

Sindre Glomnes, Emil Glückstad

Applikasjon for endringskontroll på servere

Bacheloroppgave i Dataingeniør

Veileder: Jonathan Jørgensen

Mai 2023



Sindre Glomnes, Emil Glückstad

Applikasjon for endringskontroll på servere

Bacheloroppgave i Dataingeniør
Veileder: Jonathan Jørgensen
Mai 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden

Sammendrag

Denne bacheloroppgaven omhandler utviklingen av en webapplikasjon for å standardisere prosessen for serverendringer. Applikasjonens hovedmål er å tilby en strukturert metode for brukere å gjennomføre endringer på servere ved å fylle ut dokumentasjon med nødvendig informasjon. Dokumentasjonen må deretter godkjennes av en annen bruker før endringene implementeres. Applikasjonens brukergrensesnitt er utformet som en produktkø for å gi en tydelig oversikt over prosessen.

Under utviklingen av applikasjonen ble det lagt vekt på å skape et brukervennlig grensesnitt. Dette innebærer å følge prinsipper brukervennlighet og andre relevante retningslinjer. En brukertest ble utført, der deltakerne ble bedt om å svare på ti spørsmål fra System Usability Scale, et evalueringsverktøy for å sammenligne brukervennligheten med andre applikasjoner.

Resultatet av prosjektet er en webapplikasjon skreddersydd for Geomatikk sine interne systemer. Gjennom jevnlig møter med oppdragsgiver, ble det oppnådd god kommunikasjon og forståelse av applikasjonens utvikling og formål.

Abstract

This bachelor thesis concerns the development of a web application to standardize the process for server changes. The primary goal of the application is to provide a structured method for users to carry out changes on servers by filling out documentation with the necessary information. The documentation must then be approved by another user before the changes are implemented. The application's user interface is designed as a backlog to provide a clear overview of the process.

During the development of the application, emphasis was placed on creating a user-friendly interface. This involves following principles of universal design and other relevant guidelines. A user test was conducted, in which participants were asked to answer ten questions from the System Usability Scale, an evaluation tool for comparing the usability of the application with other applications.

The result of the project is a web application tailored to Geomatikk's internal systems. Through regular meetings with the client, good communication and understanding of the application's development and purpose were achieved.

Forord

Denne rapporten er skrevet i forbindelse med det avsluttende emnet IDATT2900 Bacheloroppgave for studieretningen Dataingeniør ved Institutt for datateknologi og informatikk ved NTNU Trondheim.

Oppgaven er gjennomført med Geomatikk AS som oppdragsgiver. Vi kom i kontakt med Geomatikk om gjennomføring av bacheloroppgave gjennom grupped medlem Sindre Glomnes som har vært ansatt i bedriften siden høsten 2022. Oppgaven virket spennende og interessant særlig grunnet dens form som en fullstack utviklingsoppgave.

Under arbeidet med oppgaven har vi vært i kontinuerlig kontakt med oppdragsgiver hvor oppgaven har blitt spesifisert og krav og ønsker er formidlet for å til slutt kunne ende opp med et tilfredstillende produkt.

Vi ønsker å takke Geomatikk for oppgaven og gode rammer for utviklingen av produktet. Det rettes også stor takk til de ansatte ved Geomatikk som har gjennomført brukertester og gitt gode tilbakemeldinger på produktet. Vi ønsker også å rette en spesiell takk til Magnus Grøtan for tildeling av oppgaven, Torbjørn Danielsen som har formet oppgaven og representert oppdragsgiver og deres krav og Tor Andreas Aasbeø i rollen som vår veileder og fremste kontaktperson hos Geomatikk. Til slutt takker vi veileder ved NTNU Jonathan Jørgensen som har bidratt med gode råd og tilbakemeldinger angående de formelle kravene i bacheloroppgaven.

Trondheim, 16.05.2023



Emil Glückstad



Sindre Glomnes

Oppgavetekst

Geomatikk AS ønsket seg et verktøy for å kunne registrere endringer på servere og lagre disse i en serverhistorikk.

Den opprinnelige utformede oppgaven omhandlet dokumentering og registrering av serverendringer i tillegg til å håndtere ulike brukertilganger. Fra starten av prosjektet og igjennom hele utviklingsfasen ble det fremmet og endret krav til det endelige produktet. Kravene er presentert i visjonsdokument [Vedlegg B] og utfylt i form av brukerhistorier i kravdokumentasjonen [Vedlegg C].

Den største endringen fra opprinnelig oppgavetekst omhandler håndtering av ulike brukertilganger. Denne delen finnes fortsatt i noen grad i det endelige produktet i form av at det skilles på brukere og administratorer, men det er vesentlige endringer fra det som opprinnelig var beskrevet. I oppgaveteksten stod følgende om brukertilganger:

«Håndtere ulike typer tilganger. Enkelte skal kunne se og endre på alle servere, men andre kun skal kunne endre på servere i sone 1 og enkelte skal bare ha leserettigheter i systemet.»

Dette ble endret allerede før utviklingsfasen var påbegynt og ble erstattet med ett system rundt forespørsler om endringer. Det vil si at en hvilken som helst bruker skal kunne opprette en forespørsel om en serverendring som en annen bruker må gå igjennom og godkjenne før endringen kan registreres og bli synlig i serverhistorikken.

Innhold

Sammendrag	1
Abstract	2
Forord	3
Oppgavetekst	4
Innhold	5
Figurer	8
Tabeller	8
Terminologier	9
Forkortelser og Akronymer	9
Begreper	9
1 Introduksjon	10
1.1 Bakgrunn	10
1.2 Problemstilling	10
1.3 Rapportens struktur	11
2 Teori og relevant litteratur	12
2.1 Definisjon av brukervennlighet	12
2.2 Prinsipper for brukervennlighet	12
2.3 Universell utforming og tilgjengelighet	13
2.4 Brukeropplevelse (UX) og brukergrensesnitt (UI)	15
2.5 Måling av brukervennlighet	15
3 Metode	17
3.1 Forskningsmetode	17
3.1.1 Kundemøter	17
3.1.2 Ukentlige sammendrag	17
3.1.3 Brukertester	17
3.2 Utviklingsmetode	18
3.2.1 Wireframe prototype	18
3.2.2 Gitlab	18
3.2.3 Prioritetstabell	19
3.3 Valg av teknologi	19
3.3.1 Vue.js	19
3.3.2 TypeScript	20
3.3.3 ASP.NET Core	20
3.3.4 PostgreSQL	20
3.3.5 Azure	21
3.3.6 Vuetify	21

3.4	Rollefordeling	21
4	Resultater	23
4.1	Vitenskapelige resultater	23
4.1.1	Webapplikasjon	23
4.1.2	Brukervennlighet	29
4.1.3	Brukertest	33
4.2	Ingeniørfaglige resultater	35
4.2.1	Funksjonelle krav	35
4.2.2	Ikke funksjonelle krav	36
4.3	Administrative resultater	37
4.3.1	Utviklingsmetode	37
4.3.2	Fremdriftsplan	38
4.3.3	Timeregnskap	38
5	Diskusjon	39
5.1	Diskusjon av vitenskapelige resultater	39
5.1.1	Webapplikasjon	39
5.1.2	Brukervennlighet	40
5.1.3	Brukertest	42
5.1.4	Oppsummering	44
5.2	Diskusjon av ingeniørfaglige resultater	44
5.2.1	Funksjonelle krav	44
5.2.2	Ikke funksjonelle krav	45
5.3	Diskusjon av administrative resultater	45
5.3.1	Utviklingsmetode	45
5.3.2	Fremdriftsplan	46
5.3.3	Timeregnskap	46
5.3.4	Gruppearbeidet	47
6	Konklusjon og videre arbeid	48
6.1	Konklusjon	48
6.2	Begrensninger	48
6.3	Videre arbeid	49
	Samfunnspåvirkning	51
	Profesjonsetiske problemstillinger	51
	Bærekraftsvurderinger	51
	Referanser	54
	Vedlegg	55

Figurer

1	Utsnitt av Dashboard	24
2	Utsnitt 1 av en serverendring	24
3	Utsnitt 2 av en serverendring	25
4	Utsnitt 3 av en serverendring	26
5	Utsnitt av oppgaveside	26
6	Utsnitt av ny endring	27
7	Utsnitt av server oversikt	28
8	Utsnitt av administratorside	28
9	Håndtering av spørring og tilbakemelding	29
10	Fargekode og ikoner ved sending/sletting av endringsforespørsel	29
11	Validering	30
12	Meny i venstre marg	31
13	Listeelement av endringsforespørsel	32
14	Feilmelding	32
15	Uautorisert	33
16	Fordeling av timer per uke	38
17	Brukertest	43
18	Oppgaver satt til fullført	43

Tabeller

1	Aktivitetsoppgaver	34
2	System Usability Scale	34
3	Funksjonelle krav	36

4	Ikke funksjonelle krav	37
---	----------------------------------	----

Terminologier

Forkortelser og Akronymer

API - Application Programming Interface, programmeringsgrensesnitt.

AS - Aksjeselskap.

CI - Continuous Integration, kontinuerlig integrasjon.

IP - Internett Protocol, internett protokoll.

MVC - Model-View-Controller, model-vising-kontroller.

MVP - Minimal Viable Product, enkleste brukbare produkt.

OOP - Object-Oriented Programming, objektorientert programmering.

OS - Operating System, operativsystem.

SUS - System Usability Scale, systembrukbarhetsskala.

UI - User Interface, brukergrensesnitt.

UX - User Experience, brukeropplevelse.

VLAN - Virtual Local Area Network, virtuelt lokalnett.

Begreper

Brukerhistorie - er enkle beskrivelser av funksjonalitet ønsket i ett system satt i brukers perspektiv [19].

Fullstack - beskriver en prosess der det arbeides med både front-end og back-end i et system [2].

Gannt diagram - er et diagram for å illustrere aktiviteter satt opp i mot tidsbruk [5].

Produktkø - viser til en prioritert oversikt over funksjonalitet som skal være i det endelige systemet eller oppgaver som skal utføres i en bestemt sammenheng.[13].

Scrum - er et rammeverk for smidig utvikling hvor arbeidet deles inn i perioder såkalt sprinter [15].

Wireframe - en konseptskisse av en nettside eller applikasjon [7].

1 Introduksjon

Denne bacheloroppgaven er utført med Geomatikk AS som oppdragsgiver. Oppgaven har i korte trekk handlet om å utvikle et internt verktøy i form av en webapplikasjon for å skape oversikt over serverendringer. På skrivende tidspunkt er det påbegynt arbeid for å implementere systemet i samarbeid med oppdragsgiver.

1.1 Bakgrunn

Geomatikk AS er en landsdekkende aktør innenfor gravmeldingstjeneste i Norge, Sverige og Finland. Dette er en samfunnsviktig tjeneste som arbeider for å redusere antall graveskader på viktig infrastruktur. For å kunne oppnå dette omfatter tjenesten lagring av sensitiv informasjon i form av nøyaktig lokalisering av diverse infrastruktur. Tjenesten stiller derfor høye krav til oppetid og informasjonsikkerhet. Dette er krav som stadig har økt og ser ut til å øke videre i takt med endringen av verdensbilde. PST[14] skriver blant annet følgende i sin trusselvurdering for 2023. «I inneværende år forventer vi at russisk etterretningstjenester vil ha nye politiske og militære informasjonsbehov knyttet til konsekvensene av en NATO-utvidelse i Norden».

I Geomatikk AS sine systemer inngår det mange titalls servere. For å arbeide mot de høye kravene til sikkerhet og oppetid og for å profesjonalisere selskapet videre ønsker selskapet et internt verktøy for å enkelt kunne dokumentere og holde oversikt over alle endringer som gjøres på hver enkelt av disse serverne. En serverendring kan foreksempel være en oppdatering eller konfigurasjon av maskinvare, installasjon av ny programvare, endringer i nettverkskonfigurasjon eller brukeradministrasjon. Alle disse endringene kan ha innvirkning på serverens ytelse og funksjonalitet, og det er derfor viktig å dokumentere dem på en strukturert og konsistent måte. Ved å dokumentere enhver serverendring vil feilsøking effektiviseres da en raskt kan finne alle endringer som har påvirket en gitt server og omfattende informasjon tilhørende hver enkelt av disse endringene. Dette vil kunne være med å bidra til økt oppetid og sikkerhet.

1.2 Problemstilling

Utifra det som er beskrevet over har Geomatikk AS behov for et system som forenkler og effektiviserer arbeidsdagen til deres ansatte med ansvar for forvaltning av servere. Med tanke på dette er følgende problemstilling formulert:

«Hvordan loggføre prosessen forbundet med endringer på servere på en brukervennlig måte?»

Den første delen av problemstillingen viser til systemet som skal utvikles. Et system som dokumenterer hele prosessen forbundet med serverendringer. Det vil si fra en bruker

oppretter en forespørsel om en endring til endringen blir markert fullført.

Den siste delen viser til viktigheten av at systemet er brukervennlig og i dette tilfellet tilpasset Geomatikk AS og deres ansatte.

1.3 Rapportens struktur

Rapportens hovedkapitler med tilhørende beskrivelse av innhold for hvert kapittel følger under.

Kapittel 1 - Introduksjon inneholder bakgrunnen for oppgaven og problemstillingen som er arbeidet med.

Kapittel 2 - Teori og relevant litteratur inneholder relevant bakgrunnsteori og litteratur for oppgaven.

Kapittel 3 - Metode inneholder hvordan det er arbeidet med oppgaven og de valgene som er tatt under dette arbeidet.

Kapittel 4 - Resultater inneholder de vitenskapelige, ingeniørfaglige og administrative resultatene fra arbeidet med oppgaven.

Kapittel 5 - Diskusjon inneholder drøfting og refleksjon rundt resultatene fra kapittel 4 og hvordan gruppearbeidet har fungert.

Kapittel 6 - Konklusjon og videre arbeid inneholder konklusjoner som kan trekkes gitt diskusjonen fra kapittel 5 og deler av programmet det er naturlig å se på ved videre arbeid.

2 Teori og relevant litteratur

I dette kapitlet utforskes teorien og konseptene rundt brukervennlighet. Først defineres brukervennlighet og relaterte begreper, inkludert brukeropplevelse og brukergrensesnitt, samt deres innflytelse på den generelle brukervennligheten.

Videre vurderes betydningen av å identifisere målgruppen og å forstå brukerbehovene for å sikre at teknologiske løsninger er tilpasset og relevante for de tiltenkte brukerne. Universell utforming og tilgjengelighet diskuteres også som sentrale prinsipper for å skape inkluderende og tilgjengelige produkt for alle brukere.

Etter å ha etablert en god forståelse av definisjonen og begrepene rundt brukervennlighet, fokuseres det på metoder for hvordan dette kan evalueres. Dette kapitlet fungerer som et grunnlag for videre diskusjon om hvordan man best kan designe, implementere og evaluere teknologiske løsninger med hensyn til brukervennlighet, med fokus på brukertesting som en viktig metode for evaluering.

2.1 Definisjon av brukervennlighet

Brukervennlighet[10] kan defineres som en attributt på hvor enkelt et brukergrensesnitt er å benytte. Dette prinsippet omhandler lett tilgang og bruk av et nettsted. Et designs anvendelighet er ikke absolutt, men avhenger av funksjonene det inneholder samt brukerens situasjon (hva brukeren ønsker å oppnå og omgivelsene de befinner seg i), som sammen bestemmer hvor brukervennlig det er.

2.2 Prinsipper for brukervennlighet

De grunnleggende prinsippene for god brukervennlighet er avgjørende for å skape en vellykket og engasjerende brukeropplevelse. Disse prinsippene, også kjent som Nielsen's 10 Usability Heuristics[8], er et sett med retningslinjer. Disse kan brukes som et rammeverk for å evaluere og forbedre brukervennligheten i applikasjoner og nettsider. De ble først introdusert av Jakob Nielsen i 1994 og har blitt anerkjent som et sentralt verktøy innen brukervennlighetsdesign. De ti prinsippene og eksempler på implementering av disse i design og utvikling av applikasjoner er:

1. **Synlighet av systemstatus:** Systemet bør alltid holde brukeren informert om hva som skjer, gjennom tilbakemelding i løpet av rimelig tid.
2. **Tilpasning og bruk av standarder:** Brukergrensesnittet bør være konsistent og følge kjente standarder, slik at brukeren umiddelbart forstår hvordan det fungerer. For eksempel kan en applikasjon bruke ikoner og fargekoder som er vanlig i bransjen.

3. **Forebygging av feil:** Designet bør minimere muligheten for feil og hjelpe brukeren med å gjenkjenne, diagnostisere og rette opp i feil. For eksempel kan en applikasjon inkludere validering av inndata og gi forklarende feilmeldinger.
4. **Gjenkjenning framfor husking:** Brukeren skal ikke huske informasjon fra én del av grensesnittet til en annen. Ulike elementer som ikoner, etiketter for datafelt, samt lett tilgjengelige menyer hjelper brukeren med å gjenkjenne deler av nettsiden fremfor å huske dem.
5. **Fleksibilitet og effektivitet i bruk:** Systemet bør kunne tilpasses erfarne brukere som ønsker å arbeide raskere, samtidig som det er enkelt for nye brukere. Eksempelvis, kan en applikasjon tilby snarveier og tilpasse seg brukerens preferanser.
6. **Estetikk og minimalistisk design:** Grensesnittet bør være oversiktlig og enkelt, med fokus på nødvendig informasjon og funksjoner. For eksempel kan en applikasjon eliminere unødvendige elementer og bruke et hierarki av informasjon for å hjelpe brukeren med å navigere.
7. **Hjelp brukeren med å forstå og rette opp i feil:** Når feil oppstår, bør systemet gi tydelige og godt beskrevne feilmeldinger og veiledning for å løse problemet. Eksempelvis, kan en applikasjon gi tips om hvordan man kan løse en feil, samt tilby en «angre»-funksjon.
8. **Hjelp og dokumentasjon:** Selv om systemet skal være enkelt å bruke uten hjelp, bør tilgang til hjelp og nødvendig dokumentasjon være tilgjengelig. For eksempel kan en applikasjon ha en innebygd hjelpefunksjon og et søkbart dokumentasjonsbibliotek.
9. **Konsistens og standarder:** Brukergrensesnittet bør være konsistent i design, terminologi og interaksjon på tvers av hele applikasjonen. Det bør i tillegg følge etablerte konvensjoner og bransjestandarder. For eksempel kan en applikasjon benytte samme type knapper og navigasjonselementer gjennom hele applikasjonen, samt følge plattformspesifikke designretningslinjer.
10. **Brukerkontroll og frihet:** Brukerne bør kunne angre og gjenta handlinger enkelt og raskt, slik at de får en følelse av kontroll over systemet. Eksempelvis, kan en applikasjon tilby «angre»- og «gjenta»-knapper, slik at brukeren enkelt kan rette opp i feil eller endre valg uten å starte prosessen på nytt.

2.3 Universell utforming og tilgjengelighet

Universell utforming[11] og tilgjengelighet er to sentrale konsepter innen utvikling. Disse sikrer at produkter, miljøer og kommunikasjon er tilgjengelige og enkle å bruke for et bredt spekter av mennesker med ulike evner. De syv prinsippene for universell utforming ble utviklet i 1997 av en arbeidsgruppe bestående av arkitekter, produktdesignere, ingeniører og miljødesignforskere ved North Carolina State University. Disse prinsippene fungerer

som en veiledning for designprosessen og kan brukes til å evaluere eksisterende design og utdanne både designere og forbrukere. De syv prinsippene er:

1. **Likeverdig bruk:** Nettsiden skal være nyttig og attraktiv for personer med ulike evner. Dette innebærer å tilby samme funksjonalitet for alle brukere og unngå segregasjon eller stigmatisering. Sikkerhet og personvern skal også være tilgjengelig for alle brukere.
2. **Fleksibilitet i bruk:** Designet skal være mulig å tilpasse et bredt spekter av individuelle preferanser og evner. Dette kan inkludere valg av metoder for bruk, tilpasning til høyre- eller venstrehendte brukere, samt tilpasning til brukerens tempo.
3. **Enkel og intuitiv bruk:** Bruken av nettsiden eller applikasjonen skal være enkel å forstå, uavhengig av brukerens erfaring, kunnskap, språkferdigheter eller konsentrasjonsnivå. For å oppnå dette, bør designet eliminere unødvendig kompleksitet og være konsistent med brukerforventninger og intuisjon.
4. **Merkbar informasjon:** Designet skal kommunisere nødvendig informasjon effektivt til brukeren, uavhengig av omgivelsesforhold eller brukerens sensoriske evner. Dette kan oppnås ved å blant annet bruke kontrasterende farger, bilder og ikoner. Dette gjøres for å presentere essensiell informasjon, gi tilstrekkelig kontrast mellom viktig informasjon og omgivelsene, samt maksimere lesbarheten av essensiell informasjon.
5. **Toleranse for feil:** Designet skal redusere farer for utilsiktede handlinger. Dette innebærer å gi advarsler om farer og feil, samt tilby feilsikre funksjoner.
6. **Lav fysisk anstrengelse:** Nettsiden eller applikasjonen skal være mulig å bruke effektivt og komfortabelt med minimal tretthet. For eksempel kan man implementere tastaturnavigasjon og snarveier for å redusere behovet for musbruk og redusere gjentatte handlinger.
7. **Størrelse og plass for tilnærming og bruk:** Passende størrelse og plass bør være tilgjengelig for interaksjon og bruk, uavhengig av brukerens kroppsstørrelse, holdning eller mobilitet. Dette kan innebære å gi en klar synslinje til viktige elementer, imøtekomme variasjoner i hånd- og grepstørrelse, samt gi tilstrekkelig plass til bruk av hjelpemidler eller personlig assistanse.

Ved å følge de syv prinsippene for universell utforming i utvikling av nettsider og applikasjoner, sikrer man at disse er inkluderende, tilgjengelig og enkle i bruk for et bredt spekter av brukere. Dette bidrar til å skape en trygg og brukervennlig opplevelse for alle, uavhengig av deres evner eller forutsetninger.

2.4 Brukeropplevelse (UX) og brukergrensesnitt (UI)

En artikkel fra UX design institute fremhever de ulike aspektene av brukeropplevelse og brukergrensesnitt[18]. Brukeropplevelse (UX) og brukergrensesnitt (UI) er to relaterte, men distinkte begreper innen design og utvikling av digitale produkter. UX refererer til følelsene som en bruker får når man interagerer med et produkt eller en tjeneste. Med andre ord omhandler dette brukervennlighet, tilfredshet og effektivitet. UI, derimot, refererer spesifikt til de berøringspunktene en person bruker for å engasjere seg med et digitalt produkt. Eksempelvis skjermer, knapper og andre visuelle og interaktive funksjoner.

Noen av fokusområdene i UX-design innebærer å løse brukerproblemer og skape relevante, enkle og tilgjengelige opplevelser. Det er en omfattende disiplin som dekker alle aspekter av et produkt eller en tjeneste som brukeren kommer i kontakt med, og vurderer hvordan alle disse elementene samhandler for å skape en brukervennlig helhet. UI-design, på den annen side, omhandler utformingen av digitale brukergrensesnitt, deriblant farger, typografi, knapper, rullemekanismer, animasjoner.

Mens UX og UI er ulike aspekter ved design, er de nært knyttet og påvirker hverandre. Et godt designet UI kan forbedre UX ved å gjøre det enklere for brukeren å forstå og interagere med det digitale produktet. Dette kan igjen føre til økt brukervennlighet, ettersom brukerne vil oppleve produktet som mer effektivt, tilfredsstillende og enkelt å bruke.

2.5 Måling av brukervennlighet

System Usability Scale[17] (SUS) er et raskt og pålitelig verktøy for å måle brukervennlighet. SUS består av et skjema på ti utsagn testpersonen vurderer på en skala fra helt uenig til helt enig. Disse utsagnene kan benyttes for å evaluere et bredt spekter av produkter og tjenester, inkludert maskinvare, programvare, mobile enheter, nettsteder og applikasjoner.

Fordelene ved å bruke SUS er enkel administrasjon, pålitelige resultater selv med små prøvestørrelser, samt evnen til å skille mellom brukervennlige og ikke-brukervennlige systemer. Når man benytter SUS til måling av brukervennlighet er det flere faktorer som en bør huske. Poengene en får er ikke prosentandeler, og SUS er ikke diagnostisk. I stedet brukes det til å klassifisere brukervennligheten til det testede systemet.

SUS innebærer at deltakerne vurderer systemet basert på disse ti utsagnene:

1. Jeg tror jeg ville likt å bruke dette systemet ofte.
2. Jeg syntes systemet var unødvendig komplisert.
3. Jeg tror at jeg vil trenge hjelp fra en teknisk person for å kunne bruke dette systemet.
4. Jeg synes at de ulike funksjonene i dette systemet er godt integrert.

5. Jeg syntes at det var for mye inkonsekvens i dette systemet.
6. Jeg kan forestille meg at de fleste mennesker vil lære å bruke dette systemet veldig raskt.
7. Jeg synes at systemet er veldig kronglete å bruke.
8. Jeg føler meg trygg på å bruke dette systemet uten hjelp fra en teknisk person.
9. Jeg må lære mange nye ting før jeg kan bruke dette systemet effektivt.
10. Jeg synes at dette systemet er enkelt å bruke.

Poengene konverteres deretter til en skala fra 0-100. En SUS-poengsum over 68 anses som over gjennomsnittet, mens en poengsum under dette tallet anses som under gjennomsnittet.

3 Metode

Forskningsmetode omhandler å samle og analysere data for å besvare et forskningsspørsmål eller undersøke en problemstilling. Utviklingsmetode handler om å utvikle praktiske og funksjonelle løsninger gjennom testing og prototyping. Begge metodene brukes i en rekke fagområder og disipliner. Selv om de har ulike formål, er metodene ofte sammenkoblet i arbeidet rundt utvikling av kunnskap og nye løsninger.

3.1 Forskningsmetode

3.1.1 Kundemøter

Det etableres en serie med kundemøter som finner sted annenhver uke. På disse møtene evalueres beslutninger som er tatt. Det diskuteres rundt potensielle endringer og nye ønsker fremlegges. Det utformes en detaljert agenda som fungerer som en veiledning for de involverte partene. Under møtene produseres det referater som tjener som en pålitelig kilde for senere oppfølging og evaluering av de diskuterte punktene. Referat og møteinnkallinger med tilhørende agenda er lagt ved i Prosjekthåndboken [Vedlegg E].

3.1.2 Ukentlige sammendrag

For å opprettholde en åpen kommunikasjon og holde alle involverte parter informert, sendes det ukentlige e-postoppdateringer som presenterer fremdriften som er oppnådd siden forrige oppdatering. Disse sammendragene gir kundene mulighet til å gi tilbakemelding vedrørende eventuelle uklarheter, misforståelser eller uenigheter angående de utførte arbeidsoppgavene. Dette sikrer at kundene forblir engasjerte i de valgene som tas gjennom hele utviklingsfasen. I tillegg får de mulighet til å påvirke hele prosessen og det fremtidige produktet.

3.1.3 Brukertester

Brukertesting[9] er en sentral metode innen UX-forskning som evaluerer og forbedrer et produkts brukervennlighet. Ved å observere deltakere utføre aktiviteter i webapplikasjonen, kan forskere identifisere designproblemer, avdekke forbedringsmuligheter og lære om målgruppens preferanser.

Testene kan være kvalitativ, fokusert på innsikt og funn, eller kvantitativ, som samler metriske data om brukeropplevelsen. Tester kan utføres personlig, fjernstyrt moderert eller umoderert. Gjennom regelmessig testing og iterativt design, kan utviklere skape produkter som møter brukernes behov og forbedrer deres opplevelse.

Det vil være et begrenset antall brukere som har tilgang til systemet. Applikasjonen vil derfor bli testet av åtte brukere. Brukerne vil få tildelt et Google-spørreskjema som inneholder informasjon om hvordan man får tilgang til applikasjonen, og hvordan en skal utføre brukertesten. Brukertesting inneholder følgende elementer:

1. **Fasilitator:** Fasilitatoren er ansvarlig for å utforme og sette opp spørreskjemaet for brukertestene, samt analysere og tolke resultatene etter at testene er gjennomført. Fasilitatoren vil være til stede under brukertesten for å observere eventuelle problemer.
2. **Deltaker:** Deltakerne er realistiske brukere av systemet. De utfører oppgavene uten hjelp fra fasilitatoren, og gir tilbakemelding via det tildelte Google-spørreskjemaet.
3. **Oppgaver:** Oppgavene i brukertesten er todelt. Del 1 tar for seg realistiske aktiviteter som deltakerne kan utføre i det virkelige liv. Dette er case-oppgaver som er beskrevet i Google-spørreskjema og er relatert til systemets funksjoner og brukervennlighet. Disse oppgavene dekker alle primære funksjoner i applikasjonen. Eksempel på dette er å opprette, godkjenne og fullføre en endring, samt å finne dokumentasjon i historikk. I Del 2 besvarer brukerne ti utsagn om applikasjonens brukervennlighet.

Etter at deltakerne har gjennomført brukertestene, vil dataene samles inn. Del 2 av brukertesten vil analyseres ved hjelp av SUS for å evaluere systemets brukervennlighet. Ved bruk av SUS kan gruppen målbart vurdere applikasjonen opp mot andre applikasjoner.

Ved å gjennomføre brukertester på denne måten, gir det en verdifull innsikt i systemets brukervennlighet. I tillegg muliggjør SUS identifisering av forbedringsmuligheter for å skape et produkt som møter brukernes behov og forbedrer deres opplevelse.

3.2 Utviklingsmetode

3.2.1 Wireframe prototype

Prototypen av nettsiden utvikles ved bruk av Balsamiq Wireframe, et verktøy som forenkler oppretting av grove skisser og mock-ups. Skissene baseres på krav som er presentert under kundemøter og blir benyttet til å utforske ulike layouter og funksjonaliteter. Balsamiq Wireframe er et nyttig verktøy for å visualisere utseende og funksjonalitet til en nettside, i tillegg til at man får mulighet til å kommunisere idéer for kunden. Ved å presentere prototypen for brukeren kan man sikre enighet om applikasjonens utseende.

3.2.2 Gitlab

I utvikling av nettsiden benyttes GitLab for å administrere prosjektet og arbeide med koden. GitLab tilbyr funksjoner for versjonskontroll, prosjektstyring og kontinuerlig in-

tegrasjon (CI). Disse brukes for å organisere prosjektet, holde oversikt over endringer i koden, samt sette opp automatiske bygg og tester for å forbedre kodekvalitet. For å unngå konflikter og sikre en jevn utviklingsprosess, skal hvert grupped medlem jobbe på egne grener. Grenene skal slås sammen etter hver fullført oppgave.

I tillegg benyttes GitLabs merge request-funksjon, hvor et annet grupped medlem alltid går gjennom den nye koden før den blir lagt til i hovedgrenen (main). Dette sikrer at koden som legges til i main er av høy kvalitet og fungerer som den skal. Ved å bruke GitLab på denne måten kan utviklingsteamet samarbeide effektivt og levere en høykvalitets nettside.

3.2.3 Prioritetstabell

For å sikre at applikasjonen møter kundens behov og forventninger, samarbeider utviklingsteamet tett med kundene for å identifisere de viktigste funksjonalitetene som skal inkluderes. En prioritetstabell utarbeides for å inkludere både funksjonelle og ikke-funksjonelle egenskaper som er ønsket i applikasjonen. Egenskaper vurderes etter deres relative viktighet for det ferdige produktet og kategoriseres i henhold til deres prioritet.

Egenskaper som er avgjørende for at applikasjonen skal fungere kategoriseres som «høy prioritet». Dette omfatter funksjonaliteter som er helt nødvendige for at applikasjonen skal utføre sine kjernefunksjoner. Egenskaper som er ønsket, men ikke avgjørende for at applikasjonen skal fungere plasseres under «medium prioritet». Dette inkluderer funksjonaliteter som vil gi en mer fullstendig og forbedret brukeropplevelse, samtidig som de ikke er nødvendige for at applikasjonen skal utføre sin primære funksjon. Egenskaper som ikke er avgjørende for applikasjonens hovedfunksjonalitet, men som kan øke brukeropplevelsen eller tilføre ekstra funksjonalitet plasseres under «lav prioritet» i prioritetstabellen.

3.3 Valg av teknologi

I denne delen av kapitlet vil det begrunnes valg av teknologi som benyttet i prosjektet. De teknologiene og verktøyene som vil bli diskutert er: Vue.js; TypeScript; ASP.NET Core; PostgreSQL; Azure og Vuetify.

3.3.1 Vue.js

Vue.js[20] er et JavaScript-rammeverk som tilbyr en enkel og fleksibel syntaks, samt en rask og responsiv UI-oppførsel. Dette rammeverket er valgt da det er mulig å utvikle komplekse, moderne og interaktive webapplikasjoner som er enkle å vedlikeholde og skalere. I tillegg har Vue.js et aktivt og voksende samfunn som gir et bredt utvalg av dokumentasjon, verktøy og ressurser som støtter utviklingen av komplekse webapplikasjoner.

3.3.2 TypeScript

TypeScript[16] ble valgt som en del av dette prosjektet, grunnet flere fordeler som tilbys sammenlignet med JavaScript. TypeScript tilbyr objektorientert programmering (OOP), noe som sees på som fordelaktig rundt organisering og strukturering av koden. I tillegg fører det til enklere vedlikehold og utvidelse av prosjektet over tid.

TypeScript har også et kraftig typesystem som øker kodelesbarheten og reduserer sjansen for feil ved å fange type-relaterte problemer under kompilering. Dette gir en mer stabil og pålitelig kodebase sammenlignet med JavaScript, som ikke har sterk typing. I tillegg støtter TypeScript alle moderne ES6-funksjoner og fungerer som en superset av JavaScript, noe som betyr at det er fullt kompatibelt med eksisterende JavaScript-kode. Dette gjør det enklere å integrere TypeScript i et prosjekt og dra nytte av både JavaScript og TypeScript-funksjoner.

3.3.3 ASP.NET Core

ASP.NET Core[1] er et fleksibelt og effektivt rammeverk for å bygge moderne webapplikasjoner og tjenester. Det er en åpen kildekode-løsning som støtter tverrplattformutvikling på Windows, macOS og Linux, og gir en ensartet tilnærming til å bygge både webgrensesnitt og web-APIer.

ASP.NET Core benytter Model-View-Controller (MVC) mønsteret og Razor Pages for å skape en strukturert og produktiv utviklingsprosess. Rammeverket er konstruert med tanke på ytelse og skalerbarhet, og støtter ulike hostingalternativer som Kestrel, IIS, HTTP.sys, Nginx, Apache og Docker.

En av grunnene til at ASP.NET Core benyttes i dette prosjektet er at det inkluderer støtte for side-ved-side-versjonering, noe som muliggjør flere versjoner av rammeverket på samme maskin uten konflikter. Dette gir utviklere større fleksibilitet og reduserer risikoen for problemer ved oppdatering av avhengigheter. I tillegg er ASP.NET Core designet med tanke på sky-klar, miljøbasert konfigurering og innebygd avhengighetsinjeksjon, noe som forenkler utviklingsprosessen og gir bedre kontroll over applikasjonens ressurser og tjenester.

3.3.4 PostgreSQL

PostgreSQL[12] er et kraftig, åpen kildekode objekt-relasjonelt databasesystem som bruker og utvider SQL-språket. Dette systemet er kjent for sin pålitelighet, dataintegritet, robuste funksjonssett og utvidelsesmuligheter. Noen av grunnene til at systemet benyttes er at PostgreSQL støtter en rekke datatyper og sikkerhetsfunksjoner. I tillegg til skalerbarhet for å håndtere store datamengder og mange samtidige brukere. Det tilbyr avanserte indekseringsmuligheter, transaksjoner og parallellisering av leseforespørsler. I tillegg er

PostgreSQL svært utvidbart, med støtte for lagrede funksjoner og prosedyrer, flere programmeringsspråk og en rekke utvidelser. Dette gjør PostgreSQL til et ideelt valg for mange applikasjoner og organisasjoner.

3.3.5 Azure

Azure[22] er en skyplattform som tilbyr en rekke skytjenester for databehandling, lagring og nettverk. Med sin fleksibilitet, sikkerhet og skalerbarhet, blir Azure stadig mer populær blant bedrifter og organisasjoner over hele verden. En av grunnene til at gruppen benytter Azure er dens støtte for flere programmeringsspråk, inkludert Java, Node.js og C-Sharp.

I dette prosjektet benyttes Azure for autorisering, noe som bidrar til en bedre integrasjon med kundens egne systemer. Azure muliggjør utvikling og administrering av identiteter og tilganger for applikasjoner og brukere. Dette igjen sikrer en robust og sikker løsning for autorisering og autentisering.

3.3.6 Vuetify

Vuetify[21] ble valgt som teknologien for å bygge brukergrensesnittet i dette prosjektet, hovedsakelig på grunn av dens omfattende samling av ferdiglagde komponenter og funksjoner. Vuetify er et Vue-komponentrammeverk som er utviklet for å være enkelt å lære og givende å mestre. Det tilbyr et konsistent utseende og følelse i hele applikasjonen, samtidig som det gir tilstrekkelig fleksibilitet for å tilpasses ulike bruksområder.

Ettersom Vuetify er et åpen kildekode-prosjekt, er det kostnadsfritt og lisensiert under MIT-lisensen. Vuetify er basert på Google's Material Design-spesifikasjon, noe som innebærer at hver komponent er designet med tanke på brukervennlighet og estetikk.

Vuetify tilbyr en rekke verktøy som forbedrer utviklingsopplevelsen, fra prosjektopprettning til UI-designsett. Disse verktøyene inkluderer Figma UI-komponentsett, førsteparts Vite-støtte, forhåndskonfigurerte Vue 3-applikasjoner for TypeScript og JavaScript, IntelliSense og autokomplettering for VSCode og JetBrains-produkter, og wireframe-eksempler.

3.4 Rollefordeling

Når det gjelder rollefordeling, består gruppen av to utviklere i prosjektet. Det ene gruppe-medlemmet har hovedansvar for frontend-utvikling, mens den andre har hovedansvar for backend-utvikling. Frontend-utvikleren jobber primært med Vue.js og Vuetify for å utforme og implementere brukergrensesnittet, samt integrere det med backend-applikasjonen gjennom API-kall. Backend-utvikleren fokuserer på å utvikle og vedlikeholde serverdelen av applikasjonen, inkludert databasedesign og implementasjon, API-utvikling og sikkerhet ved hjelp av ASP.NET og PostgreSQL. Selv om hver utvikler har hovedansvar for sin

respektive del av prosjektet, er det også en kontinuerlig samarbeids- og kommunikasjonsprosess mellom begge utviklerne. Dette for å sikre en vellykket integrasjon av frontend og backend, samt for å adressere eventuelle problemer eller utfordringer som oppstår underveis.

Ved å ha klare ansvarsområder og en åpen kommunikasjon mellom utviklerne, kan gruppe-medlemmene effektivt håndtere eventuelle utfordringer og levere en webapplikasjon av høy kvalitet som oppfyller satte krav og mål.

4 Resultater

I dette kapitlet presenteres resultater og observasjoner som er kommet frem i løpet av og etter utviklingen av produktet. Resultatene presenteres gjennom tre ulike aspekter: vitenskapelige, ingeniørfaglige og administrative. Dette gjøres for å få et bilde av det ferdige produktet sett fra ulike vinkler, samt økt forståelse av alle deler og prosesser i programmet. I systemdokumentasjonen lagt ved finner en informasjon som går dypere i hvordan systemet henger sammen med blant annet klassediagram, prosjektstruktur og database modell [Vedlegg D].

4.1 Vitenskapelige resultater

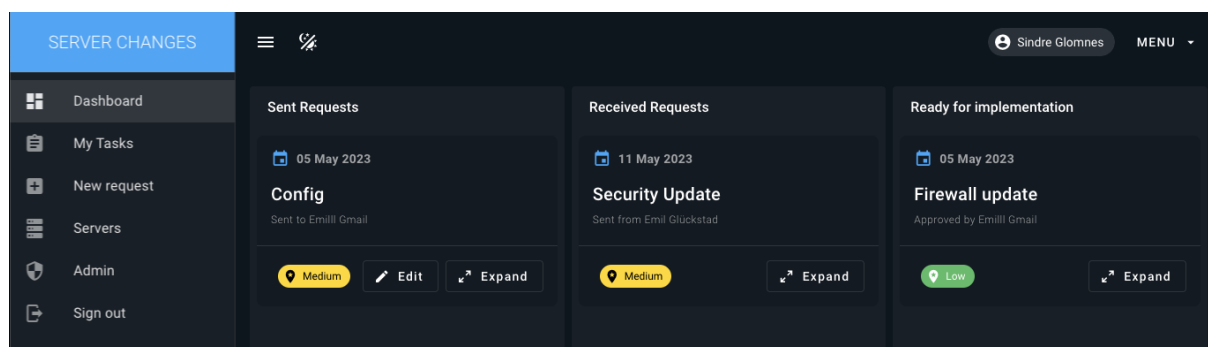
Vitenskapelige resultater tar for seg beskrivelse av produktet, designet og innhentet data. Beskrivelse av produktet og designet presenteres gjennom webapplikasjonen og brukervennlighet. For å presentere innhentet data benyttes det brukertest, som ble gjennomført etter at produktet var ferdigutviklet.

4.1.1 Webapplikasjon

Dette kapitlet tar for seg de primære sidene i webapplikasjonen og hvordan en bruker disse til å utføre handlinger. Webapplikasjonen har to fargemoduser, lys og mørk. For denne oppgaven er det valgt mørk modus.

Dashboard

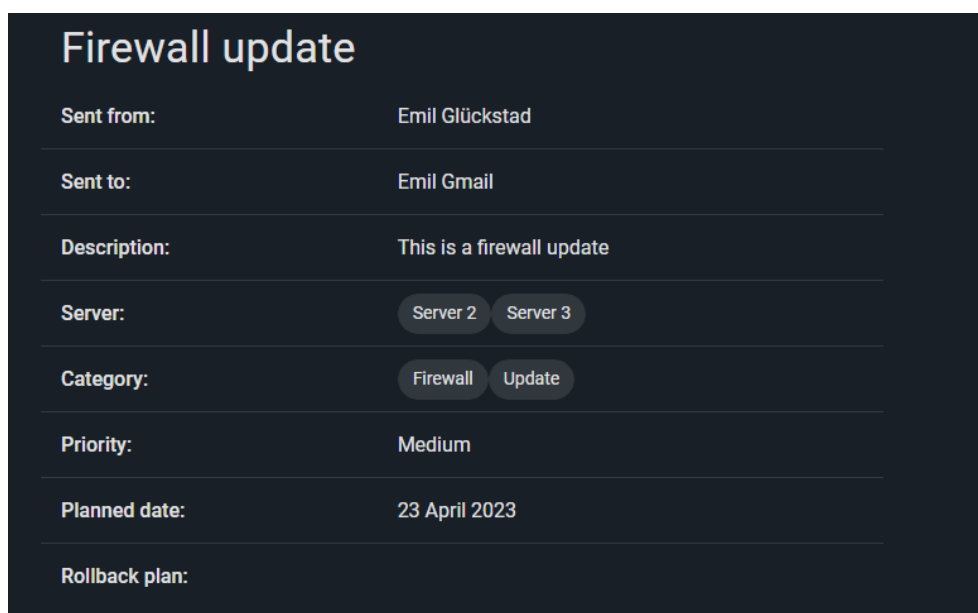
Hovedsiden i applikasjonen er Dashboard, illustrert i Figur 1. Denne gir en komplett oversikt over alle endringsforespørsler knyttet til brukeren. Disse er organisert i tre kategorier: «sendt», «mottatt» og «klar for implementering» forespørsler. Endringsforespørsler som er opprettet og sendt av brukeren, vises under «sendt», og mottatte forespørsler vises under «mottatt». Når endringsforespørsler er godkjent av mottakeren, vil de vises under «klar for implementering».



Figur 1: Utsnitt av Dashboard

Serverendring

Ved å trykke på en forespørsel eller en fullført endring i serverhistorikken, vil brukeren komme til en side hvor alt av dokumentasjon om forespørselen/endringen vises. Denne siden er illustrert i Figur 2. Dokumentasjonen inneholder informasjon om sender, mottaker, beskrivelse, tilhørende servere og kategorier, prioritet, planlagt implementasjon dato, samt tilbakestillingsplan.



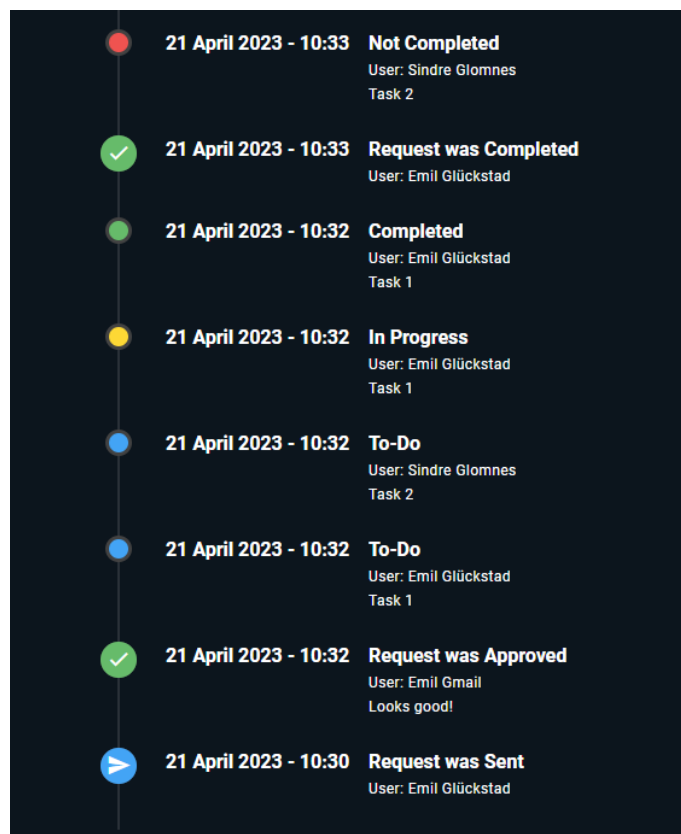
Figur 2: Utsnitt 1 av en serverendring

Serverendringen inkluderer også en risikomatrix og oppgaveoversikt, illustrert i Figur 3. Hver enkelt endring kan ha flere risikomomenter. Disse visualiseres gjennom en risikomatrix, hvor risiko er sannsynlighet multiplisert med konsekvens. En mulig risiko, kan eksempelvis, være at serveren kræsjer (Crash). Utifra matrisen ser brukeren at risikoen er i grønt område, noe som innebærer at risikograden er lav. Oppgaveoversikten som er plassert under risikomatriksen inneholder en tabell over oppgaver. Hver oppgave har en status, beskrivelse og ansvarlig bruker for implementering.



Figur 3: Utsnitt 2 av en serverendring

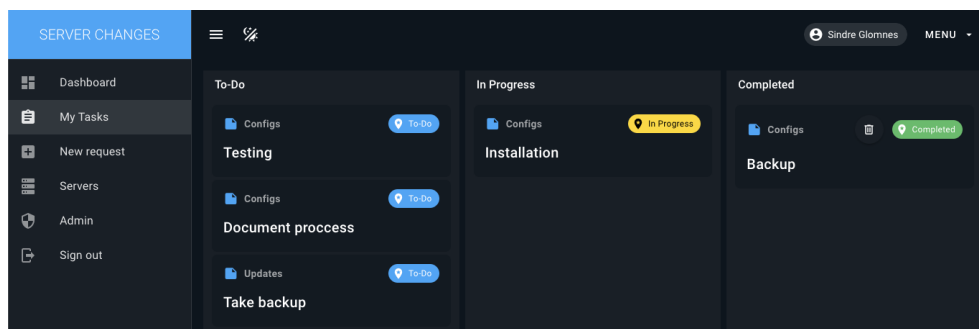
En annen del av serverendringen er historikken, vist på Figur 4. Denne inneholder informasjon om alle oppdateringer på endringen. Informasjonen viser dato, ny status, bruker og eventuelle kommentarer. Eksempel på oppdatering kan være at endringen ble godkjent, eller at en oppgave er satt til fullført.



Figur 4: Utsnitt 3 av en serverendring

Oppgaveside

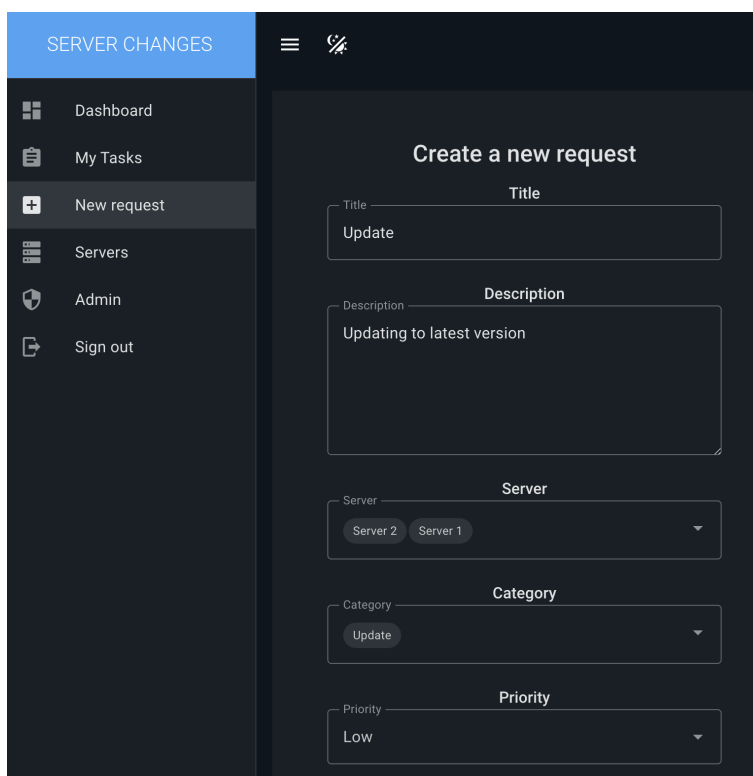
Ved å benytte menyen kan en navigere til siden for mine oppgaver, illustrert i Figur 5. Siden er satt opp som en produktkø med oppgaver som skal utføres, er under arbeid og som er fullførte. Hver oppgave inneholder informasjonen: oppgavetekst, tittel på tilhørende endring og status på oppgaven. Statusen på oppgavene kan endres ved en dra-og-slipp interaksjon. Fullførte oppgaver kan skjules ved å klikke på søppel-knappen. Hvis en endring blir markert som fullført, vil alle tilhørende oppgaver bli fjernet fra produktkøen.



Figur 5: Utsnitt av oppgaveside

Ny endring

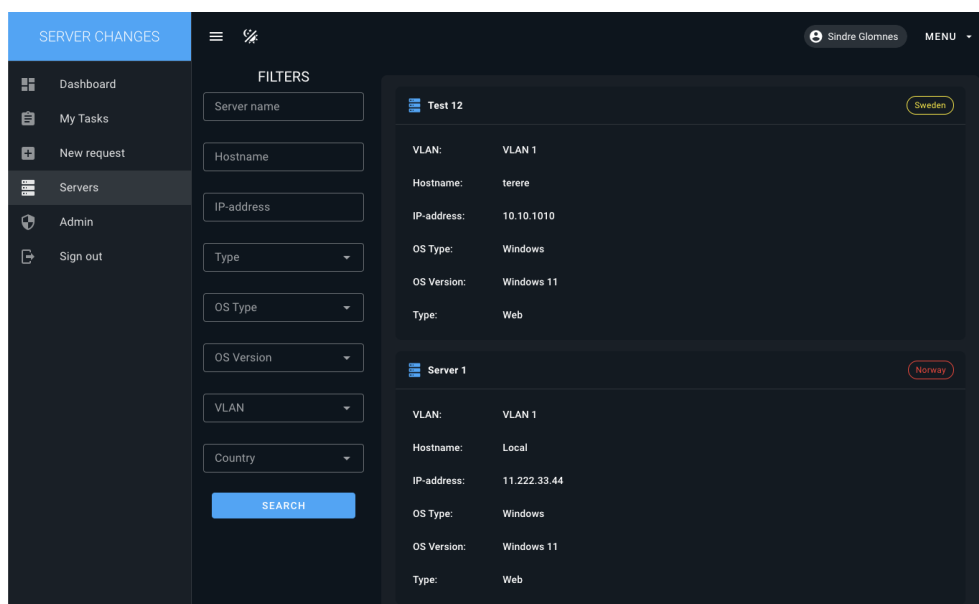
Hvis en bruker vil opprette en ny endring kan en benytte menyen til å navigere til siden for ny forespørsel, visualisert i Figur 6. For å opprette en forespørsel kreves det at brukeren velger tittel, servere endringen skal utføres på, kategorier, prioritet, planlagt dato for endringen og en bruker for godkjenning av forespørselen. Om ønsket kan brukeren legge til beskrivelse, eventuelle risikoer og oppgaver i forbindelse med endringen. En risiko krever en beskrivelse og et estimat på konsekvens og sannsynlighet, hvor begge skal ha en verdi mellom én og ti. Verdien til en risiko kan beskrives som konsekvens multiplisert med sannsynlighet. Den totale risikoen er summen av alle risikoene. Hvis denne totale risikoen overstrider en satt grense, vil det være et krav om å beskrive en tilbakeføringsplan. Hvis det skal legges til en oppgave er det et krav om beskrivelse og en bruker som skal utføre oppgaven. Ved sending vil mottaker av forespørselen varsles på e-post.



Figur 6: Utsnitt av ny endring

Servere

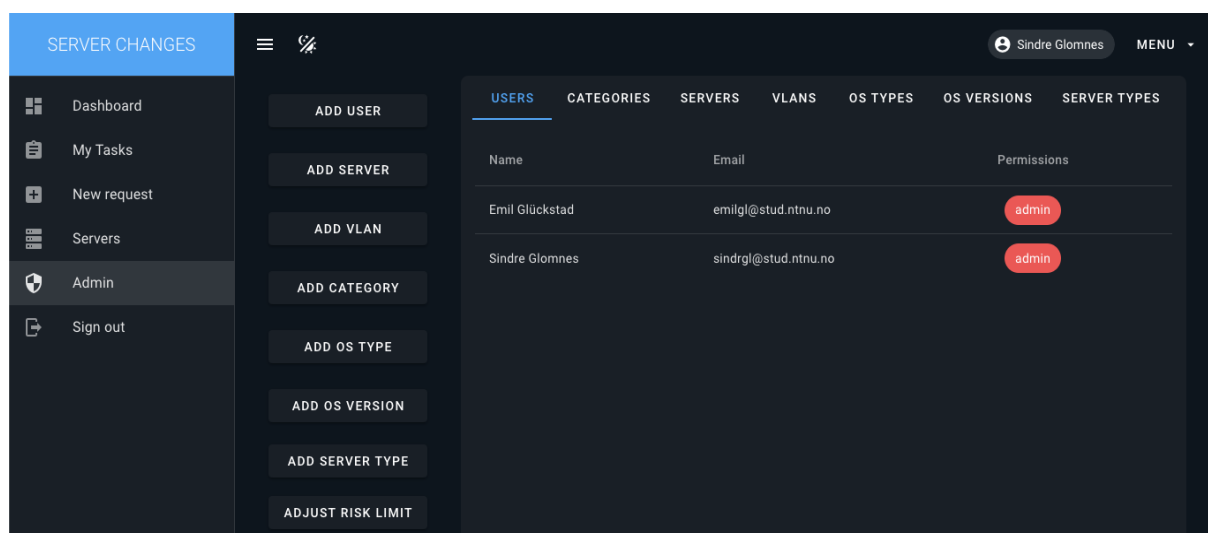
For å få oversikt over alle servere som er lagt inn i systemet, kan brukeren benytte serversiden. På siden er det mulig å filtrere serverene etter alle parameterene som er vist i Figur 7. Ved å trykke på en server i listen, vil en få opp endringshistorikken for denne serveren. Historikken inneholder alle tidligere endringer gjort på tilhørende server.



Figur 7: Utsnitt av server oversikt

Administratorside

Administratører i systemet har tilgang til administratorsiden. Denne inneholder en oversikt over all data som er mulig å legge til og endre, illustrert i Figur 8. Dette inkluderer alt fra brukere som skal ha tilgang, til os-versjoner som brukes i opprettelsen av servere. For å justere grensen for totalsum av risikoer kan en benytte knappen ved navn «Adjust risk limit». Denne bestemmer for hvilken verdi tilbakeføringsplanen skal være obligatorisk ved opprettelse av endringsforespørsel.

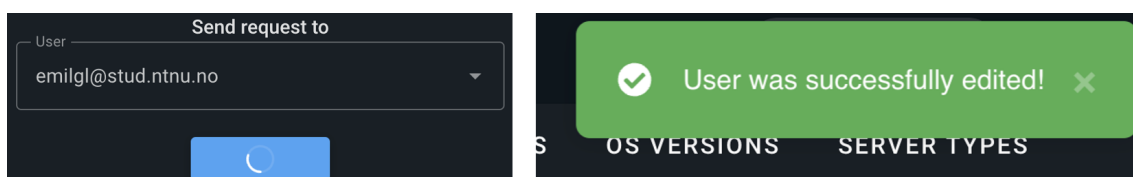


Figur 8: Utsnitt av administratorside

4.1.2 Brukervennlighet

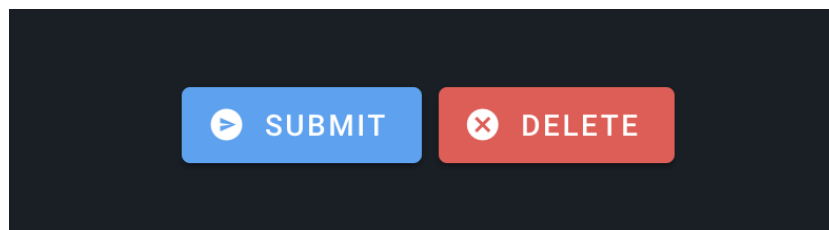
For å presentere brukervennligheten i applikasjonen benyttes Nielsen's 10 Usability Heuristics[8]. Det fremheves eksempler i applikasjonen som er designet med hensyn på disse ti punktene.

Synlighet av systemstatus: Det er benyttet ulike komponenter for å visualisere status for ulike brukerinteraksjoner. Disse er tilbakemeldinger ved interaksjon og komponenter som synliggjør at systemet er i prosess. Eksempler på slike tilbakemeldinger er presentert i Figur 9.



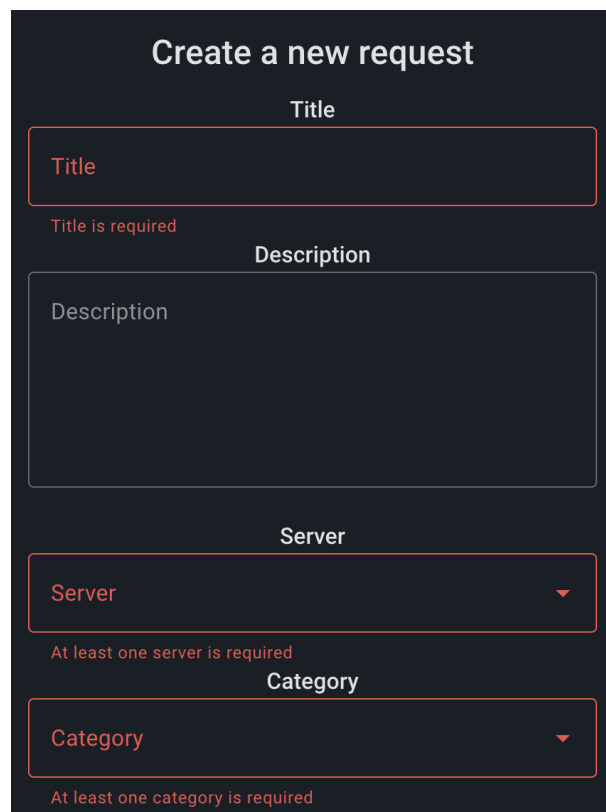
Figur 9: Håndtering av spørring og tilbakemelding

Tilpasning og bruk av standarder: I utformingen av nettsiden er Vuetify benyttet med tanke på å gi nettsiden konsistent utseende og følelse basert på Material Design. Det er benyttet standardiserte fargekoder og ikoner som går igjen på nettsiden. Figur 10 illustrer et eksempel på fargevalg og standardiserte ikoner i applikasjonen.



Figur 10: Fargekode og ikoner ved sending/sletting av endringsforespørsel

Forebygging av feil: For å forebygge feil brukes det validering av inndata. Figur 11 illustrerer opprettelse av en endring hvor brukeren er nødt til å fylle inn feltene tittel, server og kategori. Det er ikke mulig å sende en forespørsel uten disse kriteriene.

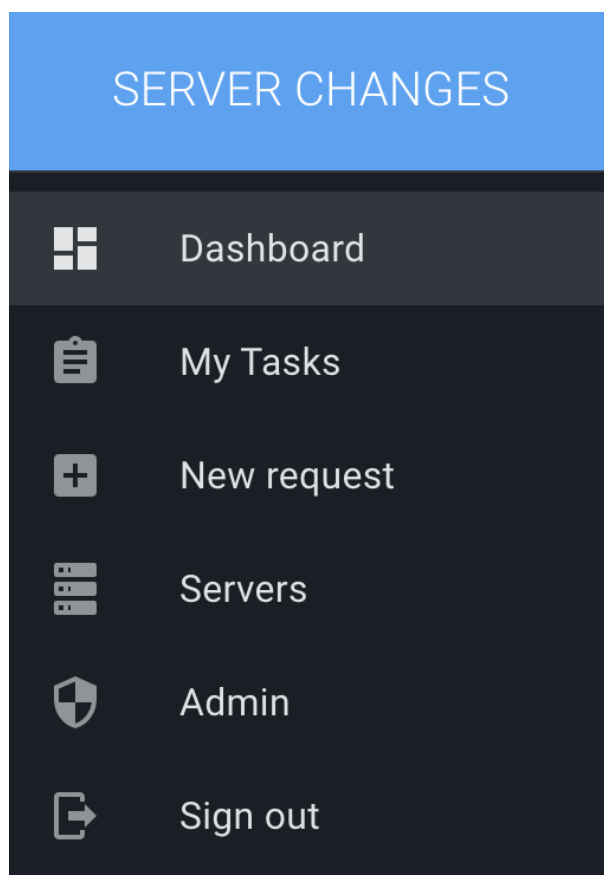


The image shows a dark-themed form titled "Create a new request". It contains four input fields, each with a validation error message in red text below it:

- Title:** The input field contains the word "Title". Below it, the error message "Title is required" is displayed.
- Description:** The input field contains the word "Description". There is no error message for this field.
- Server:** The input field is a dropdown menu showing "Server" with a downward arrow. Below it, the error message "At least one server is required" is displayed.
- Category:** The input field is a dropdown menu showing "Category" with a downward arrow. Below it, the error message "At least one category is required" is displayed.

Figur 11: Validering

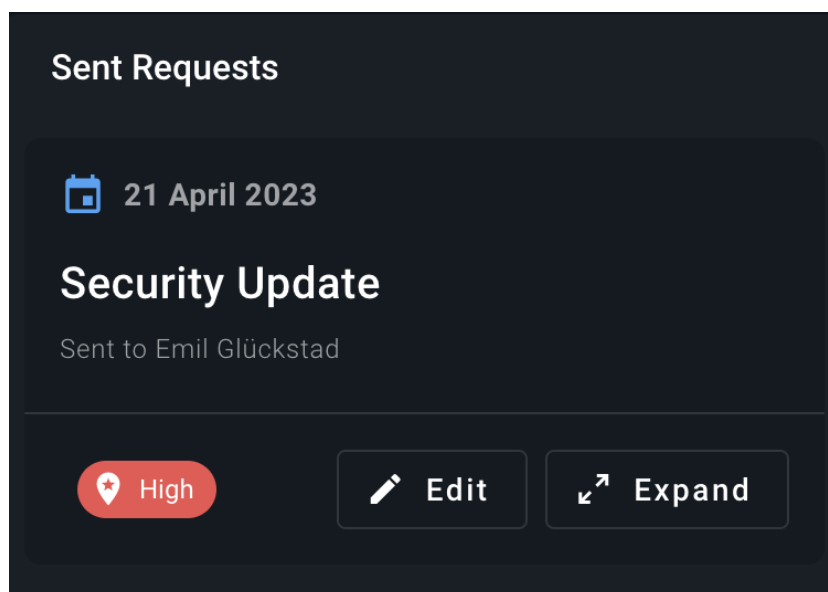
Gjenkjenning framfor husking: For å redusere den kognitive belastningen for brukerne benyttes det ikoner i navigasjonslinjen, vist i Figur 12. Navigering er lett tilgjengelig i venstre marg og som «drop down»-meny i høyre hjørne. Disse kan vises og skjules etter brukerens ønske. Det er også benyttet etiketter på alt av felt for inndata.



Figur 12: Meny i venstre marg

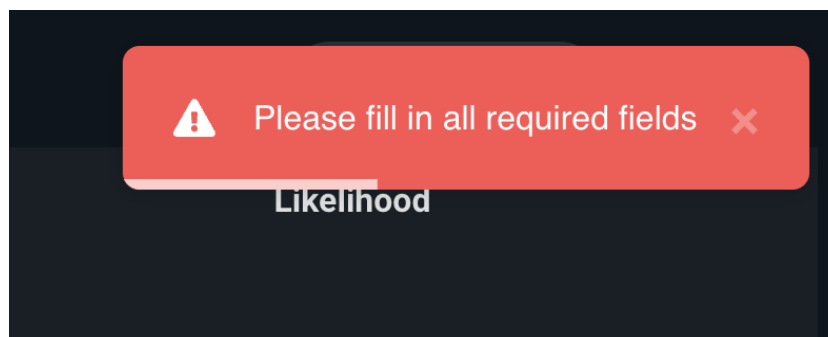
Fleksibilitet og effektivitet i bruk: Nettsiden er utarbeidet med mål om å skape en effektiv arbeidsflyt. Det er derimot ikke lagt inn spesiell fleksibilitet i forhold til om brukeren er ny eller erfaren.

Estetikk og minimalistisk design: For å oppnå høyere grad av brukervennlighet er det fokusert på estetikk og minimalistisk design. Dette er gjort ved å kun presentere nødvendig informasjon samtidig som brukeren har mulighet til å utvide for mer informasjon. Et eksempel er en endringsforespørsel i Dashboardsidene, vises i Figur 13 . Brukeren får kun se den viktigste informasjonen som tittel og prioritet. Det er også benyttet fargekoder på utvalgt informasjon som prioritet. Ved behov kan brukeren benytte knappen «Expand» for mer informasjon.



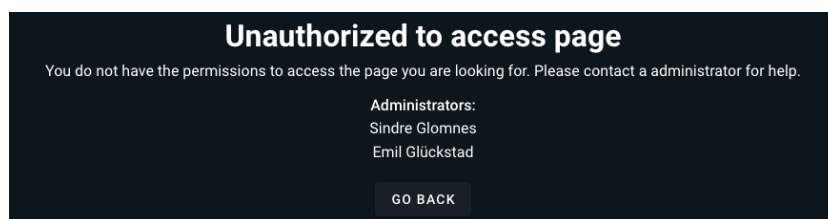
Figur 13: Listeelement av endringsforespørsel

Hjelp brukeren med å forstå og rette opp i feil: For å hjelpe brukeren å forstå problemer feil i applikasjonen brukes det toaster som vist i Figur 14. Disse inneholder en beskrivelse om hva som gikk galt, og dukker opp både for feil på klient- og serverside.



Figur 14: Feilmelding

Hjelp og dokumentasjon: For å hjelpe brukere som mangler tillatelse har applikasjonen en side hvor alle administratorene listes opp, vist på Figur 15. Denne siden forteller brukeren hvem som skal kontaktes for hjelp. Applikasjonen har ingen side for dokumentasjon.



Figur 15: Uautorisert

Konsistens og standarder: For å sørge for at webapplikasjonen har et konsistent og standard design er Vuetify benyttet[21]. Vuetify er et bibliotek som inneholder et stort utvalg komponenter basert på Material Design. De ulike komponenter, ikoner og fargekoder går igjen på nettsiden for å skape et konsistent design. Fargene endres ut fra om systemet er i mørk eller lys modus.

Brukerkontroll og frihet: For å sørge for brukerkontroll og frihet er det mulig å både redigere og slette sendte endringer. Som administrator er det mulig å redigere alle objekter som er lagt til i applikasjonen.

4.1.3 Brukertest

Det er gjennomført en brukertest for å vurdere brukervennligheten til applikasjonen [Vedlegg H]. Testen omhandler at brukerne skal ta i bruk applikasjonen for å utføre oppgaver, og vurdere hvor brukervennlig systemet er. For denne oppgaven ble det utvalgt åtte realistiske brukere som utførte brukertesten. Testen består av to deler. Del én inneholder aktiviteter med totalt 14 spørsmål. Tolv av spørsmålene har som formål å få brukerne til å benytte seg av alle hovedfunksjonene i applikasjonen, presenteres i Tabell 1. To av spørsmålene gir brukerne mulighet til å gi skriftlige tilbakemeldinger. Det blir gitt generelle beskrivelser av hva som skal utføres, uten å forklare ytterligere om hvordan det skal utføres. Brukerne må derfor finne ut selv hvordan de skal benytte applikasjonen for å løse oppgavene i tråd med beskrivelsen. Hver oppgave har tre svaralternativer: om man fikk til oppgaven, om man ikke fikk til oppgaven, eller om man er usikker. En slik oppdeling gir en indikasjon på hvilken deler av applikasjonen som er utfordrende for nye brukere å benytte.

Oppgave	Ja	Nei	Vet ikke
Fikk du opprettet og sendt forespørselen?	100%		
Fikk du slettet forespørselen?	100%		
Ser du hvem som sendte forespørselen?	100%		
Fikk du godkjent forespørselen?	87.5%	12.5%	
Fikk du lagt til kommentar?	87.5%	12.5%	
Fikk du sendt den kritiske endringen?	100%		
Jeg fikk satt begge oppgavene til fullført?	75%	25%	
Jeg fikk satt endringen som fullført?	87.5%	12.5%	

Fant du dokumentasjonen?	100%		
I dokumentasjonen, kan du se klokkeslettet forespørselen først ble sendt?	100%		
Fikk du opprettet en server?	100%		
Fikk du endret land på serveren?	100%		

Tabell 1: Aktivitetsoppgaver

Del to av brukertesten inneholder ti utsagn basert på System Usability Scale[17], presenteres i Tabell 2. Brukerne skal vurdere hvor enige de er i hvert utsagn på en skala fra 1 til 5, hvor 1 er helt uenig og 5 er helt enig. For å beregne poengsummen, multipliseres gjennomsnittspoengsummen til hvert utsagn med to, slik at det resulterende tallverdien ligger mellom ti og 100. For spørsmål hvor en verdi på fem indikerer en negativ vurdering, vil poengene bli reversert.

Utsagn	1	2	3	4	5	Poeng
Jeg tror jeg ville likt å bruke dette systemet ofte.			12.5%	25%	62.5%	9
Jeg syntes systemet var unødvendig komplisert.	62.5%	25%	12.5%			9
Jeg tror at jeg vil trenge hjelp fra en teknisk person for å kunne bruke dette systemet.	75%	25%				9.5
Jeg synes at de ulike funksjonene i dette systemet er godt integrert.			25%	37.5%	37.5%	8.25
Jeg syntes at det var for mye inkonsekvens i dette systemet.	37.5%	37.5%		25%		7.75
Jeg kan forestille meg at de fleste mennesker vil lære å bruke dette systemet veldig raskt.		12.5%		12.5%	75%	9
Jeg synes at systemet er veldig kronglete å bruke.	62.5%	37.5%				9.25
Jeg føler meg trygg på å bruke dette systemet uten hjelp fra en teknisk person.				50%	50%	9
Jeg må lære mange nye ting før jeg kan bruke dette systemet effektivt.	87.5%	12.5%				9.75
Jeg synes at dette systemet er enkelt å bruke.			12.5%	25%	62.5%	9

Tabell 2: System Usability Scale

4.2 Ingeniørfaglige resultater

I planleggingsfasen stilles det en rekke krav til produktet, både funksjonelle og ikke funksjonelle. Disse kravene er beskrevet i visjonsdokumentet [Vedlegg B] og utfylt i form av brukerhistorier i kravdokumentasjon [Vedlegg C]. Kravene ble deretter prioritert i samråd med oppdragsgiver. Denne prioritetstabellen er lagt ved rapporten som vedlegg [Vedlegg F]. Under utviklingsfasen av produktet ble krav og ønsker endret og nye krav og ønsker ble utformet. Nedenfor vil noen av kravene fra visjonsdokumentet bli presentert med status for kravet.

4.2.1 Funksjonelle krav

I Tabell 3 blir noen av de funksjonelle kravene for produktet beskrevet. 36 av 36 funksjonelle krav ble oppfylt. Kravene som er presentert her er de kravene der det er relevant med en kommentar. Den komplette listen med funksjonelle krav stilt av oppdragsgiver finnes i visjonsdokumentet [Vedlegg B].

Kategorisere endringsforespørsel	x		Underveis ble det også utformet ønske om å kunne velge flere kategorier for en endring. Dette ble utført.
Fylle ut risikomatrise	x		Ved oppretting av en serverendring fylles ikke en matrise ut, men en har mulighet for å opprette risikoer og gi de verdi fra 1-10 for sannsynlighet og konsekvens. Ved å åpne en serverendring vil risikoene visualiseres i en matrise.
Endring av avvist forespørsel	x		Dette kravet ble modifisert da det var ønskelig å kunne endre endringsforespørsler som ikke var blitt avvist eller godkjent. Dette ble implementert med versjonskontroll av endringsforespørlene.
Endre risiko på avslått forespørsel	x		Se kommentar over.
Endre oppgave på avslått forespørsel	x		Se kommentar om avvist forespørsel.
Kreve rollbackplan hvis samlet risiko overstiger en gitt grense	x		Dette kravet ble modifisert til at grensen skulle være justerbar for administrator. Dette ble utført.

Avvise endringsforespørsel	x		Det ble utformet krav om å kunne gi tilbakemelding på hvorfor det ble avvist. Dette gjelder også for godkjenning og ble utført.
Godkjenne endringsforespørsel	x		Se kommentar over.
Sletting av fullførte oppgaver tilknyttet serverendring	x		Fullførte oppgaver kan slettes fra egen oppgaveoversikt, men slettes ikke fullstendig da de fortsatt skal finnes i endringen.
Administrator kan slette bruker	x		Med sletting er det fjerning av tilgang og visning av bruker som er gjort. Brukeren er ellers lagret i databasen for å bli vist på serverendringer der de er involvert.
Hente frem all informasjon om en gitt endring	x		På siden med all informasjon ble det fremstilt ønske om å kunne se alt som er skjedd med endringen og tilknyttede oppgaver på veien deres i systemet. Dette ble utført med historikk for både serverendring og statusendring på oppgaver
Filtrere servere etter Hostname, IP, VLAN, OS type, OS versjon, land og status.	x		Filtrering på status ble forkastet og filtrering på navn ble lagt til.

Tabell 3: Funksjonelle krav

4.2.2 Ikke funksjonelle krav

I Tabell 4 blir de ikke funksjonelle kravene stilt til produktet presentert. 5 av 5 ikke funksjonelle krav ble oppfylt og disse er beskrevet her med tilhørende kommentar der det er relevant.

Ikke funksjonelle krav	Oppfylt	Ikke Oppfylt	Kommentar
Innlogging via Microsoft Office firmabruker	x		
To-faktorautentisering	x		Dette ble ikke utført programmatiske, men ved bruk av innlogging med Microsoft og Azure kan dette velges der.
Kjøre på Linux server	x		
Data lagres i relasjonell database fortrinnsvis PostgreSQL	x		PostgreSQL ble benyttet.
Lagre brukerdata i informasjonssjunkskapsel	x		
Engelsk språk	x		

Tabell 4: Ikke funksjonelle krav

4.3 Administrative resultater

De administrative resultatene innebærer presentasjon av resultatene forbundet med utviklingsmetode, fremdriftsplan og timeregnskap.

4.3.1 Utviklingsmetode

Gruppen har i arbeidet med produktet benyttet smidig utvikling. GitLab [6] er benyttet for produktkø med oversikt over oppgaver som skal utføres og som er under arbeid. De opprinnelige oppgavene ble prioritert i samråd med oppdragsgiver [Vedlegg F].

Før utviklingen ble påbegynt ble det laget en wireframe lagt ved [Vedlegg G]. Denne ble presentert for oppdragsgiver. Wireframen sammen med tilbakemeldingene på den ble brukt som en form for mal i begynnelsen av utviklingsfasen.

I løpet av utviklingsfasen arrangerte gruppen regelmessige møter med oppdragsgiver. Under disse møtene ble fremgangen presentert i form av daværende versjon av nettsiden. Dette ble en arena for oppdragsgiver til å observere foreløping produkt og komme med tilbakemeldinger. I disse møtene ble flere nye krav og ønsker utformet. Deretter ble disse oppført i produktkøen. Møteinnkallinger og referater fra møtene er presentert i prosjekt-håndboken [Vedlegg E].

4.3.2 Fremdriftsplan

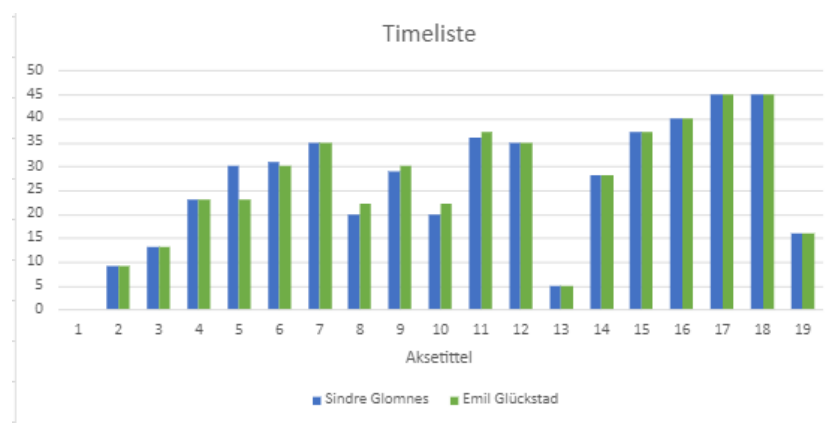
Gruppen opprettet et Gantt diagram i forbindelse med forprosjektplanen [Vedlegg A]. Dette var en skisse for fremdriften og tidsbruken under arbeidet med oppgaven. Da diagrammet var en lite detaljert skisse var det begrenset hvor mye denne ble benyttet under arbeidet. Diagrammet ble likevel brukt for sammenligning med status for oppgaven for å ha oversikt over prosjektets fremdrift.

Det mest relevante å trekke ut fra Gantt diagrammet er fristen gruppen satt for å ha utformet en MVP (Minimal Viable Product) versjon. Denne ble overholdt med mere funksjonalitet enn gruppen hadde sett for seg skulle være med på denne versjonen.

4.3.3 Timeregnskap

Det er underveis i arbeidet med oppgaven skrevet timelister for å holde oversikt over antall timer brukt. For hver uke finnes det tilhørende korte beskrivelser for hva som er gjort den uken av hvilket gruppemedlem. Timelisten er lagt ved i prosjekthåndboken [Vedlegg E].

I Figur 16 er antall timer brukt av hvert gruppemedlem for hver uke illustrert. Som en ser av figuren har gruppen arbeidet med bachelor oppgaven relativt jevnt over hele prosjektperioden.



Figur 16: Fordeling av timer per uke

5 Diskusjon

I dette kapittelet vil resultatene presentert i forrige kapittel diskuteres og bli satt opp i mot teorien i kapittel 2 og metode og valg av teknologi i kapittel 3. Kapittelet belyser styrker og svakheter ved arbeidet som er utført og årsakene til hvorfor resultatene ble som de ble.

5.1 Diskusjon av vitenskapelige resultater

Diskusjonen rundt de vitenskapelige resultatene presenteres gjennom tre hovedområder: webapplikasjon, brukervennlighet, og brukertesten. Først vil det bli redegjort for de designvalgene som er gjort i forhold til webapplikasjonen, og hvorfor disse valgene ble tatt. Deretter vurderes de tiltakene som er iverksatt for å forbedre brukervennligheten. Til slutt, basert på tilbakemeldingene fra brukertesten, vil det bli pekt på potensielle områder for videre forbedringer i brukervennligheten.

5.1.1 Webapplikasjon

Webapplikasjonen er tilnærmet lik kundens beskrivelse. Deres beskrivelse av applikasjonen omhandlet en prosess for server-endringer som er satt opp som en produktkø. Det ble presentert flere iterasjoner av grensesnittet for kunden før det ble enighet om det endelige designet. Den første iterasjonen av produktkøen var en vertikal fremstilling av sendte, mottatte og klare-for-implementering endringer. Fordelen med dette designet er rom for mer informasjon i hver komponent. Kunden ønsket en horisontal fremstilling. Det ble derfor fremvist et responsivt design hvor produktkøen er horisontal for store skjermer, og vertikal for mindre skjermer som nettbrett.

Idéen om produktkø i dashbordet er at en opprettet endring må bevege seg mot høyre for å bli fullført. Dette er en logisk måte å tenke på, og gir et kjapt overblikk over hvor i prosessen en endring ligger. Det ble valgt å sette opp produktkøen for endringer fordelt på sendt, mottatt og klar-for-implementasjon. Selv om dette etterligner en produktkø, fungerer det annerledes. En endringsforespørsel vil være på forskjellige posisjon for ulike brukere. Når en bruker sender en endringsforespørsel, vil den være synlig under «sendt». Brukeren som skal godkjenne samme forespørsel, ser den på en annen posisjon, under «mottatt». Dette er en svakhet med designet da det gir inntrykk om at endringen er på ulike stadier, avhengig av hvilken bruker man er. En annen måte å designe dashbordet på kunne vært å ha én felles rad for «venter på godkjenning». I dette designet vil sendte og mottatte endringen ha samme posisjon for begge parter.

Det har blitt avdekket noen mangler ved applikasjonen. På grunn av måten produktkøen er satt opp, kan ikke mottaker se en endring etter godkjenning. Dette er fordi endringer klare for implementasjon kun kan ses av avsender. Dette kan løses ved å gjøre endringen

synlig for alle brukere involvert i endringen. Disse brukerne inkluderer avsender, mottaker samt brukere med oppgaver knyttet til endringen. «My Tasks» siden er heller ikke en nødvendighet. Oppgaver kunne blitt endret status på direkte fra siden til den tilhørende endringen. En slik løsning hadde likevel også hatt noen ulemper i form av at det hadde manglet en fullstendig oversikt for alle aktive oppgaver. Det optimale valget kunne vært en kombinasjon av løsningene.

Kunden er fornøyd med det endelige designet. Det inneholder alt av funksjonalitet som var ønsket. I prototypen inneholdt applikasjonen mange flere sider i menyen. Disse sidene skulle brukes av administratorer for å legge til ulike objekter i programmet. I den endelige versjonen ble dette endret til én felles administratorside som inneholder alt av den samme funksjonaliteten. Dette ble vurdert som en bedre løsning da det ikke er nødvendig med en hel side til hver funksjon.

Webapplikasjonen nådde sitt mål om å tilby en løsning med alt av ønsket funksjonalitet fra kunden. Ved å forenkle menyen og konsolidere administrasjonsfunksjonene til én side, er applikasjonen mer oversiktlig og enklere å navigere. Kunden er fornøyd, og applikasjonen fungerer som en solid plattform for både avsender og mottaker i prosessen med serverendringer.

5.1.2 Brukervennlighet

For å utvikle en brukervennlig nettside er det arbeidet tett opp mot de ti prinsippene for brukervennlighet[8]. Det er samtidig først og fremst fokusert på å lage en nettside som oppfyller oppdragsgivers krav. Dette har ført til at det er gjort flere avveininger når det kommer til enkelte aspekter ved brukervennlighet. Disse avveiningene og løsninger som har blitt benyttet for å utforme en brukervennlig nettside, diskuteres i dette kapitlet.

Fleksibilitet og effektivitet i bruk: Det er som nevnt i resultat kapitlet ikke lagt inn spesiell fleksibilitet i forhold til om brukeren er erfaren eller ny til nettsiden. Dette valget er tatt på bakgrunn av nettsidens formål. Nettsiden vil være et verktøy for et fåtall utvalgte ansatte ved Geomatikk AS. Det er derfor ikke like relevant med høy grad av fleksibilitet som det er for nettsider med bred målgruppe. Gruppen har derfor fokusert på å lage et effektivt verktøy og i større grad sett bort fra fleksibilitet.

Hjelp og dokumentasjon: Nettsiden har ingen hjelpeside. Dette er et valg tatt på bakgrunn av mye av det samme som nevnt ovenfor. Det er en snever målgruppe for nettsiden. De brukerne som vil bruke den i sin arbeidshverdag vil motta opplæring før de benytter seg av den. Det er derfor ikke rettet fokus mot å utvikle en egen side for hjelpedokumentasjon.

Løsninger for brukervennlighet i fokus: De fleste prinsippene rundt brukervennlighet er tatt hensyn til og har vært naturlig å inkludere i produktet. Nedenfor vil noen av løsningene for å gjøre nettsiden brukervennlig diskuteres.

- Vuetify ble benyttet ved utvikling av klientsiden. Gruppen er fornøyd med det dette valget først og fremst for muligheten det gir til å benytte standardiserte og konsistente komponenter. Dette har igjen vært viktig i arbeidet mot å utforme et konsistent design over hele nettsiden.
- Det har vært fokusert på bruk av standardiserte ikoner og fargekoder som illustrerer de ulike elementene på nettsiden. Disse er også i stor grad gjenbrukt på nettsiden for å tilby en konsistent følelse. Fargekoder og ikoner er viktig for at sentral informasjon skiller seg fra omgivelsene.
- Det er benyttet tilbakemeldinger og elementer som visualiserer systemets status for at brukeren skal ha innsikt i og kontroll over etterfølgende av en handling.
- Validering av inndata er det viktigste grepet tatt for å hjelpe brukeren å unngå feil. Utfyllelse av datfelt er en sentral del av nettsiden. Validering av disse har derfor vært naturlig for å skape flyt på nettsiden og unngå unødvendige feil og eventuelle systemkræsje.

Universell utforming:

Under arbeidet med nettsiden har ikke universell utforming vært hovedfokus. Dette i samråd med oppdragsgiver og på bakgrunn av tids- og ressursbegrensningene for oppgaven. Det er likevel sterke sammenhenger mellom universell utforming og brukervennlighet. Derfor er det relevant å diskutere noen av prinsippene ved universell utforming.

- **Fleksibilitet:** Individuelle preferanser er begrenset på nettsiden. En individuell preferanse som er inkludert er mulighet til å arbeide i mørk eller lys modus. Når det kommer til for eksempel implementering av lyd for synshemmede er dette ikke implementert. Dette er på bakgrunn av oppgavens tidsbegrensning og at det på nåværende tidspunkt ikke er et behov for produktets snevre målgruppe.
- **Intuitiv bruk:** Som nevnt ovenfor er det arbeidet mot en konsistent brukeropplevelse. Dette er en sentral del i arbeidet med å utvikle en intuitiv nettside. Det er lagt fokus på å ha mest mulig funksjonalitet tilgjengelig innenfor færrest mulig klikk uten å gjøre dette uoversiktlig. Det konsistente designet sammen med fargekoder og ikoner har vært under fokus og bidrar til at nettsiden skal være intuitiv for alle brukere uavhengig av erfaring, kunnskap, språkferdigheter og konsentrasjonsnivå.
- **Merkbar informasjon:** Igjen er fargekoder og ikoner et sentralt verktøy for å sørge for at viktig informasjon skiller seg ut og blir kommunisert til brukeren uavhengig av brukerens sensoriske evner. Et annet grep som er tatt er og minimere informasjonen som vises frem til brukeren eventuelt klikker seg inn på et objekt for å se alt av informasjon. Dette er en balansegang da det som nevnt om intuitiv bruk er ønskelig å ha mest mulig tilgjengelig med færrest mulig klikk. Det er derfor blitt vurdert nøye hvilken informasjon som bør vises på hver enkelt del av nettsiden. Gruppen har ved fremvisning av produktet og igjennom brukertester observert at det ved noen deler av nettsiden kunne vært mere informasjon tilgjengelig uten å klikke seg videre.

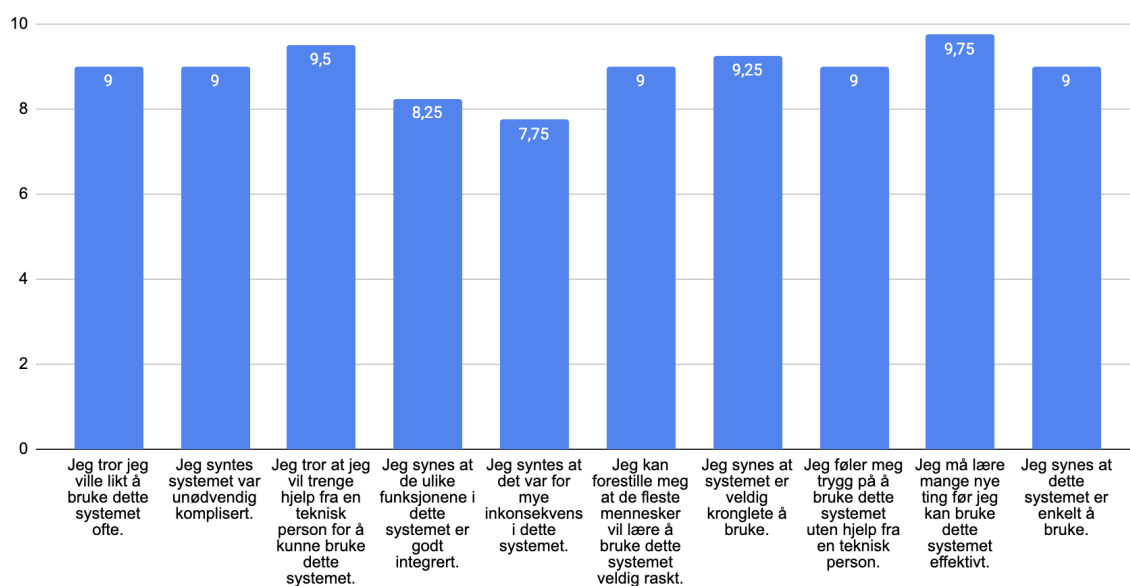
- **Lav fysisk anstrengelse:** Det er på nettsiden implementert mulighet for tastaturnavigasjon ved de fleste tilfeller. Gruppen har likevel opplevd noe inkonsistens på dette område og det er derfor satt som et forbedringspunkt for videre arbeid. Ett annet og tydelig forbedringspunkt er oppgavesiden hvor brukeren endrer status på oppgave ved «dra og slipp»- funksjonalitet. Her bør det implementeres mulighet for å endre status med klikk for å redusere behovet for musbruk ytterligere. Det å redusere behovet for musbruk er viktig som del av generell brukervennlighet for at tretthet ved bruk av nettsiden minimeres. Enda viktigere er det for universell utforming for å gjøre nettsiden tilgjengelig for brukere med ulike evner og eventuelle funksjonsnedsetninger.

Hovedprioriteringen under utviklingen har vært brukervennlighet, men det er samtidig tatt noen hensyn og gjort noen prioriteringer utfra nettsidens formål. Ved å prioritere bort noe fleksibilitet, hjelpeside og enkelte deler ved universell utforming er noe mere tid blitt tilgjengelig. Denne tiden er blitt brukt på å gjennomføre andre prinsipper ved brukervennlighet og universell utforming blant annet det som er trukket frem ovenfor.

5.1.3 Brukertest

Før resultatene av brukertesten diskuteres er det relevant å diskutere gjennomføringen av den. Gruppen ønsket å teste brukervennlighet og om nettsiden var intuitiv. Det ble derfor laget et skjema hvor det ble gitt oppgaver som forklarte hva vi ønsket testet, men ikke hvordan de skulle gjennomføres. Dette måtte testpersonene finne ut selv. Gruppen er fornøyd med gjennomføringsmetoden da både styrker og svakheter i brukervennligheten kom tydelig frem både ved resultatene og ved observasjoner underveis.

Resultatene fra brukertesten viste en viss inkonsekvens i systemet, som vist i Figur 17. Dette antyder at mens noen brukere hadde en god opplevelse, opplevde andre brukere utfordringer med systemet. Det ble observert at dette er spesielt knyttet til «My Tasks» og «Dashboard» sidene.

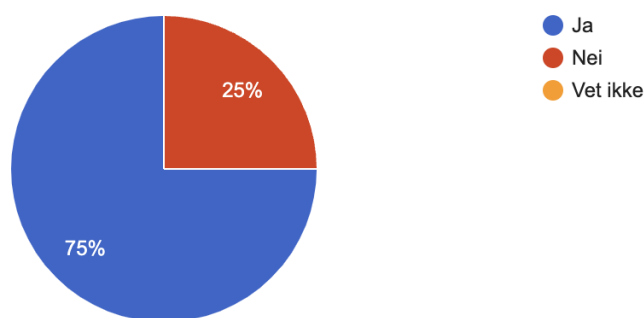


Figur 17: Brukertest

Inkonsistensen[8] i «Dashboard» og «My Tasks» kommer av at de har lignende design, men ulike interaksjoner. «Dashboard» håndterer et stort utvalg interaksjoner som redigering, sletting, godkjenning og mulighet for kommentarer. De fleste av interaksjonene krever at brukeren klikker seg inn i komponenten og dermed benytter knapper. I «My Tasks», derimot benyttes kun dra-og-slipp interaksjonen da den eneste funksjonen er å endre status. De lignende designene skapes en forventning om like interaksjoner på begge sidene. Brukerne hadde problemer med å oppdage dra-og-slipp interaksjonen, som vist i Figur 18. Dette er en mangel på merkbar informasjon[11]. For å synliggjøre interaksjonen kunne ikoner eller animasjoner vært tatt ibruk. «My Tasks» siden mangler også prinsippet om lav fysisk anstrengelse da det kun er mulig å endre status ved bruk av mus. For å løse dette kan knapper implementeres som ett alternativ å endre status.

Jeg fikk satt begge oppgavene til fullført.

8 svar



Figur 18: Oppgaver satt til fullført

Det ble verifisert at webapplikasjonen hadde problemer med synlighet av systemstatus[11]. Dette omhandler godkjenning av server-endring. Når en serverendring godkjennes blir den satt som klar for implementasjon for avsender. Dermed forsvinner den fra produktkøen til mottakeren. For å fikse problemet kan endringer som er klar-til-implementasjon synliggjøres for alle relevante brukere, som avsender og mottaker.

Muntlig tilbakemelding ble gitt angående forvirring rundt enkelte tekster i applikasjonen. Når en bruker oppretter en endringsforespørsel, må de fylle ut «Send request to», der de skal velge en annen bruker. Dette førte til forvirring, da det kunne oppfattes som om man valgte hvilken bruker som skulle utføre oppgavene tilknyttet endringen, i stedet for hvem som skulle godkjenne forespørselen. På samme måte skapte «Complete»-knappen forvirring når man skal fullføre en serverendring. Denne kunne tolkes som om knappen skulle brukes for å fullføre oppgaver relatert til endringen. Det bør derfor fokuseres på å presisere all tekst i programmet for å unngå misforståelser.

Til tross for utfordringene som ble identifisert i brukertesten, er det også viktig å anerkjenne de positive sidene av systemet. Total scoren utarbeidet fra svarene på påstandene fra SUS[17] endte på 89.5 poeng. Dette er godt over gjennomsnittsscoren for brukervennlighet målt med SUS som ligger på 68 poeng. Det er viktig å bygge videre på de sterke sidene av systemet og samtidig ta tak i de identifiserte problemene for å skape en mer konsistent og brukervennlig opplevelse for alle brukere.

5.1.4 Oppsummering

Produktkø fungerer som en brukervennlig metode for å loggføre prosessen forbundet med endringer på servere. Dette kan bekreftes fra brukertesten som oppnådde en poengsum på 89.5. Svakheten til webapplikasjonen omhandler inkonsekvenser, fleksibilitet og merkbar informasjon. Produktkøen kan forbedres med å synliggjøre endringer gjennom hele prosessen for alle relevante brukere. Dermed vil brukerne alltid være informert om hvor i prosessen endringen befinner seg. Inkonsekvensen og mangelen på fleksibilitet rundt oppgavene kan løses med å gi brukeren mulighet til å endre status på flere ulike måter.

5.2 Diskusjon av ingeniørfaglige resultater

5.2.1 Funksjonelle krav

For å sikre en klar forståelse av kundens ønsker, ble det etablert en prioritetstabell. Tabellen inneholdte alle funksjonelle krav som var nødvendig for å oppnå deres ønsker. Underveis i prosjektet ble det motatt tilbakemeldinger fra kunden, som førte til at flere funksjonelle krav ble definert og justert. Et eksempel på dette er historikk for hver enkelt endring, som ble lagt til underveis som et krav etter kundens tilbakemelding. Dette understreker viktigheten av å være i kontinuerlig kontakt med kunden for å tilpasse seg

endringer i prosjektet etter deres behov.

Ved å oppfylle alle 36 funksjonelle krav, har det blitt utviklet en applikasjon som møter kundens forventninger. Den kontinuerlige kommunikasjonen med kunden har vært avgjørende for å oppnå et vellykket resultat. Evnen til å tilpasse oss endringer har også bidratt til å sikre at applikasjonen imøtekommer kundens behov.

5.2.2 Ikke funksjonelle krav

De ikke funksjonelle kravene stilt av oppdragsgiver ble oppfylt. Av disse er det mest relevant og trekke frem datalagring og sikkerhet forbundet med innlogging.

Datalagring: Det var et krav fra oppdragsgiver å lagre data i relasjonell database. PostgreSQL ble i den sammenheng nevnt av oppdragsgiver som fordelaktig for deres systemer. Gruppen gikk likevel i gang med å se på ulike muligheter for relasjonelle databaser. Til slutt endte gruppen med å bruke PostgreSQL. Valget er begrunnet i større detalj i «Valg av teknologi» under kapittel 3.

Innlogging: Oppdragsgiver hadde et krav om å benytte innlogging med Microsoft. Dette for at de ansatte hos oppdragsgiver skulle kunne benytte firmabruker for innlogging på nettsiden. Gruppen valgte å prioritere dette i starten av utviklingsfasen slik at innloggingen var på plass. For å oppfylle kravet måtte gruppen sette seg inn i Azure og koblingen mellom Azure og kildekoden. Dette tok noe tid, men endte også med å gi fordeler som sikkerheten forbundet med å benytte en anerkjent operatør som Microsoft. En annen fordel er at ved bruk av Microsoft innlogging og Azure gjaldt kravet om to-faktorautentisering. Med Azure er det mulighet for ulike autentiseringsmetoder. Med et enkelt valg kan en styre om to-faktorautentisering kreves av brukeren eller ikke. I tillegg til at en har kontroll over hvilken metode for to-faktorautentisering som kan benyttes av brukere.

5.3 Diskusjon av administrative resultater

5.3.1 Utviklingsmetode

Gruppen brukte elementer fra smidig utvikling under arbeidet med oppgaven. Det å benytte et rammeverk som «SCRUM» for utvikling ble sett på som lite hensiktsmessig da gruppen består av kun to medlemmer. Med kun to gruppemedlemmer kommer god informasjonsflyt i større grad naturlig.

Det ble opprettet produktkø hvor oppgavene ble prioritert i samråd med oppdragsgiver. Underveis som det ble fremmet nye krav eller gjort endringer på allerede eksisterende krav ble produktkøen oppdatert. I startfasen ble det utformet en wireframe med utgangspunkt i den informasjonen som var formidlet på daværende tidspunkt. Denne ble presentert og etter tilbakemeldinger på denne startet gruppen på utviklingsfasen. Versjoner av produk-

tet ble presentert og demonstrert for oppdragsgiver underveis i møtene og det var dette som la grunnlag for utforming av nye krav og endring av tidligere krav.

Gruppen er fornøyd med utviklingsmetoden som ble brukt. Kommunikasjonen var god og kontinuerlig uten at det ble brukt unødvendig tid på rammene rundt denne kommunikasjonen. Produktkøen fungerte bra og ga oss oversikt over hva som skulle gjøres og hva som var under arbeid. Produktkøen hjalp også med å gi oversikt over hva som var gjort på backend og hva som var gjort på frontend. Dette var nyttig for samkjøring av backend og frontend ettersom gruppen som regel var fordelt slik at en av medlemmene arbeidet på frontend og den andre på backend. Til slutt fant gruppen regelmessige møter med oppdragsgiver avgjørende for å utvikle et produkt tilpasset deres behov.

5.3.2 Fremdriftsplan

Gannt diagrammet som ble opprettet i startfasen og som er lagt ved i forprosjektsplanen [Vedlegg A] ble lite brukt under arbeidet med oppgaven. Dette er mye grunnet at den ble opprettet som en skisse da gruppen satt med begrenset informasjon. Skissen var lite spesifikk og det ble derfor vanskelig å dra mye nytte ut av den. Ved å ikke forholde seg til Gannt diagrammet fikk gruppen en naturlig fremgang noe som fungerte godt. Gruppen hadde ingen problemer med å forholde seg til de fristene som var oppført i Gannt diagrammet der den mest relevante var fristen for å ha utviklet en MVP. Oversikten over fremgangen kom istedet frem igjennom timelister som ga gruppen god kontroll på prosjektets fremdrift.

5.3.3 Timeregnskap

Gruppen har skrevet timer og kategorisert aktiviteter som er arbeidet med i forbindelse med bacheloroppgaven. Dette har vært nyttig for gruppen av ulike årsaker. Først og fremst så ble det enkelt å holde kontroll over tidsbruken i forhold til den totale tiden til rådighet på 500 timer per gruppemedlem. Timelistene hjalp også til og holde oversikt over aktivitetene som var blitt gjort og fremgangen i prosjektet. Ettersom timelistene aktivt ble sett på kunne gruppen ha ønsket at den var gjort noe mere detaljert. Det var et bevisst valg å ikke bruke mye tid på timelistene, men ved utføring av prosjektet på nytt kunne det med fordel vært brukt noe mere tid på dette.

Under resultatene omhandlende timeregnskap i forrige kapittel ser en at timene hovedsaklig er jevnt fordelt utover ukene. Dette var en bevisst strategi og har bidratt til at gruppen har følt på kontroll over prosjektets fremgang og unngått unødvendig stress og hastearbeid mot slutten av prosjektet.

5.3.4 Gruppearbeidet

Gruppen er godt fornøyd med samarbeidet. Oppgaver ble fordelt utfra ønske og egenskaper og det har ikke vært noe problem knyttet til dette. Det har hele tiden vært noe å gjøre for begge gruppemedlemmene. Dette har gjort at gruppen har unngått skjevfordeling og bidratt til å skape en følelse av eierskap til prosjektet fra begge gruppemedlemmer. Det har oppstått noen diskusjoner underveis rundt ulike løsninger uten at dette har videreutviklet seg til noen form for konflikt. Diskusjonene har istedenfor vært positive og bidratt til at gruppen har funnet frem til bedre løsninger. Arbeidskontakten som ble opprettet i starten av prosjektet har ikke vært i bruk da det ikke har vært behov for dette ved noen anledninger.

6 Konklusjon og videre arbeid

6.1 Konklusjon

Gruppen har arbeidet mot et mål om å utvikle et brukervennlig verktøy i form av en nettside for Geomatikk AS. Verktøyet skal lede brukerne igjennom prosessen forbundet med endring av servere og loggføre den.

I arbeidet med produktet har gruppen jobbet opp mot flere prinsipper for brukervennlighet. Mot slutten av prosjektet ble det utført en brukertest med utvalgte ansatte hos Geomatikk. Testen var utformet slik at brukerne ble fortalt hva de skulle gjøre, men ikke hvordan de skulle gjøre det. Dette for å få en indikasjon på hvor intuitiv nettsiden var. For å måle brukervennligheten ble System Usability Scale (SUS)[17] brukt i siste del av brukertesten.

Ved gjennomføring av brukertesten var gruppen i rommet for å kunne observere testpersonene i aksjon. Det var derfor viktig å fokusere på å ikke veilede dem igjennom oppgavene eller svare på spørsmål som omhandlet hvordan de skulle utføre dem. Observasjonene sammen med svarene testpersonene avga viste et tydelig bilde av både positive deler av produktet og forbedringspunkter. Total scoren utarbeidet fra svarene på påstandene fra SUS endte på 89.5 poeng. Dette er godt over gjennomsnittsscoren for brukervennlighet målt med SUS som ligger på 68 poeng.

Når det gjelder krav til produktet stilt av oppdragsgiver, ble alt oppfylt. Dette inkluderer krav stilt i starten av prosjektet og krav som ble endret og utformet underveis.

Oppdragsgiver har fremvist stor grad av positivitet til resultatet av arbeidet. De har vært klare på at produktet skal tas i bruk og gruppen har i samarbeid med oppdragsgiver begynt arbeidet med å implementere det. Dette kan ses på som en bekreftelse på at gruppens inntrykk av at oppdragsgiver er godt fornøyd med det utførte arbeidet stemmer.

Med oppfyllelse av samtlige krav og en brukervennlighetscore godt over gjennomsnittet er det rimelig å si at gruppen oppnådde målet om å utvikle et brukervennlig verktøy for å loggføre prosessen forbundet med endringer på server.

6.2 Begrensninger

Webapplikasjonen er betydelig avhengig av Vuetify[21], et bibliotek som har blitt benyttet for å håndtere nesten alle komponentene i brukergrensesnittet. Selv om denne tilnærmingen er effektiv, medfører den en begrensning ettersom den ikke tilbyr samme fleksibilitet som ren CSS. Hvis det kommer en oppdatering til Vue.js[20], kan ikke applikasjonen oppdateres før Vuetify også er oppdatert.

Videre, på grunn av avhengigheten til Vuetify, er applikasjonen også begrenset til å bruke Vue.js. Dette betyr at hvis det skal utvikles en senere versjon av applikasjonen med andre rammeverk, som for eksempel React eller Angular, vil dette innebære betydelige endringer i applikasjonens utseende. Dette skyldes det faktum at Vuetify kun er kompatibelt med Vue.js.

6.3 Videre arbeid

Applikasjonen skal implementeres ferdig på Geomatikk sin server. Dermed vil ansatte ha tilgang til webapplikasjonen på egne enheter. Videre skal det holdes en presentasjon for alle utviklere i bedriften. Denne presentasjonen vil være en innføring i bruk av applikasjonen.

Det skal utvikles en ny versjon av applikasjonen. Denne versjonen har som mål å forbedre punktene fra brukertesten som hadde en lavere poengsum en forventet. Dette er punkter fra testen som skal forbedres:

- **Konsistens og standarder:** Klikkbare komponenter skal fungere konsistent gjennom hele applikasjon. Komponentene for endring, oppgave og server fungerer på tre ulike måter. Alle disse skal bli revidert for å ha et konsistent design hvor man kan trykke på komponent-tittelen for å ekspandere til en hel side. Oppgave skal kunne endres status på både gjennom dra-og-slip men også med knapper.
- **Fleksibelt og effektivitet i bruk:** For å unngå unødvendig mange klikk skal det bli implementert muligheten for å endre status på oppgaver direkte i dokumentasjonen til en endring som er klar til implementasjon. Dermed kan brukeren slippe å navigere til «My Tasks» siden.
- **Brukerkontroll og frihet:** For å sørge for høyere grad av brukerkontroll skal det implementeres ytterligere tiltak for å kunne angre på valg. F.eks. når en bruker fullfører en endring, vil det presenteres en dialog hvor man har mulighet til å fortsette eller angre.
- **Merkbar informasjon:** For å formidle ulike funksjoner på en god måte skal informasjon reformuleres. Enkelte knapper skal presiseres på en bedre måte hva gjør. F.eks. knappen «Complete» som brukes til å fullføre en endring skal omformuleres til «Complete server-change».
- **Lav fysisk anstrengelse:** Forbedre tastaturnarveier slik at dette er konsistent på hele nettsiden. I tillegg bør det prioriteres og gi mulighet til å endre oppgavers status med klikk og ikke kun med «dra og slipp» funksjonaliteten som tilbys for øyeblikket.

Denne versjonen av webapplikasjonen benytter en Azure konto som er laget av gruppen. Den neste versjon av applikasjonen skal integreres med Geomatikk sin Azure konto. Dette

kan bety at brukerrettigheter kan defineres gjennom Azure heller en internt i applikasjonen.

Samfunnspåvirkning

I dette kapitlet skal arbeidet med oppgaven ses fra et helhetlig systemperspektiv. Relevante profesjonsetiske problemstillinger vil bli trukket frem og resultatene vil bli drøftet i forhold til et miljømessig, samfunnsmessig, økonomisk og helsemessig perspektiv.

Profesjonsetiske problemstillinger

Gruppen har arbeidet med å utvikle et system som gir Geomatikk en prosedyre for prosessen ved å endre en server, samtidig som endringene som er gjort lagres.

På Geomatikk sine servere lagres blant annet data om infrastruktur i Norge, Sverige og Finland. Dette er sensitiv informasjon og sikkerhet er derfor svært sentralt for Geomatikk sin virksomhet. Det er derfor relevant å diskutere hvordan systemet gruppen har utviklet påvirker sikkerheten.

Ved å utvikle et system som lagrer informasjon om servere som igjen lagrer sensitiv informasjon kan en argumentere for at sikkerheten svekkes. Da det tidligere ikke har eksistert et slikt system vil det ved implementering av systemet finnes et nytt sensitivt punkt i Geomatikk sitt oppsett. Systemet er planlagt å kjøre lokalt noe som begrenser trusselen vesentlig. I tillegg er det gjort tilpassinger i utviklingen for å benytte lignende sikkerhetsmekanismer som Geomatikk har i sine andre systemer blant annet innlogging med Microsoft.

Når det gjelder gruppens prosess og løsning i forhold til vitenskapelighet er brukertesten relevant og drøfte. Gruppen arrangerte en brukertest for å få svar på hvor intuitiv og brukervennlig det utviklede produktet var. Dette ga gode svar. Testen var utformet på en måte som sa hva brukeren skulle teste, uten å fortelle hvordan. Dette er en styrke ved brukertesten. Etersom gruppen har tilbrakt mye tid på Geomatikk sine kontorer har de også blitt kjent med flere ansatte. En kan derfor argumenter for at noen av testpersonene kan ha svart annerledes underbevisst eller bevisst. Det å ha arrangert en brukertest med kun ukjente testpersoner kunne styrket vitenskapeligheten. Samtidig ville dette hatt tydelige svakheter da produktet har som formål å være et internt verktøy. Ved å holde testen med andre testpersoner ville produktet blitt testet på personer som ikke vil være en del av målgruppen for produktet.

Bærekraftsvurderinger

Her vil gruppens løsning settes opp mot bærekraft fra et samfunnsmessig, helsemessig, miljømessig og økonomisk ståsted.

Systemet som er utviklet skal skape større oversikt når det kommer til endringer på ser-

vere. Denne oversikten kan være avgjørende for å bidra til økt oppetid for Geomatikk sine servere. Noe som igjen vil kunne bidra til å unngå skader på infrastruktur som vannrør og strømledninger noe som har tydelige samfunnsmessige fordeler. En analyse av samfunnsøkonomiske konsekvenser ved graveskader ble i 2020 utformet av Vista Analyse AS på oppdrag for Kommunal- og moderniseringsdepartementet [3]. I denne rapporten nevnes blant annet ulemper ved forlenget arbeidet som reduksjon av fremkommelighet og økt støy som samfunnsmessige konsekvenser. Ved å redusere antall graveskader på viktig infrastruktur vil samfunnet spare betydelige økonomiske midler i direkte reparasjonskostnader, og indirekte kostnader i forhold til forsinkelse av prosjekter. En kan også se på det å unngå graveskader fra et helsemessig perspektiv. Eksempelvis kan det å grave igjennom en høyspentledning få svært alvorlige konsekvenser for arbeidere på en arbeidsplass.

Når det kommer til å se systemet opp i mot miljømessige forhold har systemet en negativ direkte konsekvens. Det vil være noe mere strømbruk knyttet til kjøring av en ny server for systemet ettersom det ikke tidligere har vært et lignende system. Denne konsekvensen vil likevel ikke utgjøre noen stor forskjell. Økt oppetid for Geomatikk sine servere kan som sagt bidra til færre graveskader. Derfor har systemet indirekte positive miljøkonsekvenser blant annet knyttet til å redusere utslipp ved reparasjon av infrastruktur og forlengede prosjekter.

Systemet gruppen har utviklet vil kun ha negativ direkte påvirkning i form av noe høyere strømforbruk. Det er likevel rimelig å si at de indirekte miljømessige, samfunnsmessige, helsemessige og økonomiske gevinstene veier opp for dette.

Referanser

- [1] Anderson, Rick; Luttin, Shaun; Roth, Daniel. (2022) Overview of ASP.NET Core. Hentet fra: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>
- [2] Cegal. (u.å) Fullstack. Hentet fra: <https://www.cegal.com/no/ordbok/fullstack>
- [3] Dalen, Dag Morten; Einarsdottir, Anita; Furuholmen, Jens; Ringdal, Herman; Strøm, Veronica; Vennemo, Haakon; (2020) Samfunnsøkonomisk analyse av konsekvensene av graveskader på infrastrukturanlegg i grunnen. Hentet fra: https://www.regjeringen.no/contentassets/6338268c2580455b980a484db2c1f316/va2020_samfunnsokonomiske_kostnader_graveskader_ledninger.pdf
- [4] Flowdigital. (2022) How to check if a site is user-friendly?. Hentet fra: <https://www.theflowdigital.com/blog/how-to-check-if-a-site-is-user-friendly>
- [5] Gannt. (u.å) What is a Gannt Chart?. Hentet fra: <https://www.gantt.com/>
- [6] Gitlab. (u.å) Hentet fra: <https://about.gitlab.com/>
- [7] Guilizzoni, Peldi. (u.å) What Are Wireframes?. Hentet fra: <https://balsamiq.com/learn/articles/what-are-wireframes/>
- [8] Nielsen, Jakob. (2020) 10 Usability Heuristics for User Interface Design. Hentet fra: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [9] Kate Moran. (2019) Usability Testing 101. Hentet fra: <https://www.nngroup.com/articles/usability-testing-101/>
- [10] Kjetil Sander. (2021) Brukergrensesnittdesign (UI – design) og brukervennlighet (usability). Hentet fra: https://estudie.no/brukervennlighet/#Hva_er_brukergrensesnittdesign_UI
- [11] NDA. (u.å) The 7 Principles. Hentet fra: <https://universaldesign.ie/what-is-universal-design/the-7-principles/>
- [12] PostgreSQL. (2023) About. Hentet fra: <https://www.postgresql.org/about/>
- [13] Prosjektveiviseren. (u.å) Begreper. Hentet fra: <https://prosjektveiviseren.digdir.no/godt-vite/begreper/81>
- [14] PST. (2023) PSTs nasjonale trusselvurdering 2023. Hentet fra: https://www.pst.no/globalassets/ntv/2023/ntv_2023_nor_web.pdf
- [15] Scrum. (u.å) What is Scrum?. Hentet fra: <https://www.scrum.org/resources/what-scrum-module>

- [16] Siddiqui, Tariq. (2021) Introduction to TypeScript and It's Features. Hentet fra: <https://www.htmlgoodies.com/javascript/introduction-to-typescript-and-its-features/>
- [17] Usability. (2023) System Usability Scale (SUS). Hentet fra: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>
- [18] UX Design Institute. (2022) UX vs. UI design: What's the difference?. Hentet fra: <https://www.uxdesigninstitute.com/blog/ux-vs-ui-design/>
- [19] Vrålstad, Emmanuelle. (2023). Brukerhistorier - hva gjorde jeg feil?. Hentet fra: <https://www.smidigbloggen.no/brukerhistorier-hva-gjorde-jeg-feil>
- [20] Vue. (u.å) Introduction. Hentet fra: <https://vuejs.org/guide/introduction.html>
- [21] Vuetify. (u.å) Introduction. Hentet fra: <https://vuetifyjs.com/en/introduction/why-vuetify/>
- [22] Jha, Shyamli. (2023) What is Microsoft Azure: How Does It Work and Services. Hentet fra: <https://www.simplilearn.com/tutorials/azure-tutorial/what-is-azure>

Vedlegg

Vedlegg A - Forprosjektplan

Vedlegg B - Visjonsdokument

Vedlegg C - Kravdokumentasjon

Vedlegg D - Systemdokumentasjon

Vedlegg E - Prosjekthåndbok

Vedlegg F - Prioritetstabell

Vedlegg G - Wireframe

Vedlegg H - Brukertest

