde?</p> <p>Er der kildeangivelser på figurer, grafer etc. som projektgruppen ikke selv har frembragt?</p>	+

B. Procesrapport		
Kapitel	krav	opfyldt +/–
Læring og refleksion	Er der en redegørelse for læring og refleksion?	+
	Er der en redegørelse for projektplanlægning og styring? Er der en beskrivelse af den faktiske projektplanlægning og styring, med inddragelse af væsentlige artefakter, som fx sprint backlogs, væsentlige ceremonier, som fx daily scrum, rollevaretagelse mm?	+
Identifikationg, analyse og bearbejdning af problemer	Er der en redegørelse for identifikation, analyse og bearbejdning af problemer	+
Udviklingsprocessen	Er der en redegørelse for det faktiske udviklingsarbejde. Er der en beskrivelse af faserne, iterationerne og det faktiske arbejde i dem?	+
Formidling og kommunikation	Er der en redegørelse for formidling og kommunikation	+
Samarbejde i gruppen	Er der en redegørelse for samarbejde i gruppen	+
Samarbejde med vejleder	Er der en redegørelse for samarbejde med vejleder	+

C. Kildekode		
Kapitel	krav	opfyldt +/–
Oversigt over projektets kildekode	Er der en oversigt over projektets kildekode, fx filstruktur eller javadoc?	+

D. Brugervejledning		
Kapitel	krav	opfyldt +/–
Brugervejledning	Er der en kortfattet brugervejledning?	+

E. Projektlog		
Kapitel	krav	opfyldt +/–
Projektlog	Er der en adresse på og et link til projektloggen i Github	+

F. Interne bilag		
Kapitel	krav	opfyldt +/–
Projektforslag	Er projektforslaget medtaget?	+
Inceptionsdokument	Er Inceptionsdokumentet medtaget? Inceptionsdokumentet skal afleveres i den udgave det blev bedømt.	+
Rapportkontrolskema	Er der et udfyldt rapportkontrolskemaet	+
Andet	Er der andre relevante interne bilag, dvs. materialer produceret af gruppen selv?	+

G. Eksterne bilag		
Kapitel	krav	opfyldt +/–
Eksterne bilag	Er der medtaget relevante eksterne bilag, dvs. materialer som gruppen ikke selv har produceret med som er nødvendige for at kunne læse rapporten?	+

H. Rapporttekniske elementer		
Kapitel	krav	opfyldt +/–
Layout	Er der anvendt samme layout i alle kapitler? Er layout overskueligt/harmonisk?	+
Sprog	Er rapporten skrevet i en neutral sprogtone? Er sproget let læseligt og flydende? Er der udført stavekontrol og kontrol af tegnsætning?	+
Sidenummerering	Er der korrekt og konsistent sidenummerering i rapporten?	+
Figurer/diagrammer	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?	+
Tabeller	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?	+
Sporbarhed af begreber	Er der en konsekvent brug af samme betegnelse for et givet begreb igennem rapporten?	+

## 4 Diverse interne materialer

<b>Brugsmønster:</b> Opret sag
<b>ID:</b> 1
<b>Primære aktører:</b> Sagsbehandler, afdelingsleder, administrativt personale
<b>Sekundære aktører:</b> CPR
<b>Kort beskrivelse:</b> En aktør kan oprette en sag, som gemmes i systemet.
<b>Prækonditioner:</b> Aktør skal være logget ind.
<b>Hovedhændelsesforløb:</b> Starter når en borger henvender sig til kommunen 1. Aktør indtaster CPR nummer, borgers navn, begrundelse for henvendelse 2. Gemmer indtastet data.
<b>Postkonditioner:</b> sag oprettet
<b>Alternative hændelsesforløb:</b>

Figur F.1: Brugsmønster for opret sag

<b>Brugsmønster:</b> Behandle sag
<b>ID:</b> 2
<b>Primære aktører:</b> Sagsbeandler
<b>Sekundære aktører:</b> Handleplan modul
<b>Kort beskrivelse:</b> Sagsbeandlerne starter behandling af en sag i forhold til indhentning af oplysninger, hvorefter sagen bliver vurderet. Efterfølgende kommer en sagsafgørelse og til sidst bestilles evt. ydelser.
<b>Prækonditioner:</b> Der skal være oprettet en sag.
<b>Hovedhændelsesforløb:</b> Starter når sagsbeandler vælger "Behandle sag". 1. Sagsbeandler udfylder en sagsåbningsformular. 2. hvis sagsåbningsformular ikke er udfyldt. 2.1 Alternativt hændelsesforløb Behandle sag : Manglende information 3. hvis sagsåbningsformular er udfyldt 3.1 Udredningsformularen udfyldes med dataer fra sagsåbningsformular. 3.2 Der bliver sendt data til handleplan modulet ift. oprettelse af en handleplan. 3.3 Der vælges mellem alternativt forløb indhent oplysninger eller sagsafgørelse.
<b>Postkonditioner:</b> none
<b>Alternative hændelsesforløb:</b> Indhente oplysninger fra eksterne kilder, Afgøre sagsbehandling, Manglende information

Figur F.2: Brugsmønster for behandle sag

Figure F.12 til F.25 viser hvad gruppen havde tænkt der skal ske i systemmet i første iteration.

<b>Alternativ hændelsesforløb:</b> Behandle sag: Indhente oplysninger fra eksterne kilder
<b>ID:</b> 2.1
<b>Primære aktører:</b> Sagsbehandler
<b>Sekundære aktører:</b> Sundhedssystem
<b>Kort beskrivelse:</b> Indhenter oplysninger fra eksterne systemer
<b>Prækonditioner:</b> Der skal være en udredningsformularen der er påbegyndt
<b>Alternativhændelsesforløb:</b>
1. Indtastning af oplysninger i relevante felter fra eksterne kilder i udredningsformularen.
2. Formularen gemmes
3. Sagen lukkes
<b>Postkonditioner:</b> Formularen opdateret

Figur F.3: Alternativ hændelsesforløb for behandle sag

<b>Alternativ hændelsesforløb:</b> Behandle sag : Afgøre sagsbehandling
<b>ID:</b> 2.2
<b>Primære aktører:</b> Sagsbeandler
<b>Sekundære aktører:</b> Handleplan modul, Dagbog modul
<b>Kort beskrivelse:</b> Sagsbeandler registrerer afgørelse af pågældende sag og skriver et brev til borger omkring afgørelsen. Hvis sagen ikke er afvist, udfyldes en bestilling af social indsats.
<b>Prækonditioner:</b> Der er indhentet oplysninger fra eksterne kilder.
<b>Alternativhændelsesforløb:</b>
1. Hvis sagen ikke afvises.
1.1 Udfylder formularen for bestilling af social indsats
1.2 Systemet sender oplysninger til handleplan modulet ift. færdiggørelse af handleplanen.
1.3 Systemet sender oplysninger til dagbog modulet ift. oprettelse af dagbog.
2. Der udfyldes et afgørelsesebrev.
3. Afgørelsesebrevet gemmes.
4. Sagen lukkes.
<b>Postkonditioner:</b> Sagen er opdateret. Der skal være udfyldt et afgørelsesebrev.

Figur F.4: Alternativ hændelsesforløb for behandle sag

<b>Alternativ hændelsesforløb:</b> Behandle sag : Manglende information
<b>ID:</b> 2.3
<b>Primære aktører:</b> Sagsbehandler
<b>Sekundære aktører:</b> none
<b>Kort beskrivelse:</b> Bruges i tilfælde af at en sagsåbningsformular ikke er tilstrækkeligt udfyldt.
<b>Prækonditioner:</b> Der skal være startet en sag, med delvist udfyldt formular.
<b>Alternativhændelsesforløb:</b>
1. Starter ved mangel på information i trin 2 i behandle sag. 2. Når formularen genoptages. 2.1 Åbnes formularen og manglende information udfyldes. 3. Hvis formularen er udfyldt. 3.1 Gå til trin 3 i behandle sag.
<b>Postkonditioner:</b> none

Figur F.5: Alternativ hændelsesforløb for behandle sag

<b>Brugsmønster:</b> Find sag
<b>ID:</b> 3
<b>Primære aktører:</b> Sagsbehandler
<b>Sekundære aktører:</b> none
<b>Kort beskrivelse:</b> Skal kunne søge i sager og få en sag vist
<b>Prækonditioner:</b> Der skal være oprettet en eller flere sager
<b>Hovedhændelsesforløb:</b>
Starter når den primære aktør skal finde en sag.
1. Skal søge en sag på sagsnummer, CPR nummer, eller navn.
2. Vise en liste over de sager som blev fundet.
3. Hvis en sag bliver valgt.
3.1 Vis den valgte sag.
<b>Postkonditioner:</b>
<b>Alternative hændelsesforløb:</b>

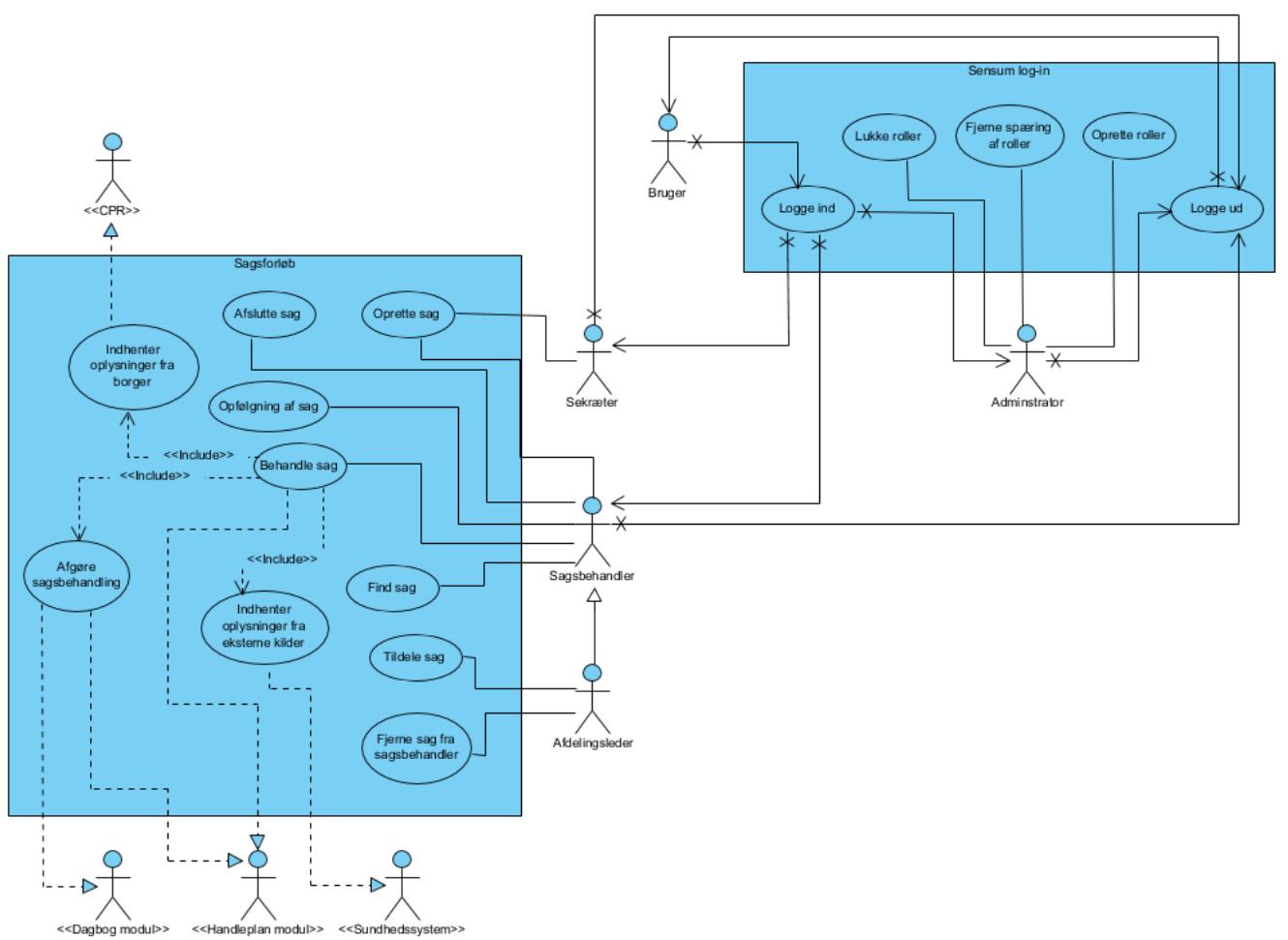
Figur F.6: Brugsmønster for find sag

Krav #	Kategori	Kravsbeskrivelse	Nytte	Risiko	Prioritet
01	Sensum - Login	Bruger kan logge ind som Sagsbehandler, afdelingsleder, sekretær eller administrator.	Middel	Lav	Lav
02	Sensum - log ud	Som sagsbehandler, afdelingsleder, sekretær eller administrator skal man kunne logge ud.	Middel	Lav	Lav
03	Sensum - Moduler	Ved login, skal man have mulighed for at vælge mellem forskellige moduler (Simulering).	Middel	Lav	Lav
04	Sagsforløb - Sagsoprettelse	Sagsbehandleren og det administrative personale skal kunne oprette en sag	Høj	Middel	Høj
05	Sagsforløb - Find sag	En sagsbehandler og afdelingsleder skal kunne søge efter en sag via et sagsnummer, fulde navn eller CPR-nummer.	Middel	Middel	Middel
06	Sagsforløb - Sagsbehandling	Under sagsbehandlingen skal en sagsbeandler have mulighed for at kunne indhente oplysninger fra den pågældende borger og vælge rette paragraf ud fra en formular ved videre sagsbehandling.	Høj	Høj	Høj
07	Sagsforløb - Sagsafslutning	Sagsbehandleren skal kunne lukke en sag, når den given borger er afsluttet.	Høj	Høj	Høj
08	Sagsforløb - Opfølgning af sag	En sagsbeandler skal kunne følge op på en sag og der skal kunne udfyldes type af opfølgning og datoen opfølgningen er sket.	Middel	Lav	Lav
09	Sagsforløb - Sagsredigering	En sag skal kunne redigeres af enten en sagsbeandler eller en sekretær.	Høj	Middel	Høj
10	Sagsforløb - Sagsafgørelsen	En sag skal kunne afgøres, hvor nødvendige informationer sendes til rette moduler for oprettelse af nødvendige handlinger ifølge med en borgeres sagsstatus.	Høj	Middel	Høj

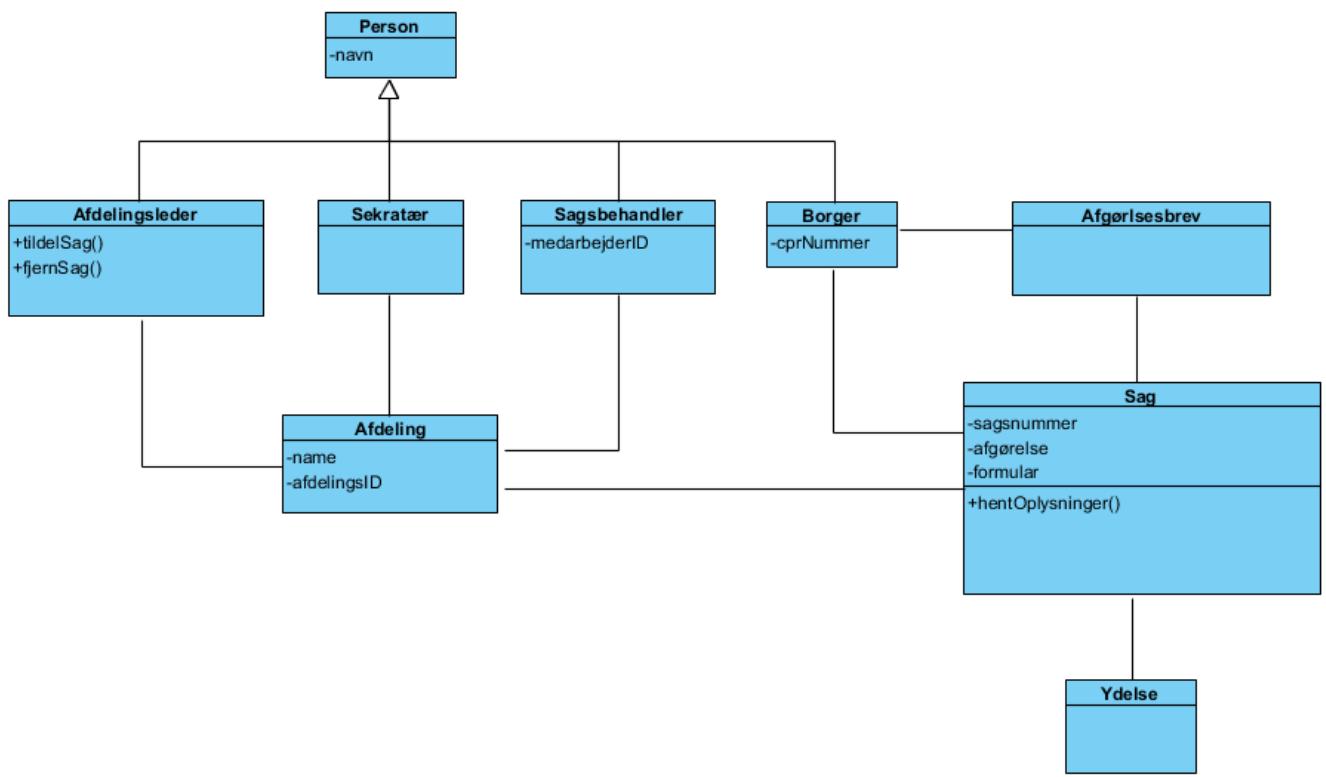
Figur F.7: Funktionelle krav for projektet

Ikke funktionelle krav					
#	Kategori	Kravsbeskrivelse	Nytte	Risiko	Prioritet
01	Modularitet	Det skal være nemt at kommunikere med andre moduler.	Middel	Lav	Middel
02	Sagsbehandlers adgang	En sagsbeandler tilhører en afdeling og kan kun se sager under samme afdeling.	Høj	Høj	Høj
03	Borgers sagsbeandler	En børger kan have flere sagsbeandler fra forskellige afdelinger.	Middel	Middel	Lav
04	Borger	Der skal kunne være flere sager åbne på en børger.	Middel	Middel	Lav
05	Borger	Borgeren skal kunne få tildelt forskellige ydelser.	Middel	Lav	Lav
06	Sagsbeandler	Sagsbeandleren skal kunne åbne flere sager inden for samme afdeling.	Middel	Lav	Lav
07	Data	Data skal kunne håndtere forskellige borgere og deres forskellige sager indenfor samme afdeling og forskellige afdelinger.	Høj	Høj	Høj
08	Data	Der skal generere sagsnummer, samt udfylde andre genererede tal og strenge.	Høj	Høj	Høj
09	Eksternt system (CPR)	Der skal kunne hentes oplysninger fra personregisteret. (Fiktiv data)	Høj	Middel	Middel
10	Eksternt system (Sundhedssystem)	Der skal kunne hentes læge noter/lægeerklæring. (Fiktiv data)	Høj	Middel	Middel
11	Eksternt system (Handleplan modul)	Skal kunne sende data til handleplan. (data sendes til fiktivt modul)	Høj	Middel	Lav
12	Eksternt system (Dagbog modulet)	Skal sende data fra afgørelsen med de nødvendige oplysninger. (data sendes til fiktivt modul)	Høj	Middel	Lav

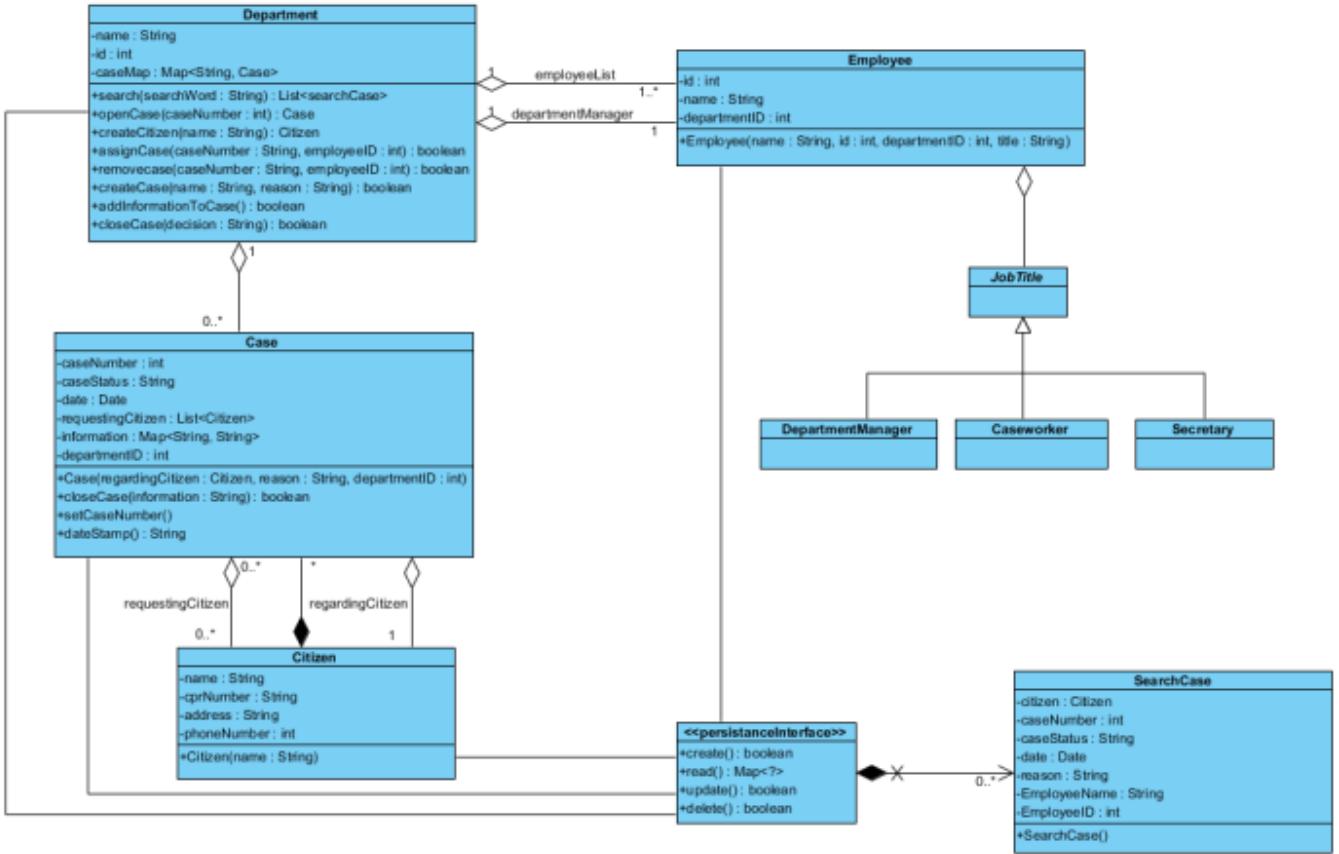
Figur F.8: ikke funktionelle krav for projektet



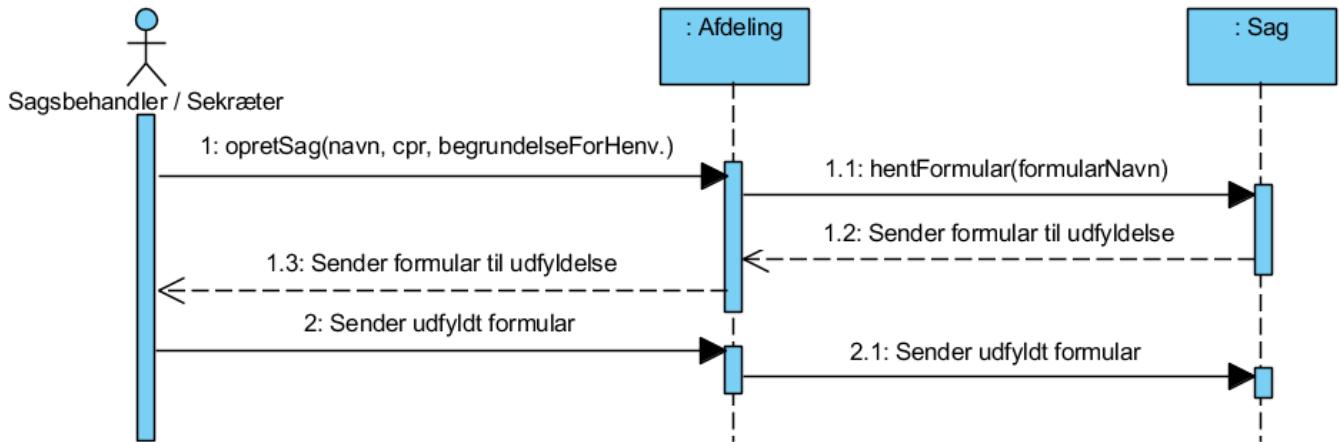
Figur F.9: Fuld størrelse af overordnede krav.



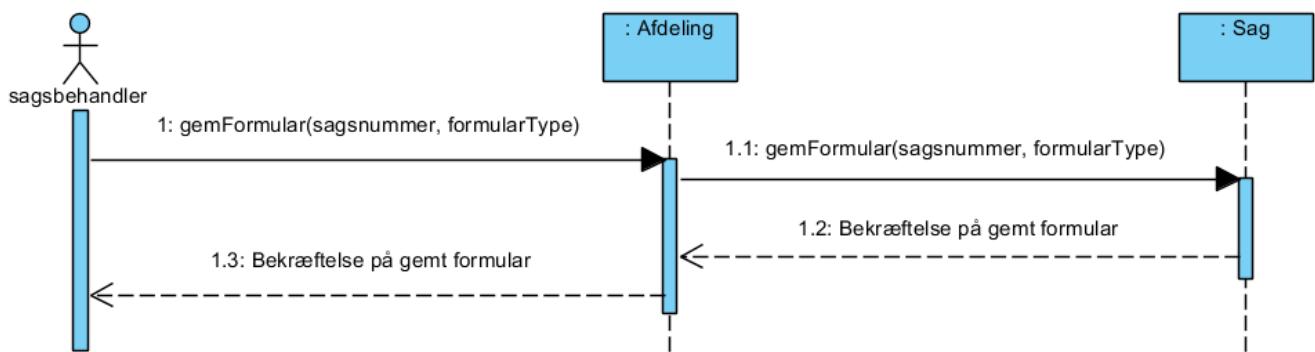
Figur F.10: Fuld størrelse af analyse klasse diagram.



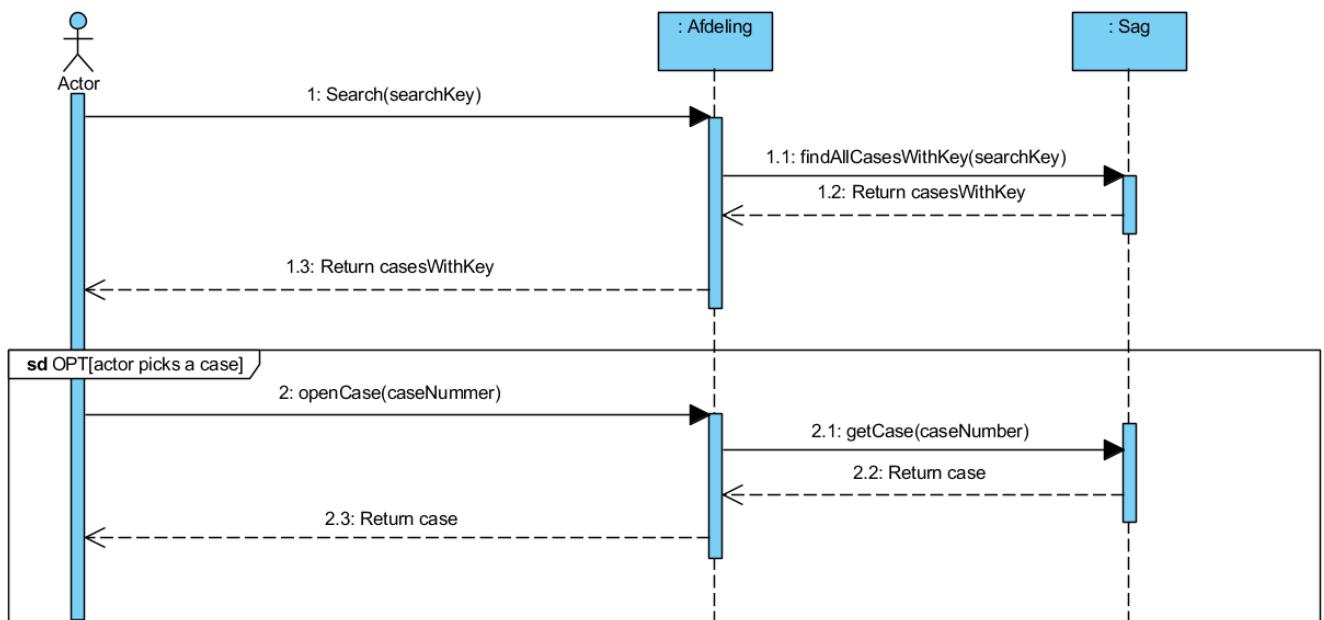
Figur F.11: Fuld størrelse af design klasse diagram.



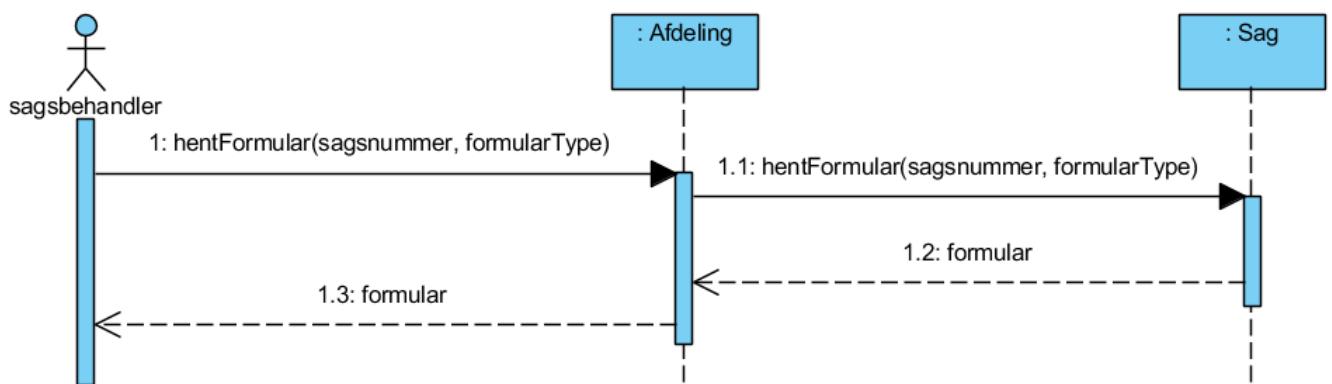
Figur F.12: Sekvensdiagram for opret sag.



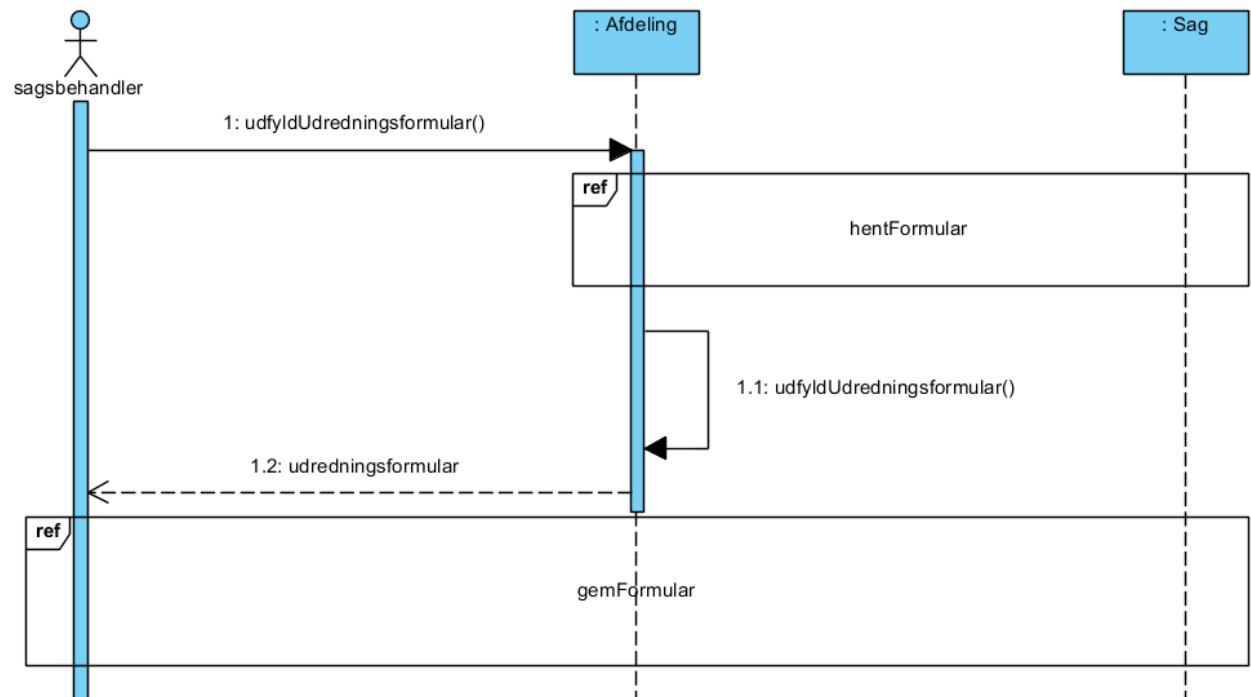
Figur F.13: Sekvensdiagram for gem formular.



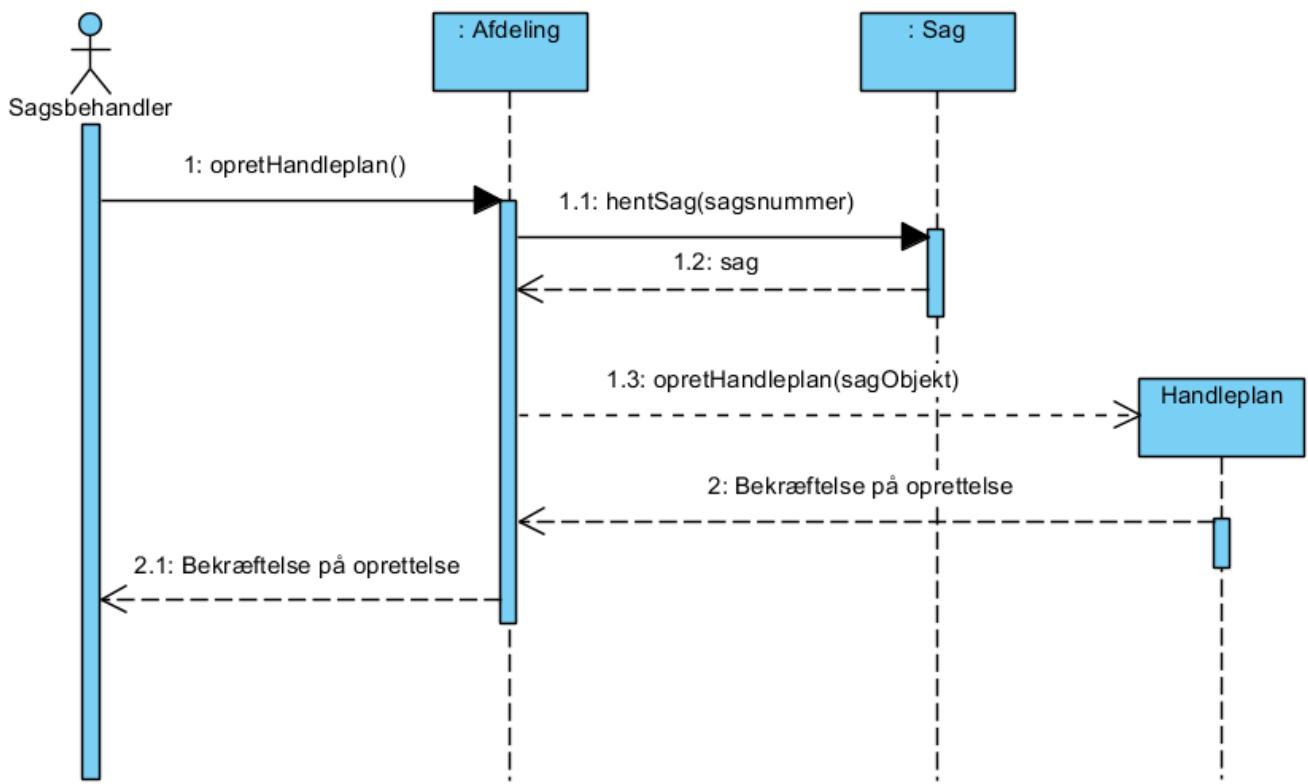
Figur F.14: Sekvensdiagram for find sag.



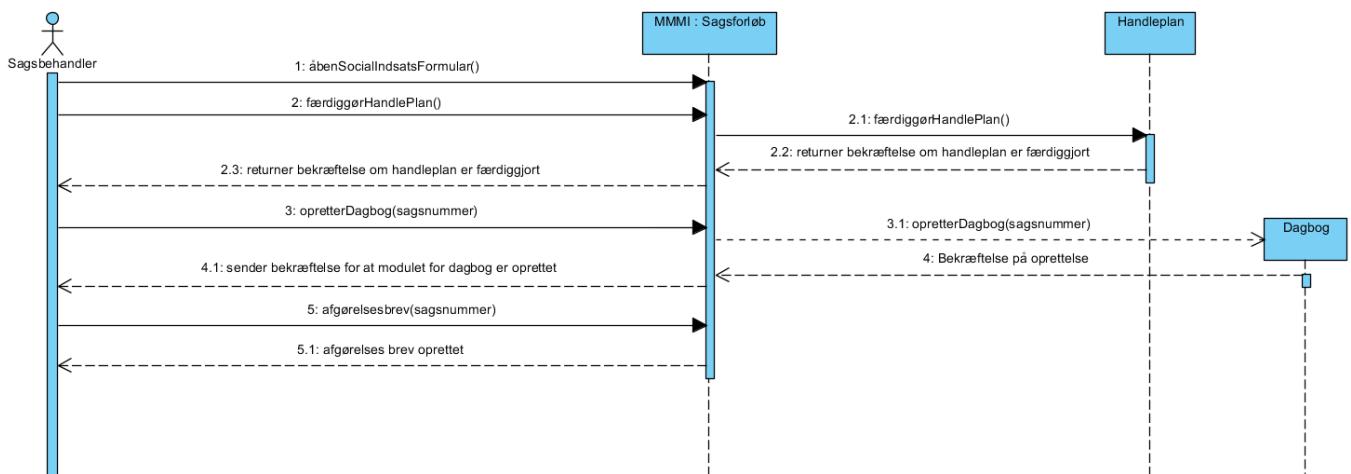
Figur F.15: Sekvensdiagram for hent formular .



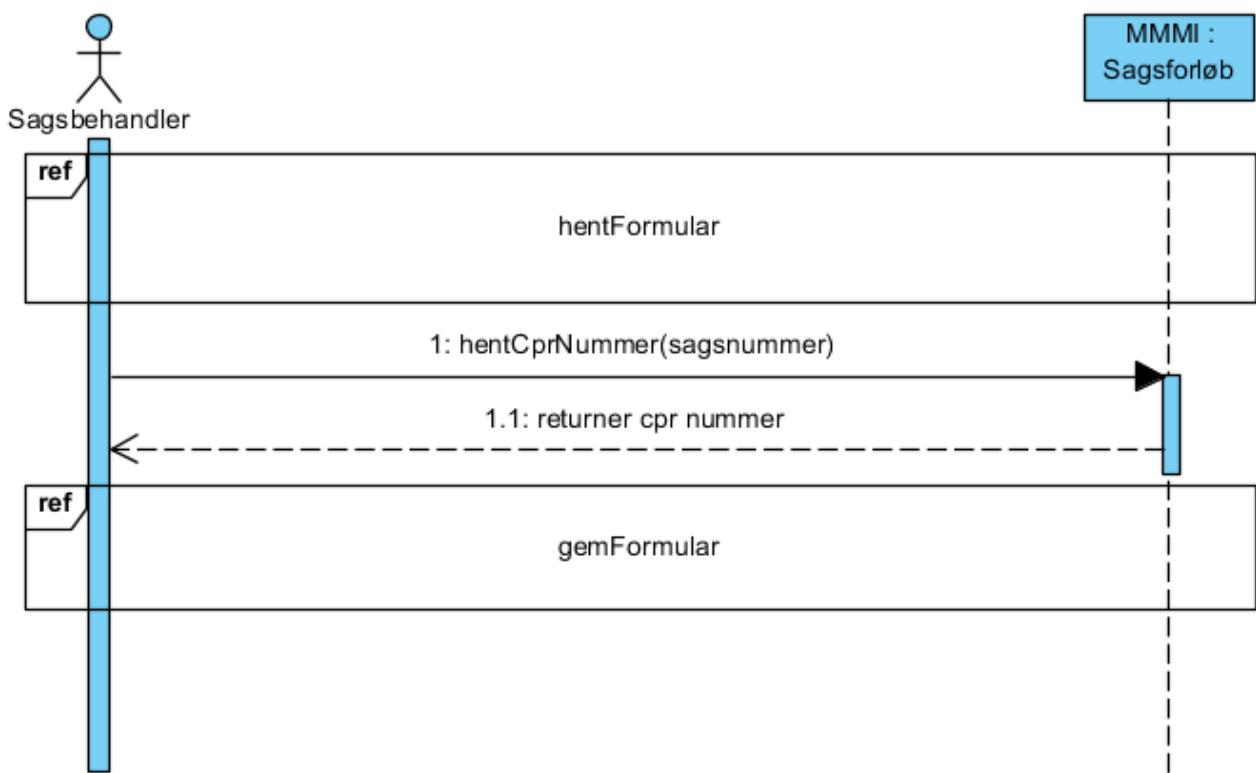
Figur F.16: Sekvensdiagram for udfyldning af udrednings formular.



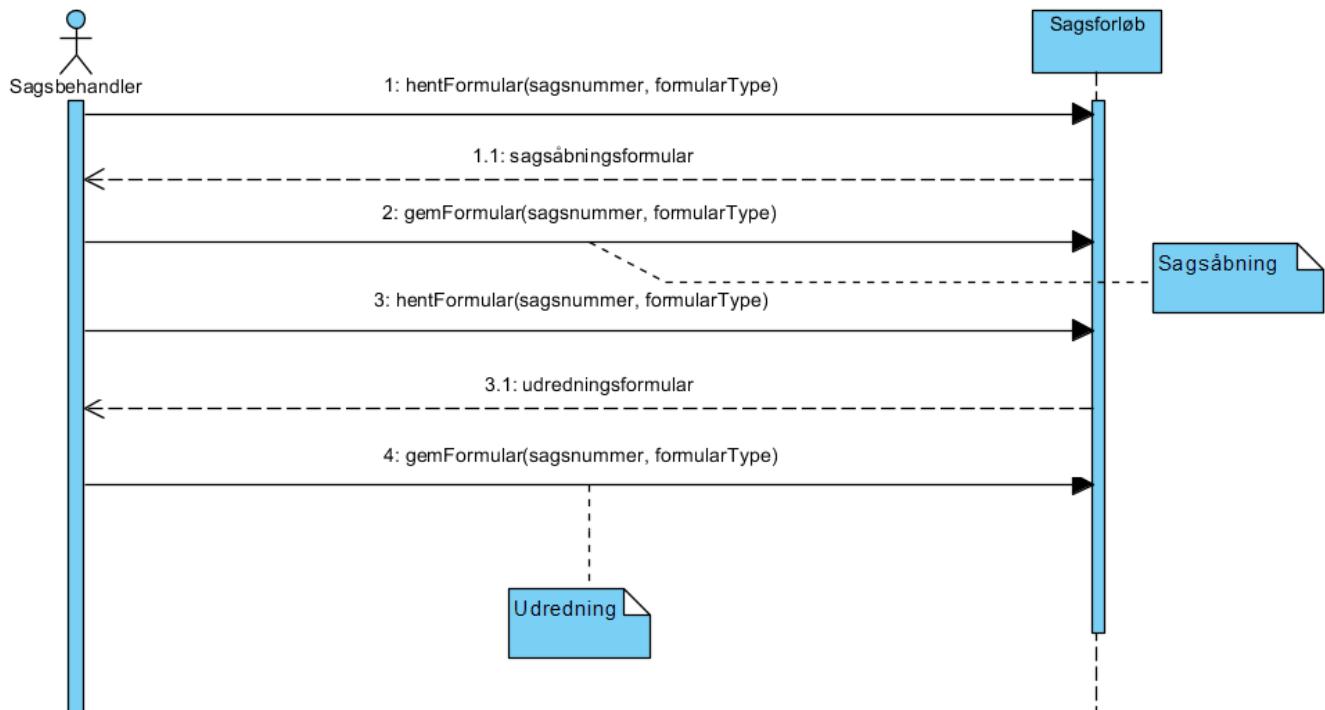
Figur F.17: Sekvensdiagram for opret handleplanen .



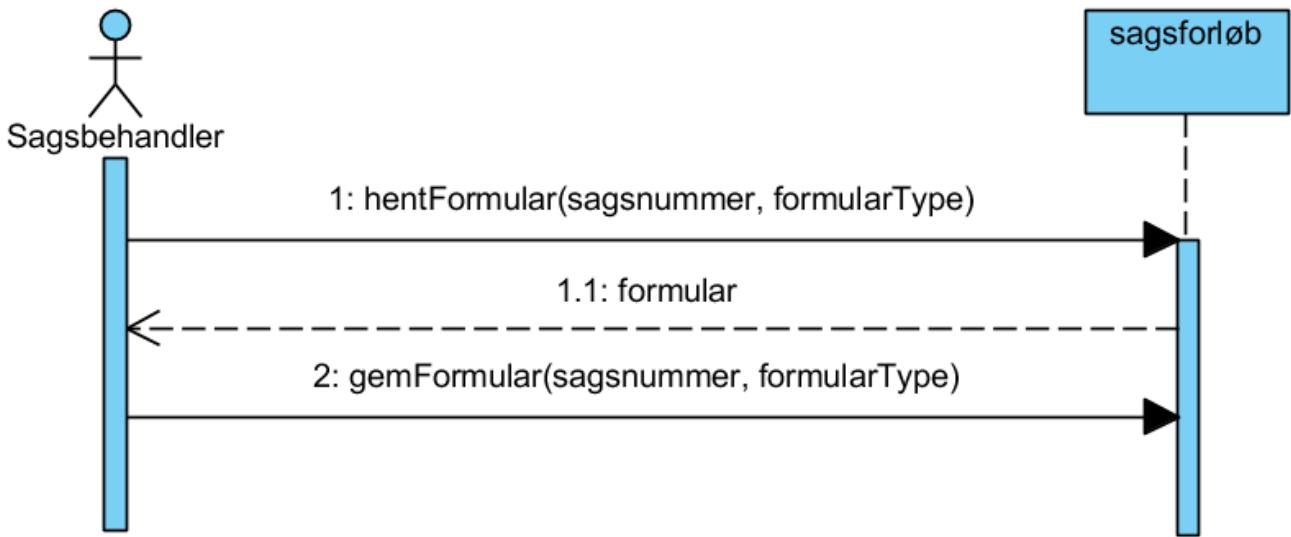
Figur F.18: Sekvensdiagram for opret handleplanen .



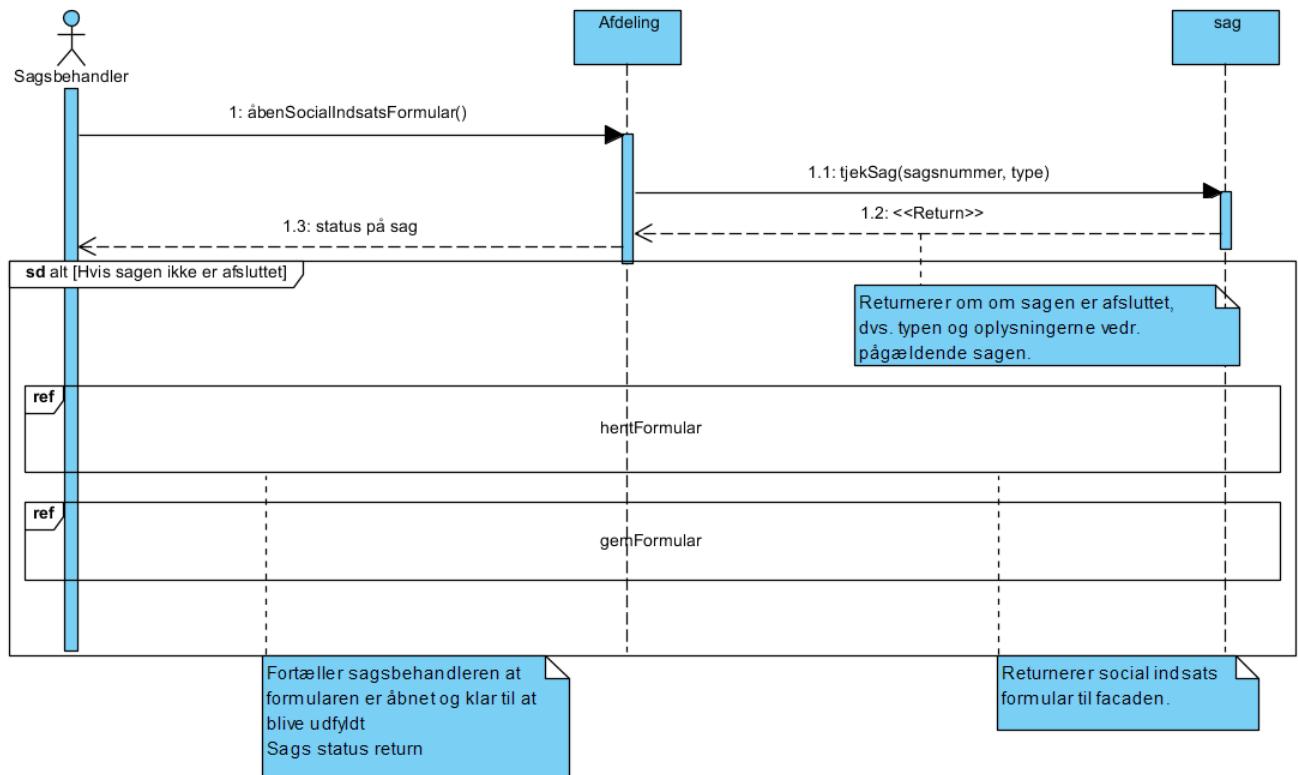
Figur F.19: Sekvensdiagram for opret handleplanen .



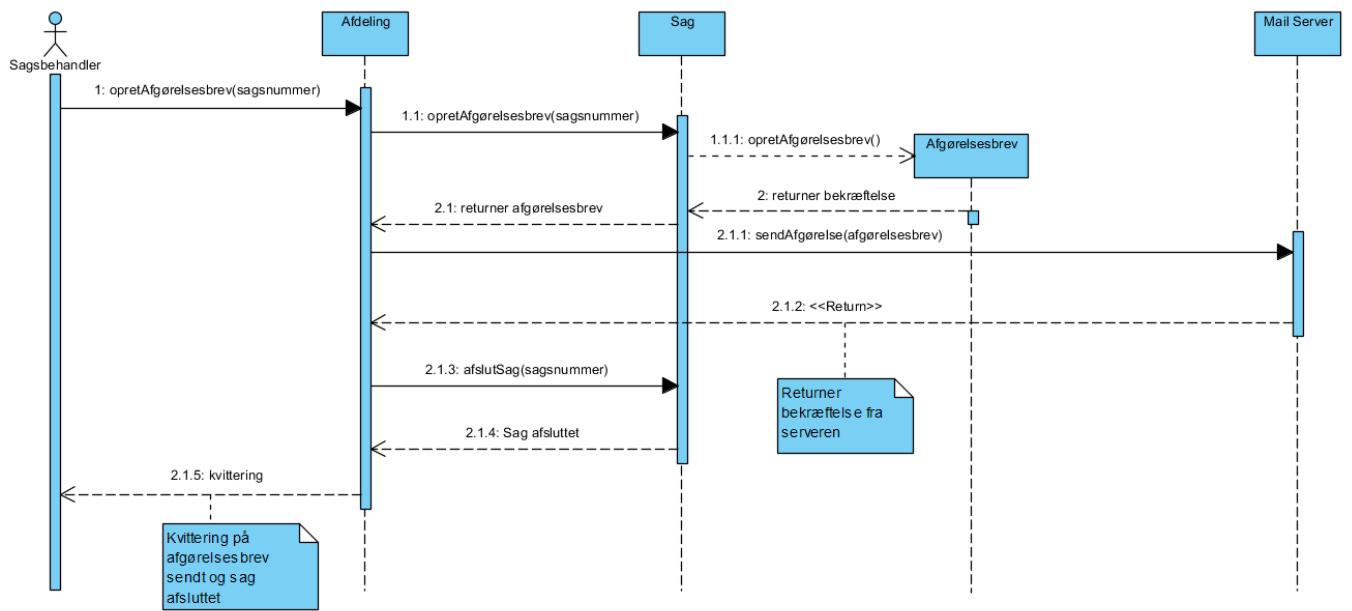
Figur F.20: Sekvensdiagram for behandel sag.



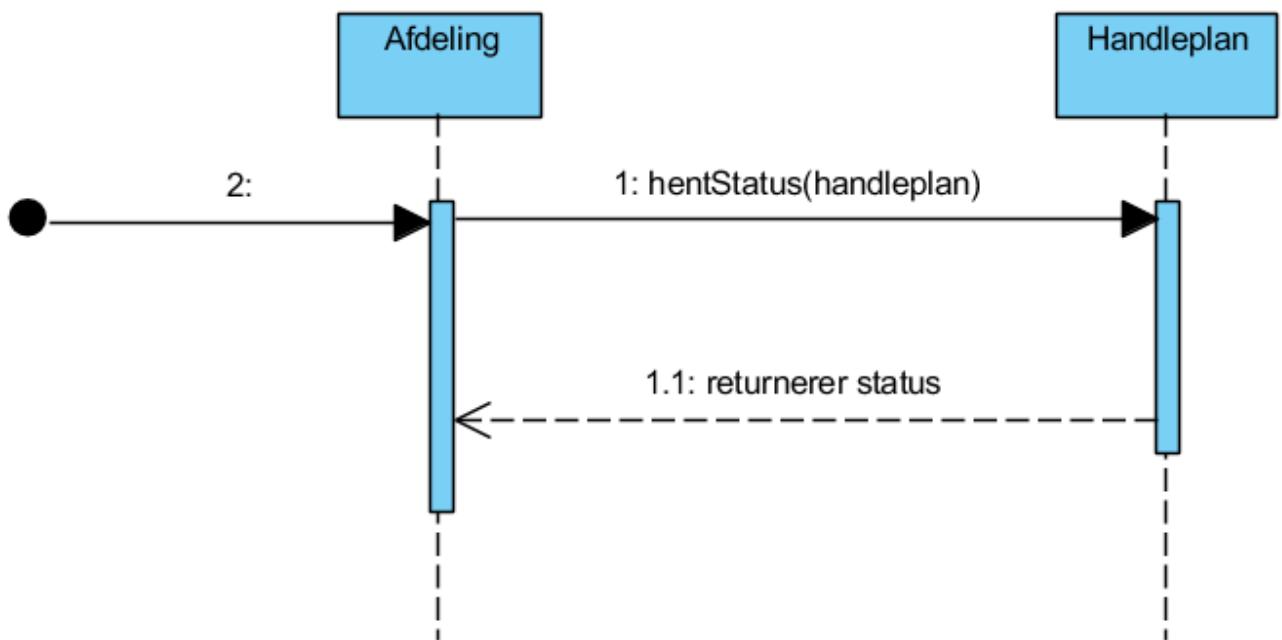
Figur F.21: Sekvensdiagram for behandel sag hvor der er mangel på information.



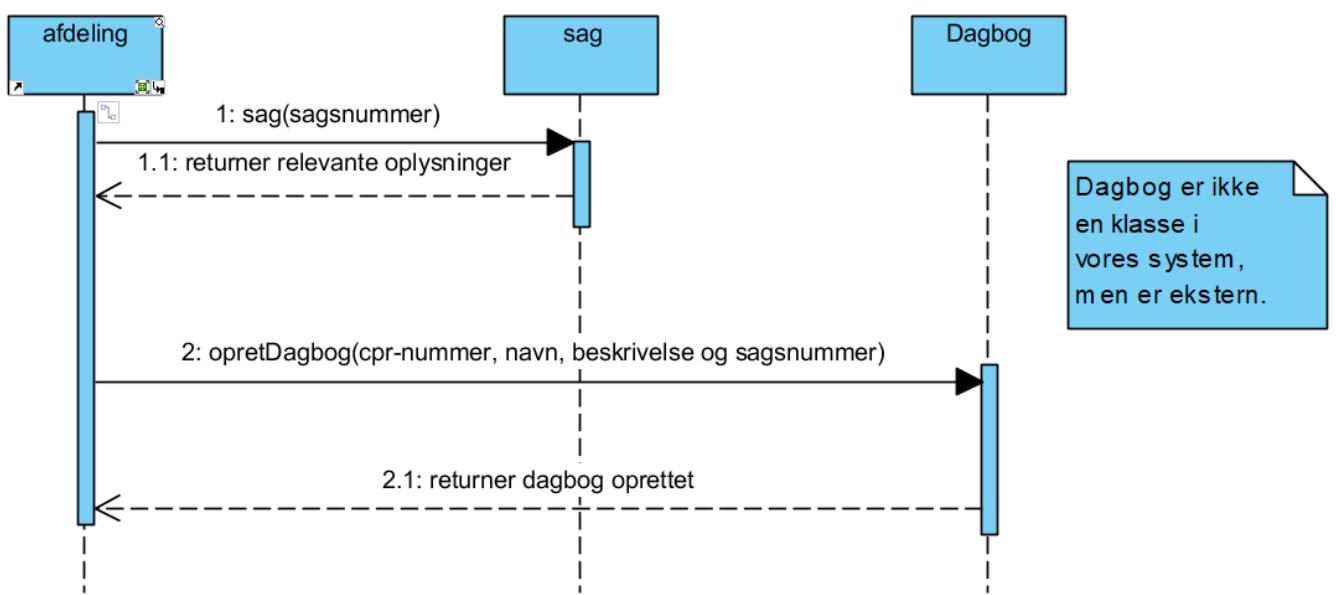
Figur F.22: Sekvensdiagram for behandel sag hvor der er mangel på information.



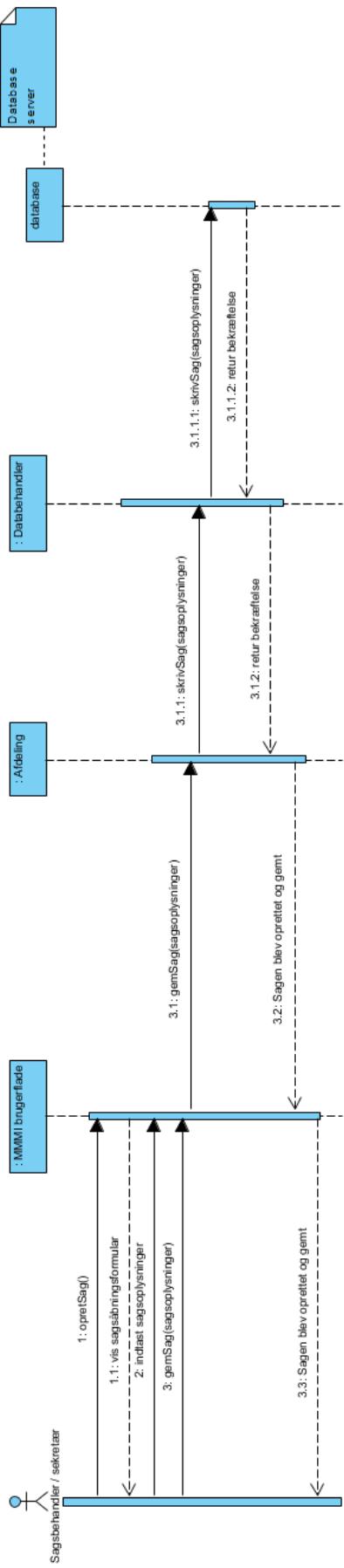
Figur F.23: Sekvensdiagram for behandel sag hvor der er mangel på information.



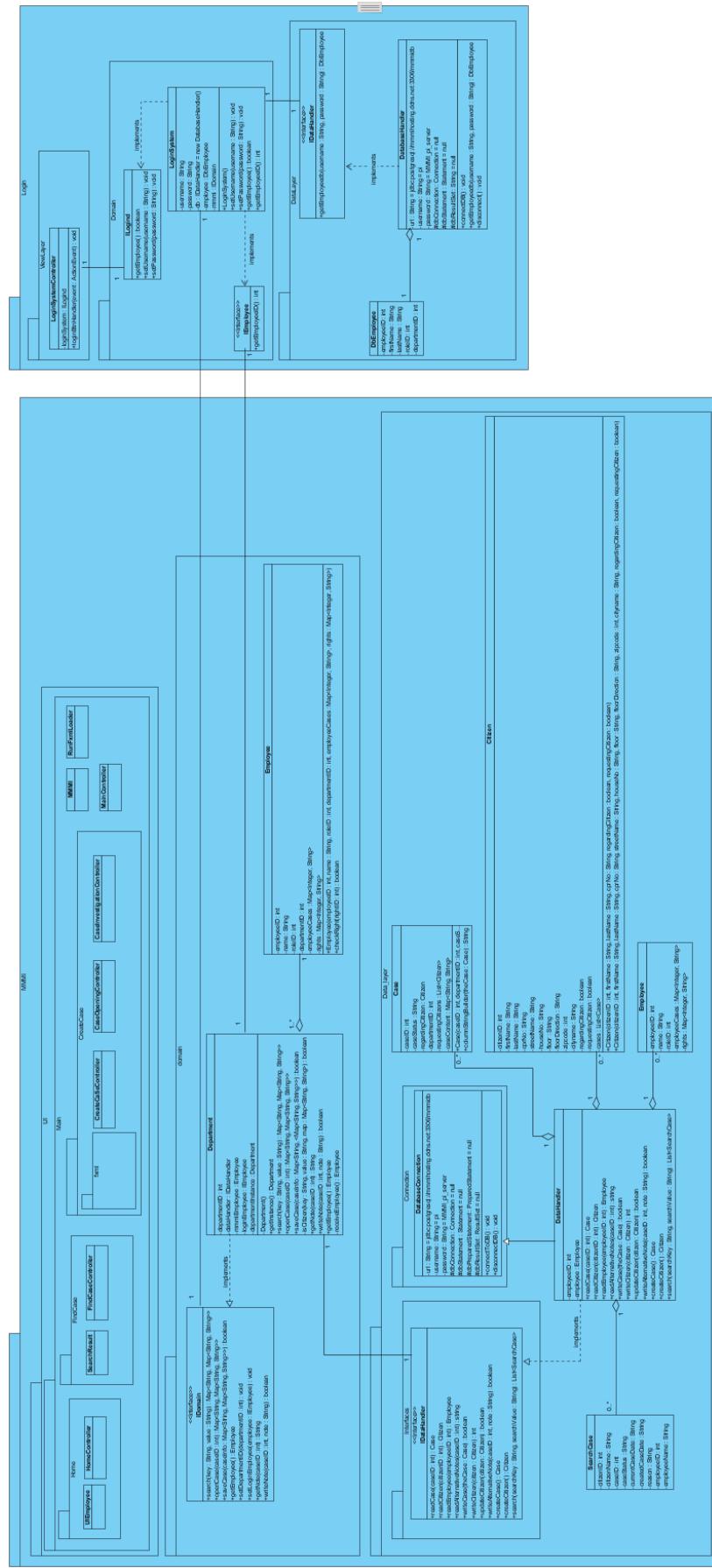
Figur F.24: Sekvensdiagram for behandel sag hvor der er mangel på information.



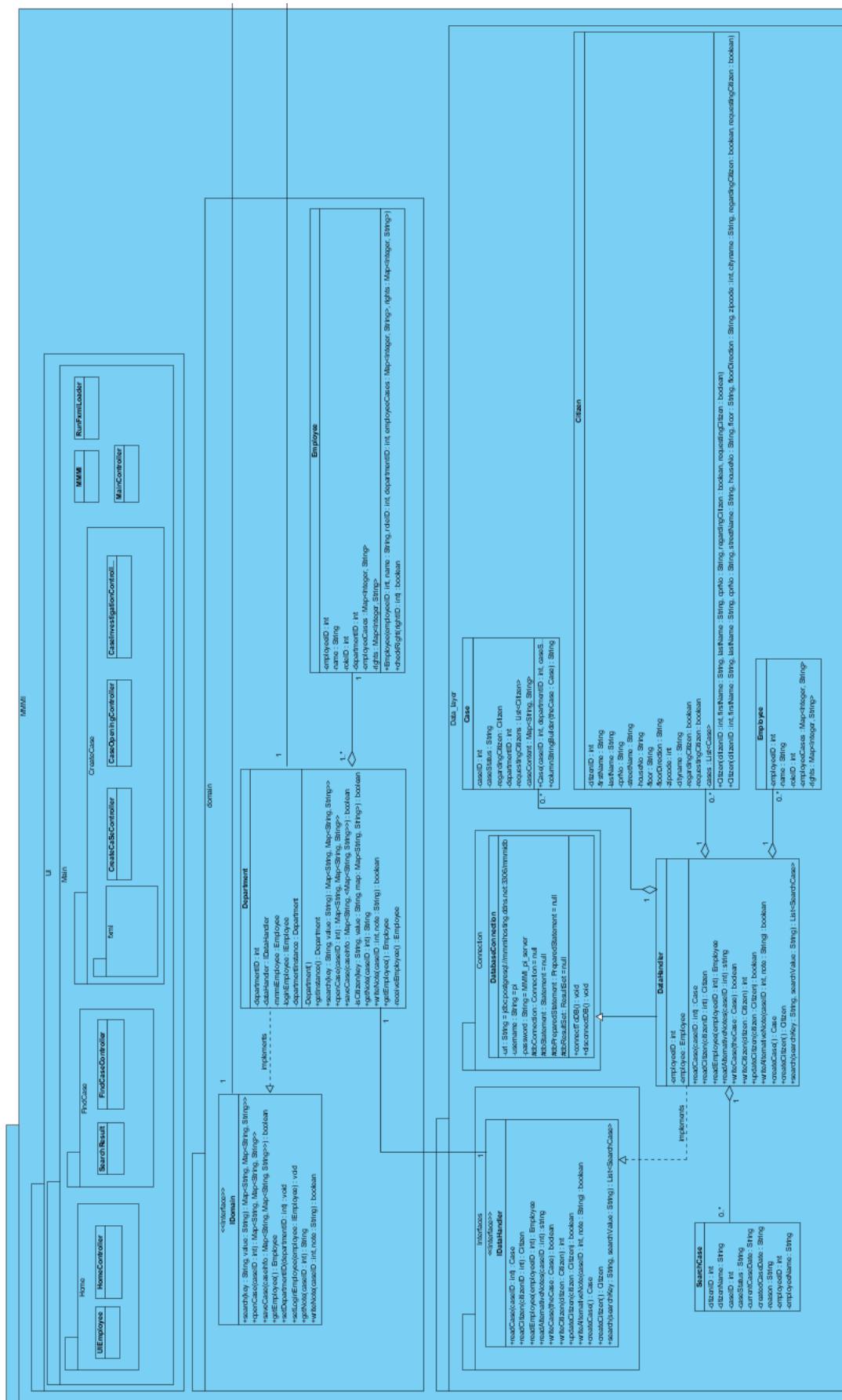
Figur F.25: Sekvensdiagram for behandel sag hvor der er mangel på information.



Figur F.26: Sekvensdiagram for opretsags opdateret fuldstørrelse.

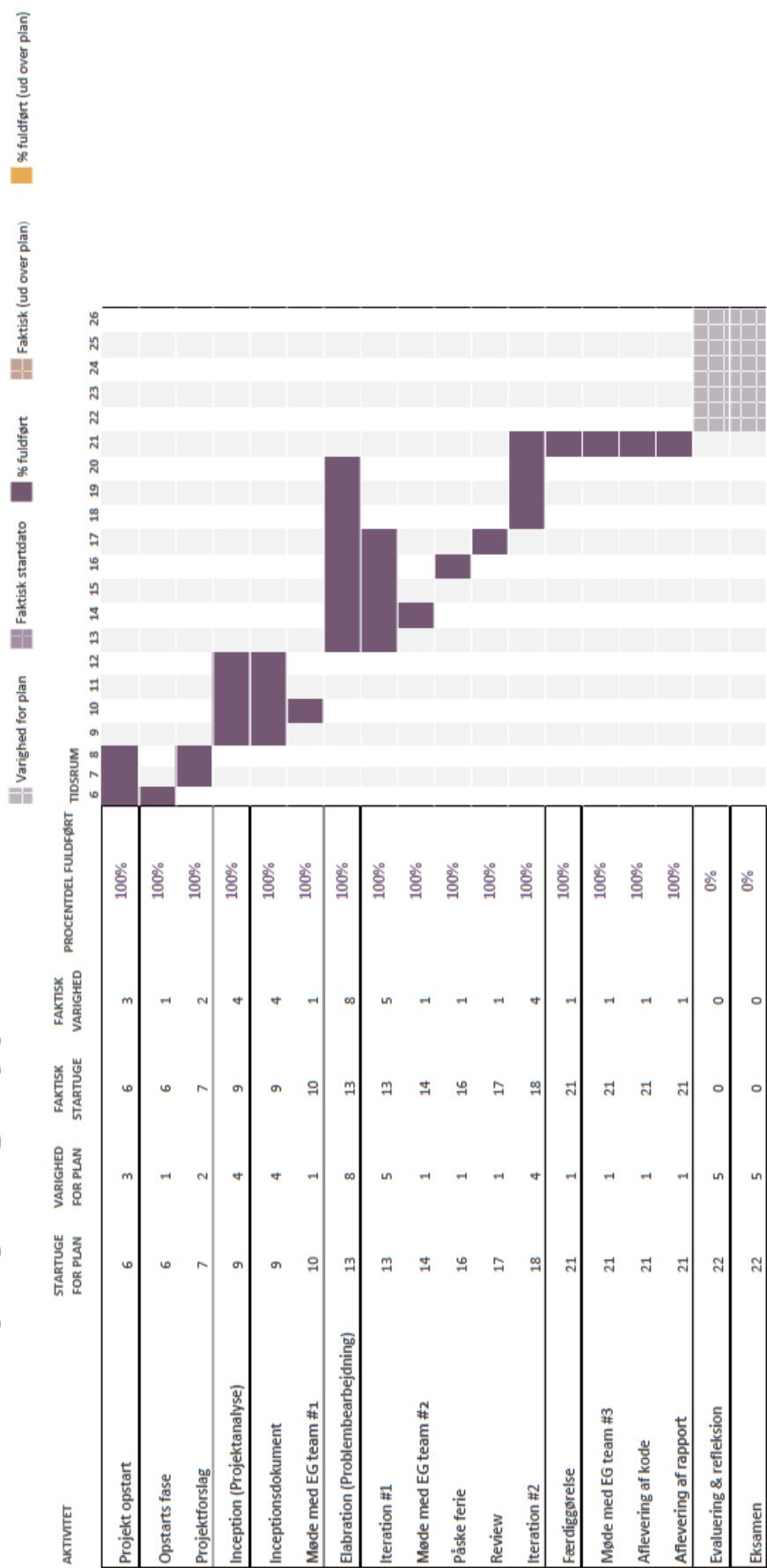


Figur F.27: helt klasse diagram med log in system.

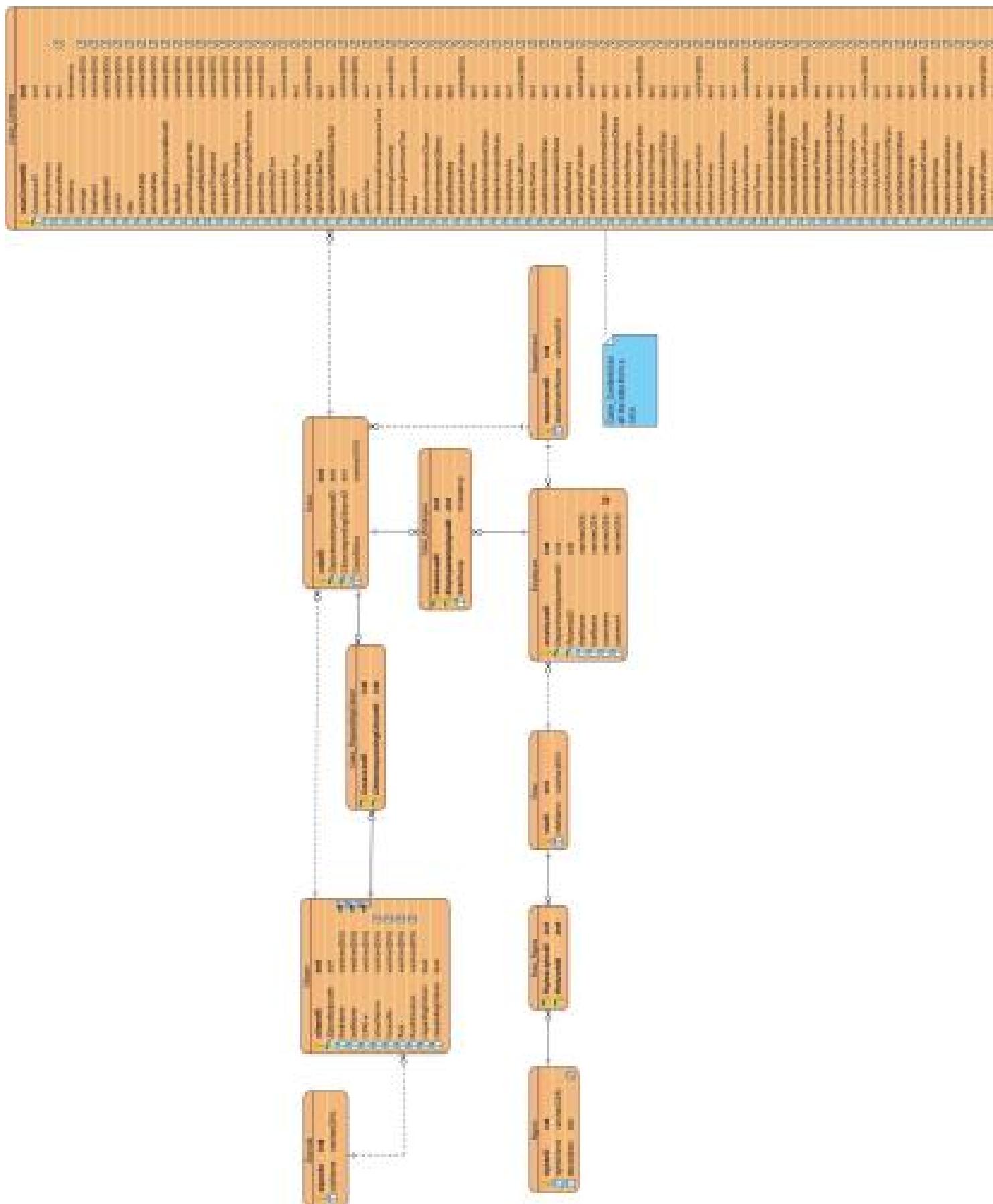


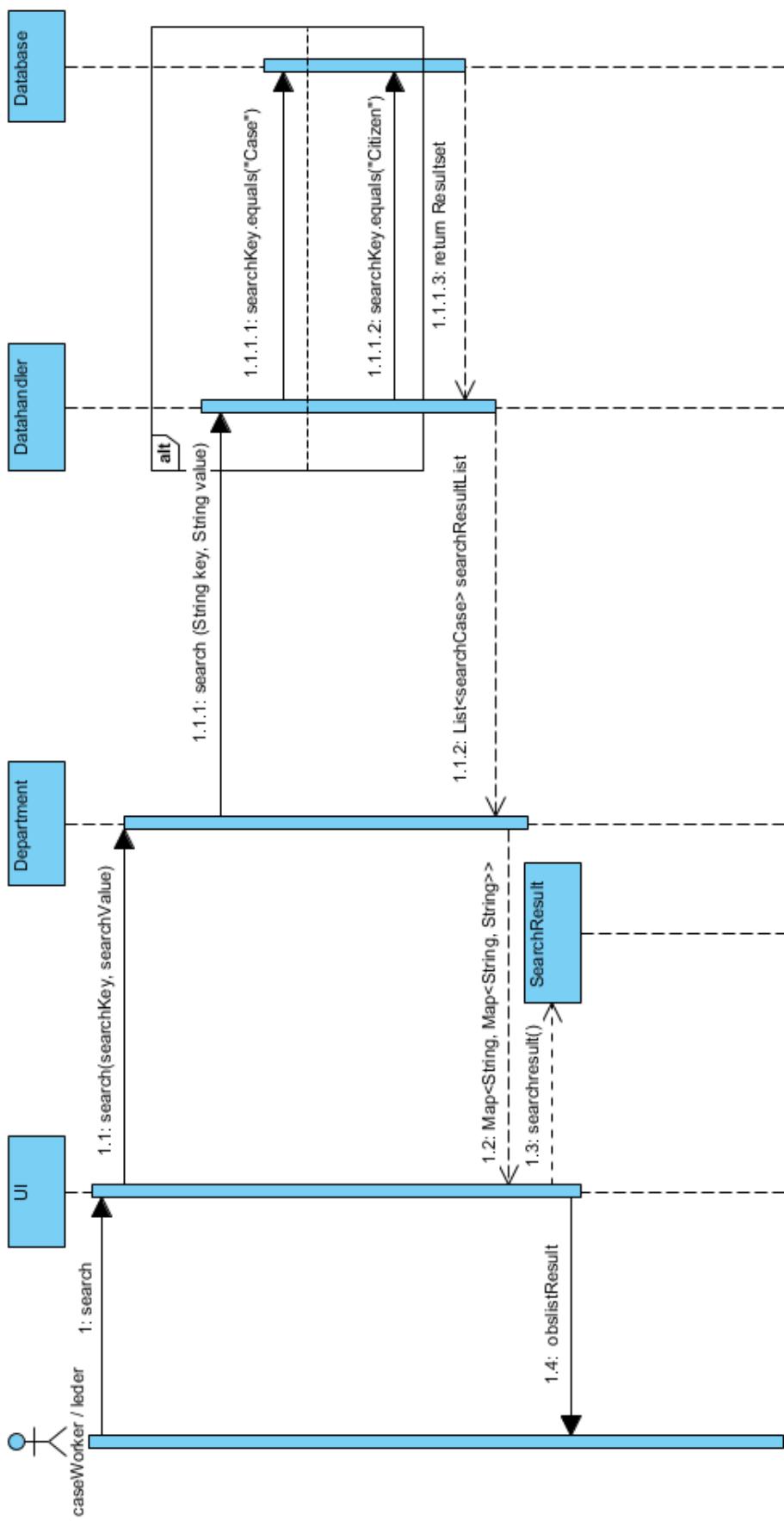
Figur F.28: helt klasse diagram.

# Semester projekt gruppe 02



Figur F.29: Tidsplan fuldstørrelse





Figur F.31: Sekvensdiagram search fuld størrelse

---

```

@Test
public void searchTestCase() {
    IDataHandler searchHandler = new DataHandler();
    String caseNumber = "123";

    List<SearchCase> searchCases = searchHandler.search("Case", caseNumber + "%1");
    Map<String, String> expectedMap = new HashMap<>();

    // Expected output
    expectedMap.put("currentCaseDate", "24/05-2019");
    expectedMap.put("createdCaseDate", "17/05-2019");
    expectedMap.put("caseStatus", "igang");
    expectedMap.put("caseReason", "Henvendelse\u00f8drejer\u00f8sig\u00f8om\u00f8en\u00f8Test");
    expectedMap.put("caseEmployeeName", "Aleksander\u00f8Henriksen");
    expectedMap.put("citizenName", "Poul\u00f8Johanson");

    // searchResult
    Map<String, String> mapContains = new HashMap<>();
    Map searchResultList = new HashMap();
    for (int i = 0; i < searchCases.size(); i++) {
        Map searchResultMap = new HashMap();
        searchResultMap.put("citizenName", searchCases.get(i).getCitizenName());
        searchResultMap.put("currentCaseDate", searchCases.get(i).getCurrentCaseDate());
        searchResultMap.put("createdCaseDate", searchCases.get(i).getCreatedCaseDate());
        searchResultMap.put("caseReason", searchCases.get(i).getReason());
        searchResultMap.put("caseEmployeeName", searchCases.get(i).getEmployeeName());
        searchResultMap.put("caseStatus", searchCases.get(i).getCaseStatus());
        searchResultList.put(searchCases.get(i).getCaseID(), searchResultMap);
    }

    // Test
    assertThat(searchResultMap, Is.is(expectedMap));

    // Print out both maps
    System.out.println("searchResult:" + searchResultMap);
    System.out.println("Expected:" + expectedMap);
}
}

```

---

Figur F.32: Unit test kode for search, når den leder efter en case

---

```

@Test
public void searchTestCitizen() {
    IDataHandler searchHandler = new DataHandler();
    String citizen = "123123-1231%Poul\u00d8Johanson%kimvej%"; // citizen to search for

    List<SearchCase> searchCases = searchHandler.search("Citizen", citizen + "%1");
    Map<String, String> expectedMap = new HashMap<>();

    // Expected output
    expectedMap.put("currentCaseDate", "24/05-2019");
    expectedMap.put("createdCaseDate", "17/05-2019");
    expectedMap.put("caseStatus", "igang");
    expectedMap.put("caseReason", "Henvendelse\u00d8drejer\u00d8sig\u00d8om\u00d8en\u00d8Test");
    expectedMap.put("caseEmployeeName", "Aleksander\u00d8Henriksen");
    expectedMap.put("citizenName", "Poul\u00d8Johanson");

    // searchResult
    Map<String, String> mapContains = new HashMap<>();
    Map searchResultList = new HashMap();
    for (int i = 0; i < searchCases.size(); i++) {
        Map searchResultMap = new HashMap();
        searchResultMap.put("citizenName", searchCases.get(i).getCitizenName());
        searchResultMap.put("currentCaseDate", searchCases.get(i).getCurrentCaseDate());
        searchResultMap.put("createdCaseDate", searchCases.get(i).getCreatedCaseDate());
        searchResultMap.put("caseReason", searchCases.get(i).getReason());
        searchResultMap.put("caseEmployeeName", searchCases.get(i).getEmployeeName());
        searchResultMap.put("caseStatus", searchCases.get(i).getCaseStatus());
        searchResultList.put(searchCases.get(i).getCaseID(), searchResultMap);
    }

    // Test
    assertThat(searchResultMap, Is.is(expectedMap));

    // Print out both maps
    System.out.println("searchResult:" + searchResultMap);
    System.out.println("Expected:" + expectedMap);
}
}

```

---

Figur F.33: Unit test kode for search, når den leder efter en citizen

# G Eksterne bilag

## 1 Case

### Samlet Social Organisation

#### En case af EG Team Online ApS

##### Indledende tekst

Efter salget af KMD i 2008 har kommunerne og Kommunerne IT-fællesskab KOMBIT<sup>1</sup> arbejdet for at sikre kommunernes indflydelse på og ejerskab til løsningerne på de områder, hvor KMD i mange år har haft monopol på at tilbyde løsninger.

I økonomiaftalen for 2013 fremgår det, at it-løsningerne for områderne kontanthjælp, sygedagpenge og sags- og partsoverblik bedst udvikles som landsdækkende løsninger, hvor alle kommuner står bag udbuddene og aftager løsninger.

Projekterne i monopolbruddet skal frigøre kommunerne fra KMD's monopol på it-løsningerne på kontanthjælp, sygedagpenge og sagsoverblik. Desuden har tværgående støttesystemer til formål at skabe konkurrence på fremtidige kommunale it-systemer.

EG Team Online blev stiftet i 1998 som et meget lille softwarefirma der leverede løsninger på bestillingsbasis. Oprindeligt startede firmaet op under navnet One2One og fokuserede deres tid på at lave små web-baserede løsninger som f.eks. CMS eller CRM-løsninger, der blev bestilt af diverse kunder.

EG Team Online har siden stiftelsen fulgt med udviklingen på velfærdsområdet med udgangspunkt i den sociale sektor og har herefter udviklet sig til at blive en innovativ teknologileverandør med speciale i løsninger, der understøtter det samlede behov i den offentlige og private social- og sundhedssektor.

I 2005 leverede de deres første store system til den sociale sektor under navnet Bosted System. Systemet har senere hen skiftet navn til Sensus Bosted. Systemet var revolutionerende for lokale bosteder og institutioner som arbejder med patientbehandling til daglig og derfor normalt var tvunget til at holde journaler og medicin-styring på papir.

I 2014 blev firmaet opkøbt af EG, tidligere kendt som EDB Gruppen, der beskæftiger mere end 2000 ansatte i Skandinavien. Fra denne dato skiftede virksomheden navn til EG Team Online.

EG Team Online har en dyb, faglig forståelse for de udfordringer, som bosteder og sagsbehandlere har i hverdagen, som virksomheden hidtil har tilbudt en løsning indenfor med det socialpædagogiske it-værktøj, Sensus Bosted.

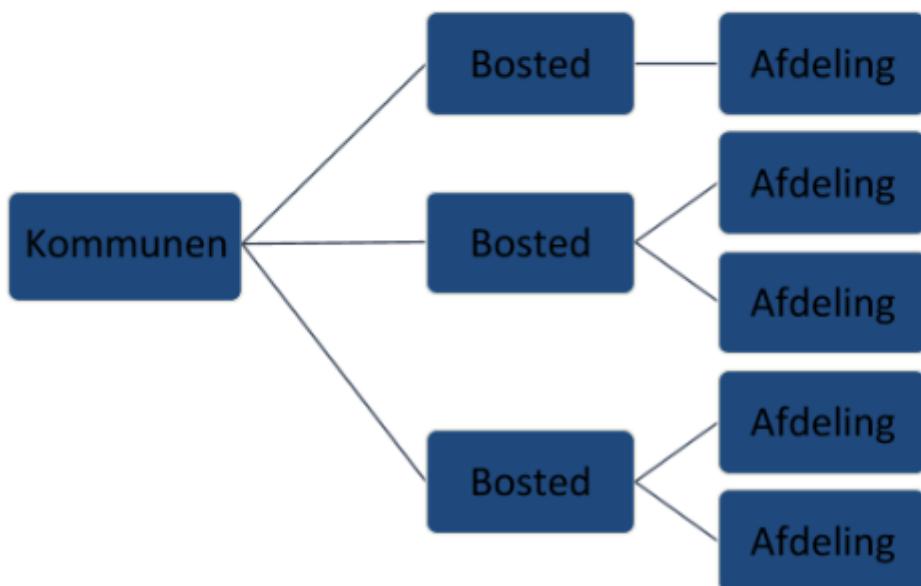
Med sags og udredningssystemet EG Sensus samt det socialpædagogiske it-værktøj, EG Sensus Bosted, har virksomheden de sidste 15 år været førende indenfor hele det specialiserede socialområde. Men tiderne skifter, de sociale institutioner bliver større og større, og kravene til dataafgrænsning og sikkerhed er stigende. Hvor der før var separate systemer til hver enkelt institution, ønsker man i forbindelse med sammenlægninger af institutioner at samle systemerne, hvilket bl.a. er en udfordring mht. dataafgrænsninger. Man må som socialpædagog på et botilbud ikke se data fra et andet botilbud. Ligesom man som socialrådgiver i voksen-handicap-området, ikke må se misbrugssager og omvendt. Brugere kan også have flere roller inde for samme organisation, fx kan en bruger være fast socialpædagog på det ene

<sup>1</sup> Kommunerne IT-fællesskab <https://www.kombit.dk/indhold/om-kombit>

botilbud, og være vikar på et andet. Rettigheder er ikke det samme, og det vil derfor være forskellige kontekster, at systemet anvendes i.

## Det socialpædagogiske område

Det socialpædagogiske område kan sammenlignes med en større organisation med forskellige områder og afdelinger. Alle afdelinger skal rapportere enten op eller ned i et vist omfang og i det kommunale er selve kommunen i toppen af fødekæden, som skitseret nedenfor:



Bostederne tager sig primært af patientbehandling. En borger har en udfordring (fysisk eller psykisk handicap eller et stofmisbrug), så hvordan kommer vi fra A til B med denne udfordring, så den kan mindskes og dermed gøre borgerens liv nemmere og bedre. Sensum Bosted tilbyder dokumentationsværktøjer, så alle trin i denne proces kan blive dokumenteret på den mindst tidskrævende måde – det handler om at skabe mere tid mellem borgeren og den ansatte som hjælper dem.

Mens bostederne tager sig af patientbehandlingen, så er sagsbehandlingen kommunens ansvar. Når borgeren møder op hos kommunen med en udfordring som f.eks. et handicap eller en kronisk lidelse, så er det kommunens ansvar at lave en udredning, dvs. at finde ud af hvad problemet er og hvordan borgeren kan blive hjulpet, og hvordan betalingen kan blive dækket. Som eksempel kan løsningen være en behandling på et bosted.

## Produkterne

### Sensum Bosted

Sensum Bosted fokuserer på at digitalisere de daglige arbejdsprocesser på bostederne, der tidligere normalt foregik på papir. Systemet erstatter kinabøger og reoler fyldt med ringbind, hvor de daglige observationer og handleplaner hidtil har været dokumenteret.

Sensum Bosted understøtter dokumentationsbehovene i alle former for specialiserede tilbud på det sociale område. Systemet sætter borgeren i centrum og sikrer, at medarbejderne har adgang til alle nødvendige oplysninger døgnet rundt. Sensum Bosted er udviklet målrettet i samarbejde med socialområdets aktører til såvel små som store sociale tilbud på hele det specialiserede socialområde og systemet understøtter hele det daglige arbejde med dokumentation, effektmåling og økonomistyring.

Sensum Bosted er en modulopbygget plug-and-play samlekasse, hvilket gør det muligt at koble moduler på efter kundens behov.



## Sensum

Kommunens behov er at have overblik over processen fra en borger møder op på kommunen til borgeren er færdig med en behandling. Kommunen har derfor brug for at få organiseret informationerne fra bostederne på en overskuelig måde.

Sensum er et sags- og udredningssystem, der skaber dette overblik over sagsbehandlingen og faciliterer kommunens arbejdsprocesser. Sensum er et såkaldt DHUV-system, hvor DHUV står for "Digitalisering af Handicap og Udsatte Voksne".<sup>2</sup>

Systemet understøtter arbejdsflowet fra borger til sag og herefter til behandling. Et sagsforløb består af en række aktiviteter, som f.eks. sagsåbning, sagsoplysning, sagsvurdering, afgørelse og valg af social indsats samt sagsopfølgningen. For DHUV er denne proces beskrevet af sagsmodellen for voksenudredning (VUM<sup>3</sup>).

I forbindelse med sagsopfølgningen er det vigtigt, at sagsbehandleren har dokumenteret sagsforløbet ved hjælp af dagbogsnotater koblet på sagen. Et dagbogsnotat vil blive oprettet på alle møder og samtaler

<sup>2</sup> Læs mere her: <https://socialstyrelsen.dk/tvaergaende-områder/sagsbehandling/vum-og-dhuv>

<sup>3</sup>1. Også: GDPR General Data Protection Regulation

foretaget med borgeren og eventuelle private kommentarer fra sagsbehandleren. Derudover vil et dagbogsnotat også typisk indeholde vedhæftninger af eventuelle formularer eller breve til og fra borgeren.

Systemet er ligesom Sensus Bosted bygget op som en modulopbygget plug-and-play samlekasse.

### Databeskyttelsesforordningen

Kommunerne har nogle meget høje krav til datasikkerhed. Den 27. april 2016 vedtog Europa-Parlamentet en ny EU-databeskyttelsesforordning (ofte kaldet Persondataforordningen<sup>1</sup>) som har til formål at styrke beskyttelsen af personoplysninger i EU. Forordningen strammer kravene omkring hvem, hvad og hvordan personfølsomme data må tilgås. Dette betyder blandt andet at borgeren til enhver tid kan kræve at få slettet eller vist sine data, samt se hvem der har tilgået/ændret i disse.

### Overordnet problemstilling

I forbindelse med de nye krav til understøttelse af store organisationer med flere botilbud og socialrådgivning under samme hat, sammenholdt med større krav til databeskyttelse (dataafgrænsning og sikkerhed) og borgerinvolvering vil EG Team Online gerne have belyst, hvordan man kan samle systemerne under en hat, og samtidigt sikre, at man kun ser data for lige præcis det, man har behov for, samt sikre at man kun ser og opbevarer data, som borgere eller pårørende har afgivet samtykke til. Belysningen foregår ved at udvikle forslag til løsninger til et eller flere moduler som anvendes på tværs af organisationen. De Moduler der ønskes belyst er Dagbog, Sagsudredning og Planlægning.

### Dagbog

Dagbog anvendes til journalføring på borgeren. Dette sker både under udredningen af borgeren og til dagligdagsnotater ude på botilbuddene. Nogle notater genereres automatisk af systemet, fx når borgeren er tildelt en ydelse, eller når borgeren får udleveret medicin. Andre skrives i hånden af medarbejderen selv. Dagbogsnotater oprettes i kontekst af brugeren, der er logget ind. Så man ved, hvor det hører til.

#### Dataafgrænsning (tilvalg):

I forhold til dataafgrænsning af dagbog, er det vigtigt, at en bruger, fx en sagsbehandler, kun kan se de notater, der er relevante for brugeren selv. Fx hvis en borger er indskrevet som misbruger på et misbrugscenter og ligeledes indskrevet på en voksenhandicap-afdeling, så må de to enheder ikke kunne se dagbogsnotater på tværs.

## Sagsudredning

I forbindelse med at en borger bliver udredt og tildelt en ydelse, som fx behandling på et misbrugscenter eller ophold på et botilbud, udarbejdes der en sag efter VUM-metoden. Denne registreres i systemet som en Sag. Sager oprettes efter en given paragraf, der fortæller hvilken type sag det drejer sig om.

*Dataafgrænsning (tilvalg):*

Man må som sagsbehandler i socialafdelingen ikke kunne se sager på borgere, der optræder i misbrugsafdelingen. Altså må en sagsbehandler ikke blive vidende om, at der i forvejen ligger en stof- eller alkoholmisbrugssag på denne borger.

## Planlægning

En borger der er indskrevet, vil have en række aktiviteter. Det kan fx være at tage tøj på om morgenen, på udflugt med afdelingen osv. Aktiviteterne illustreres med pictogrammer, som er en tegning af hvad aktiviteten går ud på, således at borgere der ikke kan læse også kan forstå planen. Planen skal kunne udskrives, så aktiviteterne for dagen med pictogrammer kan hænges op i borgerens værelse.

*Dataafgrænsning (tilvalg):*

Man skal kunne gøre aktiviteter private for uvedkommende. Det vil sige at kun særligt udvalgte vil kunne se, hvad aktiviteten går ud på. Det kan fx være hvis en borger, der i forvejen er på et botilbud skal til udredning for stofmisbrug, så er måske kun borgerens egen kontaktperson, der må vide det og kan se det i aktivitetskalenderen.

Områderne er nærmere beskrevet i Voksenudredningsmetoden (VUM) på socialstyrelsens hjemmeside <https://socialstyrelsen.dk/tvaergaende-områder/sagsbehandling/vum-og-dhuv>.