



NTNU

Kunnskap for en bedre verden

Fakultet for arkitektur og design

Institutt for design

Studieprogram Bachelor i Webutvikling (BWU)

VÅR 2019

Prosessrapport for utvikling av FAQ modul for Dokflyt

Forfattere

Benjamin Kwiek

Nana Huynh Vo

Ruben Kvamme

Innhold

1.	[OBJ]	8
1.1	OVERORDNET PROSESS	8
2.	PLANLEGGINGSFASEN	10
2.1	ARBEIDSFlyT I PLANLEGGINGSFASEN	10
2.2	SAMMENSETTING AV GRUPPEN	11
2.3	TEAMROLLER OG PERSONLIGHETSTYPER	11
2.4	16PERSONALITIES OG KOGNITIVE FUNKSJONER	12
2.4.1	<i>Resultat</i>	12
2.4.2	<i>Dimensjonene</i>	13
2.5	KONKLUSJON	15
2.6	VERKTØY	15
2.6.1	<i>Mendeley</i>	15
2.6.2	<i>Trello</i>	16
2.6.3	<i>Slack</i>	16
2.6.4	<i>Adobe XD</i>	17
2.6.5	<i>Bitbucket</i>	18
2.6.6	<i>Sourcetree</i>	18
2.6.7	<i>NPM</i>	19
2.6.8	<i>Visual Studio Code</i>	19
2.7	GRUPPEREGLER	20
2.8	FØRSTE MØTE MED VEILEDER	21
2.9	PROSJEKTPLAN	21
2.10	KOMMUNIKASJON OG MØTER	21
2.11	INFORMASJONSUTHENTING	21
2.12	EKSTERNE RESSURSER	22
2.13	RAPPORTSKRIVING OG LOGGSKRIVING	22
2.14	MØTE MED PRODUKTEIER	23
2.15	PROBLEMER I PLANLEGGINGSFASEN	23
3.	DESIGN	23
3.1	PERSONAS	24
3.2	SCENARIOS	24
3.3	STORYBOARD	24

3.4	LO-FIDELITY PROTOTYPE	24
3.4.1	<i>Arbeidsflyt i Lo-Fi prototypefasen</i>	25
3.4.2	<i>Idemyltring papirprototype</i>	25
3.4.3	<i>Første iterasjon Lo-Fi prototype</i>	25
3.4.4	<i>Møte med veileder og produkteier</i>	26
3.4.5	<i>Reevaluering av første iterasjon Lo-Fi prototype</i>	26
3.5	ANDRE ITERASJON LO-FI PROTOTYPE	27
3.5.1	<i>Brukertest andre iterasjon Lo-Fi prototype</i>	27
3.5.2	<i>Reevaluering etter brukertest</i>	27
3.6	TREDJE ITERASJON	28
3.7	PROBLEMER I LO-FIDELITY PROTOTYPEFASEN	28
3.8	HI-FIDELITY PROTOTYPE	29
3.9	ARBEIDSFlyt I HI-FIDELITY PROTOTYPEFASEN	29
3.10	FØRSTE ITERASJON HI-FI PROTOTYPE	29
3.10.1	<i>Brukertest første iterasjon Hi-fi prototype</i>	31
3.10.2	<i>Reevaluering etter brukertest</i>	31
3.11	ANDRE ITERASJON HI-FI PROTOTYPE	32
3.11.1	<i>Reevaluering etter møte med veileder og produkteier</i>	32
3.12	TREDJE ITERASJON HI-FI PROTOTYPE	33
3.13	PROBLEMER I HI-FI PROTOTYPE FASEN	33
4.	KODING	33
4.1	ARBEIDSFlyt I KODEFASEN	34
4.2	DIVERSE RUNDT KODEFASEN	34
4.2.1	<i>Nedlasting av nødvendige pakker for miljø</i>	34
4.2.2	<i>Arbeid i Git</i>	35
4.2.3	<i>Offline</i>	35
4.2.4	<i>Navngivning av filer</i>	35
4.2.5	<i>Bugfix</i>	36
4.3	FØRSTE ITERASJON	37
4.3.1	<i>Møte med produkteier og veileder samt kodevalidering</i>	39
4.3.2	<i>Reevaluering etter møte med produkteier og veileder samt kodevalidering</i>	39
4.4	ANDRE ITERASJON	39
4.4.1	<i>DF-header</i>	41
4.4.2	<i>Media Queries</i>	42
4.4.3	<i>Routing</i>	43
4.4.4	<i>Brukertest kode</i>	44
4.4.5	<i>Reevaluering etter brukertest</i>	45

4.5	TREDJE ITERASJON	45
4.6	PROBLEMER I KODEFASEN	46
4.7	SLUTTFØRING AV KODE.....	46
4.8	OPPLASTING AV OPPGAVE.....	47
4.8.1	3.10 Case study.....	47
5.	AVSLUTNING.....	48
5.1	AVSLUTTENDE TANKER	48
6.	REFERANSER.....	49

Figuroversikt

Figur 1 Forklaring av det generelle forløpet i prosjektet	8
Figur 2 generell forklaring av de iterative fasene i prosjektet	8
Figur 3 Arbeidsflyt i planleggingsfasen	9
Figur 4 Skjermdump av Trello-Tavle for prosjekt	14
Figur 5 Skjermdump av Slack gruppen i prosjektet	15
Figur 6 Skjermdump av prototype i Adobe XD	16
Figur 7 Skjermdump av repository i Bitbucket	16
Figur 8 Skjermdump av GUI i Sourcetree	17
Figur 9 Skjermdump av kode i Visual Studio Code	18
Figur 10 arbeidsflyt i lo-fi prototypingfasen	23
Figur 11 Første iterasjon av papirprototype	24
Figur 12 andre Iterasjon av papirprototype	25
Figur 13 Tredje Iterasjon av papirprototype	26
Figur 14 Arbeidsflyten i hi-fi prototypefasen	27
Figur 15 første hi-fi Iterasjon for mobil	28
Figur 16 Først hi-fi iterasjon for desktop	29
Figur 17 2.Iterasjon deskto	30
Figur 18 3.Iterasjon adminpanel desktop	31
Figur 19 Arbeidsflyt i kodefase	32
Figur 20 Eksempel på filoppsett med navngivningsregler	34
Figur 21 (venstre) Skjermdump av eksempel på bugfix på side	35
Figur 22 (høyre) Skjermdump av eksempel på bugpix i utviklervertøy	35
Figur 23 Skjermdump første iiterasjon	36
Figur 24 Skjermdump første iterasjon modal	36
Figur 25 Skjermdump første iterasjon Accordion	37
Figur 26 Skjermdump andre iterasjon	39

Figur 27 Skjermdump andre iterasjon	39
Figur 28 Skjermdump av kode for Df-Header.....	40
Figur 29 Skjermdump av media queries.....	41
Figur 30 Skjermdump av kode i Index.js for routing	42
Figur 31 Skjermdump av Cyberduck connection.....	45

1. Innledning

Denne bacheloroppgaven er delt opp i separat prosess og prosjektplan. Dette er grunnet da vi i alle tidligere fag har hatt separate prosjekt og prosessplaner og fordi vi mener at når det kommer til koding generelt er det lettere å dele inn. Vi har drøftet inndelingen av prosjektplan og prosessplan med veileder og fikk ett ja. Vi bemerker likevel at dette ikke er blitt gjort i tidligere bacheloroppgaver, men håper og tror at dette ikke vil påvirke karakteren i negativ retning.

I denne oppgaven har vi prøvd å være så prosedurale som mulig og la det tekniske og formelle språket ligge i prosjektrapporten. Det viktigste punktet i denne rapporten er å enkelt forklare arbeidsflyten i oppgaven og vise til prosessen, sekundært vil vi også gi en forklaring av gruppedynamikk, problemer i hver av fasene og refleksjoner fra de iterative fasene.

Målet med denne rapporten er å på en oversiktlig måte fremstille gruppens prosess fra planlegging til ferdig kode. Denne rapporten tar for seg kronologisk hva som skjedde og hvordan for å ha en rød tråd som gjør det lettere for leseren å forstå prosessen.

1.1 Overordnet prosess

Denne oppgaven er skrevet og gjort i Scrum, som er et agile rammeverk for prosjekthåndtering. Det innebærer iterative faser og å gå tilbake dynamisk og evaluere det som er gjort. Den overordnede prosessen i oppgaven er at vi først hentet inn informasjon, laget prototyper, testet disse og så kodet disse etterfulgt av testing av den ferdige koden. I realiteten var ikke prosessen så strømlinjeformet men snarere at informasjonsinnhenting førte til at vi laget prototyper, testing førte til at vi redesignet prototypene, testing førte til at vi kodet om og alle fasene gikk delvis inn i hverandre.

Prosjekt rammeverks endemål er å komme frem til et ønsket resultat, vi vurderte en rekke rammeverk, men kom til konklusjonen at Scrum var det beste for vårt prosjekt. Vi luket ut waterfall metode fordi den er veldig infleksibel og bedre egnet for prosjekter uten en konkret innleveringsdato. For vår del ga dette verdi med at vi kunne gå tilbake og endre i ting ettersom vi fikk bedre innsikt og fikk bekreftet eller avkreftet antakelser. Kanban, som også er et agile rammeverk, var for ikke optimalt da denne fokuserer mer på å optimalisere prosessen, hvor scrum fokuserer på å få resultater mer effektivt. I ettertid fikk vi også vite om «Adaptive Project Framework» noe som kanskje kunne vært scrums beste motstander for

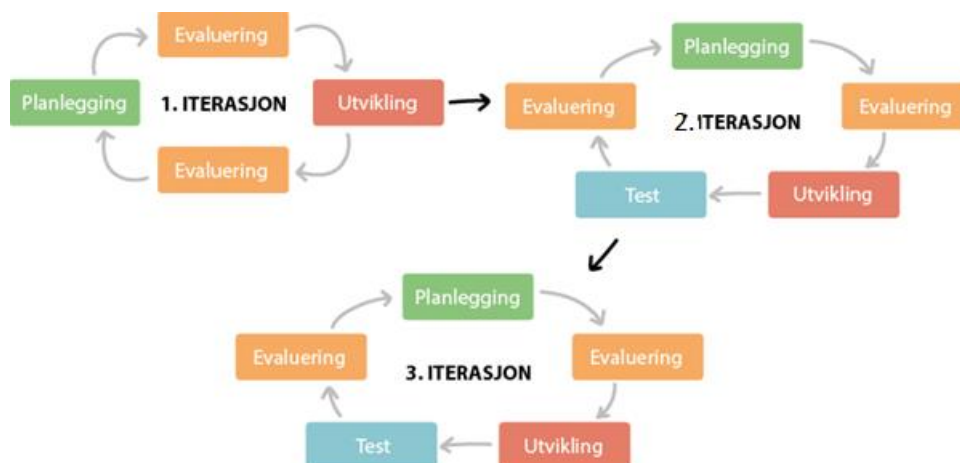
dette prosjektet. APF har brukt metodologier fra mange andre tidligere prosjektstyrings metoder og er også ett ypperlig rammeverk for prosjekter hvor du vet hva målet er. Scrum er et enda bedre rammeverk å bruke når du ikke vet hva målet egentlig er, noe vi hadde i vårt tilfelle. Selv om det også fungerte godt til vårt formål, ville vi sannsynligvis ha valgt APF ved et nytt prosjekt som har et klart mål.

For å gjøre dette enklest mulig for leser vil hvert kapittel begynne med et bilde som forklarer arbeidsflyten og iterasjoner i prosessen.



Figur 1 Forklaring av det generelle forløpet i prosjektet

Videre kan dette bildet analyseres og utvikles til det som vises i figuren nedenfor. Dette bildet nedenfor er en generell utgave av prosessen vi fulgte i for eksempel i prototyping eller koding. Vi valgte til slutt å inkludere en blanding av disse to figurene i hvert kapittel for å gi best mulig oversikt avhengig av hvilken det var. Fellesnevneren er likevel at disse viser arbeidsflyten i det gjeldende kapitelet



Figur 2 generell forklaring av de iterative fasene i prosjektet

I hvert enkelt av de store kapitlene er flyten at vi først går gjennom en innledning for å forklare kapitelet. Deretter tar generelt om ting i den fasen som vi følte var viktige å nevne.

Så går vi over til iterative faser og hvordan disse ble utført før vi avslutter med nevneverdige problemer i fasen.

2. Planleggingsfasen

Den første fasen vi fikk gjennom var planleggingsfasen. Denne fasen viser prosessen i kapitlet om forskning og design konseptualisering i prosjektrapporten, i tillegg til ytre faktorer. I denne fasen valgte vi grupperoller og mer rundt sammensetning av gruppen som Grupperegler, Kommunikasjon og møter, personlighetstesting, verktøy og prosjektrapport. I denne fasen begynte vi også første del av informasjonsinnhenting til oppgaven. Her hentet vi inn informasjon av konkurrenter, informasjon om hvordan andre har løst lignende oppgaver før og samlet pensum rundt temaene vi skulle skrive om.

2.1 Arbeidsflyt i planleggingsfasen



Figur 3 Arbeidsflyt i planleggingsfasen

1. Valg av grupperoller
2. Valg av verktøy
3. Møte med veileder
4. Prosjektplan
5. Møte med veileder og produkteier
6. Reevaluering av prosjektplan og informasjonsinnhenting

2.2 Sammensetting av gruppen

Vi samlet oss som gruppe allerede i bachelorgraden andre semester da vi samarbeider godt sammen og har tidligere erfaring fra forskjellige felt som gjorde at vi sammen utgjorde en god gruppe. Vi valgte oppgaven til Dokflyt da denne virket spennende og fordi dette var en av de få oppgavene hvor vi faktisk skapte et produkt fremfor kun skisser eller lignende. En annen grunn til at vi valgte oppgaven til Dokflyt er fordi dette er et firma vi kjenner til fra før

av som er kjent for å ha god oppfølging av studenter som skriver bacheloroppgaver for bedriften. På første gruppemøte diskuterte vi begrensninger til oppgaven og hvordan vi skulle sette opp en plan for videre gjennomføring, på dette gruppemøtet skulle vi også bestemme gruppens sammensetning og hvilke roller hver enkelt skulle ha. For å sette rollene i perspektiv blir det derfor i denne oppgaven kort referert til 'Belbins teamroller' i form av nøkkelord. På grunn av dette kommer roller også blir i stor grad utfylt av personlighets verktøyet som blir utdypet lenger nede.

2.3 Teamroller og personlighetstyper

For å optimalisere gruppedynamikken og grunnet at et gruppemedlem hadde personlighetstesting som hobby tok vi alle en personlighetstest fra 16personalities(1). Dette skulle vi analysere og bruke som supplement til valg av teamroller. Forståelse av dette personlighetsverktøyet fostrer bedre kommunikasjon og kan bedre sette i perspektiv hva som egentlig er problemet ved uenigheter.

Gruppedynamikk og gruppepersonligheter er noe som også tas opp i pensumboken «Interactive project management» (2) i kapitlet om «Emotional intelligence». Denne boken bruker også et form for personlighetssystem, men det er veldig simplifisert til prosjektstyring og gir et veldig begrenset bilde. For eks. er vår ENFP er sannsynligvis «The Wild Card» (s 31), men det at ENFP kan være litt uforutsigbar og har varierende arbeidsytelse (rykk og napp) kan også i stor grad tas ut ifra den andre modellen vi skal se på nå. Formålet til både deler av denne boken og 16personalities systemet er å komme overens med hverandre, kunne relatere i form av bedre forståelse og for å fostre ett støttende og tolerant miljø.

2.4 16 personalities og Kognitive funksjoner

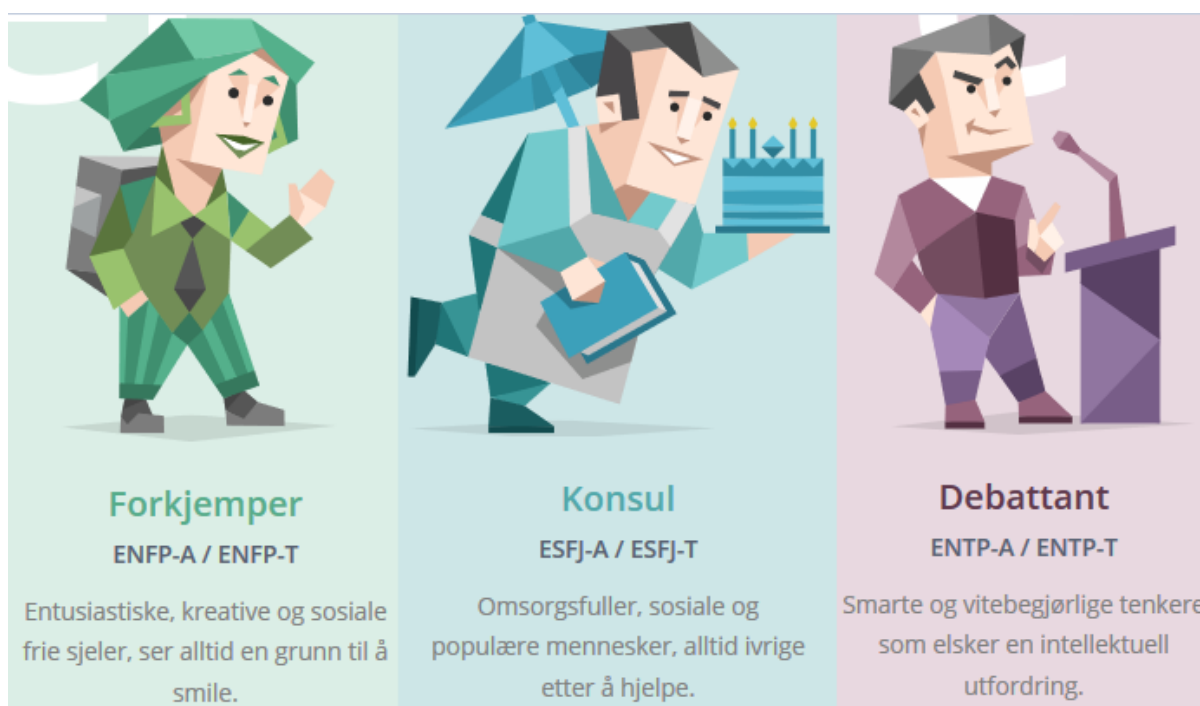
16personalities er et personlighets system som jobber med 4 dikotomier, som kan ansees som motsetninger, men brukes i det tilfelle som den mest fremtredende trekket blant de to motsetningene. Disse motsetningene er som følgende, oversatt fra engelsk til norsk (med bokstav forkortelse i parentes): Ekstrovert(E) eller Introvert(I), intuisjon(N) eller Sansning(S), føler(F) eller tenker(T), og avgjørelse(J) eller opplevelse(P).

Dette personlighetssystemet baserer seg på de kognitive funksjonene til Carl Jung, noe vi inkluderer for å få et mer helhetlig bilde av følelse og tenker dimensjonen.(3) Det er også relatert til, men ikke helt likt MBTI systemet som jeg ikke skal gå inn på forskjellene om,

men her er en oversiktlig referanse på hva de forskjellige funksjonene m.m vill si.(4) I dette så kommer også polaritetene mellom de relevante funksjonene til å bli nevnt(5).

2.4.1 Resultat

Personlighet(1)	Navn	Styrker	Svakheter	Kognitive Funksjoner(4)
Diplomat: Forkjemper: ENFP	Ruben Kvamme	Autentisk og åpent sinnet, Abstrakt, kreativ, Nysgjerrig, Observant, entusiastisk, vennlig, avslappet	Ikke praktisk, fokuserings vansker, overtenker, stressfølsom, emosjonell, misliker å bli pirket på, foretrekker det nye/muligheter fremfor det utprøvde og tidligere erfaring	Ne-Fi-Te-Si
Analytiker: Debatant: ENTP	Benjamin Kwiek	Abstrakt, kreativ, rask tenker, karismatisk, idémyldrer, energisk	Ufølsom, intolerant, diskusjonslysten, misliker praktiske saker / detaljer, foretrekker det nye/muligheter fremfor det utprøvde og tidligere erfaring	Ne-Ti-Fe-Si
Sentinel: Konsul: ESFJ	Nana Huynh Vo	Harmoni rettet i sosiale sirkler, Veldig praktisk, pliktoppfyllende, lojal, Sensitiv og varm, god til å komme overens med andre	Ufleksibel (ovenfor sosiale normer), kritikkfølsom, foretrekker det utprøvde og tidligere erfaring fremfor det nye	Fe-Si-Ne-Ti



FIGUR 4 SKJERMBILDER TATT AV TYPENE FRA 16PERSONALITIES.COM

2.4.2 Dimensjonene

Ekstrovert og introvert

Vi er alle på gruppen ekstroverte, noe som vanligvis er positivt i gruppeprosjekter om alle for kommet til ordet med tankene sine. Dette kan likevel bli en liten utfordring om en kommer til å dominere diskusjonene og de andre ikke kommer klart nok frem med synspunktene sine. For eksempel så kan potensielt ENTP i dette tilfelle fremstå som en dominant og «gjennomtenkt» person, noe som kan resultere til at de to andre, spesielt ESFJ, bli usikker på om hvorvidt det de har tenkt på er relevant nok til å nevne. ESFJs ekstroversjon er på dette området mye mer forsiktig med å fremme sin mening, da harmoni i sosiale relasjoner settes i høy prioritet. ENFP vill som oftest på dette område ytre seg om sin mening, noe som kan potensielt resultere i uenigheter.

Intuisjon og sansende

Benjamin og Ruben er intuitiv her, mens Nana er sanser. Vi er med to intuitive veldig tilpasset til idemyldring, abstraksjoner med god konseptualisering, og er derfor sterke på konseptualiserings område. Noe som kan være positivt for innovasjon, men suboptimalt om du kun ønsker å utvikle noe som kun etterligner tidligere løsninger. I vårt tilfelle var det litt begrenset hvor innovative vi kunne være, da en hjelpesentral med hovedsakelig FAQ allerede var klarert som oppgave resultatet.

Detaljer, fortiden og anerkjente standarder er veldig sansende evner vi (inkludert intuitive) har oppøvd oss gjennom livet da samfunnet er veldig sansende tilpasset. Detaljer blir med andre ord mye lettere oversett av en intuitive og det er derfor veldig gunstig å ha representanter fra begge sider i en prosjektgruppe. Noe som vi var så heldige å oppnå.

Følelse og tenking

For å utdype denne dimensjonen vill jeg referere til de kognitive funksjonene til Jung for å sette i perspektiv. Du har fire følelse og tenking funksjoner totalt, disse 4 funksjonene er de som definerer hvilke konklusjoner og avgjørelser du kommer til å ta ved subjektive problemstillinger.

Polariteter

ENFP har Introvert følelse, ESFJ har Ekstrovert følelse og ENTP har introvert tenking. I praksis så vill ENTP sin Introverte tenking være i polaritet til ESFJ sin Ekstroverte følelse, men dette vill ikke komme ut så synlig i prosessen da ESFJ vil ønske å gjøre sin jobb samtidig som harmoni opprettholdes. Mens ENTP vill legge frem sine forslag med overbevisning så vill ESFJ i de fleste tilfeller imøtekomme disse forslagene såfremt de gir

mening. Noe som derimot kan bli en reel utfordring og som ble i en kort periode ble, er om ENFP sin Introverte følelse går i opposisjon til ENTP sin introverte tenking.

Dette var noe som til dels skjedde da det skjedde en konflikt innad i gruppen rundt innlagt tid i arbeid imot resultatet av dette arbeidet. Da uenigheten omhandlet kompetansen til resultatet av arbeidet til Ruben og hva slags karakter dette arbeidet eventuelt ville blitt. Dette førte til en flammende diskusjon rundt hva som var innenfor autoriteten til en leder å si og hva som var forventet til levert arbeid. Dette ble tatt opp med veileder, diskutert og klargjort, noe som førte til forbedret samarbeid i ettertid.

Avgjørelse og oppfattelse

Vedrørende den siste dimensjonen om avgjørelse og oppfattelse, så er dette en indikator på om du ønsker å planlegge eller være mer spontant. Når vi nevnte innledningsvis ang. «Wild Card» så vill her «oppfattelse» i større grad være en indikator på dette. Noe som kommer på bakgrunn av at oppfattelses personligheter ikke setter så stor pris på å jobbe jevnt, men heller å jobbe i korte sprinter etter behov eller ønske. ENFP blir dette i større grad enn ENTP, ettersom FP kombinasjonen blir følelse spontant, mens TP blir mer analytisk spontan.

For å fylle ut disse er Benjamin trolig «The Hero» da denne vill «spontan analytisk» finne problemer hvor den ønsker å løse den og derfor tre inn for å løse den. Mens Nana vill gå som «The Gem» da ESFJ blir ansett som pålitelig, stabil og samarbeidsvillig. Stabiliteten kommer på grunnlag av «avgjørelsen» som vil ønske å planlegge og deretter følge planen som er satt. (24, 31 og 47 4).

2.5 Konklusjon

Oppsummert på det realistiske nivået så blir dette reflektert i at Benjamin var en veldig god leder for å analysere problemer og dele ut oppgaver raskt og effektivt. Han var også objektivt den mest drevne i gruppa på avansert front-end problematikk, noe som er veldig analytisk arbeid. Nana som var den mest detaljerte tenkeren ble den beste kandidaten for front-end og design detaljer. I tillegg til dette ble evnen hennes til å se når de to andre har skrevet litt for abstrakt ganske positivt for å kunne levere en bedre formulert rapport. Ruben ble derimot hjelpende til med litt av hvert innen informasjonsinnhenting, rapportskriving og testing av løsningen.

2.6 Verktøy

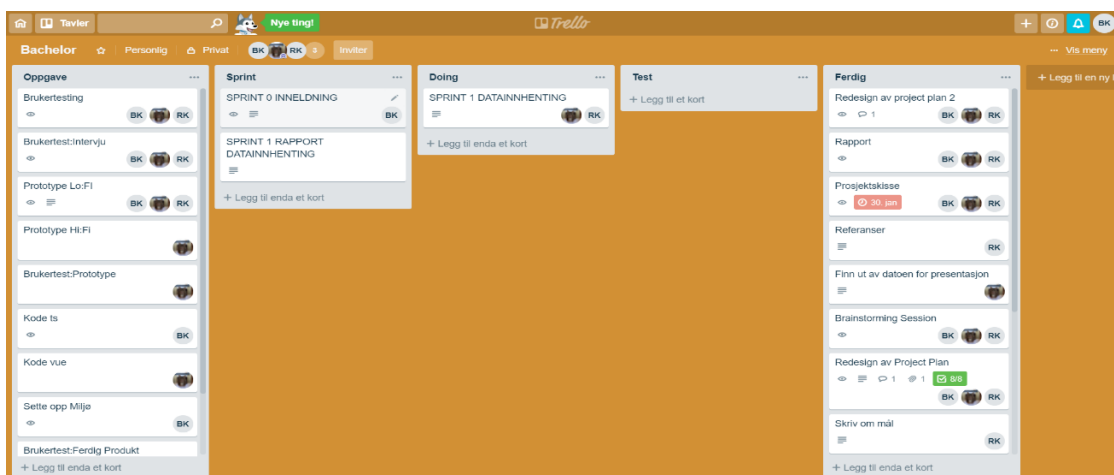
I prosjektplan fasen undersøkte vi potensielle verktøy vi kunne brukte til oppgaven. Vi tok først for oss et verktøy for Gantt Diagram og endte opp med TeamGantt da dette var gratis og enkelt å bruke. Utenom dette hadde vi flere større verktøy vi brukte til forskjellige oppgaver som forklares videre.

2.6.1 Mendeley

For referanser så anbefalte veilederen vår oss å vurdere mellom mendeley eller endnote. Mendeley er enkel å bruke og tillater deg å sette opp gruppe nettverk mappe i skyen for samarbeid om referansene. Endnote hadde ikke dette og var i følge omtale litt mer avansert å lære seg enn mendeley, noe som var hovedgrunnen til at vi gikk for Mendeley.

2.6.2 Trello

Trello er et samhandlingsverktøy som gjorde det enkelt å dele ut oppgaver. I skjermbildet nedenfor ser vi en tidlig “tavle” for prosjektet hvor gruppeleder lager oppgaver og gir disse ut til medlemmene. Grunnen for at vi valgte trello er at det var gratis og enkelt i bruk, samt at alle gruppemedlemmene hadde tidligere erfaring med trello. Det er også ett veldig kompatibelt verktøy med Scrum, da scrum avhenger av lister med oppgaver til de forskjellige sprintene. Trello tavlen er veldig fleksibel og redigerbar, noe som er konseptuelt nesten som skapt for scrum.

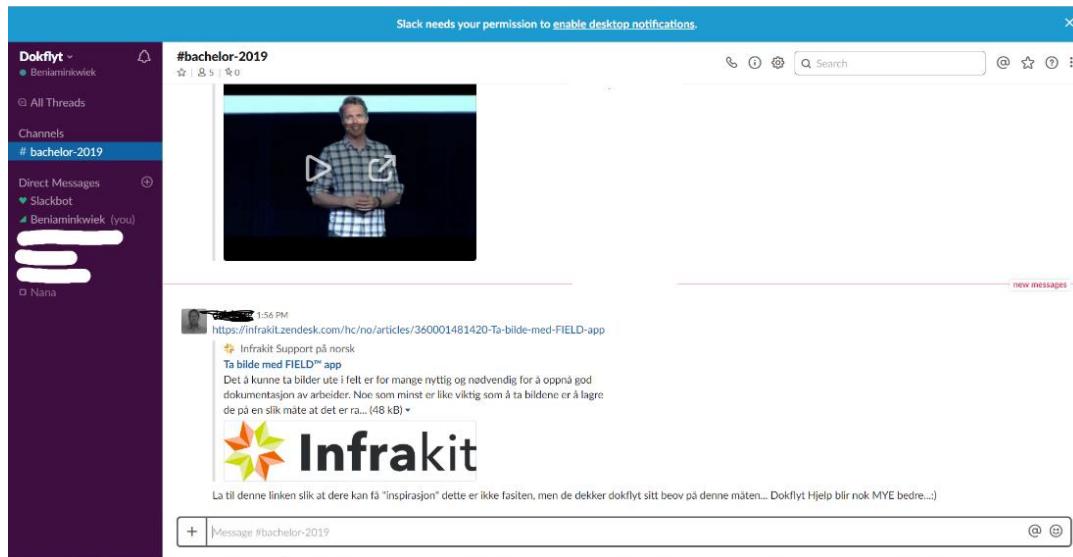


Figur 4 Skjermdump av Trello-Tavle for prosjekt

2.6.3 Slack

Et annet verktøy vi brukte for kommunikasjon var Slack. Slack er et samhandlingsmedium som på mange måter ligner på Facebook Messenger. Det er mulig å skrive til hverandre,

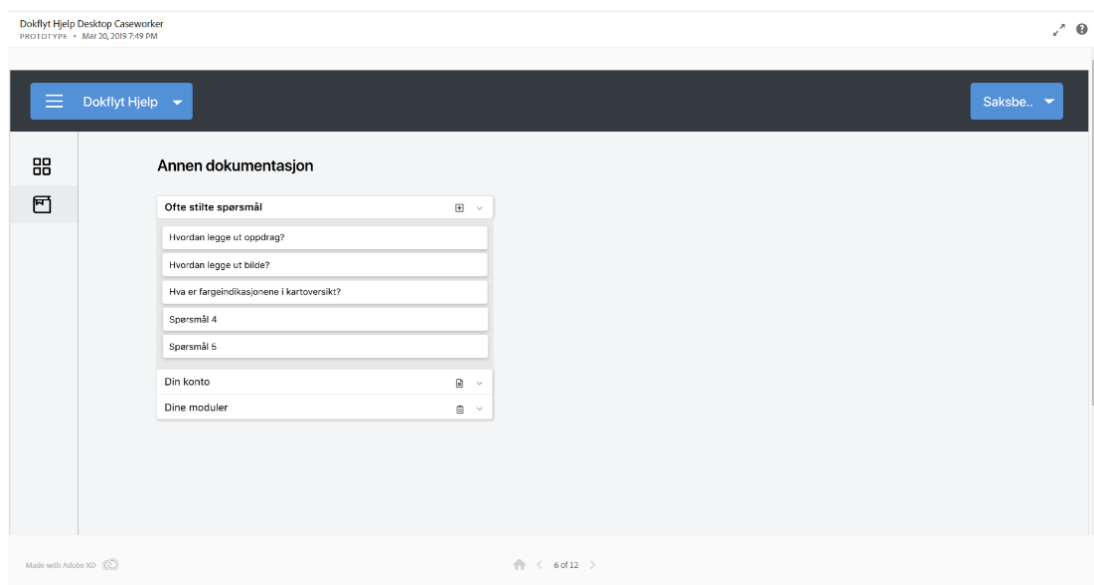
sende bilder og dokumenter samt mulighet for VoIP. Grunnen til at vi valgte Slack over for eksempel Facebook Messenger var fordi det er lett å bli ufokusert og la tankene gå til andre ting når en er på Facebook. En annen grunn er fordi Slack er kostnadsfritt og har en enkel UI å lære seg. To av gruppemedlemmene hadde tidligere brukt Slack, mens siste gruppemedlem brukte rundt 30min på å lære seg dette.



Figur 5 Skjermdump av Slack gruppen i prosjektet

2.6.4 Adobe XD

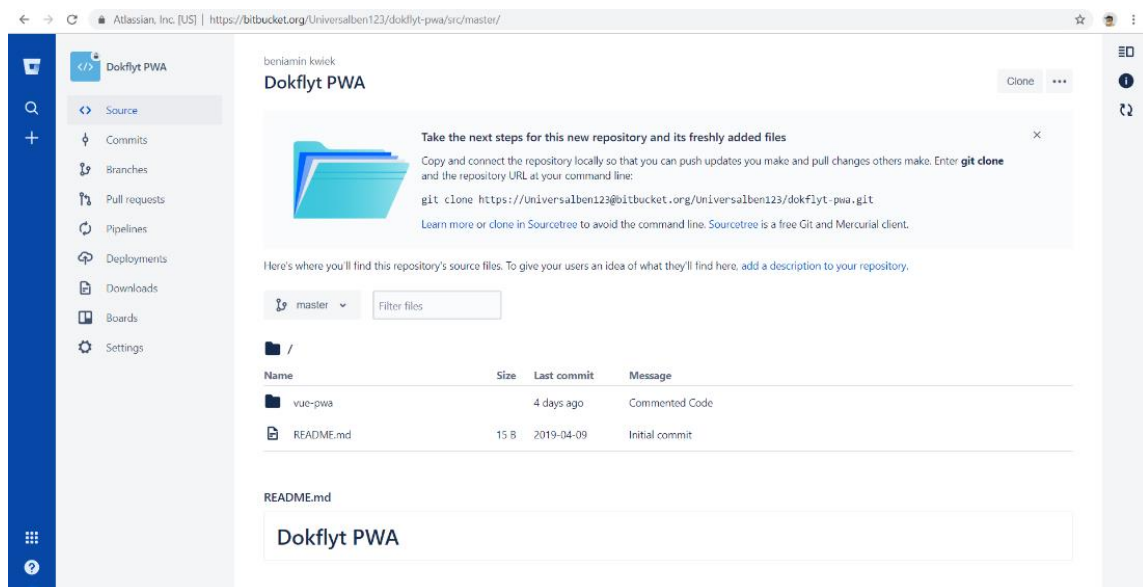
For prototyping valgte vi å bruke Adobe XD. Adobe XD er et program vi har brukt i flere fag som har den funksjonaliteten vi ønsket i en klikkbar Hi-Fi prototype. Grunnen til at Adobe XD ble valgt over konkurrentene er fordi vi hadde god erfaring med dette programmet, vi hadde tilgang til dette programmet og fordi det ga oss god kontroll over prototypingen.



Figur 6 Skjermdump av prototype i Adobe XD

2.6.5 Bitbucket

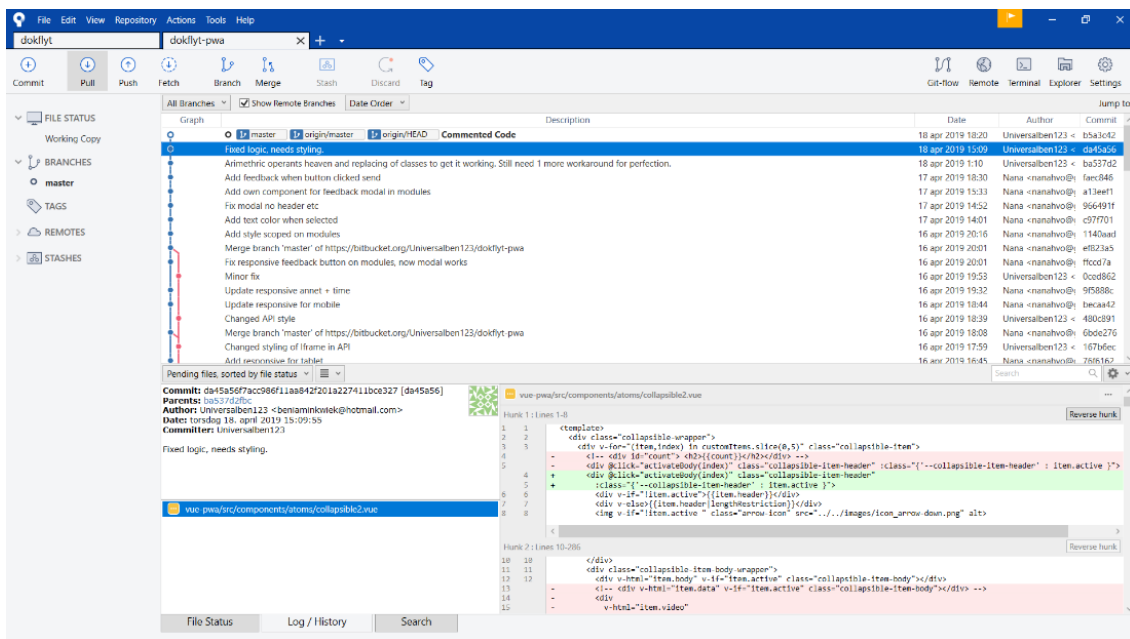
For versjonering valgte vi å bruke Bitbucket. Bitbucket er web-basert versjoneringshosting som GitHub men gir oss mulighet for å gratis ha private repositories. Dette var spesielt viktig med tanke på at koden som ble produsert skulle brukes videre av Dokflyt. Det ble fort klart at vi trengte et versjoneringsystem da to gruppemedlemmer kodet hele applikasjonen og derfor måtte ha en måte å ikke overskrive hverandres filer på. Siden disse to gruppemedlemmene hadde erfaring med Git valgte vi å gå for dette.



Figur 7 Skjermdump av repository i Bitbucket

2.6.6 Sourcetree

For at det skulle enda enklere å unngå feil valgte vi å bruke SourceTree som Git Graphic User Interface. Dette gjorde det enkelt å se hva som var endret og å forplikte(committ) samt trekke(pull). SourceTree ga oss en visuell representasjon det er vanskelig å få i ledetekst eller terminal.



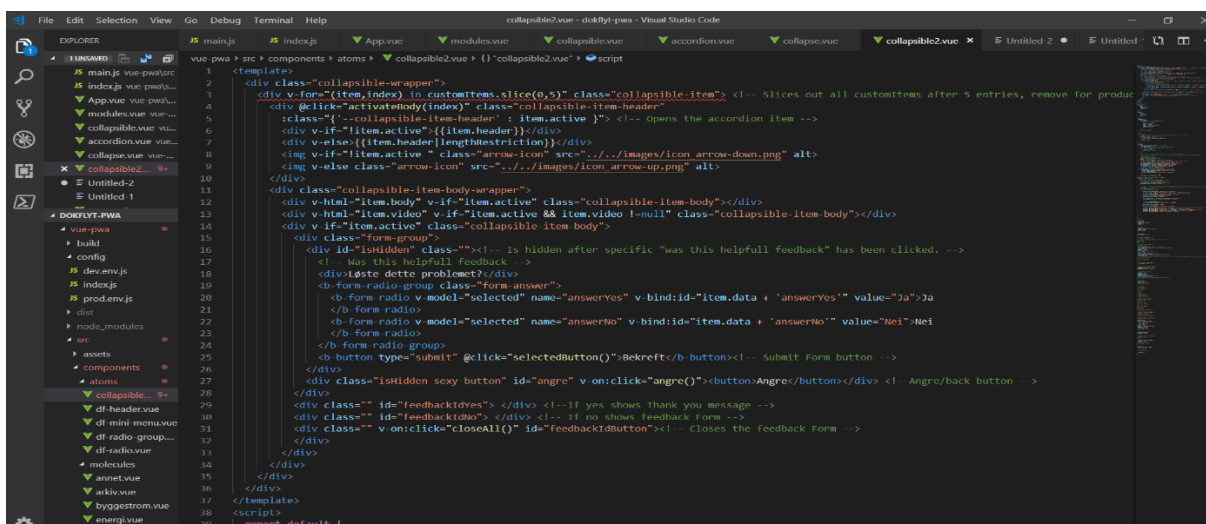
Figur 8 Skjermdump av GUI i Sourcetree

2.6.7 NPM

For package management for dependencies så valgte vi NPM. Grunnen til dette er at vi tidligere hadde erfaring med NPM og at dette er verdens største software registry, package manager og installer i ett. Ved å ha en package.json fil kunne vi også om ønskelig publisere vår oppgave. To gruppemedlemmer hadde tidligere erfaring med NPM og disse var også de som kodet oppgaven, derfor ble avgjørelsen ganske enkel. Vi valgte også NPM fordi dette var default valget som fulgte med «vue CLI» sine dokumentasjoner. Mange utviklere har begynt å se på Yarn (en konkurrent) som best practice, men ettersom NPM var vue default og yarn ikke hadde noen dokumentasjoner, mente vi det var unødvendig å endre til denne.

2.6.8 Visual Studio Code

For selve kode redigering valgte vi Visual Studio Code, som herfra kommer for enkelthet til å bli omtalt som VSC. VSC er en open source code-editor som har flere verktøy å bruke til for eksempel debugging og enkel tilgang til NPM ved innebygget cmd eller terminal for mac brukere. VSC har også innebygd Git kontroll system som enkelt gjorde at vi kunne se hvilke filer som var blitt endret og hvordan. Grunnen til at ikke valgte en code-editor som Atom eller Brackets er fordi VSC ga oss flere muligheter og verktøy til debugging som viste seg å være veldig greit iforhold til oppgaven.



Figur 9 Skjermdump av kode i Visual Studio Code

2.7 Grupperegler

Siden vi kjente hverandre følte vi ikke behovet for å ha veldig mange regler. Men vi valgte likevel å inkludere tre regler.

1. Alle skal møte opp til alle møter med mindre sykdom eller andre uforutsette hendelser hindrer dette.
2. Hvis et gruppe medlem ikke møter opp på 3 møter skal de resterende gruppe medlemmene melde dette til veileder.
3. Det skal føres logg for hvert møte, større arbeid og veiledning.

Vi valgte å inkludere disse reglene for å ha sanksjoner hvis et gruppe medlem meldte seg ut eller ikke gjorde arbeid. Den siste regelen ble laget for å alltid kunne gå tilbake å lese alt som var blitt gjort.

Grupperegul 1 var enkelt satt for å sørge for at alle bidrog i møter og at vi slapp merarbeid ved å forklare alt to ganger.

Grupperegul 2 var en sanksjon slik at vi hadde et pressmiddel hvis et medlem skulle melde seg ut eller unnlate å gjøre arbeid.

Grupperegul 3 var satt for at vi alltid kunne gå tilbake å se hva som ble gjort når. Denne loggen blir ikke lagt til som vedlegg da den tidvis inneholder personlige emner og tema.

2.8 Første møte med veileder

I dette møtet gikk vi gjennom verktøyene vi hadde valgt samt rollene vi hadde kommet frem til. Dette var for å bekrefte antakelser rundt gruppe og verktøy samt for å høre om det var noen andre verktøy vi ikke hadde vurdert som kunne vært bedre. I dette møtet fikk vi også en gjennomgang av prosjektplan og hva denne skulle inneholde.

2.9 Prosjektplan

I denne fasen laget vi prosjektplanen som skulle fungere som kart og kompass for prosjektet. Vi gikk gjennom flere faser med veileder hvor vi revaluerte prosjektplanen, tok vekk punkter og la til punkter. Til slutt fikk vi en prosjektplan som fungerte og som vi brukte for en overordnet plan for resten av prosjektet.

2.10 Kommunikasjon og møter

Vi hadde et gruppemøte hver mandag hvor vi gikk gjennom hva vi hadde gjort og hva vi måtte gå gjennom denne uken. Vi hadde også gruppemøte hver onsdag og hver torsdag før og etter veiledning. Utenom dette hadde vi et morgenmøte daglig for å oppdatere hverandre om hva vi hadde gjort dagen før og skulle gjøre den dagen. Hvis vi av forskjellige grunner ikke kunne møte opp holdt vi møte gjennom Discord. For formell kommunikasjon brukte vi Slack og oppgavegiving Trello. Hvis vi skulle gi en uformell beskjed ble Facebook Messenger brukt ved nød. Utenom møtene jobbet vi individuelt med oppgavene vi hadde fått utdelt av gruppeleder.

2.11 Informasjonsuthenting

Vi begynte allerede i planleggingsfasen å finne informasjon til hvordan vi skulle utforme applikasjonen og hvilke funksjoner vi ønsket å ha med. Informasjonsinnhenting foregikk ved at vi først fant ut hvilke konkurrenter Dokflyt hadde og hvilke løsninger allerede fantes for en hjelpesentral. Vi fant også ut hva en hjelpesentral var og hva som utgjorde en god hjelpesentral. I denne delen holdt vi et møte med Dokflyt og kom med deres kundeserviceansvarlig opp med antakelser vi kunne bruke for utforming av vår oppgave.

2.12 Eksterne Ressurser

Vi hadde hver Torsdag 1400 et gruppemøte med veileder for å få tilbakemelding på forrige ukes arbeid og for å få veiledning for hva som måtte gjøres følgende uke. Veileder ga oss mye nyttig kunnskap og gjorde at kunne ha en fremdrift som var tilfredsstillende i forhold til våre krav. I tillegg holdt vi møter med produkteier for å få klarhet i hvordan oppgaven kunne gi mest mulig verdi for Dokflyt. Produkteier ga oss informasjon om hvordan han ville at applikasjon skulle utformes og ga oss dypere innsikt i hvilke problemer kunder ofte hadde. Produkteier fungerer til daglig også som kundeserviceansvarlig for Dokflyt og kunne derfor gi god innsikt i hva som trengtes og hvilke antakelser vi kunne begynne med før vi beviste disse gjennom datainnhenting og testing.

Et gruppemedlem fikk også praksisplass hos Dokflyt, vi valgte derfor å inkludere dette i avsnittet om eksterne ressurser. Gruppemedlemmet fikk innsikt i bedriften Dokflyt, kodespråket Vue og ble et verdifullt mellomledd mellom produkteier og oss som gruppe. Dette bedret i stor grad kommunikasjonen med produkteier og effektiviteten i å få svar på eventuelle spørsmål vi måtte ha.

Vi hadde som tidligere nevnt veiledning med veileder hver uke, noe som var veldig konstruktiv for utformingen av oppgaven. Veileder ga oss feedback på materialet vi hadde produsert og belyste ting vi kanskje hadde oversett. Utenom dette var produkteier også til hjelp for å oppklare mye av spørsmål vi hadde rundt Dokflyt som bedrift og ga oss innsikt i eksisterende kundeservice. Produkteier var også kundeserviceansvarlig i Dokflyt per i dag og kunne derfor gi oss klare innsikter og antakelser vi kunne gå videre med før vi testet disse. Han gav oss mye nyttig feedback, som blant annet gikk på formelt språk og struktur i tillegg til mange generelle forslag vi burde vurdere.

2.13 Rapportskriving og loggskriving

Underveis gjennom hele oppgaven skrev vi rapport og logg. Dette var for å ha det ferskt i minne og fordi det ikke nytter å skrive et dokument på over hundre sider i skipptak. Rapportskriving ble foretatt av gruppemedlemmene for oppgavene de selv gjorde, men ble i stor grad rettet etter hvert som vi fikk tilbakemelding fra veileder av et annet gruppemedlem. Hvis et gruppemedlem behøvde hjelp som hadde med rapportskriving sendte dette en melding på Slack og så kommuniserte vi videre derfra. Rapportskrivingen gikk for det meste uten problemer, det eneste som var et problem var at norskkunnskapene innad i gruppen

varierte veldig. For å gjøre opp for dette ble oppgavene fordelt slik at den som skrev best også skrev mest.

2.14 Møte med produkteier

I dette møtet presenterte vi prosjektplanen, hvilken informasjon vi hadde innhentet samt innhentet ekstra informasjon som vi brukte som antakelser for den første papirprototypen. Her fikk vi verdifull innsikt i hva vi kunne anta til å begynne med og litt innsikt i tenkemåten til Dokflyts kunder.

2.15 Problemer i planleggingsfasen

I denne fasen støtte vi på et større problem som omhandlet prosjektplanen. Det var ingen på gruppen som hadde god nok forståelse av Gantt diagram for å få dette til å være godt nok noe som trakk ned kvaliteten på prosjektrapporten. Gruppemedlemmet som fikk i oppgave å endre Gantt diagrammet gjorde ikke nok informasjonsinnhenting rundt dette og dette ble ikke noe bedre etter re-evaluering. Vi bestemte oss derfor for å re-evaluere Gantt diagrammet sammen som gruppe og dette var tidskrevende i forhold til hva som burde vært nødvendig.

3.Design

Designkapitlet i denne rapporten tilsvare prosessen og design valgene gjort på slutten av forskningen rundt design konsept, i tillegg til lo-fi og hi-fi kapitlene. Disse er basert på tidligere forskning av brukerintervju og spørreundersøkelse, i tillegg til analysen av andre hjelpesentraler.

3.1 SWOT-Analyse

Før vi begynte konseptualiseringen av designet utførte vi en SWOT analyse på Dokfly og to konkurrerende løsninger. Dokflyt utførte vi for å kunne sammenligne og for å bedre kunne kartlegge styrker mot svakheter og muligheter mot trusler. Ved sammenligning med de to andre konturerende løsningene finner vi ut at det er rimelig viktig for Dokflyt å få implementert en hjelpesentral. Utenom vår løsning konkluderte vi også med at Dokflyt burde

øke sin innsikt og markedsføring tilstrekkelig. Da dette kan hjelpe dem å ekspandere seg som et selskap å oppnå nasjonal innflytelse.

3.2 Personas

For å få til en best mulig oppgave bestemte vi oss for å lage Personas som skulle representere gjennomsnittet av dokflyts brukere. Fordi disse var delt inn i to laget vi også to personas. En til montør og en til saksbehandler. Disse var forskjellige men samtidig like og dette er representert gjennom personas. Vi laget personas ved at vi samlet inn innsikt om brukere ved observasjoner og samtaler med produkteier og kom frem til de endelige personasene etter flere iterasjoner

3.3 Scenarios

Mens vi laget personas laget vi også scenarios for å ha noen situasjoner å sette oss inn i. Disse ble laget ved at vi tidlig diskuterte hvilke situasjoner brukerne kunne havne i ved bruk av applikasjonen og hvordan vi måtte designe applikasjonen. Scenarioene ble til etter samtaler med produkteier og etter innledende intervju hvor vi samlet inn innsikt på hvordan brukere har det i dag og hvordan de eventuelt ville hatt det ved bruk av en slik applikasjon.

3.4 Storyboard

Storyboardet ble utviklet for å illustrere hvordan brukere kommer til å bruke applikasjonen med miljøet rundt i tankene. Storyboardet var i stor grad for å vise bruken av applikasjonen for sensor og for å for vår del illustrere hvordan det skulle fungere med miljøet rundt bruker.

3.5 Lo-Fidelity prototype

For lo-fidelity prototype valgte vi og papirprototype. Dette var fordi det var enklere å endre både design og funksjoner fort og enkelt. Disse er også skissert med baktanke på Nielsens heuristikker, men ble ikke analysert i detalj før hi-fi prototypen. Det var mer enn 3 iterasjoner av papirprototypen men fordi disse hadde så små endringer valgte vi kun å inkludere de 3 iterasjonene som hadde størst betydning for videre utvikling av prosjektet.

3.6 Arbeidsflyt i Lo-Fi prototypefasen



Figur 10 arbeidsflyt i lo-fi prototypingfasen

Arbeidsflyten i å lage en papirprototype som kan ses av bilde over gikk som følger:

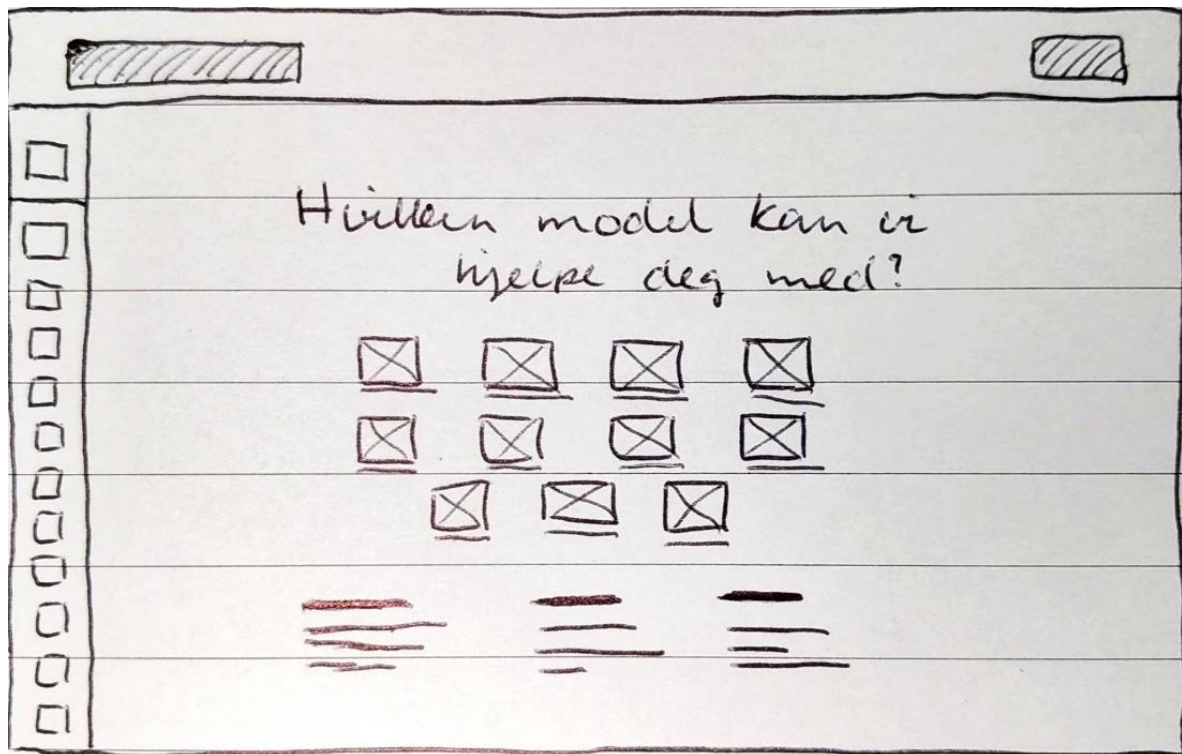
1. Vi tok med oss innsikten fra planleggingsfasen og laget 1.Iterasjon
2. Etter samtaler med Produkteier og Veileder utviklet vi 2.Iterasjon
3. Test av 2.iterasjon
4. Etter test av 2.Iterasjon utviklet vi 3.Iterasjon

3.7 Idemyldring papirprototype

Ettersom dette kunne være en hjelpesentral var vi innom mange ideer før vi landet på disse designmessige valgene. Vi vurdere blant annet en Chatbot, denne ble forkastet da det ikke var ønskelig fra produkteier samt at de store selskapene som Google og Apple ikke brukte dette. En annen ide vi var innom var å ha en søkefunksjon, men etter samtaler med produkteier viste det seg at det mest sannsynlig ikke kom til å være mange nok spørsmål i hver modul til at det svarte seg.

3.8 Første iterasjon Lo-Fi prototype

Vi hentet inn informasjon og fikk antagelser vi kunne bygge på av produkteier. Disse la grunnlag for den første papirprototypen vår. Papirprototypen ble i første omgang diskutert i ett møte hvor vi la frem konkurrerende løsninger, analyser av innhenting og bygget et grunnlag av rimelige antakelser og ideer. Etter dette møtet lagde gruppe medlemmene individuelt skisser for papirprototyper, disse ble ved neste møte diskutert og vi laget en felles papirprototype. Denne ble deretter videreutviklet individuelt og vi laget en ny papirprototype ved neste møte basert på funn og diskusjon. Slik fortsatte vi til vi hadde en papirprototype vi var fornøyd med nok til å kunne ta med til veileder for å se på denne og komme med sin tilbakemelding.



Figur 11 Første iterasjon av papirprototype

3.9 Møte med veileder og produkteier

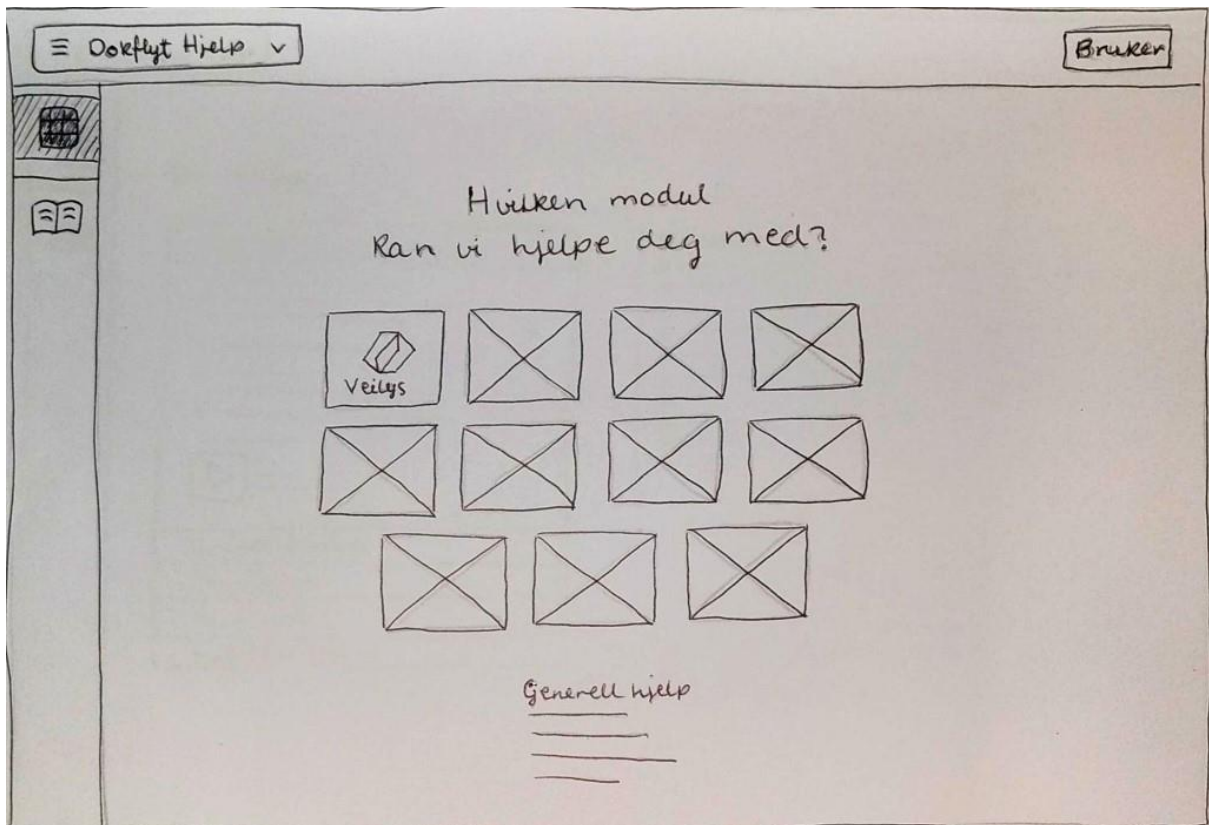
Etter å ha fått tilbakemelding fra veileder og Dokflyt var disse stort sett fornøyde. Siden en Low-fi prototype er vanskeligere å forestille seg som et ferdig produkt iforhold til en Hi-fi prototype valgte vi å fortsette med det generelle designet av denne prototypen men fortsette til neste iterasjon av prototypen med litt flere detaljer. Denne prototype skulle brukertestes på brukere av dokflyt og måtte derfor inneholde flere detaljer og flere skjermer.

3.10 Reevaluering av første iterasjon Lo-Fi prototype

Resultat av reevaluering av første iterasjon Lo-Fi prototype

- Mer detaljer
- Flere "skjermer"

3.11 Andre iterasjon Lo-Fi prototype



Figur 12 andre iterasjon av papirprototype

3.12 Brukertest andre iterasjon Lo-Fi prototype

Etter at vi hadde utviklet en papirprototype fra innsiktene vi fikk fra de tidligere undersøkelsene og produkteier testet vi denne. Fra produkteier fikk vi i denne fasen kun 4 personer å teste på. Vi mente dette var for lite for å få et representativt resultat. Vi gikk likevel videre med denne brukertesten etter diskusjoner innad hvor vi kom frem til at testen av Hi-fi prototypen måtte ha et større utvalg. Brukertesten ble foretatt på to saksbehandlere og to montører og bestod av både oppgaver og generelle oppklarende spørsmål. Brukertesten foregikk i intervjuobjektets naturlige miljø og intervjugruppen bestod av en intervjuleder, en assistent som flyttet på "bildene" ettersom intervjuobjekt trykket på linker samt en sekretær som tok notater. Innsiktene fra disse intervjuene ble deretter samlet og utviklet videre til utformingen av hi-fidelity prototypen.

3.13 Reevaluering etter brukertest

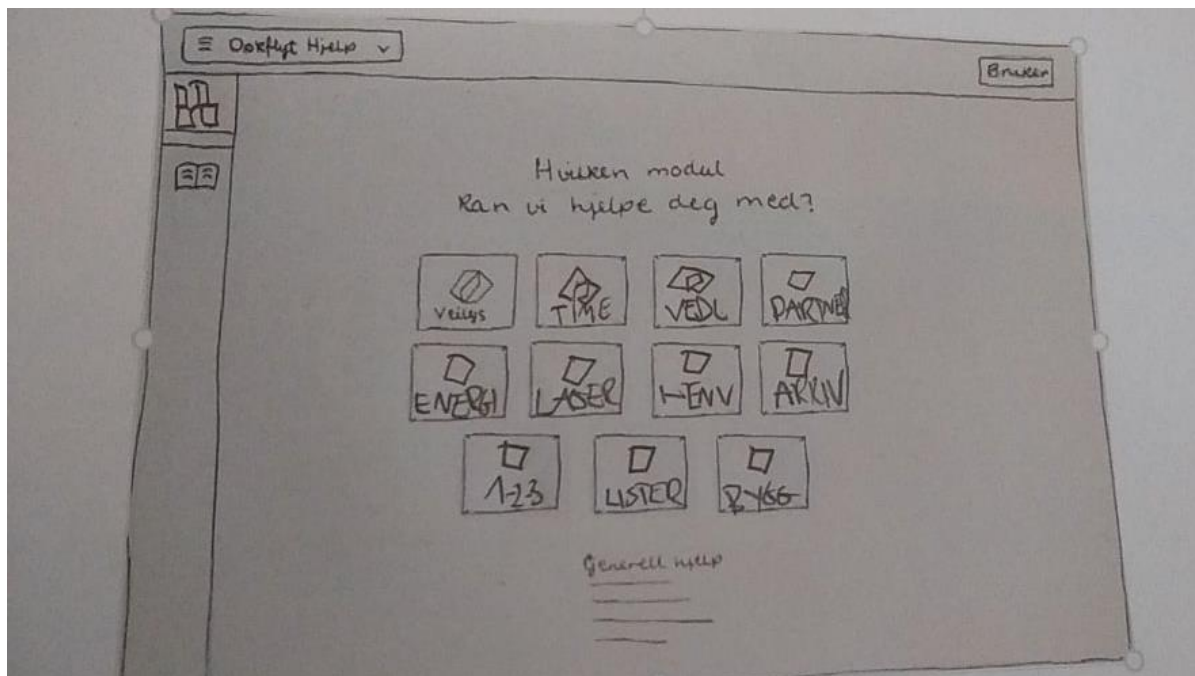
Innsiktene fra brukertesten ga oss mulighet for å evaluere papirprototypen en siste gang før utviklingen til hi-fi prototypen. Denne innsikten var blant annet at testpersonene var fornøyde med designet så vi fikk videre med dette. Administrator var fornøyd med både

design og utførelse, derfor gikk vi videre med både design og funksjoner i administratorpanelet. Vi valgte også å gjøre små endringer i design men stort sett ble 3. iterasjon en mer detaljert versjon av 2. iterasjon.

Resultat av reevaluering av andre iterasjon Lo-Fi prototype

- Mer detaljer
- Små designmessige endringer

3.14 Tredje Iterasjon



Figur 13 Tredje Iterasjon av papirprototype

3.15 Problemer i Lo-Fidelity prototypefasen

Det var ganske få problemer i Lo-fidelity fasen som vi kunne gjøre noe med. Vi hadde nevneverdig store problemer med å finne faktiske brukere fra Dokflyt til å teste for oss. Noe som også viser seg i fremtidige brukertesting, hvor vi ikke tester faktiske brukere men heller «potensielle» brukere. I tillegg til dette var det også flere omganger med sykdom innad i gruppen som måtte bli tatt hensyn til.

4.Hi-Fidelity prototype

Vi bestemte oss tidlig i prosessen for å lage en klikkbar Hi-Fidelity prototype i Adobe XD, dette var fordi vi hadde tidligere erfaring med programmet. Prototypen ble utviklet med tanke på «Nielsens heuristikker» og tok en del inspirasjon fra tidligere analyserte hjelpesentraller. Den inneholder også noen antakelser i tillegg til funn i papirprototypen som er videreført til denne prototypen arbeides med for ved brukertesting og reevaluering.

4.1 Arbeidsflyt i Hi-Fidelity prototypefasen



Figur 14 Arbeidsflyten i hi-fi prototypefasen

Arbeidsflyten i å lage en Hi-fidelity prototype kan ses av bilde over og gikk som følger:

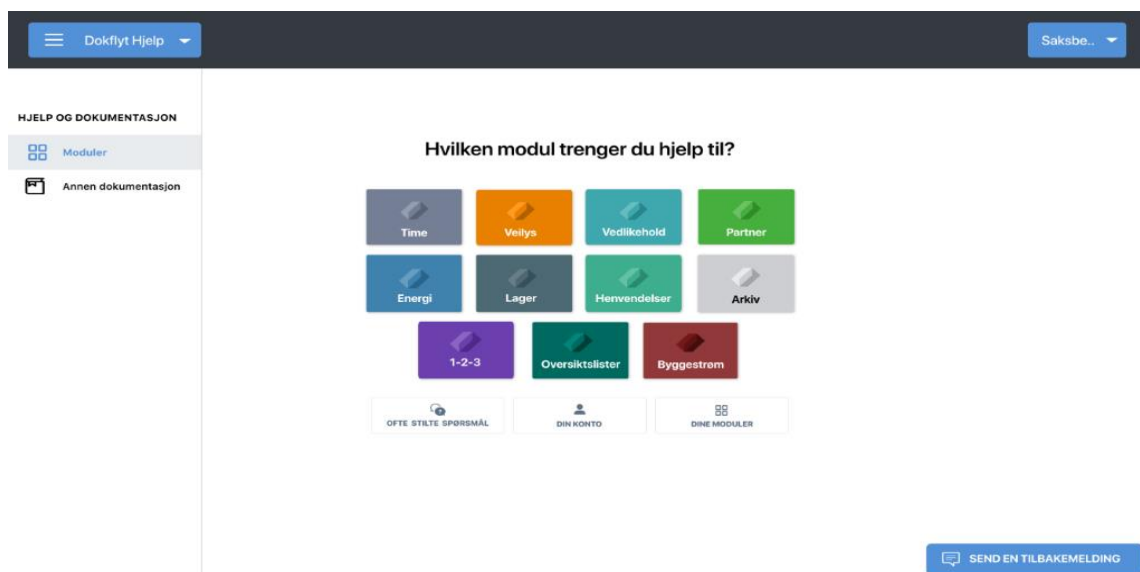
1. Først tok vi med oss innsikt fra papirprototypen og laget 1.iterasjon.
2. Brukertest av 1.iterasjon
3. Utvikling av 2.iterasjon med innsikter fra test
4. Samtale med produkteier og Dokflyt
5. 3.iterasjon etter samtaler med produkt og dokflyt.

4.2 Første iterasjon Hi-fi Prototype

Vi gikk videre med antakelsene vi hadde fått bekreftet eller avkreftet og laget en klikkbar hi-fi prototype i Adobe XD, denne inneholdt en mobilversjon og en desktop versjon. Etter samtaler med produkteier ville ikke de vi skulle lage en adminversjon og vi valgte derfor å ikke fortsette med denne i den ferdige koden. Mobilversjonen og Desktopversjonen ble brukertestet og redesignet i flere faser. Funksjoner ble lagt til og fjernet etter vurderinger fra brukertest samt innspill fra gruppe, produkteier og veileder.



Figur 15 første hi-fi iterasjon for mobil



FIGUR 16 FØRST HI-FI ITERASJON FOR DESKTOP

4.3 Brukertest første iterasjon Hi-fi prototype

Brukertesten i Hifi prototypen ble foretatt samlet men allikevel i to deler. Vi laget prototype både for mobilversjon og desktop versjon og testet disse. Til sammen hadde vi rundt 25 responedenter for å få et representativt resultat som kunne brukes. Vi holdt intervjuene i brukeren naturlige miljø og reiste ut for å intervjuer montører for mobilversjonen i felt mens saksbehandlere ofte jobbet på kontor. I denne brukertesten hadde vi en person som holdt intervjuet mens en annen var sekretær. Intervjuet ble gjort på en enhet vi hadde med oss for å gjøre det mest mulig realistisk.

4.4 Reevaluering etter brukertest

I brukertesten kom det frem at folk stort sett var fornøyde med designet for desktop så det lot vi stå til dels urørt, designet for mobil var i noen tilfeller uklart for eksempel med send en tilbakemelding knappen. Det kom frem at noen følte at for eksempel din konto og dine moduler var overflødige funksjoner.

Resultat av reevaluering av første iterasjon Hi-Fi prototype

- Fjernet “Din konto”
- Fjernet “Dine Moduler”

4.5 Andre iterasjon Hi-Fi Prototype



Figur 17 2. Iterasjon desкто

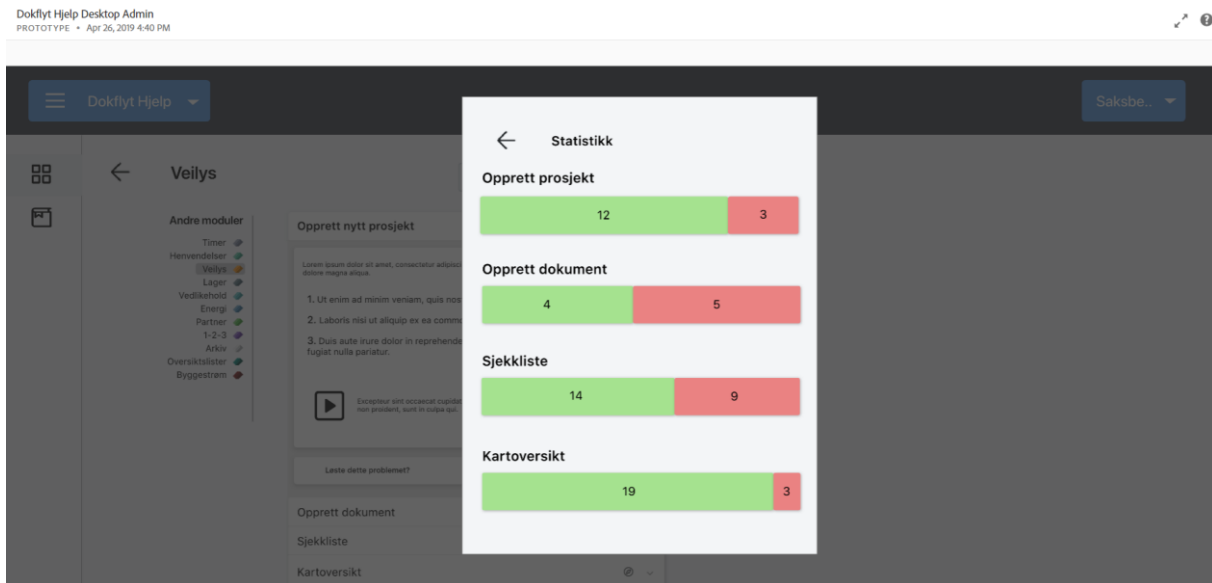
4.6 Reevaluering etter møte med veileder og produkteier

I dette møtet med veileder kom vi frem til at det ville vært smart å inkludere en administrasjonsside for statistikk og å legge til spørsmål i applikasjonen. Deretter møtte vi med produkteier som selv ville se en slik adminside i prototypen men at vi ikke skulle utvikle denne videre i koden. Dette ble begrunnet med at de Dokflyt hadde et eget admin komponentet og grunnet sikkerhet rundt dette.

Resultat av reevaluering av andre iterasjon Hi-Fi prototype

- Utviklet Administrasjonspanel

4.7 Tredje iterasjon Hi-Fi prototype



Figur 18 3. Iterasjon adminpanel desktop

4.8 Problemer i Hi-fi prototype fasen

Vi hadde få problemer i informasjonsinnhentingsfasen, de problemene vi hadde løste vi enkelt. Et problem som er verdt å nevne er at vi ikke fikk nok intervjuobjekter å teste på av produkteier for å få et representativt resultat. Dette løste vi ved å intervjuere brukere vi selv fant basert på parameter vi selv satte. Disse prøvde vi å være mest mulig like brukerne av Dokflyt sine applikasjoner. Et annet problem i denne fasen var sykdom, da vi gikk gjennom en runde med influensa som gjorde at vi måtte fordele arbeidet på en litt annen måte enn i de resterende fasene. Dette løste vi ved at de friske gruppemedlemmene gjorde ekstra mye disse ukene.

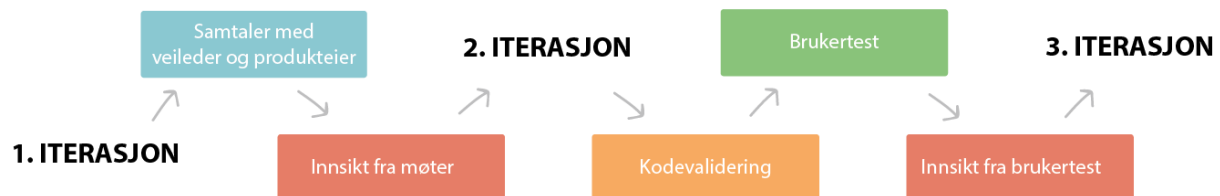
5. Koding

Denne fasen tilsvarer prosessen bak det som blir beskrevet som kodekapitlet i prosjektrapporten.

Kodefasen var den som tok lengst tid og den som krevde mest arbeid. Det var en vanskelig fase fordi Vue ikke er et språk vi er komfortable med fra skolen. Heldigvis hadde to av

gruppemedlemmene tidligere erfaring med Vue og et av disse fikk seg praksisplass hos Dokflyt i prosjektperioden.

5.1 Arbeidsflyt i kodefase



Figur 19 Arbeidsflyt i kodefase

Arbeidsflyten i å kode kan ses av bilde over og gikk som følger:

1. Første iterasjon
2. Samtaler med produkteier, veileder samt kodevalidering
3. Innsikt fra møter
4. Utvikling av andre iterasjon med
5. Kodevalidering
6. Brukertest av andre iterasjon
7. Utvikling av tredje iterasjon med innsikter fra brukertest

5.2 Nedlasting av nødvendige pakker for miljø

For å kunne kode denne oppgaven måtte vi først laste ned forskjellige pakker og tilpasse miljøet. Vi brukte NPM som tidligere nevnt som packet manager og installer. Vi installerte Vue og bootstrapVue gjennom NPM og utover kodingen installerte vi flere pakker for å få ting til å fungere. Disse var blant annet Axios for Asynkrone ajax forespørsler og Vue PWA for serviceworkers og manifest.

5.3 Arbeid i Git

Arbeidet i Git viste seg å være ganske problemfritt. Siden begge gruppemedlemmene som kodet kunne Git var det kun en gang det oppstå en merge conflict. Gruppemedlemmene var velorganisert når det kom til arbeidstid og arbeidsfiler og vi slapp derfor mye av konfliktene som kan komme ved å jobbe med Git. Videre så hadde vi en rutine om at vi alltid skulle utføre pull før vi pushet og at vi skrudde på automatisert email utsending ved push. En annen

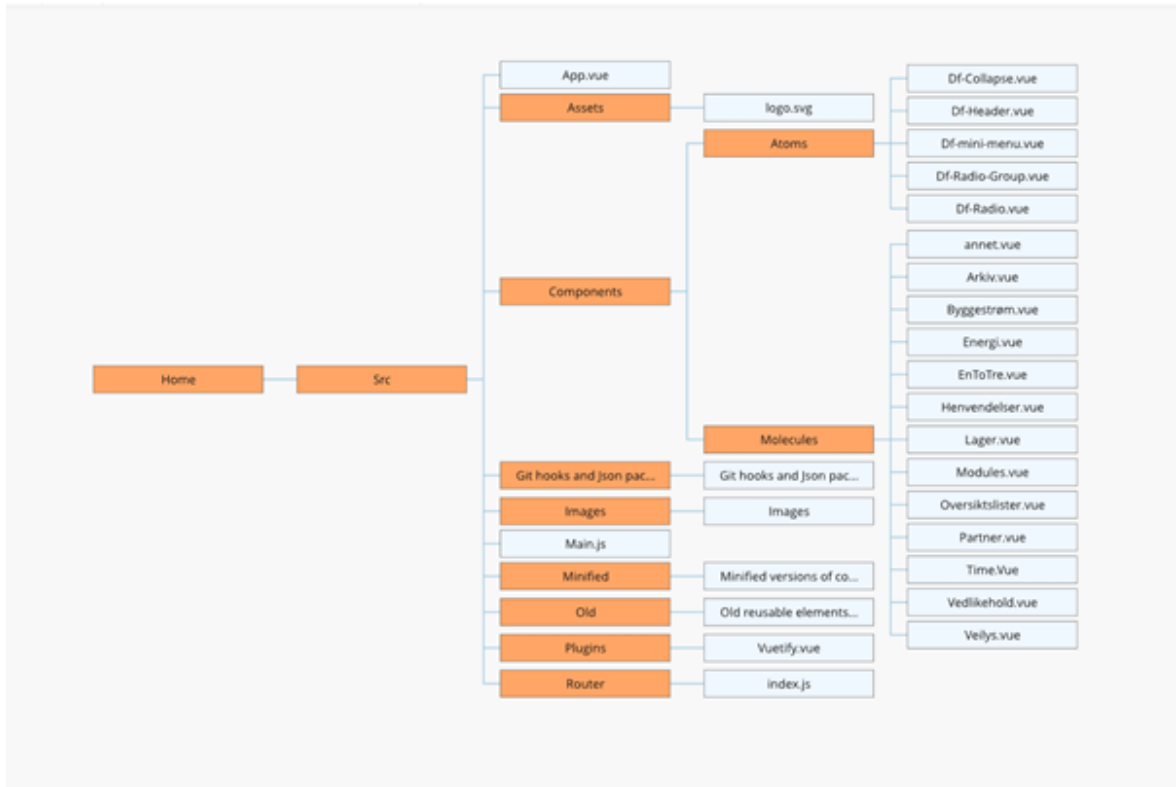
rutine vi fulgte var at hver enkelt committ skulle ha en beskrivelse på engelsk som ga mening for det andre gruppemedlemmet. Vi kommuniserte disse enkle reglene verbalt og dette hjalp oss å effektivisere kodeprosessen.

5.4 Offline

I oppgaveteksten blir det oppgitt at applikasjonen skal fungere offline, måten vi fikk dette til å fungere var å gjøre dette til en PWA. Applikasjonen utenom videoer fungerer offline, grunnen til at videoer ikke gjør det er fordi vi ikke har en database å lagre disse i. Disse kan enten lagres i en database eller alternativt i base 64 i Json apiet. Til tross på at ingen av oss tidligere har laget en PWA gikk det ganske smertefritt ved å følge en forelesning fra MIT Online Courses. Måten applikasjonen fungerer offline på er ved caching og en serviceworker i `webpack.prod.conf.js`. Caching er midlertidig lagring på harddisk i en egen cache fil mens serviceworker er en måte å kjøre script i bakgrunnen separat fra applikasjonen. Vi laget offline koden tidlig i utformingen av applikasjonen slik at vi alltid skulle være sikre på at dette fungerte gjennom hele prosessen. Ettersom vi la til API måtte vi også registrere disse i serviceworker for at disse skulle caches. Måten vi testet om det fungerte offline på var gjennom Google devtools sin application fane. I denne fanen trykket vi på offline og oppdaterte og slik kunne vi se at applikasjonen også fungerer offline.

5.4.1 Navngivning av filer

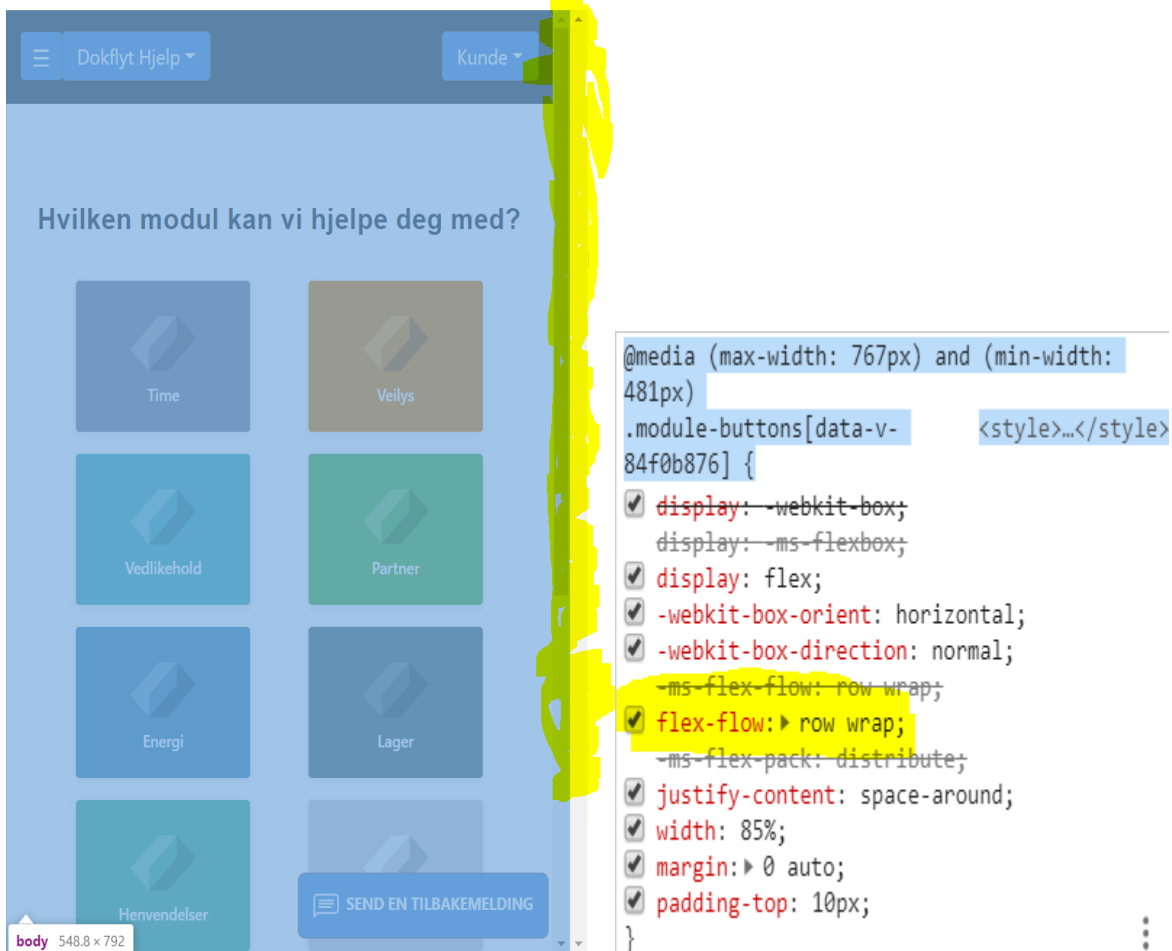
For å gjøre det enklest mulig for Dokflyt valgte vi å følge deres filoppsett med molecules og atoms som egne underfoldere i componenets folderen. Her var logikken at filer som var gjentakende skulle i molecules og ting som var side-komponenter skulle i atoms .Forklart på en annen måte er atomer små komponenter som videre kan brukes i molekyler eller større komponenter. Videre lagde vi en Old mappe for enkeltfiler som kunne brukes videre uten at disse hadde betydning for vårt prosjekt. Vi valgte også å lage minifed versjoner av alle componenets , dette er fordi en stor applikasjon går tregere jo mer bytes filene bruker og minifying kan senke dette betraktelig. Siden minified versjoner er vanskelig å lese gjennom valgte vi å inkludere fulle versjoner for sensor å bruke i denne oppgaven. Utenom dette prøvde vi så godt det lot seg gjøre å følge deres navngivningsregler, som for eksempel at collapse heter Df-Collapse og at filnavn begynner med stor bokstav.



Figur 20 Eksempel på filoppsett med navngivningsregler

5.4.2 Bugfix

Når vi oppdaget en bug så tok vi en screenshot og sendte den i Slack gruppen. Disse ble satt opp som egne oppgaver i Trello som vi senere fikset opp i. Ved utsending av buggen i Slackgruppen fulgte det også med en beskrivelse om hvordan vi kunne replikere buggen og hvis vi visste, grunnen til buggen. Vi brukte forskjellige verktøy i VSC til å debugge men det mest brukte var google chromes utviklerverktøy. Her kunne vi lett finne ut hvor feilen lå og rette opp i denne.



Figur 21 (venstre) Skjermdump av eksempel på bugfix på side

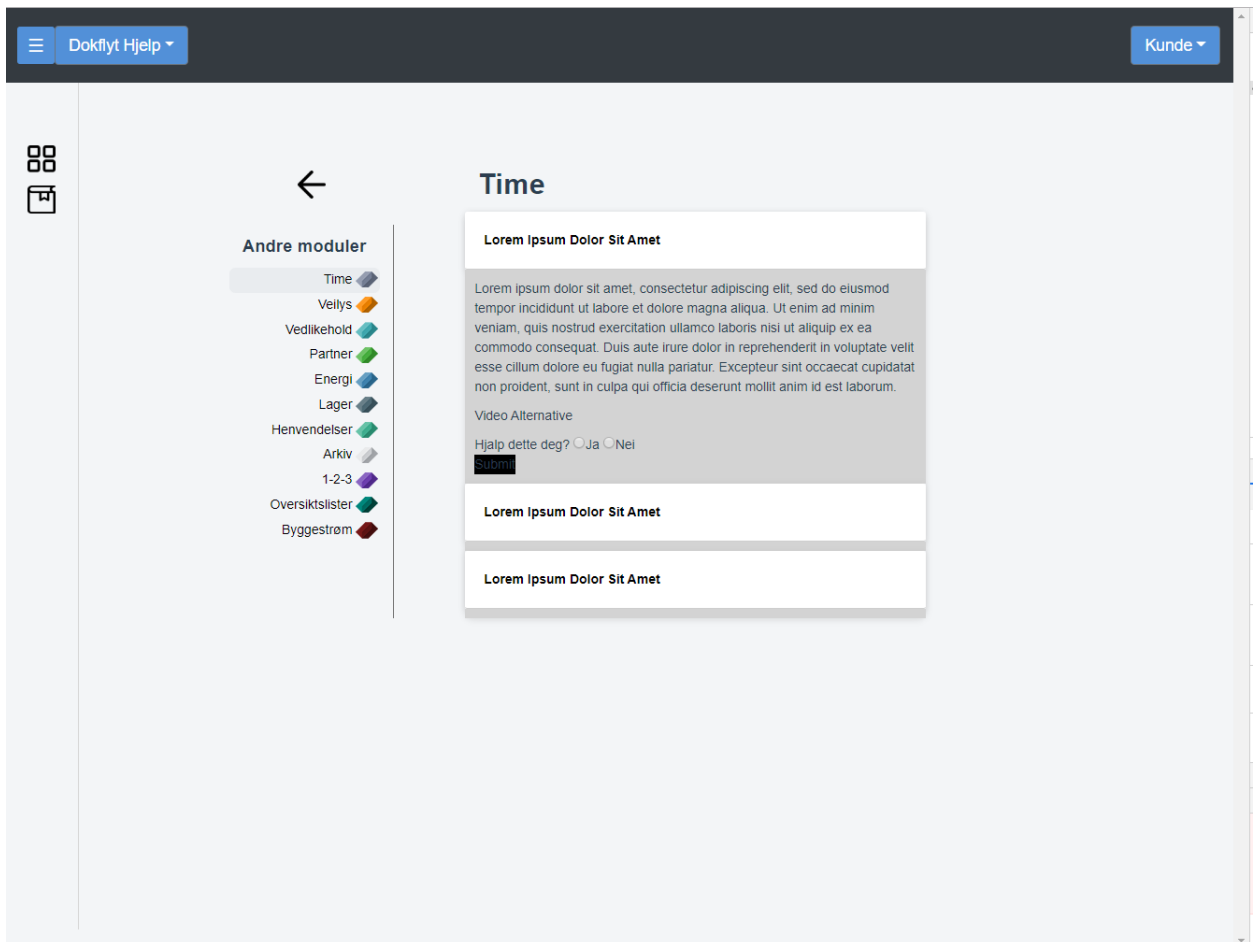
Figur 22 (høyre) Skjermdump av eksempel på bugfix i utviklervertøy

5.5 Første iterasjon

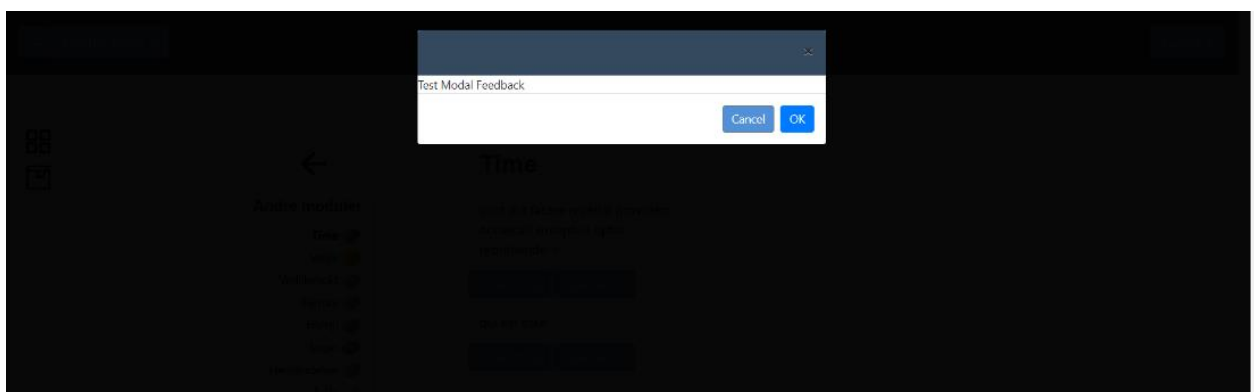
Første versjon av koden var veldig enkel, vi laget applikasjonen til en PWA eller progressive web app ved å bruke en import av Vue-pwa. Dette laget oss flere filer men de viktigste var manifest.json fil, en service-worker.js fil og en register-service-worker.js fil. Disse gjorde at det var mulig for oss å cache siden og ha denne offline. Ved å ha en PWA gjorde vi også at applikasjonen var mulig å laste ned å sette på Home screen. Utenom dette importerte vi et komponent som skulle være våre accordion eller trekkspill. Denne hadde vi lite kontroll over og kom frem til at vi måtte etter hvert erstatte. I denne iterasjonen var feedbackområdet i accordionen en modal hentet inn fra bootstrapVue.

Vi gjorde et forsøk på å installere et bibliotek slik at vi istedenfor å måtte deklare piksler, så kunne vi tatt i bruk breakpoints som er navngitte (mobile, tablet, laptop og

desktop) for media queries. Vi fikk det ikke til å fungere slik vi ønsket og avgjorde dermed at det ikke gjorde stor forskjell om vi brukte det i pikselformat eller ikke. Vi fant ikke ut årsaken til hvorfor den ikke ville fungere, men mistenkte at det skjedde en feil i installeringen av biblioteket som gjorde at den ikke ville bli registrert i applikasjonen.



Figur 23 Skjermdump første iiterasjon



Figur 24 Skjermdump første iterasjon modal



Figur 25 Skjermdump første iterasjon Accordion

5.6 Møte med produkteier og veileder samt kodevalidering

Sammen med produkteier gikk vi gjennom forskjellige funn av hi-fidelity prototypen og oppklarte hva vi kom til å inneholde. En annen ting vi kom frem til med Dokflyt var at vi skulle inkludere en Json API service som hentet inn data til accordion. Dette er for å gjøre det enklere for produkteier å hente inn informasjon siden de i produksjon bruker et egenlaget Json Api service. Videre fikk vi vite av produkteier at de hadde et eget feedback komponent de skulle implementere selv, men vi valgte å gjøre ting enklere ved å inkludere unike Ider basert på Iden hentet ut fra APIet.

5.7 Reevaluering etter møte med produkteier og veileder samt kodevalidering

I kodens andre versjon fortsatte vi med Vue-Pwa men laget en mer eller mindre fungerende applikasjon. Etter møte med produkteier og veileder kom vi frem til forskjellige deler vi måtte endre. Vi tok blant annet å la til en unik ID per spørsmål etter produkteiers ønske slik at det blir enklere for dem å bruke dette til tracking videre.

Resultat av reevaluering av Kode etter møter og kodevalidering

- La til Unik ID

5.8 Andre iterasjon

Andre iterasjon av accordion var at vi hentet inn data fra et Json API ved hjelp av axios istedenfor dummy data. Dette diskuterte vi med Dokflyt og kom frem til at det gir mer verdi for produkteier siden de selv bruker Json API. For bruk av denne applikasjonen er det i




prinsippet bare å forandre API med APIendepunkter så er denne brukbar. Resultatet ble til slutt en selvlagt accordion som hentet ut data ved hjelp av et JSON api. Det var også i denne fasen vi kom frem til at vi ikke skal lage et adminpanel. Grunnen for dette var at produkteier ikke ønsket dette på grunn av sikkerhet og at de allerede hadde et admin komponent de skulle bruke der. Vi endret også litt struktur på kode etter svarene vi fikk fra kodevalideringen for å sørge for at applikasjonen kjørte så hurtig som mulig.

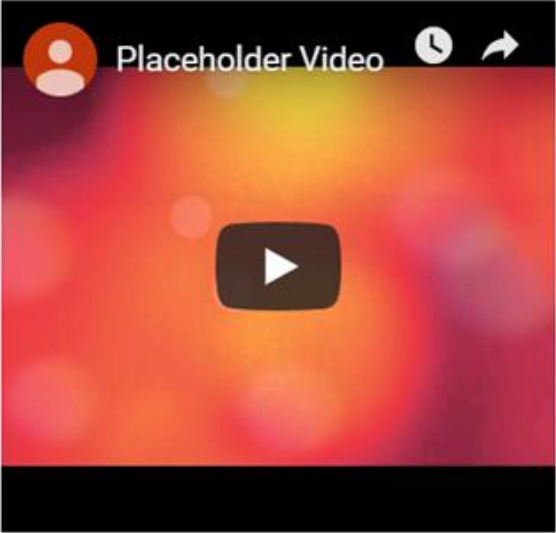
Her ble også feedback knappen endret på mobil slik at den stod feedback istedenfor tilbakemelding eller bare et ikon. Dette var fordi vi i brukertesten av hi-fi prototypen fikk tilbakemeldinger om dette.

Energi

Hvordan legge til brukere ^

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

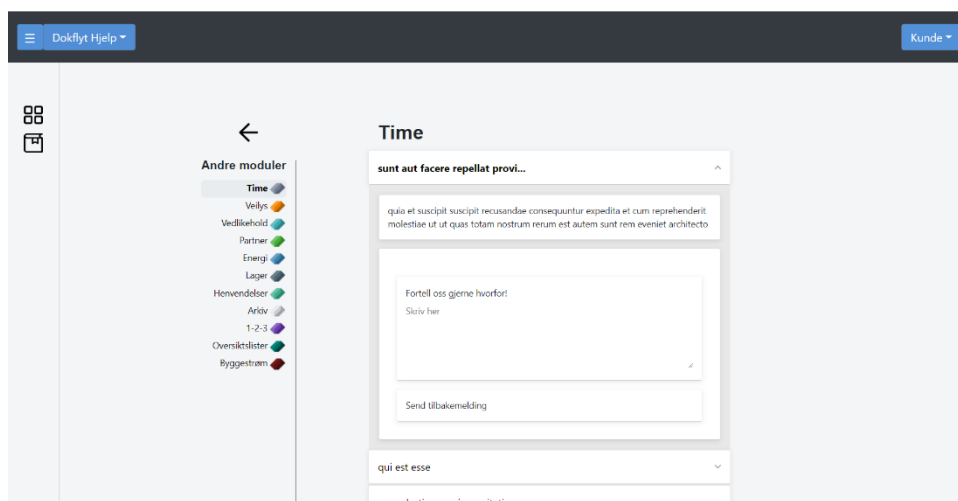
 Placeholder Video  



Løste dette problemet? ☐ Ja ☐ Nei Bekreft

Figur 26 Skjermdump andre iterasjon

Logikken bak accordion feedback måtte også utvikles slik at dokflyt enkelt kunne legges til et administrasjonspanel. Dette løste vi dette ved å lage en egen boks under hvert element. Logikken bak feedback-funksjonen var at den skulle følge Nielsens heuristikk om tilbakemelding, den skulle gi mening for bruker og den skulle være enkel å spore. Vi lagde derfor en customID for hvert enkelt svar, som samsvarte med IDen hentet ut fra API og om de svarte ja eller nei på spørsmålet om det hjalp. Videre så kom de til et feedback-skjema om det ikke hjalp og takk for tilbakemelding om det hjalp. Dette var fordi vi kom frem til etter en diskusjon at det ikke var noe poeng i å ha et feedback-skjema hvis de syntes at svaret var godt nok. Vi diskuterte om denne funksjonen var god nok eller om flere iterasjoner trengtes. Etter samtale med produkteier fikk vi vite at de har et eget komponent for feedback som de kommer til å bruke derfor lot vi siste versjon stå som en dummy. Denne følger Nielsens heuristikker og er grunnet Dokflyt sitt svar kom vi frem til at denne er god nok for formålet.



Figur 27 Skjermdump andre iterasjon

5.9 DF-header

Vi startet å utvikle headeren med alle knappene, og deretter sidemenyen. Vi har tatt i bruk bootstrap sine komponenter for «buttons» og «dropdowns», ettersom vi hadde ressursene for dette. Dette gjør det enklere for oss med tanke på at vi bruker en ferdiglaget komponent slik at vi selv ikke trenger å utvikle komponenten fra bunnen av. Dette er tidsbesparende og gir

samme resultat som det hadde gjort om vi hadde designet og utviklet knappene fra bunn. Sidemenyen er derimot utviklet fra bunn for å få utseende og funksjonalitet mest mulig likt Dokflyt sin. Da utviklingen av dette elementet startet prøvde vi ut ulike ferdiglagde sidemenyer fra ulike kodebiblioteker med den funksjonaliteten vi ønsket. Det vi fort fant ut var at det ble vanskeligere for oss å manipulere og endre på disse komponentene ettersom de var ferdigutviklet med komponenter som var vanskelig å endre på i funksjonalitet og design. Derfor valgte vi til slutt å utvikle sidemenyen fra bunn ved bruk av HTML, CSS og noe Javascript for funksjonalitet.

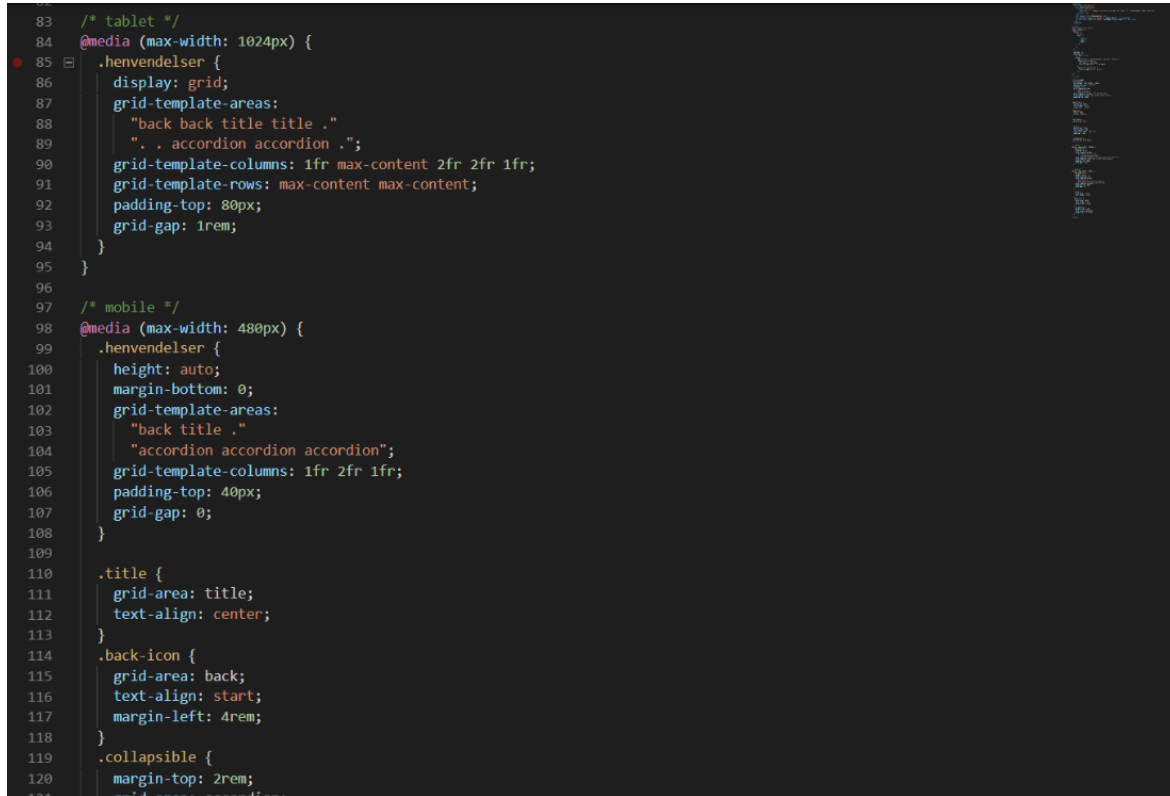
```
vue-pwa > src > components > atoms > df-header.vue > {} "df-header.vue" > template > div > div.nav > b-dropdown.nav-dropdown-user
1 <template>
2 <div>
3   <div class="nav">
4     <span class="nav-icon" v-on:click="expand = !expand">&#9776;</span>
5
6     <b-dropdown class="nav-dropdown-modules" size="lg" v-model="selected" text="Dokflyt Hjelp">
7       <b-dropdown-item>{{modules.veilyls}}</b-dropdown-item>
8       <b-dropdown-item>{{modules.utstyr}}</b-dropdown-item>
9       <b-dropdown-item>{{modules.henvendelse}}</b-dropdown-item>
10      <b-dropdown-item>{{modules.vedlikehold}}</b-dropdown-item>
11      <b-dropdown-item>{{modules.energi}}</b-dropdown-item>
12      <b-dropdown-item>{{modules.oversiktslister}}</b-dropdown-item>
13    </b-dropdown>
14    <b-dropdown class="nav-dropdown-user" size="lg" text="Kunde">
15      <b-dropdown-item>Logg ut</b-dropdown-item>
16    </b-dropdown>
17
18    <div class="sidenav sidebar" :class="{ 'sidenav-toggle-expand': expand === true}">
19      <div v-show="!expand" class="sidenav-close">
20        <router-link
21          to="/"
22          class="nav-link-close"
23          :class="{ '--menu-link-closed' : pathName === 'modules'}"
24        >
25          
30        </router-link>
31        <router-link
32          to="annet"
33          class="nav-link-close"
34          :class="{ '--menu-link-closed' : pathName === 'annet'}"
35        >
36          
41      </div>
42    </div>
43  </div>
44</template>
```

Figur 28 Skjermdump av kode for Df-Header

5.10 Media Queries

For å gjøre applikasjonen responsiv tok vi i denne iterasjonen i bruk manuell skriving av Media Queries i CSS. Gjennom prosessen av å gjøre applikasjonen responsiv reflekterte vi rundt om at dette var en metode som er blant annet veldig tidskrevende, og at det finnes mer praktiske måter å gjøre dette på i kombinasjon av slik vi gjorde det. Det finnes egne biblioteker som kan hentes inn i HTML som egne funksjoner for å deklarerer om visse elementer skal vises i desktop eller mobil. Dette kunne ha vært en mer praktisk og

tidsbesparende metode å kode løsningen på, i tillegg til å ha bidratt til ryddigere og renere kode ettersom vi hadde trengt å skrive mindre linjer med kode. Slik vi gjorde det var å deklareere synligheten av et element basert på skjermstørrelsen i CSS med forskjellige Media Query objekter for de ulike skjermstørrelsene.



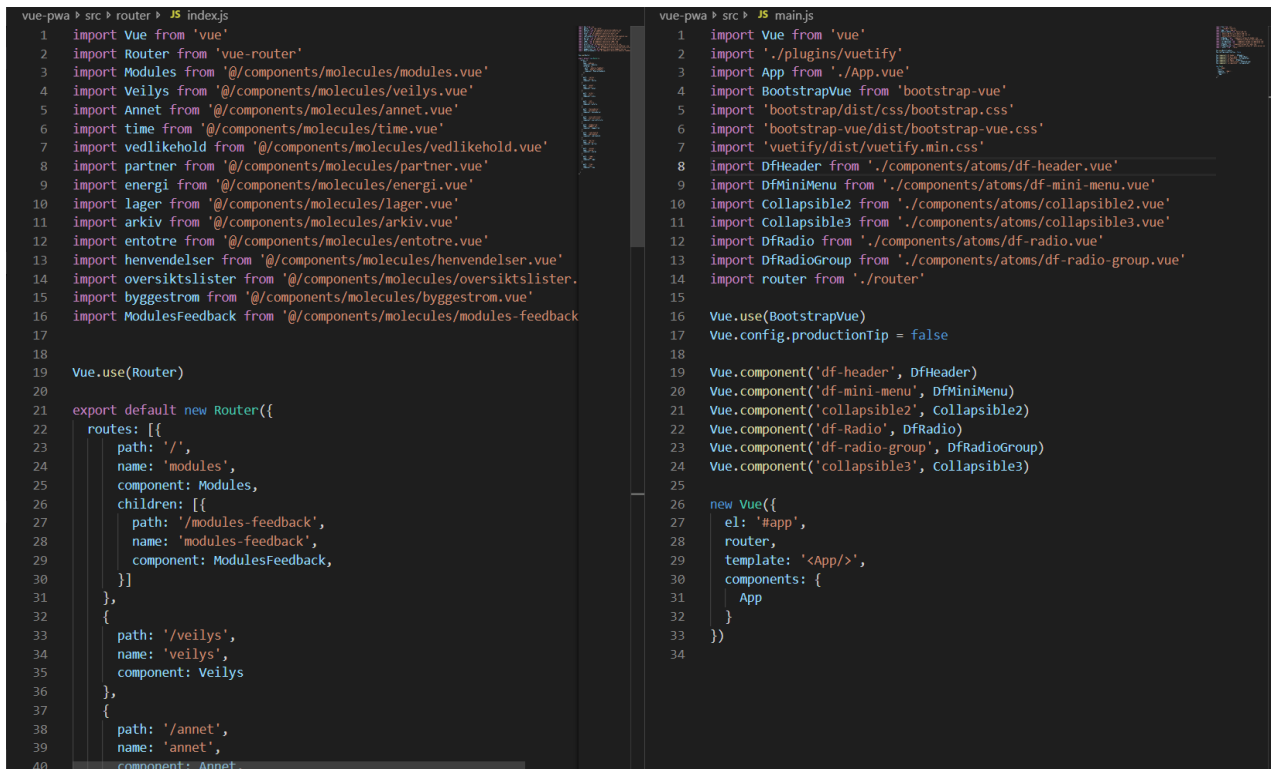
Figur 29 Skjermdump av media queries

5.10.1 Routing

Routingssystemet til Vue er muligens noe av det vi brukte mest tid på å lese oss opp på for å forstå og vedlikeholde et rent og ryddig system. I prosjektmappen hadde vi en main.js-fil og en index.js-fil. Det vi hadde problemer med å skille var at disse to filene skulle inneholde og fungere på to ulike måter hvor lagring av komponenter skulle foregå i main.js, mens lenker mellom sider skulle foregå i index.js. Denne forvirringen oppstod av at det var mulig å gjøre disse to oppgavene i main.js-filen, som gjorde oss usikre på hva hensikten med index.js-filen var.

Vi leste oss opp på dokumentasjonen til Vue sitt routingsystem og komponentregistrering, i tillegg til videoer for læring i praksis. Ved å skille disse to filene, gjør vi applikasjonen bedre integrert for Vue sitt filsystem ettersom det er slik prosjekter skrevet i Vue er strukturert. Da vi fikk skilt mellom index.js-filen og main.js-filen og lagt alle komponentene

der de skulle fikk vi et mye ryddigere system som vi hadde mer kontroll over i form av både kode- og mappestruktur. På bildet nedenfor kan vi se hvordan routingsystemet fungerer og logikken bak det.



```
vue-pwa > src > router > JS index.js
1 import Vue from 'vue'
2 import Router from 'vue-router'
3 import Modules from '@/components/molecules/modules.vue'
4 import Veilys from '@/components/molecules/veilys.vue'
5 import Annet from '@/components/molecules/annet.vue'
6 import time from '@/components/molecules/time.vue'
7 import vedlikehold from '@/components/molecules/vedlikehold.vue'
8 import partner from '@/components/molecules/partner.vue'
9 import energi from '@/components/molecules/energi.vue'
10 import lager from '@/components/molecules/lager.vue'
11 import arkiv from '@/components/molecules/arkiv.vue'
12 import entotre from '@/components/molecules/entotre.vue'
13 import henvendelser from '@/components/molecules/henvendelser.vue'
14 import oversiktslister from '@/components/molecules/oversiktslister'
15 import byggestrom from '@/components/molecules/byggestrom.vue'
16 import ModulesFeedback from '@/components/molecules/modules-feedback'
17
18
19 Vue.use(Router)
20
21 export default new Router({
22   routes: [{
23     path: '/',
24     name: 'modules',
25     component: Modules,
26     children: [{
27       path: '/modules-feedback',
28       name: 'modules-feedback',
29       component: ModulesFeedback,
30     }]
31   },
32   {
33     path: '/veilys',
34     name: 'veilys',
35     component: Veilys
36   },
37   {
38     path: '/annet',
39     name: 'annet',
40     component: Annet,
```

```
vue-pwa > src > JS main.js
1 import Vue from 'vue'
2 import './plugins/vuetify'
3 import App from './App.vue'
4 import BootstrapVue from 'bootstrap-vue'
5 import 'bootstrap/dist/css/bootstrap.css'
6 import 'bootstrap-vue/dist/bootstrap-vue.css'
7 import 'vuetify/dist/vuetify.min.css'
8 import DfHeader from './components/atoms/df-header.vue'
9 import DfMiniMenu from './components/atoms/df-mini-menu.vue'
10 import Collapsible2 from './components/atoms/collapsible2.vue'
11 import Collapsible3 from './components/atoms/collapsible3.vue'
12 import DfRadio from './components/atoms/df-radio.vue'
13 import DfRadioGroup from './components/atoms/df-radio-group.vue'
14 import router from './router'
15
16 Vue.use(BootstrapVue)
17 Vue.config.productionTip = false
18
19 Vue.component('df-header', DfHeader)
20 Vue.component('df-mini-menu', DfMiniMenu)
21 Vue.component('collapsible2', Collapsible2)
22 Vue.component('df-radio', DfRadio)
23 Vue.component('df-radio-group', DfRadioGroup)
24 Vue.component('collapsible3', Collapsible3)
25
26 new Vue({
27   el: '#app',
28   router,
29   template: '<App/>',
30   components: {
31     App
32   }
33 })
34
```

Figur 30 Skjermdump av kode i Index.js for routing

5.10.1.1 Brukertest kode

Vi brukertestet 30 personer for å finne ut av hva de likte, ikke likte og hvordan den generelle flyten i applikasjonen var. Grunnet at vi tidligere ikke hadde fått nok intervjuobjektet for å få til en representativ test tok vi denne gangen og brukertestet folk som ikke nødvendigvis hadde noe forhold til Dokflyt. Vi mente at vi hadde kommet langt nok i koden til at den skulle kunne brukes av alle og derfor ville vi heller ha en representativ gruppe over dokflyts faktiske brukere. Brukertesten foregikk ved at vi intervjuet intervjuobjektene på semi strukturert vis og oppgaver som skulle representere brukervennlighets testing.

5.10.2 Reevaluerer etter brukertest

Innsiktene fra brukertesten kom i stor grad av funksjoner som produkteier ikke ønsket eller at det var utenfor vårt ansvarsområde. Et forslag som Facebook chat går på SoMe og er utenfor vårt ansvarsområde da det er Dokflyt som må ordne dette. En chat var en annen funksjon som

ble foreslått men dette var fra produkteier sin side ikke ønsket da målet med hele denne oppgaven var å senke antall forespørsel inn til Dokflyt og hjelpe kundene å bli mer selvstendige. En chat ville i dette tilfellet mulig ha blitt brukt som førstevalg ved kontakt angående spørsmål som de kanskje finner svar på i FAQ. Videre diskuterte vi om vi ønsket å implementere en kontakt oss side. På en side er dette noe som kommer til å hjelpe på ved at brukerne får mulighet til å kontakte Dokflyt enklere, men fra produkteiers side går dette imot målet for oppgaven som er å senke antall henvendelser til kundeservice. Vi valgte til slutt å implementere dette men sørget for å skrive at de fleste spørsmål og svar finner de i FAQ og prøvde å henvende brukerne til dette.

Designmessig valgte vi å gjøre ikon og tekst større på applikasjonen da det kom frem at dette var i minste laget noen steder.

Resultat av reevaluering av Kode etter brukertest

- Revurderte elementer å inkludere
- Ikon og tekst større

5.11 Tredje iterasjon

I siste del av kodingsfasen sluttførte vi koden ved at alle manuelt gikk gjennom siden og markerte bugs. Vi testet også koden som det står om i informasjonsinnhentingsfasen automatisk. Der vi fant Bugs løste vi disse med en gang fremfor å lage en slack-post og ta det etter hvert. Vi gikk gjennom koden og kommenterte alt slik at Dokflyt enkelt kunne endre på denne om ønskelig. Det ferdige produktet inneholder også minified versjoner av både atoms og molecules slik at dokflyt om ønskelig kan bruke disse for å spare tid og plass. Vi valgte å ikke slette de normale versjonene fordi det er vanskelig for en sensor å rette minified kode. Til slutt utformet vi også en readme.txt slik at både sensor og dokflyt enkelt kunne bygge opp denne applikasjonen og sette seg inn i den.

5.12 Problemer i kodefase

Det var mange små problemer i kodefase hvor de fleste gikk ut på syntax feil, feil bruk av kode eller lignende. Disse ble løst enten ved google, stackoverflow eller ved at det andre gruppemedlemmet som kodet tok over. Det eneste store problemet vi kom over i kodefase var at vi måtte balansere oppgaven mellom å ha god kode som ga verdi for produkteier og at

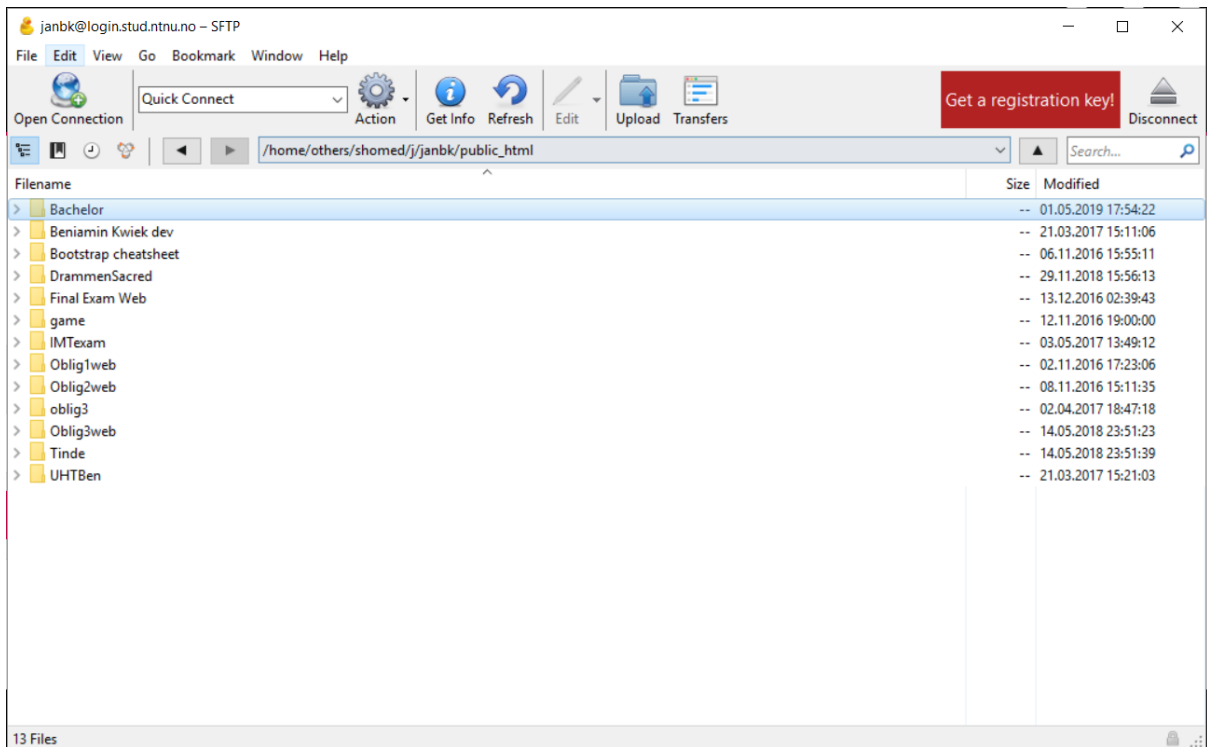
vi prioriterte nok tid til rapporten for å gi oss verdi, siden det var rapporten som ga karakteren. Her var det mange forskjellige interessenter som prøvde å dra oss i forskjellige retninger og vi måtte ta en middelvei for å prøve å gjøre det best mulig for alle parter. Naturligvis ville produkteier at koden skulle være noe de enkelt kunne bruke videre, mens veileder og andre lærere ved NTNU mente at vi måtte prioritere rapport over kode. Selv om vi leverte en fungerende kode som Dokflyt var fornøyd med da vi viste denne er det flere elementer vi hadde tatt med hvis Dokflyt hadde vært enige. Dette er blant annet “annen dokumentasjon”, et annerledes feedback felt og å koble hele applikasjonen opp mot en database. En stor ting vi absolutt ville hatt med men som vi ikke fikk Dokflyt med på var et adminpanel som kan ses av Hi-Fi prototypen. Siden produkteier ikke ville vi skulle implementere disse funksjonene eller at disse funksjonene uansett skulle erstattes lot vi være å lage de.

5.13 Slutføring av kode

I siste del av kodingsfasen sluttførte vi koden ved at alle manuelt gikk gjennom siden og markerte bugs. Vi testet også koden som det står om i informasjonsinnhentingsfasen automatisk. Der vi fant Bugs løste vi disse med en gang fremfor å lage en slack-post og ta det etter hvert. Vi gikk gjennom koden og kommenterte alt slik at Dokflyt enkelt kunne endre på denne om ønskelig. Det ferdige produktet inneholder også minified versjoner av både atoms og molecules slik at dokflyt om ønskelig kan bruke disse for å spare tid og plass. Vi valgte å ikke slette de normale versjonene fordi det er vanskelig for en sensor å rette minified kode. Til slutt utformet vi også en readme.txt slik at både sensor og dokflyt enkelt kunne bygge opp denne applikasjonen og sette seg inn i det.

5.14 Opplasting av oppgave

Vi valgte å laste oppgaven opp på vårt eget folk.ntnu.no domene slik at sensor kunne se helheten i den fungerende applikasjonen. Til dette brukte vi en FTP klient som heter cyberDuck som vi også tidligere har brukt ved opplasting av filer til vårt private nettområde.



FIGUR 31 SKJERMDUMP AV CYBERDUCK CONNECTION

Videre laget vi en mappe som het Bachelor som skulle inneholde bacheloroppgaven. I denne mappen overførte vi koden etter å ha gjort endringer i filene slik at den kunne legges ut.

Link til oppgaven finnes her: <http://folk.ntnu.no/rubenkv/dist/#/>

6. Case study

Som en avsluttende forklaring på oppgaven laget vi en illustrativ case study basert på personas og scenarios utviklet i designfasen. Case studyen er laget etter koden vi kom frem til og viser hvordan personene løser casene og er ment for å illustrere applikasjonen i bruk fremfor å bli brukt som en vitenskapelig studie. Case studyen er delt opp i scenario og personas til bruker og til saksbehandler og skal vise både produkteier og sensor hvordan det er tiltenkt at vår løsning løser oppgaven gitt.

7. Avslutning

7.1 Avsluttende tanker

Selv om vi som gruppe er fornøyd over prosessen i denne oppgaven har vi gjort oss noen refleksjoner over ting vi hadde gjort annerledes. Først og fremst gjelder dette informasjonsinnhentingsprosessen, hvor vi i stor grad la vår tiltro til at Dokflyt kom til å skaffe gode og nok intervjuobjekt til at dette skulle være representabelt for alle kunder. Hadde vi kunne gjort prosjektet på ny hadde vi brukt kunder av lignende selskap som vi skaffet selv. En annen refleksjon vi har gjort oss er at vi brukte komponenten Time som mal, for deretter å finne ut at time hadde mange ting som måtte endres før den var brukbar. Dette var tidskrevende og tok verdifull tid i kodefase. En siste refleksjon er at vi gjennom denne oppgaven har lært utrolig mye, og at NTNU burde vurdere å forandre pensum til å inkludere mer oppgaver som ligner på denne. Fra tidligere hadde vi på skolen ikke kodet i Vue men det viser seg at det er det raskest voksende JS rammeverket ute, og er etter vår erfaring og mening enklere å lære enn lignende rammeverk som Angular og React.

Referanser

1. Vladimir Novkov. 16Personalities [Internett]. NERIS Analytics Limited. [sitert 8. mars 2019]. Tilgjengelig på: <https://www.16personalities.com/articles/our-theory>
2. Lyons N, Meghan Wilker. Interactive Project Management: Pixels, People, and Process (Voices That Matter). 1st utg. Nolan M, Anderson MS, Borman C, redaktører. Berkeley: Geek Girls Guide, LLC; 2012.
3. Carl Gustav Jung. Psychological types [Internett]. Translated by Adler, Gerhard; Hull RFC [1979], redaktør. Collected Works of C. G. Jung. Princeton University Press; 1929. Tilgjengelig på: <https://www.jstor.org/stable/j.ctt5hhqtj>
4. Mbti-notes. MBTI Charts (Cognitive functions). Tumblr [Internett]. Tilgjengelig på: <https://mbti-notes.tumblr.com/charts>
5. Berens L. Cognitive Functions as Polarity Systems [Internett]. typologycentral.com. 2012 [sitert 24. april 2019]. Tilgjengelig på: <https://www.typologycentral.com/forums/myers-briggs-and-jungian-cognitive-functions/60002-cognitive-functions-polarity-systems.html>