
Forbedring i udarbejdelse af semesterevalueringsrapporter

Udvikling af applikationer – fra brugere til data, algoritmer og
test – og tilbage igen

ds321e18

Efterår 2018
Projektrapport P3

Aalborg Universitet
Selma Lagerlöfs Vej 300
DK-9220 Aalborg Øst



Institut for Datalogi
Selma Lagerløfs Vej 300
DK-9220 Aalborg Øst
<http://aau.dk>

AALBORG UNIVERSITET

STUDENTERRAPPORT

Titel:

Forbedring i udarbejdelse af semesterevalueringssrapporter

Tema:

Udvikling af applikationer – fra brugere til data, algoritmer og test – og tilbage igen

Projektperiode:

3. semester

Projektgruppe:

ds321e18

Deltager(e):

Christian Graae Zandersen
Jesper Adriaan van Diepen
Mike Lund Andersen
Nanna Søgaard Andersen
Niki Zakariassen
Rasmus Lindegaard
Simon Steiner

Vejleder(e):

Jane Billestrup

Oplagstal: 1**Sidetal:** 212

Normalsider: 97 (ekskl. bilag)

Afleveringsdato:

20. december 2018

Abstract:

The number of evaluations that a semester coordinator must consider is rising and can be too large to give a general representation of the semester.

By developing a tool, which can help the semester coordinator, it is easier to determine whether the students were overall happy or displeased with the semester or the specific course. The tool analyzes the semester evaluations by each word and in two separate ways determines the sentiment of a comment.

One of the main priorities was also to be able to make the tool welcoming for users and helpful to use so that it was used by most semester coordinators. To this purpose, a graphical user interface was developed over multiple iterations in contact with informants to make sure the layout was understandable.

Results showed, that it was indeed possible to accomplish the aforementioned functionalities. This result should make it possible for coordinators to use the tool if it's in their interest.

Indholdsfortegnelse

1	Indledning	1
I	Systemvalg	3
2	Metodologi	4
2.1	Systemudviklingsmetoder	4
2.2	Interview	7
3	PACT-analyse	11
3.1	People	12
3.2	Activities	13
3.3	Contexts	15
3.4	Technologies	16
3.5	Opsummering	17
4	Systemdefinition	18
4.1	Problemformulering	21
5	Kravspecifikation	22
5.1	Kvalitetsfaktorer	22
5.2	De specifikke krav	24
5.3	Afrunding	27
II	Modellering og Design	28
6	Analyse af problemområdet	29
6.1	Klasser	29
6.2	Struktur	33
6.3	Adfærd	35
6.4	Afrunding	39

7	Analyse af anvendelsesområde	40
7.1	Anvendelse	40
7.2	Funktioner	42
7.3	Afrunding	45
8	Arkitekturdesign	47
8.1	Kriterier	47
8.2	Komponenter	55
8.3	Opsummering	61
9	Komponentdesign	62
9.1	Modelkomponent	62
9.2	Funktionskomponent	65
9.3	Afrunding	66
10	Brugergrænsefladedesign	68
10.1	Prototype	68
10.2	Design	70
10.3	Prototypetest og -evaluering	75
10.4	Opsummering	82
III	Implementering	83
11	Data	84
11.1	Dataindsamling	84
11.2	Forbehandling	86
11.3	Database	86
11.4	Afrunding	88
12	Statistisk tilgang - Sentimentklassificering	90
12.1	Nøgleordsanalyse	90
12.2	Term Frequency	90
12.3	Inverse Document Frequency	91
12.4	Term Frequency-Inverse Document Frequency	91
12.5	Nøgleord som sentimentklassifikator	92
12.6	Afrunding	94
13	Maskinlæringstilgang - Sentimentklassificering	95
13.1	Neurale Netværk	95
13.2	Word Embedding	97
13.3	Netværksstruktur	99
13.4	Netværket som sentimentklassifikator.	100
13.5	Opsummering	102

14 Optimering af lineær algebra	103
14.1 Multiplikation	103
14.2 Matrix Data Struktur	104
14.3 Parallelisering	106
14.4 Lokalitet	107
14.5 Optimeringsbetydning	108
14.6 Afrunding	109
15 Grafisk Brugergrænseflade	110
15.1 Login	110
15.2 Sidebar	110
15.3 Valg og oprettelse af semestre	111
15.4 Sammenligning	113
15.5 Markering af kommentarer	114
15.6 Rådata	114
15.7 Kommentarvisning	115
15.8 Afrunding	116
IV Kvalitetssikring	117
16 Validering	118
16.1 Cross validation	118
16.2 Nøgleordsanalyse	119
16.3 Neuralt netværk	122
16.4 Opsummering	127
17 Unit Testing	128
17.1 Teori	128
17.2 Anvendelse af tests	129
17.3 Dækningsgrad af unit-tests	131
17.4 Afrunding	131
18 Usability Testing	132
18.1 Usability	132
18.2 Testplan	132
18.3 Testresultater	135
18.4 Afrunding	138
V Refleksion	139
19 Redesign	140
19.1 Løsninger	140
19.2 Afrunding	142

20 Diskussion	143
20.1 Kravspecifikation	143
20.2 Tilgang til projektudvikling	145
20.3 Fremtidig Udvikling	147
21 Konklusion	149
VI Bilag	154
A Interview-spørgsmål	155
B Anvendelse af de 7 trin	156
C Transkriberede interviews af informanter	159
C.1 Interview af Informant 3	188
D Funktionelle Krav	203
E Stopord	208
F Usability test opgaver	210
G Brugervejledning	211

Kapitel 1

Indledning

På Aalborg Universitet (AAU) forventes det, at alle studerende udfylder en semesterevaluering i slutningen af hvert semester. Denne skal indeholde kommentarer og tilfredshedsniveau for semestrets aktiviteter. Ydermere vurderes den studerendes oplevelse af det fysiske, psykiske og æstetiske studiemiljø. [1]

Spørgeskemaerne, som de studerende udfylder, er lavet efter et standardiseret format, hvilket betyder, at de kan sammenlignes med svar fra tidligere semestres evalueringer. Evalueringerne indeholder både kvantitative og kvalitative evalueringsmetoder [1], da de kvantitative multiple-choice spørgsmål giver en klar tilfredshedsmåling på de enkelte områder, hvorimod de mere uddybende kvalitative tekstboksse tillader de studerende at begrunde deres tilfreds- eller utilfredshed.

Antallet af studerende på AAU har i en lang årrække været stigende, og der er flere studerende end der nogensinde har været. I 1980 var der omkring 2000 studerende, i år 2000 omkring 10500 studerende og i 2018 omkring 20700 studerende [2][3]. Denne stigning i antal studerende betyder, at mængden af evalueringer også stiger, og dermed også det manuelle arbejde der bliver brugt på at læse og bearbejde dem.

Alle semestre er tilknyttet en semesterkoordinator som blandt andet har til opgave at overvære semestrets gang, og sikre at de beskrevne læringsmål bliver opfyldt. Den enkelte semesterkoordinator for et studie, skal på nuværende tidspunkt manuelt gennemgå de studerendes evalueringer af hvert semester. På baggrund af disse evalueringer skal der udarbejdes en semesterevalueringsrapport, som har til formål at beskrive de studerendes generelle holdning til semestret, samt give forslag til mulige forbedringer. Denne proces kræver i stigende grad mere af semesterkoordinatorerne, eftersom der bliver større mængder data at læse og vurdere vigtigheden af. Flere evalueringer betyder også at der kommer flere forskellige holdninger, det bliver derfor også et større stykke arbejde at gøre rapporten repræsentativ for den generelle holdning.

Store dele af arbejdet med besvarelsene indebærer at gennemgå og analysere de kvalitative kommentarer. De kvantitative multiple-choice besvarelser giver et godt overblik over den generelle holdning, men for at finde de underliggende årsager til denne holdning, er det nødvendigt at dykke ned i de enkelte kommentarer.

Udarbejdelsen af evalueringsrapporten har på baggrund af den ovenstående beskrivelse, potentiale til at blive forbedret. Dette indebærer både at forbedre dens kvalitet, samt nedsætte arbejdstiden der bliver brugt på den. Heraf udledes det initierende problem:

*Mængden af evalueringer en semesterkoordinator skal tage hensyn til er stigende og kan være for stor til at give en generaliseret repræsentation af holdningen til semestret. **Kan der laves et system på baggrund af dette, så semesterkoordinatoren får et mere overskueligt overblik over evalueringerne og dermed kan give en bedre repræsentation af semesterevalueringerne?***

Del I

Systemvalg

Kapitel 2

Metodologi

Dette kapitel vil give et grundlæggende overblik over de metoder, der gøres brug af i projektet til systemudvikling og dataindsamling. Deres styrker og svagheder vurderes i forhold til udarbejdelsen af projektet.

2.1 Systemudviklingsmetoder

Systemudvikling betegner processen fra analyse af et problem, til design af en løsning, implementering af denne og til sidst en evaluering af løsninger [4]. Dette foregår ved en implicit eller eksplicit gennemgang af en række faser, hver med sin rolle i udformningen af et system [5].

Den første fase består af informationsindsamling og analyser, til udformning af gode *systemkrav*. Dette består blandt andet af en undersøgelse af de potentielle brugere, interessenter, eksisterende systemer og eksterne indflydelser til det system, der skal udvikles. Ud fra det, udformes konkrete systemkrav til at lægge grundlag for hvad alle de senere faser skal opnå.

Det næste trin er *design* af et system, der kan leve op til de opstillede krav. Her dannes et overblik over hvordan systemet skal struktureres, samt hvilken platform og værktøjer, der skal tages i brug. Med det, kortlægges det hvordan funktionaliteten skal implementeres, og de nødvendige beregninger og algoritmer laves.

Med et design udviklet, er det næste *implementering* af dette design. Dette er selve kodningen af systemet.

Når systemet når et funktionelt stadie, skal det igennem *verificering* af hvorvidt det lever op til de systemkrav, som det blev udviklet til at opfylde. Dette består af en cyklus af tests og debugging, som alt efter projektets natur kan bestå af små rettelser til revurdering af hele delsystemer.

2.1. Systemudviklingsmetoder

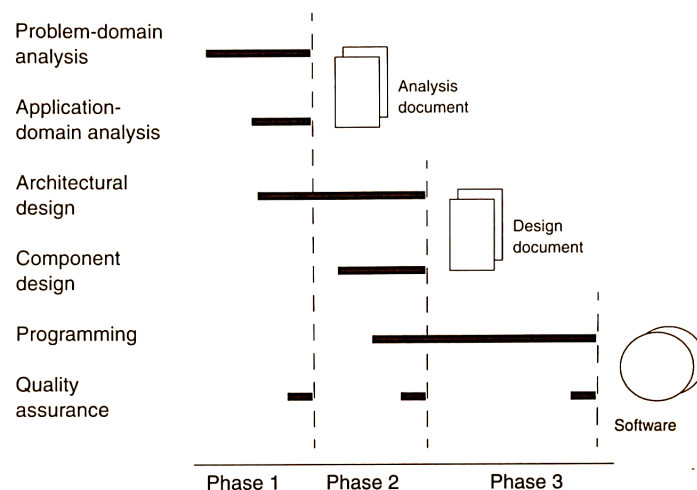
Et system der opfylder sine systemkrav, kan siges at være færdigt, men det kan også føres igennem videre *vedligeholdelse*, til at lave optimeringer og rette op på eventuelle svagheder, for at forbedre systemet yderligere.

Der er to overordnede metoder til at håndtere projektudviklingens strategi og livscyklus; vandfaldsmetoden og iterationsmetoden. [6]

2.1.1 Vandfaldsmetoden

Vandfaldsmetoden, også henvist til som den traditionelle metode, består af en sekventiel gennemgang af de primære OOA&D aktiviteter. Hver fase af udviklingsprocessen arbejdes på enkeltvist, og som det fremgår af Figur 2.1 er der en fast strategi for projektudviklingen.

Denne traditionelle metode virker bedst på projekter med en meget klar vej frem, da der ikke er fleksibilitet til at lave store ændringer undervejs [7]. Når det ikke er tilfældet risikeres der at der i de senere dele af processen findes problemer, som er forårsaget af fejl tidligere i processen. Der kan også være omstændigheder der skifter, som kan besværliggøre det videregående arbejde og true kvaliteten af det endelige produkt. For at undgå dette så vidt muligt, er processen opdelt i 3 faser som hver især slutter med en kvalitetssikring, der skal tjekke at projektet er klar til at bevæge sig fremad. Hvis dette ikke er tilfældet, er der stadig mulighed for at gå tilbage og rette i de dele af projektet der er blevet arbejdet på i den givne fase. [6]



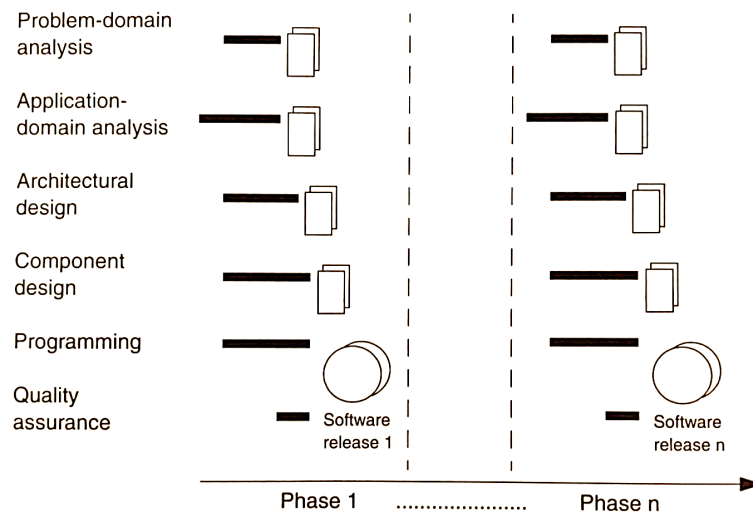
Figur 2.1: Illustration af vandfaldsmetoden [6]

2.1. Systemudviklingsmetoder

2.1.2 Iterationsmetoden

Den iterative metode beskæftiger sig med de samme aktiviteter som vandfaldsmetoden, men i stedet for at beskæftige sig med faserne lineært og systematisk, arbejdes der i iterationer. Dette vil sige at arbejdsprocessen bevæger sig samlet og hurtigt igennem alle aktiviteter med henblik på et meget lille og basalt delprodukt, for efterfølgende at gentage processen med udvidelser og forbedringer af delproduktet. Dette gentages et antal faser, indtil der rammes en begrænsende faktor, i form af tid, budget eller tilfredshed. [6]

Denne metode giver et projekt betydelig mere fleksibilitet, da der konstant revurderes tidligere analyser og designvalg. Metoden har en cyklisk natur, men det forventes dog stadig at den underliggende udvikling er lineær og fremadgående. En risiko ved at bruge iterationsmetoden er at blive fanget i et projekt, eftersom det ofte vil være nærliggende at tilføje én fase mere. [8]



Figur 2.2: Illustration af den iterative metode [6]

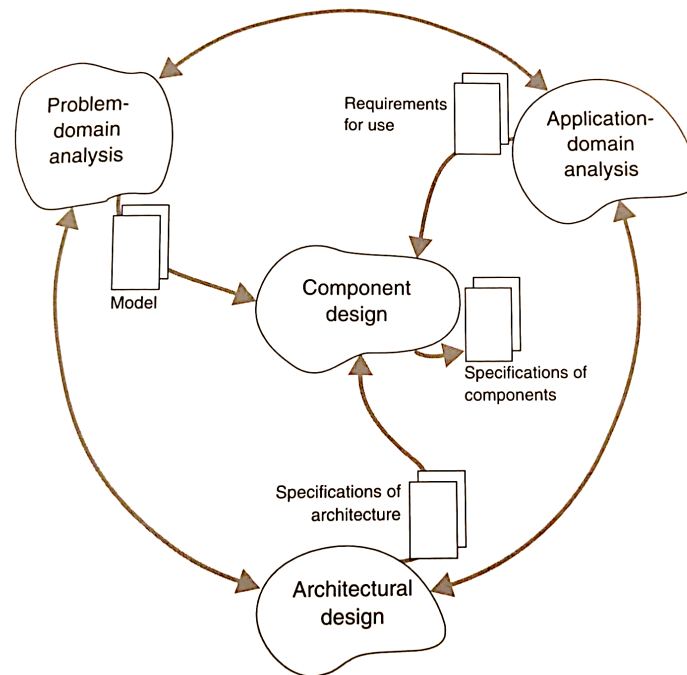
2.1.3 Valg af projektudviklingens livscyklus

Til dette projekt vælges Vandfaldsmetoden, eftersom projektet er pålagt en fast deadline i december. Den største risiko ved at anvende den iterative metode er, at dele af projektet aldrig bliver færdige, da det altid vil være nærliggende at tilføje en ekstra fase. Dette kan forårsage at projektet ikke bliver færdigt til den ønskede deadline. Desuden afholdes et kursus sideliggende omkring systemudvikling, som gennemgår metoderne i takt med hvornår vandfaldsmetoden anvender dem. Den iterative metode er derfor mere oplagt, hvis gruppemedlemmerne i forvejen kender

2.2. Interview

til alle aktiviteterne, da de hver især bliver gennemgået mange gange.

Sammenhængen mellem disse overordnede aktiviteter beskrives i Figur 2.3, hvor det fremgår at produktet af denne proces burde være en specifikation af komponenter i komponentdesignet.



Figur 2.3: Illustration af de primære aktiviteter i OOA&D [6]

2.2 Interview

Projektet laves med henblik på, at analysere og løse et problem for en specifik bruger-gruppe. Interviewet virker dermed som projektets primære metode til at forstå brugerens nuværende arbejdsproces, og tilhørende problemstillinger. Interviewet skal bruges, som et værktøj til at forstå brugerens tanker, følelser og overvejelser omkring det valgte problem, og på den måde, specifikt få viden omkring problem- og anvendelsesområdet for systemet der skal udvikles. Afsnittet vil beskrive forskellige interviewmetoder, samt en måde at strukturere processen omkring et interview, med udgangspunkt i Jens Thistedes beskrivelse af denne proces. [9].

2.2.1 Det strukturerede interview

Det strukturerede interview kendetegnes ved, at det følger en fast form med spørgsmål og struktur. Spørgsmålene vil i høj grad være lukkede og fremprovokere præcise og

2.2. Interview

korte svar.

Fordelene ved dette er at interviewerne nemt kan sammenligne svarene, og dermed analysere forskellige meninger. Ydermere er det et nemt interview at føre, eftersom de forudbestemte spørgsmål lægger et fundament for interviewet.

En af de potentielle problemer med denne type interview er, at det kan være svært at få det fulde billede af den interviewedes holdning. Det er ikke muligt at stille opfølgende spørgsmål, og derfor er det ikke muligt at få viden, der ligger udenfor de forudbestemte spørgsmål.

2.2.2 Det ustrukturerede interview

Det ustrukturerede interview har sjældent nogen form for forudbestemt struktur eller interview-guide til at styre interviewet. Det vil i høj grad være præget af åbne spørgsmål, hvor intervieweren stiller mange opfølgende spørgsmål på baggrund af de svar, som bliver givet.

En af fordelene ved denne type interview er at det ofte virker meget uformelt og minder om en hverdagssamtale, hvilket kan tillade en følelse af sikkerhed hos interviewpersonen. Intervieweren får mulighed for frit at føre interviewet i den retning, der virker spændende eller relevant.

En af ulemperne ved denne metode er dog at det på samme tid kan gøre det svært for intervieweren at føre interviewet på en effektiv måde, uden at blive ledt ud af et sidespor. Denne type interview lider under, at det kan være rigtig svært at analysere dataen og sammenligne to interviews med hinanden, eftersom de omtalte emner kan variere meget.

2.2.3 Det semistrukturerede interview

Det semistrukturerede interview tillader et kompromis mellem de to ovenstående interviewmetoder. Det nedarver dele fra begge metoder. Intervieweren har en generel idé om hvilke områder, der vil blive snakket om, inden interviewet går i gang, og bruger åbne spørgsmål for at lade den interviewede svare bredt. Strukturen tillader intervieweren at dykke mere ned i emner, som virker relevante på baggrund af den interviewedes svar, ved at stille opfølgende spørgsmål på de svar der bliver givet. Intervieweren har derfor potentiale til at kunne indsamle store mængder brugbar information, hvis der stilles de rigtige opfølgende spørgsmål, og det gøres på de rigtige tidspunkter. En af ulemperne ved denne interviewmetode er dog, at det stadig kan være svært at trække klare linjer mellem temaer for de forskellige interviews, eftersom svarene stadig er relativt åbne, og de opfølgende spørgsmål kan variere på baggrund af svarene, som den interviewede giver.

2.2.4 Valg af interview-stuktur

Brugergruppen for projektet er ikke særlig bred, eftersom det primært vil være semesterkoordinatorer det retter sig imod. For at få mest mulig brugbar information ud af de tre informanter vi har opnået kontakt med, vælger vi at bruge det semistrukturerede interview. Det tillader at vi kan forberede spørgsmål, som skal holde rammerne for interviewet, men stadig spørge frit. Vi kan også ændre spørgsmålene fra det ene interview til det næste, eftersom det alligevel er svært at trække klare linjer mellem interviewene. Dette gør det desværre også sværere at sammenligne de forskellige interviews på tværs af hinanden, eftersom ingen to interviews vil holde den samme struktur.

Gruppen vurderer at gevinsten ved den øgede information fra det semistrukturerede interview, overgår den forøgede udfordring ved sammenligning mellem interviews. Specielt fordi gruppen anvender vandfaldsmetoden, er det vigtigt at der allerede fra starten af projektet bliver dannet et så relevant og omfangsrigt billede af problemet som muligt.

2.2.5 De 7 faser

For at foretage et succesfuldt interview, der giver relevante informationer, skal det planlægges og gennemarbejdes. For at hjælpe med dette benyttes Steinar Kvaales opdeling af erkendelsesopgaven i syv faser. Denne tillader en beskrivelse af hele processen bag interviewet. Dette baseres på Kapitel 9 fra bogen "Problemorienteret projektarbejde" [10].

1. Tematisering

Tematiseringen er den del i processen, hvor undersøgelsens formål bliver fastlagt. Det vil sige at motivationen for at afholde dette interview bliver beskrevet, og en forståelse for hvad der vil opnås med interviewet findes.

2. Design

Design af interviewundersøgelsen, er planlægningsfasen hvor, der kigges på alle de 7 faser. Der udarbejdes en interview guide, som styrer interviewet. Guiden bliver lavet under hensyntagen til hvilken interviewtype er valgt, samt moralske spørgsmål. Moralske spørgsmål indebærer eksempelvis, at interviewerne undersøger hvem de er i forhold til den interviewede, og hvordan dette kunne påvirke svarerne der gives.

3. Interview

Denne fase omhandler udførelsen af interviewet, samt hvilke ting der skal tages højde for, så den reelle interviewsituationen giver det bedst mulige resultat. Antallet af interviewere skal overvejes, samt deres opgaver. Det vil som udgangspunkt ofte være én person, som fører samtalen, og en anden der fungerer som referent. Inden interviewet går i gang skal parterne aftale hvilken information, der må bruges, og

2.2. Interview

hvad den må bruges til. Den interviewede skal også briefes omkring interviewets indhold, samt debriefes for at begge parter er enige om, at de fik det ud af interviewet, som de forventede.

4. Udskrivning

Udskrivningsfasen sikrer, at de noter eller optagelsen der dokumenterer interviewet, bliver omsat til læsbar information, i form af en transskription. Dette gør, at interviewet kan genlæses, og analysen derfor nemmere kan gennemføres.

5. Analyse

For at maksimere den brugbare viden fra interviewet, skal der anvendes en passende analysemetode. Nogle af de mest brugte analysemetoder i interviewundersøgelser er de følgende:

- Meningskondensering: Sammenfatning af interviewets hovedpointer.
- Meningskategorisering: Interviewets indhold opdeles i overordnede kategorier, for at kvantificere samtalen.

6. Bekræftelse

I bekræftelsesfasen vurderes indholdets objektivitet og validitet, og dermed også dens kvalitet. På denne måde undersøges der, om yderligere information eller dataindsamling er nødvendig.

7. Rapportering

Rapporteringsfasen bruges til at formidle det overordnede resultat fra interviewundersøgelsen i rapporten, hvor det også sikres, at alle moralske og faglige hensyn er beskrevet.

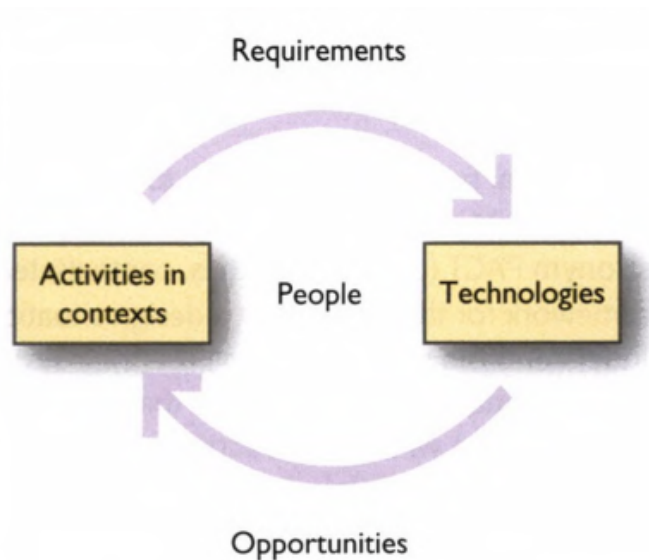
Opsummering

I dette kapitel er forskellige interviewteknikker, samt arbejdsmetoder i OOA&D beskrevet for at forstå dem, når de i de næste kapitler skal anvendes i valget og beskrivelsen af projektets system. Som det blev beskrevet i Delafsnit 2.1.3 er den traditionelle vandfaldsmodel valgt til styring af projektudviklingens livscyklus, eftersom OOA&D er ny teori i uddannelsens forløbet, og den tillader gruppen at arbejde på de forskellige faser i forbindelse med at de gennemgås i undervisningen. I Delafsnit 2.2.4 blev det besluttet at det semistrukturerede interview var ideelt, fordi det tager højde for projektets få informanter, samt fleksibiliteten som ønskes under interviewet.

Kapitel 3

PACT-analyse

En vigtig del af at lave interaktive systemer er at have et menneskeorienteret fokus, hvilket en undersøgelsen af mennesker, aktiviteter, teknologier og kontekst tillader os. Et fokus på mennesker er vigtigt, fordi teknologi ofte udvikles for i sidste ende at hjælpe de mennesker, som det påvirker. Figur 3.1 viser hvordan aktiviteter i en kontekst stiller krav til teknologierne, som senere skaber muligheder der kan ændre aktiviteterne. Dette forløb bygger videre på sig selv, da der i takt med at kravene til teknologien stiger, vil blive udviklet nye teknologier som danner muligheder for hvilke aktiviteter det kan bruges til.



Figur 3.1: Sammenhæng mellem PACT-elementer [11]

For at designe interaktive teknologier undersøges de fire elementer i PACT i henhold til det initierende problem. PACT består af People (Mennesker), Activities (Aktiviteter), Contexts (Kontekster) og Technologies (Teknologier). Analysen af

3.1. People

disse vil dermed give en bedre forståelse for, hvilken situation systemet skal udvikles til. De fire dele i PACT undersøges og analyseres enkeltvist i de følgende afsnit. Den teoretiske baggrund for analysen af disse specifikke områder, stammer fra Kapitel 2 i bogen "Designing Interactive Systems" [11].

Dataet for de fire analyseområder, i forhold til det initierende problem, indsamles primært ved hjælp af semi-strukturerede interviews med informanter. Argumentationen for valget af denne type interview findes i Delafsnit 2.2.4. Under projektet blev der foretaget flere interviews, der havde til formål at forstå den nuværende arbejdsproces, og klarlægge hvilke udfordringer der var i arbejdet med semesterevalueringssrapporten, for en semesterkordinator. For at planlægge interviewet blev Steinar Kvals 7 faser brugt, som blev beskrevet i Afsnit 2.2.5. Anvendelsen af denne model er beskrevet i Bilag B, interview-guiden som blev brugt til at føre interviewet står i Bilag A og selve transskriberinger findes i Bilag C.

3.1 People

I dette afsnit bliver de fysiske, psykologiske og sociale forskelle afklaret, omkring nøglepersonerne i udarbejdelsen af semesterevalueringssrapporten; semesterkoordinatorene på AAU.

3.1.1 Fysiske forskelle

Med alderen vil kroppen blive forringet, og blandt andet bliver synet ringere. Fra en alder af 40 år vil mange opleve nedsat synsevne, og vil derfor have sværere ved at læse små skrifttyper [12].

En semesterkordinator er en person der som minimum har en universitetsgrad. Hvis en semesterkordinator ikke allerede har problemer med synet, er der risiko for at det sker i løbet af sin ansættelse. To ud af tre af de interviewede semesterkoordinatorere benyttede allerede briller, derfor bør dette tages højde for i systemet.

3.1.2 Psykologiske forskelle

Psykologiske forskelle blandt mennesker kunne for eksempel være uddannelsesniveau, måder at huske på, hvilken erfaring vedkommende har inden for et bestemt emne, eller sågar sprog. De tre interviewede semesterkoordinatorer på AAU, stolede alle på deres hukommelse med hensyn at huske tilbage på tidligere år. Informanterne havde henholdsvis 4, 6 og 20 års erfaring som semesterkoordinatorer, hvilket gør at de har en del semestre der kan huskes tilbage på.

Alle informanterne talte flydende dansk under interviewene, og har som tidligere nævnt, færdiggjort en universitetsuddannelse, hvilket også kræver færdigheder på engelsk.

3.2. Activities

3.1.3 Sociale forskelle

For at beskrive de sociale forskelle blandt mennesker, undersøges hvad deres motivation og mål for at benytte et system er. Brugerens IT kundskaber er essentielle i forhold til at benytte et IT system, samt villigheden til at bruge det.

To ud af tre informanter har over 170 studerende på et semester, og har derfor en stor mængde data de skal have overblik over. Motivationen for at benytte et system for særligt disse to semesterkoordinatorer, ville blandt andet være at de kunne få et hurtigere overblik. Eftersom målgruppen er semesterkoordinatorer, som arbejder på universitetet, har de også erfaring med at håndtere en computer i en eller anden grad.

3.2 Activities

I dette afsnit vil aktiviteterne, som en semesterkoordinator udfører i forbindelse med evalueringen af et semester, blive undersøgt. Aktiviteterne og deres overordnede formål vil først blive undersøgt, hvorefter de primære karakteristiske egenskaber ved disse aktiviteter beskrives.

Læse og analysere semesterevalueringer

Semesterkoordinatoren modtager et hold studerendes udfyldte evalueringer, som består af både en kvalitativ og kvantitativ del. Bearbejdningstiden af dette data varierer afhængig af størrelsen på holdet. Dette fremgår også af de interviewede semesterkoordinatorer, der havde mellem 20 og 300 studerende. Der var derfor også forskel på, hvordan de arbejdede med dataet, for at danne sig et overblik. Metoderne som informanterne benyttede sig af, varierede mellem at læse evalueringerne igennem for at inddele kommentarene ved hjælp af at markere dem, eller blot tage noter løbende. Formålet med at læse og bearbejde disse semesterevalueringer, er at skrive en rapport som er repræsentativ for alle holdets studerende.

Udarbejde evalueringsrapport

De to førnævnte aktiviteter bruges som dataindsamling, hvor evalueringsrapporten er en sammenfatning af de mest centrale og sigende datapunkter, som blev indsamlet. Evalueringsrapporten sendes til det tilsvarende studienævn, som ifølge informant 2, bruger den til at sikre og beskrive kvaliteten af undervisningen.

3.2.1 Temporale aspekter

Semesterkoordinatorens udførelse af de beskrevne aktiviteter følger et fast mønster, eftersom de er tilknyttet et bestemt semester, som har en fast struktur. Efter hvert semester modtager koordinatoren et holds evalueringer, hvorefter de har omkring en

3.2. Activities

måned til at læse evalueringerne. I dette tidsrum vender de regelmæssigt tilbage til evalueringerne, indtil en evalueringsrapport er udarbejdet.

I læsningen af evalueringerne, og udarbejdelsen af evalueringsrapporten, vil koordinatoren med stor sandsynlighed ikke udføre arbejdet på én gang, og skal derfor senere kunne vende tilbage og genoptage disse aktiviteter.

Det sidste tidsmæssige aspekt er responstiden af anmodninger fra systemet. Eftersom semesterkoordinatoren kun skal anmode data og køre beregningsintensive funktioner en enkelt gang på en given maskine, og datamængden ikke er særlig stor, er dette ikke nogen bekymring.

3.2.2 Samarbejde

Semesterkoordinatorrollen fyldes kun af én person til hvert hold, hvilket betyder at opgaverne der tilfalder semesterkoordinatorer ofte er enkeltmandsopgaver. Arbejdet som de laver skal derfor ikke tage højde for at kunne samarbejde med andre.

3.2.3 Komplexitet

Kompleksiteten af de aktiviteterne er varierende. Gennemlæsningen af evalueringerne og dannelsen af et overblik er en relativt veldefineret opgave, det varierer dog hvor mange studerende de er ansvarlige for, og hvordan de normalt arbejder med evalueringerne. Ved udarbejdelsen af evalueringsrapporten øges kompleksiteten en smule, eftersom der ikke er defineret nogle specifikke krav til den. Det er givet hvilke overordnede emner den skal dække, men måden hvorpå den opstilles, og indholdet som der skrives i den er udefineret.

3.2.4 Sikkerhedskritisk

Resultatet af udarbejdelsen af evalueringsrapporten er isoleret set ikke sikkerhedskritisk, eftersom evalueringen først bliver sendt til studienævnet, som kan sende den tilbage med rettelser. Informationerne som læses igennem for at udarbejde denne rapport er derimod personfølsomme, og det bliver derfor sikkerhedskritisk at dette information kan holdes sikkert fra uvedkommende parter.

En anden overvejelse som skal tages højde for er risikoen for at semesterkoordinatoren laver en fejl. Eftersom en stor del af systemet består af at til- og fravælge informationer, vil det være svært at lave en fejl som du ikke har mulighed for at ændre tilbage. For at hjælpe koordinatoren med ikke at bruge forkert information, bør det gøres klart hvad der i øjeblikket vises.

3.3. Contexts

3.2.5 Typen af indholdet

Indholdet består - som tidligere beskrevet - primært af en oversigt over informationer, hvilket kræver et visuelt display. Søgning gennem dataen skal også være muligt, hvilket kræver tekstuel input igennem et keyboard.

3.3 Contexts

En kontekst kan betragtes som værende det bindende led mellem aktiviteter. Det kan også være det miljø, som omringer aktiviteterne.

Her ses der på lokaliteten af systemet; hvor det anvendes, hvordan det anvendes og de forhold der er inddraget i forbindelse med anvendelse af systemet.

3.3.1 Fysisk miljø

Det fysiske miljø, som systemet anvendes i, afhænger af hvordan og hvor brugeren vælger at benytte systemet, da det antages at det primært vil blive udført på en pc. Systemet vil typisk blive anvendt på et kontor på brugerens arbejdsplads.

Alt efter om brugeren har et privat kontor, kan forstyrrelser fra andre kollegaer have en indflydelse på brugerens koncentration, hvilket kan øge behovet for et system der er nemt at vende tilbage til. I et kontor på en arbejdsplads vil der typisk være god forbindelse til wi-fi, så der er mulighed for at indhente data fra blandt andet spørgeskemaerne.

3.3.2 Social kontekst

Der forekommer ikke nogen større sociale komplikationer i forhold til konteksten. Hvis brugeren ikke umiddelbart forstår systemets funktionalitet kan brugeren kontakte medarbejdere, som også anvender systemet. Systemet vil være anonymiseret, for at forebygge krænkelsen af personfølsomme informationer.

3.3.3 Organisatorisk kontekst

Systemet vil ikke overtage nogle jobs, da det blot er et værktøj til anvendelse af brugeren. Det vil heller ikke revolutionere en allerede eksisterende teknologi. Dog kan der ske problematiske ændringer i fremtiden for systemet. Formatet for spørgeskemaer kan ændres, og behovet for modernisering af teknologiske enheder og styresystemer kan skærpes. For at forebygge en negativ indvirkning på systemet i denne kontekst kræver det, at systemet er generisk, så det kan anvendes på en vilkårlig enhed af eksisterende brugere og nye brugere der burde komme.

3.4 Technologies

I teknologiafsnittet af PACT-analysen undersøges hvilke software og hardware enheder, der findes i systemet, og hvordan de modtager input, samt hvad der transformerer det til et output og fremviser dette for brugeren. Det nuværende system er et ikke teknologisk system, men i stedet en arbejdsproces, som semesterkoordinatorene udfører. Dette betyder at teknologiafsnittet af PACT-analyse, vil indblande en sammenligning mellem det nuværende system, og det ønsket udviklet.

3.4.1 Input

I den nuværende arbejdsproces, anvendes teknologiske medier kun i slutfasen, når semesterevalueringen skal skrives digitalt ind. Her anvendes mus og tastatur, som input medier.

I projektets slutprodukt, vil semesterkoordinatoren anvende teknologiske medier allerede fra starten af deres arbejdsproces. Det ønskede produkt er et analyseværktøj, som bliver stillet til rådighed på semesterkoordinatorens computer. Her ville input medierne også være mus og tastatur, anvendt til at navigere i programmet.

3.4.2 Output

Ligeledes som ved input, anvendes teknologiske medier kun i slutfasen, og her anvendes en computerskærm til fremvisning af det tekstprogram, som benyttes til at skrive semesterevalueringssrapporten.

I slutproduktet ville en skærm anvendes igennem hele processen, til fremvisning og navigation af analyseværktøjets interface.

3.4.3 Kommunikation

I det nuværende system, foregår der enkle kommunikationer i starten og i slutningen af processen. Først hvor semesterevalueringssvarelsesne modtages af semesterkoordinatoren, og hvor semesterevalueringssrapporten senere skal sendes til studienævnet.

I det fremtidige system, vil det være muligt, at systemet kan hente semesterevalueringssvarelsesne fra en server, hvori der vil være et behov for, at systemet er forbundet til internettet. Disse besvarelser skal dermed kun uploades en gang. Når en analyse er blevet udført på besvarelsesne, ville disse også blive gemt på serveren, så de ikke skal beregnes igen.

3.4.4 Systemindhold

Systemet, som det er nu, har systemindhold i form af de udfyldte spørgeskemaundersøgelser fra de studerende, samt selve semesterevalueringssrapporten, som semesterko-

3.5. Opsummering

ordinationen udarbejder.

I et fremtidigt system, vil systemindholdet stadig være de udfyldte spørgeskemaundersøgelser fra de studerende, men nu også indeholde en analyse af dataene derfra. Derudover vil semesterevalueringsrapporten ikke være en del af systemindholdet mere, da det fremtidige system vil være et analyseværktøj, og ikke et rapportskrivningsværktøj.

3.5 Opsummering

Viden omkring de interessenter, aktiviteter de udfører, konteksten som en arbejdsproces eksisterer i, og de teknologier som anvendes i systemet, er analyseret og skabt ny viden ud fra.

Generelt kunne det uddrages af analysen, at målgruppen for systemet er uniform, og handlingerne som udføres ligeledes. Dette ved at personer i målgruppen alle har samme job og derfor udviser lignende sociale og fysiske træk. Systemets kontekst gør, at specielle tilfælde indefor en af de nævnte kategorier, ikke har en betydning for den endelige funktionalitet af dette system. Afgrænsning af projektets retning gør, at det er muligt, at definere et specifikt system og danne information omkring dette.

Kapitel 4

Systemdefinition

For at kunne godkende det overordnede systems karakteristik, udarbejdes en systemdefinition. For at gøre dette benyttes BATOFF-kriterie. Dette virker også som en kortfattet beskrivelse af det ønskede system.

Systemdefinitionen er udarbejdet på baggrund af den information der resulterede fra den tidligere PACT-analyse, hvor situationen i forhold til folk, aktiviteter, kontekst, og teknologi blev undersøgt.

Formålet med systemdefinitionen er at konkludere bestemte detaljer af programmets adfærd og struktur, baseret på tidligere analyser af disse.

For at lave en systemdefinition, benyttes BATOFF kriteriet, som består af seks kategorier der skal analyseres. Hver af de seks elementer i BATOFF, henviser til centrale elementer i systemdefinitionen. Specifikt anvendes BATOFF til at beskrive de mest fundamentale beslutninger der skal tages i forbindelse med at lave en fornuftig digitaliseret løsning. Det teoretiske grundlag for en sådan BATOFF analyse stammer fra den engelske version af modellen - kaldet FACTOR - som bliver beskrevet i Kapitel 2 i bogen "Object Oriented Analysis & Design".

BATOFF-kriteriet består af:

- B** Betingelser, som systemet er udviklet til, og ud fra.
- A** Anvendelsesområde, som er de dele af en organisation, der anvender, overvåger eller kontrollerer et problemområde.
- T** Teknologien, som er både den teknologi, hvori systemet er udviklet og hvorpå systemet kører.
- O** Objekter, der er i problemområdet med identitet, tilstand og adfærd.
- F** Funktioner, der understøtter arbejdsopgaver i anvendelsesområdet.
- F** Filosofien i relation til dens kontekst

I de kommende delafsnit vil de forskellige elementer i BATOFF blive beskrevet i forbindelse med det beskrevne problem.

4.0.1 Betingelser

Dette delafsnit gennemgår de betingelser som skal opfyldes for brugeren, for at kunne løse det egentlige problem.

Systemet skal kunne håndtere at de besvarelser der modtages har forskelligt indhold og størrelse. Det vil sige at besvarelsernes tekst har forskellig længde, og at det der skrives deri, er forskelligt fra studerende til studerende.

Systemet skal anvendes som et hjælpeværktøj, så brugeren ikke er afhængig af værktøjet, og om nødvendigt, kan fravælge anvendelse af systemet. Dette betyder også, at hvis systemet skulle lave analysefejl, skal brugeren have mulighed for at lade være med at anvende den givne information.

Der skal være en hvis mængde kvalitativ data for at kunne foretage en sigende feedbackanalyse.

Alt feedback skal bringes på samme format, så analysesystemet kan skelne og kategorisere det.

4.0.2 Anvendelsesområde

Anvendelsesområdet består af de dele af en organisation som anvender, overvåger eller kontrollerer en del af problemområdet. I dette projekt består anvendelsesområdet primært af semesterkoordinatorer, som skal kunne gennemgå feedback, og skabe sig et overblik over de evalueringer som de modtager.

4.0.3 Teknologi

Dette delafsnit gennemgår både de teknologier der er blevet brugt til at udvikle systemet, samt dem hvorpå systemet vil blive anvendt.

Teknologierne brugt til udviklingen og brugen af systemet, er en PC platform med aktuelle perifere enheder.

4.0.4 Objekter

For at forstå problemområdet dannes et overblik over de objekter, som antages at finde sted i problemområdet, samt definerer og afgrænser det.

De primære objekter i vores model af problemområdet består af: *Hold*, *Semesterevaluering*, *Semesterkoordinator*, *Evalueringsrapport*

4.0.5 Funktioner

Funktioner beskriver de primære funktionaliteter som systemet tilbyder til understøttelse af arbejdsopgaver i anvendelsesområdet. Det beskrevne systems primære funktioner er: Analyse af tekstkommentarer og samling af evalueringsinformationer.

4.0.6 Filosofi

Filosofien der ligger bag systemet beskriver det overordnede ansvar som systemet har. I dette tilfælde er systemets primære ansvar at virke som et overblikdannende værktøj.

4.1 Problemformulering

I PACT-analysen blev det tydeligt at semesterkoordinatorernes metoder, og størrelsen på deres hold, varierer meget, og den resulterende rapport varierer derfor også.

Som det blev beskrevet i indledningen, er antallet af studerende på Aalborg Universitet stigende, og der er derfor også et stigende antal evalueringsbesvarelser som semesterkoordinatorerne på AAU manuelt skal gennemgå. Med den nuværende koordinatorstruktur vil dette betyde at der skal udføres et større arbejde, på den samme mængde tid. Med flere evalueringsbesvarelser kan det også være svært at lave en repræsentativ sammenfatning af alle de studerendes holdninger

BATOFF-kriteriet beskriver systemdefinitionen, som gennemgår de forskellige elementer i udviklingen af et værktøj til at analysere og visualisere semesterevalueringer. Deraf blev det klarlagt at et sådant system skulle være et hjælpeværktøj for semesterkoordinatorerne, der har til formål at optimere processen af at bearbejde semesterevalueringerne. Et sådant værktøj ville være et skridt på vejen til at gøre semesterevalueringssrapporterne nemmere at udvikle, og mere uniforme.

Disse hovedpunkter, samt den initierende problemstilling, danner grundlaget for en problemformulering, som lyder:

Hvordan kan semesterkoordinatorerne på AAU hjælpes i semesterevalueringssprocessen med et databehandlingsværktøj?

For at besvare denne hoved-problemformulering, er der følgende underproblemer der skal undersøges:

- Hvordan kan et system der gør det nemmere at kunne analysere semesterevalueringsspørgeskemaerne modelleres?
- Hvordan kan software laves, der danner et overblik over generelle holdninger i tekstuel data?

Denne problemformulering virker som en overordnet beskrivelse af problemet. For at kunne undersøge yderligere hvad der skal til for at løse det, opstilles der krav til funktionaliteten og derigennem, krav til hvordan brugergrænsefladen skal visualisere dette.

Kapitel 5

Kravspecifikation

I dette Kapitel vil kravene for produktet blive beskrevet i forhold til systemets kontekst, begrænsninger, og formål - som blev undersøgt i PACT-analysen og systemdefinitionen. Formålet er dermed at definere forventninger til systemet på baggrund af målet med systemet, og dermed sætte en basis for udvikling af dette. Målet med systemet er at kunne danne et repræsentativt overblik over kommentarerne fra semesterevalueringerne, som det fremgår af problemformuleringen. Denne kvantitative repræsentation af dataen kan bruges til udvikling af forskellige statistikker, som kan hjælpe semesterkoordinatoren med sin udvikling af evalueringsrapporter. Den teoretiske baggrund for at kunne beskrive de systemrelevante krav dækkes i Kapitel 2 af bogen "Håndbog i Struktureret Programudvikling" [13].

5.1 Kvalitetsfaktorer

Dette afsnit vil beskrive 7 af de centrale faktorer som er vigtige for systemets kvalitet. Hvert punkt vil have argumenter for hvorfor det er en vigtig-, og hvorfor det er en mindre vigtig faktor.

5.1.1 Pålidelighed

Pålidelighed hentyder til systemets mangel på fejl mens det anvendes. Dette vil sige mængden af fejl der kan opstå i programmets køretid.

Systemet bruges til at analysere og drage konklusioner på større sæt af besvarelser, og kategorisere besvarelser. Det er derfor meget vigtigt at fortolkningen af denne data er præcis, så brugeren ikke bliver misinformeret.

Gruppen af brugere er ansatte, og deres tid er derfor værdifuld. Hvis Systemets fejl får brugeren til at miste tid, er det et økonomisk tab for universitetet.

5.1. Kvalitetsfaktorer

Systemets opgave er ikke tidssensitiv, og bruges gennem 2 perioder af året, derfor er pålideligheden af systemet i forhold til 'Oppe-tid' ikke en central faktor for systemet.

5.1.2 Vedligeholdelsesbart

Denne faktor drejer sig om hvor godt et system kan vedligeholdes, og bestemmes af den tid det tager at finde og rette fejl samt problemer, efter systemet er udgivet til brugerne.

Et af systemets funktioner er at sammenligne evalueringer over flere semestre. For at denne funktion er brugbar, kræver det at systemet kan anvendes og opdateres over flere år.

En af argumenter der gør et system svært at vedligeholde er det finansielle, og tidskrævende ved det. Der skal være folk ansat til løbende at vedligeholde systemet, hvilket vil være et stort minus for mange universiteter.

5.1.3 Udvidelsesbart

Denne faktor defineres af produktiviteten af at udvikle systemet efter udgivelse, relativt til udvidelse af systemet under projektets initierende udvikling.

Systemet har ét centralt formål, og ønsker ikke at dække flere problemer eller brugergrupper. Dette minimerer nødvendigheden for udvidelse af Systemet. Udførelsen af analysen i systemet vil kunne trænes og forbedres yderligere med tiden. På trods af dette, kunne det være en fordel at kunne opgradere eller tilføje funktionaliteter til systemet med tiden.

5.1.4 Brugervenlighed

Den mængde tid det vil tage for en bruger at lære hvordan produktet skal bruges. Definerer brugervenligheden af produktet.

Systemet bruges hovedsageligt af folk som arbejder med kontorarbejde og organisatorisk arbejde.

Brugerne kommer ikke til at udnytte systemet ofte, og eftersom øget effektivitet er en central motivation for brugere, er et intuitivt design med til at øge effektiviteten.

De ønskede brugere har allerede en måde at arbejde med evalueringer, og for at overbevise dem om at integrere systemet som en ny del af arbejdsprocessen, skal det være let at anvende. Hvis ikke dette er intuitivt, vil det formodentligt ikke kunne udfordre eksisterende metoder.

5.2. De specifikke krav

5.1.5 Genbrugelighed

Systemkomponenternes evne til at blive brugt flere gange i systemet bestemmer genbrugelighed.

Givet ændringer i processen af semesterevaluering, eller formatet af den kvalitative data, er det vigtigt at evalueringsanalysen fra systemet kan genanvendes, både i den nye proces eller med det nye format. Dette vil også være relevant hvis systemet skal udvides til brug i andre institutioner.

5.1.6 Integritet

Integriteten er effektiviteten af den sikkerhed som systemet har, i forhold til at beskytte fortrolig data.

Systemet arbejder direkte med evalueringer, som indeholder personlige og følsomme data. Evalueringer indeholder navne af forelæsere og studerende, og disse data gemmes på en ekstern server. Det ville være krænkende hvis disse data blev offentligt tilgængelige.

Systemets login system indeholder også sensitiv data, i form af brugernavne og adgangskoder. Folk vil ofte bruge den samme kode flere steder, og dermed er det utrolig vigtigt at beskytte den data for at bevare brugerens sikkerhed på andre platforme og systemer.

5.1.7 Effektivitet

Ydeevnen af Systemet i forhold til dets formål bestemmer effektiviteten af dette system.

Systemet udvikles hovedsageligt for at øge arbejds effektiviteten af semesterkordinatorer, i forhold til mængden af tid brugt på at analysere evalueringer samt hvor præcise deres endelige rapporter er. Da der kun er afsat en vis mængde timer, som semesterkoordinatorerne har til at udføre deres arbejde, skal systemet ikke være en faktor der skaber spildtid.

5.2 De specifikke krav

Dette afsnit beskriver de specifikke krav, som er blevet valgt på baggrund af analysen, der blev lavet på problemområdet i PACT, samt systemdefinitionen, som findes i henholdsvis Kapitel 3 og 4.

5.2. De specifikke krav

5.2.1 Definitioner

Dette delafsnit introducerer begreber, som anvendes til at beskrive de specifikke krav.

Brugernavn	Semesterkoordinatorens aau-mail.
Adgangskode	5 til 50 tegn, A til Z, samt tal og specielle tegn.
Svar	Ét enkelt svar i en besvarelse.
Besvarelse	Én enkelt studerendes svar på en hel semesterevaluering.
Evaluerings	Én samling af besvarelser.
Analyse	En samling af evalueringer behandlet og analyseret af programmet.

Med disse definitioner fastlagt, kan de funktionelle krav og brugergrænsefladekravene beskrives.

5.2.2 Funktionelle Krav

I dette delafsnit er der formuleret funktionelle krav, som skal dækkes af systemet. Kravene bliver tildelt et nummer, for senere at kunne referere tilbage til dem, samt et navn og en beskrivelse. For at få et mere detaljeret indblik, i hvad de enkelte krav består af, kan der refereres til Bilag D.

1. **Indkapsle relevant data:** At isolere relevant data på opbevaringsstedet, så kun folk med tilladelsen til det, har adgang til specifikke evalueringsdata. Eksempelvis bør semesterkordinatorer kun kunne se evalueringer for de hold som de er tilkøbt. Dette er baseret på Delafsnit 3.2.4, om hvor sikkerhedskritisk programmet skulle være.
2. **Adskille Hold og Data:** At holde evalueringer adskilt så brugeren let kan navigere gennem alle evalueringer for et bestemt hold. Dette er fordi evalueringerne bliver gennemgået holdmæssigt, som det blev beskrevet i Afsnit 3.2.
3. **Kategorisering af feedback:** At kategorisere lignende kommentarer, for at give et overblik over emner der hyppigt refereres til. Dette er baseret på problemformuleringen i Afsnit 4.1.
4. **Tilgå ubehandlede evalueringer:** At kunne læse de ubehandlede evalueringer uden sortering eller analyse. Dette er hvad semesterkordinatoren allerede bruger, som set i Afsnit 3.2, og tillader derfor programmet at være en tilføjelse til arbejdet i stedet for en erstatning.
5. **Tilgå én studerendes besvarelser samlet:** At vise besvarelsen fra en given studerende til et givet semester. Dette er baseret på alle 3 interviews med projektets informanter, som det kan ses i Bilag C.

5.2. De specifikke krav

6. **Markering af kommentar:** At lade brugeren selv udvælge enkelte kommentarer som de finder særligt sigende, og vil kunne finde tilbage til. Dette er baseret på interviewet med informant 2 og 3, som kan læses i Bilag C.
7. **Sammenligning af årgange:** Systemet skal kunne sammenligne feedback over en periode, for at tillade semesterkordinatorer lettere at få en generel forståelse af hvordan et modul har udviklet sig over de tidligere semestre. Dette er basere på interviews i Bilag C hvor alle der blev interviewet sagde de sammenlignede med tidligere semestre.
8. **Forståelse af studerendes mening ved brug af Sentimentanalyse:** Tillad semesterkordinatorer en bedre forståelse af den generelle mening om et enkelte moduler eller projektforsløb. Dette krav er baseret på problemformuleringen i Afsnit 4.1.
9. **Genkendelse af Nøgleord:** Specifikke ord kan have en stor betydning for betydningen af resten af sætningen. Dette krav er inspireret af en brainstorm baseret på problemformuleringen i Afsnit 4.1.
10. **Fjernelse af Tegnsætning:** Tegnsætning har ikke nogen central betydning for meningen med sætningen, og tilføjer unødigt kompleksitet. Dette er en del af at fjerne støj fra input data.
11. **Fjernelse af Stopord:** Nogle ord har ingen betydning for en sætnings budskab, og kan derfor blive filtreret ud af datasættet. Formålet er at fjerne støj fra input data.
12. **Omdannelse til små bogstaver:** Forskellen mellem store og små bogstaver har minimal betydning for en sætningens budskab. Dette vil reducere støj i input data.
13. **Analysér dansk tekst:** Systemet skal kunne analysere danske besvarelser. Informanter til projektet kunne alle flydende dansk, som beskrevet i Delafsnit 3.1.2, og er én grund til at programmet skal fokusere på analyse af dansk tekst.

5.2.3 Brugergrænsefladekrav

Dette afsnit dækker over krav til hvad brugeren skal kunne se og gøre i systemets brugerflade. For at kunne visualisere funktionelle krav, skal disse brugergrænsefladekrav mødes.

14. **Login:** Der skal være en loginskærm til at autentificere brugerens information som udnytter Aalborg Universitets valideringssystem.
15. **Vælg Hold:** Brugeren skal kunne arbejde med flere separate hold.
16. **Dansk tekst:** Alt tekst i grænsefladen skal være på dansk.

5.3. Afrunding

17. **Upload data:** Brugeren skal kunne uploade nye semesterevalueringer som input til systemet.
18. **Markering:** Der skal være mulighed for at markere hver enkel kommentar.
19. **Se markerede kommentarer:** Brugeren skal kunne se sine markerede kommentarer, som en liste for hvert modul.
20. **Se enkel studerendes kommentarer:** Brugeren skal kunne vælge at se alle kommentarer fra en enkel studerende.
21. **Præsentation af analyseret data:** Systemet skal visualisere det analyserede data, i form af ordnede lister af udvalgte kommentarer, eller grafer.
22. **Tilføj hold:** Brugeren skal være i stand til at tilføje hold, for at indkapsle dataet for hver semesterevaluering.

5.3 Afrunding

Systemets krav er på dette punkt specificerede. Dette dækker over de kvalitetsfaktorer, specifikke krav og eksterne grænsefladekrav som er nævnt i dette afsnit. Med dette gjort, kan problem- og anvendelsesområdet analyseres.

Del II

Modellering og Design

Kapitel 6

Analyse af problemområdet

I dette kapitel vil der blive udarbejdet en analyse af problemområdet, som bygger på systemdefinitionen, der blev beskrevet i Kapitel 4. Når der arbejdes med problemområdet, arbejdes der med de elementer, som systemet kommer til at overvåge eller administrere. Denne analyse er vigtig, da den giver information om, hvordan problemområdet skal modelleres. For at lave en god analyse deraf, skal det undersøges, hvilken information som systemet skal behandle. For at undersøge dette, vil indholdet i de følgende tre punkter blive undersøgt, hvoraf teorien bag disse afsnit stammer fra Del 1 af "Object Oriented Analysis & Design" [6]:

- **Klasser:** Beskriver de objekter og hændelser der findes i problemområdet, og resulterer i en hændelsestabel.
- **Struktur:** Beskriver klasser og objekters konceptuelle forhold til hinanden, og resulterer i et klassesdiagram.
- **Adfærd:** Beskriver objekters dynamiske egenskaber, og resulterer i et adfærdsmønster, som beskriver karaktertræk fra alle klasser i klassesdiagrammet.

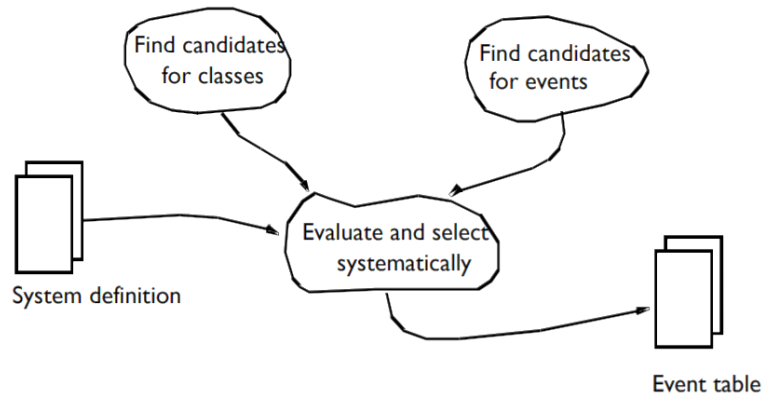
6.1 Klasser

Klasseaktiviteten består af tre nøglekoncepter: Objekter, Klasser og Hændelser. Et objekt fungerer som en instans af noget mere generelt, hvilket betyder at hvert objekt har identitet, tilstand og adfærd. Denne adfærd kan beskrives med hændelser, der er øjeblikkelige i tid, og ikke kan splittes op i flere forskellige øjeblikkelige handlinger. Klasser virker som en indkapsling af objekter, som deler struktur- og adfærdsmæssige karaktertræk. Klasser generaliserer altså objekter med fælles træk.

Klasseaktiviteten bruges til at lave en hændelsestabel for at forstå relationerne mellem forskellige klasser. Denne proces kan illustreres som vist i Figur 6.1. Her vises det, hvordan der begyndes ukritisk med at finde mulige klasser og hændelser,

6.1. Klasser

hvorefter der så evalueres og udvælges de mest relevante på baggrund af systemdefinitionen, hvilke kommer til at danne hændelsestabellen. Evalueringen og udvælgelsen sker på baggrund af faste kriterier[6, p. 63]. For klasser gælder de følgende kriterier:



Figur 6.1: Illustration af processen fra systemdefinition til hændelsestabel [6]

- Der skal kunne identificeres objekter fra klassen
- Klassen skal indeholde unik information i forhold til andre klasser
- Klassen skal omfatte flere objekter
- Klassen skal have en passende og håndterbar mængde hændelser

For hændelser gælder de følgende kriterier:

- Hændelsen skal ske øjeblikkeligt
- Hændelsen skal være udelelig. Dette betyder, at den ikke må være afhængig af udførelsen af andre hændelser.
- Hændelsen skal kunne registreres i det den sker.

Mulige kandidater til klasser blev fundet ved hjælp af en brainstorm. Kandidaterne blev derefter evalueret på baggrund af førnævnte kriterier, og skåret ned til de mest relevante. De valgte klasser, og en beskrivelse deraf, fremgår af Tabel 6.1.

6.1. Klasser

Klasser	Beskrivelse
Hold	Klassen repræsenterer en bestemt studieretning på et specifikt semester. Klassen skal indeholde informationer omkring navnet på dens studieretning, det specifikke semester og antallet af studerende.
Studerende	Klassen repræsenterer personer der er studerende på AAU, og består af den studerendes studienummer.
Semesterkoordinator	Klassen repræsenterer en personer der er ansat som semesterkoordinator på AAU og er tilkoblet et eller flere hold. Klassen består af semesterkoordinatorens e-mail som identifikation.
Semesterevaluering	Klassen repræsenterer den samlede evalueringsbesvarelse for et hold på et givet semester.
Besvarelse	Klassen repræsenterer en studerendes fulde besvarelse på en semesterevaluering.
Kommentar	Klassen repræsenterer en tekstuel kommentar skrevet af en studerende, til et bestemt spørgsmål.
Bedømmelse	Klassen repræsenterer en multiple-choice bedømmelse givet af en studerende, på et bestemt spørgsmål.
Analyseret evaluering	Klassen repræsenterer en semesterevaluering, som er blevet analyseret.
Evalueringsrapport	Klassen repræsenterer den færdige rapport, som semesterkoordinatoren skriver på baggrund af de studerendes evalueringer.
Modul	Klassen repræsenterer en del af undervisningen på AAU, i form af enten et kursus eller et projekt.
Kursus	Klassen repræsenterer de forskellige kurser, som de studerende kan have på AAU.
Projekt	Klassen repræsenterer de projekter, som de studerende kan have på AAU.

Table 6.1: Liste over klasser, og en beskrivelse deraf

De tilhørende hændelser blev fundet på baggrund af de udvalgte klasser. Disse hændelseskandidater blev også evalueret og skåret til, på baggrund af førnævnte hændelseskriterier. De valgte hændelser er vist i Tabel 6.2.

6.1.1 Sammenhængen mellem klasser og hændelser

Med disse klasser og tilhørende hændelser kan der nu opstilles en hændelsestabel som viser dynamikken af problemområdet, og de interne sammenhænge mellem hændelser og klassernes tilhørende objekter. Denne hændelsestabel er illustreret i Tabel

6.1. Klasser

Hændelser	Beskrivelse
Evalueringer modtaget	Hændelsen sker, når semesterkoordinatoren modtager de udfyldte semesterevalueringer.
Semester startet	Hændelsen sker på første skemalagte dag af semestret.
Semester afsluttet	Hændelsen sker når den sidste skemalagte ordinære eksamen er afholdt.
Semesterkoordinator begyndt	Hændelsen sker hver gang en semesterkoordinator bliver tilkoblet et nyt hold.
Semesterkoordinator stoppet	Hændelsen sker hver gang en semesterkoordinator ikke længere er tilkoblet et hold.
Evalueringsrapport sendt	Hændelsen sker på et hold ved en bestemt dato på baggrund af en tidsfrist, hvor alle udfyldte semesterevalueringer sendes til semesterkoordinatoren.
Kommentar udvalgt	Hændelsen sker hver gang, at en semesterkoordinator udvælger en specifik kommentar af særlig relevans.
Evalueringer analyseret	Hændelsen sker, når et holds semesterevaluering er færdig-analyseret.

Table 6.2: Liste over hændelser, og en beskrivelse deraf

6.3.

Hændelsestabellen kan også bruges som et værktøj til at evaluere kvaliteten af de valgte klasser og hændelser. En klasse der interagerer med for mange hændelser, kan være problematisk, da den derfor har for meget ansvar i problemområdet. I den beskrevne tabel er 'Semesterkoordinator' en sådan klasse. Denne klasse beskriver dog en ekstremt central del af problemområdet, og semesterkoordinatoren er en svær klasse at dele op i forskellige ansvarsområder. Det gælder også for en hændelse. Hvis en hændelse relaterer til mange klasser, kan det være et tegn på at hændelsen kan brydes ned i flere dele, eller at de fælles klasser hænger for meget sammen. Dette tyder dog ikke på at være et problem i den beskrevne hændelsestabel, da alle hændelserne kun relaterer til omkring to eller tre klasser.

6.2. Struktur

Hændelser	Klasser											
	Hold	Studerende	Semesterkoordinator	Semesterevaluering	Besvarelse	Kommentar	Bedømmelse	Analyseret evaluering	Evalueringsrapport	Modul	Kursus	Projekt
	Evalueringer modtaget			X	X	X	X	X	X			
	Semester startet	X	X	X								
	Semester afsluttet	X	X	X								
	Semesterkoordinator begyndt	X		X								
	Semesterkoordinator stoppet	X		X								
	Evalueringsrapport sendt			X					X			
	Kommentar udvalgt			X			X			X	X	X
	Evalueringer analyseret			X	X			X		X	X	X

Table 6.3: Indledende hændelsestabel

6.2 Struktur

Strukturaktiviteten beskæftiger sig med fire primære strukturer, hvoraf denne rapport vil beskrive, de tre anvendte i projektet. Disse nøglekoncepter er henholdsvis; generalisering, aggregering og associering.

Strukturer kan beskrives som de relationer eller forbindelser, som eksisterer mellem klasser og objekter i problemområdet.

Strukturaktiviteten anvendes til at konstruere et klassediagram, som kan danne overblik over de specifikke relationer i problemområdet. Klassediagrammet baseres på en tidligere udarbejdet hændelsestabel og de strukturmønstre som der identificeres i strukturaktiviteten.

Generalisering

En generalisering beskriver en relation mellem to eller flere specialiserede klasser og en mere generaliseret klasse. Den generelle klasse beskriver fælles egenskaber ved de mere specialiserede klasser, hvilket betyder at hver af de specialiserede klasser kan påtage sig rollen som den generelle klasse. Denne type relation visualiseres ved brug af en pil med trekantet pilhoved fra den specielle klasse til den generelle klasse, se Figur 6.2.

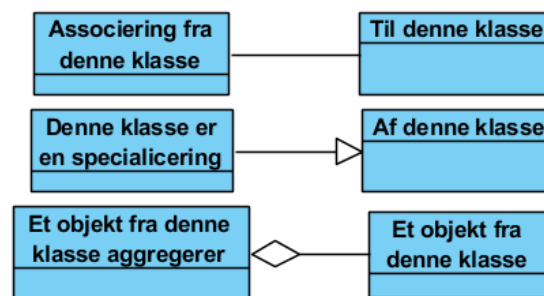
Aggregering

En aggregering beskrives som et overordnet objekt, som består af et antal mindre og underordnede objekter. En anden måde at forstå det på kunne være at det

overordnede objekt er bygget op af en række mindre objekter. En aggregering visualiseres ved brug af en pil, hvor pilhovedet er en rombe, se Figur 6.2. Ved denne strukturtype går pilen igen fra den mindre del til den overlegne del af relationen.

Associering

En association er en betydningsfuld relation mellem et antal objekter, hvor ingen af objekterne er en definerende egenskab eller del af en anden klasse i relationen. Associeringer er primært anvendt når det ikke er passende at anvende en aggregering, da denne type ville antyde for stærk en binding. Associeringer visualiseres med en almindelig linje mellem de to relaterede klasser, se Figur 6.2.



Figur 6.2: Illustration af relationspile

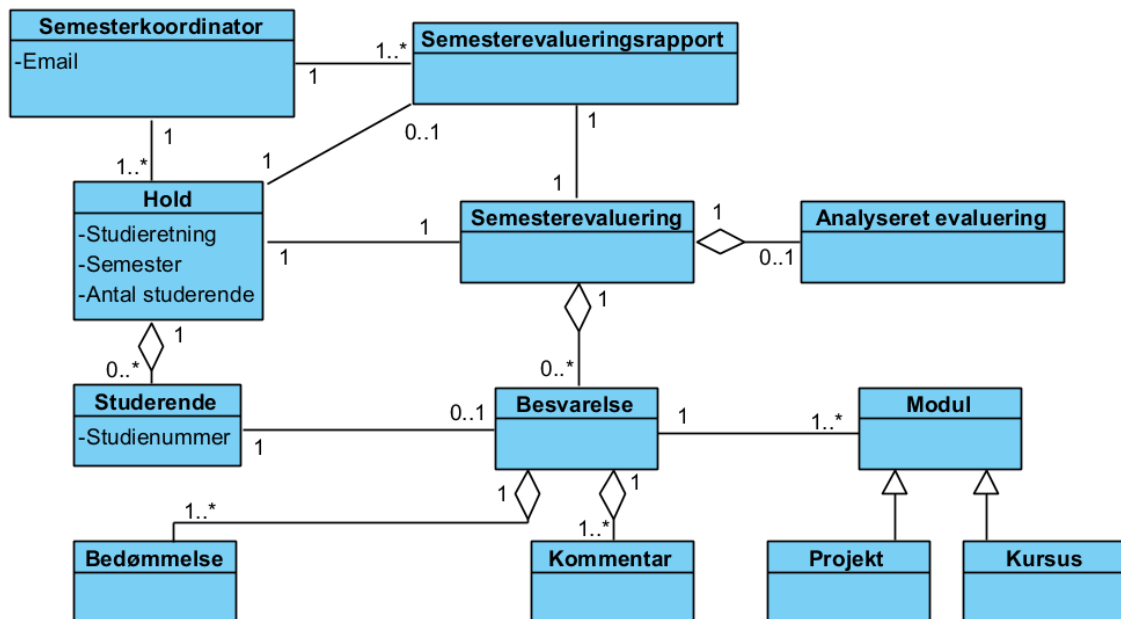
For at kunne bestemme de relationer der eksisterer mellem klasserne i problemområdet, beskrives de med henblik på deres betydning i systemet og hvordan de interagerer med de andre klasser.

Systemets problemstilling baseres på at de studerende skal udfylde en semesterevaluering efter hvert semester. Dette gøres ved at give bedømmelser på baggrund af forskellige udsagn, samt eventuelle kvalitative kommentarer. Disse besvarelser gives i forhold til de moduler, som den studerende har deltaget i på det pågældende semester. Disse moduler klassificeres som en generalisering for semesterets projekt og de tilhørende kurser.

Et hold består af et antal studerende. En række af disse studerende, tilhørende et hold, udfylder hver især deres besvarelse. Samlingen af et givent holds besvarelser defineres som en semesterevaluering. Fra denne semesterevaluering er det holdets tilkoblede semesterkoordinators opgave at lave en semesterevalueringsrapport. Denne rapport udarbejdes primært på baggrund af informationerne fra semesterevalueringen.

Med disse relationspile, kan sammenhænge der blev vist i hændelsestabellen i Tabel 6.3 illustreres som et klassediagram. Klassediagrammet fremgår af Figur 6.3.

6.3. Adfærd



Figur 6.3: Illustration af relationerne mellem klasser i form af klassesdiagram

Ved at have bestemt klasserne der eksisterer i problemområdet, samt relationer der er mellem dem, er det nu muligt at beskrive deres adfærdsmønstre.

6.3 Adfærd

Problemområdet er ofte dynamisk, og dermed bør adfærd som repræsenterer denne dynamik være en del af den udviklede model af problemområdet. For at analysere potentiel adfærd i problemområdet, kan der bruges hændelsesforløb. Et hændelsesforløb er en sekvens af hændelser som involverer et specifikt objekt.

Hvis adfærdsmønstre defineres som en ikke-sekventiel liste af et objekts mulige adfærd, kan der skabes en model. Hændelsesforløb er bedst brugte til at skabe adfærdsmønstre, som er en beskrivelse af alle mulige hændelsesforløb for alle objekter af en klasse.

Det er centralt at hændelsesforløb bestemmer rækkefølgen af individuelle hændelser gennem tid, ved brug af grundlæggende kontrolstrukturer.

Sekvens

Den første og enkleste af disse kontrolstrukturer er den sekventielle. Alt hvad der sker, foregår linært, og hændelserne sker dermed altid i en bestemt rækkefølge. Se Figur 6.4.

Selektion

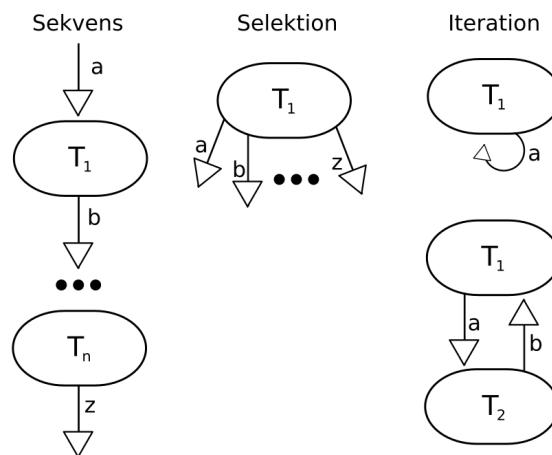
Ved selektion er der mulighed for at en af flere handlinger kan forekomme. På denne måde behøver adfærdsmønstret ikke at blive ens strømlinjet, som det vil gøre ved sekvens. Se Figur 6.4.

Iterationer

Iterationer anvendes på to måder, enten til at foretage en handling, og så vende tilbage til den originale tilstand, eller ved at foretage en handling der returnerer til en forhenværende tilstand. Se Figur 6.4.

6.3.1 Visuel beskrivelse

Adfærd kan beskrives visuelt ved brug af tilstandsdiagrammer. Et tilstandsdiagram bruger forskellige blokke som er en tilstand af et objekt. Objektet vil derefter skifte tilstand på grund af de forskellige adfærdsmønstre som opstår kronologisk. Tilstandsdiagrammer har den fordel, at de giver et visuelt kort over alle potentielle livsforløb et objekt kan have. Visuelle repræsentationer over de tre grundlæggende adfærdsmønstre kan ses på Figur 6.4.

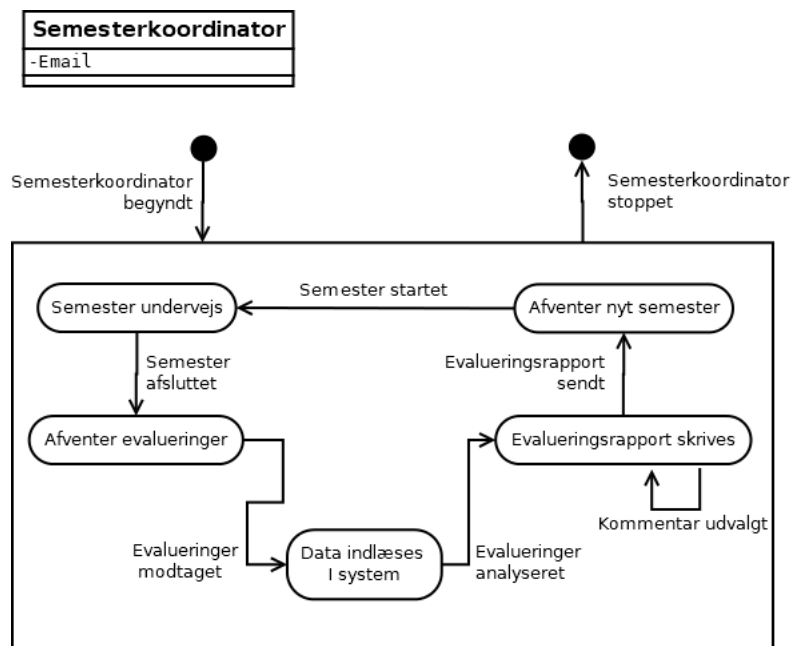


Figur 6.4: Eksempler på visuelle tilstandsrepræsentationer af adfærdsmønstre. Baseret på Figur 5.3 fra OOA&D [6]

6.3.2 Udvalgte Klasser

Følgende klasser er udvalgt som eksempler, da de indeholder en betydelig mængde af problemområdets adfærd. Der er lavet et tilstandsdiagram for hver af disse, sammen med en oversigt over klassens attributter. Klasserne og hændelserne er udvalgt fra hændelsestabellen i Tabel 6.3.

6.3. Adfærd



Figur 6.5: Et tilstandsdiagram for klassen *Semesterkoordinator*

Semesterkoordinator

Figur 6.5, viser adfærden for klassen *Semesterkoordinator*.

Objektet tilføjes til systemet, når hændelsen *Semesterkoordinator begyndt* forekommer. Dette kan føre til alle mulige tilstande, da det ikke vides hvornår i forløbet semesterkoordinatoren begynder.

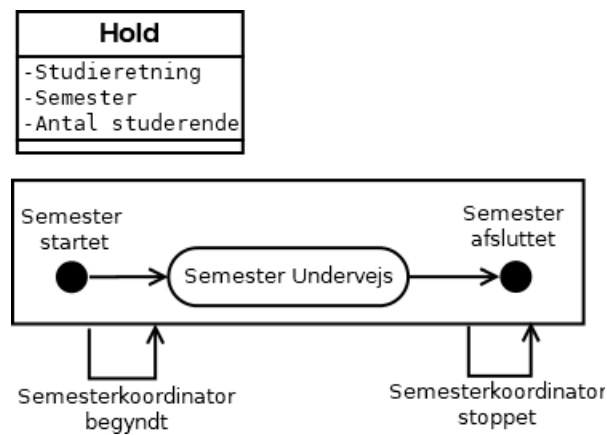
Når hændelsen *Semester startet* sker, vil tilstanden *Semester Undervejs* være aktuell. Fra denne tilstand ventes der på hændelsen *Semester Afsluttet*: Dette fører til et ventestadie, fordi de studerendes evalueringer af semestret skal bruges før *Evalueringer Modtaget* forekommer. Når dette sker, vil besvarelsene blive indlæst og analyseret.

I tilstanden *Evalueringsrapport skrives* kan der forekomme to hændelser: *Kommentar udvalgt* for kommentarer der skal tages særligt hensyn til i rapporten, eller *Evalueringsrapport Sendt* når semesterkoordinatoren har sendt rapporten.

Dette fører tilbage til den initielle tilstand: *Afventer nyt semester*. Herfra kan *Semester Startet* forekomme, og herved starter hele diagrammet forfra.

Objektet fjernes fra systemet når hændelsen *Semesterkoordinator stopper* sker.

6.3. Adfærd



Figur 6.6: Et tilstandsdiagram for klassen *Hold*

Hold

På Figur 6.6 ses en visuel repræsentation af Hold-klassens adfærd. Den har et begyndelsespunkt ved hændelsen *Semester startet*. Herfra vil den komme i tilstanden *Semester undervejs*. Fra tilstanden *Semester undervejs* fortsættes der, til afslutningspunktet *Semester Afsluttet*. I hele dette forløb kan en semesterkoordinator begynde og stoppe iterativt.

6.3.3 Hændelsestabel

Disse tilstandsdiagrammer visualiserer hændelserne i hændelsestabellen fra Tabel 6.3. Med disse visualiseringer, samt hændelsestabellen, kan det bestemmes hvor mange gange hver hændelse kan ske. En ny notation for dette kan ses på Figur 6.4, og den nye hændelsestabel kan ses på Figur 6.5.

Hændelse	Notation
Foregår nul eller én gang	+
Foregår nul eller flere gange	*

Table 6.4: Hændelsestabel notationer, taget fra forklaringen under Figur 5.7 i OOA&D [6]

6.4. Afrunding

Hændelser	Klasser											
	Hold	Studerende	Semesterkoordinator	Semesterevaluering	Besvarelse	Kommentar	Bedømmelse	Analyseret evaluering	Evalueringsrapport	Modul	Kursus	Projekt
	Evalueringer modtaget			*	+	+	+	+	+			
	Semester startet	+	*	*								
	Semester afsluttet	+	*	*								
	Semesterkoordinator begyndt	*		+								
	Semesterkoordinator stoppet	*		+								
	Evalueringsrapport sendt			*					+			
	Kommentar udvalgt			*			+			*	*	*
	Evalueringer analyseret			*	+			+		+	+	+

Table 6.5: Hændelsestabel med notation for om en hændelse kan ske én eller flere gange.

6.4 Afrunding

Analysen af problemområdet har skabt en viden omkring systemets klasser, deres indbyrdes struktur og deres adfærd. Der blev skabt et overblik over den opnåede viden ved at lave henholdsvis tilstandsdiagrammer, hændelsestabeller, samt et klasse-diagram.

Analysen af problemområdet bliver benyttet i Kapitel 7, hvor det bliver brugt i analysen af anvendelsesområde.

Kapitel 7

Analyse af anvendelsesområde

I dette kapitel beskrives analysen af anvendelsesområdet. Anvendelsesområdet referer til personer eller andre systemer, der interagerer med det tidligere beskrevne problemområde. For at få en forståelse af anvendelsesområdet, undersøges de forskellige aktører, samt de tilhørende brugsmønstre [6]. For at dække dette, undersøges de følgende to punkter:

- Anvendelse: Hvordan interagerer systemet med folk og systemer fra problemområdet?
- Funktioner: Hvordan er systemets evne til at behandle given information?

7.1 Anvendelse

Hvordan et system bruges, kan beskrives ved at identificere dets aktører, og deres brugsmønstre. En aktør er en abstraktion af brugere og andre systemer som interagerer med det ønskede system. Et brugsmønster er et mønster for interaktion mellem systemet og aktører i anvendelsesområdet.

Anvendelsesområdet og systemkrav kan bestemmes ud fra brugsmønstrene, som derfor er centrale i denne del af analysen. Der kan specificeres mange meget specifikke brugsmønstre, men for at danne forståelse af et systems brug er det mest effektivt at bestræbe at dække anvendelsesområdet med færrest mulige brugsmønstre.

I denne sammenhæng er der også potentielle sociale ændringer at tage i betragtning, fordi ændringer i det system som en organisation bruger, kan resultere i ændringer på hvordan der arbejdes med det.

Valg af aktører er baseret på spørgsmålet om hvem der skal bruge det ønskede system. Derefter, for at specificere aktørerne, skal der bestemmes opdelingen af arbejde, og de opgave-relaterede roller, i det ønskede systems sammenhæng [6].

7.1. Anvendelse

Målet med systemet er, at semesterkoordinatoren skal få en bedre forståelse af et semester ved at skabe et overblik over de leverede besvarelser. Derimod kan det have en negativ indflydelse, hvis denne generalisering ender med helt at erstatte den nuværende behandling og evaluering af individuel feedback.

Systemet udvikles som et værktøj til en bestemt opgave, som udføres af enkeltpersoner, på fortrolig data, som kun de må se. Der er derfor én primær gruppe aktører til systemet: Semesterkoordinatorer.

Brugsmønstre findes bedst ved at undersøge anvendelsesområdets opgaver, og udlede specifikke brugsmønstre fra det. De brugsmønstre som er blevet udledt til dette system handler om at få adgang til data og navigere gennem det.

Brugsmønstre

De følgende abstraktioner er blevet udviklet, for at bestemme hvilke brugsmønstre er nødvendige, for at give programmet dens nødvendige funktion, som en helhed:

Udvælg kommentar
Anvendelse: Kommentarer i systemets præsentation af dataene, vil have en knap til at tilføje kommentaren til brugerens liste af markerede kommentarer.
Objekter: Semesterkoordinator, Kommentar
Funktioner: Eksporter markeringer, Rediger markering.

Autentificering
Anvendelse: For at tilgå systemets kernefunktionalitet, autentificeres en bruger ved at indtaste sine loginoplysninger, før systemet viser informationerne knyttet til den givne bruger. Når brugeren er færdig med at bruge systemet kan de logge ud for at forhindre uvedkommende adgang til systemet.
Objekter: Semesterkoordinator
Funktioner: Verificer loginoplysninger, Afslut session.

7.2. Funktioner

Analysedata-repræsentation
Anvendelse: For hvert hold som en bruger har i systemet, laves der en analyse. Denne analyse resulterer i en række forskellige informationer, som hver især kan præsenteres for brugeren.
Objekter: Hold, Semesterevaluering, Besvarelse, Kommentar, Bedømmelse.
Funktioner: Importer analyseret data, Eksporter analyseret data, Importer udvalgt data, Vis udvalgt data.
Udvid studerende
Anvendelse: Brugeren kan vælge at se sine udvalgte kommentarer. Ved enhver kommentar, kan brugeren vælge at se andre kommentarer fra den samme studerende.
Objekter: Kommentar, Studerende
Funktioner: Vis udvalgt data.
Registrer hold
Anvendelse: Brugeren kan vælge at tilføje et nyt hold. De føres til en side hvor de kan indtaste de nødvendige oplysninger, hvormed et nyt hold kan oprettes. Brugeren kan senere redigere disse informationer, eller fjerne et hold fra systemet.
Objekter: Hold
Funktioner: Opret Hold, Rediger hold, Slet hold
Indsæt data
Anvendelse: Når et hold registreres, skal der uploades tilhørende evalueringsdata.
Objekter: Hold, Semesterevaluering
Funktioner: Indsæt rådata.

7.2 Funktioner

Anvendelsen af et system er påvirket af hvilke funktioner som systemet understøtter. Dette betyder, at funktioner er hvad der gør en model brugbar for aktørerne. I dette afsnit vil systemrelevante funktioner blive undersøgt - og hvis de er særligt komplekse - beskrevet i detalje. Teorien i dette afsnit baseres på Kapitel 7 i OOA&D-bogen [6].

7.2.1 Funktionstyper

Der er fire generelle funktionstyper, der kan bruges som et værktøj til nemmere at forstå den kontekst, som funktionen bruges i. Hver funktionstype har krav i

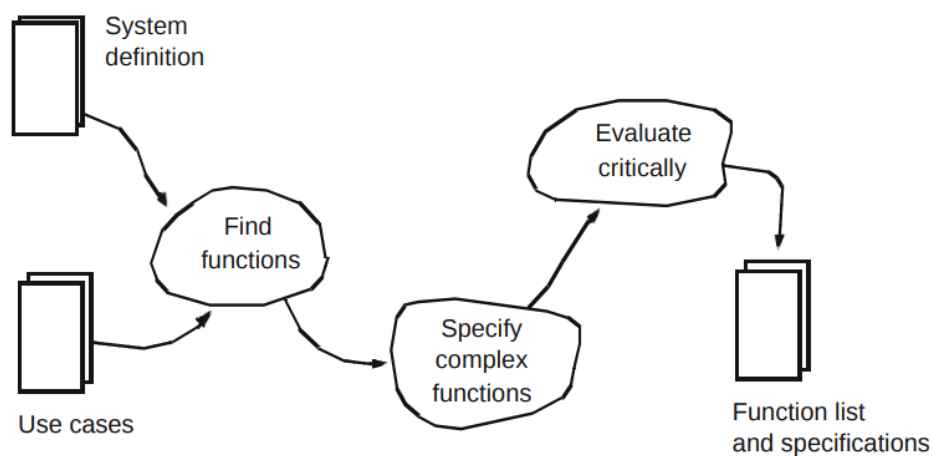
7.2. Funktioner

forhold til hvordan funktionen aktiveres, hvad den udfører, og hvilke resultater der er givet. Funktioner passer sjældent på præcis én funktionstype, og bliver derfor ofte en blanding af flere forskellige. De fire funktionstyper er beskrevet herunder:

- **Opdatering:** Aktiveret af en hændelse i problemområdet. Resulterer i en ændring af modellens tilstand.
- **Signal:** Aktiveret af en ændring i modellens tilstand. Resulterer i en reaktion i konteksten.
- **Læs:** Aktiveret af et krav om information fra en aktør. Resulterer i at systemet viser en relevant del af dens model.
- **Udregn:** Aktiveret af en aktørs efterspørgsel af information som ikke er kendt, men kan beregnes. Udregningen involverer information givet af modellen eller aktør. Resulterer i udregningens resultat.

Disse funktionstyper vil blive brugt i forbindelse med identifikationen af formålet med de enkelte funktioner.

Funktionsaktiviteten bygger videre på de brugsmønstre, der blev fundet i Afsnit 7.1, samt systemdefinitionen, som blev beskrevet i Kapitel 4. De enkelte delaktiviteter er illustreret i Figur 7.1.



Figur 7.1: Illustration af delaktiviteterne i funktionsaktiviteten [6]

Her fremgår det af funktionsaktiviteten, at der først skal identificeres funktioner, der er relevante for systemet. Nogle af disse funktioner kan have en høj kompleksitet, hvilket betyder, at de bør specificeres yderligere, for at forstå hvad der ligger bag. Det sidste der bør gøres, er at evaluere alle de identificerede funktioner. Dette kan blandt andet gøres ved at sikre en overensstemmelse mellem brugsmønstrene fra forrige delafsnit og de identificerede funktioner.

7.2. Funktioner

7.2.2 Find funktioner

For at finde relevante funktioner, tages der udgangspunkt i de centrale kilder til beskrivelsen af systemet. En sådan kilde kunne være kravspecifikationen, der blandt andet er udviklet på baggrund af systemdefinitionen. Disse funktioner kan altså udspringe fra krav, klasser, hændelser eller brugsmønstre, som alle tidligere er identificeret. På baggrund af dette blev de relevante funktioner fundet og dokumenteret i en funktionsliste i Tabel 7.1.

Funktionsnavn	Kompleksitet	Funktionstype
Lav Analyse	Meget kompleks	Opdatering
Importer udvalgt data	Kompleks	Opdatering
Eksporter analyseret data	Middel	Opdatering/Signal
Eksporter markeringer	Middel	Opdatering/Signal
Verificer loginoplysninger	Middel	Opdatering/Læs
Rediger markering	Middel	Opdatering
Indsæt rådata	Middel	Opdatering
Lav sammenligning	Middel	Opdatering
Importer analyseret data	Middel	Opdatering
Afslut session	Simpel	Opdatering
Opret hold	Simpel	Opdatering
Rediger hold	Simpel	Opdatering
Slet hold	Simpel	Opdatering
Vis udvalgt data	Simpel	Læs

Table 7.1: Funktionsliste

Funktionslisten i Tabel 7.1 specificerer navne på systemets funktioner, deres kompleksitet og funktionstype. Nogle af disse funktioner er komplekse, og funktionaltiteten af disse kan derfor være svære at forstå uden yderligere at specificere dem.

7.2.3 Specificer komplekse funktioner

På baggrund af Tabel 7.1 kan de mest komplekse funktioner nu udvælges og beskrives yderligere, ved at lave en opdeling af deres funktionelle krav. De mest komplekse funktioner i dette system er 'Importer udvalgt data' og 'Lav analyse', som vil blive beskrevet yderligere i Tabel 7.2 og 7.3.

Funktionen 'Importer udvalgt data' er kompleks, da den skal kunne finde noget udvalgt data. Stedet hvor den indhenter dataet fra, har meget mere data, end denne funktion skal hente, men det er opdelt i forskellige kategorier. Den skal derfor være i stand til at kunne identificere det korrekte data i en given kategori, og derefter indhente og samle det hele. Dette er vist i Tabel 7.2, hvor der er blevet lavet en

7.3. Afrunding

opdeling til passende delfunktioner.

Importer udvalgt data	Kompleks	Opdatering
Importer data fra hver kategori	Middel	Opdatering
Sorter udvalgt data efter kategorier	Simpel	Opdatering
Saml importeret data	Simpel	Opdatering

Table 7.2: Opdeling af funktionen: 'Indhent udvalgt data'

Funktionen 'Lav analyse' er meget kompleks, da den skal kunne gennemføre alle analyseaktiviteterne. Disse analyseaktiviteter er beskrevet som krav i Kapitel 5. De enkelte analyseaktiviteter er:

- Krav 3: Kategorisering af lignende kommentarer
- Krav 7: Sammenligning af kommentarer med tidligere årgange på samme semester
- Krav 8: Sentimentanalyse på de enkelte kommentarer
- Krav 9: Nøgleordsanalyse for at finde de ord der har størst betydning i positive og i negative kommentarer

Lav analyse	Meget kompleks	Opdatering
Importer rådata	Middel	Opdatering
Forbehandle rådata	Middel	Signal
Lav kategoriseringsanalyse	Middel	Opdatering
Lav sentimentanalyse	Middel	Opdatering
Lav nøgleordsanalyse	Middel	Opdatering

Table 7.3: Opdeling af funktionen: 'Lav analyse'

For at kunne lave analyserne, skal rådataen først importeres, hvorefter den kan forbehandles og bruges til analyserne. Alle disse opgaver laves til delfunktioner for 'Lav analyse', som illustreret i Tabel 7.3. Som det kan læses på tabellen, består funktionen af flere delfunktioner med en middel kompleksitet.

7.3 Afrunding

Aktiviteten af funktioner tillod at identificere funktioner, på baggrund af mange tidligere aktiviteter, og ydermere at kategorisere dem med de fire funktionstyper. Med disse funktioner vides nu mere specifikt hvordan diverse funktionelliteter skal

7.3. Afrunding

prioriteres.

Med de fundne funktioner på plads, bør vi nu undersøge hvordan arkitekturen af programmet skal struktureres. Et godt arkitekturdesign vil kunne hjælpe med at forstå hvordan program komponenter bør kommunikere.

Kapitel 8

Arkitekturdesign

Målet med at specificere arkitekturdesign er at lægge et solidt grundlag for udviklingen af et program til at implementere et system.

I denne sammenhæng følger dette kapitel teori fra "Object Oriented Analysis & Design." [6]

8.1 Kriterier

I dette afsnit undersøges kriterier og betingelser. Det kan ofte være besværligt at skelne mellem krav, kriterie og betingelse, da disse ofte overlapper hinanden. Et krav er en specifik funktionalitet som systemet skal kunne, hvor et kriterie er et mål der skal opnås til et bestemt niveau, et kriterie er derfor ikke entydigt, men befinder sig nærmere i et spektrum. Betingelser er derimod de ting som lægger fundamentet - men også begrænsningerne - for projektet, da det er her hardware-grundlaget eksempelvis specificeres.

Det gode design opfylder princippet; *Et godt design har ingen alvorlige svagheder*. For at opfylde dette princip, undersøges først de generelle kriterier fra bogen "Object Oriented Analysis & Design". Dette tillader opdagelsen af projektspecifikke kriterier der bør prioriteres i udviklingen af systemet, således at det gode design kan kvalitetssikres.

Før en kriterieprioritering kan finde sted, bør der undersøges hvilke designbetingelser projektet er underlagt, såsom hardwarekrav. Dette gøres for at skabe en bedre forståelse af systemets grundlag og begrænsninger. Ligeledes undersøges potentielle konflikter mellem de fundne projektspecifikke kriterier, da det ofte er tilfældet, at en opfyldelse af ét kriterie er en nedprioritering af et andet.

Undersøgelsen af projektspecifikke kriterier, konflikterne her iblandt og designbetingelser, bør herefter have skabt et bedre systemoverblik, hvilket til sidst anvendes til at udrede en prioriteret kriterieliste.

8.1. Kriterier

8.1.1 Brugbarhed, fleksibilitet, og forståelighed

Et godt system bør opfylde de generelle kriterier indenfor objekt-orienteret systemudvikling. Netop brugbarhed, fleksibilitet og forståelighed.

I dette delafsnit undersøges systemet på tværs af disse generelle kriterier for at finde relevante fokuspunkter, der senere anvendes til opstilling af designkriterier.

Brugbarhed

Dette underafsnit omhandler det første af de generelle kriterier; brugbarhed. Det har til formål at opstille brugerens behov i forhold til brugbarheden op imod den tekniske platform. Dette gøres for at analysere hvilke kriterier der er vigtige, mindre vigtige, og hvad der overordnet er teknisk muligt at prioritere.

Der ses på to komplementære principper; *Et designs opfyldelse af brugerbehov* og *Et designs opfyldelse af de tekniske betingelser*. Her indgår blandt andet brugerforventninger og et systems udnyttelse af de tilgængelige ressourcer.

Brugeren har en forventning om, at når der udføres en handling, reagerer systemet korrekt i forhold til den handling som blev udført. Det er vigtigt at brugeren kan have tillid til systemet, da dette gør systemet tiltalende, hvilket øger sandsynligheden for at brugeren vil anvende det. Dette kan opfyldes ved at alle handlinger som brugeren udfører, gemmes på systemets tilknyttede server, således at alt er synkroniseret.

Kriterierne for den tekniske platform af systemet skal tage højde for hvordan systemets ressourcer bliver anvendt. Systemet i dette projekt skal derfor fordele ressourceforbruget mellem klienten og den eksterne server.

Det er vigtigt at systemet fordeler ressourceforbruget, da dette kan forhindre at nogle handlinger har for stor belastning på en mindre del af systemet. Systemet i dette tilfælde skal forstås som et samarbejde mellem server og klienter. Dette kan opnås ved at decentralisere ressourceforbruget, så nogle opgaver udføres på den tilknyttede server, hvor andre udføres på enheden som brugeren anvender. Systemet skal anvende brugerens enhedsressourcer til at udføre primære opgaver, for at undgå unødigt belastning af serveren, og dermed udnytte alle tilgængelige ressourcer.

Fleksibilitet

Dette underafsnit undersøger det næste generelle kriterie, fleksibilitet. At et system er fleksibelt, betyder at ændringer til omkringliggende faktorer, som eksempelvis organisationen, har meget lille konsekvens for systemet.

Systemdesignere er tvunget til at stole på mangelfuld information, da ikke alle tekniske konsekvenser og krav kan undersøges, og er derfor ofte baseret på et usikkert

8.1. Kriterier

grundlag. Dette kan have store konsekvenser for systemet på længere sigt. Ved at have fokus på fleksibilitet i arkitekturdesign, kan disse konsekvenser forebygges, og en større påvirkning kan undgås. Det undersøges derfor i dette underafsnit, hvilke kriterier der bør lægges fokus på, således systemet kan opnå så stor en fleksibilitet som muligt.

En måde at forebygge ændringer på, er ved indkapsling. Indkapsling sker ved at opbygge diverse funktioner og principper separat og bygge systemet som en sammensætning af uafhængige byggeblokke. Dette tillader en nem måde at udskifte byggeblokkene senere. En ny byggeblok, er en blok der som udgangspunkt har det samme formål, men er implementeret anderledes, og dermed kan understøtte anden teknologi, eller et andet krav.

På baggrund af dette bør vores arkitektur bestå af enkeltstående og uafhængige komponenter, som muliggør let udskiftning.

Et velspecificeret og klart problem, hvor der på forhånd vides at ingen ændringer vil ske, kan tillade optimerede og specialiserede løsninger, men dette er sjældent tilfældet.

I semesterevalueringerne anvendes der i øjeblikket meget generelle spørgeskemaer, hvor de samme spørgsmål stilles for hvert kursus og projektet. Dette betyder at vi kan specificere hvilke datakolonner tilhører hvilket specifikt spørgsmål. En ændring af dette spørgeskema kan resultere i forkerte fremvisninger og potentielle designfejl. Derfor bør programmet kunne tage højde for denne potentielle ændring.

Forståelighed

Dette underafsnit undersøger det tredje og sidste generelle princip; forståelighed. Forståelighed dækker over flere emner: Systemoverblik, Abstraktion og Genanvendelse af designmønstre. Disse emner dækker alle forskellige aspekter, som hver tilbyder en unik indgang til systemets forståelighed, og ved at undersøge disse, kan der findes hvilke kriterier der i projektet bør fokuseres på.

Et system med god forståelighed gør det nemt at skabe et overblik over et ellers komplekst system. Et projekt med dårlig forståelighed, kan hurtigt risikere at alt for meget tid anvendes på at udvikle et delområde i systemet, hvor de andre tilsidesættes, eller glemmes. Det er derfor vigtigt at skabe og vedligeholde et systemoverblik. Dette bliver opnået gennem rapportens analysekapitler, samt ved kontinuerligt at dokumentere udvikling på systemet i rapporten.

Abstraktion i systemudvikling er en gruppering af lignende principper og systemer til et enkelt koncept. Der findes tre metoder til dette: Klassificering, Generalisering og Modularisering, hvilket svarer til koncepterne: Klasser, Generaliseringsstruktur

8.1. Kriterier

og komponenter.

Genanvendelse af designmønstre kan på mange forskellige niveauer hjælpe med forståeligheden, af et ellers komplekst system. Hvis et problem består af flere underproblemer, kan det ofte virke intuitivt at løse underproblemerne sekventielt. Derimod, hvis flere af disse underproblemer deler en fællesnævner, kan det hjælpe med forståeligheden hvis vi, i stedet for at udvikle mange specifikke løsninger, vælger at udvikle en fælles løsningsmetode, og dermed anvende denne på hvert nærliggende underproblem. Det er derfor vigtigt at en vis generalisering og genanvendelse af løsninger forsøges at blive opnået igennem systemet, for at få et bedre systemoverblik.

Opsummering

Igennem de foregående underafsnit om de generelle kriterier, er der i forhold til projektet, fundet frem til de følgende kriterier, som der bør lægges fokus på:

Brugbarhed: Fokus på en konstant opdateret model, fokus på en balanceret anvendelse af systemressourcer, og fokus på decentralisering af system arbejde.

Fleksibilitet: Fokus på enkeltstående og uafhængige komponenter og fokus på variable dataformat.

Forståelighed: Fokus på system overblik, og fokus på genanvendelse af design mønstre.

8.1.2 Specifikke designbetingelser

For at kunne prioritere designkriterier, kan der ikke blot analyseres hvilke kriterier der er og hvilken betydning de har. Det er også essentielt at analysere de specifikke designbetingelser, da nogle tekniske, organisatoriske og menneskelige betingelser kan være krævet, for at systemet kan anvendes korrekt og løse problemstillingen.

Med fokus på de tekniske designbetingelser, tydeliggøres det, hvad enhederne som systemet anvendes på, skal være i stand til at køre og håndtere. Her gælder det at systemet skal køre på en computer som har passende hardware samt operativsystem for understøttelse af Java SE.

De organisatoriske betingelser ligner mange af de krav som nævnes i kravspecifikationen. Derfor nævnes her blot at systemet udvikles, så alle danske love overholdes. Her er der særligt fokus på persondataloven.

Det menneskelige aspekt af systemets designbetingelser har ingen dybdegående betingelser ud over brugerens arbejdsstilling som semesterkoordinator. Derudover kan dette delafsnit også anvendes til at beskrive brugerens allerede eksisterende erfaringer med

8.1. Kriterier

brug af systemer, og hvilke designmæssige betingelser eller forventninger brugeren har gjort sig på forhånd.

Hvis en bruger har arbejdet meget med ligende systemer, har brugeren nogle forventninger om designet, men kan også hurtigt lære at anvende systemet, hvis de forventninger er mødt. Modsat kan brugeren mangle erfaring, og dermed have sværere ved at sætte sig ind i et nyt system.

Opsamlende er det endeligt muligt fastsætte, at både systemet og brugeren kræver at specifikke betingelser bliver opfyldt, for at systemet kan implementeres og anvendes.

- **Tekniske betingelser**
 - Systemet skal understøtte Java SE.
- **Organisatoriske betingelser**
 - Systemet udvikles, så alle danske love overholdes.
- **Menneskelige betingelser**
 - Brugeren er ansat som semesterkoordinator.
 - Systemudviklingen skal inddrage brugerens erfaring i designet.
 - Brugeren skal kunne læse og forstå dansk.

8.1.3 Konflikter

Det er ikke en mulighed at alle kriterier og betingelser kan blive opfyldt ligeligt. Dette forårsages af konflikter internt mellem de forskellige kriterier, tekniske betingelserne, og hvad brugeren forventer. For eksempel ønskes der fra den organisatoriske side at et system skal kunne produceres så billigt som muligt, hvilket virker som en begrænsende faktor for de andre kriterier. Hvis brugeren eksempelvis på samme tid ønsker at system skal have en lang række komplekse egenskaber, opstår der en konflikt mellem bruger og organisation. Ligeledes findes der konflikter mellem enkle kriterier som for eksempel effektivitet og fleksibilitet, hvor den ene vil modarbejde den anden.

I dette delafsnit vil der blive undersøgt konflikter mellem de nyfundne kriterier og betingelser for systemet, og denne vej igennem, skabe et overblik, så en bedre prioritering af kriterier kan blive dannet.

Kriteriekonflikter

I dette underafsnit undersøges konflikter mellem kriterierne fra Delafsnit 8.1.1. For at skabe et overblik over konflikter, udredes først en tabel over kriterierne og hvor

8.1. Kriterier

konflikter kan opstå, hvilket kan ses i Figur 8.1. På baggrund af disse identificerede konflikter, vil dette underafsnit bestå af en kort analyse af hver konflikt, og en vurdering af den potentielle påvirkning af disse konflikter.

Kriterie konflikter	#1	#2	#3	#4	#5	#6	#7
#1 Fokus på en konstant opdateret model.	/	X				X	
#2 Fokus på en balanceret anvendelse af systemressourcer.	X	/					
#3 Fokus på decentralisering af systemarbejde.			/			X	
#4 Fokus på enkeltstående og uafhængige komponenter.				/			
#5 Fokus på variable dataformat.					/		
#6 Fokus på systemoverblik.	X		X			/	
#7 Fokus på genanvendelse af designmønstre.							/

Figur 8.1: Tabel over kriteriekonflikter.

Konflikten mellem #1 og #2:

Konflikten mellem "*#1 Fokus på konstant opdateret model*" og "*#2 Fokus på en balanceret anvendelse af systemressourcer*" har sit grundlag i at kriterie #1 ønsker at holde en konstant ekstern model synkroniseret, hvilket kræver mange opdateringer til sådan en model. Derimod er ikke alle opdateringer nødvendige. Eksempelvis, behøver modellen ikke opdateres i samme øjeblik som ændringen sker, da flere ændringer kan grupperes og overføres på én opdatering. Dette vil spare systemressourcer som opfylder kriterie #2's mål.

En opfyldelse af ét kriterie opponerer derfor opfyldelsen af det andet kriterie. For at opnå en balance skal det vurderes hvor ofte sådanne opdateringer skal overføres til modellen, så en opdateret model er opbevaret. Derudover bør overbrug af vigtige systemressourcer på unødvendige opdateringer reduceres.

Konflikten mellem #1 og #6:

Konflikten mellem "*#1 Fokus på konstant opdateret model*" og "*#6 Fokus på systemoverblik*" har sit grundlag i, at hvis systemet skal opdatere den eksterne model efter hver ændringer, kan det betyde at der vil blive designet mange forskellige små opdateringer, således at kriterie #1 kan blive opfyldt. Det ville derimod skabe et simplere system hvis disse opdateringer ikke skulle finde sted flere forskellige steder, men i stedet centralt.

8.1. Kriterier

Konflikten vurderes her at være af lav konsekvens eftersom graden af systemforståelighed der mistes ved at opfylde kriterie #1 er minimal. For optimal balancering af disse kriterier skal der vurderes hvorvidt en central og generel opdateringskomponent skal designes således at systemoverblikket forholdes så overskueligt og forståeligt som muligt.

Konflikten mellem #3 og #6:

Konflikten mellem "*#3 Fokus på decentralisering af systemarbejde*" og "*#6 Fokus på systemoverblik*" har sit grundlag i at et centraliseret system ofte er simplet at forstå og implementerer, hvorimod decentraliserede systemer ofte har den omvendte effekt, hvilket resulterer i en kriterie konflikt.

Ved denne konflikt er der vurderet at kriterie #3 er af større betydning end #6. Derfor er det blevet valgt at et decentraliseret system vil blive designet, men der skal stadig ligge fokus på at holde relevante dele centrale således at et simpelt overblik kan vedligeholdes.

Betingelseskonflikter

Udviklingen af systemet har ikke fået nogle specifikke designbetingelser fra brugeren via de afholdte interviews og det har derfor været i analysen at de nødvendige betingelser findes og behandles. Dette resulterer i mange mulige løsninger, hvilket giver mere fokus på kriterier frem for betingelser. De angivne betingelser eksisterer uafhængigt af hinanden og danner ikke konflikter. Dette vil betyde at alle konflikter kan overholdes og at der ikke bør tages yderligere forbehold til dette.

8.1.4 Prioritering af kriterier

Som beskrevet tidligere i afsnittet, er et godt systemdesign balanceret i dets prioriteter, så der ikke udvikles et for stort eller for småt system. Dette medfører at der bør laves en prioritering af de kriterier, som anvendes i designet af systemet.

Tabel 8.1 viser prioritetsfordelingen for kriterierne som påvirker systemudviklingen. Her tydeliggøres vigtigheden af de individuelle kriterier, så der dannes et overblik over balancen af kriterierne. I konstruktionen af prioriteringerne, har der været fokus på en virkelighedsnær tilgang, i form af at inddrage de beskrevne konflikter.

Det ses ud fra tabellen, at det er vigtigt at systemet kan tilpasse sig organisatoriske, arbejdsrelaterede og tekniske kontekster, da dette er med til at sikre sig, at mindre ændringer i systemets funktionalitet og form, ikke ændrer kernefunktionalitetens endelige resultater. Dette understøtter også prioriteten af systemets pålidelighed, som er systemets evne til at skabe pålidelige resultater, som brugeren kan stole på er

8.1. Kriterier

	Meget vigtig	Vigtig	Mindre vigtig	Irrelevant	Let opfyldt
Brugbar (Usable)	X				
Sikker (Secure)		X			
Effektiv (Efficient)			X		
Korrekt (Correct)			X		
Pålidelig (Reliable)	X				
Vedligeholdelsesvenlig (Maintainable)			X		
Testbar (Testable)				X	
Fleksibel (Flexible)			X		
Forståelig (Comprehensible)		X			
Genanvendelig (Reusable)			X		
Transportabel (Portable)					X
Indbyrdes kompatible (Interoperable)		X			
Fokus på konstant opdateret model			X		
Fokus på balanceret anvendelse af systemressourcer					X
Fokus på decentralisering af systemarbejde			X		
Fokus på enkeltstående og uafhængige komponenter			X		
Fokus på variable dataformat		X			
Fokus på systemoverblik			X		
Fokus på genanvendelse af design-mønstre			X		

Table 8.1: Prioriteringstabel over klassiske og nyfundne kriterier, baseret på Figur 9.4 fra OOA&D [6]

korrekte.

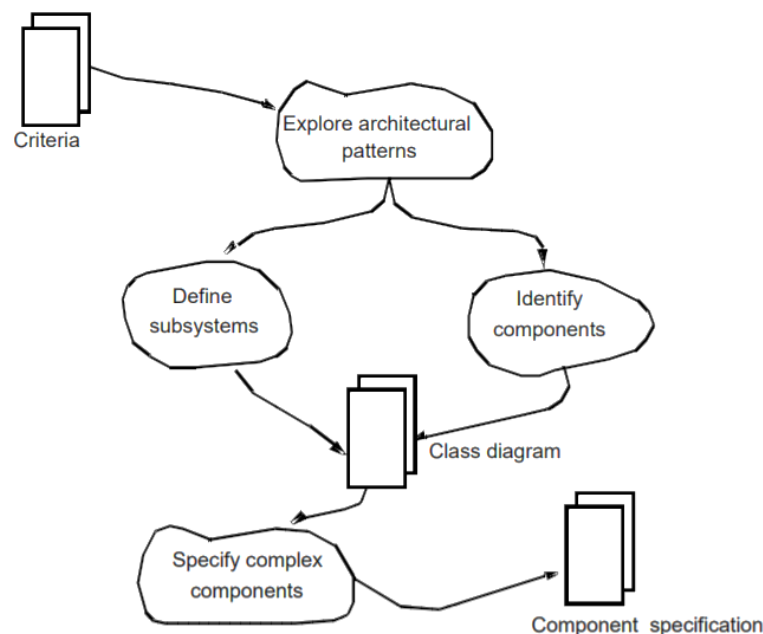
Hvorvidt det at kunne teste programmets funktionalitet, efter udviklings-relevante tests, prioriteres som irrelevant for systemets primære funktioner. Det antages at programmet som udgangspunkt ikke ændres uden yderligere teknologisk udvikling.

Det vises også at den bekostning som det har at overdrage systemet fra en teknisk platform til en anden, vil kunne håndteres let, og har derfor ikke en prioritet. Dette gælder også balanceringen af systemressourcerne, som i systemets omfang, ikke vil have stor ressourceomkostning, hvilket betyder at der kan være mindre fokus på dette og at det er let håndterligt.

8.2 Komponenter

En Komponentarkitektur er en måde at strukturere et system, ved at opdele det i flere komponenter. Formålet med at skabe et program opdelt i komponenter er at gøre programmet overskueligt og klart opdele dets ansvarsområder. Fordelen ved at have klart opdelt ansvarsområder i et system, er at det nedsætter kompleksiteten mellem komponenter, eftersom hvert komponent kun bør bekymre sig om at løse én opgave.

Komponentarkitektur-aktiviteten består af en række aktiviteter, som resulterer i et nyt klassediagram bestående af komponenter, delsystemer og afhængigheder imellem dem, samt en specifikation af de enkelte komponenter. Den overordnede sammenhæng mellem disse delaktiviteter er illustreret i Figur 8.2. Her fremgår det at det første der skal undersøges er relevante arkitekturmønstre. Disse arkitekturmønstre består af forskellige komponenter og delsystemer, der i sammenhæng danner et klassediagram. I dette klassediagram kan udvalgte komplekse komponenter specificeres, for at få en forståelse for komponentens ansvar, afhængigheder og sammenhæng med konteksten.



Figur 8.2: Illustration over delaktiviteter i komponentarkitektur-aktivitetencite [6]

En komponent er en samling af systemets dele, og har et specifikt defineret ansvarsområde. Et eksempel på dette kan være en komponent som er ansvarlig for at foretage analyse af evalueringer, eller håndtere login efterspørgsler.

Alle komponenter udgør systemet som en helhed. Det mindst abstrakte en komponent kan repræsentere er en klasse, og det mest abstrakte er et system.

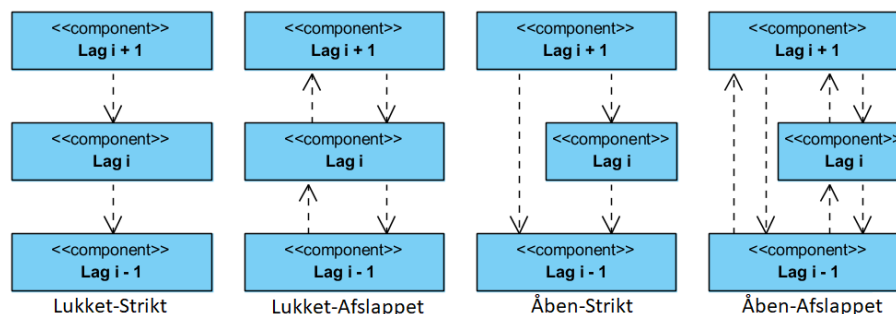
8.2. Komponenter

En komponent kan indeholde flere delkomponenter, og begrænser hvordan forskellige komponenter kan interagere med hinanden. Dette reducerer programmets indre afhængigheder, som øger modularitet. Dette resulterer i at mindre ændringer i systemet ikke kræver ændringer i hele systemet.

8.2.1 Find arkitekturmønstre

En måde at kontrollere hvordan komponenter kan interagere med hinanden, er at opdele et system i lag. Hvert lag består af en række komponenter. For at kontrollere hvordan lagene interagerer med hinanden, bruges arkitekturmønstre som kan være åbne eller lukkede, samt strikte eller afslappede. Kombinationen af disse er illustreret i Figur 8.3.

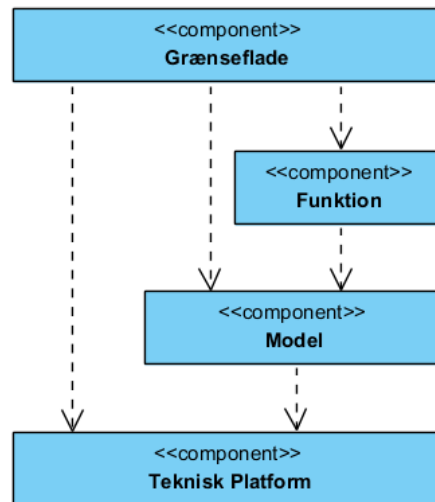
- **Lukket** arkitektur: Der kan kun bruges operationer fra laget lige over og lige under det givne lag.
- **Åben** arkitektur: Der kan bruges operationer fra alle andre lag.
- **Strikt** arkitektur: Der kan kun bruges operationer fra lag nedenfor.
- **Afslappet** arkitektur: Der kan bruges operationer fra lag både oppe fra og nede fra.



Figur 8.3: De 4 varianter af lagbaserede arkitekturer, med udgangspunkt i OOA&D Figur 10.3 [6]

En måde at bruge denne lagbaserede arkitektur på, er at modellere anvendelsesområdetets interaktion med problemområdet. Dette refereres til som det generiske arkitekturmønster, og er illustreret i Figur 8.4. Her fremgår det at det generiske mønster danner et 'Åben-Strikt' arkitekturmønster, hvilket betyder at operationer kun kan bruges fra nedenstående lag, men at det ikke er begrænset til laget lige under det givne lag.

8.2. Komponenter



Figur 8.4: Illustration over det generelle generiske arkitekturmønster, baseret på Figur 10.5 fra OOA&D [6]

'Grænseflade'-komponenten og 'Funktion'-komponenten i det generiske arkitekturmønster beskriver anvendelsesområdet, og 'Model'-komponenten modellerer problemområdet. Den tekniske platform er en udvidelse og indkapsulering af den grundliggende tekniske platform. Eftersom dette arkitekturmønster er bygget op omkring idéen om et problemområde og anvendelsesområde, vælges det som det overordnede mønster der bygges videre fra.

8.2.2 Find Delsystemer & Komponenter

Et delsystem er resultatet af at opdele et større system i flere dele. Samlingen af alle delsystemer skal dermed stadig være det originale system. Formålet med at opdele systemet i delsystemer, er at reducere kompleksiteten og opdele ansvaret for systemet. Et delsystem er en form for komponent som indeholder flere komponenter. Når der findes delsystemer og komponenter, kan det gøres med henblik på at de skal passe ind i enten model, funktion, eller grænseflade.

En måde at identificere komponenter i et system på, er at tage udgangspunkt i systemets grundlæggende lagbaserede struktur, som består af: Grænseflade-, funktion-, og modelkomponenter. Fordi komponenter bør have et klart-defineret ansvarsområde, er det muligt at nogle bekymringer ikke er relevante, og dermed ikke er inkluderet i komponentens design.

Model-komponenters primære ansvar er at opbevare objekter i problemdomænet, og dermed repræsentere problemområdet.

8.2. Komponenter

Funktion-komponenters primære ansvar er at give funktionalitet til en del af modellen. Disse komponenter tillader funktionalitet på problemområdet.

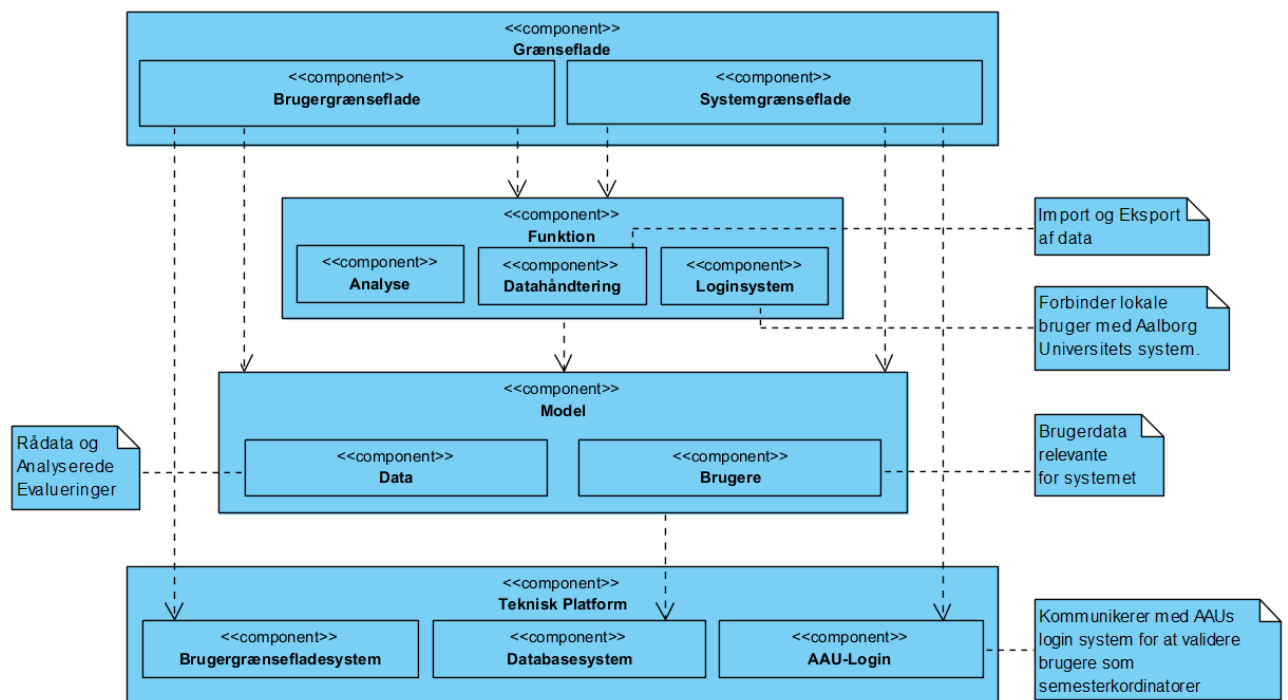
Grænseflade-komponenters primære ansvar er at tillade interaktionen mellem funktionalitet og systemets aktører. Disse komponenter skal tillade brugere at kunne bruge systemets funktioner.

Den generiske arkitektur opdeler disse 3 bekymringer i forskellige lag, som hver er et delsystem. Dette resulterer i at ét lag hovedsageligt indeholder model-centrale komponenter, et andet lag indeholder funktion-centrale komponenter, og et tredje lag indeholder grænseflade-centrale komponenter. Det sidste lag i den generiske arkitektur er den tekniske platform, som indeholder komponenter der opfylder tekniske krav som er krævet af højere lag, men ikke opfyldt af dem. Komponenter i dette lag er ofte eksterne biblioteker og uafhængige systemer.

Baseret på beskrivelsen af problemområdet, og derigennem modellen, samt funktioner og kriterier, kan systemets komponentarkitektur opdeles som illustreret i Figur 8.5.

Komponenter kan opnå en højere kompleksitet, som vil gøre klasse-diagrammer besværlige at forstå. Dermed kan en komponent blive beskrevet og detaljeret ved brug af CRC-kort som vedhænger komponenten. Et CRC-kort i denne sammenhæng, er en tekst note der kan detaljere en komponents ansvar, afhængighed, og specificerer dens forhold til konteksten.

8.2. Komponenter



Figur 8.5: Klassediagram af fundne delsystemer sat ind i den generiske arkitektur model.

Beskrivelse af komponenters ansvar

Komponenternes ansvar kan fordeles i forskellige kategorier alt efter deres egenskaber og betydning.

Grænseflade

- *Brugergrænseflade*: Komponentens ansvar er strukturen af brugerfladen.
- *Systemgrænseflade*: Grænsefladen ansvarlig for eksterne systemer har adgang til systemet.

Funktion

- *Analyse*: Komponentens ansvar er at analysere evalueringer og udregne resultater ud fra de data.
- *Datahåndtering*: Er ansvarlig for at import og eksport af data i databasen.
- *Login System*: Systemets behandling af login forsøg og forbindelse mellem lokale brugere med Aalborg Universitets system.

Model

- *Evalueringer*: Komponentens ansvar er rådata og analyserede evalueringer.

8.2. Komponenter

- *Brugere*: Denne del af modellen er ansvarlig for brugerdata.

Teknisk Platform

- *Brugergrænseflade system*: En Komponent som er ansvarlig for implementere grænsefladens design.
- *Database system*: Komponenten ansvarlig for opbevare modellen.
- *AAU Login*: Komponenten er ansvarlig for at bruge Aalborg Universitets login system til at validere brugere som semesterkordinatore, som blev bestemt i Krav 14.

Klient-Server arkitektur

Arkitekturen udnytter et database system som kan implementeres på en ekstern server. Klient-server arkitektur har flere former, og er baseret på hvordan klienter kommunikerer med den centrale server. Eksempler på Arkitekture og deres navne kan ses på Figur 8.2[6]. *G* repræsenterer grænseflade, *F* betyder funktioner, og *M* er modellen.

Klient	Server	Arkitektur
G	G, F & M	Distribueret præsentation
G	F & M	Lokal præsentation
G & F	F & M	Distribueret funktionalitet
G & F	M	Centraliserede data
G, F & M	M	Distribueret data

Table 8.2: Forskellige former for fordeling i en klient-server arkitektur, baseret på Figur 10.9 fra OOA&D [6].

For det nuværende system, ville en klient-server arkitektur være struktureret som centraliseret data. Dette betyder at serverens eneste ansvar er at opbevare data om modellen. Klienterne behøver heller ikke at kommunikerer med hindanden, siden ingen funktionalitet kræver direkte samarbejde mellem klienter.

8.3 Opsummering

Vi ved nu hvordan programmet bør struktureres og bearbejdes i forhold til kriterier, samt systemkomponenter. Generelle og projektspecifikke kriterier er blevet vurderet og balanceret for at skabe en balanceret vægtfordelling mellem diverse kriterier.

I sammenspil med disse kriterier er der blevet udviklet et første udkast til et komponentdesign som projektet bør følge. Ved at følge den designede komponentarkitektur vil projektet opfylde ønsker om modularitet og komponent kommunikation.

Vi bør nu yderligere specificerer komponentarkitekturen, for at gå fra princip til specifik løsningsorienteret arkitektur med diverse klasser og funktioner placeret i komponentarkitekturen. Dette vil skabe et mere detaljeret komponentdesign at arbejde med.

Kapitel 9

Komponentdesign

I dette kapitel bliver komponenterne, samt komponentarkitekturen, fra Kapitel 8 yderligere designet, for videre at kunne beskrive systemets komponenter. Dette opnås ved at undersøge modelkomponenter, funktionskomponenter og forbindelse mellem komponenter i dette kapitels tre afsnit, hvoraf teorien bag afsnittene stammer fra Del 5 af "Object Oriented Analysis & Design" [6]:

- Modelkomponent: Hvordan er modellen repræsenteret som klasser i systemet?
- Funktionskomponent: Hvordan er funktionerne implementeret?
- Forbind komponenter: Hvordan er komponenterne forbundet?

9.1 Modelkomponent

Dette afsnit vil gennemgå modelkomponentaktiviteten. Denne aktivitet resulterer i et revideret klassediagram, som kan bestå af nye klasser, attributter og strukturændringer.

9.1.1 Design af Modelkomponenten

Modelkomponenten er implementationen af problemområdet i systemet. Denne er ansvarlig for at opstille en opdateret model af problemområdet. På baggrund af dette er en god struktur central for modelkomponenten, og nødvendig for at gøre denne nem og hurtig at arbejde med. Når modelkomponentaktiviteten er færdig, skal den give et overblik over hændelserne, på en måde der er tæt relateret til objektorienterede programmeringssprog. Modelkomponentens ansvar, er at opbevare en opdateret repræsentation af problemområdet.

9.1.2 Repræsentation af Hændelser

For at lave en repræsentation af de allerede eksisterende hændelser fra hændelsestabellen 6.5, kan de to hændelsestyper anvendes; private og fælles.

Private hændelser er hændelser, der kun kan ske for én klasse fra problemområdet. Der findes to måder at repræsentere en privat hændelse:

1. Ved hændelser der kun kan ske én gang, skal der tildeles en ny værdi til klassens tilstands attribut.
2. Ved hændelser der kan iterere, repræsenteres disse hændelser som en ny klasse, som aggregerer til den klasse hændelsen kom fra. For hver iteration, skal systemet skabe et nyt objekt af klassen. Derudover skal hændelsesattributterne integreres i den nye klasse.

Fælles hændelser er hændelser, der involverer mere end en klasse fra problemområdet. Ligeledes, som ved private hændelser, findes der to måder at repræsentere en fælles hændelse på:

1. Hvis hændelsen er repræsenteret i tilstandsdiagrammet på forskellige måder, skal den repræsenteres i relation til den klasse der giver den simpleste repræsentation.
2. Hvis hændelsen er involveret på den samme måde, da skal mulige repræsentationer vægtes mod hinanden.

Det er nu muligt at tilføje de repræsenterede hændelser fra det forrige kapitel, i et revurderet klassediagram. Hændelsestabel 6.5 fra Kapitel 6, indeholdte kun fælles hændelser. Dette betyder at det kun er disse, der kan blive repræsenteret. Hændelserne bliver i dette tilfælde repræsenteret ved attributter i eksisterende klasser. Det gamle klassediagrams klasser kan ved hjælp af repræsentationen af hændelserne restruktueres, for at forsimple klassediagrammet, uden det mister betydning.

9.1.3 Restrukturering af Klasser

Udover repræsentation af hændelser, skal det nye klassediagram også indeholde en simplificeret udgave af det tidligere klassediagram, se Figur 6.3. Der findes flere måder at forsimple et klassediagram på, hvoraf to af dem vil blive beskrevet nedenfor:

Generalisering

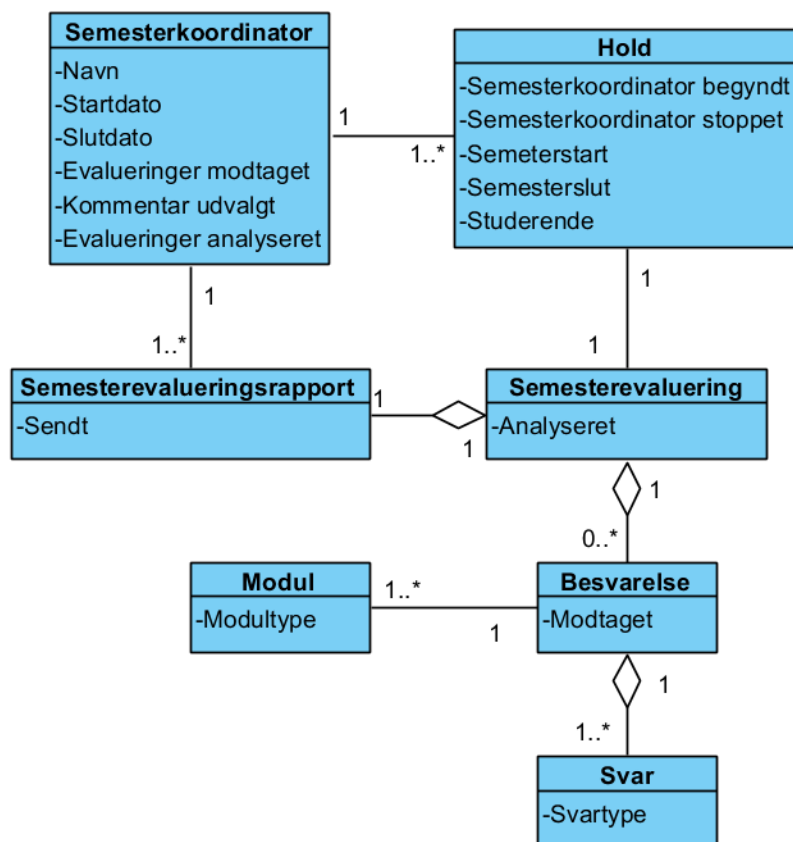
Hvis der findes to eller flere klasser, der har samme attributter og den samme klasse aggregerer fra dem alle, da kan klasserne sammenlægges til en samlet klasse. Denne nye klasse indeholder de attributter, som de sammenlagte klasser, samt en attribut, som skelner mellem de forskellige klasser.

Association

Når der tilføjes nye klasser og strukturer, skal der overvejes om gamle strukturer er overflødige. For eksempel hvis der haves et klassediagram med to klasser med en associationsstruktur i mellem dem. Da hvis der tilføjes en ny klasse, der aggregerer til de gamle klasser, skal der overvejes hvorvidt den gamle association er nødvendig.

Delkonklusion

Ud fra den opnåede viden omkring repræsentation af hændelser, og måderne hvorpå der kan simplificere et klassediagram på, da kan det initierende klassediagram revideres.



Figur 9.1: Revideret klassediagram

I Figur 9.1 ses det revideret klassediagram. I forhold til det første klassediagram (se Figur 6.3), udover de nye attributter, da er Kursus og Projekt blevet lagt ind under Modul, som nu indeholder en Modultype attribut. Kommentar og Bedømmelse er blevet generaliseret, og derfor slået sammen i én klasse, kaldet Svar, som har en

attribut med Svartype. Klassen; analyseret evaluering er fjernet, og findes nu som attribut i Semesterevaluering. Studerende er ligeledes blevet fjernet og optræder nu som attribut indeni Hold.

9.2 Funktionskomponent

Formålet med funktionskomponenten er at skabe et link mellem model og brug. Funktionskomponenten bør derfor give brugerinterfacet og andre systemkomponenter adgang til modellen, og er derfor den del af system, der implementerer de funktionelle krav.

I OOA&D beskrives adfærd, som operationer på systemets klasser. Derfor skal alle funktioner implementeres, som operationer på disse klasser, således at de kan aktiveres igennem klassens objekter.

En funktion beskriver ekstern observerbar adfærd, som relaterer til brugerens arbejde. En funktion implementeres som operationer i systemets klasser, og er derfor aktiveret igennem klassen, som udfører en databehandling og returnerer til kaldepunktet med eller uden et resultat.

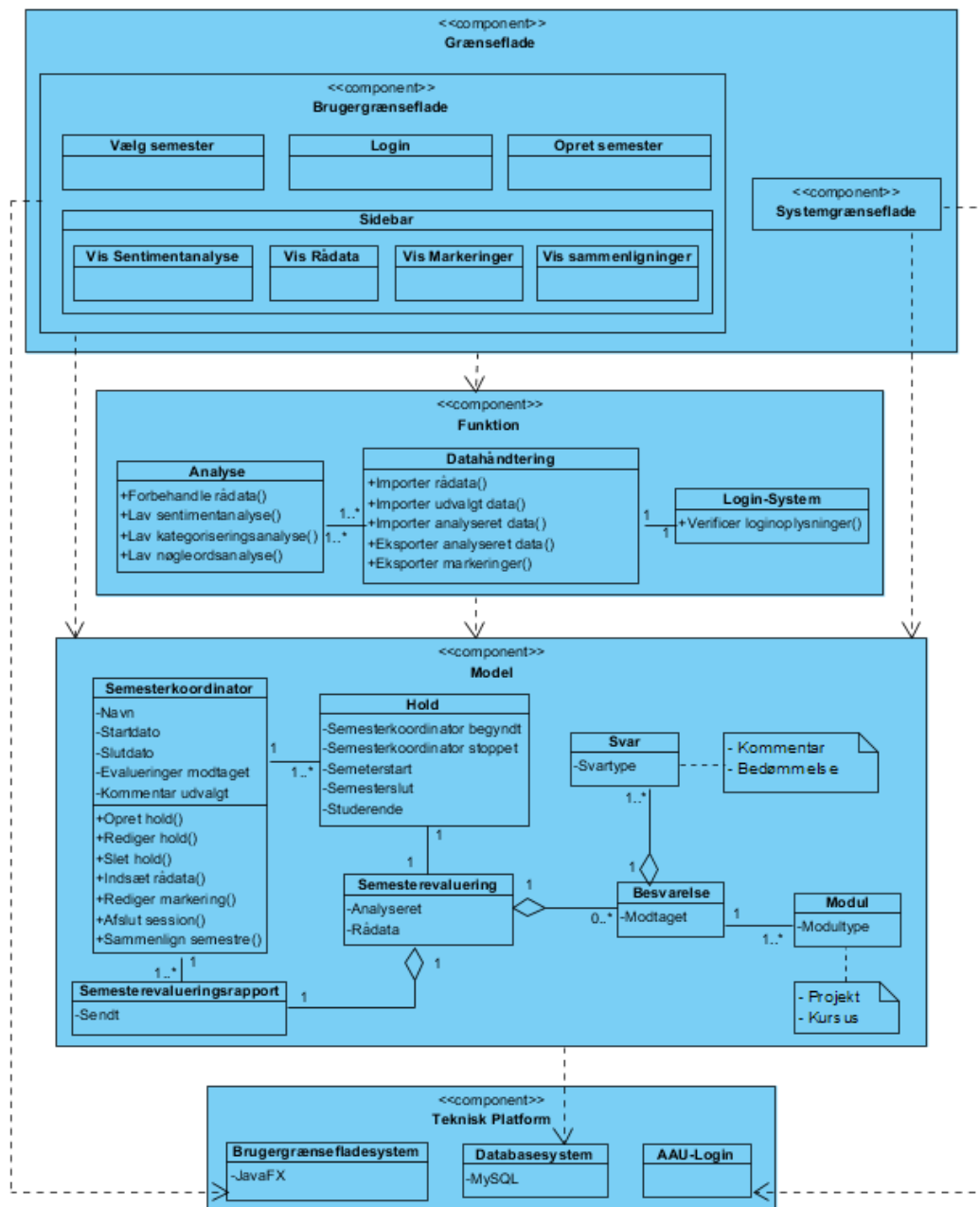
I design af funktioner forsøges det at identificere hvilke operationer, som kræves implementeret for en given funktion. Dette afhænger af funktionstypen, og et godt design princip er derfor baseret på ens funktionstyper.

Det er vigtigt i designfasen at begrænse sig til at designe funktioner, og ikke programmere dem. Dette betyder, at under design bør der ikke bekymres om den mindste detalje, fordi dette ofte kan forsage komplicerede designs. Derfor bør der ikke tages flere valg angående design end højst nødvendigt, for at undgå essentiel tvivl om selve implementering og derfor anbefales også at specificere komplekse operationer.

Funktionerne i funktionskomponenten er lavet med udgangspunkt i funktionsaktiviteten i analysen af anvendelsesområdet i Afsnit 7.2. Disse indsættes i klassediagrammet fra Figur 8.4 som bruger det generiske arkitekturdesign, og i modelkomponenten indsættes det reviderede klassediagram fra Figur 9.1. Disse ændringer og tilføjelser resulterer i Figur 9.2, som danner det endelige klassediagram.

I Figur 9.2 kan det ses hvordan de forskellige 'Views' i brugergrænsefladen er vist, hvordan funktioner funktionskomponenten er inkluderet, og hvordan det reviderede klassediagram af modellen er sat i modelkomponenten.

9.3. Afrunding



Figur 9.2: Endeligt klassediagram med udfyldte komponenter og funktioner

9.3 Afrunding

Design af komponenter har resulteret i et klassediagram, som kan ses på Figur 9.2. Dette diagram indeholder komponenter og klasser, samt relationen mellem disse. Det er dette klassediagram, som systemet bliver implementeret ud fra.

9.3. Afrunding

En del af klassediagrammet er afhængig af brugergrænsefladen, som vil udvikles i følgende kapitel.

Kapitel 10

Brugergrænsefladedesign

I dette kapitel bliver de forskellige udviklingstrin for brugergrænsefladedesign beskrevet og anvendt i projektets kontekst. Først gennemgås hvad en prototype kan gøre for design-processen, hvorefter den kan blive testet og evalueret, for til sidst at kunne designe en brugergrænseflade, der forbedrer brugerens muligheder, for at danne sig et overblik over semesterevalueringerne. Dette kapitals teoretiske grundlag baseres på bogen *Designing Interactive Systems* [11].

Envisionment-princippet siger at et design skal visualiseres for at tydeliggøre designerens idéer, samt tillade andre at evaluere dem. Dette kan gøres gennem flere forskellige medier, men her vælges det at bruge en prototype. Det følgende afsnit omhandler derfor prototyper og præsentationsformer af udviklingsprocessen. Når en passende præsentationsform er valgt, vil forskellige designovervejelser blive undersøgt, for at kunne designe prototypen, hvilket vil blive undersøgt i det efterfølgende afsnit. Disse designovervejelser kan så endelig testes og evalueres, for til sidst at opnå et design af brugergrænsefladen, der opfylder både brugergrænsefladekravene fra Delafsnit 5.2.3, og det revurderede prototypedesign.

10.1 Prototype

I dette afsnit er forskellige aspekter af valget af en prototypemodel undersøgt.

10.1.1 Formål

Prototyper er nødvendige for at kunne visualisere design-idéer, og tillade andre at teste og evaluere det, inden den endelige brugergrænseflade implementeres, således at designfejl eller mangler opdages i opløbet. Fordelen ved bruge tid på at udvikle en prototype er, at den tager relativt kort tid at udvikle, og hvis der i udviklingsprocessen opdages fejl, vil det ikke være nogen stor opgave at rette dem eller notere sig fejlen i designet. Når designet så senere implementeres i den endelige brugergrænseflade, er der sparet ressourcer ved ikke senere at skulle ændre ting, hvilket kan vise sig at

10.1. Prototype

være en større opgave. Prototypen lader altså designet færdigudvikle sig i et miljø, hvor det ikke koster ekstra tid og ressourcer at ændre det.

10.1.2 Forskellige prototypenøjagtigheder

Der findes overordnet set to slags prototyper, Hi-Fi (Høj nøjagtighed) og Lo-Fi (Lav nøjagtighed). De har forskellige egenskaber og er anvendelige i forskellige situationer.

Hi-Fi er en prototype der ligner, og opfører sig meget som det endelige produkt. Med software kunne det være en skitse af strukturen af det endelige system, hvor det er muligt at navigere, som det ville gøres i det endelige med mus og tastatur. Fordelen ved en Hi-Fi prototype, er dens høje detalje og lighed med det endelige produkt. Dog kan denne lighed med det endelige produkt også være en ulempe, da det kan risikeres, at nogle brugere ser det, som det færdige produkt, og hvis der derfor er små fejl i eksempelvis navne, kan dette stjæle fokus fra hvad brugeren egentlig skal have ud af en prototype. En anden ulempe med en Hi-Fi prototype er udviklingstiden, der potentielt bruges på at lave noget, som senere skal revideres.

Lo-Fi er en meget simpel prototype, og anvendes ofte når designet ikke er helt så udviklet. Når der arbejdes med software, vil en Lo-Fi prototype typisk bestå af papir eller karton, hvor grænsefladeelementer er tegnet. Ved evaluering vil input typisk være simuleret gennem tale og berøring, hvor papirerne så bliver udskiftet. Dette simulerer brugen af programmet i forhold til hvordan brugeren agerer. Fordelen med Lo-Fi, er den lave arbejdsbyrde det kræver for at udvikle den. En ulempe kan dog være, at det i en sådan prototype kan være svært at få en fornemmelse for de mere detaljerede dele af grænsefladen.

På baggrund af argumenterne for og imod de to former for prototyper, blev Balsamiq valgt som udviklingsværktøj til at skabe prototypen. Dette værktøj tillader os også at udvikle den på et stadie mellem Lo-Fi og Hi-Fi. Elementerne i Balsamiq er vist på et skitseniveau, som strukturmæssigt ligner et færdigt program, men visuelt er det tydeligt, at det kun er en prototype. Prototypen har ikke noget underliggende funktionalitet, hvilket betyder at det viste data er statisk. Den eneste dynamik i prototypen er en form for slideshow, der tillader brugeren at skifte mellem disse slides, ved at trykke på udvalgte knapper, der har links til andre slides.

10.1.3 Funktionalitetsgrad i prototyper

Der kan arbejdes med to forskellige funktionalitetsgrader i prototyper. Herunder er de begge opstillet med deres fordele og ulemper beskrevet.

- **Horisontal Prototype** er en prototype, der dækker hele brugergrænsefladen, men ikke indeholder den mere dybdegående funktionalitet. Dette betyder at data for eksempel er direkte kodet, og ikke ændrer sig fra gang til gang.

10.2. Design

En sådan prototype er god til at evaluere grænseflade design, men ikke reel funktionalitet.

- **Vertikal Prototype** afdækker en lille del af produktets funktionalitet. Det vil sige, at den anvendes til at teste kritiske dele af brugergrænsefladen, sammen med en kritisk del af backend funktionalitet, der er nødvendig for denne. Denne slags prototype giver ikke feedback i forhold til hele designet, men kun de få komponenter, der rent faktisk testes. Den kan bruges til gradvis evolution af programmet, hvor der laves vertikale prototyper af forskellige dele af programmet, der så sættes sammen og til sidst udgør det færdige produkt.

Prototypen har til fordel at vise brugeren, hvordan at brugergrænsefladen tænkes at blive designet på daværende tidspunkt i udviklingsprocessen. Prototypen er udviklet, som en horisontal prototype, der ved at brugeren tester det, bruges til at finde potentielle mangler i produktets design.

10.2 Design

Dette designafsnit vil primært rette sig mod design af prototypen, men eftersom brugergrænsefladen baseres på prototypen, vil brugergrænsefladen derigennem også blive designet. Prototypen bliver indledningsvist designet med henblik på de 12 designprincipper og WIMP-modellen, hvilket senere tillader os at evaluere i hvilken grad prototypen opfylder disse.

10.2.1 De 12 designprincipper

Et godt brugergrænsefladedesign for et system, bør opfylde de 12 design principper [11].

De 12 principper er:

1. **Synlighed:** Brugeren skal kunne se hvilke muligheder der er, og hvad systemet har gang i. Overvej synlighed via. lyd og berøring.
2. **Konsekvent:** Systemet skal være konsekvent med brug af design features, både konceptmæssigt og fysisk.
3. **Genkendelighed:** Brugeren skal kunne genkende anvendte symboler og metaforer, således at elementer får en betydning alene ved de symboler de er repræsenteret med.
4. **Visuelt virkemiddel/Affordans (affordance):** Systemet skal designes så brugeren intuitivt ved hvad de forskellige elementer i programmet gør.
5. **Navigation:** Brugeren skal kunne få hjælp til at navigere systemet.

10.2. Design

6. **Kontrol:** Det skal være klart for brugeren hvilke elementer denne kan bruge til at styre systemet, og også hvilken effekt brugerens handlinger har på problemområdet.
7. **Feedback:** Brugergrænsefladen skal hurtigt opdatere informationen fra systemet for brugeren, så de ved hvilken effekt deres handlinger har haft.
8. **Gendannelse:** Brugeren skal kunne 'fortryde' en handling. Desuden, hvis der er sket en fejl, så må det hele ikke gå i stå, men systemet kan hurtigt tages i brug igen.
9. **Begrænsning:** Handlinger som brugeren ikke bør udføre, der eksempelvis kan have stor effekt, bør forhindres ved at få brugeren til at bekræfte deres handlinger eller begrænse muligheden for at udføre dem.
10. **Fleksibilitet:** Systemet skal give brugeren mulighed for at foretage den samme handling på flere forskellige måder.
11. **Stil:** Ganske enkelt bør systemet være pænt og attraktivt.
12. **Høflighed (Conviviality):** Programmet bør være designet så det er høfligt og behageligt at anvende.

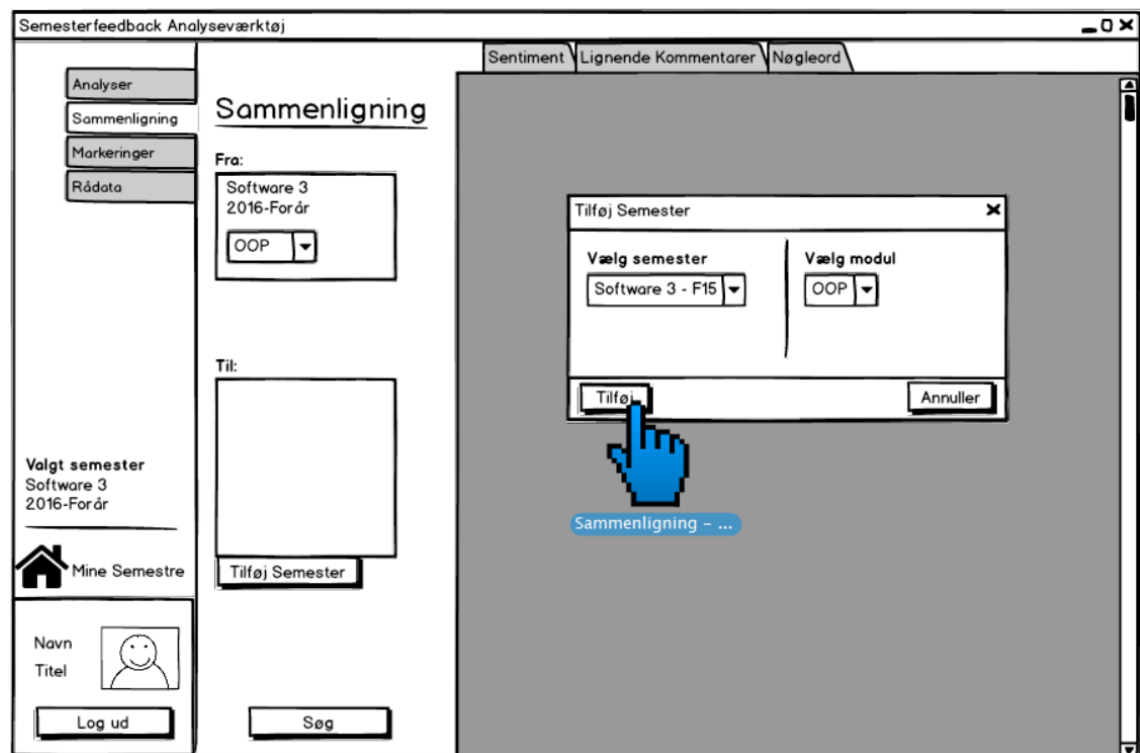
Systemet er *synliggjort* ved brug af en typisk brugerflade, det er gjort *konsekvent*, ved at de samme designmønstre anvendes ved faner, knapper, checkbokse og mere. For at gøre systemet *genkendeligt*, anvendes der fagbegreber som brugergruppen, semesterkoordinatorer, er bekendte med. Programmet er forsøgt gjort *intuitivt*, ved at anvende brugergrænsefladeelementer, som brugeren tidligere har anvendt og har kendskab til, hvilket gør at brugerens forventninger til brugergrænsefladefunktionaliteter opnås.

Der er også brugt tid på at udvikle et *stilfuldt* og *høfligt* program, som ikke skaber unødvendig støj. Dette gøres ved at gøre ting mere simple og anvende sigende tekst og ikoner.

Efter at have brugt designprincipperne, er det muligt at undersøge mere omkring brugergrænsefladedesignet ved brug af WIMP.

10.2.2 WIMP

Til brugergrænsefladen anvendes en tilgang der kaldes WIMP. WIMP står for: Windows, Icons, Menus og Pointer. De forskellige elementers funktion er beskrevet herunder. WIMP anvendes til at bygge en grænseflade, der anvender direkte manipulation, hvilket betyder at der bruges input til at påvirke programmet direkte, i stedet for blot at taste kommandoer i en terminal. På Figur 10.1 fremgår et udklip fra prototypen, som bruges til at illustrere hvordan WIMP kommer til udtryk i designet.



Figur 10.1: Udklip af prototypen, der viser hvordan WIMP er brugt

Windows

Vinduer, er afgrænsede områder på skærmen, med visuelle elementer, der tilsammen danner kanaler gennem hvilke programmer tager input. Et vindue viser det tilhørende programs output, og er en god måde at få et overblik på. Vinduer kan desuden dække over andre vinduer, så flere programmer kan være åbne på én gang, med det forreste værende det, der bliver brugt i øjeblikket.

Et eksempel på dette fremgår af Figur 10.1, hvor et vindue afgrænser funktionen af at tilføje et semester i sin sammenligning med det nuværende semester. Her bliver valget af semester og tilhørende modul afgrænset til et vindue. Ydermere er det overordnede program også et vindue, der afgrænser alt funktionaliteten af analyse- og overbliksværktøjet. Her fremgår det at 'Tilføj Semester'-fanen er forrest, eftersom det er den der i øjeblikket er i brug.

Icons

Ikoner, er små billeder, der symboliserer handlinger. Ikoner skal generelt være let genkendelige, og ikoner der bærer lighed med ikoner for lignende funktioner i andre programmer kan med fordel anvendes. Eksempelvis, er *gem* funktionen i mange

10.2. Design

kontorprogrammer symboliseret med en diskette, på trods af at det er et forlængst forældet digitalt medie. Hortons Checkliste, er en liste over punkter, der kan tjekkes ved ikon design, for at sikre sig, at de virker efter intentionen. Her er de udvalgte punkter, som har relevans for produktet:

- **Forståelig** Forstår brugeren, straks meningen med det?
- **Utvetydig** Er der andre tegn (etiketter eller andre ikoner) tilgængelige, som gør ikonet entydigt i sin kontekst?
- **Mindeværdigt** Viser ikonet objekter i handlinger? Bliver handlinger vist, som operationer på konkrete objekter?
- **Informativ** Hvorfor er dette koncept vigtigt?
- **Attraktiv** Har ikonerne pæne, glatte kanter og linjer?

I prototypen er der anvendt ikoner til simple funktioner, som at gå til hovedsiden (hjem), at minimere, ændre størrelse af vinduet og lukke programmet. Alle ikoner opfylder de ovennævnte udvalgte overvejelser, som der skal foretages for at et ikon er godt for programmets design og funktionalitet. Dette fremgår blandt andet på Figur 10.1, hvor huset kan ses, der tillader brugeren at komme til hovedsiden.

Menus

Menuer, er grupperinger af kommandoer, som regel efter kategori. En kategori kunne eksempelvis være *Filer*, som kunne indeholde gem, åben og eksporter. Disse kommandoer vælges så, typisk, ved hjælp af markøren, og resulterer så i en eller anden form for respons fra programmet.

I prototypen, er strukturen opstillet så hovedvinduet indeholder en antal underliggende menuer. Hvert af disse menuer, kan også indeholde et yderligere antal menuer, som hver repræsenterer et antal specifikke funktionaliteter. Dette ses på Figur 10.1, hvor den overordnede menu består af 'Analyser', 'Sammenligning', 'Markeringer' og 'Rådata'. Hver af disse menupunkter kan opdeles i endnu en række menupunkter. I udklipet fremgår det at den valgte hovedmenu er 'Sammenligning', hvor der efter at have lavet en sammenligning, er mulighed for at se sammenligningen i forhold til henholdsvis 'Sentiment', 'Lignende Kommentarer' og 'Nøgleord'.

Pointer

Markør, er et pegeredskab der anvendes til at udvælge elementet der ønskes interaktion med. Ved en PC er det typisk en mus eller touchpad, men det kunne også være en touchskærm, som på smartphones og tablets.

Da prototypen udvikles i Balsamiq, visualiseres en stor musemarkør tydeligt, så brugeren kan se at det er måden hvorpå brugeren kan interagere med programmet.

10.2. Design

Dette ses på Figur 10.1. Hvis der i Balsamiq trykkes med pointeren, vil de interaktive felter blinke rød, for nemmere at vise brugeren hvad det er muligt at trykke på.

10.2.3 Præsentation af prototypen

I det ovenstående delafsnit blev en del af prototypen præsenteret, se Figur 10.1, hvorpå sammenligning vises. I dette delafsnit vil der blive præsenteret yderligere to dele af prototypen, samt en beskrivelse af hvordan WIMP bliver benyttet i disse dele. Dog er W og P i WIMP, varierer ikke fra den allerede introduceret del af prototypen, som vist på Figur 10.1.

Den første side, der bliver præsenteret i dette delafsnit er vist på Figur 10.2, som viser den del af prototypen, hvor brugeres skal indtaste oplysninger til at oprette et nyt semester.

Semesterfeedback Analyseværktøj

Opret Semester

Studienavn
Vælg/Skriv semesternavn
Datalogi
Software

Semesternummer
4

År
2018
☒ Forår
☐ Efterår

Endeligt navn på semester
Software 4
2018-Forår

Vælg spørgeskema data
C:\Documents And Users\someone\d...
Gennemse

Opret

Mine Semestre

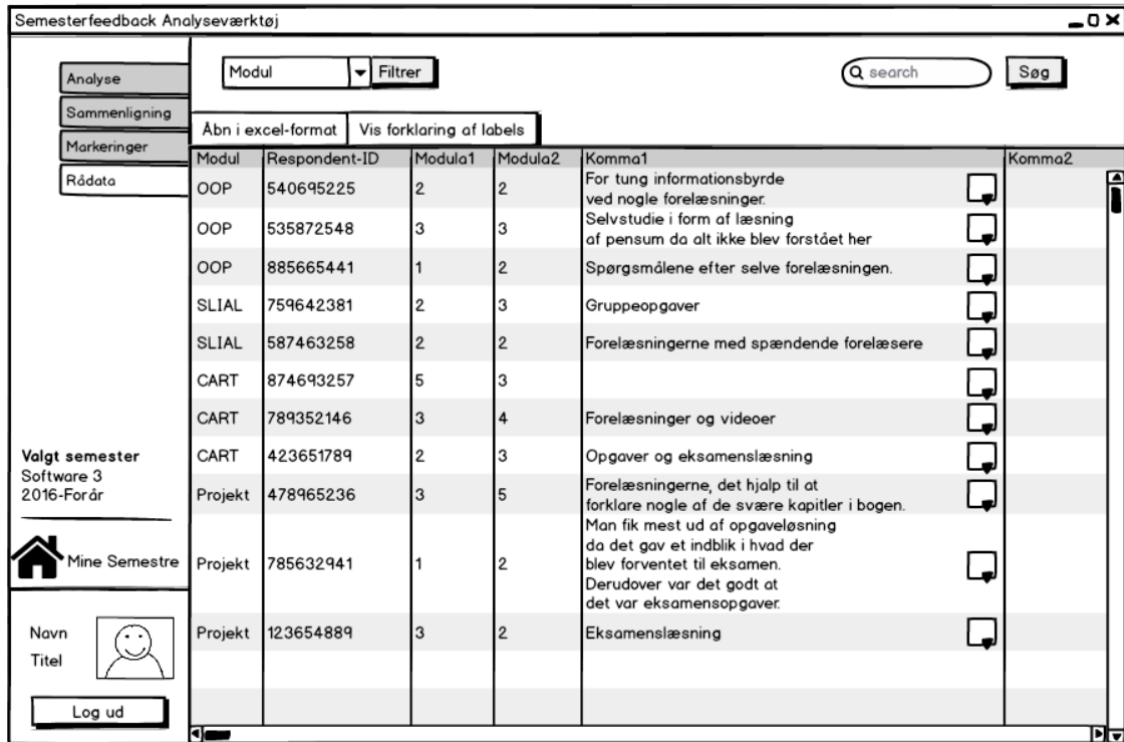
Navn
Titel
Log ud

Figur 10.2: Opret semester siden i prototypen

Der bliver i Figur 10.2 præsenteret et nyt ikon, i forhold til den tidligere, i form af en åben mappe, som skulle indikere for brugeren, at der skal findes og indsættes en fil. I denne del af prototypen, er der ikke yderligere menupunkter for brugeren at vælge i mellem.

10.3. Prototypetest og -evaluering

Den sidste del af prototypen, der bliver præsenteret, er siden hvorpå brugeren bliver præsenteret for rådataet. Dette kan ses på Figur 10.3. Denne side giver brugeren mulighed for at filtrere i dataet, samt at foretage en markering på en bestemt kommentar.



Modul	Respondent-ID	Modula1	Modula2	Komma1	Komma2
OOP	540695225	2	2	For tung informationsbyrde ved nogle forelæsninger.	
OOP	535872548	3	3	Selvstudie i form af læsning af pensum da alt ikke blev forstået her	
OOP	885665441	1	2	Spørgsmålene efter selve forelæsningen.	
SLIAL	759642381	2	3	Gruppeopgaver	
SLIAL	587463258	2	2	Forelæsningerne med spændende forelæsere	
CART	874693257	5	3		
CART	789352146	3	4	Forelæsninger og videoer	
CART	423651789	2	3	Opgaver og eksamenslæsning	
Projekt	478965236	3	5	Forelæsningerne, det hjalp til at forklare nogle af de svære kapitler i bogen. Man fik mest ud af opgaveløsning da det gav et indblik i hvad der blev forventet til eksamen. Derudover var det godt at det var eksamensopgaver.	
Projekt	785632941	1	2		
Projekt	123654889	3	2	Eksamenslæsning	

Figur 10.3: Visning af rådata i prototypen

I den del der bliver vist i Figur 10.3, ses der ingen nye ikoner, blot samme menupunkter, som også var i vist i Figur 10.1.

På baggrund af prototypens anvendelse af WIMP, samt de 12 designprincipper, kan prototypen testes og evalueres.

10.3 Prototypetest og -evaluering

Dette afsnit omhandler teori omkring fire forskellige testtyper, hvorpå en brugergrænseflade kan testes og evalueres. Disse test, deres formål og udførelse vil blive beskrevet, samt de positive og negative aspekter. Endeligt bliver en test udvalgt, samt en udførsel af denne test. Formålet ved at teste og evaluere prototypen, er at sikre at brugergrænsefladen er at undersøge, om der er nogle forvirrende elementer, samt testes der også for, hvor intuitivt prototypen er at benytte.

10.3.1 Teori

I dette delafsnit vil fire former for usabilitytests blive beskrevet: Formative test, summative test, verificerende test og sammenlignende test. Dette delafsnit bygger primært på Kapitel tre i bogen "Handbook of Usability Testing" [14].

Formative test

Denne test udføres tidligt i designprocessen. På dette stadie er målgruppen af produktet defineret, samt hvilke brugsmønstre, der skal være. Testen kan udføres på en prototype, et mockup eller sågar blot noget papir med nogle scener. Formålet ved at teste formativt, er at undersøge hvorvidt brugergrænsefladen for testpersonerne er intuitiv at benytte.

Denne test udføres, ved at en testperson og en testleder sammen gennemgår grænsefladen, hvor testlederen sørger for, at testpersonen har været rundt om det hele, uden at bede testpersonen, specifikt om at gøre noget bestemt. Det er også testlederens opgave at sørge for testpersonen tænker højt. I denne test er testpersonens tanker omkring brugergrænsefladen vigtig, da dette er testens primære fokus. Det er muligt for testlederen at spørge testpersonen, om vedkommende synes, der er nogle forvirrende elementer, eller om personen har forslag til forbedringer.

Styrken ved denne testtype er, at testlederen har mulighed for at få testpersonen til at uddybe sine tanker hvis det behov skulle opstå. Gennemgangen med testpersonen og testlederen resulterer i diskussion og undersøgelse af koncepter, som der endeligt er en hjælp til hvordan det endelige design skal være. Der vil være meget interaktion mellem testpersonen og testlederen i denne testtype. Dette gør, at den feedback testpersonen giver, kan være præget af testlederen, og dermed ikke kun testpersonens umiddelbare tanker.

Summative test

Denne testtype er en blanding af den formative test og den verificerende test. Denne test udføres typisk, når det fundamentale design er blevet etableret. Formålet med denne testtype er at udvide de opdagelser, som var resultatet af den formative test.

Denne test udføres ved at undersøge, hvor godt en testperson kan udføre nogle realistiske opgaver, samt at identificere specifikke mangler ved implementationen. Fordi denne test er en blanding af den formative test, og den verificerende test, da har den det informative aspekt fra den formative, og har kontrollerende målinger fra den verificerende test. Derfor vil udførelsen af denne testtype have en række opgaver, som testpersonen skal udføre. Det er igennem udførelse af disse opgaver, der indsamles kvantitativ data. Der vil stadig være en testleder tilstede, men i denne test vil testlederen have langt mindre interaktion med testpersonen.

Testen tillader både at få kvalitative og kvantitative data, og giver derfor et bredt billede af hvor effektivt designet er, samt hvilke problemer der kunne være med det.

10.3. Prototypetest og -evaluering

Dog fordi testen er en blanding af to, vil den hverken kunne gå i dybden med den formative del af testen, eller den verificerende.

Verificerende test

Denne testning udføres tæt på at produktet er færdigudviklet, og snart skal udgives. Formålet med denne test er at verificere, at de problemer fundet fra tidligere testning er udbedret, samt der ikke er nye problemer, der skal tages hånd om.

Før denne test udføres, skal der sættes en standard, dette kunne eksempelvis være: Testpersonerne skal kunne komme hen til en bestemt side, på et bestemt antal klik, eller afprøve om de problemer fra tidligere tests er ordnet. Det skal klarlægges inden testen, hvilken handling, der skal foretages, hvis disse standarder ikke bliver opfyldt. Det er disse standarder testen skal efterprøve. Testen foregår ved, at testpersonerne udfører en række opgaver, som udføres med enten meget lidt eller ingen interaktion med testlederen. Testpersonerne bliver bedt om ikke at tænke højt. Her er indsamlingen af kvantitative data i fokus, også selvom, der er opdaget en mulighed for, at standarden ikke bliver overholdt.

En styrke ved denne testtype er, at opgaverne resulterer i kvantitativ data, som beskriver, hvor godt de opsatte standarder bliver opfyldt. Men hvis resultatet af testen viser, at der stadig er problemer, eller at standarderne ikke bliver overholdt, da siden denne test udføres så tæt på at produktet er færdigt, må der afgøres, hvorvidt der er tid til yderligere test, eller om deadlineen for produktets udgivelse må og kan udskydes.

Sammenlignende test

Denne test kan udføres når som helst i designprocessen, og kan foretages i en kombination med en af de tre andre test. Formålet med testen er at sammenligne prototypen eller produktet med en eller flere ligende produkter, samt at forstå fordelene og ulemperne ved de forskellige designvalg.

Udførelsen af testen varierer lidt i forhold til hvor i udviklingsprocessen produktet testes. I et tidligt stadie kan testen udføres via en formativ test på flere ekstremt forskellige brugergrænseflader for at undersøge, hvilken der rammer målgruppen bedst.

Er produktet længere i udviklingen, da kan testen benyttes til at undersøge hvor effektivt et enkelt element er, eksempelvis om hvilken slags knap brugerne fortrækker. Endeligt hvis testen foretages i slutningen af produktudviklingen, da kan det endelige produkt sammenlignes med en konkurrents produkt.

Ved at sammenligne produktet, eller blot nogle elementer med andre lignende produkter, er det muligt at undersøge hvilke elementer, der virker bedst hvornår. Denne

10.3. Prototypetest og -evaluering

testning vil resultere i at en kombination af de sammenlignede produkter vil være det foretrukne, med de bedste dele af hver produkt. Der skal også overvejes, hvor forskellige eller ens de produkter, der skal sammenlignes skal være. Skal der sammenlignes med en hel side, eller blot hvilken type knap, der er den foretrukne.

10.3.2 Test/evaluering

I dette delafsnit vil prototypen blive testet af en række testpersoner. Både testen og testpersonerne vil blive udvalgt, samt med begrundelse for valget. Endeligt vil forløbet af testen blive gennemgået.

Valg af test og testpersoner

I dette underafsnit, vil der udvælges en af de gennemgåede testtyper til testning, samt begrundelse for valget deraf. Ydermere vil personerne, der skal agere testpersoner, udvælges, samt en begrundelse for valget af disse.

Der vælges i første omgang at teste formativt, da der ikke er foretaget andre test af prototypen, så det er interessant at undersøge hvilke tanker prototypen frembringer, samt hvilke problemer testpersonerne synes der er. Der vil blive foretaget yderligere test af brugergrænsefladen senere i udviklingen, se Kapitel 18.

Informanterne blev udvalgt til at være testpersoner, da de er produktets målgruppe, og er derfor potentielle brugere af produktet. De har en kendskab til deres arbejde nu, og vil kunne komme med feedback omkring, om brugergrænsefladen vil besværliggøre eller er nemmere end deres nuværende, og om de kan se sig selv benytte produkt ud fra prototypen. Ydermere kan det undersøges, hvor intuitiv grænsefladen er at benytte.

Udførsel af test

I dette underafsnit vil udførelsen af den udvalgte test blive beskrevet.

Planen var, at testpersonerne skulle være de tre informanter, som tidligere blev interviewet, men to af informanterne måtte melde afbud i sidste øjeblik, og derfor, grundet tidspress, måtte der nøjes med den ene informant.

Da testen er formativ, kommer testen til at foregå således at testpersonen, sammen med en testleder, udforsker prototypen. Testlederens opgave er at få testpersonen til at komme omkring hele prototypen, samt at sørge for at testpersonen tænker højt, eksempelvis om vedkommende synes noget er forvirrende, eller om den knap der trykkes på, gør det der forventes. For at dokumentere testresultaterne bliver skærmen hvor prototypen bliver udforsket på, filmet, samt testpersonens og testlederens samtale optages. Der vil være en observatør, der sikrer at testen foregår som den skal. Både testlederen og observatøren er personer fra projektgruppen.

10.3.3 Fundne problemer

I dette delafsnit vil resultaterne fra den formative testning blive præsenteret, samt hvilke ændringer i prototypen, som testen har medført.

Nedenfor vises en række problemer, samt løsninger på disse, opstillet i Tabel 10.1. Ydermere, vil løsningerne blive vist i form af figurer, der viser et opdateret prototypedesign.

1. Oprettelse af Hold	
Problem	Der var intet ønske fra testpersonen om at overtage den administrative del, som består i at selv oprette semester, med dertilhørende data.
Løsning	Dette var et valg der blev foretaget af gruppen, da der grundet tidspres ikke kan opsættes en administrativ del også, selvom det ville være det optimale.
2. Navngivning	
Problem	Testpersonen blev forvirret over betydningen på nogle knapper og etiketter
Løsning	Bedre navngivning, der giver et bedre billede af hvad knappen eller etiketten gør.
3. Tilhørsforhold	
Problem	Testpersonen var forvirret over hvilke knapper der hørte sammen.
Løsning	For at illustrere tydeligt tilhørsforholdet i mellem knapper, da indrammes de i deres egen boks. Desuden omrokeres nogle elementer, for at tydeliggøre hvad der er information, og hvad brugeren skal give input til.
4. Etiket-knap	
Problem	Det var ikke intuitivt for testpersonen at trykke på etiket-knappen, som indeholdte beskrivelse til de forvirrende etiketter.
Løsning	Efter testpersonens ønske laves, i stedet for en knap, en tooltip, som dukker op hvis musen hviler på etiketten.
5. Markering	
Problem	Testpersonen fandt ved første øjekast markeringen forvirrende, og havde brug for at blive forklaret hvordan denne fungerer. Dog efter forklaringen var indstillingen til markeringsfunktionen positiv.
Løsning	Efter testpersonens ønske vil der blive beskrevet i brugervejledningen hvordan markeringen fungerer.

Table 10.1: Problemer fundet ved prototypetesten

På Figur 10.4 og 10.5 illustreres ændringen der blev foretaget i prototypen, for at løse Problem 2. Det ses på 10.4 og 10.5 et udklip af den gamle og den nye prototype, for at illustrere hvilken ændring der er foretaget for at løse opgave 2. Som ses på Figur 10.5, da er 'Til' ændret til 'Sammenlign', 'Fra' er nu 'Med' og endeligt er 'Søg'

10.3. Prototypetest og -evaluering

blevet til 'Sammenlign'.

Sammenligning

Fra:

Software 3
2016-Forår

OOP ▼

Til:

Tilføj Semester

Søg

Figur 10.4: Gammel sammenligningsside

Sammenligning

Sammenlign:

Software 3
2016-Forår

OOP ▼

Med:

Tilføj Semester

Sammenlign

Figur 10.5: Ny sammenligningsside

På Figur 10.6 og 10.7 ses 'opret semester'-siden både i den opdaterede prototype, og den gamle. Figuren viser at boksen med 'Endeligt navn på semester' flyttet hen til 'Opret'-knappen. Samtidig er alt det brugeren skal give input til, samlet.

Opret Semester

Studienavn
 Vælg/Skriv semester navn ▾
 Datalogi
 Software

Semesternummer
 4 ▾

År
 2018 ▾ ☒ Forår ☐ Efterår

Endeligt navn på semester
 Software 4
 2018-Forår

Vælg spørgeskema data
 C:\Documents And Users\someone\dr Gennemse

Opret

Figur 10.6: Gammel Opret-semester-side

Opret Semester

Studienavn
 Vælg/Skriv semester navn ▾
 Datalogi
 Software

Semesternummer
 4 ▾

År
 2018 ▾ ☒ Forår ☐ Efterår

Vælg spørgeskema data
 C:\Documents And Users\someone\dr Gennemse

Endeligt navn på semester
 Software 4
 2018-Forår

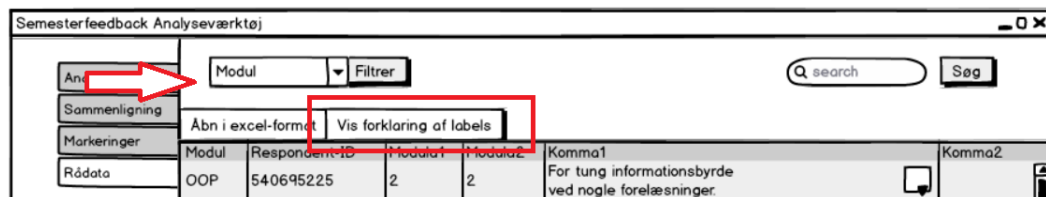
Opret

Figur 10.7: Ny opret-semester-side

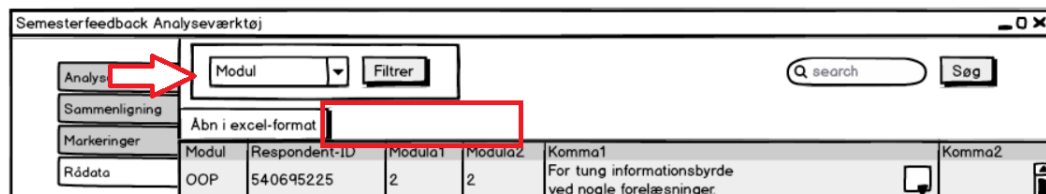
Fjernelse af etiketknappen, som bliver beskrevet i Problem 4, kan ses på Figur 10.8 og 10.9, hvor der ses et udklip af både den gamle og den nye rådataside i prototypen. Dog indeholder programmet Balsamiq, som prototypen er lavet i, ikke en tooltip. Desuden er de forvirrende navne som: 'komma1' ikke ændret, da det kan variere hvad spørgsmålene er, så de repræsenterer blot et spørgsmål. Endeligt ses der også at dropdown-menuen 'Modul' og knappen 'filtrer' er blevet indrammet, for at illustrere at de hænger sammen, som en løsning på Problem 3.

Endeligt kan det konkluderes, at selv med de ovenstående ændringer, vil brugerne ved første anvendelse af programmet, udover at læse brugervejledningen, selv skulle gå på opdagelse, og afkode hvilke muligheder produktet giver. Det er ud fra den opdaterede prototype, produktets brugergrænseflade bliver implementeret.

10.4. Opsummering



Figur 10.8: Gammel rådataside



Figur 10.9: Ny rådataside

10.4 Opsummering

I dette kapitel blev der redegjort for værktøjer til at lave et brugergrænsefladedesign. Disse blev udtrykt som WIMP og designprincipper omkring ikoner, samt prototype-teori, og test heraf. Det bidrager alt sammen til at kunne opbygge en funktionel brugergrænseflade. Derudover kendes der også forbedringer fra det originale design på grund af de test som blev foretaget med prototypen.

Med udgangspunkt i dette kapitel bliver det muligt at fortsætte med en reel, fuldt funktionel implementering af det endelige produkt.

Brugergrænsefladen er en del af løsningen, og systemgrundlaget skal også være på plads, for at brugergrænsefladen kan have den ønskede funktionelitet.

I næste del af rapporten arbejdes der med implementering, først af programmets funktionalitet, som leder hen til Kapitel 15, hvor der arbejdes med implementeringen af brugergrænsefladen.

Del III

Implementering

Kapitel 11

Data

I de kommende afsnit vil datadelen af implementeringen blive undersøgt. Det skal anvendes til både træning og validering af de to sentimentanalyser. Den grafiske brugergrænseflade skal vise rådata og sentimentanalyser. Den skal også tillade sammenligninger mellem evalueringer fra forskellige semestre.

Data er et centralt element i udarbejdelsen af programmet. Systemet bygges op omkring evnen til at vise informationer angående semesterevalueringer. For at kunne gemme den tilgængelige data og de udarbejdede analyser, oprettes og benyttes en database. Dette gør, at de forskellige informationer og strukturer som brugeren indtaster, er koblet op til et login i databasen, og ikke låst til den enkeltes computer.

11.1 Dataindsamling

Data er nødvendigt for både at træne og evaluere sentimentklassificeringsmodellerne. Det data som anvendes, bør være af høj kvalitet og kvantitet. Kvantiteten af data sikrer, at programmet lærer generelle mønstre, i forhold til at huske resultaterne.

Nødvendigheden for både kvantitet og kvalitet af dataet skaber en udfordring, og netop derfor bør dette problem blive undersøgt som en del af projektet.

11.1.1 Evalueringer

Semesterevalueringer er domænespecifikke, hvilket øger datakvaliteten betydeligt. Derimod er kvantiteten af semesterevalueringer, som vi har haft adgang til, ikke optimal. Under udviklingen af projektet har vi haft adgang til cirka 2200 danske evalueringskommentarer.

Forbehandlingen af datasættet krævede både manuel afmærkning, om det havde positiv eller negativ sentiment, samt fjernelsen af alle personfølsomme informationer. Eksempler på semesterevalueringer kan ses herunder.

"At jeg føler jeg kan bruge kursuset til noget og hvor meget jeg forstår undervejs."

11.1. Dataindsamling

"sværhedsgraden af læsestoffet"

"forberedelse og opgave regning"

"Materialet har været spændende og blevet formidlet på en god måde."

"Undervisers engagement og indsats for at gøre kurset overskueligt, forståeligt og nemt at følge".

"Det har ikke fanget min interesse."

Dette datasæt er velegnet til at evaluere det endelige projekts præcision, siden det er den samme type data, som programmet eventuelt skal kunne behandle og vurdere.

11.1.2 Scraping

Et program blev udviklet i løbet af projektet, til datascraping på Trustpilot. Herpå giver folk feedback, og deler deres oplevelser om specifikke virksomheder og hjemmesider. Når de gør dette, vurderer de også deres oplevelse med 1 til 5 stjerner.

Det er derfor let at finde tekstfeedback som er meget positivt (5 Stjerner) og meget negativt (1 Stjerne). I det udviklede scraping-program kan der efterspørges anmeldelser med en specifik mængde af stjerner, som gør dataindsamlingen mere effektiv.

Når disse data er indsamlet, samles de i 2 lister af feedback: Meget positiv, og meget negativ. Hvilken service der er anmeldt er irrelevant for projektets formål, men det vil derfor være uden for dette semesterevalueringsdomæne.

Dataindsamlingen fra trustpilot resulterede i 27.077 anmeldelser. Data fra trustpilot blev derefter opdelt i 2 dele. Den første del er et mindre udsnit af cirka 2.000 anmeldelser, som blev manuelt vurderet, om de var positive eller negative. Dette skyldes, at selvom en anmeldelse giver en bestemt mængde af stjerner, kan indeholdet af kommentaren stadig have et andet sentiment. Mængden af stjerner, som anmeldelsen havde, bestemte afmærkningen af det resterende datasæt.

Eksempler på den indsamlede data kan ses nedenunder:

"aflysning af tog fra københavn til aarhus uden forklaring næste tog to timer senere"

"elendigt simpelthen rædselsfuldt"

"kvalificeret og imødekommende personlig service"

"rædselsfuldt"

"super service og flot arbejde kan kun anbefales"

"let at komme i kontakt med folk hvis man søger nye venner eller bare vil have nogle at snakke med"

Som det kan ses, overlapper anmeldelser ikke med semesterevalueringsdomænet, og er dermed af lavere kvalitet. Det kan derimod argumenteres, at kvantiteten af dataet, samt evalueringssentimentet, når en person taler negativt eller positivt om

11.2. Forbehandling

noget, kan bruges til at finde generelle mønstre, og dermed er acceptabel til at træne modellerne med.

11.2 Forbehandling

Dette afsnit vil omhandle det indledende arbejde til analysemodellerne, hvor der skal anvendes data. Dette kan ske både ved at rengøre sætningen for tegnsætning, store bogstaver, men også stopord. Et stopord, er et ord der ikke bringer nogen yderligere betydning til sætningen. Eksempler på stopord kunne være: 'En', 'et', 'der', 'det' og 'være'.

11.2.1 Forberedende forbehandling

Der bliver foretaget to ændringer af rådataet, som begge har med henblik at gøre dataet mere ensartet: Alle specialtegn fjernes og alle ordene ændres til at være skrevet med små bogstaver.

Formålet ved at fjerne specialtegnene, er at de ikke er en del af selve ordet. De specialtegn der fjernes kunne eksempelvis være: citationstegn: ", udråbstegn: !, spørgsmålstegn: ?, punktum: . eller komma: ,.

Specialtegnene fjernes ved brug af et regulært udtryk [15], der fjerner alt som ikke er bogstaver fra a til å eller mellemrum.

Alle bogstaver ændres til små bogstaver, eftersom selve ordene ikke mister deres betydning, hvis de er skrevet med stort eller småt forbogstav.

11.2.2 Stopord

Efter dataet er ændret til kun at være med små bogstaver og uden specialtegn, da kan stopordene fjernes. Formålet ved at fjerne stopord er at kun ende med ord der har en reel betydning for kommentarens sentiment. Stopordene, som benyttes i forbindelse med systemet, kan ses i Bilag E.

For at opbevare dataet i produktet, designes, undersøges og behandles databaselaget.

11.3 Database

Dette afsnit vil beskrive teori, opsætning og brug af databaser i forhold til systemet.

11.3.1 Databasenormalisering

Formålet med at normalisere databaser er at formindske mængden af redundant data. Dette vil gøre, at der kan foretages ændringer i databasen lettere, da den

11.3. Database

samme data ikke gemmes flere steder. [16]

Redundans kan påvirke databasen på to måder, hvilket forsager at handlinger i databasen er sværere eller langsommere. Det kan påvirke ydeevnen af produktet ved at bruge unødvendige ressourcer. Det kan også påvirke vedligeholdelsen fordi den samme opdatering af data skal forekomme flere steder i databasen. [16]

Normalisering forekommer typisk af 3 former, men kan udvides til 5 former i særtilfælde. Hver af de 3 former bidrager med en ny abstraktion til normaliseringen. Et højere nr. normalform indrager også de tidligere normalformer. [16]

1. *normalform* har ingen kolonner, som gentager en anden kolonnes værdi. Desuden anvendes der udelukkende atomare værdier i tabellernes felter. [16]

2. *normalform* kræver at der ved brug af sammensatte nøgler, ikke eksisterer felter udover nøglerne, som ikke er afhængige af nøglerne. Dette betyder, at alle ikke primære attributter er fuldt afhængige af primær-nøglen. [16]

3. *normalform* har ingen afhængigheder mellem felter, som ikke indgår i primærnøglen. [16]

Efter at have omdannet database tabellerne til 3. normalform kan et E/R-diagram dannes.

Figur 11.1 viser et E/R-diagram over den normaliserede database. Her ses de enheder, som eksisterer i databasen, deres handlinger og deres attributter. Der er illustreret hvilke forhold de har til hinanden ved brug af linjer, som beskriver forholdet mellem enhederne. Hver enhed i diagrammet kan forstås som en tabel i databasen.

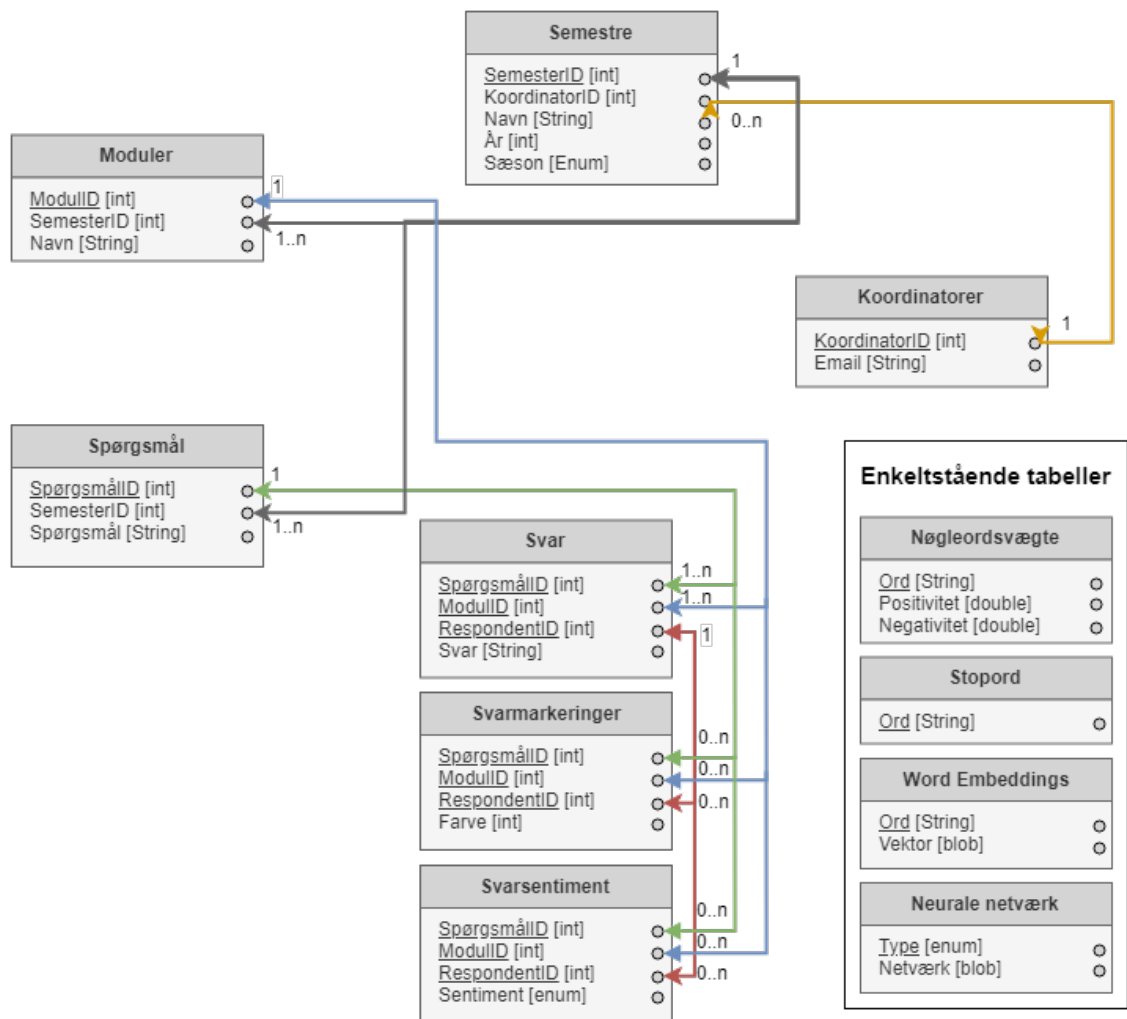
Efter at have normaliseret databasen og derved optimeret den, kan den opsættes og klargøres til brug af systemet.

11.3.2 Opsætning

For at kunne oprette og tilgå en database fra eksterne enheder, anvendes der en vært, som databasen kan oprette forbindelse til. Til dette er der brugt Unoeuro, som tillader at oprette databasesystemer.

Databasesystemet er visualiseret ved brug af phpMyAdmin som brugergrænseflade og databaserne kan behandles med MySQL, som er software der behandler data ved brug af programmeringssproget SQL.

11.4. Afrunding



Figur 11.1: ER-diagram over tabellerne i databasen

11.3.3 Databaseforbindelse

For at kunne forbinde til at den online database er der i programmets biblioteker inkluderet MySQL Connector/J 8.0.13 driver. Med denne driver, kan der laves SQL kald til databasen fra javaprogrammet.

11.4 Afrunding

Med dataindlæsning, database og preprocessing af data på plads kan dataene bearbejdes. Det vil sige at der både kan foretages analyse af dataene, hvilket vises i Kapitel 12 og 13, og at den data kan præsenteres i brugergrænsefladen, hvilket vises

11.4. Afrunding

i Kapitel 15.

Kapitel 12

Statistisk tilgang - Sentimentklassificering

En måde at analysere projektets data på, er at klassificere kommentarernes sentiment. En tilgang til dette kunne være statistisk at undersøge hvilken betydning enkelte ord har for et sentiment. Ved at beregne statistisk hvilken betydning hvert ord har for sentimentet af en sætning, kan disse resultater genanvendes på andre sætninger for at vurdere deres sentiment.

12.1 Nøgleordsanalyse

Endnu en måde at trække karakteristiske egenskaber ud af evalueringerne på, er ved at finde ord med en særlig betydning. Til dette bruges "Term Frequency-Inverse Document Frequency" (TF-IDF), som er en numerisk statistik, der kan bestemme hvor vigtigt et ord er for en kommentar i en given samling af kommentarer. Dette oversættes på dansk til "Termhyppighed-Invers dokumenthyppighed. Et 'dokument' defineres som et sammenhængende stykke tekst, som i denne kontekst dækker over kommentarer. Teorien bag kapitlet baseres på TF-IDF-siden på Wikipedia [17].

TF-IDF værdien er produktet af de to statistiske værdier: "Term Frequency" (TF) og "Inverse Document Frequency" (IDF). TF får den samlede værdi til at stige i forhold til antallet af gange et ord opstår i en kommentar, og IDF udligner dette i forhold til antallet af andre kommentarer der indeholder dette ord, hvilket hjælper med at kompensere for ord der generelt opstår mere, og derfor ikke har nogen speciel betydning.

12.2 Term Frequency

Termhyppigheden er givet ved $tf(t, k)$, hvor t er termen og k er kommentaren. Der findes forskellige varianter af hvordan den kan beregnes:

12.3. Inverse Document Frequency

$$\text{Binær}(0, 1) : \text{Hvis } k \text{ indeholder } t \text{ er den } 1, \text{ ellers } 0. \quad (12.1)$$

$$\text{Rå tælling}(f_{t,k}) : \text{Tæller antallet af gange } t \text{ findes i } k. \quad (12.2)$$

$$\text{Termfrekvens}\left(\frac{f_{t,k}}{|k|}\right) : \text{Justerer den rå tælling med længden af kommentaren.} \quad (12.3)$$

Disse tre varianter stiger i deskriptivitet i forhold til termens betydning. Binær tager udelukkende højde for om termen er tilstede i kommentaren, rå tælling tager højde for hvor mange gange termen er til stede i kommentaren og termfrekvens bruger rå frekvens, men justerer det i forhold til antallet af ord til kommentaren.

12.3 Inverse Document Frequency

Invers-dokumenthyppighed betegner hvor meget information et givent term giver. Dette måles ved at undersøge hvor hyppigt ordet er brugt i andre kommentarer, og dermed vurdere ordets specificitet i kommentaren.

Invers-dokumenthyppighed er givet ved $idf(t, K)$, hvor K er samlingen af kommentarer som termen t stilles op imod. Den er beregnet ved Formel 12.4.

$$idf(t, K) = \log \frac{|K|}{|\{k \in K : t \in k\}| + 1} \quad (12.4)$$

Hvor $|K|$ er antallet af kommentarer i samlingen K , og $|\{k \in K : t \in k\}|$ er antallet af kommentarer der indeholder termen t . Hvis ikke der er nogen kommentarer i K der indeholder t , sker der en division med 0, hvilket er grunden til at der tilføjes +1.

12.4 Term Frequency-Inverse Document Frequency

De to statistiske værdier TF og IDF kan bruges til at beregne den samlede $TF - IDF$, hvor en høj termhyppighed eksempelvis trækker den op, men en høj invers-dokumenthyppighed trækker den ned igen.

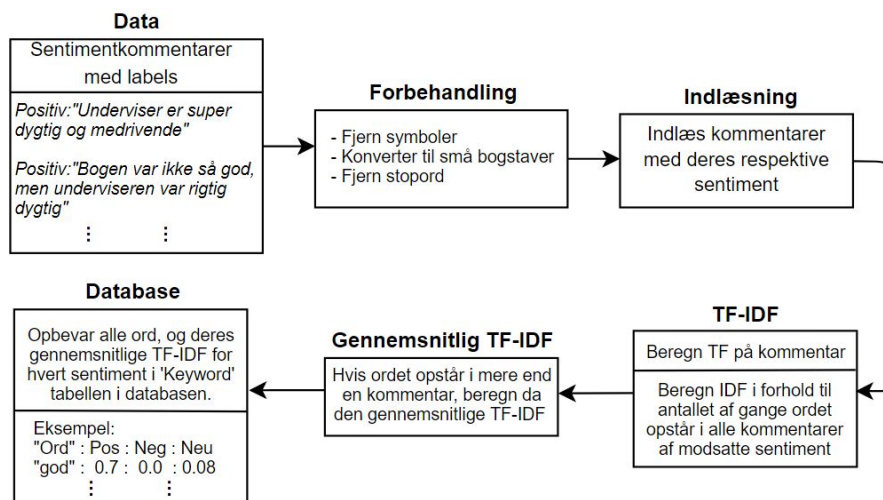
$$tfidf(t, k, K) = tf(t, k) * idf(t, K) \quad (12.5)$$

12.5 Nøgleord som sentimentklassifikator

I konteksten af semesterevalueringen kan den beskrevne måde at bestemme nøgleord bruges til at klassificere kommentarer i sentimenter. Med en række kommentarer der manuelt er klassificeret som positiv eller negativ, kan det bestemmes hvor vigtigt et ord er for et givent sentiment som helhed. Med disse vægte kan ukendte kommentarer deles op i ord, og med en undersøgelse af hvert ords TF-IDF vægt, kan kommentarens sentiment klassificeres.

For at beregne TF-IDF vægte, der er sigende for et ords betydning i forhold til et sentiment, skal det bestemmes hvordan de ønskes klassificeret. Blandt de mest brugte metoder til klassificering af sentiment med TF-IDF, er at lave featurevektorer til hver kommentar, som lærer en 'Support Vector Machine' (SVM), hvor i et n-dimensionelt rum, at disse featurevektorer fordeler, når de er positive eller negative [18][19]. Da en implementation af en SVM er omfattende, blev det bestemt at tage en mere naiv tilgang til klassifikationen.

Denne naive tilgang indebærer, at alle positive og negative kommentarer deles op i to samlinger, med deres respektive sentimenter. Med disse to samlinger, beregnes TF på én af dem, og IDF beregnes i forhold til samlingen af det andet sentiment. Dette gøres for alle ord i alle kommentarer i alle samlinger. Hvis det så skulle forekomme, at nogle termer opstår i mere end én kommentar i en samling, beregnes gennemsnittet af disses TF-IDF-vægte. Dette vil resultere i at nogle ord kan være brugt både i positive og negative kommentarer, og derfor har både en positiv og negativ TF-IDF vægt. Implementationen af hele denne proces er illustreret i Figur 12.1.



Figur 12.1: Diagram over behandling nøgleord igennem programmet

12.5. Nøgleord som sentimentklassifikator

Forløbet der bliver beskrevet i Figur 12.1 starter med kommentarer, der har en afmærkning med deres respektive sentiment. Disse kommentarer forbehandles for at fjerne diverse symboler, konvertere til små bogstaver, samt fjerne ord som "der, at og da" der ikke tilbyder nogen påvirkning for sentimentet. Denne forbehandler er også beskrevet i Afsnit 11.2. Efter forbehandling, læses alt dataet ind i programmet, for at kunne gå igennem alle kommentarer og beregne de enkelte kommentarers TF, og deres IDF i forhold til alle kommentarer af de modsatte sentiment. De enkelte termfrekvens- og invers dokumentfrekvens-beregninger er implementeret i klassen TFIDF som vist i Listing 12.1.

Listing 12.1: Overordnede metoder i TFIDF.java

```
1 public class TFIDF {
2     double termFrequency(String term, List<String> wordsInDocument)
3         Implementering af formel 11.3
4
5     double inverseDocumentFrequency(String term,
6         List<List<String>> documentsOppositePolarity)
7         Implementering af formel 11.4
8 }
```

Listing 12.1 har i sin mest grundlæggende form implementeret beregningen af termfrekvens og invers dokumentfrekvens for et enkelt term. For at kunne få mening ud af alle termer i alle dokumenter, skal der bruges en iterator, der kan læse alle de relevante datapunkter og bruge TFIDF-klassen til at beregne de enkelte termers betydning.

Listing 12.2: TFIDF_Iterator.java

```
1 Map<SentimentPolarity, List<String>> allDocuments =
2     makeSentimentCollections();
3
4 for(SentimentPolarity sentiment : SentimentPolarity.values()) {
5     Map<String, List<Double>> TFIDF_OfTerms = new HashMap<>();
6
7     for(String document : documentsCurrentPolarity) {
8         for(String term : terms) {
9             // Hvis nyt term, beregn idf og put i indeks 0
10            if(!TFIDF_OfTerms.containsKey(term)) {
11                TFIDF_OfTerms.put(term, new ArrayList<>());
12
13                idf=TFIDF.idf(term, documentsOppositePolarity);
14                TFIDF_OfTerms.get(term).add(idf);
15            }
16
17            tf=TFIDF.tf(term, terms);
18            idf=TFIDF_OfTerms.get(term).get(0);
19            tfidf = tf * idf;
```

12.6. Afrunding

```
19
20         TFIDF_OfTerms.get (term) .put (tfidf)
21     }
22 }
23
24     meanTFIDF_OfTerms.put (sentiment, calcMean (TFIDF_OfTerms))
25 }
```

De mest essentielle egenskaber som iteratoren er implementeret med, fremgår af Listing 12.2. På linje 1 går den gennem indlæsningen af alle dokumenter, sorteret efter dokumentets tilhørende sentiment. Vi er herfra interesserede i at beregne TFIDF for alle termer i de enkelte samlinger. På linje 3 går en foreach-løkke over alle sentiment, og på linje 4 bliver der lavet et Map fra alle unikke termer af dette sentiment til en liste over alle TFIDF-værdier der er beregnet for dette ene term. Herefter går den i linje 6 og 7 over alle termer i alle dokumenter i samlingen af dette sentiment. Første gang et term mødes i den nuværende samling tilføjes det til Map'et, og IDF-værdien beregnes for termen og bliver lagt i indeks 0. Denne værdi bør kun beregnes én gang, selv hvis termen findes i mange forskellige dokumenter, da IDF-værdien kun afhænger af dokumenterne i de modsatte sentiment. Herfra beregnes TFIDF for termen i det nuværende dokument, og det lægges ind i listen i Map'et. Efter TFIDF_Iteratoren er løbet igennem alle dokumenter for en enkelt samling, beregnes gennemsnittet af TFIDF for termer der er opstået mere end én gang på linje 24, og dette opbevares i et andet Map.

Map'et over de gennemsnitlige TFIDF-værdier over alle polariteter eksporteres til databasen, hvilket for hvert term beskriver dens betydning for begge sentiment. Når nye kommentarer skal klassificeres med et sentiment kan disse vægte bruges.

Denne analysemetode kan opnå nogle unikke resultater, men det garanteres ikke at metoden kan opnå de bedste mulige resultater. Derfor vil en anden metode, som anvender maskinlæring, introduceres til at bidrage til bestemmelse af sentiment. Det vil senere blive undersøgt hvorvidt det giver bedre resultater at lave en hybrid løsning, mellem disse to.

12.6 Afrunding

Den statistiske tilgang til sentimentklassificering i projektet udnytter nøgleordsanalyse i form af termhyppigheden og invers-dokumenthyppighed. Tilsammen giver dette en model, som kan klassificere sentiment af kommentarer, baseret på ordene i en specifik kommentar, relativt til ordets brug i andre kommentarer.

Denne sentimentklasificering kan bruges i samarbejde med en maskinlæringstilgang til klassificering af kommentarer.

Kapitel 13

Maskinlæringstilgang - Sentimentklassificering

I dette kapitel vil der introduceres en maskinlæringstilgang til sentimentklassificering. Et udvalg af neurale netværk vil blive anvendt til først at lære en algoritme at forstå ord, og dernæst anvende denne forståelse til at lære et andet netværk at forstå sætninger til sentimentklassifikation.

13.1 Neurale Netværk

Neurale netværk er en type optimeringsalgoritme som ønsker at mindske sine fejl i forhold til kendte data. Gennem dette kan et neuralt netværk trænes til at lave udregninger og beslutninger i forhold til mere abstrakte problemer. Neurale netværk kræver et større sæt af data for at kunne optimere sig selv. Disse datasæt skal indeholde inputs og deres tilhørende forventede outputs. Den unikke egenskab at algoritmen kan lære af sine fejl, er et resultat af et neuralt netværks struktur.

I dette projekt anvendes neurale netværk til sentimentklassificering af tekststrengene. Denne klassificering er baseret på 2 principper: "Forståelse af ord" og "Forståelsen af sætninger af ord". Disse principper leder os også til det mellemtrin, at for at kunne forstå sætninger af ord, bør vi først kunne forstå enkelte ord og deres betydning. Dette ønsker vi i projektet at løse ved hjælp af "Word Embeddings" og "Recurrent-Neural-Networks" (RNN), hvor word embeddings hjælper computeren med at forstå ord, og RNNs hjælper computeren med at forstå sætninger af ord.

13.1.1 Lag

Et lag i neurale netværk skal forstås som en række af elementer, som ikke kan kommunikere internt i det samme lag. Et neuralt netværk er opdelt i flere lag, af 3 forskellige typer [20]:

13.1. Neurale Netværk

- *Input Lag* indeholder inputdata for det neurale netværk. Værdien af elementer i dette lag er bestemt udenfor det neurale netværk.
- *Skjult Lag* indeholder noder som modtager signaler fra tidligere lag, behandler dem, og videregiver et nyt signal. Skjulte lag er den eneste type lag, der kan være mere end ét af i et netværk.
- *Output Lag* indeholder det endelige resultat af det neurale netværk, og videre-sender dermed ikke noget signal.

Dette betyder at input data vil blive behandlet gennem flere lag for at kunne udregne det endelige resultat.

13.1.2 Noder

Neurale netværk består af *noder*. Strukturen af en node varierer med hvilken type node det er. Generelle egenskaber for alle noder er at de modtager et sæt af ét eller flere tal, hvorudfra de vil udregne én ny værdi, som er dens output værdi. Dette beskrives ofte som en input gate og en output gate.

13.1.3 Kanter

Kanter forbinder noder med hinanden, mellem lag, ved at videresende én nodes output videre til en anden node. Noder vil modtage signaler fra flere andre noder i det tidligere lag, og videresender et nyt signal til alle noder i det næste lag. Dette betyder, at der er flere kanter der forbinder lag, end der er noder i et lag. En kant indeholder også en vægt som ganges på det signal den videresender. [20]

13.1.4 Long short-term memory netværk

De tidligst udviklede neurale netværk tager kun højde for nuværende input data, hvor tidligere data er irrelevante. Disse kaldes feedforward neurale netværk. Senere er der udviklet Recurring-Neural-Networks (RNN), som har en dynamisk adfærd, bestemt af tidligere input. Dette opnås af en *Recurring Neuron's* interne tilstand, ofte kaldet dens Hukommelse (Memory). I dette projekt bruges der Long short-term memory (LSTM), som er en type af RNN [21].

En Long short-term memory node består af input og output gates, som en normal node, men ud over dette, også en forget gate. Den nye gate bestemmer hvor meget der skal huskes fra tidligere iterationer. Dette tillader en LSTM node at huske værdier over længere tid. Dette er en essentiel egenskab for projektet, fordi der foretages en sentimentanalyse af tekststykker, hvor et ords betydning kan være meget afhængig af tidligere ord i sætningen.

13.2 Word Embedding

Ord er essentielt et menneskeligt princip, og i sin reneste form nærmest uforståelig for computere. Det kan gøre det udfordrende at arbejde med ord og tekst i eksempelvis optimeringsalgoritmer. En af årsagerne til at computere har det svært med ord, er den nærmest usynlige sammenhæng mellem ord. Hvordan skulle en computer vide at sammenhængen mellem "Kvinde" og "Mand" er den samme sammenhæng som findes mellem "Datter" og "Søn" eller "Mor" og "Far"?

Selv foruden relationsproblemet har computere, og mennesker, svært ved at arbejde matematisk med ord, hvilket gør det besværligt at optimere matematisk på et NLP (Natural Language Processing) problem.

De tidligste NLP-algoritmer udvalgte et begrænset ordforråd, og repræsenterede hvert ord som en onehot-vektor. En onehot-vektor er en vektor hvor alle værdier er 0, på nær én værdi på 1. Hver dimension af en onehot-vektor repræsenterer derfor et ord i det udvalgte ordforråd, og på denne måde kan ord repræsenteres og arbejdes med matematisk.

Dog løser dette kun et af vores to problemer, da vi stadig mangler at kunne lære computeren relationer mellem ord, hvilket er hvad word embedding kan gøre.

Formålet med word embedding er at repræsentere ord som en vektor så der både kan arbejdes matematisk med ord, men også således at vektoren indeholder information om ordets relation til andre ord.

13.2.1 Word2Vec

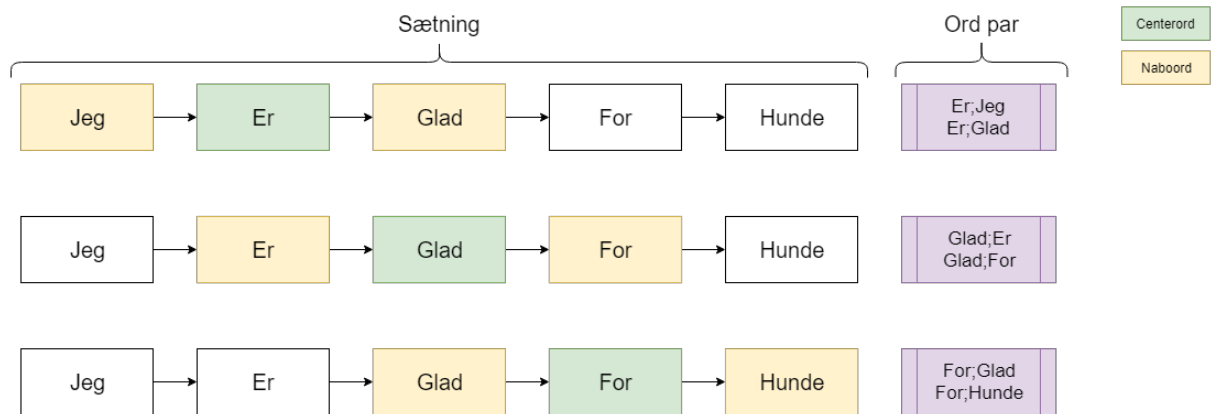
Word embedding handler om at omsætte ord til vektorer, og lade disse vektorer, når de minder om hinanden, være repræsentative for ord der har noget at gøre med hinanden. Til dette formål har vi i projektet valgt at anvende en lignende algoritme af Google's Word2Vec word embedding algoritme.

Word2Vec algoritmen består af et neuralt netværk, hvis formål er at placere hvert ord i et vektorrum således at lignende ord ligger tæt på hinanden, således at et ords betydning og relation til andre ord kan uddrages fra ordets placering i vektorrummet.

Der findes mange variationer af Word2Vec algoritmen, men i dette projekt har vi valgt at anvende skip-gram variationer som træner et ords word embedding ud fra dens naboord fundet i diverse sætninger fra et datasæt. Et eksempel på skipgram og de dannede ordpar kan ses på Figur 13.1 hvor en vinduestørrelse på 1 er anvendt. Vinduestørrelsen beskriver hvor langt til højre og venstre fra et centerord, vi kigger efter naboord.

Disse ordpar bruges dernæst som input- og output-vektorer i et neuralt netværk i onehot-format, således at netværket selv kan danne en word embedding repræsen-

13.2. Word Embedding



Figur 13.1: Eksempel på skipgram og dannelsen af ordpar med en vinduesstørrelse på 1

tation i det skjulte lag, hvilket senere bliver hentet ud af netværket når træningen har nået et acceptabelt niveau.

13.2.2 Netværk for ord

Selve netværket består af et input-lag, med en mængde af noder svarende til antallet af unikke ord i det udvalgte ordforråd. Output-laget skal være af tilsvarende størrelse. Imellem disse to lag ligger et enkelt skjult lag, hvori mængden af noder skal være mindre end input mængden. Dette gøres således at en mindre dimensionel og tættere repræsentation kan læres af netværket. Der kan ikke gives noget svar på hvad den optimale mængde er, og derfor er den eneste måde at finde et godt tal på, ved at prøve nogle forskellige og udvælge den bedste.

13.2.3 Træning og Anvendelse

Der skal først udvælges en liste af ord der vurderes til at være relevante. Dette kan gøres ved manuelt udvalg, men ofte vil det være mere effektivt at have en algoritme der udvælger dem. I projektet har vi valgt at arbejde med de 1.500 mest brugte ord i vores datasæt.

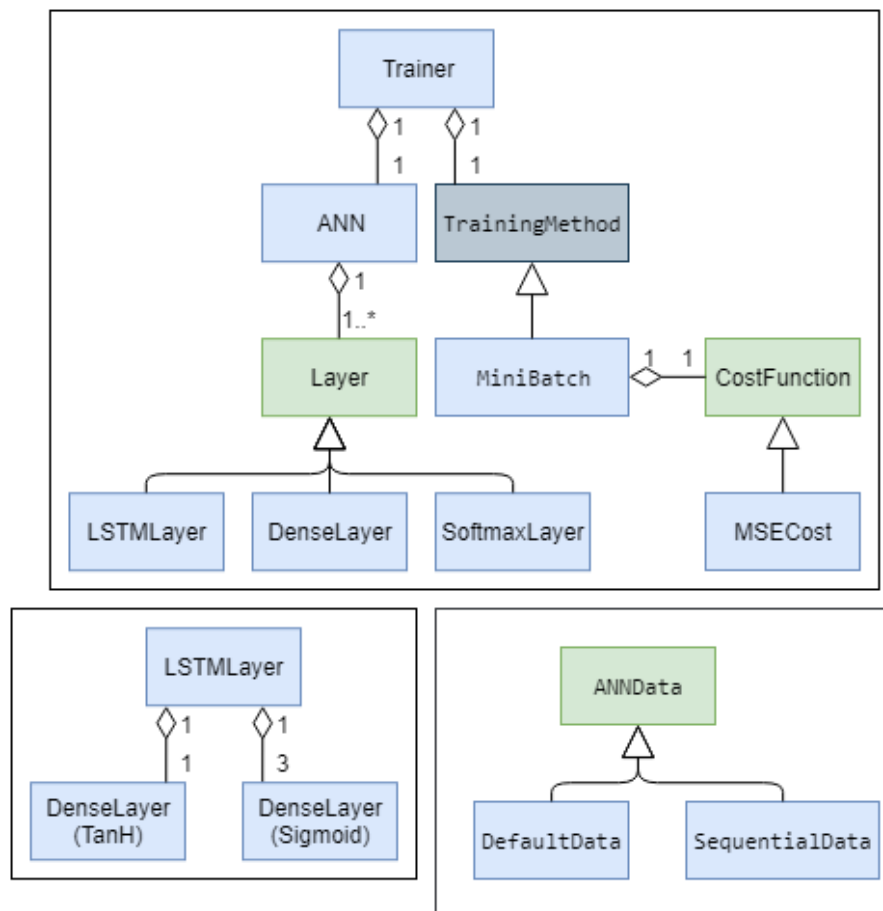
Netværket udsættes for træning til en kunstig opgave, forudsigelse af ord, baseret på tidligere generede ordpar igennem skipgram-metoden. Altså føres en vektor, repræsentativt for et ord, igennem netværket, og der fås så et output, hvor chancen for at et given ord er det næste ord, er angivet i form af et tal mellem nul og et, på den plads i vektoren der er repræsentativ for ordet. Vægtene rettes så til, gennem træningen, således at forudsigelse bliver så præcis som muligt. Når dette er opnået kan inputlaget og det skjulte lag anses som en tabel der repræsenterer de enkelte ords vektorværdier.

13.3. Netværksstruktur

Når der er opnået en forståelse af ordene i form af vektorer bliver det muligt at lave operationer med dem.

13.3 Netværksstruktur

Et udsnit af projektets centrale klasse- og interfacestruktur kan ses på Figur 13.2. En blå komponent er en klasse, en grøn komponent er en interface, og en grå repræsenterer en abstrakt klasse. I dette afsnit vil der forklares kort om hvilken essentiel rolle hver komponent har for programmets helhed.



Figur 13.2: Implementation af neurale netværk og træning

- *Trainer* klassen er ansvarlig for at optimere det neurale netværk, og mindske dens *cost*.
- *ANN* klassen er selve det neurale netværk, og aggregerer dermed flere *layers*.

13.4. Netværket som sentimentklassifikator.

- *Layer* interfacen bestemmer hvilke metoder der kræves at et lag. Mest centralt er evnen til at modtage input, udgive output, og backpropagation. Lag aggragerer ikke neuroner eller synapser, siden projektets struktur ikke udnytter separate objekter for at lave de udregninger som ofte abstraheres som neuroner og synapser.
- *LSTMLayer* implementerer *layer*. Det er projektets implementering af LSTM, og er en underklasse af *layer*. Et *LSTMLayer* indeholder fire *DenseLayers*. Tre af de *denselayers* bruger Sigmoid som en aktiverings funktion, og det sidste udnytter TanH som en aktiverings funktion.
- *Denselayer* er en implementering af et typisk normal feed-forward netværk, som udnytter aktiverings funktioner.
- *Training Method* indeholder evnen til at udregne hvor forkert et output er, og forventer at underklasser til at have en optimerings metode.
- *MiniBatch* er en optimerings metode som opdeler træning til kun at foretages mellem en bestemt mængde af kendt input og output sæt er blevet behandlet af netværket.
- *CostFunction* interfacen giver funktionalitet til at udregne cost af et forventet output relativt til input. Den forventer underklasser at underklasser har en metode som udregner fejlen relativt til et bestemt forventet tal, og bestemt aktuelt tal.
- *MSECost* implementerer *CostFunction*, og bruger ligningen *Mean Squared Error* til at udregne cost.
- *ANNDData* er klassen som er input typen til ANN. Dette betyder, at for at kunne evaluere data skal det først omdannes til *ANNDData* formatten.
- *SequentialData* er en input datatype, hvor det foretages i en sekvens. Dette er brugt i LSTM netværker, siden at tidligere data har en betydning for fremtidige resultater.

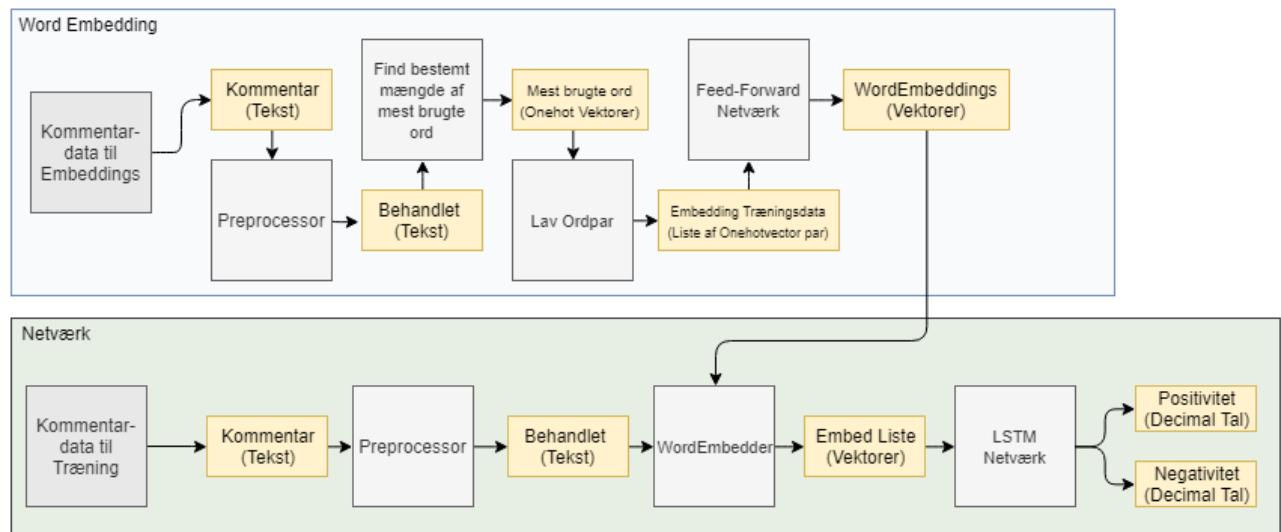
13.4 Netværket som sentimentklassifikator.

For at kunne udnytte de neurale netværk til sentimentklassifikation, skal både programmet samt data forberedes. Et overblik over denne proces kan ses på Figur 13.3, og vil blive forklaret i dette afsnit.

13.4.1 Word Embedding

For at netværket kan bruges, skal der skabes *Word Embeddings*. I vores implementation, kan disse embeddings både repræsenteres som en matrix, eller en liste af

13.4. Netværket som sentimentklassifikator.



Figur 13.3: Implementation af netværket som sentiment klassifikator.

vektorer. Det er derfor kun nødvendigt at foretage denne proces én gang.

Processen starter med et sæt af kommentarer som bliver behandlet af projektets *Preprocessor* som konverterer kommentarerne til små bogstaver, samt fjerner symboler og stopord. Derefter bliver alle ordene i kommentarerne talt op. Den totale sum af hvert ord vil midlertidigt opbevares, og derefter fjernes ord som ikke er brugt ofte nok til at indlæring, kan fremkomme. De mest brugte ord bliver omdannet til en onehot-vektorrepræsentation. Derefter bliver der dannet ordpar, som skaber par af de mest brugte ord som er tæt på hindanden i kommentarerne. Dette danner træningsdata til embeddings, som i vores implementation er en liste af onehot-vektorpar. Efter netværket er blevet trænet, skabes der word embeddings, som kan bruges af LSTM netværket.

13.4.2 Netværk

Denne proces er en del af den eventuelle vurdering af kommentarer. Kommentar-data er af samme type og format som ved word embedding, og bliver behandlet af projektets *preprocessor* på samme måde. Derefter bruges de udviklede *Word Embeddings* til at omdanne kommentarer til en liste af embeddings, som tager form som vektorer. Disse vektorer kan bruges som input til LSTM netværket, som derefter udregner 2 tal mellem 0 og 1: Hvor sikker algoritmen er på kommentaren er positiv, og hvor sikker den er på kommentaren er negativ. I vores implementation, bliver det bestemt om den er positiv eller negativ afhængig af hvilket tal er størst.

13.5 Opsummering

Et neuralt netværk er en algoritme, som kan behandle abstrakte problemer, og optimeres ved brug af datasæt. Et LSTM netværk har en hukommelse, som gør at data kan behandles i relation med tidligere input.

Word embedding skaber sammenhæng mellem ord, som er essentielt for at et program skal kunne tyde ords betydning i relation til andre ord. For at kunne skabe disse relationer omdannes ord til vektorer.

word embeddings skal kun udvikles én gang, og netværkets struktur er et samspil mellem feed-forward netværker, LSTM netværker, og word embeddings.

Denne maskinlæringstilgang består af mange beregninger som opstilles med lineær algebra. På grund af kvantiteten af beregninger, er en optimering af dem nødvendigt for at kunne træne maskinlæringsmodellen i en realistisk tidsramme.

Kapitel 14

Optimering af lineær algebra

Projektets neurale netværk er blevet implementeret i form af matricer. Dette har fordelen, af lineær algebra kan implementeres og bruges til at strukturere og opdele det neurale netværks processor. I dette kapitel, vil det vises hvordan centrale lineær algebra funktionaliteter er blevet optimeret, for at øge udregnings- og indlæringshastigheden.

14.1 Multiplikation

Matrixmultiplikation er en essentiel del af det neurale netværks feed forward og back-propagation. Multiplikation foretages mange gange, og derfor er der stort udbytte i at optimere den del af programmet. Multiplikation kombineret med Transponering bliver også ofte brugt i vores implementering. Metoden for at gange 2 matricer sammen kan ses nedenunder på Listing 14.1. Centrale dele af metoden vil nu gennemgås i efterfølgende afsnit.

Listing 14.1: Centrale del af mult metoden i DefaultMatrixBehavior.java

```
1
2  @Override
3  public Matrix mult(Matrix B) {
4      Matrix A = this;
5      . . .
6      final int aRows = A.getRows();
7      final int aCols = A.getColumns();
8      final int bRows = B.getRows();
9      final int bCols = B.getColumns();
10
11     MatrixBuilder endMatrix = new MatrixBuilder(A.getRows(),
12         B.getColumns(), true);
13
14     Parallel.forIteration(
15         aRows,
```

14.2. Matrix Data Struktur

```
15         (Parallel.MINIMUM_MATRIX_ENTRIES_FOR_PARALLEL <=
            aRows * aCols ||
            Parallel.MINIMUM_MATRIX_ENTRIES_FOR_PARALLEL <=
            bRows * bCols),
16     i -> {
17         for (int a = 0; a < aCols; a++) {
18             double localityTemp = A.getEntry(i, a);
19             for (int b = 0; b < bCols; b++) {
20                 endMatrix.additiontoEntry(i, b,
                    localityTemp * B.getEntry(a, b));
21             }
22         }
23     });
24     return endMatrix.buildDenseMatrix();
25 }
26
27 public Matrix multTrans(Matrix B) {
28     return this.mult(B.transpose());
29 }
30
31 public Matrix transMult(Matrix B) {
32     return this.transpose().mult(B);
33 }
```

14.2 Matrix Data Struktur

En matrices datastruktur har en stor betydning for en matrices funktionalitet og optimering. Det er derfor vigtigt at skabe en robust datastruktur som ikke ofrer hastigheden af at efterspørge og lagre data. Matricer i projektet er Immutable, som betyder at deres data ikke kan ændres på andre måder end gennem dens constructor. For at skabe nye matricer, kan et objekt af klassen *MatrixBuilder* utilises. Matrixbuilder objekter tillader intuitivt at skabe nye matricer ved brug af flere forskellige metoder. Dens interne datastruktur tillader også at bruge enten dynamisk allokering, eller strikt allokering. Det opnås ved brug af en *InternalMatrixStorage* interface udviklet specifikt for MatrixBuilder.

Listing 14.2: Første Linjer af MatrixBuilder.java

```
1 public class MatrixBuilder {
2     private InternalMatrixStorage internalStorage;
```

Interfacen "InternalMatrixStorage" bliver implementeret af 2 klasser: "DynamicMatrixStorage" og "StrictMatrixStorage". Metoder som efterspørger eller ændrer data i matricen udnytter matrixbuilderens internalStorage.

14.2.1 DynamicMatrixStorage

Denne måde at opbevare data på bruger objekter af klassen `ArrayList`, som indeholder `ArrayList`s, som indeholder objekter af klassen `Double`.

Listing 14.3: Første linjer af `DynamicMatrixStorage.java`

```
1 public class DynamicMatrixStorage implements
    InternalMatrixStorage {
2     private List<List<Double>> internalStorage;
```

Den største fordel ved denne implementering af datastrukturen, i forhold til klassens formål, er at plads let kan allokeres dynamisk[22]. Det er muligt at tilføje den mængde af rækker og kolonne som er nødvendigt, og `InternalMatrixStorage` objektet vil justere sig selv til at akkommodere efterspørgslen.

Derimod, er der også ulemper ved denne struktur. Selvom `ArrayList`-klassen er optimeret, kræver det stadig en operation at allokere mere plads når ny data tilføjes til listen. `ArrayList` objekter er også generisk, som gør at primitive typer ikke kan bruges. Dette betyder at der skal bruges `Double` objekter, hvor der foregår Auto-boxing og Unboxing. Dette er omdannelsen af `Double` objekter til den primitive type `double`, og tilbage igen.

Tilsammen gør dette, at `DynamicMatrixStorage` er lettere at arbejde med, men er derimod mindre optimeret for tungere operationer som kræver meget data i form af store matricer.

14.2.2 StrictMatrixStorage

Fremgangsmåden for denne klasse bruger 2-dimensionelle arrays som indeholder den primitive type `double`.

Listing 14.4: Første Linjer i `StrictMatrixStorage.java`

```
1 public class StrictMatrixStorage implements
    InternalMatrixStorage {
2     private double[][] internalStorage;
3     private int rows;
4     private int cols;
5
6     public StrictMatrixStorage(int strictRows, int strictCols) {
7         this.internalStorage = new
            double[strictRows][strictCols];
8         this.rows = strictRows;
9         this.cols = strictCols;
10    }
```

14.3. Parallelisering

Fordelene ved denne implementering, er at plads kun skal allokeres én gang, som gør fremtidige operationer hurtigere, da der ikke skal bruges tid på allokering af yderligere plads. En anden fordel er at den indeholder primitive typer, som gør at Autoboxing og Unboxing undgås.

Ulempen ved dette, er at alt plads skal allokeres før nogen lagring af data kan foretages. Arrays har heller ikke nogen indbygget evne til at allokere mere hukommelse hvis nødvendigt, i modsætning til ArrayLists.

StrictMatrixStorage er dermed bedre hvis den endelige størrelse af den matrix kendes, før den skal bygges af MatrixBuilder.

14.2.3 Datalagring i multiplikation

På linje 10 af koden i Listing 14.1 bliver der skabt en Matrixbuilder. De 2 første parametre allokerer den ønskede mængde af hukommelse, hvor den første bestemmer mængden af rows der skal allokeres til, og anden parameter bestemmer mængden af columns. Den sidste parameter er en boolean som bestemmer om matricen er *strict* eller *dynamic*. I dette tilfælde er den sat til at være "true", som tvinger den til at være strict. Dette er optimalt, siden vi kan finde størrelsen af den resulterende matrix fra multiplikationen. Den har samme mængde af rækker som matrix A, og samme mængde kolonner som matrix B. Dermed er det optimalt at bruge StrictMatrixStorage.

14.3 Parallelisering

Parallelisering er en ideel optimeringsmetode for matrix multiplikation. Paralleliserede udregninger omhandler udregningen af flere resultater på én gang. For at opnå dette, skal et problem kunne opdeles i mindre dele som kan løses uafhængigt af hinanden. For matrixmultiplikation kan dette opnås let, fordi resultatmatricens værdier er uafhængige af andre værdier i sig selv.

Parallelisering opnås igennem den statiske metode *forIteration* i projektets *Parallel* klasse. Klassen indeholder den mængde af threads som er tilpasset afhængig af systemet programmet kører på.

Metoden *forIteration* efterligner en normal for-løkke, men med nogle centrale forskelle:

En ny valgfri input parameter kan bruges, og et eksempel af dette kan ses på linje 14 af Listing 14.1. Det er en primitiv type *boolean* som bestemmer om iterationen skal paralleliseres.

På grund af at iterationen ikke er parallel, gør det også at iterationsvariablen ikke kan ændres på udover en fast værdi mellem iterationer. Derudover må en iteration

ikke baseres på andre iterationer, siden det ikke kan forventes hvornår en bestemt iteration er færdig.

Der er også en fastsat mængde af iterationer. Dette betyder at det ikke er muligt at bruge en boolean tilstand til at bestemme om løkken skal fortsætte. Dette er på grund af hvordan parallelisering er implementeret, hvor det forventes at ønskede resultat opnås efter en kendt mængde af iterationer. Dette er på grund af at paralleliseringen giver hver thread en bestemt liste af iterationer som den er ansvarlig for, før iterationer starter. For at ligefordele dette skal den totale mængde af iterationer kendes.

14.4 Lokalitet

Lokalitet omhandler at dataefterspørgsler er tætliggende, og kan dermed øge operationshastigheden. I dette afsnit vil lokalitets betydning i forhold til matrix multiplikation udforskes.

Lokaliteten af data giver stort udbytte når det kommer til optimering af operationshastigheden. Lokaliteten øges 2 steder i kodeudsnittet: Ved brug af hårdtransponering, og variabelen `localityTemp`:

- Variabelen `localityTemp` udnytter at alle operationer i næste for-løkke skal bruge den samme værdi. Dermed opnås der en større lokalitet ved at efterspørge den værdi én enkelt gang og gøre den lokal, imodsatning til at efterspørge data som er lagret længere væk flere gange. Dette optimerer informationsbrug i relation til caching
- Hård Transponering af matricer er en krævende operation, fordi alle felter skal genskrives i en matrice ved nye positioner. Det kan derfor virke underligt, at på linje 27 og 31 foretages der en hård transponering.

Vores lineær algebra bibliotek understøtter også at lave en blød transponering, som ikke flytter på noget data, men ændrer derimod på hvordan dataefterspørgsler behandles. En blød transponering har en kompleksitet på $O(1)$, og hård transponering på $O(nm)$, hvor n er mængden af rækker og m er mængden af kolonner. Som det kan ses på Listing 14.1, er paralleliseringens tråde opdelt efter rækker. Dette betyder at lokaliteten er optimal, hvis en rækkes værdier er tætliggende. Dette opnås ved en hård transponering. En blød transponering har en anden lokalitet, hvor at en rækkes værdier er længere væk fra hinanden. Ved sammenligning af blød og hård transponering, kan der ses en forbedring af hastighed ved brug af hård transponering, i forhold til multiplikation.

Som det kan ses, tillader programmets multiplikations algoritme en optimering ved brug af at skabe et nyt variabel med en forventet værdi over flere iterationer, som øger lokalitet. Det kan også ses at programmets opsætning af matricer som er opdeling af rækker, tillader en højere lokalitet ved brug af en hård transponering.

14.5 Optimeringsbetydning

I projektets forløb er forskellige optimeringer blevet sammenlignet med tidligere versioner af programmet for at undersøge hvor stor betydning den har. Udregningshastighed blev også målt for at undgå eventuelle ændringer som ved implementation gjorde programmet langsommere. Fordi matrixmultiplikation af store matricer er en essentiel operation for det neurale netværk, og foretages ofte, vil der være fokus på sammenligning af tidligere multiplikationsmetoder i dette afsnit. For at kunne diskutere effektiviteten, vil den endeligt optimerede metode til matrixmultiplikation være pivotfaktoren på **1**. Dette betyder, at hvis en metode tager den dobbelte mængde tid af den endeligt optimerede metode, vil den have en faktor af 2. I alle tests er der foretaget flere matrix multiplikationer af 2000 x 2000 matricer. På grund af at hardware har en betydning for de første iterationer af multiplikationshastigheden, bliver der foretaget et sæt af multiplikationer som ikke er med i målingen af hastighed, altså en form for warm-up. Derefter foretages der flere multiplikationer som bliver målt. Mængden af multiplikationer skal være stor nok til at hastighedsmålingen kan foregå over længere tid for at få en mere gennemsnitlig udregningstid.

Metode	Faktor
Før Optimering	87,1
StrictMatrixStorage	4,4
StrictMatrixStorage, LocalTemp-Variabel	3,3
StrictMatrixStorage, Parallisering	1,4
Endelige Optimering	1,0

Table 14.1: Faktorforskelle mellem versioner af matrixmultiplikationsmetoden. Sammenlignet med 2000x2000 størrelse matrix multipliceret 80 gange.

Som det kan ses i Tabel 14.1, er der sket en betydelig forbedring på metodens udregningshastighed. En træningssessions udbytte efter 10 timer i den endelige metode er ækvivalent med en træningssession af lidt over en måned af den uoptimerede metode. Den første optimering som brugte primitive arrays gjorde den største forskel, med en proportional forskel af faktor 19,7.

Det kan ses at parallelisering har haft en faktor 3,1 forbedring på hastigheden, relativt til implementation uden.

Brugen af LocalTemp øgede lokaliteten, og havde en faktor 1,4 forbedring af hastighed. Tilsammen giver de en faktor 4,4 forbedring, relativt til metoden som kun bruger StrictMatrixStorage.

For at perspektivere projektets matrix multiplikation, har vi sammenlignet dens hastighed med open-source biblioteker. Den specifikke hastighedsforskel kan ses i Tabel 14.2. Programmets multiplikation er sammenlignet med SuanShu, som har

14.6. Afrunding

Biblotek	Faktor
ApacheCommons-3.14	250,3
Colt-1.2.0	151,6
jlpack-0.8	28,8
jblas-1.2.4	23,1
jama-1.0.3	22,7
ejml-0.25	20,9
R-2.14	20,2
mtj-1.0.4	20,0
ParallelColt-0.10.1	13,3
ojs-38-2	3,0
ujs-0.3.0	2,3
SuanShu-253	1,4
P3	1,0

Table 14.2: Faktor forskelle mellem versioner af matrix multiplikationsmetoden i andre biblioteker. Sammenlignet med 80 2000x2000 multiplikationer.

en større samling af målinger af dens hastighed i forhold til andre biblioteker. Disse målinger er brugt til udviklingen af figuren. Ud fra dette, kan det konkluderes at projektets matrixmultiplikation er veloptimeret i forhold til andre udviklede algoritmer og metoder.

14.6 Afrunding

Det lineære algebra bibliotek er blevet optimeret til at håndtere multiplikation af store matricer ved hjælp af effektive datastrukturer med god lokalitet. Der gøres brug af at denne beregning kan paralleliseres for at udnytte den tilgængelige computerkraft. Hermed er det nu praktisk at lave maskinlæringstilgangen til sentimentklassifikation, samt den hybride løsning som præsenteres i Delafsnit 16.3.3. Disse analyser skal nu kunne præsenteres for brugeren, hvilket gøres i den grafiske brugergrænseflade.

Kapitel 15

Grafisk Brugergænseflade

Med de bagliggende analyser på plads skal der en brugerflade til for at brugeren kan bruge disse analyser og deres resultater. Dette kapitel vil omhandle, hvordan den grafiske brugergrænseflade til dette er implementeret. Denne er baseret på prototypen og de tilhørende prototypetest. Der vil i denne kontekst også beskrives hvilken indflydelse usabilitytesten har haft, og hvilke ændringer der er foretaget efterfølgende på baggrund af dette.

15.1 Login

Loginsiden indeholder kun det absolut nødvendige: Felter til at indfylde brugernavn og kode, og en knap til at godkende sin indtastning. Der har ikke været noget at ændre fra prototypen. Hvis programmet skal tages i officiel brug burde det kobles til AAU loginsystemet, men den foreløbige udgave bruger et simpelt usikkert stand-in.

15.2 Sidebar

Sidebaren som kan ses på Figur 15.1 er konstant tilstede i den venstre side af vinduet, efter at en bruger er logget ind. Den holder en fast bredde, og strækker sig til at fylde vinduets højde. Faneblade holder sig til toppen af sidebaren og informationer om brugeren og hvilket semester de arbejder med for neden, så brugeren kan finde dem i de samme ender af vinduet uanset hvor stort eller lille de gør det. Knapper til at logge ud af systemet eller vende tilbage til semestervalgssiden står i den nedre del af sidebaren for at holde dem tæt ved deres henholdsvis relevante informationer. Knappen til at vende tilbage til starten (semestervalgssiden) er indikeret med et hus-ikon, men hele det afmærkede område omkring ikonet og dets tekstbeskrivelse udgør sammen knappen. Elementer af sidebaren synliggøres kun når de er relevante for indholdet der vises ved siden af sidebaren, så brugeren ikke behøver overveje om

15.3. Valg og oprettelse af semestre



Figur 15.1: Prototypeudgaven og den implementerede udgave af sidebaren

det er noget de burde fokusere på endnu. Dette gælder blandt andet for fanerne for de muligheder brugeren har når de har valgt et semester at arbejde på. Indholdet til hver fane holdes åben når det først er læst ind, så brugeren kan skifte imellem dem hurtigt og uden at miste noget.

15.3 Valg og oprettelse af semestre

Da siden til oprettelse af semester er en forholdsvis simpel side, er der ikke foretaget nogen ændringer i den endelige version, Figur 15.2, i forhold til udseendet af elementerne på prototypen, Figur 10.6. Med den undtagelse, at der ved start af oprettelses proceduren, altså når der er trykket op knappen "Opret", dukker en indikator op, for at vise at processen er i gang og derved undgå bruger input imens. Basissiden som vises når brugeren logger ind viser de semestre brugeren har analyser af til rådighed. Disse er sorteret i rækker for hvert halvår stillet op kronologisk med de nyligste øverst. Hvert semester kan trykkes på for at komme ind på analysen af det. I det øvre højre hjørne er en knap til at oprette et nyt semester. Oprettelse af nye semestre foregår på en separat side ved at give et navn, vælge semester nummer

15.3. Valg og oprettelse af semestre

The screenshot shows a web application window titled "Semester Feedback Analysator". The main content area is titled "Opret Semester" and contains the following form elements:

- Semesternavn:** A dropdown menu with "Software" selected.
- Semesternummer:** A dropdown menu with "5" selected.
- År:** A dropdown menu with "2019" selected.
- Season:** Two radio buttons, "Forår" (unselected) and "Efterår" (selected).
- Vælg Spørgeskema data:** A section with a yellow folder icon, a text input field containing "/home/user/data.csv", and a "Gennemse" button.
- Endeligt Semesternavn:** A text box displaying the generated name "Software5 2019-Efterår".
- Opret:** A button at the bottom of the form.

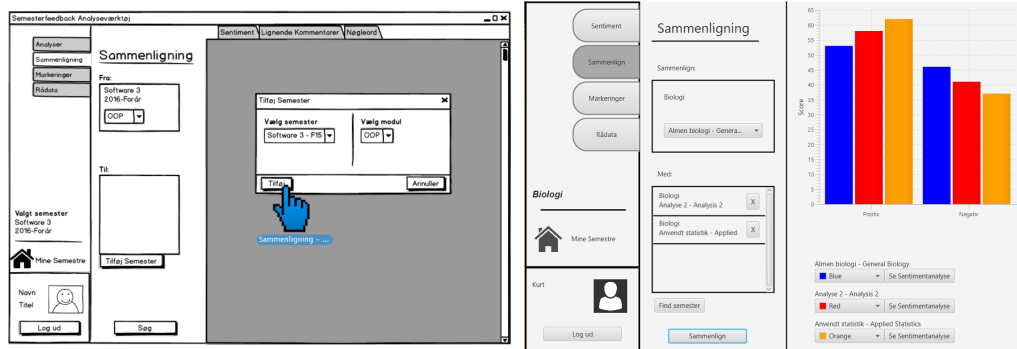
The left sidebar contains a navigation menu with a home icon and the text "Mine Semestre". Below this is a user profile section showing the email "Fake@coordinat or.com", a user icon, and a "Log ud" button.

Figur 15.2: Endeligt design af Semesteroprettelse i Feedback Analysator værktøjet

15.4. Sammenligning

og halvår, og give det en CSV fil med data for det semester fra SurveyXact. Når et semester oprettes sendes datapunkterne fra CSV-filen videre til sentimentprediktoren. Al data uploades til en MySQL database. Når analysen er færdig kan den tilgås fra semestervalgssiden.

15.4 Sammenligning



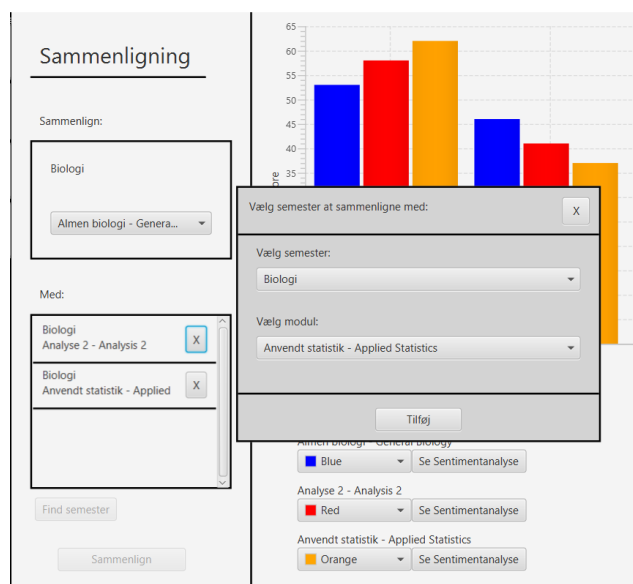
Figur 15.3: Prototypeudgaven og den implementerede udgave af sammenligningsfanen

Sammenligningsfanen som kan ses på Figur 15.3 og 15.4 indeholder endnu en sidebar som fungerer lige som den ydre: Den holder en fast bredde og fylder højden ud, igen for at holde elementer i samme relative position til hjørnerne af vinduet. Den er opstillet med to bokse over hinanden. Den øvre indeholder det semester som brugeren har valgt. Boksen indeholder kun et valg; hvilket modul brugeren vil sammenligne med andre moduler. En relativt betydelig mængde plads bruges på denne ene funktion for at passe til den anden boks, for at illustrere at det er de to boksers indhold der sammenlignes med hinanden.

Den anden boks starter tom, og op til 7 andre moduler kan udvælges til sammenligning. Denne boks tilpasser sig vindue-størrelsen for at vise så mange udvalgte moduler som der er plads til, og har en scollbar når den indeholder mere end der er plads til at vise. De valgte moduler vises som adskilte felter i boksen med en forkortelse af dets navn, og en knap til at fravælge dem fra sammenligningen igen. Ved at holde musen over disse felter kan en tooltip bringes frem, som viser navnet fuldt ud. Disse indre bokse er sat op så de ligner den første boks, igen for at pege brugeren ind på at disse bokse med et Semesternavn, Modulnavn, og en sort kant repræsenterer datapunkter til at sammenligne.

Når brugeren trykker på knappen til at føje et nyt modul til sammenligningen åbnes en ny boks hen over resultatfeltet og del af sidebaren som ses på Figur 15.4. Den er placeret der delvis for at bruge tilgængelig plads, men også for at illustrere at denne boks burde være brugerens fokus indtil de er færdige med den, uden at skjule nogen af de egentlige informationer i sidebaren som kan være relevante for hvad brugeren

15.5. Markering af kommentarer



Figur 15.4: Boksen til tilføjelse af et modul som sammenligningspunkt.

vil indfylde i det nye sammenligningspunkt boksen er åbnet for at oprette.

Boksen indeholder felter til valg af semester og modul. Hvis det valgte semester har et modul af samme navn som det der sammenlignes fra er det automatisk valgt på forhånd. Andre moduler kan også udvælges. I modsætning til prototypen er felterne længere og står under hinanden, da de potentielt skal indeholde meget lange navne.

15.5 Markering af kommentarer

Kommentarer kan markeres af brugeren med en af flere forudbestemte farver. De farver som kan anvendes til dette formål, er udvalgt ud fra nogle kriterier, som er med til at sikre, at farverne er behagelige for brugeren at kigge på, ved de elementer hvor de er anvendt.

15.6 Rådata

Det endelige udkast til Rådata siden, i prototypen ses i Figur 10.8. Den endelige version vises i Figur 15.5, og har en del ændringer fra prototypen. Først og fremmest kan fremhæves knapperne over tabellen med dataene; knappen med teksten *Filtrer* er fjernet, og i stedet er der en boks hvor der kan vælges synlige kolonner. Fjernelsen af knappen kom som følge af at der ved implementeringen af tabellen, blev brugt *FilteredList<T>* som datakilde. Denne klasse giver mulighed for at sortere indhold på baggrund af et predikat, uden faktisk at slette data fra listen [23]. Da den kan instantieres med en *ObservableList<T>* opdateres den så snart den underlæggende

15.7. Kommentarisering

Modul	Respon...	Vurderi...	Vurderi...	Aktivite...	Aktivite...	Forslag...	Aktiv d...	Forbere...	Faktore...
Grundl...	415230...	1	1	Tavleun...	Ingenting	Mere ...	3	2	Undervi...
Grundl...	415230...	1	1	Undervi...	Bøgen	Mere ...	3	2	Opgaver
Lineær ...	415230...	1	1	Undervi...	Ingenting	Ingenting	3	2	Undervi...
Grundl...	415230...	3	4				1	1	
Grundl...	415230...	1	1				2	3	
Lineær ...	415230...	2	2				2	2	
Grundl...	415230...	2	1	Grupper...		Ofte ...	2	2	Regnin...
Grundl...	415230...	2	2	Grupper...			2	2	Regnin...
Lineær ...	415230...	1	1	Ingen			2	2	Regnin...
Grundl...	415230...	1	1				4	2	
Grundl...	415230...	1	1	Gul			3	2	
Lineær ...	415230...	1	1	Grøn			2	2	
Grundl...	415230...	2	3	Lyserød	kanik	en ...	2	2	termody...
Grundl...	415230...	1	1	Blå	gaver...		2	2	hjemme...
Lineær ...	415230...	2	2	Rød	uppea...		2	2	hjemme...
Grundl...	415230...	2	3	Forelæs...			3	2	Eksame...
Grundl...	415230...	1	1	Opgave...			3	2	Eksame...
Lineær ...	415230...	1	1	Kombin...			4	2	
Grundl...	415230...	3	3	Gruppe...	De sen...	Ved ...	3	3	Hjemm...
Grundl...	415230...	1	2	Jeg kan ...	Statik o...	Statik o...	2	1	
Lineær ...	415230...	2	1				2	1	
Grundl...	415230...	2	2	opgaver...	Termod...		1	2	at opsø...

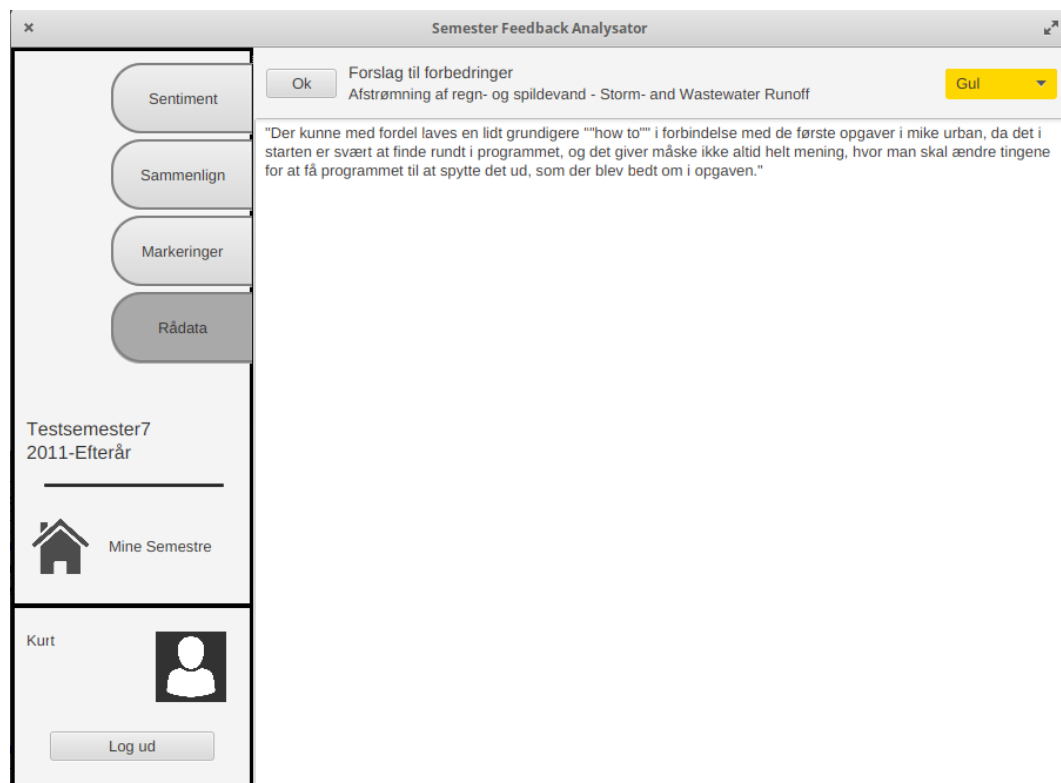
Figur 15.5: Endeligt design af Rådatasiden i Semester Feedback Analyzator værktøjet.

listes elementer ændres, og da den selv implementerer dette interface, bliver rækkerne i tabellen opdateret så snart prædikaterne ændres [24]. Prædikaterne direkte baseret på det valgte element i kombo-boksen, og indholdet i søgefeltet. Da der ved denne implementering ikke var noget observerbart ventetid på øjeblikkelig ibrugtagelse af ændrede filtre, selv når der skulle sammenlignes på over 18.000 strenge, blev det besluttet at tage filtrene i brug det øjeblik de blev indstillet, i stedet for at kræve tryk på en knap. Dette er med undtagelsen af søgefeltet, hvor der stadig skal trykkes *Søg* eller trykkes på enter-tasten mens feltet endnu er i fokus. Knappen *Åben i excel-format* er desuden ikke blevet implementeret, da der ikke har været tid til at implementere dette på en ordentlig måde.

15.7 Kommentarisering

Ved implementering af Rådata layoutet, og intern test af dette, blev et problem tydeligt: Det var ikke muligt at læse en lang kommentar i en celle i tabellen, dette ses også i Figur 15.5. Der blev ikke fundet en konkret løsning på denne udfordring, og dermed måtte der i stedet findes et fix. I dette tilfælde bestod det i, at lade brugeren dobbeltklikke på en celle med en kommentar for at åbne og tillade læsning

15.8. Afrunding



Figur 15.6: Kommentarvisning, et nyt layout der blev lavet til at fremvise et enkelt svar, fra rådata eller under fanen markeringer.

af denne. Et eksempel kan ses i Figur 15.6, her vises øverst, fra venstre mod højre: en *Ok* knap, der vil bringe brugeren tilbage dertil hvor kommentaren blev åbnet fra. Spørgsmålet, som kommentaren er skrevet til står midt for, med modulnavnet under. Længst til højre er en kombobox der tillader brugeren at vælge hvilken farve kommentaren skal markeres med. Det store område under, er beregnet selve kommentaren.

15.8 Afrunding

Hermed er der udviklet en brugerflade der dækker over organisering og præsentation af de bagvedliggende sentimentanalyser, igennem grafer over sentimentester semester-vis og ved sammenligning af enkelte moduler. Ved siden af det kan de rå data præsenteres efter brugerens ønsker og markeres til speciel uddragelse i en separat fane.

Sammenholdning af kravspecifikation og det faktiske program foregår i Kapitel 20, ydermere fremhæves problemer der blev opdaget i programmet i Kapitel 18, samt løsninger på disse i Kapitel 19.

Del IV

Kvalitetssikring

Kapitel 16

Validering

For at sikre, at produktet og det udviklede system opfylder de krav og kriterier som er forudbestemt, vil de følgende kapitler evaluere kvaliteten af produktet. Først er validering af, at de primære funktioner udfører deres handlinger korrekt og med gode resultater.

Dette kapitel vil uddybe hvordan det endelige programs evne til at analysere kan måles og evalueres. Derefter vil programmets analysekomponenter måles og valideres separat, for at bestemme effektiviteten af projektets analysemodeller.

16.1 Cross validation

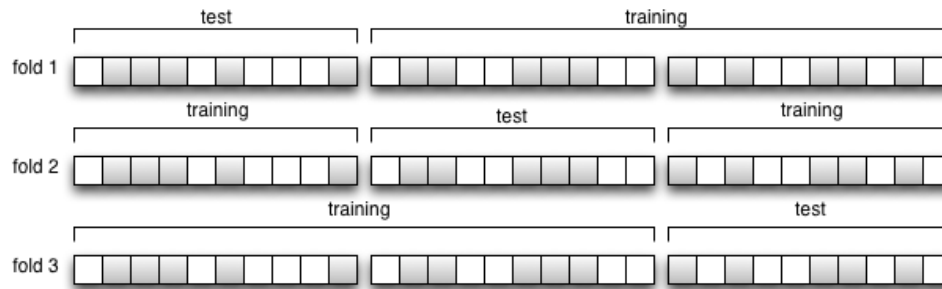
For at kunne teste de implementerede analysemodellers evne til at klassificere sentiment på kommentarer, skal de igennem en validering. Valideringen deler datasættet op i træning- og testdata. Modellen trænes på et træningssæt hvor den kender svaret, og derefter testes dens klassificeringsevne på et testsæt. Ved at træne på én del af datasættet og teste på en anden, sikres det at modellen ikke overfitter - altså ikke bare lærer svarene på dataet udenad - men derimod er i stand til at generalisere.

Eftersom der er flere måder at dele datasættet op i test og træningssæt, kan det risikeres at få en skæv fordeling, som modellen enten er rigtig god til at klassificere, eller rigtig dårlig. For at undgå dette dele det op i flere folds. En illustration af opdelingen af data med "3-fold cross validation" fremgår af Figur 16.1.

I figuren er de tre folds essentielt tre forskellige instanser af modellen, eftersom den er trænet på forskelligt data. Den gennemsnitlige klassifikationsevne over de tre folds resulterer i modellens overordnede præstationsevne.

3-fold cross validation vil i de kommende afsnit blive brugt for at sammenligne resultater.

16.2. Nøgleordsanalyse



Figur 16.1: Illustration over datafordeling i 3-fold cross validation [25]

16.2 Nøgleordsanalyse

Nøgleordsanalysen introducerer forskellige variable der kan påvirke dens egenskab til at klassificere sentiment. Dette afsnit vil undersøge nogle forskellige kombinationer af disse og sammenligne resultaterne deraf.

16.2.1 TFIDF-Parametre

TFIDF blev implementeret med muligheden for at til- og fravælge eller ændre hver af parametrene i den nedenstående liste:

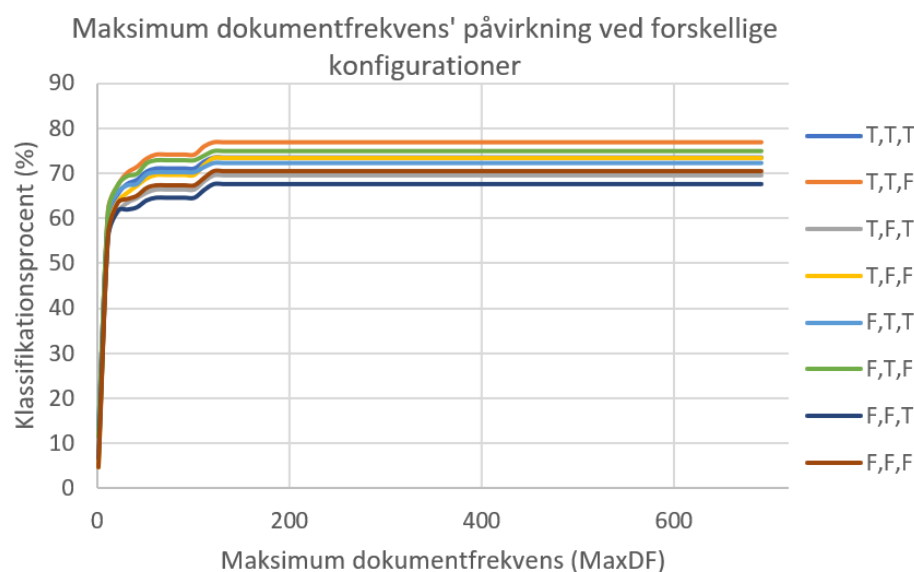
- Stopord (True, False): Fjern stopord fra datasættet, baseret på en liste over danske stopord.
- Negativ ordbog (True, False): Fastsat negativt sentiment på alle ord i en liste over negative danske ord.
- Positiv ordbog (True, False): Fastsat positiv sentiment på alle ord i en liste over positive danske ord.
- MaxDF (Heltal): Det maksimale antal kommentarer et ord må opstå i for at blive brugt.
- MinDF (Heltal): Det minimale antal kommentarer et ord skal opstå i for at blive brugt.

Disse parametre blev derfor hver især testet i en generel indledende test. Som nævnt i forrige afsnit, bliver resultatet af en konfiguration vurderet på dens gennemsnitlige klassifikationsprocent på 3-fold cross validation.

Figur 16.2 illustrerer forskellige konfigurationer, og hvordan maksimum dokumentfrekvens påvirker dem. MinDF er fastsat til 1 på alle konfigurationer, dermed består

16.2. Nøgleordsanalyse

konfigurationerne af (Stopord, Negativ ordbog, Positiv ordbog). Det fremgår, at ligemeget hvilken konfiguration, der bruges bliver klassifikationsprocenterne bedre jo større MaxDF-værdi der bruges. Fra en MaxDF-værdi på 120 og op til 700 - som er antallet af dokumenter i datasættet - er alle konfigurationer på sit bedste. Dette tyder på, at det ord der findes i flest forskellige dokumenter, er brugt i omkring 120. MaxDF bør altså fastsættes til et højt tal, for at optimere klassifikationsprocenten, og bliver derfor ignoreret, som parameter fremadrettet.

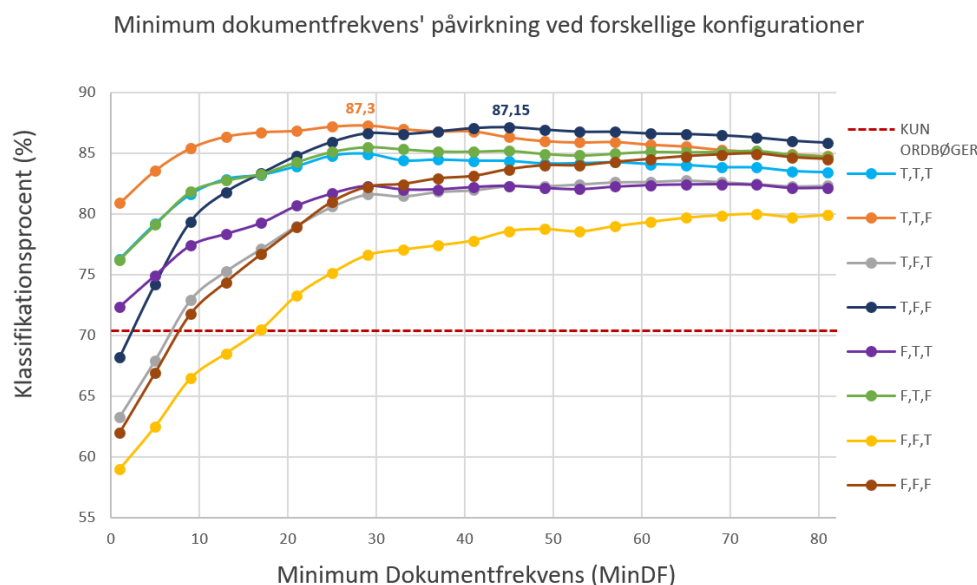


Figur 16.2: Illustration over hvordan MaxDF påvirker klassifikationsprocenten

Figur 16.3 illustrerer minimum dokumentfrekvensens effekt på alle konfigurationerne. Den røde stiplede linje viser hvor god modellen ville være, hvis kun de to ordbøger bruges, og der ses helt bort fra TFIDF-vægtene beregnet på træningsdataen. Alle konfigurationer når hurtigt op over modellen der kun bruger ordbøger, hvilket altså kunne indikere, at når ordbøgerne bruges, er det kombinationen med TFIDF, der gør den god.

På det anvendte datasæt fremgår det, at en stigning i MinDF-værdien forbedrer klassifikationsprocenten, indtil den når op omkring 60-70, hvor alle konfigurationerne flader ud. Den konfiguration der når den højeste klassifikationsprocent er den orange, med konfigurationen (True, True, False), som når en klassifikationsprocent på omkring 87,3. For at få et nærmere kig på denne konfiguration, er der lavet separate tests, som er illustreret i Figur 16.4.

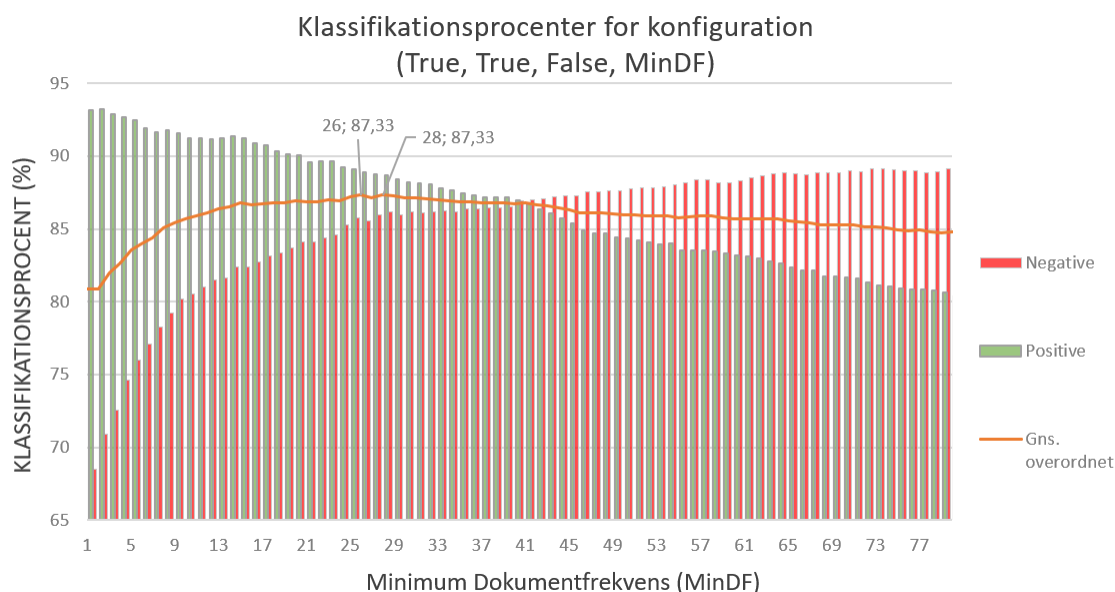
Figur 16.4 viser konfigurationen der fjerner stopord nærmere, denne benytter den negative ordbog, men ikke den positive. Figuren illustrerer hvor stor en procent-



Figur 16.3: Illustration over hvordan MinDF påvirker klassifikationsprocenten

del af klassifikationerne, der er positive og negative ved hver MinDF-værdi. Det fremgår, at der er en skæv fordeling mellem positive og negative klassificeringer ved lave værdier af minDF. Denne fordeling mødes omkring en minDF-værdi på 41 hvilket nærmer sig den maksimale klassifikationsprocent. Den maksimale klassifikationsprocent ligger på 87,33% ved minDF 26 og 28, hvilket betyder at modellen kan klassificere det korrekte sentiment på en sætning 87,33% af gangene, på det brugte datasæt.

Den fundne optimale konfiguration er specifikt bedst på dette datasæt, og der er derfor heller ingen garanti for - hvis den testes på anden data - at den kan klassificere korrekt sentiment 87,33% af gangene, eller at denne konfiguration er den bedste.



Figur 16.4: Illustration over hvordan MinDF påvirker både den positive, negative og overordnede klassifikationsprocent, ved konfiguration (Stopord, Negativ Ordbog, Ingen Positiv Ordbog)

16.3 Neuralt netværk

I dette afsnit vil maskinlæringstilgangen til sentiment analyse blive verificeret, og forskellige model parametre vil blive vurderet, for at finde en mere optimal løsning, end hvad første forsøg måske ville resultere i.

16.3.1 Word embeddings

Grundet den metode som word embeddings bliver genereret på, dannes der store mængder af data der skal trænes over, hvilket forsager at eksperimenterer med word embeddings og diverse parametre i modellen tager meget lang tid. Grundet dette problem, har gruppen valgt, kun at teste én parameter af word embedding modellen, og over kun 7 iterationer igennem datasættet.

Vi har valgt at teste i forhold til den parameter, der bestemmer hvor mange features vi danner i en embedding vector, med henholdsvis værdierne 50, 100 og 200 features. Hver iteration tager i gennemsnit omtrent 1 time, og med 7 iterationer over hver parameter indstilling har den totale træningstid været omtrent 21 timer.

Metoden, hvorpå word embedding testes, er hovedsageligt igennem en subjektiv vurdering. Denne vurdering sker ved simpelt at kigge på embeddings, og vurdere om de giver mening. Dette tester vi ved at finde de 10 ord, som ligger tættest på et udvalgt ord i vektorrummet i forhold til "cosine similarity'en" mellem diverse ord

og deres vektorrepræsentation.

Gode word embeddings vil have lignende ord, og/eller ord med lignende kategori tæt på hinanden.

Et eksempel på projektets trænede embeddings kan ses på Figur 16.5, hvilket viser de 10 mest relevante ord i forhold til ordet "pakke" med henholdsvis embeddings vectors med 50, 100 og 200 features. På figuren ses nogle ord markeret med gul, disse ord er ord, som projektgruppen fandt mest relevante i forhold til ordet "pakke", og her kan det ses, at antal mest relevante ord stiger med antal features i embeddings vektorerne, hvilket kan være et tegn på, at desto flere features en embedding vektor har, desto bedre bliver de.

Mest relevante ord for: "Pakke"					
50 Features		100 Features		200 Features	
efter:	0.5416	leveringstid:	0.4663	butik:	0.5604
rigtigt:	0.4558	vende:	0.4612	vare:	0.5353
info:	0.4508	fortryde:	0.4454	lige:	0.5338
masse:	0.4319	rette:	0.4365	returneret:	0.5320
spurgte:	0.4261	dig:	0.4344	arbejder:	0.5317
åbne:	0.4202	forklaring:	0.4334	halvt:	0.5211
senere:	0.4177	spurgt:	0.4274	hel:	0.5124
sæt:	0.4125	computeren:	0.4267	opdateret:	0.5113
hund:	0.4011	før:	0.4247	få:	0.5009
flere:	0.3994	oppe:	0.4178	smidt:	0.4987

Figur 16.5: De mest relevante ord for ordet "Pakke" med embedding vectors med henholdsvis 50, 100 og 200 features. De gulmarkerede celler er de ord som subjektivt er vurderet mest relevant for ordet "Pakke"

På baggrund af denne observation, har vi valgt at træne vores neurale sentiment netværk, baseret på embeddings med 200 features. Dog skal det også observeres, at der stadig findes mange ord, som ikke tydeligt har nogen relevans for det udvalgte ord, og dette er en åbenlys hentydning til at modellen endten mangler mere træning, eller en finjustering af model parametrene, men da træningen af disse word embeddings er meget tidskrævende, har vi valgt at nøjes med disse embeddings.

16.3.2 LSTM sentiment netværk

For at finde gode parameter indstillinger for LSTM sentiment netværket, udøver vi på samme vis som tidligere, en gennemgang af diverse parameter og fortage en undersøgelse af påvirkningen ved ændring af parametrene.

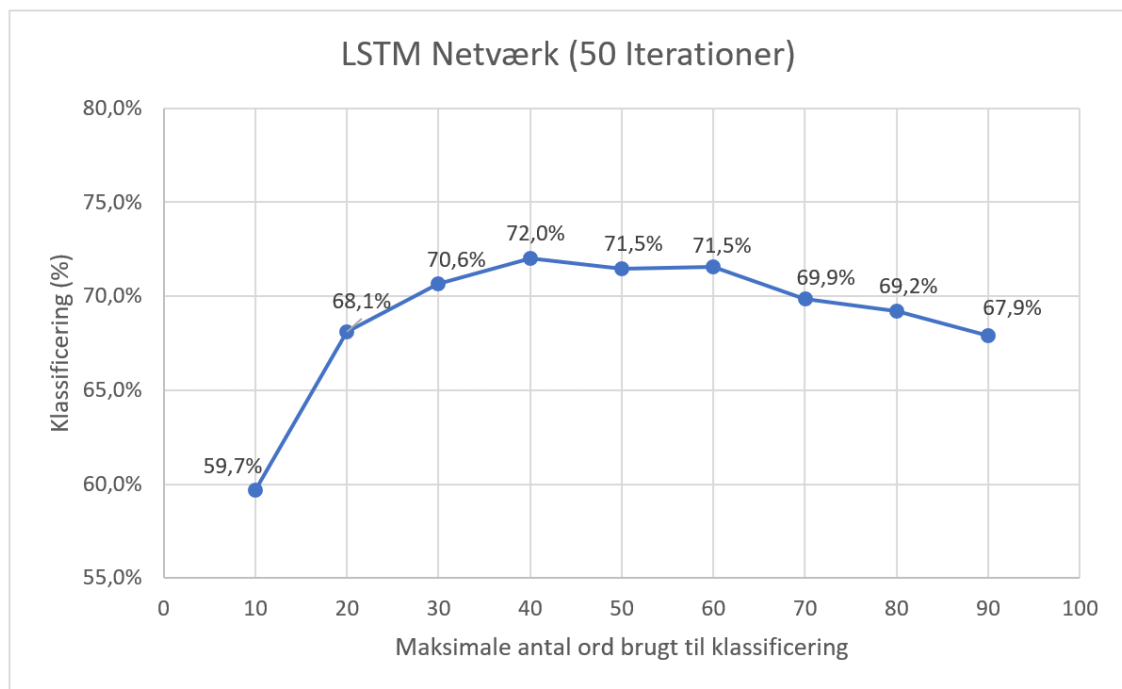
Der findes mange forskellige parameter, som kunne finjusteres, men vi har valgt at begrænse os til følgende tre: Maksimale antal ord i en sætning brugt til analyse, antal noder i LSTM netværket og antal trænings iterationer.

Antal ord til en sentiment vurdering

Det neurale netværk, hvis formål er at vurdere sentimentet på en tekststreng, skal have et fast antal ord, før den laver sin vurdering. Hvis en sætning har for mange ord glemmes disse, og hvis en sætning har for få ord, tilføjes nul ord, hvilket er ord uden betydning, indtil det rigtige antal ord er tilstede i sætningen. På trods af at nul ord ingen betydelig information bærer, kan de stadig påvirke klassifikationen, hvilket betyder, at et optimalt antal ord skal findes for netværket.

Vi udførte tests på henholdsvis 10 til 90 ord, med et spring på 10 ord mellem hver test. Resultaterne for disse tests kan ses på Figur 16.6, hvilket viser udvikling og den tilhørende klassifikationsprocent.

Vi kan observere, at de bedste resultater findes mellem 30 og 60 ord, med et toppunkt ved 40 ord. På baggrund af dette, vil fremtidige netværk være trænet på de 40 første ord af en sætning.



Figur 16.6: Graf over klassifikationsprocenten ved ændring af antal ord anvendt til en sentiment klassifikation.

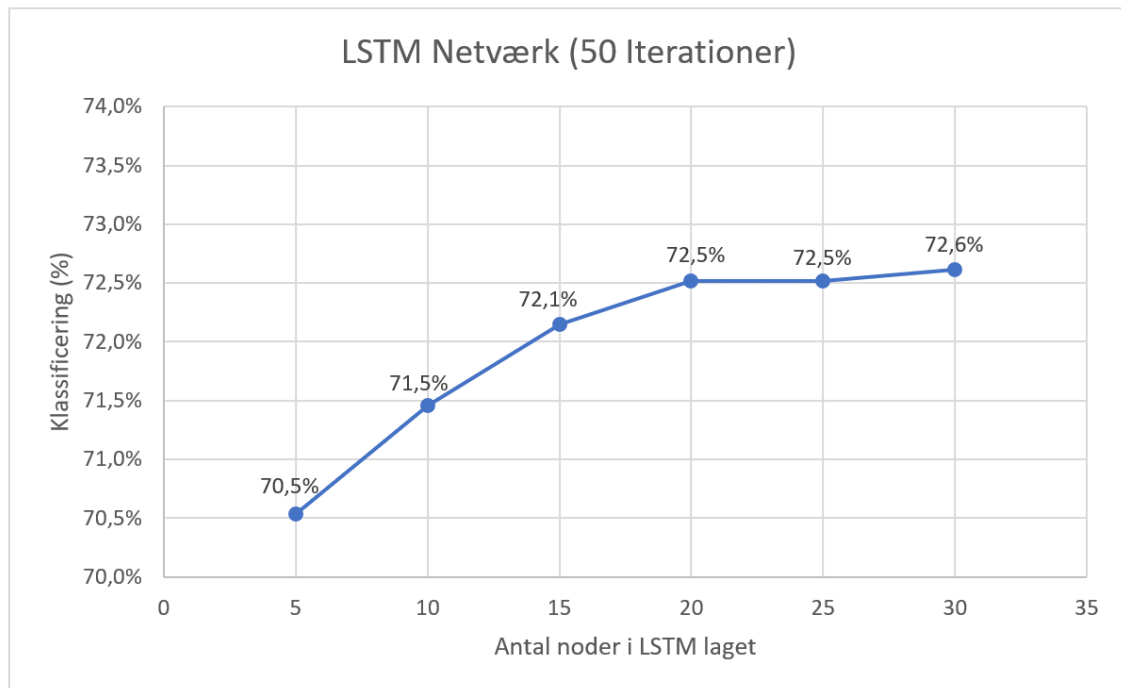
Antal noder i LSTM netværket

Det er LSTM laget i LSTM netværket, der gør netværket til en RNN i stedet for et normalt feedforward netværk. Størrelsen af dette LSTM lag ændrer på mængden af information, der kan opbevares internt i netværket over tid, og bør derfor en

16.3. Neuralt netværk

forøgelse af antal noder også øge effektiviteten af netværket. Dette har vi valgt at teste i intervallet fra 5 til 30 noder, med et spring af 5 noder, hvilket kan ses på Figur 16.7.

Der kan observeres, at effektiviteten af netværket stiger i takt med antal noder, men efter 20 noder er den øgede effektivitet meget begrænset. Flere noder øger desværre også træningenstiden, og eftersom meget lille øget effekt kan observeres efter 20 noder, har vi valgt at anvende denne parameter i vores afsluttende model.



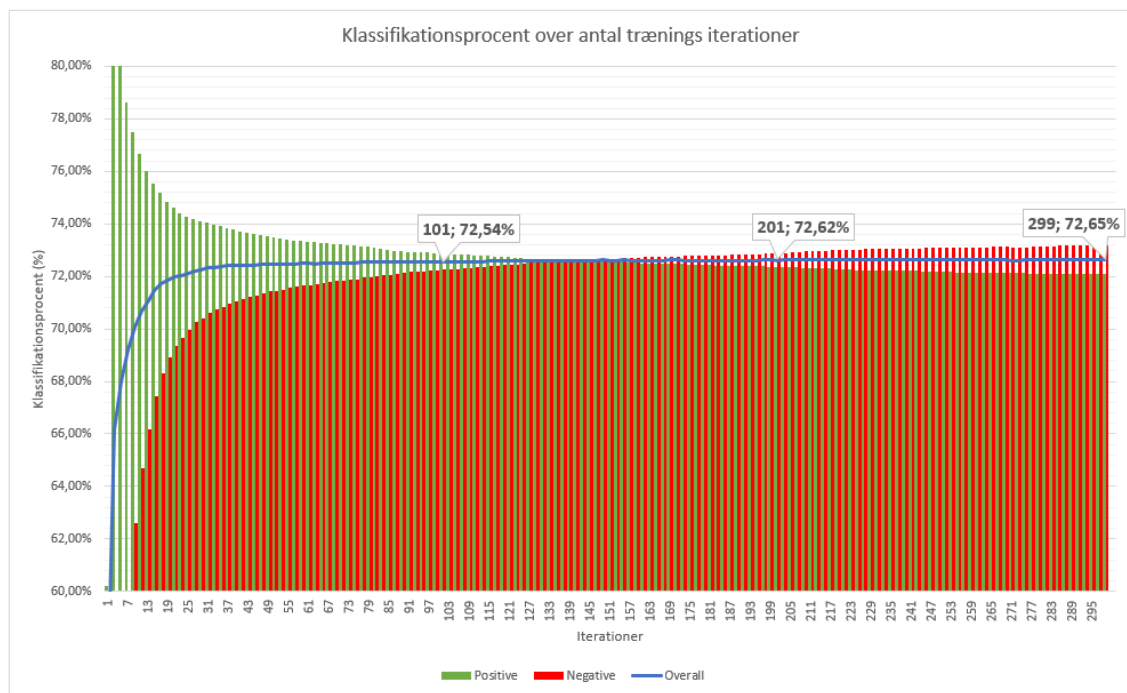
Figur 16.7: Graf over klassifikationsprocenten ved ændring af antal noder i LSTM laget i LSTM netværket.

Antal iterationer

Efter at have fundet passende model parametre, vil vi undersøge hvilken effekt antallet af trænings iterationer har på klassifikationsprocenten. Resultaterne for vores test kan ses på Figur 16.8, hvor der kan observeres, at effektiviteten stiger hurtigt over de første iterationer, men så langsomt aftager i hastighed. Der kan observeres at modellen fortsat, selv efter 200 iterationer, bliver bedre, dog med meget lille ændring. Denne lille udvikling fra 200 til 300 iterationer kan tyde på, at der er mere effektivitet at hente, men meget begrænset, og tiden det ville kræve at skaffe det, er måske for høj i forhold til den potentielle belønning.

Med de færdige parametre og iterationer, har LSTM netværket opnået en klassifika-

16.3. Neuralt netværk



Figur 16.8: Graf over klassifikationsprocenten over udviklingen af antal trænings iterationer

tionsprocent på 72,65%.

16.3.3 Hybrid løsning

For at lave en hybrid løsning mellem den statiske og neurale netværks tilgang, har vi valgt at lave, hvad vi har døbt et hybrid netværk. Det hybride netværk er et meget lille neuralt netværk med kun 2 lag, 4 inputs og 2 outputs, og dermed har modellen kun 8 vægte, der skal justeres.

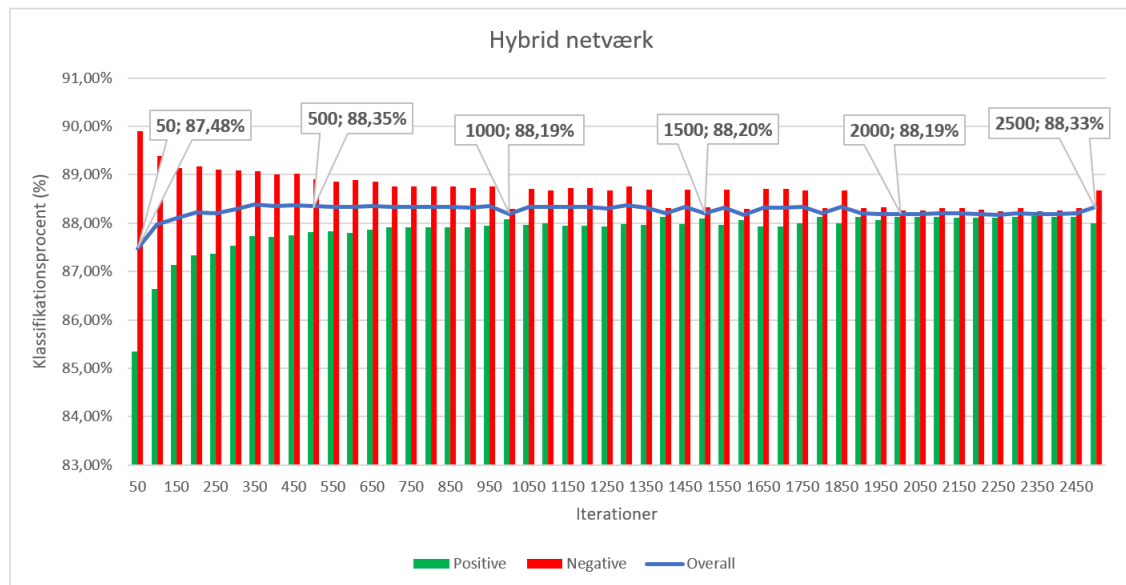
Resultatet fra det hybride netværk, som en følge af antal iterationer, kan ses på Figur 16.9.

Der kan observeres, at det hybride netværk har skubbet den samlede klassifikationsprocent fra 72,65% og 87,33% fra henholdsvis LSTM og keyword analyserne op til cirka 88,33%, hvilket er en samlet klassifikations stigning på 1%.

Selvom en stigning på kun 1% ikke virker af meget, er det interessant hvordan den hybride model ikke defineres, som summen af dens komponenter, men produktet af dem, hvilket tillader den at overgå de enkle modeller.

Det store klassifikations spring mellem LSTM og den statistiketilgang, kan være en potentiel synder, til hvorfor stigningen kun er på 1%. For at forbedre den hybride løsning, bør fokuset nok placeres på LSTM netværket, eftersom det er her

16.4. Opsummering



Figur 16.9: Graf over klassifikationsprocenten over udviklingen af antal trænings iterationer

den største forbedring nok er at finde. For at forbedre LSTM netværket, ville en potentiel løsning være bedre word embeddings, som også kort beskrevet i Delafsnit 16.3.1. Vi kan se på de nuværende word embeddings, at kvaliteten af disse er af ringere grad, eftersom stadig mange irrelevante ord dukker op omkring udvalgte ord. LSTM netværket afhænger af word embeddingerne, for at lave sin klassifikation, og hvis netværkets inputs er af ringe kvalitet, kan det være svært at forvente outputs af anden kvalitet.

16.4 Opsummering

I dette kapitel blev Cross validation brugt til at måle løsningernes præcision ved brug af 3-fold. Formålet var at måle og optimere præcisionen af de udviklede statistisk- eller maskinlærings baserede løsninger.

De forskellige parametre til den statiske løsning blev målt, og den bedste konfiguration blev valgt.

Maskinlæringsløsningen fik mængden af features i dens word embeddings, mængden af ord til sentiment vurdering, og mængden af noder målt og optimeret.

Til sidst blev begge løsninger kombineret til en hybrid løsning, som har en højere klassifikationspræcision end hver individuelle løsning.

Validering måler programmets effektivitet og kvalitet i forhold til at opfylde dens hovedfunktion. Derimod for at kvalitetesikre selve produktet, skal der også foretages unit testing.

Kapitel 17

Unit Testing

At teste programmets egenskab til at udføre forventede handlinger kan forsikre om, at programmet tilnærmelsesværdigt er implementeret korrekt. Dog kan det være svært, at få programmet til at udskrive nogle uddrag af koden for at se om funktionaliteten er korrekt, hele tiden. Dette kan i stedet gøres ved brug af unit testing, som er en måde at sikre, at allerede testede funktioner virker efter intentionen selv efter at nye funktioner implementeres. [26]

17.1 Teori

For at sikre programmets nøjagtighed og pålidelighed, skal de enkelte funktionaliteter testes. Pålidelighed til at programmet kører, og nøjagtigheden af dataet og analyserne er vigtig. For at sikre dette, testes enkelt "units" af koden. En unit kan have forskellige størrelser, men er ofte den mindste testbare del af programmet, hvilket kan være enkelte metoder eller hele klasser. Da units ofte kan modtage inputs, vil det være upraktisk at teste alle mulige inputs for alle units, men det antages at tests med få udvalgte inputs kan vise korrektheden af en given unit. [27]

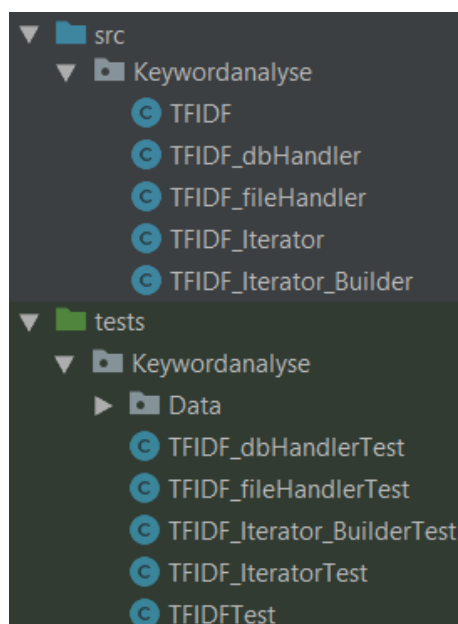
Princippet bag unit-tests er at isolere hver del af programmet, og påvise at disse individuelle dele er korrekte, med nogle meget specifikke regler, som koden skal opfylde. En af fordelene ved denne måde at teste på er, at den tillader programmører i systemet at refaktorere kode, uden at risikere et kæmpe fejlfindingsarbejde hvis ændringerne introducerer fejl. I en sådan situation kan unit-tests, enten forsikre programmøren om, at refaktoreringen overholder alle regler som er opstillet, eller afsløre præcis hvilke regler, der er blevet brudt - og herfra skal det så bestemmes, om det er de nye ændringer der skal genovervejes, eller om det er reglerne. [27]

En af ulemperne ved brugen af unit-tests i et program, er at programmøren kan risikere at få en falsk følelse af sikkerhed omkring rigtigheden af det, hvis alle tests overholdes. Unit-tests kan bevise tilstedeværelsen af fejl, men ikke bevise manglen på dem. [26]

17.2 Anvendelse af tests

Da programmering foregår i Java som beskrevet i Delafsnit 8.1.2, benyttes JUnit som er en unit-testing-struktur til Java.

Unit tests struktureres i en mappe med samme pakke- og klassestruktur, som koden der skal testes. På denne måde tydeliggøres det hvad der bliver testet i hvilke klasser. Et eksempel på en sådan struktur fremgår af Figur 17.1, hvor mappen "src" består af testbar kode, og mappen "tests" består af unit-tests af denne kode. pakkerne og klasserne fra "src" er identiske i "tests", udover en data-mappe i "tests" der gør brugen af data i tests uafhængig af hvilken anden data, der er tilgængelig i resten af projektet.



Figur 17.1: Illustration af en mulig struktur mellem kildekode og test af disse

Som det også fremgår af Figur 17.1, skrives JUnit tests i klasser. Enkelte tests tager form som metoder, men er annoteret med "@Test" for at tydeliggøre hvad dens opgave er. Denne annotering fremgår også i Listing 17.1, hvor to forskellige tests specificeres.

Listing 17.1: Eksempler på tests med JUnit

```

1 public class classNameTest {
2     @Test
3     void testEnTing() {
4         assert("Krav der tester en ting")
5     }

```

17.2. Anvendelse af tests

```
6
7     @Test
8     void testAndenTing() {
9         assert("Krav der tester en anden ting")
10    }
11 }
```

Her fremgår det også, at koden i hver test indeholder en "assert", som har til formål at tjekke at et bestemt krav er opfyldt, hvilket sikrer at testen resulterer i det forventede.

Listing 17.2: Test af at kunne tilføje og tilgå en koordinator

```
1 @Test
2 void setAndGetCoordinator() {
3     Coordinator coordinator = new Coordinator(ID, EMAIL);
4     structure.setCoordinator(coordinator);
5     assertEquals(coordinator, structure.getCoordinator());
6 }
```

Det ses af 17.2, hvordan en test i projektet kan være opsat praktisk, så den tester en funktionalitet i programmet. Denne specifikke test tester, om det er muligt at indsætte en ny semesterkoordinator i systemet, samt om denne semesterkoordinator kan tilgås igen.

Listing 17.3: Test af sammensat netværk

```
1 @Test
2 void concatNetwork() {
3     ANN resultANN = annBuilder
4         .concatNetwork(networkA)
5         .concatNetwork(networkB)
6         .build();
7
8     int sizeA = networkA.getLayers().size();
9     Layer layerA = networkA.getLayers().get(FIRST_ELEMENT);
10
11     Layer layerB = networkB.getLayers().get(FIRST_ELEMENT);
12
13     assertEquals(layerA,
14         resultANN.getLayers().get(FIRST_ELEMENT));
14     assertEquals(layerB, resultANN.getLayers().get(sizeA));
15 }
```

17.3 viser hvordan at det er muligt at teste, at funktionen *concatNetwork* kan sammensætte 2 neurale netværker. Dette bevises ved at sammenligne lagene i de individuelle netværker med det sammensatte netværk på de ækvivalente placeringer.

17.3 Dækningsgrad af unit-tests

Som nævnt, kan unit-tests bevise tilstedeværelsen af fejl, men ikke bevise manglen på dem. Dette vil sige, at selv hvis hele produktet er testet, vil der stadig være måder, hvorpå det er muligt at teste anderledes, og eventuelt støde på flere fejl.

En høj dækningsgrad af tests, sikrer at en stor del af programmet bliver kørt og at der ved nuværende brug af programmet, ikke er nogle synlige fejl.

Vores program har 630 unit tests, som overordnet set dækker 48% af alle linjer i projektet. De mest centrale dele har dog højere dækning, som det fremgår af listen herunder:

- Keywordanalyse: 89% linjer dækket
- MachineLearning: 68% linjer dækket
- LinearAlgebra: 88% linjer dækket

17.3.1 Manglende dækning

Størrelsen af projektets omfang gør, at tidsbegrænsninger resulterer i et ikke fuldt kvalitetssikret projekt med unit tests. Dette skyldes at nogle funktioner ikke nødvendigvis anvendes i projektet, men har stor relevans for eventuelle fremtidig udvikling. Produktet indeholder mere end 12000 linjer kode, hvoraf 8000 linjer er anvendt kildekode som kan testes, hvilket gør at det store omfang af tests bliver for omfangsrigt for projektets formål eller ikke essentielt.

Der er ikke skrevet tests af databasen, grundet manglen på en testdatabase. Der ikke et ønske om at blande databasetests med den egentlige data yderligere.

17.4 Afrunding

Til de centrale baggrundsfunktionaliteter er høj testdækning opnået for intern funktionalitet, men dækningen er mindre for kode opsat til fremtidig udvidelse, fravalgt funktionalitet, og databaseadgang.

Implementationen af den grafiske brugergrænseflade er svær at teste med unit tests, og disse tests ville have mindre betydning, da dets successkriterier i køretid bliver vist visuelt. For at sikre kvaliteten af brugergrænsefladen, vil Usability-testen delvist overtage ansvaret som test af brugergrænsefladen.

Kapitel 18

Usability Testing

I dette kapitel vil det udviklede produkt blive testet. Der vil blive udvalgt en testtype og testpersoner - samt en beskrivelse af testlederen og observatøren, deres opgaver og ansvarsområder. Derefter vil planen for udførelsen af testen blive gennemgået. Endeligt vil resultaterne af testene blive præsenteret, samt en beskrivelse af hvad resultaterne har medført.

Teorien omkring opsætning af testplan, udførsel af testen og datafortolkningen benytter Kapitel 3 og 5 i bogen: "Handbook of Usability Testing", [14][28].

18.1 Usability

Usability, eller anvendelighed, dækker over det at have et menneskecentreret design. Et system med en høj grad af anvendelighed vil blandt andet være [11]:

- Let at lære
- 'Effective' i den forstand at det består af passende funktionalitet og informationer på en organiseret måde
- 'Efficient' i den forstand at brugeren er i stand til at gøre ting med en passende indsats.

For at sikre at disse krav opfyldes i det udviklede system, laves der en usabilitytest. Da testen vil være relativ kort, skal det være 'Let at lære', for at kunne løse opgaverne. 'Effective' og 'Efficient' testes ved at sikre at brugeren er i stand til at udføre testene til et tilfredsstillende niveau, og på en tilfredsstillende mængde tid [11].

18.2 Testplan

I dette afsnit, vil der blive udvalgt en testtype, samt en begrundelse for valget og formålet ved netop denne test. Testpersonerne der skal udføre testen vil blive

18.2. Testplan

præsenteret, ligeledes vil testlederen og observatøren, som vil være tilstede under testen.

Dernæst vil fremgangsmåden for testen gennemgås, hvori opgaverne som testpersonerne skal udføre bliver præsenteret.

18.2.1 Udvalgelse

Testtypen, der er blevet udvalgt til usabilitytesten, er den verificerende test, som også blev dækket i Kapitel 10 under prototypetest. Den bruges sent i udviklingsforløbet, for at sikre at problemerne der blev opdaget i den formative prototypetest er blevet udbedret, og dermed at produktet efter bestemte standarder som vi opstiller, er brugbart.

Forud for udførelsen af den verificerende test bør der findes et sammenligningsgrundlag for hver opgave, hvilket kan være enten en tidsmåling eller en fejlmåling, som hjælper med at bestemme hvilken grad opgaven bliver udført til.

Testen udføres med minimal interaktion med testlederen, eftersom programmet bør være så udviklet at det ikke kræver eller løbende feedback på opgaverne.

Det centrale fokuspunkt ved denne testtype, er at samle kvantitativ data, i form af antal klik per opgave, tid per opgave, eller opgavens udførselstilfredshed på baggrund af de opstillede sammenligningsgrundlag.

Testpersonerne der skal udføre testen er igen informanterne, som er produktets målgruppe. Disse er valgt, da de kan fortælle hvorvidt de vil kunne se sig selv benytte produktet, i dets nuværende tilstand. Ydermere vil de kunne komme med ideer til videreudvikling. En ulempe ved at vælge samme testpersoner, som ved prototypetesten, er at der ikke kommer friske øjne på grænsefladen.

Der vil være to gruppemedlemmer tilstede under testen, en testleder og en observatør. Testlederens opgave og ansvar er i denne test at sørge for at testpersonerne får udført alle opgaverne, samt at stille spørgsmål omkring mulige svagheder eller mangler i det testede produkt, efter testen er færdiggjort. Desuden, hvis testpersonen af en eller anden årsag ikke kan udføre sin opgave, eller har spørgsmål, da skal testlederen kunne hjælpe.

Observatørens opgave og ansvar er at sikre at testen foregår som planlagt, samt supplere testlederen, hvis behovet opstår.

18.2.2 Opsætning

I dette delafsnit vil der blive afklaret hvilke opgaver testpersonerne skal udføre, samt hvilke standarder produktet skal kunne opfylde.

Opgaver

I den verificerende test skal testpersonerne udføre en række opgaver, som bliver vejlet op imod standarderne.

Målene med disse opgaver er at sørge for testpersonen har testet de væsentligste funktioner, som produktet tilbyder. For at opstille opgaverne blev kernefunktionaltiteterne identificeret, og opgaver bliver lavet ud fra dette. Disse indebærer:

- Autentificering
- Håndtering af semestre
- Sentimentanalyse
- Markeringer
- Sammenligninger
- Håndtering af rådata

Opgaverne er derfor lavet med henblik på at ramme disse områder, ved at gå igennem programmets mulige funktioner, og på den måde også simulere en normal brugsgennemgang af programmet. Opgaverne som dette resulterede i er vist i Bilag F.

Standarder

Det er valgt at standarderne er målt på antal klik, som benyttes til at udføre opgaven med. Tabel 18.1 viser antal klik, der minimum skal benyttes for at løse opgaven. Standarden bliver sat efter dette, men med en margen på 2 klik pr. opgave. Opgave 6 kan løses på to forskellige måder, derfor er der to forskellige tal. Tabel 18.1 beskriver også det succeskriterie der skal opnås, for at opgaven kan ses som værende gennemført. I hver af disse succeskriterier gælder det, at det skal opnås uden hjælp fra testleder eller observatør.

Hvis de beskrevne standarder opfyldes, anses testen for værende en succes. Hvis derimod at standarderne ikke bliver opfyldt, da anses testen for at have fejlet, og da skal der undersøges hvorfor den fejler, og hvad der kan gøres.

18.2.3 Udførelsen

Testen kommer til at foregå i et lokale, hvor testpersonen, testlederen og observatøren kommer til at være til stede under hele testen. Testpersonen udfører testen siddende ved et bord, hvorpå der er placeret en computer med en mus. Computeren vil under testen, optage lyd og computerskærmen, således bliver testen dokumenteret. Eftersom det kun er computerskærmen der optages, og ikke videooptagelser af testpersonen, bliver testpersonens omgivelser og umiddelbare udtryk ikke dokumenteret. Det kræver dog en større opstilling for at gøre dette, hvilket ikke var

18.3. Testresultater

Opgave	Klik	Succeskriterie
Opgave 1	1	Login med brugernavnet specificeret i opgaven
Opgave 2	11	Et semester bliver oprettet ved navn 'Testsemester7 2011-Efterår', og med datafilen 'Testfil' tilknyttet
Opgave 3	1	Semestret ved navn 'Byggeri 2018-AUTUMN' vælges
Opgave 4	0	Indse at 'Sentiment'-fanen er startside
Opgave 5	2	Semestret ved navn 'Biologi 2018-AUTUMN' vælges
Opgave 6	3 - 5	En kommentar i 'Markering'-fanen markeret med en vilkårlig farve
Opgave 7	8	Gennemfør en sammenligning mellem 'Almen Biologi' og 'Analyse 2' i det nuværende semester
Opgave 8	5	Filtrer rådata så modulet 'Analyse' er tilvalgt og kolonnen 'Respondant' er fravalgt.
Opgave 9	1	Anvend logud-knappen

Table 18.1: Usabilitytest standarder

muligt her.

Testen starter med, at testpersonen sættes ind i hvad produktet kan, og hvad hensigten med det er. Dernæst får testpersonen udleveret opgaverne, som skal udføres. Efter introduktionen vil testpersonen udføre de opgaver der er blevet stillet, og hvis der er problem vil testlederen være til rådighed. Efter alle opgaverne er udført af testpersonen, afsluttes testen. Testlederen og testpersonen vil da evaluere produktet, samt hvordan gennemgangen af opgaverne gik. Endeligt, evalueres resultaterne i forhold til stillede standarder, af testlederen og observatøren.

18.3 Testresultater

I dette afsnit vil resultaterne fra de afholdte tests blive præsenteret, samt hvad disse test har medført.

Det var planlagt at benytte alle tre informanter til usabilitytesten, men ligesom ved prototypetesten, opstod der skemaproblemer, og der var ikke muligt at nå det. Derfor endte usabilitytesten med kun at blive foretaget med én informant, som var den samme ene informant, der var testperson til prototypetesten, som bliver beskrevet i Kapitel 10. Grundet tidspres var det ikke muligt at foretage en usabilitytest med de to andre informanter, på et senere tidspunkt, inden projektaflevering.

Testpersonen fulgte slavisk opgaverne indtil Opgave 6, da spurgte testpersonen om hjælp, og herefter blev opgaverne givet fra observatøren. Dette medførte at testpersonen ikke fik udført præcis de opgaver der var stillet, men lignende opgaver, og derved kom testpersonen rundt om de dele af programmet, som der var ønsket.

18.3. Testresultater

Resultaterne fra testen fremgår af Tabel 18.2, hvor opgaverne er fra Bilag F.

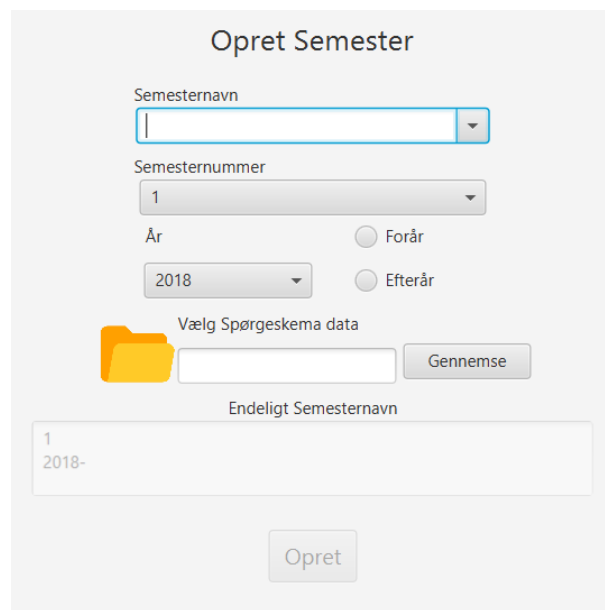
Opgave	Klik	Succeskriterie	Forklaring
Opgave 1	2	Opfyldt	Opgaven blev løst efter hensigten
Opgave 2	10	Delvist opfyldt	Der blev oprettet et semester med den rigtige datafil, dog blev navnet ikke genereret efter hensigten. I stedet for at semestret endte med at hedde: 'Testsemester7 2011-Efterår' hed den: 'Testsemester7 2011-Efterår1 2018-Efterår'. Derfor, benytter testpersonen også færre klik, end minimummet.
Opgave 3	1	Opfyldt	Opgaven blev løst efter hensigten.
Opgave 4	0	Delvist opfyldt	Testpersonen kommer ind på sentimentanalysen, dog går det ikke op for vedkommende at det er startside.
Opgave 5	2	Ikke opfyldt	Testpersonen har intet problem med at komme ud til semester oversigten, dog valgte personen et forkert semester. Så selvom at antal klik er på minimummet, er succeskriteriet ikke opfyldt.
Opgave 6	11	Ikke opfyldt	Der blev at testpersonen forestaget 11 klik, før der blev opgivet, og måtte bede om hjælp, derfor blev opgaven ikke løst på egen hånd, og derfor fejler succeskriteriet.
Opgave 7	8	Opfyldt	Opgaven blev løst efter hensigten.
Opgave 8	21	Ikke opfyldt	Der blev at testpersonen forestaget 21 klik, før der blev opgivet, og måtte bede om hjælp, derfor blev opgaven ikke løst på egen hånd, og derfor fejler succeskriteriet.
Opgave 9	1	Opfyldt	Opgaven blev løst efter hensigten.

Table 18.2: Usabilitytest resultater

18.3.1 Problem 1

I usabilitytesten løste testpersonen ikke opgave 2 efter hensigten, se opgaverne i Bilag F. Dette kan tyde på to problemer: Enten er opgaven formuleret forkert, eller er den boks hvor endeligt semesternavn ikke tydelig nok. På Figur 18.1 ses den del af produktet, hvor problemet er.

18.3. Testresultater



The screenshot shows a web form titled "Opret Semester". It contains the following fields and controls:

- Semesternavn:** A text input field with a dropdown arrow on the right.
- Semesternummer:** A dropdown menu currently showing "1".
- År:** A dropdown menu currently showing "2018".
- Season:** Two radio buttons labeled "Forår" and "Efterår", both of which are unselected.
- Vælg Spørgeskema data:** A label above a text input field, preceded by a yellow folder icon.
- Gennemse:** A button next to the "Vælg Spørgeskema data" input field.
- Endeligt Semesternavn:** A large text area displaying "1" and "2018-".
- Opret:** A button at the bottom center of the form.

Figur 18.1: Opsætning af opret semester

18.3.2 Problem 2

Ved opgave 4 var det ikke tydeligt for testpersonen at sentimentanalysen er startside, når der vælges et semester. Dette kan dog skyldes at der ikke var noget data på det valgte semester, så testpersonen kom ind på en næsten tom side. Dog kan det også skyldes at det ikke tydeligt på siden fremgår hvorhenne brugeren befinder sig.

18.3.3 Problem 3

Testperson endte med at vælge et forkert semester end det opgaven var. Problemet kunne være at enten at knapperne med semesternavnene er for små, skriften er for lille, eller også kan det være fordi der er for meget information på en gang. Endeligt kunne det også være at oversigten kræver tilvænning, og at den bliver mere overskuelig, når semesterkoordinatoren selv laver semestrene en efter en, og derfor ved hvilke semestre der er at vælge i mellem.

18.3.4 Problem 4

I forbindelse med opgaverne, der skulle udføres på rådatasiden, findes to problemer omkring denne side.

18.4. Afrunding

Problem 4.1

Testpersonen skulle guides igennem Opgave 6 og efter mange klik Opgave 8, disse opgaver er derfor ikke løst. Fordi brugeren skulle guides, da viser det at siden rådata ikke er så intuitivt at benytte, som ny bruger.

Problem 4.2

Testpersonen synes heller ikke at meningen med rådatasiden, hvilket er forståeligt, da resultatet endte med at blive noget andet end hensigten. Dataet bliver ikke repræsenteret på en overskuelig måde. Rådatasiden er dog det eneste sted hvor en ny markering kan laves.

18.3.5 Problem 5

Endeligt nævnte testpersonen selv at et at der stod 'Tilbage' på en af knapperne synes at indikere en fortrydelseshandling, hvilket ikke var tilfældet.

18.4 Afrunding

Der er udført en usabilitytest som har vist problemer med brugergrænsefladen, men desværre blev kun én enkel test udført, hvilket ikke giver et dækkende overblik. Hvordan de beskrevne problemer kan løses, beskrives i Kapitel 19.

Del V

Refleksion

Kapitel 19

Redesign

På baggrund af usabilitytesten i Kapitel 18 blev der fundet tre problemer med programmets design. I dette kapitel opstilles mulige løsninger på disse problemer.

19.1 Løsninger

I dette afsnit vil mulige løsninger på de fem fundne problemer i usabilitytesten blive opstillet. Løsningerne bliver opstillet på samme måde som problemerne, således at løsning 1 er løsningen på problem 1, osv.

Løsning 1

For at være sikker på at boksen med 'Endeligt Semesternavn' ikke er for utydelig for brugeren, vil boksen blive fremhævet, således at den ikke lige så nemt falder i et med baggrunden. Det skal dog stadig ballanceres med at det stadig skal være tydeligt for brugeren, at denne ikke er en input-boks.

Løsning 2

For at tydeliggøre for brugeren, at sentimentanalysen er det første der præsenteres når et semester er valgt, kan der indsættes en overskrift inde på siden, hvor der står 'Sentimentanalyse'.

Alternativt kan det også vælges at lave en forside, således at brugeren aktivt skal foretage en handling for at se sentimentanalysen, udover at vælge semester.

Løsning 3

For at sikre at semesteroversigten ikke er årsagen til Problem 3, forstørres skriften og derved også felterne. Dermed sikres det at brugeren nemmere kan se de forskellige semestre i oversigten.

Løsning 4

Problem 4 omhandler rådatasiden i systemet, og dertil blev der fundet to problemer.

Løsning 4.1

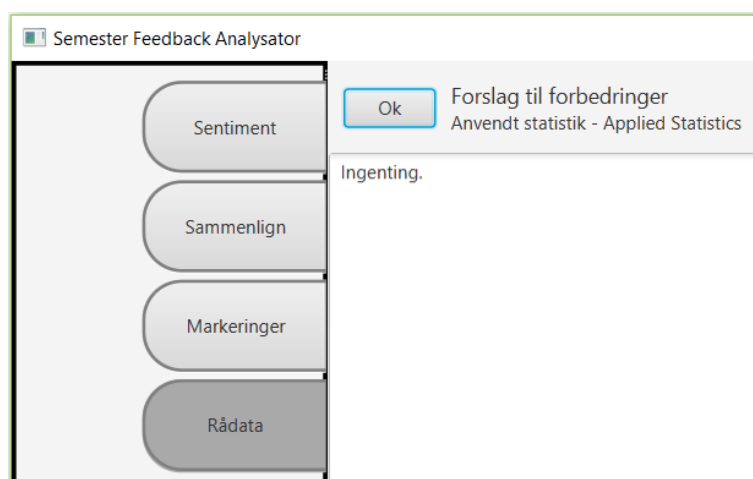
Denne løsning tager hånd om den manglende intuitivitet, omkring det at foretage en markering, samt filtrering af rådataet. For at løse dette laves en brugervejledning, se Bilag G. Denne nævnes allerede af testpersonen i prototypetesten i Kapitel 10, som et ønske. Dog er der grundet nogle nye designvalg, som bliver beskrevet i Kapitel 15, ønsket at teste intuitiviteten af disse først.

Løsning 4.2

Løsningen til den uoverskuelige præsentation af rådataet, ville være at designe det om, således at det kun er kommentarene der bliver vist, men med mulighed for at vælge at kunne se bedømmelserne også, via en filtrering. Desuden skal felterne med kommentarene være dynamiske, og udvide sig så den tilpasser sig størrelsen af kommentaren, både horisontalt og vertikalt.

Løsning 5

Testpersonen nævnte selv, at én af knapperne synes at have et misvisende navn i forhold til hvad den gjorde. Derfor, efter testpersonens ønske, er knappen 'Tilbage' ændret til 'Ok'. Hvordan denne ændring endte med at se ud, kan ses på Figur 19.1.



Figur 19.1: Ændring af navn på knap

19.2 Afrunding

Løsning 4.1 og 5 er løsninger der allerede er gennemført, siden det ikke krævede mange ressourcer i forhold til de andre løsninger.

Løsningen på Problem 3 ses på Figur 19.1.

Løsning 1, 2, 3, og 4.2, er mere krævende at implementere da disse kræver større ændringer, og bliver derfor prioriteret til videreudvikling.

Derimod kan det ikke garanteres, at disse løsninger er de mest optimale, derfor burde der foretages yderligere usabilitytests, for at sikre dette.

Videreudviklingen på de fundne problemer er blandt emnerne for diskussionen i det følgende kapitel.

Kapitel 20

Diskussion

I dette kapitel vil projektforløbet, og beslutningerne der blev taget deri, blive diskuteret og evalueret. Dette gøres for at trække linjer fra projektets tidligere faser, og frem til det endeligt udviklede produkt. Det vil dermed blive diskuteret hvordan programmets udviklingscyklus kunne have været forbedret, til hvilken grad kravene blev opnået, og hvordan programmet kan forbedres teknisk.

20.1 Kravspecifikation

I Kapitel 5 blev der udviklet et sæt af krav for programmets endelige funktionaliteter og kvaliteter. I dette afsnit vil disse krav blive gennemgået og reflekteret over, for at sammenligne det endelige produkt med det planlagte produkt. En gennemgang af alle krav, og deres status, kan ses på Tabel 20.1 og Tabel 20.2.

To krav blev ikke opnået. Det første var Krav 3: Kategorisering af lignende kommentarer. I starten af projektet blev det foreslået, at programmet skulle kunne kategorisere lignende kommentarer. Dette blev foreslået, for at kunne give et overblik over emner, der ofte blev refereret til. Funktionaliteten ville have øget programmets evne til at skabe overblik, og ville dermed styrke programmets evne til at opnå dets centrale formål.

Det andet krav som ikke blev opfyldt var Krav 5: Tilgå én studerendes besvarelser samlet. Dette krav var baseret på Semesterkordinatores arbejdsproces, som blev udforsket gennem interviews som kan ses i Bilag C. De havde nogen gange brug for at se besvarelserne som én enkel studerende havde lavet, og de besvarelser som de modtog var ikke sorteret på den måde. Dermed ville det have hjulpet programmets generelle evne til at analysere semesterevalueringer.

På grund af tidsrestriktioner, og at implementeringen af andre kernefunktionaliteter blev mere tidskrævende end ventet, resulterede det i at Krav 3 og 5 ikke blev implementeret.

20.1. Kravspecifikation

	Ikke Opnået	Delvist Opnået	Opnået
1. Indkapsling af relevante data			X
2. Adskille hold og data			X
3. Kategorisering af lignende kommentarer.	X		
4. Tilgå ubehandlede evalueringer			X
5. Tilgå én studerendes besvarelse	X		
6. Markering af kommentar			X
7. Sammenligning af årgange			X
8. Forståelse af studerendes mening ved brug af Sentimentanalyse			X
9. Genkendelse af Nøgleord			X
10. Fjernelse af tegnsætning			X
11. Fjernelse af stopord			X
12. Omdannelse til små bogstaver			X
13. Analyse af dansk tekst			X

Table 20.1: I kapitlet 5 blev der bestemt et sæt af krav for programmets funktioner. De følgende krav blev opnået.

Krav 14 blev ikke fuldt opnået. Selvom kravspecifikation kun bestemte, at der skulle være en brugergrænseflade til loginfunktionen, var det ideelt hvis brugeren kunne bruge deres AAU login til at få adgang til deres data. Grunden til at dette ikke blev implementeret, skyldes at et samarbejde og en teknisk integration med universitets system var nødvendig, som ikke kunne nås indenfor projektets tidsrammer.

20.2. Tilgang til projektudvikling

	Ikke Opnået	Delvist Opnået	Opnået
14. Loginskærm		X	
15. Valg af hold			X
16. Dansk grænseflade tekst			X
17. Upload af data			X
18. Markering af kommentarer			X
19. Se markerede kommentarer			X
20. Se enkel studerendes kommentarer			X
21. Præsentation af analyseret data			X
22. Tilføj hold			X

Table 20.2: I kapitlet 5 blev der derudover bestemt et sæt af krav for programmets brugergrænseflade. De følgende krav blev opnået.

20.2 Tilgang til projektudvikling

Tidligt i projektets forløb blev det bestemt hvordan programmets udvikling skulle foretages. Dette indebar blandt andet projektudviklingsmetoden, og hvordan data skulle indsamles. I dette afsnit vil der blive reflekteret over beslutningerne omkring metodevalg, og deres betydning for projektforløbet.

20.2.1 Metodevalg

Ved projektets begyndelse er de forskellige metoder til projektudvikling blevet beskrevet. Det blev besluttet at vandfaldsmetoden skulle bruges frem for den iterative.

Det antages, at ved at have valgt at anvende den iterative metode i stedet for vandfaldsmetoden, så ville det have krævet bedre initierende viden fra semesterets kurser end der har været muligt. At bruge vandfaldsmetoden gør, at projektet har kunnet følge kursernes gang, og har derfor givet en god dynamik til projektudviklingen.

Vandfaldsmetoden har været den rigtige metode dette semester, da der har været behov for den relevante viden fra kurser, inden det kunne anvendes. Med alt nødvendig baggrundsviden inden projektstart, ville den iterative metode have været en betydelig kandidat, da dette kunne have givet en bedre sammenhæng mellem projektdelene, da de alle ville være blevet arbejdet på i forbindelse med alle delopgaver.

20.2.2 Interview

Interviews har været en vigtig del af projektets indledende dataindsamlingsfase og bestemte den overordnede designtankegang. Det var derfor vigtigt, at denne del af programmets udvikling blev planlagt og udført optimalt.

Det semistrukturerede interview havde en positiv betydning for projektet. Med denne interviewmetode, haves der en generel idé om hvilke områder der skal diskuteres, og spørgsmålene er åbne. Derudover bliver nye spørgsmål mere dybdegående, og fokuserer på hvad der er relevant for interviewets formål. Denne interviewmetode var god, fordi projektets retning var delvist bestemt fra starten, og det semistrukturerede interview tillod at reflektere på om den retning var optimal, eller om det eventuelt skulle gentænkes. Interviewmetodens egenskab af at tillade længere svar som omhandlede emner udenfor det originale spørgsmål tillod også den interviewede at give projektgruppen idéer til funktionaliteten af programmet, som ikke nødvendigvis ville være opstået med et struktureret interview.

Et struktureret interview er opbygget med en fast form, og spørgsmålene er mere faste og lukkede. Dette gør det lettere, at få mindre isolerede svar, som let kan analyseres i forhold til andres svar på samme spørgsmål. Denne interviewmetode blev ikke valgt på grund af det lille antal informanter som vi havde adgang til, men hvis den var blevet valgt, ville de specifikke områder som vi indledningsvist havde dømt vigtige, blive uddybet mere end ved brug af det semistrukturerede eller ustruktureret interviewmetode.

Det ustrukturerede interview har ingen forudbestemt struktur og minder om en samtale. Denne interviewstruktur ville have været god til at få nye idéer til hvilke egenskaber programmet skulle have, eller andre potentielle områder som den interviewede ser problemer. Denne åbenhed ville ikke nødvendigvis bygge videre på projektets initierende tanke, men kunne derimod have været god som noget af det første der blev foretaget, for at finde reelle problemer der kan løses.

På trods af fordelene ved de andre interviewmetoder, vurderes den semistrukturerede stadig som den bedste i situationen, da der blev fundet en fyldestgørende mængde af information fra den, som både uddybede programmets retning, og gav inspiration og ideer til programmets udvikling og egenskaber.

Generelt ville en større gruppe af informanter have været brugbart, siden det ville give en mere generel forståelse af brugergruppen, og kunne eventuelt have vist problemer og inspireret løsninger, som der ellers ikke vil have været taget højde for.

20.3 Fremtidig Udvikling

Bestemte dele af det udviklede program har potentiale til videreudvikling eller forbedring. Derimod skaber begrænsningen på ressourcer og tid en udfordring i at prioritere hvilke områder der udvikles på, og i hvilket omfang.

I dette afsnit vil det blive diskuteret hvilke dele af programmet kunne udvikles videre på, og i hvilket omfang.

20.3.1 Optimering af Netværk

Neurale netværker er en kompleks abstraktion til at løse besværlige problemer som traditionelle regelbaserede algoritmer ikke er egnede til at løse. Vores implementering af LSTM-netværket kunne måske optimeres ved at teste flere konfigurationer i Afsnit 16.3, i form af word embeddings eller netværksstørrelser.

20.3.2 Videre Optimering af lineær algebra

I projektets forløb blev der udviklet et lineær algebra bibliotek for at implementere det neurale netværk. På trods af bibliotekets optimering, er der stadig måder at reducere udregningstiden for bestemte funktioner og metoder.

I forhold til matrixmultiplikation, findes der Strassens algoritme som har en lavere kompleksitet, og foretager mindre individuelle multiplikationer. I programmets matrixmultiplikation er der en asymptotisk kompleksitet på $O(n^3)$. Strassens algoritme har en asymptotisk kompleksitet på $O(n^{2.80})$. Betydningen af dette vil stige betydeligt som matrixdimensioner stiger. Problemet med algoritmen er dens implementation er meget tidskrævende, på grund af at flere dele af algoritmen kan skabe et stort overhead. Algoritmen er også udviklet til matricer som har $N \cdot N$, hvor N er en multiplum af 2. Til sammen betyder dette at algoritmens implementation kan variere betydeligt fra den oprindelige pseudokode, som ikke tager højde for de problemer. Dette er et problem fordi matrix multiplikation for projektets brug har ikke de samme forventninger. De matricer der arbejdes med har arbitrære størrelser. Derfor skal algoritmen ændres til at håndtere andre størrelser af matricer, som ofte gøres ved at fylde de resterende felter med 0, for at opnå kriteriet med en størrelse på $N \cdot N$, hvor N er en multiplum af 2. Der er derimod andre løsninger til dette problem, som alle skulle undersøges og sammenlignes for at finde den optimale løsning.

Der er også andre operationer, som transponering, hvor brug af andre, mere optimerede, datastrukturer vil kunne mindske beregningstid.

20.3.3 Videreudvikling af Grænseflade

Programmets grænseflade er evalueret med en usabilitytest af én semesterkoordinator. Derimod ville det hjælpe anvendeligheden af programmet, hvis det blev testet af flere semesterkoordinatorer, og at det blev anvendt i realistiske omgivelser. Dette ville kræve at en brugbar og fuldfunktionel version af programmet blev anvendt af semesterkoordinatorer, som et værktøj til udarbejdelse af semesterevalueringsrapporten, hvilket ikke har været muligt i projektperioden. Det ville også hjælpe at have flere semesterkordinatorer at arbejde med, så programmet kunne designes til den mere generelle gruppe af semesterkordinatorer, i stedet for kun få bestemte.

20.3.4 Andre Datasæt

Det anvendte datasæt er en blanding af data fra semesterevalueringer og online webshop-anmeldelser, for at øge kvantiteten. Fordelen ved større kvantitet i et datasæt er at indlæringen er mere generel, og lærer mønstre i stedet for at huske specifikke scenarier. Det kan derfor være svært at drage nøjagtige konklusioner omkring programmets evne til at klassificere sentiment på semesterevalueringer. Det ville derfor være at foretrække hvis der var adgang til et større datasæt af semesterevalueringer.

Kapitel 21

Konklusion

I dette kapitel vil det blive evalueret hvor vidt det udviklede produkt og resultaterne deraf, løser problemerne i den opstillede problemformulering.

Det initierende problem lød på, hvorvidt det er muligt at give et overskueligt overblik over semesterevalueringer, samt en bedre repræsentation af disse. Dette blev uddybet i projektets problemformulering, som specificerede hvilken retning projektet ville tage efter den initierende analyse, på baggrund af interviewene og PACT-analysen.

Problemformuleringen lyder som følgende:

Hvordan kan semesterkoordinatorerne på AAU hjælpes i semesterevalueringssprocessen med et databehandlingsværktøj?

Udover problemformuleringen, blev følgende underproblemer undersøgt:

- Hvordan kan et system der gør det nemmere at kunne analysere semesterevalueringsspørgeskemaerne modelleres?
- Hvordan kan software laves, der danner et overblik over generelle holdninger i tekstuel data?

For at besvare første underspørgsmål "*Hvordan kan et system der gør det nemmere at kunne analysere semesterevalueringsspørgeskemaerne modelleres?*" kræves der en forståelse af problemområdet, anvendelsesområdet og de potentielle brugere.

Den samlede analyse af problemområdet såvel som anvendelsesområdet, undersøger systemet, hvad det består af og hvem der bruger det. Her er Klasser, relationer imellem dem, adfærd, brugsmønstre, aktører og andre elementer defineret og analyseret. Et overblik over disse, gør at det er muligt at arbejde videre med områderne, og designe og implementere dem i udviklingen af systemet.

Designet af systemet og det endelige produkt, er en proces som har været gennem flere evalueringsiterationer. Evalueringen af prototypen dannede grundlag for at udvikle et brugergrænsefladedesign, og implementere det som en egentlig brugergrænseflade. Dette design blev efterfølgende evalueret endeligt via usabilitytestning. Det sidste redesign er derfor det endeligt anvendte design ved projektets afslutning.

Disse analyse- og designkapitler blev brugt til beskrive hvordan et system blev modelleret og designet, på en måde som er nem at navigere og forstå for brugere, samt som opfylder de specificerede krav.

For at besvare det andet underspørgsmål "*Hvordan kan software laves, der danner et overblik over generelle holdninger i tekstuel data?*" behøves der en teknisk implementation af et system, som kan finde mønstre, der kan blive generaliseret til at skabe overblik. I projektet blev det bestemt, hovedsageligt at fokusere på sentimentet for kommentarer.

To individuelle analysemetoder til at bestemme sentimentet i danske tekster, er udviklet med det formål at få bedre resultater, hvis metoderne supplerer hinanden.

Kvalitetssikring er en måde, hvorpå det er muligt at sikre, at de ting der implementeres, er af høj kvalitet. Her blev klassificeringsprocenten, kodetestning og anvendeligheden af produktet undersøgt.

Implementering- og kvalitetssikringskapitlerne viste, at de to sentimentmodeller, kombineret til et hybridsystem, resulterede i en model som kunne kategorisere kommentarer med deres tilhørende positive eller negative ladning, og at dette blev gjort med højere præcision end klassificeringsmodel enkeltvist.

Ud fra alle argumenter i dette kapitel, kan det afgøres, at et databehandlingsværktøj, som kan analysere semesterevalueringsspørgeskemaerne i afslutningen af et semester, har potentiale til at hjælpe semesterkoordinatorerne på AAU.

Bibliography

- [1] AAU. *Undervisnings- og semesterevaluering*. URL: <https://www.kvalitetssikring.aau.dk/kvalitet-undervisning/procedure-evaluering-undervisning-semester-uddannelser/undervisning-semester/>. 17. September, 2018.
- [2] Aalborg Universitet. *Studerende på Aalborg Universitet*. URL: <https://www.aau.dk/om-aau/aau-i-tal/studerende/>. 2017.
- [3] Aalborg Universitet. *AAU i tal 1974 - 2014*. URL: <https://www.aau.dk/om-aau/aau-i-tal/1974-2012/>. 2014.
- [4] Gyldendal. *Systemudvikling i Den Store Danske*. URL: <http://denstoredanske.dk/index.php?sideId=16854>. 19. September, 2018.
- [5] W. W. Royce. “Managing the Development of Large Software Systems”. In: *ICSE '87 Proceedings of the 9th international conference on Software Engineering*. Ed. by William E. Riddle. IEEE Computer Society Press Los Alamitos, CA, USA ©1987, Mar. 1987, pp. 328–338.
- [6] P. Nielsen J. Stage L. Mathiassen A. Munk-Madsen. *Object Oriented Analysis and Design*. Metodica, 2000.
- [7] Schwaber K. “Scrum and Agile 101”. In: *Extreme Programming and Agile Methods — XP/Agile Universe 2002*. Ed. by Wells D. & Williams L. Springer, Berlin, Heidelberg, 2002, pp. 72–73.
- [8] Ken Schwaber. “SCRUM Development Process”. In: *Business Object Design and Implementation*. Ed. by Jeff Sutherland et al. London: Springer London, 1997, pp. 120–125.
- [9] Jens Thisted. *Forsknings Metode i Praksis*. S. 183-185. Klareboderne 5, 1115 København K: Munksgaard Danmark, 2011.
- [10] Kaare Pedersen Poul Bitsch Olsen. *Problemorienteret Projektarbejde*. Samfundslitteratur, 2015.
- [11] David Benyon. *Designing Interactive Systems*. Pearson, 2014.
- [12] Sundhed.dk. *Gammelmandssyn (presbyopi)*. URL: <https://www.sundhed.dk/borger/patienthaandbogen/oejne/sygdomme/brydningsfejl/gammelmandssyn-presbyopi/>. 2018.

Bibliography

- [13] Susanne Klim Preben Thalund Madsen Stephen Biering-Sørensen Finna Overgaard Hansen. *Håndbog i Struktureret Programudvikling*. Teknisk Forlag, 1992.
- [14] Dana Chisnell Jeff Rubin. *Handbook of Usability Testing - Chapter 3*. URL:<https://www.moodle.aau.dk/pluginfile.php/1312826/course/section/398879/Rubin-cp-3.pdf>. 19. november, 2018.
- [15] Hans Hüttel. *De regulære udtryk*. URL:<http://www.hanshuttel.dk/wordpress/2012/12/06/helt-regulaert/>. 6. december 2012.
- [16] Balslev. *Normalisering af database*. URL: <https://balslev.io/programming/database/normalisering-af-databaser/>. 7. November, 2018.
- [17] Wikipedia. *tf-idf*. URL: <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>. 30. October, 2018.
- [18] Sarit Chakraborty Bijoyan Das. *An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation*. URL:<https://arxiv.org/ftp/arxiv/papers/1806/1806.06407.pdf>. 2018.
- [19] Peter T. Pham Dan Huang Andrew Y. Ng Christopher Potts Andrew L. Maas Raymond E. Daly. *Learning word vectors for sentiment analysis*. URL:<https://dl.acm.org/citation.cfm?id=2002491>. June 19 - 24, 2011.
- [20] Michael Nielsen. *Using neural nets to recognize handwritten digits*. URL:<http://neuralnetworksanddeeplearning.com/chap1.html>. 2018.
- [21] Colah. *Understanding LSTM Networks*. URL:<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. 2018.
- [22] Oracle. *ArrayList Documentation (Java 10)*. URL:<https://docs.oracle.com/javase/10/docs/api/java/util/ArrayList.html>. 2018.
- [23] Oracle. *FilteredList Documentation (JavaFX 11)*. URL:<https://openjfx.io/javadoc/11/javafx.base/javafx/collections/transformation/FilteredList.html>. 17. December, 2018.
- [24] Oracle. *ObservableList Documentation (JavaFX 11)*. URL:<https://openjfx.io/javadoc/11/javafx.base/javafx/collections/ObservableList.html>. 17. December, 2018.
- [25] Paul Martin. *Dispel Tutorial 0.8 Documentation*. URL:http://homepages.inf.ed.ac.uk/pmartin/tutorial/case_studies.html. 18. December, 2018.
- [26] Magnus Madsen. *Software Testing*. URL: https://www.moodle.aau.dk/pluginfile.php/1173653/mod_resource/content/1/Lecture%201a%20-%20Software%20Testing.pdf. 10. Februar, 2018.
- [27] Wikipedia. *Unit Testing*. URL: https://en.wikipedia.org/wiki/Unit_testing. 27. November, 2018.

Bibliography

- [28] Dana Chisnell Jeff Rubin. *Handbook of Usability Testing - Chapter 5*. URL:<https://www.moodle.aau.dk/pluginfile.php/1312826/course/section/398879/Rubin-cp-5.pdf>. 19. november, 2018.
- [29] Bertel Torp. *Dansk stopords liste / Danish stopwords*. URL:<https://gist.github.com/berteltorp/0cf8a0c7afea7f25ed754f24cfc2467b>. December, 2018.

Del VI

Bilag

Bilag A

Interview-spørgsmål

Hvilke(t) studie er du semesterkoordinator for?

Hvor længe har du været semesterkoordinator?

Hvad gør du ved de udfyldte semesterevalueringsspørgeskemaer?

Læser du alle svarene igennem?/Hvor grundigt læser du svarene igennem?

Hvor mange skemaer med skrevne svar får du for et gennemsnitligt semester?

Hvordan skaber du et overblik over svarene?

Hvor meget tid bruger du på en semesterevaluering?

Hvad vægter du højest i forhold til at skrive resume?

Hvad sker der med resuméet efter du har lavet det?

Hvem kan få adgang til denne data (svarene på spørgeskemaerne og det resume du genererer)?

Hvad/Hvordan bliver resuméet benyttet til forbedring af semestret?

Læser i "gamle" resuméer igennem for at sammenligne svar?/for at se om der er en udvikling?

Hvad ville du ønske et system skulle kunne, for at hjælpe dig med denne proces?

Bilag B

Anvendelse af de 7 trin

De 7 trin blev introduceret i Delafsnit 2.2.5, hvilke bruges til at beskrive interview-processen,

1. Tematisering

Til at lave projektet er der i gruppen 7 studerende, som har til opgave at lave et projekt på baggrund af en eller flere informanter. Ud fra denne viden, er der blevet udvalgt at interview (.....), da de har en arbejdsstilling, som gør at de bearbejder semesterevalueringerne i slutningen af semestret. Formålet med interviewet, er at få en dybere forståelse for semesterkoordinatorernes arbejdsopgaver i projektets sammenhæng, så det kan tydeliggøres, hvilke problemstillinger, der kunne være relevante at arbejde med.

2. Design

Som studerende, arbejdes der løbende med semesterkoordinatoren på studiet, og har også selv udfyldt semesterevalueringer. Dette grundlag danner en forståelse for det overordnede arbejde med kvalitetssikring, som den enkelte semesterkoordinator udfører.

Informanterne har, som udgangspunkt, ikke viden om de enkelte spørgsmål, som der vil kommes ind på i løbet af interviewet, men kender derimod gruppens motivation for interviewet, og delvist det emne der arbejdes med.

Der er ikke udformet et fast sæt af spørgsmål, som forbliver de eneste spørgsmål. Interviewet vil være mere udviklende på baggrund af den information, der gives i løbet af interviewet. Dette gør, at der kan tilføjes flere dybdegående spørgsmål, men at der også kan være spørgsmål, som mister sin relevans.

3. Interview

Der var 3 tilstede for at føre interviewet. Én person som referant, og to som førte interviewet.

Interviewpersonerne blev informeret om betingelserne for interviewet, som var at lyden ville blive optaget til brug som referencepunkt for projektet, og at det ønskedes at referere til dem ved navn, i rapporten.

Interviewpersonerne blev også informeret om, at meningen med interviewet var at forstå arbejdsprocessen, samt at den samlede information ville blive brugt til at udvikle et system, som har til hensigt at optimere arbejdsprocessen.

De blev også motiveret til at ikke føle sig begrænset af spørgsmålenes struktur, og at længere svar var foretrukket, selv hvis de ikke er omfattet af spørgsmålet. Dermed blev interviewet mere fleksibelt i dens struktur.

Den interviewede person blev informeret om hvilke emner der vil blive spurgt om, og i hvilken sammenhæng. Nogle af de centrale spørgsmål var følgende:

- Hvilke studier er de eller har været ansvarlige for.
- Hvor mange studerende har de haft samlet på et semester.
- Hvad mener de, at modtageren skal have ud af at læse semesterevalueringsrapporten?
- Specifikke egenskaber af hvad den nuværende arbejdsmodel er for at gå igennem evalueringerne.
- Hvilken rolle spiller tidligere semesterevalueringer?
- Om nogle af de programegenskaber der kunne udvikles i projektet, ville gøre deres arbejde mere effektivt.

Derefter blev interviewet afsluttet med en de-briefing, hvor der blev gennemgået hvilke informationer der blev fået, hvordan den information er blevet tydet, og hvad der gøres efter interviewet. Dette gør at alle partier forstår hindanden klart, og kan ende interviewet.

4. Udskrivning

Lydsporet af interviewet blev optaget, og på baggrund af dette blev det hele transskriberet. Transskriberingen tillader at det nemmere kan analyseres og derudover kan der trækkes specifikke citater eller argumenter ud af det.

Der blev også taget notater under interviewet, som giver mulighed for lettere at bearbejde den information som blev givet.

5. Analyse

Formålet med interviewet er forstå semesterkoordinatorens arbejdsproces og mulige mangler som de selv oplever. Eftersom interviewet er semi-struktureret, kan det naturligt opdeles i kategorier baseret på de forskellige spørgsmål. Det er mere interessant at få dannet en meningkondensering over de mest centrale ting der bliver nævnt i interviewet.

6. Bekræftelse

7. Rapportering

For at kunne bruge interviewet skal det analyseres som beskrevet tidligere. Denne analyse beskrives i PACT i Kapitel 3.

Bilag C

Transkriberede interviews af informanter

Informant 1

SS: Hvilke studier er du semesterkoordinant for dette semester?

Informant: For dette semester?

SS: Ja

Informant: Jeg er Semesterkoordinator for 1. studieår, dvs. 1. semester og 2. semester for dem man kalder KMB. KMB det står for Kemi, Kemiteknologi, så står det for bioteknologi og biologi, og M'et i midten står for Miljøteknologi. Det er 5 fagligheder, hvor der faktisk også er en diplomingeniørretning (1.47), der er egentlig 6 indgange til den, og når de så kommer ind, så blander vi kemikerne og kemitekkerne, ... (2:03- 2:07) projekter, og vi blander ... (2:08-2:18) så vi ender op med 4 fagligheder (2:21-2:29)

Så er jeg praktikkoordinator det er på diplomingeniør, og de er på praktik på egentlig på 6. semester, men det kører henover sommerferie, så de er ikke færdig endnu. (2:48 – 2:58)

NZ: Hvor mange elever er du så semesterkoordinator for, sådan ca.?

Informant: Der er 170 på 1. semester, så er der en 12 på diplomingeniør på 6. semester (3:13-3:16) , og så er der 20 på ... (3:20-3:22)

SS: Hvor mange år har du været semesterkoordinator i større eller mindre omfang?

Informant: Det har jeg været en del år, men det er på 4. år jeg er semesterkoordi-

nator for 1. studieår.

Semesterkoordinatorer, de fleste har et semester, men på 1. studieår har man et helt år. Det 1. og 2. semester, så det er en større "chunk" (4:00-4:08) kommer i gang.

SS: Du er med på hvad vi mener når vi siger den her semesterevaluationsrapport, som du skal skrive? Hvad mener du man skal have ud af at læse den, som studerende eller udefrakommende, den offentliggjorte version, eller hvilket formål den har?

Informant: Hvilket formål den har? Formålet det er at den er en del af et kvalitetssikringssystem. Som en del der ligesom dokumenterer at vi sikrer kvaliteten og vi arbejder med at løbende at forbedre vores kvalitet, derfor har vi styringsgruppemøde, derfor laver vi de her skemaer, og vi forsøger at samle op på det, så vi kan sige: Det gik da meget godt, eller at der er noget her vi skal arbejde videre med.

For mig handler det om når jeg skal formidle det her, så tænker jeg ikke på at den må gerne ligge offentligt, det er jeg sådan set ligeglad med. Men der hvor jeg har brug for at formidle, det er til studienævnet. Som får at vide hvis skoen trykker et eller andet sted, på et eller andet kursus, og så er det den helt samlet feedback til de undervisere der er på semesteret, som så får at vide "Det der er fint nok, fortsæt med det, der er måske småting der kan forbedres" og nogle andre hvor "Her skal vi altså kigge på noget"

Jeg ser det meget som et intern kvalitetssikringssystem, som kort og præcist får givet et helikopterview på hvordan det går.

SS: Så de ting den rapport der bliver offentliggjort for studienævnet hjemmeside, er det det samme som underviserne ser? (6:24-6:30)

Informant: Jeg har ikke været inde og se hvad studienævnet offentliggør, så langt følger jeg ikke processen (6:34-6:37)

Den måde jeg kører det på er at gennem semesteret har jeg styringsgruppemøder, og der kommer referat, og den bliver sendt ud til alle underviserne, så har du mulighed for at gribe nogle problematikker med det samme. Og så har vi den her surveyxact, hvor de studerende laver noget, det kommer så senere, jeg har lige fået den sidste nu her fra forårssemesteret.

Når jeg så har den + den fra semesterstyringsgruppemøder, så kan jeg lave rapporten, som er den samlede (7:15-7:17)

SS: Vi har været inde og kigge på det, det var i forhold til kemi du kaldte det et eller andet, men de kemiretninger (KMB), det der bliver offentliggjort hos dem, det

er den del af surveyxact, det er de multiple choice spørgsmål, den procent der er svaret på det, det er det eneste der bliver offentliggjort ved dem.

Informant: Jeg laver jo en rapport (7:43-7:54)

SS: Vi skulle være sikker på du udarbejdede en rapport, udover den der bliver offentliggjort.

Informant: Når jeg laver en rapport, så ligger jeg så bilag, surveyxact, og styringsgruppemøde referat. Jeg tror ikke det er noget der kan offentliggøres som sådan, det er jo persondata. Og der er jo navn på den der er i styringsgruppemøde, så det ligger allerede udover at man må offentliggøre det. styringsgruppedereferaterne og surveyxactskemaerne, det er det jeg koger sammen til en rapport, som jeg egentlig mønster på studienævnet, så studienævnet får den her idé om kører det her eller kører det ikke, og har vi styr på det. Og så hvis der er noget som vi ligesom skal arbejde på, så er det også markeret deri. Og kommer så også til underviserne så de ligesom kan (8:55-9:05)

Der er ikke ret mange der har tid til at læse vores egen selvevalueringsrapporter, de skal bare koges ned, så de er fuldstændig skarpt. Fordi vi kan skrive op og ned af stolper (9:21-9:23) det bliver kvaliteten ikke større af.

SS: Når du udarbejder den her semesterevalueringsrapport, kan du prøve at forklare lidt med den der proces med henblik på de der surveyxactskemaer, hvordan du bearbejder de data og skriver rapporten.

Informant: Hvad er kvaliteten af dataene? For det første så er der, man kan se på svarprocent, hvor mange har gidet at gå ind og svare på dette her? Og der har jeg det egentlig sådan at hvis der er mange der ikke gider gå ind og lave den her, så er det nok fordi (10:18-10:21) (At skoen ikke trykker særlig meget?) Så har de ikke så meget at brokke sig over, og så er der dem som har følt det (10:29-10:35)

Jeg har erfaring med at hvis der er noget som er virkelig rusk ravende galt så skal folk nok komme når der er noget de kan brokke sig på, når de får en platform.. (10:49-10:55)

For mig ser jeg det ikke som dårlig kvalitet at der er mange der ikke har gidet at lave den her skema, det betyder de har ikke noget de skal af med, og så kigger jeg på deres procent der er nogle kategorier:

- Meget godt
- Godt
- Middel

- Dårligt
- Meget dårligt

Og så kigger jeg på hvad de angiver som deres egen indsats, det gik dårligt og deres egen indsat var dårlig, så er det ligesom at få en vurdering af det. Men jeg synes det (12:01-12:13)

Så er der noget der kommer over i godt, og så meget godt, og så er der et par stykker, nogle enkelte der kommer over i og der synes det gik dårligt.

Og så er der andre kurser, (12:22-12:34)

Det er sådan en bel kurve, og den kan sådan flytte sig, og det er den vi kigger efter, hvor ligger den henne? (12:38-12:51)

Jeg kigger på holdet som helhed, og så også deres aktivitet, det kan jeg ikke bruge til så meget længere. (13:03-13:08) Så er deres individuelle kommentarer og jeg sidder ikke minutiøst og læser dem og noterer hvad og ligesom prøver at finde ud af hvad det handler om, der er jeg mere sådan (13:30-13:35) har vi noget vi kan kritisere?

Hvis du har et kursus på et semester der bare ikke fungerer, så alle andre kurser perfekte. Så når vi får rettet op på det her kursus, så det begynder at blive rigtig godt, så begynder de at finde småting ved de andre kurser, men det har ikke været fremme i top 3 af hvad der var at brokke sig over før, så det der kommer kommentarer og der er nogen der siger "(14:15-14:20)" mange af kommentarerne de bliver ikke så spændende, man skal lige have det der filter med, hvilken kontekst der er for det de svarer på.

Og så kan jeg have nogle andre kurser hvor bel kurven lige er rykket ned til sådan mellem-tilfreds (14:47 -14:48) der vil jeg nok bruge lidt mere tid på at se hvad det så er de påpeger der.

Er der et kursus de ikke forstår de skal have, mangler motivationen, f.eks. har jeg haft et kursus hvor de studerende får noget, i kender godt de der fag der er lidt mere blødt end det man egentlig kom ind for at læse, hvor den studerende har spurgt hvorfor skal vi have det? (15:16-15:28) Motivere for hvorfor man skal have det

SS: Du siger det afhænger af hvordan fordelingen af den her kurve var, hvor minutiøst du læser den enkelte, kommer du igennem alle punkter alle kommentarerne som de ligesom giver? Den enkelte elev?

Informant: Jojo, amen det er ikke fordi så sidder jeg amen så gider jeg ikke læse kommentarerne, jeg skimmer kommentare igennem, jeg kan hurtigt se hvis der er

nogen der kommer med noget negativt, der er også nogen der skriver, dette her var fantastisk. Nogen der siger, amen jeg synes det var træls der var nogen der stillede spørgsmål i timerne. Så har man noget hvor man siger, det kan jeg ikke rigtigt rette til. (16:27-16:37)

Jeg er en af dem der bliver evalueret, og så vil der være der synes at laboratorieøvelser var (16:48-16:57)

Her kommer nogen der peger i den ene retning og nogle peger i den anden retning, men den store majoritet har egentlig haft det (17:06-17:10)

Det er lidt sådan, det er lidt svært at sætte på tal, men er nødt til at se er der noget der popper op, hvis der er 10 som synes (17:30-17:34) men hvis der er 1 så er det måske for lidt, altså en ud af 170, der har gidet at kommentere (17:40-17:48) Det er lidt at få samlet op på hvor mange der ligesom peger i samme retning. Det har jeg så et andet kursus

SS: du nævnte noget med deres egen indsats som fremmøde, at det vidste du ikke rigtig hvad du skulle bruge det til, vi forestillede os/tænke det var en mulighed at man kunne vægte deres evalueringer på en eller anden måde, i forhold til hvor meget de selv mente de gjorde ud af undervisningen og hvor meget de dukkede op til undervisningen. "Jeg synes det var fantastisk godt, men jeg var der 1 gang" hvad har jeg egentligt at sige om øh

Informant: Vi får ikke den kobling, Jeg mener ikke vi får den kobling i skemaene, den findes selvfølgelig i dataene, men den er ikke i skemaet. Jeg tror måske i skal tænke på at nogen af disse skemaer, de bliver jo også, nogle der har tænkt at der er en anden funktion med at spørge hvad deres egen indsats er. Når du skal sidde og lave skemaer, og du har været til eksamen, "fandme lorte kursus" – og så skal du til at lave evalueringer, hvor den spørger: "Hvad er din egen indsats?". "Upps." tænker man så. Så tænker man lige over hvad du egentlig gjorde selv. Så ser man sin evaluering i en anden kontekst. Man ser ikke bare at det er et skodkursus, men ser også at ens indsats måske var ringe. Den del er også for at sørge for at dem der går ind og svarer på disse skemaer, er i stand til at sætte det i eget perspektiv, ud fra dem selv, og ikke bare ud fra hvad de har fået serveret. Det skal i jo tænke på med de her skemaer, det er ikke fordi vi er specielt interesserede i deres egen indsats, men vi vil gerne have at de har øjnene op for hvor meget de egentlig gjorde selv, når de kommer ind og kritiserer alt. Fordi der er jo ikke nogen selvevalueringsskema med.

SS: En af de andre ting som vi havde snakket om, nu ligger de tidligere semesterevalueringsskemaer de ligger på nettet, er det noget du kigger på? Kigger du på udviklingen af et semester hvis du for eksempel har første semester flere år i træk. – For eksempel hvis der kommer et udslag?

Informant: Jeg tror ikke at jeg går ind og kigger på data. Det er jo også i forhold til at det her for mig ikke er super kvantitativ data, det er lidt ulne data. Det er ikke noget hvor jeg synes vi har et hårdt gennemsnit på 1.7 af hvor godt det er gået. Det er lidt mere med "gefúl". Man har forskellige hold, og holdstørrelser, men det jeg jo gør det er at jeg kan jo godt huske hvordan det var sidst. Der var også det her kursus som de studerende smider alt deres kritik over på. Og så kigger jeg på om vi så har forbedret det, og samtidig med at de andre stadig holder det niveau de skal. – men dataet går jeg ikke tilbage og kigger, og det har noget at gøre med min fornemmelse for dataets kvalitet. Jeg tror studienævnet laver en score ift. Om det er godt, meget godt, mindre godt osv. og så kan de lave et gennemsnit på baggrund af det. Og det kunne jeg da også godt lave, men hvis det er 1.7, 1.9, hvordan skal jeg så forholde mig til det. Hvad er standardafvigelsen på denne måling? Der skal jo næsten bare være 1 der syntes det var træls fordi han ikke evnede at komme i gang, til at forskyde denne score. Så har jeg svært ved at bruge denne data.

NZ: Når du sådan sidder og kigger på de der kommentarer folk har skrevet, tager du så noter undervejs, eller hvordan gør du det? Sidder du bare og læser?

Informant: Nej jeg kunne måske godt finde på at læse den på min ipad, og så sidder jeg lige og noterer når der er en pointe man godt kunne have med, men nej jeg sidder ikke og laver noter. Du har spurgt nogen om de har noget at brokke sig over, og så siger de, "Ja jeg kan lige komme på noget". Så får du afvide at der en person syntes forelæsninger var kedelige og resten syntes det var fint. Skal jeg så til at gøre det for denne ene person? Derfor handler det meget om for mig at komme op i helikopter view og få set det overordnet. Hvordan er den gennemsnitlige fornemmelse for de her kurser.

NZ: Så du foretrækker et mere generelt overview end den enkelte kommentar.

Informant: Det er fint med den enkelte kommentar. En ting er vores kvalitetssikringssystem, men en anden er have tillid til at vi hører på dem når de har noget at sige. De skal have et sted hvor de også kan læse af.

NZ: I det overblik du gerne vil have, hvor du ser det som en helikopter?

Informant: Når det er individuelle kommentarer, så ved jeg ikke om det er svært at samle det ligesom i kategorier.

SS: Ville det hjælpe dig hvis du havde en grov fordeling af dem, i stedet for at have dem i en lang række. Hvis du kunne gå ind og sige at her er fint nok, og det her er ikke så godt.

Informant: Det er de vel allerede er de ikke?

SS: Jo, de er givet på baggrund af et bestemt spørgsmål, men under det spørgsmål kan der være positive og negative kommentarer, der kan være forslag hvis man på en eller anden måde kunne få dem fordelt ville det så gøre dit arbejde nemmere?

Informant: Det ved jeg ikke, for mig at se handler det også om hvad mængden af data jeg har og skulle sortere i? Jeg har 170 elever på første semester. Der har jeg så sådan at når de starter op på anden semester, så tvinger jeg dem til at lave den der surveyExact så jeg får næsten 170 besvarelser af data at sortere i. Men på de andre semestre, f.eks. når de er færdige med anden semester er de ikke blevet tvunget til at udfylde dem, der har jeg 130 elever hvor så halvdelen har svaret, så det er ikke fordi der er den vilde mængde data i det, eller jeg ved ikke, fordi jeg ved ikke om når datamængden er så forholdsvis begrænset om det giver mening at sortere i det. Har i nogensinde set nogen besvarelser (Rapporten med procenter)

NZ: Ja

Informant: Besvarelsesprocenten i starten er høj og så falder de af over tid. Det er heller ikke alle der kommenterer.

SS: Hvis man forestiller sig at det var en perfekt verden, og du havde alle ressourcer som du kunne ønske dig til at udarbejde denne rapport, hvad ville du så ændre (Mere tid eller andre værktøjer til at gøre den bedre).

Informant: Jeg ved det ikke, jeg synes noget af det der er svært er selve spørgeskemaet, hvad er det man spørger om og hvad får man svar på? Får vi det rigtige svar ind. Jeg synes næsten det er der jeg har mest svært ved det. Jeg har brug for at komme lidt på afstand af det, og se det som et samlet hele. Jeg kan ikke bruge at der en der brokker sig over at tekstbogen er på engelsk, det er en rammebetingelse. Jeg tænker mere på om spørger om det rigtige, og om man får de rigtige svar i forhold til hvad vi kan gøre, altså forbedringsmæssigt. Det med databehandling, det er det i lidt fisker efter ikke?

SS: Jeg kan godt se at der isoleret set simpelthen kommer for få datapunkter til at lave kategoriseringer af det, så det kan hjælpe dig, eller et eller andet sted ikke er nødvendigt.

Informant: Nu har man de her: Meget godt, godt, mellem osv. Du kan angive din egen indsats og så kan du kommentere. Det er sådan nogle ting der kan komme i sådan nogle rapporter. Så skal man måske arbejde på at styre disse kategorier.

SS: Så spørgsmålene?

Informant: Jamen så får man også bare svarene derefter. Noget af det jeg godt kunne tænke mig er at vide hvordan de studerende studerer. Altså vi har et typisk kursus er læs lærerbogen, se forelæsning, regn opgaver, tag eksamen. Men der er jo andre ressourcer, gå på nettet, se youtube videoer, man kan finde alternative bøger, måske bruger man mere gruppen til at diskutere og lære det. Hvad er det for en platform de lærer på. Det tror jeg kunne være interessant for underviseren at finde ud af, er det tekstbogen eller opgaveregningen der er godt, eller noget helt andet? Det kan jeg ikke læse ud af skemaerne, det er mere gik det godt og har du noget at brokke dig over. Jeg kan ikke rigtig se udviklingen af undervisningsformen, det har jeg lidt svært ved at bruge den.

SS: Hvor meget frihed har du til at ændre spørgeskemaerne.

Informant: Nej det har jeg ikke noget at gøre med? Jeg får dem bare når du er blevet udfyldt. Det er noget som kommer fælles oppefra, hvor man har sagt at det her er en fælles skabelon for de her skemaer, og hvad de skal indeholde. Jeg synes det med at lave de spørgeskemaer så det giver de informationer som vi gerne vil have er lidt af en kunst ikke? Det vi har nu er jo der hvor man er lige nu, og det er et kompromis af mange ting. Man kunne måske godt spørge om meget mere, men så er der mange flere studerende der falder fra når de forsøger at svare på det hele. Derfor forsøger man at gøre det kort.

Informant: En anden ting som jeg kunne interesse mig for om der er en korrelation mellem folks egen indsats og hvor godt de har fundet kurset. Det ville jo være forholdsvis nemt at tage de data og lave en graf og se hvordan punkterne de lå.

SS: Nu har vi snakket om hvordan du brugte tidligere evalueringer, hvis man igen kigger på korrelationer af dataet, du går ikke selv ind og kigger og taster det ind, men hvis du havde mulighed for at have et system hvor du kunne gå ind og se korrelationerne af tidligere semestres evalueringer, så ville man jo kunne gå ind og se udviklingen fra år til år, ville det være noget der var relevant for dig, f.eks. til denne rapport til studienævnet, ville det hjælpe dig hvis du 2 semestre fremme kunne se, ”hmm, det er da egentlig blevet bedre det her, på baggrund af de ændringer der blev lavet?”

Informant: Jo det kunne da være fint, men i skal også huske at hvis vi på 8. semester har 20 studerende og halvdelen svarer så har vi 10 studerende. Det er de små tals magi der giver de her udsving. Så zig-zagger det op og ned, hvad kan vi gøre når vi har så lidt data? I skal jo regne med at de fleste som sidder på min side og skal benytte de her data, ved jo godt hvordan man arbejder med data, og hvornår man kan sige noget er signifikant, så kan vi jo godt se at det godt kan være at gennemsnittet er steget eller faldet, eller det ser ud til at den rater dårligere, men

signifikansen er der jo ikke, vi har fået for få besvarelser og det er ikke alle der har besvaret. Der vil jeg mene at problemet bliver at datakvaliteten måske bliver en stopklods for at udnytte dataen mere. Jeg synes det kunne være meget spændende at udnytte dataet mere. Jeg underviser et kursus på første semester hvor jeg har et problem med at der er mange der dumper. Jeg hopper og danser og laver alt muligt for dem, for at de lærer deres ting, så jeg er meget interesseret i at se om der er noget der virker. Men igen kan det godt være de små tals magi der får det til at se ud som om at det virker, eller ikke virker.

SS: Vi har skrevet nogle ting ned omkring vores ideer til det system vi ville lave, og hvordan det ville hjælpe dig. Det er allerede nogle af de ting vi har været inde på, men nogle af de ting man kunne gøre hvis man måske forestillede sig at man havde et hold på 170 elever og de alle sammen svarede og kommenterede på spørgeskemaet, ville det så give mening for dig at have et værktøj der måske kunne udvælge de mest sigende kommentarer, spørgsmålet giver mere mening hvis du føler dig overvældet af kommentarer, men så havde et værktøj der kunne gå ind og plukke de kommentarer som giver mest mening for dig at handle på, og resten af dem der siger det er fint skal du så kun kigge på 1 gang.

NZ: Så ligesom en gruppering af lignende feedback, så folk der siger det samme bliver samlet.

Informant: Det i tænker er hvis folk har formuleret en kommentar med samme betydning på 5 forskellige måder, at i så kan samle dem?

SS: Ja det ville jo være håbet. Eller finde kommentarer der især stikker ud, eller måske har et forslag? – At det faktisk er konstruktiv omkring kurset, i stedet for bare kritik, så man kan få et overblik over dem, i stedet for at man læser dem igennem og de går i med mængden.

Informant: Der er det igen sådan at med den træning jeg har med at kigge på tal, så vil jeg sige, hvad ligger bag ved den algoritme som i bruger til dette. Hvor mange besvarelser ligger der til baggrund for den? For mig skal man passe på med at reagere på hvad som helst, som der bliver sagt. Det her system er begyndt at reagere mere og mere på enkeltudsagn, og det synes jeg det bliver at overreagere. F.eks. er der nogle der er i gang med at undersøge hvordan man kan understøtte studerende der har det dårligt med at dumpe eksamen, og det er baseret på en elev der har nævnt det til et studienævnsmøde. Nu har man sat en masse arbejde i gang for at undersøge dette. For mig er det for meget at gøre dette på baggrund af en person der lige er kommet med et udsagn. Man kan reagere i alle mulige retninger, og ikke have styr på noget som helst, og bruge en masse kræfter på at lukke nogle huller som egentlig ikke er særlig store.

SS: Hvad så hvis man forestiller sig at der komme de enkelte kommentarer som du snakker om. Hvis der så også var en der sagde dette sidste semester, og forrige semester. Ville det så give mening at samle de gengangere der er fra tidligere. Måske feedback fra tidligere semestre. Okay der er kun en der synes det men der har konsekvent været en der har sagt det de sidste 4 semestre, og der så danner sig et mønster over en længere periode.

Informant: Jeg tænker lidt på om man kan lave noget med, sådan nogle system hvor man kan skrive kommentarer ind, og så kan andre stemme hvor enige de er i disse kommentarer. I skal også tænke på når man sidder med sådan et skema der og skal kommentere på noget, "Jamen jeg kan lige skrive noget om noget jeg synes" men det kan også være at man synes noget andet, men det skriver man ikke fordi nu har man udfyldt denne boks. Men hvis man havde noget med at man lavede et udpluk af kommentarer fra sidste semester, som man skulle tilkendegive sin enighed på.

SS: Som forlængelse af surveyexact skemaet eller hvad tænker du på?

Informant: Hvis man tager kommentarer og andre siger om de er enige eller uenige med det. De gode data ligger i det hvor de rater det. Så har man kommentarerne hvilket er lidt et spredt felt, både i forhold til hvem der har gidet at kommentere noget, og ift. Hvad de kommenterer på. Det er jo ikke sådan at der er nogle der siger "Jeg har lige 10 feedback-punkter", nej de skriver en ting og så går man videre. Der er fire kolonner, men det er ikke sådan at der kommer en hel roman, alle har maks en kommentar. Det kunne man måske komme ud over hvis de skulle sige: Er du enig i at dette kursus har en god forelæser.

SS: Og det er så noget andre har sagt tidligere?

Informant: Ja. Som jeg ser det gør datakvaliteten det problematisk at lave en masse ud af databehandling som det er nu. Det ikke for at fjerne grundlaget for jeres projekt, men i skal være opmærksomme på hvad det er for en kvalitet af data i får ind.

SS: Du ville tilføje muligheden for at stille sig enig i nogle punkter som andre har kommenteret tidligere efter du skriver en kommentar.

Informant: Ja sådan noget, det er jo også svært det der med at man får serveret en mening, hvis man ikke lige selv havde tænkt på den. Vi ender op med et system hvor der kun er en kommentar pr. elev for de 4 kolonner, og det giver jo ikke et fuldt indblik i det. Jeg får ikke så meget ud af survey-exact skemaerne, som jeg gør med styringsgruppemøderne, hvor jeg kan have en dialog - Hvor de supplerer hinanden. Det er mere et supplement for mig, det er ikke spørgeskemaerne der er bærende for min rapport, og man kan sige at de kommentarer der til kurset, er også gennemgående på styringsgruppemøderne.

NZ: Så der kommer ikke så meget nyt i spørgeskemaerne?

Informant: Nej, jeg synes bare de bekræfter det jeg allerede ved, og så kvantificerer de det lidt, uden at jeg synes det er særlig kvantitativ alligevel. Jeg ser det f.eks. som positivt hvis ikke de gider at svare på skemaet. Det er selvfølgelig også godt hvis der er mange der ligger oppe i godt/meget godt i spørgeskemaet, og så kan der være en enkelt kommentar der gør at jeg reagerer på noget. Der handler det også om at semesterevalueringen er noget som jeg serverer opad i systemet så de kan se at der er styr på semestret.

Informant: Hvis jeg forstår jeres projekt rigtigt, så nytter det jo ikke noget at komme med nogle fancy værktøjer til at gruppere noget, hvis dem der skal bruge det siger, ”jeg har 15 elever på det her hold, og der er syv der har svaret, så giver det ingen mening for mig”. Datakvaliteten er ikke særlig god, derfor bliver man nød til som koordinator at forsøge at lave sig et overblik, og hvis der er mange sure udsagn for et kursus, så skal det selvfølgelig være noget der skal med, men man er nød til at se det på afstand.

Informant: I bliver nød til at tage hensyn til dataet. Det er lidt sådan, man føler lidt på det, og bruger dette.

Informant 2

Niki: Hej, øhm, vi vil gerne prøve at forstå din process, og hvad du går igennem, hvad du tænker og synes om det her. Det er semistruktureret så du må gerne bare slå dig fri sådan i svarene, det er vi ikke bange for... yes. Først vil vi gerne have lidt information om din rolle som semesterkoordinator, sådan hvor længe du har været semester koordinator?

Informant: På et bestemt semester, eller?

Niki: Bare sådan, altså, hvor lang tid du har været det, generelt, altså...

Informant: Jeg har været ansat i mange år, så jeg har været semester koordinator på mange forskellige semestre. De sidste år, har jeg været semester koordinator på datalogi et, og datalogi to. Og software et og software to, det vil sige det første studieår på datalogi og software.

Niki: hvor mange år i streg har du gjort det?

Informant: Det har jeg formodentligt gjort i seks eller syv år

Niki: Seks eller syv år? Det er også en god lang tid

Informant: Ja.

Niki: Hvor mange studerende har du sådan samlet set haft på et semester?

Informant: Nok ikke, jamen det ligger under det vi har i år som er det højeste. Øh, og hvad har vi, vi har omkring 200 og en 40-50 stykker på datalogi og software 1. Så det er vel mellem sådan skal vi sige mellem 100 og 250 i runde tal.

Niki: Jaer. Okay. Yes, sådan rent semester evaluering. Hva, Hvad mener du, eller, burde være formålet med de studerende, de udarbejder de her semester evalueringer?

Informant: Jamen det er jo et, et led i den kvalitets kontrol der er med underviserne, og indgår i den, en, vi kan sige en cyclus der hedder, der hedder planlægning, afvikling og evaluering. Og evalueringen fra et, fra, fra et forløb indgår så som en brik i planlægningen med den næste.

Niki: mmm

Informant: det er jo den store ide, med, med semester evalueringen. Og hvad var det så mere specifikt du spurgte om?

Niki: Det var mere altså hvad du synes, altså hvad, hva ønsker du at få ud af de her, altså hvad synes du er formålet med at i laver dem, hva' er det du synes altså skal komme ud i sidste ende?

Informant: Jo men det er vel den endelige konklusion, altså. Når vi afvikler et semester så øhm, så har vi jo møder i et forsøg på og-, styregruppemøder, i et forsøg på at identificere problemer under afvikling. Men altså på det tidspunkt hvor semester evaluering kommer på banen der, der er semesteret færdigt og de studerende har været til eksamen, og de studerende kan så endelig, endelig evaluere, vi kan sige, det samlede indtryk af semesteret. Og udrykke sig, måske, en anelse mere frit end man kan gøre undervejs i semesteret hvor man jo stadigvæk ligesom har eksamen, ø, forude.

Niki: Nu nævnte du jo selv lige styregruppe møderne...

Informant: ja.

Niki ... øhm, Hvor stor en del spiller de i din udarbejdelse af semesterevalueringssrapporten?

Informant: øh, de spil- altså de er jo en brik. Der, øh, der er jo flere forskellige input. Til semesterevalueringen. Jeg vil nok sige det største bidrag er den undersøgelse der bliver foretaget individuelt blandt de studerende.

Niki: Ja.

Informant: men altså så er der jo, ved siden af det, så er der, øh, styregruppe referaterne. Og så er der øh eventuelle input fra andre parter, underviserne primært. projektvejlederne måske, altså underviserne bredt bliver hørt. har mulighed for at ytre sig omkring de forhold der bliver bragt op, eller hvis underviserne har, har ideer til, til forandringer så, så kan de jo, så kan de jo også indgå.

Niki: mm. Er der noget du sådan vægter højere end andet? Er der noget du bruger mest tid på?

Informant: jamen det, det jeg bruger mest tid på er klart at se på det ret omfattende materiale der kommer ind fra de studerende...

Niki: Så det, det er slutevalueringerne.

Informant: ...ja, ja slutevalueringerne...

Niki: ja okay

Informant: Ja slutevalueringerne det gør jeg.

Niki: Okay. Oplever du at den information man ser i evalueringerne er nærmest redundant, kommer fra, altså oplever du at den information bliver sagt før, i møderne, og...

Informant: Ja noget genkender man, men der er bestemt også... der er bestemt også nye og væsentlige ting der dukker op i slutevalueringerne. Niki: okay

Simon: de, når når der så kommer noget nyt i en slutevaluering. Hvordan øhm, hvis, hvis det foreksempel, det kunne være en meget negativ men måske konstruktiv kommentar der kommer, men det kunne være en enkelt person der måske har det på den her måde, hvordan ville du så, eller ville du så tage vælge at tage det med i din evaluerings rapport, eller ville du måske vælge med at se det som, som et øh, ja, altså ligesom bare en enkelt, en misser, eller hvad

Informant: jeg tro-tror man må sige det ender med at blive måske subjektivt i forhold til min vurdering af hvad der er væsentligt for semestret. For hvis man siger altså bare mellem 2 og 300 studerende ytre sig om alt mellem himmel og jord i semesterevalueringerne, så kunne der skrives en lang roman, hvis, hvis man skulle have alle synspunkter med. Så det jeg forsøger på er at se, er der nogle ting der går igen. og er der nogle ting som jeg opfatter som væsentlige observationer som måske kan komme fra bare en enkelt studerende men noget som jeg synes er væsentlig i forhold til... til kurset eller projekterne på længere sigt. Så det forsøger jeg på at filtrere.

Niki: Ja, okay.

Informant: når, når jeg læser det, men altså der er jo ekstremt mange forskellige kommentarer, og ofte er der noget der bliver vurderet positivt fra nogen studerende, og det samme bliver vurderet negativt fra andre studerende, og så kan man konkludere, altså, der er blandede meninger om det. Så jeg forsøger at finde de store trends.

Niki: ja

Simon: Ja

Informant: fordi min opgave her til sidst som semester koordinator er at lave en ret kort sammenskrivning, en ret kort sammenskrivning som i formentlig kender. I har formodentlig set dem.

Simon: ja

Niki: vi har været...

Informant: fordi de er offentlige tilgængelige.

Niki: vi har prøvet at google på dem og fundet dem, og...

Simon: ja på datalogi og software i hvertfald

Niki: og det er ikke altid det er dem vi kan se der er dem du sender ind, eller i sender ind.

Informant: Jeg vil mene de minder...

Simon: ja for jer, det tror jeg, eller for datalogi og sådan noget, der er de offentlig gjort dem du sender ind. Men vi har snakket med nogle andre semester koordinatore der er det kun de procentvise besvarelser på de der multiplechoice opgaver som bliver offentliggjort.

Informant: Jamen det har jeg jo ikke, ikke så meget føling med, for jeg sender det jo videre i systemet, og så, øh, så øh

Simon: det er studienævnet

Informant: ja, studienævnet, studieledelsen hva nu der er der afgører hvad der kommer videre ud, det kan også være de filtrerer et eller andet

Simon: ja

Informant: det ved jeg faktisk ikke...

Niki: Sådan lidt bredt spørgsmål, måske, har du nogen ting du tænker der kunne ændres ved den process der, som du tænker der kunne gøre det bedre at opnå, altså... bedre formål

Informant: måske, måske kunne jeg egentlig godt forestille mig at det kunne være bedre, mere hensigtsmæssigt at i afreporterer gruppevis.

Niki: nå.

Informant: omkring jeres indtryk, altså, man kan sige vi beder grupperne om at

komme med en, med en samlet, øh, en samlet feedback.

Niki: hrm

Simon: mm.

Informant: det har vi faktisk gjort tidligere. Fordi det kan være, det kan være overvældende at få så mange forskelligeartede kommentarer. Så kan man sige den første form for bearbejdning kan ske i projektgrupperne. Men jeg tror dog alligevel uanset hvad så tror jeg det er vigtigt at vi har en ventil, vi har en mulighed for at individuelle studerende også kan sige noget, altså, som baserer sig på deres eget indtryk. Men jeg tror langt de fleste studerende vil, vil, vil få behovet opfyldt ved at kunne være med til at skrive en samlet rapport gruppevis, altså, hvori, altså hvor synspunkterne skulle afstemme, og måske nok aller vigtigst hvor man har en mulighed for at diskutere det i gruppen og finde ud af hvad gruppen mener.

Niki: Du siger i har prøvet det her før. Hvilken effekt havde det så på de svar man fik, blev de i forhold til de enkelte.. Altså blev de længere, kortere mere konkrete eller hvad?

Informant: Jamen det vi, vi gjorde for en, en hel del år tilbage, var at vi bad grupperne om at gøre det, så kan man så sige, grupperne lavede, skal vi sige en rapport, en side eller to, hvor de sådan på rent dansk, beskrev kvalitativt hvad de mente der var lykkedes.

Niki: Ja okay

Informant: Så var vi nede på et tidspunkt for nogen år siden, hvor vi gik over til de individuelle spørgsmål og havde en meget lav svarprocent, det var meget problematisk.

Simon: ja, okay.

Informant: Det er så rettet sig lidt, fordi vi har, vi har formået at iscenesætte at semesterevalueringerne skal udfyldes på et bestemt tidspunkt, altså omkring starten af det næste semester. Det har gjort at vi har fået væsentligt flere svar, hvilket jo klart er godt.

Simon: mm. Hvad var grunden til den gang, ved du det, at man ligesom udfasede det igen, og stoppede med at gøre det i grupperne?

Informant: ah, men på et tidspunkt så blev altså spørgeskema, og spørgeskema der kunne udbredes med den teknologi man har blev meget populære, og det slog også

igennem her. Og okay, så måtte man lave spørgeskema, og, og det, i starten gik det mig egentligt sådan meget på, sådan man kunne sige, på en skala fra 1-10 hvor tilfreds er du, så får man egentlig bare sådan nogle tal tilbage. Jeg må sige det der er allervigtigst for mig er de kvalitative kommentarer hvor jeg kan læse nogle udsagn fra de studerende, snarere end der hvor jeg kan kigge på de der opsummeringer, hvor mange er godt tilfredse, hvor mange er utilfredse? Jeg kigger selvfølgelig på dem, og det er jo klart en strømpil for den, den samlede skal vi kalde det success af et kursus eller projekt. Det kan man godt se der hvor vi ligger henne, men det jeg klart kigger efter er kvalitative udsagn. Og specielt kvalitative udsagn der måtte gå igen, for mange studerende.

Simon: Kan du prøve at snakke lidt om den, om den process som du ligesom gennemgår, øh, når du skal udarbejde din evalueringsrapport. Altså, du kigger jo selvfølgelig de her igennem som eleverne lavede. Skal vi prøve at snakke lidt om hvordan du griber det an?

Informant: Det er meget simpelt fordi jeg læser det igennem og så, øh, i år markerede jeg med to markere i to forskellige farver, hvad der var, vi kan sige på den positive del af brøken, på den negative del af de udsagn der kommer, og ja, og så. Jeg har jo så to studieretninger som bliver evalueret samlet, så det gør jeg jo for begge de rapporter. Og så, så er der jo en fast disponering der skal skrives, altså som der skal afrapporteres i, altså om semesteret som helhed, og kurserne. og noget om det psykiske, fysiske og æstetiske miljø, og sådan nogle faste ting.

niki/simon: mm

Informant: og dem tager jeg så hver for sig, hvad sidder der på mine opmærkninger af de kommentarer som jeg har lagt mærke til. Og som jeg mener er vigtige.

Simon: så, så du går simpelthen, du går manuelt ind og kategoriserer de forskellige kommentarer?

Informant: ja, det gør jeg.

Niki: Når du læser de her evalueringer her, går du så, altså hvilken struktur læser du dem i, er det sådan at du tager en enkelt elev og så læser du alle hans og tager alle spørgsmål, og så...

Informant: Nej, det kan ikke lade sig gøre, det er faktisk en svaghed. Det kan man ikke se, man kan kun gisne om det.

Niki: okay

Informant: fordi altså... i har formodentligt aldrig set dem?

Simon: det ved jeg ikke. Vi har set dem på en anden format, men det er

Niki: i det data vi har fået kan man se et respondent ID, sådan på hver elev.

Informant: nå, men det, det kan jeg ikke. Og, som jeg kan gisne om, at der er en studerende der har skrevet noget om semester som helhed, og så vender tilbage til det samme måske under et andet punkt der er vigtigt for projektet. Men vi kan ikke se en rød tråd om hvordan en studerende har skrevet på tværs af projekter og kurser. Og det er godt i bringer det op, det er interessant i spørger ind til det der. Det er vi flere der mener er en svaghed.

Niki: i kan ikke se en gruppering. I kan selvfølgelig ikke se navnet, men i kan ikke se nogen gruppering.

Informant: nej, nej, det er sådan set bare en simpel aggregering af, af alle de kommentarer der kommer ind under de overskrifter der nu er. Så hvem der mener hvad om kurser og projekter har vi ingen chance for at vide. Det er heller ikke interessant for os at vide hvem det er , men det kunne være interessant at vide noget om hvad det er for nogen sammenhænge.

Simon: Kender du begrundelsen for det ikke bliver gjort, er det noget med person-data eller hvordan øh? Informant: Nej det kan jeg ikke tro, det er formodentlig på grund af teknisk, enten er man ikke opmærksom på at det kunne være interessant ellers er det fordi man teknisk helt har svært ved at lave det nummer der.

Niki/simon: okay.

Niki: Så det, så ligenu gør i sådan at du går et spørgsmål igennem, eller hvad?

Informant: Jeg tager et kursus af gangen, ja.

Niki: Et kursus af gangen?

Informant: Projektet ikke også skal jeg også, altså semestret som helhed, så projektet og så et kursus af gangen. Når man skal løse store problemer, må man dele det op i mindre problemer sådan er det altid. Så det vil sige så tager jeg skyklapper på og siger: nu kigger vi på projektet, så kan jeg slå op og se hvad er der skrevet om projektet, og så kigger jeg på de trends der er.

Simon: når, når du så har kigget på de her kommentarer hvilken rolle spiller de multiplechoice, svarerne dem får du nok i en eller anden procentvis, øh, går du ind

og sammenligner eller hvordan bruger du det i din rapport.

Informant: jamen det bliver jo et meget kort udsagn om: har det været overvejende tilfredshed, eller peger det den anden retning. Altså det er bare en tør konstatering. Er det at: vi har et kursus her som er problematisk måske, og vi kan se over halvdelen af de studerende har været utilfredse med det i en eller anden grad, og så er den ikke længere.

Niki: Når du skriver din semesterevaluering, hvad mener du så man skal have ud af at læse den?

Informant: Det er jo ret klart, det, det er skrevet til systemet. Og først og fremmest systemet altså kvalitetssikringssystemet, som er jo et stort apparat der kører. I studienævnet og altså længere oppe i systemet der er jo nogle faste, nogle faste evalueringsprocedurer der kører hvert år, og dels hvert tredje år, hvor det her indgår. Så det er en brik i at sikre og beskrive kvaliteten af den undervisning som vi laver. Men altså min men altså min snævre opgave er at levere et kort sammendrag af det materiale der nu ligger i styregruppe referaterne, men jeg vil nok sige primært studerendes evaluering og undervisernes kommentarer.

Niki: Nu nævner du det her med hvert tredje år og sådan nogen ting. Kigger i nogensinde tilbage på tidligere semester og deres evalueringer, eller rapporten for den sagsskyld. Genanvender i feedbacken.

Informant: Jamen altså når man har været semester koordinator i mange år kan man godt huske hvad der stod i sidste år, og hvis jeg er i tvivl kan jeg hurtigt finde den frem, for jeg har den til at ligge. Man kan da også som semesterkoordinator bemærke at det er nu tredje år i træk at vi hører om det her, og der er ikke sket noget. Det kan man jo høre... Men det er klart at når man kommer ned i detaljerne om et enkelt kursus, og det at opsøge detaljer fra 3 eller 5 år tilbage vil være et kæmpestort arbejde, og det, det sker ikke. Også fordi man kunne jo altså, vi har begrænset tid til det her.

Simon: hvor meget... hvor meget tid har i afsat til det her, har i noget bestemt tid afsat til det her?

Informant: Der er ikke udmeldt tid til det, men altså det er jo en del af semester koordinatorens samlede arbejde. Men altså hvor meget tid jeg skal bruge på det i forhold til de andre arbejdsopgaver det er der ingen der har bestemt så det er en konkret vurdering.

Simon: du får de her evalueringer p en dato og så skal du have lavet rapporten til en anden dato?

Informant: Jeg har, jeg ved ikke, en måneds tid, en lille måneds tid til det måske.

Simon: Jeg har en her der hedder noget med, hvor højt, altså hvilke ting, du ligesom vægter højest når du skriver det ind i rapporten, jeg ved ikke, kritik, eller mere konstruktive forslag altså hvilke ting der ligesom går igen, som du ligesom vægter du vil have med i din rapport.

Informant: Ja, jeg vil jo egentlig gerne, ideelt set, gør et forsøg på at balancere det positive med det negative, og egentlig også meget gerne forsøge at fokusere på det konstruktive i stedet for det destruktive... Så et forsøg på både at se på hvad der er lykkedes, målt kvantitativt ud fra de tal der er, og målt kvalitativt ud fra de udsagn der er. Men selvfølgelig også at fastholde de væsentligste kritikpunkter, men også væsentligt efter hvad jeg mener der er væsentligt. Man bliver nødt til på en eller anden måde at være subjektiv der. Hvis det skulle køre på et bestemt mønster så skulle der meldes noget meget meget mere konkret ud. Altså at man ønsker evaluering inden for det og det og det. Det er der ikke, på en eller anden måde er der fristil, man kan skrive 10 linjer om som opsummerer dit indtryk af det her kursus som det har kørt i det her semester. Så det er i bund og grund en subjektiv ting. Altså jeg vil gætte på at hvis man satte fem forskellige mennesker til at gøre arbejdet, så ville der komme fem ret forskellige sammendrag ud. På den måde kunne man godt forestille sig at man kunne iscenesætte det langt mere struktureret, at man fra systemets side ønsker udsagn om det og det og det. Men det gør man ikke.

Niki: Hvilke eksempler kunne være 'det og det og det'?

Informant: Det kunne være foreksempel eksamensforhold. Hvad kommer der frem i det samlede evalueringsforhold omkring eksamen. Hvad kommer der frem i det samlede evalueringsforhold omkring gruppedannelse og problematikker omkring gruppedannelse. Og så er der selvfølgelig, det er der jo altid, en ret tydelig tilfredshed med undervisning og underviseren. Det tror jeg. Hvis man ville det kunne man opdele det i en lang række punkter hvoromkring man ønsker en form for konklusion.

Niki: sådan lidt mere konkrete spørgeark eller hva?

Informant: Jae, altså en ramme der går videre end blot: Der skal stå noget om de tre kurser og projektet og semestret som helhed. Det kunne man godt sige altså: for hvert kursus ønsker vi en evaluering af syv forskellige forhold. Men, det ved jeg ikke, jeg ved ingenting om hvorfor man ikke gør det. Det er ikke det man beder om, så det er heller ikke det man får.

Simon: Ej, så du har ikke noget med udarbejdelsen med de her spørgeskemaer gøre men heller ikke sådan noget længere oppe i systemet.

Informant: nej, nej slet ikke, det sker centralt på et eller andet niveau.

Niki: De her spørgeskemaer, som sagt ikke er udarbejdet af dig, er de meget konstante i det eller flukturerer det meget i hvad de spørger i over semestrene?

Informant: Jeg opfatter dem som meget konstante over mange år. Jeg ser ikke den store forskel, i kender dem godt selv, selvfølgelig ikke over så mange år. Men... Min oplevelse er at de har været i cirkulation en del år uden væsentligt forandring.

Simon: Sådan lige tilbage til de her styregruppe møder. Ideen i det som vi forstår det fra eleverne side er at vi skal... At der er en repræsentant fra hver gruppe som ligesom indsamler noget feedback. Og så tager det med til mødet. Hvordan oplever du at, eller det kan også være svært som studiekoordinator, men hvor meget oplever du ligesom at de får med fra gruppen af og hvor kommer egentligt op når man lige sidder der til mødet?

Informant: Jeg tror en grupper får talt om det inden på et eller andet niveau. Fordi jeg oplever at repræsentanten ligesom, han ligesom har fået besked på at der skal fremføres nogle synspunkter, så føler jeg da også er spontant, som dukker op og er mere personligt. Det mit indtryk at, mit hovede indtryk er at det fungerer godt, så der der, bliver diskuteret mange, mange intrrassante og væsentlige ting på de møder. Og en del grupper i hvertfald er klædt på, da det er noget der er blevet diskuteret i gruppen på forhånd. Og så oplever jeg jo også at der kan justeres på nogles ting, frem for det blog bobler ud i en semester evaluering, så er der nogle ting vi kan tage hånd om på vejs, det er jo det der er hovede ideen.

Simon: Ja, okay...

Informant: Der er jo nogle mindre ting som godt kan over en semester kan hobe sig op til at blive noget større irritation, så der kan vi jo tage livet ud af nogle af dem om jeg så må sige, ved at gøre noget undervejs.

Simon: Ja okay, så det ligesom de mindre løbende ting du tænker på der her bliver taget hånd om?

Informant: Ja, ellers også kan det være et kursus eller formidling i et kursus der ligesom er på vej et sted hen, og så kan det jo godt være at vi over et par møder kan dreje det lidt sådan at det... At flere studerende bliver... Bliver mere tilfredse med det der foregår.

Simon: Oplever du når man er til møderne... Oplever du at alle får sagt de ting de ligesom har taget med til mødet? At det når at blive diskuteret eller blive...

Informant: Det kan jeg ikke sige... Så det ved jeg ikke... Om der er nogle der brænder inde med noget?

Simon: Jeg ved ikke, er der noget bestemt tidsrum...? Jeg har ikke været med til mødet.

Informant: Ja det er der vel givet vis, men... Jeg oplever at de studerende som... Jeg oplever at de studerende eller grupperrepræsentanten får mulighed for at og sige noget om alle punkter... Jeg oplever også en gang i mellem at vi udfylder tiden temlig godt, at vi faktisk får brugt den tid der bliver afsat. Det måske halvanen time godt og vel.

Simon: Ja okay, så altså en bestemt dagsorden som i følger.

Informant: Ja det en standard dagsorden som er fastlagt fra centralens side. Som vi skal følge, vi skal gøre nogle bestemte ting.

Simon: Okay og når du går ind til mødet, så inden har du ingen fornemmelse af hvad der skal diskuteres udover de her punkter.

Informant: Nej jeg kender til intet udover dagsordnen.

Simon: Så er det, er det et problem... Eller kunne det gøres bedre hvor du ligesom havde en ide om hvilke... Hvilke kritik punkter eleverne kom med inden så du vidste hvad du skulle forberede dig på at have en diskussion på det inden.

Informant: Nej, hvis der er behov for det så gør vi det umiddelbart bagefter, så man kan involvere underviser, og det bliver sandsynligvis indskrevet i referatet hvis, hvis ehk hvis har været i kontakt med en underviser og underviser har sagt noget eller hvad nu det er. Så vi gør det, men istedet for at gøre det på forkant gør vi det på bagkant. Men vi har et referat som jeg synes er godt. Men det klart det kan gøres på flere måder. I bund og grund synes jeg at vores setup omkring styregruppe møderne er rimelig velfungerende. Nok egenligt mere velfungerende end vi kan sige om den afsluttende evaluering.

Simon: Men du mener ligesom at de har to helt forskellige hovedformål.

Informant: Ja det er jo brikker indenfor det samme spil. Men... Jeg synes at formatet med styregruppemødet og ideen med gruppe repræsentaten... Muligheden for at diskutere det i en dialog fremfor at du får vi bare en stor dyng af papir. Jeg synes der er noget, der er noget i det der gør i det at vi kan få en fælles forståelse af nogle ting ved at diskutere og tale om det det, snarere end vi bare får en hel mængde

fakta på papir og så skal der skrives en meget meget kort sammendrag.

Niki: Hvad hedder det ehm... Et meget åbent spørgsmål der kommer her... Hvis du nu boede i en perfekt verden her med uanede mængder af tid og resurser og værktøjer og alt. HVordan ville du så have denne process for semesterevalueringerne skulle se ud for dig?

Informant: Jamen i... Hvis det var en ideal verden så skulle vi jo ikke køre undervisningen som på den måde vi gør i mega store hold, så skulle vi jo ned i nogle størrelser hvor... Hvor der var meget færre studerende som deles ud i undervisningen. Så det vi gør er jo meget præet af den stordrift vi har omkring os. Og svært at forestille, det svært at forstille sig at skulle gøre noget der er væsentligt anderledes på evalueringssiden. For det ville jo fuldstændig slå os ihjel hvis vi skulle tale med den enkle studernede, tale med den enkle gruppe også. Så jeg tror hvis der skulle ske noget så handler det mere om organiseringen af undervisningen. Og den der opgør med den der stordrift vi har... Snarrer end det er måden vi evaluerer undervisningen på. Når vi har så kæmpe store hold, hvordan skal man så pokker gribe evalueringen an? Hvis man derimod kun var ganske få, så kunne man løbnede have en dialog om det. Så ville vi altid... Så kunne vi tale om det, men det er meget svært at tale om noget hvis man er et par hundrede i et meget stort rum... Så er vi ude i noget med at udvælge repræsentanter og så skal vi holde møder med de repræsentater osv. Så det vi gør er et udslag af den måde vi organisere undervisningen på.

Niki: Nu har du lidt en IT baggrund jo, er der noget du tænker der kunne automatiseres af den her store process?

Informant: Jaaaa... Ved de her spørgeskema er vi jo allerede igang med at automatisere noget i forhold til noget vi ville gøre før computeren var opfundet... Men det er jo klart man kunne gøre det endnu mere struktureret, og man kunne stille nogle analyseværktøjer til rådighed. Jeg sidder på en endnu større grad af strukturering af det materiale materiale som hentes tilbage fra de enkelte studerende. Så [hvis] man havde nogle analyseværktøjer som var bedre eller kraftigere, ... så kunne man trække langt flere interessante ting ... ud af de data. Det er oplagt. Så altså spørg på en mere struktureret måde, for at få nogle mere struktureded svar tilbage, og forsøge på at analysere det, ved at bruge den teknologi vi har. Men også som jeg kunne forstå, i var lidt inde på også at se tilbage, ikke bare tage et enkelt år men at man måske kunne se tilbage over flere år. Det kunne klart være interessant.

Simon: 31:50 Det er en tangent, men ser du det som et problem at når du giver de her spørgeskemaer til elever, de skal fylde en kommentar ud omkring noget. Når du sådan spørger dem om "Er der nogen problemer med det her" ser du det måske som et problem at de måske ... [Eksempelvis] du bliver spurgt "Er der noget galt" og så finder du måske på et eller andet der er galt, i stedet for at man måske opsøger

det hvis det er man har noget større man faktisk mener der er galt.

Informant: 32:24 Med den måde det hele bliver lagt op på, der ville jeg tro at de fleste studerende der svarer på det her, tænker i retningen af noget der ikke har fungeret så godt, og det er så det der kommer til at præge det der bliver skrevet, de udsagn som kommer. Så for hver gang der ... bliver skrevet én positiv ting, så er der skrevet 15 negative ting. ... Så ud af den spørgeramme, som man giver de studerende, kunne man godt forsøge på at balancere det. Sådan at man faktisk også for belyst de ting som faktisk fungerer. Det kan nogle gange være svært at uddrage. Måske bortset fra når man ser på de der tilfredsheds tal [med] procenter, hvordan det egentlig ligger. Men det ville jo så kræve at man udformer spørgerammen på en anden måde.

Simon: 33:32 Så lige tilbage igen, til den her perfekte verden her. Ser du at den måde som din endelige evaluering kommer til at se ud ... eller din endelige rapport den ser ud på nu, at den ville være magen til hvad der er i den perfekte verden. Eller ville der være ting du gjorde mere af.

Informant: 33:55 Det er jo klart at man kunne skrive en grundigere rapport, som tager endnu flere synspunkter ud i samlet betragtning. Det der jo så sker at i et eller andet sted i vores system, så kommer der 25 af den slags rapporter ind, og så sidder der 7 mænd i et udvalg der skal læse endnu mere materiale. Så tror jeg allerede i dag føler man at det er en byrde. Når man skal evaluere alle semestre på alle uddannelser. Hvis der så kom vældig voluminøse rapporter ind, så dør de vel egentlige bare. Der er ikke nogen der orker eller formår at have ressourcer til og reagere på dem. ... Jeg bliver bedt om at skrive sammendrag, og de kan fylde, 2-3 sider, men i de senere år er jeg blevet bedt om at lave sammendrag.

Simon: 34:51 Så den endelige evaluering du egentlig laver, føler ud at den er repræsentativ for det data du egentlig for ind.

Informant: 34:57 Nej, det er jeg ikke sikker på at jeg gør. Jeg ved godt jeg skal skrive det kort, så der er bestemt mange synspunkter som jeg ikke formår at få med. Men altså når jeg så fornemmer at længere oppe i systemet at de faktisk ber om sammendrag. Det er for meget at syntes at jeg skal skrive 15 sider i stedet for 3 sider. Det er et forsøg på at sige at systemet efterlyser, i udpræget grad, nogle meget korte, og meget filtrerede udsagn om hvordan det er.

Niki: 35:44 Så kun det vigtigste, og helt kondenseret?

Informant: 35:48 Så alternativt kunne man forestille sig at hele materialet, var til rådighed og at man så kunne på et højt niveau udtrække automatisk nogle ting der fra.

Simon: 36:00 Kvantificeret på en god måde?

Informant: 36:01 Idag interessere vi os for noget helt specifikt. Lad os nu prøve at udtrække "Hvordan var det her i de sidste 3 år". Det ville jo være et helt andet spil. Det ville jo kræve at man investerer i et setup som er helt anderledes. Relativt, at lave et eller andet standard spørgeskema man ville bruge over 5 år, så køre det for år til år, og så sætte ... mænd til at skrive 2 sider sammendrag til en festlig lejlighed.

Simon: 36:48 ... Du har selvfølgelig læst noget [om vores projekt ide] ... Men på baggrund af det du snakket i dag, og også i nogle af de andre interviews vi har haft, der tyder det på at der er flere forskellige steder i kæden man kan forbedre ting. Man kan automatisere dem. ... [vi] vil prøve at stille dig lidt mere ledende spørgsmål i forhold til hvilke ting vi havde tænkt om med vores projekt. Så noget af det vi forestillede os var at ... tillade at give dig en nemmere oversigt, og kategorisering. Eksempelvis det du faktisk gør nu, med du deler dem op i positivt og negativt, og måske finde nogle holdninger der minder om hinanden. Om det ville være en hjælp sådan for dig, i en vis grad.

Informant: Det er vel klart, hvis man kunne få et mere sikkert og mere tydeligt overblik, over de trans(38:06) der er, snarere end jeg skal kigge på, farvelade klatter på mange ark papir. Det ville da tillade en mere nuanceret og mere sikker vurdering af det materiale der kommer ind. Det ville klart være en god ting, hvis man kunne det. Fordi som sagt jeg har ingen forestillinger om at det der sker i den process jeg gør er 100% objektiv. Det er det ikke. Det kan det ikke blive. Så det afhænger af mit syn på det. Hvordan jeg oplever semestrene, hvordan jeg oplever de problemer der. Vi [har] den erfaring der er, vi har den deltagelse jeg har haft. Mange af de synspunkter der dukker op er om noget vi har talt om nu, primært på styringsgruppe møde. Indenfor både projektet og kurser, udvælge nogen ting der bliver spurgt om, som måske også kunne gøres til genstand for kvantificeringen, altså ned i detaljen. Altså nu tænker vi for eksempel eksamen

Niki: Ja.

Informant: Det godt, ja, ja, men jeg er ikke tilhænger af ekstremt komplekse og lange spørgeskemaer.

Simon: Nej.

Informant: Vi hader dem alle sammen. Det gør i, altså jeg hader dem og i hader dem, og man får altså en tendens til at stå af på den og lige pludseligt skal det begynde at gå meget meget hurtigt og så er det ikke så seriøst længere. Så det, jeg tror også det er noget man må tænke på.

Simon: Man begrænser kvaliteten af dataen du får.

Niki: Så vil jeg lige spørge tilbage til det her, hvad med sådan nogen enig / uenig / jeg ved ikke udsagn, hvis der nu stod et eller andet udsagn omkring et eller andet. Jeg synes det her ikke også, hvor man så kan trykke enig / uenig til det, vil det give mening? Altså, for dig at kunne læse sådan noget.

Informant: Altså, for mig er det personligt og det... Jeg har ikke, jeg kan ikke forholde mig videnskabeligt til hvordan man skal gøre sådan noget, altså for at lave de bedste spørgeskemaer, altså det ved jeg ikke.

Niki: Nej.

Informant: Men jeg vil hellere se at man gik i en mere kvalitativ retning med udsagn indenfor... altså brede en ramme ud kan man sige, vi vil gerne have udsagn, både det at vi vil gerne have et positivt udsagn og et negativt udsagn. Snarere at det bliver sådan en oppustning af spørgeskemaet sådan at der bliver mange mange flere spørgsmål og... Hvis det bliver for komplekst så... Respondenterne de er ikke omhyggelige nok med at forstå hvad det er der bliver spurgt om.

Simon: Nej.

Informant: Jeg tror ret hurtigt det kan gå i en retning, hvor vi får noget tilbage der er, der afspejler at man ikke gider bruge tilstrækkeligt mange ressourcer på det. Så det er ikke let.

Simon: Nej fordi tanken var ligesom at noget af det vi har hørt fra tidligere semesterkoordinatorer, det er at de mener at når du bliver bedt om skrive en kommentar til et eller andet, det kunne være projektet overordnet set, så skriver du én ting, når du nu er færdig med den boks, hvor du måske har 3 mere. 3 kommentarer mere eller noget du ikke lige tænker over. At man så i forlængelse af det, går ind og finder nogen kommentarer som andre elever har lavet på det samme punkt forrige semester eller sådan noget. Og så præsenterer du ligesom 3 eller 4 af dem og så kan du stille dig enig i dem og sige "det er jeg egentlig enig med dig i" eller det havde jeg ikke lige tænkt over at formulere.

Informant: Ja, ja, men det at se nogen ting på tværs af årgangene, både måske omkring spørgerammen, men særdeleshed også i den måde man konkluderer på er meget interessant. Det kræver der vil blive en form for understøttelse af hvad skal man sige, hvis det skal være manuelt.

Simon: Ja.

Informant: Fordi det er så stort.

Simon: Ja, ja. Hvordan det.. Nu snakker du om at du har været på datalogi og software i en række år. De studieretninger du havde tidligere var det samme mængde elever? Du havde altså samme (præcitet) af data.

Informant: Nej, vi har oplevet de sidste 10 år, at der nærmest hver eneste år kom flere og flere til, så det er jo vokset fra noget der samlet set var, skal vi sige, under 100 alt i alt, til noget der nu er mellem (200-300).

Simon: Ja, okay. Altså det er klart, det går selvfølgelig i en retning, hvor du får mere data. Hvordan oplevede du det den gang der du havde de der under 100.. 90 elever, at datamængden var?

Informant: Altså det er svært, det er svært at sige, fordi der er jo sket.. Systemet fra Aalborg Universitet har ændret holdning til hvordan det hele skal foregå, altså som jeg sagde, fra de gode gamle dage, altså de gamle dage de er altid gode.

Simon: Ja.

Informant: Men i de gode gamle dage, der lavede vi de der gruppevis.

Simon: Ja.

Informant: Så vi havde nogle få grupper og hver gruppe skrev en form for fristil, men dog indenfor.. Det fremgik at man skulle forholde sig til de her ting.

Simon: Så der var ligesom en ramme.

Informant: En ramme ikke også. Men så skrev man i rapportdelen et par sider per gruppe. Det synes jeg var meget meget interessant, fordi det jo er bearbejdet af en gruppe og det var filtreret. Man kan sige de mest hurtige holdninger man lige kunne komme med var filtreret og bearbejdet. Og på det tidspunkt, hvor vi ikke havde mere end nogle få grupper, skal vi sige.. 10 grupper måske alt i alt? Så var det også overkommeligt at læse, så det var meget interessant.

Simon: Ja, okay.

Informant: Men igen, skal jeg det optage i dag? Det er jeg ikke sikker på. Kan vi overhovedet overtale de studerende i dag til at lave sådan noget. Det er jeg heller ikke sikker på igen.

Simon: Nej, nej. JA fordi jeg tror vi startede 24 grupper på vores semester, og der skal vi selvfølgelig også trække nogle fra på første studieår.

Informant: Det er klart.

Simon: Det bliver så hurtigt endnu større mængder data.

Informant: Ja i år har vi over 30 grupper.

Simon: Okay, ja. 7 og 6 mand i hver gruppe?

Informant: ja 7, deromkring.

Simon: Ja, jeg tror ikke vi har så meget mere.

Niki: Har vi flere spørgsmål?

Nanna: Jeg har ét spørgsmål.

Informant: Ja.

Nanna: Vi hørte fra tidligere semesterkoordinator, at når man ikke svarede på spørgeskemaet, at så var det positivt. Tænker du også det?

Informant: Altså når en studerende ikke svarer på det.

Simon: Ja, som at når de ikke har noget at sige, så må de være tilfreds.

Informant: Det tror jeg ikke du kan compilere. Det tror jeg ikke.

Nanna: Det tror du ikke?

Informant: Nej. Jeg tror der kan være mange årsager til at man ikke svarer på spørgeskemaet. Der kan selvfølgelig være nogen der tænker alt er godt, så det behøves jeg ikke at fortælle. Jeg har ikke noget at brokke mig over. Det er sådan set den der. Jeg er også sikker på at der vil være nogle der siger at.. Jeg er så forvirret over det hele og der er så mange ting der er oppe at køre her, så det svarer jeg ikke på. Så jeg tror der kan være nogen som måske har det svage segment, som har mangfoldige problemer, som vi heller ikke hører fra. Så derfor synes jeg det er vigtigt at vi gør os umage for at få rigtig stor svarprocent.

Simon: Ja for du oplever også den trend, at det ligesom er nedadgående svarprocent.

Informant: Nej, det afhænger meget af hvordan vi ser det. Det har været en succes,

at det bliver taget op omkring semesterstart.

Informant 3

C.1 Interview af Informant 3

Informant: Det er fint.

Niki: Så vi kan give nogle gode citater.

Informant: Yeah.

Niki: Okay, men meningen med vores interview her, det er at vi gerne vil ønske at lave et system, som for dig som semesterkoordinator skal kunne arbejdsprocessen med at lave de her evalueringsrapporter i laver i slutningen af hvert semester og ligesom prøve at optimere den proces på en eller anden måde, og hjælpe jer. Så vil vi gerne prøve at forstå din proces, hvilke ting du synes der er gode, hvilke ting du synes er dårlige og hvad du gerne vil ændre måske og sådan nogen ting. Og dette er et semistruktureret interview, hvilket betyder at du ikke skal begrænse dig til korte svar, du skal bare gå langt ud af tangenterne, hvis du har lyst. Vi vil gerne have det hele med.

Informant: Halv et har jeg en eksamen igen, så det er maks én time.

Simon: Ja, det var vi heller ikke overrasket over.

Niki: Nej, nej det tror jeg heller ikke.

Informant: (Utydeligt?)

Niki: Yeah, jamen det er helt fint med os.

Niki: Okay, vi vil gerne høre lidt om din opgave som semesterkoordinator til at starte med, hvor længe du har været det for eksempel.

Informant: Semesterkoordinator det har jeg været i de sidste 20 år.

Niki: 20 år?

Informant: Ja.

Simon: Okay.

Informant: Jeg har haft forskellige semestre, men jeg ved ikke om det kun er ud fra min rolle som semesterkoordinator på første semester.

C.1. Interview af Informant 3

Simon: Ikke specifikt nej.

Niki: Nej, ikke specifikt nej.

Informant: Nej okay, fordi jeg er semesterkoordinator på dem og det er by, energi og miljøplanlægning på 1. semester. Jeg er semesterkoordinator på geografi 2. semester, på geografi 8. semester og ja, det er det. Og sådan lidt standby på 5. semester. Men det er ... og 10. semester med geografi også. Og det er så lidt forskelligt vil man sige, vil man. Opgaver som sådan er det samme om man nu er semesterkoordinator på 1-, 2-, eller 10. semester. Men det er klart, på første semester så er der måske flere studerende i de uddannelser, hvor jeg er semesterkoordinator. Hvor man på geografi når man har en på speciale, så er der måske 5 eller 2.. 3.. studerende nogen årgange. Så arbejdsbelastningen er lidt forskellig.

Niki: Ja, selvfølgelig.

Informant: Kan man sige. Jeg har en del erfaring på forskellige semestre (2:01 Utydeligt?).

Niki: Det lyder dejligt. Hvad hedder det ehh... Nej det har du også svaret på, spørger hvor mange studier du har været semesterkoordinator for. Hvad hedder det ehh... hvor mange studerende har du sådan samlet set på et semester sådan, hvor mange har du haft? Over alle dine studieretninger og semestre på et år.

Informant: i de 20, jo men da det (Afbrydes)

Simon: Bare et enkelt semester.

Niki: Altså, bare på ét semester, et enkelt semester over alle dem du er semesterkoordinator for. Hvor mange har du så cirka?

Informant: Hvis vi nu tager udgangspunkt i for eksempel nu i første semester på dem, der er vi 38 studerende.

Simon: Okay.

Informant: Øhhhm, sååe, ja, og.

Niki: Men, hvad med alle de andre du også er, altså alle de andre semestre?

Informant: Ja okay, men på dem første semester, så er vi omkring de 38-39 på geografi. 2. semester i år er vi omkring de 21 studerende. Sidste år var der også 38,

C.1. Interview af Informant 3

men hvis man kigger historisk, så plejer vi at have, og det gælder sikkert også for dem, de 20-25 studerende per semester. Og det er for dem der gerne vil have hele bachelor, selvfølgelig er der lidt frafald, så holdene bliver måske lidt mindre, så vi ender med 17-18 stykker på 6. semester. Og så er det på master, vi har på geografi, har vi ikke haft, det er først 2. år igen. Vi har haft en master, men sidste gang vi havde studerende på master, det var måske i 2010, så har vi ikke haft en master-uddannelse i nogle år. og så sidste år, og det var i 17 var det første gang at vi havde nogen studerende, fordi der havde været en studieordningsrevision, hvor vi så fik en master-uddannelse igen, og så var der 6 studerende, og der er nu 3 studerende, som kun er på geografi. Så det er de meget små semestre, kan man sige, men i snit 20-25 vil jeg sige.

Niki: Hvad hedder det ehh.. Når nu du laver de her semesterevalueringsrapporter her, hvad mener du så formålet skal være med at de studerende de udfylder de her. Hvad synes du formålet med det er? Eller hvad det burde være?

Informant: Formålet det er at man kan forbedre undervisningen. Måske også at de studerende kan forstå, jeg ved ikke om de læser semesterevalueringsrapporter, det tvivler jeg lidt på, men hvis de læser det, at de også får forståelse for hvorfor man måske nogen gange ikke gør nogen ting, som de foreslår på styringsgruppemøder. For eksempel, der kom en række forslag med hvad man kunne gøre anderledes, men ikke nødvendigvis at det kan blive implementeret, men formålet er helt klart kvalitetssikring af undervisningen.

Niki: Synes du så at dette formål opfyldt i dag?

Informant: Ja det gør det. Jo det gør det. Men det er også meget op til den enkelte underviser at reflektere over; hvad har jeg gjort? Eller vejleder for eksempel. Hvad har jeg gjort, eller hvad skulle jeg have gjort anderledes? Men selvfølgelig er det helt klart rart at når man skriver det, at man så får overblik over hvad er der blevet sagt. Fordi når jeg tænker på semesterevalueringsrapporter som jeg skriver som samlet, så er det de referater fra styringsgruppemøder, som man har sådan cirka 3-4 gange på et semester med de studerende + den der SurveyXact, som i udfylder, og så sætter man det sammen i ét. Det vil jeg godt kunne ønske mig. Jeg ved ikke om jeg kan komme ind med ønsker eller hvordan det kunne være bedre.

Niki: Jo, helt klart. Helt klart.

Informant: Så vil det være

(Afbrydes af ekstern person)

Informant: At det også blev en mere kvalitativ. Nu er den der SurveyXact, den

C.1. Interview af Informant 3

lægger nogle bokse og kommer med nogen inputs. Nogen gange er der mange inputs andre gange er der stort set ingen. Det er sådan lidt forskelligt hvor mange inputs man får og nogle gange har man et indtryk af at dem der har været utilfredse med noget, de bruger meget tid på det, og dem der har været tilfredse, de siger bare; "Ja, det er fint nok", og så står der ikke noget og så kan det på en måde føles lidt misvisende, fordi vi har som undervisere også en idé om; hvad synes vi der gik godt? og hvad gik mindre godt? Men hvis der så er nogle få udsagn omkring et eller andet, så vises det som om det er et meget større problem end det måske har været. Det synes jeg da helt klart også er en opgave for semesterkoordinator, at i (6:29 Utydeligt?) styringsgruppemøder, sige til de studerende "I må gerne komme med et forslag hvordan". Hvis man er utilfreds med det hele, så er det fint, men kom også med et forslag. Hvordan skal man så gøre det anderledes? Det er lidt det samme, at bare i får feedback på jeres arbejdsblade.

Niki: Ja, ja.

Informant: (6:45 Utydeligt) dårlige i har skrevet.

Niki: Ja, ja.

Informant: Og så ikke for noget, men (6:50 Utydeligt) så tænkte jeg også lidt, øhh.. hvordan?

Niki: Så du foretrækker klart de kvalitative aspekter af spørgearket?

Informant: Ja, jeg, det kunne på en måde, jeg ved ikke hvordan på andre uddannelser, men...

Niki: Så du foretrækker de kvalitative aspekter af spørgearket?

Informant: Ja, ja det kunne man på en måde. Nu ved jeg ikke hvordan det er på andre uddannelser ... hvis man f.eks. Skal planlægge andet semester, så får man studerende der var på sidste år var på andet semester med fordi de kan jo huske hvordan andet semester blev gjort når man har planlægningsmøde til næste semester så kommer de studerende fra sidste år [med, men] på første semester kan man ikke gøre det, for sidste år har man ikke nye studerende fordi man ved jo ikke hvem der kommer på første semester fordi ... vi får ... navne lige et par timer inden semesterstart. 7:38 På alle de andre semestre får man også input fra de studerende igen, så på den måde har man lidt en kvalitativ input, som jeg syntes er meget vigtigt at have med.

Simon: 7:49 Så når du udarbejder den her rapport til sidst, ... kan du prøve at tale lidt om din arbejdsproces? Fra når du modtager til [afbrudt af Informant].

C.1. Interview af Informant 3

Informant: 8:00 Jeg har alle de referater fra styringsgruppe møder, og så har jeg den der survey [fra] xact som man får, og så har jeg fra studienævnet en skabelon som man skal udfylde. Det er en standard skabelon. Hvordan var semestret generelt, hvordan var det med projekt, og så én for hvert enkelt kursus. Og så til sidst eventuelt. Den første del af blanketten kan være anonym, ... hvis der nu er nogle personfølsomme oplysninger ... , men det har jeg aldrig haft. Jeg ved ikke om jeg kan finde på et eksempel, men hvis der nu stod et eller andet omkring et kursus, som jeg ikke vil [have] bliver offentliggjort andre steder, [så] er der til sidst en boks, med en sammenfatning af alt det man lige har skrevet, som offentliggøres for alle. Men der komme så ikke personfølsomme oplysninger eller andre oplysninger ind som ikke skulle komme ud. Men det har jeg aldrig gjort, fordi for mig det hvad jeg skriver. Så det på princippet en sammenfatning er en sammenfatning til sidst fordi man tager de forskellige punkter. Og så bagefter når jeg har skrevet det så sender jeg det rundt til de studerende der har været i styringsgruppe møder, og til alle undervisere på semestret, hvor de kan komme med feedback, “er i enige med denne der sammenfatning, eller fælles evaluering, som jeg har lavet”. Så kan de eventuelt komme med input, det sker stortset aldrig. Så jeg sidder og bruger noget tid på det, med en masse evaluering, og så sender jeg det rundt, og nogle gange er der nogle få ting, så retter man det, og så sender man det til studienævnet, og så bliver det behandlet på studienævnsmøder, og så godkender de. Og så fremhæver de f.eks. Noget som f.eks. “Okay I det kursus skal man ... ”

Simon: 9:59 Synes du at ... nu siger du at du laver en sammenfatning af en sammenfatning til sidst, føler du at det samlede, de produkt der kommer ud af det, er det repræsentativt for hele det studies meninger eller alle [afbrudt af Informant].

Informant: 10:16 Det er der sådan set, men jeg tænker meget hvis man bare kunne klikke nogle bokse for eksempel, hvis der så automatisk kommer ... x-antal udsagn for eksempel, og dem synes jeg er vigtige at have med, ... så når jeg til sidst skal have [en] fælles opsamling, hvis den så kunne ... genererer automatisk ... og hvis man nu havde nogle udsagn som jeg markere ... som at den ikke skal komme med fordi den er personfølsom eller [noget andet]. ... Nu ved i ikke hvordan moodle ser ud for os, men der er nogle ting, hvor der står “Hidden for students”, så det er noget jeg kan se, men i kan ikke se det, og jeg skal skifte mine roller til at være studerende, så jeg kan se hvad kan i se. Hvis man havde det på den måde kunne vinke (11:05) nogle ting af, eller til. Eller hvis det kommer automatisk med i den der sammenfatning det kunne også være dejligt. Jo mere automatisering af den front, jo bedre.

Niki: 11:19 Hvor meget vægt lægger du på den enkelte elevs feedback? Er det mere sådan at den enkelte person siger et eller andet, har det så nogen betydning, eller skal det være mere generelt relevant.

C.1. Interview af Informant 3

Informant: 11:28 Nej det skal være mere generelt, det kan lyn-hurtigt være misvisende. Det er misbrug af statistik hvis man har nogle [Informant afbryder sig selv]. ... hvis man har 25 studerende der siger et eller andet, så skal man finde ud af at det er der også blevet sagt ... [i] styringsgruppe møder, eller er det slet ikke blevet sagt. Fordi hvis én syntes det er et rigtig dårligt kursus, for eksempel. Sådan nogle udsagn tager jeg heller ikke ... hvad skal man sige, der skal være nogle argumenter hvorfor, eller forbedringsforslag. Fordi jeg kan ikke bare [bruge] "det var dårligt", det kan jeg ikke bruge til noget. ... Jeg prøver at redde på første og andet semester og så opdrage de studerende med "I skal tænke på hvordan i selv gerne ville have feedback" og så kan man på den måde bygge videre. ... De fleste af os har været semester koordinator over flere år. Og så er det måske hvis det flere gange(12:25) [end et] enkelt tilfælde, ... så kan man ... se måske en lille smule at ... der alligevel [er] nogle kommentare, selvom det kun er få der nævner det, der kommer tilbage. Det er lidt anderledes end hvis der kun er én gang på ét år står, et eller andet. Men det kan jo både være de positive og negative ting. Man husker det ikke, men man er lidt kreativ med statistik, hvis man så siger "én ud af 25 synes noget er rigtig dårligt så må man lave om på det hele".

Simon: 12:58 ... Men hvis den her ene kommentar, ... så den angår et eller andet ... hvor der er blevet kommenteret noget lignende for 2 eller 3 semestre i streg, så du så huske tilbage til [afbrudt af Informant].

Simon: Men hvis den her ene kommentar, angår et eller andet, hvor der er blevet kommenteret noget lignende for 2 eller 3 semestre i streg, er det sådan noget du, kan du, husker du tilbage når du...

Informant: Ja, ja, ja men, der kommer jo nye, jeg følger på den måde ikke de studerende.

Simon: Nej...

Informant: Men, hvis vi, tænker vi på hvem har aftalt, at for eksempel gruppedannelsesprocess. Jeg er nu, næste uge har vi så gruppedannelse på P1, på (???). Nogle gange går det rigtigt god, andre gange så er det (???) process. Det har i måske selv været oplevet, det ved jeg ikke. Men hvis det nu er, hvis der nu sker ting som jeg synes at, semesterkoordinatoren på andet semester skal være opmærksom på så siger det så videre så på den måde kommer der (???) hurtigere igen, fordi semesterevalueringssrapporter de, kan de nå at ind(???) som andet semester næsten er gået i gang. På den måde skal, kan man god kvalitetssikre på tværs af semestre. Eller hvis der er nogle studerende med, hvad ved jeg, specielle behov eller et eller andet, skal man lige gøre opmærksom på hvordan det eventuelt kan påvirke, gruppedannelsen.

C.1. Interview af Informant 3

Simon: (???) hvis vi så forestiller det er noget til det konkrete kursus, du har så første studieår, så første semester, der er et bestemt kursus der går igen. Og du så, der kommer en eller anden kommentar på det her kursus, eller noget i den stil, så du så næste, året efter så har det samme semester her, der er så nye elever som siger det samme...

Informant: Ja, så får vi hele, ja ja, jo jo.

Simon: ...er det sådan noget du husker tilbage på.

Informant: Men jeg tager det sådan, vi tager det sådan set også, det kommer måske ikke i den der samlede evaluering kun hvis der, hvis der står kommentar fra én studerende, men selvfølgelig tager man det med, fordi man har alt materiale, stadigvæk med, og det sender man også til de planlægningsmøder, så gennemgår man det også igen, hvor man siger, der har været nogen som har sagt et eller andet. Og så kan man så spørge underviseren, fordi jeg er jo ikke med til alt undervisning. Så jeg ved ikke hvad der er blevet sagt, eller hvad der er blevet gjort. Og nogle gange siger de studerende én ting, og underviser har (???), det har jeg da slet ikke sagt, eller et eller, og det er også lidt, begge veje.

Simon: Så når du udarbejder så kigger du tilbage eller husker tilbage på de tidligere evalueringer, og bruger det.

Informant: Og det er helt klart en fordel når man har været på samme semester mange år i træk. For at man kan huske hvad der er sagt og gjort, og hvordan man kan gøre tingene anderledes. Så på den måde er der lidt en indirekte kvalitetssikring. Ligesåvel at man har de samme undervisere på kurser, sådan cirka tre år i træk, mindst.

Simon: Du sagde at, de kommentarer du godt kunne tænke dig at have, at det var primært, sådan, konstruktive, (???) det ligesom gerne det du vil arbejde med og have med videre i processen i din rapport. Sådan noget som, de kommentarer der måske siger hvilke ting der virker godt, som bare er positiv (???) ikke noget der kunne ændres, det er også noget du bruger i din rapport? Eller det ser du bare som..?

Informant: Nej, det tager jeg sådan set med, hvis der er konkrete forslag til forbedringer, dem tager jeg med og gemmer dem med, fordi den enkelte underviser på et kursus får også det udklip som står på et kursus, så hvis jeg har et kursus, så får man også at vide, ok, det og det synes de studerende fungerer godt, det synes de var mindre godt, men de foreslår at man for eksempel... Og det kan være helt lavpraktisk fra at man skal holde flere pauser, til at det skal være et eller andet om litteratur der bliver brugt eller øvelser der skal laves, om de er for store. Eller forberedelsestid passer ikke til hvad der står i semesterbeskrivelsen.

C.1. Interview af Informant 3

Niki: Hvordan skaber du et overblik over alle de her svar du får, når du sidder og skal læse de her igennem, så får du vel et langt ark af svar, hvordan skaber du dig et overblik over det her?

Informant: Ja, jeg læser det hele igennem, og så prøver jeg at se, der er en del kvantitativ, med sådan nogle grafer, med hvor mange studerende der er meget tilfreds til meget utilfreds. Så har man lidt et indtryk af det, og så er der de supplerende svar, for eksempel fra hvorfor nogen har været meget utilfreds eller meget tilfreds, for eksempel. Det prøver jeg(???), sige at, dem der har været meget tilfreds, de skriver måske ikke hvorfor de har været så tilfreds, de tænker det har været fint, jeg har været tilfreds, næste spørgsmål. Men dem der har været meget utilfreds, de bruger mange gange mere krudt på at skrive hvorfor det var. Ikke den, bedste.

Niki: Så du tager ikke nogen noter løbende, eller sætte en markeringer, eller overstregningstusch eller..?

Informant: Nah men så stor, jo... , neej, for så mange, kommentarer kommer der trods alt heller... , det er overskueligt vil jeg sige, ikke at man behøver tage noter.

Informant: Noget hvad vi gør nu på (???) og Geo er en del sammenlæsning af kurser. Så nu har vi valgt at, jeg har været en gang hvor jeg både var semesterkoordinator på GEO1 og BEM1, og så holdt jeg så styringsgruppemøder sammen, for de har en del sammenlæsning, og så kan man også se. For nogle kurser for eksempel, synes geograferne for eksempel, de har også lidt med fysik og ting og sager, det synes de er meget svært, og BEM'er måske ikke synes det, eller omvendt. Eller (???), hvad ved jeg. Og forudsætningerne er ikke altid det samme for alle studerende. Så snakker man også lidt på tværs, og så er det rigtigt, så bliver det måske lidt....

Niki: Lidt meget lige pludselig.

Informant: Så får man flere oplysninger, og så skal jeg i princippet i bund og grund kun have min BEM kasket på og tænke jeg er ligeglad med hvad geograferne har sagt. Men vi tænker for kurserne er det bedre at, fordi de sidder jo i samme lokale, og de hører den samme kursus, at man så ellers, plus at de studerende kan svare eller hjælpe hinanden lidt, hvis du siger noget så kan du komme (???).

Niki: Nu nævner du ofte de her styringsgruppemøder. Hvor stor en del, hvor meget vægter de i forhold til udarbejdelsen af en rapport her.

Informant: De vejer mere. Bevidst eller ubevidst. Fordi så sidder jeg face-to-face, med de studerende, og så kan man spørge ind til, hvad mener du når du siger at...? Det er ikke så meget på de høje semestre, men på de lave semestre at man også,

C.1. Interview af Informant 3

hjælper dem med at tænke okay, i har været (???), for det er meget meget direkte, ”og det har været dårligt”, fint, i har været utilfredse, hvorfor har i været, og så kan i komme med nogle konkrete forslag, for vi inviterer altid alle forelæsere med til alle styringsgruppemøder, men det er sjældent at de kommer. Så de skal også, hvis man læser et referat, kunne forstå, hvad er der blevet sagt. Og er der nogle ting jeg kan ændre her og nu, eller skal vi vente til næste år, inden man kan lave nogle løsninger, hvis der er nogen der, jeg har alt for mange slides eller, snakker for lavt, eller et eller andet. Alle sådan nogle kommentarer man får. Der er også nogle ting man kan direkte til underviseren, men er det vigtigt, at de, det vægter jeg højere end SurveyXact, fordi man netop har mulighed for at spørge ind til, og få den kvalitative, eller flere eksempler.

Informant: *Host* *Host* Undskyld... At de ehh så det vægter jeg højere end surveyxact. Fordi man netop har mulighed for at spørge ind til, og for den der kvalitative, eller måske flere eksempler.

Niki: Oplever du ofte at den information du får fra styregruppemøderne der er det samme som står i surveyxcat svarerne

Informant: Jov det, jo det lidt af det samme kommer igen.

Niki: Men kommer der stadig, men kommer der stadig noget nyt, eller er det bare...?

Informant: Jov jov der kommer da noget nyt. For til styregruppemøderne er der en repræsentant fra gruppen. Og selvom jeg siger at jeg som sådan ikke er interesseret i din mening, i skal på forhånd snakke i gruppen omkring de ting. Og så tager du de ting med. Men det kan da selvfølgelig godt være at nogen bliver så påvirket af det at vedkommende snakker meget ud fra han eller hun selv synes. Så på den måde kan man godt se i surveyxact at dem der ikke har været med til styregruppe møderne de kommer med nogle inputs. Jeg skal lige have lidt vand... *Host* *Host*

Informant: Så der kommer nye ting også, men generelt kan man godt se sammenhænge, eller heldigvis.

Simon: Får de repræsentanter nogle retningslinjer og rammer om hvilke ting de skal snakke med deres gruppe om eller er det bare generelt?

Informant: Jo de får en uge inden møde, der får de en dagsorden. Og så står der, og der står stor set standard dagsorden. Og så skal en af studerende skrive referat. Og så der gennemgang af de forskellige kurser og hvordan går det med projekt og projekt arbejde. Hvordan er samarbejde med vejleder. Og det, det er standard. Så man løbende kan holde øje med er der noget vi skal tage fat i.

C.1. Interview af Informant 3

Simon: Føler du, eller oplever du at alle de repræsentater for sagt alt hvad de ligesom har taget med til mødet, eller...?

Informant: Jo jo det føler jeg. Fordi de er gode og de fleste har også noter med, om hvad de har sagt i grupperne. Og nu har vi sagt at når man har 20-25 studerende så er der måske også 3-4 grupper 5 måske. Nogengange 6. det er også overskueligt, også hvis man siger nu får den ene gruppe lov til at, til sidst er der ikke så mange nye ting mere. Så generelt er det ret enige om noget fungerer eller ikke fungerer.

Niki: Synes du der er noget ehmm noget at fortrække i at ved styregruppe møderne at det er repræsentat, hvor man ligesom tager snakken med sin gruppe og så får en opsamling og så det præsenterer man for dig. Synes du der er en værdi i det i forhold til de enkle svar? Så er der noget der?

Informant: Nej jeg synes såmænd det er godt at de studerende også i grupperne også får lov til at snakke om det hvor vi som underviser eller som semesterkoordinater ikke er med, for jeg er ret sikker på at når vi ikke er med at der bliver snakket og sagt andre ting end hvis vi som underviser er tilstede. Nu er jeg typisk også, og vejleder og man er semester koordinator. Så jeg tror da det er en god måde.

Niki: Nu tænkte jeg mere i forhold til survey exact svarende. Der er det jo den enkle elev der svarer og til styregruppemøderne der har man ligesom diskuteret tingende.

Informant: Jo på den måde skulle det gerne supplere det rigtigt godt. Men jo især ved studerende eller hvis grupperne ikke har snakket sammen inden de kommer til styregruppe mødet. Så kommer der selvfølgelig nogle inputs senere fra dem som føler måske at de har læst referat fra møderne at de ikke er blevet hørt, eller jeg synes faktisk at det er meget vigtigt. Fordi i princippet er det da et kompromi med gruppen, hvad kan vi fælles sige? Og nogle gange er mange bare uenig hvor den ene synes noget fungere og andre syntes ikke det fungere.

Simon og Niki: Ehmm (Bekræftende)

Informant: Ehmm (Bekræftende)

Simon: Så der kommer surveyexact ligesom til at supplere dem i form af at den enkle får lov til at få sin mening hørt?

Informant: Ja

Simon: Okay.

Simon: Hvor... Altså nu siger du at du har ikke særlig mange studerende, hvor

C.1. Interview af Informant 3

stort et stykke arbejde er det at udarbejde dette første udkast til denne rapport?

Informant: Man bliver bedre og bedre. Først var der, ja, det ved jeg ikke hvor lang tid bruger man det. Man bruger, man skal læse og samle de forskellige elementer jo man bruger da 1-2 timer.

Simon: Så det ja, du oplever det ikke som et omfattende stykke arbejde at skulle danne dig et overblik? Informant: Hvis man har mange af sådan nogle opgaver der tager 1-2 timer så alt i alt så føles det mere, men isoleret set ser på den del, så er det ikke så stor en... Men nu igen nu har jeg ikke haft nogle hvor der virkelig er nogle store problemer så på den måde er det et mere et. Det skal ikke misforstås sådan når i citerer så skal i lige have det i sammenhæng. At det er en administrativ øvelse man den nemmere at hvis det hele kører. Så det nemmere at skrive end hvis der er nogle store ting hvor man virkelig skal snakke med nogle underviser. En eller to gange hvor man virkelig skal snakke, og så tager det lang tid hvis man virkelig skal have fat i en underviser der ikke har gjort hvad vedkommende har skulle gøre, så på den måde er der pludseligt mange flere timer man bruger på det, men generelt hvis det hele er godt som det skal og det er 9 ud af 10...

Informant: ... 9 ud af 10 semestre, får sendt en mail ud og så få tilbagemeldinger igen.

Niki: Hvis nu vi levede i en perfekt verden

Informant: ej hvor dejligt

Niki: - med uanede mængder af tid, ressourcer, værktøjer og programmer og alt det der. Hvordan vil den her proces med at lave de her semesterevalueringer se ud for dig? Hvis vi var i den perfekte verden.

Informant: Hvad er den perfekte verden for mig? Jeg synes stadigvæk det er meget vigtigt at man får snakket med de studerende, fordi de inputs... (går i stå) Lige som jeg sagde før: Hvis man på en eller anden måde kunne have de input i nogle bokse, som man så kunne vinge af, hvad der ligesom er vigtig der kommer med i den samlede evaluering. Nu hvor man bare sidder og skriver det der blev sagt til styringsgruppemøder, det man synes der var vigtigt. Det markerer man og kopierer man og så putter man det i et nyt dokument. Så på den måde føles det nogle gange som om, hvorfor skal jeg bruge det? Men der nogle andre der også skal læse det. Så hvis man på en eller anden måde kunne markere det man synes er vigtig fra styringsgruppemøderne, så det kommer med i den samlede evaluering og det samme med surveyxact. Hvis det giver mening.

Simon: Vi havde et møde tidligere i dag (Kurt) og han har jo op til 250 elever, så

C.1. Interview af Informant 3

det skulle være noget der skulle kunne bruges på forskellige niveauer, for ham er det jo uanede mængder af data han skal kigge på, og det er meget uoverskueligt, men der er selvfølgelig nogle andre mangler for semesterkoordinatorer der har færre elever.

Informant: Jaja, derfor behøver jeg heller ikke at tage noter, for så mange, hvis der er max 10 kommentarer udover et kursus udover den der kvantitative vurdering over hvor tilfreds har man været på en skala fra 1 til 5. så er det meget. Men hvis der er 200 der kan kommentere, så kan jeg godt forstå at det er meget større opgave og så kan jeg godt se i den perfekte.. Jeg vil ikke sige jeg lever i en perfekt verden, men i forhold til ham, så tror jeg at jeg har det rigtig godt. Og så må jeg være glad for vi ikke har flere studerende. (alle griner) Men vi vil gerne have flere. (Snak om vi sagtens kan skifte, og at der er en der har skiftet fra software) (alle griner) Men der er stor forskel på hvor mange man har, og det er sådan set ikke med i evalueringen, men fordi vi kun er 25, så kender jeg de fleste studerende, og så er det også nemmere for dem at henvende sig til mig hvis der er noget. Men hvis man nu er 1 ud af 200 eller 300, så er der måske også en større barriere at ringe eller skrive til semesterkoordinatoren fordi, vedkommende har jo ikke en chance for at vide hvem vedkommende er. Når vi afholder eksamen så er alle gruppe tilstede, så alle ved hvem jeg er (Noget om at huske navne (29:40-29:48)), efter eksamen plejer jeg at kunne huske navne.

Niki: Du nævnte noget med tænkte tilbage hvis der var en kommentar der ligesom går igen. Kigger du nogensinde aktivt tilbage på svarene, er det noget du reelt set går tilbage og kigger på?

Informant: Nej, jeg stoler stadig væk på min hukommelse, jeg kan huske vi har haft nogle gange måske, jo vi havde et bestemt kursus at der var nogle problemer man skal tage fat i, men det komme så på de der planlægningsmøder at vi lige tager det med igen. Generelt kan jeg godt huske. Og så har vi lige haft studieordningsrevisoin. så som sagt det hvor vi kun har haft det 1 – 2 år, så skal vi ikke huske så langt, hvis det nu var 10 - 15 år tilbage, så var det måske lidt...

Simon: Som jeg hører det på dig, så er der ikke særlig meget manuelt arbejde i at udarbejde de her og kigge på det, så den automatisering, hvis man et eller andet sted i processen skulle automatisere, det skulle være at gøre det mere præcis for dig. Når du læser dem, så læser du måske et eller andet så du få lov til at klikke dem af, og så kommer det automatisk med ind og bliver en del af din rapport.

Informant: så kunne det måske blive nemmere at sammenligne på tværs af årgange.

Simon: så du kunne gå ind og kigge på hvilke du har udvalgt fra tidligere semestre du synes var særligt vigtige.

C.1. Interview af Informant 3

Informant: så skulle man, men ulempen er lidt, at så misser man nogle standardsvar. Svarende i de kvalitative inputs fra år til år er ikke de samme, så jeg tænker det må også være svært, at tror det hele er... jeg er ikke imod automatisering af tingene, men hvis man så har været. Det skal heller ikke være så man kun kan udfylde nogle kasser, fordi så har man altid noget hvor man har noget andet. Og så har man det samme som man har nu, hvor man kan komme med nogle inputs, hvis der er nogle standard svar som de skal tænke over, så skal de gå hjem og tænke kunne det være.

Simon: Et af de forslag vi har fået fra et af de tidligere møder, det var noget med at når det er du bliver stillet et spørgsmål, og så skal du svare på det f.eks. "hvordan har gruppearbejdet været på det bestemt kursus" så en elev ser at han skal svare på, så tænker han på en ting der har været mindre godt, og så skriver han det ind og så er han egentlig færdig med den boks nu, selvom hvis han havde siddet 5 minutter længere så var han kommet i tanke om 2-3 ting, andre ting, hvor man måske kunne gå ind og finde nogle tidligere elevers kommentarer...

Simon: Hvor man måske kunne gå ind og finde nogle tidligere elevers kommentarer eller kommentarer som er givet fra tidligere semestre på det samme område her med det kursus her. Og det jeg så mener er at man kan stille sig enig med dem.

Informant: Ahhh, jo det kan man såmænd godt. Vi havde et kursus hvor det var nogle fejl i den facitliste som man fik til opgaveregningen. Og så kunne man i princippet skrive det ned i et eller andet kvalitetskompendium, og hvis man så oplever igen at der var mange fejl og man så kunne krydse af, så tænker man okay. Så kunne man sige okay, "har du opdateret den, og er den nu som den skal være". Men igen, fordi vi har ikke flere studerende end vi har, så derfor er der nogle ting som der virkelig bonner ud, så kan man huske det. Hvis man har 200 og har så mange ting. Men det kunne jo egentlig også være en god ting, at man lige kunne evaluere på de ting.

Simon: Det kunne man gøre på forskellige måder, hvordan man udvælger hvilke det skal være, men det kan måske være nogle af dem du siger er specielt relevante for et semester, eller året efter på samme semester, er det nogle af disse der bliver vist, som elever kan stille sig enige i. Så kan du sige at der måske er 15 ud af 20 der er enige i et bestemt udsagn.

Informant: Jeg har gjort det en lille smule nu du siger det, med de fejl der var i den der facitliste, hvor jeg så spurgte ind på et styringsgruppemøde, om der var noget omkring opgaveregning. Hvis ikke der var blevet gjort noget ved det var jeg da 100% sikker på at de ville sige der var rigtig mange fejl i den facitliste.

Simon: Sådan lidt mere opfølgende.

C.1. Interview af Informant 3

Informant: Ja men det kunne være godt, og det sikrer jo selvfølgelig også helt sikker kvaliteten igen.

Simon: At der ligesom bliver fulgt op på de ting der bliver kommenteret

Informant: Så kunne man hvertfald udvælge nogen, det er også det studienævnet gør når du læser semesterevalueringsrapporter, så markerer studielederen også nogle ting som det mener de er vigtigt at vi følger op på. Og når vi så er til studienævnsmøder får vi afvide at "det og det" skal i lige få snakket om når i er til planlægningsmøder. Men det kunne man sådan set også godt gøre på de studerende.

Simon: Så man sikrer sig at det er blevet rettet op på på en tilfredsstillende måde, fra deres synsvinkel.

Informant: For det kan også være at nogen gange har man en årgang der er tilfreds med det hele, og så næste årgang så er alt dårligt, eller nej i forstår godt hvad jeg mener. Der er selvfølgelig lidt nogle bølger.

Simon: Nej når i ikke er flere elever så skal der heller ikke særlig store udslag til for at det kommer til at virke meget ekstremt.

Informant: Nej for jeg gør det aldrig op i 10% er utilfredse, for kun få skal sige et eller andet. Jeg har haft et kursus engang med kvantitativ statistik, hvor kan vise tal på en meget kreativ måde. Lige hvordan man har lyst til. Hvis der nu var et bestemt kursus jeg sagde "det skal vi da af med" så kunne jeg bare lige lege lidt med tallene, eller præsentere dem på en måde. Der er nogle der tænker, bare vi har tal så har vi hardcore facts.

Simon: Det er klart at der lige skal et øje ind og og kigge på de kvalitative også, kvantitative kan ikke stå alene.

Informant: Nej det er kombinationen der som styrker det hele. At man får forklaring for hvorfor har du sagt tilfreds eller utilfreds.

Niki: Når du nu får de udfyldte evalueringer sendt til dig, når de studerende har udfyldt den, hvilken format kommer de så i? Kan du se at den her elev har svaret de her 5 ting. Altså ikke navnet.

Informant: Nej, jeg kan heller ikke følge at personen ikke har været meget tilfreds, her, og her. Jeg får bare at 25 studerende har svaret, og så mange er tilfreds eller utilfreds, og så kom det i en række. Som kan ikke have den der kryds. Fordi det kunne være rart hvis man kunne se eller følge hvis der var en der har svaret enten

C.1. Interview af Informant 3

meget positivt, eller meget negativt.

Simon: Altså så du kunne godt tænke dig det der med at du får alle denne ene elevs kommentarer til de forskellige spørgsmål på den samme række, så du kunne gå hen og sammenligne.

Informant: Selvom man kun har 25 studerende så bliver alligevel meget data, og så har vi virkelig brug for jeres hjælp til at strukturere det på en god måde. Fordi nu har jeg sådan en blogger med inputs. Men så kan man følge dem, eller hvordan man nu vil. Fordi der skal selvfølgelig helt sikkert også være et formål med hvor detaljeret vi skal gå, eller på et mere overordnet plan, hvor de generelt har været tilfreds.

Niki: Har du oplevet tidligere at kommentarer som "se ovenstående" hvor folk refererer til tidligere svar?

Informant: Næh

Niki: Vi har nemlig fået noget data hvor folk bruger dette trick, hvor den forbindelse mister man jo så.

Informant: Nej det har jeg ikke. Det eneste er at vi har BIM uddannelse i København, hvor vi opdeler det en smule. En stor del af deres feedback kan jeg ikke bruge til noget (Problemer med lokaler), fordi vi har ikke des samme problematikker som de har. Det er mest indholdsmæssigt med kurser, hvor det kan være ligemeget om det er studerende fra Aalborg eller København. På den måde er det rart, men så skulle man virkelig tænke sig om hvor specifik skulle det være, og hvornår giver det mening, eller bliver det for detaljeret. Fordi så kommer man lynhurtigt til at bruge rigtig mange timer på at analysere data.

Bilag D

Funktionelle Krav

1. Indkapsle relevant data

Formål: At isolere relevant data på serveren, så kun folk med tilladelsen til det, har adgang til specifikke evaluerings data. F.eks. semesterkordinatore kan kun se evalueringer for deres egne hold.

Dette er et nødvendigt krav, og det forventes ikke at blive ændret i produktets livscyklus.

Input: Brugernavnet vil bestemme hvilket adgangsniveau brugeren vil have. Adgangskode til at bekræfte at de har adgang til den bruger.

Funktion: Input kontrolleres til at være gyldigt. Ved ugyldigt input vil der ikke være et output. Ugyldigt input kan være indhold af ulovlige tegn, eller forkert kombination af brugernavn og adgangskode. Gyldigt input vil resultere i output.

Output: Funktionens output er evalueringer og analyser som er relevante for brugeren.

2. Adskille Hold og Data

Formål: At holde evalueringer adskilt så brugeren let kan navigere gennem alle evalueringer for et bestemt hold.

Input: Analyseresultat og evalueringer.

Funktion: Data sorteres efter hvilket hold dens feedback hører til.

Output: En sorteret liste af evalueringer efter hvilket hold det omhandler.

3. Kategorisering af feedback: Gengående emner

Formål: At fremhæve emner der fremtræder hyppigt i kommentarer henvendt til et givet modul.

Input: Kommentarer til et givet modul.

Funktion: Kommentarerne sammenlignes efter betydning og ensformethed.

Output: Ord og udtryk der går igen i kommentarerne, og hvilke kommentarer denne konklusion er draget fra.

4. Tilgå Evalueringer

Formål: At kunne læse evalueringer uden sortering eller analyse.

Input: En semesterevaluering.

Funktion: Indsætning af input på en brugerflade.

Output: Semesterevalueringen vist på en brugerflade.

5. Check serverforbindelse

Formål: At sikre om der er forbindelse eller ej til serveren.

Input: Intet.

Funktion: Send en anmodning til serveren, og konkluder om der er forbindelse baseret på dens respons, eller mangel derpå.

Output: Meddelelse om hvorvidt der er forbindelse til serveren.

6. Se alt feedback fra én studerende, angående ét semester, sammen

Formål: At vise besvarelsen fra en given studerende til et givet semester.

Input: Identifikationsnummer af en studerende, hvilket hold de tilhører, og semesterevalueringen for det hold.

Funktion: Søge semesterevalueringen igennem efter besvarelsen med den givne studerendes id. Indsættelse af besvarelsen i en brugerflade.

Output: Besvarelsen fra den givne studerende, præsenteret på en brugerflade.

7. Markering af kommentar

Formål: At lade brugeren selv udvælge enkelte kommentarer de vil kunne finde tilbage til. Der skal være flere forskellige former for markeringer til forskellige formål.

Input: En kommentar og en markeringstype.

Funktion: Gem kommentaren der bliver markeret, og hvilken markeringstype den er markeret med.

Output: Bekræftelse af at markeringen er gemt, eller advarsel om at det fejlede.

8. Sammenligning af årgange

Formål: Programmet skal kunne sammenligne feedback over en tidsperiode, for at tillade semesterkordinatore lettere at få en generel forståelse om hvordan et kursus har udviklet sig over flere år.

Input: Analyse fra flere årgange.

Funktion: Analyserne vil sammenlignes i forhold til det år de repræsenterer.

Output: Et sæt af data som repræsenterer forskelle mellem årgange.

9. Generalisering ved brug af Sentimentanalyse

Formål: Omdanne store mængder kvalitativ feedback til kvantitative repræsentationer af den samme data automatisk. Det vil tillade semesterkordinatore en bedre forståelse af den generelle attitude om et kursus i dybere detalje. For at opnå dette kræves der flere underfunktioner.

Input: Kvalitative evalueringer.

Funktion: Funktionerne vil fokusere på at først forberede input data, og derefter bruge ML funktioner til at uddrage kvantitativt data.

Output: Et generaliseret kvantitativt datasæt baseret på kvalitative evalueringer.

10. Finde det generelle sentiment

Formål: Programmet skal kunne uddrage det generelle sentiment i sæt af evalueringer. Dette er den essentielle og centrale del af at kunne omdanne kvalitativ tekst til kvantitative data.

Input: Et sæt af evalueringer.

Funktion: En trænet ML algoritme bruges til at uddrage sentimentet i sættet af evalueringer.

Output: Det generelle sentiment (Mellem positiv og negativ)

11. Genkendelse af Nøgleord

Formål: Specifikke ord har en stor betydning for, hvordan andre ord skal forstås. Eksempelvis ordet "ikke".

Input: Ét kvalitativ svar.

Funktion: Bestemte nøgleord vil uddrages fra sætninger i evalueringen.

Output: Centrale nøgleord i svar.

12. Fjernelse af Tegnsætning

Formål: Tegnsætning har ikke en central betydning for hvordan sætninger skal forstås i forhold til hvilket sentiment det indeholder. Det vil derfor hjælpe ML delen af programmet hvis disse tegn ikke skal tages i betragtning.

Input: Ét kvalitativ svar.

Funktion: Funktionen vil fjerne tegn som komma, punktum, og lignende.

Output: Ét kvalitativ svar uden tegnsætning.

13. Fjernelse af Stopord

Formål: Nogle ord har ingen betydning for et sætnings budskab, og derfor kan de blive filtreret ud af datasættet. Dette vil hjælpe træningen af ML.

Input: Ét kvalitativ svar.

Funktion: Stopord og fylde ord vil fjernes fra evalueringen.

Output: Ét kvalitativ svar uden stopord.

14. Omdannelse til små bogstaver

Formål: Store og små bogstaver har minimal betydning for en sætnings budskab, og derfor bør forskellen mellem dem ignoreres.

Input: Ét kvalitativ svar.

Funktion: Alle bogstaver i input omdannes til små bogstaver.

Output: Ét kvalitativ svar uden store bogstaver.

15. Rettelse af Stavefejl

Formål: Stavefejl kan gøre data mindre præcist, da programmer ikke vil have samme evne til at tyde stavefejl som mennesker har. Derfor ville det øge programets effektivitet, hvis det har en evne til at foretage rettelser på mindre stavefejl.

Input: Ét kvalitativ svar.

Funktion: Ord som ikke genkendes, vil forsøges at omdannes til mest lignende og passende ord som kendes i ordbogen.

Output: Ét kvalitativ svar med reducerede stavefejl.

Bilag E

Stopord

Nedenstående tabel tager udgangspunkt i listen fra [29].

ad	af	aldrig	alle	øvrigt	hvor	hvorimod
alt	anden	andet	andre	vær	hvorfra	hvorhen
at	bare	begge	blev	vore	hvilkes	hvorefter
blive	bliver	da	de	via	herpå	hvilke
dem	dit	denne	der	undtagen	heri	hermed
deres	det	dette	dig	udover	henover	herefter
din	disse	dit	dog	uden	heller	hen
du	efter	ej	eller	længere	gørende	hel
en	end	ene	eneste	ligesom	gør	gøre
enhver	er	et	far	lavet	gjorde	gjort
fem	fik	fire	flere	lave	først	gennem
fleste	for	fordi	forrige	langs	flest	foran
fra	få	får	før	indtil	ens	enten
god	godt	ham	han	imod	ellers	endnu
hans	har	havde	have	imens	derved	egen
hej	helt	hende	hendes	imellem	dermed	derpå
her	hos	hun	hvad	igennem	derfra	deri
hvem	hver	hvilken	hvis	igen	derefter	derfor
hvor	hvordan	hvorfor	hvornår	hvorved	bør	dens
i	ind	ingen	intet	tit	blandt	bude
ja	jeg	jer	jeres	tilbage	altid	bag
jo	kan	kom	komme	tidligere	allerede	alligevel
kommer	kun	kunne	lad	temmelig	alene	mellem
lav	lidt	lige	lille	således	være	været
man	mand	mange	med	syntes	vor	mest
meget	men	mens	mere	synes	vil	ville
mig	min	mine	mit	stadig	ved	mindre
mod	må	ned	nej	siden	under	var
ni	nogen	noget	nogle	senere	tre	mindst
nu	ny	nyt	når	selvom	til	to
nær	næste	næsten	og	sammen	thi	måske
også	om	op	os	samme	tag	tage
otte	over	på	se	overalt	så	nemlig
seks	selv	ser	ses	omkring	store	syv
sig	sige	sin	sine	nok	vores	vi
sit	skal	skulle	som	nogensinde	ud	ti
stor	sådan					

Bilag F

Usability test opgaver

1. Log ind.
Navn er *Fornavn*, og koden findes ikke.
2. Opret nyt semester.
Filen findes på skrivebordet og hedder: Testfil.
Semestret skal ende med at hedde: Testsemester7 2011-Efterår.
3. Gå ind på 'Byggeri 2018-AUTUMN'
4. Se sentiment for 'Byggeri 2018-AUTUMN'
5. Gå ind på 'Biologi 2018-AUTUMN'
6. Foretag en markering af en kommentar i 'Biologi 2018-AUTUMN'
7. Sammenlign 'Biologi 2018-AUTUMN' modul: Almen biologi med 'Biologi 2018-AUTUMN' modul: Analyse 2
8. Foretag en filtrering af 'Biologi 2018-AUTUMN's rådata.
Vis data for modulet Analyse.
Fravælg at kunne se 'Respondant' kolonnen
9. Log ud

Bilag G

Brugervejledning

Login

Udfyld loginoplysninger :

Brugernavn: AAU e-mail

Adgangskode: Kode til e-mail

Opret semester

For at oprette et nyt semester skal der i separate felter udfyldes dets navn, semester-nummer, årstal, halvår. I det sidste felt udvælges spørgeskemaresultater fra SurveyXact på .csv format.

Semesternavn boksen giver brugeren mulighed for at indtaste navn på semestret, eller vælge et tidligere brugt navn.

Semesternummer vælges hvilket nummer semestret er.

Boksen med **årstal** beskriver hvilket år dette specifikke semester blev/bliver afholdt. Dertil vælges om semestret blev/bliver afholdt om **foråret eller efteråret**.

Vælg spørgeskema data udvælges den fil, hvor spørgeskemaresultaterne er i.

Endelig Semesternavn viser hvordan semestret bliver repræsenteret i 'mine semestre'

Mine semestre

Her vælges et semester at arbejde med.

Sentiment

Her ses sentimentanalysen.

Sammenlign

Her kan sentimentanalysen sammenlignes på tværs af år, semestre og moduler.

Sammenlign: Vælg et modul fra det valgte semester.

Med: Valgte moduler fra valgte semestre, til sammenligning

Find semester: Laver en popup, hvori der vælges hvad der skal sammenlignes med.

Vælg semester: Vælg et semester.

Vælg modul: Vælg projekt eller et kursus.

Tilføj: Tilføj det valgte til sammenligningslisten.

Sammenlign: Sammenlign de valgte.

Farve: Skift farvene på søjlerne.

Markering

Her ses alle kommentarer med en markering.

Alle moduler: Vis markerede kommentarer fra dette modul.

Alle spørgsmål: Vis markerede kommentarer af dette spørgsmål.

Farve: Vis markerede kommentarer med denne farve.

Søg: Søg med tekst.

Rådata

Her kan alt data ses.

Dobbelt klik på en kommentar: denne vil blive åbnet, og derfra kan hele kommentaren læses, til højre kan kommentaren markeres med en farve.

Højreklik på en kommentar: mulighed for at vælge farve, og dermed markering.

Mine semestre

Sender brugeren tilbage til semesteroversigten.