

Sarah Victoria Berg  
Eirik Kvistad  
Mari Rogne

## Utvikling av «BeNet» - webløsning for praksisemnet IMT3541

Forslag til håndtering av praksisemnet ved institutt for design ved NTNU i Gjøvik i en digital plattform

Bacheloroppgave i Webutvikling  
Veileder: Carlos Vicient-Monllaó

Mai 2021



Sarah Victoria Berg  
Eirik Kvistad  
Mari Rogne

## **Utvikling av «BeNet» - webløsning for praksisemnet IMT3541**

Forslag til håndtering av praksisemnet ved institutt for design ved NTNU i Gjøvik i en digital plattform

Bacheloroppgave i Webutvikling  
Veileder: Carlos Vicient-Monllao  
Mai 2021

Norges teknisk-naturvitenskapelige universitet  
Fakultet for arkitektur og design  
Institutt for design



Kunnskap for en bedre verden



# Sammendrag

**Tittel:** Utvikling av «BeNet» – Webløsning for praksisemnet IMT3541

**Dato:** 14.05.2021

**Deltakere:** Sarah Victoria Berg, Eirik Kvistad, Mari Stalheim Rogne

**Veileder:** Carlos Vicient-Monllaó

**Oppdragsgiver:** Institutt for Design ved Norges Teknisk-Naturvitenskapelige Universitet i Gjøvik

**Stikkord:** Webutvikling, praksis, komponent-drevet-utvikling, designprinsipper, bærekraft

**Antall sider:** 123

**Antall vedlegg:** 9

Prosjektet er gitt på bakgrunn av utfordringer knyttet til gjennomføring av praksisemnet i *IMT3541 – Veiledet praksis i medie- og informatikkfag* ved Institutt for Design hos Norges Teknisk-Naturvitenskapelige Universitet i Gjøvik. Prosjektet er av høy relevans, noe som gjenspeiles i den norske regjeringens mål om mer og bedre praksis (Meld. St. (2020–2021)). Gjennomføring og koordinering av praksisopphold er et sammensatt problem, og prosessen i det aktuelle praksisemnet gjennomføres per dags dato ved hjelp av flere programvarer som ikke er tilrettelagt for en slik prosess. Dette prosjektet undersøker hvordan det kan tilrettelegges for å håndtere dette praksisemnet på en mer oversiktlig måte gjennom å utforme og utvikle en ny webløsning. Prosjektet bygger på innsikt fra to tidligere arbeid som har utforsket samme problemområde. Denne rapporten dokumenterer hvordan prosjektet har utformet og utviklet et minimumprodukt ved navn «BeNet» gjennom en utformingsfase som følger prinsipper om brukersentrert design, samt en utviklingsfase basert på en komponent-drevet tilnærming. BeNet er en webløsning som håndterer praksisemnet ved Institutt for Design på NTNU i Gjøvik.

# Abstract

**Title:** Development of «BeNet» – Web solution for the internship course IMT3541

**Date:** 14.05.2021

**Participants:** Sarah Victoria Berg, Eirik Kvistad, Mari Stalheim Rogne

**Supervisor:** Carlos Viciant-Monllaó

**Employer:** Department of Design at Norwegian University of Science and Technology in Gjøvik

**Keywords:** Web development, internships, component-driven-development, design principles, sustainability

**Number of pages:** 123

**Number of attachments:** 9

This project is based on the challenges related to the handling of the internship course *IMT3541 – Supervised Practicum in Media and Informatics* at the Department of Design at Norwegian University of Science and Technology in Gjøvik. The project is of high relevance, which is reflected in the Norwegian government's goal of more and better internships (Meld. St. (2020–2021)). The handling and coordination of the internship is a comprehensive problem, and the process in the current internship course is currently being handled through several applications which are not suited for such a process. This project examines how to facilitate the handling of the internships in a clearer manner through designing and developing a new web solution. The project is based on insights from two previous works which examined the same problem area. This report documents how the project has designed and developed a minimum viable product named “BeNet” through a design phase which follows principles of user-centered design, as well as a development phase characterized by component-driven development. BeNet is a web solution which handles the internship course at the Department of Design at NTNU in Gjøvik.

# Forord

Dette prosjektet har vært utfordrende på godt og vondt, og vi hadde ikke vært der vi er i dag hadde det ikke vært for noen helt fantastiske mennesker som støttet oss hele veien.

Vi vil takke vår veileder Carlos Vicient-Monllaó for umåtelig engasjement og dedikasjon for å få frem det beste i oss. Vi vil også sende en takk til Emil Bakke for gode råd. Man kommer ikke foruten å nevne vår oppdragsgiver Terje Stafseng og Institutt for Design ved NTNU i Gjøvik for et spennende og utfordrende prosjekt. Venner og familie har stilt opp og hjulpet oss gjennom prosjektet, enten vi behøvde faglig innspill eller noen støttende ord, tusen takk.

Sist, men ikke minst vil vi takke hverandre for et godt samarbeid gjennom en utfordrende periode med digitale møter og lite sosiale aktiviteter, uten oss selv hadde vi ikke klart det!

Gjøvik, 14. mai 2021.

*Sarah Bryn*      *Man S. Rognu*

*Eirik Kvistad*

# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>1</b>
1.1	Prosjektbeskrivelse	2
1.1.1	Utfordringer og problemer med dagens ordning	3
1.2	Problemstilling	5
1.3	Prosjektets formål og omfang	5
1.4	Rapportens oppbygning	6
<b>2</b>	<b>Prosjektorganisering</b>	<b>7</b>
2.1	Samhandlingsverktøy	7
2.2	Gruppeorganisering og kvalitetssikring	8
<b>3</b>	<b>Bakgrunn</b>	<b>10</b>
3.1	Resultater og funn fra tidligere arbeid	10
3.1.1	Webprosjekt II	11
3.1.1.1	Brukergrupper	11
3.1.1.2	Løsning i form av høynivå-prototype	12
3.1.2	Prosjektutvikling i kreativ virksomhet	13
3.1.2.1	Personas og scenarier	14
3.1.2.2	Konkurrentanalyse	16
3.1.2.3	Funksjonelle og ikke-funksjonelle krav	16
3.2	Teori	18
3.2.1	«Mobil først»	18
3.2.2	Brukertrent design	19
3.2.2.1	WCAG	20
3.2.2.2	Heuristikker og designprinsipper	20
3.2.2.3	Dark mode	21
3.2.3	Karbonutslipp og internett	22
<b>4</b>	<b>Utforme</b>	<b>24</b>
4.1	Metoder i utformingsfasen	25
4.1.1	Brukerhistorier	25
4.1.2	Skisser	26
4.1.3	Prototyping	26



4.1.4	Brukertesting .....	27
4.1.4.1	Gjennomføring av brukertester .....	27
4.1.4.2	Utvalg av testbrukere.....	28
4.1.4.3	Brukertestens oppgaver.....	29
4.1.4.4	Målinger av brukertester .....	32
4.2	<i>Resultat av utformingsfasen</i> .....	33
4.2.1	Brukerhistorier.....	34
4.2.2	Skisser .....	36
4.2.3	Første iterasjon av prototype .....	37
4.2.3.1	Målinger av første iterasjon.....	39
4.2.3.2	Drøfting av første iterasjon .....	45
4.2.4	Andre iterasjon av prototype.....	46
4.2.4.1	Målinger av andre iterasjon.....	48
4.2.4.2	Drøfting av andre iterasjon .....	54
4.3	<i>Oppsummering av utformingsfasen</i> .....	54
<b>5</b>	<b>Utvikle</b> .....	<b>55</b>
5.1	<i>Arkitektur og teknologi</i> .....	58
5.1.1	Overordnet arkitektur.....	58
5.1.2	Bakgrunn for teknologistabel .....	59
5.1.2.1	Komponent-drevet utvikling .....	59
5.1.2.2	Applikasjonsarkitektur .....	61
5.1.2.3	Vedvart lagring (databasesystem) .....	63
5.1.2.4	Skyarkitektur og tjenere.....	64
5.1.3	Prosjektets teknologistabel .....	65
5.1.3.1	Frontend.....	66
5.1.3.2	Backend.....	67
5.1.3.3	Database .....	67
5.2	<i>Implementering</i> .....	68
5.2.1	Frontend .....	68
5.2.1.1	Koordinering av komponenter .....	68
5.2.1.2	Mappestruktur .....	70
5.2.1.3	Vuex .....	71
5.2.2	Backend .....	72
5.2.3	Database .....	73
5.2.4	Autentisering .....	75
5.2.5	State/tilstand .....	77

5.2.6	Automatisk sammenstilling basert på prioriteringer .....	78
5.3	<i>Evaluering</i> .....	80
5.3.1	Ekspertevaluering .....	80
5.3.2	Systemtest .....	84
<b>6</b>	<b>Webløsningen «BeNet».....</b>	<b>88</b>
6.1	<i>BeNets håndtering av praksisemnet</i> .....	97
6.2	<i>Bruerguide</i> .....	98
<b>7</b>	<b>Diskusjon.....</b>	<b>102</b>
7.1	<i>En god brukeropplevelse</i> .....	104
7.1.1	Heuristikker og designprinsipper .....	104
7.1.2	En responsiv løsning .....	106
7.1.3	Dark mode .....	106
7.1.4	WCAG.....	107
7.2	<i>En bærekraftig løsning</i> .....	108
7.2.1	Måling av BeNet sitt digitale avtrykk .....	109
7.3	<i>Konklusjon</i> .....	111
7.3.1	Anbefaling til videre arbeid.....	111
<b>8</b>	<b>Litteraturliste .....</b>	<b>114</b>
<b>9</b>	<b>Figurliste .....</b>	<b>120</b>
<b>10</b>	<b>Tabelliste.....</b>	<b>122</b>
<b>11</b>	<b>Vedlegg .....</b>	<b>123</b>

# 1 Introduksjon

Utdanning har som formål å utruste studenter med de kunnskaper og ferdigheter som kreves i arbeidslivet, og på den måten forberede studentene på hva tiden etter endt utdanning vil innebære. Likevel ser man at overgangen til arbeidslivet kan være både utfordrende og meget ulik hverdagen som student (Regjeringen, 2021). Den norske regjeringen fremstilte et mål i stortingsmeldingen som ble godkjent 12. mars 2021 om å sikre bedre sammenheng mellom det man lærer i studiesammenheng og det man skal jobbe med etter endt utdanning (Meld. St. (2020–2021)). Blant annet mener regjeringen at en økt relevans for arbeidslivet i studiene avhenger av et systematisk og bredere samarbeid mellom studiene og arbeidslivet (Meld. St. (2020–2021), s. 19). Det er fire innsatsområder med behov for mobilisering som har blitt identifisert for å nå regjeringens mål (Meld. St. (2020–2021), s. 21):

**Innsatsområde 1:** Aktørene styrker samarbeidet

**Innsatsområde 2:** En åpen og tilgjengelig høyere utdanning

**Innsatsområde 3:** Studentaktive lærings- og undervisningsformer

**Innsatsområde 4:** Mer og bedre praksis

Innsatsområde 4, «mer og bedre praksis», er et mål om at flere studenter skal få relevant praksis i løpet av studiene, da det er en viktig læringsarena (Meld. St. (2020–2021), s.60). Praksisemner lar studentene prøve ut sin egen kunnskap tilegnet fra studiene i omgivelser som er tilnærmet reelle arbeidsforhold, og er derfor svært viktig for en mindre utfordrende overgang til arbeidslivet.

I studien *Bransjerelaterte læringsaktiviteter i medie- og designfag* utførte Terje Stafseth og Jens Barland (2021) undersøkelser angående bransjerelatert læring, inkludert praksis, i medie- og designfag ved Institutt for Design ved NTNU i Gjøvik. Undersøkelsene fastslår at studenter ved instituttet *ønsket* å ta del i det virkelige arbeidslivet. Studien viser til viktigheten av organisering og kvalitetssikring for at studentene skal få et godt læringsutbytte av praksis. For å hjelpe studentene med en mindre utfordrende overgang til arbeidslivet tilbyr Institutt for Design ved NTNU i Gjøvik praksisemnet *IMT3541 – Veiledet praksis i medie- og informatikkfag* til to av tre bachelorprogram på instituttet. Gjennomføringen av praksisemnet kommer ikke uten utfordringer.

Praksisemnet er i dag preget av en løsning som medfører ustrukturert data og en omfattende manuell jobb. Emneansvarlig fungerer som et mellomledd i alle deler av prosessen, noe som tilsier lite selvstendighet for studenter, og gjør det vanskelig å opprettholde et godt samarbeid med

næringsmiljøet. Dette prosjektet har som mål å løse utfordringene i praksisemnet ved å redusere mengden manuelt arbeid for emneansvarlig, forbedre arbeidsflyten, og tillate mer autonomi for de ulike brukergruppene involvert i systemet, og dermed legge til rette for innsatsområde 4 «mer og bedre praksis». Dette gjennom å utforme og utvikle en nettbasert løsning.

## 1.1 Prosjektbeskrivelse

Prosjektets oppdragsgiver er Institutt for design ved NTNU i Gjøvik, med emneansvarlig for praksisemnet, Terje Stafsgeng, som kontaktperson. Instituttet har fem studieprogram, hvor tre av de er bachelorprogram: interaksjonsdesign, webutvikling, og grafisk design (NTNU, u.å.b). Praksisemnet er et obligatorisk emne for interaksjonsdesign og tilbys som et valgfritt emne for webutvikling.

Emnet innebærer ulike oppgaver for emneansvarlig, student, og bedrift. Emneansvarlig må finne og kvalitetssikre bedrifter i nær proksimitet av Gjøvik som tilbys studenter ved interaksjonsdesign. I tillegg må emneansvarlig sørge for at alle studentene *får* praksisplass, dette ved å fungere som mellomledd i søknadsprosessen, samt manuelt sette sammen student og bedrift basert på begge parter ønsker.

Studenter må finne praksisplasser som virker interessante for dem, og skrive og sende søknader til de aktuelle bedriftene. Studenter ved interaksjonsdesign velger om de vil søke til bedrifter som har et samarbeid med NTNU eller om de vil finne bedrift på egenhånd. Studenter ved webutvikling *må* finne en bedrift og opprette en avtale på egenhånd, da de tar praksis som et valgfritt emne.

Bedriftene må sende informasjon om seg selv til emneansvarlig i forkant av praksisperioden. Etter å ha mottatt søknader fra interesserte studenter, må bedriften prioritere de aktuelle kandidatene. Prioriteringene må sendes tilbake til emneansvarlig. Hvilken student som starter hos hvilken bedrift, blir avgjort når den endelige kontrakten er underskrevet. Dagens ordning for praksisemnet blir illustrert steg for steg i Figur 1 i kronologisk rekkefølge. Figuren viser også at alle stegene krever at emneansvarlig (oppdragsgiver) er en deltakende aktør.

# Dagens ordning i praksisemnet



Figur 1: Dagens ordning i praksisemnet (basert på Berg, Kvistad, og Rogne (2020b)).

Håndteringen av emnet foregår i dag på flere digitale plattformer, derav læringsplattformen Blackboard, samarbeidsplattformen Microsoft Teams, og fillagringstjenesten Google Drive. Mye av den individuelle kommunikasjonen mellom emneansvarlig, student, og bedrift foregår gjennom e-poster eller over telefon.

## 1.1.1 Utfordringer og problemer med dagens ordning

Arbeidet til oppdragsgiver er i sammenheng med praksisemnet preget av mye organisering og fasilitering, hovedsakelig med et ansvar for å være et bindeledd mellom studenter og bedrifter, samt å være en veileder for studentene gjennom praksisperioden. Måten emnet er lagt opp på i dag medfører flere utfordringer og problemområder, spesielt for emneansvarlig, men også for studenter og bedrifter. Hovedproblemet er at informasjon *ikke* er samlet på ett sted. Dette problemet påvirker fire områder, som vist i Tabell 1.

Problemområde	Problem
1	Håndtering og deling av data med de ulike aktørene
2	Kommunikasjon
3	Søknad og prioriteringer
4	Historisk data

Tabell 1: Problemområder for dagens ordning basert på funn fra Berg, Kvistad, og Rogne (2020b).

### **Problemområde 1: Håndtering og deling av data med de ulike aktørene**

→ Informasjon er *ikke* samlet på et sted som medfører:

- Komplexitet
- Kontekstbytte
- Lite oversikt
- Vanskelig å finne spesifikk data

Problemområde 1, *Håndtering og deling av data*, påvirker alle involverte. Prosessen med å finne bedrifter forutsetter deling av informasjon frem og tilbake mellom emneansvarlig og aktuelle bedrifter. Studentene er også avhengig av å finne relevant informasjon om deres gjøremål, bedrifter de kan velge og viktige frister. Å forholde seg til mange plattformer fører til mer kompleksitet for prosessen da det krever kontinuerlig kontekstbytte, samt at det fører til en uoversiktlig lagring av informasjon.

### **Problemområde 2: Kommunikasjon**

→ Foregår over flere plattformer som kan føre til:

- Inkonsekvent informasjon

→ Hovedsakelig over e-post, som fører til:

- Informasjon som er vanskelig å finne
- Informasjon kan forsvinne
- Informasjon kan bli glemt

Problemområde 2, *kommunikasjon*, påvirker alle de involverte partene i praksisemnet. Å kommunisere på flere plattformer kan føre til at inkonsekvent informasjon blir delt, noe som er et problem dersom det for eksempel blir formidlet to ulike datoer for frister. Kommunikasjon over e-post utgjør en utfordring for kommunikasjonsprosessen, da det fører til at informasjon kan være vanskelig å finne, eller i verste fall forsvinne eller bli glemt (Berg, Kvistad, og Rogne, 2020b, s. 4–6).

### **Problemområde 3: Søknad og prioriteringer**

→ Bruker emneansvarlig som mellomledd som innebærer:

- Sortering og videresending (e-post)

→ Manuell sammensetning av student og bedrift basert på prioriteringer

- Krever høy konsentrasjon

Problemområde 3, *søknad og prioriteringer*, påvirker i hovedsak emneansvarlig. Alle søknadene, tre fra hver student, må sorteres av emneansvarlig og videresendes til bedriftene. Dette forutsetter høy

konsentrasjon for å sende riktig søknad til riktig bedrift. Det samme gjelder manuell matching, altså sammensetning, av prioriteringer fra bedrifter og studenter.

#### **Problemområde 4: Historisk data**

- Mangler historisk data
- Mangler samlet lagringsplass

Problemområde 4, *historisk data*, påvirker i dag hovedsakelig emneansvarlig. Manglende historikk over tidligere samarbeid medfører unødvendig bruk av tid ved gjennomføring av ny rekrutteringsrunde for hvert år. Samtidig vil manglende historisk data være et problem for en ny emneansvarlig i fremtiden. Administrator mangler også en samlet lagringsplass for all data relatert til praksisemnet. Data blir lagret på flere steder, noe som kan føre til uregelmessig informasjon.

Måten praksisemnet er lagt opp på i dag medfører samtlige fire problemområder som påvirker alle involverte i stor grad. Problemene har størst konsekvens for selve håndteringen for emneansvarlig.

## 1.2 Problemstilling

For å løse praksisemnets problemområder, samt ivareta verdien av praksisemnet for alle involverte, har følgende problemstilling blitt lagt til grunn:

Hvordan tilrettelegge for en mer oversiktlig håndtering av praksisemnet ved Institutt for Design ved NTNU i Gjøvik gjennom å utvikle en digital webløsning?

For å bedre kunne svare på problemstillingen, har prosjektgruppen stilt forskningsspørsmål som er med på å underbygge problemstillingen. For at løsningen skal kunne ivareta universell utforming, ønsker prosjektgruppen å undersøke **forskningsspørsmål 1: hvordan man kan utforme og utvikle en god brukeropplevelse gjennom prinsipper om design og universell utforming**. I tillegg er prosjektgruppen opptatt av å bidra til det globale målet om bærekraft, så det var ønskelig å undersøke **forskningsspørsmål 2: hvilke tiltak prosjektgruppen kan gjøre for å unngå overflødig karbonutslipp relatert til internett**.

## 1.3 Prosjektets formål og omfang

Formålet med dette prosjektet er å forbedre håndteringen og organiseringen av praksisemnet ved Institutt for Design, og tilrettelegge for høyere grad av selvstendighet blant studentene og bedriftene. Prosjektet skal skape et mer systematisk samarbeid mellom Institutt for Design, studenter, og

næringslivet. Prosjektet skal utvikle et minste brukbare produkt, videre referert til som MVP, som viser en bedre håndtering av praksisemnet. Å sikre bedre relevant innhold i det man lærer i studiesammenheng i forhold til det man skal jobbe med etter endt utdanning er utenfor prosjektets omfang.

## 1.4 Rapportens oppbygning

Etter det introduserende kapittelet som tar for seg formålet med prosjektet, vil hvordan prosjektet er organisert bli presentert i kapittel 0. Kapittel 0 vil redegjøre for relevant bakgrunn fra tidligere arbeid og teori som er med på å underbygge prosjektet. Kapittel 0 og 5 vil redegjøre for hvordan løsningen har blitt utformet og utviklet, hvor en gjennomgang av det endelige resultatet vil bli redegjort for i kapittel 6. Resultatet vil bli diskutert i kapittel 7, som også vil gi en konklusjon på problemstillingen, samt forslag til videre utvikling.



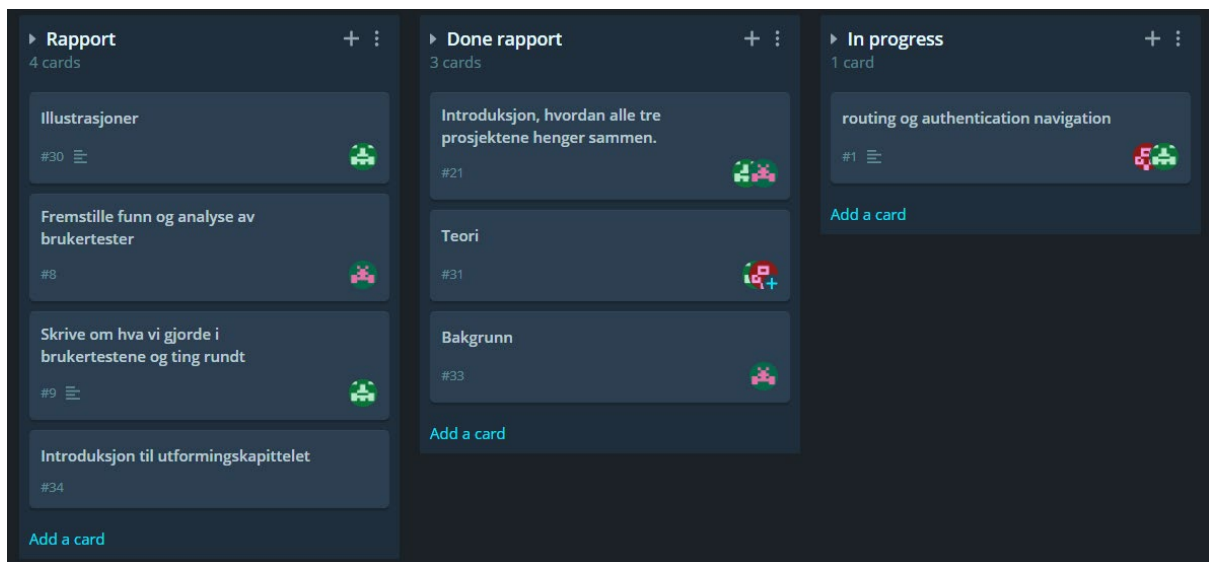
## 2 Prosjektorganisering

Prosjektgruppen består av tre medlemmer. Perioden prosjektet ble gjennomført i har til en viss grad vært preget av tiltak rundt Covid-19 pandemien, som blant annet har hatt utslag på muligheter for fysiske møter mellom medlemmene og møter med veiledere og oppdragsgiver. Dette har blitt løst ved å holde de aller fleste møter digitalt. Dette kapitlet redegjør for hvordan prosjektgruppen har organisert og fordelt arbeidet, samt hvilke samhandlingsverktøy som har blitt anvendt.

### 2.1 Samhandlingsverktøy

Det har vært viktig å kunne samarbeide i form av å se hva alle arbeider med, i tillegg til å kunne kommunisere det. Samhandlingsverktøy lagt til grunn i prosjektet består av flere forskjellige plattformer som anvendes til hvert sitt formål. Microsoft Teams har vært hovedkanalen for muntlig kommunikasjon både innad i gruppen og ved veiledninger og andre møter med andre aktører, da digitale møter har vært hovedregelen grunnet restriksjoner i forbindelse med Covid-19. I tillegg har det vært en kontinuerlig skriftlig kommunikasjon gjennom Facebook Messenger. Dokumenter og andre fildelinger har foregått på Google Drive, som tilrettelegger for redigering i sanntid, i likhet med prototypingsverktøyet Figma, som ble brukt for å utforme prototypen. Rapporten har blitt formatert og skrevet i Microsoft Word Online.

GitHub, som er et versjonskontrollsystem, inneholder koden til løsningen. Et versjonskontrollsystem tilrettelegger for samarbeid og deling av kode uavhengig av lokasjon. GitHub kan synkroniseres opp mot GitKraken. GitKraken har blitt brukt for å organisere samt fordele arbeidet i form av kanban-boards (se Figur 2). Kanban-boardet har gjort det mulig å holde god oversikt over hva som må gjøres, hva som er under arbeid, hva som har blitt gjort og samtidig hvem som har deltatt i de ulike aktivitetene.



Figur 2: Skjerm bilde av et utdrag kanban-boards i GitKraken.

## 2.2 Gruppeorganisering og kvalitetssikring

Arbeidet har blitt fordelt jevnt mellom de tre medlemmene, med et formål om at alle skal ha mulighet til å være innom alle de forskjellige rollene i et prosjekt. Av den grunn har hovedansvaret for organisering blitt fordelt i løpet av prosessen. For eksempel har rollen som prosjektleder variert, og er en rolle alle medlemmene har fått erfare (se Tabell 2). En prosjektleder i prosjektet har ansvar for å strukturere arbeidet, gi retning til de andre medlemmene og holde oversikt over den overordnede planen.

Medlem	Hovedansvar
Sarah Victoria Berg	Prosjektleder for prosjektet
Mari Stalheim Rogne	Prosjektleder for utformingsfasen
Eirik Kvistad	Prosjektleder for utviklingsfasen

Tabell 2: Fordeling av hovedansvar for medlemmene av prosjektgruppen.

Arbeidet med utforming og utvikling ble i første omgang organisert i et overordnet GANTT-skjema (se Figur 3). Se vedlegg 9 for hele skjemaet. I tillegg førte man de mindre stegene, samt alle arbeidsoppgaver tilknyttet andre ting, som for eksempel gjøremål i forhold til denne rapporten og relevante vedlegg, inn i GitKraken. På den måten har alle medlemmene mulighet til å se hva som jobbes med av hvem, hva som er ferdig og hva som gjenstår.

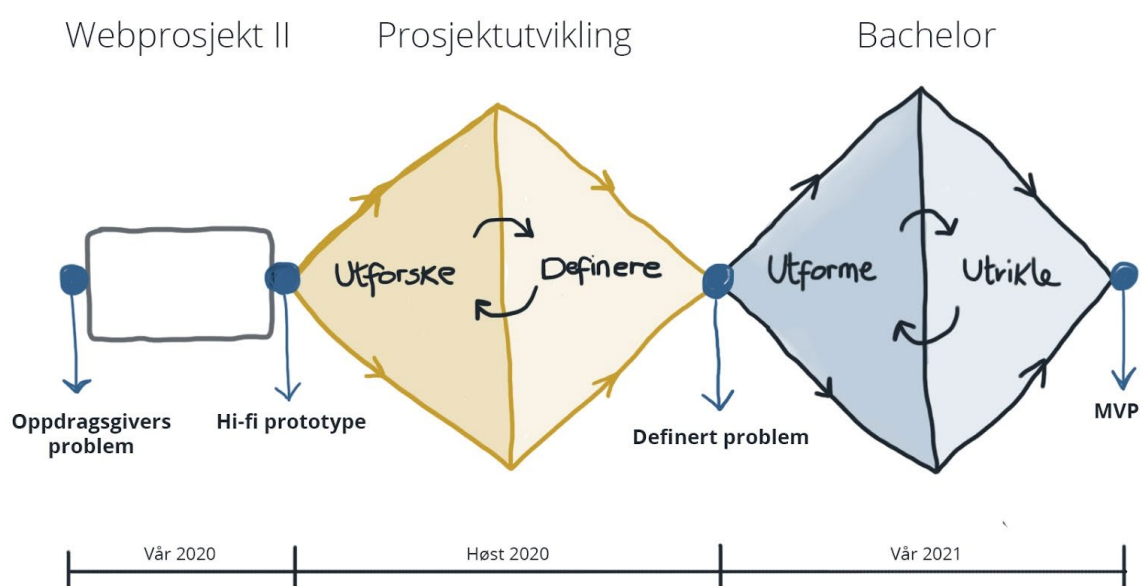


### 3 Bakgrunn

Prosjektet bygger videre på to tidligere prosjekter gjennomført av prosjektgruppen i 2020, noe som er vesentlig å legge frem da det er av betydning for utformingen og utviklingen av dette prosjektet. Dette kapitlet presenterer derfor bakgrunnen for prosjektet, det vil si et sammendrag av de viktigste resultatene og funnene fra de to foregående prosjektene, og i den sammenheng hvilke erfaringer som blir tatt med inn i dette bachelorprosjektet. I tillegg presenteres relevant teori som har blitt lagt til grunn, og som vil bli diskutert i sammenheng med resultater i diskusjonskapitlet.

#### 3.1 Resultater og funn fra tidligere arbeid

I emnet *IMT2671 – Webprosjekt II* (se vedlegg 1 av Berg, Kvistad, Rogne (2020b)), som ble gjennomført på våren 2020, ble oppdragsgivers problem introdusert for første gang. Ved gjennomført emne hadde en høynivå-prototype blitt utviklet. Herfra startet arbeidet som legger grunnlaget for dette prosjektet, ved å iterere og igjen undersøke de samme utfordringene i et større og mer omfattende innsiktsarbeid gjennom den første diamanten i double diamond-modellen. Dette ble gjort gjennom et fordypningsprosjekt i emnet *IMT3007 – Prosjektutvikling i kreativ virksomhet* (vedlegg 2 av Berg, Kvistad, Rogne (2020a)) på høsten 2020. Innsiktsarbeidet innebar blant annet intervjuer og spørreskjema, bearbeiding av empiri, samt analyse og brukertesting av den eksisterende prototypen i sammenheng med denne nye innsikten. Det er viktig å skape et klart skille mellom hva som tidligere har blitt gjort og hva som blir gjort i dette prosjektet. Figur 4 viser tidslinjen for de ulike prosjektene, hvilke faser de innebar, samt hva utfallet var fra hvert prosjekt.



Figur 4: Sammenhengen mellom prosjektene. Illustrert av Sarah Berg.

### 3.1.1 Webprosjekt II

Prosjektet i emnet *IMT2671 – Webprosjekt II* utforsket problemene presentert av oppdragsgiver i form av å utvikle en prototype på høyt nivå. Fokuset for prosjektet lå på selve utviklingen, fremfor design og innhenting av innsikt, basert på oppdragsgivers ønske. Samtidig var det nødvendig å innhente noe innsikt for å fastslå behov for funksjonalitet. Resultatene som er relevante for videre arbeid oppsummeres som følger:

- Brukergrupper
- Løsning i form av høynivå-prototype
  - Marketplace
  - Dashboard

#### 3.1.1.1 Brukergrupper

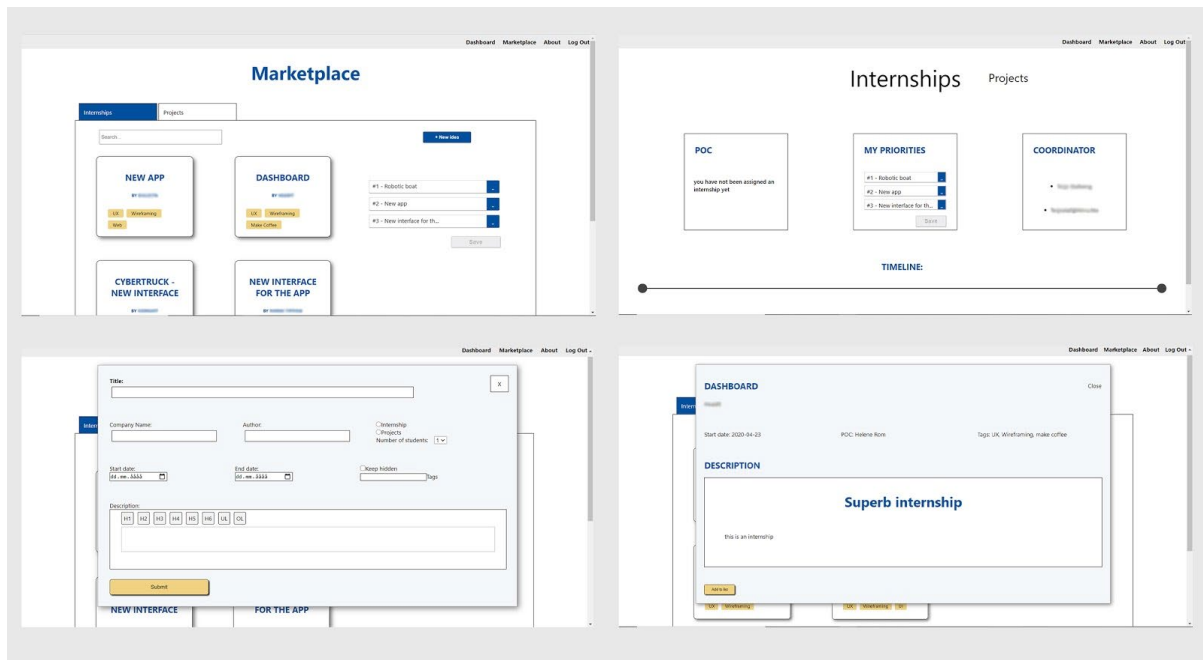
Innsiktsarbeidet identifiserte tre brukergrupper og tilhørende brukermål som stod sentralt gjennom hele prosessen. Brukergruppene ble delt i tre: studenter, administrative brukere og bedrifter, som vist sammen med deres brukermål i Figur 5. Det vil si studenter ved interaksjonsdesign og webutvikling, administrative brukere som emneansvarlig (oppdragsgiver) og andre relaterte fagpersoner, og bedrifter som er eller som ønsker å bli involvert med NTNU (Berg, Kvistad, Rogne, 2020b). Det er viktig å få frem at oppdragsgiver i dette prosjektet også er emneansvarlig i praksisemnet, og derfor er en del av brukergruppen administrative brukere.



Figur 5: Brukergruppene av løsningen. Tegnet av Sarah Berg av ikoner fra flaticon.com.

### 3.1.1.2 Løsning i form av høynivå-prototype

Prototypen ble utviklet med rammeverket React for frontend, samt PHP med MySQL-spørringer som backend, og Amazon Web Services (AWS) ble brukt til å leie infrastruktur, noe som var utfordrende. Figur 6 viser et utvalg skjermdumper fra prototypen. Løsningen baserte seg på en inndeling av konseptene «Marketplace» og «Dashboard», i tillegg til en tidslinje. Marketplace var «ansiktet utad» for løsningen, og hensikten var å vise alle tilgjengelige utlyste praksisplasser og bachelorprosjekter, samt inspirere bedrifter til å delta. Navnet bygget på konvensjoner om kjøp og salg, og navnet dashboard bygget på etablerte konvensjoner om et kontrollpanel. Hver bruker hadde sitt eget dashboard, som også varierte noe avhengig av rolle, det vil si om brukeren var student, administrator eller en bedrift. Hensikten med dashboard var å samle den viktigste informasjonen på ett sted. I tillegg inneholdt dashboard en tidslinje som viste punkter med datoer som var viktige for brukerne å merke seg.



Figur 6: Fra venstre: Marketplace, dashboard, ny utlysning og aktiv utlysning for studenter fra hi-fi prototype fra Webprosjekt II.

Resultatet av Webprosjekt II var et semi-funksjonelt system som fokuserte på prosessen for en aktuell praksisperiode, og var mest tilpasset oppdragsgivers problemområder (Berg, Kvistad, Rogne, 2020b). Prototypen manglet et godt design, samt flere vesentlige funksjonaliteter, noe som skapte et behov for mer omfattende innsiktsarbeid.

### 3.1.2 Prosjektutvikling i kreativ virksomhet

Det var hovedsakelig i emnet *IMT3007 – Prosjektutvikling i kreativ virksomhet* at prosjektgruppen avdekket problemer og fant ulike potensielle løsninger for oppdragsgivers problemer. Følgende problemstilling ble lagt til grunn for innsiktsarbeidet: «Hvordan kan en digital plattform bidra til å skape et internt bedriftsnettverk som har nytteverdi for både universitet, og deltakende studenter og bedrifter?» (Berg, Kvistad, og Rogne, 2020a, s. 8). Den første diamanten i double diamond-modellen, altså å utforske og å definere problemet, ble anvendt for å besvare problemstillingen.

Innsiktsarbeidet viste blant annet at mange studenter var villige til å reise eller bo andre steder i løpet av praksisperioden for å i gjengjeld få en god bedrift på CV-en. I tillegg ønsket flere å ha praksis i byer de ønsket å flytte til, allerede har bekjenskaper i, eller tilbake til sitt tidligere hjemsted (Berg, Kvistad, og Rogne, 2020a, s. 30). Basert på innsiktsarbeidet ble det konkludert med at fokuset burde ligge på å utvikle et robust nettverk som kan fungere i stor grad på egenhånd. Man fant at hvis oppdragsgiver kan tilby studentene et mangfold av bedrifter å velge mellom, til en viss grad uavhengig av lokasjon, vil studentene selv kunne prioritere hva som er viktigst når de velger bedrifter,

og føle på større grad av selvstendighet og ikke minst ansvar for egen fremtid (Berg, Kvistad, og Rogne, 2020a, s. 43).

Innsiktsarbeidet fokuserte i hovedsak på praksisemnet for å kunne svare på problemstillingen, men det ble også undersøkt om det var mulig å tilpasse den samme løsningen til gjennomføring av bacheloremner for Institutt for design, da det var ønskelig fra oppdragsgiver. Innsiktsarbeidet viste at prosessen for bachelor er tilnærmet lik prosessen for praksis. Man så at løsninger på mange av problemene i sammenheng med praksisemnet også kunne løse flere av problemene knyttet til bacheloremnet (Berg, Kvistad, og Rogne, 2020a, s. 8). Relevante resultater fra dette prosjektet oppsummeres som følger:

- Personas og scenarier
- Konkurrentanalyse
- Funksjonelle og ikke-funksjonelle krav
  - Overordnet
  - Profilsider
  - Autonomi
  - Automatisert e-post

### 3.1.2.1 Personas og scenarier

En persona ble laget til hver av brukergruppene. Hver persona baserte seg på innsikt innhentet gjennom kvalitative intervjuer med de ulike brukergruppene. Figur 7 viser en sammenstilling av de tre personaene (Berg, Kvistad, Rogne, 2020a).





Figur 7: Sammenstilling av personasene fra IMT3007 – Prosjektutvikling i kreativ virksomhet.

I tillegg ble det laget scenarier til personasene, to for student, ett for administrator, og ett for bedrift. Scenariene tok for seg hva personaen ønsket å oppnå ved å ta i bruk løsningen. For student ble det laget to scenarier da det ene omhandlet praksisemnet, mens det andre omhandlet bacheloremnet. Scenariene vises i Tabell 3 (Berg, Kvistad, Røgne, 2020a):

Student
Tiril har funnet tre bedrifter gjennom universitets nettverk og ønsker å søke på disse. Hun bruker løsningen til å skrive to av tre søknader, da hun ikke har tid til å skrive alle tre på en gang. Hun må derfor også sjekke når fristen for å søke er, slik at hun kan få skrevet den siste i tide.
Ved alle de seneste prosjektarbeidene har Tiril jobbet i samme gruppe, da denne gruppen mener de fungerer godt sammen, og alle er fornøyde med hverandres innsats. Gruppen har derfor bestemt seg for å skrive bachelor sammen også, og Tiril bruker derfor løsningen til å melde alle medlemmene opp i samme gruppe.
Administrator

Det er på tide å finne relevante praksisplasser til studentene. Mats trenger nå å finne flere bedrifter som vil være med i år. Mats bruker systemet for å se hvilke bedrifter som var involvert i fjor.

Bedrift

Knut har hørt at NTNU har studenter som skal ut i praksis, og tenker at det hadde vært nyttig med et samarbeid med NTNU i Gjøvik. Han vil bruke løsningen til å opprette dette samarbeidet ved å tilby en praksisplass og et prosjekt til bacheloroppgave.

Tabell 3: Scenarier for de tre brukergruppene (Berg, Kvistad, Rogne, 2020a).

### 3.1.2.2 Konkurrentanalyse

Prosjektet undersøkte og analyserte lignende løsninger som allerede er tilgjengelig på markedet. Det ble funnet tre løsninger som kunne sammenlignes med hva prosjektet skulle være: LinkedIn, NTNU Bridge og Arbeidslivsportalen, hvor fellesnevneren er at alle bygger broer mellom student og arbeidslivet (Berg, Kvistad, Rogne, 2020a, s. 27). NTNU Bridge og Arbeidslivsportalen forsøker å løse mange av de samme problemene som prosjektgruppen, da de legger til rette for håndtering av ulike former for praksis. Likevel, basert på tilgjengelig informasjon, mangler de tre løsningene vesentlige funksjonaliteter for håndtering av praksis fra et administrativt perspektiv, som for eksempel å få oversikt over hvilke studenter som er i kontakt med hvilke bedrifter (Berg, Kvistad, Rogne, 2020b, s. 25–29). Det vil si at ingen av løsningene tilstrekkelig dekker oppdragsgivers opplevde problemer, men at alle inneholder nyttig funksjonalitet som bidrar til inspirasjon inn i en ny løsning.

### 3.1.2.3 Funksjonelle og ikke-funksjonelle krav

Innsiktsarbeidet førte til at funksjonelle og ikke-funksjonelle krav ble fremstilt. De viktigste vil bli gjengitt her, da det legger grunnlaget for mye av funksjonalitet som må tas med videre. Noen av kravene var overordnet og som dermed gjaldt for hele løsningen:

- Nettsiden skal være på norsk
- Marketplace og dashboard må endres. Sidene samsvarte ikke med brukernes mentale modeller, og var derfor ikke forståelig. «Hele konseptet med disse to sidene må derfor revurderes og designes på nytt» (Berg, Kvistad, Rogne, 2020a, s. 39).

Profilsider har som formål å synliggjøre studenter og bedrifter i systemet. For studenter er det spesielt viktig å være synlig i forbindelse med søknad på praksis. For bedrifter er det et større behov for å være synlig i etterkant også, da de har et mer langsiktig mål om å være synlige fremtidige arbeidstakere. I tillegg er det gunstig med en profilside for administratorer for lagring av historisk data. Alle brukergruppene har ulike krav til hva profilsiden må inneholde (Berg, Kvistad, Rogne, 2020a, s. 37–39):

- Administrators profil er ikke synliggjort for de andre brukergruppene, og skal inneholde kontaktinformasjon og tilføye historisk oversikt over de ansatte og deres forbindelser med andre brukergrupper.
- Studentene skal presentere oppsummering av seg selv, slik at bedriftene kan få et inntrykk av dem. Informasjon bør være relevant til praksis og utdanningen studenten tar:
  - Hvilke ferdigheter studenten har
  - Hva vedkommende liker å jobbe med
  - Hva slags arbeidsoppgaver studenten ønsker å jobbe med
  - Relevante lenker til for eksempel portefølje, LinkedIn og GitHub
  - Det skal oppfordres til å laste opp profilbilde, men dog være frivillig.
- Bedriftsprofil skal inneholde generell informasjon, hovedfagfelt, logo, hvilken kompetanse de har og hva de ser etter av praksisstudenter, som vil være synlig for studentene. I tillegg skal bedriftens egen historikk over involvering med NTNU være synlig for bedriften selv.

Innsiktsarbeidet viser til at studentenes og bedriftenes autonomi står sentralt for å lette arbeidsbelastningen for oppdragsgiver, altså administrator i systemet (Berg, Kvistad, og Rogne, 2020a, s. 37), og burde derfor inngå som en viktig vurdering når informasjonsarkitekturen av en løsning skisseres. I tillegg vil autonomi for studenter og bedrifter i søknads- og ansettelsesprosessen gjøre det mulig å sløyfe steget hvor administrator må sette sammen student og bedrift basert på prioriteringer (Berg, Kvistad, og Rogne, 2020a, s. 40).

En digital løsning vil gjøre det mulig med automatiserte e-poster basert på hendelser eller endringer i systemet. Som innsiktsarbeidet avdekket vil automatiserte e-poster kunne lette på arbeidsmengden for administrator ved å automatisk sende ut påminnelser i forbindelse med frister (Berg, Kvistad, Rogne, 2020a, s. 43). Denne type e-poster kan også være hendelsesdrevet, for eksempel at administrator får tilsendt en e-post når en bedrift har foreslått en ny praksisplass. Automatiserte e-poster åpner også opp for å sende ut en e-post til bedrifter som tidligere har samarbeidet med Institutt for Design om at en ny praksisperiode nærmer seg, skulle de ønske å fortsette samarbeidet (Berg, Kvistad, Rogne, 2020a, s. 43).

Funnene og resultatene fra innsiktsarbeidet blir lagt til grunn for å videre utforme og utvikle løsningen gjennom faktisk implementering, samt testing og evaluering.

## 3.2 Teori

Dette kapitlet redegjør for teorien som er anvendt i prosjektet. Teorien vil bli diskutert i sammenheng med prosjektet i kapittel 7 Diskusjon.

### 3.2.1 «Mobil først»

I januar 2021, registrerte Datareportal (2021) i sin årlige rapport om global digital oversikt, hele 5.22 milliarder unike mobilbrukere på verdensbasis, noe som tilsier 66.6% av verdens befolkning. Denne målingen om unike mobilbrukere har en vekst på 1.8% hvert år. I tillegg er det flere og flere som bytter vekk eldre mobiltelefoner til fordel for mer moderne smarttelefoner, noe som vises ved at det gjennomsnittlig er over én million nye smarttelefoner i bruk hver dag. Det tilsier en årlig økning på 7% (Datareportal, 2021). Allerede i 2016 steg markedsandelen på mobilen over markedsandelen for datamaskin (Statcounter, u.å.). Mobiltelefonen øker stadig i popularitet og har kommet for å bli.

2016 var også det året da Google Search begynte å endre sin framgangsmåte fra rangeringssystemer som prioriterte desktop, til indeksering av mobil først (Phan, 2016). Det vil si at Google rangerer relevansen av innholdet på en nettside basert på hva mobilversjonen inneholder i sine Googlesøk. Fra 01. juli 2019 er indeksering av «mobil først» standard for Google Search på alle nye nettsider (Google Search Central, u.å.). Med andre ord er nå en «mobil først» tilnærming blitt svært viktig i dagens samfunn hvor mobiltelefonen stiller høyt. Samtidig viser Doantam Phan (2016), produktsjef for Google i 2016, at nettsider som er responsive, altså at designet tilpasser seg skjermstørrelsen (se Figur 8), og som har samme innhold samt lik struktur både på desktop og mobil er godt egnet for det nye skiftet innen søkemotoroptimalisering [SEO] i Google. I forbindelse med en «mobil først» tilnærming er med andre ord et responsivt design helt nødvendig både for brukeropplevelsen, men også for SEO.



Figur 8: Responsivt design fra desktop til mobil. Illustrert av Mari Rogne.

Mobilen har kommet for å bli, og er svært sentral i hverdagen. Viktig innhold på en nettside ment for en mobil skjerm bør plasseres innen tommelfingerens rekkevidde, da dette tilrettelegger for tilgjengelighet når mobilen blir brukt med kun én hand (McGough, 2014). Det er viktig å tilrettelegge for alle brukere som vil besøke løsningen på mobil ved å sørge for et responsivt og tilgjengelig design.

### 3.2.2 Brukersentrert design

Det er en tilnærmet konsensus blant designere innenfor interaksjonsdesign om hvordan man utformer et godt design. Ifølge Tone Nordbø (2017, s. 2), forfatter av boken *Introduksjon til interaksjonsdesign*, handler interaksjonsdesign om å gjøre løsninger brukervennlig. Dette kan man oppnå gjennom brukersentrert design som handler om å sette brukeren i fokus gjennom hele designprosessen. Ifølge Don Norman (2013, s. 9), som blant annet er forfatter av boken *The Design of Everyday Things*, er brukersentrert design en prosess som søker etter å sette fokus på brukerens behov og evner. Nordbø (2017, s. 31) formidler det samme, og legger til at prosessen krever at man kontinuerlig involverer brukeren. Jesse James Garrett (2011, s. 17) viser til at brukersentrert design lager engasjerende og effektive brukeropplevelser. Dette viser konsensusen om at utforming av et godt design er avhengig av å inkludere brukerne gjennom hele prosessen.

Med dette ønsker prosjektgruppen å stille følgende forskningsspørsmål: *hvordan utforme og utvikle en god brukeropplevelse gjennom prinsipper om design og universell utforming?* For å forsøke å svare på spørsmålet anser man WCAG, heuristikker og designprinsipper, og dark mode som relevant teori.

### 3.2.2.1 WCAG

Definisjonen av universell utforming er ifølge FNs konvensjon om rettigheter for personer med nedsatt funksjonsevne (Forente Nasjoner, 2013, s. 9):

Med «universell utforming» menes utforming av produkter, omgivelser, programmer og tjenester på en slik måte at de kan brukes av alle mennesker, i så stor utstrekning som mulig, uten behov for tilpassing og en spesiell utforming. «Universell utforming» skal ikke utelukke hjelpemidler for bestemte grupper av mennesker med nedsatt funksjonsevne når det er behov for det.

Med andre ord er universell utforming viktig for å forsikre at et produkt kan brukes av alle, uavhengig av evne. Universell utforming vil uansett gagne alle som bruker et produkt under omstendigheter som ikke er optimale, for eksempel direkte sollys eller et støyfullt miljø. I tillegg er universell utforming av IKT lovpålagt etter §18 i likestillings- og diskrimineringsloven (2018), samt etter §2 (1) i forskrift om universell utforming av IKT-løsninger (2013). Fra 1. januar 2021 *skal* alle nettsted, både nye og eksisterende, være universelt utformet, jf. forskriftens §11 (2). Forskriften krever at nettbaserte løsninger skal være i samsvar med Web Content Accessibility Guidelines (WCAG) 2.0, altså retningslinjer for tilgjengelighet på nett, jf. forskriftens §4 (1). WCAG er altså universell utforming på nett.

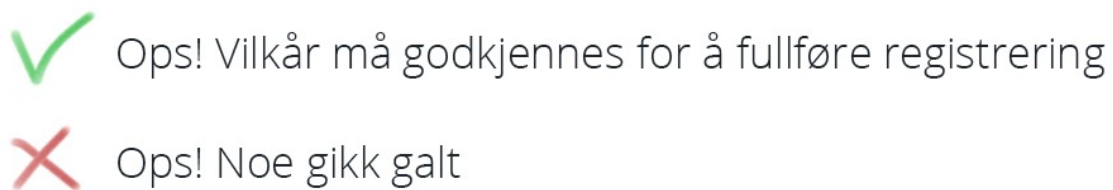
I 2020 ble WCAG 2.2 introdusert (World Wide Web Consortium [W3C], 2020), og er den versjonen prosjektet forholder seg til, selv om den foreløpig er et utkast under utvikling. Denne versjonen inkluderer alle de samme testbare kriteriene som forgjengerne 2.1 og 2.0, i tillegg til noen nye. Med andre ord: er man i samsvar med 2.2 er man automatisk også i samsvar med 2.1 og 2.0. WCAG 2.2 inneholder fire prinsipper, med sammenlagt 13 retningslinjer, som alle har flere testbare kriterier (W3C, u.å.). Disse kriteriene omhandler blant annet bruk av farge, alternativer til tekst, og tastaturnavigering. For å være i samsvar med loven, men også for å inkludere alle brukere uavhengig av evne samt nå ut til flest mulig, må man forholde seg til WCAG sine retningslinjer.

### 3.2.2.2 Heuristikker og designprinsipper

En heuristikk er en tommelfingerregel som skal hjelpe til med å sikre et brukervennlig design. I 1994 utviklet Jacob Nielsen et sett med ti heuristikker. Disse omhandler ulike aspekter ved design, blant annet når det gjelder estetikk, hindring av feil, og tilbakemeldinger (Nielsen, 2020). Nielsen (2020) viser eksempelvis til at et konsistent design som følger konvensjoner unngår å overbelaste brukernes kognitive belastning og gir dermed en bedre brukeropplevelse. Designprinsipper er regler for hva som utgjør et godt design (Nordbø, 2017, s. 37). Det finnes allerede veletablerte designprinsipper, for

eksempel skrev Don Norman (2013) om et sett designprinsipp i sin bok *The Design of Everyday Things*. Et eksempel på et av hans designprinsipp er også tilbakemeldinger.

Ifølge Norman (2013, s. 72) er det viktige med tilbakemeldinger at brukeren får gode, fortløpende, beskjeder på hva som foregår når en handling har blitt utført, men også beskjeder om tilstanden til systemet. Nielsen (2020) støtter Norman, og mener også at tilbakemeldingene må komme innenfor et fornuftig tidsrom etter at en handling er utført. Farene ved å ikke inkludere tilbakemeldinger i et system, vil etter Don Norman (2013, s. 23–24) sin mening føre til at brukerne gir opp. Han mener også at ikke-forklarende tilbakemeldinger kan være verre enn å ikke få tilbakemeldinger i det hele tatt (se Figur 9), da brukeren kan bli forvirret og frustrert, noe som igjen kan føre til at brukeren begynner å ignorere tilbakemeldingene (Norman, 2013, s. 24).

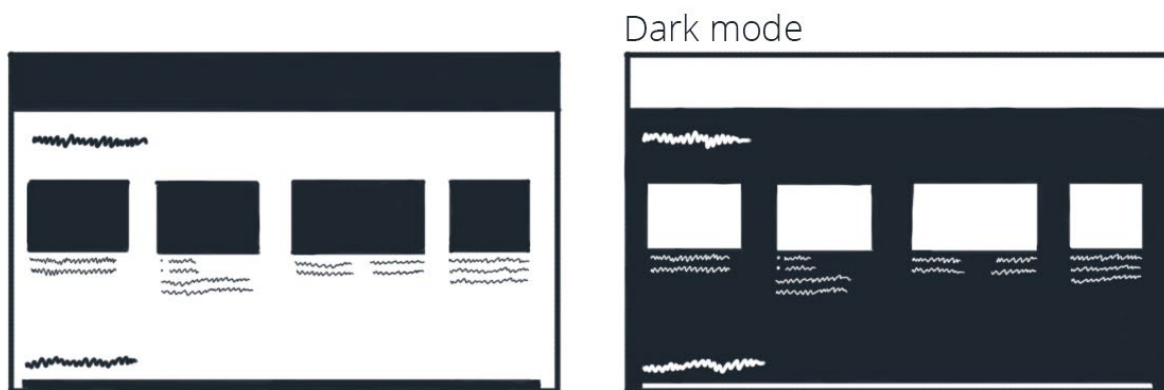


Figur 9: Ikke-forklarende tilbakemelding vs. forklarende tilbakemelding.

Tilbakemeldinger er bare én av flere heuristikker og designprinsipper, som er gode regler for å sikre et brukervennlig design, som igjen utgjør en forskjell for universell utforming.

### 3.2.2.3 Dark mode

«Dark mode» er et design som inverterer det klassiske designet «light mode» som vanligvis har mørk skrift på lys bakgrunn, og som dermed baserer seg på lys skrift på mørk bakgrunn (se Figur 10). De siste årene har popularitet rundt dark mode på digitale skjermer sett ut til å vokse etter at store aktører som YouTube og Apple begynte å tillate brukerne å selv velge om de ønsket å bruke en lys eller mørk modus. YouTube introduserte brukerne sine for dark mode i 2017 (Perez, 2017), og Apple introduserte dette til iPhone i 2019 (Dolcourt, 2019). Valget om å benytte seg av dark mode kan muligens være en personlig estetisk preferanse for noen, men det er sannsynligvis også et bevisst valg for de fleste da denne modusen har flere gode fordeler.



Figur 10: Eksempel på lys og mørk modus. Illustrert av Sarah Berg.

Ifølge Apple hjelper dark mode brukerne med å holde fokus fordi de viktige elementene skiller seg ut fra de mørke elementene, samt at det mørke designet fører til mindre anstrengelse for øynene (Apple, 2020). Google mener dark mode vil forbedre synlighet for brukere med nedsatt syn eller som er sensitive til skarpt lys, samtidig som det kan øke batterilevetiden, dog dette vil variere basert på skjermtypen (Android Developers, 2021).

Samtidig har modusen mottatt noe kritikk, for eksempel skriver Peter Kostadinov (2019) i sin artikkel *The pros and cons of Dark Mode: Here's when to use it and why* at dark mode kan senke lese kvaliteten for nærsynte brukere, da hvit tekst på mørk bakgrunn kan fremstå som mindre klar enn svart tekst på en lys bakgrunn. Heydon Pickering (2017) skriver derimot tvert om, at dark mode, eller «natt modus» som de også kaller det, er veldig «easy on the eyes» når man leser i mørket. Det mest interessante er dog at Pickering (2017) mener dark mode reduserer sjansene for migrene. Dette viser at både dark mode og light mode har sine fordeler og ulemper; hvilken modus som er best kan virke å være individuelt basert på behov og evner.

### 3.2.3 Karbonutslipp og internett

Bærekraft og klimaendringer står som noe av det mest aktuelle på dagsordenen. Ifølge artikkel 3 i Parisavtalen (2016) utarbeidet av Forente Nasjoner er alle delaktige land forpliktet til å delta med mål om å nå målsetningene etter lovens artikkel 2. Når man tenker karbonutslipp er det fort å tenke fossilt brennstoff i form av biler og fly og overforbruk. Det er enkelt å tenke at det er mer miljøvennlig å bruke det digitale fremfor det fysiske, som for eksempel et digitalt word-dokument foran en hel haug med papir. Derimot kan det antas at «mannen i gata» ikke er klar over hvor store mengder av karbonutslippene som er knyttet til internett.



Det er bedre for miljøet å sende en felles e-post til alle ens kolleger fremfor å kopiere opp utallige eksemplarer av et dokument, men også dette har dessverre noen konsekvenser for både klima og miljø (Atea, 2021). I følge Atea (2021), et ledende IT-selskap i Norden, kan man spare like mye strøm som trengs for å lade en million smarttelefoner, dersom alle nordmenn sletter så lite som 1 til 2 e-poster hver. All data, som e-poster, lagres på en server i for eksempel en serverpark. Disse må være aktive hele døgnet, hele året, noe som produserer mye varme og krever mye energi for å kjøles ned (Atea, 2021).

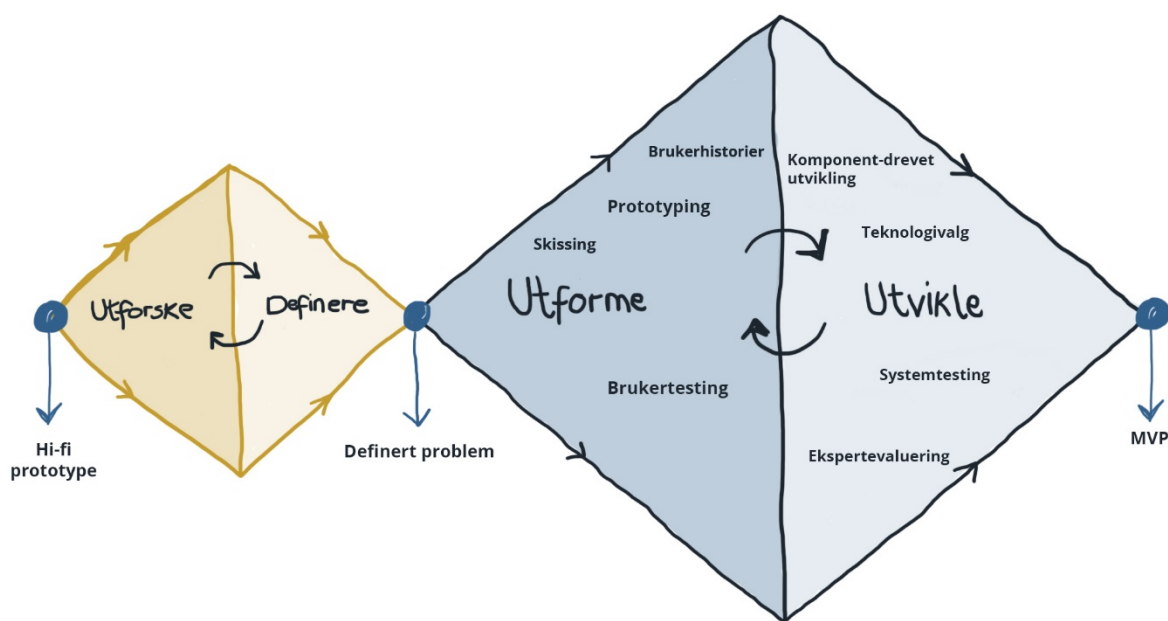
Innen 2040 vil karbonutslipp i forbindelse med enheter koblet til internett utgjøre ca. 14% av globale utslipp (Vidal, 2017). Dette betyr at alle utviklere og personer som jobber med digitale nettbaserte løsninger har et kollektivt ansvar når det kommer til hva som publiseres og blir lagret. Eidar Grande Vollan (2019) beskriver utfordringene med unødvendige mengder data på Netlife sin fagblogg som følgende:

Utfordringen er at **internett ikke er rigget for å slette innhold i samme takt som det blir produsert nytt innhold**. Det er som den rotete kjelleren. Etterhvert som innholdet blir mindre relevant blir det stappet inn der, glemt og samler støv til vi endelig en dag innser at vi ikke bruker det lenger og kaster innholdet. Det er bare det at vi altfor sjelden har ryddesjau på nett (Vollan, 2019).

Vollans utsagn får støtte av Gerry McGovern (2019) som sier at 90% av digital data aldri bli brukt igjen når det er gått 90 dager fra dataene ble lagret. Det vil si at kun 10% av data tilgjengelig på internett faktisk blir brukt. Når det kommer til mobile applikasjoner viser det seg at 62% av brukerne vil bruke appen mindre enn 11 ganger (Perez, 2016). Andrew Chen (u.å.) viser til at tre dager etter installasjon vil så mye som 77% av brukerne aldri åpne appen igjen. Dette er store tall som alle indikerer det samme: store mengder data blir liggende på store serverparker, altså lagringsplasser for digital data, unødvendig. For å opprettholde det kollektive ansvaret prosjektgruppen har som utviklere stiller man derfor et forskningsspørsmål ved *hvilke tiltak prosjektgruppen kan gjøre for å unngå overflødig karbonutslipp relatert til internett*.

## 4 Utforme

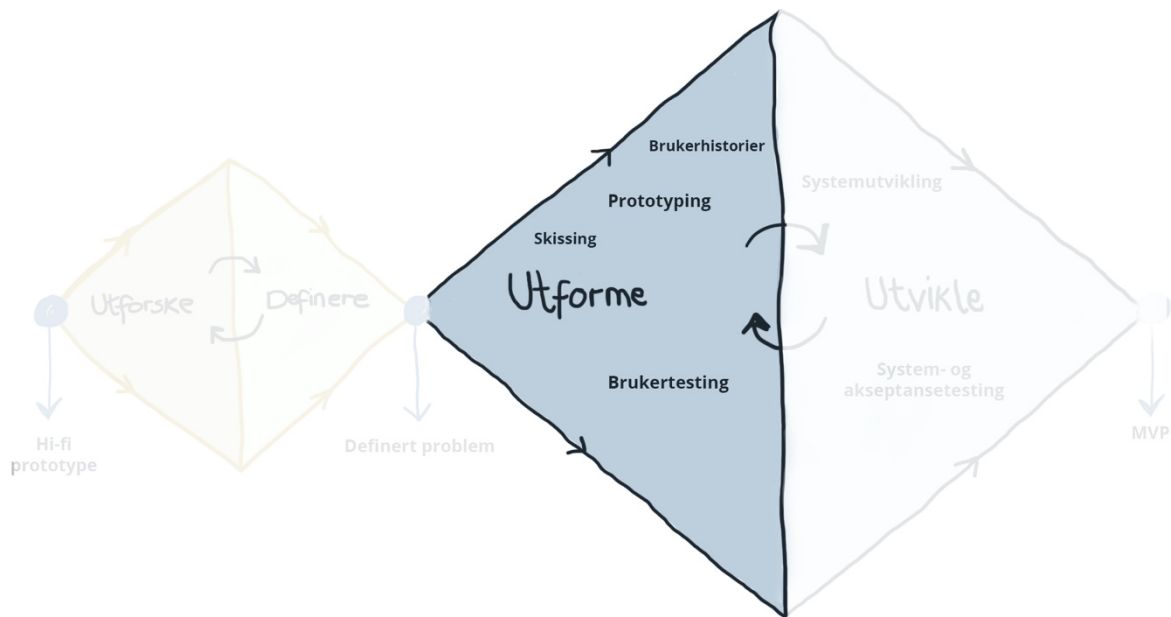
Som nevnt i kapittel 3 er arbeidet i dette prosjektet organisert gjennom double diamond-modellen, som er en tilpasningsdyktig modell hvor de fire fasene kan omhandle ulike ting avhengig av hva slags prosjekt modellen anvendes i, og hva formålet med prosjektet er (se Figur 11). Som oftest er de kalt for henholdsvis utforske, definere, utvikle og levere (Design Council, u.å.). Hvis målet for eksempel er å levere et ferdig utviklet produkt, kan den siste fasen inneholde blant annet opplæring av brukere, lansering og oppfølging. Da dette prosjektet derimot ikke skal lansere et ferdigstilt produkt, men utvikle en MVP, har de forskjellige fasene blitt tilpasset dette.



Figur 11: Double diamond prosess i prosjektet. Illustrert av Sarah Berg.

I dette prosjektet er de fire fasene navngitt *utforske*, *definere*, *utforme* og *utvikle*. Figur 11 viser hele modellen, hvor den første diamanten representerer bakgrunnen for prosjektet, som utdypet i kapittel 3.1. Den andre diamanten illustreres som større, da det er denne som er anvendt i dette prosjektet. For å utvikle en løsning er arbeidet delt opp i to faser, hvor den første fasen tar for seg utforming, og den andre fasen fokuserer på utvikling og implementering. Figur 11 viser også hvilke metoder som har blitt arbeidet med gjennom de to fasene. Den første fasen av prosjektet har som mål å **utforme** brukergrensesnittet i form av visualisering gjennom blant annet prototyping, og bygger på resultater og funn fra tidligere prosjektarbeid (se kapittel 3.1).

Videre i dette kapitlet følger anvendte metoder og resultater fra første fase, utforme (se Figur 12 nedenfor). Den andre fasen, utviklingsfasen, vil bli redegjort for i kapittel 5.



Figur 12: Utforme - første fase av double diamond som er fokus for kapittel 4.

## 4.1 Metoder i utformingsfasen

Et utvalg metoder ble tatt i bruk i utformingsfasen. Dette kapittelet vil utdype hensikten med metodene og hvordan de er anvendt i dette prosjektet.

### 4.1.1 Brukerhistorier

Brukerhistorier er beskrivelser av hva en brukerrolle har behov for å utføre ved hjelp av løsningen som skal lages, formulert på én setning. Hver brukerhistorie skal inneholde hvem som skal bruke løsningen, hva de trenger av løsningen og hvorfor de trenger det (Nordbø, 2017, s.109). Historiene baserer seg på innsikt, og er en nyttig metode for å definere krav til tjenester. Man kan dele brukerhistorien i tre deler (Entur, u.å):

1. Selve brukerhistorien (verdiforslaget), som følger et fast format for et raskt overblikk av hva historien løser. Dette prosjektet følger formuleringen fra Entur (u.å):  
Som en **[bruker]** ønsker jeg en **[funksjon]** slik at jeg **[oppnår en verdi]**.
2. Løsningsbeskrivelse som er en enkel, men spesifikk oppsummering av hvordan oppgaven skal løses. Løsningsbeskrivelse skrives etter et løst format.
3. Akseptanskriterier, som også følger et fast format. Disse kriteriene skal skrives på en slik måte at man enten kan bekrefte eller avkrefte hvorvidt kriteriet er oppfylt, og dermed forteller

noe om hva som skal lages og testes.

Brukerhistoriene i dette prosjektet baserer seg på resultater og funn fra tidligere prosjektarbeid (se kapittel 3.1), ved å ta utgangspunkt i brukermål og funksjonelle krav. Det ble totalt laget tre brukerhistorier, en til hver brukergruppe, som brukes som et grunnlag for skissering i denne fasen. I tillegg vil brukerhistoriene også bli brukt for systemtest i utviklingsfasen av prosjektet, se kapittel 5.3.2, da de legger et godt grunnlag for å lage et testscript til når løsningen skal testes (Nordbø, 2017, s. 111).

### 4.1.2 Skisser

Skissing er en god metode for å raskt visualisere og vise frem tanker og ideer. Å skissere tillater å generere en overflod av ideer, samt tenke mer kreativt, lettere diskutere og dele ideer med andre, og kunne jobbe videre med gode ideer som har potensiale (Greenberg, *et al*, 2012, s. 4). Prosjektet er inspirert av prototypen i Webprosjekt II, og tok med seg både funksjonalitet og designelementer som viste seg å være gode utgangspunkt videre inn i skissering av en ny prototype. Samtidig ble det forsøkt å tenke nytt i henhold til aspekter som ikke var optimale.

Skissene i dette prosjektet hadde som formål å skissere opp et mer brukervennlig design. Via skissering på papir ble det gjennomført tre iterasjoner med utkast til ideer. For hver iterasjon skisserte hvert medlem i prosjektgruppen ulike skisser på et ark i A4-størrelse. Deretter ble alle skissene samlet, diskutert, og videre arbeidet med, hvor man endte med et utvalg som utgangspunkt til prototyping.

### 4.1.3 Prototyping

Hensikten med skisser er å raskt illustrere ideer, mens hensikten med en prototype å lage fremstilling av hvordan produktet som skal bli produsert fungerer (Nordbø, 2017, s. 147). Hovedforskjellen er at skisser er overfladiske illustrasjoner av enkelte elementer, mens prototyper er en detaljert helhetlig sammensetning som er interaktiv, og som blir brukt til brukertesting (Nice, 2019). Prototyping er, ifølge Nordbø (2017, s. 154), en iterativ prosess, hvor en ny iterasjon har som hensikt å forbedre eventuelle problemer med for eksempel brukervennlighet og informasjonsstruktur. Man kan enten forenkle grensesnittet ved å fjerne unødvendige elementer, eller berike det i form av å legge til innhold eller visuelle elementer (Nordbø, 2017, s. 154). I dette prosjektet har begge deler blitt gjort. Prototyping gir altså mulighet til å teste ut ideer og funksjonalitet på representanter for målgruppene (Nordbø, 2017, s. 147), og det skilles mellom to typer prototyper; lavnivå-prototyper (low-fidelity) og høynivå-prototyper (high-fidelity).

Formålet med å lage en prototype var å teste ut funksjonalitet. Da en høynivå-prototype er nyttig for å kunne teste ut funksjonalitet i en større skala (Nordbø, 2017, s. 149), gikk prosessen direkte fra skisser til en høynivå-prototype. En høynivå-prototype blir laget med materialer som ligner på det endelige produktet (Nordbø, 2017, s. 149), noe som i dette prosjektet tilsvarer noe klikkbart på en digital skjerm. En høynivå-prototype *kan* lages med HTML, men ifølge Josh Doolan (2016) vil en slik prototype være tidkrevende dersom det oppstår feil i koden, samtidig som det generelt tar lenger tid å kode enn å prototype i et typisk prototypingsverktøy. Da prosjektet har begrenset med tid, ble det mindre tidkrevende verktøyet for prototyping Figma<sup>1</sup> brukt for prototyping. I tillegg muliggjør Figma å jobbe parallelt og se endringer i sanntid.

Prototypen for prosjektet ble utformet gjennom to iterasjoner av høynivå-prototyping. I tillegg til praksis tok prototypen høyde for å illustrere hvordan løsningen kan tilrettelegge for bachelorprosjekt, da man i tidligere arbeid (se kapittel 3.1) konkluderte med at løsninger på problemene i sammenheng med praksis også kunne løse mange av problemene i sammenheng med bachelor. Dette er dog ikke et fokus for løsningen, men snarere heller en måte å illustrere hvordan løsningen kan utvides til å inkludere bachelorprosjekt.

#### 4.1.4 Brukertestning

For å redegjøre om endringer fra en iterasjon til en annen faktisk har forbedret prototypen, er man avhengig av å teste endringene på ekte brukere. Målet med brukertestning er å vurdere brukbarheten av et produkt ved å avsløre hvilke deler av produktet som er problematiske for brukeren, og hvilke deler som fører til suksess (Cooper, *et al*, 2014, s. 57). Dersom brukertesten utføres på et produkt i løpet av designprosessen, altså en prototype av et produkt som ikke er lansert, vil hensikten med brukertesten være noe konkret, eksempelvis testing av navigasjon, konsept, eller funksjonalitet (Nordbø, 2017 s. 159–160). Gjennom brukertestning kan man også få svar på *hvorfor* brukerne gjør som de gjør når de samhandler med produktet (Nordbø, 2017, s. 158). For å fastslå hvor godt brukerne samhandlet med prototypen som ble laget i Figma for dette prosjektet, samt avdekke problematiske aspekter ved funksjonaliteten, var det derfor essensielt å gjennomføre brukertestning. Alle resultatene tilknyttet brukertestingen gjennomført i prosjektet finnes i vedlegg 4.

##### 4.1.4.1 Gjennomføring av brukertester

Det ble gjennomført to runder med brukertestning med oppgaver for de tre brukergruppene over en periode på to uker. Brukertestvettreglene av Eli Toftøy-Andersen og Jon Gunnar Wold (2011, s. 35)

---

<sup>1</sup> <https://www.figma.com/>

stod sentralt for gjennomføring. Disse reglene nevner blant annet at man skal vise respekt for testbrukerne, at det lønner seg å planlegge testen grundig på forhånd, og ikke minst ha en plan B i tilfelle noe ikke fungerer som planlagt. I forkant av brukertesting ble det utarbeidet en samtykkeerklæring som ble presentert for testbrukerne (se vedlegg 3). Samtykkeerklæringen redegjør for hva deltakelse i brukertesten innebærer, samt forsikrer om at det ikke blir lagret noen personidentifiserende data og at brukerne dermed forblir 100% anonyme.

Hver brukertest involverte alltid to medlemmer av prosjektgruppen, hvor en fungerte som testleder og fasilitator for å lede brukerne gjennom testen, og den andre som observatør til å ta notater og målinger. Observatøren siterte testbrukerne fremfor å skrive om, i tillegg til å notere nøyaktig hva de gjorde. Dette er viktig for at man i etterkant av brukertesten ikke trekker feil konklusjoner basert på antatte intensjoner (Toftøy-Andersen og Wold, 2011, s. 73). Brukertestene fokuserte på de faktiske observasjoner og målinger fremfor tilbakemeldinger fra testbrukerne, som anbefalt av Toftøy-Andersen og Wold (2011, s. 73).

Brukertesting foregikk enten digitalt eller fysisk avhengig av smittesituasjonen. En ulempe med å utføre digitale brukertester er at man går glipp av viktig ikke-verbal kommunikasjon som kroppsspråk. Derimot er det en fordel at brukeren får bruke sitt eget utstyr i sitt eget hjem, noe som kan bidra til at testen oppleves mindre kunstig (Toftøy-Andersen og Wold, 2011, s. 139). I Microsoft Teams tok brukerne kontroll over skjermen til fasilitator, for å selv klikke og navigere som de ville ved hjelp av fjernstyring. I noen tilfeller fungerte det ikke å la testbrukeren ta kontroll, og dermed ble plan B at testbrukerne instruerte fasilitator til å trykke hvor testbrukeren ville. Ved fysisk brukertesting ble brukerne tildelt engangshansker for å kunne navigere og klikke på egenhånd grunnet dagens krav til smittevern.

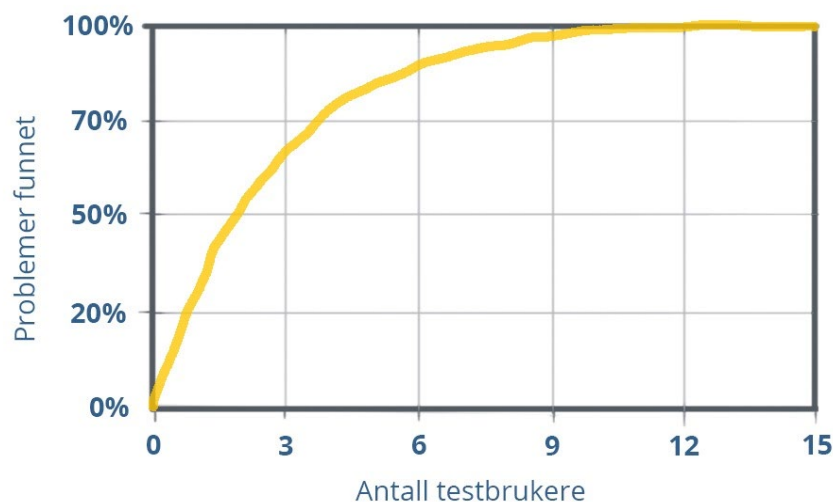
#### 4.1.4.2 Utvalg av testbrukere

Brukertesting ble utført på testbrukere som hadde fellestrekk med de tre brukergruppene. Hver testbruker fikk utdelt alle de åtte oppgavene, det vil si at hver testbruker testet løsningen som alle tre brukergruppene. Alder er ikke tatt i betraktning og heller ikke ført opp, da de tre forskjellige brukergruppene potensielt kan inneholde alle aldersgrupper. Ved første runde med brukertesting var fire av testpersonene kvinner, mens en var mann. Ved andre runde var to av testpersonene kvinner, mens tre var menn (se Tabell 4).

Runde	Antall kvinner	Antall menn
1	4	1
2	2	3

Tabell 4: Fordeling av kjønn per runde brukertesting

Prototypen ble testet på fem *nye* personer per runde med brukertesting. Utvalg av antall testbrukere per runde med brukertesting baserer seg på Jacob Nielsen (2000) sin artikkel *Why you only need to test with 5 users*. Nielsen (2000) argumenterer for at utvalg på fem personer er tilstrekkelig for å finne de fleste problemene et produkt har (se Figur 13).



Figur 13: Antall nødvendige brukertester (Nielsen, 2000). Illustrert av Sarah Berg.

Nielsen (2000) mener at et utvalg på mer enn fem personer på en runde med brukertesting vil føre til at man avdekker de samme problemene igjen og igjen. Derfor mener han det er bedre å teste på et lite antall mennesker før utviklingsgruppen gjennomfører en iterasjon på produktet, og gjennomfører så en ny runde med brukertester på fem nye personer (Nielsen, 2000). Nielsen får støtte av Toftøy-Andersen og Wold (2011, s. 62) som også skriver at de anbefaler å teste på færre personer i flere omganger, fremfor mange personer en gang.

#### 4.1.4.3 Brukertestens oppgaver

Oppgavene testbrukerne skulle løse tar utgangspunkt i scenariene for personasene (som beskrevet i kapittel 3.1.2.1) da Nordbø (2017, s. 164) viser til at det kan være lurt å formulere oppgaver mer som scenarier fremfor kommandoer for at det skal føles mer naturlige for brukeren. I tillegg ble det stilt åpne spørsmål underveis for å få et større innblikk i hvorfor brukerne valgte å gjøre det de gjorde, og hva de tenkte mens de gjorde det.

For brukergruppen studenter ble det laget fire oppgaver (Tabell 5). Formålet med oppgavene var å teste funksjonalitet for å søke på en praksisplass, men inneholdt også én oppgave om funksjonalitet for å opprette bachelorgruppe. Flest oppgaver ble gitt brukergruppen student da denne gruppen har en større samling av kjernefunksjonalitet for å sikre autonomi, som i sin tur vil minke arbeidsmengde for administrator (se kapittel 3.1.2.3). Oppgavene var som følger:

Oppgaver for brukergruppen student	
Nummer	Oppgave
1	Du har lyst til å lese mer om utlysningen til Atea AS, og fjerne den fra dine prioriteringer
2	Skriv en søknad til den prioriteringen som mangler en søknad
3	Dann en bachelorgruppe
4	Sjekk når fristen for å søke på praksisplasser er

*Tabell 5: Oppgaver for brukergruppen student ved brukertesting*

To oppgaver ble laget for brukergruppen bedrifter (Tabell 6). Fordi bedrifter har mindre kjernefunksjonalitet enn studenter, var det mulig å oppsummere det viktigste i to oppgaver:

Oppgaver for brukergruppen bedrift	
Nummer	Oppgave
5	Du har hørt at NTNU har studenter som skal ut i praksis, og tenker at det hadde vært nyttig med et samarbeid med NTNU i Gjøvik. Du vil bruke løsningen til å registrere deg og opprette dette samarbeidet ved å tilby en praksisplass.
6	Sjekk hvem som har søkt på utlysningen din og les søknaden til Jens Søre, og prioriter denne studenten.

*Tabell 6: Oppgaver for brukergruppen bedrift ved brukertesting*

For administratorer ble det laget to oppgaver (Tabell 7). Gjennom den nye løsningen er stegene for administratorer minimert, mye på grunn av autonomi for studenter og bedrifter, og kan derfor også summeres opp i to oppgaver:



Oppgaver for brukergruppen administrator	
Nummer	Oppgave
7	En ny bedrift har registrert seg i nettverket. Finn forespørselen og godkjenn den.
8	Det er på tide å finne relevante praksisplasser til studentene. Mats trenger nå å finne flere bedrifter som vil være med i år. Mats bruker systemet for å se hvilke bedrifter som var involvert i fjor.

Tabell 7: Oppgaver for brukergruppen administrator ved brukertesting

Til tross for at noen av oppgavene er formulert som kommandoer, ble oppgavene flettet mer inn i en historie til brukerne, slik at de forstod hva slags setting og kontekst løsningen brukes i. Dette ble gjort ved å gi testbrukerne navn og historier de skulle forholde seg til:

Ditt navn er Jens Søre. Du er student ved interaksjonsdesign ved NTNU og skal ut i praksis dette semesteret. **[Oppgave 1]**. Etter å ha prioritert tre bedrifter, har du skrevet to søknader, men mangler den siste. **[Oppgave 2]**. I tillegg til praksis skal du skrive bachelor dette semesteret, og må derfor opprette en bachelorgruppe. **[Oppgave 3]**. Før du avslutter tenker du at du vil gjøre noen endringer på søknaden du tidligere skrev, men har ikke tid til det akkurat nå. Du vil derfor sjekke når fristen for å skrive søknadene til praksis er. **[Oppgave 4]**.

Du heter Knut, og arbeider nå for en bedrift som lager apper og nettsider, og som har både interaksjonsdesignere, grafisk designere, og webutviklere på lønnslisten. Du vet at NTNU har et studieprogram for interaksjonsdesign, og ønsker å tilby en praksisplass til studenter ved det programmet. **[Oppgave 5]**. Etter å ha lyst ut en praksisplass venter du noen uker til studentene er ferdige med søknadsprosessen. **[Oppgave 6]**. Et par dager senere ser du at Jens Søre skal utføre praksisperioden sin i din bedrift.

Ditt navn er Mats Nilsen, og som ansatt ved NTNU er du emneansvarlig for emnet *Veiledet praksis i medie- og informatikkfag*. Du har et stort ansvar for å holde oversikt over bedrifter og studenter for å sikre en god opplevelse for alle ved gjennomføring av praksisemnet. Dette innebærer blant annet å sørge for at det er en større samling bedrifter tilgjengelig. Av og til har bedrifter fått nyss om systemet, og velger å registrere seg. **[Oppgave 7]**. I tillegg må du selv være aktiv for å rekruttere bedrifter. **[Oppgave 8]**.

#### 4.1.4.4 Målinger av brukertester

Ved gjennomføringen av brukertestene var det viktig å samle opp målbare resultat som kunne sammenlignes. For hver oppgave ble det ført opp tid brukeren brukte, antall feilklikk, og antall ganger brukeren spurte om hjelp. Disse føringene fra brukertesting blir fremstilt gjennom to forskjellige utregninger: grad av gjennomføring og tidsbasert effektivitet. Om en oppgave er mulig å gjennomføre spiller en stor rolle for begge utregningene. Gjennomføring av brukertestene viste at oppgavene var gjennomførbare; forespørsler om hjelp ansees derfor som mislykkede forsøk i utregningene for både begge. I tillegg ble problemer som oppstod for brukeren under testen ført opp i et skjema for risikovurdering som rangerte de i en alvorlighetsgrad fra lav til kritisk for å kunne fastslå hvilke problemer som har høyest prioritering for å løses (Nordbø, 2017, s. 172).

Grad av gjennomføring måler hvor effektiv løsningen er. Formel for utregning er hentet fra Justin Mifsud (u.å.) (se Figur 14). «Mengden gjennomførte oppgaver» tilsvarer i dette tilfellet hvor mange av brukerne som ikke trengte hjelp til den spesifikke oppgaven. «Totalt antall oppgaver foretatt» viser til hvor mange brukere som utførte den spesifikke oppgaven. Dersom 5 testbrukere utfører en oppgave, hvor 4 gjennomfører uten hjelp, vil utregningen bli 4 dividert på 5 multiplisert med 100%.

$$\text{Effektivitet} = \frac{\text{Antall gjennomførte oppgaver}}{\text{Totalt antall oppgaver foretatt}} \times 100\%$$

Figur 14: Formel for grad av gjennomføring (Mifsud, u.å.).

Grad av gjennomføring blir målt fra 0–100%. En studie som baserer seg på en analyse av 1100 oppgaver, gjennomført av Jeff Sauro (2011), en utdannet analytiker og forfatter, fant at den gjennomsnittlige fullføringsgraden ligger på 78%. Naturlig nok har kontekst en innvirkning på hvilken fullføringsgrad man burde sikte på. Om konsekvensene for oppgavesvikt er høye, for eksempel ved tap av penger eller i verste fall tap av liv, må man sikte på 100% (Sauro, 2011). Hvis konsekvensene er mindre, kan fullføringsgraden ligge noe lavere. Basert på dette ble fullføringsgraden i brukertesting av prototypen satt til å være minst 80% før man kunne gå videre til implementering.

Tidsbasert effektivitet måles i form av oppgavetid, altså hvor lang tid i sekunder testbrukeren bruker på å fullføre en oppgave. Formel for utregning er hentet fra Justin Mifsud (u.å.) (se Figur 15). Aktuelle måltall har som formål å gi uttrykk for effektivitet som en funksjon av tiden brukeren benytter for en gitt oppgave. Dette betyr følgelig at det er et direkte forhold mellom grad av

gjennomføring og tidsbasert effektivitet, og at tidsbasert effektivitet er brukerens grad av gjennomføring delt på brukerens brukte tid, som igjen er delt på antall brukere multiplisert med antall oppgaver (Shukairy, 2014). Dersom brukeren ber om hjelp og ikke gjennomfører oppgaven på egenhånd vil det bli ansett som et mislykket forsøk og verdien av gjennomføring tilsvare null (0). Dersom brukeren gjennomfører på egenhånd, vil verdien av gjennomføring tilsvare en (1).

$$\text{Tidsbasert effektivitet} = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR}$$

Figur 15: Formel for tidsbasert effektivitet (Mifsud, u.å.).

En risikovurdering er en kvalitativ analyse av brukertestene som omhandler å prioritere feil (Nordbø, 2017, s. 172). Problemene som oppstod ble ført inn i en tabell for risikovurdering med fire kategorier. De fire kategoriene baserer seg på en tabell fra Society for Technical Communications som gjengitt og oversatt til norsk i Toftøy-Andersen og Wold (2011, s. 152–153). Kategoriene rangeres fra kritisk til lav, hvor kritisk har høyest prioritet for å løses. Nordbø (2017, s. 172) forklarer de ulike kategoriene som følger:

- Kategorien «**kritisk**» oppstår dersom funksjonalitet mangler eller ikke fungerer som tenkt, og som gjør at oppgaven ikke er gjennomførbar.
- Kategorien «**høy**» vil ha problemer som gjør at brukeren ikke klarer å utføre oppgaven eller at brukeren ikke blir stoppet fra å utføre vesentlige feil.
- Kategorien «**medium**» tilsier problemer som ikke hindrer brukeren i å utføre oppgaven, men likevel skaper frustrasjon.
- Kategorien «**lav**» er av ikke-kritisk karakter, for eksempel visuelle problemer.

Problemene som ble avdekket i brukertesting av de to iterasjonene av prototypen ble risikovurdert i samsvar med disse kategoriene.

## 4.2 Resultat av utformingsfasen

I det foregående underkapittelet ble det redegjort for metoder og hvordan de ble anvendt i utformingsfasen. Dette kapittelet legger frem resultatene fra disse metodene.

## 4.2.1 Brukerhistorier

For å definere krav som løsningen må ta hensyn til, samt gjøre det enklere å se for seg hva brukerne skal kunne gjøre i løsningen, ble det laget brukerhistorier med utgangspunkt i brukergruppens brukermål (se kapittel 3.1.1.1) og funksjonelle krav (se kapittel 3.1.2.3). Hver brukerhistorie inneholder selve brukerhistorien, løsningsbeskrivelse og akseptansekriterier. Det ble laget en brukerhistorie for hver brukergruppe, som ble grunnlaget for skissering videre i denne fasen.

Brukergruppen student har som brukermål å finne eller foreslå en praksisplass. Brukerhistorien som vist i Figur 16 tar utgangspunkt i en søknadsprosess hvor studenten ønsker å bruke løsningen til å finne bedrifter fremfor å finne en bedrift på egenhånd.

**Brukerhistorie for student**  
**Som en** student ved institutt for design **ønsker jeg** å kunne få en oversikt over bedrifter som tilbyr praksis **slik at** jeg kan søke på disse og få erfaring med arbeidslivet.

**Løsningsbeskrivelse**  
Det er nødvendig å vise frem relevant informasjon studenten trenger for kunne gjøre en vurdering av bedriften hva gjelder egne preferanser og hva studenten selv mener er viktig i valg av praksisplass. Dette vil for eksempel være hvor bedriften holder til, og spesielt hva slags arbeid det er forventet at studenten skal ta del i.

**Akseptansekriterier**

(A) Gitt at jeg **ikke** har prioritert bedrifter, så skal jeg kunne:


- 1 Se en liste over utlysninger som tilbyr praksisplass til mitt institutt
- 2 Se relevant informasjon om aktuell utlysning og bedrift
- 3 Se hva jeg må gjøre for å søke

(B) Gitt at jeg **har** prioritert bedrifter, så skal jeg kunne:

- 1 Se hvem og hvor mange jeg har prioritert
  - a) Omprioritere
- 2 Se fristen for å skrive søknad til prioriteringene
  - a) Fristen skal være felles for alle praksisplasser

(C) Gitt at jeg **har** skrevet søknad til bedrifter, så skal jeg kunne:

- 1 Redigere søknadene
- 2 Se om det mangler søknad
- 3 Se at søknad er lagret



Figur 16: Brukerhistorie for brukergruppen student.

Brukermålene for brukergruppen bedrift er å drive relasjonsbygging med NTNU og studenter, og å utforske potensiell fremtidig arbeidskraft. Figur 17 viser brukerhistorien som tar utgangspunkt i behovet for informasjon, da bedriften må vite hva de skal og kan gjøre i løsningen for å kunne nå brukermålene sine og for å finne en kandidat til sin utlysning. Akseptansekriteriene viser til ulike funksjonaliteter bedriftene skal ha i systemet, som å registrere seg, opprette en utlysning, og prioritere studentene basert på søknadene. Akseptansekriterie A er ikke lenger relevant etter at bedriften har opprettet en bruker.

### Brukerhistorie for bedrift

**Som en** bedrift **ønsker jeg** å kunne tilby praksisplasser til studenter **slik at** jeg kan skape en relasjon med NTNU ved å utforske potensielle fremtidige arbeidstakere.

### Løsningsbeskrivelse

Det er nødvendig å vise frem relevant informasjon bedriften trenger for å vite hva som kreves av dem, hva de må gjøre, og hvordan, i tillegg til nok informasjon om studenter som sender søknader til dem til å gjøre en vurdering angående hvilken student som er best egnet for bedriften.

### Akseptansekriterier

- (A) Gitt at jeg **ikke** har tilgang til løsningen, så skal jeg kunne:
  - 1 Se vilkår for registrering
  - 2 Se relevant informasjon om hva involvering krever av meg
  - 3 Registrere meg i løsningen
- (B) Gitt at jeg **har** tilgang til løsningen, så skal jeg kunne:
  - 1 Lage utlysning om praksisplass eller prosjekt
    - a) Få tilbakemelding på om utlysningen er godkjent eller ikke
  - 2 Se oversikt og status på mine utlysninger
    - a) Aktive og ikke-aktive
- (C) Gitt at jeg **har** en aktiv utlysning, så skal jeg kunne:
  - 1 Se hvem og hvor mange som har søkt
    - a) Lese deres søknader
  - 2 Prioritere studenter som har søkt
    - a) Endre rekkefølge på prioriteringene



Figur 17: Brukerhistorie for brukergruppen bedrift.

Brukermålene til administrator er relasjonsbygging med næringslivet, kvalitetssikring og evaluering av bedrifter og å sikre at alle studenter har en plass. Figur 18 viser at administrator er avhengig av god oversikt for å holde kontroll over praksisemnet, samt nå sine brukermål.

### Brukerhistorie for administrator

**Som en** administrator ved Institutt for Design **ønsker jeg** å kunne se en oversikt over hvilke bedrifter instituttet har hatt eller har en avtale med **slik at** jeg kan ha kontroll over praksisemnet.

### Løsningsbeskrivelse

Det er nødvendig å vise frem relevant informasjon administratoren trenger for å tilby en oversikt som gir kontroll over prosessen. For eksempel hvilke studenter som har søkt, de som har påbegynt, og de som ikke har søkt. Noen av administrator sine oppgaver avhenger av hvor de andre brukergruppene er i prosessen. I tillegg vil alle funksjonaliteter for administrator være nødvendig å opprettholde uansett hvor i perioden man er, men akseptansekriteriene skrives inn hvor og når de anses som viktigst.

### Akseptansekriterier

- (A) Gitt at en praksisperiode **ikke** har startet, så skal jeg kunne:
  - 1 Se en oversikt over alle bedriftene som er registrert i systemet
    - a) Se bedriftenes kontaktinformasjon
    - b) Historisk oversikt over tidligere samarbeid
  - 2 Få varsling på e-post når en ny bedrift registrerer seg
  - 3 Motta forespørsler om utlysninger fra bedrifter
- (B) Gitt at en praksisperiode **har** startet, så skal jeg kunne:
  - 1 Se en oversikt over hvor studentene er i prosessen
    - a) Se hvilke studenter som har søkt på hvilke bedrifter

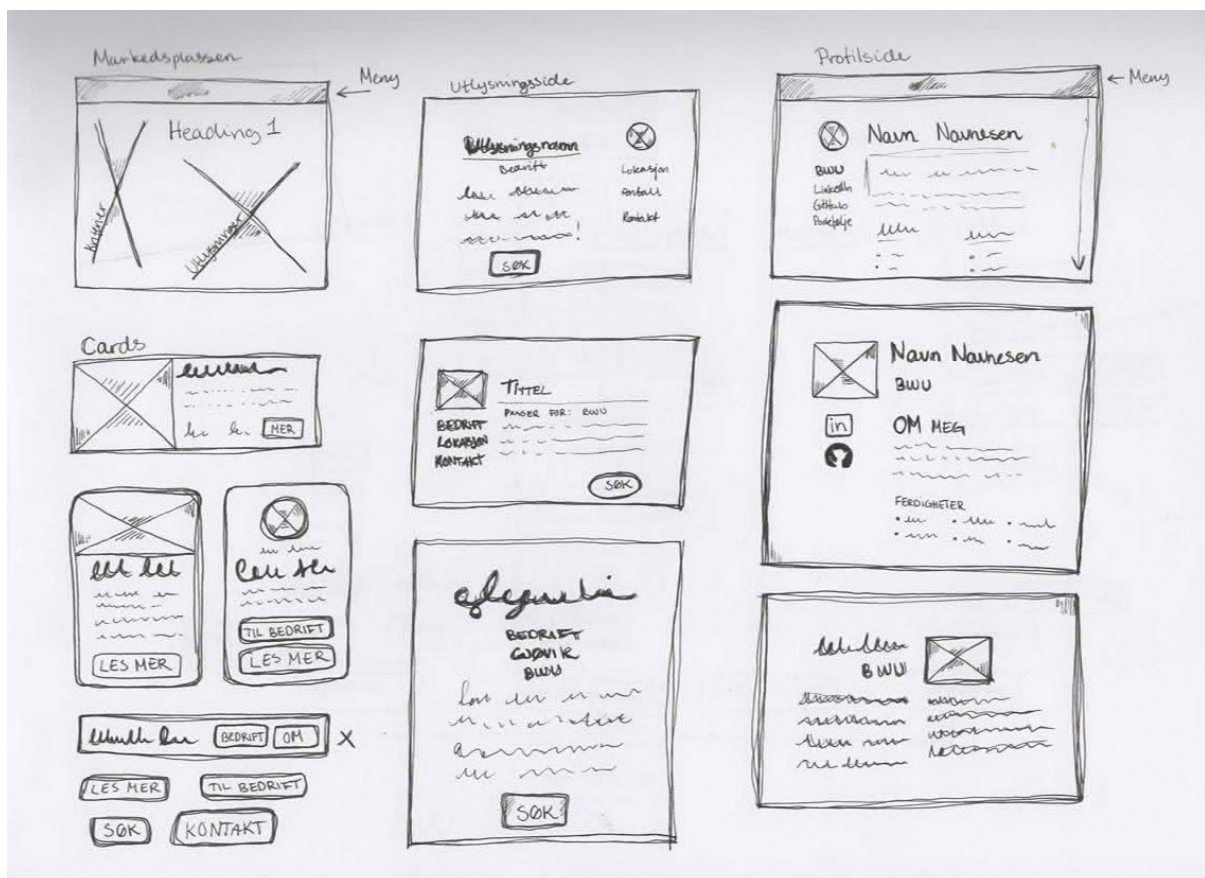


Figur 18: Brukerhistorie for brukergruppen administrator.

Som løsningsbeskrivelsen forklarer i Figur 18 er mye av funksjonaliteten til administrator avhengig av hvor de andre brukergruppene er i prosessen. I tillegg må funksjonalitetene være tilgjengelige gjennom hele praksisperioden. For eksempel viser akseptanskriteriene til at administrator skal kunne se en oversikt over alle bedriftene som er registrert i systemet, gitt at en praksisperiode *ikke* har startet. Dette er en oversikt som administrator også burde ha tilgang til etter at en praksisperiode *har* startet, men det anses som viktigst ved førstnevnte tilfelle med tanke på ny runde med rekruttering av bedrifter.

## 4.2.2 Skisser

Brukerhistoriene ble i denne fasen brukt som et utgangspunkt til nye ideer og skisser. Figur 19 viser en samling av skisser som prosjektgruppen mente hadde potensiale.



Figur 19: Samling av skisser på papir.

Skissene avslørte et behov for et navn for løsningen, et navn som er innbydende, lett å uttale, og lett å huske. Det første navnet «Bedriftsnettverk» ble ansett som noe uoriginalt og kjedelig, og gikk derfor gjennom flere iterasjoner. Det endelige navnet oppstod av en sammensmeltning av ordene «Bedrift» og «Nettverk», ved å sette de to første bokstavene fra «Bedrift» sammen med de tre første fra «Nettverk». Slik oppstod navnet «BeNet», som fulgte kriteriene for valget. Tabell 8 viser prosessen.

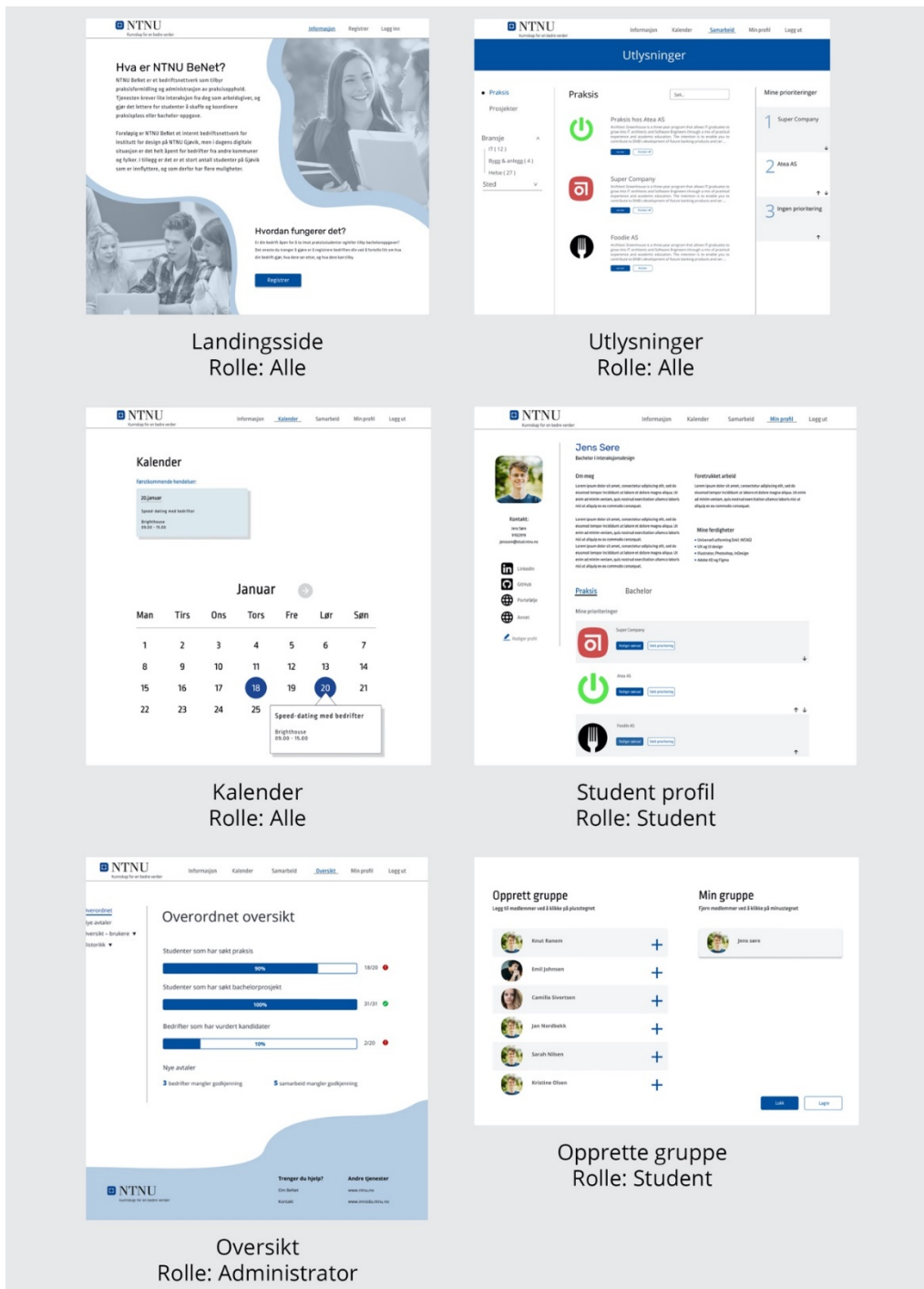
Løsningen vil videre bli referert til som BeNet.

Navneforslag	Kommentar	Valgt
Nettverk for bedrifter	Uoriginalt, kjedelig	Nei
Bedriftsnettverk	Beskrivende, langt	Nei
BedNett	Originalt, middels lengde	Nei
BeNet	Originalt, kort	Ja

Tabell 8: *Prosess for navngivning av løsningen.*

### 4.2.3 Første iterasjon av prototype

Skissene ble tatt med videre til det digitale verktøyet Figma, hvor noen av ideene ble forkastet, mens andre ble opphavet til den første versjonen av prototypen (se Figur 20). Figuren viser et utkast sider fra prototypen som representerer forskjellige oppgaver for de ulike brukergruppene.



Figur 20: Utvalg av første iterasjon av prototypen

Figur 20 viser et utvalg av sider fra prototypen. Landingsiden er den første en bruker vil møte i løsningen, hvor man kan lese om BeNet og enten gå videre til registrering eller innlogging. Det er også meningen at ikke-innloggede brukere kan se på utlysninger, det vil si tilgjengelige samarbeid mellom NTNU og bedrifter i form av praksisplasser eller bachelorprosjekter. Dette for å gi andre et større innblikk i hva BeNet er, og forhåpentligvis vil nye bedrifter la seg inspirere.



Kalender er ment som en side for å samle alt av informasjon om viktige hendelser relatert til praksis- og bacheloremnet. Denne siden er kun tilgjengelig for innloggede brukere, da for alle brukergruppene, altså student, bedrift, og administrator. Førstkommende hendelse vil bli markert i en egen informasjonsboks over selve kalenderen for å tydeliggjøre informasjonen. Resterende hendelser vil være tilgjengelig på de ulike datoene i selve kalenderen.

Studentprofil er ment å fungere som en generell søknad og CV. Studentene skal kunne skrive generelt om seg selv, hvilke ferdigheter de har, tidligere erfaring, og så videre, i tillegg til å legge inn bilde av seg selv og lenker til andre nyttige og relevante sider som LinkedIn og GitHub, eller en portefølje. Profiler inneholder også prioriteringene for praksisplass og bachelorprosjekt, funksjonalitet for å skrive søknad til praksis og opprette gruppe for bachelorprosjektet. En bedriftsprofil ser tilnærmet lik ut som studentprofil.

Oversikt er en side som kun er tilgjengelig for administrator, og som har fire undersider: Overordnet oversikt, nye avtaler, oversikt – brukere, og historikk. Den overordnede oversikten har som formål å raskt gi et innblikk i den mest nødvendige informasjonen. Den viser hvor mange studenter som har sendt inn tre søknader til praksis, hvor mange som har valgt bachelorprosjekt, og hvor mange bedrifter som har prioritert studenter for sin praksisplass. I tillegg viser denne siden hvor mange nye avtaler som venter, enten bedrifter som har registrert seg i systemet eller en ny utlysning for en praksisplass eller et bachelorprosjekt, som må vurderes. Nye avtaler viser da detaljer om nye bedrifter eller utlysninger som må vurderes. Oversikt – brukere inneholder detaljer om hvor de ulike brukerne av systemet er i praksis- eller bachelorløpet. Den siste undersiden, historikk, viser tidligere samarbeid.

Prototypen skal i tillegg til praksis illustrere tilrettelegging av bachelorprosjekt, på bakgrunn av innsiktsarbeidet i tidligere arbeid (se kapittel 3.1). Av den grunn er det blant annet mulig å opprette bachelorgruppe i prototypen (nederst til høyre i Figur 20).

#### 4.2.3.1 Målinger av første iterasjon

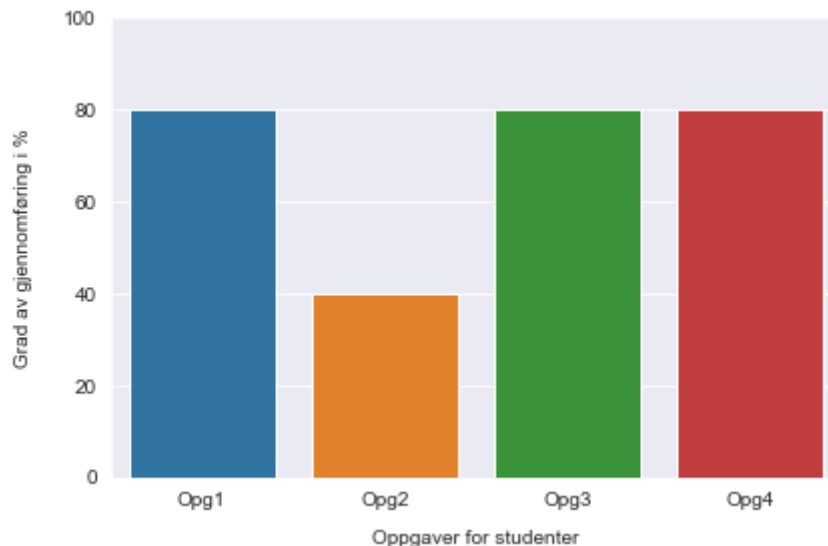
Prototypen ble i første iterasjon testet digitalt på 5 personer, 4 kvinner og 1 mann. Hver testbruker fikk utdelt alle de 8 oppgavene, oppgave 1–4 for brukergruppen student, oppgave 5–6 for brukergruppen bedrift, og oppgave 7–8 for brukergruppen administrator.

Rådata fra brukertesting for oppgave 1–4 blir presentert i Tabell 9. Tabellen viser tid i sekunder for hver oppgave, totaltid i sekunder, antall feiltrykk, og kommentarer.

Student	Tid brukt i sekunder					Feil	
Bruker	Opg1	Opg2	Opg3	Opg4	Totaltid	Antall feiltrykk	Kommentar
1	152	107	37	21	317	6	Får hint om å bruke hovedmenyen på oppgave 2
2	83	80	30	120	313	7	Spurte om hjelp for å finne utlysninger på opg1. Spurte om hjelp for å skrive søknad på opg2.
3	10	80	30	10	130	2	Hint om å skrive søknad på opg2. Fikk hint om å lagre på opg3.
4	35	198	20	131	384	9	Slet lenge med opg2. Spør om frister på opg4.
5	50	59	98	123	330	5	---

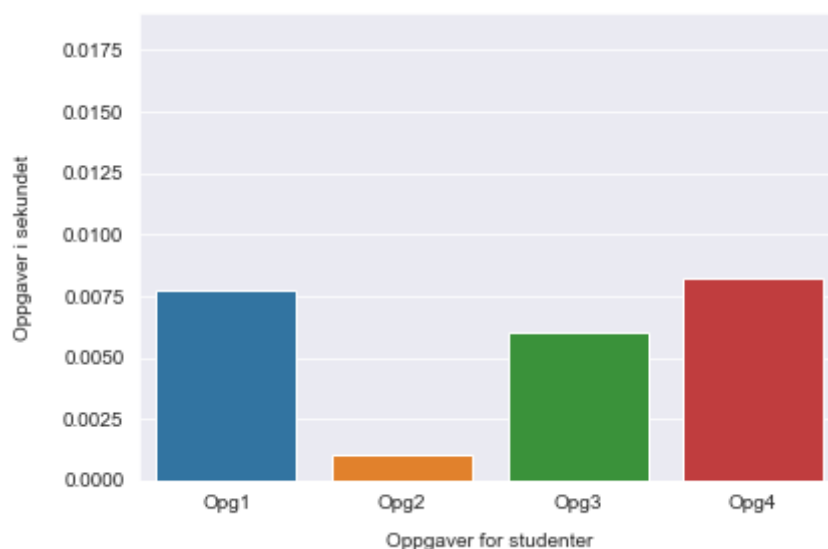
Tabell 9: Rådata fra brukertesting for brukergruppen student.

Målet for grad av gjennomføring var at alle oppgavene skulle ligge på minst 80% fullføringsgrad. Grad av gjennomføring for oppgavene for student, altså oppgave 1–4 vises i Figur 21. Målingene viser at oppgave 1, 3, og 4 har 80% i gjennomføringsgrad, og har nådd målet. Oppgave 2 har derimot en grad av gjennomføring på 40% (se Figur 21).



Figur 21: Grad av gjennomføring for brukergruppen student ved første iterasjon.

Grad av gjennomføring (Figur 21) viser at 60% (3 av 5 testbrukere), ikke klarte å gjennomføre oppgave 2 uten hjelp. Tiden for brukerne som ikke fullførte oppgave 2 ble nullt ut ved utregning av tidsbasert effektivitet. Dette betyr at de resterende 40% (2 av 5) av brukerne som klarte å gjennomføre oppgave 2 brukte lang tid på å gjennomføre, da den tidsbaserte effektiviteten for oppgavene 1 til 4 er svært lav (se Tabell 10). Tidsbasert effektivitet måles ved hjelp av en matematisk formel og tar alle fem brukerne i betraktning. Det vil si at hver søyle, altså hver oppgave, inneholder resultatene fra alle brukerne.



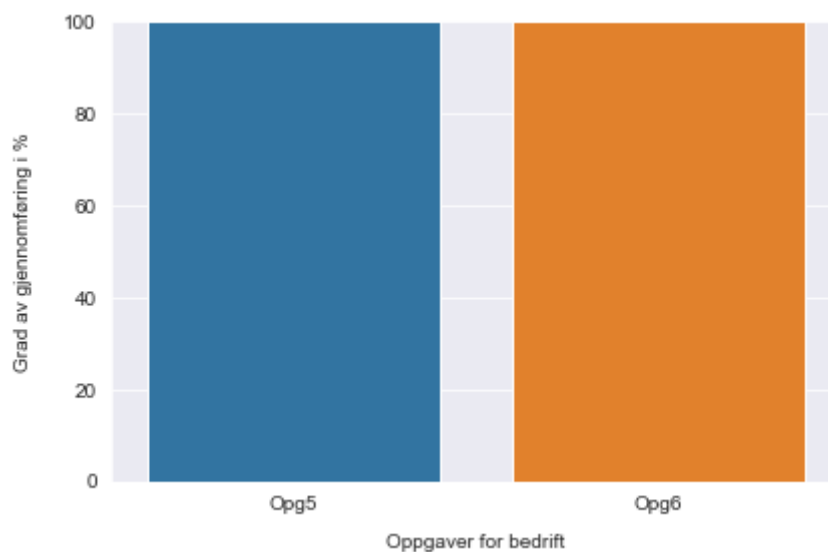
Figur 22: Tidsbasert effektivitet for brukergruppen student ved første iterasjon.

Rådata fra brukertesting for oppgave 5–6 blir presentert i Tabell 10. Tabellen viser tid i sekunder for hver oppgave, totaltid i sekunder, antall feiltrykk, og kommentarer.

Bedrift	Tid brukt i sekunder			Feil	
Bruker	Opg5	Opg6	Totaltid	Antall feiltrykk	Kommentar
1	181	78	259	0	---
2	55	40	95	1	---
3	73	14	87	0	---
4	53	25	78	0	---
5	67	50	117	0	---

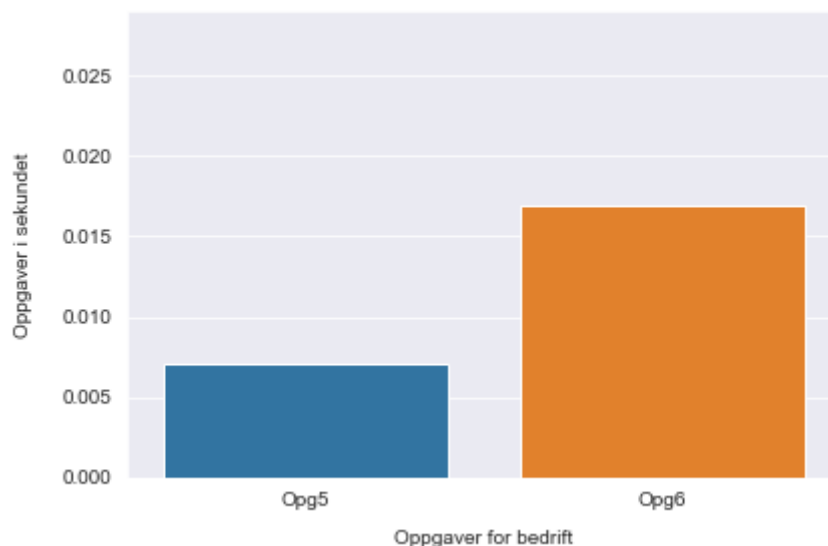
Tabell 10: Rådata fra brukertesting for brukergruppen bedrift.

Grad av gjennomføring for oppgave 5 og 6, altså oppgavene for brukergruppen bedrift nådde målet om 80% ved første iterasjon (se Figur 23). Oppgave 5 og oppgave 6 hadde begge en grad av gjennomføring på 100%, altså fullførte 5 av 5 testbrukere begge oppgavene uten å be om hjelp.



Figur 23: Grad av gjennomføring for brukergruppen bedrift ved første iterasjon.

Den tidsbaserte effektiviteten for både oppgave 5 og 6 var lav, begge med mindre enn 0.02 oppgaver gjennomført i sekundet (se Figur 24). Oppgave 5 hadde lavest tidsbasert effektivitet av de to, med mindre enn 0.01 oppgaver gjennomført i sekundet.



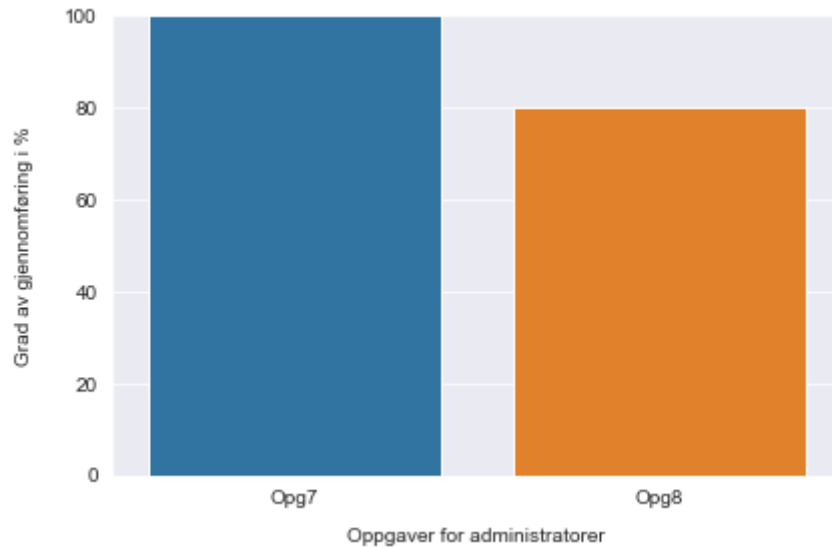
Figur 24: Tidsbasert effektivitet for brukergruppen bedrift ved første iterasjon.

Rådata fra brukertesting for oppgave 7–8 blir presentert i Tabell 11. Tabellen viser tid i sekunder for hver oppgave, totaltid i sekunder, antall feiltrykk, og kommentarer.

Admin	Tid brukt i sekunder			Feil	
Bruker	Opg7	Opg8	Totaltid	Antall feiltrykk	Kommentar
1	108	55	163	0	---
2	16	54	70	2	Hint om å vise historikk utelukkende fra 2020.
3	20	18	38	0	---
4	20	35	55	1	---
5	38	42	80	1	---

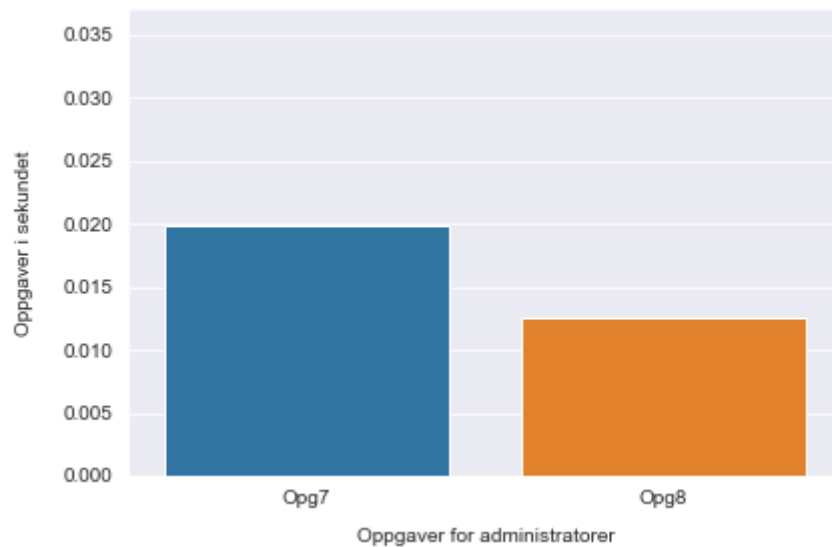
Tabell 11: Rådata fra brukertesting for brukergruppen administrator.

Grad av gjennomføring for administrator nådde, i likhet med bedrifter, målet om 80%. For oppgave 7 var grad av gjennomføring 100%, mens for oppgave 8 var den 80%, se Figur 25. Altså gjennomførte 4 av 5 brukere oppgave 8 uten hjelp.



Figur 25: Grad av gjennomføring for brukergruppen administrator ved første iterasjon.

Den tidsbaserte effektiviteten for både oppgave 7 og 8 var høyere enn 0.01 oppgaver i sekundet. Oppgave 7 hadde høyere effektivitet enn oppgave 8 (se Figur 26).



Figur 26: Tidsbasert effektivitet for brukergruppen administrator ved første iterasjon.

Brukertestene avdekket flere feil i risikovurderingen ved denne iterasjonen, én som ble vurdert av lav karakter, to av medium karakter og to av høy karakter (se Figur 27). Feilen av lav risikovurdering oppstod for brukergruppen administrator, mens feilene av medium og høy risikovurdering oppstod for brukergruppen student.

	LAV	MEDIUM	HØY	KRITISK
FØRSTE ITERASJON	Sortere etter år i historikk	Problemer med å finne frister Problemer med å finne utlysninger	Lukker fremfor å lagre. Får ikke advarsel før lukking. Problemer med å finne plassering for å skrive søknad	miro

Figur 27: Risikovurdering av funn i første iterasjon av prototypen.

Feilen av lav karakter for brukergruppen administrator, oppstod i oppgave 8 som omhandlet å sortere historisk data basert på år. Brukerne fikk i oppgave å finne bedrifter som hadde samarbeid med NTNU i 2020. Ingen hadde problemer med å finne historikken, men en av brukerne fant frem til de to samarbeidene uten å filtrere basert på år. Fordi denne brukeren måtte få hjelp for å fullføre oppgaven, ble den ansett som ikke gjennomført. Gjennomføringsgraden for oppgave 8 er derfor 80%. Hjelpen brukeren fikk var en påminning om å vise samarbeid utelukkende fra 2020, noe brukeren dermed ikke hadde noe problem med å løse. Denne feilen er derfor vurdert som en lav risiko.

Feilene av medium og høy karakter oppstod for brukergruppen student. Feilene som ble vurdert av medium karakter var problemer med å finne datoer for ulike frister og å finne «utlysninger». Dette var derimot ikke feil som fysisk hindret brukerne i å nå målet, men som førte til at brukerne brukte lang tid, opplevde frustrasjon, og ba om hint. To feil ble vurdert av høy karakter. En av feilene oppstod i oppgave 3, og gjorde at brukerne uvitende ikke fullførte oppgaven på riktig måte, da de valgt å lukke fremfor å lagre når de hadde opprettet en gruppe for bachelorprosjektet. Den andre feilen av høy risikovurdering oppstod i oppgave 2, hvor flere av brukerne ikke fant plasseringen for å skrive søknad til bedrifter de hadde prioritert. Selv om denne feilen kan virke noe lik som feilene av medium risikovurdering, ble den vurdert som høy da brukerne gav opp og måtte få hjelp for å fullføre.

#### 4.2.3.2 Drøfting av første iterasjon

Målet for grad av gjennomføring var 80%. Oppgave 2 var den eneste oppgaven som ikke nådde dette målet, da den hadde en gjennomføringsgrad på 40%. Den lave gjennomføringsgraden gjorde at den lave tidsbaserte effektiviteten også ble problematisk, da det var en indikator på at de som klarte å gjennomføre oppgaven brukte svært lang tid. I tillegg ble feilen som oppstod for oppgave 2 vurdert av høy risiko. Oppgaven omhandlet å skrive søknad for en valgt prioritering, og testbrukerne hadde store problemer med å finne plasseringen for dette. Dette kan skyldes at listen med prioriteringer var tilgjengelig på utlysningssiden – funksjonaliteten for å skrive søknad var *ikke* det. Funksjonalitet for å skrive søknad var kun tilgjengelig fra «min profil». Flere av brukerne forventet å finne disse to elementene sammen, ikke hver for seg på to ulike sider i informasjonsarkitekturen. Mye av kjernefunksjonaliteten for studenter er å kunne skrive søknad i BeNet, og ble derfor ansett som en feil som var svært viktig å rette opp i.

Den andre feilen av høy risikovurdering hvor brukerne valgte å lukke fremfor å lagre i opprettelse av bachelorgruppe, kan skyldes feilplassering (lukk var plassert til venstre, lagre til høyre) og feil bruk av primær- og sekundærknapp. I tillegg fikk ikke brukerne noen form for tilbakemelding om at de hadde glemt å lagre. I reelle omgivelser hadde denne feilen påvirket bachelorprosjekt i meget stor grad dersom den ikke hadde blitt oppdaget. Feilene som ble risikovurdert av medium karakter omhandlet i all hovedsak synliggjøring av informasjon. Det kan virke som om merkelappene i menyen ikke var tydelige nok for hva elementene inneholdt. Feilen av lav karakter ble derimot ansett som et problem i form av å tydeliggjøre oppgaven for at brukeren skal huske å filtrere på år i historikken.

Det at målet om minst 80% gjennomføringsgrad på alle de åtte oppgaven ikke var nådd var en klar indikator på at prototypen behøvde en ny iterasjon. Dessuten inneholdt risikovurderingen flere feil, hvor flest mulig burde bli eliminert før implementeringen startet. Summen av brukertesting og alle målingene i første iterasjon ledet til behovet for en ny iterasjon av prototypen, for å endre de problematiske aspektene som ble oppdaget.

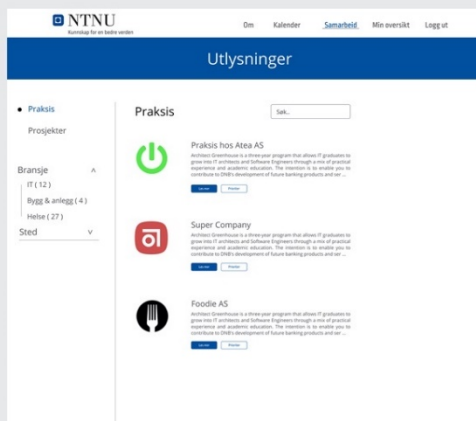
#### 4.2.4 Andre iterasjon av prototype

Målingene fra brukertesting av første iterasjon av prototypen viste at det var et behov for å gjennomføre en ny iterasjon. Den andre iterasjonen hadde som mål å forbedre prototypen ved å eliminere de oppdagede problemene fra målingene og risikovurderingen i første versjon. Figur 28 viser et utvalg av den andre iterasjonen. Begge iterasjonene av prototypen kan ses på følgende link: <https://www.figma.com/file/WUgOg6wg0v2wRw1oejLahw/BeNet?node-id=0%3A1>

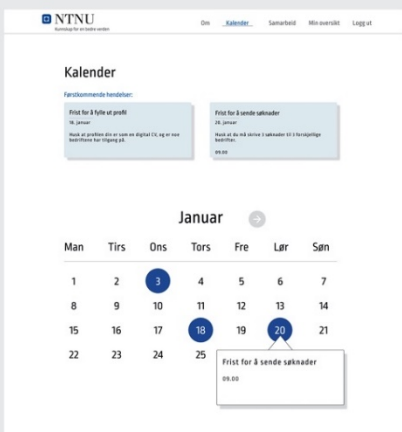




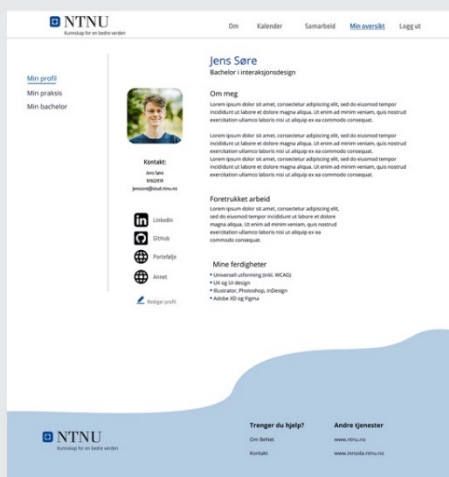
Landingside  
Rolle: Alle



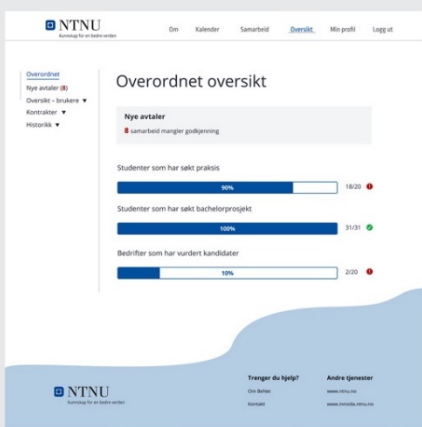
Utlysninger  
Rolle: Alle



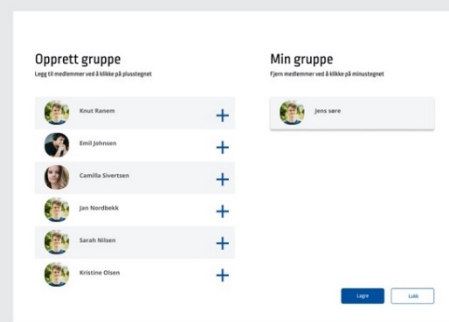
Kalender  
Rolle: Alle



Student profil  
Rolle: Student



Oversikt  
Rolle: Administrator



Opprette gruppe  
Rolle: Student

Figur 28: Utvalg av andre iterasjon av prototypen.

Utdraget vist i Figur 28 viser de samme sidene som i utdraget i Figur 20 i kapittel 4.2.3 om første iterasjon av prototypen, da det er disse sidene som har gjennomgått de største endringene. Endringene har blitt utført for å rette opp i problemene som ble avdekket i risikovurderingen av første iterasjon (se kapittel 4.2.3).

Problemer med å finne ulik informasjon som utlysninger og frister, som ble vurdert til medium risiko, ble endret ved blant annet ny navngiving av merkelapper i menyen for å tydeliggjøre hva de ulike menyelementene inneholdt. «Min profil» ble endret til «min oversikt», som ble delt inn i undersidene «min profil», «min praksis», og «min bachelor». Navngiving av «kalender» ble ikke endret. I tillegg ble prioriteringer fjernet fra «samarbeid», da å finne plassering for å skrive søknad var et problem av høy karakter, fordi brukerne ble forvirret av at prioriteringer var plassert flere steder. Prioriteringer og funksjon for å skrive søknad ble derfor plassert kun i «min oversikt».

Problemet som ble vurdert av høy risiko hvor brukerne valgte å lukke fremfor å lagre, tydeliggjorde at det var nødvendig å bytte om på plasseringen av knappene, samt at lagre-knappen måtte være primærknapp, og lukk-knappen som sekundærknapp. Ny informasjon for administrator ble flyttet fra nederst på administratorsiden «overordnet oversikt» til øverst på siden. Sistnevnte var ikke et risikovurdert problem, men snarere noe som ble endret i et forsøk på å forbedre den tidsbaserte effektiviteten.

#### 4.2.4.1 Målinger av andre iterasjon

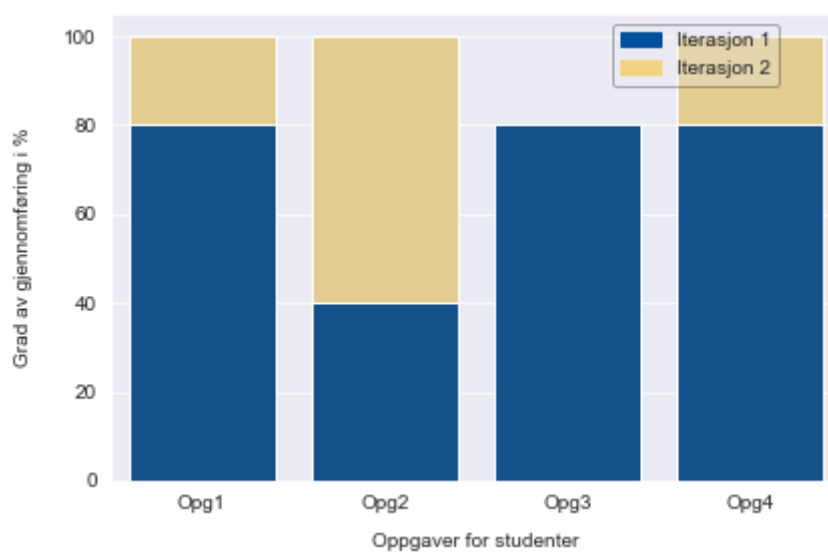
Målingene fra brukertesting av andre iterasjon vil bli vist i sammenstilling med målingene fra første iterasjon for å vise om endringer har påvirket positivt, negativt eller forblitt det samme. Andre iterasjon av prototypen ble testet fysisk på 5 personer, 2 kvinner og 3 menn. Hver testbruker fikk utdelt alle de 8 oppgavene, oppgave 1–4 for brukergruppen student, oppgave 5–6 for brukergruppen bedrift, og oppgave 7–8 for brukergruppen administrator.

Rådata fra brukertesting for brukergruppen student, altså oppgave 1–4 blir presentert i Tabell 12. Tabellen viser tid i sekunder for hver oppgave, totaltid i sekunder, antall feiltrykk, og kommentarer.

Student	Tid brukt i sekunder					Feil	
Bruker	Opg1	Opg2	Opg3	Opg4	Totaltid	Antall feiltrykk	Kommentar
1	20	29	20	10	79	2	---
2	22	10	20	35	87	2	Glemte å lagre gruppe på opg3.
3	17	10	16	9	52	0	---
4	21	19	12	10	62	0	---
5	19	15	22	26	82	0	---

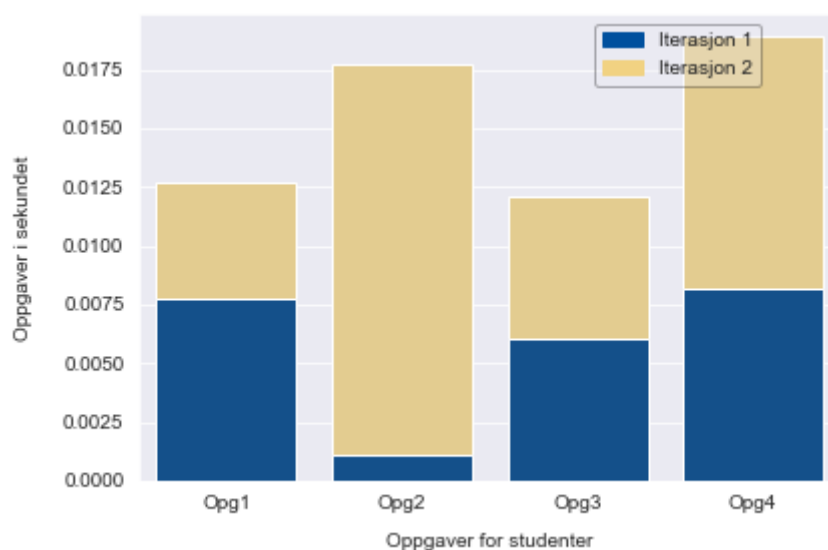
Tabell 12: Rådata fra brukertesting - brukergruppe student, andre iterasjon.

Målingene fra oppgavene for studenter i andre iterasjon av prototypen viser at graden av gjennomføring har gått opp (se Figur 29). Alle oppgavene for brukergruppen student har i denne runden nådd målet om 80% gjennomføringsgrad. Oppgave 1, 2, og 4 har steget til 100% gjennomføringsgrad, mens oppgave 3 fortsatt ligger på 80%. For oppgave 2 har graden av gjennomføring steget fra 40% til 100%.



Figur 29: Grad av gjennomføring for brukergruppen student i andre iterasjon.

Den tidsbaserte effektiviteten for oppgave 1–4 har steget (se Figur 30), altså utføres det flere oppgaver i sekundet ved andre iterasjon av prototypen enn ved første iterasjon. For oppgave 2 har både graden av gjennomføring og den tidsbaserte effektiviteten steget. Oppgave 4 er den oppgaven for brukergruppen student som har høyest tidsbasert effektivitet.



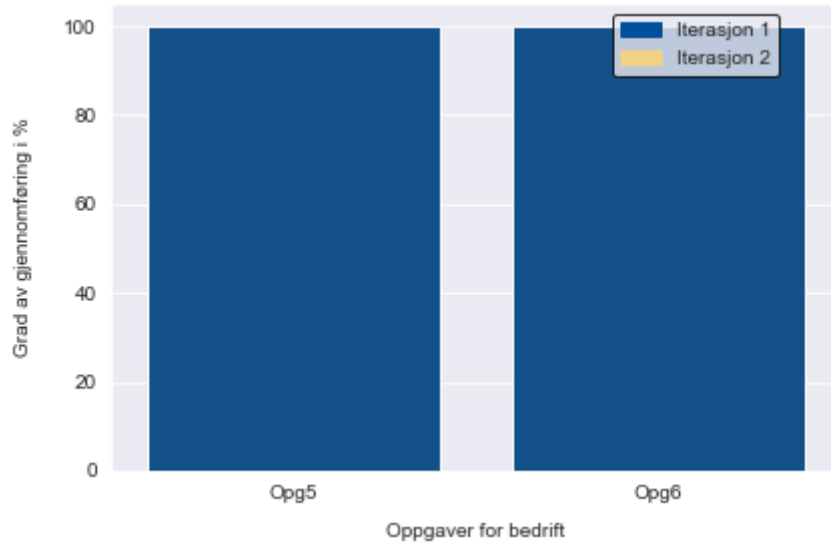
Figur 30: Tidsbasert effektivitet for brukergruppen student i andre iterasjon.

Rådata fra brukertesting for brukergruppen bedrift, altså oppgave 5–6 blir presentert i Tabell 13. Tabellen viser tid i sekunder for hver oppgave, totaltid i sekunder, antall feiltrykk, og kommentarer.

Bedrift	Tid brukt i sekunder			Feil	
Bruker	Opg5	Opg6	Totaltid	Antall feiltrykk	Kommentar
1	46	26	72	0	---
2	5	15	20	0	---
3	50	22	72	1	---
4	30	20	50	0	---
5	57	21	78	1	---

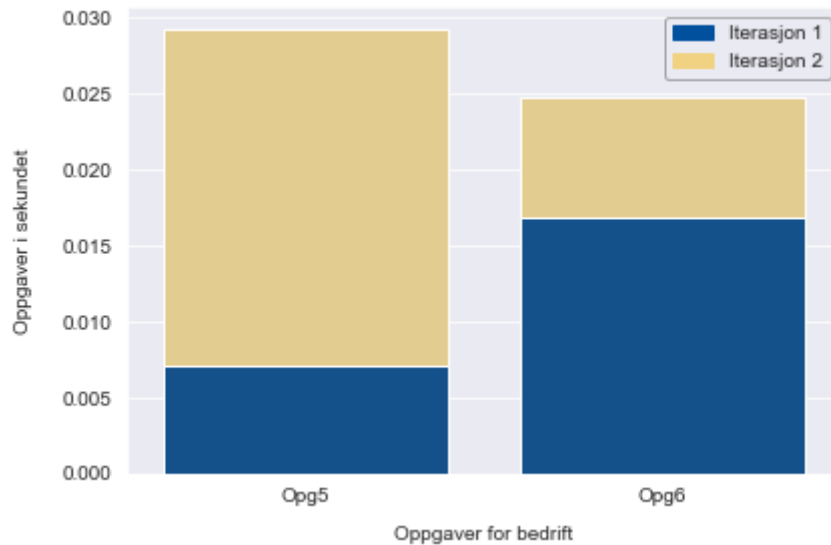
Tabell 13: Rådata fra brukertesting - brukergruppe bedrift, andre iterasjon.

Grad av gjennomføring for oppgave 5 og 6 har ikke forandret seg etter første iterasjon. Begge hadde fortsatt en gjennomføringsgrad på 100% (se Figur 31). Fargen for iterasjon 2 er ikke synlig på figuren da den strekker seg over det samme området som fargen for iterasjon 1.



Figur 31: Grad av gjennomføring for brukergruppen bedrift i andre iterasjon.

Den tidsbaserte effektiviteten for oppgave 5–6 har steget (se Figur 32). Oppgave 5 har nå steget forbi oppgave 6 i oppgaver i sekundet. Begge oppgavene ligger nå over 0.02 oppgaver i sekundet.



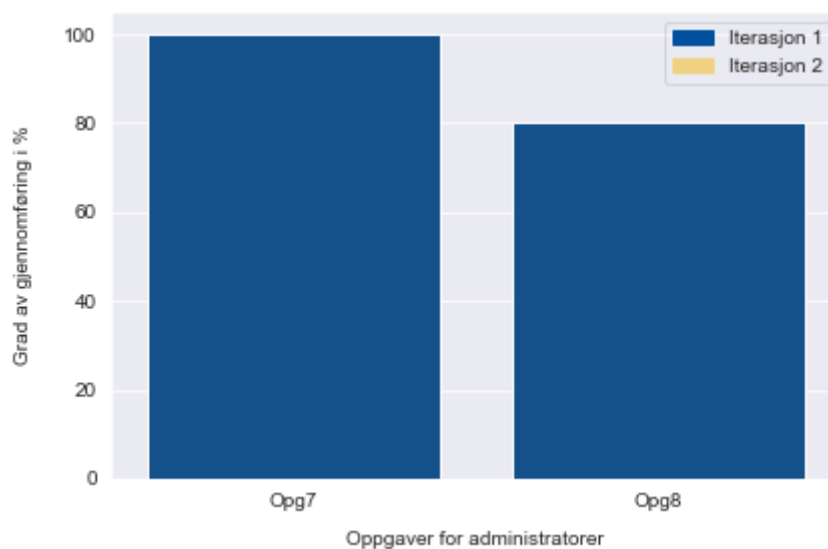
Figur 32: Tidsbasert effektivitet for brukergruppen bedrift i andre iterasjon.

Rådata fra brukertesting for brukergruppen administrator, altså oppgave 7–8 blir presentert i Tabell 14. Tabellen viser tid i sekunder for hver oppgave, totaltid i sekunder, antall feiltrykk, og kommentarer.

Administrator	Tid brukt i sekunder			Feil	
Bruker	Opg7	Opg8	Totaltid	Antall feiltrykk	Kommentar
1	10	14	24	0	---
2	10	20	30	0	---
3	18	30	48	1	---
4	15	21	36	0	---
5	23	35	58	1	Hint om å utelukkende vise innlegg fra 2020 på opg2.

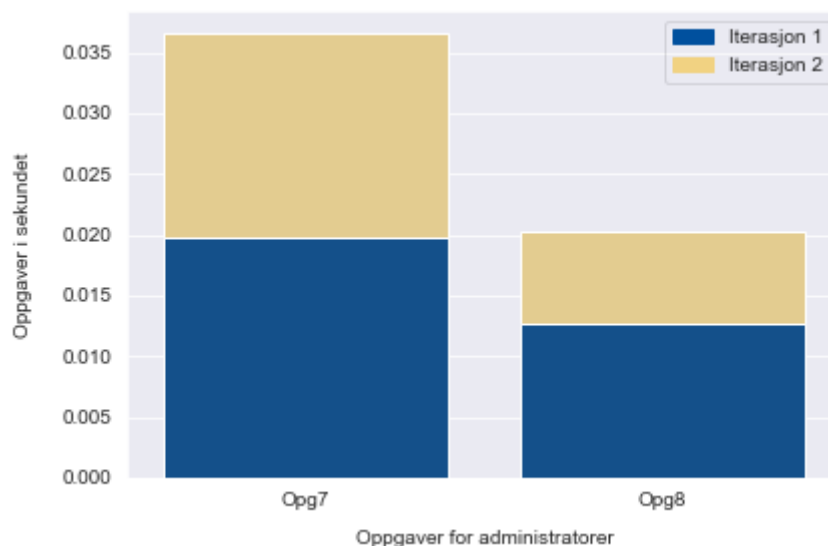
Tabell 14: Rådata fra brukertesting - brukergruppe administrator, andre iterasjon.

Grad av gjennomføring for oppgave 7–8 har, i likhet med oppgave 5–6, ikke forandret seg etter første iterasjon. Oppgave 7 har fortsatt en gjennomføringsgrad på 100%, oppgave 8 fortsatt 80% gjennomføringsgrad (se Figur 33). Fargen for iterasjon 2 er ikke synlig på figuren da den strekker seg over det samme området som fargen for iterasjon 1.



Figur 33: Grad av gjennomføring for brukergruppe administrator i andre iterasjon.

Den tidsbaserte effektiviteten for oppgave 7 og 8 har begge steget (se Figur 34). Oppgave 7 har nå steget forbi 0.035 oppgaver i sekundet, og er dermed den oppgaven med høyest tidsbasert effektivitet av alle de åtte oppgavene. Oppgave 8 har steget forbi 0.02 oppgaver i sekundet.



Figur 34: Tidsbasert effektivitet for brukergruppe administrator i andre iterasjon.

Brukertesting av første iterasjon av prototypen avdekket to problemer med medium risiko, samt to problemer med høy risiko. Andre iterasjon av prototypen satte søkelys på å eliminere flest mulige problemer. En ny risikovurdering viste at problemene av medium karakter, samt problemet av høy karakter fra oppgave 2 som omhandlet søknadsskriving ble eliminert, samtidig bestod ett problem av høy karakter og ett problem av lav karakter (se Figur 35). Ingen nye problemer ble avdekket.

	LAV	MEDIUM	HØY	KRITISK
FØRSTE ITERASJON	Sortere etter år i historikk	Problemer med å finne frister Problemer med å finne utlysninger	Lukker fremfor å lagre. Får ikke advarsel før lukking. Problemer med å finne plassering for å skrive søknad	
ANDRE ITERASJON	Sortere etter år i historikk		Lukker fremfor å lagre. Får ikke advarsel før lukking.	miro

Figur 35: Risikovurdering av funn i andre iterasjon av prototypen.

Feilen som ble vurdert av lav risiko oppstod for brukergruppen administrator i oppgave 8 på samme måte som ved første iterasjon. Brukeren fant de to samarbeidene fra 2020 i tabellen på historikk. Etter å ha blitt påminnet om å vise utelukkende disse to samarbeidene, ble oppgaven fullført raskt. Derfor er denne feilen fortsatt vurdert av lav risiko. I tillegg gjenstod ett av problemene av høy kvalitet. Dette problemet oppstod for brukergruppen student i oppgave 3, da en testbruker også ved denne iterasjonen valgte å lukke fremfor å lagre etter å ha opprettet en bachelorgruppe.

#### 4.2.4.2 Drøfting av andre iterasjon

Målingene fra brukertestene av første og andre iterasjon viser tydelig fremgang fra første iterasjon til andre iterasjon. Oppgave 3 og 8 nådde en gjennomføringsgrad på 80%, mens resterende oppgaver nådde 100%. Altså nådde alle de åtte oppgavene målet om *minst* 80% gjennomføringsgrad. Den tidsbaserte effektiviteten steg også for alle de åtte oppgavene, som viste at den andre iterasjonen var mer effektiv.

Fordi målet for gjennomføringsgrad ble nådd ved andre iterasjon ble det ikke ansett som nødvendig med en ny iterasjon av prototypen. Problemet av lav karakter var lett å utføre så fort en påminning om den fulle oppgaven ble gitt. Problemet som gjenstod av høy risiko omhandlet en oppgave som utføres i forbindelse med bacheloremnet, hvor testbrukerene lukket fremfor å lagre. Altså var det ikke tilstrekkelig å endre på plassering og funksjonalitet tilknyttet knappene for å eliminere problemet. Likevel er dette et problem som ikke vil ha en innvirkning på håndteringen og gjennomføringen av praksisemnet. Da fokuset for prosjektet er å utforme og utvikle en løsning for praksisemnet og alle oppgavene som er rettet mot dette emnet ikke hadde problemer som medførte betydelig risiko eller hindret brukerne i å fullføre, ble utformingsfasen avsluttet her for å bruke resterende tid på utviklingsfasen.

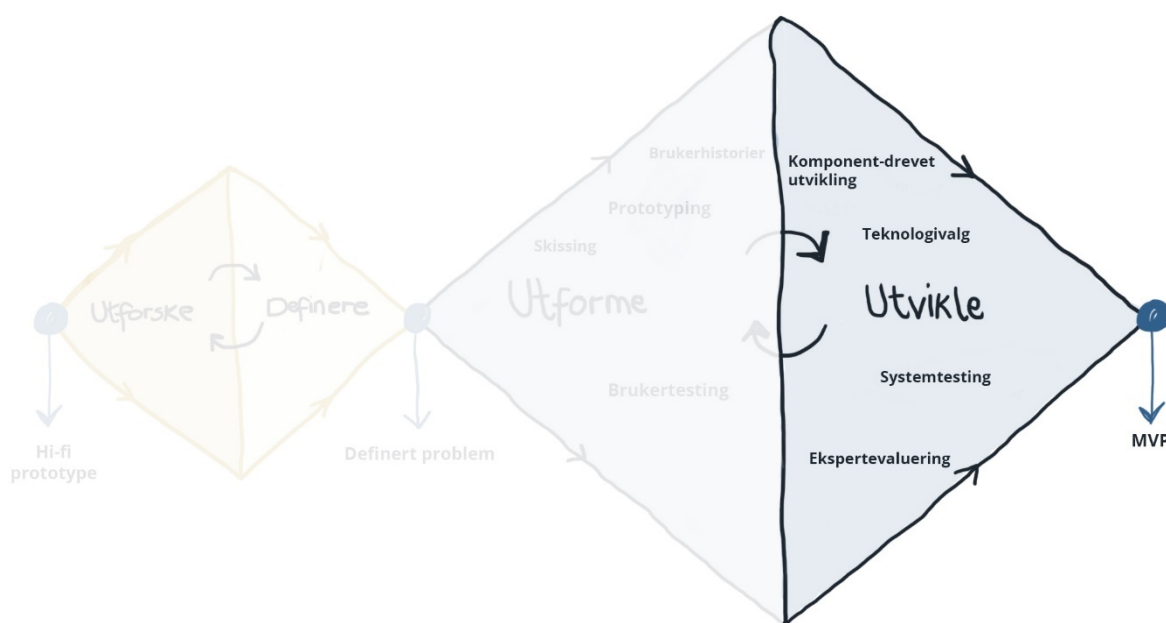
### 4.3 Oppsummering av utformingsfasen

Målet for første fase var å utforme en fungerende prototype. Resultatene viser at den første delen av prosjektet utformet et funksjonelt brukergrensesnitt for håndteringen av praksisemnet for alle tre brukergruppene, med et utgangspunkt for design og ikke minst et navn på løsningen. En svakhet er at man sitter igjen med et problem av lav karakter og et problem av høy karakter etter siste brukertest, men en styrke er derimot at det ikke er oppdaget noen problemer som medfører risiko for håndtering av praksisemnet, i tillegg til at kjernefunksjonalitet er utformet på en slik måte at brukergruppene kan gjennomføre sine oppgaver innen kort tid og med stor grad av gjennomføring. Prototypen blir tatt med til utviklingsfasen, som fokuserer på videre utvikling og implementering av webløsningen.



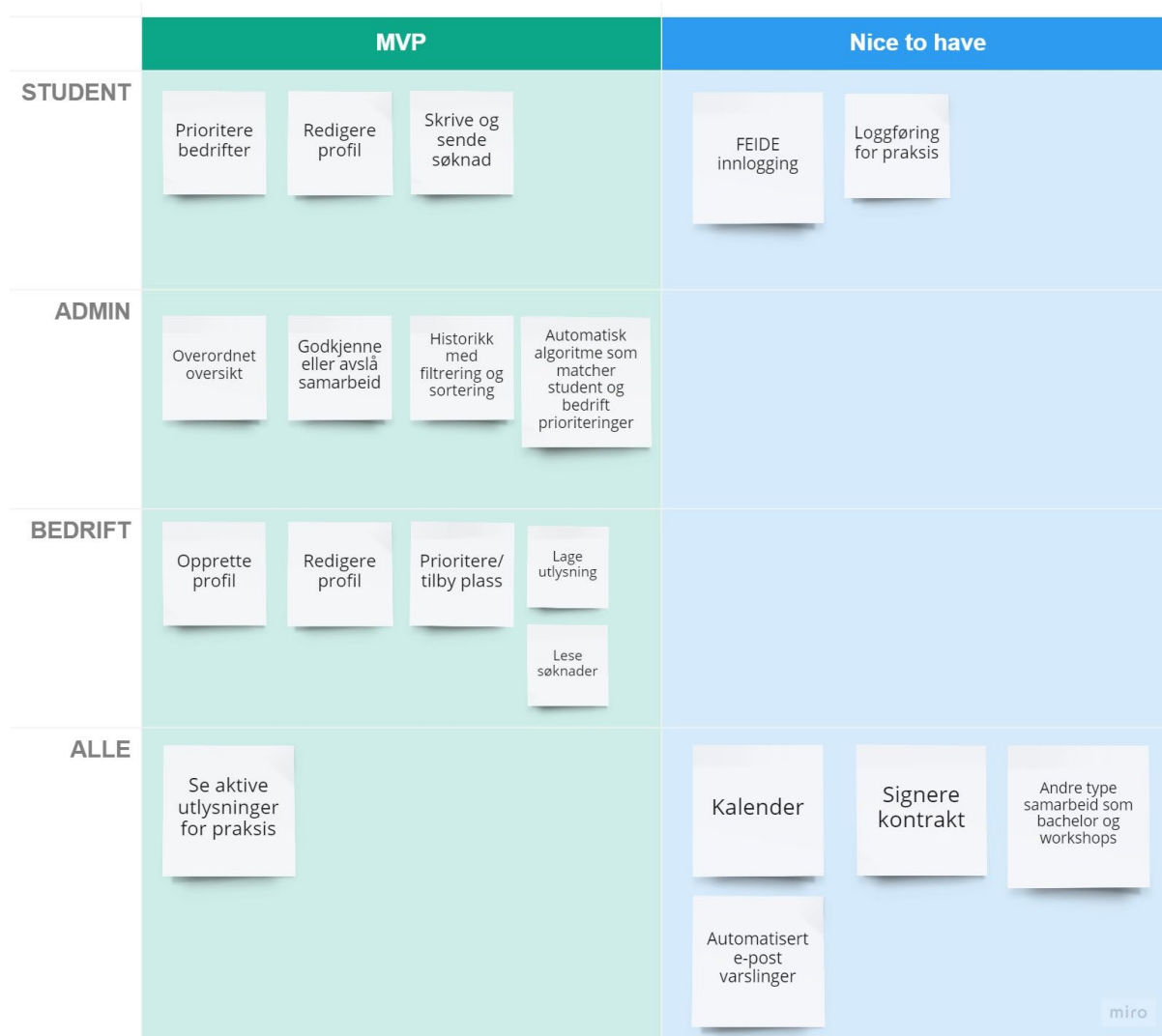
## 5 Utvikle

Den siste fasen av double diamond-modellen er utviklingsfasen (Figur 36), og har som formål å utvikle og implementere den faktiske webløsningen, ved å bruke prototypen fra utformingsfasen som et grunnlag for funksjonalitet og design som utgangspunkt. Da dette innebærer å kode løsningen fra bunnen og opp, er det også avgjørende å velge hvilke verktøy som skal tas i bruk, det vil si hva slags programmer og teknologier som skal anvendes.



Figur 36: Utvikle – andre fase av double diamond som er fokus for kapittel 5.

I prototypen kom det frem at oppdragsgiver og prosjektgruppen hadde veldig store ambisjoner og mange ideer til hva BeNet kunne inneholde. Da formål med prosjektet er å utvikle en MVP (minste brukbare produkt) som skal vise en bedre håndtering av praksisemnet som forklart i kapittel 1.3, ble det ansett som nødvendig å skille mellom funksjonalitet for MVP og funksjonalitet som ikke er kritisk for å få gjennomført praksisemnet, før utviklingsfasen startet. Med resultatet fra utformingsfasen (se kapittel 4.2) tatt i betraktning, ble det i samarbeid med oppdragsgiver kartlagt funksjonalitet i to kategorier, MVP og «nice-to-have» (se Figur 37). På denne måten blir det tydelig hvilke funksjonaliteter og muligheter som løsningen trenger for at brukerne skal kunne nå sine brukermål.



Figur 37: Funksjonalitet kategorisert i MVP og «nice-to-have»

Formålet med prosjektet er å tilrettelegge for praksisemnet, noe som preger hvilke funksjonaliteter som ble ansett som MVP. Funksjonalitet som var knyttet til selve brukerflyten for praksisemnet og prosessen med å lyse ut og tildele praksisplasser ble ansett som nødvendig som MVP-funksjonalitet. Det å kunne se aktive utlysninger for praksisplasser er vesentlig for alle brukergruppene da det er disse som er mulig å velge mellom. For studenter innebærer en MVP at de skal kunne prioritere bedrifter de ønsker, samt skrive og sende søknad. I tillegg er det nødvendig at studenter har mulighet til å redigere sin profil, da denne profilen på mange måter skal fungere som overordnet informasjon om studenten hva gjelder erfaring og motivasjon.

Da det tidligere innsiktsarbeidet (se kapittel 3.1.2.3) viste at autonomi for studenter og bedrifter ville minske arbeidsmengden for oppdragsgiver, altså brukergruppen administrator (admin i Figur 37), stod funksjonalitet som tillot oversikt sentralt for denne brukergruppen. Administrator må kunne se en overordnet oversikt som viser hvilke studenter og hvilke bedrifter som er aktive i systemet, samt en

historikk som viser tidligere samarbeid. Tidligere har administrator matchet studenter og bedrifter manuelt basert på ønsker (som forklart i kapittel 1.1). Dette er en svært krevende oppgave for oppdragsgiver, og automatisk matching ble derfor ansett som en svært viktig funksjonalitet for en MVP. Dessuten må administrator kunne godkjenne eller avslå nye utlysninger fra bedriftene som er registrert i systemet. Uten sistnevnte ville prosessen ha stoppet opp allerede før utlysningen ble tilgjengelig for studentene.

Brukergruppen bedrifter er avhengig av at de først og fremst kan registrere seg i systemet ved å opprette en profil, som de burde ha mulighet til å redigere. For å delta i praksisprosessen er en bedrift nødt til å utlyse en eller flere plasser, og dermed må funksjonalitet for å opprette en utlysning prioriteres. Søknadene til utlysningen må kunne leses, og studenter må kunne prioriteres basert på disse.

Det var viktig å vurdere om det var mulig å også tilrettelegge for bacheloremnet i systemet, da tidligere arbeid viste at det var ønskelig, og at de to emnene kunne bli lagt opp på svært lik måte (se kapittel 3.1.2). Derimot er ikke håndtering av bacheloremnet, eller andre samarbeidstyper som workshops og gjesteforelesninger, kritisk for å lage en MVP for håndtering av praksisemnet. Alt som omhandler håndtering av bacheloremnet, er derfor ansett som «nice-to-have». Problemet som gjenstod av høy risikovurdering etter andre iterasjon av prototypen fra utformingsfasen (se kapittel 4.2.4), altså oppgave 2 i brukertesting, omhandlet oppretting av grupper i forbindelse med bacheloremnet, og oppgaven har derfor blitt ansett som «nice-to-have». Problemet som gjenstod av høy risikovurdering etter andre iterasjon av prototypen fra utformingsfasen (se kapittel 4.2.4), altså oppgave 2 i brukertesting, omhandlet oppretting av grupper i forbindelse med bacheloremnet, og oppgaven har derfor blitt ansett som «nice-to-have». Problemet har derfor blitt revurdert som ikke kritisk å løse ved utviklingen av en MVP.

Flere funksjonaliteter for student, administrator, og alle brukergruppene, har blitt vurdert som «nice-to-have». Innlogging med Feide er ansett som ønskelig, da dette innloggingssystemet allerede blir brukt i andre løsninger tilknyttet NTNU. Dermed er dette noe brukergruppene administrator og student innehar fra før, men da formålet er å vise en bedre håndtering av praksisemnet, blir ikke innlogging med Feide ansett som nødvendig for en MVP. I tillegg har studenter et arbeidskrav om å føre logg gjennom hele praksisperioden. For en fullstendig løsning hadde det vært gunstig å inkludere dette i systemet, men det har ikke blitt ansett som et kritisk steg for oppdragsgiver, og derfor ikke inkludert. Kalender, kontraktsignering, og automatisert varsling på e-post har også blitt vurdert som «nice-to-have» da det ikke er funksjoner som er nødvendig for å vise hvordan gjennomføring av praksisprosessen skal fungere.

## 5.1 Arkitektur og teknologi

Arkitekturen for prosjektet tar utgangspunkt i resultatene fra utformingsfasen. Dette kapittelet har som hensikt å først dokumentere hvilke teknologiske og arkitektoniske hensyn som har blitt tatt, og deretter hvilke tilnærminger og teknologier som ble valgt. Prosjekts hensikt er å bygge et *distribuert system*, nærmere bestemt en webløsning, da de ulike aktørene skal samhandle og utveksle data gjennom en nettleser. Et distribuert system «[...] er en modell hvor komponentene i systemet befinner seg på datamaskiner i et datanett» (Distribuert databehandling, 2020).

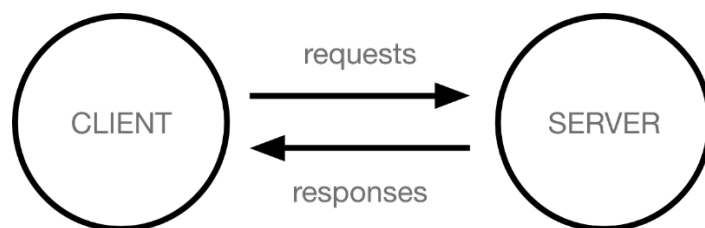
### 5.1.1 Overordnet arkitektur

Paganini (2019) sin distinksjon mellom programvarearkitektur og systemarkitektur har blitt lagt til grunn for overordnede arkitektoniske hensyn:

**Programvarearkitektur** refererer til den logiske organisasjonen av et distribuert system i programvarekomponenter [...] [Egen oversettelse].

**Systemarkitektur** refererer til plassering av disse programvarekomponentene på fysiske maskiner [...] [Egen oversettelse].

Prosjektet vil måtte forholde seg til begge, da systemarkitektur omhandler *hva* slags arkitektur som preger løsningen, og programvarearkitektur legger grunnlaget for *hvordan* løsningen kommuniserer innad arkitekturen. Som et eksempel på systemarkitektur trekker Paganini (2019) frem klient-tjenermodellen (se Figur 38). Klienter kommuniserer med tjenere (Mozilla, u.å.), og er en allmenn anerkjennelse hva angår web-applikasjoner.



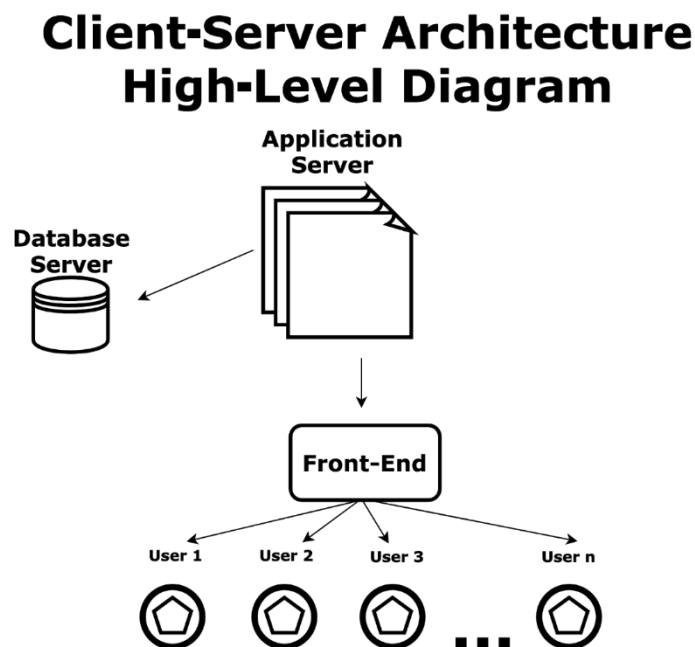
Figur 38: Klient-tjener-modellen (Mozilla, u.å.).

Det finnes flere typer programvarearkitekturer. Klienter kommuniserer med tjenere fortrinnsvis over HTTP (Mozilla, u.å.), som igjen betyr at programvarearkitekturen må være kompatibel med HTTP-protokollen. Representational State Transfer (REST) implementeres over HTTP i de aller fleste

tilfeller (Özlu, 2018). Alternative arkitekturer er CORBA, RPC eller SOAP for nettverksapplikasjoner, men er også mer krevende å implementere (Özlu, 2018). Özlu (2018) har dessuten et poeng om enkel kommunikasjon med flere typer klienter, uten at klienten trenger ytterligere konfigurering for å samhandle med tjener – noe som er i henhold til hensyn om en responsiv nettside.

### 5.1.2 Bakgrunn for teknologistabel

Gitt at overnevnte arkitektoniske hensyn har blitt tatt, har man flere muligheter hva angår forhåndsdefinerte teknologier, programvare, rammeverk, programmeringsspråk og tilnærminger. En løsning som forholder seg til klient-tjener modellen har tre konseptuelle deler en må ta stilling til, altså frontend, backend og database, som vist i Figur 39 (Esterkin, 2020).



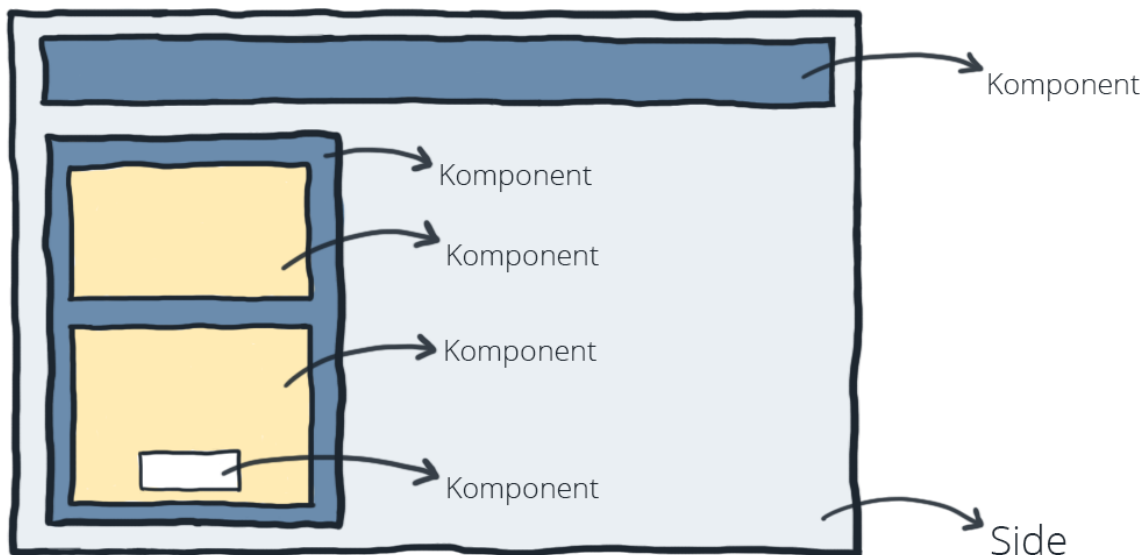
Figur 39: Client-Server Architecture (Esterkin, 2020).

Dette kapitlet har som videre hensikt å belyse ytterligere arkitektoniske og designrelaterte valg som prosjektet har stått ovenfor, da disse utgjør grunnlaget for den endelige løsningen, som vil bli redegjort for i kapittel 5.1.3 Prosjektets teknologistabel.

#### 5.1.2.1 Komponent-drevet utvikling

Et brukergrensesnitt er kontaktflaten mellom brukeren og datamaskinens operativsystem og programmer (Rossen, 2020). Det er altså hva brukeren ser. For å utvikle et brukergrensesnitt kan komponent-drevet utvikling brukes som en tilnærming for å utvikle brukergrensesnittet ved bruk av

komponenter. En komponent er kort fortalt en byggekloss i et brukergrensesnitt (Component Driven, u.å.). Formålet er å kombinere de ulike komponentene for å bygge funksjonalitet og utforme et brukergrensesnitt. Dette gjør at en komponent kan bli brukt flere steder og dermed bli tilpasset ulike bruksområder, noe som gir en mer effektiv utvikling. Figur 40 illustrerer hvordan ulike komponenter blir brukt for å bygge opp grensesnittet på en side.



Figur 40: Illustrasjon av komponent-drevet oppbygning. Illustrert av Sarah Berg.

Å bygge brukergrensesnittet ved hjelp av komponent-drevet utvikling har flere fordeler. For det første vil det bli lettere å finne problemer i koden, da man kan feilsøke på komponentnivå og teste i isolasjon (Component Driven, u.å.). Det tillater også utviklere å fokusere på en liten del av grensesnittet om gangen, noe som fører til bedre kommunikasjon mellom medlemmene av utviklingsteamet, som igjen gjør det lettere å fordele arbeid på medlemmene. I tillegg kan man fastslå at brukergrensesnittet fungerer som ønsket i ulike scenarier ved å bygge ut komponenter i isolasjon og definere deres relevante «states» (Component Driven, u.å.). Summen av fordelene ved en slik tilnærming, altså komponent-drevet utvikling, vil ifølge Tom Coleman (2017) tilrettelegge for utvikling av komplekse brukergrensesnitt, og det er på bakgrunn av fordelene beskrevet at dette prosjektet har lagt komponent-drevet utvikling til grunn for utviklingsprosessen og planleggingen.

Moderne rammeverk og bibliotek for komponent-drevet utvikling, som Vue, React og Angular, tilbyr ofte overlappende kjernefunksjonalitet. Det er imidlertid en forskjell på de ulike rammeverkene fokusområder, syntaks, ytelse og støtte. Vue har god ytelse og god dokumentasjon, samtidig som det har en slakkere læringskurve sammenlignet med React og Angular (Borrelli, 2021). React og Angular

fokuserer imidlertid på andre områder. I Angular vil man finne lignende direktiv som i Vue, men også en brattere læringskurve (Borrelli, 2021). Samtlige teknologier kommer med forhåndsdefinert funksjonalitet for å håndtere «state», altså tilstanden, til en gitt komponent. State kan bli definert som følger:

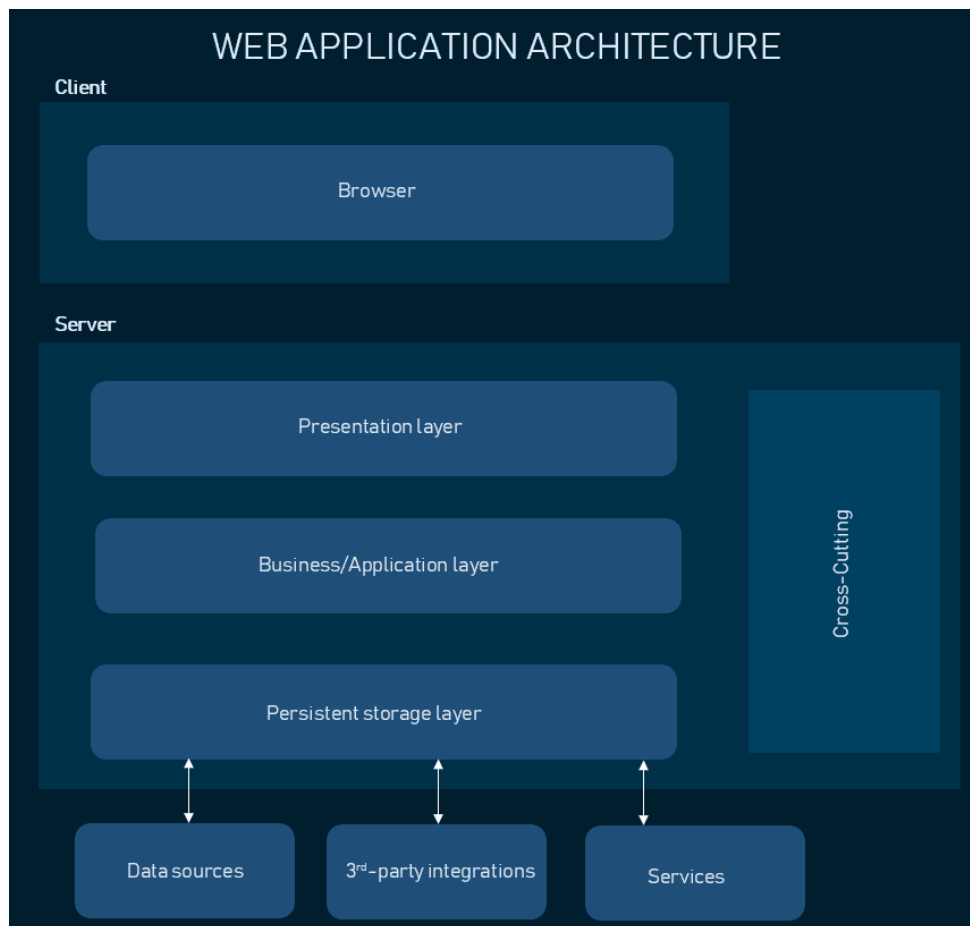
I informasjonsteknologi og informatikk blir et system beskrevet som tilstandsfull hvis det er designet for å huske tidligere hendelser eller brukerinteraksjoner; den huskede informasjonen kalles tilstanden til systemet. [Egen oversettelse] (State (computer science), 2020).

For eksempel kan hypotetiske komponent A sin synlighet være definert som at komponent A er synlig *hvis* brukeren er administrator og er innlogget.

Dersom avstanden mellom aktuelle data og komponentens plassering i hierarkiet er stor må en enten ty til «prop-drilling» (Samuel, 2020), altså sende data nedover i hierarkiet, eller finne en løsning som lar komponenten lese og skrive globale data. Prototypen i prosjektet benytter seg hyppig av betinget opptegnede komponenter på tvers av hierarkiet, noe som betyr at sistnevnte løsning vil resultere i mindre kode og mindre kompleksitet i implementeringsfasen for prosjektet. Det har derfor vært viktig å være påpasselig med å velge et rammeverk med fortrinnsvis innebygd funksjonalitet for global tilstandshåndtering, og som er lett å bruke.

### 5.1.2.2 Applikasjonsarkitektur

I kontekst av en webapplikasjon, kan applikasjonsarkitektur deles inn i presentasjonslaget, applikasjonslaget og vedvart lagring (Altexsoft 2019). Disse tre delene visualiseres i Figur 41.



Figur 41: Ulike deler av applikasjonsarkitektur (Altexsoft, 2019).

De ulike delene av applikasjonsarkitektur omtales som et arkitektonisk mønster, «3-lags mønsteret», av Altexsoft (2019). «3-lags mønsteret» må ikke nødvendigvis implementeres på måten vist i Figur 41, da en kan eksempelvis håndtere både applikasjonslaget og presentasjonslaget i klienten, altså nettleseren (Altexsoft, 2019), for eksempel ved hjelp av ensideapplikasjoner.

En ensideapplikasjon laster hele løsningen én gang, men kan implementeres på en måte som gir brukeren en illusjon om forskjellige sider (Ensideapplikasjon, 2021). Ensideapplikasjon flytter applikasjonslaget inn i nettleseren, noe som byr på en rekke utfordringer, men også fordeler. Tabell 15 er en oversikt over forskjellen på en flersideapplikasjon og en ensideapplikasjon fra Churylov (u.å.). Det er verdt å merke seg at deler av Tabell 15 kan diskuteres. For eksempel vil ikke en ensideapplikasjon automatisk gi en pen brukeropplevelse da dette avhenger av flere faktorer som design og hastighet på siden.



Hensyn	Ensidessaplikasjon	Flersidersaplikasjon
Pen brukeropplevelse	x	
Enkel å optimalisere for SEO		x
Sikkerhet		x
Mindre belastning på tjener	x	
Offline funksjonalitet	x	
Tilpasning for mobil	x	
Skalerbarhet		x
Seperasjon av data og grensesnitt	x	
Hastighet	x	
Hurtig sidelasting		x
Fungerer uten Javascript		x

Tabell 15: Sammenligning av enside- og flersideaplikasjon (Churylov, u.å.).

Ensideaplikasjoner tilrettelegger blant annet for interaktive tjenester og responsiv brukeropplevelse (Churylov, u.å.). Én potensiell forklaring på dette er at en ikke må etterspørre ny markup (struktur i form av for eksempel HTML og CSS) fra tjeneren hver gang brukeren navigerer til et nytt sted da kun deler av siden må oppdateres. Disse er som regel enklere å tilpasse for mobil, fortrinnsvis av samme grunn (Churylov, u.å.).

Om man velger å gå for en ensideaplikasjon må det understrekes at en kan fortsatt sende deler av presentasjonslaget fra tjenersiden. Det å fortsatt sende deler av presentasjonslaget fra tjener-siden kan være ønskelig av flere grunner. Blant annet påpeker Vue (u.å.b) at Google indekserer nettsider synkront, som vil si at eventuelle asynkrone data som blir hentet fra tjeneren etter siden har lastet ikke vil bli tatt hensyn til. Å hente data asynkront er en teknikk man er avhengig av i moderne ensideaplikasjoner. En konsekvens av å flytte presentasjonslaget inn i nettleseren, slik som ensideaplikasjoner gjør, er svakere søkemotoroptimalisering. Vue (u.å.b) påpeker at å sende markup fra tjeneren gir raske «time-to-content», altså tiden det tar før brukeren får se innholdet.

### 5.1.2.3 Vedvart lagring (databasesystem)

Med vedvart lagring menes lagrede data som ikke forsvinner når strømmen forsvinner (Persistent Storage, 2020). Det finnes flere databasesystemer som lar utviklere lagre data. Gitt at godt dokumenterte og ivarettatte løsninger er viktig for utviklingsløpet, har prosjektet tatt stilling til utbredte ikke-relasjonelle og relasjonelle databasesystemer. Sangeeth Raj (2020) definerer forskjellen som følger:

**SQL** (Structured Query Language) er et programmeringsspråk som brukes til å administrere data i relasjoner (vanligvis kalt tabeller) for å lagre data slik at det kalles en relasjonsdatabase [Egen oversettelse].

En **NoSQL**-database kalles en ikke-relasjonell database. På en annen side krever det ikke noe skjema, det er selvbeskrivende [Egen oversettelse].

Ifølge Raj (2020) er ikke-relasjonelle databaser skjemaløse. Det betyr at formen på dataen som skal legges inn, ikke må tilpasses en forhåndsdefinert struktur. Dette er gunstig med hensyn til uforutsette ønsker om funksjonalitet i fremtiden, og er derfor relevant i kontekst av å lage en MVP. Ulempen er at syntaksen for å kjøre spørringer mot slike systemer ikke er universelle, og varierer derfor fra system til system.

Prosjektet har også måtte ta hensyn til hvilke datamodeller prototypen impliserer. For eksempel hvis løsningen har en overvekt av visninger hvor brukeren leser data, bør man legge dette til grunn. Leung og Zhou (2016, s. 285) har målt ytelsen til et utvalg utbredte databasesystemer, og konkluderer med at de ikke-relasjonelle databasesystemene gjør det bedre på lesespørringer enn relasjonelle.

#### 5.1.2.4 Skyarkitektur og tjenere

Med hensyn til oppgavens omfang ble det besluttet å ikke konfigurere eller administrere egne fysiske tjenere. Prosjektet har derfor tatt stilling til hvilke skyarkitekturer som kan tilby en optimal balanse mellom handlingsrom og brukervennlighet med hensikt om å legge til rette for en hurtig utviklingsprosess. Spoiala (2015) [egen oversettelse] omtaler skytjenester som enten:

- Infrastruktur som en tjeneste (IaaS)
- Plattform som en tjeneste (PaaS)
- Software som en tjeneste (SaaS)
- Backend som en tjeneste (BaaS)

Erfaringer fra Webprosjekt II (se kapittel 3.1.1) tilsa at å leie infrastruktur på tjenersiden (Amazon web services) fremfor å eksempelvis leie backend som en tjeneste, medførte et betraktelig høyere koordineringsbehov, da utgangspunktet for AWS (Amazon web services) er mindre abstrahert og krever mer konfigurering og vedlikehold av utviklingsmiljø. Det gikk dessuten med tid til å bygge ut tjenersiden for å betjene grensesnittet med data, autentisering, og lignende. Dette er tid som ellers kunne blitt brukt på eksempelvis innsiktsinnsamling, eller prototyping.

I den anledning har det blitt spesifikt vurdert tjenester som kan defineres som BaaS (backend-as-a-service), da grensesnittet må kunne implementeres i henhold til prototypen, men tjenersidefunksjonalitet er fortsatt nødvendig per klient-tjener-modellen. Batschinski (u.å.) har en omfattende oversikt som tar for seg tjenestetilbydere av BaaS. Det er lite hensiktsmessig å liste hele oversikten her, men hvis prosjektet skal imøtekomme programvarearkitekturen som henviser til tidligere (REST), må tjenesten støtte dette. Prosjektet er også avhengig av en skytjeneste som kan støtte det databasesystemet som er ønskelig.

### 5.1.3 Prosjektets teknologistabel

I kapittel 5.1.2 ble det tatt en rekke hensyn som har resultert i en endelig teknologistabel. Tabell 16 er en oppsummering av forutsetninger som preger prosjektet og beslutninger for teknologi som er tatt.

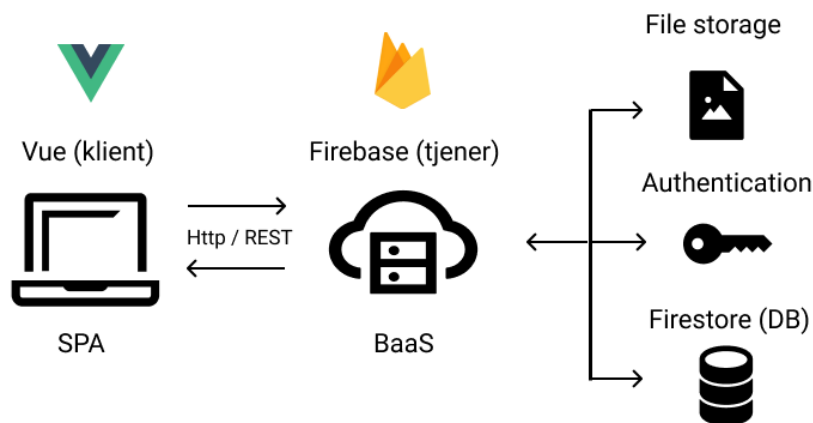
Hensyn	Beslutning	Teknologi
Organisering av kode for grensesnitt	Komponent-drevet utvikling	Vue.js
Applikasjonsarkitektur	Ensidesapplikasjon + backend-as-a-service	Vue.js + Firebase
Rendering	Klient-siden	Vue.js
Datalagring	Ikke-relasjonelt	Firestore

Tabell 16: Prosjektets teknologistabel

Prototypen og designheuristikker legger hyppig tilbakemelding til grunn, noe som inngår i beslutningsgrunnlaget om at en ensideapplikasjon er ønskelig. I tabellen ovenfor, kan leseren se at prosjektgruppen har valgt å tegne opp – eller rendere - komponenter i klienten, da det er tilstrekkelig for MVP-funksjonalitet i dette prosjektet, som beskrevet i starten av kapittel 6. Bekymringer om SEO må altså vike til fordel for en utviklingsprosess med minst mulig behov for koordinering på tvers av teknologistabelen, samt øvrige hensyn som omtalt i kapittel 6.1.2. Implementeringen er dog tilrettelagt for å gjøre det enkelt å eventuelt sende deler av presentasjonslaget fra tjener-siden ved hjelp av ytterligere teknologi, om dette skulle bli aktuelt i fremtiden. På bakgrunn av dette, har prosjektet tatt stilling til løsninger som kan:

1. Behandle data og forespørsler om data fra en klient som forklart i figur 38 om systemarkitektur.
2. Gjøre dette i henhold til REST som definert i kapittel 5.1.1 om programvarearkitektur.

Tatt i betraktning sammen med mulige teknologier utforsket i kapittel 5.1.2.4, benytter prosjektet seg av Vue.js og Firebase, med særlig fokus på assorterte tjenester tilbydd av firebase, som vist i Figur 42. Vue.js blir benyttet til frontend, mens Firebase blir benyttet til backend.



Figur 42: Infrastruktur for løsningen

### 5.1.3.1 Frontend

Valg av hvilket rammeverk som anvendes i prosjektet er basert på prosjektgruppens forkunnskap til og den eventuelle læringskurven knyttet til teknologien. Valget falt derfor på Vue.js. Dette påvirket grad av autonomitet i utviklingsprosessen, og tillot dermed mer effektivitet i utviklingen.

Som teorien om «Mobil først»-tilnærmingen i kapittel 3.2.1 tok for seg, er det svært viktig å tilrettelegge for brukere som ønsker å besøke løsningen på mobil. Dette gjøres blant annet ved å sørge for et responsivt design, hvorav man kan benytte seg av et CSS-rammeverk. Bootstrap er et av verdens mest populære frontend CSS bibliotek (BootstrapVue, u.å.). I tillegg kan Bootstrap integreres med Vue. Man er ikke avhengig av å bruke et rammeverk for å bygge en responsiv nettside, men det bidrar til å spare tid ved å unngå repeterende kode, i tillegg til å opprettholde en konsensus mellom de forskjellige medlemmene på samme prosjekt (Thapliyal, 2018). Prosjektet har hovedsakelig benyttet seg av det responsive rutenettet Bootstrap tilbyr, samt oppsett for meny og ikoner.

Sass er et skriptspråk som utvider CSS og tillater en mer kompleks syntaks som blir compilert til standard CSS (Coron, 2020). Blant annet tillater Sass bruk av variabler, noe som effektiviserer endringer av farger, fonter, skriftstørrelse, og lignende i etterkant. Ved å bruke variabler blir det lettere å unngå inkonsistente verdier for CSS-velgerne. I tillegg er det mulig å «neste» CSS-velgere inni hverandre. Dette er gunstig for å kunne følge det samme hierarkiet som i HTML, og det kan gjøre det lettere med vedlikehold. Samtidig kan overkomplisert «nesting» ha motsatt effekt, og det er derfor viktig å finne en balansegang (Sass, u.å.). Sass ble ansett som godt egnet for prosjektet, da i all hovedsak grunnet variabler og «nesting».

### 5.1.3.2 Backend

Backend er den delen av løsningen hvor tunge handlinger finner sted, som å skrive til og lese fra en database (Granevang, 2020a). I dette prosjektet blir samlingen av tjenester fra Firebase brukt for backend, som igjen er en del av Google Cloud. Firebase sin samling av tjenester er høyst relevant for prosjekts systemkrav og enkel å samhandle med, deriblant Cloud Firestore, Firebase authentication og Firebase storage. I tillegg befinner Google Cloud-regionen prosjektet benytter seg av i Frankfurt, som er en region hvor 61% av strømforbruket er karbonfri energi (CFE) (Google Cloud, 2021). Dette er et tiltak for å forminske karbonavtrykk relatert til internett. Firebase tilbyr dessuten et «software development kit» (videre henvist til som SDK) som tilrettelegger for å kjøre spørringer, onboarding av nye brukere, autentisering og lignende (Firebase, u.å.b).

### 5.1.3.3 Database

Løsningen impliserer at brukere kommer til å lese data fra databasen oftere enn de kommer til å skrive til databasen. Dette fremgår av alle visningene i grensesnittet hvor brukeren ikke har mulighet til å skrive eller endre data, men har mulighet til å lese de. Et NoSQL (ikke-relasjonelt) databasesystem vil kunne gi raskere lesespørringer, ifølge Leung & Zhou (2016, s. 285).

Firestore er et av tilbudene som blir tilbudt av Firebase, og som møter kriteriene om en NoSQL-database. Mer spesifikt er Firestore dokumentdrevet og tilbyr derfor komplekse datamodeller (Firebase, u.å.a). Dette gjør det lettere å lagre sammensatte data om eksempelvis en utlysning og dens søkere. Skjemaløse databasesystemer tillater dessuten større frihet, skulle man ha behov for å bygge ut ny funksjonalitet i fremtiden, da man ikke må omstrukturere eksisterende data for å legge til nye strukturer (GeeksForGeeks, 2020).

## 5.2 Implementering

Implementere betyr å iverksette, utføre eller realisere (Nilstun, 2020). I implementeringen har prosjektgruppen oversatt funksjonaliteten spesifisert i MVP-kartet til fungerende kode i samsvar med teknologistabelen. Kildekoden til BeNet finnes i vedlegg 8. Det er ytterligere hensyn og begrensinger som inngår i implementeringen, blant annet WCAG-kriterier, effektbruk og modularitet.

### 5.2.1 Frontend

Frontend er den delen av en løsning som brukeren ser og interagerer med (Granevang, 2020b). Estetiske valg av farger og skrifttyper i BeNet ble valgt etter NTNUs egen grafiske profil. NTNU har et krav om at alle produkter som blir produsert for NTNU skal ha innslag av «NTNU-blå», så lenge designet ikke baserer seg på nyanser fra gråtoneskalaen (NTNU, u.å.a). I tillegg ble det, med bakgrunn i teorien om dark mode (se kapittel 3.2.2.3), besluttet å implementere funksjonalitet for å bytte mellom light og dark mode. Typografisk fontvalg for all tekst blir fonten «Open Sans» brukt, som er NTNUs hovedskrifttype (NTNU, u.å.c). Fordi «Open Sans» er en sans serif-font, samt at den er luftig og har god X-høyde, egner den seg godt for lesbarhet på nett (NTNU, u.å.c). For løpende tekst, samt input-felt for skjemaer, er skriften satt til å være 1rem, altså 16px, som er et godt utgangspunkt for å sikre lesbarhet for nettstedet med mye tekst (Kennedy, 2020). Løsningen ble sjekket for brudd på WCAG 2.2 ved hjelp av verktøyet Color Contrast Analyser<sup>2</sup> og utvidelsen Siteimprove<sup>3</sup> for Google Chrome. Alle illustrasjoner og logoer i løsningen er lastet ned gratis fra Vecteezy<sup>4</sup> (se vedlegg 5), for å unngå å bruke ekte bedriftsnavn og logoer, samt bilder av ekte personer.

#### 5.2.1.1 Koordinering av komponenter

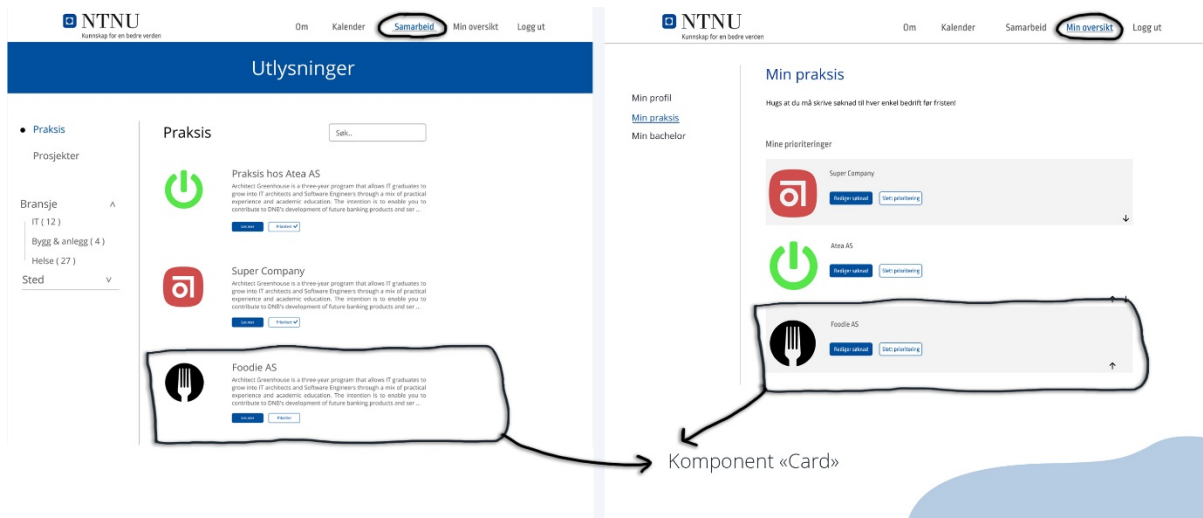
Forutsett at funksjonalitet som er nødvendig for å utvikle en MVP er fastslått, er det mulig å identifisere hvilke deler av løsningen som kan bygges ut som komponenter. Ved gjennomgang av den siste iterasjonen av prototypen (se kapittel 4.2.4 for detaljer om prototypen) ble det laget en oversikt over opplagte komponenter som var nødvendige for å utvikle en MVP. En sammenligning av de ulike sidene viste hva som blir brukt flere ganger, og hvor. Figur 43 viser et eksempel på hvordan komponenten «Card» ble identifisert, som fastslo at dette elementet var noe som ble brukt opp både flere ganger på samme sted, men også andre steder i løsningen, for eksempel på «samarbeid» og i «min oversikt».

---

<sup>2</sup> <https://www.tpgi.com/color-contrast-checker/>

<sup>3</sup> <https://siteimprove.com/nb-no/core-platform/integrations/browser-extensions/>

<sup>4</sup> <https://www.vecteezy.com/>



Figur 43: Skjerm bilde fra prototypen med eksempel på identifisering av komponent «Card».

«Card» har som funksjon å presentere de ulike brukergruppene med en effektiv hurtigvisning av ulike utlysninger. Dette formatet hviler på etablerte konvensjoner for hurtigvisning av informasjon av lignende karakter (Downs, 2019), og er et element som det ikke er ønskelig å utforme flere ganger da det utretter samme handling, men i en annen kontekst.

Ved å ta utgangspunkt i prosedyren vist i Figur 43 ble 11 komponenter identifisert fra prototypen som til sammen er med på å bygge brukergrensesnittet. Samtidig var det 9 komponenter som ikke umiddelbart var synlig i prototypen, men som ble ansett som nødvendig underveis i utviklingen. Et eksempel på en komponent som ble oppdaget i løpet av implementeringen er EditorWrapper.vue, som tilrettelegger for betinget visning for en annen komponent. Totalt sett benytter løsningen seg av 20 komponenter. Se Tabell 17 for liste over alle komponentene og hvordan de fungerer.

### Komponenter identifisert fra prototypen (11)

Komponentnavn	Funksjon
<b>About.vue</b>	Komponent som blir brukt opp på profiler.
<b>Applicant.vue</b>	Studenter som har søkt på praksisplass og som ikke er prioritert.
<b>Card.vue</b>	Fremstilling av utlysninger på «samarbeid».
<b>ContactColumn.vue</b>	Kolonnen som viser kontaktinformasjon om bedrift eller student.
<b>Footer.vue</b>	Bli brukt for bunntekst.
<b>ImgUpload.vue</b>	Laste opp bilder når man registrerer seg som bedrift.
<b>MatrixMenu.vue</b>	Fremstiller komponenter basert på overskrifter.

<b>Modal.vue</b>	Bakgrunn og overleggsboks som nøster andre komponenter (pop-up).
<b>Priorities.vue</b>	Liste med prioriteringer for studenter.
<b>SideMenu.vue</b>	Sidemeny som blir brukt på alle undersider.
<b>SiteNav.vue</b>	Hovedmeny/navigasjon.
<b>Komponenter som kom underveis (9)</b>	
<b>Card2.vue</b>	Fremstilling av bedriftens egne utlysning som vises på bedriftens profil.
<b>Contract.vue</b>	Bli brukt for å formidle tildelt praksisplass for studenter.
<b>DarkMode.vue</b>	Brukes for å skifte mellom light mode og dark mode.
<b>Editor.vue</b>	Markup for utfyllingsskjema.
<b>EditorWrapper.vue</b>	Betinget visning av Editor.vue komponenten.
<b>Match.vue</b>	Visualisering av matchet bedrift og student.
<b>MatrixMenu2.vue</b>	Lik som MatrixMenu.vue, men mer abstrakt.
<b>PureCard.vue</b>	Bli brukt for nye avtaler for administrator.
<b>SelectedApplicant.vue</b>	Prioritert student av bedrift.

Tabell 17: Oversikt over komponenter og deres funksjoner.

Webløsningen benytter seg av gjenbruk der det lar seg gjøre. Det vil si at det kan være web-elementer, eller samlinger av web-elementer som det ikke er ønskelig å utforme flere ganger da de utretter samme handling, men i en annen kontekst. Enkelte av komponentene i Tabell 17 kan virke redundante, eksempelvis card.vue og card2.vue. Dette skyldes et raskt utviklingsløp, hvor prosjektgruppen har måtte inngå kompromiss mellom utviklingsplan og problemer som oppstår underveis. Slike komponenter må videre abstraheres ut i fremtidig utvikling, av hensyn til redundant kode.

Prosjektet benyttet seg av Vue sitt v-slot direktiv, altså direkte og unikt innhold til komponenten. Ellers, har prosjektgruppen formidlet data til underkomponenter ved bruk av «props», altså delte data en gitt underkomponent kan ta imot. Ytterligere definisjoner av v-slot og props er å finne i Vue sin dokumentasjon<sup>5</sup>.

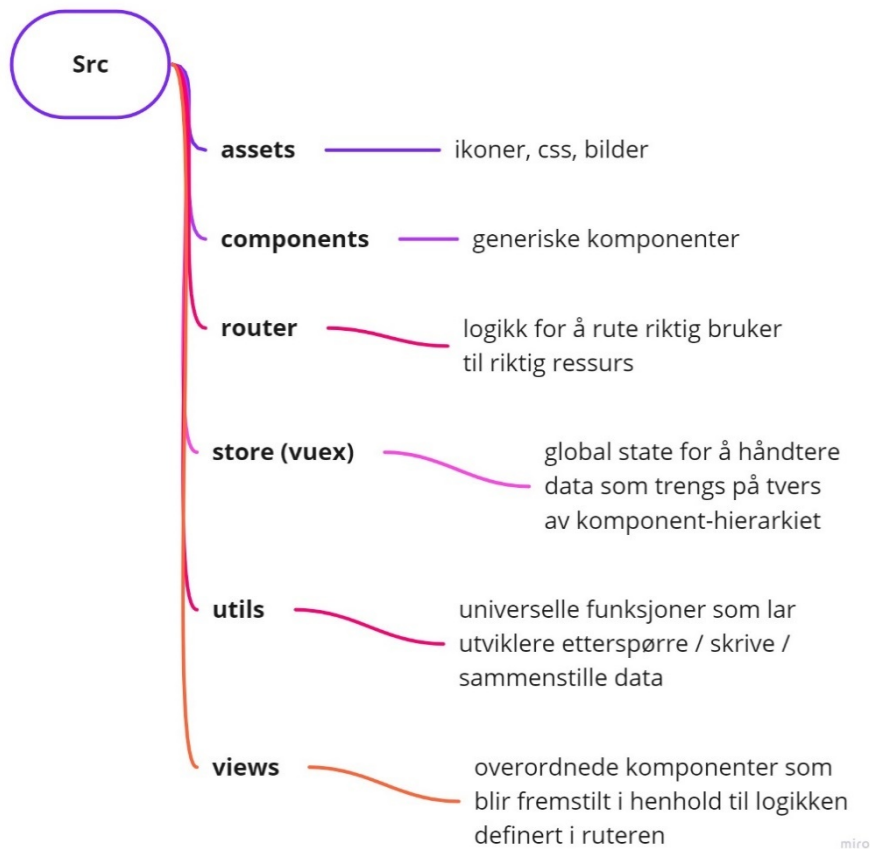
### 5.2.1.2 Mappedstruktur

Mappene i kildekode er organisert med hensikt om å tydelig skille mellom bruksområder. Frost (2013) foreslår å skille mer atomiske komponenter fra hele visninger – denne idéen har prosjektgruppen lagt til grunn for mappestrukturen. Eksempelvis vil en finne skjelettet for en gitt side

<sup>5</sup> Vue (u.å.a) *API*. Tilgjengelig fra: <https://vuejs.org/v2/api/> (Hentet: 08. mai 2021).



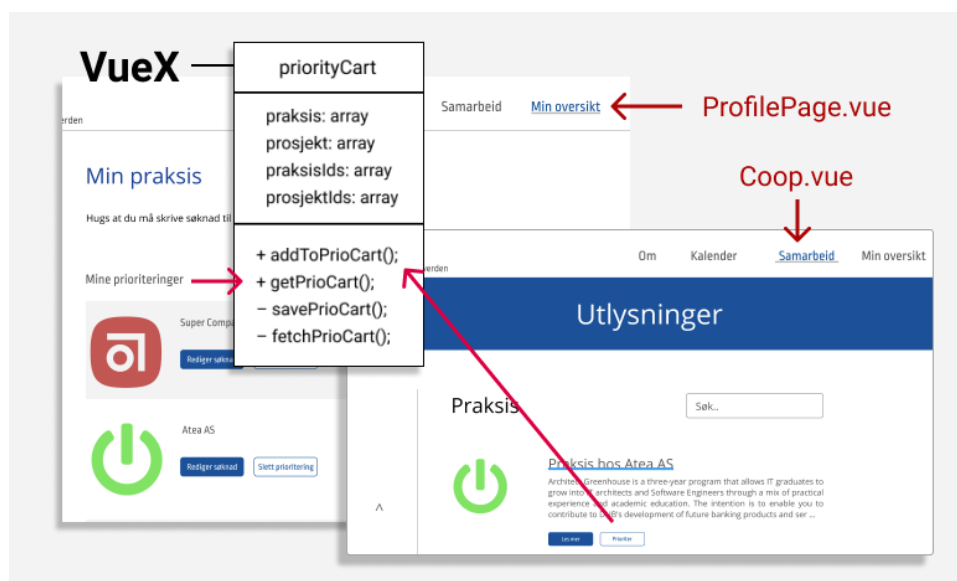
i mappen «views», mens underkomponenter befinner seg i «components». Logikken definert i «router» forholder seg henholdsvis utelukkende i visningene tegnet opp fra «views». Figur 44 gir et raskt inntrykk av strukturen.



Figur 44: Mappestruktur.

### 5.2.1.3 Vuex

I underkapittelet om komponent-dreven utvikling (kapittel 5.1.2.1), ble det understreket at prototypen forutsetter data som vil måtte deles på tvers av komponent-hierarkiet, avhengig av løsningens tilstand. Vuex lar utviklere gjøre nettopp dette. Eksempelvis legger studenter til ønsker i prioriteringslisten sin mens de blar gjennom utlysninger. Disse utlysningene vises i `Coop.vue`, men selve prioriteringen finner sted i `ProfilePage.vue`. Bare en av komponentene kan vises på samme tid og de eksisterer på ulike nivå i hierarkiet, men datastrukturen som disse komponentene leser fra og skriver til må være tilgjengelig for begge; derfor må disse dataene administreres av vuex. Denne datastrukturen kan enkelt formidles som en Unified Modeling Language (UML)-klasse med egne attributter og metoder, som vist i Figur 45.



Figur 45: Eksempel på hvordan ulike komponenter benytter samme data på fra ulike sider

## 5.2.2 Backend

Firebase sitt SDK (software development kit) inneholder en rekke metoder for å sende forespørsler i henhold til REST-mønsteret tidligere omtalt i kapittel 5.1.1. Disse metodene er abstraksjoner, og fullstendig dokumentasjon for nevnte metoder er å finne på Firebase sine nettsider (Firebase, u.å.b). Denne tilnærmingen ble brukt fordi disse abstraksjonene allerede står som implementeringer av «best practices» (Firebase, u.å.b).

Disse metodene er ytterligere konfigurert og dokumentert i en mappe kalt «utils» som ligger globalt i kildekoden, og kan dermed videre gjenbrukes i fremtidig utvikling. Her ligger det tre filer:

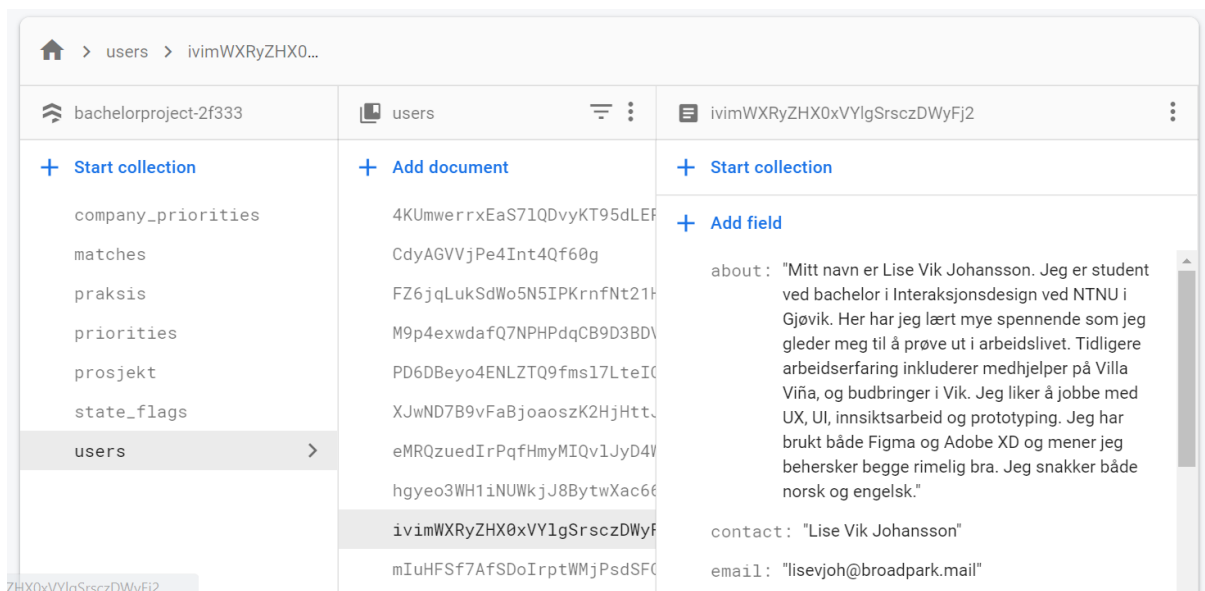
- get.js (metoder for å lese data)
- create.js (metoder for å skrive/endre data)
- gatekeeper.js (spesifikk logikk for å lese data på profilsidene)

Filen «gatekeeper.js» er en spesifikk hjelpefunksjon, da profilsidene er preget av hyppig kontekstendring. Man kan, hvis ønskelig, likevel implementere disse nettverksforespørslene selv (Firebase, u.å.c). I så fall vil endepunkter bli automatisk etablert av Firebase (disse benyttes for øvrig av de forhåndsdefinerte metodene), basert på en kombinasjon av prosjektets kredensialer samt navngitte samlinger og dokumenter.

Det vil være behov for automatisert e-post i potensiell fremtidig utvikling, gitt prosjektets definisjon av en MVP (se starten av kapittel 5). Fremtidig utvikling vil måtte ta stilling til prototypen som inkluderer en kalender hvor administrator kan sette en dato for når søknader kan sendes eller prioriteres og følgelig sende påminnelse over e-post. I utgangspunktet så prosjektgruppen for seg å bruke Cloud Functions (en tjeneste i firebase-suiten), men underveis i implementeringen ble det tydelig at dette problemet er best løst med kronologiske jobber<sup>6</sup>. Dette forutsetter funksjonalitet på tjenersiden som ikke avhenger av en innkommende nettverks-forespørsel.

### 5.2.3 Database

Databasen er organisert i samlinger og dokumenter, som vist i Figur 46. Et dokument utgjør en datastruktur som tillater å nøste ytterligere dokumenter. For dette prosjektet har mengden nøsting blitt minimert, da flate datastrukturer er bedre for skalerbarhet ifølge Firebase (2021).



Figur 46: Skjerm bilde av databasen.

Tabell 18 viser en oversikt over de ulike dokumenttypene i databasen, samt dokumentenes felter og deres datatyper.

<sup>6</sup> Cron, 2021. *Wikipedia*. Tilgjengelig fra: <https://en.wikipedia.org/wiki/Cron> (hentet: 12.mai, 2021)

<b>USERS</b>	Email: string (pålagt)
	Image_url: string (pålagt)
	Contact: string (pålagt)
	Name: string (pålagt)
	Phone: string (pålagt)
	Role: string (pålagt)
	Study: string
	Location: string
	Quali: array
	Wanted_work: string
<b>PROSJEKT &amp; PRAKSIS</b>	Approved: bool (pålagt)
	Contact: string (pålagt)
	Content: string (pålagt)
	Title: string (pålagt)
	Created_by: string (pålagt)
	Image_url: string (pålagt)
	Name: string (pålagt)
	Phone: string (pålagt)
	Spots: integer (pålagt)
	To: timestamp (pålagt)
	From: timestamp (pålagt)
	Tags: array
<b>PRIORITIES &amp; COMPANY_PRIORITES</b>	Praksis: array (pålagt)
	Prosjekt array (pålagt)
	PraksisIds: array (pålagt)
	ProsjektIds: array (pålagt)
<b>MATCHES</b>	Result: array (pålagt)
	Rest_students : array (pålagt)
	Date: timestamp (pålagt)
<b>STATE_FLAGS</b>	companies_can_prioritize: bool
	pairing_complete: bool

Tabell 18: Tabeller i databasen.

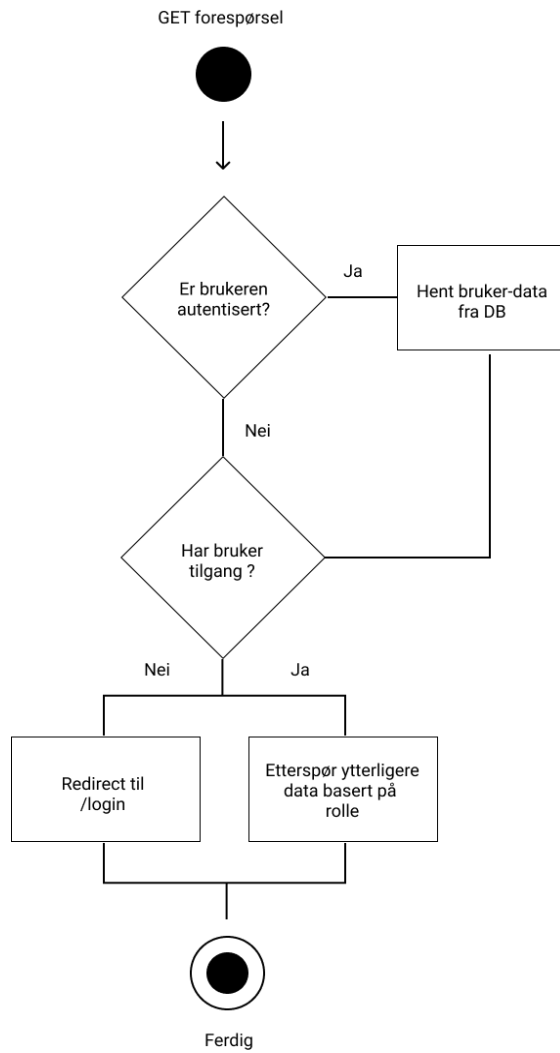
Det foreligger dessuten et spørsmål om karbonutslipp som per prosjektets andre forskningsspørsmål. Firestore er en produsenteid programvare – altså ikke åpen kildekode. Prosjektgruppen har dermed ikke anledning til å hverken måle effektbruk eller konfigurere maskinvare for å minimere effektbruk. Prosjektet har dog tatt ulike hensyn for å optimalisere datalagring på tvers av applikasjonen, som vist i Tabell 19.

Tiltak	Formål
<b>Oppdeling av samlinger etter bruksområde</b>	Redusert behov for å traversere og filtrere data i liste-visninger.
<b>Auto-genererte ID-er og gjenbruk av disse</b>	Å bruke et dokument ID som et annet dokument sin ID sparer oss før søk. Eksempelvis kan student «abc123» sine prioriteringer bli oppført som dokument «abc123» i en samling kalt «prioriteringer».
<b>Avgrensede spørringer</b>	Avgrensninger på resultat-settet fra en spørring i liste-visninger hvor brukeren ikke trenger alle resultatene
<b>Status-flagg</b>	Enkle felter i databasen som bestemmer videre spørringer. Alternativet er etterspørre all dataen man potensielt kan trenge, for så å drive betinget visning i klienten.
<b>Sletting av ubrukte data</b>	For eksempel slettes utlysninger som blir avslått av administrator, fremfor å inngå som en del av historikken

Tabell 19: Tiltak for optimalisering av datalagring

## 5.2.4 Autentisering

Løsningen har tre ulike brukergrupper: administratorer, studenter, og bedrifter. Disse har ulike tilgangsnivåer til BeNet som er beskyttet av passord. Det er derfor vesentlig at løsningen autentiserer brukere som forsøker å besøke sider de egentlig ikke skal ha tilgang til, for å unngå at informasjon blir tilgjengelig for uautoriserte brukere. Da løsningen er en ensidesapplikasjon, må brukere autentiseres hver gang nettleseren lander på en bestemt adresse. Visningen er ikke tegnet opp på tjenersiden, og det er derfor nødvendig med en måte å skjule og vise data ovenfor bestemte roller. Autentiseringen visualiseres på et konseptuelt nivå i Figur 47.



Figur 47: Autentiseringsflyt.

Vue har en rot-instans som instansieres når rot-fila lastes (index.html). Vilkårene for instansieringen skjer i main.js, hvor kildekoden først sender en forespørsel til Firebase for å få vite autentiseringsstatus for den enkelte brukeren (se Figur 48).

```

35   let app
36   auth.onAuthStateChanged(user => {
37     if (!app) {
38       app = new Vue({
39         router,
40         store,
41         render: h => h(App)
42       }).$mount('#app')
43     }
44
45     if (user) {
46       store.dispatch('fetchUserProfile', user)
47     }
48   })

```

Figur 48: Sjekk av innloggingsstatus

«Store» henviser her til Vuex, Vue sitt globale state-API, da denne dataen trengs for betinget opptegning av rot-komponenter, som nevnt i kapittel 5.1.2.2. I praksis betyr dette at den innebygde metoden fra Firebase sitt SDK først sjekker nettleserens minne etter gyldig autentiseringsbevis. Hvis denne ikke eksisterer, blir brukeren bedt om å logge inn for å uthente et nytt bevis og skriver det til nettleserens minne.

## 5.2.5 State/tilstand

BeNet er en plattform med ulike brukergrupper, som alle har tilgang til ulike oppgaver. BeNet fokuserer på prosessen som skjer før studentene skal ut til bedriftene. I denne prosessen er det flere frister som bestemmer løsningen state/tilstand. Hvilken tilstand løsningen er i avhenger av at visse oppgaver utføres før andre. BeNet opererer med tre ulike tilstander, se Tabell 20. Den første tilstanden er en tilstand som tillater studentene å prioritere bedrifter. For å unngå å havne i et scenario hvor en student endrer prioriteringer mens bedrifter prioriterer, må løsningen bytte til tilstand nummer to ved å «låse» prioriteringene til studentene og dermed åpne opp for at bedrifter kan prioritere. Den tredje tilstanden er etter at administrator har godkjent matchingen og student har fått tildelt praksisplass.

	TILSTAND 1	TILSTAND 2	TILSTAND 3
STUDENT	Se utlysninger Prioritere bedrifter Skrive søknader	Prioriteringer er låst og ingen handlinger kan utføres	Oversikt over hvilken praksisplass de har fått
BEDRIFT	Registrere bruker Utlyse plasser	Lese søknader Prioritere studenter	Oversikt over hvilke(n) student(er) de har fått
ADMIN	Foreslå endringer på utlysninger Godkjenne/Avslå	Godkjenne matching	Oversikt over historisk data

Tabell 20: Oversikt over funksjonalitet i de tre tilstandene.

Tabell 20 viser hva de tre brukergruppene kan forestå seg i hver tilstand. Optimalt sett burde disse tilstandene bestemmes av satte frister i en kalender, men da kalender ble ansett som «nice-to-have» (se kapittel 5) løser BeNet dette ved å tillate administrator manuelt å bytte mellom tilstander. Disse tilstandene er representert som binære felter i databasen: `Companies_can_prioritize` og `Pairing_complete`. Når bedrifter og studenter laster `ProfilePage.vue`, sjekkes disse feltene først, for at applikasjonen skal vite om `ProfilePage` skal tegne opp prioriterings-funksjonaliteten eller ikke.

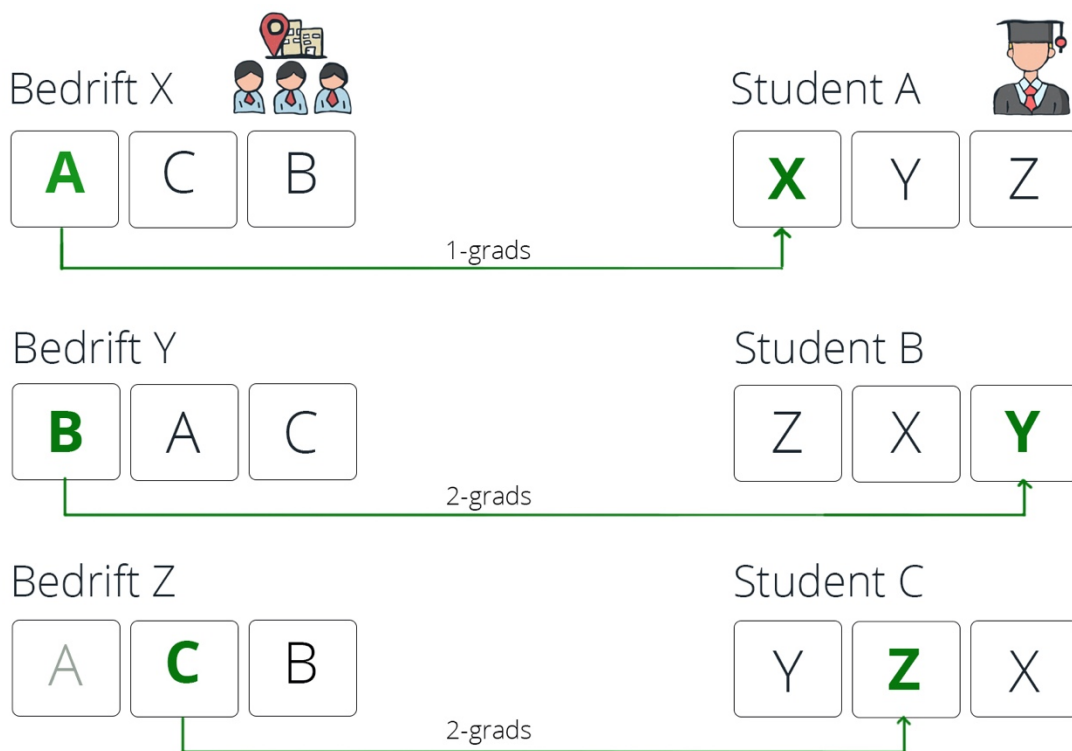
### 5.2.6 Automatisk sammenstilling basert på prioriteringer

I dagens ordning av praksisemnet er matching steg nr. 6 (se Figur 1 i kapittel 1.1) og del av problemområde 3 (se kapittel 1.1.1). Dette innebærer at oppdragsgiver manuelt må sette sammen prioriteringer fra studenter med prioriteringer fra bedrifter. Arbeidet er et tidkrevende manuelt arbeid som krever høy konsentrasjon, som BeNet løser ved å implementere en algoritme som utfører automatisk matching.

Inspirasjon til algoritmen i BeNet kommer fra Gale–Shapley algoritmen, en algoritme som har som formål å optimalisere matcher på tvers av to sett av samme størrelse. Avhengig av hvordan algoritmen brukes, «[...] kan den finne løsningen som er optimal enten for deltakerne på den ene siden av matchingen, eller for deltakerne på den andre siden» [Egen oversettelse] (Gale–Shapley algorithm, 2021). Forskjellen mellom BeNet og utgangspunktet for Gale–Shapley, er at BeNet har kommersielle hensyn å ta, samt at størrelsen på settet for utlysninger nødvendigvis ikke er lik størrelsen på settet av søkere. Med kommersielle hensyn menes at bedriftens prioriteringer tar presedens over studentens, men hvis to bedrifter B1 og B2 setter samme student S på samme prioritering, vil student S sin høyeste prioritering resultere i en match med enten B1 eller B2.



Den implementerte algoritmen i BeNet setter automatisk sammen 1-grads og 2-grads ønsker, hvor bedriftene har presedens. Denne prosessen er visualisert i Figur 49 nedenfor. 1-grads matching vil forekomme i et scenario hvor bedrift og student begge har prioritert hverandre på første plass. Dette fører til at disse automatisk vil bli låst til hverandre, og dermed fjernes fra samlingen av tilgjengelige bedrifter og studenter. Et unntak er om bedriften har flere tilgjengelige praksisplasser på samme utlysning, da vil bedriften fremdeles være tilgjengelig.



Figur 49: Illustrasjon over hvordan sammensetning av prioriteringer i BeNet fungerer.

2-grads matching forekommer i andre scenarioer hvor ikke både bedrift og student har prioritert hverandre på første plass, som vist i de to nederste tilfellene i Figur 49. Prosessen for hvordan algoritmen kjøres oppsummeres i stegene i Tabell 21.

STEG	BESKRIVELSE
1	Algoritmen henter den utlysningen som har flest plasser tilgjengelig på én utlysning. (maxSpot)
2	Det settes en teller ( $i$ ) som går fra 1 til maxSpot.
3	Rekkefølgen av utlysninger randomiseres
4	La den enkelte utlysningens plasser være $X_j$
5	Gitt at den enkelte utlysning har plass ( $X_j \geq i \leq \text{maxSpot}$ ): 5.1: Utlysningen sjekkes for 1-grads matcher – aktuelle student er nå reservert 5.2: Utlysningen sjekkes for 2-grads matcher – aktuelle studenter er nå reservert
6	Studenter og bedrifter som ikke ble tildelt noen matcher havner i en separat liste.

Tabell 21: Steg i sammensetning av prioriteringer.

Algoritmen fyller ikke opp utlysningens plasser med én enkelt iterasjon, fordi man ønsker å ta høyde for utlysninger som tilbyr flere plasser i én utlysning. Hvis en bedrift utlyser praktikantopphold med svært mange plasser, hadde den utlysningen fått tildelt alle studentene som hadde kvalifisert for en 2-grads match, med mindre man tar slike hensyn som forklart ovenfor.

## 5.3 Evaluering

Før prosjektet kunne bli ansett som ferdig, ble det gjennomført to ulike evalueringer, som begge krevde at det ble gjort endringer i webbløsningen. Den første evalueringen som ble utført var en ekspertevaluering, den andre var systemtest. De to ulike evalueringene vil bli gjort rede for sammen med resultatene i dette kapittelet.

### 5.3.1 Ekspertevaluering

En ekspertevaluering har som formål å la en ekspert utføre en evaluering av et produkt (Nordbø, 2017, s. 173). For at ekspertevalueringen skal være pålitelig er det viktig å hente inn en ekspert innenfor feltet som skal evalueres, som i dette tilfellet var interaksjonsdesign. Eksperten for evalueringen i dette prosjektet er en utdannet interaksjonsdesigner som nå jobber som tjenestedesigner.

Ekspertevalueringer blir utført ved to tilfeller: **a)** dersom produktet må evalueres og utviklingsteamet ikke har tilgang på brukere, eller **b)** dersom det er et behov for å evaluere et produkt basert på brudd på heuristikker eller designprinsipper (Nordbø, 2017, s. 173). Et av forskningsspørsmålene for dette

prosjektet stiller spørsmål ved hvordan uforme og utvikle en god brukeropplevelse gjennom prinsipper om design og universell utforming. Disse prinsippene omhandler blant annet heuristikker, som beskrevet i kapittel 3.2.2.2, og er noe som utviklingen i prosjektet har forholdt seg til. Eksperten utførte derfor **b)** en heuristisk evaluering av BeNet.

En heuristisk evaluering har som formål å evaluere brukskvalitet ved å følge heuristikker (Nordbø, 2017, s. 173). Heuristikker er tommelfingerregler for godt design, som utdypet i kapittel 3.2.2.2. Heuristikkene som blir brukt i en heuristisk evaluering kan være etablerte heuristikker, eller man kan lage egne heuristikker; det viktigste er at heuristikkene er relevante for produktet som skal evalueres (Nordbø, 2017, s. 173–175). For ekspertevalueringen av BeNet ble de veletablerte heuristikkene av Jakob Nielsen (2020) brukt, i tillegg til at det ble tatt høyde for kommentarer uavhengig av heuristikkene. Følgende heuristikker av Nielsen (2020), oversatt til norsk og gjengitt i Nordbø (2017, s. 174–175), ble brukt:

- Synlighet av systemstatus
- Sammenheng mellom system og virkelighet
- Konsistent design og standarder
- Unngå feil
- Fleksibilitet og effektivitet i bruk
- Estetikk og minimalistisk design
- Hjelp og dokumentasjon

Evalueringen ble utført ved at eksperten gjennomgikk løsningen i eget tempo og pekte ut brudd på heuristikker i en muntlig samtale, hvor bruddene ble notert. Eksperten skrev under på en samtykkeerklæring (se vedlegg 3 for innholdet i skjema) før evalueringen startet, og har dermed samtykket til at kommentarer kan gjengis. For å beholde anonymiteten ble ikke alder, navn, bosted, eller andre personidentifiserende opplysninger ført opp, annet enn utdanning og nåværende stilling. Årsaken til dette er at prosjektet ikke er meldt inn til Norsk Senter for Forskningsdata (u.å.), som sier at man ikke kan samle inn personidentifiserende data uten å melde prosjektet.

Ekspertevalueringen gav innsikt i hvordan designet på BeNet fulgte heuristikkene.

Tabell 22 viser resultatene, hvor kolonnen til venstre viser til spesifikk heuristikk, kolonnen i midten viser til alle brudd funnet av eksperten, og kolonnen til høyre viser til hvorvidt disse bruddene har blitt fikset. Den nederste raden inneholder generelle kommentarer som ikke er knyttet til en spesifikk heuristikk.

<b>Heuristikk</b>	<b>Kommentarer fra ekspert</b>	<b>Løst</b>
<b>Synlighet av systemstatus</b>	Lite tilbakemeldinger på hva man har foretatt seg i systemet. Man kan trykke på «prioriter» uten å vite om den har blitt lagt til eller ikke.	<b>Ja</b>
<b>Sammenheng mellom system og virkelighet</b>	Forståelig ordforbruk, men usikker på om ordet «samarbeid» er riktig merkelapp i menyen i dette tilfellet.	<b>Nei</b>
<b>Konsistent design og standarder</b>	Løsningen har både en «registrer»-side og en «logg inn»-side i menyen. Det er vanlig at disse er sammenslått, altså om man befinner seg på logg inn uten å ha noe å logge inn med, kan man bli videresendt til registrering derfra. Rediger profil burde ligge oppe til høyre så det ser ut som man kan redigere hele profilen, ikke bare kontaktkort.	<b>Ja</b>
<b>Unngå feil</b>	Få varsel for å bekrefte handling mangler i løsningen. Man kan slette ting ved uhell.	<b>Delvis</b>
<b>Fleksibilitet og effektivitet i bruk</b>	Lite «snarveier» i løsningen, det mangler flere innganger til viktige sider. Menyene er nesten utelukkende eneste navigasjonsmulighet, og derav mange klikk for å komme seg frem og tilbake.	<b>Ja</b>
<b>Estetikk og minimalistisk design</b>	Fine enkle illustrasjoner som ikke tar for mye oppmerksomhet, og løsningen inneholder ikke unødvendige elementer.	<b>Ja</b>
<b>Hjelp og dokumentasjon</b>	Fint at dere har en hjelp side, men om-siden med informasjon burde inneholde mer	<b>Ja</b>
<b>Andre kommentarer</b>	Brukeropplevelsen kan forbedres mye. Litt mange klikk for å navigere. Legg til flere innganger til viktige sider og følg konvensjoner.	

Tabell 22: Oversikt over brudd på heuristikkene av Nielsen (2020) i webløsningen BeNet.

Eksperten påpekte at det manglet flere tilbakemeldinger for å tydelig synliggjøre systemstatus, blant annet manglet det tilbakemelding på om en student har lagt til en praksisplass i sine prioriteringer etter å ha klikket på «prioriter». I denne og flere lignende hendelser hvor det er nyttig for brukeren å motta en tilbakemelding ble det lagt til dialogbokser som bekreftet eller avkreftet handlingen, for å informere vedkommende om handlingen var suksessfull eller ikke. Tilbakemeldinger i form av advarsel for å hindre brukeren i å gjøre feil manglet også, noe som spesielt kan bli vurdert som et

problem for administrator ved godkjenning og avslåing av nye utlysninger. Uten en advarsel kan administrator slette en ny avtale permanent fra systemet ved et uhell. Det ble derfor lagt til en dialogboks som ber om bekreftelse, samt informerer om at den nye avtalen vil bli permanent slettet fra systemet dersom den blir avslått. Denne tilnærmingen kan trolig også løse problemet som gjenstod av høy risikovurdering etter brukertest av andre iterasjon av prototypen, som omhandlet mangel på advarsel ved lukking av bachelorgruppe etter at det var gjort endringer, noe som vil føre til at gruppen ikke ble lagret (se kapittel 4.2.4). Dette er derimot noe som må bli brukertestet i en ny iterasjon av prototypen ved en senere utvidelse for å inkludere bacheloremnet i BeNet.

Det var ekspertens mening at ordbruken i BeNet i all hovedsak anvendte et forståelig ordforråd. Bruken av ordet «samarbeid» i navigasjonsmenyen var derimot usikkert, da det var mulig at «utlysninger» hadde vært et bedre og mer beskrivende ord for innholdet. Dette er derimot noe som bør testes i isolasjon før en endelig avgjørelse blir tatt, og har derfor ikke blitt endret ved gjennomføring av dette prosjektet.

Menyelementene «registrer» og «logg inn» har blitt sammenslått etter råd om å følge konvensjoner fra eksperten, slik at om en bruker ikke er logget inn vil menyen kun inneholde «logg inn». Fra denne siden blir man presentert med mulighet til å bli videresendt til registrering av ny bruker. Det er også mulig å komme seg tilbake til «logg inn» fra denne siden, noe som underbygger ekspertens mening om å legge til flere snarveier i løsningen. For å unngå at menyen er eneste navigasjonsmulighet har så kalte «snarveier» som dette blitt lagt til flere steder i løsningen, noen i form av knapper, andre i form av tekstlenker. I tillegg har «rediger profil»-knappen blitt flyttet ut fra kontaktkortet og opp i høyre hjørnet på brukerens profil, slik at det er tydelig at hele profilen kan redigeres.

Ekspertevalueringen avdekket brudd på heuristikkene «fleksibilitet og effektivitet i bruk» og «hjelp og dokumentasjon». Begge ble delvis rettet opp i. For «fleksibilitet og effektivitet i bruk» var problemet at det var få snarveier tilgjengelig i systemet, noe som senker brukerflyten og effektiviteten ved samhandling med BeNet. Snarveier ble derfor lagt til på landingssiden for å logge inn i systemet, eller til «samarbeid» dersom en bruker er logget inn. I tillegg ble det lagt til snarvei på bedriftens «vår praksis»-side for å opprette ny utlysning for bedrifter som ikke har noen utlysninger som er godkjente eller avventer godkjenning. Disse snarveiene har som formål å forbedre brukerflyten, men også effektiviteten. Bruddet på «hjelp og dokumentasjon» omhandlet for lite informasjon på landingssiden («om»-siden). Løsningen her ble en mer generisk tekst, samtidig som flere snarveier bidro til at landingssiden ble mer interaktiv og dermed mer hjelpsom da brukerne får mer hjelp til hva neste steg er og hvor de finner det, noe som gav landingssiden en større betydning for gjennom større deler av prosessen.

### 5.3.2 Systemtest

Systemtesting bør planlegges etter hvert som en løsning implementeres (Nordbø, 2017, s. 178). Denne typen testing er noe ulik fra evaluering i form av for eksempel brukertesting som tidligere utført i utformingsfasen. Systemtest skal avdekke om all funksjonalitet fungerer som tenkt, og utføres når implementeringen er ferdig (Nordbø, 2017, s. 179). En systemtest følger et testscript, som skal bestå av de oppgavene det skal være mulig å gjøre i løsningen, og ifølge Nordbø (2017, s. 178–179) kan en ta i bruk brukerhistorier som et grunnlag til et testscript. Målet med systemtesten var å kunne utføre testen uten å finne feil som hindret brukeren i å oppfylle akseptansekriteriene. Dersom en bruker møtte på feil ved gjennomføring av testen, ble det gjort endringer før neste systemtest ble utført.

Testscriptet for systemtesten baserer seg på brukerhistoriene utformet i utformingsfasen (se kapittel 4.2.1). Testscriptet inneholder alle akseptansekriteriene fra brukerhistoriene for alle tre brukergruppene, hvilke instruksjoner testbrukerne får, og det forventede resultatet. Fremfor å følge et scenario, fikk testbrukerne instruksjoner, på en slik måte at de ble fortalt hva de skulle gjøre og hvor de skulle trykke. Av den grunn ble det ikke ansett som nødvendig å teste på personer som faller innenfor brukergruppen. I tillegg var det ønskelig å gjennomføre testen fysisk. Testbrukerne er derfor nærkontakter av prosjektgruppen. Systemtesten måler ikke grad av gjennomføring eller hvor lang tid en oppgave tar, den måler kun om funksjonaliteten som testes fungerer som den skal. I tillegg til oppgaver og akseptansekriterier inneholder testscriptet hvilke plattformer og hvilke versjoner av nettlesere det testes på. Se vedlegg 6 for detaljer og resultater.

Systemtesten hadde som nevnt formål om å oppdage feil ved funksjonalitet beskrevet i akseptansekriteriene. Det var derimot forventet at akseptansekriterier som omhandler automatisk vasling med e-post ikke ville oppfylles, da funksjonaliteten for dette ble vurdert som «nice-to-have» før implementeringen startet (se kapittel 5), og derfor ikke har blitt implementert i dagens MVP. Etter 3 runder var alle akseptansekriteriene som er del av en MVP gjennomførbare.

Tabell 23 viser resultatene fra de tre testene for brukergruppen student. For student oppstod det problemer med 50% av akseptansekriteriene ved gjennomføring av første test i både Google Chrome, Microsoft Edge, og iPhone X. Etter andre test var kun 70% av kriteriene gjennomførbare for iPhone X. Systemtesten for brukergruppe student ble etter tredje test godkjent, da all funksjonalitet for alle akseptansekriteriene var mulig å gjennomføre på de tre plattformene.

**STUDENT – 10 AKSPETANSEKRITERER**

	Google Chrome	Microsoft Edge	iPhone X
<b>FEIL I TEST 1</b>	5	5	5
<b>FEIL I TEST 2</b>	0	0	3
<b>FEIL I TEST 3</b>	0	0	0

*Tabell 23: Resultat av systemtest for brukergruppe student.*

Tabell 24 viser resultatene fra de tre testene for brukergruppen bedrift. For bedrift oppstod det problemer med 27% av akseptansekriteriene ved gjennomføring av første test i både Google Chrome og Microsoft Edge, i tillegg til problemer med 64% av kriteriene for iPhone X. Ved andre test gjenstod det problemer med 45% av kriteriene for iPhone X. Brukergruppen bedrift innehar dog ett akseptansekriterie som utgjør 9% av problemene, som ikke er oppfylt etter hverken andre eller tredje test for noen av plattformene. Dette kriteriet omhandler dog varsel i form av e-post, som tidligere nevnt ikke er en del av dagens MVP, og testen ble ansett som godkjent.

**BEDRIFT – 11 AKSPETANSEKRITERER**

	Google Chrome	Microsoft Edge	iPhone X
<b>FEIL I TEST 1</b>	3	3	7
<b>FEIL I TEST 2</b>	1	1	5
<b>FEIL I TEST 3</b>	1	1	1

*Tabell 24: Resultat av systemtest for brukergruppe bedrift.*

Tabell 25 viser resultatene fra de tre testene for brukergruppen administrator. For administrator oppstod det problemer ved 22% av akseptansekriteriene ved gjennomføring av første test i alle de tre plattformene. For iPhone X gjenstod det problemer ved 22% av kriteriene etter andre test. Ved tredje test gjenstod det problemer ved 11% av kriteriene på alle de tre plattformene. Brukergruppen administrator har derimot ett akseptansekriterie som utgjør 11% av problemene som omhandler varsling gjennom e-post, noe som ikke er en del av dagens MVP. Etter tredje test ble systemtesten derfor vurdert som godkjent.

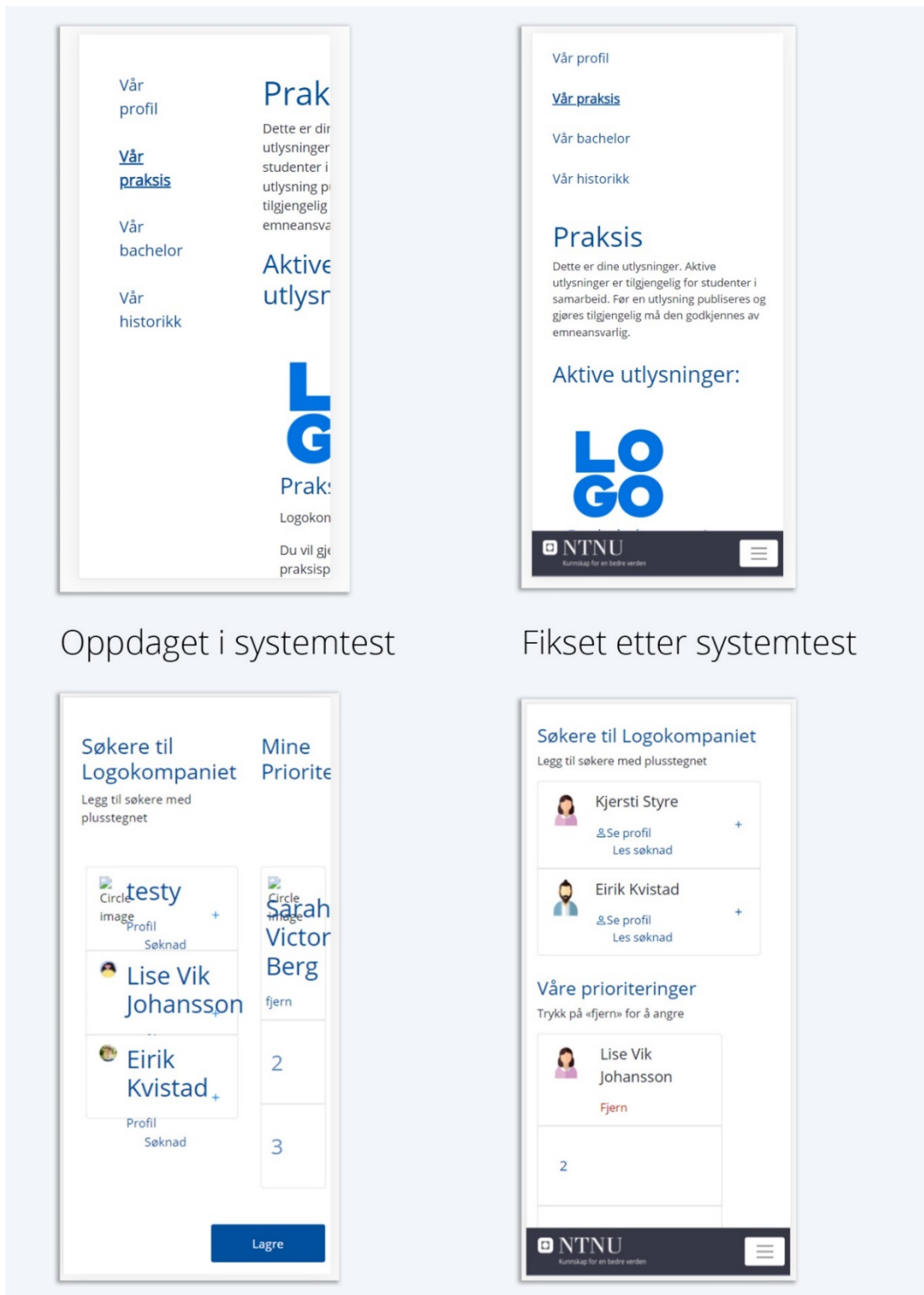
**ADMINISTRATOR – 9 AKSPETANSEKRITERER**

	Google Chrome	Microsoft Edge	iPhone X
<b>FEIL I TEST 1</b>	2	2	2
<b>FEIL I TEST 2</b>	1	1	2
<b>FEIL I TEST 3</b>	1	1	1

*Tabell 25: Resultat av systemtest for brukergruppe administrator.*

Etter å ha gjennomført systemtesten 3 ganger fikk alle akseptansekriteriene for brukergruppene godkjent, bortsett fra de som omhandlet e-post. Systemtesten testet også hvorvidt løsningen fungerer i mobilvisning, da det er viktig å tilrettelegge for brukere som ønsker å ta i bruk løsningen på mobil, som beskrevet i kapittel 3. Design og utseende i denne visningen er ikke det viktigste, det viktigste er at løsningen er funksjonell. De to første systemtestene gjorde det klart at løsningen ikke kunne oppfylle alle akseptansekriteriene på mobil. Akseptansekriterier som involverte profiler kunne ikke oppfylles da disse sidene ikke var responsive, som førte til at innhold ble liggende utenfor den mobile skjermen og at menyen forsvant. Figur 50 viser problemene da de ble oppdaget og etter at de ble rettet opp i.





Opplaget i systemtest

Fikset etter systemtest

Figur 50: Mobilversjon før (til venstre) og etter (til høyre).

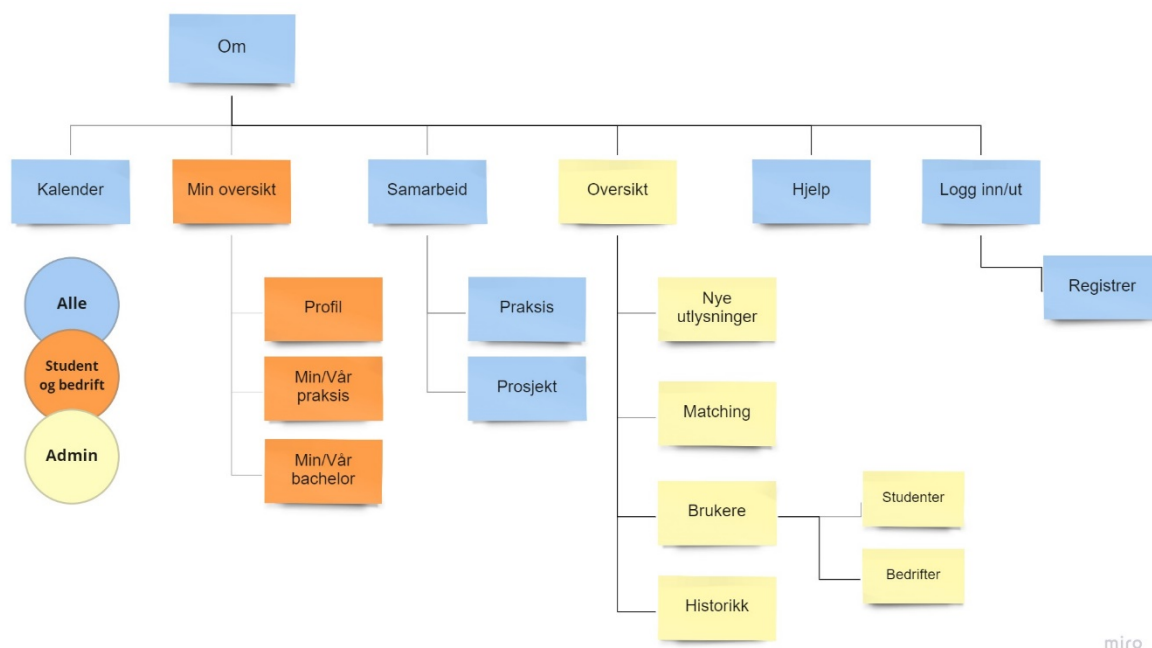
Hele systemtesten vurderes til godkjent da det var mulig å gjennomføre alle funksjonaliteter for MVP uten problemer i både Google Chrome, Microsoft Edge, og mobilvisning for iPhone X.

## 6 Webløsningen «BeNet»

Målet med prosjektet var å utvikle en MVP som forbedrer håndtering av praksisemnet ved Insitutt for Design. Gjennom implementasjon av funksjonalitet som var nødvendig for en MVP, ble resultatet webløsningen BeNet, som fokuserer på prosessen som skjer *før* studentene starter sin praksisperiode ute hos bedriftene. Dette er den delen av prosessen som krever mest av alle de tre brukergruppene. Etter at praksis har startet, forholder studenten seg i all hovedsak til praksisstedet (bedriften), og bedriften styrer sitt eget løp i samarbeid med studenten. Løsningen legger dermed ikke til rette for funksjonalitet etter at en student har blitt tildelt en praksisplass, som for eksempel loggføring, foruten om historikk og oversikt for administrator. Løsningen kan besøkes gjennom følgende link: <https://bachelorproject-2f333.web.app/>. Innloggingsinformasjon finnes i brukergruiden i kapittel 6.2.

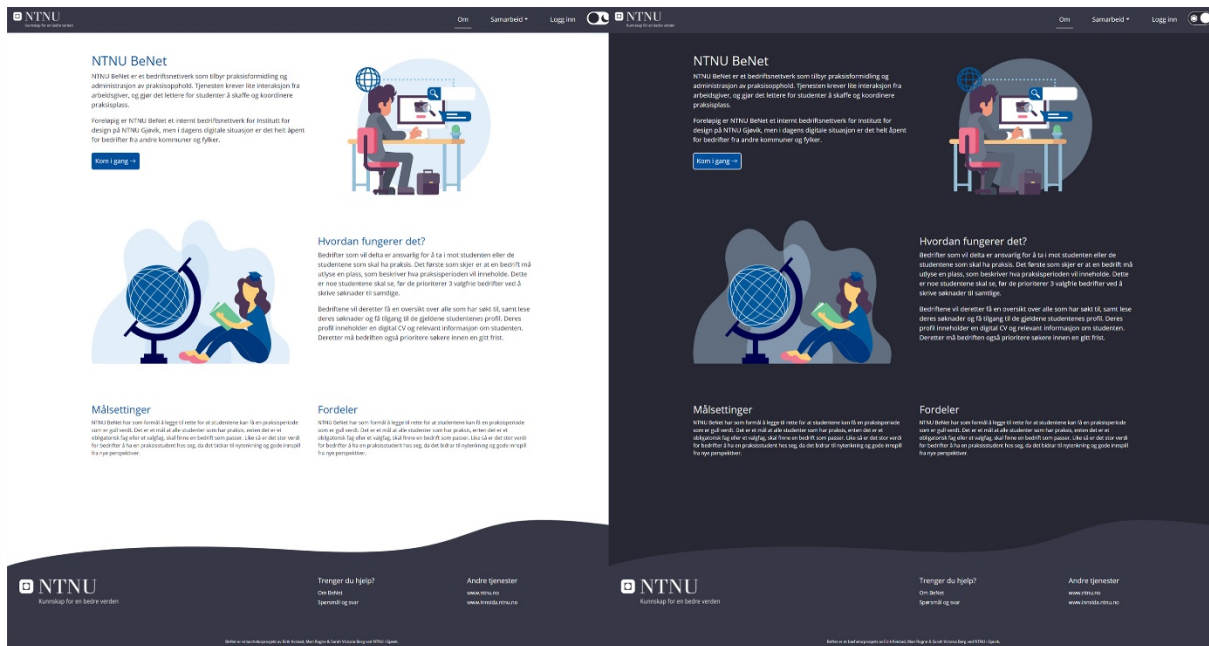
Dagens ordning av praksisemnet er preget av at informasjon *ikke* er lagret på ett samlet sted, og kan kategoriseres i 4 problemområder, som beskrevet i kapittel 1.1.1. For å ta tak i disse problemområdene samler BeNet all informasjon som tidligere har blitt spredt over flere ulike plattformer i en og samme løsning. Dette berører alle de 4 problemområdene på forskjellige måter. Blant annet minsker det kompleksitetene og fjerner store mengder med bytting av kontekst, i tillegg til at det bidrar til større oversikt for alle brukergruppene, samt gjør det lettere å finne data og informasjon. Dessuten vil den nye håndteringen av data tillate høyere grad av autonomitet for studenter og bedrifter, da de enklere kan opprette en ny utlysning, skive søknader og prioritere hverandre. Dette eliminerer emneansvarlig som mellomledd i søknadsprosessen. I tillegg bruker BeNet en algoritme for å automatisk matche bedrifter og studenter basert på prioriteringer, som gjør at emneansvarlig er fritatt fra å utføre denne oppgaven manuelt.

Figur 51 viser hvilke navigasjonsmuligheter alle tre brukergruppene har, hvor de blå elementene er sider alle innloggede brukere har tilgang til. Om man ikke har en bruker vil man ha tilgang til alle de blå elementene foruten om «kalender» og «hjelp». De gule elementene er sider som kun er utarbeidet for administrative brukere, mens de oransje representerer sider som kun studenter og bedrifter har tilgang til.



Figur 51: Sidekart for BeNet.

Avhengig av rolle, det vil si de tre brukergruppene, vil knapper og noe tekst på de forskjellige sidene variere deretter. I tillegg er landingssiden og visning av utlysninger tilgjengelig dersom brukeren ikke er logget inn. Figur 52 viser et skjermbilde av landingssiden («om»-siden) hvor brukeren ikke er logget inn. Denne siden gir overordnet informasjon om hva BeNet er og hvordan løsningen fungerer. Som bedrift er det mulig å registrere seg ved å fylle ut generell informasjon om bedriften, samt informasjon om kontaktperson. Administrator og student er ment å logge inn med Feide. Feide er ikke implementert i løsningen, derav har ikke disse brukergruppene mulighet til registrering per dags dato. Det er derimot opprettet brukere til studenter og administratorer i systemet for å tilrettelegge for testing.



Figur 52: Skjerm bilde av landingssiden («om»-siden) i light mode (venstre) og dark mode (høyre) i BeNet.

Figur 52 viser landingssiden i light mode (venstre) og i dark mode (høyre). Brukere har gjennom hele løsningen et valg om å bruke løsningen i enten light mode eller dark mode, i lys av teorien i kapittel 3.2.2.3, som viser til at både dark mode og light mode har sine fordeler og ulemper.

«Samarbeid»-siden (se Figur 53 nedenfor) er tilgjengelig for alle som besøker BeNet, og kan sammenlignes med konseptet «Marketplace» i prototypen fra tidligere prosjekt (se kapittel 3.1.1.2). På denne måten kan andre bedrifter som ikke er med i løsningen se hvordan BeNet fungerer og da hvilke bedrifter som er med, og forhåpentligvis se gevinstene ved å delta. Hensikten med at samarbeid er offentlig er derfor å inspirere nye bedrifter til å selv delta. Samarbeid inneholder alle tilgjengelige utlysninger for praksisplasser og bachelorprosjekt. Disse er delt opp i forskjellige undersider, og differensieres ved en sidemeny. Selv om BeNet fokuserer på håndtering av praksisemnet, og ikke tilrettelegger for håndtering av bacheloremnet, var det her ønskelig å inkludere bachelorprosjekter for å illustrere hvordan de to emnene kan fremstilles på samme måte. Dersom brukeren er logget inn som student, vil en ekstra knapp være tilgjengelig for hver utlysning som tillater studenten å prioritere praksisplassen. «Velkommen Bruker» oppe i venstre hjørnet er kun for å gjøre det enklere å se hvem man er logget inn som i MVPen.

Praksis

Bachelor

Frist for å søke:  
18. september

## Praksis

Ny utlysning

Finn en praksisplass eller et bachelorprosjekt!

Bruk menyen til venstre for å navigere. Husk at du som student må prioritere max tre bedrifter til praksis om du ikke finner en på egenhånd. Prioriteringene dine finner du i din oversikt.

Søk...



### Praksis i godt miljø

AMBIANCE

Vi tror at et godt tverrfaglig arbeidsmiljø er nøkkelen til å ikke bare ha det bra på jobben, men å lykkes i på markedet! Vi verdsetter våre ansatte, uten dem hadde ingenting blitt gjort. Du vil få muligheten til å være en del av vårt sterke fagmiljø, som støtter hverandre. Temakvelder arrangeres hv...

Prioriter Les mer



### Praksis hos oss!

LOGOKOMPANIET

Du vil gjennom en fantastisk praksisperiode lære drøsevis om logoer. Det er perfekt å føre på CV'en. Hver fredag er det gratis kake i kantinen. Du vil selvfølgelig få en veileder hos oss, som du kan ha kontakt med livet ut om du vil. Håper å få deg med på laget! Vi sees! Du vil lære om følgende: Lo...

Prioriter Les mer



### UX-designere

HEXATECH

Vi ser etter deg! Det er en rivende utvikling innen ytre miljø i vår bransje og hos oss vil du få muligheten til å jobbe med helt konkrete utfordringer og bruke din kompetanse for å utgjøre en forskjell. Du vil være tildelt prosjektet Ranheim-Værnes og være en del av HMS, kvalitet og ytre miljø-tea...

Prioriter Les mer



Nomad DESIGN

### Søker to ivrige studenter

NOMAD DESIGN

Nå ønsker vi å tilby hele TO praksisplasser i en rivende utvikling innen ytre miljø i vår bransje. Dere vil hos oss få muligheten til å jobbe med helt konkrete utfordringer og bruke kompetanse for å utgjøre en forskjell....

Prioriter Les mer

Figur 53: Skjerm bilde av «Samarbeid» i BeNet. Logget inn som student.

Resterende sider krever autentisering for å besøke. Brukergruppene har ulik funksjonalitet, og sidene vil derfor vise ulikt innhold og funksjonalitet etter hvilken rolle brukeren har. Når en bedrift har opprettet en bruker kan de opprette nye utlysninger, det vil si utlyse praksisplasser og/eller forslag til bachelorprosjekter. Dette kan de gjøre flere steder i løsningen. En ny utlysning vil ikke bli tilgjengelig og offentliggjort før emneansvarlig (administrator) har kvalitetssikret utlysningen. Forslag til nye utlysninger vil derfor bli sendt til administrator sin oversikt (se Figur 54), hvor administrator enten kan foreslå en endring, godta eller avslå utlysningen. Om utlysningen blir avslått vil den bli permanent slettet fra BeNet, det vil si at den slettes fra databasen for godt. Dette er på bakgrunn av tiltaket BeNet tar for å bidra til mindre karbonutslipp i forbindelse med internett.

The screenshot shows the BeNet administrator interface. At the top, there is a navigation bar with the NTNU logo and the text 'Velkommen Admin!'. The main content area is titled 'Nye utlysninger' (New job postings). Below the title, there is a sub-header: 'Foreslåtte utlysninger fra bedrifter som krever godkjenning. Du kan redigere utlysningen ved å foreslå endring. Godkjent utlysning vil bli tilgjengelig for studenter, avslåtte utlysninger vil bli permanent slettet.' (Proposed job postings from companies that require approval. You can edit the job posting by proposing a change. Approved job posting will be available for students, rejected job postings will be permanently deleted.)

There are three tabs: 'Alle (2)', 'Praksis (1)', and 'Prosjekter (1)'. The 'Alle (2)' tab is selected, showing two job postings:

- Prakis for deg som interessert i logoer.** (Praxis for you who are interested in logos.) - Description: 'Vi i logokompaniet ser etter DEG som er interessert i LOGOER for å jobbe på vår nye logogenerator. Vi jobber med både UI og UX, og jobber tett med kunder.' (We in the logo company are looking for YOU who are interested in LOGOES for our new logo generator. We work with both UI and UX, and work closely with customers.) - Actions: 'Foreslå endring', 'Godkjenn', 'Avslå'.
- Lag logoer!** (Create logos!) - Description: 'Dette er et kult bachelorprosjekt for deg som liker logoer....' (This is a cool bachelor project for you who like logos....) - Actions: 'Foreslå endring', 'Godkjenn', 'Avslå'.

At the bottom of the page, there is a footer with the NTNU logo and the text 'Kunnskap for en bedre verden'. There are also links for 'Trenger du hjelp?' (Need help?) and 'Andre tjenester' (Other services).

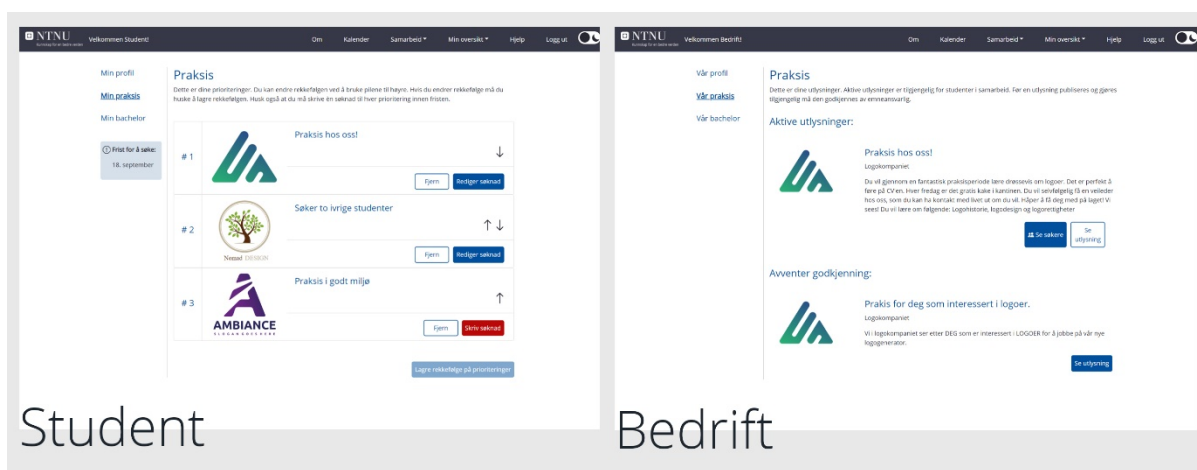
Figur 54: Visning for nye utlysninger for brukergruppen administrator.

«Min oversikt» for studenter og bedrifter viser til hvordan komponenter er med på å bygge opp brukergrensesnittet, og henter ulik data basert på rolle og visning. Figur 55 viser et skjermbilde av profilen til en student. Profilen til studentene fungerer som en CV, noe som betyr at BeNet ikke

tilrettelegger for dokumentopplasting av egenkomponert CV. Dette er fordi det sparer lagringsplass, da man unngår at potensielt alle studentene i praksisemnet laster opp tre kopier hver. For bedrifter vil profilen benytte seg av samme komponenter og dermed være likt utformet, men naturlig nok vise frem ulik informasjon.

Figur 55: Skjerm bilde av profil for student. Logget inn som student.

«Min oversikt» er delt inn i undersider for brukerens profil, praksisplasser og bachelorprosjekter. Disse kan navigeres gjennom sidemenyen. Figur 56 viser «min praksis» for student til venstre og «vår praksis» for bedrift til høyre. Visningen for bachelorprosjekter vil være lik som siden for praksis, men knapper og funksjonalitet er ikke bygd ut da håndtering av bacheloremnet er ansett som «nice-to-have» (se kapittel 5).



Figur 56: Oversikt over praksis for student (venstre) og bedrift (høyre).

«Min praksis» for brukergruppen student tilbyr funksjonalitet for å skrive søknad til de bedriftene studenten har valgt å prioritere. Ved å skrive søknad i systemet vil søknadene kun bli lagret i BeNet. Det vil si at man unngår at en student lagrer tre søknader på sin datamaskin, som blir sendt via e-post til emneansvarlig som deretter lagrer søknadene på sin datamaskin for å sortere og videresende de til bedriftene. Søknadene som studentene skriver, vil bli synlig for de aktuelle bedriftene når det er bedriftenes tur til å prioritere studenter. Bedriften vil da gjennom «vår praksis» ha mulighet til å besøke profilene til studentene som har søkt, lese søknadene, og prioritere tre studenter (se Figur 57).



The screenshot shows the NTNU BeNet interface. At the top, there is a navigation bar with the NTNU logo, the text "Velkommen Bedrift!", and several menu items: "Om", "Kalender", "Samarbeid", "Min oversikt", "Hjelp", and "Logg ut".

The main content area is divided into two sections:

- Søkere til Logokompaniet:** This section is titled "Søkere til Logokompaniet" and includes the subtitle "Legg til søkere med plusstegnet". It contains two entries:
  - Kjersti Styre:** Includes a profile picture, the name "Kjersti Styre", and links for "Profil" and "Søknad". A blue plus sign is to the right.
  - Eirik Kvistad:** Includes a profile picture, the name "Eirik Kvistad", and links for "Profil" and "Søknad". A blue plus sign is to the right.
- Våre prioriteringer:** This section is titled "Våre prioriteringer" and includes the subtitle "Trykk på «fjern» for å angre". It contains one entry:
  - Lise Vik Johansson:** Includes a profile picture, the name "Lise Vik Johansson", and a red "Fjern" button. Below the name are two rows with the numbers "2" and "3" respectively.

At the bottom right of the main content area, there are two buttons: "Lagre" (Save) and "Avbryt" (Cancel).

The footer of the page contains the NTNU logo and tagline "Kunnskap for en bedre verden" on the left. In the center, it says "Trenger du hjelp?" followed by "Om BeNet" and "Spørsmål og svar". On the right, it says "Andre tjenester" followed by "www.ntnu.no" and "www.innsida.ntnu.no". At the very bottom, there is a small line of text: "BeNet er et bachelorprosjekt av Eirik Kvistad, Mari Rognie & Sarah Victoria Berg ved NTNU i Gjøvik."

Figur 57: Visning hvor bedrift ser søkere og prioriter hvilke studenter de ønsker.

Administrator har en helt egen side for administrasjon av praksisemnet, som ingen andre brukergrupper har tilgang til. På denne siden vil administrator blant annet få presentert hvilke bedrifter og studenter som har blitt satt sammen basert på prioriteringer fra studenter og bedrifter (se Figur 58). Dette blir fremstilt i en automatisk sammensetting hvor 1-grads match vil si at student og bedrift begge har prioritert hverandre på førsteplass, og 2-grads match vil forekomme i andre scenarier.

NTNU Kunnskap for en bedre verden Velkommen Admin!

Om Kalender Oversikt Samarbeid Hjelp Logg ut

Nye utlysninger

**Matching**

Brukere

Historikk

Innstillinger

### Matching

Sammenstilling av prioriteringene til bedrifter og prioriteringene til studenter. 1-grads match vil forekomme om både bedrift og student har prioritert hverandre på førsteplass.

#### 1-grad

Bedrift	Student
Logokompaniet	Lise Vik Johansson
Logokompaniet	Eirik Kvistadd

#### 2-grad

Bedrift	Student

Godkjenn

NTNU Kunnskap for en bedre verden

Trenger du hjelp?  
Om BeNet  
Spørsmål og svar

Andre tjenester  
www.ntnu.no  
www.innsida.ntnu.no

BeNet er et bachelorprosjekt av Eirik Kvistadd, Mari Rognie & Sarah Victoria Berg ved NTNU i Gjøvik.

Figur 58: Visning for matching av bedrift og student for administrator.

Som beskrevet i kapittel 5.2.5 har BeNet tre tilstander. Administrator kan bytte mellom disse tilstandene i systemet gjennom «Innstillinger» på «Oversikt», se Figur 59. Dersom administrator slår på at bedrifter kan prioritere, går tilstanden over til tilstand to og låser handling for studenter. Den tredje tilstanden inntreffer automatisk når administrator godkjenner matchingen i «Matching» (se Figur 58 ovenfor). Administrator kan også gå til den tredje tilstanden ved å slå på matching gjennomført i «Innstillinger».

NTNU  
Kunnskap for en bedre verden

Velkommen Admin!

Om Kalender Oversikt Samarbeid Hjelp Logg ut

Nye utlysninger  
Matching  
Brukere  
Historikk  
Innstillinger

## Innstillinger

**Bedrifter kan prioritere**  
Bestemmer hvorvidt bedrifter kan prioritere / studenter kan søke. I fremtiden kommer dette til å bestemmes av en dato/ kalender.

Slå på

**Matching gjennomført**  
Denne blir satt til True når admin har godkjent matching for sesongen. Partene kan dermed ikke søke/prioritere lengre.

Slå på

NTNU  
Kunnskap for en bedre verden

Trenger du hjelp?  
Om BeNet  
Spørsmål og svar

Andre tjenester  
www.ntnu.no  
www.innsida.ntnu.no

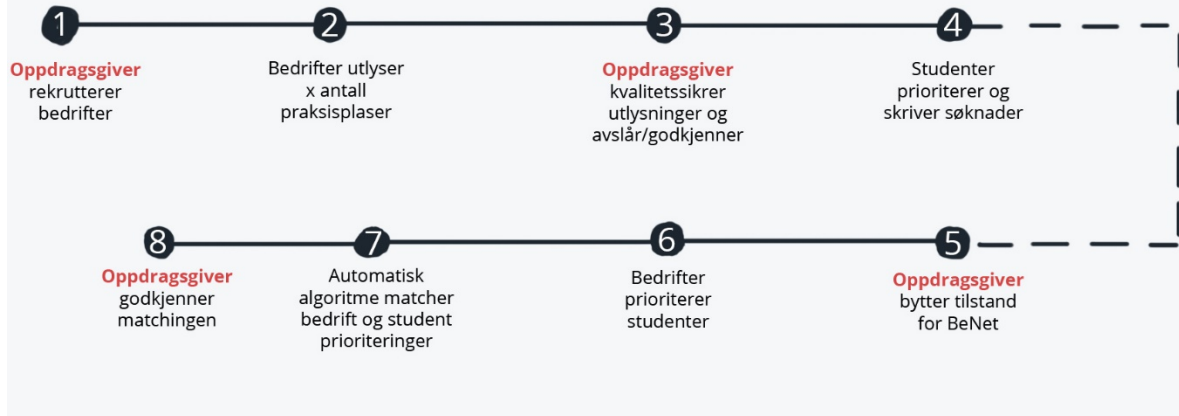
BeNet er et bachelorprosjekt av Eirik Kustad, Mari Rognne & Sarah Victoria Berg ved NTNU i Gjøvik.

Figur 59: Innstillinger for administrator.

## 6.1 BeNets håndtering av praksisemnet

BeNet sin håndtering av praksisemnet inneholder like mange steg som i dagens ordning av praksisemnet, men for oppdragsgiver og andre administratorer i systemet er involveringen i stegene av prosessen nå blitt halvert. Dette ser man ved å sammenligne Figur 1 i kapittel 1.1.1 med Figur 60 nedenfor. Oppdragsgivers involvering kreves nå i kun 4 av stegene, i kontrast til 8 av stegene i dagens ordning.

# BeNet's håndtering av praksisemnet



Figur 60: BeNets håndtering av praksisemnet.

Steg 1 forblir det samme, men har nå en ny betydning da oppdragsgiver kan bruke BeNet til å se hvilke bedrifter som var involvert i fjor, samt finne kontaktinformasjon til disse på samme sted om oppdragsgiver ønsker å invitere bedriften ved nytt semester. Som vist i steg 5 (se Figur 60 ovenfor) må oppdragsgiver selv bytte tilstand på BeNet. Ved eventuell videre utvikling vil dette steget erstattes ved å legge til frister i en kalender, men krever dog fremdeles involvering av oppdragsgiver.

## 6.2 Brukerguide

BeNet har implementert en kompleks prosess som differensierer mellom ulike tilstander over tid. Dette kapitlets hensikt er å demonstrere hvordan brukerflyten er ment å fungere, slik at man kan teste løsningen som alle tre brukergruppene og se hvordan de påvirker hverandre. Løsningen kan besøkes gjennom følgende link: <https://bachelorproject-2f333.web.app/>

For å kunne følge brukerguiden, er innloggingsinformasjon for tre brukere, en fra hver brukergruppe, tilgjengelig i Tabell 26 nedenfor.

ROLLE	EMAIL	PASSORD
ADMINISTRATOR	admin@adminsen.no	passord
BEDRIFT	marisr@stud.ntnu.no	passord
STUDENT	eirik.kvistad@gmail.com	test123

Tabell 26: Innloggingsinformasjon for en bruker for alle brukergruppene.

Brukerguiden er delt inn i flere deler, hvor hver del bytter rolle i systemet. For å illustrere brukerflyten starter brukerguiden som en bedrift som ikke har noen avtale med NTNU fra før, se Tabell 27.

---

<b>DEL 1 – BEDRIFT</b>	
<b>STEG 1</b>	Registrer en ny bedrift i systemet, det er ikke nødvendig å laste opp bilde
<b>STEG 2</b>	Gå til samarbeid, og opprett en ny utlysning med typen praksis
<b>STEG 3</b>	Gå til «vår praksis» i «min oversikt», og se at utlysningen venter på godkjenning
<b>STEG 4</b>	Logg ut.

*Tabell 27: Del 1 – brukerguide*

Bedriften har nå opprettet en utlysning, som krever godkjenning av administrator før den blir publisert og tilgjengeliggjort på «samarbeid». For del 2 er rollen administrator, se tabell Tabell 28.

---

<b>DEL 2 – ADMINISTRATOR</b>	
<b>STEG 5</b>	Logg inn som administrator
<b>STEG 6</b>	Gå til «nye utlysninger» i «oversikt» og godkjenn utlysningen
<b>STEG 7</b>	Logg ut

*Tabell 28: Del 2 – brukerguide*

Utlysningen som bedriften opprettet, er nå synlig i «Samarbeid» ettersom den har blitt godkjent av administrator. Dette betyr at utlysningen er synlig for studenter, som nå kan velge å prioritere denne utlysningen, samt skrive en søknad (Tabell 29).

---

<b>DEL 3 – STUDENT</b>	
<b>STEG 8</b>	Logg inn som student
<b>STEG 9</b>	Finn utlysningen som ble opprettet, på «samarbeid» og legg til utlysningen som prioritering nr 1
<b>STEG 10</b>	Gå til «min praksis» i «min oversikt» og skriv søknad til utlysningen og lagre
<b>STEG 11</b>	Logg ut.

*Tabell 29: Del 3 – brukerguide*

Søknaden til studenten har nå blitt sendt til bedriften som eier utlysningen. Bedriften har derimot ikke muligheten til å lese denne eller prioritere studenten før fristen for å søke har utgått. I BeNet er det administrator som manuelt må bytte tilstand for å muliggjøre dette (Tabell 30).

---

**DEL 4 – ADMINISTRATOR**

<b>STEG 12</b>	Logg inn som administrator
<b>STEG 13</b>	Gå til «oversikt» og videre til «innstillinger»
<b>STEG 14</b>	Slå på at bedrifter kan prioritere
<b>STEG 15</b>	Logg ut

*Tabell 30: Del 4 – brukerguide*

BeNet befinner seg nå i tilstand nummer to, og bedriften kan lese søknader samt prioritere de studentene som har søkt (Tabell 31). Dette betyr at studenter ikke lenger har mulighet til å endre sine prioriteringer ved å legge til nye, fjerne eksisterende eller omprioritere.

---

**DEL 5 – BEDRIFT**

<b>STEG 16</b>	Logg inn som bedrift (med den mailen du registrerte deg med)
<b>STEG 17</b>	Gå til «vår praksis» i «min oversikt» og prioriter studenten som førstevalg
<b>STEG 18</b>	Lagre
<b>STEG 19</b>	Logg ut

*Tabell 31: Del 5 – brukerguide*

Bedriften sine prioriteringer vil bli automatisk sammensatt med student prioriteringer ved hjelp av BeNet sin algoritme. Disse sammenstillingene vil kun være synlig for administrator som videre kan godkjenne disse (Tabell 32).

---

**DEL 6 – ADMINISTRATOR**

<b>STEG 16</b>	Logg inn som administrator
<b>STEG 17</b>	Gå til «oversikt» og videre til «matching»
<b>STEG 18</b>	Klikk på godkjenn
<b>STEG 19</b>	Logg ut

*Tabell 32: Del 6 – brukerguide*

I det administrator godkjenner sammenstillingene, vil BeNet gå over i neste og siste tilstand, hvor studenten kan se tildelt praksisplass og bedriftene kan se tildelt student(er) (Tabell 33).

---

**DEL 7 – STUDENT**

---

<b>STEG 20</b>	Logg inn som student
<b>STEG 21</b>	Gå til «min oversikt» og videre til «min praksis», og se tildelt praksisplass
<b>STEG 22</b>	Logg ut

*Tabell 33: Del 7 – brukerguide*

Ved å ha gjennomført denne brukerguiden er hovedfunksjonalitet testet. Ved ønske om ytterligere testing, som for eksempel å teste scenarier hvor en bedrift har flere studenter å velge mellom, finnes innloggingsinformasjon til flere studenter i «readme»-filen til prosjektet.

## 7 Diskusjon

Det ble i tidligere arbeid avdekket fire problemområder (Berg, Kvistad, Rogne, 2020b), som beskrevet i kapittel 1.1.1. Med dette ble følgende problemstilling formulert:

Hvordan tilrettelegge for en mer oversiktlig håndtering av praksisemnet ved Institutt for Design ved NTNU i Gjøvik gjennom å utvikle en digital webløsning?

For å besvare problemstillingen ble det dannet to tilhørende forskningsspørsmål, i tillegg til at de siste to fasene av double diamond-modellen ble tatt i bruk. Denne modellen inneholdt flere metoder for utforming og utvikling, og som resulterte i webløsningen BeNet. Dette kapittelet vil drøfte svaret på problemstillingen i lys av resultater og teori, samt ved hjelp av forskningsspørsmålene.

Hovedproblemet i dagens ordning av praksisemnet er at informasjon *ikke* er samlet på ett sted, noe som igjen påvirker fire områder, som diskutert i kapittel 1.1.1. BeNet løser noen av problemområdene med tilhørende problemer, se Tabell 34. Løst kan enten være «ja», «nei», eller «delvis», hvor «delvis» betyr at noen eller deler av problemene under problemområdet er løst, men ikke alle.

Problemområde	Problem	Løst
1	Håndtering og deling av data med de ulike aktørene	Ja
2	Kommunikasjon	Delvis
3	Søknad og prioriteringer	Ja
4	Historisk data	Delvis

Tabell 34: Problemområder for tidligere ordning.

Problemområde 1, *håndtering og deling av data med de ulike aktørene*, vurderes til løst da BeNet samler all informasjon på én og samme plattform. Samtidig gjenstår en svak side i forbindelse med at administrator kan foreslå endringer på en ny utlysning. Når administrator foreslår en endring vil endringen bli utført i utlysningen, men disse endringene vil ikke bli markerte for bedriften. Bedriften kan heller ikke utføre endringer selv. Slik BeNet fungerer nå vil samtidig administrator kunne godkjenne en utlysning som har blitt endret på, uten at den aktuelle bedriften godkjenner denne endringen. I tillegg fører mangel på varsling av hendelser til at bedriften ikke får med seg at det har skjedd en endring i utlysningen. Dette er problematisk da det potensielt kan føre til misforståelser og ulike forventinger til utlysningen.



Funksjonalitet for å foreslå endring har derimot blitt inkludert da det er ønskelig at administrator skal kunne forslå endringer for bedriften, da alternativet er at utlysningen blir avslått og slettes permanent fra databasen. I sistnevnte scenario må bedriften skrive hele utlysningen på nytt, bare for å risikere at også den nye utlysningen blir avslått. Så selv om funksjonaliteten for å foreslå endringer har flere svakheter og krever dialog mellom administrator og bedrift utenfor systemet, ble det ansett som en større risiko å *ikke* inkludere knappen, da alternativet trolig vil føre til større frustrasjon for bedriftene som kan føre til at de ikke lenger ønsker å være del av BeNet.

Problemområde 2, *kommunikasjon*, er delvis løst da BeNet samler all nødvendig informasjon på én plattform, som bidrar til å fjerne behovet om å stille spørsmål ved å sende e-poster, samt sørger for at informasjon holder seg konsekvent. Om konseptet bak kalender integreres skal det ikke være nødvendig å måtte henvende seg til emneansvarlig om for eksempel frister. Det er dog umulig å forutse menneskers adferd og derav om personer kommer til å sende e-poster uavhengig av at løsningen tilbyr all informasjonen som trengs for gjennomføring av praksisemnet. Dette er noe som burde testes hvis eller når BeNet lanseres og tas i bruk. BeNet løser problemene hvor e-post har blitt brukt til å kommunisere. Samtidig ser vi et behov for å bruke automatiserte e-poster for å varsle om nye hendelser i systemet, og fordi BeNet ikke innehar denne funksjonaliteten blir det ansett som en svakhet. Uten noen form for påminning om ny hendelse i systemet vil brukerne bli tvunget til å kontinuerlig besøke systemet for å sjekke, eventuelt at brukerne glemmer å besøke BeNet ofte nok til å få med seg endringer i tide.

Det at problemområde 3, *søknad og prioriteringer*, har blitt ansett som løst, da BeNet lar studenter prioritere og sender søknader, bedrifter prioritere, og gjennomfører automatisk matching. Med andre ord er de tidkrevende oppgavene for emneansvarlig (oppdragsgiver) eliminert. Samtidig kan det drøftes om den automatiske matchingen er rettferdig overfor både student og bedrift. Gitt at det er en større total mengde av studenter enn tilgjengelige bedrifter, så vil man alltid ende opp med en «restliste». Det er også mulig at spesielle tilfeller vil oppstå. Dersom for eksempel fem studenter velger de samme tre bedriftene, vil man ende opp med at to av disse studentene ikke får noen av de plassene de har søkt på og dermed blir satt i en restliste. Samtidig ville dette vært uunngåelig også ved en manuell tilnærming med samme scenario. Dette kunne ha blitt løst ved å for eksempel kreve at studenter søker på flere enn bare tre bedrifter. Uavhengig av tilnærming er emneansvarlig nødt til å gripe inn for de som havner i en restliste. Det er også verdt å merke at man ikke kan forutse om eller hvor mange som havner i en restliste, da det er mulig for studenter å finne egne praksisplasser, det vil si søke på andre bedrifter enn de som er tilgjengelige i BeNet. Hva som er den beste løsningen for å håndtere disse spesielle tilfellene må derfor gjennomgå videre undersøkelser og testing.

Det at problemområde 4, *historisk data*, er ansett som delvis løst. En oversikt over historisk data er tilgjengelig for administrator, men den burde utvides til å inneholde flere felter, som «tittel», «studieretning», og «semester», i tabellen for at historikken skal være optimal.

## 7.1 En god brukeropplevelse

En brukeropplevelse handler ifølge Tone Nordbø (2017, s. 17) om hvordan en bruker opplever det å bruke et produkt, samtidig som at brukeropplevelse er individuell og vil variere fra bruker til bruker. For dette prosjektet har teori om «mobil først», WCAG, heuristikker og designprinsipper, og dark mode, samt resultatene fra utviklingsfasen, og dermed webløsningen «BeNet», bidratt til å svare på **forskningsspørsmål 1** som var: *Hvordan kan man utforme og utvikle en god brukeropplevelse gjennom prinsipper om design og universell utforming?* Dette spørsmålet underbygger problemstillingen ved å drøfte hvordan man skal ta i bruk en digital webløsning i sammenheng med praksisemnet. Om man «tvinger» brukergruppene til å ta i bruk en webløsning for å få gjennomført eller håndtert praksisemnet burde løsningen være brukervennlig og ikke minst bidra til en god brukeropplevelse for å hindre unødvendige og mulig nyoppståtte utfordringer knyttet til emnet.

Faste heuristikker som blir fulgt gjennom hele designet i BeNet tilrettelegger for en bedre brukeropplevelse, da det utgjør et mer konsistent design som følger konvensjoner og generelt skaper en bedre brukerflyt. Samtidig utgjør WCAG en stor forskjell for brukeropplevelsen når det kommer til ulike behov brukerne har, noe som dark mode bidrar til. Da bruk av løsningen på mobil ofte kan foregå mens brukeren er «på farten», altså på bussen eller mens de er ute og går, er brukeropplevelsen blant annet avhengig av at navigasjonsbarer er plassert etter hvor den lettest kan nåes av brukeren. Selv om BeNet har svake sider innenfor disse prinsippene om godt design som må forbedres og videre utvikles, har BeNet flere sterke sider som skaper en god brukeropplevelse. Heuristikker og designprinsipper, WCAG, dark mode, og «mobil først», hjelper altså med å utforme og utvikle den gode brukeropplevelsen.

### 7.1.1 Heuristikker og designprinsipper

Ekspertevalueringen, altså en heuristisk evaluering, som ble utført av en ekspert innen interaksjonsdesign, vurderte BeNet etter anerkjente heuristikker (se kapittel 5.3.1). Heuristikker er en del av det som utgjør et brukersentret design (se kapittel 3.2.2.2). I løpet av evalueringen ble det klart at BeNet ikke var i samsvar med de heuristikkene som nettsiden ble evaluert opp mot. De fleste av disse bruddene ble derimot rettet opp.

Heuristikken «sammenheng mellom system og virkelighet» viste at merkelappen for navigasjonselementer «samarbeid» kanskje ikke var det beste valget for innholdet. Eksperten foreslo blant annet å kalle elementet «utlysninger», da det er mer beskrivende for innholdet nå. Dersom innholdet skal utvides til å inneholde andre typer samarbeid derimot, kan det tenkes at «utlysninger» kan bli for snevert, men dette er noe som burde bli revurdert ved en eventuell utvidelse av systemet. Ved gjennomføring av brukertesting av prototypen var heller ikke merkelappen «samarbeid» et problem. Samtidig er det her vanskelig å fastslå om dette ikke var noe problem fordi brukerne forstod at utlysninger befant seg bak denne merkelappen, eller om de ikke hadde problemer fordi de kunne eliminere de andre merkelappene i navigasjonsbaren, og dermed var «samarbeid» det eneste valget. For å fastslå hvilken merkelapp som er best egnet vil det trolig være gunstig å utføre en A/B-test, hvor man tester navigasjonsbaren i isolasjon. Fordi det ikke ligger nok innsikt til grunn, og fordi det ikke er tilstrekkelig med tid til å kunne utføre en god A/B-test, ble det besluttet å la merkelappen forbli «samarbeid».

Den største svakheten som ble oppdaget i ekspertevalueringen var brudd på heuristikken «unngå feil». Dette problemet ble løst ved å implementere en dialogboks som ber om bekreftelse dersom en administrator avslår en forespørsel om en ny utlysning i form av praksisplass eller bachelorprosjekt. Dersom systemet ikke hadde bedt om bekreftelse ved denne handlingen, er det sannsynlig at nye utlysninger hadde blitt avslått uten at det var hensikten. Fordi denne handlingen fører til at en avslått utlysning vil bli permanent slettet fra systemet, var det svært viktig å unngå at brukeren kunne gjøre feil. En slik dialogboks vil også være svært nyttig i andre deler av løsningen, for eksempel dersom en student har begynt på en ny søknad og klikker «avbryt». Da vil ikke søknaden bli lagret og studenten må starte på nytt. Derimot ble avslåing av ny utlysning prioritert, da det vil ha store konsekvenser for flere av brukergruppene. Samtidig blir mangel på dialogboks i forbindelse med skriving av søknader, oppretting av nye utlysninger, og andre handlinger som må lagres, ansett som en stor svakhet som bidrar til å senke nivået på en god brukeropplevelse.

Selv om noen av bruddene fortsatt medfører problematiske aspekter, har løsningen også gode og sterke sider når det kommer til heuristikkene. Eksempelvis følger løsningen heuristikken og designprinsippet om gode tilbakemeldinger på at en handling er utført. For eksempel vil studenter som prioriterer en praksisplass få en bekreftelse på at den er lagt til i listen, eller de vil få en feilmelding som forklarer at studenten allerede har prioritert tre bedrifter. Dette gjør at brukeren vet at handlingen ble gjennomført. På lignende måter viser systemet hva som har blitt gjort eller ikke gjort, og hvorfor, i tillegg til at handlinger som kreves har blitt markert. Eksempelvis vil studenter som ikke har skrevet søknad se en rød knapp med teksten «skriv søknad», mens en vil se en blå knapp med teksten «rediger

søknad» dersom søknaden er skrevet og lagret. Dette er ansett som en stor styrke ved brukeropplevelsen, da de ulike tilbakemeldingene synliggjør prosessen.

Til tross for at brukertestene av prototypen viste at brukerne kunne gjennomføre oppgavene innen en kort tid, avslørte ekspertevalueringen at brukerflyten opplevdes som tung og inneholdt unødvendig mange klikk. Dette viser at en ekspertevaluering hadde vært lønnsomt på et tidspunkt hvor det var bedre tid til å fikse opp de heuristiske bruddene som dukket opp, for eksempel etter første iterasjon av prototypen. Samtidig viser evalueringen at løsningen har både styrker og svakheter, og at løsningen krever ytterligere testing i flere iterasjoner for å virkelig svare til forventningen hva gjelder brukeropplevelse og flyt knyttet til heuristikker.

### 7.1.2 En responsiv løsning

Etter ekspertevalueringen ble det utført en systemtest, som viste at BeNet ikke fungerte optimalt på mobil. Da «mobil først» er viktig for å tilrettelegge både for brukere som ønsker å bruke mobil, men også fordi mobil tilpasning av nettsiden er vesentlig for søkemotoroptimalisering (se kapittel 3.2.1), var det viktig å rette opp i dette. Etter tre tester var alle kriteriene mulig å gjennomføre på mobil (se kapittel 5.3.2).

Hovedmenyen er plassert nederst på skjermen i mobil-visning, noe som er en stor styrke da det gjør løsningen mer brukervennlig ved bruk av én hand eller ved bruk av tommer. Dette er fordi den nederste delen av skjermen er lettest å nå (McGough, 2014). I tillegg er alle elementer som er kodet som knapper store nok til at de kan bli klikket på av en tommel uten problemer. Samtidig er design og utseende ikke optimalisert for denne bruken, da målet var at mobil-visningen skulle være funksjonell, det vil si at brukermål var gjennomførbare. Det foreligger dermed et potensial for forbedringer av det visuelle uttrykket for mobil-visning i videre utvikling for å bedre brukeropplevelsen for brukerne som besøker BeNet gjennom en mobiltelefon.

### 7.1.3 Dark mode

Dark mode er forbundet med flere fordeler, som beskrevet i kapittel 3.2.2.3. Løsningen tilbyr dark mode, altså mørk modus, hovedsakelig for å kunne forbedre synlighet for de brukerne som tjener på det, men også for å legge til rette for de brukerne som ønsker det av andre grunner. Samtidig var det ikke ønskelig at brukere som ikke tjener på det, eller som generelt foretrekker light mode skal bli låst til dark mode. Det er dermed mulig for brukeren å selv velge hvorvidt vedkommende ønsker å bruke dark mode eller ikke, basert på egne preferanser og behov.

Fordi hovedfokuset ved implementering av BeNet var å ferdigstille funksjonalitet, er noen elementer ikke tilstrekkelig tilpasset dark mode, slik som tabeller for administrator og noen av utfyllingsskjemaene. I tillegg er bildet på kalendersiden et statisk bilde som representerer hvordan kalender er ment å se ut, og er derfor ikke kompatibel med dark mode. Samtidig er dette ikke noe som gjør at dark mode ikke kan brukes, og er derfor ikke ansett som et stort problem, annet enn at det er svakhet i designet.

#### 7.1.4 WCAG

WCAG handler om universell utforming av digitale løsninger, deriblant nettsider, og er ikke bare svært viktig for å inkludere alle brukere, men også viktig for å samsvare med det norske lovverket, som forklart i kapittel 3.2.2.1. WCAG er med på å skape en god brukeropplevelse for alle brukere, uavhengig av evne, da det tilrettelegger for et mer brukervennlig system, noe som alle brukere tjener på. Eksempelvis vil WCAG gjøre det lettere å bruke systemet i krevende lysforhold eller i støyfulle miljøer, noe som gagnar alle brukere i dagens samfunn med bærbare PC-er og mobiler.

Et av kravene fra WCAG som forbedrer lesbarhet for svaksynte eller brukere med lesevansker, samt forbedrer lesbarhet i krevende lysforhold, er suksesskriterium 1.4.3 kontrast (Utilsynet, u.å.b). For å sikre at kontrasten er god, brukes mørk skrift på lys bakgrunn i light mode eller lys skrift på mørk bakgrunn i dark mode. All løpende tekst følger nivå AAA på kontrast noe som utgjør svært god lesbarhet, noe som blir komplementert med en skriftstørrelse på 1em, altså 16px, på løpende tekst. Samtidig kan kontrasten mellom skrift og bakgrunn på røde og grønne knapper forbedres, da de kun følger nivå AA på kontrast. Samtidig er minimumskravet nivå AA, noe som betyr at alle elementene er innenfor lovverket.

Alternativ tekst på ikke-tekstlig innhold, som er suksesskriterium 1.1.1 (Utilsynet, u.å.a), som bilder, er noe som har blitt ansett som svært viktig. Bilder som har blitt kodet inn av dekorative årsaker, som bildene på forsiden, har derfor en beskrivende alternativ tekst i form av en alt-tag. Derimot legger løsningen på mange måter opp til at brukere selv legger inn bilder, studenter av seg selv, og bedrifter av sin egen logo. Her var alternativet å be brukeren selv skrive inn en alternativ tekst i det vedkommende laster opp et bilde, men fordi det da er vanskelig å kontrollere hva som blir lagt inn ble dette vurdert som ikke optimalt. Derfor har det blitt kodet inn en generisk alternativ tekst etter hvor bildene blir hentet. Eksempelvis vil alle utlysninger ha et bilde, enten bedriftens opplastede logo, eller en standard logo som blir hentet fra databasen. Alle disse bildene vil ha alt-taggen «Bedrift logo». Selv om dette heller ikke er en optimal løsning, blir det ansett som mer sikkert enn at brukeren fyller inn selv, samt at det unngår at en skjermleser leser opp filnavn på bildet (Utilsynet, u.å.c).

I tillegg er BeNet tilrettelagt for tastaturnavigering. Ved tastaturnavigering er alle klikkbare elementer tilgjengelige, foruten om input-feltet for dark mode. Dark mode er derimot ikke en funksjonalitet som har innvirkning på gjennomføring av oppgaver i BeNet, og har ikke blitt prioritert ved tilrettelegging av tastaturnavigering. Samtidig er dark mode ment for blant annet å forbedre synlighet, i tillegg til at dark mode kan spare skjermtid (se kapittel 3.2.2.3), noe som betyr at tastaturnavigering burde være mulig. En annen svakhet ved tastaturnavigering er at fokuserte elementer ikke er så synlige som de burde være, spesielt på sekundærknapper, da denne knappen er markert med transparent bakgrunn og blå/hvit kant. Med andre ord kan kun endring i fargen på kanten være vanskelig å oppfatte.

## 7.2 En bærekraftig løsning

Internett er en stor bidragsyter til de globale karbonutslippene. Som vist i kapittel 3.2.3 vil mer enn 14% av karbonutslippet innen 2040 være tilknyttet internett og såkalt nettsøppel, altså unødvendig lagret data (Vidal, 2017). Prosjektgruppen er opptatt av å bidra til å skape bærekraftige løsninger, og hadde et ønske om at prosjektet skulle reflektere dette. Gjennom prosjektet har det blitt gjort ulike tiltak for å gjøre BeNet bærekraftig ved å svare på **forskningsspørsmål 2** som var: *hvilke tiltak kan prosjektgruppen gjøre for å unngå overflødig karbonutslipp relatert til internett?* Tiltakene prosjektet har tatt er blant annet å slette unødvendig data samt samle all informasjon til ett sted og implementere et valg om dark mode.

Prosjektgruppen har vurdert hvilke data som med hensikt må forbli lagret med formål om å vise frem historikk, og hvilke data som kan slettes permanent. Det ble besluttet at alle utlysninger som av ulike grunner *ikke* blir godkjent av en administrator skal bli slettet. Dette er utlysninger som ikke har blitt ansett som levedyktige, som ikke har holdt god nok standard, eller ikke er relevante for studieprogrammene. Det kan diskuteres om dette er en ulempe for bedriftene, da en avslått utlysning mulig kunne blitt brukt som utkast til en ny forbedret versjon. Dette er dog noe som anses som lite sannsynlig da administrator har mulighet til å foreslå endringer for at utlysningen skal kunne bli godkjent. De avslåtte utlysningene vil da trolig falle inn under statistikken som tilsier at 90% av informasjon på internettet aldri vil bli sett på igjen. Altså har prosjektgruppen hatt som mål å kun lagre informasjon som er av verdi for brukerne og som er del av de 10% av informasjon som faktisk blir brukt.

BeNet tilrettelegger ikke for dokumentopplasting av egenkomponert CV da all data lagres på døgnaktive serverparker som krever mye energi. Om alle nordmenn sletter så lite som 1-2 eposter hver kan man spare like mye strøm som trengs for å lade en million smarttelefoner (Atea, 2021). Dette vil si at BeNet bidrar til å spare store mengder strøm ved å *ikke* lagre potensielt 3 kopier av hver CV. I tillegg tilbyr BeNet dark mode, som kan bidra til å spare batterilevetid for noen skjermtyper (se

kapittel 3.2.2.3), noe som er med på å minske strømforbruk relatert til ladning. Dette er dog ikke noe som brukerne blir tvunget til å bruke da brukere kan velge modus basert på egne preferanser og behov.

Ved gjennomføring av praksisemnet i BeNet vil all informasjon være samlet på en plattform, noe som gjør at den bare trenger å bli lagret på ett system. Dette vil hjelpe til med å redusere informasjonsmengden på nettet og dermed også spare på kapasitet på serverparker. Samtidig benytter BeNet seg av en ikke-relasjonell database, noe som fører til overflødig data. BeNet fokuserer på å lagre så lite data som mulig i forhold til å være en bærekraftig løsning, noe som en ikke-relasjonell database strider imot. Overflødig data er derimot en konsekvens av at en ikke-relasjonell database blir brukt for å tilrettelegge for senere oppskalering og mer effektive og raskere lesespørringer. Samtidig er det muligens verdt å la brukeren vente litt ekstra fremfor å legge igjen større avtrykk i form av unødvendig data.

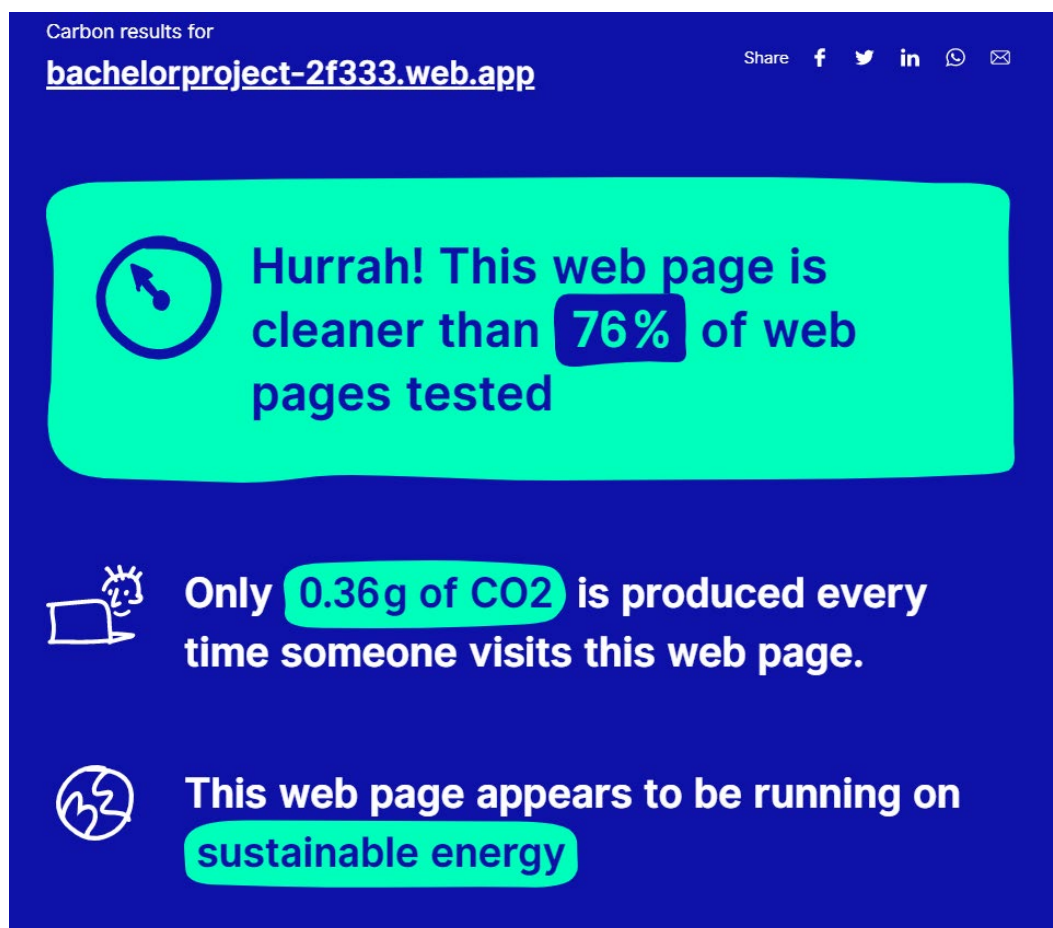
Derimot er den ikke-relasjonelle databasen Firestore som blir benyttet i prosjektet en del av Google Cloud (se kapittel 5.1.3.2 og 5.1.3.3). Ifølge Google Cloud (2021) bruker hver Google Cloud-region strøm fra nettet hvor regionen er lokalisert. Regionen BeNet benytter seg av er europa-west3, Frankfurt, hvor 61% av strømforbruket er karbonfri energi (CFE). CFE% representerer gjennomsnittlig prosentandel av tiden BeNet kjører på karbonfri energi. I retrospekt ser man at det er mulig å velge region før man konfigurerer et prosjekt i Firebase, hvor det da hadde vært mer ønskelig å ta i bruk region europa-north1, Finland, da denne regionen har 77 CFE% og er med det regionen i Europa med høyest CFE% (Google Cloud, 2021).

## 7.2.1 Måling av BeNet sitt digitale avtrykk

Å beregne karbonutslippene til et nettsted kan være en utfordring, men ifølge Website Carbon Calculator, som er et gratis verktøy som gir en enkel og grov ide om nettstedets karbonutslipp, kan man kalkulere et godt estimat ved å bruke fem viktige datagrunnlag: overføring av data, nett-trafikk, energiintensiteten av webdata, kilde for energi som blir brukt av datasenteret, karbonintensitet av elektrisitet (Website Carbon Calculator, u.å.).

Website Carbon Calculator (u.å.) utfører testen ved å måle dataen som blir overført når nettsiden lastes inn, noe som omtrent er proporsjonalt med energien som blir brukt. Dataen som blir målt blir multiplisert med energibruksdataene. Dessuten justeres også resultatet for å ta hensyn til besøkende som har deler av informasjon mellomlagret i nettleseren (Website Carbon Calculator, u.å.).

Målingen<sup>7</sup> gjennomført i prosjektet viser at BeNet er 76% mer bærekraftig enn alle nettsidene som har gjennomgått samme test i Website Carbon Calculator (se Figur 61 nedenfor). Netliffe har en lignende kalkulator som måler på tilnærmet like datagrunnlag som Website Carbon Calculator. I Netliffe sin kalkulator<sup>8</sup> fikk BeNet et resultat på 75%.



Figur 61: Website Carbon Calculator sin måling av BeNet.

Resultatene til BeNet er kun sammenlignet med andre nettsider som har gjennomgått samme testing, og gir derfor ikke en indikator på hvor bærekraftig BeNet er i forhold til nettsider på lands- eller verdensbasis. Det kan også diskuteres hvorvidt disse målingene er pålitelige. Website Carbon Calculator nevner blant annet at de i dagens utgave av kalkulatoren kun tester *hjemmesiden* til et nettsted, noe som betyr at resultatet kunne sett annerledes ut dersom man hadde testen en av sidene med mer innhold og data. Samtidig mener Netliffe (u.å.), som også kun tester hjemmesiden i sin kalkulator, at test av hjemmesiden er en god nok indikator på hvor stor nettsiden er.

<sup>7</sup> [https://www.websitecarbon.com/website/bachelorproject-2f333-web-app/?fbclid=IwAR3vwids0EasoLI-hqN1POj--2TH88\\_EaE3cREWILdZ-F\\_56ef0yWeWGYNs](https://www.websitecarbon.com/website/bachelorproject-2f333-web-app/?fbclid=IwAR3vwids0EasoLI-hqN1POj--2TH88_EaE3cREWILdZ-F_56ef0yWeWGYNs)

<sup>8</sup> <https://netlife-footprint.herokuapp.com/>



BeNet er en MVP som per dags dato inneholder lite data, noe som også var tilfellet ved gjennomføringen av testen. Dette betyr at resultatet kan se helt annerledes ut dersom testen blir gjennomført på et tidspunkt hvor BeNet blir brukt i en større skala hvor mengden data i databasen er betydelig større.

## 7.3 Konklusjon

Praksis er en viktig del av utdanning for mange. Innsatsområde 4, «mer og bedre praksis», fra regjeringens plan som ble lagt frem i stortingsmeldingen i år viser at praksis er på agendaen og stadig må forbedres, som beskrevet i kapittel 1. Dette prosjektet er derfor av høy relevans, da formålet med prosjektet er å forbedre håndteringen av praksisemnet *IMT3541 – Veiledet praksis i medie- og informatikkfag* gjennom å utforme og utvikle en MVP som et forslag til løsning på problemene de forskjellige brukergruppene opplever i dagens ordning av praksisemnet (se kapittel 1.1.1). Prosjektet skulle besvare problemstillingen som stilte spørsmål om *hvordan tilrettelegge for en mer oversiktlig håndtering av praksisemnet ved Institutt for Design ved NTNU i Gjøvik gjennom å utvikle en digital webløsning?*

Diskusjonskapittelet i denne oppgaven belyser hvordan BeNet løser problemområdene knyttet til praksisemnet i dagens ordning, samt hvordan forskningsspørsmålene blir besvart i form av hvilke tiltak BeNet tar for å bidra til en mer bærekraftig løsning, og hva man har forholdt seg til for å tilrettelegge for en god brukeropplevelse. BeNet svarer på problemstillingen ved å løse problemene i dagens ordning og legge til rette for en bedre håndtering av praksisemnet for oppdragsgiver ved å sørge for mer autonomitet i de andre brukergruppene. BeNet skaper derfor et mer systematisk samarbeid mellom Institutt for Design, studenter og næringslivet.

### 7.3.1 Anbefaling til videre arbeid

Ettersom prosjektgruppen mener at BeNet er et godt konsept og et godt forslag som løser flere av problemene til oppdragsgiver i dagens ordning, anbefaler vi at arbeidet og prosjektet tas videre. For å gjøre dette kreves det ytterligere iterasjoner av løsningen, både hva gjelder brukeropplevelse, informasjonsarkitektur, og utvide tilgjengelig funksjonalitet. Alle tre brukergruppene bør inkluderes videre i prosessen slik at de som skal bruke løsningen kan bidra med behov for hva BeNet trenger. I tillegg har prosjektgruppen noen forslag og anbefalinger til hva som bør bli tatt med i neste iterasjon.

Flere iterasjoner av hele double diamond-fasen vil være gunstig for å kartlegge videre behov for praksisemnet. Dessuten bør det gjøres en ny runde med prototyping for å teste ut funksjonalitet som ble vurdert som «nice-to-have» tidligere i prosessen (se kapittel 5). Dette gjelder spesielt

undersøkelser rundt Feide-innlogging for studenter og administratorer, samt utvikling av automatisert e-post i sammenheng med problemområde 2 som diskutert i kapittel 7, da man uten noen form for påminning om ny hendelse vil bli tvunget til å kontinuerlig besøke BeNet på eget initiativ. I tillegg må en måte å håndtere kontrakter bli implementert.

Når det kommer til Feide-innlogging er det flere aspekter som må tas hensyn til. Først og fremst bør ikke systemet være tilgjengelig for alle som har en Feidebruker. BeNet bør kun være tilgjengelig for studenter som er involvert i praksisemnet (eller fremtidige emner/andre typer samarbeid) i studieåret emnet blir utført. I tillegg må systemet ha en måte å skille mellom ansatte som skal ha tilgang som administrator, og ansatte som ikke skal ha det. En mulig løsning er å be administrator laste opp en oversikt over aktuelle studenter for emnet per sesong. Da kan løsningen kryssjekke med andre datakilder for å kun stifte brukere for relevante studenter. Denne problemstillingen krever derimot et større innsiktsarbeid ved videre utvikling av BeNet.

Den automatiske matchingen bør gå gjennom flere iterasjoner for å utvide mulighetene administrator har for å kunne manipulere resultatet. Eksempelvis bør det være mulig å overskride matchingen for å kunne unngå resttilfeller. Dessuten er det helt nødvendig å implementere en oversikt over de studentene som uansett havner i en restliste. Her er emneansvarlig uansett nødt til å gripe inn, men på hvilken måte er verdt å undersøke videre. Et alternativ er å innføre en ny tilstand, hvor søknadsprosessen blir repetert for de som ikke fikk tilbudt praksisplass første gang. Dette krever derimot at prosessen for å dele ut praksisplasser har flere frister. Et annet alternativ er at studentene som gjenstår blir manuelt eller automatisk plassert ut til ledige bedrifter. Her er det flere komplikasjoner. For det første kan det virke urettferdig for studentene som ikke får velge bedrift selv, og for det andre er det et spørsmål om hvordan man skal velge ut hvilke av de gjenstående bedriftene som skal få tildelt en student. Det vil være hjelpsomt å samle innsikt for hvordan dette problemet best bør løses ved videre utvikling av BeNet.

En videreutvikling av funksjonalitet for å foreslå endringer på en ny utlysning kreves. Dette er funksjonalitet som er ytterst viktig å implementere for å la partene samarbeide og komme til enighet på en enkel måte, og for å ikke miste menneskelige aspekter som tillit mellom NTNU og bedrift. Først og fremst bør bedriften få en varsling om nye endringer, hvor endringer tydelig må visualiseres. For det andre bør administrator bli hindret i å godkjenne den aktuelle utlysningen frem til bedriften har godkjent endringene, og for det tredje bør bedriften kunne foreslå endringer tilbake. Det er verdt å nevne at denne dialogen muligens er et steg i prosessen som kan løses på en annen måte, som for eksempel ved å tilrettelegge for integrasjoner med tjenester som tilbyr digitale møter.

Tidligere arbeid har konkludert med at gjennomføringen av bacheloremnene ved Institutt for Design kan gjennomføres på tilnærmet lik måte som praksisemnet, som BeNet til en viss grad illustrerer. Man bør derfor kartlegge videre behov utover praksisemnet, for å tilrettelegge for bachelorprosjekter i større grad, samt muliggjøre tilrettelegging for andre typer samarbeid som workshops og gjesteforelesninger. Avslutningsvis anbefales det også en større og grundigere test som sjekker etter brudd på retningslinjene til WCAG 2.2 før en eventuell lansering, ved å ta i bruk testprosedyrene<sup>9</sup> som foreslått av Uutilsynet, for å måle opp mot hvert suksesskriteria.

---

<sup>9</sup> <https://www.uutilsynet.no/regelverk/testprosedyrar-nettsteder/709>

## 8 Litteraturliste

Altexsoft (2019) *Web Application Architecture: How the Web Works*. Tilgjengelig fra: <https://www.altexsoft.com/blog/engineering/web-application-architecture-how-the-web-works/> (Hentet: 08. mai 2021).

Android Developers (2021) *Dark theme*. Tilgjengelig fra: <https://developer.android.com/guide/topics/ui/look-and-feel/darktheme> (Hentet: 12. mars 2021).

Apple (2020) *How to use Dark Mode on your mac*. Tilgjengelig fra: <https://support.apple.com/en-us/HT208976> (Hentet: 12. mars 2021).

Atea (2021) *Bli med på digital vårrengjøring for miljøet!*. Tilgjengelig fra: <https://www.atea.no/siste-nytt/bli-med-pa-digital-varrengjoring-for-miljoet/> (Hentet: 15. April 2021).

Batschinski, G. (u.å.) *BaaS Providers*. Tilgjengelig fra: <https://blog.back4app.com/baas-providers/> (Hentet: 07. mai 2021).

Berg, S. V., Kvistad, E. og Rogne, M. S. (2020a) *Applikasjon for bedriftsnettverk*. Emnerapport i IMT3007. NTNU.

Berg, S. V., Kvistad, E. og Rogne, M. S. (2020b) *IMT2671 - Web Projects 2 Project report*. Emnerapport i IMT2671. NTNU.

BootstrapVue (u.å.) *BootstrapVue*. Tilgjengelig fra: <https://bootstrap-vue.org/> (Hentet: 05. april 2021).

Borrelli, P. (2011) Angular vs. Vue vs. React: Comparing frameworks by performance, *Blog Rocket*, 11. Januar. Tilgjengelig fra: <https://blog.logrocket.com/angular-vs-react-vs-vue-a-performance-comparison/> (Hentet: 01. mai 2021).

Chen, A. (u.å.) *New data shows losing 80% of mobile users is normal, and why the best apps do better*. Tilgjengelig fra: [https://andrewchen.com/new-data-shows-why-losing-80-of-your-mobile-users-is-normal-and-that-the-best-apps-do-much-better/?utm\\_content=buffer4fa2&utm\\_medium=twitter.com&utm\\_source=social&utm\\_campaign=buffer](https://andrewchen.com/new-data-shows-why-losing-80-of-your-mobile-users-is-normal-and-that-the-best-apps-do-much-better/?utm_content=buffer4fa2&utm_medium=twitter.com&utm_source=social&utm_campaign=buffer) (Hentet: 25. mars 2021).

Churylov, M. (u.å.) *SPA vs MPA: The definitive guide for decision makers*. Tilgjengelig fra: <https://www.mindk.com/blog/single-page-applications-the-definitive-guide/> (Hentet: 07. Mai 2021).

Coleman, T. (2017) *Component-Driven Development*, *Ink by Chromatic*, 10. februar. Tilgjengelig fra: <https://www.chromatic.com/blog/component-driven-development/> (Hentet: 24. mars 2021).

Component Driven (u.å.) *Component Driven User Interfaces*. Tilgjengelig fra: <https://www.componentdriven.org/> (Hentet 10. mars 2021).

Cooper, A., Reimann, R., Cronin, D., Noessel, C., Csizmadi, J., og LeMoine, D. (2014) *About Face: The Essentials of Interaction Design*. 4. utg. Indianapolis: John Wiley & Sons.

Coron, T. (2020) *What is Sass? Your guide to this top CSS preprocessor*. Tilgjengelig fra: <https://www.creativebloq.com/web-design/what-is-sass-111517618> (Hentet: 22. april 2021).

Datareportal (2021) *Digital around the world*. Tilgjengelig fra: <https://datareportal.com/global-digital-overview> (Hentet 06. April 2021).

Design Council (u.å.) *What is the framework for innovation? Design Council's evolved Double Diamond*. Tilgjengelig fra: <https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond> (Hentet: 01. mai 2021).

Distribuert databehandling (2020) *Wikipedia*. Tilgjengelig fra: [https://no.wikipedia.org/wiki/Distribuert\\_databehandling](https://no.wikipedia.org/wiki/Distribuert_databehandling) (Hentet: 09. mai 2021).

Dolcourt, J. (2019) *Dark mode for iPhone arrives Sept. 19: How you'll use Apple's new iOS 13 feature*. Tilgjengelig fra: <https://www.cnet.com/how-to/dark-mode-for-iphone-arrives-sept-19-how-youll-use-apples-new-ios-13-feature/> (Hentet: 12. mars 2021).

Doolan, J. (2016) *Pros and Cons of Rapid Prototyping with HTML5 & Bootstrap*. Tilgjengelig fra: <https://www.mudbath.com.au/insight/pros-and-cons-of-rapid-prototyping-with-html-5-bootstrap/> (Hentet: 27. april 2021).

Downs, J. (2019) Card UI design: fundamentals and examples, *Justinmind*, 18. september. Tilgjengelig fra: <https://www.justinmind.com/blog/cards-ui-design/> (Hentet: 10. mai 2021).

Endsideapplikasjon (2021) *Wikipedia*. Tilgjengelige fra: <https://no.wikipedia.org/wiki/Endsideapplikasjon> (Hentet: 10. Mai 2021).

Entur (u.å) *Brukerhistorier*. Tilgjengelig fra: <https://design.entur.org/kom-i-gang-for-designere/brukerhistorier> (Hentet: 18. April 2021).

Esterkin, J. (2020) *Client-Server Architecture*. Tilgjengelig fra: <https://openclassrooms.com/en/courses/6397806-design-your-software-architecture-using-industry-standard-patterns/6896156-client-server-architecture> (Hentet: 07. Mai 2021).

Firebase (2021) *Choose a data structure*. Tilgjengelig fra: <https://firebase.google.com/docs/firestore/manage-data/structure-data> (Hentet: 08. mai 2021).

Firebase (u.å.a) *Cloud Firestore Data Model*. Tilgjengelig fra: <https://firebase.google.com/docs/firestore/data-model> (hentet: 01.mai 2021).

Firebase (u.å.b) *SDKs and client libraries*. Tilgjengelig fra: <https://firebase.google.com/docs/firestore/client/libraries> (Hentet: 01. Mai 2021).

Firebase (u.å.c) *Use the Cloud Firestore REST API*. Tilgjengelig fra: <https://firebase.google.com/docs/firestore/use-rest-api> (Hentet: 01. Mai 2021).

Forente Nasjoner (2013) *Konvensjon om rettighetene til mennesker med nedsatt funksjonsevne*. Barne-, likestillings- og inkluderingsdepartementet. Tilgjengelig fra: [https://www.regjeringen.no/globalassets/upload/bld/sla/funk/konvensjon\\_web.pdf](https://www.regjeringen.no/globalassets/upload/bld/sla/funk/konvensjon_web.pdf) (Hentet: 12. mars 2021).

Forskrift om universell utforming av IKT-løsninger (2013) *Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske løsninger*. Tilgjengelig fra: <https://lovdata.no/dokument/SF/forskrift/2013-06-21-732?q=universell%20utforming> (Hentet: 12. mars 2021).

Frost, B. (2013) Atomic design, *bradfrost*, 10. juni. Tilgjengelig fra: <https://bradfrost.com/blog/post/atomic-web-design/> (Hentet: 01. Mai 2021).

Gale–Shapley algorithm (2021) *Wikipedia*. Tilgjengelig fra: [https://en.wikipedia.org/wiki/Gale%E2%80%93Shapley\\_algorithm](https://en.wikipedia.org/wiki/Gale%E2%80%93Shapley_algorithm) (Hentet: 07. mai 2021).

Garrett, J. J. (2011) *The Elements of User Experience*. Berkeley: New Riders.

GeeksForGeeks (2020) *SQL vs NoSQL: Which one is better to use?*. Tilgjengelig fra: <https://www.geeksforgeeks.org/sql-vs-nosql-which-one-is-better-to-use/#:~:text=Is%20NoSQL%20faster%20than%20SQL,databases%20depends%20on%20the%20cont ext> (Hentet: 01. mai 2021).

Google Cloud (2021) *Carbon free energy for Google Cloud regions*. Tilgjengelig fra: <https://cloud.google.com/sustainability/region-carbon?fbclid=IwAR1PoLlq37m82TbAjBoOZxwC3ZoWvO9ArBi-x5GUtjSbjNbeMtsOWQSSiv4> (Hentet: 07. mai 2021).

Google Search Central (u.å.) *Mobile-first indexing best practices*. Tilgjengelig fra: <https://developers.google.com/search/mobile-sites/mobile-first-indexing> (Hentet: 14. april 2021).

Granevang, M. (2020a) backend, *Store Norske Leksikon*. Tilgjengelig fra: <https://snl.no/backend> (Hentet: 22. april 2021).

Granevang, M. (2020b) frontend, *Store Norske Leksikon*. Tilgjengelig fra: <https://snl.no/frontend> (Hentet: 22. april 2021).

Greenberg, S., Carpendale, S., Marquardt, N., og Buxton, B. (2012) *Sketching User Experiences: The Workbook*. Waltham: Elsevier.

Kennedy, E. D. (2020) The Responsive Website Font Size Guidelines, *Learn UI Design Blog*, 01. mai. Tilgjengelig fra: <https://learnui.design/blog/mobile-desktop-website-font-size-guidelines.html> (Hentet: 06. april 2021).

Kostadinov, P. (2019) *The pros and cons of Dark Mode: Here's when to use it and why*. Tilgjengelig fra: [https://www.phonearena.com/news/Dark-Mode-iPhone-Android-interface-feature-pros-cons-versus-light-mode\\_id116978](https://www.phonearena.com/news/Dark-Mode-iPhone-Android-interface-feature-pros-cons-versus-light-mode_id116978) (Hentet: 27. mars 2021).

Leung, F. og Zhou, B. (2016) Performance evaluation of Twitter datasets on SQL and NoSQL DBMS, *Web Intelligence*, 14(4), s. 275–286. Tilgjengelig fra: <http://web.a.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=1&sid=68c3b08d-369d-4545-a032-8dbde48c9090%40sessionmgr4006> (Hentet: 10. mai 2021).

Likestillings- og diskrimineringsloven (2018) *Lov om likestilling og forbud mot diskriminering*. Tilgjengelig fra: <https://lovdata.no/dokument/NL/lov/2017-06-16-51?q=forbud%20mot%20diskriminering> (Hentet: 12. mars 2021).

McGough, O. (2014) *Designing for thumbs – The Thumb Zone*. Tilgjengelig fra: <https://usabilla.com/blog/designing-thumbs-thumb-zone/> (Hentet: 04. mai 2021).

McGovern, G. (2019) *Digital is garbage*. Tilgjengelig fra: <https://gerrymcgovern.com/digital-is-garbage/#comments> (Hentet: 25. mars 2021).

Meld. St. (2020-2021)(2021) Utdanning for omstilling — Økt arbeidslivsrelevans i høyere utdanning. Ukjent sted: Kunnskapsdepartementet. Tilgjengelig fra: <https://www.regjeringen.no/no/dokumenter/meld.-st.-16-20202021/id2838171/> (Hentet: 15. april 2021).

Mifsud, J. (u.å.) *Usability Metrics – A Guide To Quantify The Usability Of Any System*. Tilgjengelig fra: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/> (Hentet: 28. april 2021).

Mozilla (u.å) *How the Web works*. Tilgjengelig fra: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/How\\_the\\_Web\\_works](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works) (Hentet: 07. Mai 2021).

Nice, B. (2019) *The difference between sketch, prototype, mockup and wireframe: what's what?*. Tilgjengelig fra: <https://medium.com/level-up-web/the-difference-between-sketch-prototype-mockup-and-wireframe-whats-what-44d2b24ca0f0> (Hentet: 27. april 2021).

Nielsen, J. (2000) *Why you only need to test with 5 users*. Tilgjengelig fra: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (Hentet: 29. februar 2021).

Nielsen, J. (2020) *10 Usability Heuristics for User Interface Design*. Tilgjengelig fra: <https://www.nngroup.com/articles/ten-usability-heuristics/> (Hentet: 06. april 2020).

Nilstun, C. (2020) *implementere*, *Store Norske Leksikon*. Tilgjengelig fra: <https://snl.no/implementere> (Hentet: 02. mai 2021).

Nordbø, T. (2017) *Introduksjon til interaksjonsdesign*. Oslo: Universitetsforlaget.

Norges Teknisk-Naturvitenskapelige Universitet [NTNU] (u.å.a) *Farger i grafisk profil*. Tilgjengelig fra: <https://innsida.ntnu.no/wiki/-/wiki/Norsk/Farger+i+grafisk+profil> (Hentet: 06. april 2021).

Norges Teknisk-Naturvitenskapelige Universitet [NTNU] (u.å.b) *Institutt for design*. Tilgjengelig fra: <https://www.ntnu.no/design> (Hentet: 18. februar 2021).

Norges Teknisk-Naturvitenskapelige Universitet [NTNU] (u.å.c) *NTNUs skrifttype*. Tilgjengelig fra: <https://innsida.ntnu.no/wiki/-/wiki/Norsk/NTNUs+skrifttype> (Hentet: 06. april 2021).

Norman, D. (2013) *The Design of Everyday Things*. New York: Basic Books.

Norsk Senter for Forskningsdata (u.å.) *Hvordan gjennomføre et prosjekt uten å behandle personopplysninger?*. Tilgjengelig fra: <https://www.nsd.no/personverntjenester/oppslagsverk-for-personvern-i-forskning/hvordan-gjennomfore-et-prosjekt-uten-a-behandle-personopplysninger/> (Hentet: 02. mai 2021).

Özlü, A. (2018) *Mastering REST Architecture — REST Architecture Details*, *Ahmet ÖZLÜ*, 29. juli. Tilgjengelig fra: <https://ahmetozlu93.medium.com/mastering-rest-architecture-rest-architecture-details-e47ec659f6bc> (Hentet: 07. Mai 2021).

Paganini, C. (2019) *Primer: Understanding Software and System Architecture*, *The New Stack*. Tilgjengelig fra: <https://thenewstack.io/primer-understanding-software-and-system-architecture> (Hentet: 07. Mai 2021).

- Parisavtalen (2016) *Parisavtalen*. Tilgjengelig fra: <https://lovdata.no/dokument/TRAKTAT/traktat/2015-12-12-32?q=parisavtalen> (Hentet: 25. mars 2021).
- Perez, S. (2016) *Nearly 1 in 4 people abandon mobile apps after only one use*. Tilgjengelig fra: <https://techcrunch.com/2016/05/31/nearly-1-in-4-people-abandon-mobile-apps-after-only-one-use/> (Hentet: 25. mars 2021).
- Perez, S. (2017) *YouTube revamps its desktop site with an updated design, optional dark theme, and faster framework*. Tilgjengelig fra: <https://techcrunch.com/2017/05/02/youtube-revamps-its-desktop-site-with-an-updated-design-optional-dark-theme-and-faster-framework/> (Hentet: 12. mars 2021).
- Persistent storage (2020) *Computer Science Wiki*. Tilgjengelig fra: [https://computersciencewiki.org/index.php/Persistent\\_storage](https://computersciencewiki.org/index.php/Persistent_storage) (Hentet: 08. mai 2021).
- Phan, D. (2016) Mobile-first indexing, *Google Search Central*, 04. november. Tilgjengelig fra: <https://developers.google.com/search/blog/2016/11/mobile-first-indexing> (Hentet: 13. mars 2021).
- Pickering, H. (2017) *A Theme Switcher*. Tilgjengelig fra: <https://inclusive-components.design/a-theme-switcher/> (Hentet: 15. april 2021).
- Raj, S. (2020) SQL vs NoSQL, *Medium*, 21. februar. Tilgjengelig fra: <https://medium.com/nerd-for-tech/sql-vs-nosql-faef10e3852d> (Hentet: 08. mai 2021).
- Regjeringen (2021) *Slik vil regjeringen sikre bedre kontakt mellom studenter og arbeidsliv*. Tilgjengelig fra: <https://www.regjeringen.no/no/aktuelt/slik-vil-regjeringen-sikre-bedre-kontakt-mellom-studenter-og-arbeidsliv/id2838493/> (Hentet: 15. april 2021).
- Rossen, E. (2020) Brukergrensesnitt, *Store Norske Leksikon*. Tilgjengelig fra: <https://snl.no/brukergrensesnitt> (Hentet: 10. mai 2021).
- Samuel, O. (2020) How To Avoid Prop Drilling in React Using Component Composition, *Javascript in plain english*, 29. Juli. Tilgjengelig fra: <https://javascript.plainenglish.io/how-to-avoid-prop-drilling-in-react-using-component-composition-c42adfcde1b> (Hentet: 01. mai 2021).
- Sass (u.å.) *Sass Basics*. Tilgjengelig fra: <https://sass-lang.com/guide> (Hentet: 22. april 2021).
- Sauro, J (2011) *What is a good task-completion rate?* Tilgjengelig fra: <https://measuringu.com/task-completion/> (Hentet: 28. april 2021).
- Shukairy, A. (2014) Usability Metrics: A Guide To Measuring The User Experience, *invesp*, 14. mai. Tilgjengelig fra: <https://www.invespro.com/blog/usability-metrics/> (Hentet: 10. mai 2021).
- Spoiala, C. (2015) *Cloud offering: Comparison between IaaS, PaaS, SaaS, BaaS*. Tilgjengelig fra: <https://assist-software.net/blog/cloud-offering-comparison-between-iaas-paas-saas-baas> (Hentet: 05. mai 2021).
- Stafseng, T., og Barland, J. (2021) Bransjerelaterte læringsaktiviteter i medie- og designfag, *Norsk medietidsskrift*, 28(1), s. 1–19. doi: <https://doi.org/10.18261/ISSN.0805-9535-2021-01-04>
- Statcounter (u.å.) *Desktop vs Mobile vs Tablet Market Share Worldwide*. Tilgjengelig fra: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/?fbclid=IwAR3TiLFQT4676QxVpRGo8dvqTWOOTLPaZyV1fn2sePLsqVG7OsE>



[u2SKe3A#monthly-200901-201612](#) (Hentet: 14. april 2021).

State (computer science) (2020) *Wikipedia*. Tilgjengelig fra: [https://en.wikipedia.org/wiki/State\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/State_(computer_science)) (Hentet: 07. mai 2021).

Thapliyal, A. (2018) *Why Bootstrap Is an Ideal Desing Framework*. Tilgjengelig fra: <https://clutch.co/web-designers/resources/why-bootstrap-is-ideal-design-framework> (Hentet: 05. april 2021).

Toftøy-Andersen, E. og Wold, J. G. (2011) *Praktisk brukertesting*. 1.utg. Latvia: Cappelen Damm.

Uutilsynet (u.å.a) *1.1.1 Ikke-tekstlig innhold (Nivå A)*. Tilgjengelig fra: <https://www.uutilsynet.no/wcag-standarden/111-ikke-tekstlig-innhold-niva/87> (Hentet: 08. mai 2021).

Uutilsynet (u.å.b) *1.4.3 Kontrast (minimum, Nivå AA)*. Tilgjengelig fra: <https://www.uutilsynet.no/wcag-standarden/143-kontrast-minimum-niva-aa/95> (Hentet: 08. mai 2021).

Uutilsynet (u.å.c) *Bilder og grafikk*. Tilgjengelig fra: <https://www.uutilsynet.no/regelverk/bilder-og-grafikk/205> (Hentet: 08. mai 2021).

Vidal, J. (2017) *'Tsunami of data' could consume one fifth of global electricity by 2025*. Tilgjengelig fra: <https://www.climatechangenews.com/2017/12/11/tsunami-data-consume-one-fifth-global-electricity-2025/> (Hentet: 25. mars 2021).

Vollan, E. G. (2019) Slett nettsøppel og redd kloden!, *Netflife*, 7. november. Tilgjengelig fra: <https://www.netlife.com/blogg/slett-nettsoppel-og-redd-kloden> (Hentet: 11. mars 2021).

Vue (u.å.a) *API*. Tilgjengelig fra: <https://vuejs.org/v2/api/> (Hentet: 08. mai 2021).

Vue (u.å.b) *Vue.js Server-Side Rendering Guide*. Tilgjengelig fra: <https://ssr.vuejs.org/#why-ssr> (Hentet: 08. mai 2021).

Website Carbon Caluclator (u.å.) *How does it work?*. Tilgjengelig fra: <https://www.websitecarbon.com/how-does-it-work/> (Hentet: 12. mai 2021).

World Wide Web Consortium [W3C] (2020) *Web Content Accessibility Guidelines (WCAG) Overview*. Tilgjengelig fra: <https://www.w3.org/WAI/standards-guidelines/wcag/> (Hentet: 16. mars 2021).

World Wide Web Consortium [W3C] (u.å.) *Web Content Accessibility Guidelines (WCAG) 2.2*. Tilgjengelig fra: <https://www.w3.org/TR/WCAG22/> (Hentet: 16. mars 2021).

## 9 Figurliste

Figur 1: Dagens ordning i praksisemnet (basert på Berg, Kvistad, og Rogne (2020b)).	3
Figur 2: Skjerm bilde av et utdrag kanban-boards i GitKraken.	8
Figur 3: Skjerm bilde av GANTT-skjema for prosjektet.	9
Figur 4: Sammenhengen mellom prosjektene. Illustrert av Sarah Berg.	10
Figur 5: Brukergruppene av løsningen. Tegnet av Sarah Berg av ikoner fra flaticon.com.	12
Figur 6: Fra venstre: Marketplace, dashboard, ny utlysning og aktiv utlysning for studenter fra hi-fi prototype fra Webprosjekt II.	13
Figur 7: Sammenstilling av personasene fra IMT3007 – Prosjektutvikling i kreativ virksomhet.	15
Figur 8: Responsivt design fra desktop til mobil. Illustrert av Mari Rogne.	19
Figur 9: Ikke-forklarende tilbakemelding vs. forklarende tilbakemelding.	21
Figur 10: Eksempel på lys og mørk modus. Illustrert av Sarah Berg.	22
Figur 11: Double diamond prosess i prosjektet. Illustrert av Sarah Berg.	24
Figur 12: Utforme - første fase av double diamond som er fokus for kapittel 4.	25
Figur 13: Antall nødvendige brukertester (Nielsen, 2000). Illustrert av Sarah Berg.	29
Figur 14: Formel for grad av gjennomføring (Mifsud, u.å.).	32
Figur 15: Formel for tidsbasert effektivitet (Mifsud, u.å.).	33
Figur 16: Brukerhistorie for brukergruppen student.	34
Figur 17: Brukerhistorie for brukergruppen bedrift.	35
Figur 18: Brukerhistorie for brukergruppen administrator.	35
Figur 19: Samling av skisser på papir.	36
Figur 20: Utvalg av første iterasjon av prototypen.	38
Figur 21: Grad av gjennomføring for brukergruppen student ved første iterasjon.	41
Figur 22: Tidsbasert effektivitet for brukergruppen student ved første iterasjon.	41
Figur 23: Grad av gjennomføring for brukergruppen bedrift ved første iterasjon.	42
Figur 24: Tidsbasert effektivitet for brukergruppen bedrift ved første iterasjon.	43
Figur 25: Grad av gjennomføring for brukergruppen administrator ved første iterasjon.	44
Figur 26: Tidsbasert effektivitet for brukergruppen administrator ved første iterasjon.	44
Figur 27: Risikovurdering av funn i første iterasjon av prototypen.	45
Figur 28: Utvalg av andre iterasjon av prototypen.	47
Figur 29: Grad av gjennomføring for brukergruppen student i andre iterasjon.	49
Figur 30: Tidsbasert effektivitet for brukergruppen student i andre iterasjon.	50
Figur 31: Grad av gjennomføring for brukergruppen bedrift i andre iterasjon.	51
Figur 32: Tidsbasert effektivitet for brukergruppen bedrift i andre iterasjon.	51
Figur 33: Grad av gjennomføring for brukergruppe administrator i andre iterasjon.	52
Figur 34: Tidsbasert effektivitet for brukergruppe administrator i andre iterasjon.	53

Figur 35: Risikovurdering av funn i andre iterasjon av prototypen. ....	53
Figur 36: Utvikle – andre fase av double diamond som er fokus for kapittel 5. ....	55
Figur 37: Funksjonalitet kategorisert i MVP og «nice-to-have» .....	56
Figur 38: Klient-tjener-modellen (Mozilla, u.å). ....	58
Figur 39: Client-Server Architecture (Esterkin, 2020). ....	59
Figur 40: Illustrasjon av komponent-drevet oppbygning. Illustrert av Sarah Berg. ....	60
Figur 41: Ulike deler av applikasjonsarkitektur (Altexsoft, 2019). ....	62
Figur 42: Infrastruktur for løsningen .....	66
Figur 43: Skjerm bilde fra prototypen med eksempel på identifisering av komponent «Card». ....	69
Figur 44: Mappestruktur. ....	71
Figur 45: Eksempel på hvordan ulike komponenter benytter samme data på fra ulike sider .....	72
Figur 46: Skjerm bilde av databasen. ....	73
Figur 47: Autentiseringsflyt. ....	76
Figur 48: Sjekk av innloggingsstatus .....	77
Figur 49: Illustrasjon over hvordan sammensetning av prioriteringer i BeNet fungerer. ....	79
Figur 50: Mobilversjon før (til venstre) og etter (til høyre). ....	87
Figur 51: Sidekart for BeNet. ....	89
Figur 52: Skjerm bilde av landingssiden («om»-siden) i light mode (venstre) og dark mode (høyre) i BeNet. ....	90
Figur 53: Skjerm bilde av «Samarbeid» i BeNet. Logget inn som student. ....	91
Figur 54: Visning for nye utlysninger for brukergruppen administrator. ....	92
Figur 55: Skjerm bilde av profil for student. Logget inn som student. ....	93
Figur 56: Oversikt over praksis for student (venstre) og bedrift (høyre). ....	94
Figur 57: Visning hvor bedrift ser søkere og prioriter hvilke studenter de ønsker. ....	95
Figur 58: Visning for matching av bedrift og student for administrator. ....	96
Figur 59: Innstillinger for administrator. ....	97
Figur 60: BeNets håndtering av praksisemnet. ....	98
Figur 61: Website Carbon Calculator sin måling av BeNet. ....	110

## 10 Tabelliste

Tabell 1: Problemområder for dagens ordning basert på funn fra Berg, Kvistad, og Rogne (2020b). ...	3
Tabell 2: Fordeling av hovedansvar for medlemmene av prosjektgruppen. ....	8
Tabell 3: Scenarier for de tre brukergruppene (Berg, Kvistad, Rogne, 2020a). ....	16
Tabell 4: Fordeling av kjønn per runde brukertesting.....	29
Tabell 5: Oppgaver for brukergruppen student ved brukertesting.....	30
Tabell 6: Oppgaver for brukergruppen bedrift ved brukertesting.....	30
Tabell 7: Oppgaver for brukergruppen administrator ved brukertesting.....	31
Tabell 8: Prosess for navngivning av løsningen. ....	37
Tabell 9: Rådata fra brukertesting for brukergruppen student. ....	40
Tabell 10: Rådata fra brukertesting for brukergruppen bedrift. ....	42
Tabell 11: Rådata fra brukertesting for brukergruppen administrator. ....	43
Tabell 12: Rådata fra brukertesting - brukergruppe student, andre iterasjon.....	49
Tabell 13: Rådata fra brukertesting - brukergruppe bedrift, andre iterasjon.....	50
Tabell 14: Rådata fra brukertesting - brukergruppe administrator, andre iterasjon. ....	52
Tabell 15: Sammenligning av enside- og flersideapplikasjon (Churylov, u.å).....	63
Tabell 16: Prosjektets teknologistabel .....	65
Tabell 17: Oversikt over komponenter og deres funksjoner.....	70
Tabell 18: Tabeller i databasen. ....	74
Tabell 19: Tiltak for optimalisering av datalagring .....	75
Tabell 20: Oversikt over funksjonalitet i de tre tilstandene. ....	78
Tabell 21: Steg i sammensetning av prioriteringer. ....	80
Tabell 22: Oversikt over brudd på heuristikkene av Nielsen (2020) i webløsningen BeNet.....	82
Tabell 23: Resultat av systemtest for brukergruppe student. ....	85
Tabell 24: Resultat av systemtest for brukergruppe bedrift.....	85
Tabell 25: Resultat av systemtest for brukergruppe administrator. ....	85
Tabell 26: Innloggingsinformasjon for en bruker for alle brukergruppene. ....	98
Tabell 27: Del 1 – brukerguide .....	99
Tabell 28: Del 2 – brukerguide .....	99
Tabell 29: Del 3 – brukerguide .....	99
Tabell 30: Del 4 – brukerguide .....	100
Tabell 31: Del 5 – brukerguide .....	100
Tabell 32: Del 6 – brukerguide .....	100
Tabell 33: Del 7 – brukerguide .....	101
Tabell 34: Problemområder for tidligere ordning. ....	102

# 11 Vedlegg

Vedlegg 1: Webprosjekt II

Vedlegg 2: Prosjektutvikling i kreativ virksomhet

Vedlegg 3: Samtykkeerklæringer

Vedlegg 4: Brukertester

Vedlegg 5: Illustrasjoner fra Vecteezy.com

Vedlegg 6: Systemtest

Vedlegg 7: Prosjektavtale

Vedlegg 8: Kildekode til BeNet

Vedlegg 9: GANTT-skjema

