



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Miside bedrift – Selvbetjeningsportal for bedriftskunder
hos Eninvest

Miside bedrift – Self-service Portal for Corporate Customers at Eninvest

Anders Fimreite

Jørgen Fjølstad

Truls Agnar Haukeland

Informasjonsteknologi

Fakultet for teknologi, miljø- og samfunnsvitenskap

Institutt for datateknologi, elektroteknologi og realfag

Veileder: Per Christian Engdal

Innleveringsdato: 13/05/2024

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 10

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Miside bedrift – Selvbetjeningsportal for bedriftskunder hos Eninvest	<i>Dato:</i> 13.05.24
<i>Forfatter(e):</i> Anders Fimreite, Jørgen Fjølstad og Truls Agnar Haukeland	<i>Antall sider u/vedlegg:</i> 47
	<i>Antall sider vedlegg:</i> 120
<i>Studieretning:</i> Informasjonsteknologi	<i>Antall disketter/CD-er:</i> Ingen
<i>Kontaktperson ved studieretning:</i> Per Christian Engdal	<i>Gradering:</i> Ingen
<i>Merknader:</i> Ingen	

<i>Oppdragsgiver:</i> Eninvest AS	<i>Oppdragsgivers referanse:</i> Ingen
<i>Oppdragsgivers kontaktperson:</i> Jan Roger Myklebust	<i>Telefon:</i> 90603339

<i>Sammendrag:</i> <p>Prosjektet handler om å utvikle en selvbetjeningsløsning for bedriftskundene hos Eninvest for å effektivisere kundebehandlingen og øke tilfredsheten hos kundene. Løsningen er Miside bedrift hvor bedriftskundene kan håndtere oppgavene på egenhånd, framfor dagens løsning der kundene må ringe eller sende e-post til kundeservice.</p> <p>The project is about developing a self-service solution for corporate customers at Eninvest to streamline customer service and increase customer satisfaction. The solution is called Miside bedrift, where corporate customers can handle tasks on their own, instead of the current solution where customers must call or send emails to customer service.</p>
--

Stikkord:

Selvbetjeningsportal	Webutvikling	Role-Based Access Control (RBAC)
----------------------	--------------	----------------------------------

Høgskulen på Vestlandet, Fakultet for teknologi, miljø- og samfunnsvitenskap

Postadresse: Postboks 7030, 5020 BERGEN

Besøksadresse: Inndalsveien 28, Bergen

Tlf. 55 58 75 00 Fax 55 58 77 90

 E-post: post@hvl.no

 Hjemmeside: <http://www.hvl.no>

FORORD

Denne rapporten er utarbeidet som en del av bacheloroppgaven tildelt av Eninvest, og omhandler utviklingen av Miside bedrift – selvbetjeningsportal for bedriftskunder hos Eninvest. Prosjektet ble gjennomført våren 2024 av Anders Fimreite, Jørgen Fjølstad, og Truls Agnar Haukeland ved Høgskulen på Vestlandet.

Takk til Eninvest for at de har stilt med bacheloroppgave, arbeidslokaler og utstyr for bachelorprosjektet. Takk til Jan Roger Myklebust, Kenneth Kvamme og andre for hjelp og veiledning gjennom prosjektet. Takk også til de ansatte i Eninvest som deltok under testing av prosjektet.

Til slutt vil gruppen takke Per Christian Engdal som har vært prosjektgruppens interne veileder ved Høgskulen på Vestlandet gjennom hele prosjektet, og har kommet med gode råd og tilbakemeldinger.



INNHALDSFORTEGNELSE

1	INNLEDNING	1
1.1	KONTEKST	1
1.2	MOTIVASJON	1
1.3	PROSJEKTEIER	1
1.4	PROBLEMBESKRIVELSE OG MÅL	2
1.5	OPPBYGGING AV RAPPORTEN	2
2	PROSJEKBESKRIVELSE	3
2.1	PRAKTISK BAKGRUNN	3
2.1.1	<i>Tidligere arbeid</i>	3
2.1.2	<i>Initielle krav</i>	4
2.1.3	<i>Initiell løsnings-idé</i>	5
2.2	AVGRENSNINGER	6
2.3	RESSURSER	6
2.4	LITTERATUR OM PROBLEMSTILLINGEN	6
3	DESIGN AV PROSJEKTET	8
3.1	FORSLAG TIL LØSNING	8
3.1.1	<i>Alternativ løsning 1 – Tilgangsnivå</i>	8
3.1.2	<i>Alternativ løsning 2 – Role-Based Access Control</i>	9
3.1.3	<i>Diskusjon av alternativene</i>	9
3.2	VALGT LØSNING	10
3.3	VALG AV VERKTØY	10
3.4	PROSJEKTMETODIKK	14
3.4.1	<i>Utviklingsmetodikk</i>	14
3.4.2	<i>Prosjektplan</i>	15
3.4.3	<i>Risikovurdering</i>	17
3.5	EVALUERINGSPLAN	18
4	DETALJERT LØSNING	19
4.1	ARKITEKTUROVERSIKT	19
4.2	SERVICELAG	20

4.2.1	<i>EVIS</i>	20
4.2.2	<i>Oppbygningen av en komponent</i>	21
4.2.3	<i>Access-komponenten</i>	21
4.2.4	<i>SelfserviceBM-komponenten</i>	22
4.2.5	<i>Sekvensdiagram</i>	23
4.3	APPLIKASJON	25
4.3.1	<i>Teknologi og struktur</i>	25
4.3.2	<i>Skisser</i>	25
4.3.2.1	<i>Tilgangsstyring</i>	26
4.3.2.2	<i>Faktura</i>	26
4.3.2.3	<i>Saker</i>	27
4.3.2.4	<i>Linjer</i>	27
4.3.2.5	<i>Kontaktperson</i>	28
4.3.2.6	<i>Hjemmeside og navigasjonsbar</i>	28
4.3.3	<i>Flytdiagram</i>	29
5	RESULTATER	30
5.1	EVALUERINGSMETODE	30
5.1.1	<i>Verifisering</i>	30
5.1.2	<i>Validering</i>	30
5.1.3	<i>Brukertesting</i>	31
5.2	EVALUERINGSRESULTAT	32
5.2.1	<i>Verifisering</i>	32
5.2.2	<i>Validering</i>	32
5.2.3	<i>Brukertesting</i>	33
5.3	PROSJEKTRESULTAT	35
5.3.1	<i>Tilgangsstyring</i>	36
5.3.2	<i>Faktura</i>	38
5.3.3	<i>Saker</i>	39
5.3.4	<i>Linjer</i>	40
5.3.5	<i>Kontaktperson</i>	41

5.3.6	<i>Hjemmeside og navigasjonsbar</i>	41
5.3.7	<i>Profil</i>	41
5.4	PROSJEKTGJENNOMFØRING.....	42
6	DISKUSJON	43
6.1	PROSJEKTGJENNOMFØRING.....	43
6.2	RESULTAT.....	43
6.3	BRUK AV TEKNOLOGI	45
7	KONKLUSJON OG VIDERE ARBEID	46
7.1	KONKLUSJON.....	46
7.2	VIDERE ARBEID.....	47
7.2.1	<i>Design</i>	47
7.2.2	<i>Lansering til kunder</i>	47
7.2.3	<i>Funksjoner</i>	47
8	REFERANSER	48
9	VEDLEGG	50

INNHOLDSFORTEGNELSE FIGURER

Figur 2-1: Prototype for tilgangsstyring i webapplikasjonen.....	5
Figur 3-1: Protection rings (Chin og Older, 2010, s. 253)	8
Figur 3-2: Flat RBAC (Sandhu, et al, 2000, s. 50).....	9
Figur 3-3: Hierarchical RBAC (Sandhu, et al, 2000, s. 52)	9
Figur 3-4: Gantt - beskrivelse av milepæler	15
Figur 3-5: Gantt-diagram for organisering og administrering av prosjektet.....	16
Figur 4-1: Overordnet arkitekturskisse	19
Figur 4-2: Serverarkitekturskisse EVIS: komponenter involvert i løsningen	20
Figur 4-3: Arkitekturskisse komponent.....	21
Figur 4-4: Illustrasjon av rollehierarki	22
Figur 4-5: Sekvensdiagram for spørring getTickets.....	24
Figur 4-6: Forenklet komponent-struktur i applikasjonen	25
Figur 4-7: Skisse faktura – oversikt over faktura sendt til en kunde.....	26
Figur 4-8: Skisse saker - meldingsboks for en valgt sak.....	27
Figur 4-9: Skisse linjer - en valgt linje med trafikkgraf.....	28
Figur 4-10: Flytdiagram - hente ut informasjon om fakturaer	29
Figur 4-11: Flytdiagram - opprette ny sak	29
Figur 4-12: Flytdiagram – endre brukers rolle	30
Figur 5-1: Resultat enhetstesting.....	32
Figur 5-2: Resultat spørreundersøkelse - viser brukerens vurdering av hvor enkelt det er å forstå og bruke tilgangsstyring siden.....	33
Figur 5-3: Resultat spørreundersøkelse - viser brukerens vurdering av verdien til funksjonaliteten som tilbys på siden for tilgangsstyring for din egen arbeidshverdag	34
Figur 5-4: Resultat spørreundersøkelse - viser brukerens vurdering av verdien til funksjonaliteten på siden for tilgangsstyring for Eninvest sine bedriftskunder og deres arbeidshverdag	34
Figur 5-5: permissionconfig.class.php	36
Figur 5-6: permissionchecker.class.php	37
Figur 5-7: Roller laget for løsningen vist hierarkisk	37
Figur 5-8: Side for faktura som ikke er betalt	38
Figur 5-9: Side for en valgt sak.....	39
Figur 5-10: Side for en valgt linje	40
Figur 5-11: Progresjon av timer jobbet i bachelor	42

INNHALDSFORTEGNELSE TABELLER

Tabell 2-1: Funksjoner utarbeidet ved workshop hos oppdragsgiver	4
Tabell 2-2: Kategorisert funksjonalitet som blir prioritert å løse i dette prosjektet	5
Tabell 3-1: Risikoanalyse versjon 1 – 23.02.24: En systematisk fremstilling av risiko, deres sannsynlighet og konsekvens, og tiltak for å håndtere disse i prosjektet.	18
Tabell 5-1: Kategorisert funksjonalitet som ble utviklet for dette prosjektet	35

Ordliste

Backendutvikling	Utvikling på serversiden av en applikasjon.
Controller	Klasse som setter opp endepunkter for applikasjonen.
Endepunkt	En URL der en webtjeneste kan kobles til en klient.
Frontendutvikling	Utvikling på klientsiden av en applikasjon.
Handler	Klasse som håndterer og validerer brukerinput.
Linjestatus	Status eller tilstanden til en linje slik som Eninvest sine kommunikasjonslinjer.
Model	Klasser som håndterer forretningslogikk.
Parprogrammering	En teknikk der to utviklere jobber sammen på en arbeidsstasjon.
Repository	En klasse med ansvar for databasefunksjoner.
Techlead	En person med ledene rolle for et utviklingsprosjekt.
Trafikkgraf	En grafisk fremstilling over data- eller nettverkstrafikk som viser mengde og/eller hastighet over tid.
Tre	En datastruktur som simulerer hierarkiet av en trestruktur med en rotverdi og underliggende noder.
Webapplikasjon	Et program som kjører i en nettleser.

Akronym

API	Application programming interface
BM	Bedriftsmarkedskunder
EVIS	Eninvest Informasjonssystem
PM	Personmarkedskunder
RBAC	Role-based access control

1 INNLEDNING

1.1 Kontekst

Eninvest sine privatkunder har per i dag en egen selvbetjenings portal som vil gi disse kundene muligheten for å gjøre enkle handlinger på egen hånd. Bedriftsmarkedskunder (BM-kunder) må derimot ringe eller sende e-post til Eninvest for å få hjelp til det aller meste. Dette kan være enkle ting slik som faktura, hvor kundebehandlerne på kundeservice kan hjelpe. Er det derimot saker som er litt mer avansert må BM-kundene bli henvist videre til bedriftsrådgiverne eller annen teknisk avdeling som kan assistere kunden. Denne løsningen skaper et unødvendig høyt trykk for ansatte i Eninvest, og kan føre til lengre ventetid for kundene.

1.2 Motivasjon

Eninvest sine ansatte opplever at mange henvendelser som kommer fra bedriftskundene til enten kundeservice eller rådgiverne, kan utføres av kunden selv hvis dette er en mulighet. Ved å gi kunden denne muligheten vil trykket på saksbehandlere for slike saker minke, og kan fokusere på saker som er mer krevende. Muligheten til selvbetjening har også blitt etterspurt av enkelte BM-kunder.

1.3 Prosjekteier

Eier av dette prosjektet er Eninvest AS, som leverer bredbånd til over 23000 privatkunder og 2000 bedriftslinjer, samt telefoni- og WiFi-løsninger til både privat- og bedriftskundene. Eninvest har rundt 60 ansatte som jobber på hovedkontoret i Førde, eller på en av avdelingene på Sandane eller i Florø (Eninvest (u.å.)). Eninvest ble etablert 10. oktober 2000, hvor de satte seg som mål å gi Sogn og Fjordane en ny og fremtidsrettet infrastruktur. De har også som mål å være det beste alternativet for innbyggerne og bedriftene i Sogn og Fjordane (Eninvest AS (u.å.)). I tillegg til å levere internett til privat- og bedriftskunder har selskapet en utviklingsavdeling som utvikler og drifter eksterne og interne løsninger. En av disse løsningene er selbetjeningsløsningen Miside privat, som er utviklet for å gi privatkundene muligheten til å utføre handlinger knyttet til sitt kundeforhold. Disse handlingene er blant annet å hente faktura, se tidligere og melde inn nye saker, melde flytting og med mer.

1.4 Problembeskrivelse og mål

Dagens løsning fører til økt trykk for Eninvest sine kundebehandlere og bedriftsrådgivere. Av denne grunnen er det ønskelig med en selvbetjenings løsning for BM-kundene slik som privatkundene har i dag. En viktig faktor er at denne selvbetjeningsløsningen må være brukervennlig for BM-kundene, slik at kundene kan løse enkelte oppgaver på egen hånd. Med bakgrunn i dette stilles følgende forskningsspørsmål: Hvordan kan en selvbetjeningsløsning utvikles for Eninvest sine bedriftskunder, som effektiviserer kundebehandlingen hos Eninvest og øker kundetilfredsheten hos kundene?

1.5 Oppbygging av rapporten

Kapittel 1: Vil gi en bakgrunn for dette prosjektet.

Kapittel 2: Vil beskrive prosjektet sin praktiske bakgrunn, slik som arbeid som er utført rundt dette prosjektet tidligere, kravene satt av oppdragsgiver, foreslåtte løsninger og resursene som vil være tilgjengelig for prosjektperioden.

Kapittel 3: Vil det bli presentert og diskuterte forskjellige løsninger, med en endelig valgt løsning, verktøyene som er valgt for løsningen, metodikken brukt for dette prosjektet, samt hvordan evalueringen er planlagt.

Kapittel 4: Vil gi en detaljert presentasjon av løsningen som skal brukes for å oppnå prosjektresultatet.

Kapittel 5: Vil resultatet av løsningen som er produsert, med evaluering av resultatet presenteres.

Kapittel 6: Vil det drøftes årsaken til at resultatet ble som det ble, spesielt avvik fra planlagt resultat.

Kapittel 7: Vil gi en konklusjon for dette prosjektet og presentere videre arbeid som kan utføres av oppdragsgiver.

2 PROSJEKTBESKRIVELSE

2.1 Praktisk bakgrunn

2.1.1 Tidligere arbeid

Eninvest utvikler sine egne datasystemer internt og har i dag systemet Eninvest Informasjonssystem (EVIS) som håndterer det meste av selskapets databehandlingsbehov. EVIS fungerer som et felles API for alle Eninvest sine applikasjoner, også for det som utvikles i dette prosjektet. Det finnes allerede Miside for privatkunder, men for bedriftskunder mangler dette. Mye av funksjonaliteten som finnes for privatmarkedskunder (PM-kunder) kan tilpasses og gjenbrukes for BM-kundene, deriblant innlogging, fakturautlisting og sakslogger. Som en forberedelse for dette prosjektet utføres en behovsanalyse av Eninvest. Dette gjøres som en workshop hvor ansatte fra ulike avdelinger kom med ideer til ulike funksjoner til selvbetjeningsløsningen, disse funksjonene er vist i Tabell 2-1.

Kategori	Funksjonalitet
Tilgangsstyring	Innlogging på vegne av
	Lese- og skriverettigheter
Faktura	Laste ned fakturakopi
	Oversikt over og endre fakturering
Saker	Sakslogg
	Opprett ny sak
Linje/samband	Linjestatus
	Trafikkgrafer
	Info om feil og planlagt arbeid
	Oversikt over leverte samband
	Logg (nedetid og brudd)
	SLA-målinger
	Oversikt over NNI-punkt (Network-to-network interface)
Sette egne referanser	
Kontaktperson	Endre kontakt- og varslingsperson
Generelt	Se bedrifts rådgiver

Tilbud/bestilling	Be om tilbud på adresse
	Endre abonnement
	Leveransestatus
	Bestille tilleggsprodukter
Ruter	Se og endre diverse innstillinger
Diverse	Domene admin
	Enkelt internt webhotell

Tabell 2-1: Funksjoner utarbeidet ved workshop hos oppdragsgiver

2.1.2 Initielle krav

De initielle kravene er satt av både Eninvest og enkelte av BM-kundene til Eninvest, som etterspurte en slik selvbetjeningsløsning. Disse kravene er fremstilt i Tabell 2-2 som viser funksjonalitet som blir prioritert i dette prosjektet. En viktig faktor som må vektlegges i forhold til utformingen er at appen skal benytte samme design og grafisk stil som resten av Eninvest sine systemer. Prosjektet skal levere en fullt funksjonell prototype, implementert med Eninvest sin eksisterende arkitektur og integrert med gjeldende versjon av EVIS. En mer utdypende oversikt over de funksjonelle egenskapene til applikasjonen finnes i Vedlegg I: Visjonsdokument.

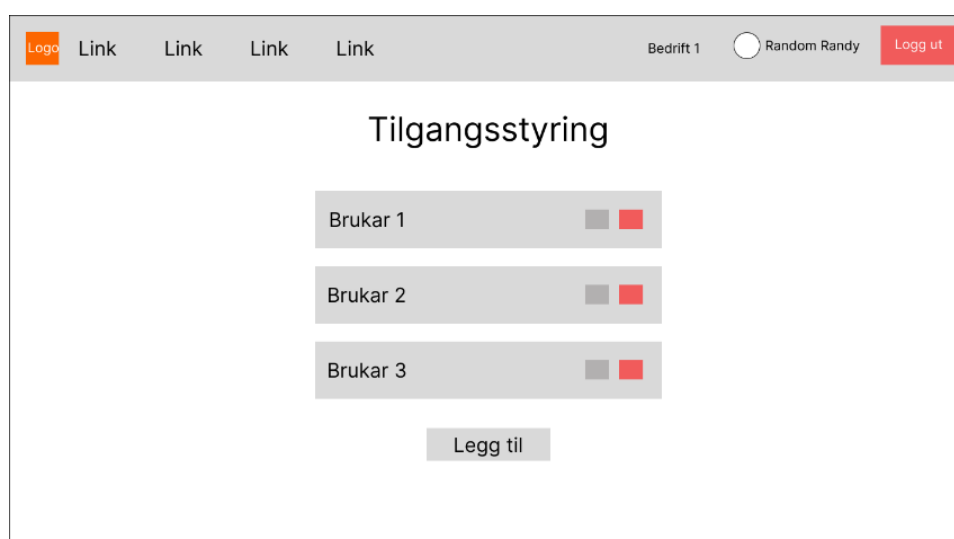
Kategori	Funksjonalitet	Kommentar
Tilgangsstyring	Innlogging på vegne av	Brukere med tilgang kan logge inn på vegne av en bedrift
	Lese- og skriverettigheter	En administrator kan kontrollere hvem som har tilgang til løsningen, samt spesifikke funksjoner i løsningen
Faktura	Laste ned fakturakopi	En bruker kan laste ned faktura eller kreditnota
	Oversikt over og endre fakturering	Gir en oversikt over fakturering og muligheten til å endre fakturering
Saker	Sakslogg	Se eldre/tidligere saker hos Eninvest
	Opprett ny sak	Opprette en ny sak, med innenfor nytt tema eller problemområde
Generelt	Se bedriftens rådgiver	Hente kontaktinformasjon til rådgiver

Kontaktperson	Endre kontakt- og varslingsperson	Administrering av kontakt og varslingspersoner som skal kontaktes i ulike tilfeller
Linje	Linjestatus	Se status på de ulike linjene bedriften har hos Enivest
	Trafikkgrafer	Se trafikken på en gitt linje
	Informasjon om feil og planlagt arbeid	Informasjon om problemer eller planlagt nedetid med en eller flere linjer
	Oversikt over leverte samband	En fullstendig oversikt over leverte samband

Tabell 2-2: Kategorisert funksjonalitet som blir prioritert å løse i dette prosjektet

2.1.3 Initiell løsnings-idé

Initiell løsnings-idé er å gi BM-kunder hos Enivest en webapplikasjon der brukere kan logge inn på vegne av bedriften og utføre enkelte handlinger knyttet opp mot tjenestene som bedriften har gjennom Enivest. I applikasjonen skal brukere og tilgang kunne styres, slik at bedriften kan gi ut og fjerne tilgang for brukere ved behov. Dette krever at enkelte brukere i bedriften har forhøyet tilgang og mulighet til å gjøre endringer rundt dette. Figur 2-1 viser en enkel skisse av hvordan brukergrensesnittet for tilgangsstyring er tenkt å se ut. Løsningen er designet med en navigasjonslinje på toppen av siden som vil inneholde lenker til de ulike sidene. Den inneholder informasjon slik som valgt bedrift og navnet til brukeren, samt oppdragsgivers logo og en knapp for utlogging.



Figur 2-1: Prototype for tilgangsstyring i webapplikasjonen

2.2 Avgrensninger

For prosjektet er det gjort avgrensninger, dette inkluderer at design av applikasjonen skal stå i stil med andre Eninvest-applikasjoner. Utviklingen av applikasjonen skal gjøres ved bruk av PHP på backend og React på frontend, hvor kode og databasetabeller skal følge eksisterende strukturer. Tilgjengelig tid setter en grense for prosjektets omfang og hva som kan realiseres, som medfører at prosjektet sin leveranse er begrenset til en funksjonell prototype.

Funksjonaliteten til applikasjonen, vist i Tabell 2-1, er avgrenset fra en lengre liste av funksjonaliteter. Fra dette er sluttresultatet ansett som kjernefunksjonalitet som vil være nødvendig å implementere som første versjon av applikasjonen.

2.3 Ressurser

For dette prosjektet gir oppdragsgiver tilgang til EVIS, som vil gi tilgang til å se systemene som allerede eksisterer. Enkelte deler av funksjonaliteten i EVIS kan brukes direkte i prosjektet eller kan tilpasses. Videre har ansatte ved Eninvest stilt seg tilgjengelige for veiledning og generelle spørsmål etter avtale. Eninvest stiller med datamaskin og alt annet utstyr som er nødvendige for dette prosjektet. Eninvest stiller også med tilgang til kontor plass i deres lokaler på Energihuset.

Hos Eninvest er Jan Roger Myklebust, leder for systemutviklingsavdelingen, og Kenneth Kvamme, techlead, tilgjengelige for diskusjon av tekniske detaljer og andre avklaringer rundt prosjektet.

Intern veileder ved Høgskulen på Vestlandet, Per Christian Engdal, veileder generelt i prosjektarbeidet og rapportskrivningen.

2.4 Litteratur om problemstillingen

Chan og Petrikat (2022, s. 118) påstår i en undersøkelse av selvbetjeningsløsninger, at dagens brukere er utdannet, engasjert, teknologisk bevisst og bryr seg om sin egen brukeropplevelse. Det er også en liten sannsynlighet for at en kunde venter til en kunderådgiver er ledig, siden kunden har et behov for raske og troverdige svar. Alt dette tyder på et ønske om muligheten for selvbetjening.

Som et resultat av dette har selvbetjeningsløsninger blitt tatt i bruk i store deler av dagens forretningsverden, hvor økt effektivitet, hastighet og brukervennlighet er viktige faktorer (Chan og Petrikat, 2022, s. 118). Considine og Cormican (2017, s. 35) fant i sin undersøkelse at funksjonalitet, design og muligheten for tilpasning er viktige faktorer for kundetilfredshet.

En undersøkelse utført av Dixon, Ponomareff, Turner og DeLisi (2017) fant at 81% av alle kunder, på tvers av bransjer, forsøker å løse problemet selv før de tar kontakt med en levende representant. Denne mentaliteten har blitt forsterket av pandemien, hvor flere og flere kunder ønsker å være mer individualistisk, siden disse kundene setter pris på enkelhet (Chan og Petrikat, 2022, s. 119). Dette blir videre bekreftet av en undersøkelse utført av Microsoft (2017) med 5000 deltakere som viste at 90% av kunder globalt forventer at en bedrift har en nettbasert selvbetjenings tjeneste. Dette fokuset på individualitet sørger for at kunden får muligheten til å samhandle med varer, tjenester eller virksomheter på sin egen tid. På denne måten vil kundetilfredsheten øke ved at kunden kan utføre det som er ønskelig uten å føle seg forhastet eller presset (Chan og Petrikat, 2022, s. 120).

Chan og Petrikat (2022, s. 122) beskriver at en selvbetjeningsløsning kan gi en 45% økning i nettrafikk og en lik reduksjon av innkommende samtaler. Dette vil derimot kreve at tjenesten er brukervennlig, operativt og vil kunne løse problemet kunden har (Chan og Petrikat, 2022, s. 123). Det er derimot ikke enkelt å utvikle et system som tilfredsstillende alle kravene og fremmer effektiv tjenestelevering satt av kundene og organisasjonen. For at dette skal kunne gjennomføres er det viktig at systemet får tid og de nødvendige ressursene slik at det kan levere de tjenestene de ble designet for. Minst like viktig er muligheten for å tilpasse eventuelle endringer som kan oppstå (Chan og Petrikat, 2022, s. 124).

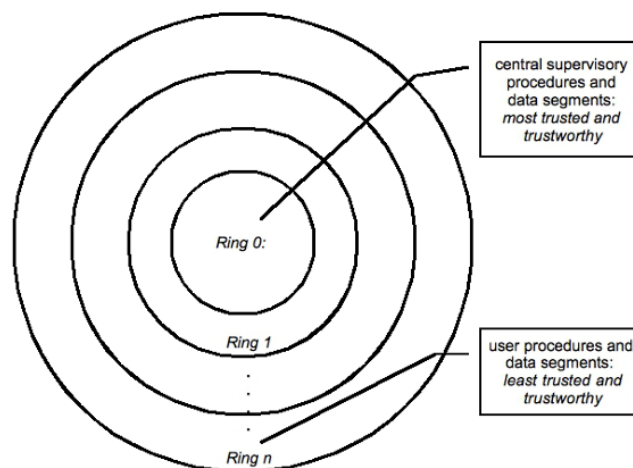
3 DESIGN AV PROSJEKTET

3.1 Forslag til løsning

For å løse problemstillingen skal det utvikles en web-applikasjon der brukere kan logge inn på vegne av en av Eninvest sine bedriftskunder. Hver bruker skal ikke ha tilgang til alt av en bedrifts data og derfor er det nødvendig å utvikle en form for tilgangsstyring. Utvikling av tilgangsstyringen er en stor del av prosjektet og kan gjøres på flere måter. Valg av måten dette skal utføres på blir tatt i samarbeid med Eninvest og er basert på sikkerheten og hvor brukervennlig løsningen er.

3.1.1 Alternativ løsning 1 – Tilgangsnivå

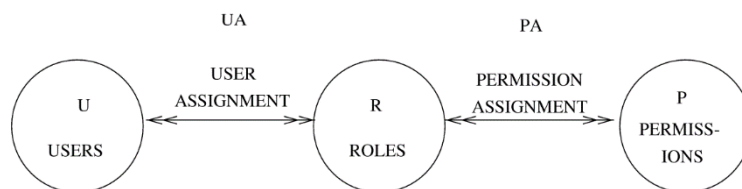
En mulig løsning på tilgangsstyring for applikasjonen er å innføre en modell med ulike tilgangsnivå, der en bruker får tilgang til data og funksjonalitet basert på hvilket tilgangsnivå den hører til. En bruker med det laveste nivået har kun tilgang til det som inngår i dette nivået, mens en bruker med et høyere nivå vil ha tilgang til det nivået og alle nivå under. Dette alternativet gjør det mulig å gi brukere ulik tilgang til systemet og kan dermed sikre sensitive data og funksjoner som ikke alle brukere skal ha tilgang til. Denne løsningen har flere likhetstrekk med Ring-Based Access Control, som brukes i operativsystem. Som Figur 3-1 viser, blir data og prosesser fordelt i ulike ringer med ulikt tilgangsnivå, der den innerste ringen har mest rettigheter og hver ring utover har mindre enn den forrige (Chin og Older, 2010, s. 253).



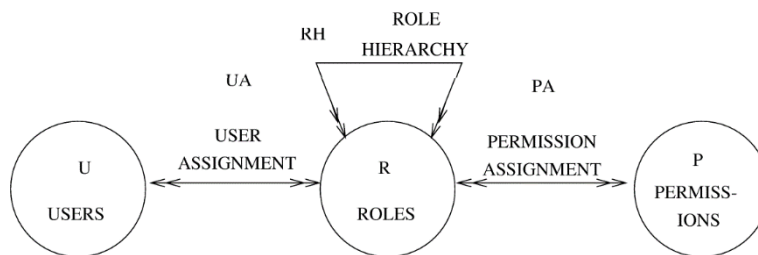
Figur 3-1: Protection rings (Chin og Older, 2010, s. 253)

3.1.2 Alternativ løsning 2 – Role-Based Access Control

Andre alternativ for utvikling av tilgangsstyring er å følge en Role-Based Access Control (RBAC) modell. Denne modellen er delt opp i roller og tilganger, der en tilgang korresponderer til en funksjon eller et endepunkt i systemet og roller er en samling av tilganger. En bruker får tildelt et sett med roller basert på hvilke deler av systemet den har behov for tilgang til. Det finnes flere typer RBAC, deriblant flat RBAC (Figur 3-2) og hierarkisk RBAC (Figur 3-3). Disse skiller i måten rollene forholder seg til hverandre. I flat RBAC er det ingen kobling mellom rollene, mens i hierarkisk RBAC er rollene fordelt i et hierarki, der en bruker med en tildelt rolle også har alle rollene som ligger under den tildelte rollen (Sandhu, Ferraiolo og Kuhn, 2000, s. 47-49).



Figur 3-2: Flat RBAC (Sandhu, et al, 2000, s. 50)



Figur 3-3: Hierarchical RBAC (Sandhu, et al, 2000, s. 52)

3.1.3 Diskusjon av alternativene

Alternativ en, der en bruker får tilgang basert på hvilket tilgangsnivå den hører til, er en enklere modell for tilgangsstyring og vil derfor være mindre krevende å utvikle. Løsningens enkelhet går derimot utover applikasjonens sikkerhet ved at en brukers tilganger ikke kan detaljstyres. For eksempel kan tilgang til faktura og sakslogg ligge på samme nivå, på denne måten får brukeren tilgang til begge selv om den bare har behov for en av dem.

Alternativ to har den fordel at man mer detaljert kan styre hvilke deler av systemet de ulike brukerne har tilgang til. Roller kan settes opp til å dekke små deler av systemet og deles ut til de brukerne som trenger det. Hierarkiet gjør at man kan lage roller som er avhengige av hverandre. For eksempel kan man ha en lese- og en skriverolle for saker, der skriverollen er avhengig av leserollen. Brukere med skriverollen vil dermed også få tilgang til å lese saker. Dette vil gjøre tildeling av roller enklere for brukere med administratorprivilegier, men denne løsningen er mer krevende å utvikle.

3.2 Valgt løsning

I samarbeid med oppdragsgiver er alternativ to, Role-Based Access Control, for utvikling av tilgangsstyring valgt. Det vil bli satt opp et rollestyringssystem som baserer seg på hierarkisk RBAC. Denne løsningen er valgt med bakgrunn i sikkerhetsnivået den tilbyr, og mulighetene for å definere tilgang på et detaljert nivå. Oppdragsgiver ser også muligheter for å anvende denne løsningen i andre deler av applikasjonsporteføljen sin. I tillegg til tilgangsstyring skal funksjonaliteten listet i Tabell 2-1 utvikles som en del av prosjektet. Denne løsningen vil som nevnt være mer tidskrevende, men er foretrukket for en enklere tildeling av roller.

3.3 Valg av verktøy

PHP

PHP er en populær server scripting språk som egner seg spesielt for webutvikling (PHP, (u.å.)). Oppdragsgiver benytter PHP for alle sine applikasjoner.

MySQL

MySQL er verdens mest populære open source database, og er rangert som den andre-mest-populære database. Siden MySQL er open source, er mange av egenskapene utviklet i nært samarbeid med brukere i over 25 år (Oracle, (u.å.)). Oppdragsgiver benytter MySQL for alle sine applikasjoner.

TypeScript

TypeScript er et programmeringsspråk som bygger på JavaScript, ved å legge til ekstra syntaks, og konverteres til JavaScript som fører til at det kan kjøres der JavaScript kan kjøres (TypeScript, (u.å.)). Gruppen velger å bruke TypeScript istedenfor JavaScript, som oppdragsgiver bruker, for dette prosjektet.

React

React er et populært JavaScript rammeverk som gir muligheten til å bygge applikasjoner ut av individuelle komponenter (React, (u.å.)). Oppdragsgiver benytter React for alle sine applikasjoner.

Shadcn

Shadcn er et React-bibliotek for ferdiglagte komponenter som kan kopieres og limes inn i applikasjonen. Fordelen med dette biblioteket, i tillegg til at komponentene er ferdiglagte, er at det er kun nødvendig å legge inn de komponentene som skal brukes (Shadcn, (u.å.)). Shadcn er valgt av gruppen for dette prosjektet.

Tailwind

Tailwind er et Cascading Style Sheets (CSS) rammeverk som inneholder flere ferdig definerte klasser som kan brukes for å lage hvilket som helst design direkte i koden (Tailwind, (u.å.)). Tailwind er valgt av gruppen for dette prosjektet.

Visual Studio Code

Visual Studio Code (VS Code) er et lettvektig kodeverktøy som kan kjøres direkte på datamaskinen og er tilgjengelig for flere operativsystemer. Den kommer med innebygd støtte for JavaScript, TypeScript og Node.js, men har et stort økosystem som tillater utvidelser for språk slik som C++, C#, Java, Python, PHP, Go og .NET (Visual Studio Code, (u.å.)). VS Code anvendes basert på at den kan enkelt tilpasses flere språk og enkel oppkobling til ekstern server over Secure Shell (SSH).

Git

Git er en gratis og open source distribuert versjonskontrollsystem som er laget for å håndtere alt fra små til store prosjekter, med enkel lokal forgrening, praktiske utgangsposisjoner og flere arbeidsflyter (Git, (u.å.)). Oppdragsgiver benytter Git for alle sine applikasjoner.

GitHub

GitHub er en webapplikasjon som tilbyr en plattform for å bygge, skalere og levere sikker programvare (GitHub, (u.å.)). Denne plattformen er allerede tatt i bruk av Eninvest for eksisterende løsninger. Oppdragsgiver benytter Github for alle sine applikasjoner.

Discord

Discord er en tjeneste for kommunikasjon ved tale, video og tekst (Discord, (u.å.)). Denne applikasjonen bruker gruppen under dette prosjektet for kommunikasjon internt mellom medlemmene, samt med veileder.

Figma

Figma er en nettbasert tjeneste for design som gir muligheten for samarbeid i sanntid. Figma gir brukeren muligheten til å designe interaktive og realistiske prototyper (Figma, (u.å.)). Denne applikasjonen bruker gruppen for prototyping og design av webapplikasjonen.

Lucidchart

Lucidchart er en nettbasert applikasjon for utvikling av diagrammer som kan hjelpe visualisere, planlegge og bygge opp virksomheten din (Lucid, (u.å.)). For dette prosjektet bruker gruppen Lucidchart for å utarbeide diagrammer.

Microsoft 365

Microsoft 365 er en sky-basert produktivitet plattform som gir tilgang til applikasjoner slik som Microsoft Teams, Word, Excel, PowerPoint, Outlook, OneDrive og mye mer (Microsoft, (u.å.)). For dette prosjektet bruker gruppen Word for dokumentasjon, Excel for Gantt-diagram og timesføring, PowerPoint for presentasjoner, OneDrive for fildeling mellom gruppen, Teams og Outlook brukes for kommunikasjon og møter.

Docker

Docker hjelper utviklere med å bygge, dele, kjøre og verifisere applikasjoner hvor som helst, uten miljøkonfigurasjon eller administrasjon (Docker, (u.å.)). For dette prosjektet brukes Docker for enkel kjøring av applikasjonen internt hos Eninvest.

NGINX

NGINX er en åpen kildekode-programvare for webtjening, reverse proxying, caching, lastbalansering, mediestrømming og mye mer. Den startet som en webserver designet for maksimal ytelse og stabilitet (NGINX, (u.å.)). For dette prosjektet brukes NGINX for webtjening.

Monday

Monday er en plattform som gir ulike byggeklosser som gjør det mulig å forme ulike arbeidsflyter (Monday, (u.å.)). For dette prosjektet brukes Monday for rapportering av bugs ved brukertesting, da det er dette systemet Eninvest bruker for dette fra før.

3.4 Prosjektmetodikk

3.4.1 Utviklingsmetodikk

Prosjektet har god tilgang på brukerrepresentanter for innhenting av krav, avklaringer og test av løsninger, noe som gjør prosjektet godt egnet for en iterativ og agil gjennomføringsmetode. Det holdes jevnlig dialog med oppdragsgiver for å få tilbakemeldinger på løsningen og for å få hjelp og veiledning i arbeidet. Av den grunn egner den iterative arbeidsmetoden seg. Å jobbe i kortere iterasjoner gjør det enklere å planlegge, ettersom det er en begrenset tidsperiode planen gjelder for. Tilbakemeldinger og prioriteringer gjort sammen med oppdragsgiver kan tas med i planleggingen av den kommende iterasjonen. Dette gjør at det fort kan endres hvilken retning som blir jobbet mot eller hva som blir prioritert når situasjonen rundt prosjektet endrer seg.

I oppstarten av prosjektet benyttes parprogrammering for å bedre sette seg inn i systemet til Enivest. Parprogrammering er en utviklingsteknikk hvor to utviklere jobber sammen på en arbeidsstasjon. Personen som skriver koden, er kalt en driver og personen som observerer og navigerer hver linje av koden er kalt en navigator. Det er flere fordeler med parprogrammering, men for dette prosjektet vil den største fordelen være at et medlem av gruppen jobber for oppdragsgiver, og har derfor gode kunnskaper om systemet. På denne måten vil medlemmene med mindre kunnskaper om systemet og teknologiene brukt få muligheten til å lære fra noen som er godt kjent med disse (Geeksforgeeks, 2023).

På slutten av hver uke blir det fylt ut en statusrapport. Her blir det listet opp hva som er gjort og hvordan arbeidet ligger an i forhold til fremgangsplanen. Dette er en mulighet til å evaluere det som er gjort og måten det er gjort på. Eventuelle problemer blir tatt opp og mulige tiltak for å utbedre problemene blir diskutert. Til slutt settes et mål for den kommende uken, noe som sørger for en felles ide om hva som er viktig å fokusere på.

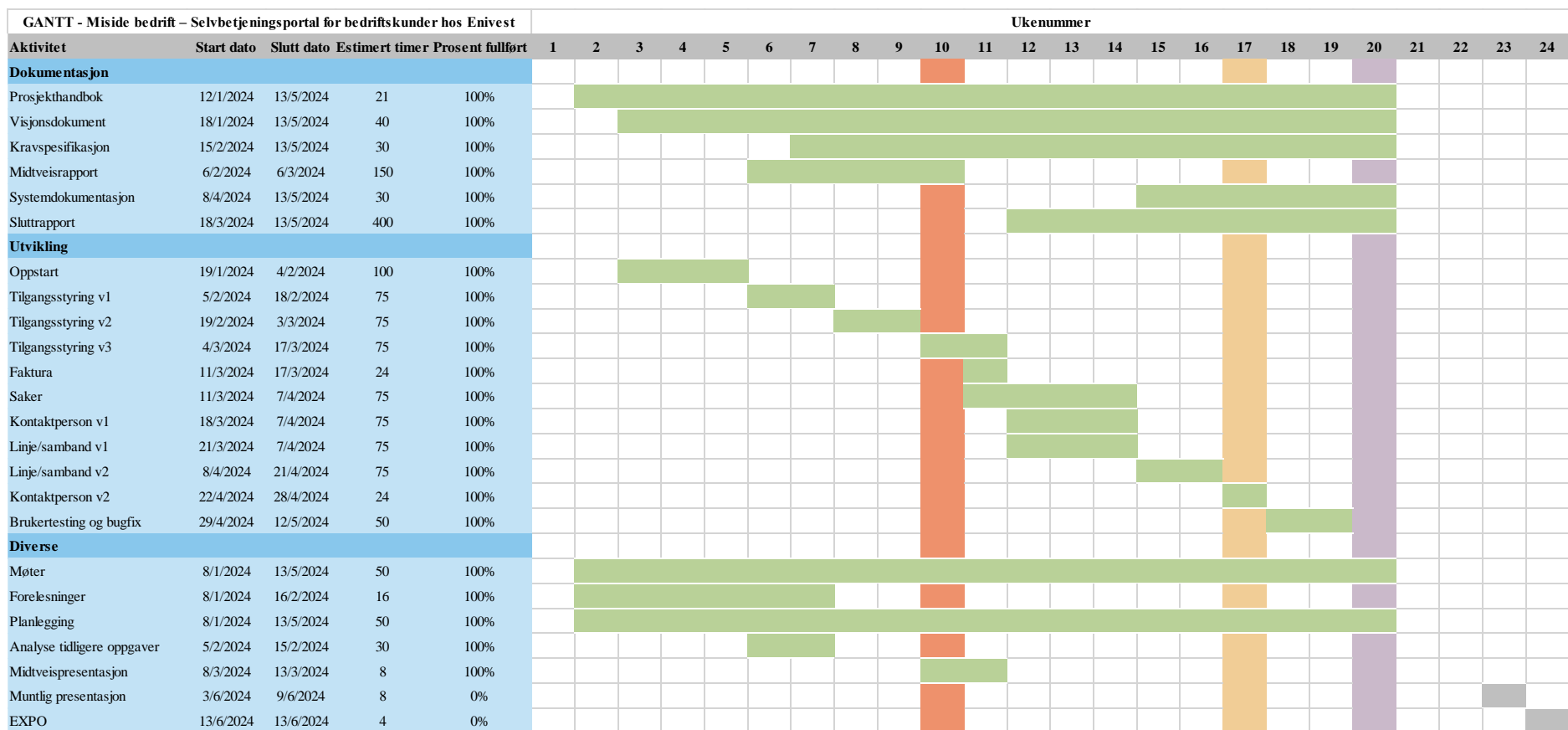
3.4.2 Prosjektplan

Prosjektplanen, vist i Figur 3-5, er utarbeidet som et Gantt-diagram hvor prosjektet er delt opp basert på ulike milepæler. Oppstarten vil fokusere på å bli kjent med eksisterende og nye rammeverk, samt prosjekteiers systemer. Etter oppstarten vil arbeidet omhandle å få tilgangsstyringen over tre iterasjoner, og deretter fokusere på å utvikle resterende funksjonene i iterasjoner. Mot slutten av utviklingen vil det utføres intern brukertesting for å validere funksjonaliteten. Parallelt med utviklingen av løsningen vil prosjektarbeidet dokumenteres i forskjellige dokumenter.

Det er satt opp tre milepæler for prosjektet, som er beskrevet i Figur 3-4. Ved første milepæl er visjon initiale krav og overordnet løsningsdesign. Som en del av dette er utviklingsarbeidet startet med prototyping og utforskning av enkelte funksjoner. Denne milepælen korresponderer også med innlevering av midtveisrapporten. Ved andre milepæl er løsningen designet og ferdig implementert. Etter dette følger en periode med testing og fiksing av feil før siste milepæl, der oppgaven er ferdig og levert inn.

Milepæler	
	Visjon, initiale krav og overordnet løsningsdesign spesifisert
	Løsning designet og ferdig implementert
	Oppgave ferdig og levert inn

Figur 3-4: Gantt - beskrivelse av milepæler



Figur 3-5: Gantt-diagram for organisering og administrering av prosjektet.

3.4.3 Risikovurdering

For dette prosjektet blir det utført en risikoanalyse hvor to hendelser, vist i Tabell 3-1, blir vurdert til å ha høyest risikoprodukt. Den første av disse risikoene er at applikasjonen ikke treffer behovet. Denne risikoen kan komme av manglende innsikt i brukerens behov og kan forårsake at produktet ikke blir tatt i bruk. Dette kan håndteres ved å inkludere ressurser med kunnskap om brukernes behov i planleggingen og utføre brukertesting av webapplikasjonen. Den andre risikoen er at brukere får tilgang til ressurser de ikke skal ha tilgang til. Dette vil komme av en dårlig oppsatt tilgangsstyring eller feil i tilgangsstyringen. Dette kan håndteres ved å utarbeide en god plan for utvikling av tilgangsstyringen og utføre sikkerhetstesting av systemet.

Hendelse/Risiko	Årsak	Sannsynlighet	Konsekvens	Risikoprodukt	Tiltak
Applikasjonen treffer ikke behovet	For lite innsikt i brukeres behov	Høy (4)	Høy (4)	16	Inkludere ressurser med kunnskap om brukernes behov i planlegging og utføre brukertesting.
Misforstått krav og spesifikasjoner	Uklar kommunikasjon, manglende involvering av interessenter	Lav (2)	Svært Høy (5)	10	Ta notater fra møtene, og spør om noe er uklart
Fravær	Sykdom, skade eller andre gyldige grunner	Lav (2)	Lav (2)	4	Viktig å varsle i god tid dersom en ikke kan møte, det vil også være mulig med hjemmekontor
Bruker har vansker med å bruke løsningen	Dårlig utformet grensesnitt	Lav (2)	Høy (4)	8	Gjennomfør brukertesting og følge prinsipper for universell utforming
Mangel på kompetanse	Jobbe innen en bransje som er ny eller ukjent	Middels (3)	Høy (4)	12	Nytte de ressursene som er tilgjengelig under arbeidet for å få størst mulig innsikt i bransjen
Blir ikke ferdig med utvikling av applikasjonen	Dårlig prioritering eller for stort omfang på løsningen	Middels (3)	Høy (4)	12	Jobbe jevnt under hele arbeidsperioden og avgrens løsning slik at den er realistisk å fullføre innen tidsperspektivet
Dårlig samarbeid	Manglende kommunikasjon, uklare roller og ansvar, mangel på felles målsetting	Svært lav (1)	Svært Høy (5)	5	Sørge for å gi konstruktiv og ærlig tilbakemeldinger om fremgangen i arbeidet. Kommunisere.
Ujevn fordeling av arbeidsoppgaver internt i gruppen	Manglende klarhet om ansvarsfordeling, kommunikasjon og deltakelse	Lav (2)	Lav (2)	4	Opprettholde effektiv kommunikasjon. Hold seg oppdatert for å unngå dobbeltarbeid.
Brukere får tilgang til ressurser de ikke skal ha tilgang til	Dårlig oppsatt tilgangsstyring	Middels (3)	Svært høy (5)	15	Utarbeide en god plan for utvikling av tilgangsstyring og utføre sikkerhetstesting

Sannsynlighet	Svært Høy (5)	5	10	15	20	25
	Høy (4)	4	8	12	16	20
	Middels (3)	3	6	9	12	15
	Lav (2)	2	4	6	8	10
	Svært Lav (1)	1	2	3	4	5
		Svært Lav (1)	Lav (2)	Middels (3)	Høy (4)	Svært Høy (5)
Konsekvens						

Tabell 3-1: Risikoanalyse versjon 1 – 23.02.24: En systematisk fremstilling av risiko, deres sannsynlighet og konsekvens, og tiltak for å håndtere disse i prosjektet.

3.5 Evalueringsplan

For å evaluere løsningen er det satt opp en evalueringsplan. Ved å følge denne planen skal det sikres at resultatet tilfredsstillende kravene som er satt. Dette gjelder både at løsningen er funksjonell, at den er sikker, at den treffer behovene og kravene og at den er brukervennlig.

Sammen med oppdragsgiver utføres det jevnlig gjennomganger og evalueringer av det arbeidet som er gjort. Dette sikrer en god fremdrift og at applikasjonen fungerer for brukstilfellene som den er tiltenkt. I disse demonstrasjonene kommer det frem tilbakemeldinger og tips som brukes videre i arbeidet.

Mot slutten av prosjektet planlegges det å utføre en runde med intern brukertesting av applikasjonen. Gjennom denne prosessen vil en gruppe bestående av ansatte i Eninvest teste applikasjonen for å luke bort feil og evaluere løsningen. Problemer som måtte dukke opp kan deretter utbedres før en eventuell retesting.

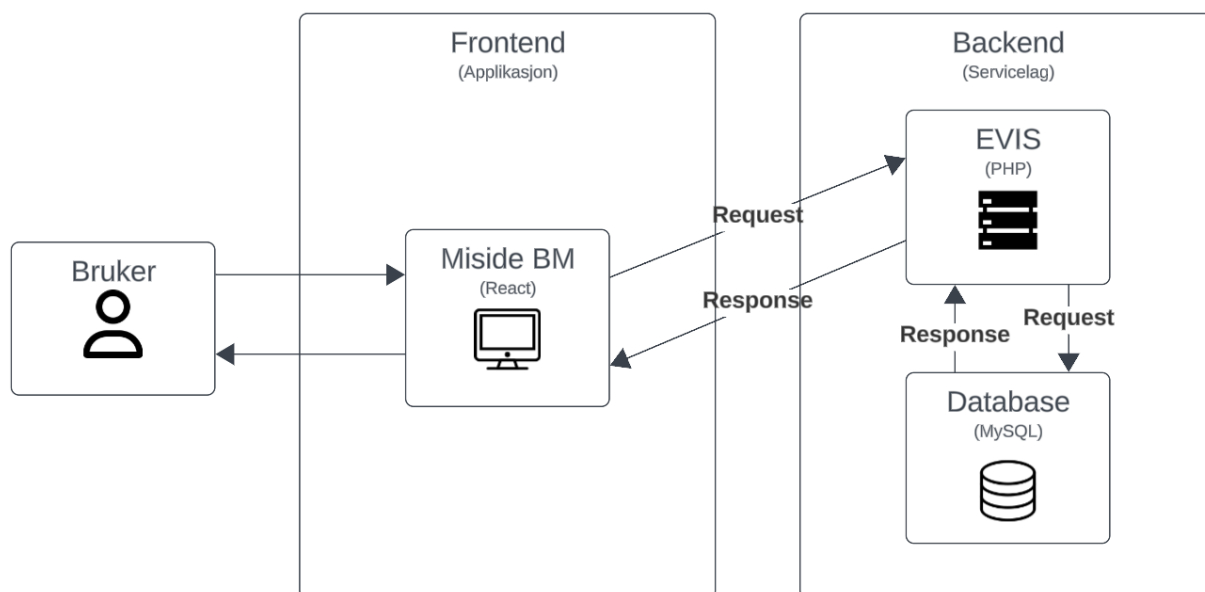
Det planlegges også å utføre en sikkerhetstest av en sikkerhetskonsulent i Eninvest. Gjennom denne testingen vil systemet utsettes for en rekke kjente sikkerhetsangrep for å sikre at det ikke er sårbart for denne typen utnyttelser. I tillegg til dette skrives det enhetstester for delene av systemet med logikk kompleks nok til at det sees på som nødvendig og gunstig. Dette er for å sikre at videre utvikling av koden ikke ødelegger den funksjonaliteten som allerede er der.

Å gjennomføre brukertesting med reelle brukere sees på som en relevant måte å evaluere løsningen på, men på grunn av prosjektets omfang og begrensinger knyttet til tid blir ikke dette utført i løpet av prosjektperioden. Prosjektet vil ikke nå et resultat som oppdragsgiver ser på som hensiktsmessig å demonstrere ovenfor reelle kunder. Derfor kommer denne typen evaluering trolig inn ved et senere tidspunkt etter dette prosjektets slutt.

4 DETALJERT LØSNING

4.1 Arkitekturoversikt

Eninvest sitt system følger en tradisjonell lagdelt arkitektur, og er inndelt i en applikasjon og et servicelag. Figur 4-1 viser hvordan dette er strukturert og samhandlingen mellom komponentene. Applikasjonen er utviklet i JavaScript-rammeverket React og servicelaget er utviklet i programmeringsspråket PHP. Data blir lagret i en MySQL-database. En bruker vil nytte løsningen gjennom applikasjonen, som sender spørringer mot servicelaget der logikk og datahåndtering blir utført og sendt tilbake. Den nye dataen som applikasjonen mottar, blir brukt til å oppdatere det som vises på brukerens skjerm.

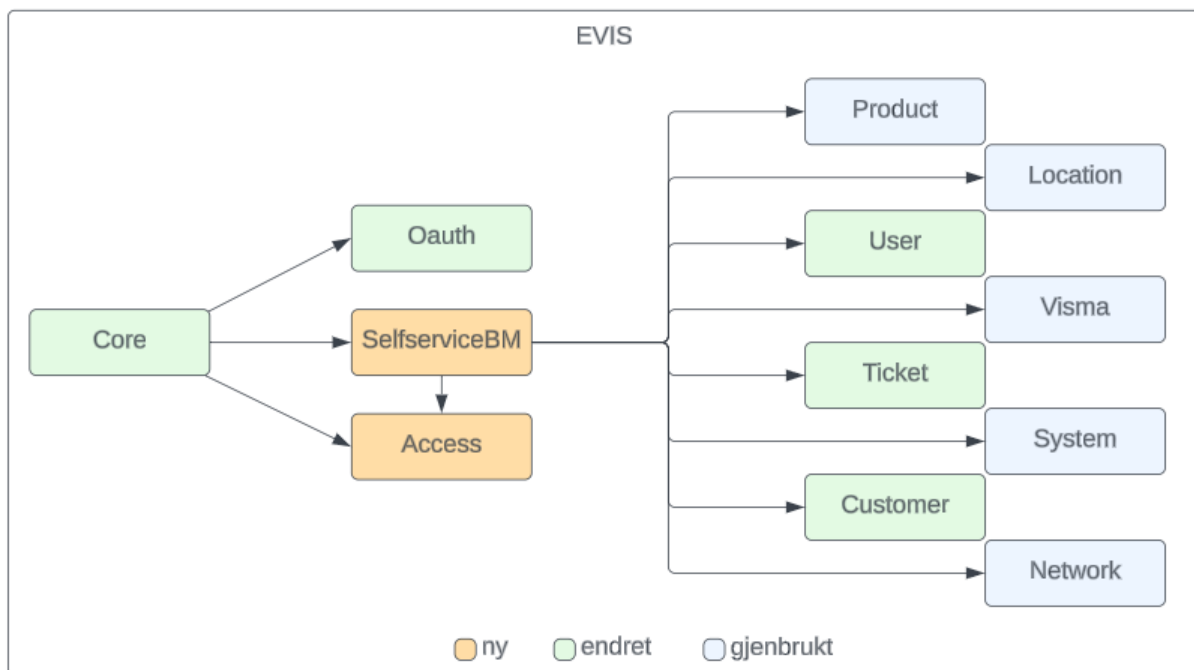


Figur 4-1: Overordnet arkitekturskisse

4.2 Servicelag

4.2.1 EVIS

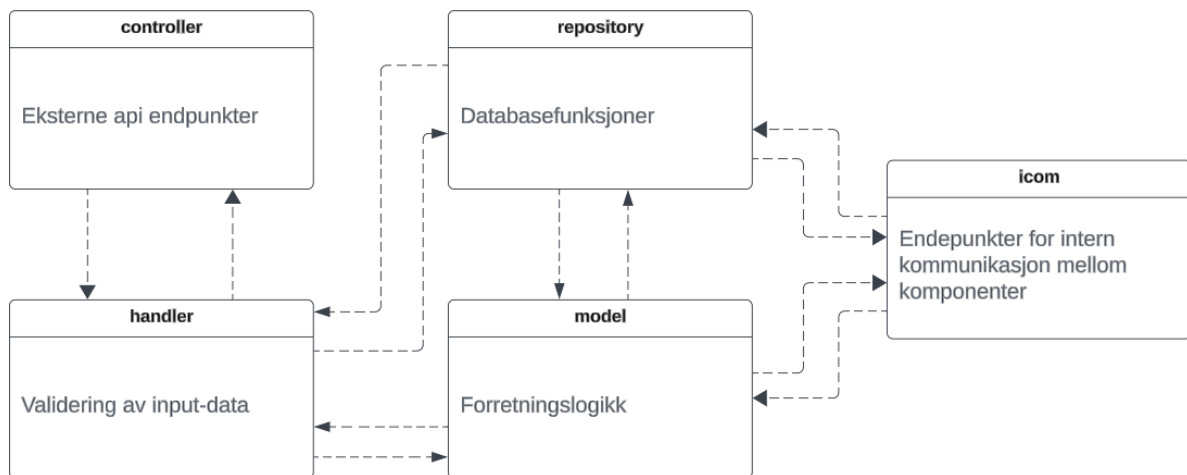
EVIS fungerer som nevnt i kapittel 2.1.1 som et felles API for alle Eninvest sine applikasjoner. API-et er bygget opp av flere komponenter som håndterer ulike behov og funksjonaliteter. Alle komponenter tilbyr et sett med endepunkter eksternt. Komponentene kommuniserer også seg imellom internt i systemet. Gjennom prosjektet er det introdusert to nye komponenter, SelfserviceBM og Access. SelfserviceBM har som oppgave å sette opp endepunktene som skal tilgjengeliggjøres for applikasjonen som utvikles for løsningen. I hovedsak aggregerer komponenten funksjonalitet og data fra andre komponenter som den gjør tilgjengelig for applikasjonen gjennom sine endepunkter. Access inneholder tilgangsstyringsfunksjonaliteten som er utviklet for dette prosjektet. Kun SelfserviceBM benytter seg av dette per nå, men for Eninvest vil det være ønskelig å ta i bruk en vellykket implementasjon i resten av systemet. Figur 4-2 viser de komponentene som er involvert i løsningen.



Figur 4-2: Serverarkitekturskisse EVIS: komponenter involvert i løsningen

4.2.2 Oppbygningen av en komponent

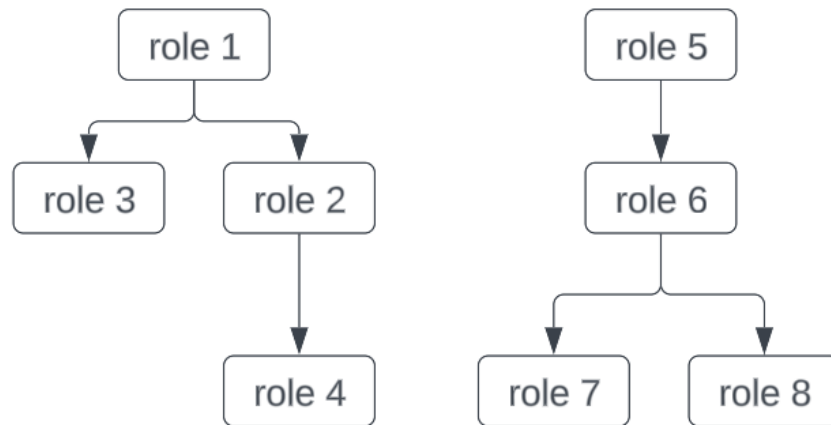
En av avgrensningene for dette prosjektet var som nevnt i kapittel 2.2 er at kode og databasetabeller skal følge eksisterende arkitektur og design. Denne strukturen for komponentene er vist i Figur 4-3, hvor en komponent består i hovedsak av de fem typene klasser. Controller-klasser setter opp endepunkter og sender spørringer videre til korrekt handler. Handler-klasser validerer input-dataen som ligger på en spørring og sender denne dataen videre til enten en repository eller en model. Model-klasser inneholder forretningslogikk og er ment å være gjenbrukbare. Repository-klasser definerer et sett med databasefunksjoner. Icom-klasser setter opp endepunkter som blir brukt for intern kommunikasjon imellom ulike komponenter.



Figur 4-3: Arkitekturskisse komponent

4.2.3 Access-komponenten

Som beskrevet i kapittel 3.2 ble det valgt å utvikle tilgangsstyringen for applikasjonen i form av en hierarkisk RBAC. Hver bruker får tildelt et sett med roller, der hver rolle gir brukeren tilgang til et begrenset sett med endepunkter i systemet. Rollene satt opp i et hierarki, som dannes ved at hver rolle kan referere til en rolle over seg. Dette gjør at rollene i systemet skaper trær som illustrert i Figur 4-4. På denne måten kan opprettes koblinger mellom roller som er avhengige av hverandre.



Figur 4-4: Illustrasjon av rollehierarki

Alle roller må knyttes til en klient. Brukere med rollen vil derfor bare få tilgang til funksjonene hvis spørringene blir sendt i fra den samme klienten som rollen er knyttet til. Dette gjør at brukere ikke kan misbruke rollene sine i en annen klient enn den som er tiltenkt.

Når en bruker får tildelt en rolle blir det også lagret hvilken kunde som tildelte rollen, dersom rollen er tildelt brukeren på vegne av en kunde. På den måten kan brukere som har tilgang til flere kunder ha ulike roller for de forskjellige kundene. Når det defineres hvilke roller som er nødvendig for et endepunkt må det spesifiseres om rollen må være tildelt på vegne av en kunde.

For å kunne styre hvilke sider som er tilgjengelig for brukere i applikasjonen brukes features. En feature tilsvarer en side eller funksjon i applikasjonen og gjøres tilgjengelig for brukere basert på rollene som de er tildelt. En feature kan ha flere roller knyttet til seg, der brukeren må ha minst en av rollene for at siden skal vises i grensesnittet. Disse kan også skrues av slik at de ikke vises uavhengig om brukeren har korrekt rolle eller ikke.

4.2.4 SelfserviceBM-komponenten

SelfserviceBM-komponentens hovedoppgave er å sette opp alle api-endepunktene som skal være tilgjengelig for applikasjonen utviklet i dette prosjektet. Det blir gjort på denne måten fordi disse endepunktene skal brukes av kunder, i motsetning til andre endepunkter i EVIS som kun benyttes internt i applikasjoner for ansatte og andre personer knyttet til Eninvest.

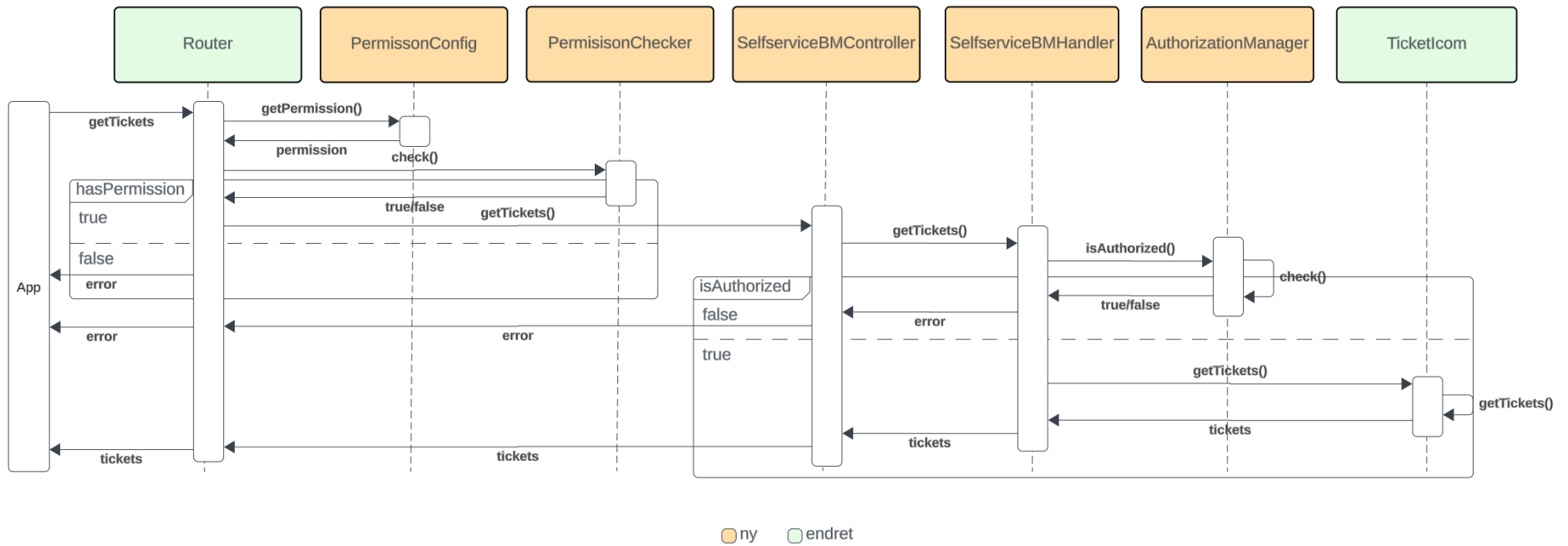
Dette følger samme stilen som den eksisterende Selfservice-komponenten, som brukes til Miside for Eninvest sine privatkunder. Vedlegg IV: Systemdokumentasjon inneholder en liste over alle endepunkter som er utviklet for dette prosjektet med forklaring.

Da disse endepunktene skal brukes av kunder er det ekstra sikkerhetssjekker knyttet til hver av dem. For hver spørring som blir sendt til komponenten blir det utført et antall sjekker basert på hvilke data spørringen prøver å endre eller hente ut. Det sjekkes at brukeren som sender spørringen har tilgang til kunden som spørringen blir sendt på vegne av. Kunden ligger i brukerens token, men en bruker kan ha mistet tilgangen til kunden imellom tildeling av token og spørringen, derfor må det sjekkes at koblingen mellom brukeren og kunden fortsatt er gyldig. I tillegg til dette vil det i de tilfellene der det er nødvendig, sjekkes om ressursen som prøver å nå tilhører kunden i spørringen. Dette er for å unngå at kunder får tilgang til andre kunders data.

4.2.5 Sekvensdiagram

I dette kapittelet kommer sekvensdiagram som beskriver flyten for en spørring gjennom servicelaget i løsningen. Sekvensdiagrammet i Figur 4-5 viser flyten til en spørring, i dette tilfellet brukstilfellet "se sakslogg", men alle spørringer vil ha tilnærmet samme flyt. Det som vil variere for de ulike spørringene, er siste delen av diagrammet der det kommuniseres med en annen komponent i systemet, i dette tilfellet ticket-komponenten.

Router-en finner ut hvilken komponent spørringen skal sendes til. Den spør PermissionConfig om hvilke roller som er nødvendig for endepunktet som prøver å nå. Dette blir sendt videre til PermissionChecker-klassen som sjekker at brukeren har de påkrevde rollene. Dersom sjekken ikke er ok, blir det sendt en feilmelding tilbake til appen. Ellers blir spørringen sendt videre til SelfserviceBM-komponenten. Her starter den i controlleren, som sender videre til handleren. Handleren sjekker med AuthorizationManager at brukeren har lov til å gjøre det den prøver å gjøre, for spørringen til slutt blir sendt videre til en annen komponent for å hente eller lagre data. Resultatet blir sendt tilbake til appen.



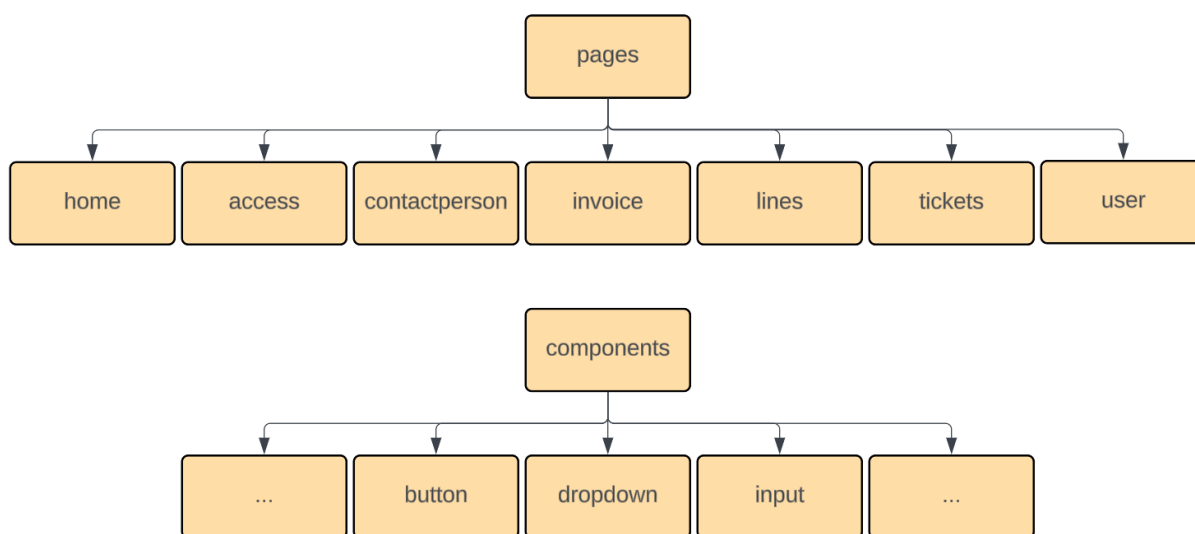
Figur 4-5: Sekvensdiagram for spørring `getTickets`

4.3 Applikasjon

4.3.1 Teknologi og struktur

Applikasjonen er utviklet med JavaScript-rammeverket React. En av styrkene til React er at det er enkelt å splitte komplekse frontend-strukturer opp i mindre komponenter som kan gjenbrukes flere steder i applikasjonen. Denne måten å strukturere applikasjonen på, gjør det enkelt å endre eller oppdatere elementer, da det kun må gjøres en plass.

Hver side i applikasjonen er en egen komponent som er bygd opp av flere komponenter etter behov. I tillegg til sidene er det laget mange gjenbrukbare komponenter, som knapp, nedtrekks-meny og tekstboks. Noen av disse er hentet i fra komponentbiblioteket Shadcn, mens andre er laget spesielt for denne applikasjonen. Disse komponentene brukes over alt i applikasjonen. Figur 4-6 viser en forenklet versjon av hvordan dette er strukturert.



Figur 4-6: Forenklet komponent-struktur i applikasjonen

4.3.2 Skisser

Skissene presentert i dette kapitlet er utformet fra brukstilfeller utarbeidet for dette prosjektet. En detaljert oversikt over de ulike brukstilfellene og fullstendige skisser av nettsiden er tilgjengelig i Vedlegg III: Kravdokument.

4.3.2.1 Tilgangsstyring

Siden for tilgangsstyringen vil vise en liste over de ulike brukerne som har tilgang til en kunde. Fra denne siden vil en bruker med rollen administrator ha muligheten til å fjerne en bruker, endre på en brukers roller samt legg til en ny bruker til kunden. Rollene som en bruker har for kunden, vil vises som et hierarkisk tre for at det skal være enkelt å se forholdet mellom de ulike rollene. En administrator kan på denne siden fjerne eller legge til en eller flere roller en bruker har. En administrator vil også ha muligheten til å legge til en bruker via epost. Når en bruker legges til kan brukeren akseptere invitasjonen fra sin profil i løsningen, hvis derimot brukeren ikke har en profil så vil det sendes en invitasjon på epost-adressen oppgitt.

4.3.2.2 Faktura

En annen funksjon for løsningen vil være å gi kundene muligheten til å hente og se faktura samt kreditnota. Denne siden vil også vise informasjon om hver enkelt faktura og gi brukeren muligheten til å laste ned en gitt faktura i PDF-format, vist i Figur 4-7. Det vil også være mulig for brukeren å gjøre enkelte endringer i forhold til faktureringen, slik som for eksempel faktureringsintervall og fakturamottaker.

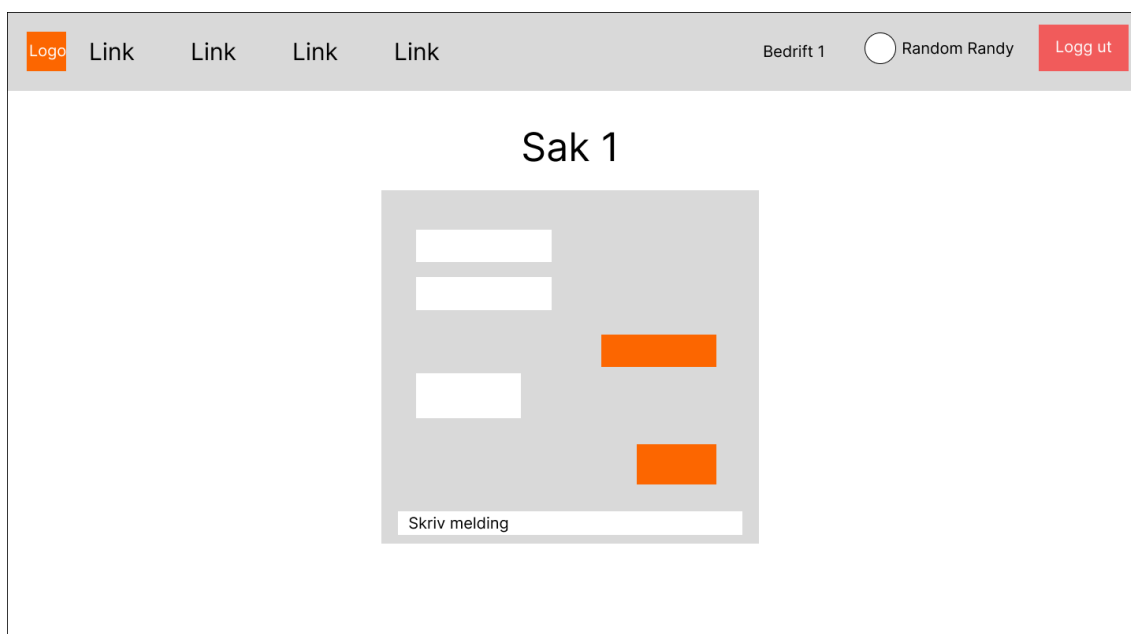


faktura	dato	last ned

Figur 4-7: Skisse faktura – oversikt over faktura sendt til en kunde

4.3.2.3 Saker

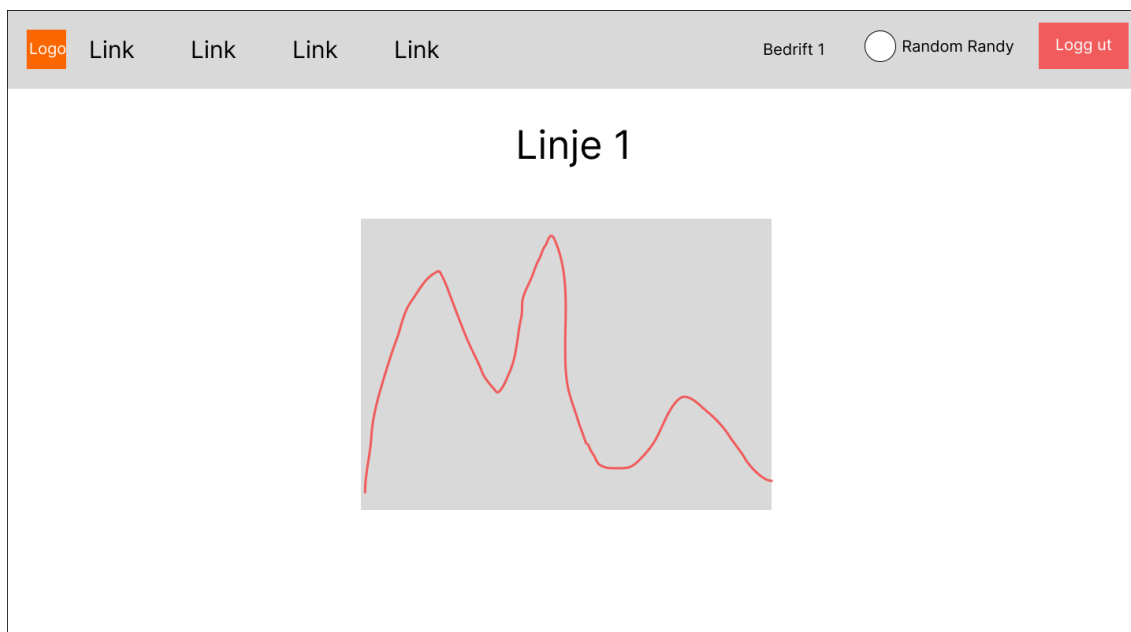
Det er designet en egen side for å samle alle sakene en kunde har hos Eninvest, som gjelder for både åpne og stengte saker. Disse sakene vil vises i en liste og ved å klikke på en sak fra listen vil brukeren bli sendt inn til en meldingsboks slik at brukeren har direkte kommunikasjon med Eninvest for den gitte saken, vist i Figur 4-8. Kommunikasjon via andre kanaler slik som epost og tekstmelding vil også vises på denne siden. På denne måten vil all kommunikasjon og informasjon knyttet til en sak samles på en side.



Figur 4-8: Skisse saker - meldingsboks for en valgt sak

4.3.2.4 Linjer

En kunde vil ofte ha flere nettverkslinjer hos Eninvest og det vil derfor være nødvendig å se disse linjene samt informasjonen knyttet til dem. Alle linjer en kunde har hos Eninvest vil listes ut på en egen side. Når brukeren har valgt en linje vil informasjonen vises på en egen side, hvor informasjon slik som kostnad, hastighet og drift status er noe av informasjonen som kan vises. Her vil det også vises en trafikkgraf, vist i Figur 4-9, som vil vise mengden trafikk på linjen i forskjellige tidsintervaller.



Figur 4-9: Skisse linjer - en valgt linje med trafikkgraf

4.3.2.5 Kontaktperson

En kunde vil i enkelte tilfeller ha behov for kontaktpersoner og varslingspersoner for ulike deler av forholdet med Eninvest. Siden for kontaktperson gir brukeren muligheten til å se, endre og fjerne disse kontaktpersonene. En kontaktperson eller varslingsperson skiller seg fra brukere av løsningen da disse ikke trenger tilgang til tjenesten.

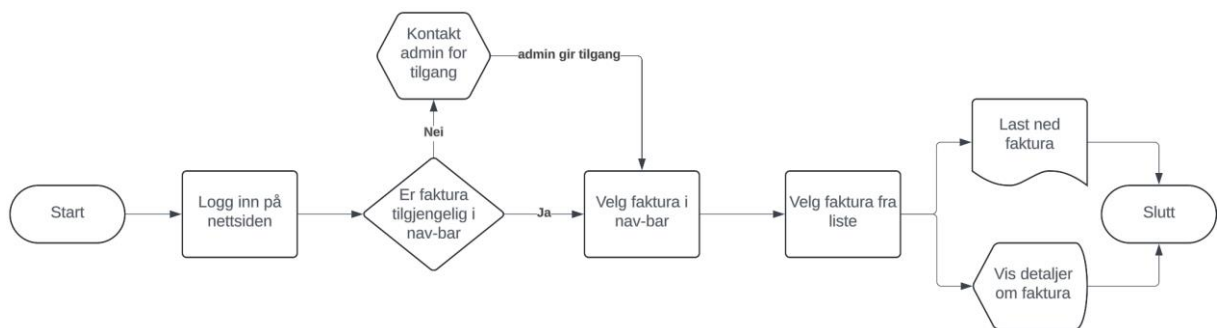
4.3.2.6 Hjemmeside og navigasjonsbar

Hjemmesiden er designet for å inneholde forskjellige moduler for å gi brukeren informasjon og navigering. Et eksempel på en slik modul er valg av kunde, som gir brukeren muligheten til å velge en av kundene brukeren har tilgang til. En annen modul designet for å være på hjemmesiden er et kort som vil inneholde informasjonen til kundens rådgiver. Det er ikke alle kundene som har en egen rådgiver derfor vil denne modulen vil kun vises for kunder som har fått en rådgiver tildelt. Navigasjonsbaren inneholder en logo, lenker til de andre sidene av løsningen og muligheten for utlogging. Lenken til en gitt side vil kun være synlig for de som har tilgang til den gitte siden.

4.3.3 Flytdiagram

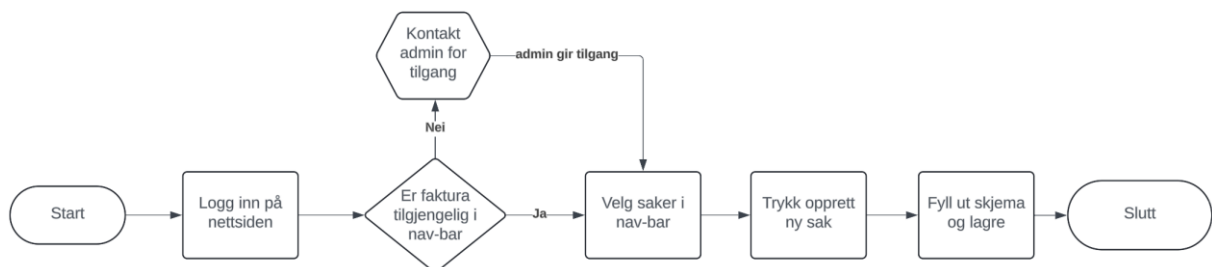
Under er det vist og beskrevet noen flytdiagram. Disse viser flyten en bruker vil følge gjennom brukergrensesnittet for å utføre noen utvalgte handlinger i applikasjonen.

Figur 4-10 viser flyten en bruker må følge for å hente ut informasjon om en faktura. Dersom brukeren ikke har tilgang til denne funksjonen, må den først få gitt tilgang av en administrator. Deretter navigerer brukeren til siden for faktura. Her velger brukeren fakturaen fra listen og får sett detaljer eller lastet ned fakturaen.



Figur 4-10: Flytdiagram - hente ut informasjon om fakturaer

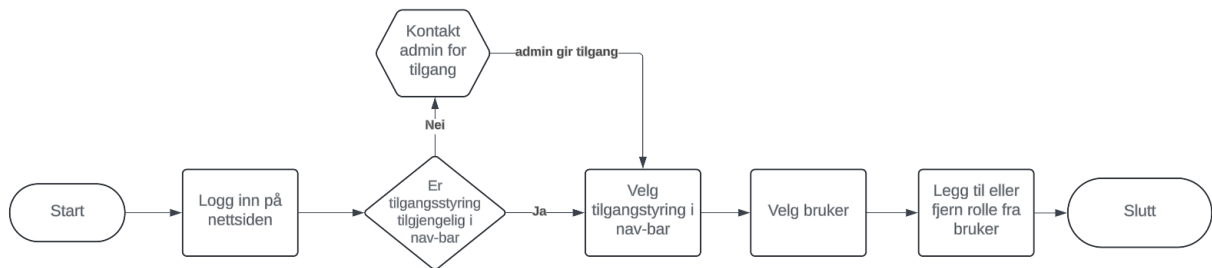
Illustrasjonen i Figur 4-11 viser prosessen en bruker utfører for å opprette en sak. Figuren viser hvordan en bruker vil gå frem for å opprette en sak i applikasjonen. For å gjøre dette navigerer brukeren til saks-siden, der kan han fylle ut et skjema og med det opprette en sak.



Figur 4-11: Flytdiagram - opprette ny sak

Diagrammet i Figur 4-12 viser flyt for å endre en brukers rolle. En bruker som har tilgang til denne funksjonaliteten, kan navigere til tilgangsstyringssiden og endre rollen til brukere

knyttet til kunden. Dette gjøres ved å velge en bruker, for så å huke av for de rollene som brukeren skal ha.



Figur 4-12: Flyttdiagram – endre brukers rolle

5 RESULTATER

5.1 Evalueringemetode

5.1.1 Verifisering

For å verifisere at koden som blir skrevet fungerer som den skal og at den fortsetter å fungere ved endringer, er det skrevet enhetstester. Dette er gjort der det er nødvendig og mulig. Flere deler av applikasjonen er ikke strukturert slik at enhetstesting enkelt lar seg gjøre, som for eksempel frontend-koden, men der det er mulig brukes denne metoden for å verifisere koden. En annen verifiseringsmetode som er tenkt å nytte er sikkerhetstesting. Dette gjøres med mål om å sjekke at applikasjonen ikke er sårbar for vanlige sikkerhetsproblemer.

5.1.2 Validering

For å være sikker på at applikasjonen løser de riktige problemene og at funksjonene utviklet fungerer som det er tenkt at de skal, blir det jevnlig utført møter oppdragsgiver. Ved disse møtene blir løsningen demonstrert og veiledere ved Eninvest kommer med tilbakemeldinger på hva som er bra og hva som bør gjøres annerledes. På denne måten blir det validert at applikasjonen treffer de problemområdene som er tiltenkt. I tillegg til disse møtene, blir det mot slutten av prosjektet utført brukertesting for å validere applikasjonen.

5.1.3 Brukertestning

For brukertestingen lages en liste over funksjonalitet som skal testes og en spørreundersøkelse som deltakerne skal fylle ut (Vedlegg VI: Brukertestning). Fem ansatte i Eninvest utfører testen, rapporterer feil og svarer på en spørreundersøkelse knyttet til testen. Testgruppen består av ansatte med roller innenfor nettverksteknologi, utvikling, økonomi, salg på bedriftsmarkedet og leveranse. Gjennom disse rollene har testerne relevant kunnskap og erfaring som gjør dem egnet til å teste og gjøre en vurdering av løsningen.

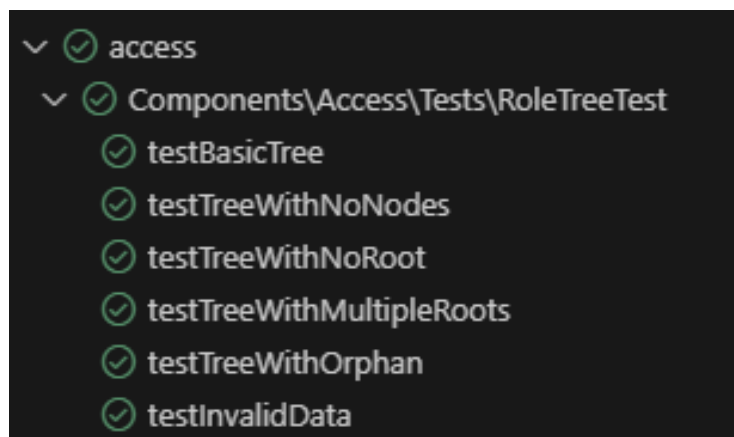
Spørreundersøkelsen gir deltakerne mulighet til å gi tilbakemelding og tanker om løsningen. Det stilles to spørsmål om løsningen som en helhet og for hver av sidene stilles fire gjengående spørsmål:

- "Hvor enkel syntes du siden for X er å forstå og bruke?" som rangeres på en skala fra "meget vanskelig" til "meget enkel".
- "Om du skal vurdere verdien til funksjonaliteten som tilbys på siden for X for din egen arbeidshverdag, hvordan vil du si at den er?" som rangeres på en skala fra "ingen verdi" til "høy verdi", samt et ekstra alternativ for "ikke relevant".
- "Om du skal vurdere verdien til funksjonaliteten på siden for X for Eninvest sine bedriftskunder og deres arbeidshverdag, hvordan vil du si at den er?" som rangeres fra "ingen verdi" til "høy verdi".
- "Hva kan forbedres med siden for X?", hvor deltakere kan gi tilbakemelding om den gitte siden.

5.2 Evalueringsresultat

5.2.1 Verifisering

Mange deler av koden er ikke strukturert på en måte som lar seg enhetsteste og med dette er verifisering utført gjennom manuell testing av utviklere flere steder i koden. Der det lar seg gjøre er det skrevet enhetstester og resultatet fra dette er vist i Figur 5-1. Enhetstestene som er skrevet gjør at endringer av koden kan gjøres med større sikkerhet og trygghet på at det ikke ødelegger funksjonaliteten.



Figur 5-1: Resultat enhetstesting

Verifisering i form av sikkerhetstesting blir planlagt, men blir ikke utført grunnet manglede tid og problemer med koordinering med sikkerhetskonsulent mot slutten av prosjektet.

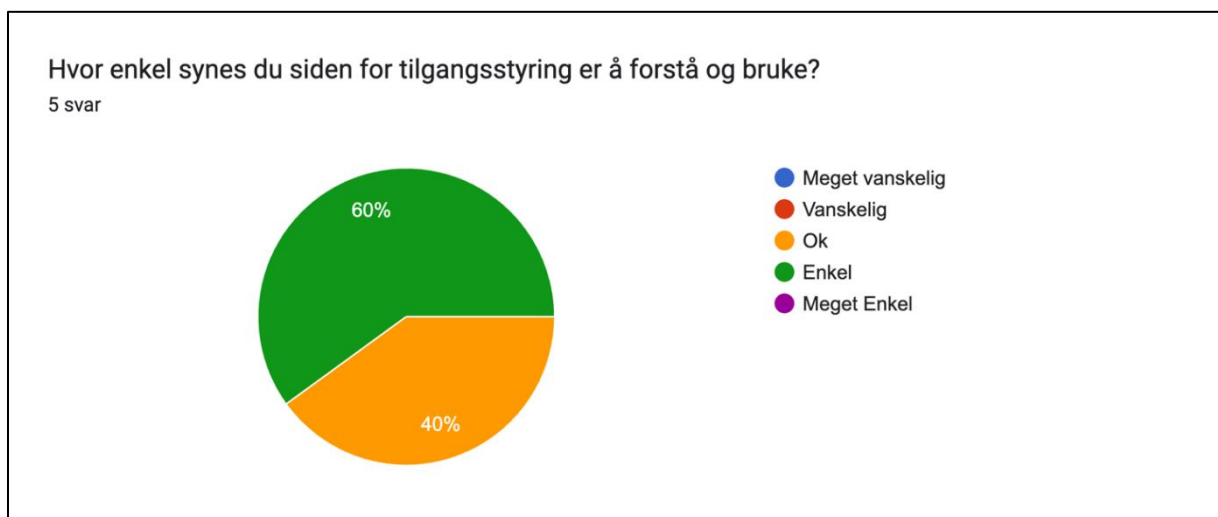
5.2.2 Validering

Demonstrasjonene med oppdragsgiver sørger for at prosjektet holder seg på det rette sporet og er til nytte for oppklaring i ulike spørsmål som oppstår. For eksempel presenteres og diskuteres de alternative løsningene tiltenkt for tilgangsstyringen, hvor konklusjonen fra oppdragsgiver er at det ansees som en god måte å håndtere rollene på. Av de to alternativene foretrekker oppdragsgiver den hierarkiske løsningen. Funksjonalitet slik som faktura, brukerprofil, saker og linjer demonstreres, hvor den tette oppfølgingen sørger for at disse utvikles korrekt. Demoene gir også oppdragsgiver muligheten til å komme med forslag til forbedringer, for mer detaljer se referat i Vedlegg II: Prosjekthåndbok.

Ved et spontant møte med digital markedsfører hos Eninvest, som også designer de ulike digitale tjenestene, utføres det en demo av løsningen for å få feedback på utformingen av løsningen og hvor godt dette passer med Eninvest sine andre systemer. Som et resultat av demonstrasjonen avdekkes og dokumenteres mindre endringer og forbedringer til designet. For mer detaljer se referat fra møtet i Vedlegg II: Prosjekthåndbok.

5.2.3 Brukertesting

Resultatet fra spørreundersøkelsen stilt ved brukertesten viser at 60% av deltakerne syntes siden for tilgangsstyring var enkel å forstå og bruke, mens 40% syntes den var ok (Figur 5-2).

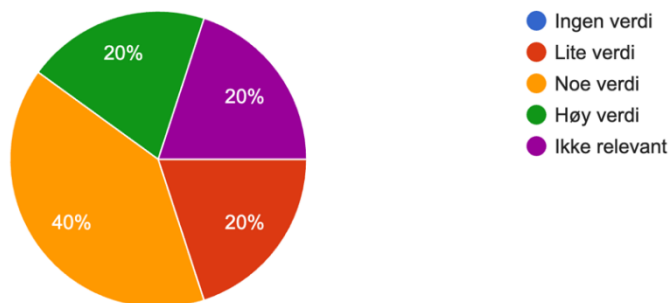


Figur 5-2: Resultat spørreundersøkelse - viser brukerens vurdering av hvor enkelt det er å forstå og bruke tilgangsstyring siden

Resultatet av spørsmål to viser en større variasjon, hvor to deltakere mente at siden for tilgangsstyring har noe verdi for deres arbeidshverdag. De resterende tre deltakerne er spredt på at denne siden har lite verdi, høy verdi eller ikke er relevant for deres arbeidshverdag (Figur 5-3).

Om du skal vurdere verdien til funksjonaliteten som tilbys på siden for tilgangsstyring for din egen arbeidshverdag, hvordan vil du si at den er?

5 svar

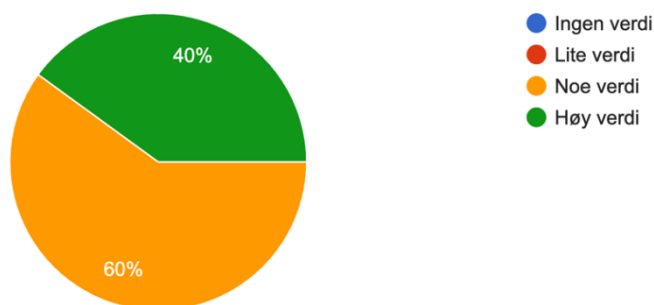


Figur 5-3: Resultat spørreundersøkelse - viser brukerens vurdering av verdien til funksjonaliteten som tilbys på siden for tilgangsstyring for din egen arbeidshverdag

Deltakerne av undersøkelsen mente at siden for tilgangsstyring har derimot mer verdi for kundene, hvor 60% mente at siden har noe verdi og resterende 40% mente at den har høy verdi (Figur 5-4).

Om du skal vurdere verdien til funksjonaliteten på siden for tilgangsstyring for Eninvest sine bedriftskunder og deres arbeidshverdag, hvordan vil du si at den er?

5 svar



Figur 5-4: Resultat spørreundersøkelse - viser brukerens vurdering av verdien til funksjonaliteten på siden for tilgangsstyring for Eninvest sine bedriftskunder og deres arbeidshverdag

For å se fullstendig resultat av spørreundersøkelsen, innrapporterte feil og tilbakemeldingene gitt fra brukertesten se Vedlegg VI: Brukertesting.

5.3 Prosjektresultat

Planlagt funksjonalitet som utvikles for løsningen er vist i Tabell 5-1. Av de totalt tolv kjernefunksjonalitetene som utvikles, er det tre som ikke implementeres grunnet manglende tid. Disse tre er "oversikt over og endre fakturering" for faktura og "linjestatus" samt "informasjon om feil og planlagt arbeid" for linje. Dette kapittelet vil gi en beskrivelse av endelig resultat for dette prosjektet, se Vedlegg V: Skjermbilder av endelig løsning."

Kategori	Funksjonalitet	Implementert
Tilgangsstyring	Innlogging på vegne av	✓
	Lese- og skriverettigheter	✓
Faktura	Laste ned fakturakopi	✓
	Oversikt over og endre fakturering	✗
Saker	Sakslogg	✓
	Opprett ny sak	✓
Linje	Linjestatus	✗
	Trafikkgrafer	✓
	Informasjon om feil og planlagt arbeid	✗
	Oversikt over leverte samband	✓
Kontaktperson	Endre kontakt- og varslingsperson	✓
Generelt	Se bedriftens rådgiver	✓

Tabell 5-1: Kategorisert funksjonalitet som ble utviklet for dette prosjektet

5.3.1 Tilgangsstyring

Hvilke roller som kreves for å bruke et endepunkt blir definert i `PermissionConfig` som vist i Figur 5-5. Her vil det for hvert endepunkt i hver komponent som bruker tilgangsstyringsfunksjonaliteten bli definert en liste med roller og om rollen må være tildelt brukeren på vegne av en kunde. Det siste blir gjort gjennom det boolske flagget i `P` funksjonen.

```
1 class PermissionConfig {
2
3     public static function getPermission(string $class, string $function): ?Permission {
4         $component = self::getComponentName($class);
5
6         return match ($component) {
7             "Selfservicebm" =>
8                 match ($function) {
9                     "getUser"           => self::P(["msb_basis"]),
10                    "getConnectedUsers" => self::P(["msb_access_control_r"], true),
11                    "deleteUserConnection" => self::P(["msb_access_control_w"], true),
12                    // ...
13                    default => throw new \Exception("Permission not found", 400)
14                },
15            default => null
16        };
17    }
18 }
```

Figur 5-5: `permissionconfig.class.php`

Konfigurasjonen definert i `PermissionConfig` benyttes videre i `PermissionChecker`, vist i Figur 5-6. Den sjekker om brukeren har fått tildelt en av rollene som er påkrevd for endepunktet og at rollen er gyldig for klienten som spørringen sendes i fra. Dersom det kreves at rollen er tildelt på vegne av en kunde utføres også denne sjekken.

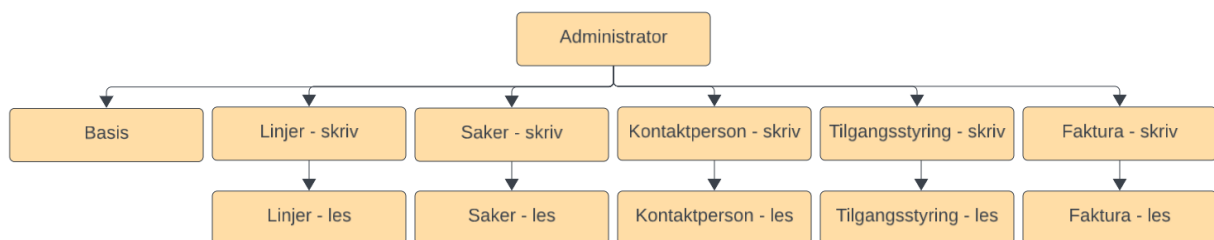
```

1 class PermissionChecker {
2
3     public static function check(Permission $permission) {
4         try {
5             $user_id = self::getUserId();
6             $customer_id = $permission->customer_required ? self::getCustomerId() : null;
7             $client_id = self::getClientId();
8
9             $repository = (new Factory())->createDatabaseRepository();
10            $user_roles = $repository->listRolesForUser($user_id, $customer_id, $client_id);
11            foreach ($user_roles as $ur) {
12                if (in_array($ur->get("code"), $permission->roles)) {
13                    return;
14                }
15            }
16        } catch (\Throwable) {}
17
18        throw new \Exception("Permission not found", 400);
19    }
20 }

```

Figur 5-6: permissionchecker.class.php

Brukere som har tilgang til siden for tilgangsstyring, vil kunne se alle brukerne som har tilgang til den gitte kunden. Her vil en administrator se og endre rollene som er tildelt for enhver bruker. Utlistingen av rollene er satt opp for å vise hierarkiet mellom de ulike rollene, hvor dette er illustrert i Figur 5-7. På denne siden kan også brukere inviteres eller fjernes fra kunden.



Figur 5-7: Roller laget for løsningen vist hierarkisk

5.3.2 Faktura

Faktura siden vil vise alle faktura og kreditnota for den valgte kunden. Som vist i Figur 5-8 vil brukeren ha muligheten til å se mer detaljer om fakturaen eller kreditnotaen, slik som forfallsdato, status, beløpet, KID og kontonummer. Det er også mulig å laste ned fakturaen eller kreditnotaen i PDF-format, samt filtrere på ubetalte faktura.

Heim Saker Linjer **Faktura** Kontaktpersonar Tilgangsstyring Test McTest AS Truls Haukeland Logg ut

Faktura

Alle Ikkje betalt

Utgående faktura: 2042365 Lukk Last ned

Forfallsdato	09.04.2024
Beløp	590,-
Restbeløp	450,-
KID	3112720423656
Kontonummer	37050322816

Utgående faktura: 1990938 Vis detaljer Last ned

Utgående faktura: 1840323 Vis detaljer Last ned

Utgående faktura: 1816005 Vis detaljer Last ned

Figur 5-8: Side for faktura som ikke er betalt

5.3.3 Saker

En bruker som har tilgang til siden for saker, vil se en liste over alle åpne og avsluttet sakene som en bedrift har hos Eninvest. Her vil det også være mulig å opprette en ny sak, hvor brukeren må velge produktet eller tjenesten det gjelder, temaet, og det må skrives en melding for at saken skal opprettes. Når bruken går inn på en sak så vil det vises ulik informasjon om saken slik som saksnummer, dato for opprettelse, status, ansvarlig avdeling og person. På denne siden, vist i Figur 5-9, vil meldingsboken vises, hvor en bruker kan sende meldinger direkte til Eninvest, med mulighet til å se sendte eposter og tekstmeldinger i dedikerte tabs for den gitte saken.

The screenshot shows the 'Faktura' (Invoice) page in the Eninvest system. The top navigation bar includes 'Heim', 'Saker', 'Linjer', 'Faktura', 'Kontaktpersonar', and 'Tilgangsstyring'. There are also buttons for 'Test Bedrift', 'Truls Haukeland', and 'Logg ut'. The main content area is titled 'Faktura' and displays the following details:

Saksnummer:	734
Oppretta:	26.04.2024, kl. 15:22:16
Status:	Open - Ny
Ansvarlig avdeling:	Kundesenteret
Ansvarlig:	Ikkje tildelt

Below the details, there are tabs for 'Melding', 'Epost', and 'SMS'. The 'Melding' tab is active, showing a message box with the following content:

26.04.2024, kl. 15:22:16

test

26.04.2024, kl. 15:22:16

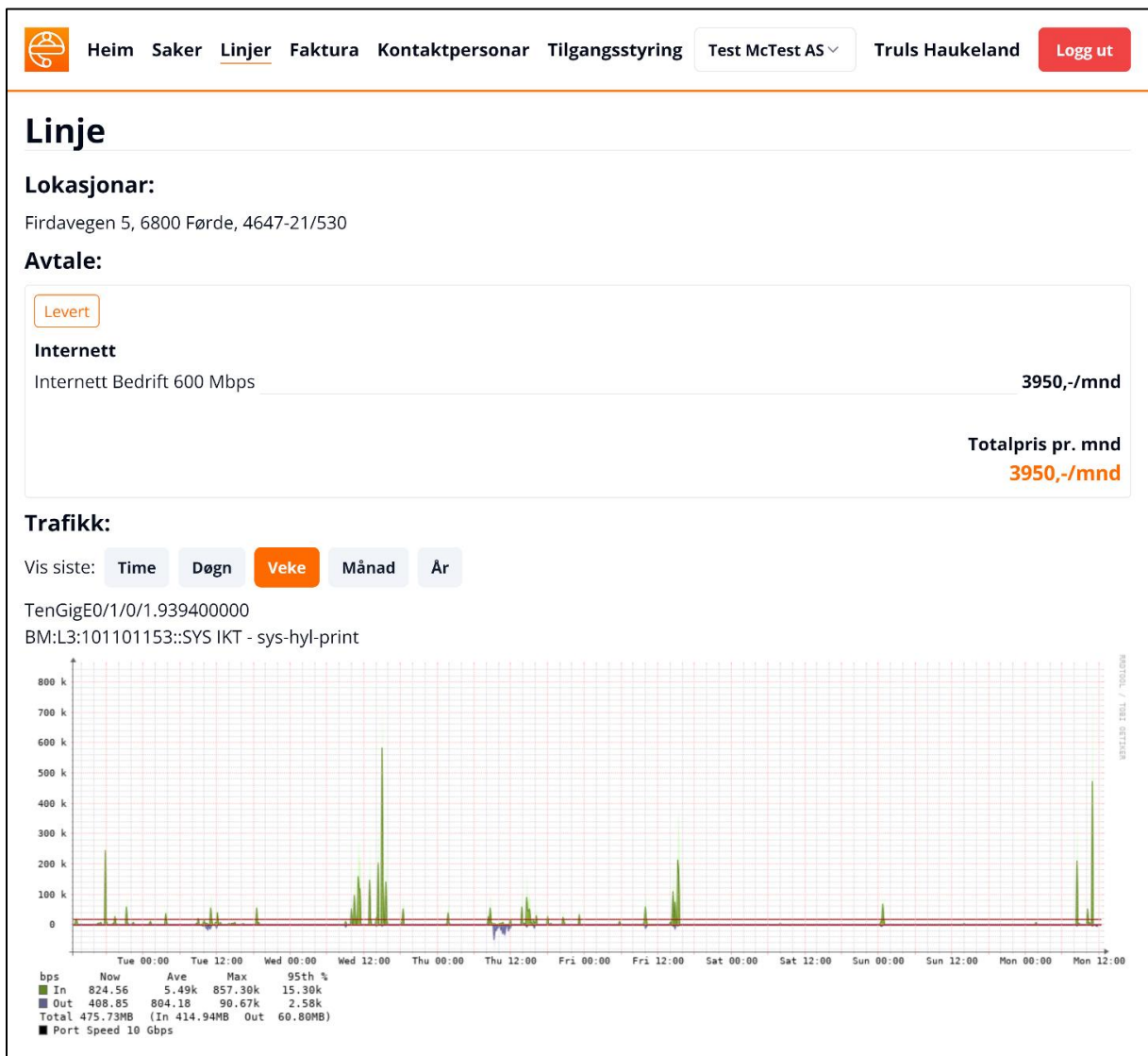
Hei!
Takk for di henvending. Vi har registrert den med følgende saksnummer: 734.
Ein av våre kundefrågjevarar kjem tilbake til deg så snart som mogeleg, og du kan forvente svar innan kort tid. Du vil bli varsla via e-post og/eller app når det er nye meldingar i saka di.
Med venleg helsing
Eninvest

At the bottom right of the message box, there is a 'Send' button.

Figur 5-9: Side for en valgt sak

5.3.4 Linjer

På siden for linjer vil en bruker som har tilgang få en oversikt over de ulike linjene kunden har, på ulike lokasjoner. Ved å velge en av disse linjene vil det vises en oversikt over avtalen den gitte linjen hører til med informasjon om status på leveransen av linjen, hastigheten til linjen og prisen på abonnementet (Figur 5-10). Her vil det også vises grafer over trafikken som har vært på linjen, hvor det er mulig å filtrere på time, dag, uke, måned og år.



Figur 5-10: Side for en valgt linje

5.3.5 Kontaktperson

Siden for kontaktpersoner gir muligheten for at brukeren kan se og endre kontaktpersoner knyttet til en bedrift. En kontaktperson må være registrert med et telefonnummer og en epost for varslinger. Det vil også være mulig å legge til hvilken avdeling personen hører til og vil være nyttig informasjon for å skille ulike kontaktpersoner.

5.3.6 Hjemmeside og navigasjonsbar

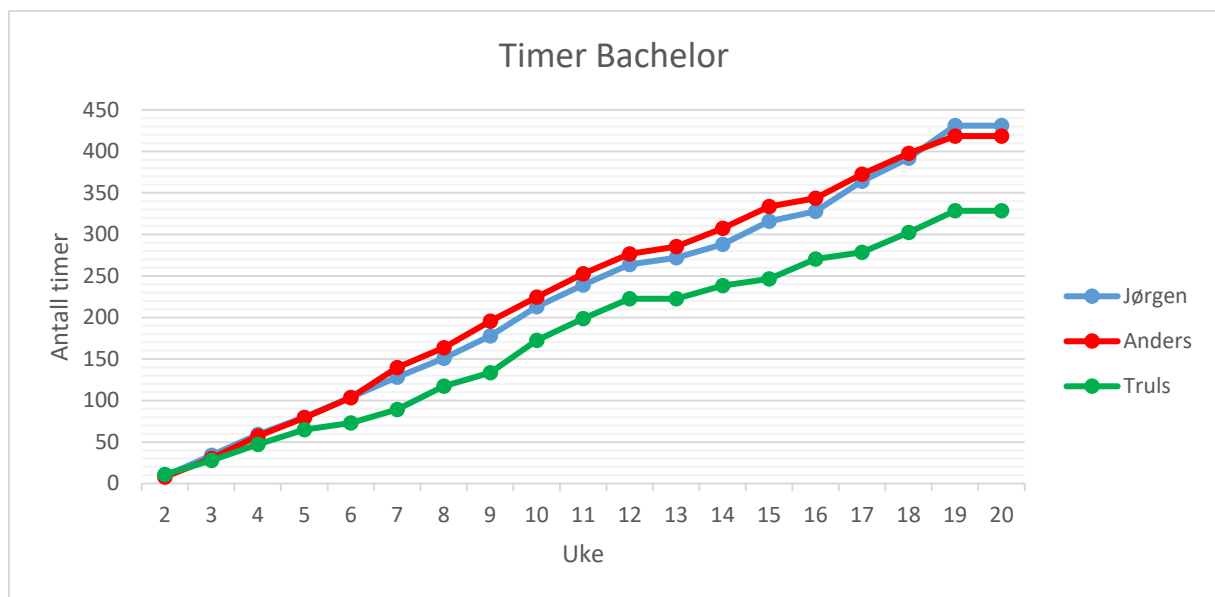
Landingssiden ved innlogging som også fungerer som hjemmesiden til løsningen var planlagt å inneholde forskjellige moduler for rask og enkel informasjon, men grunnet tid vil kun bedriftens personlige rådgiver vises her. Navigasjonslinjen på toppen av nettsiden vil være hoved metoden for navigering på nettsiden, og denne vil alltid være tilgjengelig for brukeren. De ulike sidene en bruker kan navigere til vil variere basert på hvilke sider og funksjoner den har fått tilgang til. For eksempel så vil ikke siden for tilgangsstyring vises med mindre brukeren har rollen som administrator. På navigasjonslinjen vil en bruker med tilgang til flere kunder ha muligheten til å bytte mellom disse kundene.

5.3.7 Profil

Brukere av løsningen vil også ha muligheten for å oppdatere eller endre sin egen brukerinformasjon slik som epost og telefonnummer, og endringen må verifiseres med en engangskode. Under brukerinformasjonen vil de kundene brukeren har tilgang til vises og det vil være mulig for brukeren å bytte til en annen kunde, hvis brukeren har tilgang til flere, samt fjerne koblingen til en kunde. Når brukeren blir lagt til på en bedrift vil invitasjonen vises under og brukeren må akseptere invitasjonen for å få tilgang til kunden.

5.4 Prosjektgjennomføring

For dette prosjektet stilte oppdragsgiver med alt av utstyr nødvendig for utførelse av prosjektet, samt kontorer, noe som var til stor nytte for å opprettholde kontinuerlig arbeid. En annen fordel med at Eninvest stilte med kontorer var at det var enkelt å få assistanse fra de ansatte, samt planlegging og møteinnkalling. Gantt-diagrammet blir ofte brukt og oppdateres kontinuerlig under prosjektet for planlegging og sporing av gjennomføring. Som vist i Figur 5-11 har medlemmene jobbet jevnt gjennom hele prosjektet.



Figur 5-11: Progresjon av timer jobbet i bachelor

6 DISKUSJON

6.1 Prosjektgjennomføring

Gjennom utføringen av dette prosjektet har gruppen møtt på flere små og store utfordringer. Disse er håndtert og i alt kan prosjektet sies å være en suksess. En av de første utfordringene som møtte gruppen er at to av medlemmene måtte sette seg inn i Eninvest sine systemer og det av programmeringsspråk og rammeverk som brukes der. Dette tok noe tid og starten av utviklingen gikk noe senere en ønsket. Etter hvert som gruppen fikk bedre innsikt i dette, gikk utviklingen bedre og raskere.

Gruppen hadde også til tider litt utfordringer med kommunikasjon, som lå til grunn for noen perioder med dårligere arbeidsfordeling og progresjon. Sett bort fra disse periodene hadde gruppen generelt god flyt i arbeidet. Prosjektet er stort og består av mye arbeid. Ved å prioritere den ønskede funksjonaliteten i sammen med oppdragsgiver kom det frem en liste med funksjoner som realistisk kunne utvikles innenfor prosjektperioden.

Prosjektet brukte smidig utviklingsmetodikk ved å utføre arbeidet i iterasjoner var det enkelt å tilpasse utviklingsplanen hvis en funksjon hadde behov for mer eller mindre tid til utvikling. Det iterative arbeidet sørget også for at tilbakemeldinger fra oppdragsgiver kunne enkelt bli tatt med i planlegging for fremtidige iterasjoner. Dette sørget for at en funksjon ble ferdig utviklet før arbeidet på neste ble satt i gang. Parprogrammeringen var også til stor nytte i oppstarten av prosjektet for å bli bedre kjent med systemene og teknologiene brukt hos oppdragsgiver.

6.2 Resultat

Gjennom prosjektet har det blitt utviklet en selvbetjeningsløsning for Eninvest sine bedriftskunder. Løsningen er en fullt funksjonell prototyp som legger et godt fundament for videreutvikling frem mot en endelig produksjonsklar versjon. Av de tolv funksjonene som var planlagt for løsningen ble ni ferdig implementert. Selv om oppdragsgiver hadde ønsket at alt av funksjonalitet ble utviklet, er arbeidet gruppen leverte for dette prosjektet ansett som tilstrekkelig. Det ble tidlig i prosjektet bestemt at målet var å lage et godt grunnlag for

applikasjonen med en rekke basis-funksjonalitet, og ikke en ferdig applikasjon som skulle ut til kundene. Med dette i bakgrunn stiller resultatet til oppdragsgivers forventninger.

Kravene knyttet til funksjonalitet som ble oppfylt tilhører seks forskjellige kategorier, vist i Tabell 5-1: tilgangsstyring, faktura, saker, linje, kontaktperson og generelt. En av de mest teknisk krevende delene av systemet som er utviklet er tilgangsstyringen. Det ble utforsket flere måter å gjøre dette på, før bruk av hierarkisk RBAC ble bestemt. Dette førte til en god og fleksibel løsning som gjør tilgangsstyring enkelt og øker sikkerheten til applikasjonen. Fra første gang dette ble vist til oppdragsgiver har det kommet positive tilbakemeldinger på hvordan dette kan brukes også andre steder i systemet.

Det var derimot ikke alle kravene som ble oppfylt, da det ikke var tilstrekkelig med tid for å utvikle disse. Dette er funksjonalitet knyttet opp mot oversikt over og endring av fakturering, linjestatus samt informasjon om feil og planlagt arbeid for linjer. Som nevnt i kapittel 2.4 så beskriver Chan og Petrikat (2022, s. 124) utfordringene rundt utviklingen av et system som tilfredsstillende alle kravene og fremmer effektiv tjenestelevering satt av kundene og organisasjonen. Funksjonene som ikke ble implementert ble ansett å være mindre kritiske for løsningen i sin helhet, da den vil være funksjonell uten disse. De vil derimot være til nytte for brukerne av løsningen, og kan utvikles ved en senere anledning.

Utvikling av en brukerprofil-side ble prioritert til tross for at det ikke var en del av kravene satt for applikasjonen. Dette ble gjort fordi det ble sett på som nødvendig funksjonalitet for en fungerende applikasjon. Under testingen kom det fram at det manglet en side med informasjon om kunden, i form av navn, bedriftsnummer, kontaktinfo og lignende. Dette var heller ikke en del av de originale kravene for applikasjonen, men ses på som funksjonalitet som burde ha vært med. Denne informasjonen er noe alle partene for dette prosjektet hadde oversett og av den grunn ble det ikke tatt med i funksjonalitet utviklet for endelig løsning. Det kom frem under testing at dette er en funksjon som burde ha blitt implementert.

Brukertesten utført er til stor nytte for vurdering av prosjektet opp mot vurdering av funksjonaliteten utviklet. Selv om fem deltakere ikke er så mange og det hadde vært ønskelig med flere til testen, er det til stor nytte for resultatet at de er fra forskjellige avdelinger. På denne måten er resultatet variert i forhold til ulike meninger om hva som er viktig i en applikasjon, slik som det som er utviklet for dette prosjektet. Spørreundersøkelsen utført i sammenheng med brukertesten viser at løsningen vil være til nytte for å nå målet om effektivitet og kundetilfredshet.

Under testing kom det tilbakemeldinger på at designet av applikasjonen kunne ha vært mer likt andre Eninvest-applikasjoner, som for eksempel Miside privat. For å sikre at designet av applikasjonen er mer lik andre Eninvest-applikasjoner burde oppdragsgivers designer blitt inkludert fra starten av prosjektet. På denne måten kan designet utarbeidet i wireframes passe bedre med den grafiske stilen brukt i andre applikasjoner, som Miside privat. Det ville også vært det store fordel å se mer på denne applikasjonen for inspirasjon, selv om den er mer tilpasset mobil.

6.3 Bruk av teknologi

Som nevnt i kapittel 6.1 måtte to av gruppemedlemmene sette seg inn i teknologiene som Eninvest bruker. Disse teknologiene er php, react og MySQL. Det var et av kravene satt av Eninvest at disse teknologiene skulle nyttes. Dette kommer av at det er nødvendig for integrering med eksisterende system og vil gjøre overleveringen av løsningen ved prosjektets slutt enklere. Den tekniske utformingen av løsningen følger samme strukturene som andre Eninvest-systemer av samme grunn.

Det meste av frontend-koden til Eninvest er skrevet i react med JavaScript. Gruppen tok et valg om å bruke TypeScript istedenfor JavaScript. Dette var noe gruppen ønsket og siden TypeScript er brukt noen steder i Eninvest sin kode, er det ikke noe problem for dem. I tillegg til TypeScript bruker applikasjonen noen bibliotek som ikke er brukt ellers i Eninvest. Shadcn tilbyr mange gjenbrukbare komponenter. Ved å bruke dette kan fokuset settes på å lage en fungerende applikasjon og ikke å utvikle alle disse komponentene, som er nødvendige for løsningen, fra bunnen av. For enkelte oppgaver var ikke Shadcn sine komponenter

tilstrekkelig og egne komponenter måtte utvikles, men bruk av dette er positivt for løsningen. Bruken av CSS-biblioteket tailwind gjorde utviklingen enklere for de samme grunnene som Shadcn. Valget å bruke disse bibliotekene er godkjent av Eninvest og er en god måte å teste om disse kan brukes til andre prosjekter hos dem.

7 KONKLUSJON OG VIDERE ARBEID

7.1 Konklusjon

Dette prosjektet hadde som mål å lage en selvbetjeningsløsning for Eninvest sine BM-kunder, slik som PM-kundene har i dag, som kan effektivisere kundebehandlingen og øke kundetilfredsheten. Løsningen utviklet for dette målet vil, basert på resultatene beskrevet tidligere kunne løse problemet med effektivitet og kundetilfredshet, hvis løsningen blir satt i produksjon. Som beskrevet av Chan og Petrikat (2022, s. 122) vil en løsning slik som den som er utviklet i dette prosjektet kunne gi en 45% økning i nettrafikk og en lik reduksjon av innkommende samtaler. Løsningen må derimot videreutvikles, testes av reelle brukere, og lanseres før dette kan bekreftes. Denne vurderingen vil også kreve litt tid, da kundene må få tid og mulighet til å ta den i bruk og trafikken må måles før og etter at bruken av løsningen har blitt normen for kundene.

Resultatene fra brukertesten viser at løsningen som er utviklet for dette prosjektet vil være til nytte for Eninvest sine ansatte og bedriftskunder. Alt av funksjonaliteten som ble utviklet for løsningen vil være til nytte for kundene, og vil gi de muligheten til en høyere grad av selvbetjening sammenlignet med dagens løsning. Prosjektets oppsett vil også gjøre det enkelt for oppdragsgiver å bygge videre på løsningen, ved å potensielt forbedre de delene som trenger forbedring og med nye funksjoner til løsningen. Resultatet av dette prosjektet kan også være til nytte for andre som skal lage en lignende løsning eller generelt for utvikling av nettside fra bunnen av. Dette prosjektet viser også at det er viktig å begrense omfanget etter tiden satt til utvikling av prosjektet, slik at mest mulig planlagt funksjonalitet blir utviklet for første versjon som deretter burde testes med reelle brukere.

7.2 Videre arbeid

Det første steget som burde gjøres er å fikse innmeldte feil som kom fram under brukertesten, slik at disse er håndtert før noen større endringer utføres.

7.2.1 Design

Neste steget er at design av nettsiden burde forbedres, da tilbakemeldinger fra brukertesting påpeker at dette ikke er tilstrekkelig sammenlignet med Eninvest sine andre løsninger. Kravet satt for designet av løsningen var at det skulle passe inn med de andre løsningene hos Eninvest, noe løsningen gjør, men har potensiale for grafisk forbedring slik at den står i stil med løsningen til privatkundene.

7.2.2 Lansering til kunder

Det største steget som kan gjøres i forhold til videre arbeid for dette prosjektet vil være å lansere den til kundene. For at dette skal kunne gjøres vil det derimot være nødvendig å utarbeide løsningen litt mer enn den nåværende versjonen. Siden dette prosjektet hadde fokus på å bygge en god grunnmur som kan bygges videre på, vil det være naturlig å starte med å finpusse utsende av løsningen. Når dette er gjort burde det utføres brukertesting med en håndfull reelle kunder for å få tilbakemelding på nettsiden før den slippes til alle kundene. Fra denne testingen burde det gjøres en vurdering om løsningen møter minstekravene stilt av Eninvest for å kunne bli tatt i bruk.

7.2.3 Funksjoner

I oppstarten av dette prosjektet ble det avklart med ansatte i Eninvest hvilke funksjoner som skulle prioriteres for denne versjonen av løsningen. Videre arbeid med løsningen vil være å implementere funksjonene som ikke ble prioritert. Av disse funksjonene vil administrasjon og bestilling av kundens abonnement være nyttig funksjoner for brukere av løsningen å utføre.

8 REFERANSER

- Chan, C.Y.T. og Petrikat, D. (2022) Self-Service Technology: Benefits and Challenges, *Journal of Computer Science and Technology Studies*, 4(2), s. 118-127. DOI: 10.32996/jcsts.2022.4.2.14
- Chin, S. og Older, S. (2010) *Access Control, Security, and Trust*, Boca Raton, FL: CRC Press LLC.
- Considine, E. og Cormican, K. (2017) The rise of the prosumer: an analysis of self-service technology adoption in a corporate context, *International Journal of Information Systems and Project Management*, 5(2), s. 25-39. DOI: 10.12821/ijispm050202
- Discord (u.å.) *Create space for everyone to find belonging*. Tilgjengelig fra: <https://discord.com/company> (Hentet: 19. februar 2024).
- Dixon, M., Ponomareff, L., Turner, S. og Delisi, R. (2017) Kick-Ass Customer Service, *Harvard Business Review*, januar-februar. Tilgjengelig fra: <https://hbr.org/2017/01/kick-ass-customer-service> (Hentet 23. februar 2024).
- Docker (u.å.). *Accelerate how you build, share, and run applications*. Tilgjengelig fra: <https://www.docker.com/> (Hentet: 11. april 2024)
- Eninvest (u.å.) *Om oss*. Tilgjengelig fra <https://www.eninvest.no/om-oss> (Hentet: 5. februar 2024).
- Eninvest AS (u.å.) *Om oss*. Tilgjengelig fra: <https://no.linkedin.com/company/eninvest-as> (Hentet: 29. januar 2024).
- Figma (u.å.) *How you design, align, and build matters. Do it together with Figma*. Tilgjengelig fra: <https://www.figma.com/> (Hentet: 19. februar 2024).
- Geeksforgeeks (2023). *Pair Programming*. Tilgjengelig fra: <https://www.geeksforgeeks.org/pair-programming/> (Hentet: 16. april 2024)
- Git (u.å.) *Local-branching-on-the-cheap*. Tilgjengelig fra: <https://git-scm.com/> (Hentet: 19. februar 2024).
- Github (u.å.) *Let's build from here*. Tilgjengelig fra: <https://github.com/about> (Hentet: 19. februar 2024).

Lucid (u.å.) *We can help you visualize, plan and build your business.* Tilgjengelig fra: <https://lucid.co/about> (Hentet: 19. februar 2024).

Microsoft (2017) *State of global customer service.* Tilgjengelig fra: <http://info.microsoft.com/rs/157-GQE-382/images/EN-CNTNT-Report-DynService-2017-global-state-customer-service-en-au.pdf> (Hentet: 23. februar 2024).

Microsoft (u.å.) *What is Microsoft 365?* Tilgjengelig fra: <https://support.microsoft.com/en-us/office/what-is-microsoft-365-847caf12-2589-452c-8aca-1c009797678b> (Hentet: 19. februar 2024).

Monday (u.å.) *Introduction to monday.com.* Tilgjengelig fra: <https://support.monday.com/hc/en-us/articles/115005310945-Introduction-to-monday-com> (Hentet: 16. april 2024).

NGINX (u.å.) *What is NGINX?.* Tilgjengelig fra: <https://www.nginx.com/resources/glossary/nginx/> (Hentet: 12. april 2024).

Oracle (u.å.) *What is MySQL?.* Tilgjengelig fra: <https://www.oracle.com/mysql/what-is-mysql/> (Hentet: 19. februar 2024).

PHP (u.å.) *PHP.* Tilgjengelig fra: <https://www.php.net/> (Hentet: 19. februar 2024).

React (u.å.) *React.* Tilgjengelig fra: <https://react.dev/> (Hentet: 19. februar 2024).

Sandhu, S., Ferraiolo, D. og Kuhn, R. (2000) The NIST model for role-based access control: towards a unified standard, *ACM workshop on Role-based access control*, s. 47-63.
DOI: [10.1145/344287.344301](https://doi.org/10.1145/344287.344301)

Shadcn (u.å.) *Build your component library.* Tilgjengelig fra: <https://ui.shadcn.com/> (Hentet: 19. februar 2024).

TypeScript (u.å.) *TypeScript is JavaScript with syntax for types.* Tilgjengelig fra: <https://www.typescriptlang.org/> (Hentet: 19. februar 2024).

Visual Studio Code (u.å.) *Getting Started.* Tilgjengelig fra: <https://code.visualstudio.com/docs> (Hentet: 19. februar 2024).

9 VEDLEGG

Vedlegg I: Visjonsdokument

Vedlegg II: Prosjekthåndbok

Vedlegg III: Kravdokument

Vedlegg IV: Systemdokumentasjon

Vedlegg V: Skjermbilder Endelig Løsning

Vedlegg VI: Brukertesting