

Forprosjektrapport
Bachelorprosjekt 2021
Oslo metropolitan University

Gruppenummer: 42

Tittel: Rapportgenerator

Gruppemedlemmer:

Dion Dehari, s333723
Ole Alexander Jensen Elg, s326174
Marius Andre Haugsand, s326277
Jon Magnar Austnes Lunde, s333775

Periode: Januar 2021 til Mai 2021

Oppdragsgiver: Styret.com

Kontaktperson: Kristoffer Gregersen Skappel
Tlf: 95876157
Mail: Kris@styret.com

Veileder: Tengel Aas Sandtrø
Mail: tengel@oslomet.no

Oppgave: Lage en rapportgenerator webapplikasjon.

Sammendrag

I vår bacheloroppgave skal vi lage en standalone webapplikasjon for Styret.com. Dette skal være en webapplikasjon som håndterer utlegg hvert enkelt styremedlem (heretter omtalt som bruker) har gjort på vegne av borettslaget sitt. Når en bruker er logget inn på Styret.com sin plattform skal de kunne bruke vår applikasjon til å laste opp ett eller flere utlegg som samles i en generert PDF-fil som sendes som et bilag til regnskapsavdelingen for borettslaget brukeren tilhører. Brukeren skal også kunne velge mellom ulike PDF-templates på bilaget som blir generert, disse administreres av admin-brukere for hvert enkelt borettslag.

Dagens situasjon

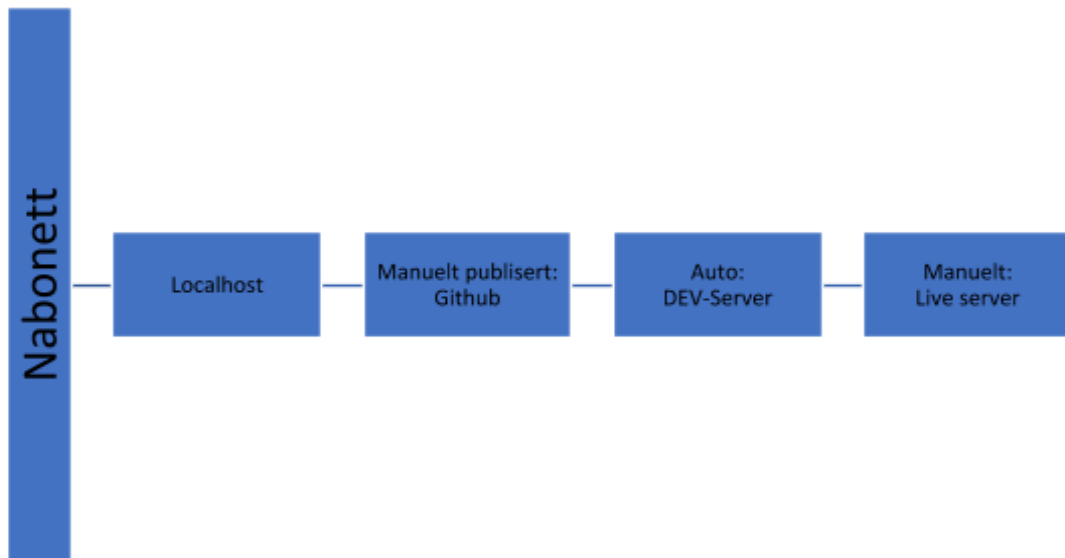
Styret.com er en 4 år gammel bedrift som tilbyr en plattform for eiendomsforvaltning som per dags dato brukes av 1700 borettslag og sameier rundt om i Norge. Denne plattformen tilbyr tre forskjellige portaler for hhv. vaktmester, styret og forretningsfører med hver sin skreddersydde GUI som lar disse aktørene utføre sine arbeidsoppgaver i borettslaget på en effektiv måte. Styret.com har mange visjoner for videre utvikling av webapplikasjonen sin, blant annet ønsker de muligheten for at medlemmer av styret i borettslaget skal kunne laste opp utlegg gjort på vegne av borettslaget som samles i et bilag som sendes til regnskap.

Mål og rammebetingelser

Det overordnede målet for denne oppgaven er å utvikle en standalone webapplikasjon der et styremedlem kan laste opp utlegg gjort på vegne av borettslaget. Det skal være mulig å registrere flere utlegg som samles i en generert PDF-fil, denne PDF-filen skal sendes videre som et bilag til regnskap.

Vi har følgende rammebetingelser:

1. Devops via Github.
2. Styret.com distribuerer API-komponenter som gir oss tilgang til deres plattform.
3. Autentisering er allerede håndtert via Azure B2C.
4. Microsoft Visual Studio.
5. MS SQL.
6. Angular 2+.
7. CSS-rammeverket Angle (Bootstrap Template).



Vi har følgende delmål for produktet vårt:

1. Skal være en standalone webapplikasjon lagd fra bunnen.
2. Ett eller flere utlegg skal kunne lastes opp og sendes til regnskap som bilag i PDF-format.
3. Mulighet for bruker til å søke etter bilag i databasen.
4. PDF-fil skal genereres ut ifra et Word-template.
5. Implementering av automatisk lagring av utkast av uferdig bilag.
6. Design skal stå i stil med resten av Styret.com platformen.
7. Applikasjonen skal tilfredsstillе dagens krav om universell utforming.
8. Det skal lanseres en alpha versjon av applikasjonen innen 8. mars.
9. Vi skal utføre enhetstesting underveis.

Wireframe av webapplikasjon

Vi har laget en wireframe-modell sammen med kunden, dette er første utkast av hvordan de ønsker at webapplikasjonen skal se ut:

The wireframes show four different views of a web application titled 'Utleggsmodule'. Each view has a sidebar with three links: 'Registrerte kontakter', 'Tidligere utlegg', and 'Administrasjon'.

- Top Left:** A login form with fields for 'Brukernavn' and 'Passord', a 'Logg inn' button, and a link for 'Glemt passord?'.
- Top Right:** A view with a 'Nytt utlegg' button and a table header: 'Dato | sum | Utlegger | status | lagt inn av'. Below the header is a large empty box labeled 'Tabell med tidligere utlegg'.
- Bottom Left:** A form for adding a new entry with fields for '*Navn på utlegger', '*Bankkonto', 'Telefon', 'Epost', and 'Generell beskrivelse'. Below these is a 'Last opp kvitering' button. At the bottom are three buttons: 'Lagre utkast', 'Send', and 'Avbryt og slett'.
- Bottom Right:** A form with fields for 'Sum' and 'Komentar'. Below these are three buttons: 'Last opp', 'slett', and 'Lagre'.

Fra øverst til venstre: Innlogging, Meny-side, Utfylling av brukerinformasjon til bilag, Opplasting av utlegg.

Løsninger/alternativer

Design av applikasjon

Angle(Bootstrap template)	Angle med modifikasjoner	CSS fra bunn
<p>Fordeler: Fordel for kunden da de bruker denne templatene fra før, likt design over hele plattformen. Det vil kreve mindre jobb fra oss siden vi er godt kjent med Bootstrap fra før. Mer tid til å implementere viktigere funksjonalitet i applikasjonen.</p> <p>Ulemper: Det vil gi kunden et nettsted som er lite originalt. Vanskeligere å spille på gjenkjennelsesfaktoren hos bruker da templatene er veldig generisk.</p>	<p>Fordeler: Elementene vil være skreddersydd for målet med webapplikasjonen og isolert sett vil det gi en bedre webapplikasjon for kunden. Elementene vil skille seg litt ut fra templatene og webapplikasjonen vil få et eget uttrykk.</p> <p>Ulemper: Kunden vil få en webapplikasjon som ikke er identisk med resten av designelementene på plattformen. Vil kreve litt ekstra ressurser for liten gevinst da elementene som tilbys er allerede.</p>	<p>Fordeler: Vi kan tilby kunden et design som er helt skreddersydd til webapplikasjonen formål og bedriftens uttrykk. Det vil være god trening for oss i å skrive CSS.</p> <p>Ulemper: Dette vil spise opp mye av tidsbudsjettet vårt og gå på bekostning av annen funksjonell kode som er viktigere for kunden. Det vil kreve ekstra påpasselighet rundt universell utforming som vi ellers ville fått gratis av et template.</p>

Generering av PDF-fil

Eksisterende løsning	Kodet fra bunnen
<p>Fordeler: Kunden får en løsning som er kvalitetssikret over tid. Veldig rask integrasjon.</p> <p>Ulemper: Kunden er avhengig av tilgjengelighet hos tilbyder av API. Den er ikke nødvendigvis skreddersydd til kundens behov. Kunden ønsker å generere PDF fra en Word-template, ikke sikkert løsningen støtter dette. Potensielt dårligere ytelse siden funksjonen ikke blir utført lokalt. Skal håndtere sensitiv data i form av kontonr. osv. For implementasjonen av en tredjeparts API må vi ta hensyn til pris.</p>	<p>Fordeler: Kunden vil få en funksjon som er tilpasset behov og som kan håndtere Word-template. Virker som et komplekst problem med stort læringsutbytte for oss.</p> <p>Ulemper: Vi antar at det er større risiko for at uforventet kompleksitet kan oppstå underveis. Veldig mye av tidsbudsjettet vårt vil gå med til å utvikle denne biten.</p>

Lagring av utlegg underveis

Automatisk lagring av utkast	Lagring via knapp
<p>Fordeler: Null ansvar for lagring av arbeid blir lagt på bruker. Kunden vil få en mer automatisert og moderne webapplikasjon med bedre brukervennlighet.</p> <p>Ulemper: Potensielt mye jobb å implementere, dette er noe vi må bruke tid på å undersøke på forhånd. Avhengig av at funksjonaliteten er godt testet for å ikke skape dårlig brukeropplevelser.</p>	<p>Fordeler: Veldig enkel funksjonalitet, krever lite ressurser fra oss. Kunden får en løsning som utfører ønsket funksjonalitet med lav risiko for systemfeil.</p> <p>Ulemper: Kunden får en "utdatert" og enkel funksjon. Høy risiko for brukerfeil, da bruker lett kan glemme å trykke på knappen. Må nok implementere en sjekk som fanger opp om bruker har glemt å trykke på knappen når han forlater siden.</p>

Søking i bilags-database

Søkefunksjon	Avansert filtrering	Enkel sortering på kolonner
<p>Fordeler: Kunden får en funksjon som øker brukervennligheten, spesielt ved store ansamlinger av data i større borettslag. Gjør det lett for bruker å finne frem til spesifikke bilag gjennom søk på utleggers navn eller hva som ble lagt ut for.</p> <p>Ulemper: Bruker må navngi bilagene på en måte som beskriver utleggene, vanskelig å kontrollere dette programmatisk. Om regnskaps-databasen vokser seg veldig stor kan søket ta lang tid om vi ikke implementerer riktig søkealgoritme.</p>	<p>Fordeler: Kunden får en søkefunksjon som er skreddersydd for deres database. Bruker får predefinerte egenskaper å filtrere på, feks. dato, utleggers navn, borettslag. Det vil gi bruker mindre frihet enn søkefunksjonen, som igjen gir mindre rom for brukerfeil og en bedre brukervennlighet.</p> <p>Ulemper: Mindre frihet for søk stiller større krav til at vi som utviklere ser alle use-cases en bruker vil trenge for å søke seg frem til spesifikke bilag.</p>	<p>Fordeler: En gammel løsning som de aller fleste brukere er kjent med.</p> <p>Ulemper: Bruker får servert hele databasen uten mulighet for å filtrere ut relevant data.</p>

Analyse av virkninger

Kunden har mange ønsker om forskjellige funksjoner webapplikasjonen skal tilby. Ved å gå for eksisterende gode løsninger innenfor design, i form av et Bootstrap-template vil vi få mer tid til å utvikle en webapplikasjon som møter flere av kundens ønsker. Vi antar at MS SQL tilbyr flere av løsningene våre allerede og da har det liten hensikt å finne opp en ny løsning på dette. Vi kommer til å bestemme oss for hvilken søkefunksjonalitet vi vil benytte oss av etter vi har hentet inn informasjon på feltet slik det står beskrevet i fremdriftsplanen.

For at kunden skal sitte igjen med en webapplikasjon som er mest mulig brukervennlig vil vi implementere automatisk lagring av utkast underveis da vi anser at mengden arbeid vi må legge ned for å oppnå en stor gevinst for bruker er verdt det.

Angående generering av PDF har vi to vidt forskjellige alternativer til løsning som vil definere omfanget på prosjektet vårt. Vi har inntrykk av at kunden ønsker å gå for den mest tidseffektive og testede løsningen som er å benytte et tredjeparts API som vil gi oss mer tid til å utvikle annen funksjonalitet. Men vi ser også verdien av å lage en egen løsning som vil gi oss et større læringsutbytte selv om det vil oppta en veldig stor del av vår tidsbudsjett. Vi kommer til å ta et møte med kunden for å fastslå dette og drøfte fordeler og ulemper rundt valget.

Vi ønsker å levere en webapplikasjon som møter flest mulig av kundens ønsker til funksjonalitet. For å få til dette må vi gå for allerede eksisterende løsninger for å spare tid. Vi tenker at det ikke er nødvendig å finne opp kruttet på nytt, men heller benytte oss av godt testede løsninger.

Kravspesifikasjon

Funksjonelle krav

Systemkrav

Prioritert funksjonalitet

Systemet skal kunne:

1. Bruke eksisterende autentisering systemer hos kunde (Azure Active Directory B2C)
2. Ha et system for brukerne slik at de kan registrere utlegg som lagres i et generert bilag med filtype PDF.
3. Støtte for opplasting av bilde på kvitteringer.
4. Gi brukeren muligheter til å se tidligere bilag laget av brukere fra borettslaget.
5. Sender ferdig generert bilag til regnskap.

Ønsket funksjonalitet

Systemet bør kunne:

1. Automatisk lagre utkast av det siste utlegg som har blitt jobbet på og ikke registrert.
2. Lage nye PDF-templates for bilag fra Word-template.
3. Gi muligheten til å søke og filtrere gjennom bilag.

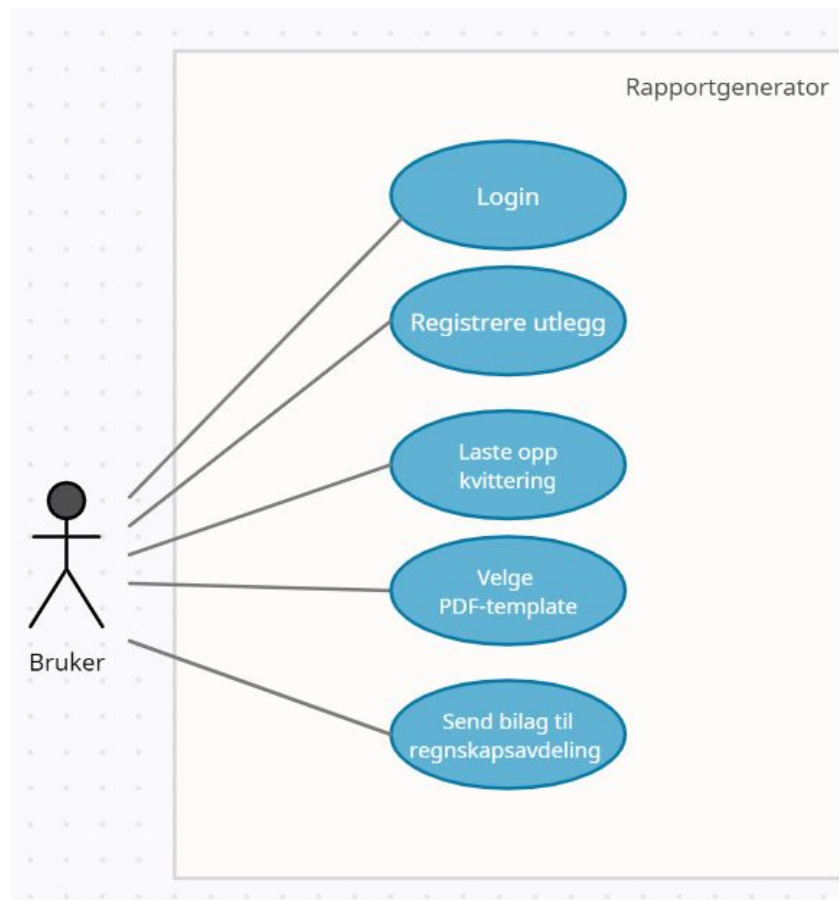
Eventuell tilleggsfunksjonalitet

System kan eventuelt kunne:

1. Registrere fordeling av honorar og liknende.
2. Ha en signeringsløsning med BankID, SMS/Email.
3. Være flerspråklig. Engelsk og norsk.
4. Sende invitasjon, påminnelse og bekreftelser for signering av bilag.

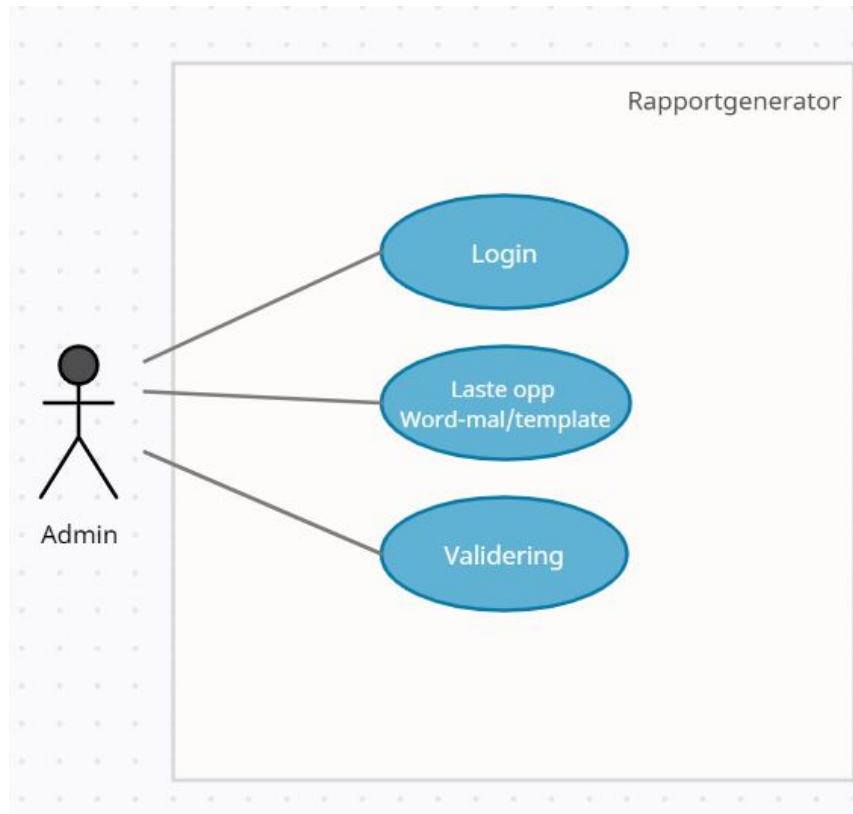
Brukerkrav

For å illustrere brukerkravene viser vi samspillet mellom applikasjonen og brukere i form av et Use Case scenario og en tekstlig beskrivelse.



Navn	Kunden vil sende bilag
Mål	Ønsker å samle ett eller flere utlegg gjort på vegne av borettslaget. De samles i en generert PDF-fil som sendes til regnskapsavdelingen for borettslaget.
Aktør	Bruker (Styremedlem)
Trigger	Behov for å håndtere utlegg gjort på vegne av borettslag.
Pre-betingelser	Bruker må ha konto hos Styret.com
Post-betingelser	Kunden får registrert og sendt utlegg.

Hovedflyt	<ol style="list-style-type: none"> 1. Bruker logger seg inn i webapplikasjonen. 2. Bruker trykker på "Nytt utlegg"-knappen 3. Bruker fyller ut brukerinformasjon. 4. Bruker trykker på "Last opp kvittering"-knappen 5. Systemet åpner popup-vindu for Opplasting av utlegg 6. Bruker registrerer informasjon om utlegget og trykker på "send"-knappen 7. Systemet validerer input for utlegg og lukker popup-vindu. 8. Bruker velger ønsket PDF-template 9. Bruker trykker på "Send"-knappen 10. Systemet validerer brukerens input. 11. Systemet generer PDF og sender den til regnskapsavdelingen.
Alternativ flyt	<p>A1.1 Systemet godkjenner ikke login-informasjon. A1.2 Bruker blir sendt tilbake til steg 1 og blir bedt om å prøve på nytt.</p> <p>A7.1 Systemet godkjenner ikke kvitteringen. A7.2 Bruker blir sendt tilbake til steg 6 i hovedløpet med beskrivende feilmelding.</p> <p>A10.1 Applikasjonen avviser input fra bruker. A10.2 Registrering av utlegg tilbakestilles og bruker sendes tilbake til steg 6 i hovedflyten.</p>



Navn	Admin laster opp PDF-template
Mål	Admin vil laste opp et PDF-template for borettslaget sitt.
Aktør	Admin
Trigger	Behov for et nytt PDF-template for generering av bilag i borettslag.
Pre-betingelser	Personen må ha admin-rettigheter.
Post-betingelser	Admin har lastet opp et nytt PDF-template til databasen.
Hovedflyt	<ol style="list-style-type: none"> 1. Admin logger seg inn med admin-rettigheter. 2. Admin trykker på "administrasjon"-knappen. 3. Systemet sender Admin til "administrasjon"-vindu.

	<p>4. Admin trykker på “Last opp template”-knappen og velger fil.</p> <p>5. Systemet validerer filtypen.</p> <p>6. Admin trykker på “Bekreft template”-knappen</p> <p>7. Systemet sender Admin tilbake til Meny-vindu.</p>
Alternativ flyt	<p>1.1 Systemet kan ikke validere innloggingsinformasjon.</p> <p>1.2 Systemet sender Admin tilbake til steg 1 med feilmelding.</p> <p>5.1 Systemet har ikke fått docx-fil.</p> <p>5.2 Systemet sender Admin tilbake til steg 4 med feilmelding.</p>

Forretningskrav

- Styret.com ønsker at medlemmer av styret i borettslaget skal kunne laste opp utlegg gjort på vegne av borettslaget som samles i et bilag som sendes til regnskap.
- De ønsker også støtte for opplasting av Word-template som bestemmer utseende på generert PDF-fil.

Ikke-funksjonelle krav

Produktkrav

Brukervennlighet

1. Nettstedet skal følge retningslinjer og krav til universell utforming.
2. Språket på applikasjonen skal være norsk bokmål.
3. Designet skal være responsivt og fungere på telefoner, nettbrett og PC-er.
4. Matche gjennomgående designet til resten av nettstedet.

Effektivitetskrav

1. Applikasjonen skal kjøre med akseptabel hastighet.

Pålitelighetskrav

1. Data skal være trygt lagret og behandlet.
2. All informasjon på nettstedet skal være oppdatert til enhver tid.

Organisatoriske krav

Verktøy og programmeringsspråk

1. Programmeringsspråkene HTML5, CSS, Javascript, C#
2. Rammeverk vi bruker er .NET, Angular 2+
3. CSS-template Angle ønskes, men vi har ganske frie tøyler
4. Nettsiden skal fungere på alle de mest brukte nettlesere som Chrome, Safari, Edge, Firefox, Opera

Utviklingsmetodikk

1. Kanban-metode

Eksterne krav

Vi kommer til å innhente informasjon om personvernlover som kan være aktuell for dataen vi skal håndtere.

Risikoanalyse

		Sannsynlighet		
		Liten	Middels	Stor
Konsekvens	Liten	Lav	Lav	Moderat
	Middels	Lav	Moderat	Høy
	Stor	Moderat	Høy	Høy

Under utviklingsprosessen kan mange situasjoner oppstå. Derfor er det viktig at vi kartlegger disse hendelsene tidlig slik at vi har en plan klar på hvordan vi skal takle dem. Vi har gjort en risikoanalyse av utviklingsprosess der vi kategoriserer de mulige risikoene.

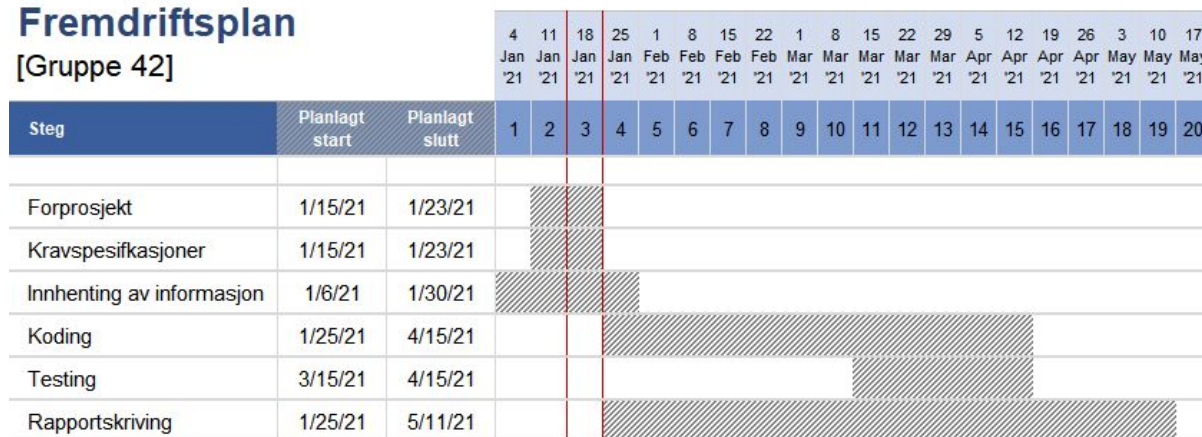
#	Risikoer	Risikovurdering			Mulige tiltak
		Konsekvens	Sannsynlighet	Risikonivå	
1	Sykdom	Liten	Middels	Lav	Hjemmekontor, Vaske hender, 2m avstand
2	Tid	Middels	Liten	Lav	Ha jevnlige møter med situasjonen oppdatering
3	Kravendringer	Stor	Liten	Moderat	God kommunikasjon med veileder og Styret.com
4	Tap av data	Stor	Liten	Moderat	Backup på github
5	Feiltolking av oppgaven	Middels	Liten	Lav	God kommunikasjon med veileder og Styret.com
6	Gruppe stridighet	Stor	Liten	Moderat	God kommunikasjon mellom gruppemedlemmene.
7	Mer arbeid enn antatt	Middels	Middels	Moderat	Kartlegge arbeid bedre.

8	Ikke tid for testing	Stor	Middels	Høy	Sikte tidlig med bra code coverage for å minimere faren.
---	----------------------	------	---------	-----	--

Arbeids- og fremdriftsplan

Fremdriftsplan

[Gruppe 42]



Milepæl	Beskrivelse	Uke	Dato
M1	Kjent med personvernlover angående data vi håndterer.	5	01.02.21
M2	Bli kjent med devops og utviklingsmiljø.	5	01.02.21
M3	Bestemme oss for PDF-generator løsning.	5	01.02.21
M4	Bestemt oss for søkeløsning i DB.	6	08.02.21
M5	Design UI/Hi-Fi prototype.	6	08.02.21
M6	Lansere en alpha-versjon.	10	08.03.21
M7	70% Code coverage av backend funksjonaliteter.	14	05.04.21
M8	70% Code coverage av frontend funksjonaliteter.	15	12.04.21
M9	Ferdigstille Meny-siden.	16	19.04.21
M10	Ferdigstille PDF-løsningen.	17	26.04.21
M11	Klargjør for lansering.	18	03.05.21

Viktige Datoer

22.01.21: Levere signert kontrakt mellom gruppen og arbeidsgiver.

23.01.21: Levere forprosjektrapport.

25.05.21: Innlevering av hovedrapport.

07.06.21: Presentasjon av prosjekt.