

Patrick Holmquist Andreassen

pandr20@student.sdu.dk
497508

Stefan Profft Larsen

stefa20@student.sdu.dk
497743

Johan Brockstedt

jobro19@student.sdu.dk
474527

Vejleder:

Rune kammersgaard Gregsen

rkgr@mmmi.sdu.dk

Troels kaldau

trkal20@student.sdu.dk
497065

Daniel Smidstrup

dasmi20@student.sdu.dk
495433

Emil Madsen

emima20@student.sdu.dk
496459

HOLD C-10



SDU

Rapport

Semesterprojekt
2. semester foråret 2021

University of Southern Denmark
Civilingeniør i Software Engineering

Det Tekniske Fakultet
Mærsk Mc-Kinney Møller Instituttet
Campusvej 55, 5230 Odense M

Kursuskode: SE2-PRO

Projektperiode: uge 4 - 22

TV 2 Krediteringsprogram

Titelblad

TV 2 Krediteringsprogram

Semesterprojekt – 2. semester forår 2021

Kursuskode: SE2-PRO

Projektperiode: 1. februar – 28. Maj 2021

Projektgruppe: C-10

Vejleder: Rune Kammersgaard Gregsen

Civilingeniør i Software Engineering

Syddansk Universitet

Det tekniske Fakultet

Mærsk Mc-Kinney Møller Instituttet

Forfattere

Patrick Holmquist Andreasen

pandr20@student.sdu.dk

Johan Brockstedt

jobro19@student.sdu.dk

Troels Kaldau

trkal20@student.sdu.dk

Stefan Profft Larsen

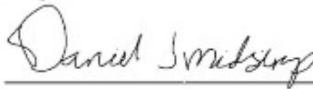
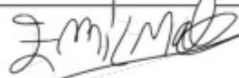
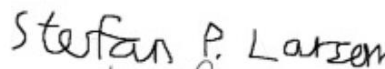
stefa20@student.sdu.dk

Emil Madsen

emima20@student.sdu.dk

Daniel Smidstrup

dasmi20@student.sdu.dk



Opgave

Opgave fylder 54 sider*

*Eksklusive formalia, referenceliste og bilag.

TV 2 Krediteringsprogram

Resume

Dette projekt er baseret på en projektcasen udleveret af SDU og udformet af TV 2 Koncernen. Casen omhandler TV 2's behov for et krediteringssystem, der kan opbevare og vise krediteringer fra flow tv, så disse ikke skal vises efter endte programmer. Dette kan ifølge casen spare TV 2 for op mod 60 mio. DKK. Herudfra er problemformuleringen opstillet:

“Hvordan udvikles et krediteringssystem, der overholder de nuværende regler for kreditering, gør det simpelt for producenter at tilføje og ændre krediteringer, gør det simpelt for TV 2 at verificere og videregende krediteringer, og gør det simpelt for seerne at se krediteringer, med minimal omkostning af sendetid?”

Projektet afgrænser løsningen til at være baseret på Java, have en Relational database og fungere som én samlet applikation.

Til løsning af projektet blev benyttet en kombination af Unified Process og SCRUM. Der blev udarbejdet et inceptionsdokument til at skabe overblik og fastsætte forventninger til projektet. Herunder blev brugsmønstre udviklet, og senere detaljeret og realiseret. Dette, sammen med databasedesignet, dækkede fundamentet for implementeringen af projektet, der senere blev testet med udviklingstest, udgivelsestest og brugertest.

Resultat

Der blev i inceptionsfasen identificeret 3 aktører, 12 funktionelle og 6 ikke-funktionelle krav, 4 tekniske og 6 sociale risici. Herudfra blev lavet et brugsmønsterdiagram, 3 detaljerede brugsmønstre, løsningsforslag til identificerede risici, samt en MoSCoW analyse og prioritering af krav.

I analyse og krav blev flere brugsmønstre realiseret, der blev dannet et pakkediagram, et EER-diagram, samt et databaseklassediagram. På baggrund af designet, blev 10 ud af 12 funktionelle krav implementeret. Disse blev herefter testet med en udgivelsestest, samt en brugertest.

Konklusion

Ud fra resultaterne i inceptionsfasen, blev kravene for projektet vurderet til at være overkommelige og de identificerede risici kunne undgås.

Vi fandt ud af, at funktionskandidaterne som vi fik dannet gennem Verb/Noun analysen, blev overflødige, da adfærden for systemet ville blive nærmere undersøgt i den dynamiske del af analysen.

Under implementeringen, udarbejdede vi logikken for klasserne og flere kontrollere blev implementeret. Det var først i 2. iteration, at vi tog hånd om kryptering af følsomme oplysninger. Test af implementeringen blev testet ved hjælp af unit tests og system tests. Gennem en brugertest fik vi indblik i systemets brugervenlighed, men havde et for spinkelt grundlag til at konkludere, hvorvidt problemformuleringen blev opfyldt til fulde.

TV 2 Krediteringsprogram

Forord

Hensigten med rapporten er at gøre rede for vores arbejde i 1. iteration, med det lokale filsystem i fokus, arbejdet i 2. iteration, med implementeringen af databasen i fokus, samt alt arbejdet før og efter iterationerne. Censor og vejleder er målgruppen for rapporten og hermed er de i fokus for bearbejdningen af rapporten. Rapporten er skrevet ud fra en case, der blev givet til os fra TV 2, der omhandler udarbejdelsen af et krediteringssystem. Krediteringssystemet skal kunne vise forskellige krediteringer for TV 2's programmer. Programmerne skal vises på forskellige måder for både det offentlige og for TV 2 selv. TV 2 skal have mulighed for at tilføje, fjerne og ændre i disse krediteringer. Der vil hermed laves et krediteringsprogram med en offentlig bruger side, en TV 2 administrator side og en producent side.

Vi vil gerne sige tak til vores semesterklasse for deres respons til vores projekt ved projektseminaret, tak til Rune Kammersgaard Gregsen, for hans vejledning og hjælp til udviklingen af både systemet og rapporten, og tak til Anton Herløv Christensen for at tage sig tid til at teste vores program.

TV 2 Krediteringsprogram

Indhold

Titelblad TV 2 Krediteringsprogram.....	i
Resume.....	ii
Forord.....	iii
Læsevejledning.....	vi
Redaktionelt.....	vii
Skriveproces.....	vii
Ansvarsområder.....	vii
Rettigheder.....	viii
1. Indledning *.....	1
1.1. Projektets rammer.....	1
1.2. Formål og mål for projektet.....	1
1.3. Problemformulering.....	1
1.4. Afgrænsninger.....	2
2. Faglig vidensgrundlag *.....	3
2.1. Begrebsdefinitioner.....	3
2.2. Besvarelse af underspørgsmål.....	3
2.3. Vidensgrundlag for metoder.....	5
2.4. værktøjer.....	5
3. Metode og planlægning.....	7
3.1. Metode.....	7
3.1.1. Anvendte procesmodeller.....	7
3.1.2. Problemanalyse.....	7
3.1.3. Krav.....	7
3.1.4. Prioritering.....	8
3.1.5. Analyse.....	8
3.1.6. Design.....	9
3.1.7. Implementering.....	9
3.1.8. Test.....	10
3.2. Tidsplan.....	11
3.3. Planlægning.....	11
3.4. Scrum.....	14
4. Hovedtekst.....	15
4.1. Overordnede krav *.....	15
4.1.1. Aktørliste.....	15
4.1.2. Brugsmønsterliste.....	16
4.1.3. Brugsmønsterdiagram.....	18
4.1.4. Supplerende krav.....	19
4.2. Detaljerede krav.....	20
4.2.1. Detaljerede brugsmønstre.....	20
4.2.2. Supplerende krav.....	21
4.3. Analyse.....	23
4.4. Design.....	26
4.4.1. Software arkitektur.....	26
4.4.2. Design beslutninger.....	27
4.4.3. Detaljeret design.....	28
4.5. Databasedesign.....	35
4.5.1. EER.....	35
4.5.2. Normalisering.....	35
4.6. Implementering.....	37
4.6.1. Controllers.....	37
4.6.2. Viewmodels.....	37
4.6.3. SQL mapping.....	38
4.6.4. Hashing.....	38

TV 2 Krediteringsprogram

4.6.5.	Søgelogik.....	39
4.6.6.	Interfaces	40
4.7.	Test.....	41
4.7.1.	Udviklingstest	41
4.7.2.	Udgivelsestest	42
4.7.3.	Brugertest.....	42
5.	Diskussion.....	43
5.1.	Brugernormalisering.....	43
5.2.	Data hentning.....	43
5.3.	Det 4. lag.....	43
5.4.	Objekt mapping.....	44
6.	Konklusion.....	45
6.1.	Inceptionsfasen	45
6.2.	Elaborationsfasen	45
6.2.1.	Analyse og design	45
6.2.2.	Implementering og test	46
6.3.	Besvarelse af problemformulering.....	46
7.	Perspektivering	47
7.1.	Privat bruger	47
7.2.	Krediterings bruger.....	47
7.3.	Eksporterings API	47
7.4.	Krediteringsvalidering.....	48
8.	Procesevaluering.....	49
8.1.	fælles vurdering.....	49
8.1.5.	Formelt review og Midtvejsseminar	50
8.2.	Individuel vurdering.....	51
8.2.5.	Patricks vurdering.....	53
9.	Referenceliste.....	55
Bilag.....		56
A.	Kildekode	56
B.	Brugervejledning	57
C.	Samarbejdsaftale	59
D.	Projektlog.....	64
E.	Rapportkontrolskema	65
F.	Scrum-Buts.....	73
G.	Scrum retrospective og sprint review for anden sprint.....	74
H.	Anden iteration design.....	75
I.	UI-design	77
J.	Analyse og design af "Tilføj kreditering" og "ændre kreditering"	81
K.	Inceptionsdokument	88
L.	Udsnit af udført udgivelsestest.....	121
M.	Brugertest	123
N.	Udklip af controller liste.....	126

Læsevejledning

Den foretrukne måde at læse rapporten på, er først at gøre dig fortrolig med rapporten gennem resumeet. Når den er læst, kan man kigge indholdsfortegnelsen igennem, for at få en forståelse for, hvad vi præcist i kommer ind på i denne rapport. Herefter skal forord læses, derefter kan man starte fra kapitel 1. *Indledning* og fortsætte derfra. Rapporten er heraf skrevet, så den gennemgår det grundlæggende forløb for semesterprojektet. Der bliver først introduceret problemet og problemformuleringen, derefter gennemgås grundlæggende metoder brugt, samt planlægningen af projektet. Derefter gennemgås krav, analyse, design og implementering for projektet. Efter en beskrivelse af hovedforløbet, diskuteres problemet, der konkluderes og til sidst kommer perspektivering. Der afsluttes med en procesevaluering, hvor man kan læse gruppens og de individuelle vurderinger af processen.

Før rapporten blev udformet, blev der udarbejdet et inceptionsdokument (se bilag K). Mange beskrivelser fra inceptionsdokumentet er stadig relevante og er derfor medtaget i rapporten. De afsnit eller elementer, der primært er baseret på inceptionsdokumentet, mærkes med (*) i overskriften. Der er dog ingen af disse afsnit der er foruden udladelser, omskrivninger og eller tilføjelser. Mærkes en overskrift med (*) er over 50% af indholdet under overskriften kopieret fra inceptionsdokumentet.

TV 2 Krediteringsprogram

Redaktionelt

Skriveproces

I starten af projektet blev et inceptionsdokument udformet. Herunder blev casen, problemformuleringen, det faglige vidensgrundlag, de kendte metoder, overordnet kravanalyse, prioritering og risici analyse bearbejdet. Efter dette blev forskellige mindre formelle dokumenter udarbejdet primært til analyse og design. Disse dokumenter indeholdt tabeller, diagrammer, analyser og refleksioner. I midten af projektets anden iteration begyndte 1 til 2 gruppemedlemmer at opsætte den endelige rapport. De begyndte at undersøge hvad diverse afsnit skulle indeholde, og de begyndte herefter så småt at skrive, det der, på daværende tidspunkt, kunne skrives om. I weekenden uge 20 blev koden færdiggjort, og hele gruppen rettede arbejdet mod rapporten. Der blev oprettet en tabel, hvor de forskellige ansvarsområder blev uddelt, og rapporten er skrevet ud fra dette skema.

Ansvarsområder

Afsnit	Ansvarlig	Bidrag fra	Kontrolleres af	Korrektur af
Forside	Troels		Johan	Johan
Titelblad	Johan		Daniel	Patrick
Resume	Johan	Troels	Patrick	Emil
Forord	Johan		Daniel	Troels
Læsevejledning	Johan	Troels	Daniel	Patrick
Redaktionelt	Troels		Johan	Daniel
Indledning	Stefan	Daniel	Troels	Patrick
Faglig vidensgrundlag	Troels	Daniel	Johan	Emil
Metode	Troels		Daniel, Emil	Stefan
Planlægning og scrum	Daniel		Johan	Troels
Krav	Stefan	Troels	Emil	Daniel
Analyse	Emil	Daniel, Troels	Patrick	Johan
Design	Daniel	Emil	Troels	Johan
Databasesdesign	Johan	Troels	Daniel	Stefan
Implementering	Troels	Daniel	Stefan	Emil
Test	Patrick	Troels, Emil	Stefan	Daniel
Diskussion	Emil	Troels	Daniel	Stefan
Konklusion	Emil	Troels	Troels	Patrick
Perspektivering	Patrick	Troels	Stefan	Troels
Procesevaluering	Stefan	Daniel, Patrick, Troels	Troels	Emil
Bilag	Patrick		Troels	

Tabel 1: Ansvarsområder

TV 2 Krediteringsprogram

Rettigheder

Element	Ophavsmand	Bruges til	Licens
JavaFX	OpenJDK	GUI	GNU General Public Licens
JFoenix	JFoenix 2015-2017	GUI	Apache license 2.0
PostgreSQL	The PostgreSQL Global Development Group	Database	PostgreSQL license
ElephantSQL	84Codes AB	Database hosting	Licens from 84codes AB
JUnit	The JUnit Team	Unittesting	Eclipse Public License 1.0
Spring-JDBC	VMware Inc.	Database Connection	Apache license 2.0
Material Icons	Google	Icons	Apache license 2.0

Tabel 2: Licens referencer

TV 2 Krediteringsprogram

1. Indledning *

Afsnittet afdækker projektets rammer, mål for projektet, og en analyse af projektets centrale problem. Dette skal sikre en fælles forståelse for projektets formål og ønskede resultater, for at sikre en entydig forståelse både for gruppe-medlemmer og læsere af rapporten.

1.1. Projektets rammer

Projektet er baseret på et case-dokument udleveret af SDU og udformet af TV 2. Casen introducerer kort TV 2 og forklarer herefter problemet. Dette opfølges med en outline, af den type løsning, der ønskes, samt krav til den færdige løsning. Problemet, som casen ligger op til, er visning af krediteringer i slutningen af TV 2's programmer. TV 2 afsætter 30 sekunder i slutningen af hvert program til at vise reklamer. De vurderer, at disse 30 sekunder i slutningen af hvert program har en værdi på 60 millioner kr. om året. Løsningen, som casen ligger op til, er et krediteringssystem, der skal digitalisere programmernes krediteringer, så krediteringerne ikke længere behøver at blive vist efter hvert program. I stedet placeres krediteringerne på en frit tilgængeligt online platform, så seere selv kan opsøge dem ved interesse. Casen opsætter en række krav til det endelige produkt. Efter vejledning af SDU (projektudbyderen), er der dog en række af disse krav, der ikke skal efterleves eller arbejdes med. Projektet skal bearbejdes igennem to iterationer. Første iteration benyttes til at udarbejde en simple udgave af projektet, som benytter et lokalt filsystem, og et inceptionsdokument. Anden iteration benyttes til at videreudvikle på første iterations system, hvor systemet skal ændres til at fungere med en relationel database.

1.2. Formål og mål for projektet

Formålet med projektet er at gruppens medlemmer tilegner sig ny viden inden for projektarbejde, krav, analyse, design, implementering og test. Denne viden opnås igennem arbejde med TV 2's case, hvor disse softwareudviklingsmetoder benyttes til at løse dette konkrete problem. Målet for projektet er at få lavet et krediteringsprogram, som overholder TV 2's krav for kreditering. Målet er herved opnået når der er konstrueret et fungerende program og der til sidst er lavet en rapport som dokumenterer processen for det fulde forløb.

1.3. Problemformulering

Problemanalysen definerer problemdomænet, men afdækker ikke de krav opstillet i casen, der definerer løsningsdomænet. Det bliver her beskrevet, at TV 2 ønsker løsningen i form af et krediteringssystem. I systemet skal producenter kunne oprette krediteringer, som seere kan få adgang til. Problemformuleringen er herved baseret både på problemdomænet og løsningsdomænet. De konkrete krav bearbejdes under overordnet kravspecifikation.

“Hvordan udvikles et krediteringssystem, der overholder de nuværende regler for kreditering, gør det simpelt for producenter at tilføje og ændre krediteringer, gør det simpelt for TV 2 at verificere og videresende krediteringer, og gør det simpelt for seerne at se krediteringer, med minimal omkostning af sendetid?”

TV 2 Krediteringsprogram

Underspørgsmål

For at understøtte forståelsen af problemformuleringen, opstilles nogle opfølgende spørgsmål, til at afklarere problemformuleringen yderligere.

- Hvad er kreditering?
- Hvad er reglerne for kreditering?
- Hvad forstås ved "omkostninger af sendetid"?
- Hvordan forstås "simpelt" i problemformuleringens sammenhæng?
- Hvordan skal producenter have mulighed for at behandle deres krediteringer?
- hvordan skal TV 2 have mulighed for at verificere krediteringer?
- Med hvilket formål skal krediteringerne videresendes?
- I hvilket format skal krediteringerne kunne videresendes?

1.4. Afgrænsninger

Problemet lægger op til brugen af en lang række teknologier herunder hjemmesider, mobilapplikationer, servere, udnyttelse af eksternt live-data gennem API's, samt selv at udbyde live-data gennem en eller flere API's. Mange af disse teknologier er dog blevet afgrænset fra SDUs (projektudbyderens) side:

- Systemet skal kun fungere for danske producenter.
- Systemet skal ikke være online, server-side, eller have rest-API.
- Systemet skal være en integreret applikation.
- Systemet fokuserer ikke på at være kompatibelt med andre systemer.
- Systemet vil være en demo, der ikke kan refereres til fra tv-programmer.

Ud fra afgrænsningerne, vil der blive arbejdet med en solitær, offline applikation. Løsningsområdet er derfor primært indskrænket til arbejde med datahåndtering og brugerflade.

TV 2 Krediteringsprogram

2. Faglig vidensgrundlag *

Afsnittet afdækker anvendte begreber, relevante metoder, koncepter, værktøjer, samt besvarer underspørgsmålene stillet til problemformuleringen. Dette skal sikre en fælles forståelse for relevante emner, for alle læsere af rapporten.

2.1. Begrebsdefinitioner

Begreb	Forklaring
Omfakturering	At omstrukturere kode uden at ændre på dets funktionalitet.
PostgreSQL	Relationel database management system.
XML	Extensive Markup Language, en fleksibel måde at dele informationer mellem personer og systemer, både offline og online.
CSV	Comma Separated Values, en kommasepareret fil som kun indeholder tal og bogstaver. God til at samle større mængder tekstbaseret data.
Sprintlog	Prioriteret liste over opgaver der skal udføres.
DBMS	Database Management System, et system der håndterer kommunikation med en database.
Seer	En person der ser et program, i et givent øjeblik.
Tv-udbyder	Et firma der tager penge for at give adgang til et antal kanaler.
Kanal-udbyder	Et firma der håndterer kanalernes indhold.
Producent	Et firma der producerer indholdet, der vises på kanalerne.
UI-design	Design af visuelle software brugergrænseflader.

Tabel 3: Begrebsdefinitioner

2.2. Besvarelse af underspørgsmål

Hvad er kreditering?

Kreditering er dokumentation for, hvem der har bidraget til et produkt. Dette er specielt relevant i produktion af film, serier, artikler, bøger, studier og andre midlertidige projekter. Uden kreditering i disse typer projekter, kan det være besværligt at bevise, hvem der har bidraget med hvad.

Hvad er reglerne for kreditering?

Der er lovkrav om kreditering ved brug af andres kreative materiale¹. Retten til kreditering kan afgives ved aftale. Ved køb, leje eller lån af kreativt materiale, medfølger der derfor en aftale om den aftalte kreditering, der skal overholdes.

¹ Se referencelist 3.

TV 2 Krediteringsprogram

Hvad forstås ved "omkostninger af sendetid"?

Flow-tv har 2 hovedindtægtskilder. Dette er seernes betaling til deres tv-udbydere og reklamer. Seernes betaling til tv-udbyderen er fast og det er derfor kun kanal-udbyderen, der modtager betaling for reklamer. Reklamer er en stor indtægtskilde og producerer indtægt alt efter hvor mange seere der er og hvor meget sendetid der sættes af til at vise reklamer. Hvert minuts sendetid der ikke sendes reklamer, kan derved ses som en omkostning for kanal-udbyderen.

Hvordan forstås "simpelt" i problemformuleringens sammenhæng?

Simpelt kan forstås på 2 primære måder.

- En sekvens med få trin
- En sekvens hvis udførelse er intuitiv og baseret på en logik delt af målgruppen

Gruppen har valgt at lægge vægt på den sidste definition og prioriterer herved intuitive sekvenser, frem for færre trin. Dette kan resultere i en sekvensudførelse med et højere tidsforbrug, men hvor alle kan forstå og gennemføre sekvensen.

Hvordan skal producenter have mulighed for at behandle deres krediteringer?

Producenter skal have mulighed for at tilføje, ændre og slette krediteringer vedrørende programmer de selv ejer. De skal have mulighed for at videregive adgangen til deres krediteringer til andre personer, eg. ansatte.

hvordan skal TV 2 have mulighed for at verificere krediteringer?

TV 2 skal have adgang til krediteringslister udgivet af producenter. De skal have mulighed for at ændre i disse krediteringslister, hvis fejl skulle opstå. De skal ikke have mulighed for helt at fjerne krediteringslister.

Med hvilket formål skal krediteringerne videresendes?

Casen giver udtryk for, at krediteringer har relevans for kreditering Danmark, et forbund der sikrer udbetaling til parter der har bidraget til produktionen af diverse medier.

I hvilket format skal krediteringerne kunne videresendes?

TV 2 krediterer alle de individuelle aktører, der har deltaget i produktionen. TV 2 inddeler dem i 4 kategorier:

- *Individer der har deltaget kreativt i produktionen:* Personerne her skal krediteres individuelt og firmanavne må derfor ikke indgå. Dette gælder individer, der har bidraget med/som:

Tilrettelægger, skuespiller, musiker, manuskriptforfatter, instruktør, komponist, tekstforfatter, kostumer, regissør, rekvisitør, scenograf, kamera, lyd, lys, redigering, producer, associeret producer, billedmixer, tekstere, mm.
- *Sponsorkreditering:* Kategoriseret efter hvad de har sponsoreret med.
- *Kreditering af ydelser af ikke-økonomisk art:* F.eks. "Tak til drengekoret fra Randers".
- *Afslutning af kreditering:* Krediteringen afsluttes med producentens vignet og logo.

TV 2 Krediteringsprogram

2.3. Vidensgrundlag for metoder

UP - Unified Process

UP er en procesmodel, som kan ses som en kombination af en plan drevet og agil model. Den består af 4 faser og et antal aktiviteter. Aktiviteterne strækker sig over faserne og er ikke fastlåste til én fase. Nogle aktiviteter bliver dog primært bearbejdet i én fase hvilket minder om den plan drevne model. Andre udføres gentagende gange, hvilket ligner den agile iterative model. Alle aktiviteter er dog flydende og kan påvirkes i alle faser.

SCRUM-modellen

SCRUM er en agil iterativ model. Den består af et udefineret antal af sprint på ofte 1-4 uger, hvorigennem der tilføjes et antal af features til det endelige produkt. Hvert sprint består af 4 faser: planlægning, udvikling, test og vurdering. SCRUM kan derved ses som en gentagelse af en plan drevet model. Forud for disse sprints vil der være en proces, hvor det overordnede produkt defineres.

MoSCoW-modellen

MoSCoW er en model til prioritering af ideer og funktioner. Det opdeler dem i det, der er nødvendigt (MUST), vigtigt (SHOULD), relevant (COULD) og ikke relevant (WOULD).

UML - Unified Modelling Language

Unified Modelling Language er en standard for udseende og opbygning af diagrammer og bruges til at beskrive struktur og forløb i objekt-orienterede softwaresystemer. UML består overordnet af to typer af diagrammer; strukturdiagrammer og adfærdsdiagrammer. Strukturdiagrammer er diagrammer som klassediagrammet, hvilket beskriver den konkrete struktur for objekt-orienteret softwaresystemer. Adfærdsdiagrammer, som brugsmønster-diagrammet, beskriver hvordan systemets omverden, og objekterne i systemet arbejder sammen.

2.4. værktøjer

GitHub

GitHub er et værktøj til håndtering af kode opbevaring, kode deling og kode sammenfletning. Det er en industristandard indenfor softwareudvikling.

Trello

Trello er en online liste skabende tavle, som benyttes til at opsætte lister med opgaver, for nemt at skabe et overblik over en projektproces.

JavaFX

JavaFX er et kodebibliotek til udvikling af grafiske brugerflader baseret på Java. Der kan benyttes FXML, en version af XML, eller der kan benyttes ren Java.

JFoenix

JFoenix er et kodebibliotek indeholdende nogle prædefinerede udvidelser af JavaFX elementer. Disse udvidelser tilføjer funktionalitet, samt et re-design af UI-elementer.

PostgreSQL

PostgreSQL er et relationelt databasesystem. Det kører på SQL og tillader samtidigt en række interpreted languages.

TV 2 Krediteringsprogram

ElephantSQL

En hostplatform specifikt til hosting af PostgreSQL databaser.

JUnit

Et kodebibliotek med en række funktioner til at opstille automatiserede tests.

Spring-JDBC

Spring-JDBC er en abstraktion af det basale bibliotek JDBC, der håndterer forbindelser mellem databaser og Java programmer.

3. Metode og planlægning

3.1. Metode

Der er brugt flere forskellige metoder igennem processen, for at opnå målet med projektet. Disse metoder beskrives, for at fremføre hvilket formål hver metode har haft, igennem processen, for at konstruere en løsning til TV 2.

3.1.1. Anvendte procesmodeller

Projektet er struktureret ud fra en kombination af 2 procesmodeller: Unified Process (UP) og SCRUM.

SCRUM er en agil metode der effektivt gennemgår aktiviteterne for solid softwareudvikling. Det er en iterativ proces og fokuserer på versionsudvikling. UP er en mere overordnet proces, der har fokus på projektførelsen som en helhed.

De 2 metoder kombineres, så UP anvendes til strukturering af projektets helhed. UP er anvendt i starten til problemanalyse, kravanalyse, samt initial analyse. SCRUM bruges til at udvikle iteration af produktet. Dette dækker over design og udviklingen af den nye iteration.

Anvendte aktiviteter

Arbejdet i projektet er kategoriseret i forskellige aktiviteter, med forskellige mål og fokuspunkter. Aktiviteterne udføres delvist kronologisk, men med et vist overlap, da information fra en senere aktivitet, kan have indflydelse på udførelsen af en tidligere aktivitet. De anvendte aktiviteter er:

- Problemanalyse
- Krav
- Prioritering
- Analyse
- Design
- Implementering
- Test

3.1.2. Problemanalyse

Formålet er at sikre en forståelse for det overordnede problem, der bearbejdes i projektet, at forstå dets årsag, dets påvirkninger, at definere det, og afgrænse hvilke dele af problemet der vil blive bearbejdet i projektet. Efter en bred forståelse af problemet er opnået, gennem den udleverede case, udarbejdes en problemformulering. Denne skal opsummere det analyserede problem, for at give et kort og præcis pejlemærke til formålet med projektet. Til sidst nedfældes afgrænsninger, for at begrænse fokuset af den udarbejdede problemformulering.

3.1.3. Krav

Formålet er, at definere forskellige typer funktionaliteter, begrænsninger, og forventninger til det endelige projekt. Her defineres, abstrakte ønsker til den endelige funktionalitet, begrænsninger til anvendte teknologier, samt krav om testbar ydeevne.

TV 2 Krediteringsprogram

For at identificere de overordnede funktionaliteter, identificeres først systemets aktører. Disse gives et navn, en beskrivelse og nogle konkrete funktionelle mål. Herefter identificeres de overordnede funktionaliteter, som disse aktører vides, eller antages, at have brug for. Funktionaliteterne nedskrives som brugsmønstre, med id, navn, formål og fremgangsmåde. De eliciterede brugsmønstre indsættes i et brugsmønsterdiagram, for at give en visuel repræsentation af, hvordan brugsmønstrene hænger sammen både internt og med aktørerne.

Supplerende krav omfatter ikke-funktionelle krav, der forgrener sig i Produktkrav, Organisationskrav, og eksterne krav. Disse identificeres ud fra de den givne case, og identificeres efter krav type. De supplerende krav hjælper med at sætte tekniske pejlemærker for hvad der skal fokuseres på, eg. performance, sikkerhed, mm. De kan også definere hvilke teknologier og teknikker, der skal anvendes i projektet.

De overordnede funktionaliteter detaljeres efterfølgende. Her udvides de med en fremgangsmåde beskrevet i naturligt sprog, relevante aktører, stakeholders, prækonditioner, postkonditioner, og alternative scenarier. Disse giver en uddybning af funktioner, herunder er alternative scenarier en meget relevant tilføjelse.

3.1.4. Prioritering

Formålet er at vurdere de fundne krav, og få opstillet en prioriteret arbejdsliste, der tager forbehold for så mange relevante parametre som muligt. De 2 primære parametre er, hvor vigtige kravene er for kunden, og hvor vigtige kravene er for projektets succes og effektivitet. For at vurdere prioriteringen af funktionerne, anvendes 2 teknikker: MoSCoW, og risikoanalyse.

MoSCoW analyse vurderer funktionaliteterne ud fra deres relevans for produktet. Ethvert produkt har et specifikt formål, hvoraf nogle funktioner er centrale for dette formål, og andre er supplerende. MoSCoW giver en overordnet forståelse, for hvilke identificerede funktioner der er centrale, vigtige, og supplerende, og hvilke der er unødvendige.

Risikoanalyse er en general analyse, af hvilke ting der kan gå galt i projektet. De identificerede trusler er relevante i sig selv, men nogle har også speciel relevans for funktionsprioriteringen. Dette er som regel centrale funktioner, med en speciel mulighed for at skabe problemer.

Ud fra de indsamlede informationer, opstilles en prioriteringsliste for funktionerne. Denne er rationaliseret, på baggrund af alle relevante informationer indsamlet, både fra de udleverede informationer, kravanalysen, samt de ovennævnte analyser. Denne prioriteringsliste anvendes som produkt-backlog til SCRUM.

3.1.5. Analyse

Formålet er at udfolde kravene, for at finde delkrav eller del-funktionaliteter, nødvendige for at opfylde de eliciterede krav. Analysen fokuserer på den abstrakte løsning, frem for den konkrete. Det konkrete formål er at identificere de overordnede klasser, de overordnede funktioner, samt systemets indhold af brugerrelevant data.

TV 2 Krediteringsprogram

Først anvendes Verb/Noun analyse på de detaljerede brugsmønstre. Dette giver en liste af potentielle klasser, attributter og metoder. Klassekandidaterne bliver grovsorteret, og detaljeret med CRC-kort. Herefter visualiseres klasserne med analyseklassediagrammer.

Den eliciterede data skal også analyseres med henblik på implementering i en relationel database. Hertil benyttes ER-diagrammer, der giver en abstrakt visualisering af dataenes sammenhæng.

3.1.6. Design

Formålet her er at finde en konkret designløsning, der bygger på den abstrakte løsning fundet i analysen, som opfylder de eliciterede krav, og som følger deres prioritering. Der skal skabes en konkret plan for implementering, så implementeringen kan deles op i håndterbare opgaver. Dette sikrer overblikket over udviklingsprocessen, hvilket gør det nemmere at prioritere tidsforbruget.

For at omdanne de abstrakte klasser, attributter, og funktioner eliciteret i analysen, anvendes brugsmønsterrealisering. Dette har til formål at finde frem til de konkrete funktioner og hjælpefunktioner, der skal anvendes for at opfylde den pågældende funktionen. Samtidigt hjælper det til at fordele funktionerne på de rigtige klasser. Dette munder ud i designklassediagrammer, med en konkret udformning, der kan implementeres.

Foruden modellering af klasser, anvendes også pakkediagrammer, for at give et overblik over, hvordan klasserne skal placeres i projektet. Pakkerne bruges som semantiske pakker, der hjælper med at holde kategorier af funktionalitet adskilt i projektet.

Programmet har en visuel brugerflade, og for at gøre denne intuitiv, anvendes UI design. Funktionerne eliciteret gennem brugsmønsterrealiseringen inddeles på sider af brugerfladen. Herefter overvejes det hvilke brugergrænseelementer de skal betjenes af. Det overvejes hvor hvert element skal placeres, dets størrelse, udformning, samt eventuel tekst og ikoner. Dette tegnes i et tegneprogram, for at kunne bedømme brugervenligheden internt, før det bliver implementeret.

Det udarbejdede ER-diagram skal også omskrives til et implementerbart design. Hertil benyttes normalisering, hvorved de tilstræbte ACID-egenskaber opnås. Den normaliserede data visualiseres herefter i et UML-diagram, som en modificeret udgave af klassediagrammer.

3.1.7. Implementering

Formålet her er at implementere det udarbejdede design. Hovedmålet er, at de nødvendige funktionaliteter bliver implementeret, samt at sikre de tre egenskaber: kvalitet, hurtig udvikling, samt lavt ressourceforbrug.

Til implementeringen anvendes en version af SCRUM. Her lægges vægt på en udvælgelse af funktioner fra product-backlog til sprint-backlog i starten af sprintet, samt en reflektering af processen i slutningen af sprintet. Refleksionen kigger på hvorvidt sprint-Backlogten er opfyldt. Blev den opfyldt for hurtigt, tilføjes flere ting i næste sprint. Blev alle ting ikke udført, vurderes det om det skyldes en fejlvurdering af loggen, dårligt design eller ineffektiv implementering. Den samlede vurdering koges ned til gode ting der skal fortsætte, samt dårlige ting der skal undgås i næste sprint.

TV 2 Krediteringsprogram

Under sprintet anvendes et issueboard til at vurdere hvor langt man er i sprintet, samt hvem der er i gang med hvad. Undervejs kan større problemer deles op i mindre håndterlige opgaver. Samtidigt kan der identificeres yderligere opgaver, der kan medtages i næste sprint.

3.1.8. Test

Formålet her er at teste, hvorvidt det udviklede produkt opfylder de elicerede krav og derudover at kvalitetsteste produktet. Test kan opdeles i 3 dele: Udviklingstest, udgivelsestest og brugertest.

Under udviklingen af programmet, vil programmet kontinuert blive testet. Dette vil blive gjort gennem manuel tests, og automatiserede test. Formålet er at sikre at ændringer ikke utilsigtet ødelægger funktionaliteten andre steder i programmet. Ved at teste hyppigt, finder man nemmere frem til, hvad der forsagede fejlen.

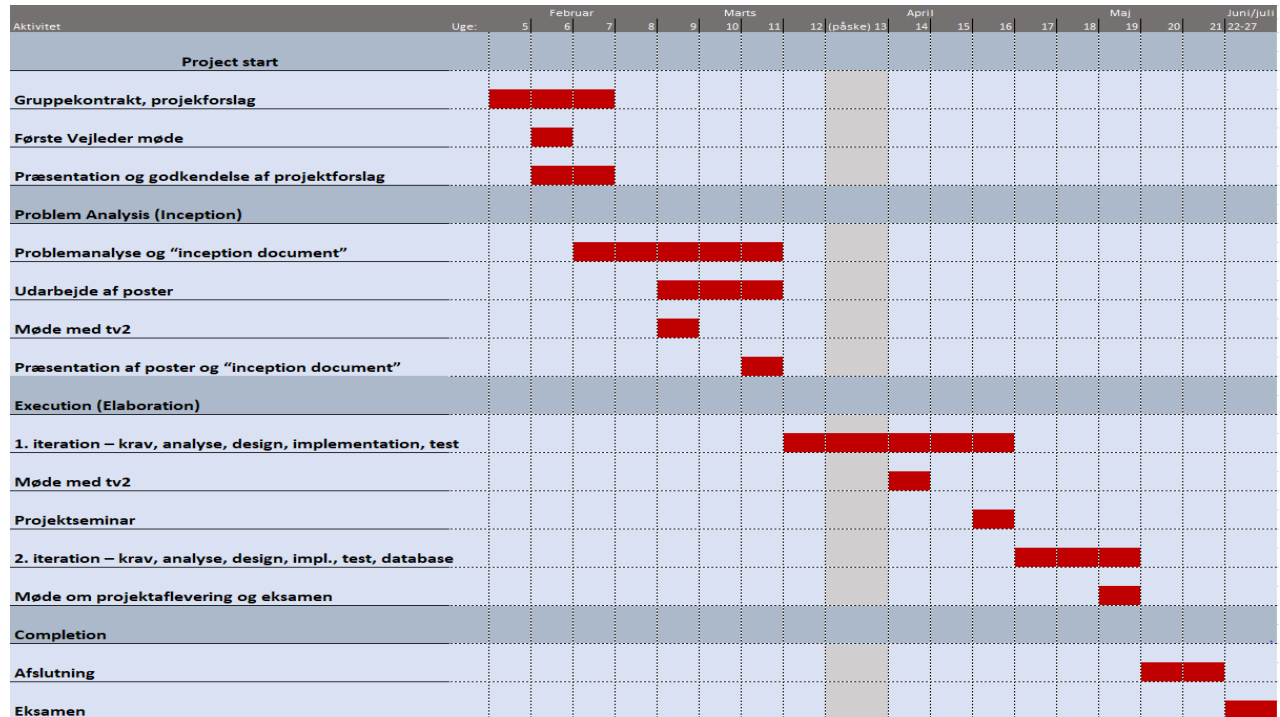
Efter en iteration, vil programmet blive testet igennem. Her er fokus på programmets funktionalitet, frem for de interne funktioner. Der tages udgangspunkt i de implementerede brugsmønstre, men der vil samtidigt implementeres et element af stresstest, for at sikre at programmet kan håndtere utilsigtet brug.

At et program er uden fejl, er ikke ensbetydende med at det opfylder dets formål. For at teste brugbarheden af programmet, benyttes en brugertest. Denne skal give indsigt i, hvordan andre mennesker tolker det produkt, vi har lavet, og om designbeslutninger omkring UI virker efter hensigten.

TV 2 Krediteringsprogram

3.2. Tidsplan

Ved begyndelsen af projektet blev der udleveret en overordnet tidsplan. Ud fra denne blev lavet en model, der skulle reflektere forskellige aktiviteterets påbegyndelse og færdiggørelse.



Figur 1: GANTT-diagram

3.3. Planlægning

Alle kravene eliciteret i kravspecifikation er ikke garanteret at blive opfyldt. Det er derfor vigtigt at få prioriteret kravene, så de kritiske krav bliver opfyldt, før der arbejdes på ikke-kritiske krav. Dette sikrer at det udarbejdede produkt får den højst mulige grad af relevans for kunden. Der er lavet en konkret prioriteringsliste, som tager udgangspunkt i hvert kravs vigtighed for projektet og tidsmæssige belastning.

MoSCoW Analyse *

Først prioriteres de supplerende krav med MoSCoW modellen.

TV 2 Krediteringsprogram

Must have	
M01	Databasen skal kunne administreres af TV 2.
M02	Producenter skal kunne tilføje projekter og roller for deres egne projekter.
M03	Der skal være offentlig adgang til informationer om projekter og kreditering.
M04	En funktionel GUI.
Should have	
M05	Krediteringssystemet skal kunne merge/linke personer som refererer til den samme person i den virkelige verden.
M06	Krediteringssystemet skal kunne eksportere en specificeret mængde af data ud fra programmet til formater som XML og CSV.
M07	Det skal være muligt at søge i krediteringssystemet for at finde bestemt data.
M08	Der skal være en tydelig (nem at bruge) reference fra TV-programmet til TV-programmets kreditering i systemet.
M09	Det skal være muligt at få informationer i krediteringssystemet fra en given produktion eller flere produktioner.
M10	Der kan være forskellige dele af systemet med forskellige adgang.
M11	Man kan hente informationer til Financial Management.
Could have	
M12	Man skal kunne se ændringer i systemet, hvorved man får en notifikation.
Won't have	
M13	At forbinde Krediteringssystemet til andre systemer som YouSee Play, Boxer Play, Stofa, etc.

Tabel 4: MoSCoW analyse

Prioriteringsliste *

For at have en prioriteret arbejdsliste, omformuleres kravene, så de kan udføres separat. Derefter opstilles de i en prioriteret liste, fra mest relevante til mindst relevante. Udgangspunktet for prioriteringen er MoSCoW og risiko analyse fra inceptionsdokumentet (se bilag K side 12-16). Prioriteringen er udarbejdet ud fra alle kravenes relevante egenskaber:

- Hvor betydningsfuldt kravet er for produktet.
- Hvor stort et læringsmæssigt udbytte det har.
- Hvor kritisk kravet er for produktets funktionalitet.
- Hvor stor risiko der er for, at kravet ikke kan opfyldes, og arbejdet derfor bliver spildt.
- Den logiske fremgang i udviklingen af softwareløsningen.

Listen benyttes igennem de næste faser, for at sikre det højest mulige udbytte af arbejdet. Prioriteringslisten er opdelt i første iteration (1-4) og anden iteration (5-12). Første iteration skaber et funktionelt system med en GUI, hvorfra man kan tilføje, ændre, søge og se krediteringer. Anden iteration bygger videre på første iteration og tilføjer flere forskellige funktionaliteter. Det er ikke forventet, at alle punkter bliver færdiggjort i løbet af anden iteration, men det ønskes at tilføje så meget funktionalitet som muligt.

TV 2 Krediteringsprogram

ID	Prioritering	Elementer i projektet
P1	1	Lokalt filsystem (1. iteration)
P2	2	Det skal være muligt at tilføje projekter og roller. - Disse tilføjede roller skal overholde de givne regler for kreditering - Det skal være muligt at redigere projekter og roller - Det skal være muligt at fjerne projekter og roller
P3	3	Systemet skal have 3 Interfaces, en til administrering (super user), en til kreditering og en offentlig.
P4	4	Basalt Graphical User Interface (GUI) - GUI'en skal inddeles i 3 dele: administrering, kreditering og offentlig. - Under administrering skal kunne ændres i krediteringer - Under kreditering skal kunne tilføjes krediteringer og ændres krediteringer. - Under offentlig skal krediteringer kunne søges og ses.
P5	5	Database (2. iteration)
P6	6	Systemet skal kunne merge/linke forskellige krediterede navne, så de peger på den samme unikke person.
P7	7	Der skal kunne oprettes forskellige administrerings- og krediteringsbrugere som kan logge ind.
P8	8	Databasen skal logge hvem der har foretaget ændringer i systemet.
P9	9	Oprettede Krediteringer skal accepteres af TV 2 og krediteringer skal have en status for hvorvidt de er accepterede. Efterfølgende ændringer af krediteringer foretaget af producenter skal ændre status tilbage til 'ikke accepteret'.
P10	10	Det skal vise den aktuelle produktion fra flow-tv.
P11	11	Det skal være muligt at eksportere en specificeret mængde af data ud fra programmet til XML-og CSV-filer.
P12	12	Det skal være muligt for administrator at se en liste med alle ændringer lavet i systemet, og hvem der har lavet disse ændringer.

Tabel 5: Prioriteringsliste

TV 2 Krediteringsprogram

3.4. Scrum

Ved begyndelsen af første iteration blev en kravprioriteringsliste opsat, hvilket blev brugt som product-backlog gennem hele projektet. Ved første iteration blev der ikke brugt SCRUM, men der blev alligevel, ved afslutningen, afholdt et møde svarende til et retrospektive og Sprint Review. Med udgangspunkt i erfaringen fra afslutningen af første iteration, blev vi introduceret til SCRUM for at benytte en agil fremgangsmåde, vedrørende udvikling af koden for resten af projektet. Der blev først valgt nogle SCRUM -buts, for at finde den udgave af SCRUM, der bedst passede til vores arbejdsform, hvilket kan ses i bilag F. Der blev besluttet at lave tre SCRUM -buts: Daily SCRUM blev reduceret til én gang om ugen, retrospektive og Sprint Review er kombineret som én opsamling og der er ikke nogen product owner. Beslutningen om ikke at have en product owner er taget, da der ikke var én person i gruppen, der havde en bedre grundlæggende forståelse for produktet. Ansvar som Product Owner, tilfaldt derfor alle gruppemedlemmer. Efter første iteration, blev der lavet en opsamling på de færdige sprint, for at videreføre hvad der gik godt og forbedre hvad der gik skidt. Der blev udvalgt en SCRUM-master, for at få skabt noget struktur omkring processen.

Trello blev brugt som et værktøj til at skabe struktur for udviklingen af projektet. Backlogen blev indsat i Trello. I Trello blev der brugt tre grupperinger af opgaver. En med opgaver der skal laves, en anden med opgaver der er i igangværende og en tredje med opgaver der er færdige. Denne udnyttelse af Trello som et værktøj til at strukturere SCRUM fungerede godt og gav alle mulighed for hele tiden at se, hvad der blev lavet af hvem, hvad der manglede og hvad der var færdigt.

Ved begyndelsen af anden iteration, blev der oprettet en sprint backlog, som indeholdt alt der skulle laves. Med udgangspunkt i kravprioriteringslisten (tabel 5) blev punkterne for anden iteration prioriteret. Backlog for anden iterations sprint kan ses under bilag G.

Efter anden iteration blev der afholdt et retrospektive og Sprint Review (se bilag G), hvor det blev fastslået, at programmet havde opnået de fleste af vores punkter fra Backlogen. Samtidig blev der diskuteret hvad der gik godt og hvad der gik skidt, for at kunne forbedre fremtidige sprints. Grundlæggende fulgte vi ikke SCRUM fuldstændigt, hvor især Daily-SCRUM ikke blev overholdt fuldstændigt. Mangel på struktur var et diskuteret problem, hvor en mere fokuseret SCRUM-tilgang kunne have hjulpet processen. SCRUM-masteren skulle have været bedre til at fastholde Daily-SCRUM og generel brug af SCRUM-artefakter. SCRUM er ikke overholdt fuldstændigt i løbet af anden iteration, men denne introduktion til SCRUM har givet en bedre forståelse for agil udvikling og brugen af SCRUM.

TV 2 Krediteringsprogram

4. Hovedtekst

Igennem to iterationer er der blevet arbejdet med konstruktion af et krediteringsprogram, som løsning på case fra TV 2. Dette afsnit gennemgår arbejdes processen igennem de to iterationer for at kunne diskutere og evaluere processen.

4.1. Overordnede krav *

Kravspecifikationen forsøger at identificere nødvendige og mulige funktioner og egenskaber for produktet. Kravene skal skabe et overblik over produktets minimale omfang og potentielle maksimale omfang. For at opfylde formålet identificeres først aktørerne, herefter aktørernes brugsmønstre og derefter visualiseres brugsmønstrene i et diagram. Størstedelen af disse krav og brugsmønstre, er fundet ud fra TV 2's Credit Management Case².

4.1.1. Aktørliste

Aktør	Beskrivelse	Mål
Administrator	Aktør ansvarlig for systemets integritet	• Tilføje brugere • Ændre brugere • Ændre krediteringslister • Sammenflette krediteringer til én person
Producent	Aktør med mulighed for at tilføje og ændre	• Tilføj egne krediteringer • Ændre egne krediteringer • Sammenflette krediteringer til en person
Offentlig	Aktør der har adgang til at søge efter og se krediteringer.	• Søg krediteringer • Se krediteringer
TV-tid	Aktør der leverer informationer om aktivt flow-tv.	

Tabel 6: Aktørliste

Aktørlisten gav 3 brugeraktører og én systemaktør. Systemaktøren TV-tid er en delvist uaktuel aktør. Ifølge projektets afgrænsninger vil produktet ikke være online, og vil derfor ikke kunne kontakte TV-tid. TV-tid som aktør er dog aktuel for det potentielle færdige produkt og medtages derfor til refleksion. De 3 brugeraktører giver 3 forskellige vinkler, som produktet skal kunne fungere med.

TV 2 Krediteringsprogram

4.1.2. Brugsmønsterliste

ID	Brugsmønster	Formål	Fremgangsmåde
1	Tilføj bruger	Tilføjer en ny bruger	- Nye brugers informationer indgives - Bruger er tilføjet til listen af brugere
2	Søg bruger	Søger efter bruger ved hjælp af søgeord	- En aktør indtaster søgeord - En liste af matchende brugere vises.
3	Ændre krediteringer	Ændre en eksisterende krediteringsliste	- En aktør søger efter en eksisterende krediteringsliste - Der ændres på krediteringslisten - Krediteringslisten opdateres med den nye information.
4	Tilføj kreditering	Tilføjer en krediteringsliste	- Information for ny krediteringsliste indgives - Krediteringslisten tilføjes til listen over krediteringer.
5	Notifikationsliste	En liste af ændringer, relevante for brugeren, vises ved login	En liste af ændringer vises ved login
6	Søg egne krediteringer	Søg efter krediteringslister ved hjælp af søgeord, hvor man kun får egne krediteringer fremvist	- En aktør søger efter egne krediteringer - Aktøren får vist en liste med givne krediteringer
7	Søg kreditering	Søg efter krediteringslister ved hjælp af søgeord	- En aktør søger efter en eksisterende krediteringsliste - En liste af matchende krediteringslister vises
8	Data eksportering	Krediteringsdata kan eksporteres i forskellige formater	- En aktør søger efter krediteringer - Formateringen vælges - En fil eksporteres med den valgte formatering

TV 2 Krediteringsprogram

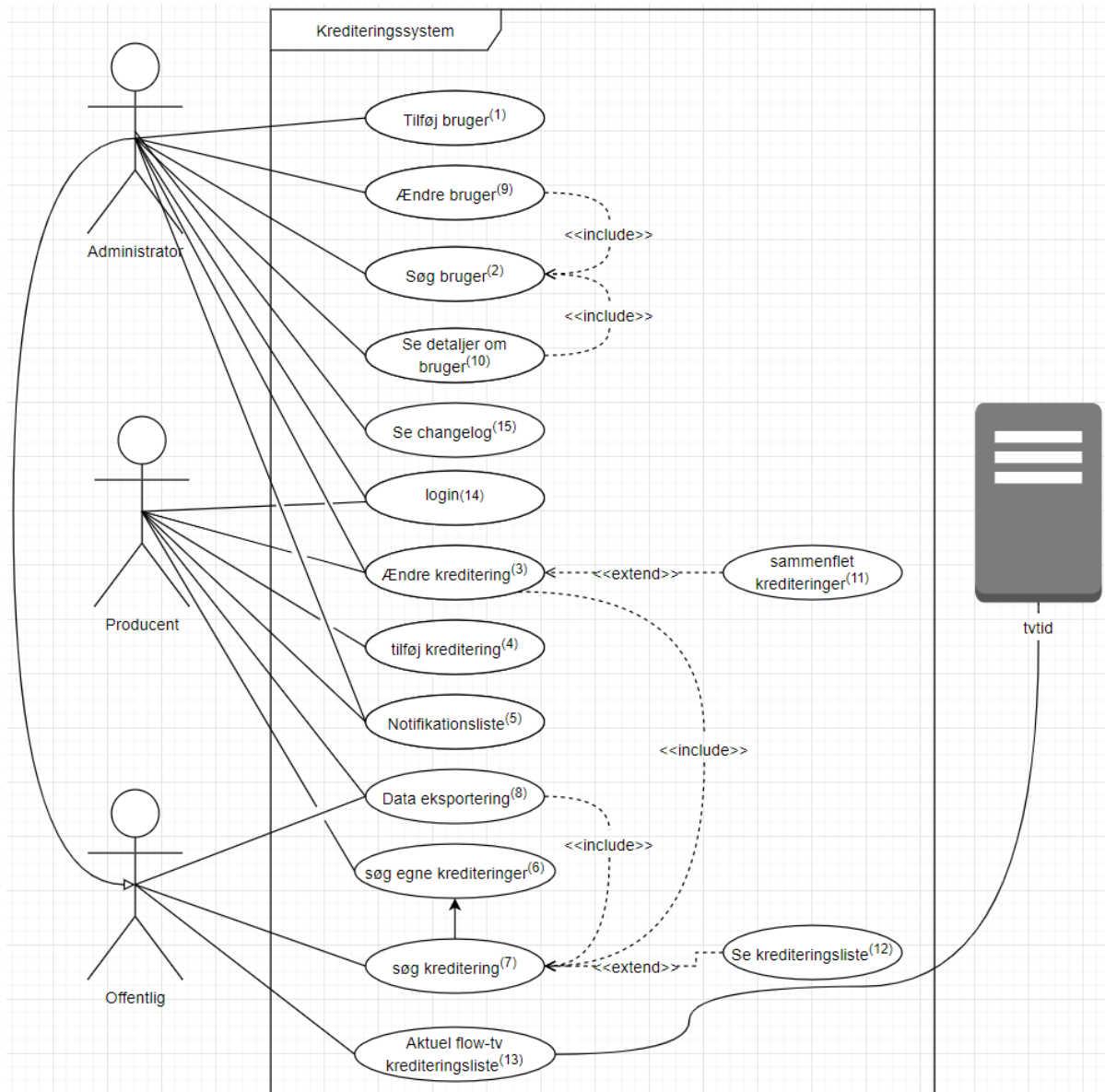
9	Ændre bruger	Ændre på en eksisterende bruger	• En aktør søger efter en eksisterende bruger • Der indgives nye informationer om brugeren • Brugeren opdateres med den nye information
10	Se detaljer om bruger	Se detaljer om en udvalgt bruger	• En aktør søger efter bruger (9), og har trykket på en af brugerne • Der fremvises information om den valgte bruger
11	Sammenflet krediteringer	Personer krediteret i krediteringslister skal sammenflettes, så flere krediteringer peger på den samme person	• Et navn indtastes ved ændring af krediteringsliste • En liste af personer der matcher navnet vises • Den rigtige person vælges • Krediteringen peger herefter på den person
12	Se krediteringsliste	Se en liste med alle dem som er krediteret ud fra de givne søgeord	• En aktør søger efter krediteringer (7) • Aktøren får vist en liste med givne krediteringer
13	Flow-tv krediteringsliste	En liste af krediteringer fra aktive flow-tv-programmer	• En liste af krediteringslister fra aktive flow-tv-programmer
14	Login	At verificere en bruger.	• brugeren afgiver logininformationer • Systemet verificerer informationerne og omdirigerer brugeren til den korrekte side.
15	Changelog	At kunne se ændringer lavet i systemet, hvornår de var lavet og af hvem.	• Administrator logger på og vælger changelog menuen • Admin kan nu se hvilke ændringer der er lavet i et givet tidsrum og af hvem.

Tabel 7: Brugsmønsterliste

TV 2 Krediteringsprogram

Brugsmønsterlisten giver os en kort beskrivelse af alle de brugsmønstre, som en eller flere aktører skal kunne have i systemet. De er beskrevet på en ret abstrakt og kortfattet måde for at give et nemt og simpelt overblik over systemets funktionaliteter.

4.1.3. Brugsmønsterdiagram



Figur 2: Brugsmønsterdiagram

Brugsmønsterdiagrammet skal give et samlet overblik over, hvad de forskellige aktører skal kunne i systemet, og hvordan visse funktionaliteter hænger sammen. Diagrammet er med til at give et abstrakt overblik systemets funktioner og brugere som en helhed.

TV 2 Krediteringsprogram

4.1.4. Supplerende krav

Ud fra de generelle brugsmønstre udarbejdes supplerende krav. Disse dækker krav til produktet, der ikke kan opstilles som funktioner. De ikke funktionelle krav dækker generelle krav til produktets implementering. De funktionelle krav dækker, i mere generelle træk end brugsmønstrene, produktets funktionaliteter.

ID	Kategori	Ikke funktionelle krav
S1	Miljøkrav	Skal udvikles i Java miljøet
S2	Miljøkrav	Skal anvende PostgreSQL Database miljø
S3	Designkrav	Skal have en 3-lags arkitektur
S4	Grænsefladekrav	Skal implementere en grafisk brugergrænseflade
S5	Designkrav	Skal separerer Presentations-, Domain-, persistence lag
S6	Brugbarhedskrav	Bruger grænsefladen skal være let og intuitiv at bruge for alle aktører

Tabel 8: Ikke funktionelle krav

ID	Funktionelle krav
f1	Databasen skal kunne administreres af TV 2
f2	Producenter skal kunne tilføje projekter og roller for deres egne projekter
f3	Der skal være offentlig adgang informationer om projekter og kreditering
f4	En funktionel GUI
f5	Krediteringssystemet skal kunne merge/linke personer som refererer til den samme person i den virkelige verden.
f6	Krediteringssystemet skal kunne eksportere en specificeret mængde af data ud fra programmet til formater som XML og CSV.
f7	Det skal være muligt at søge i kreditering systemet for at finde bestemt data.
f8	Der skal være en tydelig (Nem at bruge) reference fra Tv-programmet til Tv-programmet credits i krediteringssystemet.
f9	Det skal være muligt at få informationer i krediteringssystemet fra en given produktion eller flere produktioner.
f10	Der kan være forskellige dele af systemet med forskellige adgang
f11	Man kan hente informationer til Financial Management
f12	At subscribe til ændringer i systemet, hvorved man får notifikationer når der sker ændringer i systemet.
f13	At se en liste af krediteringer fra aktive flow-tv-programmer

Tabel 9: Funktionelle krav

TV 2 Krediteringsprogram

4.2. Detaljerede krav

Da vi nu har fundet systemets krav, skal de uddybes og detaljeres, for at få afklaret mere specifikt hvordan de forskellige funktionaliteter skal fungere. Dette giver en bedre forståelse for hvordan de forskellige brugsmønstre skal udføres, uden af vi, på dette niveau, har behov for stor teknisk viden.

4.2.1. Detaljerede brugsmønstre

Søg Kreditering*	
ID:	#7
Formål:	At levere credits som matcher søgeordene.
Oversigt:	Den offentlige bruger nøgleord til at søge efter specifikke udsendelser eller personer som de gerne vil have krediteringsoplysninger for. Systemet returnerer en liste med alle krediteringer der matcher de søgte nøgleord.
Aktør:	Offentlig, Administrator
Stakeholders:	Offentlig: Ønsker at være i stand til at søge krediteringer for tv eller film de har set. TV 2: ønsker at offentligheden er i stand til at søge efter krediteringer uden at bruge dyrebar skærmtid.
Forudsætning:	Ingen
Postconditions:	Krediteringer der matcher søgeord er returneret.
Sekvens:	1. Offentligheden vil se kreditering for en bestemt film eller tv-show. 2. Offentligheden skriver søgeord for den kreditering de ønsker at finde. 3. Systemet finder og henter al kreditering matchende søgeordet. 4. En liste af krediteringer matchende søgeord er returneret til offentligheden.
Alternative scenarier:	a. Ingen kreditering matchende søgeordene, og brugeren vil blive informeret.

Tabel 10: Detaljeret brugsmønster - søg kreditering

Tilføj kreditering*	
ID:	#4
Formål:	At tilføje krediteringer til systemet.
Oversigt:	Producenten indskrifter informationer om produktet der krediteres, og tilføjer derefter kreditering af individer.
Aktør:	Producent
Stakeholders:	Producent, der ønsker at kreditere medarbejdere. Medarbejdere, der ønsker at blive krediteret ordentligt. TV 2, der ønsker at vise og videresende de rigtige krediteringer.
Forudsætning:	Aktør er logget ind som producent eller administrator.
Postconditions:	Kreditering af udsendelse er tilføjet til systemet.
Sekvens:	1. Aktør initierer tilføjelse af kreditering. 2. Aktør tilføjer informationer om udsendelsen der krediteres for. 3. Aktør tilføjer individer der ønskes krediteret med navn og funktion. 4. Aktør gennemser krediteringslisten og accepterer den. 5. Krediteringslisten tilføjes til systemet.
Alternative scenarier:	a. Aktør afslutter proceduren før krediteringen er tilføjet til systemet.

Tabel 11: Detaljeret brugsmønster - tilføj kreditering

TV 2 Krediteringsprogram

Ændre kreditering*	
ID:	#3
Formål:	At ændre krediteringer i systemet.
Oversigt:	Producent eller administrator udvælger krediteringsliste de ønsker at ændre, og tilføjer herefter ændringer.
Aktør:	Producent, Administrator
Stakeholders:	Producent, der ønsker at kreditere medarbejdere. Medarbejdere, der ønsker at blive krediteret ordentligt. TV 2, der ønsker at vise og videresende de rigtige krediteringer.
Forudsætning:	Aktør har fundet krediteringen der ønskes ændret. Aktør er logget ind som producent.
Postconditions:	Kreditering af udsendelse er ændret i systemet.
Sekvens:	1. Aktør initierer ændring af kreditering 2. Aktør ændrer informationer om udsendelsen eller i krediterede individer. 3. Aktør gennemser krediteringslisten og acceptere den. 4. Krediteringslisten ændres i systemet.
Alternative scenarier:	a. Aktør afslutter proceduren før krediteringen er tilføjet til systemet.

Tabel 12: Detaljeret brugsmønster - Ændre kreditering

De detaljerede brugsmønstre giver en mere dybdegående og udfyldt beskrivelse af udvalgte brugsmønstre. De skal give et mere komplet indtryk af brugsmønstrets funktion, som skal hjælpe med at give en bedre forståelse af hvordan brugsmønsteret realiseres.

De detaljerede brugsmønstre gennemgår kort hvad der sker i hvert trin af brugsmønsterets udførelse, samt eventuelle intentionelle afvigelser, så som hvis brugeren ikke har opfyldt visse krav for at gennemføre funktionen med succes. De beskriver også kort krav til systemets tilstand før brugsmønsteret kan udføres, og eventuelle efterfølgende ændringer af systemets tilstand. Samtidig beskriver de også hvem der bruger brugsmønsteret, og hvem der kan have fordel af at brugsmønsteret bliver implementeret.

4.2.2. Supplerende krav

De supplerende krav skal være med til at beskrive afgrænsninger for brugen eller funktionaliteten af systemet. Disse begrænsninger skal være med til at guide udviklingen af systemet og dens funktioner.

S1 Skal udvikles i Java miljøet

TV 2 har stillet krav om at systemet udvikles i Java, da hoveddelen af deres eksisterende systemer allerede er Java baseret. Dette skal gøre det lettere for TV 2 at kunne vedligeholde det endelige produkt in house².

S2 Skal anvende PostgreSQL Database miljø

TV 2 har også stillet krav om at PostgreSQL databasemiljøet bliver anvendt som databasemiljø, da dette er et af de databasemiljøer de allerede bruger².

TV 2 Krediteringsprogram

S3 Skal have en 3-lags arkitektur

For at begrænse systemets interne tilgængelighed og kommunikation, skal der benyttes en 3-lags arkitektur. Dette skal hjælpe systemets holdbarhed over for ændringer, da det begrænser systemet til kun at kunne lave interne kald mellem to lag.

S4 Skal implementere en grafisk brugergrænseflade

For at gøre systemet nemt at bruge for både Offentlige, Administratorer og Producenter, skal der implementeres en grafisk brugergrænseflade for alle disse aktører. Brugergrænsefladen er også med til at begrænse og guide de forskellige aktører til funktioner relevante for dem.

S5 Skal separerer Presentations-, Domain-, persistence lag

For at gøre det overskueligt og håndterbart for TV 2 at kunne vedligeholde systemet skal der være en klar separeret opdeling mellem presentations-, domain-, og persistence laget. Dette sikrer at håndteringen af systemets funktioner er indkapslet og derved nemmere at vedligeholde og modificere uden at det påvirker resten af systemet.

S6 Bruger grænsefladen skal være let og intuitiv at bruge for alle aktører

Det skal være muligt for de forskellige aktører at kunne udnytte systemet på en let og intuitiv måde. Det skal være et minimalt behov for oplæring i systemet, da udefrakommende Producent aktører kommer til at skulle bruge systemet, og for de Offentlige aktører skal der i de fleste tilfælde ikke være noget behov for oplæring.

TV 2 Krediteringsprogram

4.3. Analyse

Målet med analysen i elaborationsfasen, er at producere en konceptuel model over systemets ønskede adfærd og struktur. Modellen er ikke en komplet repræsentation over systemet, men en mere udetaljeret abstrakt model, som repræsenterer de entiteter der befinder sig i den virkelige verden. Formålet med en konceptuel, abstrakt model, som ikke nødvendigvis afspejler det endelige produkt, er at det bruges som et middel til at fremme diskussion omkring det foreslåede system. De ønskede resultater er abstrakte oversigter over klasser, funktioner, attributter, samt data i en relationel database sammenhæng. Der fremvises analyse for brugsmønstret søg kreditering. Analyse for brugsmønstrene tilføj kreditering og ændre kreditering kan ses i bilag J.

Verb/Noun analyse af søg krediteringer

Først blev det detaljerede brugsmønster analyseret, hvor verber blev farvet rød og substantiver farvet blå.

Søg Kreditering	
ID	#2
Formål:	At levere kredits som matcher søgeordene
Oversigt	Den offentlige bruger nøgleord til at søge efter specifikke udsendelser eller personer som de gerne vil have krediterings oplysninger for. Systemet returnerer en liste med alle krediteringer der matcher de søgte nøgleord .
Aktør	Offentlig, Administrator
Stakeholders	Offentlig: Ønsker at være i stand til at søge kredit for tv eller film de har set . TV 2: Ønsker offentligheden til at være i stand til at søge efter kredits uden at bruge dyrebar skærm tid
Forudsætning	None
Postconditions	Krediteringer der matcher søgeord er returneret.
Sekvens	1. Offentligheden vil se kreditering for en bestemt film eller tv-show 2. Offentligheden skriver søgeord for den kreditering de Ønsker at finde 3. Systemet finder og henter al kreditering matchende søgeordet . 4. En liste af kreditering matchende søgeord er returneret til offentligheden .
Alternative scenarier	Ingen kreditering matchende søgeordene , og brugeren vil blive informeret .

Tabel 13: Verb/noun analyse - søg kreditering

Relevante verber blev sorteret som funktionskandidater og relevante navneord blev sorteret som klassekandidater og attributter.

Klassekandidater	Attributter	Funktionskandidater
------------------	-------------	---------------------

TV 2 Krediteringsprogram

Kreditering Offentlige Administrator Systemet Brugeren	Liste Krediterings-oplysninger Udsendelser Nøgleord Skærm tid Film tv-show	Leverer() Matcher(nøgleord) Søge(nøgleord) Se(kreditering) Skriver(nøgleord/søgeord) Henter (Kreditering) Informere ('Ingen kreditering')
--	--	---

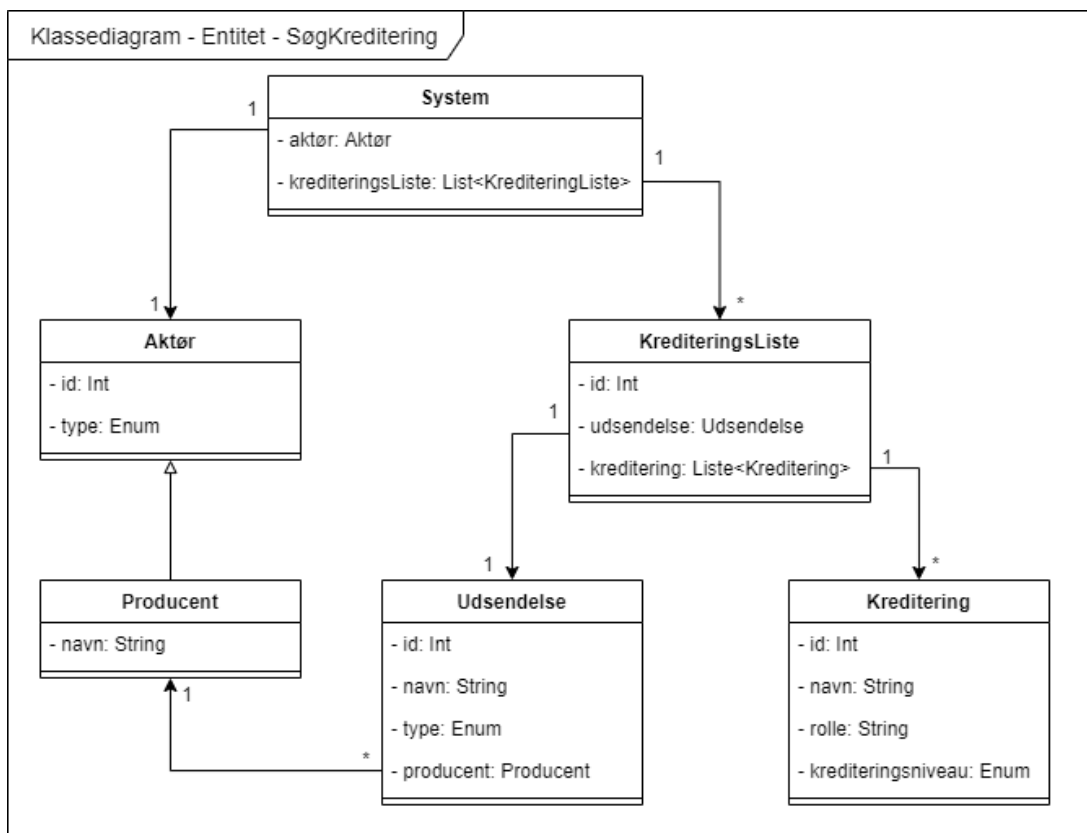
Tabel 14: Verb/Noun analyse - resultater

Herefter blev klassekandidaterne omdannet til CRC-kort, hvor det formodede ansvar og samarbejdspartnere blev beskrevet.

Klassenavn	Ansvar	Samarbejdspartnere
Kreditering	Holder informationer om de krediterede, om projektet og producenten.	System
Systemet	Håndterer søgningen	Kreditering, Administrator
Brugeren	Holder informationer om den autoriserede bruger	System

Tabel 15: CRC-kort - søg kreditering

CRC-kortene blev herefter omdannet til et analyseklassediagram, indeholdende attributterne forbundet med de forskellige klasser.



Figur 3: Klassediagram - Søg kreditering

TV 2 Krediteringsprogram

I brugsmønsteret søg kreditering, er der blevet fundet 5 analyseklasser, med hver deres ansvar og samarbejdspartnere. Klassernes ansvar og samarbejdspartnere er blevet defineret, og danner grundlag for at skitsere det med den tilhørende struktur i UML. Dette er blevet fundet ud fra brugsmønstrene skrevet i normalsprog, og har derfor en tæt knytning til problemdomænet.

Det resulterende UML-diagram er en repræsentation af den statiske analyse, omhandlende strukturen af systemet. Den statiske del af analysen skal formå at modellere systemets struktur.

Den dynamiske analyse omhandler systemets opførsel. Det vil sige systemets funktionelle aspekter. Disse blev ikke modelleret, men funktioner blev identificeret ud fra funktionskandidaterne.

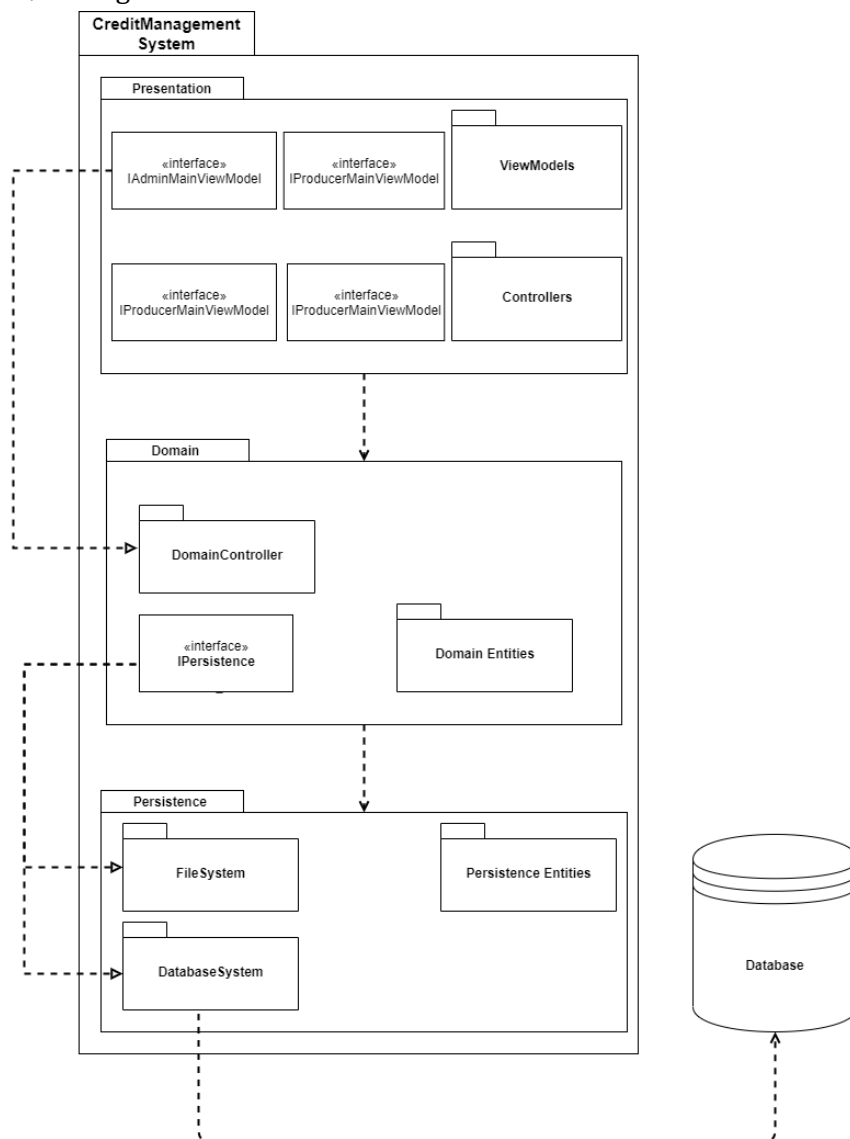
TV 2 Krediteringsprogram

4.4. Design

Med udgangspunkt i analysen laves der design for softwarearkitekturen og detaljeret design. Der beskrives processen for hvilke design beslutninger, der blev taget før påbegyndelsen af implementeringen, for både første og anden iteration.

4.4.1. Software arkitektur

Det er fastlagt fra projektudbyderen, at den grundlæggende software arkitektur, for projektet, skal være en trelagsmodel. Ved starten af første iteration blev der konstrueret et pakkediagram, som viser strukturen, med et præsenterationslag, domænelag og persistenslag. Pakkediagrammet viser hvordan vi konstruerer de tre lag, med interfaces som ledet mellem lagene. I første iteration benyttede persistens laget et lokalt filsystem, hvor dette skiftes til en database i anden iteration. Da softwarearkitekturen er opsat efter trelagsarkitektur, som benytter interfaces, bør dette skift ikke blive problematisk. Herunder ses et pakkediagram, revideret i anden iteration, som både tager højde for første og anden iteration.



Figur 4: Pakkediagram

TV 2 Krediteringsprogram

Trelagsarkitekturen bygger på, at der kun er nedadgående information. Præsentations laget kender derfor kun til domænelaget, og domænelaget kender kun til persistens laget. Samtidigt benyttes der interfaces for at kunne fastlægge hvilke forbindelser der løber imellem lagene. Denne software arkitektur giver os mulighed for nemt at kunne udskifte lag, da funktionaliteten er fastlagt gennem en envejskommunikation og interfaces. Denne opsætning gør at programmet nemmere kan videreudvikles og vedligeholdes. Dette giver et godt udgangspunkt for et software system og sikre at programmet ikke bliver rodet, eller at der skabes unødvendige afhængigheder, som vil skabe problemer i fremtidige sprints. Denne nedadgående envejskommunikation giver dog den begrænsning at persistens laget ikke har mulighed for at opdatere præsentationslaget hvis der sker ændringer fra andre brugere. Dette er en begrænsning, som vores design ikke tager højde for og en bruger er tvunget til at lave en ny søgning på data, for at hente nye ændringer. Som klargøring til implementeringen blev der opsat en mappe struktur, som matchede med det konstruerede pakkediagram.

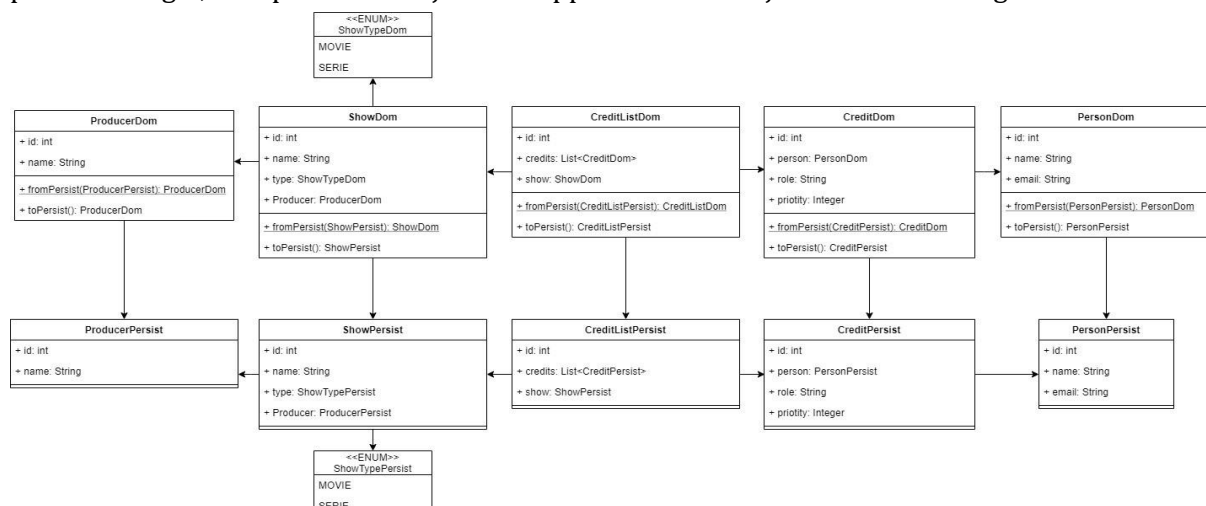
4.4.2. Design beslutninger

Singleton

Under designet blev det besluttet at anvende singletonmønsteret. Singletons benyttes for nemt at dele data mellem flere forskellige klasser, der skal have samme tilstand. Singleton gør det muligt at oprette et objekt, som kun har én instans, som herved kan tilgås fra flere forskellige klasser. Disse objekter kan tilgås for at få noget information, som kun skal eksistere i en form ad gangen. Singletonmønsteret bliver brugt til at håndtere interaktion mellem lagene og af viewmodels i præsentationslaget. Singleton benyttes også ved forbindelsen til databasen, da Databasesystem klassen er en singleton, som kun skal skabe forbindelse til databasen én gang. Derefter kan andre klasser tilgå databasen gennem denne singleton, og det er herved sikret at det er samme grundlag de benytter til at tilgå databasen.

Objekt Mapping

Der er lavet et entitetsdiagram for at skabe et overblik over, hvordan alle de forskellige entiteter skal mappes mellem domæne entiteter og persistens entiteter. Der er besluttet at benytte mapping, for at overholde trelagsmodellen. Da præsentationslaget ikke må benytte noget fra persistenslaget, skal persistensobjekter mappes til andre objekter i domænelaget. Herved vil en



Figur 5: Klassediagram - Mapping struktur

TV 2 Krediteringsprogram

udskiftning af persistenslaget kun påvirke domænelaget, og ikke præsenteringslaget. Samtidigt skal domæneobjekterne mappes tilbage til persistensobjekter, før de kan sendes til persistenslaget. Mapping logikken er udelukkende placeret i domæneobjekterne, da persistenslaget hverken kan mappe til eller fra domæneobjekter, uden at kende til dem.

UI-design

I både første og anden iteration er det lavet ui-design, hvor der er opsat modeller som viser den generelle opsætning af UI, for de grundlæggende sider (se bilag I). Disse UI- design skitser giver en bedre fælles forståelse for hvordan FXML-dokumenter skal opsættes, og de er derfor udarbejdet, for at give en fælles forståelse for UI-opsætningen.

Der er udvalgt færdige ikoner, som matcher med den generelle forståelse af ikoner, for at gøre bruger oplevelsen bedre og mere simpel. Der benyttes samme ikoner alle steder hvor samme funktionalitet finder sted, som ved alle søgefelter benyttes et forstørrelsesglasikon. Dette er valgt for at have genkendelighed, for at give brugeren en nemmere og mere simpel tilgang til systemet.

Der benyttes TV 2's røde farve² til en stor andel af det grafiske interface, da programmet er lavet til at være en del af TV 2's univers. Dette giver programmet en sammenhæng med resten af TV 2's software, og er derfor et aktivt valg, for at få en grad af sammenhæng med resten af TV 2's økosystem.

4.4.3. Detaljeret design

Ved første og anden iteration er der lavet detaljeret design, før implementeringen er begyndt. Dette til formål at give en bedre forståelse for, hvordan de forskellige brugsmønstre skal implementeres. En del af de endelige design beslutninger blev dog først truffet i løbet af implementeringen, da der ikke var en fuldstændig forståelse for system, før implementering blev begyndt. Påbegyndelsen af implementeringen gav nogle nye perspektiver, som var med til at ændre, hvordan koden i sidste ende blev udformet.

Første iteration

I første iteration var der fokus på de fire første prioriteringspunkter i prioriteringslisten, se tabel 5. Efter analyse af "søg kreditering", "ændre kreditering" og "tilføj kreditering", designes et design klasse diagram, for at kunne implementere disse brugsmønstre. Det detaljerede design for brugsmønsteret "søg kreditering", (se øvrige detaljerede design i bilag J.)

Dynamiske del

Den dynamiske del består i at finde systemets opførsel og adfærd, samt hvordan systemet håndterer events. Når vi modellerer vores adfærd for systemet, er det vigtigt at definere hvilke ansvarsområder de forskellige klasser har. Hvis en klassens ansvarsområde afhænger af andre klasser, kan det få konsekvenser for kvaliteten af vores software. Vedligeholdelse og fremtidige ændringer på systemet vil blive vanskeligere, da klassernes høje kobling med hinanden betyder, at én ændring, et sted, kan have stor indflydelse på andre komponenters funktionalitet. En disciplin som test, især af funktionalitet på klasseniveau, har gavn af denne beslutning om lav kobling, da der herved er færre mulige fejlkilder.

² <https://TV2marketing.dk/design/tv-2-brand/>

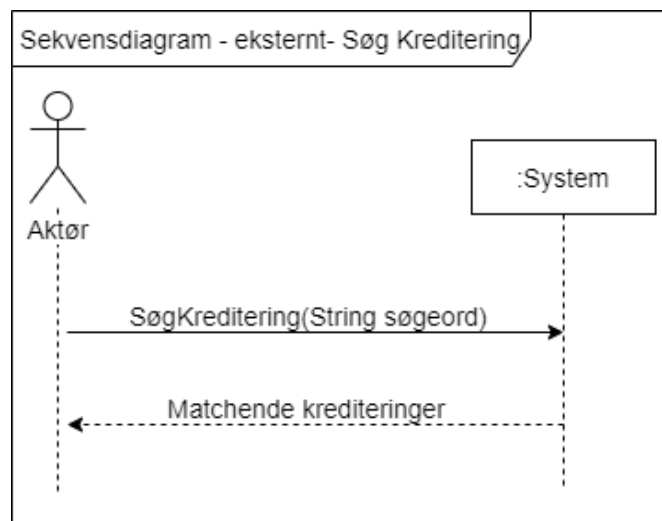
TV 2 Krediteringsprogram

Modellering af den dynamiske del af systemet løses ved hjælp af realisering af de detaljerede brugsmønstre. Hvert brugsmønster har præcist én realisering, hvor elementerne af denne realisering er klassediagrammer samt sekvensdiagrammer. Klassediagrammer viser de forskellige klassers associationer med hinanden, hvor sekvensdiagrammer viser de objekter som realiserer det ønskede brugsmønsters adfærd.

Sekvensdiagrammer – eksternt

De eksterne sekvensdiagrammer er sekvensdiagrammer, der giver et overblik over events for et brugsmønster over tid. Diagrammet viser hvordan en aktør interagerer med systemet, for at initiere det valgte brugsmønster. På eksterne sekvensdiagrammer vises systemets indre operationer ikke, men viser de systemevents aktøren kan igangsætte, samt hvilke input og output der bliver givet. Rækkefølge for hvornår de forskellige hændelser igangsættes vises også. Identificering af de forskellige systemoperationer sker ved at analysere interaktionen mellem system og aktør ud fra de detaljerede brugsmønstre.

Det eksterne sekvensdiagram, for "søg kreditering", viser, hvordan en aktør giver systemet et søgeord, hvorefter systemet vil returnere matchende krediteringer.



Figur 6: Sekvensdiagram - eksternt - Søg kreditering

Sekvensdiagrammer – internt

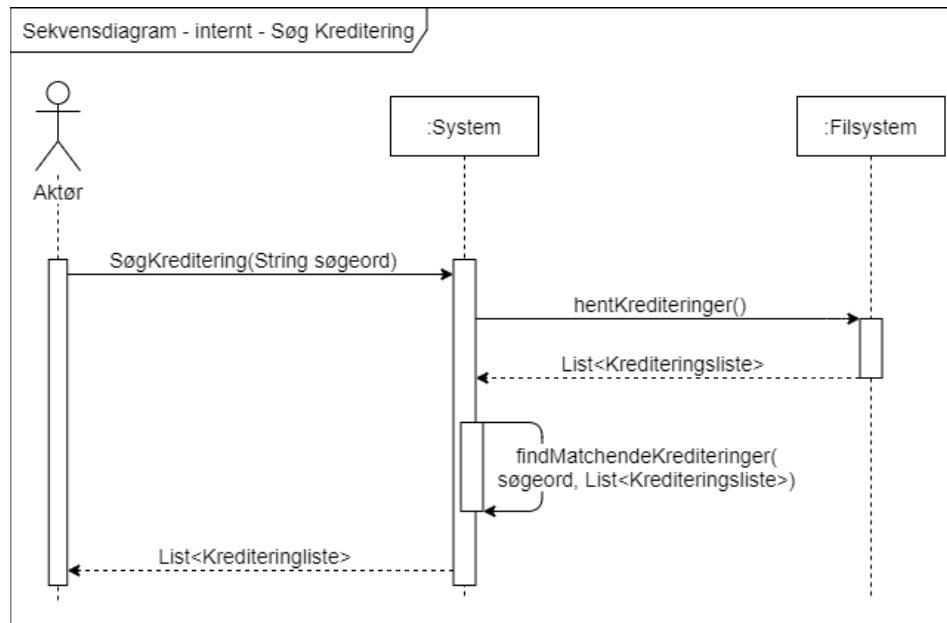
Interne sekvensdiagrammer, for system operationer, modellerer de hændelsesforløb der sker internt i vores system, samt hvordan de forskellige fundne analyseklasser kommunikerer med hinanden. Formålet med at modellere sekvensdiagrammer for interne operationer, er at give overblik over ansvaret for de klasser der samarbejder, samt identificere metoder.

Aktøren initierer operationen SøgKreditering, som kalder ned på vores System med en string som parameter, der repræsenterer de søgeord aktøren har indtastet. Systemklassen laver et kald til vores lokale Filsystem med operationen HentKrediteringer, som returnerer en liste med krediteringslister. Her er det valgt at hente alle kreditering fra filsystemet op til selve systemet, hvorefter sortering ud fra søgeordene sker på Systemet.

Dette kunne i fremtiden blive et problem, afhængigt af hvor meget data Filsystemet indeholder. Et mere effektivt alternativ ville være at søgningen/sorteringen foregår/implementeres i Filsystemet. Vi vil herved undgå at hente store mængde data op, og ligge store mængder data i vores domænelag (System), men blot kan sende de matchende krediteringer fra Filsystemet op

TV 2 Krediteringsprogram

til systemet. Dette blev vi gjort opmærksom på i vores overvejelser og tog det i betragtning i 2. iteration, hvor persistenslaget håndterer søgefunktionaliteten.



Figur 7: Sekvensdiagram - internt - søg kreditering

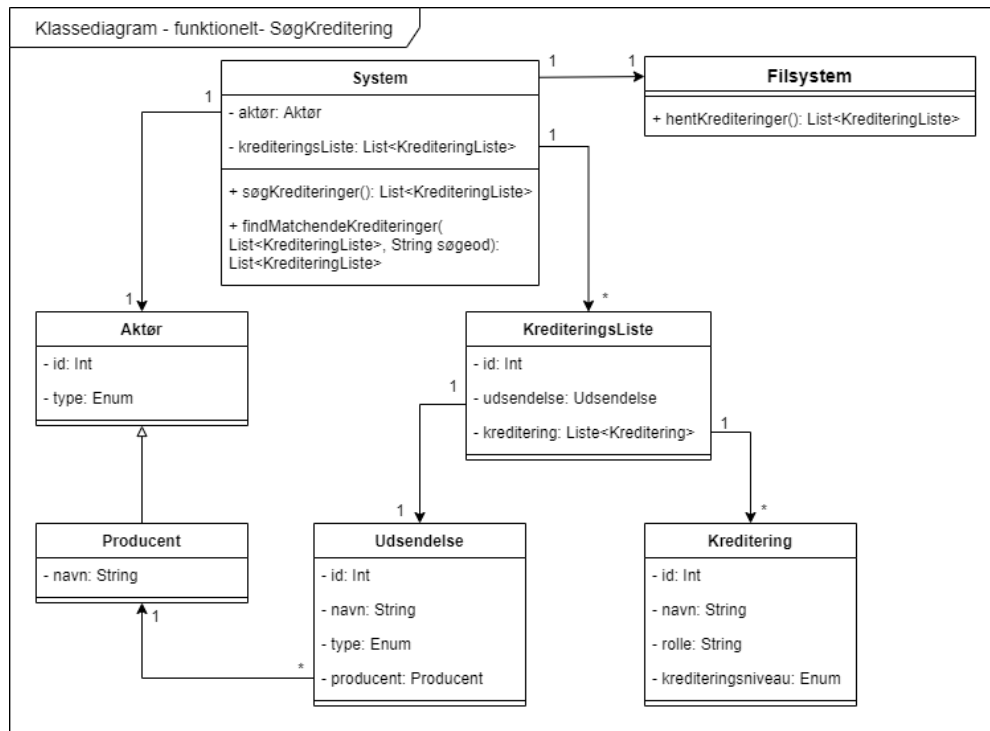
De fundne krediteringslister der matcher søgeordene returneres til sidst til aktøren og operationen er nu afsluttet.

Næste skridt i workflowet er at opdatere analyseklassediagrammet, og koble det sammen til et enkelt. Med udgangspunkt i analyseklassediagrammet og sekvensdiagrammer vil der blive konstrueret et design klassediagram, som viser den fulde funktionalitet vedrørende det bearbejdet brugsmønster.

Design klassediagram

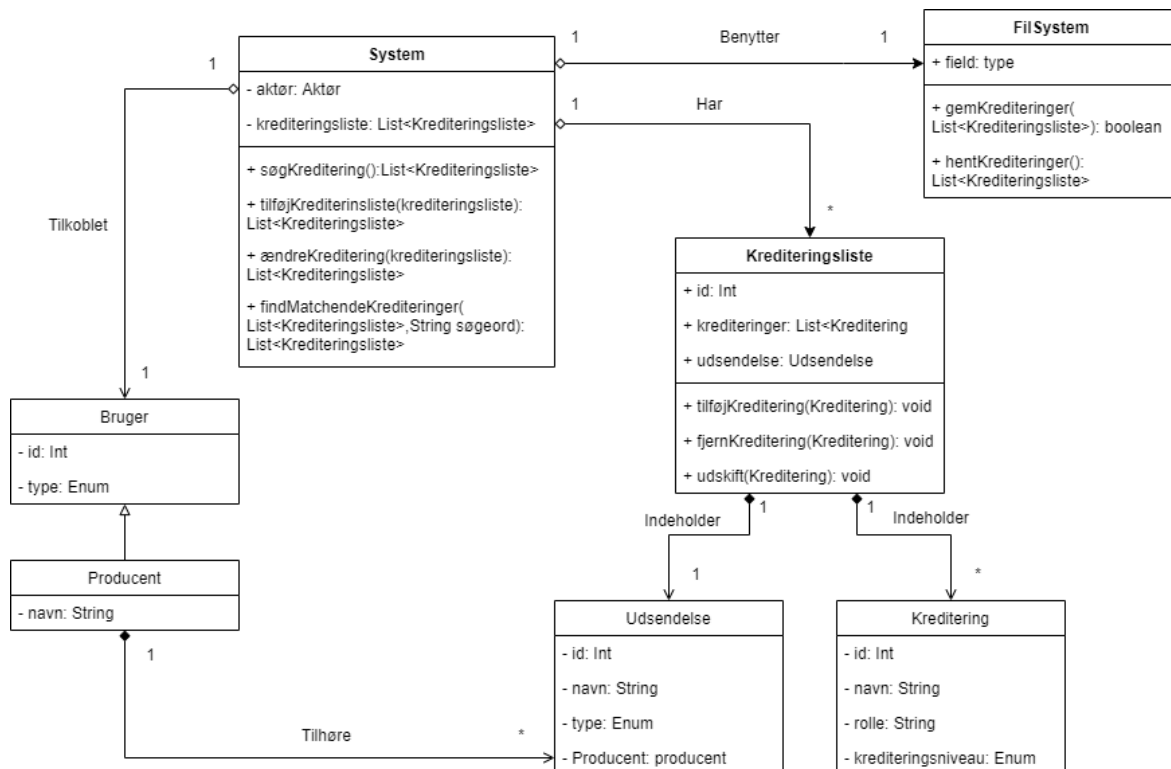
Med udgangspunkt i brugsmønster realisering og sekvens diagrammer, konstrueres et design klassediagram for "søg kreditering" brugsmønstret. Her er der tilføjet funktionaliteten fra sekvensdiagrammerne til analyseklassediagrammet fra figur 3.

TV 2 Krediteringsprogram



Figur 8: Designklassediagram - søg kreditering

Efter design af alle de tre brugsmønstre "søg kreditering", "tilføj kreditering" og "ændre kreditering" (se bilag J), er der konstrueret et samlet UML klassediagram som viser de samlede entiteter med deres funktionaliteter.



Figur 9: Designklassediagram - brugsmønsterrealisering

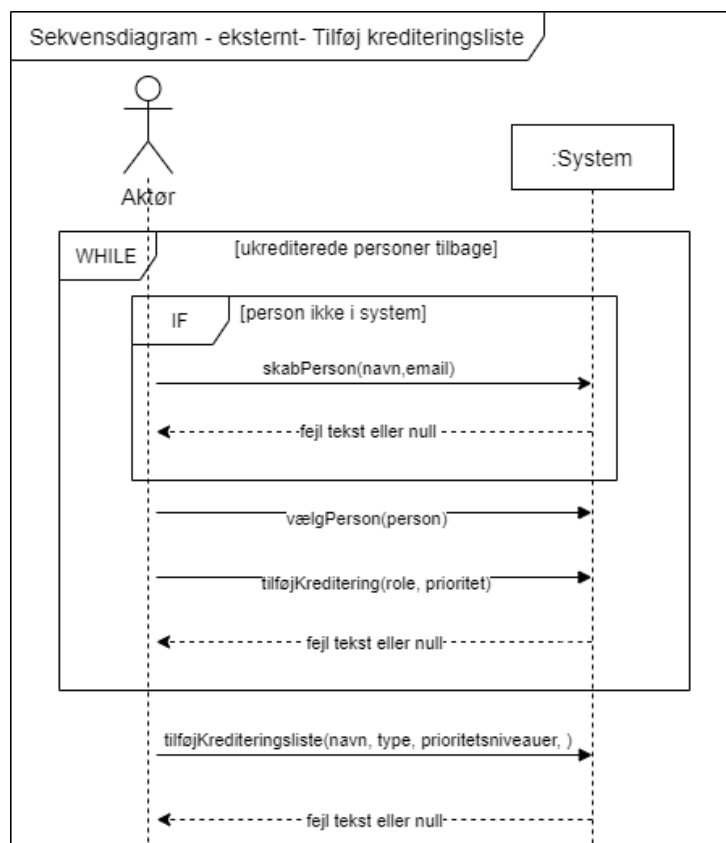
TV 2 Krediteringsprogram

Designet bygger på at ét system, som er koblet til et filsystem, har en bruger og indeholder flere krediteringslister. Disse krediteringslister indeholder en udsendelse og de krediteringer, som er sammenhængende med denne udsendelse. Udsendelsen er samtidigt forbundet til en bestemt producent. Der benyttes arv, da en producent nedarver grundlæggende funktionalitet fra en bruger, hvilket i først iteration kun tænkes at være hvilken type bruger man er.

Anden iteration

Ved anden iteration er der arbejdet med design af prioriteterne, fra prioriteringslisten, se tabel 5, punkt 5-12. I anden iteration blev de fleste punkter designet og implementeret, men punkterne 9, 10 og 11 blev fravalgt. De blev fravalgt, da de blev vurderet til ikke at ville være optimale at implementere, fordi disse funktionalitet var unødvendige for det grundlæggende produkt.

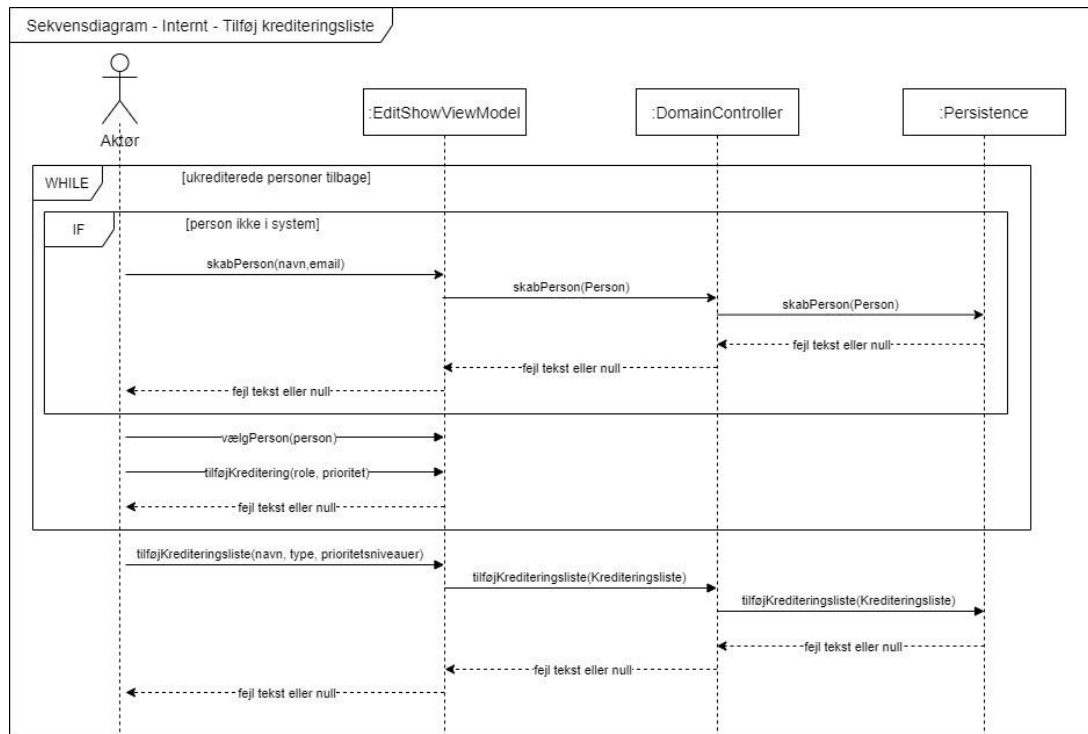
Der er lavet sekvens diagrammer for prioritering 6: "Systemet skal kunne merge/linke forskellige krediterede navne, så de peger på den samme unikke person". Der er først oprettet et eksternt sekvensdiagram, for at vise den grundlæggende interaktion med systemet, når man ønsker at tilføje en kreditering. Sekvensdiagrammet viser, at man kan tilføje flere personer hvis det ønskes. Først tjekkes det, om den person man prøver at kreditere eksistere i systemet, hvis personen ikke eksistere skal personen oprettes med navn og e-mail. Efter man har valgt en person, angiver man rolle og prioritet. Når man er færdig med at tilføje personer, kan man tilføje krediteringslisten.



Figur 10: Ekstern sekvensdiagram – "Tilføj krediteringsliste"

TV 2 Krediteringsprogram

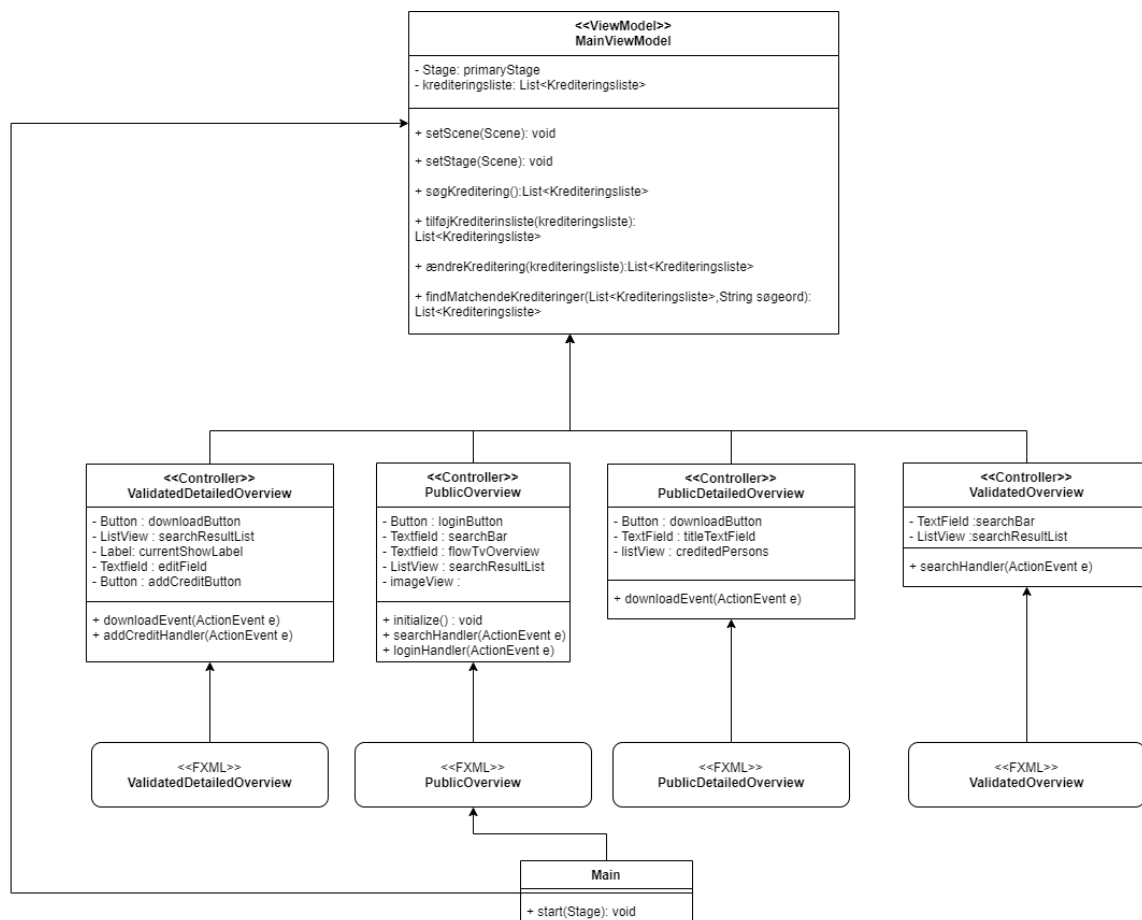
Efter det eksterne sekvensdiagram er der lavet et internt diagram, for at kunne designe den helt konkrete funktionalitet mellem brugeren og de forskellige komponenter i systemet. Det interne program har herved interaktion med brugeren igennem en EditShowViewModel, som sender information vider til Domain Controller, og til sidste sendes information til persistens laget. Hvis noget går galt undervejs, returneres en fejl besked ellers returneres null.



Figur 11: Internt sekvensdiagram – "Tilføj krediteringsliste"

Med udgangspunkt i det interne sekvensdiagram, er implementeringen påbegyndt. Mere design for de forskellige prioriteringspunkter kan ses i bilag H. Efter arbejde med de forskellige brugsmønstre, prioriteret ud fra prioriteringslisten, er der udarbejdet et grundlæggende design klasse diagram, som viser hvordan præstationslaget skal konstrueres. Der vil benyttes Viewmodels til at holde på funktionaliteten i præsentrationslaget, hvorefter der benyttes controllers til at sammenkoble funktionaliteten mellem FXML og Viewmodels.

TV 2 Krediteringsprogram



Figur 12: UML klassediagram for præsentationslaget

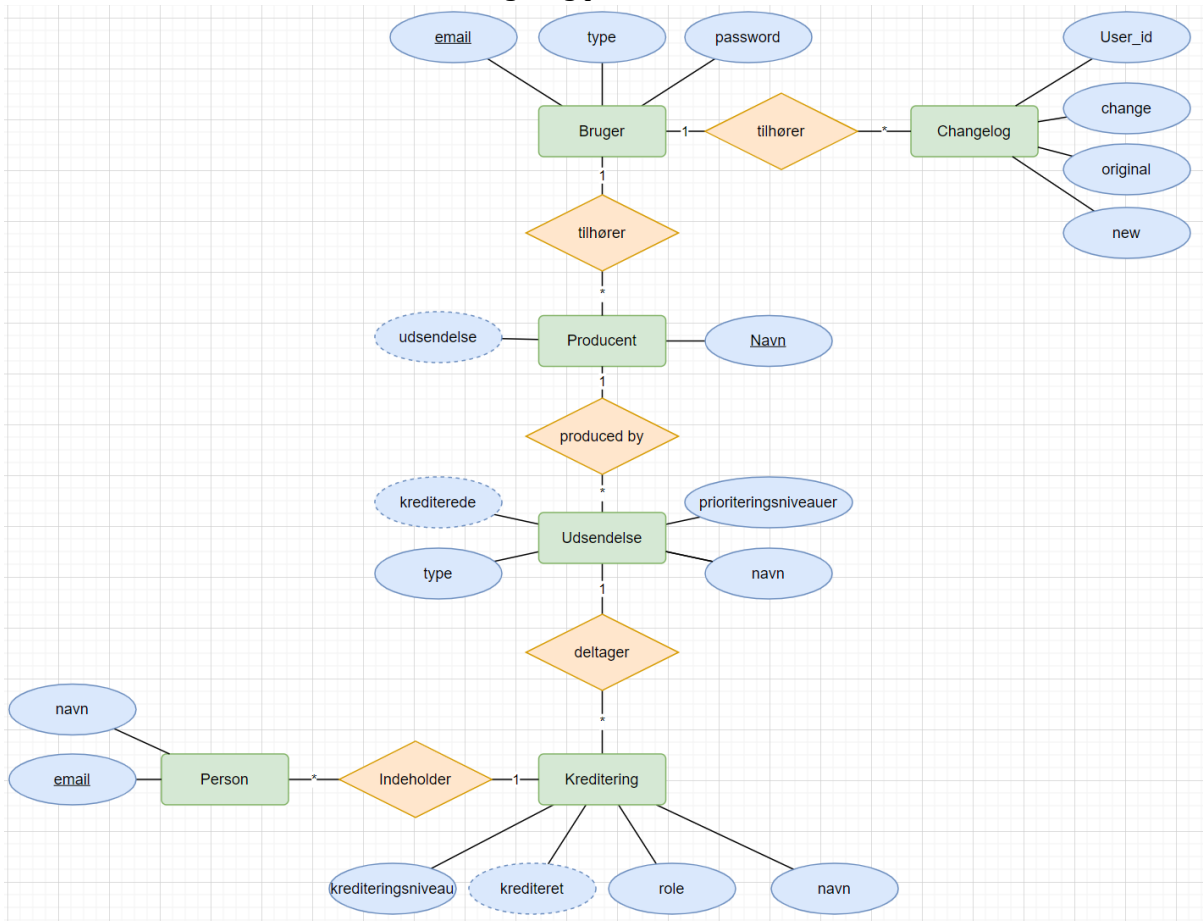
Her kan ses dette klassediagram, som kobler FXML dokumenter til controllere som er koblet sammen med MainViewModel. Dette er opsat for at skabe en separation mellem brugergrænsefladens elementer og dens funktionalitet.

TV 2 Krediteringsprogram

4.5. Databasedesign

4.5.1. EER

EER giver et konceptuelt billede af databasen, og er derfor et analyseartefakt. Det er dog tæt forbundet med det videre databasedesign, og placeres derfor herunder for nemmere kontekst.



Figur 13: EER-diagram

Før oprettelsen af databasen, i koden, dannede vi EER for at skabe os et overblik over hvordan hele databasen skulle se ud. Det ses at dataene er meget simpelt forbundet, da forbindelsen er fuldstændig lineær.

4.5.2. Normalisering

Tabeller i en database kan deles op på forskellige måder. Dette bliver kaldt for normalisering. Gennem normalisering kan der undgås redundans og inkonsistens, eller mangel på sammenhæng af forskellige data. Vi undgår hermed en masse null værdier i tabellerne gennem insertion, deletion, og update anomalies. Man vil kalde en rational database for normaliseret, når den mindst opfylder kravene for tredje normal form.

Ved **1. normalform** (Atomicitet) må en kolonne ikke indeholde data, der kan separeres logisk. Det kan ses på dataene i databasen, at ingen kolonner indeholder logisk deleligt data, og overholder derfor 1. normalform.

TV 2 Krediteringsprogram

Ved **2. normalform** (funktionel afhængighed) skal alle kolonner i en tabel afhænge af den samme eller de samme primary keys. Det kan også ses fra databasen, at alle kolonner afhænger af den samme primary key, og overholder derfor 2. normalform.

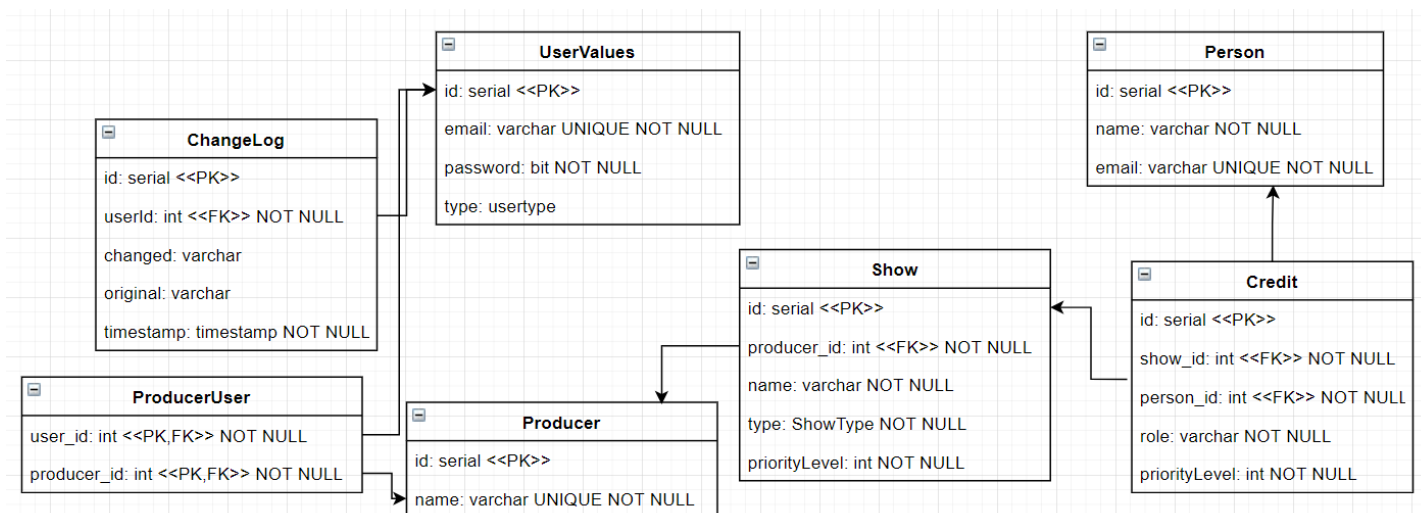
Ved **3. normalform** (ingen Transitiv afhængighed) må ingen kolonner i en tabel være indirekte afhængig af en primary key. Det vil sige at, en kolonne må kun være afhængig af sin primary key og ingen andre kolonner. I databasen er der en transitiv afhængighed i uservalues, hvor "password" afhænger af "e-mail", selvom id'et er sat som primary key. Dette er valgt, for at en bruger kan ændre sin e-mail, uden at brugeren bliver skiftet ud. Ellers overholder vores database kriterierne for 3. normalform.

Ved **Boyce-Codd normalform**, dannes en *super key* som er en kolonne der kan refereres til andre kolonner. En super key er en kolonne uden duplikationer. Det er sjældent at en 3. normalform tabel ikke rammer kriterierne for BCNF.

Ved **4. normalform** (ingen multi-afhængighed) undgår vi multi-afhængighed. Det vil sige, at værdier fra en eller flere kolonner, ikke må referere til en eller flere kolonner. Det kan ses at vi overholder 4. normalform i vores database.

Hele vores database bliver normaliseret for at formindske data redundans, forsimple eventuelle queries, og gøre det nemmere at få fat på data. Vores system har mange af de samme brugere, som har den samme slags rolle. Derfor giver det mening at splitte tabellen op, så vi ikke skriver den samme rolle på flere brugere, og i stedet referere det gennem en anden tabel. Hermed undgår vi også eventuelle fejl eller store tab af data, hvis vi skulle miste noget. Vi har vores changelog som en separat tabel, for det er ikke alle brugere som har adgang til at ændre i tabellerne.

Ud fra EER-diagrammet og normaliseringsprincipperne er der konstrueret et UML klassediagram, som viser den færdige opsætning for databasen.



Figur 14: UML klassediagram for databasen

TV 2 Krediteringsprogram

4.6. Implementering

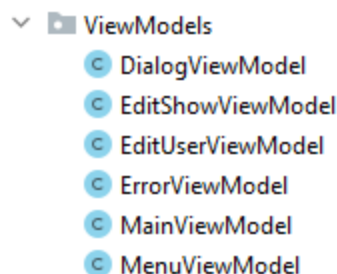
4.6.1. Controllers

UI-designet angiver hvor mange UI-pages der skal være i programmet og til hver page, skal der være en controller. Under implementeringen blev det dog klart, at den funktionalitet programmet krævede, havde brug for flere UI-pages end hvad der var designet. Dette resulterede i langt flere controllers, og herved også langt mere kommunikation på tværs af controllers (Se bilag N).

4.6.2. Viewmodels

Det større antal controllere, samt den voksende kommunikation på tværs af dem, skabte behovet for nye typer klasser i præsentationslaget. Disse klasser blev implementeret i anden iteration og blev kaldt ViewModels. De benyttede singletonmønsteret, så controllerne nemt kunne få delt adgang til dataene i Viewmodels. DomainControlleren fungerede allerede som en måde at dele data mellem controllere, men dette måtte ikke være data specifik for præsentationsimplementeringen. Viewmodels overtog derfor rollen for at holde data og logik specifik for præsentationslaget og delt mellem controllere. Viewmodels giver en vis separation mellem domæneklassen og controllerne. Der er ikke skabt direkte separation mellem controllerne og domæneklassen, men det kunne der blive. Dette ville skabe et 4. lag. Dette viewmodel lag ville holde data og logik bundet til præsentationen, men er for generelt til at placere i controllere. Laget ville ikke kunne understøtte en anden præsentationsteknologi, men ville tillade nemmere ændringer af controllerlogikken. Herved ville controllerne være reduceret til kun at indeholde logik og data helt tæt forbundet med dets view, og ikke andet.

Ud over deling af data og logik over semantisk ens controllere, er der også oprettet viewmodels til at styre globale UI-elementer. Dette er eg. DialogViewModel, der gør det muligt for alle controllere, at anvende den samme globale dialog. Der er også ErrorViewModel, som blot kan gives en streng, hvorefter den viser en fejlmeddelelse. MenuViewModel, tager sig af menu-logikken, og tillader alle controllere at sætte MenuItems med et ikon, en tekst og en OnClick-funktion. Herudover er der MainViewModel, der primært tager sig af at navigere mellem UI-pages, men også holder "SelectedCreditList" der anvendes over hele programmet. Dette er et stykke information der deles på tværs af semantisk ens controllers, og er derfor nødt til at placeres i en global Viewmodel.



Figur 15: ViewModels fra præsentationslaget

TV 2 Krediteringsprogram

Herover ses de viewmodels der anvendes. Der er 4 globale viewmodels og 2 lokale. De lokale er EditUserViewmodel og EditShowViewmode. Disse styrer sekvensen af, at oprette og redigere henholdsvis shows og viewmodels.

```
public class EditCreditListController implements Initializable {
    [...]
    private final MainViewModel mainViewModel = MainViewModel.getInstance();
    private final MenuViewModel menuViewModel = MenuViewModel.getInstance();
    private final ErrorViewModel errorViewModel = ErrorViewModel.getInstance();
    private final EditShowViewModel viewModel = EditShowViewModel.getInstance();
    private final DialogViewModel dialogViewModel = DialogViewModel.getInstance();
    [...]
}
```

Figur 16: Klasseeksempel - brug af Viewmodels

Herover ses et eksempel på en controller der anvender alle globale viewmodels, samt én lokal. Controlleren har altså brug for funktionaliteten placeret i alle disse viewmodels.

4.6.3. SQL mapping

Under implementeringen af databasen, opstod et problem med mapping fra sql til Java. Dataene modtaget fra databaselaget, var organiseret efter tabellerne i databasen og her var dataene organiseret anderledes end i Java-entiteterne. Dette var ikke håndteret i designet, og blev derfor håndteret reaktionært. Løsningen var et 3. mapping lag. Der blev skabt entitetsklasser modelleret direkte efter sql tabellerne. På den måde kunne dataene fra databasen konverteres direkte til et håndterbart Java-objekt, der herefter kunne konverteres til en mere Java-orienteret struktur.

```
private CreditPersist creditFromCreditSql(CreditSql creditSql){
    PersonSql personSql = PersonSql.fromMap(template.queryForMap(
        "SELECT * FROM Person WHERE id =:id", Map.of("id", creditSql.getPersonId())));
    PersonPersist person = new PersonPersist(
        personSql.getId(), personSql.getName(), personSql.getEmail());
    return new CreditPersist(
        creditSql.getId(), person, creditSql.getRole(), creditSql.getPriorityLevel());
}
```

Figur 17: Java funktion - creditFromCreditSql

Herover ses et eksempel på mapping-strukturen. Returdataene fra databasen mappes direkte til et Java objekt, hvorefter det mappes videre via. en funktion kaldt creditListsFromShow.

4.6.4. Hashing

Efter implementering af databasen og et simpelt login system, hvor password bliver gemt som ren tekst, blev det tydeligt at der manglede en form for hashing. Fordelen ved at hashe password er at det ikke længere bliver muligt at aflæse password direkte i database. For at løse dette problem blev der benyttet en PBKDF2 hashing algoritme med salt. Salt er en tilfældig værdi, som benyttes ved envejs hashing funktioner, for at tilføre et ekstra lag af sikkerhed. Det blev besluttet at gemme det hashede password og det tilhørende salt, for et givent password, i samme celle i databasen. Databasen holder derfor en 32 byte værdi, hvor de første 16 byte er det hashede password og de sidste 16 er det tilhørende salt. Der blev oprettet en PasswordHashing klasse i persistens laget, som benyttes af UserDom objects til at hashe deres password, når der oprettes

TV 2 Krediteringsprogram

en ny bruger. Samtidigt benyttes hashing algoritmen til at tjekke om passwordene matcher, hver gang man prøver at logge ind. Hashing algoritmen har været forholdsvis simpel at implementere, da Java har nogle biblioteker, som håndterer hashing processen. Det er en nødvendighed at have hashede passwords i sin database, og dette ville optimalt have været en designbeslutning, men problemet blev først tydeligt efter påbegyndelsen af implementeringerne vedrørende databasen og login.

```
private final SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");

public byte[] hashPassword(String password) throws InvalidKeySpecException {
    SecureRandom random = new SecureRandom();
    byte[] salt = new byte[16];
    random.nextBytes(salt);
    KeySpec spec = new PBEKeySpec(password.toCharArray(), salt, iterationCount: 65536, keyLength: 128);
    byte[] hash = factory.generateSecret(spec).getEncoded();
    byte[] combined = new byte[hash.length + salt.length];
    for (int i = 0; i < combined.length; ++i)
    {
        combined[i] = i < hash.length ? hash[i] : salt[i - hash.length];
    }
    return combined;
}
```

Figur 18: klasse - Password hashing

Herover ses hvordan et PasswordHassing objekt kan hashe et password, hvor den får et string password som parameter. Der benyttes javax.crypto pakken til at skabe et SecretKeyFactory object, som giver mulighed for nemt at kryptere en string værdi ud fra en valgt krypterings metode.

4.6.5. Søgelogik

I første iteration blev datene gemt i et filsystem. Det ville være nødvendigt for Java-koden at søge alt teksten i filerne igennem for at søge efter specifikke ting. For at gøre dette nemmere, blev alt dataene læst og mappet til Java-objekter og herefter, blev der søgt i listen af Java-objekter. Dette var en design beslutning. I anden iteration blev det valgt, også i design, at søgningen nu skulle foregå i sql databasen. Dette ville være langt mere effektivt, men da der allerede var en søgefunktion, blev den mere effektive sql-søgefunktion nedprioriteret. I lang tid blev det gamle system derfor anvendt, og først mod slutningen, blev søgefunktionen endeligt oprettet og implementeret som en funktion i databasen.

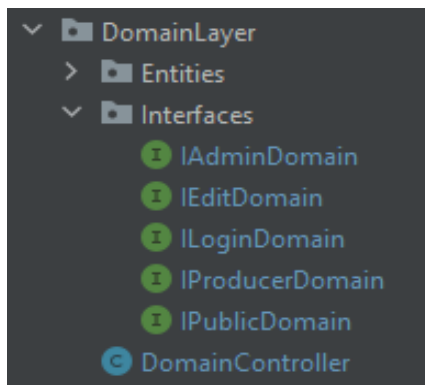
TV 2 Krediteringsprogram

```
CREATE OR REPLACE FUNCTION checkMatch(valueWords VARCHAR, searchWords VARCHAR)
  RETURNS boolean AS $$
  DECLARE matching boolean = true;
  DECLARE searchWordList text[] = string_to_array(searchWords, ' ');
  DECLARE counter INT = 1;
BEGIN
  WHILE matching AND (counter <= cardinality(searchWordList))
    LOOP
      matching = LOWER(valueWords) LIKE
concat('%', LOWER(searchWordList[counter]), '%');
      counter = counter + 1;
    END LOOP;
  RETURN matching;
END; $$
LANGUAGE plpgsql;
```

Figur 19: SQL-funktion - søgefunktion

Funktionen tager 2 strings som parametre. Den første parameter, er de værdier man vil søge på fra en tuple. Den anden parameter er de søgeord som brugeren har tastet ind. Det tjekkes, at hvert søgeord, separeret af mellemrum, er indeholdt i den første parameter. Dette er en lidt ressourcekrævende funktion, men en meget effektiv søgemetode. Funktionen har en 1-til-1 funktionalitet med søgefunktionen først skrevet i Java.

4.6.6. Interfaces



Figur 20: Domæne interfaces

Det blev i starten valgt, at der skulle implementeres 3 interfaces i domænelaget, der kunne separere logikken for de 3 brugertyper. Under implementeringen blev det dog åbenlyst, at producer og admin delte meget logik. Dette kunne have været adskilt, men dette ville resultere i dobbelt arbejde. Derfor blev oprettet et fjerde interface, beregnet til de controllere der skulle servicere både producer og admin. Dette holder stadig domænefunktionaliteten adskilt, for at sørge for, at funktionalitet ikke bliver tilgængelig for de forkerte brugere i de forkerte controllere.

TV 2 Krediteringsprogram

4.7. Test

Igennem processen er der udført forskellige test for at sikre et stabilt program. Dette er udført både i løbet af implementeringen og ved afslutningen for at teste, at programmet opnår de krav der er stillet.

4.7.1. Udviklingstest

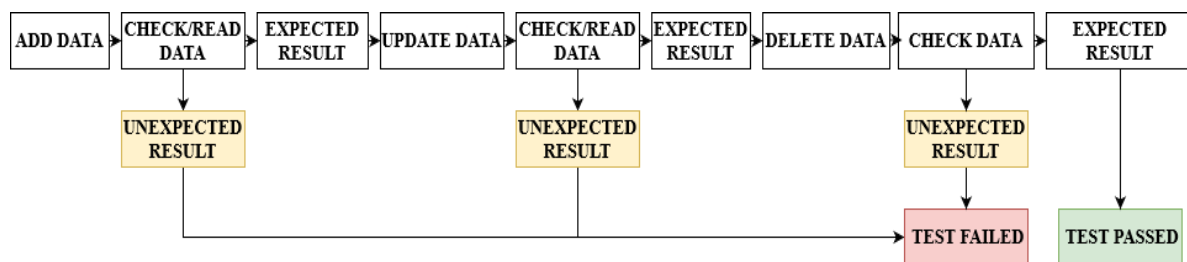
Formålet med udviklingstesten er at opfange uventede fejl under udviklingen, så de kan håndteres for at øge pålideligheden af systemet. Udover opfangning af uventede fejl forsøger man at validere, at den ønskede funktionalitet er opnået. En af fordelene ved separation mellem lagene, samt vi følger det objektorienterede paradigme, er, at debugging af individuelle klasser bliver mere effektivt.

Unit test af database

Kontinuerlig testning og validering af individuelle klassers funktionalitet udføres ved hjælp af Java-frameworket Junit. Selvom det ville være ideelt at opstille Unittest for samtlige funktioner, ville dette være et stort ekstra arbejde, der ikke blev prioriteret i implementeringen. Der er derfor kun opstillet tests på databaselaget, primært for at verificere at CRUD-operationer kan fungere uden at returnere fejl. Testen blev opbygget af 5 testmetoder, som tester på databasens:

- Persons
- Users
- Producers
- Credits
- Changelogs

Alle tests følger samme fremgangsmåde/proces for indsætning, læsning, opdatering og sletning, samt at tjekke forventede værdier. En visualisering af testen er fremvist ved figuren:



Figur 21: Unittest workflow

kontinuerlig system testing

Vi testede kontinuerligt hele programmet for observerbare fejl, ved blot at køre programmet under udvikling. I takt med at GUI'en blev udviklet, samtidig med de andre dele af programmet, blev der løbene testet nye tilførte funktionaliteter. Denne løbende test af funktionaliteter sikrer, at der er hele tiden, er et fungerende system.

TV 2 Krediteringsprogram

4.7.2. Udgivelsestest

Før programmet skal udgives, eller i dette tilfælde færdiggøres og afleveres, er der udførte en udgivelsestest. Det er vigtigt at gå hele systemet igennem, ved at bruge systemet, som det ikke er tiltænkt, for at finde eventuelle bugs, som skal fikses. Det er en god måde at finde fejl, som man ikke ville have fundet ved hjælp af standard test. Udgivelsestesten er en god måde, at teste for forskellige adfærd fra systemet, under usædvanlige forhold. Der blev testet ved, at hvert gruppemedlem brugte en time på at bruge så mange funktionalitet som muligt. Testen er god, fordi vi er mange, som tester fra flere vinkler, så vi får en bred vifte af testscenarier. Denne test er lavet da man mange gange oplever, at folk benytter systemer, på måder som det ikke var tiltænkt. Vi kommer de her udfordringer i møde, ved selvtest, så vi kan prøve at finde de forskellige huller i systemet og rette dem. De fejl, som man finder i de her typer test, kan være svære at finde i det andre test, som vi har lavet, fordi man prøver at bruge systemet på en ikke tiltænkt måde. Se et udsnit af en af gruppemedlemmernes test af systemet i bilag L.

4.7.3. Brugertest

Et stort fokuspunkt i produktet er at lave et simpelt produkt. For at teste dette blev der udviklet en brugertest (se bilag M). Testens formål var at vurdere hvor intuitivt produktet var blevet, samt at få lavet en simple acceptance test, hvor en person uden for gruppen tester, at alle funktionaliteter virker. Testen var delt i 2, hvor den første del, opstillede nogle opgaver, som testpersonen skulle forsøge at løse inde i programmet. Personen blev herefter givet point ud fra, hvor nemt vedkommende havde ved at løse opgaverne. Anden del var et kort spørgeskema, der skulle undersøge testpersonens forståelse, for de ordvalg der var blevet anvendt i programmet.

Vores test, skal teste funktionaliteten i vores system. Det vi kan se, er at vores testperson har nemt ved at guide sig rundt i systemet. Den normale funktionalitet, som at logge ind og ud, kan test personen nemt, samt at lave de mindre administrative opgaver. Men når vi kommer til opgaver, som "ændre email og password på tilføjet bruger.", så kan vi se problemerne opstår. Det er tiltænkt at systemet skal være nemt at bruge, men at de lidt svære administrative opgaver og producent opgaver, skal læres af brugeren, da det ikke er almindelige bruger, som skal have adgang til den funktionalitet.

Testen blev kun udført på én person, og der kan derfor ikke drages konklusioner om programmets brugervenlighed. Testen har dog som en acceptance test påvist at de grundlæggende funktionaliteter i programmet virker, da testpersonen efter fejlforsøg blev guidet til en korrekt gennemgang af testene. Det kan nævnes at personen selv kunne komme igennem de fleste opgaver, og fejlede kun på baggrund af semantiske misforståelser, samt en overset scrollbar. Testformen er dog anvendelig, da den kan gentages for mange individer, hvorved resultaterne nemt kan sammenlignes. Programmets brugervenlighed kan også udvikles, hvorefter den samme test foretages på nye mennesker, for at undersøge hvorvidt ændringerne har forbedret eller forværret brugervenligheden.

5. Diskussion

5.1. Brugernormalisering

Under normaliseringen af brugere, og deres rettigheder, opdagede gruppen at der var flere muligheder at anvende, med hver deres fordele og ulemper. Den første mulighed omhandlede, hvorvidt man skulle lave én tabel med brugertyper og én tabel med rettigheder, hvorefter man sammenkobler de forskellige brugere til de rettigheder, de skal have. Fordelen ved dette er, at det er nemt at lave nye brugertyper og nye rettighedstyper. Systemet kunne også udvides med en personlig rettighedstabel, der kunne give en specifik bruger en ekstra rettighed, hvilket dog gik ud over brugerens brugertype. Der kunne også tilføjes tidsbegrænsede rettigheder. I vores system havde brugeren `ProducerUser` dog brug for et ekstra stykke information "Producer", som `AdminUser` ikke havde brug for. Med det ovennævnte brugersystem, ville man ikke kunne sikre sig, at dette stykke information fandtes for alle producerusere i databasen. I stedet blev et andet system brugt. Her blev oprettet en separat tabel for hvor brugertype, så hver brugertype kunne have ekstrainformation i tabellen. Dette reducerer dog systemets fleksibilitet, da nye brugertyper skal oprettes i databaseskemaet, frem for blot at tilføje ny data til databasen.

5.2. Data hentning

Det blev besluttet, at vores data skulle ligge online på en server, så alle i projektgruppen havde adgang til den samme database. Næsten hver gang man skifter side i programmet, hentes der data fra databasen. Dette er ikke særligt effektivt, og vil med tiden skalere dårligt afhængigt af mængden af data der skal hentes. Et alternativ til dette, ville være at hente en vis mængde data op og gemme lokalt. Caching kunne være et alternativ her, da det tillader at man mere effektivt genbruger tidligere hentet data. Ulempen ved at benytte lokalt gemt data, er at det bruger ressourcer på den lokale disk, samt data man har hentet, kan være blevet forældet/opdateret undervejs. Der er derved en balance mellem altid at have det nyeste data, samtidigt med at man ikke bruger unødige data. Der er mange måder at afhjælpe dette på. I stedet for at bede om ny data, kan man query databasen, for hvorvidt dataene har ændret sig siden sidst de blev hentet. Man kan også sætte et interval, for hvornår dataene skal opdateres.

En anden mulighed er at give brugeren en knap der opdaterer dataene, så de selv bestemmer hvornår det er relevant at opdatere. En måde at spare data på, er også at begrænse antal tuples, der bliver hentet fra databasen. Dette vil være nødvendigt at implementere, for at sikre skalerbarhed. At spare på dataoverførsler blev dog ikke prioriteret i projektet, da det ikke var valgt som et fokuspunkt i kravanalysen.

5.3. Det 4. lag

Et af de ikke-funktionelle designkrav for projektet lød på, at vi skulle have en 3 lags arkitektur, der adskilte præsentrations-, domæne-, og persistens-lag. Dog opdagede vi, at mængden af logik i præsentrationslaget, med fordel kunne adskilles, for at vedligeholde den lave kobling, og undgå klasser med for meget ansvar og funktionalitet. Fordi gruppen var pålagt dette arkitekturkrav, kunne den ikke introducere et 4. lag mellem vores domæne og præsentrationslag, for at adskille yderligere. Dette afspejles i vores præsentrationslag, som kan virke meget stort. I stedet er

TV 2 Krediteringsprogram

størstedelen af den logik, der ønskes adskilt, blevet placeret i nogle Viewmodels. Ved anvendelse af et 4. lag, ville præsenteringslaget indeholde logik og data, mere specifikt rettet mod, at få UI-elementer til at køre. Større logik rettet mod UI-koordination, ville så blive placeret i det 4. lag. Dette ville binde det 4. lag meget tæt med præsenteringslaget, og ville have svært ved at tillade et skift til en helt anden præsenteringslag type. Dog ville det gøre ændringer af det eksisterende præsenteringslag nemmere.

5.4. Objekt mapping

Når vi læser/skriver data, er vi nødt til at transportere vores information, fra et lag, til et andet. Dette kræver at vi mapper vores objekter, til det tilsvarende lag de skal fragtes til. Dette bliver gjort for at sikre lagdelingen, så præsenteringslaget ikke bliver afhængigt af persistenslaget. Det kan også give mening, da entitetsklasser i forskellige lag, ofte tillægges eller fratages data, ift. entitetsklasser fra andre lag. I vores projekt er entitetsklasserne dog ens, og mapping er derfor kun brugt for at sikre lagdeling. Et alternativ kunne derfor være at adskille entitetsklasserne til deres eget lag. Ændringer i entiteterne, ville så skabe ændringer i samtlige lag, men dette er allerede tilfældet, da alle entitetsklasser skal være ens i det nuværende projekt. Det ville skabe en høj kobling med entitetsklasserne, som programmører skulle være mindfule omkring. Dog vil projektet være nødt til at ændre sig meget, før denne ændring ville føre til et reelt problem.

TV 2 Krediteringsprogram

6. Konklusion

Opsummering og diskussion af resultaterne gennemgås i følgende konklusionsafsnit, såvel som besvarelse af problemformuleringen. Problemdomænet blev analyseret på TV 2's case, hvor gruppen kom frem til følgende problemstilling.

TV 2 bruger 30 sekunder på kreditering i slutningen af hvert program, blev disse 30 sekunder brugt på reklame i stedet, ville det, ifølge TV 2, give en ekstra omsætning på 60 mio. pr. år. Der ønskes, derfor et krediteringssystem udviklet, som kan dække reglerne for kreditering, uden at vise dem i sendetiden.

6.1. Inceptionsfasen

Inceptionsfasen havde til formål at give et generelt overblik over de krav og afgrænsninger, der er for projektet, samt en vurdering af eventuelle kritiske risici og løsningsforslag hertil, for at vurdere om projektet kunne gennemføres.

Der blev udført en overordnede kravanalyse, hvilket resulterede i identificeringen af 3 aktører, 12 funktionelle og 6 ikke-funktionelle krav. Herefter blev der udformet et brugsmønsterdiagram, og 3 brugsmønstre blev detaljeret. En risikoanalyse identificerede 4 tekniske og 6 sociale risici, som blev givet en fatalitetsgrad, sandsynlighedsgrad, samt et løsningsforslag. Kravene blev opsat efter MoSCoW analysen, og blev derefter prioriteret baseret på deres risici, udbytte og forventede tidskrav.

Ud fra resultaterne i inceptionsfasen, blev projektets krav vurderet til at være overkommelige og de identificerede risici kunne undgås.

6.2. Elaborationsfasen

6.2.1. Analyse og design

Under analysen blev der i første iteration udført en Noun/Verb analyse, hvilket resulterede i et analyseklassediagram, samt en liste af funktionskandidater. Dette blev anvendt under design, hvor både analyseklassediagram og funktionskandidater blev anvendt til brugsmønsterrealisering. I anden iteration blev Noun/Verb sprunget over, for i stedet at gå direkte til brugsmønsterrealiseringen.

I den statiske del blev der foretaget en Noun/verb-analyse, hvor kandidater til klasser blev sorteret og udvalgt, samt funktionskandidater blev undersøgt. Strukturen blev fastlagt, hvorefter vi kunne udforme analyseklassediagrammer. Funktionskandidaterne blev overflødige, da adfærden for systemet ville blive nærmere undersøgt i den dynamiske del af analysen.

Den dynamiske del af systemet blev analyseret ved at konstruere sekvensdiagrammer, og herefter kunne vi opstille et konkret UML-klassediagram. Resultatet af det opdaterede diagram var en byggesten fra at gå fra problemdomænet til løsningsdomænet.

For præsentationslaget, er der udarbejdet et klasse-diagram, der viser, hvordan vi styrer de forskellige FXML-elementer ved hjælp af controllers, og en MainViewModel, hvor den primære funktionalitet er placeret. Domænelagets design blev udarbejdet ved hjælp af vores detaljerede

TV 2 Krediteringsprogram

brugsmønstre, hvor funktioner og konkrete typer blev tilføjet til analyseklassediagrammet. Det overordnede design for den lagdelte struktur og hvad lagene indeholdt, blev bygget ud fra et pakkediagram. Diagrammets interfaces gav indblik i hvordan vores kommunikation foregik mellem lagene. Singletondesignmønsteret blev implementeret i de klasser, som der kun måtte findes en enkelt instans af.

Resultatet fra design og analyse, var udvalgte designklassediagrammer, databaseklassediagrammer, anvendte designmønstre, samt et pakkediagram. Dette dannede fundamentet for det videre arbejde i implementeringen.

6.2.2. Implementering og test

Under implementeringen voksede logikken til de klasser, som vi havde designet tidligere. Der blev, derfor tilføjet flere controllere til at håndtere grafikken, såvel, som yderligere tilføjelser af viewmodels. I designdelen blev der ikke taget hånd om kryptering af sensitive oplysninger, hvilket først gik op for gruppen i løbet af 2. iteration. Hashing af sensitive oplysninger, såsom password, blev derfor implementeret for at undgå at gemme dem som Strings/raw text.

Der blev under implementeringen anvendt Unittest, primært til at teste databasen. Resten af systemet blev testet manuelt, ved at køre programmet. Efter begge iterationer, blev der gennemgået en udgivelsestest af gruppen, hvor programmets funktioner blev testet. I sidste iteration blev der derudover gennemført en brugertest, der både testede programmets funktionalitet, og samtidigt vurderede brugervenligheden af programmet.

6.3. Besvarelse af problemformulering

Problemformuleringen opstillet i inceptionsdokumentet var følgende:

"Hvordan udvikles et krediteringssystem, der overholder de nuværende regler for kreditering, gør det simpelt for producere at tilføje og ændre krediteringer, gør det simpelt for TV 2 at verificere og videresende krediteringer, og gør det simpelt for seerne at se krediteringer, med minimal omkostning af sendetid?"

Et af fokuspunkterne i problemformulering lød på, at det skulle være simpelt for alle aktører at bruge systemet. Der testes altså på de funktionelle krav. Dette blev undersøgt i vores brugertest. Resultatet af testen er ikke fuldt repræsentativt, da vores mængde af brugertests kun ligger på ét individ. En producent eller en repræsentant fra TV 2 til brugertest, havde givet væsentligt bedre resultater.

Vi fandt dog, at der ud af de 19 opstillede brugertests, kun var 4 tests brugeren ikke var i stand til at udføre, uden yderligere hjælp. Fejlene kom primært som følge af semantiske misforståelser, altså en mangel på forståelse af programmets kontekst, ikke mangel på intuitivt design. Dette indikerer, men konkludere ikke, et simpelt system.

TV 2 Krediteringsprogram

7. Perspektivering

Vi har lavet en løsning, der gør at man som seer af diverse shows på TV 2 kan slå op i systemet for at finde krediteringer på de shows, som man ser. Denne løsning bidrager til, at TV 2 kan få frirum i deres sendetid, hvilket kan være med til at skabe en økonomivækst, da de ikke skal bruge tiden på at vise krediteringer. Vi har gennem rapporten snakket om vores system, men hvis vi skulle kigge på det fremtidige arbejde i projektet, så ville vi gå gennem punkterne forneden.

7.1. Privat bruger

Vi kunne give privat personer muligheden for at få en bruger i vores system. De private brugere skal have mulighed for at kunne gemme krediteringer i favoritter, så de ikke skal lede efter krediteringer fra deres yndlingsshows. Vores system skal kunne samarbejde med TV 2 Play, så vi kan sammenkoble Play kontoen med vores system. på den måde kan der blive vist krediteringer fra senest sete TV 2 Play program, på forsiden i vores system.

Vores løsning ligger også op til, at man kunne implementere flere funktioner der inkorporerer flow tv. Man kunne lave en QR-kode, som blev vist i hjørnet af et show, hvilket ville videresende brugere til krediteringerne. Den løsning, som vi ville have kigget på fra start, men som vi anså var for svær, var subsoniske bølger. Et program skulle producere subsoniske bølger, der via en app kunne afkodes til et id på en krediteringsliste. Appen kunne herved gå direkte hen til krediteringslisten, fra det show man var i gang med at se. Dette ville virke som en QR-kode, men uden behovet for noget der forstyrrer skærbilledet. Brugere kan godt blive irriteret, hvis de skal igennem for mange steps, for at finde en kreditering. Subsoniske bølger ville løse den problematik, så brugerne har en nem måde at finde krediteringerne på.

7.2. Krediterings bruger

Vi kunne give de krediterede personer en bruger, så de havde nemmere ved at se, hvilke programmer de var krediteret på. Med en krediteringsbruger ville man også kunne anmode om at blive krediteret, hvis der var sket en fejl, som gjorde at man ikke var blevet på et program. Vi vil gerne gøre systemet til et sted, hvor folk kan vise deres krediteringer på deres CV, f.eks., ved at koble systemet sammen med LinkedIn.

7.3. Eksporterings API

Systemets krediteringer skal kunne hentes gennem en offentlig API (står for Application Programming Interface). Den offentlige API vil gøre krediteringssystemet anvendeligt på andre hjemmesider, apps eller smart-tv. API'en skal også være optimalt kompatibel med TV 2 Play, så krediteringerne kunne ses direkte i TV 2 Play appen. Eventuelt kunne TV 2 tage penge for brugen af deres software, og herved få en ekstra indtægt, ved at tilbyde andre det samme krediteringssystem som de bruger. Man vil også kunne gøre systemet kompatibelt med andre API's, for, f.eks., at kunne trække krediteringer fra sider som IMDB (står for Internet Movie Database).

TV 2 Krediteringsprogram

7.4. Krediteringsvalidering

I produktet kan producenter oprette krediteringslister som de vil. Administratorbrugere kan ændre i disse krediteringslister, men de bliver offentlige med det samme. Det ville være optimalt, hvis der blev anvendt en valideringsprocedure. Eg., En producer opretter en krediteringsliste til sit program, men det bliver ikke offentliggjort med det samme. Administratorer får en besked om den nye krediteringsliste, og bliver bedt om at validere den. Hvis de validerer den, bliver den offentliggjort med det samme. Foretager de ændringer i listen, får producenten besked om dette, og skal validere ændringerne. Foretager producenten nye ændringer, får administratoren besked igen. Sådan fortsætter det, indtil begge parter har valideret den samme version af krediteringslisten, og listen bliver herefter offentliggjort.

8. Procesevaluering

8.1. fælles vurdering

8.1.1. Gruppedannelse

Under gruppedannelsen blev det forslået, at hver studerende lavede en "dating" profil, som grupperne kunne dannes fra. Dette virkede teoretisk fornuftigt, men blev ikke udnyttet i praksis, da gruppedannelsen som helhed var valgfri. Corona havde reduceret muligheden for at finde passende grupper gennem socialisering. Mange grupper blev derfor dannet på forhånd, på baggrund af de relationer der blev dannet under første semester, hvilket skabte en gruppe af "resterende" studerende. Her blev ærlighed for at finde mennesker der passede med én, sat på prøve af desperation for ikke at ende i en tilfældigt sammensat gruppe, med mennesker som man ikke nødvendigvis kan fungere sammen med. Trods gruppedannelsesforløbet, blev en god gruppe dannet.

8.1.2. Roller

Som en del af Take-off fasen, i starten af projektet, skulle alle medlemmer af gruppen tage en Belbin rolle test. Resultatet af testen blev så brugt til at fordele team roller blandt gruppens medlemmer, hvor alle fik en hovedrolle og en eller flere suppleant rolle (se bilag C). Værdien vi fik ud af dette, var ret begrænset. Det var en måde at lære lidt om hvordan vi hver især fungerede i en gruppe, og lagde klart op til en diskussion, om hvilke roller vi selv mente at vi opfyldte i en gruppe. Trods denne rolle test, så endte vi mere ud med at indtage forskellige roller naturligt efterhånden som projektarbejdet skred frem.

8.1.3. Tidsfordeling

Vores tidsfordeling var ret god, til tide måske endda for god, især tidligt i processen. I inceptionsfasen endte vi ofte ud med at være så godt med, at vi havde svært ved at fortsætte, da vi manglede den påkrævede undervisning, fordi vi var tidsmæssigt forud.

Vi løb ind i et lignende problem i 1. iteration af elaborationsfasen. Her endte vi igen ud med at være bremset af den manglende undervisning. Her virkede det dog mere som om at vi kom bagud på grund af det, og endte ud med at have ret travlt til sidst fasen for at nå at få implementeret alle de målsatte funktionaliteter.

I 2. iteration af elaborationsfasen var vi dog mere velbalanceret. Da selve processen i 2. iteration mere eller mindre var en gentagelse af 1. iteration, stod vi ikke med samme mangel på viden. Ikke at vi var foran, men vi løb ikke ind i de samme blokader.

Projektets tidsproblemer var ikke i vores mangel på god tidsplanlægning, men mere i at semesterets undervisning tempo ikke passede med projektets tempo.

TV 2 Krediteringsprogram

8.1.4. Vejleder

Rune var en ret god vejleder. Han virkede meget kompetent og erfaren, både inden for hele den her projektproces og inden for softwareingeniørfaget. Han var god til besvarer og guide os, hvis vi havde spørgsmål, både inden for arbejdsprocessen, faglighed eller bare generelt. Det var også fint at han var klar på at være sådan nede på jorden med os. Møderne behøvede ikke bare at være professionelle og kantede, men kunne sagtens være mere afslappede og uformelle, og det faldt gerne en røverhistorie eller to af. Det er ret fedt at have en vejleder der på den måde også er menneskelig.

8.1.5. Formelt review og Midtvejsseminar

Et formelt review kan have mange fordele, da det forsøger at holde sig til at vurderer objektive fejl, mangler og goder. Men når dette review er foretaget af andre studerende, som formentligt har samme vidensniveau som vi selv har, så kan det være svært at vide om kritikken er korrekt. Samtidig kan det være svært for folk at være objektive, især når emnet kan ligge ret tæt på revieweren selv, da vores medstuderende jo skriver et lignende projekt, og derfor kan modstridende synspunkter virke som personlige angreb. Det ville have været en klar fordel hvis reviewet ikke havde været så formelt, så man kunne have en dialog og spørge ind til kritikken. Trods dette, så fungerede det med at 3 medlemmer fra hver gruppe skulle give kritik til 3 forskellige andre grupper overraskende godt. Det gjorde at man kun skulle lave et review af et inceptionsdokument i stedet for dem alle, og det gjorde at man fik 3 personer, der alle havde sat sig godt ind i dokumentet, til komme med kritik, i stedet for 27 personer der kun lige havde skimmet det.

Midtvejsseminaret fungerede i dette semester på samme måde som det gjorde sidste semester. Alle grupperne præsenterede det daværende stadie af deres program, og så skulle der gives feedback baseret på denne præsentation. Ligesom sidste semester var det ret en ret overfladisk kritik der kom ud af det. Det er ret svært at give feedback på et projekt baseret på en kort præsentation af det. Dog gav denne model rig mulighed for at vi kunne blive inspireret af alle de andre gruppers projekt, hvilket nok også er grundideen bag midtvejsseminaret.

8.1.6. Studie

Undervisningen i GSE (Grundlæggende Software Engineering) lagde sig ret tæt op ad projektet. Størstedelen af undervisningen gav grundstenene for hvordan selve projektarbejdet skulle udføres, og var essentielt for projektet. Derfor kunne det til tider også være frustrerende, når undervisningen i dette fag af og til virkede som om at det ikke holdte samme tempo som projektarbejdet. Dette ledte til at projektarbejdet gik lidt i stå, fordi den påkrævede undervisning ikke var blevet givet.

Da et af kravene til dette semesters projekt er at systemet skal udnyttet en database, har DM (Data Management) også været ret essentielt. De fleste af os har ikke rigtig haft arbejdet med databaser før, så undervisning i det har været et behov. DM har dog haft lidt det modsatte problem af GSE. Da databasen først har skulle implementeres i 2. iteration af elaborationsfasen, så har vi ikke

TV 2 Krediteringsprogram

rigtigt, fået muligheder for at udnyttet den viden, før faget nærmest var nået til ende. Men man kan vel samtidig overveje hvorvidt vi så måske havde endt med at designe og implementere databasen forkert, hvis vi have taget den i brug allerede i 1. iteration. Under alle omstændigheder så har faget givet en masse god og brugbar viden til projektet.

Projektet omhandler uundgåeligt at der skal udvikles et kørende program, så undervisning i kodning er åbenlyst brugbart. Og der er da også en hel del af det vi har lært i VOP (Videregående Objektorienteret Programmering) som vi har brugt i projektet. Faget har dog ikke været essentielt på samme måde som GSE eller DM har. Vi kunne formentligt godt have skrevet et program der gjorde det samme uden at bruge noget fra VOP, men det ville nok også ende med at være et program der kørte dårligere og mindre pålideligt.

Generelt set så har fagene på dette semester været meget passende i forhold til projektet. De har alle givet viden som vi har brugt i projektet.

8.1.7. Indlæring

Projektarbejdet er i sig selv et meget praktisk orienteret arbejde, i den forstand at man får anvendt utrolig meget af den teoretiske viden fra de forskellige fag. Det er en god måde at se den teori brugt, og det giver et indblik i hvordan meget af den rent faktisk kan bruges ude i den virkelige verden.

En kompetence som man ikke normalt går og tænker over at man skal bruge, er erfaring med online gruppearbejde. Men eftersom Corona pandemien har gjort, at vi ikke rigtig har kunnet møde fysisk, har vi måtte indfinde os med at skulle arbejde online, og dette er nærmest en hel disciplin i sig selv.

8.1.8. Ændringer

Det er nogle ret åbenlyse ændringer som vi kunne lave i forhold til næste projekt. Der er f.eks. hele det her med først at lære hvordan man laver artefakter, databaser osv. undervejs, i stedet for at have den viden på forhånd. Der er en del af de ting som vi kunne have kommet bedre fra start med, hvis vi havde haft den viden på forhånd. Og med henblik på databasen, så ville vi have undgået en masse dobbelt arbejde hvis vi havde implementeret den fra starten af.

8.2. Individuel vurdering

8.2.1. Troels' vurdering

Dette semesterprojekt har været en prøvelse. Det har fulgt den samme læringsstruktur som sidste år, hvor man først lærte det man skulle bruge til projektet et par dage inden man skal bruge det. Dog har det i år været ekstra besværligt, da der har været meget metode der skulle gennemføres. Jeg fortrækker at tage mig tiden til at lærer noget, før jeg skal bruge det. Det ville have været at fortrække, hvis vi havde udskudt starten af projektet lidt, og derved havde haft en bedre forståelse af hvad det skulle indeholde da vi startede. Min rolle i dette projekt, har også været anderledes end den jeg normalt indtager. Jeg fortrækker normalt at udvælge nogle enkelte ting

TV 2 Krediteringsprogram

jeg kan fokusere meget på og specialisere mig i. En intern ubalance i vidensgrundlag og arbejdsproces, gjorde det nødvendigt for mig at påtage mig en bredere og mere koordinerende rolle. Det har været svært at få sparring omkring programmering, grundet niveauforskelle i gruppen. Dette overlod også en større arbejdsbyrde til mig i programmeringsdelen, hvilket betød at jeg primært har kunnet arbejde med det fundamentale i programmering, og ikke har haft tid til at dykke længere ned i det teknisk udfordrende. De begrænsede programmeringskompetencer, gav også et langt større pres for, hvorvidt produktet kunne blive færdiggjort i en tilstrækkelig stand. Denne stress pressede mig til at gå direkte over til koden, hvilket reducerede tiden brugt på design. Dette semesters projekt har for mig været præget af kaos, stress, og en manglende tilfredsstillelse af slutproduktet.

8.2.2. Daniels vurdering

Dette semesterprojekt har grundlæggende været godt, da jeg har lært en masse nyt, og vi har brugt meget af det vi har lært i undervisningen. Burg af teorien giver et nyt perspektiv til, hvordan disse teoretiske modeller faktisk kan bruges. Der har generelt været et godt samarbejde i gruppen, jeg synes dog at en gruppe på 6 er for stor. Det meste af semesterprojektet har det ikke være muligt med fysisk fremmøde, hvilket har været en udfordring, og det har især gjort opstarten en smule mere besværlig. Online semesterprojektarbejde er vi dog med tiden blevet bedre til, og det blev med tiden en mindre udfordring. Generelt har information vedrørende semesterprojektet være forvirrende og manglet struktur, hvilket til tider har skabt forvirring, hvilket ikke har været optimalt. Denne teknik til læring hvor vi skal udføre et stykke arbejde først, og vi derefter lærer, hvordan vi skulle have gjort, det er hårdt men effektivt. Vi bliver tvunget til at prøve os frem, og vi kommer ud i nogle situationer hvor vi mislykkedes, men der er noget, jeg har lært meget af. Jeg synes det er en smule ærgerlig, at vi ikke har muligheden for at komme foran, da der var sat nogle faste dage, hvor man først fra disse dage fik information om næste fase. Derfor var det ikke muligt at komme foran, da der ikke var adgang til næste del af processen. I forhold til første semesters projekt, har dette projekt dog været en del mere overskueligt, da man har haft en grundlæggende forståelse for processen. Generelt synes jeg at det har været et godt semester projekt, hvor jeg personligt har lært en masse, og fået en bedre forståelse for en generelt softwareudviklings proces. Vores proces har fungeret nogenlunde, men en stor gruppe, online arbejde og en epidemi har været en udfordring.

8.2.3. Stefans vurdering

Projektarbejdet dette semester havde helt klart nogle positive ændringer i forhold til det tidligere semester, dog er der stadigvæk områder som jeg mener kan forbedres. Jeg er glad for den store forbedring gruppedannelsen har fået i forhold til sidste semester, og jeg synes at vi har fået en overraskende god gruppe. Jeg vil også vurdere at jeg selv lagde meget arbejde i at finde folk og komme i en gruppe, som jeg kunne arbejde godt sammen med.

I forhold til gruppearbejde synes jeg selv at jeg fik et stort udbytte af at kunne arbejde i mindre "arbejdsgrupper" på 2 eller 3. Det at kunne have et ekstra sæt øjne til at fange fejl eller komme

TV 2 Krediteringsprogram

med input, som man måske ikke selv havde tænkt over, synes jeg gav at man ofte fik lavet et bedre stykke arbejde i første omgang.

Projektarbejde under hele det her Corona har også været lidt af en blandet landhandel. Det har helt klart været problematisk at man ikke rigtigt har kunne mødes irl, især fordi jeg tror at det dæmper på gruppens sociale aspekt, men samtidig så har det givet erfaring med at arbejde sammen med folk, og udvikle software, over nettet.

Jeg er også ret tilfreds over det formelle Review af inceptionsdokumentet og midtvejsseminaret. Jeg synes ikke at man får nok brugbar kritik ud af det, og jeg er ret skuffet over at vi ikke i stedet har nogle mere dybdegående kritikker af enkelte gruppers arbejde. For trods at de formelle reviews er et trin op i forhold til posters, så kunne det stadigvæk godt forbedres.

8.2.4. Emils vurdering

Dette semesterprojekt var en del anderledes end tidligere forløb, men dog ikke foruden ny viden. Covid var en markant faktor for, hvordan samarbejdet og kommunikationen i gruppen forløb sig, og gjorde det på mange måder svært at koordinerer fysisk eller digitalt fremmøde. Det påvirkede også undervisningen, men synes ikke forelæsningsen online var en ulempe, men oftere en fordel, da man kan optage dem til senere brug. Måden hvorpå undervisningen var struktureret sammen med dette semesters projekt, fungerede ikke særligt godt mener jeg. Der gik for lang tid mellem at vi fik lært stoffet(teori), til det skulle bruges i projektet, og gjorde det vanskeligt at gå fra teori til praksis.

Generelt føler jeg mine kompetencer indenfor hvordan man strukturerer processen for et projekt, samt udvikling af software, er blevet markant bedre. Dette er både fordi at jeg har haft nogle markant bedre gruppemedlemmer med langt større faglige kunnen end sidste semester, men også fordi at man bliver mere rutineret indenfor selve softwareingeniørdisciplinerne, med øvelser som journaler og Reviews.

Samarbejdet i gruppen synes jeg har været tilfredsstillende. Det kan mærkes der er markant forskel på hvor langt vi individuelt er, i forhold til fx kodning. Dette mener jeg er en fordel, da man oftest altid har en "ekspert" at gå til, når man engang støder ind i problemer, plus den kode man står tilbage med giver bedre forståelse for 'best practices'.

8.2.5. Patricks vurdering

Dette semester, har været meget anderledes end det forrige, da Corona for alvor ramte og påvirkede hele vores semester. Forvirringen var stor, da vi skulle danne semesterprojekts grupper. Det sociale har ikke været i fokus grundet samfundet situation, så vi har haft svært ved at komme ind på hinanden, men på trods af det fik vi lavet en god gruppe. Måden undervisningen har været struktureret i forhold til projektet har været lidt rodet. Det gjorde det svært i projekt arbejdet, at der var en følelse af vi manglede de ting vi skulle bruge til projektet fra undervisnings siden.

Generelt har jeg følt at projektet er gået godt, trods at man har kunne mærke situationen har påvirket alle, det har været svære at holde sig motiveret, men vi har arbejdet igennem det hele på

TV 2 Krediteringsprogram

en god måde. Der har været god faglig sparring i gruppen, selvom man godt har kunne mærke niveauforskelle på nogle punkter.

Vejledningen har været god, der har ikke været mange punkter, hvor vi har følt, at vi ikke fik den rette vejledning fra Rune. Gruppektrakten er blevet overholdt, der har været god kommunikation i gruppen, som har været vigtigt i projektet. Det har været et godt semester projekt alt i alt, selvom der har været nogle strukturproblemer på det her semester.

8.2.6. Johans vurdering

Semesterprojektet i år har været en hel del anderledes. Både med at det hele foregik online, men også min arbejdsindsats. Forvirringen har været stor og det startede allerede ved gruppedannelsen. Vi kendte knap nok hinanden, så det var svært at få dannet grupper, men jeg mener selv at vi fik dannet en god gruppe. At det hele lå online (både semesterprojektet og undervisningen) har gjort det supersvært for mig. Der var en hel del distraktioner der var i spil og selvom jeg prøvede at fokusere, var det svært. Det har haft en stor betydning for min arbejdsindsats i selve projektet. Ting som de andre havde fuldstændig styr på, havde jeg for eksempel meget svært med. Det ødelagde min motivation, men jeg synes jeg er kommet godt med her til sidst. Informationer vi fik at vide om selve projektet har været forvirrende. Først får vi at vide det ene, så siger vejlederen noget andet og står der noget tredje i et dokument der er nærmest gemt væk i LitsLearning. Strukturen i undervisningen i forhold til arbejdet der blev lavet, var lidt mærkelig. Vi var ellers foran, men vi kunne ikke arbejde os mere foran, da undervisningen fortalte hvad vi skulle.

Ellers har vejledningen været god. Der var en nærmest uformel tilgang til vejledningen og det synes jeg var godt og de gav et andet billede på tingene end hvad man lærte til undervisningen.

9. Referenceliste

1. TV 2, besøgt 25/5/2021, TV 2 kanalbrand design guide, <https://TV2marketing.dk/design/tv-2-brand/>
2. A case made by a collaboration between TV 2 and SDU, Credits Management A case by TV 2 Denmark A/S,
<https://docs.google.com/document/d/1p6lQjWV76TX9uTLst2OAdmV07XfMRn5fObelzBpd2EI/edit#heading=h.lzb4q7rymmbk>

3. Retsinformation - Ophavsret

Udgiver:	Civilstyrelsen
URL:	https://www.retsinformation.dk/eli/lta/2014/1144
Set:	27/05/2021

Bilag

A. Kildekode

GitHub: https://github.com/Skashe/C-10_SDU_SEM2_SE.git

TV 2 Krediteringsprogram

B. Brugervejledning

Opsætning af programmet

Programmet afleveres med kildekode, en JAR-fil, en EXE-fil og en SQL-fil. Alle filer kan også hentes på github:

[Release 2. Iteration · Skashe/C-10_SDU_SEM2_SE \(github.com\)](https://github.com/Skashe/C-10_SDU_SEM2_SE)

både JAR og EXE filerne kræver JDK-15 installeret (JDK-16 anbefalet). Dette kan hentes på:

<https://www.oracle.com/java/technologies/javase-jdk16-downloads.html>

Programmet forbindes til en online database og kræver derfor internet for at kører. Tjek derfor dit internet, hvis programmet ikke starter. Starter programmet stadig ikke, kan det køres i kommandolinjen, for yderligere informationer.

Opsæt med egen database

Hvis programmet skal køres lokalt uden brug af den online database, så er det påkrævet at du opsætter en lokal PostgreSQL database.

Når du har oprettet en lokal database, skal du gøre 2 ting:

1) Du skal gå ind og ændre værdierne i DatabaseSystemValues klassen så værdierne passer til den lokale database.

Eksempel på klassen med værdierne fra en lokal postgres database.

```
package PersistenceLayer;

public class DatabaseSystemValues {
    public static final String url = "localhost";
    public static int port = 5432;
    public static String databaseName = "MyDatabase"; //insert database name
    public static String username = "UserName";
    public static String password = "Password";
}
```

2) Du skal køre den afleverede SQL-fil i din oprettede database.

Hvis du har udført de overstående trin korrekt, bør du nu kunne køre programmet, hvor den bruger din lokale PostgreSQL database.

Funktioner i programmet

Programmets forside indeholder et søgefelt og en liste med shows. Ser du ingen shows, og har du selv sat databasen op, er det muligt at du ikke har initialiseret dataene. Tjek derfor "opsætning af egen database" igennem igen.

Programmet har en menu i venstre side. Her finder man de fleste funktionaliteter. På forsiden er en login-knap. Her kan man logge ind som 2 forskellige brugertyper. Dette er Administrator med

TV 2 Krediteringsprogram

brugernavn "admin" og password "admin", og producent med brugernavn "producer" og password "producer".

De fleste sider er bygget op omkring lister med indhold man kan trykke på. Nogle af liste elementerne har forskellig funktionalitet, alt efter hvor man trykker på dem. Kig derfor efter ikoner placeret på listeelementerne.

Man kan altid navigere tilbage med menu-elementet nederst i menuen. Er du i tvivl om et menu-elements funktion, kan menuen udvides ved at klikke i toppen af menuen. Herved ses en kort tekst til hvert menu-element.

TV 2 Krediteringsprogram

C. Samarbejdsaftale

Samarbejdsaftale

Gruppe C-10

5. februar 2021

Gruppemedlemmer

	IM	KS	KO	OP	AN	FM	OR	AF	SP
Troels	62	36	35	82	49	10	35	76	57
Emil	57	70	45	51	42	70	25	30	60
Johan	61	59	30	51	32	56	8	19	46
Stefan	0	83	40	42	73	79	7	41	62
Daniel	56	44	60	46	58	32	53	54	36
Patrick	25	45	48	62	38	30	58	57	35

- Troels Kaldau
- Emil Madsen
- Johan Brockstedt
- Daniel Smidstrup
- Stefan Profft Larsen
- Patrick Holmquist Andreassen

Forventninger til projektet

- Vi får anvendt de projektprocesser som vi benyttede i sidste projekt.
- Vi får udvidet projektprocesserne.
- Vi får anvendt viden fra alle 2. semesters fag.
- Vi får anvendt en effektiv og sikker databasestruktur.
- Vi får lavet gode APIs til produktet der nemt kan implementeres af andre.
- Vi får udarbejdet en professionel rapport, sideløbende med projektet.

Forventninger til hinanden

- Alle yder deres bedste.
- Alle er åbne overfor spørgsmål om hjælp.
- Alle er åbne overfor kritik.
- Alle yder konstruktiv kritik til andres arbejde.
- Alle deltager aktivt i diskussioner om projektets udvikling og udfoldning.
- Alle holder øje med arbejdstider og deadlines, og overholder dem.
- Det skrives ASAP hvis man ikke kan overholde tider eller deadlines.
- Alle følger aktivt med i alle andre fag for at få relevant faglig viden til projektet.
- Alle følger aktivt med i projektudviklingen og spørger ind til ting de ikke forstår.
- Alle finder selvstændigt arbejdsopgaver de kan arbejde med i arbejdstiden.
- Alle deltager i arbejdsfordelingen og koordinerer arbejdsindsatsen med de andre.
- Alle er detaljeorienterede og bekymrer sig om deres arbejde.
- Man læser korrektur på sit eget materiale.

Arbejdstider og kommunikation

Alle tider er vejledende.

- Tirsdag 10:00 – 16:00
- Fredag 9:00 – 12:00
- Søndag 20:00 - 20:30

Værktøjer

- | | |
|---------------------|---------------------|
| • Kommunikation: | Discord & Messenger |
| • Kodedeling: | Github |
| • Dokumentdeling: | Teams |
| • Rapportskrivning: | Latex |

Gruppemøde og Vejledermøder

TV 2 Krediteringsprogram

- Vi stræber efter at holde faste gruppemøder tirsdag 10 – 16, fredag 9 – 12 og søndag fra kl. 20 samt at have Vejledermøde som en del af tirsdagsmødet. Yderligere møder kan blive aftalt løbende.
- Hvis et gruppemedlem eller vejleder ikke har mulighed for at møde op til et møde eller er forsinket SKAL vedrørende give besked til resten af gruppen hurtigst muligt (evt. et specifikt sted så som Discord eller Messenger). Hvis et gruppemedlem konstant formår at udeblive, bør der tages en seriøs samtale med vedkommende internt i gruppen, eventuelt med tilstedeværende vejleder.

Konflikthåndtering

- Hvert gruppemedlem har eget ansvar for at få vendt problemer med gruppen
- Hvis man har problemer med en anden i gruppen, holder man sproget i en sober tone.
- Man skal ytre utilfredshederne hurtigst muligt, så det ikke når at blive et problem. Tag det op med den enkelte i første omgang, og hvis det ikke bliver løst, så vend det i gruppen.

Logbog

Link til logbog:

https://github.com/Skashe/C-10_SDU_SEM2_SE/wiki

Opgaver:

- Vejledermøde dagsorden
- Problemer/Spørgsmål
- Referat af vejledermøde
- Opnået
- Målsætning til næste gang

Rollefordelingsliste:

- Troels
- Johan
- Daniel

TV 2 Krediteringsprogram

- Patrick
- Stefan
- Emil

Groft Deadlines

- **Week 5:** gruppekontrakt, projektforslag
- **Week 6:** præsentation af projektforslag
- **Week 7:** problemanalyse, “inception document”
- **Week 9:** møde med TV 2
- **Week 11:** præsentation af poster og “inception document”
- **Week 12:** 1. iteration – krav, analyse, design, implementation, test
- **Week 14:** møde med TV 2
- **Week 16:** projektseminar
- **Week 17:** 2. iteration – krav, analyse, design, impl., test, database
- **Week 19:** Møde om projektaflevering og eksamen
- **Week 20:** Afslutning
- **Week 22-27:** Eksamen

Kontaktinformation

- Troels Kaldau
 - Email: trkal20@student.sdu.dk
 - Mobil: 42803043
- Emil Madsen
 - Mail: emima20@student.sdu.dk
 - Mobil: 61 31 03 06
- Johan Brockstedt
 - Mail: jobro19@student.sdu.dk
 - Mobil: 31 93 84 93

TV 2 Krediteringsprogram

- Daniel Smidstrup
 - Mail dasmi20@student.sdu.dk
 - Mobil 21523907
- Stefan Profft Larsen
 - Mail: stefa20@student.sdu.dk
 - Mobil: 61855864
- Patrick Holmquist Andreassen
 - Mail: pandr20@student.sdu.dk
 - Mobil 50474811

Underskrifter

TV 2 Krediteringsprogram

D. Projektlog

https://github.com/Skashe/C-10_SDU_SEM2_SE/wiki

TV 2 Krediteringsprogram

E. Rapportkontrolskema

Rapportkontrolskema

Skemaet udfyldes ved i kolonnen "Opfyldt" at markere de krav, der menes at være opfyldt med et + og de uopfyldte med -. Skemaet indgår i rapporten som Bilag F.

Kapitel	Krav	Opfyldt +/-
Forside	Projekttitel, uddannelsesinstitution, fakultet, institut, uddannelse, semester, kursuskode, projektperiode, vejleder, projektgruppe og projektdeltagere (fornavn, efternavn, sdu-email). Må gerne have illustrationer.	+
Titelblad	Samme oplysninger som på forsiden, samt afleveringsdato og projektdeltagernes underskrifter (Projektdeltagernes aktive deltagelse i projektføreløbet anerkendes gensidigt ved projektdeltagernes underskrifter). Må ikke have illustrationer.	
Resumé	• En kort introduktion til projektet - hvad blev der arbejdet med og hvorfor. • Problemformuleringen og vigtige afgrænsninger • Metode - hvordan angreb I problemet og hvordan realiserede I løsningen (hvem, hvad, hvornår og hvorfor)	+

TV 2 Krediteringsprogram

	• Hovedresultater og konklusioner – hvad kom der ud af arbejdet. (max 1 side)					
Forord	Hensigten med rapporten, målgruppe, forhistorie, anerkendelser.	+				
Indholdsfortegnelse	Samlet indholdsfortegnelse for hele projektrapporten. Højst to eller tre niveauer i indholdsfortegnelse (der kan evt. være flere i selve rapporten). Afsnit på niveau 1 og 2 skal være nummererede.	+				
Læsevejledning	Vejledning i hvordan rapporten kan læses, eksempelvis i form af hvilken rækkefølge afsnittene kan læses i, og hvordan sammenhængen er mellem de forskellige dele af rapporten, herunder mellem hovedrapport og bilag. Rapportens målgruppe.	+				
Redaktionelt	Skriveprocessen og ansvarsområder i skriveprocessen. Ansvarsområder kan fx beskrives på fx følgende form: <table><tr><td>Afsnit</td><td>Ansv arlig</td><td>Bidr ag fra</td><td>Kont rolle ret af</td></tr></table>	Afsnit	Ansv arlig	Bidr ag fra	Kont rolle ret af	+
Afsnit	Ansv arlig	Bidr ag fra	Kont rolle ret af			

TV 2 Krediteringsprogram

	Afsnit a	Person a	Person b	Person a, b, c
	Afsnit b	Person b	Person a	Person a, b, c
	Afsnit c	Person c	Person b	Person a, b, c
Indledning	+			

Faglige vidensgrundlag	Begrebsdefinitioner, teori og fagligt vidensgrundlag	+
Metode og planlægning	Metode Benyttede metoder i projekt (hele projektet). Kombination af UP og Scrum. Fordele og ulemper	+
	Planlægning	+

TV 2 Krediteringsprogram

	Plan for elaborationsfasen og de enkelte iterationer. Prioritering af krav i planlægningen. Det faktiske udviklingsarbejde. Faserne, iterationerne og det faktiske arbejde i dem? Scrum: backlogs, roller, begivenheder, scrum-buts.	
Hovedtekst (skal indeholde resultater både fra iteration #1 og fra iteration #2)	Overordnede krav Opdateret resume af overordnede krav fra inceptionsdokument inklusive overordnet brugsmønsterdiagram og oversigt over supplerende krav	+
	Detaljerede krav Detaljeret brugsmønsterdiagram (hvis relevant). Detaljerede brugsmønsterbeskrivelser. Detaljerede beskrivelser af supplerende krav, fx organiseret efter FURPS+.	+
	Analyse Overvejelser, beslutninger og resultater vedr. analysemodellen, inklusive både den statiske og den dynamiske side af analysemodel.	+
	Design	+

TV 2 Krediteringsprogram

	Overvejelser, beslutninger og resultater vedr. Softwarearkitektur og detaljeret design, herunder design af persistens.	
	Databasedesign Overvejelser, beslutninger og resultater vedr. tabeldesign og SQL-forespørgsler	+
	Implementering Overvejelser, beslutninger og resultater vedr. konvertering fra design til kode illustreret gennem udvalgte centrale eksempler, samt andre vigtige implementeringsbeslutninger. Implementering af database.	+
	Test udførte test samt resultatet af dem.	+
Diskussion	Hvad er der opnået og hvad er der ikke opnået i projektet i forhold til det forventede som beskrevet i indledningen. Hvad er styrkerne og svaghederne ved resultaterne. Kunne I have opnået bedre resultater?	+
Konklusion	Opsummering af resultaterne og diskussionen af dem. Svar på problemformuleringen.	+

TV 2 Krediteringsprogram

Perspektivering	Er den fundne løsning brugbar i anden sammenhæng? Hvad bidrager løsningen og den opnåede viden til. Fremtidigt arbejde (næste skridt i projektet, hvis I havde mere tid).	+
Procesevaluering	Processen og gruppens refleksion over processen: Læringsprocessen, teamroller, samarbejdet internt i gruppen og med vejleder, projektarbejdsformen, arbejdsformer, metoder, skriveprocessen, den tidsmæssige styring af projektet, ledelse af projektet, arbejdsfordeling i projektet m.m. Hvordan ville I gribe arbejdet an, hvis I skulle starte forfra?	+
Referenceliste	Litteratur angivet på en anerkendt form. (Alle former for litteratur som bøger, artikler og hjemmesider) Kildehenvisninger i teksten. Materiale som gruppen ikke selv har fremstillet i dette projekt skal være angivet med kilde! Alle kildehenvisninger i teksten skal være anført på samme måde. Kildeangivelser på figurer, grafer etc. som	+

TV 2 Krediteringsprogram

	projektgruppen ikke selv har frembragt.	
--	---	--

Rapporttekniske elementer		Opfyldt +/-
Layout	Er der anvendt samme layout i alle kapitler. Er layout overskueligt/harmonisk.	+
Sprog	Formidler rapporten projektet faglig og sagligt. Er sproget neutralt, aktivt, upersonligt, konkret, præcist, kortfattet og korrekt. (Procesevalueringen må benytte personligt sprog)	+
Sidenummerering	Er der korrekt og konsistent sidenummerering i rapporten.	+
Figurer/diagrammer	Er alle figurer konsekvent nummererede. Er der figurtitel og figurtekst til alle figurer. Er figurtitler og figurtekster dækkende og afklarende. Er figurerne tydelige og læsbare. Er figurerne informationsgivende og i den rette sammenhæng.	+
Tabeller	Er alle tabeller konsekvent nummererede. Er der en forklarende tabeltekst til alle tabeller.	+

TV 2 Krediteringsprogram

	Er alle søjler og rækker forsynet med parametre. Er der enheder på alle relevante rækker og søjler.	
Sporbarhed af begreber	Er der en konsekvent brug af samme betegnelse for et givet begreb igennem rapporten.	+

TV 2 Krediteringsprogram

F. Scrum-Buts**We use Scrum, but:**

- having a Daily Scrum every day is too much overhead, so we only have one per workday.

At have en daglig Scrum er overkill, da vi kun mødes én gang om ugen. Vi vil i stedet for have et Scrum møde i starten af hvert møde, hvor vi lige får samlet op.

- Retrospectives and sprint review are combined

Retrospectives og review er slået sammen, da vi ikke forventer at have det store der skal godkendes, ved review, men vi vil stadig gennemgå helt projektet til slut for at sikre, at alt fungerer som det skal. Der forventes at lave en gennemgang af sprinten, retrospectives, hvor vi vil gennemgå hvad vi skal start doing, stop doing og continue doing. Efter 1.iteration vil dette møde benyttes til at forbedre SCRUM processen i 2. iteration.

- No product owner(s) are chosen, the whole group is responsible for the product backlog.

En product owner har ansvaret for at prioritere arbejdsopgaverne. Dette er noget vi indtil nu har gjort i fællesskab. Det kan skabe gnidninger hvis folk er uenige om hvad der er vigtigst, men det kan også formindske arbejdsglæden hvis man bliver bedt om at arbejde på noget, som man selv ikke mener er relevant på det givne tidspunkt. Vi fortsætter derfor som vi har gjort, hvor alle gruppemedlemmer agere som product owner. Opgave prioritering vil blive gjort ved Spring planning i fællesskab, og i et mindre omfang undervejs i sprinten. Hvis dette ikke fungerer, i.e., det tager for lang tid om at blive enige om prioriteringen, vil vi genoverveje at tildele product owner rollen til et enkelt medlem.

TV 2 Krediteringsprogram

G. Scrum retrospective og sprint review for anden sprint

Hvad vi satte os for

Backlog for anden iteration:

1. SQL-database
2. Link krediteringer
3. Login med rigtige brugere
4. Opret brugere
5. Log ændringer i database
6. Krediterings validering
7. Eksporter krediteringsfil
8. Hvis liste med loggede ændringer
9. Implementer flow-tv

Hvad vi fik lavet

- SQL-database
- Link krediteringer
- Login med rigtige brugere
- Opret brugere
- Log ændringer i database
- Eksporter krediteringsfil
- Hvis liste med loggede ændringer

Hvordan det gik

Hvad gik galt:

- Der blev ikke lavet nok design.
- Arkitekturen blev ofret for hurtigere implementering.
- Der manglede struktur og planlægning af arbejdet.

Hvad gik godt:

- Der blev investeret mere tid i arbejdet.
- Der var mere kommunikation.
- Der var en konkret ansvarsfordeling

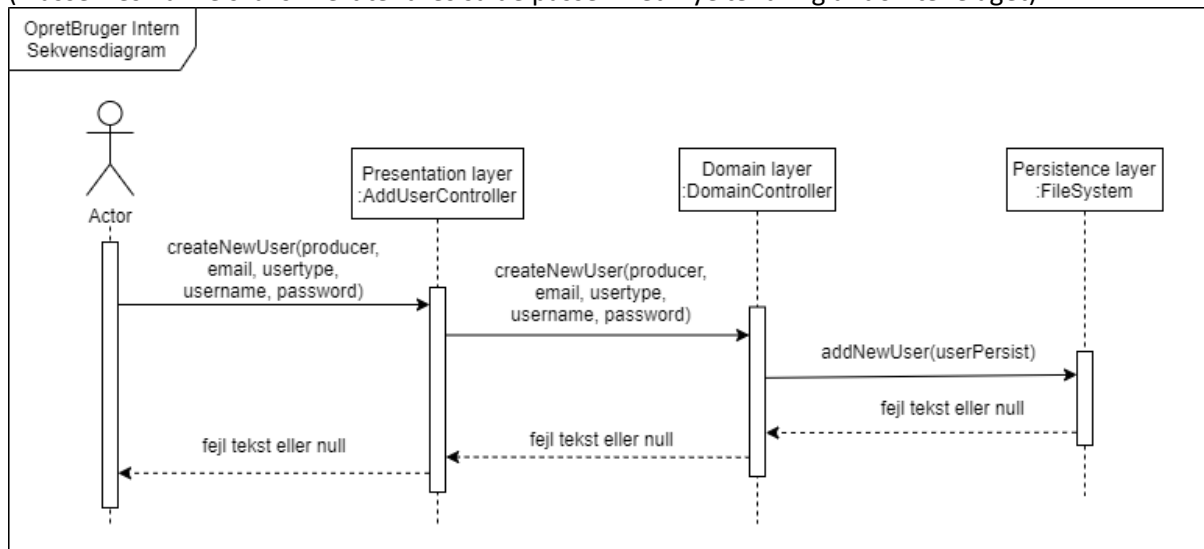
Fokus til næste sprint:

- De områder der gik galt

TV 2 Krediteringsprogram**H. Anden iteration design****Opret brugere****Admins oprettelse af nye admins eller producere.**

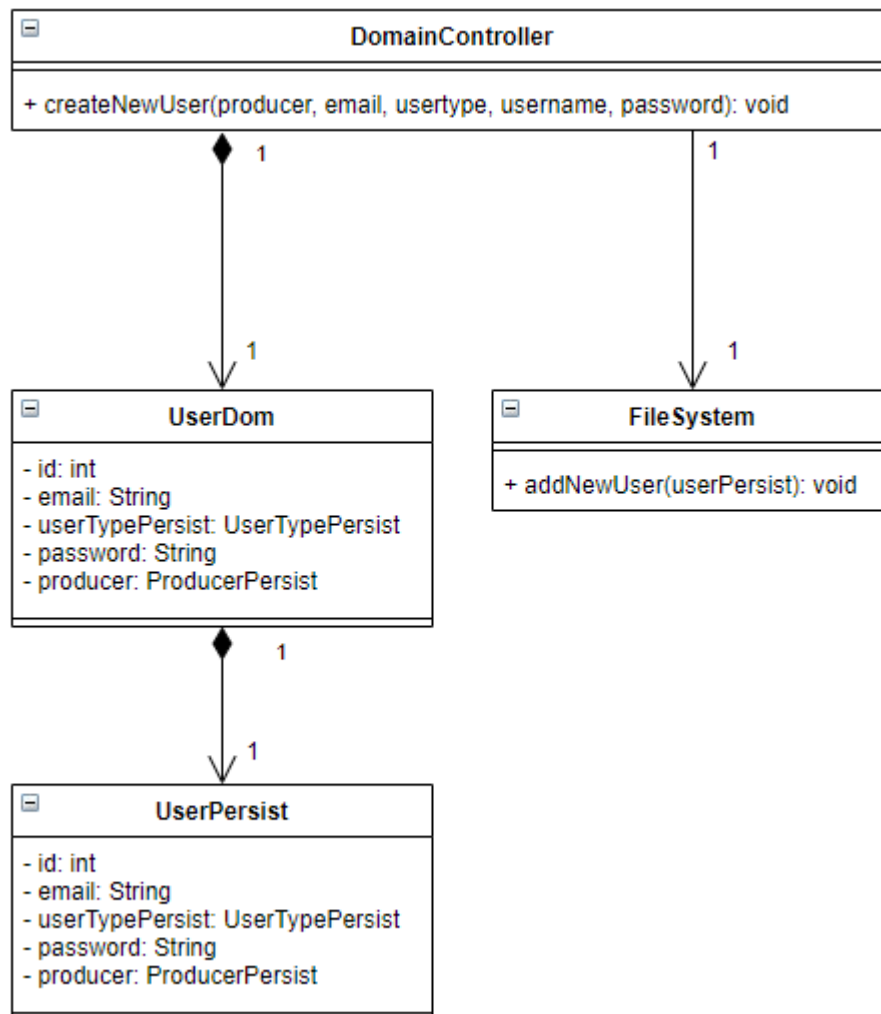
Første udgave af design

(Klassernes navne skal sikkert ændres så de passer med nye ændring af domænelaget)



Når en admin ønsker at oprette en ny bruger, startes en sekvens hvor, man tilføjer information om denne bruger, der vælges om det er en admin eller en ny producer, hvis det er en ny producer, angives producerens navn.

TV 2 Krediteringsprogram



TV 2 Krediteringsprogram

I. UI-design

UI-designing for første iteration

Offentligt overblik

The wireframe shows a public overview interface. It features a header area with a lock icon in the top right corner. Below the header is a table with four columns. Under the table is a search bar with a magnifying glass icon to its right. Below the search bar is a list of seven rows, each representing a search result. The entire interface is enclosed in a rectangular frame.

UI'et består af 4 elementer: flow-tv-overview, searchbar, search-result-list og login-button. Flow-tv-overview viser en liste af krediterede programmer der kan scrolles igennem af brugeren, men ikke ændres på. Searchbar kan anvendes til at søge efter krediterede programmer og search-result-list viser en liste med de krediterede programmer der matcher søgeordene. Login-button

Offentligt krediterings overblik

TV 2 Krediteringsprogram

Udsendelsenavn
Producent



UI'et består af 3 elementer: title-textfields til udsendelsesnavn og producent, listview af krediterede personer og download-knap. Download-knappen downloader en fil med krediteringerne.

[Valideret bruger overblik](#)



UI'et består af 2 elementer: en searchbar og en search-result-list. Brugeren kan søge efter specifikke udsendelser med searchbaren. Search-result-list viser krediterede programmer der matcher søgeordene.

TV 2 Krediteringsprogram

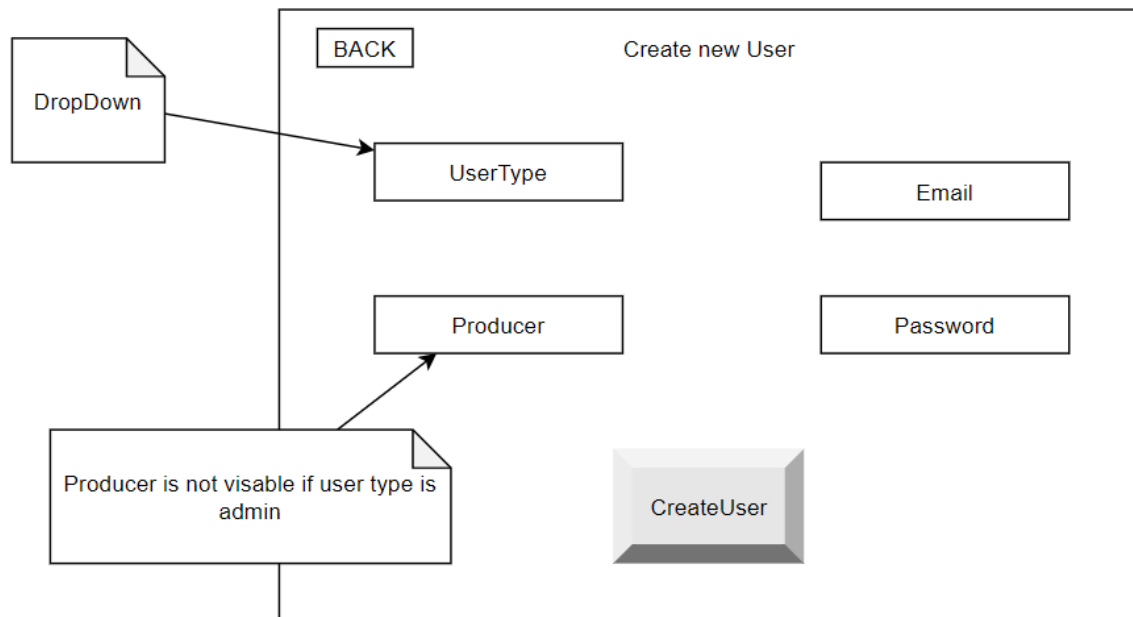
Krediterings ændrings overblik

The UI design for the 'Krediterings ændrings overblik' (Crediting Change Overview) screen is contained within a rectangular frame. At the top, there are two text input fields: 'Udsendelsenavn' (Broadcast Name) and 'Producent' (Producer). To the right of these fields is a blue square icon with a white document symbol. Below the input fields is a listview containing ten rows. Each row has a light gray background and a small blue circle with a white plus sign at the end. The listview is followed by a large, empty rectangular box, likely for a download button or additional information.

UI'et består af 4 elementer: title-textfields til udsendelsesnavn og producent, listview af krediterede personer add-knap og download-knap. Teksten i textfields'ne kan redigeres ved at trykke på krediteringer i listviewet kan krediteringer redigeres. Download-knappen downloader en fil med krediteringerne.

UI-desing for 2. iteration

TV 2 Krediteringsprogram



Når en Admin ønsker at oprette en ny bruger benyttes dette nye vindue, hvor man har tekstfelterne Username, Email og Passowrd. Man vælger derefter en User Type, som er en drop down menu, hvis man ønsker at oprette en ny producer, bliver man også bedt om at angive et producernavn.

TV 2 Krediteringsprogram

J. Analyse og design af "Tilføj kreditering" og "ændre kreditering"

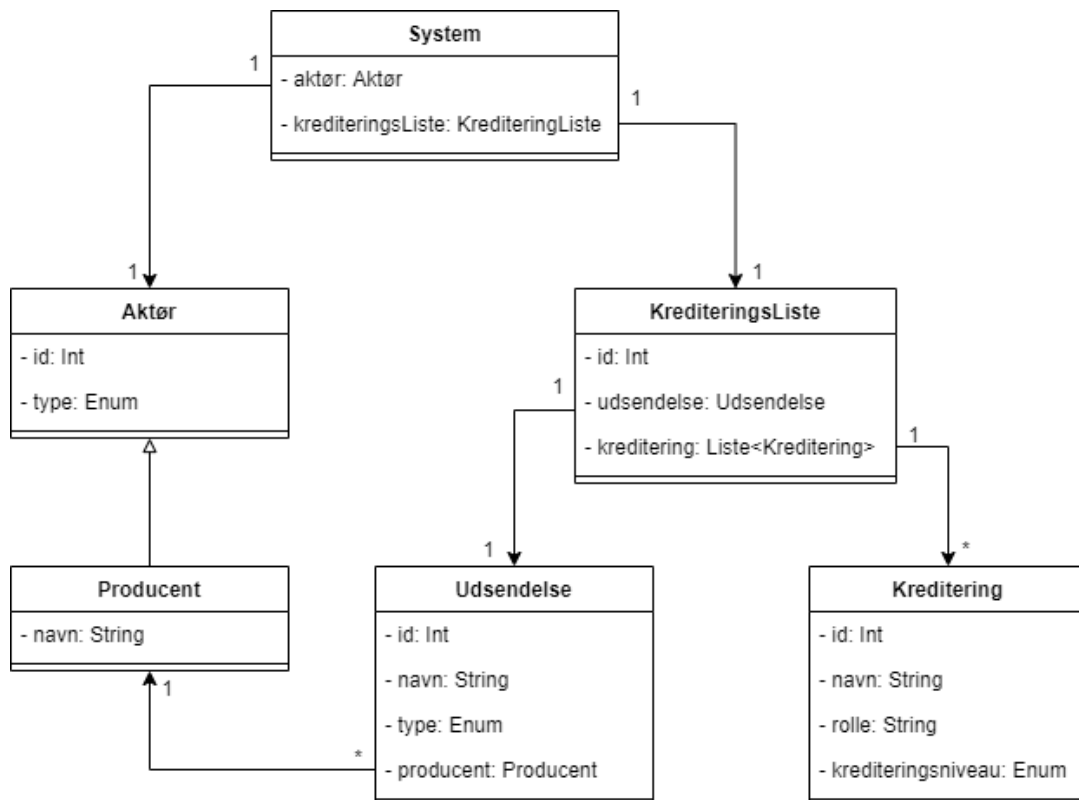
Tilføj krediteringer

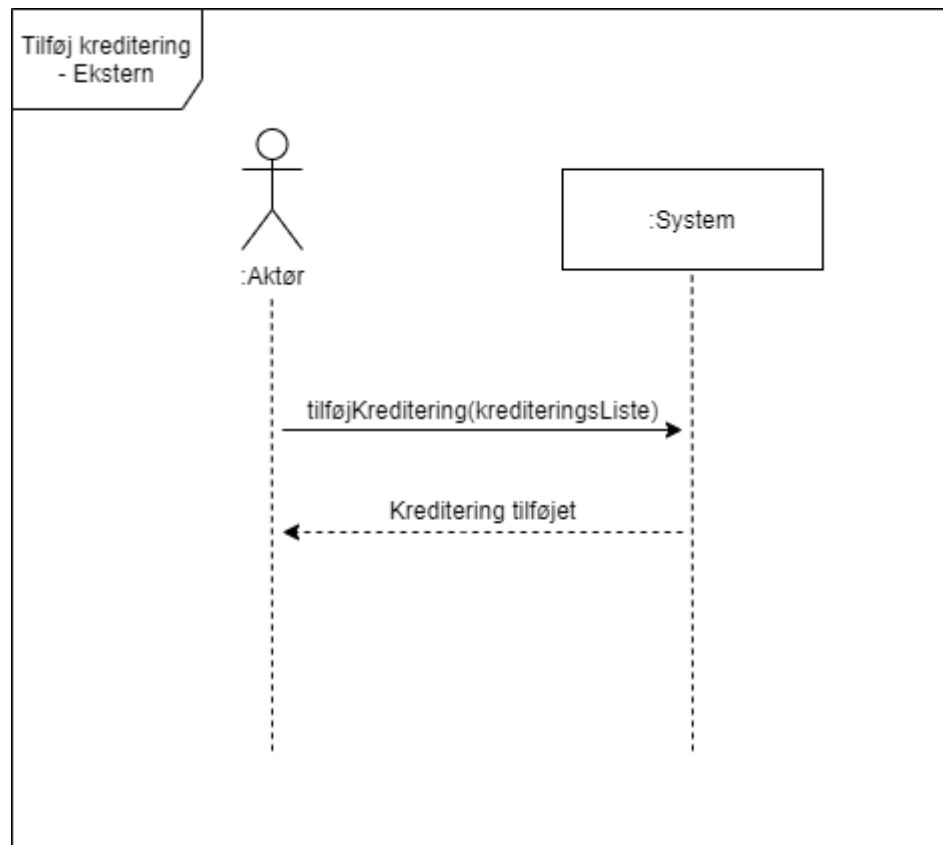
Tilføj kreditering	
ID	#4
Formål:	At tilføje krediteringer til systemet
Overblik	Producenten indskrives informationer om produktet der krediteres , og tilføjer derefter kreditering af individer.
Aktør	Producent
Stakeholders	Producent, der ønsker at kreditere medarbejdere. Medarbejdere, der ønsker at blive krediteret ordentligt. TV 2, der ønsker at vise og videregive de rigtige krediteringer.
Forudsætning	Aktør er logget in som producent eller administrator.
Postconditions	Kreditering af udsendelse er tilføjet til systemet
Sekvens	1. Aktør initierer tilføjelse af kreditering. 2. Aktør tilføjer informationer om udsendelsen der krediteres for. 3. Aktør tilføjer individer der ønskes krediteret med navn og funktion. 4. Aktør gennemser krediteringslisten og accepterer den. 5. Krediteringslisten tilføjes til systemet.
Alternative scenarier	a. Aktør afslutter proceduren før krediteringen er tilføjet til systemet.

Klassekandidater	Attributter	Funktionskandidater
Kreditering System Producent Produkt Individ Administrator Udsendelse Aktør	Navn Funktion Information Kreditering	Tilføj (Kreditering) Indskriv/tilføj (Information) Tilføj (Udsendelse) Tilføj (Individ) Gennemse Krediter Log in Acceptor Afslut Tilføj (til system)

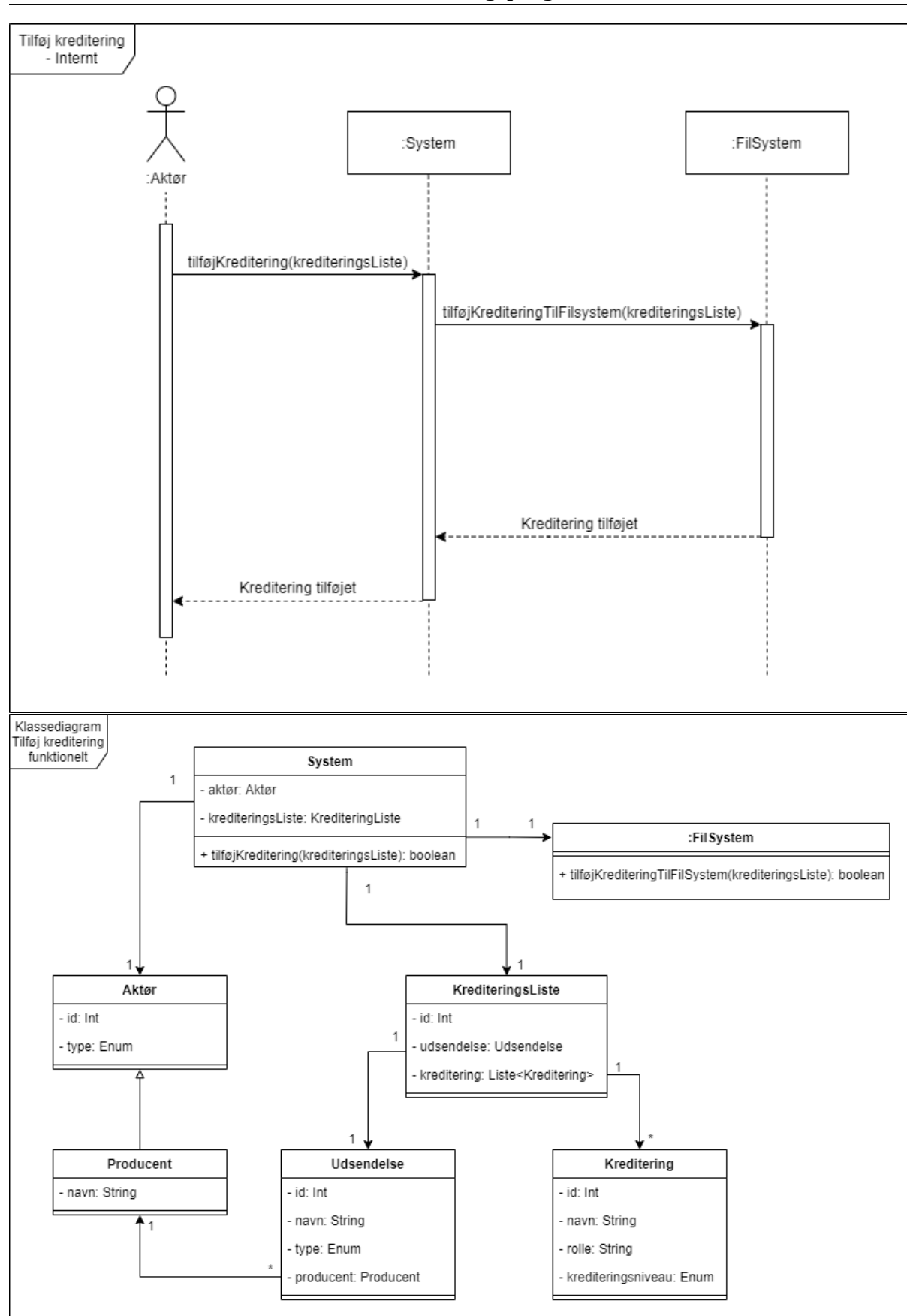
TV 2 Krediteringsprogram

Klassenavn	Ansvar	Samarbejdspartnere
Kreditering	Hold information om de krediterede, om projektet og producenten.	System, Individ, udsendelse
System	Kan oprette og tilføje af kreditering	Kreditering, Aktør
Aktør	Holder informationer om den autoriserede aktør	System
Individ	Holder information om et individ til brug i kreditering	Kreditering
Udsendelse	Holder information om en udsendelse til brug i kreditering	Kreditering



TV 2 Krediteringsprogram

TV 2 Krediteringsprogram



Ändre krediteringer

TV 2 Krediteringsprogram

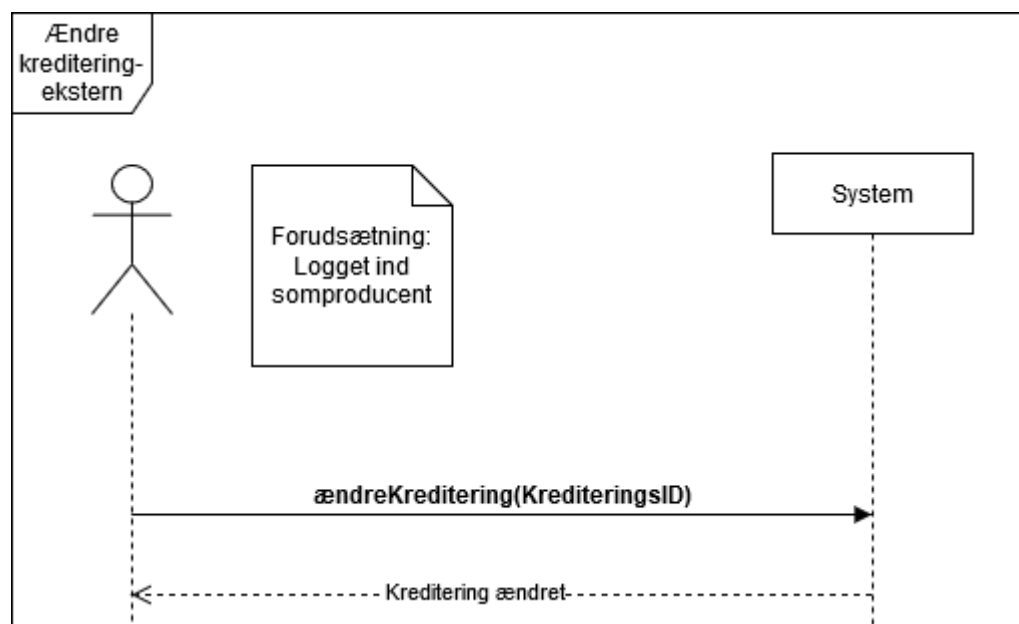
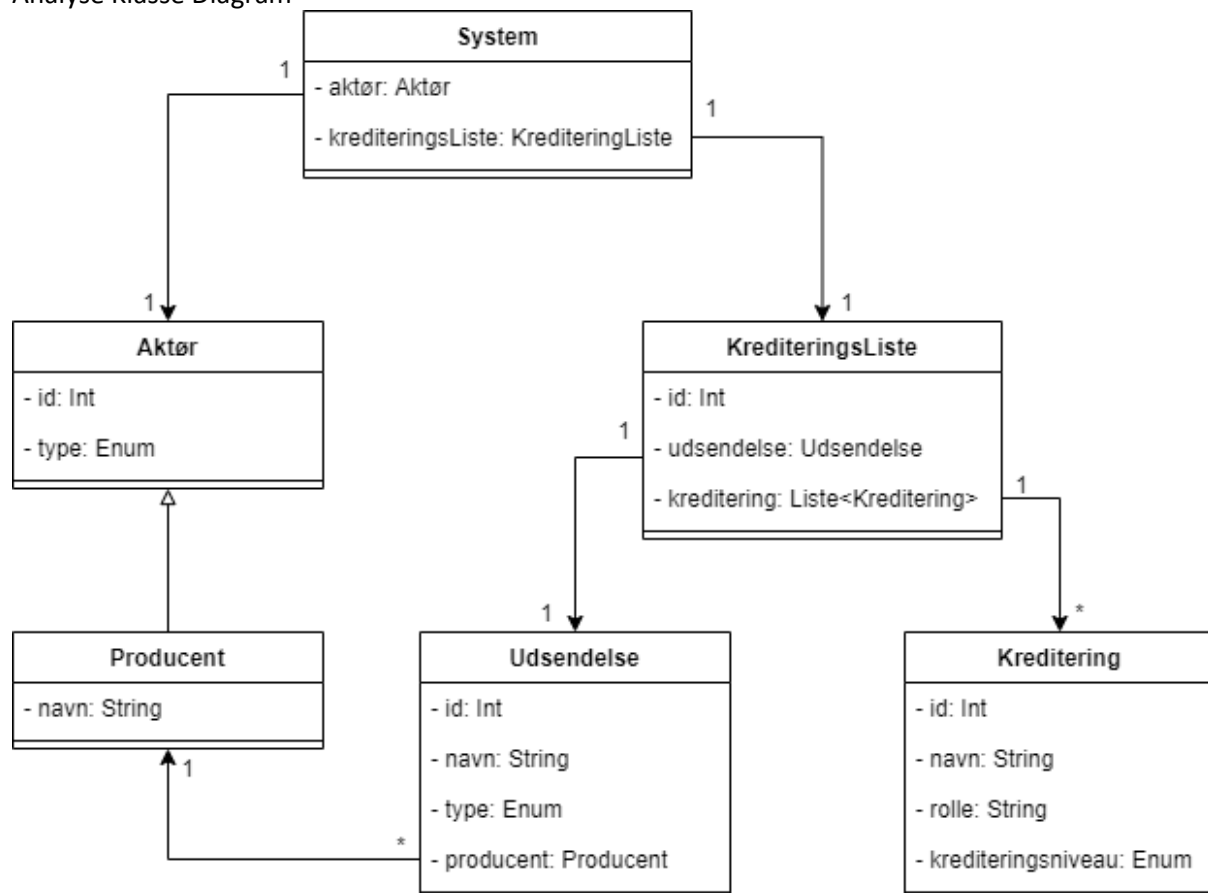
Ændre kreditering	
ID	#3
Formål:	At ændre krediteringer i systemet
Overblik	Producent eller administrator udvælger krediteringsliste de ønsker at ændre , og tilføjer herefter ændringer .
Aktør	Producent, administrator
Stakeholders	Producent , der ønsker at kreditere medarbejdere . Medarbejdere , der ønsker at blive krediteret ordentligt. TV 2 , der ønsker at vise og videregive de rigtige krediteringer .
Forudsætning	Aktør har fundet krediteringen der ønskes ændret . Aktør er logget ind som producent eller administrator .
Postconditions	Kreditering af udsendelse er ændret i systemet
Sekvens	1. Aktør initierer ændring af kreditering 2. Aktør ændrer i informationer om udsendelsen eller i krediterede individer. 3. Aktør gennemser krediteringslisten og acceptere den. 4. Krediteringslisten ændres i systemet.
Alternative scenarier	a. Aktør afslutter proceduren før krediteringen er tilføjet til systemet .

Klassekandidater	Attributter	Funktionskandidater
Kreditering System Producent Administrator Aktør	Kreditering Krediteringsliste	Ændre (kreditering) Ændre (Information) Tilføje (kreditering) Fjerne (krediteringer) Logge ind Afslutte (ændring af kreditering) Acceptere (ændringer)

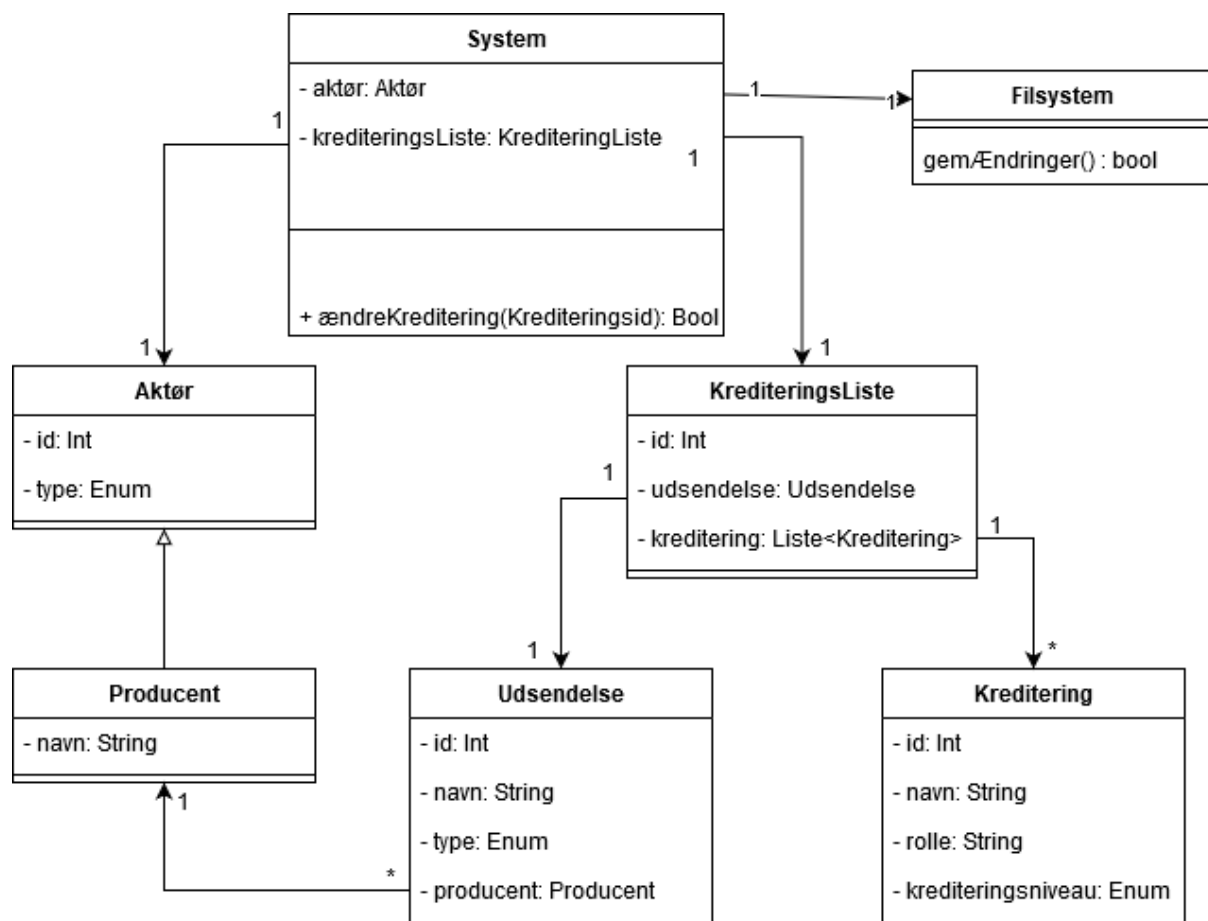
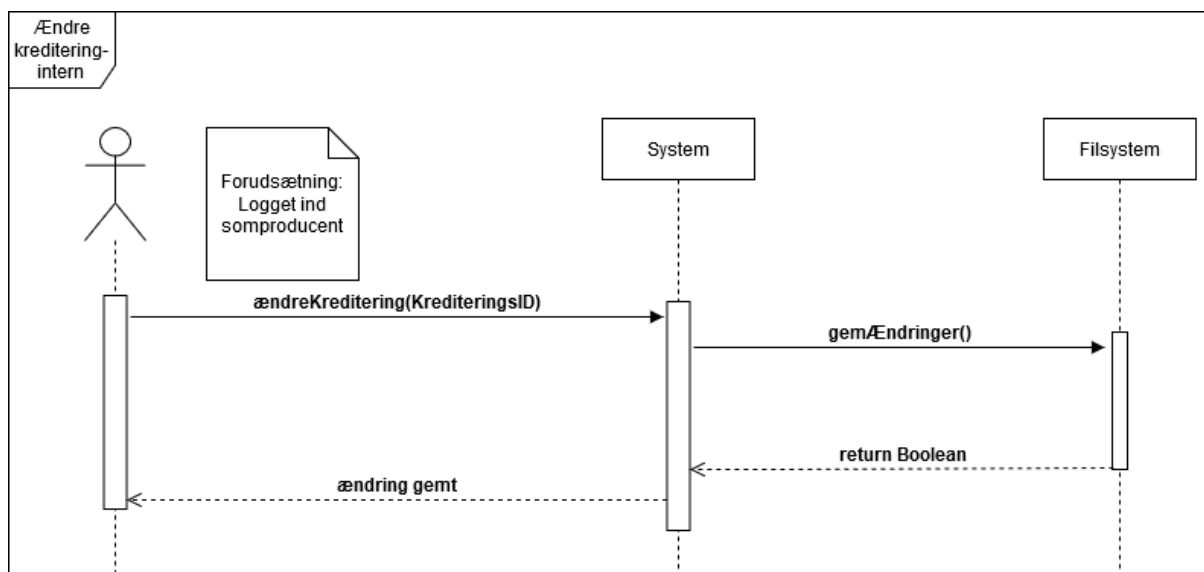
Klassenavn	Ansvar	Samarbejdspartnere
Kreditering	Indeholder information om de krediterede, om projektet og producenten	System
System	Håndterer ændringer i krediteringer	Kreditering
Aktør	Indeholder informationer om alle relevante aktører i systemet.	System
Producent	Indeholde information om en bestemt producent	System

TV 2 Krediteringsprogram

Analyse Klasse Diagram



TV 2 Krediteringsprogram



TV 2 Krediteringsprogram**K. Inceptionsdokument****Patrick Holmquist Andreasen**pandr20@student.sdu.dk
497508**Stefan Profft Larsen**stefa20@student.sdu.dk
497743**Johan Brockstedt**jobro19@student.sdu.dk
474527**Vejleder:****Rune kammersgaard Gregsen**

rkgr@mmmi.sdu.dk

Troels kaldautrkal20@student.sdu.dk
497065**Daniel Smidstrup**dasmi20@student.sdu.dk
495433**Emil Madsen**emima20@student.sdu.dk
496459**HOLD C-10****SDU** **Inceptionsdokument**

semesterprojekt
2. semester foråret 2021

University of Southern Denmark
Civilingeniør i Software Engineering

Det Tekniske Fakultet
Mærsk Mc-Kinney Møller Institut
Campusvej 55, 5230 Odense M**Kursuskode: SE2-PRO**
afleveringsdato: kl 12, 28. maj 2021

TV 2 Krediteringsprogram**Titelblad**

TV2 Krediteringsprogram

Semesterprojekt – 2. semester forår 2021
Kursuskode: SE2-PRO
Projekt periode: 1. februar - 28. maj 2021
Vejleder: Rune Kammersgaard Gregsen

Civilingeniør i Software Engineering
Syddansk Universitet
Det Tekniske Fakultet
Mærsk Mc-Kinney Møller Institut

Forfattere

Patrick Holmquist Andreassen

pandr20@student.sdu.dk

Johan Brockstedt

jobro19@student.sdu.dk

Troels Kaldau

trkal20@student.sdu.dk

Stefan Profft Larsen

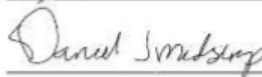
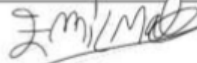
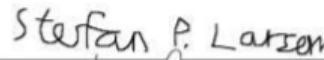
stefa20@student.sdu.dk

Emil Madsen

emima20@student.sdu.dk

Daniel Smidstrup

dasm20@student.sdu.dk

**Opgave**

Opgaven fylder 20 sider*

*Eksklusive forside, titelblad, indholdsfortegnelse, referenceliste og bilag.

TV 2 Krediteringsprogram

Indholdsfortegnelse

1 Indledning	1
1.1 Projektets rammer	1
1.2 Mål for inceptionsdokument	1
1.3 Problemanalyse	1
1.4 Problemformulering og afgrænsning	3
1.4.1 Problemformulering	3
1.4.2 Underspørgsmål	3
1.4.3 Afgrænsning	4
2 Faglige vidensgrundlag	4
2.1 Begrebsdefinitioner	4
2.2 Besvarelse af underspørgsmål	5
2.2.1 Kreditering	5
2.2.2 Reglerne for kreditering	5
2.2.3 Omkostninger ved sendetid	5
2.2.4 TV2 krediteringer	5
2.3 Metoder	6
2.3.1 UP - unified process	6
2.3.2 SCRUM-modellen	6
2.3.3 MoSCoW-modellen	6
2.3.4 UML - Unified Modeling Language	6
2.4 Relevante eksisterende løsninger	6
3 Overordnet kravspecifikation	7
3.1 Brugerkrav	7
3.2 Brugsmønster diagram	11
3.3 Supplerende krav	12
4 Kritiske risici	12
5 Prioritering	14
5.1 Moscow Analyse	14
5.2 Prioriteringsliste	15
6 Metode i elaborationsfasen	16
6.1 Primære proces modeller	16
6.2 Analyse og design	17
6.3 Implementering	17
6.4 Test	17
7 Resurser	18

TV 2 Krediteringsprogram

8 Konklusion	19
A Procesdokumenter	20
A.1 Samarbejdsaftale	20
A.2 Vejlederkontrakt	24
B Bilag	25
B.1 Logbog	25
B.2 TV2 Krediterings Regler	25
B.3 Formelt review referat	26
B.4 Checkliste for Inceptionsdokument	28

TV 2 Krediteringsprogram

1 Indledning

Projektet indledes med en beskrivelse af projektets rammer, mål for inceptionsdokumentet og en analyse af det givne problem.

1.1 Projektets rammer

Projektet er baseret på et case-dokument udleveret af SDU og udformet af TV2. Casen introducerer kort TV2 og forklarer herefter problemet. Dette opfølges med en outline, af den type løsning der ønskes, samt krav til den færdige løsning.

Problemet, som casen ligger op til, er visning af krediteringer i slutningen af TV2's programmer. TV2 afsætter 30 sekunder i slutningen af hvert program til at vise reklamer. De vurderer, at disse 30 sekunder i slutningen af hvert program har en værdi på 60 millioner kr. om året, baseret på det indhold der ellers kunne vises i samme tidsramme.

Løsningen, som casen ligger op til, er et krediteringssystem, der skal digitalisere programmets krediteringer, så krediteringerne ikke længere behøver at blive vist efter hvert program. I stedet placeres krediteringerne på en frit tilgængeligt online platform, så seere selv kan opsøge dem ved interesse.

Casen opsætter en række krav til det endelige produkt. Efter vejledning af SDU (projektudbyderen), er der dog en række af disse krav, der ikke skal efterleves eller arbejdes med.

1.2 Mål for inceptionsdokument

Inceptionsfasen har til formål at give et generelt overblik over projektet. Inceptionsfasen skal give overblik over krav, afgrænsninger og risici, hvilket derfor afdækker udfordringerne i projektet. Ud fra dette dannes en plan for hvordan casen skal løses, hvordan kravene opfyldes, afgrænsningerne overholdes og risiciene undgås. Fasen danner strukturen for projektets overgang fra problem til løsning.

I inceptionsfasen afdækkes casen først med en problemanalyse. Problemanalysen udstykker det relevante information fra de udleverede dokumenter. Kravene, eliciteret fra den udleverede dokumentation, bearbejdes. De uddybes, udvides, visualiseres og prioriteres. Ud fra den indsamlede information udledes projektets udfordringer. Hertil udtænkes strategier til at overkomme udfordringerne og/eller at undvige dem. Til sidst udledes en metode til videre udvikling af projektet.

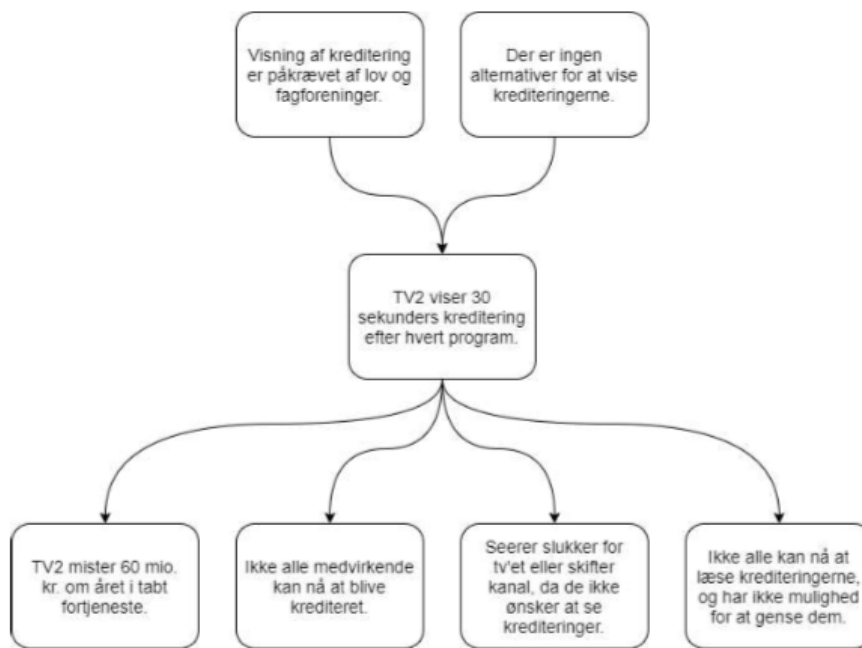
Det endelige mål for inceptionsfasen, er at stå tilbage med en komplet kravspecifikation, brugsmønstermodel og beskrivelse hertil, samt overblik over kritiske risici og løsninger. Gruppen skal endvidere vurdere, på baggrund af dette, om projektet kan gennemføres.

1.3 Problemanalyse

For at kunne arbejde med problemet er det vigtigt at afdække dets årsag og virkning. Det er derfor vigtigt at analysere problemområdet. Til dette er der udformet et problemtræ. Det

TV 2 Krediteringsprogram

mest fundamentale fremsatte problem i casen er: At TV2 viser 30 sekunders kreditering efter hvert program. Dette benyttes derfor som udgangspunktet i problemtræet.



Figur 1: Problemtræ

Problemtræet når frem til 2 identificerede årsager til TV2's problem.

Det ene er kravet om krediteringer. TV2 benytter krediteringer til at aflønne de medvirkende. Dette påkræver at krediteringerne udarbejdes. Kravet om at krediteringen skal vises, eller være offentligt tilgængelig, har dog ikke noget med aflønning at gøre. Kunne TV2 få en aftale på plads, hvor visning eller offentliggørelse af krediteringer ikke var nødvendigt, ville dette løse deres problem.

Det andet problem er manglen på et alternativ. I øjeblikket har TV2 ikke andre muligheder, for at vise krediteringer, end at vise dem efter et endt program. Et sådant alternativ vil kunne udformes som en softwareløsning. Problemet ligger herved indenfor projektets løsningsdomæne og vil derfor blive arbejdet med.

Der er 4 identificerede virkninger af det fundamentale problem. Disse gennemgås i prioriteret rækkefølge, fra det mest relevante til det mindst.

TV2 har udformet casen der arbejdes med, og er derfor den hovedsagelige interessant. Det kan aflæses fra casen, at deres primær bekymring, er de 60 millioner kr., de mener går tabt.

De medvirkende, der krediteres, har indflydelse på de fagforeninger, der bestemmer,

TV 2 Krediteringsprogram

hvem der skal krediteres. De må derfor også antages som en relevant interessent. Med den nuværende krediteringform, med de 30 sekunders kreditering, kan alle ikke nå at blive krediteret. Det kan ikke vurderes, hvor stort et tab dette er for de medvirkende, men må antages, at det har en relevans, da det er noget deres fagforeninger kæmper for.

Folk skifter primært kanal når et program slutter. Det er et naturligt stoppunkt, enten til at slukke tv'et, eller til at se om der er mere spændende programmer andre steder. Hvis TV2 kunne gå direkte til det næste program, istedet for at holde en 30 sekunders pause til kreditering, ville de have nemmere ved at beholde deres seeres opmærksomhed. Dette vil sikre flere seere, og derved flere reklame-penge.

For at få så mange krediterede på den korte tid, er hastigheden på krediteringen ofte høj. Dette gør det sværere for seere at læse, hvem der har medvirket. Dette kan formodes at være et lille tab for nogle få seere. Dog, som før nævnt, må de krediterede antages at drage fordel ved at blive krediteret. Det antages derfor at være et tab for de krediterede, hvis krediteringen er for hurtige eller utydelig.

1.4 Problemformulering og afgrænsning

Problemanalysen definerer problemområdet, men afdækker ikke de krav opstillet i casen, der definerer løsningsområdet. Det bliver her beskrevet, at TV2 ønsker løsningen i form af et krediteringssystem. I systemet skal producenter kunne oprette krediteringer, som seere kan få adgang til. problemformuleringen er herved baseret både på problemområdet og løsningsområdet. De konkrete krav bearbejdes under overordnet kravspecifikation.

1.4.1 Problemformulering

"Hvordan udvikles et krediteringssystem, der overholder de nuværende regler for kreditering, gør det simpelt for producenter at tilføje og ændre krediteringer, gør det simpelt for TV2 at verificere og videresende krediteringer, og gør det simpelt for seerne at se krediteringer, med minimal omkostning af sendetid?"

1.4.2 Underspørgsmål

- Hvad er kreditering?
- Hvad er reglerne for kreditering?
- Hvad forstås ved "omkostninger af sendetid"?
- Hvordan forstås "simpelt" i problemformuleringens sammenhæng?
- Hvordan skal producenter have mulighed for at behandle deres krediteringer?
- hvordan skal TV2 have mulighed for at verificere krediteringer?
- Med hvilket formål skal krediteringerne videresendes?
- I hvilket format skal krediteringerne kunne videresendes?

TV 2 Krediteringsprogram

1.4.3 Afgrænsning

Problemet lægger op til brugen af en lang række teknologier herunder hjemmesider, mobil-applikationer, servere, udnyttelse af eksternt live-data gennem API's, samt selv at udbyde live-data gennem én eller flere API's. Mange af disse teknologier er dog blevet afgrænset fra SDUs (projektudbyderens) side:

- Systemet skal kun fungere for danske producenter.
- Systemet skal ikke være online, server-side, eller have rest-API.
- Systemet skal være én integreret applikation.
- Systemet fokuserer ikke på at være kompatibelt med andre systemer.
- Systemet vil være en demo, der ikke kan refereres til fra tv-programmer.

Ud fra afgrænsningerne, vil der blive arbejdet med en solitær, offline applikation. Løsningsområdet er derfor primært indskrænket til arbejde med datahåndtering og brugerflade.

2 Faglige vidensgrundlag

Underspørgsmål til problemformuleringen, stillet i indledningen, samt relevante metoder, koncepter og værktøjer, beskrives i følgende afsnit. Dette skal sikre en fælles forståelse af relevante emner, for alle gruppemedlemmer, samt rapportens læsere.

2.1 Begrebsdefinitioner

Begreb	Forklaring
Omfakturere	At omstrukturere kode uden at ændre på dets funktionalitet.
PostgreSQL	Relationel database management system.
XML	Extensive Markup Language, en fleksibel måde at dele informationer mellem personer og systemer, både offline og online.
CSV	Comma Separated Values, en komma separeret fil som kun indeholder tal og bogstaver. God til at samle større mængder text-baseret data.
Sprintlog	Prioriteret liste over opgaver der skal udføres.
DBMS	Database Management System, et system der håndterer kommunikation med en database.
Seer	En person der ser et program, i et givent øjeblik.
Tv-udbyder	Et firma der tager penge for, at give adgang til et antal kanaler.
Kanal-udbyder	Et firma der håndterer kanalernes indhold.
Producent	Et firma der producerer indholdet, der vises på kanalerne.

Tabel 1: Begrebsdefinitioner

TV 2 Krediteringsprogram

2.2 Besvarelse af underspørgsmål

Underspørgsmålene blev udarbejdet i problemanalysen, for at understøtte problemformuleringen. Målet er, at give den mest nærliggende baggrundsviden, til at besvare problemformuleringen.

2.2.1 Kreditering

Kreditering er dokumentation for, hvem der har bidraget til et produkt. Dette er specielt relevant i produktion af film, serier, artikler, bøger, studier og andre midlertidige projekter. Uden kreditering i disse typer projekter, kan det være besværligt at bevise, hvem der har bidraget med hvad.

2.2.2 Reglerne for kreditering

Der er lovkrav om kreditering ved brug af andres kreative materiale. Retten til kreditering kan afgives ved aftale. Ved køb, leje eller lån af kreativt materiale, medfølger der derfor en aftale om den aftalte kreditering, der skal overholdes.

2.2.3 Omkostninger ved sendetid

Flow-tv har 2 hovedindtægtskilder. Dette er seernes betaling til deres tv-udbydere og reklamer. Seernes betaling til tv-udbyderen er fast og det er derfor kun kanal-udbyderen, der modtager betaling for reklamer. Reklamer er en stor indtægtskilde og producerer indtægt alt efter hvor mange seere der er og hvor meget sendetid der sættes af til at vise reklamer. Hvert minuts sendetid der ikke sendes reklamer, kan derved ses som en omkostning for kanal-udbyderen.

2.2.4 TV2 krediteringer

TV2 krediterer alle de individuelle aktører, der har deltaget i produktionen. TV2 inddeler dem i 4 kategorier:

- *Individer der har deltaget kreativt i produktionen:* Personerne her skal krediteres individuelt og firmanavne må derfor ikke indgå. Dette gælder individer, der har bidraget med/som:
Tilrettelægger, skuespiller, musiker, manuskriptforfatter, instruktør, komponist, tekstforfatter, kostumer, regissør, rekvisitør, scenograf, kamera, lyd, lys, redigering, producer, associeret producer, billedmixer, tekstere, mm.
- *Sponsorkreditering:* Kategoriseret efter hvad de har sponsoreret med.
- *Kreditering af ydelser af ikke-økonomisk art:* F.eks. "Tak til drengekoret fra Randers".
- *Afslutning af kreditering:* Krediteringen afsluttes med producentens vignet og logo.

TV 2 Krediteringsprogram

2.3 Metoder

Det faglige vidensgrundlag, bag metoderne benyttet i projektet, beskrives.

2.3.1 UP - unified process

UP er en procesmodel, som kan ses som en kombination af en plan drevet og agil model. Den består af 4 faser og et antal aktiviteter. Aktiviteterne strækker sig over faserne og er ikke fastlåste til én fase. Nogle aktiviteter bliver dog primært bearbejdet i én fase hvilket minder om den plan drevne model. Andre udføres gentagende gange, hvilket ligner den agile iterative model. Alle aktiviteter er dog flydende og kan påvirkes i alle faser.

2.3.2 SCRUM-modellen

Scrum er en agil iterativ model. Den består af et udefineret antal af 2-4 ugers sprints, hvorigennem der tilføjes et antal af features til det endelige produkt. Hvert sprint består af 4 faser: planlægning, udvikling, test og vurdering. SCRUM kan derved ses som en gentagelse af en plan drevet model. Forud for disse sprints vil der være en proces, hvor det overordnede produkt defineres.

2.3.3 MoSCoW-modellen

MoSCoW er en model til prioritering af ideer og funktioner. Det opdeler dem i det, der er nødvendigt (MUST), vigtigt (SHOULD), relevant (COULD) og ikke relevant (WOULD).

2.3.4 UML - Unified Modeling Language

Unified Modeling Language er en standard for udsende og opbygning af diagrammer og bruges til at beskrive struktur og forløb i objekt-orienterede softwaresystemer. UML består overordnet af to typer af diagrammer; strukturdiagrammer og adfærdsmønstre. Strukturdiagrammer er diagrammer som klassediagrammet, hvilket beskriver den konkrete struktur for objekt-orienteret softwaresystemer. Adfærdsmønstre, som brugsmønstre, beskriver hvordan systemets omverden, og objekterne i systemet arbejder sammen.

2.4 Relevante eksisterende løsninger

Det har ikke været muligt at finde eksisterende krediteringssystemer. Det mest relevante har været en krediteringsvisningsteknik, hvor rulleteksterne kun optager en del af skærmen når de vises. Herved kan reklamer, for andre ting, vises samtidigt. Så vidt vi har kunnet se, reklameres der kun for nye programmer. Der er eventuelt regler om, at der ikke må vises betalte reklamer samtidigt med rulleteksterne.

TV 2 Krediteringsprogram**3 Overordnet kravspecifikation**

Kravspecifikationen forsøger at identificere nødvendige og mulige funktioner og egenskaber for produktet. Kravene skal skabe et overblik, både over produktets minimale omfang, og potentielle maksimale omfang. Fremgangsmåden er først og fremmest at identificere aktører, og derefter at identificere, detaljere og visualisere relevante brugsmønstre.

3.1 Brugerkrav

Først udarbejdes en liste over de aktører der påvirker systemet. Da disse aktører er de eneste der skal anvende systemet, er de de eneste der definerer systemets brugerkrav. TV-tid er en delvist aktuel aktør. Ifølge projektets afgrænsninger vil produktet ikke være online, og vil derfor ikke kunne kontakte TV-tid. TV-tid som aktør er dog aktuel for det potentielle færdige produkt og medtages derfor til refleksion.

Aktør	Beskrivelse	Mål
Administrator	Aktør ansvarlig for systemets integritet.	• Tilføje brugere • Ændre brugere • Ændre krediteringslister • Sammenflette krediteringer til én person
Producent	Aktør med mulighed for at tilføje og ændre.	• Tilføj egne krediteringer • Ændre egne krediteringer • Sammenflette krediteringer til en person
Offentlig	Aktør der har adgang til at søge efter og se krediteringer.	• Søg krediteringer • Se krediteringer
TVtid	Aktør der leverer informationer om aktivt flow-tv.	• At levere flow-tv programinformation

Tabel 2: Aktørliste

TV 2 Krediteringsprogram

Fra aktørerne eliciteres brugerkravene, de opstilles i en brugsmønsterliste. Først og fremmest, skal brugsmønstrene udgøre de funktioner der er nødvendige for et funktionelt krediteringssystem. Derudover tilføjes andre brugsmønstre, der potentielt kan være værdiskabene for aktørerne. Brugsmønstrene uddybes med id, navn, formål og fremgangsmåde for udførelse.

ID	Brugsmønster	Formål	Fremgangsmåde
1	Tilføj bruger	Tilføjer en ny bruger	- Nye brugers informationer indgives - Bruger er tilføjet til listen af brugere
2	Søg bruger	Søger efter bruger ved hjælp af søgeord	- En aktør indtaster søgeord - En liste af matchende brugere vises.
3	Ændre krediteringer	Ændre en eksisterende krediteringsliste	- En aktør søger efter en eksisterende krediteringsliste - Der ændres på krediteringslisten - Krediteringslisten opdateres med den nye information.
4	Tilføj kreditering	Tilføjer en krediteringsliste	- Information for ny krediteringsliste indgives - Krediteringslisten tilføjes til listen over krediteringer.
5	Notifikationsliste	En liste af ændringer, relevante for brugeren, vises ved login	En liste af ændringer vises ved login
6	Søg egne krediteringer	Søg efter krediteringslister ved hjælp af søgeord, hvor man kun får egne krediteringer fremvist	- En aktør søger efter egne krediteringer - Aktøren får vist en liste med givne krediteringer
7	Søg kreditering	Søg efter krediteringslister ved hjælp af søgeord	- En aktør søger efter en eksisterende krediteringsliste - En liste af matchende krediteringslister vises
8	Data eksportering	Krediteringsdata kan eksporteres i forskellige formater	- En aktør søger efter krediteringer - Formateringen vælges - En fil eksporteres med den valgte formatering
9	Ændre bruger	Ændre på en eksisterende bruger	- En aktør søger efter en eksisterende bruger - Der indgives nye informationer om brugeren - Brugeren opdateres med den nye information
10	Se detaljer om bruger	Se detaljer om en udvalgt bruger	- En aktør søger efter bruger (9), og har trykket på en af brugerne - Der fremvises information om den valgte bruger
11	Sammenflet krediteringer	Personer krediteret i krediteringslister skal sammenflettes, så flere krediteringer peger på den samme person	- Et navn indtastes ved ændring af krediteringsliste - En liste af personer der matcher navnet vises - Den rigtige person vælges - Krediteringen peger herefter på den person
12	Se krediteringsliste	Se en liste med alle dem som er krediteret ud fra de givne søgeord	- En aktør søger efter krediteringer (7) - Aktøren får vist en liste med givne krediteringer
13	Flow-tv krediteringsliste	En liste af krediteringer fra aktive flow-tv programmer	En liste af krediteringslister fra aktive flow-tv programmer
14	Login	At verificere en bruger	- brugeren afgiver logininformationer - Systemet verificere informationerne og omdirigerer brugeren til den korrekte side.

Tabel 3: brugsmønsterliste

Ud fra brugsmønsterlisten, tabel 3, udvælges de mest essentielle brugsmønstre. Systemets mest essentielle funktion er muligheden for, at kunne tilføje, ændre og se krediteringer. Disse udgør grunden for produktets eksistens. Brugsmønster 7 *Søg Kreditering*, 4 *tilføj Kreditering* og 3 *ændre Kreditering* udvælges derfor, da de dækker disse essentielle funktioner, til at blive videreudviklet til detaljerede brugsmønstre.

TV 2 Krediteringsprogram

Søg Kreditering	
ID:	#7
Formål:	At levere credits som matcher søgeordene.
Oversigt:	Den offentlige bruger nøgleord til at søge efter specifikke udsendelser eller personer som de gerne vil have krediteringsoplysninger for. Systemet returnerer en liste med alle krediteringer der matcher de søgte nøgleord.
Aktør:	Offentlig, Administrator
Stakeholders:	Offentlig: Ønsker at være i stand til at søge krediteringer for tv eller film de har set. TV2: ønsker at offentligheden er i stand til at søge efter krediteringer uden at bruge dyrebar skærmtid.
Forudsætning:	Ingen
Postconditions:	Krediteringer der matcher søgeord er returneret.
Sekvens:	1. Offentligheden vil se kreditering for en bestemt film eller tv-show. 2. Offentligheden skriver søgeord for den kreditering de ønsker at finde. 3. Systemet finder og henter al kreditering matchende søgeordet. 4. En liste af krediteringer matchende søgeord er returneret til offentligheden.
Alternative scenarier:	Ingen kreditering matchende søgeordene, og brugeren vil blive informeret.

Tabel 4: detaljeret brugsmønster - Søg kreditering

Tilføj kreditering	
ID:	#4
Formål:	At tilføje krediteringer til systemet.
Oversigt:	Producenten indskrifter informationer om produktet der krediteres, og tilføjer derefter kreditering af individer.
Aktør:	Producent
Stakeholders:	Producent, der ønsker at kreditere medarbejdere. Medarbejdere, der ønsker at blive krediteret ordentligt. TV2, der ønsker at vise og videresende de rigtige krediteringer.
Forudsætning:	Aktør er logget ind som producent eller administrator.
Postconditions:	Kreditering af udsendelse er tilføjet til systemet.
Sekvens:	1. Aktør initierer tilføjelse af kreditering. 2. Aktør tilføjer informationer om udsendelsen der krediteres for. 3. Aktør tilføjer individer der ønskes krediteret med navn og funktion. 4. Aktør gennemser krediteringslisten og accepterer den. 5. Krediteringslisten tilføjes til systemet.
Alternative scenarier:	a. Aktør afslutter proceduren før krediteringen er tilføjet til systemet.

Tabel 5: detaljeret brugsmønster - Tilføj kreditering

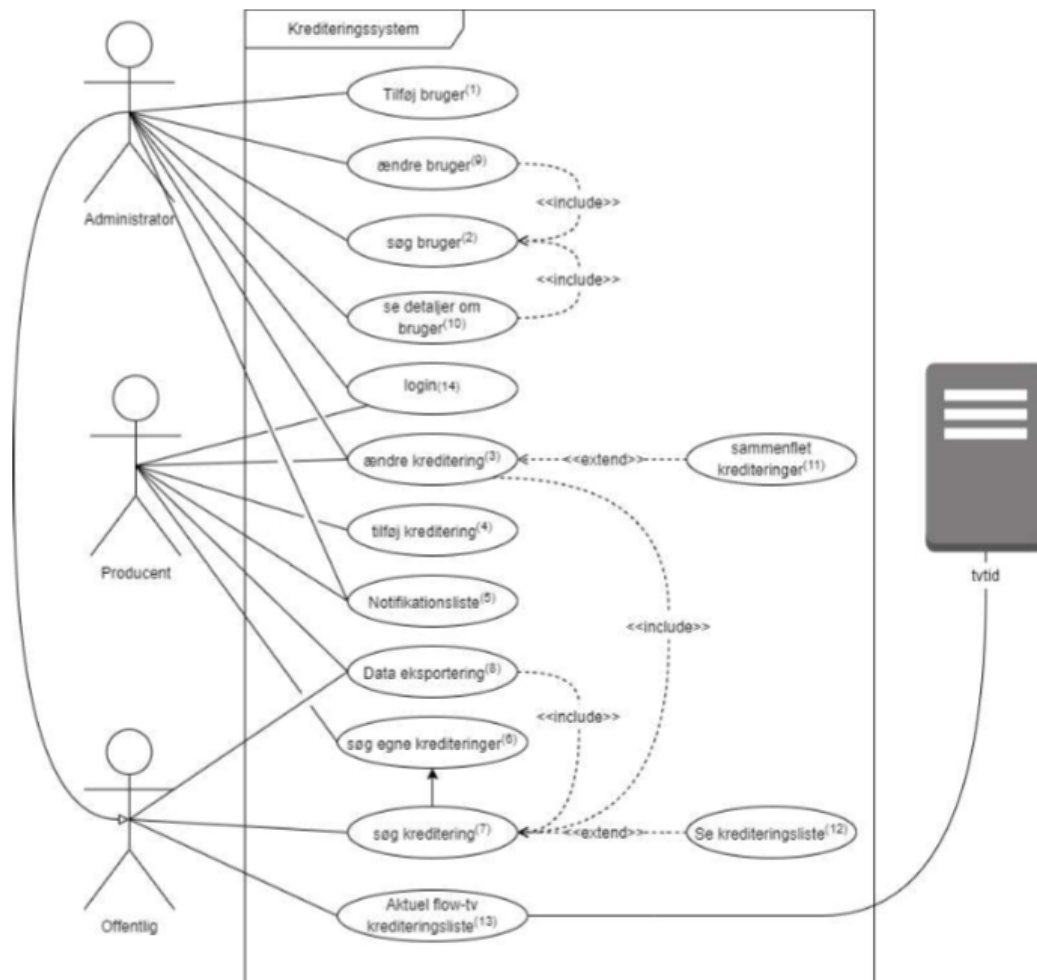
TV 2 Krediteringsprogram

Ændre kreditering	
ID:	#3
Formål:	At ændre krediteringer i systemet.
Oversigt:	Producent eller administrator udvælger krediteringsliste de ønsker at ændre, og tilføjer herefter ændringer.
Aktør:	Producent, Administrator
Stakeholders:	Producent, der ønsker at kreditere medarbejdere. Medarbejdere, der ønsker at blive krediteret ordentligt. TV2, der ønsker at vise og videresende de rigtige krediteringer.
Forudsætning:	Aktør har fundet krediteringen der ønskes ændret. Aktør er logget ind som producent.
Postconditions:	Kreditering af udsendelse er ændret i systemet.
Sekvens:	1. Aktør initierer ændring af kreditering 2. Aktør ændre informationer om udsendelsen eller i krediterede individer. 3. Aktør gennemser krediteringslisten og acceptere den. 4. Krediteringslisten ændres i systemet.
Alternative scenarier:	a. Aktør afslutter proceduren før krediteringen er tilføjet til systemet.

Tabel 6: detaljeret brugsmønster - Ændre kreditering

TV 2 Krediteringsprogram**3.2 Brugsmønster diagram**

Brugsmønsterlisten, tabel 3, visualiseres med et brugsmønsterdiagram. Dette skal give overblik over aktørernes kommunikation med brugsmønstrene. Det giver også overblik over brugsmønstrenes indbyrdes kommunikation, i og med at flere brugsmønstre gør brug af hinanden (*include*, *extend*).



TV 2 Krediteringsprogram

3.3 Supplerende krav

Ud fra de generelle brugsmønstre udarbejdes supplerende krav. Disse dækker krav til produktet, der ikke kan opstilles som funktioner. De ikke funktionelle krav, dækker generelle krav til produktets implementering. De funktionelle krav dækker, i mere generelle træk end brugsmønstrene, hvordan produktet skal kunne bruges.

ID	Kategori	Ikke funktionelle krav
S1	Miljøkrav	Skal udvikles i Java miljøet
S2	Miljøkrav	Skal anvende PostgreSQL Database miljø
S3	Designkrav	Skal have en 3-lags arkitektur
S4	Grænsefladekrav	Skal implementere en grafisk brugergrænseflade
S5	Designkrav	Skal separere Presentations-, Domain-, persistence lag
S6	Brugbarhedskrav	Brugergrænsefladen skal være let og intuitiv at bruge for alle aktører

Tabel 7: ikke funktionelle krav

ID	Funktionelle krav
F1	Databasen skal kunne administreres af TV2
F2	Producenter skal kunne tilføje projekter og roller for deres egne projekter
F3	Der skal være offentlig adgang til informationer om projekter og kreditering
F4	En funktionel GUI
F5	Krediteringssystemet skal kunne merge/linke personer som refererer til den samme person i den virkelige verden.
F6	Krediteringssystemet skal kunne eksportere en specificeret mængde af data ud fra programmet til formater som XML og CSV.
F7	Det skal være muligt at søge i krediteringssystemet for at finde bestemt data.
F8	Der skal være en tydelig (Nem at bruge) reference fra TV-programmet til TV-programmet credits i krediteringssystemet.
F9	Det skal være muligt at få informationer i krediteringssystemet fra en given produktion eller flere produktioner.
F10	Der kan være forskellige dele af systemet med forskellige adgang
F11	Man kan hente informationer til financial management
F12	At subscribe til ændringer i systemet, hvorved man får notifikationer når der sker ændringer i systemet.
F13	At se en liste af krediteringer fra aktive flow-tv-programmer

Tabel 8: funktionelle krav

4 Kritiske risici

I dette afsnit bearbejdes trusler mod projektets success. Formålet er at komme på forkant med problemer der enten kan forsinke, skade eller på anden vis forhindre projektet. Bearbejdning af kritiske risici kan indeles i 4 aktiviteter: identificering, beskrivelse, løsning og

TV 2 Krediteringsprogram

overvågning. Heraf vil de første 3 aktiviteter blive gennemgået i afsnittet.

Tabel 9 definerer risikoniveauet brugt i tabel 10 og 11.

Niveau	Beskrivelse
4	Produktet er ikke funktionelt.
3	Produktet er funktionelt men ikke brugbart.
2	Produktet er brugbart, men der er risiko for at det bliver ikke-brugbart.
1	Der er lav sandsynlighed for at produktet bliver ikke-brugbart, men funktionaliteten kan blive nedsat.

Tabel 9: beskrivelse af fatilitetsgrader

ID	Beskrivelse	Risiko	Løsning
tr1	<i>At databasen ikke fungerer efter intentionerne:</i> Det er yderst vigtigt at databasesystemet fungerer efter hensigten. Hvis databasen ikke kan gemme, vise og redigere data efter hensigten, eller hvis databasen ikke kan konsolidere duplikeret data, eller hvis den ikke kan køre operationerne inden for en rimelig tid, vil systemet som en helhed ikke kunne køre.	Fatalitetsgrad: 4 Sandsynlighed: Lav	Ved at teste databasens tilgængelighed og funktionalitet løbende, samt mindre test cases vil vi mindske sandsynligheden for database fejl.
tr2	<i>At de forkerte brugere kan tilgå de forkerte dele af systemet:</i> Det er yderst vigtigt at specifikke områder af systemet kun kan tilgås, af autoriserede brugere. Hvis de rigtige områder af systemet ikke kan tilgås, af de rigtige brugere eller kan tilgås af de forkerte brugere, vil det lede til at systemet skaber flere problemer end det løser, og systemet vil derfor miste sin brugbarhed.	Fatalitetsgrad: 3 Sandsynlighed: Lav	Ved at implementere og teste hårde kriterier i systemet, til at det kun er brugere med tilstrækkelig adgang, der kan tilgå visse dele af systemet, vil vi forsøge at undgå tilgængelighed for uønskede brugere.
tr3	<i>At brugerfladen er ufunktionel eller uoverskuelig:</i> Hvis brugerfladen ikke giver brugerne mulighed for, at udføre kendte funktioner i systemet som de burde have adgang til, vil funktionaliteten af systemet gå tabt.	Fatalitetsgrad: 1 Sandsynlighed: Mellem	Ved at teste systemet på en tredje part (ikke os selv), vil vi forsøge at fange fejl og mangler i brugerfladen.
tr4	<i>At koden kompromitteres ved en fejl:</i> Hvis koden skulle kompromitteres ved en fejl, vil det være et kæmpe tab af tid som skal indhentes.	Fatalitetsgrad: 2 Sandsynlighed: Mellem	Brugen af Git til versions-kontrol vil afhjælpe den største risiko

Tabel 10: tekniske risici

TV 2 Krediteringsprogram

ID	Beskrivelse	Risiko	Løsning
sr1	<i>At vi ikke får de nødvendige informationer under kurserne</i> Semesterprojektets grundlag for at blive fuldført afhænger af den viden og færdigheder gruppen tilegner sig gennem de andre fag. Hvis informationen og undervisning fra fagene ikke er fyldestgørende nok, vil det påvirke udfaldet af projektet.	Fatalitetsgrad: 1 Sandsynlighed: Mellem	En løsning til problemet ligger i, at alle i gruppen deltager aktivt i undervisning.
sr2	<i>Stor uenighed i gruppen der umuliggør samarbejde</i> Et succesfuldt projekt afhænger af et godt samarbejde mellem gruppemedlemmerne. Hvis der opstår stor uenighed i gruppen, hvor det ender ud i at vi for eksempel bliver sure på hinanden, kan vi ende ud i at vi bruger for lang tid på at løse uenigheden end på selve projektet.	Fatalitetsgrad: 4 Sandsynlighed: Lav	Gruppen forsøger at holde personlige diskussioner ude af projektet og husker at give plads til andre.
sr3	<i>Stor uenighed i gruppen om udførelsen af projektet</i> Samarbejdet i gruppen afhænger af enighed og forståelse for projektets udførelse. Hvis en uenighed opstår, kan det ende i en muligvis forhindring i udførelsen af det endelige produkt og rapport.	Fatalitetsgrad: 3 Sandsynlighed: Lav	Gruppen vil gå på kompromis Evt. spørge vejleder til råds.
sr4	<i>For højt ambitionsniveau der ikke kan udføres</i> Et for højt ambitionsniveau kan være en risiko for projektet, da man ofte i sin ivrighed og spænding, kan overse omfanget af det problem man står med.	Fatalitetsgrad: 2 Sandsynlighed: Lav	Sørge for at prioritere de ting som skal med i projektet, bl.a. ved hjælp af moscow, så vores must haves sat af TV2 er opfyldt.
sr5	<i>Misforståede/missede deadlines</i> En stor del af et projekt bygger sig på faste deadlines og yderligere tidsbegrænsninger for udførelse af vise aspekter af projektet. Missede deadlines kan bl.a. få os til at være bagud eller ende ud i katastrofale situationer (missede deadlines for aflevering eller eksamen).	Fatalitetsgrad: 2 Sandsynlighed: Mellem	Ved hvert møde gennemgår gruppen kommende deadlines og sørger for at holde sig til tidsplanen der er sat.
sr6	<i>Ikke udførte kritiske opgaver</i> Kritiske opgaver som enkelte personer har ansvar for kan være en risici, hvis opgaven ikke bliver udført til tide.	Fatalitetsgrad: 3 Sandsynlighed: Lav	De skal planlægges således at de kan overtages af andre, hvis ansvarspersonen ikke har mulighed for at fuldføre opgaven.

Tabel 11: Sociale risici

5 Prioritering

Kravene eliciteret i overordnet kravspecifikation, er ikke garanteret at blive opfyldt. Det er derfor vigtigt at få prioriteret kravene, så de kritiske krav bliver opfyldt, før der arbejdes på ikke-kritiske krav. Dette sikrer at det udarbejde produkt, får den højst mulige grad af relevans for kunden.

5.1 Moscow Analyse

Først prioriteres de supplerende krav med MoSCoW modellen. Denne inddeler kravene i 4 kategorier:

- MUST: nødvendige krav
- SHOULD: meget relevante krav
- COULD: relevante krav
- WON'T: fravalgte krav

TV 2 Krediteringsprogram

Must have	
M01	• Databasen skal kunne administreres af TV2.
M02	• Producenter skal kunne tilføje projekter og roller for deres egne projekter.
M03	• Der skal være offentlig adgang til informationer om projekter og kreditering.
M04	• En funktionel GUI.
Should have	
M05	• Krediteringssystemet skal kunne merge/linke personer som refererer til den samme person i den virkelige verden.
M06	• Krediteringssystemet skal kunne eksportere en specificeret mængde af data ud fra programmet til formater som XML og CSV.
M07	• Det skal være muligt at søge i kreditering systemet for at finde bestemt data.
M08	• Der skal være en tydelig (nem at bruge) reference fra TV-programmet til TV-programmets kreditering i systemet.
M09	• Det skal være muligt at få informationer i krediteringssystemet fra en given produktion eller flere produktioner.
M10	• Der kan være forskellige dele af systemet med forskellige adgang.
M11	• Man kan hente informationer til financial management.
Could have	
M12	• Man skal kunne se ændringer i systemet, hvorved man får en notifikation.
Won't have	
M13	• At forbinde Krediteringssystemet til andre systemer som Yousee Play, Boxer Play, Stofa, etc.

Tabel 12: Moscow analyse

5.2 Prioriteringsliste

For at have en prioriteret arbejdsliste, omformuleres kravene, så de kan udføres separat. Derefter opstilles de i en prioriteret liste, fra mest relevante til mindst relevante. Prioriteringen er udarbejdet ud fra alle kravenes relevante egenskaber:

- Hvor betydningsfuldt kravet er for produktet.
- Hvor stort et læringsmæssigt udbytte det har.
- Hvor kritisk kravet er for produktets funktionalitet.
- Hvor stor risiko der er for, at kravet ikke kan opfyldes, og arbejdet derfor bliver spildt.
- Den logiske fremgang i udviklingen af softwareløsningen.

Listen benyttes igennem de næste faser, for at sikre det højest mulige udbytte af arbejdet. Prioriteringslisten er opdelt i første iteration (1-4) og anden iteration (5-9). Første iteration skaber et funktionelt system, med en GUI hvorfra man kan tilføje, ændre, søge og se krediteringer. Herunder står P1 for at lave det basale persistensniveau. P2 skaber funktionerne til at kommunikere med persistensniveauet fra en java applikation. P3 skaber 3 Interfaces der hver eksponerer forskellige dele af applikationens funktionalitet. De forskellige Interfaces funktionalitet har dog forskellig adgang fra GUI'en. En adgang for en 'Super User', som har til formål at administrere systemet (tilføje og eller ændre brugere, se detaljerne på dem, ændre i krediteringerne). En adgang for producenter, hvor i de kan tilføje/ændre i krediteringer. Producenterne skal også have mulighed for at søge i krediteringer de selv har tilføjet. Til sidst skal der være en adgang for offentlige brugere, hvor de kan tilgå krediteringerne, dog kun som Read-Only. Eksportering af krediteringer skal også være mulig. Disse Interfaces skal håndtere og begrænse adgangen for disse brugere, så de dér igennem kun kan tilgå de dele af systemet, som de bør have adgang til. P4 skaber en GUI der kan tilgå systemet gennem Interfacene fra P3.

TV 2 Krediteringsprogram

ID	Prioritering	Elementer i projektet
P1	1	Lokalt filsystem (1. iteration)
P2	2	Det skal være muligt at tilføje projekter og roller. • Disse tilføjede roller skal overholde de givne regler for kreditering • Det skal være muligt at redigere projekter og roller • Det skal være muligt at fjerne projekter og roller
P3	3	Systemet skal have 3 Interfaces, en til administrering (super user), en til kreditering og en offentlig.
P4	4	Basalt Graphical User Interface (GUI) • GUI'en skal inddeles i 3 dele: administrering, kreditering og offentlig. • Under administrering skal kunne ændres i krediteringer • Under kreditering skal kunne tilføjes krediteringer og ændres krediteringer. • Under offentlig skal krediteringer kunne søges og ses.
P5	5	Database (2. iteration)
P6	6	Systemet skal kunne merge/linke forskellige krediterede navne, så de peger på den samme unikke person.
P7	7	Der skal kunne oprettes forskellige administrerings- og krediteringsbrugere som kan logge ind.
P8	8	Databasen skal logge hvem der har foretaget ændringer i systemet.
P9	9	Oprettede Krediteringer skal accepteres af TV2 og krediteringer skal have en status for hvorvidt de er accepterede. Efterfølgende ændringer af krediteringer foretaget af producenter skal ændre status tilbage til 'ikke accepteret'.
P10	10	Det skal vise den aktuelle produktion fra flow-tv.
P11	11	Det skal være muligt at eksportere en specificeret mængde af data ud fra programmet til XML-og CSV-filer.
P12	12	Det skal være muligt for administrator at se en liste med alle ændringer lavet i systemet, og hvem der har lavet disse ændringer.

Tabel 13: Kravprioriteringsliste

6 Metode i elaborationsfasen

Her beskrives de metoder som skal bruges i elaborationsfasen og hvordan de bruges.

6.1 Primære proces modeller

SCRUM er en agil metode der effektivt gennemgår aktiviteterne for solid softwareudvikling. Det er en iterativ proces og fokuserer på versionsudvikling. UP er en mere overordnet proces, der har fokus på projektforløbet som en helhed.

De 2 metoder kombineres, så UP anvendes til strukturering af alt i projektet der ikke har med den direkte softwareudvikling at gøre. Når det kommer til at udvikle funktionelle versioner, skiftes til SCRUM, hvor et sprint igangsættes og en kodeversion udvikles. Herefter kan der skiftes tilbage til UP, for at vurdere versionens påvirkning af det overordnede projekt, eller et nyt sprint kan igangsættes, for hurtigt at lave en ny version. Kombinationen skal sikre sammenhæng i projektet, uden at ofre effektiviteten i udviklingsprocessen.

TV 2 Krediteringsprogram

6.2 Analyse og design

Til at analysere og designe produktet, vil der blive anvendt en række UML-diagrammer. De mest prominente er brugsmønsterdiagram og klassediagram. Brugsmønsterdiagrammet vil blive brugt til at skabe overblik, over hvordan interaktion med systemet skal foregå. Dette giver en ramme for hvordan systemet skal designes. Klassediagrammer vil blive konstrueret, for at give et overblik over kodens struktur. Dette er især relevant før man går i gang med at kode, så man ikke spilder tid med et skrive kode der efterfølgende skal omfaktureres. Det hjælper også med at give overblik over strukturen af det endelige produkt.

Til produktdesign vil lagdeling og facademønsteret have centrale roller. Lagdeling skal hjælpe med at gøre koden overskuelig, testbar, vedligeholdelsesvenligt og genbrugeligt. Facademønsteret skal underbygge lagdelingen, ved at gøre kommunikation mellem lagende forudsigelig og konsekvent.

Ved anden iteration skal der tilføjes en database til systemet. Til konstruktion af databasen benyttes ER-diagrammer. ER-diagrammer skaber et overblik, over hvordan virkelige elementer hænger sammen, for at dette kan konstrueres som en database. ER-diagrammer giver et konceptuel indblik i hvordan en database hænger sammen. Den skaber system i den logiske brug af databasen. Herfra går man over til et klassediagram, hvilket giver en model, mere nærliggende til den reelle udformning af databasen. Ud fra klassediagrammet, skrives den konkrete relationel-database.

6.3 Implementering

Til implementering vil der blive anvendt sprintlog, issueboard og iterering. Sprintlog er en prioriteret liste, over opgaver der skal udføres, før en version er færdig. Sprintloggen skal sikre, at en komplet testbar version er færdig indenfor deadline, så den kan vurderes. Issueboardet skal hjælpe med at organisere arbejdsbyrden, og holde styr på sprintlog opgaverne. Den giver et visuelt overblik over hvor langt man er i processen, hvem der arbejder på hvad, og hvad man kan gå i gang med. Implementeringen vil forgå i iterationer, også kaldet sprints. Uden iterering, risikerer man at forsøge, at implementere for mange funktioner samtidigt. Man risikerer derved at nå den endelige deadline, uden et samlet, funktionelt produkt, der kan testes og vurderes. Iterering sikrer, at der bliver udformet en funktionel version af projektet undervejs. Herved sikrer man, at man har et fungerende produkt til den endelige deadline, selv hvis det ikke indeholder alle de påkrævede funktioner.

6.4 Test

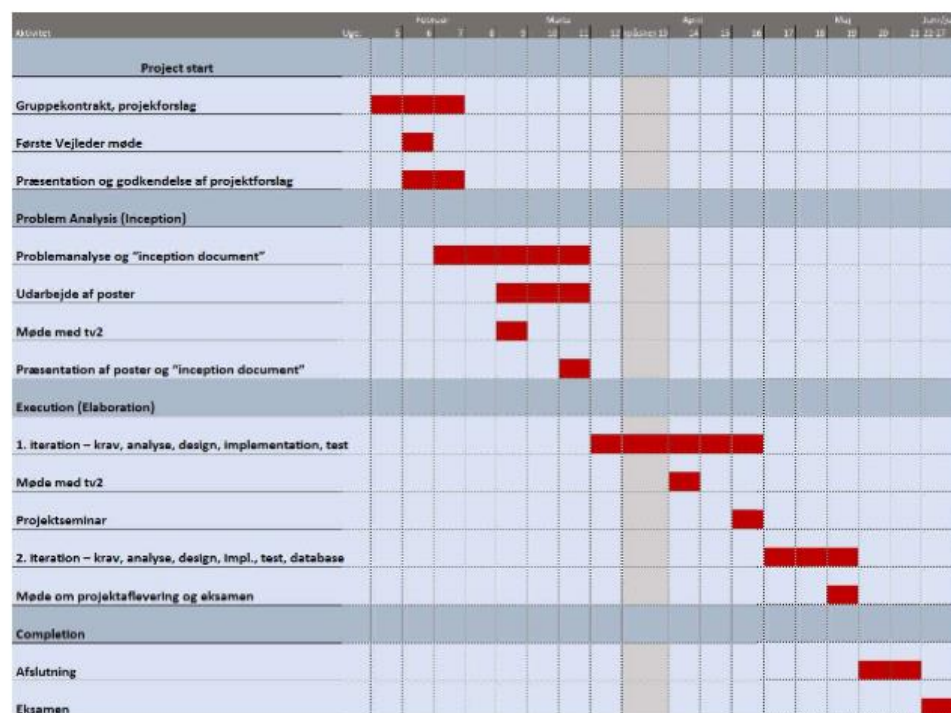
Til at teste den udviklede kode, vil der bruges både modultest, funktionstest og stresstest. Modultest er en test af individuelle komponenter, af den færdige kode. Dette skal sikre, at dele af koden der sjældent aktiveres af produktet, stadig testes. Funktionstest er test af

TV 2 Krediteringsprogram

de reelle funktioner i produktet. Det er en basal test, der sikrer, at produktet kan udføre de funktioner det skal kunne. Stresstest er en test af programmets robusthed. Det skal sikre at programmet kan overleve uventede situationer. Det skal også undersøge, om der er funktioner eller genveje i programmet, der ikke skal være der.

7 Resurser

Der er lavet et GANTT-diagram for at få et overblik over projektet, samt at have en visuel tidsplan.



Figur 2: Gantt-diagram

TV 2 Krediteringsprogram

8 Konklusion

Problemdomænet blev analyseret ved hjælp af et problemtræ, hvor der blev nået frem til 2 årsager og 4 virkninger. Årsagerne til problemet er de lovmæssige krav for at vise korrekt kreditering, samt at TV2 intet alternativ har for visning af kreditering. Den mest væsentligste virkning heraf, set fra et forretningsmæssigt perspektiv, vurderes til at være de 60 millioner kr. TV2 årligt taber i fortjeneste. Der ønskes derfor et krediteringssystem udviklet, der kan dække reglerne for kreditering, uden at vise dem i sendetiden. Den konkrete problemformulering lyder derfor således:

“Hvordan udvikles et krediteringssystem, der overholder de nuværende regler for kreditering, gør det simpelt for producere at tilføje og ændre krediteringer, gør det simpelt for TV2 at verificere og videresende krediteringer, og gør det simpelt for seerne at se krediteringer, med minimal omkostning af sendetid?”

I dokumentet har vi afdækket en kravspecifikation med 3 aktører, 12 funktionelle krav og 6 ikke funktionelle krav. De 3 funktionelle krav er afdækket detaljeret, resten er afdækket overordnet. Der er udformet et brugsmønsterdiagram, med alle aktører og alle funktionelle krav.

Der er dokumenteret 4 tekniske risici samt 6 sociale risici. De er blevet vurderet på Fatalitetsgrad samt sandsynligheden for forekomst. Til hver risici er der udtænkt enten et løsningsforslag eller et forebyggende tiltag.

Først er kravene opsat efter en MoSCoW analyse, baseret på casen, udgivet af TV2. Herefter er kravene blevet prioriteret baseret på deres risici, udbytte og forventede tidskrav. Ud fra den indsamlede information kan det forventes, at kravene til projektet er overkommelige og de identificerede risici kan undgås. Der er stadig kompetencer der mangles til at udføre projektet. Det antages dog, at disse kompetencer erhverves gennem videre undervisning. Metodeafsnittet har afdækket den overordnede fremgangsmåde for næste fase af projektet og Gantt diagrammet har lagt en løs plan for resten af projektet.

Baseret på analyse og prioritering af kravene opsat af TV2, samt en vurdering af risici, vurderes projektet, på baggrund af dette, til at kunne gennemføres.

TV 2 Krediteringsprogram

A Procesdokumenter

A.1 Samarbejdsaftale

Samarbejdsaftale

Gruppe C-10

5. februar 2021

Gruppemedlemmer

	IM	KS	KO	OP	AN	FM	OR	AF	SP
Troels	62	36	35	82	49	10	35	76	57
Emil	57	70	45	51	42	70	25	30	60
Johan	61	59	30	51	32	56	8	19	46
Stefan	0	83	40	42	73	79	7	41	62
Daniel	56	44	60	46	58	32	53	54	36
Patrick	25	45	48	62	38	30	58	57	35

- Troels Kaldau
- Emil Madsen
- Johan Brockstedt
- Daniel Smidstrup
- Stefan Profft Larsen
- Patrick Holmquist Andreassen

Forventninger til projektet

- Vi får anvendt de projektprocesser som vi benyttede i sidste projekt.
- Vi får udvidet projektprocesserne.
- Vi får anvendt viden fra alle 2. semesters fag.
- Vi får anvendt en effektiv og sikker databasestruktur.
- Vi får lavet gode APIs til produktet der nemt kan implementeres af andre.
- Vi får udarbejdet en professionel rapport, sideløbende med projektet.

Forventninger til hinanden

- Alle yder deres bedste.
- Alle er åbne overfor spørgsmål om hjælp.
- Alle er åbne overfor kritik.
- Alle yder konstruktiv kritik til andres arbejde.
- Alle deltager aktivt i diskussioner om projektets udvikling og udfoldning.

TV 2 Krediteringsprogram

- Alle holder øje med arbejdstider og deadlines, og overholder dem.
- Det skrives ASAP hvis man ikke kan overholde tider eller deadlines.
- Alle følger aktivt med i alle andre fag for at få relevant faglig viden til projektet.
- Alle følger aktivt med i projektudviklingen og spørger ind til ting de ikke forstår.
- Alle finder selvstændigt arbejdsopgaver de kan arbejde med i arbejdstiden.
- Alle deltager i arbejdsfordelingen og koordinere arbejdsindsatsen med de andre.
- Alle er detaljeorienterede og bekymrer sig om deres arbejde.
- Man læser korrektur på sit eget materiale.

Arbejdstider og kommunikation

Alle tider er vejledende.

- Tirsdag 10:00 – 16:00
- Fredag 9:00 – 12:00
- Søndag 20:00 - 20:30

Værktøjer

- **Kommunikation:** Discord & Messenger
- **Kodedeling:** Github
- **Dokumentdeling:** Teams
- **Rapportskrivning:** Latex

Gruppemøde og Vejledermøder

- Vi stræber efter at holde faste gruppemøder tirsdag 10 – 16, fredag 9 – 12 og søndag fra kl. 20 samt at have Vejledermøde som en del af tirsdagsmødet. Yderligere møder kan blive aftalt løbende.
- Hvis et gruppemedlem eller vejleder ikke har mulighed for at møde op til et møde eller er forsinket SKAL vedrørende give besked til resten af gruppen hurtigst muligt (evt. et specifikt sted så som Discord eller Messenger). Hvis et gruppemedlem konstant formår at udeblive, bør der tages en seriøs samtale med vedkommende internt i gruppen, eventuelt med tilstedeværende vejleder.

Konflikthåndtering

- Hvert gruppemedlem har eget ansvar for at få vendt problemer med gruppen
- Hvis man har problemer med en anden i gruppen, holder man sproget i en sober tone.
- Man skal ytre utilfredshederne hurtigst muligt, så det ikke når at blive et problem. Tag det op med den enkelte i første omgang, og hvis det ikke bliver løst, så vend det i gruppen.

TV 2 Krediteringsprogram

Logbog

Opgaver:

- Vejledermøde dagsorden
- Problemer/Spørgsmål
- Referat af vejledermøde
- Opnået
- Målsætning til næste gang

Rollefordelingsliste:

- Troels
- Johan
- Daniel
- Patrick
- Stefan
- Emil

Groft Deadlines

- **Week 5:** gruppekontrakt, projektforslag
- **Week 6:** præsentation af projektforslag
- **Week 7:** problemanalyse, "inception document"
- **Week 9:** møde med tv2
- **Week 11:** præsentation af poster og "inception document"
- **Week 12:** 1. iteration – krav, analyse, design, implementation, test
- **Week 14:** møde med tv2
- **Week 16:** projektseminar
- **Week 17:** 2. iteration – krav, analyse, design, impl., test, database
- **Week 19:** Møde om projektaflevering og eksamen
- **Week 20:** Afslutning
- **Week 22-27:** Eksamen

Kontaktinformation

- Troels Kaldau
 - Email: trkal20@student.sdu.dk
 - Mobil: 42803043

TV 2 Krediteringsprogram

- Emil Madsen
 - Mail: emima20@student.sdu.dk
 - Mobil: 61 31 03 06
- Johan Brockstedt
 - Mail: jobro19@student.sdu.dk
 - Mobil: 31 93 84 93
- Daniel Smidstrup
 - Mail: dasm20@student.sdu.dk
 - Mobil 21523907
- Stefan Profft Larsen
 - Mail: stefa20@student.sdu.dk
 - Mobil: 61855864
- Patrick Holmquist Andreasen
 - Mail: pandr20@student.sdu.dk
 - Mobil 50474811

Underskrifter

TV 2 Krediteringsprogram

A.2 Vejlederkontrakt**Gruppens forventninger til vejleder**

Gruppen forventer at vejlederen...

- svarer på vores spørgsmål
- Gradbøjer sikkerheden for sine svar.
- kan pege os i den rigtige retning
- forstår hvad vores projekt handler om
- sætter sig ind i de ting vi sender
- Afbud meldes indenfor 24 timer

Vejleders forventninger til gruppen**Møder**

- Gruppen har møde med vejlederen hver tirsdag et sted mellem 14.15-18
- Ekstra vejledermøder kan eventuelt aftales, efter behov, under tirsdagens vejledermøde.

Fremmøde

- Alle gruppemedlemmer forventes at møde op til vejledermøder.
- Ved forsinkelse eller udeblivelse meddeles dette hurtigst muligt, helst inden mødets begyndelse.
- Der skal meldes afbud til vejlederen og projektgruppen, ved udeblivelse fra vejledermøde.
- Afbud sker over mail til vejleder, hvor gruppemedlemmer tilføjes som CC.

Mødestruktur

- Der skrives referater til hvert møde og føres logbog.
- Skribenten af referatet bestemmes inden hvert møde.
- Dagsorden skal som minimum indeholde 3 ting: Hvad har i lavet siden sidst, hvordan går det i gruppen, hvad skal der laves til næste gang.

Logbogen indeholder

- Alle indkaldelser med dagsorden.
- Alle referater med angivelse af hvem der var til stede og begrundelser for manglende tilstedeværelse.
- Logbog skal efter mødet opdateres på github.

Vejledning og deadlines

- Når der sendes en mail til vejlederen, skal alle gruppemedlemmer tilføjes via CC.
- Dagsorden, spørgsmål og indkaldelse til vejledermøde skal sendes senest kl 12 søndagen inden vejledermødet.

Udnyttelse af vejleder

- Vejlederen udnyttes også til faglig sparring og til at få guidet projektet i den rigtige retning.
- Vejleder benyttes til håndtering af store konflikter der ikke kan håndteres af gruppen.

TV 2 Krediteringsprogram

B Bilag

B.1 Logbog

<https://github.com/Skashe/C-10_sDU_sEM2_sE>

B.2 TV2 Krediterings Regler

<https://producent.tv2.dk/media/1079/bilag-8-krediteringsregler.pdf>

TV 2 Krediteringsprogram

B.3 Formelt review referat**Formelt review 16/3/2021 af gruppe C-10****Generelt**

Generel enighed om at dokumentet skal gennemlæses for korrektur. Der er nogle områder som kan være lidt forvirrende men generelt et godt inceptionsdokument. Der skal bruges noget tid på indledningen, og det skal sikres at tabellerne er læsbare.

- Uenighed om hvorvidt forside er rodet.
- Læs korrektur.
- Tabellerne kan være forvirrende.
- Generelt et godt inceptionsdokument

Gennemgang af alle afsnit:

Alle afsnit er gennemgået en ad gangen, hvor der er givet kritik.

Forside

Uenighed om hvorvidt forsiden var en eller for rodet.

Det skal undersøges om der er bestemte regler for hvordan man må benytte SDU's logo.

Titelblad

Titelblad er som det skal være

Indledning

Skriv om casen i indledning.

Der skal beskrives baggrund og rammer for casen.

Problemanalyse

Tilføj problemtræ i problemformulering!

Problemanalyse er blot gentagning af casen.

For fastlagt på produktet i problemstilling.

For få underspørgsmål.

Beskriv afgrænsningerne.

tabel 1 forvirrende opsætning.

Der mangler afsnit om interessenter.

Afsnit 2.2 og 1.4 samme navn, hvilket er forvirrende.

Faglig vidensgrundlag

2.2 er forvirrende (samme navn som afsnit 1.4).

2.4 er irrelevant (Eksisterende løsninger).

Alle 3 reviewers synes generelt dette afsnit var fint.

Kravspecifikation

baggrund til lister.

Man kunne lav ens opsætning for funktionelle og ikke-funktionelle krav

Ikke særlig meget respons til dette afsnit, der blev udtrykt at afsnit virkede fint, men måske godt kunne uddybes lidt.

TV 2 Krediteringsprogram

Kritiske risici

Nogle forstod overhovedet ikke vores opsætning mens andre syntes det var en smart måde, som de selv ville efterligne.
Risici kan beskrives i dybden

Prioriteringer

Skriv om hvad man får ud af prioriteringer.
MoSCoW skaber ikke overblik, det skal beskrives hvad den bruges til.
De to prioriteringslister skal henvise til hinanden.
Beskriv i dybden.
Brug samme id til samme krav.
API forvirrende.

Metode

Skal skrive individuelle metoder, ikke fremgangsmåde.
Forvirring omkring hvorfor vi har sammenkoblet UP og SCRUM
Hvorfor bruges metoderne?
Nogle synes dette afsnit var fint, mens andre syntes det var forkert opsat, og manglede konkrete beskrivelser af metoderne.

Resurser

Fint gang-diagram, det er som det skal være, men det kunne godt være lidt større.

Konklusion

Der blev ikke snakket om konklusion fra nogle af de tre reviews.

Ændringer der skal laves efter review:

- indledning
- SDU-logo
- tabel formatering
- omdøbning af 2.2
- hvad er underspørgsmål?
- Forklarende afgrænsningsafsnit.
- Læs efter begreber (API, rest-API, database).
- lav api om til interfaces.
- Beskriv yderligere hvad metoder giver.
- Beskriv i konklusionen hvad vi er nået frem til at problemet helt konkret er og roden til problemet.

TV 2 Krediteringsprogram

B.4 Checkliste for Inceptionsdokument

Checkliste for Inceptionsdokument

Indhold	Kriterier for godkendelse	Opfyldt +/-/i
I. Forside	Projekttitel, uddannelsesinstitution, fakultet, institut, uddannelse, semester, kursuskode, projektperiode, vejleder, projektgruppe og projektdeltagere (fornavn, efternavn, sdu-email). Må gerne have illustrationer.	+
II. Titelblad	Samme oplysninger som på forsiden, samt projektdeltagernes underskrifter (Projektdeltagernes aktive deltagelse i projektførelsen anerkendes gensidigt ved projektdeltagernes underskrifter. Må ikke have illustrationer.	+
III. Indholdsfortegnelse	Samlet indholdsfortegnelse for hele projektrapporten. Højest to eller tre niveauer i indholdsfortegnelse, der kan evt. være flere i selve rapporten. Afsnit på niveau 1 og 2 skal være nummererede.	+
1. Indledning	Projektets rammer og baggrunden for projektet. Formål med og mål for inceptionsfasen. Problemanalysen. Problemformulering og afgrænsninger.	+
2. Faglige vidensgrundlag	Begrebsdefinitioner, teori og fagligt vidensgrundlag. Relevante eksisterende løsninger.	+
Hovedtekst		
3. Overordnet kravspecifikation	Overordnet kravspecifikation med - Overordnet brugsmønstermodel: Afgrænsning, brugsmønsterdiagram, aktørbeskrivelser, korte brugsmønsterbeskrivelser, samt detaljeret beskrivelse af udvalgte, essentielle brugsmønstre - Overordnet beskrivelse af supplerende krav	+
4. Kritiske risici	Identificerede kritiske risici, dvs. risici som vil være en trussel mod projektets succes, hvis de ikke bliver afværget: - Er alle kritiske risici identificeret - Er de identificerede risici blevet afværget, eller er der en plan for at afværge dem?	+
5. Prioritering	Prioritering der bygger på kravenes forretningsmæssige betydning (nytte/benefit), arkitektoniske betydning, risiko samt læringsmæssige udbytte.	+
6. Metode i elaborationsfasen	Metode, der vil blive brugt i elaborationsfasen, herunder en overordnet beskrivelse af den måde som UP og Scrum vil blive kombineret i projektet.	+

TV 2 Krediteringsprogram

7. Resurser	Arbejdstidsindsats uge for uge og alt i alt	+
8. Konklusion	Opsummering af resultaterne og diskussionen af dem. Svar på om formålet og målene med inceptionsfasen er opnået.	+
Procesdokumenter		
1. Procesdokumenter	Gruppekontrakt, opdateret Vejlederkontrakt opdateret Belbin gruppeprofil	+
Lag		
A. Logbog	Link til logbogen	+
B. Review	Referat af review	+
C. Checkliste for inceptionsdokument	Udfyldt checkliste	+

TV 2 Krediteringsprogram

L. Udsnit af udført udgivelsestest**Public Overview**

Test	Success kriterie
Anvend forskellige søgninger, som du ved burde vise et bestemt program.	Programmet du søger efter, bliver vist ved alle filtre.
Anvend forskellige søgninger, som du ved ikke burde vise et bestemt program.	Programmet du har fokus på, vises ikke ved nogle af filtrene.
Søg hurtigt gentagende gange, med en søgeændring der burde give ændring i resultatet	Programmets UI-performance bliver ikke medtaget under testen, og programmets generelle performance bliver ikke medtaget efter testen.
Åben og luk menuen	Layoutet ser ud som det skal og opfører sig som det skal i begge tilstande
Åben og luk menuen hurtigt gentagende gange	Layoutet forbliver fornuftigt, og programmets performance bliver ikke medtaget, hverken under testen eller efter.
Navigér til en krediteringsliste	Programmet navigerer til den korrekte side, med den korrekte data, inden for en acceptabel tidsramme.

View Credit

Test	Success kriterie
Åben og luk menuen	Layoutet ser ud som det skal og opfører sig som det skal i begge tilstande
Åben og luk menuen hurtigt gentagende gange	Layoutet forbliver fornuftigt, og programmets performance bliver ikke medtaget, hverken under testen eller efter.
Download krediteringslisten	Filen bliver downloadet til den rigtige lokation, med det forventede filformat, filnavn og filindhold.

Login

Test	Success kriterie
Åben og luk menuen	Layoutet ser ud som det skal og opfører sig som det skal i begge tilstande
Åben og luk menuen hurtigt gentagende gange	Layoutet forbliver fornuftigt, og programmets performance bliver ikke medtaget, hverken under testen eller efter.
Benyt alle kombinationer af korrekt data, forkert data og manglende data i samtlige felter.	Programmet viser de korrekte fejlmeddelelser og logger ikke ind ved brug af forkert eller manglende data.
Forsøg at log in med forkert data hurtigt gentagende gange.	Programmets generelle performance bliver ikke påvirket umiddelbart efter testen.

TV 2 Krediteringsprogram**Admin Overview**

Test	Success kriterie
Anvend forskellige søgninger, som du ved burde vise et bestemt program.	Programmet du søger efter, bliver vist ved alle filtre.
Anvend forskellige søgninger, som du ved ikke burde vise et bestemt program.	Programmet du har fokus på, vises ikke ved nogle af filtrene.
Søg hurtigt gentagende gange, med en søgeændring der burde give ændring i resultatet	Programmets UI-performance bliver ikke medtaget under testen, og programmets generelle performance bliver ikke medtaget efter testen.
Åben og luk menuen	Layoutet ser ud som det skal og opfører sig som det skal i begge tilstande
Åben og luk menuen hurtigt gentagende gange	Layoutet forbliver fornuftigt, og programmets performance bliver ikke medtaget, hverken under testen eller efter.
Navigér til ViewCredit	Programmet navigerer til den korrekte side, med den korrekte data, inden for en acceptabel tidsramme.
Navigér til EditCredit	Programmet navigerer til den korrekte side, med den korrekte data, inden for en acceptabel tidsramme.

TV 2 Krediteringsprogram

M. Brugertest

Introduktion

Følgende dokument er en brugertest for produktet udviklet i semesterprojektet 2021 på uddannelsen Civilingeniør i Software Engineering, udviklet af gruppe C-10.

Test

Introduktion til test

Programmet, der testes, er et krediteringssystem møntet på TV2. Programmet er delt i 3 dele. Den første er beregnet til offentligheden med mulighed for at se, søge og downloade krediteringslister. Den anden er beregnet til producenter, der skal kunne se, søge, oprette, redigere og slette krediteringslister til programmer ejet af producenten selv. Den tredje er beregnet til administratorer hyret af TV2, der skal kunne se, søge og redigere samtlige krediteringslister, og som samtidig skal kunne oprette og afmelde nye producenter.

Gennemgangstest

Testpersonen forsøger at udføre de beskrevne test fra tabel 2 i det udleverede program og bliver bedømt ud fra tabel 1.

Score	Beskrivelse
1	Udført uden besvær.
2	Udført efter let tøven eller orientering.
3	Udført efter fejlforsøg eller større forvirring.
4	Kunne ikke udføres uden yderligere hjælp.

Tabel 1: Bedømmelsesbeskrivelse for gennemgangstest

Test	Score
Åben og luk menu.	1
Søg efter et program, så kun det program bliver vist.	3
Åben en krediteringsliste.	1
Download en krediteringsliste.	1
Gå tilbage.	1
Login som administrator med brugernavn "admin" og passwordet "admin".	1
Tilføj bruger.	1
Ændre email og password på tilføjede bruger.	4
Slet tilføjede bruger.	1
Tilføj producent.	2
Afmeld producent.	1
Genstart producent.	1
Gå ind på databaseloggen og søg efter logs mellem den 20/05/21 og den 22/05/21.	3
Log ud.	1
Login som producent med brugernavn "producer" og passwordet "producer".	1
Opret et show med én kreditering, der krediterer en person med navnet "John Doe".	4
Ændre showets navn, samt den krediterede person til en tilfældig anden person.	4
Slet det tilføjede show.	4
Log ud.	1

Tabel 2: Opgaver og bedømmelse af gennemgangstest

TV 2 Krediteringsprogram

Spørgeskema

Testpersonen besvarer spørgeskemaet i tabel 4 efter gennemgangstesten og gennemlæsning af de givne informationer og bedømmes ud fra tabel 3.

Mulig bedømmelse	Korrekt	Delvist korrekt	Ikke korrekt
------------------	---------	-----------------	--------------

Tabel 3: Bedømmelsesudfald for spørgeskema

Spørgsmål	Besvarelse	Bedømmelse
Hvad tror du det betyder, når en producent bliver afmeldt?	En TV-kanal bliver afmeldt	Delvist korrekt
Hvad tror du det betyder, når en producent bliver genstartet?	at en TV-kanal er blevet afmeldt og derefter valgt til igen	Korrekt
Hvad tror du en krediteringsprioriteringsniveau betyder?	afgør om man vil se mere af nogle skuespillere end andre - hvilket man kan gøre ved at give dem et højt prioriteringsniveau	Delvist korrekt

Tabel 4: Spørgsmål og bedømmelse af spørgeskema

TV 2 Krediteringsprogram

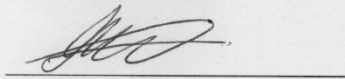
Erklæring

Testpersonen erklærer at have udført testen:

- uden hjælp
- uden forkendskab til programmet
- uden yderligere information end det, der er beskrevet i dokumentet

Testpersonen erkender endvidere at være enig i bedømmelsen for testpersonens gennemgang af programmet.

Testpersons underskrift:



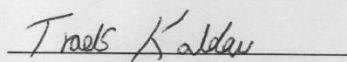
Dato:

24/5-21

Testafholderen erklærer ikke at have hjulpet testpersonen ud over informationerne givet i dokumentet.

Testafholderen erkender at have afgivet en fair bedømmelse af testpersonens gennemgang af programmet samt besvarelse af spørgeskemaet.

Testafholders underskrift:



Dato:

24/05/21

TV 2 Krediteringsprogram

N. Udklip af controller liste

- ▼ PresentationLayer
 - ▼ AdminControllers
 - AdminAddAdminController
 - AdminAddNewProducerController
 - AdminCreditListCellController
 - AdminEditAdminUsersController
 - AdminEditProducerController
 - AdminOverviewController
 - ▼ EditCreditListControllers
 - EditCreditCellController
 - EditCreditController
 - EditCreditListController
 - EditPersonCellController
 - EditPersonController
 - SetPersonController
 - > Enums
 - > Handlers
 - ▼ OtherControllers
 - ChangelogController
 - CreditCellUsers
 - EditProducerController
 - EditUserController
 - MainViewController
 - ▼ ProducerControllers
 - ProducerAddNewProducerUserController
 - ProducerCreditListCellController
 - ProducerEditProducerUserController
 - ProducerOverviewController
 - ▼ PublicControllers
 - FlowCellController
 - LoginController
 - PublicCreditListCellController
 - PublicOverviewController
 - ViewCreditCellController
 - ViewShowController

TV 2 Krediteringsprogram
