



Høgskulen  
på Vestlandet

# BACHELOROPPGAVE

Administrasjonsverktøy til kontroll og videreutvikling av chatbot

Administrative tool for control and further development of chatbot

**Braastad, Sander**

**Nødland, Ole Anders**

**Mardal, Matias Holmemo**

Fakultet for ingeniør- og naturvitenskap

Institutt for datateknologi, elektroteknologi og realfag

Dataingeniør og Informasjonsteknologi

Reza Arghandeh

21.05.2023

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> Administrasjonsverktøy til kontroll og videreutvikling av chatbot	<i>Dato:</i> 22.05.2023
<i>Forfattere:</i> Braastad, Sander Nødland, Ole Anders Mardal, Matias Holmemo	<i>Antall sider u/vedlegg:</i> 72
	<i>Antall vedlegg:</i> 10
<i>Studieretning:</i> Bachelor i dataingeniør (Braastad, Mardal) Bachelor i informasjonsteknologi (Nødland)	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Reza Arghandeh	<i>Gradering:</i> Ingen
<i>Merknader:</i>	

<i>Oppdragsgiver:</i> Bouvet ASA	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson:</i> Marek Vetter	<i>Telefon:</i> +47 450 01 552

<p><i>Sammendrag:</i></p> <p>Denne bacheloroppgaven ble utført av Sander, Matias og Ole Anders på oppdrag fra konsulentfirmaet Bouvet. Oppgaven gikk ut på å hjelpe Bouvets klient, Festspillene i Bergen heretter kalt FiB, med å gi bedre innsikt i og øke ytelsen til deres chatbot. FiB manglet en brukervennlig måte å vurdere hvorvidt chatboten på deres nettside var nyttig. Gruppen utviklet en løsning kalt FiB-Chatbot administrasjonsverktøy, heretter kalt FiB-CA. FiB-CA har som mål å gi FiB en tydelig oversikt over chatbotens prestasjoner, mulighet til å endre svaralternativene til chatboten og generere data som enkelt kan brukes til å forbedre den.</p> <p>Chatboten endte opp med å ha noen få mangler i slutten av prosjektet, men oppfylte majoriteten av alle kravene. Løsningen er per dags dato ikke implementert eller tatt i bruk. Når løsningsideen FiB-CA blir tatt i bruk vil dette hjelpe FiB med innsikt i prestasjonen til chatboten deres, samt gi mulighet for å forbedre prestasjonen uten tekniske kunnskaper.</p>
--

## Stikkord:

Webapplikasjon Flask	Administrasjonsverktøy Innsikt	Chatbot Maskinlæring
-------------------------	-----------------------------------	-------------------------

Høgskulen på Vestlandet, Fakultet for ingeniør- og naturvitenskap

Postadresse: Postboks 7030, 5020 BERGEN

Besøksadresse: Inndalsveien 28, Bergen

Tlf. 55 58 75 00

Fax 55 58 77 90

 E-post: [post@hvl.no](mailto:post@hvl.no)

Hjemmeside:

<http://www.hvl.no>

# Forord

Denne rapporten dokumenterer bakgrunnen og arbeidet bak bachelor-prosjektet «Administrasjonsverktøy til kontroll og videreutvikling av chatbot». Bachelorgruppen, heretter kalt gruppen, består av tre medlemmer som går to ulike bachelorprogram ved HVL. Sander Braastad og Matias Holmemo Mardal går dataingeniør med fordypning i maskinlæring og kunstig intelligens, og Ole Anders Nødland går informasjonsteknologi med fordypning i programutvikling.

# Takk til

I løpet av prosjektet har flere personer vært til stor hjelp. Takk til Bouvet og Marek Vetter som har stilt med kontor, god veiledning og muligheten til å jobbe med dette prosjektet. Vi vil også takke prosjekteieren, Festspillene i Bergen, som har vist gruppen tillit til å få jobbe med denne applikasjonen. FiB har vært behjelpelige underveis i prosessen og satt av tid til gruppen i deres hektiske arbeidshverdag. Gruppen takker også HVL som har stilt med Reza Arghandeh som veileder, og Carsten Gunnar Helgesen som har bidratt med ekstra veiledning på teksten. Veilederne fra HVL har hjulpet gruppen med å holde rett fokus underveis i prosjektet, samt hvordan vi best kunne bygge opp oppgaven. Til slutt vil gruppen gi en takk til alle andre som har vært behjelpelige med å gi tilbakemeldinger på teksten.

# Innholdsfortegnelse

## Contents

Forord.....	iii
Takk til.....	iii
Innholdsfortegnelse .....	iv
Begreper.....	vii
Forkortelser.....	viii
1  Innledning .....	1
1.1 Kontekst .....	1
1.2 Motivasjon .....	2
1.3 Prosjekteier og oppdragsgiver .....	2
1.3.1 Prosjekteier .....	2
1.3.2 Oppdragsgiver .....	3
1.4 Problembeskrivelse og mål .....	3
1.4.1 Problembeskrivelse.....	3
1.4.2 Mål og forskningsspørsmål .....	3
1.5 Oppbygging av rapporten .....	5
2  Prosjektbeskrivelse .....	7
2.1 Praktisk bakgrunn .....	7
2.1.1 Tidligere arbeid .....	7
2.1.2 Initielle krav.....	7
2.1.3 Initiell løsnings-idé .....	8
2.2 Avgrensninger .....	8
2.3 Ressurser.....	9
2.4 Litteratur om problemstillingen.....	10
3  Analyse og tilnærming .....	13
3.1  Analyse av krav .....	13
3.1.1 Forarbeid.....	13
3.1.2  Krav .....	13
3.2 Forslag til løsning .....	15
3.2.1 Webbasert app VS. desktop app.....	15
3.2.2 Rammeverk .....	15
3.2.3 Design.....	16

3.2.4	Database .....	16
3.3	Valgt løsning.....	16
3.4	Valg av verktøy.....	17
3.4.1	Visual Studio Code .....	17
3.4.2	Flask .....	17
3.4.3	Figma.....	17
3.5	Valg av hjelpemidler.....	18
3.5.1	GitHub .....	18
3.5.2	Verktøy for skriving.....	18
3.5.3	Discord .....	19
3.5.4	Trello .....	19
3.5.5	Bootstrap.....	19
3.5.6	Chat GPT.....	19
3.6	Prosjektmetodikk .....	20
3.6.1	Utviklingsmetodikk .....	20
3.6.2	Prosjektplan .....	21
3.6.3	Risikovurdering .....	23
3.7	Evalueringsplan .....	23
3.7.1	Brukertesting.....	23
3.7.2	Enhetstesting .....	25
4	Design og arkitektur / Detaljert løsning.....	27
4.1	Systemarkitektur.....	27
4.1.1	Kommunikasjon med chatbot.....	28
4.1.2	Datalag .....	28
4.1.3	Chatbotens logikklag.....	29
4.2	Design og oppsett .....	30
4.2.1	Hjem.....	30
4.2.2	Statistikk.....	32
4.2.3	Administrer svaralternativer .....	36
4.2.4	Tren .....	39
4.2.5	Logg inn og Registrer.....	42
4.2.6	Ekstra funksjoner .....	43
5	Evaluering og resultat .....	45
5.1	Evalueringsmetode .....	45

5.1.1	Brukertesting.....	45
5.1.2	Enhetstesting .....	46
5.2	Evalueringsresultat .....	46
5.2.1	Brukertester .....	46
5.2.2	Enhetstesting .....	47
5.3	Prosjektresultat.....	48
5.4	Prosjektgjennomføring .....	49
6	Diskusjon.....	51
6.1	Funksjonalitet og resultat .....	51
6.1.1	Valg.....	51
6.1.2	Problemer .....	54
6.1.3	Resultat .....	55
6.2	Erfaringer .....	56
7	Konklusjon og videre arbeid .....	57
7.1	Konklusjon.....	57
7.2	Videre arbeid.....	57
7.3	Generelle læringspunkter .....	59
8	Litteraturliste / referanser .....	61
	Referanser .....	61
9	Vedlegg.....	63
9.1	Vedleggliste.....	63
	Støttedokumenter .....	63
	Samarbeidsavtale.....	63
	Brukertesting.....	63
	Problembeskrivelse og sluttevaluering.....	63
9.2	Figurliste.....	63
9.3	Tabelliste.....	64

## Begreper

**API** – «Et API kjennetegnes ved at man sender enkle forespørsler til et eksternt system, som så utfører en handling og gir en respons tilbake.» (Rossen & Nätt, 2022).

**Chatbot** – Et dataprogram som svarer på spørsmål stilt av en bruker.

**CSS** – «Cascading style sheets» er et programmeringsspråk for å strukturere og fikse på web-sider (HTML eller XML).

**Data Cleaning** – Prosessen med å reparere eller fjerne feilaktig, ødelagt, feilformatert, duplikat, eller ufullstendig data innenfor et datasett.

**Discord** – Kommunikasjonsplattform.

**Django** – Et rammeverk i programmeringsspråket Python for utvikling av webbaserte applikasjoner.

**Enum** – Kort for enumerations, er en liste med faste verdier knyttet opp mot tall, som hjelper med lesbarhet og konsistens av koden.

**F1-score** – En metrisk verdi som balanserer presisjon (hvor mange valgte elementer er relevante) og gjenfinning (hvor mange relevante elementer blir valgt).

**Figma** – Et skybasert design- og prototypeverktøy som lar brukere samarbeide i sanntid.

**Flask** – Et rammeverk i programmeringsspråket Python for utvikling av webbaserte applikasjoner.

**Fremmednøkkel** – Referanse til en annen tabell i en SQL-database.

**Front-end** – Koden som beskriver det visuelle som er synlig i for brukeren av programmet.

**GANTT-Diagram** – En visuell representasjon som brukes til å planlegge prosjekter ved å vise tidslinjen for oppgaver og deres avhengigheter.

**GitHub** – Plattform for utvikling og versjonskontroll av programvare.

**HTML** – Standard markup-språk for websider.

**Iterasjon** – En iterasjon i sammenheng med iterasjonsbasert utvikling refererer til en gjentakende og tidsavgrenset arbeidsperiode.

**JavaScript** – Programmeringsspråk ofte brukt for web-utvikling.

**NLP** – «Natural language processing» refererer til en gren innenfor kunstig intelligens som fokuserer på å gi datamaskiner muligheten til å forstå betydningen av tekst og tale.

**Open source** – Kildekode som er åpent tilgjengelig for alle til å endre på, kopiere eller ta inspirasjon fra.

**Primærnøkkel** – Identifikator til en tabell i en SQL-database.

**Python** – Programmeringsspråk.

**Ruby on rails** – Et rammeverk i programmeringsspråket Ruby for utvikling av webbaserte applikasjoner.

**Ruby** – Programmeringsspråk.

**Scrum** – Smidig utviklingsmetode.

**Structured Query Language** – Et programmeringsspråk brukt for å håndtere relasjonsdatabaser.

**Teams** – Samarbeidsplattform fra microsoft.

**Trello** – Oppslagstavle for å visualisere Scrum.

**URL** – Adressen til en ressurs på internett, som for eksempel en nettside, fil eller bilde.

**Voice chat** – Stemmechat, en kanal der en kan snakke med hverandre digitalt.

**Wireframe** – En enkel visuell skisse eller modell som brukes for å representere den grunnleggende strukturen og layouten til en nettside, applikasjon eller annet digitalt grensesnitt.

## Forkortelser

Forkortelse	Ord
AI	Artificial intelligence
API	Application programming interface
CSS	Cascading Style Sheets
Enum	Enumerations
FAQ	Frequently Asked Questions (Ofte stilte spørsmål)
FiB	Festspillene i Bergen
FiB-CA	Festspillene i Bergen Chatbot Administrasjonsverktøy
FK	Foregin key (fremmednøkkel)
HTML	Hypertext Markup Language
HVL	Høgskulen på Vestlandet
NLP	Natural Language Processing
PK	Primary key (primærnøkkel)
SQL	Structured Query Language
u.d.	Uten dato
UI	User interface
URL	Uniform Resource Locator
UX	User experience
VS Code	Visual Studio Code



# 1 Innledning

## 1.1 Kontekst

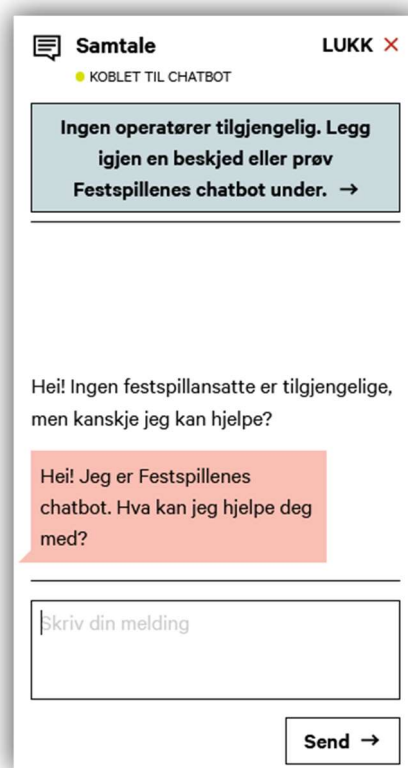
Høsten 2019 samarbeidet Festspillene i Bergen (FiB) med konsulentfirmaet Bouvet for å utvikle en chatbot som en del av deres kundeservicestrategi. Denne chatboten, som er en enkel FAQ-løsning, er designet for å veilede besøkende til relevante seksjoner på FiBs nettsted, fib.no. Ved å supplere den eksisterende kundeservicen, var formålet med chatboten å redusere belastningen på kundeservicepersonell og å tilby enkel hjelp utenom FiBs åpningstider. Dette initiativet var en del av FiBs ønske om å utforske og implementere ny teknologi.

FiB sin chatbot fungerer ved at den mottar et spørsmål via en chat på en nettside, hvorpå den genererer et passende svar. Chatboten deres er programmert til å gi det best passende svaret ut av et sett av 15 forhåndsbestemte svar. For at chatboten skal kunne gi korrekt svar, må den trenes på å forstå hvilke spørsmål som korrelerer med hvilke svar. Dette blir gjort ved å mate chatboten med treningsdata som består av forskjellige spørsmål, hver med et tilknyttet korrekt svar. Over tid lærer chatboten å gjenkjenne mønstre i spørsmålene og gir det tilsvarende forhåndsbestemte svaret.

Etter lanseringen av chatboten, ble det imidlertid tydelig at det var betydelige begrensninger i dens funksjonalitet, spesielt når det gjaldt tilgang og innsikt for FiBs webredaktører. Webredaktørene kunne bare redigere de 15 forhåndsbestemte svarene. De hadde ikke mulighet til å se innkommende spørsmål, vurdere svarene gitt av chatboten, eller trekke ut statistikk. Videre trening av chatboten viste seg også å være en manuell og tidkrevende prosess.

I sluttevalueringen fra FiB, vedlegg 9.1.9 *FiB Sluttevaluering*, blir det beskrevet at flere i FiB begynte å stille spørsmål ved chatbotens nytteverdi da det ble klart at chatboten ikke kunne forbedres eller enkelt tilpasses. Dette ble forsterket av tilbakemelding fra frustrerte kunder som ble sendt i sirkler av chatboten. Til tross for muligheten for å hente alle spørsmålene chatboten hadde mottatt med tilhørende svar i en lang liste, og muligheten til å endre svarteksten til hvert av de forhåndsdefinerte svarene, blir disse funksjonene sjelden brukt på grunn av deres kompleksitet.

Med disse begrensningene i bakhodet var det usikkert om chatboten oppfylte sitt opprinnelige formål, og FiB så derfor på mulige veier videre for å forbedre denne teknologien.



Figur 1.1 Skjerm bilde av chatboten til FiB

## 1.2 Motivasjon

Under videreutvikling av chatboten høsten 2022 avholdt Bouvet et kundemøte med FiB. Under dette møtet ble motivasjonen for denne oppgaven etablert. FiB uttrykte et ønske om mer kontroll over chatboten. Dette var som følge av at de da hadde begrenset mulighet til å endre på chatboten og ingen god måte å få oversikt over hvilke spørsmål den blir stilt. Uthenting av data fra chatboten er i tillegg en tungvint prosess.

Gjennom en manuell gjennomgang av chatbotens data ble det oppdaget spørsmål fra frustrerte kunder som etter gjentatte spørsmål ikke fikk svaret de så etter. Dette illustrerer behovet for forbedring. Prosessen for å trene chatboten på nytt er ikke ideal, da de ikke kan gjøre dette selv, men må leie inn Bouvet for å gjøre det.

FiB ønsket en løsning som gjør det enklere for dem å analysere innkommende spørsmål og svar gitt, samt forbedre prestasjonen til chatboten. For øyeblikket vet de ikke hvilke spørsmål som kommer inn og hvilke svar besøkende får, så det er usikkert om chatboten er til hjelp for besøkende eller ikke. Ved å få oversikt over hva besøkende lurte på kan de i tillegg tilpasse nettsiden til å gi svar på spørsmålene de ikke viste brukere hadde.

En detaljert liste over ønskene og behovene til FiB er listet i vedlegg 9.1.8 *Problembeskrivelse fra Bouvet*. Denne oppgaven sikter mot å hjelpe FiB med å oppfylle ønskene deres gjennom å utvikle et nytt administrasjonsverktøy.

## 1.3 Prosjekteier og oppdragsgiver

### 1.3.1 Prosjekteier

“Festspillene i Bergen ble etablert i 1953 og er flaggskipet blant Nordens festivaler for musikk og scenekunst.” (Festspillene i Bergen, u.d.). FiB har hundrevis av arrangementer årlig i en tidsperiode fra mai til juni (Festspillene i Bergen, u.d.).

FiB mottar hundrevis av spørsmål i tidsrommet festspillene pågår. For å avlaste kundeservice, samt å øke brukervennligheten, har det blitt tatt i bruk en chatbot til å svare på de enkleste spørsmålene.

### 1.3.2 Oppdragsgiver

Bouvet er et norsk konsulentfirma som ble opprettet i 2002 og har per mai 2023 over 2000 ansatte fordelt på 17 kontorer, i både Norge og Sverige. Bouvet har blant annet kontor i Bergen med 170 ansatte. Bouvet er et stort IT-selskap og har opparbeidet seg kompetanse innen flere fagfelt som blant annet systemutvikling, design og maskinlæring. (Bouvet, u.d.)



Figur 1.2 Bouvet sitt kontor i Bergen

I dag har Bouvet FiB som kunde, og er dermed ansvarlig for å drifte og utvikle nettsidene til FiB. Det er gjennom dette kundeforholdet at bachelorprosjektet får jobbe med å utvikle et administrasjonsverktøy for FiB sin chatbot.

## 1.4 Problembeskrivelse og mål

### 1.4.1 Problembeskrivelse

I dagens teknologidrevne verden er chatboter en viktig ressurs for mange organisasjoner, inkludert FiB. For øyeblikket har FiB ingen enkel og brukervennlig løsning for å administrere og tilpasse chatboten sin. Dette skaper en utfordring, ettersom de ikke har mulighet til å se detaljer som hvilke spørsmål som blir stilt til chatboten og hvordan den svarer, noe som hindrer effektiv administrasjon og tilpasning av chatboten. Uten tilgang til denne informasjonen, er det vanskelig for FiB å vite om brukerne av chatboten faktisk får gode svar på sine spørsmål.

### 1.4.2 Mål og forskningsspørsmål

For at det skal bli enklere for ikke-tekniske brukere å arbeide med, og forstå chatboten, vil dette prosjektet ha som mål å utvikle og designe et intuitivt og brukervennlig administrasjonsverktøy for FiB sin chatbot. Videre vil verktøyet gi mulighet for å vise innkommende spørsmål og svar, redigere svaralternativer og presentere relevant statistikk over spørsmål, svar og ytelse. Dette vil bidra til å sikre best mulig brukeropplevelse og effektivitet i administrasjonen av chatboten.

Kravene for administrasjonsverktøyet er listet under kapittel 3.1.2. Disse innebærer blant annet at det skal være mulig for en bruker av programmet å se hvilke spørsmål som har kommet inn til chatboten og hvilke svar den har gitt, samt mulighet til å validere spørsmål ved å avmerke dem med riktig svar. Chatboten skal kunne forbedres ved bruk av de validerte spørsmålene. Det skal også være

mulig å redigere svaralternativene til chatboten. Ut ifra dette har gruppen satt følgende forskningsspørsmål:

**Hvordan kan gruppen utvikle og designe et brukervennlig administrasjonsverktøy for FiB sin chatbot som er enkelt for ikke-tekniske brukere å forstå, med mulighet for å vise innkommende spørsmål og svar, spore ytelse og redigere svaralternativer?**

For å besvare forskningsspørsmålet og oppnå det overordnede målet med å utvikle et administrasjonsverktøy for FiB sin chatbot som kan brukes av ansatte i FiB, har det blitt identifisert fem delmål for prosjektet. Delmålene vil være veiledende for utviklingen av administrasjonsverktøyet og sikre at det imøtekommer de viktigste behovene til ansatte i FiB.

Det overordnede målet er delt opp i følgende delmål, rangert etter prioritet fra høyest til lavest:

- 1 Det skal være mulig å vise spørsmål som er stilt til chatboten og hva den svarte.
- 2 Det skal være mulig å validere riktig svar på et innsendt spørsmål.
- 3 Chatboten skal kunne trenes på nye spørsmål som er merket som riktig i verktøyet.
- 4 Det skal være mulig å spore prestasjonen til chatboten.
- 5 Det skal være mulig å endre og legge til svaralternativer for chatboten.

## 1.5 Oppbygging av rapporten

Denne rapporten beskriver løsningen gruppen produserte i semesteret V2023. Rapporten er inndelt i 9 kapitler med ulike fokusområder.

**Kapittel 1 Innledning:** I Innledningen blir konteksten til og motivasjonen for prosjektet presentert, der problembeskrivelsen og målet definert ut fra dette.

**Kapittel 2 Prosjektbeskrivelse:** Inneholder en detaljert beskrivelse av prosjektet. Her er det en introduksjon for grunnlaget dette prosjektet ble bygget på, initielle krav til løsningen og det initielle løsningsforslaget, samt hvilke avgrensninger og ressurser som gjelder for prosjektet.

**Kapittel 3 Analyse og tilnærming:** Inneholder diskusjon rundt valg av løsninger og teknologi, og en gjennomgang av arbeidsmetode og evalueringsplan. Her blir de endelige kravene for prosjektet avdekket med hjelp av prototyping, og valgene som ledet til den endelige løsningsideen drøftet.

**Kapittel 4 Design og arkitektur:** Inneholder beskrivelse av produktdesign, programvarearkitektur, teknologibruk, samt sammenligning av endelig løsning og prototype.

**Kapittel 5 Evaluering og resultat:** Utdyper hvordan resultatet blir evaluert og hva evalueringsresultatet er. I dette kapitlet blir det også gjennomgått til hvilke grad hvert av delmålene fra kapittel 1 og kravene fra kapittel 3 ble oppfylt.

**Kapittel 6 Diskusjon:** Inneholder diskusjon rundt prosjektet og resultatet. Her blir det drøftet fordeler og ulemper bak valg av design, verktøy, utviklingsmetodikk og erfaringer som ble gjort gjennom prosjektet. I tillegg blir resultatet også drøftet her.

**Kapittel 7 Konklusjon og videre arbeid:** Inneholder konklusjon for prosjektet og muligheter for videre arbeid. Det ble beskrevet hvordan løsningen burde videreutvikles, samt drøftet hvem andre som kunne bruke prinsipper og lærdom fra dette prosjektet.

**Kapittel 8 Litteraturliste / referanser:** Inneholder en liste over referanser som har blitt brukt i rapporten.

**Kapittel 9 Vedlegg:** Inneholder vedlegg til prosjektet som omhandler: Prosjekthåndbok, visjonsdokument, kravdokument, systemdokumentasjon, samarbeidsavtale, brukertester og skriv fra Bouvet og FiB.



## 2 Prosjektbeskrivelse

### 2.1 Praktisk bakgrunn

#### 2.1.1 Tidligere arbeid

Som nevnt i 1.1 kontekst så har Bouvet utviklet en chatbot for FiB, med både brukergrensesnitt for brukeren og ferdig trent modell på innsamlede og genererte spørsmål. I tillegg blir spørsmålene som stilles til chatboten lagret i en database.

Et av medlemmene i gruppen har gjort tidligere arbeid hos FiB som utplasseringsstudent fra HVL. Da jobbet han med å lage et script som kunne produsere maskinlærings-modeller egnet for FiB sin chatbot. Dette arbeidet er til dags dato ikke tatt i bruk, men kan benyttes sammen med treningsdata for å forbedre chatboten.

Av tidligere arbeid ble det delt en tabell med spørsmål stilt til chatboten, og svarene til chatboten. Denne tabellen inneholdt ingen fremmednøkler eller indikasjoner på at chatboten hadde flere tabeller i databasen sin. Det kan ikke fastslå strukturen med sikkerhet, men det er antatt at den gamle databasen ser ut som i figur 4.2. I tillegg ble det gitt tilgang til et git-repository som er en oppbevaringsplass for kode og nødvendig dokumentasjon.

#### 2.1.2 Initielle krav

De initielle kravene til FiB ble mottatt i form av en e-post fra gruppens kontaktperson i Bouvet, Marek. Kravene ble avdekket av Marek Vetter under et kundemøte med FiB og videreformidlet til gruppen da oppgaven for bacheloren ble presentert. Dette er en oppsummert liste over kravene:

- Et program som er enkelt for mindre tekniske brukere å forstå.
- Kunne generere ukes- og dags-rapport med spørsmål og svar.
- Kunne endre navnet og svaret til svarresponsene.
- Få en liste over spørsmål der chatboten var usikker på sitt svar.
- Mulighet for FiB å gi en tilbakemelding på chatbotsvarene.
- Mulighet for FiB å kunne merke riktig svar til et spørsmål der chatboten har svart feil.
- Mulighet for å vise statistikk angående chatboten på en enkel måte.

For en mer detaljert oversikt over innholdet, se vedlegg 9.1.8 *Problembeskrivelse fra Bouvet*.

### 2.1.3 Initiell løsnings-idé

Den initielle planen for prosjektet var å lage et program som ga brukeren full oversikt over hvilke spørsmål chatboten fikk, og hva den svarte. FiB hadde også flere andre ønsker og behov for applikasjonen. Disse ønskene er beskrevet i kapittel 2.1.2 Initielle krav.

Det var viktig for gruppen å komme opp med en løsning for FiB som de kunne ta i bruk, og ha nytte av etter at prosjektet var ferdig. På grunnlag av dette ønsket gruppen å lage et program som var i første omgang funksjonelt med kun de viktigste kravene, og som inkluderer funksjonalitet til de gjenværende kravene trinnvis. Dette var for å sikre at produktet kunne tas i bruk så fort som mulig, og at det var funksjonelt gjennom hele utviklingsprosessen.

Ut fra vedlegg 9.1.8 *Problembeskrivelse fra Bouvet*, ble det besluttet å fokusere på enkelte ønsker for løsningsforslaget som skulle presenteres for FiB. Dette innebar blant annet å kunne navigere stilte spørsmål og svar fra chatboten, å navigere og endre svaralternativer, å kunne se statistikk for chatboten, og å trene chatboten på ny data.

## 2.2 Avgrensninger

Resultatet av dette prosjektet vil være avgrenset med tanke på tid og ressurser tilgjengelig for å få til en god løsning. Dette prosjektet skal gjennomføres på et halvt år. Som et resultat av dette er det tvilsomt at produktet vil kunne oppfylle all funksjonalitet ønsket av prosjekteier. Dette løses med en tydelig prioritetsliste som ble utarbeidet med FiB i første brukertest. Mer info om brukertestene finnes under kapittel 5.2.1. Prioritetslisten ligger i vedlegg 9.1.2 *Visjonsdokument*.

FiB har de årlige arrangementene sine fra mai til juni og vil derfor ikke ha tid til å ta i bruk løsningen i dette tidsrommet. For å gi gruppen så god tid som mulig til å ferdigstille oppgaven ble det derfor bestemt at løsningen ikke skulle implementeres før etter dette. Derfor vil dette prosjektet være avgrenset til utvikling av løsningen, og ikke implementering.

Produktet er ment for FiB-ansatte, og vil bli spesielt tilpasset dem. Brukergruppen er liten, og skal motta opplæring til bruk av produktet. Alle brukere vil derfor vite hvordan applikasjonen skal brukes. Dette vil føre til en lavere risiko for at løsningen ikke blir brukt som intensjonelt. Det vil derfor ikke, i like stor grad, være nødvendig å legge inn advarsler og sperringer på funksjonalitet.

Produktet er planlagt å bli brukt på arbeidsmaskiner hos FiB, så ett dynamisk brukergrensesnitt er ikke et behov. At brukergrensesnittet tilpasser seg flere forskjellige typer skjerm-oppløsninger er ønskelig, men vil ikke være en prioritet under utviklingen.



## 2.3 Ressurser

Det er behov for flere ressurser for at prosjektet skal lykkes. De viktigste ressursene for vår gjennomføring er den opprinnelige prosjekt-koden til chatboten og databasen der innsendte spørsmål til chatboten blir lagret.

Den opprinnelige prosjekt-koden til chatboten er viktig ettersom tilgang til prosjekt-koden gir innsikt i hvordan chatboten fungerer, forståelse for hvordan den kan bli trent på ny data, og informasjon om hvordan chatboten lagrer informasjonen. Det er også en mulighet for at små omskrivninger må gjøres på chatbot-programmet for at den bedre skal kommunisere med den nye applikasjonen.

Tilgang til chatbotens database er essensielt ettersom hovedhensikten med applikasjonen er å kunne se og validere spørsmålene som lagres der, og deretter trene chatboten på ny validert data, slik at den blir mer treffsikker.

Både programvaren til chatboten, og databasen er eid av FiB, men er i all hovedsak håndtert av ansatte fra Bouvet. På grunnlag av dette må gruppen forholde seg til Bouvet angående tilgang til dataen. Bouvet tillater ikke håndtering av kundedata på personlige PC-er. Derfor er det behov for klientdatamaskiner. Uten disse vil ikke sikkerhetskravene til Bouvet oppfylles, og de andre nødvendige ressursene for prosjektet vil ikke være tilgjengelig. Gruppen mottok en laptop hver, fra Bouvet, som oppfylte alle sikkerhetskravene deres. Disse pc-ene fikk gruppen ta med seg og bruke der det passet dem best.

Av personer som har vært en ressurs til prosjektet nevnes blant annet Reza Arghandeh og Marek Vetter. Reza Arghandeh fra HVL har veiledet gruppen gjennom bachelorperioden slik at sluttproduktet samsvarer med kravene til bacheloroppgaven. Gruppen har også hatt en kontaktperson fra Bouvet, Marek Vetter, som har hjulpet gruppen med diverse oppgaver som jobb-pc, kontorplass, hjelp til programmering og veiledning til rapportskrivning.

På HVL sin campus i Bergen, har grupperom og arbeidsplasser blitt brukt til å arbeide med oppgaven, i tillegg har gruppen fått jobbet en dag i uken på kontoret til Bouvet. Gruppen var også i kontakt med faglærer Carsten Gunnar Helgesen for tilbakemelding på språk, struktur og eventuelle spørsmål. Gruppen har også fått hjelp av FiB i form av tilbakemeldinger på prototype og produkt slik at gruppen kan levere det produktet som er forventet.

For å få veiledning og hjelp til koding av applikasjonen og til skrivingen av rapporten, ble språkmodellen ChatGPT brukt. Språkmodellen kan hjelpe med å finne feil i kode eller tekst, og den kan komme med forslag om forbedringer. Språkmodellen hjelper også med å forstå feilmeldinger som oppstår når en prøver å kjøre kode hvor det finnes feil. En mer detaljert beskrivelse finnes under kapittel 3.5.6 ChatGPT.

Det ble brukt flere programmer gjennom dette prosjektet. Gruppen brukte Flask rammeverk for Python og Visual Studio Code som kode-editor. For deling av filer ble det brukt Microsoft Teams og Github. Valget av disse verktøyene og hjelpemidler blir gjennomgått i kapittel 3.4 Valg av verktøy og kapittel 3.5 Valg av hjelpemidler.

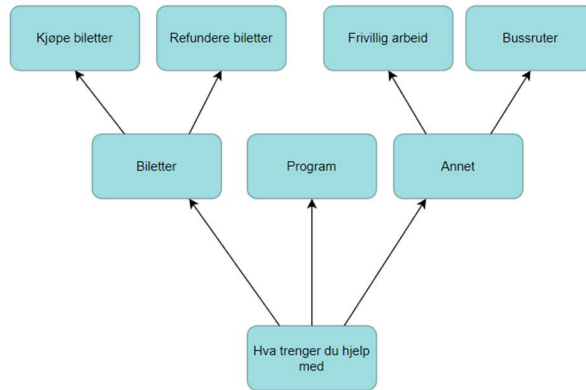
## 2.4 Litteratur om problemstillingen

Maskinlæringsmodellen til en chatbot kan ansees som «hjernen» til chatboten. Det er denne delen som avgjør i hvilken kategori et spørsmål hører hjemme. Maskinlæringsmodellen som blir brukt i chatboten til FiB er en klassifiseringsmodell. En klassifiseringsmodell tar en input, her spørsmålet, og kategoriserer dette. Kategorien spørsmålet blir plassert i, peker til et av flere ferdigdefinerte svar. For å kunne lage en maskinlæringsmodell må den først bli trent på treningsdata. Treningsdataen består av inputdata og tilhørende kategori, altså mange spørsmål med et rett svar. Ved hjelp av treningsdataen klarer modellen å finne mønster i inputdataen. Med større mengde treningsdata blir modellen flinkere til å gjenkjenne mønster og velge rett kategori (Géron, 2019, ss. 2-8).

Det er viktig å fortsette å overvåke en slik modell etter den er blitt tatt i bruk ettersom prestasjon for en slik modell kan variere. Variasjonen i prestasjon kan ha mange årsaker, men skyldes ofte at treningsdata er utdatert eller mangler datapunkt for nye problemstillinger for modellen (Géron, 2019, ss. 80-81). I tillegg til dette er det også viktig å se at de forhåndsdefinerte svarene for hver kategori fortsatt er relevant. Dette kan for eksempel bli et problem når svaret for årets program peker til programmet for 2023. Dette er riktig for øyeblikket, men vil bli en feil respons til neste år.

For å lage en velfungerende maskinlæringsmodell er et stort treningsdatasett essensielt. FiB sitter i dag med et treningsdatasett på over 4000 datapunkt, men det er uklart hvor mange av datapunktene dagens modell er trent på. Marek Vetter fra Bouvet var ikke en del av utviklingsteamet for denne chatboten, men han antar at chatboten sist ble trent i 2019 på kun 1400 datapunkt. Både 1400 og 4000 datapunkt anses som små datasett, og det er derfor vanskelig å trene en velfungerende maskinlæringsmodell på denne dataen. Modellen ligger også under risiko for å være trent på mindre relevant data ettersom chatboten ikke er blitt trent på nytt på en god stund.

Rensing av data, eller «data cleaning», er et sentralt ledd i maskinlæringsprosessen. Datarensing handler om å forbedre og oppdatere dataene som modellen er trent på, for å sikre at den lærer mest mulig relevant og korrekt informasjon. I boken «Deep Learning for Coders with fastai & PyTorch» uttaler en av forfatterne: «Cleaning the data and getting it ready for your model are two of the biggest challenges for data scientists; they say it takes 90% of their time.» (Howard & Gugger, 2020) Dette understreker viktigheten av å kontinuerlig evaluere og oppdatere modellen med nye data. Dersom en stopper med å legge til ny og/eller forbedret data, vil modellen fortsette å arbeide basert på den gamle informasjonen den har lært, noe som kan føre til feilaktige eller irrelevante svar. FiB sin chatbot får kontinuerlig inn ny data i form av spørsmål, og det vil derfor være avgjørende at denne dataen blir rensset for å sikre at chatboten kan forbedre seg og gir mer presise svar. Selv med et relativt lite datasett på 4000 datapunkter kan rensing og oppdatering av dataene gi betydelige forbedringer av modellens ytelse og nøyaktighet.



*Figur 2.1 Eksempel på trestruktur*

Konkurrerende løsninger som Chatbot.com og Intercom.com har valgt å lage chatboter basert på en trestruktur. I figur 2.1 vises en illustrasjon over hvordan en trestruktur kan brukes for en chatbot. I en slik trestruktur er det mulig å klikke seg innover i treet og innsnevre hva brukeren trenger hjelp med. En fordel med å tilby denne typen chatbot er at det ikke er behov for treningsdata til en maskinlæringsmodell. En chatbot bygget på en trestruktur som denne vil ikke uten en klassifiseringsmodell eller en annen metode for å fastslå hvilken kategori teksten tilhører, kunne ta inn spørsmål i form av fri tekst. Chatbot.com og Intercom.com tilbyr også chatbot-er basert på maskinlæringsmodeller, men dette krever treningsdata. (ChatBot | AI Chat Bot Software for Your Website, u.d.) (Intercom | Making Internet Business Personal, u.d.)

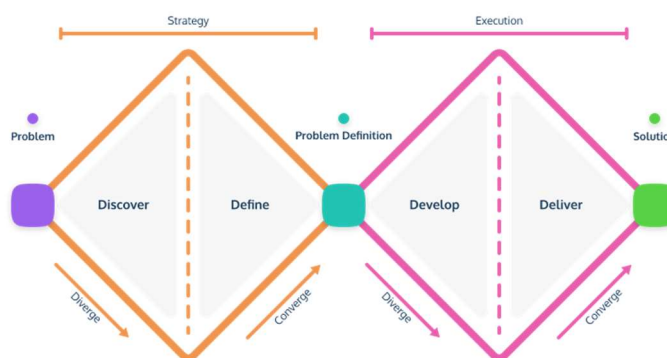


## 3 Analyse og tilnærming

### 3.1 Analyse av krav

#### 3.1.1 Forarbeid

I henhold til «the double diamond process» ble det ble brukt mye tid i begynnelsen av prosjektet til å avdekke behov, definere problemstillingen og planlegge løsningen i forkant av utviklingen. Et godt grunnarbeid vil resultere i mindre sannsynlighet for at det må gjøres omfattende endringer utover i prosjektet. Endringer som blir avdekket sent kan bli svært tidkrevende.



Figur 3.1 Double Diamond Process (How ProdPad Fits With Product Discovery, u.d.)

For å samkjøre gruppens visjon for prosjektet med FiB sin visjon, ble prototyping brukt. Det gjorde at gruppen kunne visualisere ideene sine, og dermed få klargjort og validert kravene til FiB, samt gitt dem muligheten for å sette seg inn i vår løsningsside. Dette reduserer risikoen for at prosjektet tar feil retning. (Interaction Design Foundation, u.d.) Prototypen ble laget med hensyn til de initielle kravene, og er utarbeidet i Figma. For mer informasjon om Figma, se kapittel 3.4.3 Figma, og under 5.2.1 brukertester, blir fremvisningen av prototypen beskrevet nøyere.

#### 3.1.2 Krav

Det ble laget et kravdokument, vedlegg 9.1.3 *Kravdokument*, for dette prosjektet som beskriver de funksjonelle kravene, og er utarbeidet fra prototypen. Kravdokumentet inneholder brukstilfellemodeller som visuelt viser mulighetene i applikasjonen, samt detaljerte tekstlige beskrivelser for de ulike brukstilfellene. Kapittel 2 i kravdokumentet viser domenemodellen for applikasjonen som illustrerer hvordan applikasjonen skal henge sammen med eksisterende og nytt system. Dokumentet har også bilder av de ulike sidene fra prototypen, der kravene for sidene blir beskrevet.

Kravene samsvarer i stor grad med det overordnede målet, og delmålene fra kapittel 1.4.2 Mål og forskningsspørsmål. I tillegg har FiB under behovsavdekking vist stor forståelse for usikkerheten bak hvor mange av de nye kravene som var gjennomførbare innen den begrensede tidsrammen for prosjektet.

Under kan en se minimumskravene for applikasjonen. Disse kravene beskriver funksjonaliteten som ble best mottatt og høyest prioritert under den første brukertesten hos FiB. Denne brukertesten blir beskrevet nøyere i kapittel 5.2.1 Brukertester. Kravene under er blitt avdekket som de største behovene til FiB, og er noe gruppen var komfortabel med å formidle som gjennomførbart i forkant

av prosjektstart. Disse punktene gjenspeiles også i **delmål 1-3** under kapittel 1.4.2 Mål og forskningsspørsmål.

**Krav 1:** Få en oversikt over hvilke spørsmål som er stilt til chatboten og om hvilket svar chatboten har gitt.

**Krav 2:** Kunne gå over og rette opp i tilfeller der chatboten har gitt feil svar.

**Krav 3:** Kunne ta med de nye spørsmålene ved trening av ny modell til chatboten.

I tillegg til **krav 1** til **3**, er underliggende krav også ansett som gjennomførbare innenfor tidsrammen, men med mer usikkerhet. Dette er krav som ble satt for produktet for å kunne anse prosjektet som vellykket. Disse kravene faller inn under **delmål 4-5**.

**Krav 4:** Kunne endre på de svaralternativene som chatboten gir.

**Krav 5:** Kunne se statistikk over hvor bra chatboten presterer, og hvor mange spørsmål hver kategori har fått.

**Krav 6:** Kunne sammenlikne år for år, og se om chatboten klarer å prestere bedre over tid.

Punktene under ansees i større grad som ønsker enn krav. Dette er funksjoner gruppen avdekket sammen med FiB, og som kunne være ønskelig for applikasjonen deres, men som ikke nødvendigvis var gjennomførbart under den begrensede tidsrammen for prosjektet. **Krav 7** og **8** faller delvis inn under **delmål 5** på tross av usikkerhet bak gjennomførbarhet.

**Krav 7:** Kunne lage nye svaralternativer som fungerer likt som de som allerede er i bruk.

**Krav 8:** Kunne lage svaralternativer som bruker stikkord og ikke chatbot modellen.

**Krav 9:** Kunne logge hendelser i applikasjonen.

**Krav 10:** Kunne gi brukeren av chatboten mulighet til å gi tilbakemelding på om svaret var til hjelp eller ikke.

For tydeligere prioritering og beskrivelse av krav, se kapittel 3.4 i vedlegg 9.1.2 *Visjonsdokument*.

Under første brukertest ble det avklart at disse kravene var en bedre representasjon av ønskene til FiB enn de initielle kravene under kapittel 2.1.2 Initielle krav. I de initielle kravene var det blant annet et ønske om å få en dagsrapport med spørsmål og svar. Dette har i stedet blitt løst ved å gi en generell statistikk, (**krav 5 og 6**) og å vise nye spørsmål som har kommet inn i en liste (**krav 1**). FiB ønsket også en liste over spørsmål hvor chatboten var usikker på svaret. Dette ble ikke prioritert som et krav ettersom det enkelt kunne inkluderes med **krav 1** som en filtreringsmulighet. En siste funksjon som ikke ble tatt med, er å gi FiB muligheten til å gi en poengsum til et svar. Denne funksjonen hadde ikke så stor verdi for FiB, og det ble heller foreslått å gi brukerne av chatboten muligheten til å gi tilbakemelding på svaret fra chatboten (**krav 10**).

## 3.2 Forslag til løsning

For å utvikle en god løsning er det viktig å velge et godt utviklingsmiljø, rammeverk og biblioteker for oppdraget. Den allerede eksisterende chatboten er utviklet i Python, med Flask rammeverk for å fremvise chatboten på nettsiden til FiB.

### 3.2.1 Webbasert app VS. desktop app

For å lage verktøyet FiB etterspurte var det mulig å lage det som en webbasert app eller en desktop-app. En web-app er en app som tilgjengeliggjøres gjennom en nettleser (som Chrome eller Safari) over et nettverk. En desktop-app kan også hente og motta informasjon over nett, men må installeres og kjøres lokalt på maskinen som skal bruke den.

Applikasjonen vår vil ikke kunne nyttiggjøre mange av fordelene til en desktop-app, men vil ha nytte av fordeler som kommer med en webbasert app. En av de største fordelene innebærer at applikasjonen vil være enklere å oppdatere og vedlikeholde ettersom hver enkel klient som bruker applikasjonen ikke vil trenge å laste ned en oppdatert versjon. I tillegg vil det være enklere å utvikle løsningen som en webbasert app da en ikke lengre trenger å ta hensyn til operativsystemet for hver av maskinene som tar i bruk løsningen. (Banerjee, 2022)

### 3.2.2 Rammeverk

Det finnes mange relevante webbaserte rammeverk. Django, Flask og Ruby on Rails er alle bredt brukte rammeverk for webbaserte løsninger, og ble målt opp mot hverandre for dette prosjektet. (Lomas, 2022) Et naturlig valg er å bruke samme rammeverk som chatboten bruker. Dette vil ikke direkte lønne seg for applikasjonen, men kan være gunstig ettersom det vil bli enklere for en fremtidig utvikler å sette seg inn i hvordan chatboten og administrasjonsverktøyet fungerer. Rammeverket som allerede er brukt er Flask. Dette betydde også at gruppen uansett måtte sette seg inn i Flask-rammeverket for å kunne gjøre de nødvendige endringene for at chatboten skulle kommunisere med løsningen.

Flask er et mindre omfattende rammeverk enn de andre nevnte alternativene. Dette kan gjøre det enklere og mindre tidkrevende å lære seg. (Lomas, 2022) Et mindre komplisert rammeverk er ideelt for dette prosjektet ettersom det er begrenset tid tilgjengelig for å fullføre prosjektet. Flask er i tillegg til å være et lett rammeverk også et svært kraftig. Ved hjelp av ekstra biblioteker har Flask-rammeverket all funksjonaliteten nødvendig for administrasjonsverktøyet.

Generelt har Flask noen begrensninger. Det er ikke like skalerbart som andre løsninger og er ikke designet for å håndtere svært høy trafikk. Disse begrensningene var ikke en bekymring i dette prosjektet, da FiB-CA kun skulle brukes av et begrenset antall personer om gangen. Videre er Flask-felleskapet mindre enn for noen andre Python-rammeverk, som Django, noe som kan gjøre det mer utfordrende å finne hjelp eller løsninger på spesifikke problemer. (Ranjan, 2022) Gruppen endte opp med å velge å utvikle løsningen i Flask i all hovedsak fordi chatboten brukte det, og det fremsto som det minst tidkrevende rammeverket å sette seg inn i.

### 3.2.3 Design

Under utviklingen av en applikasjon er det viktig å gå frem med brukeren i tankene. Dette innebærer å skaffe seg forståelse for hva brukeren ønsker å oppnå med applikasjonen, og hvordan brukeren vil gå frem for å oppnå dette. Dette kan oppnås med et godt visuelt hierarki som gjør det logisk og intuitivt for brukeren å finne frem til ønsket funksjonalitet (Sharp, Rogers, & Preece, 2019, ss. 216-218). For å fjerne mer tvil om hva hensikten med de ulike elementene i brukergrensesnittet er ment for, er det lagt til forklarende tekster på hver side i applikasjonen. Dette vil hjelpe en bruker av applikasjonen med navigasjon og informasjon om hva de har mulighet til å gjøre.

Gruppen valgte å begrense antall farger som skal bli brukt i FiB-CA for å gi applikasjonen et profesjonelt utseende. Begrensingen av farger bidrar til å gi en mer konsistent og strukturert visuell opplevelse, og vil i tillegg gi brukerne en følelse at produktet er pålitelig og funksjonelt. (Wong, 2021)

Gitt oppgavens avgrensning, der fokus er på andre aspekter enn et dynamisk brukergrensesnitt, mangler programmet nødvendig skalerbarhet for å fungere optimalt på en mobiltelefon. Som tidligere nevnt i kapittel 2.2 Avgrensninger, er det planlagt at produktet skal brukes på arbeidsmaskiner hos FiB, noe som gjør et dynamisk brukergrensesnitt unødvendig. Tilpasning til forskjellige skjermopløsninger er ønsket, men vil ikke være en prioritet i utviklingsfasen.

### 3.2.4 Database

Ettersom Flask ble valgt som rammeverk var det også mulig å benytte seg videre av biblioteket Flask-SQLAlchemy for å kommunisere med databasen. Dette resulterte i at koden ble kompatibel med alle de største typene SQL-databaser som SQLite, PostgreSQL, MySQL, Oracle, MS-SQL, Firebird, Sybase og mer. (SQLAlchemy, 2023). Ettersom Flask-SQLAlchemy ble brukt for oppkobling mot databasen var det ikke nødvendig å ta i bruk SQL spørringer. Dette hjalp videre med å beskytte appen mot SQL-injeksjoner. Ettersom det ble bestemt å bruke Flask-SQLAlchemy var det mulig å utsette valg av hvilken type SQL-database som skulle brukes frem til implementasjonen av programmet. Under utviklingsfasen ble SQLite brukt for en midlertidig database for feilsøking. Valget av SQLite ble gjort uten store overveielser ettersom dette var for en database som ikke skulle brukes videre i den endelige løsningen.

FiB sin chatbot hadde allerede en database den brukte. Det blir nærmere forklart hvorfor denne ikke ble brukt videre i kapittel 4.1.1 Kommunikasjon med chatbot og kapittel 4.1.2 Datalag.

## 3.3 Valgt løsning

Det er viktig for prosjektet at løsningen er intuitiv og enkel å bruke. Det skal være et oversiktlig program med tilstrekkelig dokumentasjon for videre utvikling. Dette er for at det senere skal være enklere å legge til gjenværende funksjonalitet som ikke er implementert. Gruppen valgte å kalle den valgte løsningen FiB - Chatbot Administrasjonsverktøy, heretter kalt FiB-CA.

Flere alternativer ble gjennomgått, men sluttresultatet endte med å være en webbasert løsning laget med Python, HTML og CSS i et Flask-rammeverk. Valg av løsningen er forklart i kapittel 3.2



Forslag til løsning. FiB hadde et ønske om at FiB-CA skulle være tilgjengelig hvor som helst ved bruk av innlogging. For å opprettholde krav om brukervennlighet og tilgjengelighet vil en webbasert løsning være mest passende, ettersom FiB kan bruke applikasjonen så lenge de har tilgang på en datamaskin. FiB-CA vil i all hovedsak bli brukt på arbeidsdatamaskiner, og designet vil derfor bli utviklet med tanke på at det skal brukes på en dataskjerm.

## 3.4 Valg av verktøy

### 3.4.1 Visual Studio Code

Visual Studio Code (VS Code) er en gratis, åpen kildekode-kodeeditor utviklet av Microsoft. Det er tilgjengelig for Windows, Linux og MacOS. Det gir et bredt spekter av funksjoner for utviklere, inkludert syntaks-fremheving, kodeutfylling, feilsøking og integrert støtte for Git. VS Code støtter også en rekke programmeringsspråk, inkludert, men ikke begrenset til: JavaScript, TypeScript, Python, C++ og Java (Microsoft - VS Code, 2022). Gruppen valgte å bruke VS Code som sin kodeeditor fordi medlemmene har tidligere gode erfaringer med programmet.



Figur 3.2 VS Code logo (Microsoft, 2019)

### 3.4.2 Flask

Flask er et Python-web-rammeverk som hjelper med å bygge nettapplikasjoner raskt og enkelt. Det er et mindre omfattende, fleksibelt og enkelt å lære, noe som gjør det til et populært valg både for nybegynnere og erfarne utviklere. Med Flask kan du opprette dirigeringer av URL'er, håndtere HTTP-forespørsler og -responser, og jobbe med maler for å opprette dynamiske websider. Det har også innebygd støtte for sikre informasjonskapselbaserte økter, og den modulære utformingen gjør det mulig å legge til utvidelser for database-tilkobling, autorisering og mer. Alt i alt er Flask et godt valg for å bygge nettapplikasjoner i alle størrelser og kompleksiteter. (Flask User's Guide, u.d.)

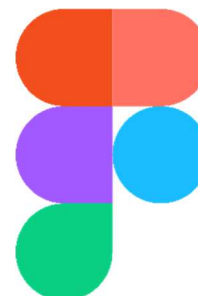


# Flask

Figur 3.3 Flask logo (Flask, u.d.)

### 3.4.3 Figma

Figma er et skybasert design- og prototype-verktøy som lar team samarbeide om design og utvikling av digitale produkter, som f.eks. nettsteder og mobilapper. Det har et brukervennlig grensesnitt som gjør at brukere kan lage og redigere vektorgrafikk, trådkisser (wireframes) og interaktive prototyper (Figma, u.d.). Gruppen valgte å bruke dette verktøyet for å visualisere en løsning og lage en prototype for produktet. Figma ble brukt fordi det er et brukervennlig verktøy som gruppen hadde tidligere erfaring med.



Figur 3.4 Figma logo (Figma, 2019)

## 3.5 Valg av hjelpemidler

### 3.5.1 GitHub

GitHub er en plattform for utvikling og versjonskontroll av programvare. Det gir utviklere muligheten til å lagre og dele koden sine på en enkel måte. GitHub gir og et bredt spekter av verktøy for samarbeid, inkludert feilsparing, oppgavehåndtering og diskusjonsforum. Det er også et stort fellesskap av utviklere fra hele verden som bidrar og deler åpen kildeprosjekter. (Lutkevich, 2023)



*Figur 3.5 GitHub logo  
(GitHub, u.d.)*

### 3.5.2 Verktøy for skriving

Google Drive er en skybasert lagringstjeneste som tilbys av Google. Det gir brukerne muligheten til å lagre og dele filer, inkludert dokumenter, regneark, bilder, videoer og andre filtyper i skyen. Google Docs og Google Sheets er to av applikasjonene som er integrert i Google Drive. Google Docs er et tekstbehandlingsprogram som lar brukerne samarbeide på dokumenter i sanntid, mens Google Sheets er et regnearkprogram som gir muligheter for å lage og redigere regneark og dele dem med andre. Begge disse verktøyene gir også muligheter til å kommentere, samarbeide og redigere samtidig på dokumentene (Nolledo, 2020). Gruppen valgte i starten å bruke denne løsningen da tjenesten gir enkel og rask fildeling og gir mulighet for samarbeid samtidig som gruppen allerede har god kunnskap til tjenesten.

Overleaf er et nettbasert samarbeidsverktøy som gir brukerne muligheten til å skrive og redigere dokumenter online, og gir også verktøy for versjonskontroll og samarbeid. I tillegg tilbyr Overleaf en rekke maler for ulike dokumenttyper, inkludert akademiske papirer, bøker og presentasjoner (Overleaf, u.d.). Under første møte mellom gruppen og veiledere fra HVL og Bouvet var det enighet om å skrive prosjektet i Overleaf, etter oppfordring fra veileder fra HVL.

I oppstartsfasen ble en kombinasjon av Overleaf og Google Drive brukt til å skrive rapportene. Overleaf bruker LaTeX som syntaks til å bygge opp dokumentet, men ettersom gruppen ikke var kjent med denne syntaksen tok det lengre tid å skrive rapporter i Overleaf sammenlignet med Google Docs. For å kunne benytte seg av Overleaf må en betale et månedlig beløp. Dette førte til at gruppen sluttet å bruke Overleaf, og heller kun bruke Google Drive. En av grunnene til å bruke Overleaf var å kunne samle dokumentene på en plass, slik at det var enkelt for veileder å kunne se over dokumentene. Denne funksjonaliteten manglet Google Drive. På bakgrunn av dette ble Teams introdusert som en erstatter for både Overleaf og Google Drive.

Microsoft Teams fungerer på samme måte som Google Drive ved at en kan jobbe sammen på samme fil i sanntid. Microsoft Teams kommer med Word og Excel, som tilsvarer Google Docs og Google Sheets fra Google Drive. Det Microsoft Teams gjør annerledes er å introdusere en mappestruktur der en kan samle filer og enkelt få oversikt. Egenskapene som gruppen likte ved Google Drive og Overleaf, fant gruppen samlet i Microsoft Teams.+

### 3.5.3 Discord

Discord er en populær kommunikasjonsplattform designet for online-samfunn. Det tilbyr en rekke funksjoner, inkludert tekst- og «voice chat», samt muligheten til å opprette og delta i forskjellige "servere" (som i prinsippet er grupper). I sammenheng med gruppeprosjekter kan Discord brukes som et verktøy for kommunikasjon og samarbeid mellom teammedlemmer. For eksempel kan teammedlemmer bruke stemme- eller tekstkanaler innenfor en Discord-server for å diskutere prosjektrelaterte oppgaver og spørsmål, dele filer og oppdateringer og koordinere møter. Discord har også muligheten til å opprette forskjellige roller og tillatelser for medlemmer, noe som gjør det enkelt å administrere og organisere teammedlemmer. Samlet sett gir det et sentralisert sted for teammedlemmer å kommunisere og holde seg oppdatert på prosjektets fremdrift. Gruppen brukte allerede dette hjelpemiddelet før prosjektstart, og konkluderte med at det ikke var noe gevinst i å flytte til en annen plattform.



Figur 3.6 Discord logo  
(Discord, u.d.)

### 3.5.4 Trello

Trello er et nettbasert prosjektstyringsverktøy som lar brukere organisere og prioritere oppgaver og prosjekter ved hjelp av et visuelt, "kortbasert" grensesnitt. Hver oppgave eller element representeres av et "kort" som kan flyttes mellom forskjellige "bord" eller kategorier, og hvert bord kan tilpasses med forskjellige lister og merkelapper. Trello gir muligheten til enkelt å dele oppgaver og prosjekter med lagkamerater, samt å samarbeide på dem. Med de nevnte egenskapene fungerer Trello godt til å bli brukt som et Scrum-board, som blir beskrevet i kapittel 3.6.1 Utviklingsmetodikk. I tillegg har gruppen positive erfaringer med Trello fra tidligere prosjekt, og Trello ble derfor brukt i bacheloroppgaven.



Figur 3.7 Trello logo  
(Freepik, u.d.)

### 3.5.5 Bootstrap

«Bootstrap er en gratis, åpen kildekode front-end utviklingsrammeverk til utvikling av nettsider og nettapper» (oversatt) (Bootstrap, 2022). Bootstrap kommer med ferdige design og funksjonalitet for ulike elementer som en trenger til nettsider, noe som gjør det enkelt og raskt å få et godt utseende på nettsiden. Rammeverket til Bootstrap er bygget opp av HTML, CSS og JavaScript.

### 3.5.6 Chat GPT

ChatGPT er en AI-språkmodell som er utviklet av OpenAI. Den er designet for å forstå og generere tekst ved å analysere og lære fra store mengder med data. ChatGPT kan brukes for å svare på spørsmål og å utføre ulike tekstbaserte oppgaver. Språkmodellen er i stand til å tilpasse seg til forskjellige situasjoner og emner, men har manglende kunnskap etter september 2021. ChatGPT kan også brukes som et hjelpemiddel for å identifisere og løse problemer i programkode. Den analyserer koden og vil foreslå mulige feil og deres årsaker, samt gi løsningsforslag for å rette opp i eventuelle feil.



Figur 3.8 ChatGPT logo  
(OpenAI, 2023)

ChatGPT er et nyttig verktøy for å få tilbakemeldinger på språk og struktur i rapporten. Ettersom ChatGPT er en språkmodell som er designet for å generere et naturlig språk, kan den gi veiledning til formuleringer, grammatikk og struktur.

## 3.6 Prosjektmetodikk

### 3.6.1 Utviklingsmetodikk

Fossefall og Smidig utvikling er to hovedretninger innen utviklingsmetodikk. Fossefall er den eldste metodikken og basere seg på en lineær framgang. Den lineære framgangen kan deles inn i fem steg: sette krav, design, kode, test og vedlikehold. Fossefall fungerer godt der sluttresultatet er klart definert før start, og hvor prosjektet krever mye forutsigbarhet. Dette skyldes at hver fase i fossefallsmodellen er grundig og omfattende, noe som sikrer at alle faser av prosjektet er fullt forstått og gjennomtenkt før en går videre. Dette reduserer risikoen for uventede problemer eller feil i senere faser. Utviklingsmetodikken egner seg ikke så god til å implementere nye krav eller ta imot endringer fra kunden underveis, og en vil ikke før i slutten av prosjektfasen finne ut om det endelige produktet samsvarer med kundens ønske og behov (Horry & Bottorff, 2022).

Smidig utvikling er en nyere utviklingsmetodikk der en går fra lineær til en sirkulær framgang. I Smidig utvikling er det sterk fokus på å inkludere kunden i utviklingen, ved å motta tilbakemeldinger på produktet og å være åpen for endringer som kunden ønsker. Smidig utvikling legger også fokus på å tidlig lage det fungerende produktet med de viktigste funksjonene som en så bygger videre på. Ved å inkludere kunden i større grad vil det være enklere å sørge for at produktet oppfyller kundens behov og krav. Det gir også mulighet til å stoppe utviklingen med et brukbart produkt om budsjettet blir nådd tidligere enn planlagt. (Dingsøy, 2022).

Da dette prosjektet ikke hadde et klart definert sluttresultat, men et behov for jevnlig tilbakemeldinger fra kunden, ble en form for smidig utviklingsmetodikk tatt i bruk for dette prosjektet. Ettersom prosjektet ikke er så stort, ble prosjektets utviklingsmetodikk basert på iterativ utvikling med elementer fra Scrum. Ifølge Dingsøy er Scrum den mest populære metoden for smidig utvikling (Dingsøy, 2022).

Elementer fra Scrum som ble tatt i bruk er blant annet Scrum-board, ettersom dette ga en god oversikt over oppgaver som måtte gjøres, påbegynte oppgaver og gjennomført oppgaver. Scrum-boardet ble benyttet for å fordele hvem som hadde ansvar for hva. Det å dele utviklingen inn i kortere etapper, kalt sprinter, er en del av Scrum-utviklingsmetoden. Det ble brukt sprinter i dette prosjektet. På slutten av hver sprint avholdes det Sprint-retrospektivt og Sprint-planlegging for å få oversikt over hva som var gjort i forrige sprint og hva som skal gjøres i neste sprint. Mellom sprintene er det også viktig å inkludere kunden slik at det tidlig kan gjøres endringer på produktet ved behov. Kundemøtene er viktige for å påse at de viktigste funksjonene for kunden blir prioritert. Underveis i sprintene avholdes det daglig uformelle statusmøter der det kommuniseres hvilke oppgaver hver enkelt prosjektdeltaker planlegger å jobbe med og om noen eventuelt trenger hjelp med en oppgave (Scrum.org, i.d.).

Sprintene i prosjektet var to uker lange, der hver sprint tilsvarte en iterasjon fra iterasjonsplanen. Iterasjonsplanen ligger i kapittel 3.6.2 Prosjektplan. Denne varigheten ga rom for å arbeide med å løse mer omfattende oppgaver sammenlignet med et kortere tidsintervall. I tillegg til dette møttes alle gruppemedlemmene minimum 3 dager i uken for å jobbe sammen. I løpet av disse dagene hadde gruppen jevnlig statusmøter, der gruppemedlemmene oppdaterte hverandre på fremgang i prosjektet, og planla videre arbeid. Dette gjorde det enkelt for gruppen å støtte hverandre, og å bidra dersom en av gruppemedlemmene møtte på utfordringer i en av oppgavene.

### 3.6.2 Prosjektplan

For å lage en oversiktlig plan over hvilke oppgaver som skulle gjøres i prosjektet ble det utarbeidet en prosjektplan i form av et GANTT-diagram. GANTT-diagrammet ligger under punkt 2.1 GANTT Skjema, i vedlegg 9.1.1 *Prosjekthåndbok*. GANTT-diagrammet er inndelt i de seks fasene nevnt under. Utviklingsmetodikken til prosjektet krever iterativ utvikling, og det er derfor utarbeidet iterasjoner som blir beskrevet under Systemutvikling.

#### **Oppstartsfasen:**

I oppstartsfasen er hovedmålet å definere målene og rammene for prosjektet, og å etablere en felles forståelse blant alle involverte parter. Det innebærer i hovedsak å diskutere og definere oppgaven og problemstillingen for prosjektet, samt å avklare forventninger, identifisere utfordringer og muligheter, og å planlegge videre arbeidsprosess.

#### **Forprosjektfasen:**

Målet med forprosjektet er å utforske og vurdere ulike innfallsvinkler og alternative løsninger for prosjektet, og å utarbeide en tydelig plan for hvordan prosjektet skal gjennomføres. Gruppen må derfor samle inn relevant informasjon knyttet til prosjektet, definere kravene og begrensningene for prosjektet, og planlegge hvordan løsningen skal designes og implementeres.

#### **Hovedprosjektfasen:**

I hovedprosjektfasen gjennomfører gruppen aktiviteter som er rettet mot å fullføre prosjektet, inkludert produksjon av produktet, rapporten og støttedokumenter. Hovedmålet med denne fasen er å realisere og levere den planlagte løsningen, og å dokumentere prosessen og resultatene på en grundig måte.

#### **Støttedokumenter:**

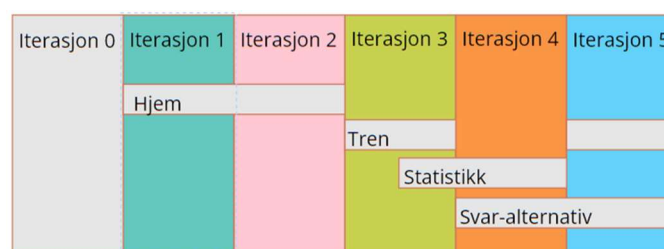
Støttedokumenter er dokumenter tilknyttet rapporten som inneholder prosjekthåndbok, kravspesifikasjon, visjonsdokument og systemdokumentasjon. Dokumentene oppdateres kontinuerlig gjennom hele prosjektperioden.

#### **Systemutvikling:**

Utviklingsdelen av prosjektet er iterasjonsbasert, og prosjektarbeidet i denne fasen er delt inn i iterasjonene 0-4, der hver av iterasjonene inneholder spesifikke mål å jobbe mot. I **Iterasjon 0** av prosjektet er målet å sette opp flask-rammeverket, og å koble det opp mot databasen. I **Iterasjon 1** er målet at FiB-CA (Administrasjonsverktøyet som utvikles) skal

kunne vise spørsmål som er blitt sendt inn, og hva chatboten har svart. Her skal gruppen også lage en ny versjon av den eksisterende databasen.

I **Iterasjon 2** skal det være mulig å validere spørsmål chatboten har svart riktig på, og å markere riktige svar der chatboten har svart feil. Etter at **iterasjon 2** er ferdig skal produktet testes for hver iterasjon. I **Iterasjon 3** skal FiB-CA kobles opp mot chatbot-modellen, og det skal være mulig å trene modellen på validerte spørsmål. Det skal også opprettes en egen side med statistikk der FiB kan sammenligne statistikk fra tidligere år. **Iterasjon 4** innebærer å lage en «logg inn»-side, redigere standard-svaralternativer, legge til treffsikkerhet til chatboten på hjemmesiden, og å lage en fullverdig statistikkside. En visuell fremvisning av hvilke sider av applikasjonen hver iterasjon omhandler kan ses i figur 3.9.



Figur 3.9 Visualisering av iterasjonsplan

Det er i tillegg noen funksjoner som er ønsket fra FiB, men som gruppen tvilte på at var gjennomførbare grunnet prosjektets tidsbegrensning. I tilfellet det er tid til noen av disse ønskene ble de plassert i en ekstra iterasjon, **iterasjon 5**. Målene for iterasjon 5 er at brukeren av chatboten skal kunne gi tilbakemelding på om svaret til chatboten var til hjelp, kunne lage sesongsvar og lage nye standard-svaralternativer.

En mer detaljert iterasjonsplan finnes i vedlegg 9.1.1 *Prosjekthåndbok* i kapittel 2.2 Iterasjonsplan.

### Kompetanseinnhenting:

Kompetanseinnhenting er en kontinuerlig aktivitet gjennom hele prosjektperioden. Dette innebærer ulike aktiviteter, som for eksempel forelesninger og undersøkelser. I løpet av kompetanseoppbyggingen vil gruppen undersøke relevant teknologi og litteratur, samt delta i møter med oppdragsgiver og veileder for å forbedre sin forståelse av prosjektet.

GANTT-diagrammet i vedlegg 9.1.1 *Prosjekthåndbok* gir en detaljert oversikt over aktivitetene i hver fase og deres tidsplaner.

### 3.6.3 Risikovurdering

Risikoanalysen avdekket flere risikoer ved prosjektet. Punktene med høyest risikoprodukt var at gruppen kunne feil-estimere tid slik at prosjektet ikke ble ferdig, og at en ikke skulle få tilgang til FiB sin data tidsnok til å kunne fullføre oppgaven. Det er på grunn av tidsavgrensningen til prosjektet at det har fått så høy risiko. For å øke sannsynligheten for å ha et ferdig produkt i slutten av prosjektet valgte gruppen å følge en iterativ utviklingsprosess. Dette innebar å raskt skaffe seg et brukbart produkt og utbedre funksjonaliteten på dette gjennom prosjektet. Dette ville også være til hjelp ved å prioritere viktige deler av prosjektet for tidligere iterasjoner. Den helhetlige risikoanalysen befinner seg i vedlegg 9.1.1 *Prosjekthåndbok*.

Tabell 3.1 Risiko 8

Nr	Hendelse /Risiko	Årsak	Sannsynlighet	Konsekvens	Risiko-produkt	Tiltak
8	Feil-estimert tid til å ferdigstille programmet	Manglende erfaring	Høy (4)	Høy (4)	16	Prioriterer oppgaver som er essensielt for applikasjonen.

Under oppsettet av risikoanalysen ble gruppen enige om at «Feil-estimert tid til å ferdigstille program» hadde høyest risiko. Dette er blant annet fordi gruppen bruker web-rammeverket Flask som de ikke har brukt før. Gruppen vil benytte seg av nøye planlegging for å hindre at de ikke bruker verdifull tid på mindre prioriterte oppgaver. Dette vil resultere i at den viktigste funksjonaliteten for FiB vil bli riktig prioritert og gjennomført i de første iterasjonene.

## 3.7 Evalueringsplan

For utvikling av FiB-CA er det viktig å kunne gi en indikasjon på hvorvidt applikasjonen gjør som forventet, er brukervennlig og dekker alle behovene til FiB. For å gjøre dette må det tas i bruk evalueringmetoder for å undersøke funksjonalitet og brukeropplevelse av applikasjonen. Evalueringmetodene som er vektlagt for å unngå dette er brukertesting og bruk av testklasser (Sharp, Rogers, & Preece, 2019, ss. 495-502).

### 3.7.1 Brukertesting

Ved utviklingen av et produkt er det viktig å validere at produktet samsvarer med kundens behov og ønsker. Har utviklerne av produktet misforstått hele eller deler av kundens behov, er det viktig at dette blir avklart tidlig. Å gjøre fundamentale endringer på et produkt som allerede er avansert i sin utviklingsprosess vil kreve mer tid og ressurser enn på et som er tidlig i sin utviklingsfase. Det er derfor viktig med en tett og jevn kontakt med kunden under utviklingen av produktet. Ved hjelp av den iterative utviklingen beskrevet i kapittel 3.6 Prosjektmetodikk, kan en ved hjelp av tett og hyppig kontakt med kunden sikre at produktet samsvarer med kundens behov.

Prototyping er en enkel og økonomisk måte å kartlegge om brukerens behov og gruppens forståelse av behovet samsvarer. Samsvarer det ikke, krever det lite tid og økonomiske midler å tilpasse prototypen til kundens visjon. Prototyping vil skje med prototypeverktøyet Figma, som er beskrevet i kapittel 3.4.4 Figma. Denne evalueringen vil være med på å kartlegge hvilke endringer som eventuelt må gjøres før utviklingen av produktet (Sandnes, 2011, s. 281).

Underveis i utviklingen vil gruppen ta i bruk formativ evaluering for å evaluere om produktet så langt samsvarer med FiB sine behov og visjon, og eventuelt innspill på hva som ikke stemmer. Formativ evaluering handler om «evaluering i løpet av utviklingsprosessen» (Sandnes, 2011, s. 306). Ved formativ evaluering er det viktig å teste produktet på faktiske brukere, for å få en realistisk evaluering av brukervennligheten til produktet. Brukerne av denne applikasjonen vil være ansatte hos FiB. Hadde produktet kun blitt testet av utviklere eller andre med urealistisk høye tekniske ferdigheter, ville gruppen fått en feil forståelse av brukervennligheten til produktet. Ved brukertesting av produktet vil det oppdages eventuelle feil og mangler i forhold til utforming eller funksjonalitet (Sandnes, 2011, s. 305).

Under brukertestene vil det bli lagt vekt på observasjon av brukerne, for å få et bedre bilde av applikasjonens brukervennlighet. Ved å observere hvordan testsubjekter navigerer og bruker applikasjonen kan utviklere identifisere mulige mangler, brukerens behov og applikasjonens brukervennlighet. Noen av de observerte problemene kan være problemer som den observerte selv ikke så (Ross, 2018). I tillegg til observasjon vil det bli brukt spørsmål som gir en enda dypere forståelse av brukernes oppfattelse av produktet. Det blir beskrevet nærmere hvordan brukertestene skal gjennomføres i kapittel 5.1.1 Brukertesting.

Underveis vil gruppen evaluere tidsstatusen til prosjektet ved å se på hvordan prosjektet ligger an etter iterasjonsplanen, som er nevnt under kapittel 3.6.2 Prosjektplan. Evalueringen av tidsstatusen vil også bli evaluert ut fra GANTT-diagrammet som ligger under 2.1 GANTT skjema i vedlegg 9.1.1 *Prosjekthåndbok*. Ved å evaluere tidsstatusen vil gruppen vite om prosjektet ligger godt an til å bli fullført. Kommer prosjektet på etterskudd må det legges ned mer arbeid eller mindre viktige oppgaver må nedprioriteres. Går prosjektet derimot raskere enn planlagt, vil det bli gitt tid til å utføre flere oppgaver.

Prosjektet vil ha en summativ evaluering, i form av en sluttevaluering fra FiB og Bouvet. Summativ evaluering handler om evaluering i ettertid, og vil her blir brukt til å evaluere om sluttproduktet samsvarer med FiB sin visjon og behov. (Sandnes, 2011, s. 306). I sluttevalueringen vil en se på om en har klart å fullføre alle målene, og eventuelt hva en ikke fikk utført og hvorfor det ikke ble utført. I sluttevalueringen vil en se på hva som kunne ha blitt gjort bedre underveis i utviklingen og hvor det er behov for videre utvikling av produktet. Sluttevalueringen vil skje etter iterasjon 4, når utviklingsfasen er planlagt å være ferdig. Sluttevaluering fra Bouvet handler om å evaluere den tekniske delen av prosjektet, og hvordan det har vært for Bouvet å jobbe med Gruppen på dette prosjektet.



### 3.7.2 Enhetstesting

For å verifisere at produktet er utviklet riktig og med god nok kvalitet, vil det bli brukt enhetstesting. Enhetstesting handler om å kunne teste en liten del av koden isolert, uten å måtte ta hensyn til hele programmet. Under testing valideres koden til å fungere under unntakstilfeller, slik at programmet blir mer robust. Et mer robust program vil ha færre defekter, som igjen betyr færre feilrettinger. Skulle det skje en feil vil enhetstestene hjelpe med å raskere finne feilen i programmet, og korrigere feilen på en slik måte at det er mindre sannsynlig at det oppstår en ny feil (Heusser, 2020).

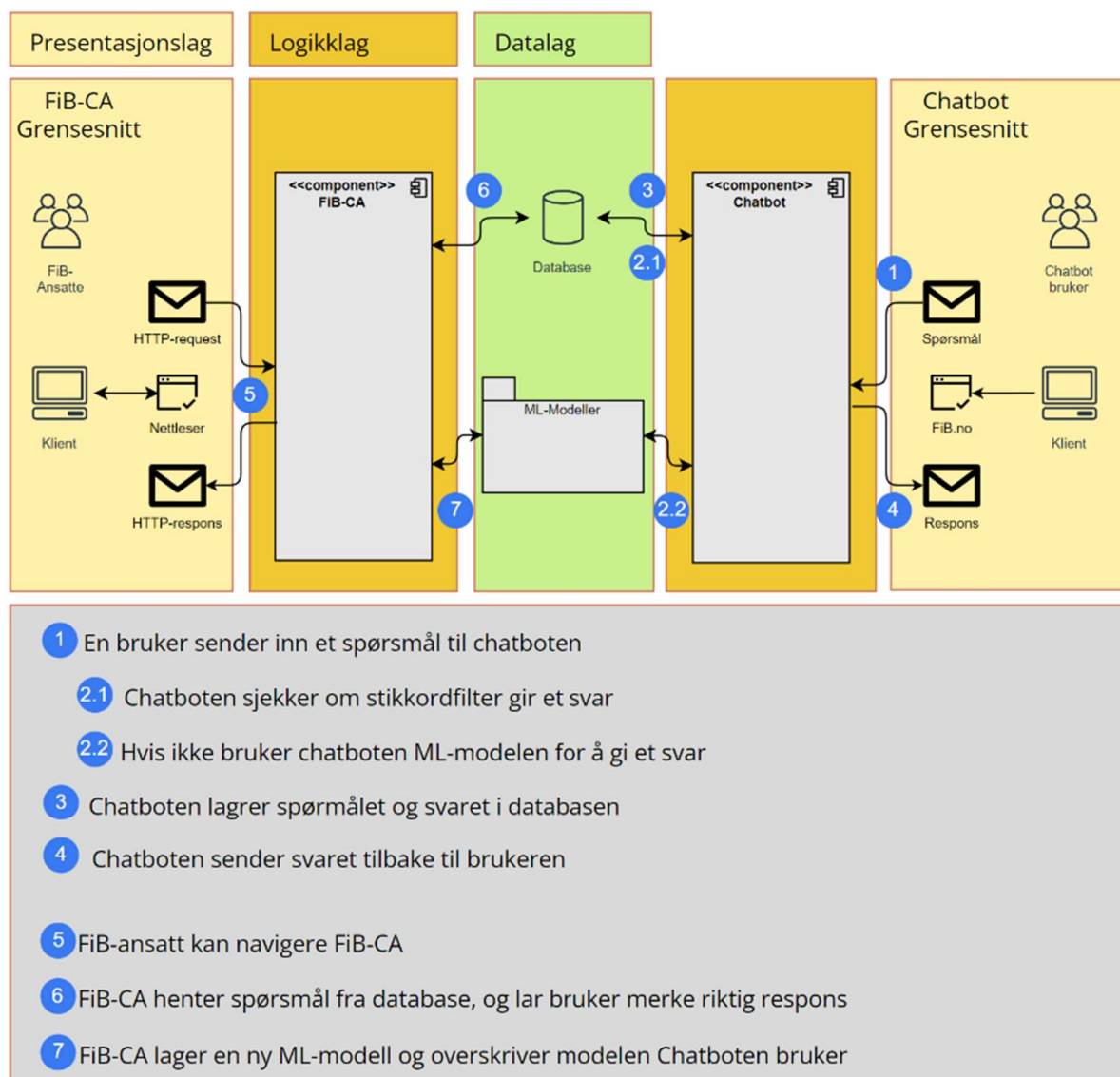
Enhetstesting gir også en form for dokumentasjon på hvordan koden fungerer. Testene fungerer som et sett med krav til koden, og beskriver hvordan den skal oppføre seg i forskjellige situasjoner. Dette kan hjelpe andre utviklere med å forstå koden og hvordan den fungerer.



## 4 Design og arkitektur / Detaljert løsning

### 4.1 Systemarkitektur

FiB-CA er bygd med en tre-lags arkitektur. Tre-lags arkitektur skiller applikasjoner i tre logiske og fysiske datalag. Dette er den dominerende programvarearkitekturen for tradisjonelle klient-serverapplikasjoner. De tre lagene består av presentasjons-laget, som er ansvarlig for å vise og motta informasjon fra brukeren, logikklaget, som er ansvarlig for å prosessere data, og datalaget, der dataen for programmet er lagret. (IBM, u.d.)



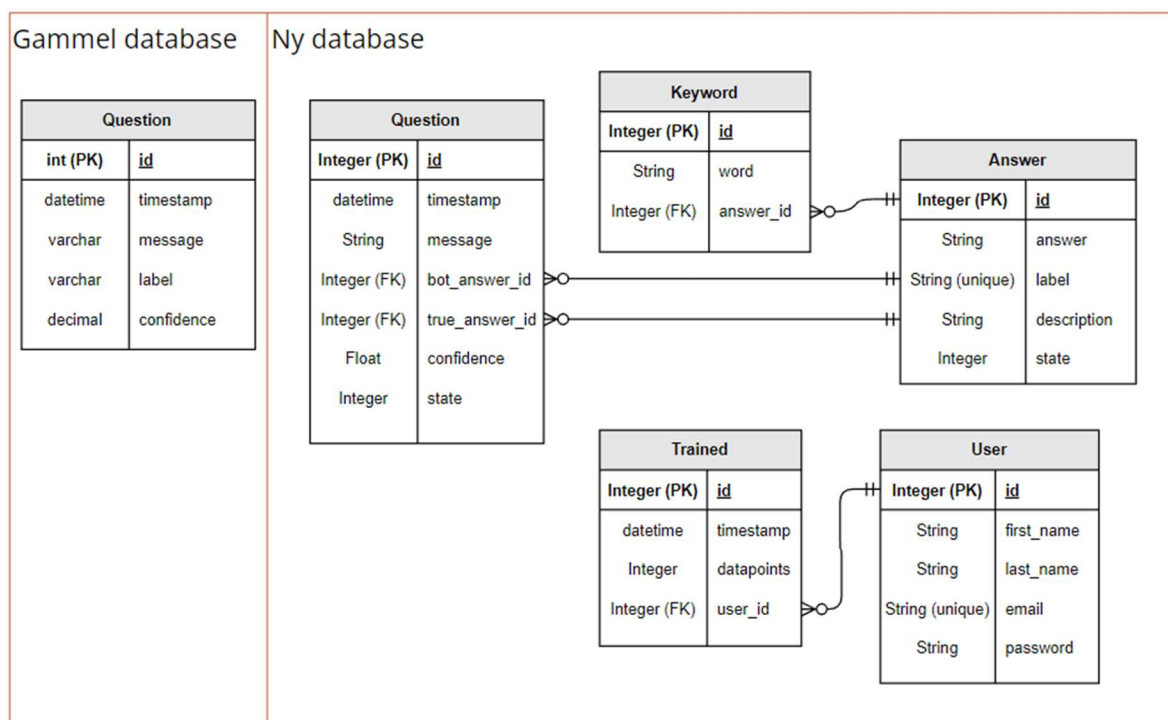
Figur 4.1 Sammenheng mellom FiB-CA og chatbot

### 4.1.1 Kommunikasjon med chatbot

FiB-CA må kunne kommunisere med den allerede eksisterende chatbot-applikasjonen for å fungere som intensjonelt. Kommunikasjonen skjer gjennom et delt datalag. Dette er mulig ettersom både FiB-CA og chatbot-programmet er bygget med en 3-lags arkitektur. Kommunikasjonen og arbeidsfordelingen mellom FiB-CA og chatboten er vist med figur 4.1.

Den allerede eksisterende løsningen brukte den gamle databasen vist i figur 4.2. Denne databasen inneholdt ikke all informasjonen FiB hadde ønske om å kunne endre på. På grunn av dette, måtte datalaget til chatboten designes på nytt slik at både chatbot-programmet og FiB-CA kunne bruke det. Det står mer om endringene til databasen i datalaget under kapittel 4.1.2 Datalag.

Endringene i datalaget førte til at det også var nødvendig med noen mindre endringer i logikklaget til chatbot-programmet. Dette var for at chatboten fortsatt skulle bruke datalaget som før, i tillegg til å kunne ta i bruk ny informasjon som blant annet svaralternativer.



Figur 4.2 Entitetsdiagram til ny og gammel database

### 4.1.2 Datalag

Ettersom gruppen ikke har fått tildelt en server for datalaget, er både databasen og mappen for lagring av modeller for øyeblikket satt opp internt i FiB-CA. Koblingen mot databasen er satt opp fra begge sider med bruk av biblioteket Flask-SQLAlchemy. Dette gjør at det ikke vil være nødvendig med betydelige endringer i noen av programmene for å koble de opp mot en ekstern database.

Som nevnt i kapittel 4.1.1 Kommunikasjon med chatbot, måtte datalaget designes på nytt for at begge programmene skulle kunne benytte det. Dette ble gjort gjennom å designe en ny database. Designet for den nye databasen vises i figur 4.2. Endringene i databasen er beskrevet under.

I den nye databasen ble chatbotens svar, label, erstattet med en fremmednøkkel til en ny tabell. Den nye tabellen ble introdusert for å gi en skalerbar måte for FiB å fjerne og legge til nye svar for chatboten. Den nye tabellen for svar kan også ha nøkkelord knyttet til seg. Dette var for å kunne bruke samme tabell for standard- og sesongsvar.

Et spørsmål refererer til to svar, der det ene peker mot hvilket svar chatboten gir, og det andre peker på det riktige svaret. Ved å ta vare på begge verdiene for hvert spørsmål, vil en vite om chatboten ga rett eller feil svar på spørsmålet. Denne informasjonen er relevant for å lage statistikk.

Tabellen for spørsmål og tabellen for svar har en kolonne «state», som sier hvilken tilstand spørsmålet eller svaret er i. Tilstanden til svaret sier om det er et sesongsvar, et standardsvar og om svaret er aktivert eller deaktivert. Tilstanden til et spørsmål sier om spørsmålet er validert, blitt trent på eller ikke blitt validert.

### 4.1.3 Chatbotens logikklag

I dette kapittelet blir det jevnlig referert til hendelser som er illustrert i figur 4.1. Disse referansene vil være merket med **(x)**, der x er nummeret på steget i illustrasjonen.

For at chatboten skulle kunne ta i bruk det nye datalaget måtte enkelte deler av logikklaget i chatboten endres. Med de nye endringene blir spørsmål stilt til chatboten lagret i databasen med de nye nødvendige dataverdiene som FiB-CA trenger **(3)**. Når et nytt spørsmål blir lagret i databasen kan FiB-CA hente ut dette **(6)**. Det ble også ordnet slik at endringer på svaralternativene ved hjelp av FiB-CA ville bli gjenspeilet i chatboten.

For å ta i bruk sesongsvar måtte det gjøres noen endringer i måten chatboten svarer på et spørsmål. I den nåværende løsningen sjekker chatboten først om spørsmålet inneholder noen av stikkordene til sesongsvar **(2.1)**, og dersom det ikke blir funnet noen av disse bruker chatboten modellen til å gi et svar på spørsmålet **(2.2)**. FiB-CA vil også ha mulighet til å trene en ny maskinlæringsmodell, som vil bli byttet ut med den eksisterende modellen i chatboten **(7)**.

Den endelige kommunikasjonen med chatboten kan få noen endringer ved produksjon. Dette kan skyldes at FiB velger en annen database, eller at det blir endringer på hvor den nye modellen skal lagres.

## 4.2 Design og oppsett

I objektorientert programmering brukes ofte programvareteknikken SOLID, som står for 5 grunnleggende prinsipper. Disse prinsippene hjelper utviklere til å lage programmeringsystemer som er enkle å vedlikeholde, forstå og utvide over tid. Et av SOLID-prinsippene er «Single responsibility»-prinsippet, som tilsier at hver komponent kun skal ha ett formål (Thorben, 2023). Dette er et prinsipp som benyttes under utviklingen av FiB-CA, og kan bli anvendt på funksjoner, klasser og visuelle komponenter. Funksjonaliteten til FiB-CA er fordelt på fire sider der hver side har et overordnet hovedformål; **Hjem** for å sjekke enkeltspørsmål, **Administrer svaralternativer** for å bestemme hva chatboten kan svare, **Tren** for å oppdatere chatboten sin maskinlæringsmodell, og **Statistikk** for å skaffe seg et overblikk over chatbotens prestasjoner. I tillegg til disse fire sidene er det også en «logg inn»-side og en registrer-side. Med denne sidefordelingen får hver side et eget unikt overordnet formål samtidig som navigasjonsbaren ikke vil bli overfylt og uoversiktlig.

Som nevnt under kapittel 3.2.3 Design, vil FiB-CA inneholde få farger for å opprettholde et profesjonelt design, samt at fargene blir brukt for å gi viktig informasjon som varsler og statistikk. For å gjøre FiB-CA mer intuitiv vil det også bli lagt vekt på å få et logisk visuelt hierarki, samt legge til forklarende tekst til de ulike sidene.

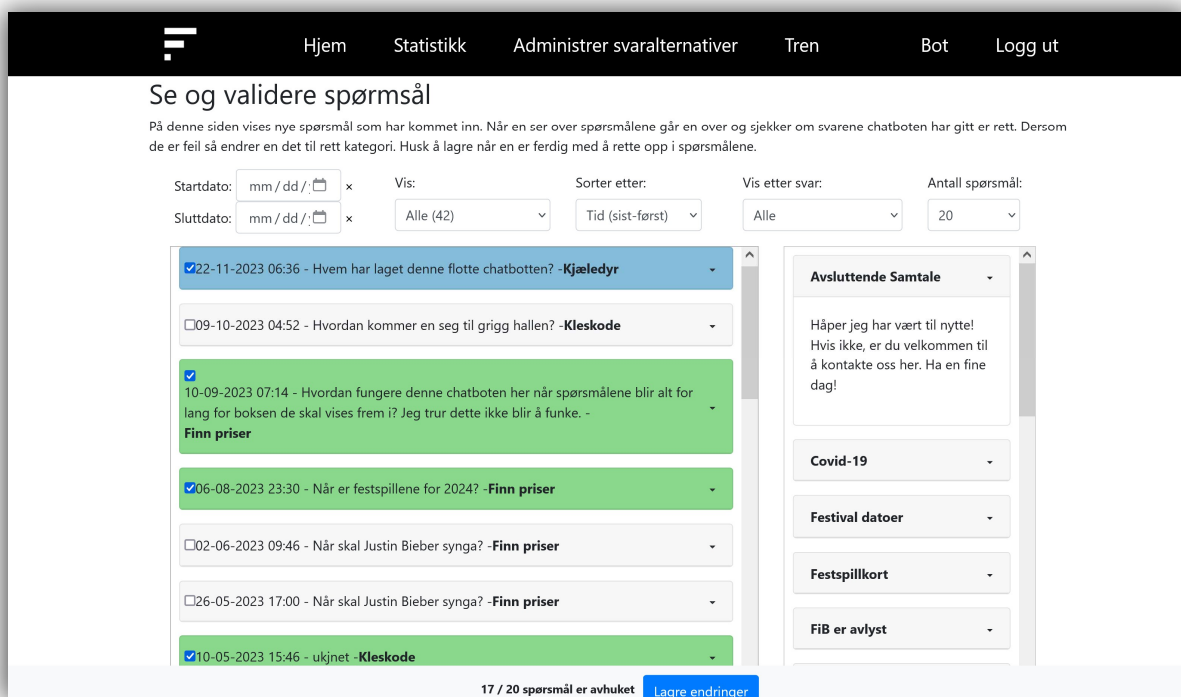
Under vil hver side bli tatt for seg av gangen, der prototypen og det endelige produktet med eventuelle endringer fra prototypen presenteres.

### 4.2.1 Hjem

Hensikten med hjem-siden er å vise spørsmål fra chatboten og gi mulighet til å validere spørsmålene, som dekker **delmål 1**: Det skal være mulig å vise spørsmål stilt til chatboten og hva chatboten har svart, samt **delmål 2**: Det skal være mulig å validere riktig svar på et innsendt spørsmål

Årsaken til at det ble besluttet å løse begge disse delmålene på samme side i applikasjonen er fordi de begge krever muligheten til å inspisere et enkeltspørsmål om gangen. Derfor vil de måtte dele mye av samme funksjonalitet.

**Delmål 1** blir oppfylt ved at Hjem-siden viser en liste med spørsmål og tilhørende svar (venstre kolonne i figur 4.3) Figur 4.3 representerer den endelige løsningen for produktet. For at brukeren skulle slippe å lære seg chatbotens forhåndsdefinerte svar eller skifte til en annen side for å indentifisere svaret, ble det også inkludert en mindre liste på siden der brukeren kunne finne svarteksten til hvert av svaralternativene. For å gjøre det enkelt å finne frem blant svaralternativene ble det kun vist den tilhørende teksten dersom brukeren klikket på kortet med svaralternativet. Et eksempel på dette blir vist i figur 4.3 med «Avsluttende Samtale».



Figur 4.3 Hjem-side – Endelig løsning

For å skape ekstra plass for informasjonen til hvert spørsmål, ble det gitt mulighet til å klikke på et spørsmål for å utvide spørsmålet til å vise mer informasjon (figur 4.4). Her kan brukeren undersøke om chatboten har gitt rett svar, og eventuelt rette svaret før spørsmålet valideres. Med denne funksjonaliteten blir **delmål 2** oppfylt.



Figur 4.4 Utvidet spørsmål

Det har blitt gjort noen små endringer fra prototypen som er vist i figur 4.5 til den endelige løsningen som er vist i figur 4.3. Blant annet inkluderer den endelige løsningen en ekstra filterfunksjon som setter begrensninger på maks antall spørsmål som kan vises samtidig. I tillegg er det tatt i bruk fargekoder for å skille de ulike type spørsmålene; hvitt er ikke-validerte, grønt er validerte og blått er validerte spørsmål som chatboten har blitt trent på. I den endelige løsningen er fremvisning av statistikk derimot fjernet fra Hjem-siden, som skissert i prototypen.

The screenshot shows the FIB web application interface. At the top, there are navigation tabs: "Hjem", "Administrer kategori", "Tren", and "Statistikk". A "Logg ut" button is in the top right corner. Below the navigation, a message states: "På denne siden vises nye spørsmål som har kommet inn. Når en ser over spørsmålene går en over og sjekke om svarene chatboten har gitt er rett. Vist de er feil så endre en det til rett kategori. Husk å lagre når en er ferdig med å rette opp i spørsmålene."

The main content area features a filter section with the following controls:

- Kategori: Alle (dropdown)
- Dato(fra-til): 01.01.22 (dropdown) to 25.01.23 (dropdown)
- Vis: Ikke validert (dropdown)
- Sorter etter: Dato (dropdown)

Below the filters is a table of questions:

Dato	Spørsmål	Svar	
<input type="checkbox"/>	19.01.23   Jeg lurer på hvem som skal spille på formiddags...	Ukjent	^
<p>Jeg lurer på hvem som skal spille på formiddagskonserten i grieghallen den 25. juni? Jeg vil veldig gjerne se konserten til Ola Norman.</p> <p>Chatbotens svar: Ukjent Chatbotens sikkerhet: 87%</p> <p>Riktig svar: <input type="text" value="Ukjent"/></p> <p><input type="button" value="Slett"/> Obs! lagre andre endringer før du sletter <input type="button" value="Neste"/> Husk å velg riktig kategori!</p>			
<input type="checkbox"/>	17.01.23   Lykke till	Avsluttende sa...	v
<input type="checkbox"/>	17.01.23   Hva koster billettene til nøtteknekkeren	Finn priser	v
<input type="checkbox"/>	13.01.23   Har Justin Bieber konsert i Bergen til sommeren	Program	v
<input type="checkbox"/>	10.01.23   Hvordan bestiller jeg billetter til svanedansen?	Kjøp av billetter	v
<input type="checkbox"/>	10.01.23   Hva koster billetter til svanedansen?	Finn priser	v
<input type="checkbox"/>	10.01.23   Hvilken forestillinger når denne helgen?	Program	v

At the bottom of the table, it says: "Antall viste elementer: 24" and "Antall merkede elementer: 0" with a "Lagre" button.

On the right side, there is a "Treff-prosent 2023" section showing a green bar at 76%. Below it is a "Fordeling kategorier" section with a pie chart and a list of categories:

- Finn priser v
- Fysisk/digital billett v
- Kjøp av billetter v
- Festspillkort v
- Festival datoer v
- Gavekort v
- Avsluttende Samtale v
- Kleskode v
- Refusjon v
- Program v
- Frivillig arbeid v
- Helsing v
- Ukjent v

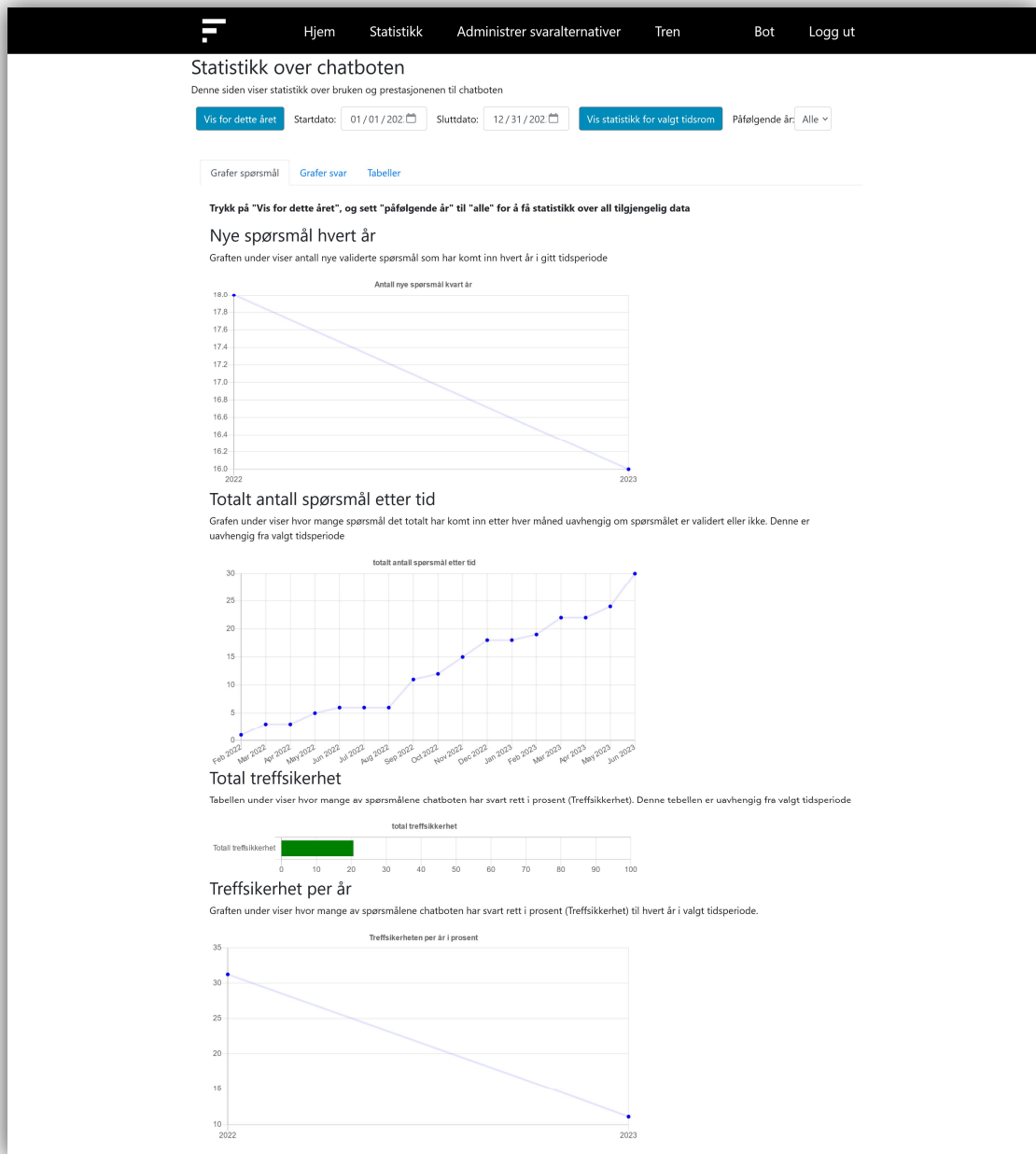
Figur 4.5 Hjem-side – Figma prototype

## 4.2.2 Statistikk

Hensikten med statistiksiden er å fremvise statistikk over prestasjonen til chatboten, og dekker **delmål 4**: Det skal være mulig å spore prestasjonen til chatboten.

Statistiksiden er delt inn i tre ulike faner, der første fane viser ulike grafer over chatbotens generelle prestasjoner, mens fane nummer to ser på hvordan de ulike svaralternativene presterer. Den siste fanen viser en blanding av statistikken fra de to første fanene, men i tabellform. Øverst på statistiksiden er det en felles meny som kan bli brukt til å velge ut et spesifikt tidsrom, og for hvor mange år det skal vises statistikk for. Denne er satt til et helt år som standard.





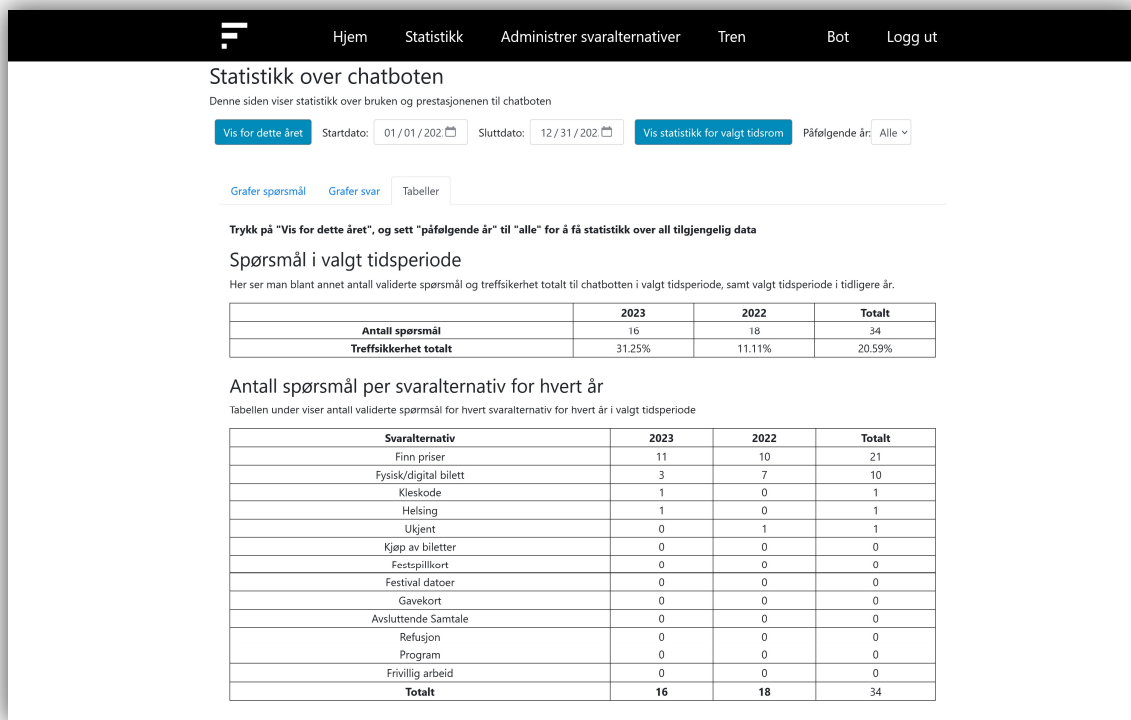
Figur 4.6 Statistikkside: Grafer spørsmål – Endelig løsning

Figur 4.6 viser den første av de tre statistikkfanene. Denne inneholder fire diagrammer der det første viser totalt antall nye spørsmål per år (innenfor valgt tidsrom), det andre viser totalt antall spørsmål måned for måned, det tredje viser den totale treffsikkerhet på alle validerte spørsmål, og det siste viser gjennomsnittlig treffsikkerhet per år, innenfor valgt tidsrom.



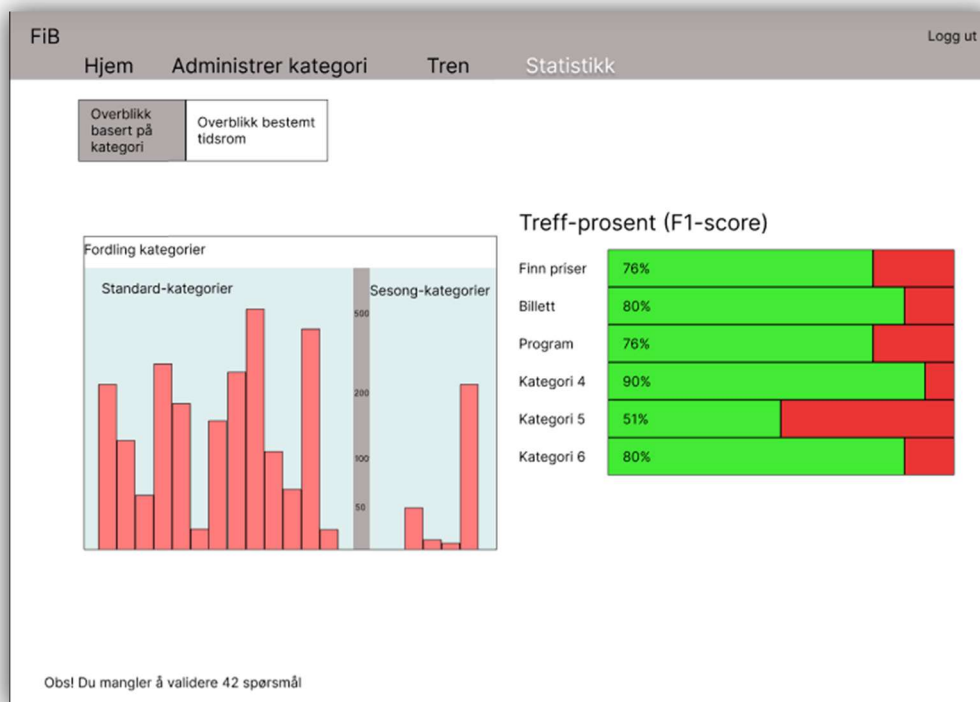
Figur 4.7 Statistikkside: Grafer svar – Endelig løsning

Figur 4.7 viser fane to av statistikkssidene. Denne inneholder kun tre diagrammer. Ett for treffprosenten, F1-score, for hver av svaralternativene innenfor valgt tidsrom, ett med F1-score for hver av svaralternativene år for år, og ett med totalt antall spørsmål per svaralternativ, innenfor valgt tidsrom.



Figur 4.8 Statistikkside: Tabeller – Endelig løsning

Figur 4.8 viser den siste fanen, som inneholder to tabeller. En med antall spørsmål og treffsikkerhet per år, og en med antall spørsmål for hvert svaralternativ per år, der begge viser for valgt tidsrom.



Figur 4.9 Statistikkside 2 – Figma prototype

Den endelige løsningen har flere visuelle endringer fra prototypen, og omfatter mer statistikk. I tillegg inneholdt prototypen to faner i stedet for tre. Figur 4.9 viser den første fanen fra prototypen, der to av grafene er videreført til den endelige løsningen. Disse er første og siste graf i figur 4.7.

Den andre fanen til prototypen er vist i figur 4.10. Den endelige løsningen har implementert all statistikken fra figur 4.10, med unntak av radene «treffsikkerhet sesongsvar» og «treffsikkerhet botsvar».

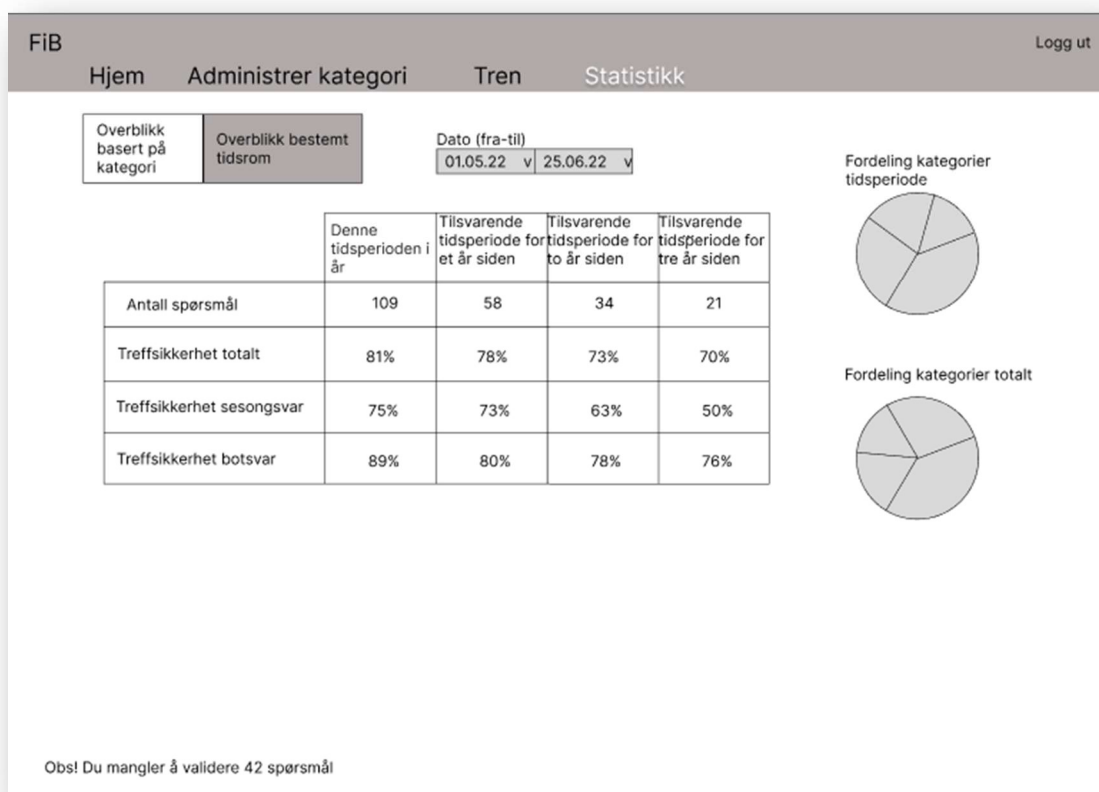


Figure 4.10 Statistikkside 1 - Figma prototype

### 4.2.3 Administrer svaralternativer

Hensikten med siden for å administrere svaralternativer er for å dekke FiB sitt behov for å kontrollere hva chatboten svarer. Denne siden dekker **delmål 5**: Det skal være mulig å endre og legge til svaralternativer for chatboten.

Den endelige «Administrer svaralternativer»-siden kan ses i figur 4.11. På denne siden er alle svaralternativene til chatboten ramset opp i to lister, der de er delt opp i to kategorier. Den ene kategorien inneholder svarene som fantes fra før, som blir kalt for standardsvar. Standardsvar inneholder svarene som maskinlæringsmodellen kan velge mellom. Den andre kategorien er ny, og blir kalt for sesongsvar. I stedet for å bruke maskinlæringsmodellen for å gi svar, bruker sesongsvar nøkkelord for å finne rett svar til et spørsmål. I figur 4.11 er «Covid-19» brukt som et nøkkelord for å trigge «Covid-19» svaralternativet.

De to typene svaralternativ har fått hver sin liste. På hver av listene kan brukeren klikke på hvert svaralternativ for å få opp mer informasjon om svarene. På denne siden er det også mulighet for å endre på svaralternativene.

The screenshot shows a web application interface for managing chatbot responses. At the top, there is a navigation bar with links for 'Hjem', 'Statistikk', 'Administrer svaralternativer', 'Tren', 'Bot', and 'Logg ut'. The main heading is 'Administrer svaralternativer', with a sub-heading explaining that the page provides an overview of chatbot responses and allows for editing and adding new ones.

The interface is divided into two main sections: 'Standard svaralternativer' and 'Sesongsvar'. 'Standard svaralternativer' lists responses based on training data, while 'Sesongsvar' lists responses based on keywords. Both sections show a list of existing responses with details like text, URL, and a 'Visningstekst' (preview) button. A 'Deaktiver' button is also present for each response.

Below the lists, there is a section for adding a new seasonal response ('Legg til nytt sesongsvar'). This section includes a 'Navn' (Name) field, a 'Svar-tekst' (Response text) area, and a 'Legg til ny sesongsvar' button. At the bottom of the page, there is a 'Lagre endringer' (Save changes) button.

Figur 4.11 Administrer svaralternativer-side – Endelig løsning

Det har ikke blitt gjort store endringer på «Administrer svaralternativer»-siden fra prototypen til det endelige produktet. Prototypen kan ses i figur 4.12. Det har blitt gjort noen små endringer for å gjøre løsningen enklere å implementere og mer brukervennlig. Et eksempel på å gjøre siden mer brukervennlig var å endre navnet fra standard kategori til standard svaralternativ.

FIB Logg ut

Hjem Administrer kategori Tren Statistikk

På denne siden ser du en oversikt over svar-kategoriene chatboten bruker, og har mulighet til å redigere og legge til nye av disse. Kategoriene er delt inn i to typer, standard-kategorier og sesong-kategorier

### Standard kategori

En standard-kategori er en kategori chatboten er trent på

**Finn priser** ▲

Navn:  Database navn:

Respons:

Lagre

- Fysisk/digital bilett ▼
- Kjøp av biletter ▼
- Festspillkort ▼
- Festival datoer ▼
- Gavekort ▼
- Avsluttende Samtale ▼
- Klaskode ▼

Ny Standar-kategori

### Sesong kategori

En sesong-kategori er en kategori som er knyttet opp mot nøkkelord. Alle spørsmål som inneholder et av nøkkelordene til en sesong-kategori vil gi det tilhørende svaret

**Covid 19 info** Aktiv ▲

Navn:  Database navn:

Respons:

Prioritert stikkord (max 5)

Covid	X	Vist et av stikkordene finnes i spørsmålet, vil dette svaret bli gitt
Koronavirus	X	
Korona	X	
negativ test	X	
Covid-19	X	

Aktiv

Slett Lagre

- Justin Bieber Inaktiv ▼
- Program 2023 Aktiv ▼

Ny Sesong-kategori

Endringer for kategorier skal ikke endre essensen av spørsmålet, men heller gjøre spørsmål tydligere eller inkludere flere relevante stikkord !!!

Figur 4.12 «Administrer svaralternativ»-side – Figma prototype

Når brukeren trykker på et av svarene blir svaret utvidet, og gir muligheten til å gjøre endringer på det. Ved endring av svarteksten til et svar måtte lenker bli skrevet i HTML format. For at brukeren skulle slippe dette ble det lagt til to inntastingsfelt og en knapp. Det ene feltet var for teksten som skulle representere lenken, og det andre feltet for selve lenken. Når brukeren klikket på knappen, ble disse feltene brukt til å lage en lenke i HTML format for dem. Denne funksjonen ble oversett under utviklingen av prototypen, men ble implementert i den endelige løsningen som kan ses i figur 4.11. Det ble i tillegg lagt til en funksjon for å deaktivere og aktivere de ulike svaralternativene. Et deaktivert svaralternativ vil hindre chatboten i å bruke svaret.

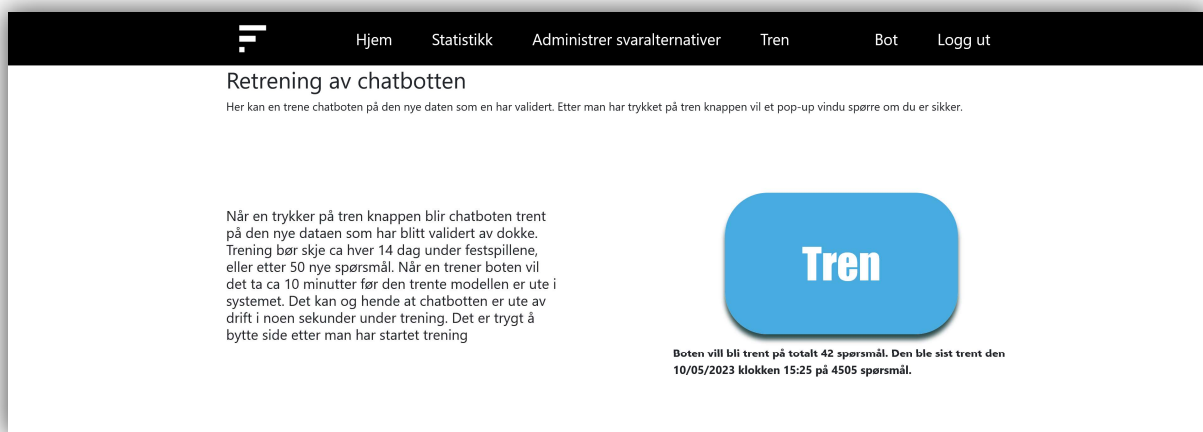
Figur 4.13 Sprettoppvindu vindu for å lage sesongsvar – Figma prototype

En større endring som ble gjort, var å flytte opprettelsen av et sesongsvar fra å være et sprettoppvindu i prototypen, til å være et skjema i bunnen av siden. Sprettoppvinduet fra prototypen kan ses i figur 4.13.

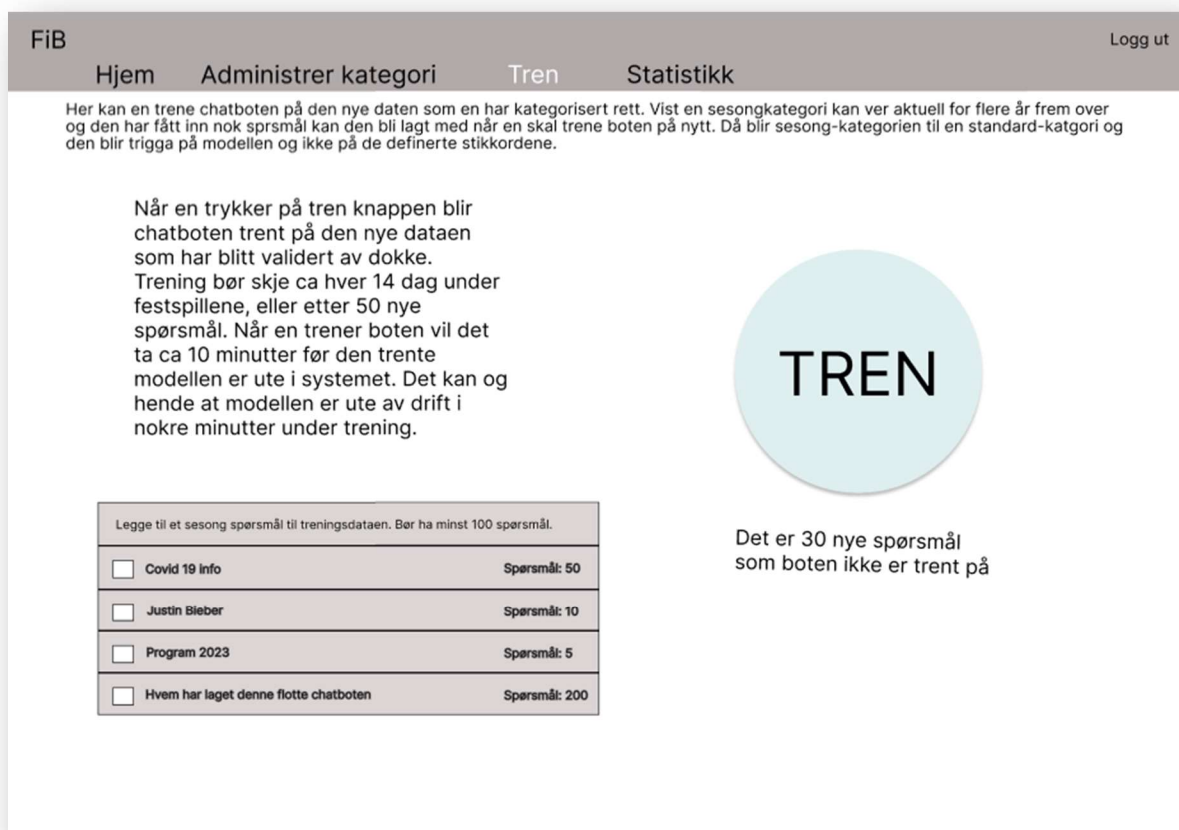
#### 4.2.4 Tren

Tren-siden skal gjøre det mulig å trene chatboten på nytt etter validering av spørsmål, og dekker **delmål 3**: Chatboten skal kunne trenes på nye spørsmål som er validert i verktøyet.

Tren-sidens hovedformål er å trene chatboten til å velge riktig svaralternativ på innkommende spørsmål. Hele operasjonen for å trene chatboten på nytt, gjennomføres ved å trykke på en knapp. Denne knappen er gjort stor for å fange blikket til brukeren. For å informere brukeren om hva det vil innebære å trene chatboten, har det blitt lagt til en del informasjon om dette. Den endelige løsningen til tren-siden kan ses i figur 4.14.



Figur 4.14 Tren-side – Endelig løsning



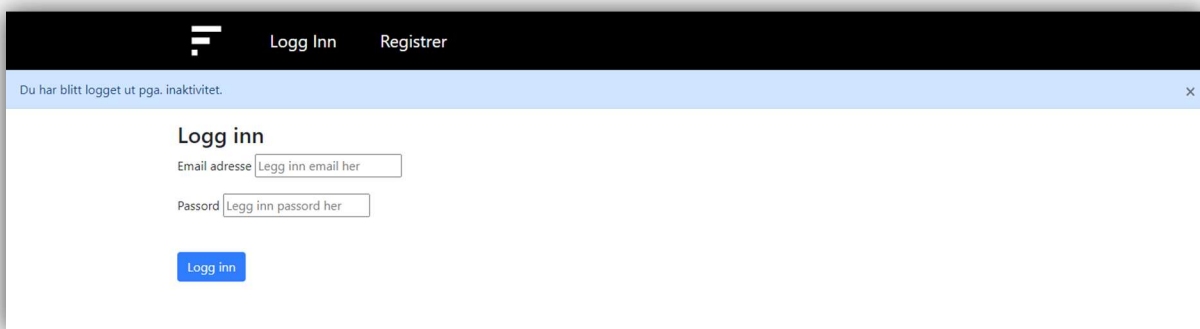
Figur 4.15 Tren-side – Figma prototype



Som nevnt i tidligere kapittel, 4.2.3 Administrer svaralternativer, har Sesongsvar blitt introdusert som en ny type svaralternativ. Ut fra prototypen som kan ses i figur 4.15 skulle brukeren kunne huke av for å inkludere sesongsvaret i treningsdataen, og på denne måten gjøre sesongsvaret om til et standard svar. Grunnet tidsmangel ble det ikke tid til å implementere dette.

## 4.2.5 Logg inn og Registrer

Det ble laget en «logg inn»-side for brukeren til å logge seg inn i applikasjonen, samt en «registrer»-side for nye brukere. Figur 4.16 viser at brukeren blir logget ut etter å ha vært inaktiv i mer enn 30 minutter. Det er verdt å merke seg at for den endelige løsningen vil Registrer-siden, figur 4.17, kun være tilgjengelige for brukere som er innlogget. Årsaken til dette er at FiB-CA skal være et lukket program som kun kan brukes av FiB-ansatte. Grunnen til at siden nå ikke trenger innlogging er at den midlertidige databasen ikke har noe redigeringsverktøy, og det derav ikke ville vært mulig å opprette en ny bruker i en tom database.



Figur 4.16 «Logg inn»-side – Endelig løsning

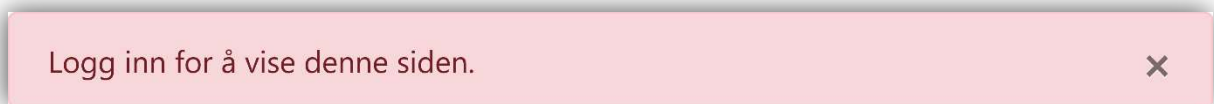


Figur 4.17 Registrer-side – Endelig løsning

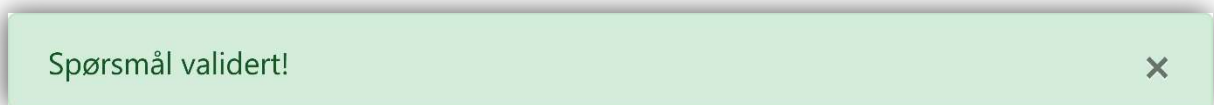
## 4.2.6 Ekstra funksjoner

Det har blitt lagt til flash-meldinger i den endelige løsningen, som er en form for varsler som dukker opp på toppen av siden. Meldingene skal være med på å gi brukeren beskjed om hva som skjer, og fargen er med på å si hva slags type tilbakemelding det er. Fargen blå uttrykker en nøytralt varsl, der figur 4.16 viser et eksempel på dette, mens rødfargen i figur 4.18 uttrykker at noe uønsket har skjedd. Får brukeren derimot en grønn melding som i figur 4.19 forteller det at noe ønsket har skjedd.

Når brukeren validerer spørsmål, sletter spørsmål eller trener chatboten på nytt, vil aktiviteten bli logget. Dette er for å ha muligheten til å spore tilbake til ønsket hendelse.



*Figur 4.18 Negativ tilbakemelding*



*Figur 4.19 Positiv tilbakemelding*



## 5 Evaluering og resultat

### 5.1 Evalueringsmetode

Som beskrevet i kapittel 3.7 Evalueringsplan, er det viktig med jevnlig brukertester og systemtesting av produktet. Ved hjelp av systemtesting og brukertesting kan en verifisere og validere applikasjonen på en god måte.

#### 5.1.1 Brukertesting

Ved hjelp av brukertestene kan en validere om produktet dekker FiB sine behov, og lever opp til forventningene til FiB. Brukertestene blir gjennomført ved at ideen til produktet vises frem, og gruppen mottar tilbakemeldinger på nødvendige endringer. Tilbakemeldingene vil også bidra til en mer effektiv prioritering av de gjenværende oppgavene. Det var i utgangspunktet planlagt fire brukertester: en visuell idéskisse, to tester under utviklingen, og en sluttevaluering.

FiB vil være med i den første brukertesten, hvor prototypen blir vist. Prototypen utvikles i Figma, som er beskrevet i kapittel 3.4.3 Figma. Ved hjelp av prototypen kan gruppen visuelt vise FiB hvordan de har oppfattet kravene gruppen har fått, og hvordan kravene er implementert med funksjoner i et grensesnitt. Målet med å vise prototypen er å la FiB gi tilbakemeldinger på om funksjonene er oppfattet rett, og om det er noen endringer som må gjøres før implementeringen. Et annet ønske er å få fram hvilke deler av applikasjonen som skal ha høyest prioritering under utviklingen.

Etter at utviklingen av applikasjonen har startet, vil gruppen ha to brukertester. Under brukertestene vil testsubjektene være FiB-ansatte, som er fremtidige brukere av applikasjonen. Brukertestene vil skje etter iterasjon to og tre, der FiB-ansatte tester ut funksjonalitetene som har blitt implementert så langt. Brukertestenes første trinn handler om observasjon, som er nevnt som et viktig punkt i brukertesting i kapittel 3.7.1 Brukertesting. I første trinn forklarer gruppen hvilke funksjoner som er implementert, og etterpå gis testsubjektene oppgaver som tar i bruk funksjonene. Underveis i testingen vil gruppen observere hvordan testsubjektene utfører oppgavene. Målet med observasjonen er å finne ut om applikasjonen er intuitiv, slik at det er enkelt å utføre oppgavene og manøvrere seg på nettsiden. Trinn to handler om veiledning til hvordan testsubjektene utfører oppgavene som de eventuelt ikke fikk til eller utførte på feil måte. Målet med trinn to er å avklare hvordan misforståtte eller uopnådde oppgaver utføres slik at gruppen kan få tilbakemeldinger fra testsubjektene på hvordan dette kunne ha blitt gjort bedre. Gruppen vil også finne ut om det var funksjonaliteter som testsubjektene så etter, men som ikke finnes. I trinn tre, spørsmål, stilles testsubjektene generelle spørsmål om applikasjonen med den hensikt å få testsubjektene til å dele tilbakemeldinger om de ulike delene av applikasjonen. Målet med trinn tre er å få en felles forståelse av hvordan testsubjektene oppfattet ulike deler av applikasjonen, samt eventuelle mangler. Et siste punkt under brukertestene er å få oversikt over hvilke funksjoner som er viktigst for FiB at blir implementert, slik at gruppen kan prioritere oppgavene rett.

Når siste iterasjon er ferdig vil det gjennomføres en sluttevaluering der gruppen presenterer for FiB hva som har blitt gjort og hva det ikke ble tid til å gjøre. Målet er å la FiB gi tilbakemelding på hvorvidt det endelige produktet samsvarer med deres krav, behov og forventninger. Mot slutten av bacheloroppgaven vil også Marek Vetter fra Bouvet gi en sluttevaluering. Marek er gruppens veileder og kontaktperson fra Bouvet.

### 5.1.2 Enhetstesting

Som beskrevet i kapittel 3.7.2 Enhetstesting, vil det bli lagt vekt på å skrive enhetstester for å sikre en robust applikasjon. Enhetstestene skal dekke testing av hjelpefunksjoner for applikasjonen. For å oppnå dette, vil testene bli skrevet ved hjelp av "Flask-Testing", en utvidelsepakke som er spesialdesignet for Flask-rammeverket. Denne pakken gir mulighet for enhetstesting av Flask-applikasjoner, noe som gjør det enklere å validere og verifisere funksjonaliteten til hver enkelt komponent i applikasjonen. (FlaskTesting, 2017)

## 5.2 Evalueringsresultat

### 5.2.1 Brukertester

For å få utbytte av brukertestene er det viktig å få med seg både positive og negative tilbakemeldinger om produktet, for å deretter kunne forbedre løsningen basert på disse. Utviklingen av produktet både startet senere og gikk fortere enn det gruppen hadde planlagt. Det ble derfor ikke hold brukertestene på de planlagte tidspunktene etter iterasjonsplanen.

#### Brukertest 1

Under brukertest 1 ble prototypen som er beskrevet under kapittel 3 Prototyper, vedlegg 9.1.3 *Kravdokument*, vist for FiB. FiB hadde flere gode tilbakemeldinger om prototypen, og uttrykte at den samsvarte og dekket deres visjon og behov. Prototypen viste også ekstrarfunksjonen sesongsvar, som fikk gode tilbakemeldinger fra FiB. Sesongsvar har derfor blitt lagt til som ekstra mål, dette er nevnt under kapittel 3.1 Analyse av krav. FiB savnet derimot en beskrivende tekst om hva de ulike sidene er til, og ønsket å ha statistikk over hvert år i stedet for hver dag. Tilbakemeldingene førte til at det ble gjort justeringer på prototypen, slik at den samsvarte med FiB sine ønsker. Under denne brukertesten ble det også utformet en prioriteringsliste over kravene i samarbeid med FiB. Det står mer om hvordan de forskjellige kravene ble prioritert i kapittel 3.1.2 Krav.

#### Brukertest 2

Brukertest 2 skulle etter planen gjennomføres etter at iterasjon 2 var ferdig, men ettersom det gikk fortere å implementere kravene til hver iterasjon enn antatt, var iterasjon 3 fullført ved brukertest 2. De ulike iterasjonene er beskrevet nærmere i kapittel 3.6.2 Prosjektplan. Som følge av at gruppen hadde kommet lenger i utviklingen av applikasjonen enn planlagt, fikk FiB mulighet til å teste flere funksjoner enn forventet. Under testingen av systemet var FiB positive til at de enkelt kunne se og validere spørsmålene chatboten hadde blitt stilt. De savnet derimot muligheten til å sortere spørsmålene etter tilhørende svar, samt en beskrivelse av funksjonaliteten på de ulike sidene. FiB opplevde applikasjonen som intuitiv og enkel å manøvrere seg i. Gruppen mottok derimot

tilbakemelding på at knappen for å trene chatboten på Tren-siden ikke opplevdes som trykkelig. Statistiksiden hadde på dette tidspunktet store mangler, og det ble derfor ikke lagt vekt på denne siden under testen. Etter brukertesten ble de etterspurte funksjonene lagt til. Brukertesten er beskrevet nøyere i vedlegg 9.1.6 *Skjema for brukertesting 2 FiB*.

### Brukertest 3

Brukertest 3 ble gjennomført som planlagt tidsmessig, men ikke i henhold til iterasjonsplanen. Dette skyldes at utviklingen av applikasjonen gikk raskere enn planlagt. På tidspunktet for brukertest 3 hadde gruppen kommet godt i gang med iterasjon 5. Under testingen av applikasjonen var FiB svært positive til at de nå kunne sortere spørsmålene etter hvilke svaralternativ de pekte mot, og at de hadde tilgang til en rekke ulike former for statistikk som enkelt kunne vises fram for ledelsen. Det var ikke lenger viktige funksjoner som manglet, men det var noen elementer som forstyrret flyten. Eksempler på dette er at når en legger til nye stikkord til et sesongsvar, så lukkes det aktuelle sesongsvaret, og en må åpne sesongsvaret på nytt for å kunne legge til det neste stikkordet. Noe annet FiB savnet var en tilbakemelding fra applikasjonen når noen gjorde endringer. Brukertesten er beskrevet nøyere i vedlegg 9.1.7 *Skjema for brukertesting 3 FiB*.

### Sluttevaluering

Det ble gjort et forsøk på å gjennomføre en sluttevaluering der gruppen skulle presentere prosjektet for en større gruppe FiB-ansatte. Som følge av at april/mai er den travleste perioden for FiB, ble det i stedet avtalt et møte etter at bacheloren er levert. FiB formulerte derfor en skriftlig sluttevaluering, vedlegg 9.1.9 *FiB Sluttevaluering*. Sluttevalueringen tar utgangspunkt i samme løsning som ble vist under brukertest 3. I sluttevalueringen informerer Silje Vestvik, Teamleder for kommunikasjonsteamet hos FiB, om at de verdsatte muligheten for å se spørsmål og chatbotens svar i sanntid, samt raskt kunne forbedre informasjonen på fib.no basert på innkommende spørsmål. Løsningen fikk også ros for å tilby en enkel måte å validere chatbotens svar og forbedre kvaliteten kontinuerlig. Videre mottok gruppen tilbakemelding på at muligheten for å legge til nye, sesongbaserte svar er nyttig for en organisasjon som FiB, som mottar mange spørsmål knyttet til det unike programmet som presenteres hvert år.

Mot slutten av prosjektet mottok gruppen en sluttevaluering fra Marek Vetter, som er vedlagt i vedlegg 9.1.10 *Bouvet Sluttevaluering*. I sluttevalueringen uttrykker Marek fornøyelse med både produktet som har blitt utviklet og prosessen frem til det endelige produktet. Marek skriver at «De har oppført seg profesjonelle overfor kunde og styrt kundedialogen på en god måte».

## 5.2.2 Enhetstesting

Enhetstesting tar for seg testing av hjelpefunksjoner relatert til statistikk, sortering og filtrering av data. Testklassene kan bli kjørt i VS Code og gir svar på om testene var suksessfulle eller ikke. For mer informasjon om testene, samt hvordan de kjøres, se vedlegg 9.1.4 *Systemdokumentasjon*.

Testklassene gjorde det enklere å teste unntakstilfeller, samt å finne feil eller mangler med koden, som igjen bidro til å produsere en mer robust kode. Testklassene var ikke til like stor hjelp under utviklingen som først antatt, men de vil trolig være til stor hjelp dersom applikasjonen skal

videreutvikles. Ved videreutvikling av FiB-CA kan testklassene benyttes av noen som ønsker å endre på funksjoner med en tilknyttet test-funksjon.

## 5.3 Prosjektresultat

Sluttresultatet for prosjektet er en lokal webapplikasjon der ansatte hos FiB kan overvåke prestasjonen til chatboten deres. Hver side i applikasjonen inneholder en forklarende tekst om alle funksjonene på siden, slik at en fersk bruker skal kunne bruke alle funksjonene uten ekstra assistanse.

**Delmål 1** sier at FiB-CA skal være i stand til å vise spørsmål som er stilt til chatboten, samt vise chatbotens svar. Som nevnt i kapittel 2.2 Avgrensninger, er FiB-CA per dags dato ikke satt opp på serveren med chatboten til FiB, og har derfor ikke tilgang til de nye spørsmålene chatboten mottar, eller hva den svarer. Det første delmålet ble likevel ansett som oppfylt etter iterasjon 1, da FiB-CA var i stand til å vise alle spørsmål og svar som befant seg i den ny-designede databasen laget for applikasjonen.

Etter at en bruker har logget inn i FiB-CA vil de bli sendt til hjem-siden der de har tilgang til alle spørsmålene som har blitt stilt til chatboten. Det vises som standard 20 spørsmål, men her kan brukeren velge antall spørsmål som skal bli vist. Det er også andre filtre for visning av spørsmål (kapittel 4.2.2 Statistikk). Denne funksjonaliteten samsvarer med **krav 1** fra kapittel 3.1.2 Krav, og kravet anses derfor som oppfylt.

**Delmål 2** angir at det skal være mulig å validere riktige svar på et innsendt spørsmål. Dette delmålet bygger på delmål én i den forstand at en må kunne se et spørsmål for å kunne validere om det er riktig eller galt. Som nevnt over har FiB-CA for øyeblikket ikke tilgang til spørsmålene chatboten mottar. Applikasjonen er derimot i stand til å endre riktig svar og validere om et svar er riktig for alle spørsmålene applikasjonen har tilgang til. Dette ble implementert i iterasjon 2, og delmålet har vært ansett som fullført etter den iterasjonen.

Når en bruker er inne på hjem-siden kan brukeren validere et spørsmål ved å åpne spørsmålskortet, velge riktig svar dersom chatbotsvaret var feil, og deretter trykke «valider». Brukeren kan også validere ved å huke av sjekkboksen til spørsmålskortet. Når brukeren trykker på «lagre endringer» er spørsmålet validert. Dette oppfyller **krav 2** fra kapittel 3.1.2 Krav.

**Delmål 3** innebærer at chatboten skal kunne trenes på validerte spørsmål i FiB-CA. Dette var hovedfokuset for arbeid under iterasjon 3, og var ansett som oppfylt i iterasjon 3. Det ble senere oppdaget at treningen av chatboten ikke fungerte, men dette ble korrigert i iterasjon 5. Chatboten trenes kun på spørsmål FiB-CA har tilgang til, så det er viktig at all relevant treningsdata er tilgjengelig for at trening av chatboten skal fungere optimalt.

Funksjonaliteten for **delmål 3** finner du på tren-siden. Denne siden inneholder forklaring av hvordan chatboten skal trenes og hvor ofte dette burde gjøres. Funksjonaliteten til **delmål 3** er også beskrevet av **krav 3** fra kapittel 3.1.2 Krav.



**Delmål 4** tilsier at det skal være mulig å spore ytelsen til chatboten. For gjennomføringen av denne oppgaven må to forhold ligge i grunn; at FiB-CA har tilgang til spørsmål fra tidsperioden brukeren vil spore ytelse, og at spørsmål fra denne tidsperioden er validert. Disse spørsmålene må være validert for at FiB-CA skal kunne si noe om hvorvidt chatbotens prediksjoner var riktig eller ikke. For dette prosjektet ble ytelse vist gjennom egne sider dedikert til å vise statistikk. Dette ble jobbet med på tvers av iterasjon 3, 4 og 5.

På statistiksiden vil brukeren få en oversikt over statistikk og ytelsen til chatboten, og brukeren kan velge hvilken tidsperiode de vil se statistikk for. Statistiksiden inneholder 3 faner: «Grafer spørsmål», «Grafer svar» og «Tabeller». «Grafer spørsmål» viser statistiske grafer over spørsmål som har blitt stilt til chatboten, «grafer svar» viser statistikk over ytelsen til svarene som chatboten gir, samt antall svar fra hvert svaralternativ. «Tabeller» viser antall spørsmål og svarenes treffsikkerhet for valgt tidsrom og tidligere år. Fanen inneholder også en tabell som viser antall svaralternativer som ble brukt for valgt tidsrom sammenlignet med tidligere år. Dette oppfylder **krav 5 og 6** fra kapittel 3.1.2 Krav.

**Delmål 5** angir at det skal være mulig å endre og legge til svaralternativer for chatboten. Det å endre svaralternativer var enkelt ettersom svaralternativene ble lagret i en database i stedet for å være hardkodet i iterasjon 0. Dette gjorde det enkelt å endre både svartekst og titler for svaralternativene. Denne funksjonen er etterspurt i **krav 4**, og ble fullført i iterasjon 4. I tillegg ble det i iterasjon 5 gjort det mulig å legge til sesong-svaralternativer, som etterspurt i **krav 8**. Det ble ikke tid til å gjennomføre **krav 7**, som var å legge til nye standard-svaralternativer. På grunn av dette er **delmål 5** kun delvis oppfylt.

På «Administrer svaralternativer»-siden kan brukeren endre på navnet og svaret til et svaralternativ, samt gjøre svaret inaktivt. Brukeren kan også opprette et nytt sesongsvar med tilhørende nøkkelord som svaret skal trigges av. Videre loggfører FiB-CA hver gang en bruker validerer eller sletter spørsmål og når chatbot-modellen blir trent på nytt. Dette dekker **krav 9**.

Det eneste kravet som enda ikke er nevnt i dette kapittelet er **krav 10**. Dette ba om en mulighet for å gi brukeren av chatboten mulighet til å gi tilbakemelding på om svaret var til hjelp eller ikke. Dette ble ikke gjennomført på grunn av tidsbegrensninger.

Alle disse delmålene har vært med på videre definere **hovedmålet**, som er å utvikle et administrasjonsverktøy for FiB sin chatbot som kan brukes av ansatte i FiB. Løsningen er per dags dato ikke implementert og den kan derfor for øyeblikket ikke benyttes av ansatte i FiB. Ettersom dette er det beskrevet som en begrensning i kapittel 2.2 Avgrensinger, anses hovedmålet som oppnådd

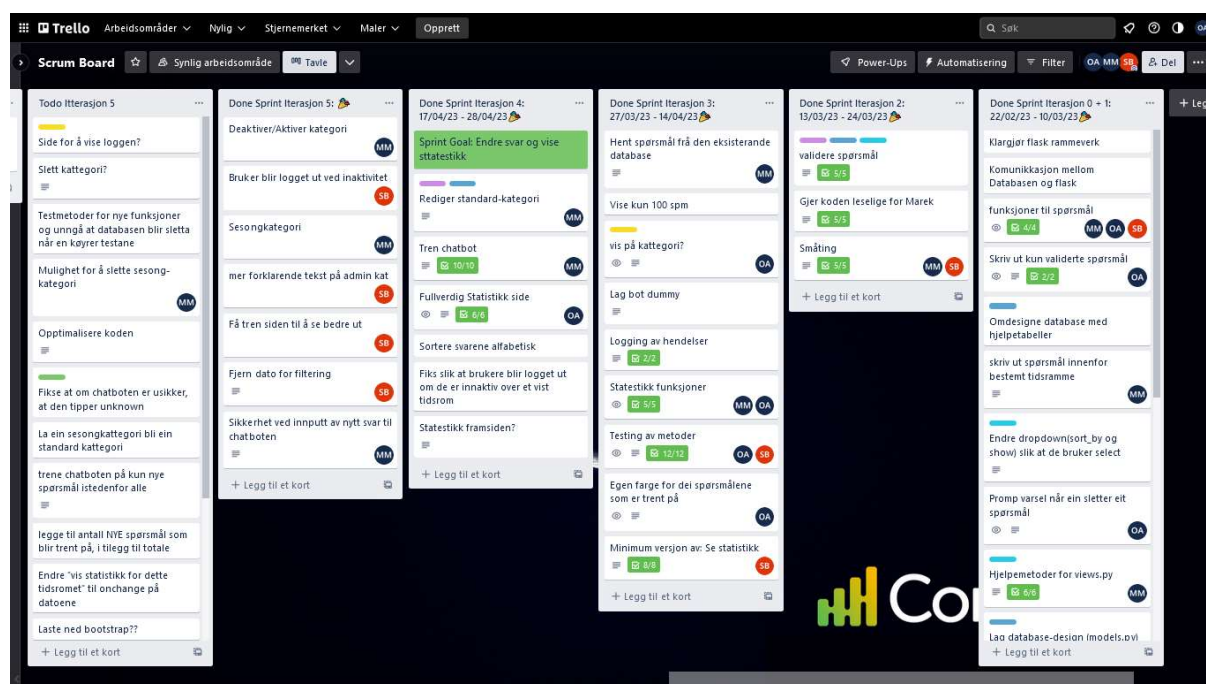
## 5.4 Prosjektgjennomføring

Ved bruk av iterasjonsbasert utvikling klarte gruppen å prioritere de viktigste funksjonalitetene først, og deretter bygge på med nye funksjonaliteter. Trello ble brukt for å få oversikt over hvilke oppgaver som hørte til hvilken iterasjon, statusen til de ulike oppgavene, samt gruppens arbeidsfordeling. Trello er nærmere beskrevet i kapittel 3.5.4 Trello, og et utsnitt av programmet kan ses i figur 5.1.

Kombinasjonen av den iterasjons-baserte utviklingen og Trello gjorde at gruppen hadde god oversikt over hvilke oppgaver som kunne og burde gjøres, som igjen førte til et økt utviklingstempo.

Underveis i utviklingen var det situasjoner der gruppen nesten var ferdig med en iterasjon, slik at det ikke på en god og effektiv måte var arbeid til alle. Dette løste gruppen med å begynne med neste iterasjon mens den tidligere iterasjonen ble ferdigstilt.

Som beskrevet i kapittel 3.6.2 Prosjektplan, forventet gruppen kun å få gjennomført fire iterasjoner, men som følge av at utviklingen av produktet gikk raskere enn først antatt fikk gruppen mulighet til å starte på en femte iterasjon. Hovedgrunnen til dette var godt strukturert og effektivt arbeid med produktet, i tillegg til at gruppen overestimerte omfanget av å sette seg inn i et nytt rammeverk. Sistnevnte skyldes gruppens forsøk på å forebygge risiko 8, som er beskrevet i kapittel 3.6.3 Risikovurdering. Risiko 8 handler om å feilestimere tidsbruken og ta på seg mer arbeid enn det som er gjennomførbart. GANTT-diagrammet som følger i vedlegg 9.1.1 *Prosjekthåndbok*, viser en oversikt over gruppens planlagte og faktiske tidsbruk per iterasjon.



Figur 5.1 Oversikt over Trelloen som er brukt av gruppen

For å tidlig sikre seg tilgang til nødvendig programkode fra FiB, beskrevet som en høy risiko i vedlegg 9.1.1. *Prosjekthåndbok*, måtte gruppen først få arbeidsmaskiner fra Bouvet. Dette førte til at gruppen tidlig tok kontakt med Bouvet for å fikse maskiner. Likevel gikk det flere uker før gruppen fikk tilgang til datamaskiner, som følge av at Bouvet støtte på problemer med å opprette nye, digitale brukere i sine interne systemer. Dette medførte heldigvis ingen stor forsinkelse for gruppen.

# 6 Diskusjon

## 6.1 Funksjonalitet og resultat

### 6.1.1 Valg

Under dette prosjektet er det tatt flere avgjørelser som i ulik grad har hjulpet gruppen i utviklingen av produktet. I ettertid vises det tydelig at arbeidet gjort i forkant av utviklingsprosessen har vært vesentlig for det endelige resultatet. I denne tidsrammen ble det gjort flere avgjørelser om valg av verktøy, hjelpemidler, utviklingsmetode og design som har gitt påfallende resultater.

#### Flask

Som nevnt i kapittel 3.2.2 Rammeverk ble rammeverket Flask valgt for dette prosjektet. Flask viste seg å ha mange fordeler som ikke ble avdekket da dette valget ble tatt. Flask gjorde det blant annet mulig å skrive Python, som for-løkker og if-setninger, rett i HTML-templatene. Det var også mulig å sende over verdier fra backend som kunne brukes direkte i denne Python-koden. Dette lot gruppen enkelt vise frem all informasjonen som skulle vises. Å velge Flask tidlig i prosessen viste seg å være verdifullt da det trolig sparte gruppen mye tid, og var avgjørende for at prosjektet har kommet så langt som det har gjort.

Til tross for disse fordelene med Flask, møtte gruppen noen utfordringer med det. Uten tidligere erfaring med Flask, måtte gruppen starte på nybegynnernivå og deretter bygge på med mer avanserte funksjoner. Dette resulterte i at koden ikke var optimalt strukturert mot slutten av utviklingen. En annen utfordring gruppen møtte på var å få lastet inn statistikken når siden ble lastet inn. På grunn av feil bruk av Flask, fungerte ikke den opprinnelige løsningen til dette problemet. Etter å ha prøvd ut flere alternativer ble det funnet en løsning.

Gruppen har derfor reflektert over mulige forbedringer de ville gjort dersom prosjektet skulle blitt utført på nytt. En bedre mappestruktur fordelt på de ulike sidene i applikasjonen kunne ha bidratt til mindre rot i koden, noe som ville ha gjort koden mer lesbar og vedlikeholdbar.

#### Videre valg av verktøy og hjelpemidler

Bootstrap hjalp gruppen med å effektivt forbedre utseendet til FiB-CA. Bruk av Bootstrap-klasser er ikke like fritt som å skrive CSS fra bunnen av, men går mye raskere å bruke. Dette var en stor fordel for dette prosjektet da tid var en av de største begrensningene. I noen tilfeller lot det seg ikke gjøre å utbedre utseendet med Bootstrap, og da ble CSS brukt for å oppnå designet gruppen ønsket.

For godt samarbeid ble GitHub, Trello, Discord og Teams brukt. GitHub ble brukt for å samarbeide med kode, Teams for dokumenter, Trello for å visualisere arbeidsoppgaver, og Discord som generell kommunikasjonskanal. Discord ble brukt som kommunikasjonskanal til fordel for Teams. Årsaken til dette er at gruppen gjennomgående har kommunisert gjennom Discord på hele studiet, og derfor var mer komfortable med å benytte dette som kommunikasjonskanal. Alle gruppemedlemmene

hadde i tillegg tidligere erfaring med de andre nevnte hjelpemidlene. Hjelpemidlene har vært med på å bidra til god kommunikasjon og arbeidsflyt.

### Valg av utviklingsmetodikk

Som nevnt i kapittel 5.4 Prosjektgjennomføring var iterasjonene mindre omfattende enn hva gruppen hadde sett for seg, samt at noen iterasjoner overlappet hverandre. Til tross for at tidsestimaterne for iterasjonene ikke har vært pålitelige, har iterasjonene likevel vært hjelpsomme. Det har blant annet bidratt til å synliggjøre hvilke oppgaver gruppen måtte prioritere til enhver tid for å kontinuerlig ha et produkt å vise til.

Avgjørelsen om å starte neste iterasjon før den foregående iterasjonen var ferdig, er tatt for å unngå ineffektivt arbeid. Dette var for det meste gunstig, men førte noen ganger til at oppgaver ikke ble gjennomført i henhold til hva som burde vært prioritert. I tillegg førte overlappingen til at «Sprint-planning» ble gjennomført etter at sprinten var påbegynt.

### Hjem-side

På hjem-siden ble det introdusert en ekstra filtreringsfunksjon for å bestemme hvor mange spørsmål som skal bli lastet inn på siden. Her kan brukeren velge mellom 20, 40, 100 og alle. Dette ble gjort da det under utviklingsprosessen ble oppdaget at det var tidkrevende dersom applikasjonen lastet inn for mange spørsmål av gangen. En annen fordel med dette er at om det vises få spørsmål av gangen, blir brukeren nødt til å lagre endringene sine for hver gang den vil laste inn nye spørsmål. Denne funksjonen bidrar til at brukerne lagrer arbeidet sitt oftere, og fremmer dermed både effektivitet og datasikkerhet.

En annen endring på hjem-siden var beslutningen om å ikke inkludere statistikken fra prototypen. Dette valget ble tatt med tanke på estetikk og funksjonalitet. Statistikken tok opp unødvendig plass, og bidro til et mindre profesjonelt utseende. I tillegg tok denne informasjonen fokuset vekk fra oppgavene som skulle gjennomføres på hjem-siden. Ved å ekskludere statistikken ble siden mer oversiktlig, noe som kan bidra til økt produktivitet og brukertilfredshet.

### Statistikkside

Ut fra vedlegg 9.1.8 *Problembeskrivelse fra Bouvet*, kom det ikke tydelig fram hvilken type statistikk som var ønskelig. Under utviklingen av prototypen ble det derfor designet et simpelt forslag til statistikk som kunne være ønskelig for FiB. Under den første brukertesten fremsto FiB som fornøyd, og ga noen konkrete tilbakemeldinger på prototypen. I tillegg til de konkrete tilbakemeldingene ga FiB inntrykk av å ønske noe mer, uten å lykkes i å viderefremme hva dette skulle være. Årsaken til dette kan kanskje være at de ikke selv var sikre på hvilken statistikk de ønsket, eller hva som var mulig for gruppen å produsere med tilgjengelig data. Det kan også være fordi de ikke ønsket å presentere gruppen med en uoverkommelig mengde arbeid.

Grunnet usikkerheten bak hvilken statistikk FiB ønsket etter presentasjon av prototypen, ble det brukt tid på å implementere forskjellig statistikk som gruppen anså som nyttige. FiB tok godt imot denne statistikken når de fikk se den under tredje brukertest. Dette endte godt for dette prosjektet,

men kunne potensielt ført til dårlig bruk av tid ettersom gruppen i stor grad gjettet på hva FiB ønsket.

### Administrer svaralternativer

Under utformingen av prototypen ble det opprinnelig planlagt å bruke et sprettoppvindu for å lage et nytt sesongsvar, som er illustrert i figur 4.12. Dette var en funksjon gruppen forsøkte å utvikle i FiB-CA, men som til slutt ble implementert som et skjema på bunnen av «Administrer svaralternativer»-siden. Bakgrunnen for denne endringen var utfordringer knyttet til utviklingen av sprettoppvinduet, kombinert med ønsket om å ikke bruke mer verdifull tid på denne oppgaven. Selv om et sprettoppvindu kanskje ville ha vært en mer intuitiv løsning, var det ingen negative tilbakemeldinger fra FiB om den endelige implementasjonen.

I den endelige løsningen ble det også lagt til en funksjon som gjorde det mulig å aktivere og deaktivere både standard- og sesongsvar. Dette valget ble gjort for å unngå å slette svarene, da det kan skape uregelmessigheter i statistikken. Dersom et standardsvar blir deaktivert, må modellen til chatboten trenes på nytt slik at den ikke lenger foreslår det deaktiverte svaret. I motsetning til dette krever deaktivering av et sesongsvar ikke ny trening, ettersom disse svaralternativene trigges av nøkkelord.

Disse valgene understreker nødvendigheten av å balansere tidsbruk, funksjonalitet og brukervennlighet i utviklingen av produktet. Samtidig viser de også hvordan gruppen håndterte utfordringer og tilpasninger på en effektiv måte.

### Tren

Tren-siden ble ikke helt ferdig. Denne siden manglet muligheten til å huke av sesongsvaralternativer, for å inkludere de i treningsdataen og gjøre de om til standard-svaralternativer. Dette er den eneste delen av **delmål 5** som ikke ble gjennomført. I Kapittel 7.2 blir det forklart hvordan **delmål 5** kan bli implementert, og at det først bør bli implementert en måte å slette svaralternativet på.

Årsaken til at gruppen møtte på vanskeligheter med å slette svaralternativer var fordi omfanget av dette ikke var vurdert riktig. Når dette problemet skulle løses ble det avdekket at sletting av et svaralternativ ville få følgekonskvenser for statistikken og spørsmålene som pekte mot dette svaralternativet. Det var mye usikkerhet rundt hvordan dette skulle håndteres på en god måte. I tillegg til dette var oppgaven en av de siste som ble påbegynt før slutten av prosjektet. Dette resulterte i at det ikke var gjenværende tid til å gjennomføre oppgaven. Det er mer informasjon om hvordan dette kan løses i kapittel 7.2 Videre arbeid.

### Avgrensninger

For dette prosjektet bel det i kapittel 2.2 Avgrensninger, satt en begrensning på at programmet ikke skulle bli satt i produksjon. Denne avgrensningen gjorde at det endelige produktet kun kan kjøres lokalt uten «live» oppdateringer mot chatboten. FiB kan ikke umiddelbart ta i bruk produktet før mindre, men tidkrevende, endringer er gjort. Endringene vil være å sette FiB-CA inn i FiB sitt

eksisterende system, samt gjøre noen få endringer på chatboten. Endringene er nærmere beskrevet i kapittel 7.2 Videre arbeid. Ved å sette denne avgrensningen ble det tid til å implementere de fleste funksjonene som var ønsket av FiB. Uten denne begrensningen ville det trolig ikke vært mulig å implementere ønskede funksjoner, som opprettelse av sesongsvar og statistikk.

### 6.1.2 Problemer

Et av de større problemene gruppen møtte på under utviklingen av applikasjonen var at prediksjonene til den nye chatboten ikke var riktig. Det var lagt inn metoder både for å trene ny modell og å lage prediksjoner med modellen i tredje iterasjon. På dette tidspunktet var det derimot ikke tilstrekkelig treningsdata til at chatbotmodellen kunne gi en bra prediksjon. Det ble derfor ikke avdekket at prediksjonene til chatboten var feil før gruppen fikk tilgang til treningsdataen og kunne forvente prediksjonene skulle samsvare med et riktig svar. Nok treningsdata for å forvente dette var ikke tilgjengelig før i femte iterasjon av utviklingen.

Det ble brukte mye tid på å feilsøke hvordan applikasjonen brukte maskinlærings-modellen for å lage predikasjoner, og hvordan dataen ble preprosessert før modellen ble laget. Feilen lå derimot i det ene treningsscriptet for modellen. Dette var et ferdig script som både var utviklet og testet i forkant av denne oppgaven, og var ikke mistenkt å inneholde feilen.

Problemet med de ukorrekte prediksjonene til chatboten kunne blitt oppdaget tidligere dersom nok treningsdata for å teste treningsscriptet hadde blitt gitt tidligere. Dette er noe gruppen etterspurte tidlig i prosessen, men som ikke ble fulgt opp tilstrekkelig. Tilgang til nok treningsdata er noe som burde vært prioritert å anskaffe tidligere.

Annen informasjon som gruppen ikke hadde tilgang til var svarteksten til de forskjellige svaralternativene for chatboten. Dette er ikke data som var essensiell for å teste hvordan applikasjonen fungerte, men er noe som kunne vært gunstig å ha for en bedre opplevelse under brukertestene. Denne informasjonen ble ikke etterspurt tydelig nok, og det resulterte i at den heller ble hentet ut av FiB sin allerede eksisterende chatbot ved å stille spørsmål som utløste de ulike svarene.

FiB-CA trengte funksjonalitet for å legge til lenker med alternativ tekst når en bruker redigerte svaralternativene. Dette var nødvendig for å kunne inkludere lenker i svartekstene. Ved å åpne opp for dette, kunne brukere av FiB-CA også legge inn annen HTML og JavaScript-kode i form av HTML-injeksjoner. Dette kunne blitt en sikkerhetsrisiko for både FiB-CA og potensielt FiB.no, ettersom svartekstene fra svaralternativene vil bli vist frem på nettsiden deres. Det ble ikke avdekket noen ferdigdefinert sikkerhetsfunksjon der en kunne velge hvilke HTML-kode som skulle godtas og hvilke den ikke skulle godta. Løsningen endte opp med å være et stikkord-filter, der det ble definert ulovlige ord og tegnsetting. Dersom det blir brukt ulovlig tegnsetting i inntastningsfeltene i FiB-CA, vil dette ikke bli godtatt, samt at det kommer en feilmelding.

### 6.1.3 Resultat

Under behovsavklaringsfasen hos FiB foreslo gruppen at besøkende av chatboten kan ha mulighet til å gi tilbakemelding på svaret de mottok, noe som FiB ønsket. Dette ble ikke implementert på tross av at det kunne vært lønnsomt for FiB-CA. En tilbakemelding fra brukere på hvordan chatboten svarte, kunne gitt FiB-ansatte innsikt i hvorvidt brukere av chatboten var fornøyde med svarene de mottok. I tillegg kunne en slik funksjon bidratt til å validere spørsmål automatisk i FiB-CA, og ansatte ville dermed sluppet å validere spørsmålene manuelt. Det var flere grunner til at en slik tilbakemeldingsfunksjon ikke ble tatt med, men hovedgrunnen var tidsbegrensningen. Andre grunner var at denne oppgaven lå noe utenfor hovedmålet til prosjektet, og at gruppen ikke hadde tilgang til koden for chat-vinduet.

Prototypen som ble laget var til stor hjelp for å formidle gruppens visjon for løsningen, og for å kunne visualisere for FiB hvordan gruppen hadde tolket deres problemstilling og behov. Dette gjorde det enkelt for FiB å sette seg inn i planen for prosjektet, og å påpeke mangler eller overfladisk funksjonalitet. FiB hadde ikke noen innvendinger til løsningsideen som helhet, men ønsket noen mindre endringer. Disse kan leses om i kapittel 5.2.1 under Brukertest 1. Til tross for at FiB ikke hadde noen innvendinger til løsningsideen var prototypen til stor hjelp for å forsikre at prosjektet gikk i riktig vei og til å unngå unødvendig arbeid.

En oversiktlig plan over den endelige løsningen har også gjort det enklere å utvikle applikasjonen med hensyn til senere implementasjoner. Dette har gjort at når mer funksjonalitet skulle bli implementert i applikasjonen, lå det allerede et grunnlag for å implementere dette i den eksisterende koden.

Gruppens løsningsforslag inkluderte gode muligheter for sortering og statistikk. Sorteringsfunksjonene er nyttig for gjennomgang av større mengder data, og for å identifisere utfordringer med FiBs kommunikasjon. Statistikkfunksjonene vil kunne brukes til å gjøre bedre analyser av hva besøkende som bruker chatboten lurer på, spore utvikling over tid, samt å følge med på treffsikkerheten til chatboten.

**Delmål 5** ble ikke oppnådd fullt ut ettersom det ikke er mulig å legge til et nytt standard-svaralternativ. Det var ment å kunne omgjøre et sesong-svaralternativ til et standard-svaralternativ når dette mottok mange nok spørsmål til å kunne inkluderes i treningsdataen for chatboten. Dette ble ikke tatt med i den endelige løsningen på grunn av tidsbegrensninger. Å gjøre en sesong-svaralternativ til et standard-svaralternativ vill fjerne behovet for bruk av stikkord for å filtrere ut spørsmål, ettersom prediksjonen til selve chatboten da i stedet blir brukt. Dette hadde vært gunstig ettersom et stikkord kan forekomme i spørsmål som egentlig tilhører andre svaralternativer, mens prediksjonene til chatboten vil ta hele setningen med i avgjørelsen.



Figur 6.1 Eksempel på hvordan bruker av chatbot kunne gitt tilbakemelding

Målene for iterasjon 5 inkluderte både å muliggjøre at brukerne av chatboten kan gi tilbakemeldinger på mottatte svar, samt å utvikle nye standard svaralternativer. Det er viktig å merke seg at det ble informert til FiB under første brukertest at det trolig ikke ble tid til å fullføre alt som er planlagt.

Både FiB og Bouvet er svært positive til løsningen som er utviklet gjennom dette prosjektet. Dette gjenspeiles i Vedlegg 9.1.9 *FiB sluttevaluering* og Vedlegg 9.1.10 *Bouvet sluttevaluering*, der begge parter uttrykker entusiasme for å ta videre dette prosjektet.

## 6.2 Erfaringer

Gruppen har innad samarbeidet vært svært godt i denne bacheloroppgaven. Gruppen har møtt opp og jobbet sammen til faste tider hver tirsdag, onsdag og fredag. Arbeidsdagene har vært effektive med lite dødtid. Det har vært enkelt for gruppen å diskutere løsningsalternativer og ta beslutninger. Det har vært lav terskel for å spørre om hjelp, og dette er blitt benyttet i stor grad. I ettertid ser gruppen likevel at det kunne lønnet seg med noen planlagte pauser i arbeidsdagen, ettersom kvaliteten og effektiviteten på arbeidet falt mot slutten av dagen.

Gruppen har fulgt en plan fra begynnelsen av prosjektet. Planen ble laget i form av et GANTT-diagram og en iterasjonsplan (vedlegg 9.1.1 *Prosjekthåndbok*), som har bidratt til at alle gruppemedlemmene til enhver tid har hatt oversikt over frister, prioriteringer og neste fokusområde. Planen la grunnlag for at gruppen fikk gjennomført nødvendig grunnarbeid i forkant av utviklingen, noe som har hjulpet for resultatet til prosjektet. Til tider har mindre oppgaver blitt feilprioritert som følge av at oppgaven hadde høyere prioritert i planen, men viste seg senere å være lavere prioritert. Slike hendelser kunne forekomme sjeldnere om gruppen var flinkere til å gjøre endringer i planen når ny informasjon dukket opp. Idéer for muligheter i FiB-CA ble presentert for FiB, noe som gir en forventning til applikasjonen og kan skape misnøye om disse ikke ble gjennomført. Et eksempel på dette finnes i vedlegg 9.1.9 *FiB Sluttevaluering*, der FiB har nevnt mulighet for å lage nye standard-svaralternativer. Dette ble presentert som en ide, men gruppen fikk ikke tid til å implementere det i FiB-CA. Det har ikke oppstått noen problemer for gruppen som følge av dette, og FiB har gjennom prosjektet hatt stor forståelse for gruppens bakgrunn og begrensede tidsramme.

Det er verdt å merke seg at gruppen kunne ha vært flinkere til å sende flere purringer om ressurser, og til å be om tilbakemeldinger fra de involverte partene, Bouvet, FiB og HVL-veileder. Dette kunne ha bidratt til å fremskynde prosessen og å sikre ressurser og tilbakemeldinger på et tidligere stadium i prosjektet. Som et resultat mottok gruppen nødvendige ressurser og tilbakemeldinger senere enn forventet, noe som kan ha påvirket arbeidet og progresjonen i prosjektet.



# 7 Konklusjon og videre arbeid

## 7.1 Konklusjon

Hvert av delmålene 1-4 er oppfylt, delmål 5 er nesten oppfylt, og alle kravene utenom krav 7 og 10 er oppnådd. I kapittel 5.3 Prosjekresultat beskrives det i større detalj hvordan dette er gjennomført.

Tilbakemeldingen fra FiB angående løsningsforslaget til studentgruppen har vært svært positiv. FiB mener at løsningen dekker alle funksjonene som deres webredaktører og kundebehandlere har savnet, og at implementering av løsningen vil redusere frustrasjon og usikkerhet rundt chatboten. Dette vil gi FiB et tydeligere bilde av chatbotens nytteverdi for både kunder og kundeforholdere. FiB har også uttrykt stor tilfredshet med samarbeidet med studentgruppen og deres evne til å lytte, utfordre og tilpasse løsningen basert på tilbakemeldinger. FiB kan enkelt se for seg å ta i bruk løsningen, og tror den vil bidra positivt til arbeidet med kundeforståelse og -analyse, og på sikt forbedre chatbotens evne til å hjelpe kundene.

Bachelorprosjektet hadde som mål å utvikle og designe et intuitivt og brukervennlig administrasjonsverktøy for FiB sin chatbot, for å gjøre det enklere for ikke-tekniske brukere å interagere med, og forstå chatboten. Målet ble oppnådd ved å utvikle et verktøy som oppfylte flere delmål, inkludert visning av spørsmål og svar, validering av svar, trening av chatboten på nye spørsmål, sporing av ytelse, og redigering av svaralternativer.

Det var noen funksjoner som ikke ble fullført på grunn av tidsbegrensninger. En av disse var muligheten til å gjøre om sesong-svaralternativ til standard-svaralternativer. En annen var den manglende tilbakemeldingsfunksjonen fra besøkende av chatboten. Likevel ses prosjektets hovedmål som oppnådd, ettersom FiB sine kunder vil få en bedre brukeropplevelse når de bruker chatboten, takket være FiB-CA.

## 7.2 Videre arbeid

For at FiB skal kunne ta i bruk applikasjonen utviklet i dette prosjektet må Bouvet sette opp en server for å kjøre applikasjonen på. Dette vil trolig foregå på en Microsoft Azure server med en delt database mot den endelige løsningen. Det må også gjøres små endringer i selve applikasjonen for FiB sin chatbot, for at den skal kunne kommunisere med FiB-CA.

FiB-CA er ikke blitt ferdigutviklet og har derfor enkelte mangler. To funksjoner som burde prioriteres å gjennomføre er mulighet til opprettelse av nye standard-svaralternativer, og mulighet for å slette svaralternativer.

**Opprettelse av nye standard-svaralternativer** var planlagt å gjennomføres ved å endre et sesong-svaralternativ. Årsaken til dette er at et standard-svaralternativ er et svaralternativ som blir gitt av maskinlærings-modellen. Dette krever at svaralternativet allerede har en betydelig mengde treningsdata. Planen var at brukeren skulle kunne opprette et sesong-svaralternativ, og endre denne

til et standard-svaralternativ når FiB-CA hadde mottatt nok treningsdata for dette. For å endre et sesong-svaralternativ til et standard-svaralternativ er det nok å endre svaralternativet sin «state» til STANDARD, og trene chatboten på nytt. Det er ikke blitt laget en metode for dette ettersom FiB-CA enda ikke kan slette svaralternativer. Det kan også bli ødeleggende for prestasjonen til maskinlæringsmodellen å ha mange svaralternativer dersom hver av dem ikke har nok treningsdata.

Mulighet til å **slette svaralternativer** var en del av iterasjon 5, men viste seg å være mer utfordrende enn antatt. Årsaken til dette var usikkerhet rundt hva som skulle gjøres med alle spørsmålene som pekte mot dette svaralternativet. Dersom spørsmålene ble fjernet ville dette påvirke statistikken for chatboten, og føre til tap av spørsmål som kunne blitt brukt som treningsdata for et annet svaralternativ. Om spørsmålene blir beholdt vil de enten ende opp med å peke mot et svaralternativ som ikke eksisterer, eller peke mot feil svaralternativ. Dette vil føre til feil på statistikksiden. En mulig løsning er å ikke slette svaralternativet, men å introdusere en ny «state» som sier den skal ekskluderes fra alt utenom statistikk. Foreløpig er det mulig å deaktivere kategorier, men dette gjør kun at chatboten ser bort fra svaralternativene, og alternativene vil fortsatt være synlig og føre til dårligere oversikt i FiB-CA.

Det kunne vært gunstig å ha mulighet til å **gi ulike rettigheter** til ulike brukere av FiB-CA. Dette kunne senket terskelen for innlogging til applikasjonen for FiB-ansatte. Funksjonaliteter som kunne vært gunstig å begrense til færre brukere er opprettelse av nye brukere, legge til å endre nye svaralternativer og å trene selve modellen på nytt. Ettersom FiB har frivillige ansatte under Festspillene ville det vært gunstig at de kun har mulighet til å validere spørsmål.

Dersom en ny side for å **legge til og administrere eksisterende brukere** ble lagt til, kunne dette bidratt til å skaffe en oversikt over hvem som bruker applikasjonen og hvilke rettigheter disse brukerne har.

Det kunne også vært gunstig med en enkel måte for en bruker å **se over hendelser i loggen**. En logg vil vise all aktivitet og endringer som er gjort i FiB-CA. For øyeblikket er ikke loggen for FiB-CA leselig i applikasjonen. Dette er informasjon som ville vært gunstig for en admin-bruker i FiB-CA å ha tilgang til.

Noe som kunne vært til fordel for FiB er å utvide funksjonaliteten slik at brukeren av chatboten kunne **gi en tilbakemelding på svaret til chatboten**. Dette er informasjon som kunne blitt vist frem som statistikk i FiB-CA, og muligens hjulpet med å automatisk validere svaret for å minske arbeidsmengden i FiB-CA.

Med implementering av disse forslagene i videre arbeid, vil FiB være i stand til å ta i bruk denne applikasjonen til neste Festspillene i Bergen i 2024. Ved å ta i bruk administrasjonsverktøyet med de foreslåtte endringene, vil FiB kunne forbedre sin chatbots ytelse og brukervennlighet ytterligere, samt tilpasse chatbotens svar til ulike situasjoner og behov under Festspillene i Bergen 2024.

## 7.3 Generelle læringspunkter

FiB-CA er spesifikt tilpasset FiB sin chatbot, og er ikke et program som kan brukes for en annen type chatbot. Av den grunn kan det være utfordrende og lite hensiktsmessig å tilpasse dette verktøyet til en allerede eksisterende chatbot. Om chatboten skulle være basert på en klassifiseringsmodell, og svaralternativer og innsendte spørsmål kan lagres i en database med samme struktur som den FiB-CA har, vil programmet være mulig å tilpasses slik at det kan brukes for denne chatboten også. Dette er lite sannsynlig ettersom det er stor variasjon i hvordan chatboter er utviklet.

Begrensningen for hvilke chatboter FiB-CA kan samhandle med er basert på chatbotens struktur. Dette åpner likevel for muligheten til å skape en egen klassifiserings-chatbot som er kompatibel med FiB-CA. For å sikre at overskrifter og forklarende tekst i FiB-CA samsvarer med formålet til den nye chatboten, vil det være nødvendig med tilpasninger i FiB-CA. Dette er imidlertid kun mulig for chatboter som benytter en klassifiserings-basert maskinlæringsmodell.

Selve løsningsideen og funksjonaliteten til FiB-CA er noe som kan utvikles for de fleste chatboter. FiB-CA er blant annet med på å forbedre nøyaktigheten til en maskinlæringsmodell gjennom produksjon og bruk av ny treningsdata. De fire paragrafene under beskriver funksjonalitet fra FiB-CA som kan være nyttig for andre chatboter.

**Å forbedre nøyaktigheten** til en maskinlæringsmodell vil være gunstig uavhengig av om den brukes for en chatbot eller en annen løsning. FiB-CA gjør dette gjennom rensing og produksjon av treningsdata. En svakhet med denne løsningen er at det krever et menneske for å validere og gå over dataen. Dette vil i mange tilfeller ikke være gunstig ettersom kostnadene for et menneske til å produsere treningsdata for en stor mengde data vil være høy og ha en minimal innvirkning på nøyaktigheten til en modell med nok treningsdata. Et annet alternativ til FiB-CA sin måte å produsere treningsdata på er å konvertere allerede eksisterende data fra eksterne kilder slik at det kan brukes i en maskinlæringsmodell. Et eksempel på metode for å gjøre dette er DataScraping, som er prosessen der en programvare henter og ekstraherer data fra nettsider. Dette er noe som krever teknisk erfaring.

**Måling og statistikk av prestasjonene** til en chatbot er nyttig for alle maskinlæringsmodeller der det kan oppstå et ønske om å erstatte løsningen, forkaste den, utbedre den eller formidle resultatene den produserer. Statistikken vil bidra til å identifisere styrker og svakheter i chatbot-applikasjonen. Dette vil deretter gi eiere av løsningen muligheten til å justere sin strategiske planlegging, basert på de innsamlede dataene.

**Å overvåke inputdataen** vil lønne seg i mange tilfeller. For en chatbot er dette gunstig for å vurdere hvilke type spørsmål chatboten må kunne være i stand til å svare på, og hvorvidt chatboten er trent på, eller har tilgang til den etterspurte dataen. For andre maskinlæringsmodeller kan det også være gunstig å se om inputdataen er som forventet. Dette kan hjelpe dem å feilsøke om outputdataen ikke er som forventet.

**Generell tilpasning** av en chatbot vil gjøre det enklere for bedrifter som ønsker å tilpasse chatbotens svar til spesifikke kampanjer, sesonger eller hendelser. FiB-CA gjør dette blant annet gjennom

muligheten for å redigere svaralternativer og legge til sesongsvar som utløses av nøkkelord. Dette gjør verktøyet enda mer fleksibelt og tilpasningsdyktig.

## 8 Litteraturliste / referanser

### Avklaring på bruk av ChatGPT:

Chat.openai.com (ChatGPT) er listet opp som en referanse. Denne referansen har kun blitt brukt for tilbakemeldinger og veiledning på rapporten, hjelp til feilmeldinger i utvikling, forståelse av kode og innhenting av informasjon.

ChatGPT har IKKE blitt brukt for å generere tekst til bacheloroppgaven.

## Referanser

- Banerjee, S. (2022, mai 6). *Desktop vs Web Application – What is Better to Develop?* From RS Web Solutions: <https://www.rswebsols.com/tutorials/software-tutorials/desktop-vs-web-application>
- Bootstrap. (2022, August 1). *Bootstrap*. Hentet fra TechTarget: <https://www.techtarget.com/whatis/definition/bootstrap>
- Bouvet. (u.d.). *Bouvet Norge*. Hentet fra Bouvet: <https://www.bouvet.no/>
- ChatBot | AI Chat Bot Software for Your Website*. (u.d.). Hentet fra chatbot.com: <https://www.chatbot.com/>
- Dingsøy, T. (2022, Desember 21). *smidige utviklingsmetoder*. Hentet fra Store Norske Leksikon: [https://snl.no/smidige\\_utviklingsmetoder](https://snl.no/smidige_utviklingsmetoder)
- Discord. (u.d.). *MERKEVARE*. Hentet fra <https://discord.com/branding>
- Festspillene i Bergen. (u.d.). *Om Festspillene i Bergen*. Hentet February 15, 2023 fra Festspillene i Bergen: <https://www.fib.no/om-festspillene/>
- Figma. (2019, Januar 22). Hentet fra <https://commons.wikimedia.org/wiki/File:Figma-logo.svg>
- Figma. (u.d.). *About Figma, the collaborative interface design tool*. Hentet January 24, 2023 fra Figma: <https://www.figma.com/about/>
- Flask. (u.d.). Hentet fra Freebie supply: <https://freebiesupply.com/logos/flask-logo/>
- Flask User's Guide*. (u.d.). Hentet fra Flask: <https://flask.palletsprojects.com/en/2.1.x/>
- FlaskTesting. (2017, Desember 19). *FlaskTesting*. From flask-testing.readthedocs: <https://flask-testing.readthedocs.io/en/latest/>
- Freepik. (u.d.). *flaticon*. Hentet fra [https://www.flaticon.com/free-icon/trello\\_6124991](https://www.flaticon.com/free-icon/trello_6124991)
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2. utg.). O'Reilly.
- GitHub. (u.d.). *GitHub Logos and Usage*. Hentet fra GitHub: <https://github.com/logos>
- Heusser, M. (2020, November 12). *Why is unit testing important for developers?* Hentet fra TechTarget: <https://www.techtarget.com/searchsoftwarequality/answer/Is-unit-testing-an-important-aspect-of-software-development>
- Horry, L., & Bottorff, C. (2022, Mars 25). *What Is Waterfall Methodology? Here's How It Can Help Your Project Management Strategy*. Hentet fra Forbes Advisor: <https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>
- How ProdPad Fits With Product Discovery*. (u.d.). Hentet fra ProdPad: <https://www.prodpad.com/resources/how-prodpad-fits/discovery/>

Howard, J., & Gugger, S. (2020). *Deep Learning for Coders with fastai & PyTorch*. Sebastopol, CA: O'Reilly Media, inc.

IBM. (u.d.). *What is three-tier architecture?* Hentet fra ibm.com: <https://www.ibm.com/topics/three-tier-architecture>

Interaction Design Foundation. (n.d.). *Prototyping*. From Interaction Design Foundation: <https://www.interaction-design.org/literature/topics/prototyping>

Intercom | *Making Internet Business Personal*. (u.d.). Hentet fra intercom: <https://www.intercom.com>

Lomas, A. (2022, oktober 30). *Top 10 Web Development Frameworks for 2023*. From netsolutions.com: <https://www.netsolutions.com/insights/top-web-development-frameworks/>

Lutkevich, B. (2023, Februar). *Github*. From Techtarget: <https://www.techtarget.com/searchitoperations/definition/GitHub>

Microsoft - VS Code. (2022). *Download Visual Studio Code - Mac, Linux, Windows*. Hentet January 24, 2023 fra Microsoft - VS Code: <https://code.visualstudio.com/download>

Microsoft. (2019, Juni 5). From <https://code.visualstudio.com/brand>

Nolledo, M. (2020, Juli 28). *What is Google Drive? A guide to navigating Google's file storage service and collaboration tools*. From BusinessInsider: <https://www.businessinsider.com/guides/tech/what-is-google-drive-guide?r=US&IR=T>

OpenAI. (2023, Mars 18). *ChatGPT\_logo*. Hentet fra Wikimedia Commons: [https://commons.wikimedia.org/wiki/File:ChatGPT\\_logo.svg](https://commons.wikimedia.org/wiki/File:ChatGPT_logo.svg)

Overleaf. (u.d.). *Overleaf*. Hentet January 24, 2023 fra Overleaf, Online LaTeX Editor: <https://www.overleaf.com/>

Ranjan, R. (2022, Desember 7). *Flask vs Django in 2023: Which Framework to Choose and When?* From netsolutions.com: <https://www.netsolutions.com/insights/flask-vs-django/>

Ross, J. (2018, September 10). *The Role of Observation in User Research*. From UXmatters: <https://www.uxmatters.com/mt/archives/2018/09/the-role-of-observation-in-user-research.php>

Rossen, E., & Nätt, T. H. (2022, Desember 28). *API*. Hentet fra SNL: <https://snl.no/API>

Sandnes, F. E. (2011). Universell utforming av IKT-systemer. I F. E. Sandnes, *Universell utforming av IKT-systemer* (s. 385). Oslo: Universitetsforlaget.

Scrum.org. (i.d.). *What is Scrum*. Hentet fra scrum.org: <https://www.scrum.org/resources/what-is-scrum>

Sharp, H., Rogers, Y., & Preece, J. (2019). *Interaction Design: beyond human-computer interaction* (5. utg.). John Wiley & Sons, Inc.

SQLAlchemy. (2023, mai 10). *Features*. From sqlalchemy.org: <https://www.sqlalchemy.org/features.html>

Thorben. (2023, 03 03). *SOLID Design Principles Explained: The Single Responsibility Principle*. From Stackify: <https://stackify.com/solid-design-principles/>

Wong, E. (2021). *Principle of Consistency and Standards in User Interface Design*. From Interaction Design Foundation: <https://www.interaction-design.org/literature/article/principle-of-consistency-and-standards-in-user-interface-design>

# 9 Vedlegg

## 9.1 Vedleggliste

### Støttedokumenter

- 9.1.1 Prosjekthåndbok
- 9.1.2 Visjonsdokument
- 9.1.3 Kravdokument
- 9.1.4 Systemdokumentasjon

### Samarbeidsavtale

- 9.1.5 Samarbeidsavtale undertegnet

### Brukertesting

- 9.1.6 Skjema for brukertesting 2 FiB
- 9.1.7 Skjema for brukertesting 3 FiB

### Problembeskrivelse og sluttevaluering

- 9.1.8 Problembeskrivelse fra Bouvet
- 9.1.9 FiB Sluttevaluering
- 9.1.10 Bouvet Sluttevaluering

## 9.2 Figurliste

Figur 1.1 Skjerm bilde av chatboten til FiB .....	1
Figur 1.2 Bouvet sitt kontor i Bergen .....	3
Figur 1.3 Arbeidslokalet til Bouvet i Bergen.....	3
Figur 2.1 Eksempel på trestruktur .....	11
Figur 3.1 Double Diamond Process (How ProdPad Fits With Product Discovery, u.d.) .....	13
Figur 3.2 VS Code logo (Microsoft, 2019) .....	17
Figur 3.3 Flask logo (Flask, u.d.) .....	17
Figur 3.4 Figma logo (Figma, 2019).....	17
Figur 3.5 GitHub logo (GitHub, u.d.) .....	18
Figur 3.6 Discord logo (Discord, u.d.).....	19
Figur 3.7 Trello logo (Freepik, u.d.).....	19
Figur 3.8 ChatGPT logo (OpenAI, 2023) .....	19
Figur 3.9 Visualisering av iterasjonsplan.....	22
Figur 4.1 Sammenheng mellom FiB-CA og chatbot .....	27
Figur 4.2 Entitetsdiagram til ny og gammel database .....	28
Figur 4.3 Hjem-side – Endelig løsning .....	31
Figur 4.4 Utvidet spørsmål.....	31

Figur 4.5 Hjem-side – Figma prototype .....	32
Figur 4.6 Statistikkside: Grafer spørsmål – Endelig løsning .....	33
Figur 4.7 Statistikkside: Grafer svar – Endelig løsning .....	34
Figur 4.8 Statistikkside: Tabeller – Endelig løsning .....	35
Figur 4.9 Statistikkside 2 – Figma prototype .....	35
Figure 4.10 Statistikkside 1 - Figma prototype .....	36
Figur 4.11 Administrer svaralternativer-side – Endelig løsning .....	37
Figur 4.12 «Administrer svaralternativer»-side – Figma prototype .....	38
Figur 4.13 Sprettoppvindu vindu for å lage sesongsvar – Figma prototype .....	39
Figur 4.14 Tren-side – Endelig løsning .....	40
Figur 4.15 Tren-side – Figma prototype .....	40
Figur 4.16 «Logg inn»-side – Endelig løsning .....	42
Figur 4.17 Registrer-side – Endelig løsning .....	42
Figur 4.18 Negativ tilbakemelding .....	43
Figur 4.19 Positiv tilbakemelding .....	43
Figur 5.1 Oversikt over Trelloen som er brukt av gruppen .....	50
Figur 6.1 Eksempel på hvordan bruker av chatbot kunne gitt tilbakemelding .....	55

### 9.3 Tabelliste

Tabell 3.1 Risiko 8 .....	23
---------------------------	----