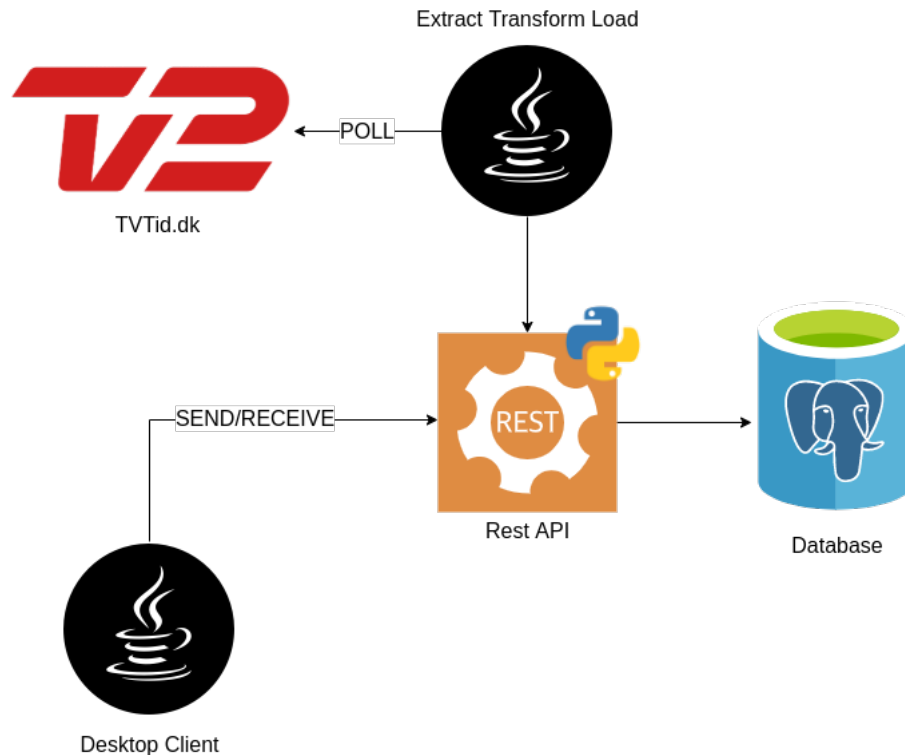


# Creditoro

Et krediteringssystem



Softwareteknologi

Semesterprojekt 2. semester, ST2-PRO

**Projektperiode:** 01.01.2020 - 29.05.2020

**Afleveringsdato:** 29.05.2020

## Projektgruppe 06:

Jakob Rasmussen, jakra19@student.sdu.dk

Kenneth M. Christiansen kechr19@student.sdu.dk

Kevin K. M. Petersen, kepet19@student.sdu.dk

Kristian N. Jakobsen, kjako19@student.sdu.dk

Mathias N. Rasmussen, mara816@student.sdu.dk

Simon Jørgensen, sijo819@student.sdu.dk

**Vejleder:** Henrik Lykkegaard Larsen, hlla@mmmi.sdu.dk

Syddansk Universitet  
Det Tekniske Fakultet  
Mærks Mc-Kinney Møller Instituttet  
Campusvej 55, 5230 Odense M

**Title:** Creditoro

**Institution:** Syddansk Universitet  
Det Tekniske Fakultet, Mærsk Mc-Kinney Møller Institut  
Campusvej 55, 5230 Odense M

**Uddannelse:** Softwareteknologi

**Semester:** 2. Semester

**Semestertema:** Udvikling af cyber-physical softwaresystemer

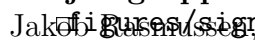

**Kursuskode:** ST2-PRO

**Projektperiode:** 01.01.2020 - 29.05.2020

**ECTS:** 10 ECTS

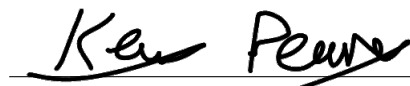
**Vejleder:** Henrik Lykkegaard Larsen

**Projektgruppe:** 06

   
Jakob Rasmussen, jakr19@student.sdu.dk



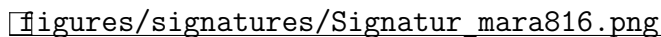
Kenneth M. Christiansen, kechr19@student.sdu.dk



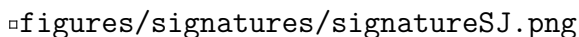
Kevin K. M. Petersen, kepet19@student.sdu.dk



Kristian N. Jakobsen, kjako19@student.sdu.dk



Mathias N. Rasmussen, mara816@student.sdu.dk



Simon Jørgensen, sijo819@student.sdu.dk

Antal sider: 26 Sider

Bilag: 1 bilag

Ved at underskrive dette dokument bekræfter hvert enkelt gruppemedlem, at alle har deltaget lige i projektarbejdet, og at alle således hæfter kollektivt for rapportens indhold.

# Indhold

<b>1</b>	<b>Indledning</b>	<b>1</b>
1.1	Projektrammer . . . . .	1
1.1.1	Krav til Projektet . . . . .	1
1.1.2	Tidsplan . . . . .	1
1.2	Formål med Inceptionsfasen . . . . .	3
1.3	Problemanalyse . . . . .	3
1.3.1	Igangsættende Problem . . . . .	3
1.3.2	Identifikation af Problemet . . . . .	4
1.4	Problemformulering & Afgrænsning . . . . .	4
<b>2</b>	<b>Faglig Vidensgrundlag</b>	<b>5</b>
2.1	Begrebsdefinitioner . . . . .	5
2.2	Fagligt Vidensgrundlag . . . . .	5
2.2.1	JAVA . . . . .	5
2.2.2	Python . . . . .	5
2.2.3	Database . . . . .	6
2.2.4	Ubuntu & Docker . . . . .	6
2.3	Relevante Eksisterende Løsninger . . . . .	6
<b>3</b>	<b>Overordnet Kravspecifikation</b>	<b>7</b>
3.1	Aktørliste . . . . .	8
3.2	Overordnet Brugsmodel . . . . .	9
3.3	Liste over Brugsmodeller . . . . .	10
3.4	Overordnet Supplerende Krav . . . . .	11
3.5	Detaljerede Beskrivelse af Udvalgte Essentielle Brugsmodeller . . . . .	11
<b>4</b>	<b>Kritiske Risici</b>	<b>13</b>
<b>5</b>	<b>Prioritering</b>	<b>14</b>
5.1	Forretningsmæssige Betydning . . . . .	14
5.2	Arkitektonisk Betydning . . . . .	15
5.3	Læringsmæssige Udbytte . . . . .	15
5.4	X-faktor . . . . .	15
<b>6</b>	<b>Metoder &amp; Værktøjer</b>	<b>16</b>
6.1	Metoder i Inceptionsfasen . . . . .	16
6.1.1	Brugsmodeldiagram . . . . .	16
6.1.2	FURPS . . . . .	16
6.1.3	MoSCoW . . . . .	16
6.2	Metoder i Elaborationsfasen . . . . .	17
6.2.1	UP & Scrum . . . . .	17
6.2.2	Værktøjer . . . . .	18
<b>7</b>	<b>Ressourcer</b>	<b>19</b>
<b>8</b>	<b>Konklusion</b>	<b>20</b>

<b>9</b>	<b>Procesdokumenter</b>	<b>21</b>
9.1	Gruppekontrakt . . . . .	21
9.2	Vejlederkontrakt . . . . .	24
9.3	Belbin Gruppeprofil . . . . .	25
<b>A</b>	<b>Bilag</b>	<b>26</b>
A.1	Logbog . . . . .	26

# 1 Indledning

Når et program bliver broadcastet på en TV station, skal krediteringer vises. Dette gøres i slutningen af programmet, i maksimalt 30 sekunder. Det betyder, at der ikke altid er tid til at vise alle krediteringer, og derfor prioriteres de før de vises.

Hvis de 30 sekunder for hvert program kan frigøres, kan danske TV stationer bruge tiden på at vise noget andet, som f.eks. reklamer. Derved kan TV 2 øge deres årlige indtægter med op til 60 millioner kroner.

TV 2 har brug for et system, der kan administrere krediteringer for programmer produceret i Danmark. Hertil skal der kunne tilføjes nye krediteringer i systemet for nye produktioner, samt det skal være muligt at kunne søge efter eksisterende krediteringer. Det skal være muligt at kunne se hvilken rolle en given person har haft i en produktion, da denne person kan have haft flere forskellige roller på flere forskellige produktioner.

## 1.1 Projektrammer

Denne sektion har til formål at opridse rammerne for projektet, samt hvilket område projektgruppen arbejder indenfor.

### 1.1.1 Krav til Projektet

Systemet skal så vidt muligt skrives i programmeringssproget Java.

Krediterings-data skal lagres i en database, og i dette projekt skal den brugte database være SQL baseret. Der skal bruges PostgreSQL.

Systemet forventes ikke at være et færdigt system, men en række forslag til løsninger der opfylder systembehovet. Forslagene skal inkludere:

- Krav
- Analyse
- Design
- Implementering
- Test

Producere der kan tilføje og redigere i krediteringerne, skal kun have mulighed for at redigere i de produktioner, de selv ejer.

Det forventes at krediteringssystemet er kompatibelt med andre systemer (f.eks. fra Stofa, YouSee etc.).

### 1.1.2 Tidsplan

Tidsplanen har til formål at skabe overblik og styring over projektet. Den giver gruppemedlemmerne et overblik over, hvornår de forskellige dele af projektet skal starte og slutte, og derved bliver det hurtigt klart hvis tidsplanen skrider.

figures/grantt\_udvidet.png

**Figur 1:** Tidsplan for projektet

## 1.2 Formål med Inceptionsfasen

Formålet med inceptionsfasen er at fastlægge systemets omfang, der bliver udformet en overordnet kravspecifikation, kravene prioriteres og metoderne i elaborationsfasen beskrives. Dette sker gennem en nærmere undersøgelse af problemstillingen, indsamling af information og under kundemøder hvor kravene indsamles (eliciteres).

Målene for inceptionsfasen kan således opstilles i punktform:

- At gennemføre kravudvikling
- At identificere kritiske risici
- At fastlægge projektets metoder i elaborationsfasen

## 1.3 Problemanalyse

### 1.3.1 Igangsættende Problem

TV2 ønsker at frigøre 30 sekunders krediterings tekster efter hvert program, så de i stedet kan bruge tiden på at vise reklamer. Problemet består i at disse krediterings tekster, så skal vises på en anden platform.

I tabel 1 ses kravene fra TV2's projektcase:

Beskrivelse	Type
"Vi har brug for et krediterings system der kan håndtere dansk TV content"	En vag opgave
"Dette inkluderer muligheden for at oprette nye krediteringer i systemet, når en ny produktion bliver lavet, samt at have mulighed for at søge efter en given produktion og få en liste af krediteringer, forbundet til denne. Det burde også være muligt at se hvilken rolle en given person har haft i en produktion, eftersom en person kan have flere forskellige roller i forskellige produktioner."	Ønske om en bestemt løsning
"Producers/TV-stationer burde være i stand til at redigere krediteringer for programmer/produktioner de ejer. De burde også være i stand til at redigere disse produktioners ID. Systemadministratorer skal kunne vedligeholde (oprette, læse, opdatere og slette) personer, krediteringer og personer."	Ønske om en bestemt løsning
"Til slut skal systemet kunne offentliggøre en service som andre systemer kan bruge. Disse systemer kan f.eks. være en hjemmeside eller en applikation. Disse andre systemer skal også kunne bruge API'et, så data'et kan blive brugt i allerede eksisterende systemer (såsom TVTID.dk - TV 2's TV-Guide)."	Ønske om en bestemt løsning
"En form for adgangskontrol skal implementeres, til de beskyttede dele af systemet (oprettelse, opdatering, slettelse, osv. af data)	Ønske om en bestemt løsning
"Der skal være en offentligt tilgængelig del af systemet, hvor det er muligt at se krediteringer uden at logge ind."	Ønske om en bestemt løsning
"Nuværende løsning er begrænset til 30 sekunder, og dermed kan alle krediteringerne ikke altid vises i praksis"	Et problem

**Tabel 1:** Krav fra TV2s projektcase

### 1.3.2 Identifikation af Problemet

Som det er nu bliver krediteringer vist i slutningen af et program. Ifølge reglerne for visning af krediteringer, må krediteringer ikke vises mere end 20 sekunder for produktioner under 60 minutter, og 30 sekunder for produktioner over 60 minutter. Dette giver en del problemer. For det første betyder den begrænsede varighed, at ikke alle medarbejdere kan krediteres. Dette ender ud i at der skal prioriteres i krediteringerne, før de bliver vist på TV. Derved får alle medarbejdere ikke den anerkendelse de burde. Hvis krediteringer flyttes til et eksternt system, og derved ikke bliver vist på TV, kan man undgå at skulle prioritere. Alle kan derved få den fortjente kredit. Derudover vil det også give mulighed for at vise noget andet, som f.eks. reklamer eller promoveringer for andre programmer (red: eget indhold).<sup>1</sup>

Et sådan eksternt system vil også hjælpe med oprettelsen af nye krediteringer, ved at gøre processen hurtigere og nemmere, samt mere overskueligt. Dertil har gruppen valgt at arbejde med samtlige/alle problemstillinger givet af TV2 i tabel 1, og lave en prototype til et funktionsdygtigt system. Angående valget med at arbejde med samtlige problemstillinger præsenteret af TV2, har gruppen konkluderet det som værende realistisk jævnfør figur 1. Denne prototype vil kunne bruges som et udkast til et endeligt system.

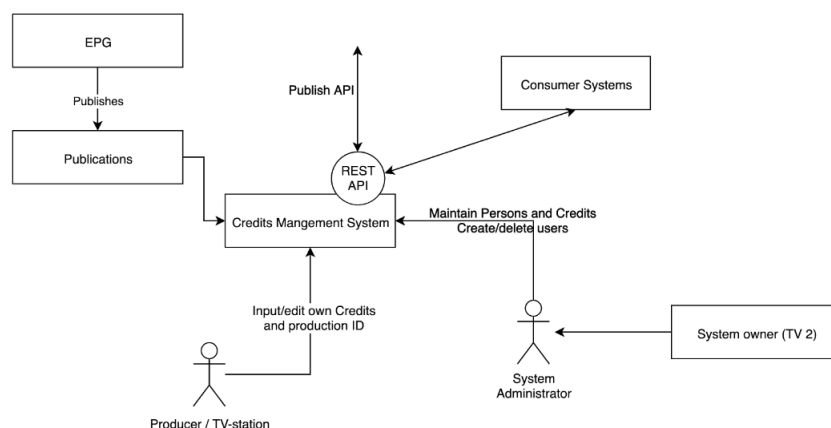
## 1.4 Problemformulering & Afgrænsning

*Hvordan kan vi udvikle et samlet krediteringssystem, der giver mulighed for at erstatte rulletekster efter et endt program?*

1. Hvem skal kunne håndtere krediteringer?
2. Hvordan skal krediteringerne gøres tilgængelige, og hvordan skal seerne refereres dertil?
3. Hvordan kan man oprette et system som kan indeholde krediteringer?

Projektgruppen har valgt at afgrænse dette projekt, ved at konstruere en prototype til et system.

Projektgruppen har valgt at lave et system der ligger tæt op ad det oprindelige foreslag fra TV2s projektcase. Dette indebærer alle kravene i tabel 1.



**Figur 2:** Foreslag til systemtegning - © TV2

I et produktionsklart system vil det være ideelt at have en webside, men dette har vi konkluderet som værende uden for projektet.

<sup>1</sup>DR's krediteringsregler for TV



## 2 Faglig Vidensgrundlag

### 2.1 Begrebsdefinitioner

De begreber der findes i denne rapport er defineret i tabel 2:

Begreb & Definition	
REST API	REST: Repræsentativ tilstandsoverførsel (dansk oversættelse af REST) er en software-arkitektonisk stil, der definerer et sæt begrænsninger, der skal bruges til at oprettelse af webtjenester API: API står for Application Programming Interface, og er en software-grænseflade, der tillader et stykke software at interagere med andet software
EPG	EPG er en forkortelse for Electronic Programme Guide. Det er en generelt betegnelse for elektronisk programoversigt over TV-programmer
GDPR	Databeskyttelsesforordning, der har til formål at styrke og harmonisere beskyttelsen af personoplysninger i EU
SQL	Structured Query Language er et programmeringssprog til relationelle databaser
Person	Person er de medarbejdere der skal krediteres. Det kunne f.eks. være en skuespiller eller en lydmand.
Personinformation	Personlig information som fx e-mail, tlf. nr. osv.
Swagger UI	Brugergrænseflade til vores REST Api. ( <a href="https://swagger.io/">https://swagger.io/</a> )
Docker	Bruges til containeriseringen, så host styresystemet bliver agnostisk, og softwaren bliver indkapslet. Det forhindrer også "it works on my machine"problemer.

**Tabel 2:** Begrebsdefinitioner

### 2.2 Fagligt Vidensgrundlag

Dette afsnit har til formål at dækker over den faglige viden gruppen skal have, for at kunne udføre projektet.

#### 2.2.1 JAVA

At have kendskab til Java er en vigtig forudsætning for udarbejdelsen af projektet. Systemet vil hovedsageligt blive programmeret i sproget Java. Det faglige niveau svarer til 2. semesterstuderende på Softwareteknologi. Dette indebærer blandt andet forståelse af JavaFX og Scenebuilder. Arbejdet med Java i projektet forudsætter derudover forståelse for basale programmeringsprincipper og forståelse for det objektorienterede programmeringsparadigme.

#### 2.2.2 Python

Et grundlæggende kendskab til Python kræves for at kunne forstå - samt implementere REST Api'et.

### 2.2.3 Database

Systemet vil indeholde en lang række data, som skal lagres i en database. Det er derfor nødvendigt at have forståelse for databaser, databasestrukturer, relationelle SQL-databaser og SQL-queries. Databasen der vil blive benyttet i projektet er PostgreSQL, en basal viden om databasesproget/programmeringssproget SQL er derfor nødvendig. Al nødvendig viden er givet i SDU's 'Data Management' kursets pensum.

### 2.2.4 Ubuntu & Docker

En basal forståelse for Ubuntu (eller andet Linux baseret distro) og Docker kræves for opsætning af REST Api'et og databasen.

## 2.3 Relevante Eksisterende Løsninger

### IMDb

IMDb (Internet Movie Database) er en online database bestående af film, serier, medvirkende m.m. Man har mulighed for at søge efter informationer ved at referere til blandt andet førnævnte titler. IMDb har også et ratingsystem, der gør det muligt at bedømme film, serier etc. **Rotten Tomatoes**

Rotten Tomatoes er på lige fod med IMDb, en database for film, serier, medvirkende m.m. Man kan på Rotten Tomatoes også søge information. Rotten Tomatoes distancerer sig fra IMDb, ved både at tage ratings fra sine brugere og et panel af anmeldere.

### 3 Overordnet Kravspecifikation

Systemet afspejler det system TV2 har lagt op til i projektcasen. Der er tale om et system, hvor alle kan se - og nogle kan redigere krediteringer for programmer. Systemet skal kunne tilgås via en dansk brugergrænseflade. Det skal indeholde forskellige brugerroller; systemadministrator, kanaladministrator, producer, royalty bruger og gæst.

Systemadministratoren skal have rettigheder til at gøre alt. Dette gælder f.eks. at oprette krediteringer, kanaladministratorer og producere. Kanaladministratoren skal kunne redigere, oprette og slette krediteringer for egen kanal. En producer skal kunne tilføje og redigere i krediteringerne for egne produktioner, og en gæst skal kunne se krediteringer for alle programmer. Det skal være muligt at kombinere personer som refererer til den samme person i den virkelige verden. Når to forskellige producere vil oprette en kreditering for et program, skal krediteringen være associeret med en person og vedkommendes rolle. Det betyder altså, at det skal være muligt at oprette personer der kan sammenflettes (f.eks. med UUID).

TV2 har ikke lov til at lagre persondata, såsom et CPR-nummer eller et telefonnummer, så det skal være muligt at identificere personer i systemet og sikre at krediteringerne er korrekt forbundet til de rigtige personer. Det skal være muligt at eksportere en specifik mængde data i forskellige formater såsom XML og CSV. Derudover skal databasen være søgbar, så det er nemt at finde personer, programmer og lignende. Det er vigtigt at systemet er nemt at bruge, så seerne nemt kan se krediteringerne for det program de lige har set.

TV 2 kunne være interesseret i at integrere systemet med andre systemer (YouSee Tv, Boxer Play osv.), og det er derfor vigtigt at systemet er kompatibelt med krediteringer i andre systemer. Det kunne også være interessant at have muligheden for at få notifikationer når noget nyt sker i systemet. Samrådet for Ophavsret og Producentforeningen kunne også være interesseret i at modtage en form for meddelelse hver gang der er blevet tilføjet noget nyt til systemet, hvor de kan godkende krediteringerne og ud fra disse udbetale royalties. Derudover kunne det også være interessant at brugergrænsefladen kunne understøtte flere sprog.

For at beskytte dele af systemet (tilføjelse/redigering/sletning af data osv.), skal der indføres en form for adgangskontrol. Der skal være en offentligt tilgængelig del af systemet, hvor det er muligt at se krediteringerne for et et program uden at skulle logge ind.

I tabel 3 ses kravene opsummeret i en tabel for bedre overblik.

ID	Navn	Beskrivelse
K01	Brugergrænseflade	Systemet skal tilgås via en dansk brugergrænseflade
K02	Brugerroller	Systemet skal indeholde brugerroller
K03	Tildel roller	Kanaladministrator skal kunne tildele producer- og kanaladministrator roller
K04	Slet bruger	Systemadministratoren skal kunne slette brugere
K05	Se krediteringer	Alle skal kunne se krediteringer
K06	Søg efter krediteringer	Alle skal kunne søge efter og se krediteringer for alle programmer
K07	Opret krediteringer	Specielle brugere, kanaladministratore og systemadmin skal kunne oprette krediteringer for et givent program

K08	Rediger krediteringer	Specielle brugere, kanaladmin og systemadmin skal kunne redigere krediteringer for egne programmer
K09	Slet kreditering	Kanaladmin og systemadmin skal kunne oprette/redigere/slette krediteringer under egen kanal
K10	Søg efter personer	Alle skal kunne søge efter personer
K11	Knyt personer til krediteringer	Personer skal kunne knyttes til krediteringer så man kan se hvilke programmer en person har deltaget i Systemadmin, kanaladmin og producer skal kunne se persondata som email og tlf. nr.
K12	Link personer i den virkelige verden	Det skal være muligt at linke personer i krediteringer til personer i den virkelige verden, så der krediteres korrekt
K13	Eksporter data	Brugere skal kunne eksportere data til forskellige formater såsom XML og CSV
K14	Importering af data	Systemet skal kunne importere EPG data via TVTid.dk
K15	Integration	Systemet skal kunne integreres med andre systemer (Yoursee Play, Boxer Play, osv.)
K16	Notifikationer	Systemet skal sende notifikationer til relevante brugere
K17	Sprogvalg	Systemet skal kunne understøtte flere sprog

**Tabel 3:** Liste af krav fra overordnet kravspecifikation

### 3.1 Aktørliste

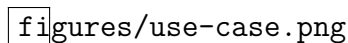
Aktørlisten er sat op sådan, at overaktøren har samme rettigheder som underaktøren, samt ekstra rettigheder. F.eks. har systemadministratoren samme rettigheder som kanaladministratoren, plus rettigheder til at oprette kanaler, slette personer osv.

Aktør	Beskrivelse	Mål & tjenester
Systemadministrator (p)	Systemadministratoren har samme rettigheder som alle andre aktører. Denne aktør står primært for at slette personer, samt at oprette og slette kanaler. Til disse kanaler skal systemadministratoren tildele kanaladministratorroller.	<ul style="list-style-type: none"> <li>- Slette personer</li> <li>- Oprette og slette kanaler</li> <li>- Tildele og fjerne kanaladministratorroller</li> </ul>
Kanaladministrator (p)	Kanaladministratoren skal godkende nyoprettede krediteringer, og har rettigheder til at slette krediteringer under egen kanal. Kanaladministratoren står f.eks. til at tildele producer- og andre kanaladministratorroller for egen kanal.	<ul style="list-style-type: none"> <li>- Godkende nyoprettede krediteringer</li> <li>- Slette krediteringer under egen kanal</li> <li>- Tildele producer- og kanaladministratorroller for egen kanal</li> </ul>
Royalty bruger (p)	Royalty bruger repræsenterer firmaer, som Registrering Danmark, der sørger for at betale royalties til medarbejdere på produktioner.	<ul style="list-style-type: none"> <li>- Kan eksportere data</li> </ul>

Producer (p)	Produceren for et program står for at oprette nye krediteringer, samt rette krediteringer for egne produktioner.	<ul style="list-style-type: none"> <li>- Kan oprette krediteringer</li> <li>- Kan redigere i eksisterende krediteringer for egne produktioner</li> <li>- Kan logge ind og af</li> <li>- Kan se personinformation</li> </ul>
Gæst (p)	Gæsten repræsenterer offentligheden, som skal kunne søge efter informationer og krediteringer om forskellige programmer samt personer. Gæsten kan se alle produktioner en person har været med i. Gæsten kan også ændre sproget for brugergrænsefladen.	<ul style="list-style-type: none"> <li>- Kan søge efter og se krediteringer og personer for forskellige programmer</li> <li>- Se alt hvad en person har været med i</li> <li>- Ændre sproget</li> </ul>

**Tabel 4:** Aktørlisten

## 3.2 Overordnet Brugsmønstermodel

figures/use-case.png

**Figur 3:** Overordnet brugsmønster over Creditoro systemet

Menneskene er aktører.

Cirklerne beskriver handlinger aktørerne kan lave.

Pilende betyder Extends, hvilket vil sige aktørerne arver funktionalitet

### 3.3 Liste over Brugsmønstre

ID	Navn	Aktør
B01	Opdater egen royalty bruger	Systemadministrator (p), Royalty Bruger (p)
B02	Opdater person	Systemadministrator (p), kanaladministrator (p), producer (p)
B03	Opret Producer	Systemadministrator (p), kanaladministrator (p)
B04	Opret person	Systemadministrator (p), kanaladministrator (p), Producer (p)
B05	Fjern producer	Systemadministrator (p)
B06	Fjern person	Systemadministrator (p)
B07	Opret kreditering	Systemadministrator (p), Kanaladministrator (p), producer (p)
B08	Fjern kreditering til program	Systemadministrator (p), kanaladministrator (p)
B09	Opdater egne kreditering	Systemadministrator (p), kanaladministrator (p), Producer (p)
B10	Læse krediteringer	Systemadministrator (p), kanaladministrator (p), Royalty Bruger (p), Producer (p), Gæst (p)
B11	Se person profil	Systemadministrator (p), kanaladministrator (p), Royalty bruger (p), Producer (p), Gæst (p)
B12	Søge efter personer og programmer	Systemadministrator (p), kanaladministrator (p), Royalty Bruger (p), Producer (p), Gæst (p)
B13	Log ind	Systemadministrator (p), kanaladministrator (p), Royalty Bruger (p), Producer (p)
B14	Log af	Systemadministrator (p), kanaladministrator (p), Royalty Bruger(p), Producer (p)
B15	Kan se personinformation	Systemadministrator (p), kanaladministrator (p), Royalty Bruger (p), Producer (p)
B16	Godkende nye krediteringer	Systemadministrator (p), Kanaladministrator (p)
B17	Afvis nye krediteringer	Systemadministrator (p), Kanaladministrator (p)
B18	Opret kanaladministrator	Systemadministrator (p), kanaladministrator (p)
B19	Opret systemadministrator	Systemadministrator (p)
B20	Ændre sprog	Systemadministrator (p), kanaladministrator (p), Royalty Bruger (p), Producer (p), Gæst (p)

**Tabel 5:** Brugsmønstre

### 3.4 Overordnet Supplerende Krav

Her bruges FURPS for supplerende krav.

FURPS	Krav
Functionality	Skal kunne kreditere produktionsroller som er angivet af DRs Krediteringsregler Skal overholde GDPR
Usability	Systemet skal kunne understøtte flere sprog
Reliability	Hvis serveren til systemet genstarter, startes del-systemerne igen automatisk. Der vil ikke være behov for at ligge systemet ned regelmæssigt for at kunne foretage backup.
Performance	Databasen skal kunne håndtere 10000 nye brugere - samt 15000 krediteringer årligt i 25 år, uden at ofte brugte kald til REST Api'et bliver sløvt (reponsetid på mere end 300 ms)
Supportability	Systemet vil indeholde unit tests, og komme med en rapport over hvor stor en procendel der er dækket af dette. Systemet vil blive forbundet til det centraliserede fejllognings system <b>Sentry</b> . Systemet er installerbart vha. Docker via Docker-compose. Det vil være muligt at konfigurere system indstillinger via en <code>.env</code> (miljø) fil. En opsætningsguide vil være at finde sammen med kildekoden.

**Tabel 6:** FURPS

### 3.5 Detaljerede Beskrivelse af Udvalgte Essentielle Brugsmønstre

Ud fra MoSCoW har vi valgt 2 brugsmønstre vi mener er essentielle for systemet. De to brugsmønstre er 'Læs kreditering' og 'Opret kreditering'. Til disse to brugsmønstre har vi lavet detaljerede brugsmønstre som kan ses i tabel 7 og tabel 8.

<b>Brugsmønster:</b> Læs kreditering
<b>ID:</b> UC10
<b>Primære aktører:</b> Systemadministrator, kanaladministrator, producer, royalty bruger, gæst
<b>Sekundære aktører:</b>
<b>Kort beskrivelse:</b> Alle skal kunne se krediteringen for programmerne.
<b>Prækonditioner (Pre conditions):</b>
<b>Hovedhændelsesforløb (main flow):</b> <ol style="list-style-type: none"><li>1. Brugsmønstret starter når en aktør vil se kredit for et program</li><li>2. Aktøren søger efter programmet</li><li>3. Aktøren trykker på det ønskede program</li><li>4. Systemet checker hvilken rolle aktøren har</li><li>5. Aktøren bliver omdirigeret til den passende visning af krediteringen</li></ol>
<b>Postkonditioner (post conditions):</b> En kreditering bliver vist
<b>Alternative hændelsesforløb (alternative flow):</b> Step 2: Hvis programmet ikke findes, får vedkommende besked om at programmet ikke findes

**Tabel 7:** Brugsmønster: Læs kreditering

<b>Brugsmønster:</b> Opret kreditering
<b>ID:</b> UC07
<b>Primære aktører:</b> Systemadministrator, kanaladministrator, Producer
<b>Sekundære aktører:</b>
<b>Kort beskrivelse:</b> Producenten opretter en kreditering. Heri angives alle der har bidraget til produktionen af TV-programmet, filmen el. lign.
<b>Prækonditioner (Pre conditions):</b> Aktøren skal være logget på systemet
<b>Hovedhændelsesforløb (main flow):</b> <ol style="list-style-type: none"> <li>1. Brugsmønstret starter når en administrator eller producer vil oprette en kreditering</li> <li>2. Aktøren trykker på knappen 'Opret Kreditering'</li> <li>3. Systemet checker aktørens rolle</li> <li>4. Aktøren er forbundet til en kanal, og angiver programmets titel</li> <li>5. Systemet checker om der allerede findes et program med den angivne titel</li> <li>6. Aktøren krediterer alle der har medvirket i produktionen af programmet</li> <li>7. Aktøren sender den færdige kreditering videre til godkendelse</li> <li>8. Postkonditioner (post conditions):</li> <li>9. En kreditering er blevet oprettet</li> </ol>
<b>Alternative håndelsesforløb (alternative flow):</b> Step *: Aktøren kan til enhver tid afbryde oprettelsen af krediteringen Step 4: Hvis aktøren er systemadministrator, er vedkommende ikke forbundet til en kanal, og kan skifte hvilken kanal krediteringen skal oprettes ved. Step 5: Hvis programmets titel allerede eksistere, gøres aktøren opmærksom på dette. Step 8: Hvis krediteringen afvises, laves de fornødne ændringer, og den nye kreditering sendes videre til godkendelse.

**Tabel 8:** Brugsmønster: Opret kreditering



## 4 Kritiske Risici

Nedenfor ses en tabel med de fem identificerede risici. 'Risikovurdering' vurderer hvor høj sandsynligheden er for at de identificerede risici indtræffer. 'Indvirkning' vurderer hvor meget det vil påvirke projektet, hvis risikoen bliver en realitet. 'Håndtering' sætter en mulig handlingsplan, såfremt risikoen bliver aktuel. 'Kritisk?' vurderer om risikoens indtræf er kritisk for projektet.

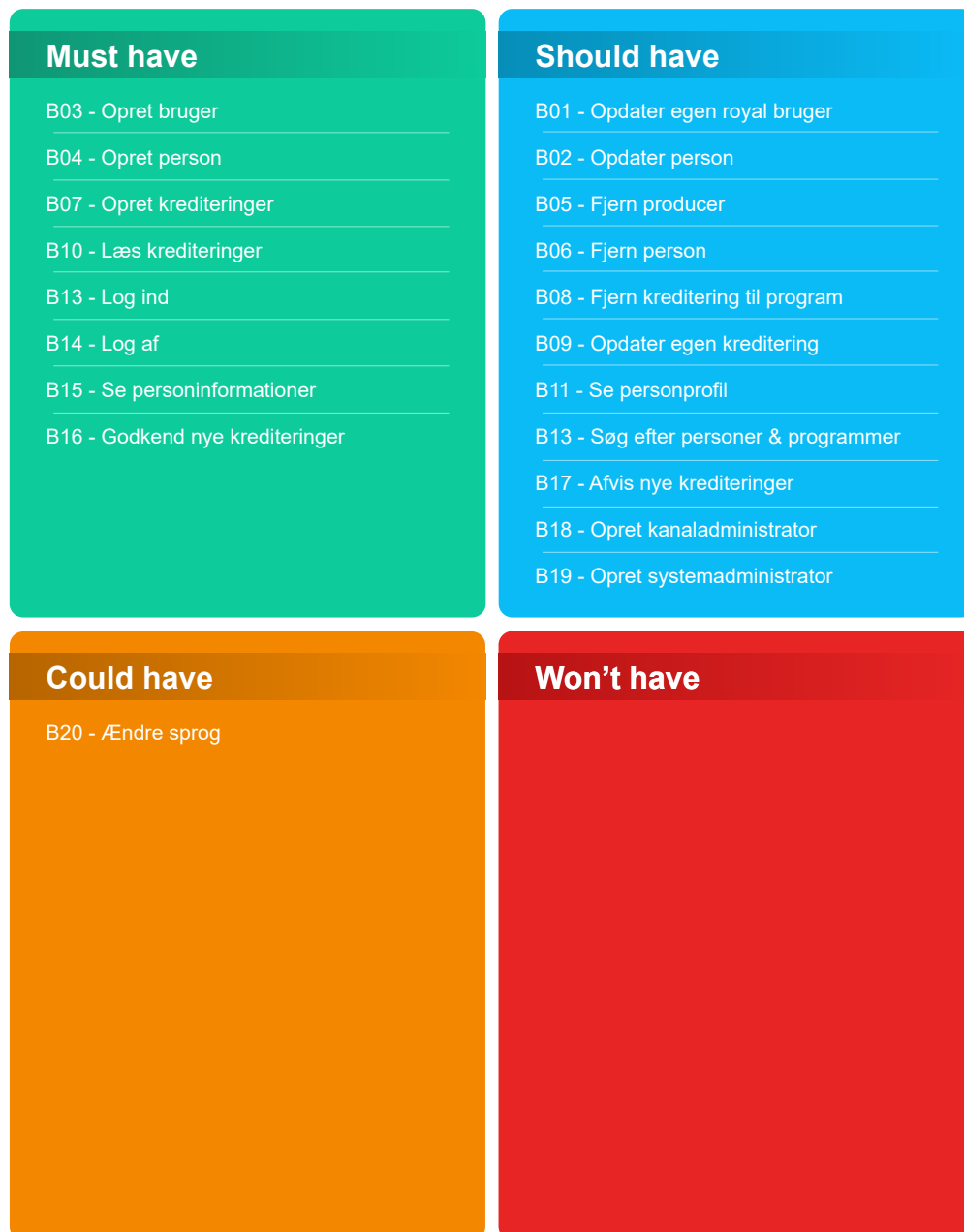
Risiko	Risikovurd.	Indvirk.	Håndtering	Kritisk?
COVID-19	Høj	Høj	I tiden under hjemmekarantænen, vil al skolerelateret arbejde, inkl. undervisning, foregå online. For at undgå komplikationer, stilles der højere forventninger til at udleveret læsning fra undervisningen vil blive nærstuderet, samt at al undervisning følges så godt som muligt. Vejleder står til rådighed på mail såfremt presserende spørgsmål skulle opstå	Ja
REST API	Middel-høj	Høj	Der vil fra starten af projektets construction fase af være ekstra fokus på, om det er realistisk at nå at lave REST-API'et. Øjeblikket gruppen vurderer det ikke er muligt, vil REST-API'et blive droppet, og gruppens SCRUM board vil i fællesskab blive opdateret.	Ja
Tidsplan overholdes ikke	Middel-høj	Høj	Overholdes tidsplanen ikke, vil der blive kigget på de aktuelle arbejdsopgaver, og alt efter om arbejdsbyrden er for stor eller lille, vil der blive tilpasset derefter.	Ja
Systemet møder ikke kravene	Lav	Høj	Hvad kunne gøres anderledes? Hvad kan der gøres fremadrettet for at opnå et mere tilfredsstillende resultat. Gruppen perspektiverer samlet	Ja
Mistet medlem	Lav	Middel-høj	Et mistet medlem vil lede til manglende arbejdskraft. Arbejdsbyrden/arbejdsopgaverne vil derfor blive opdateret, til de nye gruppeforhold.	Ja

**Tabel 9:** Kritiske risici

## 5 Prioritering

### 5.1 Forretningsmæssige Betydning

Til prioriteringen af brugsmønstrene er MoSCoW metoden benyttet. Resultatet kan ses nedenfor på figur 4.



**Figur 4:** MoSCoW

Her ses de forskellige brugsmønstre i en MoSCoW. MoSCoW bruges til at finde ud af hvad der skal udvikles først. Da der ikke blev fundet ting i Inceptionsfasen, som der ikke skulle med denne gang, kom der ikke noget i "Won't have" sektionen.

## 5.2 Arkitektonisk Betydning

De arkitektoniske krav der har en arkitektonisk betydning for systemet er:

- Brugergrænsefladen til klienterne vil være på Dansk.
- Brugergrænsefladen(Swagger UI) til REST Api'et vil være på Engelsk.
- Persistens vil blive håndteret af en database
- Databasen vil være PostgreSQL
- Systemet vil automatisk starte efter evt. server genstart.

## 5.3 Læringsmæssige Udbytte

- Supplerende krav - REST Api
  - Vi har lagt op til et system der vil afspejle den løsning som TV2 oprindeligt lagde op til. Den løsning indeholder et REST Api, som er vurderet som værende udenfor det der forventes at kunne som studerende på nuværende tidspunkt. Ved at vælge sådan en løsning, vil det give os et rigtigt godt redskab i vores programmerings-værktøjskasse i fremtidige projekter.

## 5.4 X-faktor

- Der mangler reference til funktionkrav 14 - doobelt klip på mig og kig ned
- K14 - Importering af data
  - Integrationen mellem andre systemer, og gå de forskellige systemer til at spille sammen er spændende. Det er her med til at give systemet et løft, da det automatiserer oprettelsen af nye programmer i systemet.

## 6 Metoder & Værktøjer

### 6.1 Metoder i Inceptionsfacen

#### 6.1.1 Brugsmønsterdiagram

Det første skridt i udviklingen af et brugsmønster er at finde og definere de forskellige aktører der vil interagere med systemet. En aktør kan defineres som alt der kommunikerer med systemet og ikke selv er en del af systemet. Et eksempel på dette kunne være en kunde på en webshop. Disse aktører opstilles i en tabel sammen med de brugsmønstre hver aktør kan tilgå.

Da kravindsamling er en evolutionær aktivitet, bliver alle aktører ikke nødvendigvis identificeret i første iteration. Det er muligt at identificere primære aktører i løbet af første iteration, og først senere i forløbet blive i stand til at identificere sekundære aktører, når man får mere viden om systemet. *Primære* aktører interagerer med systemet for at opnå påkrævede systemfunktioner, og ud fra det, få noget ud af at bruge systemet. *Sekundære* aktører støtter systemet så de primære aktører kan gøre deres arbejde. Når aktørerne er fundet kan brugsmønstrene findes. Et brugsmønster angiver et scenarie en aktør kan interagere med.

Når både aktører og brugsmønstre er fundet, kan man opstille et brugsmønster-diagram for at give en visuel forståelse for hvilke aktører der kan tilgå hvilke brugsmønstre. Et eksempel på et brugsmønsterdiagram kan ses på figur 3

Brugsmønstermodellen hjælper udvikleren med at forstå brugeren, så systemet kan konstrueres. Idet der er et samlet billede af hvordan systemet skal se ud, vil udviklingen være mere målfast da kravene og deres forhold til brugeren er klare.

#### 6.1.2 FURPS

FURPS er en model til klassificering af ikke-funktionelle krav, og er med til at give en detaljeret beskrivelse af kravene. Akronymet står for:

**Functionality:** Hvad kunden vil have. Dette inkluderer også sikkerhedsforanstaltninger.

**Usability:** Hvor effektivt er produktet fra brugerens synspunkt? Er produktet æstetisk acceptabelt? Er dokumentationen fyldestgørende?

**Reliability:** Hvad er den mest acceptable system nedetid? Er systemfejl forudsigelige? Er det muligt at demonstrere hvor præcise resultaterne er? Hvordan bliver systemet gendannet?

**Performance:** Hvor hurtigt skal systemet være? Hvad er den maksimale responstid? Hvad er gennemløbet? Hvor meget hukommelse bruger systemet?

**Supportability:** Kan systemet testes? Er det muligt at konfigurere systemet, udvide det, installere det, og yde service på systemet.

#### 6.1.3 MoSCoW

MoSCoW er en prioriterings model. Den bruges ofte i software udvikling.

Modellen i sig selv kan dog bruges, men det anbefales at man bruger den sammen med en **Agile proces**.

MoSCoW er en vigtig model i software udvikling da, den beskriver hvilken del af softwaren der minimum skal laves før det virker. Der laves en prioritering liste med kunden om hvad de så gerne vil have først. Det bliver så stillet op i en MoSCoW model.

**Must have** betyder skal have og i software udvikling betyder det, som er minimum der skal være med for at softwaren virker.

**Should have** betyder det som burde være med det kunden rigtig gerne vil have med.

**Could have** betyder det som kunne være med. Hvis der er tid nok.

**Won't have (this time)** Det som der slet ikke skal prioriteres nu, men måske en anden gang.

## 6.2 Metoder i Elaborationsfasen

### 6.2.1 UP & Scrum

#### UP

'UP' er en forkortelse af 'unified process' UP består af fire faser. De fire faser er beskrevet i den rækkefølge som de eksekveres

#### Inceptionsfasen

Inceptionsfasen er der for at finde ud af om projekt overhovedet kan gennemføres, Bestemme hvilket anvendelseområde systemet har. Identificere vigtige krav og kritiske risici.

#### Elaborationfasen

I Elaborationfasen er for at lave en iterativ udvikling af de forskellige krav, design, analyse og test ud fra den overordnede kravspecifikation og den prioritering der er lavet. I Elaborationfasen vil der blive brugt Scrum.

#### Konstruktionfasen

I Konstruktionfasen vil fokus være på udvikling af komponenter og andre funktioner. I fasen vil der blive brugt UML til at identificere, hvilke klasser og komponenter der skal være. Det er i denne fase kodingen kommer til at ske og den første iteration af software produktet.

#### Overgangfasen

I Overgangfasen vil der være fokus på at få et færdigt software produkt. Det vil man gøre ved at, se om man har implementeret det er aftalt og hører ens bruger om de er tilfredse med produktet

#### Scrum

Scrum vil blive brugt i Elaborationsfasen, til at nedbryde de krav vi har defineret i inceptionsfasen (se krav). Der vil blive benyttet en sprint periode på 1 uge, da det liner op med det ugentlige vejledermøde. Da sprint perioden er kort (normalt bruges 1-4 uger) er det vigtigt at vi får brudt vores Epics ned til User Stories der kan nåes indenfor 1 sprint. Vi vil i projektet bruge værktøjet ZenHub til GitHub for at integrere Scrum ind i projektet. Dette giver os mulighed for at samle vores projekt management og kode på vores GitHub side (<https://github.com/creditoro>). Vi har til projektet et Scrum board med følgende kolonner:

New Issues	Icebox	Backlog	In Progress	Done	Closed
	Issues med lav prioritet	Kommende issues	Igangværende issues	Færdige issues der bliver lukket næste sprint møde	

Tabel 10: Scrum Board

Alle issues er sorteret fra top til bund alt efter prioritet.

I dette projektet er der blevet defineret følgende Scrum roller

Rolle	Personer
Scrum master	Kristian
Product owner	Jakob
Developers	Alle

**Tabel 11:** Scrum roller

### 6.2.2 Værktøjer

I tabel 12 ses de værktøjer projektgruppen har benyttet under inceptionfasen og de værktøjer, der skal bruges fremadrettet.

Værktøj	Beskrivelse
PostgreSQL	PostgreSQL er en open-source objekt-relational database server.
GitHub	GitHub er en web-baseret kollaborations platform henvendt til software udviklere, der gør det muligt at versions-kontrollere projekter.
Overleaf	Overleaf er en online skriveplatform for <b>LaTeX</b> , hvor man kan være flere brugere der skriver samtidig.
UML	Unified Modeling Language
IntelliJ	Integreret udviklings miljø, som primært bruges af gruppens medlemmer.
ZenHub	ZenHub er en platform der gør det lettere at anvende Scrum i praksis.
Scrum Board	Et Scrum Board er et værktøj, der har til formål at gøre opgaverne i Sprint og Backlog synlige og overskuelige.
Pair Programming	Pair programming er en softwareudviklingsteknik, hvor to programmører arbejder sammen ved én computer.
Klassediagram	Bruges til visuelt at vise hvordan softwaresystemer er opbygget. I diagrammet beskrives systemets klasser, metoder og værdier klassen indeholder, samt klassernes relationer til hinanden.

**Tabel 12:** Værktøjer til projektarbejdet

## 7 Ressourcer

I tabel 13 ses det antal timer projektgruppen har samarbejdet om projektet (angivet pr. person). Den tid de enkelte medlemmer har brugt hjemme er ikke opgjort i tabellen.

Uge	7	8	9	10	11	12
Antal timer	6	11	8,5	6	10	15
Timer i alt	6	17	25,5	31,5	41,5	56,5

**Tabel 13:** Ressourcer brugt på projektet

Gruppens fremtidige arbejde kommer til at følge tidsplanen, der kan ses på figur 1. Det forventes at der bliver brugt omkring 10 timer/ugen pr. gruppemedlem, men hvis der er krav til mere, er gruppemedlemmerne indforstået med dette.

## 8 Konklusion

Ud fra projekt casen fik vi redegjort for det igangsættende problem samt de rammer vi skulle arbejde inden for. På baggrund af problemanalysen kunne der laves en problemformulering med relevante underspørgsmål. Vi fik identificeret kravene stillet af TV2 og hvad de indebar, samt kritiske risici. Vi fik fastslået hvilke aktører der interagerede med systemet, samt hvilke brugsmønstre de hver især fulgte. Ud fra dette kunne vi opstille et brugsmønsterdiagram, der gav en visuel repræsentation over systemet. Ved hjælp af MoSCoW metoden kunne vi udvælge de vigtigste brugsmønstre og gå yderligere i dybden med disse, ved at lave et detaljeret brugsmønster.



## 9 Procesdokumenter

### 9.1 Gruppekontrakt

#### Forventninger & Mål

- Forventer alle yder sin del
- Der forventes der bliver lagt en passende mængde tid i arbejdet. fx 2 timer er ikke nok pr. Uge.
- Ambitionsniveauet er alle gør sit bedste, så vi i sidste ende føler at vi har gjort en hel hjertet indsats.

#### Gruppens Arbejdstider

- Kører med det akademiske kvarter
- Vi mødes kl 08, her under gælder det akademiske kvarter.
- Sygdom Læge eller lignende, skal man informere gruppen. Inden kl 08 på dagen.
- Gruppen forkroster pause, efter vejleder møde. 30 min frokost pause.
- Hvis der gives hjemme arbejde, skal det laves til det aftalte tidspunkt.
- Går hjem kl 14:00 medmindre andet aftales.

#### Gruppemøder

- Gruppen mødes fast.
- Møder forgår som udgangspunkt på SDU hver tirsdag, men der er muligheder for at kunne aftale andet i løbet af projektet hvis der er brug for det.
- Hvis man kommer for sent, kan der pålægges en straf, med mindre oversagen kan begrundes.
- Straffen kan være: spiselige eller drikkelige ting.
- Hvis man ikke dukker op til møderne, skal det meddeles til medlemmerne helst inden vi mødes kl 08:00.
  - Hvis man gentagende gange udebliver, skal det løses internt i gruppen, ellers skal vejlederen inddrages.
- Hyggesnak godtages i passende mængder, men tiden skal bruges fornuftigt.

## Organisering af møder

- Ordstyrer bliver valgt på dagen.
- Referent bliver valgt på dagen.
- I Gruppen vil man forsøge at skrive referat og logbog samlet.
  - Der er en der har hoveddeansvaret for referat og logbog, resten supplerer til dem.
- Alle deadlines og indbyrdesaftaler overholdes og hvis det ikke er muligt, skal det skrives ind i logbogen.
- Hver møde skal skrives ind i logbogen og det vil sige dagsorden, eventuelt problemer osv.

## Arbejdsindsats

- Der bliver aftalt fra gang til gang hvordan der skal arbejdes.
- Pair programming
- GitHub: Merge request uden reviews

## Vejledermøde

- Der holdes vejleder møde hver eneste uge
- Der sendes senest en møde indkaldelse med lokale, link til dagsorden, samt materialer, via outlook til vejlederen senest senest 23.59 Fredag.

## Kursusdeltagelse

- Det forventes af hver gruppemedlem har læst op og fået styr på de emner, vi arbejder med, inden møderne og arbejdsdagene.
- Der skrives så vidt muligt kommentar i koden.
  - Kommentarende skrives på engelsk.

## Brug & Revision Samarbejdsaftalen

- Samarbejdsaftalen tages i brug løbende.
- Aftalen tages i brug ved konflikter.
- Aftalen kan blive revideret løbende, hvis der er behov for det.
  - Den nye aftale skal godkendes af alle medlemmer.

## Værktøjer

- Primære kommunikationsmidler
  - Tekst: Discord, Mail
  - Samtaler: Discord
- Github
  - Kodebibliotek
  - Logbog
  - Referat
  - Aftaler
- Fildeling
  - OneDrive

## Belbien Teamroller

Belbien teamroller se i billag side 25 tabel 14

## Kontakt Oplysninger

- Jakob Rasmussen
  - jakra19@student.sdu.dk
  - 52 40 56 62
- Kenneth Munk Christiansen
  - Kechr19@student.sdu.dk
  - 28 67 66 78
- Kevin Kamper Meejach Petersen
  - kepet19@student.sdu.dk
  - 50 30 88 58
- Kristian Nymann Jakobsen
  - kjako19@student.sdu.dk
  - 22 80 53 26
- Mathias Nickolaj Rasmussen
  - mara816@student.sdu.dk
  - 28 12 89 41
- Simon Jørgensen
  - sijo819@student.sdu.dk
  - 42 83 25 60

## 9.2 Vejlederkontrakt

- Der skal laves en gennemgang og dagsorden til hvert vejledermøde. Det skal sendes til vejleder senest fredag kl. 23.59.
- Alt skal så vidt muligt holdes i GitHub.
- Vejleder læser alle dokumenter og materialer igennem, vi sender, og dette skal gøres før vejledermødet.
- Vi tager ansvar for eget studie, og står selv for fremmøde til undervisning.
- Vejledermøde er som udgangspunkt hver tirsdag kl. 11.15.
  - Hvis tidspunktet ændres aftales det med vejleder via mail.
- Vejleder gør gruppen opmærksom på om gruppearbejdet er på afveje, samt at eventuelle deadlines ikke overholdes, så gruppen kan nå i mål uden konflikter.
- Vejleder giver besked via mail hvis han bliver forhindret i at afholde vejledermøde, eller hvis tidspunktet bliver ændret.
- Hvis gruppen skulle blive utilfreds med vejleders indsats, skal dette håndteres hurtigst muligt.
- Hvis gruppen kommer på afveje i forhold til samarbejdsaftalen, skal vejleder være i stand til at vejlede gruppen gennem eventuelle konflikter.

### 9.3 Belbin Gruppeprofil

Teamrolle	Navn	Navn	Navn
Idemand	Mathias	Kevin	Jakob
Kontaktskaber	Kevin	Kenneth	
Koordinator	Kristian	Mathias	
Opstarter	Simon	Mathias	
Analysator	Jakob		
Formidler	Kevin	Kristian	
Organisator	Kenneth	Simon	
Afslutter	Kenneth	Simon	
Specialist	Kristian	Jakob	

**Tabel 14:** Belbien Teamroller

## A Bilag

### A.1 Logbog

**Logbog** [Github.com/creditoro/logbook](https://github.com/creditoro/logbook)