



## Forside IS-304: 2018

Tittel:

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder	Janis Gailis
Oppdragsgiver:	Rune Solberg

Studenter:

Etternavn	Fornavn
Ajmi	Adam
Eskeland	Anders
Methi	Håkon
Olsen	Rune Vålandsmyr
Tanase	Andrei

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA X	NEI__
Kan besvarelsen brukes til undervisningsformål?	JA X	NEI__
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA X	NEI__



# Bachelorprosjekt: Screenz

## Brukervennlig innholds-designer

# Forord

Denne rapporten er dokumentasjonen til bachelorprosjektet for oss i gruppe Atemos Screenz, Vi på Atemos Screenz er 3. årsstudenter på IT og Informasjonssystemer ved Universitetet i Agder (UiA). Bachelorprosjektet gjennomføres for Screenz, hvor produkteier er Rune Solberg.

Bachelorprosjektet skal bygges på og gjennomføres med kunnskap vi har opparbeidet oss gjennom 3 år på studiet.

Vi vil gjerne takk vår veileder Janis Gailis som på tross av at han har vært i utlandet har gitt oss gode tilbakemeldinger og god råd for gjennomføring av prosjektet. I tillegg vil vi takke kontorfellesskapet CoWorx i sin helhet for både bruk av lokaler, men også en god del av personene som jobber der for testing av prototype og annen interessant dialog og rådgivning underveis.

Og sist, men ikke minst så rettes en stor takk til Hallgeir Nilsen for alt han har gjort, både dette siste semesteret, men også gjennom hele bachelorprogrammet. Alt av forarbeid og organisering iht bachelorprosjektet, til gjennomføring samt konstruktive tilbakemeldinger i forelesningene.

# SAMMENDRAG

Screenz er et produkt som er utviklet og designet lokal i Kristiansand, men laget med det internasjonale markedet som målestokk. Produkteier Rune Solberg har i flere år jobbet med grafisk design og UX i den tradisjonelle markedsføringsbransjen, og også opparbeidet seg god kjennskap til markedet for Digitale Skilt. Han har derfor sett behovet for en sammenslåing og utviklingspotensiale av disse, og laget programvaren Screenz.

Tjenesten fungerer i seg selv strålende, men det har manglet en vital del, og det er hvordan lage et godt innhold av høy kvalitet, men med en lav terskel. Hvordan få kundene til å lage sine egne profesjonelle produktkampanjer, og legge dette til skjermen som allerede er portal for markedsføring for butikken.

Vi har derfor utarbeidet en tjeneste som kan implementeres i et allerede eksisterende produkt og som gjør at kundene til Screenz kan lage sitt eget innhold uten å bruke tredjepart (grafiske designere).

Produktet vårt er en web-tjeneste som kobles opp mellom "legg til nytt innhold" og "last opp i Screenz-tjenesten". Grunnet at den er webbasert så trenger man ikke å laste ned egen programvare, og alle vanlige bildeformater er støttet. I tillegg genereres det en URL-lenk som enkelt legges inn i Screenz.

Gjennom Scrum-basert prosjektstyring, har vi skaffet oss domenekunnskap, laget use-cases og brukerhistorier, laget design og funksjoner, og til slutt testet og implementert løsningen som en fungerende prototype.

Underveis har vi også testet ut flere forskjellige verktøy for å sikre oss at prosjektet ble gjennomført slik at vi nådde våre mål best mulig. Vi har også støttet på "the real world" i form av tekniske problemer, under og overestimering av tidsbruk, problem med kommunikasjon mellom ressurspersoner og oss, prosjektgjennomføringsproblemer, og sist men ikke minst: opplevd utfordringen det er å sette seg inn i en produkteiers system og forsøke å videreutvikle det til produkteiers standard.

<b>Bachelorprosjekt: Screenz</b>	<b>1</b>
<b>Forord</b>	<b>2</b>
<b>SAMMENDRAG</b>	<b>3</b>
<b>1. INTRODUKSJON</b>	<b>7</b>
1.1 Om oss	7
1.2 Oppdragsgiver	7
1.3 Hva er Screenz?	7
1.4 Prosjektbeskrivelse	8
1.5 Begrunnelse av valg av prosjekt	8
1.5.1 Fokusering og innsnevring av prosjektoppgaven	9
1.6 Terminologi	9
<b>2. METODE</b>	<b>11</b>
2.1 Kvalitetssikring	11
2.1.1 Rune Solberg som kvalitetsstandard	11
2.1.2 The Iron Triangle	12
2.2 Scrum	13
2.3 Kvalitet i vårt prosjekt	14
2.3.1 Verktøy	14
2.3.2 Kommunikasjon	17
2.3.3 Gruppearbeid	19
2.3.4 Testing av prototype	20
2.3.5 Føring av dagbok	20
<b>3. ANALYSE</b>	<b>21</b>
3.1 Brukerhistorier	21
3.1.1 Prioritering/MoSCoW	21
3.1.2 Eksempler på Brukerhistorier	22
3.2 Use-Cases	23
3.2.1 Eksempel på Use-case	24
3.3 Kravspesifikasjon	25
<b>4. DESIGN</b>	<b>26</b>
4.1 State-chart	27
4.2 Sekvensdiagram	28
4.3 Designprinsipper	29
4.4 Design-skisser og prototyping	33
4.5 Brukertestning	37
<b>5. IMPLEMENTASJON</b>	<b>39</b>

5.1 Arkitektur	39
5.2 Oppsett av prosjekt med instruks fra Panteon	40
5.3 Oppsett av vårt nye prosjekt	41
<b>6. PROSJEKTGJENNOMFØRING</b>	<b>42</b>
Pre-sprint	42
Sprint 1	43
16/02 - Veiledningsmøte 1	43
Plan	43
Review/refleksjon	44
Sprint 2	44
Plan	45
Review/refleksjon	45
Sprint 3	45
06/04 - Veiledningsmøte 2	45
Plan	45
Review/refleksjon	46
Sprint 4	46
Plan	46
Review/refleksjon	46
Sprint 5	47
Plan	47
Review/refleksjon	47
Sprint 6	47
Plan	47
Review/refleksjon	48
Sprint 7	48
11/05 - Veiledningsmøte 3	48
Plan	49
Review/refleksjon	49
<b>6. RESULTAT</b>	<b>50</b>
<b>7. DISKUSJON</b>	<b>51</b>
7.1 Utfordringer med sen informasjon fra Peter Hamre ved Panteon	51
7.2 Scrum	52
7.3 Hva kunne vi gjort annerledes?	53
<b>8. KONKLUSJON</b>	<b>55</b>
<b>9. REFERANSER</b>	<b>57</b>
<b>10. VEDLEGG</b>	<b>58</b>
Vedlegg 1 - Evaluering fra oppdragsgiver	58
Vedlegg 2 - Selvevaluering	59

Vedlegg 3 - Brukerhistorier	61
Vedlegg 4 - Use Cases	67
Vedlegg 5 - Instruksjoner for oppsett/gjenskapelse av error for Panteon	72
Vedlegg 6 - Backlog	77
Vedlegg 7 - Referater fra Veiledningsmøtene	81
Første veiledningsmøte - 16.02.2018	81
Andre veiledningsmøte - 10.04.2018	82
Tredje veiledningsmøte - 11.05.2018	83

# 1. INTRODUKSJON

## 1.1 Om oss

Vår gruppe, ATEMOS, består av fem medlemmer; Adam Ajmi, Anders Eskeland, Håkon Methi, Rune Vålandsmyr Olsen og Andrei Tanase. Navnet ATEMOS baserer seg på første bokstav i etternavnet på oss, pluss en S på slutten. ATEMOS har en dobbeltbetydning fordi det er et spansk ord som betyr “vi knytter”.

De fleste av oss har samarbeidet på emner tidligere, og det har fungert bra. Derfor så vi for oss at vi skulle fungere godt sammen også denne gangen.

## 1.2 Oppdragsgiver

Oppdragsgiveren vår er Rune Solberg i Rude AS som har kontor på CoWorx. CoWorx er et kontorfellesskap i Kristiansand sentrum som tilbyr kontorplasser til gründere og oppstartsbedrifter. Rune Solberg er grafisk designer med mange års erfaring, og han kjenner derfor den tradisjonelle markedsføringsbransjen godt. I tillegg har han utviklet god kjennskap til Digitale Skilt. Han har derfor sett behovet for en sammenslåing og utviklingspotensiale av disse, og slik ble ideen til programmet Screenz født. Han har flere prosjekter samtidig, og Screenz er et av dem. Han er ikke selv programmerer, og outsourcer dette til et selskap som heter Panteon. Her er Peter Hamre hovedansvarlig for utviklingen av Screenz. Vi har hatt et møte med Peter Hamre, og benyttet oss av epost for å få svar på spørsmål vi lurte på angående det tekniske. Han var vår kontaktperson når det kom til det programmeringsbiten.

## 1.3 Hva er Screenz?

Screenz er en web-basert tjeneste ([screenz.io](https://screenz.io)) som hjelper deg å administrere dine digitale skilt (digital signage). Med digitale skilt menes det visning av multimedieinnhold på digitale visningsflater, fremfor å bruke vanlige trykte plakater, eller andre fysiske markedsføringseffekter (Digital Skilting, 2016). Det finnes lignende løsninger der ute i dag, men de er enten altfor kompliserte og krever høy teknisk kompetanse for å bruke, eller mangler en del funksjonaliteter som vår produkteier tror er viktig for brukerne. Ofte bærer de også preg av å se litt gammeldagse ut, slik at det får litt uprofesjonelt eller utgått designmessig brukergrensesnitt. Dette fører til at mange bedrifter ikke har råd, eller er villig til å betale for en slik tjeneste. Screenz vil tilby enkle og brukervennlige løsninger som



bedriften selv kan administrere. Dette reduserer også behovet for å bruke 3.parts designere til å produsere innholdet.

## 1.4 Prosjektbeskrivelse

Vår oppgave er å lage en web-basert innholds-designer til Screenz.io. I denne designeren skal kundene enkelt kunne legge til bilder og lage sine egne digitale reklameplakater uten at de trenger å bruke penger på å outsource denne oppgaven til 3.part. Målet er at designer skal kunne utføre en funksjon som er enklere å bruke enn for eksempel Adobe Photoshop, men skal gi det samme profesjonelle resultatet. Det skal være få og enkle redigeringsmuligheter, som gjør at terskelen for å bruke designeren er lav, men resultatet er høy kvalitet.

Det skal også nevnes at Rune Solberg var veldig åpen for forslag i henhold til hva vi ønsket å gjøre. Han hadde noen ideer til prosjektrammer, men vi kunne mer eller mindre styre innholdet selv. Han mente at jo flere retningslinjer han gav oss, jo vanskeligere var det for oss å levere det han hadde beskrevet og det ville være en demotiverende faktor for oss. Desto mer vi selv kom opp med, jo mer hadde vi lyst til å gjennomføre og ofre for at vår egen ide og produkt skulle bli en realitet. Dette var en filosofi vi var enige i, men vi ser i etterkant at dette ikke bare har positive effekter. Det resulterer i dertil mer ansvar og egeninnsats på oss som gruppe. Å gjennomføre et delvis selvkomponerende prosjekt satt inn i rammen til en produkteiers allerede fungerende produkt gjør mål og delmål vanskeligere å nå.

## 1.5 Begrunnelse av valg av prosjekt

Vi ble som nevnt gitt stor frihet i forhold til hva vi ønsket å gjennomføre av prosjekt, etter at vi fikk tilslaget på å kunne gjennomføre bachelorprosjektet hos Rune Solberg. Da vi hadde lite domenekunnskap innen område digitale skilt, ba vi om forslag fra Rune som kjenner markedet godt. Vi fikk da 3 forslag:

- Beacons - Bluetooth-teknologi som sender ut et «her er jeg»-signal som kan fanges opp diverse støttede utstyr når du befinner deg i nærheten.
- Designer - Brukervennlig web-basert tjeneste for å produsere innhold til reklameplakater
- Knapp - Bluetooth-teknologi for å kunne sette starte en hendelse, som å endre på innholdet på en skjerm eller publisering av informasjon på sosiale medier, via å trykke på en enkelt frittstående knapp.

Valget vårt ble designeren da vi følte at bluetooth-teknologi og beacons var områder vi hadde lite kunnskap og vi ville hatt store problemer med å kunne produsere et

fungerende produkt i løpet av perioden bachelorprosjektet gikk. De fleste av oss var enige om at beacons var den teknologien som var mest interessant, men at dokumentasjonen på dette var så vag at det ville vært vanskelig å gjennomføre.

### 1.5.1 Fokusering og innsnevring av prosjektoppgaven

Vi oppdaget nokså tidlig at vi trengte å innsnevre oppgaven, da vi ble klar over at det er 2 hovedkategorier innen multimedialinnhold: bilder eller film. Da vi antok at det er lettere å redigere et bilde gjennom en web-tjeneste enn å redigere en film, så kuttet vi ut film og fokuserte kun på redigering av bilder. Dette gjør også at samlet trafikkbelastning blir mindre, og det gjør at kravspesifikasjonene generelt blir lavere når det gjelder vårt bachelorprosjekt.

Underveis har vi innsnevret fokuset for å kunne lage et mer tilfredsstillende fungerende produkt. Noen eksempler er:

- beskjæring (cropping) av bilde
- automatisk tilpassing til Facebook, Instagram eller andre sosiale medier
- automatisk tilpassing til skjermers sideforhold (vi har kun landskap og portrett 16:9 - 9:16)
- søkemulighet, for eksempel søke på tags eller andre aktuelle temaer

## 1.6 Terminologi

I dette prosjektet bruker vi noen ord som vi synes det er viktig å definere på forhånd.

**Template** - Vi bruker ofte ordet template i denne rapporten. En template er rett og slett en mal for hvordan et skjermbilde/view skal se ut. I en template er posisjonen til teksten og bilde satt. Man kan da velge å endre bilde og tekst.

**View** - Et view er en template som er ferdig utarbeidet, med bilde og tekst, som er lagret som et eget bilde. Se på det som det ferdige innholdet som skal vises på skjermen.

Vi har i brukertester sett at View ikke er et brukervennlig nok navn, men har ikke endret til et annet navn grunnet at view brukes gjentagende gjennom rapporten inkludert diagrammer.

**Playlist** - En spilleliste der man kan velge mediafiler som skal spilles av til gitte tider i døgnet i Screenz. Her er det viktig å presisere at i Screenz-systemet så blir et view sett på som en mediafil.

**Digital Signage** - Visning av multimedialinnhold på digitale visningsflater, typisk smart-TV'er og lignende utstyr. Det er dette markedet Screenz ønsker å få fotfeste i.

**MVP** - Står for minimum viable product. Det betyr at produktet dekker minimumskravene til de første brukerne, og inkluderer minimums funksjonalitet for at systemet skal fungere.

## 2. METODE

### 2.1 Kvalitetssikring

Hva er kvalitet?

“Enkelt sagt er kvalitet evnen til å tilfredsstille kundens eller brukerens krav og forventninger.” (Kvalitet, 2018)

Hva er kvalitetstesting?

“Kvalitetssikring, planlagte og systematiske aktiviteter som gjøres for å oppnå at et produkt eller en tjeneste vil oppfylle kravene til kvalitet.”

Kvalitet er et relativt begrep hvor det finnes flere definisjoner. Kvalitet er subjektivt, og det kan variere fra person til person om hva man synes god kvalitet er. Kvalitet kan deles i to; prosesskvalitet og produktkvalitet.

Prosesskvaliteten er sett med våre øyne som systemutviklere. Er prosessen mot å lage et produkt av høy kvalitet?

Produktkvaliteten går på kvaliteten av sluttproduktet. Det er brukerens tilfredshet med produktet som er viktigst i denne delen av kvalitetssikringen.

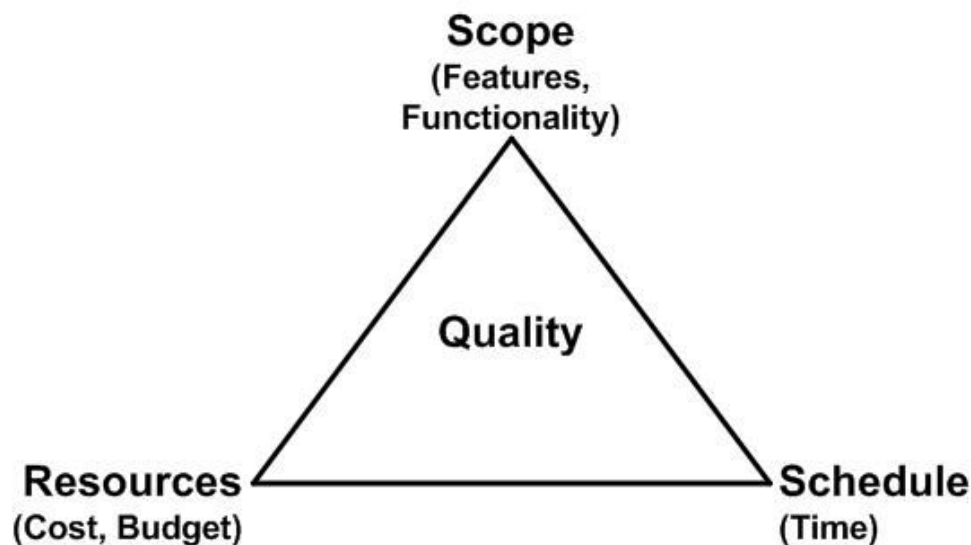
#### 2.1.1 Rune Solberg som kvalitetsstandard

Uansett hvordan vi definerer kvalitet gjennom modeller, teorier og testing, så er det til syvende og sist produkteier som er den som bestemmer om ideen/funksjonen er av god kvalitet. Det er han som i ettertid skal bestemme om dette er godt nok for programvaren Screenz. Rune har informert oss om at designet ikke er viktig, det er funksjonen og brukervennligheten som skal stå i fokus. Han vil uansett i ettertid bruke sin kompetanse innen grafisk design/UX til å utforme et brukergrensesnitt som han er fornøyd med. Rune har sine metoder å teste Screenz på, for fra hans ståsted, så er det kundene som bestemmer om det er brukervennlig eller ikke. Hvis ikke kundene av Screenz er fornøyde og istedenfor bruker andre løsninger, så er heller ikke Rune fornøyd med resultatet. Han bruker mye tid på å teste løsningen før det oppgraderes og sette ut i ny versjon, alt fra fysiske brukertester, til Silverback 3 som er en programvare til brukertesting.

## 2.1.2 The Iron Triangle

I følge “The Iron Triangle” består kvalitet av en kombinasjon av tre elementer; ressurser, omfang(scope) og tid. (Project Management Triangle, 2018)

- Ressurser. I denne delen av triangelet inngår kostnad, budsjett og tilgjengelige ressurser. I vårt tilfelle har vi ikke noe budsjett eller kostnader, men vi er fem studenter, som er en ressurs i seg selv.
- Omfang/funksjonalitet. Omfang, eller “Scope” går på spesifisere kravspesifikasjoner for at sluttproduktet skal bli bra nok.
- Tid. Tid går rett og slett på hvor mye tid vi har til rådighet. I tillegg er det tidsfrister underveis som vi har måttet forholde oss til.



Copyright 2003-2006 Scott W. Ambler

Figur 1: “The Iron Triangle”

Fra vårt perspektiv så kan vi sette opp de 3 elementene slik:

- Ressurs: vi har ikke noe budsjett i form av penger i vårt prosjekt. Men vi regner oss selv, vår kompetanse og kunnskap, samt våre private PC'er som ressurser. I tillegg har vi hatt Coworx og produkteier som tilgjengelig ressurs. Alle hjelpemidler som UiA kan stille opp med er også ressurser, som grupperom, Lynda.com og samt all informasjon vi har fått gjennom hele studiet de siste 3 år.
- Omfang: størrelsen eller omfanget på prosjektet har blitt snevret inn i løpet av bachelorprosjektet. Både grunnet at man tror man klarer mer enn man faktisk gjør i løpet av den tiden man har til rådighet, men også nedkutting av funksjoner da man alt for ofte gaper over for mye, og glemmer at man bør ta utgangspunkt i bare de aller mest nødvendige funksjonalitetene sluttproduktet må ha for å kunne fungere. Her har også tilbakemeldinger fra produkteier vært med på å bestemme hvilket omfang sluttproduktet skal omfange.

- Tid: vi har i teorien 2, kanskje til og med 3 tidspunkter som konkrete mål. Første er innlevering av rapporten, 16 mai. Deretter er det 23 mai, hvor visning av produktet skal gjennomføres. Til slutt så er det eksamen, 5 juni. Karaktermessig så er det innlevering av rapport og eksamen som har noe innvirkning på karakteren. I tillegg kan det nevnes at vårt andre fag i dette semesteret har vært en stor tidstyv.

## 2.2 Scrum

For å sikre oss at prosesskvaliteten skulle være så bra som mulig så brukte vi Scrum som rammeverk for å organisere arbeid. Vi valgte å jobbe med scrum fordi det er en agil metode som gjør det mulig for oss og gå tilbake til tidligere oppgaver og gjøre endringer. Det er muligheten til å gjøre endringer når det er behov for det, uansett hvor langt man er kommet i utviklingsprosessen, som gjør at scrum er den beste metoden å bruke etter vår mening. Fordi man jobber i korte iterasjoner, får man mulighet til å gi tilbakemeldinger på hva som fungerte den forrige sprinten, og gjøre justeringer i neste sprint dersom det var noe som ikke fungerte like bra. I Scrum har man en backlog hvor alle oppgaver man har blir lagret. I hver sprint henter man oppgaver fra backlogen utifra hvor mye tid man har beregnet på hver oppgave, og når det bør implementeres/utføres.

### Tidsestimat

En av utfordringene vi hadde med å bruke Scrum er at det kan være vanskelig å estimere hvor mye tid de ulike oppgavene tar. Det er bedre å ta med litt for få oppgaver i hver sprint, enn for mange. Dette er fordi at dersom man da skulle bli ferdig kan man legge til flere oppgaver fra backlogen. Hvis man har for mange oppgaver igjen når sprinten er over, må man ta de med seg i neste sprint i tillegg til at man "bryter det man har lovet" i sprinten.

Det har vært noe variasjon i hvordan vi har estimert oppgavene, men slik vi endte opp med å estimere oppgavene fra sprint 4 og utover var at vi ga hvert backlogitem i en sprint backlog en prioritet i form av 1-5 stjerner i AirTable, og at hvert item i sprintbackloggen hadde et tidsestimat i form av timer. Vi har ikke tatt i bruk noen av de mer anerkjente teknikkene, som f.eks Planning poker, til våre estimeringer. Vi føler ikke at vår prosess har vært så veldig annerledes av den grunn, da slik det fungerer nå er det en todelt prosess der vi først, som nevnt, prioriterer i stjerner, og deretter i form av timer. Er det uenighet i antall stjerner eller antall timer så skal man forsvare meningen sin, og det har ofte ført til diskusjon rundt scopet av det som skal gjøres i backlogen og lar oss komme til enighet om ting som kunne skapt konflikt i ettertid.

Et viktig element som vi mangler fra planning poker er at vi ikke meddeler våre estimater "samtidig". En av oss spør løst "hvor lang tid tar denne oppgaven?" og de som ønsker å dele sine meninger gjør det. Det gjør gjerne at 2 av oss 5 i gruppa utdyper våre meninger og at vi da risikerer at resten henger seg på uten å uttrykke meningen sin i det hele tatt. Dette bygger på idéen om at det er en svært vanlig tankefeil at idéene som fremlegges først har størst tyngde og alt som kommer etter ses i relasjon til den første idéen (Anchoring, 2018) . I etterkant ser vi at vi kunne løst dette på en bedre måte, da gjerne med at alle legger frem hva de tenker er en god stjerne-rating eller tidsestimat *samtidig* slik at man kan minimere innflytelsen på hverandres estimater.

Tidsestimatene bruker vi da til å finne ut hvor mye arbeid vi kan få gjort i en sprint, der vi ser på det totale antall timer vi har til rådighet i gruppen og prøver å fylle opp disse timene med oppgaver fra backlog som dras inn i sprintbacklogen.

### **Scrum vs Waterfall**

Grunnen til at vi valgte en agil metode over en ikke-agil metode som for eksempel waterfall er fordi der gjør man alle stegene trinnvis, og har ikke mulighet til å gå tilbake til et tidligere trinn uten å begynne alle de neste trinnene på nytt. Dette gir oss dårligere fleksibilitet til å gjøre endringer når de eventuelt skulle trenge.

## **2.3 Kvalitet i vårt prosjekt**

### **Generell oversikt og fremgang**

For å sikre at både prosesskvaliteten og produktkvaliteten skal bli så god som mulig har vi tatt i bruk flere metoder for dette. Vi har først å fremst hele tiden hatt en generell plan over hele perioden med milepæler. Denne planen har inneholdt viktige datoer for når vi skal ha statusmøter, innleveringsfrister og eksamensdato.

### **Statusmøter**

Statusmøtene har hjulpet oss med tanke på hvordan vi burde gjøre endringer i fremgangsmåten vår i forhold til hvordan vi har jobbet fram til møte. På et av møtene fikk vi råd om at vi burde begynne å skrive dagbok for å få en oversikt over hva vi har gjort hver dag. Dette har vi prøvd å følge opp gjennom hele perioden, men har ikke vært like flinke hele perioden. Spesielt i starten ble ikke dagboken brukt konsekvent. Dette ble det dog en bedring på etterhvert.

#### **2.3.1 Verktøy**

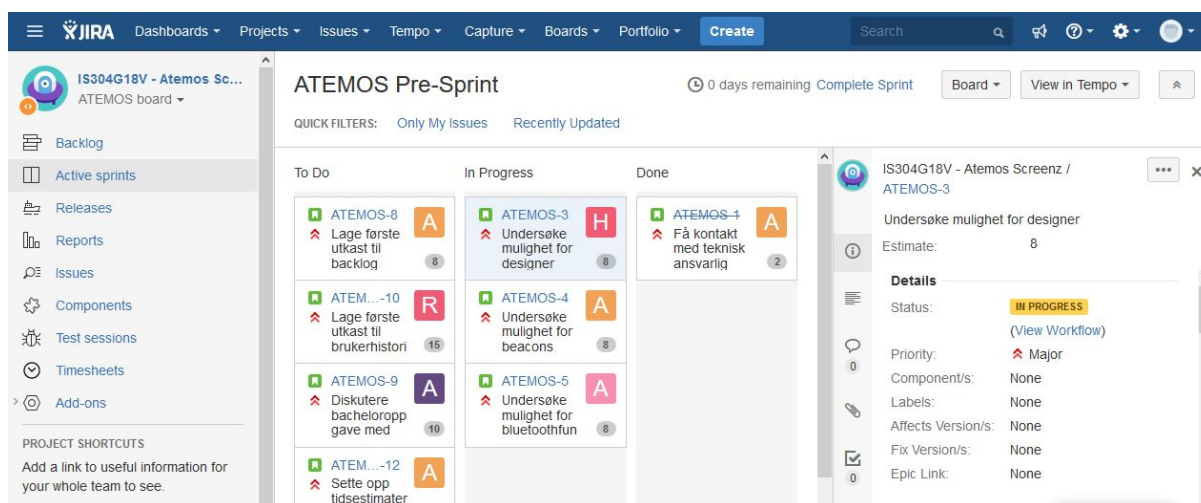
Vi har brukt flere forskjellige verktøy i form av programvare og fysiske artefakter for å gjennomføre dette prosjektet.

## Rapportskriving og generell tekstdokumentasjon

Alt av rapportskrivning, møterefater og generell dokumentasjon har skjedd i Google docs. Dette er fordi vi anser dette som den beste tekstbehandleren hvor man kan dele dokumentene og skrive i sanntid, samtidig som andre. Vi har delt mappe og undermapper på Google Drive, hvor alle dokumenter ble lagret, inkludert bilder, powerpointer osv. Tidlig i prosessen ga vi også veileder Janis Gailis og Rune Solberg lesetilgang til mappen vår i Drive, så de skulle få oversikt over hva vi holdt på med.

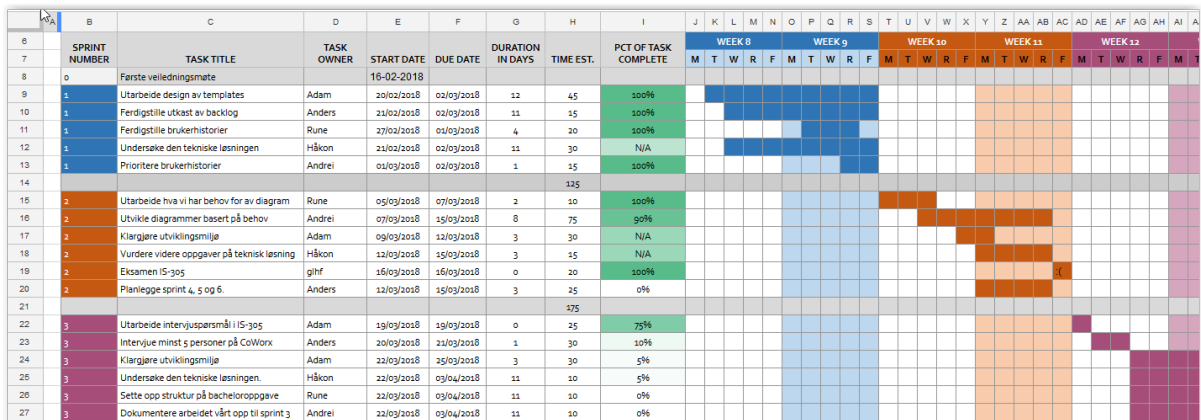
## Prosjektstyringsverktøy

Vi oppdaget tidlig i semesteret at UiA tilbyr Jira, som er et prosjektstyringsverktøy. Grunnet at vi kjenner til at flere bedrifter bruker Jira, så ønsket vi å benytte oss av denne tjenesten, rett og slett for å kunne ha nytte av det når vi senere skal ut i jobb. Vi prøvde å bruke Jira i begynnelsen, men gikk etter første veiledningsmøte over til Gantt-chart, som er et excel dokument, etter råd fra veileder. Jira klarte vi aldri å bli helt kloke på, og det endte opp med at vi ikke brukte det slik det var ment for å brukes, dvs det var ikke til hjelp for å gjennomføre prosjektet for oss. Gantt-chart fungerte greit, men da vi etter en stund kom over Airtable, som er en webtjeneste lignende som Jira, synes vi det var mer brukervennlig, og hadde funksjonene vi var ute etter. Også i Airtable har Rune og Janis hatt tilgang til å se hvilke endringer vi har gjort, og hvilke backlog-items vi har tatt med i sprintene.

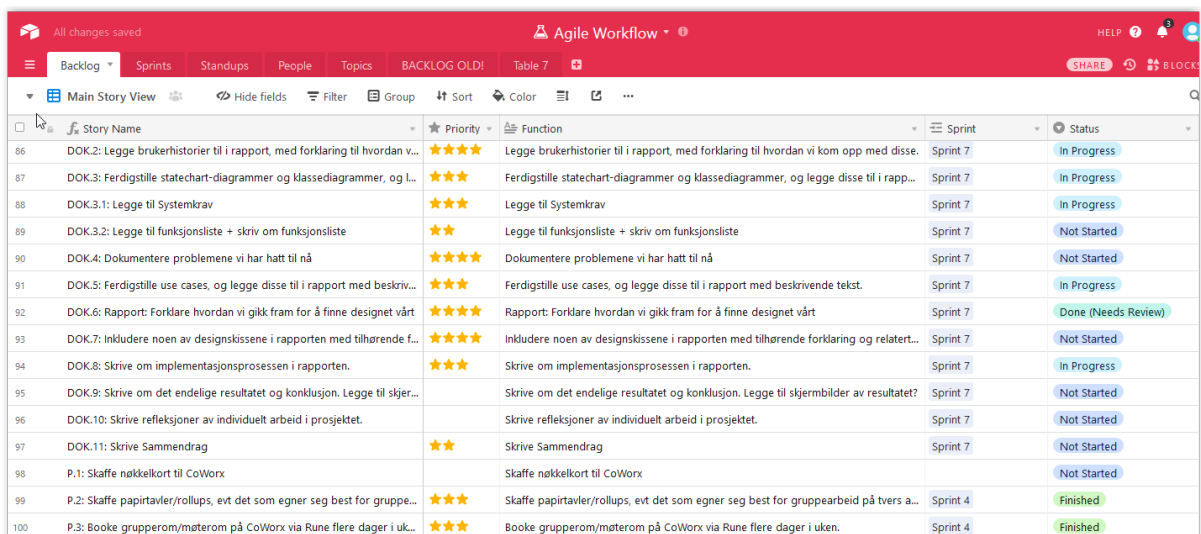


Figur 2: skjermbilde av Jira





Figur 3: skjermbilde av Gantt-chart



Figur 4: skjermbilde av AirTable

## Prototypeverktøy

Når det kom til utviklingen av prototypen vår brukte vi først InVision. I InVision laster man opp bilder og kan lage "usynlige knapper" ved å markere områder som linker frem og tilbake fra bilde til bilde. Slik kan man lage en fungerende prototype. Det er også muligheter for kommentarer i InVision ved at man setter inn punkter som inneholder kommentarer i prototypen, og det brukte vi en god del til å komme frem til bedre designløsninger.



Figur 5: Kommentar i InVision.

Senere fikk vi råd fra Rune at vi kunne bruke Figma, som er et bilderedigeringsverktøy som tillater samarbeid i sanntid, i tillegg til funksjonene som også er tilgjengelige i InVision. Det var Figma vi brukte da vi senere utførte brukertester.

### 2.3.2 Kommunikasjon

#### **Produkteier som ressurs**

Den kanskje viktigste ressursen vi har brukt for å sikre kvaliteten i prosjektet er kommunikasjon med Rune Solberg. Han kjenner domenet og er til slutt den som bestemmer om produktet eller funksjonene er ok eller ikke. Vi har hatt møter hvor vi har tatt opp brukerhistorier, designskisser og generelle spørsmål om oppgaven med

Rune for å få hans godkjenning. I tillegg har han kommet med sine innspill på hvordan ting burde vært gjort annerledes, i tillegg til å foreslå hva vårt neste steg bør være.

### **Veileder som ressurs**

Som veileder har vi har Janis Gailis. Janis har vi alle tidligere hatt som foreleser, i for eksempel IS-104 Brukergrensesnitt og IS-105 Datakommunikasjon og Operativsystem, samt diverse valgfag. Janis har siden høsten 2017 hatt tilholdssted i Wales, Storbritannia, der han har hatt jobb som trener for landslaget i roing. Vi har derfor måtte hatt alle veiledningsmøte via Google.Meet. Disse har blitt gjennomført på CoWorx. I tillegg har vi hatt fysisk møte med Janis på UiA en gang, ca midt i semesteret. All annen kommunikasjon har blitt gjennomført via mail, bortsett fra et par ganger via LinkedIn.

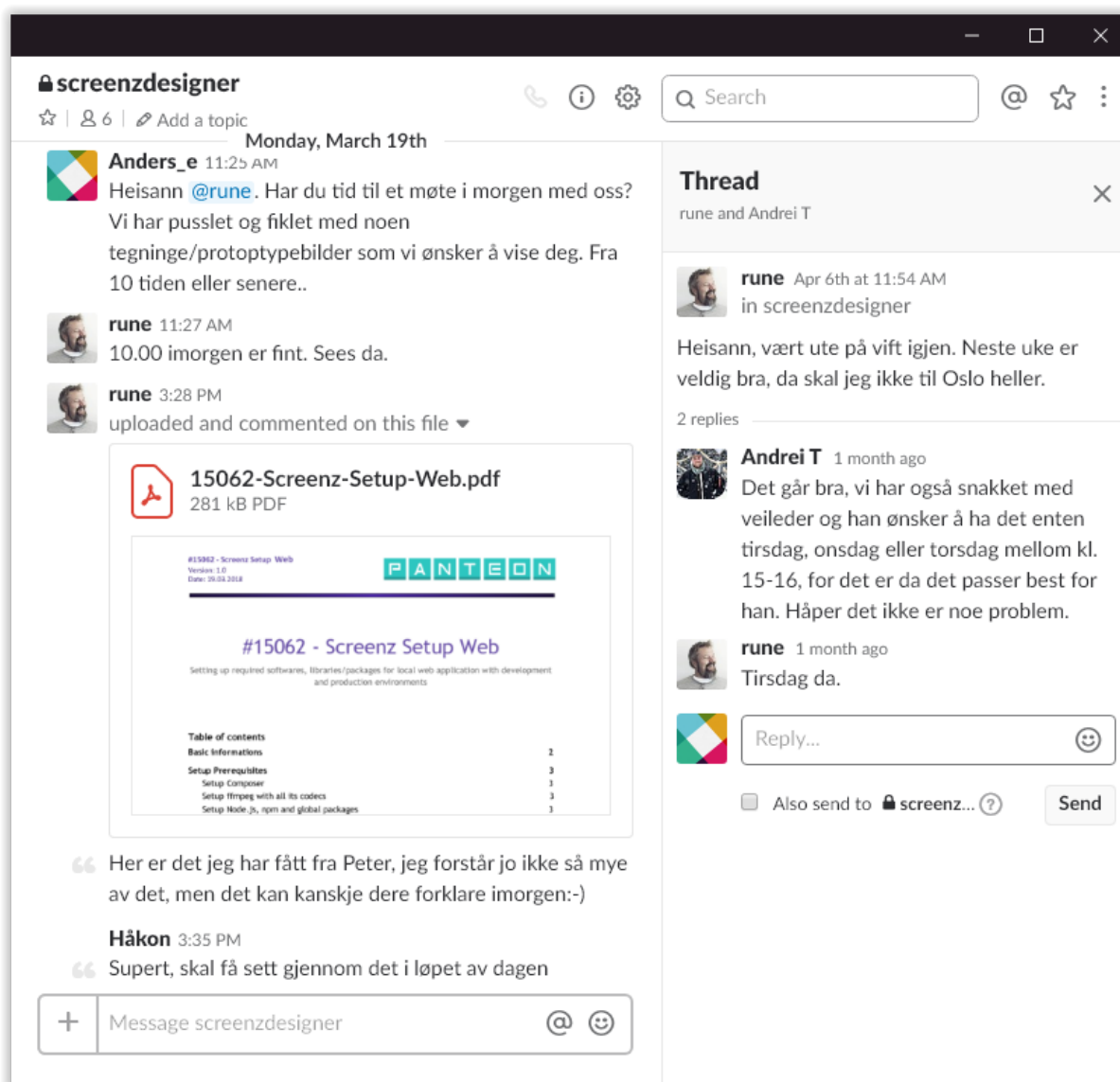
### **Panteon - ekstern utvikler**

Vi har også hatt et møte med utvikleren Rune Solberg bruker, Peter Hamre, for å få en bedre forståelse av det tekniske aspektet med oppgaven. Peter er entreprenøren bak, samt daglig leder for, firmaet Panteon AS. Han er en veldig travel mann, og har vært vanskelig å få tak i utenom dette ene møte. Vi har hatt litt kontakt på epost, men svarene har ikke alltid vært like klare. I tillegg så har det ofte tatt lang tid før Petter har besvarte våre eposter. I utgangspunktet skulle vi for eksempel fått kildekode av han 20.februar, men forskjellige årsaker fikk vi den ikke før 20.mars. Dette skjedde til tross for flere purringer via epost. Alle i gruppa fikk kopi av epostene som ble sendt ut som var relatert til prosjektet vårt. Denne utfordringen vi opplevde her står mer utfyllende beskrevet i 5.2 Oppsett av foreliggende prosjekt og 7.1 på Diskusjon/refleksjon.

### **Kommunikasjon med Rune Solberg**

Utover avtalte møter, har vi ofte jobbet på CoWorx. Da har det vært enkelt for oss å kontakte Rune når vi lurte på et eller annet. Vi har også hatt kontakt med Rune gjennom Slack, hvor vi har sendt forespørsler som nødvendigvis ikke har hastet så mye. Slack er et kommunikasjonsprogram hvor man blant annet kan chatte sammen i grupper. Rune opprettet gruppen Screenzdesigner, hvor alle oss i gruppen Atemos Screenz og Rune kan chatte sammen i en lukket gruppe.

Slack har mange flere funksjoner enn bare chat-funksjonen, men vi har ikke benyttet oss av flere funksjoner.



Figur 6: skjermbilde fra Slack

### Kommunikasjon internt på gruppe Atemos

Kommunikasjonen mellom oss i gruppa har stort sett foregått på Facebook-Messenger. Her har vi enkelt fått kontakt med hverandre dersom det skulle være noe. I tillegg har vi brukt mail, men stort sett bare hvis vi ønsket å snakke med eksterne parter og samtidig ønsket å informere resten av gruppa.

### 2.3.3 Gruppearbeid

#### Gruppearbeid - oppmøte

I starten møttes vi ca. 3-4 ganger i uka, men de siste 2 månedene har vi møttes 5 dager i uka fordi vi så at vi jobbet mer effektivt når vi satt sammen. Frem til mars så hadde vi i tillegg IS-305 Forvaltning av IT-ressurser som gikk parallelt med IS-304

Bacheloroppgave i informasjonssystemer. Dette tok mye tid, så vi fikk dessverre mindre tid til å bruke på bachelorprosjektet i denne perioden. Etter at eksamen i IS-305 var gjennomført fikk vi konsentrert oss mye mer om IS-304.

### **Gruppearbeid - arbeid med oppgaver og backlog**

I og med at vi har kjørt prosjektet vårt med Scrum som rammeverk, har vi operert med en backlog. Dette har gjort at vi hele tiden har hatt en liste med oppgaver som vi skal gjennomføre. Når vi har gjennomført oppgaver fra backlogen, har vi fått minst en annen i gruppa til å se over om oppgaven er utført tilfredsstillende nok. På denne måten får vi kvalitetsikret oppgaven.

#### 2.3.4 Testing av prototype

Når det kommer til å sikre at produktkvaliteten skal være av høy nok kvalitet, så er det viktig å utføre testing. For å sikre oss at designet vårt var brukervennlig og forståelig nok, utførte vi brukertester. Vi lagde da en prototype i Figma, hvor det er mulighet for å navigere mellom skjermbilder vi har laget. Dette diskuteres videre i 4.5 Brukertesting. Vi har også brukerhistorier med akseptanskriterier og use-cases som kan brukes til testing. Vi har ikke kommet langt nok i implementasjonsarbeidet til at vi har tenkt særlig på testing av software, og det har til dels blitt nedprioritert til fordel for funksjonalitet mtp. vår erfaring med Panteon og å starte fra bunnen av på nytt prosjekt (se kap. 5.2 og 7.1)

#### 2.3.5 Føring av dagbok

Vi fikk tips fra vår veileder at vi burde føre dagbok. Dette skulle hjelpe oss å huske hva som ble gjort de forskjellige dagene og gjøre det lettere for oss å kunne dra det inn i rapporten på slutten av semesteret.

Da rapporten etterhvert ble mer og mer prioritert, så vi at dagbok var gull verdt. Selv om det ikke bare var å kopiere teksten inn i rapporten, så var det nå relativt enkelt å skrive hva som skjedde til hvilke tider. Det gjaldt også underveis når vi lurte litt på hva vi hadde blitt enige om, hva vi hadde diskutert og for eksempel hvilke ting vi hadde bestemt oss for å nedprioritere underveis. Også veiledningsmøter og sprintevalueringer har blitt dokumentert på lik linje som dagbøkene.

I begynnelsen var vi ikke flinke nok til å dokumentere faktisk arbeid som ble gjort de gangene vi møtes, men da vi begynte å skrive dagbok for hver dag ble dette litt mer oversiktlig. Men på grunn av vi ikke var vant med å skrive dagbok var det en del dager hvor det ikke ble ført noe. Arbeidet ble gjort men ikke dokumentert skriftlig.

## 3. ANALYSE

I denne delen av rapporten samlet vi inn data for å få en bedre forståelse av hva vi trenger å implementere i systemet vårt. Våre analyser kom også godt med da vi senere skulle begynne på designfasen.

### 3.1 Brukerhistorier

Vi har valgt å utarbeide brukerhistorier for å få et bedre bilde av hva vi trenger for at prosjektet vårt skal bli vellykket med tanke på funksjonalitet og brukervennlighet. Utarbeidingen av brukerhistoriene skjedde først ved at vi diskuterte med Rune Solberg om hvilke funksjoner han ønsket seg. Etter vi hadde diskutert med Rune, gikk vi sammen og kom fram til noen brukerhistorier innad i gruppa. Når vi hadde utarbeidet dem ferdig, avtalte vi et møte med Rune for å få hans godkjenning på disse.

Vi har laget brukerhistorier for brukere som skal bruke systemet for å produsere et view. Vi har også laget noen brukerhistorier for designeren som skal legge til templates.

#### 3.1.1 Prioritering/MoSCoW

Vi har prioritert brukerhistoriene med tanke på hvor kritisk funksjonalitet de har. For prioriterte brukerhistoriene har vi valgt å bruke MoSCoW-metoden. Denne metoden er delt i 4; Must, Should, Could, Won't.(Haughey, 2014)

Prioritet	Forklaring
<b>Must</b>	Definerer et behov hos brukeren som må (must) oppfylles for at systemets egenskap skal bli fullstendig. Her ligger også valg av funksjonalitet som påvirker at lover og regler blir fulgt, men som i seg selv ikke gir økonomiske fortrinn.
<b>Should</b>	Definerer et behov hos brukeren som bør (should) oppfylles hvis tidsrammene tillater det for at systemet skal bli mest mulig optimalt.
<b>Could</b>	Definerer et behov hos brukeren som kan (could) oppfylles hvis tidsrammene tillater det, men er ikke kritisk for systemets egenskap. Dette er behov som kan beskrives som "kjekt å ha" i systemet.

<b>Won't</b>	Definerer et behov som ikke trenger å være til stede for at systemets egenskaper skal kunne utføres og blir ikke implementert grunnet tidsrammer eller andre årsaker.
--------------	---

### 3.1.2 Eksempler på Brukerhistorier

Her følger noen eksempler på våre brukerhistorier. For en komplett liste, referer til vedlegg 3

#### **1 Som designer**

##### **Ønsker jeg å kunne laste opp templates**

**Slik at mine kunder kan ta i bruk mine designs i Screenz.**

Beskrivelse: Målet med designeren er at kunder skal kunne benytte slike templates til å forhåpentligvis enkelt kunne opprette egne skjermbilder av høy kvalitet uten ekstern hjelp fra en kvalifisert designer. Disse templatene vil i første omgang bli laget av en kvalifisert designer. Målet er å konstruere dette på en slik måte at en designer vet hvordan han kan generere en template og at den kan lastes opp i tjenesten vår.

Prioritering: Won't have.

Grunnen til dette er at vi kan forøvrig bare implementere disse vha. Hardkoding i første omgang frem til vi ser at bruk av templates fungerer i praksis.

Akseptansekriterier: Designer skal enten være en egen brukergruppe i systemet, eller så skal enhver bruker laste opp en template. Systemet må kunne tolke slike filer slik at det samles sammen til en template som kan refereres til i systemet og konverteres til et bilde.

#### **4 Som bruker**

##### **Ønsker jeg å kunne laste opp et bilde til designeren.**

**Slik at jeg kan bruke det i templatene.**

Beskrivelse: Brukere av screenz vil gjerne lage designet selv, og må da kunne laste opp bilder for å lage designet som skal vises på skjermen.

Prioritering: Must have - Dette er en kjernefunksjon i systemet og må være til stede for at det skal være noe verdi.

Akseptansekriterier: Funksjonen er akseptert når brukeren kan laste opp et bilde og benytte seg av dette i designeren.

## **8 Som en bruker**

**Ønsker jeg å kunne velge templates når jeg går inn i designeren**

**Slik at jeg kan bestemme blant de templatene jeg ønsker å designe videre på**

Beskrivelse: Brukeren skal kunne bla igjennom de forskjellige templatene som er tilgjengelig og velge en av de han ønsker å ta i bruk.

Prioritering: Must Have - Vil være en viktig og essensiell del av MVP.

Akseptansekriterier: Når brukeren kan velge en template fra et utvalg av templates.

## **19 Som bruker**

**Ønsker jeg å lagre en template som jeg har redigert ferdig**

**Slik at jeg kan sette det opp på skjermene mine**

Beskrivelse: Når brukeren har redigert templatene slik at den ser ut slik som brukeren selv ønsker, skal det være mulig å lagre templatene. I denne prosessen vil brukeren måtte velge navn på den ferdigredigerte templatene før den kan lagres. Vi har også vurdert annen informasjon som kan fylles inn på dette stadiet. Tags kan brukes til sortering i senere tid, f.eks at man gir et ferdigsnekret design en tag med navn "Jul" slik at det enklere kan filtreres.

Prioritering: Must have.

Akseptansekriterier: Kravet oppfylles når brukeren kan lagre en template (Fra template til skjermbilde, eksportering?) og at systemet lagrer dette på en slik måte at man kan finne resultatet i andre deler av systemet, som når man skal definere hvilke media som skal legges inn i en spilleliste på en gitt skjerm. Mellomlagring er positivt, men forøvrig ikke et krav for akseptanse. Andre idéer, som f.eks bruk av tags, er heller ikke et krav for akseptanse.

## **3.2 Use-Cases**

En Use-Case (UC) er i motsetning til en brukerhistorie en mer detaljert beskrivelse av en prosess som brukeren går gjennom når man bruker systemet. Hvor en brukerhistorier fokuserer på selve funksjonen som brukeren ønsker i systemet, gir en Use-Case en mer oversiktlig og komplett syn på brukeropplevelsen og hva som må



til for å opprette brukeropplevelsen i hele systemet (front/back-end). En Use-Case tar hensyn til variasjoner i form av valgforgreninger eller bruker-/systemfeil og det blir spesifisert et nytt løp i UC dersom dette skjer i form av Alternate Courses og Exceptions.

Use case var noe vi tok i bruk etter råd fra Janis. Han hadde en tanke om at våre brukerhistorier gjerne var litt for vage, og at måten vi prioriterte på der og da (Gantt chart) baserte seg på diffuse enkeltsetninger som f.eks de som står på bildet under:

Utarbeide design av templates
Ferdigstille utkast av backlog
Ferdigstille brukerhistorier
Undersøke den tekniske løsningen
Prioritere brukerhistorier

Vi har ikke følt at vi har klart implementert en Use Case i implementasjonsarbeidet, men ved å lage Use Cases har vi vært nødt til å ta hensyn til f.eks errorbehandling og andre viktige elementer innen systemutvikling, og hvordan disse *faktisk* skal utarte seg i et teknisk system.

### 3.2.1 Eksempel på Use-case

#### **ID: UC04 - Bruker lagrer et ferdig tilpasset view i designerverktøyet.**

**Beskrivelse:** Når brukeren er ferdig i designeren skal det være mulig å lagre det ferdig tilpassede viewet slik at det kan brukes andre steder i systemet, f.eks legges inn i en aktiv playlist. Når brukeren trykker på lagre knappen skal det komme en bar hvor det konfirmeres for brukeren at viewet er lagret og at man skal kunne få muligheten til å navigere seg ut.

**Aktører:** Bruker (Screenz kunde)

**Antagelser:** Bruker er logget inn og har redigert ferdig et nytt view i designeren og ønsker å lagre det.

**Brukshyppighet:** Lav til middels lav.

**Triggers:** Brukeren trykker på save knappen i designeren.

**Pre-conditions:** Brukeren er ferdig med å lage et view i designeren.

**Post-conditions:** Det nye viewet skal bli lagret i databasen med navn og ID. Viewet skal vises i view menyen og kunne brukes i en aktiv spilleliste.

**Main Course:**

1. Bruker trykker på knapp for å lagre viewet.
2. Informasjon som navn, bilde og annet content brukeren har definert selv hentes ut fra variablene som brukes av systemet og skrives til et nytt objekt. (Se AC1)
3. Systemet kaller på back-end API og sender over objektet (Se EX1)
4. Server mottar objektet og kjører metode for å lagre informasjon i databasen og venter på konfirmasjon fra databasen. (Se EX2)
5. Det lagres i databasen og databasen sender konfirmasjon tilbake til server.
6. Server sender en melding til front-end om at viewet har blitt lagret.
7. I front-end dukker det opp en bar med konfirmasjon om at viewet er lagret med mulighet for videre navigasjon.

#### **Alternate Courses:**

**AC1** Dersom brukeren prøver å lagre det samme navnet som på et allerede eksisterende view.

1. Det skal komme en pop-up hvor brukeren har mulighet til å velge om en ønsker å lagre over et allerede eksisterende view eller lagre view under nytt navn.
2. Bruker velger å overskrive den eksisterende filen (Se AC2)
3. Gå til MC3

**AC2** Bruker velger nytt navn på overskrivende fil

1. Pop-up vindu åpnes med mulighet for å skrive et nytt navn.
2. Bruker trykker på ny lagreknapp definert i nye vinduet.
3. Gå til MC3

#### **Exceptions:**

**EX1** Systemet klarer ikke å lagre den ferdigredigerte templatens.

1. Try/Catch (eller tilsvarende i PHP) der catch blir aktivert pga mangler i/manglende fil eller annen error i systemet som f.eks nettverksbrudd/timeout
2. Log error i systemlog. Riktig error melding blir hentet basert på error code. Errormelding hentes fra en liste definert i koden med pregenererte meldinger basert på errorcode.
3. Systemet viser en error i form av en pop-up til brukeren i designverktøyet.

**EX2** Noen av verdiene som sendes over i objektet er av ugyldig type

1. Metode for lagring i database sjekker at informasjonen har rett type. I dette tilfellet er informasjonen feil.
2. Skriv ut errorkode basert på hvilket element som fikk feil under kjøring av metoden.
3. Send koden tilbake som svar på API-kall
4. Systemet skriver ut meldingen i en pop-up melding til bruker.

## 3.3 Kravspesifikasjon

For å komme opp med en kravspesifikasjon har vi valgt å bruke FACTOR. FACTOR består av Funksjonalitet, Applikasjonsdomene, forhold(Conditions), Teknologi, Objekter og ansvar (Responsibility). Vi har gjort det på denne måten for å få en

overordnet oversikt over hva systemet trenger for å fungere. Både hva det skal innholde, men også hvem som skal bruke det. (Mathiassen et al, 2000, s39-41)

- Funksjonalitet beskriver hvilke hovedfunksjoner systemet skal inneholde.
- Applikasjonsdomene beskriver hvem som administrerer, overvåker og bruker systemet.
- Forhold sier noe i forhold til hvilke forhold systemet vil bli utviklet og brukt i.
- Teknologi sier noe om hvilken teknologi som blir brukt i utviklingen, samt teknologien systemet skal kjøre på.
- Objekter beskriver hovedobjektene i vårt problemområde.
- Til slutt har vi "Ansvar" som forklarer systemets overordnede ansvar i forhold til konteksten den er satt i.

### **FACTOR for vårt system**

**F** - Redigere, og legge til views gjennom å bruke et designerverktøy

**A** - Ansatte i bedrifter som skal bruke designeren

**C** - Systemet skal kunne brukes av "hvem som helst" som jobber i en bedrift, uten at det skal være behov for spisset IT-kompetanse.

**T** - I utviklingen bruker vi Angular, PHP, LAMP - stack, node.js, html/css/js. Når det skal brukes bruker vi en PC og en eller flere monitorer.

**O** - Brukere, Templates, Views

**R** - Designverktøy for brukere av Screenz.

## **4. DESIGN**

Denne delen av rapporten forklarer hvordan vi gikk frem for å designe systemet på en hensiktsmessig måte. Vi har også bevisst valgt å ikke ta med noen designelementer med i rapporten, og dette vil forklares før underkapitlene.

Vi har sammen med veileder kommet frem til at det er noe mangler på artefakter fra design av det tekniske systemet, og da har særlig ER-diagram og klassediagram vært nevnt.

ER-diagram virket noe meningsløst da vi, slik vi tolket det, bare ville sette det opp slik at en bruker av Screenz og brukerens record i en databasetabell hadde en databasekobling opp mot en view-tabell som arver noe fra en template-tabell. Å utarbeidet dette fullt ut hadde vært mye mer aktuelt hvis vi hadde hatt god kommunikasjon med Panteon og deres utviklingsteam. Da hadde vi fokusert på å koble vårt system på deres eksisterende system, og det ville da vært interessant å sett hvordan våre systembehov hadde vært satt sammen i den eksisterende system- og databaseløsningen.

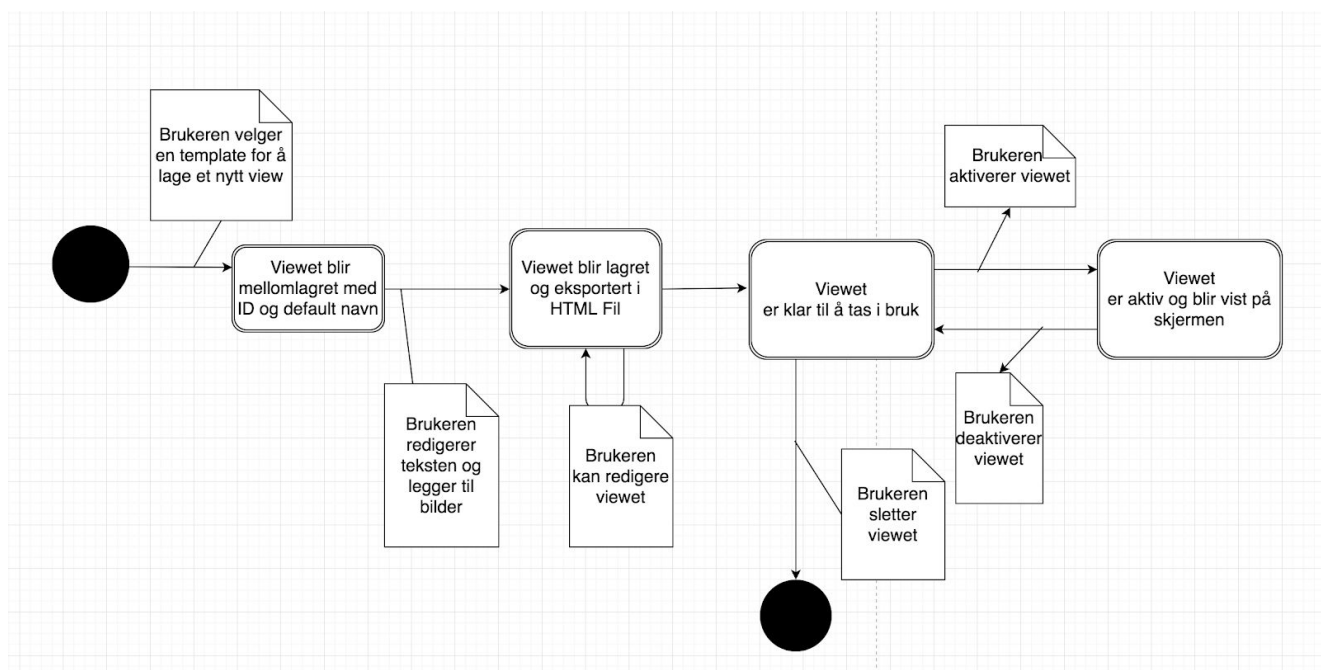
Det samme gjelder også et klassediagram. Siden vi ikke ville koble oss på den eksisterende løsningen og bare ville bygge en prototype av en mindre funksjon tenkte vi ikke at dette ville være hensiktsmessig. I retrospekt er det enklere å argumentere i mot et ER-diagram enn et klassediagram da vi ser for oss at et klassediagram kunne være mye mer nyttig for oss internt. Vi føler ofte at vi har en felles forståelse for det tekniske. Derimot ville det å hatt en felles mal vi utarbeidet oss i fra, slik vi ser det, vært nødvendig hvis prosjektet hadde vært større en vår lille del av det eksisterende systemet, enten det er en bokstavelig del eller en figurativ del.

## 4.1 State-chart

Et state-chart, eller tilstandsdiagram, viser oss hvilke tilstander et objekt kan være i. Tilstandsdiagrammet beskriver flyten fra én tilstand i systemet til en annen. Tilstander blir forklart som betingelser hvor et objekt eksisterer og hvordan den endrer seg dersom en handling blir utført.

Tilstandsdiagrammet under (Figur 7) skal illustrere funksjonen om å opprette et nytt view basert på en template. Diagrammet viser hele prosessen fra når brukeren går inn i systemet og velger en template som han/hun ønsker å jobbe med til det er ferdig redigert og klar til å tas i bruk. Diagrammet viser også de forskjellige mulighetene brukeren har i systemet.

Formålet med å lage state-charter under (Figur 7) var å gi oss et mer oversiktlig bildet over flyten til et view i systemet. Det hjalp oss med å se hvilke tilstander objektet template måtte gjennom før det ble om til view og fikk oss til å stille noen spørsmål i hvilken grad noe er "aktivt" hvis det er en fil som tas i bruk.



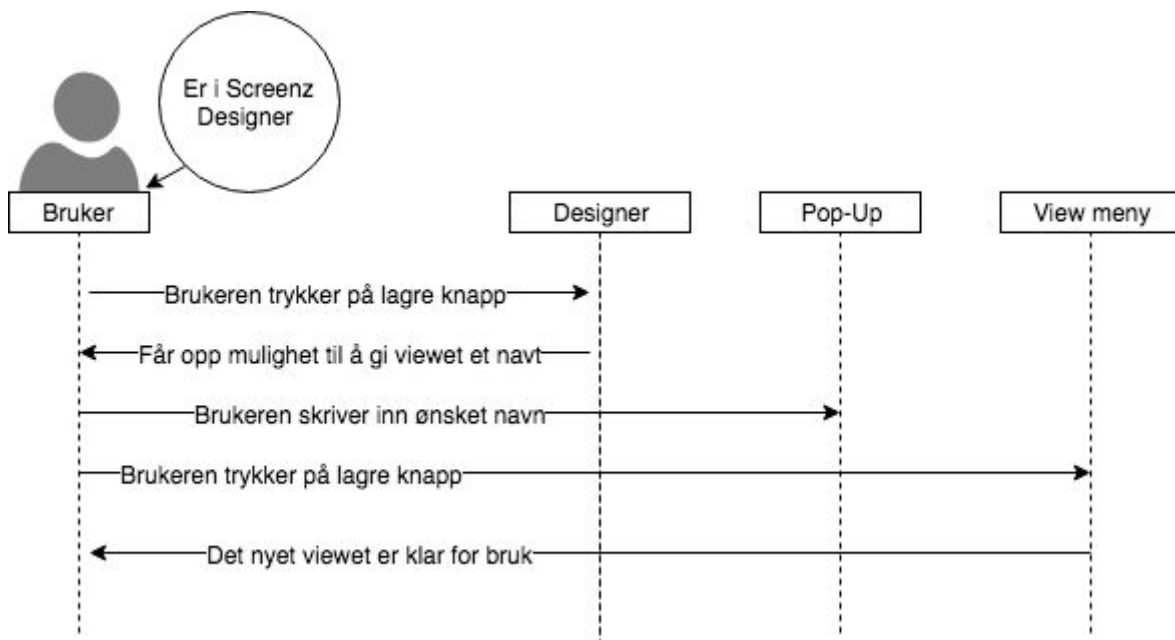
Figur 7: State-chart

I diskusjon med Rune hadde noen av oss tidligere fått nyss om at slik Screenz kjørte, så var det ikke bare en nettsjenerne men også en lokal klient som lastet ned mediafilene lokalt uten behov for nettverkstilkobling. Det betyr at filene lastes ned en gang og må resynkroniseres når noe oppdateres på nettsjenerne, og at når ting vises på skjermen så hentes de lokalt. I diagrammet har vi tenkt at når et view er aktivt, så blir viewet låst i databasen som en transaksjon slik at det ikke kan endres på. Det gir ikke noe mening når filene bare lastes ned lokalt og kjøres derfra. Dette kom frem under felles fremvisning og diskusjon rundt diagrammet. Skulle vi endret diagrammet, hadde vi gjerne sagt i loopen helt til høyre noe slik som “Media resynkroniseres og view lastes ned til klient på nytt”.

## 4.2 Sekvensdiagram

Sekvensdiagram viser oss hvordan objekter samhandler etter brukeren har interagert med et objekt. Et sekvensdiagram gir en systematisk oversikt over hvordan systemet fungerer i henhold til hva man ønsker med hvert museklikk eller tastetrykk.

Sekvensdiagrammet under (figur 8) er basert på brukerhistorie nummer 19. Den viser alle steg som brukeren må igjennom for å få ønsket utbytte. Funksjonen som brukeren ønsker er å kunne lagre en ferdig redigert template slik at den kan bli tatt i bruk på en skjerm som et view.



Figur 8: Sekvensdiagram

Brukeren trykker på lagre knappen, deretter gir systemet brukeren muligheten til å gi det nye viewet et navn. Navnet som brukeren skriver inn blir lagret. Deretter blir viewet vist i view menyen og den er klar til å bli tatt i bruk.

Grunnen til at dette sekvensdiagrammet er dratt frem er fordi dette sekvensdiagrammet skapte en diskusjon som førte til en designendring. Slik dette diagrammet er satt opp nå, og hvordan vi tolket designverktøyet når den ble laget, er det slik at man alltid navigerer seg ut fra designverktøyet etter brukeren lagrer sitt design. Her kom det frem på et møte at vi hadde glemt å tenke ut at en bruker gjerne hadde ønsket å lagre sitt design underveis og gjerne gjøre noe annet som f.eks å bytte nettleserfane for å gjøre noe annet som kommer opp. Det er et vanlig scenario at man må svare på en melding, svare en telefon eller tilsvarende. Da spekulerte vi at brukere gjerne ville ønsket å lagre for tryggheten sin del uten at dette skulle ulempelig for brukeren.

### 4.3 Designprinsipper

Når det kommer til designprinsipper velger vi å skille mellom det rent grafiske og det som omhandler grensesnitt og brukererfaring (UI og UX)

Når det kommer til det grafiske så er det meningen at det vi utvikler skal kunne implementeres i en senere versjon av systemet til Screenz. Det gjør at vi i praksis slipper å tenke en del på fargevalg og utseende på grafiske elementer da dette ofte

finnes fra før eller følger visse retningslinjer. En lilla fargepalett preger hele systemet, og det meste av knapper og andre elementer finnes jo fra før av i den aktive Screenz-tjenesten vi har fått tilgang til. Hele vårt design preges av dette.

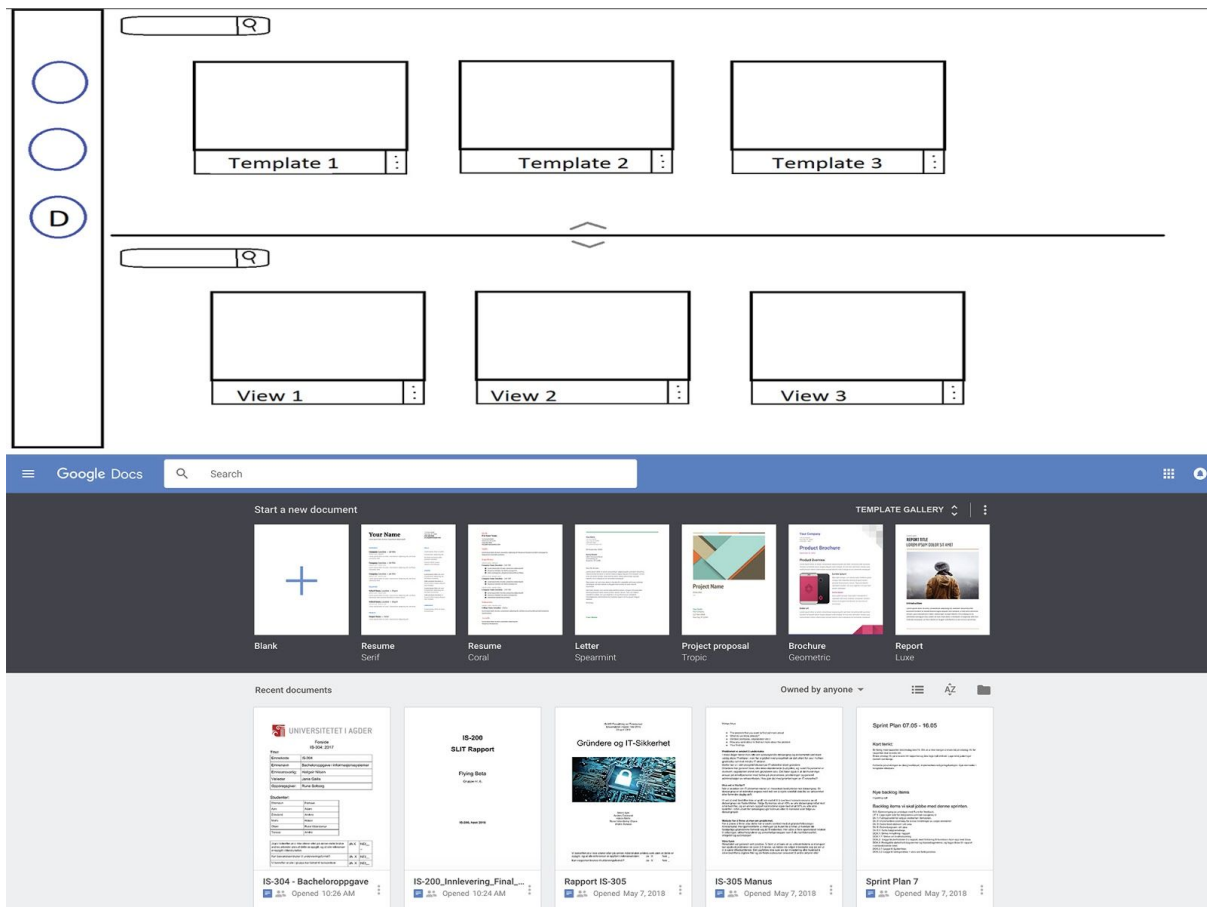
Det som er unntaket er selve designverktøyet. Vi har i dialog med Rune kommet frem til at selve designverktøyet skal skille seg fra resten av systemet ved bruk av et "dark theme" fremfor det lysere designet til resten av systemet. Ikke bare skal det være et figurativt skille, men det er noen som hevder at typisk lyst-på-mørkt design skaper mindre press på øynene over tid ("Light-on dark-color scheme", 2018) Det kan tenkes at man er særlig utsatt hvis man bruker øynene ekstra mye som en da gjerne gjør i et designverktøy.

Når det kommer til UI og UX så har mye av fokuset stått på å se på hvordan andre konkurrenter har løst samme problemstilling tidligere. I følge Rune så kan vi for det første vie vår tillit til at andre, langt større aktører har løst problemene på en funksjonell måte og at vi bare må tilpasse det vår målgruppe deretter. Det kan gjerne også gjøre det lettere for våre brukere å benytte seg av systemet hvis bruksmønsteret følger noe de allerede er kjent med. De vi har studert mest er Canva.com, Google Slides/Microsoft Powerpoint og Infoskjermen.no. Canva er et digitalt verktøy for å genere grafikk som postere, facebook-bannere og annet grafisk promomateriale som kan brukes til digital signage. Det er også flere som bruker vanlig powerpoint-presentasjoner til digital signage og det er grunnen til at slike tjenester har blitt studert. Til slutt har vi infoskjermen.no, som er en konkurrent til Screenz som selv har utviklet en tjeneste som tillater brukere å lage infoskjermer med noe eget design.

Vi brukte mye tid på å lage et design som skulle være skalerbart og brukbart fra alt til 5 templates til 100+ templates, men Rune var kritisk til dette. Han presiserte at hans egen erfaring var at det var bedre å ha flere gode designs for flere mengder data, enn å jage etter et design som følger prinsippet "one size fits all". Hans erfaring er at designet trolig vil fungerte dårligere totalt sett og at ting vil se veldig tomt ut i starten. Når vi ikke har planer om å ha noe mer enn 5 templates i starten så ville det trolig se lite appellerende ut. Det ville også være en enkel sak å bytte fra ett design til et annet hvis designet fantes fra før av og mengden ble overskredet eller omvendt. For eksempel kunne vi sagt at vi hadde 3 forskjellige designs, og de ville endret seg utifra hvor mange templates som var tilgjengelige i kategoriene 0-19 templates, 20-99 templates og 100+ templates. Siden vi ikke planlegger å lage mer enn 5 templates så har vi bestemt oss at MVP bare vil inkludere et design for 0-19 templates.

For å gi et konkret eksempel for noen av prinsippene i dette avsnittet kan vi sammenligne våre tidligere design med google sin valgprosess for

templates/tidligere filer, hvorfor vi gikk bort fra dette designet og heller valgte å følge noe som allerede eksisterte i Canva. Først følger en figur (figur 9) av template-valg i våre tidligere skisser i InVision satt ved siden av Google sin løsning som da var inspirasjonen:



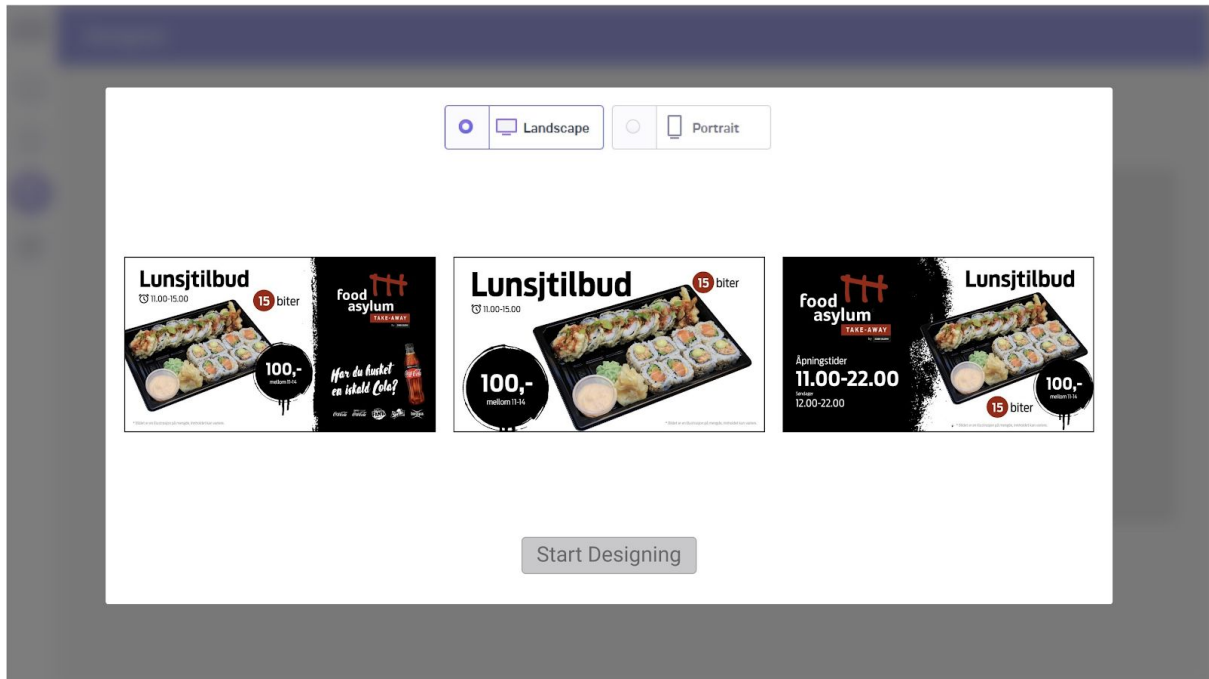
Figur 9: InVision vs google

Man ser nokså tydelig at vårt design preges av mønsteret google bruker, med valg av template øverst og valg av tidligere filer (views) under. Tanken her var at vi ville følge bruksmønsteret som allerede eksisterte i deres tjeneste og at vi da trodde vi kunne støtte flere hundre templates. Vi hadde også en idé om en “rullegardinpane” som kunne dras oppover og nedover slik at brukeren stod selv fritt til å velge hva skjermarealet skulle brukes til. Rune syntes ikke at rullegardinpane i midten på vårt design var noe god idé. Hans erfaring er at det er vanskelig å bruke og overbeviste oss om at vi burde finne en annen løsning. Han var heller ikke overbevist om at google sin løsning ville fungere bra på så veldig mange templates heller, og vi trakk samme konklusjon da vi så at google sin løsning ikke hadde noe filtrering av templates, men heller bare en løs gruppering med store thumbnails som ville vært upraktisk hvis antallet oversteg 100+.

Dette førte til at vi nå har to steder der man velger template i Screenz:



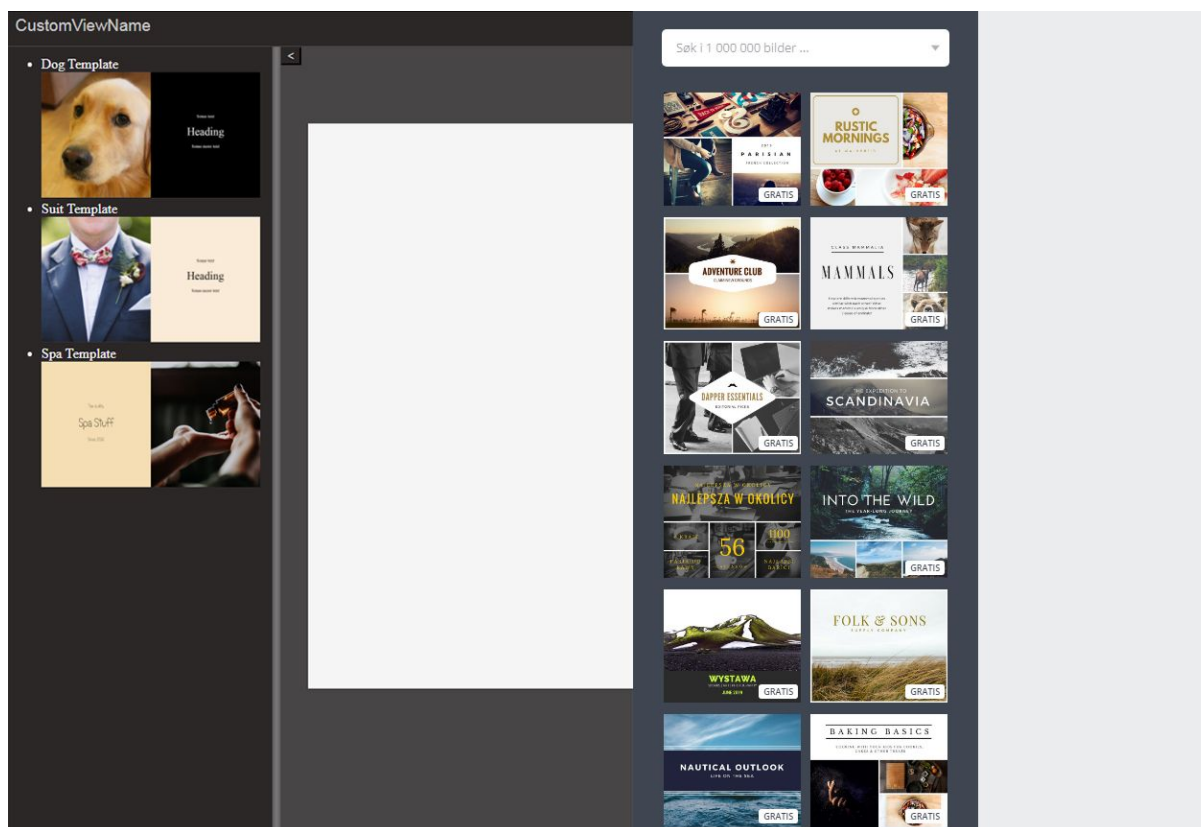
Først så kan man velge template når man velger å lage et nytt design men før man er navigert inn i designerverktøyet. Da vil man i vår prototype i Figma bli presentert følgende modale pop-up etter å ha trykket på knapp for å legge til et nytt design/view.



Figur 10: valg av template prototype figma

Målet i vår MVP er å kunne levere en håndfull templates, og da vil et slikt grensesnitt trolig fungere bedre enn det vi hadde i første prototype. Man velger simpelthen templatene man ønsker å bruke ved et tastetrykk og trykker seg deretter videre med “Start Designing”. Det er tenkt at det vil være forskjellige templates for forskjellig skjerm-orientering. Skulle det blitt mer templates over tid ville det krevd et nytt design etterhvert her med mindre man bestemmer seg for å ha noe form for “default templates” og da også en helt tom template som kan justeres senere.

Man kan også velge nye templates etter at man har blitt ført inn i designverktøyet, da vi så det som ulempe at man bare kunne velge dette på vei inn i designverktøyet: Hva om brukeren er usikker og vil eksperimentere med to eller tre forskjellige for å se hva brukeren synes passer best? I design-teori så brukes begrepet “safe exploration” for designavgjørelser som skal tillate slik bruk (Tidwell, J. 2010 s.37). Hadde vi bare hatt mulighet om å velge template i pop-upen som ble vist over måtte brukeren navigert seg ut av verktøyet og laget en ny fil for hver gang. Derfor bestemte vi oss for å ta i bruk en løsning som allerede finnes i Canva:



Figur 11: valg av template i screenz designer vs canva

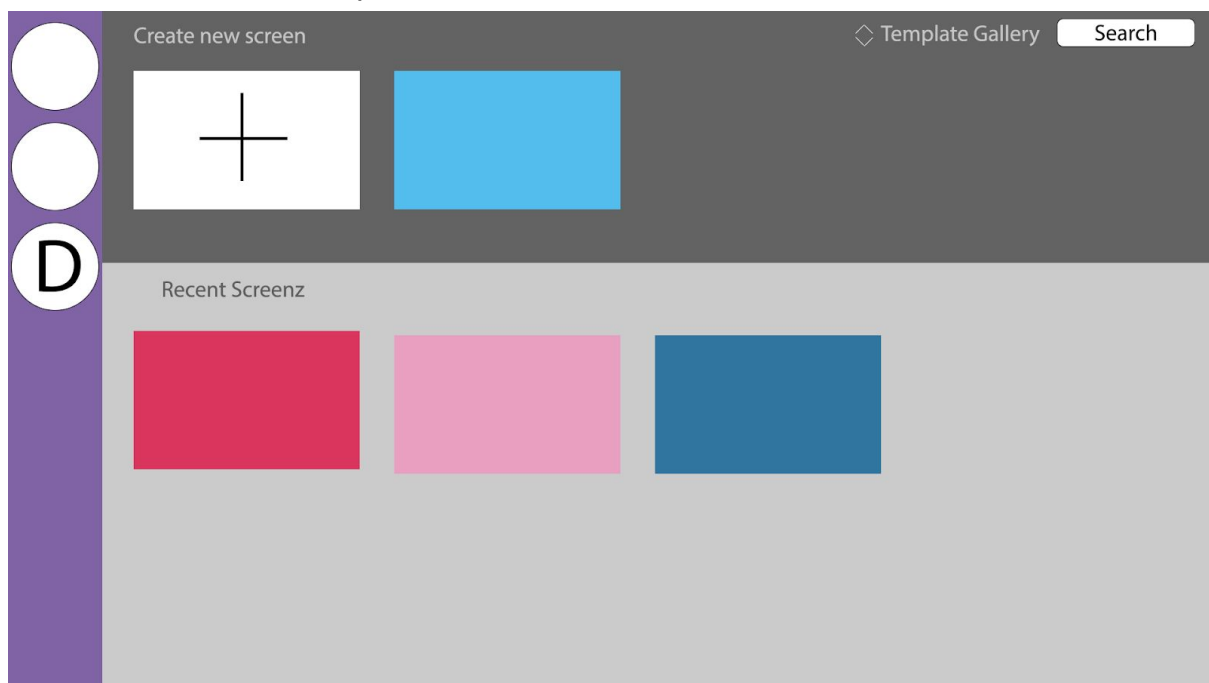
Til venstre på figur 11 ser man venstresiden av designverktøyet vårt. Her er det en egen fane der man kan velge aktiv template i designverktøyet ved tastetrykk. Til høyre ligger Canva sin allerede eksisterende løsning. Forøvrig er det ikke noe behov for noe søkemotor, og alle templates vil i første omgang være gratis, så dette har vi ikke tatt høyde for i vår MVP.

## 4.4 Design-skisser og prototyping

Når vi i starten skulle begynne å skissere ble dette ofte gjort igjennom diskusjon med Rune når vi satt sammen på CoWorx. Møtene var ofte slik at han viste oss hvordan eksisterende løsninger til konkurrenter fungerte i dag og hva han mente var problemene som brukerne støtte på, samt andre ting han ønsket å løse. Deretter begynte vi å skissere løsningsforslag. Det ble enten gjort individuelt på papir eller sammen på tavle for å diskutere. Rune tegnet opp flere eksempler på hvordan ting kunne bli seendes ut og hva som kunne føre til eventuelle problemer. Disse fikk vi tilsendt senere over Slack sånn at vi kunne ta det med videre i designprosessen. Når Rune ikke var tilgjengelig satt vi ofte å diskuterte designavgjørelser på universitetet der vi tegnet opp nye forslag eller idéer på krittavle. Alt i alt har disse møtene vært svært nyttige for det har vist seg flere ganger at vi tidlig i designfasen

ikke hadde samme forståelse for hvordan ting skulle fungere. Når alle i gruppen måtte tegne og forklare sin egen forståelse kunne vi fange opp i disse misforståelsene tidlig slik at vi ikke kom videre i designprosessen med vidt forskjellige idéer. Noe som har vært ulempelig er at vi ikke har diskutert “mange” idéer, men brukt svært mye tid på noen færre idéer. Det ble på et tidspunkt prøvd ut at alle skulle lage sitt eget forslag til menyen som kommer før designverktøyet, også skulle hver i gruppen vise det de kom frem til. Erfaringen her var at dette ble lite gjennomført og at vi med en gang gikk tilbake til å bare diskutere noen få idéer sammen på f.eks tavle. Det kan tenkes at vi har hatt en idé som utgangspunkt som ikke passet godt til vårt verktøy, men at det fremsto som beste alternativ da alternativene har vært fåtallige.

Vi gikk over til digitale prototyper veldig raskt da vi var for det meste bedre kjent med digitale verktøy og vurderte at vi ville spare lite tid på en papirprototype. Her var det opp til hver av oss individuelt hvilke verktøy vi brukte, og verktøyene har variert fra Microsoft Paint til Adobe Photoshop. Fokuset her var ikke på utseende, men på flyt i grensesnittet og eventuell brukertesting. Vi fikk også høre fra Janis under et veiledningsmøte at vi ikke burde bruke tid og krefter på en grafisk imponerende prototype: Dette mente Janis kunne føre til at brukere ikke var like villige til å gi kritikk på noe det virket som vi hadde brukt mye tid og krefter på, i frykt for å såre noen av oss eller bli sett på som “slem”.

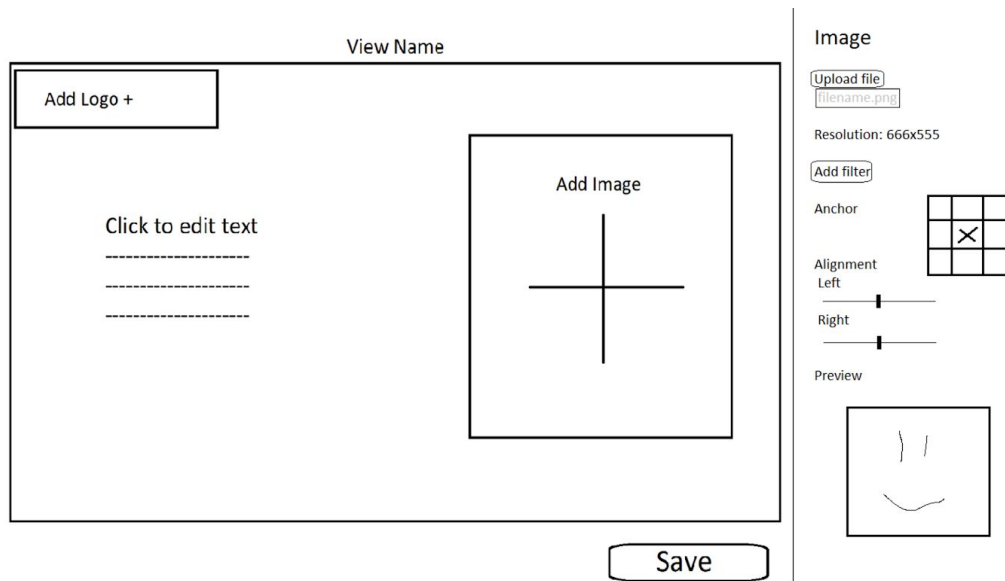


Figur 12: skisse for design laget i Photoshop



Figur 13: prototype for opplasting av bilde i Figma

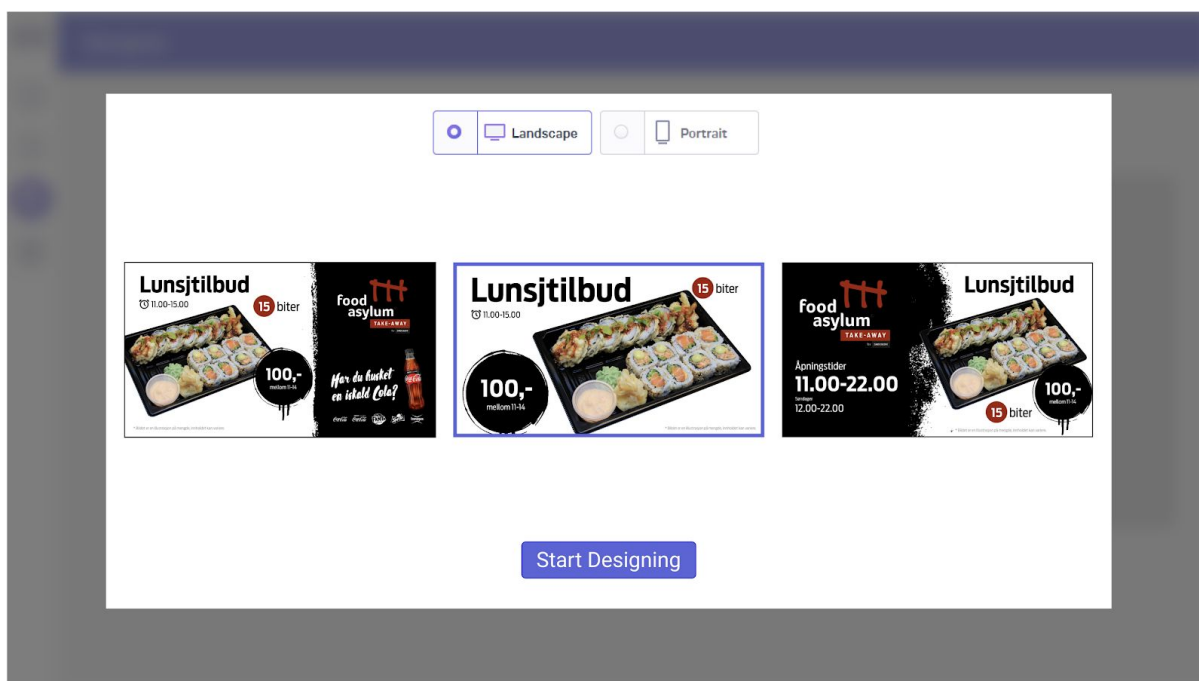
For vår første digitale prototype tok vi i bruk verktøyet InVision. InVision er et verktøy for å bygge digitale prototyper, der man kan laste opp egen grafikk og deretter legge til navigasjon vha. det de kaller for "hotspots". Hotspots lar deg trykke på et sted i ett bilde og flytte deg til et annet bilde. Med dette kan man lage en digital prototype med full navigasjon så lenge man bruker tid på å tegne hvert bilde i grensesnittet individuelt. InVision har også en funksjon for å legge inn punkter i hvert bilde med en kommentar. Dette brukte vi til å kommunisere eventuell kritikk og forbedringer til design med hverandre.



Figur 14 skisse for opplasting av bilde i Paint

Når vi hadde et ferdig utkast til prototypen i InVision tok vi dette med oss for å vise til Rune og gjøre klar til en evt brukertest. Her kom Rune med svært mye god feedback som gjorde at vi heller ønsket å lage et nytt utkast før vi testet noe på brukere. I tråd med dette byttet vi også til et nytt verktøy som ble anbefalt av Rune; designverktøyet Figma.

Figma er en nettleserbasert designløsning der man kan være flere som redigerer prototyper og design sammen i sanntid, a la google sine skyløsninger. Figma lar deg laste opp egne bilder slik som i InVision eller bruke verktøyene innad som et bilderedigeringsverktøy til å lage eget design. Her har vi for det meste brukt sistnevnte med unntak av små elementer som logoer og ikoner.



Figur 15: valg av template i prototypen i figma

Her brukte vi noe tid på å gjøre det grafisk tilfredsstillende da vi bemerket at vi kunne få utskrevet stil-kode i CSS av hvert element. Ulempen her er at vi gikk bort i fra Janis sitt tidligere råd om å holde det grafiske noe nedtonet i brukertesten. Det gjenstår å se om vi får særlig utbytte av CSS-funksjonen, men vi tror uansett det var greit for oss selv internt å ha designet noenlunde ferdig slik at vi slapp å tenke over dette i implementasjonen selv om det kan ha påvirket de videre brukertestene negativt.

## 4.5 Brukertesting

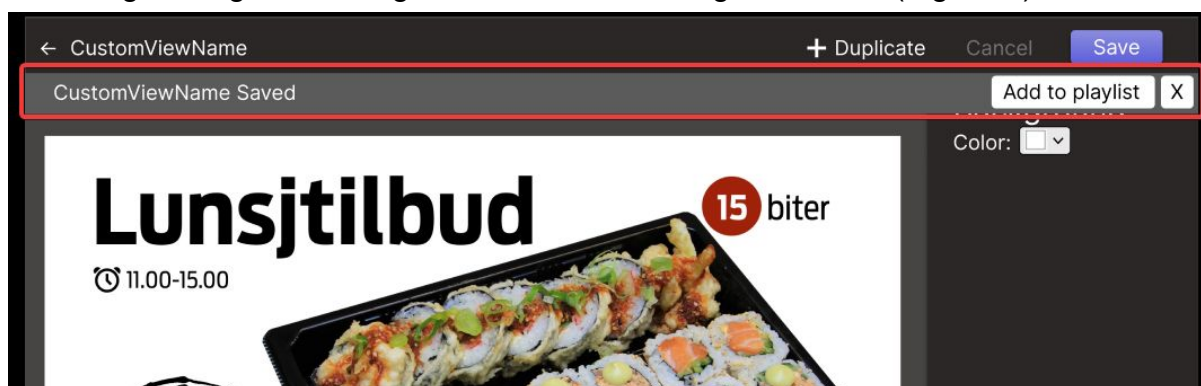
I vår brukertesting har vi gjort det som kalles for “Guerilla User Testing”. Enkelt og greit betyr dette at man tester tilfeldige brukere som ikke vet at de skal bli testet på forhånd og ikke er kjent med systemet (Simon P.D., 2017). Man observerer testbrukeren underveis og noterer hvordan de løser oppgaven, gjerne eventuelle problemer som oppstår.

Måten vi gjennomførte brukertestene på var at vi gikk to personer sammen bort til en tilfeldig person enten på CoWorx eller Value Accounting i håp om at de hadde tid og var villige til å teste systemet vårt. Vi ga dem så et scenario som de skulle gjennomføre i Figma-prototypen vår. De ble også bedt om å “tenke høyt” og forklare sine avgjørelser underveis. Deretter ga vi dem en ekstra “utfordring” for å teste om

de klarte å forstå programvaren på et mer abstrakt nivå enn det prototypen var lagt opp til.

Etter at vi var ferdige med brukertestene satt vi igjen med svært blandede resultater som gjorde det vanskelig å trekke noe god konklusjon. Folk gjorde ting mye forskjellige, men det var noen mindre detaljer vi følte vi kunne gjøre mer synlige eller kommunisere klarere.

Noen brukere begynte å trykke seg rundt uten å nødvendigvis scanne brukergrensesnittet. Flere av testbrukerene tok seg mindre enn 1 sekund med “scanning” på en ny side før de begynte å trykke på knapper og ting i grensesnittet. Dette har vi forstått som “satisficing” (Tidwell, J. 2010 s.39) innen design-teori: I vårt tilfelle var den knappen som da gjerne var mest tydelig “Save”-knappen, for vi erfarte at flere trykket på Save like etter at de hadde navigert seg inn i designverktøyet. Dette kunne vært et problem hvis save hadde f.eks vært “save & quit”, men dette hadde vi allerede tatt høyde for. I første omgang i tidligere designprosess tenkte vi at Save skulle navigere deg ut etter en popup men under diskusjon og utvikling av et sekvensdiagram kom vi frem til at flere brukere gjerne bare ville trykke save etterhvert som de designet for å være trygge på at de ikke mistet sitt tidligere arbeid. Derfor gjorde vi dette om til en liten bar på toppen med en konfirmasjon og mulighet til å navigere seg ut av designeren slik man ser i figuren under (Figur 15).



Figur 16: visning av saved-bar etter lagring av view

En av tingene som kom nokså tydelig frem var at språket som blir brukt innad i utviklingen, samt hvordan vi og Rune har definert elementnavn, ikke ble godt nok kommunisert i testen. Eksempelvis så kaller vi malene som brukes i designeren for templates, men ferdigredigerte templates for “views”. Når vi da ber testerene designe en plakat og det står “add view” i brukertesten så skaper dette forvirring hos brukeren. De som stoppet opp på dette steget ble spurt om det etter brukertesten var over, og selv om de forsto “template” så var “view” mye mer uklart. I prototypen som vi testet brukte vi dummytekst (Lorem Ipsum) på førstegangssiden for designeren. Her ville det vært mulig å forklare systemets egen sjargong for brukeren. View er ikke noe vanlig navn for det vi på norsk har kalt en plakat, og det burde vurderes om

det hadde vært bedre med et annet navn. Canva kaller det man lager for “designs”. Vi kunne vurdert det samme, evt. noe mer gjenkjennelig og folkelig som “poster” o.li.

En ulempe med måten vi testet på er at vi ikke sjekket bakgrunnen til de vi testet. Det gjør at vi ikke kan dømme teknisk kompetanse når det kommer til brukertestene, og derfor ikke trekke noen konklusjoner som f.eks at de med lavere teknisk kompetanse sliter med vårt system. Det kan godt hende at det tekniske nivået på Value Accounting og CoWorx er betraktelig mye høyere eller lavere enn de vi ser for oss som brukere, typisk butikksjefer o.li. Siden vi ikke kan si dette med sikkerhet så er det derimot også mulig at den tekniske bakgrunnen er mye *mer* variert enn vi selv tror. Vi tror også at sistnevnte faktisk er tilfellet mtp. mangfoldet av forskjellige resultater og tilbakemeldinger vi fikk av brukertesting.

Det må også nevnes at CoWorx og Value Accounting ikke ble valgt fordi de representerte forskjellige brukere i målgruppen vår, men fordi det var bekvemmelig på den ene dagen vi satt av til full brukertesting. CoWorx var der vi jobbet den dagen, og Value Accounting var i samme bygget som CoWorx. Når vi da hadde spurt alle som var tilgjengelige på CoWorx, gikk vi simpelthen inn på Value Accounting sine kontorer og spurte der til vi ikke hadde mer tid igjen og måtte dra hjemover. På CoWorx er bakgrunnene mye mer varierte, men Value Accounting stiller trolig noen større krav til en faglig bakgrunn og å kunne ta i bruk digital programvare. Dette kan være grunnen til at andelen som kom seg godt igjennom på Value Accounting var noe høyere, men ikke betraktelig med antallet brukertester vi hadde.

## 5. IMPLEMENTASJON

I denne delen av rapporten går vi gjennom hvordan vi har gått fram for å implementere løsningen vi har kommet opp med.

Programmeringsspråk som vi har brukt er HTML, CSS og TypeScript. Vi valgte å bruke WebStorm fra JetBrains som IDE siden vi hadde litt erfaring med det fra før. Vi har benyttet oss av GitHub for å samarbeide på koden og holde versjonskontroll.

### 5.1 Arkitektur

Det foreliggende prosjektet kjører Angular4 i front-end og Symfony/PHP7.1 i back-end. Vårt prosjekt kjører i hovedsak på Angular4 som er et rammeverk for javascript som benytter seg av TypeScript som programmeringsspråk. MySQL 5.7 brukes som databaseløsning, og i det foreliggende prosjektet til Screenz så kobles det hele sammen via et REST-basert API. Da dette er en prototype for å vise



frontend-funksjonalitet har vi ikke utviklet noen spesiell backend-løsning, men det ville sannsynlig også vært i samme rammeverk som det foreliggende prosjektet.

## 5.2 Oppsett av prosjekt med instruks fra Panteon

Sent i mars fikk vi et PDF-dokument med instruks på hvordan det eksisterende prosjektet til Screenz skulle settes opp. Når vi først leste gjennom dokumentet så dette veldig greit og rett fram ut, ut og vi antok at det ikke skulle ta veldig lang tid for oss å sette dette opp. Derfor ble dette noe utsatt i perioden rundt påskeferien da vi ikke anså dette som en stor risiko og ble ikke virkelig påbegynt en gang i første halvdel av april.

Etterhvert jobbet vi med å sette opp prosjektet etter instruksene vi fikk fra Panteon, og fulgte instruksene til punkt og prikke. Vi traff da på en del bugs under konfigureringen og oppsett av prosjektet, og ingen deler av prosjektet ville kompilere. Dette førte til at vi var nødt til å bruke tid på feilsøking av bugs. Vi har dokumentert over 100 timer med arbeid på bare feilsøking, og antar at det har vært mye mer med tanke på kommunikasjon vi har hatt med eksterne personer (venner og bekjente) for å prøve å løse problemet.

Problemet ser ut til å forekomme pga feil med noen node-modules og dependencies som Angular sin command-line interface (CLI) benytter seg av for å bygge prosjektet. Etterhvert som prosjektet bygges, kommer det advarsler om feil i dependencies og det hele avsluttes med feilmeldinger som fører til at prosjektet aldri blir kompilert. Vi føler vi har vært svært grundige i vår feilsøking, der vi har satt dette opp lokalt på en windows-maskin med XAMPP og flere Ubuntu-baserte LAMP-stacks på lokal installasjon, virtuell maskin og på nettet med DigitalOcean til ingen nytte. Vi har også lagt flere dokumenter der vi har selv dokumentert hvert eneste steg vi har foretatt når vi har fulgt dokumentet vi fikk fra Panteon i håp om at de selv kunne klare å gjenskape problemene, men som nevnt tidligere så har diskusjonen mellom oss og Panteon vært helt fraværende etter vi fikk utlevert prosjektet og medfølgende PDF. Et av disse dokumentene er lagt ved som vedlegg (Vedlegg 6). Selve oppsettsdokumentet kan dessverre ikke inkluderes da kildekoden ikke er åpen for andre enn oss, Panteon og Rune, og dokumentet inkluderer flere lenker til filer som brukes i prosjektet som også ikke er åpne for allmennheten.

Som følge av alt dette tok vi en avgjørelse om å begynne på et nytt prosjekt helt fra begynnelsen av, med formål om å kunne bruke de samme rammeverkene og ressursene som det foreliggende prosjektet slik at det ville vært mer brukbart for Rune i ettertid. Hvis det hadde vært ønskelig kunne en da sett på hvordan noe slikt skulle implementeres, eller om det kanskje også hadde vært mulig å implementert det direkte i sin helhet.

## 5.3 Oppsett av vårt nye prosjekt

Vi har prøvd å bruke de samme språkene og rammeverkene som Panteon har benyttet seg av for å gjøre arbeidet lettere i en eventuell integrasjonsprosess.

I implementasjonen av designerverktøyet har vi valgt å bryte templates ned til klasser i angular. Disse klassene brukes til å lagre både templates og views, samt til å generere viewene før de blir vist frem på en skjerm.

Templates og views lagres hovedsakelig i html/css, så vi har valgt å dele dette opp i tre forskjellige klasser. Her har vi delt det opp i Templates, Content, og Style.

Templates inneholder metadata om en template og en liste med Content objektene som templatene inneholder.

Content inneholder informasjon og metadata om hva som skal vises i templates.

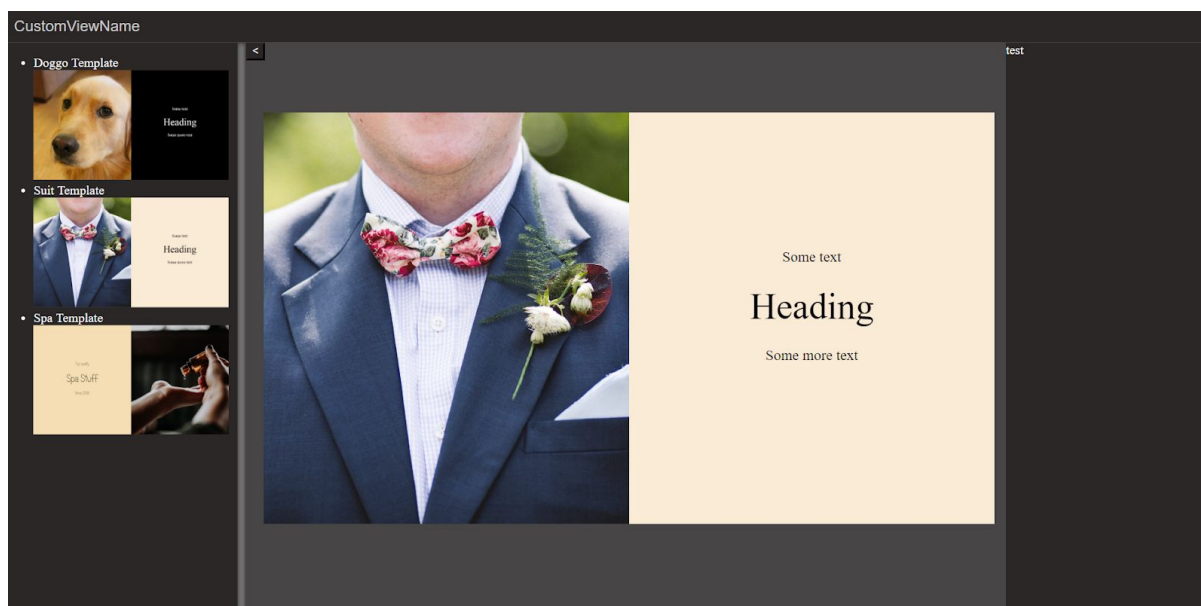
F.eks om det er en <div> eller <p> tag, hva den inneholder, hvor den skal plasseres, og hvilke style objekter som hører til den.

Style klassen inneholder det som ville vært spesifisert i css filen til templatene, og spesifiserer utseende på content objekter.

For å vise en template til brukeren benyttes de samme dataene som brukes til å generere views for å gjenskape templatene i en container i designeren.

Vi har hatt hovedfokus på funksjonalitet under implementasjonen. Da det vi utvikler i hovedsak er en prototype for å vise "proof of concept" har vi valgt å ikke ha like stort fokus på sikkerhet, skalerbarhet og robusthet som på funksjonalitet.

Til nå har vi implementert det generelle brukergrensesnittet, samt støtte for lagring og valg av templates.



Figur 17: brukergrensesnitt i vårt system

Fremover mot eksamen og expo vil vi prøve å få implementert funksjoner for redigering av templates og lagring av views, samt støtte for å kunne hente tidligere lagrede views og vise disse frem.

## 6. PROSJEKTGJENNOMFØRING

Denne delen av rapporten viser en oversikt over hvordan vi jobbet gjennom hele tiden vi hadde til rådighet. Fordi vi valgte scrum som prosjektgjennomføringsmetode så har vi delt opp hele prosjektperioden i sprints. I hver sprint hadde vi en plan over hva vi skulle gjøre fram til en gitt dato. På slutten av hver sprint hadde vi et review-, og retrospektmøte hvor vi gikk gjennom planen og vurderte hvordan den hadde blitt gjennomført. Vi valgte å fremstille prosjektgjennomføringen på denne måten fordi det gir et innblikk i hvordan prosessen vår har vært i kronologisk rekkefølge, med det meste vi har gjort av endringer og tilpasninger underveis.

### Pre-sprint

Januar - 15/02

I denne fasen av prosjektet måtte vi først og fremst bestemme oss for hvilken spesifikk oppgave vi ville jobbe med ref Innledning 1.5. Vi hadde flere møter med Rune Solberg for å diskutere hvilke muligheter vi hadde. Valget falt som kjent på å jobbe med designeren. I tillegg brukte vi en del tid på å få bedre domenekunnskap om Screenz, som vi raskt oppdaget var en mer kompleks løsning enn det vi først antok. Herunder passer uttrykket: The devil is in the details, som betyr at noe først

kan virke både enkelt og raskt, men viser seg å være langt mer komplisert og tidkrevende enn først antatt.

Møte hos Panteon ble også gjennomført i pre-sprint. Her ble vi lovet tilgang til database og kildekode innen kort tid, som noen dager senere ble endret til 20. februar grunnet oppgradering av selve Screenz-programvaren. Ellers fikk vi informasjon om programmeringsspråk og litt informasjon om hvordan samarbeidet mellom Rune Solberg og Panteon fungerte.

I denne perioden var også Jira det vi planla å bruke som prosjektstyringsverktøy. Vi kom derimot aldri skikkelig i gang med bruk av Jira, selv om vi undersøkte det en del online.

Pre-sprinten føles i etterkant som en litt hektisk periode, for vi fikk produsert lite konkret med tanke på programmering, men vi diskuterte og kjørte mye brainstorming sesjoner, som var veldig lærerikt for å bli enige med oss selv hva vi ønsket å gjøre med bachelorprosjektet og hvilket mål vi ønsket å nå når både rapporten skulle leveres og når eksamen skal gjennomføres.

Denne perioden ble det veldig mye fokus på IS-305, og ikke så mye om bacheloroppgaven. Det ble utarbeidet noen brukerhistorier, uten at de ble prioritert eller utdypet.

## Sprint 1

16/02-02/03

### 16/02 - Veiledningsmøte 1

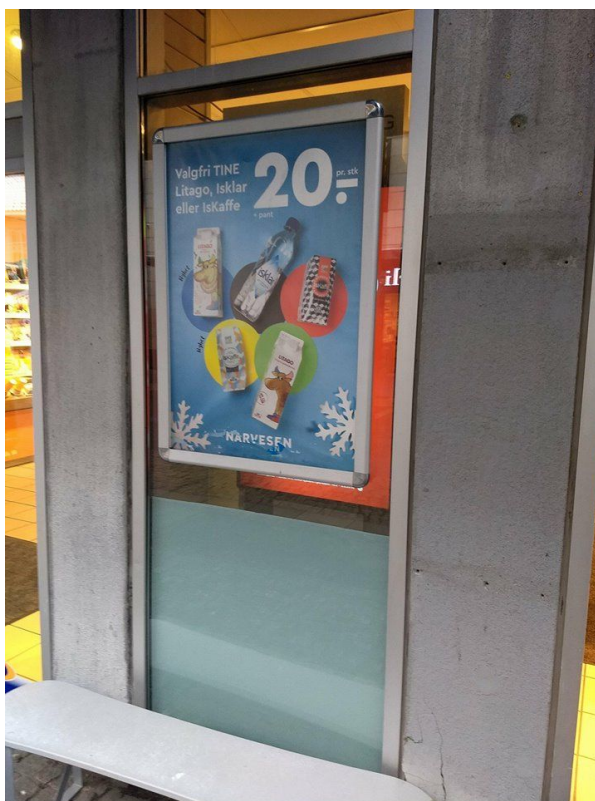
Denne sprinten ble startet samme dag som vi hadde vårt første veiledningsmøte med Janis. I dette møte ble vi enige om å konkretisere oppgaven tydeligere, og ta bort enkelte funksjoner som videoredigering. Deretter skrev vi en konkret prosjektbeskrivelse og sendte den til Rune og Janis. Etter planen burde vi ha fått kildekode 20.februar, og derfor planla vi å ikke jobbe med programmeringen før vi hadde sett på den og vet hva vi måtte gjøre.

### Plan

Vi planla å legge en plan for resten av semesteret for å få et bedre overblikk over frister som må overholdes. Vi skulle også ha et møte med Rune for å gå gjennom våre brukerhistorier.

## Review/refleksjon

Brukerhistoriene vi hadde laget ble vist for første gang til produkteier. Vi gjennomførte i tillegg en rekognoseringstur hovedsakelig opp og ned Markensgate i Kristiansand for å undersøke hvor mange butikker som bruker en eller annen form for TV-skjerm for reklame eller informasjon.



Figur 18: Bildet av butikk i Markensgate

Det var stor variasjon i hvordan skjermene ble anvendt, og hvorvidt de ble brukt på en gunstig måte. Det verste tilfellet var en kiosk som hadde plassert en digital skjerm bak en fysisk plakat akkurat som om det var glemt bort helt. Et bilde av dette har vi tatt med her til venstre.

Et generelt “roadmap” ble også laget for å sikre at vi hadde kontroll over alt som skulle skje i løpet av semesteret og som gjør at vi kan legge en plan som lar seg gjennomføre til innlevering av rapport og eksamen.

Vi ble tipset om AirTable og brukte litt tid på å forstå konseptet. Vi bestemte oss for å bruke AirTable istedenfor for Jira eller Scrumdesk.

Vi brukte mye tid på å diskutere funksjoner og brukergrensesnitt ut fra brukerhistoriene, og laget et førsteutkast på en funksjonsliste.

Både brukerhistorier og funksjonslisten ble prioritert etter MoSCoW-metoden, og vi satt også opp vanskelighetsgrad. Vi ble enige i at brukerhistoriene ble ferdigstilt, selv om vi var klare på at revurdering nok ville være nødvendig.

Vi burde ha gjennomført enda flere møter med Rune Solberg for å få bedre domenekunnskap. Vi merker at jo mer i detaljene vi planlegger og diskuterer, desto flere detaljer rundt selve Screenz har vi enten ikke fått med oss, eller rett og slett ikke husker.

## Sprint 2

05/03-16/03

### Plan

Planen vår for sprint 2, gikk på å utarbeide hva vi hadde behov for med tanke på diagrammer og tabeller med tanke på analyse- og designfasen. Vi ville ikke lage unødvendig mange av disse bare for å lage de, men ønsket at de skulle ha en nytteverdi for oss. Vi ville også klargjøre den tekniske løsningen og vurdere videre oppgaver med den. Vi hadde også i slutten av denne sprinten eksamen i IS-305, og derfor gikk mye av planleggingen av denne sprinten til å prioritere denne.

### Review/refleksjon

Vi fikk diskutert hvilke diagrammer og tabeller vi følte vi burde ha med i rapporten. Også diskutert endel rundt statechart-diagrammet, for å ha oversikten over steg for steg i en funksjon. Vi ba Rune Solberg om bilder/templates som allerede eksisterer i Screenz. Både for å forstå systemet bedre, men også for å forsøke å gjenskape produktet via reverse engineering. Vi begynte på prototype via mockup i InVision. Prototypene ble også revidert etter diskusjon og kommentert internt hos oss i InVision.

## Sprint 3

19/03-11/04 (Påskeferie 24/03-02/04)

### 06/04 - Veiledningsmøte 2

Mot slutten av denne sprinten hadde vi vårt andre veiledningsmøte. Dette var rett etter påske og vi følte at vi hadde mistet litt momentum i arbeidsoppgavene våre. Janis ba oss heller konsentrere oss om hva vi skulle gjøre videre, enn å fokusere på hva vi var misfornøyd med, dvs at vi følte vi lå litt etter skjema. Vi var alle enige i at vi

burde ha enda tettere samarbeid med Rune og jobbe mer på CoWorx. Rune påpekte at vi holdt på å gå i “freelance-fella” som er at man tror man er like fokusert når man jobber hjemmefra enn hvis man møter opp på et kontor. I tillegg ble vi enige at vi skulle forsøke å bruke diverse artefakter i designprosessen vår, som for eks whiteboard eller annet papir vi kan ha på veggen. Vi skulle også sette opp en deadline for brukertest, og få Rune til å booke grupperom for oss oftere og gjerne skaffe eget CoWorx nøkkelkort.

## Plan

Denne sprinten ble litt oppstykket pga. påskeferie. Vi hadde også en delinnlevering i IS-305 rett etter påsken, og derfor ble det en prioritet. Vi hadde også som mål og strukturere rapporten vår og bestemme oss for hva vi vil skrive om i den. Vi hadde også som mål å dokumentere arbeidet vårt bedre fram til sprint 3. I tillegg ønsket vi å begynne utviklingen av designeren.

## Review/refleksjon

Vi begynte på dette tidspunktet å kjenne at vi ligger litt etter skjema, både med selve rapportskrivningen, men også at vi fikk tilgang på kildekode og database en god del senere enn først antatt. Det var satt opp en liten deadline i forhold til når vi skulle anse at vi ikke kommer til å få database og kildekode slik vi er lovet. For da måtte vi heller gå mot et alternativ B, som ville være å programmere uten å implementere det direkte inn i den eksisterende kildekoden vi ventet på. Samt lage prototype uten å ha fått kildekoden. Fristen vi hadde satt var 20 mars, og vi fikk tilsendt dokument med oppsett akkurat på dagen 20 mars.

## Sprint 4

12/04-20/04

## Plan

Fra denne sprinten ønsket vi å ha et eget sprint-planning møte, og et review-møte på slutten av sprinten. Dette var fordi vi ikke var helt fornøyd med hvordan arbeidet har gått til nå, og vi ønsket derfor å reflektere over hva vi kan gjøre annerledes etter hver sprint. Dette var også en av grunnene til at vi valgte å spesifisere backloggen og utvide den kraftig for å ha en bedre oversikt over hva som bør gjøres. Vi hadde nå over 100 oppgaver i backloggen. Vi ønsket også å bruke AirTable mer aktivt for å sikre bedre prosesskvalitet. For første gang hadde vi Scrum-master, og det var Adam som ble valgt til dette.

I denne sprinten ønsket vi å sette opp en dev-server i Digital Ocean. Vi ønsket også å lage prototyper til brukertesting som vi ønsket å få gjennomført i slutten av denne

sprinten, eller i starten av neste. Denne prototypen skulle være i InVision. Vi ønsket også å jobbe mer på CoWorx, for å bedre samarbeidet mellom oss og Rune, og også få mer produktivt arbeid.

## Review/refleksjon

Det å sette opp prosjektet viste seg å være mer utfordrende enn vi hadde trodd. Nesten hele sprinten gikk med på å prøve å sette opp serveren med instruksene vi hadde fått fra Peter. Vi har estimert at vi brukte over 100 timer på å prøve å sette opp serveren uten hell. Vi har sendt mail til Peter for å få veiledning med tanke på dette. Vi legger dette på is fram til vi får tilbakemelding på hva vi gjør feil.

Vi laget en prototype, men vi ønsket å vise den til Rune, før vi gikk videre til brukertesting. Ellers så gikk mye av tiden i denne sprinten til å skrive på IS-305-rapporten.

## Sprint 5

23/04-27/04

### Plan

I forrige sprint gikk mye av tiden på oppsett av prosjekt uten vellykket resultat. Dette førte til at mange av oppgavene som var planlagt ikke ble gjort, og ble derfor overført til denne sprinten. Planen er å jobbe videre med design. Istedenfor å bruke InVision vil vi prøve å bruke Figma, som vi fikk anbefalt fra Rune. Videre håper vi å utføre brukertestene vi ikke fikk gjennomført forrige sprint, pga tiden vi brukte på oppsett av servere. Dette begynner å haste, og bør derfor være en prioritet. I tillegg ønsker vi å begynne på et oppsett av den tekniske løsningen, uavhengig av at vi ikke fikk satt opp serveren.

## Review/refleksjon

Prototypingen i Figma har gått greit. Vi har lært oss hvordan det fungerer etterhvert, og har blitt mer effektive etter hvor lang tid vi har brukt på det. Fordelen med Figma er samarbeidsfunksjonen som gjør at flere kan jobbe med prototypen samtidig.

Nytt teknisk oppsett har gått ok. Vi valgte å bygge vårt prosjekt på samme versjon av de forskjellige rammeverkene som Panteon bruker, men uten å sette opp en backend løsning. Vi har fått kjørende angular-prosjekt i rett versjon. Utfordringen har vært reverse engineering med tanke på bootstrap og en del andre moduler. Litt manglende kunnskap om Angular gjør at dette tar litt tid enn planlagt.



## Sprint 6

30/04-04/05

### Plan

I denne sprinten planla vi å få til brukertesting. Vi hadde forhørt oss med Rune, og mener vi var klare for det nå. Vi bestemte oss for å ikke gjøre noe mer designarbeid, før brukertesting ble fullført.

I tillegg skulle vi sette opp nytt repo på github, og vi ønsket å prøve oss fram med "reverse engineering". Vi begynte også på utviklingen av designerverktøy og implementerte funksjoner basert på backlog-items. Vi ville prioritere designverktøyet, deretter landingssiden og til slutt backend. Dette var rekkefølgen vi ville prioritere implementeringen ut hele perioden vi hadde til rådighet.

Vi ønsket også å klargjøre rapporten til delinnlevering til Janis, tirsdag 8.mai. Vi sendte han en mail for å høre om dette var greit.

### Review/refleksjon

Vi fikk endelig utført brukertesting. Vi testet på gründere på CoWorx og ansatte hos Value Accounting, som ligger rett ved siden av. Vi fikk også satt opp et repo på github. Utviklingen av designerverktøyet har gått bra, og vi har fått implementert det meste av det vi ønsket å få til i denne sprinten. Noen oppgaver gjenstår som vi tok med videre til neste sprint.

Når det kommer til rapporten har den fått en god struktur, og vi vet hva vi skal skrive om, men den mangler fortsatt innhold. Selv om den ikke var ferdig slik vi hadde tenkt, så ble den oversendt Janis som delinnlevering slik at vi kunne få noen konstruktive tilbakemeldinger.

## Sprint 7

07/05-16/05

### 11/05 - Veiledningsmøte 3

Veiledningsmøte 3 ble avholdt på Coworx 5 dager før innlevering av rapporten. Status denne gangen var mer oppløftende enn forrige møte. Vi følte at prosjektet og rapporten var på god vei til å bli ferdig, og at det ikke var noen hindringer på vår vei

de siste dagene. Mye av grunnen til dette er at vi er blitt flinkere til å jobbe i Scrum, dvs vi starter og avslutter sprintene våre på en god og fornuftig måte med gode resultater. Vi har dessverre ikke helt kontroll over tidsbruk i prosjektet i sin helhet, og det får vi litt kritikk for. Rune Solberg kan heller ikke helt si med sikkerhet hvor langt vi er kommet med tanke på en eventuelt MVP/Beta release av det vi har produsert. Men han er generelt fornøyd med innsatsen vår, og ser klar forbedring siden forrige veiledningsmøte. Gjenbruk av kode er heller ikke opp til han, men Panteon. Janis påpeker at planning poker var en metodikk vi kunne ha brukt for å ha bedre kontroll på prosjektet i sin helhet og for å bedre kunne planlegge oppgavene vi har gjennomført.

Janis ber oss også være gjennomtenkte på bruken av "prosjektgjennomføring" slik at innhold ikke overlapper med resultat - konklusjon - refleksjon senere i rapporten. Dette er vi klar over fra før. I tillegg forsøke å koble teori til praksis bedre enn vi har gjort, slik som bruk av FACTOR.

Vi setter også opp et tidspunkt for prøveeksamen sammen med Janis. I tillegg ber vi Janis om skriftlig tilbakemelding på rapporten, slik at vi lettere kan rette opp og forbedre den.

Andre ting Janis nevner som vi bør forsøke å få med i rapporten er klassediagrammer, bilder av whiteboard og annet som kan dokumentere prosessen vi har vært gjennom.

Delen av rapporten som omhandler brukergrensesnitt fikk vi god tilbakemelding på,

## Plan

Dette blir den siste sprinten før rapporten skal leveres inn. I denne sprinten skal rapporten fullføres, og alle backlog-items knyttet til rapporten tas med i denne sprinten. Utover rapporten ønsker vi å fortsette å utvikle designverktøyet, og hvertfall implementere redigeringsfunksjonalitet. I tillegg skal vi gjøre elementer i templates klikkbare.

## Review/refleksjon

På grunn av denne sprinten varer til etter rapporten er levert, har vi valgt å holde sprint review onsdag 16.05.2018. Vi har blitt enige at vi ikke kommer til å inkludere review og refleksjon av denne sprinten i rapporten.

## 6. RESULTAT

Resultatet totalt sett er at vi ikke ble ferdig på en slik måte som vi håpet på i begynnelsen. Selv et godt stykke inn i prosjektet hadde vi fortsatt stor tro på at vi skulle kunne få til å sette opp den opprinnelige databasen på egen server slik at løsningen kunne fungere "Live" med vår funksjonalitet implementert.

For å ta utfordringene først, så er det flere årsaker til at vi ikke har kommet i havn med prosjektet slik vi ønsker. Vi ikke har fått opp serveren og programmeringsspråket Angular har vært mer komplisert å sette seg inn i enn forventet. Men selve prosessen med å lage en solid funksjon har også være mer komplisert og tidkrevende enn vi først trodde. Jo mer vi har jobbet og diskuterte om hvordan vi ønsker at systemet skal fungere og se ut, jo flere detaljer har dukket opp som er vanskelig for oss å kunne anta eller forutse effekten av. Vi har derfor måttet konferere ofte med Rune Solberg for å bekrefte om ideene våre er gode eller ikke. Selve denne prosessen har tatt mer tid og samtidig gitt oss stadige påminnelser om at vi ikke er så gode domenekjennere som vi håpet at vi skulle være. Screenz har rett og slett vært mer komplisert enn først antatt å sette seg inn i.

På den positive siden så har vi produsert flere prototyper som Rune har godkjent og vi har også fått testet prototype på eksterne tilfeldige testere. I tillegg har vi klart å produsere en fungerende egenprodusert Angular versjon. Vi har lært oss Screenz litt etter litt og fått et godt innblikk i markedet rundt Digital Signage. Fokus på brukervennlighet har hele tiden vært med i alle prosessene vi har utført. Så på det grunnlaget har vi ikke mistet fokuset på oppgaven gjennom semesteret. I tillegg har vi utforsket mange forskjellige programvarer og løsninger som har gitt oss inspirasjon, og vi har også utforsket flere verktøy som vi har stor tro på at vi kan bruke i forskjellige situasjoner en gang i fremtiden.

Rapporten er vi også godt fornøyd med. Selv om Scrum som prosjektstyingsform har vært litt varierende, så har vi produsert og dokumentert fremdriften. Brukerhistorier, user cases, funksjonslister og prioriteringsmetoder ble etterhvert til tabeller, diagrammer og prototyper. Sammen med skjermbilder viser rapporten at vi har jobbet med flere ideer og ført disse ideene gjennom flere faser og dermed har en rød tråd som gjenspeiler prosjektbeskrivelsen.

Rune Solberg har ikke vært involvert i rapporten, bortsett fra hans bidrag i form av uttalelse om hvordan han føler at vi har utført prosjektet hos han.

I retrospekt ser vi at vi i prosjektforslaget har satt opp en del punkter som vi ikke har gjennomført slik vi først hadde planlagt. Dette grunnet at vi ikke hadde nok

domenekunnskap på daværende tidspunkt, og derfor ble det gjort en del antagelser som vi senere burde ha forandret når prosjektet ble mer formelt utarbeidet iht utviklingsverktøy, prosjektstyringsverktøy, samt at Rune faktisk ikke var tilstede hele tiden grunnet jobboppdrag utenfor Coworx/Kristiansand. Prosjektbeskrivelsen tar også for seg alle mulighetene vi hadde, dvs designer, beacons og bluetooth-knapp, hvor vi som kjent valgte bort de 2 sistnevnte.

I prosessen vi har vært gjennom har vi sett at ideer kommer og går ganske fort. Det gjør at en god idé ikke blir dårlig, men blir erstattet av enda bedre ideer raskere enn vi antok. Produkteier har stor kunnskap om Digital Signage og aktører som er inne på markedet pr nå. Det kombinert med sin erfaring innen grafisk design gjør at han fort har en mening om en idé er god, mindre god eller rett og slett dårlig. Vi som ekstern gruppe var i utgangspunktet blanke innen dette markedet, og det gjør at idéer som vi kom opp med kunne virke gode, men som ikke ble avdekket som mindre gode grunnet vår manglende erfaring på området. Det gjør at endel idéer og prosesser ofte ble dobbeltarbeid eller ekstrajobb i selve prosjektet sett i ettertid. Blant annet ville tidlig i prosjektet gjennomføre en Google Design Sprint for å få igang en god prosess på ideer tidlig, men Rune kunne raskt informere om at en slik prosess er et dårlig utgangspunkt uten at han som produkteier var sterkt involvert.

## 7. DISKUSJON

I denne delen av rapporten diskuterer vi utfordringer vi har hatt, hvordan vi evt har løst dem, og hvordan vi kunne gjort ting annerledes.

### 7.1 Utfordringer med sen informasjon fra Peter Hamre ved Panteon

Den utfordringen som klart har vært vårt største hinder har vært samarbeidet mellom oss og Peter fra Panteon. Panteon har utviklet det tekniske systemet til Screenz. I starten av prosjektet, 06. februar, så hadde vi et møte med Peter i lag med Rune. Da diskuterte vi rammene rundt bachelorprosjektet og det hele virket veldig lovende. Vi skulle bli overlevert en del kildekode i form av et privat GitHub repository, og det ble insinuert at Peter var villig til å samarbeide og hjelpe underveis slik at det vi skulle utvikle kunne brukes på samme plattform. Til dels så har dette blitt gjort, og det er urettferdig å si at Peter ikke var samarbeidsvillig, men det hele har vært veldig skuffende. Etter møtet fikk vi ikke tilsendt noe kildekode, og i løpet av de neste 6 ukene purret vi Peter flere ganger på e-post om når vi skulle motta kildekoden. Etterhvert fikk vi noen korte meldinger ca. ukentlig med varierende beskjeder, og på ett tidspunkt fikk vi beskjed om at vi ikke kunne få tilsendt kildekode likevel og måtte sette opp en server. Denne serveren skulle Peter koble seg til med via FTP og en

databasebruger til en MySQL-database, samt også ha tilgang til et adminpanel. Her brukte vi tid på å sette opp en server både på DigitalOcean og på UH-iaaS. Når han da kom tilbake til oss, fikk vi beskjed om at han trengte mer tilgang enn han hadde definert i sine første krav. Disse kravene var noe mer kompliserte og det ble brukt en del tid på dette. Deretter sluttet Peter å svare på e-post i noen uker, og i denne perioden kunne vi ikke se at det ble gjort noe forsøk på å bruke serverene våre. En slik dialog hadde vi i en og en halv måned før vi fikk tilsendt noe som helst av kode. Vi hadde sammen blitt enige om at hvis vi ikke hadde fått noe som helst innen 20. mars, altså en måned etter den lovede fristen, så skulle vi starte på vårt eget prosjekt. Ironisk nok så fikk vi tilsendt koden på akkurat denne datoen. Da fikk vi en .PDF med instruksjer for oppsett og selve kildekoden, som også gjorde at oppsettet av de tidligere serverene var unødvendig.

Når vi prøvde å sette opp prosjektet med instruksene vi fikk så støtte vi på problemene som har blitt nevnt tidligere i rapporten under 5.2 oppsett av prosjekt. Etter dette ble kommunikasjonen helt fraværende. Istedenfor å få et svar direkte fra Peter som var til hjelp i troubleshooting, fikk vi beskjed fra Rune at Peter hadde sagt i en mail til Rune at han ikke kom til å ha tid til oss i det hele tatt i perioden fremover. Det gjorde at vi effektivt bare hadde kastet bort masse tid på å sette opp et prosjekt som vi ikke kunne garantere at vi kunne ta i bruk. Siden dette var ingen garanti gikk vi over til å bygge vårt eget prosjekt fra bunnen av i samme systemarkitektur.

Vi føler at dette var et klokt valg fra vår side, selv om det selvfølgelig er mulig at problemet kunne vært løst etter bare en ekstra time eller to med troubleshooting. Vi ser dette som lite sannsynlig, og ville ikke blitt overrasket om det dukket opp mye flere slike problemer etterhvert som vi begynte utviklingen. Da ville vi bare havnet i samme situasjon igjen, og på vårt tidsskjema var det absolutt ikke tid til en ekstra 100 timer brukt på troubleshooting.

## 7.2 Scrum

Vi bestemte oss for at vi skulle bruke scrum tidlig i prosessen, men i hvilken grad vi har brukt alt som inngår i scrum har variert. I starten var vi ikke like flinke til å ha review-meetings etter at sprinten var ferdig. Vi dokumenterte heller ikke sprint-planningen i et eget dokument, men la til oppgaver som skulle gjøres i et gantt-chart. Backloggen vår inneholdt svært få elementer på dette tidspunktet. Etter sprint 4 dokumenterte vi sprint planning og review i egne dokumenter. Vi stykket også opp backloggen og fikk over 100 backlog-items. Når oppgavene ble mer konkrete var det lettere for oss å forholde oss til en oppgave, og å sette den som ferdig og derfor få en bedre følelse av progresjon. Når vi skulle dokumentere sprintene i denne rapporten var det lett for oss å dokumentere sprint 4 -7. Vi hadde

litt større problemer med å dokumentere de 3 første sprintene, men vi hadde noen dagboknotater og møterefater som hjalp oss med dette, så vi fikk dokumentert disse sprintene også. I starten byttet vi litt på hvilket verktøy vi skulle bruke med tanke på Scrum og prosjektstyring. Først var vi innom Jira og deretter Gantt-chart i google sheets, men vi valgte etterhvert AirTable som var lettere å bruke for oss. Her lagret vi alle backlog-items og satt opp sprintene med items.

### **Gruppearbeid**

Tidlig i semesteret møttes vi ikke hver dag i uka. Senere ble vi enige om at vi måtte møtes mandag-fredag, fordi vi jobbet mer effektivt når vi jobbet sammen. Dette var nødvendig fordi tiden ble stadig en mer kritisk faktor, særlig med tanke på at kildekoden kom en måned for sent, og vi var ikke like produktive som vi burde i perioden vi ventet på denne. I de senere sprintene møttes vi oftere på CoWorx. Dette var fordi det var da lettere å ta kontakt med Rune Solberg hvis det var noe vi trengte hans godkjenning på eller veiledning om.

### **Konflikt med IS-305**

I starten av semesteret var det mye tid som gikk på IS-305. Dette var fordi de første 2 månedene hadde vi 1-2 forelesninger, pluss 1-2 individuelle oppgaver i uken. I tillegg skulle vi skrive en rapport på 15 sider som et gruppeprosjekt, i tillegg til at eksamen kom 16.mars. Etter dette ble det noe mer fokus på bachelorprosjektet, men rapporten i IS-305 skulle leveres i begynnelsen av Mai, så derfor måtte vi prioritere denne også. Det endte med at vi bestemte oss for at 2 av oss skulle ha hovedfokus på IS-305 rapporten, mens resten fortsetter kun med fokus på bacheloroppgaven. Dette har gått greit, men vi mener at IS-305 har "stjålet" alt for mye tid i forhold til størrelsen på faget.

## **7.3 Hva kunne vi gjort annerledes?**

Da det er første gang vi gjennomfører et bachelorprosjekt og også følger et fungerende produkt til produkteier, så har vi alle sammen gjort ting vi ikke har gjort før. Det resulterer naturlig nok til at en god del ting kunne vært utført annerledes, når vi ser prosjektet i retrospekt. Når vi i tillegg også legger til at flere av oppgavene har vi lite eller ingen erfaring med, for eksempel design, implementering i 3 parts kildekode og ukjent programmeringsspråk, så har utfordringene stått i kø. For å uansett oppsummere hva vi kunne gjort annerledes har vi kategorisert oppgavene.

### **Prosjektmessig**

Selv om vi har byttet ut Jira og Gantt med AirTable, så burde vi har skrevet dagbøker fra dag 1, samt gjennomført Scrum bedre. Sprint-planning, sprint-review og backlog burde vært gjennomført fra dag 1. Vi burde også hatt noe mer konkret kontroll over

tidsestimatene som f.eks timeføring. Hadde vi f.eks presentert burndown charts på sprint review/retrospect hadde det nok vært mer tydelig i hvilken grad vi har jobbet i sprintene, noe som gjerne kunne være vanskelig å forestille seg der og da.

Noe som vi føler har vært en klar mangel hos oss har vært mangelen på sprint burndown charts. Vi brukte i starten av prosjektet mye tid på å finne ut av hvordan vi kunne automatisere dette med Jira, men dette fikk vi aldri helt til. Dette var hovedmotivatoren til at vi gikk bort fra Jira og over til Gantt-chart, slik at det ville vært enklere å gjøre manuelt, og deretter AirTable. AirTable skulle også være kraftig nok til å automatisere dette, men mye av denne funksjonaliteten ligger bak en betalingsmur. Et alternativ vi vurderte i starten av prosjektet var ScrumDesk, og her er det svært enkelt å loggføre timer og skrive ut burndown charts, med begge disse to elementene som mangler i vårt prosjekt. Det kunne tenkes at et valg av en tjeneste vi allerede var kjent med hadde vært bedre for kvalitetssikringen i det lange løp.

Fordi vi mangler sprint burndown charts, er det vanskelig for oss internt å vite om vi har jobbet jevnt nok i løpet av en sprint. Vi har jo bare tidsestimater, men uten noen konkretisering av dette er det vanskelig å kontrollere hva som blir gjort til forskjellig tidspunkt.

### **Oppmøte**

Vi kunne ha møttes mer på CoWorx slik at Rune var mer i nærheten, spesielt tidligere i prosessen. Ved å møtes opp på UiA så mistet vi litt forbindelsen til Rune og prosjektets kjerne. Men grunnet at vi har diskutert mye i plenum så har vi opplevd det bedre å sitte på grupperom enn i åpent landskap på CoWorx. Det som senere viste seg som en mulighet var at vi ikke trengte å benytte det åpne landskapet, men heller kunne booke en av flere møterom på CoWorx via Rune siden vi selv ikke hadde adgang på CoWorx sine interne bookingsystemer. Dette burde vi strengt tatt forstått tidligere.

### **Domenekunnskap**

Altfor mange ganger innså vi at vi hadde for dårlig domenekunnskap om både Screenz-softwaren, digital signage- markedet og prosessen fra idé til ferdig produkt. Her burde vi ha brukt mer tid på CoWorx, testet mer i selve Screenz-programmet, og brukt Rune mye mer som ressurs enn vi har gjort.

### **Arbeidsfordeling**

I retrospekt ser vi at en av de aller mest avgjørende valgene vi tok var å fordele oppgaver slik at ikke alle jobbet med samme oppgave samtidig. Grunnen var at det ble fort at 1 eller 2 jobbet, mens resten så på. Ved delegering så jobbet vi mye mer effektivt. Vi burde ha gjort dette flere sprinter tidligere enn vi gjorde. Å starte med å

ikke jobbe sammen ville nok ikke vært en god ide, for vi måtte helt klart avklare en god del ting i begynnelsen slik at prosjektet fikk en form og mål vi alle var enige i

### **Designmessig**

Vi burde ha vært igjennom en tydeligere designprosess med Rune tidligere for å kunne finne ut hva han ønsket seg av design, samt kunne skrotet mindre gode idéer før de ble utarbeidet. Derimot innebærer det at domenekunnskapen og designprosessmetodikk er grunnleggende god, noe den i stor grad ikke var hos oss den første perioden.

### **Teknisk utvikling**

Tidligere teknisk utvikling, samt oppsett av server, burde vært på agendaen. Vi hadde dokumentet en stund, men somlet i noen uker like etter pga påskeferie. Det var en felles enighet om at vi skulle sette opp prosjektet i påskeferien, men vi erfarte at ingen ville jobbe noe med det i ferien og heller benyttet ferien til fritid.

### **Testing**

Grunnet at teknisk utvikling lå etter skjema, så ble også prototyper lidende for det. Det gjorde at testing ble utsatt grunnet at prototypene ikke ble ferdige. Vi skulle gjerne ha sett at dette punktet ble gjennomført mye tidligere enn det faktisk ble gjort.

### **Rapporten**

Vi burde ha startet å skrive rapporten før. Vi hadde lenge kun en rapportmal liggende, og denne kunne ha blitt fylt ut med mange punkter på et mye tidligere stadium. Det har gjort at mye tid som kunne blitt brukt på utvikling måtte brukes på dokumentering.

### **Risikoanalyse**

Det vi ikke har tatt med som kunne vært praktisk å ha, er en risikoanalyse. I starten av perioden burde vi satt oss ned og sett på hvilke potensielle uønskede hendelser som kunne oppstå. Vi måtte da satt ned hvor sannsynlig det var at hver hendelse skjedde, og sette opp tiltak deretter. Disse tiltakene kan være forebyggende, men også tiltak som iverksettes dersom en uønsket hendelse allerede har skjedd. ("Risikostyring", 2018) Hvis vi hadde hatt dette klart på forhånd kunne vi *kanskje* agert annerledes når vi ikke fikk kildekoden på den satte tiden.

## **8. KONKLUSJON**

Alt i alt har dette vært en flott opplevelse og en lærerikt erfaring for oss alle sammen. Etter tilslaget av prosjekt for Rune Solberg, så har vi tatt mange avgjørelser, fått masse ny erfaring og kompetanse, og fått en godt innblikk i hvordan en gründer har



tatt en ide og realisert den til en konkret tjeneste. Og det var stort sett akkurat det vi håpet på fra begynnelsen da valget falt på Screenz-prosjektet.

Selv om veien har vært lærerik og inspirerende, så har det så absolutt vært store utfordringer som vi har måttet løse på vår vei. Noen har vi klart å løse, andre måtte løses med en alternativ plan-B. Å sette seg inn i produktet Screenz, den tekniske løsningen vi aldri fikk satt opp (inkludert den etterhvert ikke-eksisterende kommunikasjonen med Panteon), samt hvor vanskelig det faktisk er å komme opp med en brukervennlig løsning, har vært de største utfordringene vi har hatt underveis.

Innblikket vi har fått i Digitale Skilt har også vært veldig spennende. Dette visste vi lite eller ingenting om fra før. Bare det å først høre Runes visjoner rundt emnet, for så å ta seg en tur oppover Markensgate og se de forskjellige løsningene de forskjellige butikkene har, var kjempeinteressant.

Motivasjonen til gruppen generelt har av nevnte årsaker også gått litt opp og ned, men som både vi og oppdragsgiver påpeker, så har vi avsluttet svært sterkt og er godt fornøyd med produktet som foreligger til slutt.

## 9. REFERANSER

Wikipedia (2016) *Digital Skilting*,

Hentet 03/05-18 fra [https://no.wikipedia.org/wiki/Digital\\_skilting](https://no.wikipedia.org/wiki/Digital_skilting)

Store Norske Leksikon (2018) *Kvalitet*,

Hentet 10/5-18 fra <https://snl.no/kvalitet>

Store Norske Leksikon (2014) *Kvalitetssikring*,

Hentet 10/5-18 fra <https://snl.no/kvalitetssikring>

Wikipedia (2018) *Project Management Triangle*,

Hentet 09/05-18 fra [https://en.wikipedia.org/wiki/Project\\_Management\\_Triangle](https://en.wikipedia.org/wiki/Project_Management_Triangle)

Wikipedia (2018) *Anchoring*,

Hentet 14/05-18 fra <https://en.wikipedia.org/wiki/Anchoring>

Haughey, D. (2014) *Moscow Method*,

Hentet 15/05-18 <https://www.projectsmart.co.uk/moscow-method.php>

Mathiassen et al (2000) *Object-oriented analysis and design*.

Aalborg: Marko Publishing

Wikipedia (2018) *Light on dark color scheme*,

Hentet 06/05-18 fra: [https://en.wikipedia.org/wiki/Light-on-dark\\_color\\_scheme](https://en.wikipedia.org/wiki/Light-on-dark_color_scheme)

Tidwell, J. (2010) *Designing Interfaces*,

Canada; O'Reilly Media

Simon D.P.(2017) *The Art of Guerrilla Usability Testing*,

Hentet 14/05-18 fra

<http://www.uxbooth.com/articles/the-art-of-guerrilla-usability-testing/>

Wikipedia (2017) *Risikostyring*,

Hentet 15/05-18 fra <https://no.wikipedia.org/wiki/Risikostyring>

# 10. VEDLEGG

## Vedlegg 1 - Evaluering fra oppdragsgiver

Evaluering av bachelor-oppgave:

### **Screenz Designer**

Gjøre det mulig for innehavere av små og mellomstore butikker å ta kontroll over sin egen markedsføring på reklameskjermer uten bruk av reklamebyrå eller designer for det daglige kommunikasjonsbehovet.

Gruppen ble utfordret til å komme med forslag på løsning basert på initiell oppgave presentasjon. Da kunne gruppen selv styre kompleksitet til oppgaven basert på den samlede kompetansen.

Oppdragsgiver : Screenz v/ Rune Solberg

#### **Oppdragsgivers inntrykk av utført arbeid:**

Det var en veldig godt sammensatt gruppe rent kompetansemessig. Det virket som de var godt rustet til å takle de forskjellige problemstillingene de ville bli satt ut for. Ansvarsrollene i gruppen er tydelige, og de har virket veldig flinke på samarbeidet internt. Det har blitt gjennomført obligatoriske styringsgruppe-møter, med veileder tilstede. Oppdragsgiver har savnet litt mer innblikk i hva som utvikles underveis. Oppdragsgiver har per skrevet evaluering ikke sett ett ferdig produkt så det er ikke mulig å evaluere den delen.

#### **Konklusjon:**

Opptrukt svært profesjonelle i styringsgruppe-møtene. Agenda ble stort sett sendt pr. mail før møtene. God stemning mellom oppdragsgiver og gruppen. I starten av prosjektet la de veldig stor vekt på den tekniske plattformen for å få tilgang til den og få satt denne opp internt, da dette ble utsatt fra Screenz side grunnet ny lansering av plattform følte det som gruppen dabbet litt av, før de fant ny giv, ved å konkludere med at de ville fortsette med standalone produkt som heller siden kunne implementeres i løsningen. Det var et godt valg. Man burde her samtidig med den initielle tekniske setupen også startet jobben med brukeropplevelse, user stories og skissing av løsningen slik at man var klar når den tekniske plattformen var på plass. De siste ukene av prosjektet har gruppen jobbet veldig bra og fått gjort veldig mye.

Jeg gleder meg til å se produktet når det er klart til presentasjon og jeg tror de greier å dra dette i land.

#### **Rune Solberg**

Gründer av Screenz

rune@screenz.io

tel. 92 83 46 09

## Vedlegg 2 - Selvevaluering

Gruppen i sin helhet har ikke jobbet sammen før, men alle har jobbet sammen med noen av de andre gjennom hele bachelorprogrammet. Det gjør at vi alle var positive til samarbeidet, vi hadde stor tro på at vi skulle klare å lage et godt produkt, og det mener vi også har vi har klart! Vi har ikke hatt noen gruppekontrakt, og det har ikke på noe tidspunkt vært nødvendig i ettertid heller. Vi var fra starten veldig klar over at noen hadde større tekniske kompetanse i form av programmeringserfaring, mens andre var mer analytisk og mer innstilt på å skrive rapporten. Alle var uansett opptatt av produktet i sin helhet, dvs lage et produkt vi alle er fornøyde og stolte av, som vi håper kan bli brukt på en eller annen måte av produkteier i fremtiden. I perioden frem til ca 2 veiledningsmøte jobbet vi ofte på de samme tingene for å kunne diskutere hva og hvordan prosjektet skulle utvikles. Men vi oppdaget at det ble bortimot sosial loffing, da noen jobbet for eks med prototype, men de andre så på og kommenterte. Vi fikk utrettet mye mer når vi delte opp oppgavene slik at design ble utført av 3 stk, men 3 satt og skrev rapport eller utførte andre analyseoppgaver. Dette stemmer også overens med produkteiers filosofi, ikke gjør alle ting selv, deleger oppgave, for da blir hver oppgave utført av folk med riktig kompetanse. For produkteier blir det outsourcing, mens vi kan kalle det delegering. Vi kvalitetsikret uansett fremgangen med at sjekke hverandres arbeid på tvers av "fagfelt", samt at alle uansett var opptatt av å inneha en total oversikt over prosjektet. Og apropos kvalitet, vi har hele tiden hatt fokus på at produktet skal være brukervennlig, samt at den som til slutt har siste ord angående kvalitet, er produkteier Rune Solberg.

Adam Ajmi -

I prosjektet har jeg bidratt i tilnærmet hver eneste del av prosjektet. Jeg har vært aktiv i analysearbeidet og har sett en del på prosjektstyring og prosjektstyringsverktøy. Jeg har også brukt noe tid på design av brukergrensesnitt både i InVision og Figma og har vært med på testing av brukere sammen med Andrei. Av implementasjon har jeg jobbet aktivt med oppsett av servere slik at Panteon kunne overføre sin kildekode til oss, og når vi omsider fikk kildekoden, jobbet med å få dette kjørende på en server på DigitalOcean. Til tross for mye egeninnsats her og forhøring med andre var ikke dette mulig etter flere ukers intensivt arbeid uten noe hjelp fra Panteon. Etterhvert som vi begynte på et eget teknisk prosjekt har jeg vært noe mer fraværende på implementasjonsarbeidet da Håkon og Andrei har jobbet flittig og jeg har brukt tid på dokumentasjon i ettertid. Jeg har til slutt brukt mye tid på å skrive på og lese over rapporten, samt bidra der jeg føler det har vært hensiktsmessig. I rapporten har jeg skrevet mye om design, oppsett av foreliggende teknisk prosjekt og prosjektstyring.

Andrei Tanase -

Totalt sett er jeg fornøyd med hvordan prosjektet har gått. Jeg valgte å fokusere på designarbeid og den tekniske siden av prosjektet. I begynnelsen av prosjektet jobbet jeg med skisser av brukergrensesnittet. Derpå har jeg bidratt til prototypen som vi lagde i både InVision og Figma, jeg har også vært med å utført brukertesting sammen med Adam. Jeg har også brukt litt tid på å utarbeide alle diagrammer vi har hatt behov for og knyttet de opp mot brukerhistoriene i vårt prosjekt. Mot slutten av prosjektet har jeg jobbet for det meste på implementasjonsarbeidet sammen med Håkon. På selve rapporten har jeg bidratt mest på

design og diagrammer, men også litt på implementasjonsdelen. Jeg har også vært med å gitt tilbakemeldinger og kommentarer på andre deler av rapporten.

Håkon Methi -

Jeg er svært fornøyd med arbeidet gruppen har gjennomført. Selv har jeg hatt hovedfokus på design og utvikling av prosjektet vårt, og litt mindre fokus på dokumentasjonen/rapporten. Jeg var også aktivt med på feilsøking og oppsett av server løsningen vi fikk fra Panteon. I rapporten har jeg hatt hovedfokus på implementasjonsdelen, og bidratt litt her og der. Det meste av mitt arbeid har gått til utvikling av design, prototype, og implementasjonen.

Rune Vålandsmyr Olsen -

Jeg er fornøyd med hvordan arbeidet med prosjektet har blitt bedre underveis i prosessen. Jeg har hatt et hovedfokus på skriving i rapporten gjennom hele prosessen, og da særlig fokus på metode, kvalitetssikring, prosjektgjennomføring og diskusjon. Jeg har også skrevet dagbøker for at vi bedre skulle huske hva vi gjorde til enhver tid. Da vi så at IS-305 tok mer tid enn vi hadde trodd på forhånd, så valgte vi for en periode og dele opp arbeidet litt slik at jeg og Anders hadde mer fokus på å fullføre rapporten i IS-305, mens de andre holdt på med design og den tekniske løsningen i IS-304. Jeg synes vi kom litt tregt i gang i starten, men jeg synes generelt at arbeidet vårt den siste 1,5 måneden har vært veldig bra og effektiv jobbing, og vi har fått gjort mye. Hadde vi vært like effektive hele perioden, i tillegg til at vi hadde fått den tekniske løsningen på plass til riktig tid, så hadde vi fått et langt bedre resultat.

Anders Eskeland

Jeg er som de andre generelt fornøyd med arbeidet som er blitt utført i løpet av dette semesteret. Dette selv om det opprinnelige målet ikke ble nådd slik jeg hadde håpet. Min innsats var i begynnelsen veldig bred og variert, men ble etterhvert fokusert slik at jeg ikke hadde for mange baller i luften. Da ble det enklere å utføre og bli ferdig med oppgaver som jeg tok til meg. For, jeg og Rune brukte mye tid på IS-305 rapporten mens flere andre i gruppa fokuserte på det tekniske og designmessige i form av oppsett av prosjektserver og prototype. Jeg har hele tiden arbeidet for å ha oversikt over hele prosjektet, selv om jeg ikke aktivt har jobbet med alle elementene. Som nevnt hadde vi andre fag som tok mye tid i begynnelsen av prosjektet, og det gjorde at jeg ofte har følt at vi burde ha kommet lengre, samt at de tekniske problemene vi hadde virkelig har gjort at vi har stått i stampe eller stått på stedet hvil.

En annen viktig ting er at jeg føler at jeg har fått brukt mye av kunnskapen og kompetansen jeg har opparbeidet meg gjennom hele bachelorprogrammet. Det har gitt meg en god følelse og vært verdifullt for meg. Ikke bare i dette bachelorprosjektet, men også i jobbsøking som også har gått parallelt med dette bachelorprosjektet.

## Vedlegg 3 - Brukerhistorier

### Designer

#### 1 Som designer

**Ønsker jeg å kunne laste opp templates**

**Slik at mine kunder kan ta i bruk mine designs i Screenz.**

Beskrivelse: Målet med designeren er at kunder skal kunne benytte slike templates til å forhåpentligvis enkelt kunne opprette egne skjermbilder av høy kvalitet uten ekstern hjelp fra en kvalifisert designer. Disse templatene vil i første omgang bli laget av en kvalifisert designer. Målet er å konstruere dette på en slik måte at en designer vet hvordan han kan generere en template og at den kan lastes opp i tjenesten vår.

Prioritering: Veldig lav. Won't have.

Grunnen til dette er at vi kan forøvrig bare implementere disse vha. Hardkoding i første omgang frem til vi ser at bruk av templates fungerer i praksis.

Akseptansekriterier: Designer skal enten være en egen brukergruppe i systemet, eller så skal enhver bruker laste opp en template. Systemet må kunne tolke slike filer slik at det samles sammen til en template som kan refereres til i systemet og konverteres til et bilde.

#### 2 Som designer

**Ønsker jeg å bestemme hvem som kan bruke mine templates**

**Slik at bare kundene som betaler meg får lov til å bruke mine designs i Screenz**

Beskrivelse: Målet med å kunne bestemme hvem som skal ha tilgang til diverse templates er at en designer skal kunne lage templates for flere kunder/bedrifter og holde styr på at hver kunde skal kun se templates som den har betalt for.

Prioritering: Veldig lav, Won't have.

Akseptansekriterier:

#### 3 Som designer

**Ønsker jeg å kunne endre på templates (endre, legge til, trekke fra)**

**Slik at templatene kan oppdateres**

Beskrivelse: Kundene sine behov vil nok endre seg over tid, og dette vil gå i tråd med det designeren trolig blir betalt for å gjøre.

Prioritering: Veldig lav. Won't have.

Akseptansekriterier: Funksjonen blir akseptert når en designer kan endre på opplastede templates.

#### **4 Som bruker**

**Ønsker jeg å kunne laste opp et bilde til designeren.**

**Slik at jeg kan bruke det i templatene.**

Beskrivelse: Brukere av screenz vil gjerne lage designet selv, og må da kunne laste opp bilder for å lage designet som skal vises på skjermen.

Prioritering: Must have - Dette er en kjernefunksjon i systemet og må være til stede for at det skal være noe verdi.

Akseptansekriterier: Funksjonen er akseptert når brukeren kan laste opp et bilde og benytte seg av dette i designeren.

#### **5 Som bruker**

**Ønsker jeg å kunne redigere tekst i en template**

**Slik at jeg kan tilpasse bilder til relevante tilbud/salg.**

Beskrivelse: Brukeren ønsker å endre på tekst fordi bedriften har ulike tilbud med forskjellige priser og derfor er det gunstig at det går an å redigere teksten.

Prioritering: Must have - Dette er en kjernefunksjon i systemet og må være til stede for at det skal være noe verdi.

Akseptansekriterier: Funksjonen er akseptert når brukeren kan endre på teksten i en template.

#### **6 Som bruker**

**Ønsker jeg å kunne søke på templates etter tags**

**Slik at jeg lettere finner typen template jeg er ute etter.**

Beskrivelse: Brukeren vil noen ganger ha behov for å bruke en template med et tema som passer til anledningen. Dette kan f.eks være i forbindelse med jul, påske, sommersalg osv. Når brukeren har mulighet til å søke etter tags, så får brukeren mer relevante templates ut i fra hva han er ute etter.

Prioritering: Should have. Noe tvil om den kanskje skulle vært Could Have. Argumentet her var hvor viktig det faktisk er for brukerne å kunne filtrere templates. Det vil bli problematisk over tid når det er veldig mange templates, men som en del av en MVP vil dette være prioritert nokså lavt.

Akseptansekriterier: Akseptert når brukeren kan søke gjennom templates basert på forhåndsdefinerte tags. Designere må også kunne legge til egne tags på sine templates.

## **7 Som bruker**

### **Ønsker jeg å velge templates og se en preview**

#### **Slik at jeg ser hvordan det kommer til å se ut før jeg begynner å redigere**

Beskrivelse: Brukeren ønsker å kunne se en forhåndsvisning av templates for å kunne velge korrekt template raskest mulig.

Prioritering: Could have.

Akseptansekriterier: Hver template skal ha en kontekstmeny der man skal kunne velge en preview. Ved valg av preview skal det enten vises en "modal" popup med et skjermbilde eller tilsvarende løsning.

## **8 Som en bruker**

### **Ønsker jeg å kunne velge templates når jeg går inn i designeren**

#### **Slik at jeg kan bestemme blant de templatene jeg ønsker å designe videre på**

Beskrivelse: Brukeren skal kunne bla igjennom de forskjellige templatene som er tilgjengelig og velge en av de han ønsker å ta i bruk.

Prioritering: Must Have - Vil være en viktig og essensiell del av MVP.

Akseptansekriterier: Når brukeren kan velge en template fra et utvalg av templates.

## **9 Som en bruker**

### **Ønsker jeg å kunne tilpasse fargevalget i en template**

#### **Slik at templatene kan bli tilpasset firmaets fargevalg/design.**

//Dette er vel et eksempel på en spesifisert "små-endring"?

Beskrivelse: Brukeren ønsker å kunne endre på fargene i en template slik at det passer med firmaets fargevalg og logo.

Prioritering: Should have.

Akseptansekriterier: Akseptert når kunden kan endre på farger i en template. Designeren må også kunne spesifisere diverse fargevalg for templatene hvis denne funksjonen skal kunne



tas i bruk. Det betyr at dette må kunne implementeres fra designersiden, og deretter utnyttes på brukersiden.

### **10 Som bruker**

**Ønsker jeg å kunne rotere på opplastede bilder i en template**

**Slik at jeg slipper å gjøre det på min egen datamaskin hvis det gjør at bildet passer bedre.**

Beskrivelse: Brukeren ønsker å kunne rotere bildene i templatene slik at han slipper å gjøre dette på forhånd før han legger til bildet i designeren.

Prioritering: Could have.

Akseptansekriterier: Akseptert når brukeren kan rotere bildet i templatene.

### **11 Som bruker**

**Ønsker jeg å kunne fritt flytte på elementene i en template**

**Slik at jeg kan sette preg på mine skjermer med unik posisjonering**

Beskrivelse: Brukeren ønsker å fritt kunne flytte på elementene i en template fordi han vil ha kreativ frihet for å sette et personlig preg på templatene. Dette kan være nyttig når brukeren finner en template han synes passer bra, men ikke perfekt. Det at han da kan flytte på elementer gjør at templatene blir mer fleksible og kan brukes til flere formål.

Prioritering: Could have.

Akseptansekriterier: Akseptert når brukeren kan flytte elementene fritt i templatene.

### **12 Som bruker**

**Ønsker jeg å kunne selv sette inn elementer i en template**

**Slik at jeg kan bygge templates opp slik jeg vil**

Beskrivelse:

Prioritering: Won't have.

Akseptansekriterier

### **13 Som bruker**

**Ønsker jeg at når jeg beveger på elementer så vises det stiplede justeringslinjer**

**Slik at jeg enklere kan sentrert plassere elementer på templatene.**

Beskrivelse: Når brukeren nærmer et element til X eller Y-aksen, skal den automatisk klynge ("snappe") til denaksen.

//Tilleggsfunksjoner som snapping til andre elementer, eller snappe til et grid system kan bli lagt til.

Prioritering: Could have.

Akseptansekriterier: Funksjonen anses som akseptert når systemet viser justeringslinjer i en X/Y akse og brukeren kan justere elementer

#### **14 Som bruker**

**Ønsker jeg at når jeg beveger på elementer så vises min posisjon i piksler  
Slik at jeg kan plassere elementer med pikselpresisjon.**

Beskrivelse: Programmet skal vise piksel koordinater til elementer når de er selecta.

Prioritering: Could have.

Akseptansekriterier: Funksjonene anses som akseptert når brukeren kan se piksel koordinater til elementene han jobber med. (ønsker å flytte?)

#### **15 Som bruker**

**Ønsker jeg å kunne strekke ut bilder (fit to frame)  
Slik at jeg kan fylle ut en forhåndsdefinert ramme.**

Beskrivelse:

Brukeren ønsker å kunne velge at et bilde skal auto-justeres til å passe inn i en ramme slik at størrelsesforholdet i bildet blir ivaretatt. Arealet i rammen som ikke blir fylt av bildet blir da gjennomsliktig [default].

// Skal man kunne endre farge på overflødig areal?

Prioritering: Should have.

Akseptansekriterier:

Funksjonen ansees som akseptert når brukeren kan velge å auto-justere ett bilde til å passe en ramme. Funksjonen garanterer ikke at dette gjør at resultatet

#### **16 Som bruker**

**Ønsker jeg å kunne justere/klippe ett bilde til å passe inn i en ramme  
Slik at det passer inn i en forhåndsdefinert ramme.**

Beskrivelse:

Brukeren ønsker å kunne flytte, resize, eller justere på et bilde for å få ønsket resultat i en ramme. Da ønsker brukeren å kunne se hele bildet i en gråtone/lav opacity utenfor rammen, og elementene i rammen i full farge.

Prioritering: Should have.

Akseptansekriterier:

Funksjonen ansees som akseptert når brukeren kan justere på ett bilde for å vise ønsket del/porsjon av bildet i en forhåndsdefinert ramme.

### **17 Som bruker**

**Ønsker jeg å kunne velge mellom et sett med filter på bildene mine i en template  
Slik at jeg kan videre sette preg på mine designs.**

Beskrivelse: Dette vil være en seleksjon med filtre som brukeren kan sette på bildene. Valgene vil bestå av de vanligste filtertypene, som f.eks svart-hvitt, sepia eller andre populære filter a la instagram. Det er planlagt at det vil være en egen dedikert knapp i grensesnittet for dette, eller tilsvarende (egen tab? etc).

Prioritering: Could have.

Akseptansekriterier:

Funksjonen ansees som akseptert når brukeren kan velge filter fra en seleksjon av populære filtre og bruke disse på bilder i templates.

### **18 Som bruker**

**Ønsker jeg å kunne endre farge gradient, lysstyrke, kontrast, etc, på bildene mine  
Slik at jeg kan gjøre justeringer for å øke kvaliteten på bildene mine(?).**

// Sub funksjonalitet av filter funksjonen. Ikke viktig funksjonalitet.

Beskrivelse:

Brukeren ønsker å kunne gjøre litt mer avanserte endringer på bildene sine som å endre kontrast, lysstyrke, og farge gradient.

Prioritering: Won't have.

Akseptansekriterier:

Funksjonen ansees som akseptert når brukeren kan gjøre mer avanserte endringer på bildene sine.

### **19 Som bruker**

**Ønsker jeg å lagre en template som jeg har redigert ferdig  
Slik at jeg kan sette det opp på skjermene mine**

Beskrivelse: Når brukeren har redigert templatene slik at den ser ut slik som brukeren selv ønsker, skal det være mulig å lagre templatene. I denne prosessen vil brukeren måtte velge navn på den ferdigredigerte templatene før den kan lagres. Vi har også vurdert annen informasjon som kan fylles inn på dette stadiet. Tags kan brukes til sortering i senere tid, f.eks at man gir et ferdigsnekret design en tag med navn "Jul" slik at det enklere kan filtreres.

Prioritering: Must have.

Akseptansekriterier: Kravet oppfylles når brukeren kan lagre en template (Fra template til skjermbilde, eksportering?) og at systemet lagrer dette på en slik måte at man kan finne resultatet i andre deler av systemet, som når man skal definere hvilke media som skal legges inn i en spilleliste på en gitt skjerm. Mellomlagring er positivt, men forøvrig ikke et krav for akseptanse. Andre idéer, som f.eks bruk av tags, er heller ikke et krav for akseptanse.

## **20 Som bruker**

**Ønsker jeg å redigere en template som jeg har lagret tidligere  
Slik at jeg kan gjøre endringer over tid.**

Beskrivelse: I grensesnittet der man ser designs man har laget tidligere vil det være mulig å kunne gå videre innover til den faktiske designeren med alle innstillinger fra det valgte tidligere ferdigstilte designet. Når man da er ferdige med å redigere designet igjen og går til å lagre, skal brukeren få velge om man skal overskrive den tidligere templatens med dette nye arbeidet, eller om man vil lagre det under et nytt navn.

Prioritering: Should have. Noe tvil om det burde være Must have, funksjonen er veldig nyttig for brukerne og er trolig ikke veldig krevende å gjennomføre så den bør prioriteres høyt blant andre Should have's.

Akseptansekriterier: Bruker må kunne navigere til designeren med forhåndssette innstillinger basert på valgt design/template. Dette må så kunne lagres, enten ved å overskrive det gamle designet eller lage en ny kopi. Det er best om begge alternativene eksisterer, men det vil fortsatt aksepteres hvis det bare er mulig å lage nye kopier. Hvis det ikke går an å lagre det som en ny fil, må det være mulig å duplisere designet på annet vis hvis brukerhistorien skal aksepteres.

## **Vedlegg 4 - Use Cases**

**UC1 - Bruker søker etter en template og velger så en template fra søkeresultatene (obs, lav prio da søk ikke er viktig for mvp)**

ID: UC01

**Beskrivelse:** På landingsside til designeren skal det være et søkefelt. I dette søkefeltet kan brukeren skrive inn søkeord. Brukerens input skal da kunne brukes av systemet til å se om det er noen templates som samsvarer, enten basert på templatens tittel eller tags som er lagt inn i templatene.

**Aktører:** Ordinær bruker av Screenz (kunde)

**Antagelser:** Brukeren har navigert seg bort til designeren fra et annet punkt i systemet og har ikke rørt landingssiden. Brukeren er logget inn.

**Brukshyppighet:** Middels-lav hyppighet. Ikke alle brukere vil benytte tjenesten, og de som velger å bruke den vil trolig bruke den svært hyppig i et døgn eller flere døgn, og så ikke røre tjenesten på en uke eller flere.

**Triggers:** Brukeren skriver i søkefeltet på landingssiden på designeren.

**Pre-conditions:** Man er på landingssiden til designeren.

**Post-conditions:** Landingssiden skal reflektere søkeresultatene, enten det er flere resultater eller ingen gyldige resultater. Når man er navigert bort til designverktøyet, så skal designverktøyet vite hvilken template som ble valgt. Dette skal reflekteres i valgmulighetene i designverktøyet.

**Main Course:**

1. Brukeren skriver et søkeord i søkefeltet (Se AC1)
2. Systemet henter ut resultater basert på samsvar mellom tags og søkeord.
3. Resultatene vises frem for brukeren.
4. Brukeren velger et av resultatene via museklikk (se AC2, AC3 og EX2)
5. Brukeren sendes videre til designverktøyet med informasjon om hvilken template som var markert.
6. Designverktøyet spør systemet om innstillinger og informasjon basert på brukerens valgte template.
7. Systemet returnerer informasjonen til designverktøyet. (se EX3)
8. Designverktøyet reflekter valgt template til brukeren.

**Alternate Courses:**

**AC1** Brukeren velger ikke å søke i det hele tatt og bare velger en av templatene som vises "by default"

1. Brukeren velger en template ved å dobbeltklikke på den for å gå inn til designeren.
2. Gå til MC5.

**AC2** Brukeren velger å se en preview av templatene fra kontekstmeny før de velger.

1. Bruker klikker knapp for kontekstmeny
2. Bruker klikker på «Preview»
3. Modalt vindu åpnes med en forhåndsvisning av templatene med forhåndsdefinerte variabler.
4. Bruker klikker på X i øvre høyre hjørne eller utenfor det modale vinduet for å få det bort.
5. Brukeren går tilbake til enten MC1 eller MC4.

**AC3** Brukeren velger å se på egenskapene til templatene fra kontekstmeny før de velger.

1. Klikk på knapp for kontekstmeny

2. Klikk på egenskaper
3. System henter informasjon om egenskaper og viser dette til brukerne i et nytt modalt vindu.
4. Bruker lukker det modale vinduet og havner tilbake på MC4

**AC4** Brukeren velger å ikke bruke noen av søkeresultatene og tømmer søkefeltet.

1. Brukeren begynner å slette input fra søkefeltet.
2. Systemet registrerer at det ikke lenger er input i søkefeltet og går tilbake til å vise "by default" templates.
3. Gå til MC1 eller AC1.

#### **Exceptions:**

**EX1** Systemet returner ikke et gyldig resultat etter et søk

1. Database returnerer ingen gyldige resultat (tom liste)
2. Systemet ser at listen er tom og returnerer et forhåndsdefinert svar til brukeren.
3. Svaret fra systemet vises frem til brukeren. ("Ingen templates samsvarer med ditt søk")
4. Bruker går til AC4.

**EX2** Systemet klarer ikke å hente informasjonen ut fra systemet.

1. Hvis error i system: Log error i systemlog. Hvis systemet ikke kan nås: Ha en timeout i front-end etter gitt mengde ventetid.
2. Informer brukeren om at det ikke er mulig å hente ut forespurt informasjon med feilmelding basert på enten timeout eller system error.
3. Forlat designverktøyet og naviger tilbake til landingsside for designer.

## **UC2 - Bruker velger et aktivt element for redigering i en template ved å klikke på elementet.**

**ID:** UC02

**Beskrivelse:** Når man skal bruke designverktøyet er det to alternativer for hvordan vi kan løse dette. Vi kan enten la brukeren velge aktivt element ved museklikk, eller vi kan ha det statisk der brukeren ikke berører templaten/viewet i det hele tatt foruten tekstfelt og innstillinger. Planen er at innstillinger for det valgte elementet kommer opp i sidemeny til høyre. Denne use-casen beskriver førstnevnte, der man "aktiverer" et element ved å klikke på det i verktøyet. Hvilke elementer i en template som er valgbare avhenger av hva templaten selv har definert som klikkbart/redigerbart område.

**Aktører:** Ordinær bruker (screenz-kunde)

**Antagelser:** Det antas her at løsningen vi går for er den der elementene klikkes på først for å aktivere elementet og deretter redigere utseende og innstillinger for elementet.

**Brukshyppighet:** Høy. Når man først er i designverktøyet er det sannsynlig at bruker klikker frem og tilbake på elementene flere ganger før de blir fornøyde.

**Triggers:** Bruker trykker på et forhåndsdefinert redigerbart element i

**Pre-conditions:**

**Post-conditions:**

**Main Course:**

1. Template lastes inn i designverktøyet fra designmenyen.
2. Designverktøyet ser hvilke elementer som kan redigeres, og markerer disse med en egen ramme for å insinuere at de er "interactable".
3. Bruker klikker på et av elementene.
4. Designverktøyet setter elementet som ble klikket på som aktivt element.
5. Meny/Kolonne på høyre-/venstreside fylles inn med informasjon om hva som kan redigeres basert på det aktive elementet.

**Alternate Courses:**

**Exceptions:**

**EX1** Klarer ikke å hente ut innstillinger om hvilke elementer som kan redigeres fra elementet som ble klikket på.

## **UC3 - Bruker velger å redigere tekst i en valgt template i designerverktøyet**

**ID:** UC03

**Beskrivelse:** Brukeren er inne i designerverktøyet og ønsker å redigere tekst i en valgt template. For å gjøre det må brukeren først ha trykket på tekstfeltet som han/hun skriver inn i. Planen er at vi skal ha inline editing, det vil si at når brukeren trykker på tekstfeltet og begynner å skrive så blir teksten redigert der den står.

**Aktører:** Ordinær bruker (screenz-kunde)

**Antagelser:** Brukeren har navigert seg til designerverktøyet og har trykket på et tekstfelt som han/hun ønsker å redigere.

**Brukshyppighet:** Middels-Høy. Det å endre tekst i en template er en av kjernefunksjonene av systemet.

**Triggers:** Brukeren trykker på et av tekstfeltene.

**Pre-conditions:** Brukeren er i designerverktøyet.

**Post-conditions:** Når brukeren har redigert ferdig ønsket tekst, skal den nye teksten vises i rett tekstfelt.

**Main Course:**

1. Brukeren er i designerverktøyet.
2. Brukeren trykker på et av tekstfeltene og begynner å skrive.
3. Teksten blir lagret underveis
4. Når brukeren er ferdig med redigeringen, kan han/hun trykke på et annet element og gå videre i redigeringsprosessen. (Se AC2)

**Alternate Courses:****AC1** Brukeren trykker seg vekk fra tekstfeltet

1. Teksten blir lagret

**AC2** Brukeren kan trykke på lagre knapp

1. Dersom brukeren trykker på lagre knapp i designerverktøyet blir templatet lagret som view
2. Tekstfeltet som brukeren hadde valgt er deselected nå.

**Exceptions:****EX1** Dersom brukeren bruker et font som ikke støtter spesifikke tegn/bokstaver. Eks. æøå

## **ID: UC04 - Bruker lagrer et ferdig tilpasset view i designerverktøyet.**

**Beskrivelse:** Når brukeren er ferdig i designeren skal det være mulig å lagre det ferdig tilpassede viewet slik at det kan brukes andre steder i systemet, f.eks legges inn i en aktiv playlist. Når brukeren trykker på lagre knappen skal det komme en bar hvor det konfirmeres for brukeren at viewet er lagret og at man skal kunne få muligheten til å navigere seg ut.

**Aktører:** Bruker (Screenz kunde)**Antagelser:** Bruker er logget inn og har redigert ferdig et nytt view i designeren og ønsker å lagre det.**Brukshyppighet:** Lav til middels lav.**Triggers:** Brukeren trykker på save knappen i designeren.**Pre-conditions:** Brukeren er ferdig med å lage et view i designeren.**Post-conditions:** Det nye viewet skal bli lagret i databasen med navn og ID. Viewet skal vises i view menyen og kunne brukes i en aktiv spilleliste.**Main Course:**

8. Bruker trykker på knapp for å lagre viewet.
9. Informasjon som navn, bilde og annet content brukeren har definert selv hentes ut fra variablene som brukes av systemet og skrives til et nytt objekt. (Se AC1)
10. Systemet kaller på back-end API og sender over objektet (Se EX1)
11. Server mottar objektet og kjører metode for å lagre informasjon i databasen og venter på konfirmasjon fra databasen. (Se EX2)
12. Det lagres i databasen og databasen sender konfirmasjon tilbake til server.
13. Server sender en melding til front-end om at viewet har blitt lagret.
14. I front-end dukker det opp en bar med konfirmasjon om at viewet er lagret med mulighet for videre navigasjon.

**Alternate Courses:**



**AC1** Dersom brukeren prøver å lagre det samme navnet som på et allerede eksisterende view.

4. Det skal komme en pop-up hvor brukeren har mulighet til å velge om en ønsker å lagre over et allerede eksisterende view eller lagre view under nytt navn.
5. Bruker velger å overskrive den eksisterende filen (Se AC2)
6. Gå til MC3

**AC2** Bruker velger nytt navn på overskrivende fil

4. Pop-up vindu åpnes med mulighet for å skrive et nytt navn.
5. Bruker trykker på ny lagreknapp definert i nye vinduet.
6. Gå til MC3

### **Exceptions:**

**EX1** Systemet klarer ikke å lagre den ferdigredigerte templatene.

4. Try/Catch (eller tilsvarende i PHP) der catch blir aktivert pga mangler i/manglende fil eller annen error i systemet som f.eks nettverksbrudd/timeout
5. Log error i systemlog. Riktig error melding blir hentet basert på error code. Errormelding hentes fra en liste definert i koden med pregenererte meldinger basert på errorcode.
6. Systemet viser en error i form av en pop-up til brukeren i designverktøyet.

**EX2** Noen av verdiene som sendes over i objektet er av ugyldig type

5. Metode for lagring i database sjekker at informasjonen har rett type. I dette tilfellet er informasjonen feil.
6. Skriv ut errorcode basert på hvilket element som fikk feil under kjøring av metoden.
7. Send koden tilbake som svar på API-kall
8. Systemet skriver ut meldingen i en pop-up melding til bruker.

## Vedlegg 5 - Instruksjoner for oppsett/gjenskapelse av error for Panteon

### **STEP 0 - Create droplet**

SKRIV INN HVER ENESTE COMMAND HER!

Droplet, One-click app med 16.04 LAMP

ip = xxx.xxx.xxx.xxx

user = root

base pw = xxxxxxxxxxxxxxxxxxxx

pw = xxxxxxxxxx

For å koble til:

ssh root@xxxxxxxx

sudo apt-get update

sudo apt-get upgrade (tar noe tid)

### **STEP 1 - MySQL install**

```
sudo ufw allow 3306
```

```
sudo nano /root/.digitalocean_password
```

Kopier ut passordet til MySQL

På denne serveren:

```
root_mysql_pass="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```

Kjør: mysql\_secure\_installation

```
mysql -u root -p
```

```
paste root_mysql_pass xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
GRANT ALL PRIVILEGES ON *.* TO 'screenzadmin'@'%' IDENTIFIED BY 'xxxxxxx';
```

```
CREATE DATABASE screenz_web COLLATE utf8mb4_general_ci;
```

```
USE screenz_web;
```

```
FLUSH PRIVILEGES;
```

For å tillate tilkobling fra alle hosts, må man endre bind address i mysqld.cnf:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
F6 "bind"
```

Endre bind-address til 0.0.0.0

Fjerne linje med bind-adress eller kommenter ut fra filen.

Pull prosjektet fra git (insert git link her). Alternativt kan dette gjøres senere via SFTP som er på by default i DigitalOcean.

Legg til tilkobling i MySQL workbench med ip xxxxxxxxxxxxxxxx, bruker screenzadmin og pw xxxxxxxxxxxx

kjør .sql fil i MySQL workbench mot serveren (husk USE screenz\_web;)

## **STEP 2 Installer ffmpeg og composer.**

```
apt install composer
```

```
sudo apt-get install -y ffmpeg
```

## **STEP 3 Installer node.js, npm og angular1.2.7**

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

Mulig konflikt med npm, går i infinite loop pga ghost folder.

bør fungere til tross, men vær obs.

```
npm install -g node-gyp
```

```
npm install -g node-sass
```

```
npm install -g @angular/cli@1.2.0
```

#### **STEP 4 Installere og konfigurere PHP7.1**

PHP 7.0 er installert fra før. I stedet for å avinstallere kan vi fortelle apache å bruke en nyere versjon.

Først må vi installere ny versjon:

```
sudo apt-get install software-properties-common
```

```
sudo add-apt-repository ppa:ondrej/php
```

```
sudo apt-get update
```

```
sudo apt-get install php7.1
```

Deretter må man enable extensions/modules. De som må aktiveres er:

CURL

Intl

Mbstring

Mcrypt

Mysqli

Pdo\_mysql (by default?)

Xmlrpc

```
phpenmod <modnavn>
```

Kan være CURL mangler, OBS!

```
sudo apt install php7.1-modulnavn
```

a la

```
sudo apt install php7.1-curl
```

Mcrypt er kanskje ikke installert OBS OBS OBS

Gå inn i php.ini og endre disse verdiene:

```
max_input_time = 120
```

```
memory_limit = 512M
```

```
file_uploads = On
```

```
upload_max_filesize = 2048M
```

```
max_execution_time = 300
```

```
sudo apt install php-pear
```

Noe har nå gått galt med php-installasjonen. Det virker som php-pear har installert sin egen versjon av php eller noe.

Planen nå er enten å reinstallere alt av PHP eller å prøve php-brew.

Etter reinstall ble følgende kommando kjørt for php pear:

```
$ wget http://pear.php.net/go-pear.phar  
$ php go-pear.phar
```

Nå får jeg error om at XML extension not found.

```
Sudo apt-get install php7.1-xml
```

Kjørte pear install helt greit. Brukte default paths på alt, Pear endret php.ini.

Byttet php-versjon på apache2 med følgende kommando

```
a2dismod php7.0
```

```
a2enmod php7.1
```

```
service apache2 restart
```

Lurt å gjøre dette før man begynner å rote med php i det hele tatt!

Lastet ned libzmq3 med følgende kommando:

```
echo "deb
```

```
http://download.opensuse.org/repositories/network:/messaging:/zeromq:/release-stable/xUb
```

```
untu_16.04/ ." >> /etc/apt/sources.list
```

```
wget
```

```
https://download.opensuse.org/repositories/network:/messaging:/zeromq:/release-stable/xUb
```

```
untu_16.04/Release.key -O- | sudo apt-key add
```

```
apt-get install libzmq3-dev
```

Nødvendig for å installere zmq med PECL.

Naturligvis følger pecl zmq install

```
sudo pecl install zmq-beta
```

Ser ut til å ha fungert helt fint.

Legge til extension i php.ini

Copy paste inn i php.ini:

```
extension=zmq.so
```

## **STEP 5 Videre installasjon av angular**

For å prøve å fikse dependencies, laster vi ned hver eneste pakke som er del av angular og installerer globalt.

```
npm install -g @angular/animations@4.2.0
```

```
npm install -g @angular/common@4.2.0
```

```
npm install -g @angular/core@4.2.0
```

```
npm install -g @angular/compiler@4.2.0
```

```
npm install -g @angular/forms@4.2.0
```

```
npm install -g @angular/http@4.2.0
```

```
npm install -g @angular/platform-browser@4.2.0
```

```
npm install -g @angular/platform-browser-dynamic@4.2.0
```

```
npm install -g @angular/router@4.2.0
```

```
npm install -g @angular/upgrade@<4.2.0
```

```
npm install -g rxjs@5.0.1
npm install -g zone.js@0.8.4
```

npm WARN @angular/core@4.2.0 requires a peer of rxjs@^5.0.1 but none is installed. You must install peer dependencies yourself.

npm WARN @angular/core@4.2.0 requires a peer of zone.js@^0.8.4 but none is installed. You must install peer dependencies yourself.

Kjørt npm install i mappen til prosjektet. Fikk følgende warnings:

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN fsevents@1.2.0 had bundled packages that do not match the required version(s). They have been replaced with non-bundled versions.

npm WARN @ng-bootstrap/ng-bootstrap@1.1.2 requires a peer of @angular/common@^5.0.2 but none is installed. You must install peer dependencies yourself.

npm WARN @ng-bootstrap/ng-bootstrap@1.1.2 requires a peer of @angular/core@^5.0.2 but none is installed. You must install peer dependencies yourself.

npm WARN @ng-bootstrap/ng-bootstrap@1.1.2 requires a peer of @angular/forms@^5.0.2 but none is installed. You must install peer dependencies yourself.

npm WARN @angular/forms@5.2.10 requires a peer of @angular/platform-browser@5.2.10 but none is installed. You must install peer dependencies yourself.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.0 (node\_modules/fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.0: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

Prøver å kjøre opp prosjektet med ng serve

Får følgende errors:

<https://pastebin.com/nmEvSHi9>

UNRELATED TO STEPS:

ng --version på fersk install av ubuntu uten LAMP.

Bare installert node, node-gyp, node-sass og angular/cli@1.2.0:

@angular/animations: error

@angular/common: error

@angular/compiler: error

@angular/core: error

@angular/forms: error

@angular/http: error

@angular/platform-browser: error

@angular/platform-browser-dynamic: error

@angular/router: error

@angular/cli: error

@angular/compiler-cli: error

@angular/language-service: error

## Vedlegg 6 - Backlog

### 1.0.A Analyse:

A.0.1 Hva skal produktet til slutt være? Veldig brukervennlig! For alle typer brukere.

A.1 Få en oversikt over designmessige funksjoner

Hvor har vi hentet det fra?

Hvorfor bruker vi disse?

A.2 Dokumentere bruksområder, brukerbehov, og forventet forkunnskaper for brukere.

Hvilke avgjørelser har vi tatt på bakgrunn av dette?

A.3 Kjøre en "test" eller heuristikk test på funksjonene vi ser for oss.

### 2.0.D Design:

InVision prototype + intervju:

InVision - Lag side for å laste opp bilde i view.

D.1 Lage nye utkast for designermeny (template/view lister)

Vurder mulighet for felt til høyre med muligheter for å filtrere ut templates/views

Pop-up/vindu i forgrunnen for å velge om man vil til templates eller view i

designermeny, må da være en grei måte å kunne navigere fra den ene til den andre etter at valget er tatt.

D.2 Implementer nytt utkast for designer designermeny basert på feedback fra Rune.

D.3 Lage en grafisk forbedret versjon av prototypen for brukertesting.

- Hvor mye grafisk bedre? Tenk over at hvis det ser for profesjonelt ut så er ikke brukere like villige til å gi negativ feedback

D.3.1 Implementer forbedret versjon av designmeny (tidligere splashscreen).

D.3.2 Implementer forbedret versjon av designer.

D.3.3 Implementer endpoint(?) for prototypen.

- Dette blir da der bruker setter ferdig view inn i spilleliste og legger ut på skjerm

D.4 Oppdatere highlighting/navigasjon i den grafisk oppdaterte prototypen.

D.5 Gjennomgang av prototype med Rune for feedback.

D.5.1 Implementer eventuelle endringer basert på feedback fra Rune.

D.6 Gjennomføre tester/intervju i starten av uke som starter 23.04.

D.7 Endre prototype basert på feedback fra brukertesting

D.8 Utarbeide nødvendige diagrammer

- State chart
- Klasse diagram
- Sekvens diagram

### **3.0 Implementasjon:**

#### **3.1.OS Oppsett av server på Digital Ocean**

- OS.1 Sette opp fungerende LAMP-stack på DO-server
- OS.2 Sette opp API-server (Symfony) på DO-server
- OS.3 Sette opp Angular app på DO-server
- OS.4 Kjøre opp applikasjonene på DO-server
- OS.5 Sette opp Virtual Hosts på DO-server

#### **3.2.UF Utvikling av frontend**

- UF.1 Lage egen side for designeren som kan navigeres til
  - Være en knapp i sidebar eller lignende.
- UF.2 Lage en template for testing.
- UF.3 Vise frem ferdigredigerte views sammen med andre mediefiler slik at de kan legges inn i en spilleliste.

#### **3.3.DM Utvikling av designermeny**

- DM.1 Implementere API kall for søk mot backend.
  - DM.1.1 Implementere filter knapper for tilpasset søk mot backend.
- DM.2 Parser for informasjon fra databasen for å vise template i designermenyen
- DM.3 Parser for informasjon fra databasen for å vise views i designermenyen
- DM.4 Lage klikkbare ikoner basert på templates fra database
- DM.5 Lage kontekstmeny på templateikoner
  - DM.5.1 Lage funksjon for å vise preview av en template i kontekstmeny
  - DM.5.2 Lage funksjon for å vise egenskapene til en template i kontekstmeny
  - DM.5.3 Lage klikkbare ikoner basert på views fra database
- DM.6 Lage kontekstmeny på viewsikoner
  - DM.6.1 Lage funksjon for å vise preview av ett view i kontekstmeny
  - DM.6.2 Lage funksjon for å vise egenskaper av en view i kontekstmeny
  - DM.6.3 Lage funksjon for å slette ett view i kontekstmeny
  - DM.6.4 Lage funksjon for å gi view nytt navn i kontekstmeny
  - DM.6.5 Lage funksjon for å duplisere et view i kontekstmeny
- DM.7 Lage funksjon for å åpne view i designverktøyet i en ny fane.
- DM.8 Gjøre nytt kall på API for henting av template/view ved å scrolle nedover
- DM.9 Navigasjon til designverktøyet i lag med informasjon om hva som ble trykket på

#### **3.4.DV Utvikling av designverktøy**

- DV.1 Funksjonalitet for valg av elementer i templatene.
- DV.2 Implementere sidemeny for å vise innstillinger av valgte elementer
- DV.3 Endre tekst element i ett view
  - DV.3.1 Endre font i tekst i ett view
  - DV.3.2 Endre skriftstørrelse i tekst i ett view

- DV.3.3 Endre farge på tekst i ett view
- DV.3.4 Endre alignment på teksten (mot høyre, mot venstre eller sentrert) ?
- DV.4 Endre bilde element i ett view
- DV.4.1 Endre posisjonering av bilde innenfor forhåndsdefinert bilderamme (fit/fill/custom)
- DV.4.2 Endre zoom på bilde i ett view
- DV.4.3 Endre/legge til filter på bilde i ett view
- DV.5 Endre bakgrunn i ett view
- DV.5.1 Endre fokuspunkt til bakgrunnen i ett view.
- DV.5.2 Laste opp bakgrunnsbilde.
- DV.5.3 Sette bakgrunnsfarge.
- DV.6 Endre maskable object i ett view
- DV.6.1 Endre fargen på maskable object
- DV.6.2 Endre mask på maskable object
- DV.7 Laste opp bilde til designeren.
- DV.8 Implementere mellomlagring i designeren.
- DV.9 Implementere mulighet for å flytte på elementer
- DV.9.1 Implementere mulighet for å flytte på bilderamme
- DV.9.2 Implementere mulighet for å flytte på maskable objects
- DV.9.3 Implementere mulighet for å flytte på tekst
- DV.10 Implementere undo/redo funksjonalitet.
- DV.11 Implementere navngivningsfunksjon
- DV.12 Implementere lagre funksjon

### **3.5.UB Utvikling av backend**

- UB.1 Dokumentere hvilken informasjon designeren trenger for å laste inn en template
- UB.2 API for å hente ut informasjon om template fra database basert på template-ID
- UB.3 API for å hente ut tidligere lagrede views fra database basert på bruker-ID.
- UB.4 API for å Lagre view i databasen.
- UB.5 Implementere mellomlagring av views i database.
- UB.6 API for å hente ut søkeresultater av templates
- UB.6.1 Søkeresultater basert på navn på template
- UB.6.2 Søkeresultater basert på tags i template
- UB.7 Filtrering av søkeresultater basert på ønsket orientering (landscape/portrait)
- UB.8 Utarbeide databasemodeller
- UB.8.1 Databasemodell for template
- UB.8.2 Databasemodell for view
- UB.8.3 Databasemodell for nylig brukte templates
- UB.9 Lagre view som .html-fil basert på instruksjoner fra designverktøy
- UB.10 Legge til views i en spilleliste.

Hva skal være de første templatene som skal vises? Mest populære? Sist brukte? Hvordan skal vi ha kontroll over hvilke templates som nylig er blitt brukt av en bruker? Associative entity i database med composite key (brukerID og TemplateID) og timestamp.

Sortering av templates vha sidebar/sidemeny a la komplett/finn?



#### **4.0.DOK Dokumentasjon:**

Levere utkast av bacheloroppgaven til Janis (praktisk gjøremål eller dokumentasjon?)

DOK.1 Skrive innledning i rapport

DOK.1.1 Skrive noe om kvalitetssikring

DOK.2 Legge brukerhistorier til i rapport, med forklaring til hvordan vi kom opp med disse.

DOK.3 Ferdigstille statechart-diagrammer og klassediagrammer, og legge disse til i rapport med beskrivende tekst.

DOK.3.1 Legge til Systemkrav

DOK.3.2 Legge til funksjonsliste

DOK.4 Dokumentere problemene vi har hatt til nå

DOK.5 Ferdigstille use cases, og legge disse til i rapport med beskrivende tekst.

DOK.6 Rapport: Forklare hvordan vi gikk fram for å finne designet vårt

DOK.7 Inkludere noen av designskissene i rapporten med tilhørende forklaring og relatert brukerhistorie/Use-Case

DOK.8 Skrive om implementasjonsprosessen i rapporten.

DOK.9 Skrive om det endelige resultatet og konklusjon. Legge til skjermbilder av resultatet?

DOK.10 Skrive refleksjoner av individuelt arbeid i prosjektet.

DOK.11 Skrive Sammendrag

DOK.12 Skrive om bruk av AirTable og bytte bort fra Jira og Gantt Chart.

DOK.13 Skrive om oppsett av server og bytte bort fra serverløsningen.

DOK.14 Skrive om oppsett av nytt prosjekt i samme rammeverk (reverse engineering)

DOK.15 Skrive om implementasjon av designverktøyet. (blir flere underkategorier her underveis).

DOK.16 Skrive om prototyping gjort i InVision og Figma.

DOK.17 Skrive om brukertesting og resultater av det.

DOK.18 Skrive om vårt samarbeid, kommunikasjon og kontakt med Rune.

DOK.19 Skrive om vår kommunikasjon med Panteon

### **5.0.P Praktiske gjøremål:**

P.1 Skaffe nøkkelkort til CoWorx

P.2 Skaffe papirtavler/rollups, evt det som egner seg best for gruppearbeid på tvers av lokaler.

Anders har skaffet endel, må se an om det er nok eller om annet også bør fremskaffes

P.3 Booke grupperom/møterom på CoWorx via Rune flere dager i uken.

P.4 Gi en kopi av backlog og første sprint plan etter veiledningsmøte til Rune og Janis for verifisering

P.5 Tredje veiledningsmøte med Janis

P.6 Prøveeksamen for Janis

P.7 Klargjøre prosjektet for expo 23.05

## **Vedlegg 7 - Referater fra Veiledningsmøtene**

### **Første veiledningsmøte - 16.02.2018**

Vi har ingen agenda per idag. Det blir da "backlog" til neste møte.

Vi prøver å justere oppgaven, og konkretisere den. Tatt vekk redigering av video, og heller fokusere på bilde redigering. Vi får en kopi og general release den 20. Februar. Vi bør vente med programmering til vi vet hva som skal gjøres. Vi skal kjøre intensivt uke 8.

- Sendt link til google docs til Janis -

Burde skrive ned business case. Deretter brukerhistorier og funksjonsliste. Så gjøre papirprototype. Vi har skrevet ned research stage presprint. Kvalitetssikring.

Må definere en konkret prosjektbeskrivelse. Sende den til Janis og Rune fortløpende.

Vi skriver ned plan og utifra det hva vi har fått gjort og hva vi sliter med, og hvordan Rune og Janis kan hjelpe oss til å holde prosjektet i spor.

Teste opp brukerhistorier mot Rune og hva han tenker og syns.

Dagbok føring - skrive ned prosjekt dagbok hvor alt som vi har snakket om på møter og klarer å spore fram tilbake når vi skal begrunne valg seinere og reflektere i rapporten seinere.

Hva skal vi fokusere på - Finn den ene ideen som vi skal fokusere på. Spek den ut og prøv den ut. Hypotese - bevise det eller motbevise det.

Rapportskrivning. Kan beskrive hva vi har gjort alt fra problem til løsning i riktig timeline. Og beskrive hva som har vært problem.

## Andre veiledningsmøte - 10.04.2018

Vi føler at vi er veldig utsatt.

Bare fokuser på det dere kan løse. Ikke bruk tiden på å tenke på hvordan det burde vært.

Prøver å holde gantt chart oppdatert. Glidd ut etter påske. Samme med AirTable.

Gått litt på tomgang. Lett å distrahere.

Oppdeling av oppgaver.

Arbeid i par på det tekniske. Mer kommunikasjon.

Hatt et felles rom - Felles papirtavle?

Freelancere - Svært få greier det. De sitter i hjemmekontoret. Sitter på universitetet, der dere har andre oppgaver. Annet du må gjøre her, der etc. Klarer ikke å fokusere på oppgaven som faktisk skal gjøres.

Booke grupperom. Tilgang på bygget etter stengetid.

Sitte mye mer på CoWorx?

Booke grupperom via Rune.

Få tak i nøkkelkort på bygget!

Separere fag på universitetet bedre.

Redd for å eksponere oss selv for feil. Hvis man ikke eksponeres, lærer man ingenting.

InVision - Skisser må reworkes. Særlig siden der man velger template/view.

Templatesiden er ikke sånn Rune ville løst den. Ikke kjent user pattern.

Funksjonelt så er den ok. Vår idé var å følge slik google docs/sheets/ etc ser ut.

Lage prototype som kan testes ut mot brukere?

Jobbe med design og teknisk/programmering samtidig:

Samtidig som man jobber med design, finn ut det tekniske! Kan vi bruke x rammeverk til å løse designforslag y? Lage tekniske demoer for å vise at designet fungerer i prinsipp.

Hva ønsker Rune? Gå inn på testversjon, lag ny plakat, endre tekst, lagre den og så kan den legges til som innhold i en spilleliste.

Tasks er for løst definert. Vi må være mer spesifikke med hvordan vi definerer våre oppgaver. Gjerne mindre oppgaver!

Selvklebende rollups. Kjøper de på kontorutstyrbutikk (???) . Ikke staples men den type butikk.

Når det gjelder designsiden. Bare vise liggende design.  
Hva skjer når vi bikker 1000+ templates?

Sette av dato for brukertest.

Når det gjelder intervju: intervjuobjekt føler seg ofte forpliktet til å gi feedback. Det fører til at mye feedback ofte er dårlig.

Veldig klart: Hva ønsker vi av Rune?  
Hva ønsker vi av Janis?

Send e-post til Janis hvis vi ønsker at noe skal sees på.

Hvilke dager passer best for Janis? Fredag passer best.

## Tredje veiledningsmøte - 11.05.2018

Vi føler vi ligger godt an.

Janis har vært inne å lest på rapporten. "Anonymous Duck".

Rapporten ligger greit an, men må fortsatt jobbe frem mot expo.

Nokså effektivt gruppearbeid. Noe deling på rapportarbeid, design, teknisk.

Bedre til å bruke scrum. Sprint planning + review/retrospective.

Håper på tilbakemelding fra Janis sånn at vi kan forbedre rapporten fra Janis.

Rune kommer inn og forteller status. Vi blir enige om å stoppe

Rune sier at han er fornøyd at vi sitte mer på CoWorx og jobbe.

Janis spør Rune om budsjettering - han sier at det ligger litt an på hvor vi er mellom first release og hvor vi ligger nå.

Vi har underestimert oppgaver og overestimert andre oppgaver.

Vi ønsker å fokusere på expoen for at det skal se bra ut. Og heller jobbe med backend fram mot eksamen 05.06.2018.

Ikke noe konkret prosjektplan men vi planlegger fra sprint til sprint.  
Dokumentere estimering i rapporten.  
Ha et regnskap på estimering fra sprintene vi estimerte.  
Hvordan kunne vi estimert bedre. F.eks bruk av planning poker?

Refleksjon var bra,  
Sprint blir mer som dokumentasjon, janis sier at vi kan ta det ut eller la det ligge.  
Fare for overlapp.

Refleksjon til slutt, hva kunne vi gjort bedre, oppsummering. Dra inn refleksjon fra sprintene.

Rune sier at vi har begynt å jobbe mer i siste perioden. Rune sier at han er usikker om han skal ta med seg koden videre. Det er opp til Panteon å implementere funksjonaliteten inn i screenz.  
Rune tar med seg "erfaringene".

Gjerne vise et resultat til Rune etter muntlig eksamen, eller noe før.  
Vi kommer til å følge med på Screenz i fremtiden.

Har vist frem grafisk prototype, men ikke teknisk til Rune. Tilsendt Zeplin-tilgang.  
Styringsgruppemøte. Målet er å sette opp en status.  
Litt for lite satt opp som en "presentasjon".

Eksamen blir som en "status" for siste versjon av vårt prosjekt.  
Vi skal vise og fortelle hva vi har planlagt og hvor langt vi har kommet.

En del bytte mellom verktøy. I en vanlig bedrift ville ikke dette vært noe vanlig.

Skriftlig tilbakemelding på rapporten fra Janis.  
Kvalitet har mye å gjøre med kontekst. Teknologiske delen er ofte der "kvalitet" ligger.  
Continuous Integration, skrive om dette.  
Brukergrensesnittdelen er bra.

Nevner Mathiessen, men er ikke koblet sammen godt. Da gjerne FACTOR. Teori i kontekst!  
Mer ting fra whiteboards. Bilder fra de?

Vi må estimere totalt timer brukt.

Prøveeksamen med Janis.  
31. Mai om morgningen. Før kl. 11. Bør gå greit. Hvis noe f.eks mangler i rapporten så blir det noe som spørres om på eksamen. Klassediagram f.eks. Hvis det mangler, så blir man gjerne spurt om det.

Exportere bildet til URL slik at vi kan vise fram i Screenz. Kan prøve å hoste det på dropbox, og få URL til å vise på Screenz.

Ta en Pc fra Rune med Screenz software slik vi kan vise fram Screenz på TV på Expo.  
Ta TV-en i gangen.