Webløsning for Tempero Energitjenester AS

Lena Amdal-Larsen, Vilde G. Bertelsen, Ida M. R. Gjeitsund
IDG3101 - Fordypningsprosjekt

Innhold

Innledning ... 3
Forskningsspørsmål og avgrensninger ... 4
Målene våre for prosjektet ... 4
Sammendrag ... 5
Gjennomføring av prosjektet ... 6
Organisering ... 6
Intervju med oppdragsgiver ... 7
Norske lover og regler ... 8
Universell utforming ... 8
Personvern ... 8
Design av brukergrensesnitt ... 9
Teknologiske løsninger ... 10
Betalingsløsninger ... 10
Rammeverk ... 12
Beregningsmodeller ... 13
Informasjon ... 14
Begrensning av feil ... 16
Matematiske utregninger ... 16
Spørreundersøkelse ... 17
Analyse ... 17
Gjennomføring av spørreundersøkelsen ... 18
Side- og funksjonalitetskart ... 19
Prototyping og brukertesting ... 21
Gjennomføringen av brukertester ... 22
Resultater ... 24
Data fra spørreundersøkelsen ... 24

Konsept 29
Registrering av kunder 29
Beregningsmodeller 31
Veiledning 34
Andre funksjonaliteter 34
Data fra brukertestingen 35
Koding av beregningsmodell 37
Upresise utregninger 39
Drøfting 39
Sikkerhetsfølelse ved betaling 39
Bruk av beregningsmodeller 40
Gjennomføring av veiledning 42
Lo-fi prototypen 43
Måloppnåelse 44
Konklusjon 45
Kilder 47
Vedlegg 50

Innledning

Dette emnet gikk ut på å fordype seg i et selvvalgt prosjekt. Etter anbefaling fra veileder bestemte vi oss for å bruke dette emnet som et «forprosjekt» til bacheloroppgaven som kommer i følgende semester.

Vi måtte selv finne en oppdragsgiver som hadde et prosjekt vi kunne fordype oss i. Vår oppdragsgiver ble Tempero Energitjenester AS.

Tempero er et lite rådgivningsfirma som ble stiftet i 2012, lokalisert i Trondheim. De har kunder i Norge, Sverige og Danmark. Tempero sitt største forretningsområde er løsninger for energiforsyninger, effektiv energi og effektbruk i bygg, samt inneklima. De er altså involvert i alle tekniske løsninger i et bygg, med unntak av selve bygningskroppen.

Tempero sine kunder er hovedsakelig større byggeiere og industribedrifter. Noen eksempler på kunder er Studentsamskipnaden i Trondheim, Gjøvik og Ålesund, HNT sykehus, Sintef Eiendom og Indre Fosen kommune.

Tempero jobber både innenfor nybygg og rehabilitering av eksisterende bygg. Deres rolle er å være byggherrens «kompetanse» i møte med entreprenører, rådgivende ingeniørfirma og automatikkleverandører, i små og store byggeprosjekter.

I forbindelse med Tempero sine arbeidsoppgaver har de utviklet et antall beregningsmodeller i Excel for utregning av blant annet energiforsyning, varme, ventilasjon og energiforbruk.

Tempero mener at med å skrive inn riktig data så kan disse beregningsmodellene benyttes direkte av kunder, uten å bruke eksterne rådgivningsselskaper for å få svar på hvilke krav en skal stille til tekniske anlegg, med utgangspunkt i de målsetningene de har. Dette gjelder spesielt ved mindre rehabiliteringer. På denne måten kan slike prosjekter gjennomføres mye raskere og billigere da byggherrer får de grunnleggende data som trengs for å direkte ta kontakt med leverandører, elektrikere, rørleggere, osv.

Tempero ser for seg at disse beregningsmodellene, som de nå har i Excel, kan gjøres tilgjengelig via en nettside. Det er dette prosjektet vårt omhandler.

Tempero sine krav til produktet er at kunden skal ha mulighet til å leie beregningsmodeller, fylle inn relevant data, og få tilbake riktig svar. Ved siden av hver beregningsmodell ønsker de informasjon som forklarer hva modellen beregner. De ønsker også en måte kunden kan kontakte Tempero for veiledning, dersom det trengs.

Forskningsspørsmål og avgrensninger

Hovedproblemstillingen vår i dette prosjektet baserte seg på oppgaven vi ble gitt av Tempero og hva de ønsket utført. Vi stilte derfor spørsmålet:

> «Hvordan kan vi designe en sikker og brukervennlig webløsning som tilgjengeliggjør utvalgte beregningsmodeller for Tempero Energitjenester sine kunder?»

Vi la fokuset vårt på sikkerhetsfølelsen til brukeren og på at løsningen skulle være brukervennlig, ettersom vi tenkte at dette ville være de viktigste punktene i systemet vi skal designe, utenom selve funksjonaliteten. Beregningsmodellene som skal brukes i systemet kan være veldig kompliserte, noe som kan påvirke brukeropplevelsen. Vi ønsket derfor å undersøke hva som er den beste måten å presentere beregningsmodellene for kunder, og samtidig sørge for at de har en trygg opplevelse ved selvstendig bruk av disse modellene.

I oppgaven vi fikk fra Tempero la de også trykk på at de ønsket en måte å veilede kundene sine gjennom systemet. Vi ville derfor også se på hva som er den beste måten for Tempero Energitjenester og kundene deres å gjennomføre veiledning i bruken av beregningsmodellene i dette systemet.

Ettersom oppdragsgiveren ønsket at brukere skulle kunne leie betalingsmodeller gjennom systemet, dannet vi oss tidlig en hypotese om at ulike betalingsløsninger ville gi ulik trygghetsfølelse hos brukere. Vi ønsket derfor også å undersøke dette, for å finne ut hva som ville være den beste løsningen å implementere i systemet.

Målene våre for prosjektet

I starten av prosjektet, etter noen møter med oppdragsgiver, ble vi enig om målene våre for prosjektet. Disse målene var åpne og innholdet kunne forandres underveis, men vi gjorde dem lukkede nok til å holde oss på rett bane.

Målene vi satte for prosjektet var som følger:

- Gjøre godt med research
- Finne passende teknologier for systemet
- Utføre spørreundersøkelser for å få god innsikt i brukergruppen
- Lo-fi prototyping og brukertesting av prototypene
- Hi-fi prototyping og brukertesting av prototypene
- Kode minst en beregningsmodell

Vi hadde fokus på å utføre research, for å skape et solid grunnlag for bacheloroppgaven vi skal gjennomføre i neste semester. Inkludert under research var undersøkelse av teknologier, norske lover, design av brukeropplevelse og beregningsmodellene til Tempero.

I tillegg til å intervjue vår kontaktperson i Tempero, som vi vil komme tilbake til senere i rapporten, bestemte vi oss for å utføre en brukerundersøkelse hvor respondentene skulle være i den rette målgruppen.

Vi hadde også satt et mål om å utvikle flere iterasjoner av både lo-fi og hi-fi prototyper, med brukertesting av hver iterasjon.

Til slutt valgte vi å kode minst en beregningsmodell, for å sette oss inn i hovedfunksjonaliteten i systemet.

Sammendrag

Videre i rapporten vil vi fortelle om hvordan vi organiserte prosjektet. Deretter vil vi legge frem hvordan vi har gjort research på norske lover, teknologiske løsninger, betalingsløsninger, brukergrensesnitt og beregningsmodellene vi fikk fra Tempero. De viktigste metodene vi har brukt gjennom prosjektet er intervjuer, spørreundersøkelse, side- og funksjonalitetskart, prototyping og brukertesting.

Deretter vil vi presentere svarene vi fikk i spørreundersøkelsen vi gjennomførte, konseptet vi kom opp med og data fra brukerundersøkelsen av prototypen. Vi har også skrevet om hvordan vi laget en kodet versjon av en av Tempero sine beregningsmodeller for å teste ut om researchen vi hadde funnet kunne implementeres.

Etter dette vil vi legge frem våre egne tanker rundt de ulike forskningsspørsmålene våre, sett i lys av research og relevant teori. Vi ser også på hvorfor vi endte opp med

kun én iterasjon av prototypen vår og går gjennom hvilke av prosjektmålene vi har gjennomført.

Til slutt vil vi oppsummere drøftingen rundt forskningsspørsmålene og presentere en konklusjon hvor vi legger frem hvordan resultatene vi har samlet i prosjektet svarer på hovedproblemstillingen vår.

Gjennomføring av prosjektet

I denne delen av rapporten skal vi fortelle om researchen vi har gjort, metodene vi har brukt og generelt om hvordan vi har planlagt og gjennomført prosjektet.

Organisering

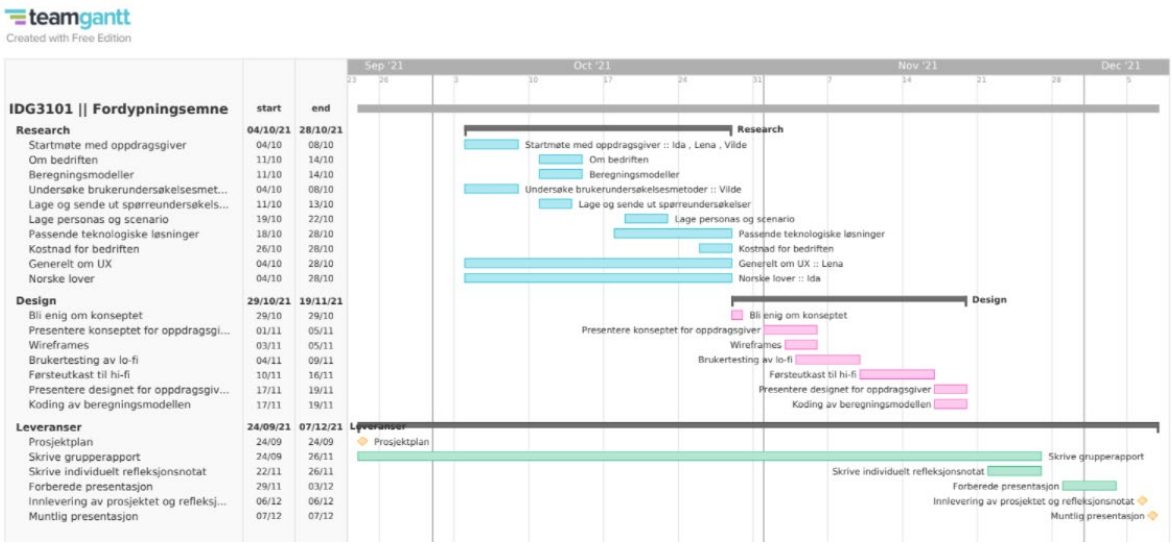
Prosjektgruppen vår består av tre medlemmer som har jobbet sammen i tidligere prosjekter. Ett av medlemmene våre hadde ikke mulighet til å møte opp fysisk, derfor har alt av prosjektgjennomføringen blitt gjort digitalt. Grunnet COVID-19 pandemien har vi allerede god erfaring med dette.

Arbeidet som ble gjort i dette prosjektet ble fordelt mellom alle i gruppa. Vi bestemte oss for å bruke ledelsesmodellen «flat struktur» (Andersen og Schwencke, 2020, s. 63). En gruppe med flat struktur har ingen leder, og ansvaret blir fordelt likt mellom alle gruppemedlemmene.

For å gjøre prosjektet så effektivt som mulig ble vi enige om ulike organiseringsverktøy vi skulle bruke underveis. Vi brukte Google Disk til å samle all research, og andre dokumenter, på samme sted. Vi hadde også en felles Google Kalender for å holde oversikt over møter, agenda for møtene og alle innleveringsfrister. Miro ble brukt til brainstorming (brainstorm, 2021) og vår primære kommunikasjonsmetode, for både skriftlig og muntlig kommunikasjon, var Discord.

For å få oversikt over selve prosjektet brukte vi TeamGantt. TeamGantt er en programvare som lar deg lage Gantt-diagrammer på nett. Gantt, som står for «Generalized Activity Normalization Time Table», er en måte å visualisere prosjektets tidsplan på (Rolstadås, 2021). Vi brukte Gantt for å holde oversikt over hvem som skulle gjøre hva til hvilken tid, og for å se hvilke oppgaver som var avhengige av at andre oppgaver ble utført først.

Vi delte prosjektet inn i tre deler: research, design og innleveringer (Se figur 1). Når noen hadde gjort oppgaven sin kunne de markere oppgaven som ferdig, slik at resten av gruppa fikk beskjed. Vi hadde også ukentlige møter hvor vi oppdaterte hverandre om hva som hadde blitt gjort, og fordelte nye oppgaver.



Figur 1 – Vårt Gantt-diagram

Intervju med oppdragsgiver

Tidlig i prosjektet satte vi opp et intervju med vår kontaktperson hos Tempero Energitjenester, med mål om å finne ut mer om hva oppdragsgiver ønsket at vi skulle gjøre.

Vi satte opp en enkel intervjuguide, som skulle hjelpe oss med å holde struktur gjennom intervjuet (Andersen og Schwencke, 2020, s. 134). Intervjuguiden inneholdt en liste med spørsmål som skulle gi oss svar på tanker som hadde dukket opp etter at vi startet prosjektet. Disse spørsmålene omhandlet blant annet informasjon om beregningsmodellene og brukergruppen til systemet.

Intervjuet foregikk over nett, hvor en av oss ledet intervjuet og de to andre noterte ned svar og tanker, som vi deretter gikk over i gruppa.

Nedenfor er en kort oppsummering av de viktigste svarene vi fikk:

- Designet måtte svare til fagfolk; det må ikke hindre effektiviteten til brukeren
- Tidsperiodene som brukerne vil ha behov for ulike modeller kan variere ut ifra deres prosjekter, så et godt alternativ for betaling vil være å leie modeller

- Man bør kunne leie enkelte beregningsmodeller, samt pakker med flere beregningsmodeller som er relatert til hverandre
- Flere fra samme firma skal kunne logge seg inn og bruke beregningsmodellene
- Oppdragsgiver ønsket å kunne veilede brukerne inne i systemet
- Firmaene som bruker tjenestene til Tempero kommuniserer hovedsakelig på norsk

Norske lover og regler

I starten av prosjektet gjorde vi litt research om norske lover og regler, for å forsikre oss om at løsningen vår ikke brøt med noen av disse. Hovedsakelig fokuserte vi på forskriftene om universell utforming av IKT-løsninger og lover som angår personvern.

Universell utforming

I Norge har vi lover på plass som skal sikre at alle webløsninger er tilrettelagt for flest mulig mennesker. «Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)-løsninger» (2019) stiller krav om at alle nettløsninger skal oppfylle 35 spesifikke kriterier fra standarden «Retningslinjer for tilgjengelig webinnhold (WCAG) 2.0» (W3C, 2011). Vi gikk derfor gjennom hvert av disse punktene og noterte oss hvilke som var relevante for prosjektet vårt og egne tanker rundt hvordan vi skulle løse disse.

Personvern

Ettersom løsningen vi skulle lage hadde behov for et brukersystem, og skulle behandle personopplysninger, så vi oss nødt til å undersøke retningslinjer rundt personvern. De viktigste tingene vi fant i vår research var at man bør begrense lagringen av sensitive personopplysninger, slik som etnisk opprinnelse, religion og seksuell orientering (Datatilsynet, 2019). Virksomheten har ansvaret for å ha full oversikt over hvordan de behandler personopplysningene og bør ha egne retningslinjer på plass for å kunne gi sine brukere innsyn i personlige data ved forespørsel.

Vi har i utgangspunktet ikke noe tenkt behov for å lagre noe som er klassifisert som sensitive personopplysninger. Når databasene blir utformet ved videreutvikling av prosjektet, vil vi ta en vurdering på hvilke personopplysninger som skal lagres i

systemet og begrense disse mest mulig. Det vil også være viktig for oss å skrive ned hvorfor disse opplysningene blir lagret, slik at oppdragsgiver senere kan gi denne informasjonen til sine kunder dersom de ber om innsyn i lagrede data.

Utover dette har gruppa diskutert at med unntak av opplysninger kundene gir om sin bedrift, så velger de selv hva de ønsker å gi av info når de oppretter egne brukere i systemet. Hvis kundene kun ønsker å bruke fornavn, og ikke registrere noe etternavn i systemet, så tenker vi at det er et valg de skal få frihet til å ta selv. Vi ønsker også å være flinke til å informere om hvilke data som blir lagret underveis når brukerne registrerer seg, slik at de kan føle seg trygge på hvordan opplysningene deres blir behandlet.

Design av brukergrensesnitt

Jakob Nielsen er en av skaperne av Nielsen Norman Group og han har lagt frem 10 generelle prinsipper for et godt interaksjonsdesign. De er heuristiske, altså enkle fremgangsmåter eller strategier, og *ikke* spesifikke regler du må følge (Nielsen, 2020).

Noen av disse prinsippene er relevante for oss fordi to av forskningsspørsmålene våre omhandler sikkerhet og brukervennligheten i webløsningen. Prinsippene vi hadde i bakhodet gjennom hele prosessen var:

- Synlighet av systemstatus
    - Løsningen bør holde brukeren informert om hva som skjer og gi tilbakemelding når de utfører handlinger
- Konsistens og standarder
    - Når man, som utvikler av designet, har bestemt seg for et system, bør man fortsette å bruke det på hele siden. Brukeren skal ikke lure på hva ting betyr. Det er også smart å følge standarden til andre like nettsider
- Feilforebygging
    - Om brukeren gjør noe feil, bør man ha gode feilmeldinger så de forstår hva de har gjort galt og hva de skal gjøre for å fikse feilen
- Estetisk og minimalistisk design
    - Brukergrensesnittet bør ikke inneholde informasjon som sjeldent blir brukt. Jo mer ting siden inneholder, jo vanskeligere er det for brukeren å finne det de faktisk trenger

Ikke alle disse prinsippene kommer fram i vår lo-fi prototype, men vi vil ta de med oss videre når vi skal lage en hi-fi prototype og utvikle webløsningen i neste semester.

Teknologiske løsninger

Før vi ferdigstilte konseptet vårt, ønsket vi å gjøre noe research på teknologiene vi skulle bruke i utviklingen av webløsningen. Hovedpoenget med dette var å undersøke hvilke muligheter, og eventuelt begrensninger, vi hadde, slik at vi ikke lovet bort mer enn vi kunne holde.

Betalingsløsninger

Systemet vi skal utforme har behov for integrering med betalingsløsninger. Ettersom vi ønsket å forske på sikkerhetsfølelsen til brukerne i systemet valgte vi å se på de mest kjente betalingsløsningene. Vi tenkte at logisk sett vil brukere føle seg tryggere på betalingsløsninger de har brukt tidligere. Dette er også første gang noen av oss i gruppa kommer til å jobbe med betalingsløsninger, så vi ønsket å finne noe som var godt dokumentert. Etter litt brainstorming valgte vi å se på betalingsløsningene Vipps, Klarna, PayPal og Stripe.

For hver av disse kikket vi på hva Tempero Energitjenester måtte betalt for å bruke tjenestene deres, om brukerne av systemet som er bedrifter vil ha mulighet til å bruke tjenesten, om det er åpen bruk uten å måtte opprette en egen avtale og om brukere må registrere en egen konto. Vi kikket også litt på dokumentasjonen deres, men alle de aktuelle løsningene så ut til å være godt dokumenterte. Nedenfor, i tabell 1, kan du se en sammenligning av de andre punktene vi kikket på.

	Vipps	Stripe	PayPal	Klarna
Priser	* Kun betaling ved bruk * 2,99% av kjøpesummen per transaksjon	* Kun betaling ved bruk * 2 kr fast gebyr per transaksjon * 2.4% av kjøpesummen for norske bankkort * 2.9% av kjøpesummen for internasjonale bankkort	* Priser varierer ut fra valgt tjeneste * 2,80 kr fast gebyr * 1,9% av kjøpesummen for kjøpere uten konto * 3,4% av kjøpesummen for kjøpere med konto	N/A. Pris ut fra bedriftsavtale.
Bedrifter kan betale via tjenesten	Nei	Ja (med firmakort)	Ja	Nei
Åpen bruk uten avtale med bedriften	Nei	Ja	Nei	Nei
Kunde må registrere konto via tjenesten	Ja	Nei	Ja / Nei *	Nei

** Noen av løsningene til PayPal krever ikke konto for betaling, men bruk av portalen o.l. vil ha krav om det.*

Tabell 1 – Sammenligning av løsningene

Noen av løsningene hadde klare problemer som vi ikke kunne overse. Betalingssystemet til Vipps er kun tilgjengelig for privatkunder, og vi måtte også opprettet en avtale med dem for å få lov til å teste systemet.

Klarna hadde gjemt all informasjon om priser, og man måtte kontakte salgsavdelingen for å få opprettet en avtale. Det er ikke noe vi kan gjøre på vegne av firmaet, så dette ble også uaktuelt.

PayPal hadde en hel del kompliserte gebyrer på toppen av alle tjenester, og det virket som at dette fort kunne bli dyrere enn man antar.

Stripe var det eneste alternativet hvor man fritt kunne sette opp systemet uten noen avtale. Man betaler kun per transaksjon som blir gjennomført og det er ingen krav om at brukere må opprette konto, ettersom Stripe er en løsning for å la kunder betale med eget bankkort.

Vi konkluderte derfor med at Stripe var det beste alternativet for vår oppdragsgiver, men ville allikevel undersøke hvilke av disse betalingsløsningene brukerne våre følte seg trygge på gjennom brukerundersøkelsene vi gjennomførte senere.

Kvitteringer

Vi har også vært i kontakt med en autorisert regnskapsfører for å høre om hvordan det vil fungere i forhold til kvitteringer, spesielt ved salg mellom bedrifter. Regnskapsføreren mente at det må presenteres en offisiell kvittering, som må inneholde ting som organisasjonsnummer, pris og merverdiavgift. Det holder ikke for bedrifter at de leverer bankkontoutskrift med kjøpesummen til regnskapsfører. Vi ble derfor nødt til å høre med oppdragsgiveren vår om de hadde et regnskapssystem vi kan integrere disse løsningene med.

Rammeverk

Vi gjorde også noe research på ulike rammeverk vi kunne bruke til å utvikle systemet etter designprosessen. Med tanke på gruppa sin kompetanse, og kunnskap om hvor mye tid vi hadde til rådighet, så visste vi at vi ønsket å holde oss til teknologier vi hadde brukt tidligere, eller lignende teknologier, slik at vi kunne være mest mulig effektive.

Det finnes massevis av ulike rammeverk vi kan bruke til vår front-end. Vi ønsket å bruke et komponent-basert rammeverk for å spare tid, ettersom flere av sidene vi må bygge i prosjektet vil ha mye likt innhold, som for eksempel i beregningsmodellene. Rammeverket måtte også være basert på programmeringsspråket JavaScript, ettersom det er språket som flest i gruppa er komfortable med. Det må også ha

mulighet til å lage alle funksjonene vi har blitt enige om. I tillegg er det greit om rammeverket er godt dokumentert.

React er, ifølge Stack Overflow sin årlige spørreundersøkelse for utviklere, det mest brukte rammeverket på verdensbasis (Stack Overflow, 2021). 41.4% av profesjonelle utviklere i undersøkelsen svarte at det er et rammeverk de bruker. Vue er et annet Javascript-rammeverk som har vokst veldig i popularitet de siste årene. I Stack Overflow Survey 2021 var det hele 20.09% av profesjonelle utviklere som svarte at de har brukt Vue.

Vi vurderte disse to rammeverkene opp mot hverandre, og ser at i forhold til vårt prosjekt vil begge passe bra. Med tanke på at alle i gruppa har brukt React tidligere, så bestemte vi oss for å holde oss til det vi var kjent med.

Vi vurderte også om vi skulle bruke Next.js, som er et rammeverk på toppen av React. Dette ble vurdert fordi React hovedsakelig skaper enkeltside-applikasjoner som blir gjengitt fra klienten til brukeren, noe som ofte ikke er ideelt for søkemotoroptimalisering (Hund, 2016).

Når en søkemotor henter ut informasjon om nettsider, vil den hente ut metadata som finnes på alle undersidene til nettsiden, og vil også sjekke hastigheten på sidene, som påvirker hvordan siden blir rangert blant andre søkeresultater. React sitt oppsett gjør at denne prosessen går veldig tregt på grunn av manglende hyperlenker, noe Next.js løser ved at de serverer ulike sider fra en server som gjør at hyperlenkene er intakte og fungerende. Oppdragsgiveren vår nevnte i intervjuet vi gjennomførte at de ønsker at kunder skal kunne finne frem til siden på egenhånd. For å øke sjansene for at dette skjer, tenkte vi at søkemotoroptimalisering var viktig og vil derfor vurdere å bruke Next.js når vi kommer til utviklingsfasen i neste semester.

Beregningsmodeller

Hovedfunksjonaliteten i webløsningen vi skulle utforme var bruken av beregningsmodellene. Tempero Energitjenester har mange beregningsmodeller, med store variasjoner i kompleksitet og hvor mye data som skal fylles inn. Vi spurte derfor oppdragsgiver om å få tilsendt noen av disse modellene for å kunne analysere hvordan vi overfører bruken av disse til en webløsning. Tempero sendte oss fem ulike beregningsmodeller, med ulik kompleksitet, som vi har tatt en kikk på.

Vi valgte å fokusere på tre ulike områder da vi analyserte disse: hvordan funksjonaliteten kan overføres til kode, hvordan gjøre oppsettet av skjemaene så enkle som mulige for brukerne av systemet og hvordan sørge for sikkerhet i form av å begrense feil som brukere kan gjøre under utfylling.

Informasjon

En av de mest tydelige utfordringene vi møtte på var mengden informasjon som formidles gjennom hvert av skjemaene. Ett enkelt felt i et skjema kan ha behov for flere setninger som forklarer hva brukeren skal fylle inn. Det er også ofte tilleggsinformasjon som kan være relatert til en gruppe med felter, forklaringer som gjelder hele skjemaet og resultatene som brukeren får etter ferdig utfylling. Teksten i noen av resultatfeltene vil også forandre seg ut fra hva resultatet blir.

Å ha store mengder informasjon på en nettside tar opp mye plass, gir brukere en større kognitiv belastning og de fleste brukere leser uansett ikke informasjon eller instruksjoner før de allerede har gjort flere forsøk på å få til oppgaven på egenhånd (Krug, 2014, s. 51). Instruksjoner som vises på siden bør derfor kuttes ned til det mest nødvendige. I vårt tilfelle kan mye av informasjonen være nødvendig for brukerne, som nå skal fylle ut disse skjemaene på egenhånd for første gang. Vi tenker at etter hvert som brukerne blir mer kjent med skjemaene og hvordan nettsiden fungerer er det ikke lenger nødvendig at all informasjon vises på skjemasiden. Vi vil allikevel at informasjonen skal være lett tilgjengelig for brukeren når de trenger den.

Løsningen vi kom opp med ble derfor å plassere hjelpsom informasjon og instruksjoner inn i skjulte tekstbokser som ble vist når man trykket på et ikon. Vi vurderte både et «spørsmålstegn»-ikon og et «info»-ikon (Se figur 2), men ble enige om at vi likte spørsmålstegnet best. Dette valget ble tatt ettersom vi så for oss at brukerne ville begynne å lete etter dette når de hadde spørsmål, og det ville virke logisk å da klikke på et spørsmålstegn. Et «info»-ikon kan tolkes av enkelte som noe man bør sjekke før man prøver å fylle inn skjemaet selv. Dette var kun en hypotese vi kom opp med, så vi inkluderte dette som et punkt vi ville teste i prototypen vi skulle lage senere.

Figur 2 – «Info»-ikon og «Spørsmålstegn»-ikon

Vi tenker også at noe mer generell informasjon for skjemaet kan være nyttig å ha fast på siden. Derfor vil vi også teste hvordan brukerne reagerer på å ha en tekstboks med informasjon ved siden av skjemaet, plassert slik at de lett kan ignorere den dersom de ønsker det.

Vi så også en trend i alle beregningsmodellene der det var mer enn én skjemaetikett koblet til ett enkelt felt. Dette vil skape problemer dersom man prøver å gjøre det samme på en nettside. Dersom et inndatafelt har flere tilhørende skjemaetiketter, kan dette bli forvirrende for brukere som navigerer via bruk av skjermlesere, tastatur eller lignende. Vi prøvde oss frem med litt ulike løsninger for dette (Se figur 3), hvor beskrivelsen og måleenheten ble plassert på ulike plasser rundt inndatafeltet. Vi endte opp med et alternativ hvor beskrivelsen står på toppen av feltet som en vanlig skjemaetikett, måleenheten står på høyre siden av feltet for enkel lesbarhet og instruksjonsknappen står til venstre for skjemaetiketten. Den ble plassert på venstre side ettersom lengden på skjemaetikettene varierer veldig, og dette ville derfor sørge for en mer stabil plassering. Vi har i tillegg tenkt oss at både instruksjonsknappen og måleenheten må gjøres tilgjengelige for skjermlesere, og instruksjonsknappen må være mulig å navigere til ved bruk av tastatur.

Figur 3 - Alternativer for plassering av skjemaetiketter

Begrensning av feil

En utfordring vi får i forhold til sikkerhet er mulighetene for feil utfylling fra brukere. Ettersom nøyaktige resultater er ekstremt viktig for prosjektene til brukergruppen vår, så ønsket vi å redusere mulighetene for at noe kan gå galt under utfyllingen. For å fikse dette vil vi først og fremst bruke funksjonalitet som allerede eksisterer i HTML.

Alle inndatafeltene bør være av typen «nummer» (Mozilla, 2021), som automatisk vil avvise alle forsøk fra brukere på å sette inn andre type verdier enn tall. Vi vil også kunne bruke egenskapen for å sette minimums- og maksimumsverdier på feltene, dersom oppdragsgiver er villig til å hjelpe oss med å bestemme hvilke verdier som er logisk for hvert enkelt inndatafelt.

Det er også viktig for oss å kunne bruke den innebygde egenskapen for å kunne velge trinnene som tallet i inndatafeltet skifter mellom. I noen steder i beregningsmodellene vil det kun brukes heltall, mens andre steder er brukere avhengig av at man skal kunne sette inn tall med et visst antall desimaler for at resultatene skal bli presise. Derfor må vi passe på at man kan velge dette for hvert enkelt inndatafelt dersom vi bruker et komponentbasert utviklingssystem senere i prosjektet.

Til slutt er det også viktig at vi kan sette egenskapen «required» i hvert enkelt felt som krever utfylling, noe som vil sørge for at brukeren ikke får sendt inn skjemaet dersom de har glemt å fylle inn noe i et felt.

Matematiske utregninger

I vår research fant vi også en del informasjon om hvor upresise matematiske utregninger kan være i de fleste kodespråk, blant annet i JavaScript som vi skal jobbe med (Petersen, 2021). Hvis vi ikke er påpasselige med testingen av de kodede modellene, kan altså alle resultatene bli feil, noe som kan skape store problemer i prosjektene til brukerne.

For oppdragsgiveren sin del er det også veldig viktig at de beregningene som blir gjort ikke er synlige for folk som bruker siden. Siden beregningsmodellene er ment for å leies, ikke kjøpes, så ønsker de ikke å gi bort de matematiske beregningene. Vi må derfor sørge for at dette ikke er synlig gjennom nettleseren, og vil vurdere en løsning hvor beregningene gjøres på serversiden når vi kommer så langt inn i utviklingen.

Til slutt ble vi enige om at når det kom til kodingen vil vi bruke ukontrollerte inputs i alle skjemaer hvor beregninger blir utført. Dette vil si at når brukeren forandrer verdien i ett av feltene, vil ikke dette umiddelbart forandre på resultatet. Resultatet vil kun bli forandret når brukeren velger å trykke på en knapp. Gjennom vår research om retningslinjene for universell utforming så vi også at punkt '3.2.2. Inndata' (W3C, 2011) sier at endringen av verdien til et inndatafelt ikke skal medføre automatiske endringer på siden. Vi tenker at dette gir mening fordi verdien i ett enkelt felt kan bestå av mange sifre, og en umiddelbar forandring på siden vil kunne distrahere brukeren, som da lettere kan gjøre feil i utfyllingen.

Spørreundersøkelse

Vi valgte å utføre en spørreundersøkelse for å få bedre innsikt i brukergruppen vi skal lage systemet for. Undersøkelsen skulle også oppklare spørsmål vi ikke kunne løse på egenhånd eller ved hjelp av oppdragsgiver, slik som:

- Er et slikt system ønsket av brukergruppen?
- Hva slags type visuelt design bør nettsiden ha?
- Hvilke betalingsmetoder er brukergruppen mest komfortabel med?
- Hvordan ønsker brukergruppen å kommunisere med Tempero Energitjenester?

Analyse

For å lage en god brukerundersøkelse, måtte vi undersøke noen av metodene som finnes. Vi gikk igjennom kapittel syv i boken «Prosjektarbeid: En veiledning for studenter», hvor det er god informasjon om forberedelsene til og gjennomføringen av undersøkelser, bearbeidingen av det innsamlede materialet, og analysen av resultatene.

Brukerundersøkelser er primærmateriale, ettersom det er informasjon man skaffer på egenhånd gjennom vitenskapelige undersøkelser, spesielt utviklet for et prosjekt. De vanligste brukerundersøkelsesmetodene er intervjuundersøkelser, spørreskjemaundersøkelser og observasjoner. Metodene kan være enten kvantitative eller kvalitative, hvor kvantitative metoder befatter seg med tall og det som er målbart. Resultatet av undersøkelsen blir ofte presentert i tabeller, grafer, eller lignende. Kvalitative metoder undersøker et mindre antall personer og er mer

rettet mot tolkningen av resultatene fra respondentene (Andersen og Schwencke, 2020, s. 132).

Vi valgte å gjennomføre en spørreskjemaundersøkelse, fordi det gir større sannsynlighet for å få svar fra respondentene dersom de kan svare når de selv har tid. I tillegg kan respondentene svare på undersøkelsen anonymt, og de kan bruke så lang tid de selv ønsker. En spørreundersøkelse kan både være kvalitativ og kvantitativ, men vi valgte å gjennomføre den kvalitativt, fordi vi hadde tilgang til et begrenset antall personer fra brukergruppa.

Når vi diskuterte og kom opp med spørsmål, unngikk vi å spørre om det meste av personlige opplysninger som kan identifisere respondentene, for å holde undersøkelsen så anonym som mulig. Vi spurte bare om aldersgruppe, rolle i bedrift og ansvarsområde i forbindelse med rehabiliterings- og nybyggprosjekter.

Gjennom undersøkelsen samarbeidet vi tett med vår kontaktperson hos Tempero Energitjenester, for å lage relevante spørsmål og sørge for at fagspråket var forståelig. Han hjalp oss også ved å dele undersøkelsen med sine klienter, for at undersøkelsen skulle nå ut til de mest relevante personene.

Spørsmålene vi stilte omhandlet blant annet om de bruker mobil, nettbrett eller datamaskin i sitt arbeid, betalingsmetoder de er kjent med, deres preferanser for nettsidedesign, og bruk av beregningsmodeller. Disse spørsmålene hadde fokus på sikkerhetsfølelse, spesielt når det kom til betalingsmodell, for å se hva brukergruppen føler seg mest trygge på.

Vi brukte Google Skjemaer for oppsett av spørreskjema. Google Skjemaer gjør det lett å både sette opp undersøkelser og analysere svarene gjennom automatisk genererte diagram.

Gjennomføring av spørreundersøkelsen

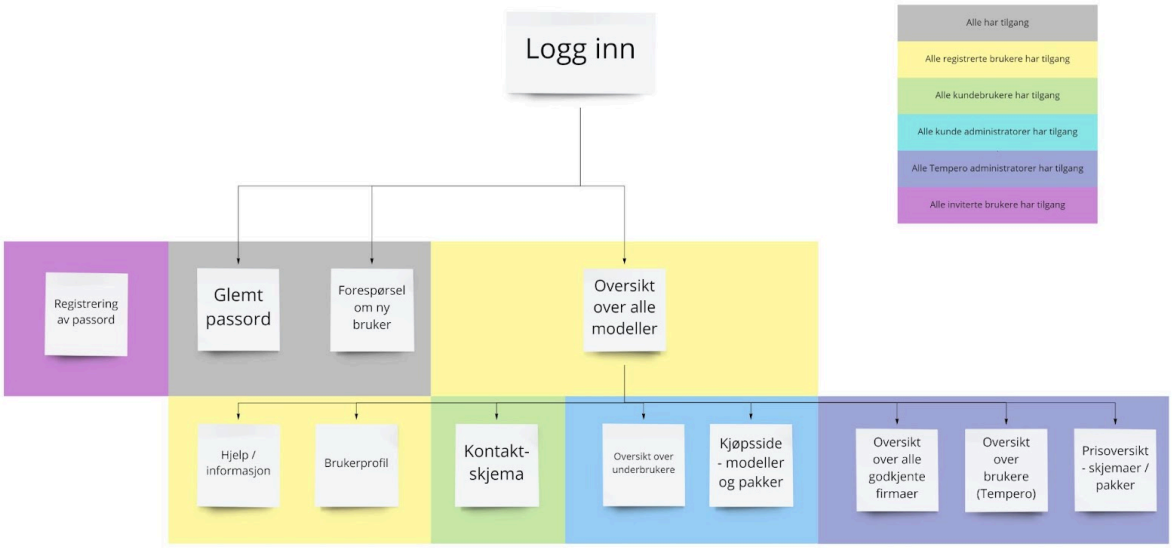
Før vi kunne sende ut spørreundersøkelsen, måtte den testes. Vi pilottestet undersøkelsen på en ekstern testperson, samt oppdragsgiver, som vil si at vi gikk igjennom undersøkelsen som om de var del av brukergruppen (Schade, 2015). Gjennom pilottestingen oppdaget vi flere feil, som gjorde at vi måtte skrive om flere spørsmål og vrake noen av dem. I tillegg bidro Tempero med flere aktuelle spørsmål som vi la til.

Vi ga respondentene en tidsfrist på en uke til å svare på spørreundersøkelsen. Deretter startet vi med å analysere resultatene.

Side- og funksjonalitetskart

Etter å ha sett på kravene fra prosjekteier og svarene fra spørreundersøkelsen, begynte vi å brainstorme konseptet til webløsningen. For å komme i gang bestemte vi oss for å lage et sidekart.

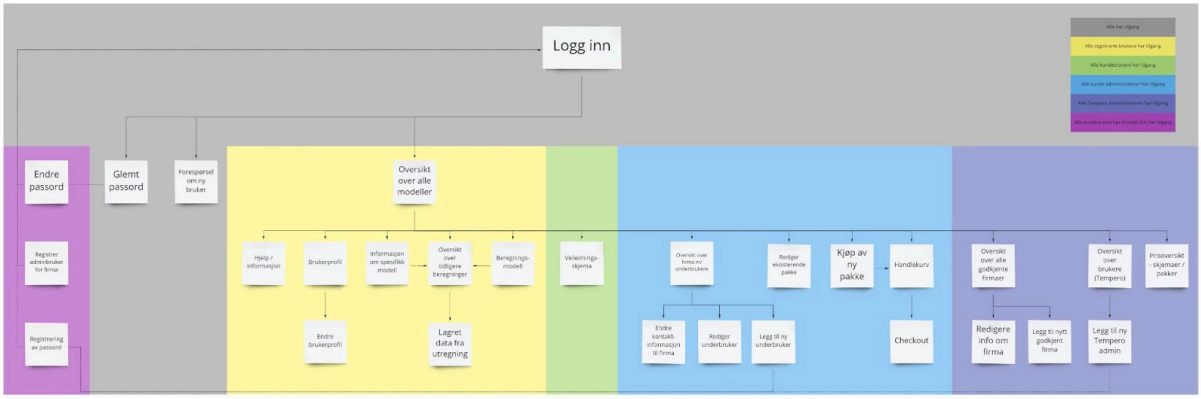
Vi valgte å lage et sidekart fordi det hjalp oss med å visualisere informasjonsarkitekturen til websiden før den ble laget. Måten man lager et sidekart på er at man tar hovedidéene og underkategoriene man har kommet opp med og setter de inn i et diagram av bokser forbundet med piler, for å vise hvordan sidene henger sammen (Lopuck, 2012). Se figur 4 for første versjon av sidekartet vårt.



Figur 4 – Første versjon av sidekart

I prosessen med å lage sidekart kom vi opp med mange idéer. Derfor bestemte vi oss for å lage en oversikt over hva vi tenker å inkludere i det enkleste brukbare produktet, et såkalt «MVP», og hva vi tenker er kjekt å ha med dersom vi får mulighet til det, også kjent som «nice-to-have». Dette gjorde at vi fikk en visuell oversikt over hva som var nødvendig for denne webløsningen og hva som var mindre viktig (Se figur 1 i vedlegg). Måten vi bestemte om noe var «MVP», var å se på om det var behov for funksjonen i systemet for å oppfylle minimumskravene.

Etter vi laget denne oversikten, så vi at det var mye vi ikke hadde inkludert i sidekartet vårt, derfor laget vi en ny versjon av det (Se figur 5).



Figur 5 – Andre versjon av sidekart

Vi hadde dermed oversikt over hva innholdet på hver underside skulle være, men ikke hvilke funksjoner de inkluderte. Derfor laget vi også et funksjonalitetskart som var basert på sidekartet. Det viser hvilke funksjoner man finner på de forskjellige sidene og hvordan de henger sammen med hverandre.

I figur 6 kan du se et eksempel på funksjonalitetene på siden «Oversikt over alle beregningsmodeller».

Figur 6 – Kommentar i Miro som vises funksjonalitet på en underside

Prototyping og brukertesting

«The only way to really know whether an idea is reasonable is to test it» (Norman, 2013, s. 227)

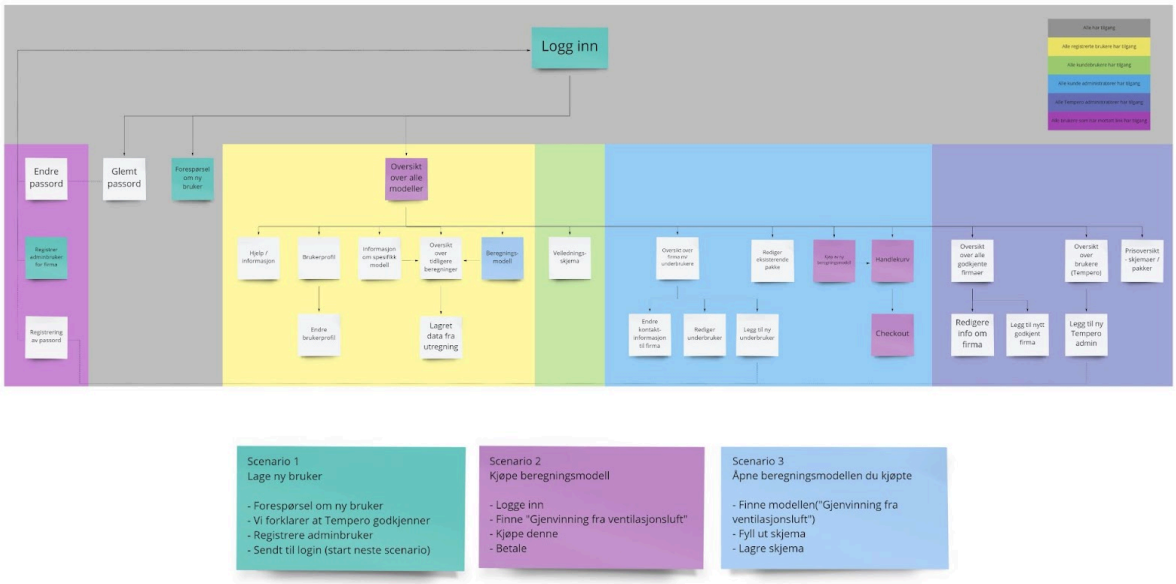
Da vi hadde blitt enige om hvilken funksjonalitet som skulle være på de enkelte sidene, startet vi med å lage en digital prototype for å teste ut idéene våre. Målet med prototypingen var å bli enige om hvordan komponentene på sidene skulle plasseres, og å teste noen av de viktigste funksjonene vi hadde blitt enige om. Det aller viktigste for oss er at systemet er brukervennlig. I denne iterasjonen av prototypen valgte vi å ikke gjøre noe spesifikk testing av sikkerhetsfølelsen til brukeren ved bruk av systemet, ettersom vi tenker at dette kan påvirkes veldig av at testpersonen ikke må bruke sin egen betalingsinformasjon og annen personlig informasjon. Vi tar også høyde for at deler av brukeropplevelsen kan påvirkes av mangelen på estetisk design (Lee og Koubek, 2010).

Ettersom vi hadde gjort en såpass grundig jobb med å kartlegge funksjonaliteten, hadde vi et godt grunnlag for å vite hva vi trengte å gjøre videre. Vi så at en del av sidene ville ha et ganske likt oppsett, og valgte derfor å ikke skissere ut alle sidene, men heller fokusere på noen av de viktigste.

Vi valgte å fokusere på funksjoner som vi tenkte var viktigst for bruken av siden, og ble enige om tre scenarioer å teste på brukere:

- Opprette en ny bruker
- Leie en ny beregningsmodell
- Bruke en beregningsmodell man allerede har leid

Med disse tre funksjonene i tankene, brukte vi teknikken *Journey Mapping* (Gibbons, 2018) og skrev ned undersider og steg som brukere måtte gå gjennom for å kunne fullføre disse oppgavene. Vi prøvde å sette opp stegene i en logisk rekkefølge, slik at en bruker kunne gjennomføre scenarioene etter hverandre under brukertesten. Deretter laget vi en kopi av sidekartet vårt og markerte sidene vi trengte å prototype for å gjennomføre brukertest av disse funksjonalitetene (Se figur 7).



Figur 7 - Sidekart og scenarioer med markerte undersider

Vi ønsket å sette opp den første prototypen sammen som en gruppe, for at det skulle være mest effektivt og vi kunne bli enige om idéene på en og samme tid. Vi satte derfor opp en workshop med gruppa. Under workshopen valgte vi å bruke «Balsamiq». Balsamiq er et nettbasert verktøy hvor man kan lage trådskisser til mobil- eller datamaskin, som ser ut som enkle skisser. Vi ønsket å holde det første designet low-fidelity for å fokusere på plasseringen av elementer og testing av funksjonaliteten på sidene, uten å bruke for mye tid på å planlegge ut fargevalg og andre estetiske detaljer.

I Balsamiq har de også forhåndsskapte komponenter man kan dra inn på siden og tilpasse etter eget ønske, som gjør det mer effektivt enn om man lager alt selv. Flere personer kan også jobbe i prosjektet på samme tid, som gjør det til et flott verktøy for gruppesamarbeid.

Da vi ble ferdige med utkastet til prototypen vår, eksporterte vi sidene til en klikkbar PDF som vi kunne bruke i brukertestingen vi hadde planlagt.

Gjennomføringen av brukertester

Da vi skulle gjennomføre brukertestingen ble vi nødt til å være litt kreative. På grunn av tidsbegrensninger måtte alle tre gruppemedlemmer gjennomføre brukertester hver for seg. Dette ga oss flere utfordringer. Vi måtte sørge for at brukertestene ble gjennomført på en mest mulig lik måte, for å unngå at testingen ble påvirket og ga

oss feilkilder. Ettersom vi skulle gjennomføre testene med kun ett gruppemedlem til stede, så måtte vi også sørge for at vi hadde nok tid til å notere ned alt testpersonene foretok seg under testen, så vi ikke gikk glipp av noe viktig data.

Vi laget oss derfor noen regler for brukertestene. Hver av oss skulle kun teste på én person av gangen, både fordi systemet er ment for denne typen bruk og for å gi oss selv mulighet til å gi mer fokus til denne ene testpersonen. Vi skulle helst teste systemet på personer innenfor aldersgruppa vi hadde kommet frem til gjennom brukerundersøkelsen, men testpersonene behøvde ikke ha kunnskap om fagfeltet som brukergruppa vår jobber i, ettersom det ikke var relevant i noen av scenarioene vi skulle teste. Vi bestemte også at vi skulle gjennomføre to brukertester hver, altså seks brukertester totalt, ettersom man sjelden får ny informasjon etter den femte brukertesten man gjennomfører (Nielsen, 2000). Hvor man gjennomførte testen var opp til hver enkelt, men det var et krav om at man måtte kunne se skjermen og musepekeren til testpersonen for å kunne følge med på hva de foretok seg.

For å sørge for en mest mulig lik gjennomføring ble det skrevet et manus vi skulle lese opp for testpersonene. Manuset startet med en introduksjon av hva vi jobber med i oppgaven og hvordan brukertesten skulle gjennomføres, og ga testpersonene mulighet til å stille spørsmål før vi startet. Det ble skrevet en tekst for hvert scenario og også noen svar på spørsmål vi tenkte kunne dukke opp underveis. Dette kunne være informasjon om hva som skulle stå i enkelte bokser i prototypen hvor det ble brukt fylltekst, hvor dette kunne være relevant for hvordan testpersonene – og også tenkte brukere av systemet – interagerer med sidene.

Det ble deretter laget et eget ark som skulle deles ut til brukeren hvor det sto en beskrivelse av alle scenarioene, i tilfelle de skulle glemme oppgaven som ble gitt til dem. Dette ble gjort for at testpersonen ikke skulle bli stresset av å måtte huske oppgaven, ettersom en tenkt bruker av systemet ville hatt et klart mål i hodet.

Vi ble også enige om at vi skulle be brukerne om å tenke høyt på starten av testen, noe som også ble skrevet inn i manuset. Gruppemedlemmet som utførte testen skulle notere ned alt som ble sagt, ting som ble trykket på og hvor lang tid testpersonen brukte på å utføre de ulike delene av oppgavene. Spesielt viktig var det dersom noe tok lang tid, som kunne indikere at det var noe vi trengte å endre på.

På slutten av hver test hadde vi også skrevet ned noen generelle spørsmål vi ville stille testpersonene for å prøve å få mer informasjon ut av dem som kanskje ikke kom frem underveis i testen. Vi skrev også ned alle de “riktige” stegene for hvert scenario, slik at vi kunne sammenligne disse med hva som faktisk ble gjort etter brukertesten var gjennomført.

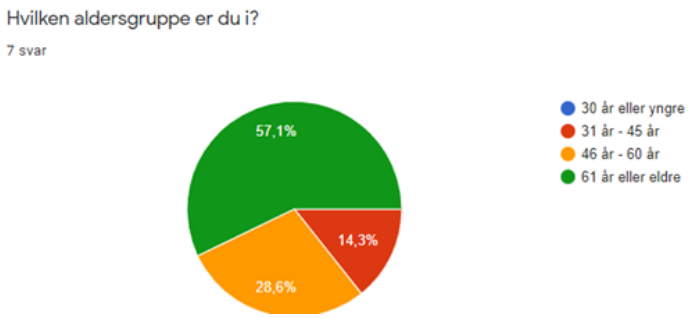
All denne forberedelsen hjalp oss med å gjennomføre gode brukertester, som vi fikk mange nyttige tilbakemeldinger fra. Resultatene fra brukertesten vil bli presentert i neste del av rapporten.

Resultater

I denne delen av rapporten vil vi legge frem resultatene vi fikk ut av researchen og metodene våre. Vi går gjennom data vi fikk fra spørreundersøkelsen, detaljer om konseptet og prototypen vi designet, hva vi fant ut gjennom brukertestingen vi gjennomførte og til slutt forteller vi om prosessen med å kode beregningsmodellen og hva vi fikk ut av dette.

Data fra spørreundersøkelsen

Ut fra spørreundersøkelsen vi gjennomførte, fikk vi syv innsiktsfulle svar. For å starte undersøkelsen, spurte vi et enkelt spørsmål om respondentenes aldersgruppe, som vist i figuren under (Se figur 8).



Figur 8 - Svar om respondenters aldersgrupper

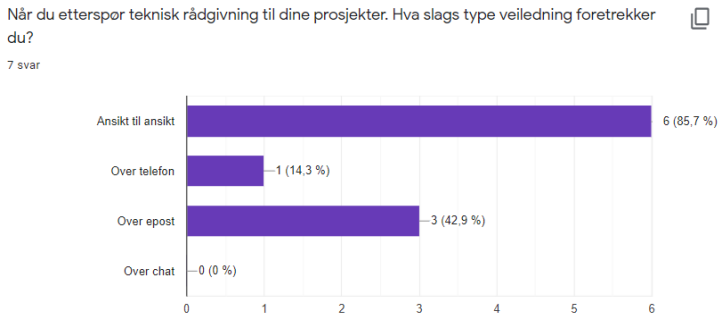
Flertallet viste seg å være i aldersgruppen 61 og eldre. Denne informasjonen ble blant annet brukt for å velge testpersoner til brukertestingen.

Vi spurte deretter et viktig spørsmål angående interessen for webløsningen. Er et system for leie og bruk av beregningsmodeller nyttig for Tempero sine kunder? (Se figur 9).

Figur 9 - Svar om respondentene har nytte av et system for leie og bruk av beregningsmodeller på nett

Alle respondentene svarte «Ja», noe som bekreftet at et slikt system var ønsket.

Deretter spurte vi respondentene om hvilken type kommunikasjon de foretrakk når det kom til veiledning med Tempero. I dette spørsmålet kunne de velge ett eller flere svaralternativer. Nedenfor, i figur 10, vises resultatene.

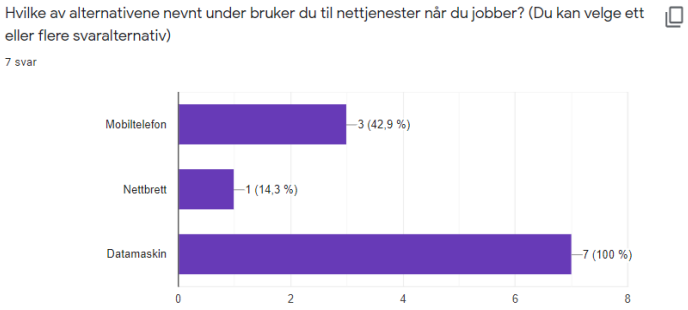


Figur 10 - Svar om foretrukne typer veiledning

Seks av syv respondenter foretrekker at veiledningen foregår ansikt til ansikt, tre foretrekker e-post og en person svarte at de foretrekker telefon. Ingen ønsket en chatfunksjon på nettsiden. I tillegg til å velge hvilke typer veiledning respondentene foretrakk, ba vi dem begrunne svaret sitt. Kort oppsummert mente de ansikt til ansikt er mer effektivt ved behov for forklaringer, samtalen blir mer detaljert og det å komme frem til en løsning er lettere. Én respondent mener at e-post gir best oversikt.

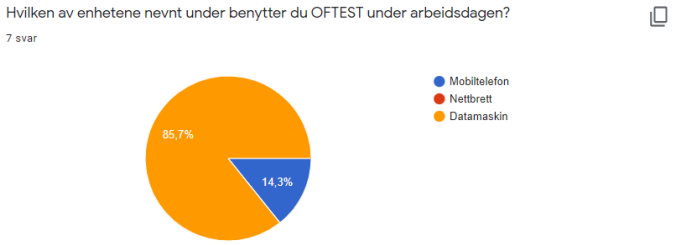
Deretter stilte vi tre spørsmål angående respondentenes bruk av enheter i jobbsammenheng. Det første spørsmålet angikk *hvilke* enheter respondentene

bruker under arbeidsdagen. De kunne velge ett eller flere svaralternativer (Se figur 11).



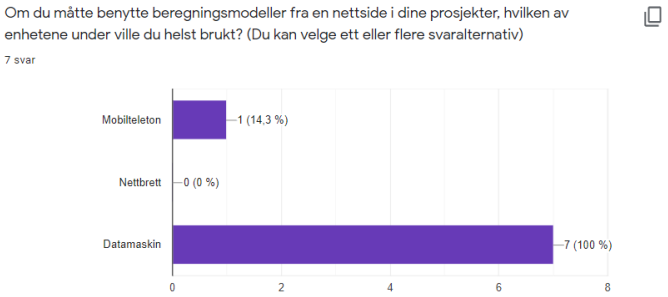
Figur 11 - Svar om bruk av enheter under arbeidstid

Det andre spørsmålet handlet om hvilken enhet de bruker *oftest* (Se figur 12).



Figur 12 - Svar om mest brukte enhet under arbeidstid

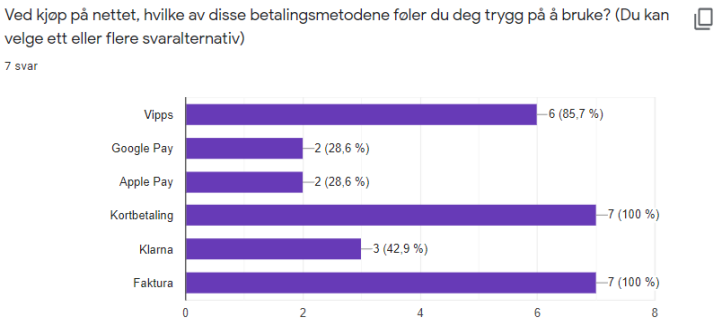
Til slutt spurte vi om hvilke enheter de potensielt *ville* ha brukt i et slikt system. Her kunne de også velge ett eller flere svaralternativer (Se figur 13).



Figur 13 - Svar om ønsket enhet ved bruk av nettside

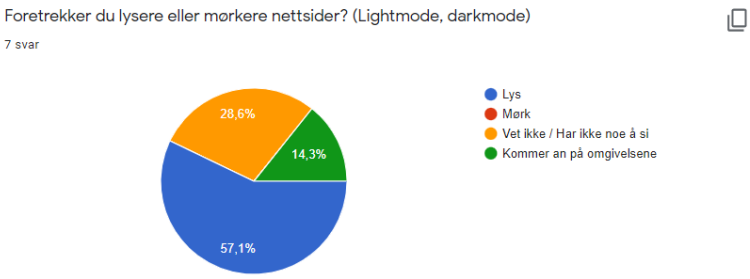
Når det kom til hvilken enhet respondentene ville brukt, ble svaret hovedsakelig «Datamaskin». Svarene vi fikk viser at vi bør fokusere på å designe for datamaskin, men at systemet også trenger en mobilversjon.

Angående betalingsmetoder på nett, valgte vi å stille to spørsmål. Vi spurte både om hvilke betalingsmetoder respondenten hadde hørt om, og fulgte opp med et spørsmål om hva de følte seg tryggest på. Bildene under viser resultatene (Se figur 14 og 15).



Alle hadde kjennskap til Vipps, kortbetaling og faktura, noe vi hadde forventet. Alle svarte også at de følte seg trygge på både kortbetaling og faktura. Dette viste oss at vi kan forholde oss til kun kortbetaling og faktura, og se bort ifra de andre betalingsmetodene.

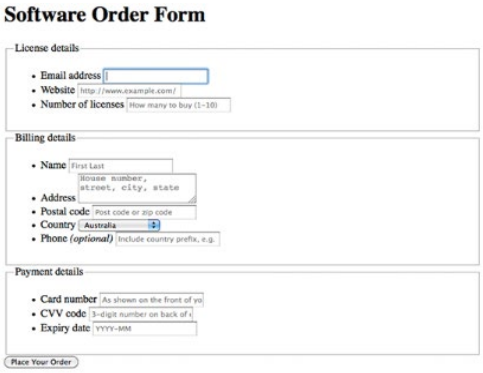
Til slutt spurte vi respondentene to spørsmål angående visuelt design. Det første angikk hva de foretrakk av lyse og mørke nettsider, som vist i figur 16.



Figur 16 - Svar om preferanse mellom lys og mørk nettside

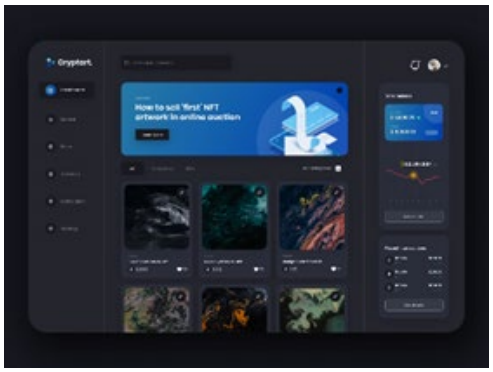
Resultatene viste at folk flest foretrakk lysere nettsider, mens resten enten var usikre eller var likegyldige.

Det andre spørsmålet angående design handlet om hva de likte best av fire forskjellige alternativer, som vist under (Se figur 17, 18, 19 og 20). På grunn av at Google Skjema komprimerer bilder i spørreskjemaene sine, så ble bildekvaliteten veldig lav.

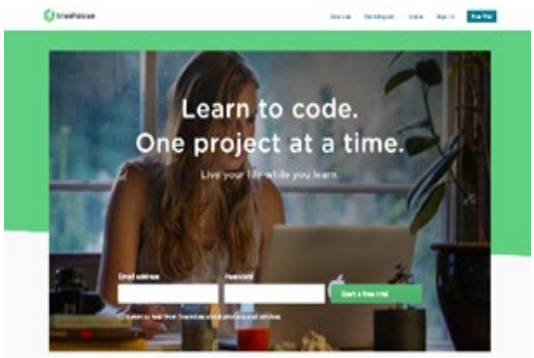


Figur 17 - Alternativ 1

Figur 18 - Alternativ 2

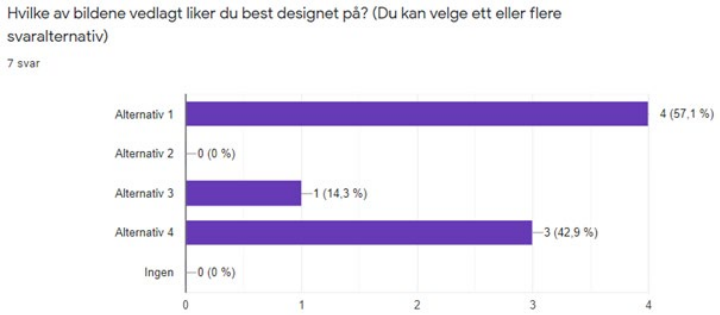


Figur 19 - Alternativ 3



Figur 20 - Alternativ 4

Disse alternativene har alle forskjellige stiler og temaer. «Alternativ 1» har ingen ekstra styling, «alternativ 2» er mer intens med sterk blåfarge og mønster, «alternativ 3» er mørk og oversiktlig, og «alternativ 4» er et eksempel på flatt design (Interaction Design Foundation, 2021). I figur 21 kan du se hva respondentene valgte.



Figur 21 - Svar om preferanse av alternativer for nettsidedesign

De fleste av respondentene svarte at de foretrakk alternativ 1. Dette resultatet viser oss at designet vårt bør være minimalistisk.

Spørreundersøkelsen viste seg å være ekstremt nyttig. Den satte opp et rammeverk vi måtte holde oss innenfor, oppklarte usikkerheter og ga oss generell innsikt i brukergruppens tanker og meninger. Gjennom prosjektet har vi hatt disse resultatene i bakhodet.

Konsept

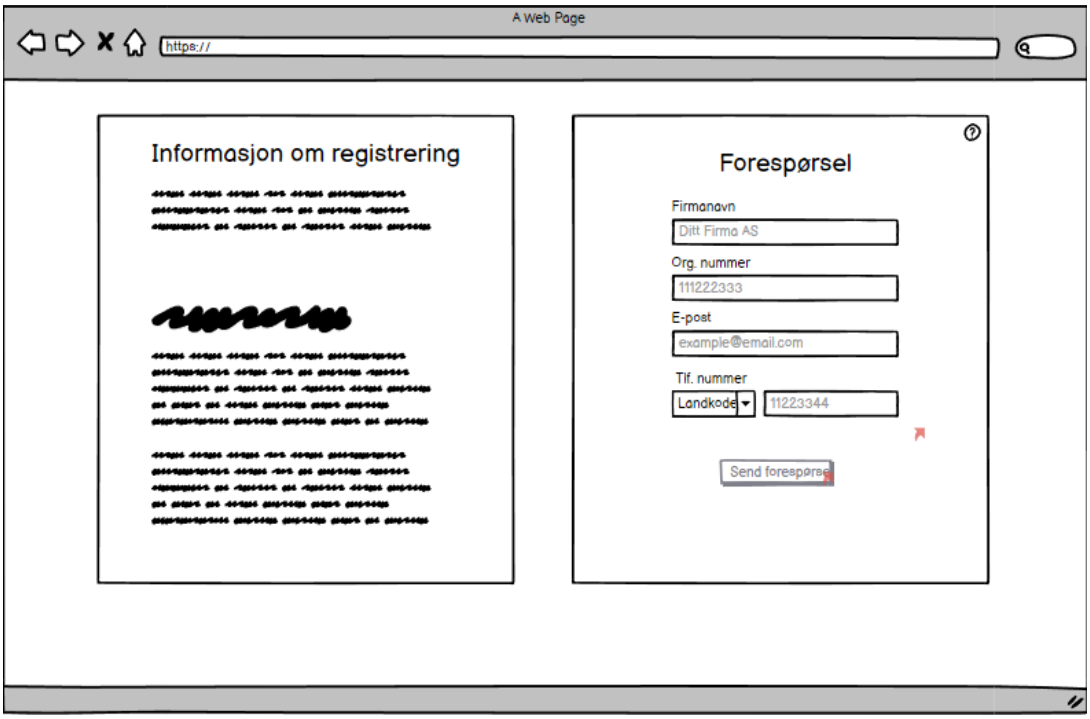
Etter vi hadde gjort grundig research utformet vi et konsept for å få en klar idé om all funksjonalitet vi ville ha med i webløsningen vår.

Basert på våre innhentede data ble konseptet for vår løsning en responsiv nettside som blant annet inneholder et brukersystem, beregningsmodeller, en sikker betalingsløsning, bruk av beregningsmodeller og et system for veiledning.

Registrering av kunder

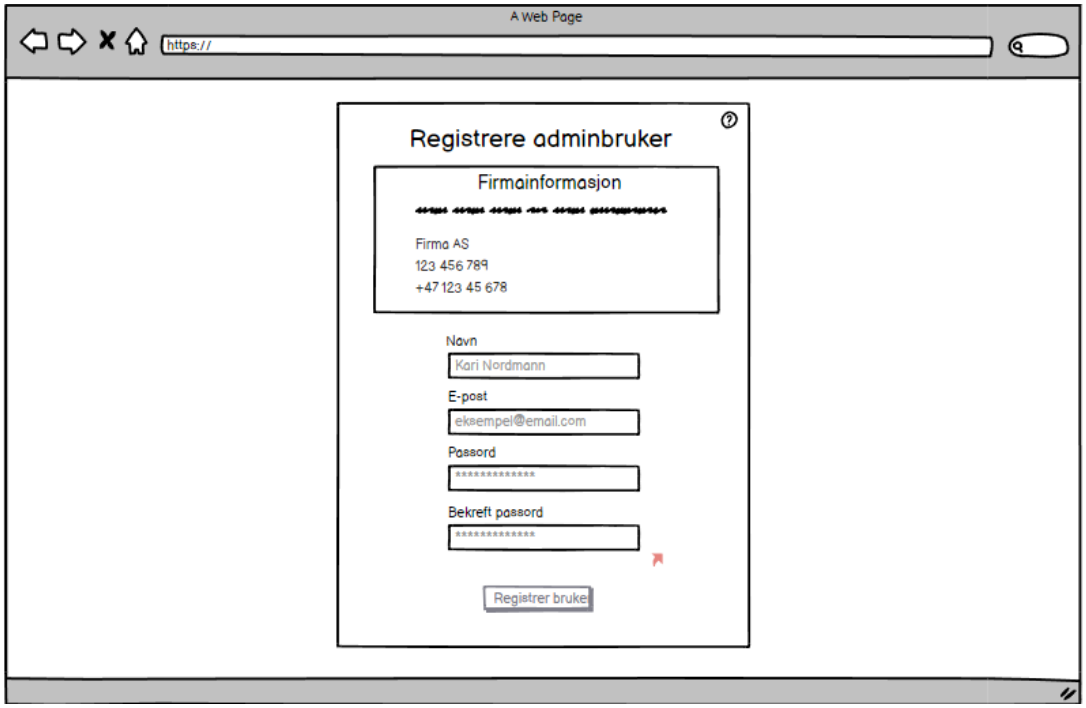
Tempero ønsket seg en registreringsløsning hvor firmaer må søke om tillatelse for å lage en bruker før de registrerer seg. Grunnen til dette var at de ikke ønsket å ha tilfeldige firmaer i systemet som de ikke vet hvem er.

Nye brukere av løsningen vil derfor først måtte registrere firmaet sin kontaktinformasjon via et kontaktskjema (Se figur 22). Tempero vil få beskjed via e-post om at det er en ny bruker som ønsker tilgang til systemet. Dersom Tempero ønsker å godkjenne det nye firmaet, kan de logge seg inn i systemet og legge de til som et godkjent firma, via en egen administratorside.



Figur 22 - Kontaktskjema for registrering av firma

Den nye brukeren vil deretter få en e-post med en automatisk generert link. Hvis brukeren trykker på denne linken kan de registrere en administratorbruker for firmaet sitt (Se figur 23), og deretter logge seg inn.



Figur 23 - Registreringsskjema for firmaadministratorbruker

Brukertyper i systemet

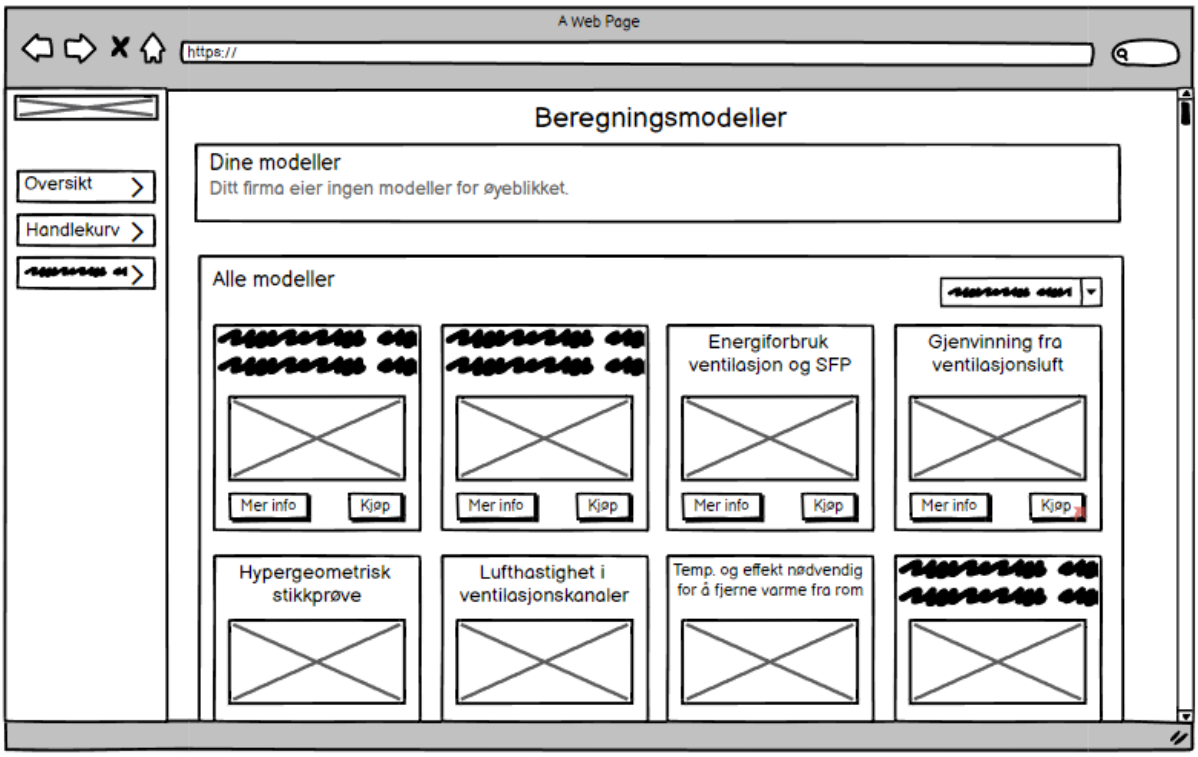
Løsningen vil ha tre ulike brukere: «Firmaadministrator», «Underbruker» og «Systemadministrator».

Firmaadministratorer vil ha mulighet til å opprette opptil fem underbrukere for sitt firma. De har også mulighet til å slette disse om det trengs. Underbrukerne vil ikke ha mulighet til å administrere andre brukere i firmaet.

Tempero vil fra starten av ha én bruker som administrerer alle andre brukere i systemet. Denne brukeren vil også ha mulighet til å registrere ekstra administratorbrukere for ansatte i Tempero, ved behov.

Beregningsmodeller

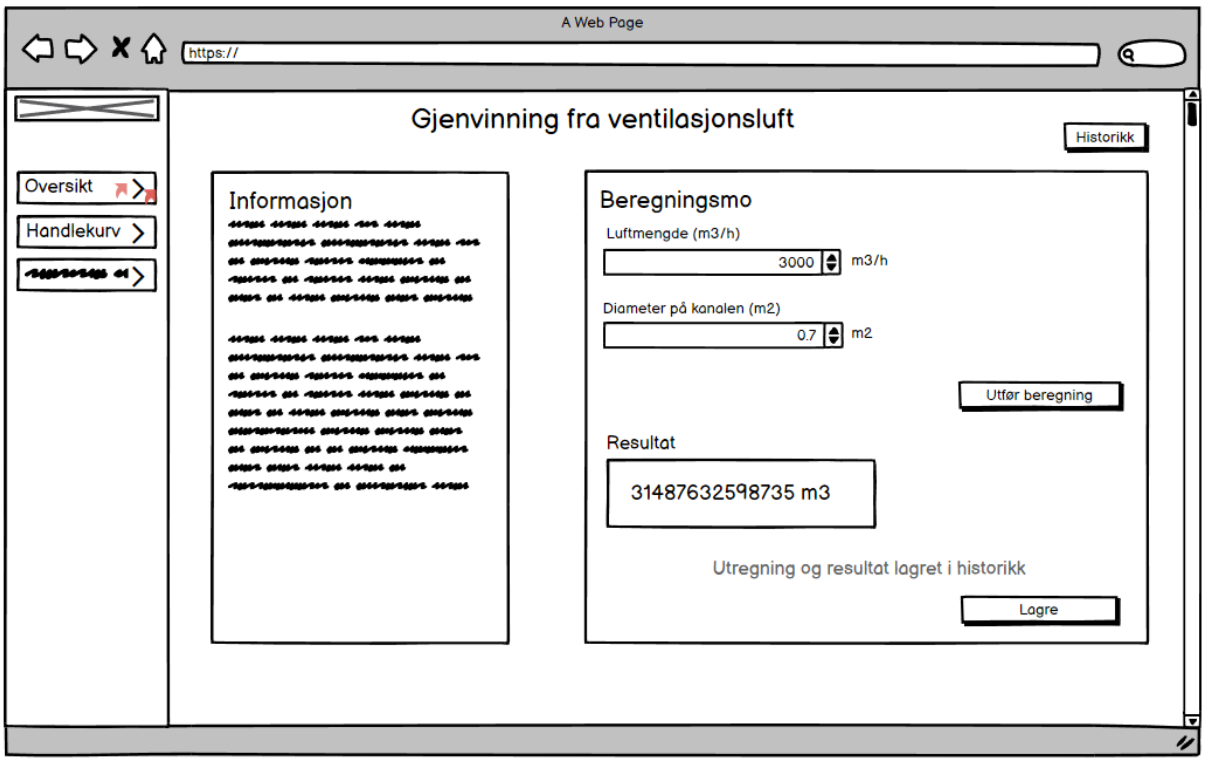
Beregningsmodellene vil være sortert inn i ulike kategorier, og det vil være mulig å filtrere beregningsmodellene etter kategori på hovedsiden. Slik blir det lettere å finne frem blant dem. Vi ønsker også å vise kundene en forhåndsvisning av beregningsmodeller de ikke har tilgang på (Se figur 24).



Figur 24 - Oversikt og forhåndsvisning over alle beregningsmodeller, filtrert etter «Alle modeller»

Når kundene bruker beregningsmodellene så vil beregningene, inkludert informasjonen i alle felter, lagres slik at de kan hente det frem igjen senere (Se figur 25). Inne på siden vil det også være informasjon om hvordan den spesifikke beregningsmodellen fungerer.

Systemadministratorene vil kunne se og bruke alle modellene i systemet. Dette vil være nyttig når Tempero skal gjennomføre veiledning med kundene sine.

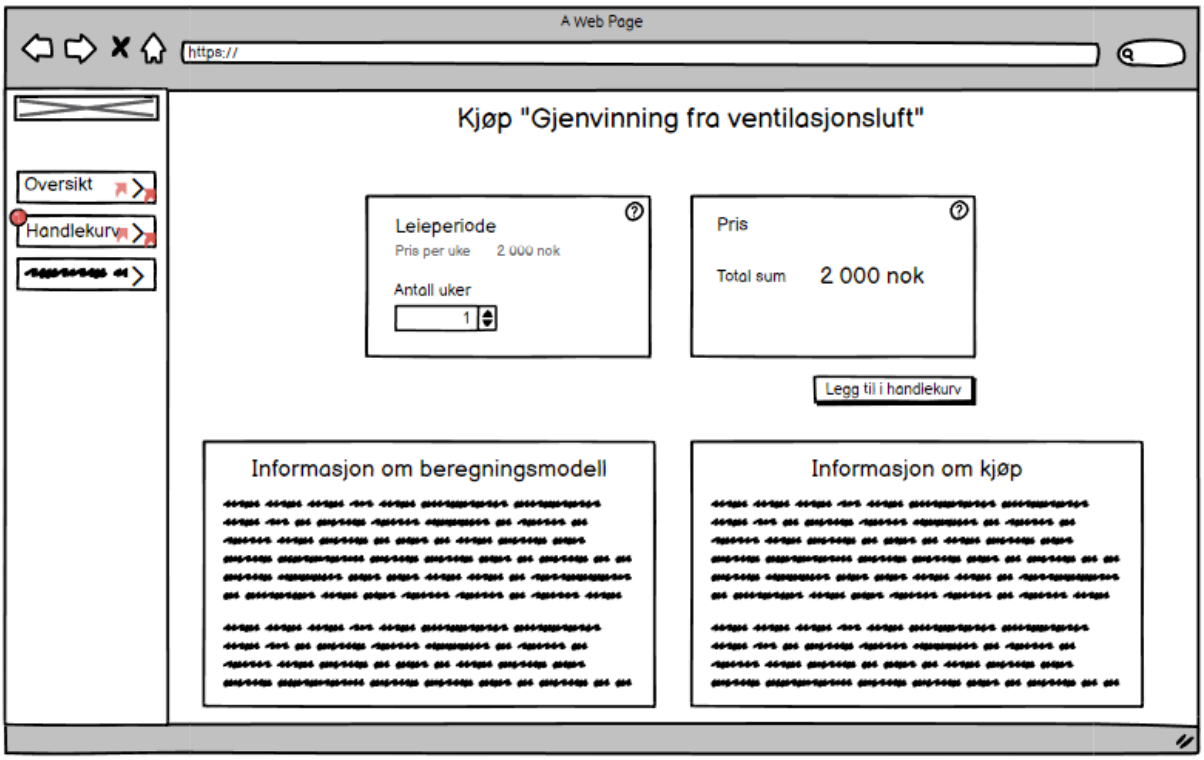


Figur 25 - En beregningsmodell med informasjonsboks og lagret resultat

Leie av beregningsmodeller

I firmaet er det kun administratorbrukeren som kan leie nye beregningsmodeller og pakker, underbrukerne vil ikke ha mulighet til dette. Alle brukere i samme firma vil ha tilgang på de samme beregningsmodellene og lagret data som er tilknyttet disse.

Hver enkelt beregningsmodell og hver «pakke» med beregningsmodeller vil ha en egen pris. Kundene vil ikke eie disse modellene, de leier tilgang for en valgt tidsperiode (Se figur 26). Tempero sine administratorbrukere kan gå inn i systemet og endre prisene etter ønske.

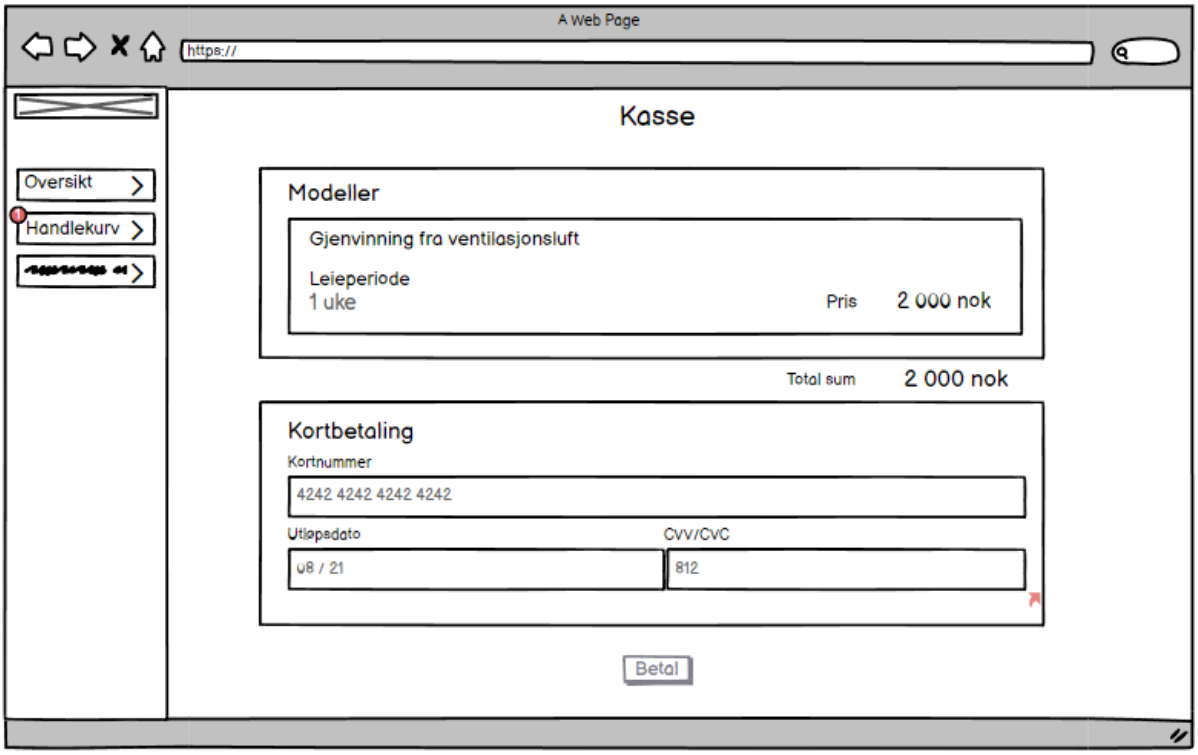


Figur 26 - Kjøpeside for en beregningsmodell med en leieperiode på 1 uke

Betalingsløsning

Ved leie av modeller og pakker i systemet vil brukerne ha mulighet til å betale med bankkort, som vil være vår «MVP»-løsning for betaling (Se figur 27). Et alternativ vil være faktura. Oppdragsgiveren vår har dessverre ikke mulighet til å gi oss tilgang til fakturasystemet de vanligvis bruker. Vår løsning blir dermed at kundene kan velge faktura ved betaling, som vil sende en forespørsel til systemadministratoren i Tempero. Tempero må deretter be regnskapsføreren sin lage en faktura for kjøpet som sendes til firmabrukeren.

Når det kommer til kvittering for kjøp, har Stripe en løsning som gjør at vi kan gi kunden kvittering med all viktig informasjon (*Provide receipts*, 2021). Kvittering for alle kjøp vil bli sendt på e-post til administratorbrukeren.



Figur 27 - Kortbetaling for kjøp av beregningsmodell

Veiledning

For at Tempero skal kunne gjennomføre veiledning med kundene sine, har vi gjennom data vi fikk fra spørreundersøkelsen, ønsker fra oppdragsgiver og brainstorming innad i gruppa kommet frem til at en god løsning kan være å ha et kontaktskjema inne i systemet, hvor kundene kan sende en melding til Tempero og be om veiledning. Tempero vil da automatisk få e-post som inneholder denne meldingen og kontaktinformasjonen til kunden som har sendt den. Deretter velger de selv hvordan de ønsker å gå frem med veiledningen.

Andre funksjonaliteter

Vi tenker at det er fordelaktig med en hjelpeside hvor kundene kan finne informasjon om hvordan systemet fungerer. Vi vil derfor ha en slik underside for innloggede brukere, som vil inneholde dokumentasjon for bruk av systemet.

Informasjonen som vises på hjelpesiden vil variere ut fra hvilken type bruker man har. Brukere vil kun se informasjon om bruk av funksjoner som de selv har tilgang til. Tempero sine systemadministratorer vil kunne se informasjon om alle sider, for å kunne veilede andre brukere i systemet.

«Nice-to-have»

Etter samtale med oppdragsgiver ønsker vi å implementere språkvalg i systemet. Det vil hovedsakelig være mulig å velge mellom «Norsk bokmål» eller «Engelsk». Dette vil gjøre systemet mer tilgjengelig. Dette er ikke inkludert i vårt «MVP» fordi flertallet av kundene til Tempero kommuniserer på norsk.

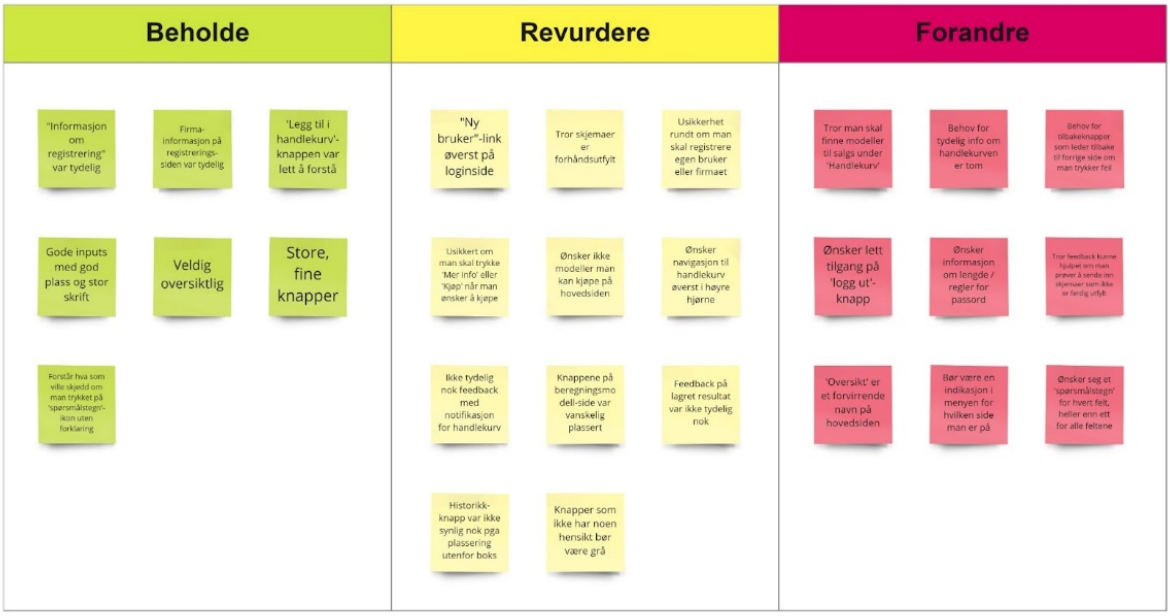
En annen «nice to have»-funksjon er muligheten for å ha to forskjellige fargetemaer i systemet, ett som er lyst og ett som er mørkt. Det lyse temaet vil være standard, etter ønske fra respondenter i spørreundersøkelsen, men de vil ha mulighet for å endre til et mørkt tema. Vi har ellers tenkt at systemvalg på enhetene deres skal kunne overstyre disse innstillingene.

Data fra brukertestingen

Brukertestene vi gjennomførte, ble fordelt mellom gruppemedlemmene og gjennomført hver for oss. For å samle resultatene hadde vi et møte hvor vi gikk gjennom alle svarene vi hadde fått. Vi sorterte dataene fra brukertestene inn i tre kategorier:

- *Beholde* - positive tilbakemeldinger på funksjonalitet vi bør ta med oss videre
- *Revurdere* - varierte tilbakemeldinger hvor vi bør revurdere løsningene våre
- *Forandre* - dårlige tilbakemeldinger hvor vi *må* finne nye løsninger

Vi kategoriserte alle tilbakemeldingene vi fikk inn i en tabell i Miro, for å få oversikt over hva vi bør se på i neste iterasjon av designet (Se figur 28).



Figur 28 - Oversikt over tilbakemeldinger i Miro

De fleste positive tilbakemeldingene vi fikk handlet om størrelsen på elementene og at sidene var veldig oversiktlige. Vi fikk også gode tilbakemeldinger på den forklarende informasjonen vi hadde skrevet om brukerregistreringen, fra de som valgte å spørre om dette under testen.

Vi hadde under researchfasen vår bestemt oss for å teste bruken av ikoner med spørsmålstegn som kunne trykkes på for å få mer informasjon i skjemaer. Vi hadde ikke satt opp funksjonaliteten for disse i prototypen, men hadde plassert ikoner på flere av skjemaene for å se hvordan testpersonene reagerte på det. Her fikk vi tilbakemeldinger som viste at testpersonene forstår hva som ville skjedd dersom man kunne trykke på ikonet, men at de ønsker at det skal være et ikon ved siden av hvert enkelt inndatafelt.

En ting som gjentok seg i flere av testene var at testpersonene ble veldig forvirret av plassholderne vi hadde skrevet inn i inndatafeltene (Se figur 29). Vi tenker at i skjemaene hvor man skal fylle ut personlig informasjon eller informasjon om eget firma, så kan det hende at testpersonene ville reagert annerledes dersom de hadde en klar idé om hvilken informasjon de skulle fylle ut. Da ville de muligens oppfattet at informasjonen som sto i feltene ikke var relatert til dem eller deres firma. Dette er en hypotese vi tenker at vi må teste ut i neste iterasjon av prototypen. Vi vurderer også å forandre på den visuelle fremstillingen av plassholderne for å gjøre det tydeligere at

skjemaet ikke er forhåndsutfylt, eller eventuelt å fjerne irrelevante plassholdere dersom de fremtidige brukertestene gir oss lignende resultater.

Figur 29 - Eksempel fra prototypen - skjema med plassholdere i inndatafelter

Vi spurte også et par av brukerne, som trodde at skjemaene var forhåndsutfylte, om det ville hjulpet dem dersom det kom opp feilmeldinger under hvert av inndatafeltene som forklarte at de måtte fylle inn informasjon i feltet før innsending. Dette var de veldig positive til, og ut fra researchen vi gjorde om design av brukergrensesnitt var det noe gruppa uansett hadde tenkt å inkludere senere i utviklingen. Vi ønsker allikevel at brukerne av systemet skal kunne forstå hvordan det fungerer uten å måtte få en feilmelding, så vi tror at en kombinasjon av disse løsningene vil fungere godt.

Vi fikk også andre tilbakemeldinger på ting vi ønsker å forandre på i neste iterasjon av prototypen. Alle tilbakemeldingene vi har fått viser oss hvor viktig det er å teste systemet på personer og forandre ting underveis i prosessen.

Koding av beregningsmodell

Tempero Energitjenester AS har en stor mengde beregningsmodeller. Disse brukes for å regne ut blant annet hvor kraftig en vifte trenger å være for å kjøle ned et

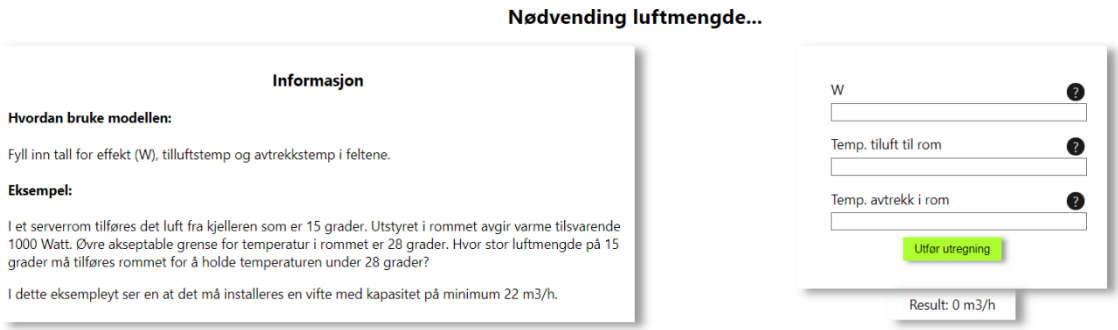
serverrom. Beregningsmodellene er høyst relevante for bedrifter som skal bygge nybygg og rehabiliterte bygg.

Dagens modeller eksisterer i Excel-ark, og inneholder formler for utregning, med felter hvor man fyller inn data gitt av bedrifter. Tempero utfører disse utregningene, for deretter å sende resultatene til sine kunder. Som nevnt, ønsker Tempero at disse modellene skal kunne brukes direkte av kundene deres, gjennom en nettside hvor kundene selv kan leie modeller og utføre beregninger.

Vi bestemte vi oss for å kode et par av modellene vi hadde fått tilsendt. Beregningsmodellene er hovedfunksjonen i systemet vi vil bygge, så vi ville forsikre oss om at vi ikke støtte på noen uforutsette problemer i kodingen og teste ut løsningene vi hadde kommet opp med under analysen av beregningsmodellene.

Vi brukte rammeverket React til å kode beregningsmodellene. All koden er plassert inn i én skjemakomponent som tar inn verdier fra brukeren og utfører beregninger ved hjelp av en funksjon. Beregningen som kommer fra funksjonen, vil da vises under dette skjemaet. Ved senere utvikling vil skjemaet deles opp i ulike komponenter som kan brukes i forskjellige typer beregningsmodeller.

For tilgjengelighet og brukervennlighet, har vi skjemaetiketter koblet sammen med hvert felt, samt et «spørsmålstegn»-ikon per felt som man kan holde musepeker over for å se relatert informasjon (Se figur 30). Hvert felt har en gitt minimums- og maksimumsverdi, satt av Tempero, for å skape trygghet hos brukerne, ettersom det kan redusere risikoen for feil utfylling (Se figur 2 i vedlegg).



Figur 30 - Utklipp av hvordan beregningsmodellen ser ut i nettleser

Upresise utregninger

Gjennom research fant vi et stort problem med måten vi originalt tenkte å kode beregningsmodellene. Javascript er ett av flere kodespråk som er dårlig på utregninger som inkluderer desimaler. Figur 31 viser et eksempel på dette.

```
> 16:08:10.359 0.1 + 0.2
<· 16:08:10.367 0.30000000000000004
```

Figur 31 - Eksempel på en enkel beregning gjort av JavaScript

Én løsning på dette problemet er å bruke noen av JavaScript sine innebygde metoder og funksjoner i utregningen, som «parseFloat()», «toPrecision()», og «toFixed()». En kombinasjon av disse metodene vil sørge for at resultatet blir presist, som vist i figur 32.

```
> 16:09:17.334 parseFloat((0.1 + 0.2).toPrecision(12))
<· 16:09:17.336 0.3
> 16:10:17.053 parseFloat((0.1 + 0.2).toFixed(2))
<· 16:10:17.060 0.3
```

Figur 32 - Eksempel på en enkel beregning gjort av JS, med funksjoner

Drøfting

Tidlig i prosjektet, etter å ha hørt oppdragsgivers ønske, kom vi opp med følgende problemstilling:

> «Hvordan kan vi designe en sikker og brukervennlig webløsning som tilgjengeliggjør utvalgte beregningsmodeller for Tempero Energitjenester sine kunder?»

Vi avgrenset denne problemstillingen og kom opp med noen forskningsspørsmål, som vi la frem i innledningen til denne rapporten, samt et mål for fordypningsemnet.

I denne delen av rapporten skal vi drøfte og analysere resultatene vi har fått, og sette de inn i teoretisk sammenheng, samt svare på forskningsspørsmålene våre. Vi vil også forklare hvorfor vi ikke fullførte alle målene vi satt for dette emnet.

Sikkerhetsfølelse ved betaling

Et av forskningsspørsmålene omhandlet hvordan ulike betalingsløsninger kan påvirke sikkerhetsfølelsen til brukeren. Det er logisk at en betalingsmåte brukeren har

hørt om eller brukt tidligere vil bli sett på som mer sikker, enn nye og ukjente betalingsløsninger. En undersøkelse som omhandler nettforbrukere viser at betalingsløsningen er den fjerde viktigste egenskapen en kunde ser etter når en skal kjøpe noe på nett (Willy, 2017, s. 29).

Gjennom vår spørreundersøkelse så vi at 100% av respondentene hadde hørt om kortbetaling, faktura og Vipps. 85,7% følte seg trygge på Vipps, og 100% følte seg trygge når de betalte med kort eller faktura.

På grunn av disse svarene og researchen vi gjorde om de ulike betalingsmetodene, ble den beste betalingsløsningen for bedriftens kunder å betale med bankkort, gjort mulig gjennom bruk av Stripe.

En negativ side ved kortbetaling, i dette tilfellet, er at ikke alle firma har et firmakort som kan brukes til slike kjøp. Alternativet er at man kan sette opp mulighet for betaling med faktura, men på grunn av at vi ikke kan bruke Tempero sitt nåværende fakturasystem foreslo de å sette opp alle fakturaene manuelt. Vi tenker at dette ikke er en optimal løsning, fordi bruk av faktura vil enten bli basert på tillit, altså at kunden får tilgang til beregningsmodellen før de har betalt, eller at kunden må vente på modellen til de har mottatt og betalt fakturaen.

Ettersom Tempero ønsker å godkjenne alle firmaer som skal bruke systemet, kan de selv velge om de har tillit til et firma. Når de skal godkjenne et nytt firma vil de ha tilgang på all firmainformasjon hvor det vil være krav om å skrive inn organisasjonsnummer, slik at Tempero kan sjekke om firmaene eksisterer ved å søke i Brønnøysundregisteret. Dette vil minske risikoen om de ønsker å ha et fakturasystem hvor kundene får tilgang til modellene før de har betalt.

Vi tenker uansett at denne løsningen vil være mindre effektiv på grunn av at den skaper en del manuelt arbeid for Tempero og deres regnskapsfører. Uavhengig av tillitsforhold kan det også oppstå tilfeller hvor firmakundene ikke betaler fakturaer, som igjen vil skape mer arbeid med purringer og inkasso.

Bruk av beregningsmodeller

Et annet forskningsspørsmål omhandlet hvordan vi best kan presentere beregningsmodellene til Tempero Energitjenester slik at kundene føler seg trygge på å bruke disse på egenhånd.

Slik som løsningen er i dag bruker Tempero Excel-ark til å utføre beregningene, og kundene bidrar kun med å dele data som er relevant for byggeprosjektene med Tempero. I våre spørreundersøkelser har vi spurt om kundene til Tempero faktisk ønsker en webløsning hvor de kan utføre disse beregningene selv. Det kom frem at alle respondentene ønsket dette.

Når kundene skal utføre beregningene selv, vil det være større muligheter for at de kan gjøre feil, ettersom dette er noe Tempero har håndtert for dem frem til nå. Vi har derfor gjort research på hvordan Excel-arkene kan kodes inn på en nettside. Vi fant ut at dersom man bruker innebygd funksjonalitet i kodespråk som HTML og JavaScript, kan man begrense mye av feilene som kan oppstå.

Vi har også hatt stort fokus på mengden informasjon som skal vises på nettsidene. Beregningsmodellene vi skal vise på siden er komplekse, og det kan i mange tilfeller være nyttig for brukerne å kunne se instruksjoner for hvordan man utfører beregningene. Gjennom vår research har vi funnet ut at store mengder informasjon kan gi brukere en større kognitiv belastning (Krug, 2014, s. 51) og at det derfor blir vanskeligere for brukere å finne frem til det de ønsker (Nielsen, 2020). Vi valgte derfor å teste ut dette i prototypen vår, hvor vi hadde større informasjonsbokser på enkelte sider hvor vi følte at brukeren kunne ha behov for instruksjoner. Vi testet samtidig hvordan brukere ville reagere på at informasjonen lå gjemt bak «spørsmålstegn»-ikoner som de selv må trykke på for å få frem informasjon.

I brukertestingen reagerte testpersonene veldig positivt på de instruksjonene de fikk lest opp, dersom de spurte om hva som skulle stå i informasjonsboksene som var fylt med «dummytekst». Her tenker vi at det kan være mulige feilkilder, fordi testpersonene ble under brukertesten fortalt hva som ville stå i informasjonsboksene og slapp derfor å lese gjennom alle instruksjonene selv. Dette kan bety at selv om instruksjonene var forståelige så vil ikke brukere nødvendigvis være fornøyde med å ha informasjonen synlig til enhver tid, spesielt når de må lese gjennom det på egen hånd. Hvis de bruker en beregningsmodell ofte, vil instruksjonene på siden bli kjent for dem, og det kan da føles som at instruksjonene tar opp unødvendig mye plass.

Et motargument kan være at modellene inneholder komplekse beregninger som man vanligvis ikke går rundt og husker. Det kan derfor være nyttig i mange tilfeller å ha informasjonen tilgjengelig når man trenger den.

Vi tenker at ved videre utvikling vil vi fortsette med å teste ulike hybridløsninger hvor informasjonen er gjemt for brukeren, men alltid lett tilgjengelig. Hybridløsningen vi testet, hvor informasjonen var gjemt bak «spørsmålstegn»-ikoner, ble godt mottatt. Vi har sett for oss at dette vil fungere best på steder hvor det er lite informasjon og brukeren lett kan huske det de har lest, ettersom informasjonen vil forsvinne når man seg trykker vekk fra ikonet.

En annen løsning vi har diskutert er å ha de viktigste instruksjonene i en boks med sammenleggbart innhold øverst på siden for hver beregningsmodell, slik at brukeren kan åpne og lukke boksen ettersom de har behov for å se mer detaljerte instruksjoner. Dette er noe vi tenker å teste ved eventuell videreutvikling av prosjektet.

Oppsummert mener vi at den beste måten å presentere beregningsmodellene for kunder, slik at de føler seg trygge på å bruke de på egen hånd, er via en webløsning. Ved å fremstille beregningsmodellene som kodede skjemaer kan man begrense en del feil som kan oppstå ved utfylling av komplekse data, og kundene vil ha mer frihet til å utføre beregningene når de selv ønsker.

I webløsningen tenker vi også at det er viktig at de har tilgang på informasjon ved behov. Vi har ikke kommet opp med en helt optimal løsning, men tenker at en hybridløsning hvor informasjonen er tilgjengelig, men ikke alltid synlig, vil være best for dette systemet. Det kan også være nyttig for brukerne å motta veiledning fra Tempero i den tiden de bruker systemet.

Gjennomføring av veiledning

Det siste forskningsspørsmålet vårt omhandlet derfor hvordan Tempero Energitjenester best kan gjennomføre veiledning med kundene sine i bruken av den nye webløsningen.

Under intervjuet med oppdragsgiveren vår kom det frem at de ønsket en måte å veilede kundene sine på gjennom webløsningen. De gjennomfører også i dag veiledning med alle kundene sine, og det foregår hovedsakelig ansikt til ansikt eller via e-post. I webløsningen ønsket de i utgangspunktet en form for chatløsning.

Dataene vi fikk fra spørreundersøkelsen vi gjennomførte viste at ingen av brukerne foretrakk chatløsning som et alternativ for veiledning i systemet. De fleste ønsket

ansikt til ansikt, og en stor del ønsket også veiledning via e-post. Dette ligner på systemet som eksisterer i dag.

I boka *The Design of Everyday Things* (2013, s. 32) nevner Don Norman at teknologi kan gjøre livet lettere for mennesker, men at det også kan skape mer kompleksitet og gi brukere mye frustrasjon. Vi tenker at dette kan være grunnen til at respondentene i spørreundersøkelsen ikke ønsker et nytt system for veiledning, ettersom de er fornøyde med dagens ansikt til ansikt-løsning og er godt kjent med bruk av sitt eget e-postsystem. Veiledningen som skal gjennomføres kan også inneholde veldig komplekse detaljer, og brukerne har fortalt oss at de synes at en muntlig løsning er mye mer effektiv. En chatfunksjon inne i webløsningen vil da være en ekstra, unødvendig funksjonalitet de blir nødt til å lære seg i et allerede nytt og ukjent system.

Etter en samtale med oppdragsgiver om hva kundene deres ønsket, forsto de at chatløsningen var unødvendig og ville skape mye ekstra arbeid for dem selv, ettersom de måtte bruke løsningen til å svare på mange av kundenes spørsmål. De ønsket likevel at vi skulle gi kundene mulighet til å kontakte dem gjennom systemet.

Ut fra alle disse opplysningene kom vi opp med et kompromiss om å inkludere et kontaktskjema inne i løsningen, hvor kundene kan skrive inn et spørsmål og Tempero vil kontakte dem for å avtale videre veiledning. På denne måten kan kundene fylle ut hva de ønsker veiledning om, og selv velge hvordan de vil gjennomføre den. Vi vil også ha kontaktinformasjonen til Tempero Energitjenester lett tilgjengelig på siden, slik at kundene kan kontakte dem via e-post eller telefon. Vi tenker at dette vil være den mest optimale løsningen vi kan bidra med i webløsningen, men at den beste måten å gjennomføre selve veiledningen på vil være opp til oppdragsgiveren og brukerne av systemet.

Lo-fi prototypen

Ettersom problemstillingen og forskningsspørsmålene våre omhandlet brukervennlighet, bestemte vi oss for å fokusere på brukergruppen. I spørreundersøkelsen vi gjennomførte fikk vi vite at 6 av 7 brukere hovedsakelig bruker datamaskin i jobbhverdagen. Grunnet dette, og tiden vi hadde tilgjengelig, laget vi kun en iterasjon av en low-fidelity prototype for datamaskin.

De siste årene har det blitt veldig populært å designe for «mobil først». En grunn til dette er at det skal være enklere å prioritere det viktigste innholdet på siden når man starter med en liten skjerm og skalerer oppover. Dette vil gi brukeren en bedre opplevelse (Morales, 2021).

Et alternativ til «mobil først» er å fokusere på hvor og hva brukeren skal bruke løsningen til, som er det første steget i «Reisedrevet design» (Mesibov og Levin, 2017). Med tanke på brukergruppen vår følte vi at dette var den beste metoden for oss å følge i designprosessen til dette systemet.

Vi har vurdert om dette kan påvirke plasseringen av innholdet på siden negativt. Ettersom det ikke vil være store mengder ulikt innhold på hver side, og vi ønsker å holde designet mest mulig minimalistisk, har vi kommet frem til at dette ikke bør bli et problem. Derfor bestemte vi oss for å ikke designe for mobil, men holde oss til å designe prototypen for datamaskin i denne delen av prosessen.

Siden flertallet i brukergruppen vil bruke løsningen på datamaskin ble dette også det mest naturlige valget, ettersom vi skulle bruke prototypen til å teste funksjonaliteten på brukere.

Måloppnåelse

I starten av dette prosjektet satte vi oss noen mål som vi ønsket å gjennomføre. I tabell 2 kan du se en oversikt over målene våre, og informasjon om hvilke vi fullførte.

Mål for fordypningsemnet	Fullført
Gjøre godt med research som skal gi oss et godt grunnlag for bachelor	Ja
Finne passende teknologier for systemet	Delvis
Utføre spørreundersøkelser for å få god innsikt i brukergruppen	Ja
Lo-fi prototype og brukertesting av prototype	Ja
Hi-fi prototype og brukertesting av prototype	Nei
Kode minst en beregningsmodell	Ja

Tabell 2 - Oversikt over måloppnåelse

Målet om *passende teknologiske løsninger* ble delvis fullført fordi vi i starten tenkte at vi skulle finne passende teknologier for både front-end og back-end. Etter vi kom litt lenger inn i oppgaven så oppdaget vi fort at vi ikke ville ha tid til å kode noe relatert til

back-end i dette semesteret. Derfor fokuserte vi i stedet på front-end teknologier og teknologier for integrering av betalingsløsninger.

Målet om å lage *hi-fi prototype,* valgte vi å ikke fullføre. Etter at vi hadde laget og brukertestet lo-fi prototypen, så var det få uker igjen til innlevering av prosjektet. På grunn av begrenset tid og en stor rapport som skulle skrives, fikk vi ikke mulighet til å lage en ny iterasjon av lo-fi prototypen til testing. Vi føler det var behov for å se resultatene fra minst én iterasjon til, før vi kunne starte på designet av en eventuell hi-fi prototype.

Konklusjon

I løpet av denne rapporten har vi lagt frem prosessen vår med å utvikle en webløsning for Tempero Energitjenester AS. Webløsningen skulle gjøre beregningsmodellene som Tempero bruker i dag tilgjengelige, slik at kundene deres kan bruke disse modellene på egenhånd.

I denne delen av rapporten har vi diskutert og reflektert over valg vi har tatt. Vi har gjennom denne refleksjonen svart på forskningsspørsmålene våre og forklart hvorfor vi ikke fullførte alle målene vi hadde satt.

Forskningsspørsmålene vi stilte i starten av prosjektet er saker vi tenker er viktige for vår oppdragsgiver og brukerne av systemet. Vi har gjennom research og ulike metoder hatt et stort fokus på at systemet vi utvikler skal bli mest mulig brukervennlig, og å forhindre feil som kan skje når man håndterer komplekse data hvor nøyaktigheten til resultatene har mye å si for kundenes prosjekter.

Vi har oppdaget at det å presentere beregningsmodellene på en nettside er både ønsket av kundene til Tempero og kan forhindre feil gjennom riktig bruk av ulike kodespråk. Å finne en god balanse i mengden instruksjoner til hver beregningsmodell vil også gjøre siden mer brukervennlig, og kan hjelpe brukerne av systemet med å føle seg trygge på å bruke modellene på egenhånd.

De vil også ha mulighet til å få veiledning fra Tempero i bruken av systemet, dersom de skulle ønske dette. Her kom vi frem til at den beste løsningen er å beholde løsningen de bruker i dag. Kundene ønsker ikke å veiledes inne i webløsningen, og etter at vi la frem resultatene fra vår research ble Tempero enige i at det allikevel ikke var et stort behov for en chatløsning, eller lignende, inne i systemet. Vi tenker

allikevel at det kan være fint for noen kunder å ha mulighet til å kontakte Tempero via webløsningen og at kontaktinformasjonen deres skal være lett synlig på siden.

Vi undersøkte også hvordan ulike betalingsløsninger kunne påvirke sikkerhetsfølelsen til brukerne av systemet. Vi kom frem til at standardløsninger som bruk av bankkort og faktura, noe kundene allerede er godt kjent med, vil være best for at de skal føle seg trygge når de skal leie beregningsmodeller til sine firmaer.

Dersom man fortsetter å utvikle systemet, med alle disse idéene i bakhodet, tenker vi at det kan resultere i en webløsning som er både sikker og brukervennlig. Det er fortsatt en del funksjonalitet og annet som bør testes ved videre utvikling. Det vil være nødvendig å gå gjennom flere iterasjoner av både lo-fi og hi-fi prototyper, og teste disse på personer innenfor brukergruppen for å forsikre seg om at sikkerheten og brukervennligheten opprettholdes.

Kilder

Andersen, E.S., Schwencke, E. (2020) *Prosjektarbeid en veiledning for studenter*, 5 utg. Fagbokforlaget.

brainstorm (2021) *Store norske leksikon*. Tilgjengelig fra: http://snl.no/brainstorm (Hentet 24. november 2021)

Datatilsynet (2019) *Personopplysninger*. Tilgjengelig fra: https://www.datatilsynet.no/rettigheter-og-plikter/personopplysninger/ (Hentet: 27.november.2021).

Forskrift om universell utforming av IKT-løsninger (2019) *Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)-løsninger.* Tilgjengelig fra: https://lovdata.no/dokument/SF/forskrift/2013-06-21-732 (Hentet: 27.november.2021).

Gibbons, S. (2018). *Journey Mapping 101*. Tilgjengelig fra: https://www.nngroup.com/articles/journey-mapping-101/ (Hentet: 25.november.2021).

Hund, P. (2016). *SEO vs. React: Web Crawlers are Smarter Than You Think*. Tilgjengelig fra: https://www.freecodecamp.org/news/seo-vs-react-is-it-neccessary-to-render-react-pages-in-the-backend-74ce5015c0c9/ (Hentet 25.november.2021).

Interaction Design Foundation (2021). *What is Flat Design?* Tilgjengelig fra: https://www.interaction-design.org/literature/topics/flat-design (Hentet: 23.november.2021).

Krug, S. (2014). D*on't Make Me Think, Revisited. A common sense approach to web usability*. New Riders.

Lee, S., Koubek, R.J. (2010). T*he effects of usability and web design attributes on user preference for e-commerce web sites*. Computers in Industry 61, 329–341. Tilgjengelig fra: https://doi.org/10.1016/j.compind.2009.12.004 (Hentet: 20. september 2021).

Lopuck, L. (2012). *Web Design: How to Create a Sitemap*. Tilgjengelig fra: https://www.dummies.com/web-design-development/site-development/web-design-how-to-create-a-sitemap/ (Hentet: 17.november 2021).

Mesibov, M., Levin, J. (2017). *Mobile First Is Just Not Good Enough: Meet Journey-Driven Design.* Tilgjengelig fra: https://www.smashingmagazine.com/2017/02/mobile-first-is-just-not-good-enough-meet-journey-driven-design/ (Hentet 25. november.2021).

Morales, J. (2021). *Mobile First Design Strategy: The When, Why and How*. Tilgjengelig fra: https://xd.adobe.com/ideas/process/ui-design/what-is-mobile-first-design/ (Hentet 11. november.2021).

Mozilla (2021). *<input type="number">*. Tilgjengelig fra: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/number (Hentet 23. november.2021).

Nielsen, J. (2020).*10 Usability Heuristics for User Interface Design*. Tilgjengelig fra: https://www.nngroup.com/articles/ten-usability-heuristics/ (Hentet 4. oktober.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 25. november.2021).

Norman, D. (2013). T*he Design of everyday things*, 2 utg. Basic books.

Petersen, J.V. (2021). *Learn to Perform Math Operations in JavaScript*. Tilgjengelig fra: https://www.codemag.com/article/1811041/JavaScript-Corner-Math-and-the-Pitfalls-of-Floating-Point-Numbers (Hentet 23. november.2021).

Provide receipts (2021). Tilgjengelig fra: https://stripe.com/docs/terminal/features/receipts (Hentet 25. november.2021).

Rolstadås, A. (2021). Gantt-diagram. *Store norske leksikon*. Tilgjengelig fra: http://snl.no/Gantt-diagram (Hentet: 15. oktober.2021).

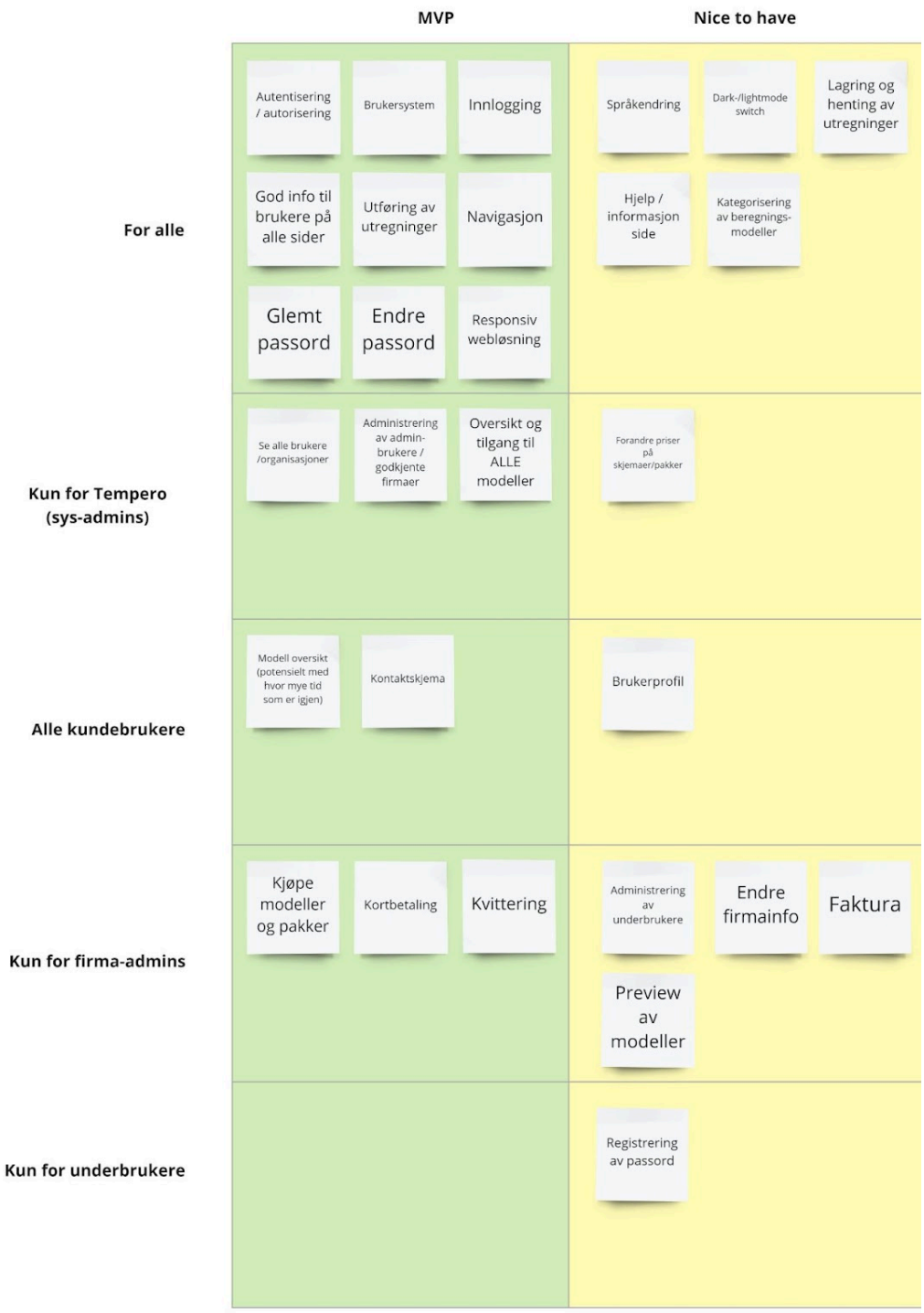
Schade, A. (2015). *Pilot Testing: Getting It Right (Before) the First Time*. Tilgjengelig fra: https://www.nngroup.com/articles/pilot-testing/ (Hentet 23. november.2021).

Stack Overflow (2021). *Stack Overflow Developer Survey 2021*. Tilgjengelig fra: https://insights.stackoverflow.com/survey/2021/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2021 (Hentet 23. november.2021).

W3C (2011). *Retningslinjer for tilgjengelig webinnhold (WCAG) 2.0*. Tilgjengelig fra: https://www.w3.org/Translations/WCAG20-no/ (Hentet 20. oktober.2021).

Willy, K. (2017). *The truth about online consumers*. (2017 Global Online Consumer Report). KPMG. Tilgjengelig fra: https://assets.kpmg/content/dam/kpmg/xx/pdf/2017/01/the-truth-about-online-consumers.pdf (Hentet 26. november. 2021).

Vedlegg



Figur 1 - Oversikt over «MVP» vs «nice-to-have»

	A	B	C	D	E	F
1	**Modell**	**Input/felt**	**Min verdi**	**Max verdi**	**Enhet**	
2	**Temp. og effekt nødvendig for å fjerne varme fra rom**					
3		Watt (mengde energi) tilført romme	1	10 000 000	Watt	1000 Watt = 1 kW
4		m3/h tilluft	1	500 000	m3/h	Kubikkmeter luft pr.time
5		Temp. tilluft til rom	1	100	grader C	
6		Temp. avtrekk I rom	1	100	grader C	
7	**Gjenvinnning fra ventilasjonsluft**					
8		Energipris kr/kwh	0	100	kr	Pris for 1 kWh
9		Luftmengde m3/h	1	60 000	m3/h	Kubikkmeter luft pr.time
10		Temp. avtrekksluft	1	100	grader C	
11		Temp. uteluft	1	100	grader C	
12		Gjennvinningsgrad i %	0	100	prosent	
13		Lengde begrensningsperiode i time	1	8784	timer	8760 timer er et år. Men kan også bruke andre timetall
14	**Lufthastighet i ventilasjonskanaler**					
15		V (m3/h)	1	60000	m3/h	Kubikkmeter luft pr.time
16		d (m2)	0,1	10	meter	Diameter på en sirkulær ventilasjonskanal målt i meter
17	**Energiforbruk ventilasjon og SFP**					
18		SFP planlagt	1	10	P[kW]/V[m3/h]	Effekt som viften i ventilasjonsanlegget trenger dividert på luftmengdwe
19		Luftmengde m3/h	1	50 000	m3/h	Luftmengde som skal gjennom ventilasjonskanaler pr. time
20		SFP i aktuelle prosjekt	1	10	P[kW]/V[m3/h]	
21		Timer i aktuelle tidsperiode	1	8 784	timer	8760 timer er et år. Men kan også bruke andre timetall
22						

Figur 2 – Oversikt over minimum og maksimum verdier for ulike inndatafelt